Grant Agreement N° 872592

PLATOON

Digital platform and analytic tools for energy

Deliverable D2.5
PLATOON Reference Architecture (v2)

Contractual delivery date:
M12

Actual delivery date:
31st of March 2021

Responsible partner:
P5: ENG, Italy

Project Title	PLATOON – Digital platform and analytic tools for energy
Deliverable number	D2.5
Deliverable title	PLATOON Reference Architecture
Author(s):	Martino Maggio (ENG), Francesco Arigliano (ENG)
Responsible Partner:	P5: Engineering Ingegneria Informatica SpA - ENG
Date:	31.03.2022
Nature	R
Distribution level (CO, PU):	PU
Work package number	WP2 – Reference Architecture, Interoperability and Standardization
Work package leader	ENG, Italy

Abstract:	Deliverable 2.5 describes the second version of PLATOON reference architecture including the logical components in terms of functionalities and interfaces, their relationships and the description of the key processes related to data exchange and security. This document is mainly created to support the developers of reference implementation of PLATOON platform and pilots or other external stakeholders that want to be integrated with PLATOON and use its interoperability capabilities.
Keyword List:	Architecture, Interoperability, Toolbox, Security, Standards, IDS, IoT, Edge, FIWARE, COSMAG

The research leading to these results has received funding from the European Community's Horizon 2020 Work Programme (H2020) under grant agreement no 872592.

This report reflects the views only of the authors and does not represent the opinion of the European Commission, and the European Commission is not responsible or liable for any use that may be made of the information contained therein.

Editor(s):	Martino Maggio (ENG), Francesco Arigliano (ENG)
Contributor(s):	Francesco Arigliano (ENG), Martino Maggio (ENG), Vincenzo Savarino (ENG), Valentin Sanchez Pelaez (TECN), Andrej Campa (CS), KHAN Hammad Aslam (ENGIE), Dileep Ramachandrarao Krishna Murthy (IAIS), Jose Maria Barriga Garcia (INDRA), Kemele M Endris (TIB), Philipp Rohde (TIB)
Reviewer(s):	Philippe Calvez (ENGIE) – Erik Maqueda (TECN)
Approved by:	Philippe Calvez (ENGIE) – Platoon Coordinator Erik Maqueda (TECN) – Technical Coordinator Eduardo Jimenez (IND) – Exploitation Coordinator
Recommended/mandatory readers:	Software designers, software developers, Energy technical managers

Document Description

Document Revision History

Version	Date	Modifications Introduced	
		Modification Reason	Modified by
0.1	01/07/2020	Table of content definition	Martino Maggio, Francesco Arigliano
0.2	27/07/2020	Added initial PLATOON architecture high level overview	Martino Maggio, Francesco Arigliano
0.6	05/10/2020	Updated PLATOON architecture high level overview, added Architecture components description and requirements	Martino Maggio, Francesco Arigliano
0.9	16/11/2020	Updated Architecture components description, Edge capabilities and deployment scenarios. Added Data exchange scenarios	Martino Maggio, Francesco Arigliano
0.91	01/12/2020	Updated Architecture components description, Edge capabilities and deployment scenarios, Data exchange scenarios, Added Alignment with relevant European initiatives	Martino Maggio, Francesco Arigliano
1.0	11/12/2020	Overall integration and revision for review version	Martino Maggio, Francesco Arigliano
1.1	21/12/2020	Changes and correction following review comments	Martino Maggio, Francesco Arigliano
1.2	27/2/2022	Changes after EU reviewer comments	Martino Maggio, Francesco Arigliano, Philipp Rohde
2.0	23/03/2022	Second version of the deliverable (D2.5) with updated content mainly related to IDS components and cloud-edge framework	Martino Maggio, Francesco Arigliano
2.1	26/03/2022	Final version after internal review	Martino Maggio

Table of Contents

Table of Contents 5
List of Figures 7
List of Tables 7
Terms and abbreviations 8
Executive Summary 9
Introduction 10
1 PLATOON requirements 12
1.1 Platform architecture 12
1.2 Platform interoperability 13
1.3 Security Privacy and Sovereignty 13
1.4 Data Analytics toolbox 13
1.5 Edge Computing 14
1.6 Marketplace 14
2 PLATOON architecture high level overview 15
3 Architecture components description 19
3.1 Interoperability layer 20
3.2 Data Management 25
3.3 Intelligence 30
3.4 Marketplace 33
3.5 Security Privacy & Sovereignty 35
3.5.1 Data Sovereignty and IDS Components 43
4 Edge capabilities and deployment scenarios 48
4.1 Deployment scenarios 51
4.1.1 Distributed control plane scenario 51
4.1.2 Centralized control plane scenario 53
4.1.3 Aggregated control plane scenario 54
4.2 Platoon Edge-Cloud Framework 56
5 Data exchange scenarios 59
5.1 Intra organizational flows 59
5.1.1 IoT Real Time Flow 59
5.1.2 Pilot Dataset flow 60
5.1.3 Complex Event Processing flow 60
5.2 Federated Query Flow 61

5.3 Data Analytics Tools flows ... 62
5.4 Security Flows ... 64
5.4.1 End User Authentication and Authorization flow ... 64
5.4.2 Client Authentication and Authorization flow ... 65
5.4.3 Access and Usage Control flow ... 66
5.4.4 Consent flow ... 67
5.5 Inter Organizational Data Exchange Flow ... 68
6 Alignment with relevant European initiatives ... 70
6.1 FIWARE ... 70
6.2 COSMAG ... 71
Conclusion ... 73
References ... 74
Annex I – PLATOON Requirements ... 75

List of Figures

FIGURE 1: PLATOON REFERENCE ARCHITECTURE LOGICAL VIEW 15
FIGURE 2: RELATION BETWEEN PLATOON ARCHITECTURE AND TASKS/WPS 18
FIGURE 3: PLATOON REFERENCE ARCHITECTURE COMPONENT VIEW 19
FIGURE 4: PLATOON SECURITY AND PRIVACY DETAILED COMPONENT VIEW 36
FIGURE 5: IDS CONNECTORS COMMUNICATION 44
FIGURE 6: IDS COMPONENTS INTEGRATION WITH PLATOON SECURITY 44
FIGURE 7: PLATOON ARCHITECTURE EDGE CAPABILITIES 50
FIGURE 8: DISTRIBUTED CONTROL PLANE SCENARIO LAYERS 52
FIGURE 9: DISTRIBUTED CONTROL PLANE SCENARIO COMPONENTS 53
FIGURE 10: CENTRALIZED CONTROL PLANE SCENARIO LAYERS 53
FIGURE 11: CENTRALIZED CONTROL PLANE SCENARIO COMPONENTS 54
FIGURE 12: AGGREGATED CONTROL PLANE SCENARIO LAYERS 55
FIGURE 13: AGGREGATED CONTROL PLANE SCENARIO COMPONENTS 56
FIGURE 14: PLATOON EDGE-CLOUD FRAMEWORK 58
FIGURE 15: IOT REAL TIME FLOW 59
FIGURE 16: PILOT DATASET FLOW 60
FIGURE 17: COMPLEX EVENT PROCESSING FLOW 61
FIGURE 18 - FEDERATED QUERY FLOW 62
FIGURE 19 DATA ANALYTICS TOOL DATA FLOW FOR THE ON-PREMISE IMPLEMENTATION 63
FIGURE 20: DATA ANALYTICS TOOL DATA FLOW FOR THE MICRO-SERVICE IMPLEMENTATION 63
FIGURE 21: END USER AUTHENTICATION AND AUTHORIZATION FLOW 65
FIGURE 22: CLIENT AUTHENTICATION AND AUTHORIZATION FLOW 66
FIGURE 23: ACCESS AND USAGE CONTROL FLOW 67
FIGURE 24: CONSENT FLOW 68
FIGURE 25: INTER ORGANIZATIONAL DATA EXCHANGE FLOW 69
FIGURE 26: COSMAG FEDERATED DATA SOLUTION SPACE (3) 72

List of Tables

TABLE 1 - PLATOON AND FIWARE COMPONENTS MAPPING 71

Terms and abbreviations

API	Application Programming Interface
CB	Context Broker
CEP	Complex Event Processing
CIM	Common Information Model
COSMAG	Comprehensive Architecture for Smart Grid
DAPS	Dynamic Attribute Provisioning Service
EC	European Commission
ETSI	European Telecommunications Standards Institute
FQP	Federated Query Processing
GDPR	General Data Protection Regulation
GE	Generic Enabler
IAM	Identity and Access Management
IDS	Industrial Data Space
IDSCP	IDS Communication Protocol
IEC	International Electrotechnical Commission
IoT	Internet of Things
LLUC	Low Level Use Case
OEMA	Ontology for Energy Management Applications
OTP	One-Time Password
OWL	Ontology Web Language
PEP	Policy Enforcement Point
PDP	Policy Decision Point
PV	Photovoltaic Panel
RBAC	Role Based Access Control
RDF	Resource Description Framework
SAREF	Smart Appliances REFerence ontology
SCADA	Supervisory Control And Data Acquisition
SGAM	Smart Grid Reference Architecture
SSO	Single Sign-On
UC	Use Case
UML	Unified Modelling Language
UKB	Unified Knowledge Base
WP	Work Package

Executive Summary

Deliverable 2.5 describes the second version of PLATOON reference architecture including the logical components in terms of functionalities and interfaces, their relationships and the description of the key processes related, for instance, to data exchange and security. This document is mainly created to support the developers of reference implementation of PLATOON platform and pilots or other external stakeholders that want to be integrated with PLATOON and use its interoperability capabilities.

The chapter 1 of this deliverable resumes the main requirements categories that have been taken in consideration in the design of the logical architecture. These requirements are the ones that were elicited during the activities of work package 1 through the involvement of the different stakeholders of the project. The requirements are classified in different categories and the complete list can be found in the Annex I – PLATOON Requirements.

Chapter 2 introduces the overall architecture describing the different logical layers, the main components and their capabilities in relation to the PLATOON objectives. This section gives to the reader a static view of the architecture showing how the different architectural elements are related.

The PLATOON architectural components will be then described in detail in chapter 3: for each one is provided a table in which are specified the main features of the component, including its detailed description, the functionalities, logical interfaces and the list of specific requirements covered; moreover, in this section are also specified the interactions among the components supported by the usage of UML component diagrams.

Chapter 4 covers the aspects related to the Edge capabilities of the PLATOON architecture: in this section it is, in particular, described, how the different key components of the architecture, based on possible real scenarios and specific computational capabilities, could be deployed (e.g. at edge, central or cloud level).

Chapter 5 presents a dynamic view of the PLATOON reference architecture, in terms of interactions and data flows, depicted through UML sequence diagrams, that can occur between the components described in the previous chapters. A particular attention is paid, in this section, to the security related data flows covering different typical scenarios (e.g. authentication, authorisation, data usage control etc.)

Finally, chapter 6 describes the relations between the PLATOON reference architecture and three main initiatives that have been taken in consideration for the design, but also for the future reference implementation of the technical framework: the International Data Spaces (IDS), the FIWARE framework and the COSMAG architecture.

Introduction

This document describes the reference architecture of PLATOON platform. The main scope of the PLATOON architecture is to identify the main building blocks, capabilities, interfaces and interactions that enable interoperability between heterogeneous technologies across different use cases in the field of Energy domain. The current document represents the updated version of the initial architecture (deliverable 2.1) in which is reflected the progresses and feedbacks coming from the real implementation phase of the pilots. In particular, the first version of the architecture has been updated with the introduction of the IDS components and the improvement of the aspects related to the edge capabilities of the platform.

The process for the definition of the architecture started from the analysis of the requirements elicited in work package 1 related to the pilot use cases, but also to overall objectives of the PLATOON project that driven the main identification of the logical components of architecture. As explained in the following chapter, the requirements that affected the architecture can be considered a sub-set of the entire group identified in WP1, the ones specifically related to the logical architecture capabilities. Despite the fact that this deliverable is an input for the rest of technical developments of PLATOON, the architectural design phase itself was also influenced by the on-going work related to the other tasks of the technical work packages (mainly WP2, WP3 and WP4), that will be finally in charge of the detailed design and concrete implementation of the specific architectural components.

It is important to remark that this document provides technical specifications (i.e. architecture and interfaces) only at logical level, leaving any other consideration related to the concrete implementation of the components (e.g. justification for specific choices about the technologies to use for the implementation, reuse of existing open source components and their improvements to fit PLATOON requirements) to the above mentioned technical work packages (see Figure 2 for further details about the role of the different WPs in the PLATOON architecture implementation).

As mentioned above, the final aim of the PLATOON architectural specification is to be intended in the achievement of logical and technological interoperability rather than the definition of a new Energy platform architecture. The interoperability capability is realised at data collection level through the definition of IoT and Data connectors and at data harmonisation with PLATOON semantic components and related data models.

Beside the interoperability aspects, the PLATOON architecture reflects other specific characteristics to be highlighted such us the security, privacy and sovereignty aspects. The architecture has been designed to provide all the basic capabilities to ensure authentication and authorisation functionalities, together with secure data transfer between different parties inside the same ecosystem. Moreover, specific functionalities have been identified to guarantee personal data privacy and consent management.

Another key element of PLATOON architecture is modularity, not only in terms of the decoupling of its components, but also because it can be instantiated according to the identified context (e.g. Pilots' specificities) by simply using a specific part of the framework.

This document provides a logical view (both static and dynamic) of the PLATOON architecture, maintaining a technological agnostic approach: indeed, the way in the in which the different logical component could be implemented (e.g. the development languages, technologies) and deployed can be different, depending of the specific requirements of the pilots that will be experimented in the project, and in general by the stakeholder that want to adopt the PLATOON framework. The document contains also some references about possible concrete approaches for deployment and baseline frameworks and standards that can be used to concretely implement the PLATOON architecture. This deliverable highlights the compliance of the architectural approach with some existing initiatives: in particular in this second release of the PLATOON architecture it have been introduced some components of the IDS ecosystem (1) that during the implementation phase of the project has been officially adopted as main way to exchange data, between pilots, in a secure way. Moreover the document contains references to the FIWARE framework (2) and COSMAG specification (3), that influenced the definition of PLATOON reference architecture.

1 PLATOON requirements

This section will introduce the WP1 requirements that represent the main input for the logical architectural design. It will be summarized the list of the requirements and how they are covered by the architectural layers and modules to be presented in the following chapters. A specific attention will be paid to the energy-related requirements and challenges describing how the architecture will cover them. Considering the extensive list of requirements produced in the initial elicitation phase, not all of them have been considered as relevant for the architectural design, only a subset of has been taken in consideration: in next sections are described the specific requirement categories that have been analysed in this deliverable, with a brief summary of the requirements that should be covered by the architecture.

Anyway, for reference, at the end of the document is reported the completed list of the requirements including the ones that are not addressed by the reference architecture. Every requirement is identified by a unique number that is referenced by the architectural modules described in the following chapters.

This section presents the requirements of WP1 which are the main input element for the logical architectural design. The list of requirements and how they are covered by the architectural layers and modules will be summarised. The architectural modules will also be presented. Particular attention will be paid to energy-related requirements and challenges, describing how the architecture will cover them. Given the long list of requirements produced in the initial phase of the use case specifications, a prioritisation of these requirements has been considered and some of the requirements have not been considered for the architectural design. In the following sections the specific categories of requirements that have been analysed in this deliverable are described, with a brief summary of the requirements that should be covered by the architecture.

At the end of the document is the complete list of requirements, those that are considered and not considered by the reference architecture. Each requirement is identified by a unique number which is referenced by the architecture modules described in the following chapters.

1.1 Platform architecture

This category covers the general requirements for the entire PLATOON platform reference architecture that defines its scope and characteristics: for instance, PLATOON architecture must avoid vendor lock-in providing a technologically agnostic environment leveraging open source components and public standard. The PLATOON reference architecture will be also compliant with other public initiative as the COSMAG reference architecture (3), FIWARE (2), SGAM (4) and IDS (1). In particular IDS reference architecture ensures data sovereignty, security and privacy when sharing data/tools among platform users. In order to be compatible with IDS, PLATOON reference architecture must be able to integrate the main mandatory components of IDS reference architecture, i.e. the connector and the identity provider (a.k.a DAPS). The PLATOON reference architecture must allow some of the components of the reference architecture to be hosted at the component level (e.g. edge computing), on premise and in the cloud.

Finally, the PLATOON reference architecture must enable the exploitation of digital services (both data and data analytics tools) through a Marketplace.

1.2 Platform interoperability

The energy domain is characterized by the presence of many actors, often large organizations, that provide many technological solutions and proprietary systems. PLATOON project does not want to enter the arena imposing new technologies and approaches asking everyone to adapt to them. Instead, the objective is to maximize flexibility and deploy very high interoperability capabilities. PLATOON will provide common open source data model(s) (5) in order to promote interoperability and reusing, as much as possible, open source and widely-used ontologies and standards for the three main areas of the energy supply chain under the scope of PLATOON. For instance, SAREF (6) ontology for smart buildings and Core IEC Standards for smart grids (7). Nevertheless, the existing ontologies in energy sector don't cover all the needs expressed by the project use case, thus, the ontology(es) to be developed by PLATOON must extend or/and to create ontological modules to represent other knowledge which is not present in the well-known ontologies. Moreover, PLATOON will provide common API (8) and an "Interoperability Layer" to allow the integration of heterogeneous data that will be used in different pilots. In particular it must be able to deal with the common data formats as per defined in the pilot requirements in deliverable D1.1 (9): Files (csv, xml, JSON, CAD and jpeg), Logs (csv and txt), SQL and NoSQL (mat and tdms format).

1.3 Security Privacy and Sovereignty

Security, privacy and sovereignty are three very key aspects that every architecture has to take into consideration with particular attention. In PLATOON these aspects must be considered at two levels.

The first one concerns the normal infrastructure and security services within each organization and it is necessary to achieve adequate quality standards by implementing all recognized good practices. This aspect is more related to the specific pilot internal security requirements, so PLATOON architecture will define some key enablers for security, but the concrete security framework should be addressed by the pilot itself.

The second aspect, very related with the interoperability objective of PLATOON, will focus on exchanges of information between different organizations and this will mainly be covered by aligning the architecture with the IDS initiative.

1.4 Data Analytics toolbox

Data Analytics toolbox will be the final end-users service to be provided by PLATOON. PLATOON will develop a data analytics toolbox containing both generic big data tools and energy specific analytical tools for different applications. The toolbox has to provide the possibility to process data in batch and real time manner.

PLATOON Data Analytics Toolbox should allow distributed implementation of the different parts of the data analytics tools at different levels of the architecture (i.e. at the edge and on cloud/premise) so that the developed tools make an efficient use of the available storage and processing capability. The PLATOON Data Analytics Toolbox should be compatible with the main widespread open-source and free to use big data batch and real time processing frameworks (e.g. Hadoop V2.0, Spark, Kafka, etc.) and storage solutions (RDBMS, HDFS, HBase, NoSQL databases. Cache, time series) so that it will be possible to use already existing and validated libraries (e.g. Spark MLib, Spark ML, etc.) without having to write all the code from scratch. Also, by using these open source solutions the project will benefit from a global software development community.

1.5 Edge Computing

In order to make the PLATOON architecture flexible and adaptable to very different real contexts, it has to be possible to distribute some of its components at the edge; the developed Edge Computing solutions should provide support to main industry protocols (e.g. Modbus, BACnet, etc.) and allow to execute locally microservices and applications in order to enable distributed and autonomous data processing. Depending on the use case, applications might include data cleaning, data compression, feature extraction, event detection with threshold-based notifications and alarms, analytics algorithms, remote monitoring and diagnostics, and any other custom software. Also, at the edge all the security and privacy aspects must be addressed. PLATOON platform Edge Computing solutions should have a fault-tolerant design both from a software and hardware perspective so that they will be used for critical applications where no failures are permitted.

1.6 Marketplace

PLATOON must support the proliferation of business communities in the energy sector and in particular of value-added services on top of data coming from the field. Fundamental will be the implementation of a marketplace where tools, data and services can be described and exchanged safely. In order to maximize the added value of the marketplace, the technology with which the applications will be distributed must be based on containerization and the description based on metadata must be harmonized.

One of the objectives of the PLATOON Marketplace is to centralize all the different providers in one single one-stop shop including a complete and easy way to find catalogue of all of the products and services offered by the different providers choosing the most suitable assets to fit specific requirements.

The PLATOON Marketplace must be compatible with the IDS reference architecture so that it will possible to access an ecosystem of trusted companies that meet certain minimum requirements regarding security, privacy and sovereignty.

PLATOON Marketplace has to be compliant with GDPR (10) to avoid data privacy issues with any personal data that might be stored, provided or shared through the PLATOON Marketplace.

2 PLATOON architecture high level overview

This chapter will present, at high level, the PLATOON reference architecture, that will be then detailed in chapter 3 through the description of the components. The layers and the components of the architecture have been directly identified from the requirements resumed in chapter 1 and listed in detail in Annex I.
The following picture depicts the overall architecture showing, for the sake of simplicity, only the main logical components and interactions. The architecture can be divided in logical layers described in the next sections.

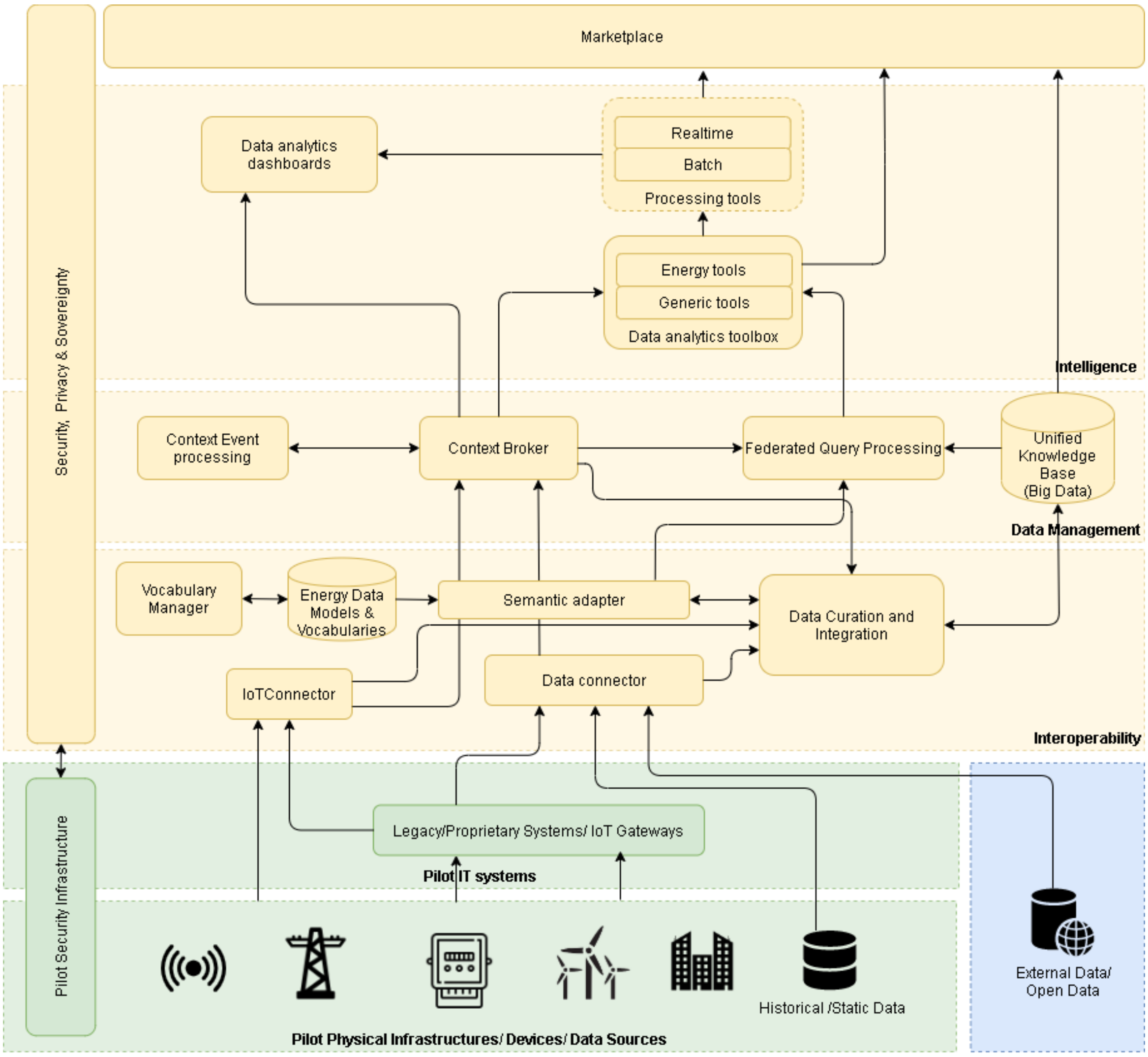
Marketplace
Security, Privacy & Sovereignty
Data analytics dashboards
Realtime
Batch
Processing tools
Energy tools
Generic tools
Data analytics toolbox
Intelligence
Context Event processing
Context Broker
Federated Query Processing
Unified Knowledge Base (Big Data)
Data Management
Vocabulary Manager
Energy Data Models & Vocabularies
Semantic adapter
Data Curation and Integration
IoTConnector
Data connector
Interoperability
Pilot Security Infrastructure
Legacy/Proprietary Systems/ IoT Gateways
Pilot IT systems
Historical /Static Data
Pilot Physical Infrastructures/ Devices/ Data Sources
External Data/ Open Data

Figure 1: PLATOON reference architecture logical view

Physical infrastructure and data sources

This layer includes all the data sources that are provided in the physical sites of each pilot or in his organizational context. For example, a renewable energy production plant, a building or a complex of buildings, or single devices such as energy meters. In this layer it is also included data that an organization may have collected in the past and that is available for running his business. They can be both real time data coming from sensor or historical series of various types such as periodic measurements from sensor or performance indicators made over the years or static data that describes the objects features relevant for the organization (e.g. description of the characteristics of a building or plant, configuration parameters of a device, etc).

