

# Testbed-12 TopoJSON, GML Engineering Report

# Table of Contents

1. Introduction .....	6
1.1. Scope .....	6
1.2. Document contributor contact points .....	6
1.3. Future Work .....	6
1.4. Foreword .....	6
2. References .....	7
3. Terms and definitions .....	8
3.1. Arc .....	8
3.2. Client .....	8
3.3. Coordinate .....	8
3.4. Feature .....	8
3.5. GeoJSON .....	8
3.6. Geometry .....	8
3.7. Interface .....	8
3.8. Operation .....	8
3.9. Service .....	9
3.10. TopoJSON .....	9
3.11. Web Feature Service .....	9
3.12. Abbreviated terms .....	9
4. Overview .....	10
5. Background .....	11
5.1. TopoJSON .....	11
5.2. Web Feature Service .....	13
5.3. Filter Encoding .....	14
5.4. Potential Integration Points .....	15
6. TopoJSON Testing .....	16
6.1. Testing Architecture .....	16
6.2. TopoJSON WFS .....	17
6.3. Encodings .....	19
6.3.1. Key Points .....	20
6.3.2. Items to Note .....	21
7. Findings and Recommendations .....	22
7.1. Findings .....	22
7.2. Recommendations .....	22
Appendix A: Revision History .....	23
Appendix B: Bibliography .....	24

Publication Date: 2017-05-15

Approval Date: 2017-02-20

Posted Date: 2016-10-31

Reference number of this document: OGC 16-056

Reference URL for this document: <http://www.opengis.net/doc/PER/t12-A005-3>

Category: Public Engineering Report

Editor: Jeff Harrison

Title: Testbed-12 TopoJSON, GML Engineering Report

---

## **OGC Engineering Report**

### **COPYRIGHT**

Copyright © 2017 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

### **WARNING**

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is 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, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

## 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.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

## **Abstract**

This OGC document evaluates TopoJSON as an encoding that may be delivered across a common, standard OGC service interface such as WFS.

## **Business Value**

The OGC WFS provides an interoperable method to access and update geodata across network-connected components. However, results from previous OGC activities and operational deployments indicate that transferring large volumes of geodata from a WFS over a network with poor or very low bandwidth can take a significant amount of time, and network capacity.

To help meet this challenge OGC Testbed 12 developed prototype implementations and conducted Technology Integration Experiments (TIEs) to assess TopoJSON as an encoding that may be delivered across a common, standard OGC service interface such as WFS. Use of TopoJSON as an output format for WFS may reduce redundancy in feature data representations and decrease the geospatial data file size.

## **What does this ER mean for the Working Group and OGC in general**

TopoJSON is a Javascript Object Notation (JSON) format for encoding geographic data structures in a shared topology. A TopoJSON topology represents one or geometries that share sequences of positions called arcs. TopoJSON, as an extension of GeoJSON, supports multiple geometry types: Point, LineString, Polygon, MultiPoint, MultiLineString, MultiPolygon, and GeometryCollection.

Rather than representing geometries discretely, geometries in TopoJSON files are stitched together from shared line segments called arcs. Arcs are sequences of points, while line strings and polygons are defined as sequences of arcs. Each arc is defined only once, but can be referenced several times by different shapes, thus reducing redundancy and decreasing the geospatial data file size.

A TopoJSON encoding always consists of a single topology object. A topology may contain any number of named geometry objects. The term “TopoJSON object” may refer to either a topology or a geometry object it contains. To include information on the coordinate range for a topology or geometry a TopoJSON object may have a member named “bbox” or Bounding Box.

Given the structure of WFS operations, the WFS OutputFormat and BBox parameters of the Getfeature request may be potential 'integration points' for TopoJSON. The source data for TopoJSON WFS output formats may be common

feature representations currently used on WFS including GML and GeoJSON.

### **How does this ER relate to the work of the Working Group**

If some level of integration of TopoJSON and WFS can be achieved, interoperability for the geospatial community will be enhanced - resulting in more potential data reuse, reduced data and application development costs and enhanced understanding of shared environments represented as both features and arcs.

### **Keywords**

ogcdocs, testbed-12, web services, WFS, GML, JSON, TopoJSON, GeoJSON

### **Proposed OGC Working Group for Review and Approval**

This document will be submitted to the OGC WFS Standards Working Group (SWG) for review and comment.

# Chapter 1. Introduction

## 1.1. Scope

This OGC document evaluates TopoJSON as an encoding that may be delivered across a common, standard OGC service interface such as WFS.

## 1.2. Document contributor contact points

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

*Table 1. Contacts*

Name	Organization
Jeff Harrison	The Carbon Project
Mark Mattson	The Carbon Project

## 1.3. Future Work

This Engineering Report recommends further investigation into TopoJSON as part of an overall API for representing geospatial Resources as GeoJSON. If some level of integration of TopoJSON and WFS can be achieved, interoperability for the geospatial community will be enhanced - resulting in more potential data reuse, reduced data and application development costs and enhanced understanding of shared environments represented as both features and arcs.

## 1.4. Foreword

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.



# Chapter 2. References

The following documents are referenced in 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.

