AIOTI

ALLIANCE FOR INTERNET OF THINGS INNOVATION

High Level Architecture (HLA)

Release 3.0

AIOTI WG03 – IoT Standardisation

June 2017

Table of Contents

Table of Figures

1 Highlights and recommendation

In the context of the AIOTI WG03 and by following the evolution on IoT Architectural aspects and available specifications, AIOTI WG03 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 IoT landscape and Semantic Interoperability aspects.

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. 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 an initial 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
- 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.
- Establishes the link to other architectures and frameworks such as Big Data.

An annex describes possible relationships of the HLA functional model with other models.

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

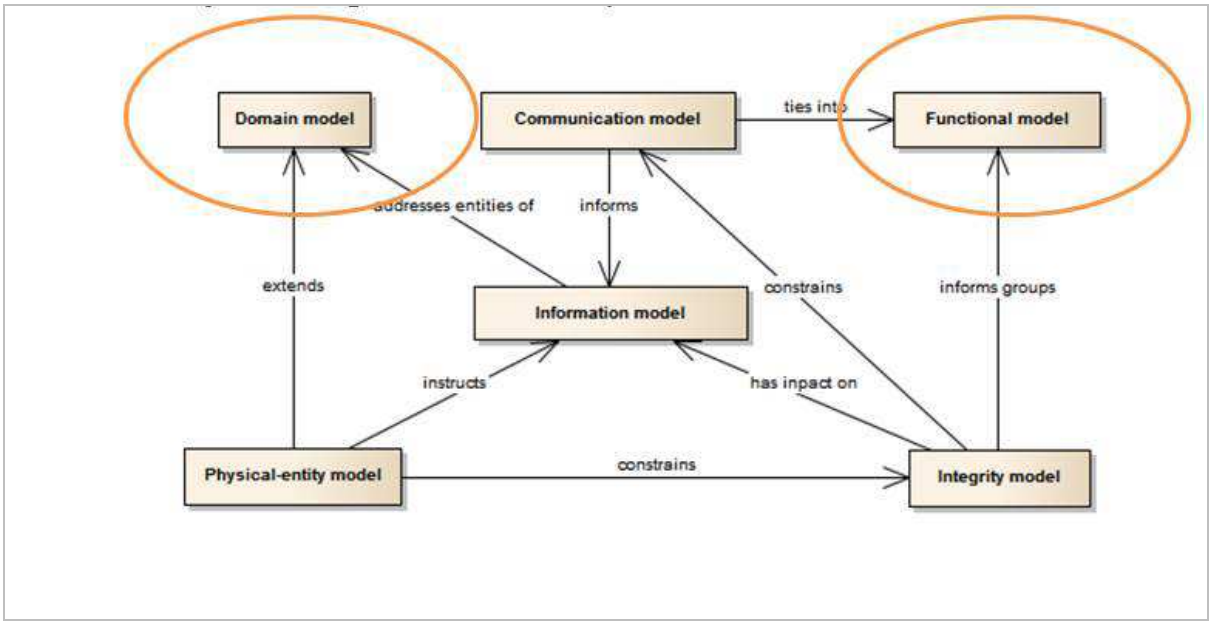

Figure 3-1: Architectural Models based on ISO/IEC/IEEE 42010

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].

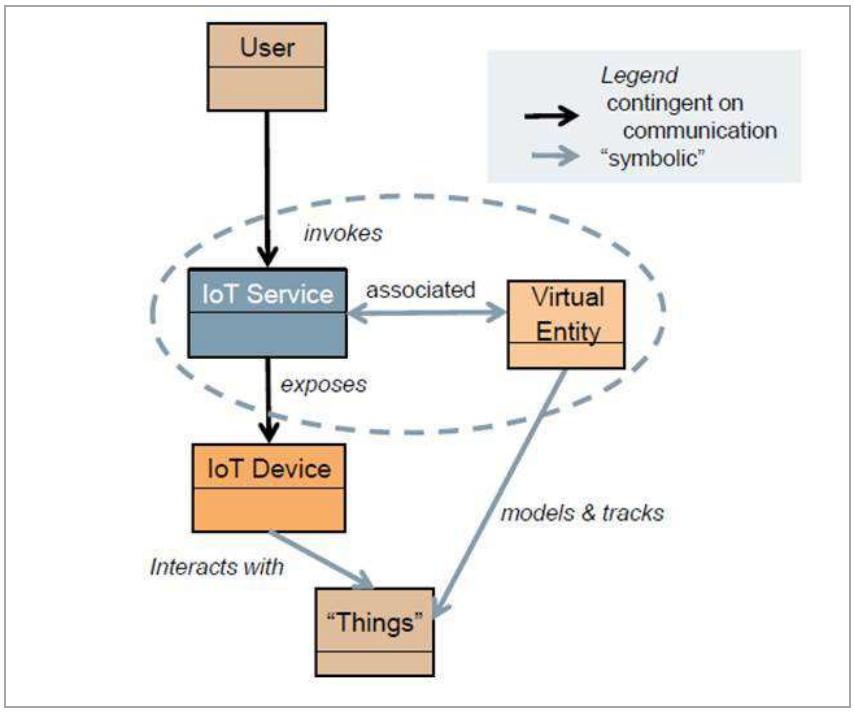

Figure 4-1: Domain Model

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's services.
- **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.

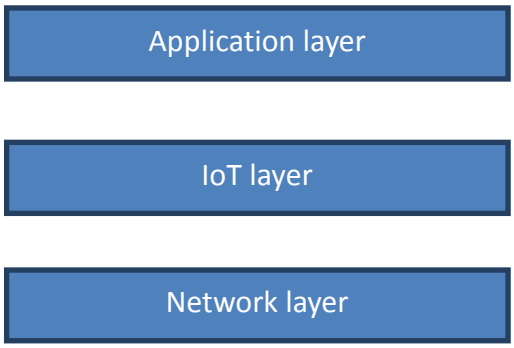

Figure 5-1: AIOTI three-layer functional model.

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 be inferred.

5.2 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".

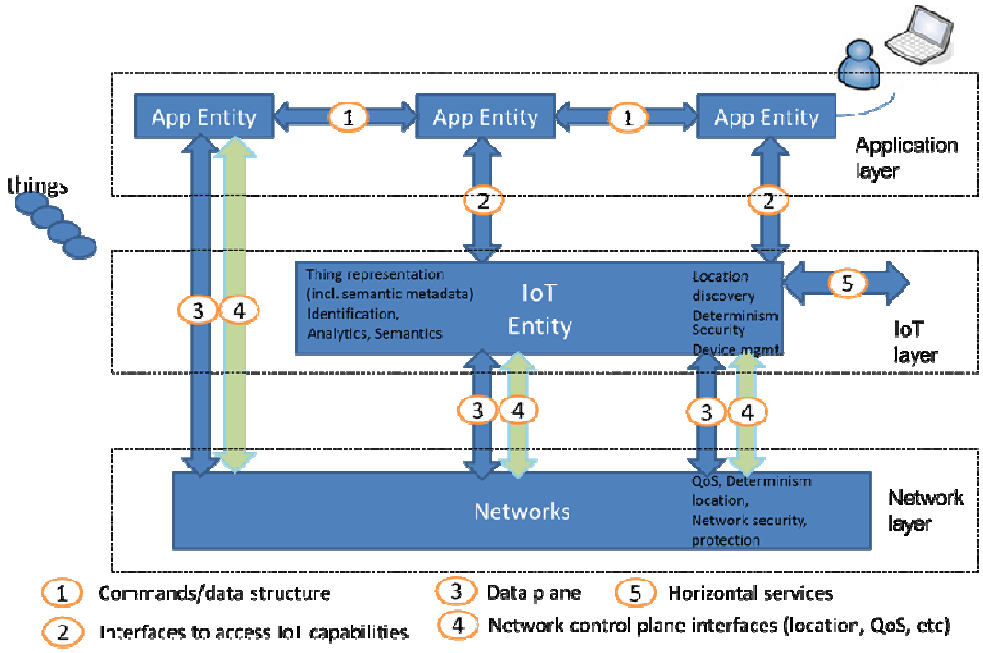

Figure 5-2: AIOTI HLA functional model

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].

