ETSI TR 103 535 V1.1.1 (2019-10)

TECHNICAL REPORT

SmartM2M;
Guidelines for using semantic interoperability in the industry

Reference

DTR/SmartM2M-103535

Keywords

interoperability, IoT, oneM2M, privacy, SAREF, semantic

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Contents

List of figures

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

Foreword

This Technical Report (TR) has been produced by ETSI Technical Committee Smart Machine-to-Machine communications (SmartM2M).

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

1.1 Context for the present document

The design, development and deployment of - potentially large - IoT systems require to address a number of topics - such as privacy, interoperability or privacy - that are related and should be treated in a concerted manner. In this context, several Technical Reports have been developed that each address a specific facet of IoT systems.

In order to provide a global a coherent view of all the topics addressed, a common approach has been outlined across the Technical Reports concerned with the objective to ensure that the requirements and specificities of the IoT systems are properly addressed and that the overall results are coherent and complementary.

The present document has been built with this common approach also applied in all of the other documents listed below:

- ETSI TR 103 533 [i.1]
- ETSI TR 103 534 [i.2]
- ETSI TR 103 536 [i.3]
- ETSI TR 103 537 [i.4]
- ETSI TR 103 591 [i.5]

1.2 Scope of the present document

Major efforts are on-going in the IoT community regarding the development of semantic interoperability for IoT. This progress has been notably accomplished by the involvement from academic players. However, semantic in IoT is complex, often misunderstood and its benefits are not well perceived by the industrial players.

The main objective of the present document is to push semantic interoperability in IoT forward in raising awareness about its importance in industry in order to unlock the potential economic value of IoT. A major focus is on the development of guidelines on how to use semantic interoperability in the industry.

2 References

2.1 Normative references

Normative references are not applicable in the present document.

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] ETSI TR 103 533: "SmartM2M; Security; Standards Landscape and best practices".

[i.2] ETSI TR 103 534 (all parts): "SmartM2M; Teaching Material".

[i.3] ETSI TR 103 536: "SmartM2M; Strategic/technical approach on how to achieve interoperability/interworking of existing standardized IoT Platforms".

[i.4] ETSI TR 103 537: "SmartM2M; PlugtestsTM preparation on Semantic Interoperability".

[i.5] ETSI TR 103 591: "SmartM2M; Privacy study report; Standards Landscape and best practices".

[i.6] AIOTI Report: "High Level Architecture (HLA) Release 4.0 June 2018".

NOTE: Available at https://aioti.eu/wp-content/uploads/2018/06/AIOTI-HLA-R4.0.7.1-Final.pdf.

[i.7] IoT-EPI: "IoT Platforms Interoperability Approaches", White Paper, IoT-EPI Platform Interoperability Task Force, 2017 updated in 2018 by "Advancing IoT Platforms Interoperability Book", July 2018, White Paper.

NOTE: Available at http://iot-epi.eu/wp-content/uploads/2018/07/Advancing-IoT-Platform-Interoperability-2018-IoT-EPI.pdf.

[i.8] Inter-IoT project.

NOTE: Available at http://www.inter-iot-project.eu.

[i.9] InterIoT GOIoTP: "Generic Ontology for IoT Platforms".

NOTE: Available at http://docs.inter-iot.eu/ontology.

[i.10] VICINITY project.

NOTE: Available at https://www.vicinity2020.eu.

[i.11] VICINITY Deliverable D1.6: "VICINITY Architectural Design".

NOTE: Available at https://www.vicinity2020.eu/vicinity/content/d16-vicinity-architectural-design.

[i.12] BIG-IoT project.

NOTE: Available at http://big-iot.eu/.

[i.13] Stefan Schmid et al: "An Architecture for Interoperable IoT Ecosystems". 2nd International Workshop on Interoperability & Open Source Solutions for the Internet of Things (InterOSS-IoT 2016) at the 6th International Conference on the Internet of Things (IoT 2016), 7 November 2016, Stuttgart, Germany. Springer, LNCS.

[i.14] A.S. Thuluva et al: Recipes for IoT Applications: "The 7th International Conference on the Internet of Things (IoT 2017)", 22.-25. October 2017, Linz, Austria. ACM.

[i.15] ACTIVEAGE project.

NOTE: Available at http://www.activageproject.eu.

[i.16] ACTIVAGE Deliverable D3.2: "ACTIVAGE Interoperability layer architecture".

[i.17] MONICA project.

NOTE: Available at http://www.monica-project.eu.

[i.18] MONICA Deliverable D3.1: "IoT Enabled Devices and Wearables 2", 2018.

NOTE: Available at https://www.monica-project.eu/sdm_downloads/d3-2-iot-enabled-devices-and-wearables-2/.

[i.19] S. Meiling and al: MONICA in Hamburg: "Towards Large-Scale IoT Deployments in a Smart City".

[i.20] ETSI TS 103 264: "SmartM2M; Smart Appliances; Reference Ontology and oneM2M Mapping".

NOTE: Available at https://www.etsi.org/standards#page=1&search=TS%20103%20264.

[i.21] ETSI TS 103 410 (all parts): "SmartM2M; Smart Appliances Extension to SAREF".

NOTE: Available at https://www.etsi.org/standards#page=1&search=TS%20103%20410.

[i.22] ETSI TS 118 112: "oneM2M; Base Ontology (oneM2M TS-0012 version 2.0.0 Release 2)".

[i.23] ETSI SAREF: "SAREF ontology".

NOTE: Available at http://saref.etsi.org. The documentation of SAREF v2.1.1 will be available here soon. The source of the ontology are available as Turtle or RDF/XML Visualize it with VOWL.

[i.24] W3C SSN Editor's Draft: "Semantic Sensor Network Ontology".

NOTE: Available at http://w3c.github.io/sdw/ssn/.

[i.25] W3C Recommendation: "Semantic Sensor Network ontology".

NOTE: Available at https://www.w3.org/TR/vocab-ssn/.

[i.26] ETSI TR 118 507: "oneM2M; Study on Abstraction and Semantics Enablement (oneM2M TR-0007 Release 2)".

[i.27] Moreira, J. L., Daniele, L. M., Ferreira Pires, L., van Sinderen, M. J., Wasielewska, K., Szmeja, P., Paprzycki, M. (2017): "Towards IoT platforms' integration: Semantic Translations between W3C SSN and ETSI SAREF". Paper presented at SEMANTiCS conference 2017, Amsterdam, Netherlands.

[i.28] Mainflux project.

NOTE: Available at https://www.mainflux.com/.

[i.29] Ervin Varga, Draško Draškovic, Dejan Mijic: "Scalable Architecture for the Internet of Things", Publisher: O'Reilly Media, Inc., February 2018, ISBN: 9781492024132.

[i.30] ETSI TR 103 545: "SmartM2M; Pilot test definition and guidelines for testing cooperation between oneM2M and Ag equipment standards".

[i.31] ISO 11783: "Tractors and machinery for agriculture and forestry -- Serial control and communications data network".

[i.32] ETSI EN 302 637-3: "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service".

[i.33] IETF RFC 8428: "Sensor Measurement Lists (SenML)", C. Jennings et al. August 2018.

[i.34] Nova: Nova, 2004: "The Ontology Problem: A Definition with Commentary".

[i.35] Jaehun Joo (2011): "Adoption of Semantic Web from the perspective of technology innovation: A grounded theory approach". Int. J. Hum.-Comput. Stud. 69, 3 (March 2011), 139-154.

[i.36] ETSI TS 118 121: "oneM2M; oneM2M and AllJoyn® Interworking (oneM2M TS-0021)".

[i.37] ETSI TS 118 114: "oneM2M; LWM2M Interworking (oneM2M TS-0014)".

[i.38] ETSI TS 118 124: "oneM2M; OIC Interworking (oneM2M TS-0024)".

[i.39] ETSI TR 118 556: "oneM2M; Summary of Differences between Release 2A & Release 3 (oneM2M TR-0056)".

[i.40] ETSI TS 118 133: "Interworking Framework (oneM2M TS-0033 v0.1.1)".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

cyber security (or cybersecurity): collection of tools, policies, security concepts, security safeguards, guidelines, risk management approaches, actions, training, best practices, assurance and technologies that can be used to protect the cyber environment and organization and user's assets

domain ontology: concepts which belong to a part of the world, such as energy, building or Environment

IoT LSP: Internet of Things Large Scale Pilots which are part of the H2020 Work Program 2016-2017

oneM2M: Partnership Project (EPP) on M2M launched by a number of SSOs including ETSI

Open Source Software (OSS): computer software that is available in source code form

NOTE: The source code and certain other rights normally reserved for copyright holders are provided under an open-source license that permits users to study, change, improve and at times also to distribute the software.

source code: any collection of computer instructions written using some human-readable computer language, usually as text

standard: output from a Standards Setting Organization (SSO)

Standards Setting Organization (SSO): any entity whose primary activities are developing, coordinating, promulgating, revising, amending, reissuing, interpreting or otherwise maintaining standards that address the interests of a wide base of users outside the standards development organization

NOTE: In the present document, SSO is used equally for both Standards Setting Organization or Standards Developing Organizations (SDO).

upper ontology: also called a top-level ontology or foundation ontology, is an ontology that models very general concepts common across several domains

NOTE: An important function of an upper ontology is to support broad semantic interoperability among a large number of domain-specific ontologies by providing a common starting point for the formulation of definitions.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

AD	Automated Driving
ADN	Application Dedicated Node
AE	Application Entity
AEF	Agricultural Industry Electronics Foundation
AIOTI	Alliance for the Internet of Things Innovation
API	Application Programming Interface
ASN	Application Service Node
BLE	Bluetooth Low Energy
CBOR	Concise Binary Object Representation
CEN	European Committee for Standardization
CIM	Core Information Model

CSE	Common Service Entity
DATEX	Data Exchange format For Exchanging Traffic Information
DL	Description Logic
DMAG	Data Modelling Activity Group
DSF	Demand-side Flexibility
EC	European Commission
EPI	European Platforms Initiative
ERP	Enterprise Resource Planning
ESB	Enterprise Service Bus
ETL	Extract, Transform and Load
ETSI	European Telecommunications Standards Institute
EXI	Efficient XML Interchange
GPS	Global Positioning System
HGI	Home Gateway Initiative
ICT	Information and Communication Technology
IIoT	Industrial IoT
IoT	Internet of Things
IoT-EPI	IoT European Platform Initiative
iPaaS	Integration Platform as a Service
ISG	Industry Specification Group
ISO	International Organization for Standardization
ITS	Intelligent Transport System
JSON	JavaScript Object Notation
LSP	Large Scale Pilot
M2M	Machine-to-Machine
MN	Middle Node
MQTT	Message Queuing Telemetry Transport
NGSI	Next Generation Service Interface
OGC	Open Geospatial Consortium
OIC	Open Interconnect Consortium
OPC	Open Platform Communications
OPC-UA	OPC Unified Architecture
OSS	Open Source Software
OWL	OntologyWeb Language
P2P	Point-to-Point
PIM	Platform-Specific Information Model
RDF	Resource Description Framework
SAREF	Smart Applications REFerence ontology
SDO	Standard Development Organization
SDT	Smart Device Template
SEAS	Smart Energy Aware Systems
SIL	Semantic Interoperability Layer
SOSA	Sensor, Observation, Sampler and Actuator
SPINE	Smart Premises Interoperable Neutral-message Exchange
SSN	Semantic Sensor Network
SSO	Standards Setting Organization
TC	Technical Committee
UA	Unified Architecture
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WoT	Web of Things
XML	eXtensible Markup Language

4 Semantic interoperability in the context of IoT

4.1 A global approach to IoT Systems

4.1.1 Major characteristics of IoT systems

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 standardized 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.

4.1.2 The need for an "IoT-centric" view

4.1.2.1 Introduction

In support of an "IoT-centric" approach, some elements have been used in the present document in order to:

- support the analysis of the requirements, use cases and technology choices (in particular related to interoperability);
- ensure that the target audience can benefit from recommendations adapted to their needs.

4.1.2.2 Roles

A drawback of many current approaches to system development is an exclusive focus on the technical solutions without considering the individual in these multiple capacities (e.g. user of an IoT device, professional) which may lead to suboptimal or even ineffective systems that hinder maximizing the benefits of IoT. In the case of IoT systems, a very large variety of potential stakeholders are involved, each coming with specific - and potentially conflicting - requirements, expectations and, possibly, vested interests. 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 characterized and analysed are System Designer, System Developer, System Deployer, End-user, Device Manufacturer. Certain these roles are to an extent addressed in the present document.

4.1.2.3 Reference Architecture(s)

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 standardization. Moreover, 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.

4.1.2.4 Guidelines

The very large span of requirements, Use Cases and roles within an IoT system make it difficult to provide prototypical solutions applicable to all of the various issues addressed. The approach taken in the present document is to outline some solutions but also to provide guidelines on how they can be used depending on the target audience. Such guidelines are associated to the relevant roles and provide support for the decision-making. The AIOTI High-Level Architecture (see [i.6]) will be referred to in the present document.

4.2 Purpose and target group

The present document addresses the topic of semantic interoperability in the context of its potential usage by the industry in the development of IoT systems. The main objective of the present document is to concretely foster the use of semantic interoperability in IoT by identify why it is important in industry IoT projects, to analyse the advantages and drawback of the available solutions and to provide guidelines on how to use semantic interoperability in the industry in order to unlock the overall economic value of IoT.

The target group for the present document is the community of people that design, develop, implement and validate IoT systems, that have to understand the benefits of semantic interoperability, to decide on the modality and extent of its usage and to characterize and prepare the necessary actions (e.g. training) with respect to those who will use it in the development of IoT systems.

4.3 Content of the present document

Clause 5 is making an in-depth analysis of the state-of-the-art of semantic interoperability. It defines the different approaches that are used, in particular the ontologies. The solutions from academics, standards and industry are analysed and compared.

Clause 6 is giving different aspects of the adoption of semantic interoperability with the case of the industry as a specific case. To this extent, after analysing the approaches currently adopted in the industry to deal with interoperability, it addresses the drivers and inhibitors to market adoption. The case of ontologies is analysed in detail in order to understand what the blocking factors are and how they can be overcome.

Clause 7 is providing concrete and actionable guidelines towards those in charge of making decisions regarding the use of semantic interoperability solutions and of implementing those decisions within the overall IoT systems technical and cultural development environment.

5 State of the art of semantic interoperability

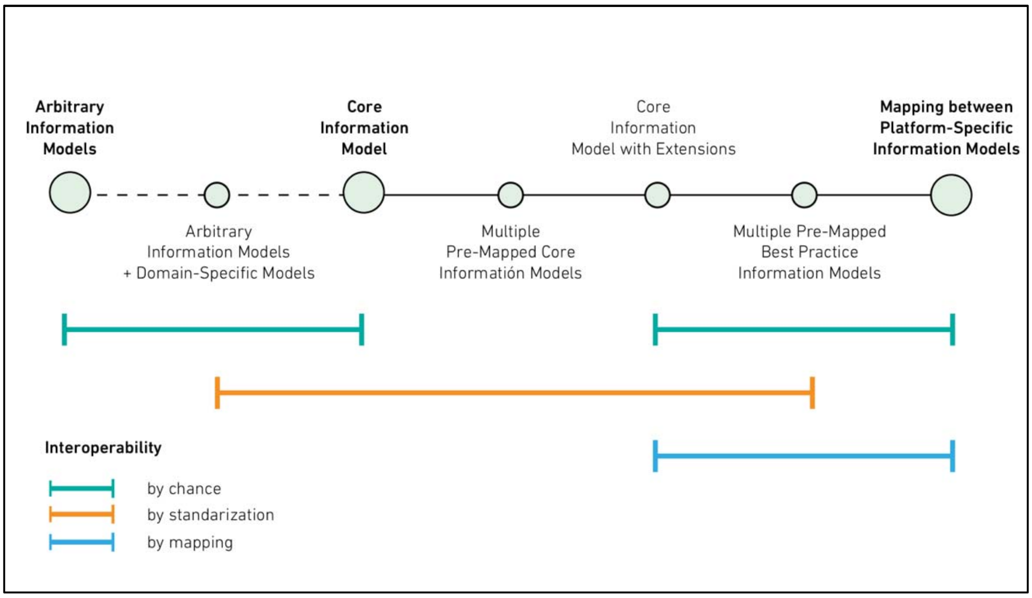
5.1 Semantic interoperability: Approaches and classification systems

5.1.1 Semantic approaches

The main expectation of semantic interoperability is to provide a shared unambiguous meaning of what the "things" that two (or more) platforms may agree upon, thus bridging the potential semantic gap coming from different descriptions and implementations of the "thing" under concern. The challenge of semantic interoperability is in general a cross-platform issue, though it can be also met with two components on the same platform.

