ETSI TR 103 537 V1.1.1 (2019-09)

TECHNICAL REPORT

SmartM2M;
Plugtests™ preparation on Semantic Interoperability

Reference

DTR/SmartM2M-103537

Keywords

interoperability, IoT, oneM2M, privacy, SAREF, semantic, testing

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

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 and 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.12]: "SmartM2M; Security; Standards Landscape and best practices".
- ETSI TR 103 534 [i.13]: "SmartM2M; Teaching Material: Part 1: IoT Security and SmartM2M; Teaching Material; Part 2: IoT Privacy".
- ETSI TR 103 535 [i.1]: "SmartM2M; Guidelines for using semantic interoperability in the industry".
- ETSI TR 103 536 [i.9]: "SmartM2M; Strategic/technical approach on how to acheive interoperability/interworking of existing IoT Platforms".
- ETSI TR 103 591 [i.14]: "SmartM2M; Privacy study report; Standards Landscape and best practices".

1.2 Scope of the present document

The present document intends to define and prepare the organization of a Plugtests™ event on Semantic Interoperability based on AIOTI High Level Architecture, oneM2M base ontology (linked to ETSI SmartM2M SAREF one) and oneM2M Service Layer information sharing to demonstrate a more practical/industrial use. This work includes test configurations and scenarios as well as guidelines for the test organization and reporting.

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 103 535: "SmartM2M; Guidelines for using semantic interoperability in the industry".

[i.2] "Advancing IoT Platforms Interoperability", IoT European Platforms Initiative (IoT-EPI), River Publishers, 2018.

[i.3] "Semantic Interoperability", AIOTI WG03, Release 2.0, 2015.

[i.4] "Semantic Interoperability as Key to IoT Platform Federation", M. Jacoby, A. Antonic, K. Kreiner, R. Lapacz and J. Pielorz, 2017.

[i.5] ETSI TS 103 264 (V2.1.1): "SmartM2M; Smart Appliances; Reference Ontology and oneM2M Mapping".

[i.6] ETSI TS 118 133: "oneM2M; Interworking Framework (oneM2M TS-0033)".

[i.7] ETSI TS 118 112: "oneM2M; Base Ontology (oneM2M TS-0012)".

[i.8] ETSI TS 118 113: "oneM2M; Interoperability Testing (oneM2M TS-0013)".

[i.9] ETSI TR 103 536: "SmartM2M; Strategic/technical approach on how to achieve interoperability/interworking of existing standardized IoT Platforms".

[i.10] ETSI TS 118 115: "oneM2M; Testing Framework (oneM2M TS-0015)".

[i.11] "High Level Architecture (HLA)", AIOTI WG03, Release 4.0, 2018.

[i.12] ETSI TR 103 533: "SmartM2M; Security; Standards Landscape and best practices".

[i.13] ETSI TR 103 534 (all parts): "SmartM2M; Teaching Material (Part 1: IoT Security and Part 2: IoT Privacy)".

[i.14] ETSI TR 103 591: "SmartM2M; Privacy study report; Standards Landscape and best practices".

[i.15] ETSI TS 118 123: "oneM2M; Home Appliances Information Model and Mapping (oneM2M TS-0023)".

[i.16] ETSI TS 118 121: "oneM2M; oneM2M and AllJoyn® Interworking (oneM2M TS-0021)".

[i.17] ETSI TS 118 114: "oneM2M; LWM2M Interworking (oneM2M TS-0014)".

[i.18] ETSI TS 118 124: "oneM2M; OCF nterworking (oneM2M TS-0024)".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

ontology: formal specification of a system, defining its components as objects with their main concepts, properties, attributes and relationships versus other components (derived from ETSI TS 118 112 [i.7])

semantics: meta-data describing the content and meaning of a data structure that relates it to the real system it describes

semantic interoperability: ability of IoT devices and platforms to exchange data with unambiguous, shared meaning (derived from Wikipedia)

semantic interoperability testing: validating that a data source and sink are compatible and have the same semantics for a specific data structure

semantic interworking: ability of IoT devices and platforms to exchange data by the means of intermediate components responsible for the mapping of data

semantic-unaware platform: IoT platform which does not support semantics

3.2 Symbols

Void.

3.3 Abbreviations

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

AIOTI	Alliance for IoT Innovation
BO	Base Ontology
CFG	Configuration
CIM	Core Information Model
CSE	Common Services Entity
CTI	Centre for Testing and Interoperability
DUL	DOLCE Ultra Lite
EPI	European Platforms Initiative
ERP	Enterprise Resource Planning
ETSI	European Telecommunication Standards Institute
EU	European Union
HMI	Human Machine Interface
HPA	High Pressure Alarm
ICT	Information and Communication Technology
IoT	Internet of Things
IoT-EPI	IoT European Platforms Initiative
IP	Internet Protocol
ISA	International Society of Automation
IT	Information Technology
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
LSP	Large Scale Pilot
LWM2M	Lightweight M2M
M2M	Machine-to-Machine
Mcc	Reference Point for M2M Communication with CSE
OCF	Open Connectivity Foundation
OWL	Web Ontology Language
PT	Pressure Transmitter
PV	Pressure Value
RDF	Resource Description Framework
SAREF	Smart Applications REFerence ontology
SDT	Smart Device Template
SI	Semantic Interoperability
SSN	Semantic Sensor Network
TC	Technical Committee
TR	Technical Report
TS	Technical Specification
V	Vessel
W3C	World Wide Web Consortium
WG	Working Group
WiFi	Wireless Fidelity
WoT	Web of Things
XML	eXtensible Markup Language

4 Semantic Interoperability Plugtests^{TM} 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 a focus on the technical solutions, which may lead to suboptimal or even ineffective systems. In the case of IoT systems, a very large variety of potential stakeholders are involved, each coming with specific - and potentially conflicting - requirements and expectations. Their elicitation requires that the precise definition of roles that can be related to in the analysis of the requirements, of the use cases, etc.

Examples of such roles to be characterized and analysed are:

- System Designer
- System Developer
- System Deployer
- Device Manufacturer
- Interoperability test organizer
- Interoperability test technical expert

More roles can be defined but the present document will focalise on the ones above.

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. An example of such architecture is the AIOTI High Level Architecture, described in [i.11].

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.

4.2 Main objectives of the present document

As part of its activities towards platforms interoperability, the present document aims at preparing a Plugtests^{TM} event on Semantic Interoperability. For this Plugtests^{TM} event, the interoperability will be based on AIOTI High Level Architecture, oneM2M base ontology (linked to ETSI SmartM2M SAREF one) and oneM2M Service Layer information sharing, with the objective to demonstrate a more practical/industrial use. The present document will include test requirements, configurations and test descriptions in preparation of the event. This work is expected to be developed in close collaboration with the ETSI Centre for Testing and Interoperability (CTI) and will deliver examples of test scenarios and testing organization.

4.3 Purpose and target group

The purpose of the present document is described in clause 1.2.

The target group of readers for the present document is described in clause 4.1.2.2, "Roles".

4.4 Content of the document

The first part of the present document intends to identify the testing requirements from the semantic interoperability standards, especially those collected in ETSI TR 103 535 [i.1] and ETSI TR 103 536 [i.9].

In a next step, the present document focuses on the test configurations and additional elements involved such as components, protocols, data models when appropriate.

Then, the present document defines a set of related interoperability test scenarios based on results in these Technical Reports, but also use case documents from AIOTI, oneM2M, SmartM2M, W3C, etc. Scenarios showing interworking of semantic-unaware systems with systems supporting semantic interoperability are included as well. The scenarios are described from a user point of view, following the ETSI methodology as defined in ETSI Testing Framework [i.10].

Each scenario description clarifies the different actors involved in the test, the pre-conditions, trigger, main and alternative operational flows, as well as post-conditions and test sequence.

Finally, the present document identifies and describes the event preparation requirements like infrastructure, IT and related tools. In this step, it provides guidelines/cook-book on requirements for anonymous reporting of the Plugtests™ outcomes and results.

The organization (logistics/administration), detailed test description and the conduction of the event including the support to participants, are outside the scope of the present document.

5 Requirements for testing semantic interoperability

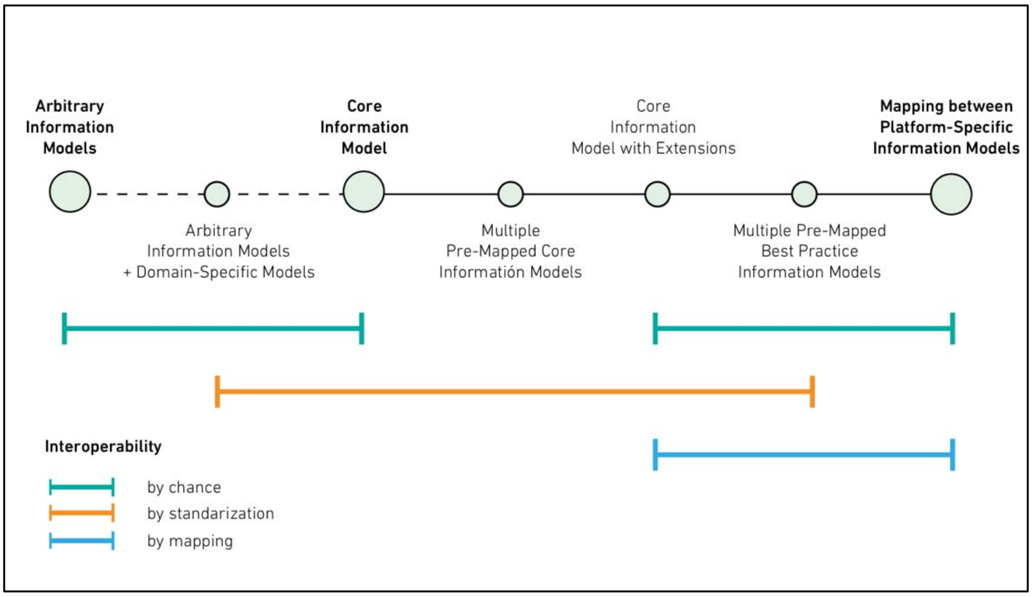
5.1 Approaches for Semantic Interoperability (SI)

5.1.1 Possible approaches

The main expectation of semantic interoperability is to provide an unambiguous meaning of what the "things" are that two (or more) platforms may share and agree upon, thus bridging the potential semantic gap coming from different description 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.2]) 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 should 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).


