Representing and Querying Geospatial Information in the Semantic Web

Kostis Kyzirakos

Dept. of Informatics and Telecommunications,
National and Kapodistrian University of Athens, Greece

Outline

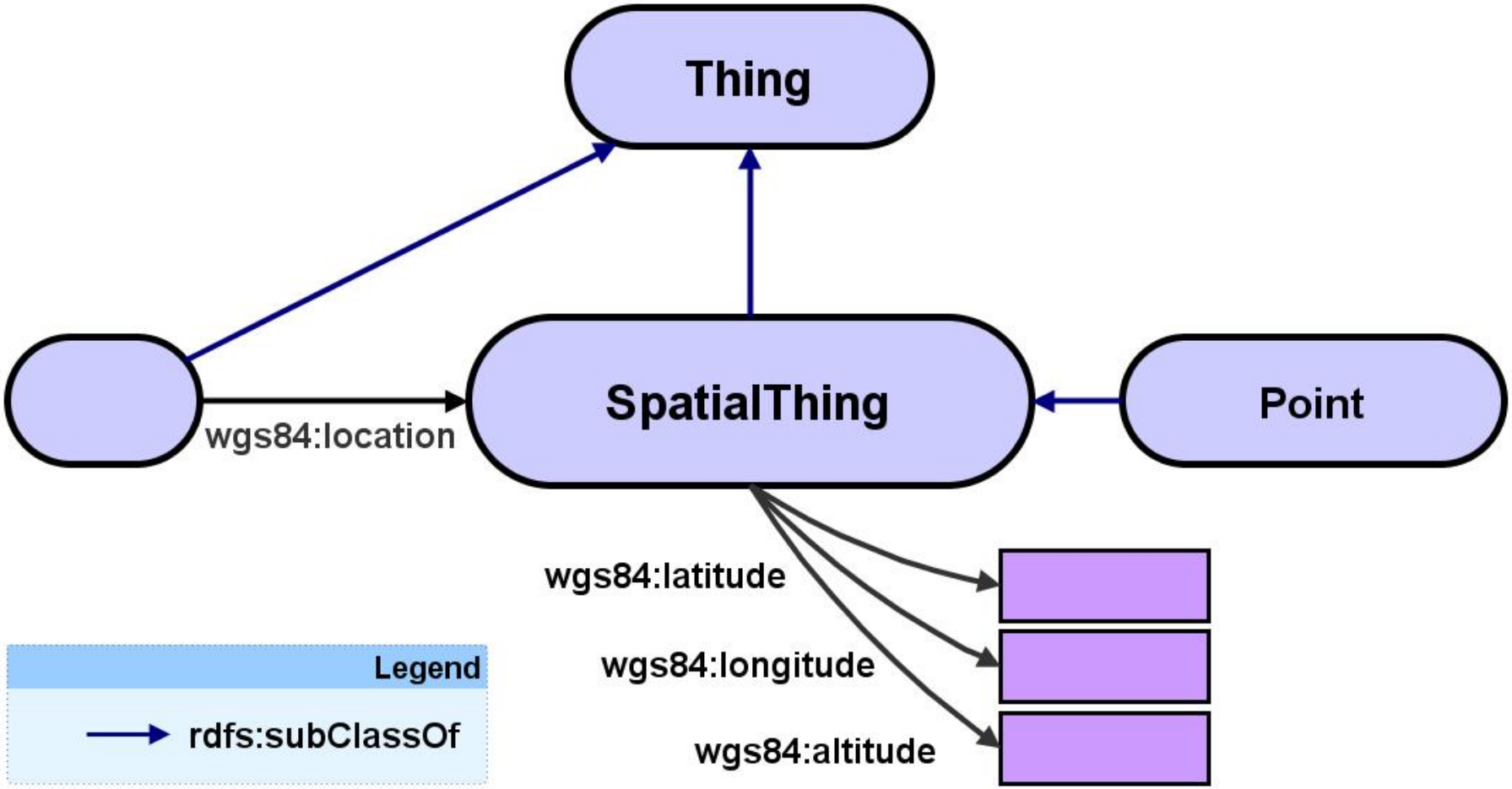
- Introduction
- The data model stRDF
- The query language stSPARQL and a comparison to GeoSPARQL
- The system Strabon for stSPARQL and GeoSPARQL
- Experimental Evaluation
- Real Time Fire Monitoring
- Conclusions

Main idea

How do we **represent** and **query geospatial information** in the Semantic Web?

- Develop appropriate **vocabularies** and **ontologies**
- **Extend RDF** to take into account the **geospatial** dimension
- **Extend SPARQL** to **query** the new kinds of data
- Use **Open Geospatial Consortium (OGC)** and other **geospatial industry standards**

W3C Basic Geo Vocabulary

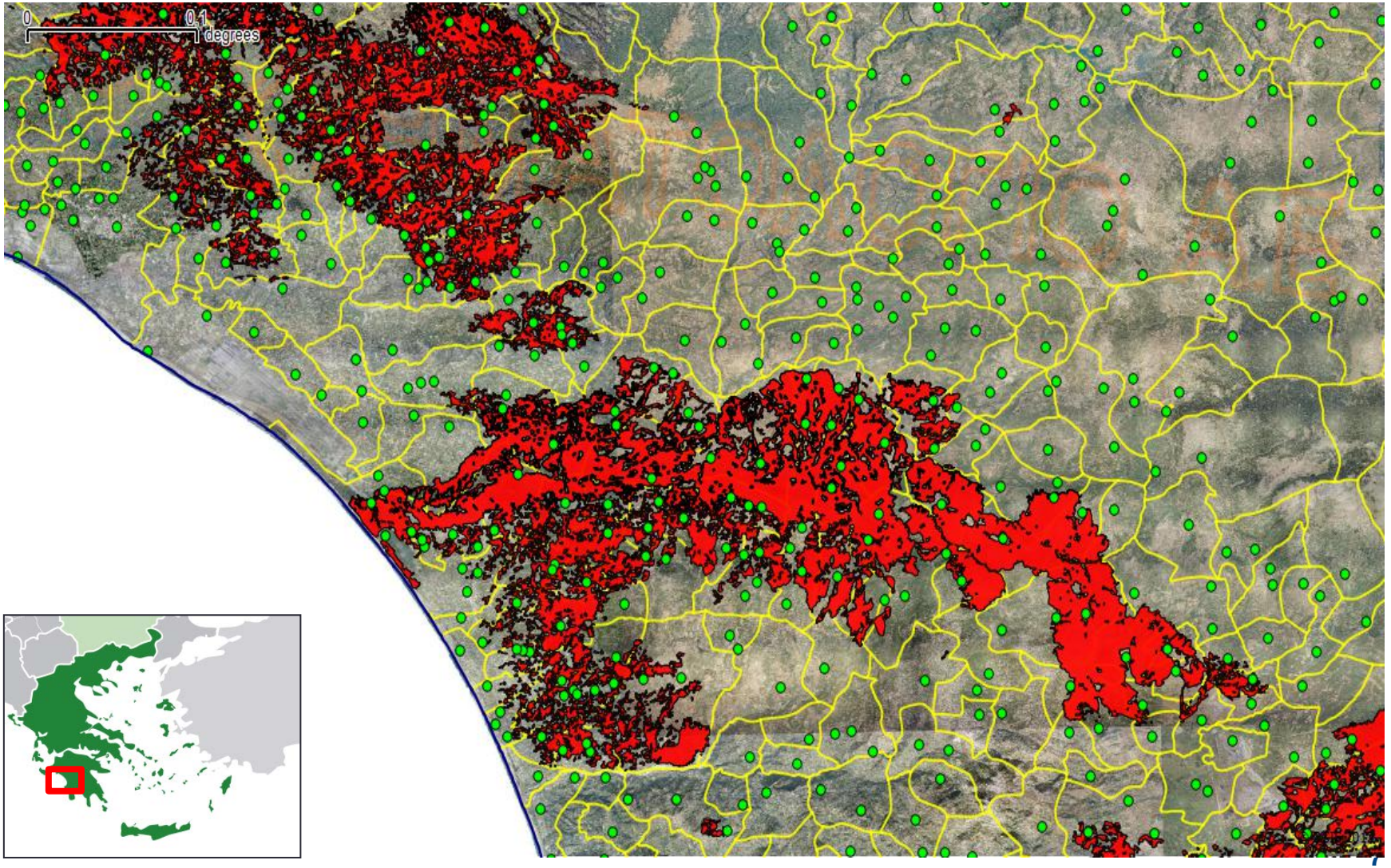


The data model stRDF

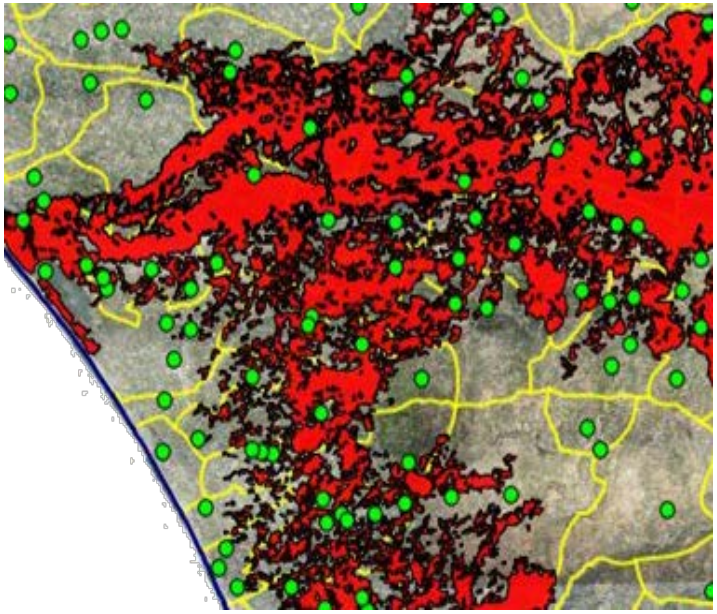
The Data Model stRDF

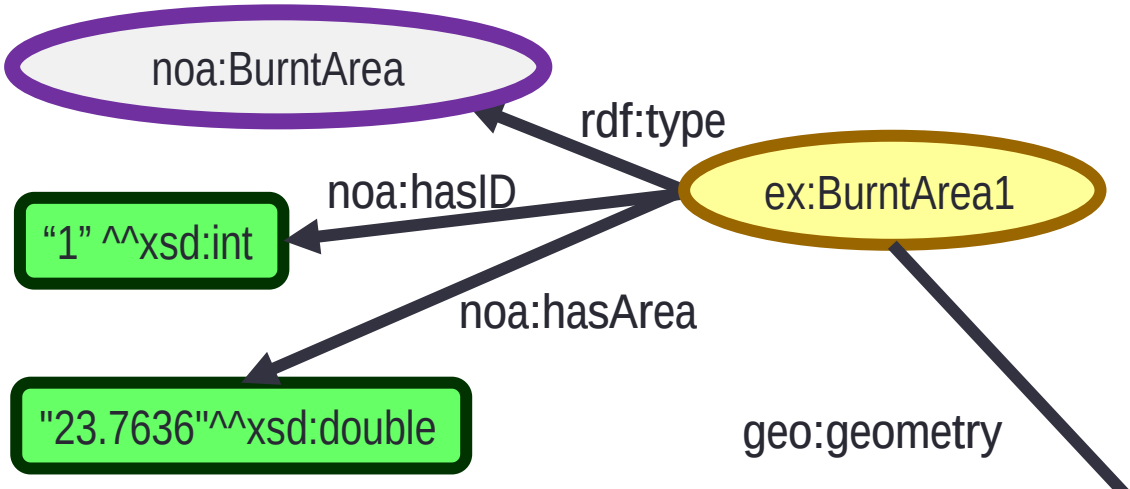
- stRDF extends RDF with: [ESWC'10]
  - **Spatial literals** encoded by Boolean combinations of linear constraints
  - New datatype for spatial literals (strdf:geometry)
  - **Valid time of triples** encoded by Boolean combinations of temporal constraints
- stRDF (most recent version) [ISWC '12]
  - **Spatial literals** encoded in Well-Known Text/GML (OGC standards)
  - Valid time of triples ignored for the time being

Burnt Area Products



Burnt Area Products





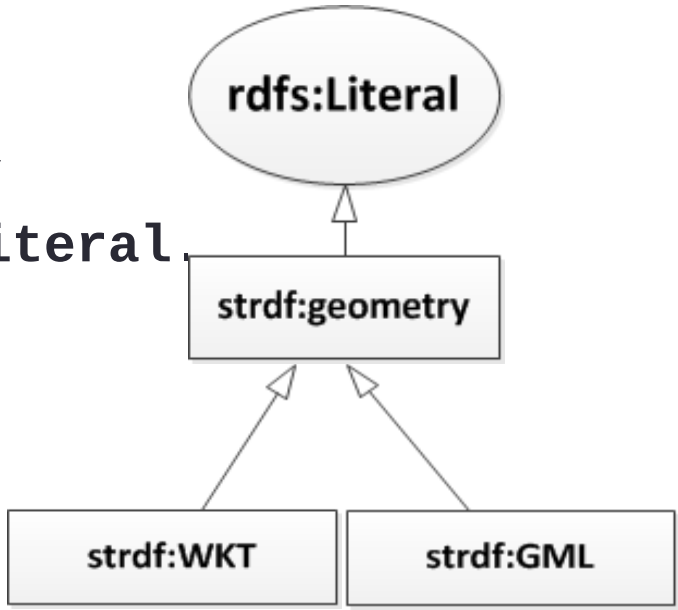
Spatial Literal (OpenGIS Simple Features)

```
"POLYGON(( 38.16 23.7, 38.18 23.7, 38.18
                        23.8, ... 38.16
23.8, 38.16 3.7));
<http://spatialreference.org/ref/epsg/412
1/>"^^strdf:WKT
```

Spatial Data Type Well-Known Text

The stRDF Data Model

```
strdf:geometry rdf:type rdfs:Datatype;
               rdfs:subClassOf rdfs:Literal.
```




```
strdf:WKT   rdf:type rdfs:Datatype;
            rdfs:subClassOf strdf:geometry.

strdf:GML   rdf:type    rdfs:Datatype;
            rdfs:subClassOf   strdf:geometry.
```

Introduction
The data model stRDF
The query language stSPARQL
The system Strabon
Experimental Evaluation
Real Time Fire Monitoring
Conclusions

The query language stSPARQL

stSPARQL: Geospatial SPARQL 1.1

We define a **SPARQL extension function** for each function defined in the **OpenGIS Simple Features Access** standard

- Basic functions
  - Get a property of a geometry (e.g., **`strdf:srid`**)
  - Get the desired representation of a geometry (e.g., **`strdf:AsText`**)
  - Test whether a certain condition holds (e.g., **`strdf:IsEmpty`**, **`strdf:IsSimple`**)
- Functions for **testing topological spatial relationships** (e.g., **`strdf:equals`**, **`strdf:intersects`**)
- **Spatial analysis** functions
  - Construct new geometric objects from existing geometric objects (e.g., **`strdf:buffer`**, **`strdf:intersection`**, **`strdf:convexHull`**)
  - Spatial metric functions (e.g., **`strdf:distance`**, **`strdf:area`**)
- **Spatial aggregate** functions (e.g., **`strdf:union`**, **`strdf:extent`**)

stSPARQL: Geospatial SPARQL 1.1

Select clause

- Construction of new geometries (e.g., **`strdf:buffer(?geo, 0.1)`**)
- Spatial aggregate functions (e.g., **`strdf:union(?geo)`**)
- Metric functions (e.g., **`strdf:area(?geo)`**)

Filter clause

- Functions for testing topological relationships between spatial terms (e.g., **`strdf:contains(?G1, strdf:union(?G2, ?G3))`**)
- Numeric expressions involving spatial metric functions (e.g., **`strdf:area(?G1) ≤ 2*strdf:area(?G2)`**)
- Boolean combinations

Having clause

- Boolean expressions involving spatial aggregate functions and spatial metric functions or functions testing for topological relationships between spatial terms (e.g., **`strdf:area(strdf:union(?geo))>1`**)

Updates

stSPARQL: An example (1/2)

Find coniferous forests that have been affected by fires


```
SELECT ?forest ?burntArea
WHERE {
   ?burntArea   rdf:type noa:BurntArea;
                noa:hasGeometry ?baGeom.
   ?forest      rdf:type noa:Region;
                clc:hasLandCover noa:coniferousForest;
                clc:hasGeometry ?fGeom.
   FILTER(strdf:intersects(?baGeom ?fGeom))
}
```

