

## Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <2019-12-13>

External identifier of this OGC® document: <http://www.opengis.net/doc/BP/ogcapi-common/1.0>

Internal reference number of this OGC® document: YY-nnnrx

Version: n.n

Category: OGC® Best Practice

Editor: Charles Heazel

### OGC API-Common Best Practices

#### Copyright notice

Copyright © 2019 Open Geospatial Consortium

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>

#### Warning

This document defines an the OGC Best Practices on the implementation and use of the OGC API-Common standard. This document is not an OGC Standard and may not be referred to as an OGC Standard. It is subject to change without notice. However, this document is an official position of the OGC membership on this particular technology topic.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OGC® Best Practice

Document subtype:

Document stage: Draft

Document language: English

## License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

# Table of Contents

1. Scope .....	5
2. Conformance .....	6
3. References .....	7
4. Terms and Definitions .....	9
4.1. <b>term name</b> .....	9
5. Conventions .....	10
5.1. Identifiers .....	10
6. OGC API Philosophy .....	11
7. Modular APIs .....	12
8. Navigation .....	13
9. Architecture .....	14
9.1. Datasets .....	14
9.2. Processes .....	14
9.3. Server-side rendering .....	15
9.4. Data partitioning .....	15
9.5. Resource Paths .....	16
10. API Description .....	19
11. Clause containing normative material .....	20
11.1. Requirement Class A or Requirement A Example .....	20
11.1.1. Requirement 1 .....	20
Annex A: Conformance Class Abstract Test Suite (Normative) .....	22
A.1. Conformance Class A .....	22
A.1.1. Requirement 1 .....	22
A.1.2. Requirement 2 .....	22
Annex B: Title ( {Normative/Informative} ) .....	23
Annex C: Revision History .....	24
Annex D: Bibliography .....	25

## **i. Abstract**

<Insert Abstract Text here>

## **ii. Keywords**

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, <tags separated by commas>

## **iii. Preface**

### **NOTE**

Insert Preface Text here. Give OGC specific commentary: describe the technical content, reason for document, history of the document and precursors, and plans for future work. > Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

## **iv. Submitting organizations**

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

Organization name(s)

## **v. Submitters**

All questions regarding this submission should be directed to the editor or the submitters:

Name Affiliation

# Chapter 1. Scope

**NOTE**

Insert Scope text here. Give the subject of the document and the aspects of that scope covered by the document.

# Chapter 2. Conformance

This Best Practice defines XXXX.

Requirements for N target types are considered: \* AAAA \* BBBB

Conformance with this Best Practice shall be checked using all the relevant tests specified in Annex A (normative) of this document.

In order to conform to this OGC® Best Practice, a software implementation shall choose to implement: \* Any one of the conformance levels specified in Annex A (normative). \* Any one of the Distributed Computing Platform profiles specified in Annexes TBD through TBD (normative).

All requirements-classes and conformance-classes described in this document are owned by the document(s) identified.

# Chapter 3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

Insert References here. If there are no references, state “There are no normative references”.

References are to follow the Springer LNCS style, with the exception that optional information may be appended to references: DOIs are added after the date and web resource references may include an access date at the end of the reference in parentheses. See examples from Springer and OGC below.

Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. *J. Mol. Biol.* 147, 195–197 (1981)

May, P., Ehrlich, H.C., Steinke, T.: ZIB Structure Prediction Pipeline: Composing a Complex Biological Workflow through Web Services. In: Nagel, W.E., Walter, W.V., Lehner, W. (eds.) *Euro-Par 2006. LNCS*, vol. 4128, pp. 1148–1158. Springer, Heidelberg (2006)

Foster, I., Kesselman, C.: *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco (1999)

Czajkowski, K., Fitzgerald, S., Foster, I., Kesselman, C.: Grid Information Services for Distributed Resource Sharing. In: 10th IEEE International Symposium on High Performance Distributed Computing, pp. 181–184. IEEE Press, New York (2001)