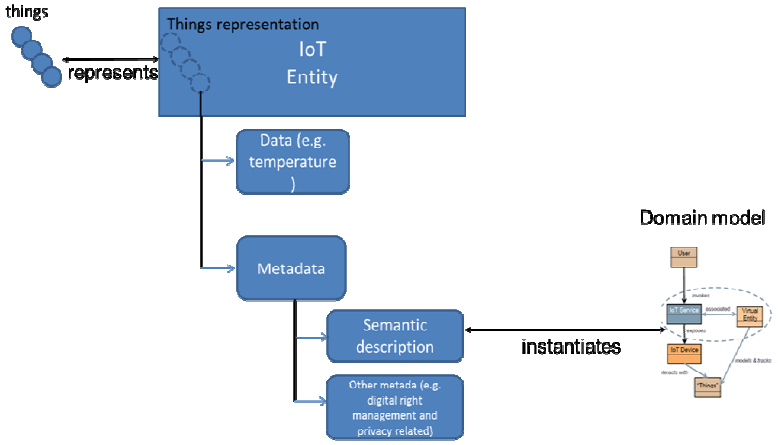

Figure 5-3: Relationship between a thing, a thing representation and the domain model

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 accountability, 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 WG04 that is addressing the policy issues for security and privacy.

5.4 Identification for AIOTI HLA

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 the assembly and commissioning of the systems, and is also relevant for system operations, especially in case of flexible and dynamic interactions between system components. In addition, identification of other entities like data types, properties, or capabilities is needed; however that is related to semantics expressions and ontologies for such entities and not to dedicated identifiers.

IoT systems provide interaction between users and things. In order to achieve this, device components (sensors and actuators), service components, communication components, and other computing components are needed, as shown in Figure 5-4. The virtual entity plays a special role in IoT as it provides the virtual representation of things in the cyber world; it is closely linked to the thing for which identifiers are essential.

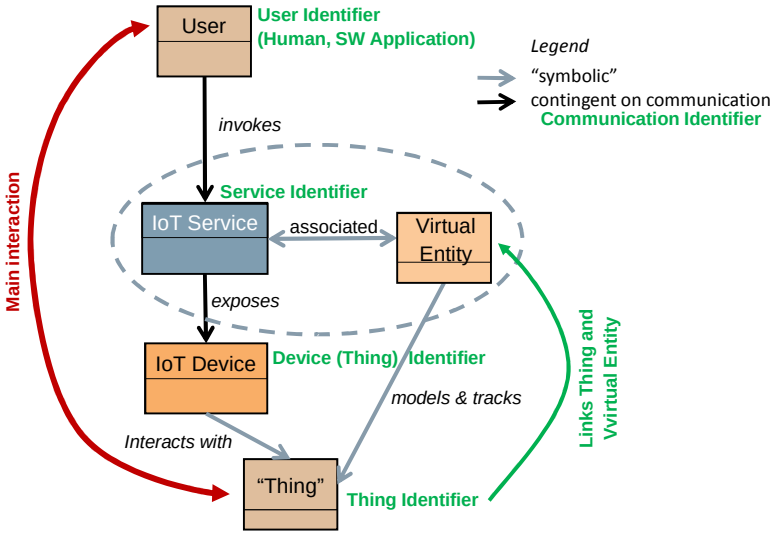

Figure 5-4: Identifiers examples in the IoT Domain Model

In general an identifier is a pattern to uniquely identify a single entity (instance identifier) or a class of entities (type identifier) within a specific context. Figure 5-4 shows some examples of identifiers for IoT.

Things are at the centre of IoT and unique identification of Things is a prerequisite for IoT systems. Kevin Ashton who coined the term “Internet of Things” in 1999 linked the term with identification, specifically Radio Frequency Identification RFID. RFID is one means of identification, but many more exist, given that a thing could be any kind of object:

- Goods along their lifecycle from production to delivery, usage, maintenance until end of life
- Weather conditions in a certain area
- Traffic flow at an intersection
- Vehicles and containers for tracking purposes
- Animals and field yields for smart farming applications
- Humans in case of health and fitness applications
- Digital objects like e-books, music and video files or software

Some of the things are directly connected to a communication network while others are only indirectly accessed via sensors and actors. 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. As identification applies to systems in general, many identification means already exist and are often standardized as domain specific solutions. Users may prefer different identification schemes. A property management company identifies things according to the building location, floor and room number while the producer of the thing uses its own serial number scheme. Furthermore the identifier might be unique within its current usage context, but new applications may result in conflicts if the same identifier is used in other systems. This raises issues of interoperability, uniqueness and linkage between identifiers which need further elaboration.

Users that interact with the things could be humans or software applications. Identification of the users is needed, especially if access is limited and/or tracked. From a security point of view, authentication requires a second step to validate the claims asserted by the identity. Privacy concerns must also be considered.

In the case of communication networks, the source and destination of the communication relationships must be identified. Here the identifiers are bound to the specific communication technology and defined as part of the standardization of the technology. IP networks use IPv4 and IPv6 addresses, Ethernet and WLAN use MAC addresses and fixed and mobile phone networks use phone numbers. Communication identifiers may not be a good fit as thing identifiers as the communication address of a thing may change (e.g. the communication interface or network topology may change if a different communication service provider is selected). Furthermore, some things don’t have communication interfaces whilst others may have more than one (e.g. for redundancy reasons).

6 Deployment considerations for HLA

6.1 Introduction

This section 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 visualisation of large data sets. Those disciplines are equally applicable to IoT data sets.
- **Virtualisation:** 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.

6.2 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 section we focus on an overview of cloud native principles as well as recent edge computing initiatives, namely ETSI 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.

6.2.1 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)

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

- Fine-grained micro-services architectures, lightweight containers deployment and service orchestration.
- DevOps with holistic service monitoring and decentralized continuous delivery.

6.2.2 Edge cloud initiatives

6.2.2.1 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 6-1 shows the framework for Mobile Edge Computing consisting of the following entities:

- Mobile Edge Host, including the following:
  - mobile edge platform;
  - mobile edge applications;
  - virtualisation infrastructure;
- Mobile Edge System Level management;
- Mobile Edge Host level management;
- External related entities, i.e. network level entities.