- OGC 06-121r9, OGC® Web Services Common Standard

*NOTE: This OWS Common Standard contains a list of normative references that are also applicable to this report.*

- TopoJSON GitHub <https://github.com/mboostock/topojson>
- The TopoJSON Format Specification, <https://github.com/mboostock/topojson-specification#2-topojson-objects>
- The GeoJSON Format, draft-ietf-geojson-04, <https://tools.ietf.org/html/draft-ietf-geojson-04>
- TopoJSON API Reference, <https://github.com/mboostock/topojson/wiki/API-Reference>
- TopoJSON Wiki, <https://github.com/mboostock/topojson/wiki>
- Open Geospatial Consortium (OGC), Geography Markup Language (GML) Encoding Standard, <http://www.opengeospatial.org/standards/gml>
- Open Geospatial Consortium (OGC), Web Feature Service (WFS), <http://www.opengeospatial.org/standards/wfs>
- Filter Encoding Implementation Specification, <http://www.opengeospatial.org/standards/filter>

# Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9] shall apply. In addition, the following terms and definitions apply.

## 3.1. Arc

Rather than representing geometries discretely, geometries in TopoJSON files are stitched together from shared line segments called arcs.

## 3.2. Client

The software component that can invoke an operation from a service.

## 3.3. Coordinate

One of a sequence of  $n$  numbers designating the position of a point in  $n$ -dimensional space.

## 3.4. Feature

An application object that represents a physical entity, e.g. a building, a river, or a person. A feature may or may not have geometric aspects.

## 3.5. GeoJSON

GeoJSON is an open standard format designed for representing simple geographical features, along with their non-spatial attributes, based on JavaScript Object Notation. The features include points (therefore addresses and locations), line strings (therefore streets, highways and boundaries), polygons (countries, provinces, tracts of land), and multi-part collections of these types.

## 3.6. Geometry

The geometry data type is used to house information on recognized objects, like points, lines, and polygons.

## 3.7. Interface

The named set of operations that characterize the behavior of a service.

## 3.8. Operation

The specification of a transformation or query that a service may be called to execute.

## 3.9. Service

The distinct part of the functionality that is provided by an entity through interfaces.

## 3.10. TopoJSON

TopoJSON is an extension of GeoJSON that encodes topology. As its name implies, TopoJSON is a topological geospatial data format. In contrast to a geometry, where each shape is encoded with separate (and often redundant) arrays of coordinates, a topology encodes sequences of coordinates in line fragments called arcs that can be shared. For example, the border between California and Nevada is an arc that is shared by both polygons. The main benefit of a topology is that it improves shape simplification by avoiding artifacts that would be caused by simplifying shapes independently. It also enables applications like map coloring and selective meshing, and makes the format more compact by removing redundant coordinates.

## 3.11. Web Feature Service

The Web Feature Service standard specifies the behavior of a service that provides transactions on and access to geographic features in a manner independent of the underlying data store.

## 3.12. Abbreviated terms

Some more frequently used abbreviated terms in this document include:

- API: Application Programming Interface
- BBox: Bounding Box
- ER: Engineering Report
- GML: Geography Markup Language
- JSON: Javascript Object Notation
- KVP: Key Value Pair
- OGC: Open Geospatial Consortium
- TIE: Technology Integration Experiment
- WFS: Web Feature Service
- XML: Extensible Markup Language

# Chapter 4. Overview

This document contains the following sections:

- **Preface** - This section presents information on the business value and what this ER means for the WFS Working Group and OGC in general.
- **Introduction** - This section presents information on scope and document contributor contact points.
- **References** - This section presents information on documents that are referenced in this Engineering Report.
- **Background** - This section presents information on the Background of this Testbed 12 thread, including studies within OGC and by external organizations and individuals.
- **TopoJSON Testing** - This section presents information on the techniques, component implementations, the findings of Technology Integration Experiments conducted to assess TopoJSON as an encoding that may be delivered across a common, standard OGC service interface such as WFS.
- **Findings and Recommendations** - This section presents information on the findings of this Testbed 12 investigation.

# Chapter 5. Background

This section provides background information on the potential integration of TopoJSON as an encoding that may be delivered across a common, standard OGC service interface such as WFS.

## 5.1. TopoJSON

This OGC document evaluates TopoJSON as an encoding that may be delivered across a common, standard OGC service interface such as WFS.

TopoJSON is an extension of GeoJSON that encodes topology. Rather than representing geometries discretely, geometries in TopoJSON files are stitched together from shared line segments called arcs. Arcs are sequences of points, while line strings and polygons are defined as sequences of arcs.

