Open Geospatial Consortium

Approval Date: 2013-01-18

Posted Date: 2013-02-05

Reference number of this document: OGC 12-103r3

Reference URN for this document:
http://www.opengis.net/def/doc-type/per/ows9-cci-semantic-mediation

Category: Public Engineering Report

Editor(s): Gobe Hobona, Roger Brackin

OGC® OWS-9 CCI Semantic Mediation Engineering Report

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.

Document type:	OGC® Engineering Report
Document subtype:	NA
Document stage:	Approved for public release
Document language:	English

Abstract

The OWS-9 Cross Community Interoperability (CCI) thread built on progress made in the recent OWS-8 initiative by improving interoperability between communities sharing geospatial data through advances in semantic mediation approaches for data discovery, access and use of heterogeneous data models and heterogeneous metadata models. This OGC engineering report aims to present findings from CCI thread activities towards advancement of semantic mediation involving heterogeneous data models, gazetteers and aviation data available through web services conformant to OGC standards.

This Engineering Report was prepared as a deliverable for the OGC Web Services, Phase 9 (OWS-9) initiative of the OGC Interoperability Program. The document presents the work completed with respect to the Cross Community Interoperability thread within OWS-9.

Keywords

ogcdoc, ows9, cci, semantic mediation, ogc web services, wps, wfs-g, sparql, gazetteer, ontologies, semantics

What is OGC Web Services 9 (OWS-9)?

OWS-9 builds on the outcomes of prior OGC interoperability initiatives and is organized around the following threads:

- **Aviation**: Develop and demonstrate the use of the Aeronautical Information Exchange Model (AIXM) and the Weather Exchange Model (WXXM) in an OGC Web Services environment, focusing on support for several Single European Sky ATM Research (SESAR) project requirements as well as FAA (US Federal Aviation Administration) Aeronautical Information Management (AIM) and Aircraft Access to SWIM (System Wide Information Management) (AAtS) requirements.

- **Cross-Community Interoperability (CCI)**: Build on the CCI work accomplished in OWS–8 by increasing interoperability within communities sharing geospatial data, focusing on semantic mediation, query results delivery, data provenance and quality and Single Point of Entry Global Gazetteer.

- **Security and Services Interoperability (SSI)**: Investigate 5 main activities: Security Management, OGC Geography Markup Language (GML) Encoding Standard Application Schema UGAS (UML to GML Application Schema) Updates, Web Services Façade, Reference Architecture Profiling, and Bulk Data Transfer.

- **OWS Innovations**: Explore topics that represent either new areas of work for the Consortium (such as GPS and Mobile Applications), a desire for new approaches to existing technologies to solve new challenges (such as the OGC Web Coverage Service (WCS) work), or some combination of the two.

- **Compliance & Interoperability Testing & Evaluation (CITE)**: Develop a suite of compliance test scripts for testing and validation of products with interfaces implementing the following OGC standards: Web Map Service (WMS) 1.3 Interface Standard, Web Feature Service (WFS) 2.0 Interface Standard, Geography Markup Language (GML) 3.2.1 Encoding Standard, OWS Context 1.0 (candidate encoding standard), Sensor Web Enablement (SWE) standards, Web Coverage Service for Earth Observation (WCS-EO) 1.0 Interface Standard, and TEAM (Test, Evaluation, And Measurement) Engine Capabilities.

The OWS-9 sponsors are: AGC (Army Geospatial Center, US Army Corps of Engineers), CREAF-GeoViQua-EC, EUROCONTROL, FAA (US Federal Aviation Administration), GeoConnections - Natural Resources Canada, Lockheed Martin Corporation, NASA (US National Aeronautics and Space Administration), NGA (US National Geospatial-Intelligence Agency), USGS (US Geological Survey), UK DSTL (UK MoD Defence Science and Technology Laboratory).

Contents

Page

Figures

Page

Tables Page

OGC® OWS-9 CCI Semantic Mediation Engineering Report

1 Introduction

1.1 Scope

The OWS-9 Cross Community Interoperability (CCI) thread built on progress made in the recent OWS-8 initiative by improving interoperability between communities sharing geospatial data through advances in semantic mediation approaches for data discovery, access and use of heterogeneous data models and heterogeneous metadata models. This OGC engineering report aims to present findings from CCI thread activities towards advancement of semantic mediation involving heterogeneous data models, gazetteers and aviation data available through web services conformant to OGC standards.

The engineering report briefly introduce relevant details of the semantic interoperability and mediation. The document will make recommendations on the advancement of the semantic mediation architecture developed in the previous OGC web service (OWS) testbed. Based on the scenario adopted by the CCI thread, the document will also discuss the pros and cons of adopting relevant standards. The engineering report will offer recommendations on how specific OGC standards may be adopted or modified in order to support semantic mediation.

1.2 Document contributor contact points

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

Name	Organization
Gobe Hobona	Envitia Ltd.
Roger Brackin	Envitia Ltd.
Luis Bermudez	Open Geospatial Consortium
Doug Caldwell	U.S. Army Corps of Engineers
Brett Parsons	Compusult
Perry Peterson	Pyxis Innovation
Idan Shatz	Pyxis Innovation

Robin Houtmeyers	Luciad
Mattia Santoro	CNR-IIA
Fabrizio Papeschi	CNR-IIA
Stefano Nativi	CNR-IIA
Max Craglia	EC-JRC

1.3 Revision history

Date	Release	Editor	Primary clauses modified	Description
2012/11/1	0.0.1	GH	All	Initial version

1.4 Future work

None planned.

1.5 Forward

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.

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 06-121r3, *OpenGIS® Web Services Common Standard*

OGC 09-138, *Fusion Standards Study Engineering Report*

OGC 10-184, *Fusion Standards Study, Phase 2 Engineering Report*

OGC 11-063r6, *OWS-8 CCI Semantic Mediation Engineering Report*

OGC 08-167r1*, Semantic annotations in OGC standards Discussion Paper*

OGC 07-006r1, *OpenGIS® Catalogue Services Specification 2.0.2*

OGC 12-104, *OWS-9 Engineering Report - CCI - Single Point of Entry Global Gazetteer*

OGC 12-151, *OGC OWS-9 Aviation Architecture Engineering Report*

ISO TC 211 N 2705*, Report from stage 0 Project 19150 Geographic information – Ontology*

3 Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 06-121r3] and in OpenGIS® Abstract Specification shall apply. In addition, the following terms and definitions apply.

3.1
feature
representation of some real world object or phenomenon

3.2
data fusion
the act or process of combining or associating data or information regarding one or more entities considered in an explicit or implicit knowledge framework to improve one's capability (or provide a new capability) for detection, identification, or characterization of that entity

3.3
metadata
data about data.

3.4
model
abstraction of some aspects of a universe of discourse [ISO 19109]

3.5
interoperability
capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units [ISO 19119]

3.6
syntactic interoperability
the aspect of interoperability that assures that there is a technical connection, i.e. that the data can be transferred between systems

3.7
semantic interoperability
the aspect of interoperability that assures that the content is understood in the same way in both systems, including by those humans interacting with the systems in a given context

4 Conventions

4.1 Abbreviated terms

ER	Engineering Report
GML	Geography Markup Language
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OWL	Web Ontology Language
OWS	OGC Web Service
OWS-8	OGC Web Services Initiative, Phase 8
OWS-9	OGC Web Services Initiative, Phase 9
RDF	Resource Description Framework
SDI	Spatial Data Infrastructure
SKOS	Simple Knowledge Organization System
SOA	Service Oriented Architecture
SRS	Spatial Reference System
URL	Uniform Resource Locator
URN	Uniform Resource Names

WFS	Web Feature Service
WMS	Web Map Service
WS	Web Service
XML	eXtensible Markup Language

5 Background Information

In this section a review of relevant recent works and standards is presented.

5.1 OWS-8 CCI Semantic Mediation

In this section the service architecture adopted in the OWS-8 CCI thread is presented. As illustrated in Figure 1, the source data for the CCI thread was the NGA LTDS and USGS TNM products. Each source dataset was provided through a web feature service (WFS). The services are registered in a catalogue service where each service is associated with a domain ontology. These WFS services were invoked via the OGC mediation component. The mediation component not only is a WFS and CSW client, but also implements a WFS and WMS interface. The mediation component, as a CSW client, accesses a symbology registry to generate maps based on the feature type, the symbols, and rules registered for those feature types. The mediation component also translates between instances of domain models in GML and RDF, queries a knowledge base and integrates the results in a map with proper styles. The knowledge base contains the common model (Rosetta Mediation Model, RMM), ontologies representing each data model, mappings from each data model to RMM, and rules.

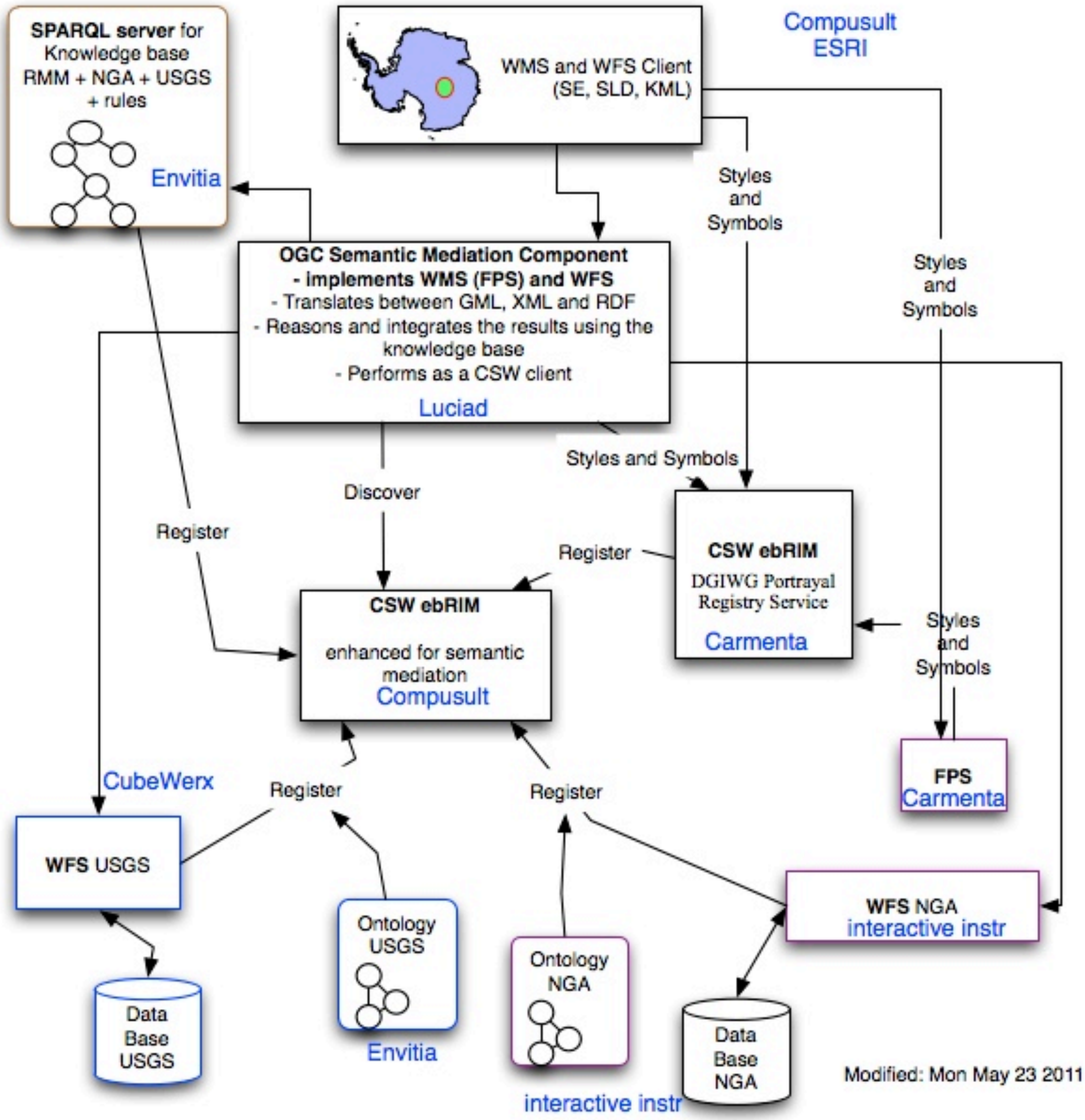

Figure 1. OWS-8 CCI Architecture

Within this architecture, the Web GUI Clients provide the interface between the user and the rest of the system. The CSW is the principal component for resource discovery. The WFS provide vector data supplied by the NGA and USGS. The semantic mediator harmonizes data retrieved from the WFS. The FPS renders data obtained from the semantic mediator and the WFS services. The knowledge base is the collection of RDF-encoded documents generated from mapping specifications and the RMM, TNM and TDS data models. The SPARQL server is the web application that offers a SPARQL interface for the querying of the RDF-encoded knowledge base.

The OWS-8 CCI thread was able to develop RDF-encoded ontologies of the NGA TDS and USGS TNM data models and a mediating model called the RMM-S. A semantic mediation component that allows a client to query multiple data sources was implemented and offered through an interface conformant to the OGC Web Feature Service standard. Mappings from TNM-to-RMM-to-TDS and others from TDS-to-RMM-to-TNM were developed. Mappings between different representations of the same coordinate reference systems were defined. Finally, the mappings were applied in a demonstrator based on the OWS-8 CCI architecture. It can therefore be concluded that OGC standards can successfully support semantic mediation through the architecture proposed in the OWS-8 CFP.

6 Data Sources

The following data sources were provided for the CCI thread.

6.1 NGA GeoNET Names Server

The GeoNET Names Server (GNS), managed by the NGA, serves names for areas outside the United States and its dependent areas, as well as names for undersea features.

6.2 USGS Geographic Names Information System

The Geographic Names Information System (GNIS), managed by USGS, currently contains information about domestic U.S. placenames and Antarctic names.

6.3 UCSB Alexandria Digital Library

The Alexandria Digital Library (ADL) gazetteer developed by the University of California in Santa Barbara (UCSB) provides gazetteer objects and a hierarchical type scheme for geographic features.

6.4 NGA Topographic Data Store

OWS-9 adopted the Local Topographic Data Store (TDS), developed by the NGA as part of the National System for GEOINT (NSG). The Local TDS contains primarily topographic features that are typically extracted to the Local level at 1:50K and 1:100K scales.

6.5 USGS The National Map

OWS-9 also adopted data from The National Map (TNM), which is a collaborative effort among the USGS and other Federal, State, local, and Tribal partners to improve and deliver topographic information for the Nation. With applications ranging from recreation to scientific analysis to emergency response, the National Map is one of the most comprehensive topographic data products available.

7 OWS-9 CCI Architecture

The architecture adopted for the OWS-9 CCI thread is presented in this section. The architecture was implemented by integrating the following services:

- 52°North Conflation WPS
- Compusult CSW
- CubeWerx WFS publishing the USGS national map data (TNM)
- CubeWerx WFS-G publishing the USGS GNIS gazetteer
- CubeWerx cascading WFS-G providing a channel to both the NGA and USGS WFS-Gs
- Envitia SPARQL Server publishing the Knowledge Base of semantic mappings and ontologies
- Envitia WPS Semantic Mediator for Aviation
- Envitia WPS for Geocoding and Geoparsing Volunteered Geographic Information (VGI)
- Envitia WFS-G Semantic Mediator providing a semantically mediated channel to both the NGA and USGS WFS-Gs
- CNR/JRC GEO Discovery and Access Broker (DAB) for mediating between NGA TDS and USGS TNM data
- Interactive Instruments GmbH (ii) WFS supplying NGA Topographic Data Store (TDS)
- Secure Dimensions PEP for WFS NGA TDS
- Secure Dimensions PEP for WFS VGI hosted by Cubewerx
- Secure Dimensions PEP for WFS MINUSTAH hosted by Cubewerx
- Secure Dimensions PEP for WFS USGS hosted by the Carbon Project
- Secure Dimensions PEP for CCI CSW hosted by Compusult

- Intergraph WFS-G NGA GNS
- OpenGeo VGI WFS for publishing user generated content from Twitter, Open Street Map and Ushahidi
- GMU CSW for Provenance

Interoperability between the components was supported by the following encodings:

- OWS Context Document provided by CREAF and Envitia
- Conflation Rules provided by GMU
- Information Models and Encodings provided by Envitia and ii
- Metadata encodings provided by GMU

This engineering report is concerned with a subset of the above listed components. However, related CCI engineering reports cover the other components listed.

An illustration of the geonames components is presented in Figure 2. A discussion of this aspect of the architecture is presented in section 10.

