H2020 Work Programme 2016-2017
H202-IoT-02-2016
Duration: 36 months

CROSS FERTILISATION THROUGH ALIGNMENT, SYNCHRONISATION AND EXCHANGES FOR IoT

H2020 – CREATE-IoT Project

Deliverable 06.02

Recommendations for commonalities and interoperability profiles of IoT platforms

Revision:	**1.00**
Due date:	**31-07-2018 (m19)**
Actual submission date:	**30-09-2018**
Lead partner:	**ETSI**

Dissemination level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Summary					
No and name	**D06.02 Strategy and coordination plan for IoT interoperability and standard approaches.**				
Status	<Released>	**Due**	m19	**Date**	31-07-2018
Author(s)	E. Darmois (ETSI), D. Raggett (ERCIM), O. Vermesan (SINTEF), R. Bahr (SINTEF), M. Serrano (NUIG), A. Tringale (ISMB), F. Sottile (ISMB)				
Editor	E. Darmois (ETSI)				
DoW	Recommendations for commonalities and interoperability profiles of IoT platforms. The work has been carried out within task T06.01 (IoT Interoperability, standards approaches, validation and gap analysis), and is the second out of three deliverables from this task. The task coordinates the activities with the AIOTI WG on standardisation, SDOs and other various IoT Global Alliances for the validation in usage context of most promising standards and gap analysis identification. It addresses interoperability and integration, through open IoT platforms.				
Comments					

Document history			
Rev.	**Date**	**Author**	**Description**
0.00	27-02-2017	SINTEF	Template/Initial version.
0.01	22-12-2017	ETSI	Initial description of work, and structure.
0.02	24-07-2018	ETSI	Revamped Table of Content with alignment on AG02 work.
0.03	07-08-2018	ETSI	Improved version with most of sections 3 and 6.
0.04	14-08-2018	ETSI, SINTEF	First version with sections mostly filled (except section 5)
0.05	20-08-2018	SINTEF	Section 4 AUTOPILOT.
0.06	25-08-2018	ETSI, SINTEF	All section filled, overall review and improvement.
0.07	17-09-2018	ISMB, UNP	Sections 4.3.4, 5.3.5 and 6.2.4 on MONICA and IoF2020
0.08	19-09-2018	ETSI	Review of sections 4 to 6. Lightweight overall review.
0.09	22-09-2018	SINTEF	Update IoT reference architecture
0.10	25-09-2018	NUIG	Sections 3.1, 3.2, Section 3.3, 4.3.1, 5.3.2 and 6.2.1
0.11	25-09-2018	ETSI	Finalisation (including SYNCHRONICITY sections)
1.00	26-09-2018	SINTEF	Final version released

Disclaimer
The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

The document reflects only the author's views and the EC is not liable for any use that may be made of the information contained therein.

Table of contents

1. EXECUTIVE SUMMARY

1.1 Publishable summary

The Internet of Things (IoT) seeks to enable services based upon sensors and actuators that are connected to the Internet. Unlocking the benefits of IoT across many potential application areas, reducing the costs and risks, and providing the confidence needed for sustainable growth of the IoT ecosystem will require that interoperable platforms, standards and technologies be available to the IoT systems designers and developers.

The primary purpose of this document is to outline the basic requirements for a common interoperability approach for the IoT Large-Scale Pilots (LSPs). The focus of the document is analysing the support for interoperability within the LSPs and which commonalities or discrepancies can be found. This common approach is based on a technical framework for IoT interoperability and how it can be supported by Reference Architectures, Interoperability support mechanisms and the support from Platforms and Technologies.

The results available at the mid-life of the IoT LSPs show that there is a common view across the IoT LSPs (and the associated CSAs) about some the requirements on interoperability, despite a relative dispersion in terms of platforms and technologies used for the implementation of the LSPs Use Cases.

Beyond the common reference and a preliminary synthesis developed in the current document (which is a complemented by deliverable D06.05 "Initial report on IoT standardisation activities" that has a similar approach regarding the questions of standardisation), other complete and operational documents will be developed by CREATE-IoT and the LSP representatives in the IoT LSP Activity Group 2 (IoT standardisation, architecture and interoperability), in particular via a number of Workshop – to be held in the second half of 2018 and in 2019 - addressing the above topics in details.

1.2 Non-publishable information

None, the document is classified as public.

2. Introduction

2.1 How to use this document

2.1.1 Scope and purpose

The primary purpose of this document is to outline the choices and possible approaches of the IoT Large-Scale Pilots (LSPs) regarding interoperability and platform/technology activities. The current choices in the LSPs – in particular in the LSP Use Cases under development – is examined, both in terms of support of interoperability (and some associated mechanisms) and platform and development technology choices. Based on this, a detailed analysis is made in order to identify the common choices across LSPs that can be used not only by the LSPs, but also by other IoT systems development projects.

2.1.2 Target group

The target group for this document is the community of people that have to address the definition of the LSPs from inception to implementation, in particular regarding the main technical choices that have to be made in order to ensure that the implementations will be effective, interoperable and scalable:

- The identification and description of the Use Cases selected by the LSPs;
- The selection of the reference architecture for the description of the interoperability layers and the main building blocks for the implementation of the Use Cases;
- The identification of the main elements of the framework that will be used for the implementation of the selected Use Cases (e.g., development methodology, development environments;
- The identification of commonalities and differences regarding IoT platforms interoperability;
- The identification of best practices based on commonalities and difference identified;
- The provision of practical guidelines for the LSPs regarding common solutions;
- The provision of guidelines for the IoT community based on the feedback from the LSPs.

2.2 Contributions of partners

This deliverable is the second deliverable of CREATE-IoT Task 06.01 (IoT Interoperability, standards approaches, validation and gap analysis). The list below shows the specific contribution of partners to the current deliverable and recalls the overall scope of their expected contribution to Task 06.01.

ETSI: As Task Leader and editor of the deliverable, ETSI has contributed to the definition of the overall content and scope of the deliverable, to the definition of the IoT Interoperability Framework, to the synthesis of the main elements of the Framework based on the Activity Group 02 Workshops, and to the review of the deliverable.

ERCIM has contributed to the definition of the overall content and scope of the deliverable, to several sections of the document (based in particular on the work done in Activity Group 02), and to the review of the deliverable.

NUIG provided contributions to various sections of the document based on its involvement in one of the LSPs (e.g. ACTIVAGE), and to the review of the deliverable.

SINTEF has contributed to the definition of the overall content and scope of the deliverable, provided contributions to various sections of the document based on its involvement in one of the LSPs (e.g. AUTOPILOT), and to the review of the deliverable.

2.3 Relations to other activities in the project

The present document is one of the deliverables of CREATE-IoT Work Package 6 "IoT Interoperability and Standardization". WP06 is structured into two complementary tasks:

- Task 06.01 (IoT Interoperability, standards approaches, validation and gap analysis) focuses on practical topics regarding the implementation of LSP Use Cases;
- Task 06.02 (Pre-normative and standardisation activities) focuses on the contributions from the LSPs and CREATE-IoT to the IoT standards ecosystem. The present document is a deliverable of this task.

A very important part of the work for the present deliverable has been done in the context of the IoT LSPs Activity Group 02 (AG02 - "IoT standardisation, architecture and interoperability"). The Activity Group 02 coordinates the activities of the LSPs and of the two associated Coordination and Support Actions (CSA), CREATE-IoT and U4IoT, on interoperability and standardisation. The information has been gathered and discussed in three AG02 workshops held in the first half of 2018 (on January 10th, April 26th and June 6th). The objectives of these workshops were to establish a common basis across the different IoT Large-Scale Pilots (LSPs) regarding their results related to topics such as: mapping pilot architecture approaches based on possible reference architecture models; interoperability framework and reference implementations based on use cases; analysis of components and standards, existing IoT platforms that may be reused/tested across multiple use cases and enable interoperability across those.

The present deliverable is a follow-up of Deliverable 06.07 "Strategy and coordination plan for IoT interoperability and pre-normative and standardisation activities" whose primary purpose is to outline the basic requirements for a common interoperability and standardisation strategy to be adopted by the IoT Large-Scale Pilots (LSPs). Deliverable 06.07 is a basic, initial reference in defining and understanding the main issues regarding IoT interoperability and standards. It sets the scene for additional deliverables in WP06.

The present deliverable is complementary to deliverable D06.05 "Initial report on IoT standardisation activities" (produced in Work Package 6 Task 06.02). Whereas the present deliverable focuses on interoperability and platform aspects, D06.02 focuses on the standardisation aspects.

In addition, this standardisation strategy that the current document is addressing is potentially coming in support of CREATE-IoT Work Package 2 (IoT Large-Scale Pilots Ecosystems Arena for Sharing Common Approaches), in particular when it comes to open APIs or common methodologies. This is the case of deliverable D02.02 (Reference architecture for federation and cooperation between IoT deployments).

3. AN INTEROPERABILITY FRAMEWORK FOR IOT

3.1 Requirements for IoT interoperability

The Internet of Things is in the process to be considered as the technology that will change the way we solve problems and define services and business, IoT is a very large domain, spanning across variety of business sectors (e.g., food, health, industry, transportation, etc.). The main challenge faced by those who want to develop and deploy IoT systems is the Interoperability and scalability, it is well know the number of IoT devices is large and thus the solutions and services must be deployed with the design idea that will be widely used by many users. In contrast to this design principle imposed by IoT, many of the available solutions in the market have been developed in the form of application silos, where interoperability is limited by the scope of the specific solutions selected and cannot operate in other systems or domains; however in contrast this market approach is able to support large number of devices and users connected at the same time providing same reliability and resilience in their approach. This can be seeming to be a trade-off between interoperability and scalability, but certainty is not, it is important to identify the best practices to design, develop and implement large-scale interoperable IoT systems using as many generic solutions, but following standards, that potentially can be apply the design principles, even if they are in different sectors. The main objective for building an interoperability framework is reduce the duplication and fragmentation either in the domain ecosystem or across different ecosystems. As a result, the huge potential for the IoT is being held back by fragmentation into incompatible platforms, standards and technologies.

A key objective for IoT systems designers and developers is to make sure that interoperability is supported wherever it is needed within or across IoT systems. Interoperability can be seen through various angles: operational behaviour, information exchange, etc.

Though interoperability is often seen as a means for two systems to exchange information at the network layer, however there are much more aspects to interoperability that not only rely in networking protocols. In particular, a layered approach to interoperability has become the accepted paradigm for the description of systems (and this is also true for IoT systems) with a specific focus on four layers [1]:

- Technical Interoperability: usually associated with communication protocols and the infrastructure needed for those protocols to operate;
- Syntactic Interoperability: usually associated with data formats and encodings along with techniques for compressing them;
- Semantic Interoperability: associated with shared understanding of the meaning of the exchanged content (information);
- Organisational Interoperability: associated with the ability of organisations to effectively communicate and transfer information even across different information systems, infrastructures or geographic regions and cultures.

3.2 Main elements for IoT interoperability frameworks

3.2.1 Introduction

Interoperability requires agreements between elements of a system that may be of very different nature, it is not only technology but also other systems and elements in the ecosystem. As per the LSPs (as well as other actors in the IoT community), coming to a common understanding and to the possibility to adopt similar solutions is a latent possibility, mainly for its nature of ecosystem integrity, some elements have to be elicited that allow the expression and the comparison of the proposed solutions: they are the components of a framework for IoT Interoperability.

This section presents a quick overview of main elements of such a framework: reference architectures, support mechanisms, platforms and standards. On top of these, pre-normative activities are also touched upon: they are the basis from which new framework elements will emerge to provide new solutions to current challenges. The descriptions provided are applicable to any large ecosystem and constitutes a basic principle for IoT Interoperability frameworks.

3.2.2 Support of common IoT communication protocols

The Internet of Things market has diversified into having different adopted technologies using different communication protocols i.e. Wi-Fi, BLE. In some areas emerging technologies looking to solve the same problem are looking to be the adopted universal technology i.e. SigFox Lora, etc. Due to the fact that different types of devices can be used in different locations of the ecosystem, there should be no limitation on the supported communication protocols.

3.2.3 Support for M2M communications

The interaction between IoT nodes can follow the concept of Machine-to-Machine (M2M) communication. M2M refers to solutions that allow machines to communicate with back-end information systems and/or directly with other machines, in order to provide real-time data and run processes. M2M communication can be based on events and/or polling (predefined time intervals). M2M applications amongst other related to secure the information transmitted consider the following steps in the lifecycle: data collection, data transmission, data validation and response to available information.

3.2.4 Support of the main IoT middleware platforms

The access to the different adopted/integrated platforms should be facilitated. The diversity on adopted IoT middleware platforms not only at the consortium level but outside the consortium, demands to have support to the most adopted IoT middleware platforms. The designed system requires to be connected to the different IoT platforms to access its services with the objective to facilitate the software wrappers that will facilitate the interoperability.

3.2.5 Extensibility for different sensor types

The system architecture must be able to receive data from multiple sensor types. For this, the extensibility of all components of the system must be taken into account. It should be able to easily support extensions, upgrades and inclusion of new modules as they are being integrated.

3.2.6 User Device Detection Capability

The system architecture must be able to provide tools and services for checking the capacity of the user's device according to the characteristics of the device required for its application. By using device-detection techniques and exchange of communication protocols this information must be verified at the first moment, meaning at the first attempt of the connection, to avoid that the user becomes aware of the incompatibility after having begun to use the system.

3.2.7 Semantic and syntactic interoperability

Syntactic interoperability must be enabled, and it can be achieved through a simple mapping-translation mechanism, by using dedicated software components to interpret a model and get the necessary information to be passed to other component, usually these components are called wrapper and connect producers and consumers of the data.

Semantic interoperability must be supported in order to exchange not only data, but information and features related to the source of the information i.e. location, status, technology associated, etc. facilitating the disappearance of the vertical information silos of the different heterogeneous platforms that the current IoT data lakes represents. In order to achieve seamless interoperability between the different data silos or a transparent exchange of information at different levels. The information that describes the data features, data context, situational sensors location and technology associated etc., is called metadata. Semantic interoperability can be enabled by implementing an information system that manipulate the data formats and its metadata using a formal language i.e. JSON-LD, RDF, OWL that allows dedicated components to translate-interpret the data and abstract the necessary information for its use on specific purposes.

3.2.8 Gateway Capabilities and Protocol Conversion

The system architecture must have gateway capabilities and support multiple interfaces to work with different protocols and operation modes: a) the gateway could be running at the device layer, the gateway capabilities from the system that will allow devices connected through different types of wired or wireless technologies such as the CAN bus, ZigBee, Bluetooth or Wi- Fi to the system. b) In the second case at the network layer, the system architecture will host the gateway and its capabilities connecting the devices using P2P or VPN protocols.

3.2.9 Unique Device ID / Naming

An identifier system must be developed/selected from the available ones in order to be able to identify each device in a unique way. In addition, there should be no limitation on the number of devices that can be connected due to the lack of identifiers. The number of identifier codes must be large enough to accompany all current and future devices.

The technique to identify the devices must be compliant to device standards and extensive across different platforms e.g. MAC address and NOMs. Nevertheless, each connected device must be recognized in order to process data from and to the device and as required naming techniques can be applied using convention and more high-level like URIs or URL if we are taking about applications and data descriptions.

3.2.10 Standard protocols for device communications

Devices need a standard protocol to exchange data with any platform. This protocol should provide a minimum of data fields; such as ID, security, authentication, position, etc. The series of protocols are technology dependent and they should be compliant with European and the International standards.

3.2.11 Reference Architectures

In order to achieve interoperability, a lot of elements such as models, definitions or well-defined set of vocabularies need to be agreed upon by the IoT stakeholders in order to ensure a common understanding about the concepts, this is also a preamble to standardisation. Moreover, given the need to be able to deal with a potential very large variety of IoT systems architecture, it is also necessary to create, adapt or adopt High Level reference Architectures (HLA). There is a number of already existing reference architecture, some of them proposed by standards organisations.

The Reference Architectures are addressed in more details in section 4.

3.2.12 Support of design and development

For a number of IoT projects, in particular those who span large domains (e.g., Smart Cities), cross-domain interoperability is a key requirement for achieving large scale deployment of IoT-enabled services. On top of a reference architecture model, other elements are required such as cross-application interoperability points (describing where interoperability is supported) and some supporting mechanisms (describing how the support is provided). On top of ad-hoc approaches, project by project, some specifications and standards are emerging to this purpose.

The support mechanisms to interoperability are addressed in more details in section 4.

3.2.13 Platforms and technologies

There are hundreds of IoT platforms available for the development of IoT systems. The question of a choice of platform(s) by IoT system designers is complex. Some dimensions have to be considered such as their scope and breadth, the maturity and ownership of their components, and the level of support by standards. A growing role is played in this domain by the availability of a large set of solutions coming from Open Source Software communities and eco-systems. The issues related to platforms and technologies are addressed in more details in section 6.

3.2.14 Standards and pre-normative activities

Standards are a key element in the IoT Interoperability Framework. A first requirement is to clearly outline the support offered by the current state-of-the-art in standardisation. Beyond this, it is also important to outline the gaps and overlaps (in particular the standards gaps and overlaps): the missing elements of the IoT landscape, mostly due to its complexity, that need to be identified before they may be resolved in the near future.

Pre-normative activities explore promising directions, and just as importantly, attempt to present these in ways that are easy to explain to other communities, thereby helping to build a shared understanding on what new standards are needed.

As already pointed out, these aspects are addressed in the companion deliverable D06.05 (Initial report on IoT standardisation activities) [2].

3.2.15 Alignment with other IoT architectures

A key requirement for the system architecture is the alignment with the reference models of other IoT projects, especially The Alliance for Internet of Things Innovation (AIOTI) for example. The AIOTI HLA architecture model is suitable for guiding the development of any LSP architecture. The use of the AIOTI vision of the Internet Architecture of Things will be useful to use its results and other projects to avoid reinventing a new architectural model from scratch and align and be compatible with those projects.

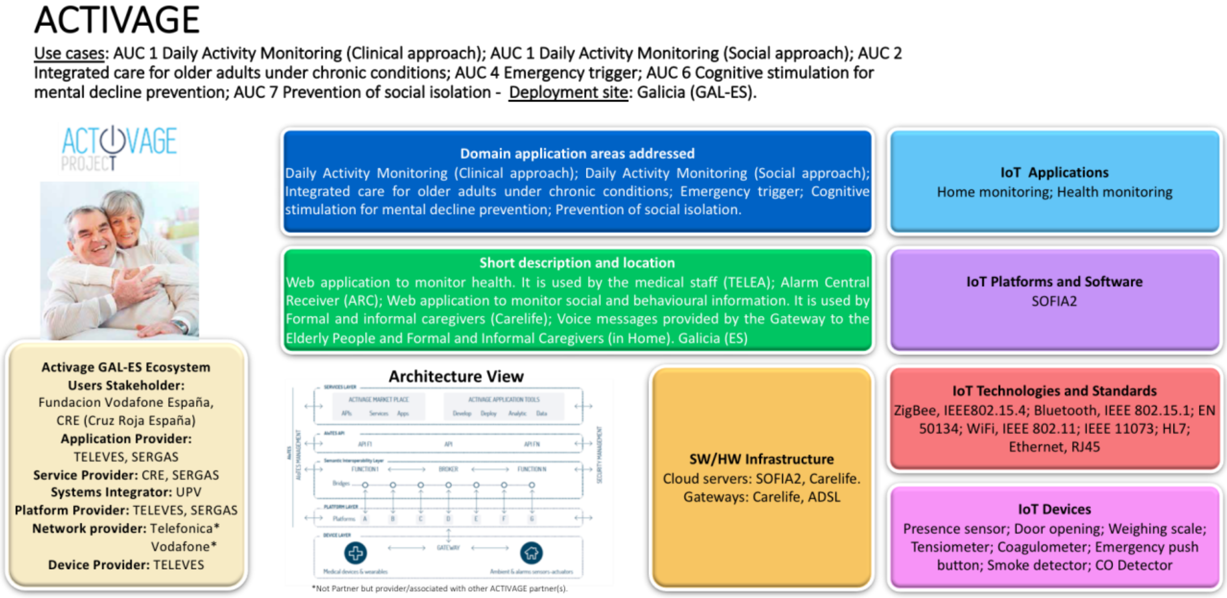
3.3 The Interoperability Framework in the LSP Use Cases

The LSP or Large-Scale Pilots in Europe are a sort of Ecosystem, where Technology, Stakeholders and Software Solutions interact in a way that all together creates a synergy of data exchange. Based on the application domain data exchange must follow specific patterns or standards. Interoperability is a requirement that at the LSP level is a condition that must be clearly specified in order to guarantee the full integration of a solution and the correct provisioning of services across the multiple solutions in the ecosystem. As per definition the requirements for interoperability defines user-centric approaches and thus use cases are the main drivers for those data interoperable specifications.

Each of the IoT European Large-Scale Pilots (LSPs) have developed its own Interoperability Framework, based on the requirements of their applications (considering in particular the sector – e.g. health, transport, etc. – in which they operate), the Use Cases that have been selected or the actual situation of the pilot sites for those Use Cases. These parallel definitions have resulted in different Interoperability Frameworks (IF). However, despite differences, a number of features of these IFs are similar or quite close, making their analysis and comparison a meaningful exercise.
This analysis has been done in three workshops held by the IoT LSPs Activity Group 2 (IoT Standardisation, Architecture and Interoperability) in the first half of 2018 (on January 10th, April 26th and June 6th) from which a significant part of the information presented in this document was derived.

The objective of these workshops was to establish a common basis across the different IoT Large-Scale Pilots (LSPs) regarding their results related to topics such as: mapping pilot architecture approaches based on possible reference architecture models; interoperability framework and reference implementations based on use cases; analysis of components and standards, existing IoT platforms that may be reused/tested across multiple use cases and enable interoperability across those.

All the LSPSs have defined a number of Use Cases that are under development in a number of pilot sites. For the sake of information sharing, all these Use Cases have been gathered in a systematic and comparable manner by Activity Group 01 (IoT Focus Area Sustainability) that presents the main characteristics of these Use Cases in a common template (as illustrated in Figure 1). These results are used in the present report for the analysis of reference architectures, interoperability support mechanisms, and platforms and technologies.



ACTIVAGE

Use cases: AUC 1 Daily Activity Monitoring (Clinical approach); AUC 1 Daily Activity Monitoring (Social approach); AUC 2 Integrated care for older adults under chronic conditions; AUC 4 Emergency trigger; AUC 6 Cognitive stimulation for mental decline prevention; AUC 7 Prevention of social isolation - Deployment site: Galicia (GAL-ES).

Domain application areas addressed
Daily Activity Monitoring (Clinical approach); Daily Activity Monitoring (Social approach); Integrated care for older adults under chronic conditions; Emergency trigger; Cognitive stimulation for mental decline prevention; Prevention of social isolation.

IoT Applications
Home monitoring; Health monitoring

Short description and location
Web application to monitor health. It is used by the medical staff (TELEA); Alarm Central Receiver (ARC); Web application to monitor social and behavioural information. It is used by Formal and informal caregivers (Carelife); Voice messages provided by the Gateway to the Elderly People and Formal and Informal Caregivers (in Home). Galicia (ES)

IoT Platforms and Software
SOFIA2

Activage GAL-ES Ecosystem
Users Stakeholder:
Fundacion Vodafone España,
CRE (Cruz Roja España)
Application Provider:
TELEVES, SERGAS
Service Provider: CRE, SERGAS
Systems Integrator: UPV
Platform Provider: TELEVES, SERGAS
Network provider: Telefonica*
Vodafone*
Device Provider: TELEVES

Architecture View

*Not Partner but provider/associated with other ACTIVAGE partner(s).

SW/HW Infrastructure
Cloud servers: SOFIA2, Carelife.
Gateways: Carelife, ADSL

IoT Technologies and Standards
ZigBee, IEEE802.15.4; Bluetooth, IEEE 802.15.1; EN 50134; WiFi, IEEE 802.11; IEEE 11073; HL7; Ethernet, RJ45

IoT Devices
Presence sensor; Door opening; Weighing scale; Tensiometer; Coagulometer; Emergency push button; Smoke detector; CO Detector

Figure 1: An example of LSP Use Case template

The above template gathers information related to the main elements of the IoT Interoperability Framework: Reference Architecture, Platforms, Devices and Standards. This document addresses in particular the information related "Architecture View" (in section 4) and "IoT Platforms and Software" (in section 6).

4. INTEROPERABILITY: REFERENCE ARCHITECTURES

4.1 Introduction

This section introduces the principles and main examples of Reference Architecture Models. The objective is to have a comparison point with respect to the architectural models selected and used by the IoT LSPs in the definition, the design and the implementation of their Use Cases.

The Reference Architecture models presented incorporate, in most cases, several viewpoints that each address a specific way to analyses an IoT system. Some of these views will be important for the structuring of IoT systems in support of interoperability: some generic principles (e.g., layering) can be outlined. One of the objectives of this section is to understand how the LSPs Reference Architecture models have adopted these interoperability principles and therefore to outline the commonalities between the LSPs.

Note: The generic Reference Architecture models, as well as those chosen by the IoT LSPs, are analysed in more details in the Deliverable D02.02 "Reference architecture for federation and cooperation between IoT deployments" CREATE-IoT Work Package 02 [5].

4.2 Reference Architecture Models

4.2.1 Rationale

In order to achieve interoperability, a lot of elements such as models, definitions or well-defined set of vocabularies need to be agreed upon by the IoT stakeholders in order to ensure a common understanding of the concepts. Moreover, given the need to be able to deal with a potential very large variety of IoT systems architectures, it is also necessary to create high level reference architectures (HLA).

