High Level Architecture (HLA)

Release 5.0

AIOTI WG Standardisation

December 2020

Table of Contents

Figures

Release history

Release	Date of publication	Major enhancements
3.0	June 2017	
4.0	June 2018	New clause 6 (Identifiers for IoT), updated clause 7.3.5 (Big Data related initiatives), new clause 7.4 (Privacy aspects), updated clause 7.5 (Virtualization), new clause 8.5 (Big Data Value Association)
5.0	December 2020	New (renumbered) clause 7.4 (Security aspects), updated (renumbered) clause 7.6 (Virtualization), new (draft) clause 7.7 (IoT platforms), new clause 8.6 (3D Layered Architecture), new clause 9.3 (IoT-enabled Data Marketplaces)

1. Highlights and recommendation

In the context of the AIOTI WG Standardisation (AIOTI WG03) and by following the evolution on IoT Architectural aspects and available specifications, AIOTI WG Standardisation has developed a High Level Architecture (HLA) for IoT that should be applicable to AIOTI Large Scale Pilots. The HLA takes into account existing SDOs and alliances architecture specifications. This document is an integral part of a set of deliverables from AIOTI WG03 that also cover other aspects such as IoT landscape and Semantic Interoperability.

AIOTI WG03 recommends that the HLA be the basis for further discussion with the Large Scale Pilot (LSP) and AIOTI WGs in order to promote architectural convergence with SDOs, alliances, consortia and other relevant parties.

NOTE – In line with the AIOTI WG03 engagement model, other relevant parties include - but are not limited to open source projects, policy makers, regulators, pilots and test beds, research organizations, companies.

Further development of the HLA should be an incremental exercise taking into account the LSP WGs' feedback, however it should remain high level and not compete with established SDOs, alliances and open source projects.

2. Objectives of this document

This document provides a proposal for a high-level IoT architecture to serve as a basis for discussion within AIOTI, referred to as the AIOTI HLA (High-level architecture). The proposal results from discussions within the AIOTI WG03 and takes into account the work of SDOs, Consortia, and Alliances in the IoT space. Throughout the proposal, AIOTI WG03 has kept in mind the need to support instantiation for all Large Scale Pilot deployments.

This document:

- Introduces the use of ISO/IEC/IEEE 42010 by AIOTI WG03
- Presents a Domain Model and discusses the "thing" in IoT
- Presents a Functional Model
- Introduces the Identifiers for IoT
- Provides deployments considerations related to relevant IoT architectural matters such as cloud and edge computing, Big Data, virtualization, security, privacy and (platform) interoperability
- Links this work with the AIOTI WG03 Semantic Interoperability work and the SDO Landscape work
- Provides mapping examples to some existing SDO/Alliances' architectural work related to functional models: ITU-T, oneM2M, IIC, BDVA.
- Establishes the link to other architectures and frameworks such as Big Data and IoT-enabled Data Marketplaces

The annexes provide different types of information, including possible relationships of the HLA functional model with other models.

NOTE 1 - The main enhancements of Release 4.0 of this document from its previous Release (R3.0, June 2017) concern Identifiers for IoT, Privacy, Virtualization and Big Data related aspects.

NOTE 2 - The main enhancements of Release 5.0 of this document from its previous Release (R4.0, June 2018) concern Security, Virtualization, IoT platforms, 3D Layered Architecture and IoT-enabled Data Marketplaces related aspects.

Based on past discussions within AIOTI WG03, the following Release(s) of this document will potentially provide enhancements on the following new or partially developed topics, still with respect to IoT architectural concerns: Artificial Intelligence for IoT, Autonomous Systems and IoT, IoTplatforms, 3D Layered Architecture. In this perspective, the present document contains some placeholder (empty) clauses for some potential new or partially developed topics.

3. Use of ISO/IEC/IEEE 42010

A key recommendation from AIOTI WG03 is that architectures should be described using the ISO/IEC/IEEE 42010 standard. This standard motivates the terms and concepts used in describing an architecture and provides guidance on how architecture descriptions are captured and organized.

ISO/IEC/IEEE 42010 expresses architectures in terms of multiple views in which each view adheres to a viewpoint and comprises one or more architecture models. The ISO/IEC/IEEE 42010 standard specifies minimal requirements for architecture descriptions, architecture frameworks, architecture description languages and architecture viewpoints.

AIOTI WG03 recommends using ISO/IEC/IEEE 42010 to capture relevant views and supporting models.

The AIOTI HLA described in this document puts the "thing" (in the IoT) at the centre of value creation. While the body of the proposal is consistent with ISO/IEC/IEEE 42010, AIOTI WG03 does not provide a complete architecture description for IoT which conforms to the standard. Figure 3-1 provides an overview of architectural models as described in ISO/IEC/IEEE 42010.

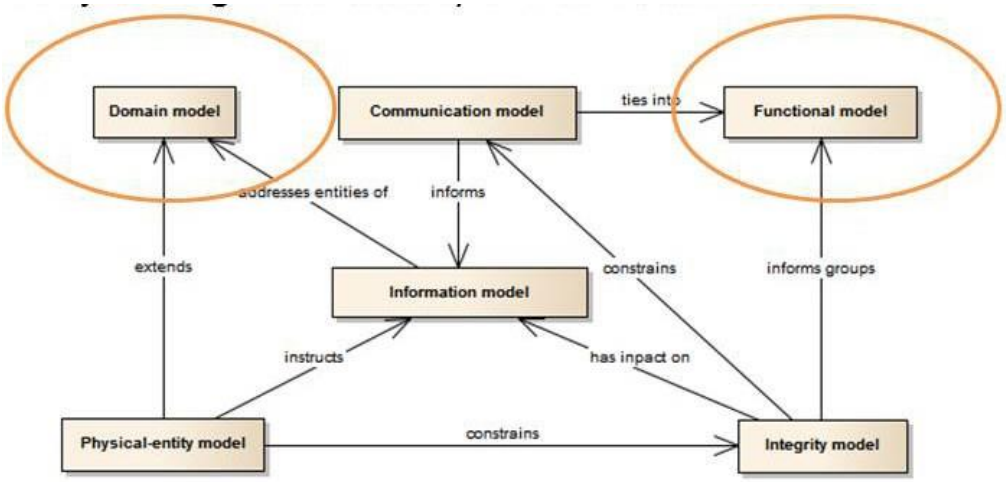


Figure 3-1: Architectural Models based on ISO/IEC/IEEE 42010

With respect to Figure 3-1, AIOTI WG03 focuses its recommendations on the Domain and Functional models (while other models can be considered for future releases of this document):

- The Domain Model describes entities in the IoT domain and the relationships between them.
- The Functional Model describes functions and interfaces (interactions) within the IoT domain.

4. AIOTI Domain Model

The AIOTI Domain Model is derived from the IoT-A Domain Model. A more detailed description of the IoT-A domain model is available under this reference [1].

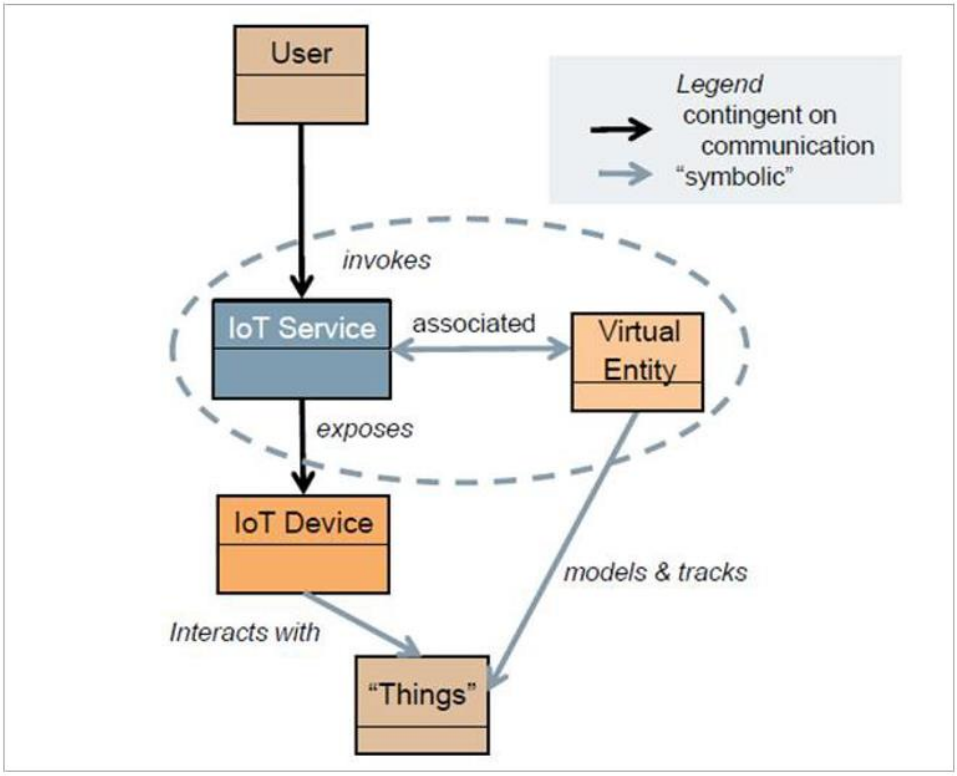


Figure 4-1: Domain Model

The domain model captures the main concepts and relationships in the domain at the highest level. The naming and identification of these concepts and relationships provide a common lexicon for the domain and are foundational for all other models and taxonomies.

In this model, a User (human or otherwise) interacts with a physical entity, a Thing. The interaction is mediated by an IoT Service which is associated with a Virtual Entity, a digital representation of the physical entity. The IoT Service then interacts with the Thing via an IoT Device which exposes the capabilities of the actual physical entity.

5. AIOTI Functional model

The AIOTI Functional Model describes functions and interfaces (interactions) within the domain.

Interactions outside of the domain are not excluded, e.g. for the purpose of using a big data functional model.

5.1 AIOTI layered approach

The functional model of AIOTI is composed of three layers as depicted in Figure 5-1:

- The Application layer: contains the communications and interface methods used in process- to-process communications
- The IoT layer: 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'sservices.
- The Network layer: the services of the Network layer 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.

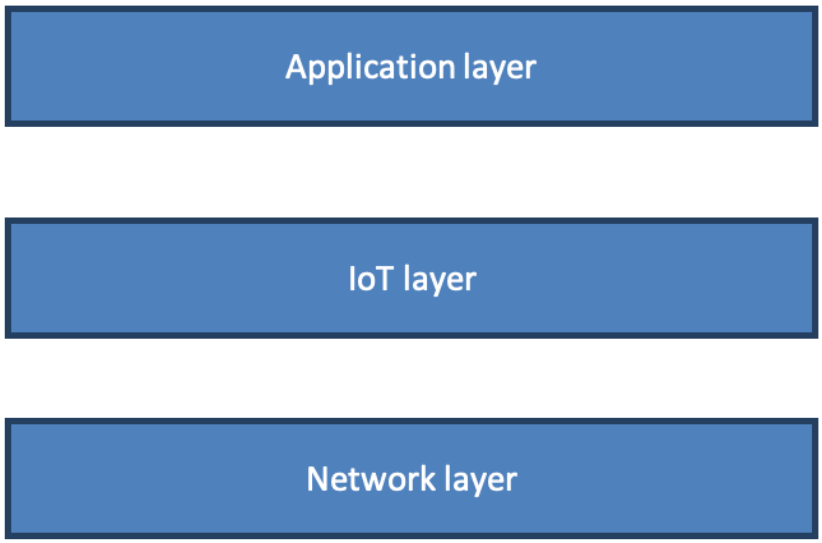


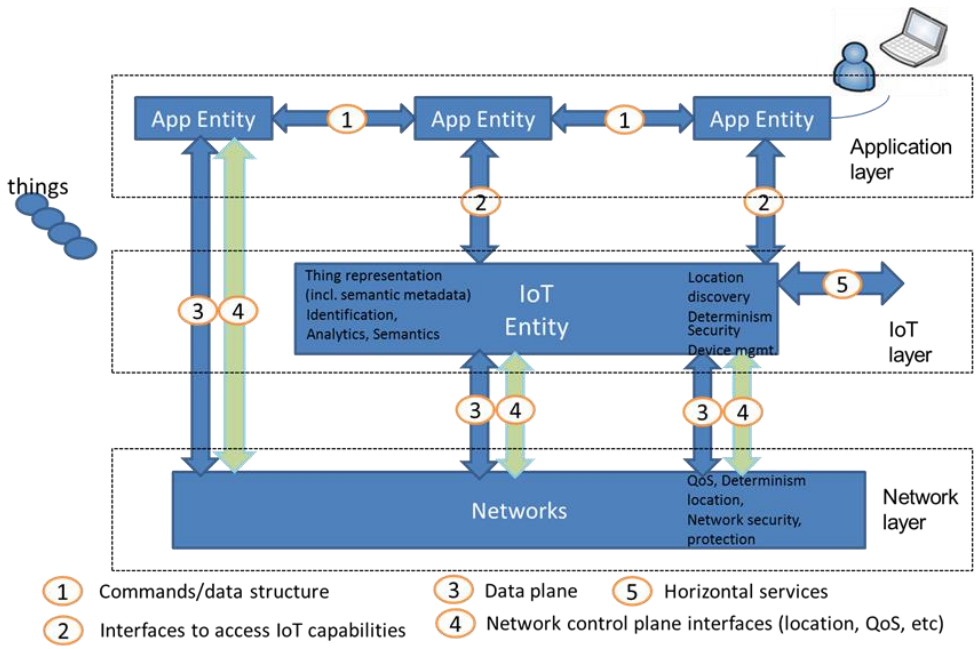
Figure 5-1: AIOTI three-layer functional model.

NOTE - The term layer is used here in the software architecture sense. Each layer simply represents a grouping of modules that offer a cohesive set of services; no mappings to other layered models or interpretation of the term should beinferred.

5.3 AIOTI High level functional model

The AIOTI functional model describes functions and interfaces between functions of the IoT system. Functions do not mandate any specific implementation or deployment; therefore, it should not be assumed that a function must correspond to a physical entity in an operational deployment. Grouping of multiple functions in a physical equipment remains possible in the instantiations of the functional model. Figure 5-2 provides a high level AIOTI functional model, referred to as the "AIOTI HLA functional model".

Figure 5-2: AIOTI HLA functional model



Functions depicted in Figure 5-2 are:

- App Entity: is an entity in the application layer that implements IoT application logic. An App Entity can reside in devices, gateways or servers. A centralized approach shall not be assumed. Examples of App Entities include a fleet tracking application entity, a remote blood sugar monitoring application entity, etc.
- IoT Entity: is an entity in the IoT layer that exposes IoT functions to App Entities via the interface 2 or to other IoT entities via interface 5. Typical examples of IoT functions include: data storage, data sharing, subscription and notification, firmware upgrade of a device, access right management, location, analytics, semantic discovery etc. An IoT Entity makes use of the underlying Networks' data plane interfaces to send or receive data via interface 3. Additionally, interface 4 could be used to access control plane network services such as location or device triggering.

- Networks: may be realized via different network technologies (PAN, LAN, WAN, etc.) and consist of different interconnected administrative network domains. The Internet Protocol typically provides interconnections between heterogeneous networks. Depending on the App Entities needs, the network may offer best effort data forwarding or a premium service with QoS guarantees including deterministic guarantees.

According to this functional model a Device can contain an App Entity and a Network interface, in this case it could use an IoT Entity in the gateway for example. This is a typical example for a constrained device. Other devices can implement an App Entity, an IoT Entity and a Network interface.

Interfaces depicted in Figure 5-2 are:

- 1: defines the structure of the data exchanged between App Entities (the connectivity for exchanged data on this interface is provided by the underlying Networks). Typical examples of the data exchanged across this interface are: authentication and authorization, commands, measurements, etc.
- 2: this interface enables access to services exposed by an IoT Entity to e.g. register/subscribe for notifications, expose/consume data, etc.
- 3: enables the sending/receiving of data across the Networks to other entities.
- 4: enables the requesting of network control plane services such as: device triggering (similar to "wake on lan" in IEEE 802), location (including subscriptions) of a device, QoS bearers, deterministic delivery for a flow, etc.
- 5: enables the exposing/requesting services to/from other IoT Entities. Examples of the usage of this interface are to allow a gateway to upload data to a cloud server, retrieve software image of a gateway or a device, etc.

The AIOTI HLA enables the digital representation of physical things in the IoT Entities. Such representations typically support discovery of things by App Entities and enable related services such as actuation or measurements. To achieve semantic interoperability, the representation of things typically contains data, such as measurements, as well as metadata. The metadata provide semantic descriptions of the things in line with the domain model and may be enhanced/extended with knowledge from specific vertical domains. The representation of the things in the IoT Entities is typically provided by App Entities or IoT Entities residing in devices, gateways or servers.

A one to one mapping between a physical thing and its representation shall not be assumed as there could be multiple representations depending on the user needs.

Figure 5-3 provides the relationships between the physical things, their representations and the link to semantic metadata which are an instantiation of the domain model described earlier in this document. Further information about AIOTI Semantic Interoperability is available from [6].

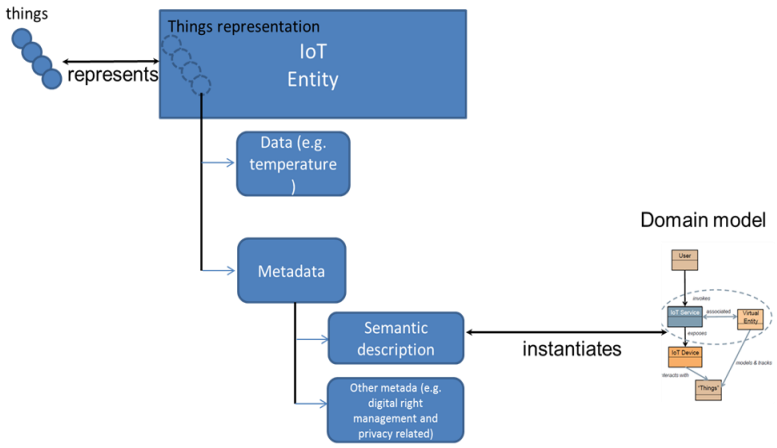


Figure 5-3: Relationship between a thing, a thing representation and the domain model

5.3 HLA Security and Management considerations

Security and Management are fully recognized as important features in the AIOTI HLA. AIOTI HLA argues that security and management should be intrinsic to interface specifications.

All the depicted interfaces shall support authentication (including mutual authentication), authorization and encryption at hop by hop level. End to end application level security could also be achieved via securing interface 1. It is fully recognized that there could be additional and diverse security needs for the different LSPs.

As far as security and management are concerned, there are several aspects of interest, including without limitation the aspects set forth below:

- Device and gateway management are broadly defined as software/firmware upgrade as well as configuration/fault and performance management. Device management can be performed using interface 5 via known protocols e.g. BBF TR-069 and OMA LWM2M. Additionally Device and gateway management could also be exposed as features to cloud applications using interface 2.
- Infrastructure management in terms of configuration, fault and performance is not handled in this version of the HLA but is fully recognized as important aspect for future study.
- Data life cycle management, which is relevant in each of the three main layers set forth in paragraph 5.1 if, where and to the extent any data enters, travels through, is derived or is otherwise processed in such layer or between several layers. Data management takes the data-centric approach in order to focus on the specific data and its data classification(s), the phase(s) of the data life cycle will be in when processed in such layer(s), and the respective processing purposes. The data life cycle can be split in seven main phases as set forth below, where each phase will need to be taken into account, on the basis of if, where and to what extent applicability:
    - Obtain/collect
    - Create/derive
    - Use

- Store
- Share/disclose
- Archive
- Destroy/Delete

- Digital rights management, includes identity, access, rights of use and other control and rights management of the application, IoT and network layers, as well as the data therein, including without limitation derived data (metadata) control and use thereof.
- Compliance management, when such data life cycle and digital rights management are landscaped, the respective actors identified and the authentication, authorization and encryption at hop by hop level in the application, IoT and network layers and the data therein are architected as well, these security and management domains combined would need to be addressed and (re)considered from a compliance point of view, including without limitation safety, security, data minimisation and data retention obligations, security breach notification and disclosure obligations, (personal) data protection compliance, official mandatory policies compliance and the like, also here: if, where and to the extent applicable.

NOTE - AIOTI WG03 is in close cooperation with AIOTI WG that is addressing the policy issues for security and privacy.

6. Identifiers for IoT

In any system of interacting components, identification of these components is needed in order to ensure the correct composition and operation of the system. This applies to all lifecycle phases of a system from development to assembly, commissioning, operations, maintenance and even end of life. Especially in case of flexible and dynamic interactions between system components identification plays an important role.

Identifiers are used to provide identification. In general, an identifier is a pattern to uniquely identify a single entity (instance identifier) or a class of entities (i.e. type identifier) within a specific context.

IoT is about interaction between things and users by electronic means. Both things and user have to be identified in order to establish such interaction. Various other entities are involved in the interaction like sensor and actuation devices, virtual representations of the thing (virtual entities), service entities and communication relationships are part of an IoT system and identification is also relevant for them. Figure 6-1 shows the different entities with the related identifiers in the IoT Domain Model.

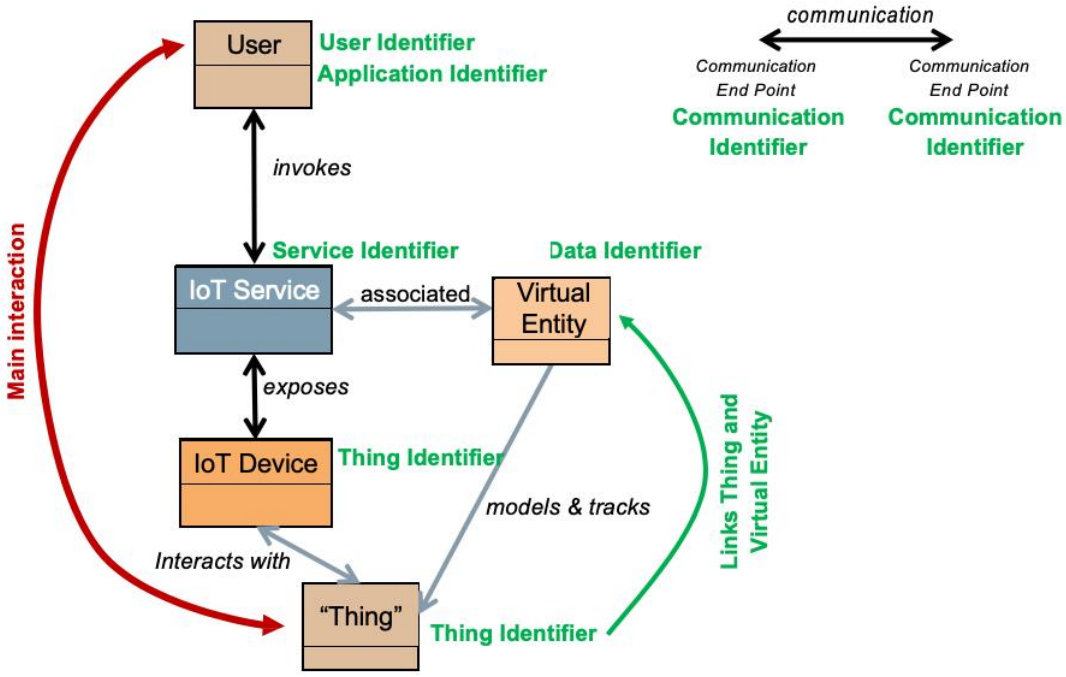


Figure 6-1: Identifiers examples in the IoT Domain Model

In general, the following categories of identifiers have to be considered for IoT systems:

- Thing Identifier

  Thing identifiers identify the entity of interest of the IoT application. This can be for example any physical object (e.g. machines, properties, humans, animals, plants) or digital data (e.g. files, data sets, metadata); basically, anything that one can interact with. Identification can be based on inherent patterns of the thing itself like face recognition, fingerprints or iris scans. In most cases a specific pattern will be added to the thing for identification by technical means like printed or engraved serial numbers, bar codes, RFIDs or numbers stored in the memory of devices.

- Application & Service Identifier

  Application and Service identifiers identify software applications and services. This also includes identifiers for methods on how to interact with the application or service (i.e. Application Programming Interfaces, Remote Procedure Calls)

- Communication Identifier

  Communication identifiers identify communication (end) points (e.g. source, destination) and sessions. Communication identifiers are usually bound to the specific communication technology and defined as part of the standardization of the technology.

- User Identifier

  User identifiers identify users of IoT applications and services. Users can be humans, parties (e.g. legal entities) or software applications that access and interact with the IoT application or service.

- Data Identifier

  This class covers both identification of specific data instances and data types (e.g. meta data, properties, classes).

- Location Identifier

  This class is about Identification of locations within a geographic area (e.g. geospatial coordinates, postal addresses, room numbers).