Arbitrary Information Models
Core Information Model
Core Information Model with Extensions
Mapping between Platform-Specific Information Models
Arbitrary Information Models + Domain-Specific Models
Multiple Pre-Mapped Core Information Models
Multiple Pre-Mapped Best Practice Information Models
Interoperability
by chance
by standarization
by mapping

NOTE: Source: IoT-EPI Task Force, [i.2], based on [i.4].

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). A similar approach would apply for the case of multiple ontologies, as described in clause 6.

More information on and examples of these approaches can be found in the companion ETSI TR 103 535 [i.1]. Some are also described in clause 5.1.3 of the present document.

5.1.2 Commonalities and differences between SI approaches

The most common way to achieve semantic interoperability is via "ontologies" that are an explicit specification of a shared "understanding" that can be processed automatically by machines. Recent standardization efforts have produced a number of IoT-specific ontologies, such as SAREF, oneM2M Base Ontology (BO), SSN Ontology and others (see the AIOTI WG03 analysis in [i.3]).

The IoT ontologies will in general offer different perspectives on (parts of) the IoT system and describe a way to model the central part of an IoT system. However, standardized IoT ontologies may result from different approaches: high-level abstraction (e.g. oneM2M BO), deep taxonomies (e.g. SSN that extends a top-level ontology DUL), or deployment orientation (e.g. Open-IoT weather station model).

IoT ontologies often need to be extended (e.g. Core ontologies) or customized before being used in a concrete application thus creating the need for careful validation of different implementations which is the purpose of the Plugtests^{TM}.

5.1.3 Examples of different approaches

5.1.3.1 SAREF

The Smart Appliances/Applications REFerence ontology (SAREF) is the result of an EU initiative launched in 2013 with the support of ETSI in order to create a shared semantic model based on consensus to enable the missing interoperability among smart appliances. SAREF can be considered as an addition to existing communication protocols to enable the translation of information coming from existing (and future) protocols to and from all other protocols that are referenced to SAREF. For example, a home gateway enriched by SAREF can associate devices in a home with each other and with different service providers.

The initial focus was on the optimization of energy management in smart buildings. The first resulting semantic model - SAREF - was standardized by ETSI in November 2015 (ETSI TS 103 264 [i.5]). SAREF is a first ontology standard in the IoT ecosystem and sets a template and a base for the development of similar standards for other verticals.

Since its first release, SAREF continues to evolve systematically into a modular network of standardized semantic models, with additional extensions such as SAREF for Energy, SAREF for Environment and SAREF for Buildings. More work is on-going in a number of other domains such as Smart Cities, Smart AgriFood, Smart Industry and Manufacturing, Automotive, eHealth/Ageing-well and Wearables. The objective is to make SAREF a "Smart Application REFerence ontology", which enables better integration of semantic data from various vertical domains.

5.1.3.2 oneM2M semantic interoperability approaches

The oneM2M standard supports different approaches for semantic interoperability requiring a before agreement between applications and devices to share data between them (see [i.6]).

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.
See: "oneM2M TS-0012 oneM2M Base Ontology" (ETSI TS 118 112 [i.7]).

2) Common vocabulary (basic serialization format XML or JSON): Smart Device Template (SDT) for the home domain.
See: "oneM2M TS-0023 Home Appliances Information Model and Mapping" (ETSI TS 118 123 [i.15]).

3) Resources specializations: oneM2M FlexContainer resources specialized with a technology-specific data model
See: "oneM2M TS-0021 oneM2M and AllJoyn Interworking" (ETSI TS 118 121 [i.16]).

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.
See: "oneM2M TS-0014 LWM2M Interworking" (ETSI TS 118 114 [i.17]) and "oneM2M TS-0024 OCF Interworking" (ETSI TS 118 124 [i.18]).

A work item called "oneM2M WI-0056 Evolution of Proximal IoT Interworking" has been defined to provide an harmonization of the work done for interworking between oneM2M and specific proximal IoT technologies, such as AllJoyn®, LWM2M and OCF. The idea is to enable interworking with external "proximal" IoT technologies without the need for oneM2M applications to be aware of the details of device specific technology.

5.1.3.3 W3C Web of Things

The approach of W3C for the Web of Things (WoT) is to focus on the role of Web technologies as a basis for services spanning IoT platforms ranging from microcontrollers to cloud-based server farms. In this context of or a platform of platforms, shared semantics are essential for discovery, interoperability, scaling and layering on top of standardized protocols and existing platforms with metadata classified into things, security and communications.

Things are considered to be virtual representations (objects) for physical or abstract entities. They are having events, properties and actions as a basis for easy application scripting. A clean separation between the application and transport layers simplifies scripting by decoupling the details of protocols and message formats, allowing servers to use the protocols that best fit the particular context. Communications metadata allows servers to identify how to communicate with other servers.

Thing descriptions are expressed in terms of W3C's Resource Description Framework (RDF). This includes the semantics for what kind of thing it is, and the data models for its events, properties and actions. The underlying protocols are free to use whatever communication patterns are appropriate to the context according to the constraints set by the given metadata.

5.2 Features for semantic interoperability testing

This clause identifies the main features that could be relevant for an interoperability test.

The features described below apply to any type of implementation in an IoT node: server, gateway, device or application. The features that can be tested are divided into two categories:

- Ontology management, which includes the handling of the ontology by the node:
  - Acquisition and storage of the tested ontology by the node, which can be static, e.g. loading a file, or dynamic, through discovery and learning of the ontology identification.
  - Instantiation of the ontology mapped to the data structure or resource tree of the node.
  - Update of the ontology in the node.
- Data management, which includes the usage of the ontology by the node:
  - Ability of the implementation to generate a request referring to the ontology.
  - Ability of the implementation to understand a request referring to the ontology.
  - Ability of the implementation to understand a gap of mapping in the ontology when receiving a request.
  - Ability of the implementation to generate a response referring to the ontology.

NOTE: Even though it is theoretically possible, the dynamic treatment of semantics by the platform is currently not recommended given the level of the technology and the scalability issues this feature would trigger in a real case. The interworking between different ontologies should be considered at the border of a system rather than internally to a system.

5.3 Objective of a semantic interoperability Plugtests™ event

Semantic interoperability Plugtests™ aim at testing the capability of IoT platforms, devices and applications to exchange data with unambiguous and shared meaning paving the way for machine computable logic, inferencing, knowledge discovery and data federation.

As described in the AIOTI semantic interoperability report [i.3], interoperability involves the following capabilities:

- Exchange of meaningful, actionable information between two or more systems across organizational boundaries.
- A shared understanding of the exchanged information using a common ontology between all interacting entities or by introducing a mapping in case of different ontologies.
- An agreed expectation for the request and the response to the information exchange.

Applying these capabilities to semantics operations can be considered as requirements for running semantic interoperability tests.

6 Test configurations

6.1 Introduction

Test configurations illustrate the interacting entities covering the different test scenarios [i.8].

For the sake of clarity, the following points should be noted:

- Intermediate interworking proxy entities are not illustrated in the test configurations: An IoT device/application could interact with the IoT platform directly or through an interworking proxy, according to the objective of the test scenario.
- Ontologies could be defined using advanced conceptualizations like SAREF, oneM2M base ontology, etc. or lightweight serializations like Smart Device Template (SDT), NGSI-LD, etc.

6.2 Single IoT platform

6.2.1 CFG-01 Single IoT device/Application on a single IoT platform

This test configuration covers the following examples:

- A sensor publishing data on an IoT platform.
- An actuator executing commands received from an IoT platform.

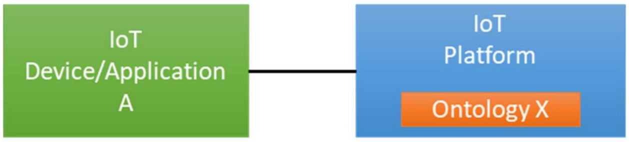


Figure 2: Configuration CFG-01 - Single IoT device/Application on a single IoT platform

6.2.2 CFG-02 Two IoT devices/Applications on a single IoT platform

This test configuration covers the following examples:

- An IoT application collecting data from a sensor both registered on the same platform.

- An IoT application controlling an actuator both registered on the same platform.
- Interaction between two IoT devices both registered on the same platform.
- Interaction between two IoT applications both registered on the same platform.

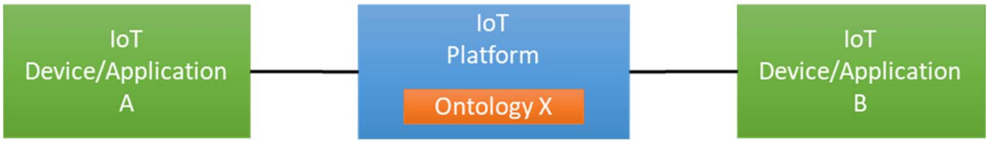


Figure 3: Configuration CFG-02 - Two IoT devices/Applications on a single IoT platform

6.3 Multiple IoT platforms using the same ontology

6.3.1 CFG-03 Single IoT devices/Applications on multiple IoT platforms

This test configuration supports multi-hop using N platforms and covers the following examples:

- A sensor publishing data on IoT Platform B passing by IoT Platform A using the same ontology.
- An actuator executing commands received from the Platform B passing by IoT platform A using the same ontology.

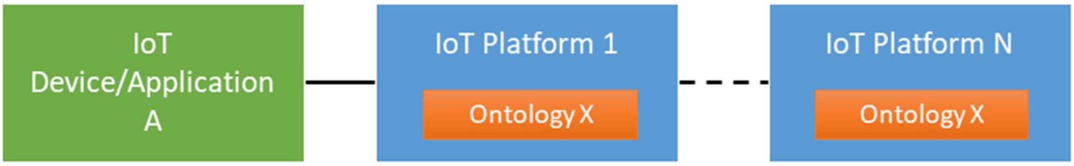


Figure 4: Configuration CFG-03 - Single IoT devices/Applications on multiple IoT platforms

6.3.2 CFG-04 Two IoT devices/Applications on multiple IoT platforms using a common ontology

This test configuration supports multi-hop using N platforms and covers the following examples:

- An IoT application collecting data from a sensor each one registered to a different IoT platform but using the same ontology.
- An IoT application controlling an actuator each one registered in a different IoT platform but using the same ontology.
- Interaction between two IoT devices each one registered in a different IoT platform but using the same ontology.

