Alliance for AI, IoT and Edge Continuum Innovation

High Level Architecture (HLA) Report

Release 7.0

AIOTI WG Standardisation

18 November 2025

Executive Summary

The AIOTI High-Level Architecture (HLA) Report is the outcome of comprehensive discussions of the AIOTI WG Standardisation focusing on large-scale IoT and Edge Computing pilot deployments and related works of the Standards Development Organisations (SDOs).

Specifically, this report presents the use of ISO/IEC/IEEE 42010 within the IoT domain and proposes different deployment options considering architectural matters related to cloud and edge computing, Big Data, virtualization, security, privacy and (platform) interoperability.

Based on past discussions within AIOTI WG Standardisation, this Release provides enhancements on the following new or partially developed topics, still with respect to IoT architectural concerns:

- ISO/IEC 30141:2024 IoT - Reference architecture
- Relationship to EUCloudEdgeIoT.eu Open Continuum Reference and Mapping of HLA to Compositional view of the Continuum Reference Architecture, based on results from the following in EUCloudEdgeIoT.eu projects: 6G-Cloud architecture, COGNIT Architecture and CODECO Architecture
- Generative AI, AI Agents and Agentic AI

Table of Content

Table of Figures

List of Tables

1. Objectives

This document provides a proposal for a high-level IoT and Edge Computing 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 WG Standardisation and takes into account the work of Standards Development Organisations (SDOs), Consortia, and Alliances in the IoT and Edge Computing space.

This document:

- Introduces the use of ISO/IEC/IEEE 42010 by AIOTI WG Standardisation
- 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 WG Standardisation 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.

Based on past discussions within AIOTI WG Standardisation, this Release provides enhancements on the following new or partially developed topics, still with respect to IoT architectural concerns:

- ISO/IEC 30141:2024 IoT - Reference architecture
- Relationship to EUCloudEdgeIoT.eu Open Continuum Reference and Mapping of HLA to Compositional view of the Continuum Reference Architecture, based on results from the following in EUCloudEdgeIoT.eu projects: 6G-Cloud architecture, COGNIT Architecture and CODECO Architecture
- Generative AI, AI Agents and Agentic AI

2. Use of ISO/IEC/IEEE 42010

A key recommendation from AIOTI WG Standardisation 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 WG Standardisation 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 WG Standardisation does not provide a complete architecture description for IoT which conforms to the standard. **Figure 1** provides an overview of architectural models as described in ISO/IEC/IEEE 42010.

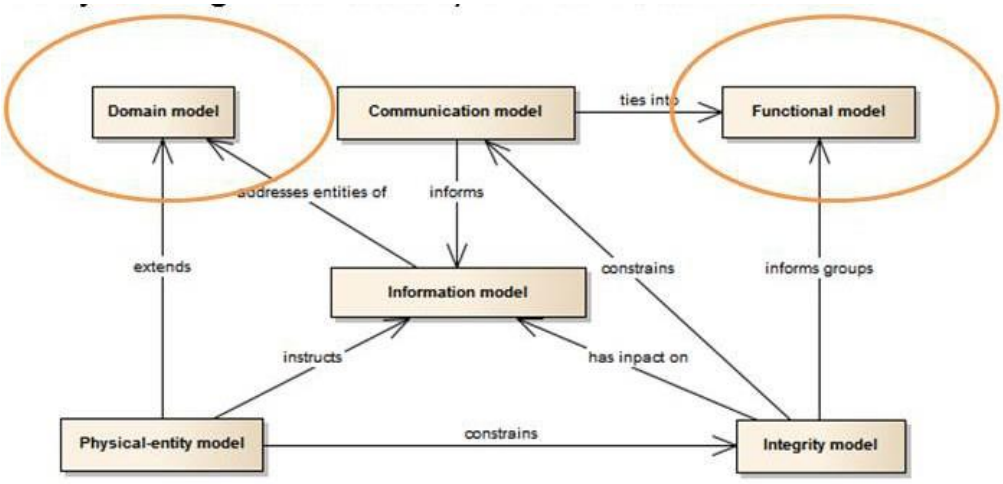

Figure 1: Architectural Models based on ISO/IEC/IEEE 42010

With respect to **Figure 1**, AIOTI WG Standardisation 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.

3. 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].

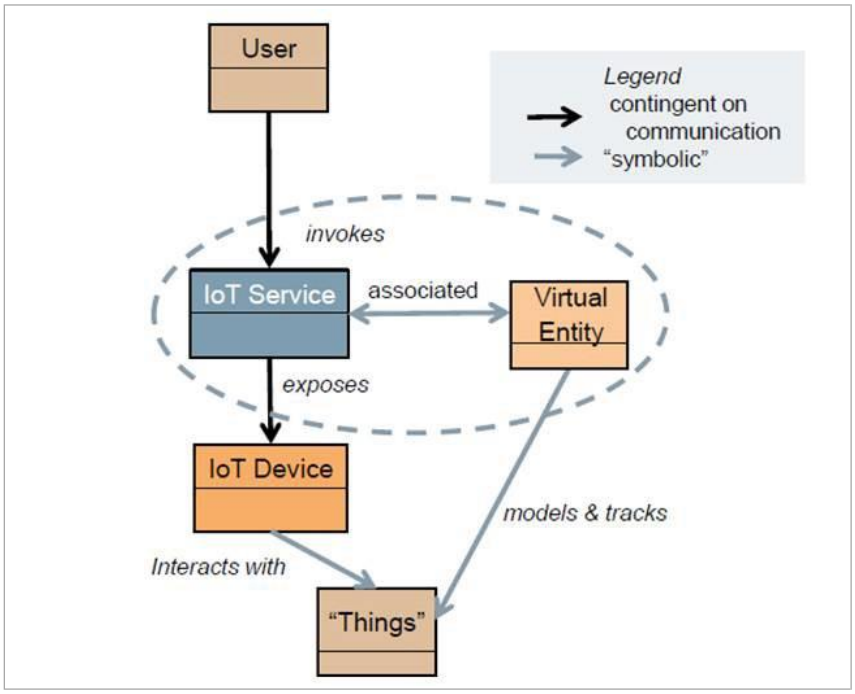

Figure 2: Domain Model

The domain model, shown in **Figure 2**, 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.

4. 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.

4.1 AIOTI layered approach

The functional model of AIOTI is composed of three layers as depicted in **Figure 3**:

- **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.

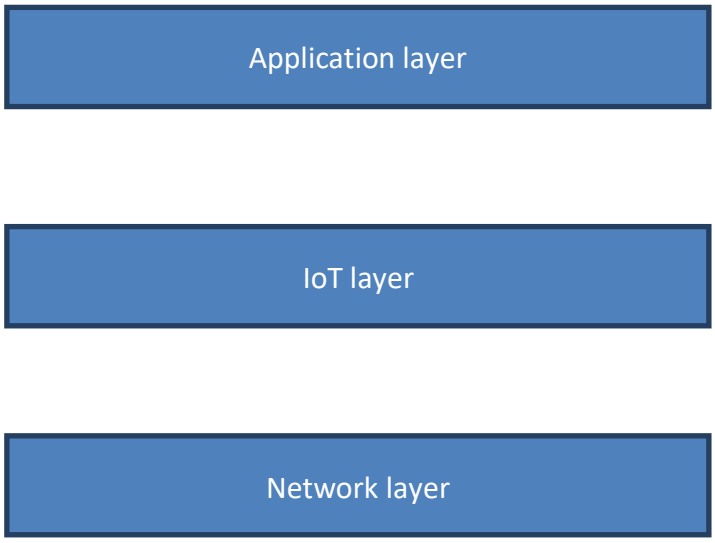

Figure 3: AIOTI three-layer functional model

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.

4.2 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 4** provides a high level AIOTI functional model, referred to as the "AIOTI HLA functional model".

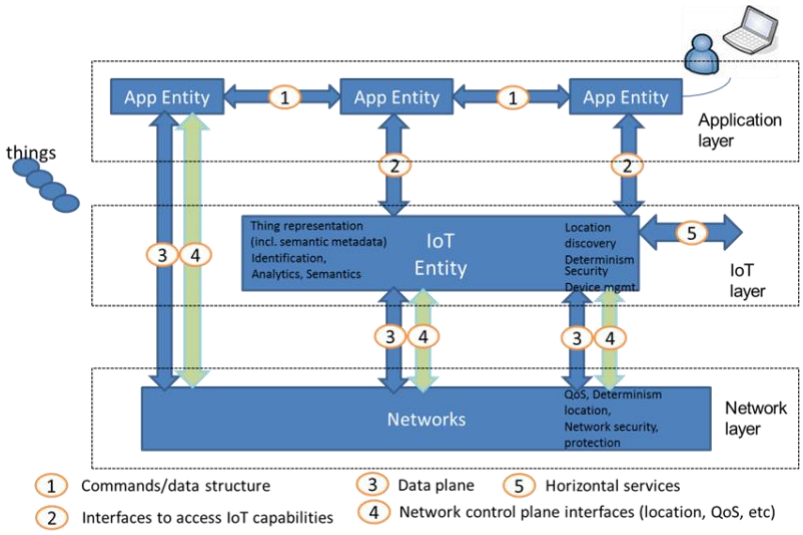

Figure 4: AIOTI HLA functional model

Functions depicted in **Figure 4** 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. When designing IoT based systems, it can be convenient to take a knowledge centred perspective that hides the details of IoT devices and their communication technologies in favour of semantic descriptions of virtual objects, their affordances and information models (ontologies). A knowledge centred perspective enables a system of systems approach for distributed knowledge fusion and low code (intent-based) control, as well as simplifying orchestration and overall management.
- **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 4** 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 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].

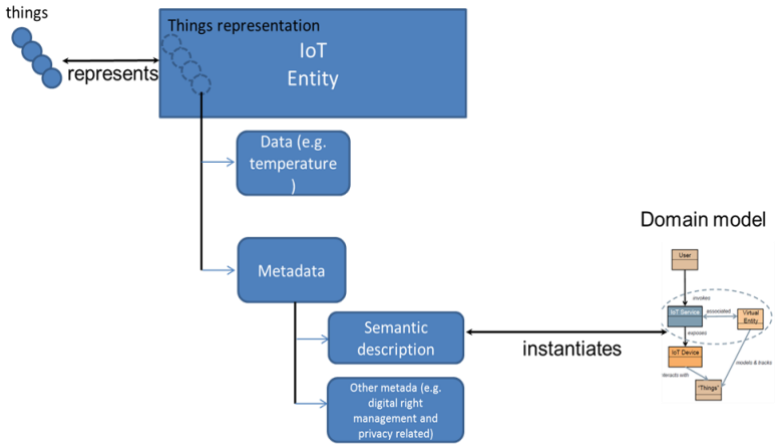

Figure 5: Relationship between a thing, a thing representation and the domain model

4.3 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.

5. 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** shows the different entities with the related identifiers in the IoT Domain Model.

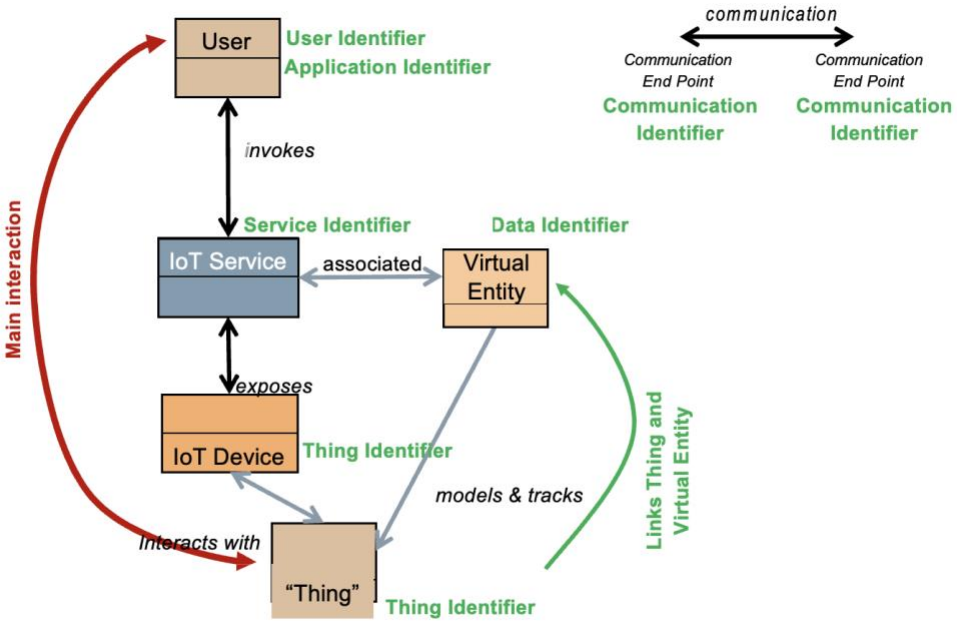

Figure 6: Identifiers examples in the IoT Domain Model

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 WG Standardisation. [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.

6. HLA Deployment considerations

6.1 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.

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 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 in **Table 7** 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)
- Fine-grained micro-services architectures, lightweight containers deployment and service orchestration.
- DevOps with holistic service monitoring and decentralized continuous delivery.

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

6.2.2 Edge cloud initiatives

6.2.2.1 ETSI Multi-access Edge Computing

Multi-access 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 from Mobile Edge Access to 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 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.

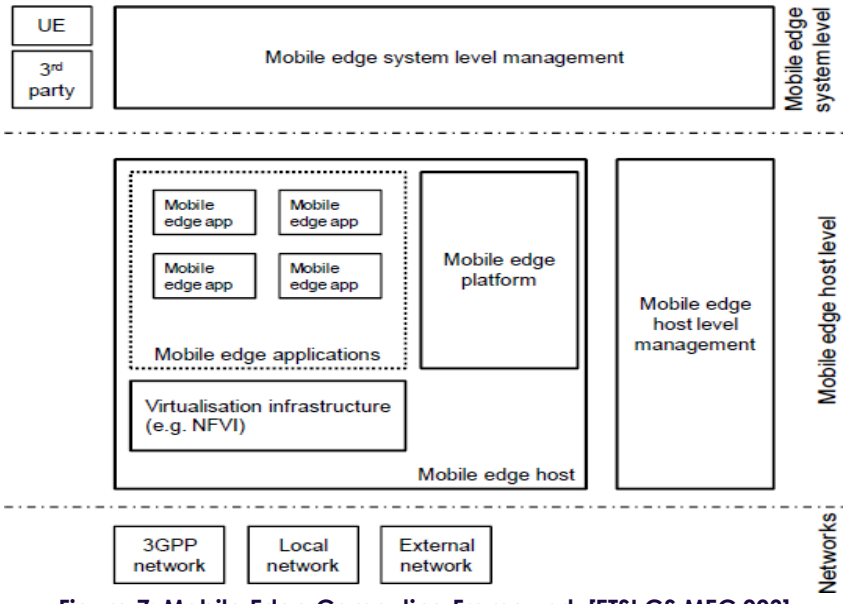

Figure 7: Mobile Edge Computing Framework [ETSI GS MEC 003]

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** could be mapped to App Entity in HLA.

6.2.2.2 EUCEI

The EuEdgeCloudIoT initiative is currently developing a reference CEI architecture, derived from input from key projects in different topics: metaOS, cognitive computing, and swarm computing. Each of these projects is proposing a reference architecture for orchestration aspects across IoT-Edge-Cloud, adopting different approaches with a common purpose: the better support of applications deployment and runtime across an **IoT-Edge-Cloud continuum**.

Common to these projects is the integration of AI/ML to bring specific benefits to the overall infrastructure, industry, and citizens. Aspects such as improved data sovereignty, together sense-reason-act loops based on localised, low-latency networks and processing nodes are some of the topics that are being covered by the design of the proposed architecture.

Relevant in this context is the definition of a taxonomy deriving from the different projects, which is expected to provide more alignment across different concepts, e.g., offloading at a network level and at a computational level. **Figure 8** provides an overview on the categorized aspects being covered by this taxonomy.

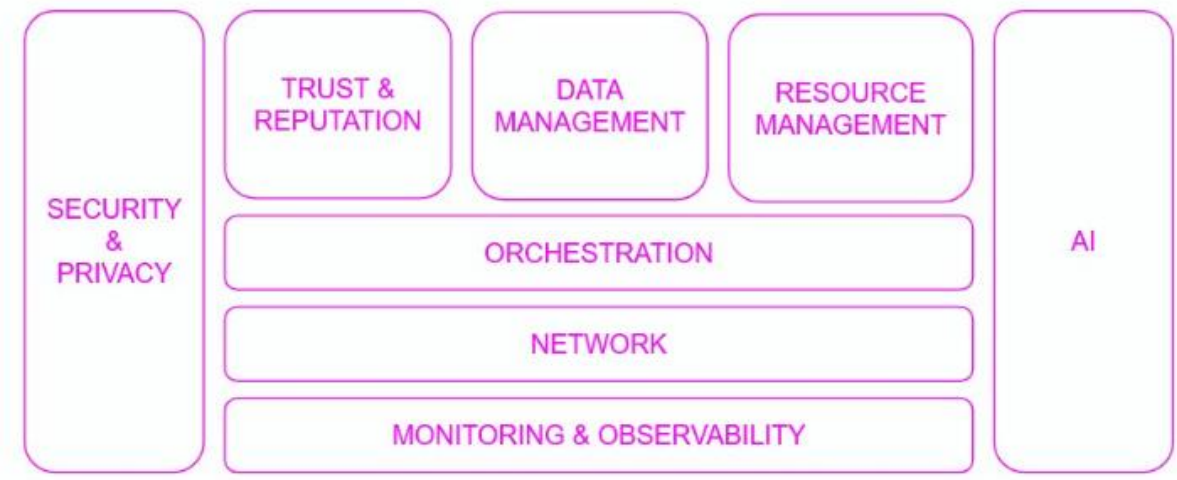

Figure 8: Overview on the categorized aspects in the EuEdgeCloudIoT initiative taxonomy, copied from https://zenodo.org/records/8403593

Important to highlight is the integration of data management and observability together with computational and networking orchestration, handled by an AI/ML orchestration plane. At the current stage, the functional blocks mentioned are being defined, based on reference architectures of active CEI RIAs.

6.2.2.3 Web of Things

The Web of Things supports virtualisation of IoT devices using the Resource Description Framework (RDF) for describing the affordances and semantics of virtual objects acting on behalf of sensors & actuators, services, data streams, network components, software, see e.g., W3C WoT use cases. This lowers the cost and complexity for developing applications using heterogeneous technologies and standards.

Affordances are described, in a programming language neutral way, in terms of the properties, actions and events exposed by the virtual objects, aka *things*. *Thing Descriptions* further cover security and protocol bindings for connecting to the IoT devices.

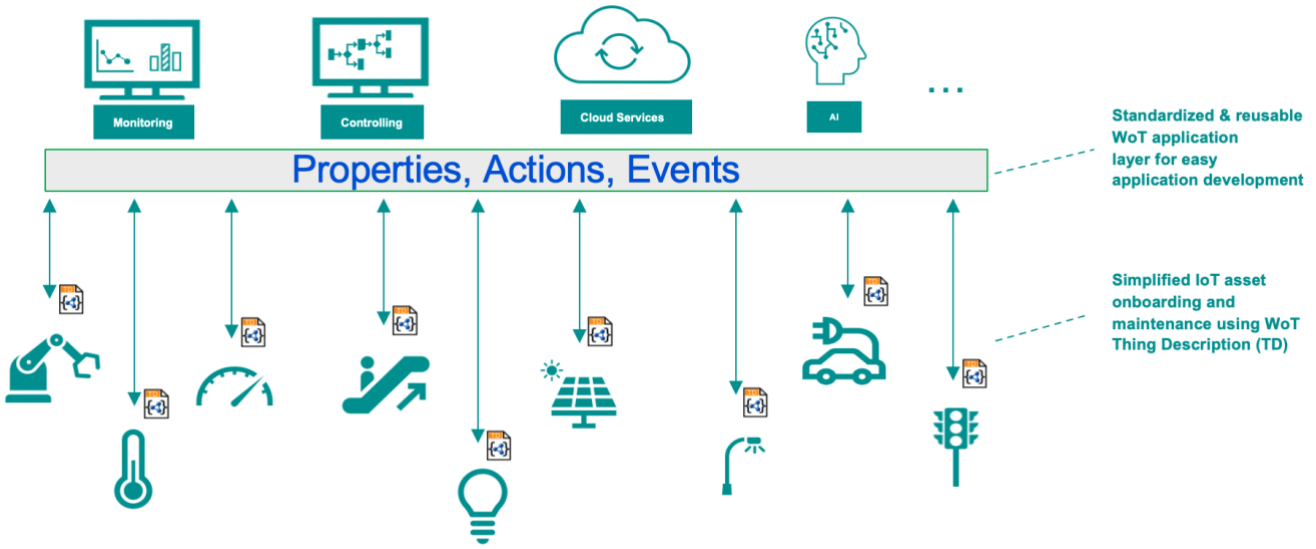

Figure 9: Web of Things simplifies application development, figure provided by Sebastian Käbisch (co-chair W3C WOT WG)

W3C has developed related standards for the architecture, discovery, scripting API and binding templates. A representation of the Web of Things architecture is provided in **Figure 10**. This illustrates how the Web of Things can be applied across the cloud-edge continuum.

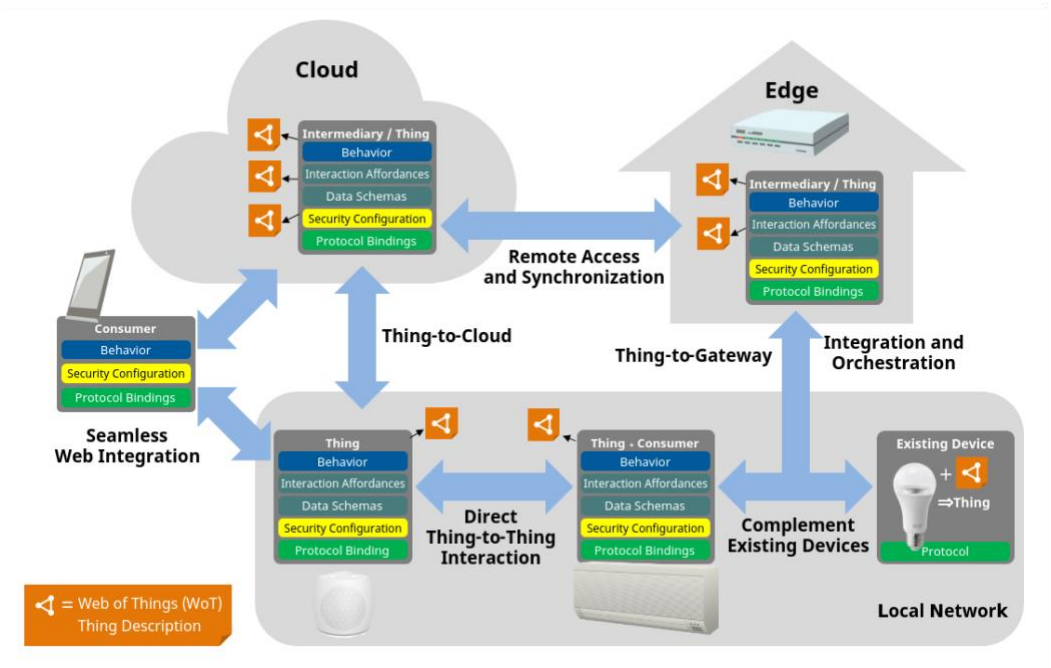

Figure 10: Web of Things Architecture, copied from https://www.w3.org/TR/wot-architecture11/#sec-wot-architecture

6.2.2.4 W3C MiniApps for IoT

A W3C MiniApp is a software application type defined by a lightweight file package that can be distributed through different digital means, including the Web, and processed and executed by specific user agents, like **super applications** (e.g., WeChat, Alipay, and Baidu) and **operating systems**. MiniApps are usually task-oriented services, targeting concrete use cases and scenarios (e.g., on-the-go orderings, transport bookings, and casual games). Since they do not require installation from native app marketplaces, MiniApps are characterised by a seamless UX and instant user interaction, improving user engagement.

Another characteristic is the resource efficiency. MiniApps are designed to consume fewer resources, including storage, memory, and access to native hardware capabilities, than traditional native apps. The convenience of the solution, efficient design, and advanced integration with popular payment methods and service ecosystems (e.g., authentication, social networking and instant messaging) make MiniApps a candidate technology for creating applications at the edge of the network.

W3C MiniApp for IoT is a new paradigm under incubation in the **W3C MiniApps Ecosystem Community Group**. This specification describes use cases and standardisation requirements to apply the MiniApps concept on top of IoT networks and devices. The specification includes scenarios where MiniApps for IoT can fit, including light applications for home appliances, networking devices, shop checkout pads, public information interactive displays, industrial control systems, and microcontroller kits.

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** (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.

6.3.2 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 ecosystem as shown in **Figure 11**:

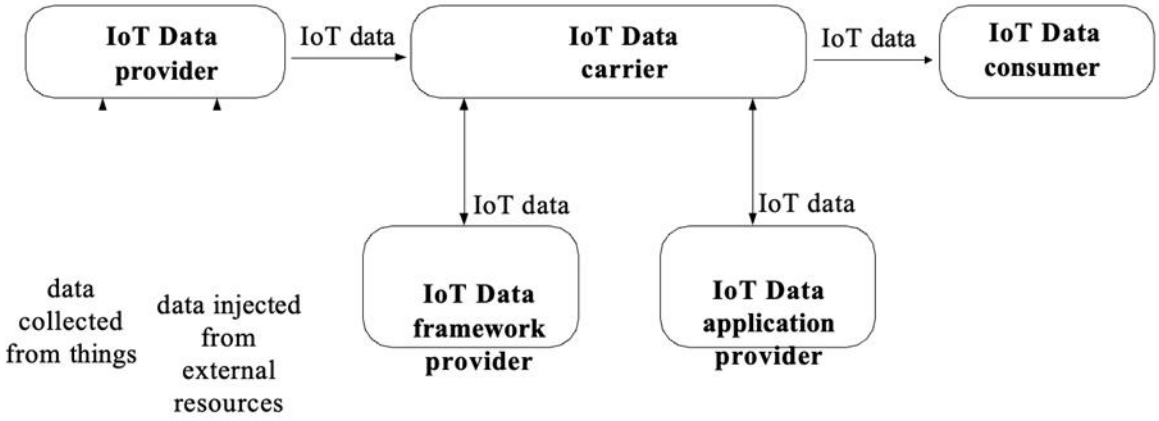

Figure 11: IoT data roles [8]

- **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.

Table 1 provides a mapping between ITU Y.4114 [8] and AIOTI HLA:

Table 1: Mapping of ITU Y.4114 to 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**

6.3.3. 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 12** provides an abstract representation of the various IoT data operations and related data flows [8].

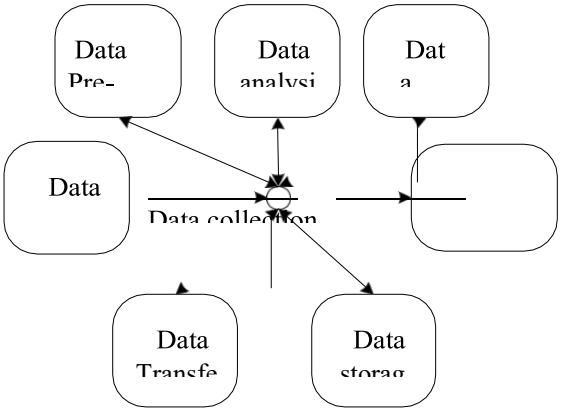

Figure 12: IoT data operations [8]

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.

6.3.4 AI enabled by Big Data

The advent of Big Data, as outlined in section 6.3.1, has significantly advanced the capabilities and applications of AI across various sectors, particularly within the context of the IoT. Big Data as a paradigm enables the extensive collection, storage, management, analysis, and visualization of datasets. These datasets are essential in understanding how AI leverages information to generate insights, automate processes, and make informed decisions.

The integration of AI into Big Data ecosystems is a significant advancement and a natural progression, given that AI depends to a large degree on large volumes of diverse, high-velocity data, which are the very characteristics identified in ITU-T Y.4114. These characteristics not only favour AI algorithms but also align with the IoT data operations detailed in section 6.3.3. These operations which include the collection, pre-processing, transfer, storage, query, analysis, and visualization of data are crucial for the seamless functioning of AI within IoT frameworks.

Based on the IoT data roles introduced in section 6.3.2, the integration of Big Data in AI becomes even more evident. IoT Data Providers collect and inject data into the system, supplying valuable input that is processed and analysed by AI algorithms. At the same time, IoT Data Application Providers, which offer applications for data analysis and visualization, serve as platforms for AI to deliver its predictive and analytical capabilities, enhancing the decision-making processes within IoT ecosystems. Furthermore, the IoT Data Framework Providers, by supplying the necessary infrastructure for data operations, enable the scalable processing and analysis capabilities required for AI applications. This collaboration underscores the critical role of AI in enhancing the operational efficiency and capabilities of IoT systems through Big Data analytics.

Moreover, the integration of AI with Big Data directly impacts the sequencing of IoT data operations, as discussed in section 6.3.3. AI can optimize the flow of data through predictive analytics and intelligent automation, thereby enhancing the capabilities of both cloud and edge computing in processing and analysing data in (near) real-time. This optimization is crucial for managing the high-velocity data streams characteristic of IoT devices and systems, ensuring timely and effective decision-making.

Nevertheless, the convergence of Big Data and AI may also necessitate a re-evaluation of data privacy, security, and governance strategies. As AI systems are increasingly employed to analyse Big Data within IoT contexts, the principles of data veracity and value, as highlighted in section 6.3.1, become central to maintaining the integrity of AI-generated insights. Ensuring the ethical use, transparency, and protection of Big Data in AI applications is paramount to sustaining trust and mitigating risks associated with data misuse.

In essence, AI enabled by Big Data is not only a technological advancement but a transformative force that leverages the Big Data principles, roles, and operations. As we continue to explore and expand upon the capabilities of AI within the Big Data ecosystem, it is imperative to adhere to the guidelines and insights provided in sections 6.3.1, 6.3.2, and 6.3.3, thereby ensuring a coherent, consistent, and forward-looking approach to AI and Big Data integration within IoT and beyond.

6.3.5 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 were 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.

