OGC® DOCUMENT: 22-054

External identifier of this OGC® document: http://www.opengis.net/doc/PER/eoep-Hack2018

Open Geospatial Consortium

2022 WEB MAPPING CODE SPRINT SUMMARY ENGINEERING REPORT

ENGINEERING REPORT

DRAFT

Submission Date: 2018-12-20
Approval Date: 2018-07-10
Publication Date: 2028-06-12
Editor: Gobe Hobona, Joana Simoes

Open Geospatial Consortium

Note

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.

CONTENTS

LIST OF FIGURES

I EXECUTIVE SUMMARY

The code sprint focused on the following group of specifications:

- OGC API — Maps candidate Standard
- OGC API — Tiles Standard
- OGC API — Styles candidate Standard
- OGC Symbology Conceptual Model: Core Part

The sprint objectives for the Standards Working Groups (SWGs) were to:

- Develop prototype implementations of OGC Standards, including implementations of draft OGC Application Programming Interface (API) Standards;
- Test the prototype implementations;
- Provide feedback to the Editor about what worked and what did not; and
- Provide feedback about the specification document.

This engineering report makes the following recommendations for future innovation work items:

- TBA
- TBA
- TBA

The engineering report also makes the following recommendations for things that the Standards Working Groups should consider:

- TBA
- TBA
- TBA

II KEYWORDS

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

hackathon, application-to-the-cloud, testbed, docker, web service

III SECURITY CONSIDERATIONS

No security considerations have been made for this document.

IV SUBMITTERS

All questions regarding this document should be directed to the editor or the contributors:

NAME	ORGANIZATION	ROLE
Gobe Hobona	Open Geospatial Consortium	Editor
Joana Simoes	Open Geospatial Consortium	Editor
Keith Pomakis	MariaDB	Contributor
Núria Julià Selvas	UAB-CREAF	Contributor
Joan Maso	UAB-CREAF	Contributor
Clemens Portele	interactive instruments GmbH	Contributor
Tom Kralidis	Meteorological Service of Canada	Contributor
Andreas Matheus	Secure Dimensions	Contributor
Add Name	Add Name	Contributor
Add Name	Add Name	Contributor
Add Name	Add Name	Contributor
Add Name	Add Name	Contributor
Add Name	Add Name	Contributor

V ABSTRACT

The subject of this Engineering Report (ER) is a code sprint that was held from November 29th to December 1st, 2022 to advance OGC API Standards that relate to web mapping, and others that relate to styling and symbology encoding standards. The code sprint was hosted by the Open Geospatial Consortium (OGC) and EuroGeographics. The code sprint was sponsored by Ordnance Survey (OS), and was held as a hybrid event with the face-to-face element hosted at the Mundo Madou centre in Brussels, Belgium.

1

SCOPE

1 SCOPE

A Code Sprint is a collaborative and inclusive event driven by innovative and rapid programming with minimal process and organization constraints to support the development of new applications and open standards. Code Sprints experiment with emerging ideas in the context of geospatial standards, help improve interoperability of existing standards by experimenting with new extensions or profiles, and are used for building proofs of concept to support standards development activities and enhancement of software products.

The code sprint described in this engineering report focused on the following specifications:

- OGC API — Maps candidate Standard
- OGC API — Tiles Standard
- OGC API — Styles candidate Standard
- OGC Symbology Conceptual Model: Core Part

The code sprint was organized to provide a collaborative environment that enables software developers, users, and architects to work together on open standards that relate to web mapping, styles, and symbology. The engineering report presents the sprint architecture, the results of the prototyping, and a discussion resulting from the prototyping.

2

NORMATIVE REFERENCES

2 NORMATIVE REFERENCES

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Joan Masó , Jérôme Jacovella-St-Louis: OGC 17-083r4, *OGC Two Dimensional Tile Matrix Set and Tile Set Metadata*. Open Geospatial Consortium (2022). https://docs.ogc.org/is/17-083r4/17-083r4.html.

Joan Masó, Jérôme Jacovella-St-Louis: OGC 20-057, *OGC API — Tiles — Part 1: Core*. Open Geospatial Consortium (2022). https://docs.ogc.org/is/20-057/20-057.html.