The IoT European Platforms Initiative (IoT-EPI) has addressed this issue (see [i.7]) in a global manner with a model that is depicted in Figure 1. There are two dimensions in their analysis:

- the main approaches related to the technical solution that can range from a single Core Information Model (CIM) that every platform need to comply to (irrespective of the domain or sector) up to the possibility to define the models that a platform considers as appropriate, while ensuring that these models can be aligned by using a semantic mapping that can be shared across platforms;
- the type of interoperability that can be expected: "by chance" (where a platform will interoperate with another one only if their models happen to be the same), "by standardization" (where platforms agree on whole or part of a common standardized model) or "by mapping" (where some translation "logic" is applied between different models).



NOTE: Source: IoT-EPI Task Force [i.7].

Figure 1: Possible approaches to semantic interoperability

The preparation and undertaking of semantic interoperability PlugTests™ will address the validation of interoperability "by standardization" or "by mapping" and will focus on the approaches ranging from Core Information Model (CIM) to Multiple Pre-Mapped Best Practice Information Models (as described in Figure 1).

More information and examples of these approaches can be found in the companion ETSI TR 103 537 [i.4].

5.1.2 Classification systems

A classification system is a grouping of something based on some criteria. There are many different types of classification systems:

- **Glossary** is a terminological dictionary which contains a list of designations from a subject field, together with equivalents in one or more languages.
- **Dictionary** is an alphabetical list of terms in a particular domain of knowledge with the definitions for those terms.
- **Taxonomy** is a simple hierarchical arrangement of entities where a parent-child kind of relationship is defined.
- **Thesaurus** is a reference work that lists words grouped together according to similarity of meaning containing synonyms and sometimes antonyms. Unlike a dictionary, a thesaurus entry does not give the definition of words.
- **Topic map** is a standard for the representation and interchange of knowledge, with an emphasis on the findability of information.
- **Meta data repository** is a database created to store metadata. Metadata is information about the structures that contain the actual data. Metadata may describe the structure of any data, of any subject, stored in any format.
- **Microformat** is a web approach to semantic markup which uses tags supported for other purposes to convey additional metadata and other attributes in different contexts.
- **Ontology** is the specification of conceptualizations used to help programs and humans share knowledge. This includes definitions and indications of how concepts are inter-related which collectively impose a structure on the domain and constrain the possible interpretations of terms.

5.1.3 Ontologies components and types

The main components of an ontology are concepts, relations, instances and axioms:

- **Concepts** represent a set or class of entities or things within a domain including both primitive and defined concepts.
- **Relations** describe the interactions between concepts or a concept's properties including taxonomies and associative relationships. Both concepts and relations could be organized into taxonomy.
- **Instances** are the things represented by a concept.
- **Axioms** are used to constrain values for classes or instances.

Two types of ontologies exist:

- **Upper ontology** is a model of the common relations and objects that are generally applicable across a wide range of domain ontologies. It usually employs a core glossary that contains the terms and associated object descriptions as they are used in various relevant domain ontologies.
- **Domain ontology** represents concepts which belong to a part of the world, such as building, energy or environment. Each domain ontology typically models domain specific definitions of terms. Since domain ontologies are written by different people, they represent concepts in very specific and unique ways and are often incompatible within the same project.

5.2 Existing solutions from academia, standards and industry

5.2.1 H2020 IoT European Platform Initiative (IoT-EPI)

5.2.1.1 Introduction

The IoT European Platforms Initiative (IoT-EPI) projects are addressing, in particular, the question of IoT and the platforms to connect Smart Objects. Their global aim was to extend IoT into a web of platforms for connected devices and objects that supports smart environments, businesses, services and persons with dynamic and adaptive configuration capabilities.

The IoT-EPI projects are developing various interoperability solutions that address different layers in the IoT architecture; and offer mechanisms for providing interoperability between different IoT platforms. The IoT-EPI Task Force on Platforms Interoperability has published a White Paper entitled "IoT Platforms Interoperability Approaches - White Paper 2018" that was published in a book [i.7] where more details on these projects can be found.

5.2.1.2 The SymbIoTe project

SymbIoTe is an H2020 Research and Innovation Action and member of the IoT-European Platforms Initiative cluster that addresses a challenging objective of creating an interoperable IoT ecosystem. SymbIoTe facilitate the cooperation of vertical IoT platforms, which are today typically offered as closed systems, to simplify the development of cross-domain and cross-platform IoT applications. The main goal of SymbIoTe is to design a secure and flexible system interoperability middleware across IoT platforms to facilitate platform collaboration, rapid development of IoT applications and dynamic and adaptive intelligent objects and environments.

SymbIoTe offers a unified view on different platforms to a new generation of cross-platform IoT applications. It enables the discovery of IoT devices across platforms and relies on a common semantic representation of IoT resources (services or devices). A resource is a uniquely addressable entity in symbIoTe architecture and, as a generic term, may refer to IoT devices, virtual entities, network equipment, computational resources and associated server-side functions (e.g. data stream processing). SymbIoTe does not require the platforms to adapt to a static predefined model but gives platform owners freedom to choose the model used for describing their resources and data. Each platform has to provide a Platform-Specific Information Model (PIM) used to describe its advertised resources. To ensure basic interoperability between platforms, each PIM has to be an extension of the Core Information Model (CIM).

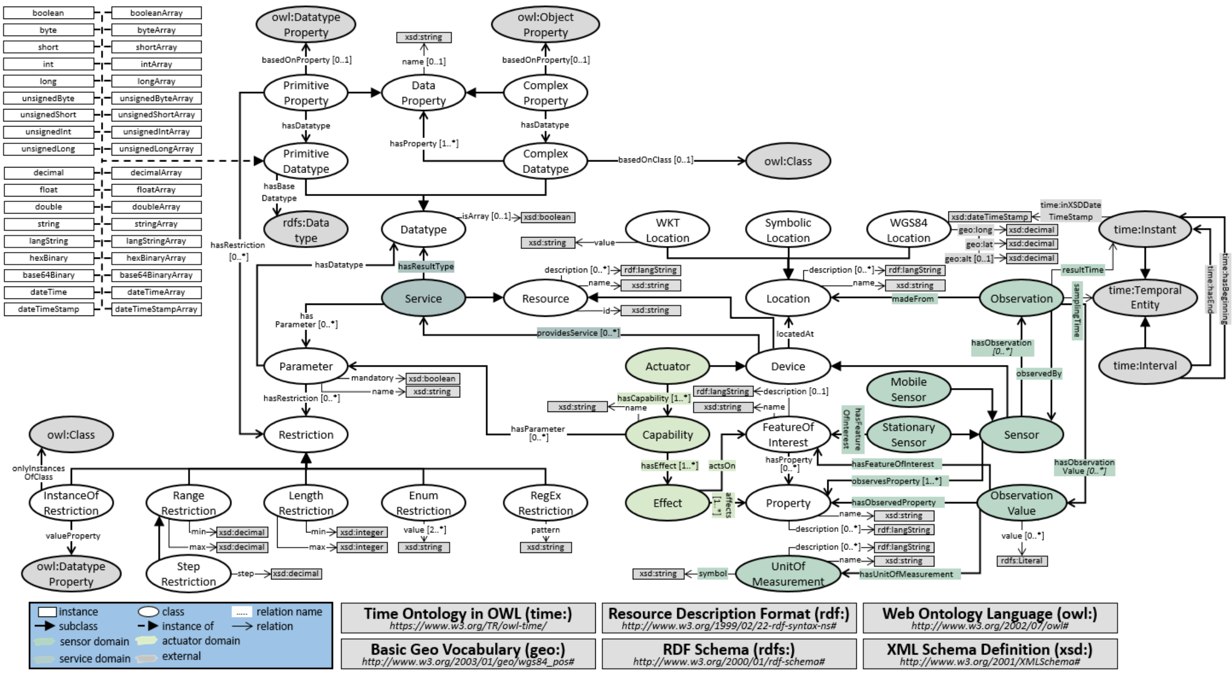


Figure 2: SymbIoTe data model

Figure 2 describes SymbIoTe Core Information model. The main class is "Resource". "Service", "Actuator" and "Sensor" are subclasses of "Resource" class and represent resources types that can be registered with SymbIoTe Core.

5.1.1.3 The Agile IoT project

The AGILE project aims to address technical and syntactic interoperability at hardware and software levels. On the hardware front, AGILE designs hardware components extending the current state-of-the-art of available IoT gateway platforms with a twofold objective: to develop a so called "Maker's Gateway" by extending the capabilities of the most adopted and low-cost Raspberry Pi platform; and to develop a modular hardware gateway design for industrial purposes. On the software front, the objectives of the AGILE project are to release open source code through the Eclipse Foundation to the community of IoT software developers/makers, helping them to easily configure their devices or gateways according to the platform environment.

Agile IoT provides a gateway data model which is composed by the following elements:

- **Device:** The Device API abstracts operations over the real device. It sends commands to a Protocol API instance to which the device is binded.
- **RecordObject:** Contains the measurement from a sensor and the metadata regarding its source.
- **DeviceComponent:** A device component from which get a data stream, like a sensor.
- **DeviceOverview:** An overview of a device a protocol knows about.
- **DeviceManager:** The Device Manager list the registered devices available from the gateway available protocols. This API is responsible to instantiate the reference DBus object for every device and offer management and integration points.
- **Protocol:** A Protocol object that abstract operations over a specific protocol implementation. Every Device has one Protocol used to establish and use a communication exchange.
- **ProtocolManager:** The Protocol Manager list the managed protocols in the gateway. From this interface is possible to add and remove protocols and start/stop multiprotocol discovery.
- **ProtocolProfile:** A Protocol set of settings that translate human readable properties to actual Protocol implementation details. For example, the user requests the temperature value over a BLE connected device and the Protocol receive all the specific information (like services and characteristics ids) to enable the data reading.
- **ProtocolStatus:** Identify a status of the Protocol, indicating if operational (AVAILABLE) or if for some reason further user-side checks are required.

Figure 3 shows a partial view of Agile gateway data model.

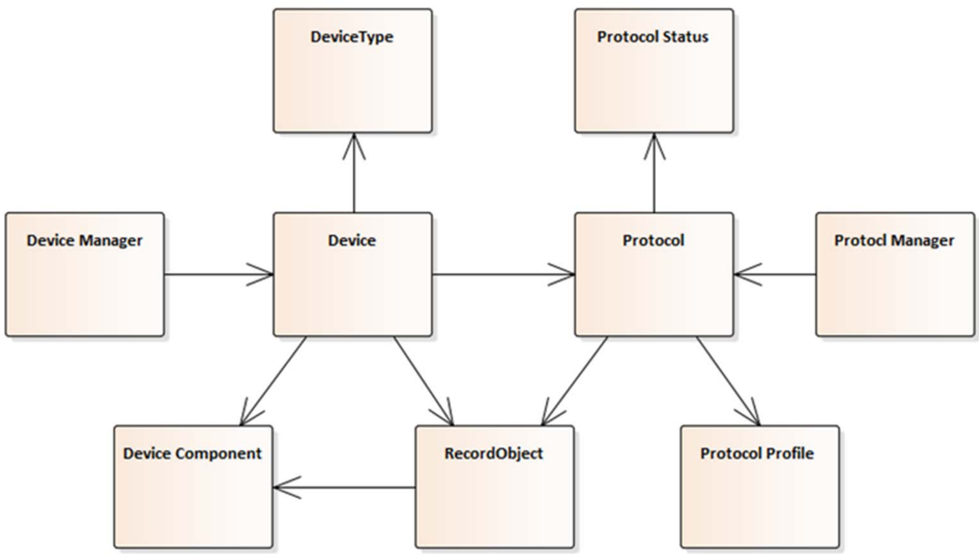


Figure 3: Partial view of Agile IoT gateway data model

5.2.1.4 The Inter IoT project

The objective of the H2020 INTER-IoT project (ICT30 project, 2016-2019) [i.8] is to provide interoperability of heterogeneous IoT platforms. The solution proposed is a layer-oriented architecture, that enables interconnection at device, network, middleware, application and data & semantics levels. It is integrated in a global framework that also provides a methodology, associated APIs and tools to enable the development of IoT applications.

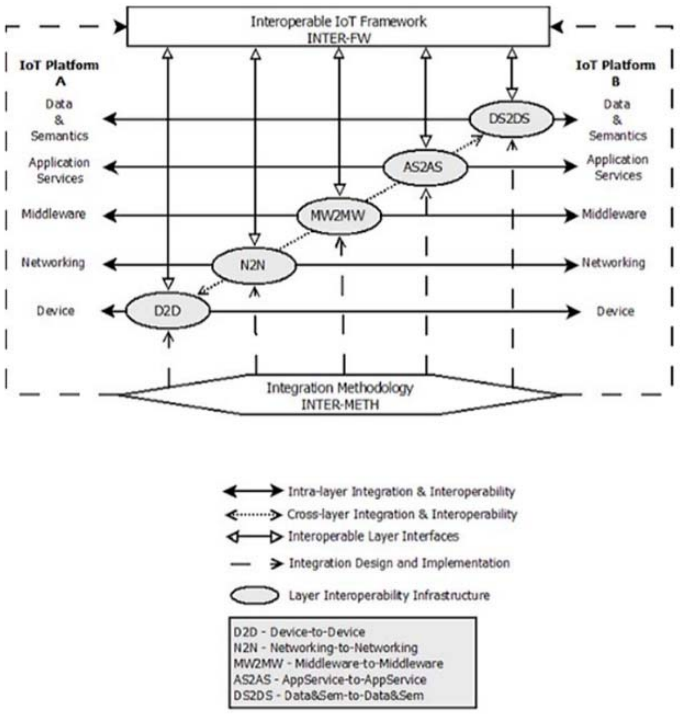


Figure 4: INTER-IOT interoperability global approach

Regarding the inter-connection at Semantics & data layer (DS2DS), the approach is to develop the common interpretation of data and information based on a global shared ontology in order to achieve universal semantic and syntactic interoperability between heterogeneous data sources. INTER-IoT has developed a generic Central Ontology, called GOIoTP [i.9], based on the W3C SSN, addressing any IoT platform, device and service. It targets a modular data structure for the description of entities most commonly appearing in IoT in the context of interoperating various different IoT entities (platforms, devices, services, etc.) as well as a reference meta-data model.

The core ontology, GOIoTP is extended with a multi-module ontology GoOIoTPex for concrete entities. Figure 5 shows the platform module ontology which is an extension of the SSN sosa:Platform extension. It also shows how the GOIoTPex extension defines a collection of subclasses of iiot:Middleware for supporting different platforms such as OM2M, FIWARE or UNIVERSAAL.

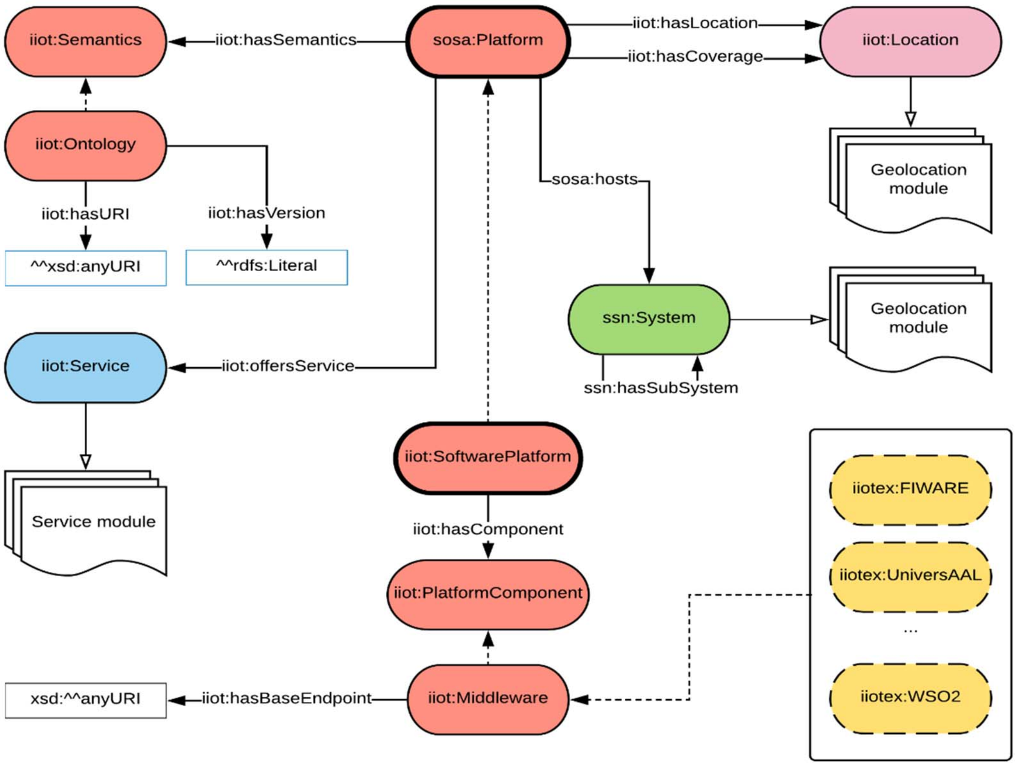


Figure 5: GOIoTP platform module

Figure 6 shows the GOIoTP device module centered around 4 classes -- iiot:IoTDevice, sosa:Sensor, sosa:Actuator and sosa:Sampler -- and how it relates to the platform module.