Spatial Function

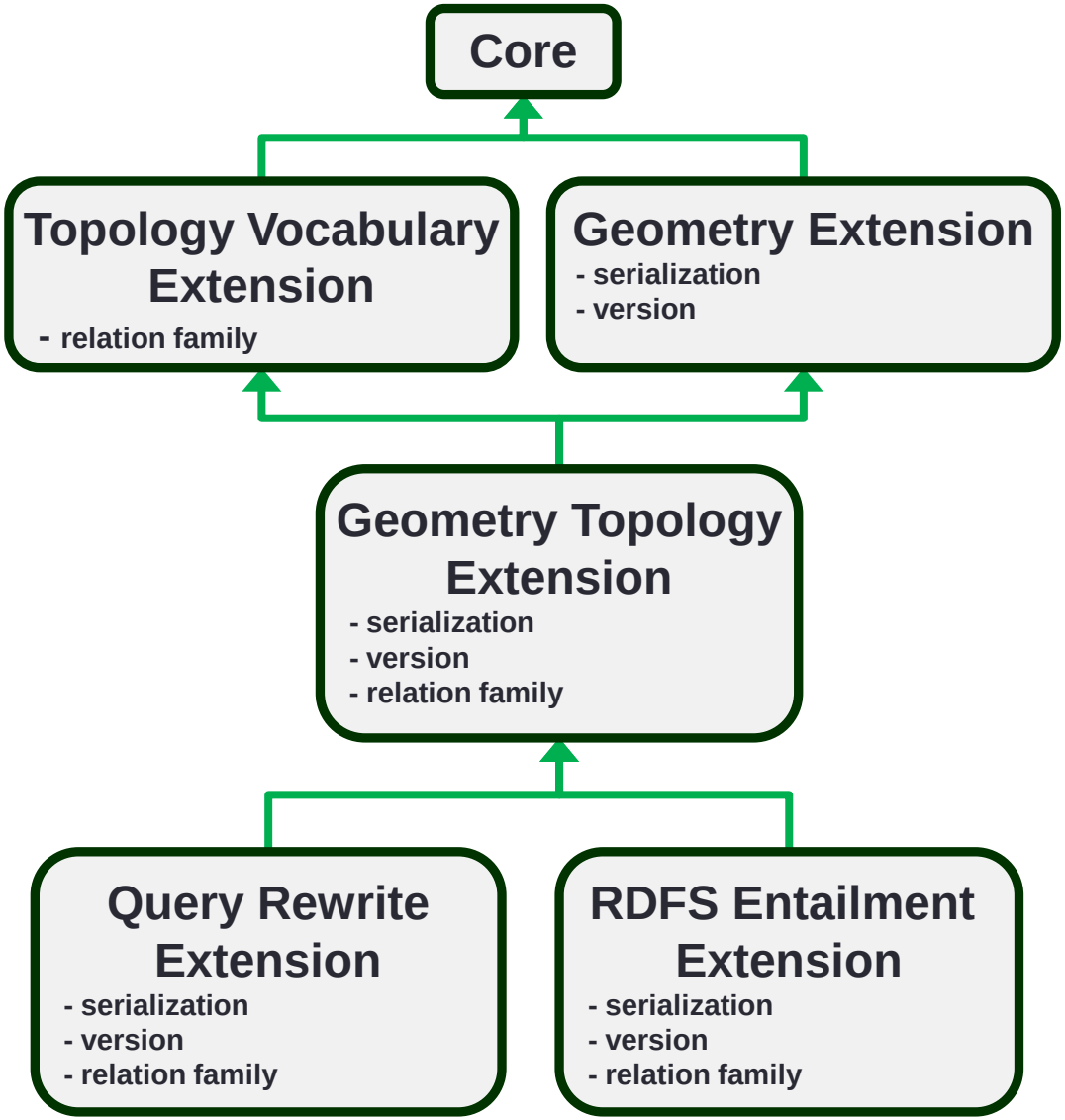
stSPARQL: An example (2/2)

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

Spatial Aggregate

```
SELECT ?burntArea
(strdf:intersection(?baGeom
                    strdf:union(?fGeom
 AS ?burntForest)
WHERE {
   ?burntArea   rdf:type noa:BurntArea;
                noa:hasGeometry ?baGeom.
   ?forest      rdf:type noa:Region;
                clc:hasLandCover noa:coniferousForest;
                clc:hasGeometry ?fGeom.
   FILTER(strdf:intersects(?baGeom ?fGeom))
}
GROUP BY ?burntArea ?baGeom
```

The OGC Standard GeoSPARQL



Parameters

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

stSPARQL and GeoSPARQL

GeoSPARQL is a recently completed OGC standard

stSPARQL and **GeoSPARQL** have been developed independently

Functionalities **similar to stSPARQL**:

- Geometries are represented using **literals** similarly to stSPARQL.
- The same families of **functions** are offered for querying geometries.

Functionalities **beyond stSPARQL**:

- **Topological relations** can now be **asserted** as well so that reasoning and querying on them is possible.

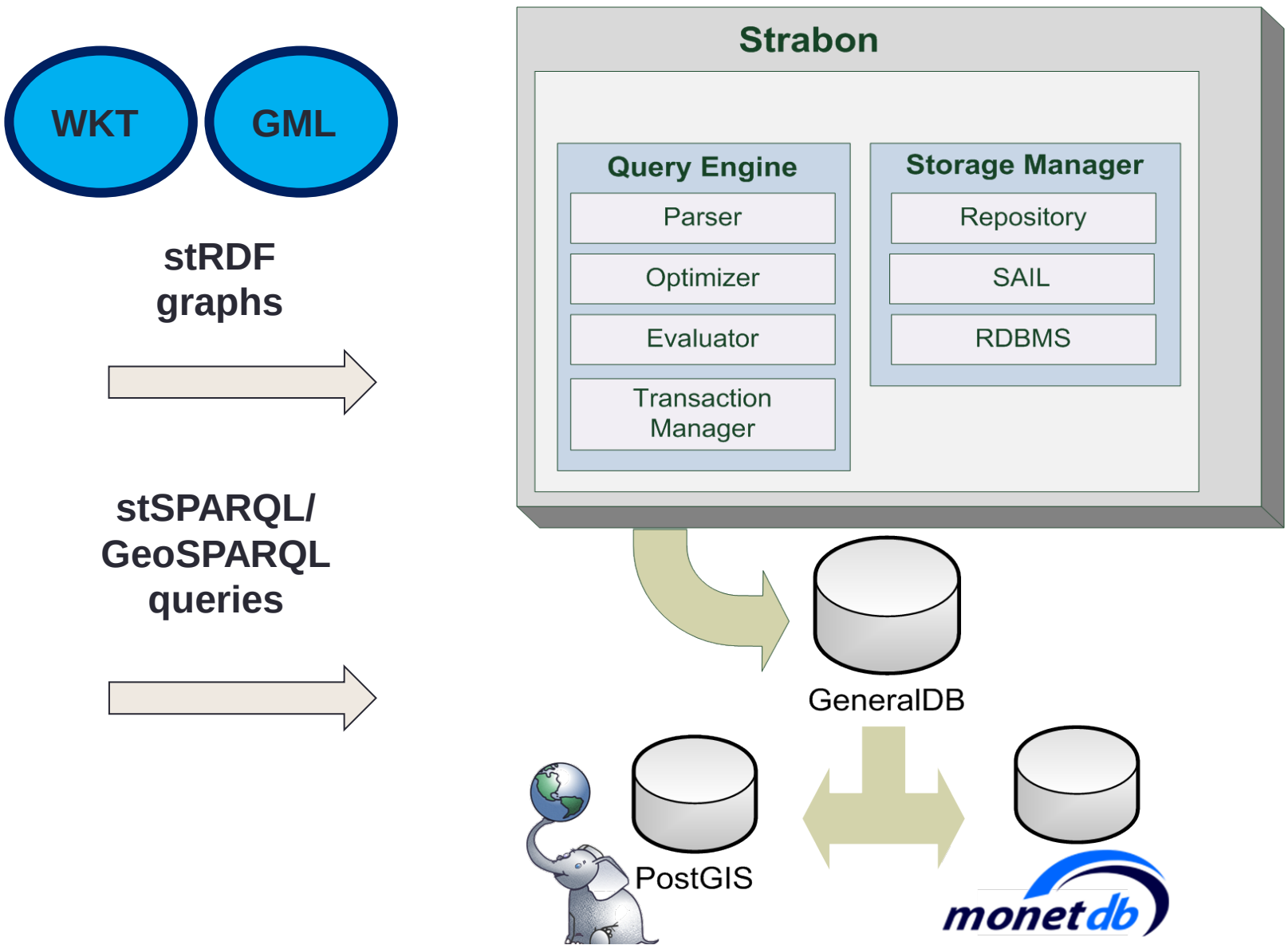
Strabon supports both stSPARQL and GeoSPARQL

Introduction
The data model stRDF
The query language stSPARQL
The system Strabon
Experimental Evaluation
Real Time Fire Monitoring
Conclusions

The system Strabon

strabon.di.uoa.gr

Strabon Architecture



Storage Scheme

type_2

SUBJECT	OBJECT
1	3

measures_4

SUBJECT	OBJECT
1	5
7	8

hasLocation_6

SUBJECT	OBJECT
1	7

hasSpatialExtent_8

SUBJECT	OBJECT
1	9

CATE	OBJECT
	3
	5
	7
	9

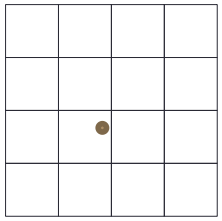
uri_values

ID	VALUE
1	ex:sensor1
2	rdf:type
3	ex:Sensor
4	ex:measures
5	ex:Temperature
6	ex:hasLocation
7	ex:location1
8	strdf:hasSpatialExtent

label_values

ID	VALUE
9	"POINT(37.94194 23.63722)<http://srid.org/ref/epsg/4326/>

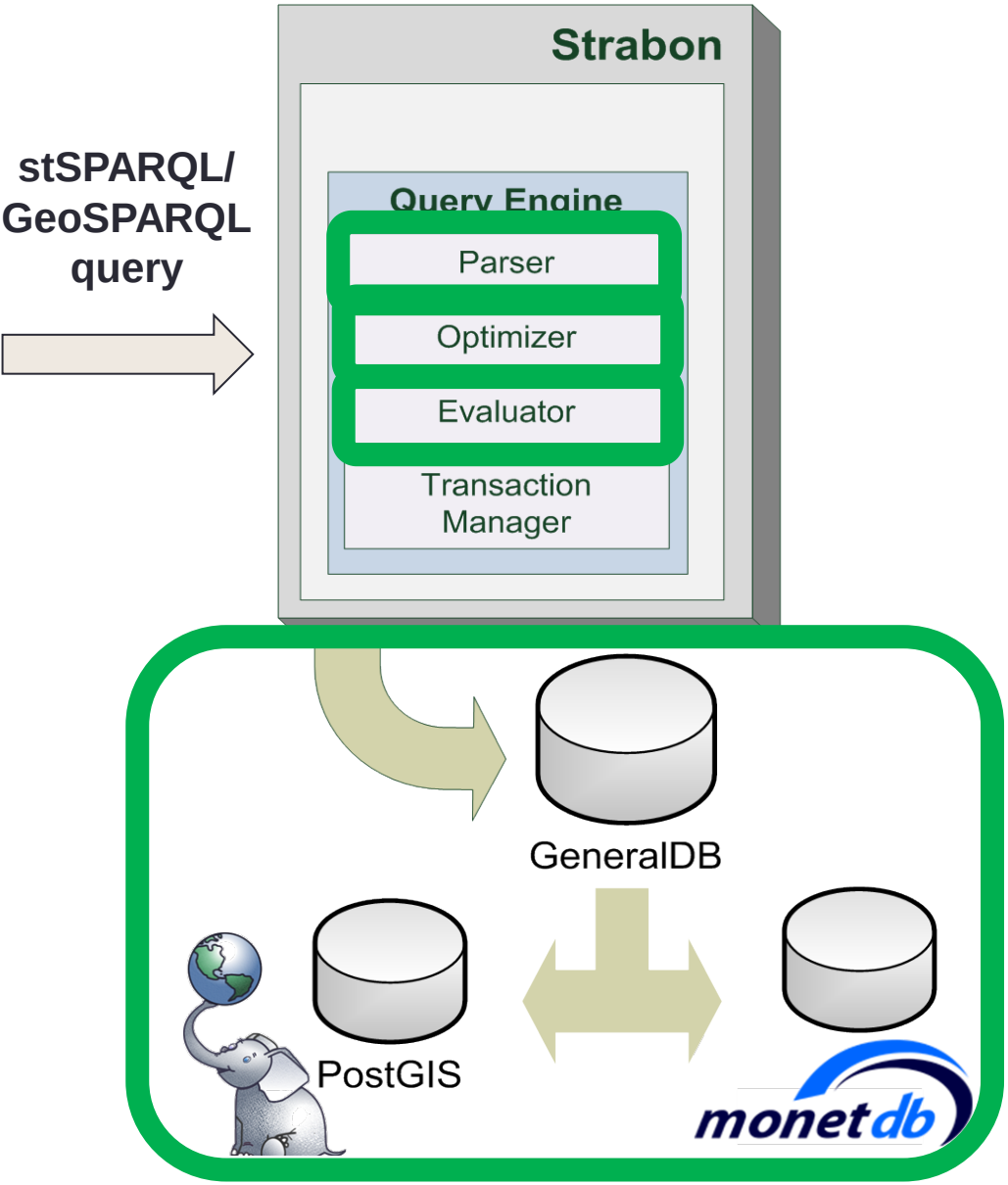
geo_values

ID	VALUE
9	

datatype_values

ID	VALUE
9	ogc:WKT

Query Processing



- Parser generates abstract syntax tree
- Abstract syntax tree mapped to internal algebra of Sesame
- Standard optimizations performed
- Evaluator produces corresponding SQL query
- DBMS evaluates the SQL query
- Post-processing

Query Processing (cont'd)

- **Deviate** from the evaluation strategy of Sesame for SPARQL extension functions
- **Push** the evaluation of extension functions **to underlying DBMS**
  - **Spatial predicates** evaluated by PostGIS
  - **Spatial joins** now affect query plan
    - Avoid Cartesian products
- Results may be returned in well-known industry formats
  - KML/KMZ
  - GeoJSON
  - GML

Introduction

The data model stRDF

The query language stSPARQL

The system Strabon

Experimental Evaluation

Real Time Fire Monitoring

Conclusions

Experimental Evaluation

Experimental Evaluation

- **Goal**: Evaluate the performance of Strabon vs other systems
- Real workload based on geospatial linked datasets
  - **150 million** triples
- Synthetic workload
  - **Half a billion** triples

Strabon vs other systems

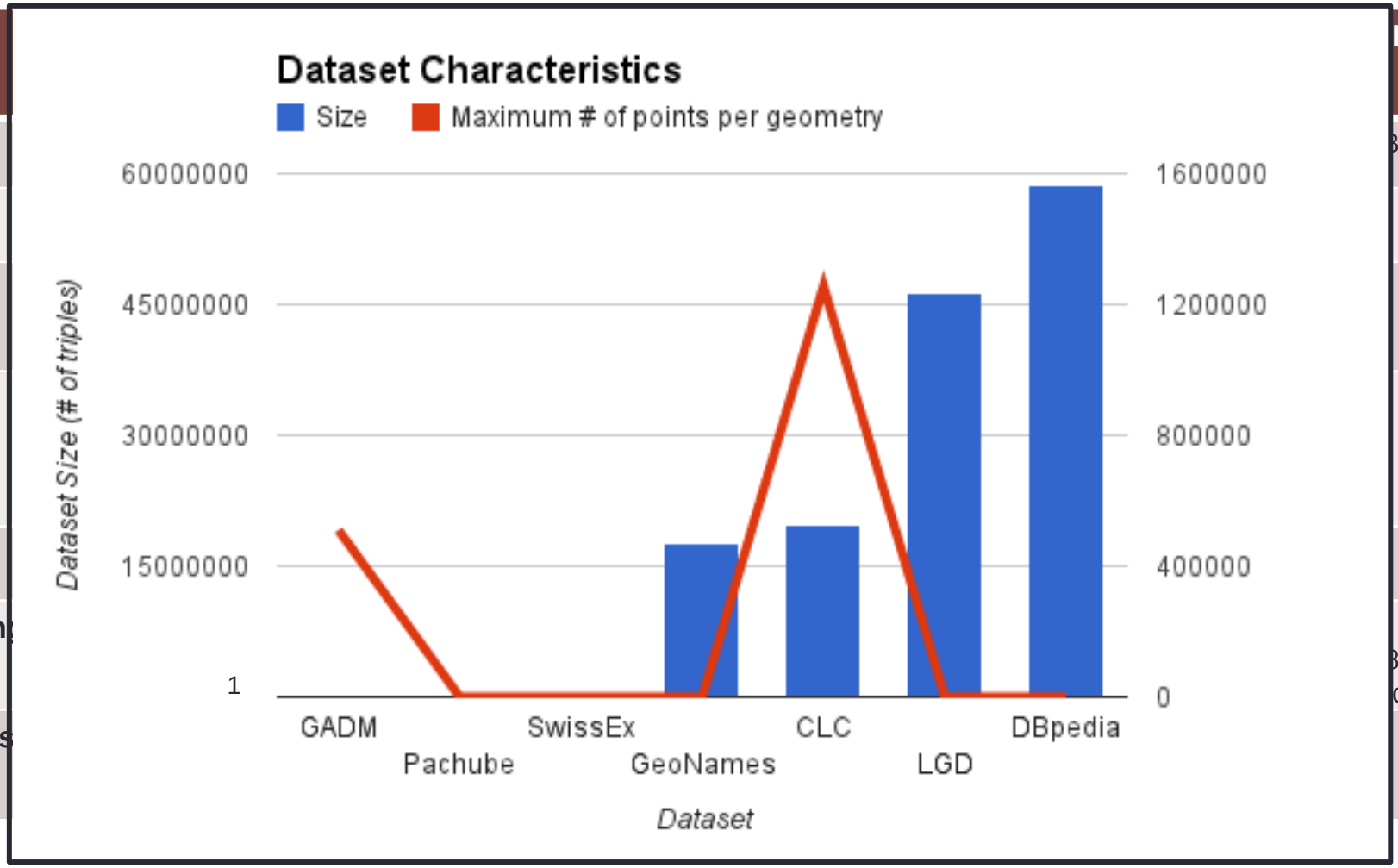
- Strabon over PostgreSQL (Strabon-PG)
- Strabon over System X (Strabon-X)
- Implementation over RDF-3X (Brodt et. al)
- Parliament (BBN Technologies)
- Openlink Virtuoso
- Naive Implementation over Sesame

