PlanetData

Network of Excellence

FP7 – 257641

D16.4 Call 2: Linked Map data access/update service

Coordinator: Jesús Barrera (GEOSLAB)
With contributions from: Francisco J Lopez-Pellicer (UNIZAR)
1st Quality reviewer: Luciano Serafini (FBK)
2nd Quality reviewer: Davor Orlic (JSI)

Deliverable nature:	Prototype (P)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	M42
Actual delivery date:	M44
Version:	1.0
Total number of pages:	16
Keywords:	Linked data, geographic information, maps, service, SPARQL endpoint

Abstract

This deliverable describes the deployment of a SPARQL node that provides access/update to the data processed in the project which are stored in a triplestore compliant with OGC GeoSPARQL specification.

Executive summary

The Work Package 16 of the PlanetData project includes the setup of a read-write SPARQL endpoint supporting GeoSPARQL specification for the Linked Map subproject. The OGC GeoSPARQL standard supports representing and querying geospatial data on the Semantic Web. GeoSPARQL defines a vocabulary for representing geospatial data in RDF, and it defines an extension to the SPARQL query language for processing geospatial data. This deliverable describes how such SPARQL endpoint has been setup and how it interacts with other parts of the Linked Map Project. The software used for the deployment of this endpoint is Strabon. The document explains the installation and configuration of Strabon, as well as how this endpoint will be used by the Linked Map Service, that is, the service built by Work Package 17 of the PlanetData project for the Linked Map subproject.

Document Information

IST Project Number	FP7 - 257641	**Acronym**	PlanetData
Full Title	PlanetData		
Project URL	http://www.planet-data.eu/		
Document URL	http://wiki.planet-data.eu/web/D16.4		
EU Project Officer	Leonhard Maqua		

Deliverable	**Number**	D16.4	**Title**	Call2: Linked Map data access/update service
Work Package	**Number**	WP16	**Title**	Call2: Linked Map Linked data provisioning

Date of Delivery	**Contractual**	M42	**Actual**	M44
Status	version 1.0		final ☒	
Nature	prototype ☒ report □ demonstrator □ other □			
Dissemination level	public ☒ restricted to group □ restricted to programme □ consortium □			

Authors (Partner)	Jesús Barrera (GEOSLAB), Francisco J Lopez-Pellicer (UNIZAR)			
Responsible Author	**Name**	Jesús Barrera	**E-mail**	jesusb@geoslab.com
	Partner	GEOSLAB	**Phone**	+34 976 065152

Abstract (for dissemination)	This deliverable describes the deployment of a SPARQL node that provides access/update to the data processed in the project which are stored in a triplestore compliant with OGC GeoSPARQL specification.
Keywords	Linked Data, geographic information, maps, service, SPARQL endpoint

Version Log			
Issue Date	**Rev. No.**	**Author**	**Change**
2014/04/01	0.1	Jesús Barrera Francés	Template instantiation
2014/04/15	0.2	Jesús Barrera Francés	First version
2014/04/25	0.3	Francisco J Lopez-Pellicer	Contributions
2014/04/28	0.4	Jesús Barrera Francés	Internal revision
2014/04/29	0.5	Jesús Barrera Francés	First draft
2014/05/08	0.6	Francisco J Lopez-Pellicer	Draft ready for QA
2014/05/21	0.7	Jesús Barrera Francés	Adjustments after reviewers' comments
2014/05/26	1.0	Francisco J Lopez-Pellicer	Final release

Table of Contents

Abbreviations

ACL	Access Control List
BCN25	Base Cartográfica Numérica 1:25.000 (Numeric Cartographic Base 1:25.000)
BTN25	Base Topográfica Nacional 1:25.000 (National Topographic Base 1:25.000)
LMS	Linked Map Service
OGC	Open Geospatial Consortium
VGI	Volunteer Geographic Information
WMS	Web Map Service

List of figures

1 Introduction

The Linked Map subproject of the PlanetData project is carrying out the conversion of a set of official governmental data to Linked Data and the linking of the resulting dataset to Volunteer Geographic Information (VGI) sources (such as OpenStreetMap and Wikipedia). The aim of this work is to demonstrate the feasibility of Linked Data as a useful infrastructure to address a complex problem: the recording of the provenance and reliability metadata about VGI data in order to develop a set of metrics that may assist information producers to the evaluation of the quality of VGI data. This task will be accomplished in the Linked Map subproject using a web portal that will allow volunteers to review the reliability of VGI data. These quality reviews will be integrated into the data collection as Linked Data, enriching the final report on the quality of VGI data.