- Protocol Identifier

  Protocol identifiers inform for example communication protocols about the upper layer protocol they are transporting or applications about the protocol they have to use in order to establish a specific communication exchange.

As listed, identifiers are used to identify various types of entities for many purposes and within different context. This leads to a wide variety of, sometimes even contradicting, requirements. Special operating constraints for many IoT applications (e.g. constrained devices and networks, entities without processing capabilities) further contribute to that. In general, no single identification scheme fits all needs. Furthermore, various identifiers schemes are already in use and standardized for years. They are often application or domain specific, but also generic identifier schemes that cover a wide application area exist. These existing schemes will be used in IoT, and new schemes might be added. IoT applications have to deal with the variety of identification schemes and as long as they are used in their defined context this should not be a problem. Mapping and resolution between different schemes is already a standard feature of today's solutions. Still, system architects should have in mind that IoT systems might be used in a wider context and have to interact with other IoT systems in the future. For identifiers that will be impacted by that, an identification scheme that can already handle such situations or can be easily extended should be considered.

Security and privacy are important for identifiers. The specific requirements strongly depend on the use case and identified entity. As part of a security and privacy threat and risk analysis, also the specific requirements related to the identifiers have to be identified and relevant legal and regulatory frameworks have to be taken into account in order to ensure state of the art security and privacy.

A detailed analysis of Identifiers in IoT [20] has been done by the IoT Identifier task force of AIOTI WG3. [20]

- evaluates IoT identification needs;
- classifies the different identification schemes;
- evaluates and categorizes related requirements;
- provides examples of identifier standards and elaborates their applicability for IoT;
- discusses allocation, registration resolution of identifiers;
- considers security and privacy issues;
- and discusses interoperability of identifiers.

7. Deployment considerations for HLA

7.1 Introduction

This clause highlights deployment considerations for AIOTI HLA. The deployment of AIOTI HLA may relay on the following technologies and concepts:

- Cloud and Edge Computing: AIOTI HLA is typically deployed using cloud infrastructures. Cloud native principles can be applied to ensure scaling and resilience for IoT. In certain use cases, deploying edge cloud infrastructures[1], will be beneficial to allow data processing locally. AIOTI HLA has been designed to allow for distributed intelligence, it is therefore compatible with Cloud and Edge computing.
- Big data: collecting, storing and sharing data is an integral part of IoT, therefore also for AIOTI HLA. Big data can be seen as the set of disciplines, such as storing, analysing, querying and visualization of large data sets. Those disciplines are equally applicable to IoT data sets.
- Virtualization: ensuring flexibility and scale is one of the major challenges for deploying IoT.

Virtualization would help scaling IoT for a large number of use-cases.

7.2 Cloud and Edge computing

AIOTI HLA is designed to be a largely distributed system because it fully recognizes that every entity (including devices and gateways in the field domain) can run applications, without being specific about the application logic. Cloud computing is an important enabler for deploying IoT with distributed intelligence. It provides the computing infrastructure needed for large and distributed deployments of IoT. In this clause we focus on an overview of cloud native principles as well as recent edge computing initiatives, namely ETSI ISG MEC [12] and OpenFog. More emphasis has been put on edge computing, see [14], aspects because it has been identified as important for several emerging use cases such as in the industrial IoT space. Annex III introduces a comparison table for device, edge and cloud computing forms.

[1] Edge cloud is a cloud infrastructure that is located closely to the devices.

7.2.1 Cloud principles

There are several agreed principles for cloud native offerings, these include:

- Horizontal scalability: adding cloud resources at run time without any disruption to ongoing operations in terms of communication, processing, storage, and monitoring.
- No single point of failure: providing fault tolerance through node replication techniques or disaster recovery site.
- High data throughputs: needed for massive amounts of connections or massive data sets (e.g. generated by video streams or data logs).
- Fine-grained micro-services architectures, lightweight containers deployment and service orchestration.
- DevOps with holistic service monitoring and decentralized continuous delivery.

7.2.2 Edge cloud initiatives

7.2.2.1 ETSI Mobile Edge Computing

Mobile Edge Computing (MEC) [12] is a technology which is currently being standardized in an ETSI Industry Specification Group (ISG) of the same name (recently renamed Multi-access Edge Computing). MEC provides an IT service environment and cloud-computing capabilities at the edge of the network (e.g. within the Radio Access Network (RAN) and in close proximity to subscribers). The aim is to reduce latency, ensure highly efficient network operation and service delivery, and offer an improved user experience.

MEC represents an architectural concept and APIs to enable the evolution to 5G, since it helps advance the transformation of the mobile broadband network into a programmable world and contributes to satisfying the demanding requirements of 5G (but not only) in terms of expected throughout, latency, scalability and automation.

The market drivers of MEC include business transformation, technology integration and industry collaboration. All of these can be enabled by MEC and a wide variety of use cases can be supported for new and innovative markets, such as e-Health, connected vehicles, industry automation, augmented reality, gaming and IoT services.

Figure 7-1 shows the framework for Mobile Edge Computing consisting of the following entities:

- Mobile Edge Host, including the following:
    - mobile edge platform;
    - mobile edge applications;
    - virtualization infrastructure;
- Mobile Edge System Level management;
- Mobile Edge Host level management;

- External related entities, i.e. network level entities.

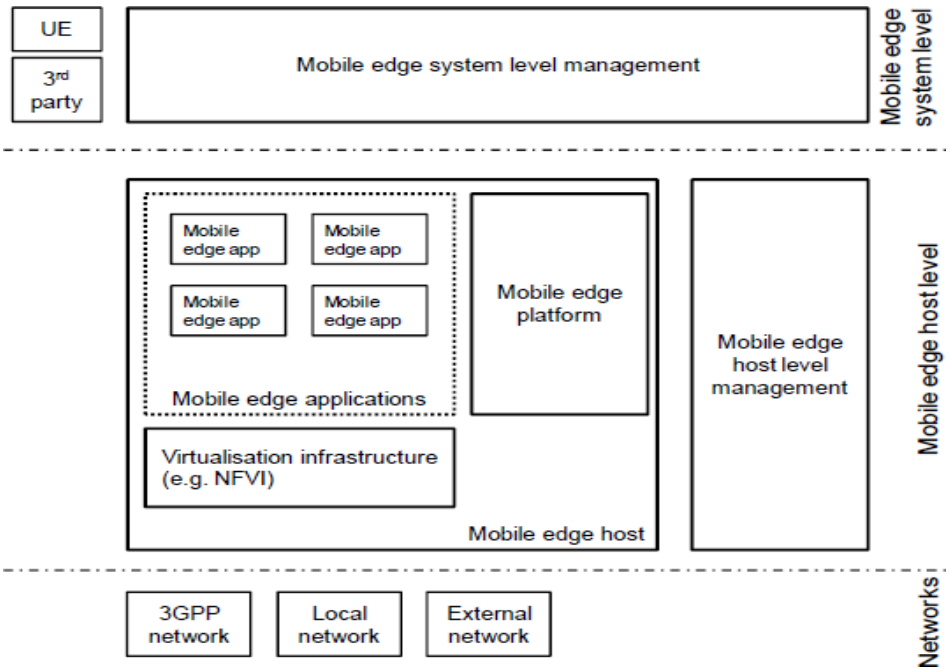


Figure 7-1: Mobile Edge Computing Framework [ETSI GS MEC 003]

MEC can be used as computing infrastructure for AIOTI HLA in particular where IoT Entities and App Entities of HLA reside at the edge of the network, i.e. close to IoT devices. For instance, Mobile edge app in Figure 7-1 could be mapped to App Entity in HLA.

7.2.2.2 OpenFog

The OpenFog Architecture is a system-level architecture that extends elements of computing, networking and storage across the cloud through to the edge of the network. OpenFog consortium sees this approach as a mean to accelerate the decision-making velocity. The architecture is argued to serve use cases that cannot be served with centralised “cloud only” approach. The OpenFog Consortium, formed in November 2015, is based on the premise that an open architecture is essential for the success of a ubiquitous fog computing ecosystem for IoT platforms and applications. More information about OpenFog can be found using this reference [15].

The goal of the OpenFog architecture is to facilitate deployments which highlight interoperability, performance, security, scalability, programmability, reliability, availability, serviceability, and agility. The following figure provides a possible scenario for deploying OpenFog. One can notice this approach allows for both edge to cloud and edge to edge communications, referred to in the OpenFog model as East/West.

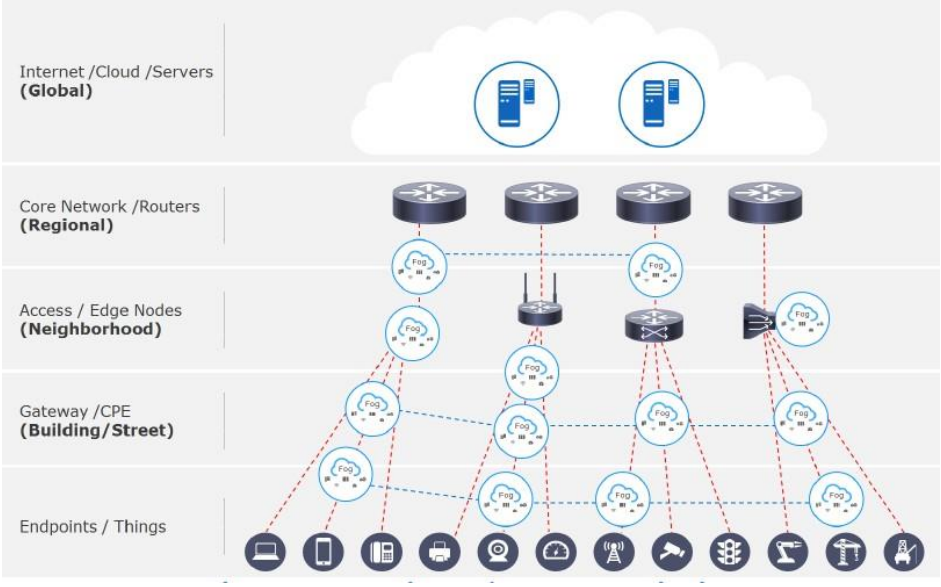


Figure 7-2: OpenFog cloud hierarchy

OpenFog cloud infrastructure elements can host both App Entities and IoT Entities in the context of AIOTI HLA context.

7.3 Big Data

7.3.1 Definitions

The following big data definitions are important to understand what big data is about and what the relationships to IoT are.

- Big Data (ITU-T Y.3600 [7]): A paradigm for enabling the collection, storage, management, analysis and visualization, potentially under real-time constraints, of extensive datasets with heterogeneous characteristics. Examples of datasets characteristics include high-volume, high-velocity, high-variety, etc.
- IoT Big Data characteristics (ITU-T Y.4114 [8]): IoT data set characteristics of high-volume, high-velocity and/or high-variety related to the challenges of IoT data set operations, in some cases without human intervention. Additional dimensions of data, such as veracity, variability etc., may also be associated with the IoT Big Data characteristics. Operations on IoT data sets include collection, pre-processing, transfer, storage, query, analysis and visualization.

NOTE - It is also recognized that IoT data sets can be characterised as small data in certain scenarios.

In the context of Big Data, we can distinguish 3 data types:

- Structured data are often stored in databases which may be organized in different models, such as relational models, document models, key-value models, graph models, etc.
- Semi-structured data do not conform to the formal structure of data models, but they
- contain tags or markers to identify data.

- Unstructured data do not have a pre-defined data model and are not organized in any defined manner.

Within all data types, data can exist in formats such as text, spreadsheet, video, audio, image, map, etc. According to ITU-T Y.3600 [7], we can distinguish the following data dimensions:

- Volume: refers to the amount of data collected, stored, analysed and visualized, which Big Data technologies need to resolve.
- Variety: refers to different data types and data formats that are processed by Big
- Data technologies.
- Velocity: refers to both how fast the data is being collected and how fast the data is processed by Big Data technologies to deliver expected results.
- Veracity: refers to the certainty level of the data.
- Value: refers to the business results from the gains in new information using Big Data technologies.

7.3.2 IoT data roles

Based on the consideration of IoT system and IoT Big Data characteristics, five key IoT data roles, i.e. the key roles which are relevant in an IoT deployment from a data operation perspective, are identified for the IoT ecosystemas shown in Figure 7-3.

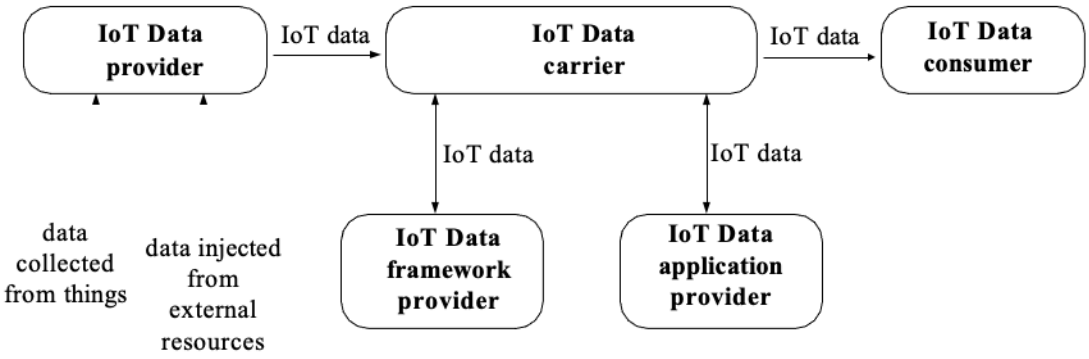


Figure 7-3: IoT data roles [8]

- IoT Data provider: collects data from things, injects data processed within the IoT system as well as data from external sources, and provides them via the IoT Data carrier to the IoT Data consumer (optionally, the applications provided by the IoT Data application provider may execute relevant data operations with the support of the IoT Data framework provider).
- IoT Data application provider: provides applications related to the execution of IoT data operations (e.g. applications for data analysis, data pre-processing, data visualization and data query).
  - The applications provided by the IoT Data application provider can interact with the infrastructure provided by the IoT Data framework provider (e.g. storage cloud) through the IoT Data carrier or run on the infrastructure itself provided by the IoT Data framework provider (e.g. scalable distributed computing platform).
- IoT Data framework provider: provides general IoT data processing capabilities and related infrastructure (e.g. storage and computing resources, data processing run time environment) as required by IoT Data provider, IoT Data carrier, IoT Data application provider and IoT Data consumer for the support of the execution of data operations.
- IoT Data consumer: consumes IoT data. Usage of the consumed data depends on the application purposes.
- IoT Data carrier: carries data among IoT Data provider, IoT Data framework provider, IoT Data application provider and IoT Data consumer.

An actor of a concrete IoT deployment can play multiple roles. As an example, an actor executing data analysis plays the role of IoT Data application provider, but also plays the role of IoT Data provider when it sends the results of this data analysis to other actors.

The following table provides a mapping between ITU Y.4114 [8] and AIOTI HLA:

IoT data roles according to ITU Y.4114	HLA Entity(ies)
IoT Data Provider	App Entity, IoT entity
IoT Data application provider	App Entity Note: typically, the IoT Data application provider manages the lifecycle of IoT applications, i.e. App Entity in HLA
IoT Data framework provider	IoT Entity
IoT Data consumer	App Entity
IoT Data carrier	Networks

Table 7-1: Mapping of ITU Y.4114 to AIOTI HLA

7.3.3 IoT data operations

Considering that the diverse set of concrete IoT deployments does not imply a unique logical sequencing of the various IoT data operations, Figure 7-4 provides an abstract representation of the various IoT data operations and related data flows [8].

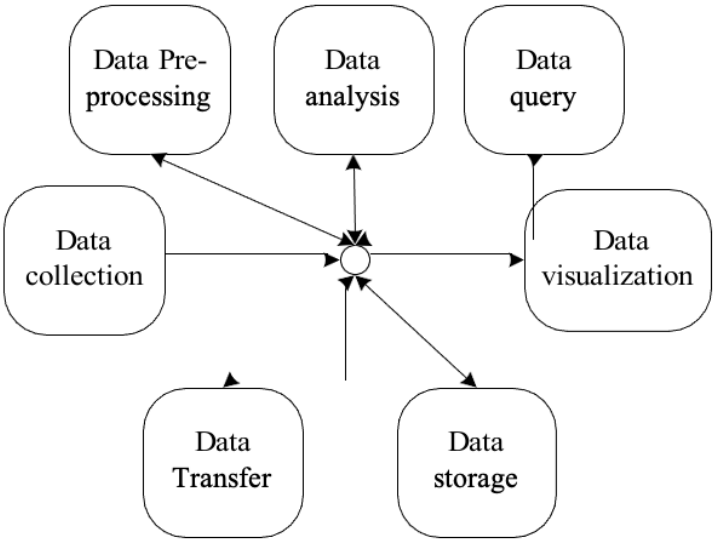


Figure 7-4: IoT data operations [8]

The sequencing of IoT data operations highly depends on the service and deployment scenarios. Cloud computing and edge computing are two technologies that may be implemented in the IoT for support of different IoT data operation sequences: e.g. cloud computing can be used to perform data analysis in differed time, i.e. after data are transferred to or acquired by the remote IoT platform, while edge computing can be used to perform near real time data analysis and actuators control locally such as at gateway level.

7.3.4 AI enabled by Big Data

NOTE- Topic for study in following Release(s) of this document.

7.3.5 Big Data related initiatives

GSMA proposes an architectural framework for the delivery of Big Data services based on the Internet of Things [27]. This framework identifies the key functions and interfaces that enable IoT Big Data services to be delivered, and it makes selections and recommendations particularly in the area of interfaces that support the creation of an IoT Big Data ecosystem.

According to GSMA, the key challenges for Big Data in the context of IoT are:

- Devices: scalability (number of IoT devices), variety of IoT devices, intelligence of IoT devices, risk of IoT device malfunction.
- Data management: update frequency, historical data.
- Context data: much IoT data will make more sense when put in context with other data.
- Privacy issues.

TMForum proposes a set of data analytics tools to be used for Big Data [28]. Data Analytics concerns the identification, design and deployment of strategies, processes, skills, systems and data that can provide actionable intelligence resulting in business value. It is about the harnessing of the different varieties, volume, and velocity of data. To execute on this, and to deliver improvements in areas such as customer

experience or reduction in customer churn, there are a number of operational issues including data integration.

BDVA [30], the private counterpart to the EU Commission to implement the Big Data Value Public-Private-Partnership (BDV PPP), aims to “to develop the Innovation Ecosystem that will enable the data-driven digital transformation in Europe, delivering economic and societal benefits, and, achieving and sustaining Europe’s leadership on Data-Driven Value Creation and Artificial Intelligence”.

BDVA has defined 4 strategic priorities to guide the Association activities and outcomes: to provide Data Innovation Recommendations; to develop the Innovation Ecosystem to enable the data-driven digital transformation in Europe; to guide standards and to provide input for the respective “Standards development organisations”; and, to improve the adoption of technologies through “Know-How and Skills” and best practices exchange Data.

BDVA maintains and fulfils a Strategic Research and Innovation Agenda (SRIA) for Big Data Value domain, contributes to the Horizon 2020 Work Programmes and calls for proposals and it monitors the progress of the BDV PPP. BDVA manages over 25 working groups organised in Task Forces and subgroups, tackling with all the technical and non-technical challenges of the Big Data Value.

ISO JTC1 WG09 has been the home for the Big Data Standardisation activities in ISO, with a foundational input from the NIST Big Data Framework [2]. The WG09’s Big Data activities have been transferred in May 2018 into the new ISO JTC1 SC42 “Artificial Intelligence” [32], whose scope is the standardization in the area of Artificial Intelligence, serving as the focus and proponent for JTC 1's standardization program on Artificial Intelligence and providing guidance to JTC 1, IEC, and ISO committees developing Artificial Intelligence applications.

Two Technical reports have been developed related to Big data reference architecture:

- ISO/IEC TR 20547-2:2018 Information technology -- Big data reference architecture -- Part 2: Use cases and derived requirements
- ISO/IEC TR 20547-5:2018 Information technology -- Big data reference architecture -- Part 5: Standards roadmap

Other work in progress includes specifications related to Big data reference architecture:

- ISO/IEC AWI TR 20547-1 [Under development] Information technology -- Big data reference architecture -- Part 1: Framework and application process
- ISO/IEC DIS 20547-3 [Under development] Information technology -- Big data reference architecture -- Part 3: Reference architecture and Artificial Intelligence
- ISO/IEC AWI 22989 [Under development] Artificial Intelligence Concepts and Terminology
- ISO/IEC AWI 23053 [Under development] Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)

Relevant working groups include:

- ISO/IEC JTC 1/SC 42/SG 1 - Computational approaches and characteristics of artificial intelligence systems Working group
- ISO/IEC JTC 1/SC 42/SG 2 - Trustworthiness Working group
- ISO/IEC JTC 1/SC 42/SG 3 - Use cases and applications Working group
- ISO/IEC JTC 1/SC 42/WG 1 - Foundational standards

In the context of the ITU-T standardization activities related to IoT, Study Group 20 ("Internet of things (IoT) and smart cities and communities (SC&C)"), central ITU-T expert group for IoT, has supervised the research and pre-standardization activities of the ITU-T FG-DPM, Focus Group on Data Processing and Management to support IoT and Smart Cities & Communities [29], which completed its work in July 2019.

The ITU-T FG-DPM's Terms of Reference included, among others, the study and survey of technologies, platforms and standards for data processing and management, the promotion of data management frameworks, including related security and trust aspects, the investigation of emerging technologies and trends to support data management including blockchain, and the identification of standards challenges.

The deliverables produced by the FG-DPM have concerned different areas of relevance: Use Cases, Requirements and Applications; Framework, Architectures and Core Components; Data sharing, Interoperability and Blockchain; Security, Privacy and Trust including Governance; Data Economy, commercialization and monetization [29]. The ITU-T Study Group 20 took in charge the FG-DPM deliverables at the FG closure, and is progressing related specifications (as Recommendations or Supplements).

7.4 Security aspects

NOTE - Enhancements specific to HLA matters may be developed in a following Release of this document.

As an essential enabling property for Trust, security is a key feature of all IoT systems and needs to be dealt with in a global manner. One key challenge is that it is involving a variety of users in a variety of use cases. One of the many characteristics of IoT is that the number of communicating entities is very large and the number of possible relationships per device is larger than, say, with cellular telecommunication. The purpose of security technologies is multi-fold:

- Confidentiality: Information shared by Party A with Party B is only visible to these two parties. If Party C can access the information, it cannot ascertain the meaning of the content. Confidentiality is primarily achieved using cryptographic.
- Integrity: Information shared by Party A with Party B can be proven by Party A not to have been manipulated by a 3rd party (e.g., Party C). Party B can verify this is the case. Proof and verification of document integrity is primarily achieved using cryptographic hash functions which have specific characteristics.
- Availability: This addresses the aim of ensuring that an authorized party (e.g., Party A) is able to access services or information when needed. In other words, that Party A has access only to those assets it is allowed to access and that they are available to Party A when legitimately demanded, and that an adversary, Party C, does not have access. The technologies that address this include Identity Management, Authentication and Access Control, in addition considerations in the availability domain include reliability and resilience which, whilst not strictly addressed by security technology, impact on availability.

Whilst the population of cellular telecommunications devices is very large the nature of the connection is pre-defined by the SIM containing the subscriber mobile identity and its association to a single trusted provider (holder of the symmetric key used in the network/device authentication process). An IoT device, unless a specific example of a cellular enabled IoT device containing a SIM, does not have a predefined security association to a trusted entity.

As a trivial example IoT communications security may be considered as equivalent to sending presents to somebody. To ensure the recipient does not know the content before unwrapping, the sender masks the content by wrapping the gift – this makes the content confidential. The intended recipient is clearly indicated on the label as is the sender – this identifies the parties to the transaction and depending on how names are written may confer some proof of identity. Finally, in order to ensure the package is not damaged, the sender adds packaging that protects it – this is some means of ensuring the integrity of the package is maintained in transit. Translating this to IoT, data from A to B can be encrypted to confer confidentiality. The parties A and B have to be able to prove their identity to confer authenticity to the exchange, and the parties can add data to the package that will be used to assure and verify the integrity of the package.

The general purpose of security technology is to give confidence to the stakeholders that the risk of cyber-attacks, or any other attack on the assets of a system, are mitigated. Hence one of the purposes of security design is to minimize the probability of any loss of confidentiality, integrity and/or availability ("unwanted incident"). Achieving security in IoT systems is a challenge of high complexity since there are many unknowns that have to be resolved prior to overall security being achieved. As an example, the form and function of an

IoT device, its identity, its set of security credentials, the algorithms it deploys to assure each of confidentiality, identity and integrity, the means by which it interacts with peers and other systems, all of these have to be known.