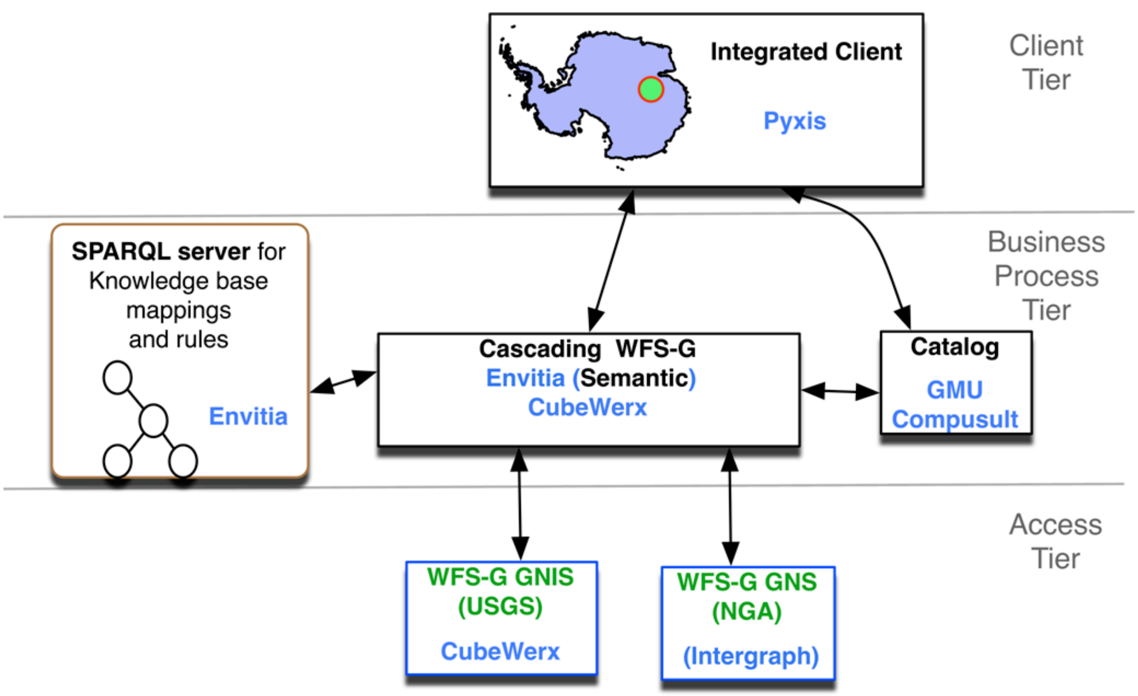

Figure 2. OWS-9 CCI Geonames semantic mediation architecture

An illustration of the VGI geoparsing components is presented in Figure 3. A discussion of this aspect of the architecture is presented in section 12.

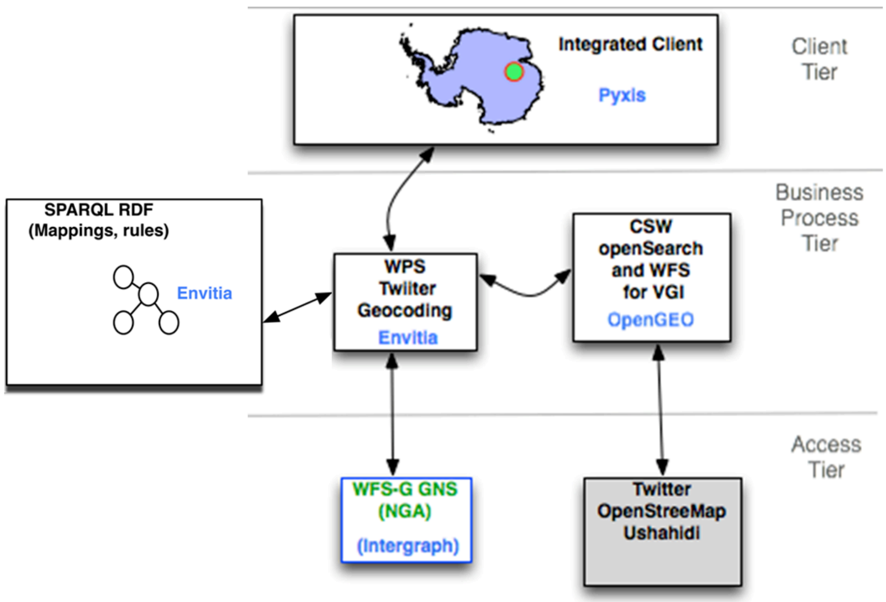

Figure 3. OWS-9 CCI VGI Geocoding and geoparsing architecture

An illustration of the CCI Aviation semantic mediation architecture is presented in Figure 4. A discussion of this aspect of the architecture is presented in section 11.

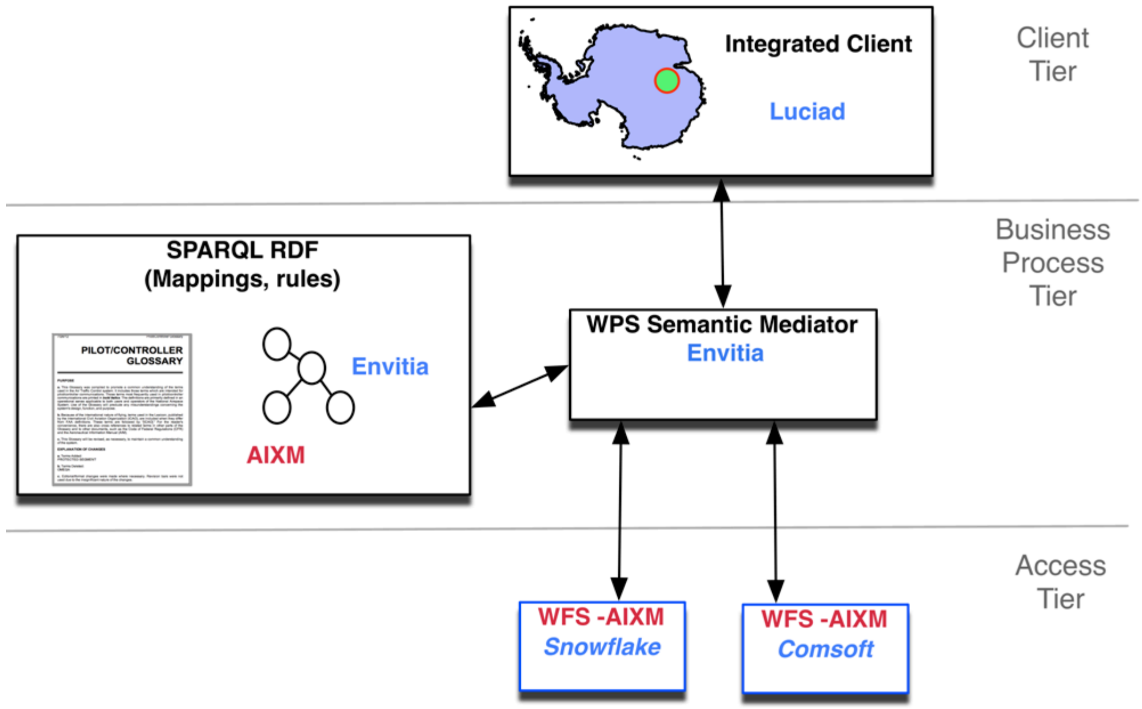

Figure 4. OWS-9 CCI Aviation semantic mediation architecture

An illustration of the conflation and broker aspects of the architecture is presented on Figure 5 and discussed in section 13.

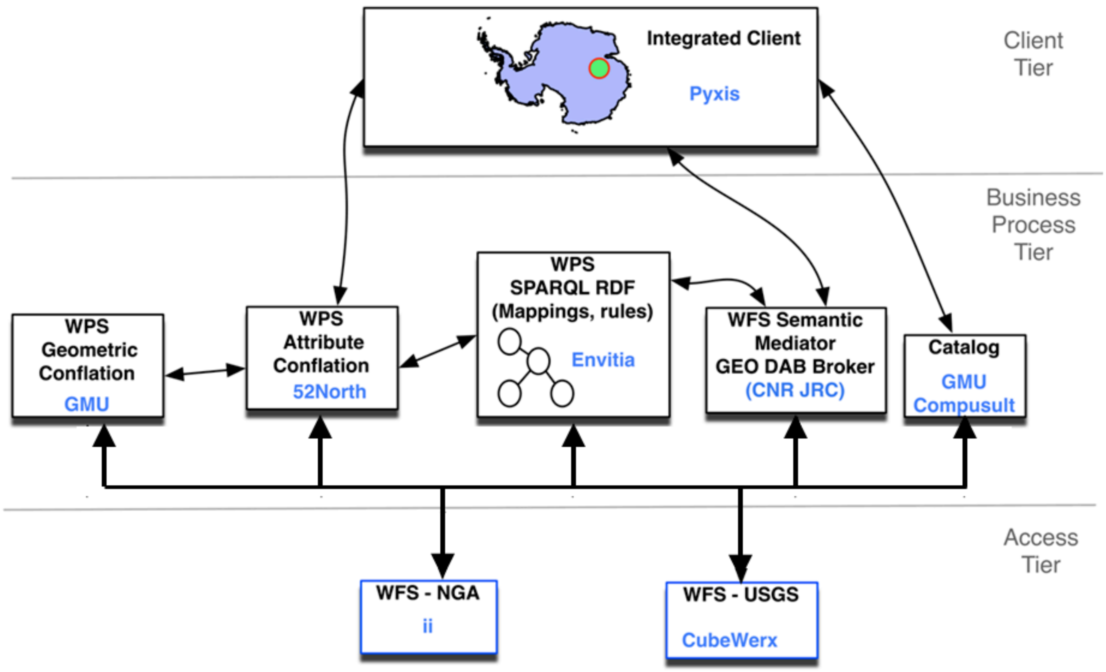

Figure 5. OWS-9 CCI conflation and broker architecture

8 SPARQL Server

The SPARQL server is a web application that allows for querying RDF documents through SPARQL. The SPARQL server relies on a triple store that may be encoded in RDF or held in a relational database.

8.1 Interface

In OWS-8 the SPARQL Server was provided through the W3C-defined interface that allows clients to send HTTP Get requests containing SPARQL queries. Responses to SPARQL SELECT queries are encoded in the SPARQL Results XML format, whereas responses to CONSTRUCT and DESCRIBE queries are encoded in RDF/XML. In OWS-8 opportunities to define OWS interfaces for SPARQL servers were identified for GetCapabilities operations and Error reporting through ows:ExceptionReport messages.

In OWS-9, the CCI thread experimented with the potential to wrap a W3C-based SPARQL Server within an OGC WPS. The reason for selecting the WPS as a wrapper for the SPARQL Server is that it would enable OGC-conformant web services to exploit the SPARQL Server with minimal modification to their existing interfaces.

In OWS-9 the WPS Execute operation was used to provide a process that acts as a proxy for SPARQL queries. The WPS process was configured to support a single input parameter of type literal string called “query” for receiving SPARQL queries in CDATA containers. Due to the presence of tags(containing “<” and “>” characters) in the queries it is necessary to encapsulate them in CDATA elements that prevent XML parsers from attempting to read the tags.

```
<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
  <ows:Identifier>com.envitia.rdf.SparqlQueryProcess</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>query</ows:Identifier>
      <wps:Data>
        <wps:LiteralData><![CDATA[SELECT   ?a WHERE  {?a
<http://www.w3.org/2000/01/rdf-schema#domain>
<http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0#RailwayGeocurve>
   }]]></wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:ResponseDocument storeExecuteResponse="false" lineage="false"
status="false">
      <wps:Output>
        <ows:Identifier>result</ows:Identifier>
        <ows:Title>result</ows:Title>
        <ows:Abstract>result</ows:Abstract>
      </wps:Output>
    </wps:ResponseDocument>
  </wps:ResponseForm>
</wps:Execute>
```

The result of executing a SPARQL SELECT query in the WPS is an ExecuteResponse document that includes SPARQL Response XML elements encapsulated within a WPS ComplexData element. The result of executing a SPARQL CONSTRUCT query in the WPS is an ExecuteResponse document that includes an RDF/XML document encapsulated within a WPS ComplexData element. An illustration of the response is shown in the following listing.

```
<ns:ExecuteResponse service="WPS" version="1.0.0">
   <ns:Process ns:processVersion="1.0.0">
      <ns1:Identifier xmlns:ns1="http://www.opengis.net/ows/1.1">
com.envitia.rdf.SparqlQueryProcess
</ns1:Identifier>
      <ns1:Title/>
   </ns:Process>
   <ns:Status creationTime="2012-08-05T21:35:34.356+01:00">
      <ns:ProcessSucceeded>Succesfully processed.</ns:ProcessSucceeded>
   </ns:Status>
   <ns:ProcessOutputs>
      <ns:Output>
```

```
          <ns1:Identifier
xmlns:ns1="http://www.opengis.net/ows/1.1">result
</ns1:Identifier>
          <ns1:Title xmlns:ns1="http://www.opengis.net/ows/1.1">result
</ns1:Title>
          <ns:Data>
             <ns:ComplexData mimeType="text/xml">
                <sparql xmlns="http://www.w3.org/2005/sparql-results#">
                   <head>
                      <variable name="a" />
                   </head>
                   <results>
                      <result>
                         <binding name="a">
   <uri>http://metadata.dod.mil/..#RailwayGeocurve.trackOrLaneCount</uri>
                         </binding>
                      </result>
                      <result>
                         <binding name="a">
   <uri>http://metadata.dod.mil/..#RailwayGeocurve.surfaceSlope_lowerValue
                            </uri>
                         </binding>
                      </result>
                   </results>
                </sparql>
             </ns:ComplexData>
          </ns:Data>
       </ns:Output>
   </ns:ProcessOutputs>
</ns:ExecuteResponse>
```

8.2 Discovery of SPARQL Servers and Ontologies

A WPS Application Profile describes how WPS shall be configured to serve a process that is recognized by OGC. The WPS specification allows service providers to develop application profiles that standardize solutions within particular communities. The use of an application profile enables customization and optimization of interfaces in order to achieve high interoperability. The standard requires that an application profile consist of the following mandatory elements:

- An OGC URN that uniquely identifies the process (mandatory)
- A reference response to a DescribeProcess request for that process (mandatory)

The standard also allows the following optional elements:

- A human-readable document that describes the process and its implementation.
- A WSDL description for that process.

To support discovery of SPARQL Servers provided through WPS interfaces, a workflow consisting of the following sequence of actions was devised:

1. Search for and discover services of type 'SPARQL Server' through a CSW
2. Retrieve the process identifier of the SPARQL Server from a WPS GetCapabilities response
3. Retrieve the titles and metadata of the available ontologies from the SPARQL Server through a SPARQL DESCRIBE query referenced from a WPS DescribeProcess response
4. Query any triple that 'isDefinedBy' by the referenced ontology through a SPARQL SELECT query.