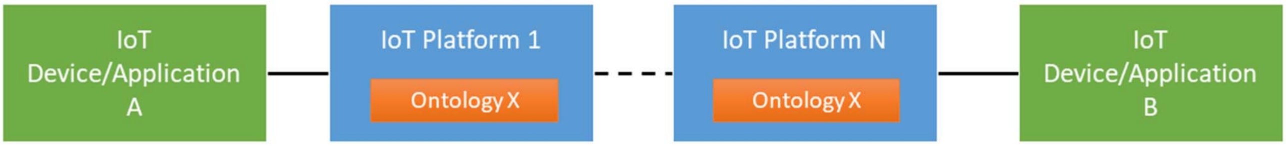


Figure 5: Configuration CFG-04 - Two IoT devices/Applications on multiple IoT platforms using a common ontology

6.4 Multiple IoT platforms using different ontologies

6.4.1 CFG-05 Single IoT devices/Applications on multiple IoT platforms using different ontologies

This test configuration supports multi-hop using N platforms and covers the following examples:

- A sensor publishing data on IoT Platform B passing by IoT Platform A, when both platforms are using different ontologies.
- An actuator executing commands received from the Platform B passing by IoT Platform A using different ontologies.

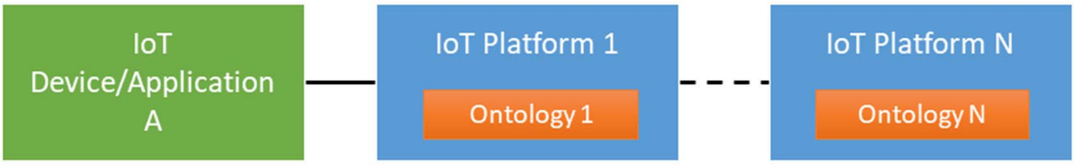


Figure 6: Configuration CFG-05 - Single IoT devices/Applications on multiple IoT platforms using different ontologies

6.4.2 CFG-06 Multiple IoT devices/Applications on multiple IoT platforms using different ontologies

This test configuration supports multi-hop using N platforms and covers the following examples:

- An IoT application collecting data from a sensor each one registered to a different IoT platform but using different ontologies.
- An IoT application controlling an actuator each one registered in a different IoT platform but using different ontologies.
- Interaction between two IoT devices each one registered in a different IoT platform but using different ontologies.

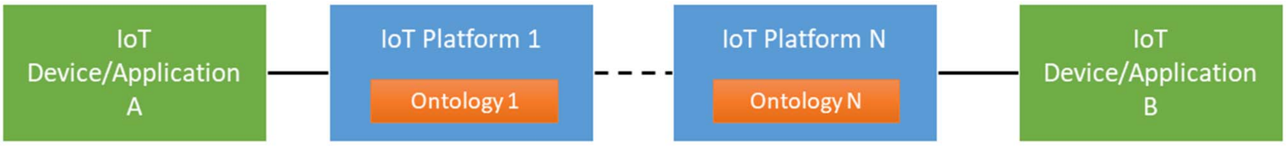


Figure 7: Configuration CFG-06 - Multiple IoT devices/Applications on multiple IoT platforms using different ontologies

7 Examples of possible testing scenarios

7.1 Introduction

This clause is based on the results of ETSI TR 103 535 [i.1] and ETSI TR 103 536 [i.9], but also use case documents from AIOTI, oneM2M, ETSI TC SmartM2M, W3C, etc. It describes possible scenarios that would demonstrate the features and requirements to be tested during a Plugtests™ event. These scenarios foster on the capability for cross-platform and cross-domain exchange of data. They also show interworking of semantic-unaware systems with systems supporting semantic interoperability.

Accordingly, the next sections in clause 7 present a set of generic testing scenarios for interested parties willing to validate semantic interoperability. These testing scenarios are generic, they avoid to focus on a specific ontology, like oneM2M and SAREF. They cover both inter and intra platforms interoperability, where platform can be any type of node: server, gateway, device. In the oneM2M architecture, inter platforms interoperability would mean testing the Mcc' reference point. They organized according to their high-level objective, from the simplest which aims to validate interworking between one device and one platform, both being enabled with the same semantics, up to the most complicated which aims at demonstrating interoperability between devices connected to platforms that use different semantics, possibly in different vertical domains.

Table 1 lists the possible scenarios which have been identified to test semantic interoperability, ordered according to the complexity of their high-level objective. The following sets of scenarios with progressivity of their high-level objectives are described:

A) Single platform semantic interoperability in the same vertical domain: devices exchange data and commands with a single platform, all these entities are enabled with the same ontology.

B) Cross-platform semantic interoperability in the same vertical domain: the previous set is enriched with scenarios where the data exchange involves at least two platforms, both operating in the same vertical domain, and with all entities enabled with the same ontology.

C) Cross-vertical domain interoperability: devices/applications exchange data and commands through at least two platforms, each operating in a different vertical domain, with all the entities enabled with the same semantics (for example SAREF).

D) Interworking with semantic-unaware systems: semantic-unaware platforms and/or devices exchange data with devices/applications registered with a semantic-enabled platform and using the same semantics as the platform;

E) Heterogeneous semantics interoperability (different ontologies): devices/applications exchange data and commands through at least two platforms, with all the entities registered at each platform enabled with the same semantics, but the platforms use different ontologies (for example SAREF for one platform and SSN for the other one).

Table 1: Summary of possible scenarios and their objectives

High level objective	Name of the scenario	Configuration	Scenario Objective
A - Single platform semantic interoperability in the same vertical domain.	A-1 - sensor data reporting on a single IoT platform using an ontology.	An IoT device and a single platform (CFG-01).	Validate the correct flow of data from an IoT device to an IoT platform using an ontology.
	A-2: data exchanged between two IoT devices on a single IoT platform using a common ontology.	Two IoT devices and a single platform (CFG-02).	Validate the correct flow of a data between two IoT devise registered on the same IoT platform and using an ontology.
B - Cross-platform semantic interoperability in the same vertical domain.	B-1: sensor data reporting between platforms in the same vertical domain using the same ontology	An IoT device, Platform A, Platform B and an IoT application (CFG-03/CFG-04).	This scenario allows to validate the correct flow of data from an IoT device to an IoT application (upstream direction), when all entities involved use the same semantics.

High level objective	Name of the scenario	Configuration	Scenario Objective
	B-2: command sent by an IoT application through platforms in the same vertical domain using the same ontology.	An IoT application, platform A, Platform B and an IoT device (CFG-03/CFG-04).	This scenario allows to validate the correct flow of a command from an IoT application to an IoT device (downstream direction), when all entities involved use the same semantics.
C - Cross-vertical domain interoperability.	C-1: sensor data reporting between platforms across different domains. The two platforms use the same ontology.	A platform in vertical Domain A, another platform in vertical Domain B, an IoT device and an IoT application (CFG-04).	This scenario allows to validate the correct flow of data from an IoT device to an IoT application (upstream direction) across different domains. The two platforms use the same ontology.
D - Interworking with semantic-unaware systems.	D-1: sensor data reporting between platforms. One of the platforms is semantic-unaware, the other one is semantic-aware.	An IoT device registered to a (semantic-unaware-) Platform A An IoT application registered to a Platform B, which includes an interworking entity for transforming data from Platform A and make it semantically manageable (CFG-06).	This scenario allows to validate the correct flow of data from an IoT device to an IoT application (upstream direction) across different domains. One of the platforms is semantically-unaware, so the current flow of information involves also data transformation and linking in order to make it suitable for management by the semantic-aware platform.
	D-2: Semantic interworking between legacy IoT devices in the same platform using the same ontology.	Two IoT devices and a single IoT platform (Two interworking proxies are required to perform data mapping) (CFG-02).	Validate the correct flow of a data between two IoT devices on the same IoT platform and using an ontology. The communication between the IoT devices and the IoT platform is performed through interworking proxies.
E - Heterogeneous semantics interoperability (different ontologies).	E-1: Sensor data reporting between platforms in the same vertical domain using different ontologies.	An IoT entity (device/application) registered to Platform A that uses a different ontology than Platform B An IoT entity (device/application) registered to Platform B (CFG-05/CFG-06).	This scenario allows to validate the correct flow of data from the IoT device up to the IoT application (upstream direction), when all entities use the same semantics as the platform they are registered with, but both platforms use different ontologies. This scenario requires an entity to perform the mapping between both ontologies, or both are mapped to a unified ontology.
	E-2: sensor data reporting between platforms across different domains. The two platforms use different ontologies.	A platform in vertical Domain A, another platform in vertical Domain B, an IoT device and an IoT application (CFG-06).	This scenario allows to validate the correct flow of data from an IoT device to an IoT application (upstream direction) across different domains. The two platforms use different ontologies.

7.2 Scenario A-1: Sensor data reporting on a single IoT platform using an ontology

7.2.1 Test configuration

CFG-01: Single IoT device/Application on a single IoT platform.

7.2.2 Scenario high level objective

A - Single platform semantic interoperability in the same vertical domain.

7.2.3 Description

The objective of this scenario is to validate the correct operation between an IoT device and an IoT semantic platform using the same ontology in the same domain.

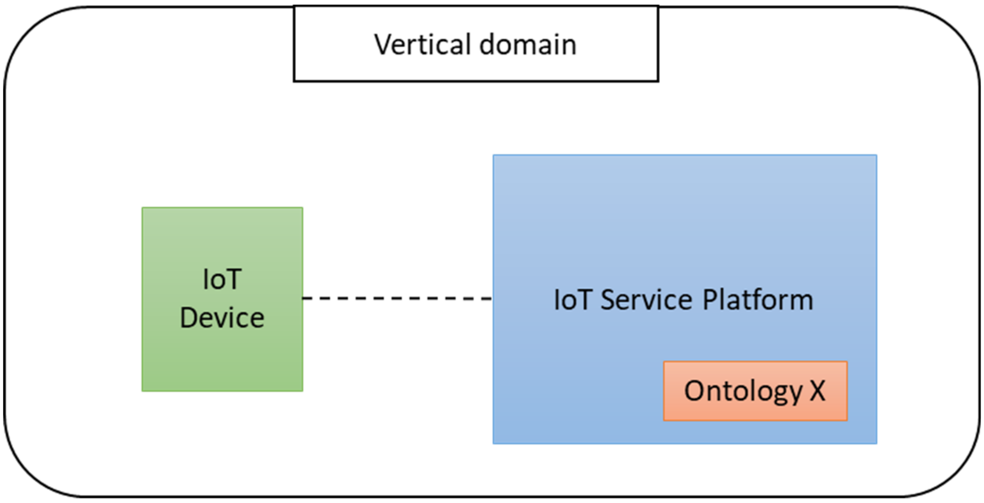


Figure 8: Semantic interworking between two IoT devices in the same domain

The scenario aims at validating that an IoT device is able to interact with an IoT platform using an ontology in the same vertical domain. For example, the IoT device publish data on the semantic IoT platform or execute a command received form the IoT platform.

