HELLENIC REPUBLIC
National and Kapodistrian
University of Athens

Representation, Querying and Visualisation of Linked Geospatial Data

Konstantina Bereta and George Stamoulis

RoD Tutorial

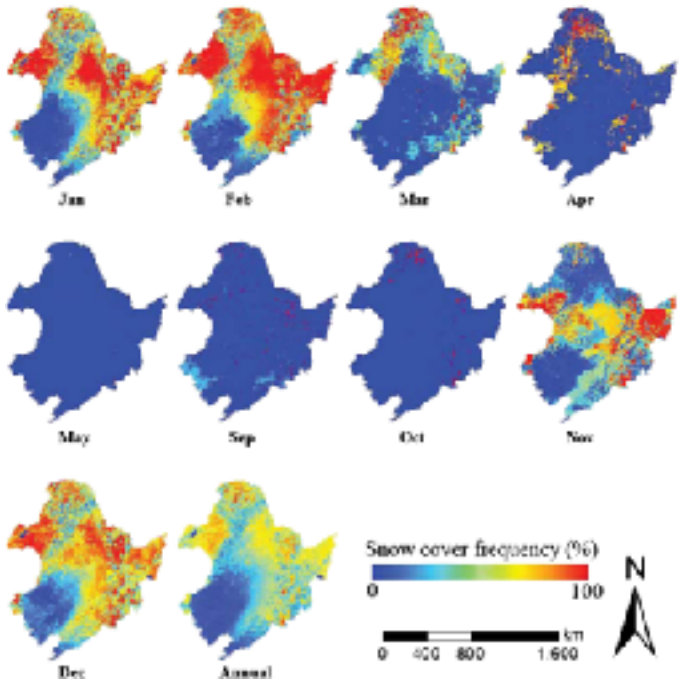
October 4, 2018

Outline

- Introduction
- Previous related research in other areas
- Motivation
- Extensions of RDF and SPARQL for the representation and querying of geospatial data
- Geospatial description logics and geospatial ontology-based data access systems
- Implemented systems, evaluation and comparison
- Visualisation of linked geospatial data
- Open issues

Why Spatial (and Temporal) Data?

- Spatial and temporal data are very important in reality: Everything that happens, happens sometime, somewhere.
- Decision making can be substantially improved if we know when and where things take place.
- This tutorial is for the “where”.



Previous Research

Geographic Information Systems (GIS) Research

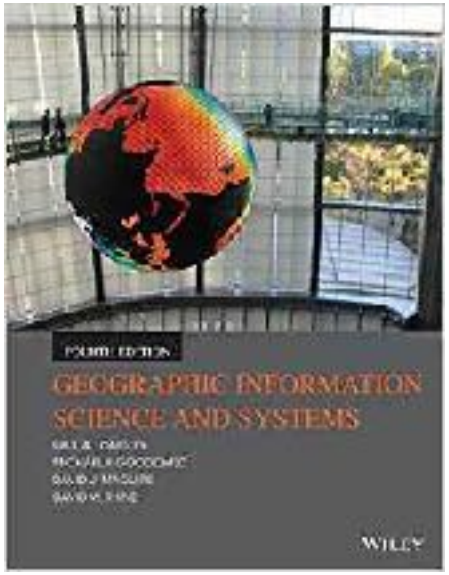
- Lots of interesting theoretical and practical work by GIS researchers.
- Topics covered:
  - Geographic data and their representation
  - Geographic data modelling and geographic databases
  - GIS software
  - Cartography and map production
  - Spatial data analysis and decision making
  - Geospatial data on the Web

Geographic Information Systems Research (cont'd)

- Industrial impact:
  - Lots of relevant standards by the Open Geospatial Constortium (OGC).
  - State-of-the-art GIS software e.g., ArcGIS or QGIS.

Geographic Information Systems Research (cont'd)

- Educational impact:
  - Relevant ideas have found their way in well-known GIS books.

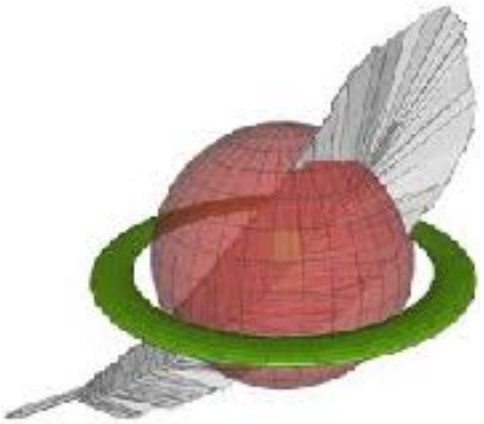


Spatial Database Research

- Lots of interesting theoretical and practical work by database researchers.
- Topics covered:
  - Data models and query languages
  - Storage structures and indexing techniques
  - Query processing
  - User interfaces
  - Implemented systems
  - Applications

Spatial Database Research (cont'd)

- Industrial impact:
  - The OGC standard "OpenGIS Simple Feature Access - Part 2: SQL option" introduced geospatial data in SQL in 2010.
  - Spatially enabled databases (e.g., PostGIS, Oracle Spatial, Spatialite).



Spatial AI Research

- Lots of interesting theoretical and practical work by Artificial Intelligence researchers.
- Topics covered:
  - Spatial logics
  - Spatial constraint networks
  - Implemented systems
  - Applications
  - Contributions to other AI research areas like planning and commonsense reasoning

Motivation

Geospatial Data on the Web

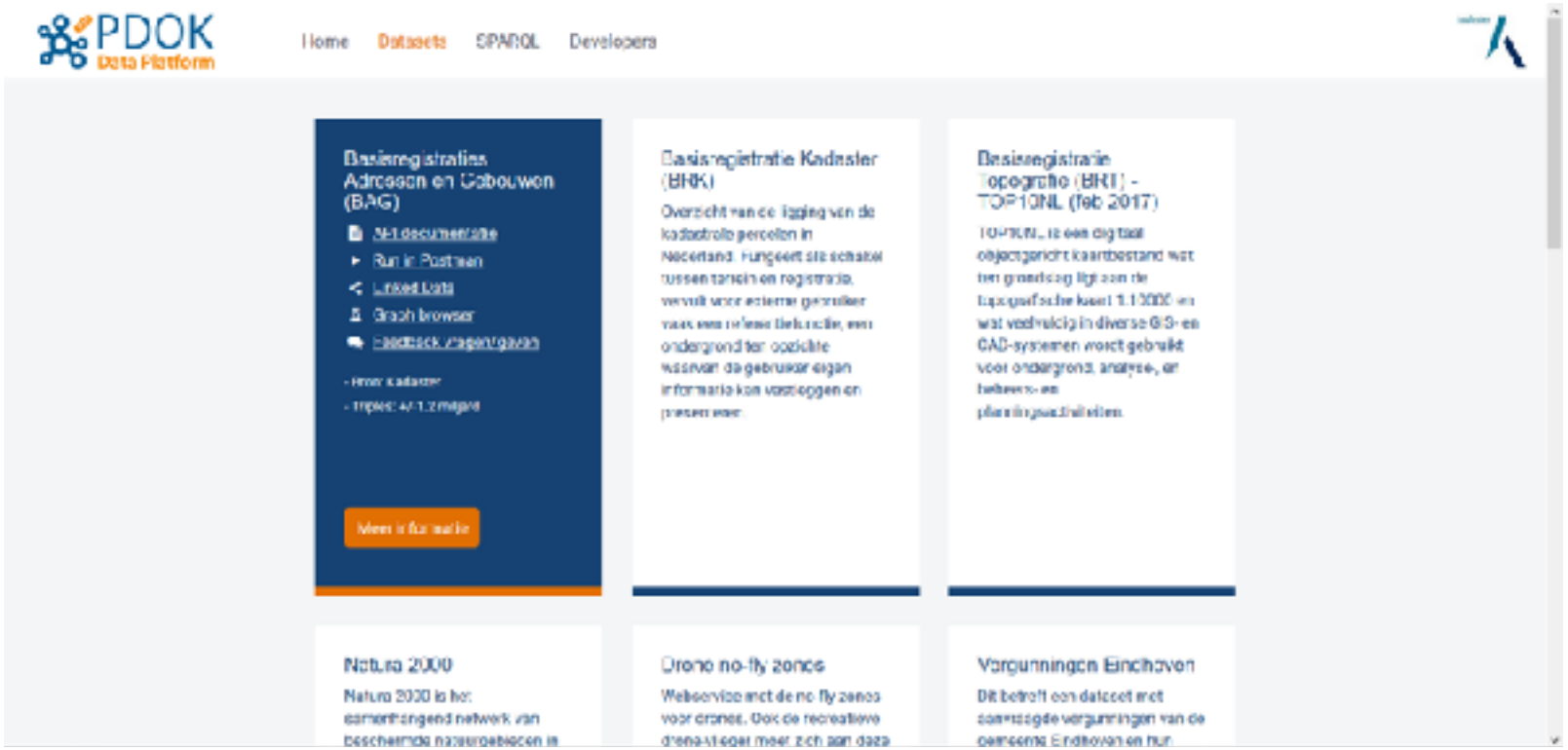
- Very popular and useful map software.

bing™ maps

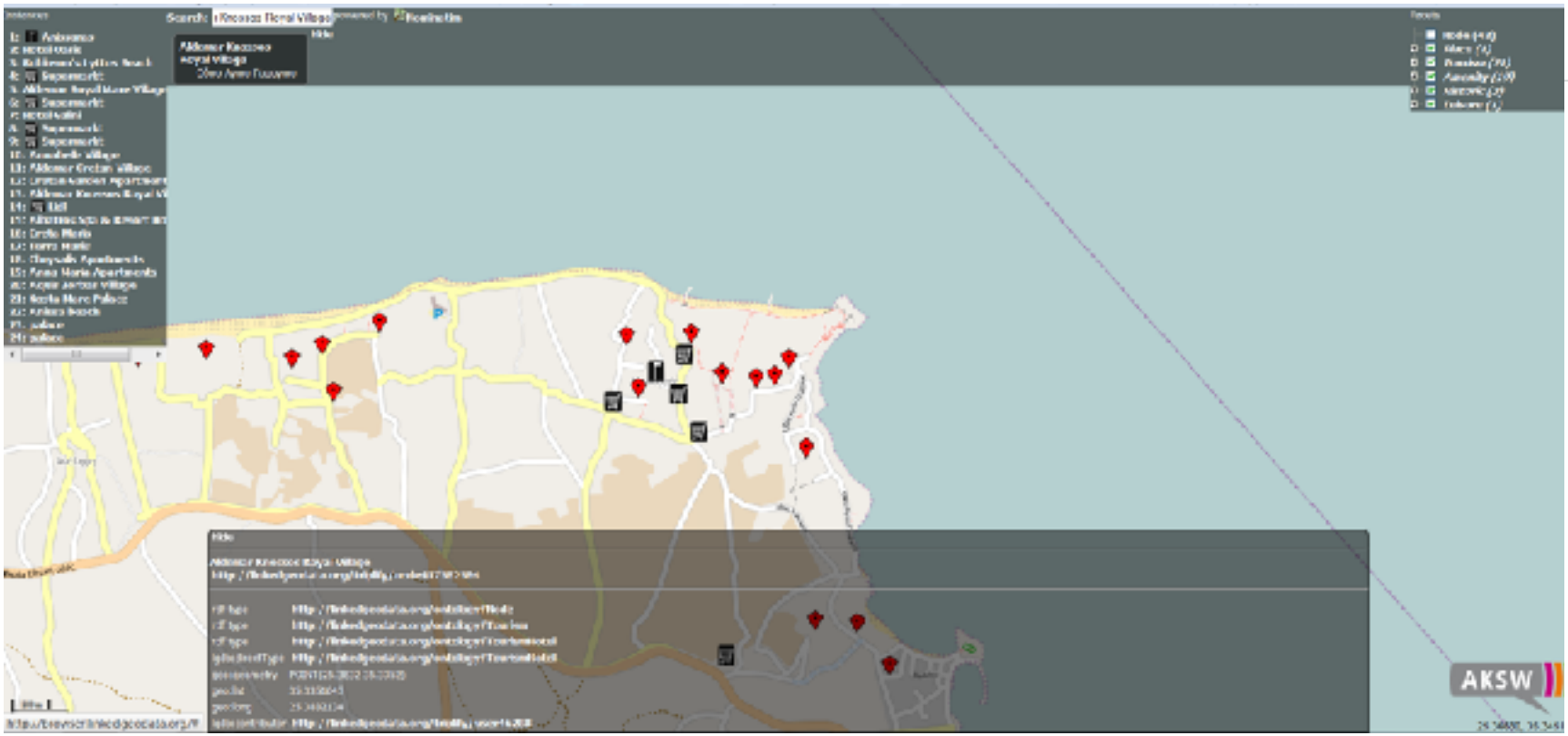
Open Government Data

Linked Geospatial Data- Ordnance Survey (United Kingdom)

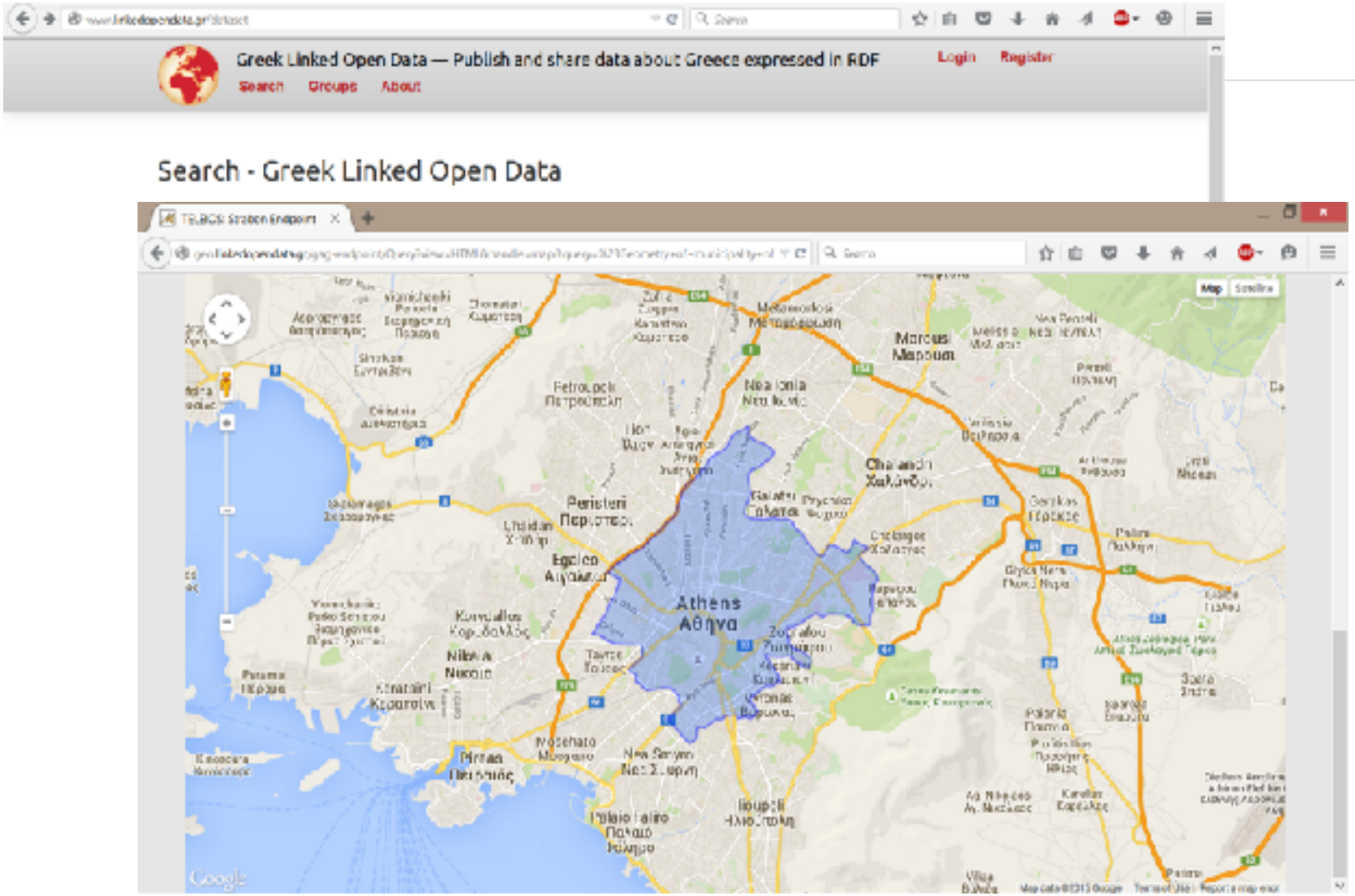
Linked Geospatial Data- Kadaster (The Netherlands)



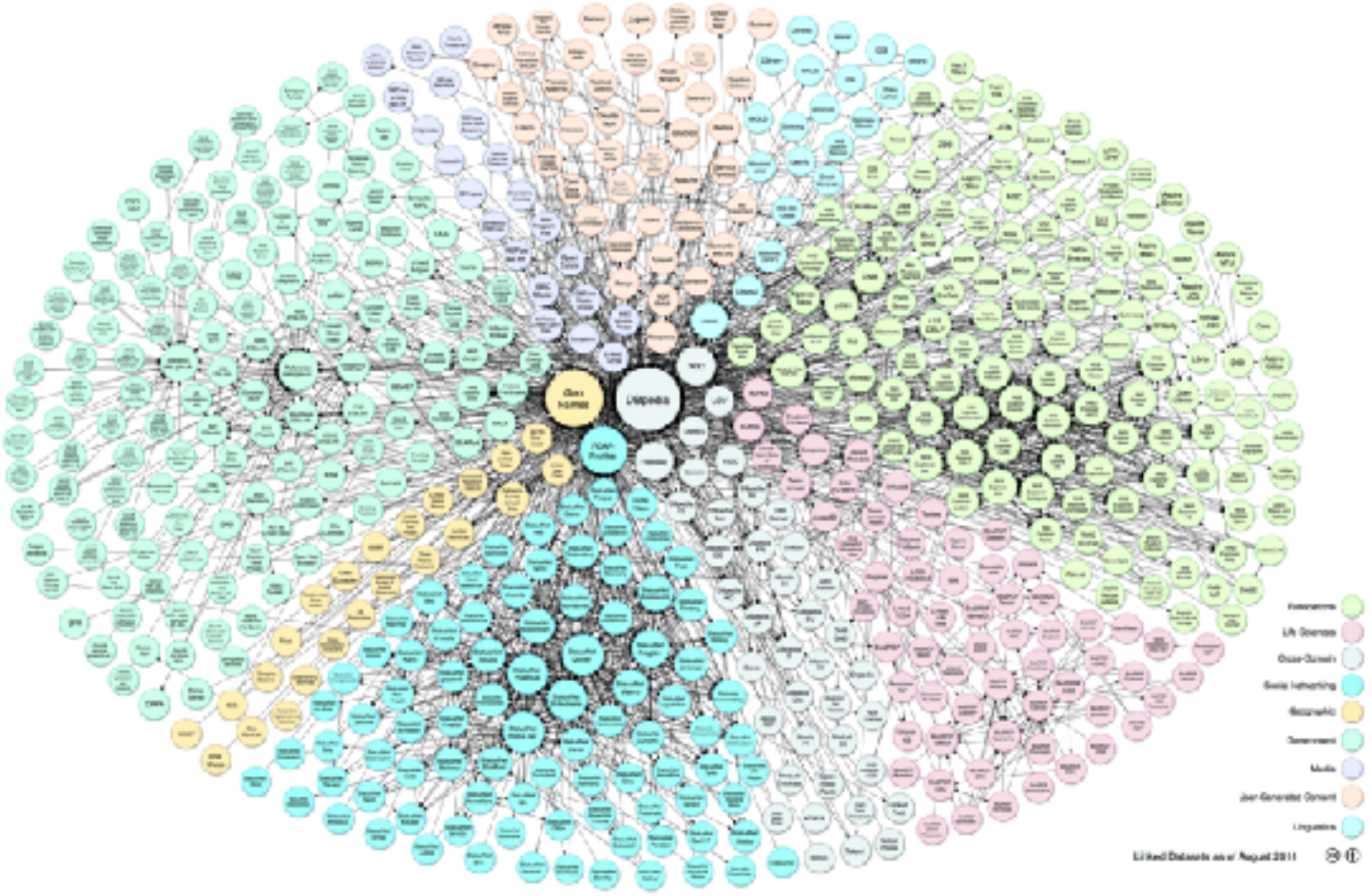
OpenStreetMap



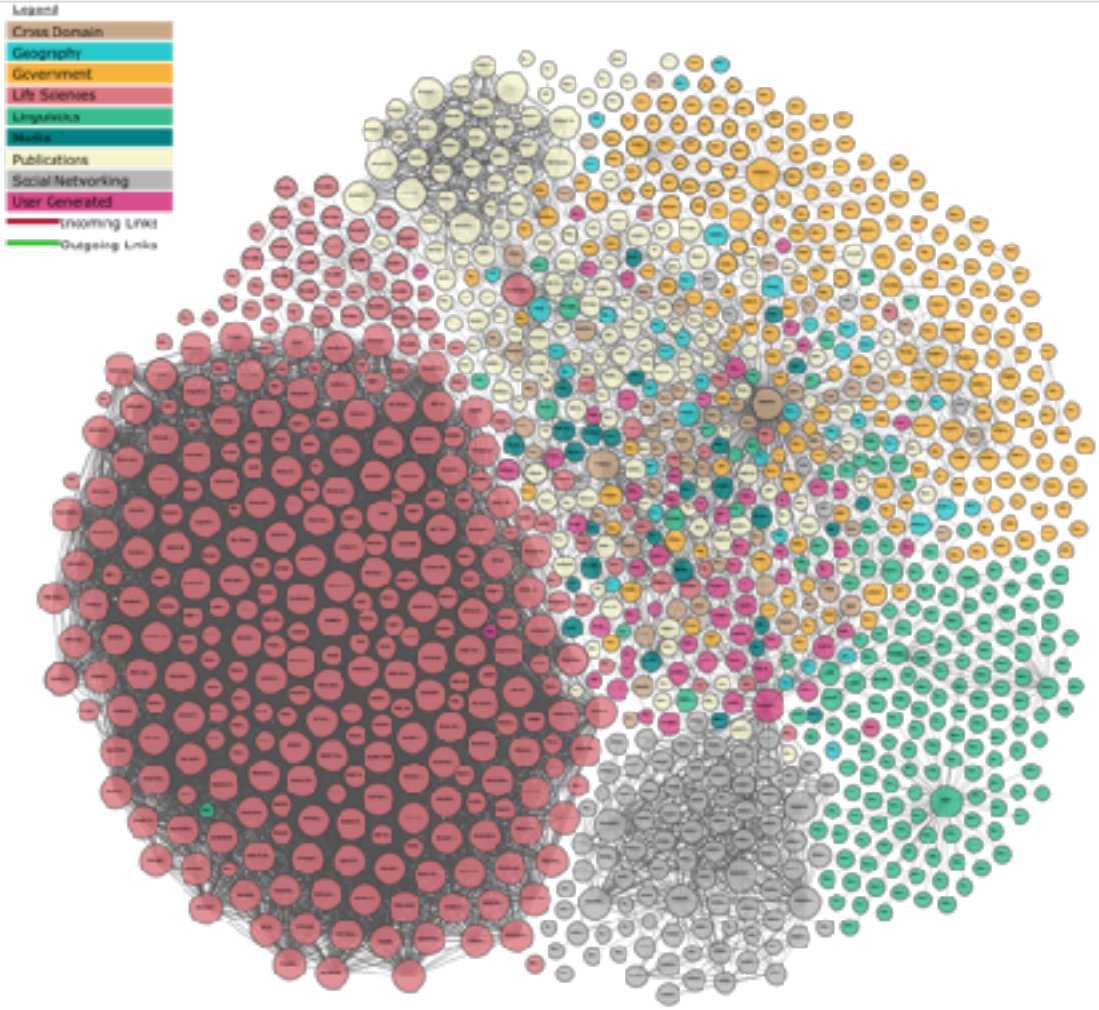
http://www.linkedopendata.gr



LOD Cloud (Aug. 2014): Lots of Geospatial Data



LOD Cloud (Aug. 2017): Lots of Geography Data



Geospatial Extensions of RDF and SPARQL

Overview

- Early papers:
  - Kolas (2007)
  - Perry's PhD dissertation (2008)
  - Koubarakis and Kyzirakos (2010)
- More recent proposals:
  - The OGC standard GeoSPARQL (2012)
  - The data model stRDF/stSPARQL (2012)
  - The framework RDF^{i} (2013)