This deliverable is part of the Work Package 16. This work package is responsible for the provisioning of an official geographic dataset and several VGI sources, their transformation into RDF data, their integration using RDF links and their publication through a SPARQL endpoint. The subject of the document is the latter task: setting up a read-write endpoint supporting the OGC GeoSPARQL specification [1], which is an extension to the SPARQL protocol [2]. To achieve this goal, we have used the semantic database Strabon [3] which supports the querying of data using the aforementioned standard. This endpoint is the bridge between the work done in the Work Package 16 and the data access services built in the Work Package 17 that includes the Linked Map Service (LMS), that is, a kind of service created in the context of the project that addresses the interoperability between map services and Linked Data (this service is fully described in the deliverable D17.1 [4]).

Figure 1 describes the conceptual architecture of the Linked Map project and the relations between the portal, the LMS instance, external Web Map Services [5] (proxied by the LMS instance) and the GeoSPARQL endpoint.

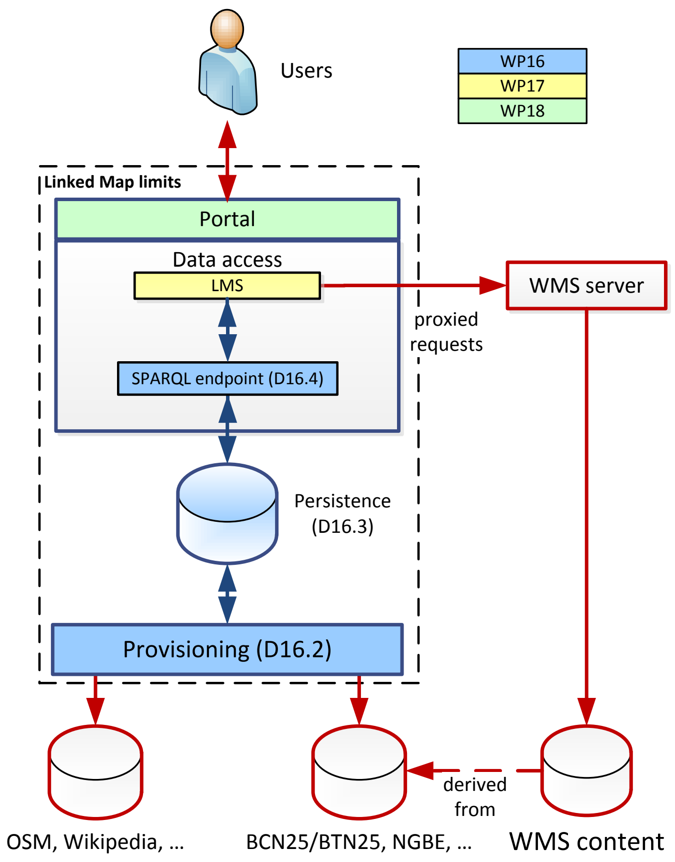


Figure 1 – Linked Map overview

The following sections describe the steps followed in order to set up a GeoSPARQL endpoint using Strabon and how this endpoint acts as backend of a LMS instance.

2 Setting up a SPARQL endpoint

In the deliverable D16.2 [6], we describe the persistence system selected for the storage of the data which were going to be processed in the project and transformed into RDF. This persistence system should be able to manage complex geometries expressed in GeoSPARQL. Hence, the use of storage systems that support GeoSPARQL natively or via extensions should be considered in the project. The storage system of choice is Strabon. Strabon is a semantic spatiotemporal RDF store developed in the context of the European FP7 project SemsorGrid4Env[1] (*Semantic Sensor Grids for Rapid Application Development for Environmental Management*).

As Figure 2 shows, Strabon is built by extending the well-known RDF store Sesame[2] to manage thematic, spatial and temporal data that are stored in a backend repository in a PostgreSQL database extended with the PostGIS module [7]. PostGIS is an extension of PostgreSQL that adds support for storing and querying geographic objects.

