

OGC® DOCUMENT: YY-999

External identifier of this OGC® document: <http://www.opengis.net/doc/WP/geosparql3d>



Open
Geospatial
Consortium

GEOSPARQL 3D WHITE PAPER

TECHNICAL PAPER

CANDIDATE SWG DRAFT

Version: 1.0

Submission Date: 2029-03-30

Approval Date: 2029-03-30

Publication Date: 2029-03-30

Editor: Editor One, Editor Two

Notice: This document is not an OGC Standard. This document is an OGC White Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC White Paper should not be referenced as required or mandatory technology in procurements.

License Agreement

Use of this document is subject to the license agreement at <https://www.ogc.org/license>

Copyright notice

Copyright © 2025 Open Geospatial Consortium

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

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

I. KEYWORDS	v
II. PREFACE	vi
III. SECURITY CONSIDERATIONS	vii
IV. SUBMITTING ORGANIZATIONS	viii
V. SUBMITTERS	viii
1. SCOPE	2
2. CONFORMANCE	4
3. NORMATIVE REFERENCES	6
4. TERMS AND DEFINITIONS	8
5. ABSTRACT	10
6. KEYWORDS	12
7. CONVENTIONS	14
8. INTRODUCTION	16
9. CURRENT CAPABILITIES	18
9.1. GeoSPARQL	18
9.1.1. Requirements addressed	18
9.1.1.1. Vocabulary	18
9.1.1.2. Geometry Relations	19
9.1.1.3. Literals	19
9.1.1.4. Query functions with 3D support	20
9.1.2. Adoption of GeoSPARQL 1.1	20
10. REQUIREMENTS FOR GEOSPARQL 3D	22
10.1. Proposed extensions for GeoSPARQL 3D	22
10.1.1. Extension 1: 3D representations	22
10.1.1.1. GitHub Issue URI	22
10.1.1.2. Category	22
10.1.1.3. Description	22

10.1.2. Extension 2: Relations of 3D geometries	23
10.1.2.1. GitHub Issue URI	23
10.1.2.2. Category	23
10.1.2.3. Description	23
10.1.3. Extension 3: Appearance of 3D geometries	23
10.1.3.1. GitHub Issue URI	23
10.1.3.2. Category	23
10.1.3.3. Description	23
10.1.4. Extension 4: Multi-component 3D geometries	24
10.1.4.1. GitHub Issue URI	24
10.1.4.2. Category	24
10.1.4.3. Description	24
10.1.5. Extension 5: Positioning of 3D geometries	24
10.1.5.1. GitHub Issue URI	24
10.1.5.2. Category	24
10.1.5.3. Description	24
10.1.6. Extension 6: Alignments of GeoSPARQL 3D	25
10.1.6.1. GitHub Issue URI	25
10.1.6.2. Category	25
10.1.6.3. Description	25
10.1.7. Extension 7: Alignments of Engineering CRS to Geospatial CRS	25
10.1.7.1. GitHub Issue URI	25
10.1.7.2. Category	25
10.1.7.3. Description	25
10.1.8. Extension 8: Geometry Extrusion	26
10.1.8.1. GitHub Issue URI	26
10.1.8.2. Category	26
10.1.8.3. Description	26
10.1.9. Extension 9: Geometry Attributes	26
10.1.9.1. GitHub Issue URI	26
10.1.9.2. Category	26
10.1.9.3. Description	27
10.1.10. Extension 10: Non-topological Query Functions – 3D Extension	27
10.1.10.1. GitHub Issue URI	27
10.1.10.2. Category	27
10.1.10.3. Description	27
 11. BENEFICIARIES AND BENEFITS	 29
11.1. Beneficiaries	29
11.1.1. Beneficiary 1: Someone who benefits	29
11.2. Benefits	29
11.2.1. Benefit B1: My benefit	29
 12. ANNEX N: N	 31
 13. ANNEX O: HISTORY	 33



KEYWORDS

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

OGC, GeoSPARQL, 3D



PREFACE

To come...



SECURITY CONSIDERATIONS

The following security considerations apply...