GeoSPARQL and stRDF/stSPARQL

- The two proposals offer constructs for:
  - Developing **ontologies** for spatial and temporal data
  - Encoding **spatial and temporal data** that use these ontologies in **RDF**
  - **Extending SPARQL** to query spatial and temporal data
- **Temporal data** is covered only by stRDF/stSPARQL

GeoSPARQL

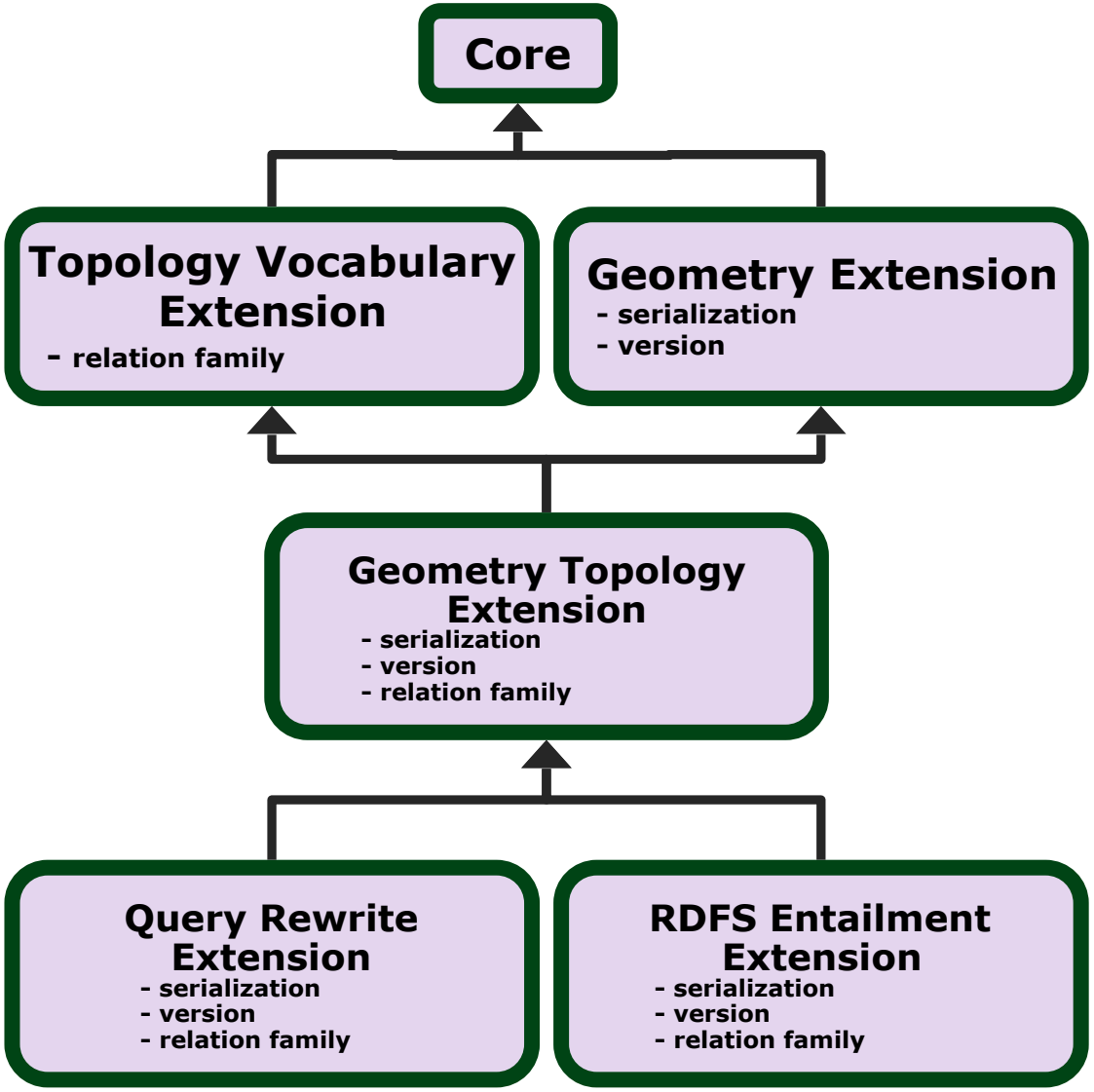
[Perry and Herring, 2012]

GeoSPARQL is an **OGC standard.**

Main functionalities:

- Representing geospatial information is done using **high level ontologies** inspired from GIS terminology
- Geometries are represented using **literals** of **spatial datatypes**
- Literals are serialized using OGC standards **WKT** and **GML**
- Families of **functions** are offered for querying geometries

GeoSPARQL Components

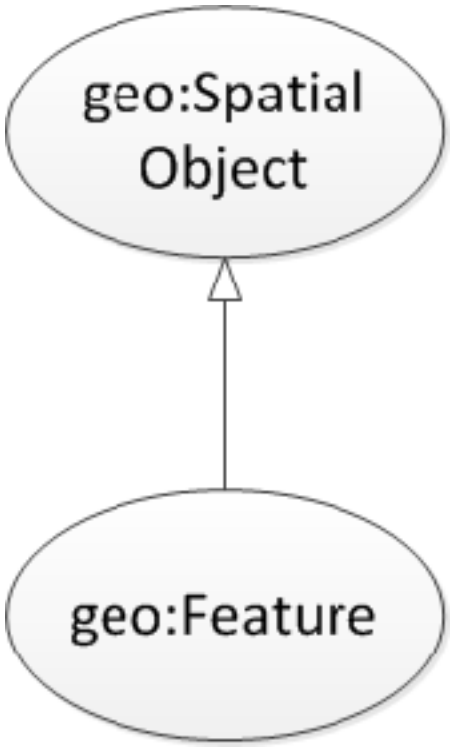


Parameters

- **Serialization**
  - WKT
  - GML
- **Relation Family**
  - Simple Features
  - RCC-8
  - Egenhofer

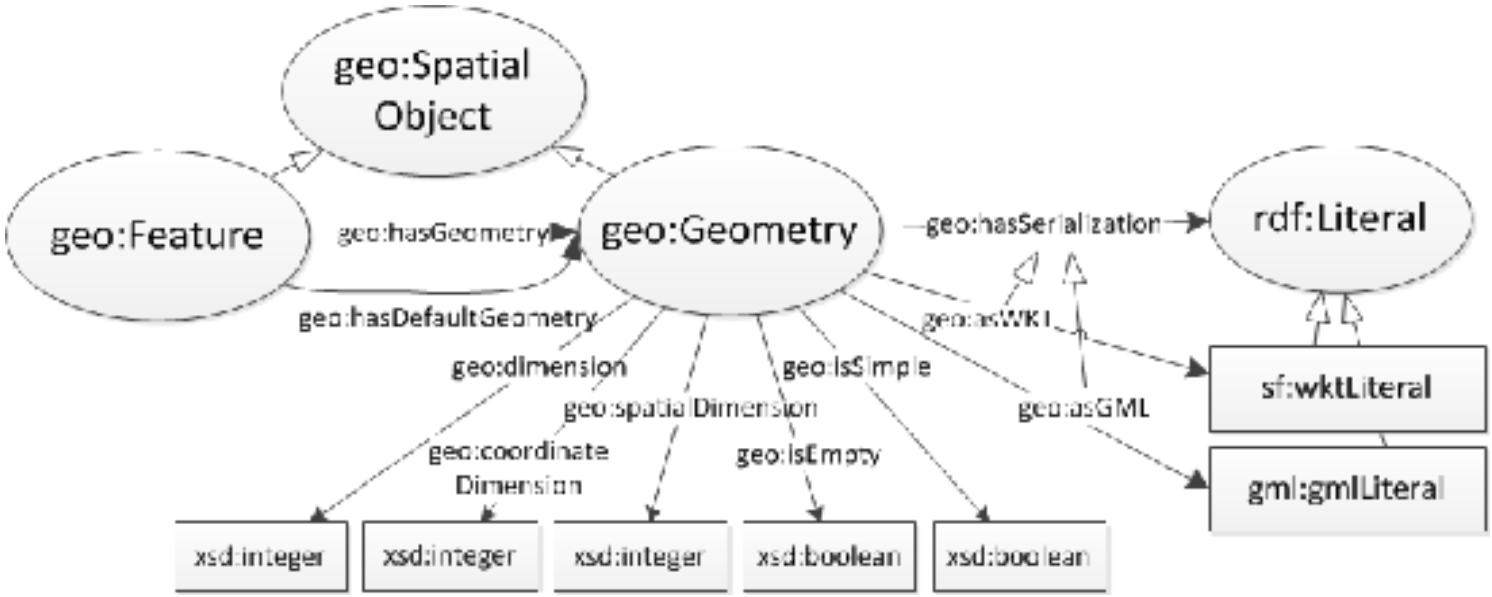
GeoSPARQL Core

- Defines two **top level classes** that can be used to organize geospatial data



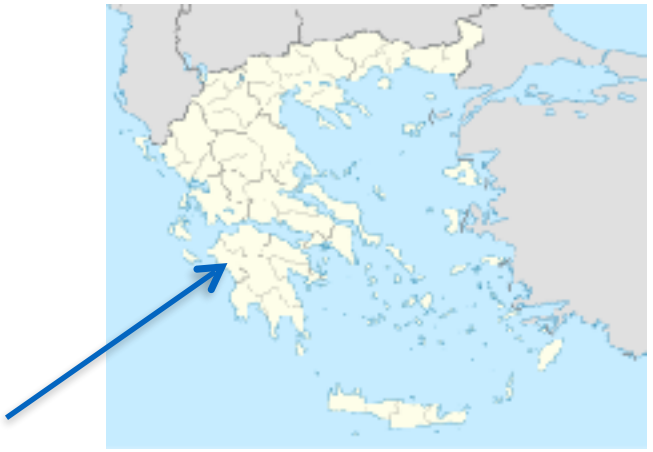
GeoSPARQL Geometry Extension

- Provides vocabulary for asserting and querying data about the **geometric attributes** of a feature



Example

```
gag:Olympia
    rdf:type gag:MunicipalCommunity;
    gag:name "Ancient Olympia";
    gag:population "184"^^xsd:int;
    geo:hasGeometry ex:polygon1.
```



Ancient Olympia

```
ex:polygon1
    rdf:type geo:Geometry;
    geo:asWKT "http://www.opengis.net/def/crs/OGC/1.3/CRS84
              POLYGON((21.5 18.5,23.5 18.5,
                       23.5 21,21.5 21,21.5 18.5))"
                ^^sf:wktLiteral.
```

GeoSPARQL Geometry Extension (cont'd)

The following **non-topological query functions** from the "OpenGIS Simple Feature Access" standard are also offered:

- `geof:distance`
- `geof:buffer`
- `geof:convexHull`
- `geof:intersection`
- `geof:union`
- `geof:difference`
- `geof:symDifference`
- `geof:envelope`
- `geof:boundary`

Example Query

Find forests near municipal communities.

```
SELECT ?r ?c
WHERE {
       ?r rdf:type clc:Region;
          geo:hasGeometry ?rGeom;
          clc:hasCorineLandCover ?f.
       ?f rdfs:subClassOf clc:Forest.
       ?c rdf:type gag:MunicipalCommunity;
          geo:hasGeometry ?cGeom.

FILTER(geof:distance(?rGeom,?cGeom,uom:metre) < 1000)}
```

GeoSPARQL Geometry Topology Extension (cont'd)

The following **topological query functions** from the "OpenGIS Simple Feature Access" standard are offered:

- `geof:sfEquals`
- `geof:sfDisjoint`
- `geof:sfIntersects`
- `geof:sfTouches`
- `geof:sfCrosses`
- `geof:sfWithin`
- `geof:sfContains`
- `geof:sfOverlaps`

Example Query

Find forests that border municipal communities.

```
SELECT ?r ?c
WHERE {
       ?r rdf:type clc:Region;
          geo:hasGeometry ?rGeom;
          clc:hasCorineLandCover ?f.
       ?f rdfs:subClassOf clc:Forest.
       ?c rdf:type gag:MunicipalCommunity;
          geo:hasGeometry ?cGeom.

FILTER(geof:sfTouches(?rGeom,?cGeom))}
```

GeoSPARQL Geometry Topology Extension (cont'd)

- The previous family of functions are based on the **DE-9IM** model studied by Clementini and Felice.

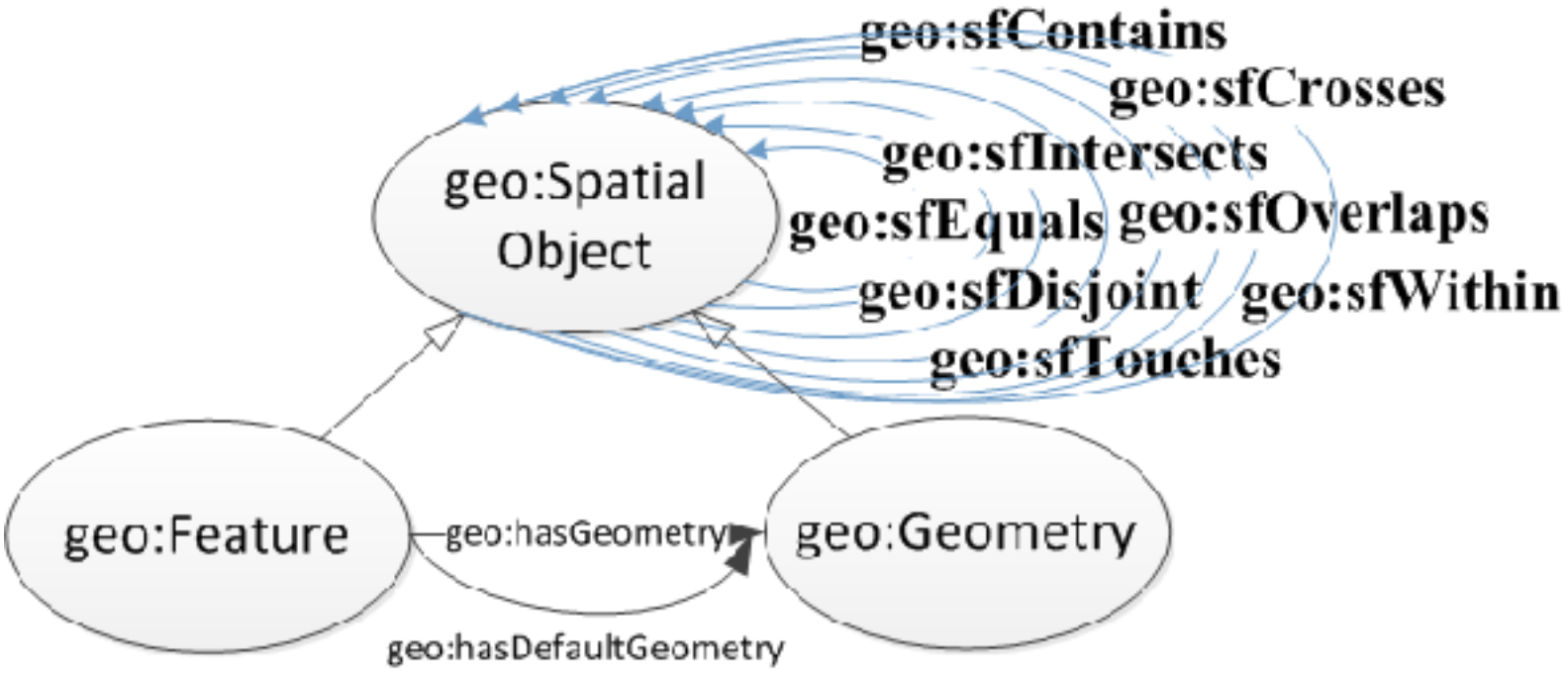
- Similarly, the family of functions in the **Egenhofer** and **RCC-8** frameworks are offered.

GeoSPARQL Topology Vocabulary Extension

- This extension is used for representing **topological information** about features.
- Topological information is **inherently qualitative** and it is expressed in terms of **topological relations** (e.g., containment, adjacency, overlap etc.).
- Topological information can be **derived from geometric information** or it might be captured by **asserting explicitly the topological relations** between features.

Topological Relations

- The topological relations of the "OpenGIS Simple Feature Access" standard are offered.



Topological Relations (cont'd)

- Similarly, the topological relations of the **Egenhofer** and the **RCC-8** framework.

- GeoSPARQL offers us **vocabulary for expressing these topological relations in the database and the queries.**

Example

```
gag:Olympia rdf:type gag:MunicipalCommunity.

gag:OlympiaMunicipality rdf:type gag:Municipality.

gag:WesternGreece rdf:type gag:Region.


gag:Olympia geo:sfWithin gag:OlympiaMunicipality.
gag:OlympiaMunicipality geo:sfWithin gag:WesternGreece.
```

Query I

Find the **municipality** that contains the community of Ancient Olympia.

```
SELECT    ?m
WHERE {
  ?m rdf:type gag:Municipality.
  ?m geo:sfContains gag:Olympia.
}
```

Answer

```
gag:OlympiaMunicipality
```

Query II

Find the **region of Greece** that contains the community of Ancient Olympia.

```
SELECT    ?m
WHERE {
  ?m rdf:type gag:Region.
  ?m geo:sfContains gag:Olympia.
}
```

Answer

```
gag:WesternGreece
```

Query II (cont'd)

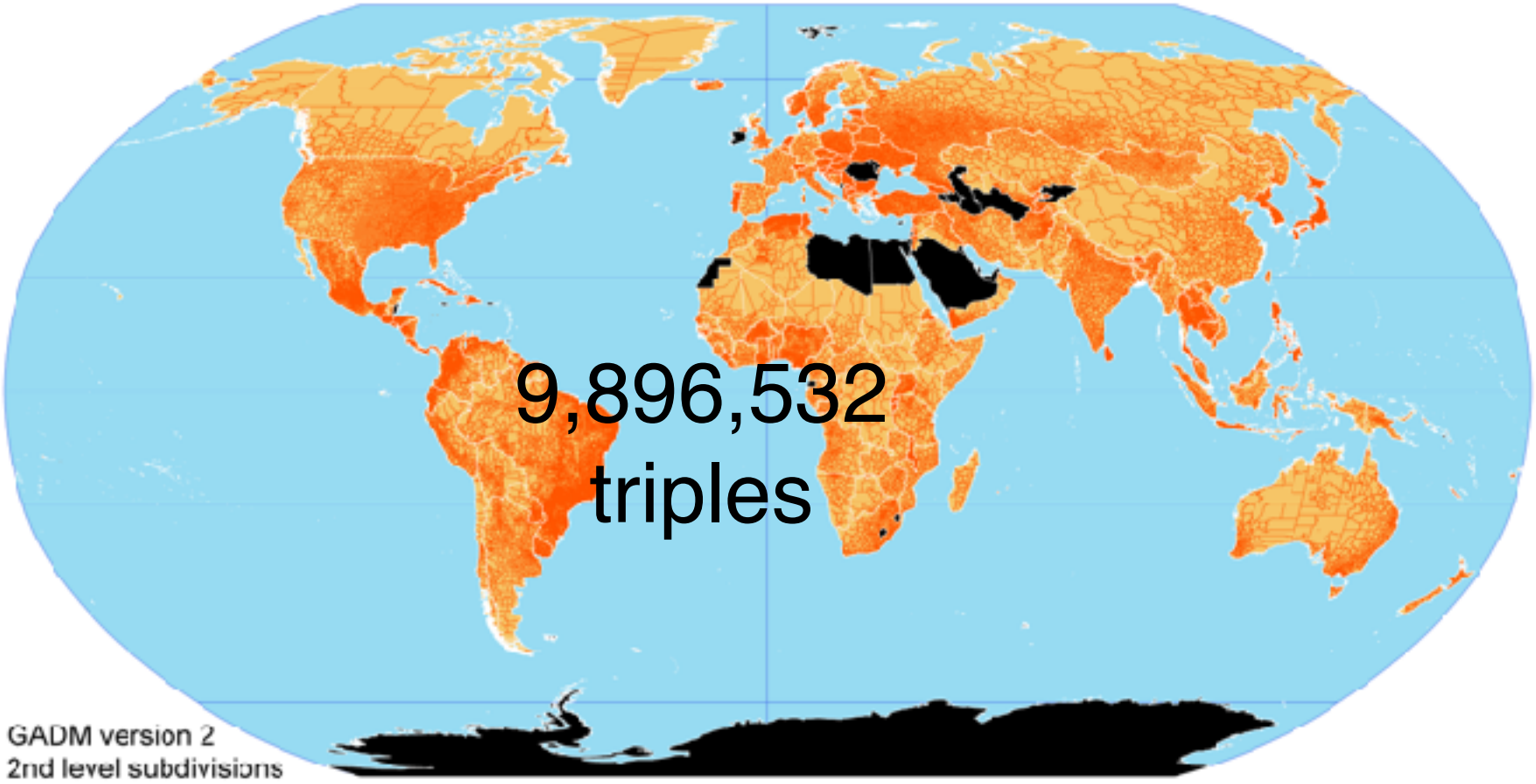
- The answer to Query II can be computed by **reasoning about the transitivity** of relation `geo:sfContains`.

- The GeoSPARQL standard **does not cover such entailed topological relations** between spatial objects. Is this a problem?

Administrative Geography of Great Britain

73,546,231 triples

Global Administrative Areas (GADM)



Nomeclature of Territorial Units for Statistics (NUTS)

Home

LV00 - Latvija

This file is a human-readable representation of both the feature and its geometry.