External Data Sources /Open Data

It represents all the data sources external to the organizational context (i.e. out of the PLATOON ecosystem) useful for integrating the knowledge base. For example, they can be various types of data such as weather historical series or weather forecasts or, in general, public domain Open Data.

Pilot IT Systems

They are all the possible the proprietary/legacy IT systems that manage the operational and historical databases within the organization. They can manage different type of information such as data collected form IoT devices and any type of Energy related infrastructure: these systems can include IoT gateways that are in charge of translating and adapting IoT (proprietary) protocols or other typology of platforms (e.g. SCADA compliant) providing data to dedicate legacy protocols

In the most of cases, for technical reasons or due to company policies, these systems represent the only possible interface for communication among devices and higher-level components in the PLATOON architecture.

Interoperability layer

The interoperability layer is responsible for transforming the data that is collected by data sources into structures that can be managed by systems to be exploited. In particular the capabilities of this layer can be summarised in the following processes:

- Data collection: the interoperability layer must have the ability to capture and manage heterogeneous type of data through IoT Connectors to connect with physical devices such as sensors and embedded systems, and Data connectors in charge of collecting data from legacy/proprietary systems.
- Semantic Adaptation/Mapping: this will process will include the adoption common semantic models and the concrete adaptation is made through a component that implements the semantic modelization using the Energy Data models defined in T2.3 (5).
- Data Curation Integration: this defines the logical rules that allow to validate the quality of the data, filtering those that are not optimal for processing and data ingestion and harmonization in a common language/format.

Data Management

This layer is in charge of managing data (historical and real-time) providing it through standard API to the upper layers. The scope of the components of this layer is to provide a unified knowledge base in which the data collected and harmonised in the interoperability layer can be accessed through (semantic) federated queries. This layer will provide the specific big data technologies needed to manage the large amount of data produced by pilots. Moreover, through the mean of a Context Broker the data management layer will manage real-time and context data using a publish-subscribe approach.

Intelligence

intelligence layer represents a key part of the PLATOON architecture: this layer is the one designated for processing information from the lower levels in order to provide value-added services. It includes all types of big data analysis and artificial intelligence, both real time and batch processing. The Data Analytics toolbox will be formed of all the data analytics tools that will be developed in the project by the different partners for the different use cases defined in the deliverable. These tools will allow the extraction of value from heterogeneous data sources.

Marketplace

This component will be in charge of publishing and enabling search for different type of assets (including datasets, service and applications (e.g. data analytic tools) providing also functionalities to describe them through metadata that includes the properties of the assets and the way to access them. The marketplace will be the way in which pilots can share, with the rest of the ecosystem, data and applications that will be accessible through standard metadata description and API as per defined in deliverable D2.2 (8). The marketplace, depending on the specific case, can also enable additional functionalities related, for instance to the asset monetisation and transaction monitoring.

Security, Privacy and Sovereignty

This is a transversal layer covering all the aspects related to security, privacy and sovereignty. Specifically, these include authentication and authorisation capabilities, functionalities to ensure confidentiality and integrity of the communications, data usage control and personal data management. This layer is also logically connected with the specific security frameworks of pilots’ infrastructure providing functionalities to the rest of the architectural components that have to run in a secure a reliable environment.

The PLATOON reference architecture, as mentioned previously, is the logical baseline for the following concrete implementation of the technological framework. The following picture shows the different tasks and work packages that address, in terms of detailed design and implementation the architectural components of PLATOON. Detailed information about the inner modules of these components, public API and reference implementation technologies, can be found in the respective deliverables.

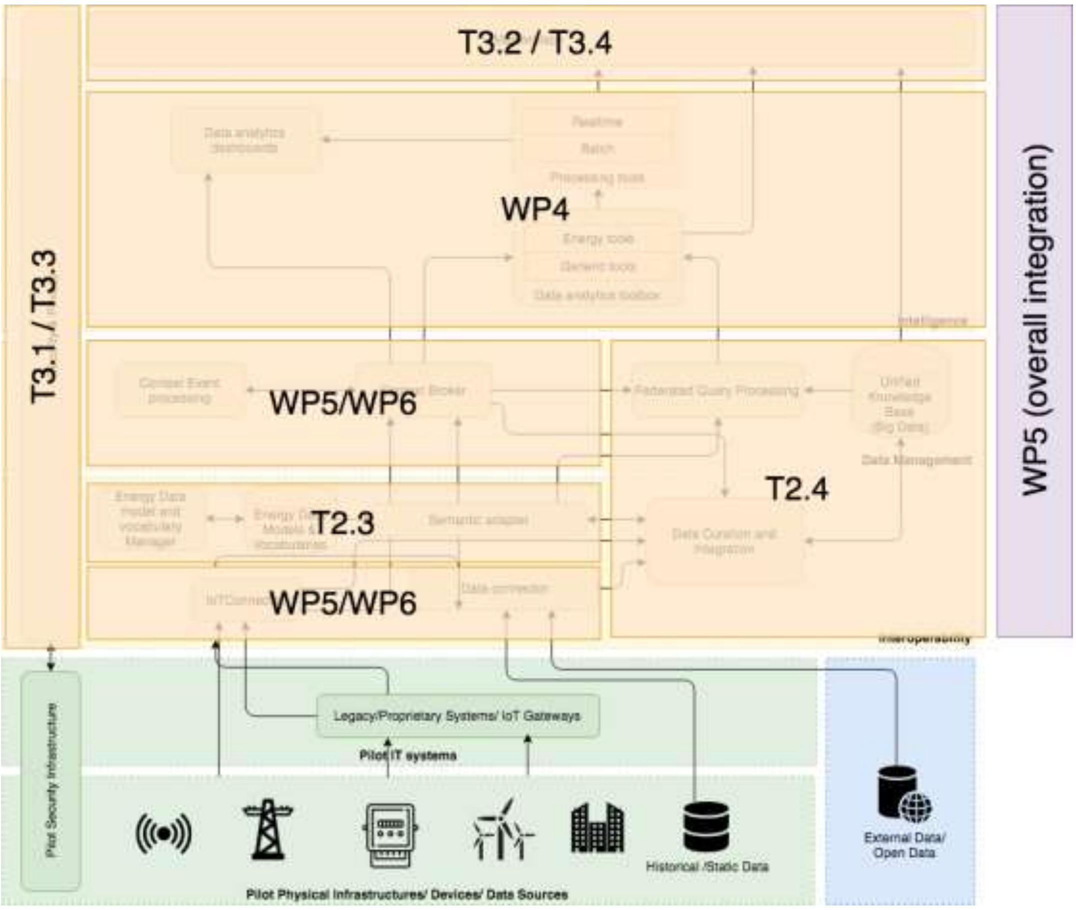
T3.2 / T3.4
T3.1 / T3.3
WP5 (overall integration)
Data analytics dashboards
Realtime
Batch
Processing tools
WP4
Energy tools
Generic tools
Data analytics toolbox
Intelligence
Context Event processing
WP5/WP6
Context Broker
Federated Query Processing
Unified Knowledge Base (Big Data)
Data Management
T2.4
Energy Data model and vocabulary Manager
Energy Data Models & vocabularies
T2.3
Semantic adapter
Data Curation and Integration
IoTConnector
WP5/WP6
Data connector
Interoperability
Pilot Security Infrastructure
Legacy/Proprietary Systems/ IoT Gateways
Pilot IT systems
Historical /Static Data
External Data/ Open Data
Pilot Physical Infrastructures/ Devices/ Data Sources

Figure 2: Relation between PLATOON architecture and tasks/WPs

3 Architecture components description

The following sections will detail the different layers and component of the PLATOON reference architecture. For each component it is included a table describing the overall scope, its functionalities, logical interfaces and relationships with the other components. Moreover, are also listed the identification numbers of the requirements covered by that component that represent the rationale of its presence in the overall PLATOON architecture. Complete description of the requirement can be found in the annex of this deliverable.

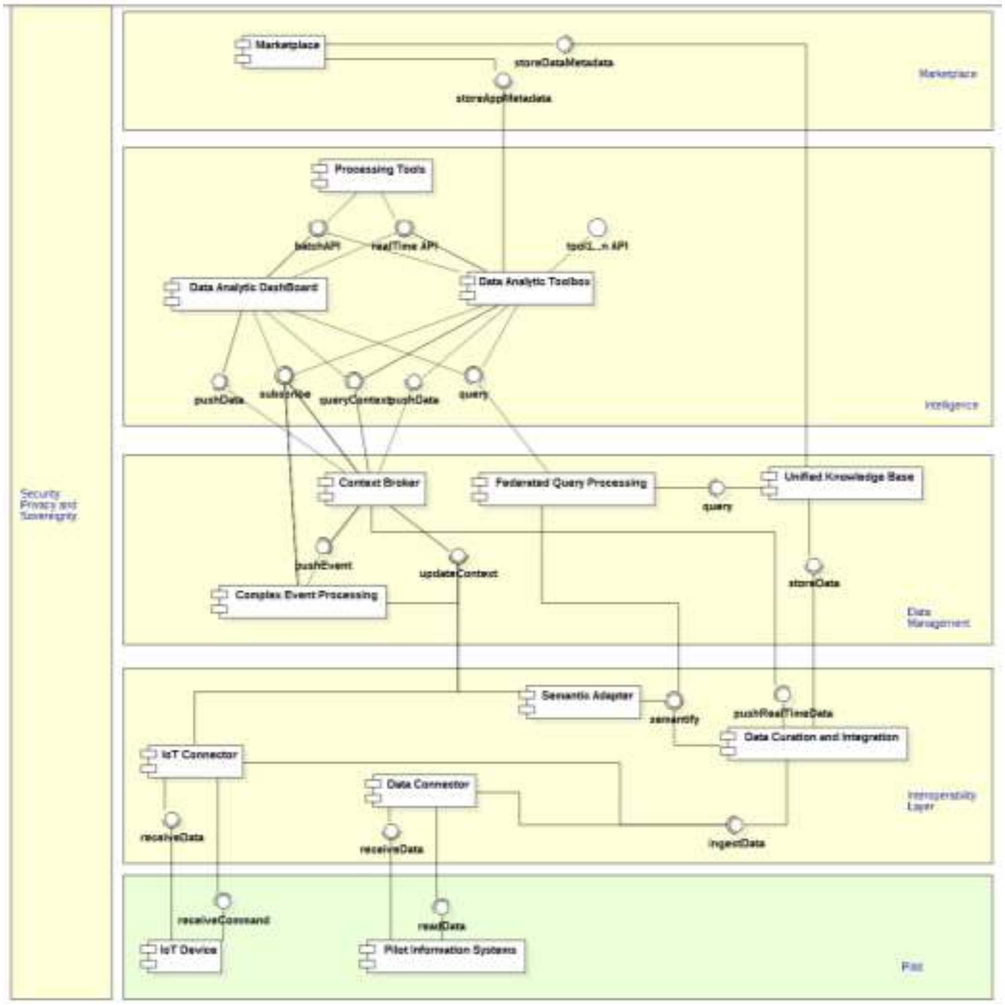

Figure 3: PLATOON reference architecture component view

3.1 Interoperability layer

<table>
<tr><td><b>Component Name</b></td><td><b>Data Connector</b></td></tr>
<tr><td><b>Module Description</b></td><td>These types of components are necessary to allow the integration of Legacy/Proprietary Systems by providing endpoints that support their formats and protocols. They are components, often ad hoc developed, which have the responsibility to correctly query the legacy systems and / or to expose interfaces suitable for receiving information from them (pull/push). Data Connectors should also be able to deal with security aspects (authentication and authorization of the channel). and data sovereignty (IDS) aspects interacting with the <i>Security, Privacy & Sovereignty</i> module of PLATOON’s reference architecture. This component should be able to access to legacy/proprietary data using several approaches (e.g. read from API, read csv or json file, read from SQL or NoSQL databases, etc.), furthermore this component must interact with the <i>Semantic Adapter</i> to convert non-semantic data into semantic data, the <i>Data Curation and Integration</i> module to harmonize and/or integrate the data with additional data and/or metadata or with the <i>Context Broker</i> if no further data processing is needed.</td></tr>
<tr><td><b>Main functionalities</b></td><td>The Data Connector provides the following functionalities:<ul><li>Read data from the legacy/proprietary systems</li><li>Expose an interface to allow legacy/proprietary systems to send data to the Data Connector</li></ul></td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table><tr><th>Interface name</th><th>Description</th></tr><tr><td>receiveData</td><td>to receive data from Legacy/Proprietary systems</td></tr></table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component
13001, 13014, 13017, 13018, 13019,</td></tr>
<tr><td><b>Interaction with other components</b></td><td><table><tr><th>Interfacing Comp.</th><th>Interface Description</th></tr><tr><td>Legacy/Proprietary Systems</td><td>These are the data sources used by the data connector to read data.</td></tr><tr><td>Security Privacy & Sovereignty</td><td>The data connector interacts with this module to provide security and sovereignty over the data.</td></tr><tr><td>Semantic Adapter</td><td>This component is used to convert the raw data into semantic data using common data models.</td></tr><tr><td>Context Broker</td><td>This component is used by the Data Connector to update data, if data already maps with the common data models</td></tr><tr><td>Data Curation and Integration</td><td>The raw data coming from the Legacy/Proprietary systems that need to be</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td></td><td>processed are provided to the Data Curation and Integration module</td></tr></table></td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>IoT Connector</b></td></tr>
<tr><td><b>Module Description</b></td><td>The role of the IoT Connector is to mediate between the raw data coming from the device and the representation of the virtual entity to the Data Management Layer: it is a software module that enables devices, sensors and actuators to send their data to and be managed using their own native protocols. IoT Connectors should also be able to deal with security aspects (authentication and authorization of the channel) and data sovereignty aspects. In addition, they should provide other common services to the device programmer.
This is the Component should cover the whole set of IoT protocols and standard communication interfaces exposed by devices of Pilot Data Sources layer.</td></tr>
<tr><td><b>Main functionalities</b></td><td>The IoT connector provides the following functionalities:<ul><li>Retrieve raw data from the underling IoT sensors, devices or actuators</li><li>Support most of the IoT protocols and standard interfaces to read data from sensors or devices</li><li>Provide functionalities to perform action on the IoT sensors, devices or actuators</li></ul></td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table><tr><th>Interface name</th><th>Description</th></tr><tr><td>receiveData</td><td>to receive data from IoT Devices</td></tr></table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component
13001, 13014, 13018, 13019, 13038</td></tr>
<tr><td><b>Interaction with other components</b></td><td>The following table reports the component interacting with the IoT Connector.<table><tr><th>Component</th><th>Interface Description</th></tr><tr><td>IoT devices, gateways</td><td>These data sources will send data to the IoT Connector.</td></tr><tr><td>Security, Privacy & Sovereignty</td><td>The IoT connector interacts with this module to provide security and sovereignty over the data.</td></tr><tr><td>Semantic Adapter</td><td>This component is used to convert the raw data into semantic data using common data models.</td></tr><tr><td>Context Broker</td><td>This component is used by the IoT Connector to update data, if data already maps with the common data models</td></tr></table></td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Data Curation and Integration</b></td></tr>
<tr><td><b>Module Description</b></td><td>This component is in charge of integrating and harmonizing and pre-processing data from legacy systems and / or external data sources, in order to be ingested by the Unified Knowledge Base.
Heterogeneous data from different sources will be pre-processed by standardizing, cleaning, and normalizing values and transformed into a common data model represented using PLATOON data model/ontology. Such transformed data will then be integrated to the Unified Knowledge Base/Graph.</td></tr>
<tr><td><b>Main functionalities</b></td><td>The Data Curation and Integration component provides the following functionalities:
• Data ingestion from heterogeneous data sources including streaming data
• Metadata management: provenance, mapping rules, and constraints
• Pre-processing data by standardizing, normalization and aggregation of data values
• Semantification, transformation of raw data represented in different data model to semantic data representation model (RDF data model)
• Linking and enrichment of data to external as well as to unified knowledge base
• Validation of data quality based on user defined constraints
• Integration of heterogeneous data into a common data model and knowledge base</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>ingestData</td><td>Receive data and data set from information systems and trigger specific ingestion process</td></tr>
<tr><td>pushRealTimeData</td><td>This is the call back interface exposed in order to permit the CB to send new punctual data update coming from IoT devices</td></tr>
</table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component
13014, 13017, 13018, 13019, 13022</td></tr>
<tr><td><b>Interaction with other components</b></td><td><table>
<tr><th>Component</th><th>Interface description</th></tr>
<tr><td>Semantic Adapter</td><td>Provides semantic data transformed from non-semantic (raw) data from IoT devices as well as from legacy systems.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td>Data Connector</td><td>Connection point to legacy and external data sources where data will be ingested to the Unified Knowledge Base as well as data that is used for pre-processing and enrichment.</td></tr>
<tr><td>Unified Knowledge Base</td><td>Integrated data is stored and managed by this component.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Semantic Adapter</b></td></tr>
<tr><td><b>Module Description</b></td><td>The role of the Semantic Adapter is to provide the facilities to share data models and harmonize data representations from different formats. It performs data transformation to semantic data from non-semantic data. In PLATOON architecture context, semantic adapter would be connected to data connectors (ingestion point of data into PLATOON from various sources) to receive non semantic data, would do the data transformation to semantic data and pass on transformed semantic data to upstream components. Semantic adapter has the capability to work with both real time and batch data and that speed is fundamentally governed by the type of connector it connects to.</td></tr>
<tr><td><b>Main functionalities</b></td><td>List of main functionalities provided by the component
<ul>
<li>Data ingestion with provenance for each data connector</li>
<li>Data transformation rules definition for each source</li>
<li>Data transformation from non-semantic to semantic data</li>
<li>Capable of handling both batch and real time data</li>
<li>Semantic data does not store any data (other than provenance)</li>
</ul></td></tr>
<tr><td><b>Main logical Interfaces</b></td><td>
<table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>semantify</td><td>This is the interfaces where other components ask the semantic adapter to transform raw or proprietary data in semantic data according to shared or standard models.</td></tr>
<tr><td>subQuery</td><td>query interface using SPARQL</td></tr>
</table>
</td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component: 13014, 13018, 13019</td></tr>
<tr><td><b>Interaction with other components</b></td><td>
<table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>Data Connectors</td><td>Feed non-semantic data to adapter as they are directly connected to the data source</td></tr>
</table>
</td></tr>
</table>

Context Broker	With each data connector individually identified as data publisher for context broker, data after transformation will be sent to Context broker
Federated Query Processing	Retrieve mapping for schemas and vocabularies. Semantic data transferred to federated query component for possible push scenarios to connected upstream services
Data Curation and Integration	Transformed data is sent for possible connection within KB by this component
Data Transformation Rules	Data mapping rules govern how to transform non-semantic data to semantic data with respect to each data source
IoT Connectors	Data received from IoT devices of pilot systems is transformed to semantic data in real-time and sent to context broker and data curation and integration module for storage

Component Name	**Vocabulary manager**
Module Description	Shared semantics is the key to maximizing the information power of data; the Vocabulary Manager's role is to manage the shared data models that will be used to unify heterogeneous data. A set of standard data models available as part of PLATOON, containing terms to use in the representation of data. The adoption of shared data models has relevance for both Data Providers and Application Providers giving the possibility to reuse the same applications in different cases, with few technical activities, also enabling APP Providers to be part of a common market
Main functionalities	Providing shared semantics and data representation. The PLATOON common data models make use of several standard ontologies such as SAREF, SEAS and OntoWIND. Also, thinking on future project exploitation, in the future, the PLATOON data model will likely need to be extended/modified and new data models might be incorporated. The vocabulary manager is the component responsible for managing the different data models/vocabularies available in the PLATOON ecosystem. In addition, the vocabulary allows to visualize and query the different ontologies through a GUI. It also supports machine-to-machine communication allowing to query the different ontologies directly through an IDS connector.
Main logical Interfaces	

Interface name	**Description**

<table>
<tr><td></td><td><table><tr><td>IDS connector – Query message</td><td>Machine-to-machine communication allowing to query the different ontologies directly through an IDS connector according to the IDS information model.</td></tr></table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component
13014, 13015, 13016</td></tr>
<tr><td><b>Interaction with other components</b></td><td>Used by all the north bound PLATOON architecture components</td></tr>
</table>

3.2 Data Management

Component Name	**Context Data Broker**
Module Description	The Context Data Broker is a central pivot in near real time information flows, it enables discovering, gathering and publishing of near real time context information through Context Management APIs. Context Broker, through its interface, makes available the context information regardless data source and using different type of interaction: query and subscription, represents the synchronous and asynchronous interactions with context data source. Synchronous interactions are performed using a query mechanism to obtain context information; the component allows building queries, using different types of filters, in order to retrieve information with high level of precision. The asynchronous interaction is performed by publish-subscribe mechanism: a notification is generated when published data meets the subscription conditions; this feature is really useful to avoid the implementation of a polling process on data sources of interest, allowing to be notified when the context information changes.
Main functionalities	• Context Availability, represents the operations to identify which context data sources are managed by PLATOON platform. • Entity Management: represents the synchronous interactions with context data source. It includes a query mechanism to obtain context information and update operation to push information; it allows building powerful queries, using different types of filters, in order to retrieve information with high level of precision.

<table>
<tr><td></td><td>

- Subscription management, performs the asynchronous interaction by publish-subscribe mechanism: a notification is generated when published data meets the subscription conditions; this feature is really useful to avoid the implementation of a polling process on data sources of interest, allowing to be notified when the context information changes.