The MetadataType provided by OWS Common was used to support the encoding of titles and URLs of descriptions of ontologies. The MetadataType uses the following XLink attributes:

- xlink:href: Mandatory reference to a remote resource or local payload.
- xlink:role: Optional reference to a resource that describes the role of this reference.
- xlink:title: Optional description of the meaning of the referenced resource in a human-readable fashion

Use of the aforementioned attributes is presented in the following listing.

```
<ns:ProcessDescriptions xmlns:ns="http://www.opengis.net/wps/1.0.0"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsDescribeProcess_response.xsd"
   xml:lang="en-US" service="WPS" version="1.0.0">
   <ProcessDescription statusSupported="true"
      storeSupported="true" ns:processVersion="1.0.0">
      <ns1:Identifier xmlns:ns1="http://www.opengis.net/ows/1.1">
           com.envitia.rdf.SparqlQueryProcess
      </ns1:Identifier>
      <ns1:Title xmlns:ns1="http://www.opengis.net/ows/1.1">
           com.envitia.rdf.SparqlQueryProcess
      </ns1:Title>
      <ns1:Metadata xmlns:ns1="http://www.opengis.net/ows/1.1"
         xmlns:xlin="http://www.w3.org/1999/xlink" xlin:title=
"OWS-9 Semantic Mappings"
         xlin:href="http://someserver/rdfengine/sparql?query=
describe+%3Chttp%3A%2F%2Fwww.opengeospatial.org%2Fows9%3E&amp;output=
xml&amp;stylesheet=&amp;pw=0ws8" />
      <ns1:Metadata xmlns:ns1="http://www.opengis.net/ows/1.1"
```

```
          xmlns:xlin="http://www.w3.org/1999/xlink" xlin:title=
"OWS-8 Semantic Mappings"
          xlin:href="http://someserver/rdfengine/sparql?query=
describe+%3Chttp%3A%2F%2Fenv032011.appspot.com%2Fmappings.rdf%3E&amp;
output=xml&amp;stylesheet=&amp;pw=0ws8" />
       <ns1:Metadata xmlns:ns1="http://www.opengis.net/ows/1.1"
          xmlns:xlin="http://www.w3.org/1999/xlink" xlin:title=
"OWS-8 NGA TDS model"
   xlin:href="http://someserver/rdfengine/sparql?query=describe+
%3Chttp%3A%2F%2Fmetadata.dod.mil%2Fmdr%2Fns%2FGSIP%2F3.0%2Ftds%2F3.0
%3E&amp;output=xml&amp;stylesheet=&amp;pw=0ws8" />
       <ns1:Metadata xmlns:ns1="http://www.opengis.net/ows/1.1"
          xmlns:xlin="http://www.w3.org/1999/xlink" xlin:title=
"OWS-8 USGS TNM model"
   xlin:href="http://someserver/rdfengine/sparql?query=describe+
%3Chttp%3A%2F%2Fwww.usgs.gov%2Fprojects%2Fows8%3E&amp;output=xml&amp;
stylesheet=&amp;pw=0ws8" />
       <DataInputs>
       ...
       </DataInputs>
       <ProcessOutputs>
        ...
       </ProcessOutputs>
   </ProcessDescription>
</ns:ProcessDescriptions>
```

The URL presented by the Href attribute presents an RDF/OWL description of the referenced ontology. The preceding illustration presents Href links to different ontologies; as demonstrated in OWS-8, it is possible to have multiple ontologies provided by the same SPARQL Server. The Href link retrieves documents of the form presented in the following listing:

```
<rdf:RDF>
  <owl:Ontology rdf:about="http://env032011.appspot.com/mappings.rdf">
    <owl:imports rdf:resource="http://env032011.appspot.com/rmm.rdf"/>
    <owl:imports rdf:resource="http://www.usgs.gov/projects/ows8"/>
    <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#date"
    >2011-09-01</dc:date>
    <dc:title>OGC OWS-8 CCI Mappings</dc:title>
    <owl:versionInfo
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >0.3</owl:versionInfo>
    <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#date"
    >2011-08-04</dc:date>
    <owl:imports
rdf:resource="http://env032011.appspot.com/geosparql.rdf"/>
    <dc:description
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Semantic mediation mappings</dc:description>
    <owl:imports
rdf:resource="http://metadata.dod.mil/mdr/ns/GSIP/3.0/tds/3.0"/>
    <dc:title>OWS-8 CCI Mappings ontology</dc:title>
```



```
  </owl:Ontology>
</rdf:RDF>
```

```
<rdf:RDF>
  <owl:Ontology rdf:about="http://www.opengeospatial.org/ows9">
    <owl:imports
rdf:resource="http://env032011.appspot.com/geosparql.rdf"/>
    <dc:title>OGC OWS-9 CCI Mappings </dc:title>
    <dc:description
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >OWS-9</dc:description>
    <dc:date rdf:datatype="http://www.w3.org/2001/XMLSchema#date"
    >2012-06-13</dc:date>
    <owl:versionInfo
rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >0.3</owl:versionInfo>
  </owl:Ontology>
</rdf:RDF>
```

The complete list of classes and individuals recorded as belonging to the referenced ontologies can then be retrieved using a SPARQL SELECT query of the form:

```
SELECT  ?a WHERE { ?a <http://www.w3.org/2000/01/rdf-schema#isDefinedBy>
<http://somenamespace/ontology>     }
```

8.3 Lessons Learnt: version control and modularity for mappings

As the OWS testbeds continue to build semantic mappings, version control and modularity will become more important. In OWS-9 mappings recorded in OWL were annotated with an rdfs:isDefinedBy reference to the ontology they were defined in, to distinguish them from OWS-8 mappings. Both sets of mappings however make use of the same TDS and TNM ontologies developed in OWS-8. The use of rdfs:isDefinedBy together with owl:versionInfo facilitates the versioning of mappings, which improves the maintenance of the mappings.

8.4 Possible WPS Profile for SPARQL Servers

As a result of the testing of WPS wrappers for SPARQL Servers in OWS-9, the thread proposes the following inputs for WPS-enabled SPARQL Servers.

- Query: A single mandatory string that takes SPARQL queries such as SELECT and CONSTRUCT wrapped in a CDATA element.

The thread proposes the following outputs for WPS-enabled SPARQL Servers:

- Result: A single mandatory ComplexData element that encapsulates a SPARQL Response XML element.

8.5 Change Request

As a result of the testing of WPS wrappers for SPARQL Servers in OWS-9, the following change request was submitted.

Table 1. Change request for WPS Process Description metadata

Reference	268
Reason for change	This issue arose from the OWS-9 CCI Semantic Mediation activity [OGC 12-103] during the development of a WPS wrapper for SPARQL Servers. The issue however also applies to all WPS implementations. The ows:MetadataType class provided by OWS Common is used within the WPS Process Description. However, the ows:MetadataType class is severely limited when the metadata is needed to be within the wps:DescribeProcess response document. That is, rather than link to an external document as is currently the case, some use cases may require the metadata to be presented within the DescribeProcess response itself. In such use cases, the current structure of the wps:ProcessDescription and its use of the ows:MetadataType classes is limited.
Summary of Change	The wps:DescribeProcess should be revised to support a choice of ows:MetadataType, gmd:MD_Metadata or gmd:CI_Citation. That is 'choice' in XSD terms where implementations can only use one of the options provided. The latter classes are provided by ISO 19115.
Consequences if **not approved**	The WPS processes described will end up being used inappropriately by service consumers, potentially leading to a loss of trust in the processes being offered by service providers.

8.6 Registration of SPARQL Servers

With the wrapping of SPARQL Servers in a WPS, it becomes possible to consider registration of the SPARQL Server in terms of ISO 19119, the international standard for geospatial service architecture. This ISO standard also provides a selection of metadata for describing services.

In OWS-8 the following minimal metadata based on ISO 19115 was used to describe the SPARQL Server.

```
<?xml version="1.0" encoding="UTF-8"?>
<gmd:MD_Metadata ..>
   <gmd:language>
      <gco:CharacterString>en</gco:CharacterString>
   </gmd:language>
   <gmd:contact />
   <gmd:dateStamp>
      <gco:Date>2011-07-19</gco:Date>
   </gmd:dateStamp>
   <gmd:metadataStandardName>
      <gco:CharacterString>ISO Metadata Standard Geographic Info
      </gco:CharacterString>
   </gmd:metadataStandardName>
   <gmd:metadataStandardVersion>
      <gco:CharacterString>ISO 19115:2003</gco:CharacterString>
   </gmd:metadataStandardVersion>
   <gmd:identificationInfo>
      <gmd:MD_DataIdentification>
         <gmd:citation>
            <gmd:CI_Citation>
                 ...
            </gmd:CI_Citation>
         </gmd:citation>
         <gmd:abstract />
         <gmd:resourceMaintenance>
            <gmd:MD_MaintenanceInformation>
               <gmd:maintenanceAndUpdateFrequency />
               <gmd:dateOfNextUpdate>
                  <gco:Date>2012-07-19</gco:Date>
               </gmd:dateOfNextUpdate>
            </gmd:MD_MaintenanceInformation>
         </gmd:resourceMaintenance>
         <gmd:descriptiveKeywords>
            <gmd:MD_Keywords>
               <gmd:keyword>
                  <gco:CharacterString>Sparql Server
                  </gco:CharacterString>
               </gmd:keyword>
            </gmd:MD_Keywords>
         </gmd:descriptiveKeywords>
         <gmd:language>
            <gco:CharacterString>en</gco:CharacterString>
         </gmd:language>
         <gmd:extent>
            <gmd:EX_Extent>
               <gmd:geographicElement>
                  <gmd:EX_GeographicDescription>
                     <gmd:geographicIdentifier>
                        <gmd:MD_Identifier>
                           <gmd:authority>
                              <gmd:CI_Citation>
                                 <gmd:title>
```

```
                                    <gco:CharacterString>The
World</gco:CharacterString>
                                 </gmd:title>
                                 <gmd:date />
                              </gmd:CI_Citation>
                           </gmd:authority>
                           <gmd:code>
                              <gco:CharacterString />
                           </gmd:code>
                        </gmd:MD_Identifier>
                     </gmd:geographicIdentifier>
                  </gmd:EX_GeographicDescription>
               </gmd:geographicElement>
               <gmd:geographicElement>
                  <gmd:EX_GeographicBoundingBox>
                     <gmd:westBoundLongitude>
                        <gco:Decimal>-180.0</gco:Decimal>
                     </gmd:westBoundLongitude>
                     <gmd:eastBoundLongitude>
                        <gco:Decimal>180.0</gco:Decimal>
                     </gmd:eastBoundLongitude>
                     <gmd:southBoundLatitude>
                        <gco:Decimal>-90.0</gco:Decimal>
                     </gmd:southBoundLatitude>
                     <gmd:northBoundLatitude>
                        <gco:Decimal>90.0</gco:Decimal>
                     </gmd:northBoundLatitude>
                  </gmd:EX_GeographicBoundingBox>
               </gmd:geographicElement>
            </gmd:EX_Extent>
         </gmd:extent>
      </gmd:MD_DataIdentification>
   </gmd:identificationInfo>
   <gmd:distributionInfo>
      <gmd:MD_Distribution />
   </gmd:distributionInfo>
</gmd:MD_Metadata>
```

In OWS-9 the following comprehensive metadata based on ISO 19119 was produced for registering the SPARQL Server. It should be noted that this approach also registers the operations offered by the SPARQL Service and provides URLs for invoking the operations.

```
<gmd:MD_Metadata ..>
    <gmd:language>
        <gco:CharacterString>en</gco:CharacterString>
    </gmd:language>
    <gmd:hierarchyLevel>
        <gmd:MD_ScopeCode
codeList="http://www.isotc211.org/2005/resources/codeList.xml#MD_ScopeCode"
codeListValue="service">service</gmd:MD_ScopeCode>
```

```
    </gmd:hierarchyLevel>
    <gmd:hierarchyLevelName>
        <gco:CharacterString>SPARQL Server</gco:CharacterString>
    </gmd:hierarchyLevelName>
    <gmd:contact />
    <gmd:dateStamp>
        <gco:Date>2011-07-19</gco:Date>
    </gmd:dateStamp>
    <gmd:metadataStandardName>
        <gco:CharacterString>ISO19119</gco:CharacterString>
    </gmd:metadataStandardName>
    <gmd:metadataStandardVersion>
        <gco:CharacterString>2005/PDAM 1</gco:CharacterString>
    </gmd:metadataStandardVersion>
    <gmd:identificationInfo>
        <srv:SV_ServiceIdentification>
            <gmd:citation>
                <gmd:CI_Citation>
                      ...
                </gmd:CI_Citation>
            </gmd:citation>
            <gmd:abstract>
                <gco:CharacterString> This service publishes triple stores
encoded in RDF and allows the RDF to be queried in
SPARQL</gco:CharacterString>
            </gmd:abstract>
            <gmd:resourceMaintenance>
                <gmd:MD_MaintenanceInformation>
                           ...
                </gmd:MD_MaintenanceInformation>
            </gmd:resourceMaintenance>
            <gmd:descriptiveKeywords>
                <gmd:MD_Keywords>
                       ...
                </gmd:MD_Keywords>
            </gmd:descriptiveKeywords>
            <srv:serviceType>
                <gco:LocalName>SPARQL</gco:LocalName>
            </srv:serviceType>
            <srv:serviceTypeVersion>
                <gco:CharacterString>0.0.1</gco:CharacterString>
            </srv:serviceTypeVersion>
            <srv:extent>
                <gmd:EX_Extent>
                    <gmd:geographicElement>
                        <gmd:EX_GeographicBoundingBox>
                                ...
                        </gmd:EX_GeographicBoundingBox>
                    </gmd:geographicElement>
                </gmd:EX_Extent>
            </srv:extent>
            <srv:couplingType>
                <srv:SV_CouplingType
codeList="http://someurl#SV_CouplingType"
codeListValue="tight">tight</srv:SV_CouplingType>
            </srv:couplingType>
```



```
            <srv:containsOperations>
                <srv:SV_OperationMetadata>
                    <srv:operationName>

<gco:CharacterString>GetCapabilities</gco:CharacterString>
                    </srv:operationName>
                    <srv:DCP>
                        <srv:DCPList codeList="http://someurl#DCPList"
codeListValue="HTTPGet">HTTPGet</srv:DCPList>
                    </srv:DCP>
                    <srv:connectPoint>
                        <gmd:CI_OnlineResource>
                            <gmd:linkage>

<gmd:URL>http://someserver/wps/WebProcessingService</gmd:URL>
                            </gmd:linkage>
                        </gmd:CI_OnlineResource>
                    </srv:connectPoint>
                </srv:SV_OperationMetadata>
            </srv:containsOperations>
            <srv:containsOperations>
                <srv:SV_OperationMetadata>
                    <srv:operationName>

<gco:CharacterString>DescribeProcess</gco:CharacterString>
                    </srv:operationName>
                    <srv:DCP>
                        <srv:DCPList codeList="http://someurl#DCPList"
codeListValue="HTTPGet">HTTPGet</srv:DCPList>
                    </srv:DCP>
                    <srv:connectPoint>
                        <gmd:CI_OnlineResource>
                            <gmd:linkage>

<gmd:URL>http://someserver/wps/WebProcessingService</gmd:URL>
                            </gmd:linkage>
                        </gmd:CI_OnlineResource>
                    </srv:connectPoint>
                </srv:SV_OperationMetadata>
            </srv:containsOperations>
            <srv:containsOperations>
                <srv:SV_OperationMetadata>
                    <srv:operationName>
                        <gco:CharacterString>Execute</gco:CharacterString>
                    </srv:operationName>
                    <srv:DCP>
                        <srv:DCPList codeList="http://someurl#DCPList"
codeListValue="HTTPPost">HTTPPost</srv:DCPList>
                    </srv:DCP>
                    <srv:connectPoint>
                        <gmd:CI_OnlineResource>
                            <gmd:linkage>

<gmd:URL>http://someserver/wps/WebProcessingService</gmd:URL>
                            </gmd:linkage>
```



```
                        </gmd:CI_OnlineResource>
                    </srv:connectPoint>
                </srv:SV_OperationMetadata>
            </srv:containsOperations>
            <srv:operatesOn>
                <gmd:MD_DataIdentification>
                    <gmd:citation>
                        <gmd:CI_Citation>
                            ...
                        </gmd:CI_Citation>
                    </gmd:citation>
                    <gmd:abstract >
                        <gco:CharacterString>OWS Semantic Mediation models
and mappings used in OWS-9 and OWS-9</gco:CharacterString>
                    </gmd:abstract >
                    <gmd:descriptiveKeywords>
                        <gmd:MD_Keywords>
                            <gmd:keyword>
                                <gco:CharacterString>SPARQL RDF OWL
SKOS</gco:CharacterString>
                            </gmd:keyword>
                        </gmd:MD_Keywords>
                    </gmd:descriptiveKeywords>
                    <gmd:language>
                        <gco:CharacterString>en</gco:CharacterString>
                    </gmd:language>
                    <gmd:extent/>
                </gmd:MD_DataIdentification>
            </srv:operatesOn>
        </srv:SV_ServiceIdentification>
    </gmd:identificationInfo>
    <gmd:distributionInfo>
        <gmd:MD_Distribution />
    </gmd:distributionInfo>
</gmd:MD_Metadata>
```

9 WFS-Gs

The Gazetteer Service Application Profile of the Web Feature Service (WFS-G) is a Best Practice specification published by the OGC. A WFS-G allows a client to search and retrieve instances of locations recorded using a vocabulary of well known placenames. The National Geospatial Intelligence Agency (NGA) and the United States Geological Survey (USGS) both provide WFS-G for their GNS and GNIS gazetteers respectively. A detailed description of these WFS-G's is presented in the OWS-9 SPEGG Engineering Report [OGC 12-104].

10 Mediation towards a Single Point of Entry Global Gazetteer

The US Government lacks a Single Point of Entry Global Gazetteer (SPEGG), where a user can submit a single query and access global geographic names data across multiple Federal names databases. This is a hindrance to military and civilian users who require access to names data both within and outside the United States. Currently, to obtain authoritative cross border geographic names data, users must make two queries with differing input parameters against both the GNS gazetteer managed by the NGA and the GNIS gazetteer managed by the USGS.

In an ideal situation, users would be querying a single physically unified gazetteer. However, currently there is a significant difference in granularity between the GNIS and GNS. This makes querying against both gazetteers a challenge, as the user must understand and semantically map the feature types across databases. An alternative option, is to provide a SPEGG service founded on semantic mediation. The CCI thread explored the potential for establishing a SPEGG based on best practices for Gazetteers built on Web Feature Services (WFS-G).

10.1 Mapping Principles

The CCI thread used the DSG codes from the GNS, feature classes from the GNIS and terms from the ADL feature type thesaurus. The GNS and GNIS gazetteers differ in granularity due to being initially developed by different committees of the US Board on Geographic Names (BGN), namely the Domestic Names Committee and the Foreign Names Committee. The former is responsible for names in the United States and its dependencies, while the latter is responsible for names outside the United States. The GNS and GNIS databases consequently have limited overlap and have quite different levels of granularity in the descriptions. The GNS groups features into more than 600 classes identified through DSG codes. In contrast, the GNIS groups features into 66 feature classes.