Origine\_Point  
"coordinates": [0,0]

Under\_Point  
"coordinates": [0,-1]

Under\_LineString  
"arcs": [3]

Left\_Polygon  
"arcs": [[0,1]]

Right\_Polygon  
"arcs": [[2,-1]]

```
"arcs":[  
  [[1,2],[0,-2]],  
  [[1,0],[-1,0],[0,2][1,0]],  
  [[1,2],[1,0],[0,-2],[-1,0]],  
  [[0,-1],[2,0]]  
]
```

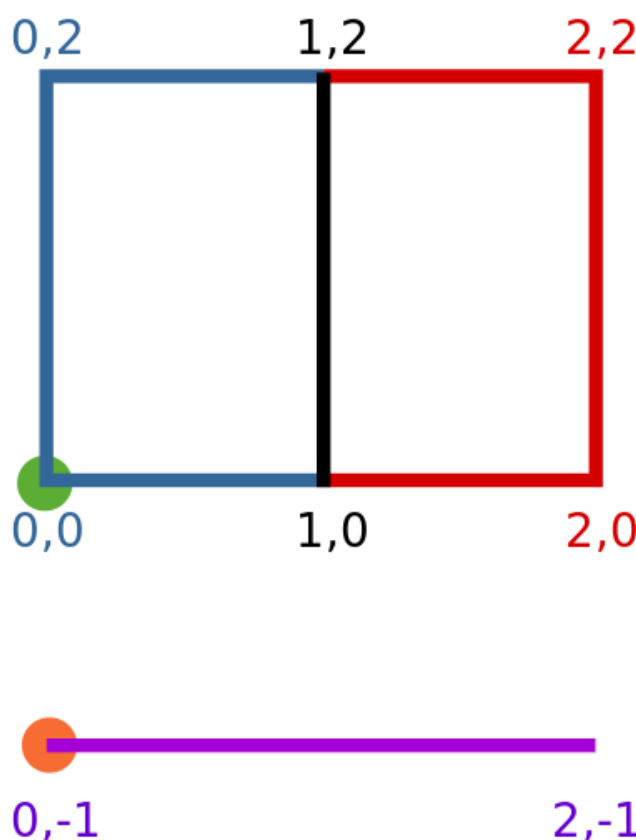


Figure 1. TopoJSON Stitching

An encoding sample from <https://github.com/mbostock/topojson/wiki/Introduction> is shown below.

```

{
  "type": "Topology",
  "transform": {
    "scale": [0.036003600360036005, 0.017361589674592462],
    "translate": [-180, -89.99892578124998]
  },
  "objects": {
    "aruba": {
      "type": "Polygon",
      "arcs": [[0]],
      "id": 533
    }
  },
  "arcs": [
    [[3058, 5901], [0, -2], [-2, 1], [-1, 3], [-2, 3], [0, 3], [1, 1], [1, -3], [2,
-5], [1, -1]]
  ]
}

```

The main benefit of TopoJSON is that each arc is defined only once, but can be referenced several times by different shapes, thus reducing redundancy and decreasing the file size. It also enables applications like map coloring and selective meshing, and makes the format more compact by removing redundant coordinates.

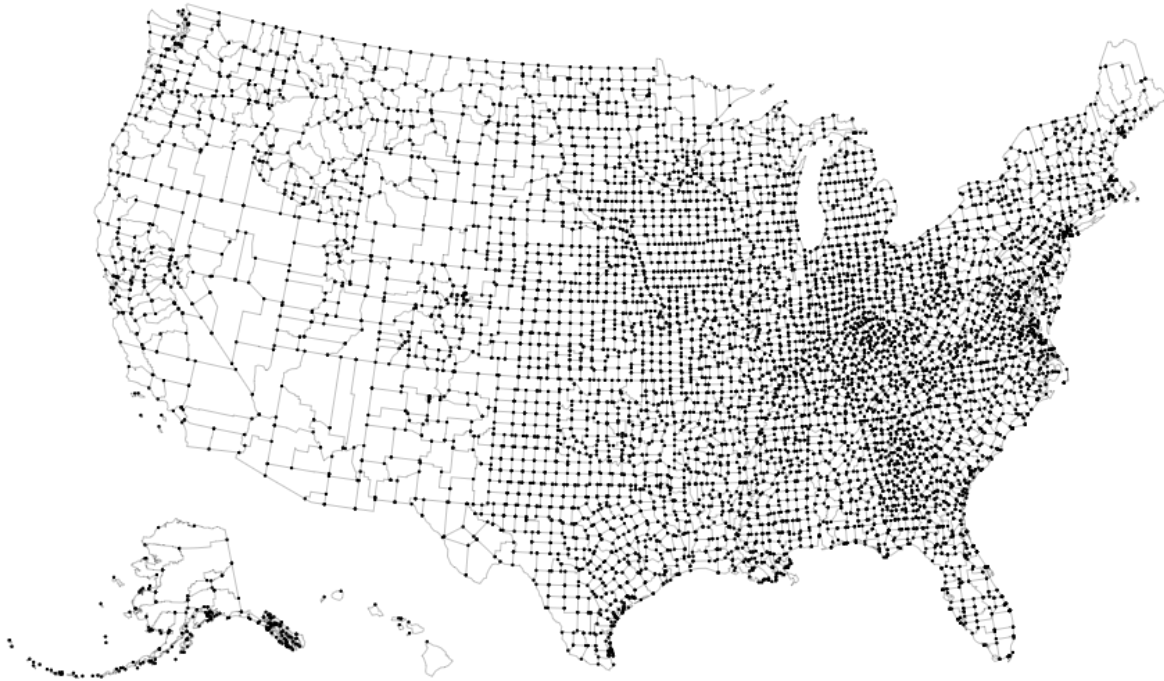


Figure 2. TopoJSON Example

A TopoJSON encoding always consists of a single topology object. A topology may contain any number of named geometry objects. The term TopoJSON object may refer to either a topology or a geometry object it contains. To include information on the coordinate range for a topology or geometry a TopoJSON object may have a member named “bbox” or Bounding Box. A geometry is a TopoJSON object where the type member’s value is one of the following strings: “Point”,

“MultiPoint”, “LineString”, “MultiLineString”, “Polygon”, “MultiPolygon”, or “GeometryCollection”.