Real world workload: Data

	DBpedia	GeoNames	LGD	Pachube	SwissEx	CLC	GADM
Size	7.1GB	2.1GB	6.6GB	828KB	33MB	14GB	146MB
Triples	58,722,893	17,688,602	46,296,978	6,333	277,919	19,711,926	255
Spatial Terms	386,205	1,262,356	5,414,032	101	687	2,190,214	51
Distinct Spatial Terms	375,087	1,099,964	5,035,981	70	623	2,190,214	51
Points	375,087	1,099,964	3,205,015	70	623	-	-
Linestrings	-	-	353,714	-	-	-	- 79,831 vertices
Polygons	-	-	1,704,650	-	-	2,190,214	51 ( )

Real world workload: Data

	DM
Size	
Triples	
Spatial Terms	
Distinct Spatial Terms	
Points	
Linestrin	331 ces
Polygons	)



Real world workload: Queries

	Commonly Used	Spatial Selection	Spatial Join
Query 1	X		
Query 2	X		
Query 3		X	
Query 4		X	
Query 5			X
Query 6	X		X
Query 7	X		X
Query 8			X

Queries with Spatial Joins	Point	Line	Polygon
Query 5	X X		X X
Query 6	X	X	X
Query 7			X
Query 8	X	X	X

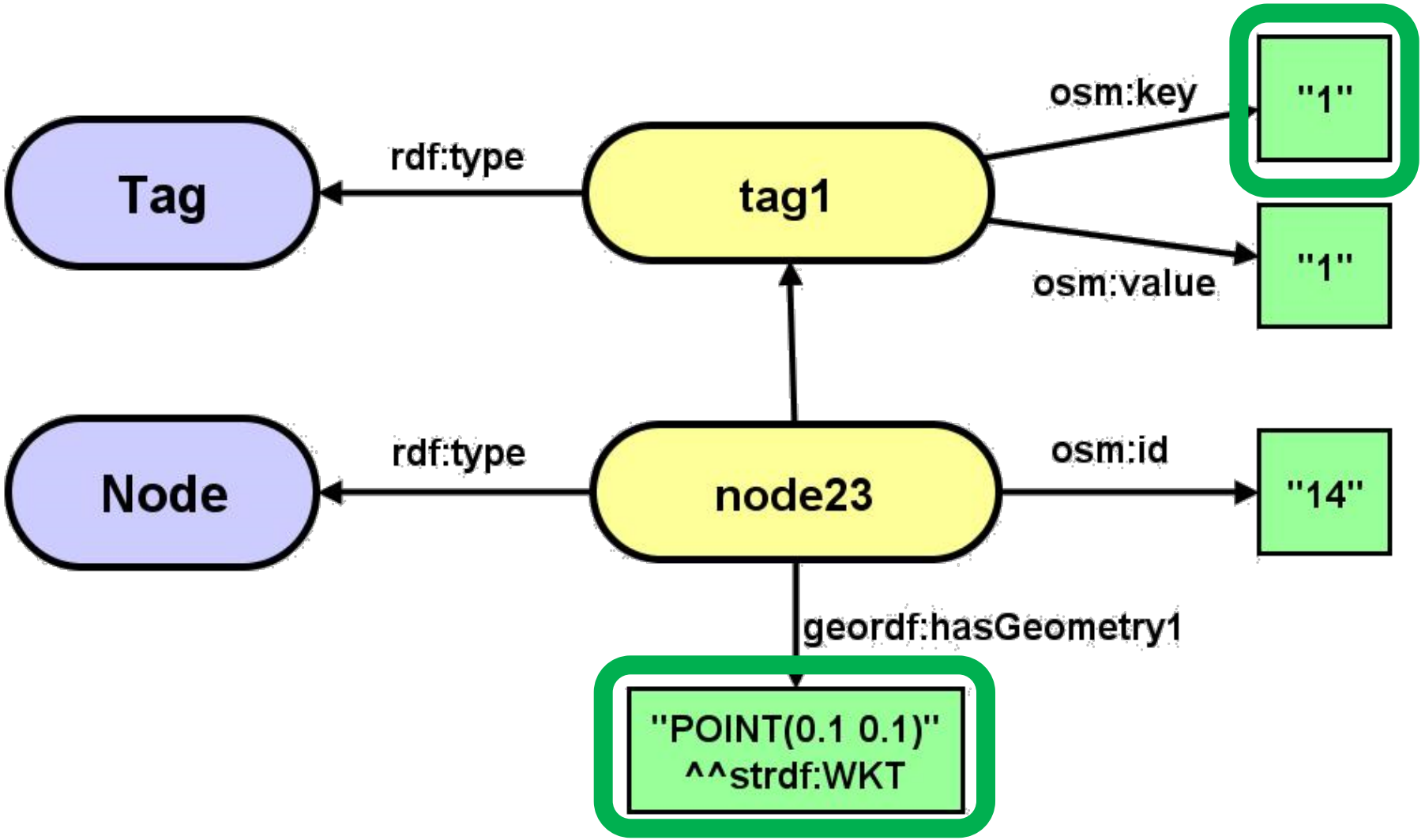
Real world workload: Results

Cache State	System	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Cold (sec.)	Naive	0.08	1.65	>8h	28.88	89	170	844	1.699
	Strabon PG	2.01	6.79	41.39	10.11	78.69	60.25	9.23	702.55
	Strabon X	1.74	3.05	1623	46.52	12.57	2409	>8h	57.83
	Parliament	2.12	6.46	>8h	229.72	1130	872	3627	3786
Warm (sec.)	Naïve	0.01	0.03	>8h	0.79	43.07	88	708	1712
	Strabon PG	0.01	0.81	0.96	1.66	38.74	1.22	2.92	648.1
	Strabon X	0.01	0.26	1604.9	35.59	0.18	3196	>8h	44.72
	Parliament	0.01	0.04	>8h	10.91	358.92	483.29	2771	3502

Synthetic Workload

- Workload based on a synthetic dataset
  - Dataset based on OpenStreetMaps
  - **10 million** triples (**2 GB**) up to **half a billion** triples (**50GB**)
    - Implemented custom bulk loader
  - Triples with **spatial literals**: **1** up to **46 million** triples
- Response time of queries with various **thematic** and **spatial selectivities**

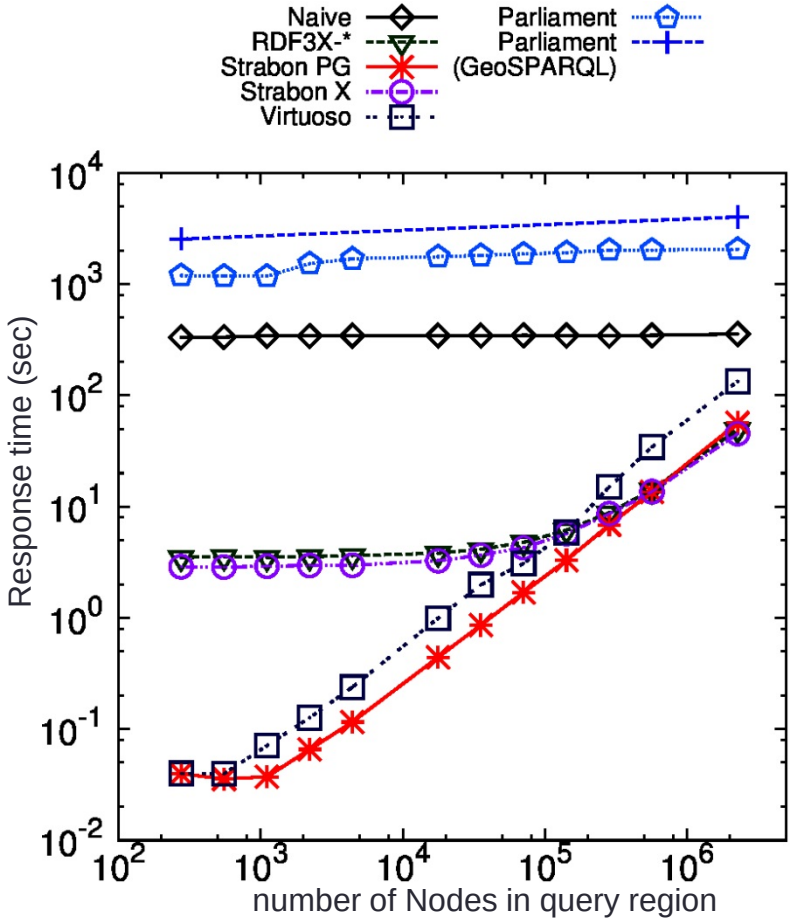
Synthetic Workload: Sample Data



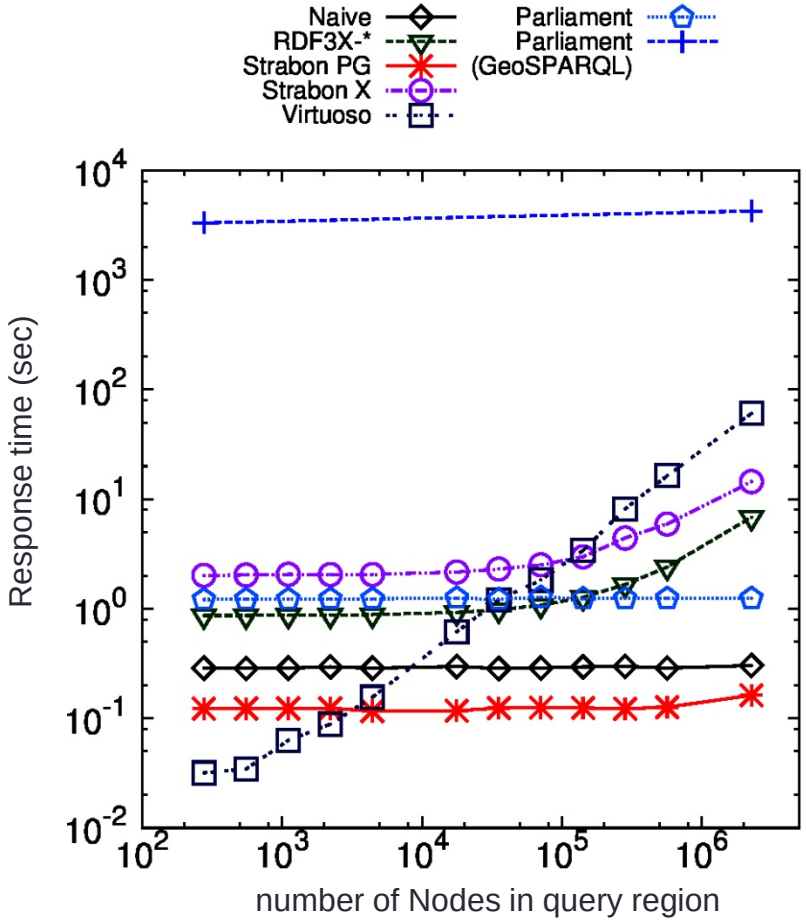
Synthetic Workload: Sample stSPARQL Query

```
SELECT *
WHERE {
    ?tag geordf:key "1" .
    ?node geordf:hasTag ?tag .
    ?node geo:hasGeography1 ?geo .
FILTER (strdf:inside(?geo,
     "POLYGON((-1 -1, 0.056568542 -1,
    0.056568542 0.056568542,
    -1 0.056568542, -1 -1))"^^strdf:WKT
      ))
}
```