The following standards have been developed related to Big data reference architecture:

- ISO/IEC TR 20547-1 Information technology -- Big data reference architecture -- Part 1: Framework and application process
- ISO/IEC TR 20547-2:2018 Information technology -- Big data reference architecture -- Part 2: Use cases and derived requirements
- ISO/IEC 20547-3:2020 Information technology -- Big data reference architecture -- Part 3: Reference architecture
- ISO/IEC 20547-4:2020 Information technology — Big data reference architecture — Part 4: Security and privacy
- ISO/IEC TR 20547-5:2018 Information technology -- Big data reference architecture -- Part 5: Standards roadmap and Artificial Intelligence:
- ISO/IEC 22989:2022 Artificial Intelligence Concepts and Terminology
- ISO/IEC 23053:2022 Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)

Relevant working groups include:

- ISO/IEC JTC 1/SC 42/WG 1 - Foundational standards
- ISO/IEC JTC 1/SC 42/WG 2 - Data
- ISO/IEC JTC 1/SC 42/WG 3 - Trustworthiness
- ISO/IEC JTC 1/SC 42/WG 4 – Use cases and applications
- ISO/IEC JTC 1/SC 42/WG 5 – Computational approaches and computational characteristics of AI systems

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).

6.4 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 lawmakers 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.

More recently, the European Union significantly enhanced its legislative framework with the introduction of **NIS2**, the **Cybersecurity Act**, and the **Cybersecurity Resilience Act** (CRA). NIS2, being a revision of the original Network and Information Systems Directive, extends its reach by encompassing a wider array of sectors and digital entities, thereby imposing stricter security and incident reporting requirements. The Cybersecurity Act strengthens the mandate of the European Union Agency for Cybersecurity (ENISA), facilitating enhanced coordination across Member States and introducing a unified cybersecurity certification scheme for ICT products, services, and processes. The CRA aims to enhance the resilience of critical infrastructure, ensuring that such entities are better prepared to withstand and recover from disruptive incidents. Collectively, these measures aim to elevate the EU's cybersecurity posture through comprehensive coverage of preventive, detective, and responsive capabilities across critical and digital service providers.

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.

6.5 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.

6.6 Virtualization

6.6.1 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 13** 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.

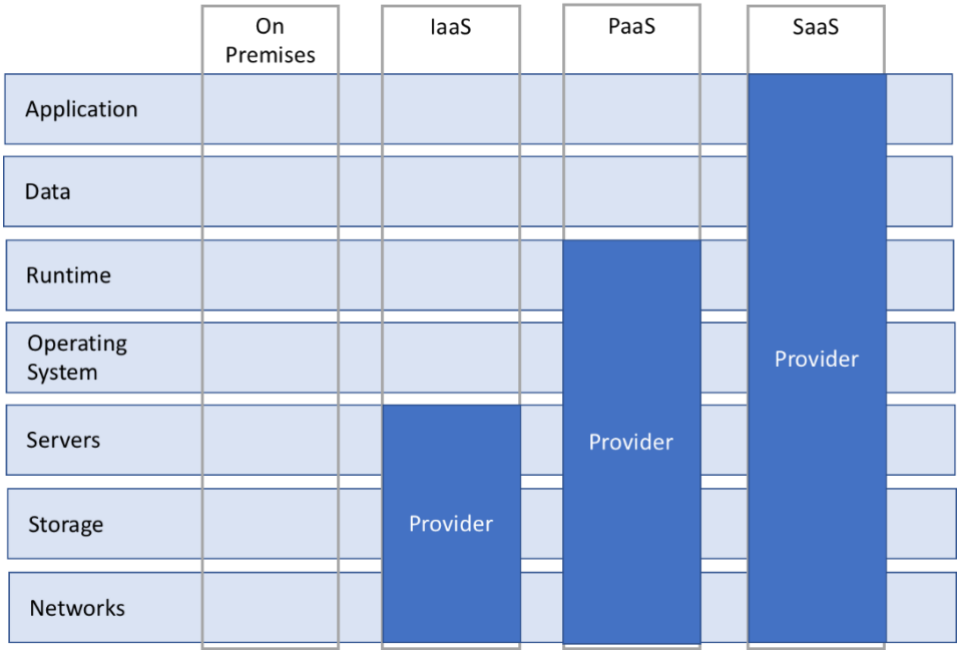

Figure 13: The potential of Cloud Computing Service Models

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).

6.6.2 Approaches to IoT Virtualization

Three approaches are outlined below.

The first one (see clause 6.6.2.1) 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 6.6.2.2) is using a "standards-based" approach and seeks the adaptation of the virtualization technologies coming from Cloud Computing. A third approach (see clause 6.6.2.3) 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. The fourth approach is device virtualisation (see clause 6.6.2.4) which describes one way of virtualizing the IoT devices, where devices may be highly resource constrained or not.

6.6.2.1 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 14** provides an example of such system.

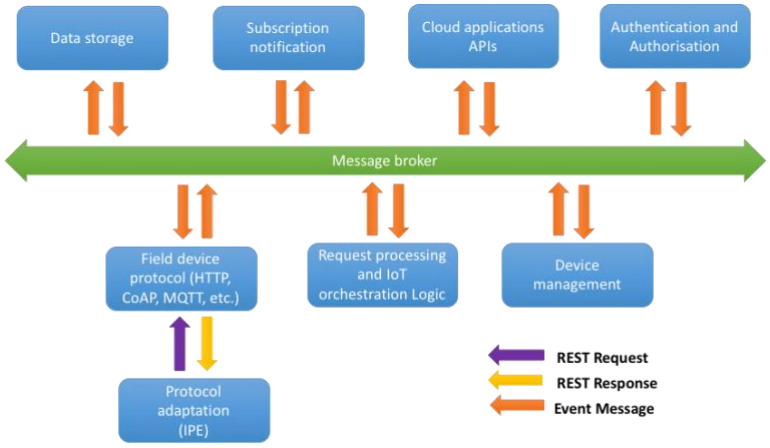

Figure 14: Microservices conceptual framework for IoT Virtualization

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 15 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.

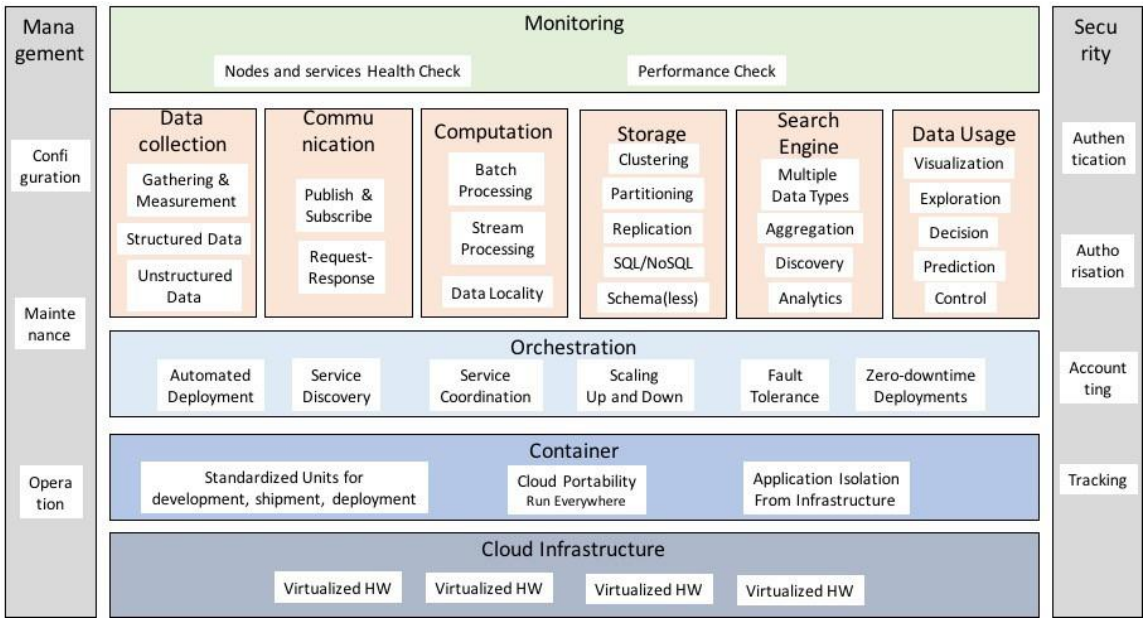

Figure 15: A microservices-based functional architecture for IoT Virtualization

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].

6.6.2.2 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 16** 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.

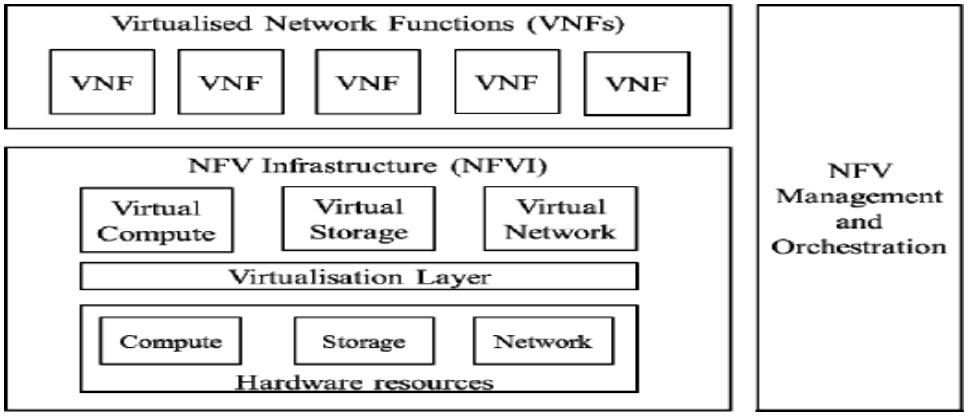

Figure 16: High Level NFV Framework

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.

6.6.2.3 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 17**.

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.

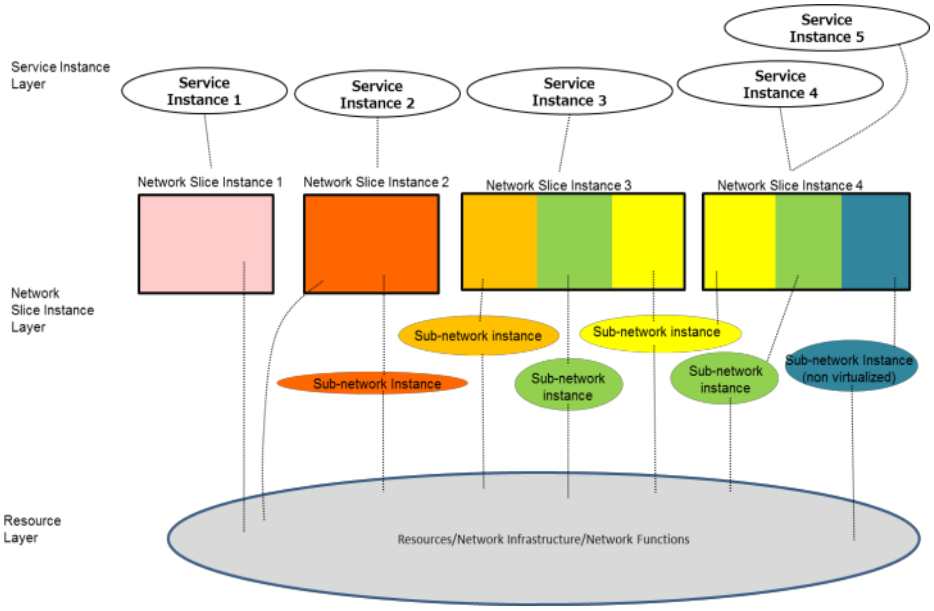

Figure 17: NGMN Network Slicing conceptual outline [10]

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 WG Standardisation 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.

6.6.2.4 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 [35].

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 [36]. As shown in **Figure 18**, the "IoT Daemon" encompasses the above abstractions. This way, multitudes of IoT devices can be integrated and interfaced.

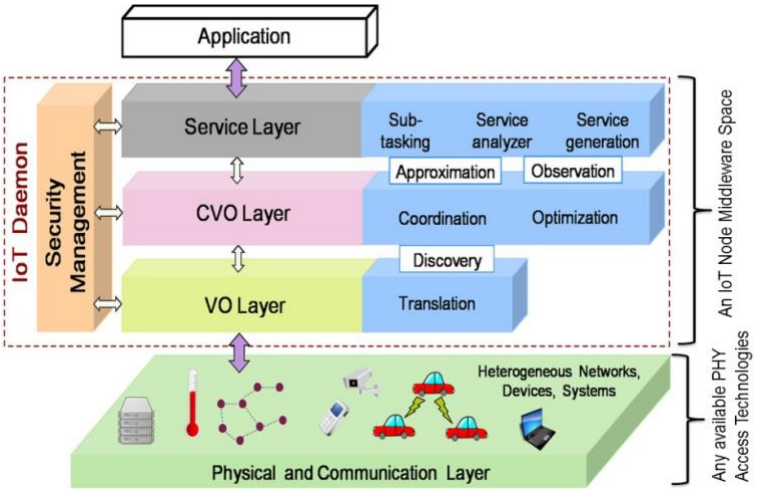

Figure 18: A high level architecture of (Composite) Virtual Objects

NOTE: 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.

Figure 18 provides an abstract view of the interfaces between VO, CVO and Service layer functionalities.

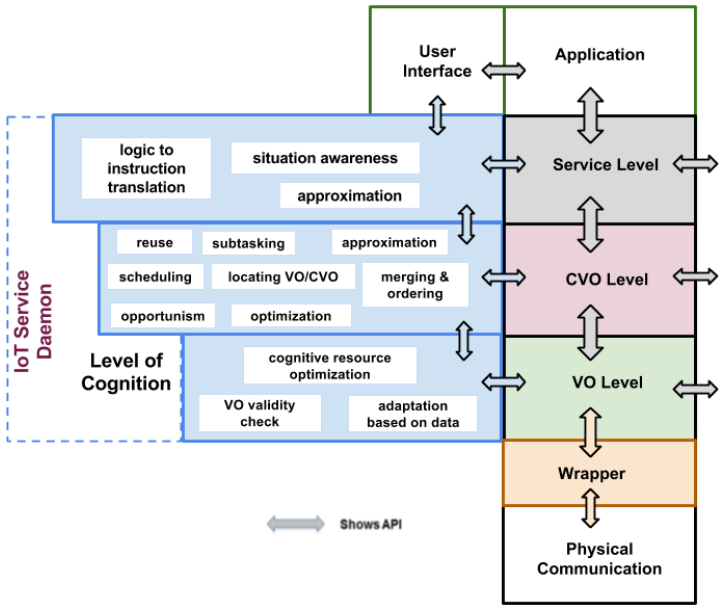

Figure 19: IoT device architecture and interfaces between the different layers

With respect to the two virtualization approaches described in clauses 6.6.2.1 and 6.6.2.2, 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 6.6.2.1 and 6.6.2.2. 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.

6.6.3 Comparing the IoT virtualization approaches

This clause is comparing the approaches described in clause 6.6.2.

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 20 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. The virtualization layer only has access to the virtual devices as supplied by the device 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.

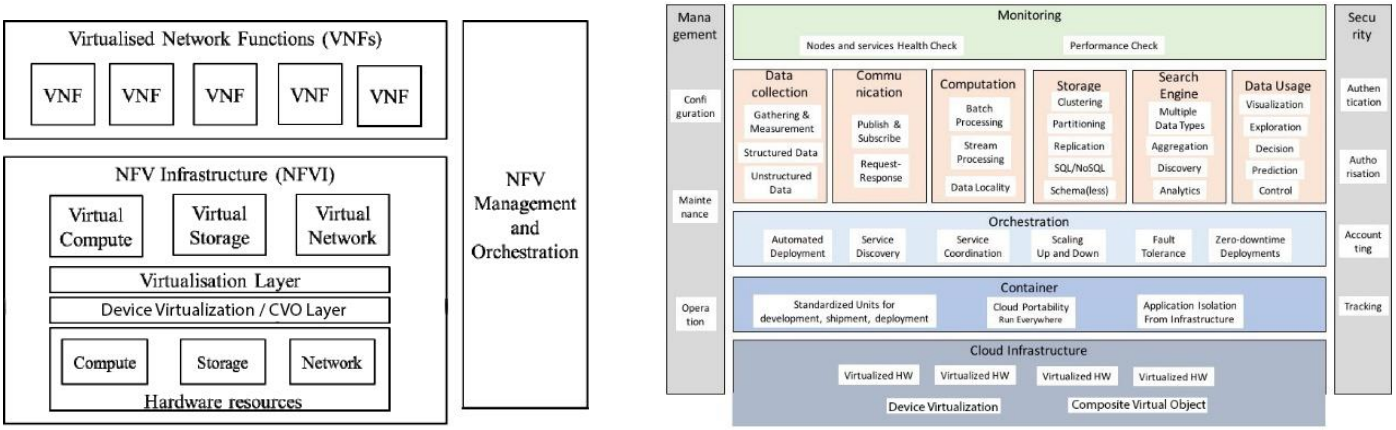

Figure 20: How Device Virtualization and Composite Virtual Objects can be leveraged by other approaches

6.6.4 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.

6.6.4.1 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 21**.

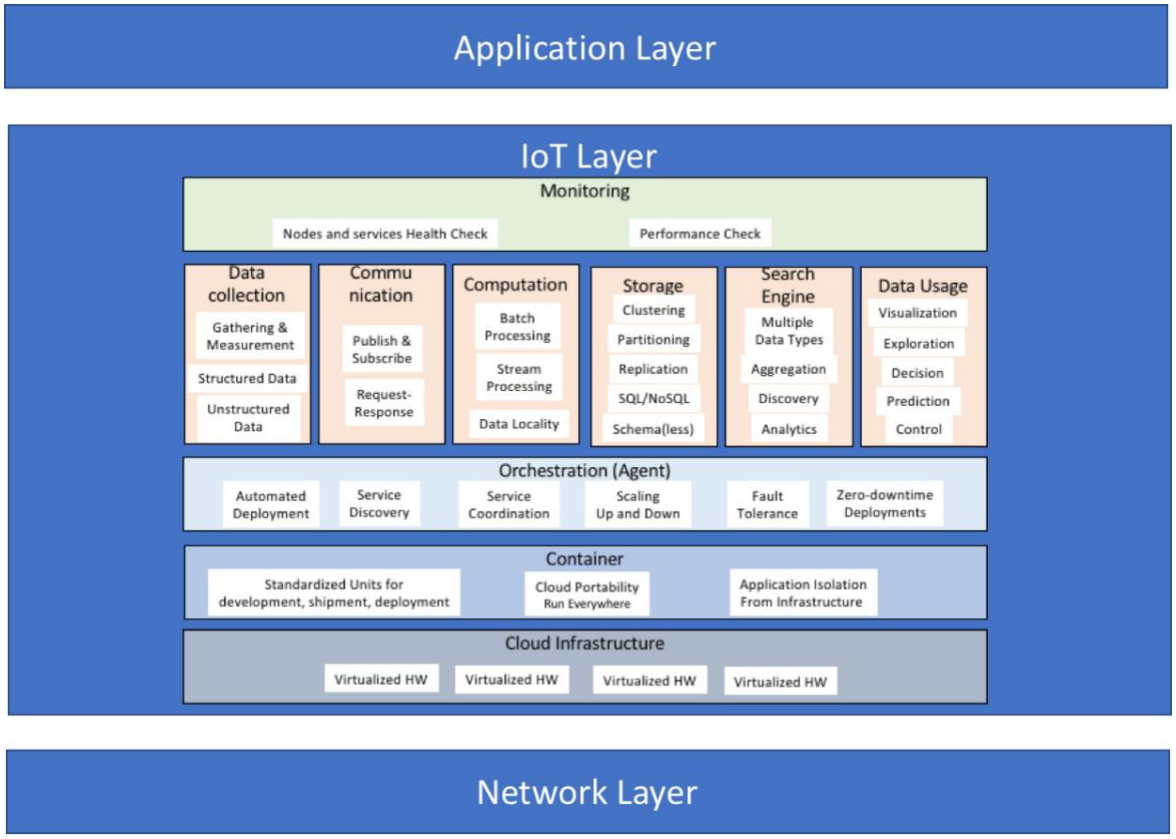

Figure 21: Mapping of microservice-based functional architecture on AIOTI HLA

6.6.4.2 The mapping of a microservices-based functional architecture on the 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 6.6.2.1 is shown in **Figure 22**:

- 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 22** 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.

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.

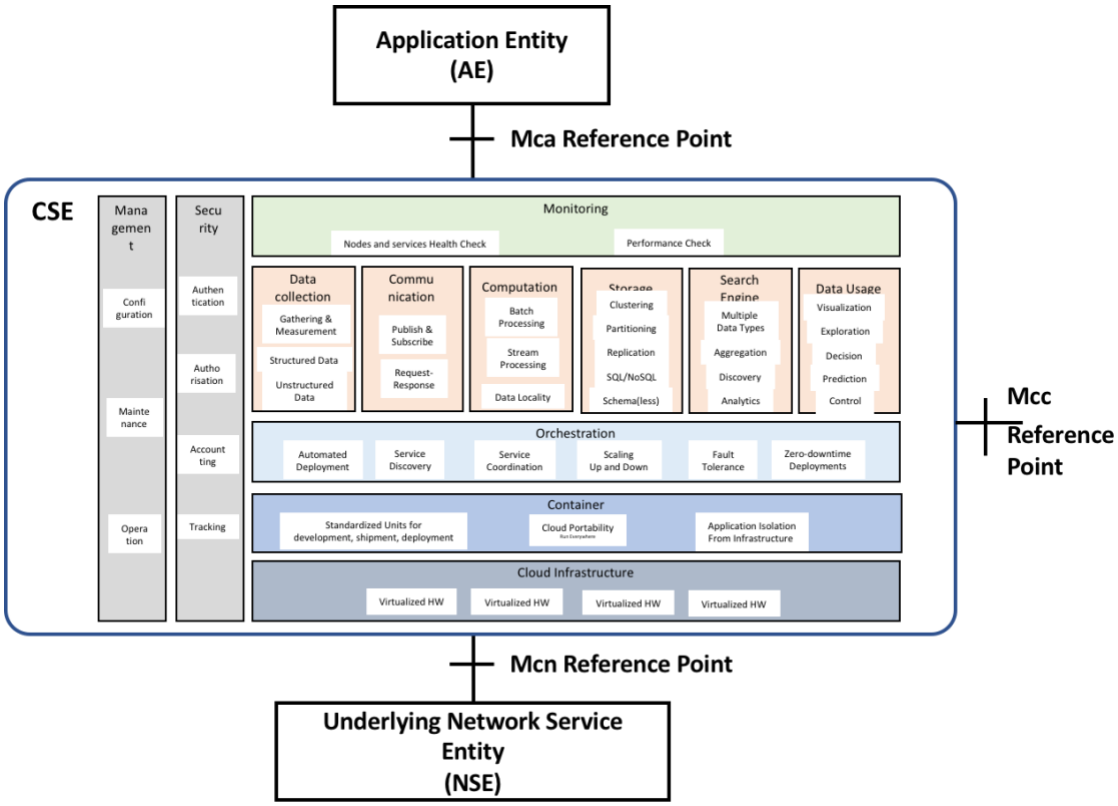

Figure 22: Mapping of microservices-based functional architecture on oneM2M Common Service Entities

6.5 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.

6.7.1 Generalities on IoT platforms

6.7.1.1 IoT platform to platform interoperability

There are several approaches that cover IoT platform interoperability, such as:

- **ETSI STF 547 TR 103536**
- Data lake approach (for example, **IBM Data Lake**)
- Direct integration between IoT platforms (for example, **Emnify IoT SuperNetwork**)

Some other approaches are described in more details below.

6.7.1.1.1 Approach: usage of intermediate standardized platform

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.

6.7.1.1.2 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 **Figure 23**.

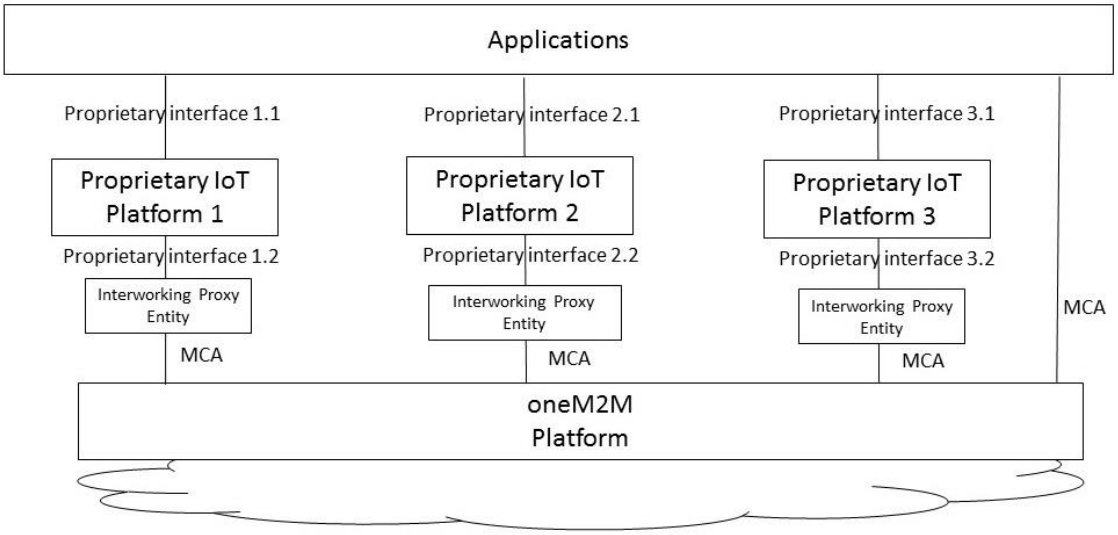

Figure 23: AUTOPILOT Federated IoT Architecture

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.

6.7.1.1.2.1 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.

6.7.1.1.2.2 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 23**.

The left part of **Figure 24** shows the oneM2M IoT platform compliant to AIOTI HLA (identical to the one shown in **Figure 27**) and the right part shows (an IoT platform compliant to) the AIOTI HLA as shown in **Figure 4**. As additional entity, **Figure 24** shows the Interworking Proxy Entity, a specialized oneM2M AE (Application Entity) that allows the oneM2M system to interact with any non-oneM2M (Proprietary) system.

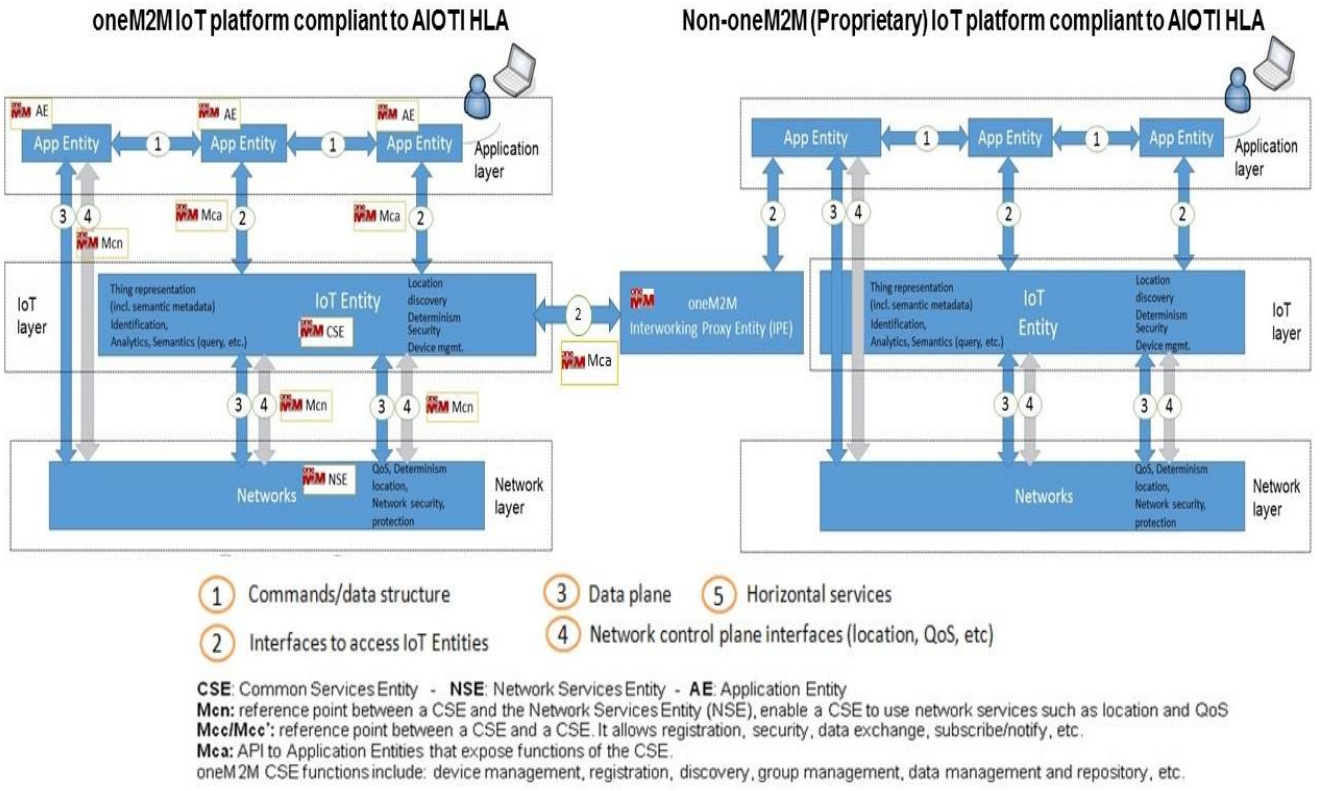

Figure 24: oneM2M IoT Platform Interoperability with AIOTI HLA-compliant IoT platform

6.8 Data Spaces in the AIOTI High-Level Architecture

6.8.1 Data Spaces

While the term **data space** was coined more that 10 years ago[2], it was not until recent years that a number of position papers such as BDVA[3,4], OpenDei[5], and initiatives, such as IDSA[6], or GAIA-X[7,8] or FIWARE[9] have started to propose a common understanding.

OpenDei provides a comprehensive definition:

*From a technical perspective, a **data space** can be seen as a data integration concept which does not require common database schemas and physical data integration but is rather based on distributed data stores and integration on an "as needed" basis on a semantic level.*

[2] https://en.wikipedia.org/wiki/Dataspaces
[3] Towards a European-Governed Data Sharing Space. Enabling data exchange and unlocking AI potential. April 2019 https://bdva.eu/sites/default/files/BDVA%20DataSharingSpace%20PositionPaper_April2019_V1.pdf
[4] Towards a European-Governed Data Sharing Space. Enabling data exchange and unlocking AI potential. November 2020 https://www.bdva.eu/sites/default/files/BDVA%20DataSharingSpaces%20PositionPaper%20V2_2020_Final.pdf
[5] https://design-principles-for-data-spaces.org/
[6] https://www.internationaldataspaces.org/wp-content/uploads/2019/03/IDS-Reference-Architecture-Model-3.0.pdf
[7] https://www.data-infrastructure.eu/GAIAX/Redaktion/EN/Publications/gaia-x-technical-architecture.pdf?__blob=publicationFile&v=5. Release-June 2020
[8] https://www.data-infrastructure.eu/GAIAX/Redaktion/EN/Publications/gaia-x-technical-architecture.pdf
[9] https://www.fiware.org/marketing-material/fiware-for-data-spaces -(release June 2021)

Abstracted from this technical definition, a data space can be defined as a federated data ecosystem within a certain application domain and based on shared policies and rules

FIWARE provides a definition which is aligned:

*A **data space** can be defined as a decentralized data ecosystem built around commonly agreed building blocks enabling an effective and trusted sharing of data among participants.*

In this position paper, we will assume that a data space is a **trustworthy decentralized environment for data sharing**.

