ETSI GS CIM 016 V1.1.1 (2021-04)

Context Information Management (CIM); NGSI-LD Testing Framework: Test Template

Disclaimer

The present document has been produced and approved by the cross-cutting Context Information Management (CIM) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGS/CIM-0016v111

Keywords
API, IoT, 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 Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) cross-cutting Context Information Management (CIM).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**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.

Executive summary

The present document is providing operational guidance to the one developing and using the NGSI-LD test suites. It identifies the different test configurations which may be encountered within ISG CIM architectures. It also defines a template for the Test Purposes (TP) description and finally lists the Implementation Conformance Statements (ICSs).

Introduction

The ISG CIM group has defined an API for exchange of information contextualized in time, space and relation to other information using a property graph model with the intent that the associated protocol (called NGSI-LD) becomes the "glue" between all kinds of applications and databases associated with services for Smart Cities, Smart Agriculture, Smart Manufacturing, etc.

To be successful, the NGSI-LD API specification needs to be well understood and well implemented. The community of users will not be solely highly professional engineers employed by big companies but will include many small teams and SMEs and even hobbyists. Therefore, it is essential that the developers have access to not only the standard but also a test specification and a testing environment to check that their work is (and remains) conformant to the ETSI NGSI-LD specification.

The developers will usually write integration tests to validate the behaviour of their NGSI-LD implementation, but it is important to assert compliance to the specification based on a test suite agreed by the group creating the API specification, i.e. ETSI ISG CIM. Therefore, it is very important to create a set of ETSI-approved test cases.

What is more, the existence of such a test suite will likely help to increase the adoption of the NGSI-LD specification by giving developers a ready to use and complete set of sample requests.

The present document defines the underlying structure of the test suite: it first identifies the different test configurations, then defines the template to draft the test purposes and finally specifies the Implementation Conformance Statements (ICSs) aimed at listing the target capabilities of the NGSI-LD specification.

1 Scope

The Testing Framework (document format) specifies a testing framework defining a methodology for the development of the test strategies, test systems and resulting test specifications. The present document identifies the implementation under test (scope of the testing), the format for the test specification, the test architecture, the points of control and observation, the naming conventions (e.g. for test case ID and test case grouping ID), etc. It also provides the Implementation Conformance Statement which is basically a checklist for a client-owner so they know what parts of the specification will be tested and if any is optional. The ICS will be published as a separate GS.

2 References

2.1 Normative 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.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

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 necessary for the application of the present document.

[1] ETSI GS CIM 009 (V1.3.1): "Context Information Management (CIM); NGSI-LD API".

NOTE: Available at https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.03.01_60/gs_CIM009v010301p.pdf.

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] ISO/IEC 9646-7: "Information technology - Open Systems Interconnection - Conformance testing methodology and framework - Part 7: Implementation Conformance Statements".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

NOTE: The letters "NGSI-LD" were added to most terms to confirm that they are distinct from other terms of similar/same name in use in other organizations, however, in the present document the letters "NGSI-LD" are generally omitted for brevity.

NGSI-LD Central Broker: NGSI-LD Context Broker that only uses a local storage when serving NGSI-LD requests, without involving any external Context Sources

NGSI-LD Context Broker: architectural component that implements all the NGSI-LD interfaces

NGSI-LD Context Consumer: agent that uses the query and subscription functionality of NGSI-LD to retrieve context information

NGSI-LD Context Producer: agent that uses the NGSI-LD context provision and/or registration functionality to provide or announce the availability of its context information to an NGSI-LD Context Broker

NGSI-LD Context Registry: software functional element where Context Sources register the information that they can provide

NOTE: It is used by Distribution Brokers and Federation Brokers to find the appropriate Context Sources which can provide the information required for serving an NGSI-LD request.

NGSI-LD Context Source: source of context information which implements the NGSI-LD consumption and subscription (and possibly provision) interfaces defined by the present document

NOTE: It is usually registered with an NGSI-LD Registry so that it can announce what kind of information it can provide, when requested, to Context Consumers and Brokers.

NGSI-LD Distribution Broker: NGSI-LD Context Broker that uses both local context information and registration information from an NGSI-LD Context Registry, to access matching context information from a set of distributed Context Sources

NGSI-LD Federation Broker: Distribution Broker that federates information from multiple underlying NGSI-LD Context Brokers and across domains

3.2 Symbols

Void.