A single TopoJSON file can contain multiple feature collections without duplication, such as states and counties. Or, a TopoJSON file can efficiently represent both polygons (for fill) and boundaries (for stroke) as two feature collections that share the same arc mesh. As a result, TopoJSON is substantially more compact than GeoJSON. The shapefile below of U.S. counties is 2.2M as a GeoJSON file, 436K as a boundary mesh, a reduction of 80.4% even without simplification. TopoJSON can also be more efficient to render since shared control points need only be projected once. To further reduce file size, TopoJSON uses fixed-precision delta-encoding for integer coordinates rather than floats. Like GeoJSON, TopoJSON files are easily modified in a text editor and amenable to gzip compression.



*Figure 3. Example Used to Show TopoJSON Saving Compared to GeoJSON*

In TopoJSON the Bounding Box member includes information on the coordinate range for a topology or geometry. The value of the bbox member must be a  $2*n$  array where  $n$  is the number of dimensions represented in the contained geometries, with the lowest values for all axes followed by the highest values. The axes order of a bbox follows the axes order of geometries.

## 5.2. Web Feature Service

The OGC Web Feature Service (WFS) Implementation Specification allows a client to retrieve geospatial data encoded in Geography Markup Language (GML) and other formats from multiple Web Feature Services. The specification defines operations for data access and manipulation operations on geographic features, using HTTP as the distributed computing platform. Via these interfaces, a Web user or service can combine, use and manage geodata — the feature information behind a map image.

This International Standard specifies the behavior of a service that provides transactions on and access to geographic features in a manner independent of the underlying data store. It specifies discovery operations, query operations, locking operations, transaction operations and operations to manage stored parameterized query expressions:

- Discovery operations allow the service to be interrogated to determine its capabilities and to retrieve the application schema that defines the feature types that the service offers.
- Query operations allow features or values of feature properties to be retrieved from the underlying data store based upon constraints, defined by the client, on feature properties.
- Locking operations allow exclusive access to features for the purpose of modifying or deleting features.
- Transaction operations allow features to be created, changed, replaced and deleted from the underlying data store.
- Stored query operations allow clients to create, drop, list and described parameterized query expressions that are stored by the server and can be repeatedly invoked using different parameter values.

This International Standard defines eleven operations:

- GetCapabilities (discovery operation)
- DescribeFeatureType (discovery operation)
- GetPropertyValue (query operation)
- GetFeature (query operation)
- GetFeatureWithLock (query & locking operation)
- LockFeature (locking operation)
- Transaction (transaction operation)
- CreateStoredQuery (stored query operation)
- DropStoredQuery (stored query operation)
- ListStoredQueries (stored query operation)

Some WFS servers may also support additional non-GML feature encodings and client applications may access them using the outputFormat parameter domains. However, the WFS International Standard does not describe how a server would operate upon such encodings. This is an important distinction for TopoJSON interoperability testing, demonstration and operational implementation.

## 5.3. Filter Encoding

The OGC Filter Encoding Implementation Specification describes an XML and KVP encoding of a system neutral syntax for expressing projections, selection and sorting clauses collectively called a ‘query expression’. As background, a fundamental operation performed on a set of data or resources is that of querying in order to obtain a subset of the data which contains certain desired information that satisfies some query criteria and which is also, perhaps, sorted in some specified manner.

This International Standard defines the XML encoding for the following predicates.

- A standard set of logical predicates: and, or and not.
- A standard set of comparison predicates: equal to, not equal to, less than, less than or equal to,



greater than, greater than or equal to, like, is null and between.

- A standard set of spatial predicates: equal, disjoint, touches, within, overlaps, crosses, intersects, contains, within a specified distance, beyond a specified distance and BBOX.
- A standard set of temporal predicates: after, before, begins, begun by, contains, during, ends, equals, meets, met by, overlaps and overlapped by.
- A predicate to test whether the identifier of an object matches the specified value.

## 5.4. Potential Integration Points

Given the structure of WFS operations, the WFS OutputFormat and BBox parameters of the Getfeature request may be potential 'integration points' for TopoJSON. The source data for TopoJSON WFS output formats may be common feature representations currently used on WFS including GML and GeoJSON.