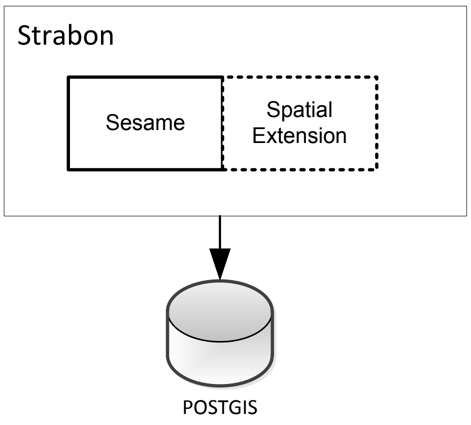


Figure 2 – Strabon architecture overview

In the context of the Linked Map project, a Strabon SPARQL endpoint has been deployed in order to provide a GeoSPARQL endpoint. Although the Strabon website contains a pretty detailed installation guide for Debian/Ubuntu[3], it has been necessary to make some changes to adapt the installation to CentOS.

First, we install third party software needed by the installation: Java 6, Maven, Apache Tomcat, Mercurial.

```
$> yum install maven2
$> yum install java-1.6.0-openjdk.x86_64
$> yum install tomcat6.noarch
$> yum install mercurial.x86_64
```

The next step is the installation of PostgreSQL and PostGIS. It is necessary to make some extra steps[4] as the PostgreSQL 9.1 RPM package is not in the official package repository of CentOS. First, the official `yum` repository must be configured to ignore searches related to PostgreSQL. This is achieved by editing the file `/etc/yum.repos.d/CentOS-Base.repo` and adding the string `exclude=postgresql*` to the `base` and `update` sections. Next, the appropriate PostgreSQL RPM package file should be downloaded

[1] http://www.semsorgrid4env.eu/

[2] http://www.openrdf.org/

[3] http://www.strabon.di.uoa.gr/UserGuide

[4] https://wiki.postgresql.org/wiki/YUM_Installation

from `http://yum.postgresql.org/repopackages.php#pg91`. Once the RPM file is downloaded, it is installed in added to the local repository using the RPM manager:

```
    $> rpm -ivh pgdg-centos91-9.1-4.noarch.rpm
```

At this moment we are able to install PostgreSQL 9.1 and PostGIS 1.5 in the server:

```
    $> yum install postgresql91.x86_64
    $> yum install postgresql91-server.x86_64
    $> yum install postgis91.x86_64
```

Now we need to configure the database. The following steps require using the user postgres:

```
    $> sudo su postgres
```

The next step is setting a password to the user `postgres`:

```
    $> sudo -u postgres psql -c "ALTER USER postgres WITH PASSWORD
    'postgres';"
```

To enable spatial features, we must follow the next steps:

- Configuring the environment variable POSTGIS_SQL_PATH pointing to the folder where PostGIS 1.5 has been installed.

```
    $> POSTGIS_SQL_PATH=usr/pgsql-9.1/share/contrib/postgis-1.5
```

- Creating a database named `template_postgis` that will be used as template for spatial databases.

```
    $> createdb -E UTF8 -T template0 template_postgis
```

- Enabling the use of the SQL Procedural language in such template.

```
    $> createlang -d  template_postgis plpgsql
```

- Loading PostGIS routines in the template database.

```
    $> psql -d template_postgis -f $POSTGIS_SQL_PATH/postgis.sql
    $> psql -d template_postgis -f $POSTGIS_SQL_PATH/spatial_ref_sys.sql
```

- Grating users to alter tables that contain metadata about spatial columns and reference systems.

```
    $> psql -d template_postgis -c "GRANT ALL ON geometry_columns TO
    PUBLIC;"
    $> psql -d template_postgis -c "GRANT ALL ON geography_columns TO
    PUBLIC;"
    $> psql -d template_postgis -c "GRANT ALL ON spatial_ref_sys TO
    PUBLIC;"
```

- Performing garbage collection for optimizing the template `template_postgis`.

```
$> psql -d template_postgis -c "VACUUM FULL;"
$> psql -d template_postgis -c "VACUUM FREEZE;"
```

- Grating non-superusers to create databases using the template `template_postgis`.

```
$> psql -d postgres -c "UPDATE pg_database SET datistemplate='true'
WHERE datname='template_postgis';"
$> psql -d postgres -c "UPDATE pg_database SET datallowconn='false'
WHERE datname='template_postgis';"
```

At this point we can create a spatial enabled database named `endpoint` cloning the template `template_postgis.`

```
$> createdb endpoint -T template_postgis
```

Now, the PostgreSQL database is enabled for providing spatial support to Strabon.