</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td>
<table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>updateContext</td><td>Create or modify virtual entities or attribute values</td></tr>
<tr><td>queryContext</td><td>Retrieve context information</td></tr>
<tr><td>subscribe</td><td>Subscribe entity or attributes in order to be notified when value change.</td></tr>
</table>
</td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component
13005, 13014</td></tr>
<tr><td><b>Interaction with other components</b></td><td>
<table>
<tr><th>Component</th><th>Interface description</th></tr>
<tr><td>Semantic Adaptor</td><td>exchanging the context information to obtain a translation in harmonized data formats.</td></tr>
<tr><td>Complex Event Processing</td><td>exchanging the context information to detect specific events and managing the corresponding actions.</td></tr>
<tr><td>Unified Knowledge Base</td><td>providing context information.</td></tr>
<tr><td>Federated Query Processing</td><td>providing context information.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Federated Query Processing</b></td></tr>
<tr><td><b>Module Description</b></td><td>The federated query processing component provides a unified access interface to a federation of heterogeneous data sources integrated in the unified knowledge base. The unified knowledge base might be fragmented horizontally or vertically. Each fragment might be stored in a different data management system (e.g., Virtuoso or MongoDB). Hence, the federated query processing component is dealing with the problem of heterogeneity. The role of federated query processing is to transform input queries into sub-queries against the fragments of the unified knowledge base. Further, a query plan is generated in order to minimize the number of contacted fragments and to speed up the processing of each individual sub-query, as well as the gathering of the results into the query answer.</td></tr>
<tr><td><b>Main functionalities</b></td><td>Main functionalities provided by the Federated Query Processing component includes:<ul><li>Providing a unified query interface using the W3C recommendation language to query RDF data (SPARQL)</li><li>Combining data from the Unified Knowledge Base with data in motion via the Semantic Adapter</li><li>Aggregating data from heterogeneous distributed data sources</li><li>Exploration of Knowledge Base based on different patterns</li><li>Serving query results in different formats, e.g., JSON</li></ul></td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table><tr><th>Interface name</th><th>Description</th></tr><tr><td>query</td><td>unified query interface using SPARQL</td></tr></table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component: 13014, 13022</td></tr>
<tr><td><b>Interaction with other components</b></td><td><table><tr><th>Component</th><th>Interface description</th></tr><tr><td>Semantic Adapter</td><td>Provides synchronous semantic data transformation from non-semantic (raw) data generated by IoT devices as well as from legacy systems.</td></tr><tr><td>Context Data Broker</td><td>Provides context to events in data process.</td></tr><tr><td>Unified Knowledge Base</td><td>Access to the unified knowledge base is provided by this component. The Knowledge base is stored in a federated data store.</td></tr></table></td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Unified Knowledge Base</b></td></tr>
<tr><td><b>Module Description</b></td><td>This component aims to solve the issues related to the big data storing. Data coming from various heterogeneous sources differ in terms of types, characteristics and constraints imposed by their owners after being harmonized are collected by this component.
The unified knowledge base represents data that is pre-processed and transformed according to the PLATOON semantic data models and stored in a distributed way.</td></tr>
<tr><td><b>Main functionalities</b></td><td>Main functionalities of Unified Knowledge Base include:<ul><li>Provide data storage solving different data complexities such as volume, velocity and variety</li><li>Manage data partitioning and distribution to provide efficient access to data</li></ul></td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table><tr><th>Interface name</th><th>Description</th></tr><tr><td>store</td><td>Interface to store data in the UKB</td></tr><tr><td>query</td><td>Interface to retrieve data from the UKB</td></tr></table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component
13009, 13022</td></tr>
<tr><td><b>Interaction with other components</b></td><td><table><tr><th>Component</th><th>Interface description</th></tr><tr><td>Federated Query Processing</td><td>Access data stored in a federated fashion. The FQP component joins data from different fragments of the unified knowledge base.</td></tr><tr><td>Marketplace</td><td>Access different fragment/partitions of the unified knowledge base.</td></tr></table></td></tr>
</table>

<table>
<tr><th>Component Name</th><th>Complex Event Processing</th></tr>
<tr><td>Module Description</td><td>The Complex Event Processing module processes a huge number of events and get valuable information from them. In a large-scale system, devices can generate several simple events; these events contain semantic information that could be quite limited to detect complex situations, so an additional processing logic is required.
It is in charge of analysing context information in real-time in order to recognize complex event patterns and to enable adequate responses. It works by reacting not only to single events but also responding to a combination of events, in sequence or in parallel, triggering meaningful actions for applications or context update events.</td></tr>
<tr><td>Main functionalities</td><td>Pattern Management, defines the rules to detect the types of possible patterns supported. Examples of pattern typologies are single event, sequence of events for a given time window, using aggregation function on one or more events, etc.
Event Processing Management, defines the rules to analyse a set of events under specific conditions. Example of these conditions could be detection of events in a specific time interval, recognition of events that satisfy the same criteria about their attributes, etc.
Actions Management, defines the rules to establish the actions responding to events.</td></tr>
<tr><td>Main logical Interfaces</td><td><table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>ruleManagement</td><td>Create, modify, delete rules defining event match temporal patterns and rules</td></tr>
<tr><td>eventListener</td><td>Receive real time event from context broker</td></tr>
</table></td></tr>
<tr><td>Requirements Mapping</td><td>List of requirements covered by this component
13005, 13014</td></tr>
<tr><td>Interaction with other components</td><td><table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>Context Data Broker</td><td>Receiving and analysing context information in order to find patterns and execute the actions to update the context information</td></tr>
</table></td></tr>
</table>

3.3 Intelligence

<table>
<tr><td><b>Component Name</b></td><td><b>Processing tools</b></td></tr>
<tr><td><b>Module Description</b></td><td>The role of the Processing tools is to provide the computational capabilities along with the algorithms necessary to extract information and knowledge from data gathered from heterogeneous sources.

The Processing tools module will be formed of the different analytic tools that will be used to process data. For instance, Python, SPARK, Kafka, etc. These tools will allow the extraction of value from heterogeneous data sources. There will be two main groups of data analytics tools and each of them will use different processing tools:

Real Time Processing Tools
There are some use cases such as Pilot3a where due to the frequency of the data and the type of response that is required where it is very likely that will need real time or near real time processing. This component represents all application modules that process context data from devices in real or near real time manner. Such applications must be able to combine historical data with real-time data in order to provide both analytical and operational services. Examples of processing tools for real time processing are Kafka or SPARK-Streaming.

<i>Batch Processing Tools</i>
There are some pilots such as pilot 3b and 3c that due to the nature of the use cases it is not required real-time processing and batch (or micro-batch) processing is enough. This component represents all application modules that process context data from devices in batches. Examples of processing tools for batch processing are Python or SPARK.</td></tr>
<tr><td><b>Main functionalities</b></td><td>Real time /batch data processing</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table><tr><th>Interface name</th><th>Description</th></tr><tr><td>realTimeAPI</td><td>Interface to invoke the tools in real time</td></tr><tr><td>batchAPI</td><td>Interface to invoke the tools providing large amount of data</td></tr></table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component

13022, 13023, 13024, 13025, 13026, 13030</td></tr>
</table>

Interaction with other components	**Interfacing Component**	**Interface Description**
	Data Analytics Toolbox	Invoke processing tools for elaborations
	Data Analytics Dashboard	Retrieve data to report

Component Name	**Data Analytics Toolbox**
Module Description	The Data Analytics toolbox will be formed of all the data analytics tools that will be developed in the project by the different partners for the different use cases defined in the deliverable. These tools will allow the extraction of value from heterogeneous data sources. There will be two main groups of data analytics tools: 1. *Energy specific tools*: Tools which have been specifically developed for the specific applications or services for the specific domains of the energy value chain (e.g. digital twin for a Wind Turbine electric generator, Electricity Balance Optimiser, Predictive Maintenance Tool for Electric Transformers, HVAC Optimiser…) 2. *Generic tools*: data analytics tools that complement the energy specific tools and that are applicable for different applications and domains (e.g. data pre-processing tools, visualisation tools, graph processing tools, etc.). The Data Analytics Toolbox plays a central role in the PLATOON federated platform and it is also strongly related to the interoperability layer. In fact, the interoperability layer is the key element that will enable the use (and reuse) of the tools from the PLATOON Data Analytics Toolbox by the partners of the project. Equally, a common API specification has been defined in deliverable D2.2 that should be used by all the PLATOON data analytics tools. Data analytics toolbox should be interoperable and able to be deployed in any environment. Depending on the specific application field each of the tools will be developed using different programming languages and libraries. In order to meet these requirements, a container-based design is proposed. A container is a standardized executable software unit that includes the source code and all its dependencies and the correspondent packages. The containerized software is isolated from its environment. Hence, a container is easy to ship and deploy into

<table>
<tr><td></td><td>different environments, even if these are different from that used during development.</td></tr>
<tr><td><b>Main functionalities</b></td><td>Data valorisation (value extraction from data).</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td>
<table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>toolsAPI</td><td>Tool specific API</td></tr>
<tr><td>pushData</td><td>Interface to receive data from other components</td></tr>
</table>
</td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component
From 13021 to 13035</td></tr>
<tr><td><b>Interaction with other components</b></td><td>
<table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>Processing tools</td><td>Invoke real time/batch elaboration. In fact, depending on the specific application, in some cases processing tools and data analytics tools might be embedded in the same container (layered container) as per explained in deliverable D4.1. In some other cases they will be embedded in different containers and orchestrated via Docker Compose o Kubernetes.</td></tr>
<tr><td>Context Broker</td><td>Get context data to be analysed</td></tr>
<tr><td>Federated Query Processing</td><td>Access data stored in a federated fashion. Federated Query Processing component joins data from different partitions of the unified knowledge base.</td></tr>
<tr><td>Unified Knowledge Base</td><td>Access to unified knowledge base is provided by this component. The Knowledge base is stored in a federated data store. Furthermore, for the cases where all the data is stored in a single central database, the data analytics tools will be directly linked to that specific database instead of to the whole unified knowledge base.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Data Analytics Dashboard</b></td></tr>
<tr><td><b>Module Description</b></td><td>This module is in charge of use data analytics outcomes allowing the creation of new information from raw and processed data, obtaining immediate visual impact and useful insights. The analytics dashboard will help end users understand how processes can be optimized or better decisions can be made. The Dashboard offers plugins and sub-components for several kind of data visualizations. Besides, the component offers graphical tools intended to perform advanced data analysis such as those related to location or business intelligence.</td></tr>
<tr><td><b>Main functionalities</b></td><td><ul><li>Data visualisation</li></ul></td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table><tr><th>Interface name</th><th>Description</th></tr><tr><td>pushData</td><td>Interface to receive data from other components</td></tr></table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>List of requirements covered by this component
13028, 13029, 13030, 13031, 13032, 13033</td></tr>
<tr><td><b>Interaction with other components</b></td><td><table><tr><th>Interfacing Component</th><th>Interface Description</th></tr><tr><td>Processing tools</td><td>Display the result of the processing performed by the tool</td></tr><tr><td>Context Broker</td><td>Display the information relating to one or more entities contained in the CB and their variations</td></tr><tr><td>Federated Query Processing</td><td>Access data stored in a federated fashion. Federated Query Processing component joins data from different partitions of the unified knowledge base.</td></tr></table></td></tr>
</table>

3.4 Marketplace

This component provides functionalities to publish, search and browse for different assets: data and analytics services. Asset offerings can be organised into groups/categories in a hierarchical fashion to allow for an easy navigation and discovery of them. Metadata define characteristics and properties of assets (metadata could include also way to exchange / access datasets) They may also be inherited from a higher level in a category hierarchy. The module lets the asset providers be able to define the technical description of the assets they own.

<table>
<tr><th>Component Name</th><th>Marketplace</th></tr>
<tr><td>Module Description</td><td>This component provides functionalities to publish, search and browse for different assets: IoT data and analytics services. Asset offerings can be organised into groups/categories in a hierarchical fashion to allow for an easy navigation and discovery of them. Metadata define characteristics and properties of assets (metadata could include also way to exchange / access datasets) They may also be inherited from a higher level in a category hierarchy. The module lets the asset providers be able to define the technical description of the assets they own. The initial users and testers of this module will consist of the pilot participants that are able to publish metadata of their data and data analytics tools. Also, open source data and open source generic data analytics tools will be made available and accessible to download.
A common API specification has been defined in deliverable D2.2 specifically for the market place that should be used by the rest of the components of the architecture to interact with the marketplace.</td></tr>
<tr><td>Main functionalities</td><td>Marketplace encapsulates the functionality of the IDS Broker, IDS App Store and Clearing house:<ul><li>Metadata-registry for the Data and apps (neither the data or the apps are stored)</li><li>Facilitates the contract definition/negotiation between data consumers and data producers</li><li>Facilitate and keep tracks of the exchange of payments (Log registry)</li></ul></td></tr>
<tr><td>Main logical Interfaces</td><td><table><tr><th>Interface name</th><th>Description</th></tr><tr><td>storeAppMetadata</td><td>Set metadata for an app in the store</td></tr><tr><td>storeDataMetadata</td><td>Set metadata for a data-source in the store</td></tr></table></td></tr>
<tr><td>Requirements Mapping</td><td>13012, from 13047 up to 13065</td></tr>
<tr><td>Interaction with other components</td><td><table><tr><th>Interfacing components</th><th>Interface Descriptions</th></tr><tr><td>Data Analytics Tools</td><td>Data analytics tools will be the integrated apps in app store</td></tr><tr><td>Unified Knowledge Base</td><td>Access to unified knowledge base is provided by this component. The Knowledge base is stored in a federated datastore. Furthermore, for the cases where all the data is stored in a single central database, the marketplace will be directly linked to that specific database instead of to the whole unified knowledge base.</td></tr></table></td></tr>
</table>

3.5 Security Privacy & Sovereignty

This module provides flexible security and privacy capabilities in order to accommodate the different needs of specific target cases, by providing support for confidentiality, integrity, authentication, immutability and non-repudiation. It provides mechanisms to authenticate and secure data in transit, establishing trust among different architectural components of PLATOON platform, providing authentication both for end users identities and backend components, and therefore authorization enforcement for accessing the protected data resources.

The module, in addition, provides functionalities in line with the privacy by design principles. It ensures that the system complies with privacy regulations, in particular GDPR, during its operation and data exchanges. It performs Data Usage Control, to give access to personal or other policy regulated data, only if the intended use complies with those defined by usage policies. In the specific case of end user personal data, the Consent Management functionality additionally enforces GDPR compliancy of data requests, based on consent given by data owners (data subject). Furthermore, any personal data and their interrelationships, are hidden from plain view, thus they cannot easily be abused.

PLATOON platform is envisaged as a federated platform where several organizational platforms can interact. The following security and privacy modules have the objectives to meet inter organization security and privacy requirements. If each of the individual organization platforms already covers the requested security and privacy requirements then we only to protect the data flows between platforms at the communication level. Therefore, PLATOON will mainly focus on developing the solutions to ensure security and privacy in the data exchange between platforms. As part of the project no specific security and privacy component will be developed for protecting the individual platforms from the different partners. This means that the internal security and privacy issues of the existent pilot infrastructures (that already have their security capabilities) will be maintained and managed by the pilot owners and will be not replaced by PLATOON security component that, instead cover the security and privacy inside the PLATOON platform and in in particular in the data exchange flows between different pilots. In any case it will be necessary to assure interoperability between the PLATOON security and the existing security systems of the pilots to avoid any security breach in the internal data transferring (pilot infrastructure – PLATOON platform). This interoperability and secure data flow can be achieved in different ways (e.g. at application and networking level) to be identified during the implementation phase and depending by the specific technologies already in place in the pilot infrastructure. More technical details about PLATOON security capabilities will be provided in (11).

To summarize, the Security & Privacy module provides means to identify components, protect communications and data exchange transactions, but even control the use of data been exchanged, i.e. data sovereignty.

According to the above consideration the security, privacy and sovereignty layer will be composed of the following components:

- Identity and Access Management (IAM)
- Certification and Provisioning Services
- Data Access Control
- Data Usage Control

- Consent Manager
- Audit Logger
- Transparency Dashboard

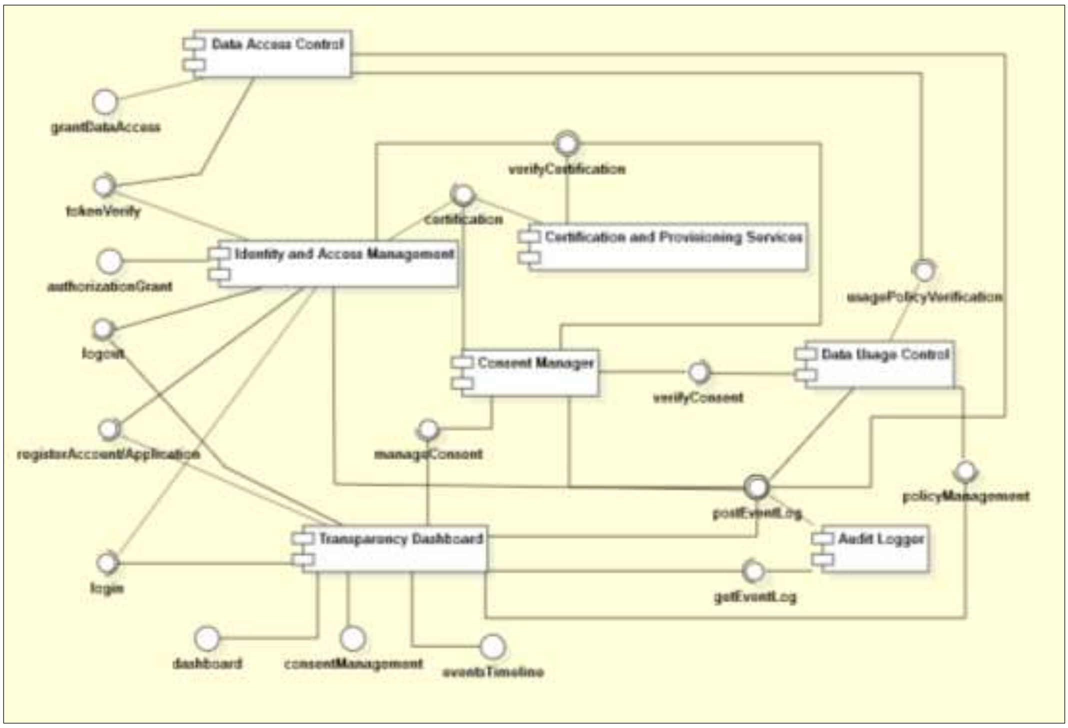

Figure 4: PLATOON Security and Privacy detailed component view

Component Name	**Identity and Access Management (IAM)**
Module Description	This component provides the functionalities of Identity Provider, offering services to create and manage identity credentials and information, which are then certificated by Certification and Provisioning component. Identities can represent users or client applications with related roles, even grouped under organizations, groups, and so on. Therefore, the IAM will provide authentication methods to validate credentials associated to those identities. The component acts also as Authorization Server, enabling client applications to validate end users' credentials on their behalf and lastly generating and validating associated access tokens (e.g. according to OAuth2 (12) authorisation framework). In addition to users, also each component itself can verify identity of any other one, by the means of Authorization Server Oauth2 functionalities (client credentials grant) and related certificates issued by Certification component.

<table>
<tr><td><b>Main functionalities</b></td><td>Identity Manager functionalities:
<ul>
<li>Users, client applications, organizations management.</li>
<li>Authentication both for end users identity (login, logout, SSO) and for application/internal components.</li>
<li>Authorization Server: making access decisions (token enforcement) based on an authenticated identity, namely securing APIs and providing user authorization to access protected resources. (e.g., access token generation and validation).</li>
<li>Dynamic Attribute Provisioning after authenticating.</li>
</ul></td></tr>
<tr><td><b>Main logical Interfaces</b></td><td>
<table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>registerAccount</td><td>Register a user with specific roles</td></tr>
<tr><td>registerApplication</td><td>Register an application and configure grants and access roles</td></tr>
<tr><td>login</td><td>Login request to Identity System, parameter according to the configured authentication type (e.g. password, Multi-factor and/or federated authentication.)</td></tr>
<tr><td>logout</td><td>Logout request to Identity system, optionally in case of SSO logout to all applications</td></tr>
<tr><td>authorizationGrant</td><td>Get authorization grant, namely the resource owner's authorization (to access its protected resources) used by the client to obtain an access token. Different protocol and flows are envisaged.</td></tr>
<tr><td>tokenVerify</td><td>Verify authorization token</td></tr>
</table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>13020, 12009, 12021, 12025, 12033</td></tr>
<tr><td><b>Interaction with other components</b></td><td>Certification and Provisioning Service, Data Access Control
<table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>Marketplace</td><td>Authentication (both for end users and communicating components) and application authorization</td></tr>
<tr><td>Data Analytic Dashboard</td><td>Authentication (end users) and application authorization</td></tr>
<tr><td>Data Analytic Tools</td><td>Authentication (end users) and application authorization</td></tr>
<tr><td>Complex Event Processing</td><td>Application authorization</td></tr>
<tr><td>IoT Connector</td><td>Application authorization</td></tr>
<tr><td>Data Connector</td><td>Application authorization</td></tr>
<tr><td>Certification and Provisioning Service</td><td>Internal communication in Security and Privacy layer to issue, validate and</td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td><table><tr><td></td><td>revoke Digital Signatures and related digital certificates (X.509)</td></tr><tr><td>Data Access Control</td><td>call to verify access token</td></tr></table></td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Certification and Provisioning Services</b></td></tr>
<tr><td><b>Module Description</b></td><td>This component is in charge of issue, validate and revoke Digital Signatures and related digital certificates (X.509) (13) used by each platform entity. It sends the digital certificate to a component (e.g. Data Connector) in a secure and trustworthy way, to certificate its identity and related metadata attributes and to enable secure encrypted communication. (e.g. Transport Layer Security).
Each Component must undergo certification and thus have a valid X.509 certificate, in order to establish trust among interacting parties and to enable the possibility of remote attestation. This requirement allows for trustworthy identification and authentication by using a central public key infrastructure (PKI).</td></tr>
<tr><td><b>Main functionalities</b></td><td>Component and identify certification
Digital Certificates and key pairs management and provisioning (Public Key Infrastructure)</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table><tr><th>Interface name</th><th>Description</th></tr><tr><td>certification</td><td>Provide Digital Certificates and related metadata for a specific entity (component or account)</td></tr><tr><td>verifyCertification</td><td>Verify a specific certificates and its status</td></tr></table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>12009, 12025, 12033</td></tr>
<tr><td><b>Interaction with other components</b></td><td><table><tr><th>Interfacing Component</th><th>Interface Description</th></tr><tr><td>Identity and Access Management</td><td>Send and verify certificates</td></tr><tr><td>Data Connector</td><td>Send and verify certificates</td></tr><tr><td>Marketplace</td><td>Application metadata</td></tr></table></td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Data Access Control</b></td></tr>
<tr><td><b>Module Description</b></td><td>This component ensures further authorization enforcement restricting unauthorized access to data resources, by relying on Role Based and/or Attribute Based Access Control models (RBAC, ABAC). Once the access token has been validated by the Authorization Server (IAM), this component then can perform Access Control decisions on requested data, also relying on policy languages such as XACML (Extensible Access Control Mark-up Language) (14) or ODRL (15) (Open Digital Rights Language).
The component will act as a Policy Enforcement Point (PEP), by monitoring the platform actions and intercepting incoming data requests. It will grant access on the decisions made by the Data Usage Control (acting as a Policy Decision Point).</td></tr>
<tr><td><b>Main functionalities</b></td><td>Role-based access control (RBAC)
Attribute-based access control (ABAC)
Policy Enforcement Point.
Policy Decision Point</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>grantDataAccess</td><td>Grant access to a specific resource against specific access and usage policy and context</td></tr>
</table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>13020, 12009, 12025, 12033, 12036</td></tr>
<tr><td><b>Interaction with other components</b></td><td><table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>Identity and Access Management</td><td>Validate access token</td></tr>
<tr><td>Data Usage Control</td><td>Verify usage rules</td></tr>
<tr><td>Audit Logger</td><td>Store access events</td></tr>
<tr><td>Data Connector</td><td>Perform access and usage enforcement</td></tr>
<tr><td>Marketplace</td><td>Perform access and usage enforcement</td></tr>
</table></td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Data Usage Control</b></td></tr>
<tr><td><b>Module Description</b></td><td>As the Usage Control concept is an extension of Access Control, the target of this component is to enforce restrictions on data usage and data processing, after access to data has been granted by the IAM and Data Access components. Those policies compose a Usage Contract, regulating what may be done with a data asset and what not and for which purpose. This component is in charge of ensuring that a component (acting as Data Consumer) handles the data provided by another component (acting as Data Provider) according to the usage policies specified. It will act as a Policy Decision Point (PDP), with its decision engine allowing or denying an action, and optionally requiring the modification of the action.
Data Usage Control component must guarantee that even its own enforcement system is trusted; this can be achieved by interacting with Certification and Provisioning Services.
Additionally, it enables the specification and management of the policy rules themselves.
In the specific case of a request involving Personal Data, the component also enforces the compliancy with data usage/data processing Consents given by Data Owner, according to the GDPR regulation, by interacting with the Consent Manager component.</td></tr>
<tr><td><b>Main functionalities</b></td><td>Policy Decision Point
Policy Administration and Management Point (Specification and management of the data usage policies).</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>policyManagement</td><td>lifecycle management of usage rules</td></tr>
<tr><td>usagePolicyVerification</td><td>Check if a policy is verified. Policy is referred to a specific resource and context</td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>13020, 12007, 12009, 12033</td></tr>
<tr><td><b>Interaction with other components</b></td><td><table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>Data Access Control</td><td>It provides verification that the usage rule associated to the requested resource is valid</td></tr>
<tr><td>Consent Manager</td><td>In case of Personal Data, it verifies the legal basis of usage rules</td></tr>
<tr><td>Audit Log</td><td>Store event log on usage rules lifecycle management and their execution during policy verification.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Consent Manager</b></td></tr>
<tr><td><b>Module Description</b></td><td>It is a consent-based and user-centric component for management and enforcement of Personal Data Usage Consents for those services having the role of Data Consumer and/or Data Provider. This component is in charge, in case of data requests involving end user personal data, of ensuring compliancy with GDPR regulation, by enforcing data usage/processing Consents given by the Data Owner. In particular, the component acts as an intermediary and as a tool of communication between data subjects and controllers/processors, supporting the entire end-to-end process of generation and management of dynamic consents. The end user privacy is ensured by enabling him to grant and withdraw consents to third parties for accessing and processing his own data. Consent authorizes a Data Source to provision data to a Data Consumer and authorizes Data Requester to process that data. Consent must be linked with a Data Usage Policy to be formalized.
Optionally, the Consent Manager can even handle the actual data requests between components, by issuing authorisation tokens associated to a given Consent.</td></tr>
<tr><td><b>Main functionalities</b></td><td>End User Consent lifecycle management
Consents verification and enforcement of related restrictions and personal Data Usage Policies.</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td>
<table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>manageConsent</td><td>Management of Consent lifecycle (Consent form, modification, etc.)</td></tr>
<tr><td>verifyConsent</td><td>Verify if consent is valid and its status</td></tr>
</table>
</td></tr>
<tr><td><b>Requirements Mapping</b></td><td>13020, 12007, 12033, 12035</td></tr>
<tr><td><b>Interaction with other components</b></td><td>
<table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>IAM</td><td>It gets from IAM the identification of user</td></tr>
<tr><td>Data Usage Control</td><td>It provides validation check of consent and related privacy rules (purposes, processing,..)</td></tr>
<tr><td>Transparency Dashboard</td><td>It provides consent list for a specific logged user</td></tr>
<tr><td>Audit Logger</td><td>It sends consent related events</td></tr>
<tr><td>Certification and Provisioning Services</td><td>It interacts with that modules for Consent signature (user, application)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Transparency Dashboard</b></td></tr>
<tr><td><b>Module Description</b></td><td>Transparency Dashboard provides an overview of the data usage compliant to the enforced usage policies and restrictions, by the means of a set of charts and statistics. The target of this components is to accomplish transparency concerning data flows and data usage in compliance both with usage control policies and GDPR regulation. For this purpose, Data Usage Control and Audit Logger components complement each other.</td></tr>
<tr><td><b>Main functionalities</b></td><td>Data usage charts visualization
Filtering on visualized charts
Consent management</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td>
<table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>dashboard</td><td>Visual dashboard about the personal data usage or data usage</td></tr>
<tr><td>consent management</td><td>Visual consent lifecycle management</td></tr>
<tr><td>eventsTimeline</td><td>Visual timeline of occurred events.</td></tr>
</table>
</td></tr>
<tr><td><b>Requirements Mapping</b></td><td>13020, 13066, 12030</td></tr>
<tr><td><b>Interaction with other components</b></td><td>
<table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>Consent Manager</td><td>Visualization and consent management</td></tr>
<tr><td>Data Usage Control</td><td>Get data usage statistics (personal or not)</td></tr>
<tr><td>Audit Logger</td><td>It gets the occurred event logs to display the visual timeline</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Audit Logger</b></td></tr>
<tr><td><b>Module Description</b></td><td>This component is in charge of tracking several events such as the usage of data during data exchange transactions and the enforcement of usage restrictions, providing evidence of compliant data usage: how and by whom data was accessed or processed. This data will be used later on Transparency Dashboard.
This component can work either with event-based (centralized) or flow-interception (distributed) way of logging data exchange transactions.</td></tr>
<tr><td><b>Main functionalities</b></td><td>Centralized or distributed auditing of data exchange transactions by either receiving events or actively intercept data flows.
Query and Filtering services of audited data</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td>
<table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>getEventLog</td><td>Query and Filtering event logs</td></tr>
<tr><td>postEventLog</td><td>Store specific event/s</td></tr>
</table>
</td></tr>
<tr><td><b>Requirements Mapping</b></td><td>13020, 12006, 12009</td></tr>
<tr><td><b>Interaction with other components</b></td><td>
<table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>Transparency Dashboard</td><td>Get/register events</td></tr>
<tr><td>Data Usage Control</td><td>register events</td></tr>
<tr><td>Data Access Control</td><td>register events</td></tr>
<tr><td>Consent Manager</td><td>register events</td></tr>
<tr><td>Data Connector</td><td>register events</td></tr>
<tr><td>IAM</td><td>register events</td></tr>
<tr><td>Other components requesting event registration (TBD)</td><td>register events</td></tr>
</table>
</td></tr>
</table>