IV

SUBMITTING ORGANIZATIONS

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

- Organization one
- Organization two
- Organization three

V

SUBMITTERS

NAME	AFFILIATION	OGC MEMBER
Nicholas J. Car	KurrawongAI	Yes



1

SCOPE



SCOPE



2

CONFORMANCE



CONFORMANCE



3

NORMATIVE REFERENCES



NORMATIVE REFERENCES

There are no normative references in this document.

The background features a dark blue-grey color with several thin, light yellow lines intersecting at various points. Three of these intersection points are marked with small yellow dots. The lines create a network of triangles and other geometric shapes across the page.

4

TERMS AND DEFINITIONS



TERMS AND DEFINITIONS

No terms and definitions are listed in this document.



5

ABSTRACT



ABSTRACT

To come...



6

KEYWORDS



KEYWORDS

To come...

7

CONVENTIONS



CONVENTIONS



8

INTRODUCTION

GeoSPARQL is an open standard that enables storage and exchange of spatial data on the Web, based on the Resource Description Framework (RDF). Although not specifically bound, the current version of GeoSPARQL (1.1) is mainly geared towards spatial objects having zero to two dimensions. In other words, things that can be displayed on a flat surface. However, for three-dimensional spatial objects both supply and demand are increasing.

A future version of GeoSPARQL is expected to have extended capabilities supporting three-dimensional space. Not only would that make GeoSPARQL more useful for 3D geospatial data, it would also help in industries and knowledge domains that are not mainly focused on geospatial data, like Building Information Modelling (BIM) and Computer Graphics (CG). This paper describes which extensions to GeoSPARQL for 3D are most desirable, and how they could be achieved.

The paper consists of three main parts. The first explains the need for additional 3D capabilities in GeoSPARQL. It lists expected benefits, and describes how users and implementers of GeoSPARQL can reap them. The second part describes current capabilities of GeoSPARQL. In its current state, GeoSPARQL does allow modelling three-dimensional objects, but with relevant limitations. The last section describes requirements for extended 3D capabilities in GeoSPARQL. The latter is based on a market consultation, which resulted in a collection of use cases for additional 3D capabilities. By analysing those use cases, as well as current developments within and outside the OGC, insight in the most needed extensions to GeoSPARQL can be obtained. Requirements are then weighed against feasibility; some extensions are easier to achieve than others.

This paper should be interesting for the following audiences:

- Current and prospective users of GeoSPARQL;
- Current and prospective implementers of GeoSPARQL;
- Members of related OGC working groups.



9

CURRENT CAPABILITIES

9.1. GeoSPARQL

GeoSPARQL is the most common geospatial extension of SPARQL. It was accepted as an OGC standard in 2012 and revised as GeoSPARQL 1.1 in 2024.

According to the standard document, “The OGC GeoSPARQL standard supports representing and querying geospatial data on the Semantic Web. GeoSPARQL defines a vocabulary for representing geospatial data in RDF, and it defines an extension to the SPARQL query language for processing geospatial data”.

9.1.1. Requirements addressed

In order to define which capabilities GeoSPARQL needs to adopt for full 3D compatibility, we first take a look at GeoSPARQL 1.1 current capabilities with regards to 3D.

9.1.1.1. Vocabulary

GeoSPARQL 1.1 defines a class `Geometry` as a subclass of `SpatialObject`.

An instance of `Geometry` is not restricted to two dimensions.

A fine-grained classification of `Geometry` can use the Simple Features Vocabulary (TODO: add link) which extends the class `Geometry` with further types, such as *Point*, *Polygon* etc.

The Simple Features vocabulary allows for the definition of 3D variants of:

- (Multi)Points
- (Multi)LineStrings
- (Multi)Polygons

It does not include commons 3D primitives, such as *Cube* or *Mesh* surfaces which are integral parts of 3D representations.

Concerning metadata of 3D models, GeoSPARQL 1.1 provides properties which can be reused in 3D contexts. In particular, the properties are:

- `geo:hasVolume`
- `geo:hasMetricVolume`

Further 3D-related metadata properties such as projection matrices are not part of the current GeoSPARQL 1.1 standard.