In the period to approximately mid-2016, the EU regulatory landscape related to cyber security was relatively fragmented with legal obligations and principles scattered across numerous legal acts. Due to recent technological advancements and increased connectivity, the risk of becoming a victim of a cybercrime has also increased. Thus, EU law-makers been taking steps to increase cyber resilience across Member States by making the respective regulatory landscape more concise, among others. In this respect, they have adopted Directive (EU) 2016/1148 of the European Parliament and of the Council of 6 July 2016 concerning measures for high common level of security of network and information systems across the Union (commonly referred to as the "NIS Directive"), being the first EU horizontal legislation addressing cybersecurity challenges.

Overall, it is strongly recommended that any application of security technology adopts the risk analysis approach and the cataloguing of the system identified in relevant standards (e.g., ETSI TS 102 165-1 [43], ETSI TR 103 305-x [44]) since security mechanisms, processes, procedures, are all reliant for their success, on understanding of risk. Taking care of security at the early stage of designing/adapting/deploying an IoT system is very essential and an important topic for further work on HLA within AIOTI.

7.5 Privacy aspects

The General Data Protection Regulation (GDPR) [34] that became applicable as of the 25 May 2018 introduces - among other - two new elements concerning privacy that are of high relevance for the scope and the objectives aspired by the present document: the principle of accountability and the obligation of privacy by design.

More specifically, the GDPR introduces the principle of accountability as a form of ''umbrella principle''. Under the new law, public and private organizations of all sizes processing personal information must not only do what they have been expected so far to do concerning processing of personal information (e.g. retain personal information as short as possible, as long as necessary), but also be able to demonstrate that they did so. Organizations are, therefore, expected to maintain evidence throughout the processing of personal information, irrespective of whether they will be actually requested to provide them to enforcement authorities or other auditing bodies. GDPR requires organizations to be able to show evidence that they ''did the right thing'', but to this end it leaves them free to decide upon the technical means they employ.

Moreover, the GDPR also introduces the principle of data protection by design, meaning that privacy protection should be taken into account in the design of business operations, processes and services. Basically, the GDPR does formally introduce Privacy by Design, as the basic principle on which the rest of the principles already identified by AIOTI can be built upon, namely:

- No personal data by default principle, that implies refraining from any collection or creation
- of personal data by default, except for cases where such collection or creation is legally required and to the exact extent required.
- As-If' X-by-Design, that refers to the requirement that ecosystems are designed and
- engineered as-if these will process personal data at an immediate and/or later stage.

- De-Identification by Default, that refers to the de-identification, sanitization or deletion of personal data as soon as the legal basis for keeping such data ceases.
- Data Minimization by Default, that stipulates that personal data shall only be processed
- where, when and to the extent required; otherwise, this data shall be deleted or de-identified.
- Encryption by Default, that refers to the requirement to encrypt personal data by default, while capturing both digital rights and digital rights management.

Note that these principles are extensively addressed in ongoing AIOTI studies.

Overall, both the principle of accountability and privacy by design are highly relevant for IoT architectures, as they should affect basic choices at an early stage. Those two principles on HLA, briefly discussed above, pave the ground for future work focused on privacy within AIOTI, potentially, to be concretely applied to HLA.

7.6 Virtualization

7.6.1 Combining IoT and Cloud Computing

The new IoT systems that emerge at industrial scale will typically require very high numbers of connected devices (and therefore strong requirements for scalability or deployment automation) as well as stringent non-functional requirements (such as low latency). Those IoT systems will also require a high degree of availability, adaptability and flexibility: in particular, the resources they use may have to be available in a very dynamic manner, both in terms of configuration and run- time flexibility. The models provided by Cloud Computing have been designed to serve such requirements in mind, and they seem very attractive in the context of the design, development and deployment of IoT systems.

Cloud computing is allowing the provision of very sophisticated capabilities – for computing, storage, analytics, etc. – to very dynamic and potentially massive number of users. It provides functional and non-functional support (e.g., low latency fault-tolerance, horizontal scalability, cost-optimization, or geo-optimization together with Service Level Agreements (SLAs), and security.

Virtualizing IoT builds on two key pillars which are strongly related. First, cloud native principles (as described 7.2.1) need to be applied to the distributed IoT platforms. Those principles include: micro services, no single point of failure, high throughput, horizontal and vertical scalability, DevOps, etc. All those principles must apply independently from underlying private or public cloud technology. Second, the network must evolve to provide the level of flexibility, QoS and isolation needed for massive consumer, enterprise or industrial IoT deployments. This means the capability of offering and flexibly managing, eventually through APIs, network slices and chaining functions end-to-end. The role of an all IP network, preferably based on IPv6, will be crucial in ensuring security and QoS.

The benefits of virtualization are largely documented, see e.g., [23]. In the context of IoT the key benefits of virtualization are:

- Rapid service innovation through software-based deployment and operationalization of IoT services.
- Improved operational efficiencies resulting from common automation and operating procedures.
- Reduced power usage by migrating workloads and powering down unused hardware.
- Greater flexibility on assigning IoT virtualized functions and objects to hardware.
- Improved capital efficiencies compared to dedicated hardware implementations.

The following aspects are crucial for the widespread use of IoT in daily life using virtualization [33]:

- Reuse of IoT devices for different verticals,
- Composition of multitude of IoT devices to offer new services through abstraction,
- Representation of physical world objects using IoT, and
- Bringing cognitive functionality in IoT for better service orchestration.

An important aspect is the deployment model where several possibilities are offered by the Cloud Service Providers: Platform-as-a-Service, Infrastructure-as-a-Service, Software-as-a- Service, etc. The Figure 7-5 presents the possible usages of such offerings in delegating more and more important parts of the underlying layers to a third-party in charge of hiding complexity, resource usage, etc.

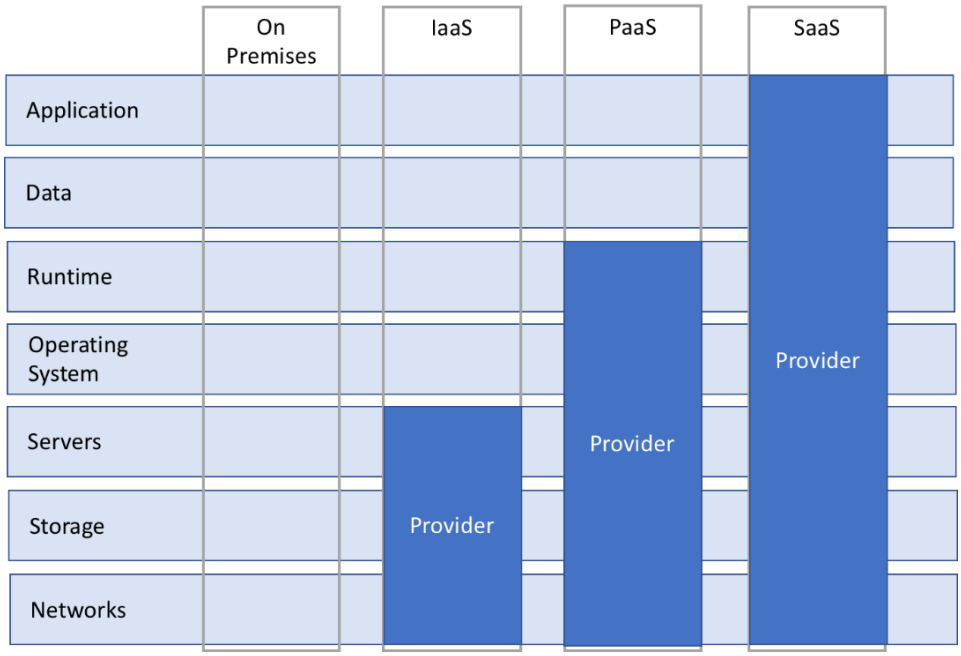


Figure 7-5: The potential of Cloud Computing Service Models

The main challenge of IoT Virtualization is to design and develop systems that can benefit from the flexibility of the "XaaS" offerings (IaaS, PaaS, SaaS), of the vast amount of available (Open Source) software components together with the possibility to rely on the support of standards (such as oneM2M).

7.6.2 Approaches to IoT Virtualization

Three approaches are outlined below. The first one (see clause 7.6.2.1) is regarding the application of Cloud Computing techniques and solutions to IoT systems: it comes with a practice of the Cloud Computing community where the role of (in particular Open Source communities) prevails on an approach based on standards. The second one (i.e. NFV) (see clause 7.6.2.2) is using a "standards-based" approach and seeks the adaptation of the virtualization technologies coming from Cloud Computing. A third approach (see clause 7.6.2.3) concerns device virtualization: using Virtual Objects (VO's) and Virtual Composite Objects (VCO's), it aims to make it easier to use and reuse IoT devices in a multitude of applications. This can be regarded as a layer between devices and the virtualization layer.

7.6.2.1 Microservices-based Architectures for Virtualization

The Cloud Computing community has developed new approaches for the engineering of Cloud- based systems that can be used for IoT Virtualization. Two important aspects are the following:

- Microservices. Microservices are an architectural approach to developing applications as a set of small services, where each service is running as a separate process, communicating through simple mechanisms. IoT system architectures based on microservices must be able to support the split of monolithic services into a number of microservices that are able to evolve relatively independently from each other and to communicate in a safe, secure and efficient manner.
- Architectures. The possibility to split an IoT system into microservices that can be implemented by various (possibly Open Source Software) components goes with the risk of a lack of structure of the resulting implementation: the definition of architectural layers in a functional architecture supporting the most effective selection and combination of such components is a key element.

A microservices-based architecture relies on the use of: 1/ microservices as a (software engineering) means to structure the systems and 2/ inter-process communications models synchronous (e.g., RESTful) or asynchronous (e.g. message broker). Each service subscribes to the events that it is interested in consuming, and then receives these events reliably when the events are placed on the queue by other services. Figure 7-6 provides an example of such system.

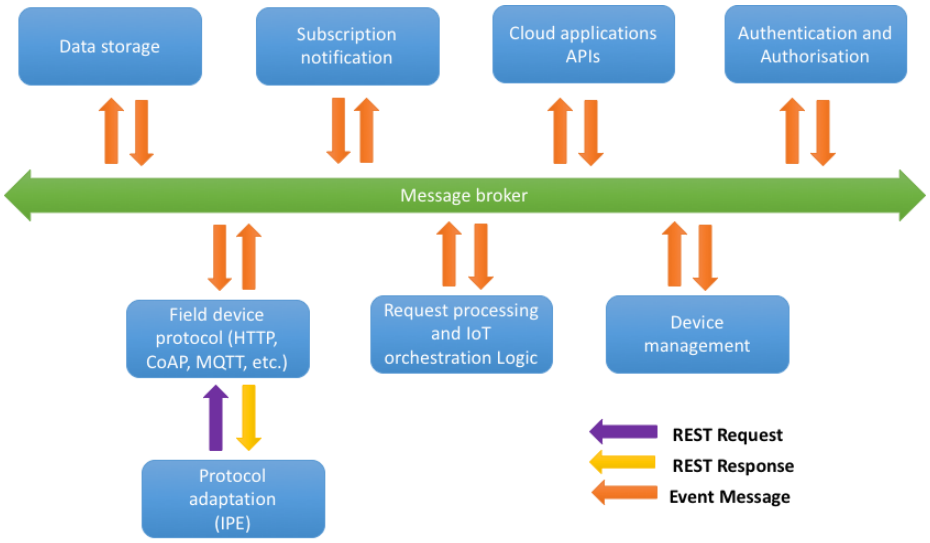
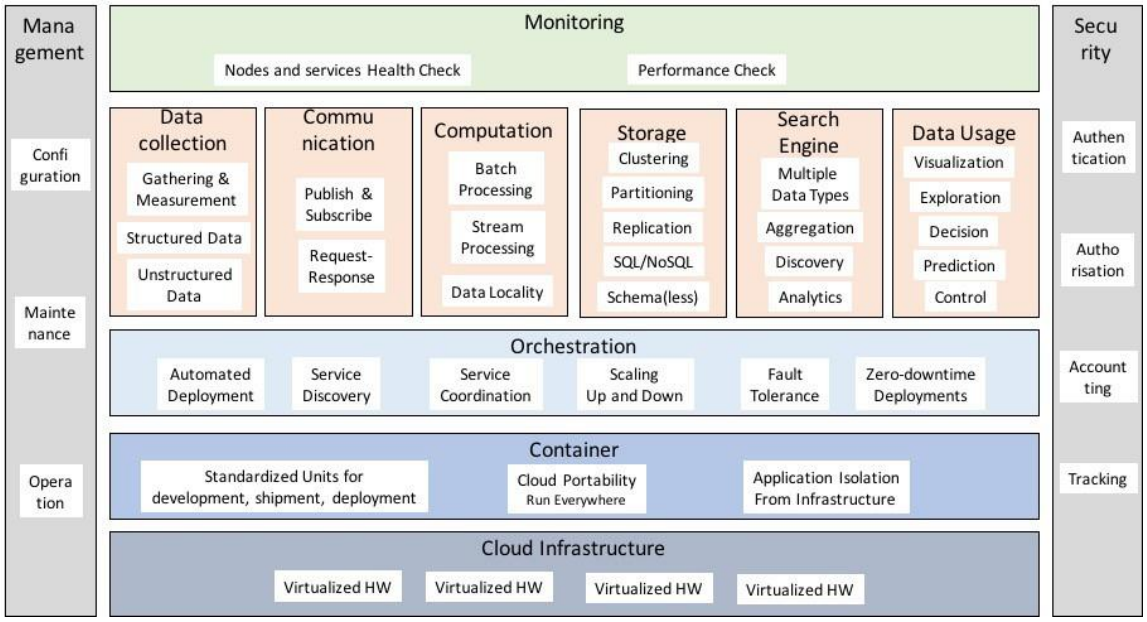


Figure 7-6: Microservices conceptual framework for IoT Virtualization

The possibility to define architectural layers and group them in a functional architecture for IoT virtualization may allow for the most effective selection and combination of microservices-based components.

Figure 7-7 introduces an example of a structuration of the functional architecture into layers (and sublayers) with an indication of the main functions that are expected to be provided in each of the layers and sublayers. In addition, two vertical functions are added related to cross- layer functionality: security and management.

Figure 7-7: A microservices-based functional architecture for IoT Virtualization



It must be noted that this architecture is one example (amongst other possible ones) which is in particular dealing with a structuration of the generic microservices that could be found in an IoT Layer. More on this approach can be found in the ETSI Technical Reports 103 527 [21] and 103 528 [22].

7.6.2.2 Virtualization in the NFV Architecture

The NFV ISG has initially worked on the identification of use cases for virtualization and their implication on the virtualization of traditional network functions. Based on this, the ISG has defined the NFV Architectural Framework, its main components and reference points [24].

More specifically, the ISG has defined the "NFV Infrastructure" (NFVI): "The NFVI is the totality of the hardware and software components which build up the environment in which VNFs are deployed. The NFVI is deployed as a distributed set of NFVI-nodes in various locations to support the locality and latency requirements of the different use cases and the NFVI provide the physical platform on which the diverse set of VNFs are executed; enabling the flexible deployment of network functions envisaged by the NFV Architectural Framework" [25].

The high level NFV framework (see [24]) can be seen in Figure 7-8 and consists of three main domains:

- Virtualized Network Function (VNF): the software implementation of a network function which is capable of running over the NFVI.
- NFV Infrastructure (NFVI): includes the diversity of physical resources and how they can be virtualized. The NFVI supports the execution of the VNFs.
- NFV management and orchestration (MANO): covers the orchestration and lifecycle management of physical and/or software resources that support the infrastructure virtualization and the lifecycle management of VNFs. NFV Management and Orchestration focuses on all virtualization-specific management tasks necessary in the NFV framework.

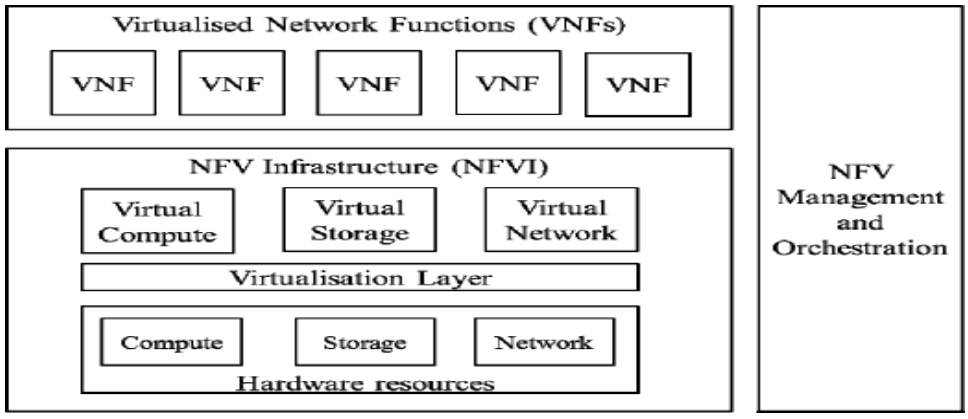


Figure 7-8: High Level NFV Framework

Regarding IoT Virtualization, the question is whether or not NFV can be used as an IoT Virtualization Framework. The answer is that, as long as the IoT functions that are targeted for virtualization are matching the ones defined in the NFV Architectural Framework, the latter can be used as an IoT virtualization framework where a VNF is replaced by an "IoT Virtualized Function". The main advantage of this approach is that the Reference Points defined by the NFV Architectural Framework can be used by the virtualized IoT system.

7.6.2.3 Network Slicing and Virtualization

Several initiatives, such as 3GPP, BBF, ETSI ISG NFV, IETF and ITU-T, are working on network slicing. The concept of network slicing has been introduced initially by the NGMN 5G whitepaper referenced in [10]. Slicing enables multiple logical self-contained networks to use a common physical infrastructure platform. Those logical networks enable a flexible stakeholder ecosystem for technical and business innovation that is integrating network and cloud resources into a programmable, software-oriented network environment as shown in Figure 7-9.

The logical self-contained networks can be realized by using: (1) virtualization, which is often defined as the act of moving physical systems to a digital environment and (2) Network Functions Virtualization (NFV) [11], which is the principle of separating network functions from the hardware they run on by using virtual hardware abstraction.

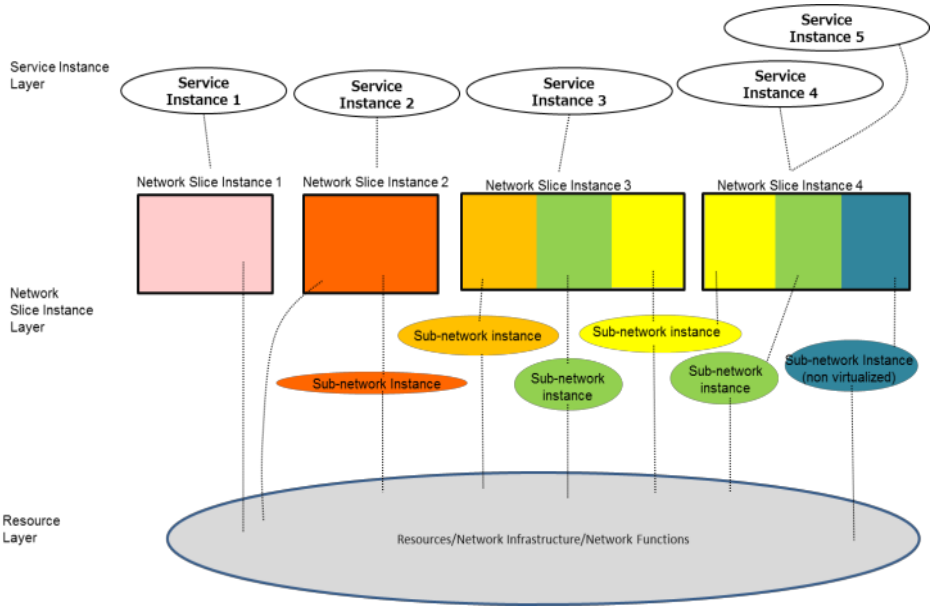


Figure 7-9: NGMN Network Slicing conceptual outline [10]

From the perspective of 3GPP [9], network slicing enables operators to create networks customized to provide optimized solutions for different market scenarios which demands diverse requirements, e.g. in the areas of functionality, performance and isolation. This is a key requirement from HLA and related IoT use cases and stakeholders such automotive, energy, cities, etc.

One of the key benefits of the network slicing concept, from IoT perspective, is that it enables value creation for vertical segments that lack physical network infrastructure, by offering network and cloud resources that can be used in an isolated, disjunctive or shared manner allowing a customized network

operation. Furthermore, network slicing can be used to support very diverse requirements imposed by IoT services and as well as flexible and scalable to support massive connections of different nature.

In particular, services such as smart households, smart grid, smart agriculture, and intelligent meter reading, will usually require supporting an extremely large number of connections and frequently transmitted small data packets. Other services such as smart vehicles and industrial control will require millisecond-level latency and nearly 100% reliability.

AIOTI is focusing on several key challenges to enable the fast deployment of IoT in Europe and globally, such as:

- Cope with IoT Rapid technological development
- Enlarge Users' take up and acceptability of IoT
- Enable fast move into deployment of IoT
- Avoid Risk of fragmentation in IoT
- Support cooperation on International level on IoT

As IoT is one of the most important enabling technologies for the vertical industries in Europe, AIOTI can serve as platform for these vertical industries and ensure that their needs are met by aligning their requirements. Network slicing can be used as the key enabler for the support and promotion of IoT in 5G scenarios.

NOTE - AIOTI WG03 in cooperation with the vertical AIOTI WGs can contribute on this topic in at least:

- collect requirements coming from AIOTI vertical industries members on how network slicing can be used to enable IoT in 5G scenarios,
- describe the relation between these collected requirements, the network slice types and the possible cross-industry domain customized services used to enhance the competence of vertical industries,
- describe how the AIOTI High Level Architecture (HLA) is used to specify IoT network slice architectures in 5G scenarios.

7.6.2.4 Device Virtualization

This clause describes one way of virtualizing the IoT devices, where devices may be highly resource constrained or not. This can function as an abstraction between physical devices and a virtualization layer by grouping devices into more complex virtual objects. Where the microservices-based Architectures and the NFV architectural framework focus on enabling actions, the device virtualization focusses on how individual devices are represented to the network. By grouping together devices that are supposed to intricately collaborate and represent them as one device to the network, a lot of clutter and complexity can be left out of the other virtualization layers.

The idea is to enable each IoT node with multiple functionalities based on its capability. Majorly, three important layers are identified, apart from the necessary connectivity layers such as PHY, MAC and Network layers: (i) Virtual Object (VO) layer, (ii) Composite Virtual Object (CVO) layer and (iii) Service layer. NOTE 1 - This virtualization architecture is based on the work of the EU Project iCore [33] and the European Commission's (EC) IoT-A project which looked into IoT architectural reference model **Error! Reference source not found.**.**Error! Reference source not found.**

Using VO abstraction of each device makes it easy to reuse the IoT devices. For example, the ambient light control in a smart building could indeed use the projector VO to realize that there is a movie/slide project in a particular room therefore lights can be turned-off.

The idea is to reuse IoT devices in multitudes of applications. Further, the CVO layer can help interface the IoT devices to interact with other devices and mashup multiple VOs to offer smart applications. For example, a smart home has requirements such as energy reduction, light control, climate control, security, etc. By combining multiple VOs these requirements could be served.

At the Service layer, multiple application requirements could be addressed. As given in the previous example, we can see that an ambient light control application can use information from the projector by querying IoTs in the vicinity to learn and make intelligent decisions. Of course, this requires semantic interoperability and languages such as OWL [35], etc. This is similar to a service-oriented architecture, multiple services from individual nodes or group of nodes can be merged with minimal human intervention.

An important aspect of this abstraction and segregation is that it supports distributed operation **Error! Reference source not found.**. As shown in Figure 7-10, the "IoT Daemon" encompasses the above abstractions. This way, multitudes of IoT devices can be integrated and interfaced.

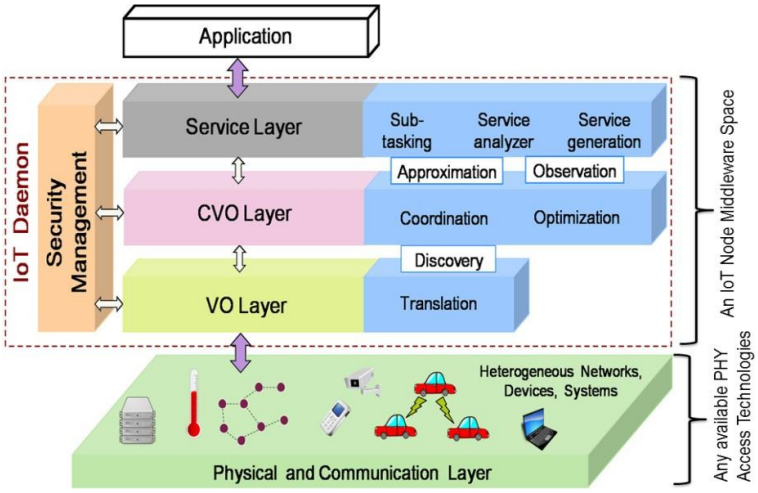


Figure 7-10: A high level architecture of (Composite) Virtual Objects

NOTE 2 - The cognitive capability can vary depending on the capability of the devices: some devices may have just enough capability to sense and send, then those devices may not have CVO and service levels. In some cases, sensors may not have even this capability and have their virtual presence in another device, for example a server or a powerful device, or an aggregator node like a raspberry pi.