Decentralisation is a particularly important characteristic as showed in **Figure 25**. It provides an example of data spaces with five organisations engaged in carrying out operations on data. The figure highlights

- two layers: the processing layer, and the data layer; and
- three concepts
- data exchanges: a relationship that involves organisations,
- data interoperability: a capability between processing systems, and
- data operations: activities carried out by processing systems.

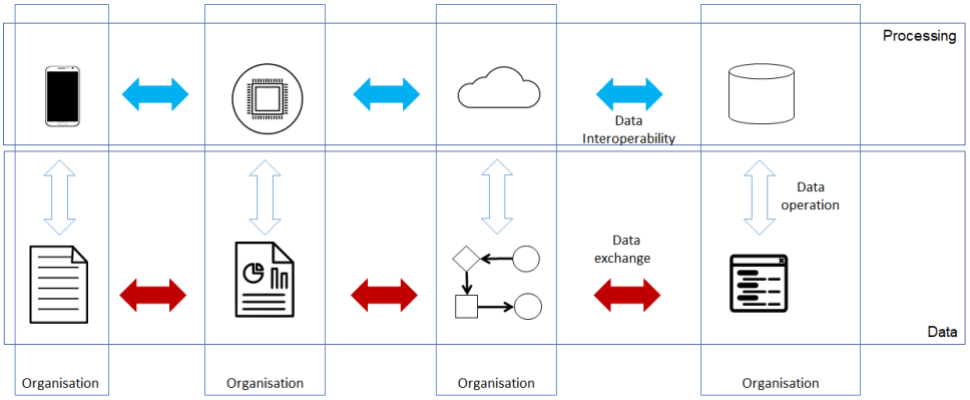

Figure 25: Decentralised data space example

6.8.2 Data Spaces using the HLA representation

The following mapping of the Data Spaces to the HLA representation is based on the information provided in the AIOTI report "Guidance for the Integration of IoT and Edge Computing in Data Spaces", Release 1. In particular, **Figure 26** shows the data space example described in **Figure 25**, using the HLA representation. The difference is the addition of the network layer which puts emphasis on interoperability properties.

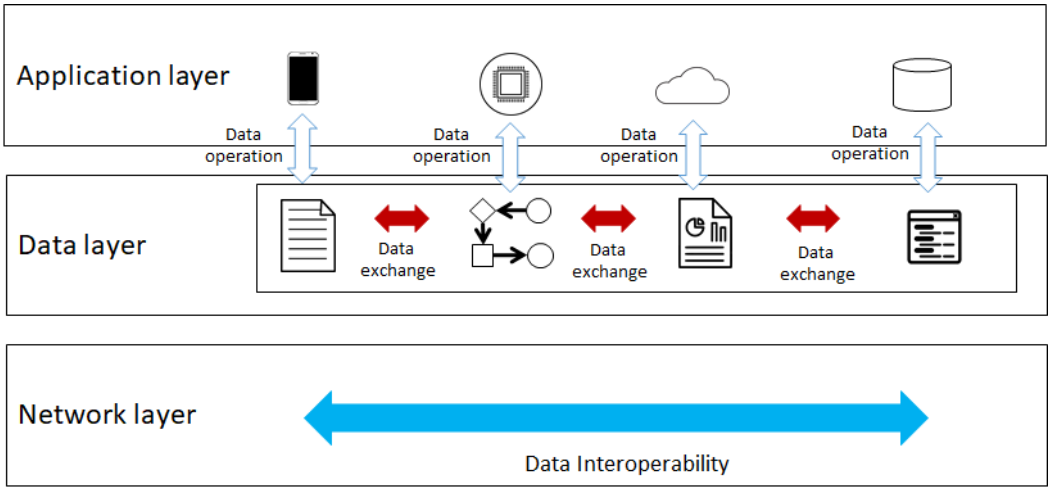

Figure 26: Data space example using the HLA representation

6.8.3 Digital Twin using the HLA representation

The two layers (processing layer, data layer) and the three concepts (data exchange, data interoperability and data exchange) can be used to illustrate data spaces in various configurations. Figure 26 provides an example illustrating AI capability in a digital twin:

- the processing layer focuses on knowledge handling and reasoning, while the data layer focuses on and knowledge representation and storage;
- data exchange takes place between the handling and reasoning capabilities of the virtual entity and the same capabilities of the physical entity;
- data interoperability is enabled by knowledge representations agreed between the virtual and the physical entity and
- data operations are carried out handling and reasoning capabilities of the virtual entity and the same capabilities of the physical entity.

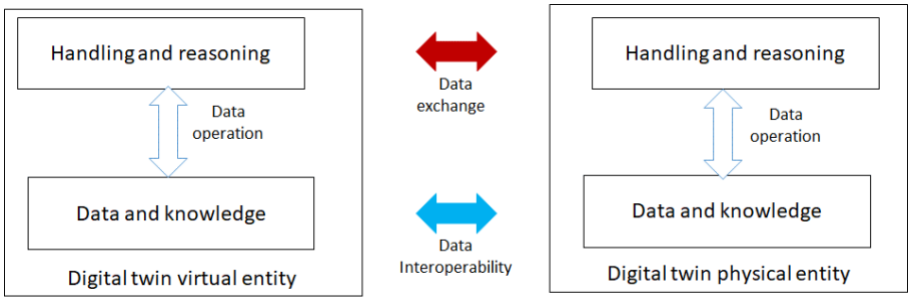

Figure 27: AI capability in a digital twin example

Figure 28 shows the digital twin example described in **Figure 27** using the HLA representation (depicted in Figure 4).

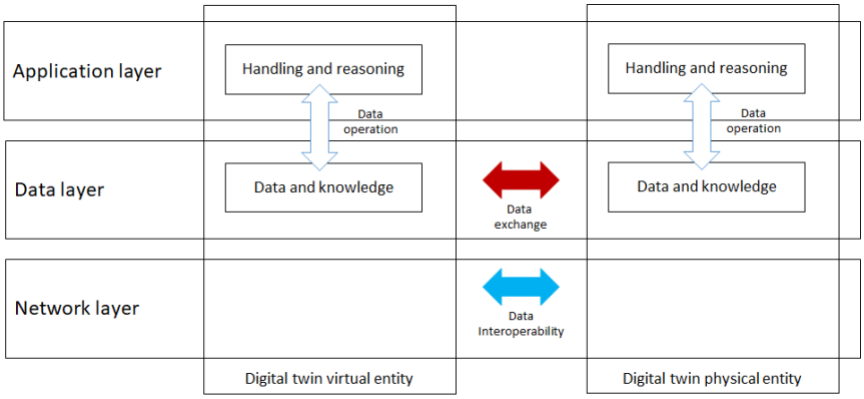

Figure 28: HLA representation of digital twin example

6.8.4 Computing Continuum Perspective

A computing continuum perspective integrating IoT and edge computing is needed. **Figure 29** shows a data spaces where this continuum is visualised from left to right:

- IoT devices carry out some data operations and exchange data,
- Edge systems carry out further data operations and exchange further data,
- Cloud systems carry out further data operations and exchange further data

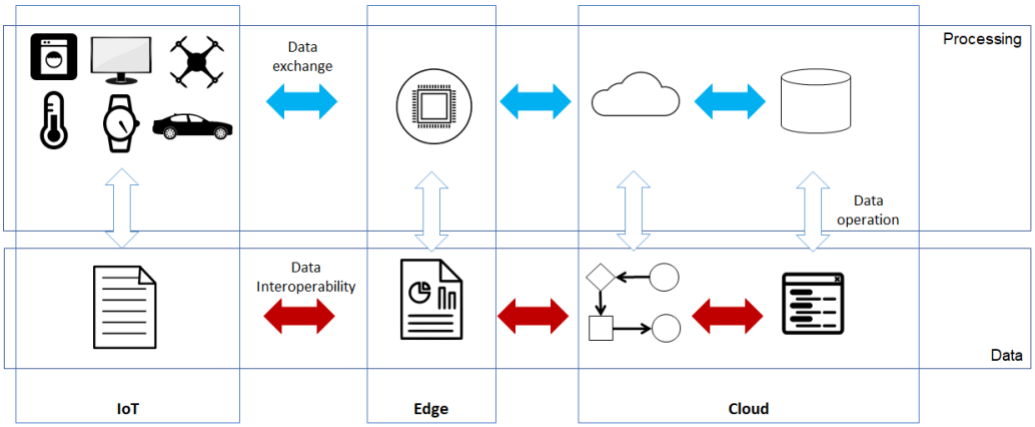

Figure 29: Computing continuum perspective of data spaces

Figure 30 shows the same computing continuum perspective using the HLA representation.

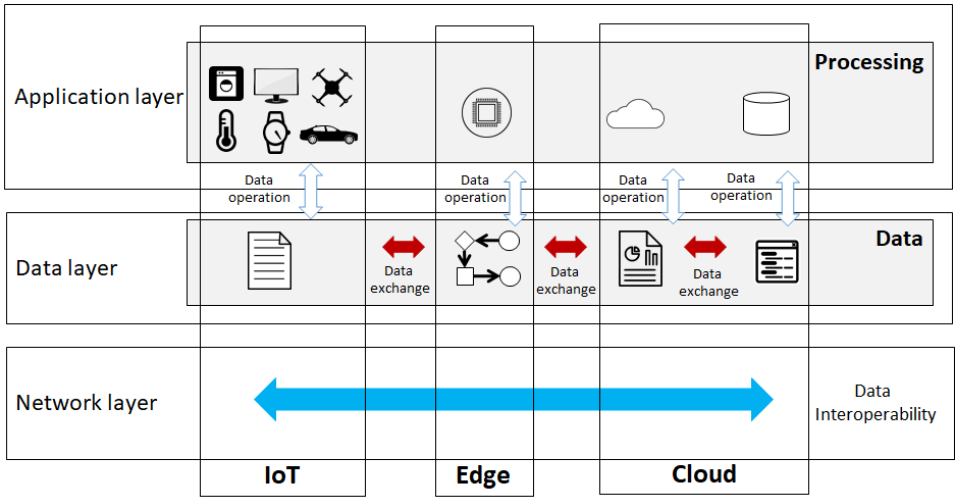

Figure 30: Computing continuum perspective of data spaces based on HLA

6.8.5 Federated Systems Perspective

A federated system perspective can also be needed. **Figure 31** shows this: while data exchange can take place within a data space ecosystem, two separate ecosystems can also exchange data. Federation is suitable in particular to achieve cross domain exchange e.g. between the energy and the transport domain as shown in **Figure 32**.

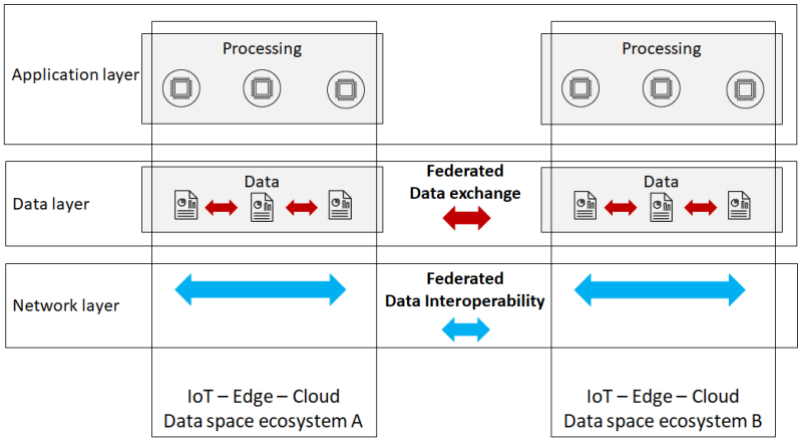

Figure 31: Federated systems perspective of data spaces

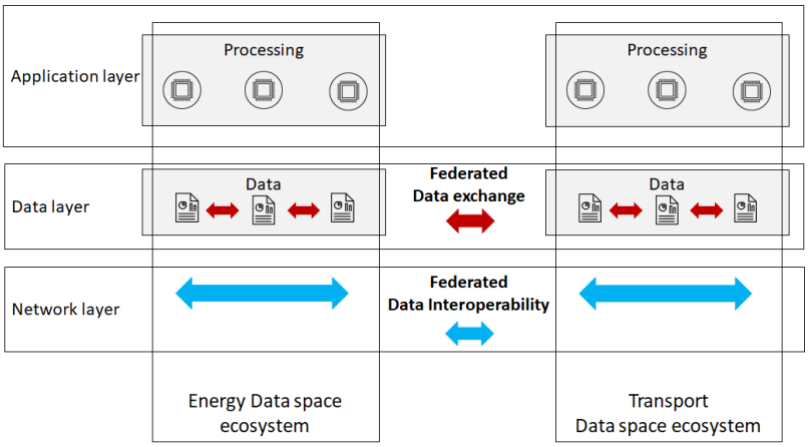

Figure 32: Domain perspective of data spaces

6.8.6 Data Collecting and Trading Perspective

A data marketplace perspective can also be needed. **Figure 33** shows a data collecting system, a data trading system, consisting of a market place, data providers and data consumers.

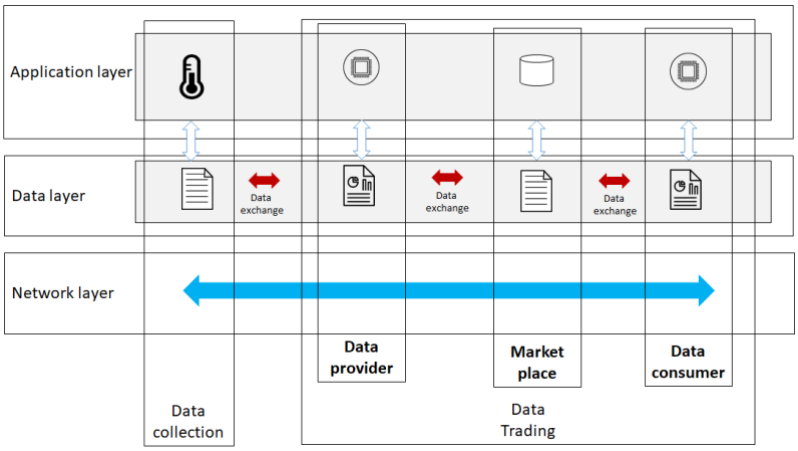

Figure 33: Data collecting system and data marketplace

7 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.

7.1 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 34**, 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.

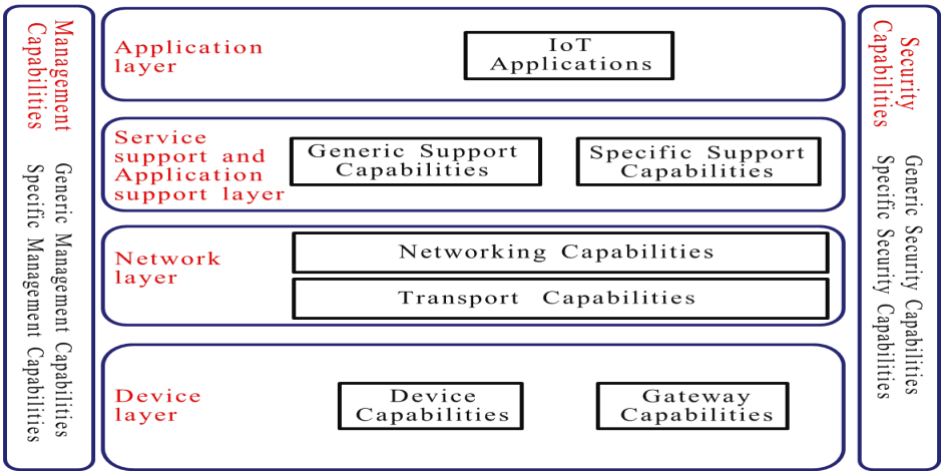

Figure 34: ITU-T Y.4000 IoT Reference Model

Figure 35 provides an initial high level mapping of the ITU-T Y.4000 IoT Reference model to AIOTI HLA functional model.

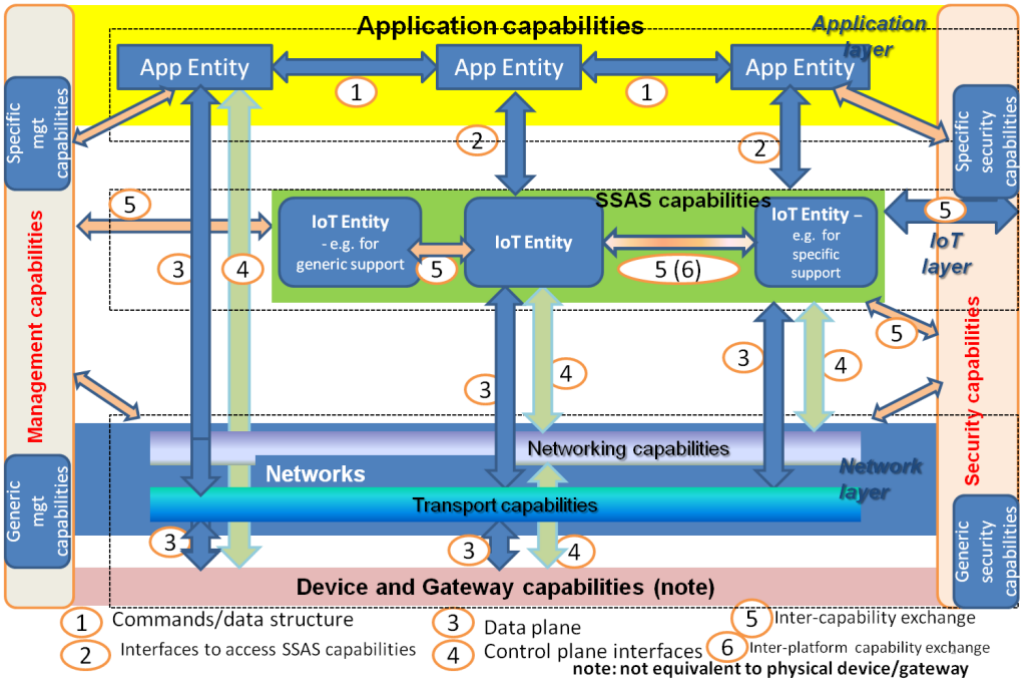

Figure 35: ITU-T IoT Reference Model mapping to AIOTI HLA functional model

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").

7.1.1 ITU-T Coordination of Networking and Computing (CNC)

ITU-T is currently developing specifications on the coordination of networking and computing in IMT-2020 networks and beyond (CNC). The topic, while not specific to IoT, aims to address the network's simultaneous support of critical service requirements on computing, networking and storage resources raised by the emergence of new services and applications, such as - but not limited to - cloud virtual reality, Internet of vehicles services, large-scale scientific data applications.

The coordination among resources of the same or different types is necessary: by applying coordination of utilization, computing control and management, storage and networking for the purposes of provisioning and optimization, the satisfaction of requirements of resources users, and the improvement of resources utilization, may be achieved.

Recommendation ITU-T Y.3400 "Coordination of networking and computing in IMT-2020 networks and beyond – Requirements" - approved by ITU-T SG13 in December 2023:

specifies the requirements for CNC, with requirements in the areas of measurement of resources, identification and addressing of resources, awareness of resources, joint scheduling of resources, unified management and orchestration of resources, resource transaction, energy saving, QoS assurance, fixed, mobile and satellite convergence, intelligence and automation, security and privacy.

Appendix I of Y.3400 illustrates some relevant application scenarios for CNC.

Various CNC related draft Recommendations are also under progress within SG13, including:

Y.IMT2020-CNC-FW "Coordination of networking and computing in IMT-2020 networks and beyond – Capability framework";

Y.IMT2020- CNC-RS "Future networks including IMT-2020 - Requirements and capability framework of resource scheduling for coordination of networking and computing";

Y.IMT2020-QoS-cnc-req "QoS assurance-related requirements and framework for computing and network convergence supported by IMT-2020 and beyond";

Y.M&O-CNC-fra "Management and orchestration related requirements and framework for coordination of networking and computing in IMT-2020 networks and beyond";

Y.FMSC-CNC "Fixed, mobile and satellite convergence - Coordination of networking and computing for IMT-2020 networks and beyond".

Other ITU-T SG13 specification efforts in relation with the coordination of networking and computing (but focused on NGNe (NGN evolution)) include:

Y.2501 "Computing Power Network - framework and architecture";

Y.2502 "Computing power network - Authentication and orchestration architecture";

Y.CPN-CL-Arch "Requirements and architecture of CPN control layer for network resource in NGNe";

Y.CPN-exp-reqts "Requirements of capability exposure in CPN";

Y.CPN-TP-Arch "Requirements and functional architecture of computing power network transaction platform";

Y.Suppl.CPN- UC "Use cases of Computing power network in NGNe to Y.2500 series";

Y.Suppl.CPN-EF "Expectations and information flows of resource awareness and service orchestration in computing power network".

NOTE: The status of all ongoing ITU-T work items can be accessed (by authorized users) **here**.

7.2 oneM2M

Figure 36 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).

oneM2M has specified all interfaces depicted in **Figure 36** 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.

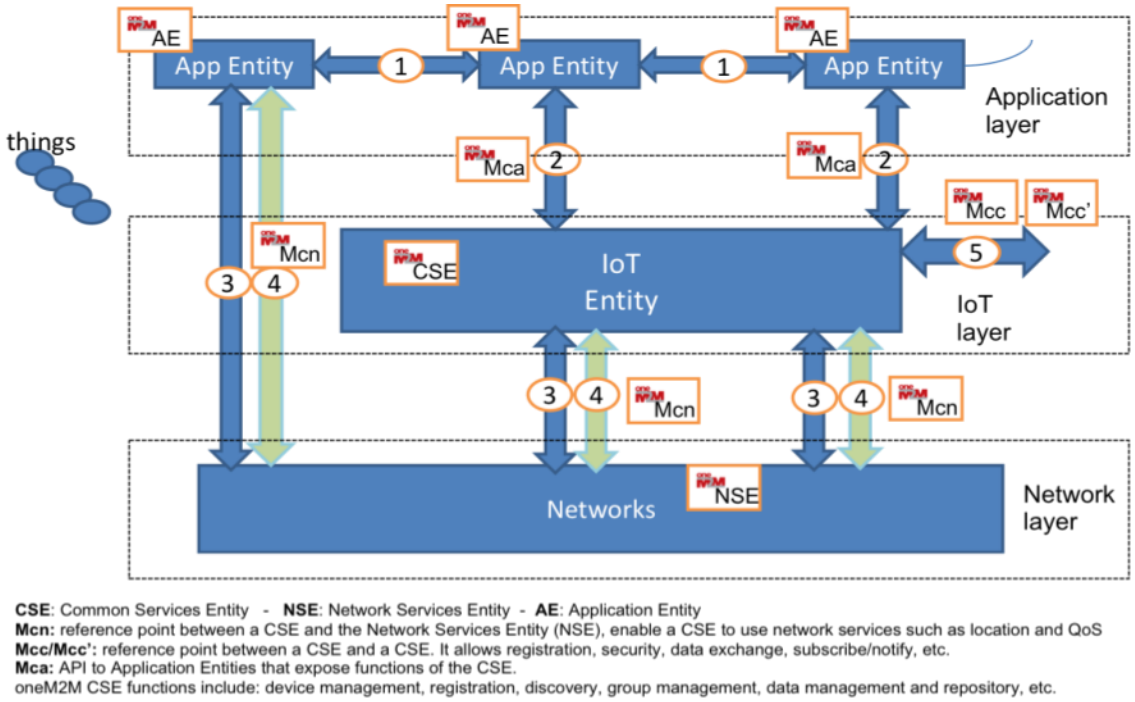

Figure 36: Mapping oneM2M to AIOTI HLA

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 37 provides a three-tier architecture as specified in [5].

Figure 37: IIC Three-Tier IIoT System Architecture

It is important to be noticed that in 2019, the IIC and the OpenFog Consortium were combined to focus on Industrial IoT, fog and edge computing. The organizations have been working together to drive the momentum of the industrial internet, including the development and promotion of industry guidance and best practices.

The OpenFog Architecture defined a system-level architecture to extend elements of computing, networking and storage across the cloud through to the edge of the network. The OpenFog architecture, shown in **Figure 38** that is now integrated in the IIC architecture s argued to serve use cases that cannot be served with centralised "cloud only" approach.

The OpenFog Consortium, was 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.

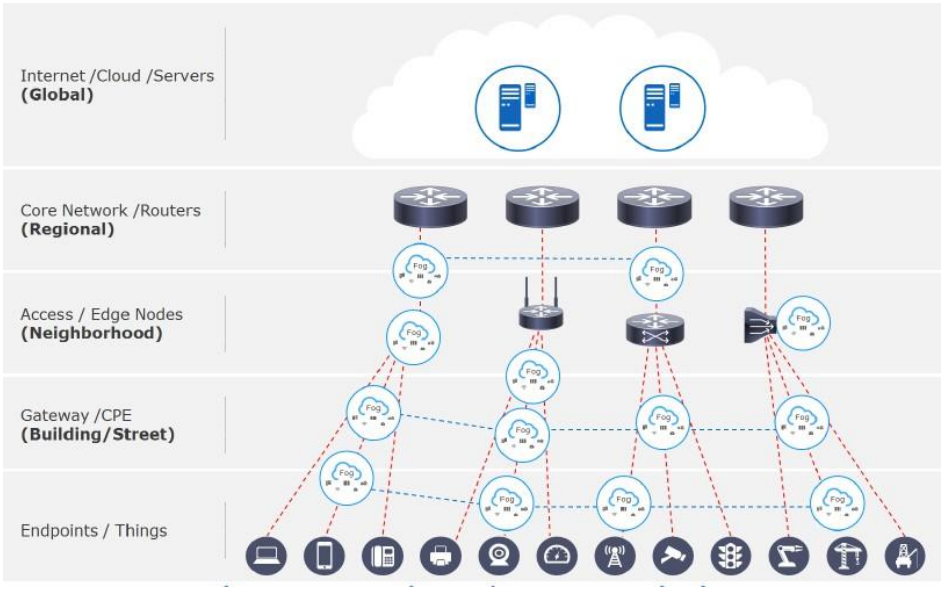

Figure 38: OpenFog cloud hierarchy

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

The IIC edge Computing Publications are listed below, which include as well the publications that had been initiated by the OpenFog consortium:

- OpenFog Reference Architecture
- OpenFog Reference Architecture Executive Summary
- Fog and IoT: An Overview of Research Opportunities
- From Cloud to Fog and the Internet of Things
- Fog Computing and its Role in the Internet of Things
- Fog Networks
- Fog Networking: An Overview of Research Opportunities

- Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are
- OpenFog Security Requirements and Approaches

The mapping of IIC to the AIOTI HLA is depicted in the following **Figure 39**.

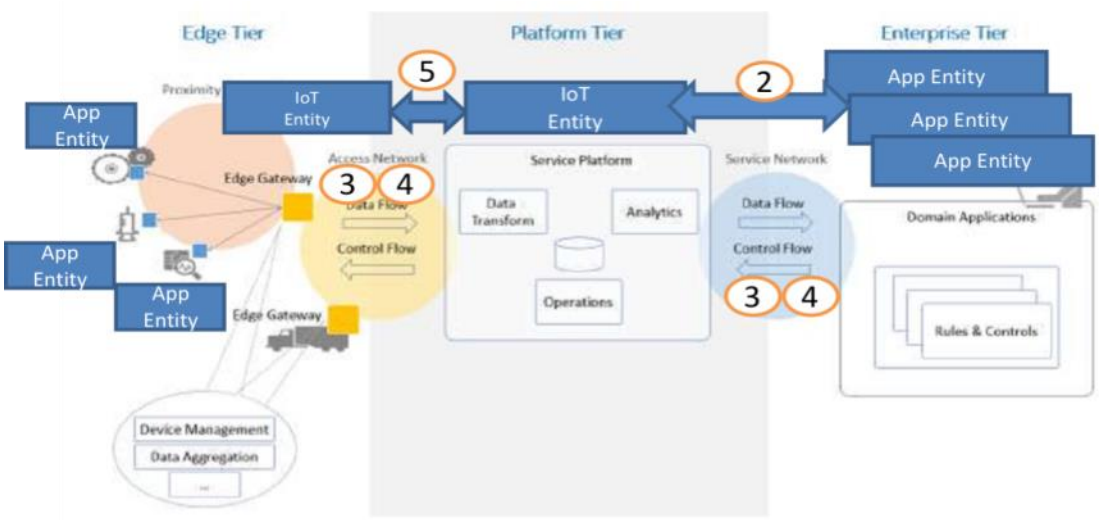

Figure 39: Mapping HLA to IIC three tier IIS architecture

In **Figure 39**, 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.

The reference architecture model as shown in **Figure 40** 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.