#### NOTE

Foster, I., Kesselman, C., Nick, J., Tuecke, S.: *The Physiology of the Grid: an Open Grid Services Architecture for Distributed Systems Integration*. Technical report, Global Grid Forum (2002)

National Center for Biotechnology Information, <http://www.ncbi.nlm.nih.gov>

ISO / TC 211: ISO 19115-1:2014 Geographic information — Metadata — Part 1: Fundamentals (2014)

ISO / TC 211: ISO 19157:2013 Geographic information — Data quality (2013)

ISO / TC 211: ISO 19139:2007 Geographic information — Metadata — XML schema implementation (2007)

ISO / TC 211: ISO 19115-3: Geographic information — Metadata — Part 3: XML schemas (2016)

OGC: OGC 15-097 OGC Geospatial User Feedback Standard. Conceptual Model (2016)

OGC: OGC 12-019, OGC City Geography Markup Language (CityGML) Encoding Standard (2012)

OGC: OGC 14-005r3, OGC IndoorGML (2014)



# Chapter 4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this Best Practice.

For the purposes of this document, the following additional terms and definitions apply.

## 4.1. term name

text of the definition

# Chapter 5. Conventions

This sections provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

## 5.1. Identifiers

The normative provisions in this document are denoted by the URI

<http://www.opengis.net/spec/{standard}/{m.n}>

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

# Chapter 6. OGC API Philosophy

The Best Practices document contains information useful to a developer or user of OGC API-Common, but not required by the standard.

# Chapter 7. Modular APIs

This section of the Best Practice discusses OGC principals and practices regarging modular APIs.

# Chapter 8. Navigation

## *Navigating the API*

This section of the Best Practice addresses the different ways to navigate an API, the features of OGC Web APIs to facilitate navigation, and the impact of different navigation techniques on the client.

# Chapter 9. Architecture

The following "architecture" captures the core concepts which inform all OGC API standards.

## 9.1. Datasets

Web APIs implement a Resource Oriented approach to Web-based distributed computing. A coherent set of APIs must be based on a common understanding of the Resources to be shared. For OGC APIs, those resources would be identified using an abstract concept of a geospatial dataset. Such a dataset would have the following characteristics:

- A dataset can be a vector feature collection, a coverage/imagery, or a collection of datasets (therefore potentially a mix of any of these things)
- A dataset has associated metadata, including some essential information:
  - Information about what type of data set it is (vector features (and what type of vector features if limited to one type, e.g. polygons, lines, points), coverages (values) / imagery (pixels), or sub-datasets-- more than one of those things)
  - A textual identifier (e.g. which figures in the resource path)
  - A title (short name / description)
  - Access point for the dataset (could be hosted locally or remote)
  - Geospatial & temporal extent
  - Resolution/scale
  - Units/Range/Bit-Depth/Channels/Dimensions etc. for imagery/coverages
  - A description of queryables, if applicable
- Keywords/Tags, and longer descriptions are also a commonly useful piece of metadata information
- Any other ISO 19115 metadata fields can also be associated with the dataset, but are nowhere near as essential to discovering and using geospatial data as those mentioned above. Meta data containing at minimum those essential elements can always be retrieved in ISO 19115 and potentially other formats.

## 9.2. Processes

Processes take one or more datasets as input, add parameters, and produce one or more datasets as output. This ties the processes together with the data delivery services on both ends.

I suggested so far 3 kinds of processes which can all run on a server where the data lives:

1. The complex process built as a container or executable, as typical of WPS
2. Process description languages such as WCPS
3. Pre-defined named processes such as 'vectorization', 'buffering', 'rasterization' or 'rendering of a styled map'

All of these kinds of processes could share aspects such as taking in an OGC API dataset as input and their output being usable as an OGC API dataset, for direct access and/or asynchronous delivery, and support multiple data partitioning/access mechanisms, estimates/billing elements, and so on.

## 9.3. Server-side rendering

Highlighting here that the rendering of a styled maps based on multiple source data sets and a style as a parameter, outputting a styled 'rendered map' imagery dataset as a result, fits perfectly well the description of a process (3).