Real-world Workload: 100 million triples – warm caches

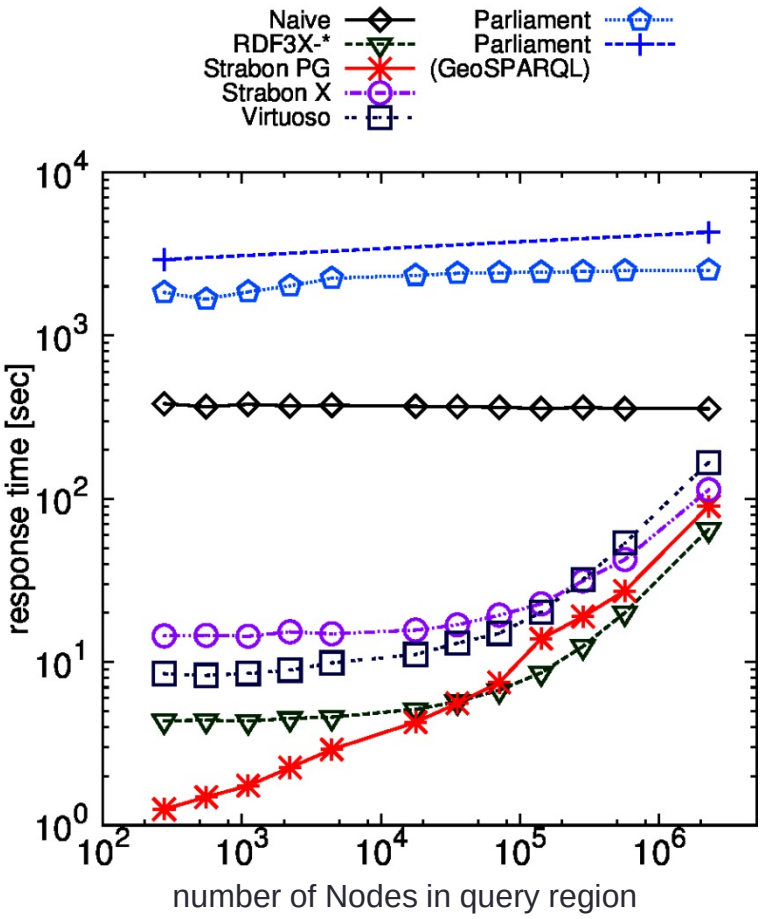


Tag 1

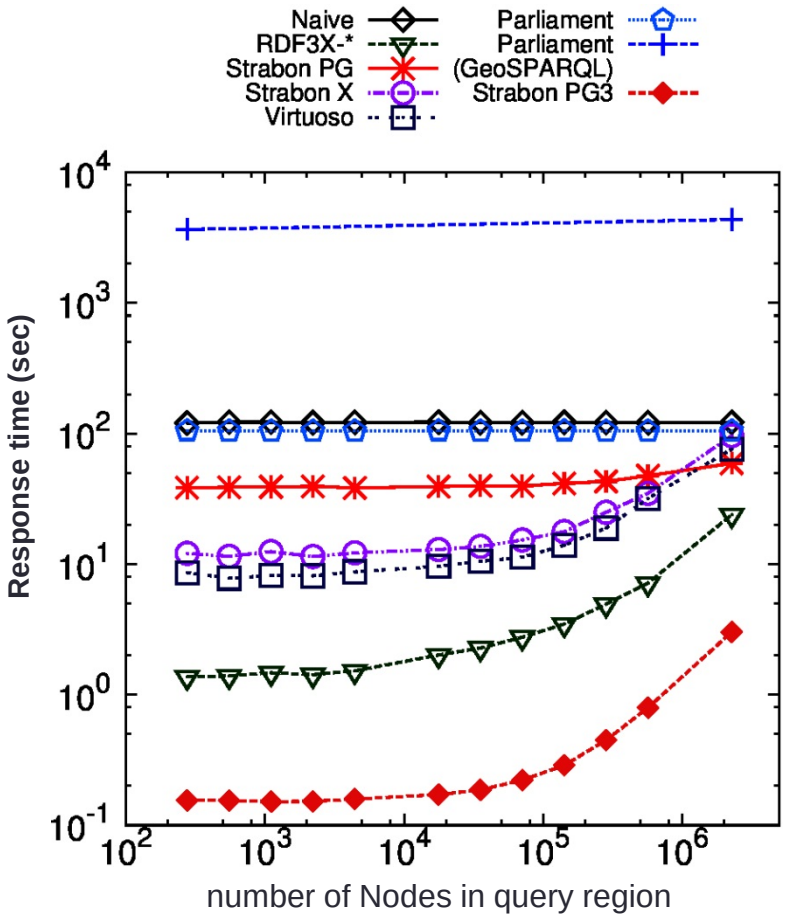


Tag 1024

Real-world Workload: 100 million triples – cold caches

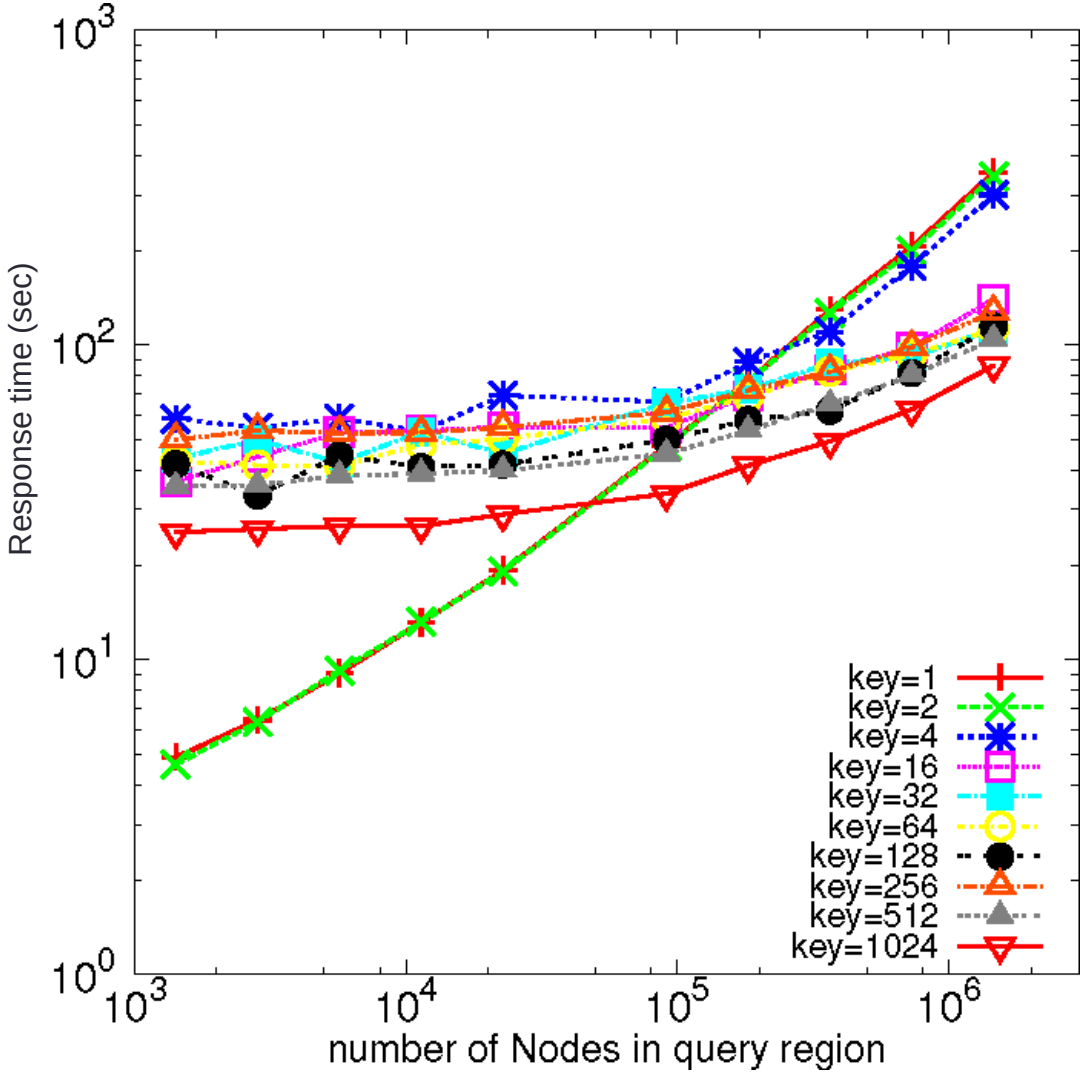


Tag 1

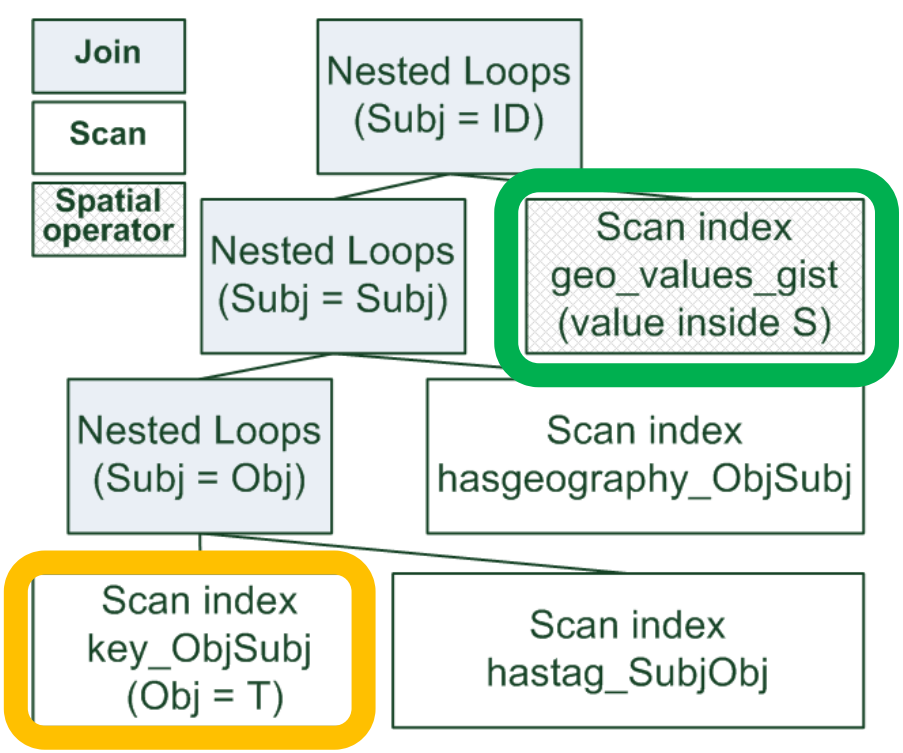


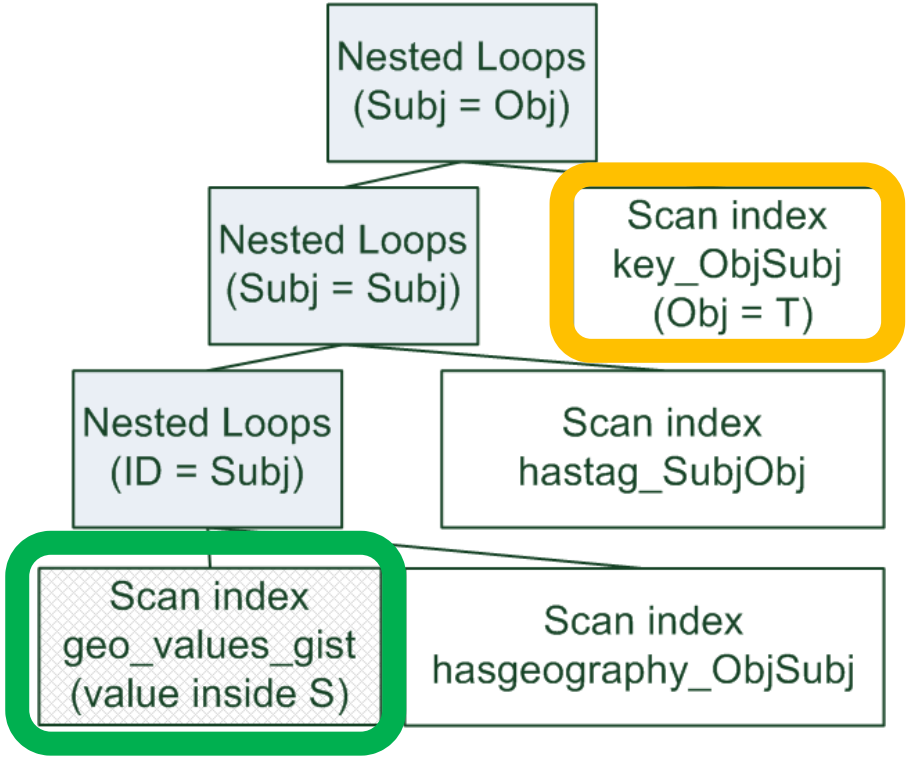
Tag 1024

500 million triples – tags comparison

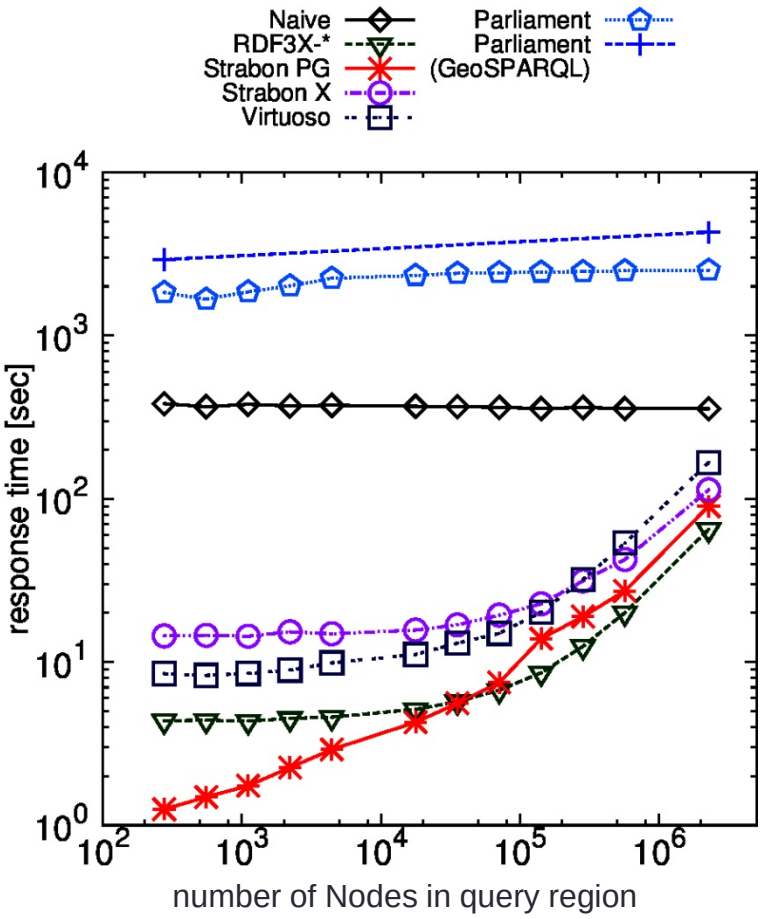


Real-world Workload: Query Plans

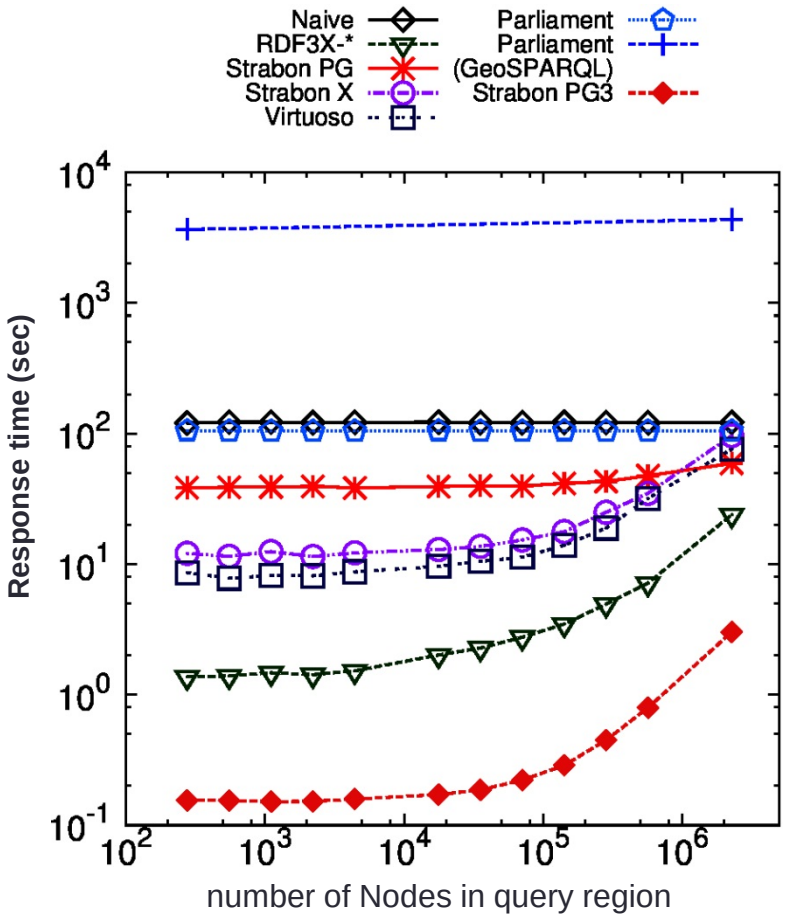




Real-world Workload: 100 million triples – cold caches



Tag 1



Tag 1024

Findings

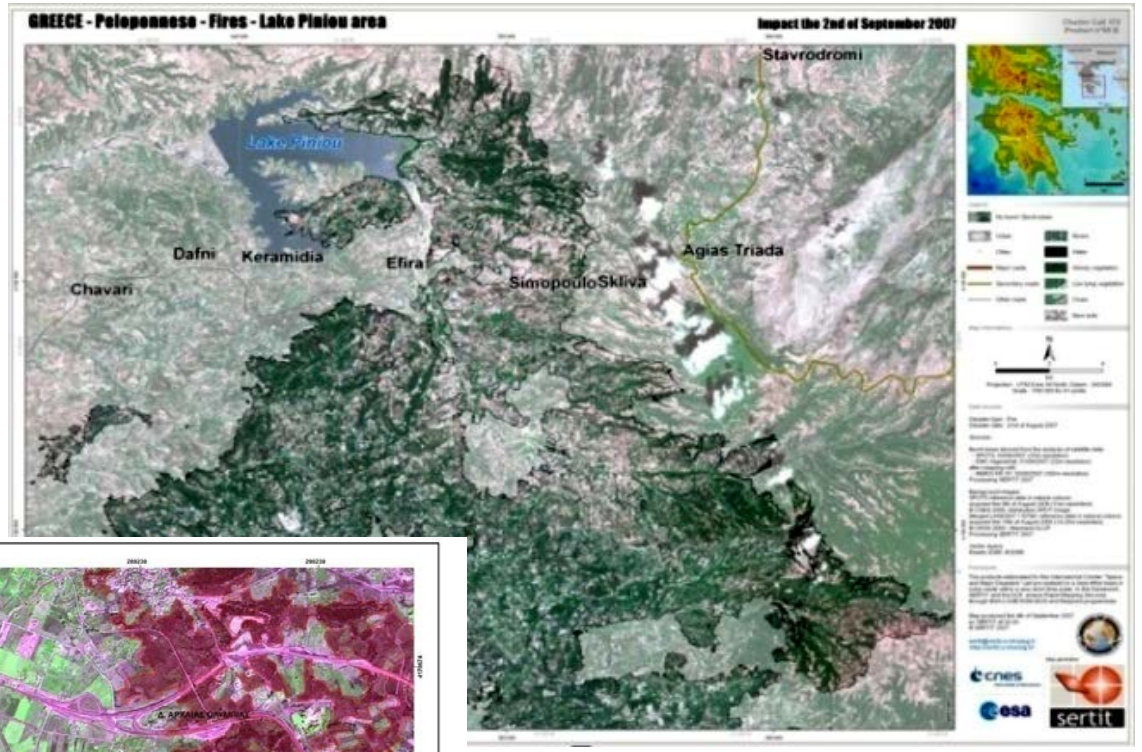
- Strabon over PostgreSQL outperforms other systems in case of warm caches
- Results in case of cold caches mixed
- PostreSQL optimizer needs to take into account spatial selectivity
  - PostGIS 2.0 moves towards it

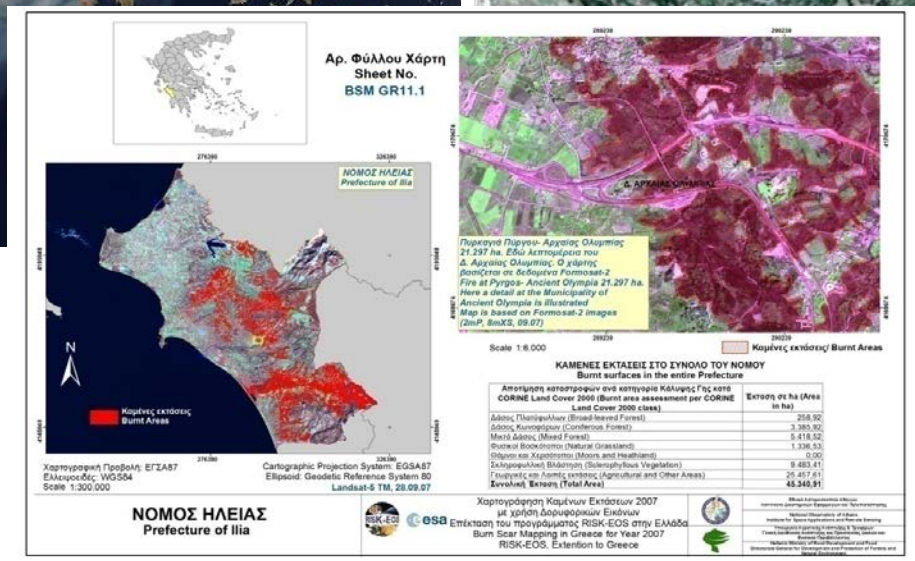
System	Language	Index	Geometries	CRS support	Geospatial Function Support
Strabon	stSPARQL/ GeoSPARQL*	R-tree-over-GiST	WKT / GML support	Yes	• OGC-SFA • Egenhofer • RCC-8
Parliament	GeoSPARQL*	R-Tree	WKT / GML support	Yes	•OGC-SFA •Egenhofer •RCC-8
Oracle	GeoSPARQL?	R-Tree, Quadtree	WKT / GML support	Yes	•OGC-SFA •Egenhofer •RCC-8
Brodt et al. (RDF-3X)	SPARQL	R-Tree	WKT support	No	OGC-SFA
Perry	SPARQL-ST	R-Tree	GeoRSS GML	Yes	RCC-8
AllegroGraph	Extended SPARQL	Distribution sweeping technique	2D point geometries	Partial	•Buffer •Bounding Box •Distance
OWLIM	Extended SPARQL	Custom	2D point geometries	No	•Point-in-polygon •Buffer •Distance
Virtuoso	SPARQL	R-Tree	2D point geometries	Yes	SQL/MM (subset)
uSeekM	SPARQL	R-tree-over GiST	WKT support	No	OGC-SFA