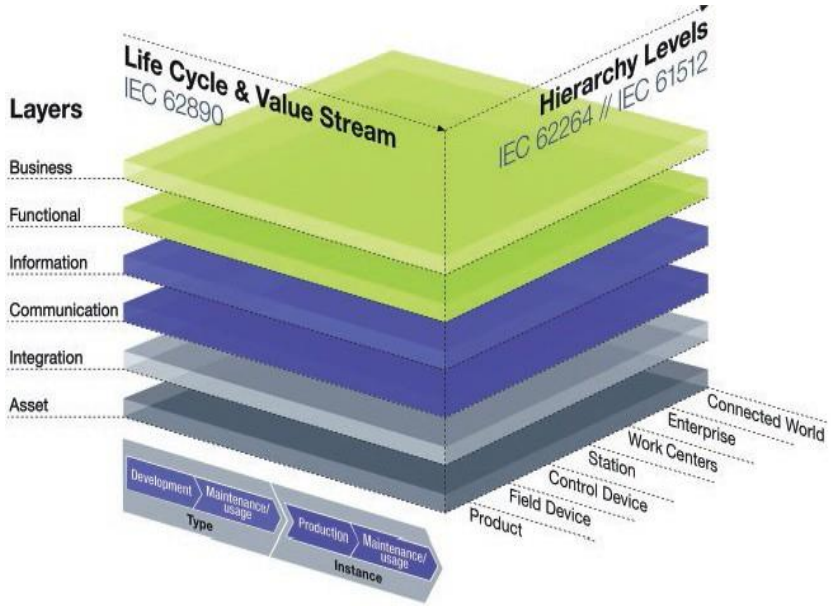

Figure 40: RAMI 4.0 reference architecture

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

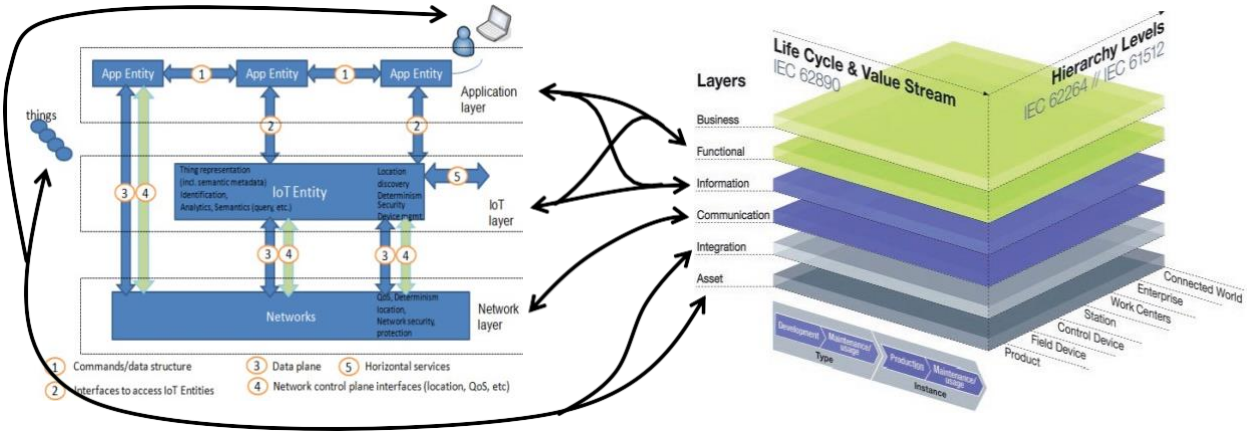

Figure 41: Mapping RAMI 4.0 to AIOTI HLA – functional model

The following explanations can be made as regards **Figure 41**:

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 42**.

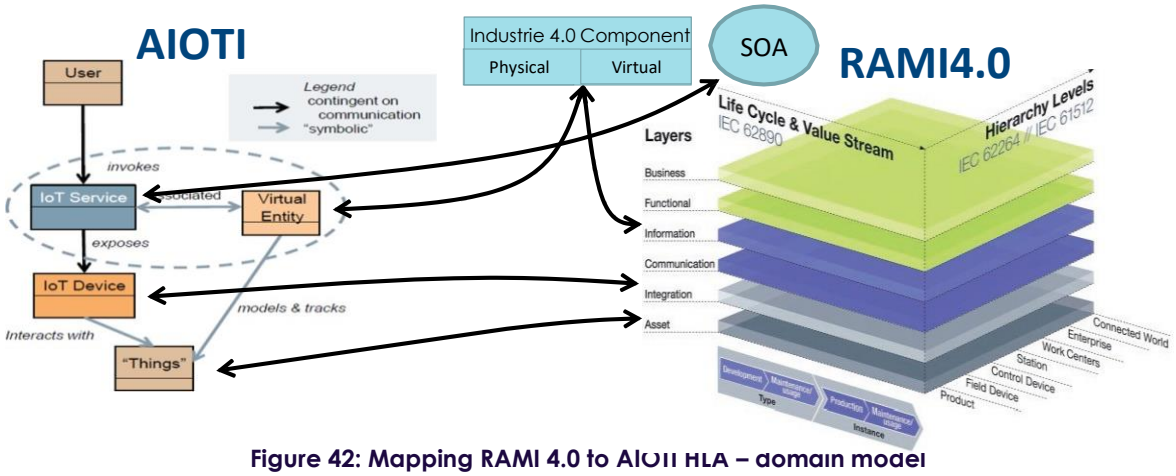

Figure 42: Mapping RAMI 4.0 to AIOTI HLA – domain model

The following explanations can be made as regards **Figure 42**:

- 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.

7.5 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.

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 and beyond), 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.

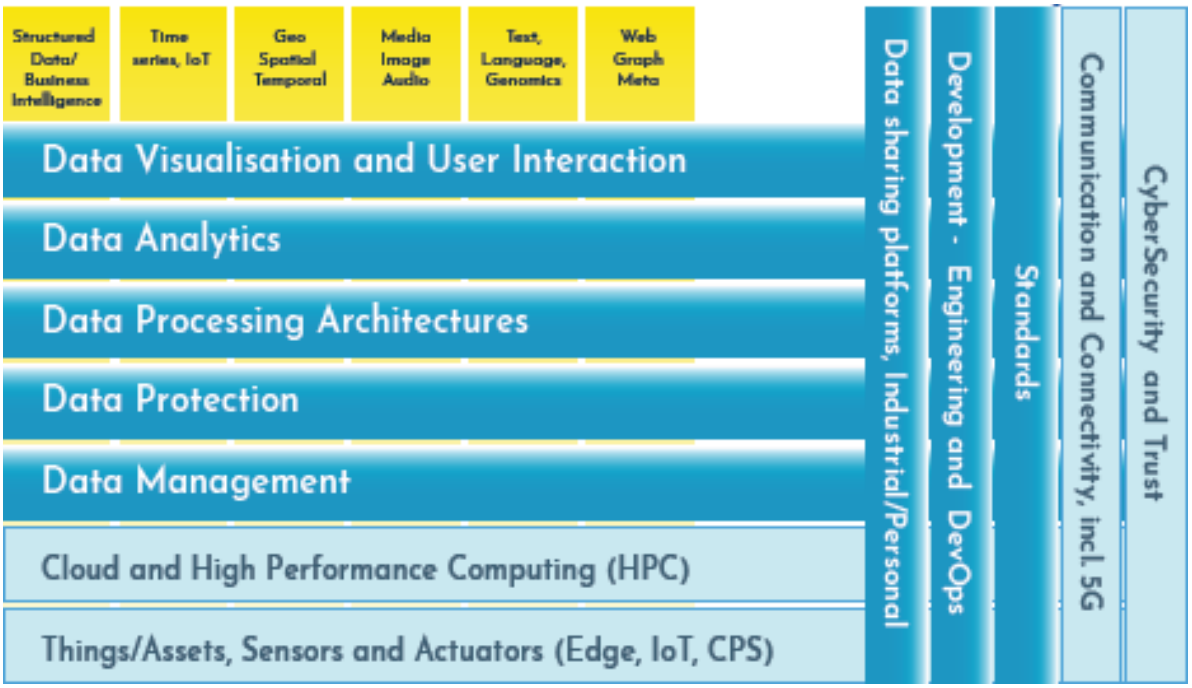

Figure 43: Big Data Value Association – BDV Reference Model

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.

7.5.1 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 WG Standardisation 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 44**.

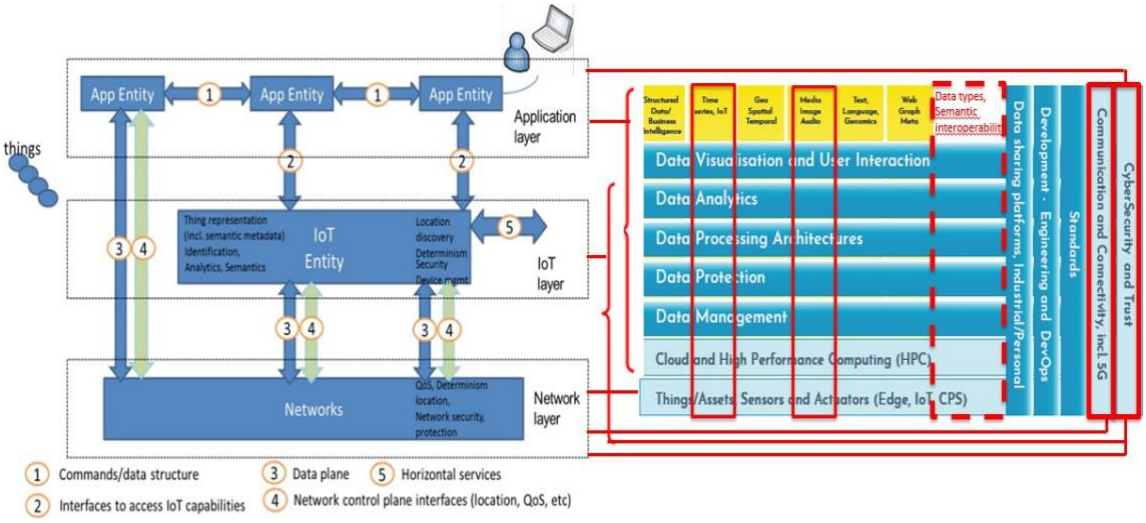

Figure 44: BDV Reference Model mapping to the AIOTI HLA

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 45**.

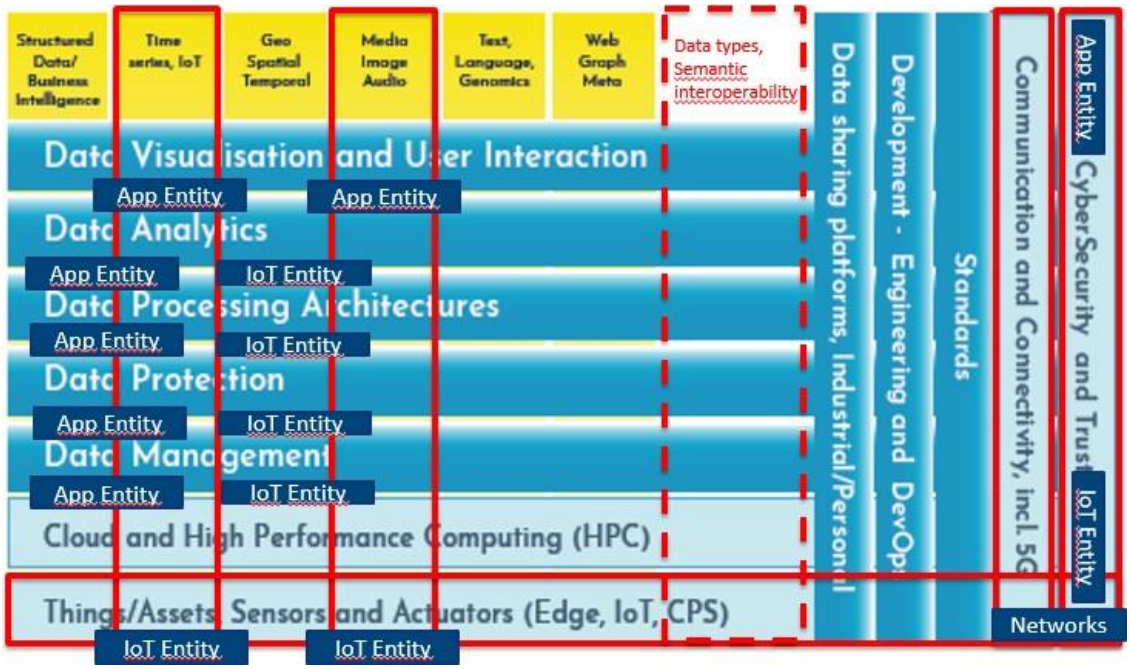

Figure 45: AIOTI HLA mapping to the BDV Reference Model

7.6 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 46**.

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 analysed and shortly described in this clause.

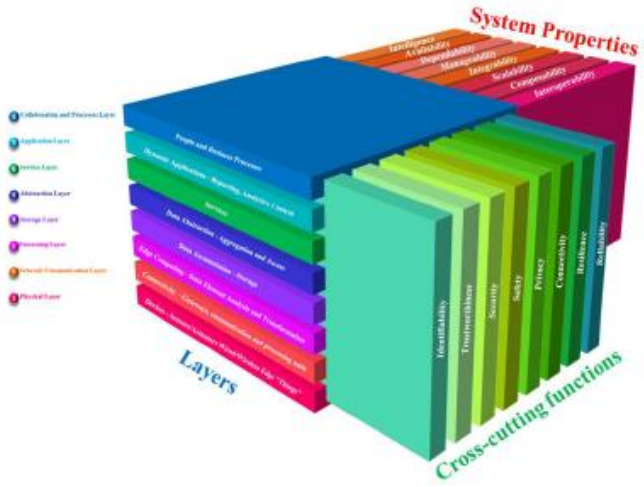

Figure 46: The three main views in the 3D Model (Layers, Cross-cutting functions, and Properties) [41]

The "Layers" view in the 3D model, see **Figure 47**, 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.

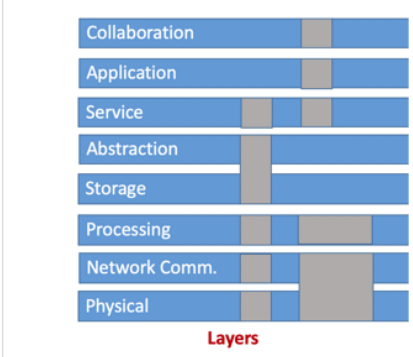

Figure 47: The Layers view in the 3D Model (Layers, Cross-cutting functions, and Properties) [41]

The "Cross-cutting Functions" view, see **Figure 48**, 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.

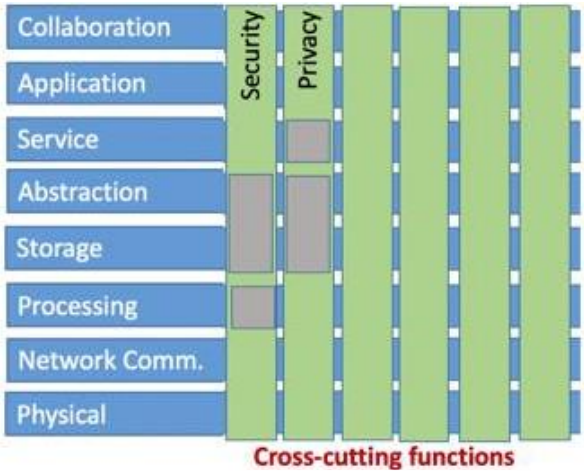

Figure 48: The Cross-cutting Functions in the 3D Model (Layers, Cross-cutting functions, and Properties) [41]

The "Properties" view, see **Figure 49**, 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.

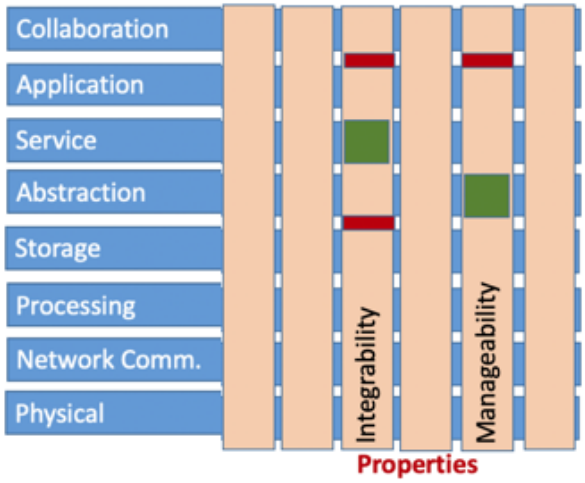

Figure 49: The Properties view in the 3D Model (Layers, Cross-cutting functions, and Properties) [41]

7.7 ISO/IEC JTC1

ISO/IEC JTC1 started in 2018 the activity group AG8 on meta reference architecture, the purpose of which was to provide guidance on the specification architecture standards. AG8 was disbanded in 2023 further to the provision of a guidance document which pointed out the following points:

- ISO/IEC/IEEE 42010 Architecture description provides the concepts (stakeholders concerns, viewpoints and views)
- ISO/IEC 30141 ED2 (IoT reference architecture) to include a construction view that allows to use construction patterns. A variety of construction patterns can be used, such as SGAM in smart grids or RAMI in smart manufacturing, or another reference architecture standard, such as ISO/IEC 30188 Digital Twin Reference Architecture.

Further to AG8 study, ISO/IEC JTC 1/SC 7/WG42 has started the development of two standards:

- ISO/IEC/IEEE 42024 Architecture fundamentals
- ISO/IEC/IEEE 42042 Reference architecture

In parallel ISO/IEC JTC 1/SC 41 is about to publish ISO/IEC 30141 ED2 (IoT reference architecture) and has started ISO/IEC TR 40141 (Guidance on reference architecture). The figure below depicts the approach used:

- The first step is to use ISO/IEC Ed2 30141 to guide the selection of a derived requirements, construction patterns and guidance:
    - ISO/IEC Ed2 30141 is extended by construction patterns, or architecture patterns that can be used at the reference architecture level. Some constructions patterns are already specified such as the IoT component capability pattern, the IoT enterprise networking pattern or the RAMI 4.0 pattern.
    - ISO/IEC Ed2 30141 is supported by guidance document such as ISO/IEC TR 40141 or ISO/IEC 30149:2024.
    - The second step is to use the derived requirements, construction patterns, guidance to guide the specification of an IoT implementation architecture.
    - The third step is to implement the IoT system.

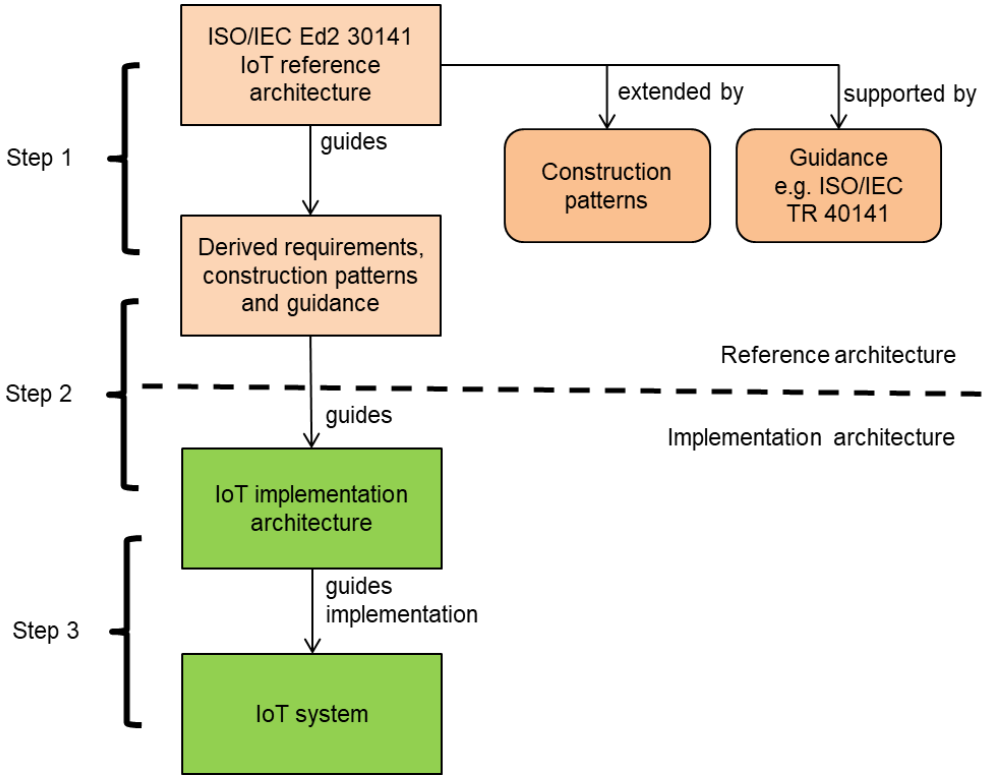

Figure 50: ISO/IEC JTC1 Reference architecture approach

The approach is versatile, as showed in the figure below:

- the requirements are those described in the architecture views of ISO/IEC Ed2 30141;
- the construction patterns include the component capability pattern, the IoT enterprise system pattern described annexes of ISO/IEC Ed2 30141, and the HLA (considered as a pattern).
- the guidance includes this document and the IoT trustworthiness principles (ISO/IEC TS 30149:2024).

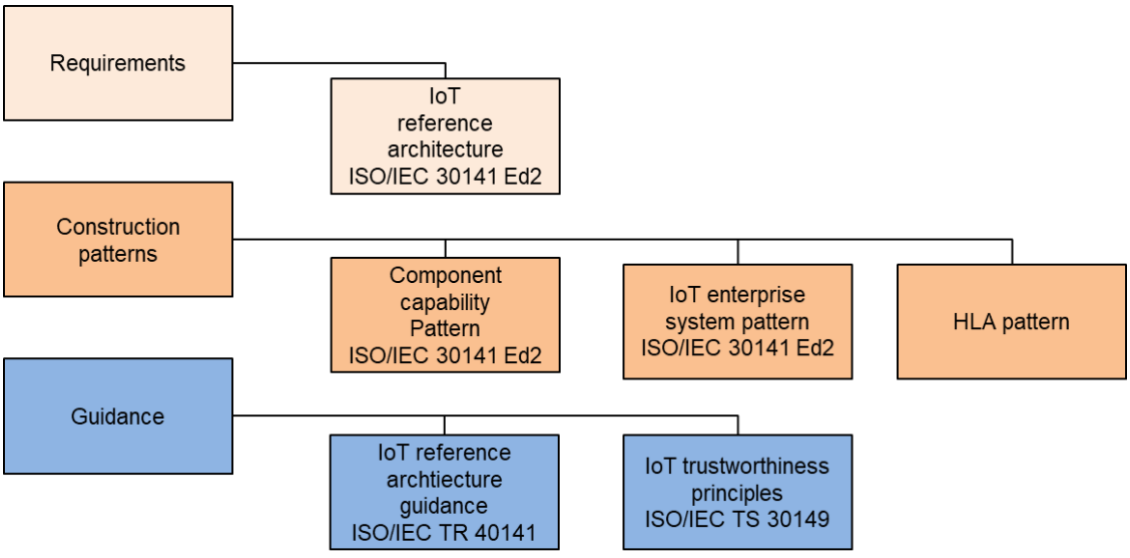

Figure 51: Example of using the ISO/IEC JTC1 Reference architecture

7.7.1 ISO/IEC 30141:2024 IoT - Reference architecture

This section provides a brief description of ISO/IEC 30141:2024 IoT - Reference architecture

As shown in **Figure *52***, the ISO/IEC 30141:2024 IoT - Reference architecture building blocks are:

- IoT Environment can be composed of IoT Systems and IoT components
- IoT System is composed of IoT components.
- The IoT component view is shown in **Figure *53***.

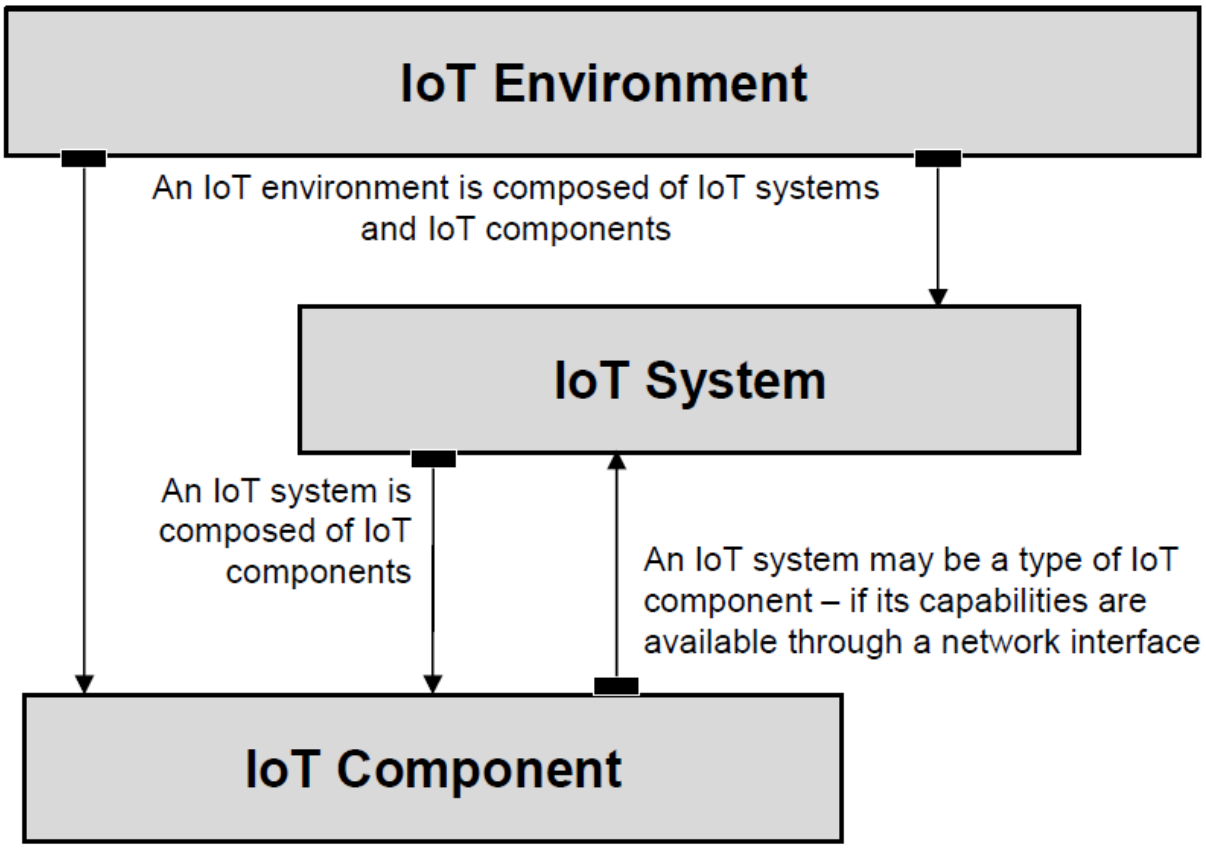

Figure 52: ISO/IEC 30141:2024 IoT - Reference architecture, copied from ISO/IEC 30141:2024 IoT - Reference architecture

Application entities are part of the ISO/IEC 30141 IoT Environment layer. However, is worth noting that there are application entities that are clearly created to be a part of the IoT Environment. There are also other applications that are more of separate IT systems, mainly dealing with other kind of information than IoT data but where IoT data are provided to that system by an IoT system. Whether or not such separate IT system should be considered to be a part of the IoT environment could be investigated and is up to an IT architect designer. It can be assumed that the interface to such separate IT system is a part of the IoT Environment but not the separate IT System itself.

The network infrastructure provides the connection between the components of the IoT System. Thus, the network infrastructure is also a part of the IoT System, comprising the system together with the IoT Components. Furthermore, since an IoT System can be a system of systems, there can be network infrastructures on many layers.

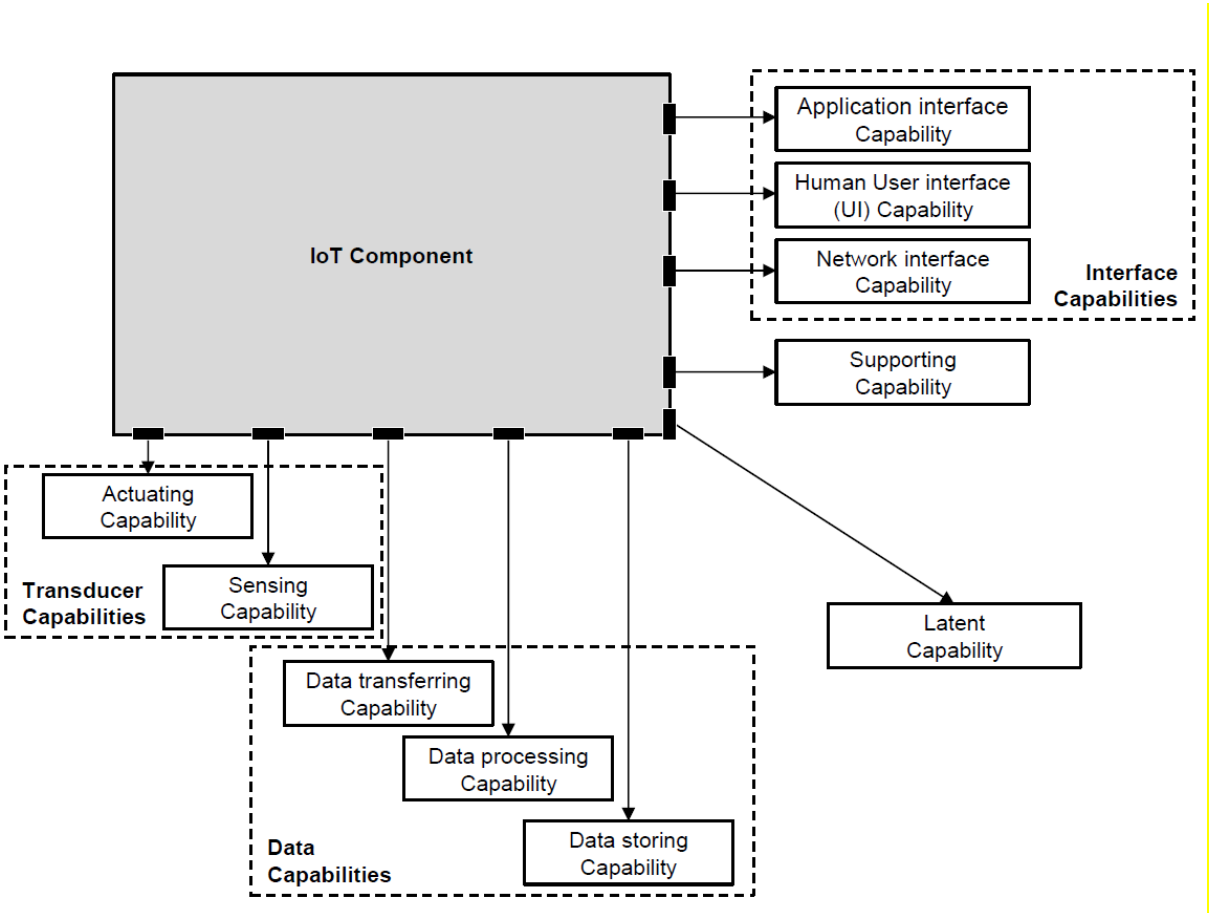

Figure 53: IoT Component view based on ISO/IEC 30141:2024 IoT - Reference architecture, copied from ISO/IEC 30141:2024 IoT - Reference architecture

8 Relationship to other functional models or systems

8.1 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 54** shows in particular interactions with Big Data frameworks and other service platforms (banking, maps, open data, etc.).