View the feature as: Turtle, RDF/XML

View the geometry as Turtle, RDF/XML, GML, KML

Feature

rdf:type	ramon:NUTSRegion, spatial:Feature
rdfs:label	LV00 - Latvija
ramon:name	Latvija
ramon:level	2
ramon:code	LV00
ngeo:geometry	nuts:LV00_geometry
spatial:P	LV0
spatial:P	http://dbpedia.org/resource/Daugavpils
spatial:EC	http://nuts.geovocab.org/id/EE
spatial:EC	http://nuts.geovocab.org/id/LT
spatial:EC	http://nuts.geovocab.org/id/EE0
spatial:EC	http://nuts.geovocab.org/id/LT0
spatial:EC	http://nuts.geovocab.org/id/EE00
spatial:EC	http://nuts.geovocab.org/id/LT00
spatial:EC	http://nuts.geovocab.org/id/EE004
spatial:EC	http://nuts.geovocab.org/id/LT005
spatial:EC	http://nuts.geovocab.org/id/LT006
spatial:EC	http://nuts.geovocab.org/id/LT003
spatial:EC	http://nuts.geovocab.org/id/LT008
spatial:EC	http://nuts.geovocab.org/id/LT009
spatial:EC	http://nuts.geovocab.org/id/EE008
owl:sameAs	http://esdtwrap.ontologycentral.com/dic/geo#LV0C
owl:sameAs	http://rdfdata.eurostat.europa.eu/ramon/nuts2008/LV00
owl:sameAs	http://ec.europa.eu/eurostat/ramon/rdfdata/nuts2008/LV00
owl:sameAs	http://nuts.psi.enakting.org/id/LV00

Geometry

rdf:type	ngeo:Polygon

316,246 triples

Pending Extension in schema.org

- Topological relations between places in schema.org

Knowledge Graphs with Geospatial Information

- Current extension of Yago2 with geospatial information including topological relations between geo-entities.

The Query Rewrite Extension

- Enables the **translation of qualitative topological information appearing in a query to quantitative.**
- This is done by rewriting of queries with triple patterns involving topological relations into queries with topological functions on geometries.
- The rewriting is based on a set of **RIF rules** defined in the standard.

The RDFS Extension

- Enables standard RDFS reasoning for GeoSPARQL classes and properties.

The Data Model stRDF

[Kyzirakos, Karpathiotakis & Koubarakis ISWC 2012]

- An extension of RDF for the representation of **geospatial information that changes over time**.
- **Geospatial dimension**:
  - **Spatial data types** are introduced.
  - Geospatial information is represented using **spatial literals** of these datatypes.
  - **OGC standards WKT and GML** are used for the serialization of spatial literals.
- **Temporal dimension**
- Proposed independently and around the same time as GeoSPARQL (starting with an ESWC 2010 paper by Koubarakis and Kyzirakos).

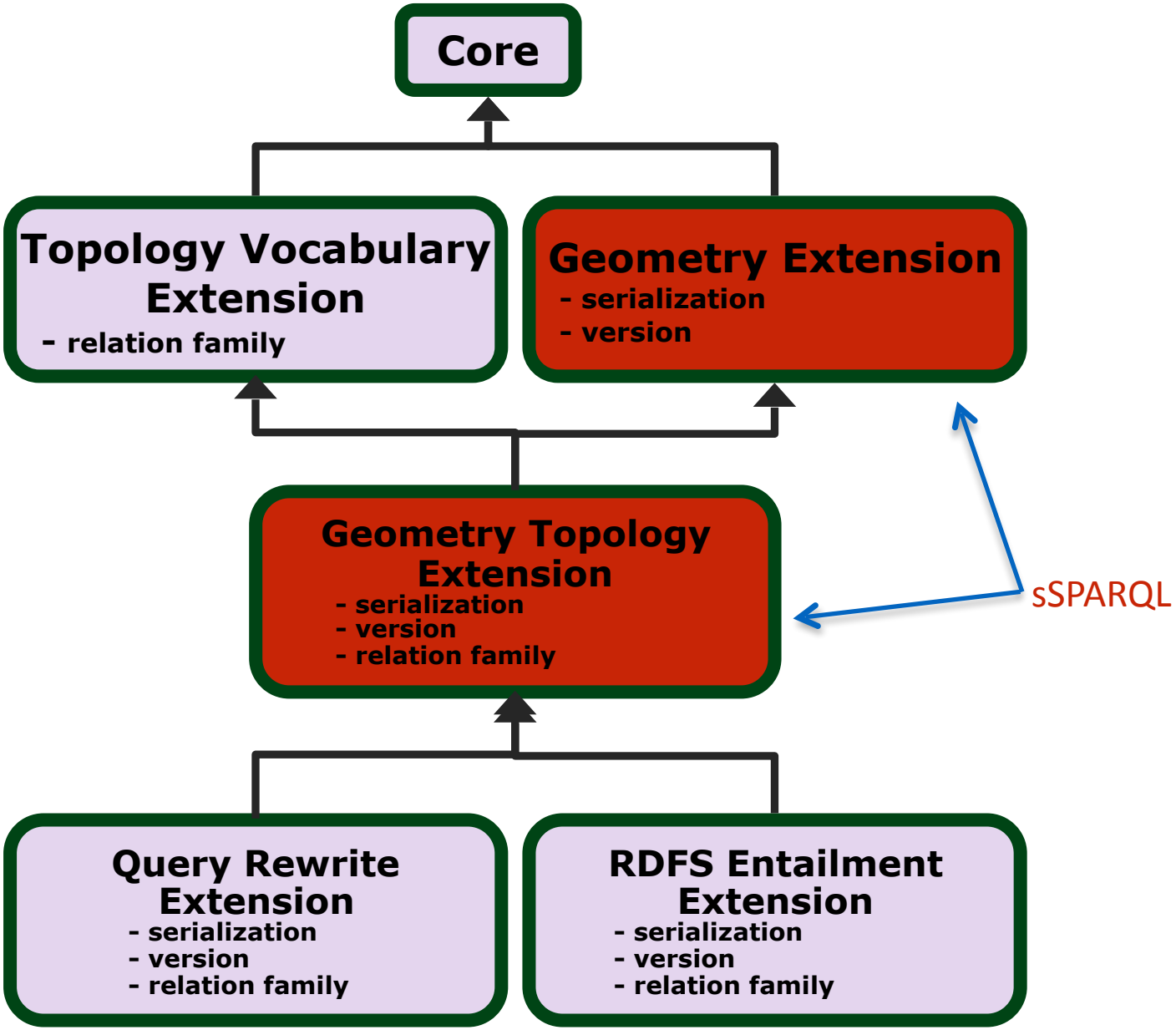
The Query Language stSPARQL

- It is an extension of SPARQL 1.1
- It offers families of functions for querying geometries. The same functions as in the Geometry Extension and Geometry Topology Extension of GeoSPARQL.
- In addition the following **spatial aggregate functions** are offered:

```
strdf:geometry strdf:union(set of strdf:geometry A)
strdf:geometry strdf:intersection(set of strdf:geometry A)
strdf:geometry strdf:extent(set of strdf:geometry A)
```

- **Temporal dimension** (not covered in this tutorial. See the ESWC 2013 by Bereta, Smeros and Koubarakis).

GeoSPARQL vs. stSPARQL



Example of stSPARQL

Compute the parts of burnt areas that lie in coniferous forests.

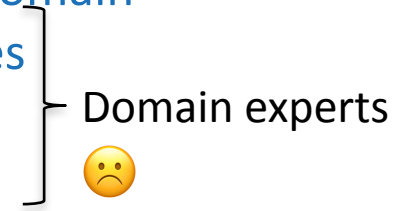
```
SELECT ?burntArea (strdf:intersection(?baGeom,
                   strdf:union(?fGeom)) AS ?burntForest)
WHERE {  ?burntArea  rdf:type noa:BurntArea;
                     strdf:hasGeometry ?baGeom.
         ?forest rdf:type clc:Region;
                 clc:hasLandCover clc:ConiferousForest;
                 strdf:hasGeometry ?fGeom.

FILTER  (strdf:intersects(?baGeom,?fGeom)) }
GROUP BY ?burntArea ?baGeom
```

Geospatial description logics and geospatial ontology-based data access

Motivation

- Publishing data as RDF and correlating them with linked geospatial data is useful
- Challenges:
  - Domain experts (earth scientists, geologists, etc.) and domain specific applications heavily rely on geospatial databases
  - Original data are stored in them
  - Conversion to RDF is not always practical
    - Frequent updates
    - Large datasets
    - Different tools need to be used



Representing Geospatial Information in Description Logics

- Use OWL-DL [Katz et al. 2005]
- Define a spatial concrete-domain DL
  - $\mathcal{ALC}(\mathrm{RCC8})$ [Lutz and Milisic, 2007]
  - $\text{DL-Lite}^{\Box}_{\mathcal{F},\mathcal{R}}(\mathrm{RCC8})$ Özçep and Möller, 2012]

- Use OWL and a spatial ABOX
  - RacerPro [Wessel-Möller, 2009]
  - PelletSpatial [Stocker and Sirin, 2009]
  - [Grütter et al., 2008]
- Keyword queries over spatial OBDA sources [Eiter et al. 2013]
- Spatial ontology-mediated query answering over mobility streams [Eiter et al. 2017]

Geospatial Ontology-based Data Access

- Geospatial RDB2RDF systems: GeoTriples, TriplesGeo
- Mapping languages: R2RML (W3C standard), OBDA
- OBDA systems:
  - Ontop [Rodriguez-Muro et al., JWS'15]
  - Ultrawrap [Sequeda et al., JWS'13]
- Geospatial OBDA systems:
  - Ontop-spatial [Bereta & Koubarakis, ISWC'16]
  - Oracle Spatial and Graph 12c release 2

OBDA Mappings

[MappingDeclaration] [[mappingId gag_geometry

target gag:geometry/{gid}/ gag:asWKT {geo}^^geo:wktLiteral .

source select distinct gid,geom from gag

mappingId clc_geometry

Target clc:/{gid}/ clc:hasGeometry clc:/geometry/{id}/ . clc:/{gid}/ clc:asWKT {geom} ^^geo:wktLiteral .

source select distinct gid, geom from clc

mappingId clc_id

target clc:/{gid}/ clc:hasID {gid} . clc:/{gid}/ clc:hasLandUse {code_00} .

source select distinct gid, geom, code_00 from clc

mappingId clc_type

target clc:/{gid}/ clc:type clc:type . clc:/{gid}/ rdf:type clc:Area .

source select distinct gid, geom from clc]]

R2RML example

```
[ a                rr:TriplesMap ;
rr:logicalTable      [ a          rr:R2RMLView ;
                              rr:sqlQuery  "select distinct gid,geom from gag"                  ] ;
rr:predicateObjectMap  [ a          rr:PredicateObjectMap ;
                               rr:objectMap  [ a          rr:ObjectMap , rr:TermMap ;
                                                    rr:column    "geo" ;
                                                    rr:termType  rr:Literal                          ] ;
                               rr:predicate  clc:asWKT                    ] ;
                               rr:subjectMap        [ a          rr:TermMap , rr:SubjectMap ;
                                          rr:template  gag:{gid} ;  rr:termType  rr:IRI ]] .
```

Virtual Triples

clc:20440 rdf:type geo:Geometry;

geo:asWKT “POLYGON(…)”^^geo:wktLiteral .

Clc:20512 rdf:type geo:Geometry;

geo:asWKT “POLYGON(…)”^^geo:wktLiteral .

…

	gid integer	code_00 character varying(100)	id character varying(18)	remark character varying(20)	area_ha numeric	shape_leng numeric	shape_area numeric	geom geometry
1	20440	BroadLeavedForest	EU-1900387		9169698	72.0513238	5691.69698	0103000002
2	20512	BroadLeavedForest	EU-1900769		3331793	35.0822057	9033.31793	0103000002
3	20543	BroadLeavedForest	EU-1900881		9247076	6.17830328	189.247076	0103000002
4	20797	BroadLeavedForest	EU-1901587		7822436	3.55811923	107.822436	0103000002
5	20804	BroadLeavedForest	EU-1901816		1899830	97.0454395	8618.99830	0103000002

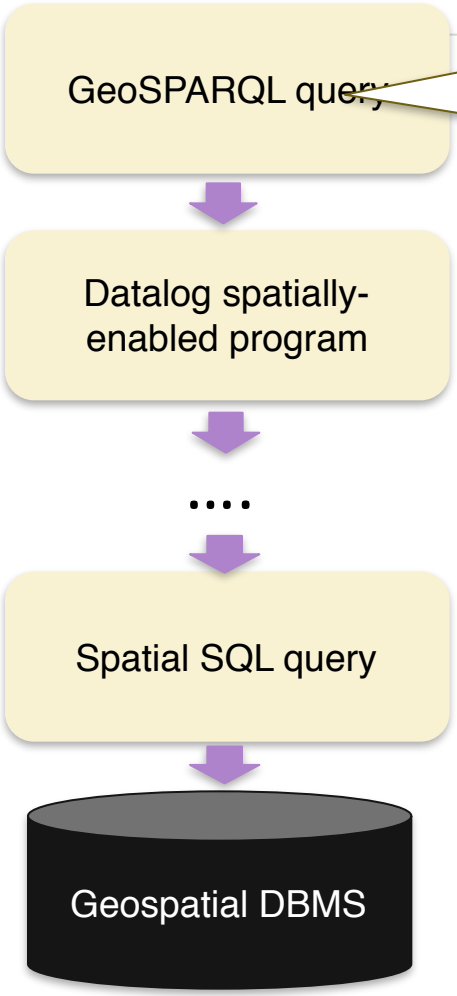
Example GeoSPARQL query

Select CORINE areas, their land use and the administrative division they belong to.

```
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX gag: <http://geo.linkedopendata.gr/gag/ontology/>
PREFIX clc: <http://geo.linkedopendata.gr/corine/ontology#>

SELECT  DISTINCT ?x1 ?x2 ?lu
 WHERE {
?x1 geo:asWKT ?g1 .
 ?x2 geo:asWKT ?g2 .
 ?x2 clc:hasLandUse ?lu .
FILTER (geof:sfIntersects(?g1,?g2))
}
```

GeoSPARQL-to-SQL translation

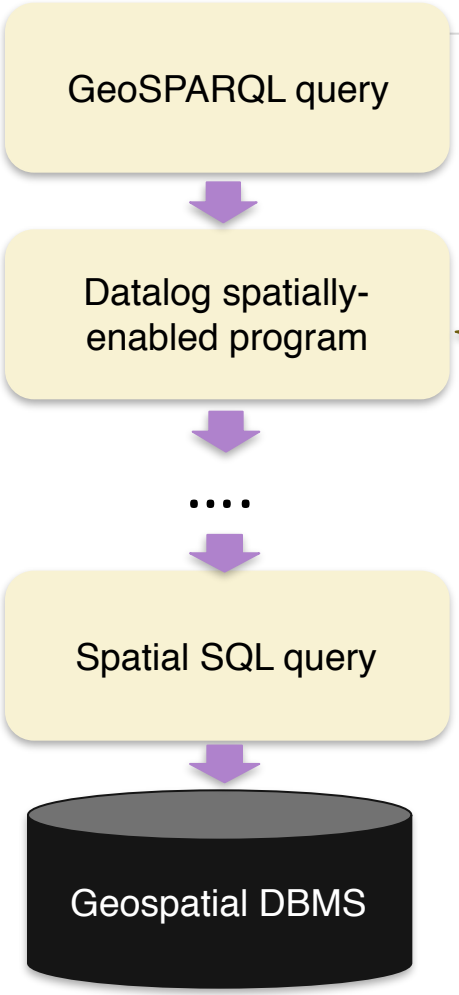



```
SELECT ?x1 ?x2
 WHERE{
    ?x1 geo:asWKT ?g1 .
     ?x2 geo:asWKT ?g2 .
     FILTER (geof:sfOverlaps(?g1,?g2)) }
```

OR

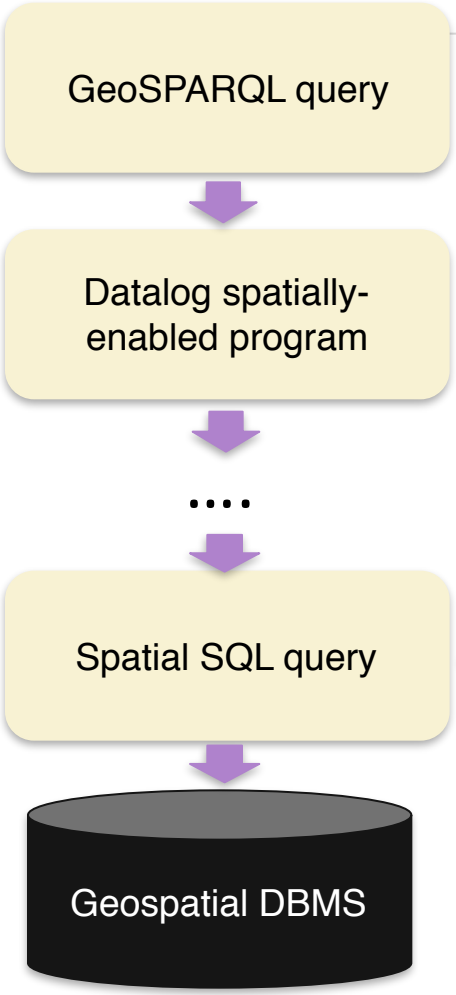
```
SELECT ?x1 ?x2
WHERE{
   ?x1 geo:sfOverlaps ?
x2 }
```

GeoSPARQL-to-SQL translation



ans1(x1,g1) :- ans2(g1,g2,x1,x2)
ans2(g1,g2,x1,x2) :- ans4(g1,g2,x1,x2), **SF-OVERLAPS**(g1,g2)
ans4(g1,g2,x1,x2) :- ans8(g1,x1), ans9(g2,x2)
ans8(g1,x1) :- http://www.opengis.net/ont/geosparql#asWKT(x1,g1)
ans9(g2,x2) :- http://www.opengis.net/ont/geosparql#asWKT(x2,g2)

GeoSPARQL-to-SQL translation




```
SELECT *
FROM (
SELECT DISTINCT
   1 AS "x1QuestType", ...
   10 AS "g1QuestType", NULL AS "g1Lang",
CONCAT('<http://www.opengis.net/def/crs/EPSG/
0/4326> ' , ST_AsText(QVIEW2."strdfgeo")) AS "g1"
 FROM
geo_values QVIEW1,
geo_values QVIEW2
WHERE
(ST_Overlaps(QVIEW2."strdfgeo",QVIEW1."strdfgeo"
)) AND
QVIEW1."id" IS NOT NULL AND
QVIEW1."strdfgeo" IS NOT NULL AND
QVIEW2."id" IS NOT NULL AND
QVIEW2."strdfgeo" IS NOT NULL
) SUB_QVIEW
```

Example GeoSPARQL Query

```
PREFIX gag: <http://geo.linkedopendata.gr/gag/ontology/>
PREFIX clc: <http://geo.linkedopendata.gr/corine/ontology#>

SELECT distinct ?x1 ?lu
WHERE {
        ?x1 geo:asWKT ?g1 .
        ?x2 geo:asWKT ?g2 .
        ?x2 clc:hasLandUse ?lu .
        FILTER(geof:sfIntersects(?g1,?g2)) }
```

Raster Data Sources

```
[[ mappingId      chicago
  target         :{rid} rdf:type :rasterCell ;  :hasGeometry {rast} .
  source          select rid,rast from chicago;
mappingId       gadm
target          : {id_0} rdf:type :AdministrativeDivision; geo:hasGeometry :{gid} .
                 :{gid} geo:asWKT {geom}^^geo:WKTLiteral .
 source          select * from usa_adm2 ]]
```

GeoTIFF image of Chicago imported in PostGIS as table (raster geometries)

Data sources

CHICAGO[rid | rast]

USA_ADM2[gid | id_0 | iso | name_0 | id_1 | name_1 | id_2 | name_2 |geom]

Shapefile describing USA administrative divisions and boundaries (vector geometries)

Example Query

Retrieve administrative divisions that intersect with raster cells of the GeoTIFF image of Chicago.

```
SELECT ?adm
WHERE{
   ?r rdf:type :rasterCell .
   ?r :hasGeometry ?rast .
   ?adm rdf:type :AdministrativeDivision .
   ?adm geo:hasGeometry ?g .
  ?g geo:asWKT ?geom .
FILTER(geof:sfIntersects(geom,rast))
```

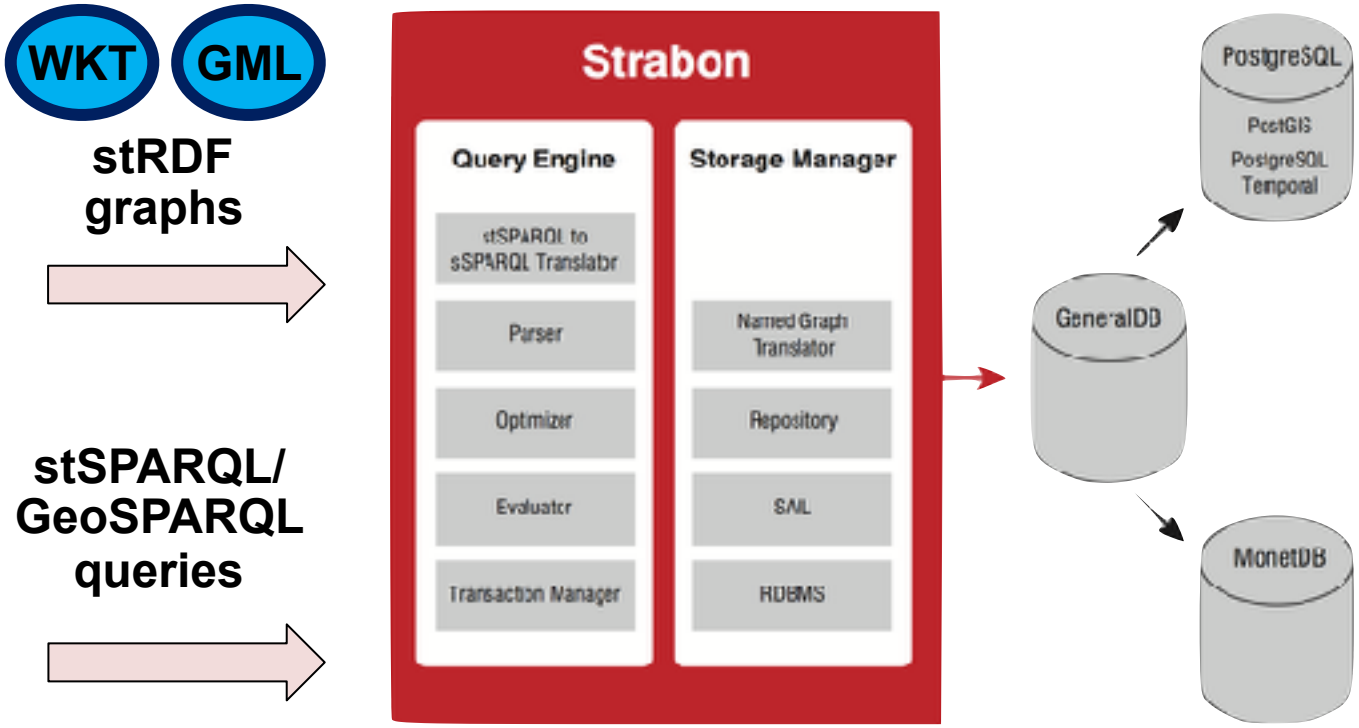
Vector geometries will be bound

Raster geometries will be bound

Implemented systems, evaluation and comparison

Strabon (ISWC 2012, ESWC 2013)

Find more at: http://strabon.di.uoa.gr



Strabon - Geospatial features

Support for:

- stRDF and stSPARQL
- GeoSPARQL (core, geometry extension, geometry topology extension)
- Multiple Coordinate Reference Systems (CRS)
- Builds on Sesame RDBMS
- Geospatial relational database as back-end (PostGIS, MonetDB)
- R-tree index

Parliament

- Developed by Raytheon BBN Technologies (Dave Kolas). Available at: http://www.parliament.semwebcentral.org/
- First GeoSPARQL implementation. Supports:
  - Core
  - Topology vocabulary
  - Geometry
  - Geometry Topology
  - RDF entailment
  - Multiple CRS
  - R-tree index

uSeekM

OPEN SAHARA

Spatial plugin for Sesame by OpenSahara. Supports:

- GeoSPARQL
  - Core
  - Topology Vocabulary
  - Geometry
  - Geometry Topology
  - RDFS entailment
- No multiple CRS. Only WGS84
- Open source (Apache v2.0). Available at: https://dev.opensahara.com/projects/useekm

GraphDB

- Developed by Ontotext. Former OWLIM.
- GeoSPARQL support
- Apache Lucene index
- Closed source. Available at: https://ontotext.com/products/graphdb/editions/

Allegrograph

FRANZ INC.