Possible instantiation: luminosity sensor + oneM2M platform + ontology SAREF.

7.2.4 Actors/Entities involved

The scenario involves the following actors:

- IoT device: a sensor measuring data.
- IoT platform: including the ontology X.

7.2.5 Scenario sequence/flows

Pre-conditions:

- The IoT platform in operating status.
- The IoT device is registered under the IoT platform.

Step 1: Acquisition, storage and instantiation of the common ontology by the IoT platform

- Objective: the IoT platform discovers (if relevant) and obtains ontology X.
- Validation: ontology is successfully loaded in each of the platforms.

Step 2: Update of the common ontology in the IoT platform

- Objective: the IoT platform discovers (if relevant) and update the ontology X.
- Validation: ontology X is successfully loaded in each of the platforms.

Step 3: Ability of the IoT device to generate and send a request to the IoT platform referring to the common ontology (upstream)

- Objective: the IoT device publish a data on the IoT platform referring to the common ontology.
- Validation: the data issued from the IoT device to the IoT platform is correctly formed.

Step 4: Ability of the IoT device to receive a response from the IoT platform referring to the common ontology (downstream)

- Objective: the IoT device receives the request from the IoT platform using the common ontology and executes the command.
- Validation: the IoT device has successfully executed the command.

7.3 Scenario A-2: Data exchanged between two IoT devices on a single IoT platform using a common ontology

7.3.1 Test configuration

CFG-02: Two IoT devices/Applications on a single IoT platform.

7.3.2 Scenario high level objective

Two IoT devices and a single platform.

7.3.3 Description

The objective of this scenario is to validate the correct operation between two IoT devices using the same ontology in the same domain.

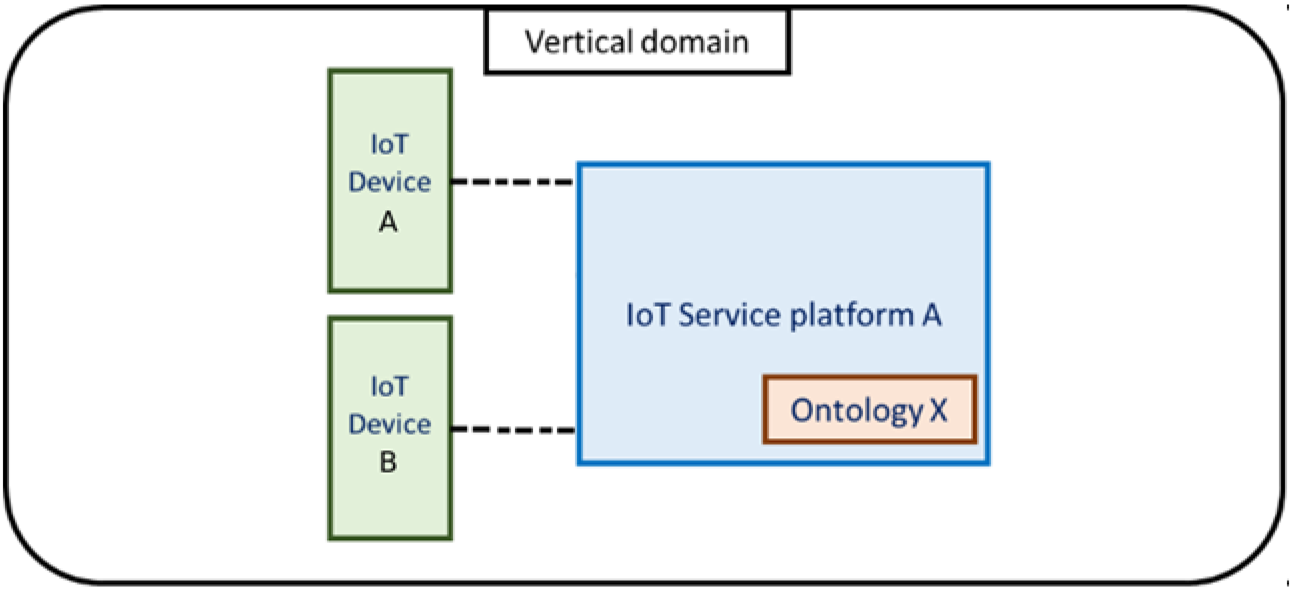


Figure 9: Semantic interoperability between two IoT devices in the same domain

The scenario aims at validating that two IoT devices are able to exchange data between each other through both operating under the same platform and using the same ontology in the same vertical domain.

Possible instantiation: For example, the status of lamp from vendor A is updated according to a luminosity sensor from vendor B.

7.3.4 Actors/Entities involved

The scenario involves the following actors:

- IoT device A: a sensor measuring data.
- IoT device B: an actuator controlling a system.
- IoT platform: including the ontology X describing the semantic used by IoT device A and IoT device B.

7.3.5 Scenario sequence/flows

Pre-conditions:

- The IoT platform in operating status.
- The IoT device A is registered under the IoT platform.
- The IoT device B is registered under the IoT platform.

Step 1: Acquisition, storage and instantiation of the common ontology by the IoT platform

- Objective: the IoT platform to discover (if relevant) and obtain the common ontology.
- Validation: the ontology is successfully loaded in the IoT platform.

Step 2: Update of the ontology in the IoT platform

- Objective: the IoT platform to discover (if relevant) and update the common ontology.
- Validation: the ontology is successfully loaded in the IoT platform.

Step 3: Ability of IoT platform to receive a request from IoT device A

- Objective: IoT device A send a request to the IoT platform referring to the common ontology.
- Validation: the request issued from IoT device A to the IoT platform is correctly formed.

Step 4: Ability of IoT platform to send a response back to IoT device A

- Objective: IoT platform sends a response back to IoT device A referring to the common ontology.
- Validation: the response sent from the IoT platform to IoT device A is correctly formed.

Step 5: Ability of IoT device B to receive a request from the IoT platform

- Objective: IoT device B receives a request from the IoT platform referring to the common ontology.
- Validation: the request issued by the IoT Platform to IoT device B is correctly formed.

Step 6: Ability of IoT device B to send a response back to the IoT platform

- Objective: IoT device B sends a response to the IoT platform referring to the common ontology.
- Validation: the response issued by IoT device B to the IoT platform is correctly formed.

7.4 Scenario B-1: sensor data reporting between platforms in the same vertical domain using the same ontology

7.4.1 Test configuration

CFG-04: Two IoT devices/Applications on multiple IoT platforms using a common ontology.

NOTE: This scenario can also be run with configuration CFG-03 (Single IoT devices/Applications on multiple IoT platforms), where data is stored on Platform B.

7.4.2 Scenario high level objective

Cross-platform semantic interoperability in the same domain.

7.4.3 Description

Objective: The objective of this scenario is to validate the correct operation between two platforms using the same ontology in the upstream direction.

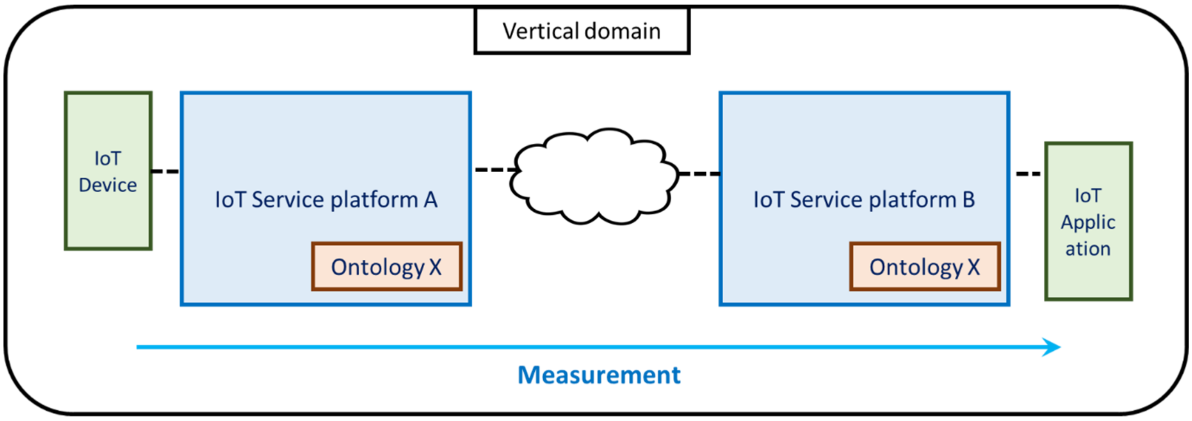


Figure 10: Cross-platform semantic interoperability in the same domain - upstream direction

The scenario aims at validating that an application operating in an IoT platform is able to retrieve and process the meaning of a measurement performed by a sensor operating under another IoT platform, but while both of them use the same ontology and are operating in the same vertical domain. For example, the status of a light in Platform A is retrieved by a smartphone application running Platform B.

Possible instantiation: Temperature sensor + oneM2M platform + ontology SAREF + smartphone application.

7.4.4 Actors/Entities involved

The scenario involves the following actors:

- IoT device: making the measurement.
- IoT Platform A: including the ontology X, which describes the semantic used by the IoT device. The IoT device is operating under this IoT platform.
- IoT Platform B: including the same ontology.
- IoT application: retrieving the measurement from the IoT device and the ontology from the IoT platform.

7.4.5 Scenario sequence/flows

Pre-conditions:

- Both platforms are in operating status.
- Both platforms can communicate with one another and exchange data.
- The IoT device is registered under Platform A.
- The application is registered under Platform B.

Step 1: Acquisition and storage of the tested ontology by the Platform A [Platform B]/Instantiation of the ontology mapped to the data structure of the Platform A [Platform B]

- Objective: each platform discovers (if relevant) and obtains the common ontology.
- Validation: the ontology is successfully loaded in each of the platforms.

Step 2: Update of the ontology in the Platform A [Platform B]

- Objective: each platform discovers (if relevant) and update the common ontology.
- Validation: the ontology is successfully loaded in each of the platforms.

Step 3: Ability of the Platform B to generate a request referring to the ontology

- Objective: the IoT application requests the status of the IoT device through Platform B, referring to the common ontology.
- Validation: the request issued from Platform B to Platform A is correctly formed.