Charles Heazel: OGC API — Common — Part 1: Core (Draft). OGC 19-072, Open Geospatial Consortium, http://docs.ogc.org/DRAFTS/19-072.html

Charles Heazel: OGC API — Common — Part 2: Geospatial Data (Draft). OGC 20-024, Open Geospatial Consortium, http://docs.ogc.org/DRAFTS/20-024.html

Leonard Daly, Rollin Phillips: OGC 20-058, *Interoperable Simulation and Gaming Sprint Year 2 Engineering Report*. Open Geospatial Consortium (2021). https://docs.ogc.org/per/20-058.html.

Clemens Portele: OGC API — Styles (Draft). OGC 20-009, Open Geospatial Consortium, http://docs.ogc.org/DRAFTS/20-009.html

Markus Lupp: OGC 05-078r4, *OpenGIS Styled Layer Descriptor Profile of the Web Map Service Implementation Specification*. Open Geospatial Consortium (2007). https://portal.ogc.org/files/?artifact id=22364.

3

TERMS, DEFINITIONS AND ABBREVIATED TERMS

3 TERMS, DEFINITIONS AND ABBREVIATED TERMS

This document uses the terms defined in OGC Policy Directive 49, 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 document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

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

3.1. API

An Application Programming Interface (API) is a standard set of documented and supported functions and procedures that expose the capabilities or data of an operating system, application, or service to other applications [adapted from ISO/IEC TR 13066-2:2016].

3.2. coordinate reference system

A coordinate system that is related to the real world by a datum term name [source: ISO 19111]

3.3. dataset

a set of data, published or curated by a single agent, and available for access or download in one or more representations (modified from DCAT: https://www.w3.org/TR/vocab-dcat-2/#dcat-scope).

Note 1 to entry: A Web API implementing OGC API — Common often gives access to a single dataset which may be comprised of one or more *geospatial data resources*.

3.4. **geospatial data resource**

web accessible resource that consists of a set of geospatial data

Note 1 to entry: In Web APIs implementing *OGC API — Common — Part 2: Geospatial Data*, geospatial data resources are referred to as collections and are defined in the *collections* conformance class.

Note 2 to entry: *geodata* is sometimes used in this document as an abbreviation of *geospatial data*

3.5. **geospatial resource aspect**

web accessible resource that represents a component of geospatial information (metadata, schemas…) or geospatial data accessed using a particular mechanism and data model (e.g., feature items, tiles, maps, coverages,…) of a more generic geospatial data resource (e.g., a collection)

Note 1 to entry: Not to be confused with a web accessible resource representation. While resource representations share the same path and are selected by format negotiation, geospatial aspects use different paths. Commonly a geospatial aspect is a subpath of a geospatial data resource.

3.6. **map tile**

tile that contains information in a raster form where the values of cells are colors which can be readily displayed on rendering devices

Note 1 to entry: Map tiles are generated in combination with *OGC API — Maps*.

3.7. **OpenAPI Document**

A document (or set of documents) that defines or describes an API. An OpenAPI definition uses and conforms to the OpenAPI Specification (https://www.openapis.org)

3.8. geographic information

information concerning phenomena implicitly or explicitly associated with a location relative to the Earth (source: ISO 19101)

3.9. map

portrayal of geographic information as a digital image file suitable for display on a computer screen (source: OGC 06-042)

3.10. portrayal

presentation of information to humans (source: ISO 19117)

3.11. map tile

tile that contains information in a raster form where the values of cells are colors which can be readily displayed on rendering devices

Note 1 to entry: Map tiles are generated in combination with *OGC API — Maps*.

3.12. tile

geometric shape with known properties that may or may not be the result of a tiling (tessellation) process. A tile consists of a single connected “piece” without “holes” or “lines” (topological disc).

In the context of a 2D *tile matrix*, a *tile* is one of the rectangular regions of space, which can be uniquely identified by row and column integer indices, making up the tile matrix.

In the context of a geospatial data *tile set*, a *tile* contains data for such a partition of space as part of an overall set of tiles for that tiled geospatial data.

Note 1 to entry: From OGC 19-014r1: Core Tiling Conceptual and Logical Models for 2D Euclidean Space

Note 2 to entry: From OGC 17-083r4: OGC Two Dimensional Tile Matrix Set and Tile Set Metadata standard

Note 3 to entry: Tiles are useful to efficiently request, transfer, cache, display, store and process geospatial data for a specific resolution and area of interest, providing deterministic performance and scalability for arbitrarily large datasets.

Note 4 to entry: Tiles can contain a variety of data types, such as grid-based pictorial representations (map tiles), coverage subsets (coverage tiles), or feature-based representations (vector tiles).