3.3 Abbreviations

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

API	Application Programming Interface
ATS	Abstract Test Suite
GR	Group Report
GS	Group Specification
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
ICS	Implementation Conformance Statement
IoT	Internet of Things
ISG	Industry Specification Group
ISO	International Organization for Standardization
IUT	Implementation Under Test
JSON	JavaScript Object Notation

JSON-LD	JSON Linked Data
NGSI	Next Generation Service Interfaces
PDU	Protocol Data Unit
PICS	Profile Implementation Conformance Statement
RDF	Resource Description Format
RDFS	Resource Description Format Schema
SUT	Implementation Under Test
TB	Technical Body
ToR	Terms of Reference
TP	Test Purposes
TSS	Test Suite Structure
TSS&TP	Test Suite Structure & Test Purposes
URL	Universal Resource Locator

4 Prerequisites and Test Configurations

4.1 Test Configurations

Test configurations are defined upon the different architectures' options defined in clause 4.3 of ETSI GS CIM 009 [1]. Considered architectures are:

- **Centralized architecture:** A Central Broker stores all the context information. There are Context Producers that use update operations to update the context information in the Central Broker and there are Context Consumers that request context information from the Central Broker, either using synchronous one time query or asynchronous subscribe/notify operations.
- **Distributed architecture:** All information is stored by the Context Sources. Context Sources implement the query and subscription part of the NGSI-LD API as a Context Broker does. They register themselves with the Context Registry, providing information about what context information they can provide, but not the context information itself.
- **Federated architecture:** The architecture works in the same way as the distributed architecture described in clause 4.3.3 of ETSI GS CIM 009 [1], except that instead of simple Context Sources, whole domains are registered with the respective Context Broker as point of access. Typically, the domains will be registered to the federation Context Registry on a more coarse-grained level, providing scopes, in particular geographic scopes, that can then be matched to the scopes provided in the requests.

Test configurations are defined to test different entities such as NGSI-LD Broker, NGSI-LD Context Producer, NGSI-LD Context Consumer, NGSI-LD Context Source, etc.

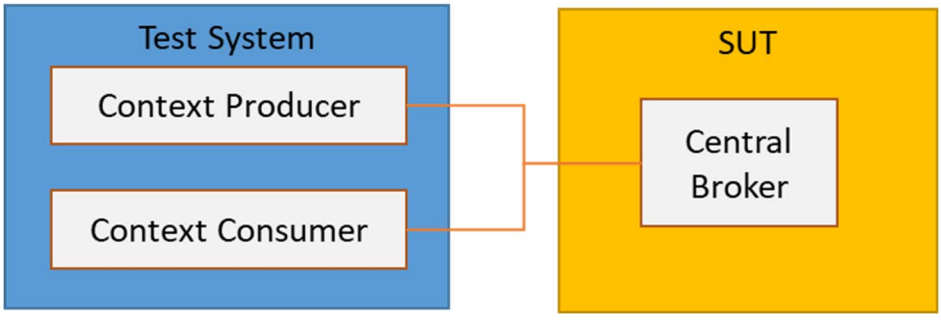

Figure 4.1-1: Test configuration 1 (CF01)

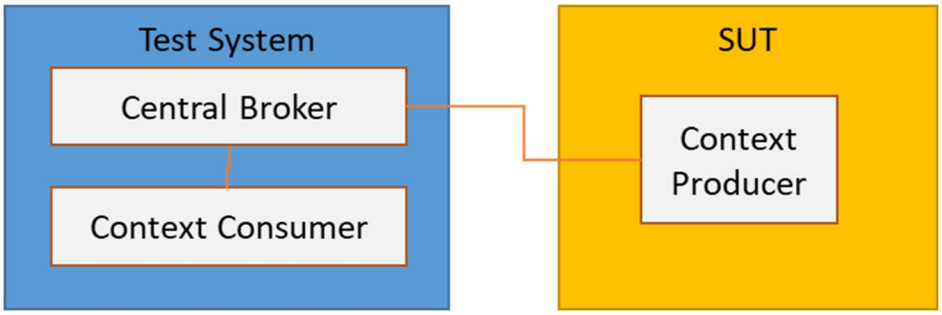

Figure 4.1-2: Test configuration 2 (CF02)

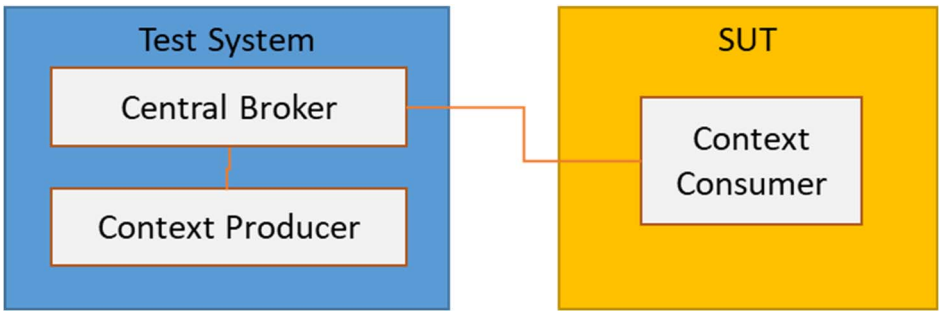

Figure 4.1-3: Test configuration 3 (CF03)