Step 4: Ability of the Platform A to understand a request referring to the ontology

- Objective: the Platform A receives the request from Platform B and uses the common ontology to retrieve the requested data from the IoT device.
- Validation: Platform A has successfully retrieved the requested data from the IoT device.

Step 4a: Ability of the Platform A to understand a gap of mapping in the ontology when receiving a request

- Objective: the platform A receives the request from Platform B and uses the common ontology to retrieve the requested data from the IoT device. The ontology describing the IoT device does not match the common ontology.
- Validation: Platform A has successfully reported an error when receiving the request from Platform B.

Step 5: Ability of the implementation to generate a response referring to the ontology

- Objective: the Platform A uses the common ontology to retrieve the requested data from the IoT device and sends a response to Platform B. The response is provided to the IoT application.
- Validation: The IoT application has obtained successfully a valid data from the IoT device (this may be displayed to the testing person).

7.5 Scenario B-2: command sent by an IoT application through platforms in the same vertical domain using the same ontology

7.5.1 Test configuration

CFG-04: Two IoT devices/Applications on multiple IoT platforms using a common ontology.

NOTE: This scenario can also be run with configuration CFG-03 (Single IoT devices/Applications on multiple IoT platforms), where data is pushed towards the IoT device (for example for device management).

7.5.2 Scenario high level objective

Cross-platform semantic interoperability in the same domain.

7.5.3 Description

Objective: The objective of this scenario is to validate the correct operation between two platforms using the same ontology in the downstream direction.

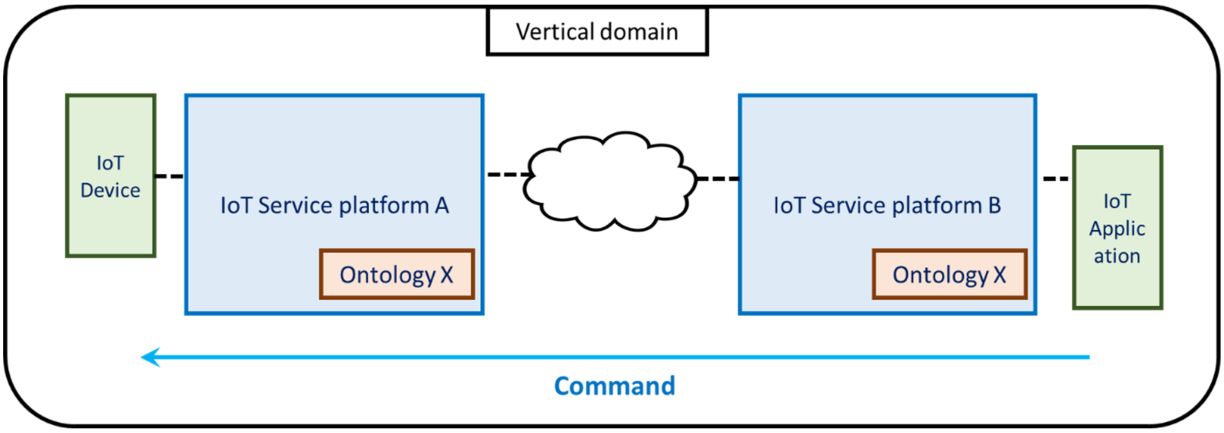


Figure 11: Cross-platform semantic interoperability in the same domain - downstream direction

The scenario aims at validating that an application operating in an IoT platform is able to send a command to an IoT actuator device operating under another IoT platform, but while both of them use the same ontology and are operating in the same vertical domain. This scenario demonstrates the same capability as Scenario B-1, but in the reverse direction.

Possible instantiation: A smartphone application running on a platform enabled with SSN can trigger the switch-on/switch off of a heater in Platform A also enabled with SSN.

7.5.4 Actors/Entities involved

The scenario involves the following actors:

- IoT application: sending a command to the IoT device and retrieving the ontology from the IoT Platform B.
- IoT Platform A: including the ontology X, which describes the semantic used by the IoT device The IoT device is operating under this IoT platform.
- IoT Platform B: including the same ontology and serving the IoT application.
- IoT actuator device: executing the action received from the IoT application.

7.5.5 Scenario sequence/flows

Pre-conditions:

- Both platforms are in operating status.
- Both platforms can communicate with one another and exchange data.
- The IoT device is registered under Platform A.
- The application is registered under Platform B.

Step 1: Acquisition and storage of the tested ontology by the Platform A [Platform B]/Instantiation of the ontology mapped to the data structure of the Platform A [Platform B]

- Objective: each platform discovers (if relevant) and obtains the common ontology.
- Validation: the ontology is successfully loaded in each of the platforms.

Step 2: Update of the ontology in the Platform A [Platform B]

- Objective: each platform discovers (if relevant) and updates the common ontology.
- Validation: the ontology is successfully loaded in each of the platforms.

Step 3: Ability of the Platform B to generate a request referring to the ontology

- Objective: the IoT application sends a command to the IoT device through Platform B, referring to the common ontology.
- Validation: the request issued from Platform B to Platform A is correctly formed.

Step 4: Ability of the Platform A to understand a request referring to the ontology

- Objective: the Platform A receives the request from Platform B and uses the common ontology to forward the command to the IoT device which executes the command.
- Validation: the IoT device has successfully executed the command (this may be displayed to the testing person).

Step 4a: Ability of the Platform A to understand a gap of mapping in the ontology when receiving a request

- Objective: the Platform A receives the request from Platform B and uses the common ontology to forward the command to the IoT device. The ontology describing the IoT device does not match the common ontology.
- Validation: Platform A has successfully reported an error when receiving the request from Platform B.

7.6 Scenario C-1: Cross-Domain Interoperability, Same Ontology

7.6.1 Test configuration

CFG-04: Two IoT devices/Applications on multiple IoT platforms using a common ontology.

7.6.2 Scenario high level objective

Cross-platform heterogeneous semantic interoperability across different domains.

7.6.3 Description

Objective: The objective of this scenario is to validate the correct operation (in the upstream direction) between two platforms applied to different domains in the upstream direction. The two platforms use the same ontology. This scenario requires that a linking between the two domains is in place. It is assumed that this linking or mapping takes place within Platform B (see Figure 12).

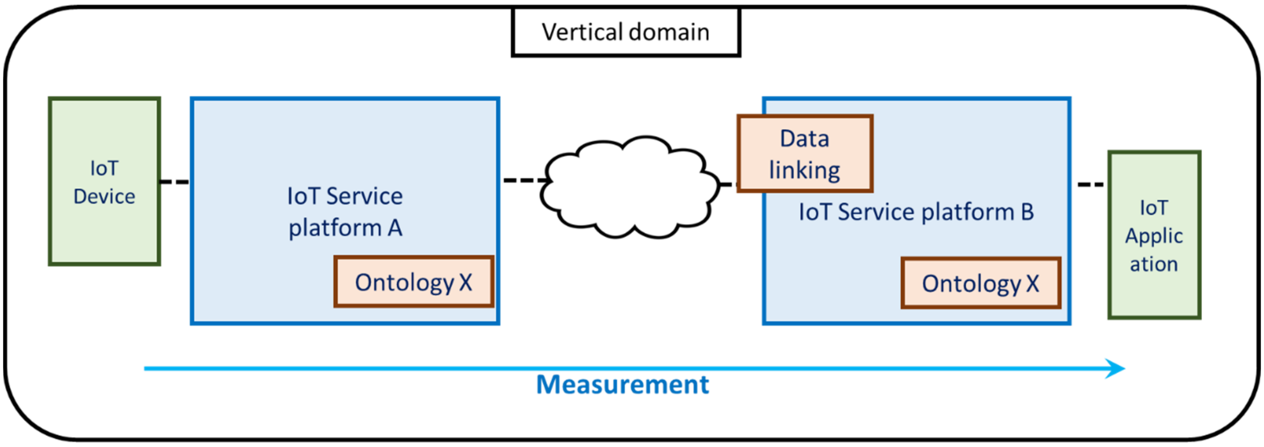


Figure 12: Cross-platform semantic interoperability applied to different domains within the same ontology- upstream direction

The scenario aims at validating that an application operating in an IoT platform (Platform B) is able to retrieve and process the effect on its vertical domain of a measurement performed by a sensor operating under another IoT platform (Platform A).

Possible instantiation:

- Platform A is a generic data collection platform, in which an IoT input device sensor S is capable of measuring the rotating speed of a shaft, exposing it as property "rotatingSpeed".
- Platform B is a process monitoring platform in which an entity Rotating Compressor C is defined, representing the status of a compressor described by ontologyX.
- Both platforms are enabled with the ontology X: SAREF.
- Among the features of compressor C is the current value of the shaft rotating speed, "compressorSpeed".
- The status of compressorSpeed changes according to the current value of the measurement rotatingSpeed obtained through Platform A.

7.6.4 Actors/Entities involved

The scenario involves the following actors:

- IoT device PT: making the measurement.
- IoT Platform A: including the ontology X, which describes the semantic used by the IoT device. The IoT device PT is operating under this IoT platform.
- IoT Platform B: including the same ontology X, which describes the semantic used by the IoT application representing vessel V. The platform includes an entity able to perform the mapping between features of PT (in particular, the value of the current measurement) and features of V (in particular PV, the current value of vessel pressure).
- IoT application: retrieving the measurement from the IoT device PT form Platform A and updating features PV and HPA in Platform B according to ontology X.

7.6.5 Scenario sequence/flows

Pre-conditions:

- Both platforms are in operating status.
- Both platforms can communicate with one another and exchange data.
- The IoT device PT is registered under Platform A.

- The application representing V is registered under Platform B.
- Platform B runs an entity able to perform the linking between data obtained from platform A and data managed internally by Platform B.

Step 1: Acquisition and storage of the tested ontology by the Platform A [Platform B]/Instantiation of the ontology mapped to the data structure of the Platform A [Platform B]

- Objective: each platform discovers (if relevant) and obtains its own ontology.
- Validation: the ontology is successfully loaded in each of the platforms.

Step 2: Update of the ontology in Platform A [Platform B]

- Objective: each platform discovers (if relevant) and updates its own ontology.
- Validation: the ontology is successfully loaded in each of the platforms.

Step 3: Execution of the ontology mapping by Platform B

- Objective: Platform B starts the components responsible to perform the linking between elements from platform A to elements from platform B.
- Validation: the mapping component is successfully loaded and running in Platform B.

Step 4: Ability of the Platform B to generate a request referring to the ontology

- Objective: the IoT application V requests the status of the IoT device PT through Platform B, referring to ontology X.
- Validation: the request issued from Platform B to Platform A is correctly formed.