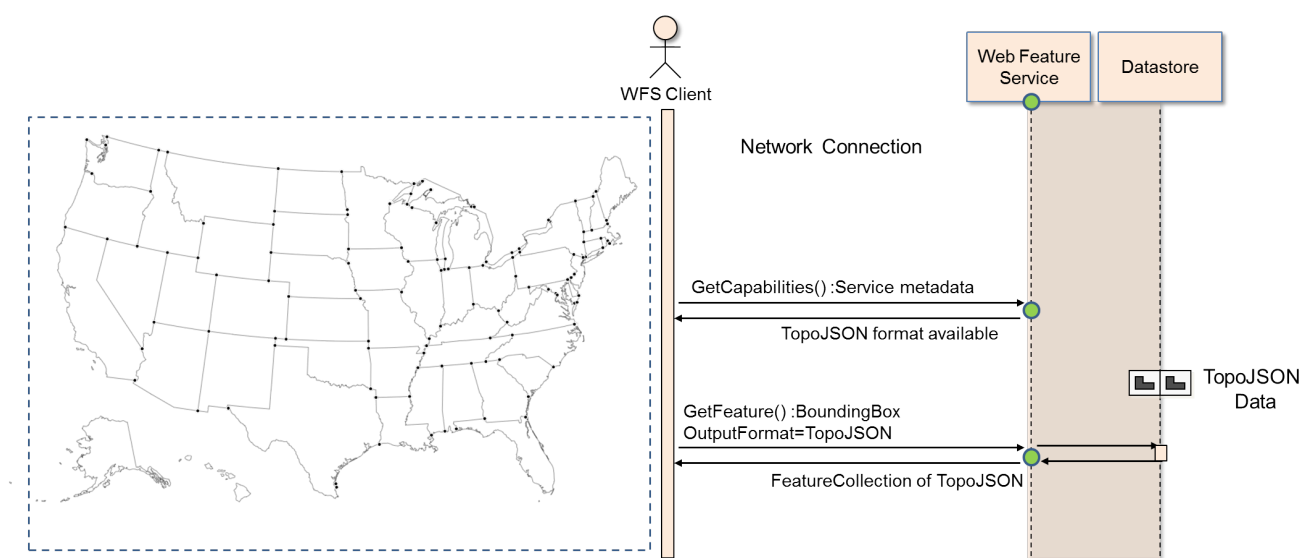


Figure 4. Potential Integration Points

The optional WFS `outputFormat` parameter specifies the format used to encode feature data in the response to a query operation. The default value is `"application/gml+xml; version=3.2"` indicating that feature data in the response document shall be encoded using GML (see ISO 19136:2007). Every WFS that conforms to the WFS International Standard must support this default value.

However, a WFS may advertise additional values for the `outputFormat` parameter in its capabilities document indicating that multiple output formats, including previous versions of GML and other encodings like TopoJSON, are supported.

# Chapter 6. TopoJSON Testing

This section provides an analysis of the prototype implementations, various approaches, pros and cons and performance aspects of data size reduction, compression techniques explored in OGC Testbed 12 and findings.

This section presents information on -

- Testing Architecture
- TopoJSON WFS
- Encodings

## 6.1. Testing Architecture

Given the structure of WFS operations, the WFS OutputFormat and BBox parameters of the Getfeature request were used as integration points for TopoJSON.