## 9.4. Data partitioning

For all of data access / exchange mechanisms, e.g. to retrieve coverages/vector features directly, for a process to access its input, or to retrieve the output from a process, or throughout a daisy chain of processes, there are a variety of ways in which data can be partitioned for efficient access. The most efficient way most likely depends on the overall workflow and the implementations on both ends. I am suggesting most of the OGC API should be agnostic of this and support many such ways, so that both ends of such connection or a workflow manager can negotiate the best approach. Examples of different ways to partition/access data include:

- Bounding boxes
- Tiles
- n-dimensional sub-setting
- DGGS cells
- Single point value

This is the overall diagram of these ideas that I presented a couple weeks back at the OWS Common telecon and got the chance to explain in more details in person to some of you:

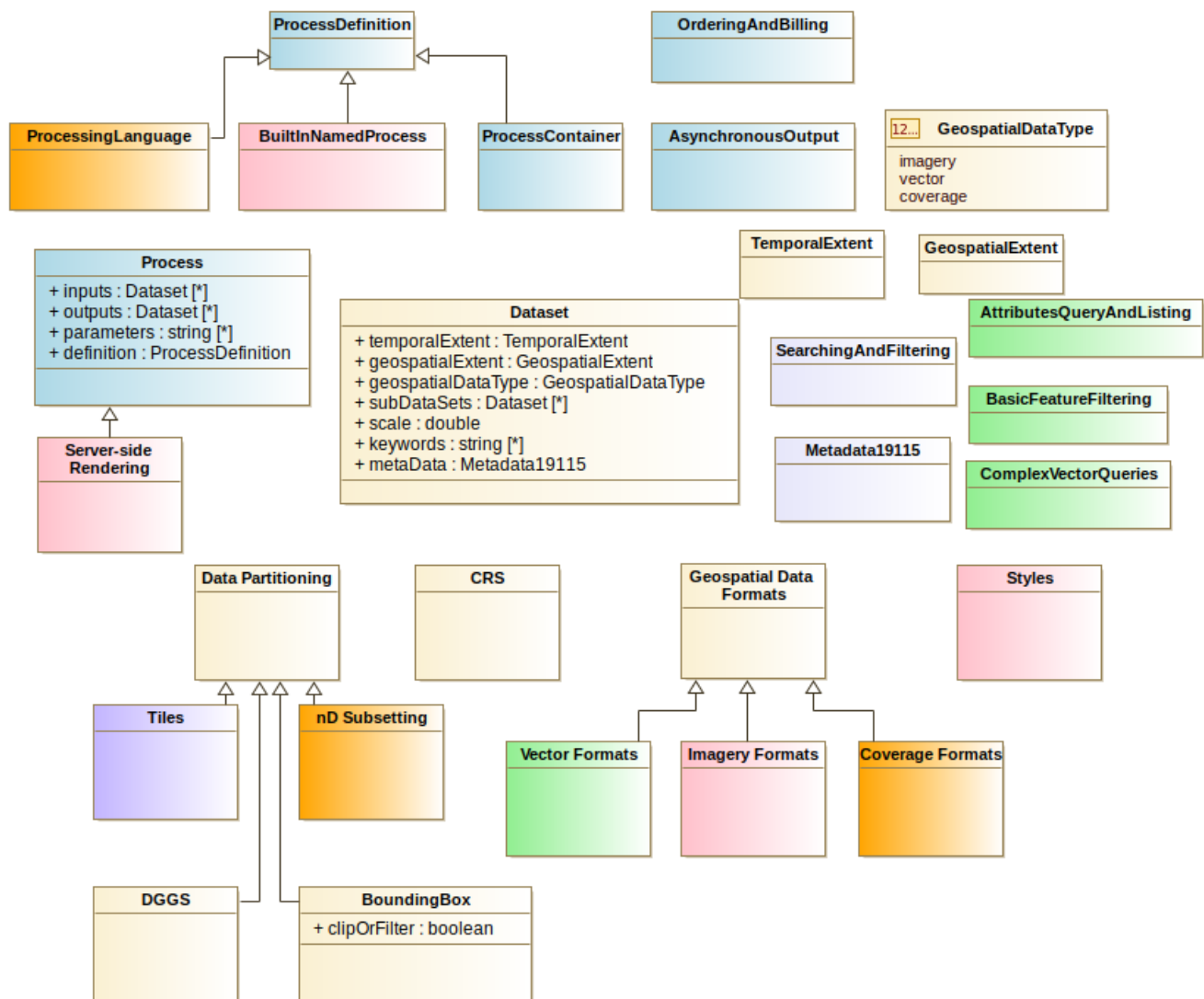


Figure 1. Dataset Concept

You might notice that this is color-coded based on where these API blocks originated from in classic services:

- Orange: Coverages (WCS)
- Red: Server-side rendering (WMS)
- Blue: Processes (WPS)
- Green: Vector features (WFS)
- Very light blue: Catalogs (CSW)
- Violet: Tiles (WMTS)

## 9.5. Resource Paths

The resources exposed through OGC APIs are accessed through standardized URL templates. These "paths" and their associated resources are described in [Table 1](#). This table documents the paths and resources as of December 2019. This information will be updated as these standards mature.

Table 1. OGC Dataset Paths



Path Template	Resource
<b>Common</b>	
/collections	Metadata describing the spatial collections available from this API.
/collections/{collectionId}	Metadata describing the collection which has the unique identifier <b>{collectionId}</b>
<b>Features</b>	
/collections/{collectionId}	The Feature Collection resource identified by the <b>{collectionId}</b> parameter.
/collections/{collectionId}/items	The individual Features in a Feature Collection
<b>Coverages</b>	
/collections/{collectionId}/coverage	A general description of the coverage identified by {collectionId} including the coverage's envelope.
/collections/{collectionid}/coverage/description	returns the whole coverage description consisting of domainset, rangetype, and metadata (but not the rangeset)
/collections/{collectionid}/coverage/domainset	returns the coverage's domain set definition