3.5.1 Data Sovereignty and IDS Components

Today, facilitating a secure and standardized data exchange and linked data in a trusted business ecosystem is essential. The International Data Spaces (IDS) reference architecture plays a vital role in achieving this goal in PLATOON. The goal is to facilitate a decentralized platform enabling data and services exchange. Based on existing European standards, PLATOON offers to build a solution that manages the pilots' data ensuring governance, and sovereignty by adapting IDS' architecture and integrating it with the security and Consent Manager modules of the reference architecture. The complete IDS specification can be found in the official IDS refence architecture [16], in this section we will define some key logical components derived from IDS that are defined as part of the PLATOON to cover the aspect of Data security and Sovereignty in conjunction with the other components of the architecture. **The Connector** is the key building block of the International Data Spaces Architecture allowing the different entities that are part of the ecosystem to exchange, share and process digital content maintaining data sovereignty of the Data Owner. Data Apps are data services encapsulating data processing and/or data transformation functionality bundled as container images for simple installation by application container management.

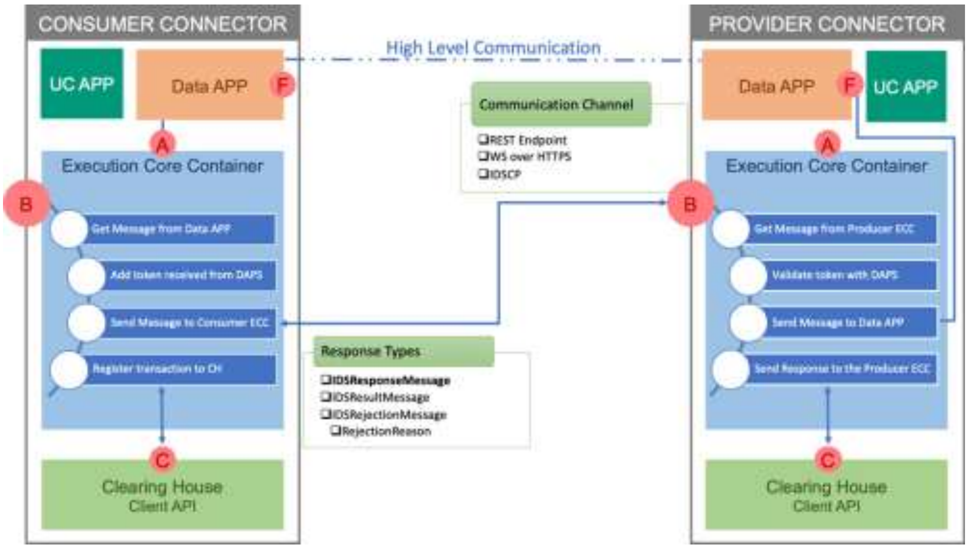

Figure 5: IDS connectors communication

Two different entities (Data provider/Data consumer) using two certified IDS connectors can exchange data, using secure transmission protocols, assuring the reliability of the parties involved defining specific usage control data policies.

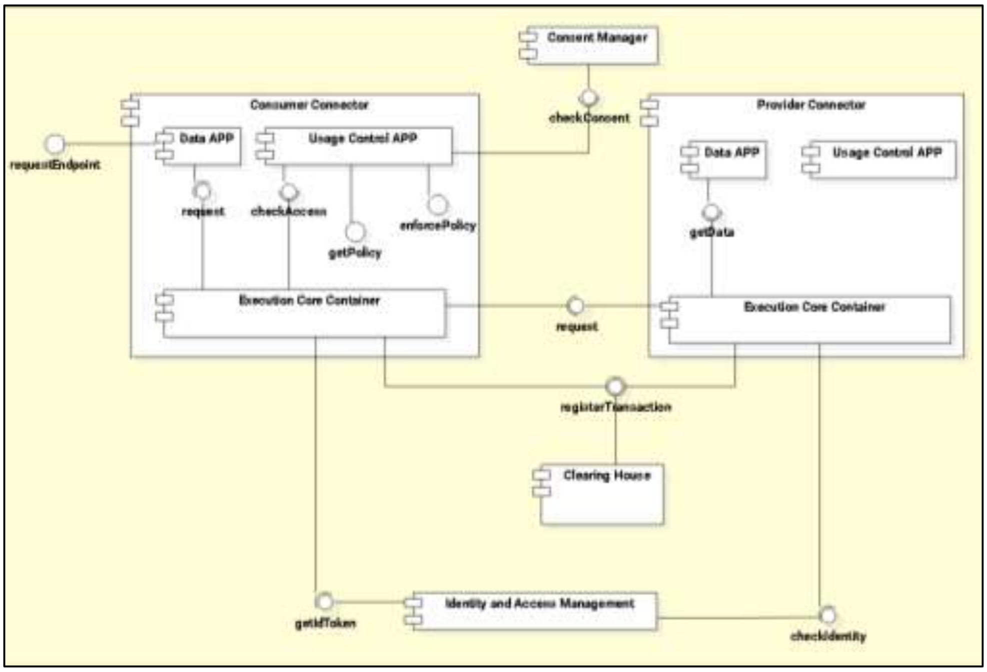

Figure 6: IDS components integration with PLATOON security

The Figure 6 shows how the components of the IDS connectors along with the IAM and Consent Manager, already belonging to PLATOON security layer, are connected to ensure secure interchange between different organizations. In the following we will describe the IDS components integrated in PLATOON and in section 5.5 is presented the flow by which the data interchange takes place. In addition, the consumer connector can query the Vocabulary Manager (IDS Vocabulary Provider developed in D3.5) defined in the interoperability layer and exchange metadata linked to the PLATOON data models defined in D2.3/D2.7 to understand the datasets/services offered by the provider connector.

<table>
<tr><th>Component Name</th><th>Usage Control APP</th></tr>
<tr><td>Module Description</td><td>It is in charge of providing and applying usage control policies to the data.
Usage Control APP acts only when a connector assumes the role of data consumer.</td></tr>
<tr><td>Main functionalities</td><td>After retrieving all the policies associated with a dataset from the usage agreement previously negotiated between the parties, it takes care of enforcing those policies.
In addition, if personal data is present in a dataset UC is responsible for verifying consent with the Consent Manager.</td></tr>
<tr><td>Main logical Interfaces</td><td><table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>getPolicy</td><td>retrieves from the data interchange contract the policies associated with a given dataset</td></tr>
<tr><td>enforcePolicy</td><td>ensures that all policies associated with a dataset are respected and, if necessary, performs any necessary data transformations or data deletion when necessary.</td></tr>
</table></td></tr>
<tr><td>Requirements Mapping</td><td>12007, 12009, 12012, 12026, 12027, 12030, 12035, 12031,13008</td></tr>
<tr><td>Interaction with other components</td><td><table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>Execution Core Container</td><td>Consumer ECC invokes UC in order to enforce policy before send data to Consumer Data APP</td></tr>
<tr><td>Consent Manager</td><td>UC APP invokes Consent Manager in order to retrieve consent to use a dataset that contains personal data</td></tr>
<tr><td></td><td></td></tr>
</table></td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Data APP</b></td></tr>
<tr><td><b>Module Description</b></td><td>Data APP represents a data application build to generate and/or consume data on top of the Execution Core Container.
When the connector acts as a data provider Data APP performs the task of accessing the actual data and retrieving it, for example by making a query to a database or invoking a service. when the connector acts as a data consumer Data APP performs the task of exposing an endpoint where to request access to a given dataset.</td></tr>
<tr><td><b>Main functionalities</b></td><td>Get dataset from physical repository
Provide public data access interface</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>request</td><td>request access to a given dataset</td></tr>
<tr><td>getData</td><td>Retrieve data from repository or internal IT system</td></tr>
</table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>12009, 12012, 12037, 12031, 13008</td></tr>
<tr><td><b>Interaction with other components</b></td><td><table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>Execution Core Container</td><td>ECC asks for actual data when Data APP acts as Provider. Data APP forward external request to ECC when it acts as consumer.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Execution Core Container</b></td></tr>
<tr><td><b>Module Description</b></td><td>Execution Core Container (ECC) is the core component of the Connector. It is in charge of the data exchange using specific data transmission (e.g. HTTP/HTTPS, WS over HTTPS or IDSCP2) taking advantage of the IDS Information Model to represent the data.</td></tr>
<tr><td><b>Main functionalities</b></td><td>When the connector acts as Consumer, it forwards requests from Data APP to a provider's ECC after getting a valid token from the IAM.
When the connector acts as a Provider it fulfils requests by obtaining datasets from the provider Data APP</td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table>
<tr><th>Interface name</th><th>Description</th></tr>
<tr><td>request</td><td>receives requests to retrieve a dataset</td></tr>
</table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>12006,12009, 12012, 12021, 12030, 12037, 12031, 13008</td></tr>
<tr><td><b>Interaction with other components</b></td><td><table>
<tr><th>Interfacing Component</th><th>Interface Description</th></tr>
<tr><td>IAM</td><td>The interactions with an external Identity Provider to require and validate a token.</td></tr>
<tr><td>Clearing House</td><td>The communication with the Clearing House for registering the transactions</td></tr>
<tr><td>Data APP</td><td>It forwards requests from Data APP (consumer role). It retrieves datasets through the Data APP (provider role)</td></tr>
<tr><td>Usage Control</td><td>When acting as a data consumer after receiving a data set from the ECC Provider it invokes the UC to verify and enforce access and consent policies.</td></tr>
</table></td></tr>
</table>

<table>
<tr><td><b>Component Name</b></td><td><b>Clearing House</b></td></tr>
<tr><td><b>Module Description</b></td><td>This component acts as an intermediary in the business ecosystem as well as PLATOON Marketplace. This means that the Clearing House mediates between a Data Provider (DP) and a Data Consumer (DC), ensuring both parties meet their contractual obligations, such as:<ul><li>The DP sharing data with the DC according to Usage Contracts and Data Usage Policies defined</li><li>The DC using data according to Usage Contracts and Data Usage Policies defined and effecting payment to the DP as agreed.</li></ul>For each data exchange transaction, the DP attaches metadata to the data requested by the DC, specifying data usage restrictions, pricing information, payment entitlement, time of validity, etc.</td></tr>
<tr><td><b>Main functionalities</b></td><td><ul><li>Log Message: Each party in a data exchange process logs their contractual obligation to the Clearing House.</li><li>Query Message: Each party can query the Clearing House to ensure the contractual obligation is preserved.</li></ul></td></tr>
<tr><td><b>Main logical Interfaces</b></td><td><table><tr><th>Interface name</th><th>Description</th></tr><tr><td>registerTransaction</td><td>Store transaction details</td></tr></table></td></tr>
<tr><td><b>Requirements Mapping</b></td><td>12006, 12009, 12037, 12031, 13008</td></tr>
<tr><td><b>Interaction with other components</b></td><td><table><tr><th>Interfacing components</th><th>Interface Descriptions</th></tr><tr><td>Execution Core Container</td><td>Provider and Consumer ECC record transactions on the CL</td></tr></table></td></tr>
</table>

4 Edge capabilities and deployment scenarios

With the increasing number of connected devices in electric power systems, edge computing has started gaining momentum over the past years. With the growth of connected devices, data is rapidly growing and unsustainable under cloud-only models. Internet of Things (IoT) has made small sensors, devices and things available to the internet, offering advanced control and monitoring services. Handling a massive number of sensors and data sources, that continuously collect high-resolution data, and managing large volumes of data have been identified as two of the major challenges. With an increasing number of IoT devices, the network bandwidth is approaching the limit, despite the improvements made in the network technologies. Furthermore, data centers cannot guarantee acceptable transfer rates and response times from IoT devices required by many applications. Edge computing brings computation and data storage closer to where it is needed, as close as possible to the originating source. For this reason, edge computing is being used to build content delivery networks that decentralise the provision of data and services and bring it closer to the end users.

With the use of edge computing, privacy and security are even more important since the edge devices are usually more easily physically accessed compared to central computers. Data must be encrypted, and different encryption mechanisms must be used because data is transferred across different network nodes. Also, end devices can have resource constraints and so appropriate encryption mechanisms must be used. Edge computing also enables the shift of data ownership from service providers to end users.

Edge computing capabilities of the PLATOON platform will enable distributed computing, storage and control at the edge of the PLATOON system. The motivation behind it is to address latency and security and reduce required network bandwidth and energy consumption.
PLATOON edge capabilities will be deployed in the pilot sites IT system infrastructure to be close to pilot physical infrastructure and data sources. Edge infrastructure will provide another data source and asset control for the PLATOON.
The diagram below extends on the PLATOON platform with the PLATOON edge architecture.

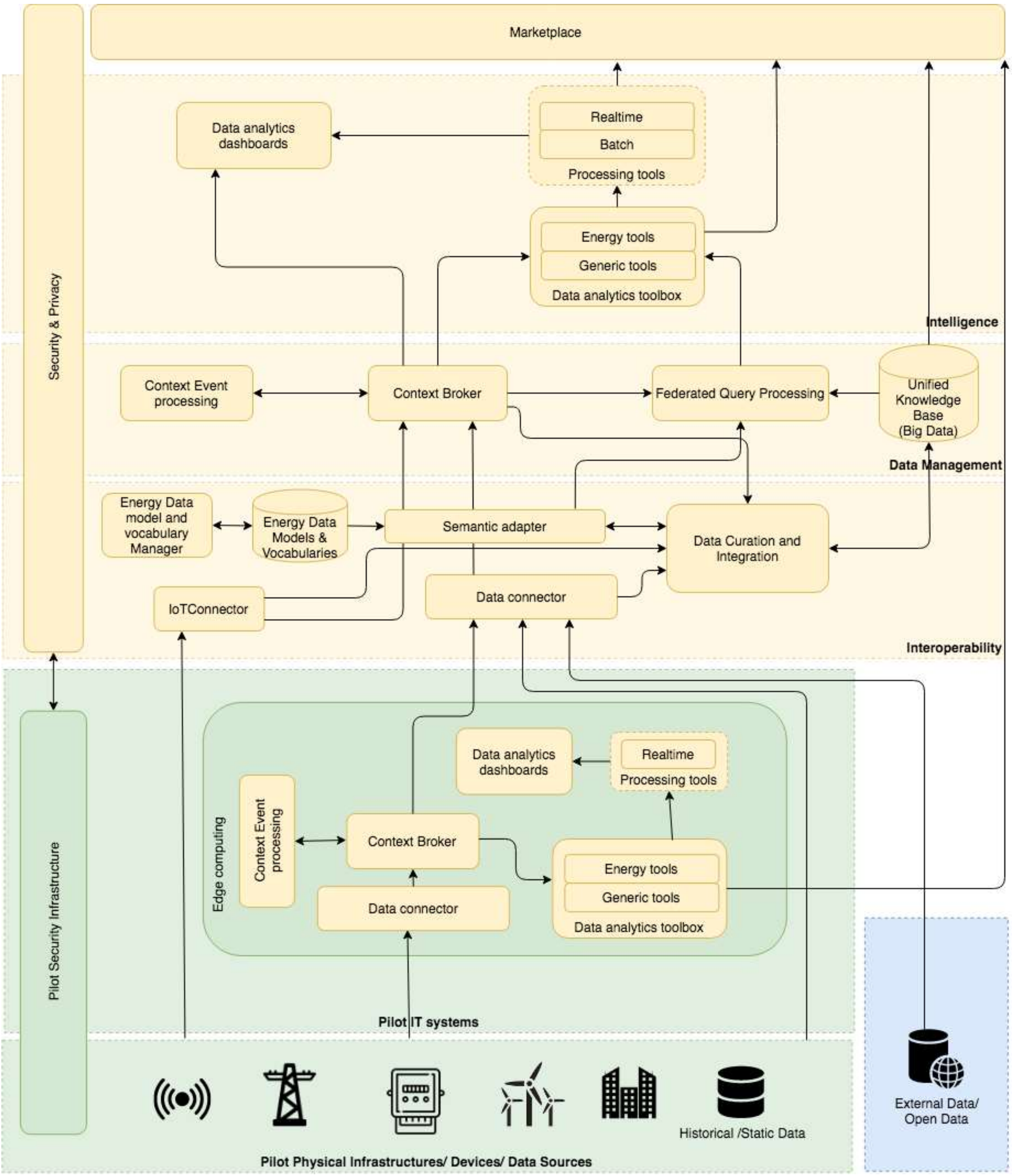

Figure 7: PLATOON architecture edge capabilities

It is important to remark that the architecture presented in the above figure does not imply that all the depicted components should be duplicated at cloud side and edge side in every concrete implementation of the PLATOON architecture, but simply that a specific component, in relation to its role, could be deployed at cloud or edge side depending on real conditions, specific pilots needs and non-functional requirements (e.g. privacy or security constraints). For this reason, there could be situations where data collection and processing will be entirely performed in the cloud and others in which hybrid situations can be adopted. Moreover, the

different deployment of a component at the edge or cloud will change its scope and capabilities due to the different hardware and networking conditions.
For instance, two of the most resource consuming components, the Data Analytics toolbox and processing tools, in most of the contexts, should be deployed on the cloud, but in some cases, for simple and limited data processing tasks (e.g. lightweight real-time analytics and report), it could be also possible to deploy a limited version of these components at the edge.

4.1 Deployment scenarios

We have proposed several deployment scenarios to address various PLATOON project use cases. To distinguish between the Edge node and central computer, we have to point out that the Edge node only incorporates reduced PLATOON architecture. Minimally it should include data connector and Data analytics tools component. Therefore, it has to do at least some transformation of the input data or enable events triggering. On the other hand, due to reduced system resources of edge devices, it is not suitable to implement all the components at the edge. Furthermore, the main priority for the edge node is expected to be real-time processing and response. Data analytic tools at the edge usually process only raw data. The central layer in the use case scenarios presents the central computer, which can be deployed in the cloud, close to the premises of the pilot or even close to the asset from which it reads the data and to whom it provides the control signals. To conclude with, the main difference between the Edge node and a Central node is that the Edge node processes data from a single or limited number of assets locally (rapid response) but has reduced resources by means of processing power and deployed components of the PLATOON architecture. The central computer, on the contrary, processes data from multiple assets in a central environment (takes time to collect, process and send back data to each asset) but has much larger processing and storage resources which allows to do more complex operations and process more data at the same time obtaining larger generalisation capability.

4.1.1 Distributed control plane scenario

Each edge site has its own independent control plane. On the one hand, deployment provides greater autonomy from the central data center, but on the other hand, the large quantity of independent control planes makes it harder to maintain.

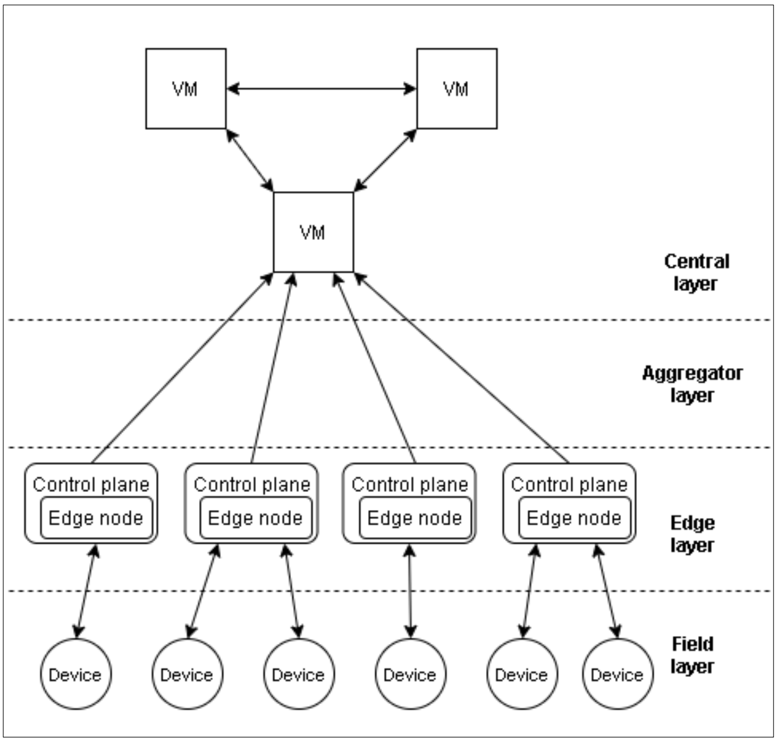

Figure 8: Distributed control plane scenario layers

The advantage of a distributed control plane scenario is that it allows easy management of different use cases and adds more functionality to the edge. In the case of a network connection loss, all required services at the edge node can keep running, independently from the central node, and send the data when the connection becomes available again. At the edge, real-time data analytics is performed, enabling real-time observability and autonomous reactive decision making. The analytic toolbox on the central computer is responsible for post-processing of the data obtained by analytical toolbox of Edge node. Furthermore, the analytical toolbox at the central computer operates with aggregated data from distributed Edge nodes to provide complete system observability.