Real Time Fire Monitoring

papos.space.noa.gr/fend_static

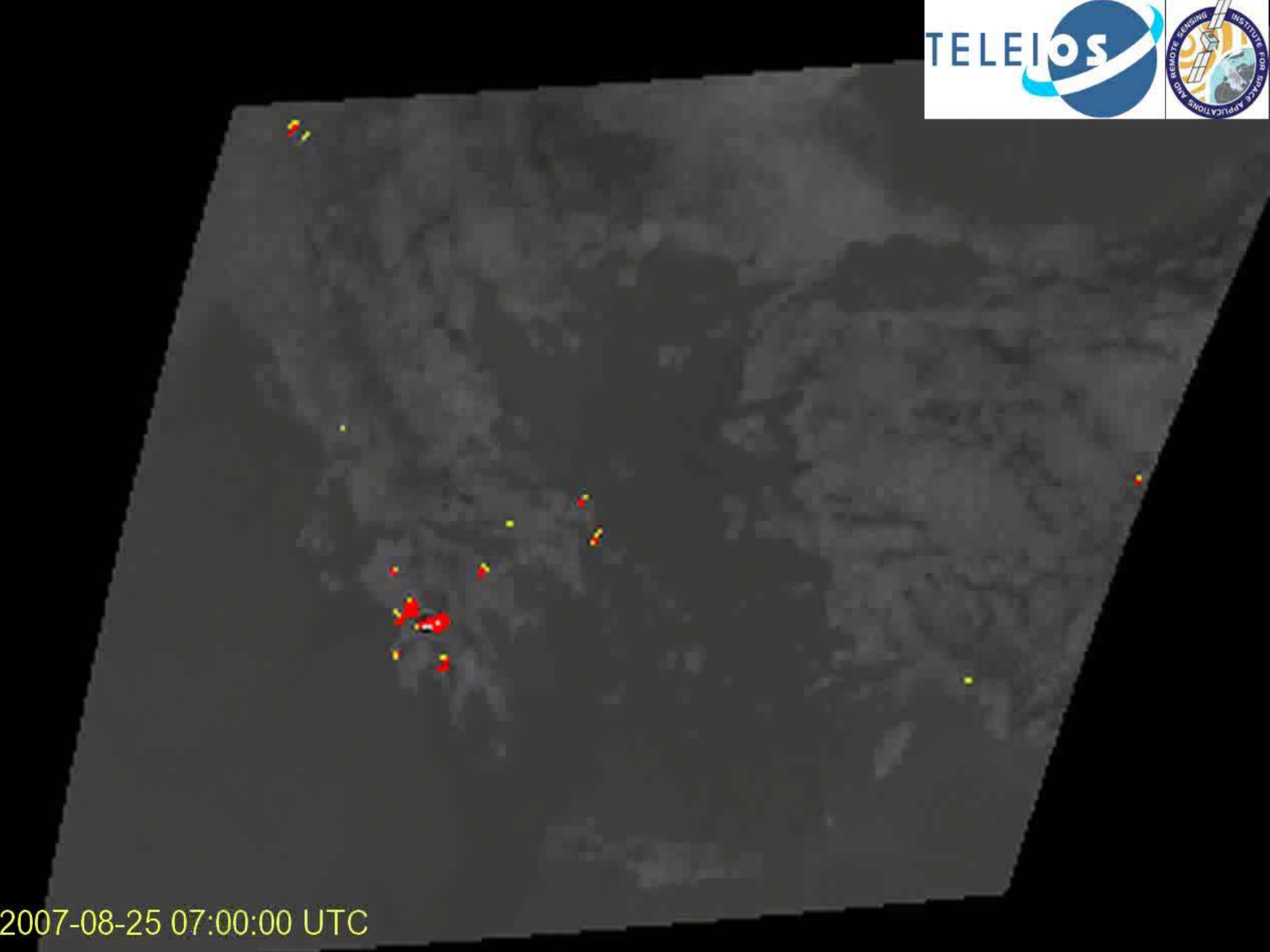
Wildfire Monitoring and Burnt Area Mapping (NOA)



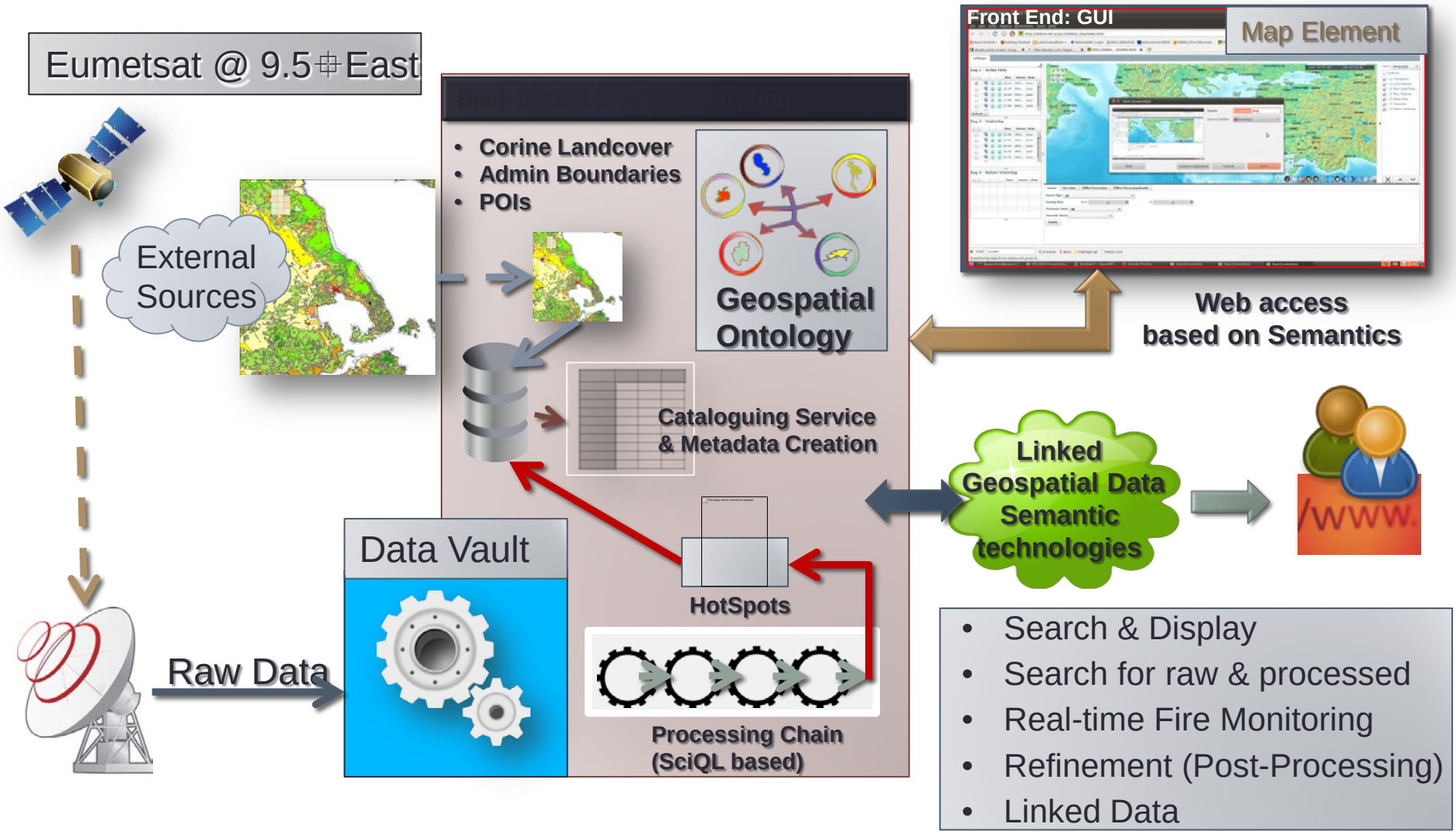


EMERGENCY RESPONSE SERVICE GMES

INSTITUTE FOR SPACE APPLICATIONS AND REMOTE SENSING


TELEIOS
INSTITUTE FOR SPACE APPLICATIONS AND REMOTE SENSING
2007-08-25 07:00:00 UTC

Fire monitoring application



High Level Data Modeling

- Need for representing
  - Standard product **metadata**
  - Standard product **semantic annotations**
  - **Geospatial information**
  - **Temporal information**

- Need to link to other data sources
  - **GIS data**
  - Other information on the **Web**

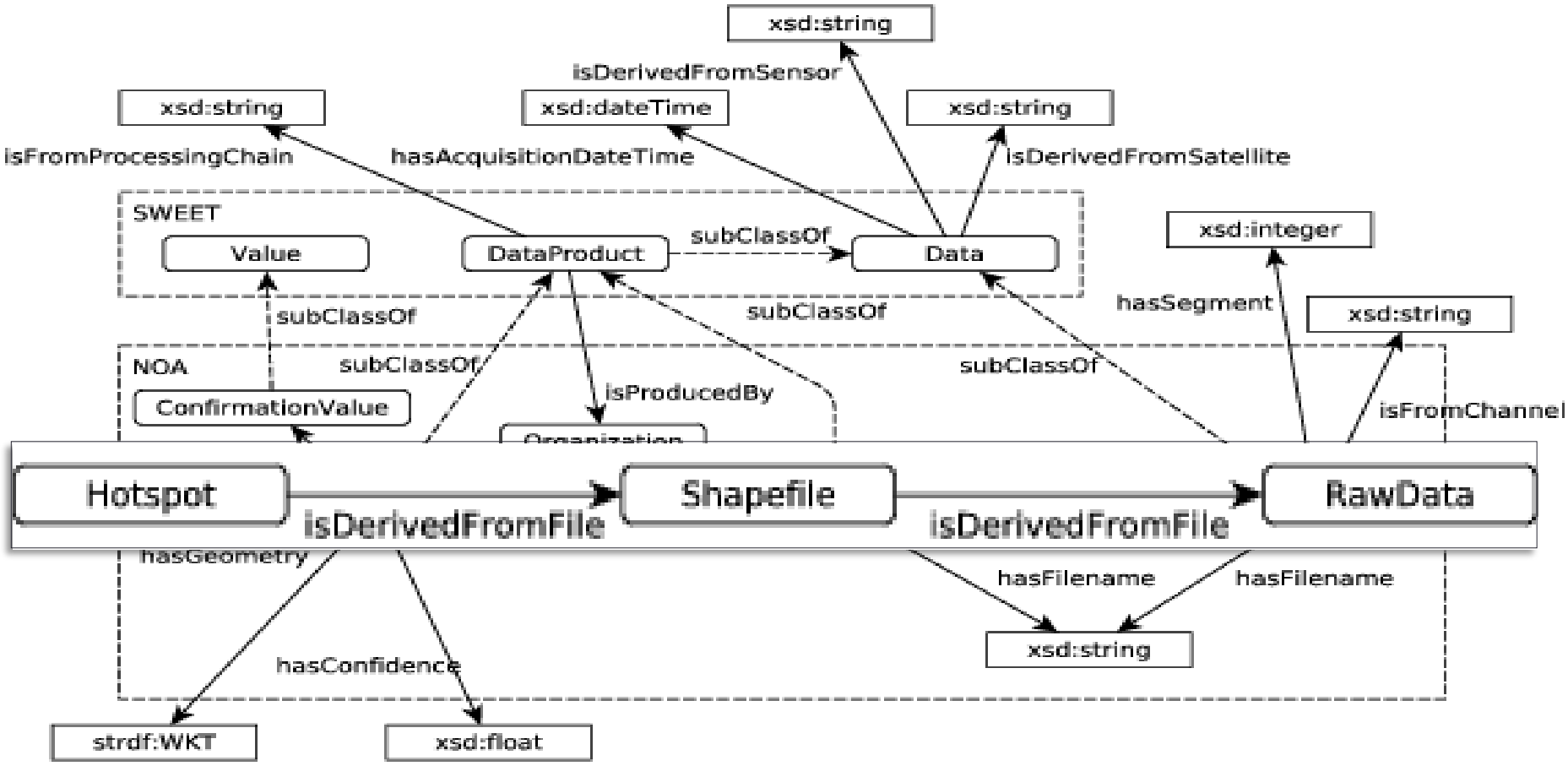
Fire Monitoring Application

- Improving the fire monitoring service using Semantic Web technologies
  - **Representing** fire related products using ontologies
  - **Enriching products** with linked geospatial data
  - **Improving accuracy** with respect to:
    - Underlying land cover/land use
    - Persistence in time

[ISWC 2012 Semantic Web Challenge 3'rd place winner]