/collections/{collectionid}/coverage/rangetype	returns the coverage's range type information (i.e., a description of the data semantics)
/collections/{collectionid}/coverage/metadata	returns the coverage's metadata (may be empty)
/collections/{collectionid}/coverage/rangeset	returns the coverage's range set, i.e., the actual values in the coverage's Native Format (see format encoding for ways to retrieve inspecific formats)
/collections/{collectionid}/coverage/all	returns all of the above namely the coverage's domainset, rangetype, meatadata, and rangeset comparable to a GetCoverage response
<b>Maps and Styles</b>	
Note: A map is associated with a resource. {resource} is a place holder for a path segment appropriate for a resource type.	
/collections/{collectionId}/{resource}/map	A data structure with specific information necessary to get a fragment of the map representing the resource collection.
/collections/{collectionId}/{resource}/map/{styleId}	A map representing the geospatial resource identified by /collections/{collectionId}/{resource}.
<b>Tiles</b>	
Note: A tile is associated with a resource. {resource} is a place holder for a path segment appropriate for a resource type.	

Path Template	Resource
/collections/{collectionId}/{resource}/tile/{styleId}/{tileMatrixSetId}/{tileMatrixId}/{tileRow}/{tileCol}	Tile representation of real-world elements at a given resolution restricted by the selected Tile Matrix Set. {styleId} is optional.
/collections/{collectionId}/{resource}/tiles	An enumeration of the Tiles and Styles that are available at some TileMatrixSetId
/tileMatrixSet/{tileMatrixSetId}	A description of the TileMatrixSet identified by the {tileMatrixSetId} identifier.
<b>Processes</b>	
/process	Lists the processes this API offers.
/processes/{process-id}	Returns a detailed description of a process.
/processes/{process-id}/jobs	Returns the running and finished jobs for a process (GET), Executes a process, i.e. creates a new job. Inputs and outputs will have to be specified in a JSON document that needs to be send in the POST body. (POST)
/processes/{process-id}/jobs/{job-id}	Returns the status of a job of a process.
/processes/{process-id}/jobs/{job-id}/results	Returns the result of a job of a process.
<b>Records (Catalog)</b>	
/collections/{collectionId}/tbd	TBD

# Chapter 10. API Description

## *Using and Abusing OpenAPI*

OpenAPI is a useful tool for documenting the design of an API. In addition, numerous tool vendors have produced products to assist in the use of OpenAPI documents. These go so far as to generate stub-code for the use of API and client developers.