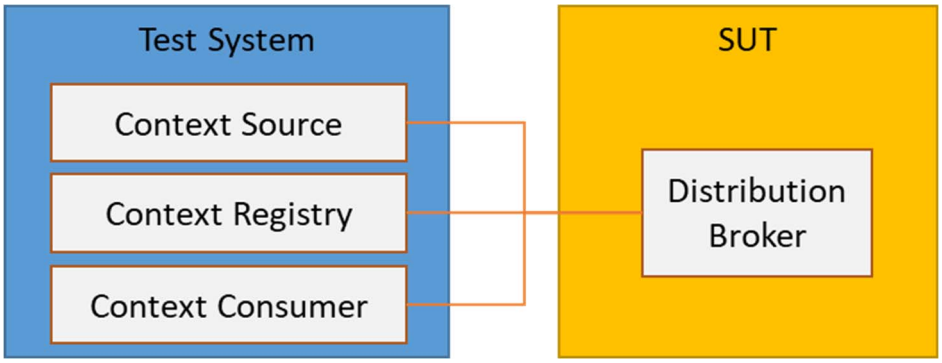

Figure 4.1-4: Test configuration 4 (CF04)

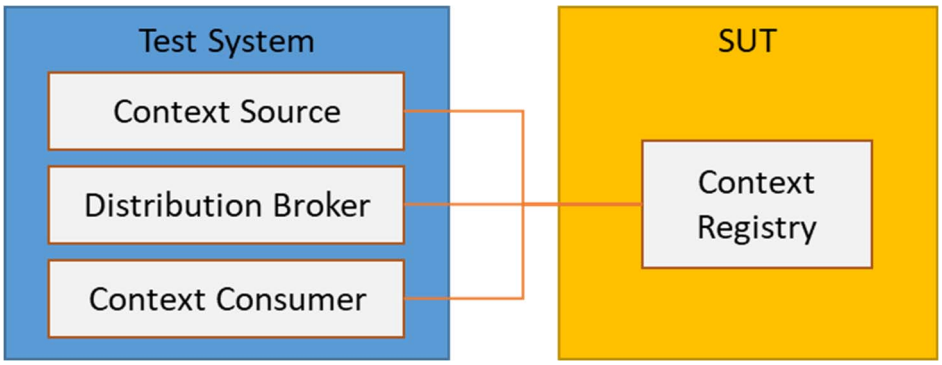

Figure 4.1-5: Test configuration 5 (CF05)

5 Implementation Conformance Statement

An ICS pro forma which conforms to this ICS pro forma specification shall be technically equivalent to annex A, and shall preserve the numbering and ordering of the items in annex A.

An ICS which conforms to this PICS pro forma specification shall:

a) describe an implementation which claims to conform to ETSI GS CIM 009 [1];

b) be a conforming ICS pro forma which has been completed in accordance with the instructions for completion given in clause A.1;

c) include the information necessary to uniquely identify both the supplier and the implementation.

6 Test Purposes (TP)

6.1 Introduction

A test purpose is an informal description of the expected test behaviour. As such it is written in prose.

When needed to clarify the TP, it is helpful to add some graphical presentations, mainly tables, and include message sequence charts.

6.2 TP definition conventions

In order to increase the readability of the TP, the following two recommendations should be followed:

- Each TP should be presented in a table, containing two main parts:
  - The TP header, which contains the TP identifier, the TP objective and the external references (ICS and base standard).
  - The behaviour part, which contains the test behaviour description. This part can be optionally divided in the three following parts, in order to increase the readability:
    - the initial conditions;
    - the expected behaviour;
    - the final conditions.
- The prose describing the test behaviour (including initial and final conditions) should follow some rules, as for instance the use of reserved keywords and syntax.

Table 6.2-1: TP template

TP Id		
Test objective		
References		
Config Id		
PICS Selection		
Initial conditions		
Expected behaviour	**Test events**	**Direction**
	when { **}**	SUT ← Client
	then { **}**	SUT → Client

Table 6.2-2: Description of the fields of the TP template

TP Header	
TP ID	The TP ID is a unique identifier. It shall be specified according to the TP naming conventions defined in clause 6.3.
Test objective	Short description of test purpose objective according to the requirements from the base standard. The test objective clearly indicates which requirement is intended to be tested in the test purpose. This part eases the understanding of the TP behaviour. This also eases the identification of the requirements, which were used as a basis for the test purpose. It is recommended to limit the length of the test objective to one sentence.
References	The reference indicates the clauses of the reference standard specifications in which the conformance requirement is expressed. In the reference row, the TP writer indicates, in which clauses of the protocol standards, the requirement is expressed. This information is critical, because it justifies the existence and the behaviour of the TP. The reference row may refer to several clauses. When the clause containing the requirement is big (for instance, more than ½ page), it is recommended to indicate the paragraph of the clause where the requirement was identified. The reference to the base standard actually is precise enough to enable the TP reader to identify quickly and precisely the requirement.
Config ID	The Config ID is a unique identifier for the configuration required to execute the test.
PICS selection	The PICS selection row contains a Boolean expression, made of ICS parameters. It is recommended to use ICS acronym, which clearly identify the role of the ICS. A mapping table is included in the TP document to link the ICS acronym with its corresponding reference in the ICS document.
Initial conditions	The initial conditions define in which initial state the SUT has to be to apply the actual TP. In the corresponding Test Case, when the execution of the initial condition does not succeed, it leads to the assignment of an inconclusive verdict.
Expected behaviour	Definition of the events, which are parts of the TP objective, and the SUT are expected to perform in order to conform to the base specification. In the corresponding Test Case, Pass or Fail verdicts can be assigned there.