- Quad store developed by Franz Inc
- Closed source. Available at http://www.franz.com/agraph/allegrograph/
- No GeoSPARQL support
- Supports only points
- Only a few spatial operations supported (Buffer, Bounding Box, Distance)

OpenLink Virtuoso

- Developed by OpenLink. Available at: http://virtuoso.openlinksw.com/ Supports:
- No GeoSPARQL
- Points only
  - Serialized as typed literals
- Spatial operations (subset of SQL/MM)
- Multiple CRS
- R-tree

Stardog

- Limited GeoSPARQL support:
  - WKT literals. Native support for points. Use of JTS library for polygons
  - Operators
    geof:relate, geof:distance, geof:within, geof:nearby, geof:area
  - Geospatial features only offered in enterprise edition

Brodt et al.

- Built on top of RDF-3X by University of Stuttgart
- No GeoSPARQL support
- Geometries represented as typed WKT literals
- Only WGS84 supported
- OGC-SFA spatial operations as SPARQL filter functions
- R-tree supported (but only used for spatial selections)

Perry PhD thesis

- Implementation on top of Oracle 10g by Wright State University
- Support for SPARQL-ST
- GeoRSS GML serialization of geometries
- Spatial and temporal variables
- Spatial and temporal filters (RCC8, Allen)
- R-tree support

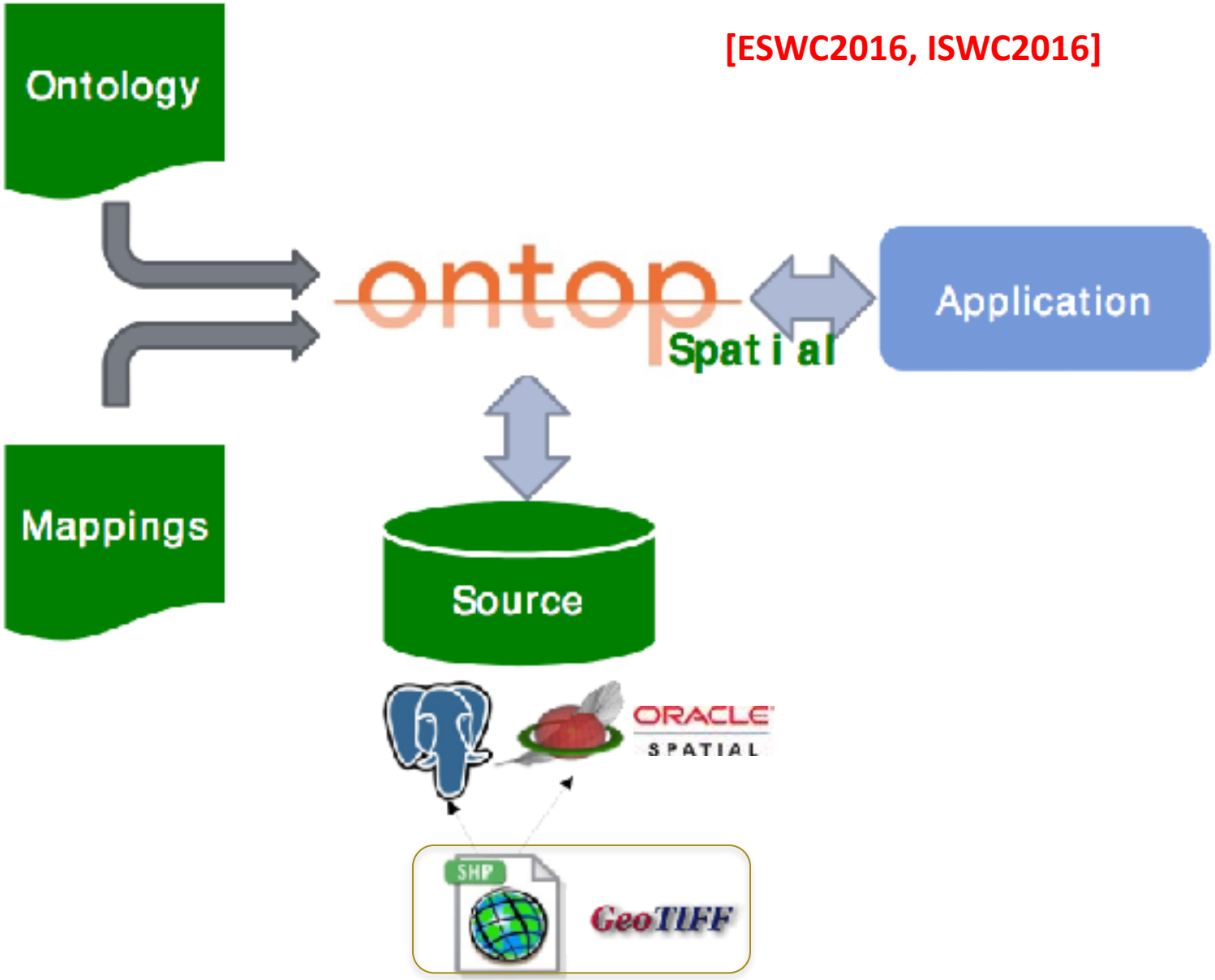
Oracle spatial and Graph

ORACLE SPATIAL

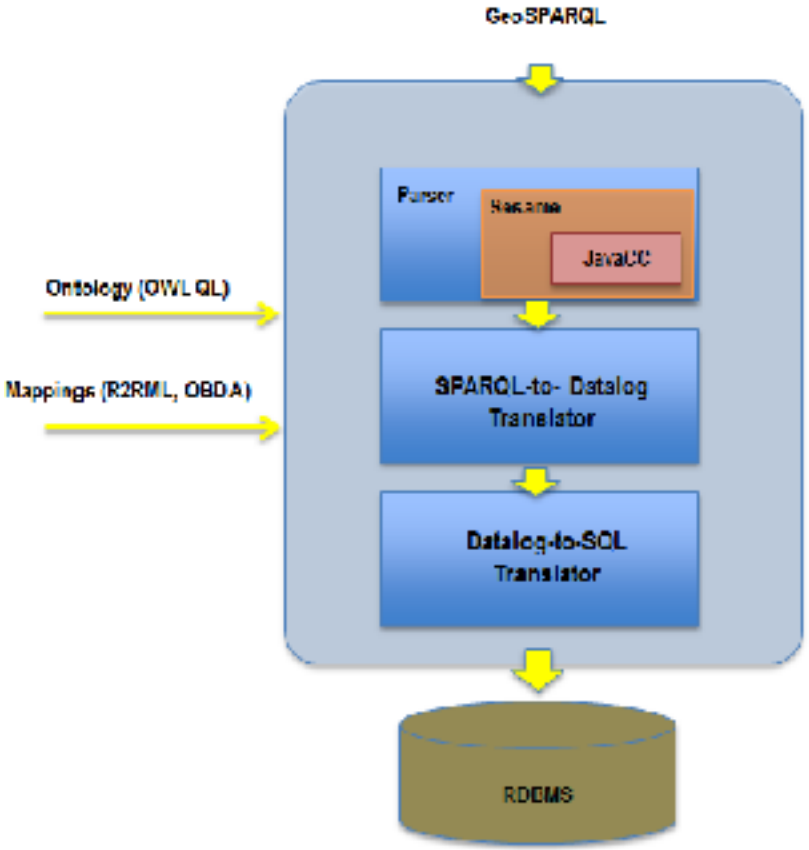
- Developed by Oracle
- GeoSPARQL support
- CRS support
- Recently added support for virtual RDF graphs (as of Oracle Spatial and Graph 12c Release 2)

http://ontop-spatial.di.uoa.gr

[ESWC2016, ISWC2016]



Architecture overview (extending Ontop)



Evaluation

Geographica Benchmark

- Evaluation of the state-of-the-art geospatial RDF stores [Garbis et al., ISWC 2013]
  - Real workload
  - Synthetic workload
- Stresses all recent systems in heavily spatial queries and workload.
- Open source Java framework available at http://geographica.di.uoa.gr

Real workload

Dataset	Size	#triples	#geometries	#points per geometry (avg)
Geonames	45MB	400K	22000	1
DBpedia	89MB	430K	8000	1
LGD	29MB	150K	12000	1
GAG	33MB	4K	325	400
CLC	401MB	630K	45000	140
Hotspots	90MB	450K	37000	4

Execution times in real workload

Non topological construct functions

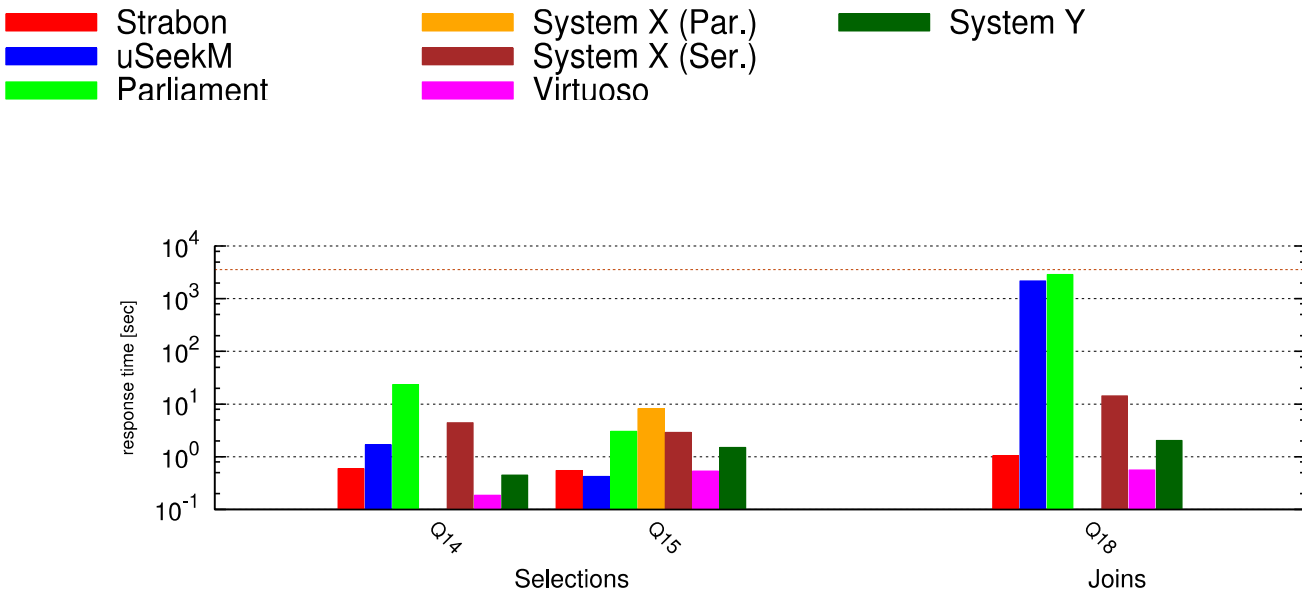
Spatial selections

Spatial joins

Aggregate functions

Query	Strabon	uSeekM	Parliament	System X - Parallel	System X - Serial
Q1	42.33	38.11	152.71	62.58	293.85
Q2	22.48	21.47	90.23	44.02	204.65
Q3	29.48	27.06	98.56	45.86	213.47
Q4	7.65	3.22	23.16	19.62	309
Q5	14.60	4.17	21.63	23.6	236.6
Q6	23.62	19.58	-	39.87	199.25
Q7	0.36	1.22	2.42	-1	-1
Q8	0.42	0.57	7.69	-1	-1
Q9	0.83	1.27	35.03	-1	-1
Q10	0.73	1.51	76.85	-1	-1
Q11	2.66	2.96	195.07	-1	-1
Q12	0.79	0.55	2.39	8.87	6.07
Q13	0.82	0.89	63.14	-1	-1
Q14	0.5	2.29	24.34	13.33	11.35
Q15	0.5	0.09	3.44	10.24	10.27
Q16	2.79	5.52	53.2	-1	-1
Q17	3.00	1.6	35.89	-1	-1
Q18	4.52	2233.73	2880.2	>1h	14.51
Q19	1272.54	>1h	>1h	>1h	>1h
Q20	115.93	>1h	>1h	>1h	396.29
Q21	113.26	>1h	>1h	>1h	409.54
Q22	26.33	>1h	>1h	-1	-1
Q23	26.29	>1h	>1h	-1	-1
Q24	26.66	>1h	>1h	-1	-1
Q25	342.87	>1h	>1h	>1h	1,879.45
Q26	343.3	534.81	2040	909.18	>1h
Q27	343.72	>1h	>1h	-1	>1h
Q28	3.56	-	-	-	-
Q29	258.35				

Adding systems with limited geospatial functionalities



Geospatial RDF stores vs OBDA

Dataset	*Table Size*	*No. of rows/ geometries*	*Avg #points/ geometry*
Corine Land Cover (CLC)	283MB	44834	187.84
Hotspots	35 MB	37048	5
Global Administrative Geography (GAG)	24 MB	326	3020.14
OSM-Buildings	42 MB	155474	6.5
OSM-Landuse	20 MB	40220	19.4
OSM-places	2.4 MB	13043	1
OSM-points	12 MB	61664	1
OSM-railways	2 MB	4996	13.3
OSM-roads	250 MB	514403	19
OSM-waterways	16 MB	20565	39.84

Queries

No	Query	#BGPs	results
00	Equals GADM P	1	0
01	Contains_GADM_P	1	9
02	Contains GADM P	1	0
03	Equals GADM L	1	1
04	Overlaps GADM L	1	0
05	Contains GADM L	1	0
06	Intersects CLC L	1	5
07	Contains_CLC_L	1	0
08	Equals_CLC_L	1	5
09	Overlaps CLC L	1	0
10	Overlaps CLC P	1	132
11	Intersects CLC P	1	533
12	Contains_CLC_P	1	401
13	Equals_CLC_P	1	0
14	Intersects_LGD_P	2	2749
15	Intersects_LGD_B	2	2749
16	Intersects_LGD_PL	2	2626
17	Intersects LGD P	2	2522
18	Intersects_LGD_LU	2	2722
19	Intersects_LGD_ROA	2	2387
20	Intersects_LGD_bigP	1	729189
21	Intersects_LGD_P2	3	5

Table 1: Spatial selections description

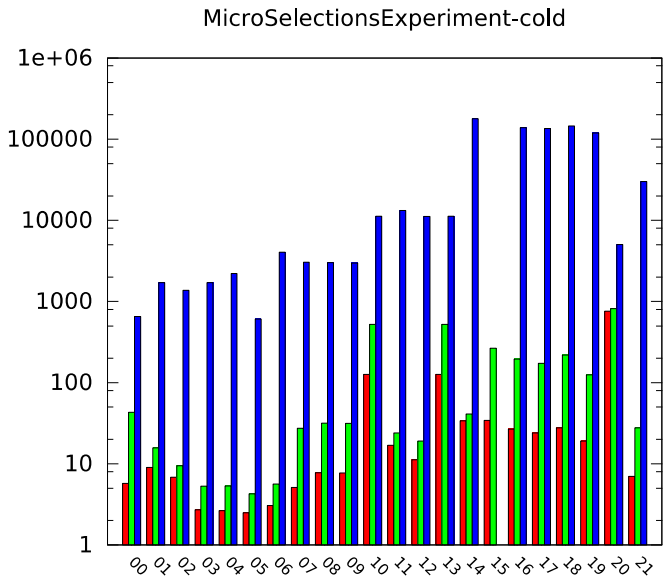
No	Query	#BGPs	results
00	Within CLC GADM	2	34114
01	Intersects GADM GADM	2	1556
02	Overlaps GADM CLC	2	17035
03	Intersects LGD GADM	3	154725
04	Intersects_LGD_LGD_Mus	4	2
05	Intersects_LGD_GADM	2	819319
06	Intersects LGD LGD	1	3686229
07	Crosses LGD LGD Roads	4	178602

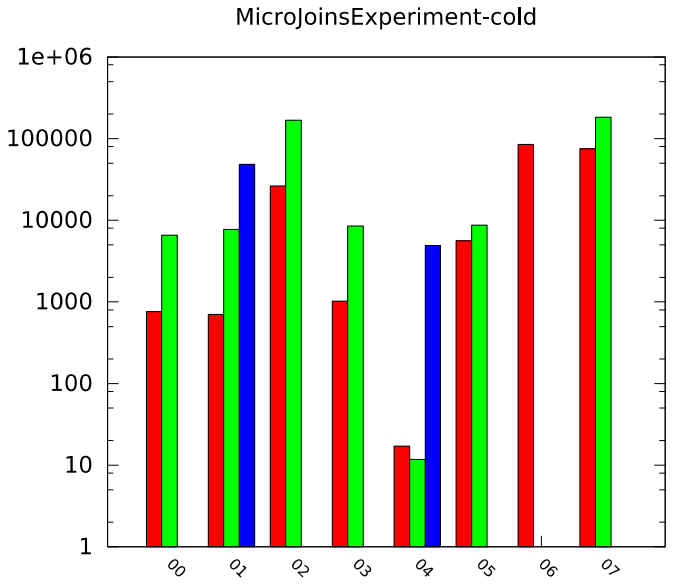
Table 2: Spatial joins description

Highly selective query

Poorly selective query

Evaluation





Ontop-spatial
Strabon
System O

Translated queries

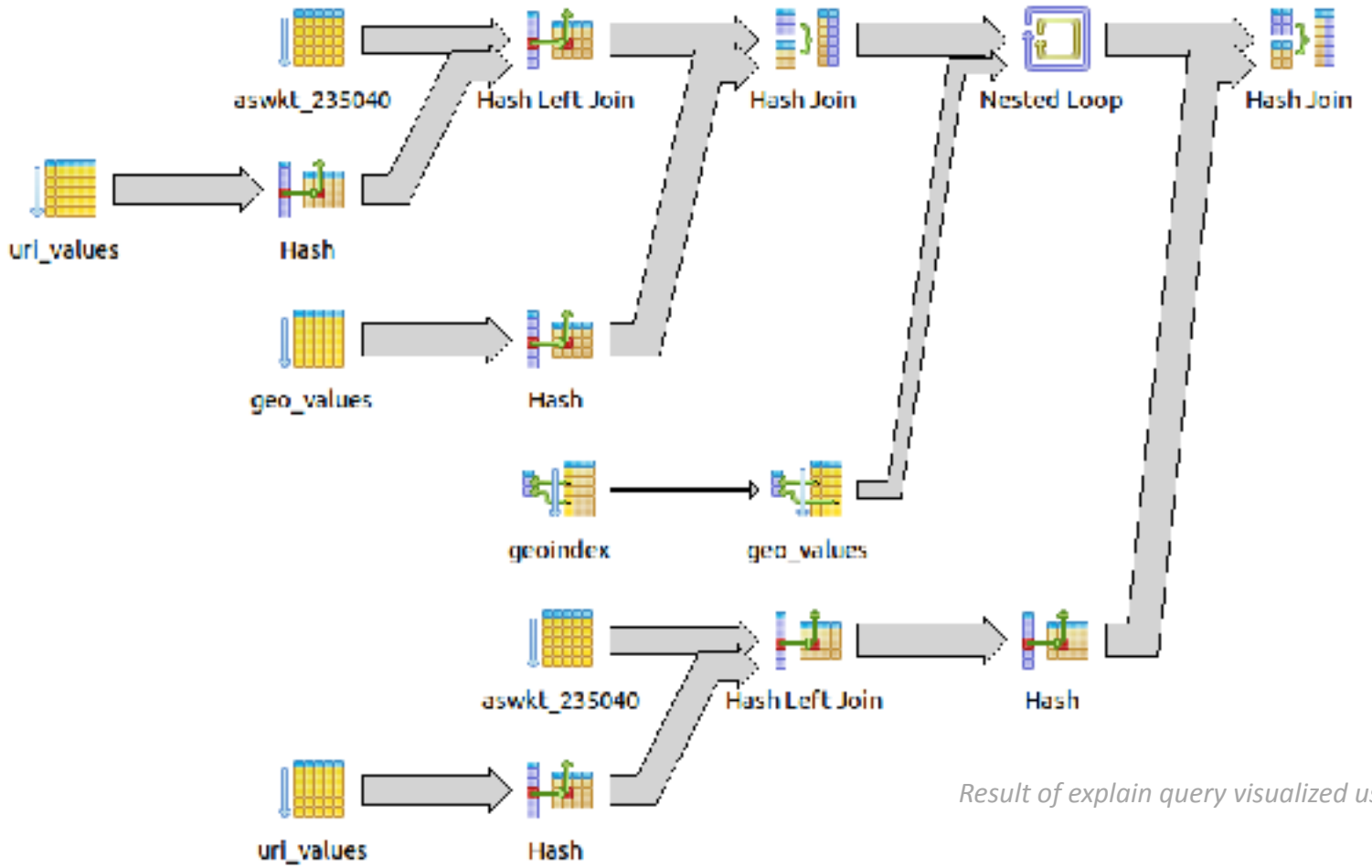
- Strabon query

```
SELECT a0.subj, u_s2.value, a2.subj, u_s1.value
FROM aswkt_855211 a0
INNER JOIN geo_values l_o2  ON (l_o2.id = a0.obj)
INNER JOIN geo_values l_o1 ON
((ST_Within(l_o1.strdfgeo,  l_o2.strdfgeo)))
INNER JOIN aswkt_135992 a2  ON (a2.obj = l_o1.id)
LEFT JOIN uri_values u_s2  ON (u_s2.id = a0.subj)
LEFT JOIN uri_values u_s1  ON (u_s1.id = a2.subj)
```

- Ontop-spatial query

```
SELECT 1 AS "s1QuestType", NULL AS "s1Lang", ('http://geo.linkedopendata.gr/clc/'
 || REPLACE(..... || '/') AS "s1",
 1 AS "s2QuestType", NULL AS "s2Lang", ('http://geo.linkedopendata.gr/gag/ont/'
 || REPLACE(...'/') AS "s2"FROM clc QVIEW1,
 gag QVIEW2 WHERE QVIEW1."gid" IS NOT NULL
 AND QVIEW1."geom" IS NOT NULL
 AND QVIEW2."gid" IS NOT NULL
 AND QVIEW2."geometry" IS NOT NULL
 AND(ST_Within(QVIEW1."geom",QVIEW2."geometry"))
```

Spatial join query 6: Strabon query execution



Result of explain query visualized using PgAdmin III

DB statistics for spatial join 6

Strabon db statistics

node type	count	sum of times	% of query
Hash	4	751.907 ms	0.0 %
Hash Join	2	18,036.091 ms	0.0 %
Hash Left Join	2	14,028.716 ms	0.0 %
Index Scan	1	526,737.250 ms	0.3 %
Nested Loop	1	164,863,821.747 ms	99.7 %
Seq Scan	5	1,780.556 ms	0.0 %

Ontop db statistics

node type	count	sum of times	% of query
Append	1	699.975 ms	0.1 %
Index Scan	49	514,878.104 ms	61.0 %
Nested Loop	49	326,798.817 ms	38.7 %
Result	1	958.691 ms	0.1 %
Seq Scan	49	1,348.070 ms	0.2 %

Tables produced using https://explain.depesz.com/

Performance Evaluation: Strabon vs. Ontop-spatial on a 30 GB dataset

Operation (geof:intersects)	Selectivity	Geometry types	Strabon	Ontop-spatial	Remarks
Spatial Selection	high	* (irrelevant)	100 msecs	100 msecs	
Spatial Selection	low	Point-Polygon	100 msecs	100 msecs	
Spatial Selection	low	Polygon-Polygon	500 msecs	100-200 msecs	
Spatial Join	high	Point - Polygon	< 1000 msecs	< 1000 msecs	
Spatial Join	high	Polygon-Polygon	100000 msecs	100000 msecs	
Spatial Join	low	Polygon-Polygon	>40 mins	10 mins	Sometimes the difference here is order(s) of magnitude

Performance Evaluation and Scalability of Strabon and Ontop-spatial

- Defined and used the benchmark Geographica (http://geographica.di.uoa.gr/).
- Strabon has better performance and functionality than Parliament, uSeekM, System X, Virtuoso, System O, and System Y (longer version of ISWC 2013 paper).
- Ontop-spatial has better performance than Strabon and System O (long version of ISWC 2016 paper).