The creation and utilization of reference architectures may provide a blueprint for the design and development of future systems and components and can be used for various purposes such as:

- To communicate on a common view and language about a system, within a sector, across industry, customers, regulators, etc.;
- To support the analysis and evaluation of variant implementations of an architecture;
- To integrate various existing state-of-the-art approaches into one model;
- To support the transition from an existing legacy architecture to a new architecture;
- To help assessing conformance to identified standards or interoperability requirements;
- To document decisions taken during the development process of a system.

4.2.2 Examples

Some examples developed in Projects, Standardisation organisations or Alliances are listed in the table below. They come from different sources such as research projects, standards organisations or alliances.

Table 1: Examples of IoT Reference Architectures

Organisation	Description
AIOTI	The HLA primarily introduces a domain model, which describes entities in the IoT domain and the relationships between them, and a functional model, which describes functions and interfaces (interactions) within the IoT domain. The AIOTI functional model describes functions and interfaces between functions of the IoT system.

	Functions do not mandate any specific implementation or deployment
ETSI CIM NGSI-LD	ETSI NGSI-LD is the specification of a standard Application Programming Interface (API) to manage Context Information Management at large scale. The adoption of this API in order to get access to context data allows breaking silos and vendor-locks and abstracts the complexity and low-level details of the multiple IoT protocols that may be part of the same system. An open-source reference implementation of NGSI-LD standard is provided with FIWARE Context Broker component.
IEEE P2413	IEEE P2413 defines an architectural framework for the IoT that provides a reference model which defines relationships among various IoT verticals (e.g., transportation, healthcare, etc.) and common architecture elements. It also provides a blueprint for data abstraction and the four elements for trust that include protection, security, privacy, and safety."
IIC	The Industrial Internet Consortium (IIC) has developed the Industrial Internet Reference Architecture (IIRA) in order to address the need for a common architecture framework to develop interoperable IIoT systems for diverse applications across a broad spectrum of industrial verticals in the public and private sectors.
Industrie 4.0	Industrie 4.0 aims at connecting all stakeholders involved in the business processes of the manufacturing and process industry so that all participants involved share a common perspective and develop a common understanding. The Reference Architectural Model Industrie 4.0 (RAMI 4.0) is a three-dimensional map in support of the most important aspects of Industrie 4.0.
IoT-A	The IoT-A (Internet of Things - Architecture) project has proposed an IoT-A Architectural Reference Model (ARM) together with the definition of an initial set of key building blocks. The IoT-A ARM is a set of best practices, guidelines, and a starting point to generate specific IoT architectures
ISO/IEC CD 30141	ISO/IEC CD 30141 (developed by ISO/IEC JTC 1) is a standardized IoT Reference Architecture using a common vocabulary, reusable designs and industry best practices. It uses a top down approach, provides a generic IoT conceptual model, a high-level system-based reference model and five architecture views: functional view, system view, user view, information view and communication view.
ITU-T SG13 Y.2060	ITU-T Y.2060 provides an overview of the IoT; clarifies the concept and scope of the IoT, identifies

	the fundamental characteristics and high-level requirements of the IoT and describes the IoT reference model: four layers (application, service & application support, network and device), complemented with two verticals (management and security).
OASC	OASC addresses a technical plane and focuses on off-the-shelf and open data platforms and solutions that have been extensively assessed before. A clear objective is to guarantee interoperability among cities with a "driven-by-implementation" approach relying on three domains: Common APIs, Data Models and Open Data Platform.
oneM2M	oneM2M is developing specifications for the service layer for machine-to-machine communication and the IoT. oneM2M aims to provide common services layer to IoT applications and devices of different service domain/verticals. The oneM2M Common Services Layer provides common service functions to applications and devices in the form of APIs: by providing the common services layer, different vendors and service domains can use the same APIs

4.2.3 Main elements

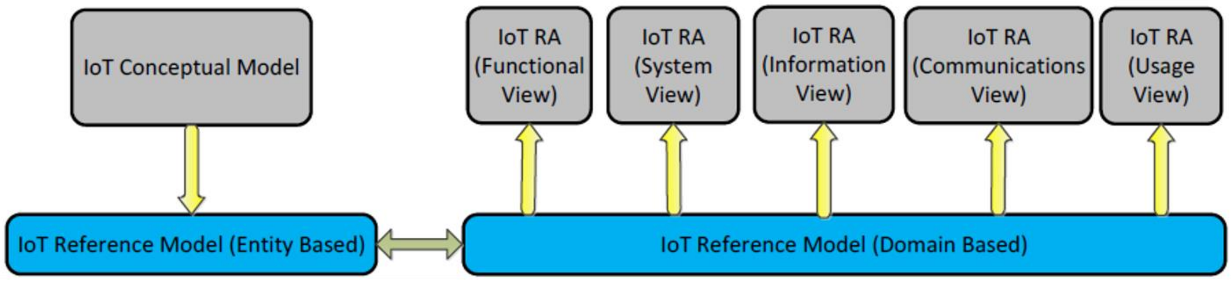
A key element of the Reference Architecture is the Conceptual Model. It is a high-level representation of the major stakeholders or the major (business) domains in the system and how they interact.

A Conceptual Model is an initial support to the common work of various stakeholders but is, by nature, not sufficient and will be complemented by a number of other views that correspond to various approaches to the definition of an IoT system.

As an example, on top of the IoT Conceptual Model it has developed, ISO/IEC CD 30141 defines 5 additional views:

- The Functional View which is a technology-agnostic view of the functions necessary to form an IoT system. The functional view describes the distribution of and dependencies among functions for support of activities described in the user view, and addresses the following concepts;
- The System View which describes the generic components including devices, sub-systems, and networks to form an IoT system. While the functional view describes an IoT system through its functional components, the system view describes it through its physical components;
- The Communications View which describes the principal communications networks which are involved in IoT systems and the entities with which they connect;
- The Information View which defines the structure (e.g. relations, attributes, services) of the information for Entities on a conceptual level. Data is defined as pure values without relevant or useable context. Information adds the right context to data and offers answers to typical questions like why, who, what, where and when;
- The Usage View which focuses on how the IoT system is developed, tested, operated and used from a user perspective.

The Figure 2 summarizes the approach taken in ISO/IEC CD 30141 [6] and how the various views are derived from the Conceptual Model.


IoT Conceptual Model
IoT RA (Functional View)
IoT RA (System View)
IoT RA (Information View)
IoT RA (Communications View)
IoT RA (Usage View)
IoT Reference Model (Entity Based)
IoT Reference Model (Domain Based)

Source: ISO/IEC CD 30141

Figure 2: Deriving Views from the Conceptual Model in ISO/IEC CD 30241

4.2.4 The Functional view

Amongst the different view outlined above, the Functional View plays an important role. Whereas the Conceptual model aims at allowing communication amongst potentially very different stakeholders (e.g., users, business developers, etc.), the Functional View will be used for the articulation of requirements (and potentially Use Cases) and the early stages of design. Very often, the IoT Reference Architecture presented by an IoT project is in fact a Functional View of the IoT system.

The Functional View will address the arrangement of functionality across the IoT system and will be key in the actual support of interoperability since it is a recognized way to introduce elements such as interoperability points, mechanisms, APIs (that are addressed in section 5).

From a functional perspective, two main elements have to be taken into consideration: layering and cross-cutting functions. Both aspects are addressed in more details in the following sub-sections.

4.2.4.1 Horizontal Layers

The benefits of introducing layers in a Reference Architecture is now well documented (with the ISO Stack model as a first example).

IoT systems are no different from others from this standpoint. However, the massive deployment of such IoT systems occurs at the same time of the generalisation of the use of Cloud Computing (and the emergence of IoT Virtualisation) and the enormous progress in Big Data and analytics.

Any layering of IoT systems must take these into account. In addition, new requirements regarding the computation and storage of IoT data have fostered the adoption of Edge Computing solutions that Reference Architectures must support.

The IoT LSPs have taken this into account together with the (specific) needs of their deployment domain. A consolidation of the horizontal layers (and components) in the five LSPs has been done and is presented in [4].

Table 2: Layers, components in the LSP architectures

Layers	Components	Description
Collaboration and processes	Business System Integration	Enables integration with existing enterprise and other external systems
Applications	Visualisation	Present device data in rich visuals and/or interactive dashboards
Service	Development Environment	Provide integrated development environment to simplify development of apps

	Service Orchestration	Support mashup of different data streams, analytics and service components
	Advanced Analytics	Allows insights from data to be extracted and more complex data processing to be performed
Abstraction	Event and Action Management	Simple rules engine to allow mapping of low-level sensors events high level events and actions
	Basic Analytics Action	Provides basic data normalisation, reformatting, cleansing and simple statistics
Storage	Storage/Database	Cloud based storage and database capabilities (not including on-premises solutions)
Processing	Device Management	Enables remote maintenance, interaction and management capabilities of devices at the edge
	Edge Analytics	Capabilities to perform processing of IoT data of devices at edge as opposed to cloud
Networks & Communications	Connectivity Network / Modules	Offers connectivity networks / HW modules enabling air interface connectivity
	Edge Gateway (HW based)	Offers IoT gateway devices from bridge connectivity from IoT nodes into the cloud-based platform
Physical / Device Layer	Operating System	Offers low-level system, SW managing SW/HW and runs applications
	Modules & Drivers	Offers adaptable modules, drivers, source libraries that reduce development and testing time
	MPU/MCU	Offers multi-purpose programmable electronic devices at Microprocessor/Microcontroller level

Regarding the layers, all the LSPs have adopted very similar approaches:

- All the LSPs share the above layers from "Applications" down to "Physical / Device Layer";
- All the LSPs with the exception of AUTOPILOT have adopted a "Collaboration and Process" layer.

4.2.4.2 Cross-cutting Functions

Some requirements of IoT systems have to be taken into consideration across several layers, often referred to as "cross-cutting".

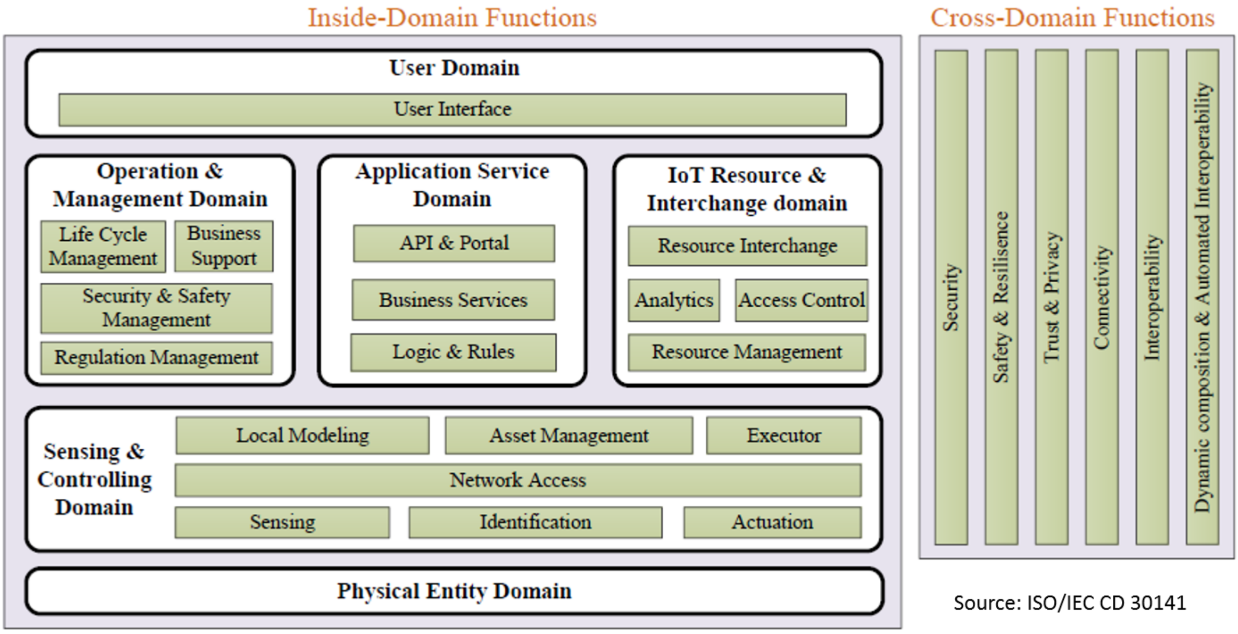
The need to support such consistent views across several layers has an impact on the Reference Architecture and, from a functional viewpoint, will correspond to cross-cutting functions.

They refer to properties of the IoT system which are not resulting from just functional components but more from the interactions amongst these components.

Cross-cutting functions may be less easy to design and deliver than those provided within layers. However, they are of extreme importance for the proper operation of the IoT systems. For example, security is of crucial importance to any IoT system.

The same is true for trust and privacy. Similarly, safety is very important to Industrial IoT Systems.

Amongst the Reference Architectures listed in Table 1, the ISO/IEC is a good example which is relatively inclusive of all the LSPs Reference Architectures when it comes to cross-layers functions.



Source: ISO/IEC CD 30141

Figure 3: IoT Reference Architecture functional view (ISO/IEC)

More details on the (cross-cutting) Cross-Domain functions can be found in Table 3.

Table 3: Cross-Layer Functions in ISO/IEC CD 30141

Cross-Layer Function	Description
Security	The security function refers to the ability of IoT system to ensure the confidentiality, integrity, authenticity and confirmation of the exchanged information. The IoT RA integrates security policies for IoT components as key part of system design. For example, asset management in the SCD enables operations management including system configuration, policy, software and firmware updates and other lifecycle management operations. In the RID, access control and the resource management are responsible for data security, data access control and data rights management.
Safety and Resilience	The safety and resilience function are a superset of fault tolerance and closely related to autonomic computing capabilities of self- healing, self-configuring, self-organizing and self- protecting, e.g., the IoT component can take advantage of the hierarchical network to do self- optimization

Trust and Privacy	The trust & privacy function is to distinguish different levels of trust for a party (e.g., application, system, network, etc.) during data transmission or exchange in order to protect the confidentiality of data. Usually, validation is required before the trust is established and trust can be enhanced by reputation-based approaches. Privacy may be achieved mostly via authentication. To prevent leaking of confidential data, additional data access rules may be used to meet necessary requirements for data requisition, removal, encryption, etc.
Connectivity	The connectivity function provides the capability of heterogeneous integration for IoT components, which may belong to different networks or using different technologies, to achieve seamless connection of each entity.
Interoperability	The interoperability function provides the capability to exchange information of an IoT system with a common interpretation of information. Basically, two levels of data interoperability are considered. Syntactic interoperability is to exchange information in a common data format with a common protocol to structure the data. Semantics interoperability is to interpret the meaning of the symbols in the messages correctly.
Dynamic composition and Automated Interoperability	The dynamic composition & automated interoperability function provides a flexible method of composing services so that the IoT components can be dynamically integrated at run-time to enable adaptable services. Semantic interoperability is required to support the dynamic composition.

In the case of the IoT LSPs, the support of cross-cutting functions may vary from one LSP to another one. Nevertheless, security is a common concern and the cross-cutting function is present (to various degrees) in all the LSPs.

4.3 Reference Architecture(s) in the Large-Scale Pilots

As already pointed out, all the LSPSs have defined a number of Use Cases that are under development in a number of pilot sites.

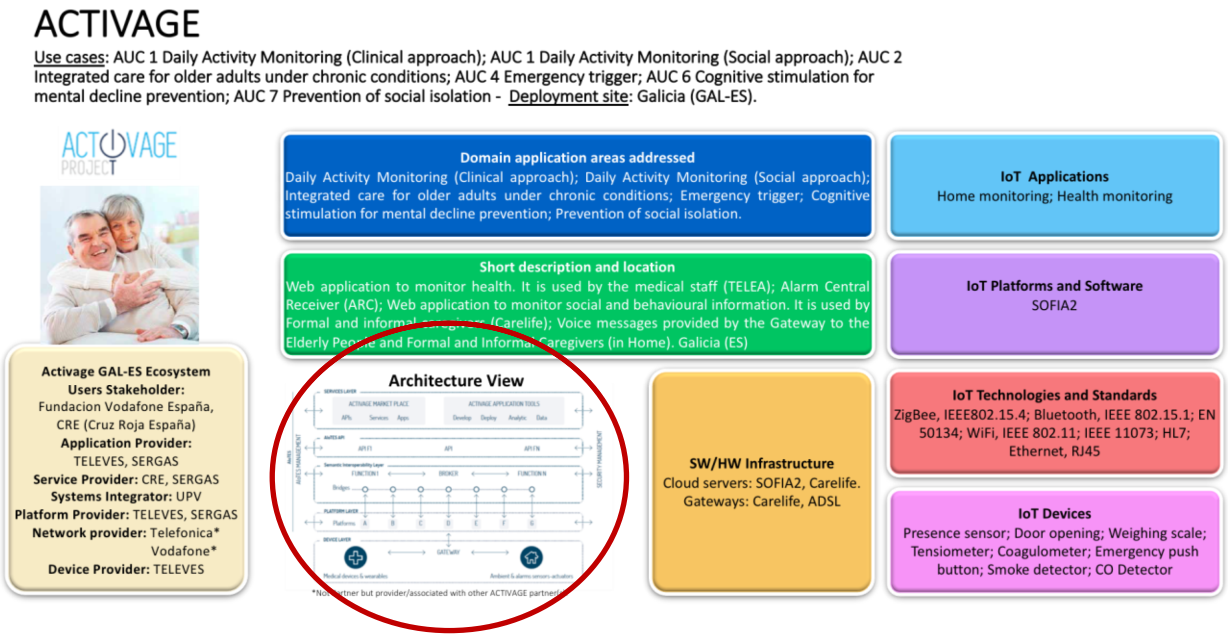
This sub-section is going to address the LSP Reference Architecture from the point of view of these use cases. This approach will allow for the identification of commonalities and differences both across the various Use Cases of a single LSP, as well as across several LSPs. Figure 4 is showing an example of use case template with the indication of the Architecture view deemed as most significant from the perspective of the Use Case designers and developers. This view may be common to all the Use Cases of one LSP but may also differ. In most cases, this will be functional view, but in a few cases, it may be a more system-oriented view.

4.3.1 ACTIVAGE

4.3.1.1 Use Cases and Reference Architecture

ACTIVAGE has developed (and documented) 9 different use cases as part of the Large Scale IoT Pilot (LSP) for ageing well. This use case shave been conceived as a unique opportunity to scale up the demand side considering strategically important to improve ageing well in the target populations.

The aim is to align, set-up, deploy and measure relevant Use Cases that provide a value for the users of the nine Deployment Sites across Spain, France, Italy, Germany, Greece, Finland and United Kingdom.



ACTIVAGE

Use cases: AUC 1 Daily Activity Monitoring (Clinical approach); AUC 1 Daily Activity Monitoring (Social approach); AUC 2 Integrated care for older adults under chronic conditions; AUC 4 Emergency trigger; AUC 6 Cognitive stimulation for mental decline prevention; AUC 7 Prevention of social isolation - Deployment site: Galicia (GAL-ES).

Domain application areas addressed
Daily Activity Monitoring (Clinical approach); Daily Activity Monitoring (Social approach); Integrated care for older adults under chronic conditions; Emergency trigger; Cognitive stimulation for mental decline prevention; Prevention of social isolation.

IoT Applications
Home monitoring; Health monitoring

Short description and location
Web application to monitor health. It is used by the medical staff (TELEA); Alarm Central Receiver (ARC); Web application to monitor social and behavioural information. It is used by Formal and informal caregivers (Carelife); Voice messages provided by the Gateway to the Elderly People and Formal and Informal Caregivers (in Home). Galicia (ES)

IoT Platforms and Software
SOFIA2

Activage GAL-ES Ecosystem
Users Stakeholder: Fundacion Vodafone España, CRE (Cruz Roja España)
Application Provider: TELEVES, SERGAS
Service Provider: CRE, SERGAS
Systems Integrator: UPV
Platform Provider: TELEVES, SERGAS
Network provider: Telefonica* Vodafone*
Device Provider: TELEVES

Architecture View

*Not partner but provider/associated with other ACTIVAGE partner(s)

SW/HW Infrastructure
Cloud servers: SOFIA2, Carelife.
Gateways: Carelife, ADSL

IoT Technologies and Standards
ZigBee, IEEE802.15.4; Bluetooth, IEEE 802.15.1; EN 50134; WiFi, IEEE 802.11; IEEE 11073; HL7; Ethernet, RJ45

IoT Devices
Presence sensor; Door opening; Weighing scale; Tensiometer; Coagulometer; Emergency push button; Smoke detector; CO Detector

Figure 4: Example of a Use Case template with the Architecture view

The 9 Use Cases are covered achieving specific challenges related to health preservation. In most cases they address more than one category of needs and contribute to solve more than one challenge. Within ACTIVAGE, "Ageing Well with IoT" is considered as the goal to extend healthy living years of older adults living independently and autonomously in their preferred environments by the massive adoption of IoT solutions.

The consequences are not only impacting positively at individual level in terms of dignity preservation, maintenance of quality of life, perception of better health, but also at the aggregated level as a population with expected benefits in terms of reducing the demographic and financial pressure of the healthcare systems, especially in European societies.

1. Daily activity monitoring
2. Integrated care
3. Monitoring assisted persons outside home
4. Emergency trigger
5. Exercise promotion
6. Cognitive stimulation
7. Prevention of social isolation
8. Safety, comfort and Safety at home.
9. Support for transportation and mobility

One of the most accepted measurement scales in different studies in Europe and World Wide is the Clinical Frailty Scale (Figure 5): ACTIVAGE will concentrate on IoT solutions for older people classified under categories 1 to 5. For category 6 "Moderately frail", the use of IoT solutions can positively impact the end users and their care network by preserving the independent living.

ACTIVAGE focuses in deploying IoT solutions that work towards keeping older people away from category 6 and beyond, which already represents a significant cost in care for informal carers and for the formal healthcare systems.

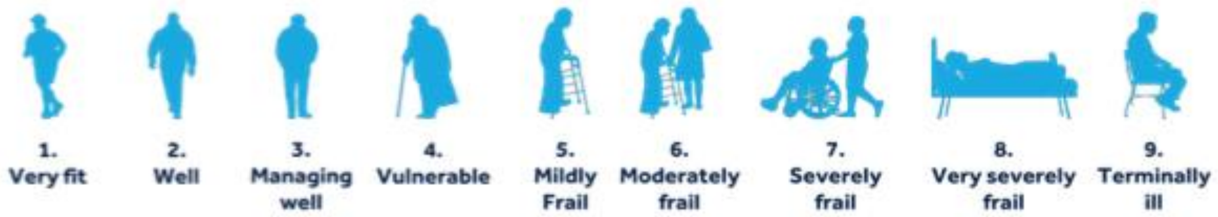


Figure 5: Clinical Frailty Scale, (Source: Dalhousie University [19]).

ACTIVAGE has developed (and documented) different overall interoperability use cases. The following interoperability use cases have been identified to highlight the technical interoperability functional requirements as well as to demonstrate the achieved "seamless" programmability of the devices or sensors in a connected experience.

1. Use Case 1 - Interoperability of Applications across Deployment Sites

The first use case portrays a scenario where an Active and Healthy Ageing (AHA) Application is built over ACTVAGE platform in order to be functional in multiple Deployment Sites (DSs) and respective platforms. This use case is a reverse approach compared to traditional interoperability use cases in which a top down approach is used. The main goals are that:

- Applications Data can be exchanged between multiple DSs.
- Applications can be developed for multiple DSs.
- Applications initially developed in a specific DS can easily be transferred to another one. Furthermore, developers can extend and exploit the products in more than one deployment site, only adapting their application to ACTIVAGE instead that adapting them to individual IoT platforms.

2. Use case 2 – Interoperability within Deployment Sites

This use case considers a single AHA Application in a single DS. However, different stakeholders within the DS may manage different platforms. The main goals are that:

- The Application data used in an IoT application can be used in another IoT application within the deployment site
- Applications within the DS can be developed for multiple stakeholders and/or platforms.
- Applications initially developed to run in a specific IoT Platform within the DS can be transferred to another IoT Platform(s) within the DS.

3. Use case 3 – Interoperability within Deployment Site, with a single gateway hosting

This use case considers a single AHA application in a single DS, and, for IoT protocol connectivity reasons at device layer, two different IoT platforms are hosted in the same gateway. ACTIVAGE can support this use case in two ways:

- First approach: make dedicated bridge between the two platforms. One of the two platform (A) is connected to ACTIVAGE, gathers all sensor data / sends all actuator commands and dispatches them to the platform B if it is the one connected to the device.
- Second approach: deploy ACTIVAGE in the gateway and use the interoperability bridges platform-ACTIVAGE and platform-ACTIVAGE to dispatch.

The Reference Architecture selected by the Use Cases is shown in Figure 6.

ACTIVAGE architecture, as shown in Figure 6 is designed to serve as common framework to build interoperable smart living solutions in the form of apps, software tools and services that can be deployed, extended and replicated at deployment sites across Europe. In other words, to allow future support of any additional platforms and services as far as they comply with defined interoperability framework and standards.