Fig. 7-11 provides an abstract view of the interfaces between VO, CVO and Service layer functionalities.

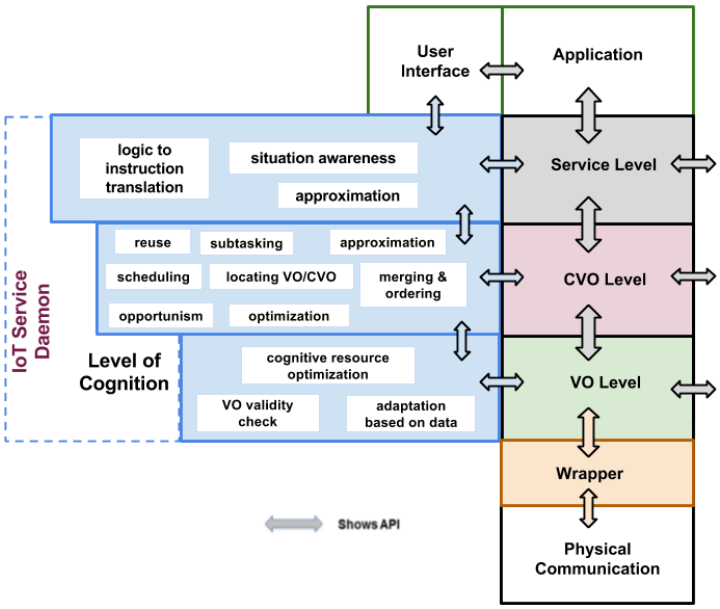


Figure 7-11: IoT device architecture and interfaces between the different layers

With respect to the two virtualization approaches described in clauses 7.6.2.1 and 7.6.2.2, VOs and CVOs focus less on the network, and more on the interaction with the individual devices.

This device virtualization can be actually part of any bigger platform, and, in particular, integrated in both the virtualization layer architectures described in clauses 7.6.2.1 and 7.6.2.2. Specifically, in the microservices architecture the “virtualized HW” can contain VOs and CVOs as defined above. Similarly, in the NFV framework, the virtualization layer can also contain VOs and CVOs. The Service layer can be used as interface in both the

microservices architecture and NFV framework or can be made transparent. The key value with VOs and CVOs is that these objects can indeed make use of the available resources optimally, collaborate with other IoT devices, offer redundancy and more, at the device level rather than at the whole architecture/framework level.

7.6.3 Comparing the IoT virtualization approaches

This clause is comparing the approaches described in clause 7.6.2.

NOTE - Network slicing is not subject to comparison, the main reason being that network slicing is, to a large extent, one illustration of the use of the NFV architecture, which would lead to very similar findings.

The microservices-based architectures and the NFV architectural framework

have been developed in different contexts. In particular, NFV in addressing primarily the "traditional" networks (e.g., those operated by Telecom Service Providers) and focuses on their major Network Functions. In contrast, the microservices-based functional architecture is spanning across high-layers of the "IoT Stack" and potentially addresses a larger set of "IoT functions".

The NFV architectural framework has been defined with the expectation that its approach to virtualization should be supported by a very precise set of standards (developed by NFV or not) supporting Reference Points. The challenge posed to virtualization is to make sure that the support of standards will not be compromised.

An important difference between the NFV approach and the microservices-based approach is that NFV is more focused on the functions related to the network and does not systematically take into account higher-layer functions.

The technologies available for the implementation of microservices-based applications have reached a level of maturity and effectiveness that has made their usage become mainstream in software engineering. The development of the Virtualized Network Functions of NFV is largely based on this approach. This is a strong enabler to the adoption of microservices-based architectures.

Despite the differences outlined above, the two approaches are not mutually exclusive and microservices (and microservices-based architectures) can be used in the NFV context, for example for the implementation of Virtualized Network Functions.

As opposed to the other two approaches, and anticipated above, the device virtualization approach focusses on the interaction between the individual devices. Virtual Objects and Virtual Composite Objects are a method to introduce an abstraction layer through which the devices and groups of devices present themselves to the network. Instead of a collection of very small and specific functionalities, the devices are grouped together to form complete virtual devices.

This group reports as a single entity to the network virtualization layer. The virtual devices can host subroutines, relieving the network virtualization layer of that effort.

Moreover, the network virtualization is unaware of the differences between virtual devices and real devices. Therefore, the development of network virtualization can proceed orthogonally to the device virtualization layer. The result is that a significant amount of clutter can be removed from the logic in the networked virtualization layer, and that the operations are more intuitive for a human designer.

Figure 7-12 illustrates how a Device Virtualization Layer with Composite Virtual Objects can be part of other approaches. In the NFV approach, the Device Virtualization resides between the hardware resources and the virtualization layer and is unaware of the difference.

In the microservices-based approach, the solution resides just below the virtualized hardware. A similar reasoning applies where the microservices-based approach acts as if the devices are real, while subroutines and clustering happen out of the scope of the network virtualization layer, and therefore simplifying the development.

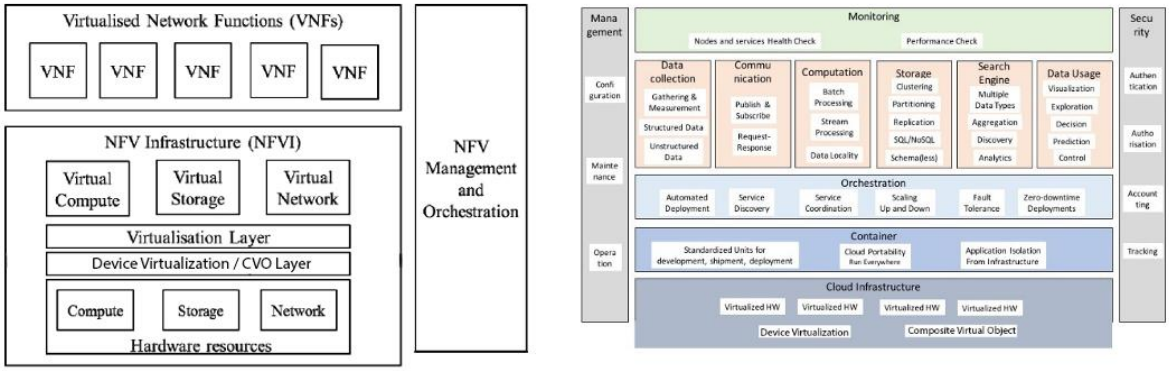


Figure 7-12: How Device Virtualization and Composite Virtual Objects can be leveraged by other approaches

7.6.4 The mapping of the IoT virtualization approaches on the AIOTI HLA

This clause is showing how a microservices-based functional architecture can be mapped on the AIOTI HLA.

In addition, another example of microservices-based functional architecture mapping is presented, the mapping on the oneM2M architecture.

NOTE - The relationship between the NFV architecture and the AIOTI HLA is not addressed here and may be developed in next Releases of this document.

7.6.4.1 The microservices-based approach and the AIOTI HLA

The mapping of a microservices-based functional architecture on the AIOTI HLA is straightforward since, as it has been outlined above, this example of microservices functional architecture has been defined with the goal to generically support IoT functions (e.g. location, discovery, identification). As a consequence, the example can be mapped on the IoT layer of the AIOTI HLA, as shown in Figure 7-13.

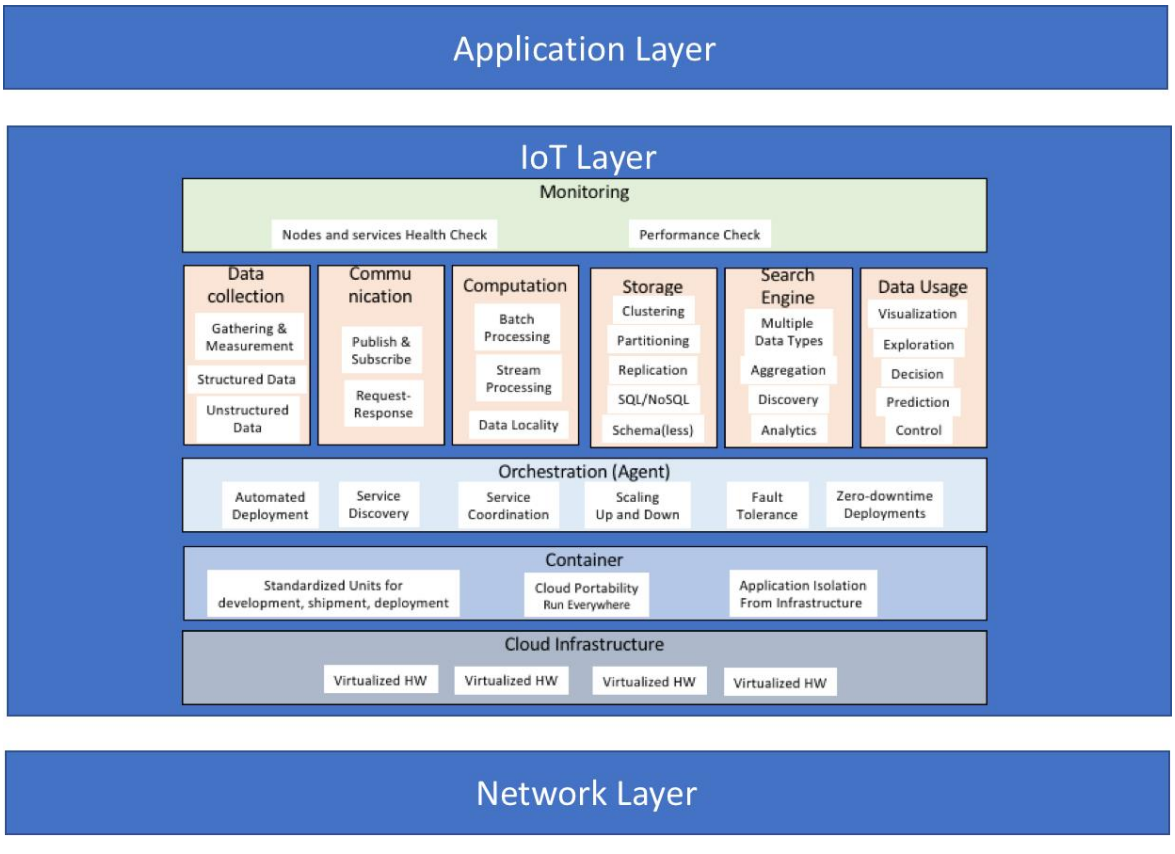


Figure 7-13: Mapping of microservice-based functional architecture on AIOTI HLA

7.6.4.2 The mapping of a microservices-based functional architecture onthe oneM2M architecture

Like for NFV, the oneM2M architectural framework has been defined with the expectation that its approach to virtualization should be supported by a very precise set of standards (developed by NFV or not) supporting Reference Points. Here again, the challenge posed to virtualization is to make sure that the support of standards will not be compromised.

oneM2M defines a list of Common Service Functions (CSFs) as an "informative architectural construct which conceptually groups together a number of sub-functions". The CSF descriptions are provided for the purpose of understanding of the oneM2M Architecture functionalities and are informative. The CSFs contained inside the Common Services Entity (CSE) can interact with each other but oneM2M TS-0001 [26] does not specify how these interactions take place.

The respective positioning of oneM2M Common Service Entities (CSE) and the microservices in the microservices-based functional architecture described in clause 7.6.2.1 is shown in Figure 7-14:

- There is a difference between the CSFs (that are specified via a standard) and the microservices that are one possible implementation of (a subset of) a CSF;
- All (or part of) the microservices described in Figure 7-7 can be included in a given CSE. The choice of microservices and their implementations can (and probably will) be different from one CSF to another. Consequently, there is no standardised mapping of one CSF to microservices.

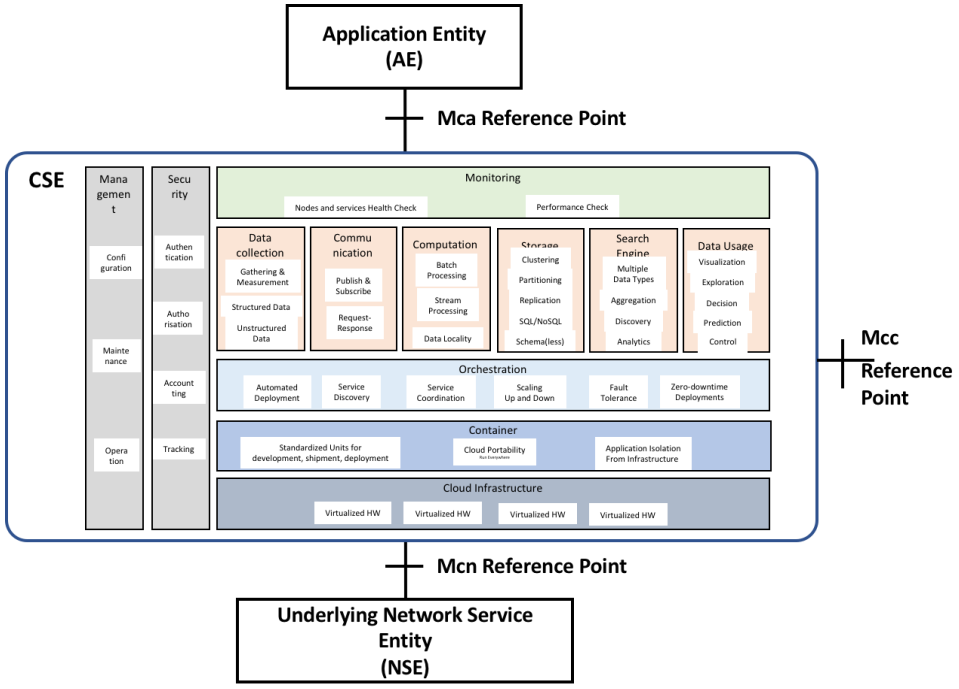


Figure 7-14: Mapping of microservices-based functional architecture on oneM2M Common Service Entities

The CSFs have not been defined with a microservices-based architecture in mind. Indeed, the choice of dividing a CSE into microservices should always be left up to specific implementations, which means that the optimizations made for two different deployment scenarios may result in two different choices of grouping into microservices.

7.7 IoT platforms

Editor's note – This clause is in draft stage. Its completion is expected in following Releases of this document. Consideration could be also given to interoperability between IoT platforms and other non IoT platforms.

The focus of the industry has gradually shifted to the design and development of IoT systems with the purpose to offer full-fledge systems dealing with a vast number of devices (with various computing and interaction capabilities) and potentially integrating these devices into larger systems implementing often complex business processes. This has been enabled by the emergence of IoT devices with higher computing capacity and the possibility of producing massive amounts of data that will be collected, transformed, stored and managed by larger (non IoT specific) information systems which transform this data into qualitative information to trigger useful actions.

The "standardised IoT platforms" will have to address the challenges and probably not all of the existing ones will be able to make it.

Three main challenges have to be addressed by IoT standardisation (organisations) and by the "standardised" platforms (an example is oneM2M), that some of these organisations are developing:

- The "advanced technology" challenge posed by e.g., the incorporation of Big Data or Virtualization;
- The "business sector" challenge with the question of which level of genericity can be provided in support of the development of large IoT systems for Smart Cities, Intelligent Transport or Industrial IoT;
- The "standards" challenge posed by the role of emerging approaches such as Open Source.

IoT systems are often seen as an extension to existing systems needed because of the (potentially massive) addition of networked devices. However, this approach does not take stock of a set of essential characteristics of IoT systems that push for an alternative approach where the IoT system is at the centre of attention of those who want to make them happen. This advocates for an "IoT-centric" view.

Most of the above-mentioned essential characteristics may be found in other ICT-based systems. However, the main difference with IoT systems is that they all have to be dealt with simultaneously.

The most essential ones are:

- Stakeholders. There is a large variety of potential stakeholders with a wide range of roles that shape the way each of them can be considered in the IoT system. Moreover, none of them can be ignored.
- Privacy. In the case of IoT systems that deal with critical data in critical applications (e.g., e-Health, Intelligent Transport, Food, Industrial systems), privacy becomes a make or break property.
- Interoperability. There are very strong interoperability requirements because of the need to provide seamless interoperability across many different systems, sub-systems, devices, etc.
- Security. As an essential enabling property for Trust, security is a key feature of all IoT systems and needs to be dealt with in a global manner. One key challenge is that it is involving a variety of users in a variety of Use Cases.
- Technologies. By nature, all IoT systems have to integrate potentially very diverse technologies, very often for the same purpose (with a risk of overlap). The balance between proprietary and standardised solutions has to be carefully managed, with a lot of potential implications on the choice of the supporting platforms.
- Deployment. A key aspect of IoT systems is that they emerge at the very same time where Cloud Computing and Edge Computing have become mainstream technologies. All IoT systems have to deal with the need to support both Cloud-based and Edge-based deployments with the associated challenges of management of data, etc.
- Legacy. Many IoT systems have to deal with legacy (e.g., existing connectivity, back-end ERP systems). The challenge is to deal with these requirements without compromising the “IoT centric” approach.

A drawback of many current approaches to system development is a focus on the technical solutions, which may lead to suboptimal or even ineffective systems. In the case of IoT systems, a very large variety of potential stakeholders are involved, each coming with specific – and potentially conflicting – requirements and expectations. Their elicitation requires that the precise definition of roles that can be related to in the analysis of the requirements, of the Use Cases, etc. Examples of such roles to be characterised and analysed are: System Designer, System Developer, System Deployer, End-user, Device Manufacturer.

In order to better achieve interoperability, many elements (e.g., vocabularies, definitions, models) have to be defined, agreed and shared by the IoT stakeholders. This can ensure a common understanding across them of the concepts used for the IoT system definition. They also are a preamble to standardisation. Moreover, given the need to be able to deal with a great variety of IoT systems architectures, it is also necessary to adopt Reference Architectures, in particular Functional Architectures such as the AIOTI High-Level Architecture.

A very large number of IoT platforms have been developed with the initial purpose of ensuring that a device could interact with other devices or equipment, providing connectivity from point-to-point to more universal. Standard Development Organisations (SDOs) and Standard Setting Organisations (SSOs) have developed a number of approaches that focused on interoperability, initially at the network level and now well beyond. Many standards have been defined with the possibility to serve as a basis for the development of platforms that – in the best case - deal with interoperability in a generic manner, across a variety of business sectors, with a variety of possible implementations. Such "standardised platforms" are relying on reference architectures, interoperability stacks addressing different layers, generic protocol adaptors, etc.

7.7.1 Generalities on IoT platforms

7.7.1.1 IoT platform concepts and taxonomies

Editor's note: content could be brought here based on results of efforts such as IoT-EPI, relevant H2020 projects, ETSI STFs, other. To consider also if some relevant examples of different platform types could be appropriate for insertion here (or in specific subclause).

7.7.1.2 Identified gaps related to IoT platforms

Editor's note: content could be brought here based on results of specific efforts such as ETSI STFs.

7.7.2 Positioning of IoT platforms in HLA

Editor's note: content could address the positioning of main platform components in HLA layers (with possible needed extensions/details of HLA), with consideration of interoperability aspects.

7.7.3 IoT platform to platform interoperability

7.7.3.1 Generalities

Editor's note: content could cover the need of platform interoperability (cross-domain etc.), the specific issues for platform interoperability and the dimensions to be addressed (data management (incl. (real time) data discovery, data storage, data exchange and commercialization, data security and privacy, open data, …), context awareness, stakeholders' roles, …). Content could be brought here based on results of efforts such as relevant H2020 projects, ETSI STF 575, other.

7.7.3.2 Approaches for IoT platform to platform interoperability

Editor's note: content could cover the different approaches for platform interoperability, each approach being described according to how it addresses identified platform interop issues and deal with the dimensions identified in 7.6.3.1 (pros and cons of each approach).

7.7.3.2.1 Approach "1": ETSI STF 547 TR 103536

Editor's note: to be developed.

7.7.3.2.2 Approach "2": Data lake approach

Editor's note: to be developed.

7.7.3.2.3 Approach "3": usage of intermediate standardized platform

Editor's note: It is for consideration further generalization in terms of general "intermediate standardized platform" (including consideration about abstraction from Autopilot-specific text).

Interoperability between proprietary IoT platforms can be accomplished using a common platform that acts as an intermediate platform interconnecting the proprietary IoT platforms (and possibly devices and services) and allows them to exchange information. One important characteristic of such common platform is its standardisation: it should provide open interfaces as well it should enable standardised ways of mapping some of the interfaces used by the proprietary platforms to its open interfaces.

7.7.3.2.3.1 The oneM2M platform as intermediate standardized platform

An intermediate standardized platform which fulfils the common platform characteristic highlighted above, is the oneM2M platform.

The following describes an application example of the oneM2M platform as intermediate standardized platform, based on the outcome of the European Commission's Horizon 2020 AUTOPILOT (Automated Driving Progressed by Internet of Things) project.

The Horizon 2020 AUTOPILOT project focuses on creating a connected IoT ecosystem for automated vehicles and uses oneM2M as an interoperability platform. [37] and [38] discuss the IoT platform interoperability challenges and their solutions as proposed in the AUTOPILOT project.

The AUTOPILOT Federated IoT architecture [38] is shown in **Error! Reference source not found.**.

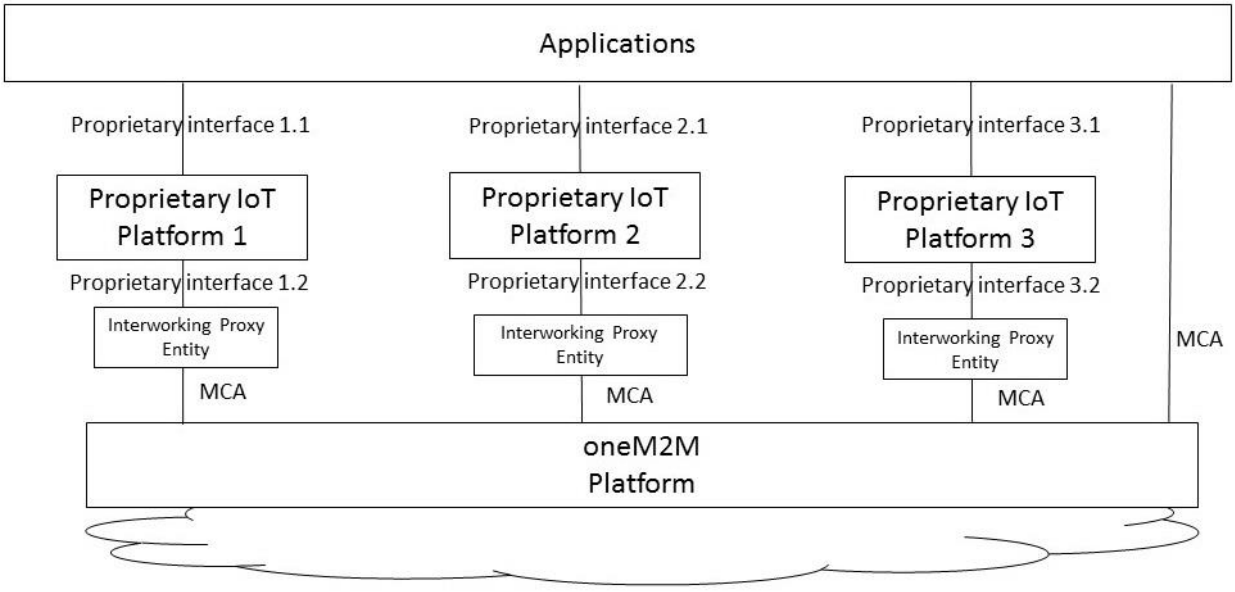


Figure 7-15: AUTOPILOT Federated IoT Architecture

The AUTOPILOT Federated IoT architecture includes devices and gateways; in-vehicle and road-side IoT platforms exchange information with several distributed IoT platforms, which may be deployed at different levels.

The following two types of IoT platforms are distinguished:

- proprietary IoT platforms: used by some applications, organisations and services to exchange specific data with specific devices or vehicles. NOTE 1 - The platforms of this type used in the AUTOPILOT project are: Watson IoT Platform™, FIWARE and Huawei Ocean Connect.
- oneM2M interoperability platform: the central IoT platform that acts as a hub interconnecting the proprietary IoT platforms (and possibly devices and services) and allowing them to exchange information. This interoperability platform is based on the oneM2M machine to machine standards, which are adopted by the project as the standards for interoperability. NOTE 2 - In the AUTOPILOT project, the Sensinov oneM2M-based platform is used.