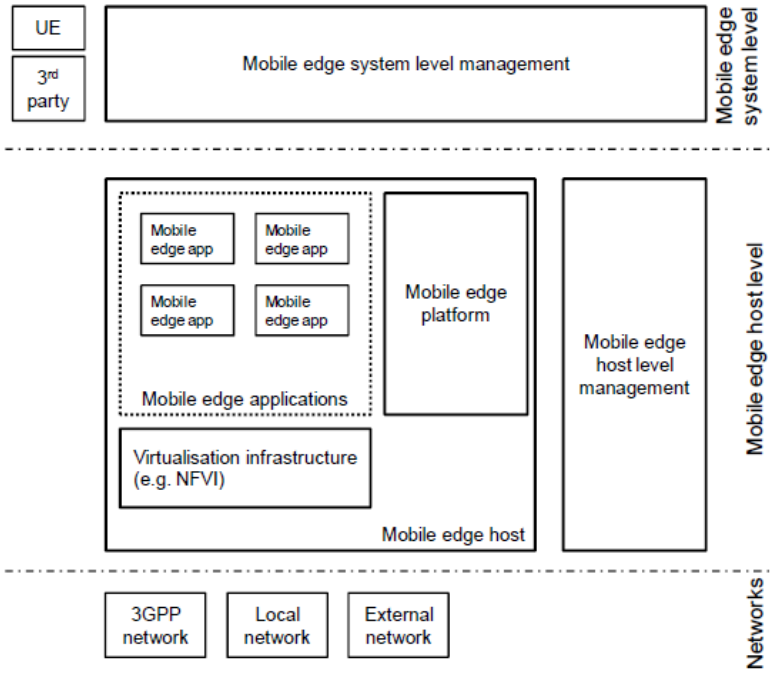

Figure 6-1: Mobile Edge Computing Framework [ETSI GS MEC 003]

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 6-1 could be mapped to App Entity in HLA.

6.2.2.2 OpenFog

The OpenFog Architecture is a system-level architecture that extends elements of compute, 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.

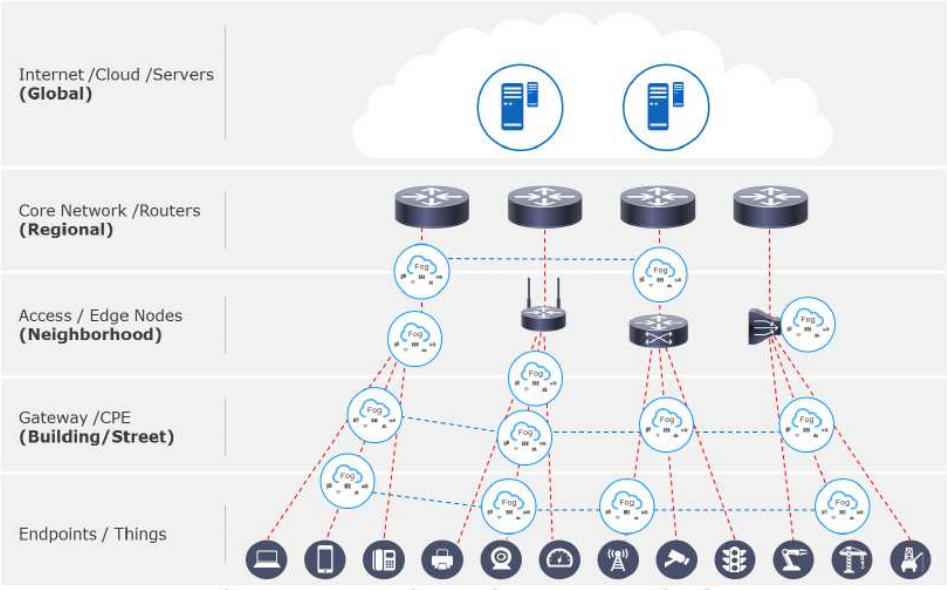

Figure 6-2: OpenFog cloud hierarchy

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

6.3 Big Data

6.3.1 Definitions

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

- **Big Data** (source 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** (source 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 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.

6.3.2 IoT data roles (source ITU-T Y.4114)

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 ecosystem as shown in Figure 6-3.

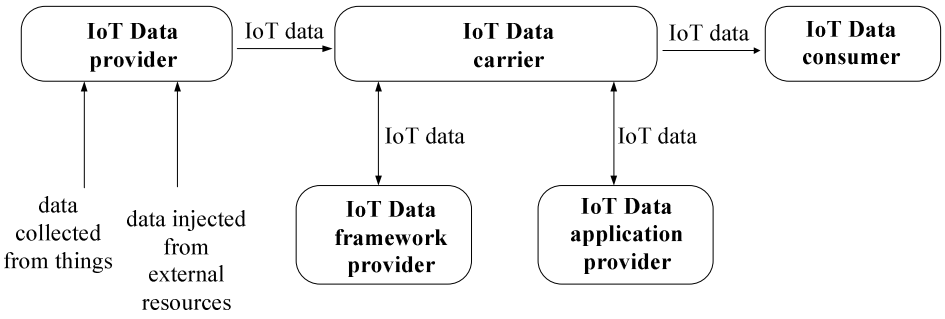

Figure 6-3: IoT data roles

- **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 provide a mapping between ITU Y.4114 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 6-1: Mapping of ITU Y.4114 to AIOTI HLA

6.3.3 IoT data operations (source ITU-T Y.4114)

Considering that the diverse set of concrete IoT deployments does not imply a unique logical sequencing of the various IoT data operations, Figure 6-4 provides an abstract representation of the various IoT data operations and related data flows.

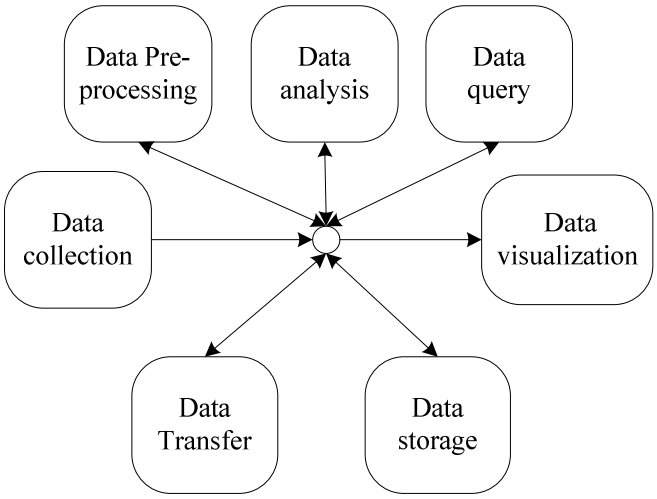

Figure 6-4: IoT data operations

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.

6.4 Virtualization

Virtualizing IoT in an all IP environment enables wide scale and profitable IoT growth.

While it may sound as a trendy term, Virtualizing IoT goes much beyond a mere technology paradigms. The market motivations are about flexibility for building new services at run time, scalability and ensuring carrier and enterprise grade availability.

Virtualizing IoT builds on two key pillars which are strongly related. First cloud native principles (as described 6.2.1) 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.

6.4.1 Network Slicing and Virtualization

Several initiatives, such as 3GPP, BBF, ETSI ISG NFV, IETF and ITU-T, started 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 6-5.

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 Virtualisation (NFV) [11], which is the principle of separating network functions from the hardware they run on by using virtual hardware abstraction.