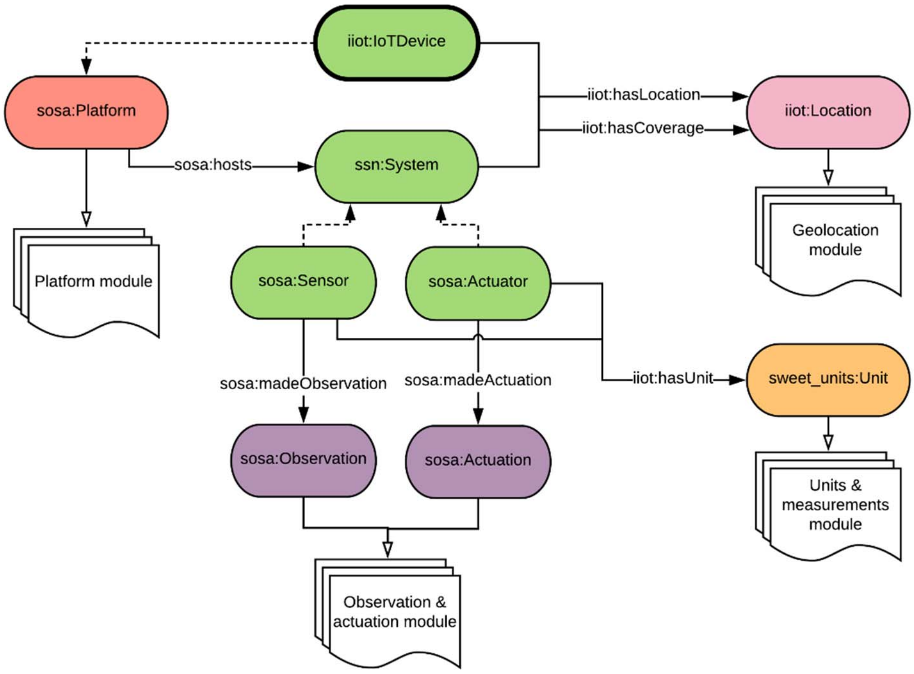


Figure 6: GOIoTP device module

The main objective of this ontology, even though it is yet another one, is to allow IoT platforms using different data formats and ontologies to share information by translating their own semantics into the central ontology, thus reducing the number of combinations needed for universal semantic interoperability, especially when addressing cross-domain ontologies and cross-platform interoperability. This solution is under test in two application domains: transportation and logistics applications and smart health/wearables applications. A small project has also created a bridge to the OM2M platform.

The approach of having a reference ontology that can be used to translate the different ontologies and prevent one-to-one semantic interoperability combinations is positive. However, Inter-IoT has applied this concept by creating yet another core ontology which is not standardized, even if it is built as a derivation and extension of the SSN.

5.2.1.5 The Vicinity project

The objective of the H2020 VICINITY project (ICT30 project, 2016-2019) [i.10] is to provide "interoperability as a service" for heterogeneous IoT objects and entities from different ecosystems. The objects interact with the VICINITY platform which enables the development of value-added services and the sharing of cross-domain information.

From a standardization standpoint, VICINITY aims to support IoT interoperability by employing a generic IoT based on widely accepted standards as well as data using proprietary formats. The approach is to provide a standard way to both Discover and Access the IoT objects. It is based on the work of the W3C Web of Things (WoT) groups and leverages Semantic Web technologies to describe, expose and consume web things.

Figure 7 shows the High-level logical VICINITY architecture [i.11]. Among other functions, the VICINITY Cloud provides a set of services to setup peer-to-peer interoperability between IoT environments (named as virtual neighbourhoods), including services for device semantic discovery and registration. The VICINITY Node includes a Gateway API which provides a semantic interoperability interface for VICINITY Adapters/Agents to register IoT objects, access shared IoT objects or discover and query IoT objects.

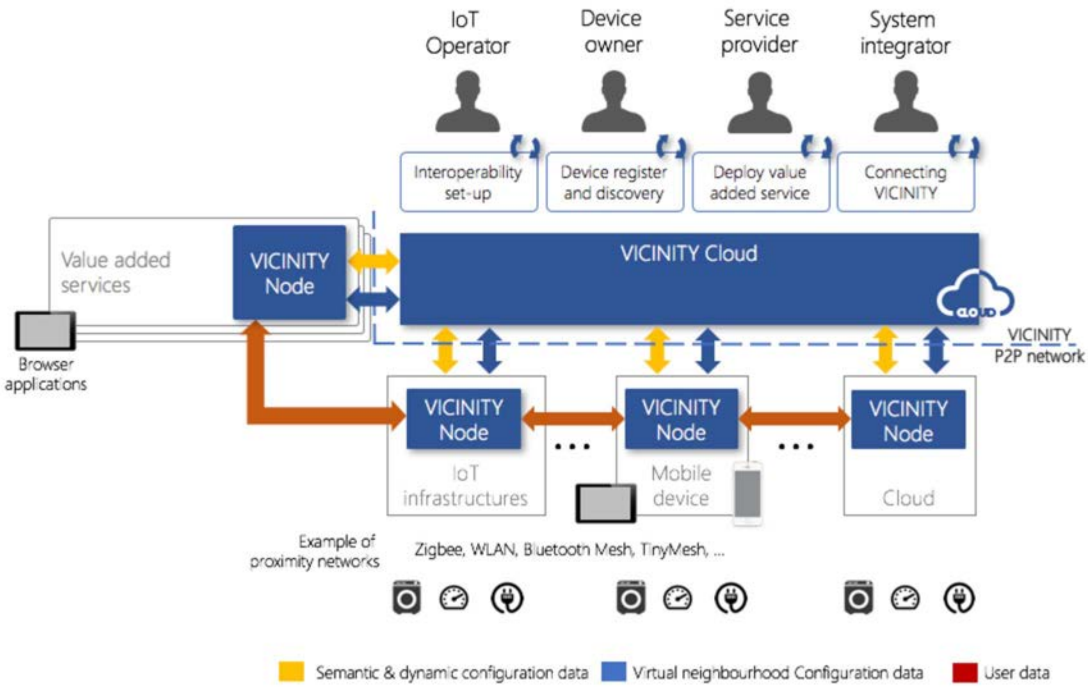


Figure 7: High-level logical VICINITY architecture

The VICINITY ontology is shown in Figure 8. It lies at the core of interoperability approach for Discovery of VICINITY nodes in the cloud and access of data in VICINITY nodes. As can be seen in Figure 8, this ontology fully fits in the standardization landscape. It has been adopted by the W3C WoT Working Group (shown in blue), it reuses parts of the SAREF4bldg (shown in white) and of W3C time, SSN and org ontologies.

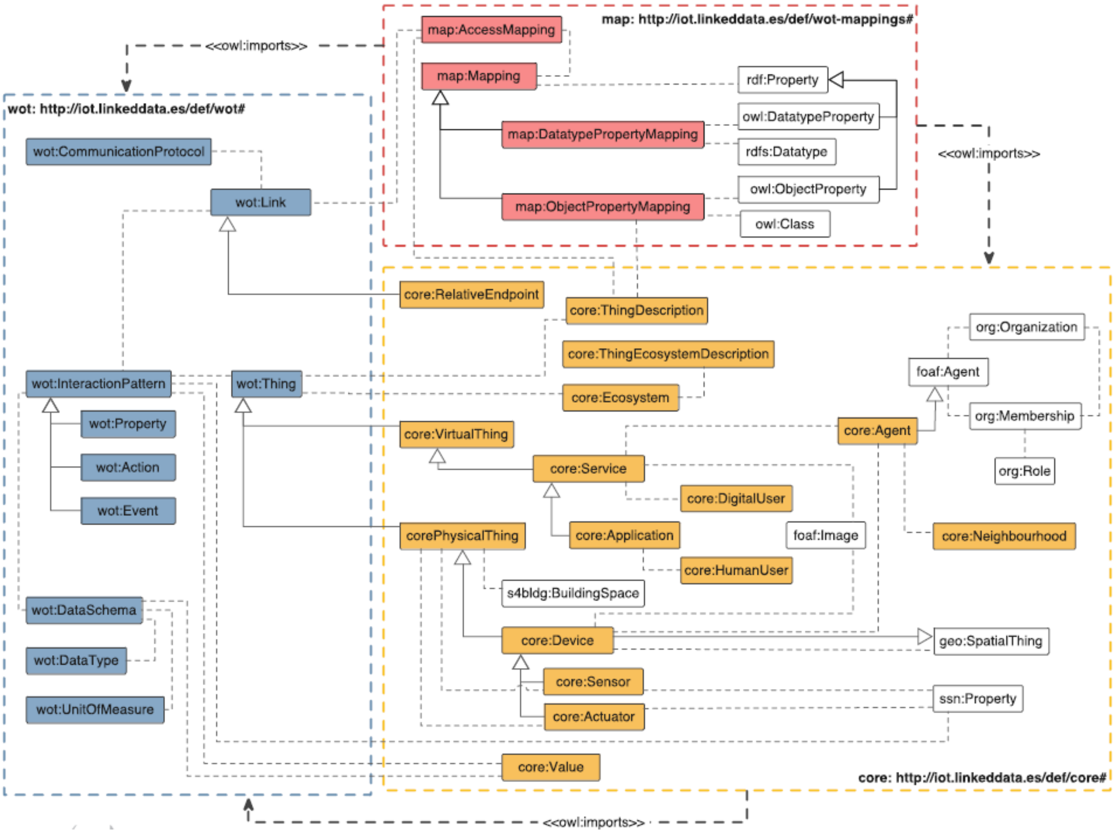


Figure 8: VICINITY ontology network

VICINITY plans to test and demonstrate its solution capabilities in use cases implemented across four pilot sites and address the following domains: Energy, Smart building, Healthcare and Mobility.

5.2.1.6 The BIG-IoT project

The objective of the H2020 BIG-IoT project (ICT30 project, 2016-2018) [i.12] is to demonstrate interoperability in different Smart Cities. It focuses on providing open source tools to enable IoT ecosystems interoperability and cross-standard, cross-platform and cross-domain IoT services and applications.

Figure 9 shows the BIG-IoT Conceptual-model-for-its-IoT-ecosystem [i.13]. Its Web API links providers and consumers offerings. A provider registers its offerings on the marketplace by providing an offering description for each offering. To increase interoperability between different IoT platforms, the offering description should be provided in a machine interpretable manner, e.g. based on RDF models. All relevant communication metadata are provided on how the offering can be accessed (e.g. endpoint URL, which HTTP method, etc.). The description may also include information about the region (e.g. the city or spatial extent) where the resources relate to, the price for accessing the resources, the license of the data provided, the access control list, etc. Consumers discover offerings of interest on the marketplace by providing an *(offering) query*. The marketplace identifies all matching offerings and returns them to the consumer. The consumer can then choose the offerings of its interest.

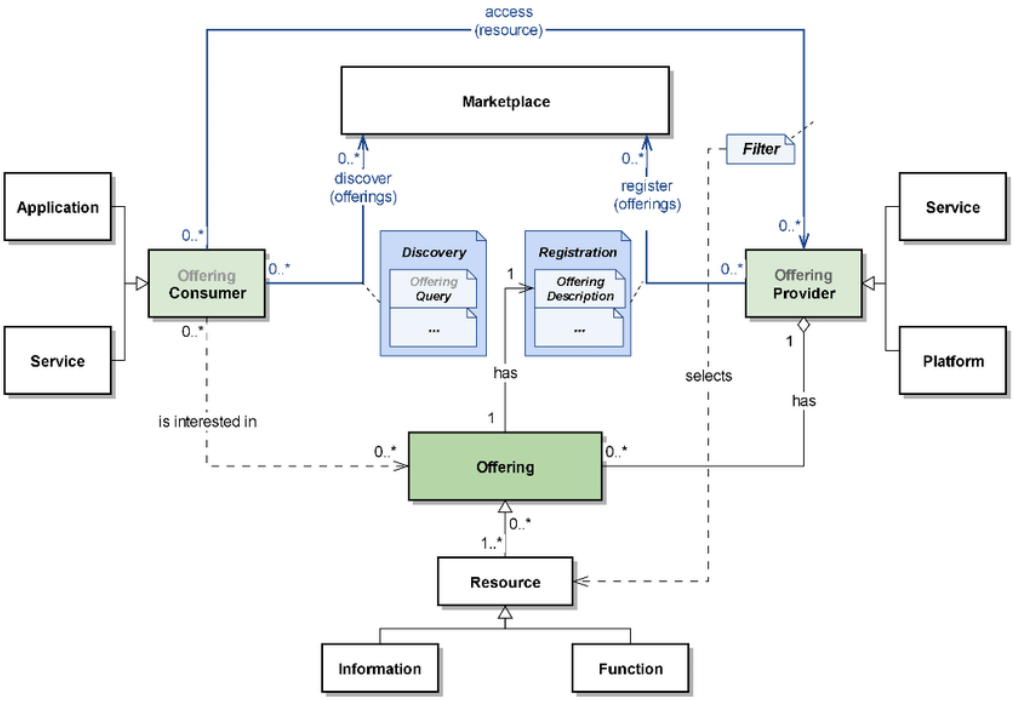


Figure 9: Conceptual-BIG-IoT model-for-an-IoT-ecosystem

Figure 10 and Figure 11 show the related ontology developed by the project. It relates to its unified Web API for IoT platforms, named the BIG IoT API, aligned with the standards developed by the W3C Web of Things group and the schema.org ontology. These models enable a composer to not only understand the data formats and interfaces of the constituent offerings, but also to aggregate offerings using semantic-based service composition rules. This approach is expected to provide a generic way of describing custom compositions of IoT offerings [i.14].

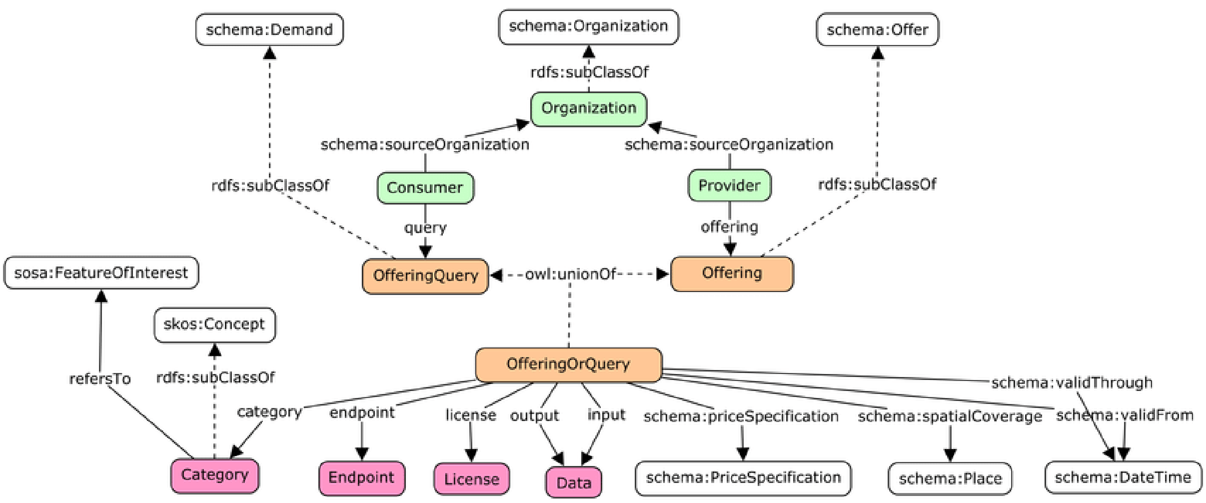


Figure 10: Model for describing offerings of IoT platforms, things or services

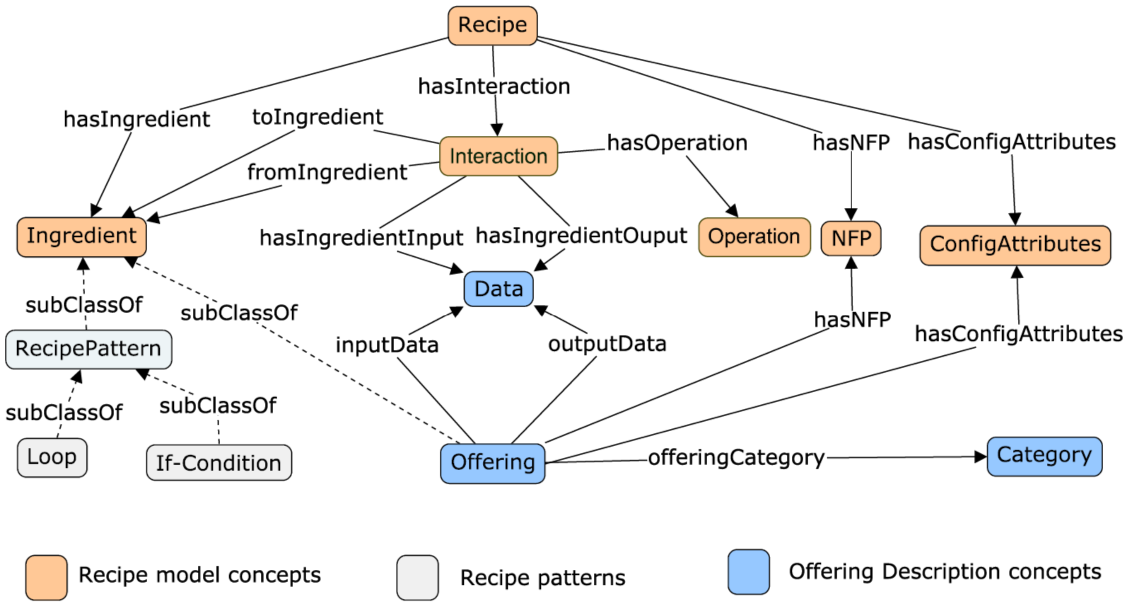


Figure 11: Model for describing IoT offering compositions

The project planned to implement and demonstrate its services and applications in different cities: Barcelona, Piedmont and Berlin/Wolfsburg.