The proprietary IoT platforms are connected to the oneM2M interoperability platform through oneM2M Interworking Proxy Entities (IPEs). Each proprietary IoT platform may configure the IPE to share selected data types, relevant to Automated Driving vehicles and applications, with the oneM2M interoperability platform. The goal of this process is that such data may then become accessible and be shared by all the connected proprietary IoT platforms through the oneM2M interoperability platform.

7.7.3.2.3.1.1 Details about IoT platform interoperability in the AUTOPILOT project

The AUTOPILOT IoT platform aims to enable a large-scale and open IoT ecosystem, where new "things" (sensors, vehicles), services, applications, and IoT platforms may be plugged in easily, and may start exchanging information with the rest of the ecosystem components. In particular, as no single "standard IoT platform" exists, the AUTOPILOT architecture has to rather cope with a multitude of proprietary IoT solutions distributed over various physical infrastructures and dedicated to different geographic areas, services, or providers. The key challenge exists to connect these proprietary IoT platforms and make them communicate with each other to exchange information.

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

- oneM2M IoT Standards: Proprietary IoT platforms are interconnected through a standard oneM2M interoperability platform and oneM2M interworking gateways. By adopting the oneM2M standards, AUTOPILOT aims to facilitate interoperability between the various IoT platforms, sensors, and services of the architecture by using:
    - oneM2M interoperability platform to act as a central hub connecting the various proprietary IoT platforms, allowing them to exchange data and information through standard oneM2M protocols and APIs.
    - Interworking Proxy Entity (IPE), that is a specialized oneM2M AE (Application Entity) that allows the oneM2M system to interact with any non-oneM2M system, in a seamless way, through the Mca interface [39]. It has the capability to remap a specific data model to oneM2M resources and maintain bidirectional communication with the non-oneM2M system.
- IoT Data Models: by using IoT data required to be exchanged across the IoT platforms, based, whenever possible, on existing data models and specifications (such as DATEX II |45] for exchanging car park availability and traffic data, and SENSORIS |46] for sharing vehicle location and object detection data). The AUTOPILOT IoT data models cover the following packages:
    - Vehicle location and detection messages, based on SENSORIS,
    - Event and object detection messages to be consumed by AD vehicles, based on SENSORIS and DATEX II,
    - Traffic situations, based on DATEX II,
    - Parking availability information, based on the DATEX II parking extension,
    - Messages specific to automated valet parking, car sharing, rebalancing, and platooning.
- Standardised Ontologies: Semantic interoperability is supported by semantically standardising IoT data field values (e.g. hazard types, vulnerable road user types, detected object types, etc.) using ontologies.

7.7.3.2.3.2 oneM2M IoT platform interoperability with AIOTI HLA-compliant IoT platform

The support for IoT platform interoperability based on the solution provided by the EC H2020 AUTOPILOT project can be encompassed in the AIOTI HLA as shown in Figure 7-16.

The left part of Figure 7-16 shows the oneM2M IoT platform compliant to AIOTI HLA (identical to the one shown in Figure 8.3) and the right part shows (an IoT platform compliant to) the AIOTI HLA as shown in Figure 5.2. As additional entity, Figure 7-16 shows the Interworking Proxy Entity, a specialized oneM2M AE (Application Entity) that allows the oneM2M system to interact with any non-oneM2M (Proprietary) system.

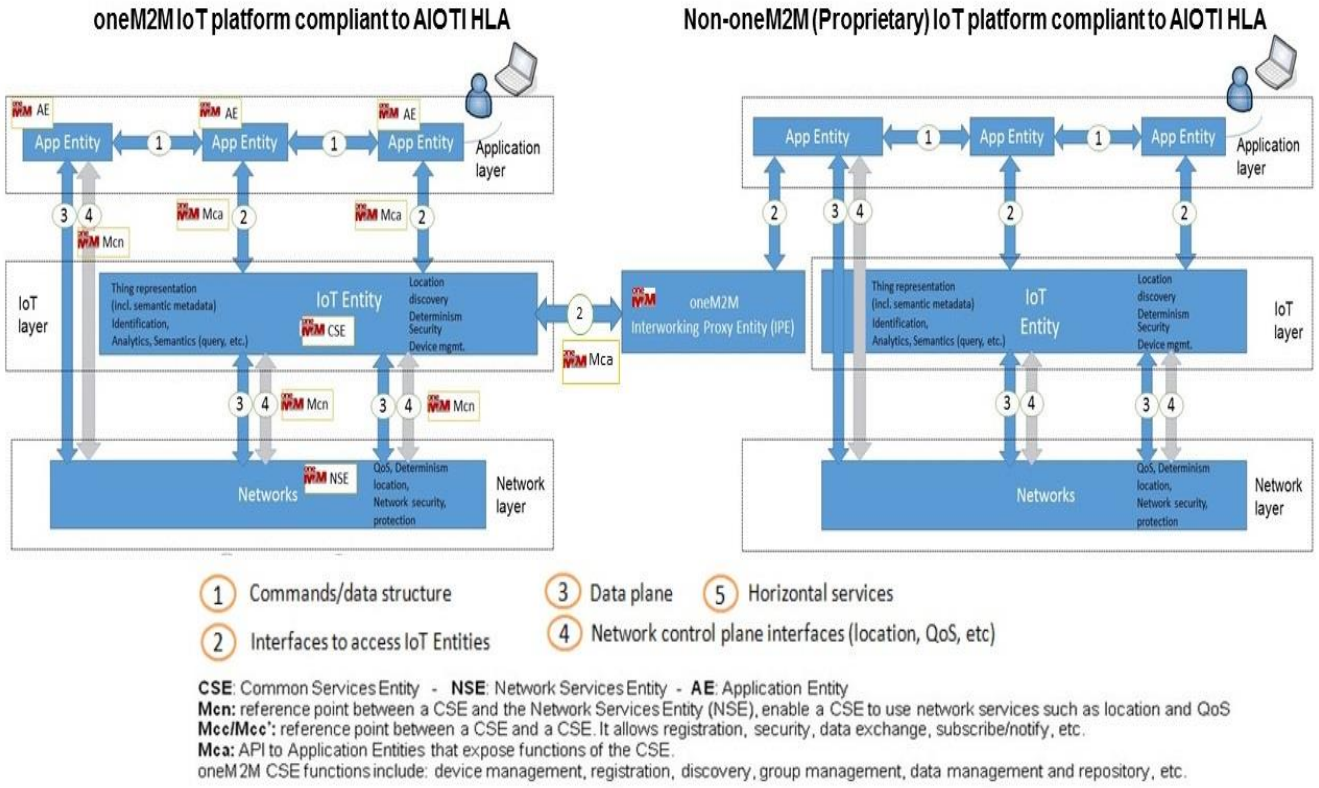


Figure 7-16: oneM2M IoT Platform Interoperability with AIOTI HLA-compliant IoT platform

7.7.3.2.4 Approach "4" (e.g. direct integration between platforms)

Editor's note: to be developed

8 Mapping of SDOs' work to the AIOTI HLA functional model

The purpose of this clause is to provide examples of mapping of existing SDO/alliances/projects architectures to the AIOTI HLA functional model. The intent of this mapping exercise is three- fold:

- Demonstrate that AIOTI HLA is closely related to existing architectures and architectural frameworks
- Provide positioning of existing standards vis-à-vis the HLA
- Derive any possible important gaps in the HLA (even if the HLA aims to remain high-level)

This clause does not intend to be exhaustive, other mappings can be added in future releases of this document.

8.1 ITU-T

In ITU-T Recommendation Y.4000 "Overview of the Internet of Things" [3], ITU-T has developed an IoT Reference Model which provides a high level capability view of an IoT infrastructure. As shown in Figure 8-1, the model is composed of the following layers, providing corresponding sets of capabilities [Note - likewise for the AIOTI HLA, a layer represents here a grouping of modules offering a cohesive set of services]:

- Application Layer (Application capabilities)
- Service Support and Application Support Layer (SSAS capabilities - distinguished into Generic support capabilities and Specific support capabilities)
- Network Layer (Network capabilities - distinguished into Networking capabilities (Control plane level) and Transport capabilities (Data plane level))
- Device Layer (Device/Gateway capabilities)

The Security capabilities and Management capabilities - both distinguished into Generic Security (Management) capabilities and Specific Security (Management) capabilities – are cross-layer, i.e. they can be provided in support of different capability groupings.

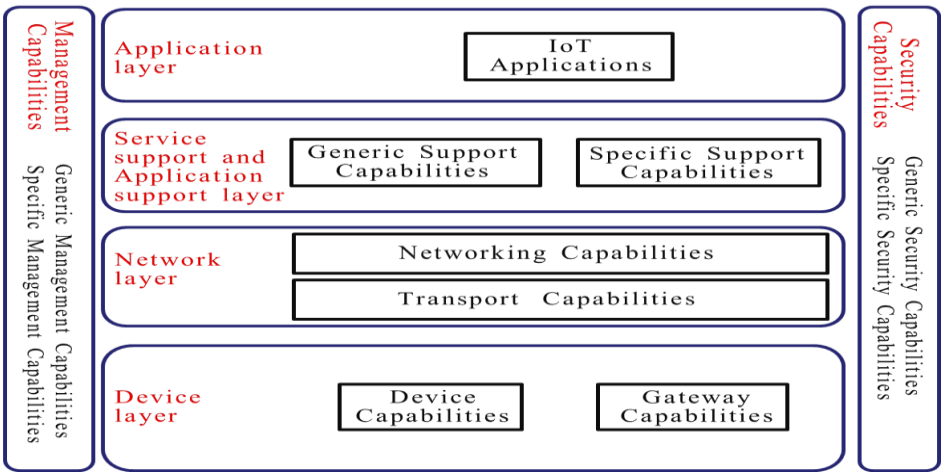
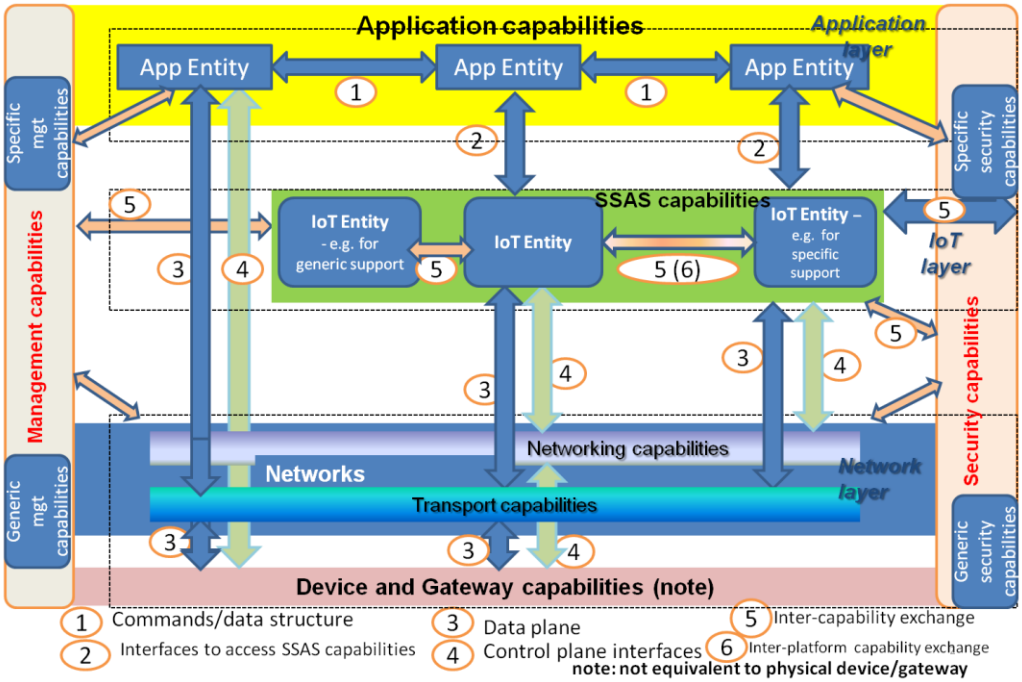


Figure 8-1: ITU-T Y.4000 IoT Reference Model

Figure 8-2 provides an initial high level mapping of the ITU-T Y.4000 IoT Reference model to AIOTI HLA functional model.

Figure 8-2: ITU-T IoT Reference Model mapping to AIOTI HLA functional model

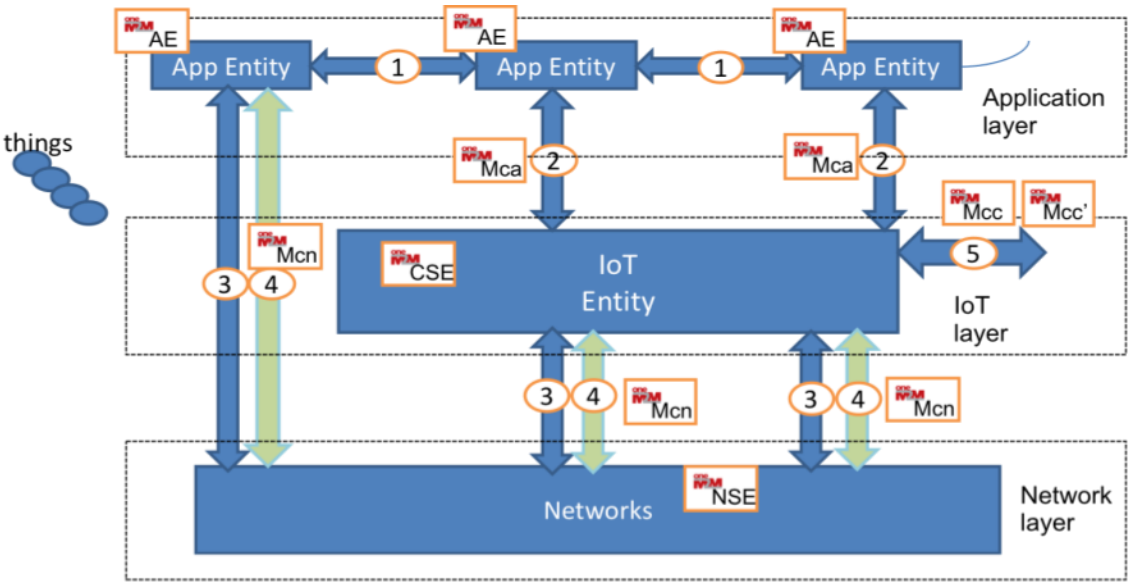


Various detailed studies related to IoT functional framework and architectural aspects have been developed or are currently in progress within ITU-T; relevant ones include ITU-T Rec. Y.4401 (“Functional framework and capabilities of the Internet of things”), ITU-T Recommendation F.748.5 (“Requirements and reference architecture of M2M service layer”) and ITU-T Recommendation Y.4416 (“Architecture of the Internet of Things based on NGN evolution”).

8.2 oneM2M

Figure 8-3 provides the mapping between oneM2M and the AIOTI HLA functional model. oneM2M specifies a Common Services Entities (CSE) which provide IoT functions to oneM2M AEs (Applications Entities) via APIs [4]. The CSEs also allows leveraging underlying network services (beyond data transport) which are explicitly specified in oneM2M and referred to as Network Services Entity (NSE).

Figure 8-3: Mapping oneM2M to AIOTI HLA



CSE: Common Services Entity - **NSE**: Network Services Entity - **AE**: Application Entity
Mcn: reference point between a CSE and the Network Services Entity (NSE), enable a CSE to use network services such as location and QoS
Mcc/Mcc': reference point between a CSE and a CSE. It allows registration, security, data exchange, subscribe/notify, etc.
Mca: API to Application Entities that expose functions of the CSE.
oneM2M CSE functions include: device management, registration, discovery, group management, data management and repository, etc.

oneM2M has specified all interfaces depicted in Figure 8-3 to a level that allows for interoperability. Three protocols binding have been specified for Mcc and Mca reference points: CoAP, MQTT, Websockets, and HTTP. As regards the Mcn reference point, normative references have been made to interfaces specified by 3GPP and 3GPP2 in particular.

However, oneM2M does not specify vertical specific data formats for exchange between App Entities according to AIOTI HLA interface 1. This can however be achieved by interworking with other technologies such as ZigBee, AllSeen, etc.

8.3 IIC

The Industrial Internet reference Architecture (IIRA) is a standard-based open architecture [5]. “The description and representation of the architecture are generic and at a high level of abstraction to support the requisite broad industry applicability” (source IIC). Figure 8-4 provides a three-tier architecture as specified in [5].

Figure 8-4: IIC three tier IIS architecture

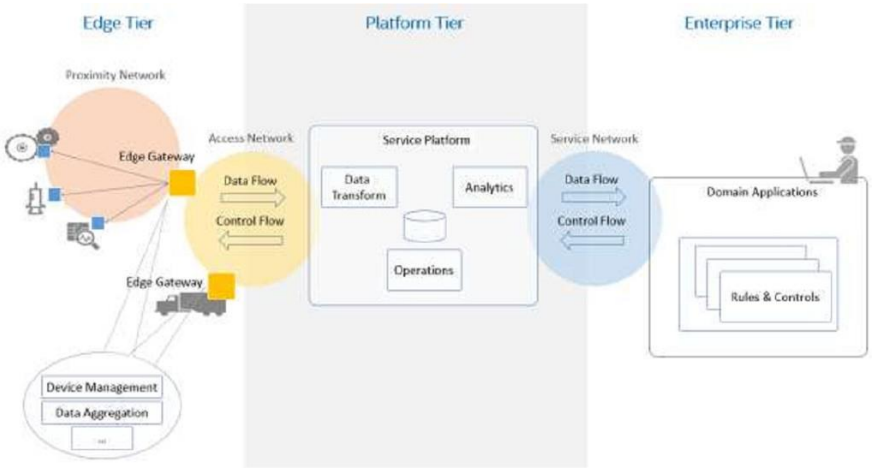
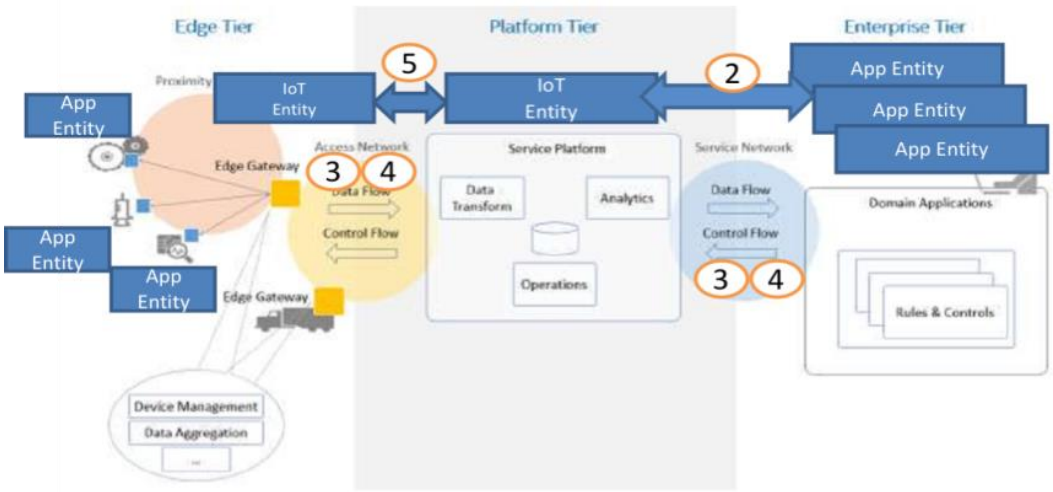


Figure 8-5: Mapping HLA to IIC three tier IIS architecture



The mapping of IIC to the AIOTI HLA is depicted in the following Figure. In Figure 8-5, devices in the IIC proximity domain would typically run App Entities according to the AIOTI HLA. The Edge gateways would in turn be mapped to IoT Entities, implementing as an example device management for proximity network devices.

Interactions with the network for the purpose of data exchange or other network services are depicted through the interface 3 and 4 from the AIOTI HLA. Finally, the Application Domain in IIC would be equivalent to AIOTI App Entities running in the enterprise data centres.

8.4 RAMI 4.0

Industrie 4.0 covers a highly diverse landscape of industries, stakeholders, processes, technologies and standards. To achieve a common understanding of what standards, use cases, etc. are necessary for Industrie 4.0, a uniform architecture model (the Reference Architecture Model Industrie 4.0 (RAMI 4.0)) was developed by VDI/VDE GMA & ZVEI in Germany [16], serving as a basis for the discussion of interrelationships and details. RAMI 4.0 has been further defined by DIN as DIN SPEC 91345 [17] and IEC as IEC PAS 63088 [18].

Besides the reference architecture model, RAMI 4.0 defines the I4.0 component which links the assets in the Industrie 4.0 environment like products, production machines or production lines and systems with their virtual presentation in cyber space the so called administration shell.

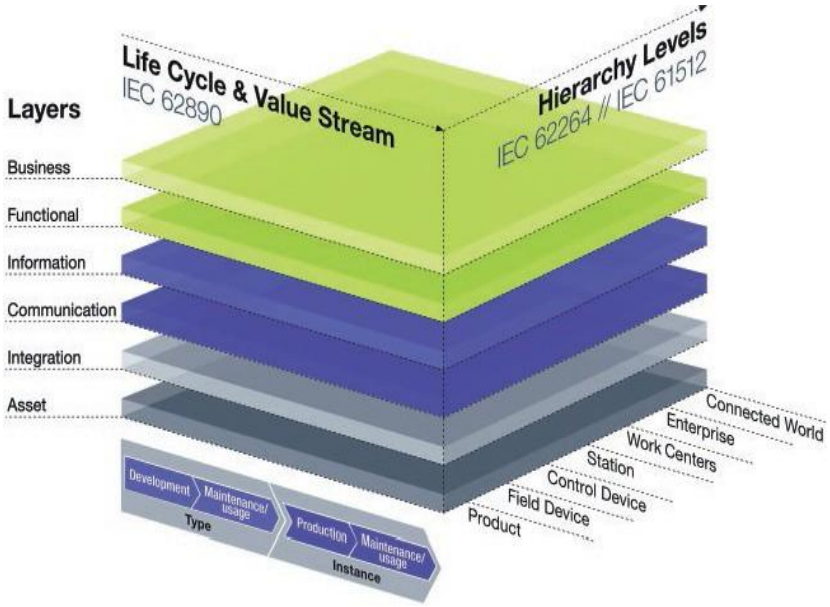


Figure 8-6: RAMI 4.0 reference architecture

The reference architecture model as shown in Figure 8-6 structures the Industrie 4.0 space into its fundamental aspects. It expands the hierarchy levels of IEC 62264 [19] by adding the "Field Device" and "Product" or work piece level at the bottom, and the "Connected World" going beyond the boundaries of the individual factory at the top. The left horizontal axis represents the life cycle of systems or products and the value stream of production. It also establishes the distinction between "Type" and "Instance". Finally, the six vertical layers on the left define various architectural viewpoints on Industrie 4.0 that are relevant from a system design and standardization point of view. The specific characteristics of the reference architecture model are therefore its combination of life cycle and value stream with a hierarchically structured approach of various architectural views.

The mapping of RAMI 4.0 to the AIOTI HLA – functional model - is depicted in the following Figure.

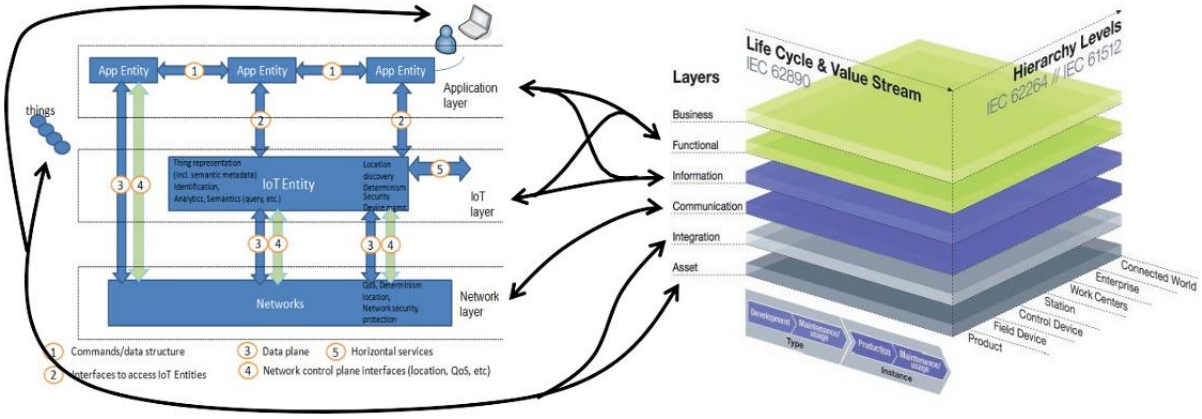


Figure 8-7: Mapping RAMI 4.0 to AIOTI HLA – functional model

The following explanations can be made as regards Figure 8-7:

As the AIOTI HLA and RAMI 4.0 have different purposes and approaches only a rough mapping can be performed and a 1 to 1 relation between the components in the two models is not always possible.