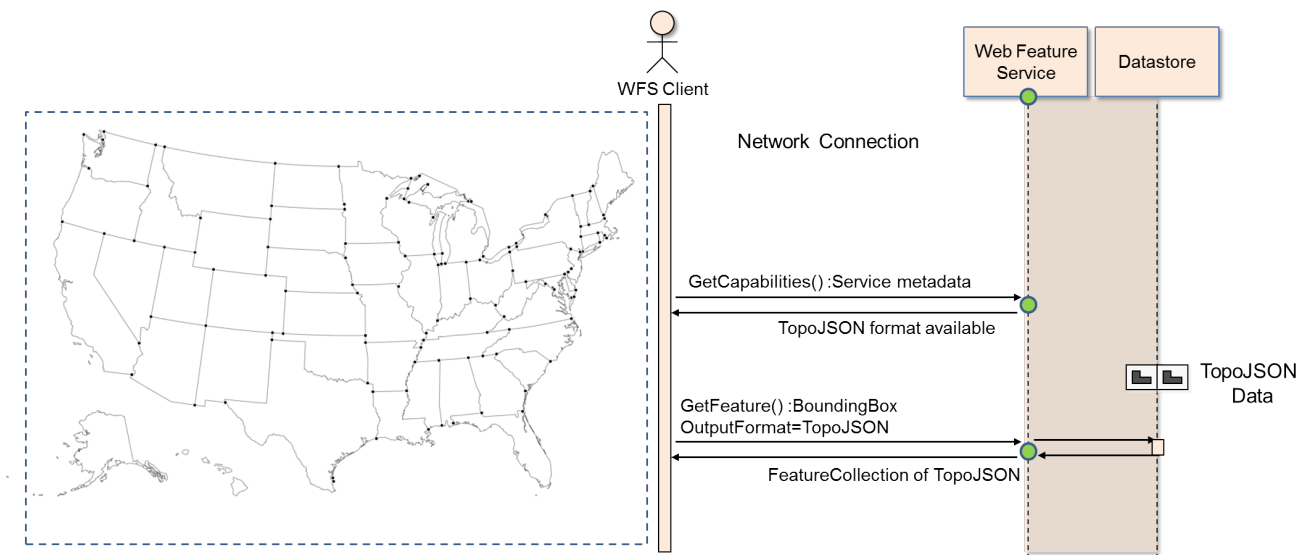


Figure 5. Integration Points for TopoJSON

TopoJSON WFS output format may augment common feature representations currently used on WFS such as GeoJSON, shown below.

```

{
  "type": "FeatureCollection",
  "features": [{"type": "Feature",
    "_id": "1",
    "geometry": { "type": "MultiPolygon",
      "coordinates": [ [ [ [-122.412693526847, 37.7594367982446], [-122.412661937571,
        37.7591065315166], [-122.413141300785, 37.7590776043661], [-122.413208208506,
        37.7590735647947], [-122.413456701731, 37.7590585663082], [-122.413969677293,
        37.7590276048958], [-122.413994321968, 37.7592877755075], [-122.414080558708,
        37.7592825720693], [-122.414166788208, 37.7592773630598], [-122.414253017841,
        37.7592721596105], [-122.414580701496, 37.7592523789956], [-122.414632974612,
        37.7598041810355], [-122.414210412616, 37.7598294125848], [-122.413791635565,
        37.7598544180112], [-122.413584677147, 37.7598669088383], [-122.413594004952,
        37.7599643912657], [-122.412748990251, 37.7600167225604], [-122.412693526847,
        37.7594367982446] ] ] ] },
    "properties": {
      "cp:LAND_ID": "0000050203",
      "cp:LAND_NAME": "O'Connell, John High School",
      "cp:CATEGORY": "Unified School District",
      "cp:CITY_OWNED": "1",
      "cp:DEPT": 7,
      "cp:SCHOOL_TYP": "Grades 9-12"}
  }
}

```

The optional WFS outputFormat parameter specifies the format used to encode feature data in the response to a query operation. The default value is "application/gml+xml; version=3.2" indicating that feature data in the response document shall be encoded using GML (see ISO 19136:2007). Every WFS that conforms to the WFS International Standard must support this default value.

A WFS may advertise additional values for the outputFormat parameter in its capabilities document indicating that multiple output formats, including previous versions of GML and other encodings like TopoJSON, are supported.

## 6.2. TopoJSON WFS

This section described the WFS implemented to satisfy the TopoJSON WFS requirement for Testbed 12.

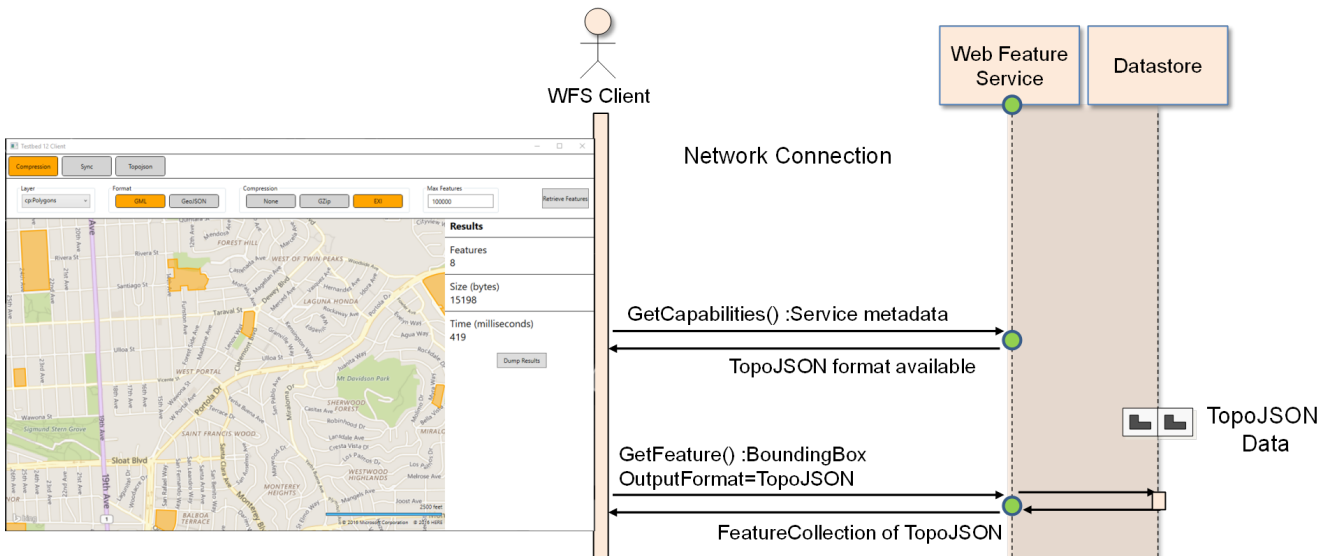
This experiment was performed using one WFS from The Carbon Project. The table below lists the product version and endpoint of the server.