3.13. tile matrix

tiling grid in a given 2D coordinate reference system, associated to a specific scale and partitioning space into regular conterminous *tiles*, each of which being assigned a unique identifier

Note 1 to entry: From OGC 17-083r4: OGC Two Dimensional Tile Matrix Set and Tile Set Metadata standard

Note 2 to entry: Each tile of a tile matrix is uniquely identifiable by a row and a column integer index. The number of rows is referred to as the *matrix height*, while the maximum number of columns is referred to as the *matrix width* (the number of columns can vary for different rows in *variable width tile matrices*).

3.14. tile matrix set

tiling scheme consisting of a set of *tile matrices* defined at different scales covering approximately the same area and having a common coordinate reference system.

Note 1 to entry: From OGC 17-083r4: OGC Two Dimensional Tile Matrix Set and Tile Set Metadata standard

3.15. tile set

a set of *tiles* resulting from tiling data according to a particular *tiling scheme*

Note 1 to entry: From OGC 19-014r1: Core Tiling Conceptual and Logical Models for 2D Euclidean Space, but adapted to clarify that in the context of this document, a tile set refers specifically to a set of tiles containing data and following a common tiling scheme.

3.16. Web API

API using an architectural style that is founded on the technologies of the Web [source: OGC API — Features — Part 1: Core]

3.17. Abbreviated terms

API	Application Programming Interface
CRS	Coordinate Reference System
GIS	Geographic Information System
OGC	Open Geospatial Consortium
OWS	OGC Web Services
REST	Representational State Transfer

4

HIGH-LEVEL ARCHITECTURE

4 HIGH-LEVEL ARCHITECTURE

As illustrated in Figure 1, the sprint architecture was designed with the view of enabling client applications to connect to different servers that implement open geospatial standards that relate to web mapping, styles and symbology. Implementations of OGC API — Maps, OGC API — Tiles, and OGC API — Styles were deployed in participants' own infrastructure in order to build a solution with the architecture shown below in Figure 1.

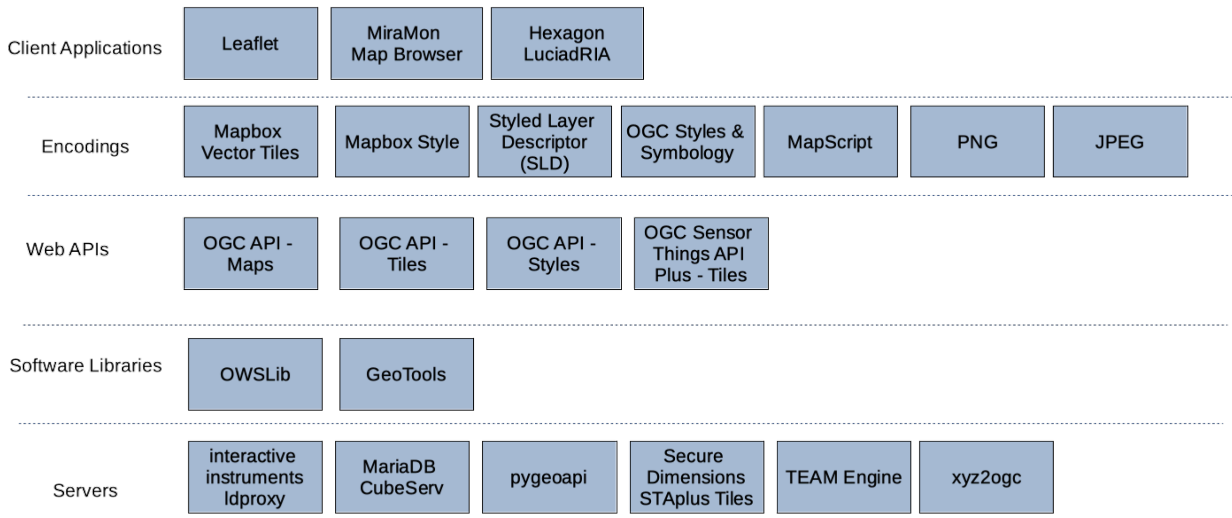

Figure 1 — High Level Overview of the Sprint Architecture

The rest of this section describes the software deployed and standards implemented during the code sprint.

4.1. Approved and Draft Standards

This section describes the approved and draft standards implemented during the code sprint.

4.1.1. OGC API — Maps

The OGC API — Maps draft specification describes an API that can serve spatially referenced and dynamically rendered electronic maps. The specification describes the discovery and query operations of an API that provides access to electronic maps in a manner independent of the underlying data store. The query operations allow dynamically rendered maps to be retrieved from the underlying data store based upon simple selection criteria, defined by the client.