The design of this architecture is based on the analysis carried out in section three, where the challenges to be addressed in architecture are defined. Issues such as security, privacy, semantics, data processing and data sharing should be considered so as to develop the ACTIVAGE architecture. In addition to these gaps, it should be noted that the Deployment Sites architectures and the selected IoT Platforms are extremely heterogeneous therefore the architecture should efficiency and effectively integrate a wide spectrum of open and commercial platforms and IoT devices. ACTIVAGE architecture, described from top to bottom, will enable a set of tools and services and will provide them in form of a marketplace (section 5.1.1) where developers and entrepreneurs make use of the advantages of ACTIVAGE ecosystem. ACTIVAGE IoT Semantic Interoperability Layer plays an important role in the ACTIVAGE architecture. It enables and orchestrates the interconnection of heterogeneous IoT devices, European platforms and smart living services within a common ecosystem of solutions. It will enable application developers, integrators, service providers a common framework to build interoperable smart living apps and services that can be deployed, extended and replicated at deployment sites available across Europe.

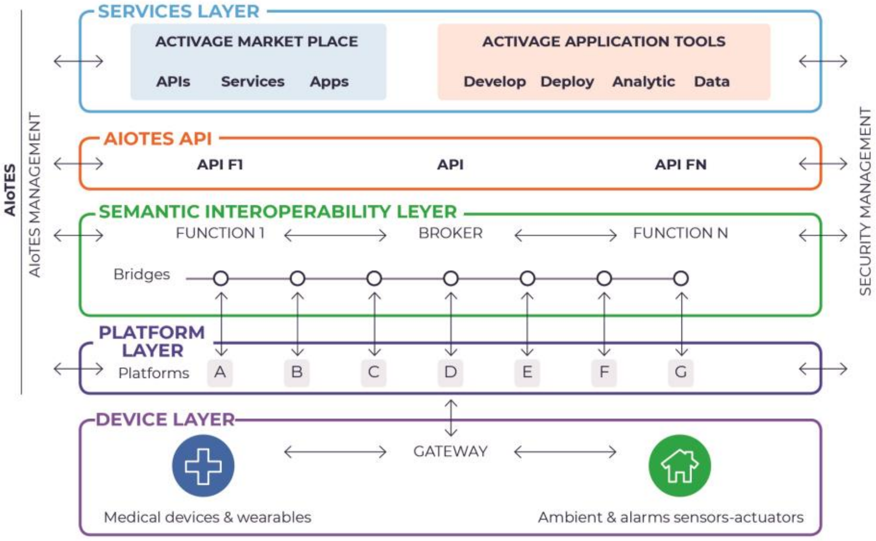


Figure 6: ACTIVAGE simplified Reference Architecture

- **ACTIVAGE Services Layer**
  The ACTIVAGE Services layer is composed by the Market Place and the Application Tools, The IoT marketplace is an open one-stop-shop that facilitates access to all the necessary applications and services that end-users need. ACTIVAGE Marketplace is a cloud-based environment for developers to publish their services and users to obtain, combine and deploy them to the environment of their applications. ACTIVAGE Applications tools focus on analytics, deployment and development of data services (Figure 7).
- **ACTIVAGE AIOTES API**
  An API is a set of clearly defined methods of communication between various software components. More specifically an API is a set of rules ('code') and specifications that software programs can follow to communicate with each other. In ACTIVAGE an API serves as an interface between different software programs and facilitates their interaction, like the way the user interface facilitates interaction between humans and computers. APIs are becoming extremely popular in the recent times as a pattern to overtake the interoperability limitations and particularly in the continuous creation and expansion of IT services. Other important driver of the API popularity is the creation of application-centric ecosystems, allowing third parties to create clients, use data and add value to the features of existing systems.
- **ACTIVAGE Semantic Interoperability Layer**
  Internet of Things (IoT) research and industry communities have realized that a common IoT problem to be tackled is the interoperability of the information. In ACTIVAGE we reviewed recent trends and challenges on interoperability, and discuss how often and how flexible

semantic technologies, open the services frameworks and tenable their information models to support. Extensible discussed the Internet of Things (IoT) refers to things (objects) and the virtual representations of these objects on the Internet. IoT Interoperability then shall define how the things "talk" amongst other things and communicate with other systems in order to expose their capabilities and functionalities "services".

Interoperability through the platforms presented in ACTIVAGE is carried out by means of the IoT Interoperability Layer (Figure 8). Semantic Interoperability in the broader sense has been defined in section four, at this point we just focus on describing the components of the interoperability layer which, by definition, is an abstraction layer that allows the communication between an application of the marketplace and the ACTIVAGE platform.

- Communications are managed by a message broker which participates in every communication in IoT Semantic Interoperability Layer. A general API is used to access the broker that exposes basic common operations (message pub/sub, topic creation, basic resources management...), enabling interchangeability of the actual implementation of the broker.

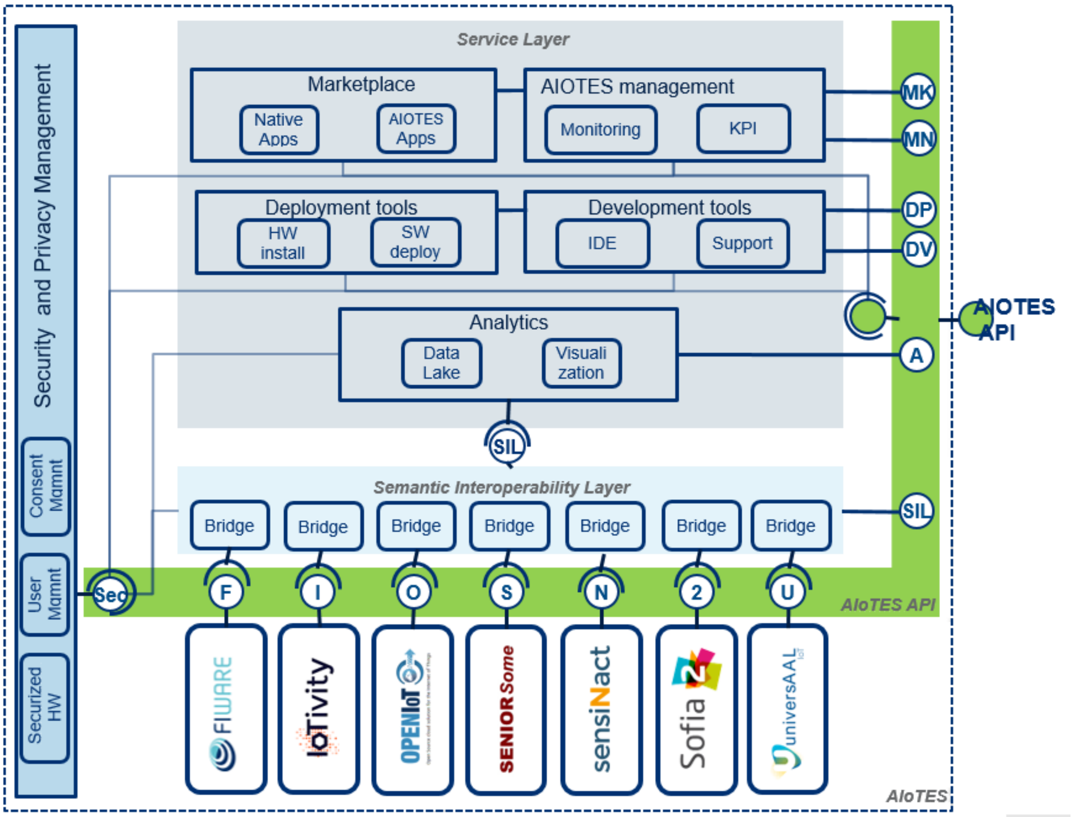


Figure 7: ACTIVAGE Services Layer Functional Blocks

- **Device Layer**
  The Device Layer is the Foundation layer of an IoT System. This layer is in charge to perform sensing information from the Physical world and actuating to modify the behavioural of the connected objects. At higher abstraction layers (Cloud, Application), the gathered data is analysed and used to monitor the overall system and to facilitate decision taken. In some systems, these decisions could imply control decisions, i.e. parameter changes in the connected objects via the actuator devices, in order to close the control loop with the Physical world.
- **ACTIVAGE IoT Ecosystem Suite (AIOTES) Management**
  The ACTIVAGE IoT Ecosystem Suite is a set of Techniques, Tools and Methodologies for interoperability at different layers between heterogeneous IoT Platforms and an Open Framework for providing Semantic Interoperability of IoT Platforms for AHA, addressing

trustworthiness, privacy, data protection and security. More concretely, as it can be seen in Figure 59 AIOTES groups together most of the building blocks of the architecture, namely, Marketplace, API, Interoperability Layer, Platform Layer, AIOTES Management and Security Management Framework.

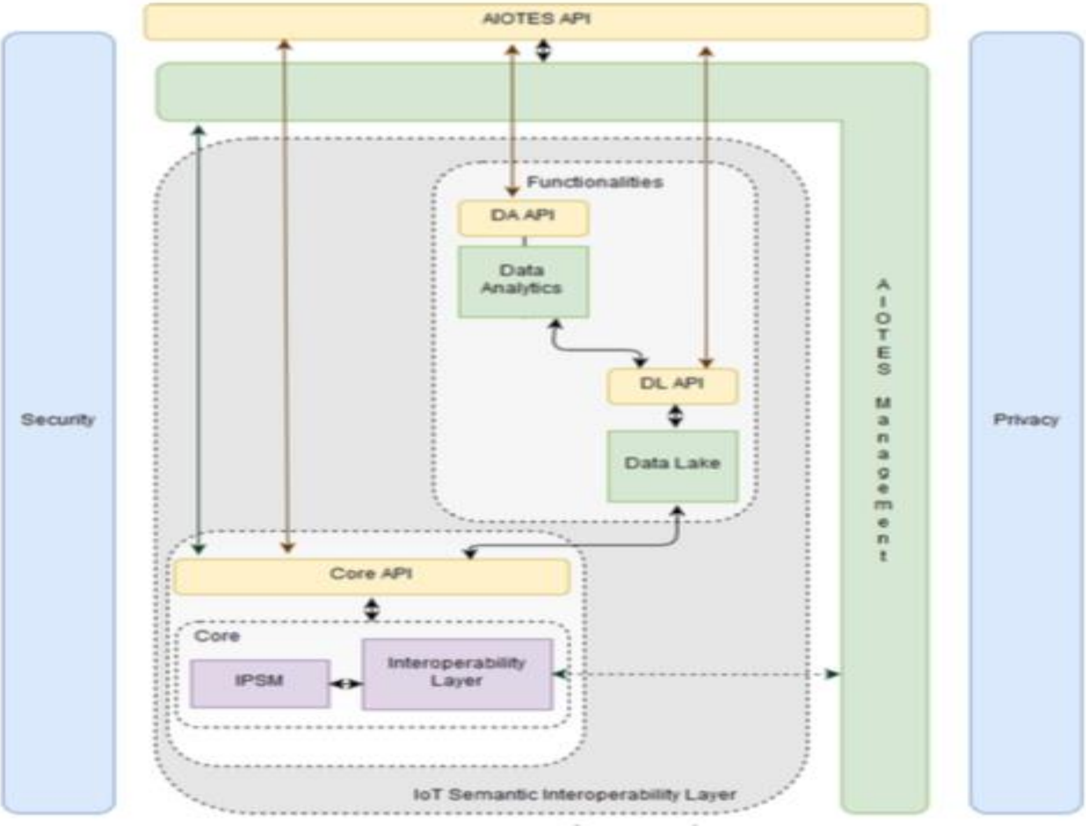


Figure 8: ACTIVAGE Semantic Interoperability Layer (SIL) Functional Blocks

- **Security Management Framework**
  The security management framework aims at providing a trustful digital environment to enable confidence in the global architecture. This confidence is reflected in the ability to guarantee four qualities:
  - Confidentiality: Only the legitimate (or possessive) recipient of a data block or message may have an intelligible view of it.
  - Authentication: When sending a data block or message or when connecting to a system, the identity of the sender or the identity of the user is known.
  - Integrity: It guarantees that a block of data or a sent message has not been modified, accidentally or intentionally.
  - Non-repudiation: The author of a block of data or a message cannot deny his work.

  The security management framework will be based on the principle of a Public Key Infrastructure (PKI). PKI is a set of technical and organizational means to establish a strong guarantee of confidence in the validity of a digital identity. These means will allow to have a common security understanding in the project.

4.3.1.2 Comments

The project, has worked towards implement the idea of innovating the Active and Healthy Ageing (AHA) sector by both, designing solutions introducing IoT technology and by using IoT technology to improve current healthcare services. One of the objectives is to build the first European IoT ecosystem across 9 Deployment Sites (DS) in seven European countries. The ecosystem must be replicable by means of reusing underlying open and proprietary IoT platforms and scaling up software technologies and promote standards. ACTIVAGE aims to integrate new necessary interfaces for enabling interoperability across heterogeneous platforms and the project's ecosystem will set the foundations for connecting providers, service providers, and users.

ACTIVAGE project focuses on the need to overcome a fundamental barrier and technical limitations of IoT ecosystems: the lack of interoperability across IoT platforms and things. In today's society where people rely in the Internet as one of the most powerful communication, today we are dealing with vertically oriented and mostly closed systems. IoT architectures are built on heterogeneous standards or even proprietary interfaces. This causes interoperability problems when developers aim to create cross-platform and cross-domain applications, hiding the creation of broadly accepted IoT ecosystems. Service and application providers who want to use services in the ecosystem need to adapt to the platform-specific API and information models and therefore, only can provide application and services for a small number of platforms creating barriers and limiting business opportunities. This concerns especially for small innovative enterprises, which cannot afford to provide solution across multiple platforms. In addition, ACTIVAGE not only focuses on providing interoperability through the previously mentioned European platform but also the architecture that should be able to support other future platforms and services, providing thus a common framework for allowing to deploy IoT services in any Deployment Site platforms that comply with the standards of the architecture.

4.3.2 AUTOPILOT

4.3.2.1 Use Cases and Reference Architecture

AUTOPILOT has developed (and documented) 14 Use Cases [8]. The Reference Architectures in support of these Use Cases are discussed below.

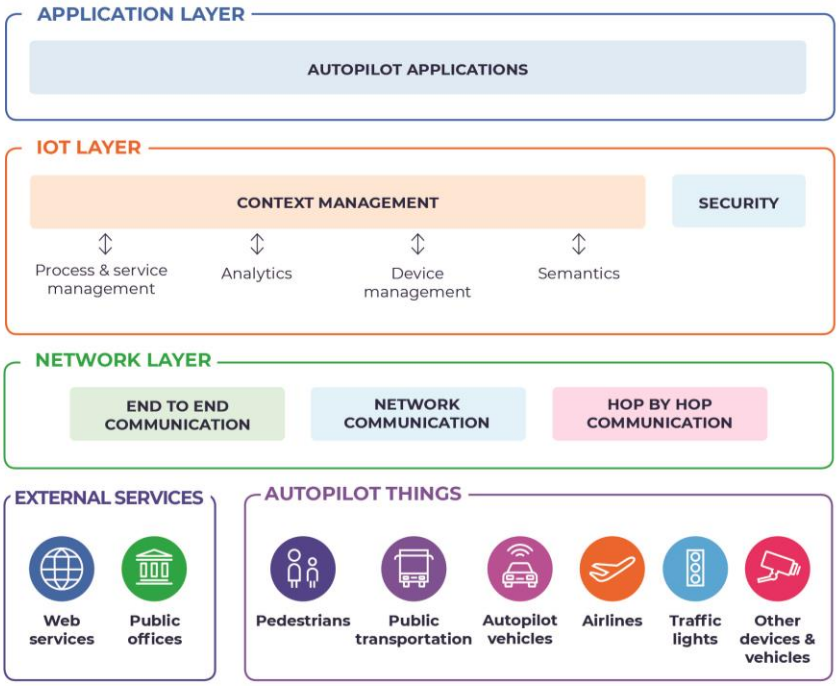


Figure 9: AUTOPILOT reference architecture

The Reference Architecture selected by most of the Use Cases is shown in Figure 9. It is developed to leverage autonomous driving (AD) and innovative mobility services based on open IoT platforms.

The AUTOPILOT architecture is used as a common framework to realise IoT-based automated driving use cases and consists of a set of services that have various capabilities including processing, communication, resource management, context management, and security.

The architecture is composed of an applications layer, an IoT layer, a network layer and an external service (e.g. web services) as well as IoT devices layer.

Within the IoT layer, open IoT platforms based on interoperable and federated models support IoT applications and services of the different use cases and the European pilot sites.

The AUTOPILOT IoT architecture (Figure 10) builds on and borrows building blocks from, relevant IoT architectures such as AIOTI [20] and IoT-ARM [21]. The development of the IoT architecture follows an incremental approach.

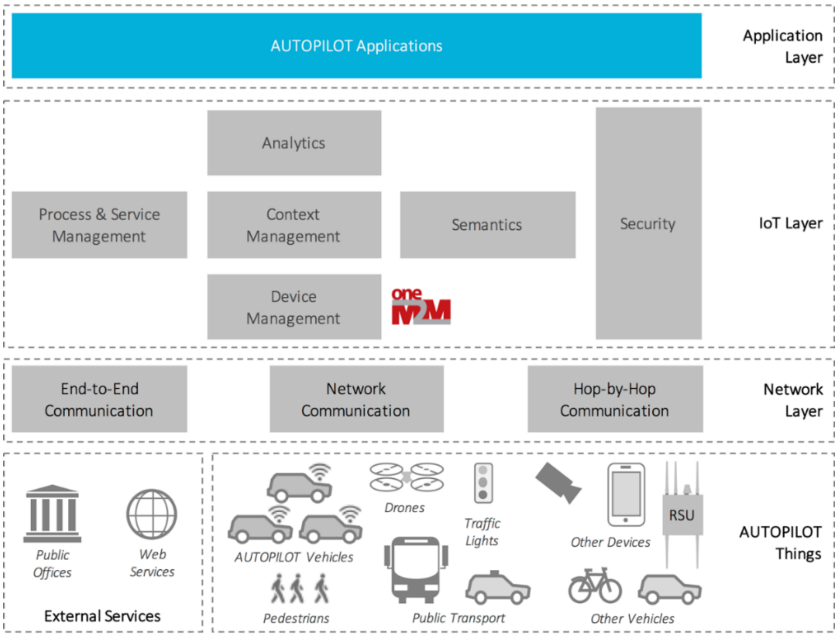


Figure 10: AUTOPILOT IoT Architecture: Functional view

The AUTOPILOT target IoT architecture aims to provide a global IoT service coverage through features such as **openness**, **flexibility**, **interoperability** between IoT platforms, leveraging of **standards** for communication and interfacing and **federation** of in-vehicle, road-side unit and pilot site IoT platforms.

As shown in Figure 10, the AUTOPILOT IoT architecture has four main functional layers:

- **Things Layer:** Includes the AUTOPILOT "things" (vehicles, cameras, drones, road side units, etc.) and external services provided by public offices or private web services.
- **Network Layer:** Enables communication throughout the IoT ecosystem.
- **IoT Layer:** Enables the IoT functionality through a set of IoT building blocks: *device management, context management, process & service management, semantics, analytics and security*. Each of these functional building blocks is specified in detail in the AUTOPILOT deliverable D1.3.
- **Application Layer:** Contains services that leverage the AUTOPILOT IoT. In AUTOPILOT, this includes services provided by the use cases to the AD vehicles and users (e.g. drivers, car sharing customers, etc.).

Given that AUTOPILOT has several large-scale pilot sites, the architectural components of the open IoT platform (infrastructure, IoT devices, services, etc.) are inherently physically distributed. AD functions themselves have varying requirements in terms of speed of access (latency), availability and range (covered area). While some localised mission critical functions, such as warning other vehicles in the immediate proximity that a pedestrian is jaywalking, need to be accessible within very low latency.

Other functions, such as notification about a parking spot being made available, need to cover wider areas but are less demanding in terms of latency. As a result, the AUTOPILOT IoT platform was designed and implemented as a federation of IoT platforms.

On top of the above reference architecture provided by most of the Use Cases, a few Use Cases have chosen to present a more "system"-oriented Reference Architecture as in the two examples shown in Figure 11.

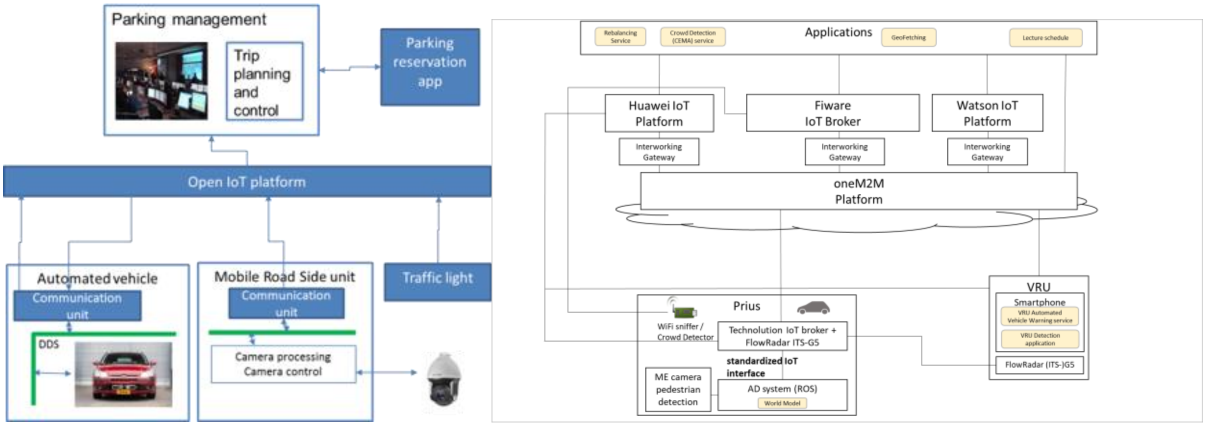


Figure 11: Two examples of System Architectures in AUTOPILOT Use Cases

As a concrete prototype example, Figure 12 shows the in-vehicle high-level architecture for the AUTOPILOT Brainport test site in The Netherlands and includes concepts and components that are common to the prototypes [12]. The architecture conceptually always separates safety-critical and non-safety-critical components to allow critical components in the vehicle to work even in the event of communication loss with external entities (e.g. IoT services or V2X communication). Safety-critical components address functionalities in the operational level of the vehicle, thereby requiring high reliability and timely updates.

Vehicle autonomous driving (AD) application contains the operational vehicle planning and control that will be specific to each vehicle provider and use case. In-vehicle sensors are connected to the system via in-vehicle communication (e.g. CAN bus, Ethernet) and will also be vehicle provider specific. Examples of internal sensors are cameras, radars, motion sensors, etc.

Non-safety-critical components address functionalities in the tactical and strategic levels of the vehicle, thereby posing less strict time and reliability requirements. IoT applications will consume and process data coming from external entities such as IoT Cloud services, other vehicles in the surroundings (V2X), Roadside units (RSUs) and Vulnerable Road Users (VRUs) via external communication links such as Cellular LTE or V2X ITS-G5 communication.

The "Vehicle world model" component creates a high-level view of the surroundings that can be used by either the Vehicle AD or IoT applications in both safety-critical and non-safety-critical levels [12].

This component will combine and fuse data coming both from in-vehicle sensors to create the "Ego world model" and external entities such as IoT cloud services via the IoT broker, roadside units or other vehicles (V2X), to create the "Shared world model". Depending on applications' requirements, the output is one or more application specific vehicle world models.

Each world model provides high-level description of objects (e.g. shape of cars, pedestrians), road/lane (e.g. road shape) and optional semantics information (e.g. classification of objects). This allows, for example, operational path planning algorithms to make the best decision at a certain point in time based on an extended view of the environment that is currently relevant to the ego vehicle.

In the event of communication loss with the external entities, the vehicle world model component will rely solely on the in-vehicle sensor data to create the world model that would correspond in this case to the "ego world model".

In this manner, the overall system benefits from external data when available to extend its range of awareness and yet it remains robust against communication failure.