http://papos.space.noa.gr/fend_static/

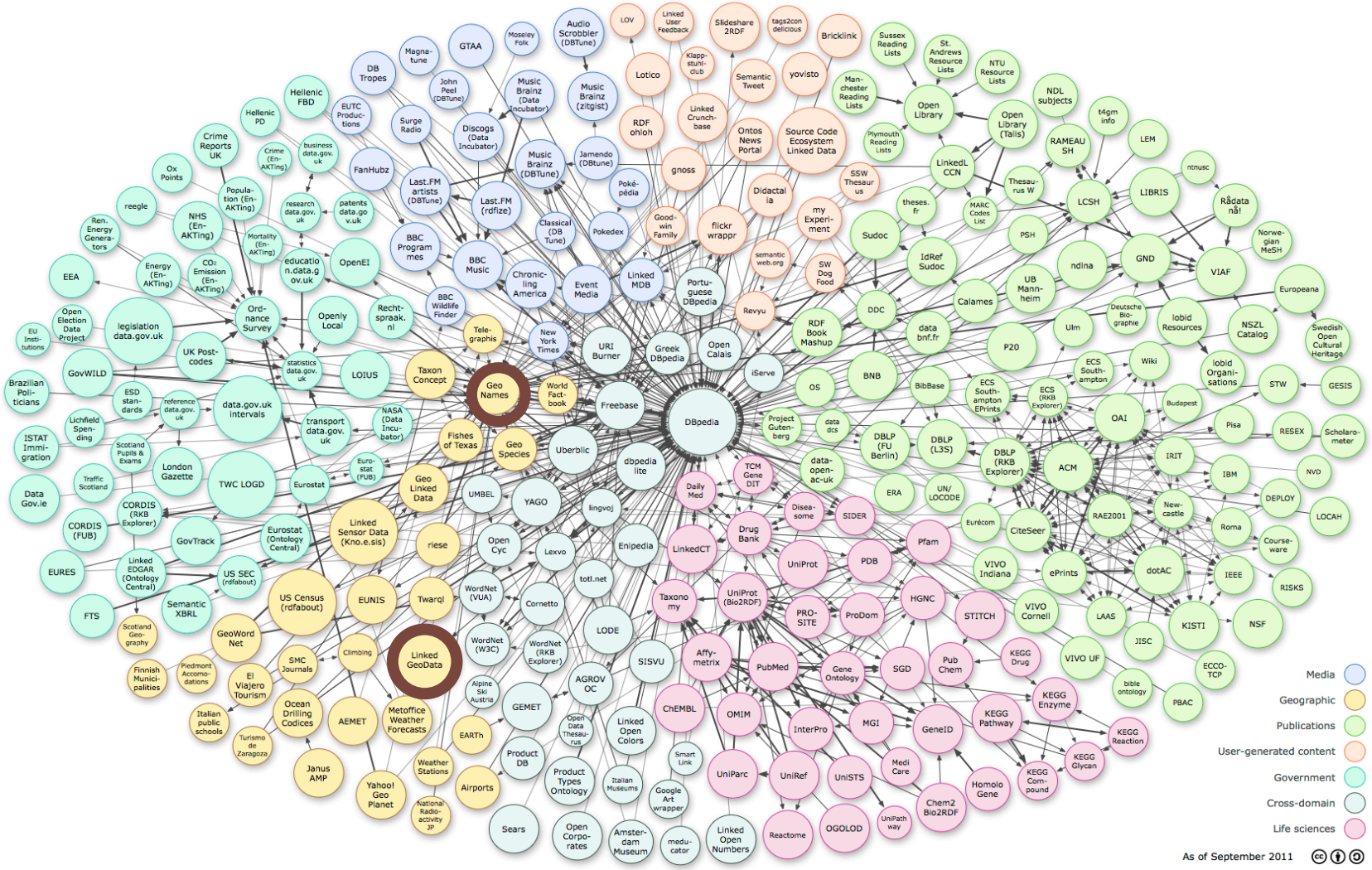
NOA Ontology



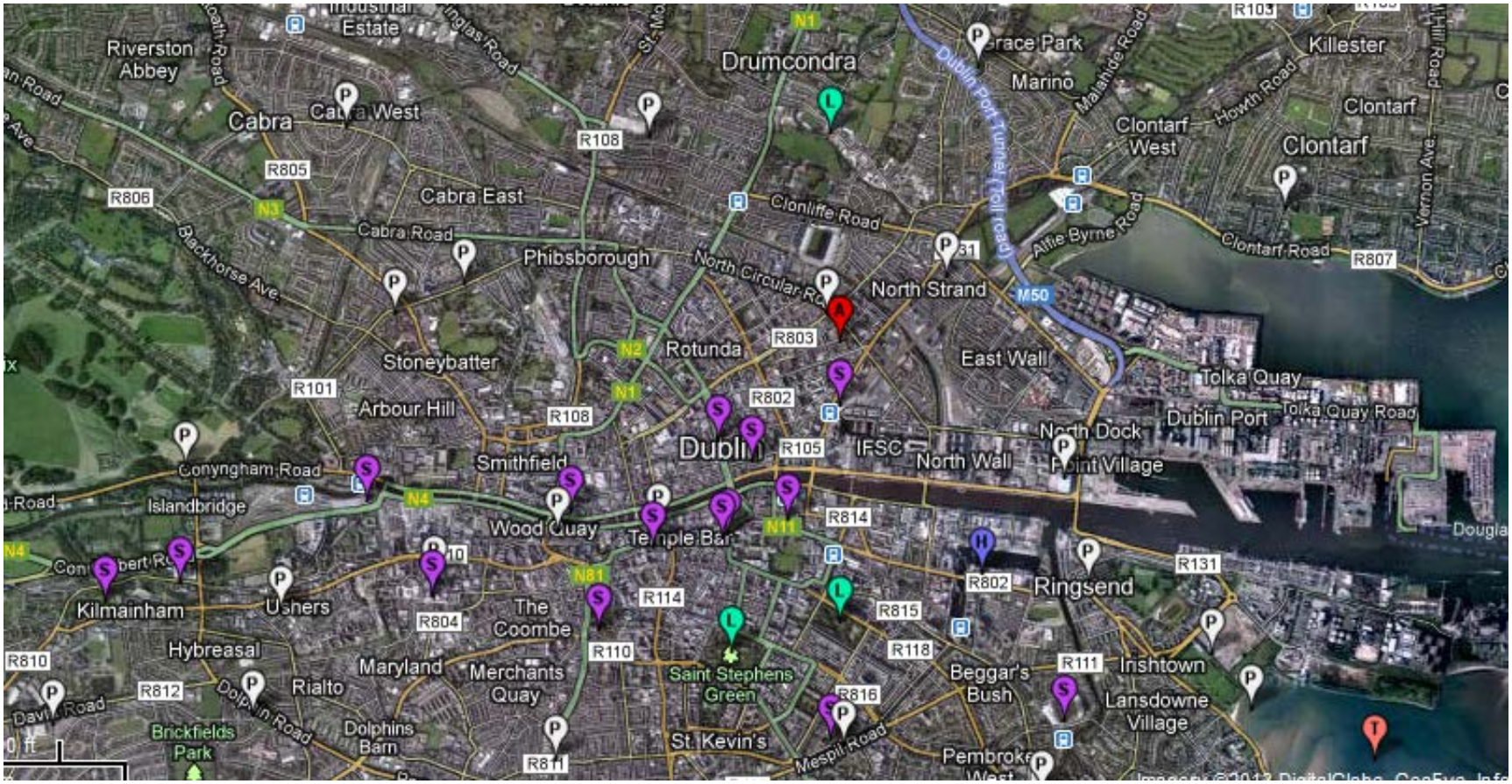
Linked Geospatial Data

- Datasets that we developed and published as linked data:
  - Corine Land Use / Land Cover
  - Coastline of Greece
  - Greek Administrative Geography
- Portal: **http://www.linkedopendata.gr/**
- Datasets from Linked Open Data Cloud
  - LinkedGeoData
  - GeoNames

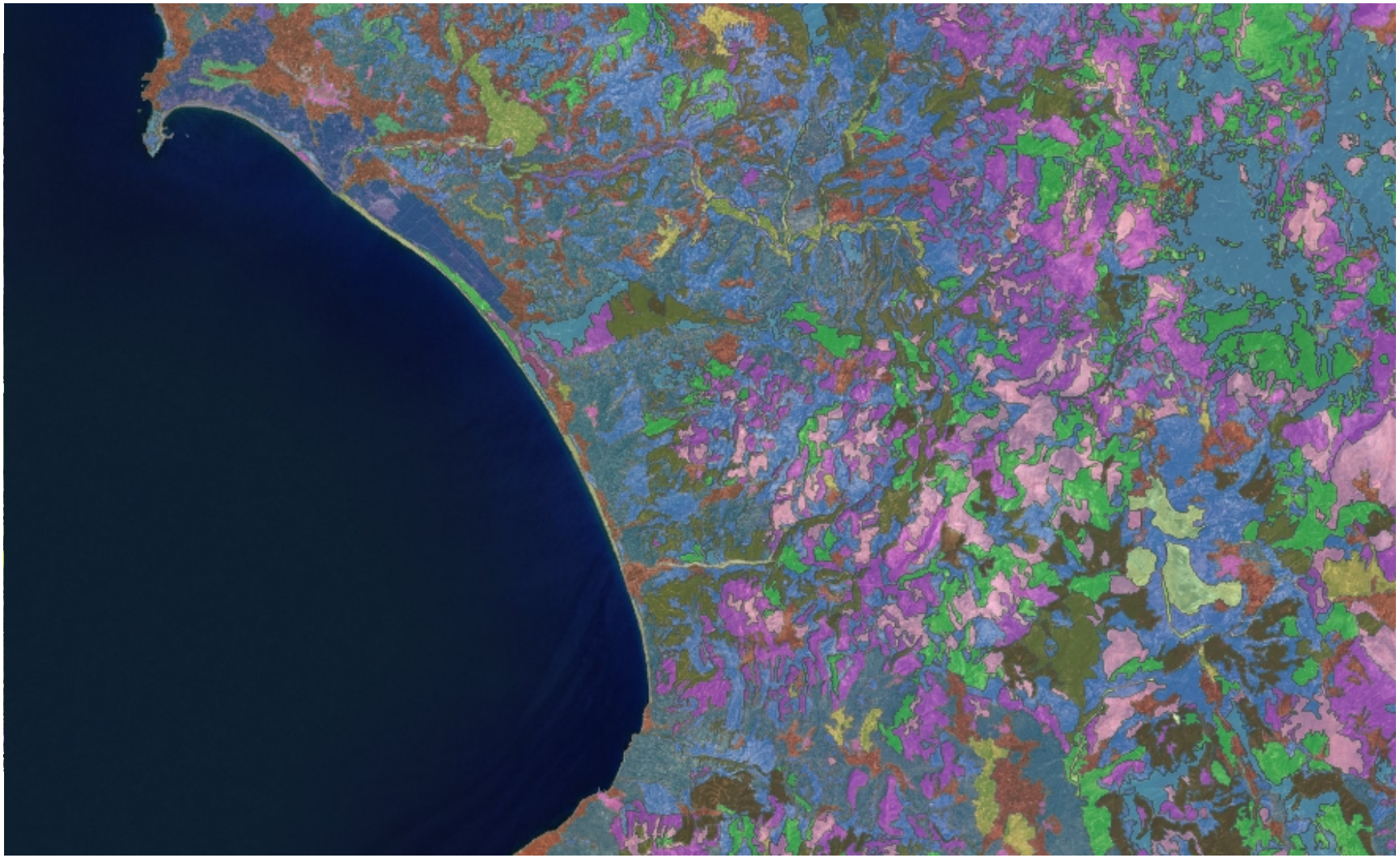
Linked Open Data Cloud



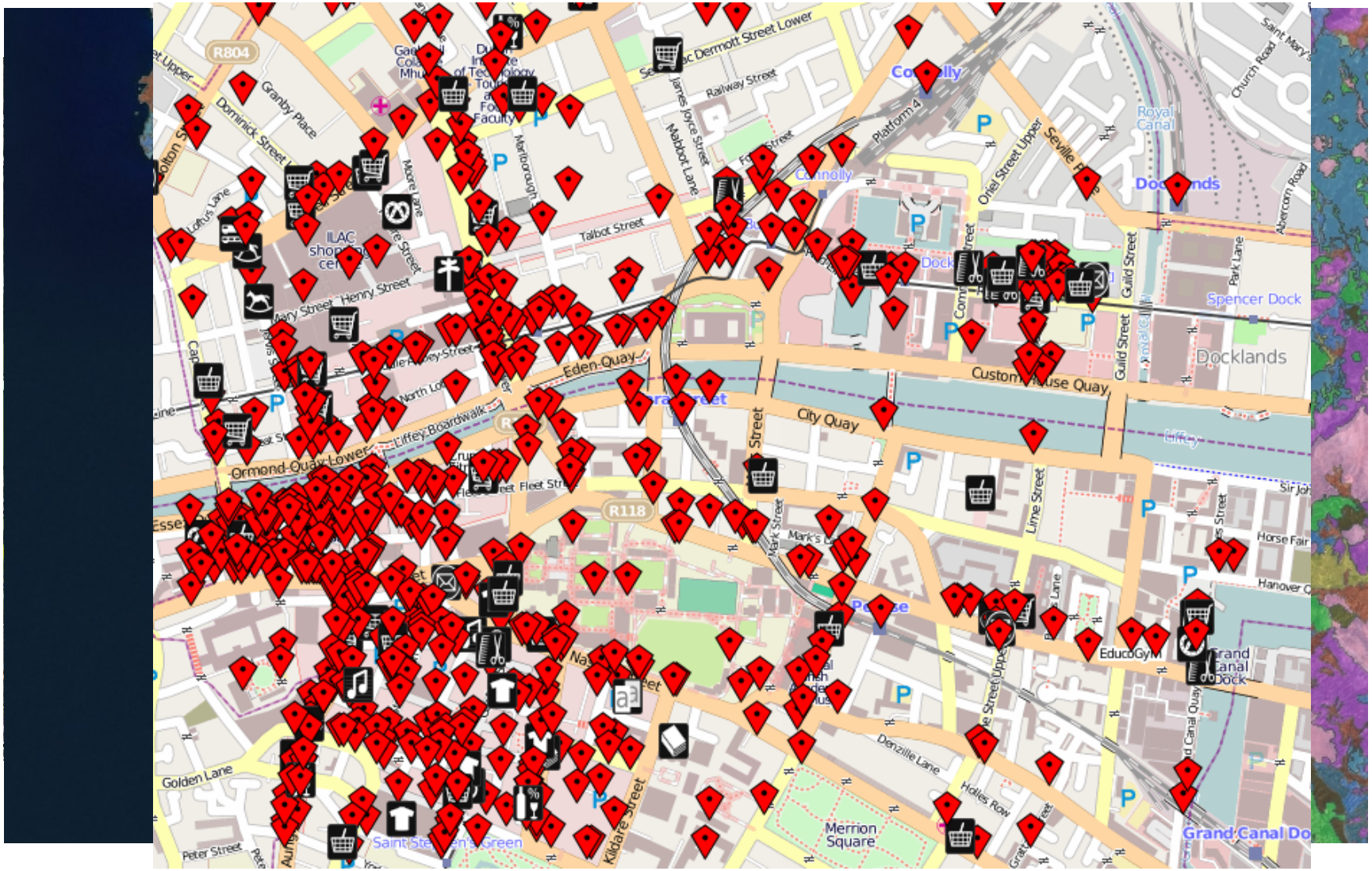
Linked Open Data



Linked Open Data



Linked Open Data



Linked Open Data

Improvements

Using ontologies and stRDF to model knowledge extracted from satellite images, metadata of satellite images and auxiliary geospatial data can improve tasks like:

- **Generated maps** combining diverse information sources
- **Increase hotspot accuracy** correlating them with auxiliary data

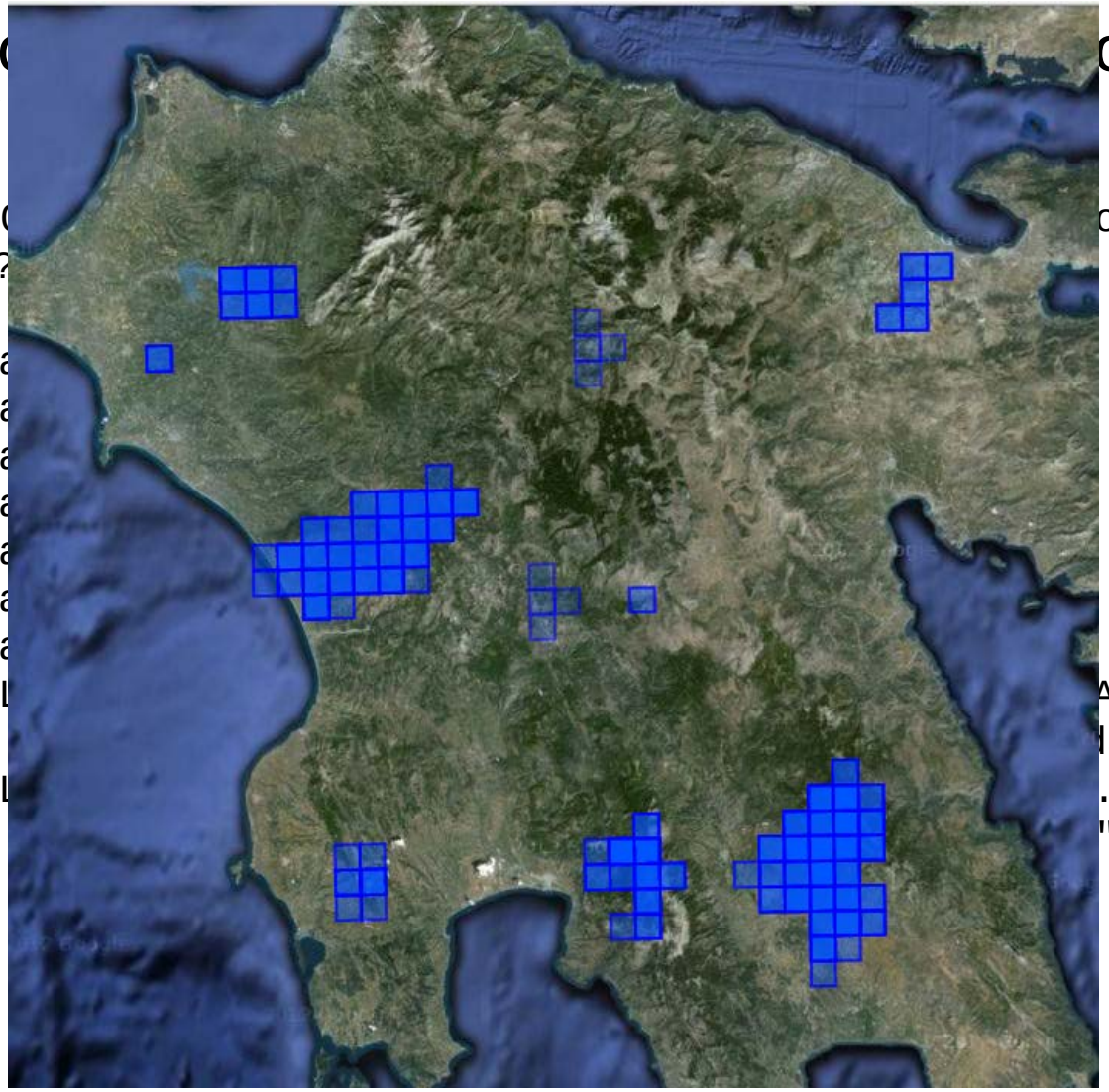
Discovering EO Data

Get all hotspots detected in Peloponnese on 24/08/2007.

```
SELECT ?h ?hGeo ?hAcqTime ?hConfidence ?hConfirmation ?hProvider
   ?hSensor ?hSatellite
WHERE {   ?h  rdf:type noa:Hotspot ;
          noa:hasGeometry ?hGeo ;
          noa:hasAcquisitionTime ?hAcqTime ;
          noa:hasConfidence ?hConfidence ;
          noa:isProducedBy ?hProvider ;
          noa:hasConfirmation ?hConfirmation ;
          noa:isDerivedFromSensor ?hSensor ;
          noa:isDerivedFromSatellite ?hSatellite .
          FILTER("2007-08-24T00:00:00"^^xsd:dateTime <= ?hAcqTime &&
                  ?hAcqTime <= "2007-08-24T23:59:59"^^xsd:dateTime).
          FILTER(strdf:contains("POLYGON((21.027 38.36, 23.77 38.36,
                 23.77 36.05, 21.027 36.05, 21.027 38.36))"
                 ^^strdf:WKT, ?hGeo) ) . }
```

Discovering EO Data

Get all h on 24/08/



```
SELECT ?h ?h                                                    ovider
   ?hSensor ?
WHERE {   ?h
          noa
          noa
          noa
          noa
          noa
          noa
          noa
          FIL                                                   AcqTime &&
                                                                ateTime).
          FIL                                                   .77 38.36,
                                                                "
```