5.2.2 H2020 Large Scale Pilots (LSP)

5.2.2.1 Introduction

The IoT European Large-Scale Pilots (LSP)Programme includes the innovation consortia that are collaborating to foster the deployment of Internet of Things (IoT) solutions in Europe through integration of advanced IoT technologies across the value chain, demonstration of multiple IoT applications at scale and in a usage context and as close as possible to operational conditions. The LSPs target Reference Zones in Cities, Smart Living Environments for Aging Well, Wearables for Smart Ecosystems, Smart Farming and Food Security and Autonomous Vehicles in a Connected Environment.

5.2.2.2 The Autopilot Project

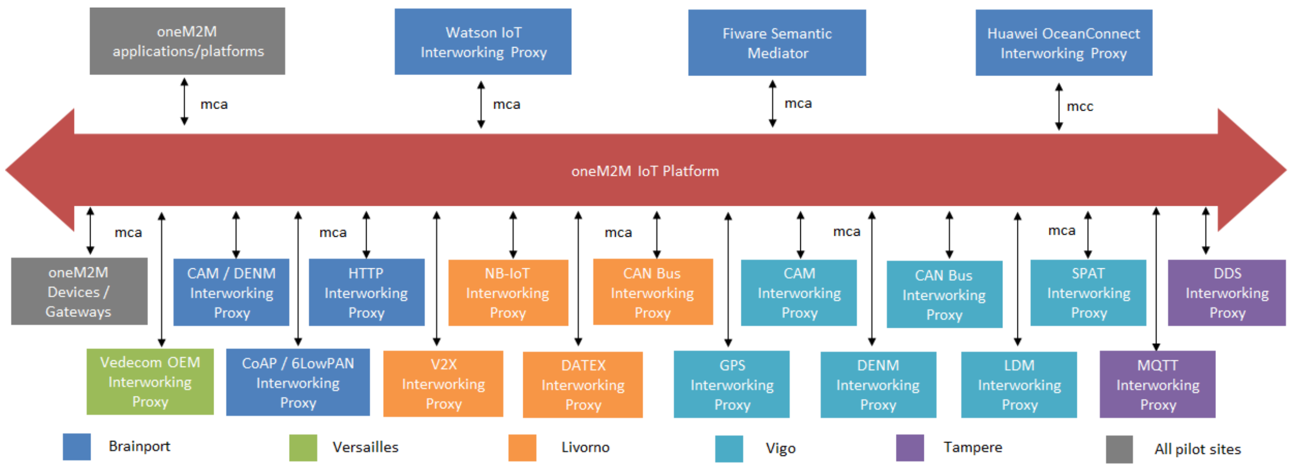


Figure 12: AUTOPILOT interworking components

In AUTOPILOT, the contents of the oneM2M messages exchanged between the IoT platforms, devices, applications and vehicles, through the oneM2M interoperability platform are standardized. A Data Modelling Activity Group (DMAG) was created in AUTOPILOT for this purpose. The scope of the data model standardization activity in AUTOPILOT covers the IoT messages and data fields required to implement the project's use cases uniformly across the pilot sites. This will allow AD vehicles to access the same types of data regardless of their locations (pilot sites) and to be able to process the data and work with it. Work of the DMAG is based on reusing and possibly extending, existing standards rather than creating new models. Figure 13 illustrates an overview of Autopilot data model packages and their dependencies.

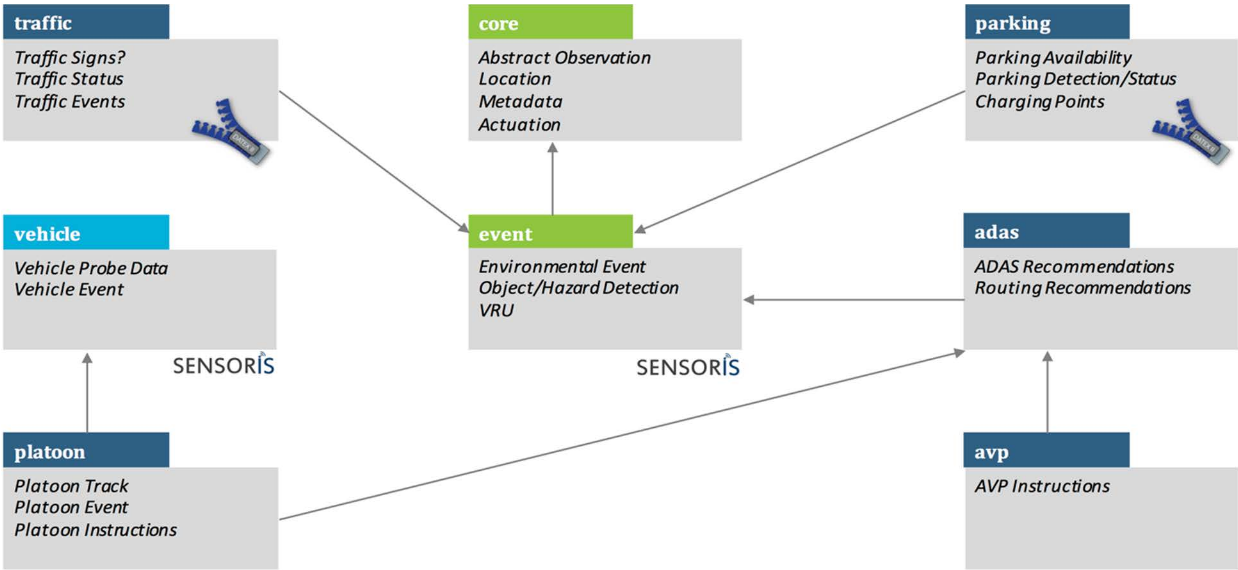


Figure 13: AUTOPILOT data model

The vehicle package covers the messages sent from the AD vehicles to the IoT platform. The vehicle package is based on the SENSORIS data model. In addition to vehicle GPS data and statuses, SENSORIS messages include events detected by vehicles along their paths, e.g. object detection, environment status, lane boundary detection, etc.

The parking spot service allows end users to access information about parking spot locations and availability. Information about parking spots include both static and dynamic data. DATEX II to represent the parking package, since DATEX II is an official multi-part standard, maintained by CEN Technical Committee 278, CEN/TC278 (Road Transport and Traffic Telematics).

5.2.2.3 The ACTIVAGE LSP

The objective of the H2020 ACTIVAGE project (LSP project, 2017-2020) [i.15] is to build the first European IoT ecosystem that will enable the deployment and operation at large scale of Active & Healthy Ageing IoT based solutions and services. The proposed solutions reuse and scale up underlying open and proprietary IoT platforms, technologies and standards and integrate new interfaces needed to provide interoperability across heterogeneous platforms.

The project solution is based on the ACTIVAGE IoT Ecosystem Suite (AIOTES), a set of techniques, tools and methodologies for interoperability at different layers between heterogeneous existing IoT Platforms and an Open Framework for providing Semantic Interoperability of IoT Platforms, addressing trustworthiness, privacy, data protection and security.

The ACTIVAGE architecture [i.16], illustrated in Figure 14, is in compliance with the non-standardized IoT-A reference model and has been designed to provide semantic interoperability, which enables and orchestrates the interconnection of heterogeneous IoT devices, open IoT platforms and smart living services within a common ecosystem of solutions.

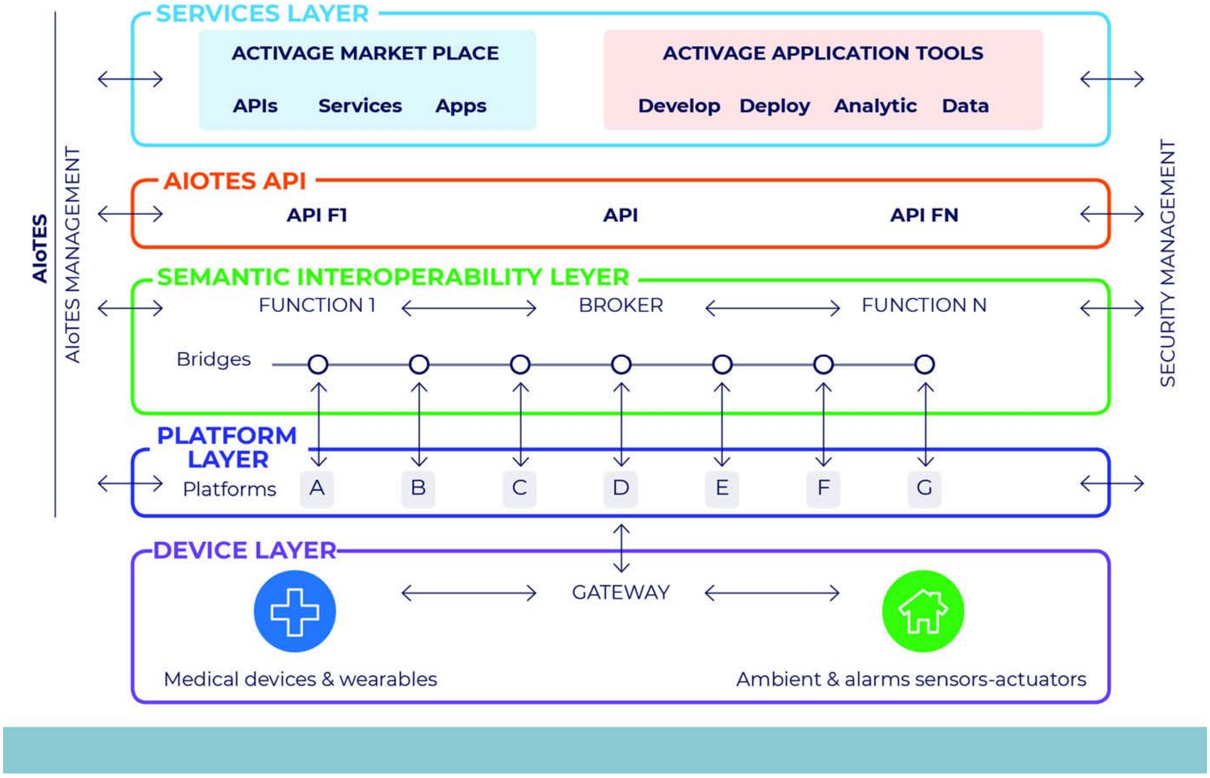


Figure 14: Overall ACTIVAGE architecture

The interoperability approach is based on the Semantic Interoperability Layer (SIL) as a Shared Space and the SIL API to develop and deploy services.

The SIL is based on standards satisfying several basic interoperability requirements from data representation (RDF), data models (OWL), data access (SPARQL) and standard data serialization for the exchange of data between distributed and possibly heterogeneous nodes. RDF and OWL play a central role in the SIL definition. The RDF model allows to represent arbitrary kinds of information related to entities in the shared space. OWL helps the SIL to remain independent from the underlying data models addressing the specific features contained in the data represented in RDF. Furthermore, the SIL ensures that the supported IoT interoperability is not limited to (dynamic) sharing of data in runtime but enables as well the (dynamic) sharing of functionality while eliminating the need for domain-/application-/device specific APIs. The SIL enables publish/subscribe as well as request/response brokering. It reuses components developed by the INTER-IoT, BIG-IOT and VICINITY projects. SIL aims at creating a semantic space, in which everything is represented based on shared understandings of the sharing parties from their areas of interest, while outside the shared space, the sharing parties may continue to rely on diverting technological specifics of their own.

5.2.2.4 The Monica LSP

The objective of the H2020 MONICA project (LSP project, 2017-2020) [i.17] is to provide for large scale demonstration of an IoT ecosystem based on multiple existing and new IoT technologies for Smarter Living. This eco-system is expected to enable cities to use IoT technologies to meet sound, noise and security challenges at big, open-air cultural and sport events, which attract and affect many people. The concept of the data handling is illustrated in Figure 15.

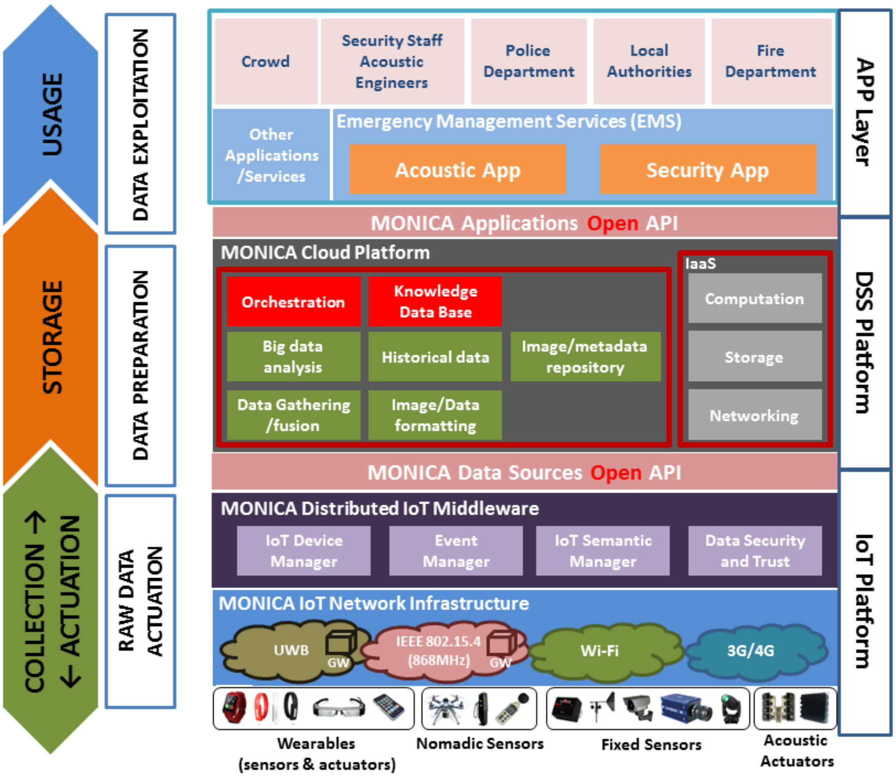


Figure 15: The MONICA Concept

In order to manage heterogeneous devices and inputs in MONICA, an adaptation layer provides access to IoT devices and streams, transforming data to comply with OGC SensorThings Model and API as well as adding additional metadata according to the standards used in the project.

Whenever possible, the MONICA IoT platform is based on open, ETSI standards and 3GPP global specifications to ensure interoperability between devices provided by different manufacturers and applications developed by different software factories. Central to MONICA is also the global OneM2M specifications which enable development of harmonised IoT standards, allowing the operator to connect devices, applications and services regardless of the baseband technologies used. The MONICA Distributed IoT Middleware applies standards such as MQTT for publish/subscribe. SAREF and the W3C SSN Ontology for semantic modelling are re-used in the service layer and enable interoperability with external platforms. The alignment of the content of the data streams with the IoT Resource ontology with regards to identification and metadata is currently in progress [i.19].

Figure 16 shows the global architecture of MONICA.