- The HLA Network layer represents the IoT communication capabilities and maps to the RAMI
- 4.0 Communication Layer
- The HLA IoT and App Layer represent functional and information components that map to the RAMI 4.0 Functional and Information layers
- Things, People, HW components map to the RAMI 4.0 Asset and Integration layer
- Note that functions at the network, IoT and App Layer like routers, data storage and processing would appear at the RAMI 4.0 functional layer from a functional point of view and in the physical representation at the asset layer

The mapping of RAMI 4.0 to the AIOTI HLA – domain model - is depicted in the following Figure.

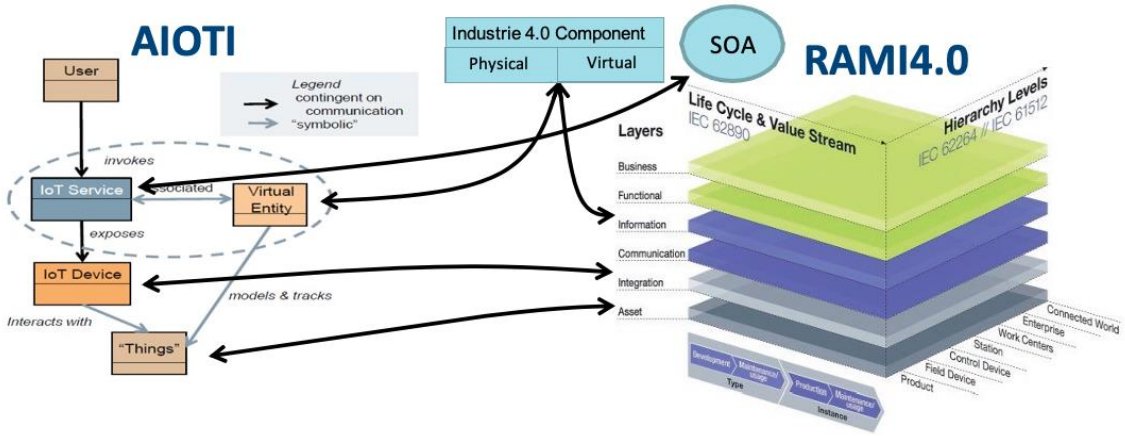


Figure 8-8: Mapping RAMI 4.0 to AIOTI HLA – domain model

The following explanations can be made as regards Figure 8-8:

- The Things in HLA are equivalent to the Asset layer of RAMI 4.0. They are the physical part of the I4.0 component and can appear at all hierarchy levels from products to field devices like sensor to whole production lines and even factories.
- In HLA, Things are represented by virtual entities in the digital world. This corresponds to the virtual part of the Industrie 4.0 component of RAMI 4.0
- The HLA IoT Device performs the interaction between the physical things and the digital world. In RAMI 4.0 this is a task of the Integration layer.
- With the HLA IoT Service the Service Oriented Architecture (SOA) approach of RAMI 4.0 is supported.

8.5 Big Data Value Association

The BDVA Big Data Value Reference Model (from the BDVA SRIA 4.0 document [31]) is shown in the figure below.

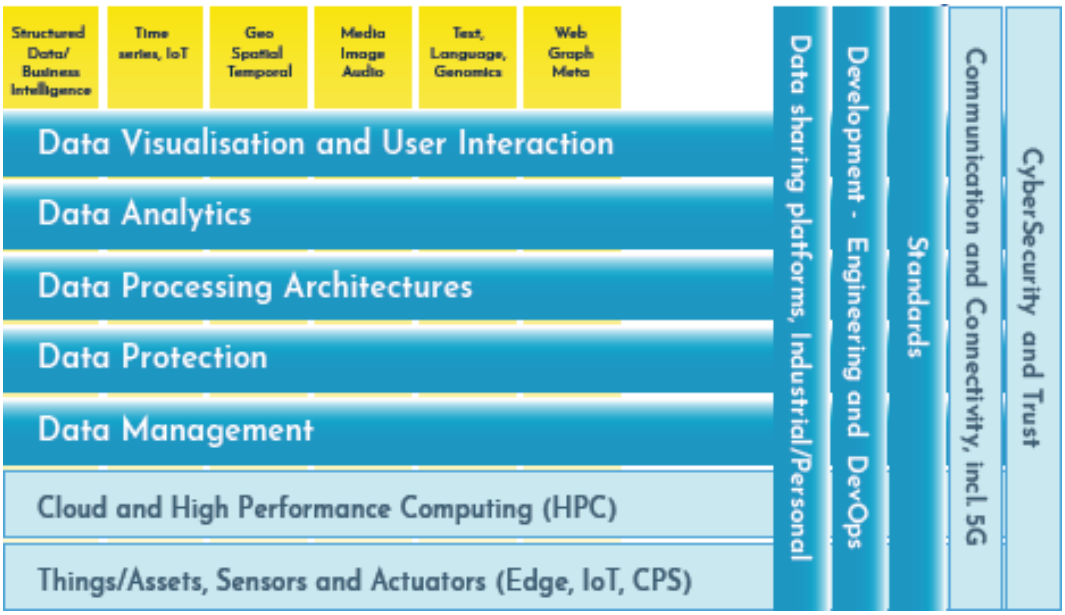


Figure 8-9 - Big Data Value Association – BDV Reference Model

The BDV Reference Model has been developed by the Big Data Value Association (BDVA), taking into account input from technical experts and stakeholders along the whole Big Data Value chain as well as interactions with other related PPPs. An explicit aim of the BDV Reference Model in the SRIA 4.0 document is to also include logical relationships to other areas of a digital platform such as Cloud, High Performance Computing (HPC), IoT, Networks/5G, CyberSecurity etc.

The BDV Reference Model may serve as common reference framework to locate Big Data technologies on the overall IT stack. It addresses the main concerns and aspects to be considered for Big Data Value systems.

The BDV Reference Model is structured into horizontal and vertical concerns.

- **The horizontal concerns** cover specific aspects along the data processing chain, starting with data collection and ingestion, reaching up to data visualization. It should be noted, that the horizontal concerns do not imply a layered architecture. As an example, data visualization may be applied directly to collected data (data management aspect) without the need for data processing and analytics. Further data analytics may take place in the IoT area – i.e. Edge Analytics. Logical areas are shown, but they might execute in different physical layers.
- **The vertical concerns** address cross-cutting issues, which may affect all the horizontal concerns. In addition, verticals may also involve non-technical aspects (e.g., standardization as technical concerns, but also non-technical ones).

Given the purpose of the BDV Reference Model to act as a reference framework to locate Big Data technologies, it is purposefully chosen to be as simple and easy to understand as possible. It thus does not have the ambition to serve as a full technical reference architecture. However, the BDV Reference Model is compatible with such reference architectures, most notably the emerging ISO JTC1 WG9 Big Data Reference Architecture – now being further developed in ISO JTC1 SC42 Artificial Intelligence [32].

The remainder of this clause elaborates the technical areas as expressed in the BDV Reference Model.

Horizontal concerns:

- Big Data Applications: Solutions supporting big data within various domains will often consider the creation of domain specific usages and possible extensions to the various horizontal and vertical areas. This is often related to the usage of various combinations of the identified big data types described in the vertical concerns.
- Data Visualization and User Interaction: Advanced visualization approaches for improved user experience.
- Data Analytics: Data analytics to improve data understanding, deep learning, and meaningfulness of data.
- Data Processing Architectures: Optimized and scalable architectures for analytics of both data-at-rest and data-in- motion with low latency delivering real-time analytics.
- Data Protection: Privacy and anonymization mechanisms to facilitate data protection. It also has links to trust mechanisms like Blockchain technologies, smart contracts and various forms for encryption. This area is also associated with the area of CyberSecurity, Risk and Trust.
- Data Management: Principles and techniques for data management including both data life cycle management and usage of data lakes and data spaces, as well as underlying data storage services.
- Cloud and High Performance Computing (HPC): Effective big data processing and data management might imply effective usage of cloud and high performance computing infrastructures. This area is separately elaborated further in collaboration with the Cloud and High Performance Computing (ETP4HPC) communities.
- IoT, CPS, Edge and Fog Computing: A main source of big data is sensor data from an IoT context and actuator interaction in Cyber Physical Systems. In order to meet real- time needs, it will often be necessary to handle big data aspects at the edge of the system.

Vertical concerns:

- Big Data Types and semantics: The following six big data types have been identified - based on the fact that they often lead to the use of different techniques and mechanisms in the horizontal concerns, which should be considered, for instance for data analytics and data storage: 1) Structured data; 2) Times series data; 3) GeoSpatial data, 4) Media, Image, Video and Audio data; 5) Text data, including Natural Language Processing data and Genomics representations; 6) Graph data, Network/Web data and Meta data. In addition, it is important to support both the syntactical and semantic aspects of data for all big data types.
- Standards: Standardisation of big data technology areas to facilitate data integration, sharing and interoperability.
- Communication and Connectivity: Effective communication and connectivity mechanisms are necessary for providing support for big data. This area is separately elaborated further with various communication communities, such as the 5G community.
- Cybersecurity: Big Data often need support to maintain security and trust beyond privacy and anonymization. The aspect of trust frequently has links to trust mechanisms such as blockchain technologies, smart contracts and various forms of encryption. The CyberSecurity area is separately elaborated further with the CyberSecurity PPP community.
- Engineering and DevOps: for building Big Data Value systems. This area is also elaborated further with the NESSI (Networked European Software and Service Initiative) Software and Service community.
- Data Platforms: Marketplaces, IDP/PDP, Ecosystems for Data Sharing and Innovation support. Data Platforms for Data Sharing include in particular Industrial Data Platforms (IDPs) and Personal Data Platforms (PDPs), but also include other data sharing platforms like Research Data Platforms (RDPs) and Urban/City Data Platforms (UDPs). These platforms include efficient usage of a number of the horizontal and vertical big data areas, most notably the areas for data management, data processing, data protection and cybersecurity.
- AI platforms: In the context of the relationship between AI and Big Data there is an evolving refinement of the BDV Reference Model – showing how AI platforms typically include support for Machine Learning, Analytics, visualization, processing etc. in the upper technology areas supported by data platforms – for all of the various big data types.

8.5.1 Mapping of the BDV Reference Model to the AIOTI HLA

NOTE 1 - The mapping of the BDV Reference Model to the AIOTI HLA described in this clause reflects the initial understanding of the team of AIOTI WG03 people who have contributed to the study and is subject to further enhancements in next Release(s) of this document.

A mapping of the BDV Reference Model to the AIOTI HLA is shown in Figure 8-10.

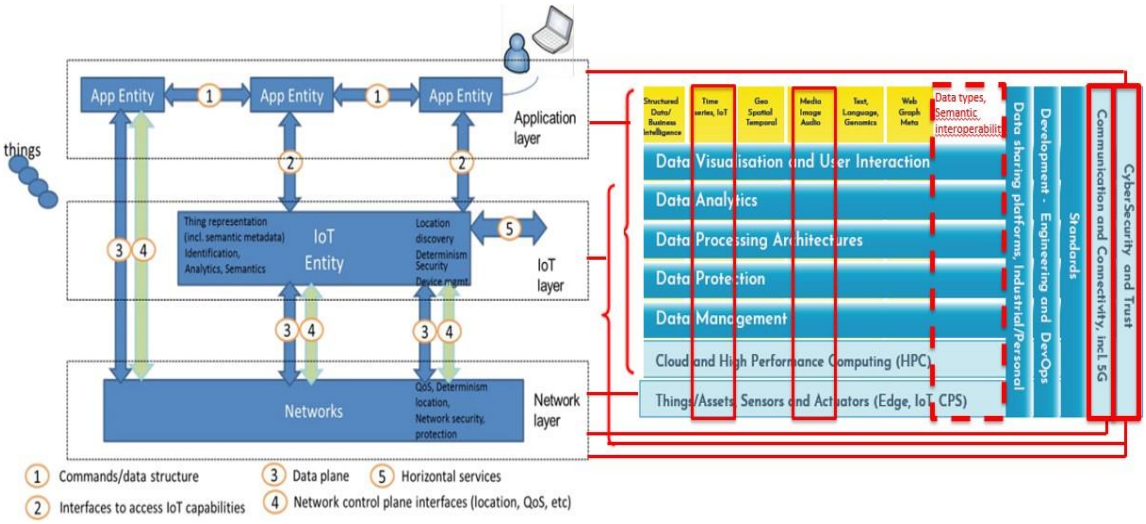


Figure 8-10 - BDV Reference Model mapping to the AIOTI HLA

NOTE 2 - The BDV Reference Model shows technical areas and capabilities, but without a particular layering perspective. The different capabilities may reside in different clients and servers in different configurations.

NOTE 3 - The Time Series/IoT and Media/Image/Audio Data types of the BDV Reference Model, because of their particular interest in an IoT context, are marked in red across the various technical areas of the BDV Reference Model.

NOTE 4 - The Semantic Interoperability focus through data types of the BDV Reference Model is shown via (red) dotted line in order to highlight its relevance in both the BDV Reference Model and the AIOTI HLA context.

The followings are key considerations concerning the BDV Reference Model mapping to the AIOTI HLA.

The App Entities of the AIOTI HLA provide application logic which may include data visualization and user interaction services, data analytics capabilities, various kinds of data processing capabilities, data protection support and data management logic, as well as support for cloud/HPC execution. In addition, the App Entities may include support for Cybersecurity and Trust.

The IoT Entities of the AIOTI HLA may include access and management capabilities for sensors and actuators, but also support for data analytics (edge analytics), data processing, data protection and data management. In addition, the IoT Entities may include support for Cybersecurity and Trust.

The Networks of the AIOTI HLA are linked to the Communication and Connectivity area of the BDV Reference Model. In particular, they support short-range and long-range connectivity and data forwarding between entities, and both synchronous and asynchronous communication mechanisms, with appropriate QoS support. The Networks also include support for IoT devices' communication and connectivity. In addition, they may include support for Cybersecurity and Trust.

NOTE 5 - The BDV Reference Model areas of, respectively, "Data Sharing platforms, Industrial/Personal", "Development, Engineering and DevOps" and "Standards" are not mapped to the AIOTI HLA in the above figure. The first area might be relevant for IoT data management, the second area might be relevant for the total life cycle of IoT data, the third area is relevant for all areas (layers).

A corresponding mapping of the AIOTI HLA (entities) to the BDV Reference Model (technical areas and capabilities) is shown in Figure 8-11.

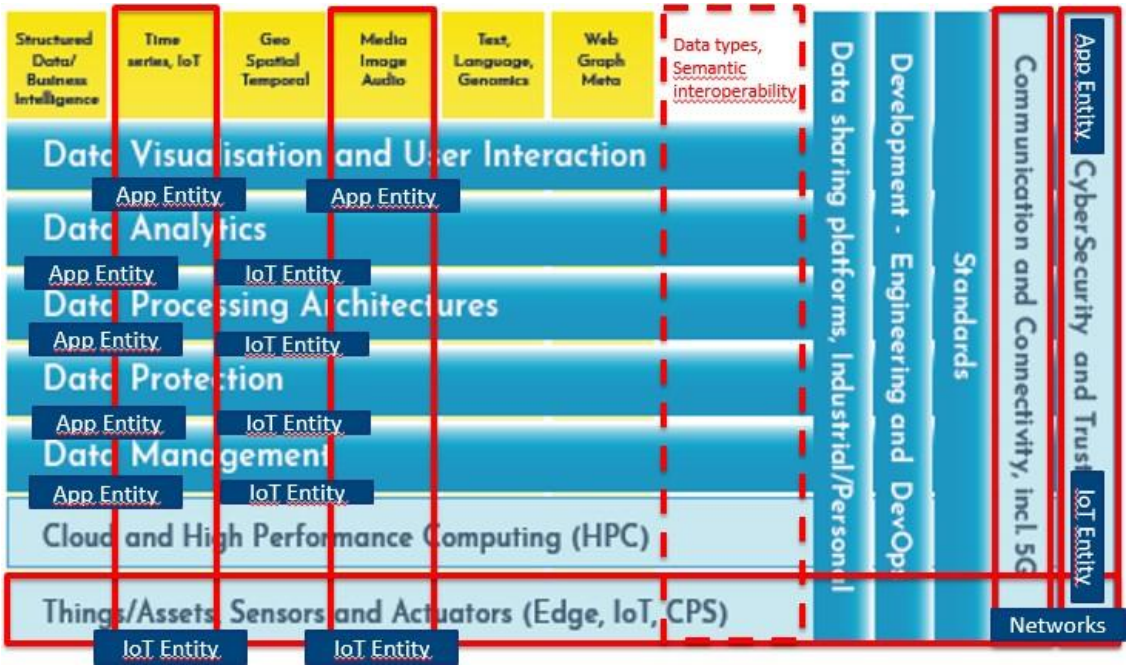


Figure 8-11 - AIOTI HLA mapping to the BDV Reference Model

8.6 3D IoT Layered Architecture

NOTE – The mapping of the 3D IoT Layered Architecture to the AIOTI HLA functional model

is not specifically discussed in this Release of the document. Nevertheless, an obvious consideration is that only the “Layers” view of the 3D IoT Layered Architecture applies for the mapping to the AIOTI HLA functional model.

The 3D IoT Layered Architecture (aka the 3D model), specified in [40] and [41], is an approach to define, identify and co-relate multiple IoT system features, architectural characteristics and properties in Large Scale pilot (LSP) IoT systems, see Figure 8-12.

The principle of this Reference Architecture is to use a number of 2D views that are a projection of the 3D view on a specific plane. In particular, a preliminary analysis of how the stakeholders are involved in the definition of an IoT system can be aligned by using each of the three main views analyzed and shortly described in this clause.

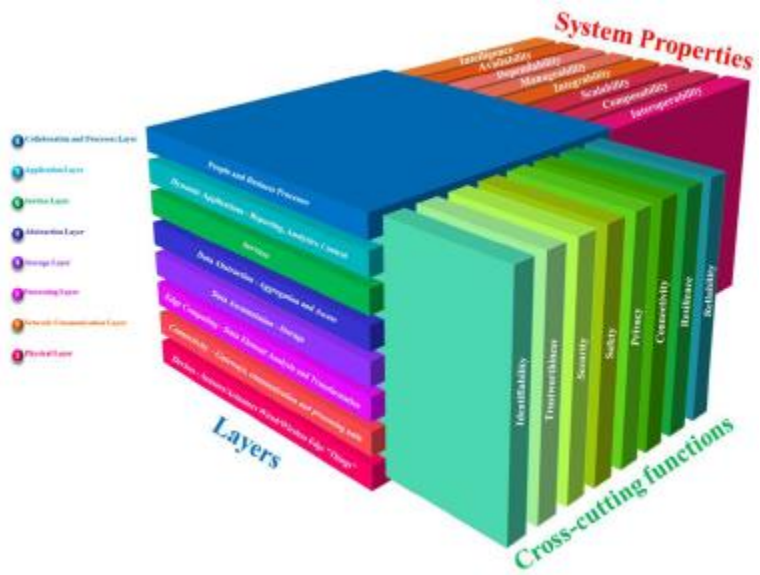


Figure 8-12: The three main views in the 3D Model (Layers, Cross-cutting functions, and Properties) [41]

The “Layers” view in the 3D model, see Figure 8-13, refers to the overall characteristics of IoT Systems from a functional and operational perspective. It includes aspects from physical devices, networking, cloud infrastructures, data, services and applications but also collaboration. The main usage of this view is to facilitate the identification of necessary functional blocks for interoperability at the different “layers” in IoT systems.

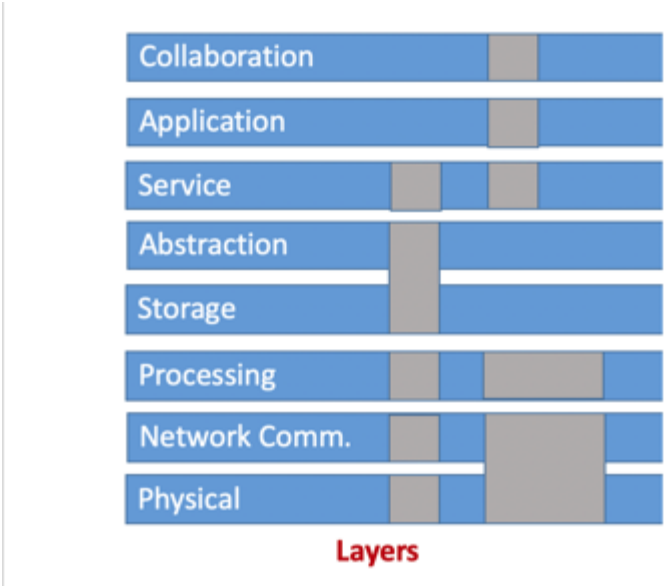


Figure 8-13: The Layers view in the 3D Model (Layers, Cross-cutting functions, and Properties) [41]

The “Cross-cutting Functions” view, see Figure 8-14, refers to properties of the IoT system which are not resulting from just functional components but more from the interactions amongst these components. It includes security, safety & resilience, trust and privacy, connectivity, interoperability, dynamic composition and automated interoperability. The main usage of this view is to support the protected and reliable exchange of information.

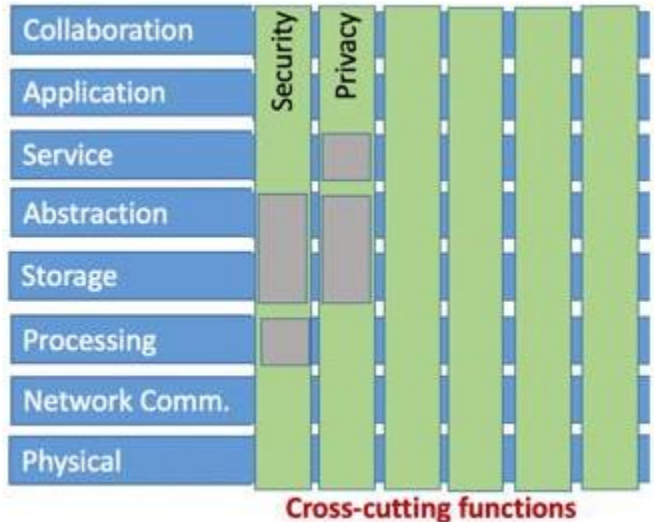


Figure 8-14: The Cross-cutting Functions in the 3D Model (Layers, Cross-cutting functions, and Properties) [41]

The “Properties” view, see Figure 8-15, refers to features and characteristics of the IoT systems that are not associated with the data but with the administrative and managing aspects of the IoT infrastructure and the system itself. It includes Intelligence, Availability, dependability, manageability, integrity, scalability composability and Interoperability. The main usage of this view is for identification of the properties characterizing IoT systems or applications.

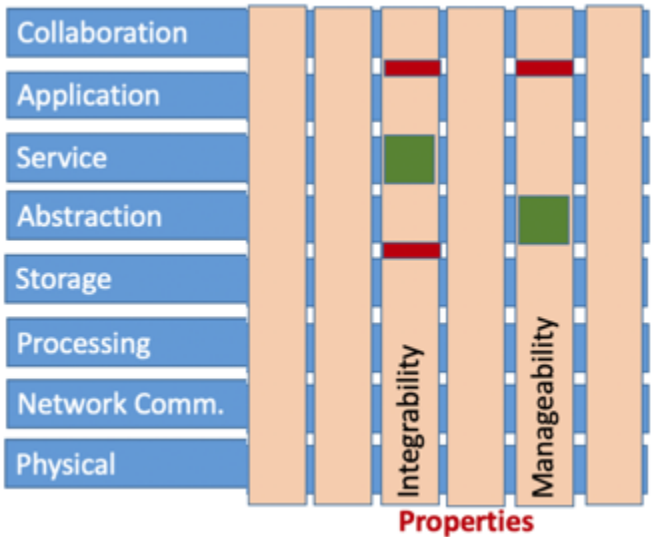


Figure 8-15: The Properties view in the 3D Model (Layers, Cross-cutting functions, and Properties) [41]

9. Relationship to other functional models or systems

9.1 Introduction

This clause provides relationship between the AIOTI functional model and other functional models. While the AIOTI HLA functional model depicts interfaces within the IoT system, other external interfaces are extremely important to study for the purpose of operational deployments at large scale. Figure 9-1 shows in particular interactions with Big Data frameworks and other service platforms (banking, maps, open data, etc.).