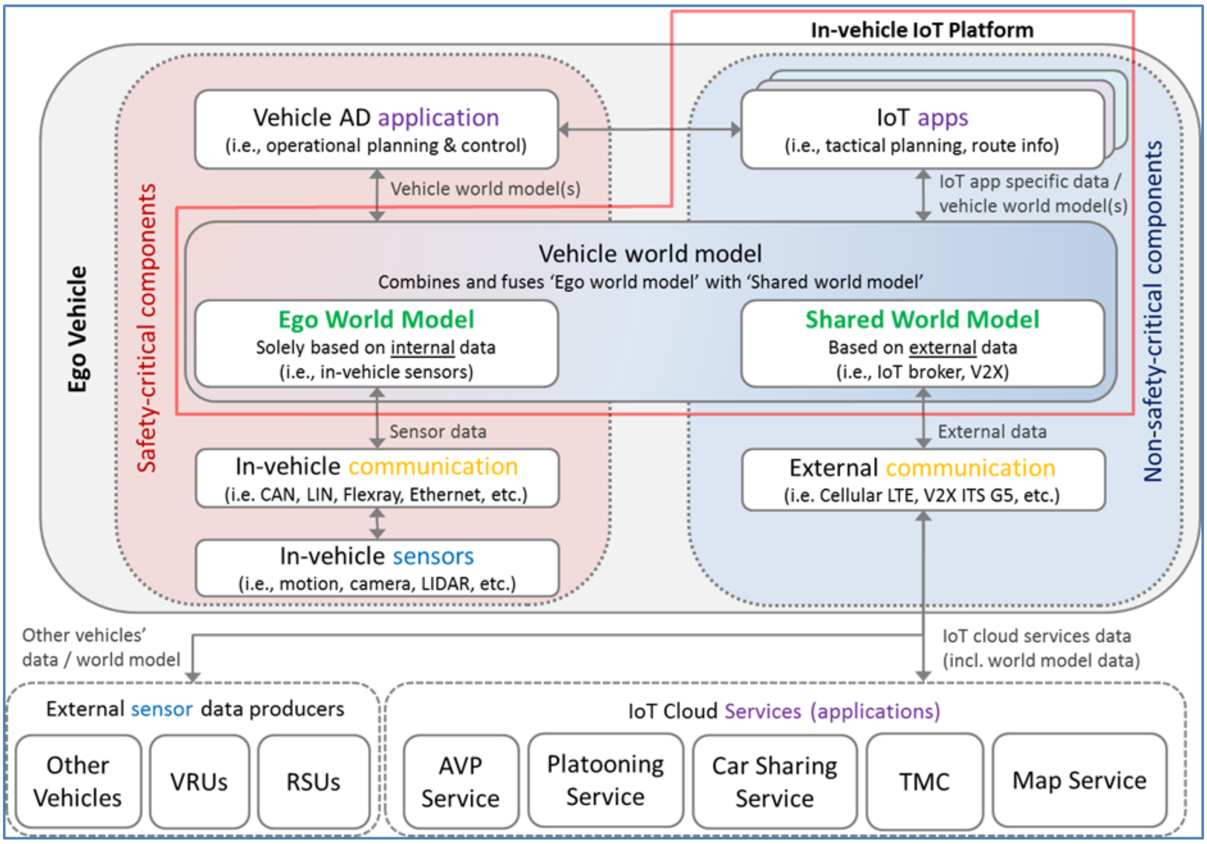

In-vehicle IoT Platform
Ego Vehicle
Safety-critical components
Non-safety-critical components
Vehicle AD application
(i.e., operational planning & control)
IoT apps
(i.e., tactical planning, route info)
Vehicle world model(s)
IoT app specific data / vehicle world model(s)
Vehicle world model
Combines and fuses 'Ego world model' with 'Shared world model'
Ego World Model
Solely based on internal data
(i.e., in-vehicle sensors)
Shared World Model
Based on external data
(i.e., IoT broker, V2X)
Sensor data
External data
In-vehicle communication
(i.e. CAN, LIN, Flexray, Ethernet, etc.)
External communication
(i.e. Cellular LTE, V2X ITS G5, etc.)
In-vehicle sensors
(i.e., motion, camera, LIDAR, etc.)
Other vehicles' data / world model
IoT cloud services data (incl. world model data)
External sensor data producers
Other Vehicles
VRUs
RSUs
IoT Cloud Services (applications)
AVP Service
Platooning Service
Car Sharing Service
TMC
Map Service

Figure 12: AUTOPILOT in-vehicle high-level architecture for Brainport pilot site [12]

In the AUTOPILOT project, the "in-vehicle IoT platform" comprises the "Vehicle world model" and IoT applications which together build the bridge from the in-vehicle system to the external IoT world [12]. The high-level architecture above intentionally hides components that might be vehicle provider specific, such as high- and low-level controllers that can be specified in different ways within the "Vehicle AD application". Also, the use of a standardized IoT broker such as the oneM2M platform is conceptually grouped into the "Shared world model" sub-component that is gathering data from external entities.

The AUTOPILOT functional architecture of the vehicle IoT platform is illustrated in Figure 13 [12]. This architecture is a high-level decomposition of the vehicle IoT platform into major components which aim to accomplish the general basic functionalities addressed in the project.

The AUTOPILOT architecture components can be classified as in-vehicles components and external components and are described below.

In-vehicles Components [12]:

- **In-vehicle IoT applications**: Consume and process application specific data via IoT broker and can interact with other components in the system (e.g. world model). According to the type of application, they can cover data management functionalities and/or interface for the final user (UI).
- **IoT broker**: Connects vehicle with other IoT brokers in the cloud, edge or in other vehicle/roadside units. Since it is directly related to the communication with external applications and other systems, the IoT Broker is the interface between the vehicle and the eternal world (cloud, edge, another vehicle).
- **IoT devices**: Other devices (in the edge, cloud) that are connected to the car. Software modules implement drivers to virtualize such physical IoT devices (sensors and actuators) into vehicle IoT platform, in such a way to satisfy the IoT device adaptation functionality.
- **IoT bridge**: Connects the IoT Broker, and therefore the IoT world, to the development environment inside the car. It takes care of the exchange of data between IoT and non-IoT

components. Considering its bridge position between the internal components and interface toward the external world, this component fulfils part of the APIs functionality, and satisfies the syntactic Interoperability functionality. In some cases, the IoT bridge and the IoT broker can be merged into a single component.

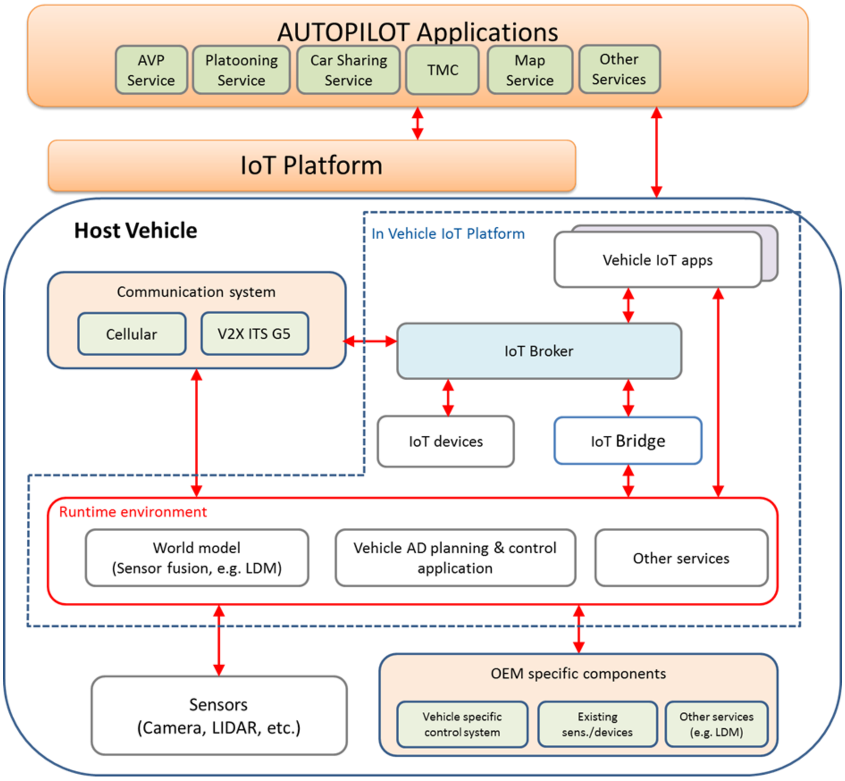


Figure 13: AUTOPILOT functional architecture of the vehicle IoT platform[12]

- **World model**: Creates a high-level view of the surroundings that can be used by planning/control applications and IoT apps. The vehicle world model will combine and fuse data coming both from internal sensors and external entities such as the IoT cloud services via the IoT broker, the roadside units or the other vehicles (V2X). The vehicle world model component will include a high-level description of objects (e.g. shape of cars, pedestrians), road/lane (e.g. road shape) with optional semantics information (e.g. classification of objects). This will allow the high-level path planning and control to make the best decision at a certain point in time. Context-awareness functionality is satisfied with such a type of architectural component.
- **Vehicle path planning and control**: High-level planning and control that can leverage IoT data to improve its functionalities (e.g. global route or speed advice coming from the IoT apps). This will be complementary to the already present low-level control and actuators functions in the vehicle. Data Management functionality deals with the collection of information from external elements to the vehicle (i.e. cloud, RSU, other vehicles and infrastructures), exploiting data in order to create services such as planning, and control application related to AD system.
- **Other services**: Basically, anything that can be used to support the car functionalities but that is not directly connected to the planning/control or WM (e.g. traffic light recognition, license plate identification, Vehicle Platform configuration and Remote Management, etc.).
- **Sensors**: Refers to sensors that are not OEM specific (e.g. MAP, MAF, lambda, etc.) and whose purpose is usually associated with AD functions (stereo cameras, LIDAR, etc.). Software modules implementing drivers to adapt and virtualize such sensors are needed too
- **OEM specific components**: Relates to components that are OEM specific such as actuators for power steering and brakes, inputs to gearbox, or vehicle sensors needed for the "normal" vehicle functions (MAP, MAF, ABS, etc.). Software modules implementing drivers to

virtualize such OEM specific components into vehicle IoT platform are needed, in such a way to satisfy the OEM systems communication functionality.

- **Communication system**: The components that provide communications to the outside. Be it a simple IP based cellular network or V2X ITS-G5. These communication media can provide information of the outside world or can send information to the outside world.

External Components [12]:

- **AUTOPILOT applications**: Interface the IoT Platform and implement AUTOPILOT function in the cloud. Each application communicates with the vehicle via the IoT platform. An application can also comprise a component that runs in the vehicle platform. These components can be either an IoT application or an In-vehicle application, depending on the level of integration with the IoT platform.
- **IoT Platform**: Implements the IoT functions at the cloud or edge level. It comprises also other vehicles and roadside elements.

4.3.2.2 Comments

The AUTOPILOT project monitors several standardisation activities. A short summary of the activities can be found below. For further information, refer to AUTOPILOT deliverable D5.7 (Standardisation plan) [13].

ETSI TC ITS (Intelligent Transport Systems) activities are of special interest, WG1 (User and Application requirements) in particular. Areas involved are communication network, vehicle IoT integrations and platforms, IoT ecosystems, IoT platforms, security architectures, vehicle IoT testing, and test specifications.

Regarding ETSI TC Cyber, the security architecture, described in standards, are used as guideline in the AUTOPILOT architecture design phase, with reference to IoT architecture and specification, and to security, privacy and data specification. Areas involved are IoT platform, vehicle IoT integration and platform, communication network, and IoT ecosystem.

ETSI TC ERM (EMC and Radio spectrum Matters); several AUTOPILOT test sites consider the usage of vehicle based on-board units (OBU) and road-side units (RSU) equipment. DSRC is a well-established and stable communication technology between road-side equipment and moving vehicles, which can be used for reference for communication purposes (ETSI EN 300 674-2-1/2).

The overall AUTOPILOT concept is based on 4G/5G and LTE-V2X in combination with G5 technology of road side units placed along ITS application corridors/roads. With regards to the AUTOPILOT test sites, special safety issues taking extreme test source voltages and regulated lead-acid battery power sources used on vehicles must be considered, so that accident risks are minimized (ETSI EN 302 571). Areas involved are mobile communication network.

ETSI ISG CIM is the industry specification group on context information management, developing an API using the OMA (Open Mobile Alliance) next generation service context interfaces) as a starting point. For the standards being developed, it is important to specify the content information model, and AUTOPILOT will likely use and influence the area of ITS and automated vehicles. Areas involved are IoT platforms based on FIWARE and OneM2M.

Concerning CEN TC278, the most relevant WGs for AUTOPILOT are WG15 (eSafety/eCall) and WG16 (Co-operative systems). Areas involved are vehicle IoT integrations and platforms, and communication networks.

Within IEEE, 802.11p, 802.15, 802.20 and P1609 are of special interests. Areas involved are vehicle IoT integrations and platforms, communication networks, IoT ecosystems, and IoT platforms.

The IEC standard family for security for industrial automation and control systems (ISA/IEC-62443) is addressing policies and procedures versus functional requirements and is relevant as references for actors involved in developing products associated with IoT. Areas involved are IoT platforms, vehicle IoT integrations and platforms, communication network, and IoT ecosystems.

Regarding ISO and IEC, the ISO/IEC JTC1/SC27 is responsible for the development of international standards and technical reports/specifications within the field of information and IT security. The ISO/IEC 27000 family standard is widely adopted to manage security related aspects for information systems, and AUTOPILOT reference it as a possible baseline for the overall IT security management by the service suppliers.

The ISO/IEC 15408-1 is evaluation criteria for IT security and is used as a guide for the AUTOPILOT architecture, particularly for security, privacy and data specification, and for IoT architecture and specification. Areas involved are IoT platforms, vehicle IoT integrations and platforms, communication networks, and IoT ecosystems.

The ISO TC204 is responsible for the overall system aspects and infrastructure aspects of intelligent transport systems (ITS) as well as the coordination, including the schedule for standards development, considering the work of existing international standardization bodies.

Vehicle Control Systems (WG14) is of special interest due to activities on standardizing performance requirements and test procedures for many of the new ITS features in cars, such as automatic parking, intelligent cruise control, backing-up aid, lane departure warning, collision warning, etc. Areas involved are vehicle IoT integrations and platforms, and communication networks.

The third-generation partnership project (3GPP) covers cellular telecommunications network technologies, including radio access, the core network, and service capabilities (including work on codecs, security, quality of service) and thus provides complete system specifications.

The specifications also provide hooks for interworking with non 3GPP radio accesses, such as Wi-Fi. The overall AUTOPILOT concept is based on 4G/5G and LTE-V2X, these communication technologies are specified in the 3GPP Global Initiative.

5G is still under specification procedure with regards to standardization, nevertheless narrow-band IoT will play a significant role in parking services planned in AUTOPILOT sites and other ITS field trials due to weak battery life time.

Therefore, all mobile communication technologies are considered related to narrow-band IoT (NB-IoT) sensor availability and base station upgrades.

NB-IoT is a promising brand-new standard which can be considered by AUTOPILOT architecture during the design phase.

Security aspect for LTE support of V2X services are used as a guideline in the AUTOPILOT architecture phase, with reference to the security aspect. Areas involved are vehicle IoT integrations and platforms, IoT platforms, communication networks, and IoT ecosystems.

4.3.3 IoF2020

4.3.3.1 Use Cases and Reference Architecture

IoF2020 has developed (and documented) 19 Use Cases. The Reference Architectures in support of these Use Cases are discussed below.

The holistic approach developed by IoF2020 for the production and usage of the Reference Models and associated information is shown in Figure 14:

- At Use Case level, an architecture is refined into a system implementation
- At Project level, the IoT Reference Architecture is complemented with:
  - An IoT Catalogue of reusable components
  - An IoT Lab gathering configurations and instances

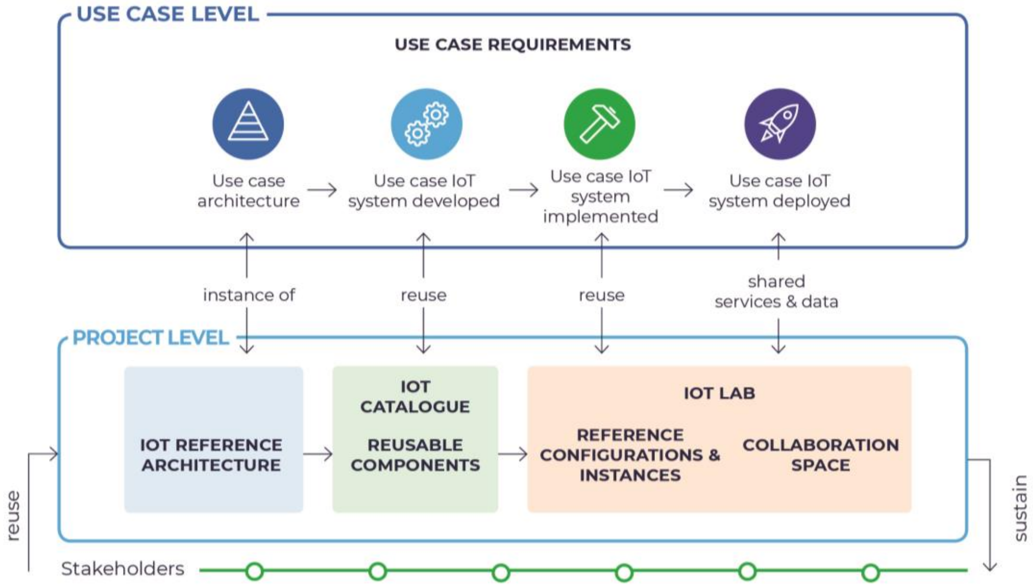


Figure 14: The IoF2020 holistic approach to Reference Models

IoF2020 has adopted a Functional Architecture that refers to ITU-T Y.2060. Figure 15 shows the empty functional template that is filled by each Use Case with a set (in general different from one Use Case to another one) of functions and sub-functions in the respective "horizontal" domain layers or "vertical" cross-cutting capabilities

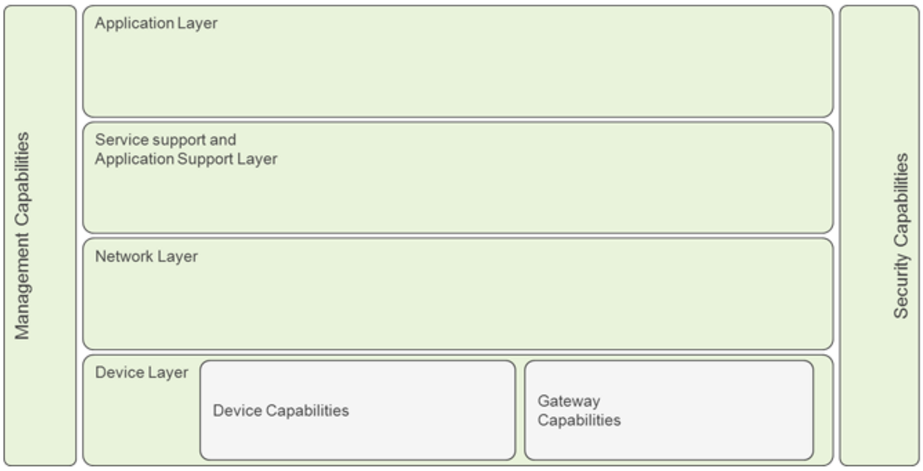


Figure 15: The Functional Model template in IoF2020

The IoT functional view classifies the role of each component from the IoT point of view. It serves the purpose of placing each component of the UC within a categorization suitable to understand of what the most suitable provider of infrastructure or technology is suitable to offer such component.

4.3.3.2 Comments

Within the IoF2020 project, in order to better align with on-going IoT trends and standardization efforts, as well as following the recommendation by AIOTI WG03, the information regarding each Use Case was done by depicting all the main functionalities within the ITU-T Y.2060 IoT Reference Model.

In IoF2020, each UC was specified by defining and analysing a minimal set of architectural views. Such views include: a domain model; a deployment view; an "IoT" Functional view; a "Business Process Hierarchy" valid for the agri-food domain; a description of the Interoperability Endpoints;

an Information model; a summary of gaps; a selection of assets identified for re-use; a Security, Privacy and Trust Analysis.

At the same time, IoF2020 embraced a demand-driven methodology in which end-users from agri-food are actively driving the entire development process, aiming at cross-fertilisation, co-creation and co-ownership of results. All innovative IoT technologies initially selected in the use cases have a value proposition for end-users. Nevertheless, their development was possibly based on a large set of market and technical assumptions that have not been tested in a systematic way in their operational environment. Therefore, IoF2020 applied an innovative approach, the lean multi-actor approach to test these assumptions in real bottlenecks of end-users in their operational environment and with value chain stakeholders, using so called MVPs (Minimum Viable Products). Feedback is translated into technical improvements that better meet end-user needs and better fit into the production environment and the value chain.

4.3.4 MONICA

4.3.4.1 Use Cases and Reference Architecture

MONICA has developed (and documented) the following 4 Use Case Groups, namely:

1. Sound Monitoring and Control
2. Crowd and Capacity Monitoring and Management
3. Missing Persons/Locate Staff Members
4. Health/Security Incidents

The Reference Architecture, shown in Figure 16, in support of these Use Cases Groups is discussed below. The MONICA IoT platform consists of control systems which monitor the data collected and which can perform automated actions. Components analyse data and detect critical incidents, supporting operators in assessing the situation and making decisions. To ensure data security and trust, the solution is built on Privacy by Design principles.

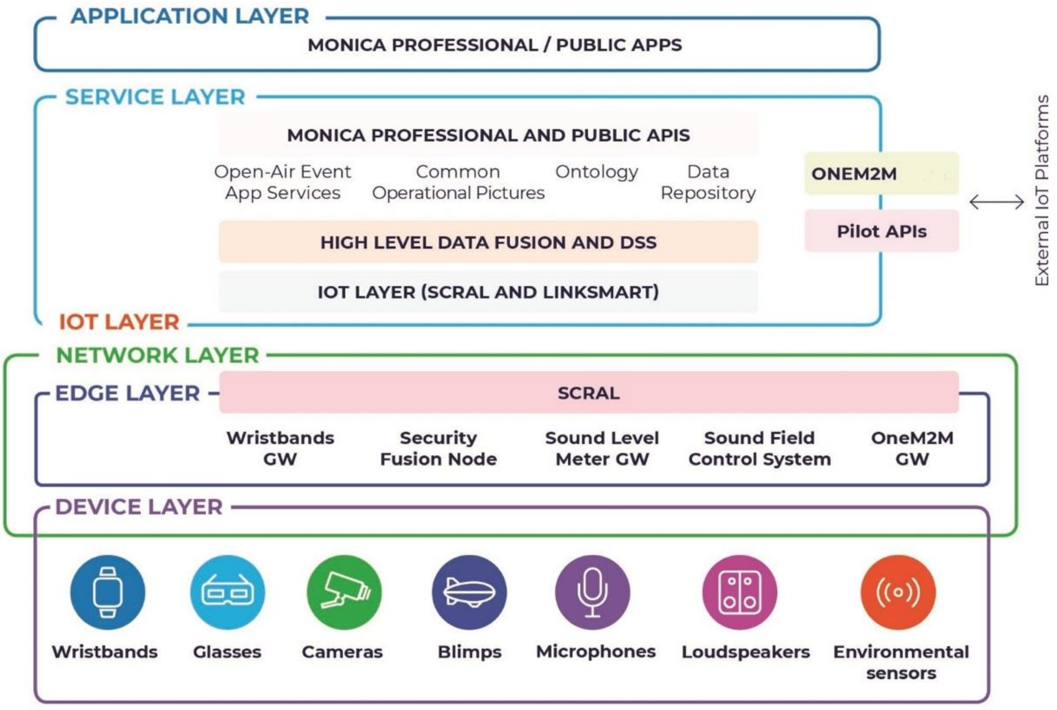


Figure 16: The MONICA simplified Reference Architecture

Based on open standards and architectures, the platform can be incorporated with existing smart city systems, replicated to fit other settings or used to develop new smart city applications. To This aim the oneM2M modules have been incorporated in the MONICA platform.

The MONICA architecture has been defined following the HLA developed by the Working Group 3 (WG03) of the AIOTI. The WG03 last updated the HLA in June 2017, which at the same time is described using the ISO/IEC/IEEE 42010 standard (ISO/IEC/IEEE 42010, 2011). Related documents to the HLA defined by the WG03 are available at [18].

The functional model of AIOTI is composed of three main layers:

- The **Application layer** containing the communications and interface methods used in process-to-process communications.
- The **IoT layer** that groups IoT specific functions, such as data storage and sharing, and exposes those to the application layer via interfaces commonly referred to as Application Programming Interfaces (APIs). The IoT Layer makes use of the Network layer's services.
- The **Network layer**, which services can be grouped into data plane services, providing short and long-range connectivity and data forwarding between entities, and control plane services such as location, device triggering, QoS or determinism.

Figure 17 depicts a mapping of the current MONICA Architecture with the reference HLA defined by the WG3 of the AIOTI [18].

As it can be seen, the MONICA Network Layer and Edge Layer are seamlessly mapped with the Network layer of the reference HLA, enabling interconnections between heterogeneous networks and providing device connectivity to the Internet via different network technologies.

The MONICA IoT Layer, which includes the Adaptation Layer and the Middleware Layer, is mapped into IoT entities, since they provide IoT functions to App Entities or other IoT Entities via LinkSmart.

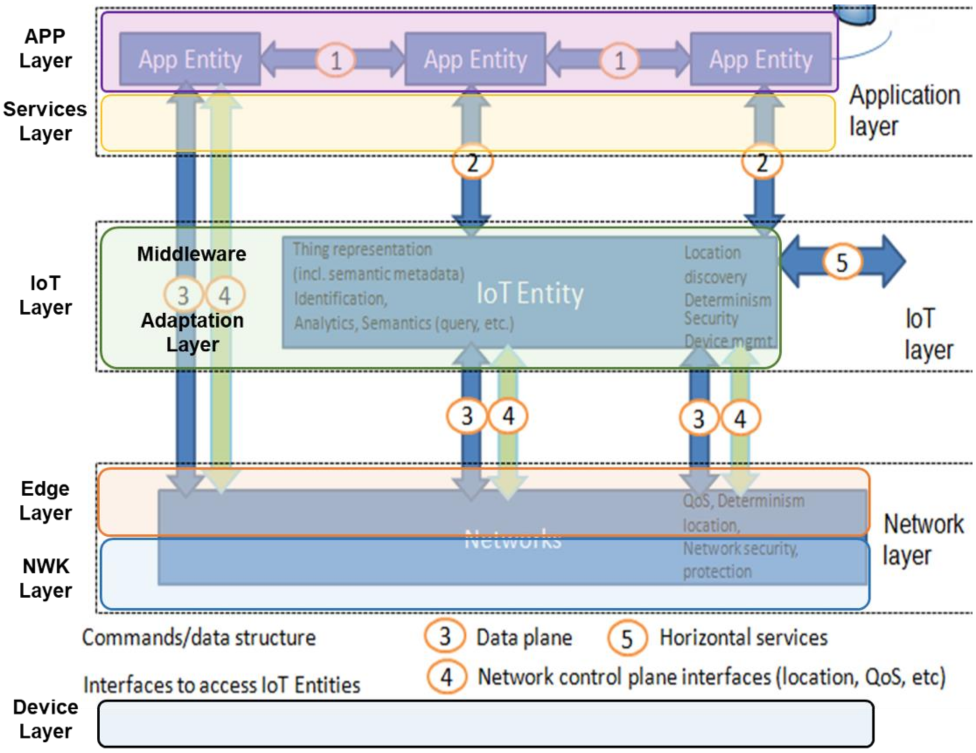


Figure 17: MONICA HLA Mapping with AIOTI HLA.

