

## Geospatial data in RDF – stSPARQL

Presenter: Kostis Kyzirakos



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



# Outline

---

- Main idea
- Early works
- The data model stRDF
- Examples of publicly available linked geospatial data
- The query language stSPARQL

# Main idea

---

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

Extend RDF to take into account the geospatial dimension.

Extend SPARQL to query the new kinds of data.

# Early works

---

## SPAUK

[Kolas and Self, 2007]

- Geometric attributes of a resource are represented by:
  - introducing a **blank node** for the geometry
  - specifying the geometry using **GML vocabulary**
  - associating the blank node with the resource using **GeoRSS vocabulary**
- Queries are expressed in SPARQL utilizing appropriate geometric vocabularies and ontologies (e.g., the topological relationships of RCC-8).
- Introduces a new **PREMISE** clause in SPARQL to specify spatial geometries to be used in a query
- Use some form of the **DESCRIBE** query form of SPARQL for asking queries about geometries

# Early works

---

## SPARQL-ST

[Perry, 2008]

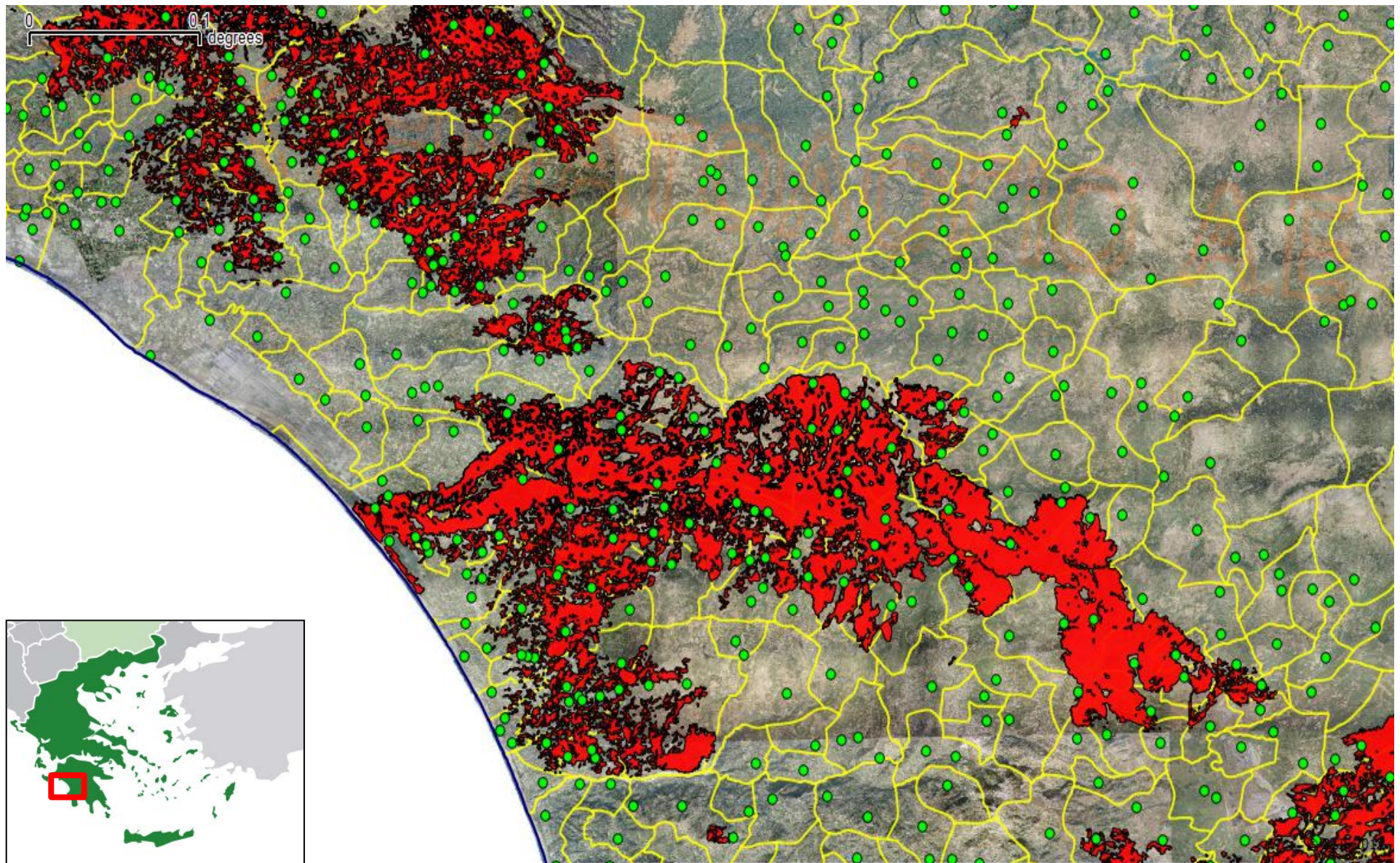
- Assumes a particular upper ontology expressed in RDFS for modeling **theme**, **space** and **valid time**.
- Spatial geometries in SPARQL-ST are specified by **sets of RDF triples** that give various details of the geometry.
- SPARQL-ST provides a set of built-in spatial conditions that can be used in **SPATIAL FILTER** clauses to constrain the geometries that are returned as answers to queries.