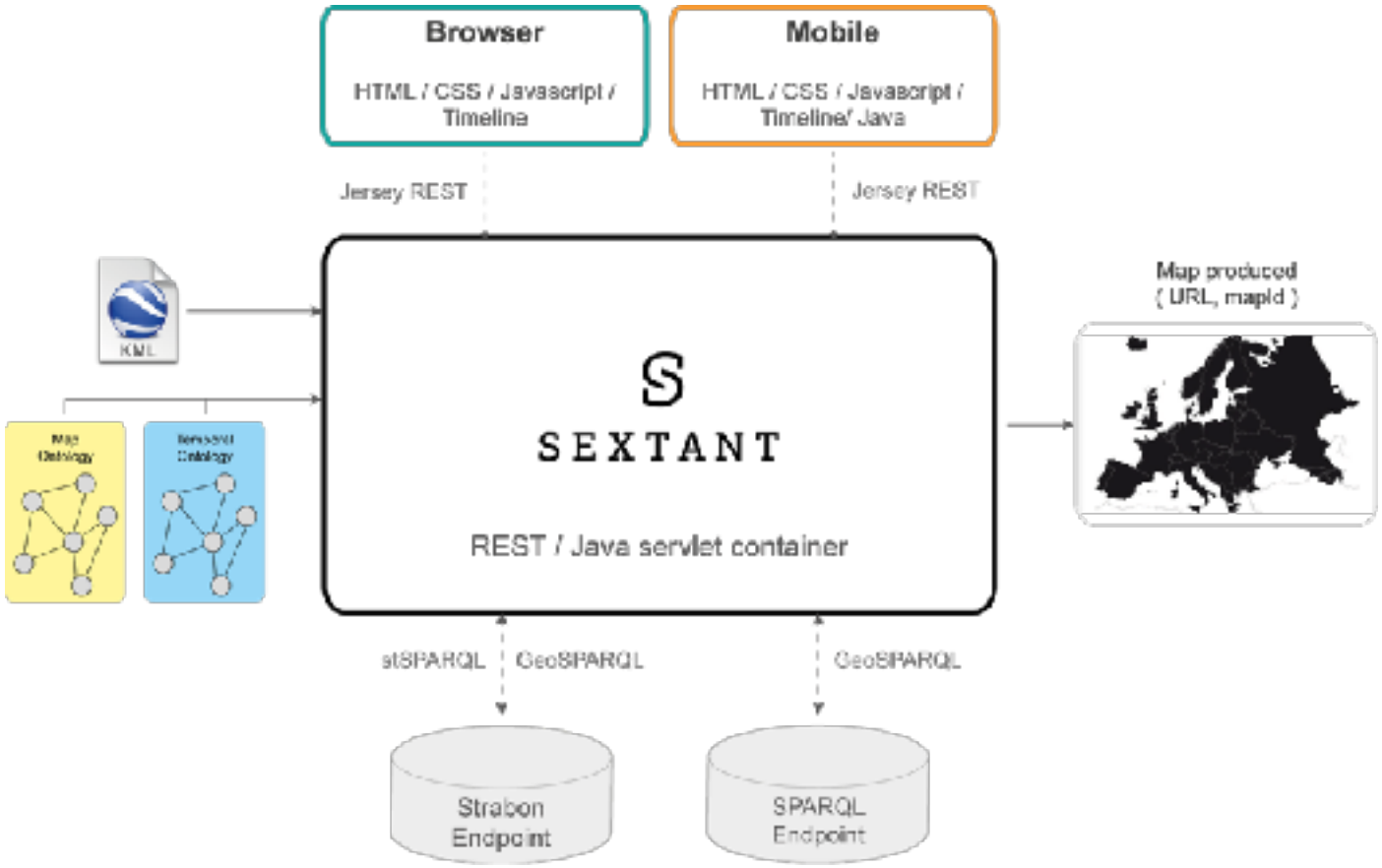
Scalability

- Strabon and Ontop-spatial can scale to **100GB of data and answer queries in milliseconds** if the geometries are points and/or the selectivity of the query is high.
- More complex geometries have an impact on performance.

Visualizing Time-Evolving Linked Geospatial Data

Architecture

Find more at: http://sextant.di.uoa.gr



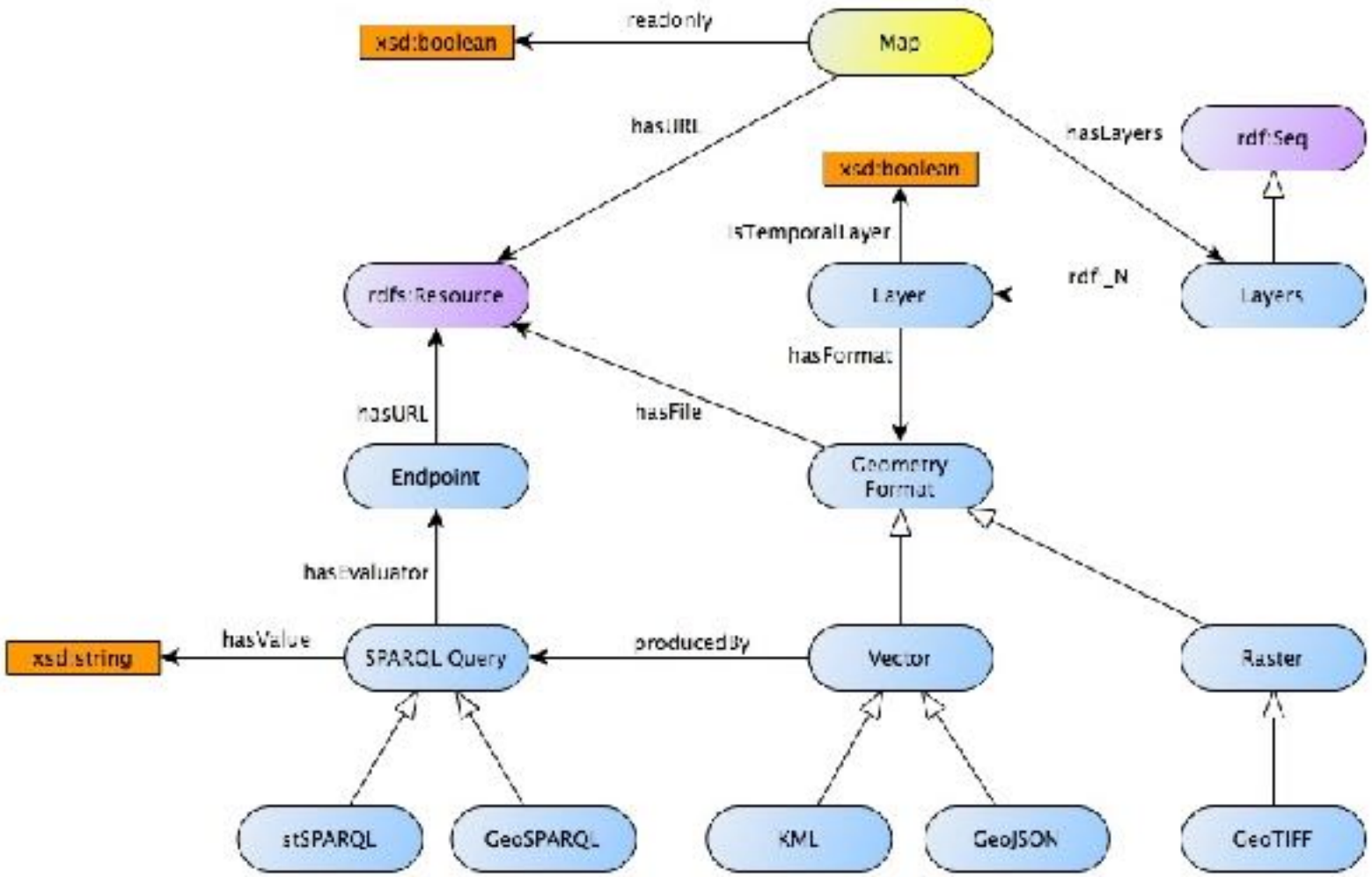
Thematic Maps

Definition: A *thematic map* is a type of map designed to show a particular theme connected with a specific geographic area. These maps can portray physical, social, political, cultural, economic, sociological, agricultural, or any other aspects of a city, state, region, nation, or continent.

SPARQL Endpoints + Geospatial File Formats =

Thematic Maps

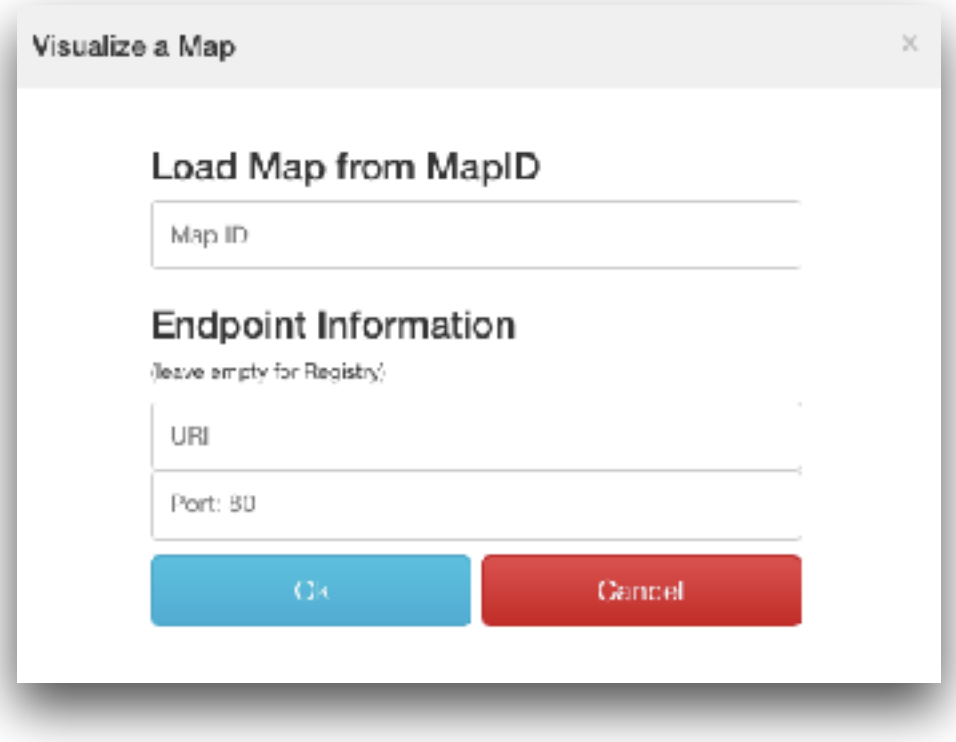
Map Ontology



Map Sharing

Share maps using map URI, or the load map modal

Map URI: http://<domain>/Sextant/?mapid=<mapID>



Temporal Dimension

Strabon DBMS and SIMILE Timeline

Implementation of he valid time component of stRDF and stSPARQL in system Strabon allows us to query both the spatial and the temporal dimension. Enriching our results with temporal information allows us to create layers with valid time. Using the SIMILE Timeline widget we can make these layers appear and disappear from the map according to their valid time.

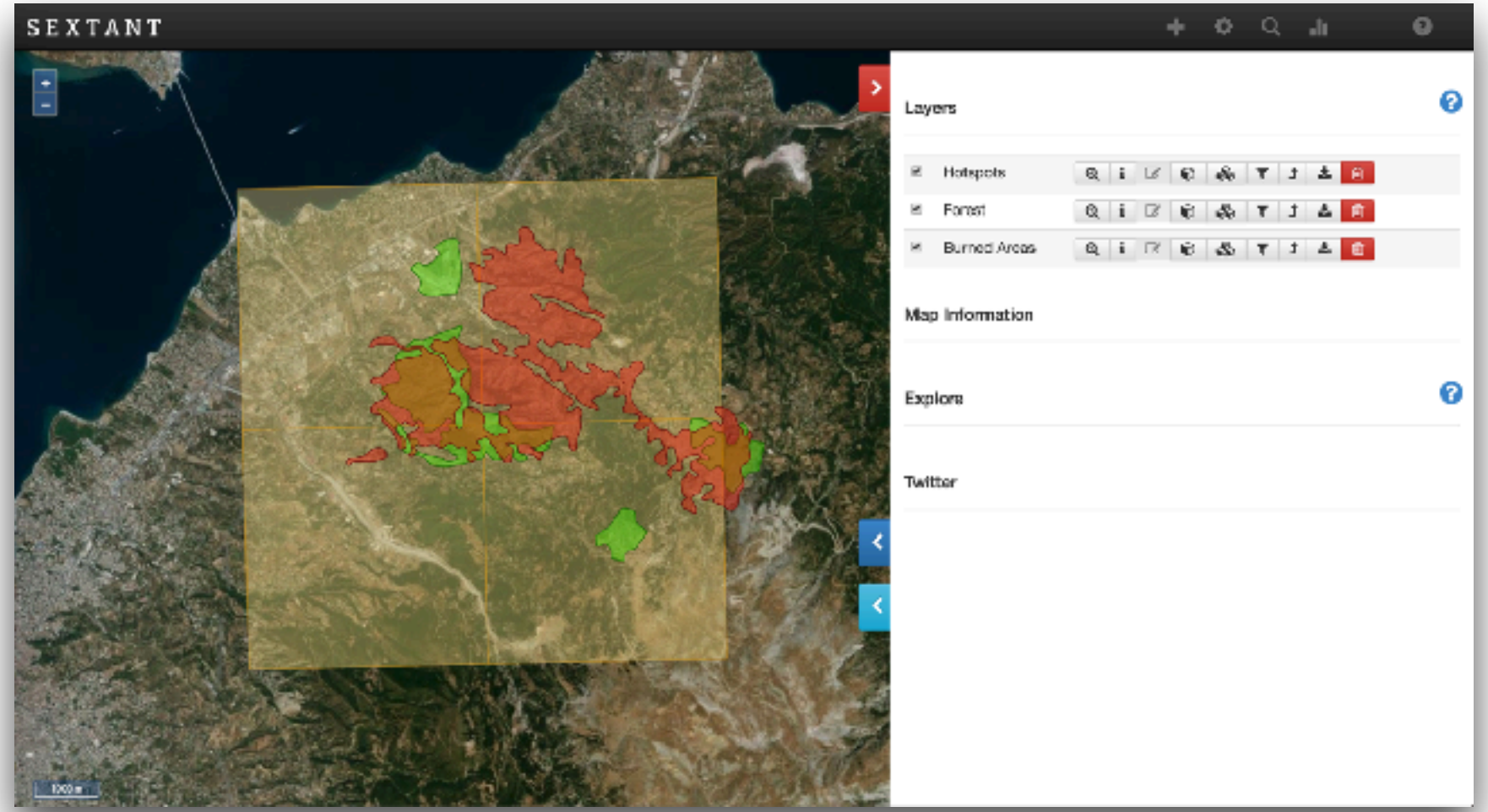
Temporal Dimension

Query example in stSPARQL

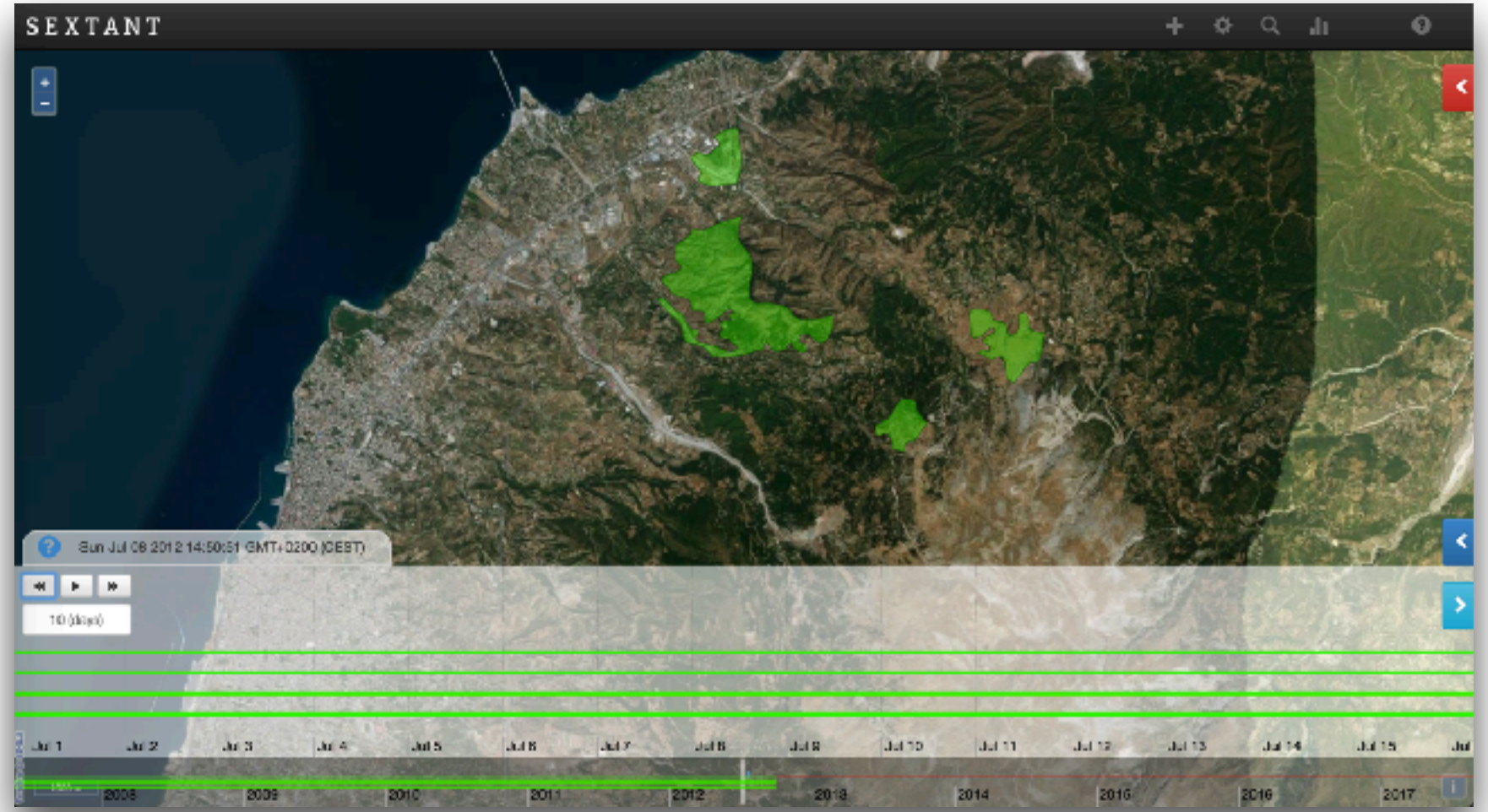
```
PREFIX noa: <http://teleios.di.uoa.gr/ontologies/noaOntology.owl#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX strdf: <http://strdf.di.uoa.gr/ontology#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

SELECT  DISTINCT (strdf:transform(?hGeo, <http://www.opengis.net/def/crs/EPSG/0/4326>) as ?
geometry) ?time
WHERE {
    ?h  rdf:type noa:Hotspot ;
        noa:hasAcquisitionTime ?time;
        noa:hasGeometry ?hGeo .
    FILTER(?time = "2012-07-18T11:30:00"^^<http://www.w3.org/2001/XMLSchema#dateTime>)
    FILTER (strdf:within(?hGeo, "POLYGON((21.70 38.23, 21.70 38.33, 21.91 38.33, 21.91
38.23, 21.70 38.23));http://www.opengis.net/def/crs/EPSG/0/4326”^^strdf:WKT))
}
```