The next step is to deploy the SPARQL endpoint in Apache Tomcat. First, we need to add a user to Apache Tomcat by editing `tomcat-users.xml` file. We should add to that file the following line:

```
<user username="endpoint" password="endpoint" roles="manager"/>
```

Now we are ready to install Strabon from the sources:

1. Create a directory where we want to place the sources of the Strabon on our development machine, and `cd` into that directory, e.g.:

```
$> mkdir Strabon
$> cd Strabon
```

2. Get a clone of Strabon from the Strabon repository:

```
$> hg clone http://hg.strabon.di.uoa.gr/Strabon/
```

3. The source code of the endpoint is located at the folder Strabon of the source code tree that we just cloned, so `cd` into that directory, e.g.:

```
$> cd Strabon
```

4. Edit the `endpoint/WebContent/WEB-INF/connection.properties` file and define the PostgreSQL host, the database name and the credentials that will be used by for storing stRDF metadata (a temporal extension of RDF [8]).
5. Compile the endpoint by calling:

```
$> mvn clean package
```

The result of this action is the creation of a WAR file inside the target directory.

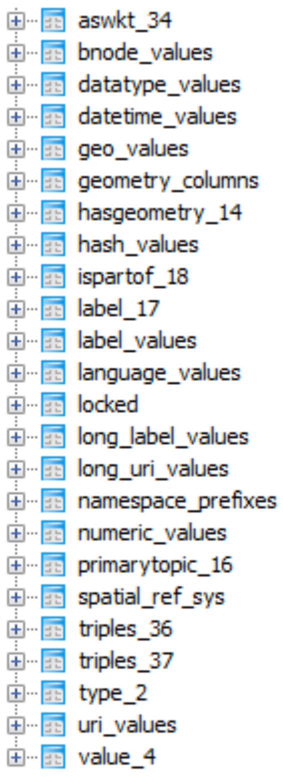
6. Deploy the endpoint by placing the WAR file that was created inside the `webapps` folder of the Apache Tomcat installation, e.g.:

```
$> sudo cp endpoint/target/strabon-endpoint-*.war
/var/lib/tomcat6/webapps/strabonendpoint.war
```

7. Start the Apache Tomcat server, e.g.:

```
$> sudo /etc/init.d/tomcat start
```

Strabon will auto create the tables shown in Figure 3.



```
aswkt_34
bnode_values
datatype_values
datetime_values
geo_values
geometry_columns
hasgeometry_14
hash_values
ispartof_18
label_17
label_values
language_values
locked
long_label_values
long_uri_values
namespace_prefixes
numeric_values
primarytopic_16
spatial_ref_sys
triples_36
triples_37
type_2
uri_values
value_4
```