Step 5: Ability of the Platform B to generate a well-formed request for Platform A

- Objective: the linking entity translates the request of the status of the IoT device from Platform B into a request for platform A, referred to the correct element (the IoT sensor device) registered in platform A.
- Validation: the translated request in Platform B is correctly formed and refers to the right element.

Step 6: Ability of the Platform A to understand a request referring to the ontology

- Objective: Platform A processes the request from Platform B and uses the ontology X to retrieve the requested data from the IoT device PT.
- Validation: Platform A has successfully retrieved the requested data from the IoT device.

Step 6a: Ability of the Platform A to understand a gap of mapping between domains when receiving a request

- Objective: Platform A processes the request from Platform B and uses the ontology X to retrieve the requested data from the IoT device. The request after the mapping does not match the requirements of platform A, which may be the result of a mapping gap or error.

  The mismatch may be:

  - Request is formally invalid (e.g. malformed).
  - Request is invalid with reference to the selected device (e.g. the device does not exist, or it does not support the requested attributes).

- Validation: Platform A has successfully reported an error when processing the request from Platform B.

Step 7: Ability of the implementation to generate a response referring to the ontology

- Objective: Platform A uses ontology X to retrieve the requested data from the IoT device PT; the response is provided to Platform B.

- Platform B receives the response from Platform A and uses the linking service to handle the routing of data in the response to the internal data structure managed by Platform B.
- Dhe data is passed to application V.
- Validation: Application V has received the data requested, and features PV and HPA are updated successfully (this may be displayed to the testing person).

7.7 Scenario D-1: Interworking with Semantic-unaware Systems

7.7.1 Test configuration

CFG-06: Multiple IoT devices/Applications on multiple IoT platforms using different ontologies.

7.7.2 Scenario high level objective

Interworking with semantic-unaware systems. Implies data transformation and mapping in order to make it semantically tractable.

7.7.3 Description

Objective: The objective of this scenario is to validate the correct operation (in the upstream direction) between Platform A (semantic-unaware) and Platform B (semantic-aware).

Semantic-aware Platform B, having ontology Y, obtains data from Platform A, which is semantic-unaware (e.g. a Modbus server). Data obtained from Platform A should be mapped to ontology Y before it can be used.

This mapping, together with possible data manipulation and transformation, takes place within Platform B (see Figure 13)

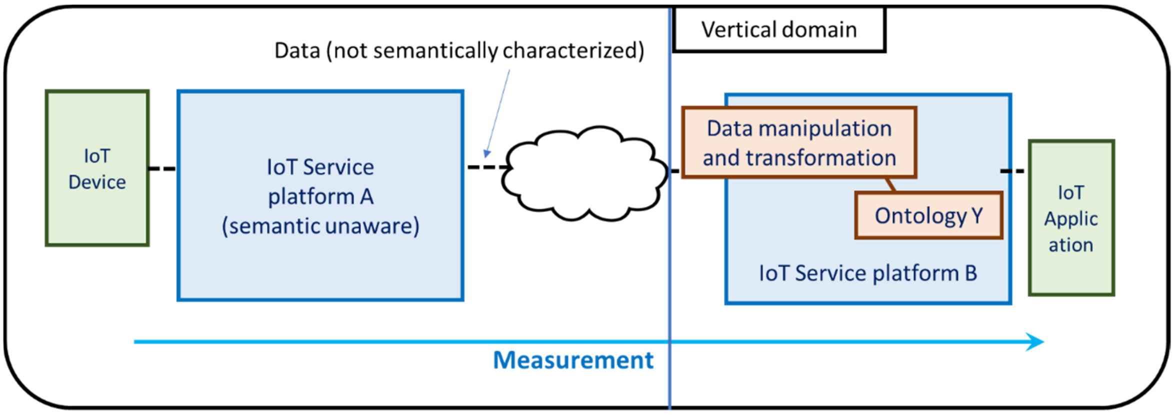

Vertical domain
Data (not semantically characterized)
IoT Device
IoT Service platform A (semantic unaware)
Data manipulation and transformation
Ontology Y
IoT Service platform B
IoT Applic ation
Measurement

Figure 13: Interworking with semantic-unaware platform - upstream direction

The scenario aims at validating that an application operating in an IoT Platform B is able to retrieve and process the effect on its vertical domain of a measurement performed by a sensor operating under another IoT Platform A, which is semantic-unaware.

In order to fix the ideas, the following example assumes that Platform A is a Modbus server.

Possible instantiation:

- Platform A is a generic data collection platform, semantic-unaware, in which an IoT input device PT is capable of measuring a pressure value.
- The value of the physical measurement obtained is transferred into a data container within Platform A (e.g. a Modbus register), thus losing any semantic characterization.

- Platform B is a process monitoring platform in which an entity V is defined, representing the status of a process vessel.
- Among the features of vessel V are:
    a) PV: the current value of pressure in the vessel; and
    b) HPA: a high-pressure alarm linked to the current value of PV.
- Platform B obtains the needed data from Platform A, and data obtained is transformed into another data type, which is semantically characterized and tagged according to ontology Y before being transferred to application V.
- The status of both PV and HPA changes according to the current value of the measurement PT obtained through Platform A. An alarm condition may be raised or reset on Platform B, reflecting the current value of the measurement PT obtained through Platform A.

7.7.4 Actors/Entities involved

The scenario involves the following actors:

- IoT device PT: making the measurement.
- IoT Platform A: not platform aware, as soon as input value form PT is stored within its data structure, it loses any semantic characterization. The IoT device PT is operating under this IoT platform.
- IoT Platform B: including the ontology Y, which describes the semantic used by the IoT application representing vessel V.
- Platform B includes an entity able to create a mapping that adds a semantic layer upon the data structure defined by Platform A, thus allowing data coming from Platform A to be managed according to ontology Y (in particular, the value of the current measurement) and features of V (in particular PV, the current value of vessel pressure). This mapping, that may include also data transformation and manipulation (e.g. transforming integer values carried by Modbus registers into floating-point values), is most likely created ad-hoc for each instance of the pair <Platform A, Platform B>
- IoT application: retrieving the measurement from the IoT device PT form Platform A and updating features PV and HPA in Platform B according to ontology Y.

7.7.5 Scenario sequence/flows

Pre-conditions:

- Both platforms are in operating status.
- Both platforms can communicate with one another and exchange data.
- The IoT device PT is registered under Platform A.
- The application representing V is registered under Platform B.
- Platform B runs an entity able to transform semantic-unaware data from Platform A into semantically characterized data suitable for usage by Platform B.
- The ad-hoc data manipulation that is needed for the above B is known and its representation in a form suitable for driving the mapping capability of Platform B has been created.

Step 1: Acquisition and storage of the tested ontology by Platform B/Instantiation of the ontology mapped to the data structure of the Platform B/ Instantiation of the data manipulation structure on Platform B

- Objective: Platform B discovers (if relevant) and obtains its own ontology.
- Validation: the ontology is successfully loaded in Platform B.

- Platform B discovers and obtains the data structure that drives the transformation of data from Platform A.
- Validation: the data structure is successfully loaded in Platform B.

Step 2: Update of the ontology in Platform B

- Objective: Platform B discovers (if relevant) and updates its own ontology.
- Validation: the ontology is successfully loaded in Platform B.
- Platform B discovers and obtains the updated data structure that drives the transformation of data from Platform A.
- Validation: the data structure is successfully loaded in Platform B.

Step 3: Execution of the ontology mapping by Platform B

- Objective: platform B starts the components responsible to perform the mapping between both ontologies.
- Validation: the mapping component is successfully loaded and running in platform B.

Step 4: Ability of Platform B to generate a request referring to the ontology

- Objective: the IoT application V requests the status of the IoT device PT through Platform B, referring to the ontology Y.
- Validation: the request issued from Platform B to Platform A is correctly formed.

Step 5: Ability of Platform B to map a request from ontology Y to semantic-unaware data structure in Platform A

- Objective: the mapping entity translates the request of the status of the IoT device from Platform B into a request referring to the data structure of Platform B.
- Validation: the translated request in Platform B is correctly formed.

Step 6: Ability of the Platform A to understand a request

- Objective: Platform A processes the request from Platform B and retrieves the requested data from the IoT device PT.
- Validation: Platform A has successfully retrieved the requested data from the IoT device.

Step 7: Ability of the implementation to generate a response referring to the ontology

- Objective: Platform A retrieves the requested data from the IoT device PT; the response is provided to Platform B.
- Platform B receives the response from Platform A and maps it to ontology Y; the result of the mapping is passed to application V.
- Validation: Application V has received the data requested, mapped to ontology Y, and features PV and HPA are updated successfully (this may be displayed to the testing person).

7.8 Scenario D-2: semantic interworking between vendor-specific IoT devices in the same platform using the same ontology

7.8.1 Test configuration

CFG-02: Two IoT devices/Applications on a single IoT platform.

7.8.2 Scenario high level objective

Semantic interworking in the same platform.

7.8.3 Description

The objective of this scenario is to validate the correct operation between two IoT devices using the same ontology in the same domain.

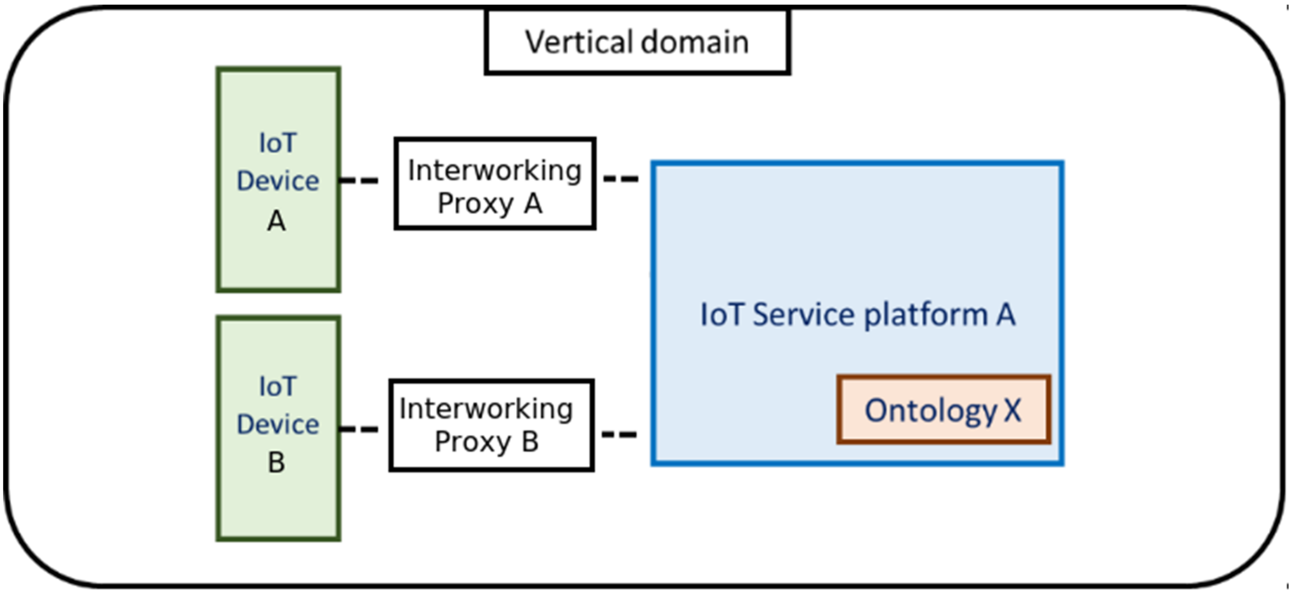


Figure 14: Semantic interworking between two IoT devices in the same domain

The scenario aims at validating that two vendor-specific IoT devices are able to exchange data between each other through both operating under the same platform and using the same ontology in the same vertical domain. For example, the status of lamp from vendor A is updated according to a luminosity sensor from vendor B.