Discovering EO Data

Get all hotspots detected in Peloponnese on 24/08/2007.

```
SELECT ?h ?hGeo ?hAcqTime ?hConfidence ?hConfirmation ?hProvider
   ?hSensor ?hSatellite
WHERE {   ?h  rdf:type noa:Hotspot ;
          noa:hasGeometry ?hGeo ;
          noa:hasAcquisitionTime ?hAcqTime ;
          noa:hasConfidence ?hConfidence ;
          noa:isProducedBy ?hProvider ;
          noa:hasConfirmation ?hConfirmation ;
          noa:isDerivedFromSensor ?hSensor ;
          noa:isDerivedFromSatellite ?hSatellite .
          FILTER("2007-08-24T00:00:00"^^xsd:dateTime <= ?hAcqTime &&
                  ?hAcqTime <= "2007-08-24T23:59:59"^^xsd:dateTime).
          FILTER(strdf:contains("POLYGON((21.027 38.36, 23.77 38.36,
                 23.77 36.05, 21.027 36.05, 21.027 38.36))"
                 ^^strdf:WKT, ?hGeo) ) . }
```

Retrieving a Map Layer (1/3)

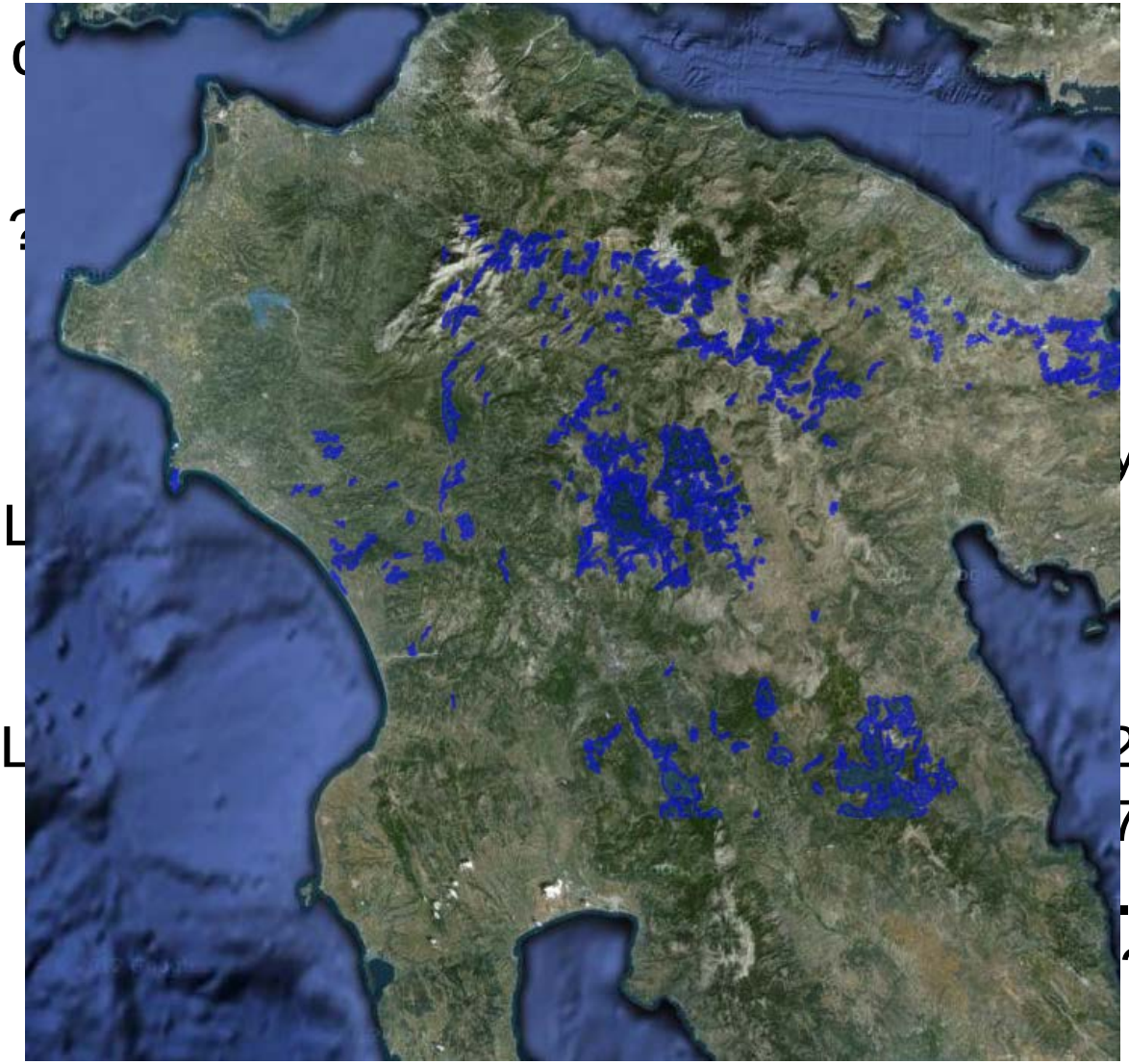
Get all coniferous forests in Peloponnese

```
SELECT  ?a ?aGeo
WHERE{?a rdf:type clc:Area;
         clc:hasLandUse ?aLandUse;
          noa:hasGeometry ?aGeo.
       ?aLandUse rdf:type ?aLandUseType.
    FILTER(?aLandUseType =
                clc:ConiferousForest).

    FILTER(strdf:contains("POLYGON((21.027
              38.36, 23.77 38.36, 23.77 36.05,
              21.027 36.05, 21.027 38.36))”
                        ^^strdf:WKT,?aGeo)).
}
```

Retrieving a Map Layer (1/3)

Get all c

```
SELECT
WHERE{?

                                                    ype.
      FIL

                                                    .
      FIL                                           21.027
                                                    77 36.05,
                                                    .36))”
                                                    ?aGeo)).
}
```



Retrieving a map layer (2/3)

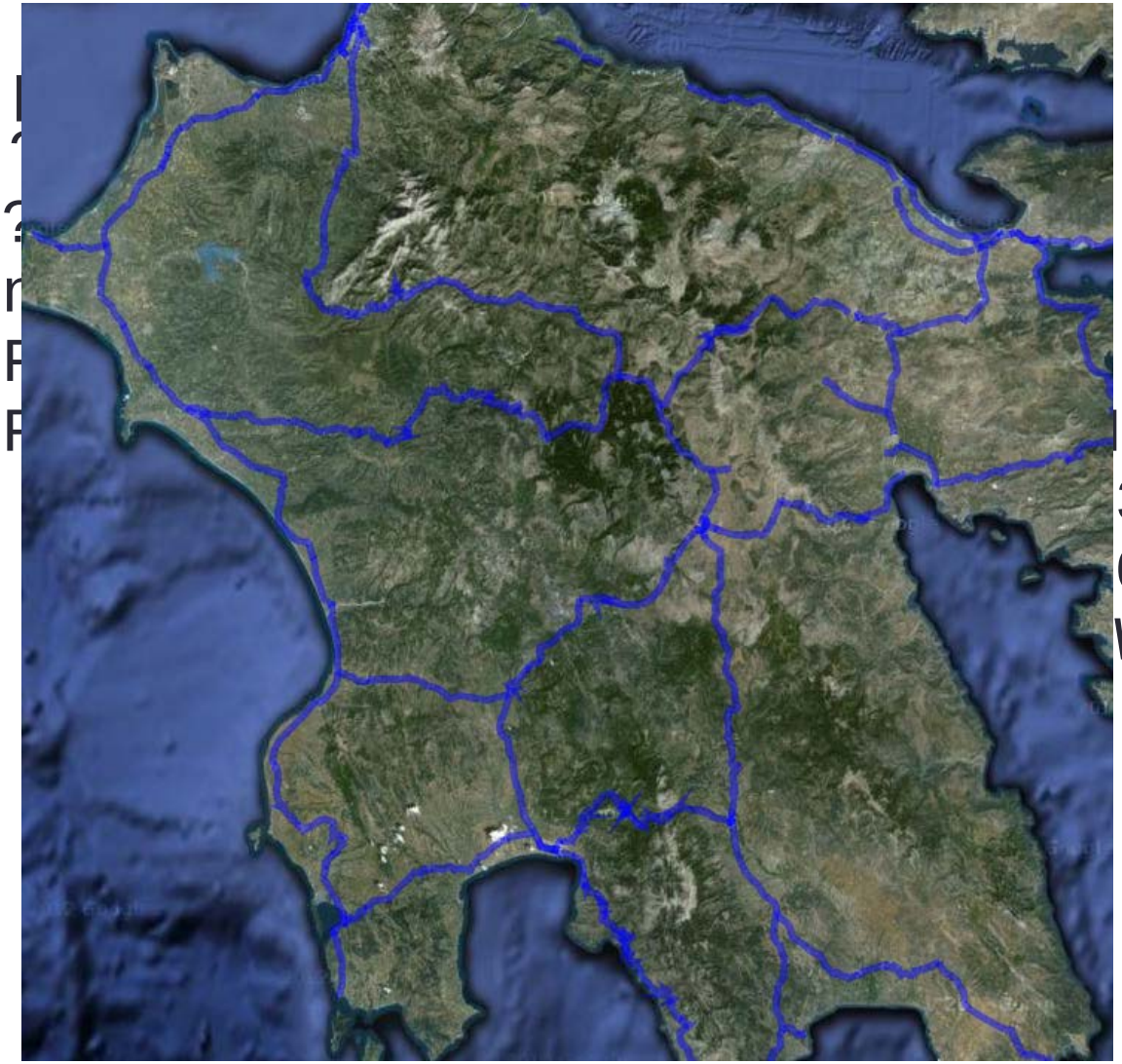
- Get all primary roads in Pelloponnese

```
SELECT ?r ?rGeo
WHERE{ ?r  a ?rType ;
       noa:hasGeometry ?rGeo .
       FILTER(?rType = lgdo:Primary) .
       FILTER(strdf:contains("POLYGON((
              21.027 38.36, 23.77 38.36,
              23.77 36.05, 21.027 36.05,
              21.027 38.36))"^^strdf:WKT,
              ?rGeo) ).
}
```

Retrieving a map layer (2/3)

- Get all

```
SELECT
WHERE{ ?
r
F
F
                                                    .
                                                    (( 
                                                    36,
                                                    05,
                                                    WKT,

}
```



Retrieving a Map Layer (3/3)

Get all capitals of prefectures of the Peloponnese.

```
SELECT ?feature ?fName ?fGeo
WHERE{?feature rdf:type gn:Feature;
                    noa:hasGeography ?fGeo;
                    gn:name ?fName;
                    gn:featureCode ?fCode.
          FILTER(?fCode = gn:P.PPLA
                  || ?fCode = gn:P.PPLA2 ) .
        FILTER(strdf:contains("POLYGON((21.51
                    36.41, 22.83 36.41, 22.83
                    37.69, 21.51 37.69,
                    21.51 6.41 ))”
                         ^^strdf:WKT, ?fGeo)).
}
```

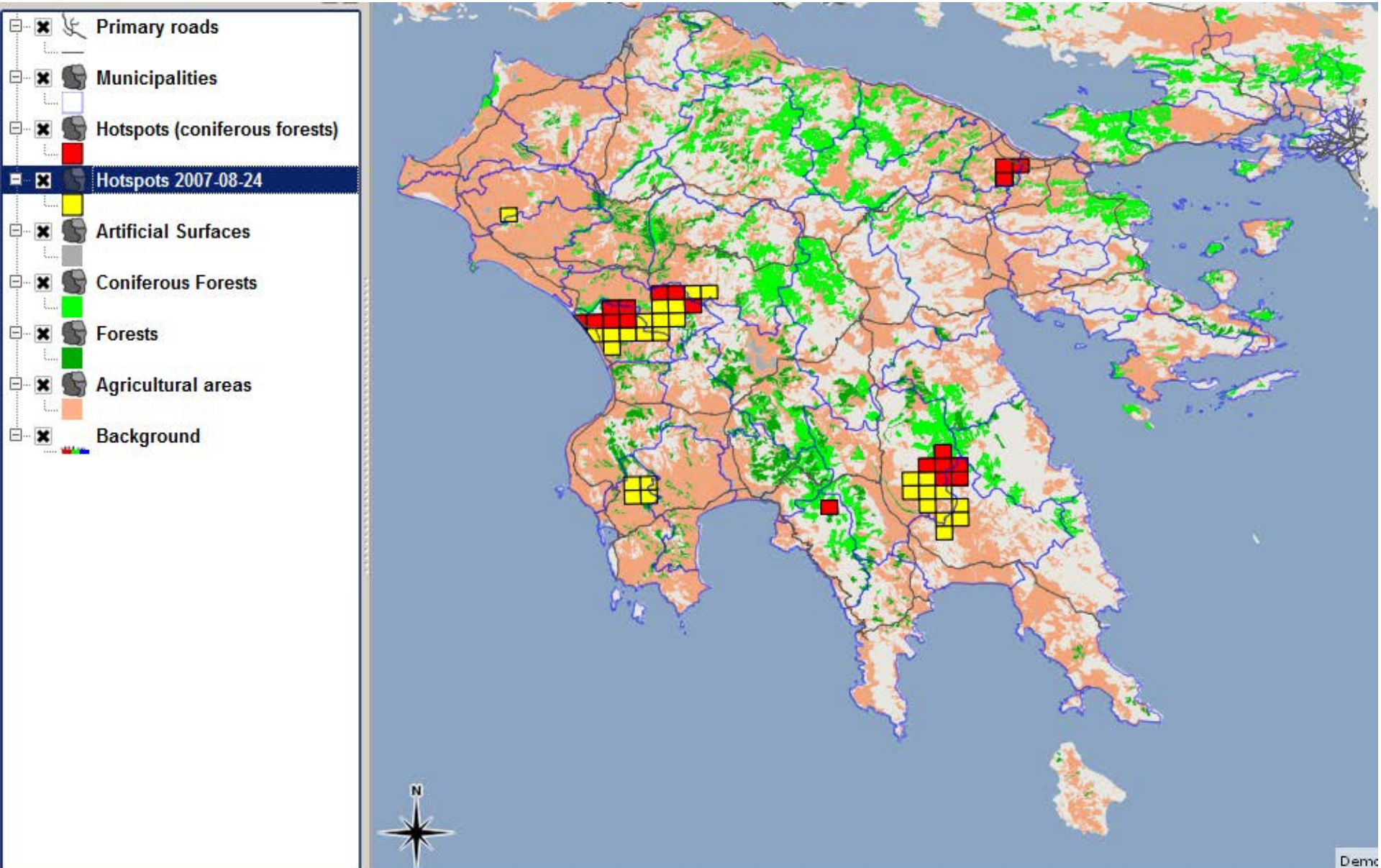
Retrieving a Map Layer (3/3)

Get all

nese.

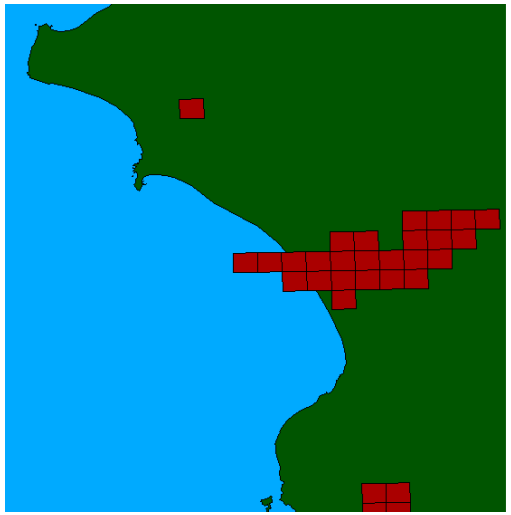
```
SELECT
WHERE{

o;

.

) .
(21.51
2.83

eo)).

}
```

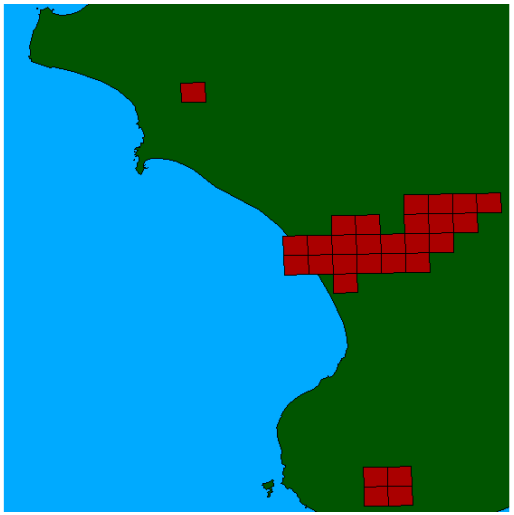
Final Map

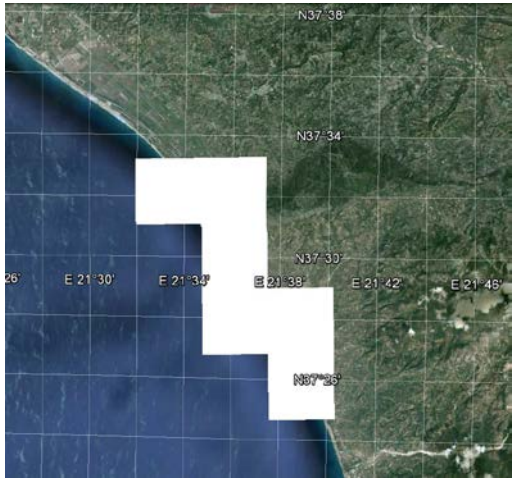

Primary roads
Municipalities
Hotspots (coniferous forests)
Hotspots 2007-08-24
Artificial Surfaces
Coniferous Forests
Forests
Agricultural areas
Background
N