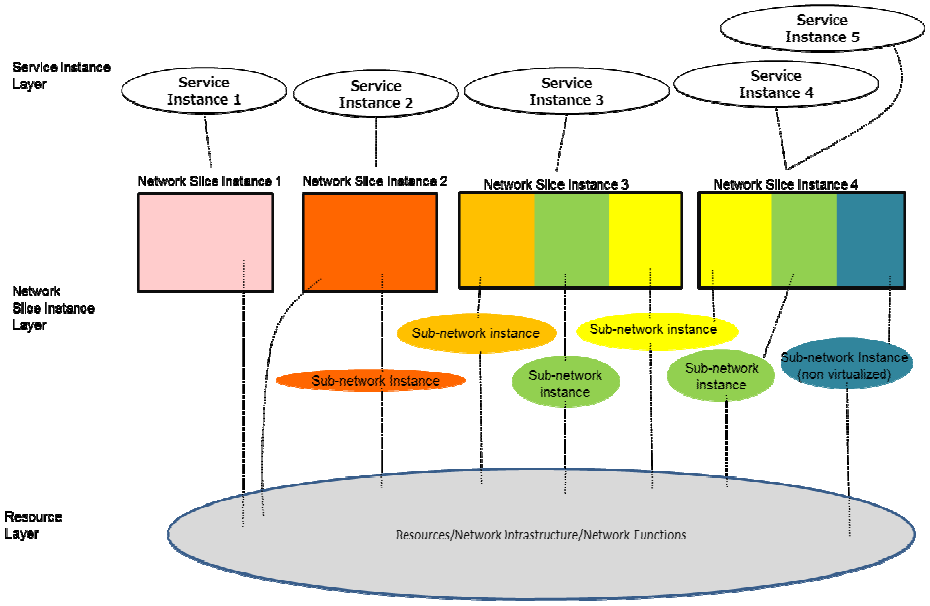

Figure 6-5: NGMN Network Slicing conceptual outline [10]

From the perspective of 3GPP [9], network slicing enables operators to create networks customised 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 slices architectures in 5G scenarios.

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

The purpose of this section is to provide examples of mapping of existing SDO/alliances 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 section does not intend to be exhaustive, other mappings can be added in future releases of this document.

7.1 ITU-T

In ITU-T Recommendation Y.2060 "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 7-1, 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.

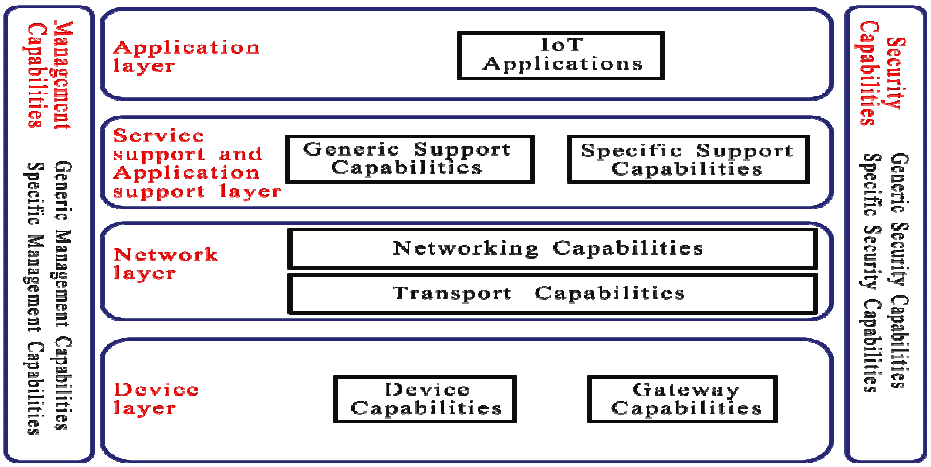

Figure 7-1: ITU-T Y.2060 IoT Reference Model

Figure 7-2 provides an initial high level mapping of the ITU-T Y.2060 IoT Reference model to AIOTI HLA functional model.

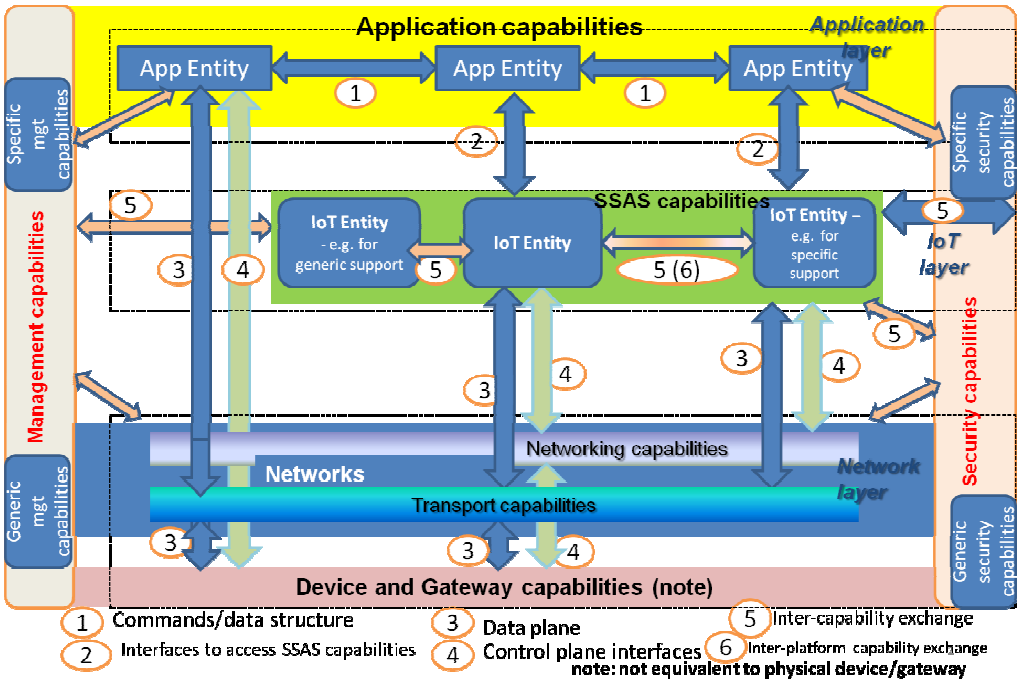

Figure 7-2: ITU-T IoT Reference Model mapping to AIOTI WG03's HLA functional model

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.2068 ("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 draft Recommendation Y.NGNe-IoT-Arch ("Architecture of the Internet of Things based on NGN evolution").

7.2 oneM2M

Figure 7-3 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).

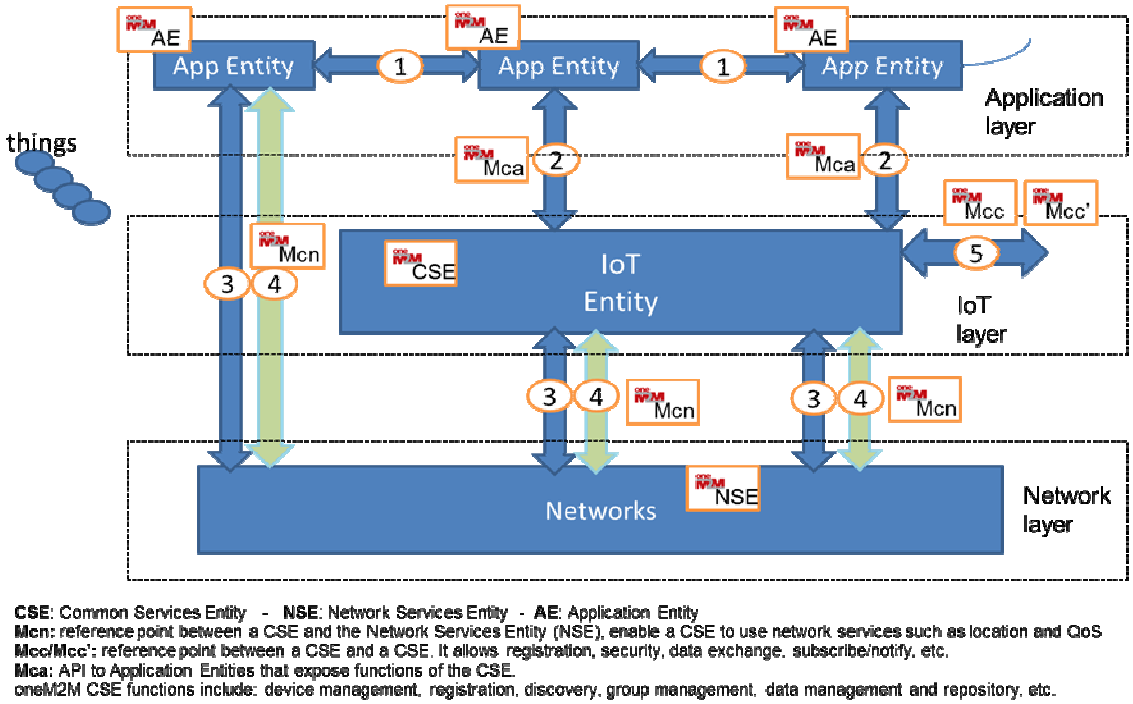

Figure 7-3: Mapping oneM2M to AIOTI HLA

oneM2M has specified all interfaces depicted in Figure 7-3 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.