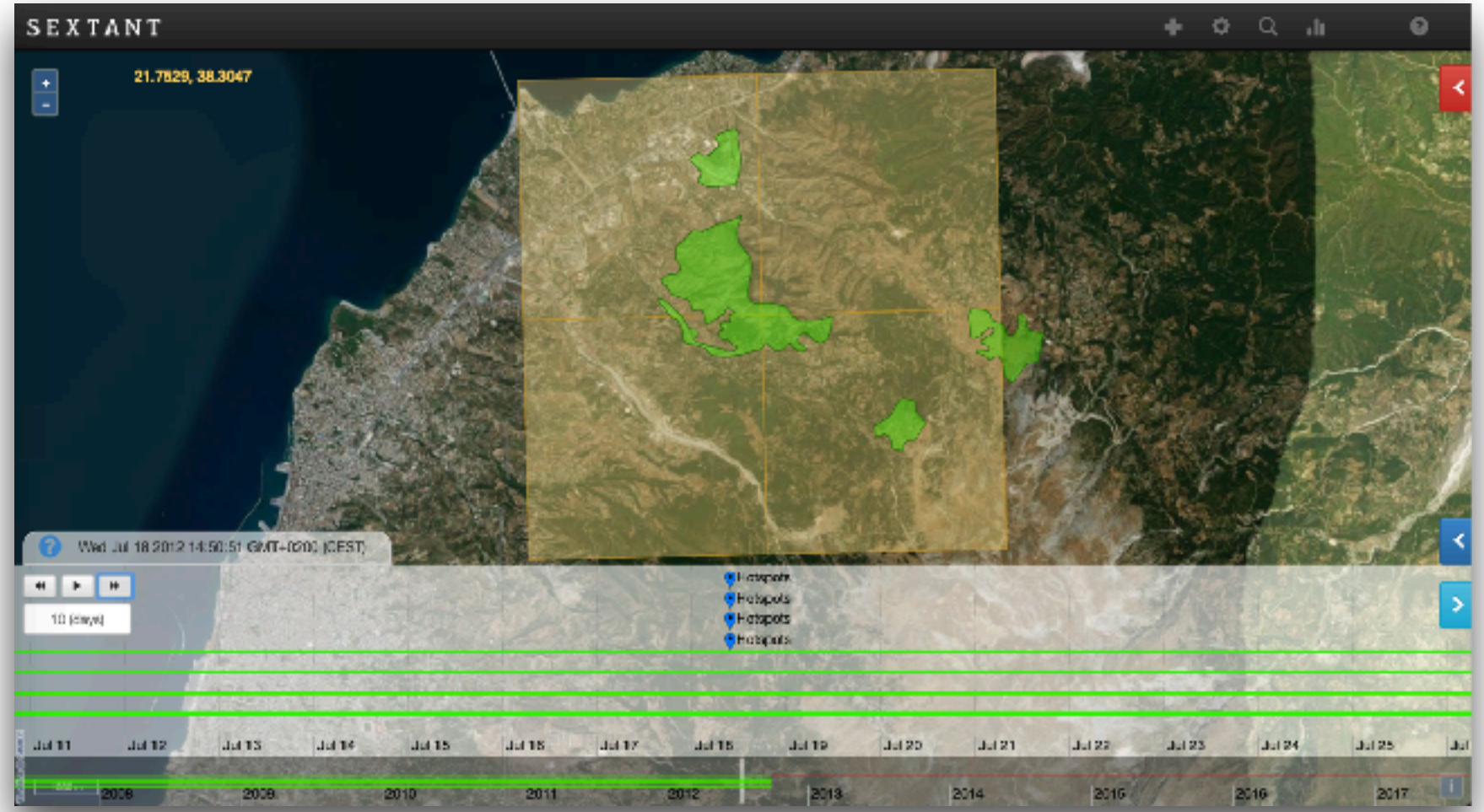
Temporal Dimension



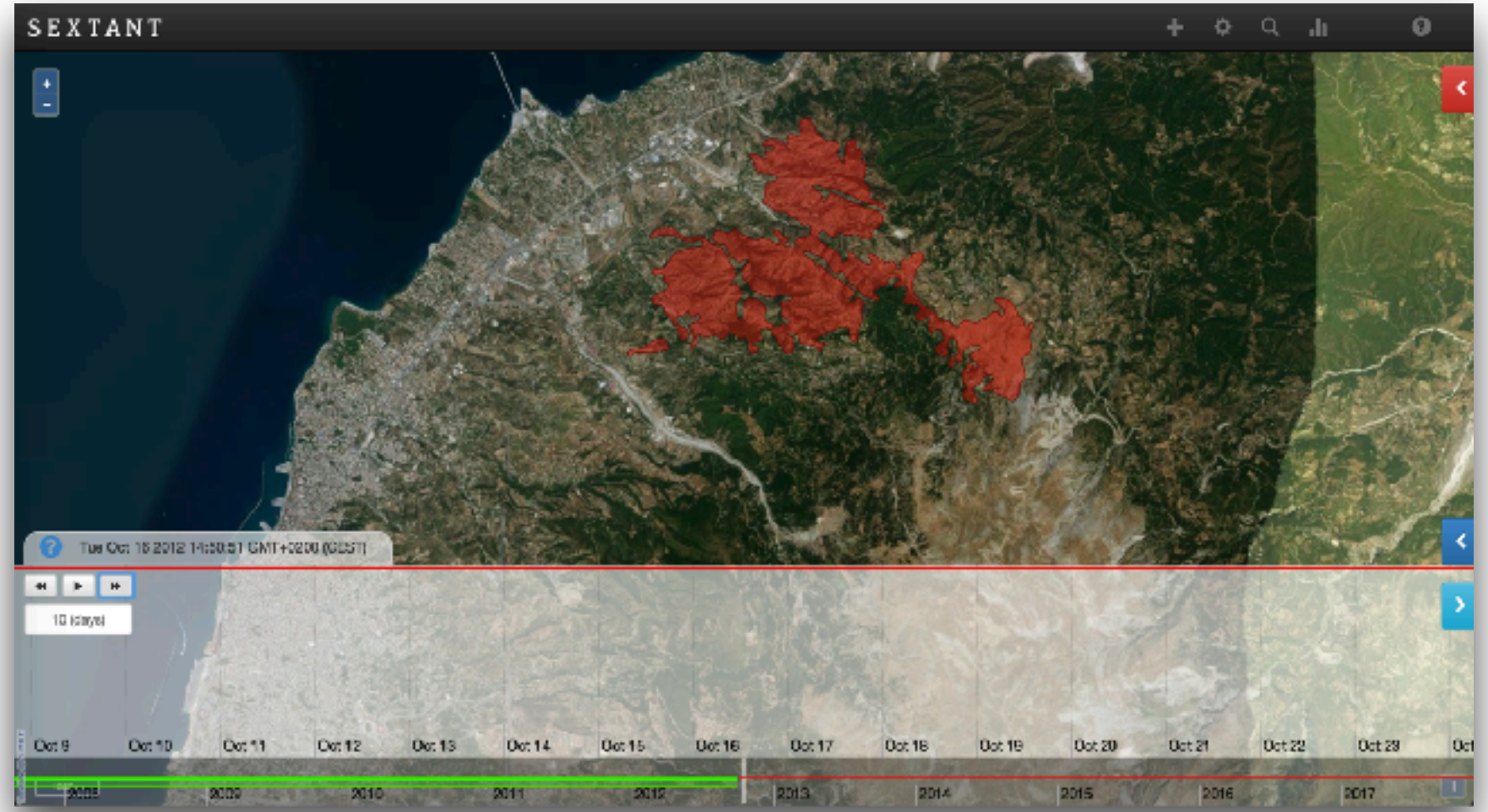
Temporal Dimension



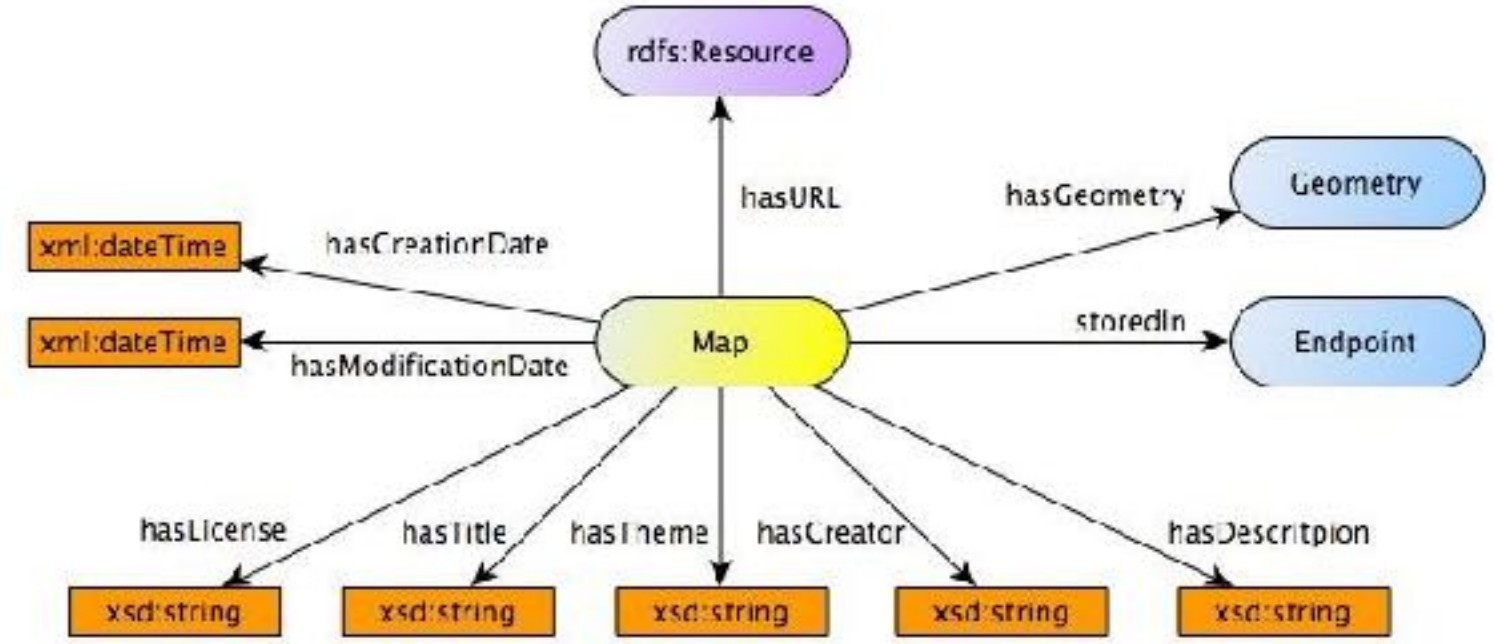
Temporal Dimension



Temporal Dimension

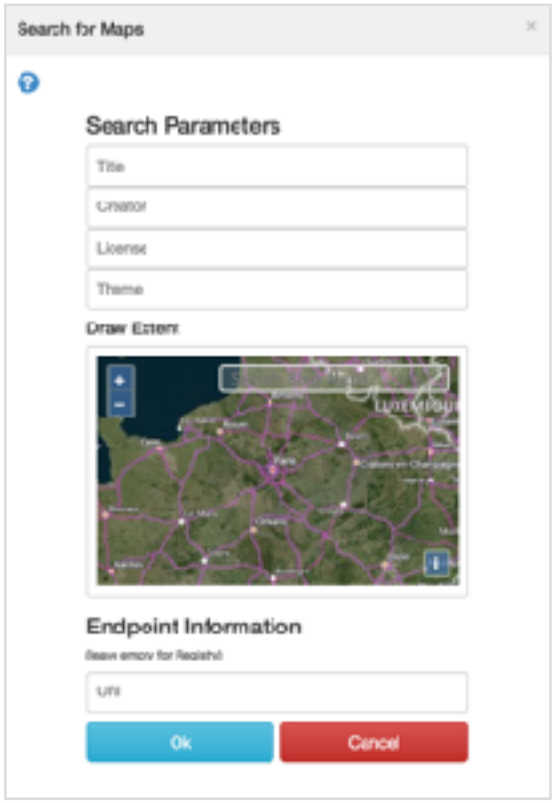


Map Metadata



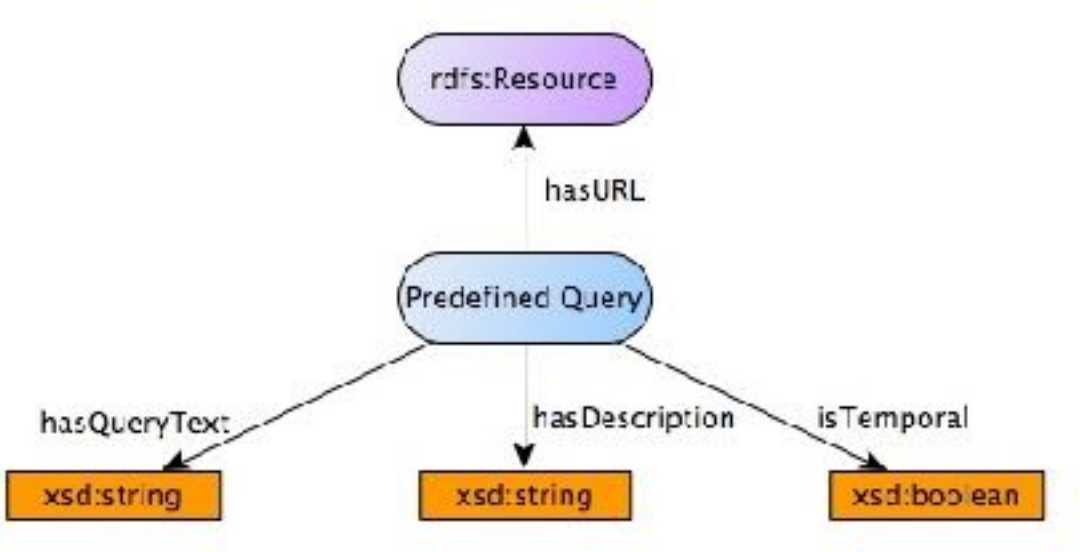
Map Registry

Definition: *Map registries*, are SPARQL endpoints that hold all the map information and metadata to assist us in saving and retrieving the maps.

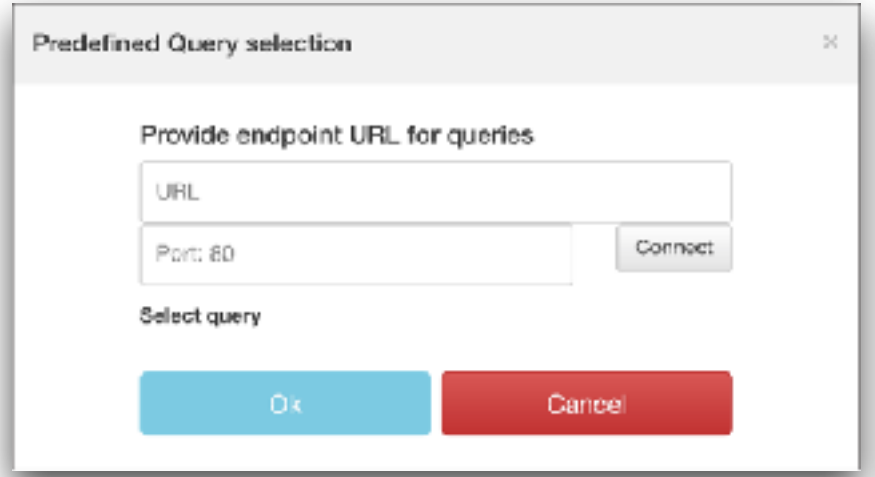


Predefined Queries

Queries that are **created by an expert** and are stores as triples in a SPARQL endpoint. **Non-expert users** can provide the URL of the endpoint and get a list of the **descriptions** of all the predefined queries available, then select one and visualize it on the map.

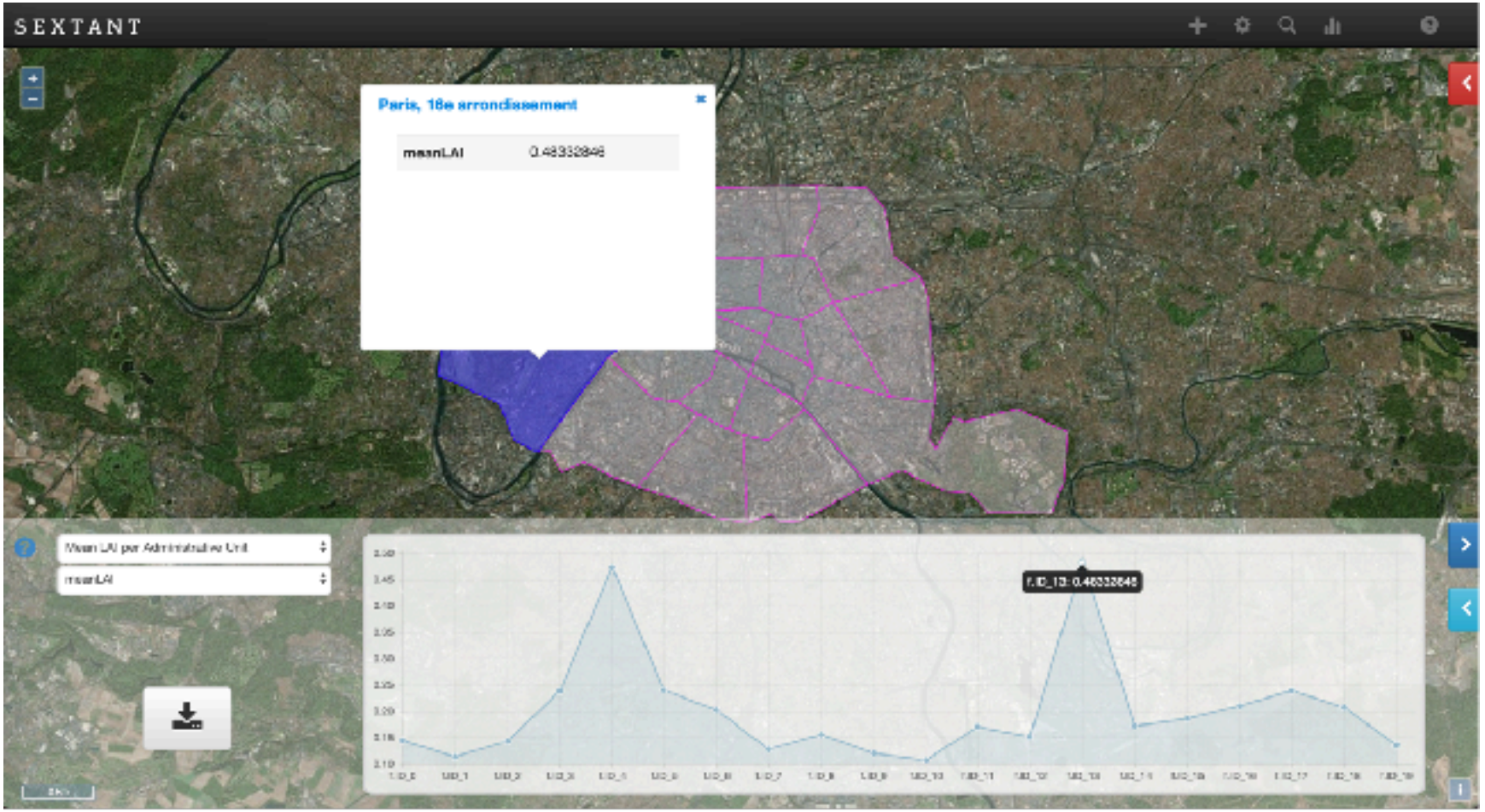


Predefined Queries



Statistical Charts

Automatic creation of charts over a layer's attribute



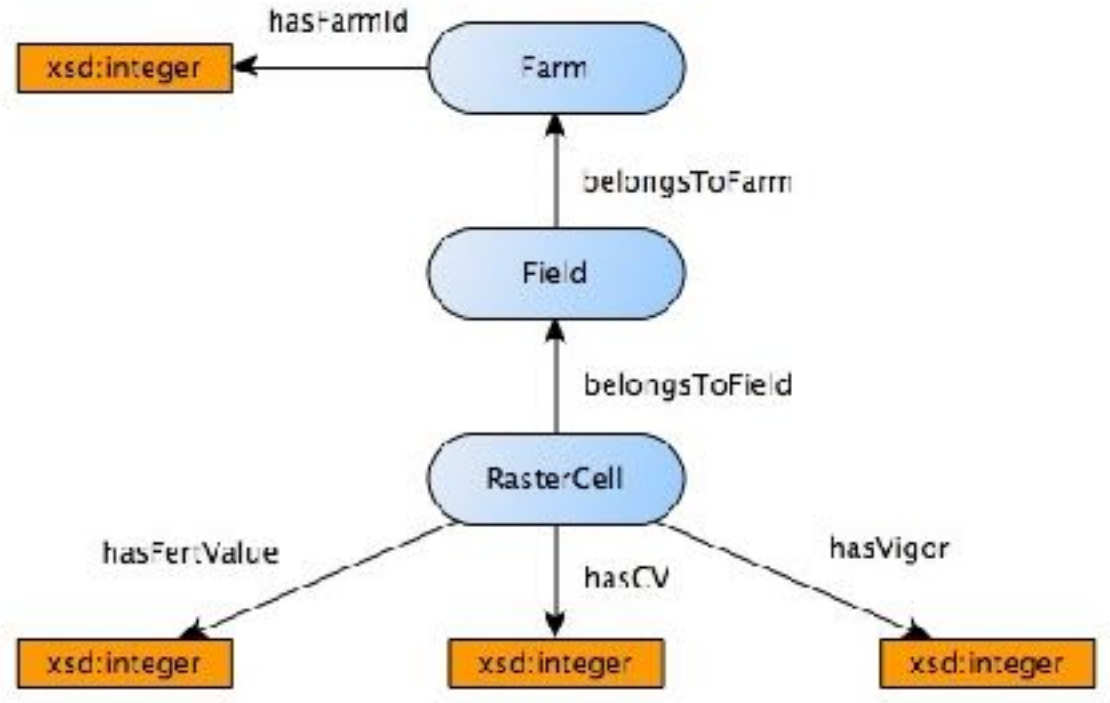
Statistical Charts

Enhancement of Datasets

- The RDF **Data Cube** Vocabulary (W3C recommendation).
- Define the **dimensions** and the **measures** of the dataset.
- Use the classes **qb:DimensionProperty**, **qb:MeasureProperty**, and the property **qb:order** from Data Cube to enhance our schema.
- A **Self-adjusting query** to create all possible charts.