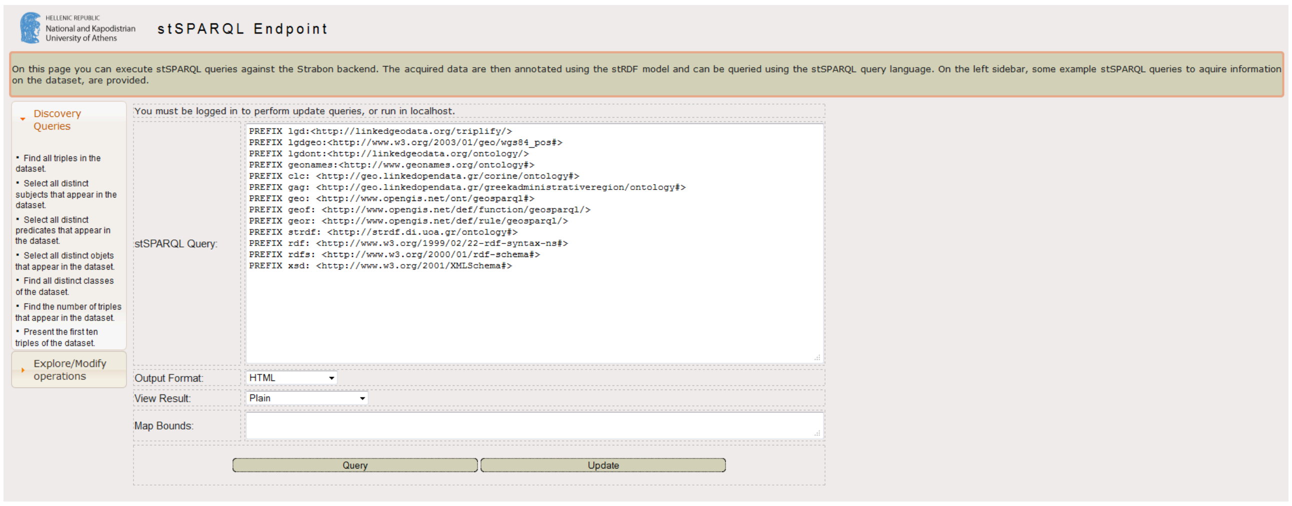
Figure 3 – Strabon Tables

Among others, the table `bnode_values` will store blank nodes references, the table `geo_values` will be used to store geometries and the table `label_values` will be used to store textual information.

The GeoSPARQL endpoint set up in the Liked Map project can be accessed through the URL:

```
http://descartes.cps.unizar.es:8080/strabonendpoint/
```

This URL shows a simple SPARQL client (Figure 4), included with the Strabon distribution, which makes easy the execution of query and update operations against the backend.



HELLENIC REPUBLIC
National and Kapodistrian
University of Athens

stSPARQL Endpoint

On this page you can execute stSPARQL queries against the Strabon backend. The acquired data are then annotated using the stRDF model and can be queried using the stSPARQL query language. On the left sidebar, some example stSPARQL queries to aquire information on the dataset, are provided.

Discovery Queries

- Find all triples in the dataset.
- Select all distinct subjects that appear in the dataset.
- Select all distinct predicates that appear in the dataset.
- Select all distinct objets that appear in the dataset.
- Find all distinct classes of the dataset.
- Find the number of triples that appear in the dataset.
- Present the first ten triples of the dataset.

Explore/Modify operations

You must be logged in to perform update queries, or run in localhost.