9.1.1.2. Geometry Relations

Relations between geometries have been defined using three different sets of rules:

- Simple Features Relations
- Egenhofer Relations
- Region Connection Calculus RCC8

All geometry relations are only defined for 2D geometries and do not take into account the third dimension.

9.1.1.3. Literals

A first requirement for 3D support is the ability to save 3D data in a knowledge graph. GeoSPARQL 1.1 defines a variety of String literal formats, which are investigated for the storage of 3D data.

Table 1

Literal Type	Z-Coordinate Supported	2.5D	3D
WKT Literal	Yes	Yes	Yes
GML Literal	Yes	Yes	Only with extension Schema
KML Literal	Yes	Yes	As import from COLLADA
GeoJSON Literal	Yes	Yes	Yes
DGGS Literal			

GeoSPARQL 1.1 also does not restrict the usage of coordinate reference systems with 3D support. There are currently almost 300 coordinate reference systems in the database epsg.io which can be used to describe 3D data encoded in the GeoSPARQL graph literals listed above.

9.1.1.4. Query functions with 3D support

GeoSPARQL 1.1 functions currently do not offer fully-featured 3D support. However, there are functions which may take into account the Z coordinate, if they are available.

Table 2

GeoSPARQL function	Z-Coordinate Supported	2.5D	3D
geof:is3D	Yes	Yes	Yes
geof:minZ	Yes	Yes	Yes
geof:maxZ	Yes	Yes	Yes

These functions check for the presence of Z coordinates or filter out maximum and minimum Z coordinates of the given geometry.

9.1.2. Adoption of GeoSPARQL 1.1



10

REQUIREMENTS FOR GEOSPARQL 3D

This section provides an overview of feedback received on the current version of the GeoSPARQL standard (version 1.1) regarding 3D usage. It helps to identify some of the barriers to use, and to outline requirements that have not been addressed that may encourage greater uptake.

10.1. Proposed extensions for GeoSPARQL 3D

10.1.1. Extension 1: 3D representations

10.1.1.1. GitHub Issue URI

<https://github.com/opengeospatial/ogc-geosparql/issues/583>

10.1.1.2. Category

Semantic improvement

10.1.1.3. Description

GeoSPARQL should include ways to represent 3D data in a knowledge graph.

3D data should be included in common 3D formats and 3D data should be includable as a text literal and a file representation.

Some common formats which could be considered for inclusion are:

- [Polygon File Format \(PLY\)](#)
- [Wavefront OBJ Format \(OBJ\)](#)
- [GLTF Format \(GLTF\)](#)
- [X3D Format](#)

10.1.2. Extension 2: Relations of 3D geometries

10.1.2.1. GitHub Issue URI

<https://github.com/opengeospatial/ogc-geosparql/issues/416>

10.1.2.2. Category

Semantic improvement

10.1.2.3. Description

GeoSPARQL should include ways to represent relations between 3D geometries and relations between 3D geometries and geometries of lower dimensions. The relations should be expressable in property relations and should be queryable using SPARQL extension functions.

10.1.3. Extension 3: Appearance of 3D geometries

10.1.3.1. GitHub Issue URI

<https://github.com/opengeospatial/ogc-geosparql/issues/592>

10.1.3.2. Category

Semantic improvement

10.1.3.3. Description

GeoSPARQL should include ways to represent materials and textures of 3D geometries, so that geometries can be styled accordingly.

Materials include:

- Colors of surfaces with light diffusion parameters
- Images as textures, which are associated with surfaces of the 3D object

10.1.4. Extension 4: Multi-component 3D geometries

10.1.4.1. GitHub Issue URI

<https://github.com/opengeospatial/ogc-geosparql/issues/591>

10.1.4.2. Category

Semantic improvement

10.1.4.3. Description

GeoSPARQL should include ways to define multi-component 3D geometries, whereas each component expresses its own semantics. For example, parts of a building could have different semantics according to the function of the building components and would be classified as such in an RDF graph.