This layer uses the underlying Networks' interfaces to send and receive data from the physical devices (Device Layer of the MONICA architecture) that are not aware of the IoT world. The AIOTI HLA scheme does not explicitly include a Device Layer but this can be considered as part of its Network Layer while MONICA HLA explicitly represents it with a dedicated layer to highlight its importance within the project.

Finally, the MONICA's APP Layer and API's layer are mapped into the AIOTI HLA APP Entities since they implement - and enable - application logics.

4.3.4.1.1 Architectural focus

The current version of the MONICA architecture is depicted in Figure 18.

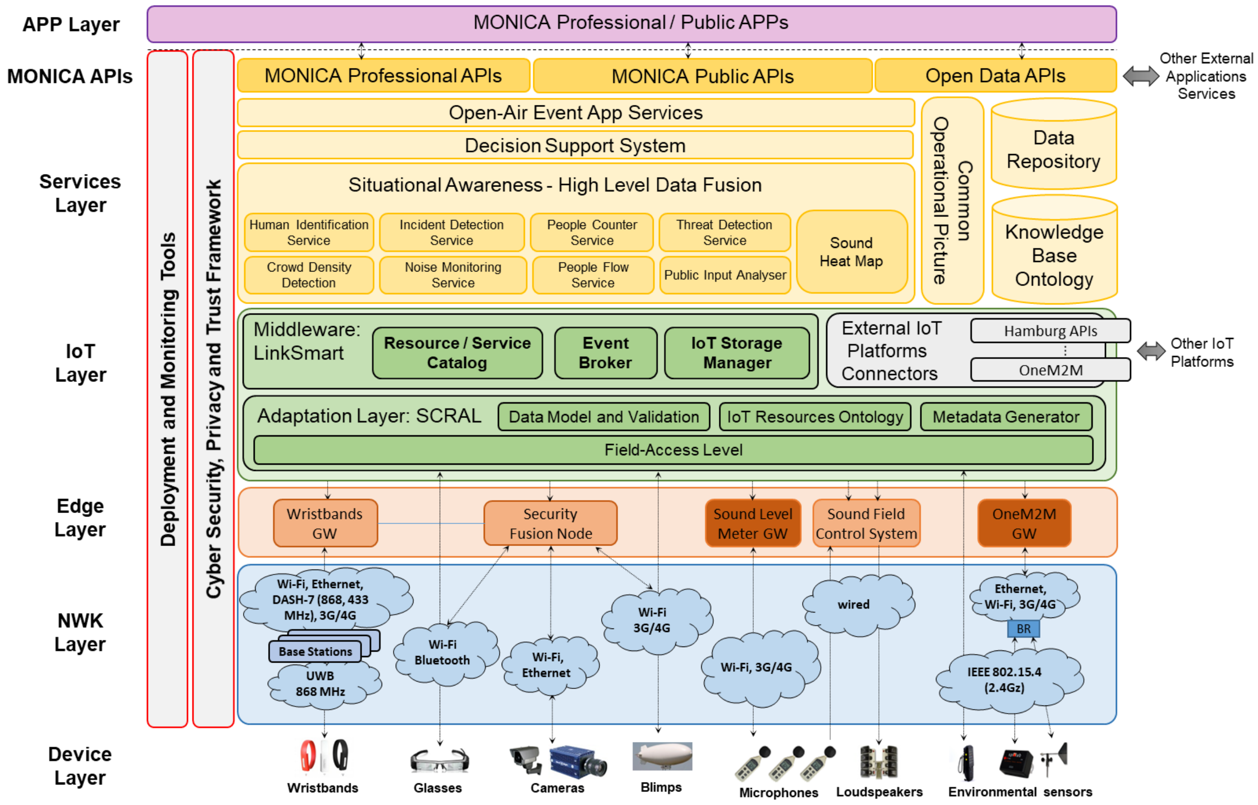


Figure 18: Functional View of the MONICA Architecture.

As it can be observed, the architecture comprises the following subsystems, also called layers:

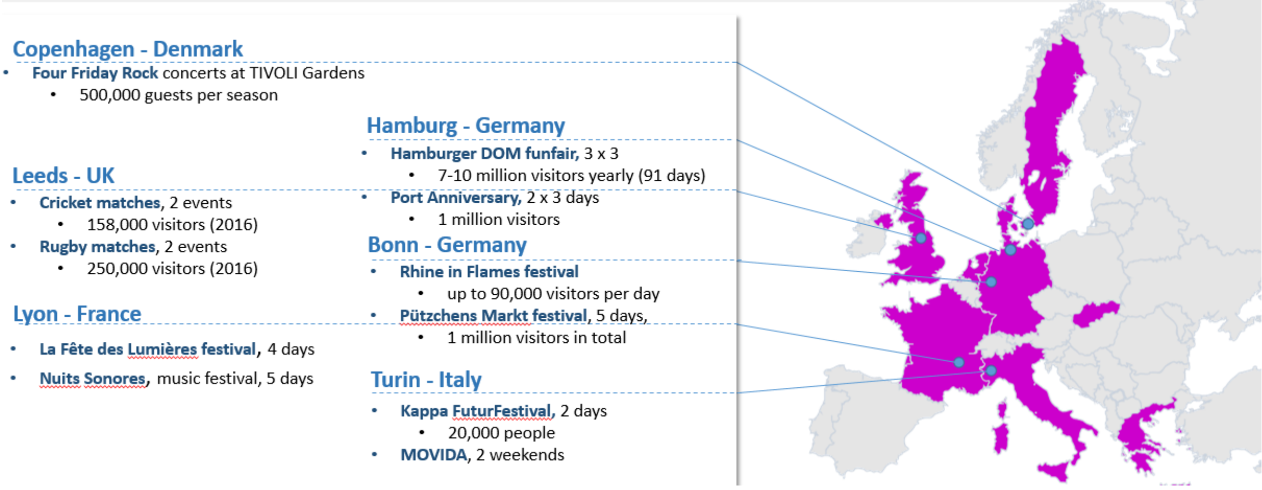
- The ***Device Layer*** includes all IoT wearables (*e.g.*, wristbands and glasses) and IoT sensors, which can be fixed (*e.g.*, sound level meters, loudspeakers, cameras, environmental sensors) or mobile (*e.g.*, wireless sound level meters, cameras installed in a Blimp).
- The ***Network Layer*** that allows the effective communication between the heterogeneous IoT wearables, IoT devices and the IoT platform modules. This layer is responsible of forwarding data coming from the IoT wearables and IoT sensors as well as of responding to service requests coming from upper layers;
- The ***Edge Layer*** includes a set of processing modules (*e.g.*, the Wearables GW running localization algorithms, Processing Units executing video-based algorithms, the Sound Field Control System (SFCS) for managing the sound quality and noise reduction) that process real-time data directly from the *Device Layer*. To this purpose, these modules need to be deployed locally in the pilot site to avoid the latency introduced by the upper layers of the platform. Moreover, these modules require an efficient and scalable Network Infrastructure.
- The ***IoT Layer*** is composed of the following three subcomponents:
  - The *Adaptation Layer*, here represented by the SCRAL, providing technology independent management of physical resources and uniform mapping of data into standard representations that can be easily handled by the upper platform modules;
  - The *Middleware*, here represented by the LinkSmart, which offers storage and directory services for resources registered in the IoT platform;

- The *External IoT Platform Connectors*, handling the communication with external IoT platforms and the integration of data coming from outside (*e.g.* from the Hamburg Smart City platform). In addition, MONICA integrates the oneM2M interfaces (Mca and Mcc/Mcc') allowing the platform to expose the IoT data according to the OneM2M standard;

- The ***Services Layer***, where the intelligence of the platform is implemented, and specific processing modules are integrated to provide technical solutions compliant with the application requirements. The services modules are combined together with knowledge base components and decision support tools whose aim is to propose a set of intervention strategies to assist human operators in gathering context-sensitive information and decision making;
- The ***MONICA APIs Layer***, which provides service access points for MONICA application developers and external application developers that want to access MONICA functionalities and information streaming from the platform;
- The **Cyber *Security and Privacy Framework***, enabling trust-based communication, policy management and technical support across all levels of the platform. More specifically, this framework ensures secure data flows and storage, protected information exchange and trusted federation mechanism to facilitate private information sharing;
- The ***Deployment and Monitoring Tools.*** These tools belong to a transversal framework able to easy the platform deployment (*e.g.* modules belonging to the *Device* and *Network* layers) and used for checking the operational status of the devices, networks and overall system. Moreover, these tools are also used for measuring performance metrics and tracing pilot events.

4.3.4.1.2 Use Cases

The following picture shows MONICA pilots Figure 19 where use cases and technologies will be implemented during the project.



Copenhagen - Denmark

- **Four Friday Rock** concerts at TIVOLI Gardens
  - 500,000 guests per season

Hamburg - Germany

- **Hamburger DOM funfair,** 3 x 3
  - 7-10 million visitors yearly (91 days)
- **Port Anniversary,** 2 x 3 days
  - 1 million visitors

Leeds - UK

- **Cricket matches**, 2 events
  - 158,000 visitors (2016)
- **Rugby matches**, 2 events
  - 250,000 visitors (2016)

Bonn - Germany

- **Rhine in Flames festival**
  - up to 90,000 visitors per day
- **Pützchens Markt festival**, 5 days,
  - 1 million visitors in total

Lyon - France

- **La Fête des Lumières festival**, 4 days
- **Nuits Sonores**, music festival, 5 days

Turin - Italy

- **Kappa FuturFestival**, 2 days
  - 20,000 people
- **MOVIDA**, 2 weekends

Figure 19: MONICA Pilots overview.

At the beginning, fifty generic high-level use cases have been defined and grouped in 12 categories, and the subsequent prioritisation by the pilots resulted in the selection of four use case groups for implementation.

The main use case group selected are the following:

- **UGG1: Sound Monitoring and Control**: Copenhagen (DK), Lyon (FR), Bonn (DE), Torino (IT).
- **UCG2: Crowd and Capacity Monitoring**: Copenhagen (DK), Lyon (FR), Bonn (DE), Leeds (UK), Torino (IT), Hamburg (DE)

- **UCG3: Missing Persons/Locate staff members**: Copenhagen (DK), Hamburg (DE), Bonn (DE), Leeds (UK), Torino (IT), Lyon (FR).
- **UCG4: Health/Security Incidents**: Copenhagen (DK), Bonn (DE), Hamburg (DE), Torino (IT), Leeds (UK), Lyon (FR).

The MONICA architecture applies to all Use Cases prioritised and selected. Based on pilot requests, in fact, some architectural modules are used and others not.

As an example, in Leeds the *ASFC Acoustic module* is not used to limit noise in the neighborhood. Some pilots use only UWB wearable, other pilots use only crowd wristbands (based on 868), and some pilots use both. In all these cases, the MONICA architecture structure remains the same.

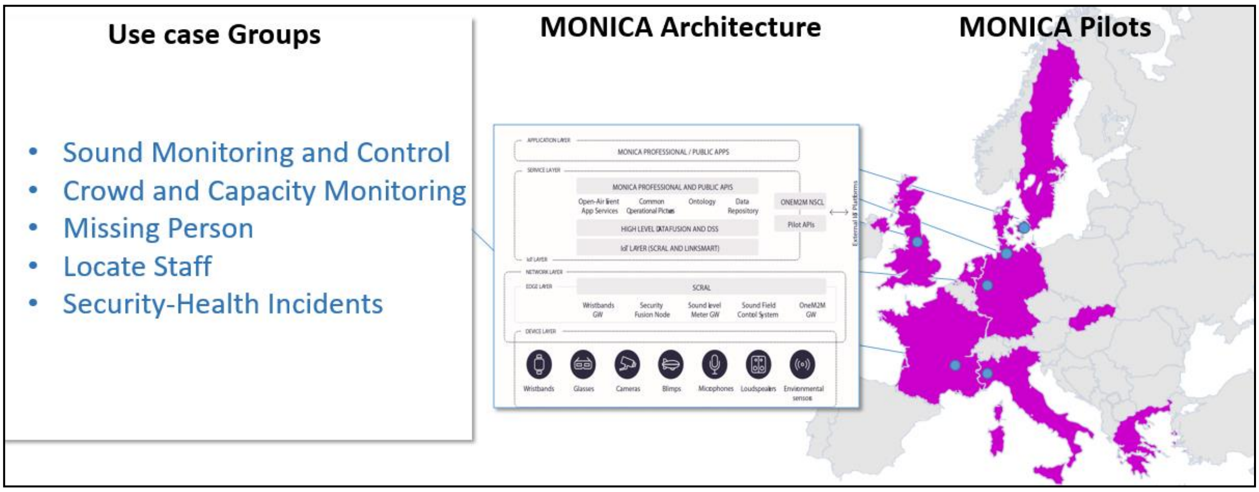


Figure 20: MONICA Use Cases and Architecture.

4.3.4.2 Comments

The MONICA reference architecture has been defined as a result of two main phases, namely, *bottom-up* and *top-down*. In the *bottom-up* phase, technology partners have described in detail their main components and their expertise requested by MONICA. In this phase, MONICA reused as much as possible partners experience as well as assets from previous EU projects related to Internet of Things (IoT) (*e.g.*, ALMANAC, IMPRESS, ebbits). After that, the MONICA architecture has been defined following the AIOTI - High Level Architecture (HLA). After that, in the *top-down* phase, architecture components have been preliminary tested by means of UML sequence diagrams by taking as input uses cases (UCs) reported in D2.1. The purpose of these sequence diagrams was to clarify how the MONICA platform would work and which components are relevant to achieve different tasks.

The MONICA architecture has been documented following the standard ISO/IEC/IEEE 42010 - "*Systems and software engineering - Architecture description*" (ISO/IEC/IEEE 42010, 2011), which implies a process based on a set of relevant architecture viewpoints. For MONICA, three functional viewpoints have been defined, namely *functional view*, *deployment view* and *information view*.

The *functional view* describes the components, their functionality, and their interactions. The main identified architecture components are:

- *Wearable devices* that include two types of wristbands (low cost ones based on 868MHz and the Ultra-Wide Band (UWB) ones that provide a more accurate localization service) and glasses.
- The *IoT platform*, composed of the adaptation layer named *Smart City Resource Adaptation Layer* (SCRAL), developed in the ALMANAC EU project, and the LinkSmart middleware, adopted in several EU projects (*e.g.*, ALMANAC, IMPRESS, ebbits).

- The *Situational Awareness* that retrieves the information necessary to identify a big picture of the current situation and detect anomalies conditions that will be immediately notified to the Decision Support System (DSS) module.
- The *Decision Support System* module proposes to the users a set of intervention strategies based on the incidents detected by the *Situational Awareness* framework.

The *deployment view* describes how and where the system will be deployed, which physical components are needed, what are the dependencies, hardware requirements and physical constraints. The *information view* describes the application domain models and the data flow as well as data distribution.

4.3.5 SYNCHRONICITY

4.3.5.1 Use Cases and Reference Architecture

SYNCHRONICITY has developed (and documented) 6 Use Cases. The Reference Architectures in support of these Use Cases are discussed below.

The Reference Architecture selected by the Use Cases is shown in Figure 21.

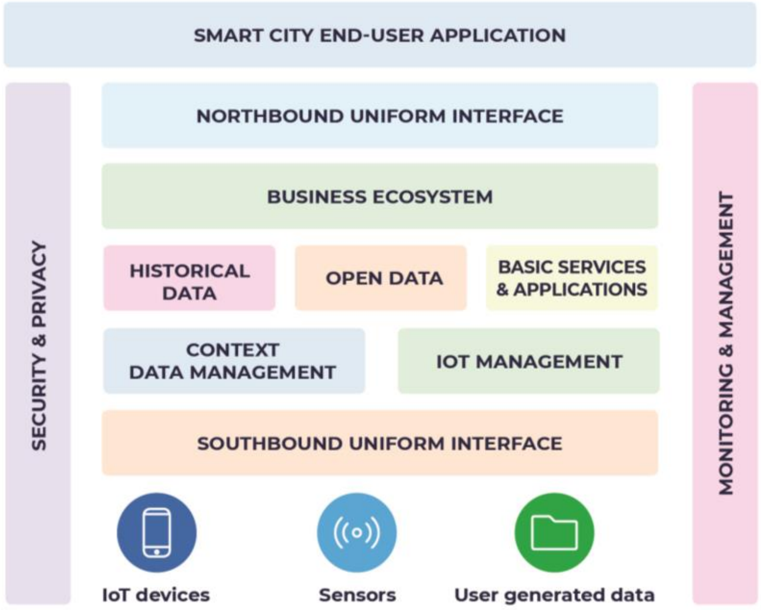


Figure 21: SYNCHRONICITY simplified reference architecture

On top of the above reference architecture provided by most of the Use Cases, a few Use Cases have chosen to present a more "system"-oriented Reference Architecture as in the two examples shown in

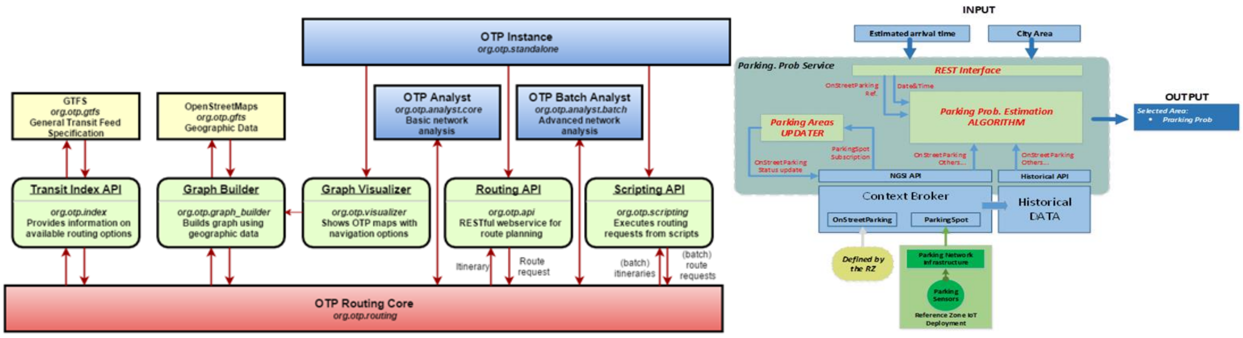


Figure 22: Two examples of System Architecture in SYNCHRONICITY

4.3.5.2 Comments

The approach taken for the design of the Reference Architecture has been to analyse the various deployments on pilot sites in partner cities and to map the findings with an overview of standardised technologies and available standards. One of the objectives of this analysis has been to identify commonalities within similar other projects.

The analysis was based on the use of a survey aiming at identifying key aspects of core technologies and of the main functionalities that are at the core of smart city platforms

These commonalities were the base of the SYNCHRONICITY framework designed for its applicability by a vast and diverse range of cities.

4.4 Commonalities, differences and (wider) applicability

Some commonalities have been identified across the Reference Architectures of the LSPs. This section will introduce a first approach for a consolidated Reference Architecture on which all the architectures of the five IoT LSP can be mapped.

Beyond this initial consolidation, it may also be important to evaluate the applicability of the proposed Reference Architecture beyond the IoT LSPs sectors.

The example of the Industrial IoT (IIoT) is chosen for a first analysis, based on a comparison with two Reference Architectures developed in that sector: RAMI 4.0 and IIRA.

4.4.1 A common view of the IoT LSPs Reference Architectures

The work of the LSPs after they have reached the half of their life-time has produced a number of specific Reference Architectures that share a lot of commonalities.

The Figure 23 below has the objective to capture some of these commonalities. It is an initial attempt to provide a Reference Architecture that can be applied by all LSPs and possibly beyond.

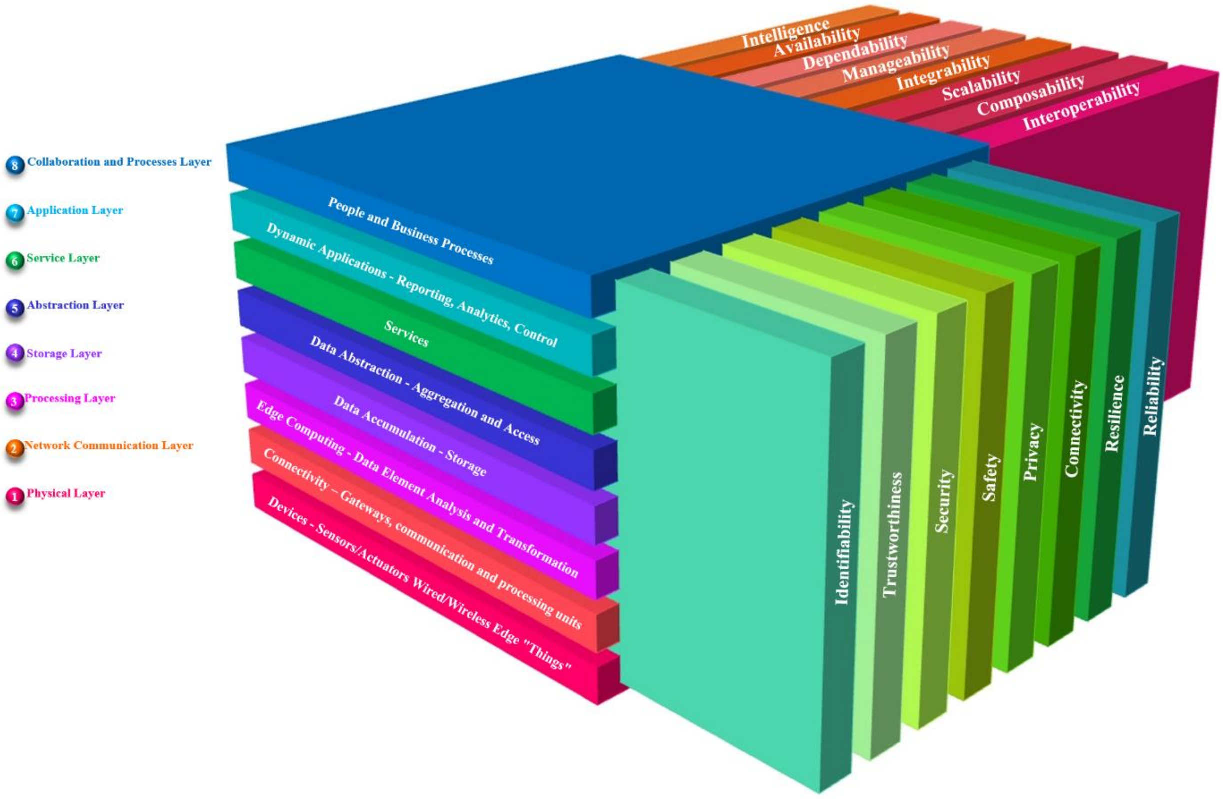

Intelligence
Availability
Dependability
Manageability
Integrability
Scalability
Composability
Interoperability
8 Collaboration and Processes Layer
7 Application Layer
6 Service Layer
5 Abstraction Layer
4 Storage Layer
3 Processing Layer
2 Network Communication Layer
1 Physical Layer
People and Business Processes
Dynamic Applications - Reporting, Analytics, Control
Services
Data Abstraction - Aggregation and Access
Data Accumulation - Storage
Edge Computing - Data Element Analysis and Transformation
Connectivity – Gateways, communication and processing units
Devices - Sensors/Actuators Wired/Wireless Edge "Things"
Identifiability
Trustworthiness
Security
Safety
Privacy
Connectivity
Resilience
Reliability

Figure 23: A blueprint for IoT Focus Area Reference Architecture

The IoT Reference Architecture proposed in Figure 23 is building on the common approach to "horizontal" functional layering that reflects the IoT implementations across various domains. The IoT architecture framework is expanded with two additional dimensions that are partly addressed by LSPs, while including other views of the IoT systems.

The three dimensions of the proposed model are referring to different concerns in the elaboration of the functions of an IoT system. These three dimensions are described below:

- Layers. The layers described here are those that have been retained in the LSP Presentation document (see [4]) and described in greater detail in Table 2.
- Cross-cutting Functions. This dimension addresses additional functionalities that are not linked to a single layer but whose provision requires spanning across several layers. Examples of such cross-cutting functions are security and privacy that are described in all LSPs Reference Architectures as cross-layer functions.
- Properties. This dimension addresses the global properties of the IoT system that re (or not) provided b y a proper implementation of functions (at all layers) and cross-cutting functions. As an example, trustworthiness is resulting in particular from the proper implementation of the security and privacy cross-cutting functions.

This initial blueprint will be further elaborated, with a first objective to ensure the mapping of the three-dimensional model Reference Architecture of Figure 23 with the LSPs Reference Architectures.