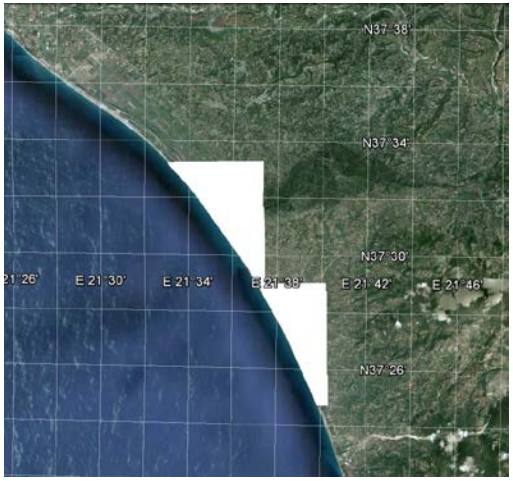
Semantic Enrichment for Hotspots

- **Enrich** hotspot products
  1. Connect each hotspot with a municipality that it is located
- **Improve accuracy** with respect to **underlying area**
  2. Eliminate false alarms in sea
  3. Eliminate false alarms in inconsistent land cover areas
  4. Keep land part of the polygon





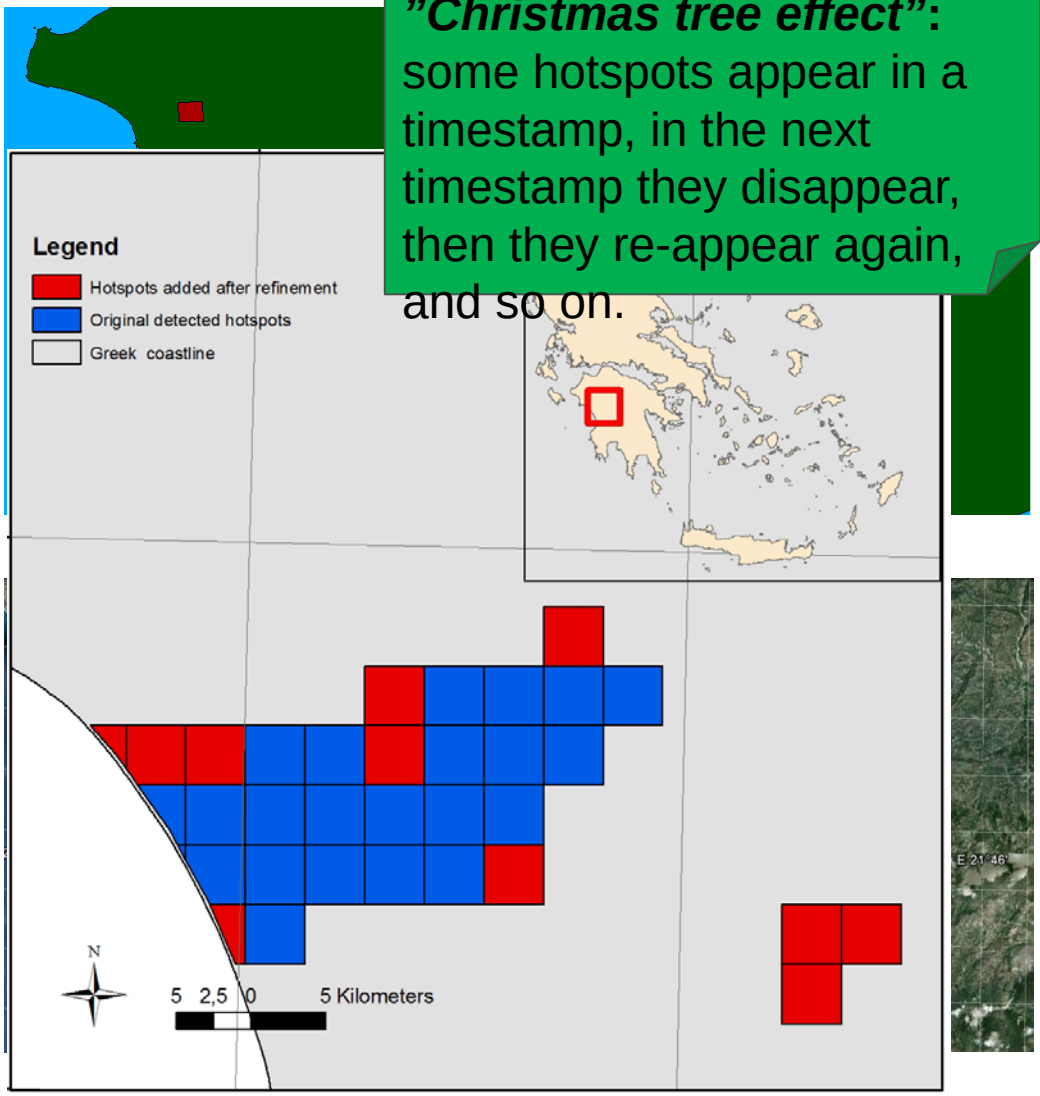




Semantic Enrichment for Hotspots

- **Enrich** hotspot products
  1. Connect each hotspot with a municipality that it is located
- **Improve accuracy** with respect to **underlying area**
  2. Eliminate false alarms in sea
  3. Eliminate false alarms in inconsistent land cover areas
  4. Keep land part of the polygon
- **Improve accuracy** with respect to **temporal persistence** of each hotspots
  5. Remove “Christmas tree” effects

***”Christmas tree effect”*:** some hotspots appear in a timestamp, in the next timestamp they disappear, then they re-appear again, and so on.

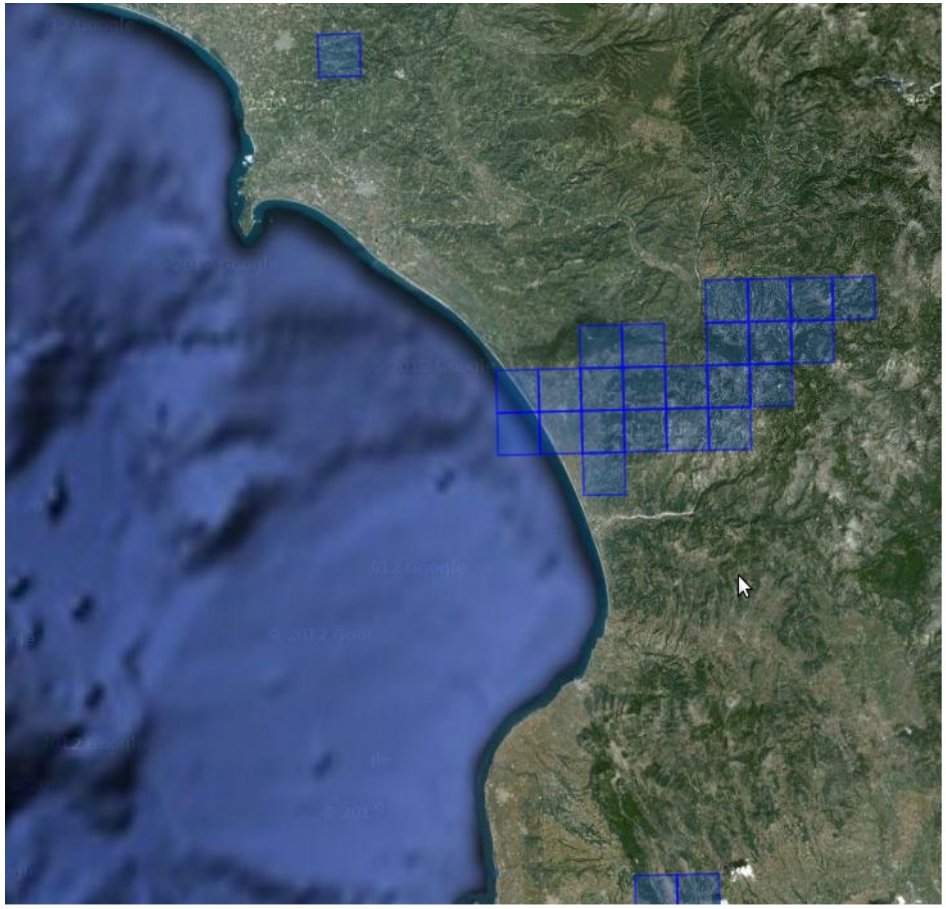


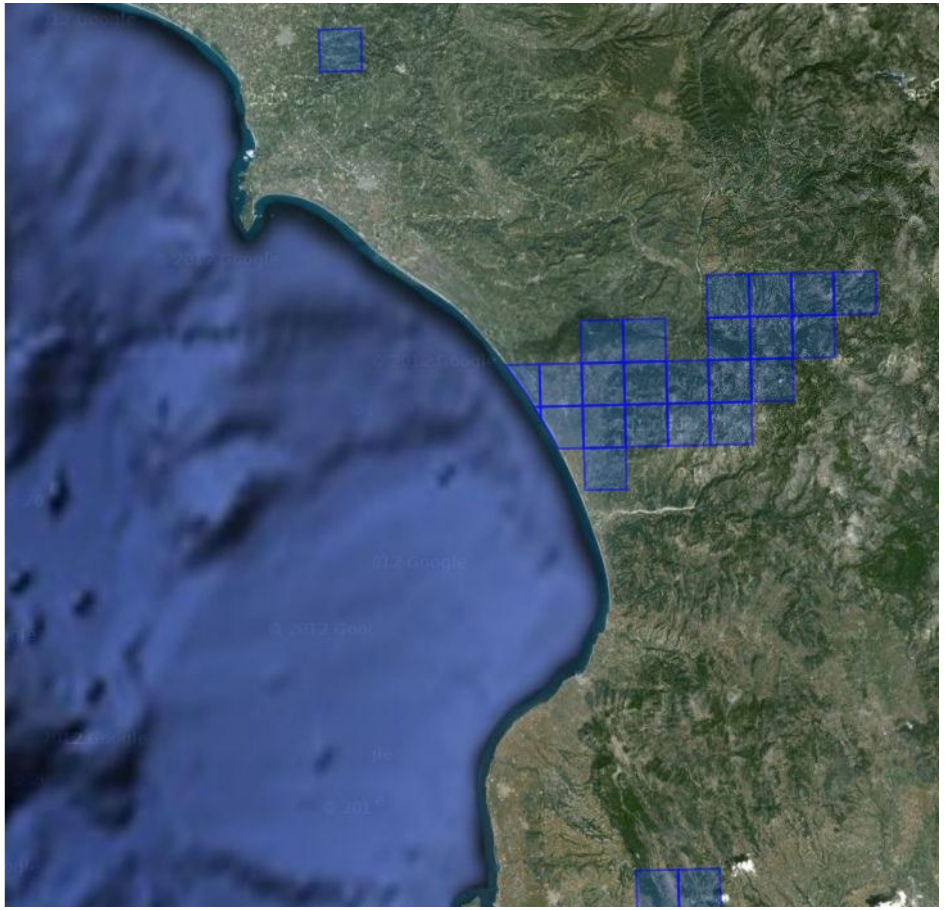
Improving the Accuracy of EO Data

Correlate fire products with auxiliary data to increase their thematic accuracy e.g., delete the parts of the polygons that fall into the sea.

```
DELETE {?h noa:hasGeometry ?hGeo}
INSERT {?h noa:hasGeometry ?dif}
WHERE {
SELECT DISTINCT ?h ?hGeo
   (strdf:intersection(?hGeo, strdf:union(?cGeo)) AS ?dif)
WHERE {
   ?h rdf:type noa:Hotspot.
   ?h strdf:hasGeometry ?hGeo.
   ?c rdf:type coast:Coastline.
   ?c strdf:hasGeometry ?cGeo.
   FILTER( strdf:anyInteract(?hGeo, ?cGeo)}
GROUP BY ?h ?hGeo
HAVING strdf:overlap(?hGeo, strdf:union(?cGeo))}
```

Improving the Accuracy of EO Data





```
GROUP BY ?h ?hGeo
HAVING strdf:overlap(?hGeo, strdf:union(?cGeo))}
```

Fire monitoring service

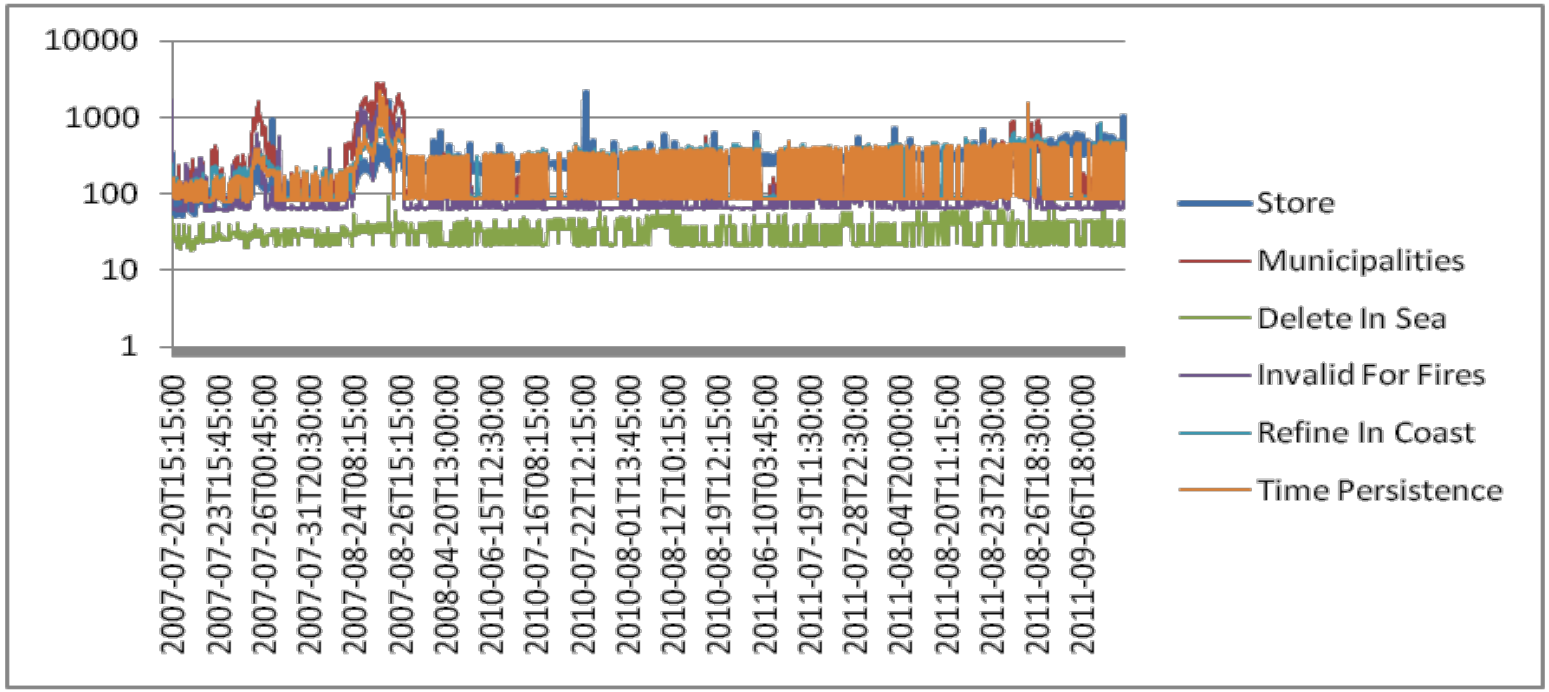
- The fire monitoring service was used **operationally** during the **fire season** of **2012**

- Used in a **daily basis** by the
  - Greek civil protection agency
  - Greek fire brigade
  - Greek army

- Initial user feedback very encouraging!

Fire monitoring service

- Product ingestion, processing and refinement is completed in less than 12 seconds
- More refinement operations to be added later given the five minutes time frame

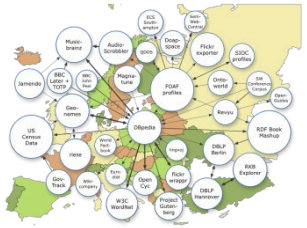


Conclusions

Future Work

- Use even larger datasets
- Compare with other systems
  - GeoSPARQL implementation of Oracle
- stSPARQL query processing in MonetDB
- Go distributed!
  - Federated queries

Thanks! Any Questions?



- Strabon
  - Manolis Koubarakis, Kostis Kyzirakos, Manos Karpathiotakis, Charalampos Nikolaou, Giorgos Garbis, Konstantina Bereta, Kallirroi Dogani, Stella Giannakopoulou and Panayiotis Smeros
  - Web site: http://strabon.di.uoa.gr
  - Mercurial repository: http://hg.strabon.di.uoa.gr
  - Trac: http://bug.strabon.di.uoa.gr
  - Mailing list: http://cgi.di.uoa.gr/~mailman/listinfo/strabon-users
- Real Time Fire Monitoring Service, National Obervatory of Athens
  - http://papos.space.noa.gr/fend_static
- Greek Linked Open Data
  - http://www.linkedopendata.gr
- TELEIOS EU Project
  - http://www.earthobservatory.eu
- SemsorGrid4Env EU Project
  - http://www.semsorgrid4env.eu