The ADL Feature Type Thesaurus contains a hierarchical scheme of terms for classifying geographic places recorded in a gazetteer. The thesaurus is intended for general use and is designed to support many types of gazetteers by improving interoperability through a shared gazetteer. The 2002 version of the thesaurus consisted of 210 preferred terms and 1046 non-preferred terms.

It was therefore necessary to review feature classes from both these gazetteers alongside each other in order to determine the true meanings of the concepts represented by the feature classes. Consequently, the thread identified at least two types of mapping approaches applicable to gazetteers. The approaches included Conceptual Mapping and Applied Mapping.

Conceptual mapping considers only the definitions of the feature classes. The approach maps a source feature class to one or more target feature classes based only on the definitions of the classes. This approach is closely related to the OWS-8 approach which mapped feature types, properties and enumerants in one application schema to those from another. As with OWS-8, conceptual mapping does not consider the feature instances contained in the databases being mapped.

Applied mapping not only considers the definitions of the feature classes but also the content of the gazetteer databases as well. This approach therefore considers feature instances as well as feature classes. A hypothetical example is that of a GNS sugar refinery, which maps to a GNIS locale conceptually but if there are no sugar refineries in the GNIS, there would be no match. In other words, if there are no features collected in the GNIS, even if the categories can be logically mapped, there would be no match. A benefit of this approach is that it reduces the number of false positives; that is, it improves the precision of searches. A disadvantage is that it requires that the designer of the mappings to understand all content of the gazetteer databases in order to do the mappings and thus may not be standard practice.

Following discussions with subject matter experts from the Army Geospatial Center and the NGA, the CCI thread decided to adopt the conceptual mapping approach.

10.2 Procedure and Guidance for Mappings

This section presents guidance provided by subject matter experts from the sponsor organizations.

- **Determining a Match – The 'Or' Case:** In order to identify classes as a match, the definitions should be very close. An 'or' in the description may indicate that a class is broad. As an example, the GNS definition of an arch is 'a natural or man-made structure in the form of an arch', while the GNIS definition of an arch is 'Natural arch-like opening in a rock mass (bridge, natural bridge, sea arch).' Because the GNS definition is broader than the GNIS definition (it includes manmade arches), this would not be an exact match, but rather a narrowMatch.

- **Determining a No Match:** Feature classes not recorded by the GNIS gazetteer but recorded by the GNS should, ideally, not be returned in searches from GNIS viewpoints.

- **Singular versus Plural Feature Classes:** While some feature classes have singular and plural classes (MT and MTS in GNS; summit and range in GNIS), many do not. The CCI thread made the assumption that any singular feature classes could also refer to plural if there is not an explicit plural feature class. For example the BAY and BAYS in the GNS would match the GNIS class bay. In cases where the GNS has a single feature class that may be singular or plural, i.e., crater(s), it would be an exactMatch to the GNIS feature class crater, even though this is not plural.

- **Abandoned/Former (GNS) and (historical) (GNIS):** The NGA concepts of abandoned and the GNIS concept of (historical) do not exactly match and there is no way to determine if a GNIS feature is (historical) without parsing the name, as (historical) is embedded in the name and not a separate field. Abandoned implies no longer used, while the USGS concept of (historical) implies that it no longer serves the original function. For example, an historical Post Office simply means the building is not being used as a post office, not that the building has been abandoned or is no longer in use. This rule would also apply to NGA DSGs, like former inlet or former sugar mill. The sponsors advised not matching NGA abandoned features.

- **Intermittent features:** The GNS records intermittent features which should be mapped to perennial features of the same type in the GNIS. For example, GNS intermittent streams would map to GNIS streams. There are intermittent streams in the GNIS, but there is no way to distinguish intermittent from perennial.

- **Handling NGA "Section of ..." DSGs:** A number of the NGA DSGs have 'section of ...' as part of their description, i.e., section of plain or section of lake. In these cases, since the GNIS does not collect names for sections of features (my assumption), there would be noMatch for these. Unless, of course, the term 'section of' refers to something that is collected by USGS.

- **Multiple GNIS Feature Classes:** A number of the features in the GNIS may be found under multiple feature classes, even though they appear to be the same thing when related to a GNS DSG. A common one is almost any facility, which may be described as a building or a locale. Research institutes may be listed under buildings, schools, and hospitals. Forests may be forests if administered or woodlands if not administered (which is not possible to tell from the codes). In these cases, all possible mappings should be listed.

10.3 Representation of Gazetteer Mappings in OWL and SKOS

The mappings were designed initially using semantic relationships of the Simple Knowledge Organization Systems (SKOS) and captured in a Microsoft Excel spreadsheet. The use of spreadsheets was intended to enable the easy exchange of mappings between the CCI thread and subject matter experts, who may not necessarily have access to ontology editors such as Protégé. An extract of the spreadsheet is presented in Table 2.

Table 2. An extract of the Gazeteer mappings in tabular form

skos:prefLabel	skos:definition	rdf:about	skos:exactMatch	skos:broadMatch
Bar	a shallow ridge or mound of coarse unconsolidated material in a stream channel, at the mouth of a stream, estuary, or lagoon and in the wave-break zone along coasts	#T.BAR	#usgs.gnis.bar	
Barracks	a building for lodging military personnel	#S.BRKS		#usgs.gnis.military
battlefield	a site of a land battle of historical importance	#L.BTL		#usgs.gnis.locale
Bay	a coastal indentation between two capes or headlands, larger than a cove but smaller than a gulf	#H.BAY	#usgs.gnis.bay	
Beach	a shore zone of coarse unconsolidated sediment that extends from the low-water line to the highest reach of storm waves	#T.BCH	#usgs.gnis.beach	
beach ridge	a ridge of sand just inland and parallel to the beach, usually in series	#T.RDGB		#usgs.gnis.ridge

The spreadsheet was developed through the following phases:

- Phase 1: The first phase was automated and involved detecting lexical equivalents between terms (i.e. terms spelt the same way). This found approximately 30 terms shared between GNS and GNIS.
- Phase 2: The second phase was manual and involved reviewing the full definition of GNS terms. In several cases, the GNS term is appended to the GNIS definition

in brackets thereby making it easier to detect the narrower or broader relationship between a GNS term and a GNIS term.

- Phase 3: The third and final phase involved validation by subject matter experts and the revision of the mappings in accordance with feedback received from the subject matter experts.

An illustration of how some of the semantic relationships that were applied is presented in Figure 6. The use of semantic relationships from SKOS was intended to support the alignment of concepts across the GNS, GNIS and ADL controlled vocabularies. The following SKOS semantic relationships were adopted:

- The skos:closeMatch, skos:exactMatch, skos:broadMatch, skos:narrowMatch and skos:relatedMatch properties are used to state mapping (alignment) links between SKOS concepts in different concept schemes, where the links are inherent in the meaning of the linked concepts.

- The properties skos:broadMatch and skos:narrowMatch were used in OWS-9 to state a hierarchical mapping link between concepts.

- The property skos:relatedMatch was used in OWS-9 to state an associative mapping link between concepts.

- The property skos:closeMatch, which is used to link two concepts that are sufficiently similar that they can be used interchangeably in some information retrieval applications, was **not** used in OWS-9 because determining the ‘closeness’ of a match would have been a subjective process. This report however acknowledges that there are approaches under development in academia for calculation of semantic similarity that could enable the calculation of the ‘closeness’ of a match through more objective means.

- The property skos:exactMatch was used in OWS-9 to link concepts, indicating a high degree of confidence that the concepts can be used interchangeably across a wide range of information retrieval applications.

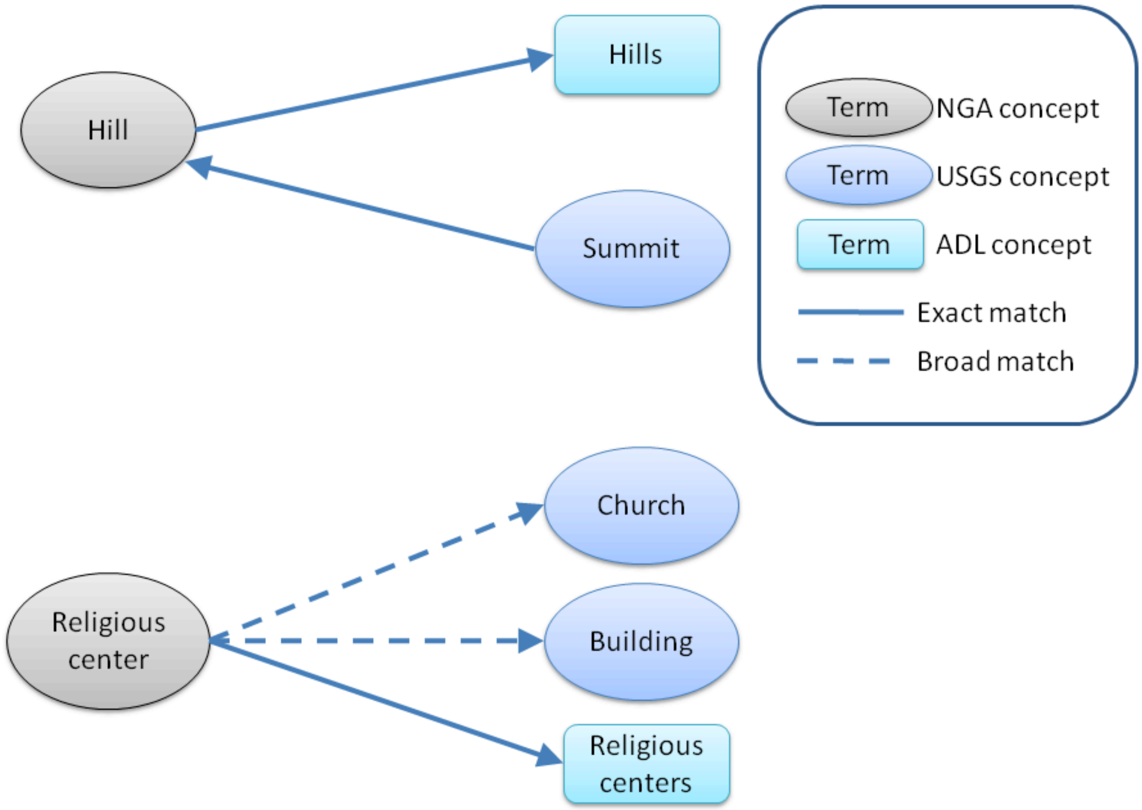

Figure 6. Illustration of semantic mappings between concepts from the NGA and USGS gazetteers

Following the capture of semantic relationships between concepts on the mappings spreadsheet, an RDF/OWL document was then generated from the spreadsheet. As shown in the following listing a skos:Concept element was generated for each GNS DSG code, with the actual code used by the skos:Concept as an identifier. Similarly GNIS feature classes were used as identifiers for skos:Concept elements. The natural name of the DSG code was used as the value of the skos:prefLabel property and the definition of the DSG code was used as the value of the skos:definition property. The skos:inScheme property, which relates a resource to a concept scheme, was used to distinguish between GNS, GNIS and ADL concepts.

```
    <skos:Concept rdf:about="&tds;A.ADM3">
         <rdf:type rdf:resource="&owl;Thing"/>
         <skos:definition rdf:datatype="&xsd;string"
             >a subdivision of a second-order administrative
division</skos:definition>
         <isDefinedBy rdf:datatype="&xsd;anyURI"
             >http://www.opengeospatial.org/ows9</isDefinedBy>
         <skos:prefLabel rdf:datatype="&xsd;string"
             >third-order administrative division</skos:prefLabel>
         <skos:altLabel rdf:datatype="&xsd;string"
             >urn:x-ogc:def:locationType:NGA:ADM3</skos:altLabel>
```



```
        <skos:inScheme rdf:resource="&tds;nga.gns"/>
        <skos:exactMatch rdf:resource="&www;adl#adl.1132"/>
        <skos:broadMatch rdf:resource="&www;adl#adl.9"/>
        <skos:broadMatch
rdf:resource="&projects;ows9#usgs.gnis.civil"/>
    </skos:Concept>
```

The following SPARQL query demonstrates how to retrieve mappings from the SPARQL server, with the skos:broadMatch predicate as a constraint. Other predicates such as skos:exactMatch could also be used. Alternatively, multiple predicates could be integrated into a single query through use of a SPARQL UNION operator.

```
SELECT  ?sourceHref ?sourceLabel ?sourceDefinition ?predicate
?targetHref ?targetLabel ?targetDefinition WHERE  {
?sourceHref <http://www.w3.org/2004/02/skos/core#inScheme>
<http://metadata.dod.mil/mdr/ns/gsip/3.0/tds/3.0#nga.gns>.
?sourceHref <http://www.w3.org/2004/02/skos/core#prefLabel>
?sourceLabel.
?sourceHref <http://www.w3.org/2004/02/skos/core#definition>
?sourceDefinition.
?sourceHref <http://www.w3.org/2004/02/skos/core#broadMatch>
?targetHref.
?sourceHref ?predicate ?targetHref.
?targetHref <http://www.w3.org/2004/02/skos/core#prefLabel>
?targetLabel.
?targetHref <http://www.w3.org/2004/02/skos/core#definition>
?targetDefinition.
?targetHref <http://www.w3.org/2004/02/skos/core#inScheme>
<http://www.usgs.gov/projects/ows9#usgs.gnis>.
}
```

10.4 Cascading with the WFS-G Semantic Mediator

The WFS-G Semantic Mediator was configured to work directly with the NGA and USGS WFS-Gs, but also to work through the cascading WFS-G provided by CubeWerx. The GetFeature operation of the WFS-G has the special ability to describe a resource (i.e. gazetteer) as well as to publish the resource as well. The three main feature types offered by a WFS-G are SI_LocationInstance, SI_LocationType and SI_Gazetteer. A cascading WFS-G needs to be able to cascade requests for these feature types. The challenge for a cascading WFS-G with semantic mediation capabilities is that it has to resolve the different location types offered by the underlying WFS-Gs accessed by the cascading WFS-G. The two feature types requiring semantic mediation in a cascading WFS-G were identified to be the SI_LocationInstance and SI_Gazetteer.

The features presented by the SI_Gazetteer response are important because they present the location types that are supported by the WFS-G. To mediate the GetFeature requests for the SI_Gazetteer feature type, therefore the cascading WFS had to return a collation of all location types supported by the underlying WFS-G. This would enable a client that accesses the NGA or USGS WFS-Gs to use the same location types when accessing the cascading WFS-G. The challenge for the semantic mediator however was in resolving the location types used by the client to URI references used in the OWL concepts. To address this challenge, each OWL concept was annotated with its GNIS or GNS location type reference using the skos:altLabel predicate.

To mediate the GetFeature requests for the SI_LocationInstance feature type, two separate approaches were considered for filtering features through semantic constraints. The first approach filters features on the semantic mediator and the second approach filters features on the underlying WFS-Gs (i.e. those provided by the NGA and USGS). This report hereinafter refers to the first approach as the ‘thick mediator’ approach and the second approach as the ‘thin mediator’ approach.

Within the ‘thick mediator’ approach, each GetFeature request sent to the semantic mediator is processed to extract the bounding envelope, if there is one, and to extract any semantic constraints. If there is a bounding envelope in the request, a new request is constructed with the bounding coordinates and then sent to the NGA and USGS WFS-Gs. The responses from the NGA and USGS WFS-Gs are then filtered to extract feature instances based on the semantic constraints that were extracted from the GetFeature request sent to the semantic mediator.

Within the ‘thin mediator’ approach, a GetFeature request with multiple ‘Or’ constraints for each location type is derived from the semantic mappings and sent to the underlying WFS-G. The WFS-Gs would then return locations of types specified in the GetFeature requests. An initial obstacle to implementing this second approach, was that one of the WFS-Gs did not have support for ‘Or’ filters. The WFS-G was later revised to provide support for ‘Or’ filters. However, another issue arose as one of the WFS-Gs could not handle more than a specific number of hits from the non-spatial part of a query. It was therefore decided to use the first approach for filtering features, which involved filtering features on the semantic mediator.