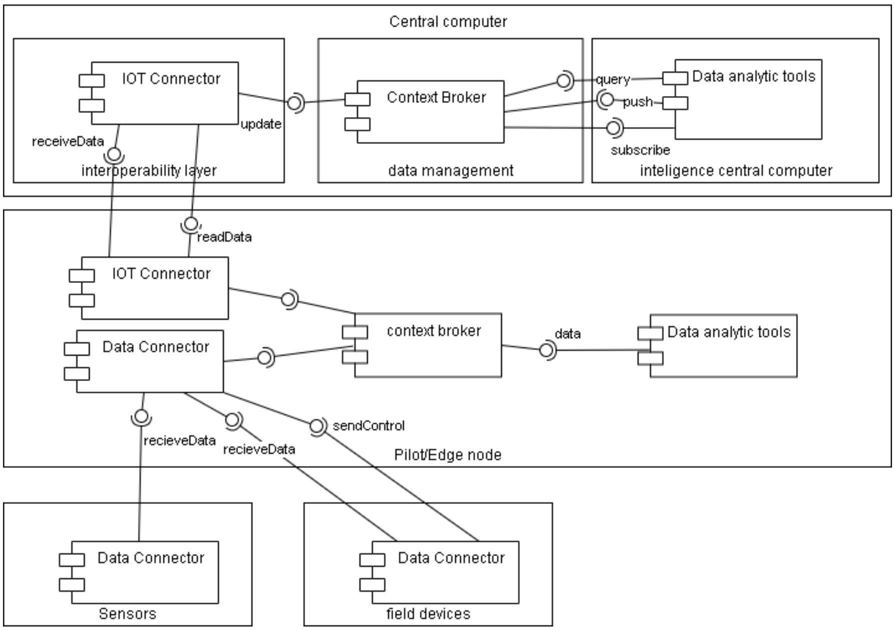

Figure 9: Distributed control plane scenario components

4.1.2 Centralized control plane scenario

All edge sites are managed from a control plane in the central data center. Such architecture greatly simplifies the management of computing resources across the network. However, the instances of the edge cannot be managed in case of partitioning.

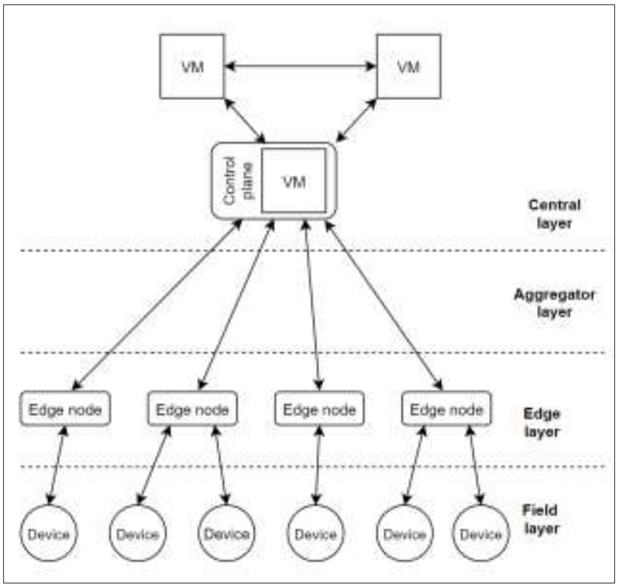

Figure 10: Centralized control plane scenario layers

The central computer includes the control plane, which is responsible for identifying Edge nodes, their management and orchestration. Such architecture provides a centralized view of the infrastructure. Data analytic toolbox services in the central computer should be resilient if one of the nodes fails. Since the control plane is placed on the central side, the architecture is more secure. However, in the critical situation at the Edge node, such architecture does not allow users to interfere in order to mitigate the problem. In such configuration, the main idea is that the services in analytical toolbox at the Edge pre-processes the data (i.e., feature extraction or simple transformation) from the raw data sources, while the centralized decision making is done by the services from the analytical toolbox in the central computer.

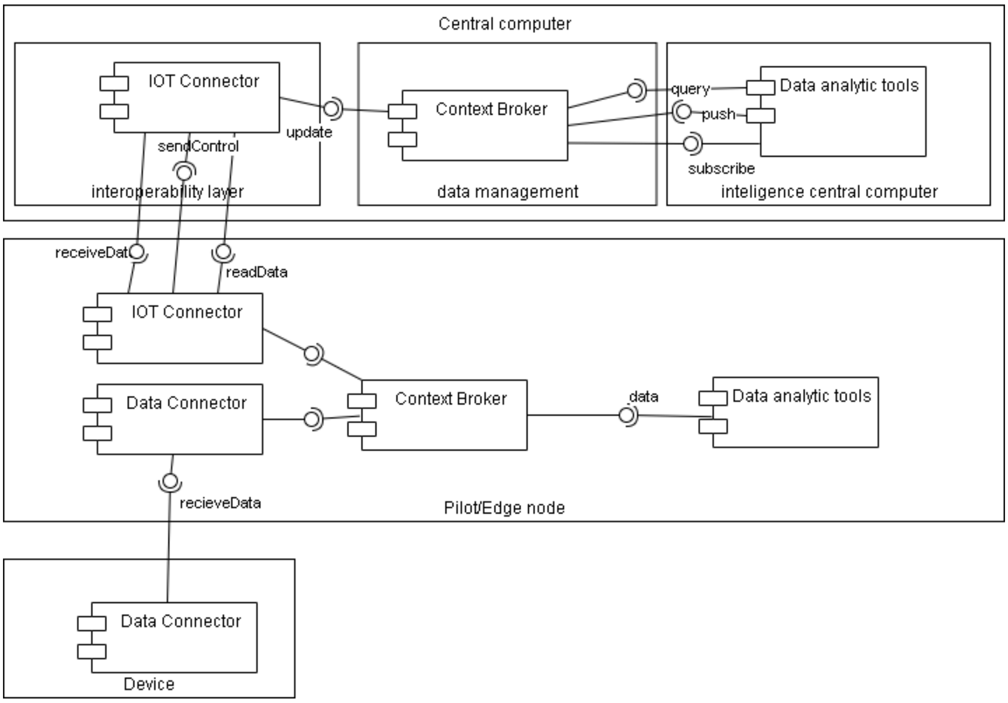

Figure 11: Centralized control plane scenario components

4.1.3 Aggregated control plane scenario

Individual edge sites are aggregated and managed from a control plane in one of the edge nodes. Such architecture enables partitioning for a specific region. Therefore, different autonomous zones might exist in the system, allowing even higher flexibility and easier management of a particular zone.

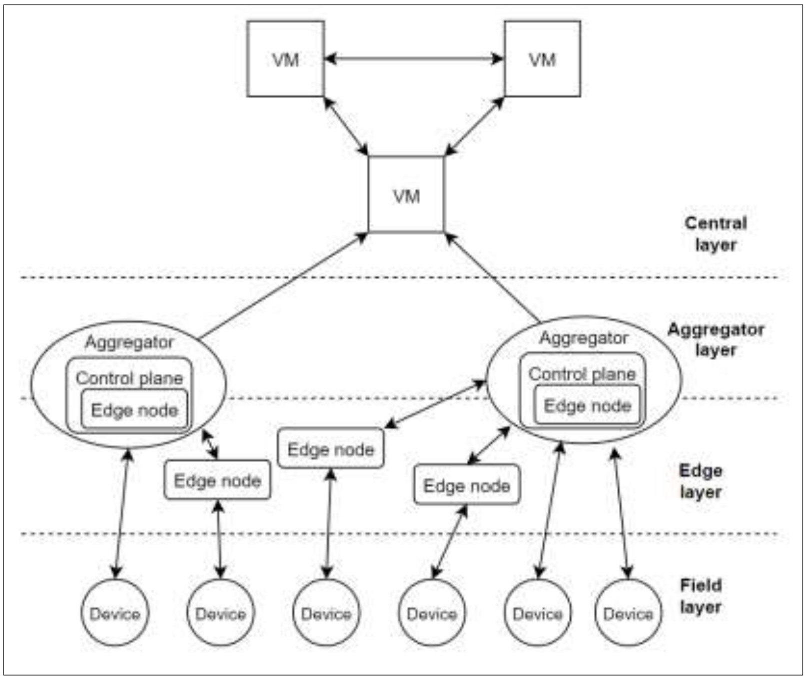

Figure 12: Aggregated control plane scenario layers

In this approach, the edge nodes are divided into different self-sufficient and autonomous zones performing their own specific tasks. One of the Edge nodes controls the management and orchestration of all other Edge nodes within a particular zone, which are providing pre-processed data to this particular main controllable edge node. The advantage of such approach is that it provides better response times than a centralized control plane scenario, since the control plane is part of one of the Edge nodes located at the pilot's premises. Performing the main analytics at one node improves real-time controllability and mitigates latency compared to a centralized control plane scenario. Furthermore, since the zones are closed and therefore not interdependent, they are generally also more secure. If one of the zones fails or is compromised, others are not affected. However, the idea of aggregated edge node functionality is still similar to the central computer approach; therefore, it is reasonable to include some additional components at the Aggregated node (e.g. data analytical dashboard for observability of the corresponding self-sufficient zone). Nevertheless, for observability of a network as a whole and additional post-processing the autonomous zones are still sending analysed data to the central computer. The analytic tools, in the edge node with the control plane, should be resilient to the missing data from other nodes. In the case of a network disconnection loss of zones, all required services at the edge node with the control plane can still run, independently from the central computer, and send the data when the connection is available.

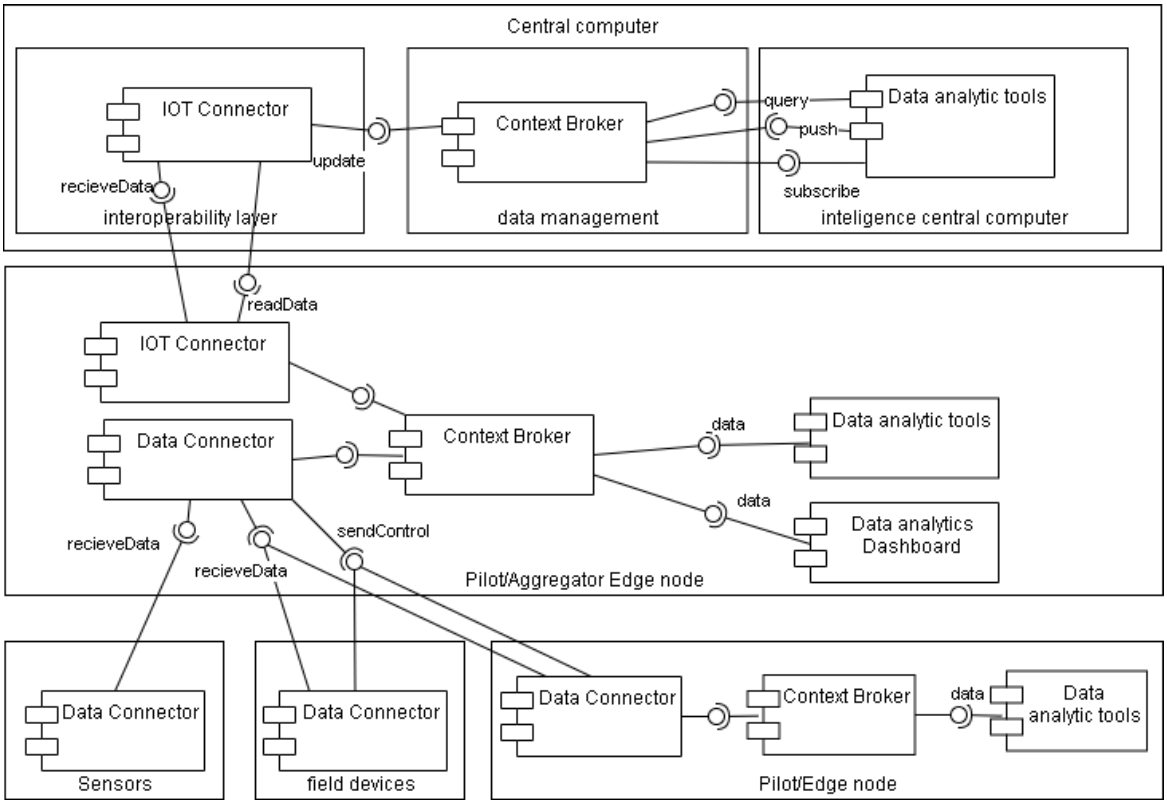

Figure 13: Aggregated control plane scenario components

4.2 Platoon Edge-Cloud Framework

In addition to what has been presented in the previous section the work on edge/cloud paradigm in relation to PLATOON architecture proceeded inside the WP4 (see deliverable D4.2) defining a more detailed in relation to the edge-cloud Framework, providing also some details about the implementation carried out in the Platoon project. Although the components related to interoperability and data connectors remained the same, the edge-cloud Framework presents one step further and introduces the tools that enhanced the functionalities related to node and service management and exploiting the real-time capabilities of the edge node. Some of the components can also be deployed at the edge (e.g. semantic enhancement related components) to extend the interoperability also down to the field devices. The scope of the edge-cloud framework is to make the platform more resilient to external influences that are out of our control (e.g. issues with network outages), enable easy deployment and the orchestration of the services at the edge, offload network transfer, increase local observability and enable autonomous reactions.

The main idea of the PLATOON edge-cloud framework is to connect the field sources, enabling pre-processing and performing some analytics at the edge. The edge device is fully capable of running some of the analytics defined in D4.1. The output of the pre-processing or analytical tools run at the edge can be connected to other nodes and the results can be exploited by other services in the cloud or central computer. To address different requirements more efficiently, some components are optional, for instance, some of them are not needed in real-time processing but might be essential in batch processing. Furthermore, the architecture also addresses the issues with latencies and fault tolerance in different scenarios to fully exploit the PLATOON edge-cloud computing framework.

In Figure 14 is presented the framework with the reference to the tools selected for its implementation. The framework that is container-based reuses existing containers provided by the community and the it can be easily enhanced to provide wider functionality. For instance, the pilots can decide if they need a specific tool at the edge for visualization (e.g. Grafna) or not. In addition, the framework allows users to install a specific tool in a few steps and deploy it to all required nodes once the framework is deployed. One of the main advantages of the framework is that it is based on well-known and maintenance tools; this guarantee high security and stability are and the maintenance from the user perspective is minimal.

The main functional components and features of the framework are:

- **Node management:** The node management component is provided by the Rundeck[1] tool. The node management can be done on all nodes, on individual nodes or nodes grouped in so-called clusters. Nodes are managed through SSH from a single point of entry. Management can be done manually or automatically through the schedules. Some of the management functions available, but not limited to, are: restarts, recovery of logs, provisioning, updating, user account management, production patching.
- **Service management:** According to the requirements of the analytical toolbox, as defined in D4.1, the framework supports the deployment of micro-service (MS) or packaged service in a container. MS deployment and management is done from the Rundeck tool. The Rundeck supports node clustering to simplify MS management. However, in such a case, the user must manually deploy the MS and create the daemon. In this case, the service level of integration into the OS system is higher in exchange for more complex management of such services. With respect to containers, the framework uses the Portainer tool for container management. In this case, the management is done by means of the user interface. All the analytic services are fully packaged in a portable computing environment. The container encapsulates all the libraries, binaries, configuration files and dependencies, making it more portable. However, the level of the integration due to the overhead is lower, as in the case of MS.
- **Orchestration:** Container orchestration is provided by Docker Swarm by default. Since edge nodes are basically quite simple and usually only a few services are deployed, orchestration is more practical in the case of a cloud environment where more complex management is required not only for administration but also for scaling and maintaining Docker applications.
- **Security:** Security is by default provided by SSH and by optional VPN. Additionally, each tool has its own security measures. For example, Docker Swarm uses a public key infrastructure (PKI) system build into docker. Additionally, all nodes use mutual Transport Layer Security (TLS) to authenticate, authorise and encrypt communication[2]. Portainer provides security controls, e.g. to force only HTTPS and enable using SSL certificates. In the case of Rundeck, it supports a few different mechanisms to

[1] https://www.rundeck.com/

[2] https://docs.docker.com/engine/swarm/how-swarm-mode-works/pki/

authenticate and determine the privileges such as: Single Sign On, JAAS, Container Authentication and Preauthenticated mode[3].

- **Monitoring:** Monitoring of the resources vital in the time-critical situation. Therefore, the framework provides the Munin tool to monitor resources and create alarms that can be used to trigger necessary management of the services before they fail or run out of resources.
- **Scalability:** In the case of using dockers of deployment of the services, the framework is fully horizontally scalable to the new edge nodes. They can be easily incorporated into the framework by provided tools. The framework scalability is provided, but not ensured for the case of analytical tools, which is out of the domain of this framework and is a feature of the distributed analytic design.

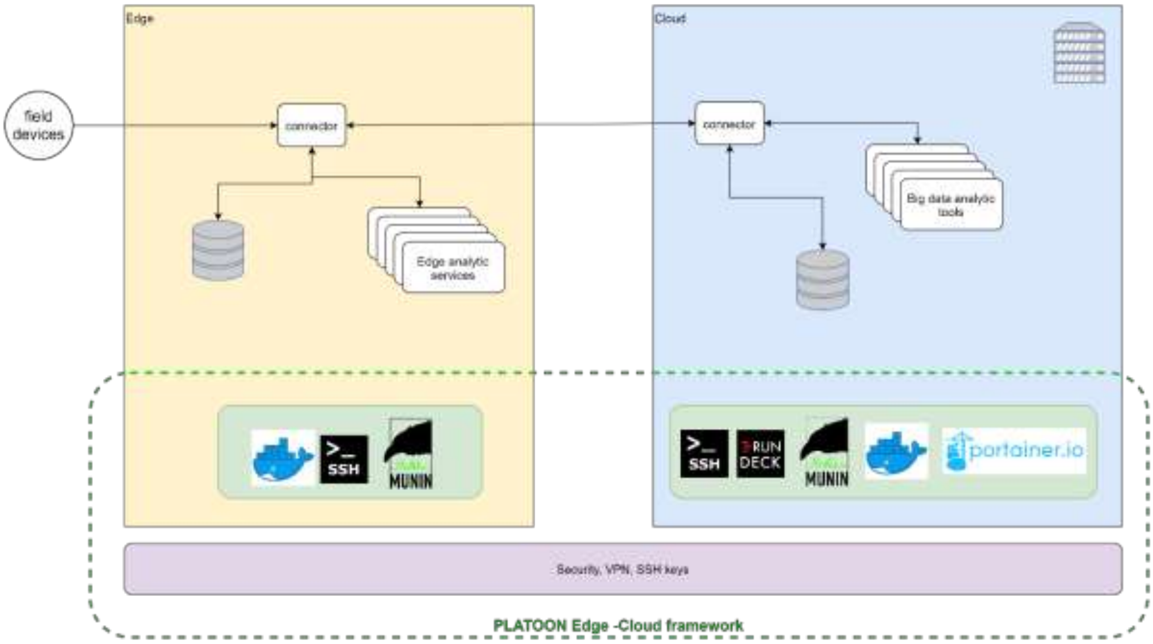

Figure 14: PLATOON Edge-Cloud framework

In summary, as depicted in the Figure 14, the edge-cloud framework is composed of tools for deploying Docker services and managing/automating the edge/cloud infrastructure. The framework consists of the following open-source tools: Portainer[4] - an open-source tool for managing containerized applications Rundeck - an open-source service for running automation tasks across a set of nodes Munin[5] - an open-source application for monitoring computer systems, networks and infrastructure.

The Framework deployment installs and configures the tools, set up SSH keys and a VPN on a cloud server, as well as the edge nodes.

The PLATOON edge cloud framework is open source and it is available on the PLATOON project GitHub[6] page. The framework is free software that can be redistributed and modified according to of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

[3] https://docs.rundeck.com/docs/administration/security/authentication.html#propertyfileloginmodule

[4] https://github.com/portainer/portainer

[5] https://munin-monitoring.org/

[6] https://github.com/PLATOONProject/edge-cloud-framework

5 Data exchange scenarios

This section will present a dynamic view of the PLATOON reference architecture, in terms of interactions and data flows that can occur between the components described in the previous chapters. The flows, that are depicted through UML sequence diagrams, are related to different types of scenarios mostly at intra-organizational level, omitting for simplicity the security aspects that will be treated in the following sections.

5.1 Intra organizational flows

The following flows are related to data management processes that take place within the same organization within a secure and trusted perimeter; nevertheless, there are security components that are omitted in this section to simplify the diagrams.

5.1.1 IoT Real Time Flow

In this scenario an IoT sensor or device sends measurements / data in real time to the Unified Knowledge Base (UKB) and then to the Marketplace. As precondition the IoT device has to be configured in order to establish a secure communication channel with the IoT Connector. This step is specific to the IoT device and the communication protocol supported.

1. The device sends raw data to the IoT Connector using its own format and protocol.
2. The IoT Connector feeds the Data Curation and Integration with the raw data using Data Curation interface.
3. Data Curation pre-processes raw data (validating and enriching) then send it to the Semantic Adapter.
4. After the data mapping with PLATOON shared semantic data models, the Semantic Adapter sends back semantic data to the Data Curation and Integration.
5. Data Curation stores semantic data to the UKB.
6. Dataset metadata are stored in the Marketplace
7. Marketplace confirms the content update.

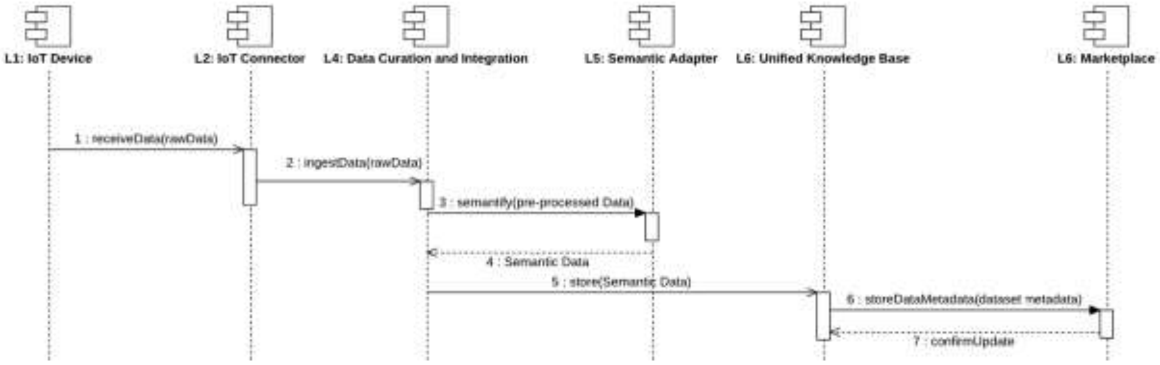

Figure 15: IoT Real Time Flow

5.1.2 Pilot Dataset flow

Similarly to the previous flow the following diagram shows a scenario where the input raw data is generic a dataset is loaded from information systems of the Pilot side and after a semantification phase is sent to the Marketplace.

1. The Pilot Information System requires authentication on the Platoon Security Layer. More information about this step is detailed in section 5.4.
2. Platoon Security Layer authorizes the Pilot Information System.
3. The dataset is sent to the Data Connector using proprietary format and protocol.
4. The Data Connector feed the Data Curation and Integration with the raw data using Data Curation ingestion interface.
5. Data Curation pre-processes raw data (validating and enriching) then send it to the Semantic Adapter.
6. After map data with PLATOON shared data models the Semantic Adapter sends back semantic data to the Data Curation and Integration.
7. Data Curation stores semantic data to the UKB.
8. Dataset metadata are stored in the Marketplace.
9. Marketplace confirms the content update.

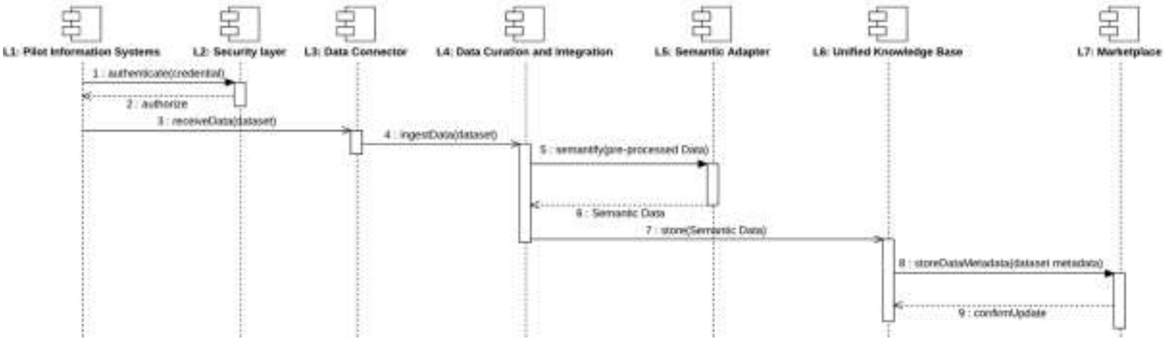

Figure 16: Pilot Dataset flow

5.1.3 Complex Event Processing flow

This flow describes how the CEP together with the Context Broker can add a first basic level of processing to the PLATOON architecture allowing to infer new information as real time data arrives from the southbound:

1. The event processing rules are defined using CEP rule management interface (Setup phase).
2. The CEP subscribes a CB virtual entity or attribute (Setup phase).
3. The CB send the subscription information back to the CEP (Setup phase).
4. The device sends data to the IoT Connector using its own format and protocol.