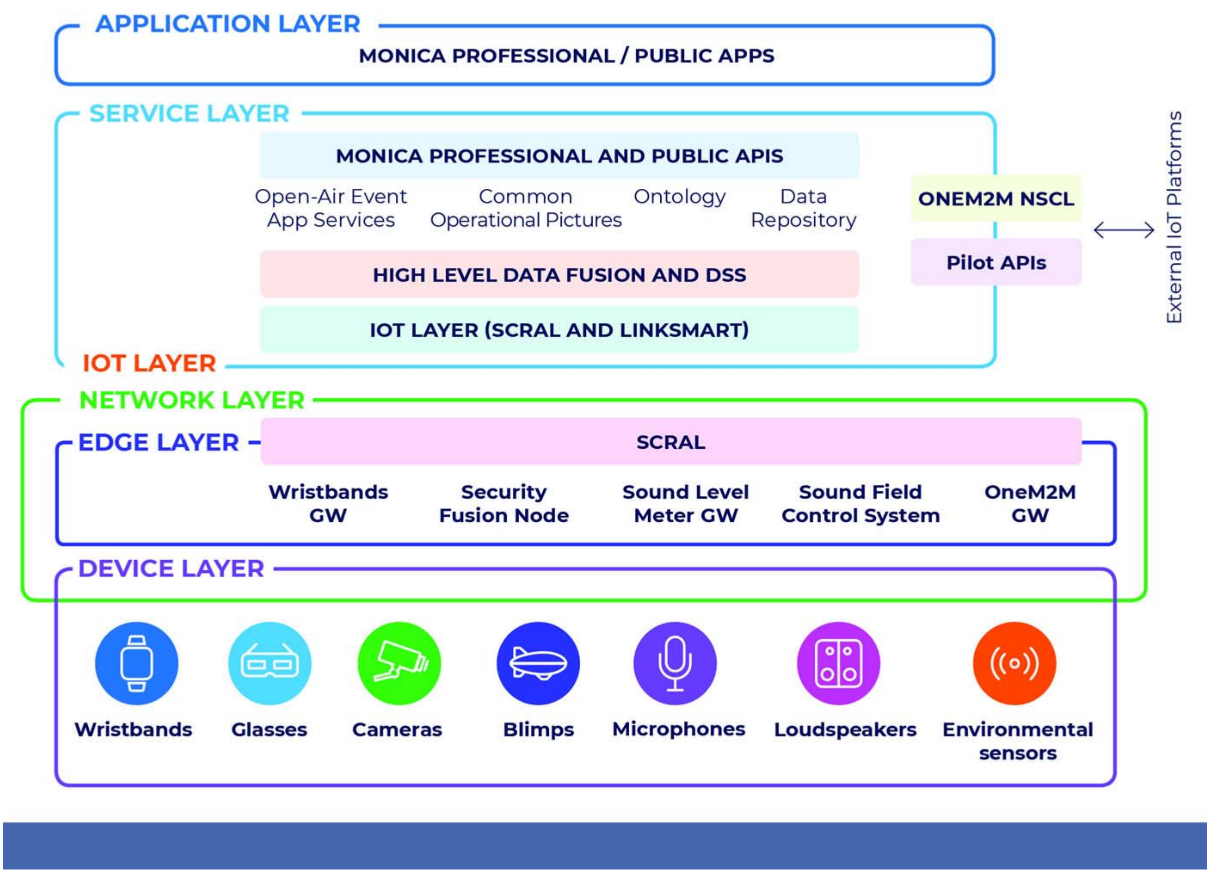


Figure 16: MONICA global architecture

Demonstrations are planned in six pilot European cities [i.18] and target thirteen events addressing the following applications: Sound monitoring and control, Crowd and capacity monitoring, Security -Health incidents and Missing persons/Locate staff members.

5.2.3 Standards

5.2.3.1 Introduction

Several standards addressed the need for semantic interoperability to unlock the real potential of IoT such as oneM2M, ETSI NGSI-LD, ETSI SAREF, OPC UA and W3C SSN.

5.2.3.2 oneM2M

The oneM2M global initiative is an international partnership project established in June 2009 by the seven most important SDOs in the world and various alliances and industries. The main goal is to define a globally agreed M2M service platform by consolidating currently isolated M2M service layer standards activities.

The oneM2M system architecture is composed of the following four functional entities: the Application Dedicated Node (ADN); the Application Service Node (ASN); the Middle Node (MN); and the Infrastructure Node (IN). Each node contains a Common Services Entity (CSE), an Application Entity (AE) or both. An AE provides application logic, such as remote power monitoring, for end-to-end M2M solutions. A CSE comprises a set of service functions called Common Services Functions (CSFs) that can be used by applications and other CSEs. CSFs include registration, security, application, service, data and device management, etc.

The oneM2M standard adopted a RESTful architecture, thus all services are represented as resources to provide the defined functions.

The oneM2M standard supports different approaches for semantic interoperability requiring a before agreement between applications and devices to share data between them. The main approaches are:

1) Pure ontology-based solution (RDF/OWL serialization format): oneM2M base ontology extended with a domain-specific ontology e.g. SAREF. For more details, see ETSI TS 118 112 [i.22].

2) Common vocabulary (basic serialization format XML or JSON): Smart Device Template (SDT) for the home domain.

3) Resources specializations: oneM2M FlexContainer resources specialized with a technology-specific data model. For more details, see ETSI TS 118 121 [i.36].

4) Blackbox resources: Basic oneM2M resources (Container, ContentInstance and Group) extended with an external domain-specific data model. The contentInstances resources are considered as black boxes and could contain any domain-specific data model. For more details, see ETSI TS 118 114 "LWM2M Interworking" [i.37] and ETSI TS 118 124 "OIC Interworking" [i.38].

A work item called ETSI TR 118 556 "Evolution of Proximal IoT Interworking" [i.39] has been defined to provide a harmonization of the work done for interworking between oneM2M and specific proximal IoT technologies, such as AllJoyn, LWM2M and OIC. The idea is enabling interworking with external "proximal" IoT technologies without the need for a oneM2M applications to be aware of the details of device specific technology. For more details, see ETSI TS 118 133 "Interworking Framework" [i.40].

5.2.3.3 Smart Device Template (SDT)

The SDT (Smart Device Template) is an initiative from HGI to find consensus amongst various SDOs and industry alliances to derive a common approach for device modelling.

The Smart Device Template (SDT) is a template which is used to model the capabilities, actions and events of connected devices. The intent of the SDT is to be able to model any type of connected device using a well-accepted and standardized format. The main application of SDT is to enable a uniformly structured APIs to applications that need to interact with connected devices using an abstraction layer as an intermediary logic. SDT data model hides the technology-specific, native language format of devices of different technology type from the applications.

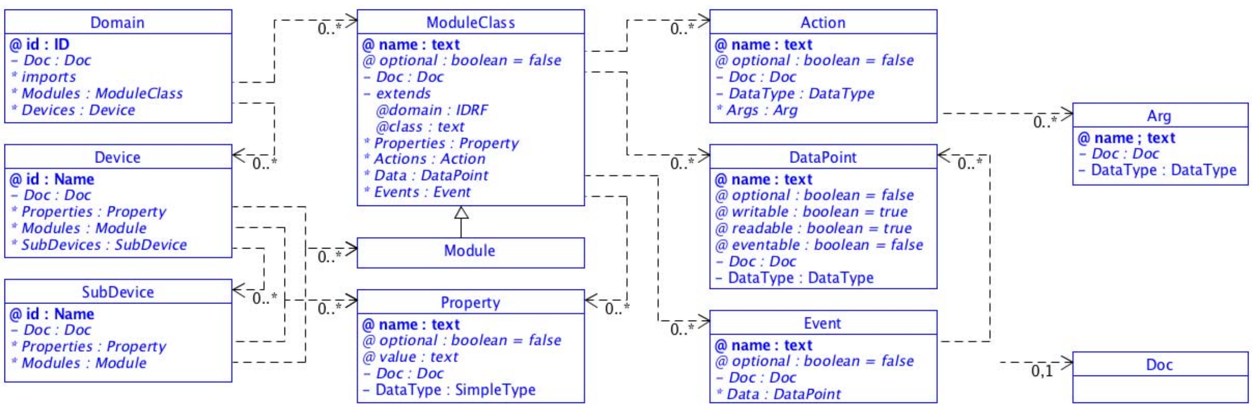


Figure 17: SDT structure overview

Figure 17 presents an overview of the structure of SDT. There can be different choices of the details of a SDT, each one optimized for a particular market segment and the types of devices used in that market segment. SDT structure is composed mainly with the following elements:

- Domain allows labelling of different SDT templates for different technologies and/or industry segments.

- Device may represent basic or complex things. All the different devices which one needs to model within a Domain are composed of one or more Modules.

- SubDevices are optional components of a Device. They represent physical sub-devices and services inside another device.

- Property is used to append to Devices and their ModuleClass with arbitrary additional information.
- Module is basically constraints or templates for how to model functionality of real things or appliances within the Domain.
- ModuleClass is defined at the Domain level. Each one describes some functionality and services and are reused as much as possible.
- DataPoint represents an aspect of a device which can be read/written.
- Action is an efficient way of describing arbitrary sequences of operations/methods required for in automation.
- Event is needed for automation protocols which "push" information, instead of relying on polling by the software application.

5.2.3.4 NGSI-LD

In any IoT system, it is important to gather and manage context information, processing that information and informing external actors, enabling them to actuate and therefore alter or enrich the current context. The ETSI Industry Specification Group for cross-cutting Context Information Management (ISG CIM) is working on how to standardize context data modelling within a smart application incorporating the latest advances from Linked Data and is defining a Context Information Management API which provides a simple way to query and update contextual data. ETSI ISG CIM working group defines an API called NGSI-LD.

The NGSI-LD Information Model prescribes the structure of context information supported by an NGSI-LD system. It is defined at two levels: the Core Meta-model and the Cross-Domain Ontology. The former amounts to a formal specification of the "property graph" model. In the other hand, the cross-domain ontology is a set of generic, transversal classes which aim to avoid conflicting or redundant definitions of the same classes in each of the domain-specific ontologies. Below these two levels, domain-specific ontologies or vocabularies can be devised. For example, the SAREF Ontology can be mapped to the NGSI-LD Information Model.

Figure 18 illustrates NGSI-LD Meta-Model in terms of classes and their relationships, NGSI-LD Cross-Domain Ontology and an example of a NGSI-LD domain-specific model.

NSGSI-LD Meta-model elements are:

- NGSI-LD Entity, NGSI-LD Relationship and NGSI-LD Property which are subclasses of rdfs:Resource.
- An NGSI-LD Value represents complex data structures and can be either an rdfs:Literal or a node object (in JSON-LD language).
- An NGSI-LD Property is of type rdf:Property.
- An NGSI-LD Relationship has an object stated through hasObject which is of type rdf:Property.

NGSI-LD Cross-Domain Ontology elements are:

- Location Properties: convey geospatial information.
- Temporal Properties: convey temporal information.
- unitCode Property: provides the units of measurement of an NGSI-LD Value.
- Geometry Values: they are a special type of NGSI-LD Value intended to convey geometries corresponding to geospatial properties.
- Time Values: they are a special type of NGSI-LD Value intended to convey time instants or intervals representations.

Figure 18 shows an example of a NGSI-LD domain-specific model which introduces specific entity types required for a particular domain, parking.

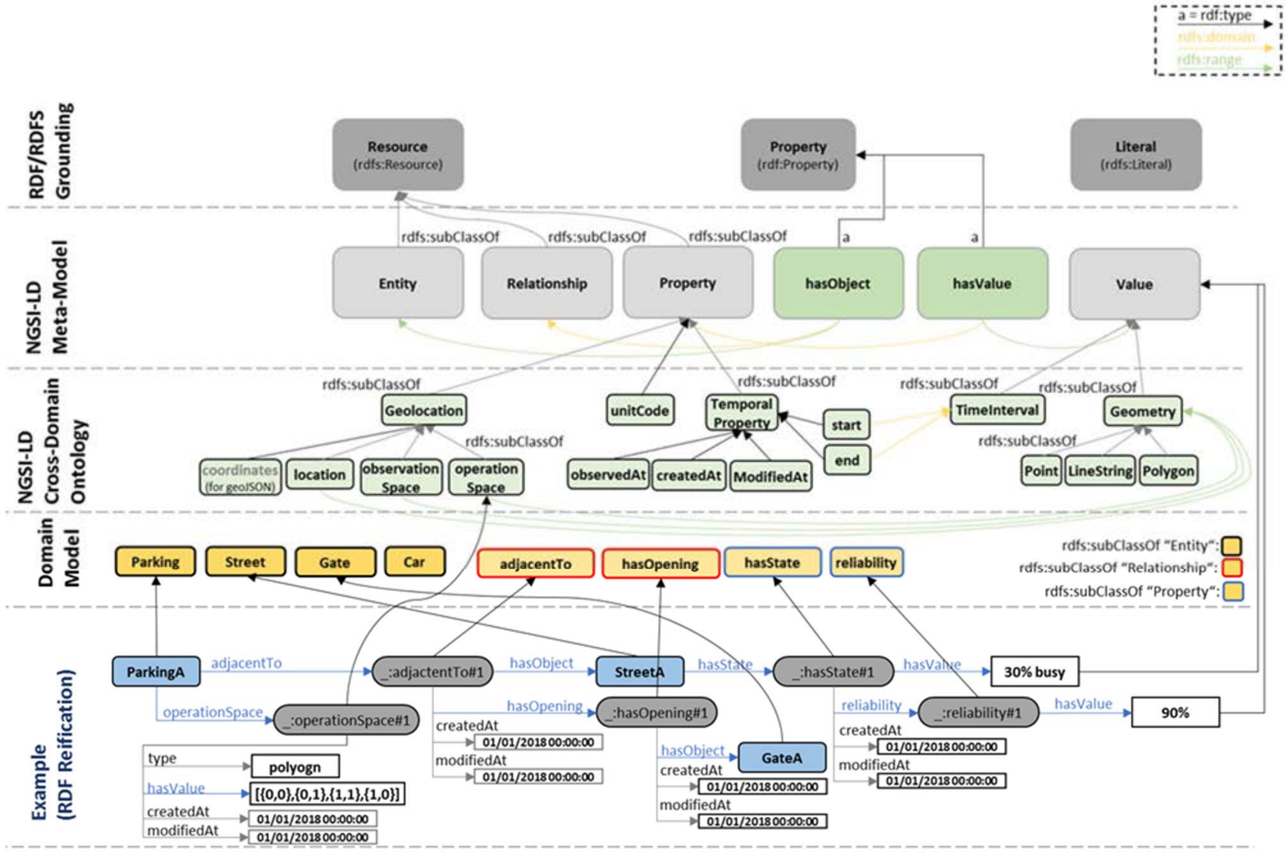


Figure 18: NGSI-LD Information model

5.2.3.5 OPC-UA

OPC UA (Unified Architecture) is a standard for horizontal communication from machine to machine (M2M) and for vertical communication. It is promoted as the foundation for digitalisation in the context of Industry 4.0. OPC UA provides a framework that can be used to represent complex information as Objects in an Address Space. These Objects consist of Nodes connected by References.

For the purpose of semantic interoperability, the unification of the information representation between information producers (servers) and consumers (clients) is based on the notion of type. Type definitions may be abstract and may be inherited by new types to reflect polymorphism.

There is one overall OPC UA information model, which describes all basic types. This information model is incorporated into every OPC UA server and can be used by developers as a foundation for the representation of their own specific data model.

Furthermore, OPC UA supports the notion of Companion Industry Standards Information Models for vertical standardization, plus vendor specific extensibility as depicted in Figure 19.

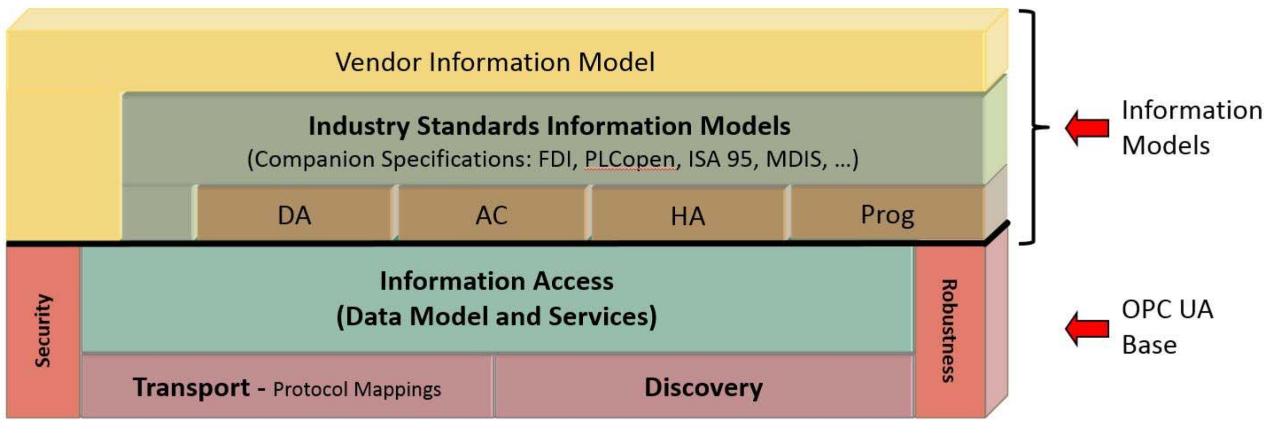


Figure 19: OPC-UA support for Information Models

Several Companion Industry Standards are being defined and extended by various Industry Organizations. Most likely, a Companion Standard is articulated in several Standards, often nested or otherwise related to each other as shown in Figure 20.