6.3 TP Identifier naming conventions

The TP identifier identifies uniquely the test purposes. In order to ensure the uniqueness of the TP identifier, it follows a naming convention.

The more useful and straightforward naming convention consists of using the test suite structure, to form the first part of the TP identifier. Then the final part consists of a number to identify the TP order within a TP group.

The TP identifier is formed by the abbreviation "TP", followed by abbreviation representing the group of the following TSS levels. Each field of the TP identifier is separated by a "/".

Proposed TP naming convention:

TP/NGSI-LD/<Group>/<Sub Group>/<Sub Sub Group>/<Test Scenario>_<Variant>

A TP identifier, following the above convention of the table could then be:

TP/NGSI-LD/CI/Prov/E/002_03

6.4 Rules for the behaviour description

The following global rules apply, when writing the behaviour description:

- The behaviour description is written in an explicit, exhaustive and unambiguous manner.
- The behaviour description only refers to externally observable test events (send/receive PDUs, timer, counters, etc.) or to events or states, which can be directly or indirectly observed externally.
- All test events used in the behaviour description are part of the procedures specified in the standards.
- The wording of the test events in the behaviour description is explicit, so that the ATS writers do not have to interpret the behaviour description.

- All test events in the behaviour description should result as far as possible in one ATS statement.

The test behaviour is described in prose. This enables to use different ways to express similar behaviour. But using different expressions to define identical behaviours can lead to some misinterpretation of the test purposes. Also, the meaning and the expected order of the test events have a clear and unique meaning for different readers.

Thus, the present document recommends to use pre-defined keywords in order to express clearly and uniquely the test behaviour.

Table 6.4-1 shows some recommended pre-defined keywords and their context of usage. The pre-defined keywords are also likely to be used in combination with the "{" "}"delimiters, in order to clearly delimitate their action in the test behaviour description.

Table 6.4-1 does not present an exhaustive list, so that additional keywords might be defined as necessary. The definition of additional keywords is included in the corresponding TSS&TP document.

Table 6.4-1: List of pre-defined keywords for the behaviour description

Behavioural keywords	
with	with, together with "{" "}" delimiters is used to express the initial conditions, which consist of a set of events, to be executed before starting with the test behaviour corresponding to the test objective. EXAMPLE: With { the SUT having sent a container create request message and ... }
ensure that	ensure that, together with "{" "}" delimiters is used to define the place of the expected behaviour (TP body) or the final conditions. EXAMPLE: ensure that { when { the SUT receives a valid container create request message... }

when/then	when combined with then enables to define the test behaviour involving a combination of stimuli and response events. The when/then combination is used when the occurrence of an event is triggered by the realization of a previous event.
	EXAMPLE:
	ensure that {
	when {
	a XXX signal is activated }
	then {
	the SUT sends a message containing YYY Value indicating "True"} }

Event keywords	
the SUT	Event in the TP is expressed from the point of view of the SUT. This avoid any misinterpretation.
receives	states for an event corresponding to the receipt of a message by the SUT.
having received	states for a condition where the SUT has received a message.
sends	states for an event corresponding to the sending of a message by the SUT.
having sent	states for a condition where the SUT has sent a message.
from/to	Indicates the destination or the origin of a message as necessary (interface, ...) EXAMPLE: ensure that { when { the SUT receives a valid XXX message from the YYY port.. }
on expiry of	Indicate the expiry of a timer, being a stimulus for forthcoming event. EXAMPLE: ensure that { on expiry of the Timer T1, the SUT sends a valid XXX message... }
after expiry of	Used to indicate that an event is expected to occur after the expiry of a timer. EXAMPLE: ensure that { the SUT sends a valid XXX message after expiry of the minimum timer interval }
before expiry of	Used to indicate that an event is expected to occur before the expiry of a timer. EXAMPLE: ensure that { the SUT sends a valid XXX message before expiry of the maximum timer interval }

Event attribute keywords	
valid	Indicates that the event sent or received is a valid message according to the protocol standard, thus: • containing all mandatory parameters, with valid field values; • containing required optional fields according to the protocol context, with valid field values.
invalid	Indicates that the event sent or received is an invalid message according to the protocol standard. Further details describing the invalid fields of the message is added. EXAMPLE: With { the SUT having sent an invalid XXX message containing no mandatory YYY parameter... }
containing	Enables to describe the content of a sent or received message.
indicating	Enables to specify the interpretation of the value allocated to a message parameter. EXAMPLE: With { the SUT having sent a valid XXX message containing a mandatory YYY parameter indicating "ZZZ supported"... }
Logical keywords	
and	Used to combine statements of the behaviour description.
or	
not	

Create Entity TP example:

TP Id	TP/NGSI-LD/CI/Prov/E/002_03	
Test objective	Check that the SUT accepts an Entity create request	
Reference	TS-XXXX [X], clause XXXX	
Config Id	CF_XX	
Parent Release	V1.3.1	
PICS Selection	PICS_XX	
Initial conditions	**with {** **the SUT being** in the "initial state" **}**	
Expected behaviour	**Test events**	**Direction**
	when { **the SUT receives** a valid CREATE Request **from** the client **containing** URL **set to** /ngsi-ld/v1/entities/ **and** Header: Content-Type **set to** application/ld+json **and** body **containing** Entity resource representation **}**	SUT ← Client
	then { **the SUT sends** a valid Response **containing** Response Status Code **set to** 201 (CREATED) **and** Header: Link **containing link to created resource** **}**	SUT → Client

Annex A (normative): ICS Pro forma

A.0 The right to copy

Notwithstanding the provisions of the copyright clause related to the text of the present document, ETSI grants that users of the present document may freely reproduce the ICS pro forma in this annex so that it can be used for its intended purposes and may further publish the completed ICS pro forma.

A.1 Guidance for completing the ICS pro forma

A.1.1 Purposes and structure

The purpose of this ICS pro forma is to provide a mechanism whereby a supplier of an implementation of the requirements defined in ETSI GS CIM 009 [1] may provide information about the implementation in a standardized manner.

The ICS pro forma is subdivided into clauses for the following categories of information:

- guidance for completing the ICS pro forma;
- identification of the implementation;
- global statement of conformance;
- ICS pro forma tables.

A.1.2 Abbreviations and conventions

The ICS pro forma contained in annex A is comprised of information in tabular form in accordance with the guidelines presented in ISO/IEC 9646-7 [i.1].

Item column

The item column contains a number which identifies the item in the table.

Item description column

The item description column describes in free text each respective item (e.g. parameters, timers, etc.). It implicitly means "is <item description> supported by the implementation?".

Reference column

The reference column gives reference to the relevant sections in core specifications.

Status column

The various status used in this annex are in accordance with the rules in table A.1.2-1.

Table A.1.2-1: Key to status codes

Status code	Status name	Meaning
m	mandatory	The capability shall be supported. It is a static view of the fact that the conformance requirements related to the capability in the reference specification are mandatory requirements. This does not mean that a given behaviour shall always be observed (this would be a dynamic view), but that it shall be observed when the implementation is placed in conditions where the conformance requirements from the reference specification compel it to do so. For instance, if the support for a parameter in a sent PDU is mandatory, it does not mean that it shall always be present, but that it shall be present according to the description of the behaviour in the reference specification (dynamic conformance requirement).
o	optional	The capability may or may not be supported. It is an implementation choice.
n/a	not applicable	It is impossible to use the capability. No answer in the support column is required.
c.<integer>	conditional	The requirement on the capability ("m", "o", "n/a") depends on the support of other optional or conditional items. <integer> is the identifier of the conditional expression.
o.<integer>	qualified optional	For mutually exclusive or selectable options from a set. <integer> is the identifier of the group of options, and the logic of selection of the options.

Mnemonic column

The Mnemonic column contains mnemonic identifiers for each item.

Support column

The support column shall be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 9646-7 [i.1], are used for the support column:

N or n	not supported by the implementation
N/A, n/a or no	answer required (allowed only if the status is N/A, directly or after evaluation of a conditional status)
Y or y	supported by the implementation

References to items

For each possible item answer (answer in the support column) within the ICS pro forma there exists a unique reference, used, for example, in the conditional expressions. It is defined as the table identifier, followed by a solidus character "/", followed by the item number in the table.

EXAMPLE 1: A.5/4 is the reference to the answer of item 4 in table A.5.

EXAMPLE 2: A.6/3b is the reference to the second answer (i.e. in the second support column) of item 3 in table 6 of annex A.

Values allowed column

The values allowed column contains the type, the list, the range or the length of values allowed. The following notations are used:

range of values: <min value> .. <max value>

EXAMPLE: 5 .. 20

list of values: <value1>, <value2>, ..., <valueN>

EXAMPLE 1: 2 ,4 ,6 ,8, 9

EXAMPLE 2: '1101'B, '1011'B, '1111'B

EXAMPLE 3: '0A'H, '34'H, '2F'H

list of named values: <name1>(<val1>), <name2>(<val2>), ..., <nameN>(<valN>)

EXAMPLE: reject(1), accept(2)

length: size (<min size> .. <max size>)

EXAMPLE: size (1 .. 8)

Values supported column

The values supported column shall be filled in by the supplier of the implementation. In this column, the values or the ranges of values supported by the implementation shall be indicated.

Prerequisite line

A prerequisite line takes the form: Prerequisite: <predicate>.

A prerequisite line after a clause or table title indicates that the whole clause or the whole table is not required to be completed if the predicate is FALSE.

A.1.3 Instructions for completing the ICS pro forma

The supplier of the implementation shall complete the ICS pro forma in each of the spaces provided. In particular, an explicit answer shall be entered, in each of the support or supported column boxes provided, using the notation described in clause A.1.2.

If necessary, the supplier may provide additional comments in space at the bottom of the tables or separately.

More detailed instructions are given at the beginning of the different clauses of the ICS pro forma.