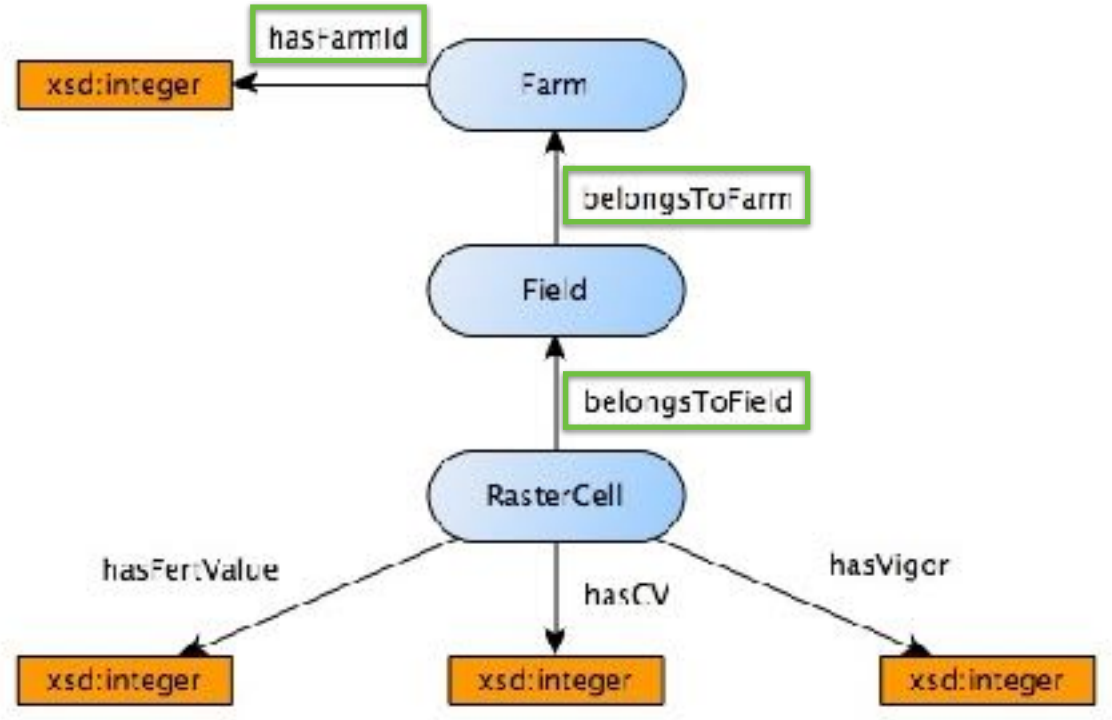
Statistical Charts

Enhancement of Ontology Schema



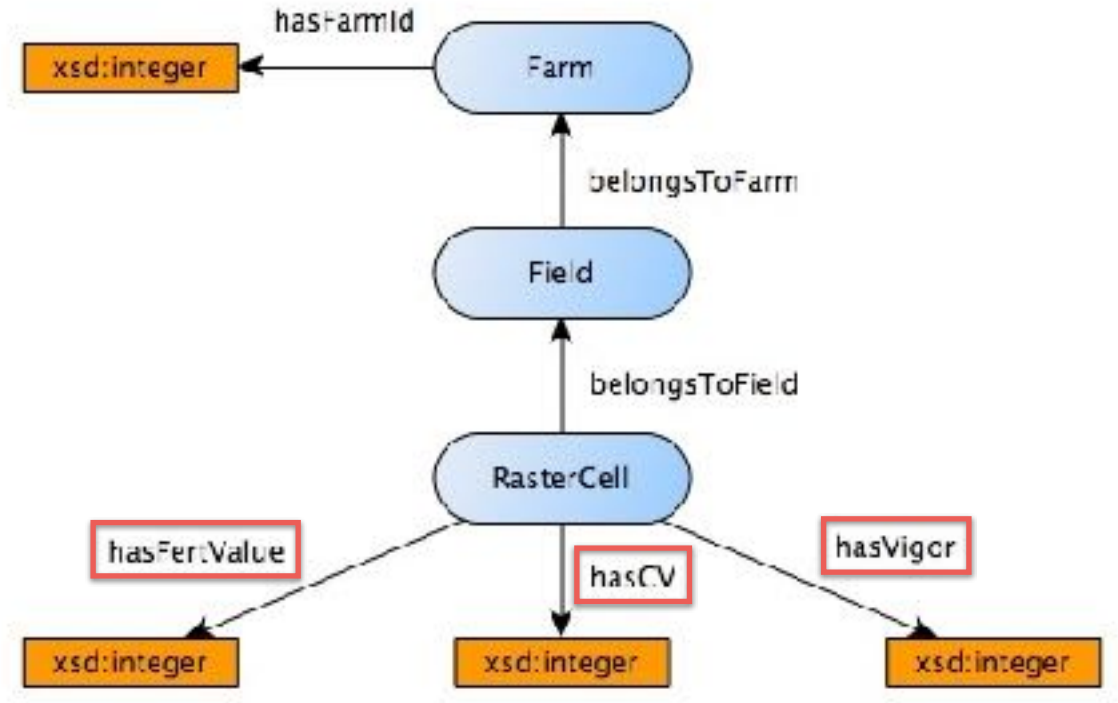
Statistical Charts

Enhancement of Ontology Schema



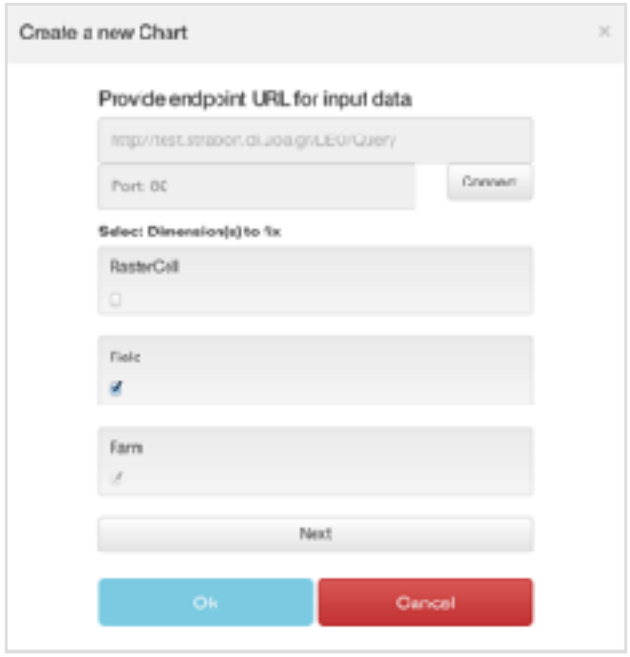
Statistical Charts

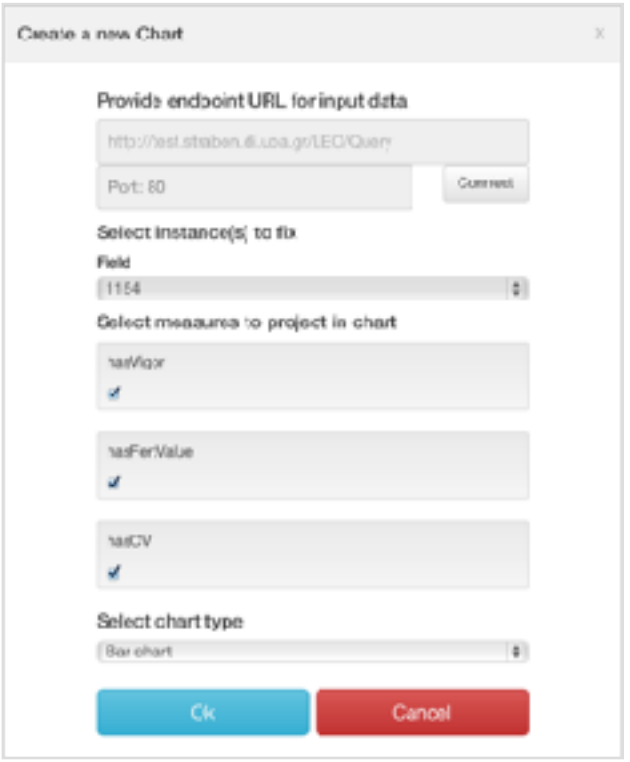
Enhancement of Ontology Schema



Statistical Charts

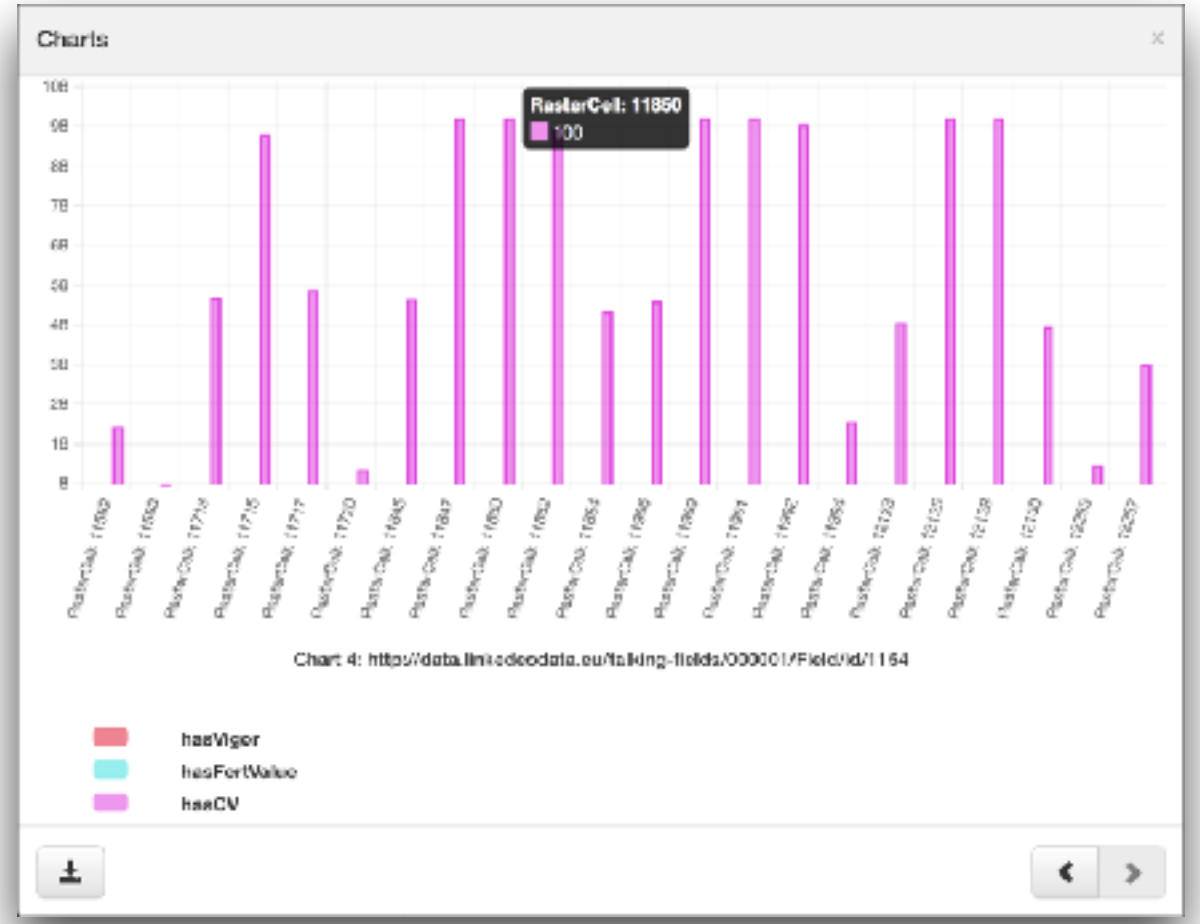
Self-adjusting Query





Statistical Charts

Self-adjusting Query



Statistical Charts

Self-adjusting Query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX tf: <http://data.linkedeodata.eu/talking-fields/ontology#>

SELECT DISTINCT ?RasterCell ?hasVigor ?hasFertValue ?hasCV
WHERE {
    ?Farm rdf:type tf:Farm .
    ?Field rdf:type tf:Field .
    ?RasterCell rdf:type tf:RasterCell .

    ?Field tf:belongsToFarm ?Farm .
    ?RasterCell tf:belongsToField ?Field .

    OPTIONAL { ?RasterCell tf:hasVigor ?hasVigor. } .
    OPTIONAL { ?RasterCell tf:hasFertValue ?hasFertValue. } .
    OPTIONAL { ?RasterCell tf:hasCV ?hasCV. } .

    ?Field rdf:type ?typeField .
    FILTER ( ?Field = <http://data.linkedeodata.eu/
                       talking-fields/000001/Field/id/1154>
            && ?typeField = tf:Field ) .

}
ORDER BY ?RasterCell
```

SELECT — Select the free dimension(s) and the measurement(s)

WHERE {

Static part of the query.

These patterns represent the ontology scheme and in particular the connection of the classes using objectProperties.

(we can add here all the attributes)

The measurements the user selected

Fixed dimension(s) from the user

Complete !bound variables with "null" so that we can have fixed size tuples in results

}

Statistical Charts

Self-adjusting Query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX tf: <http://data.linkedeodata.eu/talking-fields/ontology#>

SELECT DISTINCT ?RasterCell ?hasVigor ?hasFertValue ?hasCV
WHERE {
    ?Farm rdf:type tf:Farm .
    ?Field rdf:type tf:Field .
    ?RasterCell rdf:type tf:RasterCell .

    ?Field tf:belongsToFarm ?Farm .
    ?RasterCell tf:belongsToField ?Field .

    OPTIONAL { ?RasterCell tf:hasVigor ?hasVigor. } .
    OPTIONAL { ?RasterCell tf:hasFertValue ?hasFertValue. } .
    OPTIONAL { ?RasterCell tf:hasCV ?hasCV. } .

    ?Field rdf:type ?typeField .
    FILTER ( ?Field = <http://data.linkedeodata.eu/
                       talking-fields/000001/Field/id/1154>
            && ?typeField = tf:Field ) .

}
ORDER BY ?RasterCell
```

SELECT — Select the free dimension(s) and the measurement(s)

WHERE {

Static part of the query.

These patterns represent the ontology scheme and in particular the connection of the classes using objectProperties.

(we can add here all the attributes)

The measurements the user selected

Fixed dimension(s) from the user

Complete !bound variables with "null" so that we can have fixed size tuples in results

}

Statistical Charts

Self-adjusting Query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX tf: <http://data.linkedeodata.eu/talking-fields/ontology#>

SELECT DISTINCT ?RasterCell ?hasVigor ?hasFertValue ?hasCV
WHERE {
    ?Farm rdf:type tf:Farm .
    ?Field rdf:type tf:Field .
    ?RasterCell rdf:type tf:RasterCell .

    ?Field tf:belongsToFarm ?Farm .
    ?RasterCell tf:belongsToField ?Field .

    OPTIONAL { ?RasterCell tf:hasVigor ?hasVigor. } .
    OPTIONAL { ?RasterCell tf:hasFertValue ?hasFertValue. } .
    OPTIONAL { ?RasterCell tf:hasCV ?hasCV. } .

    ?Field rdf:type ?typeField .
    FILTER ( ?Field = <http://data.linkedeodata.eu/
                       talking-fields/000001/Field/id/1154>
            && ?typeField = tf:Field ) .

}
ORDER BY ?RasterCell
```

SELECT	Select the free dimension(s) and the measurement(s)
WHERE {	
	Static part of the query. These patterns represent the ontology scheme and in particular the connection of the classes using objectProperties. (we can add here all the attributes)
	The measurements the user selected
	Fixed dimension(s) from the user
	Complete !bound variables with "null" so that we can have fixed size tuples in results
}	

Statistical Charts

Self-adjusting Query

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX tf: <http://data.linkedeodata.eu/talking-fields/ontology#>

SELECT DISTINCT ?RasterCell ?hasVigor ?hasFertValue ?hasCV
WHERE {
    ?Farm rdf:type tf:Farm .
    ?Field rdf:type tf:Field .
    ?RasterCell rdf:type tf:RasterCell .

    ?Field tf:belongsToFarm ?Farm .
    ?RasterCell tf:belongsToField ?Field .

    OPTIONAL { ?RasterCell tf:hasVigor ?hasVigor. } .
    OPTIONAL { ?RasterCell tf:hasFertValue ?hasFertValue. } .
    OPTIONAL { ?RasterCell tf:hasCV ?hasCV. } .

    ?Field rdf:type ?typeField .
    FILTER ( ?Field = <http://data.linkedeodata.eu/
                       talking-fields/000001/Field/id/1154>
           && ?typeField = tf:Field ) .

}
ORDER BY ?RasterCell
```

SELECT — Select the free dimension(s) and the measurement(s)

WHERE {

Static part of the query.

These patterns represent the ontology scheme and in particular the connection of the classes using objectProperties.

(we can add here all the attributes)

The measurements the user selected

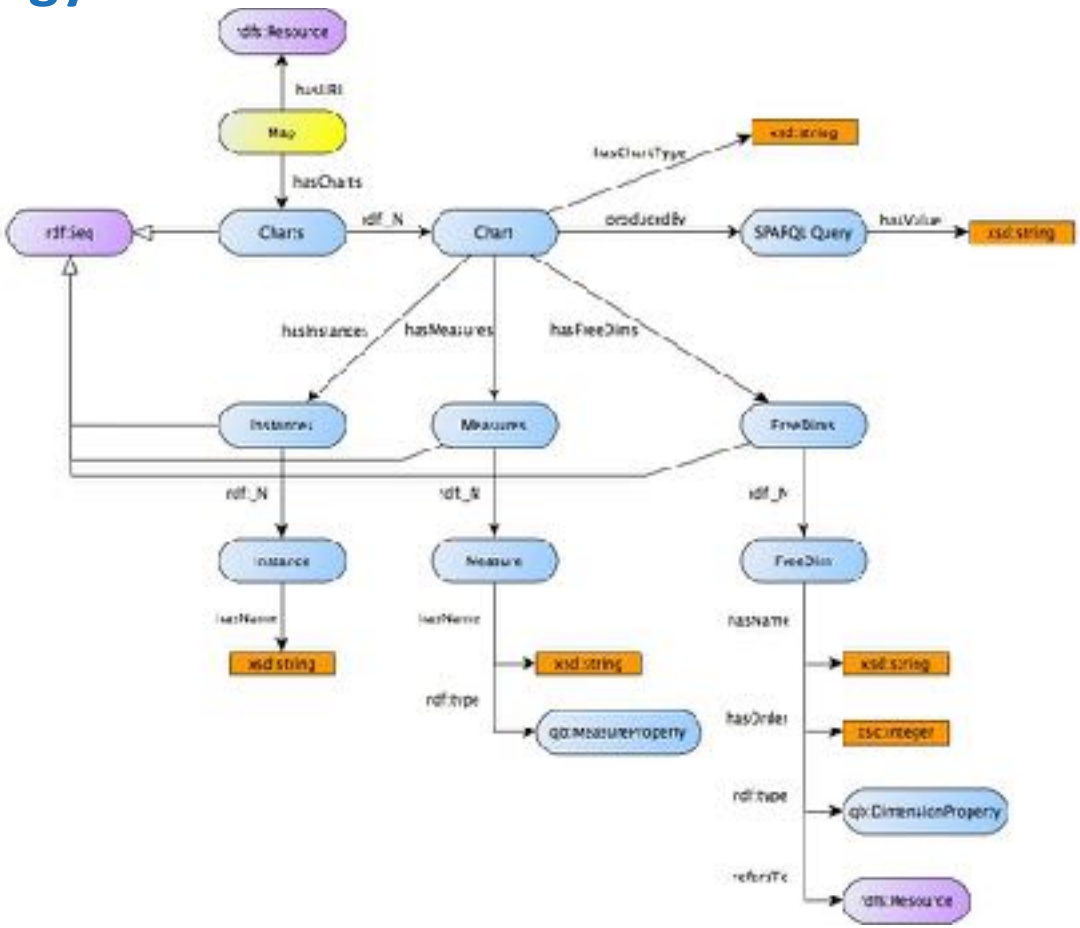
Fixed dimension(s) from the user

Complete !bound variables with "null" so that we can have fixed size tuples in results