Vendor	Product	Supported Versions	Endpoint
The Carbon Project	CarbonCloud	1.0.0+	<a href="http://ows12.azurewebsites.net/wfs?service=WFS&amp;request=GetFeature&amp;typeName=cp:school_poly&amp;maxFeatures=3&amp;outputFormat=topojson">http://ows12.azurewebsites.net/wfs?service=WFS&amp;request=GetFeature&amp;typeName=cp:school_poly&amp;maxFeatures=3&amp;outputFormat=topojson</a>

Capabilities of the prototype CarbonCloud server for TopoJSON are listed below.

<b>Property</b>	<b>CarbonCloud</b>
Versions	1.0.0+
Operations	GetCapabilities DescribeFeatureType GetFeature Sync Transaction GET PUT POST DELETE
Output Formats	GML v3.2 GML v3.1.1 GML v2.1.2 JSON KML CSV TopoJSON GZIP EXI
Spatial Operators	Disjoint Equals Intersects Touches Crosses Contains Overlaps BBOX Within Beyond
Spatial Operands	gml:Envelope gml:Point gml:LineString gml:Curve gml:Polygon gml:Surface gml:MultiPoint gml:MultiLineString gml:MultiCurve gml:MultiPolygon gml:MultiSurface
Scalar Operators	PropertyIsBetween PropertyIsEqualTo PropertyIsGreaterThan PropertyIsGreaterThanOrEqualTo PropertyIsLessThan PropertyIsLessThanOrEqualTo PropertyIsLike PropertyIsNotEqualTo
Logical	And, Or
Number of CRSs	> 10
GMLSF Support	Yes
REST Support	Yes
GeoJSON Support	Yes
TopoJSON Support	Yes
Compression Support	Yes

The Carbon Project's prototype client supporting Sync, Compression and TopoJSON for Testbed 12 is shown below.



### 6.3. Encodings

In Testbed 12 the ability of WFS to support TopoJSON outoutformat was assessed by loading test data, including polygons representing schools in San Francisco, converting the test data to TopoJSON in response to a properly formatted request sent to the Endpoint listed above.

The resulting TopoJSON encoding returned by the WFS is shown below.

```

{
  "type": "Topology",
  "transform": {
    "scale": [3.6000036000036E-06, 1.73646866468665E-06],
    "translate": [-180, -89.99892578125]
  },
  "objects": {
    "feature1": {
      "type": "MultiPolygon",
      "arcs": [0]},
    "feature2": { "type": "MultiPolygon","arcs": [1]},
    "feature3": { "type": "MultiPolygon",
      "arcs": [2]}},
    "arcs": [[[ [15996458, 73573664], [9, -190], [-133, -17], [-19, -2], [-69, -9],
[-142, -17], [-7, 149], [-24, -3], [-24, -3], [-24, -3], [-91, -11], [-15, 318], [118,
14], [116, 15], [57, 7], [-2, 56], [235, 30], [15, -334]] ], [[ [16005579,
73558925], [-12, 14], [75, 165], [-76, 92], [-53, -11], [-28, 33], [-71, -18], [-63,
-146], [8, -107], [31, -36], [18, -63], [77, -92], [-6, -12], [-308, 359], [-3, 5], [-
2, 3], [-2, 4], [-2, 4], [-1, 4], [-2, 4], [-2, 3], [-2, 4], [-1, 4], [-2, 4], [-1,
4], [-2, 5], [-1, 4], [-2, 4], [-1, 4], [-2, 5], [-1, 4], [-1, 4], [-1, 5], [-1, 4],
[-2, 5], [-1, 4], [-1, 5], [-1, 4], [0, 5], [-1, 4], [-1, 5], [-1, 5], [-1, 4], [0,
5], [-1, 5], [-27, 325], [-15, 181], [57, 60], [17, -28], [507, -592], [1, -35], [0,
-1], [0, 0], [0, -1], [0, -1], [0, -1], [0, 0], [0, -1], [0, -1], [0, 0], [0, -1], [0,
-1], [0, 0], [-1, -1], [0, -1], [0, 0], [0, -1], [0, -1], [0, 0], [0, -1], [0, -1],
[0, 0], [-1, -1], [0, -1], [0, 0], [0, -1], [0, -1], [0, 0], [-1, -1], [0, 0], [0,
-1], [0, -1], [0, 0], [-1, -1], [0, 0], [0, -1], [0, -1], [0, 0], [-86, -198]] ]], [[
[15992141, 73571082], [-9, 180], [-1, 0], [-9, 177], [98, 12], [16, -357], [-95,
-12]] ]]]
}

```

### 6.3.1. Key Points

1. Geometry objects in TopoJSON WFS are similar to those in GeoJSON WFS, except that a TopoJSON geometry object defines its coordinates as a sequence of the containing topology's arcs, as referenced by zero-based index. For example, a MultiPolygon might be defined as

```
{ "type": "MultiPolygon", "arcs": [2]}
```

1. The topology's `arcs` array is, in effect, a multi-line string. As in GeoJSON, each point has at least two dimensions: *x* and *y*. However, unlike GeoJSON, each coordinate in TopoJSON is represented as an *integer* value relative to the previous point. The first point is relative to the origin, `0,0`. To convert to latitude and longitude (or absolute coordinates), the topology defines a linear `transform` consisting of a scale and translate. For example:

```
"transform": {
  "scale": [3.6000036000036E-06, 1.73646866468665E-06],
  "translate": [-180, -89.99892578125]
},
```