Regardless of whether the filtering is applied at the semantic mediator level or the underlying WFS-G level, the CCI thread identified the need for a complex type to support the provision of semantic constraints to OGC Filters. The CCI thread considered previous proposals by OGC members, specifically, the thread considered use of the classifiedAs property and use of the IsInstanceOf. The following listing shows example usage of these properties.

```
    <ogc:Filter>
      <ogc:classifiedAs>
        <ogc:PropertyName>Type</ogc:PropertyName>
        <ogc:Literal>
         http://www.example.com/Ontology#ChalkPit
        </ogc:Literal>
```

```
      </ogc:classifiedAs>
   </ogc:Filter>
```

```
<ogc:Filter>
   <sem:IsInstanceOf>
      <ogc:PropertyName>/RegistryObject/@id</ogc:PropertyName>
   <sem:Concept>http://example.com/ontologies/gis_datatypes#GML
</sem:Concept>
   </sem:IsInstanceOf>
</ogc:Filter>
```

Semantic Annotations in OGC standards discussion paper [OGC 08-167r2] discussed the use of semantics with the classifiedAs filter operator, specified in the CSW specification [OGC 07-006r1]. The approach presented in the Discussion Paper would work well for ontology-assisted classifications. However, where the ontology has been used to specify a different type of relation, e.g. broadMatch, the proposed classifiedAs filter would not be appropriate. In the case of semantic mediation, subjects can be related to objects through any number of predicates, it is therefore necessary to have a more generic semantic filter that accepts both the target concept and the predicate as parameters.

Another approach has previously been proposed by Ionic in 2007, for Catalogue support for semantic annotations. They proposed using a property called IsInstanceOf. Their approach also provides a useful foundation for the use of semantic filters in OGC standards. However, the approach is limited to 'instance of' relationships.

An enhancement of the above approaches is to include the predicate and target concept in the filter as shown below. OWS-9 proposes this approach which allows any predicate to be specified in a filter through the SemanticRelation property of the PropertyIsSemanticallyRelatedTo type.

```
<ogc:Filter>
   <ogc:PropertyIsSemanticallyRelatedTo>
      <ogc:PropertyName>locationType</ogc:PropertyName>
   <ogc:ResourceIdentifier>http://metadata.dod.mil/..#P.PPL
</ogc:ResourceIdentifier>
   <ogc:SemanticRelation>http://www.w3.org/2004/02/..#broadMatch
</ogc:SemanticRelation>
   </ogc:PropertyIsSemanticallySimilarTo>
</ogc:Filter>
```

The proposed PropertyIsSemanticallyRelatedTo type is represented in Unified Modeling Language (UML) in Figure 7. As illustrated, the proposed type extends the ComparisonOpsType, defined in the OGC Filter standard.

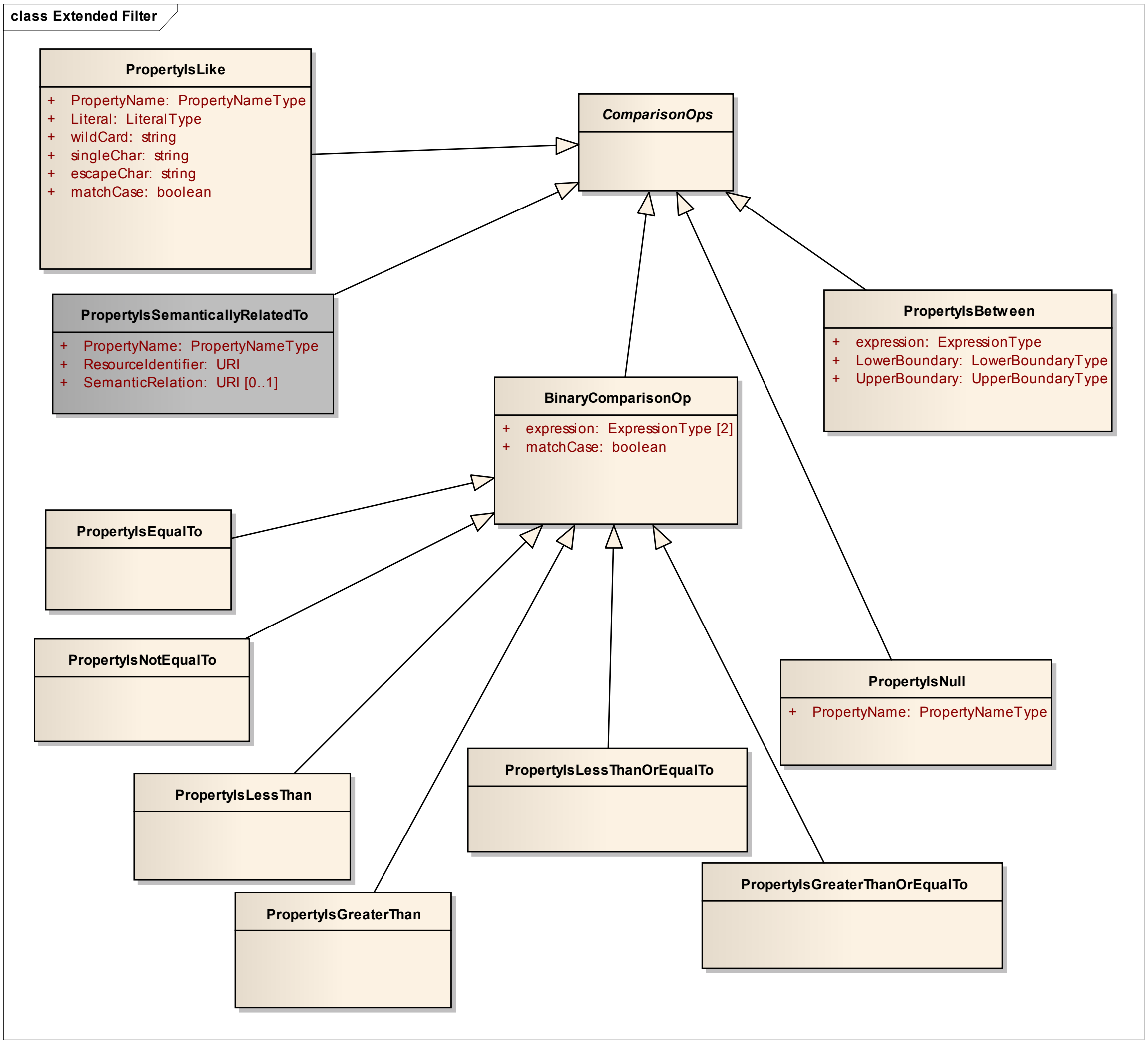

Figure 7. UML class model of proposed extension to Filter comparison operations

The semantic mediator was also configured to filter by fuzzy match through a PropertyIsLike comparison operator. The implemented fuzzy match operator calculates the number of characters in the input string that are equivalent to characters in the placenames and at the same position in both the input string and the placename. The degree of match is then calculated as a function of the number of matching characters and the length of the placename string. The function returns a value of 1 for a complete match and a value of 0 for no match. To enable the client to control the filtering of locations, the minimum required match to allow placenames through the filter is entered through a minimumFuzzyMatch attribute delivered through the PropertyIsLike comparison operator. This minimumFuzzyMatch attribute is a proposal from the OWS-9 CCI thread. It was observed that it is possible for different services to implement different fuzzy match algorithms. It is therefore recommended that metadata identifying the implemented fuzzy match algorithm be presented in the GetCapabilities response.

An alternative approach for implementing fuzzy matching is presented in the OWS-9 SPEGG report [OGC 12-104], which also proposes a PropertyMatches comparison operator. The Semantic Mediation ER concludes that the proposed PropertyMatches operator is more appropriate for supporting fuzzy matches than extending the PropertyIsLike operator because the proposed PropertyMatches operator allows for the specification of the algorithm being used to test for fuzzy matching.

10.5 Lessons Learnt: Nested Filters in KVP requests

In the ISO 19112 application schema used by WFS-G, each location instance is classified by a location type property that includes a title and an 'href' identifier as attributes. The WFS-G best practice requires that the href attribute reference a resolvable URL that returns a description of the location type. An initial implementation of the USGS WFS-G adopted KVP requests with XML encoded filters as values of the location type href parameter as shown in the following listing.

```
<SI_LocationInstance>
...
<locationType title="Island"
href="http://wfsg.usgs.gov/cubewerx/cubeserv.cgi?config=full&amp;
datastore=full&amp;service=WFS&amp;request=GetFeature&amp;typeName=
iso19112:SI_LocationType&amp;FILTER=&lt;ogc:Filter&gt;&lt;ogc:
PropertyIsEqualTo&gt;&lt;ogc:PropertyName&gt;name&lt;/ogc:PropertyName&gt
;&lt;ogc:Literal&gt;Island&lt;/ogc:Literal&gt;&lt;/ogc:PropertyIsEqualTo
&gt;&lt;/ogc:Filter&gt;" />
...
</SI_LocationInstance>
```

The use of these XML encoded filters to constrain KVP requests resulted in nested filters, where the outer filter is the filter of the client's request and inner filter is the filter of the location type href property. This caused usability problems on the semantic mediator as the inner filter required double-URL encoding to make it URL safe. URL encoding uses escape characters to convert URLs into a format that can be transmitted over the internet. By double URL-encoding, this report refers to the URL encoding of text that has already been URL encoded. The use of Filter XML in location type href values results in a situation where the base of a KVP request is not URL encoded, the filter of the KVP request is single-URL encoded and the href value used in the filter is double-URL encoded. To address this issue, the CCI thread revised the USGS WFS-G to use StoredQueries that resolve to descriptions of location types. The use of StoredQueries enabled the href value passed with the filter to be single-URL encoded.

It should be noted that the nested filter issue could have affected any WFS-G client attempting to use the location type href attribute as a filter constraint in an HTTP GET request. The issue is not limited to the semantic mediator only. It is therefore recommended that the WFS-G Best Practice document should recommend the use of

StoredQueries instead of XML encoded filters as values of the location type href attribute.

10.6 Change Request

As a result of the prototyping of semantic support for WFS-G, the following change request was submitted.

Table 3. Change request for semantics support in OGC Filters

Reference	269
Reason for change	This issue arose from the OWS-9 CCI Semantic Mediation activity [OGC 12-103] during the prototyping of a semantically assisted cascading WFS-G. None of the Filter comparison operators allows for testing of semantic relationships. In the semantic web, semantic relationships are specified by associating a subject(s) to an object(o) through a predicate (p) thereby forming a triple of the form (s,p,o). The OWS-9 CCI thread considered use of the classifiedAs property that is described in the CSW specification. The thread also considered use of the IsInstanceOf property proposed in 2007. Whereas both approaches work well for describing classifications and inheritance, both approaches prescribe the nature of the relationship between the subject and the object (i.e. classification and instance of). None of the approaches therefore allows for the specification of different predicates between subjects and objects. For semantic mediation, however the nature of the relationships between subjects and objects can vary. Therefore flexibility is needed in specifying the predicates.
Summary of Change	It is proposed that the Filter specification should include a PropertyIsSemanticallyRelatedTo comparison operator that accepts a propertyname, resourceidentifier and zero or more semanticrelationships.
Consequences if **not approved**	Without this extension, the Filter specification will not be able to exploit the full potential of the semantic web.

11 Mediation between Aviation Data and User Terminology

Delivery of aviation data encoded in the Aeronautical Information Exchange Model (AIXM) has been demonstrated in previous OWS testbeds. An approach was demonstrated, in OWS-8, that allowed emergency responders to search data based on their own semantics. In OWS-9, there was a cross-thread requirement to demonstrate semantic mediation of aviation data through mediation between AIXM data and user terminology. An overview of the cross-thread requirement is presented in Figure 8.

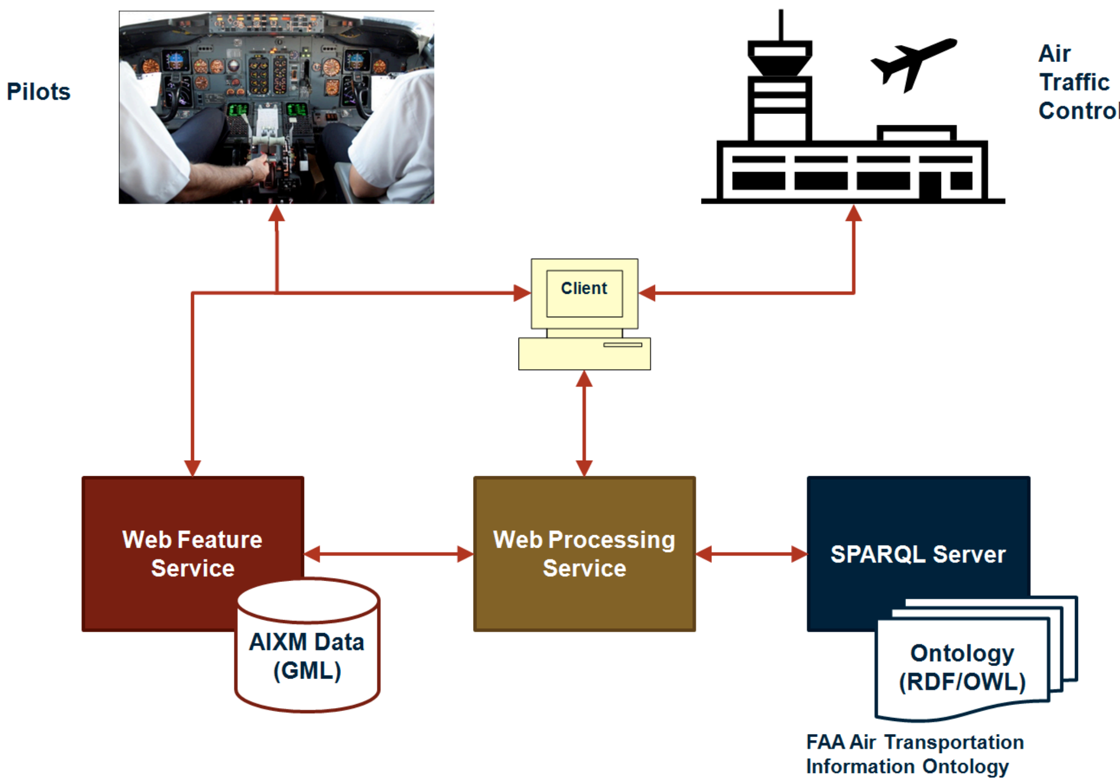

Figure 8. Overview of the components supporting the CCI-Aviation cross-thread activity

Users within this context were identified as pilots and thus an extract from the FAA Air Transportation Information Ontology (ATIO) was provided by the FAA. The ATIO ontology has been developed by the FAA through the JPAMS programme. The next section describes the representation of the pilots' glossary[1] within the ATIO ontology.

[1] http://www.faa.gov/air_traffic/publications/media/pcg.pdf

11.1 The Pilots Glossary within the Air Transportation Information Ontology

The Pilots' Glossary has been developed to facilitate shared understanding of the terms used in aviation. It includes those terms used in communication between pilots and Air Traffic Controllers. The ATIO ontology includes terms from the pilot glossary and specifies generic (broader) and specific (narrower) relationships between terms. An extract from the ATIO ontology showing a selection of terms from the pilots' glossary is presented in Table 4.