stSPARQL Query:

```
PREFIX lgd:<http://linkedgeodata.org/triplify/>
PREFIX lgdgeo:<http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX lgdont:<http://linkedgeodata.org/ontology/>
PREFIX geonames:<http://www.geonames.org/ontology#>
PREFIX clc: <http://geo.linkedopendata.gr/corine/ontology#>
PREFIX gag: <http://geo.linkedopendata.gr/greekadministrativeregion/ontology#>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX geof: <http://www.opengis.net/def/function/geosparql/>
PREFIX geor: <http://www.opengis.net/def/rule/geosparql/>
PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

Output Format: HTML

View Result: Plain

Map Bounds:

Figure 4 – Strabon SPARQL client

3 Integration with Linked Map Service

The deliverable D15.1 [9] collects all the requirements identified in the Linked Map project. Some of the requirements explicitly state the need for a SPARQL endpoint that supports SPARQL 1.1 Update and geographic resources. These requirements are fulfilled by the use of Strabon described in the previous section.

The deliverable D15.1 also describes the features of the LMS interface. The LMS interface should give access to RDF descriptions of resources following Linked Data best practices and should allow to edit, to update and to delete resources. Next we describe which is the role of the SPARQL endpoint provided by Strabon for solving this issue.

Figure 5 describes the interaction between a LMS instance and a Strabon endpoint. On behalf of the user, the LMS instance makes spatial queries to the Strabon endpoint for retrieving spatial objects that may be present in maps and RDF descriptions of such resources. There are two typical cases:

- When the user requests a machine-readable description of a map produced by a WMS GetMap request, the LSM server queries for RDF resources stored in Strabon that may appear in the spatial extent of the map (a bounding box) and belonging to the datasets shown in the map. For this task, the LSM server constructs a `SELECT` query that retrieves information about each matching resource.
- When the user requests a machine-readable description of a resource the LMS server makes a `DESCRIBE` query for such resource.

Since version 1.0, the LMS server caches the responses in a local triple store (Apache Jena TDB[5]) avoiding making redundant queries to Strabon.

Additionally, the user may request the LMS instance to create, to update or to delete some resources. These requests are stored locally in the Apache Jena TDB, and, if required, stored permanently in Strabon by means of SPARQL Updates. These modifications are performed as a batch operation periodically.

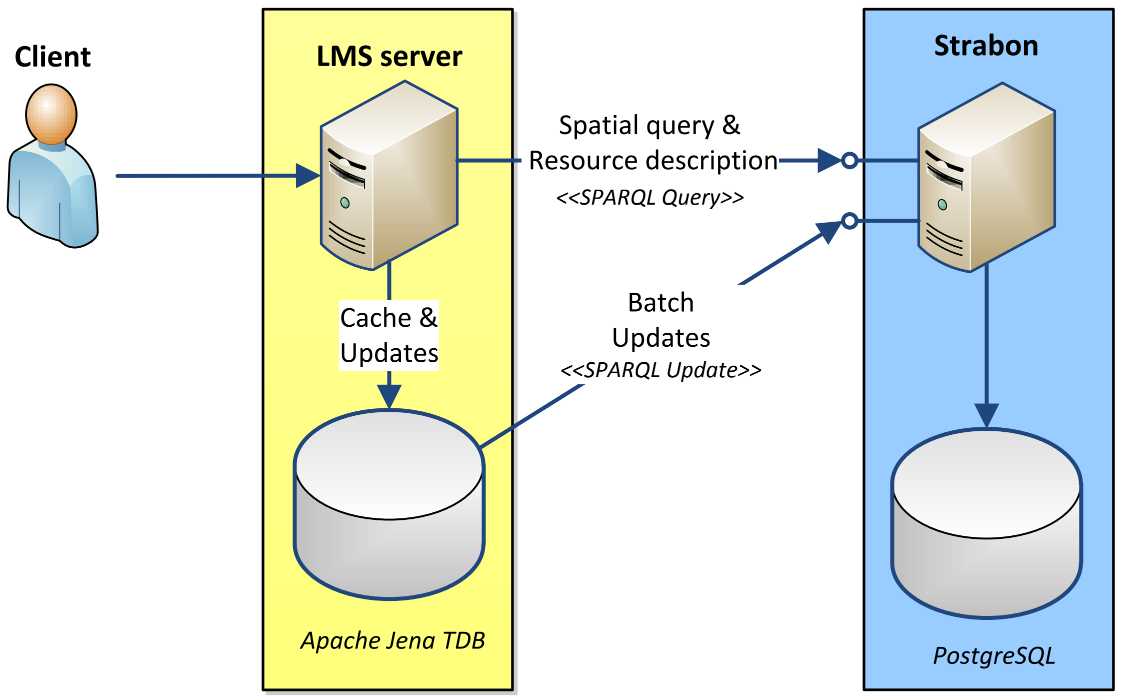


Figure 5 – Interaction between a LMS instance and the access/update service

[5] http://jena.apache.org/documentation/tdb/

4 Conclusions

In this document, we have described the setting up of a read-write endpoint supporting GeoSPARQL specification. This document is the last deliverable of Work Package 16, where the provisioning of data for the project has been fully covered, and acts as the bridge between the work done in this Work Package and the LMS developed in Work Package 17.

This goal has been achieved by means of the use of Strabon which supports the querying of data using the aforementioned standard. The document has described the steps followed in order to set up a GeoSPARQL endpoint using Strabon and how this endpoint fulfils the requirements established for the LMS regarding RDF data management and querying.

References

[1] M. Perry and J. R. Herring, Eds., “OGC GeoSPARQL - A Geographic Query Language for RDF Data,” OGC 11-052r4, 2012.

[2] L. Feigenbaum, E. Torres, and K. G. Clark, “SPARQL Protocol for RDF,” W3C, 2008.

[3] K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis, “Strabon: A Semantic Geospatial DBMS,” in *Lecture Notes in Computer Science*, vol. 7649, no. 19, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 295–311.

[4] J. Barrera and F. J. Lopez-Pellicer, “D17.1 Call 2: Linked Map Read-write Linked Data enabled OGC Web map server,” PlanetData, 2014.

[5] J. de la Beaujardiere, Ed., “OpenGIS® Web Map Server Implementation Specification,” Open Geospatial Consortium Inc., OGC 06-042, Mar. 2006.

[6] F. J. Lopez-Pellicer and J. Barrera, “D16.2 Call 2: Linked Map Provisioning service,” PlanetData, 2014.

[7] C. Strobl, “PostGIS,” *Encyclopedia of GIS*, pp. 891–898, 2008.

[8] K. Bereta, P. Smeros, and M. Koubarakis, “Representation and Querying of Valid Time of Triples in Linked Geospatial Data,” in *The Semantic Web: Semantics and Big Data*, vol. 7882, no. 18, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 259–274.

[9] F. J. Lopez-Pellicer and J. Barrera, “D15.1 *Call 2: Linked Map requirements definition and conceptual architecture*,” PlanetData, 2014.