10.1.5. Extension 5: Positioning of 3D geometries

10.1.5.1. GitHub Issue URI

<https://github.com/opengeospatial/ogc-geosparql/issues/591>

10.1.5.2. Category

Semantic improvement

10.1.5.3. Description

GeoSPARQL should include ways to position 3D geometries in a 3D space. Commonly 3D geometries are rotated, translated and scaled using commonly defined operators in computer graphics. Similar operations are needed for the relative positioning of 3D objects in GeoSPARQL, as properties and potentially as functions.

10.1.6. Extension 6: Alignments of GeoSPARQL 3D

10.1.6.1. GitHub Issue URI

- <https://github.com/opengeospatial/ogc-geosparql/issues/590>
- <https://github.com/opengeospatial/ogc-geosparql/issues/574>
- <https://github.com/opengeospatial/ogc-geosparql/issues/571>

10.1.6.2. Category

Semantic improvement

10.1.6.3. Description

GeoSPARQL 3D should be aligned to other vocabularies and standard which currently provide 3D support in different knowledge domains. Especially alignments to ifcOWL and the X3D vocabulary would position the GeoSPARQL vocabulary as a link between these different standards.

10.1.7. Extension 7: Alignments of Engineering CRS to Geospatial CRS

10.1.7.1. GitHub Issue URI

<https://github.com/opengeospatial/ogc-geosparql/issues/586>

10.1.7.2. Category

Semantic improvement

10.1.7.3. Description

GeoSPARQL 3D should provide the opportunity to align a local coordinate system in which most 3D geometries are defined with a coordinate reference. While this work might only be partially done within the scope of GeoSPARQL itself, GeoSPARQL should be aligned with the emerging Ontology CRS developments of OGC and provide necessary functions or properties to create the link.

10.1.8. Extension 8: Geometry Extrusion

10.1.8.1. GitHub Issue URI

- <https://github.com/engeos/geosparql/issues/556>
- <https://github.com/engeos/geosparql/issues/547>

10.1.8.2. Category

Semantic improvement

10.1.8.3. Description

GeoSPARQL 3D should provide the opportunity to extrude 2D geometries to 3D geometries and vice versa.

10.1.9. Extension 9: Geometry Attributes

10.1.9.1. GitHub Issue URI

- <https://github.com/engeos/geosparql/issues/568>
- <https://github.com/engeos/geosparql/issues/550>
- <https://github.com/engeos/geosparql/issues/549>
- <https://github.com/engeos/geosparql/issues/548>
- <https://github.com/engeos/geosparql/issues/558>

10.1.9.2. Category

Semantic improvement

10.1.9.3. Description

GeoSPARQL 3D should provide functions and properties that describe essential properties of a 3D Geometry such as its minimum and maximum height, width and depth and its CompactnessRatio.

10.1.10. Extension 10: Non-topological Query Functions – 3D Extension

10.1.10.1. GitHub Issue URI

- <https://github.com/opengeospatial/ogc-geosparql/issues/556>

10.1.10.2. Category

Semantic improvement

10.1.10.3. Description

GeoSPARQL 3D should provide the opportunity to execute non-topological query functions on 2D and 3D geometries commonly used in geospatial databases. Proposed extensions include following functions:

- geometry extrusion to the specified line segment
- geometry extrusion to the specified height
- spatiotemporal geometry extrusion to the specified line segment with specific start and end time



11

BENEFICIARIES AND BENEFITS

This section describes the beneficiaries and benefits of representing data, including geospatial data, using semantic and graph technologies. Furthermore, a collection of use cases demonstrate how semantic and graph technologies are used together with spatial data to tackle real world problems.

11.1. Beneficiaries

11.1.1. Beneficiary 1: Someone who benefits

11.2. Benefits

The benefits of semantic and graph technologies are outlined below.

11.2.1. Benefit B1: My benefit



12

ANNEX N: N



13

ANNEX O: HISTORY



BIBLIOGRAPHY





BIBLIOGRAPHY

RDF	World Wide Web Consortium, <i>RDF 1.1 Concepts and Abstract Syntax</i> , W3C Recommendation (25 February 2014). https://www.w3.org/TR/rdf11-concepts/
TTL	World Wide Web Consortium, <i>RDF 1.1 Turtle Terse RDF Triple Language</i> , W3C Recommendation (25 February 2014). https://www.w3.org/TR/turtle