A.2 Identification of the implementation

A.2.1 Introduction

Identification of the Implementation Under Test (IUT) and the system in which it resides (the System Under Test (SUT)) should be filled in so as to provide as much detail as possible regarding version numbers and configuration options.

The product supplier information and client information should both be filled in if they are different.

A person who can answer queries regarding information supplied in the ICS should be named as the contact person.

A.2.2 Date of the statement

.....

A.2.3 Implementation Under Test (IUT) identification

IUT name:

.....

.....

IUT version:

.....

.....

A.2.4 System Under Test (SUT) identification

SUT name:

.....

.....

Hardware configuration:

.....

.....

.....

Operating system:

.....

.....

A.2.5 Product supplier

Name:

.....

Address:

.....

.....

.....

Telephone number:

.....

Facsimile number:

.....

E-mail address:

.....

Additional information:

.....

.....

.....

A.2.6 Client (if different from product supplier)

Name:

.....

Address:

.....

.....

.....

Telephone number:

.....

Facsimile number:

.....

E-mail address:

.....

Additional information:

.....

.....

A.2.7 ICS contact person

(A person to contact if there are any queries concerning the content of the ICS.)

Name:

.....

Telephone number:

.....

Facsimile number:

.....

E-mail address:

.....

Additional information:

.....

.....

.....

A.3 Identification of the reference specifications

This ICS pro forma applies to the following standard:

[1] ETSI GS CIM 009:"Context Information Management (CIM); NGSI-LD API".

A.4 Global statement of conformance

Are all mandatory capabilities implemented? (Yes/No) .....

NOTE: Answering "No" to this question indicates non-conformance to the NGSI-LD standard specification.

Non-supported mandatory capabilities are to be identified in the ICS, with an explanation of why the implementation is non-conforming, on pages attached to the ICS pro forma.

A.5 Tables

A.5.1 Functional Entities, Roles and Reference Points

NGSI-LD Metamodel existing of Entities, Relationships, Properties, Values defined in ETSI GS CIM 009 [1] are listed in table A.5.1-1:

- An NGSI-LD Entity is a subclass of rdfs:Resource [1].
- An NGSI-LD Relationship is a subclass of rdfs:Resource [1].
- An NGSI-LD Property is a subclass of rdfs:Resource [1].
- An NGSI-LD Value shall be either a rdfs:Literal or a node object (in JSON-LD language) to represent complex data structures [1].
- An NGSI-LD Property shall have a value, stated through hasValue, which is of type rdf:Property [1].
- An NGSI-LD Relationship shall have an object stated through hasObject which is of type rdf:Property [1].