4.1.2. OGC API — Styles

OGC API — Styles specifies building blocks for Web APIs that enable map servers and clients as well as visual style editors to manage and fetch styles that consist of symbolizing instructions that can be applied by a rendering engine on features and/or coverages. The API implements the conceptual model for style encodings and style metadata. The model defines three main concepts, namely the style, stylesheet, and style metadata. The concepts are explained below:

- The **style** is the main resource.
- Each style is available in one or more **stylesheets** — the representation of a style in an encoding like OGC SLD 1.0 or Mapbox Style. Clients will use the stylesheet of a style that fits best based on the capabilities of available tools and their preferences.
- For each style there is **style metadata** available, with general descriptive information about the style, structural information (e.g., layers and attributes), and so forth to allow users to discover and select existing styles for their data.

4.1.3. OGC API — Tiles

OGC API — Tiles specifies a standard for Web APIs that provide tiles of geospatial information. The standard supports different forms of geospatial information, such as tiles of vector features ("vector tiles"), coverages, maps (or imagery) and potentially eventually additional types of tiles of geospatial information.

Vector data represents geospatial objects such as points, lines, and polygons. Tiles of vector feature data (i.e. Vector Tiles) represent partitions of vector data covering a large area (e.g. lines representing rivers in a country).

In this context, a map is essentially an image representing at least one type of geospatial information. Tiles of maps (i.e. Map Tiles) represent subsets of maps covering a large area (e.g. a satellite image).

4.1.4. OGC STAplus — an extension of the OGC SensorThings API

STAplus is an extension to the OGC SensorThings data model. Inspired by Citizen Science applications, STAplus supports the 'ownership concept' whereby observations collected by sensors are owned by (different) users that may express the license for re-use. In addition to the ownership and license abilities, the extension specified by STAplus supports the expression of explicit relations between observations and the creation of group(s) of observations.

4.1.5. OGC Styled Layer Descriptor

The Styled Layer Descriptor (SLD) encoding standard defines an encoding that supports user-defined symbolization and coloring of geographic feature and coverage data. SLD addresses the need for users and software to be able to control the visual portrayal of the geospatial data. The ability to define styling rules requires a styling language that the client and server can both understand. SLD is often used in combination with the OGC Symbology Encoding Standard (SE).

4.2. Open Source Software Projects

This section describes open source software products that were deployed during the code sprint.

4.2.1. OSGeo GeoTools

The GeoTools product is a software library, implemented in Java, that offers a variety of geospatial tools.

4.2.2. OSGeo Leaflet

Leaflet is an open-source JavaScript library for mobile-friendly interactive maps. It works across all major desktop and mobile platforms, can be extended with a variety of plugins, and offers a well-documented API.

4.2.3. ldproxy

ldproxy is an implementation of the OGC API family of specifications, inspired by the W3C/OGC Spatial Data on the Web Best Practices. ldproxy is developed by interactive instruments GmbH, written in Java (Source Code), and is typically deployed using docker (DockerHub). In addition to supporting commonly used data formats for geospatial data, an emphasis is placed on an intuitive HTML representation.

The current version supports PostgreSQL/PostGIS databases, GeoPackages and WFS 2.0 instances as backends for feature data. MBTiles is supported for tilesets.

ldproxy implements all conformance classes and recommendations of “OGC API — Features — Part 1: Core” and “OGC API — Features — Part 2: Coordinate Reference Systems By Reference” and is an OGC Reference Implementation for the two standards. ldproxy also supports the OGC API Records draft as well as the draft extensions of OGC API — Features (that is Part 3, CQL2, Part 4 and most of the current proposals discussed by the Features API working group). It supports GeoJSON, JSON-FG, HTML, FlatGeoBuf, CityJSON, glTF 2.0 and GML as feature encodings.

ldproxy also has implementations for additional resource types: Vector and Map Tiles, Styles, Routes, 3D Tilesets.

ldproxy is distributed under the Mozilla Public License 2.0.

4.2.4. OSGeo OWSLib

OWSLib is a Python package for client programming with OGC Web Service (OWS) standards, and their related content models. OWSLib also supports some OGC API Standards.

4.2.5. OSGeo pygeoapi

pygeoapi is a Python server implementation of the OGC API suite of standards. The project emerged as part of the next generation OGC API efforts in 2018 and provides the capability for organizations to deploy a RESTful OGC API endpoint using OpenAPI, GeoJSON, and HTML. pygeoapi is open source and released under an MIT license. pygeoapi is an official OSGeo Project as well as an OGC Reference Implementation.

pygeoapi supports numerous OGC API Standards. The official documentation provides an overview of all supported standards.