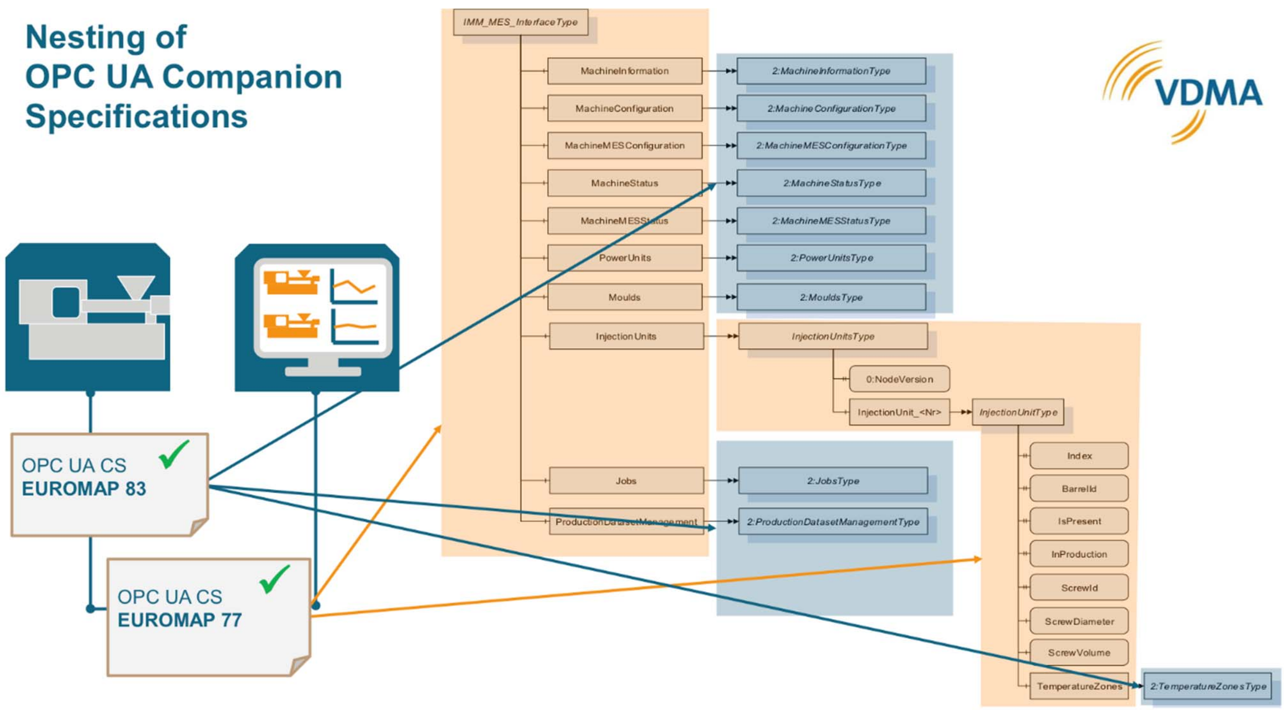


Figure 20: OPC UA Companion Specifications

EUROMAP can be taken as an example. EUROMAP is the European umbrellas association of plastics and rubber machinery manufacturers. It provides technical recommendations for this branch of the industry defining, besides others, mechanical and electrical interfaces between the machines. EUROMAP publishes two CSs:

- EUROMAP 83 specifies General Type definitions. This is the basis for all other EUROMAP interfaces based on OPC UA.
- EUROMAP 77 specifies the Data exchange between injection moulding machines and MES.

5.2.3.6 ETSI SAREF

The Smart Appliances/Applications Reference Ontology (SAREF) [i.23] is a standardized ontology for IoT devices and solutions published by ETSI in a series of Technical Specifications. The initial Technical Specification was published in November 2015 and updated in 2016 [i.20]. Its objective was to build a reference ontology [i.29] as an "interoperability language" for appliances relevant for energy efficiency and provide a shared model of consensus that facilitates the matching of existing assets (standards, protocols, data models, etc.) in the Smart Appliances domain.

SAREF is conceived in a modular way in order to allow the definition of any device from pre-defined building blocks, based on the function(s) that the device performs. These building blocks allow separation and recombination of different parts of the ontology depending on specific needs. Moreover, a SAREF device can be used for the purpose of offering a commodity, such as water or gas. It can also measure a property, such as temperature, energy or smoke. Moreover, a device may consist of other devices. Figure 21 shows an overview of the main classes of SAREF and their relationships.

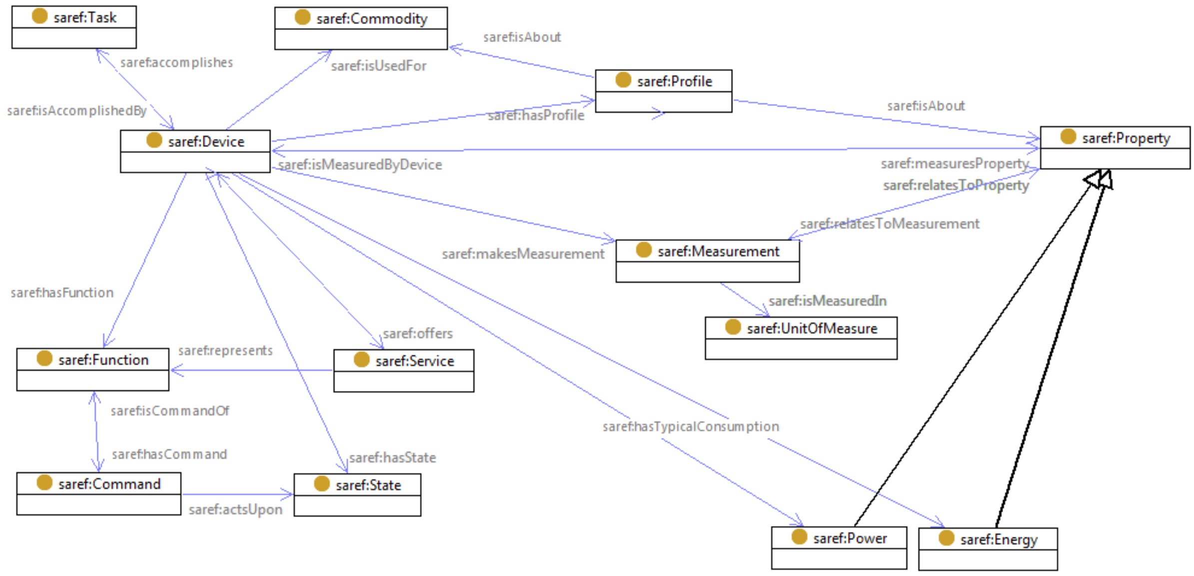


Figure 21: Main classes of the SAREF ontology

Devices can be classified in types (subclasses) that reflect different trends of usage. Therefore, according to this distinction, devices are classified in three main categories: function-related, energy-related and building-related, respectively. These categories are shown in Figure 22 as subclasses of the Device class.

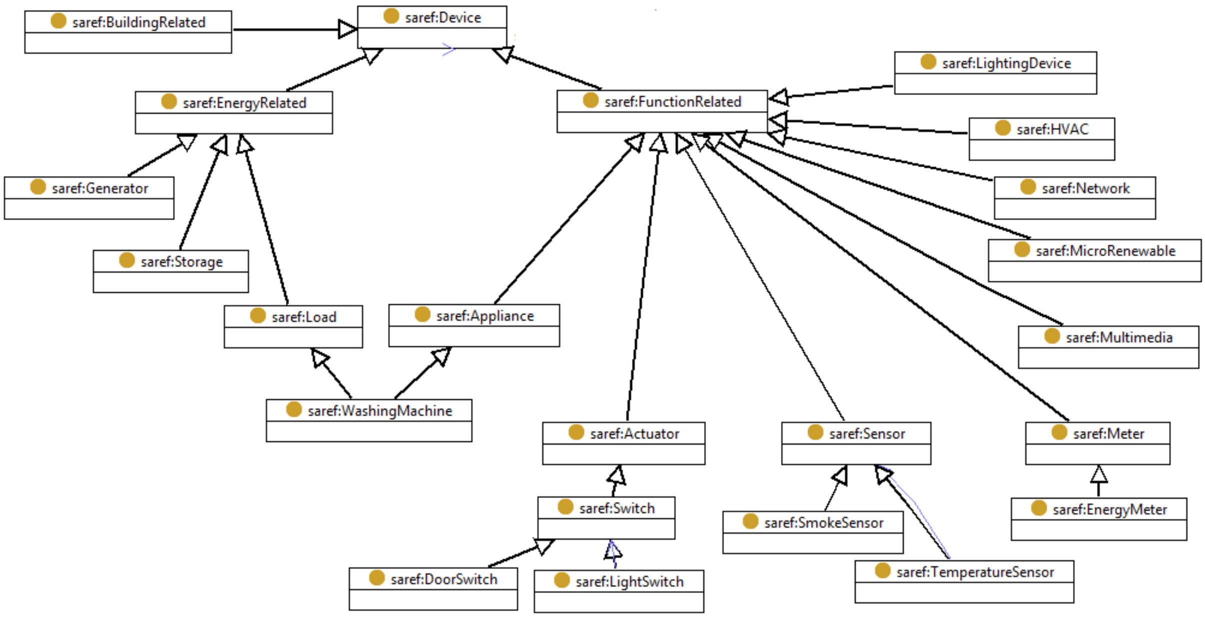


Figure 22: SAREF type of devices

Main directions

SAREF is the reference ontology for smart appliances and contains recurring concepts that are used in several domains. Currently, the proposal to change the SAREF acronym from the original "Smart Appliances REFerence ontology" to, e.g. "Smart Applications REFerence ontology" as accepted (in ETSI TC SmartM2M), to better reflect the fact that SAREF is not limited to smart appliances and energy efficiency but can serve as upper reference model to enable better integration of data from various vertical domains in the IoT. As smart appliances are not restricted to only one domain, it is possible that specific concepts for a certain domain are not part of SAREF.

Since the very first specification that addressed Smart Appliances [i.20], SAREF has been extended [i.21] to provide different domains with a proper ontology that reflects the specific needs of that domain. First extensions to SAREF cover SAREF for Energy, SAREF for Environment and SAREF for Buildings (January 2017). From Smart Appliances, SAREF is being evolved to cover the following domains: Smart Cities, Smart AgriFood, Smart Industry & Manufacturing in a first step, then Smart Automotive, eHealth/Ageing-well, Wearables and Water domains in a second step.

Another ongoing activity has the objective to consolidate the SAREF ontology with new reference ontology patterns, using the experience of the SEAS (Smart Energy Aware Systems) project mainly in the Smart Energy domain, e.g. with description of the physical systems and their connections, value association for their properties and the activities by which such value association is done.

Relationship with oneM2M base ontology

The SAREF ontology supports a direct mapping with the oneM2M Base Ontology [i.22] and thus runs with oneM2M-compliant communication platforms. Figure 23 shows the mapping between SAREF and the oneM2M Base Ontology. It is described in detail in [i.20] and [i.22] and illustrated in Figure 23.

In general, the oneM2M system needs to represent knowledge as a hierarchy of concepts (ontologies), either external or internal to the oneM2M domain, using a shared vocabulary to denote the classes, properties and interrelationships of those concepts. Storage, discovery and management of ontologies (including both oneM2M Base Ontology and external ontologies e.g. SSN [i.24], SAREF) within the oneM2M platform enable the support of basic and advanced semantic functionalities within the oneM2M platform.

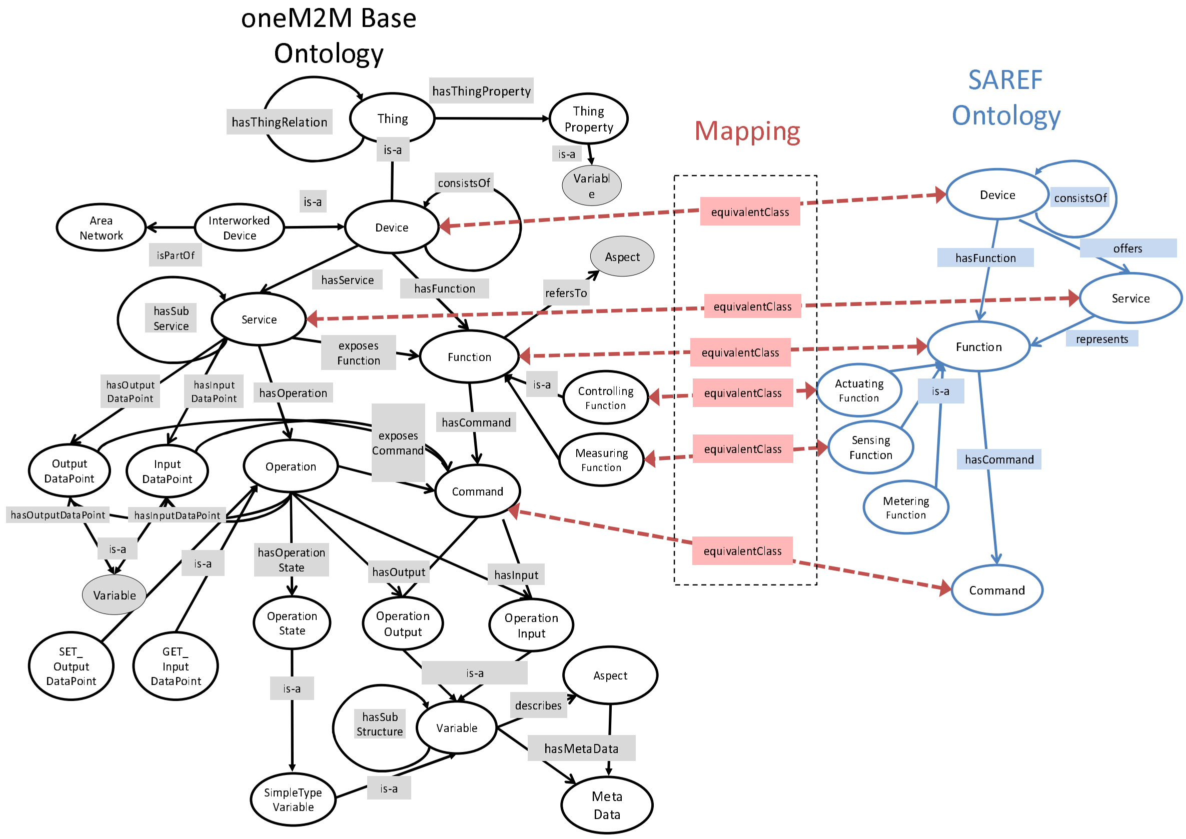


Figure 23: Mapping between SAREF and the oneM2M Base Ontology

The SAREF initiative has been welcomed by the Smart Appliance and IoT Industry (including the Smart Watering) which clearly indicated the intention to adopt the SAREF ontology and its related communication framework. As confirmed in the EC Rolling Plan for ICT Standardization 2017, SAREF is a main ontology standard in the IoT ecosystem and sets a template and a base for the development of similar standards for the other verticals to unlock the full potential of the IoT.

The availability of a network of standardized semantic models that consistently grow and systematically extend SAREF allows implementers and manufacturers of Smart Appliances - and more in general IoT devices - to fully support various, multiple and cross-domain use cases for their devices. It enhances the interoperability between their devices and the devices of other manufacturers and allows them to broaden their market.

SAREF has been adopted by the Energy@home and EEBus industry associations as a basis for an extension - called SAREF4ENER - to interconnect their (different) data models. SAREF was used as the core technology and standard for the DSF (Demand-side Flexibility) development and Proof-Of- Concept demonstration targeting Smart Energy distribution to individual homes. The study demonstrated an integrated infrastructure that seamlessly uses different standards for Energy, Smart Grids, Smart Meters, Smart Appliances and Machine to Machine (M2M) communication. The demonstration showcased the alignment of ETSI SAREF4ENER, oneM2M Architecture, oneM2M Base Ontology, oneM2M protocol and EEBus SPINE. It was shown at three major events: European Utility Week (3-5/10/17 Amsterdam), ETSI IoT week (23-26/10/17 Sophia Antipolis) and Dedicated SAREF/Study SMART 2016/0082 event (27-29/11/17 EC premises). The ETSI Smart Appliances specification aims to be deployed in the European market in a potential of 250 million European dwellings as a first step and potentially worldwide later.

5.2.3.7 W3C SSN

The Semantic Sensor Network (SSN) ontology is a widely-recognized ontology published by W3C as a Recommendation and currently in its Version 2.0 since October 2017 [i.25]. It is a joint contribution with the Open Geospatial Consortium (OGC) standard, extending and improving the SSN ontology published in 2011.

SSN focuses on the description of sensing devices and the observations they make of the physical world, the involved procedures, the studied features of interest, the samples used to do so and the observed properties, as well as actuators. SSN follows a horizontal and vertical modularization architecture by including a lightweight but self-contained core ontology called SOSA (Sensor, Observation, Sample and Actuator) for its elementary classes and properties. SOSA acts as minimal interoperability fall-back level, i.e. it defines those common classes and properties for which data can be safely exchanged across all uses of SSN, its modules and SOSA. The SSN ontology supports a domain-independent and end-to-end model for sensing applications and it can be used with domain ontologies and other ontologies to model the observation and measurement data produced by the sensors. The Web Ontology Language Description Logic (OWL DL) is used to encode sensor descriptions considering mapping between the ontologies and OGC models.