Table 4. Extract from the Aviation Mappings spreadsheet

Term	Term definitions	thing (generic)	specific (type / kind)	varietal (sub-type / sub-kind)
Closed Runway	CLOSED RUNWAY- A runway that is unusable for aircraft operations. Only the airport management/military operations office can close a runway.	RUNWAY INFORMATION	RUNWAY STATUS INFORMATION	CLOSED RUNWAY
Closed Traffic	CLOSED TRAFFIC- Successive operations involving takeoffs and landings or low approaches where the aircraft does not exit the traffic pattern.	AIRPORT INFORMATION	AIRPORT STATUS INFORMATION	CLOSED TRAFFIC
Clutter	CLUTTER- In radar operations, clutter refers to the reception and visual display of radar returns caused by precipitation, chaff, terrain, numerous aircraft targets, or other phenomena. Such returns may limit or preclude ATC from providing services based on radar.	DISPLAY IMAGE	RADAR DISPLAY IMAGE	RADAR CLUTTER

The CCI thread decided to use SKOS to represent the terms from the pilots' glossary as presented by the ATIO ontology. SKOS offered the semantic relationships necessary for representing generic (broader) and specific (narrower) relationships between terms. Each term from the glossary was given a unique numeric identifier for use as the rdf:About value. The term itself was encoded in a skos:prefLabel property and alternative terms specified through a skos:altLabel property. The definition of the term was encoded in the skos:definition property. Each SKOS concept derived from a term in the pilots' glossary was then associated with an AIXM feature type via a shared broader concept from the ATIO ontology. For example, in the following listing the pilots' term "Precision Approach Runway Category II" is associated with the AIXM Runway feature type via the concept "Runway Information" from the ATIO ontology.


```
<skos:Concept rdf:about="&jpams;jpams.10540">
   <rdf:type rdf:resource="&owl;Thing" />
   <skos:prefLabel rdf:datatype="&xsd;string">Precision
      Approach Runway Category II</skos:prefLabel>
   <skos:definition rdf:datatype="&xsd;string">Precision
      Approach Runway Category II</skos:definition>
   <isDefinedBy rdf:resource="http://www.opengeospatial.org/ows9"/>
   <skos:inScheme rdf:resource="&jpams;faa.jpams.pilotglossary" />
   <skos:broader rdf:resource="&jpams;jpams.1193" />
</skos:Concept>

<skos:Concept rdf:about="&jpams;jpams.1193">
   <rdf:type rdf:resource="&owl;Thing" />
        <rdf:type rdf:resource="&skos;Concept"/>
        <skos:definition rdf:datatype="&xsd;string"
            >RUNWAY INFORMATION</skos:definition>
        <skos:prefLabel rdf:datatype="&xsd;string"
            >RUNWAY INFORMATION</skos:prefLabel>
        <isDefinedBy rdf:resource="http://www.opengeospatial.org/ows9"/>
<skos:inScheme rdf:resource="&jpams;faa.jpams.pilotglossary"/>
        <skos:broader rdf:resource="&jpams;jpams.1239"/>
</skos:Concept>

<skos:Concept rdf:about="&jpams;aixm.Runway">
   <rdf:type rdf:resource="&owl;Thing" />
   <skos:altLabel rdf:datatype="&xsd;string">RW</skos:altLabel>
   <skos:altLabel rdf:datatype="&xsd;string">RWY</skos:altLabel>
   <skos:altLabel rdf:datatype="&xsd;string">RY</skos:altLabel>
   <skos:definition rdf:datatype="&xsd;string">Runway</skos:definition>
   <skos:prefLabel rdf:datatype="&xsd;string">Runway</skos:prefLabel>
   <isDefinedBy rdf:resource="http://www.opengeospatial.org/ows9"/>
   <skos:altLabel rdf:datatype="&xsd;string">runway</skos:altLabel>
   <skos:inScheme rdf:resource="&jpams;faa.jpams" />
   <skos:broader rdf:resource="&jpams;jpams.1193" />
</skos:Concept>
```

To query the ATIO ontology through the WPS interface of the SPARQL Server, the following request can be sent:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
  <ows:Identifier>com.envitia.rdf.SparqlQueryProcess</ows:Identifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>query</ows:Identifier>
      <wps:Data>
```



```
        <wps:LiteralData><![CDATA[
      SELECT  ?concept ?label ?definition ?broader ?AIXMFeatureType
WHERE  {?concept <http://www.w3.org/2004/02/skos/core#inScheme>
<http://www.faa.gov/jpams/ows9#faa.jpams.pilotglossary>.
?concept <http://www.w3.org/2004/02/skos/core#prefLabel> ?label.
?concept <http://www.w3.org/2004/02/skos/core#prefLabel> "Precision
Approach Runway Category II"
^^<http://www.w3.org/2001/XMLSchema#string>.
?concept <http://www.w3.org/2004/02/skos/core#definition> ?definition.
?concept <http://www.w3.org/2004/02/skos/core#broader> ?broader.
?AIXMFeatureType <http://www.w3.org/2004/02/skos/core#broader> ?broader.
                }
      ]]></wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:ResponseDocument storeExecuteResponse="false" lineage="false"
status="false">
      <wps:Output>
        <ows:Identifier>result</ows:Identifier>
        <ows:Title>result</ows:Title>
        <ows:Abstract>result</ows:Abstract>
      </wps:Output>
    </wps:ResponseDocument>
  </wps:ResponseForm>
</wps:Execute>
```

An initial implementation of the aviation semantic mediator provided by Envitia was integrated with the Snowflake Software AIXM WFS. The initial implementation communicated with the Snowflake WFS through HTTP GET requests, without going through an access control layer. As the testbed scenarios evolved, a requirement emerged to also support the COMSOFT AIXM WFS. This presented the additional challenge that the COMSOFT WFS is protected by an authentication layer that requires all clients to be authorized in order to access the service. The COMSOFT WFS also presented the challenge that it required all communication to be through SOAP messaging over HTTP POST. The initial implementation of the aviation semantic mediator was therefore revised to support both WFS services. Rather than return GetFeature KVP requests as initially implemented, the revised aviation semantic mediator returned an XML element that included the service endpoint and a list of AIXM feature type names that are semantically related to the queried user term. It is observed that the emerging OWS Context standard could be used for encoding the response from the aviation semantic mediator.

To query the Aviation semantic mediator, the following request can be sent. The user term is entered through the “query” input query parameter:

```
<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
```



```
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
  http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">

<ows:Identifier>com.envitia.ows9.cciaviation.CCIAviationProcess</ows:Iden
tifier>
  <wps:DataInputs>
    <wps:Input>
      <ows:Identifier>query</ows:Identifier>
      <wps:Data>
        <wps:LiteralData>Precision Approach Runway Category
II</wps:LiteralData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>
  <wps:ResponseForm>
    <wps:ResponseDocument storeExecuteResponse="false" lineage="false"
status="false">
      <wps:Output>
        <ows:Identifier>result</ows:Identifier>
        <ows:Title>result</ows:Title>
        <ows:Abstract>result</ows:Abstract>
      </wps:Output>
    </wps:ResponseDocument>
  </wps:ResponseForm>
</wps:Execute>
```

The response from the above request is shown next.

```
<?xml version="1.0" encoding="UTF-8"?>
<ns:ExecuteResponse xmlns:ns="http://www.opengis.net/wps/1.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_response.xsd"
    serviceInstance="http://someserver/wps/WebProcessingService?
REQUEST=GetCapabilities&amp;SERVICE=WPS"
    xml:lang="en-US" service="WPS" version="1.0.0">
    <ns:Process ns:processVersion="1.0.0">
        <ns1:Identifier xmlns:ns1="http://www.opengis.net/ows/1.1">
com.envitia.ows9.cciaviation.CCIAviationProcess</ns1:Identifier>
        <ns1:Title xmlns:ns1="http://www.opengis.net/ows/1.1">
com.envitia.ows9.cciaviation.CCIAviationProcess</ns1:Title>
    </ns:Process>
    <ns:Status creationTime="2012-11-09T11:35:52.837Z">
        <ns:ProcessSucceeded>The service successfully processed the
request.</ns:ProcessSucceeded>
    </ns:Status>
    <ns:ProcessOutputs>
        <ns:Output>
            <ns1:Identifier
xmlns:ns1="http://www.opengis.net/ows/1.1">result</ns1:Identifier>
            <ns1:Title
```



```
xmlns:ns1="http://www.opengis.net/ows/1.1">result</ns1:Title>
          <ns:Data>
             <ns:ComplexData mimeType="text/xml">
                <owc:offering
xmlns:owc="http://www.opengis.net/owc/1.0">
                   <owc:operation code="GetCapabilities" method="GET"

   href="http://demo.snowflakesoftware.com:8081/OWS-
9_AIXM51/OWS9_AIXM51?">
                      <owc:layer>aixm:Runway</owc:layer>
                      <owc:layer>aixm:RunwayElement</owc:layer>
                   </owc:operation>
                   <owc:operation code="GetCapabilities" method="POST"
                      href="http://91.221.120.150:13871/cadas-
aimdb/wfs">

   <owc:layer>aixm:RunwayCentrelinePoint</owc:layer>
                      <owc:layer>aixm:RunwayDirection</owc:layer>
                      <owc:layer>aixm:Runway</owc:layer>
                      <owc:layer>aixm:TouchDownLiftOff</owc:layer>
                      <owc:layer>aixm:RunwayElement</owc:layer>
                      <owc:layer>aixm:RunwayBlastPad</owc:layer>
                      <owc:layer>aixm:RunwayVisualRange</owc:layer>
                   </owc:operation>
                </owc:offering>
             </ns:ComplexData>
          </ns:Data>
       </ns:Output>
    </ns:ProcessOutputs>
</ns:ExecuteResponse>
```


11.2 Interaction with the Aviation Client

The user interface of the Aviation client, supplied by Luciad, contains a search field that allows a user to fill in the pilot's terms as illustrated in Figure 9. A user enters a term and presses enter. Upon pressing enter, the Semantic Mediator WPS is contacted to resolve the WFS data sources / AIXM feature types. The request sent to the WPS by the client contains only the user term. The Semantic Mediator WPS contacts the SPARQL Server to retrieve semantic mappings and then resolves the user term to a collection of AIXM feature types. The Semantic Mediator WPS then parses the GetCapabilities documents of the referenced WFS (i.e. one from Snowflake Software and another from COMSOFT) to determine which feature types identified by the semantic mappings are also offered by the WFSs. The resulting WFS / AIXM links are resolved by using an optional OGC Filter, and the result is visualized on a map as illustrated in Figure 10. The optional OGC Filter can be configured in the client UI.

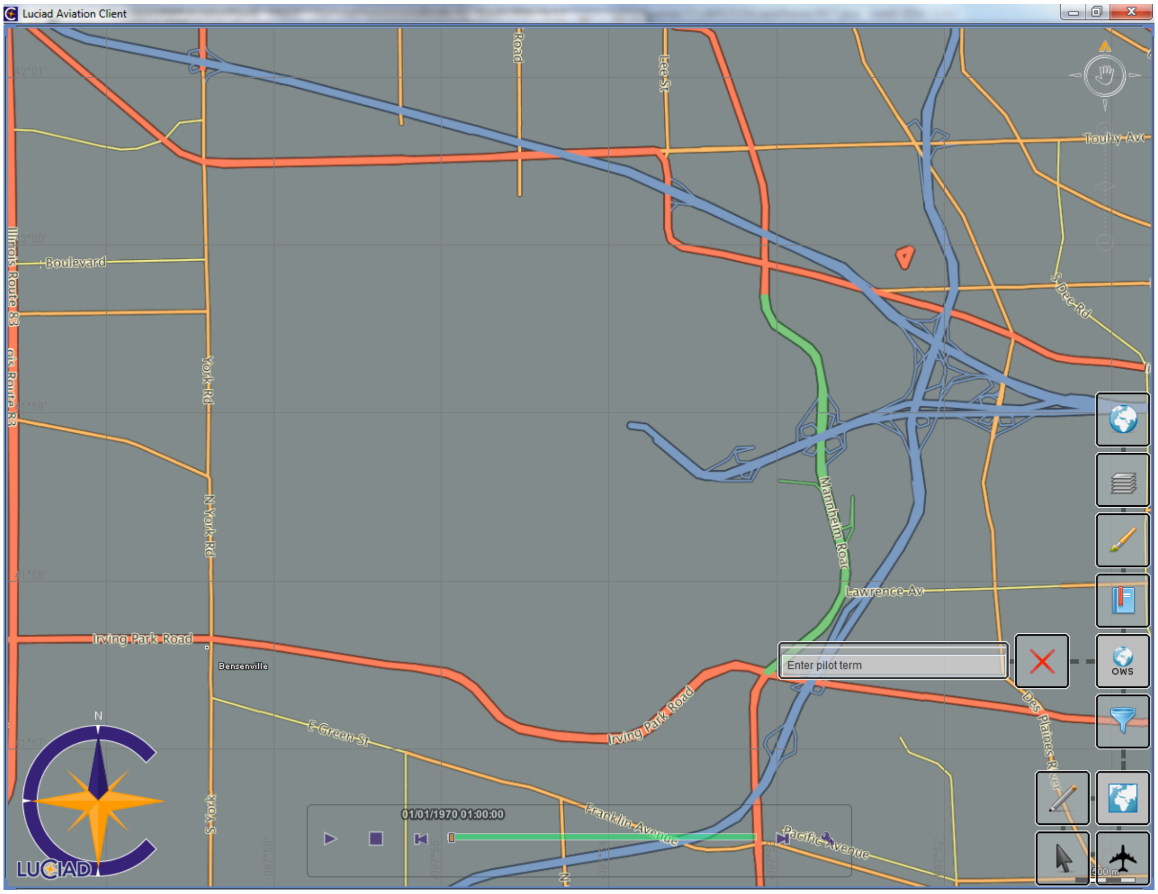

Figure 9. Screenshot of the Aviation Client at standby, awaiting a user to enter a term

The illustration in Figure 10 shows the result of a search for "High Speed Taxiway" by a user. The search returns aixm:TaxiwayElement features amongst others, which are then portrayed on the client.

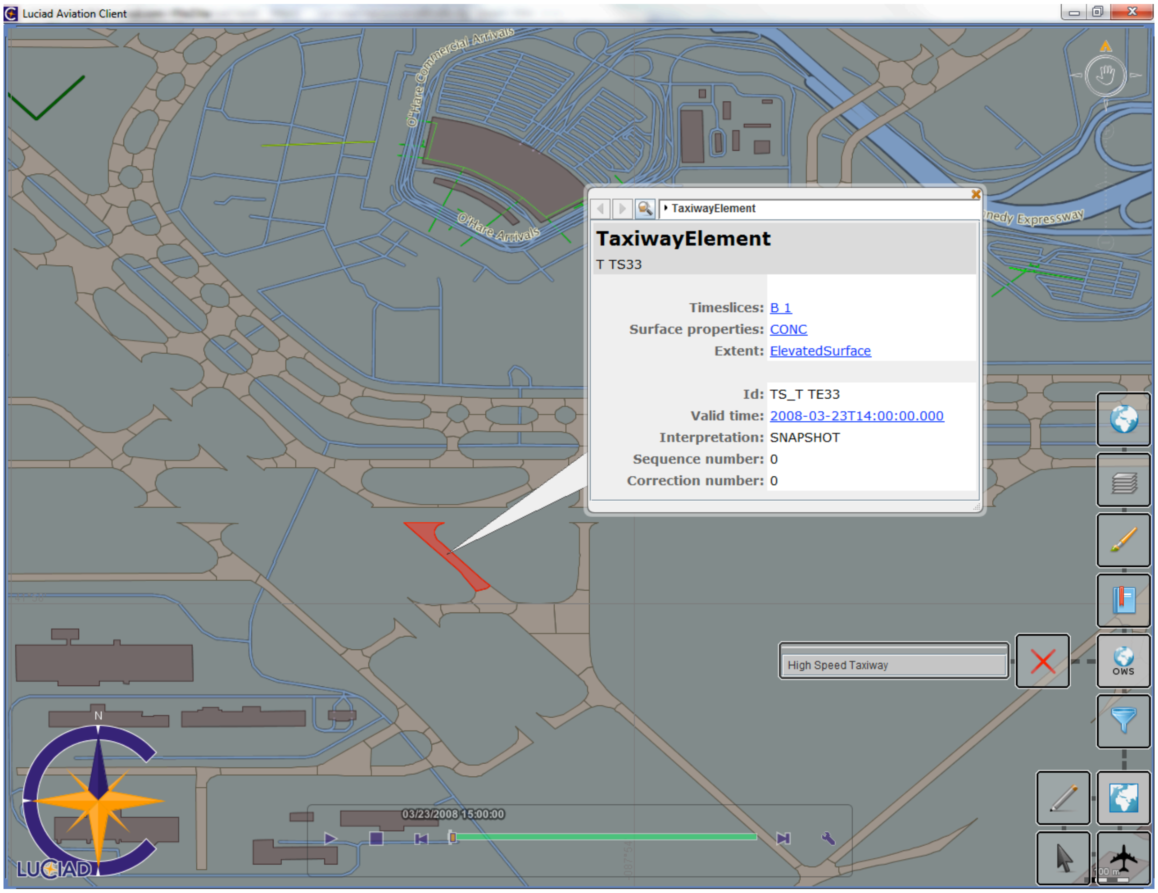

Figure 10. The aviation client returning an aixm:TaxiwayElement from a query of High Speed Taxiway

12 WPS for Geoparsing VGI

One of the tasks of the CCI thread was to demonstrate semantic mediation involving volunteered geographic information (VGI). Twitter was selected as one of the sources of the VGI. Twitter is an online social networking service and microblogging service that enables its users to send and read text-based messages, of up to 140 characters, known as "tweets". Although Twitter allows applications to attach geographic coordinates to tweets, applications seldom include their geo-locations in tweets. The consequence is that it becomes challenging to portray tweets geospatially on a map. To overcome this challenge the OWS-9 CCI architecture included support for geoparsing tweets. Geoparsing is the process of assigning geographic coordinates to textual words and phrases in free texts; it is a subclass of Geocoding, which refers to the translation of an address or structured text to geospatial coordinates.

The CCI thread considered two separate approaches for geoparsing tweets. The first approach considered involved testing for placenames on the cascading WFS-G. This included breaking each tweet into its constituent words, removing stop-words and then sending each word to the cascading WFS-G through a WFS GetFeature request that includes multiple ‘Or’ constraints in a Filter. The cascading WFS-G would then return any placenames it detects from the words sent to it. An example WFS-G request constructed from a tweet of “Road blocked because of flooding in Port-au-Prince, Haiti” is shown in the following listing. Sending words from tweets to the cascading WFS-G resulted in errors, which were attributed to unexpected characters in the content of tweets. Due to this unpredictability of tweets, the thread then considered an alternative approach for geocoding tweets.