Beyond this, considering the continuous progress in the LSPs and the parallel developments of Reference Architectures in other sectors that are not in the scope of the LSPs, such as Industrial IoT which is analysed below, further refinements are under consideration (in particular in the IoT LSP Activity Group 02) and will be exposed in a further deliverable from CREATE-IoT work package WP06.

4.4.2 Looking beyond the IoT LSPs domains: the example of Industrial IoT

Two Reference architectures are developed, discussed and elaborated in the Industrial IoT (IIoT) domain, namely RAMI 4.0 and IIRA which are analysed below. It is important to note that these two architectures have complementary purposes, thus explaining some differences.

4.4.2.1 Reference Architecture Model Industrie 4.0

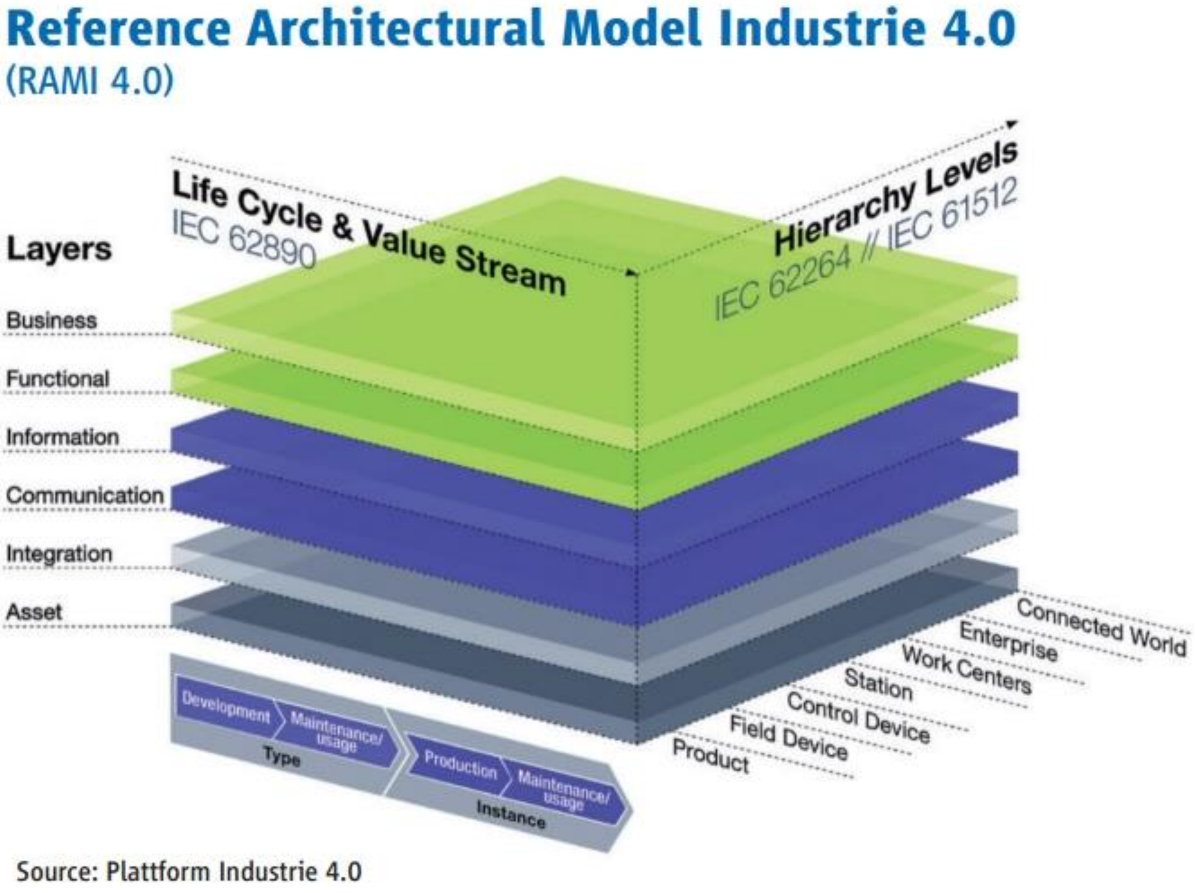
The Reference Architecture Model Industrie 4.0 (RAMI 4.0) is a three-dimensional map in support of the most important aspects of Industrie 4.0 which aims at connecting all stakeholders involved in the business processes of the manufacturing and process industry so that all participants involved share a common perspective and develop a common understanding.

RAMI is a three-dimensional model reflecting:

- The Life-Cycle and Value Stream, from development to decommissioning, with both the perspectives of the products and services to be offered and of the supporting processes and workflows;
- The Layers, in support of business and interoperability:
  - Business; Functions; Information; Communication; Integration, Asset (the "Things")
- The (factory) Hierarchy Levels: from Product to Connected World

RAMI is mostly concentrating on Manufacturing and the integration of the flexible factory into the connected world. One important objective of RAMI 4.0 is to support the convergence of Information Technology (IT) and Operational Technology (OT, i.e. IT systems in the shop floor).

Figure 24 is presenting the three dimensions of RAMI 4.0 together with some of the supporting standards.



Source: Plattform Industrie 4.0

Figure 24: The Reference Architectural Model Industrie 4.0 (RAMI 4.0)

4.4.2.2 Industrial Internet Reference Architecture (IIRA)

The IIRA is a standards-based open architecture for IIoT systems. The architecture description and representation are generic and at a high level of abstraction to enable a broad industry applicability.

IIRA uses the ‘ISO/IEC/IEEE 42010:2011 standard (Systems and Software Engineering–Architecture Description) as a Conceptual Model to define the architecting conventions and common practices and provides an ontology for the description of architectures.

The IIRA is defining four viewpoints:

- Business: identifies stakeholders and their business vision, values and objectives;
- Usage: addressing the expected system usage, it is typically represented as sequences of activities involving human or logical (e.g. system or system components) users;
- Functional: focuses on the functional components, their structure and interrelation, the interfaces and interactions;
- Implementation: deals with the technologies needed to implement functional components

On top of the functional model dimension, the IIRA considers two dimensions regarding:

- System Characteristics: refers to behaviours and properties resulting from the interactions of the parts of the system. Safety and security are two examples of such characteristics;
- Cross-cutting Functions: refers to the additional functions that need to be provided in order to enable the major system functions. Connectivity is an example of such cross-cutting functions.

It can be noted that what is called "Cross-cutting Functions" earlier in this document is referred to as "System Characteristics" in IIRA. There is however a great similarity between both.

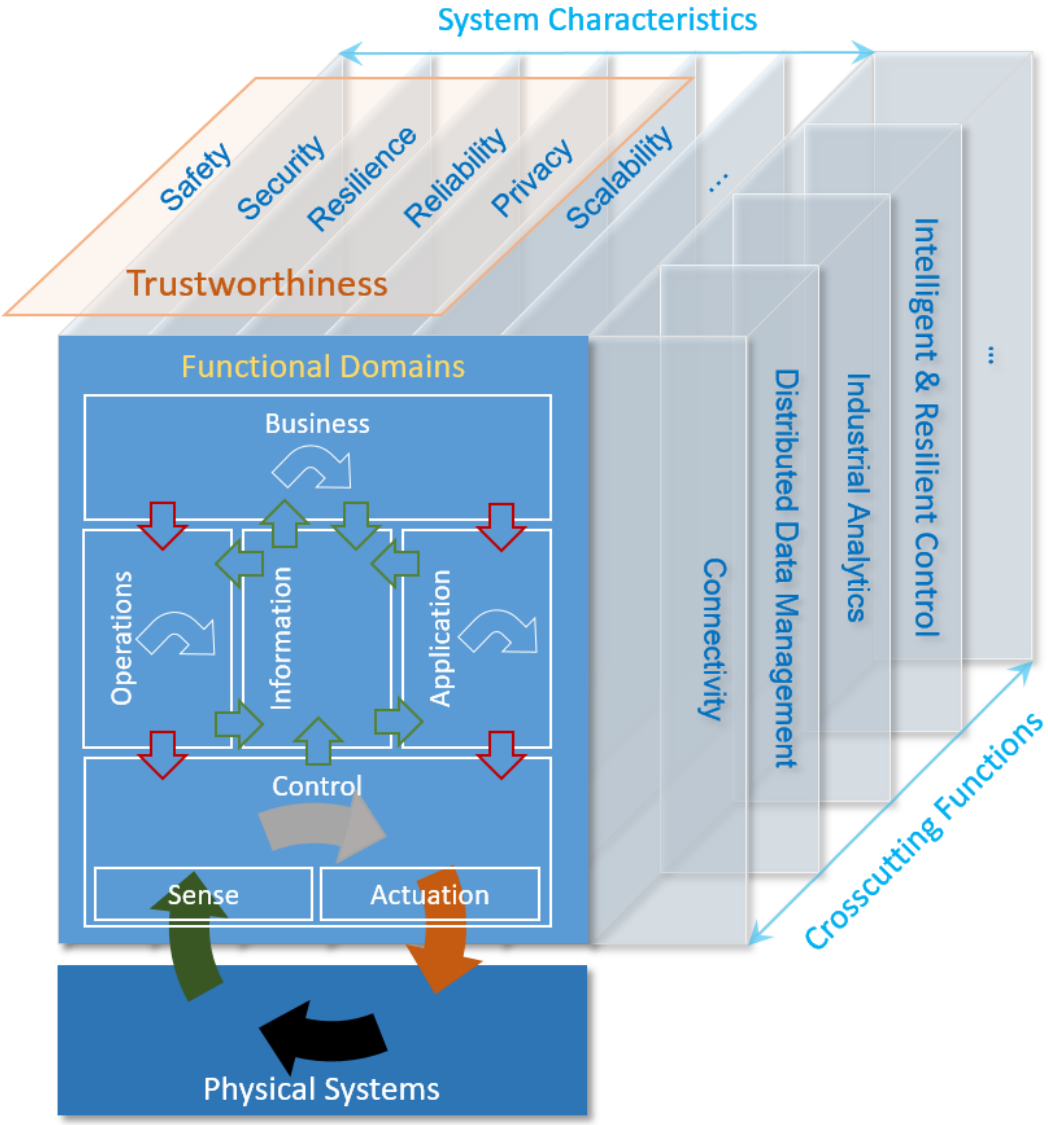


Figure 25: IIRA Functional Domains, Crosscutting Functions and System Characteristics (Source: IIC)

4.4.2.3 Comparing RAMI 4.0 and IIRA

RAMI 4.0 and IIRA are two Reference Architectures in the Industrial IoT domain. Both architectures have the same goal of supporting the convergence of OT and IT.

However, while Industrie 4.0 (and RAMI 4.0) focuses on Manufacturing (i.e. making things), IIC (and IIRA) addresses cross-industry commonality and interoperability (i.e. making things work).

Industrie 4.0 and IIC have issued a White Paper [16] dealing with the alignment of their respective architectures. It identifies some complementarities, similarities and differences such as:

- The Life-Cycle model of RAMI and the Solution CXycle of IIRA are complementary;
- The layer dimension of RAMI and the functional view of IIRA bear some similarities;
- The usage and implementation viewpoints of IIRA have no match in RAMI;
- The hierarchical dimension of RAMI has no match in IIRA;

The Industrie 4.0/IIC White Paper is proposing a Functional Mapping between RAMI 4.0 and IIRA that is shown in Figure 26.

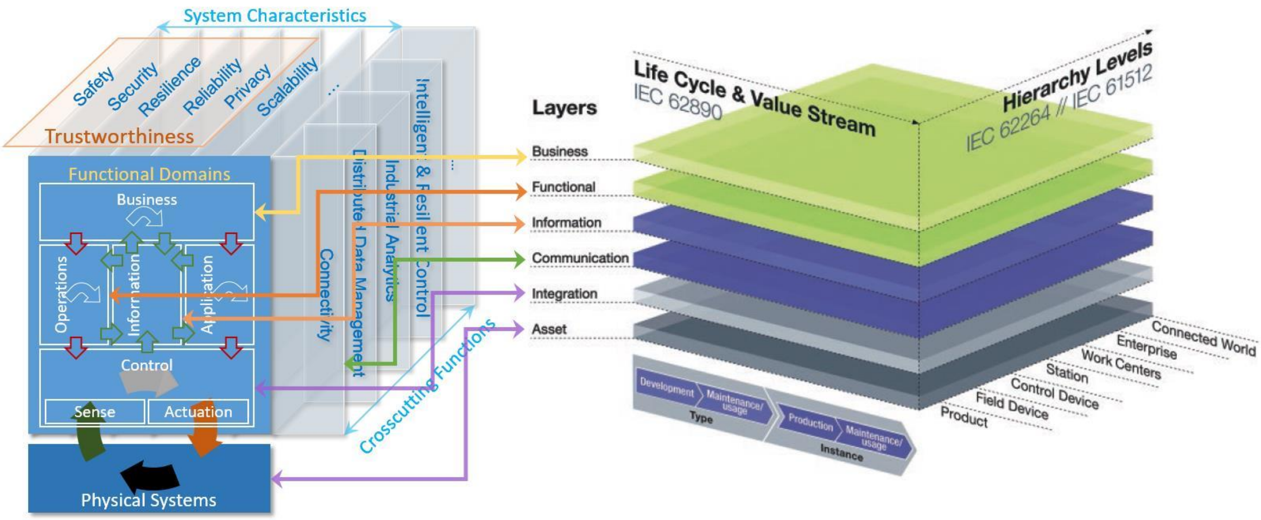


Figure 26: IIRA and RAMI 4.0 Functional Mapping

4.4.2.4 RAMI 4.0, IIRA and the LSPs Reference Architecture

The IIRA and RAMI 4.0 Functional Mapping (shown in Figure 26 and much more detailed in [16]) shows that there is a strong possibility of developing successfully a mapping between the LSPs Reference Architecture outlines above and the two architectures that prevail in IIoT and Manufacturing.

Such a mapping will rely on a more precise definition of the LSPs Reference Architecture that is outlined in the present document and will be further enhanced in the upcoming IoT LSP Activity Group 02 (Standardisation, Architecture and Interoperability) workshops to be held in 2018 and early 2019.

5. Interoperability Support in LSPs

5.1 Introduction

For several IoT projects, in particular those who span large domains (e.g., Smart Cities), cross-domain interoperability is a key requirement for achieving large scale deployment of IoT-enabled services.

On top of a reference architecture model, other elements are required such as cross-application interoperability points (describing where interoperability is supported) and some supporting mechanisms (describing how the support is provided).

As an alternative to ad-hoc approaches, project by project, some specifications and standards are emerging to this purpose.

5.2 Elements of Interoperability Support

A major challenge addressed in standardisation (and in research and pre-standardisation) is to understand how interoperability can be supported by higher-level constructs, then offer in particular a more dynamic support to Application Programming Interfaces.

Several approaches, under consideration and development in the LSP community and beyond, are listed below.

Even if their effectiveness is (for some of them) still under evaluation, it is expected that they will reach a level of maturity at the end of the LSPs development.

5.2.1 Interoperability Patterns

Syntactic and semantic interoperability provide effective – and in particular more dynamic - support mechanisms that may help the IoT application developers and ensure that they can effectively use supporting standards proposed by the industry.

In [17], six generic interoperability patterns, numbered from I) to VI) have been identified that apply to systems in general, one related to platforms and the definition of efficient IoT platforms that can support one or more of these patterns and one related to standards and which kind of support they provide.

The six patterns (described in below) are:

I. Cross-Platform Access. The basic pattern where an application can interoperate with several platforms.
II. Cross-Application Domain Access. This pattern expands the previous with the ability to interoperate with platforms in different domains.
III. Platform Independence. The same application or service can be used on top of two different IoT platforms (e.g. in different regions) without changes.
IV. Platform-Scale. With this pattern, the focus is on integrating platforms of different scale.
V. High-Level Service Facades. This pattern extends the interoperability requirements from platforms to higher-level services where not only platforms but also services offer information and functions via the common API.
VI. Platform-to-Platform. This pattern extends the interoperability requirements across platforms.

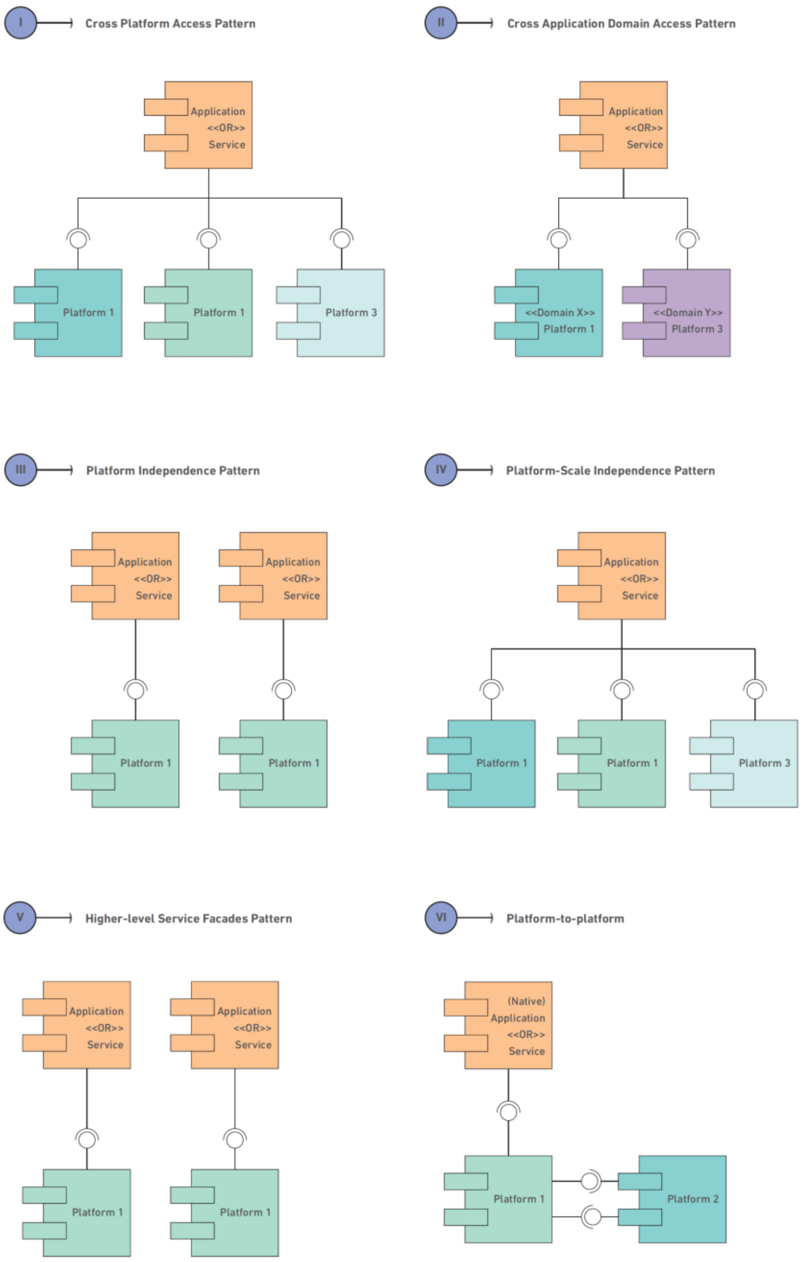

I
Cross Platform Access Pattern
Application <<OR>> Service
Platform 1
Platform 1
Platform 3
II
Cross Application Domain Access Pattern
Application <<OR>> Service
<<Domain X>> Platform 1
<<Domain Y>> Platform 3
III
Platform Independence Pattern
Application <<OR>> Service
Application <<OR>> Service
Platform 1
Platform 1
IV
Platform-Scale Independence Pattern
Application <<OR>> Service
Platform 1
Platform 1
Platform 3
V
Higher-level Service Facades Pattern
Application <<OR>> Service
Application <<OR>> Service
Platform 1
Platform 1
VI
Platform-to-platform
(Native) Application <<OR>> Service
Platform 1
Platform 2

Figure 27: Six Patterns of Interoperability [17]

5.2.2 Interoperability Points and Mechanisms

There is a potentially very large number of ways to architecture an IoT system and no specific guaranty that its different sub-systems will be able to interwork properly. In order to reduce the difficulty of providing interoperability on this context, a fruitful approach is to reduce the number of contact points between the elements of the IoT system and provide standardised way to manage the necessary interactions.

The two essential elements in support of this approach are the following:

- Interoperability Points which are the main interfaces that allow applications to interact with the supporting platform. These interfaces must be independent from the specific software components that realize them and offer various potential implementation variants. Examples of such interfaces in the context of the LSPs are northbound interfaces, southbound interfaces or shared data models.

- Interoperability Mechanisms represent the actual interface specifications at the Interoperability Point. Examples of such mechanisms are standard API (e.g., for security, data storage) and guidelines.

The LSPs have developed such points and mechanisms that are addressed below in this section.

5.2.3 Market Places and APIs

The new paradigm for IoT systems that are subject to large-scale deployments is to use layered, potentially cloud-based or edge-enabled architectures. Such architectures come with strong requirements on the connectivity between actors (e.g. sensors, gateways, platforms, data processing and analytics functions, etc.) and their support requires complex interoperability schemes.

IoT systems and application developers are expecting that the very large number of devices to be deployed and connected to the network are able to interoperate seamlessly with the largest range possible of platform services (e.g. data analytics, monitoring, visualization, etc.) and very diverse end-user/end-customers applications.

A new approach is to consider the stakeholders of the IoT systems can be seen as consumers and providers within an emerging "application marketplace" which can be seen as a new platform to extends the "traditional" IoT platforms with forms of brokerage that support automated discovery, trading and even pricing.

Within such an IoT marketplace, the IoT device owners have the possibility to selectively grant access and trade their data with many potential vendors. The support of a multi-vendor and multi-owner environment is creating an environment where monetization and efficiently development of innovative solutions can be effectively fostered.

Marketplace architectures are in general supported by:

- The publication of several Application Programming Interfaces (APIs) that hide the actual underlying provision of the service from the consumer of the service. The implementation of the service can change without impacting the rest of the system and the evolution of the APIs can be mastered via the publication mechanism.
- An approach based on Microservices where any service (whichever its size and scope) can be published and consumed. This approach provides system flexibility; supports lean software principles; and allows fast adaptation to support emerging standards without impacting the whole system architecture. It is in general supported by many Open Source communities.

5.3 Interoperability Support in the LSPs Implementations

5.3.1 Introduction

The purpose of this section is to introduce the approach that the LSPs, after a few months of their existence, have considered with respect to standards and how they can help them address the requirements outlined in the section above. Maybe it is yet a proper "strategy", but the role of this document is to be a support to filling the gap between an "approach" and a "strategy". The respective sections can be very different from one LSP to another one (e.g. different levels of maturity).

5.3.2 Interoperability Support in ACTIVAGE

Internet of Things (IoT) research and industry communities have realized that a common IoT problem to be tackled is the interoperability of the information. In ACTIVAGE we reviewed recent trends and challenges on interoperability, and discuss how often and how flexible semantic technologies, open the services frameworks and tenable their information models to support. Extensible discussed the Internet of Things (IoT) refers to things (objects) and the virtual representations of these objects on the Internet. IoT Interoperability then shall define how the things “talk” amongst other things and communicate with other systems in order to expose their capabilities and functionalities “services”. Interoperability through the platforms presented in ACTIVAGE project is carried out by means of the IoT Interoperability Layer. Semantic Interoperability in the broader sense has been defined in section four, at this point we just focus on describing the components of the interoperability layer which, by definition, is an abstraction layer that allows the communication between an application of the marketplace and the ACTIVAGE platform.

The Interoperability Layer where communications are managed by a message broker that participates in every communication in IoT Interoperability Layer is shown in Figure 28. As it is shown a general API is used to access the broker that exposes basic common operations (message pub/sub, topic creation, basic resources management...), enabling the ability to interchange the actual implementation of the broker. On the one hand, one of the benefits is the isolation of the communication responsibility in a single element, which in turn makes profiling, scaling and adaptation to enterprise infrastructures easier. On the other hand, it allows complete decoupling between components. The broker can be divided, from a functional point of view, into four blocks, namely, API Requester Manager, Platform Request Manager, Data Flow Manager and Message Queue.

The functional block in charge of handling request received from the API proxy is called API Requester Manager. This block allows bookkeeping of active sessions and their respective callbacks; forwarding requests to the Platform Request Manager for further processing; and providing feedback to the caller.

The Platform Request Manager prepares and sends requests to specific platforms through bridges, using already established permanent data streams, which it creates during start-up with the help of Data Flow Manager, or it creates new data streams. All data streams that go south, from the Platform Request Manager to bridges, go through permanent data streams, which can be either routed through IPSM, or bypassing it and connecting directly to the bridges. All data streams that go north, from the bridges to the Platform Request Manager, need to be created as needed.

During request pre-processing the Platform Request Manager is potentially assisted by some middleware services, such as routing or the device registry. It sends requests to underlying platforms as/when needed. The Data Flow Manager acts as orchestrator of data flows from the platforms (bridges) to the original caller, utilizing already established permanent data streams or creating new ones and ensuring that all intermediaries are included in the path. Finally, The Message Queue, only receives and provides the messages to the corresponding components, including ad-hoc temporary topics for single requests, and fixed platform channels.

The IoT Platform Semantic Mediator (IPSM) block manages the ontologies and provides semantic interoperability through the translation among the different platform ontologies and the ACTIVAGE ontology. This translation is performed by means of ontology alignment.

The information exchange is facilitated thanks to the use of platform bridges. These manage the communication with the subjacent platforms by translating its requests and answers. Different bridges might need to use HTTP, REST, sockets or other technologies to talk to the platforms, but these will be translated northwards into messages. The decision to connect the platforms directly to the abstraction layer instead of interconnecting all platforms among themselves, simplifies considerably the interoperability.