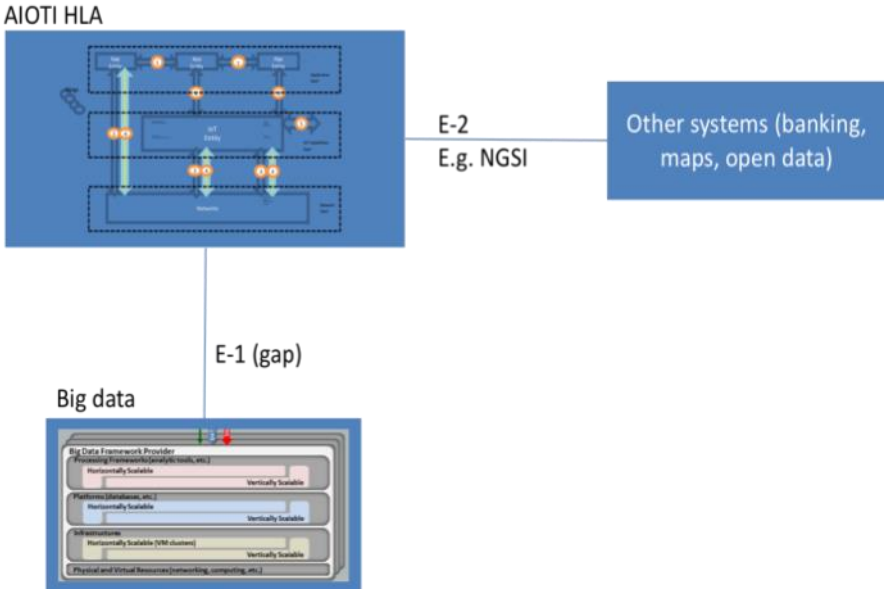

Figure 54: Relationship to other systems

Figure 54 shows 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 Framework of IoT-Big Data integrated architecture

8.2.1 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 55**.

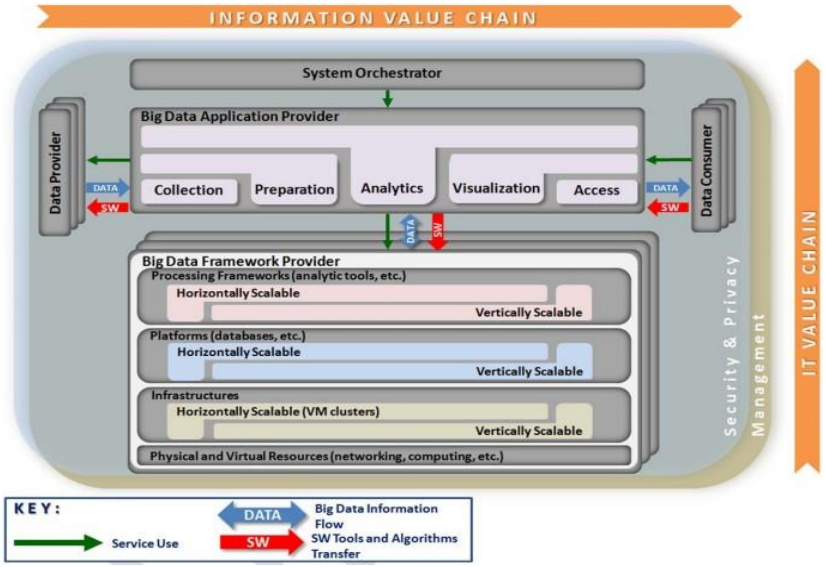

Figure 55: NIST Big Data reference architecture

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 56** depicts this mapping example.

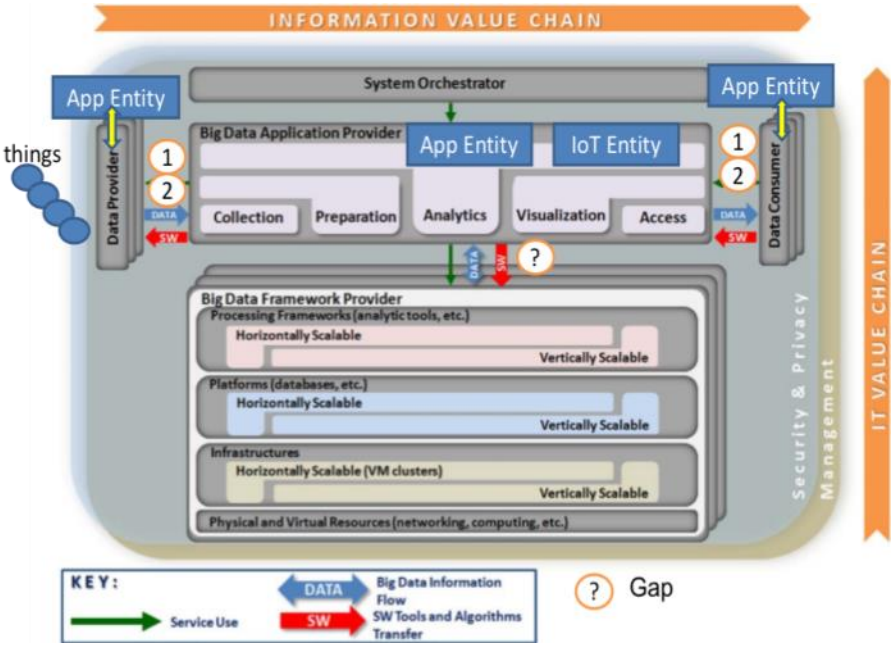

Figure 56: Mapping of AIOTI functional model entities to NIST big data reference architecture

In **Figure 56** 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 IoT-enabled Data Marketplaces

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

Figure 57 provides a possible high-level architecture for an IoT-enabled Data Marketplace [42].

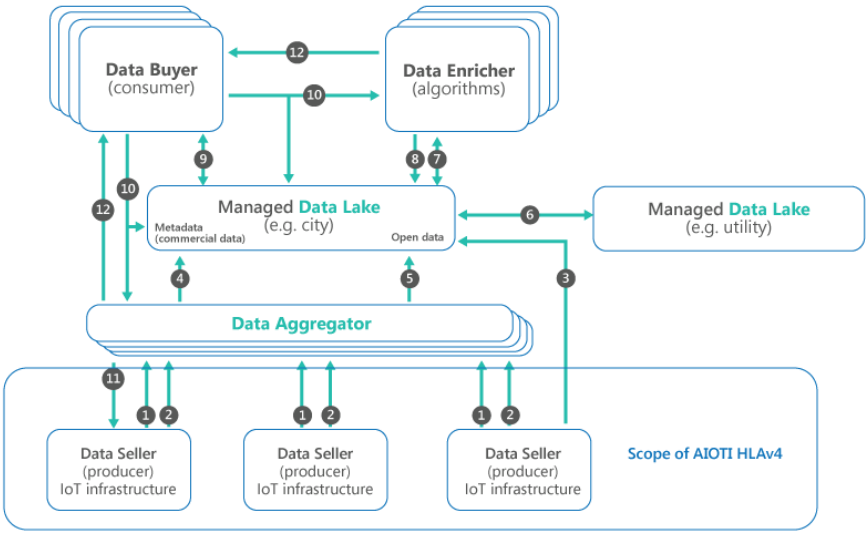

Figure 57: A possible high-level architecture for an IoT-enabled Data Marketplace

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**[10] 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.

8.3.2 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 57**.

- **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.

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

- **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**[11]**;**
  - Ensuring a fair distribution of revenue streams and avoiding the creation of new monopolies.

8.3.3 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.

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

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.

8.3.3.1 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.

8.3.3.2 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 favour 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.

8.3.4 Market inhibitors and technology gaps of a Mobility Data Marketplace

Figure 58 illustrates market inhibitors of a Mobility Data Marketplace.

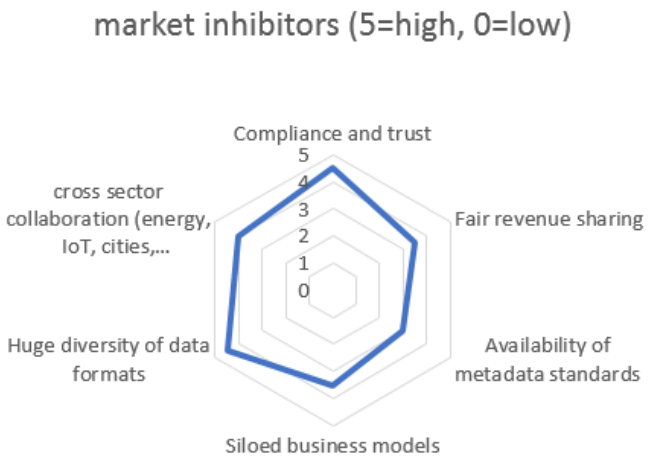

Figure 58: Market inhibitors of a Mobility Data Marketplace

- **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.

8.4 Relationship to other service platforms

Figure 59 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.

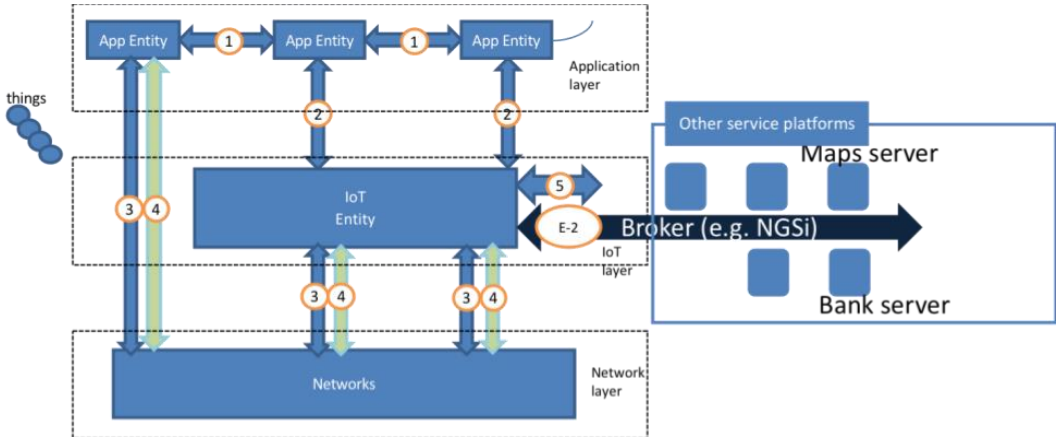

Figure 59: E-2 interface illustration

Figure 60 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.

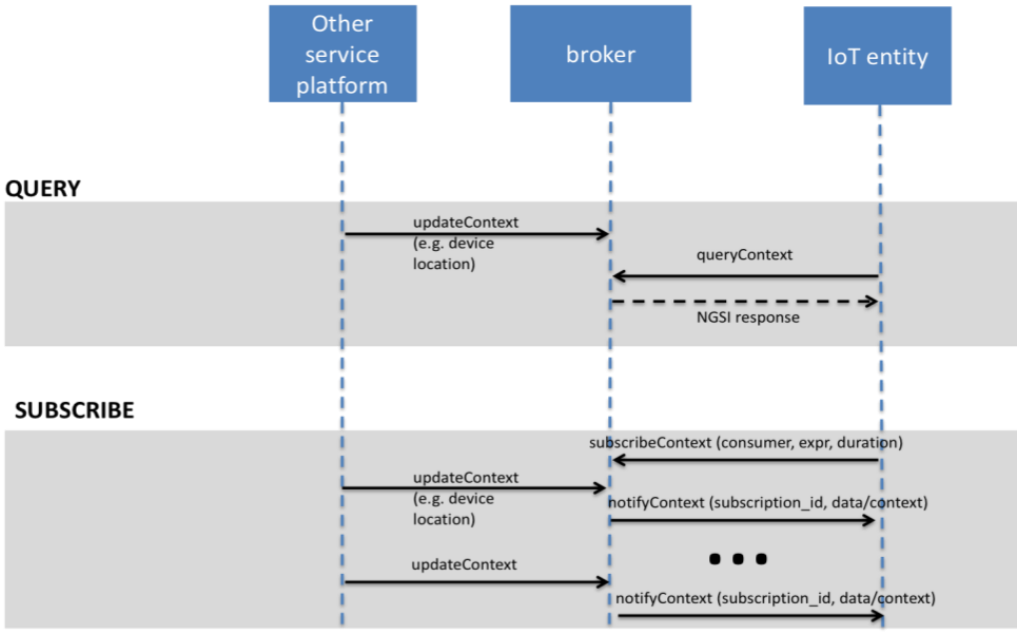

Figure 60: Example of message flow illustrating the E-2 interface

8.5 Relationship to EUCloudEdgeIoT.eu Open Continuum Reference - Mapping of HLA to Compositional view of the Continuum Reference Architecture

8.5.1 Architectures proposed in EUCloudEdgeIoT.eu projects

This section includes a brief description of architectures proposed in several EUCloudEdgeIoT.eu projects.

8.5.1.1 6G-Cloud architecture

The prime target of 6G-Cloud is to provide the blueprint for the overall service-oriented 6G system architecture design, composed of basic network functionalities and multiple control and management frameworks over a multi-stakeholder cloud environment spanning from extreme edge to central clouds, forming that way cloud continuum. This cloud continuum approach is generally still in the conceptual phase, and many challenges in both its design and implementation need to be addressed. One of the key challenges is enabling the ability to provide scalable orchestration that can work atop the cloud continuum and provide benefits from the flexible placement of network functions.

Furthermore, the management and orchestration platform implementing Network Services (NS), may significantly benefit from the so-called native AI approach in which all network functions, including the User Plane (UP), Control Plane (CP) and the Management and Orchestration Plane (MOP). The adoption of native AI approaches needs a flexible AI-powered orchestration able to orchestrate/update AI functions of any 6G system plane on an intelligent and flexible basis. The notarisation of the networking solutions and the need for each plane programmability need flexible support from inter-function and inter-plane communication.

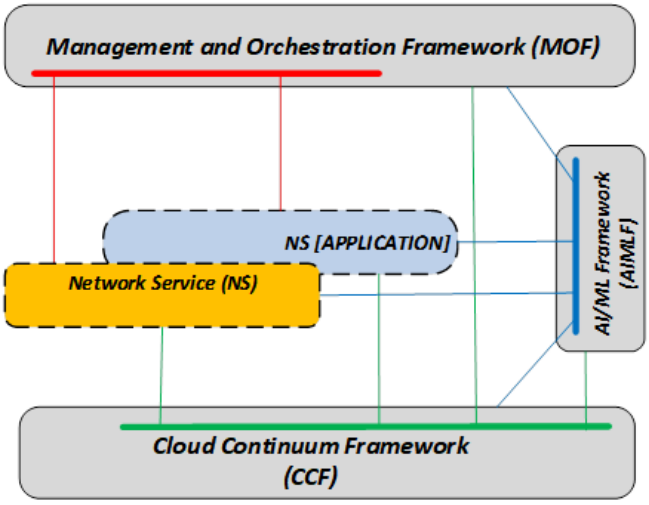

Figure 61: Overall 6G-Cloud orchestration architecture.

The 6G-Cloud architecture is based on the paradigms described above and it has been decomposed into several frameworks using the separation of concern approach. The main idea is to have loosely coupled cooperative frameworks of which each can be updated independently. In the system, the following frameworks are proposed:

- The Cloud Continuum Framework (CCF) that provides and manages cloud continuum resources of multiple resource providers. The CCF functions can be orchestrated and AI-driven. CCF creates Resource Partitions to be used by NSs and handles a dynamic resource pool, as the data centres or hosts can be dynamically attached and removed.
- The Management and Orchestration Framework (MOF) that is compatible with CCF, i.e. it can use cloud continuum resources. So far, there are no orchestration solutions that can work atop Cloud Continuum. The MOF per se is programmable, its functions can be orchestrated and AI-driven. In some implementations, NS management functions can be a part of NS.
- The AI/ML Framework (AIMLF), which is responsible for orchestration and performance monitoring of AI/ML-driven functions. To reduce the uncertainty related to AI/ML-driven decisions, the AIMLF of 6G-Cloud uses Network Digital Twin (NDT). As has been already described the AI/ML-driven functions can be also part of a Network Service, so AIMLF also supports them.

The mentioned frameworks have to cooperate to achieve their goals and to support NS efficiently. The CCF and MOF subsystems have to make possible deployment of different NSs, NDTs and applications.

For designing an end-to-end SBA for 6G networks, it is important to look beyond the traditional 5G network and consider the true cloud-native design in a big picture. With the cloud-native design as the key element in 6G networks, SBA will extend from CN to RAN and further reach the MOF and CCF. In addition, the AI/ML capabilities will be essential to support the network operation and optimization. The envisioned SBA for 6G introduces a comprehensive and unified approach that integrates various network domains, functions, and resources through a service-based architecture design principle and standardized service interfaces. Central to this architecture is the concept of cloud-native design, modularity and unified management, enabling NFs, RAN, Core, AI/ML or cloud infrastructure, to communicate seamlessly through clearly defined service and business interfaces and APIs.

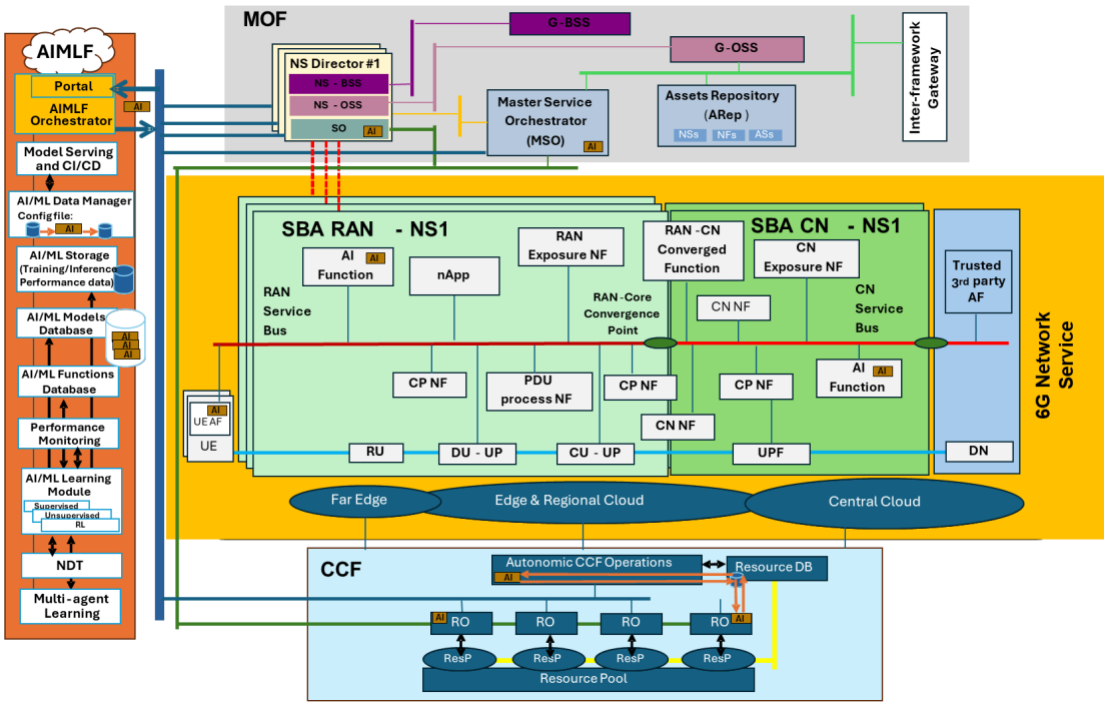

Figure 62. 6G-Cloud consolidated architecture design

Considering the abovementioned points, **Figure *61*** depicts the initial design of 6G-Cloud architecture. It includes service-based RAN and CN as the evolved mobile network, as well as the key management and support frameworks (AIMLF, MOF, CCF). **Figure 62** shows how different components are interconnected and utilize respective cloud and network resources over the Cloud-Edge/Far-Edge Infrastructure for 6G services. A 6G-Cloud NS can be considered as a virtual end-to-end mobile network including all physical network parts (RAN, Core) as well as computation infrastructure (Cloud, Edge, Far Edge) that are necessary for deploying applications and services with specific Service Level Agreements (SLAs) and Quality of Service (QoS). Each 6G-Cloud NS can include various Network Slices that may correspond to different applications and verticals.

One key concept in 6G-Cloud architecture is the end-to-end SBA in RAN and CN. Moving one significant step beyond 5G network architecture, the virtual RAN, intelligent RAN controller, RAN control applications, and network exposure functions are unified in the new RAN architecture, where RAN control plane network functions can be implemented under a new service bus with standardized interfaces and APIs. The advantages from SBA CN are inherited by RAN, while the RAN design will take extra mechanism to ensure the performance, and compatibility with old reference-point design in RAN. The network functions as a service (NaaS) approach allows the flexible instantization of functions over hybrid cloud infrastructure based on needs.

Beyond service-based network functions, the architecture incorporates a dedicated service-based AI/ML framework (AIMLF), which provides AI and ML capabilities as reusable services. This AIMLF delivers AI/ML capabilities, including but not limited to analytics and optimization, to intelligent functions across network segments, enhancing automation and improving efficiency of network management and operations.

The MOF, another integral component, operates as a unified platform managing the lifecycle of services across all network layers and domains. This framework oversees resources within RAN, Core Networks, and cloud infrastructure, ensuring coherent operations, efficient resource allocation, and rapid deployment and scaling of services.

Furthermore, the NS is underpinned by the CCF, responsible for seamlessly orchestrating cloud resources across diverse cloud platforms, ranging from centralized data centres to edge and extreme-edge nodes. The Cloud Continuum manages cloud resource partition, handles service placement, ensures efficient utilization of distributed cloud resources, and addresses varying latency, bandwidth, and computational requirements. It plays a central role in supporting NS deployment, particularly in the context of distributed and cloud-native deployments typical of future 6G environments.

8.5.1.2 COGNIT Architecture

COGNIT represents a response to the call of the European Commission to work together towards a brighter, more sustainable, and more inclusive digital future for a technologically-sovereign Europe.

An effective platform for the cognitive cloud-edge continuum must address a number of unsolved challenges, many of them derived from constrained resource devices, infrastructure heterogeneity, and the need to meet criteria such as performance, resilience, security, data sovereignty, and energy efficiency. A disaggregated architecture is required, making use of AI, automation, and portability to manage and adapt resources and workloads, and to respond in real time to possible incidents and security threats. Edge application developers willing to speed up computation, save energy, and cut costs will need a way to combine their edge devices with the many resources available across the cloud-edge continuum.

This innovative approach requires computationally-intensive data processing functions to be easily executed outside edge devices, sensors, and actuators.

It is with that vision in mind that this Horizon Europe project (2023-2025) focused on a new distributed Function-as-a-Service (FaaS) paradigm for edge application management and smart orchestration, which will change how applications and services are deployed and executed in the cloud-edge continuum. Our AI-enabled adaptive serverless framework provides applications with secure and portable access to a continuous data processing environment that abstracts the large-scale, geo-distributed, and low-latency capabilities provided **by the cloud-edge continuum.**

COGNIT enables the seamless, transparent, and trustworthy integration of data processing resources from cloud providers and on-premises data centers in the cloud-edge continuum, and their automatic and intelligent adaptation to optimise where and how data is processed according to application requirements, changes in application demands and behaviour, and the operation of the infrastructure in terms of the main environmental sustainability metrics.

The main objectives of the COGNIT Framework are:

- Support a new innovative **Serverless paradigm for edge application management**, based on code offloading.
- Enable the **on-demand deployment** of large-scale, highly distributed and self-adaptive serverless environments using existing data processing resources from cloud/edge infrastructure providers, including local data centres, cloud providers, and 5G/telecom operators.
- **Optimise where data is processed** according to changes in application demands and behaviour, and energy efficiency heuristics.

COGNIT tackles the challenges of mobile and far-edge device applications that require computationally intensive data processing beyond the capabilities of local hardware. By leveraging code offloading, COGNIT enables data-heavy tasks to be executed outside of devices—such as sensors, actuators or low power CPUs—leading to improved power efficiency, reduced storage needs, and enhanced performance even when hardware acceleration is unavailable. While code offloading is a proven method for saving energy and increasing responsiveness, its broader adoption has been hindered by integration complexities with cloud management, dynamic system configurations, scalability issues, and the absence of Offloading-as-a-Service solutions. COGNIT addresses these challenges by introducing a new distributed Serverless model, integrated within a cognitive cloud-edge management platform, to enable the development of elastic, scalable, and efficient edge-based applications.

To deliver seamless end-user experiences, COGNIT empowers developers to integrate cloud-edge processing directly into their applications using its distributed Serverless model. This approach allows developers to offload specific code fragments or tasks from end-user devices to the cognitive continuum, improving computation speed, reducing energy consumption, saving bandwidth, and ensuring low latency. While existing Function-as-a-Service (FaaS) solutions—such as AWS Lambda, Azure Functions, and Google Cloud Functions—have gained traction by simplifying infrastructure management, they are designed primarily for proprietary public clouds, with small, short-lived functions. Open source frameworks such as Apache OpenWhisk, Iron Functions, Fission, Kubeless, OpenFaaS are designed homogeneous datacentres with commodity infrastructure. COGNIT extends the Serverless paradigm to heterogeneous and distributed environments, bridging the gap between edge devices and public/private cloud services to unlock new levels of flexibility and scalability for next-generation applications.

Table 2: COGNIT Serverless Cloud-Edge Model vs traditional Serverless/FaaS Cloud Model

Serverless/FaaS Cloud Model	COGNIT Serverless Cloud-Edge Model

PROGRAMMING		
Programming model	Interconnected functions (code) defined at a Cloud provider	Single program source code on device that follows an asynchronous model
Where the function runs	On Cloud provider	On cloud-edge location
When the function runs	On event, cloud event-driven	On demand, application logic-driven
Application profile	Low footprint	Compute-intensive data processing
Program state	Stateless	Stateless / Stateful
Maximum runtime	Short (e.g. <900 seconds)	None
Maximum capacity	Limited (e.g. < 3GiB memory)	None
Communication patterns	Workflows and state machines	Results forwarding to other functions
Deployment requirements	Basic capacity (e.g. memory) & quotas	Performance, cost, security, and energy
OPERATION		
Scaling	Public Cloud provider responsible	Private Cloud provider responsible
Deployment	Public Cloud provider responsible	Private Cloud provider responsible
Fault Tolerance	Public Cloud provider responsible	Private Cloud provider responsible
INFRAESATRUCTURE		
Infrastructure	Single centralised cloud	Dynamic distributed cloud/edge
Location	Single centralised cloud	Developer selects (cloud-edge continuum)
Special-purpose devices	None	Hardware (GPU), services (AI)

The COGNIT Framework is based on the architecture described in Figure 63:

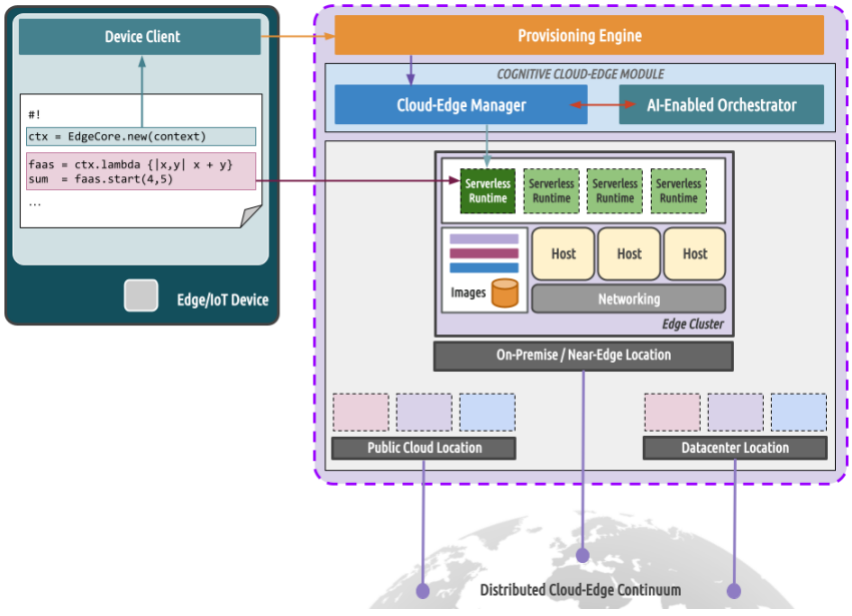

Figure 63. General view of the COGNIT Architecture.

These are main components of the COGNIT Architecture:

- The **Device Client** allows the device to offload functions to any tier in the cloud-edge continuum, according to some requirements provided by the device itself; it will

communicate with the Provisioning Engine in order to create Serverless Runtime for the execution of device application tasks.

- The **Serverless Runtime** is in charge of executing the functions offloaded by the devices and storing data uploaded by the devices or from external storage backends.
- The **Provisioning Engine** is responsible for managing the lifecycle of the Serverless Runtimes.

The final two architectural components are grouped together as part of the Cognitive Cloud-Edge Module that allows the management of the cloud-edge continuum resources in an intelligent and adaptive way

- The **Cloud-Edge Manager** is responsible for scheduling the Serverless Runtime according to the deployment plan provided by the AI-Enabled Orchestrator
- The **AI-Enabled Orchestrator** is the component that, according to the device requirements and infrastructure availability, will optimally schedule the Serverless Runtime on the cloud-edge continuum resources.

COGNIT architecture has been designed to address heterogenic use cases needs, and 4 different use cases were selected to gather requirements, validate and piloting. These requirements were collected through a combination of information provided by each Use Case leader through an plus a series of regular collaborative workshops with each Use Case:

- **Smart Cities:** Connected vehicles and autonomous driving are set to revolutionize transportation, enhancing road safety, reducing accidents, and improving efficiency. To realize this vision, systems must be interoperable, intelligent, and secure, while supporting multi-tier edge applications that operate seamlessly across the cloud–edge continuum. Key challenges include managing dense networks of edge infrastructure and coordinating diverse services with varying QoS requirements.
- **Wildfire Detection:** Europe has witnessed a significant increase in the number and ferocity of so-called 'mega-fires', a phenomenon linked with climate change. Edge/IoT devices, coupled with AI/ML, can play an important role in preventing and fighting wildfires. The main challenges will be rare events (suspected fire) with sudden peaks of extremely high offloading demands, and effective energy management to counter the lack of dedicated power supply.
- **Energy:** Supporting the energy transition in Europe requires new solutions providing wide and open accessibility of energy data, and individual energy independence of a household and/or small energy clusters. Smart edge applications using advanced AI/ML algorithms could be used for monitoring, predicting, and managing both energy production and consumption. The main challenges will be resource-constrained environments (i.e. energy meters) and the very high security requirements for distribution system operators (DSO), as the energy sector moves away from a hierarchical, centralized structure towards a more decentralized and distributed way of managing energy assets and networks.
- **Cybersecurity:** Moving computation and data processing services to the edge, far from secured data centers, leaves systems exposed to new threats. Edge Computing requires a new generation of intelligent security mechanisms to be deployed along with edge applications, implementing advanced authentication and authorization policies. This Use Case explores resilient anomaly detection and remediation in a smart mobility context.