7.3 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 7-4 provides a three-tier architecture as specified in [5].

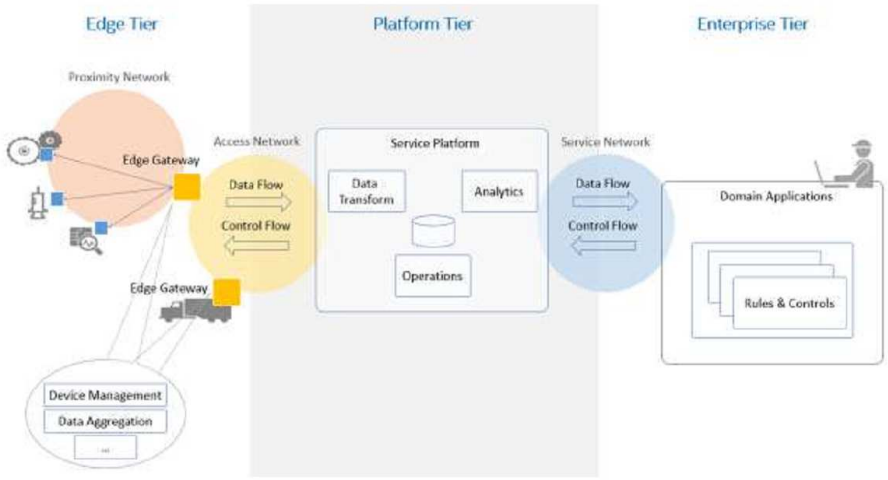

Figure 7-4: IIC three tier IIS architecture

The mapping of IIC to the AIOTI HLA is depicted in the following Figure.

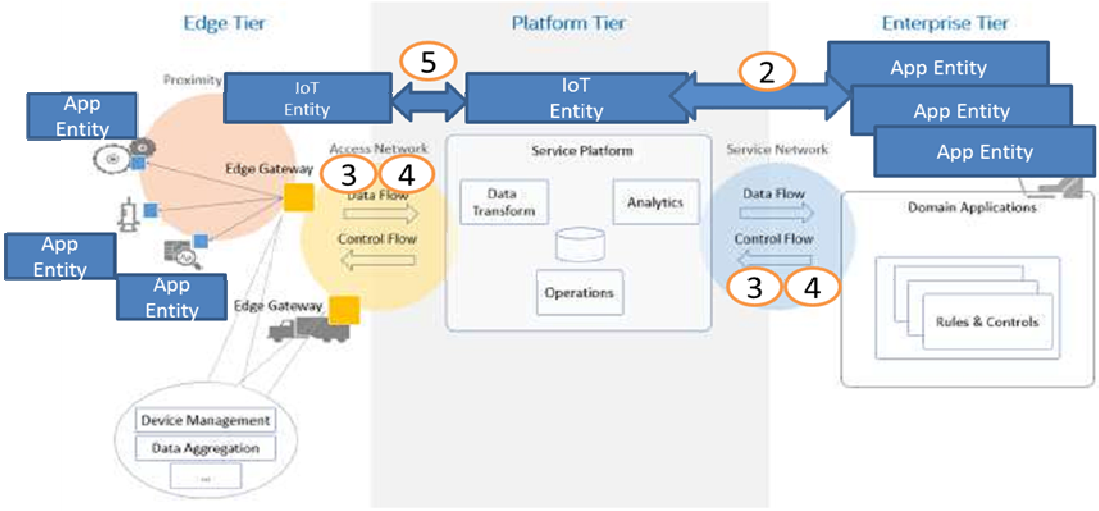

Figure 7-5: Mapping HLA to IIC three tier IIS architecture

In Figure 7-5, 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.

7.4 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.

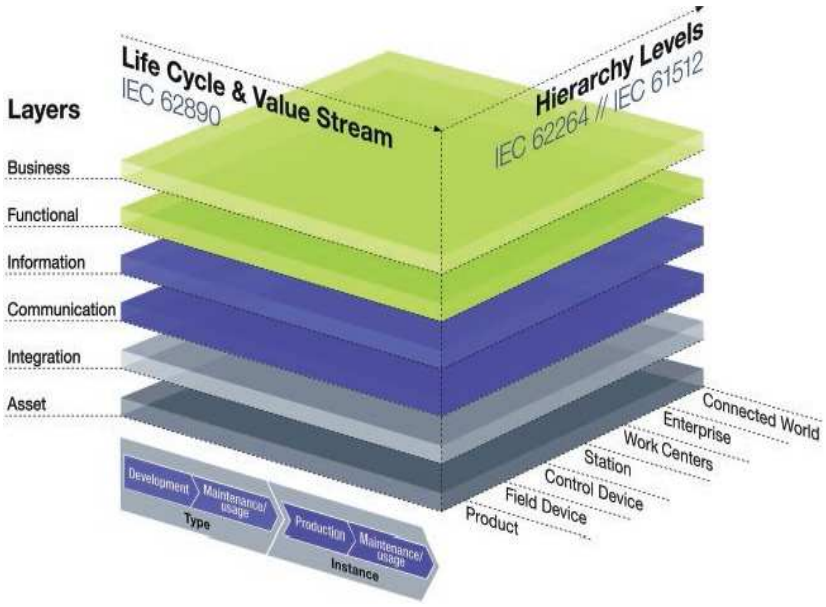

Figure 7-6: RAMI 4.0 reference architecture

The reference architecture model as shown in Figure 7-6 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 represent 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 a various architectural views.

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

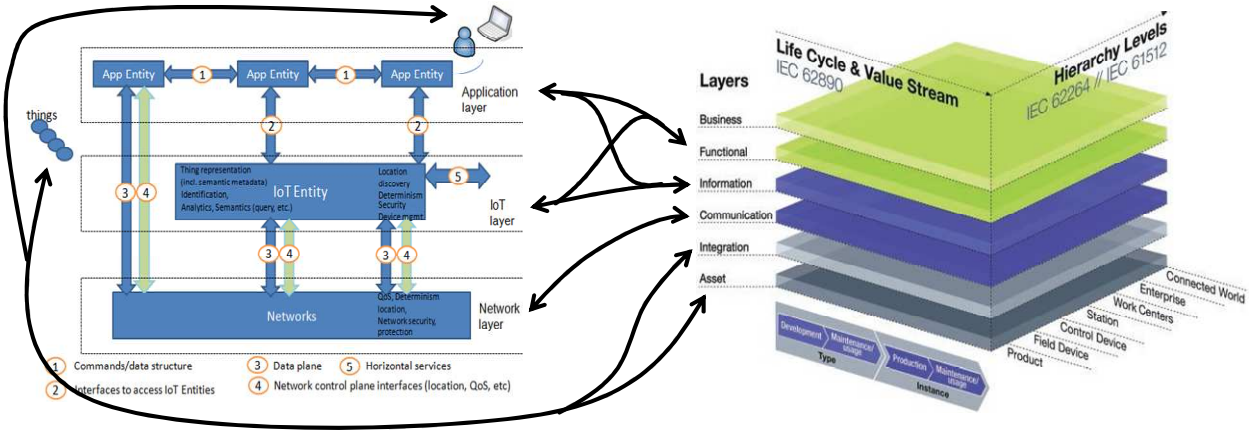

Figure 7-7: Mapping RAMI 4.0 to AIOTI HLA – functional model

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

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 an 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.