1. Arc coordinates may have additional dimensions beyond  $x$  and  $y$ ; the meaning of these dimensions is outside the scope of this specification. For example, to support for dynamic simplification at different zoom levels, the visual importance of each point as computed by a simplification algorithm (e.g., Visvalingham) may be stored along with each point so that arcs can be rapidly filtered as needed.
2. While GeoJSON is agnostic about winding order, TopoJSON recommends that all sub-hemisphere polygons be in clockwise winding order; counterclockwise winding order indicates the polygon that covers an area greater than one hemisphere.
3. TopoJSON does not support GeoJSON features, nor feature collections. Instead, these objects are converted to their respective geometries and geometry collections. In addition, TopoJSON allows optional identifiers (`id`) and properties to be stored directly on geometry objects.

Participants would like to give credit to...

<https://github.com/mbstock/topojson/wiki/Introduction>

...for providing guidance while introducing TopoJSON Key Findings to the OGC Community in references 1., 2., 3., 4., and 5. above. Results of the WFS GetFeature Request were mapped to the examples in 'TopoJSON Introduction' and, largely, fit.

### 6.3.2. Items to Note

Initial items to note based on issuing TopoJSON outoutformat requests to the test WFS are included in the Findings section. It is anticipated that these will be updated.

# Chapter 7. Findings and Recommendations

This OGC document evaluates TopoJSON as an encoding that may be delivered across a common, standard OGC service interface such as WFS. The source data for TopoJSON WFS output formats are common feature representations currently used on WFS including GML and GeoJSON.

This section presents information on the findings of Technology Integration Experiments.

## 7.1. Findings

Technology Integration Experiments conducted in Testbed 12 indicate -

1. It is possible to produce an encoding of TopoJSON on WFS via the OutputFormat parameter of the GetFeature request.
2. It is possible for a WFS to advertise additional values for the outputFormat parameter in its capabilities document indicating that multiple output formats, including encodings like TopoJSON, are supported.
3. Although a 'GetFeatures' operation is used to access the outputFormat parameter for TopoJSON, the result set is technically an arc, not a feature.
4. Methods for calculating the scale are uncertain at the time of this report draft. Participants believe it is the max width of the Projection / 9999999 or some other factor for Longitude.
5. Because Participants didn't know how to make the scale, the current scale in the output TopoJSON is hard coded and only works with 4326 projection.
6. It's possible to add edge detection and other things to make the results set smaller. This would only help in certain data scenarios (stuff that has a lot of adjacent geometries). Participants are currently not doing that.

## 7.2. Recommendations

TopoJSON is an extension of GeoJSON that encodes topology, mainly for size reduction purposes. Feature data like interconnected political regions represented as polygons can gain size reduction advantages using TopoJSON. However, processing TopoJSON may require different client tools than GeoJSON, and the format is often converted into GeoJSON after transfer.

It is also to note that although a 'GetFeatures' operation is used to access the outputFormat parameter for TopoJSON, the result set is technically an arc, not a feature.

However, the overall movement in the open geospatial community is towards representing geospatial assets as Resources instead of Services. JSON is a key encoding format for future geospatial Resources.

Accordingly, this Engineering Report recommends further investigation into TopoJSON as part of an overall API for representing geospatial Resources as GeoJSON. If some level of integration of TopoJSON and WFS can be achieved, interoperability for the geospatial community will be enhanced - resulting in more potential data reuse, reduced data and application development costs and enhanced understanding of shared environments represented as both features and arcs.



# Appendix A: Revision History

Table 2. Revision History

<b>Date</b>	<b>Editor</b>	<b>Release</b>	<b>Primary clauses modified</b>	<b>Descriptions</b>
April 15, 2016	Jeff Harrison	0.01	All	Initial draft version
June 30, 2016	Jeff Harrison	0.2	All	First draft version
September 30, 2016	Jeff Harrison	0.3	All	Second draft version
October 24, 2016	Jeff Harrison	0.4	Integrated comments from IPTeam review	Third draft version
October 31, 2016	Jeff Harrison	0.5	Editing for format and presentation	Fourth draft version

# Appendix B: Bibliography

- [1] Web: The TopoJSON Format Specification, <https://github.com/mboostock/topojson-specification#2-topojson-objects>
- [2] Web: Open Geospatial Consortium (OGC), Web Feature Service (WFS), <http://www.opengeospatial.org/standards/wfs>
- [3] Web: Filter Encoding Implementation Specification, <http://www.opengeospatial.org/standards/filter>
- [4] Web: Open Geospatial Consortium (OGC), Geography Markup Language (GML) Encoding Standard, <http://www.opengeospatial.org/standards/gml>
- [5] Web: The GeoJSON Format, draft-ietf-geojson-04, <https://tools.ietf.org/html/draft-ietf-geojson-04>
- [6] Web: TopoJSON API Reference, <https://github.com/mboostock/topojson/wiki/API-Reference>
- [7] Web: TopoJSON Wiki, <https://github.com/mboostock/topojson/wiki>