In the services group of components, the most important are the Platform Registry and Capabilities, that contains the information of all connected Platforms including their type and service capabilities, the Resource Discovery that creates requests to obtain the necessary information from the platforms, and the Resource Registry, that contains a list of resources (e.g. devices) and their properties that can be quickly consulted.

In the second phase, the Routing and Roaming Service will be expected to allow the communication with a particular device independently of the platform it is currently connected to, while Authentication and Accountability (not shown) would provide services for the security and monitoring of all the actions.

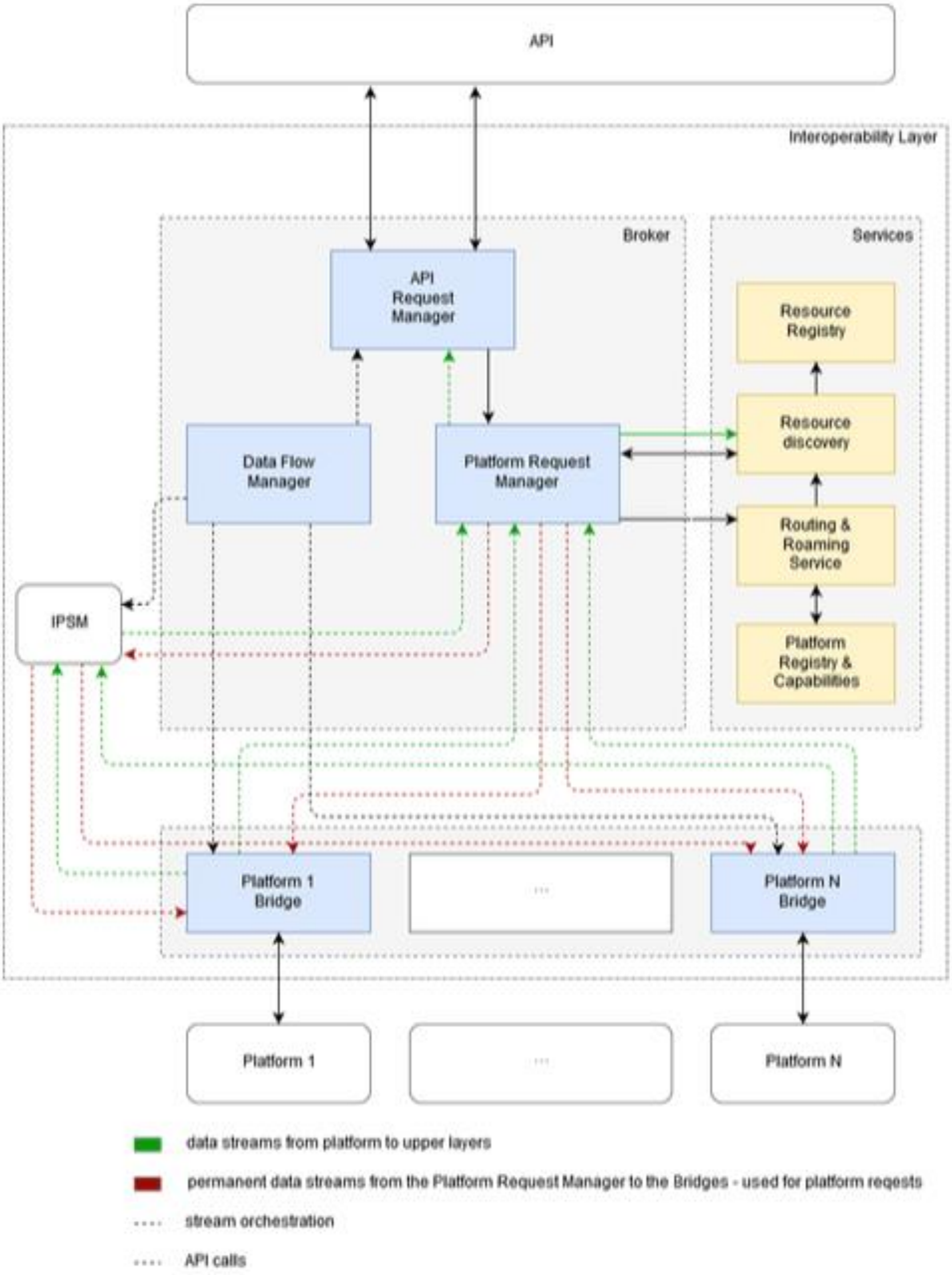


Figure 28: ACTIVAGE Interoperability Reference Architecture

5.3.3 Interoperability Support in AUTOPILOT

AUTOPILOT focuses on using standard based IoT architectures following AIOTI-WG03 and OneM2M standards to ensure deploying interoperable and replicable IoT platforms and architectures. The Task 2.5, “Pilot Readiness Verification”, verifies the interoperability and functionality during the development phase, while the Task 4.2 “Technical evaluation” evaluates the suitability of IoT technologies for automated driving. Security and privacy by design has a key impact of AUTOPILOT and is covered in the Task T1.5 "Security, Privacy and Data Specifications". AUTOPILOT is fostering the development of conformance testing in order to promote the development of a compliancy assessment framework for Automated Driving IoT platforms. These impacts are the main target of the Task 5.5 "Standardisation and conformance assessment". WP5 is taking care of the standardisation and compliancy assessment issues, and active involved during the specification and the development process to ensure interoperability, replicability and sustainability of the AUTOPILOT results. Interoperability between the IoT platforms is addressed in Task 2.3 and interoperability assessment between the IoT technologies and IoT architectures for the required provision of services for connected and automated driving are discussed in Task 4.5 – User Acceptance.

Interoperability in AUTOPILOT is achieved based on the following three principles:

- **oneM2M Interoperability Platform and Interworking Gateways:** where proprietary IoT platforms are interconnected through interworking gateways and the oneM2M interoperability platform.
- **Standardised IoT Data Models:** IoT data requiring to be exchanged across the IoT platforms are standardised.
- **Standardised Ontologies:** To achieve semantic interoperability, IoT data fields values (e.g. hazard types, vulnerable road user types, detected object types, etc.) are semantically standardised in ontologies.

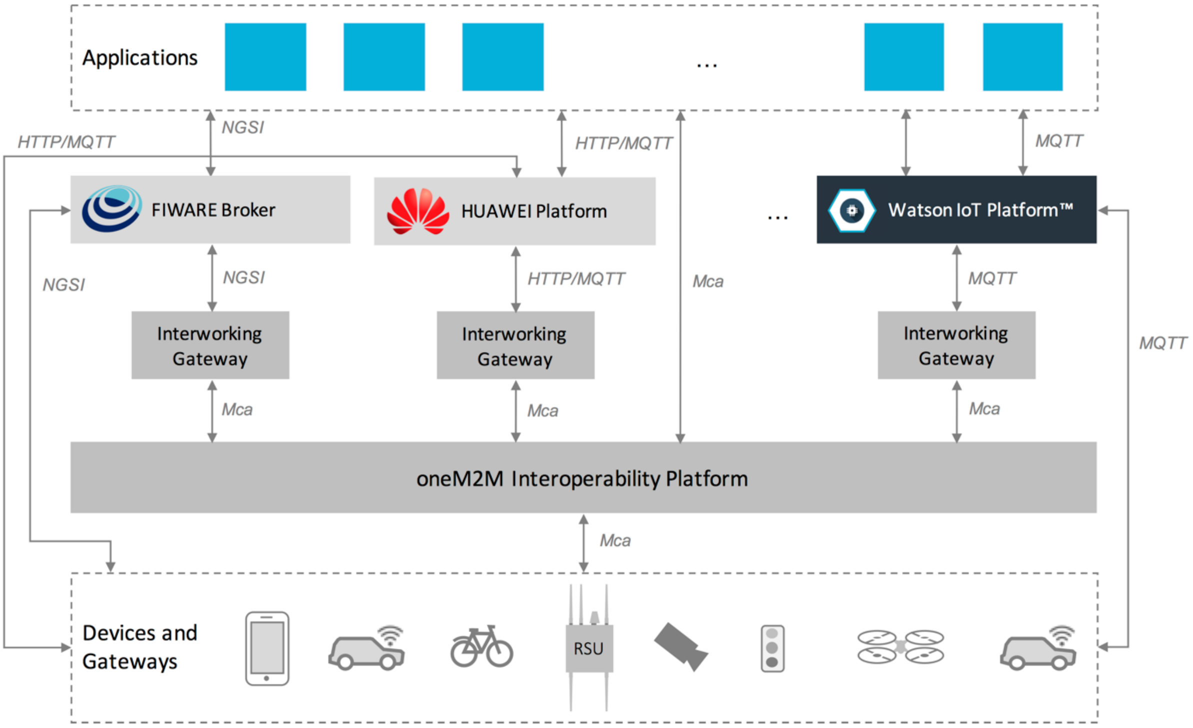


Figure 29: AUTOPILOT Federated IoT Architecture

AUTOPILOT IoT architecture was designed as a federation of IoT platforms, allowing it to be open and flexible. Developers plug their own (proprietary) IoT platforms or devices in the architecture and exchange data with existing IoT platforms and devices. As each IoT platform provides a different set of services (features) and may expose a different interface and use a different data exchange protocol, the interoperability approach used allows for openness and flexibility. An open oneM2M standard IoT platform, referred to as the oneM2M interoperability platform, interconnects these proprietary IoT platforms provided by the project partners. The proprietary IoT platforms collect data from connected devices. They exchange IoT data and events with the interoperability platform through oneM2M interworking proxies.

The proprietary IoT platforms are networked through the oneM2M interoperability platform and are connected to this through oneM2M interworking gateways. The interworking gateway of a given proprietary IoT platform may be configured to share selected data types with the interoperability platform. Such data will then become accessible to all the connected IoT platforms through the oneM2M interoperability platform. This is useful for sharing data relevant to all the AD vehicles and applications, such as detected hazards, vulnerable road users, objects, etc.

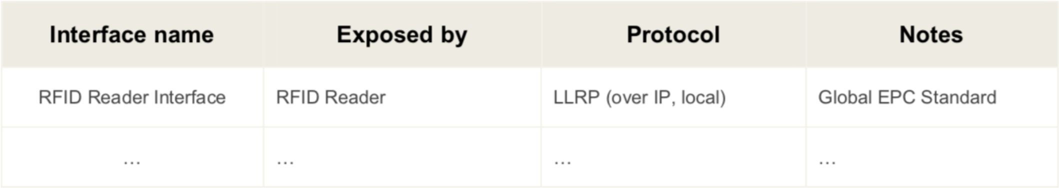
The Watson IoT Platform™ instances are connected to the oneM2M interoperability platform through oneM2M connectors, called interworking proxies. The SENSINOV oneM2M platform is used as the interoperability bridge for linking the AUTOPILOT IoT platforms. Information exchanges between different IoT platforms is done using interworking gateways to facilitate interoperability between the platforms.

5.3.4 Interoperability Support in IoF2020

Building on the experience being generated on the field, the plan in IoF2020 was to establish a common view, for each of the Use Cases. This was an important preparatory step towards further activities in the project, which aim at the full realization of the IoT vision across the 19 IoF2020 Use Cases, ensuring that deployed components and solutions can prospectively inter-operate so to deliver added-value functionalities to various stakeholders – possibly maximizing re-use of common IoT enablers across different Use Cases and trials.

This can only be achieved by leveraging common interoperability endpoints and data models and allowing secure and controlled exchange of information and capabilities across heterogeneous components.

As part of the general guidelines for Use Case Architecture analysis, each Use Case must be specified by defining and analysing a minimal set of architectural views. One of these, was the description of the Interoperability Endpoints.



Interface name	Exposed by	Protocol	Notes
RFID Reader Interface	RFID Reader	LLRP (over IP, local)	Global EPC Standard
…	…	…	…

Figure 30: IoF2020 Interoperability Endpoints example.

This Interoperability Endpoints View summarizes the main endpoints, which can be exploited to integrate available systems to other systems.

Its main purpose is to help identifying the most suitable entry points to access available legacy and IoT systems, deployed in each UC, referencing the standards and protocols, which must be implemented to perform such integration. An example of the list of Interoperability End-points as exemplified in Figure 30.

While this is not a “standard” view (such information is typically spread across the information, communication and deployment views), it has been adopted to facilitate the identification of technical synergies.

5.3.5 Interoperability Support in MONICA

To ensure interoperability, MONICA has designed the architecture following the AIOTI-High Level Architecture and incorporated open IoT standards such as oneM2M and OCG SensorThing API. In particular, MONICA interoperability is guaranteed by the *IoT Layer* that comprises two open source frameworks: the LinkSmart (IoT middleware) and the SCRAL (IoT abstraction layer). The SCRAL framework provides interoperability over devices. In fact, applications can access any kind of devices whichever proprietary protocol they may speak, over a uniform web-service based interface. In addition, the SCRAL framework exposes available metadata and semantic information for connected devices and streams; thus, enabling the *Service Layer* to access such an information when needed. More specifically, the *SCRAL* performs data modelling according to the open standard *OGC SensorThings API*; thus, addressing the syntactic interoperability of the IoT.

The LinkSmart middleware enables search and discovery of these devices and their resources by platform services and applications. Moreover, it provides unified APIs and protocols for historical and (near) real-time data access between the lower layers (*i.e.* Device Layer and Edge Layer) and the upper one (*i.e.* Service Layer and App Layer) as depicted in MONICA architecture. The main components of the LinkSmart middleware are reported as follows:

- **Event Broker** provides a message bus for efficient asynchronous communication of sensor data streams implementing the publish/subscribe communication pattern. The Message Queue Telemetry Transport (MQTT) [22] is recognized as the de-facto standard for Publish/Subscribe communication in the IoT messaging domain. As a Publish/Subscribe protocol, MQTT provides several features like topic wildcards, different level of quality of service, retained messages, last will and testament, and persistence sessions.
- **Resource Catalog** exposes a lightweight JSON-based RESTful API and provides a registry of integrated ICT data sources, their basic meta-information and deployment configuration, including information on how their data can be accessed. The SCRAL is supposed to register the available devices and their resources so that applications and services can discover these devices and learn how to communicate with them.
- **Service Catalog** has similar functionality as the described above Resource Catalogue with the difference that it provides a registry for middleware and Monica Platform services. Service Catalog enables search and discovery of available platform services by attributes, such that applications and services can dynamically discover required services without prior configuration of the endpoints. In addition to that, the endpoint of the Service Catalog API can be advertised on the network using DNS-SD, which enables automatic system discovery in the deployment environment.
- **Historical Datastore** provides a repository with historical data from integrated sensor systems compliant with the OGC SensorThings v1.0. Thus, it allows access to the historical data from integrated sensor systems is one of the main functional requirements from applications to the middleware.

Finally, the MONICA *IoT Layer* integrates the oneM2M modules allowing the platform exposing to external platforms the IoT data according to the oneM2M standard.

5.3.6 Interoperability Support in SYNCHRONICITY

On top of the Reference Architecture, SYNCHRONICITY has developed the approach of defining Interoperability Points (the main interfaces that allow applications to interact with the supporting platform) and Interoperability Mechanisms (the actual interface specifications at the Interoperability Point).

The Interoperability Points defined in SYNCHRONICITY are listed below:

- Context Management API: the API to access to real-time context information;
- Shared Data Model: Guidelines and catalogue of common data models in different verticals to enable interoperability for applications and systems;
- Market Place API: an API to expose functionalities such as catalog management, ordering management, revenue management, SLA, license management, etc.;
- Security API: an API to register and authenticate user and applications in order to access to the platform-enabled services;
- Data Storage API: an API to allow to access to historical data and open data.

The Interoperability Points are represented in Figure 31:

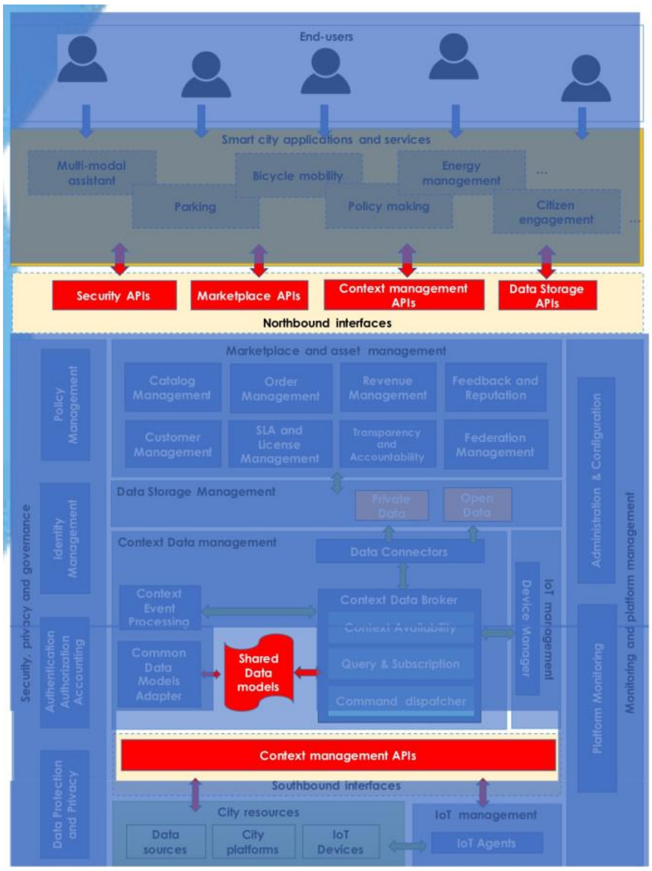


Figure 31: SYNCHRONICITY Interoperability Points

6. Interoperability: Platforms and Technologies

6.1 Introduction

This section addresses the issues related to the platforms used within the various LSPs implementations. Those platforms will be potentially very different in span and scope (even within a single LSP). The purpose of this section is to identify the major platform requirements, the main platforms used and how these platforms can interoperate when necessary.

6.1.1 Considerations for platforms selection and usage

There are hundreds of IoT platforms available for the development of IoT systems. The question of a choice of platform(s) by IoT system designers is complex. Some dimensions have to be considered:

- *Scope and breadth*. Some of the existing IoT platforms may address a specific problem or a limited technical environment, offering a point solution addressing a part of the IoT stacks. On the other hand, some platforms can be very general purpose and integrate the IoT system in a larger (enterprise) system.
- *Maturity and ownership*. The available platforms may have different development status, technical readiness levels and user adoption level. Moreover, they can be proprietary as well as opened (e.g., open source).
- *Standards support*. They available platforms can also have very different support to interoperability and to the standards in support of it

6.1.2 The Service Platform

Amongst the many IoT platforms in use, with a very diverse scope of functionality, the IoT Service Platform plays an important role since its main objective is to provide an abstraction layer between the applications and the IoT devices and to provide a built-in support for a very large number of standards (existing or forthcoming).

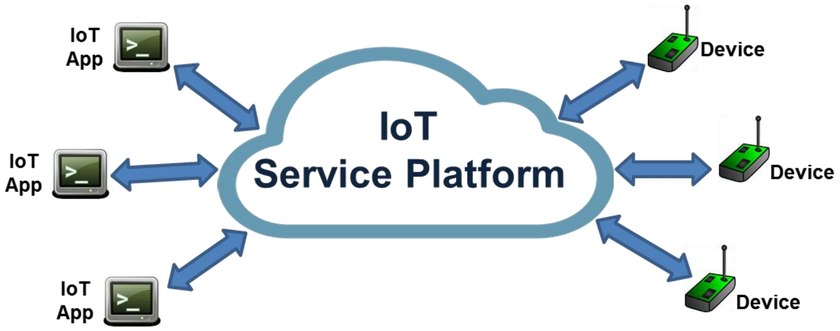


Figure 32: The IoT Service Platform

An IoT Service Platform is essentially:

- An Intelligent layer between applications, networks and devices;
- Offering a coherent set of standardized functionalities;
- And an enabler for communication and data interoperability.

The IoT service platform is the actual implementation/deployment of an abstract IoT architecture (entities and interfaces) like the ones outlined in the previous section.

6.2 Platforms and Technology Support in the LSPs Use Cases

In the context of the IoT LSPs Use Cases, the issues related to the choice and usage of the platforms can vary from one LSP to another for a variety of reasons, such as:

- The requirements on cross-site interoperability may be more or less stringent;
- The support of connectivity or applications can induce specific choices;
- Local interworking with legacy applications can require pre-defined choices or the use of specific adaptors;

The Figure 33 shows the example of the platforms and technologies for AUTOPILOT. In this case, the requirement of interoperability across pilot sites has led to the choice of oneM2M as a unifying IoT service platform. On top of this, the particularities of each pilot site (such as the nature of the application deployed or the legacy platforms) may lead to the choice of site-specific platforms and technologies.

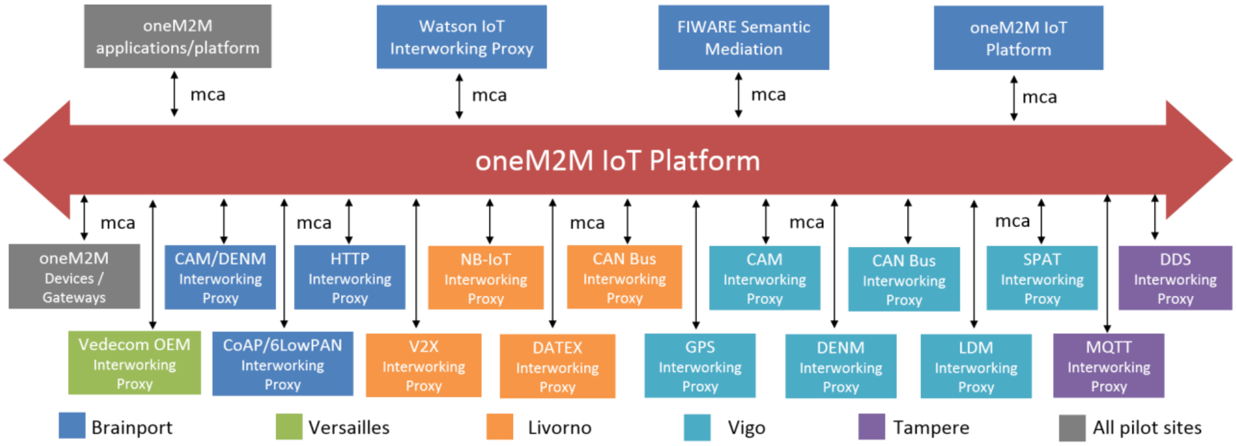
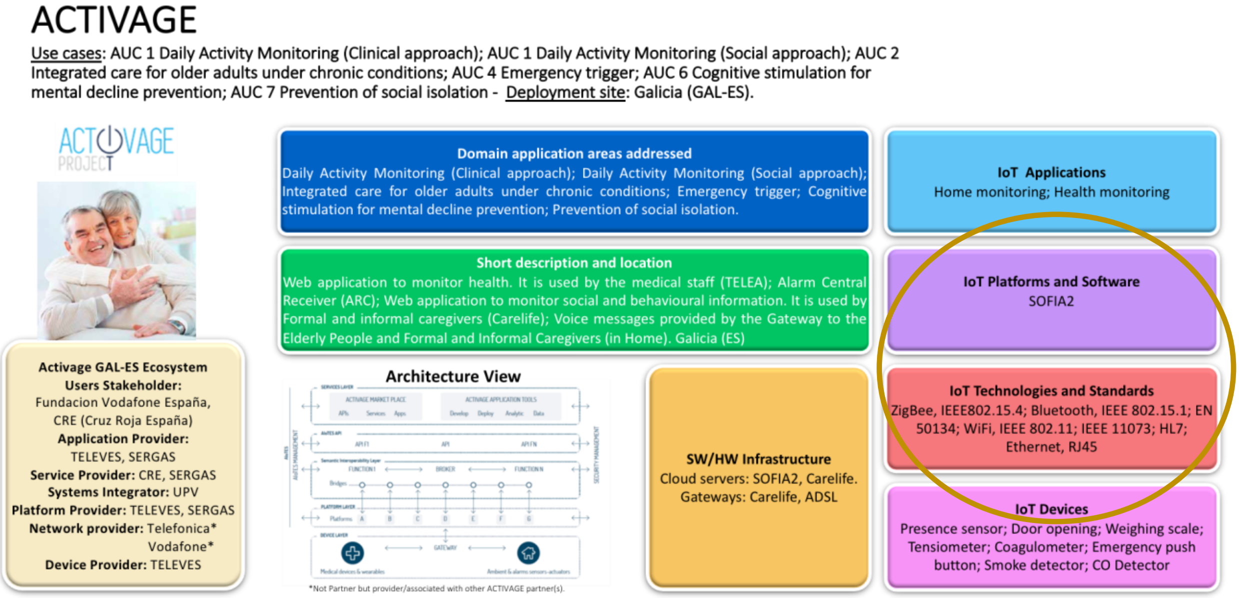


Figure 33: The platforms across the AUTOPILOT use cases

As already pointed out, the IoT LSPs have provided information on the various Use Cases by a Use Case template such as the one shown as an example (from ACTIVAGE) on Figure 6. The template shows the main Platforms and Software used in a given Use Case and, in a few cases, some technologies (in the "IoT Technologies and Standards" box).



ACTIVAGE

Use cases: AUC 1 Daily Activity Monitoring (Clinical approach); AUC 1 Daily Activity Monitoring (Social approach); AUC 2 Integrated care for older adults under chronic conditions; AUC 4 Emergency trigger; AUC 6 Cognitive stimulation for mental decline prevention; AUC 7 Prevention of social isolation - Deployment site: Galicia (GAL-ES).