The SSN ontology is published in a modular architecture that supports the use of the ontology for diverse applications. It revolves around the central Stimulus-Sensor-Observation pattern. Figure 24 and Figure 25 illustrate the SSN classes and properties, on observation and actuation perspectives respectively. With colour enabled, the figures also show the SOSA core ontology in green.

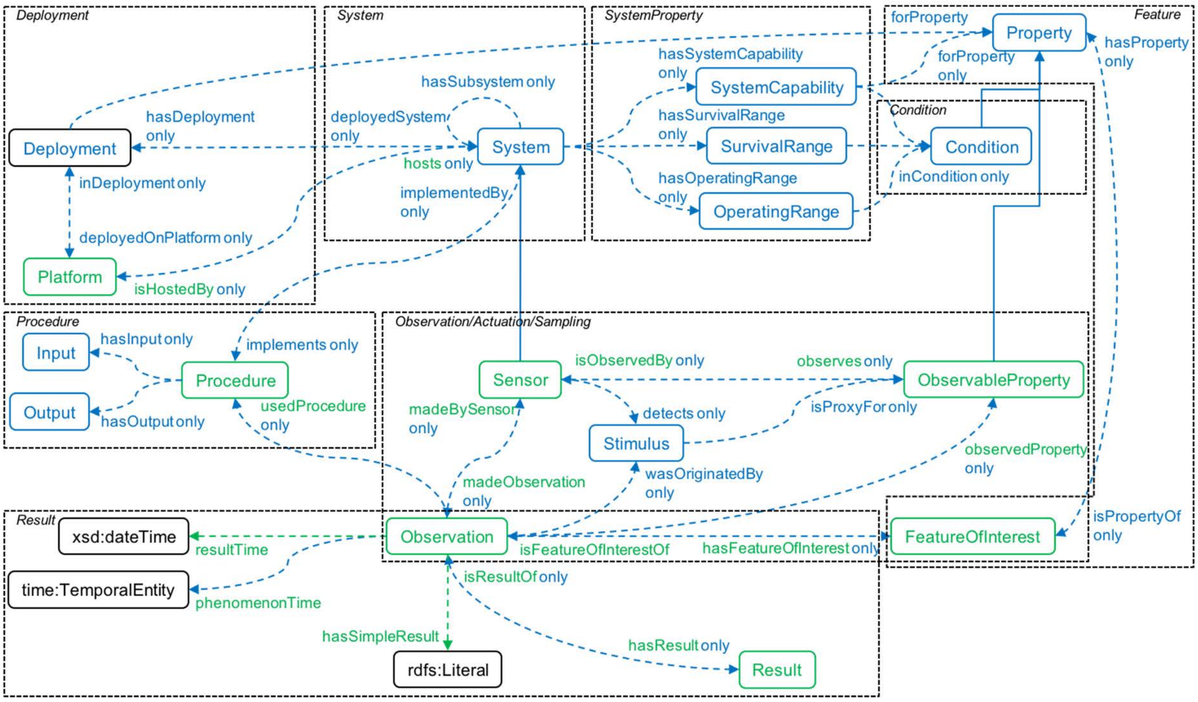


Figure 24: Overview of the SSN classes and properties (observation perspective)

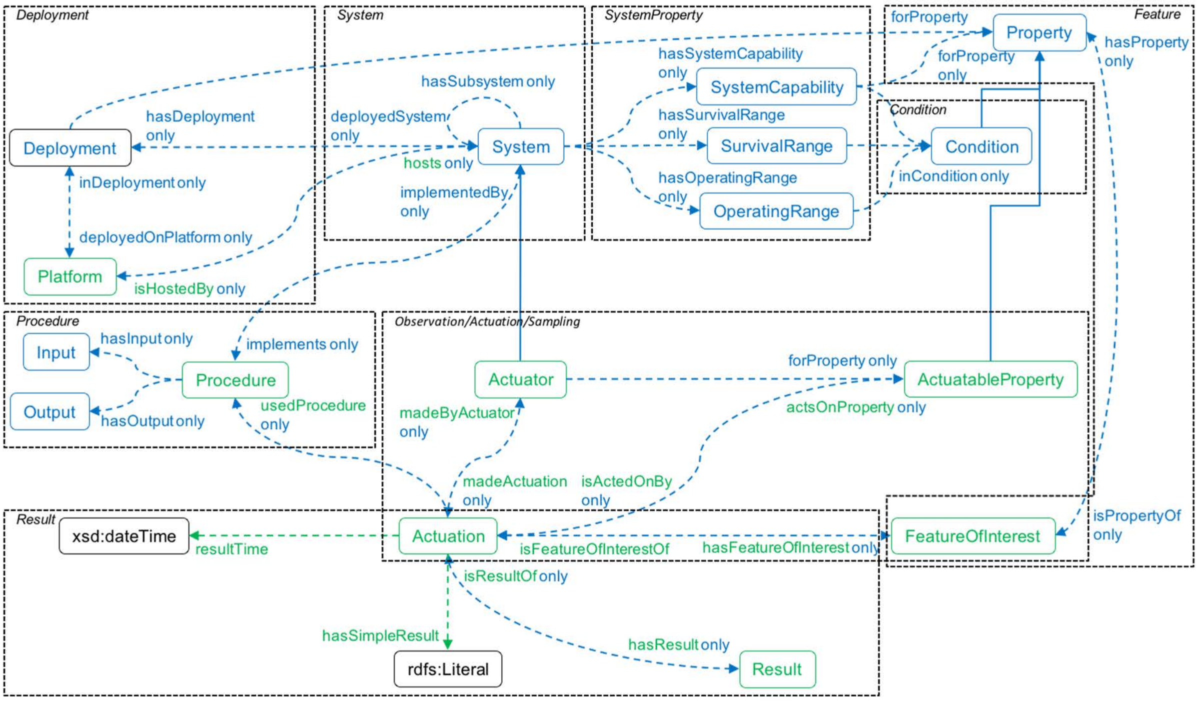


Figure 25: Overview of the SSN classes and properties (actuation perspective)

The figures also show the main conceptual modules (e.g. system, feature, condition, observation/actuation sampling, deployment, result, etc.) built on the pattern to cover the key sensor concepts and the main relationships between these modules [i.26]. Examples based on a certain number of use cases is provided with the ontology. The SSN ontology does not describe domain concepts, such as time and locations, since these concepts are intended to be included from other ontologies via OWL imports.

The SSN ontology is implemented in many platforms. A mapping between SAREF and the W3C® SSN ontology has been published in [i.27] with the objective to enable the semantic integration of IoT platforms relying on them, i.e. to align both ontologies. This work is a very preliminary evaluation based on the translation from SSN to SAREF of a wind sensor ontology. The mapping could be partially achieved, with issues such as the "lack of measurement capabilities in SAREF to represent the collection of measurement properties of a component of a sensor", the composition relationship between a system and its sub-systems in SSN or the necessary creation of a subclass for a new sensor type (e.g. *saref:WindSensor*) when it is not already specified in the ontology.

5.2.4 Industry Solutions

5.2.4.1 Watson

Watson IoT platform is an instantiation, a customization and an extension of Watson IoT products and services to IoT Focus Areas. Watson IoT Platform provides powerful application access to IoT devices and data to help rapidly compose analytics applications, visualization dashboards and mobile IoT apps. It allows to perform powerful device management operations, store and access to device data, connect a wide variety of devices and gateways. The Watson IoT Platform communicates with applications and devices by using the Watson IoT Platform API and the Watson IoT Platform messaging protocol.

Watson IoT Platform offers a rich data model to describe avertical domain in terms of device abstraction and data consumed and generated by interacting entities. The most relevant elements for the Watson interworking proxy are described in the following list:

- **Device Type:** Device types are intended to be groups of devices which share common characteristics. In order to register devices to Watson IoT Platform, the Watson interworking proxy needs to create a device type first.
- **Device:** A device is defined as an entity that has a connection to the internet, has data it wants to get into the cloud and can accept commands from applications as well.
- **Device Info:** Contains predefined properties related to a Device or a Device Type.

- **Metadata:** Stores custom attributes about device type or a device.
- **Event:** Events are the mechanism by which devices publish data to the Watson IoT Platform. Devices control the content of their messages and assign a name for each event that is sent. The Watson IoT Platform uses the credentials that are attached to each event received to determine which device sent the event. This architecture prevents devices from impersonating one another. Applications can process events in real time and see the source of the event and the data contained in the event. Applications have to be configured to define which devices and events they subscribe to.
- **Command:** Commands are the mechanism by which applications communicate with devices. Only applications can send commands and the commands are sent to specific devices. The device has to determine which action to take on receipt of any given command. Devices can be designed to listen for any command or to subscribe to a specified list of commands.

Figure 26 illustrates the main concepts of Watson IoT Platform data.

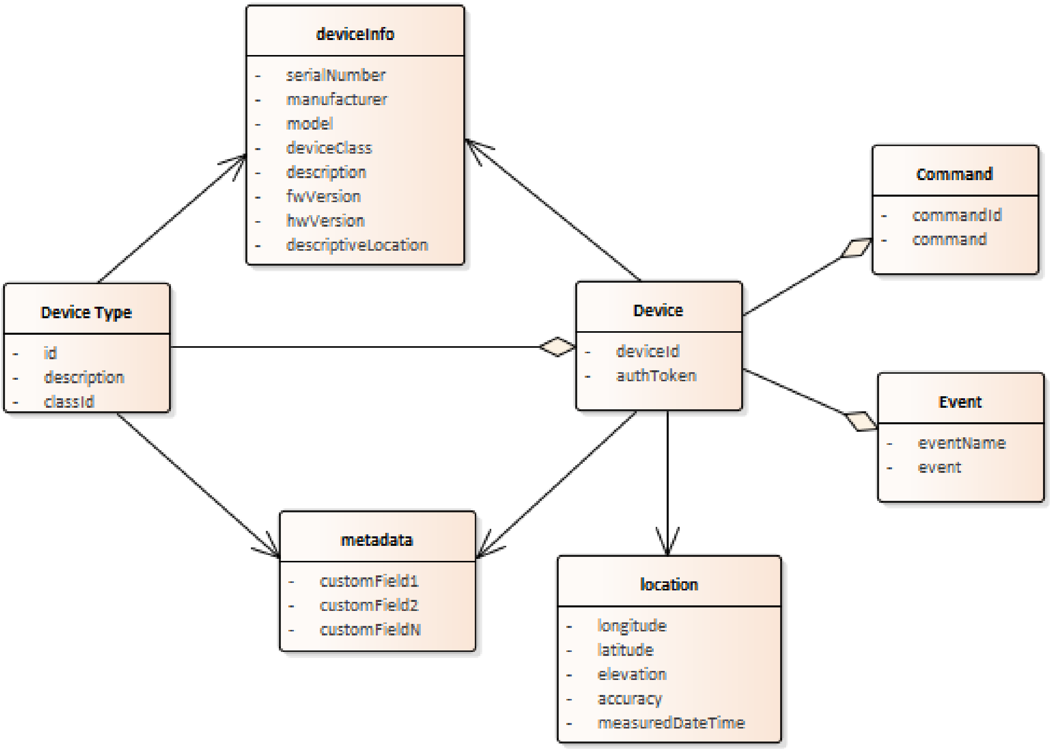


Figure 26: IBM Watson data model

5.2.5 Open source

5.2.5.1 Mainflux

Mainflux [i.28] is a secured open source and patent-free IoT cloud platform written in Go, based on a set of microservices. It allows device, user and application connections over various network protocols, like HTTP, MQTT, WebSocket and CoAP, making a seamless bridge between them.

It serves as software infrastructure and set of microservices for building complex IoT solutions, IoT applications and intelligent products. Mainflux exposes southbound API to connect devices and northbound API to connect applications, then it allows message routing between them. The whole bottom-to-top system (device, Mainflux platform and application) together forms what is called a "vertical solution," very specific to the given use case. The IoT Cloud Platform intends to facilitate the interoperability of an IoT solution with existing enterprise applications and other IoT solutions.

Figure 4-1 of [i.29] (Location of Mainflux inside the IoT system) available at https://www.oreilly.com/library/view/scalable-architecture-for/9781492024132/ch04.html depicts a typical IoT system and shows where Mainflux sits in the overall architecture.

In order to conform to existing standards as much as possible, the Mainflux system uses SenML (IETF RFC 8428 [i.33]) semantics as a message model coming from the clients. The platform uses its own format to encapsulate the SenML messages coming from the sensors. Each of the adapters constructs this type of message, putting in the "Payload" field the SenML content that came from the sensors. Each adapter also adds the message metadata, like the ID of the publisher or the protocol over which the message was published.

SenML

IETF RFC 8428 [i.33] was published in August 2018. It defines a format for representing simple sensor measurements and device parameters in Sensor Measurement Lists (SenML). Representations are defined in JavaScript Object Notation (JSON), Concise Binary Object Representation (CBOR), Extensible Markup Language (XML) and Efficient XML Interchange (EXI), which share the common SenML data model. A simple sensor, such as a temperature sensor, could use one of these media types in protocols such as HTTP or the Constrained Application Protocol (CoAP) to transport the measurements of the sensor or to be configured. Figure 27 illustrates the SenML labels defined in the RFC. The SenML provides minimal semantics with the objective that processors with very limited capabilities could easily encode a sensor measurement into the media type, while at the same time, a server parsing the data could collect a large number of sensor measurements in a relatively efficient manner.

Name	Label	CBOR Label	JSON Type	XML Type
Base Name	bn	-2	String	string
Base Time	bt	-3	Number	double
Base Unit	bu	-4	String	string
Base Value	bv	-5	Number	double
Base Sum	bs	-6	Number	double
Base Version	bver	-1	Number	int
Name	n	0	String	string
Unit	u	1	String	string
Value	v	2	Number	double
String Value	vs	3	String	string
Boolean Value	vb	4	Boolean	boolean
Data Value	vd	8	String (*)	string (*)
Sum	s	5	Number	double
Time	t	6	Number	double
Update Time	ut	7	Number	double

Figure 27: SenML data model labels

5.2.6 Other projects

5.2.6.1 Pilot Test for interfacing oneM2M platform with Smart Agriculture (STF-542)

The objective of this pilot test [i.30] is to validate the possible cooperation between the oneM2M platform and AEF ISO 11783 standards [i.31] implemented for communication inside and between agriculture & forestry machines. ETSI TC ITS standards, such as ETSI EN 302 637-3 [i.32] (Decentralized Environmental Notification Basic Service) are part of this cooperation in the use case to be demonstrated during the pilot. The main scenario envisioned for this pilot consists in the dissemination of a warning message to road vehicles as soon as an agriculture or forestry equipment leaves the field for road transport. The coordination between the detection of this event and the sending of the notification message is done using an oneM2M platform in the tractor.

Figure 28 illustrates the functional architecture and interworking between the two domains, built upon the different protocol stacks from each domain. The AGRI-IPE is seen as a specialized on the ISO 11783 [i.31] network. The pilot test is controlled by an application identified as Application Entity (AE) for Safety. This application hosts an algorithm that triggers an actuator when a safety condition is met and an alert should be sent to the neighbouring vehicles. The ITS-IPE is seen as an application by the ITS protocol stack. It subscribes to the EventTrigger resource of the Safety AE. When receiving a notification on the EventTrigger resource (change to ON), it retrieves the latest instance of all resources associated to the AGRI-IPE and triggers the ITS protocol stack to send an alert.

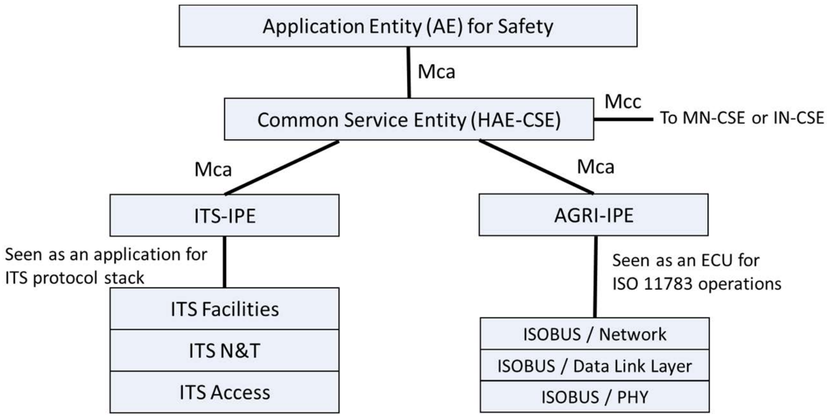


Figure 28: Interworking Reference Model in the Agriculture Equipment

This pilot test is based on a common data model. The data considered include the parameters of the ITS messages and those required to assess the triggering condition. Each of the parameters is associated to a set of meta-data: parameter name, type, size and valid range for values. The parameters have been split into four groups, depending on whether they are originating from the ISO 11783 [i.31] (used/not used for triggering the event) or from the ITS side. An additional group contains purely internal parameters.