7.8.4 Actors/Entities involved

The scenario involves the following actors:

- IoT device A: a sensor measuring data.
- IoT device B: an actuator controlling a system.
- Interworking proxy A: reads data from IoT device A, convert it from IoT device A specific format to ontology X, and send it to the IoT platform.
- Interworking proxy B: receive a request from IoT platform, convert it from ontology X to IoT device specific format, and control IoT device B.
- IoT platform: including the ontology X describing the semantic used by IoT device A and IoT device B.

7.8.5 Scenario sequence/flows

Pre-conditions:

- The IoT platform in operating status.
- The IoT device A is registered under the IoT platform via interworking proxy A.
- The IoT device B is registered under the IoT platform via interworking proxy B.

Step 1: Acquisition, storage and instantiation of the common ontology by the IoT platform

- Objective: the IoT platform to discover (if relevant) and obtain the common ontology.
- Validation: the ontology is successfully loaded in the IoT platform.

Step 2: Update of the ontology in the IoT platform

- Objective: the IoT platform to discover (if relevant) and update the common ontology.
- Validation: the ontology is successfully loaded in the IoT platform.

Step 3: Ability of IoT platform to receive a request from IoT device A using a specific format

- Objective: Interworking proxy A to receive data from IoT device A the IoT platform using a specific format.
- Validation: the request issued from IoT device A to interworking proxy A is correctly formed.

Step 4: Ability of IoT device A to send a request to the IoT platform referring to the ontology

- Objective: Interworking proxy A to send data to the IoT platform, referring to the common ontology.
- Validation: the request issued from interworking proxy A to the IoT platform is correctly formed.

Step 5: Ability of IoT device B to receive a request from the IoT platform referring to the ontology

- Objective: Interworking proxy B receives data from the IoT platform through interworking proxy B, referring to the common ontology.
- Validation: the request received by Interworking proxy B from the IoT platform is correctly formed.

Step 6: Ability of IoT Platform A to send a request to the IoT device B using a specific format

- Objective: Interworking proxy B sends a request to IoT device B using a specific format.
- Validation: the request issued from interworking proxy B to IoT device B is correctly formed.

7.9 Scenario E-1: Sensor data reporting between platforms in the same vertical domain using different ontologies

7.9.1 Test configuration

CFG-06: Multiple IoT devices/Applications on multiple IoT platforms using different ontologies

NOTE: This scenario can also be run with configuration CFG-05 (Single IoT devices/Applications on multiple IoT platforms using different ontologies), where data is stored on Platform B.

7.9.2 Scenario high level objective

Cross-platform heterogeneous semantic interoperability in the same domain (different ontologies).

7.9.3 Description

Objective: The objective of this scenario is to validate the correct operation between two platforms using different ontologies in the upstream direction. This scenario requires an entity performing the mapping between both ontologies. A similar scenario could be defined where both ontologies are mapped to a unified ontology.

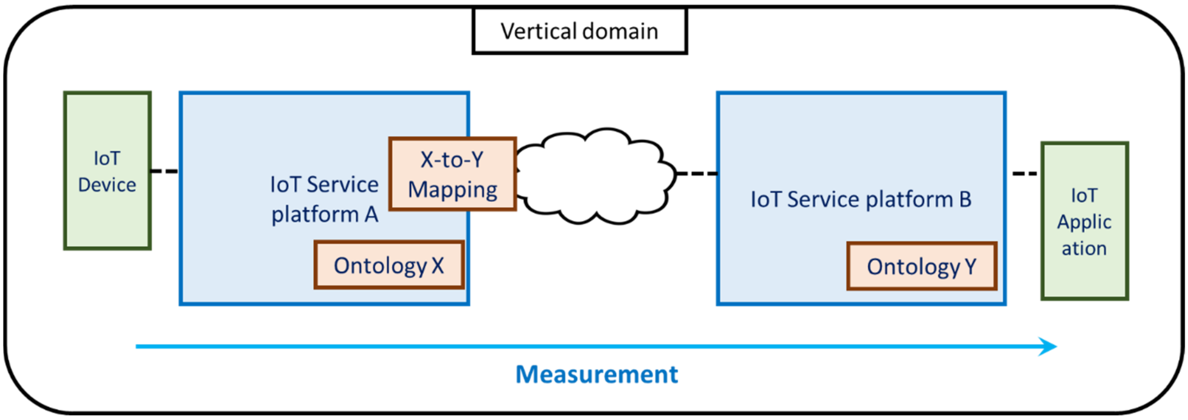


Figure 15: Cross-platform semantic interoperability in the same domain but with different ontologies- upstream direction

The scenario aims at validating that an application operating in an IoT service platform is able to retrieve and process the meaning of a measurement performed by a sensor operating under another IoT platform using a different ontology, but while both of them are operating in the same vertical domain. For example, the status of a light in Platform A is retrieved by a smartphone application running in Platform B.

A similar scenario could be defined in the downstream direction, where the data flow tested is a command issued by the IoT application.

Possible instantiation: the status of a wearable running on a platform enabled with SSN (for example from the MONICA LSP implementation) is transferred to a smartphone application registered on a oneM2M platform enabled with SAREF.

7.9.4 Actors/Entities involved

The scenario involves the following actors:

- IoT device: making the measurement.
- IoT service Platform A: including the ontology X, which describes the semantic used by the IoT device The IoT device is operating under this IoT service platform. The platform includes an entity able to perform the mapping between ontology X and ontology Y.
- IoT service Platform B: including the ontology Y, which describes the semantic used by the IoT application.
- IoT application: retrieving the measurement from the IoT device and the ontology Y from the IoT service platform.

7.9.5 Scenario sequence/flows

Pre-conditions:

- Both platforms are in operating status.
- Both platforms can communicate with one another and exchange data.
- The IoT device is registered under Platform A.
- The application is registered under Platform B.
- Platform A runs an entity able to perform the mapping between both ontologies.

Step 1: Acquisition and storage of the tested ontology by the Platform A [Platform B]/Instantiation of the ontology mapped to the data structure of the Platform A [Platform B]

- Objective: each platform discovers (if relevant) and obtains its own ontology.

- Validation: the ontology is successfully loaded in each of the platforms.

Step 2: Update of the ontology in the Platform A [Platform B]

- Objective: each platform discovers (if relevant) and update its own ontology.
- Validation: the ontology is successfully loaded in each of the platforms.

Step 3: Execution of the ontology mapping by the Platform A

- Objective: Platform A starts the components responsible to perform the mapping between both ontologies.
- Validation: the mapping component is successfully loaded and running in Platform A.

Step 4: Ability of the Platform B to generate a request referring to the ontology

- Objective: the IoT application requests the status of the IoT device through Platform B, referring to the ontology Y.
- Validation: the request issued from Platform B to Platform A is correctly formed.

Step 5: Ability of the Platform A to map a request from the ontology Y to the ontology X

- Objective: the mapping entity translates the request of the status of the IoT device from Platform B into a request referring to the ontology X.
- Validation: the translated request in Platform A is correctly formed.

Step 6: Ability of the Platform A to understand a request referring to the ontology

- Objective: the Platform A processes the request from Platform B and uses the ontology X to retrieve the requested data from the IoT device.
- Validation: Platform A has successfully retrieved the requested data from the IoT device.

Step 6a: Ability of the Platform A to understand a gap of mapping in the ontology when receiving a request

- Objective: the Platform A processes the request from Platform B and uses the ontology X to retrieve the requested data from the IoT device. The ontology describing the IoT device after translation does not match the ontology X, which may be the result of a mapping gap or error.
- Validation: Platform A has successfully reported an error when processing the request from Platform B after it was mapped to its own ontology.

Step 7: Ability of the implementation to generate a response referring to the ontology

- Objective: the Platform A uses the common ontology to retrieve the requested data from the IoT device and maps it to ontology Y before it sends a response to Platform B. The response is provided to the IoT application.
- Validation: The IoT application has obtained successfully a valid data from the IoT device (this may be displayed to the testing person).

7.10 Scenario E-2: Cross-Domain Interoperability, Different Ontologies

7.10.1 Test configuration

CFG-06: Two IoT devices/Applications on multiple IoT platforms using two different ontologies.

7.10.2 Scenario high level objective

Cross-platform heterogeneous semantic interoperability across different domains. The two platforms use two different ontologies.

7.10.3 Description

Objective: The objective of this scenario is to validate the correct operation (in the upstream direction) between two platforms applied to different domains in the upstream direction. The two platforms use two different ontologies. This scenario requires that a linking between the two domains is in place. It is assumed that this linking or mapping takes place within Platform B (see Figure 16).