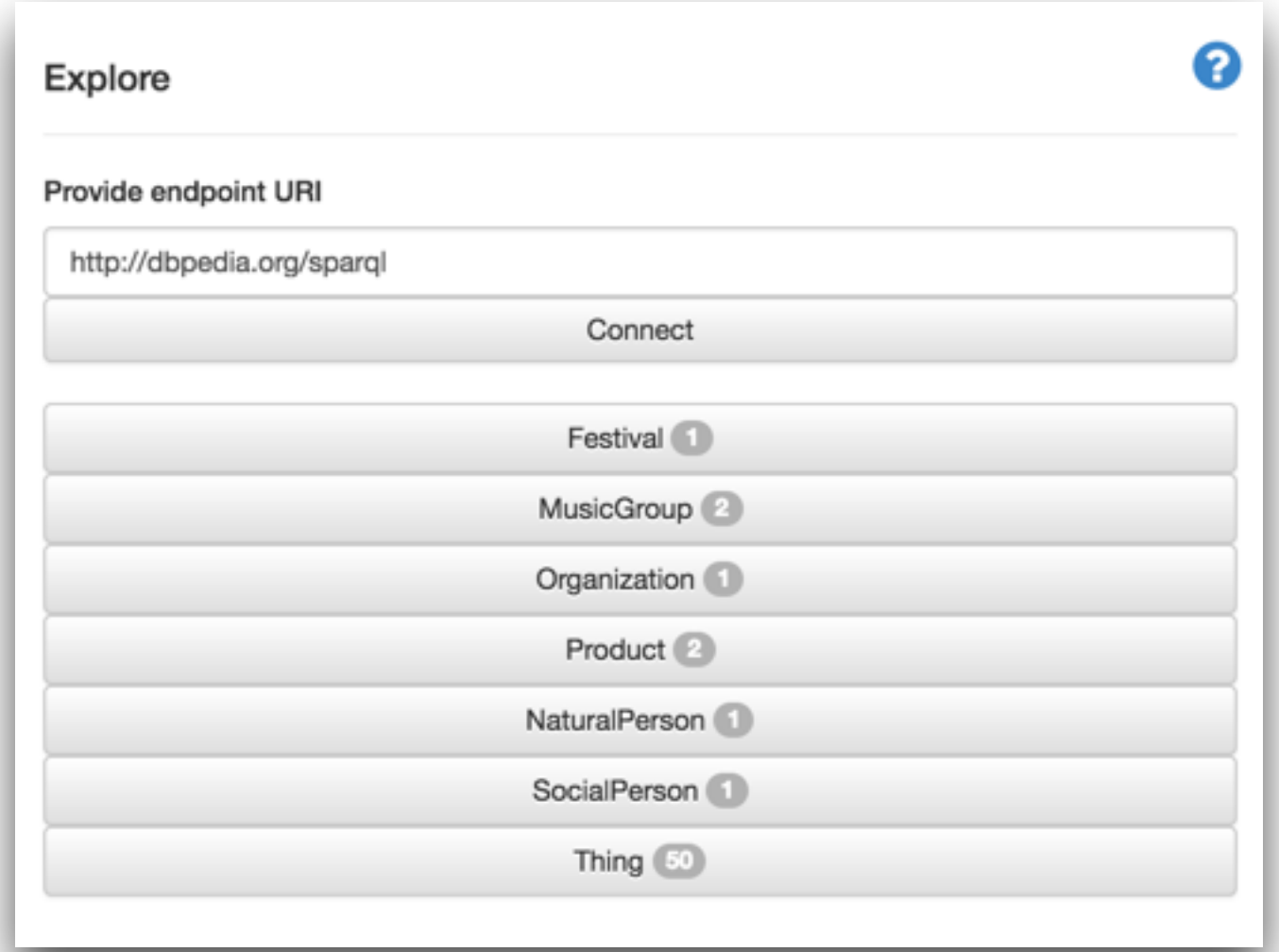
}

Statistical Charts

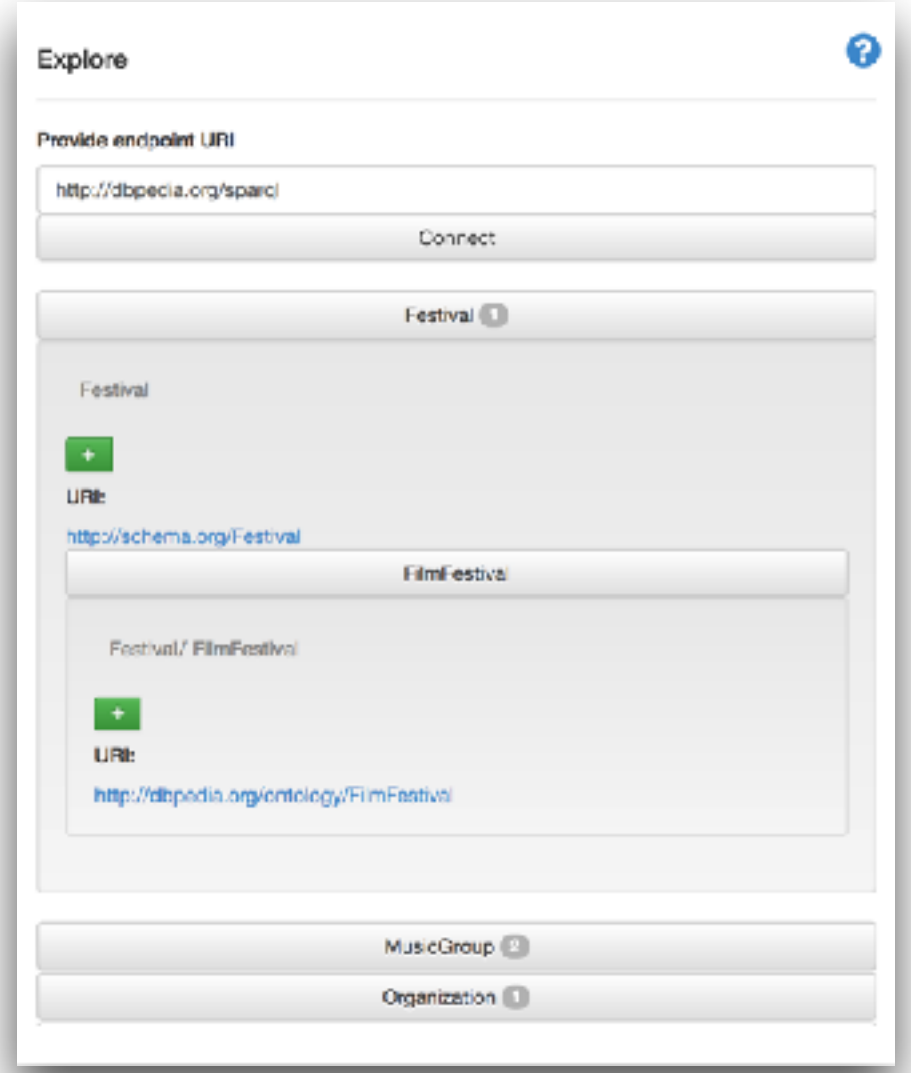
Chart Ontology



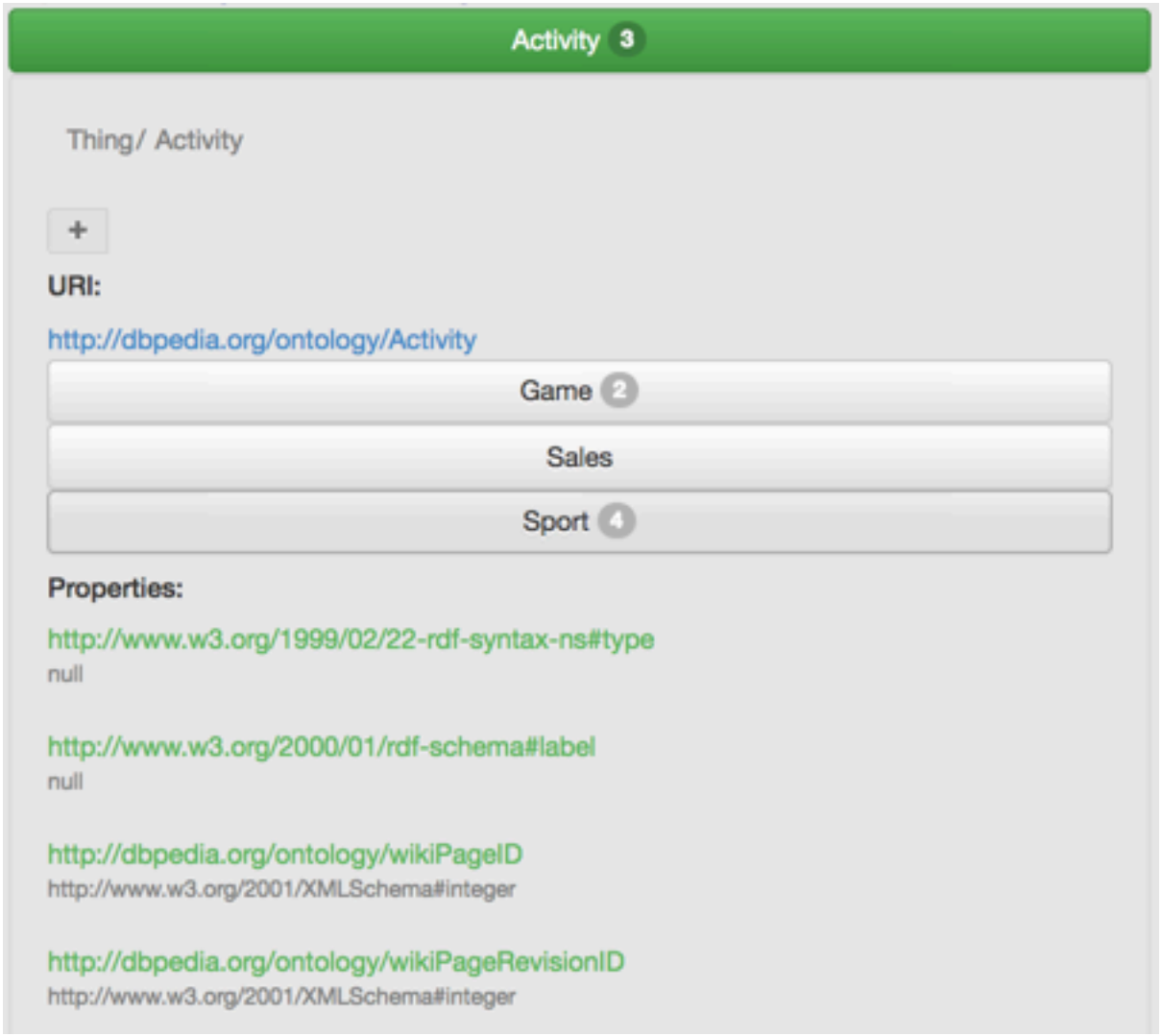
Explore


Explore
Provide endpoint URI
http://dbpedia.org/sparql
Connect
Festival 1
MusicGroup 2
Organization 1
Product 2
NaturalPerson 1
SocialPerson 1
Thing 50

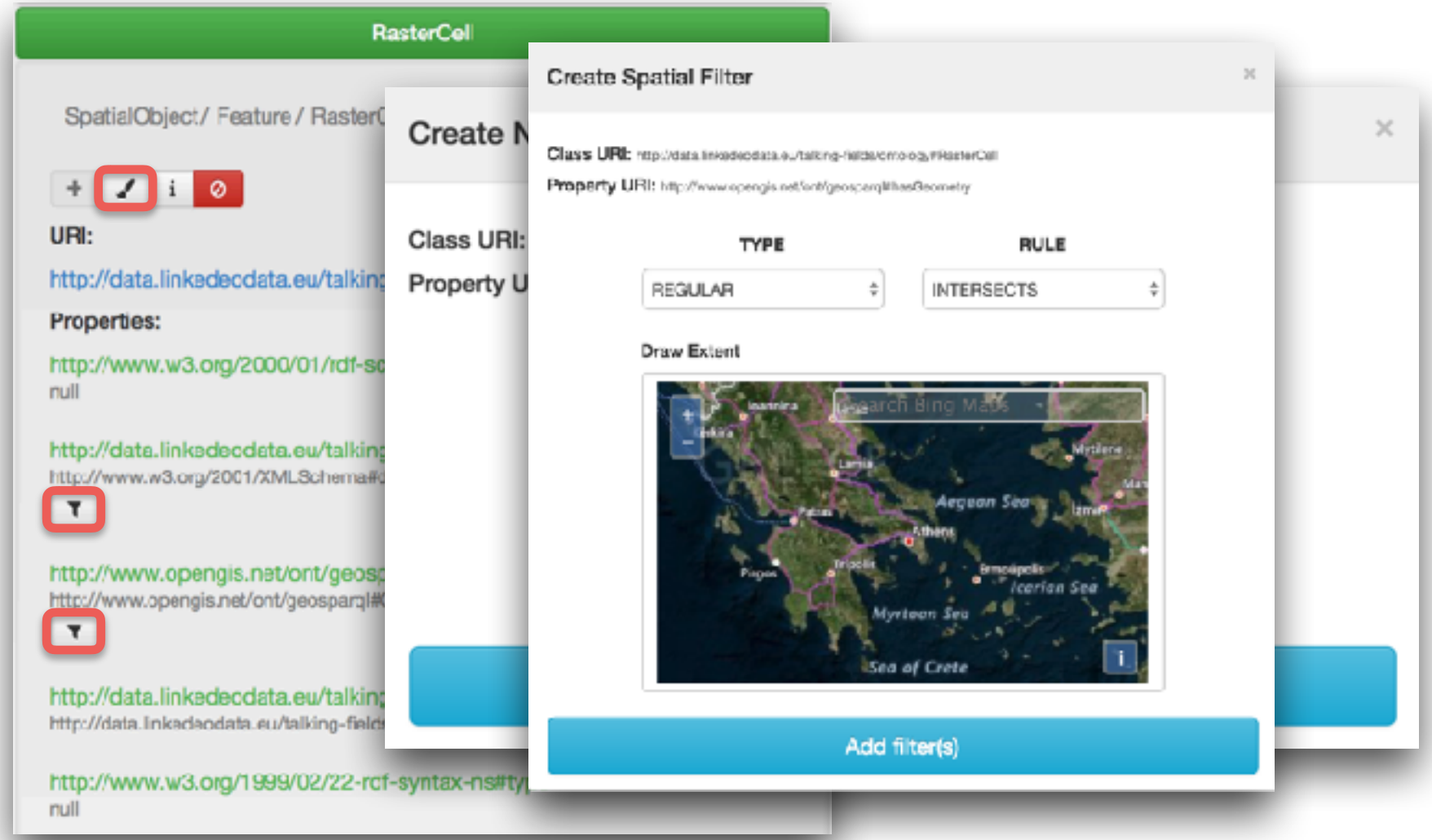
Explore


Explore
Provide endpoint URI
http://dbpedia.org/sparql
Connect
Festival 1
Festival
+
URI:
http://schema.org/Festival
FilmFestival
Festival/ FilmFestival
+
URI:
http://dbpedia.org/ontology/FilmFestival
MusicGroup 2
Organization 1

Explore


Activity 3
Thing/ Activity
+
URI:
http://dbpedia.org/ontology/Activity
Game 2
Sales
Sport 4
Properties:
http://www.w3.org/1999/02/22-rdf-syntax-ns#type
null
http://www.w3.org/2000/01/rdf-schema#label
null
http://dbpedia.org/ontology/wikiPageID
http://www.w3.org/2001/XMLSchema#integer
http://dbpedia.org/ontology/wikiPageRevisionID
http://www.w3.org/2001/XMLSchema#integer

Explore


RasterCell
SpatialObject / Feature / RasterC
URI:
http://data.linkedecdata.eu/talkin
Properties:
http://www.w3.org/2000/01/rdf-sc
null
http://data.linkedecdata.eu/talkin
http://www.w3.org/2001/XMLSchema#
http://www.opengis.net/ont/geosp
http://www.opengis.net/ont/geosparql#
http://data.linkedecdata.eu/talkin
http://data.linkedecdata.eu/talking-field
http://www.w3.org/1999/02/22-rcf-syntax-ns#ty
null
Create N
Class URI:
Property U
Create Spatial Filter
Class URI: http://data.linkedecdata.eu/talking-fields/ontology#RasterCell
Property URI: http://www.opengis.net/ont/geosparql#hasGeometry
TYPE
REGULAR
RULE
INTERSECTS
Draw Extent
Search Bing Maps
Ioannina
Lamia
Aegean Sea
Athens
Patras
Pylos
Tripolis
Kalamata
Icarian Sea
Myrtoan Sea
Sea of Crete
Add filter(s)

Describe

Describe results

About: *http://schema.org/Organization*

Subject	Predicate	Object
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/7-Eleven	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Aardman_Animations	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/About.com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Academy_of_Motion_Picture_Arts_and_Sciences	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Acorn_Computers	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Activision	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Ad_Lib,_Inc.	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Adnams_Brewery	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Aermacchi	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Aero	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Airspeed_Ltd.	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Albanian_Armed_Forces	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Alfa_Romeo	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/Algerian_People's_National_Armed_Forces	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization

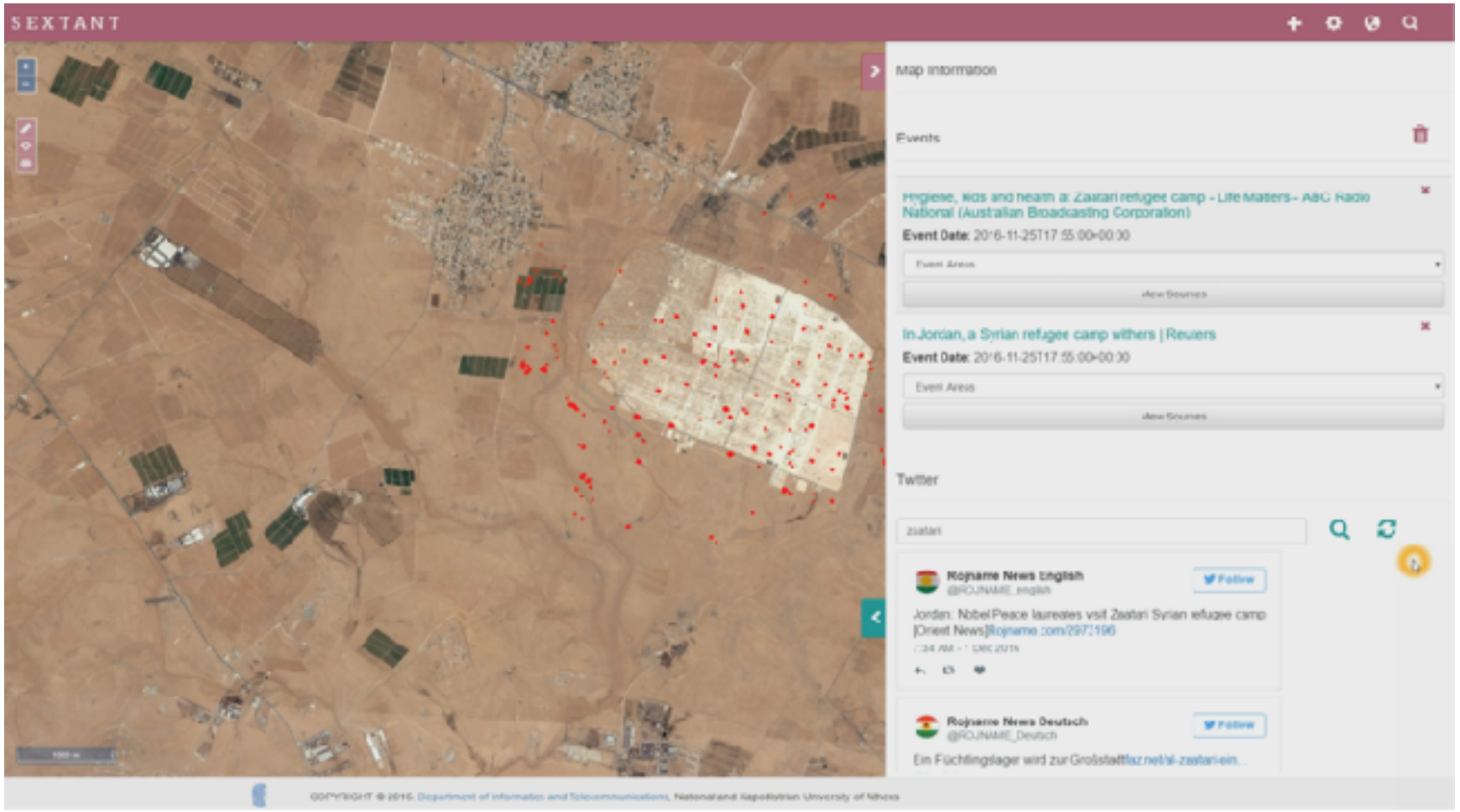
Describe

Describe results

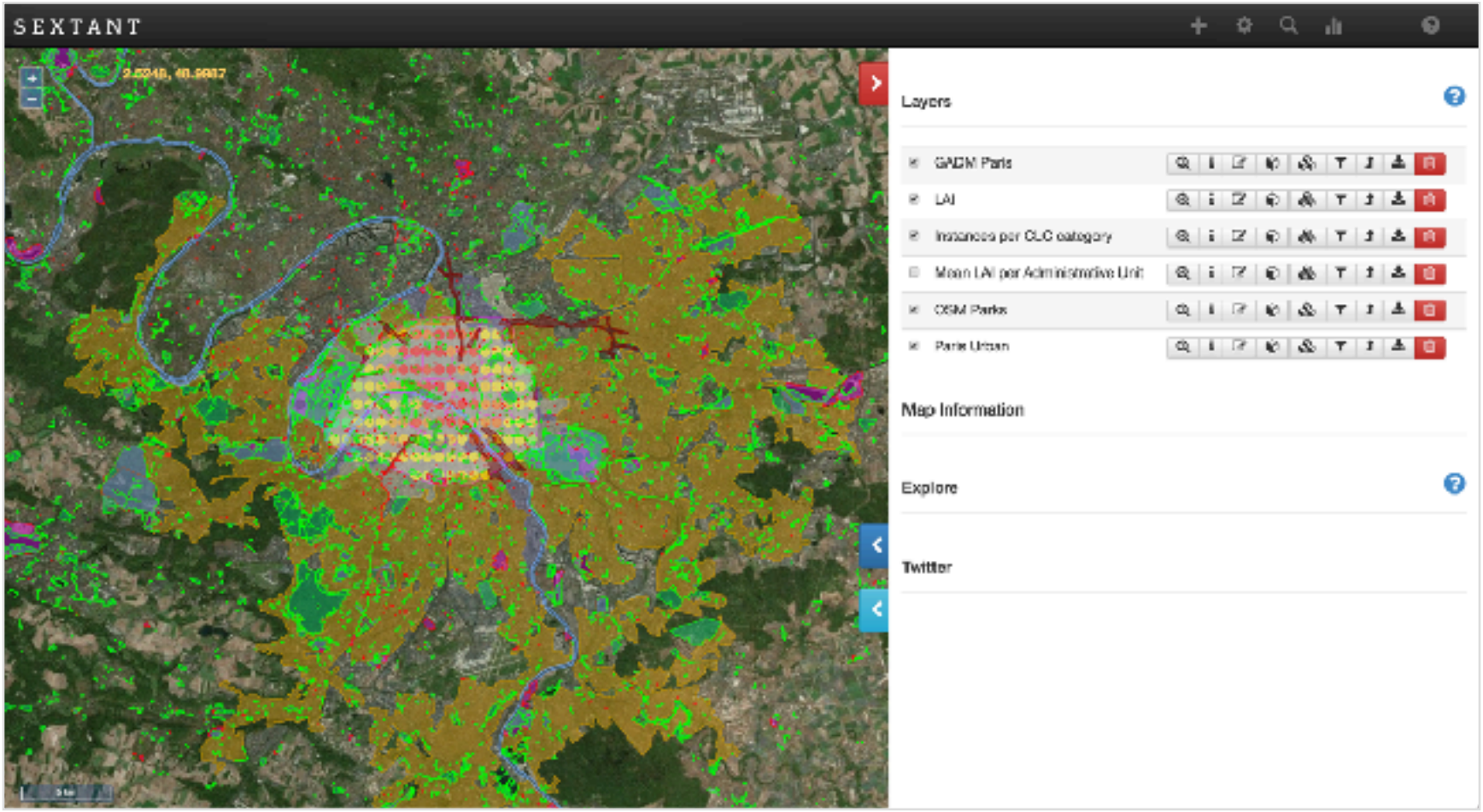
About: http://dbpedia.org/resource/3Com

Subject	Predicate	Object
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/2002/07/owl#Thing
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/ontology/Company
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Agent
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#SocialPerson
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.wikidata.org/entity/Q24229398
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.wikidata.org/entity/Q43229
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/ontology/Agent
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/ontology/Organisation
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://schema.org/Organization
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://umbel.org/umbel/rc/Business
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://umbel.org/umbel/rc/Organization
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/Abstraction100002137
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/Company108058098
http://dbpedia.org/resource/3Com	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://dbpedia.org/class/yago/ElectronicsCompany10000936

Change Detection Service



Green City


SEXTANT
Layers
GADM Paris
LAI
Instances per CLC category
Mean LAI per Administrative Unit
OSM Parks
Paris Urban
Map Information
Explore
Twitter

Open Problems

Open Questions (Theory)

- The **data complexity of query processing for stSPARQL and GeoSPARQL** has not been studied so far.
- We have done so only for the original stSPARQL proposal based on constraints (ESWC 2010).
- Similarly for other interesting problems such as **query containment** etc.
- The **foundations of Ontop-spatial** deserve further study (comparison with the theory of spatial description logics).

Open Questions (Practice)

- How can we built an (even more) scalable geospatial RDF store like Strabon on top of Apache big data technologies?
  - Analyzed the pros and cons of using GeoSpark, SIMBA and SpatialSpark.
    - GeoSpark seems to be the most mature and is continuously been improved by its developers.
- How can we handle 10^{12} triples with 10^{8} polygons? (the volume of geospatial data owned by a national cartographic agency)

Open Questions (cont'd)

- How do we represent and query raster data on the Semantic Web?
  - Raster extension of Ontop-spatial.
  - Array database extension of Ontop-spatial.
  - Work on "Coverages in Linked Data" by the OGC/W3C Spatial Data on the Web working group.

Other relevant research topics of interest

- Natural language query processing for large geospatial knowledge bases (e.g., Yago2).
- Example queries:
  - Find parks in Bonn close to Fraunhofer IAIS.
  - Find rivers that cross cities of Greece and their length is more than 20km.
- Current work in the context of WDAqua (Answering Questions using Web Data, a Marie Sklodowska-Curie Innovative Training Network, http://wdaqua.eu/).

Thanks! Questions?

For more, see the web page of our group http://kr.di.uoa.gr .