- The main challenges will be the migration of workloads between edge nodes and managing a DevSecOps pipeline in a multi-provider edge context with dynamic (geo-dependent) security policies.

Modern transportation systems must be interoperable, intelligent, and secure, while supporting multi-tier edge applications deployed seamlessly across the cloud–edge continuum. By leveraging data locality and cloud-native practices, these systems can reduce overhead and ensure robust management.

Figure 64 describes the deployment architecture of "Smart-Cities" use case, related to urban transportation. Leveraging ACISA's **Mobility-Hub (M-Hub)** edge computing capabilities, it was explored the deployment V2X Transit Signal Priority (TSP) service in several intersections of Granada (Spain), aimed to reduce transit times for emergency vehicles and public transportation. COGNIT enabled the deployment of distributed Digital Twins of the urban road infrastructure, seamlessly activated from each M-Hub upon a FaaS request. These Digital Twins supported dynamic "what-if" analyses to evaluate V2X Transit Signal Priority (TSP) service requests in real time, determining whether requests should be approved or denied.

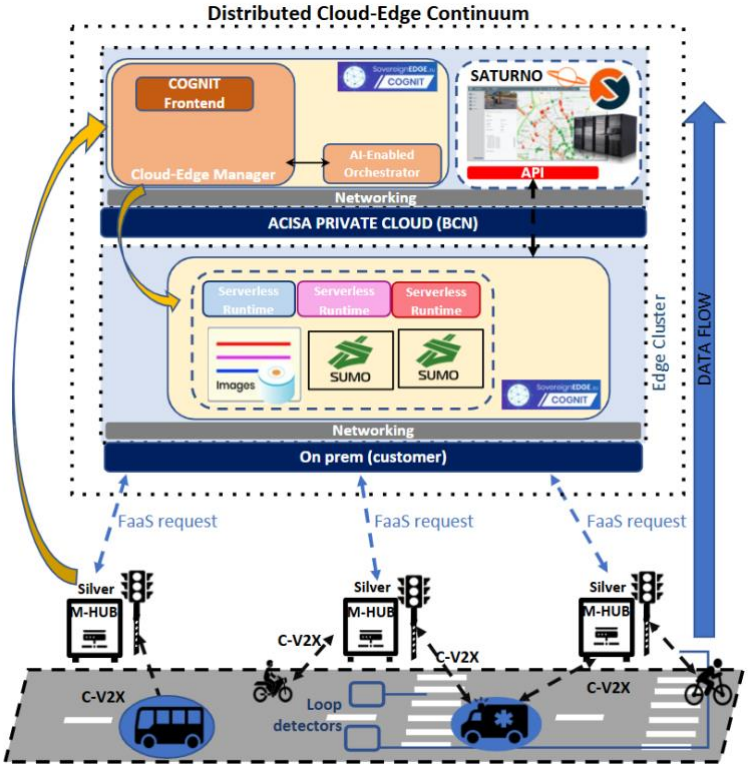

Figure 64. General view of the UC1 "Smart City" Architecture.

The reference scenario involves the following steps:

1) A vehicle initiates a priority (green light) request by sending a standardized V2X Signal Request Extended Message (SREM) message (containing, e.g. bus ID, Lane ID, delay information, and other related data if available), first intercepted by an RSU that forwards the request to the relevant M-Hub, which filters incoming V2X messages for priority requests.
2) Upon receiving a priority request, a FaaS request will be initiated by the M-Hub, to offload the processing to decide whether priority should be given.

3) The function will make use of either traditional algorithms or AI models to decide whether to grant or deny priority, assessing the traffic situation at the intersection under consideration based on the following data from the offloading M-Hub, included as arguments to the Serverless Runtime:

   - Data stored in the particular M-Hub node that initiated the FaaS request, e.g. traffic light controller status, loop detector data, and V2X Cooperative Awareness Messages (CAMs) from other vehicles;

   - Data from the vehicle contained in the V2X SREM message received by the M-Hub node, e.g. bus/vehicle ID, lane ID, other related data and delay information if available.

4) In case the function needs extra data from external systems, it will be able to request the missing info either from ACISA's traffic management platform Saturno, SAE system from the public transport operator control centre (CC), or any other related information system.

5) Once the priority request has been processed and analysed, a message will be sent back to the requestor through another standardized ETSI message for V2X, Signal request Status Extended Message (SSEM).

8.5.1.2.1 Mapping of COGNIT Architecture into EuEdgeCloudIoT Reference Architecture

The Cognit reference architecture can be seamlessly mapped into EuEdgeCloudIoT (CEI) reference architecture, see **Figure 8**, as showed in **Figure 65**. Through these mappings, the COGNIT project demonstrates how its technical building blocks integrate seamlessly with the CEI reference model to enable secure, intelligent, and efficient cloud-edge-IoT ecosystems.

This diagram illustrates how the COGNIT project components align with the CEI Reference Architecture. The CEI architecture is structured around core functional domains such as security & privacy, trust, data management, resource management, orchestration, networking, monitoring, and AI.

1. In the COGNIT framework, the Cloud-Edge Manager component, specifically the JWT Token Generator is where Security and Trust functions are represented. They ensure controlled access, secure communication, and reliable operation across distributed edge locations.

2. The Monitoring Agent directly maps to Monitoring & Observability, providing real-time system insights and performance data.

3. The DaaS (Data-as-a-Service) module map into Data Management, enabling efficient handling, processing, and delivery of data at the edge.

4. Serverless Runtime, in charge of executing the functions offloaded, and the Provisioning Engine, responsible for managing its lifecycle, along with the Cloud Edge manager components, fit into Resource Management, as they handle provisioning, allocation and use of resources.

5. Finally, the AI Orchestrator, corresponds to the AI domain in CEI, driving autonomous orchestration and adaptive decision-making.

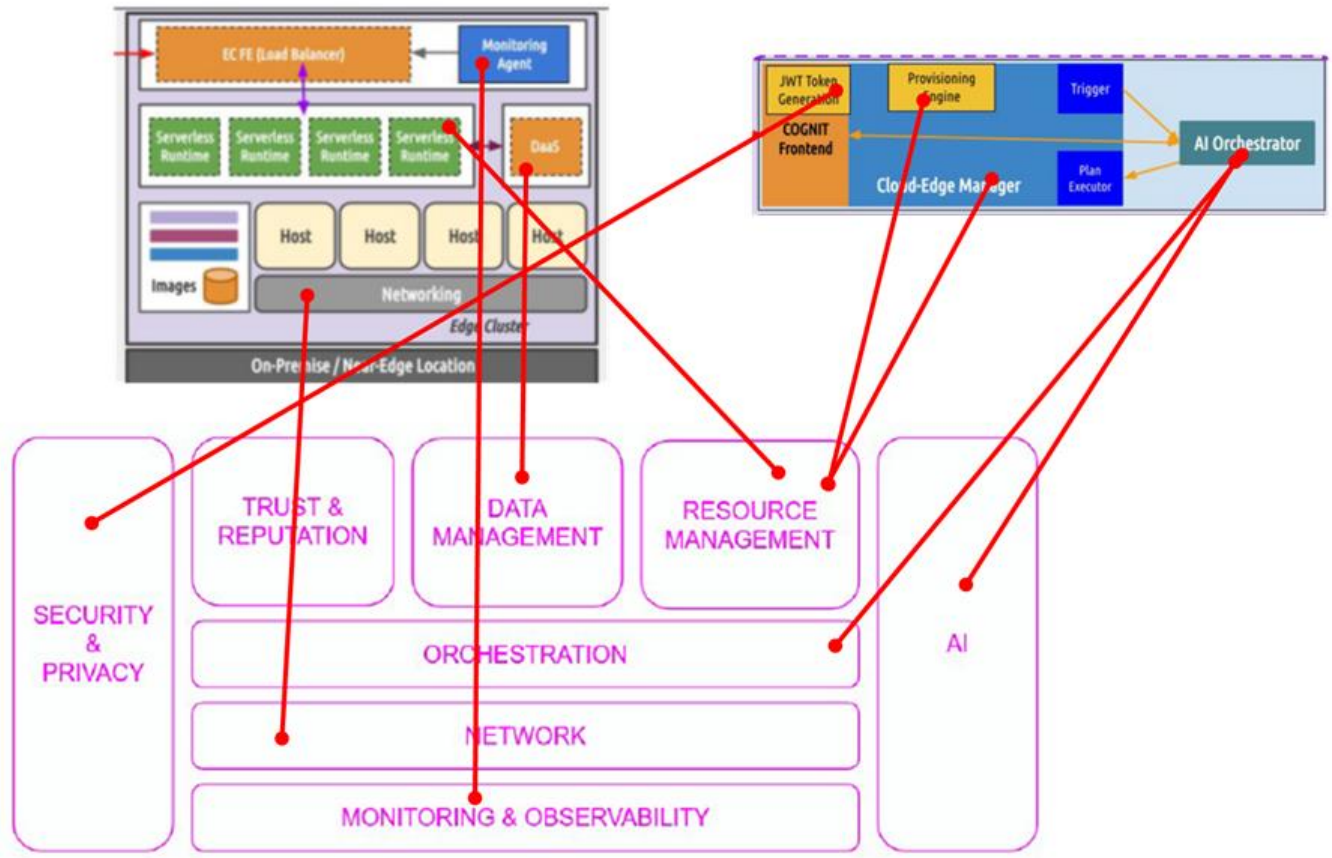

Figure 65 Cognit components mapped to the CEI Reference Architecture

8.5.1.3 CODECO Architecture

CODECO is a containerised application orchestration framework (TRL 4-5), independent and yet pluggable to Kubernetes[12] (K8s). CODECO aims at providing support for a flexible and cognitive *Cloud-Edge-IoT (CEI)* infrastructure. The notion of infrastructure in CODECO considers different infrastructure perspectives, taking into consideration:

- the **networking** perspective, where the network is perceived both as the *underlay* and *overlay* infrastructure required to support the deployment of containerised applications across CEI;
- the **data observability** perspective, bringing input on the data dependencies that micro-services may have, and that is important to consider during the application deployment and re-deployment;
- the **computational** perspective, bringing input on the suitable nodes to serve an application.

CODECO addresses the orchestration across these three different perspectives in an integrated approach, which is named as **data-compute-network** orchestration.

12 https://kubernetes.io/

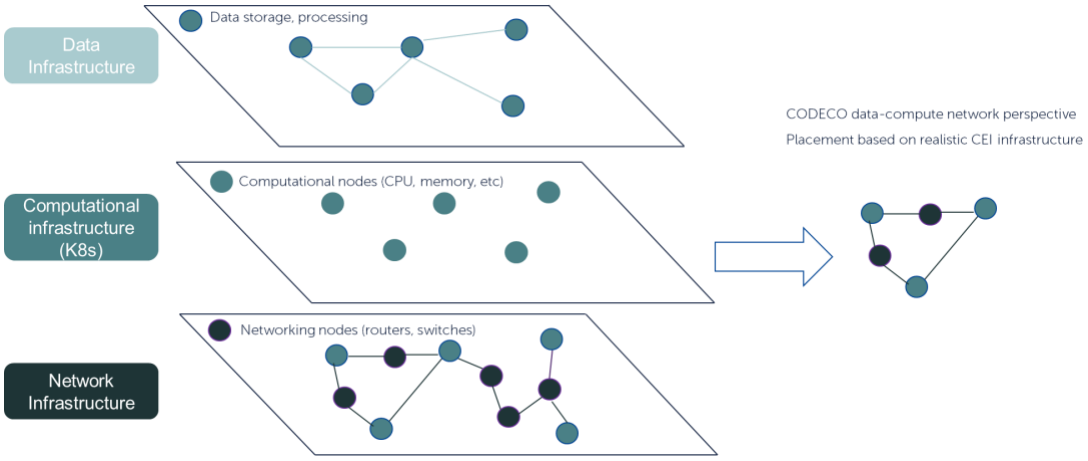

Figure 66: The CODECO data-compute-network approach for CEI.

The motivation to develop such an orchestration framework relates with the current evolutionary aspects of the CEI continuum. In this context, applications (and their components) need to be lightweight, highly portable, and mobile, to run anywhere and everywhere. Secondly, there is a need to handle large amounts of data often with very different features both in terms of computation (processing) and exchange (networking), including those with low latency demands. Thirdly, it is important to guarantee a smooth data flow, to allow for a smooth and secure integration end-to-end, across Edge-Cloud environments. Fourthly, real-time decision-making on where to compute and to store data must be supported.

As a cognitive and decentralised orchestration framework, CODECO embraces a next generation vision where data and services are stored and computed across the Internet, whole-chain, in a holistic cooperation between Cloud and far Edge. Components of containerised applications, which in CODECO are addressed as *micro-services*, reside in isolated, self-sufficient containers, which can be deployed and executed in any underlying environment. A flexible and secure networking infrastructure provides adaptable interconnection that adapts to the needs and the surrounding environment of the services being run.

Relevant in CODECO is an adaptation based on application requirements (e.g., required bounded latency) and user requirements (e.g., compliance required with GDPR). A functional representation of CODECO and its open-source components is provided in ***Figure 67***. Based on this functional level, CODECO offers the following aspects:

Automated configuration, focusing on supporting application setup and application runtime across Edge-Cloud, by taking into consideration compute, network, and data aspects. The key aspects of this contribution are supported by the CODECO *Advanced Configuration and Management (**ACM**)* component.

Data as a resource. CODECO addresses, via its *Metadata Manager (**MDM**)* component data as a resource in the sense that available snapshots from the overall Edge-Cloud infrastructure, integrating different perspectives (application, user, system, data, network) at different instants of the CODECO operational workflow can be provided to different CODECO components, to assist in detecting relevant changes.

Dynamic scheduling and workload migration. Supported by the CODECO component *Scheduling and Workload Migration (**SWM**)*, CODECO brings in novel QoS models that consider data-network-compute requirements to provide a best match between applications and available infrastructure (nodes, their computational and data properties, as well as network nodes and links), and to schedule and re-schedule application workloads across single cluster and federated cluster environments, considering application and user requirements.

Context-awareness and privacy preserving decentralised learning. CODECO relies on context-awareness to be able to achieve a joint data-network-compute orchestration, and on privacy-preserving decentralised learning and inference to best support readjustment of aspects such as the processing capability, computational resources, networking resources and interconnections in real-time. Context-awareness and decentralised learning are part of the CODECO *Privacy-preserving Decentralised Learning (**PDLC**)* component.

Infrastructure adaptation based on a cross-layer data-compute-network approach. Via the CODECO *Network Management and Adaptation component (NetMA)*, CODECO assists in adapting not just computational (node resources) but also the networking infrastructure interconnecting such nodes, having as starting point for the exposure of metadata within one cluster and across federated clusters, from far Edge to Cloud.

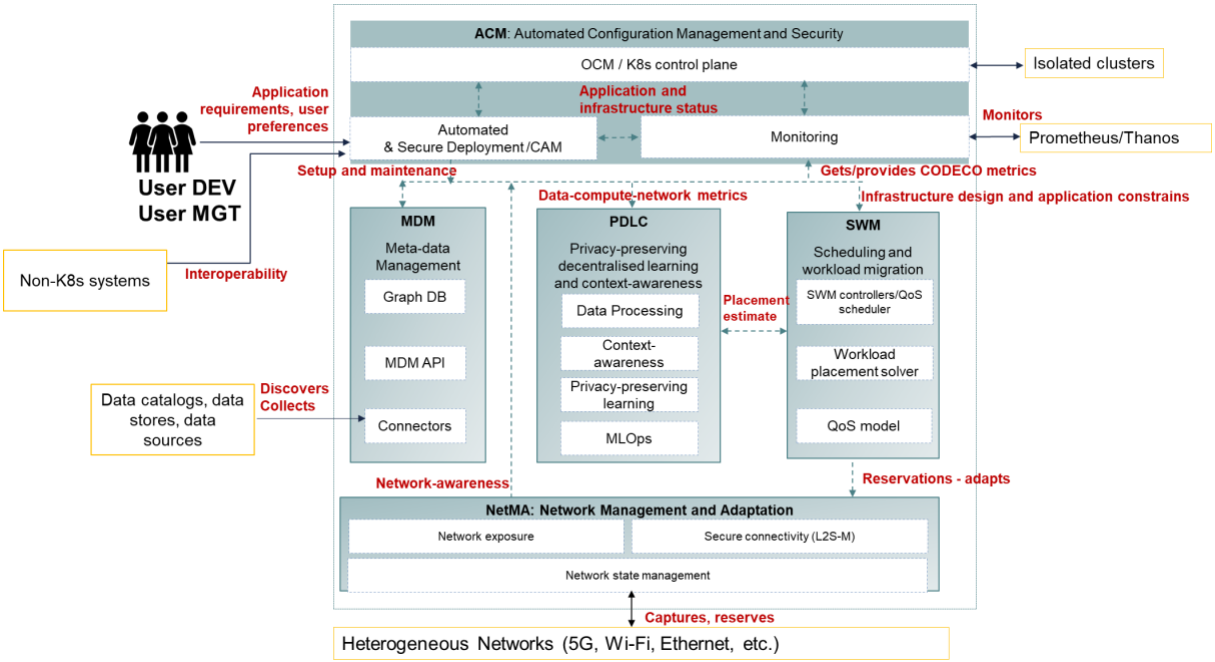

Figure 67: Functional representation of the CODECO K8s framework and its components.

The only interface of CODECO to the user, as shown in ***Figure 67***, is the ACM component, which focuses on supporting application setup and runtime from the far Edge to the Cloud, considering the input provided by the user.

The user in CODECO is perceived as an **application developer (DEV)** or as a **cluster manager (MGT)**.

To setup CODECO, a user relies on the CODECO framework. ACM installs the entire CODECO framework, handling the overall CODECO configuration. The user can specify, via the CODECO **Application Model (CAM)**, the requirements of the application and of its micro-services. These requirements are relevant to allow CODECO to deploy the application across CEI in an efficient way.

Via CAM, the user specifies also **target performance profiles** that is used in CODECO (PDLC) to meet a specific level of **greenness**, or **resilience** during its operation.

Once CODECO is installed, all CODECO components can be used. The CODECO components have been devised to allow each component to work independently, with other future open orchestration frameworks, requiring little adaptation.

The **CODECO MDM** component provides data and system observability to the other CODECO components, treating data as an integral part of the application workload, and integrating observability perspectives from different categories, e.g., application perspective, system perspective, network perspective, at different points in the CODECO operational workflow.

SWM handles the scheduling and rescheduling of the application workload, based on the *CODECO Application Model (CAM)*. CAM is supported by ACM and provided by the user during application setup, based on the novel data-compute-network approach proposed by CODECO.

The currently available approach in SWM for handling placement decisions relies on a graph-based optimization approach which in contrast to the K8s filtering and score approach is expected to provide an optimal match between application requirements and available resources (computational, network, data). SWM is also a control plane component, co-located with ACM and the K8s control plane, in master nodes.

PDLC is at the heart of the CODECO orchestration. Based on the infrastructure data collected by ACM (via Prometheus), NetMA, and MDM, PDLC has two functions. Firstly, it provides an aggregated cost view of a specific target performance profile for the available infrastructure which can be used by other components and is currently being considered in SWM to further define the optimal workload placement. Secondly, it provides an estimate on the overall system stability based on privacy-preserving decentralised learning approaches. PDLC is currently envisaged to operate on both K8s master and worker nodes.

NetMA provides the network awareness to CODECO and handles also secure connectivity across pods. Network awareness is provided via network probing based on the sub-component Network State Management (NSM). Then, exposure of network metrics (across multi-cluster environments) is based on the ALTO protocol. NetMA provides in addition **secure connectivity** via **L2S-M**[13]. Hence, NetMA is a component responsible for interconnecting the *Software-defined Networking (SDN)* world with K8s.

To be able to achieve a flexible orchestration, CODECO counts with monitoring across the three different CEI infrastructure perspectives. NetMA monitors the networking infrastructure; MDM monitors the data infrastructure; ACM monitors the system (computational nodes) infrastructure based on the K8s metrics server Prometheus[14,15]. This approach allows to explore the integration of new metrics as well as of providing the collected metrics of CODECO to future orchestration frameworks.

The CODECO framework is deployed as an application: each CODECO component is based on a modular approach, integrating different sub-components that are expected to be built as one or more independent (dockerized) micro-services.

13 http://l2sm.io

14 https://prometheus.io/

15 The current version of CODECO considers Prometheus as basis for the metrics gathering and analysis. However, for multi-cluster, multi-tenant environments the consortium is also considering Thanos due to the higher availability, and extended storage capabilities.

8.5.1.3.1 CODECO to CEI Continuum Mapping

As an orchestration framework, CODECO can operate at the Edge, Edge to Edge, Edge to Cloud, including far Edge to Cloud. The operation of CODECO relates with the management of an Edge-Cloud infrastructure which is based on the abstraction principles of K8s.

CODECO as a framework pluggable to K8s is developed to provide a flexible K8s operation considering mobile heterogeneous environments, and a reach to the far Edge. CODECO is developed in a way that allows it to operate in a single cluster or multiple clusters, across multi-tenant environments.

A cluster in this case is an abstraction from the K8s concept (rf. to section 2.1) which can be mapped solely to an Edge environment (including a far Edge environment), Edge-to-Edge, or Edge-to-Cloud. The principles of CODECO provide the capability to support Edge-to-Edge orchestration. The cluster abstraction concept always integrates a control plane (K8s master node, rf. to section 2.1) and a service plane (K8s worker nodes). From an Edge-Cloud continuum perspective, a cluster is expected to integrate at least one Edge node. A representation on this potential mapping for single cluster operation is illustrated in ***Figure 68***.

Cluster 1 provides an initial example for a centralised mapping, where the control plane of CODECO (and of K8s) resides in the Cloud, while K8s worker nodes being monitored and managed in CODECO are in the far and near Edge. As illustrated, CODECO components such as ACM and SWM (control plane of CODECO) would then reside in the Cloud, co-located with the K8s control plane. Sub-components of MDM, PDLC, and NetMA, the sub-components related with resource monitoring, would be located on the K8s worker nodes (far and near Edge in this example). While this representation places the CODECO control plane in the Cloud, it should be highlighted that it could also run in the near Edge or far Edge, as represented in **Cluster 2** for the near Edge. In this example, worker nodes are also running MDM, NetMA due to monitoring, and PDLC (learning at the Edge).

Cluster 3 provides a representation of the mapping of CODECO to the far Edge. CODECO can run on a single node or as represented, run on multiple nodes. The example here provided shows that in K8s worker nodes, NetMA is the only component active (network monitoring), while the other components are set to run co-located with the K8s control plane. MDM is not represented, to highlight that some components in CODECO are also optional. For instance, a user MGT may want to optimize the infrastructure just in terms of computational and networking resources, and hence, it is not required to run MDM.

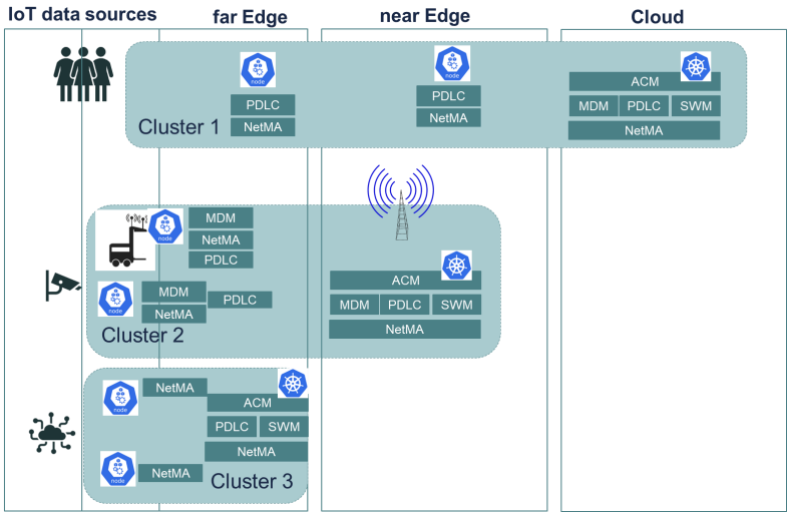

Figure 68: CODECO and relation to the CEI continuum, single cluster representation.

Figure 69 provides a high-level representation of the CEI mapping of CODECO for multi-cluster environments. In this case, a cluster may be located at a single region (far Edge, near Edge) or span different areas (e.g. far Edge and Cloud; far Edge and near Edge; far Edge-near Edge-Cloud). Multi-cluster operation handled by CODECO takes into consideration different clusters as in K8s (independently of location) but brings to K8s CEI awareness in the sense that it **takes into the placement process data, network and computational awareness**. This allows CODECO to best adapt to, for instance, far Edge environments, where devices may be constrained in terms of computational resources and networking resources. It allows CODECO also to react if the suitable nodes running the application do have the expected resources, but an abnormal pattern was detected with the data workflow (e.g., stale database) and with the networking resources (e.g., temporarily congested link).

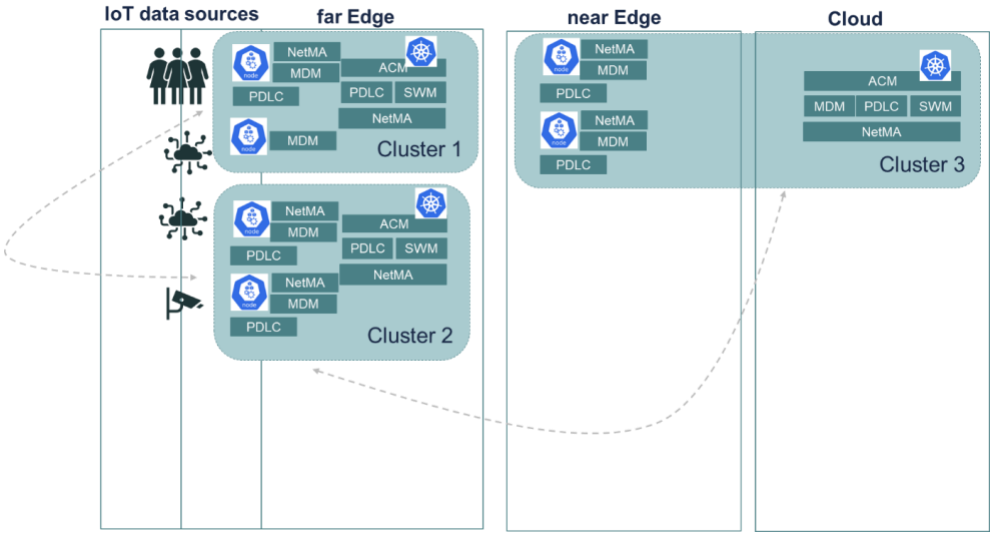

Figure 69: CODECO examples for multi-cluster, multi-tenant environments across the CEI continuum.

8.5.2 HLA representation of CEI Reference Architecture

Based on the CEI EU funded project information provided in **Section 8.5.1**, **Section 6.8.3** (*Digital Twin using the HLA representation*), i.e., **Figure 27**, **Figure 28**, and **Section 6.8.4** (*Computing Continuum Perspective*), i.e., **Figure 30**, we propose in **Figure 70**, a preliminary HLA representation of the CEI Reference Architecture, that is shown in **Figure 8**.

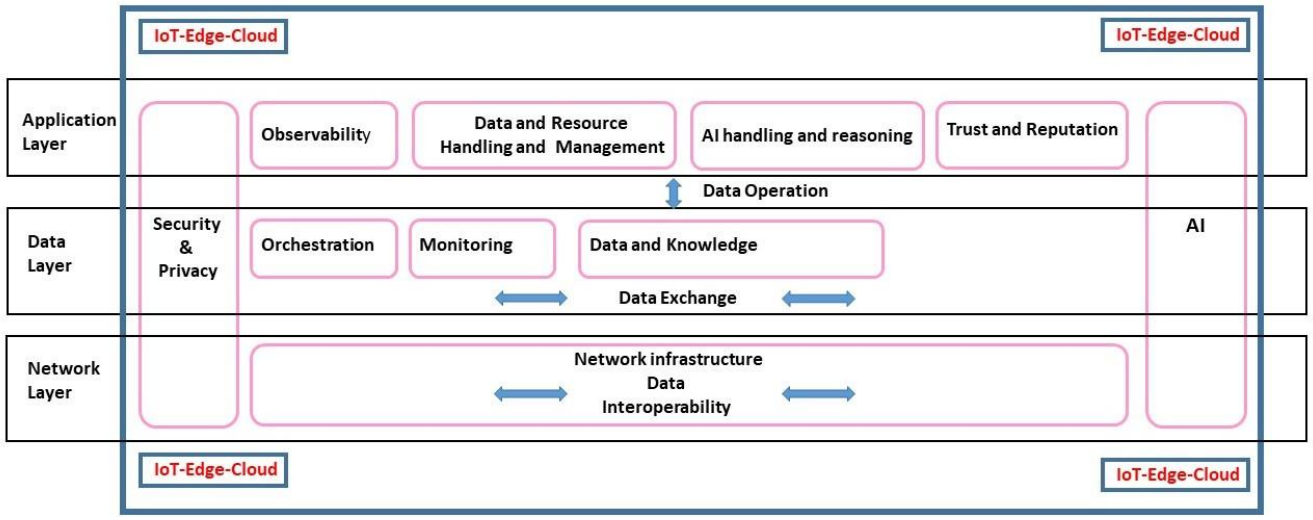

Figure 70: Initial HLA representation of CEI Reference Architecture

The CEI Reference Architecture is composed of building blocks developed in several CEI EU funded projects. Section 8.5.1 describes an initial mapping of different architectures developed in some CEI EU funded projects into the CEI Reference architecture.

Currently, the mapping of HLA to the CEI Reference architecture is not distinguishing whether the CEI reference architecture building blocks are distributed among IoT, Edge and Cloud domains. More details on this distribution will be provided in future versions of this AIOTI HLA report.

Note that a preliminary distribution of the CEI reference architecture building blocks among IoT, Edge and Cloud domains is provided in Section 8.5.1.2.1 and in Section 8.5.1.3.1 of this report and in the AIOTI report "Towards a Computing Continuum Reference Architecture", May 2025.

9. Artificial Intelligence for IoT

Artificial Intelligence covers a broad range of techniques that can be loosely divided into symbolic approaches and those based upon artificial neural networks. Symbolic approaches include the use of graph databases and ontologies for information models, and rules for reasoning. Artificial neural networks are trained on large datasets including text, images, video, audio and structured data. Federated machine learning distributes training to the Edge, avoiding the need to transfer sensitive data to the Cloud.