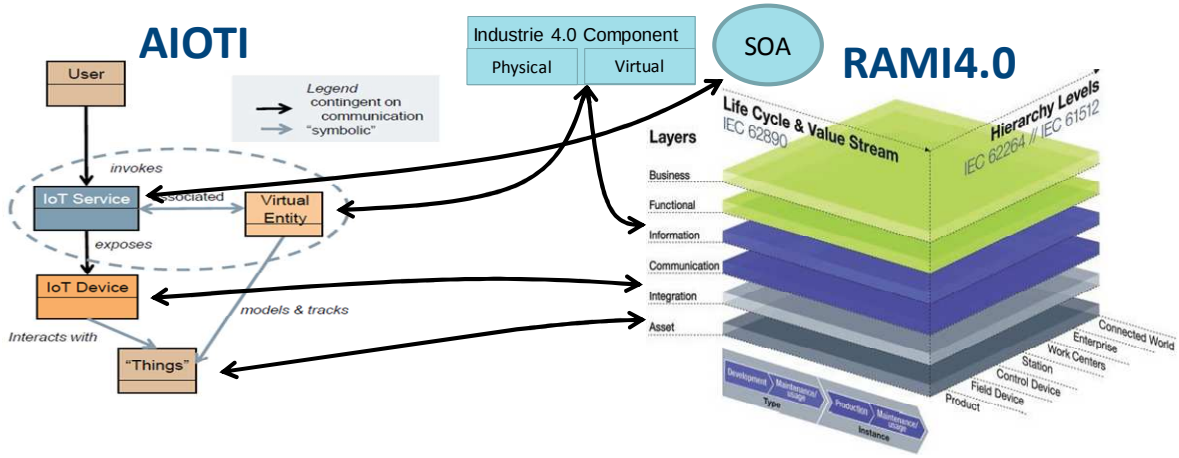

Figure 7-8: Mapping RAMI 4.0 to AIOTI HLA – domain model

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

- 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 Relationship to other functional models or systems

8.1 Introduction

This section 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 8-1 shows in particular interactions with Big Data frameworks and other service platforms (banking, maps, open data, etc.).

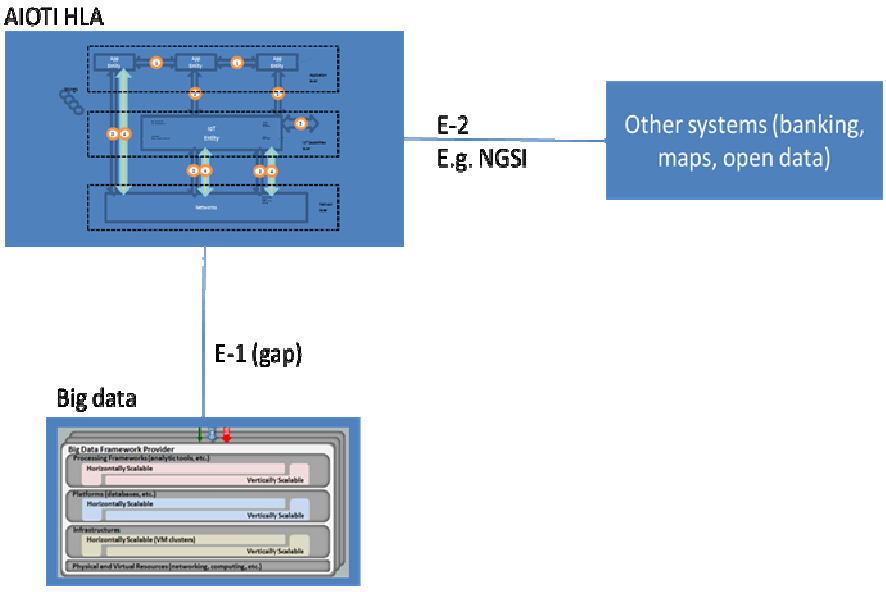

Figure 8-1: Relationship to other systems

Figure 8-1 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.

8.2 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 8-2.

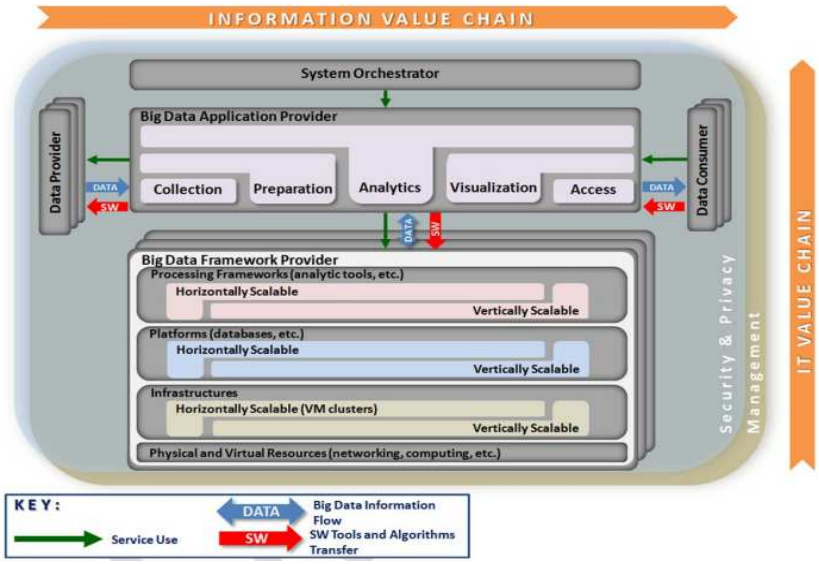

Figure 8-2: NIST Big Data reference architecture

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 8-3 depicts this mapping example.

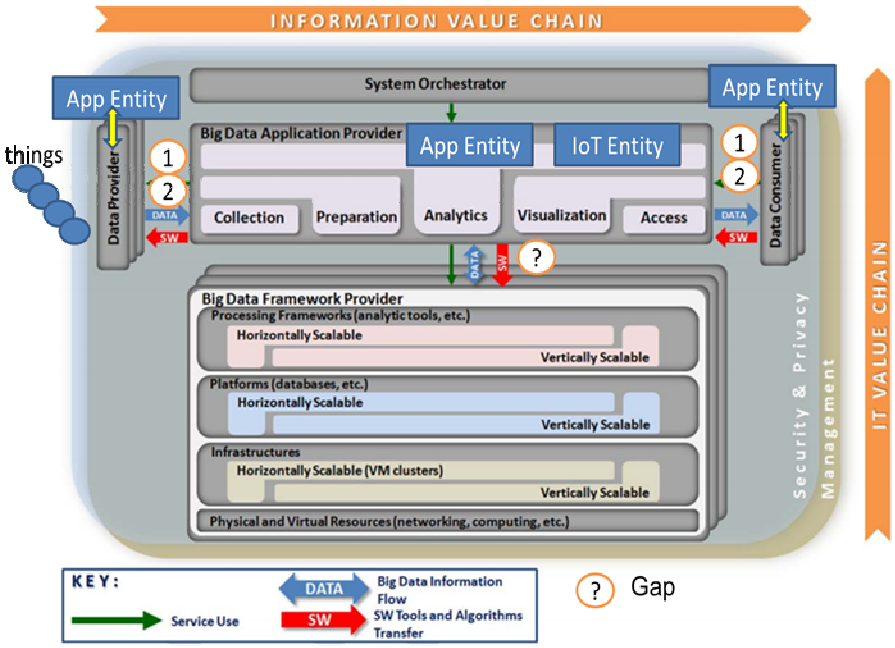

Figure 8-3: Mapping AIOTI functional model entities to NIST big data reference architecture

In Figure 8-3 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.