5. The IoT Connector adapt data format and update a virtual entity in the Context Broker using common API.
6. The Broker notifies to the CEP the update.
7. After processing event the CEP, if necessary, update the information on the CB

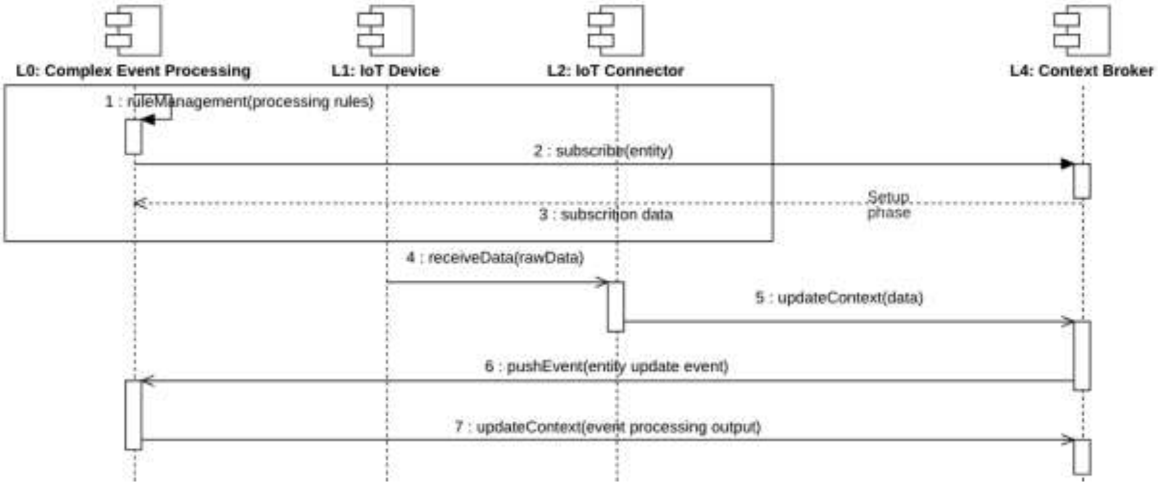

Figure 17: Complex Event Processing flow

5.2 Federated Query Flow

This is the scenario that occurs when Data Analytic Dashboard or Toolbox need to get data from entities in the UKB represented using heterogeneous data models. Federate Query Processing (FQP) is invoked with the following flow:

1. The Client submits a SPARQL query specifying entities and filter conditions
2. The FQP rewrites the input query into sub-queries that can be answered by the fragments of the UKB.
3. The FQP forwards the first sub-query to the Semantic Adapter.
4. The Semantic Adapter executed the query against the fragment of the UKB.
5. The Semantic Adapter receives the result for the first sub-query
6. If the source was a non-semantic source, the Semantic Adapter semantifies the sub-query result (e.g., in the case of MongoDB. in the case of Virtuoso or Fuseki, the semantics are preserved).
7. The FQP receives the semantified sub-query result.

8 - 12 The same as 3 - 7 for the other sub-queries

13. The FQP combines the partial results from the sub-queries into the final query result
14. Finally, the Client receives the requested data

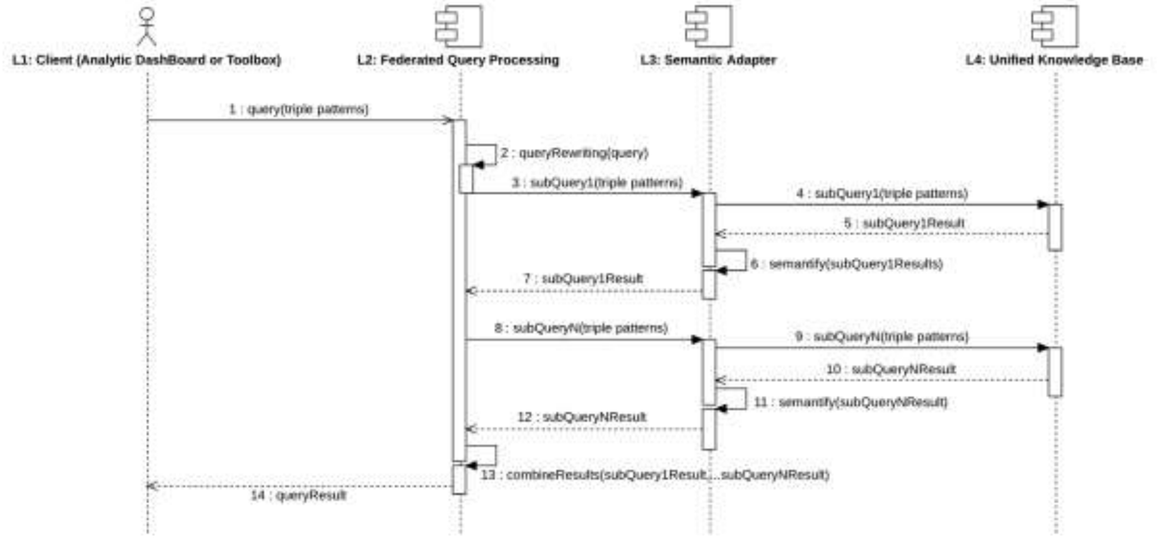
L1: Client (Analytic DashBoard or Toolbox)
L2: Federated Query Processing
L3: Semantic Adapter
L4: Unified Knowledge Base
1 : query(triple patterns)
2 : queryRewriting(query)
3 : subQuery1(triple patterns)
4 : subQuery1(triple patterns)
5 : subQuery1Result
6 : semantify(subQuery1Results)
7 : subQuery1Result
8 : subQueryN(triple patterns)
9 : subQueryN(triple patterns)
10 : subQueryNResult
11 : semantify(subQueryNResult)
12 : subQueryNResult
13 : combineResults(subQuery1Result,..subQueryNResult)
14 : queryResult

Figure 18 : Federated Query Flow

5.3 Data Analytics Tools flows

This is the scenario where raw data is sent via API from the UKB/specific Database to a certain Data Analytics tool where the data is processed and the processed data is sent back to the UKB/specific database and to the Data Analytics Dashboard.

1. Raw data is from the UKB/specific Database to a certain Data Analytics tool. The data is sent through the API defined in D2.2 and according to the Data Model defined in D2.3.
2. The Data Analytics tools runs the code and call the required processing tools (Python, SPARK, etc.) through the specific API as per defined within the Data Analytics Tool Container as per defined in D4.1.
3. Processed data is sent back from the Data Analytics tool to the UKB/specific Database for storing the results. The data is sent again through the API defined in D2.2 and according to the Data Model defined in D2.3.
4. Processed data is sent from the Data Analytics tool to the Data Analytics Dashboard for visualising the results. The data is sent again through the API defined in D2.2 and according to the Data Model defined in D2.3.

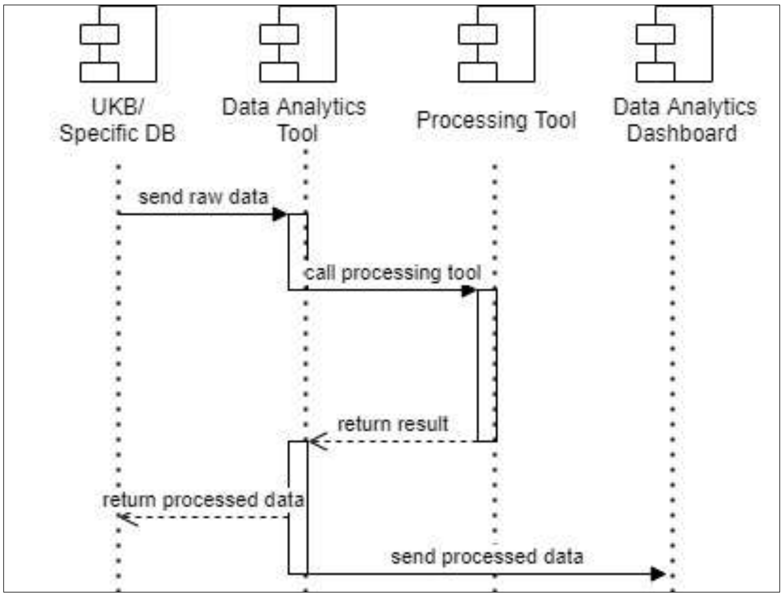

Figure 19 Data Analytics Tool Data flow for the On-premise implementation

This flow represents the case where the Data Analytics tool is implemented in the customer infrastructure (also called “on-premise” approach in task T4.1 where there is no data exchange between different companies. In the scenario where the data analytics tool is implemented as a micro-service a Data Connector with IDS capabilities (i.e. an IDS connector) will need to be implemented between the UKB/specific Database and the corresponding Data Analytics tool, as per explained in D4.1 (16; 11; 11). In this case the data is sent again through the API defined in D2.2 (8) and according to the Data Model defined in D2.3 (5). Within the IDS connector a Data App will be implemented to translate from the API defined in D2.2 to the IDS protocol and vice versa. The other difference is that as the Data Analytics Dashboard will most likely be implemented in the customer infrastructure the data will be first stored in the UKB/Specific DB and from there it will be sent to the Data Analytics Dashboard for visualisation. This is shown in the figure below:

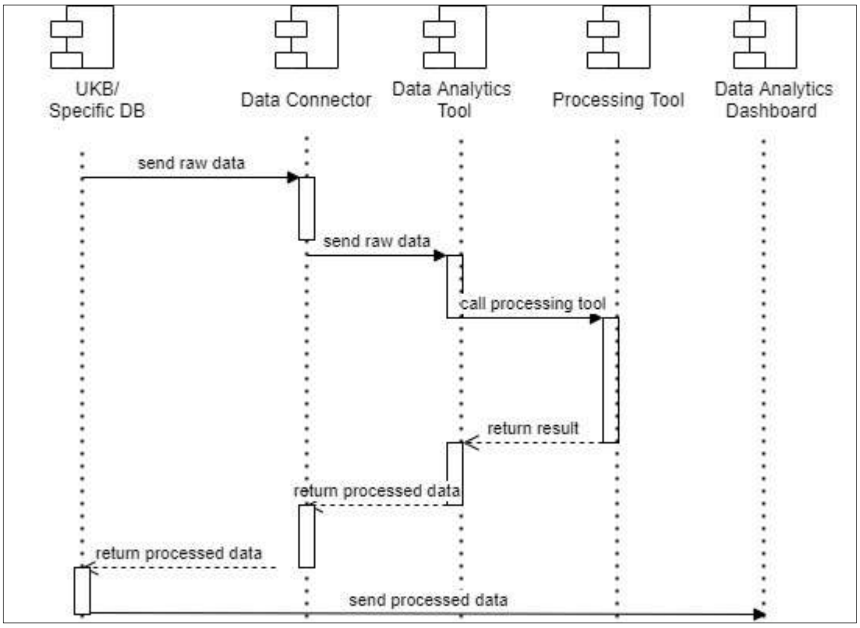

Figure 20: Data Analytics Tool Data flow for the micro-service implementation

5.4 Security Flows

This section describes the data flows that can occur between several components of the PLATOON architecture and the ones composing the Security and Privacy layer.

As common guideline all following flows will ensure that each communication between PLATOON Architecture components will take place in a secure and trustworthy way. Firstly, this involves encryption of communication channels through Transport Layer Security and HTTPS protocols, relying on individual X.509 certificates and PKI (Public Key Infrastructure) provided by the Certification and Provisioning Component. For simplicity, details related to this aspect will be omitted in the interactions depicted in following diagrams.

Secondly, the need of authenticating end users and authorizing a delegated application to act on their behalf and to access protected resources is accomplished by the OAuth2 framework, to which the Security and Privacy component relies on. Nevertheless, also interactions between architectural components will be regulated by a dedicated client authentication and authorization flow, in which each component, registered as a client application, will have its own credentials and relative access tokens regulating inter-component communication itself.

The Security layer will provide a two-levels access-token authorization enforcement:

- Access token associated to End User authenticated identity, used for authorizing data request for user's resources.
- Access token associated to internal application/component identity, used for authorizing access to component services/resources (e.g. exposed APIs).

Finally, Data Access and Usage Control and Consent flows will perform further authorization enforcements of data usage policies, based on either Role Based or Attribute based Access Control models, and eventually data privacy enforcement according to a managed Consent and the GDPR.

5.4.1 End User Authentication and Authorization flow

This scenario gives a high-level overview of the component interactions during end users authentication and authorization. End Users will be able to authenticate and at the same time to authorize an intermediate platform component (e.g. Data Analytics Dashboard) to access their own protected resources, according to the OAuth2 authorization flow (e.g. Authorization Code grant). The access token resulting from authorization flow will be used in the next data requests and validated by the Identity and Access Management (IAM), in order to grant access to protected resource owner's data.

Following steps describe the diagram depicted in the Figure:

1. The authentication flow is activated by each component (e.g. Data Analytics Dashboard) accessed by an end-user of the Platoon platform through an User Agent.
2. This component redirects the request to the IAM component, in order to display the form to start the login process.
3. The User (acting on User Agent) submits the login form filled with its own credentials.
4. The IAM validates the user credentials, in addition:
    a. In case of identity federation the IAM is in charge of mediating the entire process.
    b. If multi factor authentication is enabled, the OTP process will start.

5. If validation is successful, the authorization code is provided and used to retrieve the access token signed by the Certification and Provision Services component.
6. The component can then retrieve end user info by using received access token and eventually (not shown in the diagram) access further resources owned by the authenticated user (if the request will pass further validation made by the Access and Data Usage Control and Consent flow).

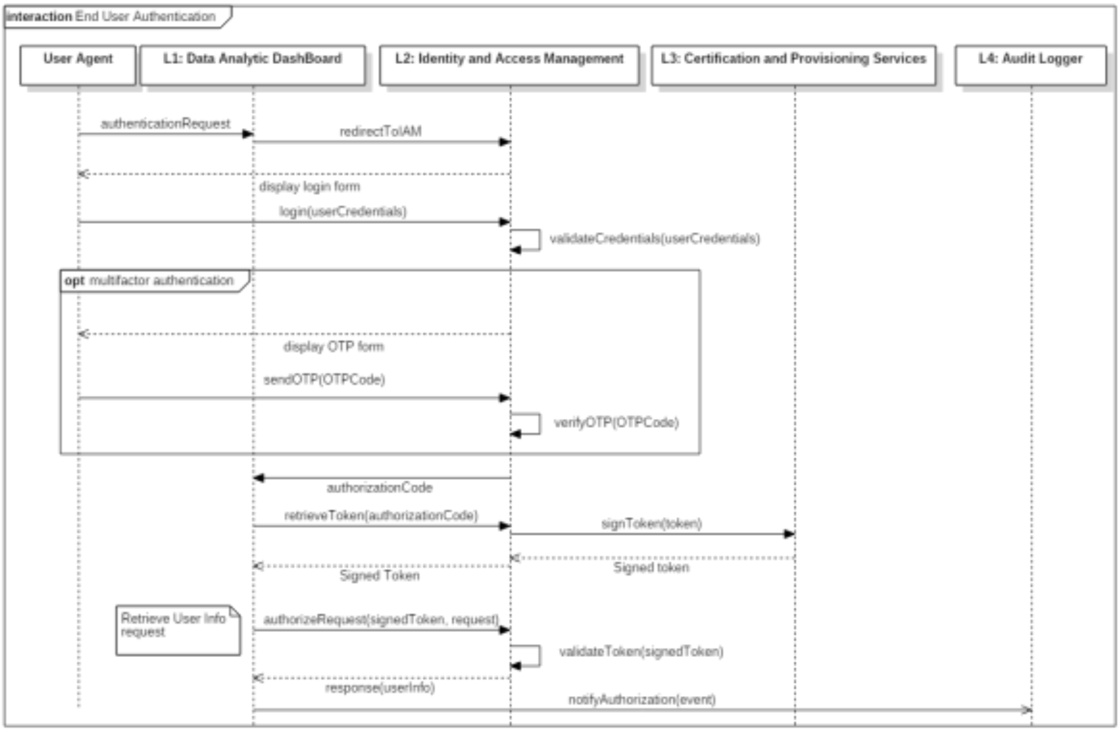
interaction End User Authentication
User Agent
L1: Data Analytic DashBoard
L2: Identity and Access Management
L3: Certification and Provisioning Services
L4: Audit Logger
authenticationRequest
redirectToIAM
display login form
login(userCredentials)
validateCredentials(userCredentials)
opt multifactor authentication
display OTP form
sendOTP(OTPCode)
verifyOTP(OTPCode)
authorizationCode
retrieveToken(authorizationCode)
signToken(token)
Signed Token
Signed token
Retrieve User Info request
authorizeRequest(signedToken, request)
validateToken(signedToken)
response(userInfo)
notifyAuthorization(event)

Figure 21: End User Authentication and Authorization flow

5.4.2 Client Authentication and Authorization flow

This flow deals with the common issue of authenticating applications or components identities and thus authorizing them to access protected services (e.g. an API exposed by the Context Broker), in the context of Machine to Machine (M2M) communications: the system authorizes applications rather than users (as described in the previous End user flow).

To clarify, in addition to any exchange initiated from any internal component in lower layers of Platoon architecture (e.g. IoT Connector pushing data to Context Broker), an inter-component M2M communication can be triggered even starting from an user initiated flow (e.g. Data Analytics Dashboard requesting user's data to underlying components).

In both cases, the Identity and Access Management (IAM) will act as an Authorization Server: relying on Client Credentials flow from OAuth2 framework, it will authorize a requesting component, registered as client application, to access a resource or API provided by a target component. Therefore, requesting component must attach a valid access token, generated after authenticating its own credentials, to access requested resources.

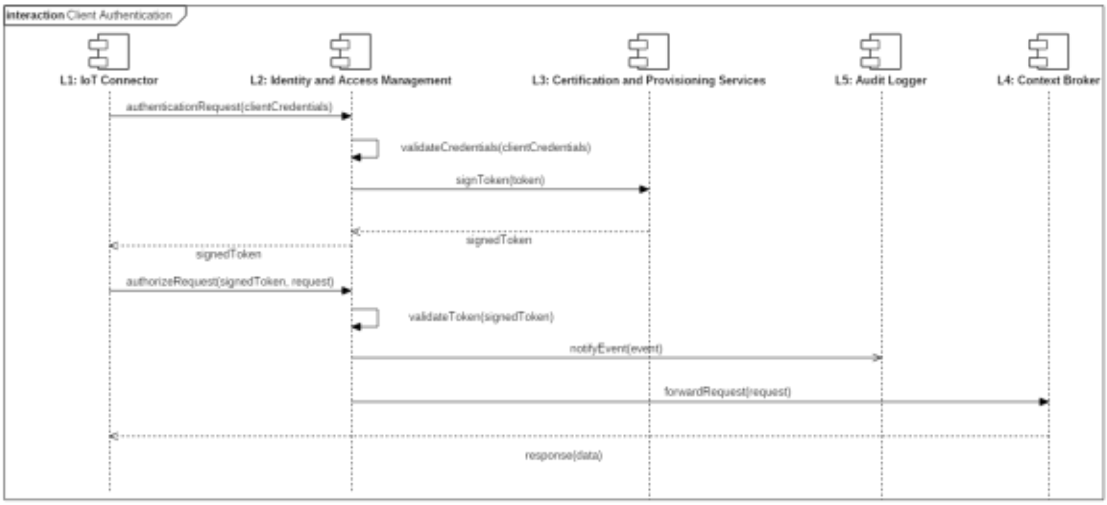
interaction Client Authentication
L1: IoT Connector
L2: Identity and Access Management
L3: Certification and Provisioning Services
L5: Audit Logger
L4: Context Broker
authenticationRequest(clientCredentials)
validateCredentials(clientCredentials)
signToken(token)
signedToken
signedToken
authorizeRequest(signedToken, request)
validateToken(signedToken)
notifyEvent(event)
forwardRequest(request)
response(data)

Figure 22: Client Authentication and Authorization flow

Following steps describe the diagram depicted in the Figure:

1. A requesting connector (e.g. IoT Connector), in order to be authorized to send a data request to a destination component (e.g. Context Broker), sends an authorization request issuing its own client credentials to the Identity and Access Management (IAM).
2. The IAM validates client credentials, looking for a match with any registered application; then, it generates and signs a new access token through the Certification and Provisioning Service, which is returned to the requesting component.
3. Requesting component sends the received access token along the initial data request, in order to be authorized and forwarded to the destination component.
4. The IAM validates input token, notifies the event to the Audit Logger and in case of successful validation grants the access, by forwarding the request to the target component.
5. Finally, the destination component sends the response directly to the requesting one.

5.4.3 Access and Usage Control flow

This flow provides a high-level overview of the components involved in the access and usage control of exchanged information between different organizations (inter organization flow). In general, this exchange will be mediated by the data connector component of the data requestor and data provider. All the interactions among internal component (intra organizational interactions) will be mediated by the client authorization flow (see 5.4.2).

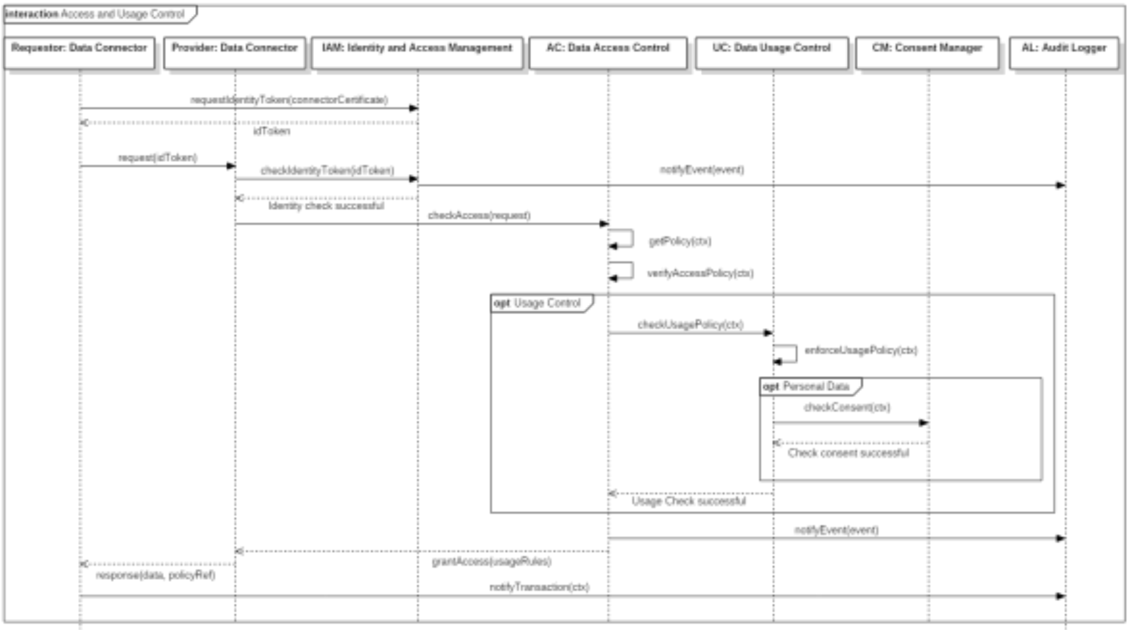

Figure 23: Access and Usage Control flow

1. Data request is mediated by the requestor data connector. An identity token (idToken) is requested through the Identity and Access Management (IAM), presenting the connector’s X.509 certificate.
2. The requestor data connector sends the returned idToken along the request sent to the provider data connector, which in turn verifies it before to start the access grant process. The identity verification is notified and registered by the Audit logger component.
3. The provider data connector invokes the Data Access Control which in turn enforces the access policy according to the context (ctx) defined by the specific request.
4. The request is verified against the identified policy for granting permission to access the requested resources.
5. If usage control has to be performed, the Data Access Control invokes the Data Usage Control, to perform enforcement of specific restrictions regulating the data processing obligations; namely, what may be done with the requested resources and what not.
6. If the requested resources refer to personal data, the Data Usage Control interacts with the Consent Manager to check the legal basis (Consent) on the usage of requested resources.
7. Finally, the Data Access Control grants the access, providing any identified usage obligation and restriction (usageRules) to the provider data connector, which in turn gives the response to the requestor party by attaching also the reference to the data usage policy information.