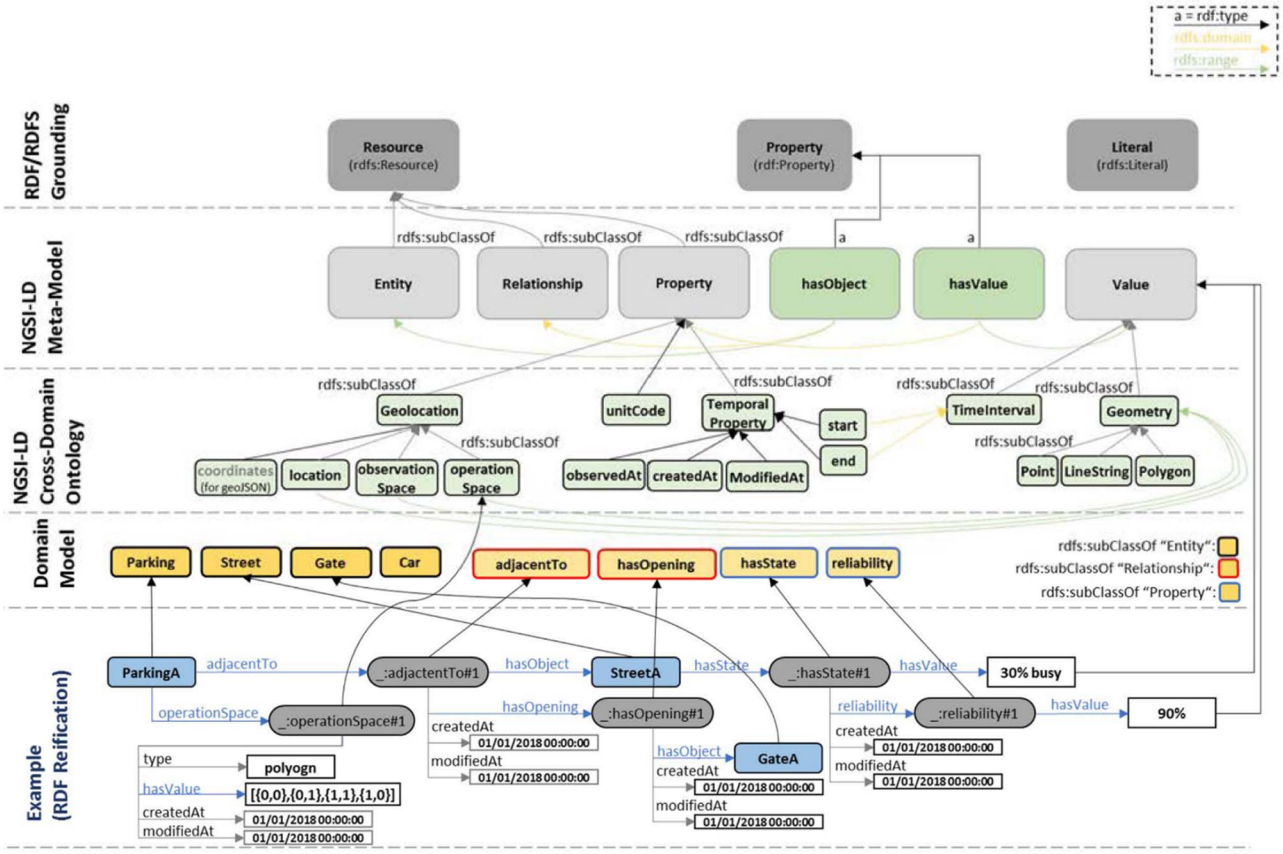

Figure A.5.1-1: Illustration

Table A.5.1-1: NGSI-LD Core Metamodel

Item	Entity Type	Reference clause	Subclass	Condition	Support
1	Entity	[1] 4.2.2	RDFS: Resource	C.1	O Yes O No
2	Relationship	[1] 4.2.2	RDFS: Resource	C.1	O Yes O No
3	Property	[1] 4.2.2	RDFS: Resource	C.1	O Yes O No
4	Value	[1] 4.2.2	RDFS: Literal or node	C.1	O Yes O No
5	GeoProperty	[1] 5.2.7	RDFS: Resource	C.1	O Yes O No
6	EntityInfo	[1] 5.2.8	RDFS: Resource	C.1	O Yes O No
7	CsourceRegistration	[1] 5.2.9	RDFS: Resource	C.1	O Yes O No
8	RegistrationInfo	[1] 5.2.10	RDFS: Resource	C.1	O Yes O No
9	TimeInterval	[1] 5.2.11	RDFS: Resource	C.1	O Yes O No
10	RegistrationInfo	[1] 5.2.12	RDFS: Resource	C.1	O Yes O No
11	Subscription	[1] 5.2.12	RDFS: Resource	C.1	O Yes O No
12	GeoQuery	[1] 5.2.13	RDFS: Resource	C.1	O Yes O No
13	NotificationParams	[1] 5.2.14	RDFS: Resource	C.1	O Yes O No
14	EndPoint	[1] 5.2.15	RDFS: Resource	C.1	O Yes O No
15	BatchOperationResult	[1] 5.2.16	RDFS: Resource	C.1	O Yes O No
16	BatchEntityError	[1] 5.2.17	RDFS: Resource	C.1	O Yes O No
17	UpdateResult	[1] 5.2.18	RDFS: Resource	C.1	O Yes O No
18	NotUpdatedDetails	[1] 5.2.19	RDFS: Resource	C.1	O Yes O No
19	EntityTemporal	[1] 5.2.20	RDFS: Resource	C.1	O Yes O No
19	TemporalQuery	[1] 5.2.21	RDFS: Resource	C.1	O Yes O No
C.1:	The SUT shall be explicitly checked which entity type is going to be tested.				

Table A.5.1-2: Cross-Domain Ontology

Item	Entity Type	Reference clause	Subclass	Condition	Support
1	Geolocation	[1] 4.2.4	RDFS: Property	C.1	O Yes O No
2	unitCode	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
3	Temporal Property	[1] 4.2.4	RDFS: Property	C.1	O Yes O No
4	TimeInterval	[1] 4.2.4	RDFS: Property	C.1	O Yes O No
5	Geometry	[1] 4.2.4	RDFS: Property	C.1	O Yes O No
C.1: The SUT shall be explicitly checked which entity type is going to be tested.					

Table A.5.1-3: Geolocation

Item	Entity Type	Reference clause	Subclass	Condition	Support
1	Coordinates	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
2	location	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
3	Observation space	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
4	Operation space	[1] 4.2.4	RDFS: Property	C.1	O Yes O No
C.1: The SUT shall be explicitly checked which entity type is going to be tested.					

Table A.5.1-4: Temporal Property

Item	Entity Type	Reference clause	Subclass	Condition	Support
1	observedAt	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
2	createdAt	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
3	modifiedAt	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
4	start	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
5	end	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
C.1: The SUT shall be explicitly checked which entity type is going to be tested.					

Table A.5.1-5: Timeinterval

Item	Entity Type	Reference clause	Subclass	Condition	Support
1	Start	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
2	end	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
C.1: The SUT shall be explicitly checked which entity type is going to be tested.					

Table A.5.1-6: Geometry

Item	Entity Type	Reference clause	Subclass	Condition	Support
1	Point	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
2	LineString	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
3	Polygon	[1] 4.2.4	RDFS: Literal or node	C.1	O Yes O No
C.1: The SUT shall be explicitly checked which entity type is going to be tested.					