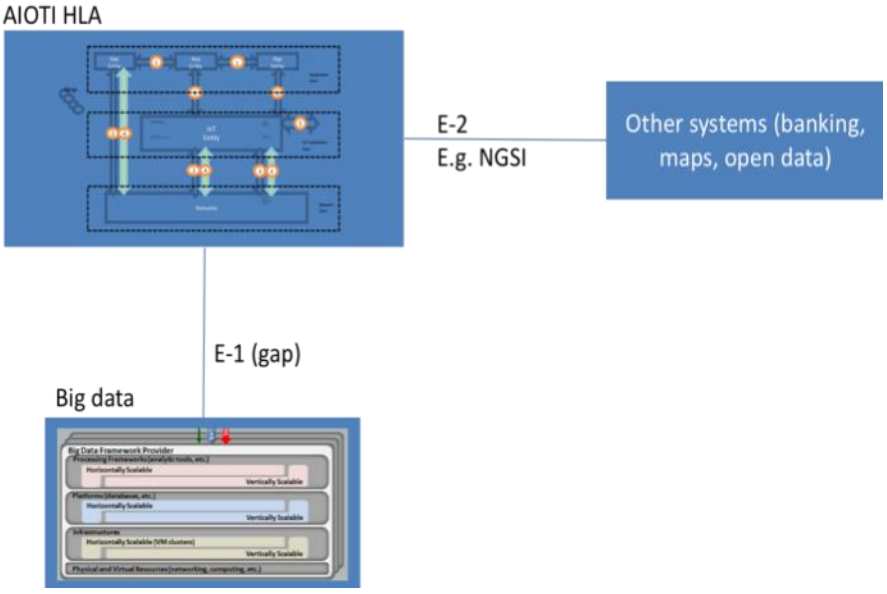


Figure 9-1: Relationship to other systems

Figure 9-1 show in particular two interfaces:

- E-1: used to integrate with big data architectures, e.g. as documented by NIST in [2].
- E-2: used to exchange context information with other service platforms: location, maps, banking, etc. In the context of Fiware, interface E-2 is implemented using APIs based on the OMA NGSI protocol.

9.2 Framework of IoT-Big Data integrated architecture

NOTE- This topic is for further development in following Release(s) of this document.

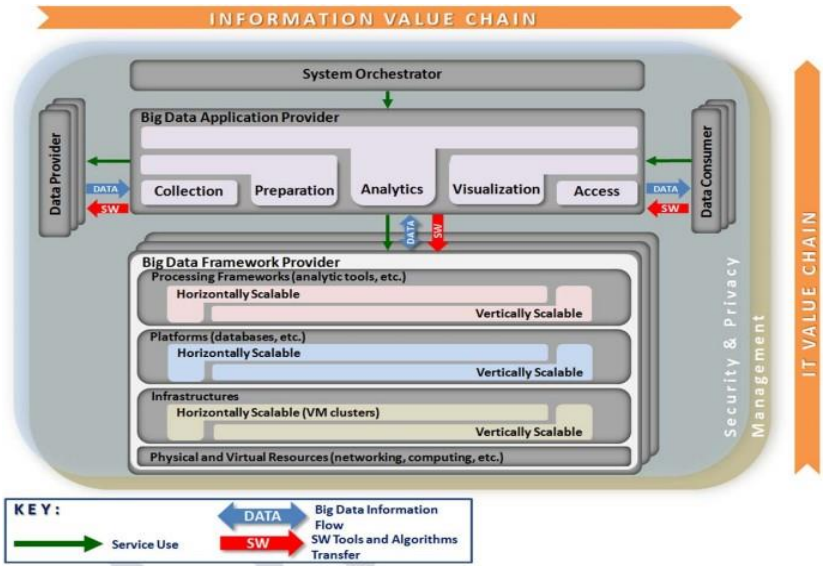
9.2.1 Approach for IoT-Big Data integration

NOTE- Topic for study in following Release(s) of this document.

9.2.2 Relationship to NIST Big Data framework

The NIST Big Data interoperability framework has been described to a great extent in the following document [2]. Of particular interest to the scope of this deliverable is the NIST Big Data Reference architecture which is depicted in Figure 9-2.

Figure 9-2: NIST Big Data reference architecture



When considering the relationship between AIOTI HLA functional model and the NIST Big Data reference architecture, it is possible to consider a Data Provider as a HLA App Entity running in a Device or Gateway. The Big Data Application Provider could be an HLA IoT Entity or an App Entity running in a cloud server infrastructure, e.g. performing data aggregation. Finally, a Data Consumer could be an App Entity running in a Utility back-end server. Figure 9-3 depicts this mapping example.

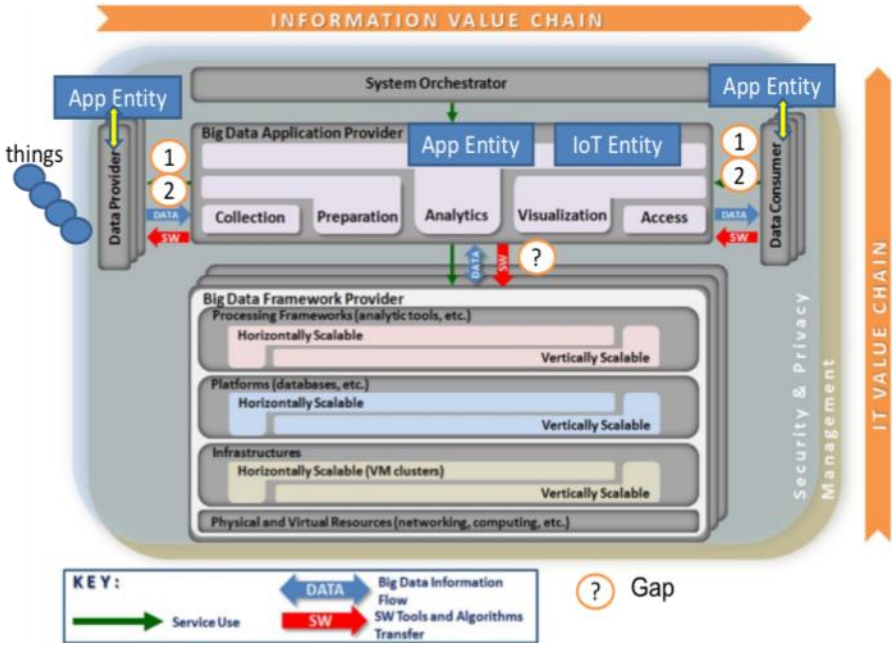


Figure 9-3: Mapping of AIOTI functional model entities to NIST big data reference architecture

In Figure 9-3 the interface depicted with ("?") to a Big Data Framework Provider could be important in Large Scale Deployments of AIOTI. Further study is needed to figure-out current standardization developments related to this interface. A standardized interface may provide market benefits and remove dependency on a particular provider for the Big Data framework.

9.3 IoT-enabled Data Marketplaces

9.3.1 High-level architecture of an IoT-enabled Data Marketplace

Figure 9-4 provides a possible high-level architecture for an IoT-enabled Data Marketplace [42].

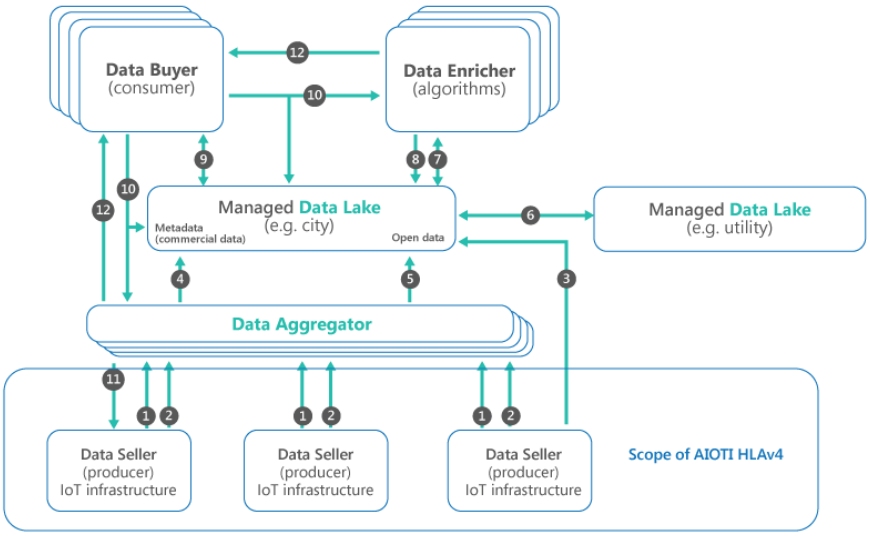


Figure 9-4: A possible high-level architecture for an IoT-enabled Data Marketplace

This reference architecture includes functions that could be mapped to different stakeholders, and multiple functions can be implemented by the same administrative stakeholder in a given operational deployment.

- Data Sellers are entities that deploy an IoT infrastructure, for example smart energy meters. These entities are interested in selling the collected data or subsets of that data. This sale must be in accordance with privacy regulations and data owners' consent. A Data Seller would typically publish both commercial data (1) and open data (2) using a Data Aggregator. Alternatively, the open data may be contributed directly to a Managed Data Lake (3).

- Data Aggregators are programmed to aggregate mostly 'dumb' data streams from different sources, merging these data streams to create more valuable sources of information. A Data Aggregator would typically contribute both open data (5) and metadata pertaining to commercial data sets (4) to a Managed Data Lake. Metadata would provide a semantic description of the data as well as the terms of contractual agreements governing data transactions. The Data Aggregator would be responsible for transacting data on behalf of data producers in exchange for a portion of associated revenue streams.

- Managed Data Lakes[1] would typically store a massive amount of data and metadata to enable data discovery, as shown in arrows (7) and (9). This reference architecture assumes that a Managed Data Lake does not store commercial data. Following a Data Buyer's discovery of data of interest to them, that Data Buyer would subscribe to an automated smart contract (10) for the agreement and immediate pay-out of the Data Seller's expected price (11). In other scenarios it would remain possible for the Data Seller to receive a revenue stream in a periodic manner, for example once a month. The provider responsible for the Managed Data Lake would automatically receive a commission on every transaction facilitated, a key requirement for the financial sustainability of the data lake.

    1. After the settlement of the payment, the actual data would be exchanged peer-to-peer (12) between a Data Buyer and Data Aggregator.
    2. A Managed Data Lake could also contain mirrors of metadata from other lakes. The mirroring process is shown in (6).

- Data Enrichers are entities buying commercial data or consuming open data (7) with the intention of applying algorithms to enrich data and resell new data sets as a value-added service, typically to provide analytics yielding new insights and predictions. A Data Enricher would contribute its metadata back to a Managed Data Lake (8).

- Data Buyers consuming data streams or downloading data sets (12) are interested in the additional value that external data can bring to their internal data.

[1] Data lakes have been covered in this blog: https://news.itu.int/what-will-keep-smart-cities-busy-2019/

9.3.2 Fundamental concepts for successful deployment of an IoT-enabled Data Marketplace

Certain concepts are fundamental to the successful deployment of IoT-enabled Data Marketplaces adopting the high-level architecture shown in Figure 9-4.

- Metadata provide descriptions of the data assets up for sale by different stakeholders as well as the methods to transact in these assets. It is important that data sellers and buyers share a common understanding of what the data is about. Reaching this common understanding would only be possible with a standard or agreed ontology. NOTE - ITU-T SG20 and Open Geospatial Consortium could be the two initiatives to consider this standards gap.
- Mirroring metadata is the concept of exposing metadata in a third-party data lake. This mechanism allows for cross-domain data discoverability.
- Cross-domain data discoverability facilitates the distributed, collaborative development of data-driven solutions in line, for example, with the principles put forward by the EU Digital Single Market.
- Blockchain and distributed ledger technologies provide means to build trust into every transaction without the need for central authorities. They are capable of enabling micropayments without transaction fees. They are also valuable in providing proof-of-origin for data sets as well as proof-of-integrity for data lakes.
- Decentralized, yet federated: the shown reference architecture describes a data economy without need for a central entity or centralized powers, which could offer a foundation for a fair distribution of revenue streams. The federation is achieved through the mirroring process.
- Governance presents some of the most complex problems in this space. It is difficult to define sustainable governance models for new technology solutions when new models appear continuously and the oldest model is only a few years' old. The governance challenge is two-fold:
    - Keeping up with evolving models and technologies, such as blockchain and distributed ledger technologies, including "their potential to transform and even reinvigorate the governance of cities[2];
    - Ensuring a fair distribution of revenue streams and avoiding the creation of new monopolies.

2. Sarah Barnes, Smart cities and urban data platforms: Designing interfaces for smart governance. City, Culture and Society

9.3.3 The example of a Mobility Data Marketplace [47]

Smart mobility is reaching an inflection point driven by two market developments:

A. electric mobility which is finally entering the mainstream and

B. car (and infrastructure) connectivity being leveraged beyond its originally intended uses such as infotainment and optimized navigation.

Connectivity, combined with advances in sensor technology, are driving a paradigm shift towards a crowdsourcing data driven smart mobility through the deployment of new services for energy efficiency, usage-based insurance, parking, retail, maintenance, etc. Electric mobility enlarges the plethora of possible applications through opening-up the set of possible use cases to a wealth of cross domain ones with deep impact on the energy sector which is facing the challenges of ensuring resiliency and maximizing the use of renewables.

The discussion is not anymore about the need for mobility data marketplaces or not. The discussion is more about:

- how will it happen?
- what are the remaining technology and governance gaps to be addressed before reaching wide scale deployments?
- what synergies will it have with smart cities, smart energy marketplaces, etc.

Concerning the applications and cross-cutting use cases driving the need for data marketplaces, similar to the Internet development, it's not feasible to predict the future applications or use cases that innovators will come-up with, as long as the infrastructure is built in a user-centric, components reuse and fair sharing of revenue streams in mind.

Today, some pilot use cases are explored to accelerate deployment of EV charging points related to housing companies, smart mobility stations, private parking space providers (like railway stations, retail, commercial parking space operators, etc.) smart districts, smart lighting, etc.

Use cases include examples where a car can become an energy resource to allow a train station for instance to become resilient against sudden disruption in the electricity grid. Other use cases relate to maximising the use of renewables and trading flexibility with the energy providers who need to shape demand during peak hours.

9.3.3.1 Actors of a Mobility Data Marketplace

When it comes to smart and electric mobility, the actors could be described as follows:

- Data Sellers: they include automotive OEMs, mobility and fleet management service providers, charge point operators, power suppliers, energy grid operators, etc.
- Data Buyers: they include potentially all of the above players in addition to (entrepreneur) application developers, home and building energy management service providers, etc. The data buyers will typically use processed and context enriched data to provide value to end users and generate new revenue streams. Examples of new revenue streams include trading flexibility to energy providers, prediction of the formation of potholes and the whole area of user enriched mapping.
- Data Marketplace: similar to digital marketplaces, data marketplaces connect together data producers and data consumers with different options for financial settlements and the range of value-added services provided. Data marketplace providers will typically incentivize the data producers to continue producing quality data of interest to data buyers.

9.3.3.2 Possible business models for a Mobility Data Marketplace

The followings are possible business models for a Mobility Data Marketplace:

- Neutral host: assumes that a neutral entity, that is not specifically owned by any of the data producers or consumers is responsible for collecting the data, sharing the data and managing the data lifecycle according to user consent and applicable regulations. The ownership of the neutral host service provider could be a joint venture between stakeholders including, OEMs, cities, transportation, etc. This model may speak in favor of coopetition (cooperating and competing at the same time) which is a key for the success of a data driven mobility.
- Federated data marketplace: assumes multiple data marketplaces share and mirror metadata (information about data) allowing any user to discover data-sets stored in third parties' marketplaces and eventually acquiring them without being directly affiliated with that market place.
- Hybrid data marketplace: assumes both of the previous models where for example a neutral host could be implemented for mobility while a federation approach would allow to onboard data sets from energy and smart cities data marketplaces.

9.3.4 Market inhibitors and technology gaps of a Mobility Data Marketplace

Figure 9-5 illustrates market inhibitors of a Mobility Data Marketplace.

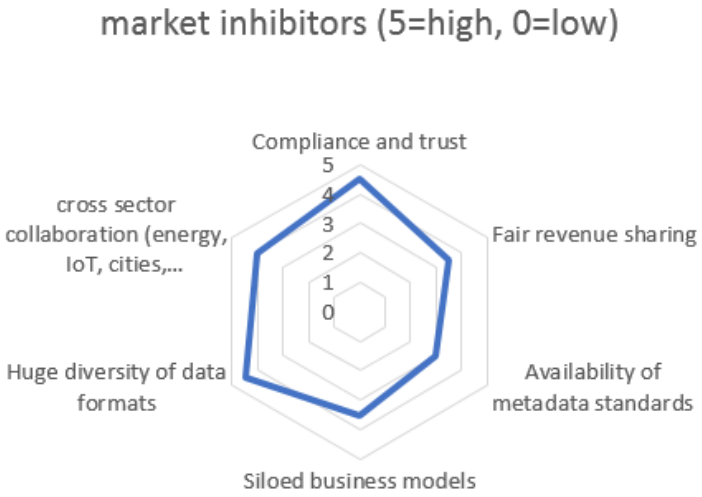


Figure 9-5: Market inhibitors of a Mobility Data Marketplace

- Compliance and trust: as we build cross domain applications, privacy protection for both personal and non-personal data becomes very challenging at the technical level. Several solutions have been explored at length by academia, but their wide scale implementation did not enter the mainstream yet. Users have also lost some trust in service providers but the situation is changing since GDPR entered into force.

- Cross sector collaboration: energy, ICT, IoT and smart cities have traditionally focused on their own needs without paying much attention to cross sector collaboration. Building successful marketplaces supporting the EU digital single market will need increased collaboration because eventually a big proportion of use cases will be cross sectors.

- Diversity of data formats: different data formats have proven to prevent cost efficient integration at scale. All vendors claim to have RESTful API, but their own. The market needs to solve data interoperability issues through a limited number of APIs and data models. Eventually when more experience is built, regulation can help in order to reduce the number of possible options.

- Siloed business models: Working in isolation, the mobility sector may not be capable of transforming mobility and bringing new services to consumers, the same applies to the energy, smart cities, etc. This transformation will call for all the sectors to cooperate and compete at the same time (coopetition). Interacting and learning from experiences of successful cross-sector marketplaces, creating interfaces with other marketplaces and collaborate extensively with technology providers and connectivity providers will be essential to move beyond a siloed approach.

- Availability of metadata standards: data proliferation argues for the need of metadata, an approach to describe what the data is about and what it could be used for. The buyer must A. have the means to discover accurately data and B. understand its value and intended use. This is the role of metadata standards.

- Fair revenue sharing: building data marketplaces would need creating the conditions for fair revenue sharing models and avoiding new monopolies. As we build operational experience with data marketplaces, this aspect needs particular attention from a governance and policy making perspective.

9.4 Relationship to other service platforms

Editor's note: the implementation of the platform interoperability approaches described in 7.7.3.2 could be considered in this clause (reference points/interfaces, stakeholders, …). Also, use cases developed by LSPs could be useful for specific implementation guidelines on platform-to-platform interoperability related to different IoT sectors.

Figure 9-6 shows the interface E-2 to other service platforms. Interface E-2 is a multipoint interface that allows to connect the IoT Entity to other service platforms such as a maps server. The rationale for E-2 is the need to provide integration of IoT data with other non IoT data. Typically, E-2 consists of a publish/subscribe based protocol such as MQTT or OMA NGSI. The FIWARE project suggests the use of APIs specified on top of the OMA NGSI protocol for the E-2 interface.

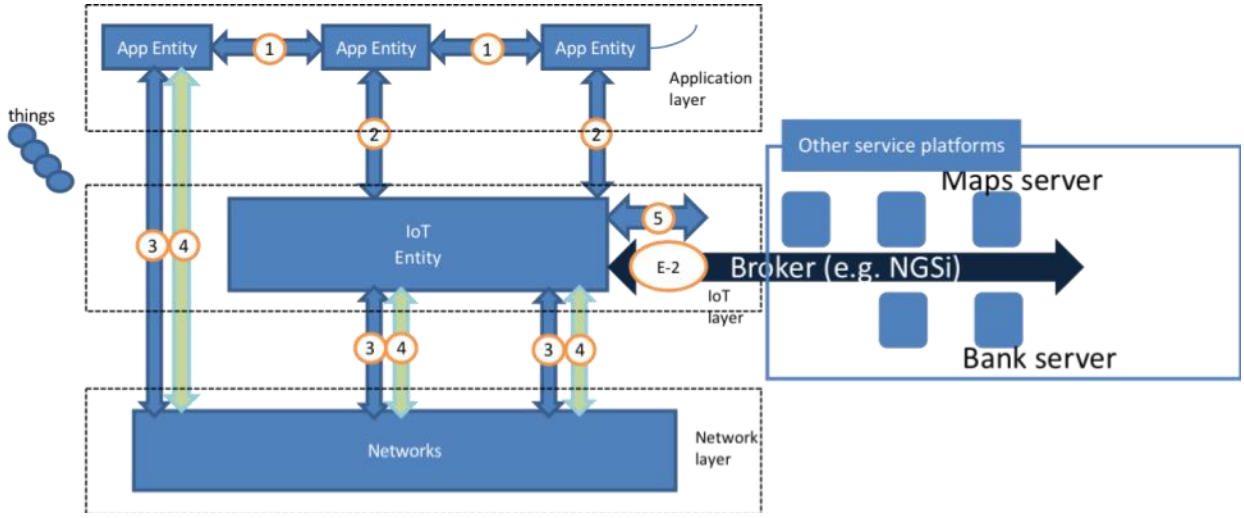


Figure 9-6: E-2 interface illustration

Figure 9-7 provides an example of message flow using the E-2 interface. In this example two kinds of interactions on the E-2 interface are depicted. The first interaction is query based where the IoT Entity query the information from the Broker functionality. In the second interaction, the IoT Entity subscribes for a specific event and gets notifications when the event occurs.

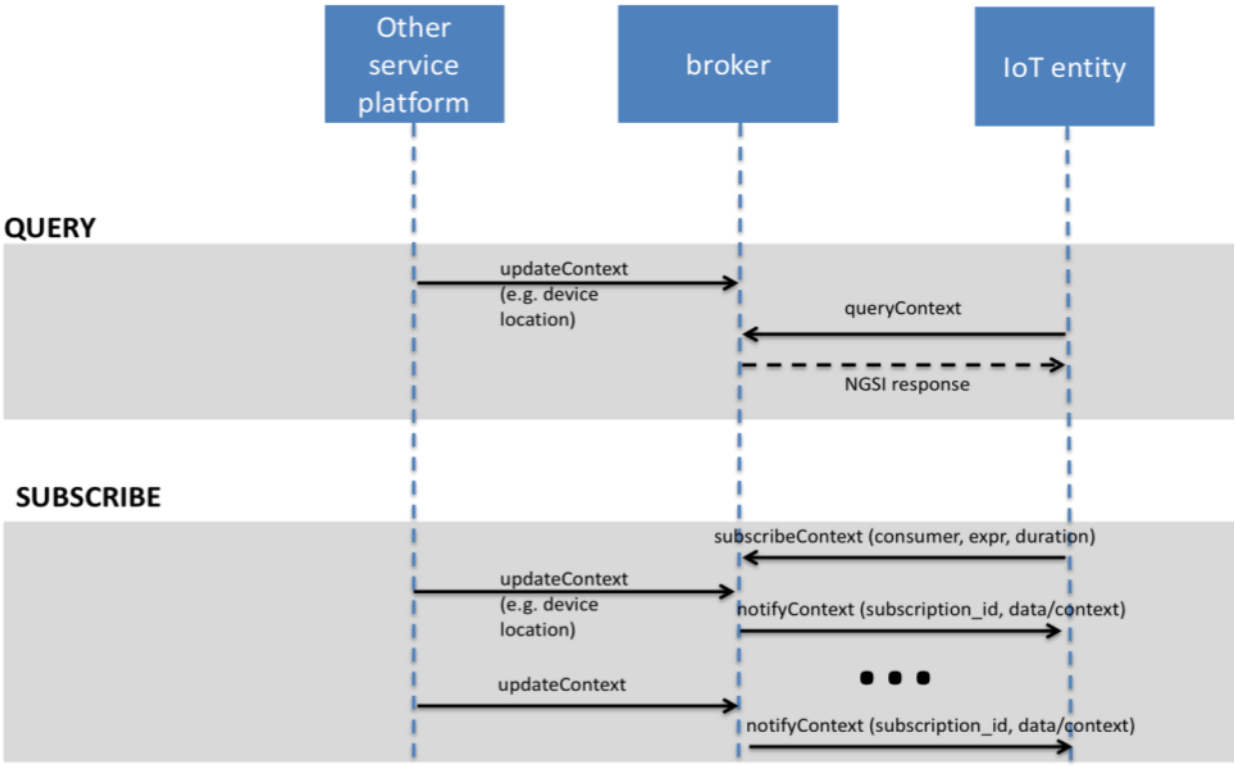


Figure 9-7: Example of message flow illustrating the E-2 interface

10. Artificial Intelligence for IoT

NOTE- Topic for study in following Release(s) of this document.