However like so many other good tools, OpenAPI was not designed for the size and complexity that we are seeing in Geospatial APIs. This section discusses various features of the OpenAPI specification that can be abused to provide a scalable API Description.

# Chapter 11. Clause containing normative material

Paragraph

## 11.1. Requirement Class A or Requirement A Example

Paragraph – intro text for the requirement class.

Use the following table for Requirements Classes.

Requirements Class	
<a href="http://www.opengis.net/spec/ABCD/m.n/req/req-class-a">http://www.opengis.net/spec/ABCD/m.n/req/req-class-a</a>	
Target type	Token
Dependency	<a href="http://www.example.org/req/blah">http://www.example.org/req/blah</a>
Dependency	urn:iso:ts:iso:19139:clause:6
Requirement 1	<a href="http://www.opengis.net/spec/ABCD/m.n/req/req-class-a/req-name-1">http://www.opengis.net/spec/ABCD/m.n/req/req-class-a/req-name-1</a> requirement description
Requirement 2	<a href="http://www.opengis.net/spec/ABCD/m.n/req/req-class-a/req-name-2">http://www.opengis.net/spec/ABCD/m.n/req/req-class-a/req-name-2</a> requirement description
Requirement 3	<a href="http://www.opengis.net/spec/ABCD/m.n/req/req-class-a/req-name-3">http://www.opengis.net/spec/ABCD/m.n/req/req-class-a/req-name-3</a> requirement description

### 11.1.1. Requirement 1

Paragraph - intro text for the requirement.

Use the following table for Requirements, number sequentially.

Requirement 1	/req/req-class-a/req-name-1
	Requirement 'shall' statement

Dictionary tables for requirements can be added as necessary. Modify the following example as needed.

Names	Definition	Data types and values	Multiplicity and use
name 1	definition of name 1	float	One or more (mandatory)

<b>Names</b>	<b>Definition</b>	<b>Data types and values</b>	<b>Multiplicity and use</b>
name 2	definition of name 2	character string type, not empty	Zero or one (optional)
name 3	definition of name 3	GML:: Point PropertyType	One (mandatory)

# Annex A: Conformance Class Abstract Test Suite (Normative)

## NOTE

Ensure that there is a conformance class for each requirements class and a test for each requirement (identified by requirement name and number)

## A.1. Conformance Class A

### A.1.1. Requirement 1

<b>Test id:</b>	/conf/conf-class-a/req-name-1
<b>Requirement:</b>	/req/req-class-a/req-name-1
<b>Test purpose:</b>	Verify that...
<b>Test method:</b>	Inspect...

### A.1.2. Requirement 2

# Annex B: Title ( {Normative/Informative} )

## NOTE

Place other Annex material in sequential annexes beginning with "B" and leave final two annexes for the Revision History and Bibliography

# Annex C: Revision History

Date	Release	Editor	Primary clauses modified	Description
2016-04-28	0.1	G. Editor	all	initial version



# Annex D: Bibliography

*Example Bibliography (Delete this note).*

The TC has approved Springer LNCS as the official document citation type.

Springer LNCS is widely used in technical and computer science journals and other publications

## NOTE

- For citations in the text please use square brackets and consecutive numbers:  
[1], [2], [3]

– Actual References:

[n] Journal: Author Surname, A.: Title. Publication Title. Volume number, Issue number, Pages Used (Year Published)

[n] Web: Author Surname, A.: Title, <http://Website-Url>

[1] OGC: OGC Testbed 12 Annex B: Architecture. (2015).