The integration of Artificial Intelligence within the Internet of Things represents a significant leap forward in the technological capabilities of IoT systems and applications. This section discusses the relationship between AI and IoT, highlighting the transformative potential AI brings to IoT ecosystems. Furthermore, in this section we address the importance of standards and operational frameworks which are necessary for the successful deployment of AI in IoT systems.

9.1 Data, Information and Knowledge

Whilst the IoT is often framed in terms of data, that ignores the importance of knowledge. Data is raw facts and figures. To become useful information, data needs to be structured, organised and processed. Knowledge is a step further that combines information with metadata such as data handling policies, ontologies and integrity constraints. Knowledge is actionable information.

A knowledge centred perspective layers on top of the virtualisation of sensors and actuators. What are the sensors and actuators, where are they located, what do they measure and in what units? How is the information they provide used to support functional requirements? This includes the means to make sense of multiple sources of knowledge, at different levels of abstraction, in terms of processes and significant events.

Similarly, goals and intents can be translated into concrete actions for controlling systems, e.g. the timing of traffic lights during the rush hour, the movement of a robot arm, or the operation of an entire production process on the factory floor. AI supports the knowledge centred perspective, e.g. providing meaningful interpretations of multiple video streams for city traffic, and techniques for optimising the operation of complex systems.

9.2 Reasoning

Reasoning is the process of drawing inferences from new and existing information. Logical reasoning is based upon deductive proof, see e.g., **Reasoning2010**. Other forms of reasoning are rationally plausible based upon prior knowledge, and may need revision as new knowledge is acquired, see e.g., **W3C-Plausible.** This includes inductive reasoning that generalises across a series of observations, analogical reasoning based upon similarities between different cases, and abductive reasoning that seeks the best explanations for some given observations, or to formulate plans for the best ways to achieve given goals. Plausible reasoning is an extension from logic to argumentation, drawing upon metadata to assess certainty of inferences. This includes probabilistic and qualitative approaches, see e.g., **Qualitative-Reasoning**, **Qualitative-Survey**. Plausible reasoning is more general than logic in that it supports knowledge that is uncertain, context sensitive, imprecise, incomplete, inconsistent and subject to change.

9.3 The Role of AI in Enhancing IoT Capabilities

AI technologies such as machine learning and deep learning can provide the means to analyse the vast amounts of data generated by IoT devices. This can enable more intelligent decision-making processes and automating complex tasks without human intervention. Thus, the synergy between AI and IoT not only enhances the efficiency and effectiveness of IoT solutions but also unlocks new opportunities for innovation across various sectors.

9.4 The Role of AI for Control of IoT

In addition to analysing data from IoT devices, AI can be applied to controlling IoT systems through the use of lightweight notations for describing system behaviour in terms of the abstractions presented by IoT virtualisation. This includes dynamic orchestration for optimising utilisation of resources across the compute continuum. Machine learning can be applied to train real-time models of behaviour, including recognising and acting upon events.

9.5 Importance of Frameworks and Standards for AI in IoT

To realise the full potential of AI in IoT, it is essential to establish robust frameworks and standards that ensure interoperability, security, and privacy. These frameworks should facilitate seamless data exchange among diverse IoT devices and platforms while providing a secure environment for AI algorithms to operate.

Building on the IoT-enabled Data Marketplaces architecture presented in section 8.3, AI for IoT requires standardised interfaces and protocols that enable the efficient aggregation, processing, and analysis of IoT data. Such standards are crucial for developing AI applications that can adapt to the dynamic nature of IoT environments and support the diverse roles within the IoT ecosystem, from data providers and consumers to framework providers and application developers.

Furthermore, the deployment of AI in IoT contexts must address the governance challenges identified in section 8.3.4, ensuring that AI applications adhere to ethical guidelines, respect user privacy, and contribute to fair and sustainable economic models. The development and implementation of AI technologies in IoT should be guided by principles that promote transparency, accountability, and inclusivity, ensuring that the benefits of AI are accessible to all stakeholders within the IoT ecosystem.

In conclusion, the integration of AI technologies within the IoT offers the promise of more intelligent, efficient, and responsive systems that can better address the complex challenges of modern societies. Therefore, it is crucial to develop and adhere to comprehensive frameworks and standards that support the ethical, secure, and effective deployment of AI in IoT applications.

9.6 Generative AI, AI Agents and Agentic AI

This section is largely based on the findings coming from the following survey papers: [SaRo25], [RaHo25] and [Pati25]. Several concepts related to AI are listed and briefly explained.

Artificial Intelligence (AI) can be considered as being an essential tool for solving challenging problems in business analytics and decision-making. AI applications are expected to be applied in every possible industrial sector and to affect all aspects of society [Sagr19].

In particular, the release of ChatGPT in November 2022 can be considered as a key contribution on the development and public perception of AI, stimulating its adoption, industrial investment and research activities.

9.6.1 Key AI concepts

Key AI concepts that play a significant role in the development of AI are:

Large Learning Models (LLMs): such as GPT-4 [AcAd23], PaLM [ChNa23] DeepSeek and Kimi K2 are trained on massive datasets of text from books, web content, and other types of contents. These types of models have large capabilities related to natural language understanding, answering questions, summaries and summarization, dialogue logic, and in some cases as well symbolic reasoning [RoTs25a]. Within the AI Agent architectures, LLMs serve as the primary decision-making engine, allowing the agent to parse user queries, plan multi-step solutions, and generate human-like responses. For example, an AI agent used for customer support that is powered by the GPT-4 LLM is able to interpret customer complaints, query backend systems using tool integration, and could then respond in a contextually appropriate and emotionally aware manner [RoTs25b].

Large Image Models (LIMs): such as CLIP [RaKi21] and BLIP-2 [LiLi23] DeepSeek Janus-Pro-7B extend the agent's capabilities into the visual domain. Trained on pairs of image-text, the LIMs can enable perception-based tasks including (a) image classification, (b) object detection, and (3) vision-language grounding. Such capabilities are becoming essential for agents operating in domains such as autonomous vehicles [ElAs25], [PaLe24], robotics [SoZh23], and visual content moderation [AhKh24].

Generative AI: In particular, the success of ChatGPT promoted the use of Generative Agents, which are LLM (Large Language Model) based systems, developed to produce outputs such as text,I mages, and program code from user prompts, i.e., explicit, visible input a user provides to an AI model to request a specific task or information, see e.g., [LuAl24], [ZhCh24].

Such Generative AI agents were quite rapidly adopted across several applications, ranging from conversational assistants (e.g., GitHub Copilot [PeKa23]) and platforms for content generation (e.g., Jasper [LiLa19]) towards creative tools (e.g., Midjourney [Ja-Ro22]). These developments were considered as significant improvements in sectors, such as digital design, marketing, and software prototyping throughout 2023 and beyond.

The generative models are highly communicative, but they are having a reactive behavior, since they are able to produce output only when they are explicitly user prompted and are mainly not able to function autonomously or engage in self-initiated reasoning [GaBa23], [PePa24].

Some Key Characteristics of Generative AI are:

- **Reactivity**: Their operations are triggered by user-specified prompts and they as well lack internal states, persistent memory, or goal/objective following mechanisms, see e.g., [LiWa23], [AlMi24].

- **Multi-modal Capability**: Modern generative systems can produce a diverse array of outputs, including coherent narratives, executable code, realistic images, and even speech transcripts

- **User Prompt Dependency and Statelessness**: Typical generative systems are stateless in that they do not retain context across interactions unless explicitly prompted, see e.g., [DeLe24], [ChLe24]. Although, recent advancements like GPT-4.1 support larger context windows-up to 1 million tokens-and are better able to utilize that context enabled by the improved long-text comprehension, see e.g., [OpenAI25]. Furthermore, Generative AI design also lacks intrinsic feedback loops e.g., [PaJo24], state management, e.g., [Nabb24], or multi-step planning a requirement for autonomous decision-making and iterative goal refinement, e.g., [WeZh25].

Although Generative AI systems have a remarkable generative fidelity, such systems are constrained by their inability to act upon the environment or manipulate digital tools independently. Examples are: inability to (1) search the internet, (2) parse real-time data, or (3) interact with APIs without human intervention or other supporting tools.

Currently, the AI landscape experienced a quite rapid transformation, evolving from the use of standalone LLMs toward more autonomous, and more task-oriented platforms [S'aCu24]. This evolution experienced two major post-generative phases: AI Agents and Agentic AI.

AI Agents: In particular, the limitations of generative AI on being constrained in acting upon the environment or manipulate digital tools independently, such as handling dynamic tasks, maintaining state continuity, or executing multi-step plans, have led to the development of AI tool-augmented systems, commonly referred to as AI Agents, see e.g. [BaAs23].

The AI agents build upon the language processing backbone of LLMs, but they support additional infrastructure such as memory buffers, tool-calling APIs, reasoning chains, and planning routines able to bridge the gaps between the passive response generation and successful active task completion. This architectural evolution marks a critical shift in AI system design: from content creation to autonomous task execution, see e.g., [LiDu25], [GuCh25]. The evolution from generative systems to AI Agents identifies a progressive layering of functionality, which ultimately enables the development of agentic behaviours.

Figure 71 shows the three key characteristics of AI Agents, which are:

- **Autonomy**: A key feature of AI Agents is their ability to function with minimal or no human intervention after their deployment [HaSc23].

- **Task-Specificity**: AI Agents are purpose-built for narrow, and well-defined tasks [Kris25], [PaSh25]. They are in particular, optimized to execute repeatable operations within a fixed domain, such as email filtering [EzSh24], [SiPa20], database querying [KhSa24], or calendar coordination [BuWi25], [Enda24].

- **Reactivity and Adaptation**: AI Agents often include basic mechanisms for interacting with dynamic inputs, allowing them to respond to real-time stimuli such as user requests, external API calls, or state changes in software environments [DeGu25], [RaMe24]. Some AI agent systems integrate basic learning capabilities [PaSk07] or integrate updated context buffers to refine behavior over time, particularly in settings like personalized recommendations or conversation flow management [KaSt24], [HuLi23], [BaAl22].

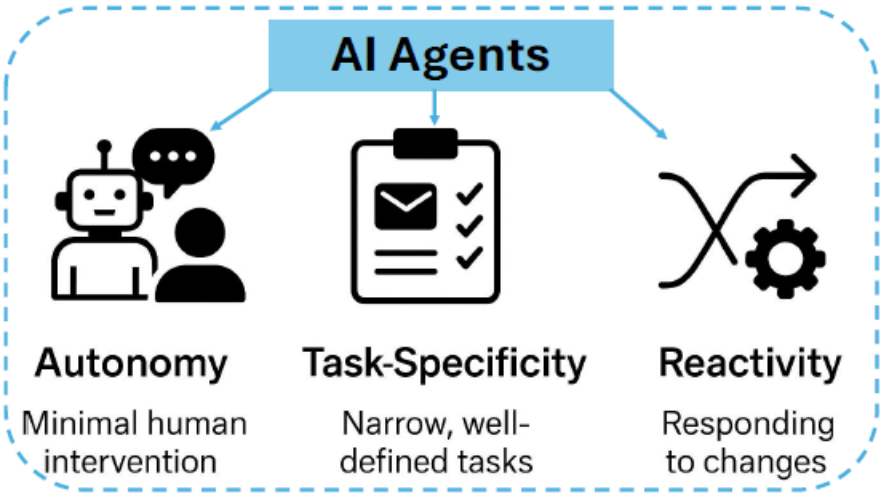

Figure 71: Key characteristics of AI Agents autonomy, task-specificity, and reactivity for agent design and operational behavior, copied from [SaRo25]

Agentic AI: As use cases increasingly require context retention, task interdependence, and as well adaptability across dynamic environments, the single AI agent model proves insufficient [PeLi24], [ShSh25]. In particular, although AI Agents represent a significant improvement in artificial intelligence capabilities, such as in automating narrow tasks through tool-augmented reasoning, current research studies identify significant limitations that constrain their applicability. Such constraints are scalability in complex, dynamic, multi-step, and/or cooperative scenarios [WuYu23], [FeXu25]. The Agentic AI paradigm, is a more advanced mechanism that can alleviate the AI Agent constraints. In particular, Agentic AI extends the capabilities of traditional AI Agents by enabling multiple specialized AI agents to collaborate and complete goals and tasks through collaborative reasoning and multi-step planning [NiLi25], structured communication see e.g., [ZhTa25], [MiRa25], shared memory [XuLi25], [RiCr25], and dynamic role assignment [AcKu25].

Currently, there are research and standardization activities (in e.g., 3GPP) on investigating how AI agents and Agentic AI systems can be applied and used in cellular (e.g., 6G) core networks, see e.g., [ChSu25], [LiSh25], [MoHo25], [FiAt22], [KhSa22], [LeGr22], [3GPP TR 22.870].

In particular, the AI Agent acts as a deterministic component with limited scope, while Agentic AI reflects distributed intelligence, characterized by goal decomposition, inter-agent communication, and contextual adaptation, demonstrating key characteristics of the modern agentic AI frameworks.

Examples of Agent to Agent communication protocols are:

- MCP (Model Context Protocol), see: https://github.com/modelcontextprotocol
- A2A (Agent to Agent) protocol, see: https://a2aprotocol.ai/ and https://a2aprotocol.ai/blog/impact-analysis-google-donating-a2a-protocol-linux-foundation

Key differences between AI agents and Agentic AI functionalities

An example of a conceptual illustration that shows the distinction between AI Agents and Agentic AI through the analogy of smart home systems is shown in **Figure 72**.

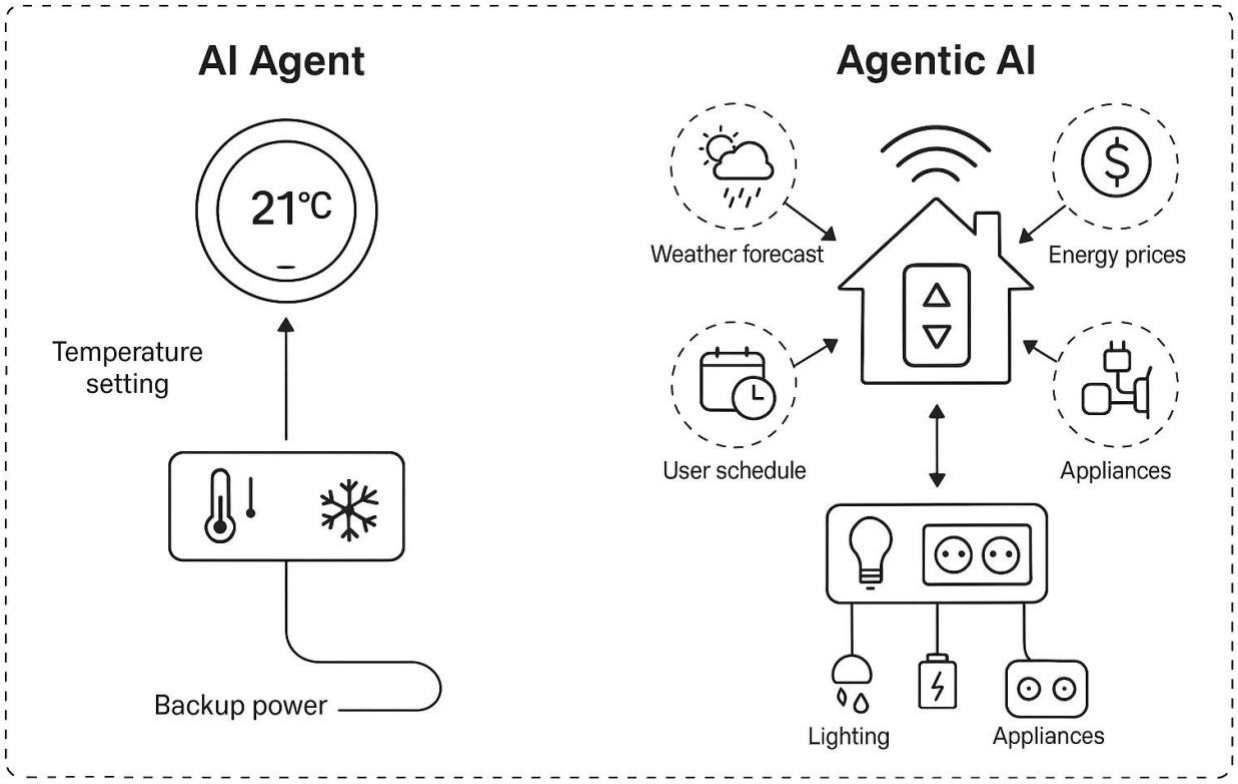

Figure 72: Comparative illustration of AI Agent vs. Agentic AI synthesizing conceptual distinctions. Left: A single-task AI Agent. Right A multi-agent Agentic AI system, copied from [SaRo25]

As depicted in **Figure 72**, the left part of the figure depicts a traditional AI Agent in the form of a smart thermostat.

This standalone AI agent receives a user-defined temperature setting and then autonomously controls the heating or cooling system to maintain the target temperature. While the right part of the same figure depicts an Agentic AI system that is embedded in a comprehensive smart home ecosystem. In this Agentic AI system, multiple specialized AI agents interact synergistically in order to manage diverse aspects such as (1) user daily schedule, being aware about the user presence or absence, (2) weather forecasting, (3) energy pricing optimization, (4) security monitoring, and (4) appliance management. Important to note that these agents are not just reactive modules. In particular, (1) they communicate dynamically, (2) share memory states, and (3) collaboratively align actions toward a high-level system goal (such as optimizing comfort, safety, and energy efficiency in real-time). As example, the AI agent that controls the weather forecast task might signal upcoming heatwaves, and in this way prompting the appliance management AI agent to start early pre-cooling via solar energy before peak pricing hours, using the information that was communicated previously, via the energy pricing AI agent). At the same time, the system might delay high-energy tasks (appliance management AI agent) or activate surveillance systems (during occupant absence, integrating decisions across domains.

Figure 72 illustrates that a single AI Agent acts as a deterministic component with limited scope, while Agentic AI reflects distributed intelligence, characterized by goal/task decomposition, inter-agent communication, and contextual adaptation, demonstrating key characteristics of the modern agentic AI frameworks.

While AI Agents and Agentic AI systems represent increasingly autonomous and interactive systems, both paradigms utilize generative architectures as their foundations, especially LLMs and LIMs and

generative AI. The key differences between AI Agents and Agentic AI systems are provided in Table 3, in terms of scope, autonomy, architectural composition, coordination strategy, and operational complexity, which is based on the work published in [SaRo25] and derived from AutoGen [WuBa23] and ChatDev [QiLi23]. In particular, Table 3 compares their definitions, levels of autonomy, capacity for handling task complexity, collaboration styles, learning and adaptation scope and typical application domains.

Table 3: Key Structural, Functional, and Operational Differences Between AI Agents and Agentic AI Systems, based on [SaRo25]

Feature	AI Agent	Agentic AI
Definition	Autonomous software programs that perform specific task	Systems of multiple AI agents collaborating to achieve complex goals
Autonomy Level	High autonomy within specific tasks.	Broad level of autonomy with the ability to manage multi-step, complex tasks and systems.
Task Complexity	Typically handle single, specific tasks.	Handle complex, multi-step tasks requiring coordination.
Collaboration	Operate independently.	Involve multi-agent information sharing, collaboration and cooperation.
Learning and Adaptation	Learn and adapt within their specific domain.	Learn and adapt across a wider range of tasks and environments.
Applications	Customer service chatbots, virtual assistants, automated workflows.	Supply chain management, business process optimization, virtual project managers.

9.6.2 Architectural evolution from traditional AI agents to Agentic AI systems

Agentic AI systems inherit the modularity of AI Agents but extend their architecture to support distributed intelligence, inter-agent communication, and iterative planning.

Figure 73, depicts the architectural evolution from traditional AI Agents to modern Agentic AI systems. The left part of the figure shows the AI Agent key capabilities, such as Perception, Reasoning and Action, and expands, in the right part of the figure to the Agentic AI advanced capabilities, including Specialized Agents, Advanced Reasoning & Planning, Persistent Memory, and Orchestration. **Figure 73** depicts as well emergent properties such as Multi-Agent Collaboration, System Coordination, Shared Context, and Task Decomposition, and all are enclosed within a dotted boundary signifying layered modularity and the transition to distributed, adaptive agentic AI intelligence.

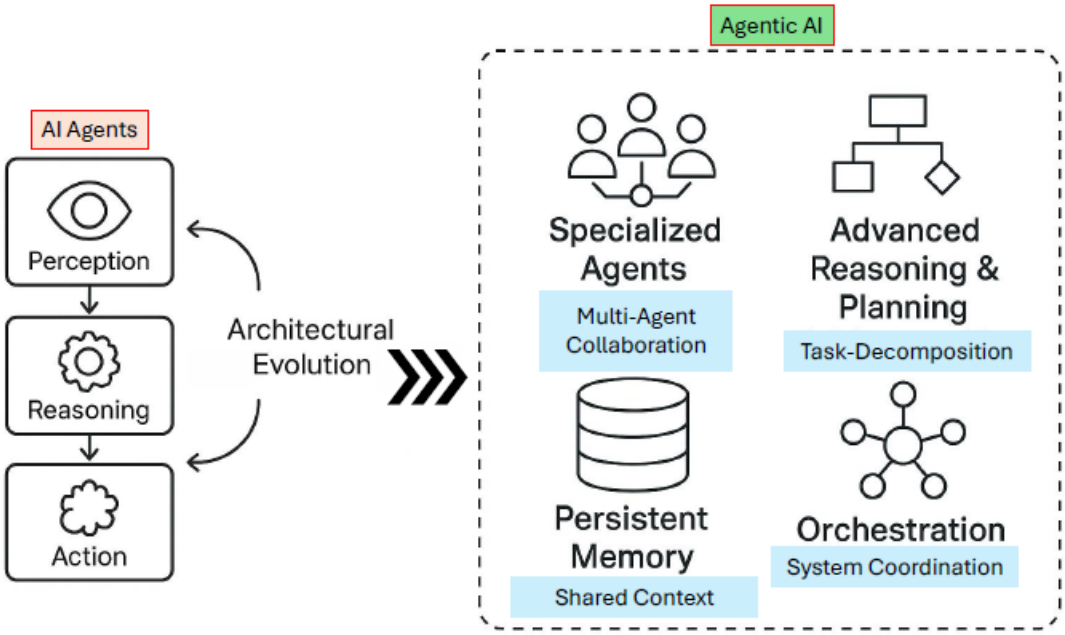

Figure 73: Architectural evolution from traditional AI Agents to modern Agentic Ai systems, copied from [SaRo25]

9.6.3 *Application of* AI Agents and Agentic AI

A significant categorization of AI Agents and Agentic AI application has been performed and documented in *[SaRo25]*. *This categorization is depicted in* **Figure 74** *and in detail outlined in* **Table 4** *and*

Table 5, *see [SaRo25] for details.*

In particular, for AI Agents, four primary use cases are reviewed:

1) Customer Support Automation and Internal Enterprise Search, where single-agent models handle structured queries and response generation;
2) Email Filtering and Prioritization, where agents assist users in managing high-volume communication through classification heuristics;
3) Personalized Content Recommendation and Basic Data Reporting, where user behavior is analysed for automated insights;
4) Autonomous Scheduling Assistants, which interpret calendars and book tasks with minimal user input.

In contrast, Agentic AI applications encompass broader and more dynamic capabilities, reviewed and discussed in four categories as well:

1) Multi-Agent Research Assistants that retrieve, synthesize, and draft scientific content collaboratively;
2) Intelligent Robotics Coordination, including drone and multi-robot systems in fields like agriculture and logistics;
3) Collaborative Medical Decision Support, involving diagnostic, treatment, and monitoring subsystems;
4) Multi-Agent Game AI and Adaptive Workflow Automation, where decentralized agents interact strategically or handle complex task pipelines.

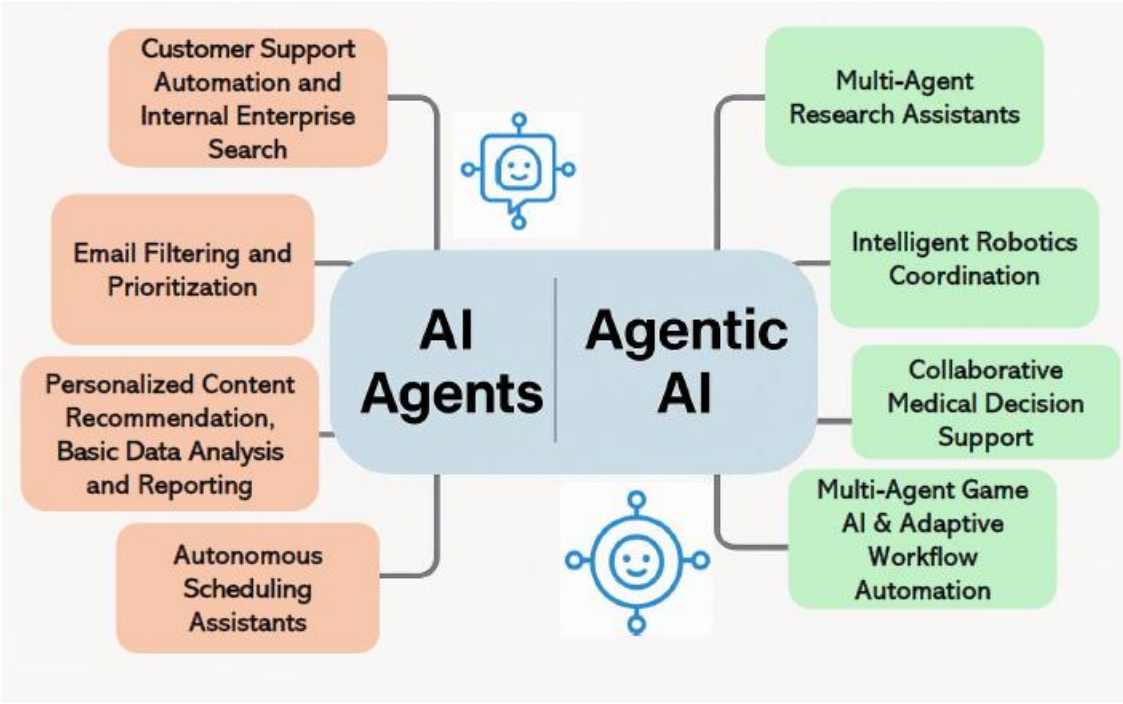

Figure 74: Categorized applications of AI Agents and Agentic AI across eight core functional domains, copied from [SaRo25]

The Application areas and Operational Characteristics for AI Agents and for Agentic AI are briefly listed in **Table 4** and

Table 5, respectively.

Table 4: Representative AI Agents (2023–2025): Applications and Operational Characteristics, based on [SaRo25] and [WuBa23]

Model / Reference	Application Area	Operation as AI Agent
ChatGPT Deep Research Mode **OpenAI (2025)** Source Link	Research Analysis / Reporting	Synthesizes hundreds of sources into reports; functions as a self-directed research analyst.
Operator **OpenAI (2025)** Source Link	Web Automation	Navigates websites, fills forms, and completes online tasks autonomously.
Agentspace: Deep Research Agent **Google (2025)** Source Link	Enterprise Reporting	Generates business intelligence reports using Gemini models.
NotebookLM Plus Agent **Google (2025)** Source Link	Knowledge Management	Summarizes, organizes, and retrieves data across Google Workspace apps.
Nova Act **Amazon (2025)** Source Link	Workflow Automation	Automates browser-based tasks such as scheduling, HR requests, and email.
Manus Agent **Monica (2025)** Source Link	Personal Task Automation	Executes trip planning, site building, and product comparisons via browsing.
Harvey **Harvey AI (2025)** Source Link	Legal Automation	Automates document drafting, legal review, and predictive case analysis.
Otter Meeting Agent **Otter.ai (2025)** Source Link	Meeting Management	Transcribes meetings and provides highlights, summaries, and action items.
Otter Sales Agent **Otter.ai (2025)** Source Link	Sales Enablement	Analyses sales calls, extracts insights, and suggests follow-ups.
ClickUp Brain **ClickUp (2025)** Source Link	Project Management	Automates task tracking, updates, and project workflows.
Agentforce **Agentforce (2025)** Source Link	Customer Support	Routes tickets and generates context-aware replies for support teams.
Microsoft Copilot **Microsoft (2024)** Source Link	Office Productivity	Automates writing, formula generation, and summarization in Microsoft 365.

Project Astra **Google DeepMind** **(2025)** Source Link	Multimodal Assistance	Processes text, image, audio, and video for task support and recommendations.
Claude 3.5 Agent **Anthropic (2025)** Source Link	Enterprise Assistance	Uses multimodal input for reasoning, personalization, and enterprise task completion.

Table 5: Representative Agentic AI Models (2023–2025): Applications and Operational Characteristics, based on [SaRo25] and [QiLi23]

Model / Reference	Application Area	Operation as Agentic AI
Auto-GPT (2023) Source Link **[YaYu23]**	Task Automation	Decomposes high-level goals, executes subtasks via tools/APIs, and iteratively self-corrects.
GPT Engineer Open Source (2023) Source Link	Code Generation	Builds entire codebases: plans, writes, tests, and refines based on output.
MetaGPT (2023), Source Link **[HoZh23])**	Software Collaboration	Coordinates specialized agents (e.g., coder, tester) for modular multi-role project development.
BabyAGI Nakajima (2024) Source Link	Project Management	Continuously creates, prioritizes, and executes subtasks to adaptively meet user goals.
Voyager Wang et al. (2023), Source Link **[WaXi23]**	Game Exploration	Learns in Minecraft, invents new skills, sets subgoals, and adapts strategy in real time.
CAMEL Liu et al. (2023), Source Link **[LiHa23]**	Multi-Agent Simulation	Simulates agent societies with communication, negotiation, and emergent collaborative behavior.
Einstein Copilot Salesforce (2024) Source Link	Customer Automation	Automates full support workflows, escalates issues, and improves via feedback loops.
Copilot Studio (Agentic Mode) Microsoft (2025) Source Link	Productivity Automation	Manages documents, meetings, and projects across Microsoft 365 with adaptive orchestration.
Atera AI Copilot Atera (2025) Source Link	IT Operations	Diagnoses/resolves IT issues, automates ticketing, and learns from evolving infrastructures.
AES Safety Audit Agent AES (2025) Source Link	Industrial Safety	Automates audits, assesses compliance, and evolves strategies to enhance safety outcomes.
DeepMind Gato (Agentic Mode) Reed et al. (2022), Source Link **[ReZo22]**	General Robotics	Performs varied tasks across modalities, dynamically learns, plans, and executes.
GPT-4o + Plugins OpenAI (2024) Source Link	Enterprise Automation	Manages complex workflows, integrates external tools, and executes adaptive decisions.