Annex I Additional mappings

Annex I-1 Mapping to ETSI SmartBAN

ETSI SmartBAN technical committee addresses all aspects related to BANs (Body Area Networks). These include:

- aspects and operations related to BANs from lower layers up to service and application layer
- aspects related to heterogeneity/interoperability management, including syntactic and semantic interoperability

ETSI SmartBAN currently addresses verticals that are related to eHealth, wellbeing/wellness and personal safety. Figure I.1 shows the scope of ETSI SmartBAN.

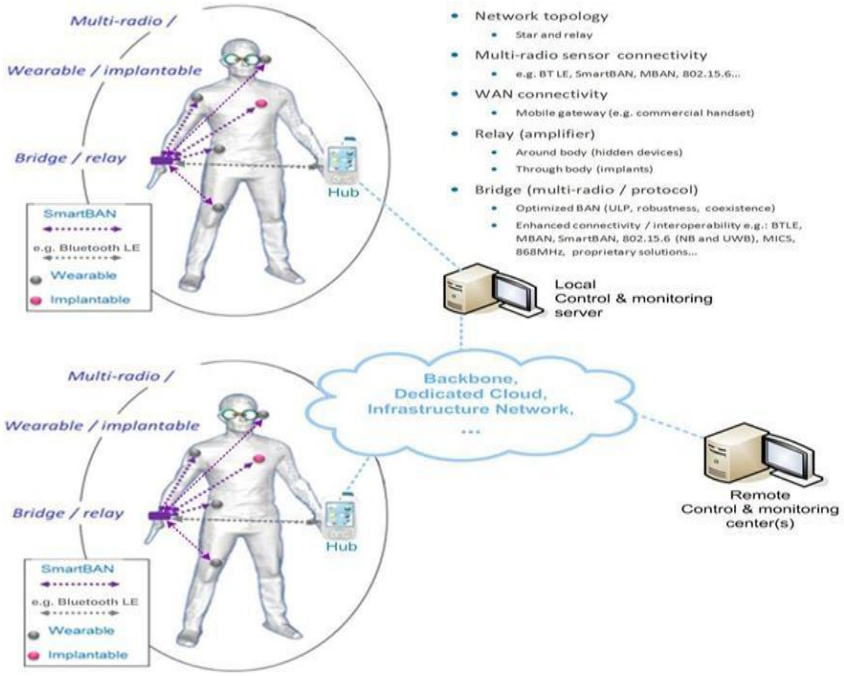


Figure I-1: ETSI SmartBAN deployment example concepts

ETSI DTR/SmartBAN-004 reference architecture provides a layered reference architecture for SmartBAN. The reference architecture is depicted in the following figure I.2 which shows a layered approach with an Application Layer, a Service Layer, a Semantic Layer and a Data provision layer.

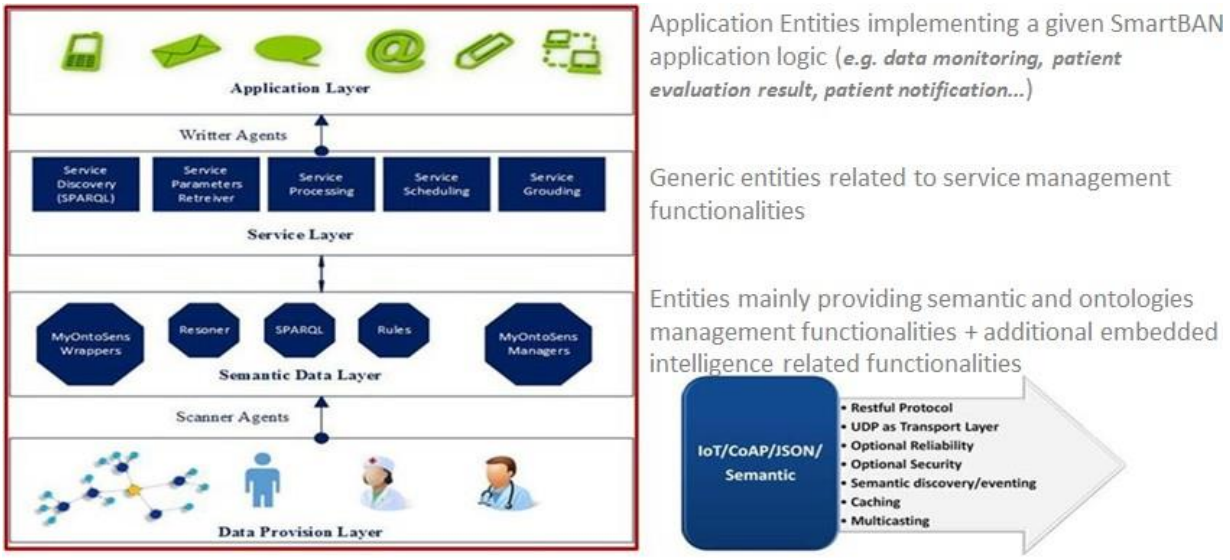


Figure I-2: ETSI SmartBAN reference architecture

Key observations about this reference architecture include:

- A distributed multi-agent based IoT architecture for both:
  - allowing generic and secure interaction/access to any BAN data/entities,
  - providing a unified IoT platform for BAN distributed monitoring and control operations.
- The architecture is semantic enabled. It relies on ETSI SmartBAN data/service model and corresponding ontologies (ETSI DTS/SmartBAN-009 and DTS/SmartBAN-009r1 standards).

The following figure I-3 provides a binding between the ETSI SmartBAN architecture and the AIOTI HLA:

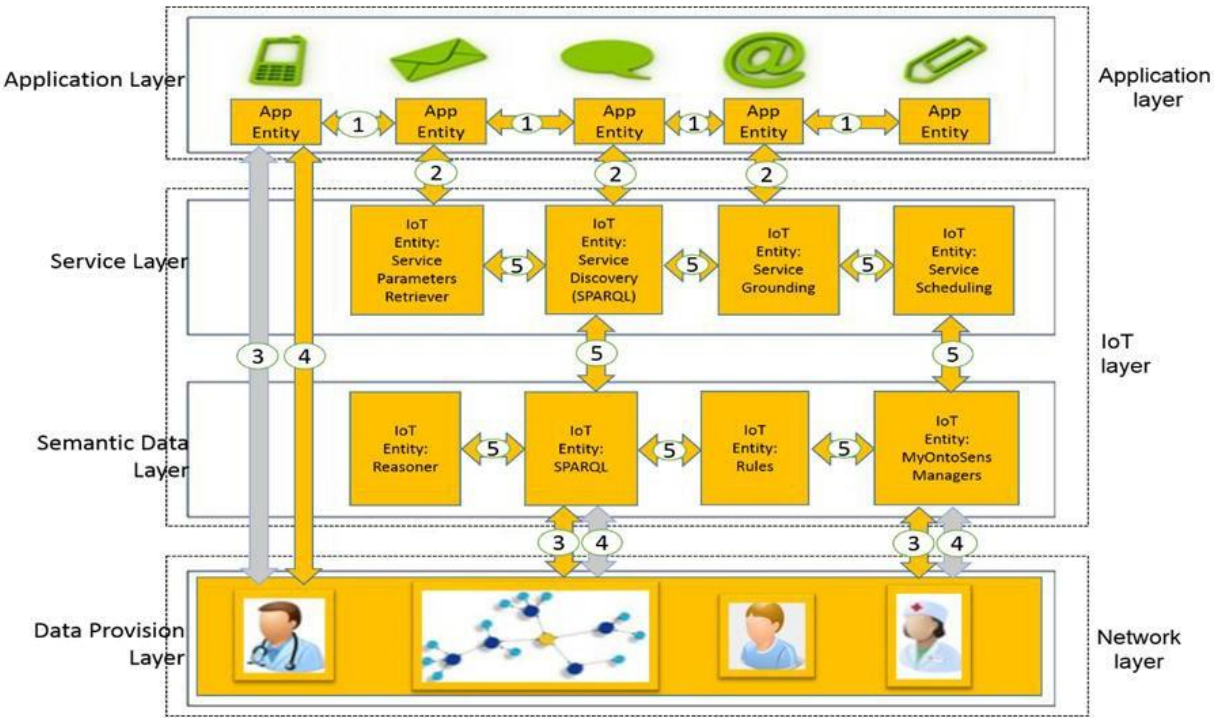


Figure I-3: ETSI SmartBAN reference architecture mapping to AIOTI HLA

In this figure we can see:

- Direct mapping between ETSI SmartBAN and AIOTI application layers is provided
- Each entity of ETSI SmartBAN Service and Semantic Data layers can fully be considered as an IoT entity and thus is considered to be a part of the AIOTI HLA IoT Layer,
- SmartBAN Data Provision Layer and IoT Network Layer have exactly the same role (direct mapping).

Annex II IoT standards gaps and relationship to HLA

Editor's note: this Annex should be enhanced in order to align with the HLA related progress of the AIOTI WG3 "Gaps" Task Force.

The work of standardisation never stops whichever domain is concerned, IoT being no different. At any moment, new issues arise that cannot be dealt with given the current status of (in particular technical) standardisation. The emergence of these gaps, and the initiatives taken for their resolution, define the evolution of the roadmap of standards development organisations.

In October 2016, ETSI has published a report [13] aiming at the identification of gaps related to IoT. Those gaps were in three categories: technical, business and societal (the latter category including security or privacy). Amongst those gaps, a certain number can be mapped on the AIOTI HLA, thus showing where the problems arise and where – in the IoT standardisation landscape - their resolution can be anticipated.

Those gaps are listed in Table II-1 below that lists a certain number of gaps and a tentative identification of the areas of the AIOTI HLA Functional model where their impact is most visible.

Gap	Impact
Competing communications and networking technologies	Network layer
Easy standard translation mechanisms for data interoperability	IoT and application layers
Standards to interpret the sensor data in an identical manner across heterogeneous platforms	IoT layer
APIs to support application portability among devices/terminals	IoT layer
Fragmentation due to competitive platforms	Not specific to HLA
Tools to enable ease of installation, configuration, maintenance, operation of devices, technologies, and platforms	Mostly IoT layer, also Appl. and Network
Easy accessibility and usage to a large non-technical public	Not specific to HLA
Standardized methods to distribute software components to devices across a network	IoT and network layers
Unified model/tools for deployment and management of large scale distributed networks of devices	All layers; critical in IoT layer
Global reference for unique and secured naming mechanisms	All layers
Multiplicity of IoT HLAs, platforms and discovery mechanisms	Addressed by HLA

Certification mechanisms defining “classes of devices”	Network layer
Data rights management (ownership, storage, sharing, selling, etc.)	All layers
Risk Management Framework and Methodology	All layers; interface definition

Table II-1: IoT Gaps mapped on the AIOTI HLA

Annex III Advantages and disadvantages of end device, edge and cloud computing

Table III-2 below lists some advantages/disadvantages of end device, edge and cloud computing options.

Topic	End device computing	Edge computing	Cloud computing
Real time/low latency processing (e.g. time constrained control loops, synchronous operation)	+ Minimizes communication delays for local sensors and actors. However limited computing resources could delay complex algorithms and all involved sensors and actors may not be part of the same end device	+ Low communication delay. Could be placed in best distance to all involved components	– High communication delay. Shared computing platform is often not real time capable
Network bandwidth and availability	+ No network needed. Local data pre-processing reduces upstream bandwidth needs	+ Local data pre-processing reduces upstream bandwidth needs	– Always requires network connectivity. Bandwidth demands could be high depending on application
Computing & storage resources	– Low resource footprint of some devices puts limitations on processing capabilities	– + Resources could be scaled more flexibly to processing needs, but still has limitations	+ Abundant resources that can be scaled to all processing needs
Offline capabilities (e.g. emergency operation)	+ Works without network as long interaction with remote components is not needed	+ – Requires only local network connectivity	– Requires always network connectivity
Energy consumption/ carbon footprint	– Local processing increase energy usage which is critical for battery powered end devices and devices that do energy harvesting. No sharing of infrastructure is possible.	+ – Can reduce overall power consumption by using otherwise lightly loaded CPU resources in existing edge devices (e.g. routes, base stations) and sharing that infrastructure between several applications. However sharing capabilities might	+ – Use of latest energy efficient technologies and optimized use of shared infrastructure optimizes use of energy resources. Bringing all data to the cloud without local processing however increase network utilization and power

		be limited.	consumption
Costs	+ –	+	+

	Dedicated investment in end devices needed. However Sensors and actors are needed anyway.	No investment in additional resources needed if existing infrastructure can be reused and shared (gateways, base stations).	No need to invest in dedicated computing infrastructure (capex and opex).
Deployment flexibility	– Deployment of new functionality may require HW update	+ – Provides some flexibility for deployment of new applications, but with limitations	+ Provides highest flexibility in application deployment
Device/service reliability/availability	– Usually no redundancy available	– + Only limited redundancy	+ Managed service platforms provide high availability
Management	– Remote Management needed. Might be limited due to device and network constrains	+ – Remote management needed	+ Central management of resources. Infrastructure managed by service provider
Big Data	– Processing usually limited to data of the device itself	+ – Can process data from sources in the surrounding, but that may provide only a limited view on the overall data	+ Can process and store large amounts of data from various sources.
Backup & Recovery	– No or limited local backup. Remote backup might be limited due to device and network constrains	+ – Local and remote backup approach	+ Backup & recovery is integral part of cloud offerings

Table III-2: Advantages and disadvantages of end device, edge and cloud computing

Annex IV References

[1] IoT-A project: http://www.meet-iot.eu/iot-a-deliverables.html

[2] NIST big data interoperability framework: http://bigdatawg.nist.gov/V1_output_docs.php

[3] Recommendation ITU-T Y.4000 (ex-Y.2060) "Overview of the Internet of Things": https://www.itu.int/rec/T-REC-Y.4000/en, 2012

[4] oneM2M Functional Architecture Release 1, http://www.etsi.org/deliver/etsi_ts/118100_118199/118101/01.00.00_60/ts_118101v010000 0p.pdf

[5] Industrial Internet Reference Architecture, http://www.iiconsortium.org/IIRA.htm

[6] AIOTI WG03 deliverable on Semantic Interoperability

[7] Recommendation ITU-T 3600 (2015), Big data – Cloud computing based requirements and capabilities: http://www.itu.int/rec/T-REC-Y.3600-201511-I

[8] Recommendation ITU-T Y.4114 (2017), Specific requirements and capabilities of the Internet of Things for Big Data: https://www.itu.int/rec/T-REC-Y.4114-201707-I

[9] 3GPP TR 23.799, "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Study on Architecture for Next Generation System", 3GPP TR 23.799, V14.0.0, Release 14, December 2016 (https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3008)

[10] NGMN Alliance, "Description of Network Slicing Concept", Version 1.0, January 2016, http://www.ngmn.org/uploads/media/160113_Network_Slicing_v1_0.pdf

[11] ETSI ISG NFV, "Network Functions Virtualisation White paper on NFV Priorities for 5G", ETSI ISG NFV, Issue 1, February 2017, http://portal.etsi.org/NFV/NFV_White_Paper_5G.pdf

[12] ETSI GS MEC 003 Mobile Edge Computing (MEC); Framework and Reference Architecture, ETSI GS MEC 003 V1.1.1 (2016-03), March 2016, http://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p

.pdf

[13] ETSI Smart M2M, "IoT LSP use cases and standards gaps", TR 103 376, V1.1.1 (2016-10) http://www.etsi.org/deliver/etsi_tr/103300_103399/103376/01.01.01_60/tr_103376v010101p.pdf

[14] Motivation Challenges Opportunities in Edge Computing https://www.researchgate.net/publication/307888414_Motivation_Challenges_Opportuniti es_in_Edge_Computing

[15] OpenFog Whitepaper, February 2016, https://www.openfogconsortium.org/white-paper- reference-architecture/white-paper-download-open-fog-reference-architecture/

[16] VDI/VDE GMA, ZVEI: Status Report - Reference Architecture Model Industrie 4.0 (RAMI 4.0) , July 2015, https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/janu ar/GMA_Status_Report___Reference_Archtitecture_Model_Industrie_4.0 _____RAMI_4.0_/GMA

-Status-Report-RAMI-40-July-2015.pdf

[17] DIN SPEC 91345:2016-04 – Referenz architektur modell Industrie 4.0 (RAMI 4.0), April 2016, http://www.din.de/de/ueber-normen-und-standards/din-spec/din-spec-veroeffentlichungen/wdc-beuth:din21:250940128

[18] IEC PAS 63088:2017 Smart manufacturing - Reference architecture model industry 4.0 (RAMI 4.0), March 2017, https://webstore.iec.ch/publication/30082

[19] IEC 62264-1:2013 Enterprise-control system integration - Part 1: Models and terminology, May 2013, https://webstore.iec.ch/publication/6675

[20] AIOTI WG03, „Identifiers in Internet of Things (IoT)“, Version 1.0, February 2017, https://aioti.eu/wp-content/uploads/2018/03/AIOTI-Identifiers_in_IoT-1_0.pdf.pdf [Accessed 10.04.2018]

[21] "Virtualized IoT Architectures with Cloud Back-ends", ETSI TR 103 527, 2018.

[22] "Landscape for open source and standards for cloud native software for a Virtualized IoT service layer ", ETSI TR 103 528, 2018.

[23] "Network Functions Virtualisation (NFV): Use Cases", ETSI GS NFV 001, 2013

[24] "Network Functions Virtualisation (NFV): Architectural Framework", ETSI GS NFV 002, 2014

[25] "Network Functions Virtualisation (NFV): Infrastructure Overview", ETSI GS NFV-INF 001, 2014

[26] "oneM2M Functional Architecture Baseline Draft", oneM2M-TS-0001, 2014

[27] GSMA Association Official Document CLP.25, ["IoT Big Data Framework Architecture", Version 1.0, 20 October 2016, https://www.gsma.com/iot/wp-content/uploads/2016/11/CLP.25-v1.0.pdf [Accessed 25.05.2018]

[28] TMForum, Data Analytics, https://www.tmforum.org/data-analytics/ [Accessed 25.05.2018]

[29] ITU-T FG-DPM, ITU-T Focus Group on Data Processing and Management to support IoT and Smart Cities & Communities, https://www.itu.int/en/ITU-T/focusgroups/dpm/Pages/default.aspx

[30] Big Data Value Association, http://www.bdva.eu/

[31] Big Data Value Association, European Big Data Value Strategic Research and Innovation Agenda, http://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf

[32] ISO/IEC JTC1/SC42 Artificial Intelligence, https://www.iso.org/committee/6794475.html

[33] iCore, www.iot-icore.eu

[34] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)

[35] Alessandro Bassi, Martin Bauer, Martin Fiedler, Thorsten Kramp, Rob van Kranenburg, Sebastian Lange, Stefan Meissner (Editors), Enabling Things to Talk -- Designing IoT solutions with the IoT Architectural Reference Model, Springer, 2013

[36] Chayan Sarkar; Nambi S. N., Akshay, et al., "DIAT: A Scalable Distributed Architecture for IoT", IEEE Internet of Things Journal, DOI 10.1109/JIOT.2014.2387155, pp.1-8, 2014, Preprint.

[37] M. Djurica, G. Romano, G. Karagiannis, Y. Lassoued, G. Solmaz, "oneM2M-Based, Open, and Interoperability IoT Platform for Connected Automated Driving", (submitted to) 13th ITS European Congress, the Netherlands, 3-6 June 2019

[38] Report on the Implementation of the IoT Platform, EC H2020 AUTOPILOT, 2018, to be retrieved via (visited in February 2019) https://autopilot-project.eu/wp-content/uploads/sites/16/2018/10/AUTOPILOT-D2.3-Report-on-the-Implementation-of-the-IoT-Platform-v0.3.pdf

[39] "Developer guide: Interworking Proxy using SDT", oneM2M TR-0039-V-0.0.5, 21-09-2017, to be retrieved via (seen in June 2019), http://www.google.nl/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=2ahUKEwjy2ePqkf_iAhWQecAKHRysCGwQFjABegQIABAC&url=http%3A%2F%2Fwww.onem2m.org%2Fcomp

onent%2Frsfiles%2Fdownload-file%2Ffiles%3Fpath%3DDraft_TR%25255CTR-0039-Developer_guide-SDT-based_implementation-V0_0_5.docx%26Itemid%3D238&usg=AOvVaw0EBIHTKC8t5XQdC_7Eq23v

[40] "Recommendations for commonalities and interoperability profiles of IoT platforms", CREATE-IoT deliverable D06.02, Revision: 1.00, 30 September 2018, to be retrieved via: https://european-iot-pilots.eu/wp-content/uploads/2018/11/D06_02_WP06_H2020_CREATE-IoT_Final.pdf

[41] "Workshop on LSPs use cases: integration and standardisation alignment", CREATE-IoT deliverable D06.09, Revision: 1.00, 22 March 2019, to be retrieved via: https://european-iot-pilots.eu/wp-content/uploads/2020/06/D06_09_WP06_H2020_CREATE-IoT_Final.pdf

[42] Market Drivers and High Level Architecture for IoT enabled Data Marketplaces, https://aioti.eu/wp-content/uploads/2019/02/IoT-data-market-places-drivers-and-architechtures-white-paper-Elloumi-De_Block-Samovicz.pdf

[43] CYBER; Methods and protocols; Part 1: Method and pro forma for Threat, Vulnerability, Risk Analysis (TVRA), ETSI TS 102 165-1 V5.2.3, 2017

[44] CYBER; Critical Security Controls for Effective Cyber Defence; ETSI TR 103 305-x

[45] DATEX II, CEN TS 16157

[46] Sensor interface specification, https://sensoris.org/

[47] High-Level Architecture for IoT-enabled Data Marketplaces - application to mobility, https://euagenda.eu/upload/publications/whitepaper-did-we-just-reach-the-mobility-sector-data-marketplaces-tipping-point.pdf

Annex V Editors and Contributors to this Deliverable

Editors:

Marco Carugi, Huawei, Germany (NEC Europe, UK, till June 2018)

Omar Elloumi, Nokia, France

Main Contributors:

Omar Elloumi, Nokia, France

Jean-Pierre Desbenoit, Schneider Electric, France Patrick Wetterwald, Cisco, France

Georgios Karagiannis, Huawei, Germany Juergen Heiles, Siemens, Germany

Paul Murdock, Landis+Gyr, Switzerland Marco Carugi, Huawei, Germany

Ovidiu Vermesan, Sintef, Norway

Martin Serrano, Insight Centre for Data Analytics, Ireland Carlos Ralli Ucendo, Telefonica, Spain

Arthur van der Wees, Arthur's Legal, Netherlands Franck Le Gall, EGM, France

Marc Girod Genet, Telecom SudParis, France Thomas Klein, IBM, Germany

Jason Mansell, Tecnalia, Spain Sergio Campos, Tecnalia, Spain

Emmanuel Darmois, Commledge, France Aitor Corchero, EURECAT, Spain François Ennesser, Gemalto, France Arne Berre, Sintef, Norway

Said Gharout, Orange, France

Mahdi Ben Alaya, Sensinov, France

Joachim Koss, Germany

Kees Kroep, UDelft,The Netherlands

R. Venkatesha Prasad, EWI, UDelft, The Netherlands

Reviewers:

Patrick Guillemin, WG03 Chair, ETSI, France

Georgios Karagiannis, WG03 Vice-Chair, Huawei, Germany

About AIOTI

AIOTI is the multi-stakeholder platform for stimulating IoT Innovation in Europe, bringing together small and large companies, start-ups and scale-ups, academia, policy makers and end-users and representatives of society in an end-to-end approach. We work with partners in a global context. We strive to leverage, share and promote best practices in the IoT ecosystems, be a one-stop point of information on all relevant aspects of IoT Innovation to its members while proactively addressing key issues and roadblocks for economic growth, acceptance and adoption of IoT Innovation in society.

AIOTI's contribution goes beyond technology and addresses horizontal elements across application domains, such as matchmaking and stimulating cooperation in IoT ecosystems, creating joint research roadmaps, driving convergence of standards and interoperability and defining policies. We also put them in practice in vertical application domains with societal and economic relevance.

AIOTI is a partner for the European Commission on IoT policies and stimulus programs, helping to identifying and removing obstacles and fast learning, deployment and replication of IoT Innovation in Real Scale Experimentation in Europe from a global perspective.

AIOTI is a member driven organisation with equal rights for all members, striving for a well-balanced representation from all stakeholders in IoT and recognizing the different needs and capabilities. Our members believe that we are the most relevant platform for connecting to the European IoT Innovation ecosystems in general and the best platform to find partners for Real Scale Experimentation.