8.3 Relationship to other service platforms

Figure 8-1 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.

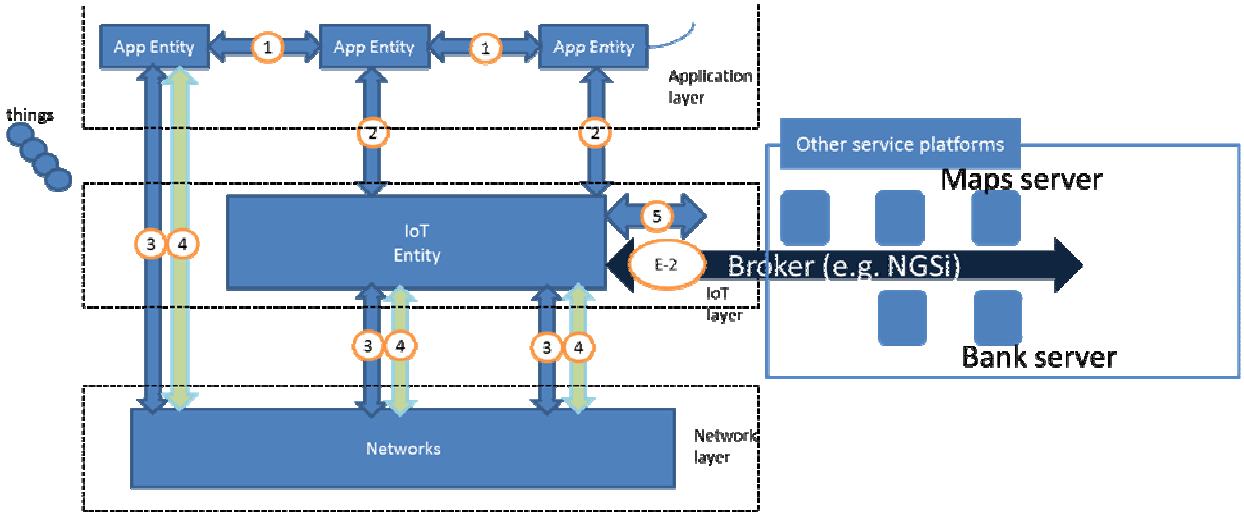

Figure 8-4: E-2 interface illustration

Figure 8-4 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.

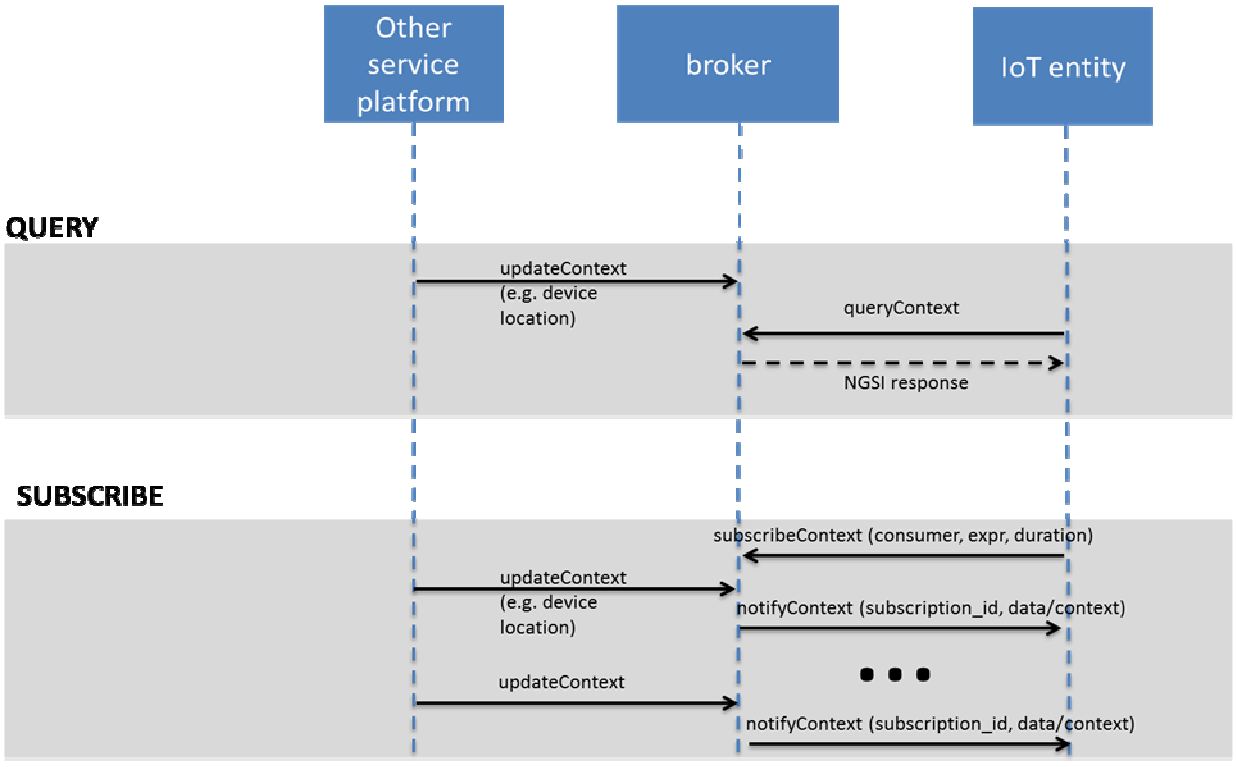

Figure 8-5: Example of message flow illustrating the E-2 interface

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.