A.5.2 API Operations

Table A.5.2-1: Context information Provision

Item	Entity Type	Reference clause	Condition	Support
1	Create Entity	[1] 5.6.1	C.1	O Yes O No
2	Update Entity Attributes	[1] 5.6.2	C.1	O Yes O No
3	Append Entity Attributes	[1] 5.6.3	C.1	O Yes O No
4	Partial Attribute Update	[1] 5.6.4	C.1	O Yes O No
5	Delete Entity Attribute	[1] 5.6.5	C.1	O Yes O No
6	Delete Entity	[1] 5.6.6	C.1	O Yes O No
7	Batch Entity Creation	[1] 5.6.7	C.1	O Yes O No
8	Batch Entity Creation or Update (Upsert)	[1] 5.6.8	C.1	O Yes O No
9	Batch Entity Update	[1] 5.6.9	C.1	O Yes O No
10	Batch Entity Delete	[1] 5.6.10	C.1	O Yes O No
11	Create or Update Temporal Representation of an Entity	[1] 5.6.11	C.1	O Yes O No
12	Add Attributes to Temporal Representation of an Entity	[1] 5.6.12	C.1	O Yes O No
13	Delete Attribute from Temporal Representation of an Entity	[1] 5.6.13	C.1	O Yes O No
14	Modify Attribute instance in Temporal Representation of an Entity	[1] 5.6.14	C.1	O Yes O No
15	Delete Attribute instance from Temporal Representation of an Entity	[1] 5.6.15	C.1	O Yes O No
16	Delete Temporal Representation of an Entity	[1] 5.6.16	C.1	O Yes O No
C.1: The SUT shall be explicitly checked which entity type is going to be tested.				

Table A.5.2-2: Context information Consumption

Item	Entity Type	Reference clause	Condition	Support
1	Retrieve Entity	[1] 5.7.1	C.1	O Yes O No
2	Query Entities	[1] 5.7.2	C.1	O Yes O No
3	Retrieve temporal evolution of an Entity	[1] 5.7.3	C.1	O Yes O No
4	Query temporal evolution of Entities	[1] 5.7.4	C.1	O Yes O No
C.1: The SUT shall be explicitly checked which entity type is going to be tested.				

Table A.5.2-3: Context information Subscription

Item	Entity Type	Reference clause	Condition	Support
1	Create Subscription	[1] 5.8.1	C.1	O Yes O No
2	Update Subscription	[1] 5.8.2	C.1	O Yes O No
3	Retrieve Subscription	[1] 5.8.3	C.1	O Yes O No
4	Query Subscriptions	[1] 5.8.4	C.1	O Yes O No
5	Delete Subscription	[1] 5.8.5	C.1	O Yes O No
C.1: The SUT shall be explicitly checked which entity type is going to be tested.				

Table A.5.2-4: Context Source Registration

Item	Entity Type	Reference clause	Condition	Support
1	Register Context Source	[1] 5.9.2	C.1	O Yes O No
2	Update Context Source Registration	[1] 5.9.3	C.1	O Yes O No
3	Delete Context Source Registration	[1] 5.9.4	C.1	O Yes O No
C.1: The SUT shall be explicitly checked which entity type is going to be tested.				

Table A.5.2-5: Context Source Discovery

Item	Entity Type	Reference clause	Condition	Support
1	Retrieve Context Source Registration	[1] 5.10.1	C.1	O Yes O No
2	Query context source registrations	[1] 5.10.2	C.1	O Yes O No
C.1: The SUT shall be explicitly checked which entity type is going to be tested.				

Table A.5.2-6: Context Source Registration Subscription

Item	Entity Type	Reference clause	Condition	Support
1	Create Context Source Registration Subscription	[1] 5.11.2	C.1	O Yes O No
2	Update Context Source Registration Subscription	[1] 5.11.3	C.1	O Yes O No
3	Update Context Source Registration Subscription	[1] 5.11.4	C.1	O Yes O No
4	Query Context Source Registration Subscriptions	[1] 5.11.5	C.1	O Yes O No
5	Delete Context Source Registration Subscriptions	[1] 5.11.6	C.1	O Yes O No
C.1: The SUT shall be explicitly checked which entity type is going to be tested.				

Annex B (informative): Bibliography

ETSI GR CIM 008 (V1.1.1): "Context Information Management (CIM); NGSI-LD Primer".

NOTE: Available at https://www.etsi.org/deliver/etsi_gr/CIM/001_099/008/01.01.01_60/gr_CIM008v010101p.pdf.

Annex C (informative): Change history

Date	Version	Information about changes
September 2021	0.1.1	TTF004 Milestone A
October 2021	0.2.0	Change from GR to GS
October 2021	0.3.0	Add introduction and executive summary Add tests configurations
October 2021	0.3.3	Naming conventions in test configurations
October 2021	1.0.1	Align version with expectation from the ToR
March 2021	1.1.0	Final version approved by ISG-CIM
March 2021	1.1.1	Technical Officer review for EditHelp publication pre-processing

History

Document history		
V1.1.1	April 2021	Publication