9.6.4 Examples of Standardisation activities on Agentic AI

Currently several standardization activities related Agentic AI are being investigated and some of them already started. The SDOs that are currently cover the Agentic AI topic are:

3GPP (examples):

- 3GPP 6G Work Items discussions on Agentic core network: during a 3GPP TSG SA WG2 meeting that took place in Goteborg, Sweden in August 2025, a list 6G Work items focusing on Agentic core networks for 6G were discussed.
- 3GPP TR 22.870 "Study on 6G Use Cases and Service Requirements": a Technical Report that discusses among others Agentic AI 6G use cases.

ETSI (examples):

- ETSI GR ENI 051 V4.1.1 (2025-02), "Study on AI Agents based Next-generation Network Slicing"

IETF (examples):

- IETF draft-rosenberg-ai-protocols-00, "Framework, Use Cases and Requirements for AI Agent"
- IETF draft-stephan-ai-agent-6g-00, "AI Agent protocols for 6G systems"
- IETF draft-du-ai-agent-communication-6g-aspect-00, "Use Cases and Requirements of AI Agent Communication from 6G Aspect"
- IETF draft-yu-ai-agent-use-cases-in-6g-01, "AI Agent Use Cases and Requirements in 6G Network"
- IETF draft-akhavain-moussa-ai-network-00, "AI Network for Training, Inference, and Agentic Interactions"
- IETF draft-hw-ai-agent-6g-00, "AI Network for Training, Inference, and Agentic Interactions"
- IETF draft-campbell-agentic-http-00, "A Best Current Practice for Agentic Interactions over HTTP"
- IETF draft-wahl-scim-agent-schema-01, "System for Cross-domain Identity Management: Agentic Identity Schema"
- IETF draft-rosenberg-oauth-aauth-00, "AAuth - Agentic Authorization OAuth 2.1 Extension"
- IETF draft-huang-acme-scalable-agent-enrollment-00, "Extending Certificate Enrollment Protocols for Scalable Agentic AI Identity"
- IETF draft-huang-rats-agentic-eat-cap-attest-00, "Capability Attestation Extensions for the Entity Attestation Token (EAT) in Agentic AI Systems"
- IETF draft-zhao-nmop-network-management-agent-02, "AI based Network Management Agent(NMA): Concepts and Architecture"
- IETF draft-yue-anima-agent-recovery-networks-00, "Task-Oriented Multi-Agent Recovery Framework for High-Reliability in Converged Mobile Networks"
- IETF draft-liu-agent-context-protocol-00, "Agent Context Protocol"
- IETF draft-yang-ioa-protocol-00, "Internet of Agents Protocol (IoA Protocol) for Heterogeneous Agent Collaboration"
- IETF draft-jiang-cats-reference-acn-00, "CATS Reference Model for AI-Agent Communication Network"
- IETF draft-zhang-agent-gap-network-00, "Problem Statement and Gap Analysis for Agent-enabled Mobile Core Network"

9.6.5 References

[SaRo25] R. Sapkota, K. I. Roumeliotis, M. Karkee "AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges", Information Fusion, 2025, see: https://doi.org/10.1016/j.inffus.2025.103599

[RaHo25] T. Raheem; G. Hossain, "Agentic AI Systems: Opportunities, Challenges, and Trustworthiness", IEEE International Conference on Electro Information Technology (eIT) 2025, see: https://ieeexplore.ieee.org/abstract/document/11103638

[Pati25] A. Kumar Pati, "Agentic AI: A Comprehensive Survey of Technologies, Applications, and Societal Implications", IEEE Access, 2025, se: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=11071266

[Sagr19]] Sagramsingh R. AI: a global survey. Washington, DC: IEEE-USA; 2019.

[AcKu25] D. B. Acharya, K. Kuppan, and B. Divya, "Agentic ai: Autonomous intelligence for complex goals–a comprehensive survey," IEEE Access, 2025.

[ChSu25] Ziqi Chen, Qi Sun, Nan Li, Xiang Li, Yang Wang, Chih-Lin, "Enabling Mobile AI Agent in 6G Era: Architecture and Key Technologies", IEEE Network, Vol. 38, Issue 5, pp 66-75, 2025, see: https://dl.acm.org/toc/ieeenetw/2024/38/5

[LiSh25] Xu Li, Weisen Shi, Hang Zhang, Chenghui Peng, Shaoyun Wu, Wen Tong, "The Agentic-AI Core: An AI-Empowered, Mission-Oriented Core Network for Next-Generation Mobile Telecommunications", Elsevier, Engineering 2025, see: https://www.sciencedirect.com/science/article/pii/S209580992500325X?via%3Dihub

[MoHo25] H.G. Moussa, A. Akhavain, S. M. Hosseini, B. McCormick, "Distributed Learning and Inference Systems: A Networking Perspective", IEEE Network, 2025, see: https://ieeexplore.ieee.org/document/11015802

[FiAt22] C. Fiandrino, G. Attanasio, M. Fiore, J. Widmer, "Toward native explainable and robust AI in 6G networks: Current state, challenges and road ahead," Computer Communications, Volume 193, 2022, see: https://www.sciencedirect.com/science/article/abs/pii/S0140366422002389

[KhSa22] L. U. Khan, W. Saad, D. Niyato, Z. Han and C. S. Hong, "Digital-Twin-Enabled 6G: Vision, Architectural Trends, and Future Directions," in IEEE Communications Magazine, vol. 60, no. 1, pp. 74-80, January 2022

[LeGr22] K.-D. Lee and C. Gray-Preston, "Everyday Living Assisted by 6G Applications and Solutions," IEEE Wireless Communications Magazine, October 2022. (Everyday Living Assisted by 6G Applications and Solutions | IEEE Journals & Magazine | IEEE Xplore

[3GPP TR 22.870] 3GPP TR 22.870 "Study on 6G Use Cases and Service Requirements", 3GPP Release 20, June 2025, see: https://www.3gpp.org/ftp/Specs/archive/22_series/22.870/22870-031.zip

[DeGu25] Z. Deng, Y. Guo, C. Han, W. Ma, J. Xiong, S. Wen, and Y. Xiang, "Ai agents under threat: A survey of key security challenges and future pathways," ACM Computing Surveys, vol. 57, no. 7, pp. 1–36, 2025.

[S'aCu24] J. S´anchez Cuadrado, S. P´erez-Soler, E. Guerra, and J. De Lara, "Automating the development of task-oriented llm-based chatbots,"Interfaces, pp. 1–10, 2024.

[LuAl24] Y. Lu, A. Aleta, C. Du, L. Shi, and Y. Moreno, "Llms and generative agent-based models for complex systems research," Physics of Life Reviews, 2024.

[ZhCh24] A. Zhang, Y. Chen, L. Sheng, X. Wang, and T.-S. Chua, "On generative agents in recommendation," in Proceedings of the 47th international ACM SIGIR conference on research and development in Information Retrieval, pp. 1807–1817, 2024.

[PeKa23] S. Peng, E. Kalliamvakou, P. Cihon, and M. Demirer, "The impact of ai on developer productivity: Evidence from github copilot," arXiv preprint arXiv:2302.06590, 2023.

[LiLa19] J. Li, V. Lavrukhin, B. Ginsburg, R. Leary, O. Kuchaiev, J. M. Cohen, H. Nguyen, and R. T. Gadde, "Jasper: An end-to-end convolutional neural acoustic model," arXiv preprint arXiv:1904.03288, 2019.

[Ja-Ro22] A. Jaruga-Rozdolska, "Artificial intelligence as part of future practices in the architect's work: Midjourney generative tool as part of a process of creating an architectural form," Architectus, no. 3 (71, pp. 95–104, 2022.

[YaYu23] H. Yang, S. Yue, and Y. He, "Auto-gpt for online decision making: Benchmarks and additional opinions," arXiv preprint arXiv:2306.02224, 2023, see: https://arxiv.org/abs/2306.02224

[BoWi25] P. Bornet, J. Wirtz, T. H. Davenport, D. De Cremer, B. Evergreen, P. Fersht, R. Gohel, S. Khiyara, P. Sund, and N. Mullakara, Agentic Artificial Intelligence: Harnessing AI Agents to Reinvent Business, Work and Life. Irreplaceable Publishing, 2025.

[HaSc23] A. I. Hauptman, B. G. Schelble, N. J. McNeese, and K. C. Madathil, "Adapt and overcome: Perceptions of adaptive autonomous agents for human-ai teaming," Computers in Human Behavior, vol. 138, p. 107451, 2023.

[Kris25] N. Krishnan, "Advancing multi-agent systems through model context protocol: Architecture, implementation, and applications," arXiv preprint arXiv:2504.21030, 2025.

[PaSh25] H. Padigela, C. Shah, and D. Juyal, "Ml-dev-bench: Comparative analysis of ai agents on ml development workflows," arXiv preprint arXiv:2502.00964, 2025.

[EzSh24] C. S. Eze and L. Shamir, "Analysis and prevention of ai-based phishing email attacks," Electronics, vol. 13, no. 10, p. 1839, 2024.

[SiPa20] D. Singh, V. Patel, D. Bose, and A. Sharma, "Enhancing email marketing efficacy through ai-driven personalization: Leveraging natural language processing and collaborative filtering algorithms," International Journal of AI Advancements, vol. 9, no. 4, 2020.

[KhSa24] R. Khan, S. Sarkar, S. K. Mahata, and E. Jose, "Security threats in agentic ai system," arXiv preprint arXiv:2410.14728, 2024.

[Enda24] C. G. Endacott, "Enacting machine agency when ai makes one's day: understanding how users relate to ai communication technologies for no. 4, p. zmae011, 2024.

[RaMe24] M. Raees, I. Meijerink, I. Lykourentzou, V.-J. Khan, and K. Papangelis, "From explainable to interactive ai: A literature review on current trends in human-ai interaction," International Journal of Human-Computer Studies, p. 103301, 2024.

[PaSk07] Z. Pawlak and A. Skowron, "Rudiments of rough sets," Information sciences, vol. 177, no. 1, pp. 3–27, 2007.

[KaSt24] S. Kapoor, B. Stroebl, Z. S. Siegel, N. Nadgir, and A. Narayanan, "Ai agents that matter," arXiv preprint arXiv:2407.01502, 2024.

[HuLi23] X. Huang, J. Lian, Y. Lei, J. Yao, D. Lian, and X. Xie, "Recommender ai agent: Integrating large language models for interactive recommendations," arXiv preprint arXiv:2308.16505, 2023.

[BaAl22] A. M. Baabdullah, A. A. Alalwan, R. S. Algharabat, B. Metri, and N. P. Rana, "Virtual agents and flow experience: An empirical examination of ai-powered chatbots," Technological Forecasting and Social Change, vol. 181, p. 121772, 2022.

[AcAd23] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al., "Gpt-4 technical report," arXiv preprint arXiv:2303.08774, 2023.

[ChNa23] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, et al., "Palm: Scaling language modeling with pathways," Journal of Machine Learning Research, vol. 24, no. 240, pp. 1–113, 2023.

[RoTs25a] K. I. Roumeliotis, N. D. Tselikas, and D. K. Nasiopoulos, "Llms for product classification in e-commerce: A zero-shot comparative study of gpt and claude models," Natural Language Processing Journal, vol. 11, p. 100142, 6 2025.

[RaKi21] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., "Learning transferable visual models from natural language supervision," in International conference on machine learning, pp. 8748–8763, PmLR, 2021.

[RoTs25b] K. I. Roumeliotis, N. D. Tselikas, and D. K. Nasiopoulos, "Think before you classify: The rise of reasoning large language models for consumer complaint detection and classification," Electronics 2025, Vol. 14, Page 1070, vol. 14, p. 1070, 3 2025

[LiLi23] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language image pre-training with frozen image encoders and large language models," in International conference on machine learning, pp. 19730– 19742, PMLR, 2023.

[SoZh23] S. Sontakke, J. Zhang, S. Arnold, K. Pertsch, E. Bıyık, D. Sadigh, C. Finn, and L. Itti, "Roboclip: One demonstration is enough to learn robot policies," Advances in Neural Information Processing Systems, vol. 36, pp. 55681–55693, 2023.

[ElAs25] M. Elhenawy, H. I. Ashqar, A. Rakotonirainy, T. I. Alhadidi, A. Jaber, and M. A. Tami, "Vision-language models for autonomous driving: Clip-based dynamic scene understanding," Electronics, vol. 14, no. 7, p. 1282, 2025.

[PaLe24] S. Park, M. Lee, J. Kang, H. Choi, Y. Park, J. Cho, A. Lee, and D. Kim, "Vlaad: Vision and language assistant for autonomous driving," in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 980 987, 2024.

[AhKh24] S. H. Ahmed, M. J. Khan, and G. Sukthankar, "Enhanced multimodal content moderation of children's videos using audiovisual fusion," arXiv preprint arXiv:2405.06128, 2024.

[WuBa23] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu, et al., "Autogen: Enabling next-gen llm applications via multi-agent conversation," arXiv preprint arXiv:2308.08155, 2023, see: https://www.researchgate.net/publication/373164024_AutoGen_Enabling_Next-Gen_LLM_Applications_via_Multi-Agent_Conversation_Framework

[GaBa23] L. Gabora and J. Bach, "A path to generative artificial selves," in EPIA Conference on Artificial Intelligence, pp. 15–29, Springer, 2023.

[PePa24] G. Pezzulo, T. Parr, P. Cisek, A. Clark, and K. Friston, "Generating meaning: active inference and the scope and limits of passive ai," Trends in Cognitive Sciences, vol. 28, no. 2, pp. 97–112, 2024.

[LiWa23] X. Liu, J. Wang, J. Sun, X. Yuan, G. Dong, P. Di, W. Wang, and D. Wang, "Prompting frameworks for large language models: A survey," arXiv preprint arXiv:2311.12785, 2023.

[AlMi24] K. Alizadeh, S. I. Mirzadeh, D. Belenko, S. Khatamifard, M. Cho, C. C. Del Mundo, M. Rastegari, and M. Farajtabar, "Llm in a flash: Efficient large language model inference with limited memory," in Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 12562–12584, 2024.

[DeLe24] P. Denny, J. Leinonen, J. Prather, A. Luxton-Reilly, T. Amarouche, B. A. Becker, and B. N. Reeves, "Prompt problems: A new programming exercise for the generative ai era," in Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1, pp. 296–302, 2024.

[ChLe24] C. Chen, S. Lee, E. Jang, and S. S. Sundar, "Is your prompt detailed enough? exploring the effects of prompt coaching on users' perceptions, engagement, and trust in text-to-image generative ai tools," in Proceedings of the Second International Symposium on Trustworthy Autonomous Systems, pp. 1–12, 2024.

[OpenAI25] OpenAI, "Introducing gpt-4.1 in the api," 4 2025.

[PaJo24] A. Pan, E. Jones, M. Jagadeesan, and J. Steinhardt, "Feedback loops with language models drive in-context reward hacking," arXiv preprint arXiv:2402.06627, 2024.

[Nabb24] K. Nabben, "Ai as a constituted system: accountability lessons from an llm experiment," Data & policy, vol. 6, p. e57, 2024.

[WeZh25] H. Wei, Z. Zhang, S. He, T. Xia, S. Pan, and F. Liu, "Plangenllms: A modern survey of llm planning capabilities," arXiv preprint arXiv:2502.11221, 2025.

[BaAs23] A. Bandi, P. V. S. R. Adapa, and Y. E. V. P. K. Kuchi, "The power of generative ai: A review of requirements, models, input–output formats, evaluation metrics, and challenges," Future Internet, vol. 15, no. 8, p. 260, 2023.

[LiDu25] Y. Liu, H. Du, D. Niyato, J. Kang, Z. Xiong, Y. Wen, and D. I. Kim, "Generative ai in data center networking: Fundamentals, perspectives, and case study," IEEE Network, 2025.

[GuCh25] C. Guo, F. Cheng, Z. Du, J. Kiessling, J. Ku, S. Li, Z. Li, M. Ma, T. Molom-Ochir, B. Morris, et al., "A survey: Collaborative hardware and software design in the era of large language models," IEEE Circuits and Systems Magazine, vol. 25, no. 1, pp. 35–57, 2025.

[NiLi25] L. Ning, Z. Liang, Z. Jiang, H. Qu, Y. Ding, W. Fan, X.-y. Wei, S. Lin, H. Liu, P. S. Yu, et al., "A survey of webagents: Towards next-generation ai agents for web automation with large foundation models," arXiv preprint arXiv:2503.23350, 2025.

[WuYu23] Z. Wu, C. Yu, C. Chen, J. Hao, and H. H. Zhuo, "Models as agents: Optimizing multi-step predictions of interactive local models in modelbased multi-agent reinforcement learning," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 37, pp. 10435–10443, 2023.

[FeXu25] Z. Feng, R. Xue, L. Yuan, Y. Yu, N. Ding, M. Liu, B. Gao, J. Sun, and G. Wang, "Multi-agent embodied ai: Advances and future directions," arXiv preprint arXiv:2505.05108, 2025.

[ZhTa25] R. Zhang, S. Tang, Y. Liu, D. Niyato, Z. Xiong, S. Sun, S. Mao, and Z. Han, "Toward agentic ai: generative information retrieval inspired intelligent communications and networking," arXiv preprint arXiv:2502.16866, 2025.

[MiRa25] E. Miehling, K. N. Ramamurthy, K. R. Varshney, M. Riemer, D. Bouneffouf, J. T. Richards, A. Dhurandhar, E. M. Daly, M. Hind, P. Sattigeri, et al., "Agentic ai needs a systems theory," arXiv preprint arXiv:2503.00237, 2025.

[XuLi25] W. Xu, Z. Liang, K. Mei, H. Gao, J. Tan, and Y. Zhang, "A-mem: Agentic memory for llm agents," arXiv preprint arXiv:2502.12110, 2025.

[RiCr25] C. Riedl and D. De Cremer, "Ai for collective intelligence," Collective Intelligence, vol. 4, no. 2, p. 26339137251328909, 2025.

[PeLi24] L. Peng, D. Li, Z. Zhang, T. Zhang, A. Huang, S. Yang, and Y. Hu, "Human-ai collaboration: Unraveling the effects of user proficiency and ai agent capability in intelligent decision support systems," International Journal of Industrial Ergonomics, vol. 103, p. 103629, 2024.

[ShSh25] H. Shirado, K. Shimizu, N. A. Christakis, and S. Kasahara, "Realism drives interpersonal reciprocity but yields to ai-assisted egocentrism in a coordination experiment," in Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems, pp. 1–21, 2025.

[QiLi23] C. Qian, W. Liu, H. Liu, N. Chen, Y. Dang, J. Li, C. Yang, W. Chen, Y. Su, X. Cong, et al., "Chatdev: Communicative agents for software development," arXiv preprint arXiv:2307.07924, 2023, see: https://www.researchgate.net/publication/372416377_Communicative_Agents_for_Software_Development

[HoZh23] S. Hong, X. Zheng, J. Chen, Y. Cheng, J. Wang, C. Zhang, Z. Wang, S. K. S. Yau, Z. Lin, L. Zhou, et al., "Metagpt: Meta programming for multi-agent collaborative framework," arXiv preprint arXiv:2308.00352, vol. 3, no. 4, p. 6, 2023, see: https://arxiv.org/abs/2308.00352

[WaXi23] G. Wang, Y. Xie, Y. Jiang, A. Mandlekar, C. Xiao, Y. Zhu, L. Fan, and A. Anandkumar, "Voyager: An open-ended embodied agent with large language models," arXiv preprint arXiv:2305.16291, 2023., see: https://arxiv.org/abs/2305.16291

[LiHa23] G. Li, H. Hammoud, H. Itani, D. Khizbullin, and B. Ghanem, "CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society" pp. 51991–52008, arXiv:2303.17760, 2023 https://arxiv.org/abs/2303.17760

[ReZo22] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, et al., "A generalist agent," arXiv preprint arXiv:2205.06175, 2022, see: https://arxiv.org/abs/2205.06175

10. Highlights and recommendation

In the context of the AIOTI WG Standardisation 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 considers existing SDOs and alliances architecture specifications. This document is an integral part of a set of deliverables from AIOTI WG Standardisation that also cover other aspects such as IoT landscape and Semantic Interoperability.

AIOTI WG Standardisation recommends that the HLA be the basis for further discussion with the relevant EU funded projects, in particular Large-Scale Pilots (LSP) and AIOTI WGs to promote architectural convergence with SDOs (such as ITU-T **SG13** and **SG20**, ISO-IEC JTC1 **SC7** and **SC41**), alliances, consortia and other relevant parties.

In line with the AIOTI WG Standardisation engagement model, other relevant parties include - but are not limited to open-source projects, policy makers, regulators, pilots and testbeds, research organizations, companies.

Based on past discussions within AIOTI WG Standardisation, this Release provides enhancements on the following new or partially developed topics, still with respect to IoT architectural concerns:

- ISO/IEC 30141:2024 IoT - Reference architecture
- Relationship to EUCloudEdgeIoT.eu Open Continuum Reference and Mapping of HLA to Compositional view of the Continuum Reference Architecture, based on results from the following in EUCloudEdgeIoT.eu projects: 6G-Cloud architecture, COGNIT Architecture and CODECO Architecture
- Generative AI, AI Agents and Agentic AI

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

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 75** shows the scope of ETSI SmartBAN.

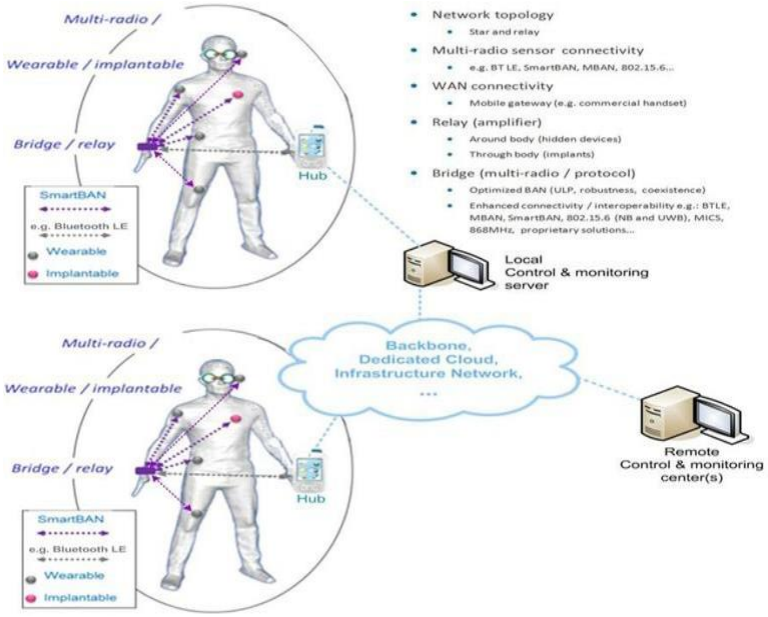

Figure 75: ETSI SmartBAN deployment example concepts

ETSI DTR/SmartBAN-004 reference architecture provides a layered reference architecture for SmartBAN. The reference architecture is depicted in the following **Figure 76** which shows a layered approach with an Application Layer, a Service Layer, a Semantic Layer and a Data provision layer.

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).

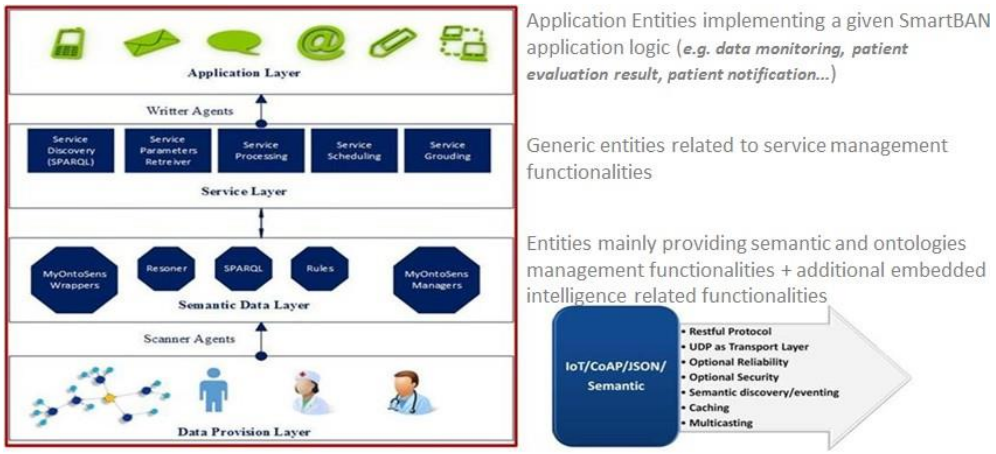

Figure 76: ETSI SmartBAN reference architecture

The following **Figure 77** provides a binding between the ETSI SmartBAN architecture and the AIOTI HLA:

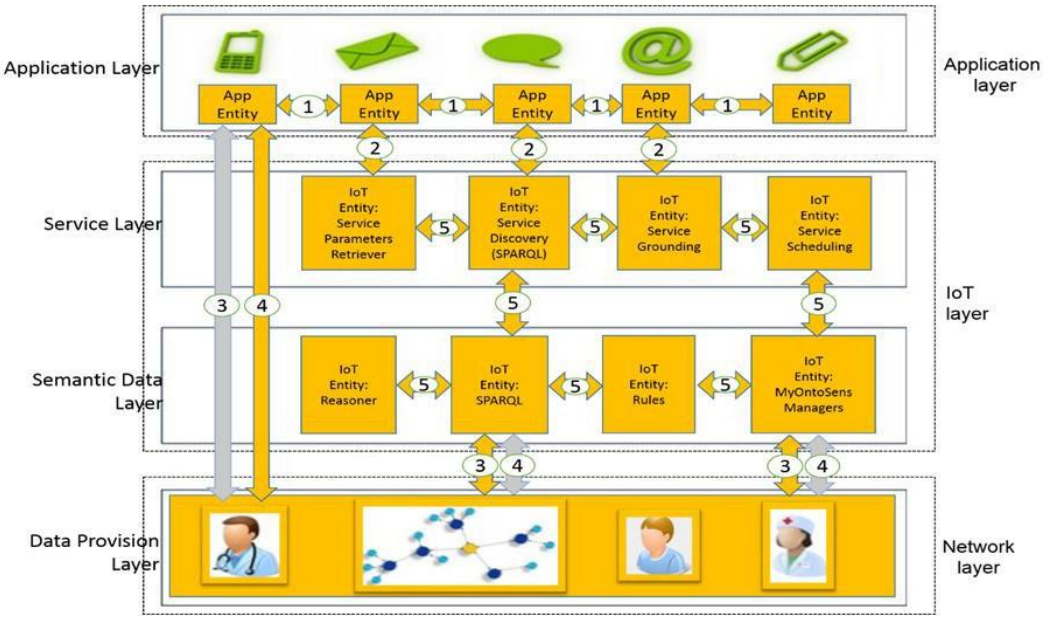

Figure 77: ETSI SmartBAN reference architecture mapping to AIOTI HLA

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 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 6** 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.

Table 6: IoT Gaps mapped on the AIOTI HLA

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

More IoT and Edge standardisation gaps were identified in the following reports:

AIOTI High Priority IoT Standardisation Gaps and Relevant SDOs, Release 3.0, January 2024

AIOTI Report High Priority Edge Computing Standardisation Gaps and Relevant SDOs R2, April 2022

StandICT EU Report IoT Standardisation Gaps

StandICT EU Edge Computing Standardisation Gaps

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

Table 7 below lists some advantages/disadvantages of end device, edge and cloud computing options.

Table 7: Advantages and disadvantages of end device, edge and cloud computing

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 be limited.	**+ –** 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
Costs	**+ –**	**+**	**+**

Topic	End device computing	Edge computing	Cloud computing
	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

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 3, February 2021: https://www.etsi.org/deliver/etsi_ts/118100_118199/118101/03.22.00_60/ts_118101v032200p.pdf

[5] Industrial Internet Reference Architecture, https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/11/IIRA-v1.10.pdf

[6] AIOTI WG Standardisation 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] [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?specific ationId=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_103376v01010 1p.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/januar/GMA_Status_Report_Reference_Architecture_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 WG STANDARDISATION, „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%2Fcomponent%2Frsfiles%2Fdownload-file%2Ffiles%3Fpath%3DDraft_TR%25255CTR-0039-Developer_guide-SDT-based_implementation-V0_0_5.docx%26Itemid%3D238&usg=AOvVaw0EBlHTKC8t5XQdC_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

[48] ISO/IEC FDIS 30141 ED2 Internet of Things (IoT) - Reference architecture. https://www.iec.ch/dyn/www/f?p=103:38:309236433809611::::FSP_ORG_ID,FSP_APEX_PAGE,FSP_PROJECT_ID:20486,23,104064

[49] ISO/IEC WD 30188 Digital Twin – Reference architecture. https://www.iec.ch/dyn/www/f?p=103:38:309236433809611::::FSP_ORG_ID,FSP_APEX_PAGE,FSP_PROJECT_ID:20486,23,104896ISO

[50] ISO/IEC 30149:2024 Internet of Things (IoT) - Trustworthiness principles. https://webstore.iec.ch/publication/67281

[51] ISO/IEC TR 40141 Guidance on reference architecture

Contributors

Editors:

Georgios Karagiannis, AIOTI WG Standardisation Chair, Huawei

Damir Filipovic, AIOTI Secretary General

Contributors:

Antonio Kung, Trialog
Antonio Lalaguna, ACISA
Arne Berre, SINTEF
Artur Krukowski (RFSAT)
Asbjorn Hovsto, Hafenstrom
Dave Raggett, GEIE ERCIM
Georgios Karagiannis, Huawei
George Suciu, BEIA
Juergen Heiles, Siemens
Kees Kroep, TU Delft
Krzysztof Piotrowski (IHP-Microelectronics)
Marco Carugi, Huawei
Martin Alvarez Espinar, Huawei
Martin Serrano, Insight Centre for Data Analytics
Omar Elloumi, Nokia
Ovidiu Vermesan, SINTEF
Rute Sofia, fortiss
R. Venkatesha Prasad, TU Delft
Said Gharout, Orange
Thomas Klein, IBM
Zbigniew Kopertowski (Orange)
Vasileios Karagiannis, Austrian Institute of Technology

Acknowledgements

About AIOTI

AIOTI is the multi-stakeholder platform for stimulating IoT and Edge Computing Innovation in Europe, bringing together small and large companies, 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 and Edge Computing 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 and Edge Computing Innovation in society. AIOTI's contribution goes beyond technology and addresses horizontal elements across application domains, such as matchmaking and stimulating cooperation in IoT and Edge Computing ecosystems, creating joint research roadmaps, driving convergence of standards and interoperability and defining policies.