5.4.4 Consent flow

This flow provides a high-level overview of the components involved in the user centric management of personal data usage Consents. The end user privacy is ensured by enabling him to grant and withdraw consents to third parties for accessing and processing his own data.

The modification of consent modifies the usage rules that can be enforced in the access and usage control flow.

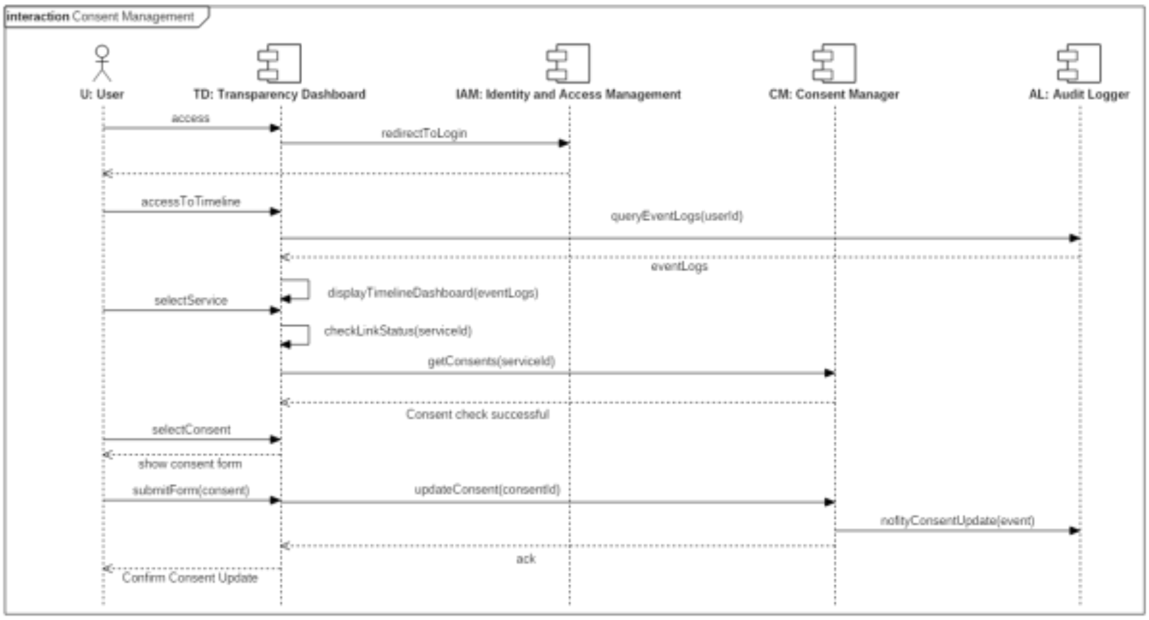

Figure 24: Consent flow

1. The user, namely each owner of personal data processed for a specific purpose, accesses to his/her Transparency Dashboard, to check usage events (provided by Audit Logger component) of personal data according to the given consents. Login is managed by Identity and Access Management, according to the End user authentication flow (see 5.4.1).
2. The user selects a specific service to control personal data processing. The transparency tool retrieves and checks the service link record between the service and the authenticated user and it retrieves all the consents related to that service.
3. The user selects and checks a specific consent and wants to modify it. The dashboard displays the consent form and the user either performs some opt-in/opt-out on each personal data and related usage specifications or withdraws the entire consent.
4. The Transparency Dashboard invokes the Consent Manager component to modify the status of the selected consent which in turn performs the requested modification and notifies the event to Audit logger. The modification and related update event can be visualized by the user by means of the Transparency Dashboard.

5.5 Inter Organizational Data Exchange Flow

The scenario depicted in Figure 25 shows how dataset are exchanged between two organisation using IDS based connector components. In the PLATOON customisation the IDS protocol is enhanced adding the personal data verification performed by the Consent Manager in the cases where are involved personal data.

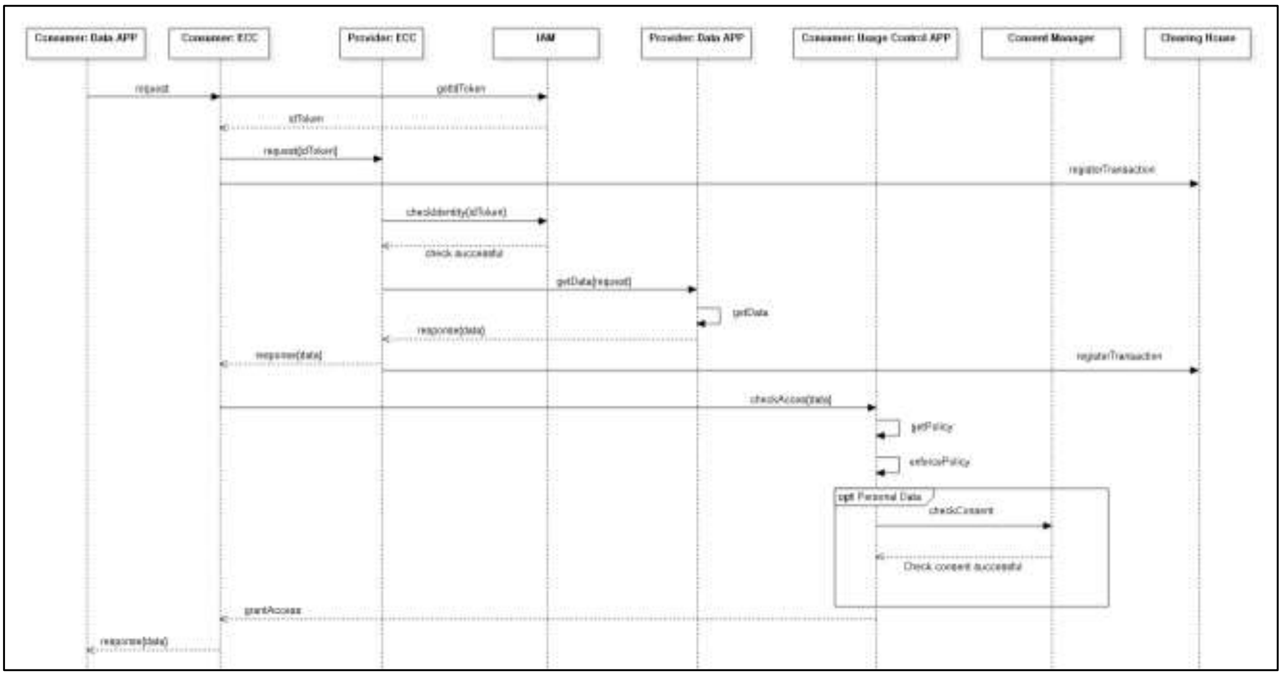

Figure 25: Inter organizational data exchange flow

1. When the Consumer Data APP receives a request to its endpoint forwards it to the Consumer ECC (Execution Core Container)
2. ECC requests an identity token (idToken) through the IAM (Identity and Access Management)
3. The Consumer ECC sends the returned idToken along the request sent to the Provider ECC
4. The Consumer ECC register the request transaction on the Clearing House
5. The Provider ECC check the idToken against the IAM
6. If the IdToken is successful verified the Provider ECC send the request to the Provider Data APP
7. Provider Data APP gets the data
8. Provider Data APP sends the response to the Provider ECC with data
9. Provider ECC sends response to the Consumer ECC
10. Provider ECC registers the transaction on the Clearing House
11. After Consumer ECC received the data check data access invoking Consumer Usage Control
12. Consumer Usage Control gets and enforces usage policy and if the data includes personal data invokes the Consent Manager to check the consent.
13. Consumer Usage Control sends the access grant to the Consumer ECC
14. Finally, Consumer ECC sends data to the Consumer Data APP to satisfy the initial request.

6 Alignment with relevant European initiatives

In the design process of the PLATOON a specific attention was paid in the analysis and relation with existing European projects. Three main input initiatives have been analysed: the International Data Spaces (IDS) whose functionalities have been fully adapted and adopted by PLATOON , the FIWARE framework and the COSMAG architecture. In this section will present the relations between the PLATOON architecture and FIWARE /COSMAG assets and their potential exploitation in the project activities, while the IDS related component, that are fully part of the PLATOON architecture have been described in section 3.5.1.

6.1 FIWARE

FIWARE is a framework composed of several open source components, namely Generic Enabler (GE), which aims to ease the development of Smart Solution for different domains such as Smart Cities, Smart Industry, Smart Agrifood and Smart Energy.

The core of FIWARE is the Orion Context Broker that enables to manage context information in a highly decentralized and large-scale manner. The current version of Orion supports FIWARE NGSI API specifications while the new under development release will expose the new ETSI NGSI-LD standard (17). The other FIWARE components fit in different categories, such as IoT and Robots interaction, Context Data management, publication, and monetization, but also Processing, analysis, and visualization of context information

PLATOON reference architecture includes some logical components that could easily implemented using FIWARE Generic Enablers. Below, it is described how some of the FIWARE components can be related to the main components of the PLATOON reference architecture, providing a suitable reference implementation of them.

IoT Connector and Data Connector

The IoT connector and the Data Connector in the PLATOON reference architecture are separated logical components because they refer respectively to real-time and batch/static data flows, but they perform a very similar task. FIWARE Agents are committed to perform both tasks in the FIWARE approach.

A FIWARE IoT Agent is a component that allow to a group of devices to send their data to a Context Broker using their own native protocols dealing also with security aspects. This brings a standard interface to all IoT interactions at the context information management level. Each group of devices can use their own proprietary protocols and disparate transport mechanisms. FIWARE provides a set of agents supporting different IoT protocols/data format also relevant for the Energy Domain (e.g. OPC-UA, LoRaWAN, LWM2M etc)

Context Broker and Complex Event Processing

Context Broker (CB) and Complex Event Processing (CEP) components of the PLATOON reference architecture perfectly match the corresponding components of the FIWARE framework. The CB is responsible for managing context data from devices and systems by providing a uniform interface to the south and north bounds. The CEP works together with the CB by intercepting update events and inferring an additional level of compounded events over time.

Data Analytics Dashboard

FIWARE provides GE related to data analytics that could implement the Data Analytics Dashboard: for instance, Knowage that is an open source suite for business analytics over traditional sources and big data systems is one of the reference implementations of the **Data Visualization GE**.

Knowage supports visualization of the data analytics, providing new self-service capabilities that give autonomy enabling the end-user to build his own analysis and explore his own data space, also combining data that come from different sources.

Unified Knowledge Base

In the FIWARE ecosystem there are several components designed to achieve a level of persistence and historicization of data. There are dedicated and extensible components to historicize data as it is updated in the Context Broker. There are other persistence components including Open Data, Big Data, and NoSql. By combining these tools in different ways, it is possible to develop a FIWARE based Unified Knowledge Base of PLATOON architecture

The following table shows the correspondence between some logical components in the PLATOON reference architecture and some concrete FIWARE components that can implement them entirely or in part. This has to be considered only a tentative list, that can be further extended:

PLATOON Components	FIWARE Generic Enablers
IoT Connector, Semantic Adapter	IoT Agents
Context Broker	Orion Context Broker
Complex Event Processing	Perseo
Data Analytics Dashboard	Knowage, Wirecloud
Unified Knowledge Base	STH-Comet, Cygnus, Draco, Cosmos, QuantumLeap, Idra

Table 1 - PLATOON and FIWARE components mapping

6.2 COSMAG

Comprehensive Architecture for Smart Grid (COSMAG) refers to the analysis and collection of specifications to define possible data exchange scenarios and data format in the Smart Energy sector. Following are highlighted some key aspects presented in COSMAG that can be directly related and addressed by PLATOON reference architecture:

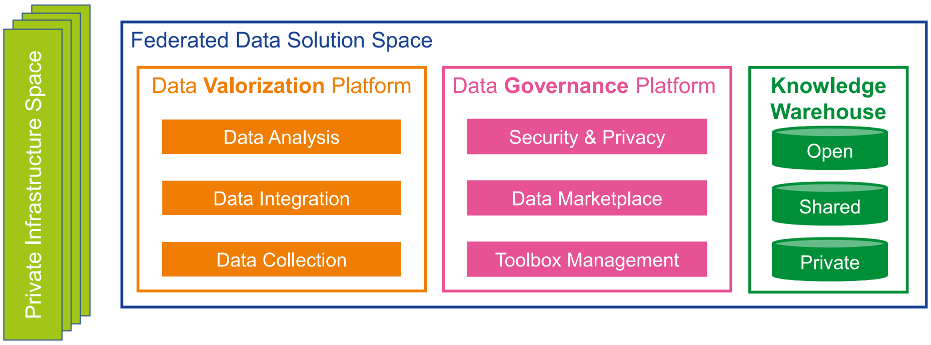

Figure 26: COSMAG Federated Data Solution Space (3)

- COSMAG introduces the concept of Federated data solution Space in which the logical elements of the three main pillars (Data Valorisation Platform, Data governance Platform and Knowledge Warehouse) are completely covered by the by the components of the PLATOON reference architecture
- Key elements identified by COSMAG as enablers of data economy includes the concept of Semantic Interoperability, Context Management, data Sovereignty and Open API, topics that are addressed by PLATOON architecture
- Suggested technical solutions, such as FIWARE Context Broker and other Generic Enablers, as presented in the previous section, are perfectly suitable to implement the PLATOON architecture
- The adoption of the ETSI CIM (Common Information Model) standard and SAREF for the identification of ontologies in Energy systems is recommended by COSMAG. The PLATOON semantic data models and common vocabularies reuses CIM and SAREF as baseline that has been extended to fit project pilot requirements (5)

Conclusion

Deliverable D2.5 reported the specification of PLATOON reference architecture. The document described in detail the components of the architecture their functionalities, interfaces and relationships. It has been provided both static and dynamic views of the architecture, highlighting the requirements addressed by the different components. Moreover, the deliverable included references to main European initiatives that influenced the architectural design and that can also provide building blocks to create a reference implementation of PLATOON platform.

The scope of this document was twofold: on one side it has to be considered high level specification to guide the detailed design of the specific components their implementation and deployment to be addressed by the other PLATOON technical tasks (see section 2).
On the other side the reference architecture is also a tool to help the project Pilots, but also any organisation/third party partner, to be integrated in the PLATOON ecosystem, identifying, for instance, the main interoperability points for the interaction of the Energy systems and infrastructures. The PLATOON architecture, therefore, should be considered a key asset to facilitate the platform interoperability in the Energy domain, that defines the main elements and processes to allow communication with legacy systems, semantic interoperability and provisioning of big data analytics services. The logical and agnostic approach of the architecture leaves to the specific developers and Pilot owners to adopt the most suitable technologies to implement the needed PLATOON components to fit their specific requirements.

This deliverable represents the second version of the architecture that has been updated taking in consideration the work carried in the implementation tasks and in particular the pilot deployments. In particular, two aspects have been updated: the inclusion in the architecture of the IDS Connector components, that have been adopted by the project pilots for the secure exchange of data, between different organisation, and the edge-cloud framework that detailed the approach of concrete instantiation of the PLATOON components at edge and cloud level.

References

[1] IDS, «International Data Space,» [Online]. Available: https://www.internationaldataspaces.org/. [Consultato il giorno 3 2020].

[2] FIWARE Foundation, «FIWARE Platform,» [Online]. Available: https://www.fiware.org/. [Consultato il giorno 2020].

[3] COSMAG, « Comprehensive Architecture for Smart Grid,» 2019. [Online]. Available: https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=57632 .

[4] CEN-CENELEC-ETSI, «Smart Grid Reference Architecture (SGAM),» 2012. [Online]. Available: https://ec.europa.eu/energy/sites/ener/files/documents/xpert_group1_reference_architecture.pdf.

[5] PLATOON, «D2.3 - Common Data Model for Energy,» 2020.

[6] ETSI, «Smart Applications REFerence Ontology,» [Online]. Available: https://saref.etsi.org/. [Consultato il giorno 2020].

[7] IEC, «IEC Standards,» 2020. [Online]. Available: https://www.iec.ch/technical-committees-and-subcommittees#tclist.

[8] PLATOON, «D2.2 - Open API Specification,» 2020.

[9] PLATOON, «D1.1 - Business case definition, requirements and KPIs,» 2020.

[10] EU, «GDPR,» 2016. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679.

[11] PLATOON, «D3.2 - Data governance framework and security model».

[12] IETF, «The OAuth 2.0 Authorization Framework,» [Online]. Available: https://tools.ietf.org/html/rfc6749.

[13] ITU, «X.509 certificates,» 2008. [Online]. Available: https://www.ietf.org/rfc/rfc5280.txt.

[14] OASIS, «eXtensible Access Control Markup Language (XACML) Version 3.0,» 2013. [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf.

[15] IETF - ODRL, 2018. [Online].

[16] IDS, «IDS Reference Architecture Model v3.0,» [Online]. Available: https://www.internationaldataspaces.org/wp-content/uploads/2019/03/IDS-Reference-Architecture-Model-3.0.pdf.

[17] PLATOON, «D4.1 - Analytical Toolbox Design,» 2020.

[18] ETSI, «Context Information Management (CIM),» 2019. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_CIM009v010101p.pdf.

[19] Á. P. A. C. J. M. D. l. V. F. &. H. J. J. Alonso, «Industrial data space architecture implementation using FIWARE,» *Sensors,* p. 2226, 7 2018.

Annex I – PLATOON Requirements

Requirement ID	Description	Requirement Group	Requirement type
13001	As a PLATOON platform user, I want the PLATOON Reference Architecture to allow integration of already existing solutions that I have and my legacy systems so that I can avoid rework and I can concentrate on value added tasks.	Platform-Reference Architecture	Functional
13002	As a PLATOON platform user, I want the PLATOON reference architecture to be technologically agnostic to avoid vendor lock-in.	Platform-Reference Architecture	Non-Functional
13003	As a PLATOON solution provider, I want the PLATOON reference architecture to reuse already available open source solutions and only create or improve those aspects that are not covered by the existing solutions o that I can avoid rework and concentrate on value added tasks.	Platform-Reference Architecture	Non-Functional
13004	As a PLATOON project partner, I want the PLATOON reference architecture to be compliant with the COSMAG reference architecture so that I comply with what it was asked in the call and what it was stated in the DoA. In this case being compliant means that the developed reference architecture can be mapped with the Federated Data Solution Space and that it takes into consideration the conclusions and recommendations defined in the COSMAG architecture document.	Platform-Reference Architecture	Non-Functional
13005	As a PLATOON solution provider, I want the PLATOON reference architecture to be compatible with FIWARE to enhance its adoption after project completion. In this case being compatible with FIWARE means that the developed reference architecture must allow the integration with the API used by the FIWARE Context Broker.	Platform-Reference Architecture	Functional
13006	As a PLATOON platform user, I want the PLATOON reference architecture to be applicable for the whole energy value chain (i.e. generation, transport/distribution and end use of energy) so that it can be implemented in all the use cases defined in D1.1that comprise the whole energy value chain.	Platform-Reference Architecture	Non-Functional
13007	As a PLATOON platform user, I want the PLATOON reference architecture to be mappable to the different layers of SGAM reference architecture so that it enables the interoperability in the smart grid area.	Platform-Reference Architecture	Non-Functional
13008	As a PLATOON platform user, I want the PLATOON reference architecture to be compatible with IDS reference architecture so that it ensures data sovereignty, security and privacy when sharing my data/tools with other platform users. In order to be compatible with IDS, PLATOON reference architecture must be able to integrate the main mandatory components of IDS reference architecture, i.e. the connector and the identity provider (a.k.a DAPS).	Platform-Reference Architecture	Functional
13009	As a PLATOON solution provider, I want the PLATOON reference architecture to be compatible with the main opensource big data storage solutions (SQL, HDFS, HBase, NoSQL databases...) and batch and real-time processing solutions (e.g. Hadoop V2.0, Spark, Flink, Kafka...) so that I can reuse already existing solutions and focus on value added tasks.	Platform-Reference Architecture	Functional
13010	As a PLATOON solution provider, I want the PLATOON reference architecture to allow some of the components of the reference architecture to be hosted at the component level (e.g. edge computing), on premise and in the cloud so that I can optimise my solutions regarding data exchange, storage and processing.	Platform-Reference Architecture	Functional
13011	As a PLATOON platform user, I want the PLATOON reference architecture to be implementable in the main cloud platform providers (i.e. Amazon Web Services, Microsoft Azure and Google Cloud) so that I can avoid vendor lock-in, enhance its adoption and reduce infrastructure costs.	Platform-Reference Architecture	Functional
13012	As a data/app provider, I want the PLATOON reference architecture to enable the exploitation of digital services (both data and data analytics tools) through the PLATOON Marketplace to enhance the exploitation of my data and tools.	Platform-Reference Architecture	Functional

13014	As a app provider, I want all the pilots belonging the same pilot group (RES generation and in particular wind turbines, smart grids and smart buildings) to use common data models and ontologies so that I can develop common generic tools and be able to validate them in different large scale pilots to prove its generalization capacity. If possible, it would be desirable to define a single PLATOON data model and ontology. However, due to the differences between different sectors and applications it is recognised that is very likely that it might be almost impossible to define such a single ontology for all of them.	Platform-Interoperability	Non-Functional
13015	As an app provider, I want the defined PLATOON data model(s) to be open source so that it enhances the adoption of the developed tools and I can reuse already existing solutions.	Platform-Interoperability	Non-Functional
13016	As a app provider, I want the defined PLATOON ontology(es) to take into account and reuse as much as possible already open source and widely-used ontologies and standards for the three main areas of the energy supply chain under the scope of PLATOON so that it enhances the adoption of developed tools and I can reuse already existing solutions. For instance, SAREF ontology for smart buildings and Core IEC Standards for smart grids. Nevertheless, the existing ontologies in energy sector don't cover all the domains. Thus, the developed PLATOON ontology(es) must extend or/and to create ontological modules to represent other knowledge which is not present in the well-known ontologies	Platform-Interoperability	Non-Functional
13017	As PLATOON platform user, I want the PLATOON Interoperability layer to allow the integration of heterogeneous data that will be used in different pilots. In particular must be able to deal with the following formats as per defined in the pilot requirements in deliverable D1.1: Files (csv, xml, JSON, CAD and jpeg), Logs (csv and txt), SQL and NoSQL (mat and tdms format).	Platform-Interoperability	Functional
13018	As a PLATOON solution provider, I want the PLATOON Interoperability Layer to define common APIs so that the integration between different platforms is replicable and scalable.	Platform-Interoperability	Functional
13019	As a PLATOON solution provider, I want the developed common APIs to be open source so that it enhances its adoption and I can reuse already existing solutions. In this sense the developed PLATOON APIs must consider existing standardisation activities related to Open API and data interoperability, such as NGSI-LD.	Platform-Interoperability	Non-Functional
13020	As PLATOON platform user, I want that a common "Security and Privacy" guideline is defined so I am sure that any platform that forms part of the PLATOON federated platform meets certain minimum requirements regarding security and privacy.	Platform - Security and Privacy	Non-Functional
13021	As a PLATOON platform user, I want the Data Analytics Toolbox to allow the development, implementation and validation of tools required for the different use cases so that I can meet the use case requirements defined in deliverable D1.1.	Platform-Data Analytics Toolbox	Functional
13022	As PLATOON platform user, I want the PLATOON Data Analytics to be able to process big volumes of heterogenous data in batches as well as in real-time so that I can meet use case requirements defined in deliverable D1.1. There are some pilots such as pilot 1a, 3a, 3b and 3c that due to the nature of the use cases it is not required real-time processing, thus, batch or micro-batch processing is enough. However, there are other use cases such as 2a and maybe 2b, that due to the frequency of the data and the type of response that is required, they will need real time processing.	Platform-Data Analytics Toolbox	Functional
13023	As an app provider, I want the PLATOON Data Analytics Toolbox to allow distributed implementation of the different parts of the data analytics tools at different levels of the architecture (i.e. at the edge and on cloud/premise) so that the developed tools make an efficient use of the available storage and processing capability.	Platform-Data Analytics Toolbox	Functional
13024	As an app provider, I want the PLATOON Data Analytics Toolbox to be compatible with the main widespread open-source and free to use big data batch and real time processing frameworks (Hadoop V2.0, Spark, Kafka...) and storage solutions (RDBMS, HDFS, HBase, NoSQL databases..., Cache, time series) so that I can use of already existing and validated libraries (e.g. Spark MLib, Spark ML, etc.) without having to write all the code from scratch. Also, by using these open source solutions I can benefit from a global software development community.	Platform-Data Analytics Toolbox	Functional