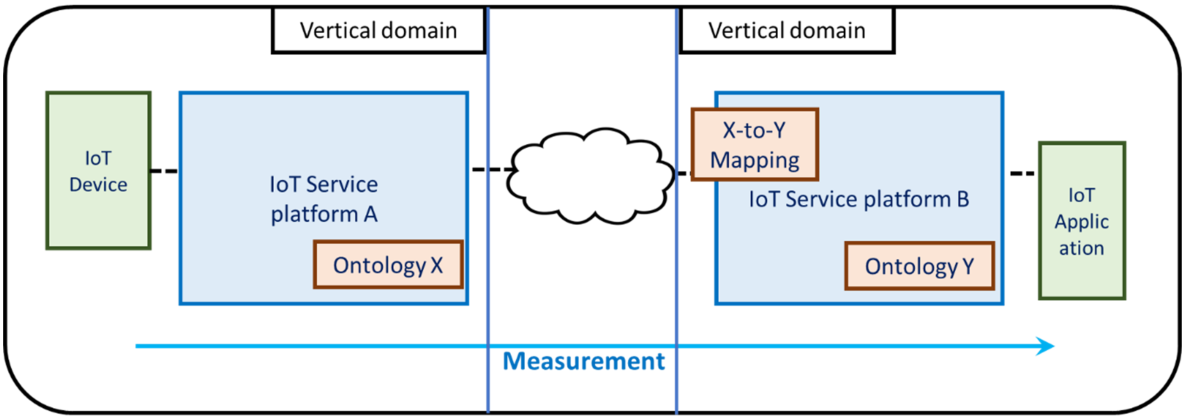


Figure 16: Cross-platform semantic interoperability applied to different domains using different ontologies - upstream direction

The scenario aims at validating that an application operating in an IoT platform (B) is able to retrieve and process the effect on its vertical domain of a measurement performed by a sensor operating under another IoT platform (A).

Possible instantiation:

- Platform A is a generic data collection platform, in which an IoT input device pressure transmitter (PT) is capable of measuring a pressure value.
- Platform B is a process monitoring platform in which an entity Vessel (V) is defined, representing the status of a process vessel.
- Among the features of vessel V are a) PV: the current value of pressure in the vessel and b) HPA (High Pressure Alarm): a high-pressure alarm linked to the current value of PV.
- The status of both PV and HPA change accordingly to the current value of the measurement PT obtained through Platform A. An alarm condition may be raised or reset on Platform B, reflecting the current value of the measurement PT obtained through Platform A.
- Ontologies:
  - ontology X: oneM2M base ontology;
  - ontology Y: an ontology based on ISA-95 standard.

7.10.4 Actors/Entities involved

The scenario involves the following actors:

- IoT device PT: making the measurement.
- IoT Platform A: including the ontology X, which describes the semantic used by the IoT device. The IoT device PT is operating under this IoT platform.

- IoT Platform B: including the ontology Y, which describes the semantic used by the IoT application representing vessel V. The platform includes an entity able to perform the mapping between features of PT (in particular, the value of the current measurement) and features of V (in particular PV, the current value of vessel pressure).
- IoT application: retrieving the measurement from the IoT device PT form Platform A and updating features PV and HPA in Platform B according to ontology Y.

7.10.5 Scenario sequence/flows

Pre-conditions:

- Both platforms are in operating status.
- Both platforms can communicate with one another and exchange data.
- The IoT device PT is registered under Platform A.
- The application representing V is registered under Platform B.
- Platform B runs an entity able to perform the mapping between both ontologies.

Step 1: Acquisition and storage of the tested ontology by the Platform A [Platform B]/Instantiation of the ontology mapped to the data structure of the Platform A [Platform B]

- Objective: each platform discovers (if relevant) and obtains its own ontology.
- Validation: the ontology is successfully loaded in each of the platforms.

Step 2: Update of the ontology in Platform A [Platform B]

- Objective: each platform discovers (if relevant) and updates its own ontology.
- Validation: the ontology is successfully loaded in each of the platforms.

Step 3: Execution of the ontology mapping by Platform B

- Objective: Platform B starts the components responsible to perform the mapping between both ontologies.
- Validation: the mapping component is successfully loaded and running in Platform B.

Step 4: Ability of the Platform B to generate a request referring to the ontology

- Objective: the IoT application V requests the status of the IoT device PT through Platform B, referring to the ontology Y.
- Validation: the request issued from Platform B to Platform A is correctly formed.

Step 5: Ability of the Platform B to map a request from the ontology Y to the ontology X

- Objective: the mapping entity translates the request of the status of the IoT device from Platform B into a request referring to the ontology X.
- Validation: the translated request in Platform B is correctly formed.

Step 6: Ability of the Platform A to understand a request referring to the ontology

- Objective: Platform A processes the request from Platform B and uses the ontology X to retrieve the requested data from the IoT device PT.
- Validation: Platform A has successfully retrieved the requested data from the IoT device.

Step 6a: Ability of the Platform A to understand a gap of mapping in the ontology when receiving a request

- Objective: Platform A processes the request from Platform B and uses the ontology X to retrieve the requested data from the IoT device. The ontology describing the IoT device after translation does not match the ontology X, which may be the result of a mapping gap or error.

- Validation: Platform A has successfully reported an error when processing the request from Platform B after it was mapped to its own ontology.

Step 7: Ability of the implementation to generate a response referring to the ontology

- Objective: Platform A uses ontology X to retrieve the requested data from the IoT device PT; the response is provided to Platform B.

- Platform B receives the response from Platform A and maps it to ontology Y; the result of the mapping is passed to application V.

- Validation: Application V has received the data requested, mapped to ontology Y, and features PV and HPA are updated successfully (this may be displayed to the testing person).

8 Guidelines for the preparation of a Plugtests^{TM} event

8.1 General guidelines

The present document has documented generic scenarios and configurations to be used during a semantic interoperability Plugtests^{TM} event. Further preparation work is needed to be able to organize and set up the running of such an event.

The initial preparation activity from the organization team would require to choose with interested stakeholders the objective and purpose of the test among the possible situations described in clause 6.1, together with the event date and venue.

At this stage, it is important to identify the relevant specifications and ontologies to be tested and the corresponding test configurations, from the configurations defined in clause 7.

From the results of ETSI TR 103 535 [i.1], it appears that the number of standardized semantic-enabled frameworks is limited. Such an event would then have to choose whether the tests are run inside one framework, using specifications from the same origin (for example, like the semantic interoperability that were organized by oneM2M in December 2017) or across different frameworks and set of specifications (for example, mixing SAREF and SSN implementations), allowing more platforms and implementations to be involved in the test.

Once the purpose of the event is agreed, dedicated test specifications that describe unitary test scenarios need to be written, to support the interoperability test. Each scenario will test one specific feature in a specific configuration and permit to declare whether interoperability is achieved for that feature, based on specific validation criteria which can be human observable (e.g. an application shows the successful reception of a measured value on an HMI) or obtained through logging tools in the implementations. The detailed testing scenarios are written using the flows of the applicable generic scenarios described in clause 6 as a baseline.

8.2 Guidelines for IT and infrastructure needed to run the test

The organizing team is responsible to set up the logistics and infrastructure of the test venue to support the different configurations and implementations. This includes, but is not restricted to proposing an IP network with WiFi and Ethernet access, as well as the routers and servers required to enable the communication between the different implementations. If radio communications other than Wi-Fi are necessary, they should be requested beforehand during the preparation virtual meetings or provided by the testers.

The organizing team is also responsible to collect the relevant information from the testing teams about the features they support and prepare the test scheduling. The reporting tool to be used during the interoperability testing should be linked to the event schedule, to enable each testing team to signal the tests that passed and those which failed. This tool should respect a strict confidentiality of the results, in order to not disclose the status of an implementation to its competitors.

During the event, the test schedule should be updated according to the progress of the interoperability tests and the potential issues found by the testers.

If relevant, conformance tests between a test system developed for this purpose and a tested implementation can be organized. This allows to verify that the implementation tested under these test suites is ready to run the interoperability test.

8.3 Guidelines for the preparation of test reporting

The test reporting should be prepared from the early stages of the event preparation.

The test report will identify clearly:

- the specifications and ontologies used by the different implementations;
- the infrastructure and tools used to support the testing;
- the list of detailed tests run during the event, classified by the features tested (e.g. one group of tests for ontology management, another group for the usage of an ontology and yet another group to validate the successful exchange of data between the two implementations);
- the global statistics of results obtained during the event;
- recommendations for the update of standards and ontologies used during the test.

The preparation of the present document requires continuous exchange between the test organizing team and the participants, from the early days of the preparation until the completion of the event.

Virtual meetings with future attendees should be organized during the preparation phase before the event, to explain the test organization to future participants and ensure that a maximum number of potential questions has been answered before the start of the event. These meetings could also help fine tune the testing specification according to the target results expected by the participants.

During the event, periodic wrap-up meetings should be organized to share the statistics of the test results, discuss with the group of testers the potential issues or ambiguities found in the ontologies used, the specifications or the interpretation of how the interworking entities should behave. Potential improvements of the definition of the different entities and ontologies can also be discussed. However, at this stage, as highlighted before, detailed results should not be disclosed to respect the confidentiality of the different implementations.

After the event, the organizing team will summarize these discussions to produce a testing report to be distributed to the different interested parties and stakeholders, when possible to the standardization body, once again preserving the confidentiality of the detailed results

9 Conclusion

Stakeholders wishing to organize and run a semantic interoperability Plugtests™ event will find in the present document all the necessary information to do so.

First step would be to characterize their global approach to IoT semantic interoperability, as explained in clause 5.

Second step would be to identify the objective of the Plugtests™ event, together with the features to be tested and when relevant, the set of standards against which the interoperability test will be run. This test has been described in clause 5.

Next step is to determine the testing configurations to be demonstrated according to the objectives determined in the first step. The possible testing configurations are given in clause 6.

Based on these configurations, some scenarios and testing sequences need to be agreed and documented. Clause 7 provides a list of generic scenarios and related data flows to be instantiated according to the objective of the interoperability test. These scenarios are described and ordered according to the complexity of their high-level objective, the last ones being the most challenging as they intend to show semantic interoperability across platforms enabled with heterogeneous ontologies, or not enabled with semantics.

Finally, the Plugtests™ event should be prepared by an organization team, clause 8 provides them with a detailed list of guidelines in that aim, including guidelines for the IT and infrastructure needed to run the test or for the preparation of the test reporting.

The present document has covered all the steps needed to define and prepare the organization of a Plugtests™ event on Semantic Interoperability.

Annex A:
Change History

Date	Version	Information about changes
November 2018	0.0.1	Preliminary draft with table of content and scope uploaded to SmartM2M
March 2019	0.2.0	Stable draft uploaded to SmartM2M
May 2019	0.2.1	Final draft uploaded to SmartM2M
June 2019	0.2.2	Final draft with comments received during the RC updated
July 2019	0.2.3	SmartM2M#50 conditional approval fulfilled and review by Technical Officer for PU

History

Document history		
V1.1.1	September 2019	Publication