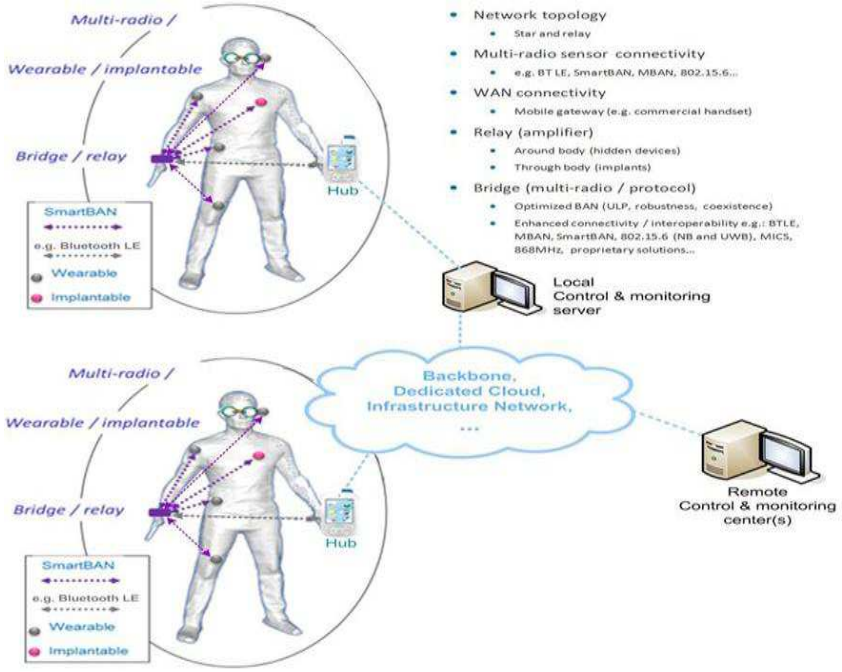

Figure I-1: ETSI SmartBAN deployment example concepts

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

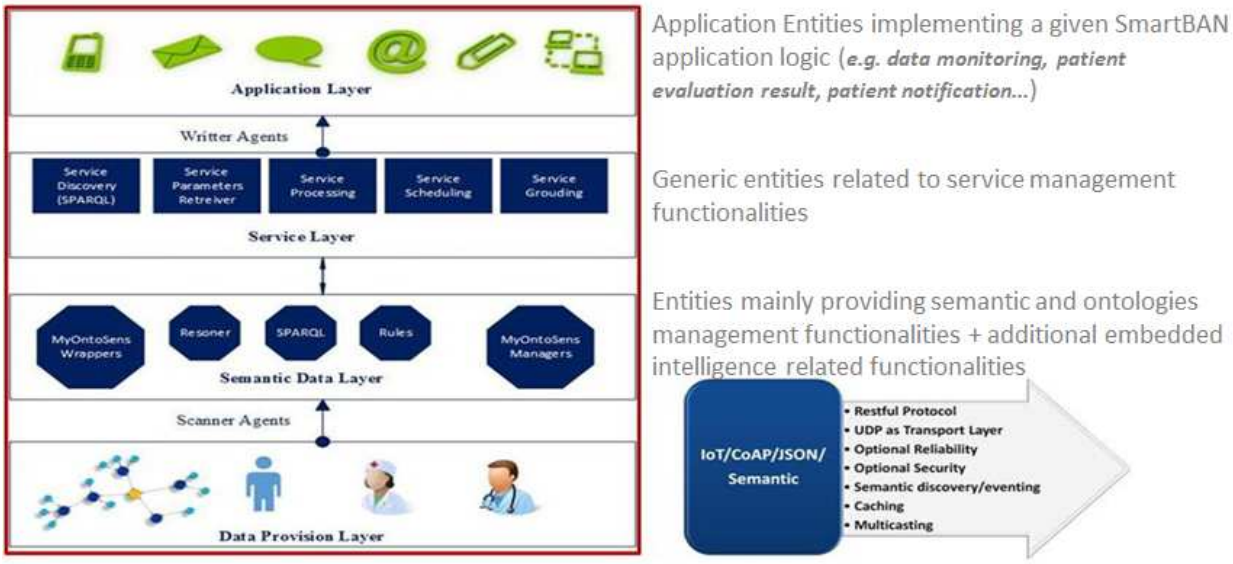

Figure I-2: ETSI SmartBAN reference architecture

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 provides a binding between the ETSI SmartBAN architecture and the AIOTI HLA:

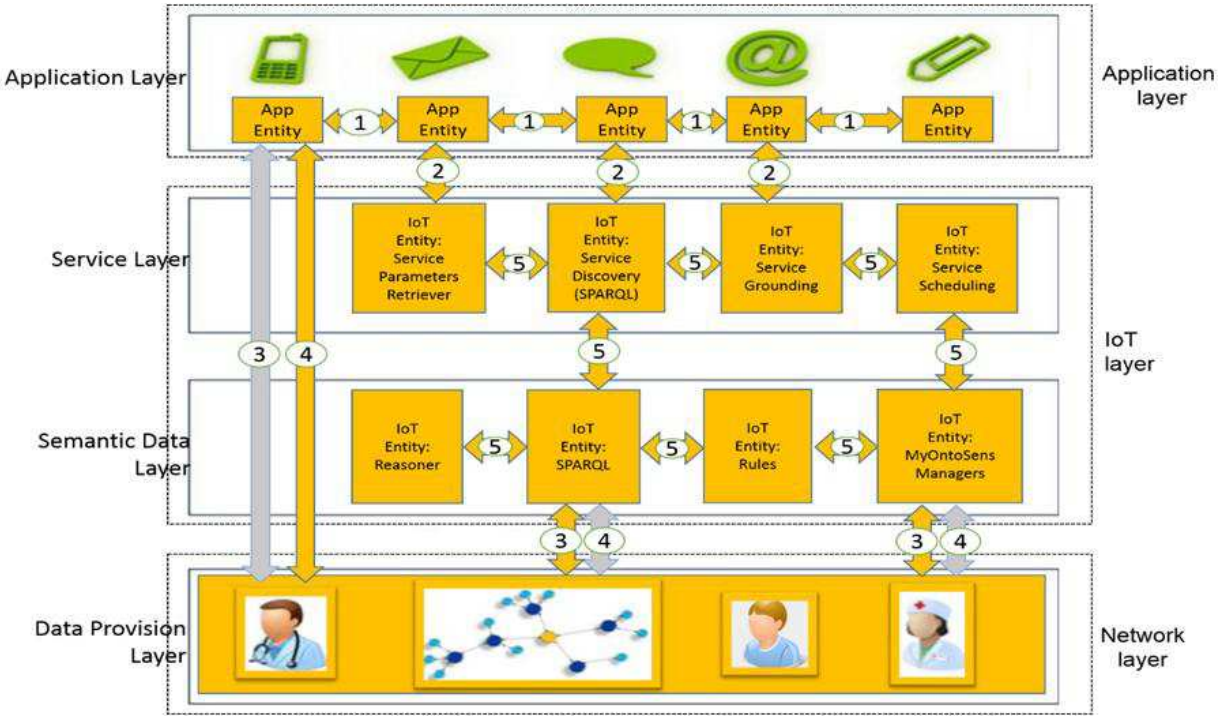

Figure I-3: ETSI SmartBAN reference architecture mapping to AIOTI HLA

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

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 standardisation 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 were the problems arise and where – in the IoT standardisation landscape - their resolution can be anticipated.

Those gaps are listed in Table 8-1 below that list a certain number of gaps and a tentative identification of the areas of the AIOTI HLA Functional model were 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 8-1: IoT Gaps mapped on the AIOTI HLA

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

Table 8-2 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	**+ –** 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 consumption

		might be limited.	
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 8-2: Advantages and disadvantages of end device, edge and cloud computing

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.2060 “Overview of the Internet of Things”: https://www.itu.int/rec/T-REC-Y.2060-201206-I

[4] oneM2M Functional Architecture Release 1 http://www.etsi.org/deliver/etsi_ts/118100_118199/118101/01.00.00_60/ts_118101v010000p.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

[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_Opportunities_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/januar/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

Editors:

HLA – Editor:

Omar Elloumi, Nokia, France

HLA – Functional Model 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, NEC Europe, UK
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

Contributing Companies, Projects, Initiatives, SDOs:

Additional Contributing Experts:

Reviewers:

Patrick Guillemin, WG03 Chair, ETSI, France
Georgios Karagiannis, WG03 alternate Chair, Huawei, Germany

Acknowledgements

The AIOTI would like to thank the European Commission services for their support in the planning and preparation of this document. The recommendations and opinions expressed in this document do not necessarily represent those of the European Commission. The views expressed herein do not commit the European Commission in any way.