Figure 29 illustrates a preliminary semantic model that has been defined for the pilot test data, with the objective to remain close to the oneM2M base ontology and SAREF. It is built around the Agri Equipment device and addresses mainly the parameters common to both ISO 11783 [i.31] and C-ITS standards (e.g. Location-time information and Vehicle State information) [i.32] for the cooperation between both IPEs through the M2M platform.

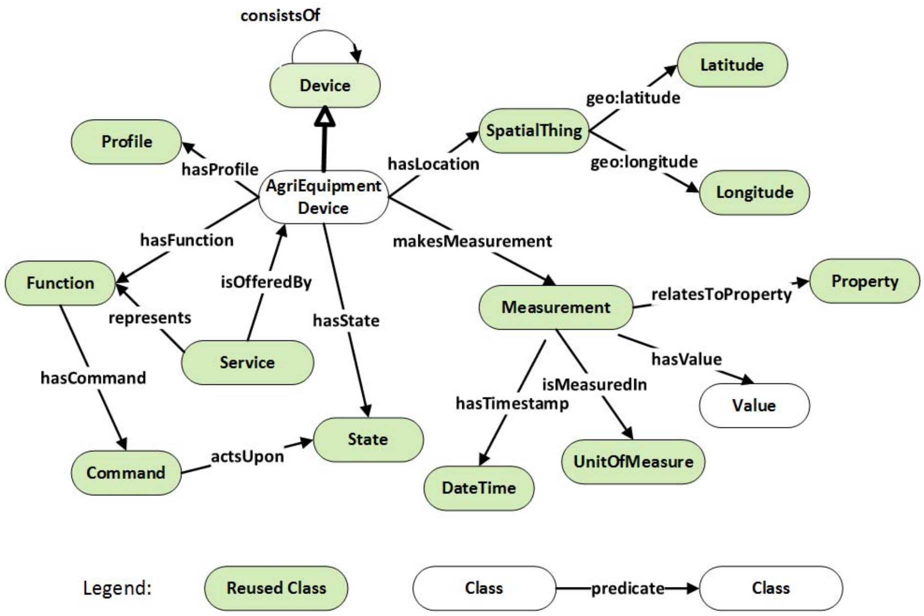


Figure 29: Pilot semantic model (main classes)

6 Semantic interoperability adoption analysis

6.1 The need for semantic interoperability in industry

The lack of semantic interoperability is often considered the largest market inhibitor for the uptake of IoT:

- IoT service providers are faced with heterogeneous and vendor-specific installations. Centralized management of IoT solution oftentimes forces the owners to go through costly replacements to adopt mono-vendor solutions.
- Installation of new equipment requires costly system integration because devices are often designed to communicate with specific applications only.
- There is no uniform manner to access and filter the huge amounts of datasets that are generated. Huge amounts of data are generated but never get analysed and used.
- IoT systems remain isolated from their surroundings and environment, resulting in poor or non-existing synergies.

Semantic interoperability in industry allows:

- Centralized management of heterogenous IoT infrastructure allows increased efficiency by setting global policies, quicker reactions and optimized decisions across all buildings. It also brings-down operational costs thanks to a single software set. The challenge is how to get there without retrofitting the building to a single technology/provider.
- Continuous solution integration/operation: The main idea is to quickly plug and play new equipment, networks and services in a cost-efficient manner and without disturbing the ongoing IoT system management operations.
- Efficient data exposure: IoT devices generate huge amounts of data. The exposure of these data sets through modern APIs allows proliferation of new services such as situational awareness, energy efficiency, preventive maintenance and smart data.
- Wider integration allows the IoT system to give rise to fully integrated solution supporting mass scale deployment in multiple domains. The IoT system stops existing on its own and starts to interwork with other verticals.

6.2 Status of semantic adoption by industry

6.2.1 Introduction

Today, companies deal with the lack of interoperability using different approaches depending on customer needs and internal skills. Approaches go from manual file export and import, Extract, Transform and Load tools, Point-to-Point integration, Enterprise Service Bus, Integration Platform as a Service, to semantic interoperability platforms.

6.2.2 Manual file export and import

The simplest method of integrating data from one application to another is to export the data to a file and import the file into the target system. An integrator might export records from an application to a file and then manually import the file into your IoT platform. This approach has limitations. In general, fields are not the same in both systems. An integrator will have to transform the data before importing which takes time and often introduces errors.

6.2.3 Extract, Transform and Load (ETL)

ETL automates the process of integration data between two systems and keeping data synchronized with minimal human intervention. A copy of data is extracted from a source, then translated to match specific format and loaded into the destination system. Many companies use ETL software to extract, transform and load data from different IoT systems into a specific platform for reporting and data analytics. They move data in batches, often on an hourly or daily basis which is applicable only to non-time-sensitive data.

6.2.4 Point-to-Point integration (P2P)

Most enterprises implement Point-to-Point (P2P) connections between applications for near real time processes such as monitoring, alerting or triggering. As the number of applications in an enterprise increases, the number of point-to-point connections required becomes unmanageable. Maintaining P2P integrations becomes a burden driving the company into an endless cycle of regression testing and fixing.

6.2.5 Enterprise Service Bus (ESB)

ESB uses a hub-and-spoke approach in place of many point-to-point connections. The ESB serves as the hub with spokes connecting to the other applications. It acts as a central broker, accepting messages from one application, performing integration tasks and sending messages to another application through near real time communication. The company builds connections between their applications and the ESB and they code the transformation and other integration tasks that are required.

6.2.6 Integration Platform as a Service (iPaaS)

iPaaS It is another hub-and-spoke approach, but it provides the capabilities of the other approaches as a multi-tenant cloud service. An iPaaS offers a user-friendly dashboard for designing and maintaining connections and integrations, monitoring results and resolving errors. It comes with a broad array of application and technology connectors. An iPaaS includes connectors to main southbound devices, northbound applications. Many iPaaS offers a software developer kit that enables customers and partners to build their own connectors. Some also provide a marketplace to make certified third-party connectors available to other users of the platform. The iPaaS vendor keeps connectors up to date as SaaS companies update their products, testing new versions and making changes as necessary.

6.2.7 Semantic interoperability platform

Semantic interoperability platform enables heterogeneous devices and applications to understand exchanged data in a similar way, implying a precise and unambiguous meaning of the exchanged information. It takes advantage of both the structuring of the data exchange and the codification of the data, including standard and ontologies so the interacting can interpret the data. The support of semantic interoperability paves the way to computable logic, inferencing, knowledge discovery and data federation between different processes. E.g. oneM2M platform extended with SAREF ontology.

6.3 Market drivers

6.3.0 Introduction

Several market drivers affected positively the adoption of the semantic by industry [i.21] such as improving existing services, providing new services and public policy support.

6.3.1 Improving existing services

When an existing system is unsatisfactory, the user needs to improve the existing system is strong. From the perspective of IoT platform providers, semantic is a solution to the lack of interoperability. In some cases, users first required an introduction of applications of semantic technologies and then vendors participated in its development project. However, in many cases, vendors promoted first the improvements of services through the implementation of information systems applying to semantic technologies. This fact shows that vendors have to be proactive at the early introduction of semantic to their customers.

6.3.2 Providing new services

Requirements for providing innovative services, which were difficult previously, became the motive for adopting applications based on Semantic. Semantic technology was also introduced to devise business models for new services. From the perspective of suppliers, the adoption of semantic resulted from user requirements such as context-awareness, collaboration, data sharing and automation and many others required today by industrial domains. In a variety of industrial areas including smart cities and industry 4.0, user requirements for context-aware services beyond simply providing information became the motive for the adoption of semantic.

6.3.3 Public policy support

Public policy support played a crucial role in the adoption of semantic interoperability in industry. Many companies indicated that public sponsorship for the projects and proactive roles of the relevant agencies including standardization bodies related to semantic led to increased focus on semantic and its adoption. Projects for technology development and its applications, that are supported by the public organizations, deploy knowledge necessary to technology innovation and become a driving force for innovation diffusion.

6.4 Market inhibitors

6.4.1 Introduction

Several market inhibitors affected negatively the adoption of the semantic by industry [i.34] such as lack of environmental conduciveness, lack of killer applications and successful cases, complexity and immaturity, uncertainty regarding scalability and performance and difficulties to perceive immediate value.

6.4.2 Lack of familiarity with semantic

The current situation is not yet propitious to the awareness for semantic due to immature supplier technology, weak development capabilities, insufficiency of experts and culture issues in industry. The following factors are necessary for fostering a suitable environment: proactive marketing activities, positive recognition among users, collaborative efforts, ontology expert's cultivation and user-friendly modelling tools and procedures.

6.4.3 Lack of killer applications and successful cases

Killer applications and successful cases become an excellent guideline for the successful adoption of the semantic adoption. Semantic in industry will proliferate when it is highly successful in a certain area. Users want to demonstrate systems or predict test results before they adopt the semantic technologies. Unfortunately, suppliers suffer from problems regarding demonstration, observation and verifiability of the system because reference models and killer applications are lacking.

6.4.4 Complexity and immaturity

Many developers feel that semantic is complex to understand in terms of its application process. Complexity makes developers feel uncertain about the result of semantic adoption. They have a low opinion of the maturity level of Semantic tools as a result of the perceived gap between academic and industrial perspectives.

6.4.5 Uncertainty regarding scalability and performance

Current semantic reasoning systems have difficulties processing large-scale data. lack of technology standards and solutions or tools supporting project development, difficulty in cost projection and quality assurance, requirements for greater intensive knowledge on domains as well as development methodologies are causing uncertainty regarding ontology adoption promises.

6.4.6 Difficulties to perceive immediate value

The potential value of a new technology is associated with the perception of its benefits. However, semantic interoperability is a long process. The reason behind the effort for semantic adoption should be the expectation of improving services and their utilization in the mid-long future rather than immediate increase in productivity, such as cost reduction and high efficiency.

6.5 The ontology problem

6.5.1 Introduction

The ontology problem ([i.34] and [i.35]) is a fundamental challenge for semantic adoption by industry including upper ontology, domain ontology and ontology Integration issues.

6.5.2 No generally-accepted upper ontology in use today

There is no generally-accepted, comprehensive, standardized upper ontology in use today. Upper Ontologies are difficult to design compared to domain ontologies because they describe our consensus reality and concepts, they define are more abstract. The skillsets needed to design Upper ontologies are quite different domain ontologies. Ontology developers do not have many options to develop their domain ontology. They could for example develop their own upper ontology first. A huge task that they should not have to undertake and probably do not have time to complete. Developers could try to use one of the various existing upper ontologies. Developers could decide to just not use an upper ontology which makes things simpler for the moment but making the concepts in the ontology ambiguous and essentially undefined. It is clear here that there is no an optimal choice when dealing with upper ontologies.

6.5.3 Many fragmented knowledge niches

There are many knowledge niches containing tens of thousands of class definitions that still relatively limited in their conceptual breadth, depth and resolution. Today, most vertical domains have yet to be modelled ontologically. However, before creating any new domain-ontology, someone has to come up with a concrete benefit for doing so for example, applications or services that make use of this domain-ontology to solve a real problem. Sharing data in a seamless way requires ontologies to be made publicly available in some manner. In addition, they need to somehow to be connected together so that they can be normalized and mapped to one another easily using for example an upper ontology. Until that happens, achieving semantic interoperability will remain very limited to some scenario dealing with some general knowledge or working with domain-specific concepts defined by the small set of currently existing domain ontologies.

6.5.4 The ontology integration nightmare

It easy for developers to develop new ontologies from scratch but quite hard to make them compatible with other existing ontologies. This could explain why, today, there are few mappings between existing ontologies and an increasing number of small, non-integrated ontologies about different domains. In theory, it should be easy to integrate all ontologies together, however the task of actual doing such integration is difficult in practice. Integrating ontologies is not as simple as just mapping classes in one ontology to corresponding classes in another ontology. Because it turns out that it is not merely the names and properties of classes that are significant to defining their meanings and mappings, but also their inheritance paths in their respective ontologies. The difficulty in integrating ontologies is in figuring out how to express the similarity and difference in meaning between concepts, relationships, attributes and their constraints. In addition, the complexity of such integration increases exponentially to the number of concepts being integrated.

7 Guidelines for using semantic interoperability in the industry

7.1 Introduction

This clause proposes a set of guidelines regarding the successful adoption of semantic interoperability by industry covering technical and non-technical aspects. These guidelines are of two kinds:

- Strategy guidelines indented primarily for the persons in charge of making high-level strategy decisions as to the directions to be taken by the company technical roadmap, understand the implications on the organization and how the skills of the engineers can be adapted.
- Technical guidelines intended primarily for the IoT system designers and developers when they will have to make choices regarding the type of technical solution to select.

7.2 Strategy guidelines

7.2.1 Decide adoption and promote it

Absorptive capacity is the ability of companies to recognize the value of new, external information, assimilate it and apply it to commercial ends. It means the ability to evaluate, accept and apply innovation to achieve organizational objectives and depends on knowledge source and prior knowledge and it influences innovation adoption. When dealing with semantic, proactive attitude in analysing trends or technological features and a determined will for a successful introduction is required for semantic adoption. In addition to efforts in analysing technological trends, experts have to persuade internally their department heads and resolved any conflict with managers who have a negative opinion of the semantic.

7.2.2 Invest in communication and training

Company need to provide educational programs for developers who do not have enough understanding or knowledge of semantic and persuaded them to participate in the programs. In general, the degree of academic knowledge of emerging technologies such as semantic is higher than that of company organizations. Under such environment, company efforts to communicate with their developers and train them is essential to overcome their knowledge gap and can align the capability of the semantic with the needs of customers.

7.2.3 Outline expectation upfront

There is a significant discrepancy in expectations between suppliers and users. Users expect that many more things will be possible through the adoption of semantic, while suppliers recognize that it is difficult to reveal demonstrable effects that would match user expectations. In other words, there is a gap between the user perspective expecting substantial performance and that of supplier recognizing some limitations due to the early stage nature of semantic. Some developers believe that their services would improve with the semantic without considering consider whether the semantic is appropriate for their services. The gap resulted from the frequent promotion that the reasoning engine can enable fantastic services that are not possible with existing technologies such as database and data mining.

7.2.4 Promote success and expand diffusion

Even though semantic is adopted, further efforts will be necessary to make it easier for the system to get diffused in an organization. A stage model of technology diffusion consists of initiation, adoption and acceptance, adaptation, routinization and infusion. The level of services in terms of both quantity and quality has not only to reach a critical mass, but ontologies need also to be shared to be cost-effective. An increased investment budget for extending systems based on semantic enables such systems to offer sustainable services that demonstrate positive results, such as service improvement and productivity.

7.3 Technical Guidelines

7.3.1 Use an upper ontology

Upper ontologies such as oneM2M base ontology provide a common ontological foundation for semantic interoperability across domains. They offer a high-level compatibility and plausibility check for domain ontologies and their semantic integration. In general, fundamental concepts defined by upper ontologies covers space and time, categories and individuals, time and space, objects and processes, etc. E.g. oneM2M Base ontology is an upper ontology that provide general concepts required for IoT semantic interoperability such as Device, Function, Service, Location, Operation, Input, Output, etc. which are common to multiple IoT domains.

7.3.2 Reuse existing domain ontologies

The ability to effectively and efficiently perform ontology reuse represents a potential solution to the problem of standardization and paves the way towards the realization of the Semantic interoperability. It is more cost effective to build an ontology reusing existing ontologies than from scratch. Reusable domain ontologies like ETSI SAREF provide opportunities for developers to exploit and reuse existing concepts and relationships to integrate their devices and build their applications with much ease and efficiency. However, reusing an ontology is far from an automated process and instead requires significant effort from developers and knowledge experts.

7.3.3 Insert ontologies in the development process

In general, when it comes to developing larger scale IoT systems, many companies prefer to start the project with small Proof of Concepts (PoC) limited in terms of technologies, data sources and scope. During the PoC phase, the need for semantic interoperability is not necessarily visible. By analogy with security, if not initially anticipated, semantic interoperability becomes extremely costly and almost impossible to integrate properly in the future. For this reason, semantic interoperability in general and ontologies in particular should be inserted at an early stage in the development process to ease the mass scale deployments of IoT systems and avoid vendor-lock in.

Annex A: Change History

Date	Version	Information about changes
September 2018	0.0.2	Uploaded on SmartM2M portal for discussion at SmartM2M #47 (replaced by V0.6.3)
September 2018	0.0.3	Uploaded on SmartM2M portal for discussion at SmartM2M #47
January 2019	0.0.14	Final inclusion of content for clauses 4 to 7
March 2019	0.2.1	Final Draft, comments from TC SmartM2M review of Stable Draft incorporated
March 2019	0.2.2	Document revision changed from 2.0 to 2.2, respective changes in Annex A Change History and History
July 2019	0.2.2	ETSI Technical Officer review before EditHelp registration for Publication

History

Document history		
V1.1.1	October 2019	Publication