```
<GetFeature
    xmlns="http://www.opengis.net/wfs"...>
<Query typeName="iso19112:SI_LocationInstance"
srsName="urn:ogc:def:crs:EPSG::4326">
<ogc:Filter>
<ogc:Or>
      <ogc:PropertyIsEqualTo>

<ogc:PropertyName>alternativeGeographicIdentifier</ogc:PropertyName>
        <ogc:Literal>Road</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>

<ogc:PropertyName>alternativeGeographicIdentifier</ogc:PropertyName>
        <ogc:Literal>blocked</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>

<ogc:PropertyName>alternativeGeographicIdentifier</ogc:PropertyName>
        <ogc:Literal>flooding</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>

<ogc:PropertyName>alternativeGeographicIdentifier</ogc:PropertyName>
        <ogc:Literal>Port-au-Prince</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>

<ogc:PropertyName>alternativeGeographicIdentifier</ogc:PropertyName>
        <ogc:Literal>Haiti</ogc:Literal>
      </ogc:PropertyIsEqualTo>
</ogc:Or>
</ogc:Filter>
</Query>
</GetFeature>
```

The second approach considered, which was successfully implemented by the CCI thread, involved checking for placenames on the WPS itself. This second approach followed the following sequence of steps.

1. A subset of the NGA GNS, using only Haiti locations, was cached during initialization of the WPS. This is done before query time to ensure that processing at query time is achievable in reasonable time.

2. When a user sends a request to the WPS, the current Tweets with the word Haiti are retrieved from Twitter either through the OpenGeo WFS or through the Twitter API.

3. Each word in the tweet is tested against each placename in the cached GNS subset

4. If a placename is found in the tweet, a new feature is created using the GNS geometry, GNS placetype, twitter username and tweet creation date.

5. The GNS placetype is then used to retrieve its USGS GNIS matches using the semantic mediator

6. The GNIS placetypes are also added to the created feature

The following listing shows example features generated from geoparsed tweets.


```
       <topp:geotweet gml:id="geotweet.3">
          <topp:the_geom>
             <gml:Point srsName="urn:x-ogc:def:crs:EPSG:4326">
                <gml:pos>18.91 -72.58</gml:pos>
             </gml:Point>
          </topp:the_geom>
          <topp:placetype>urn:x-
ogc:def:locationType:NGA:PPL</topp:placetype>
<topp:placetype>
http://www.usgs.gov/projects/ows9#usgs.gnis.populated_place
</topp:placetype>
          <topp:placename>Jean</topp:placename>
          <topp:user>ncl svy002231080</topp:user>
          <topp:created>Tue Dec 18 11:40:06 GMT 2012</topp:created>

          <topp:text>Chapelle Jean Bosco  in haiti inundated</topp:text>
       </topp:geotweet>

          <topp:geotweet gml:id="geotweet.0">
          <topp:the_geom>
             <gml:Point srsName="urn:x-ogc:def:crs:EPSG:4326">
                <gml:pos>19.41 -71.98</gml:pos>
             </gml:Point>
          </topp:the_geom>
          <topp:placetype>urn:x-
ogc:def:locationType:NGA:PPL</topp:placetype>
<topp:placetype>
http://www.usgs.gov/projects/ows9#usgs.gnis.populated_place
</topp:placetype>
          <topp:placename>Harry</topp:placename>
          <topp:user>ncl svy002231080</topp:user>
          <topp:created>Tue Dec 18 11:40:21 GMT 2012</topp:created>

          <topp:text>Blvd Harry Truman and rue  de docteur dehoux in
haiti - big crack cars cannot pass </topp:text>
```



```
        </topp:geotweet>

        <topp:geotweet gml:id="geotweet.1">
            <topp:the_geom>
                <gml:Point srsName="urn:x-ogc:def:crs:EPSG:4326">
                    <gml:pos>19.41 -71.98</gml:pos>
                </gml:Point>
            </topp:the_geom>
            <topp:placetype>urn:x-
ogc:def:locationType:NGA:PPL</topp:placetype>
<topp:placetype>
http://www.usgs.gov/projects/ows9#usgs.gnis.populated_place
</topp:placetype>
            <topp:placename>Harry</topp:placename>
            <topp:user>ncl svy002231080</topp:user>
            <topp:created>Tue Dec 18 11:53:41 GMT 2012</topp:created>