4.2.6. STAplus Viewer App by Secure Dimensions

The STAplus Viewer App is a proof of concept implementation as Web-Browser application based on JavaScript and Leaflet. The implementation further leverages JS libraries from STAM (SensorThings API Map) developed by Datacove for INSPIRE.

The STAplus Viewer App is based on the Leaflet JavaScript library. It is however possible to base the application on OpenLayers. This alternative implementation is available from the COS4Cloud project. Even though it looks and feels slightly different, both implementations provide the same functionality.

4.2.7. TEAM Engine

The Test, Evaluation, And Measurement (TEAM) Engine is a testing facility that executes test suites developed using the TestNG framework or the OGC Compliance Test Language (CTL). It is typically used to verify specification compliance and is the official test harness of the OGC Compliance Testing Program (CITE).

4.2.8. xyz2ogc

The xyz2ogc program supports the generation of OGC API – Tiles metadata from existing XYZ tilesets. The program is implemented in the Go programming language.

4.3. Proprietary products

This section describes proprietary software products that were deployed during the code sprint.

4.3.1. MariaDB CubeWerx CubeServ

The CubeWerx server (“cubeserv”) is implemented in C and currently implements the following OGC Standards and draft specifications.

- Multiple conformance classes and recommendations of the OGC API — Tiles — Part 1: Core Standard.
- Multiple conformance classes and recommendations of the OGC API — Maps — Part 1: Core candidate Standard.
- All conformance classes and recommendations of the OGC API — Features — Part 1: Core Standard.
- Multiple conformance classes and recommendations of the OGC API — Records — Part 1: Core candidate Standard.
- Multiple conformance classes and recommendations of the OGC API — Coverages — Part 1: Core candidate Standard.
- Multiple conformance classes and recommendations of the OGC API — Processes — Part 1: Core Standard.
- Multiple versions of the Web Map Service (WMS), Web Processing Service (WPS), Web Map Tile Service (WMTS) and Web Feature Service (WFS) Standards.
- A number of other “un-adopted” OGC Web Service draft specifications including the Testbed-12 Web Integration Service, OWS-7 Engineering Report — GeoSynchronization Service, and the Web Object Service prototype.

The cubeserv executable supports a wide variety of back ends including Oracle, MariaDB, SHAPE files, etc. It also supports a wide array of service-dependent output formats (e.g., GML, GeoJSON, Mapbox Vector Tiles, MapMP, etc.) and coordinate reference systems.

4.3.2. GNOSIS Map Server

The GNOSIS Map Server is written in the eC programming language and supports multiple OGC API specifications. GNOSIS Map Server supports multiple encodings including GNOSIS Map Tiles (which can contain either vector data, gridded coverages, imagery, point clouds or 3D meshes), Mapbox Vector Tiles, GeoJSON, GeoECON, GML and MapML. An experimental server is available online at https://maps.ecere.com/ogcapi and has been used in multiple OGC Innovation Program initiatives.

4.3.3. Hexagon Luciad RIA

LuciadRIA is a product that enables applications running in a web browser to offer 2D, 3D, or 4D visualization of satellite and other imagery, vector-based data and dynamic content, such as tracks. LuciadRIA can connect to implementations of OGC API Standards, as well as implementations of OGC Web Service standards.

5

RESULTS

5 RESULTS

The code sprint included multiple software products and implementations of OGC and ISO Standards. This section presents some of the results from the code sprint.

5.1. MariaDB CubeWerx CubeSERV

TBA

5.2. ldproxy

TBA

5.3. pygeoapi

TBA

A

ANNEX A (NORMATIVE) ANNEX TITLE

A

ANNEX A (NORMATIVE) ANNEX TITLE

Annex content.

NOTE: Place annex material in sequential order and set `obligation` attribute as “normative” (default) or “informative” according to the case.

B

ANNEX B (INFORMATIVE) REVISION HISTORY

B

ANNEX B (INFORMATIVE) REVISION HISTORY

DATE	RELEASE	AUTHOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2016-04-28	0.1	G. Editor	all	initial version

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Andreas Matheus: OGC 21-068, *OGC Best Practice for using SensorThings API with Citizen Science*. Open Geospatial Consortium (2022). https://docs.ogc.org/bp/21-068.pdf.