Domain application areas addressed
Daily Activity Monitoring (Clinical approach); Daily Activity Monitoring (Social approach); Integrated care for older adults under chronic conditions; Emergency trigger; Cognitive stimulation for mental decline prevention; Prevention of social isolation.

IoT Applications
Home monitoring; Health monitoring

Short description and location
Web application to monitor health. It is used by the medical staff (TELEA); Alarm Central Receiver (ARC); Web application to monitor social and behavioural information. It is used by Formal and informal caregivers (Carelife); Voice messages provided by the Gateway to the Elderly People and Formal and Informal Caregivers (in Home). Galicia (ES)

IoT Platforms and Software
SOFIA2

Activage GAL-ES Ecosystem
Users Stakeholder:
Fundacion Vodafone España, CRE (Cruz Roja España)
Application Provider:
TELEVES, SERGAS
Service Provider: CRE, SERGAS
Systems Integrator: UPV
Platform Provider: TELEVES, SERGAS
Network provider: Telefonica*
Vodafone*
Device Provider: TELEVES

Architecture View

SW/HW Infrastructure
Cloud servers: SOFIA2, Carelife.
Gateways: Carelife, ADSL

IoT Technologies and Standards
ZigBee, IEEE802.15.4; Bluetooth, IEEE 802.15.1; EN 50134; WiFi, IEEE 802.11; IEEE 11073; HL7; Ethernet, RJ45

IoT Devices
Presence sensor; Door opening; Weighing scale; Tensiometer; Coagulometer; Emergency push button; Smoke detector; CO Detector

*Not Partner but provider/associated with other ACTIVAGE partner(s).

Figure 34: Platforms, Software and Technologies in the use case template (an example)

This information on Platforms and Technologies is provided in tables (one for each LSP) with the following columns:

- Platform or technology: short name of the identified element;
- #UC:	number of Use Cases where the platform/technology is used;

- Owner: Entity in charge of platform development and maintenance;
- Status: P for Proprietary; S for Standards-based; O for Open Source
- G/S: G for a Generic (i.e. cross-sector) platform; S for a (sector) Specific platform

Some of the elements in the tables correspond to technologies and are in *italic* font.

6.2.1 ACTIVAGE

6.2.1.1 Use Cases and the support of Platforms and Technologies

The idea of an Internet-of-Things (IoT) platform is to provide the intelligent environment that interconnects the physical world with the digital world. Rephrasing, IoT platform (IoT middleware) is an integrated physical/virtual entity system that enables the communication between the machines and devices and then the acquisition, processing, transformation, organization and storing machine and the sensor data. The IoT platform employs various applications and components to provide fully interoperable IoT services and management of those. This includes, networks, IoT environments, IoT devices (sensors, controllers, actuators, tags and tag readers, gateways) and the attached physical devices, IoT operations and management, and external connectivity with suppliers, markets and temporary stakeholders of the IoT system.

The IoT platforms provide the ability on creating application and services in a structure environment that is formed by functional components. The proven existing blocks are usually common and repeated across many IoT applications and services as shown in Figure 35. This IoT platform feature contributes to the development cycle and time to market while reduces the overall cost of the IoT implementation.

- Connectivity & normalization: eliminates the heterogeneous data as it brings different protocols and different data formats into one software interface ensuring accurate data streaming and interaction with all devices.
- Device management: ensures the connected the network devices are working properly, seamlessly running patches and updates for software and applications running on the device or edge gateways.
- Data Storage: scalable storage of device data brings the requirements for hybrid cloud-based databases to a new level in terms of data volume, variety, velocity and veracity.
- Processing & action management: brings data to life with rule-based event-action-triggers enabling execution of intelligent actions based on specific sensor data.
- Analytics: performs a range of complex analysis from basic data clustering and deep machine learning to predictive analytics extracting the most value out of the IoT data-stream.
- Visualization: enables humans to see patterns and observe trends from visualization dashboards where data is vividly portrayed through line-, stacked-, or pie charts, 2D- or even 3D-models.

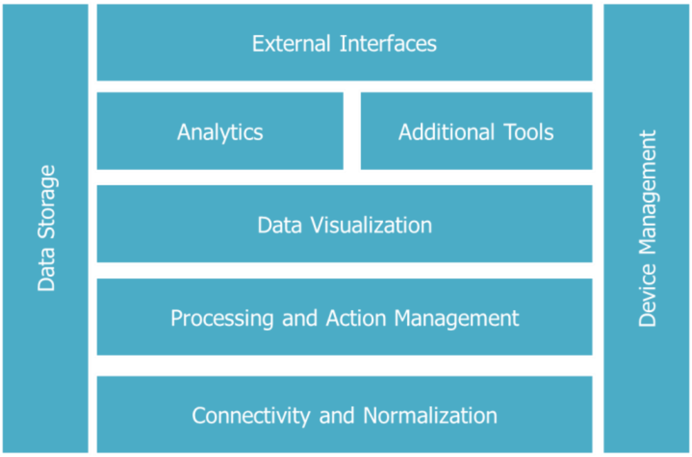


Figure 35. Functional components of IoT platforms [6].

- Additional tools: allow IoT developers prototype, test and market the IoT use case creating platform ecosystem apps for visualizing, managing and controlling connected devices.
- External interfaces: integrate with 3rd-party systems and the rest of the wider IT-ecosystem via built-in application programming interfaces (API), software development kits (SDK), and gateways.

For the purposes of the ACTIVAGE project, platforms focusing on the application layers are desired, offering means to transform the information received from the devices and sensors into meaningful knowledge. Thus, the selection should be based on platforms residing on the right or top part of the above diagram.

A second criterion is the availability of the platforms, since open-source solutions are preferred to proprietary solutions. Thus, platforms such as Microsoft Azure, IBM Watson or Oracle Integration Cloud are not preferred.

The following platforms are selected to be used in the ACTIVAGE project:

- universAAL
- Sofia2
- OpenIoT
- sensiNact
- FIWARE
- IoTivity
- Seniorsome

These platforms comprise a comprehensive set of open-source, application-oriented IoT platforms, covering a relatively wide range of functionalities.

Furthermore, there is a past experience for using these platforms in other European projects, which will be of use throughout the ACTIVAGE project. ACTIVAGE has developed (and documented) 9 Use Cases. The standards in support of these Use Cases are mapped in the table below.

Table 4: Platforms and Technologies Support in ACTIVAGE Use Cases

Platform or Technology	#UC	Owner	Status	G/S
FIWARE	3,5,6	FIWARE Foundation	S/O	G
IoTivity	1,6,8	IoTivity	O	G
OpenIoT	1,5,7	OpenIoT	O	G
sensiNact	1,3,4	LETI	O	G
Seniorsome	1,2,7	Finland Health	P	S
SOFIA2 IoT Platform	1,2,8	SOFIA2	O	G
universAAL	1,4,9	universAAL	O	G

6.2.1.2 Comments

The IoT platform itself can be located in the cloud, located on premise or involve a combination of both. Cloud-based IoT platforms are significant as they are user based and has successful record of exploitation.

Through their design they can offer flexibility by the implementation in different contexts, holistic usability by all possible users and productivity as they use efficiently all the types of available services.

Additional services of the IoT platform can include resource interchanges to enable access to resources outside of the IoT system, network services, cloud integration services and many other services as defined by the individual platform provider.

Summarizing the components, the functionality of IoT platforms is illustrated in Figure 36.

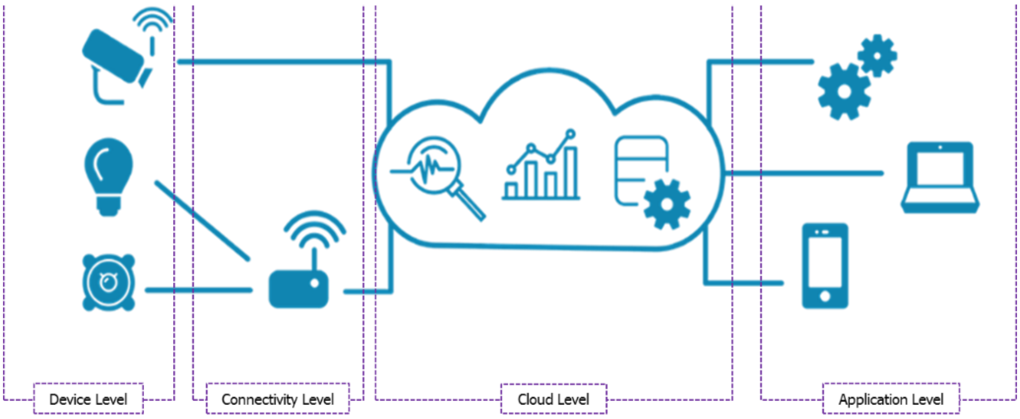


Figure 36. IoT platform levels.

The IoT platform components ensures that the communication between the device and the output accomplishes, that data is collected and formatted correctly and functions such as remote updates and access are facilitated. Every Level of IoT platform utilizes a group of elements to deliver efficiently the proper output to the next Level of operations. More explicitly:

- *Device* Level (Data Generations & Collection). At this Level the data are generated through various IoT sources. It constitutes the logic consolidation process, the service operations and the data standardization. As the data are originated from different type of sources they variate in format and periodicity. Subsequently they have various security, privacy, and quality requirements. Moreover, the sizes of the data concerning sensors are most of the types, more than the actual measure. To deal with that, filtering techniques are established to the current Level in order to filter the important information. Integrated Development Environments (Java, HTML5), IoT data model and execution engine, Workflow and business logic modeller, mobile applications, search applications and security/ authentication systems are some features forming this Level.
- *Connectivity* Level (Communication Technologies). This Level includes all the hardware and software within network and enterprise that facilitates any type of connection with the cloud. The communication between sensors and from sensors to relay nodes can occur through ZigBee technology Then Internet using various network infrastructure technologies, such as Wi-Fi, WiMAX, LTE, 3G, etc. facilitate the communication. Ethernet is used between various analysis servers. M2M/Data acquisition, Device Management, Complex event processing, Condition based Monitoring, Data transport and security/ authentication systems are some additional systems utilized by the Connectivity Level.
- *Cloud* Level (Data Management and Processing) is the virtualised and optimised hosting where the data is sent from the device and where it should be formatted for data processing (data management, data analytics, etc.). It also facilitates remote device management and removes software updates. Cloud computing has a pivotal role in the IoT platform structure. A secure interoperable frame of data management is created across all the correlated elements. Using the capabilities of the aforementioned technology, data are transferred in storage where in storage where they are accessible to people and analytic applications,
  - *Application* Level (Data Interpretation) is where the output should be sent. This level includes smart devices, sensors, different type of smart components, mobile apps or an internal system and forms that the data will be sent. Additionally, the Level includes the use of statistical and optimization tools to refine, monitor, and analyse structured and unstructured data for enabling different services. At this level methods and algorithms for Bid Data management, collection and annotation are established. Their main objective is the self-Organization and self-Management towards the evolution of IoT. Statistical Programming, Text and data mining, Image and video processing, Predictive models, Machine and Deep learning algorithms, Optimizing and Simulation and Visual analytics are some significant examples of data formation.

6.2.2 AUTOPILOT

6.2.2.1 Use Cases and the support of Platforms and Technologies

AUTOPILOT has developed (and documented) 14 Use Cases. The standards in support of these Use Cases are listed in the table below.

Table 5: Platforms and Technologies Support in AUTOPILOT Use Cases

Platform or Technology	#UC	Owner	Status	G/S
FIWARE	1	FIWARE Foundation	S/O	G
FIWARE Semantic	3	FIWARE Foundation	O	G
Huawei IoT	3	Huawei	P	G
Kuantic Server	2	Kuantic	P	S
oceanConnect	1	Huawei	P	G
oneM2M	12	oneM2M	S	G
PEXSI Platform	2	VEDECOM	P	S
Raspberry PI	1	Raspberrypi.org	P/O	G
Sensinov oneM2M	4	Sensinov	P/S	G
Watson IoT	6	IBM	P	G

6.2.2.2 Comments

The development and integration of IoT devices used in different AUTOPILOT use cases are used to support autonomous driving functions. IoT sensor devices and mobile IoT objects (mobile robots and/or micro aerial vehicles) are integrated with the IoT infrastructure (sensor/actuators, connectivity and communication) developed for different use cases and seamlessly deployed into the five pilot sites that are forming various IoT ecosystems including other IoT devices, vehicle IoT platforms and Open IoT platforms.

The AUTOPILOT IoT service platform is a federation of several IoT platforms, allowing it to be **open** and **flexible**. An open oneM2M standard IoT platform, referred to as the *oneM2M interoperability platform*, interconnects these *proprietary* IoT platforms provided by the project partners. The proprietary IoT platforms collect data from connected devices. They exchange IoT data and events with the interoperability platform through oneM2M interworking proxies.

Currently, several IoT platforms have been deployed for the pilot sites:

- FIWARE IoT platform, used in the Dutch pilot site
- Watson IoT Platform used in the Dutch and Spanish pilot sites
- HUAWEI OceanConnect IoT platform, used in the Dutch pilot site
- TIM oneM2M IoT platform, used in the Italian pilot site
- SENSINOV oneM2M platform, used in the Finnish, French and Dutch pilot sites

oneM2M Interworking proxies are currently being developed for the non-oneM2M-compliant platforms.

Across the pilot sites, devices are currently being connected to the IoT platforms. The AD functionality is being adapted to support IoT data for all the project use cases.

To facilitate interoperability between the IoT platforms and to make the use cases pilot-site-independent, a Data Modelling Activity Group (DMAG) is specifying a common data model for the whole project, based on existing standards: SENSORIS for vehicle messages and detection events and DATEX II for parking and traffic information.

Ontologies may be used in AUTOPILOT to define controlled vocabularies and semantic mappings for some of the data model field values. This would allow for flexibility and openness, while facilitating interoperability. Several ontologies have been reviewed so far, but the actual work on developing the ontologies has not started yet. It is planned to start in the third quarter of 2018.

The development and integration of IoT devices into IoT ecosystem are adapted to the pilot sites infrastructure. The use cases map the AUTOPILOT architecture and the IoT devices are integrated into different architecture components and interfaced/connected to the use case and pilot site IoT platform (infrastructure, connectivity, services, etc.). IoT devices used in different AUTOPILOT use cases support/enhance the autonomous driving functions. The IoT devices used are adapted to the autonomous driving function requirements in terms of speed of access (latency), availability and range (covered area).

6.2.3 IoF2020

6.2.3.1 Use Cases and the support of Platforms and Technologies

IoF2020 has developed (and documented) 19 Use Cases. The platform and technologies in support of these Use Cases are listed in the table below.

Table 6: Platforms and Technologies Support in IoF2020 Use Cases

Platform or Technology	#UC	Owner	Status	G/S
365FarmNet	6	365FarmNet	P	S
AgroSense	1	Corizon	O	S
Apache Cassandra	1	Apache	O	G
Apache Flink	1	Apache	O	G
Apache Spark	1	Apache	O	G
Arvalis IoT Platform	1	Arvalis	P	S
Atland FMIS	2	Atland	P	S
Connecterra IoT	1	Connecterra	P	S
Cygnus	1	FIWARE	O	G
EBBITS	1	ISMB	O	G
EPCIS	3	GS1	S	G
FIWARE (in particular Broker)	8	FIWARE Foundation	O	G
FISpace	1	FISpace	P	S
LinkSmart (Free, Open Source IoT Platform)	1	Fraunhofer	O	G
MongoDB	1	mongoDB	P	G
OpenStack	1	OpenStack	O	G
Qlip platform for automatic calibration and validation	1	Qlip	P	S
ThingWorx IoT	1	ThingWorks	P	G
VIRTUS (XMPP Based Architecture for Secure IoT)	1	ISMB	O	G

6.2.3.2 Comments

In IoF2020, is clear the fact that the field of IoT platforms is unconsolidated and this reflects on the choices made in the project. The use cases use many platforms. And a relatively high number of use cases report they use 365FarmNet and/or FIWARE. Also, important, is that aspects such as device management, enrolment, firmware deployment/upgrades and decentralized security models are important aspects and need early consideration when deploying IoT at scale. The configuration management strategies that are completely feasible when deploying tens of devices, break when a solution is scaled to thousands of devices.

With FIWARE being used relatively often and being a very modular platform, it was also studied which components from the FIWARE ecosystem are used specifically:

- Context broker - A broker that allows sharing objects and their properties, and supports updates, queries, registrations and subscriptions.
- Device management and IoT Agent - This component collects data from devices using heterogeneous protocols and translates them into the standard platform language suitable for the context broker.
- Identity management - A generic component that supports authentication tasks for users' access to networks, services and applications, including secure and private authentication from users to devices, networks and services.

6.2.4 MONICA

6.2.4.1 Use Cases and the support of Platforms and Technologies

MONICA has developed (and documented) 4 Use Case Groups, namely:

- Sound Monitoring and Control
- Crowd and Capacity Monitoring and Management
- Missing Persons/Locate Staff Members
- Health/Security Incidents

The platforms and technologies in support of these Use Cases Groups are listed in the table below.

Table 7: Platforms and Technologies Support in MONICA Use Cases

Platform or Technology	#UC	Owner	Status	G/S
ASFCS (Adaptive Sound Field Control System)	1	DTU	P	S
Sound Lever Meter GW	1	B&K	P	S
SCRAL (Smart City Resource Adaptation Layer)	1-4	ISMB	O	G
LinkSmart	1-4	linksmart.eu	O	G
GOST platform (implements OGC SensorThings API)	1-4	github.com/gost/server	O	G
Mosquitto - MQTT broker	1-4	mosquitto.org	O	G
oneM2M	1-4	oneM2M	S	G
RIOT	1, 2	riot-os.org	O	G
Raspberry PI	1, 2	Raspberrypi.org	P/O	G
UWB-based wristbands and GW	2-4	DEXELS	S	S
Docker	1-4	Docker, Inc.	P	G
SFN (Security Fusion Node)	2, 4	Kingston University	P	S

6.2.4.2 Comments

As listed in the table above, MONICA has adopted and integrated different IoT platforms and technologies to demonstrate the 4 Use Case Groups (UCGs). These UCGs have been demonstrating in 6 EU cities (Turin, Bonn, Hamburg, Copenhagen, Leeds, Lyon) and in 11 different events including concerts, festival and sports event. However, the developed IoT platform, which is composed of the LinkSmart middleware and the SCRAL adaptation framework, is the same used in all UCGs. The IoT platform integrated the GOST (Go-SensorThings) IoT server that implements the sensing profile (part 1) of the OGC SensorThings API standard including the MQTT extension.

Thanks to the MONICA IoT platform, various IoT devices have been integrated such as Sound Level Meters from B&K through its GW, and environmental sensors (based on RIOT) through its GW running in the Raspberry PI platform. Regarding the wearable devices, wristbands (based on the UWB standards and 868 MHz radio chip) and smart glasses have been integrated too. In addition, other IoT systems such as the AFCS (used to control the sound in the concert area and to reduce the sound limit in the neighbourhood), the Security Fusion Node (used to collect results from different video-based algorithms) and other external IoT smart city platforms have been integrated. One of these smart city platforms is the Hamburg one that will be integrated thanks to the oneM2M module of the MONICA platform.

The IoT Layer components as well as the upper layers ones have been deployed on a cloud platform (MCP) that uses virtual machines as level of virtualization. The MCP uses two levels of virtualization - virtual machines and containers. The base virtual machine is CentOS 7, and the base container technology is Docker 1.12.6.

6.2.5 SYNCHRONICITY

6.2.5.1 Use Cases and the support of Platforms and Technologies

SYNCHRONICITY has developed (and documented) 6 Use Cases. The standards in support of these Use Cases are listed in the table below.

Table 8: Platforms and Technologies Support in SYNCHRONICITY Use Cases

Platform or Technology	#UC	Owner	Status	G/S
CKAN	2	IoT Lab	O	G
Cygnus	2	FIWARE	O	G
IDAS	2	FIWARE	O	G
OpenDataSoft	1	OpenDataSoft	P	G
OpenTripPlanner	2	OpenTripPlanner	O	S
Organicity API	2	Organicity		S
Orion Context Broker	5	FIWARE	O	G
STH Comet	2	FIWARE	O	G
WSO2	1	WSO2	O	G

7. FINDINGS AND FUTURE WORK

7.1 Early lessons learned

The LSPs have reached the half of their duration at the time of the writing of the present document. For

- Despite a large variety of use cases spanning very diverse sectors, it appears that the LSPs have agreed on relatively similar functional architectures that can be mapped on a common architecture. In particular, all the LSPs share a common view of the functional layers (with 8 layers). This functional architecture is, for most of the LSPs, complemented by the adoption of cross-cutting functions (e.g., security or data management) that is introducing another dimension for a more precise evaluation of the IoT system properties (e.g. integrability or scalability);
- A number of mechanisms in support of interoperability have been defined by the LSPs. If the usage of APIs is largely accepted and documented, some extensions like the Marketplaces are still under evaluation and do not seem to be subject to generalization rapidly;
- There is a very large diversity of platforms due to the highly heterogeneous nature of the LSPs use cases. The selection of platforms between proprietary, standardized or Open Source solutions is conditioned more by the need to integrate with legacy than by the application of objective selection criteria. From this standpoint, the evaluation of the role and benefits of the standardized platforms would be an important information in order to help those in charge of the selection of platforms.

7.2 Future Workshops on Standards

The present report is reflecting the progress of interoperability at large and of specific aspects (e.g., reference architectures, interoperability mechanisms, platforms and technologies) activities at the end of the first half of the LSPs.

The work is going to continue

- Within the LSPs, in particular for the resolution of the major standards gaps, and
- Within the LPS Activity Group 02 (Standardisation, Architecture and Interoperability) which will organise more Workshops (1 in 2018 and 3 in 2019) with a series of associated CREATE-IoT deliverables.

A final deliverable (D06.03 "Assessment of convergence and interoperability in LSP platforms") will summarize the global findings at the end of (most of) the LSPs at the end of 2019.

8. REFERENCES

[1] "Semantic Interoperability Manifesto" IERC 2015, Online at http://www.internet-of-things-research.eu/pdf/IERC_Position_Paper_IoT_Semantic_Interoperability_Final.pdf

[2] "Initial Report on IoT Standardisation activities", CREATE-IoT, Deliverable D06.05, 2018.

[3] "Strategy and coordination plan for IoT interoperability and pre-normative and standardisation activities ", CREATE-IoT, Deliverable D06.07, 2018.

[4] IoT European Large-Scale Pilots Programme Team. Large-Scale Pilots Projects. Online at: https://european-iot-pilots.eu/resources/iot-lsps-brochures/

[5] "Reference architecture for federation and cooperation between IoT deployments", CREATE-IoT, Deliverable D02.02, 2018.

[6] "Internet of Things Reference Architecture (IoT RA)", ISO/IEC CD 30141, Online at: https://www.iso.org/standard/65695.html.

[7] ACTIVAGE (ACTivating InnoVative IoT smart living environments for AGEing well); https://european-iot-pilots.eu/project/activage/

[8] AUTOPILOT (AUTOmated driving Progressed by Internet Of Things); https://european-iot-pilots.eu/project/autopilot/

[9] IoF2020 (Internet of Food and Farm 2020); https://european-iot-pilots.eu/project/iof2020/

[10] MONICA (Management Of Networked IoT Wearables); https://european-iot-pilots.eu/project/monica/

[11] SYNCHRONICITY (Delivering an IoT enabled Digital Single Market for Europe and Beyond). Online at: https://european-iot-pilots.eu/project/synchronicity/

[12] F. Visintainer, L. Altomare, et. al. *Initial Open IoT Vehicle Platform Specification*. AUTOPILOT Deliverable D1.5, October 2017.

[13] G. Larini, et. al. *Standardisation plan*. AUTOPILOT Deliverable D5.7, May 2017.

[14] "Reference Architecture Model Industrie 4.0", Platform Industrie 4.0, on-line at: http://www.plattform- i40.de/I40/Navigation/EN/Home/home.html).

[15] "Industrial Internet Reference Architecture (IIRA)", Industrial Internet Consortium, on line at: http://www.iiconsortium.org/IIRA.htm

[16] "Architecture Alignment and Interoperability", Joint Industrie 4.0 and IIC White Paper, on-line at: https://www.iiconsortium.org/pdf/JTG2_Whitepaper_final_20171205.pdf

[17] "Advancing IoT Platforms Interoperability", River Publishers, Gistrup, 2018, 978-87-7022-005-7 (ebook), IoT European Platforms Initiative (IoT-EPI) White Paper, online at: https://iot-epi.eu/wp-content/uploads/2018/07/Advancing-IoT-Platform-Interoperability-2018-IoT-EPI.pdf

[18] AIOTI WG3 – loT Standardisation, "High Level Architecture (HLA)", Release 3.0, June, 2017. Download High Level Architecture (HLA) Release 3.0 June 2017.

[19] Dalhousie University, Faculty of Medicine, Geriatric Medicine Research. *Clinical Frailty Scale*, March 2016. Online at: http://geriatricresearch.medicine.dal.ca/clinical_frailty_scale.htm

[20] Alliance for Internet of Things Innovation (AIOTI). Online at: https://aioti.eu

[21] ARM-IoT. Online at: https://developer.arm.com/products/architecture/system-architecture

[22] MQTT - M2M/IoT connectivity protocol. Online at: http://mqtt.org/