            <topp:text>congestion Harry Truman in haiti dehox</topp:text>
        </topp:geotweet>
```


12.1 Lessons Learnt: Multilingual VGI Geoparsing

It was observed that the use of ontologies in semantic mediation is agnostic to languages. This was evidenced by the ability of the geoparsing WPS to detect placenames in both English and French. Although not fully explored in OWS-9, multilingual VGI geoparsing could prove a useful capability in international spatial data programmes such as the European Union's INSPIRE or the Global Earth Observation System of Systems (GEOSS).

13 GEO Discovery and Access Broker

The GEO Discovery and Access Broker (DAB) framework was developed in the EuroGEOSS research programme of the European Commission (www.eurogeoss,.eu) under the leadership of the National Research Council of Italy (CNR), with contributions from the European Commission's Joint Research Centre (JRC) (for the semantic component). The DAB is currently operated by the CNR with the support of the JRC for the vocabulary services and semantic component. It is now a part of the GEOSS Common Infrastructure (GCI) and provides intermediation services between various capabilities and information systems. The GEO DAB framework consists of the following components: Discovery Broker (with an extension for semantic augmentation), and Access Broker.

Within OWS-9, the DAB was configured to apply the semantic mappings between NGA TDS and USGS TNM data. This enabled the validation of the approach adopted in OWS-8. The DAB was able to successfully interact with the SPARQL Server, retrieving the semantic mappings and applying them to transformations on NGA TDS and USGS TNM data. The DAB was also able to successfully interact with the integrated client supplied by Pyxis Innovation.

The GEO Access Broker makes it possible for users to access and use the datasets resulting from their queries according to a common grid environment they have configured by selecting the following common features: Coordinate Reference System (CRS), spatial resolution, spatial extent (e.g., subset), and data encoding format. This feature is crucial to allow effective integration and analysis of data coming from heterogeneous sources. In normal practice, the manual manipulation to the data ahead of analysis is necessary. The GEO Access Broker is designed to reduce this burden from the user by providing some automation of data manipulation tasks.

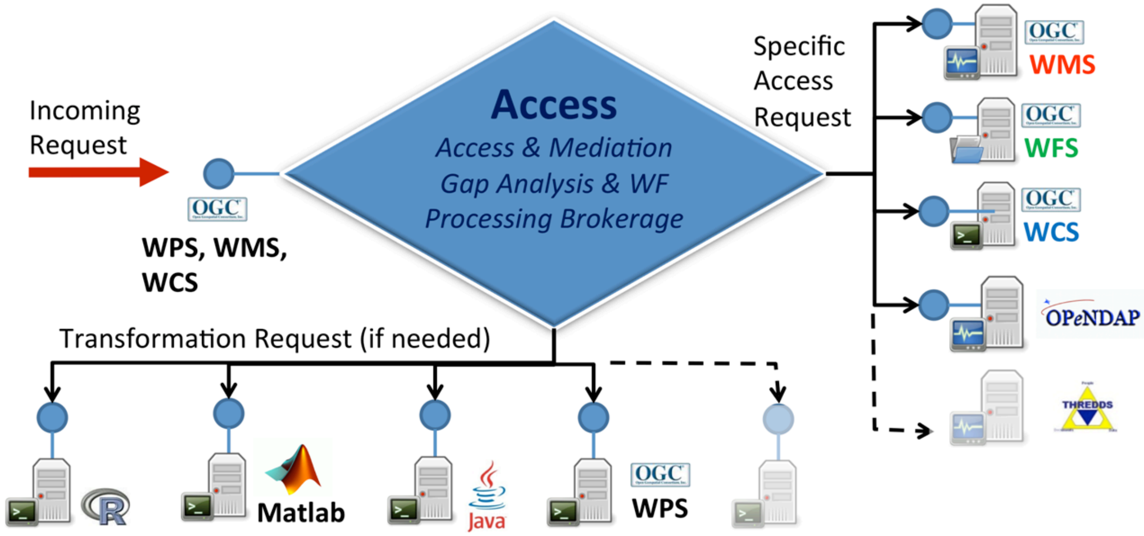

Figure 11. Simplified diagram of the GEO Access Broker

In keeping with the System-of-Systems (SoS) principles, the GEO Access Broker carries out this task by supplementing, but not supplanting, the access services providing the datasets requested. This is achieved by brokering the necessary transformation requests (those that the access services are not able to accomplish) to external processing services. Figure 11 depicts a simplified diagram of the GEO Access Broker.

In OWS-9, the GEO Access Broker was enhanced to provide a new transformation type: the Feature Schema Transformation. This way, client applications can access feature data from heterogeneous OGC WFS servers using a single feature schema for the returned features. Figure 12 depicts the enhanced framework of the GEO Access Broker after the developments in OWS-9.

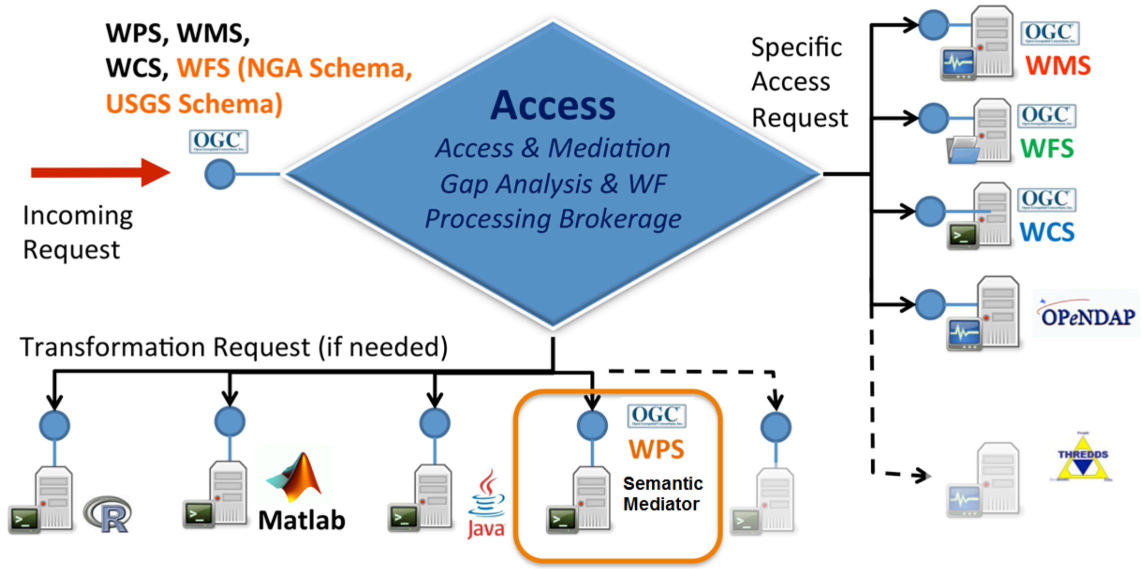

Figure 12. Simplified diagram of the GEO Access Broker with OWS-9 enhancements

The list of available interfaces to submit requests to the Access Broker was extended to support OGC WFS. The schema transformation rules are provided by the WPS SPARQL Service (Envitia). This is invoked when the Access Broker identifies the need to transform feature schema. The two schemas considered in OWS-9 were the USGS TNM schema and the NGA TDS schema.

14 Integrated Client

The PYXIS WorldView browser was provided for interacting with the various services forming the OWS 9 architecture. WorldView runs as a standalone application that includes a component for searching and a 3D globe for portrayal. Once the application launches, initial searches adopt the bounding coordinates of the visible area, which is the whole world at startup. It was observed that running a search when the complete globe is in view triggers a request with whole world bounding coordinates (-180,-90,180,90). The semantic mediator and the USGS and NGA gazetteer services tend to take a longtime (longer than a minute) to respond for bounding boxes of the whole world because the services offer millions of records. To ensure that the services respond in reasonable time, the browser was configured to request a specific maximum number of features. The semantic mediator includes the requested maximum number of features in the requests sent to the USGS and NGA gazetteer services thereby ensuring that the client has control over the performance of the services.

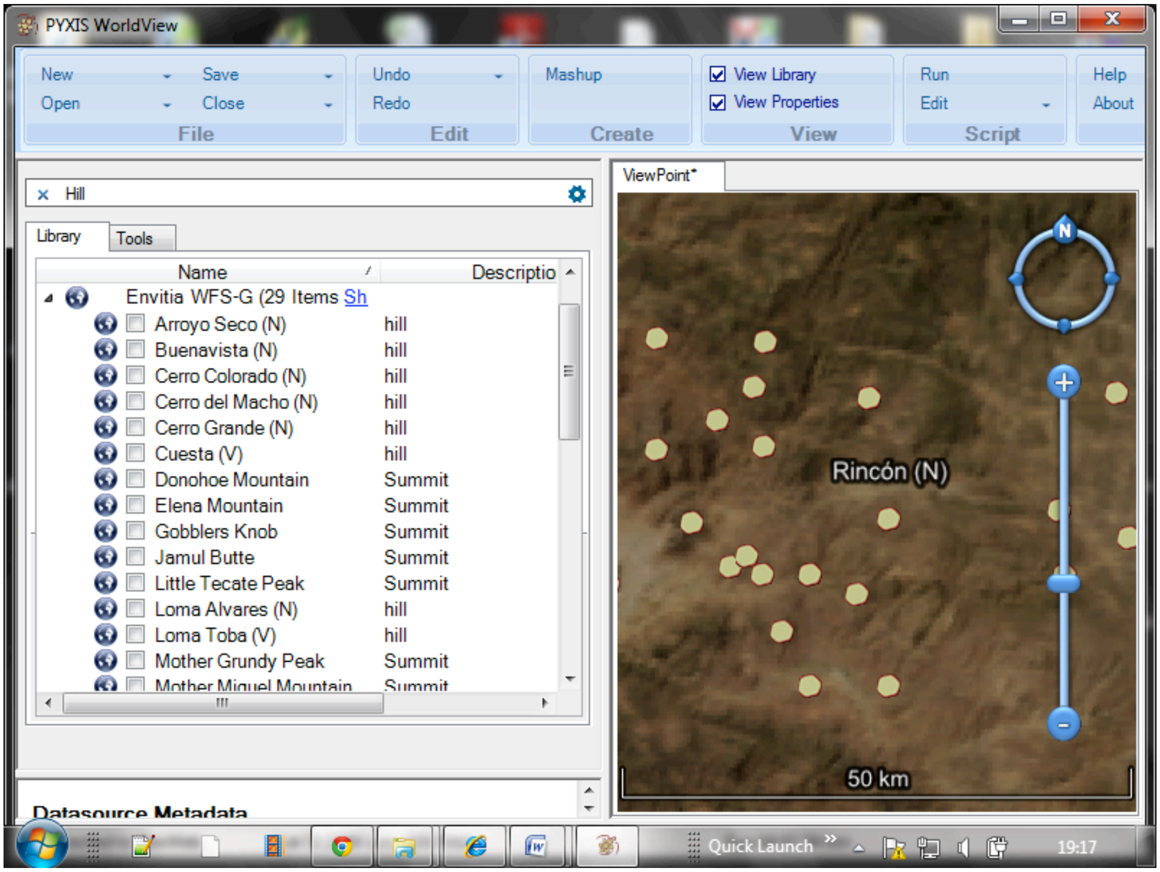

Figure 13. Screenshot of the PYXIS WorldView browser

15 Scenarios

The CCI thread designed a Geonames scenario to demonstrate the WFS-G capabilities researched during this testbed. A separate scenario was designed by the wider OWS-9 community for demonstrating Aviation capabilities, including the semantic mediation between user terms and AIXM feature types. In this report, we present only the Geonames scenario. The aviation scenario is described in the Aviation Architecture engineering report [OGC 12-151].

15.1 Geonames Scenario

The following scenario was designed for demonstrating the geonames aspect of CCI. The scenario includes a demonstration of the application of semantic mediation to WFS-G.

1. SAR planner receives a report of a crash on a mountain named Rincin in Mexico
2. SAR planner initiates an exact query to find all mountains named Rincin.
   a. This is a misspelling ... the correct spelling is Rincon with an accent on the ‘o’

   b. This is just a query on the name

3. Portal returns no results
4. SAR planner finds crash in Mexico initiates a fuzzy query to find all mountains named Rincin in Mexico.
   a. Constrain query to administrative area ... Mexico
   b. Use USGS feature term 'Summit' for query
5. Portal returns Rincon (with accent on o)
   a. Show fuzzy query with correct display of Unicode characters
6. SAR planner looks for nearby ranches within bounding box around Rincon
   a. Constrain query to bounding box [-116.789, 32.189,-115.875, 32.874]
   b. Use NGA feature term 'RNCH' for query
7. Portal issues queries for ranches
8. SAR planner queries Rincon (with diacritics to get coordinates)
   a. Demonstrates ability to enter Unicode characters
9. Portal returns name and coordinates
10. SAR planner queries closest airfield to Rincon
    a. Use coordinates from Rincon
    b. Use USGS feature term 'Airport' for query
11. Portal returns closest airfield
12. SAR planner receives request for elevation details on nearby mountains in order to set up communications tower
13. SAR planner queries all mountains within 25 miles of Rincon mountain
    a. Use USGS feature term 'Summit' for query
    b. Radial search query
14. Portal returns list of mountains

15. SAR planner modifies query to obtain elevations for mountains by intersecting locations with elevation data
    a. Leverages output of previous query as input to elevation data service via WPS
16. WPS returns list of mountains with elevations
17. SAR planner concludes by showing map of Rincon, airports, mountains (with elevations) and ranches

The following sections describe the use cases adopted in implementing the scenario.

15.2 WFS-G Semantic Mediation Use Case

The following sequence diagram describes the interaction between components involved in the semantic mediation of WFS-G through a semantically-assisted cascading WFS-G.

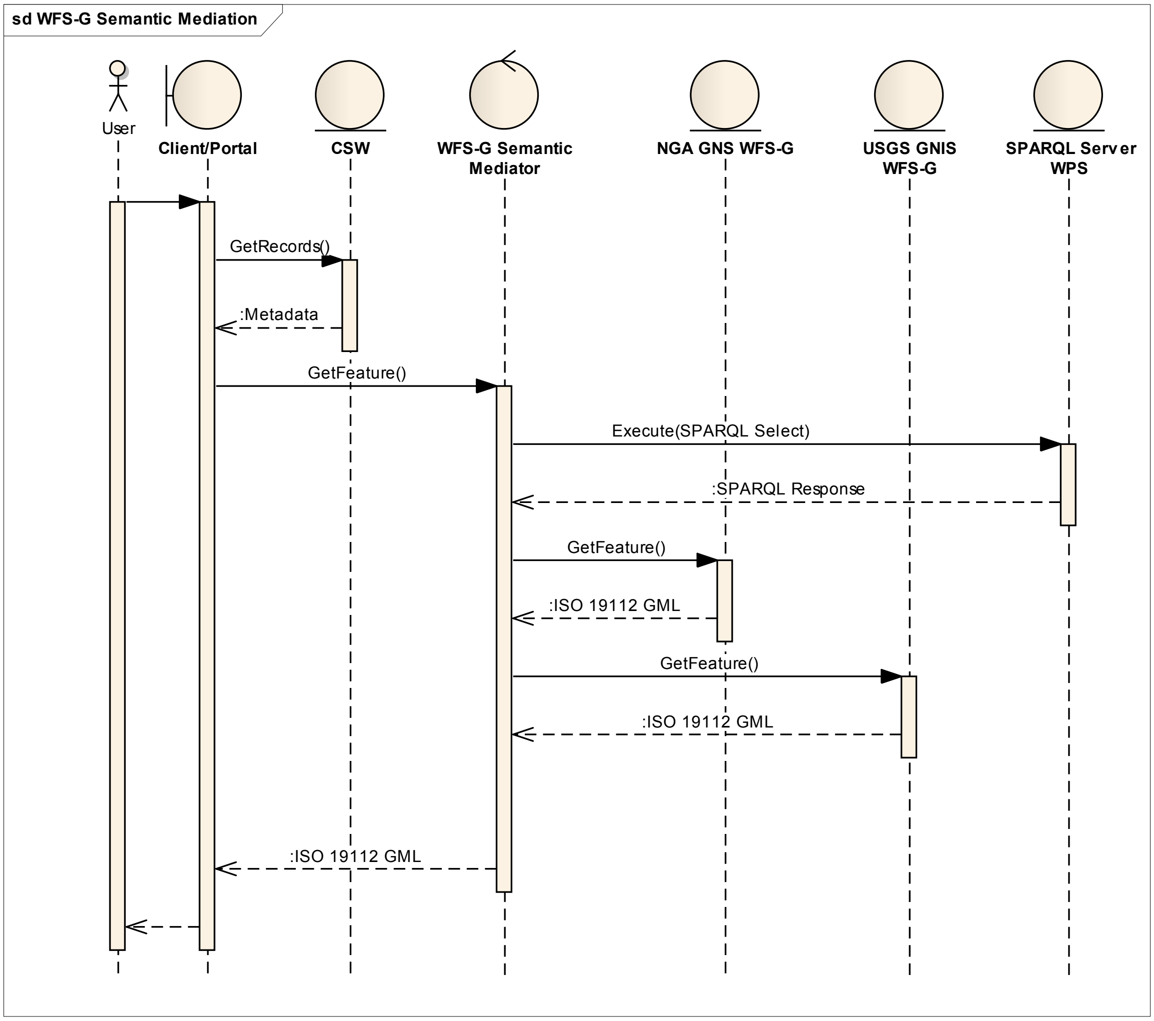

Figure 14. Sequence diagram for the WFS-G Semantic Mediation approach

15.3 WPS Semantic Mediation

The following sequence diagram describes the interaction between components involved in the semantic mediation of AIXM WFS through a WPS. The primary difference between the WFS-G and WPS Semantic Mediation approaches presented is that the former returns data from the underlying services whereas the latter only returns metadata about where data can found. The latter approach therefore requires the client to access the data directly once the semantic mediator has identified where the data can be retrieved from.

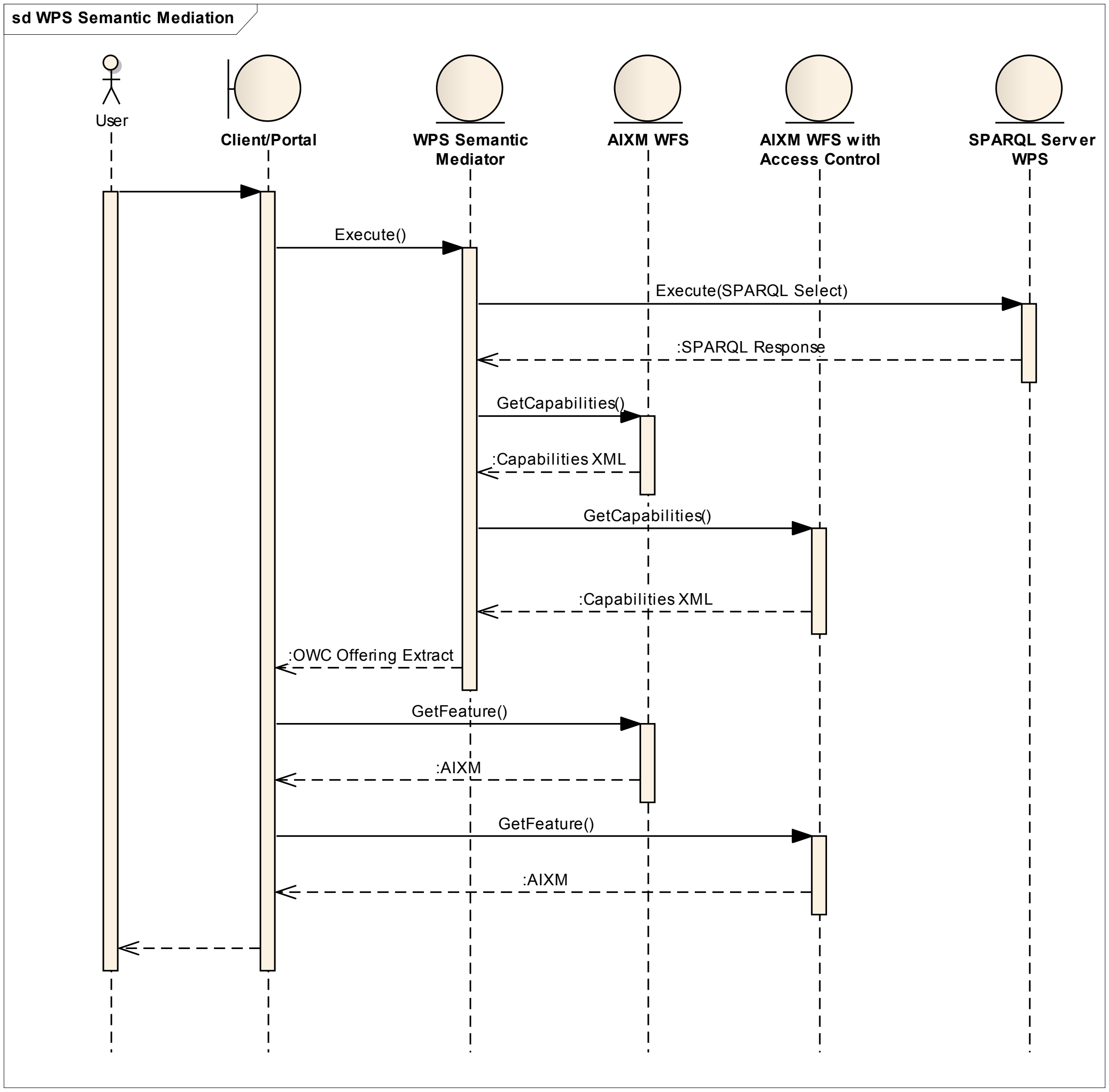

Figure 15. Sequence diagram for the cross-thread CCI-Aviation WPS Semantic Mediation approach

15.4 Broker Semantic Mediation Use Case

Figure 16 depicts the sequence diagram representing the steps executed to access data from the NGA WFS (providing TDS data) through queries using the USGS TNM schema. In this case, the diagram does not show the access to USGS WFS since this does not involve any schema transformation. When the request is submitted, the Access Broker retrieves and applies the mapping rules to generate an appropriate request for sending to the NGA WFS. The generated request is submitted to the NGA WFS; if necessary, the retrieved feature members are then further processed to match user's request – e.g. the user might have requested an output with CRS not supported by NGA WFS, in this case the Access Broker can invoke the appropriate external CRS transformation service (these steps are not captured in the diagram because they are not relevant to semantic mediation functionality). Finally, the schema transformation is performed on each feature member by retrieving and applying the transformation rules.

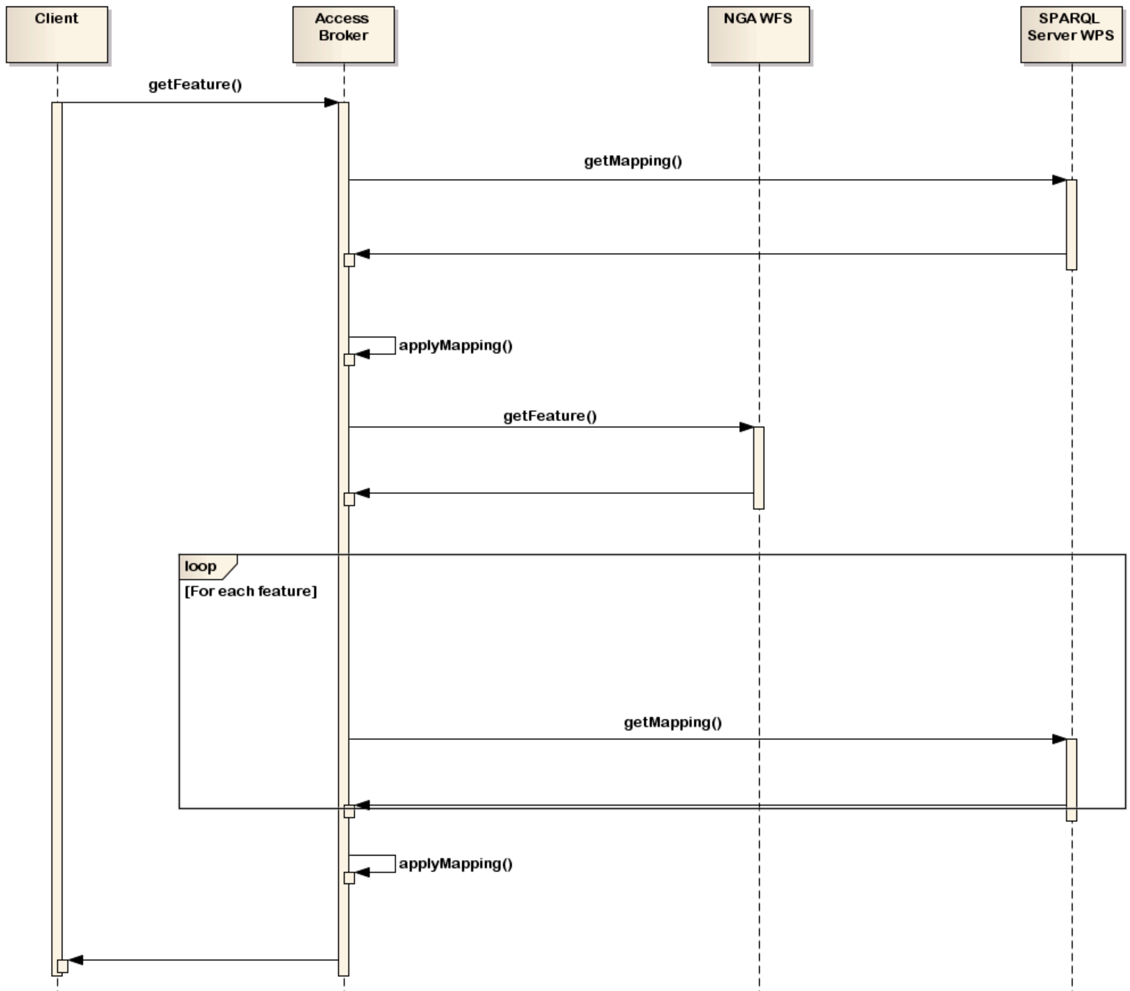

Figure 16. Sequence Diagram for the GEO Access Broker

16 Conclusions

The CCI thread was able to demonstrate enhancements to the semantic mediation architecture through inclusion of an integrated client, WFS-G services and a broker component. The thread demonstrated semantic mediation between WFS-G services and proposed a filter comparison operator to support semantics in OGC web services. The thread also demonstrated use of a broker to provide semantic mediation capabilities. Also demonstrated was semantic mediation between the FAA pilots' glossary and AIXM data. With the successful implementation of the OWS-9 CCI architecture and demonstration of the semantic mediation capability, this report concludes that OGC web services can provide semantic mediation through the architecture described in this report. The next section presents a series of recommendations to enable and advance this capability.

17 Recommendations

The following recommendations are made by this report.

17.1 Additional metadata for the WPS Process descriptions

Section 8.5 recommends a change to WPS Process Descriptions to allow for additional metadata to be presented by a WPS DescribeProcess response.

17.2 Support for semantics in OGC Filters

Section 10.6 recommends a change to OGC Filters to include a comparison operator called PropertyIsSemanticallyRelatedTo.

17.3 Metadata for describing fuzzy match algorithms

Section 10.4 of this report recommends that metadata identifying the implemented fuzzy match algorithm be presented in the GetCapabilities response.

17.4 Use of StoredQueries as values of location types href attributes

Section 10.5 of this report recommends that the WFS-G Best Practice document should recommend the use of StoredQueries instead of XML encoded filters as values of the location type href attribute.

17.5 Possible WPS Profile for SPARQL Servers

This report recommends the definition of a WPS Profile for SPARQL Servers as defined in section 8.4.

17.6 Future Work

Both OWS-8 and OWS-9 have successfully demonstrated semantic mediation of datasets supplied by WFS. An issue raised by participants is that the link-structure of semantic web documents creates a dependency between them that could affect a system if a single document is removed from a document collection. Future work should therefore consider strategies for resilience of systems offering semantic mediation capabilities.

Another issue raised by participants is the potential for WFS-G to publish data in the new GeoSPARQL standard. Amongst the benefits of using GeoSPARQL to support WFS-G is the potential to specify containment of places within one another in a transitive manner, for example: a SPARQL Server would be able to determine that if Location A is inside

Location B, and Location B is inside Location C, then Location A must be inside Location C. Future testbeds should therefore explore the potential for exporting GeoSPARQL from WFS-G and its exploitation in WPS-enabled SPARQL Servers.

The work undertaken within this testbed demonstrated the use of WPS for geoparsing VGI. One of the challenges of VGI in the form of tweets is that placenames are often abbreviated due to the 140 character limit. Future work should therefore explore transactional capabilities in WFS-G that could allow for abbreviations of placenames to be derived from VGI and added to WFS-G, automatically, based on the placenames used overtime. Such a capability could involve use of the Rule Interchange Format (RIF) to support placename detection.

18 Acknowledgements

The authors acknowledge the support of Deborah L. J. Cowell (FAA) and Candice Buchanan (Nexa Corporation) on the FAA Air Transportation Information Ontology.

The authors also acknowledge the support of Doug Caldwell (US Army Corps of Engineers), Trent Palmer(NGA), Louis Yost (USGS), Glenn Guempel (USGS) and Paul Birkel(NGA/Mitre) on the semantic mapping of gazetteers.