13025	As an app provider, I want the PLATOON Data Analytics Toolbox to be compatible with the following programming languages Python, R, C++, Java ,Scala and MATLAB so that I can use the legacy systems and data analytics tools that I already have.	Platform-Data Analytics Toolbox	Functional
13026	As an app provider, I want the PLATOON Data Analytics Toolbox to be compatible with main cloud providers (Amazon Web Services, Microsoft Azure and Google Cloud) so that I can offer my tools to a wide range of app consumers. It must also be compatible with the coming GAIA-X European initiative. It must allow the implementation (at least should have connectors to IaaS part of the clouds) of the tools into these platforms.	Platform-Data Analytics Toolbox	Functional
13027	As an app provider, I want the PLATOON Data Analytics Toolbox to allow a modular implementation of the tools allowing the combination of different generic tools that can be shared amongst use cases (signal processing, outlier removal, visualisation…) and use case specific tools (digital twin, soft sensor…) so that we can use and share already existing generic solutions and focus on developing value adding modules.	Platform-Data Analytics Toolbox	Functional
13028	As an app provider, I want the PLATOON Data Analytics Toolbox to allow a seamless integration with different data sources so that I can easily connect my tools to the data from different data providers so I can build better models and offer my tools to a wide range of consumers. In order to facilitate the integration with the data source, the tools within the PLATOON Data Analytics platform must follow the common ontologies developed for the PLATOON interoperability layer.	Platform-Data Analytics Toolbox	Non-Functional
13029	As an app user, I want the PLATOON Data Analytics Platform to allow the semantification of data producing Knowledge Graphs in RDF Semantic Linked Data format so that I can perform analytics natively on the RDF representations of data.	Platform-Data Analytics Toolbox	Functional
13030	As an app user, I want the PLATOON Data Analytics Toolbox to be compatible with existing widely used open source Data Analytics frameworks (e.g. Databricks, Knime, etc.) that allow the development, implementation and validation of tools that have user friendly interfaces that make it easy to deploy and use the data analytics tools by energy experts without a deep knowledge in coding. It would be desirable that the compatible Data Analytics frameworks allow easy implementation and validation of the data analytics tools following the drag & drop concept similar to the MATLAB Machine Learning suite but based on open-source software and libraries. Additionally, these types of frameworks already have some generic built-in tools that could be used as part of the project such as pre-processing functions to be able to prepare the data before feeding it to the tools (e.g. change parameter types, remove Nan values, remove columns).	Platform-Data Analytics Toolbox	Functional
13031	As an app user, I want the PLATOON Data Analytics Toolbox to allow fine tuning the hyperparameters of the data analytics tool (weights, thresholds…) so that I can adapt the tool for my specific dataset.	Platform-Data Analytics Toolbox	Functional
13032	As an app user, I want the PLATOON Data Analytics Toolbox to allow conducting and tracking different experiments with different hyperparameters so that I can validate the tools before implementing them in the production system.	Platform-Data Analytics Toolbox	Functional
13033	As an app user, I want the PLATOON Data Analytics Toolbox to allow the representation of the results in a user-friendly interface using specific KPIs for the specific application so that energy experts without a deep knowledge in data analytics can understand them.	Platform-Data Analytics Toolbox	Functional
13034	As an app provider, I want the PLATOON Data Analytics Toolbox to allow the implementation of the tools using containers and/or microservice concept so that they can be easily implemented according to the required specific business model (e.g. pay per license / pay per use).	Platform-Data Analytics Toolbox	Functional
13035	As an app user, I want that the PLATOON Data Analytics Toolbox to enable the automatic productification of tools following the DevOps principles so the deployment process is faster and more efficient.	Platform-Data Analytics Toolbox	Functional
13036	As an app provider, I want the PLATOON Data Analytics toolbox to have version control using solutions like GIT or similar so that I can track code changes in my tools.	Platform-Data Analytics Toolbox	Functional

13037	As an app user, I want the PLATOON Data Analytics Toolbox to have monitoring tools that automatically generate internal alarms when the tools are down or they have been receiving bad data for a long time so that I can quickly identify issues with the tools or with the data quality and quickly solve them.	Platform-Data Analytics Toolbox	Functional
13038	As a PLATOON partner, I want the developed Edge Computing solutions to be based on industry protocols (e.g. Modbus, BACnet, etc.) so that I can implement them in the different use cases defined in D1.1	Platform-Edge Computing	Functional
13039	As an app provider, I want the developed Edge Computing solutions to allow to execute locally microservices and applications so that it enables distributed and autonomous data processing. Depending on the use case, applications might include data cleaning, data compression, feature extraction, event detection with threshold-based notifications and alarms, analytics algorithms, remote monitoring and diagnostics, and any other custom software.	Platform-Edge Computing	Functional
13040	As a data/app provider, I want the developed Edge Computing solutions to allow local data storage so that allows autonomous operation and no data is lost in case it is disconnected from the network.	Platform-Edge Computing	Functional
13041	As a PLATOON platform user, I want the developed Edge Computing solutions to allow for permission-based access control, secure encrypted communication, certificate management and integration into existing security solutions so that the security of my system is not compromised by the installation of edge computing devices.	Platform-Edge Computing	Security
13042	As a data provider, I want the developed Edge Computing solutions to allow the implementation of anonymisation solutions so that I can decouple the device from the owner’s identity and comply with legal and ethics requirements (e.g. GDPR) defined in deliverable D1.5.	Platform-Edge Computing	Privacy
13043	As PLATOON platform user, I want the developed Edge Computing solutions to allow the remote management of individual edge computing instances via run, halt, configure, and update procedures so that I can enable/disable and control the different edge computing devices remotely. It would be desirable to allow orchestration tools that manage and coordinate many edge sites and workloads, potentially leading toward a peering control plane or “self-organizing edge. In this sense the developed solutions must have an open API that allows remote applications to communicate with the edge computing infrastructure via REST, WebSockets, or JSON-RPC. It would also be desirable to allow automated data and workload relocations for load balancing across geographically distributed hardware.	Platform-Edge Computing	Functional
13044	As a PLATOON platform user, I want the developed Edge Computing solutions to allow to virtualize the hardware function using the “software defined hardware" scheme so that I can change the behaviour of the devices updating the deployed software, instead of a more traditional approach that requires the replacement of the embedded firmware. It would be desirable to have automated edge commission/decommission operations, including initial software deployment and upgrades of the resource management system’s components.	Platform-Edge Computing	Functional
13045	As a PLATOON platform user, I want the developed Edge Computing solutions to guarantee the latencies required for each scenario deployed especially among those with low and unreliable bandwidths.	Platform-Edge Computing	Functional
13046	As a PLATOON platform user, I want the developed Edge Computing solutions to have a fault-tolerant design both from a software and hardware perspective so that I can use them for critical applications where no failure is permitted. Nevertheless, it must be highlighted that developing a completely fault tolerant design is a complicated task that requires vast amounts of resources and time. Thus, thus type of development will probably not be possible given the time and resources allocated for the Edge Computing development task for this project.	Platform-Edge Computing	Functional
13047	As an app provider, I want the PLATOON Marketplace to allow to download the code from the marketplace as self-contained integrated solution following the container technology so that I can sell my apps according to the pay per license business model.	Platform-Marketplace	Functional
13048	As an data/app provider, I want the PLATOON Marketplace to ensure that the provided datasets and tools are used by the app consumers under the specific agreed conditions (e.g. cannot be shared with third parties, cannot be exploited with third parties) so that I maintain the IPR of my data/tools.	Platform-Marketplace	Functional

13049	As an app provider, I want the PLATOON Marketplace to allow the implementation and exploitation of tools as a microservice so that so that I can sell my apps according to the pay per use business model.	Platform-Marketplace	Functional
13050	As a data/app provider, I want the PLATOON Marketplace to ensure that the tools that have been trained with the data owner data can only be used under specific conditions agreed between the data owner and app provider (e.g. it can only be used with data owner, it can be shared with others without showing the raw data and paying a fee to the data owner, etc.) so that I maintain the IPR of my data/tools.	Platform-Marketplace	Functional
13051	As a data/app consumer, I want the PLATOON Marketplace to centralize all the different providers in one single one-stop shop including a complete and easy to find catalogue of all of the products and services offered by the different providers so that I can compare the different services offered by the different providers and choose the alternative that suits best my requirements.	Platform-Marketplace	Functional
13052	As a data/app consumer, I want the PLATOON Marketplace to contain a detailed description of the offered datasets and tools (metadata) so that I can better understand the applicability and limitations of the of the offered datasets and tools.	Platform-Marketplace	Functional
13053	As a data/app consumer, I want that all the products and services offered in the PLATOON Marketplace pass a quality and authenticity check so that I can be sure that the products and services meet the specifications defined in the description and that including them in my business will bring a benefit.	Platform-Marketplace	Process
13054	As a data/app provider, I want the PLATOON Marketplace to be easily findable and accessible through the main media channels (magazines, social media, etc.) so that I can reach as much potential consumers as possible.	Platform-Marketplace	Marketing
13055	As a data/app provider, I want the PLATOON Marketplace to allow defining usage policies for data and app services and publish metadata including these usage policies directly on the marketplace so that the whole exploitation process is much faster and efficient.	Platform-Marketplace	Functional
13056	As a data/app provider, I want the PLATOON Marketplace to allow automatic transfer of data and tools with the associated usage constraints as soon as the product or service has been purchased on the marketplace so that the whole exploitation process is much faster and efficient.	Platform-Marketplace	Functional
13057	As a data/app provider, I want the PLATOON Marketplace to allow tracking information about data and apps transaction automatically so that it facilitates the billing and auditing process.	Platform-Marketplace	Functional
13058	As PLATOON solution provider, I want the PLATOON marketplace to be based on existing open source solutions so that I avoid rework and I can focus on value added tasks.	Platform-Marketplace	Non-Functional
13059	As a PLATOON platform user, I want the PLATOON marketplace to follow a decentralized approach in line with the PLATOON federated platform concept so that I can avoid having a centralized repository and a company that will play the role of the marketplace administrator reducing the exploitation costs beyond the project lifetime. In this sense, the PLATOON marketplace should follow a similar approach to the CKAN open source solution but allowing the exploitation of tools as well as data. Apart from CKAN, there are other already available marketplace solutions that have been created within the scope of different initiatives that should be considered.	Platform-Marketplace	Functional
13060	As a data/app provider, I want the PLATOON marketplace to have an external interface where external organizations that are not part of the PLATOON federated platform can see and purchase the different available services so that I can reach a wider public of potential consumers.	Platform-Marketplace	Functional
13061	As a data/app provider/consumer, I want that all the companies that can offer and purchase services through the PLATOON marketplace follow a checking process that I can be sure that they meet certain minimum requirements regarding security, privacy and sovereignty.	Platform-Marketplace	Process
13062	As a data/app provider/consumer, I want that the PLATOON Marketplace to be compatible with the IDS reference architecture so that I can access an ecosystem of trusted companies that meet certain minimum requirements regarding security, privacy and sovereignty. In addition, this will avoid having a specific	Platform-Marketplace	Process

	certification authority as the the access rights of companies will be granted by the IDS certification authority.		
13063	As a data/app provider/consumer, I want the PLATOON Marketplace to have a payment gateway so that the paying process is faster and more efficient.	Platform-Marketplace	Functional
13064	As a data/app provider/consumer, I want the PLATOON Marketplace to allow a negotiation step so that a consumer can identify datasets or apps then request a price from the provider and the provider can then assign a specific price depending on the consumer (using 'unit prices' as a base pricing scheme).	Platform-Marketplace	Functional
13065	As a data/app provider/consumer, I want that the PLATOON Marketplace to implement generic data security mechanisms (secure communication, identity management), data access (authorization management) control and data usage control aspects so that the security of my system is not compromised.	Platform-Marketplace	Security
13066	As a data/app provider/consumer, I want that the PLATOON Marketplace to be compliant with GDPR so that I avoid data privacy issues with any personal data that might be stored, provided or shared through the PLATOON Marketplace (e.g. personal contact details, bank account details, etc.).	Platform-Marketplace	Privacy
13067	As a PLATOON project partner, I want that external companies including the selected companies for open calls follow the PLATOON reference architecture and the common data models, ontologies and APIs defined so that I can share data with them and I can use the tools developed by the external companies.	Platform-Collaboration	Non-Functional
13068	As a PLATOON project partner, I want that external companies including the selected companies for open calls follow the security, privacy and sovereignty solutions defined in WP3 so that I can ensure that the platforms from the external company does not suppose a cybersecurity risk to the rest of the platforms that form the PLATOON federated platform and that I can be sure that when I exchange data with these external companies the data security, privacy and sovereignty requirements are maintained.	Platform-Collaboration	Security and Privacy
13069	As a PLATOON project partner, I want that external companies including the selected companies for open calls follow the Data Analytics Toolbox and Edge Computing solutions defined in in tasks T4.1 and T4.2, respectively, so that they meet the corresponding requirements defined in this document.	Platform-Collaboration	Non-Functional
12001	As a Platoon Vocabulary Provider, I want to create Platoon-energy related vocabularies so that I can contribute to open communication and integration.	Data Exchange / Security	Functional
12002	As a Platoon Data Consumer, I want to select a specific vocabulary from a Platoon Vocabulary Hub so that I can correctly interpret the data.	Data Exchange / Security	Functional
12003	As a Platoon Data Provider, I want to describe the dataset properties (e.g. data format, date and time of creation, dataset owner, metadata, etc.) and register the metadata at the Broker Service Provider, so that my dataset can be found through searches on its characteristics/metadata.	Data Exchange / Security	Functional
12004	As a Platoon Data Provider, I want to define a comprehensive pricing model for my datasets, so that I can establish different price types (e.g. pay per transfer, pay for access per day/month/year, etc.) so I can generate extra revenues.	Data Exchange / Security	Functional
12005	As a Platoon MarketPlace Operator, I want to be able to account usage of transferred and received data, so that I can perform clearing and settlement service duties for all financial and data exchange transactions.	Data Exchange / Security	Functional
12006	As a Platoon Data Provider/Consumer, I want to log data transaction details in the Clearing house and receive reports and statistics regarding transferred/received data usage, so that I can receive information about billing, correct use of datasets, demand and supply studies, pricing, be able to resolve conflicts, etc.	Data Exchange / Security	Functional
12007	As a Platoon Data Owner, I want to define a Data Usage Policy so that I can retain management rights, define rules and conditions on how data must be used by Data Consumers (e.g. who can see my data and which parts, prohibit forwarding to 3rd parties and other participants, merging data, the use that can be given to my data, etc.)	Data Exchange / Security	Functional
12008	As a Data User/Administrator, I need a data management framework so that I can easily integrate and process data with different time resolutions.	Data Exchange / Security	Functional

12009	As a Platoon Data Provider/Consumer, I want to have an IDS-compliant Connector so that I can manage data and participate in the data exchange process as per IDS standards	Data Exchange / Security	Functional
12010	As a Platoon App Provider, I want different data providers to use a common data model and APIs, so that I can easily use data from different sources to train my models without extra integration required and to be able to offer my models to different app consumers.	Data Exchange / Security	Functional
12011	As a Platoon App Consumer, I want different app providers to use a common data models and APIs, so that I can easily use models from different providers.	Data Exchange / Security	Functional
12012	As a Platoon Data Provider, I need to have a metadata based on IDS vocabulary so that I can explicitly define terms and conditions that guarantee data sovereignty.	Data Exchange / Security	Functional
12013	As a Platoon App Provider, I want data providers to convert data in a semantic format (RDF) by reusing/extending domain ontology or creating a new domain ontology from scratch, so that I can be able to derive and infer more information using data relationships.	Data Exchange / Security	Functional
12014	As a Platoon Data Provider, I want to publish different versions of a data source and mark versions as deprecated.so that my dataset is always up to date	Data Exchange / Security	Functional
12015	As a Platoon Data Consumer, I want to search/query for a relevant dataset in the Broker Service Provider, so that I can find the relevant dataset useful for my business.	Data Exchange / Security	Functional
12016	As a Platoon Data Owner, I want to update my dataset´s metadata at the Broker Service Provider so that the dataset is up to date.	Data Exchange / Security	Functional
12017	As a Platoon Vocabulary Provider, I want to manage, edit, update, extend and publish different versions and mark versions as "deprecated" of Platoon-energy related vocabularies so that any modification results in a new version of the vocabulary in order to stay consistent with its users.	Data Exchange / Security	Functional
12018	As a Platoon MarketPlace Operator, I want to provide an registration interface for data providers, so that they can register their dataset´s metadata.	Data Exchange / Security	Functional
12019	AS a Platoon MarketPlace Operator, I want to store data provider´s metadata so that it will be visible to all participants within the Platoon ecosystem.	Data Exchange / Security	Functional
12020	As a Platoon MarketPlace Operator, I want to provide a query interface (optional: GUI) for data consumers, so that they can search for a specific dataset.	Data Exchange / Security	Functional
12021	As a Data Provider/Consumer, App Provider/Consumer, I want to have a unique identity in the Platoon ecosystem in the form of a certificate, so that secure and trusted connections to other participants can be established during the data exchange.	Data Exchange / Security	Functional
12022	As an App Provider, I want to be sure that companies I am going to share my models with meet minimum legal and IPR requirements so that they don´t make fraudulent use and exploitation of provided analytics tools.	Data Exchange / Security	Non-Functional
12023	As a Data Owner, I want clear definitions of possible licenses for the data I provide as open data, so that the usage limitation is clear.	Data Exchange / Security	Functional
12024	As a Data User, I want the API to be described in a standard manner (e.g. OpenAPI specification), so that it will be more transparent (e.g. data access, type of authorizations, possible responses, etc.)	Data Exchange / Security	Functional
12025	As a Data Owner, I need to limit data access via API only for authorized entities, so that data is only available for intended parties.	Data Exchange / Security	Functional
12026	As a Platoon Data Provider, I want personal data to be anonymized so that I comply with GDPR.	Data Exchange / Security	Functional
12027	As a Platoon Data provider, I want personal data aggregated so that I keep confidential critical business keys.	Data Exchange / Security	Functional

12028	As Data Provider, I want companies that are going to have access to data, to meet certain minimum security requirements, so that I can be sure that my data is going to be safe outside my system and that GDPR is going to be complied with	Data Exchange / Security	Functional
12029	As Data Consumer, I want the companies that are going to provide my data to meet security requirements so that they do not create cybersecurity threats to my system.	Data Exchange / Security	Functional
12030	As Data Consumer, I want to know the specific data privacy and usage requirements so that I can be sure to comply with GDPR and only use the data sticking to them.	Data Exchange / Security	Functional
12031	As a Data Consumer, I want to be sure that the companies that provide the data I am going to use meet ownership and sovereignty requirements, so that I have the permission to use data (e.g. data has been retrieved legally, not stolen).	Data Exchange / Security	Functional
12032	As a Data Consumer, I want to be sure that the companies that provide the data meet quality and provenance requirements (original source and all the subsequent transformations), so that I can be sure that the models I develop based on the data are going to perform well when applied to other datasets.	Data Exchange / Security	Functional
12033	As Data Provider/Consumer and App Provider/Consumer, I want to be part of an ecosystem where all the stakeholders meet data quality, security and privacy requirements, so that I don´t have to check them every time I want to create a new data connection.	Data Exchange / Security	Functional
12034	As a Data Provider, I want my data to be encrypted every time it is sent to a Data Consumer, so that were there to be a malicious attack and the data is intercepted, they cannot extract any valuable information from my data.	Data Exchange / Security	Functional
12035	As a Platoon Data Provider/Consumer, I want to manage consent over personal data, so that I will be able to manage who can access and process my personal data.	Data Exchange / Security	Functional
12036	As a Platoon Data Provider/App Provider/Service Provider, I want to access control functionalities so that only authenticated and authorized people can access the data/services.	Data Exchange / Security	Functional
12037	As a Platoon Data Provider/Consumer, I want to comply with state of the art solutions regarding security, so that the data exchange process is secure.	Data Exchange / Security	Functional
12038	As a Data Consumer / Energy Company (i.e. Energy production company, TSO, DSO, ESCO, etc.), I want to be able to access data from different companies within the same/different sectors, so that I can do benchmarking with similar scenarios in order to be able to assess competitiveness level.	Data Exchange / Security	Functional
12039	As a Data Consumer/ Equipment Manufacturer (OEMs, Tier 1 suppliers, etc.), I want to be able to access asset operational data, so that I can understand equipment behaviour in real scenarios and be able to improve the design.	Data Exchange / Security	Functional
12040	As an App Consumer(energy company such as production companies, TSO, DSO, ESCO, etc.), I want to be able to use data analytics tools that have been already developed and validated by other companies so that I can get value from my data at low cost and low time to market and be able to use my time to focus on the critical parts of my business while offering a better service.	Data Exchange / Security	Functional
12041	As Data Provider, I want to manage/post-process/enhance the data produced at my facilities, so that I can assess the availability and performance of the plant (subsystems and assets) and its remaining lifetime, in order to improve operation programs and implement optimized maintenance strategies that will reduce operation and maintenance costs (OPEX) of the plants.	Data Exchange / Security	Functional
12042	As Data Provider/User, I want to have real-time access to the Renewable Energy production data and consumption demands and patterns, so that I can have a reliable overview on the grid capacity and provide services adapted to the needs of the different agents connected and/or involved in the grids (DR, frequency balance, etc.)	Data Exchange / Security	Functional
12043	As Platoon MarketPlace Operator, I want to have real-time access to energy consumption needs of customers I aggregate and represent with online information of the electricity wholesale market so that I can offer them an optimized service (energy quality and availability at best prices) for their specific quantities and timeframes.	Data Exchange / Security	Functional

12044	As Data Consumer, I want to have access to databases and analytical tools that can analyze my generation/consumption patterns so that I can better match my energy needs with the electricity I generate and demand from external suppliers to be able to optimize the return of my investment. (prosumers)	Data Exchange / Security	Functional
12045	As Data Consumer, I want to have easy access to big volumes of data (from renewable power plants and smart grids) at low cost, so that I can develop applications and can offer added value products or services (SaaS and PaaS).	Data Exchange / Security	Functional
12046	As Data Provider/Consumer (wind turbine OEM), I want to obtain data from different subsystems and components in the operational wind turbines I manufacture to be able to extract knowledge of how the performance of each of them is affected by others, so that I can acquire deeper knowledge of the technical features and performance of the critical systems and components and improve new model design.	Data Exchange / Security	Functional
12047	As Data Provider/Consumer (wind turbine OEM), I want to have legal access to competitor data so that I can offer an increased portfolio to customers (benchmarking, predictive maintenance, etc.)	Data Exchange / Security	Functional
12048	As a Data User (energy community), I want to access open data with open access and license, so that I can use the data to engage the renewable community (e.g. hackathon, journalism, policy making, etc.)	Data Exchange / Security	Functional
12049	As a Data User (energy community), I want to see a demo and tutorial of possible analysis (sample data provided), so that I can learn to use the analytic tool to perform analysis.	Data Exchange / Security	Functional
12050	As a Platoon App Provider, I want to be able to receive raw data from different data providers, so that I can train my models with a wide range of scenarios and achieve a strong generalization capacity.	Data Exchange / Security	Functional
12051	As a Platoon Data Consumer, I want to use well-documented data (creation, conditions, missing values, domain specific knowledge treatment, etc.), so that I can select and fit the ML pipeline (pre-processing models and evaluation)	Data Exchange / Security	Functional
12052	As a Platoon Data Consumer (AI app), I need to have consistent and high-quality data so that I can produce useful and reliable results	Data Exchange / Security	Functional
12053	As Data User (energy community), I want to see data analysis show case and examples so that I know what type of specific analysis can be performed and what data type/structure/format is required for this analysis.	Data Exchange / Security	Functional
12054	As a Data User (ML Engineer/Data Scientist), I need the data to be properly labelled so that it can be used for development and testing of AI-based analytics.	Data Exchange / Security	Functional
12055	As a Data user, I want to be able to see if API stacking scenario can be done so that I can see what type of data mash up is possible.	Data Exchange / Security	Functional
12056	As a Data User (ML Engineer), I want an option of pseudonymization and data sampling, so that I can use the data for scientific disseminations which showcase opportunities and quality of the provided ML approaches of the generic Big Data Analytics toolbox.	Data Exchange / Security	Functional
12057	As a Data User (Data scientist/ML API developer), I want an easy set up (e.g. pipelines), so that API stacking can be performed correctly.	Data Exchange / Security	Functional
12058	As a Data User (Data Scientist) and Data Owner, I need an assessment method to see if the data is provided as a high-quality data (i.e.. data veracity ensured), so that we can use it to assess the data and avoid an undesired result cause by “garbage-in, garbage-out” concern.	Data Exchange / Security	Functional
12059	As a Data User (Data Scientist) and Data Owner, I need verifiable date, so that the provenance is clear.	Data Exchange / Security	Functional
12060	As a Data User (Data Scientist) and Data Owner, I need the data to be accompanied by metadata, so that it has a higher level of interoperability due to a more facilitated interpretation.	Data Exchange / Security	Functional