# stRDF and stSPARQL

---

- Similar approach to SPARQL-ST (**theme**, **space** and **valid time** can be represented) *[Koubarakis and Kyzirakos, 2010]*
- **Linear constraints** are used to represent geometries
- Constraints are represented using literals of an appropriate datatype
- Formal approach
- New version to be presented today uses **OGC standards** to represent and query geometries

# Example



# Example with simplified geometries



# Example in stRDF

**geonames:Olympia**

**geonames:name "Ancient Olympia";**

**owl:sameAs dbpedia:Olympia\_Greece;**

**rdf:type dbpedia:Community .**



**geonames:Olympia strdf:hasGeometry**

**"POLYGON((21.5 18.5, 23.5 18.5,  
23.5 21, 21.5 21, 21.5 18.5));**

**<http://www.opengis.net/def/crs/EPSG/0/4326>"^^**

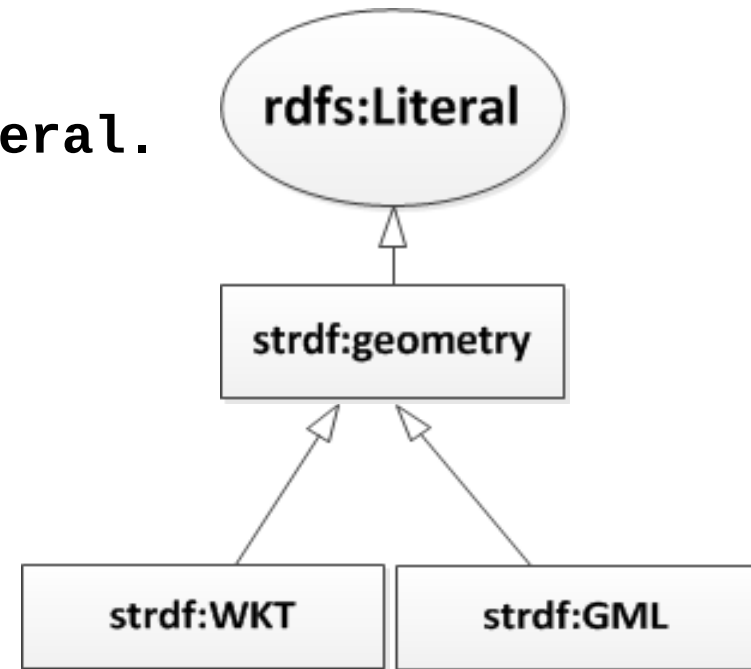
**strdf:WKT**

**Spatial  
literal**

**Spatial  
data type**

# The stRDF Data Model

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



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

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

# The stRDF Data Model

---

We define the datatypes **strdf:WKT** and **strdf:GML** that can be used to represent spatial objects using the WKT and GML serializations.

- **Lexical space:** the finite length sequences of characters that can be produced from the WKT and GML specifications.
  - Literals of type **strdf:WKT** consist of an optional URI identifying the coordinate reference system used.

e.g., **"POINT(21 18);  
<http://www.opengis.net/def/crs/EPSG/0/4326>"  
^^strdf:WKT**

# The stRDF Data Model

---

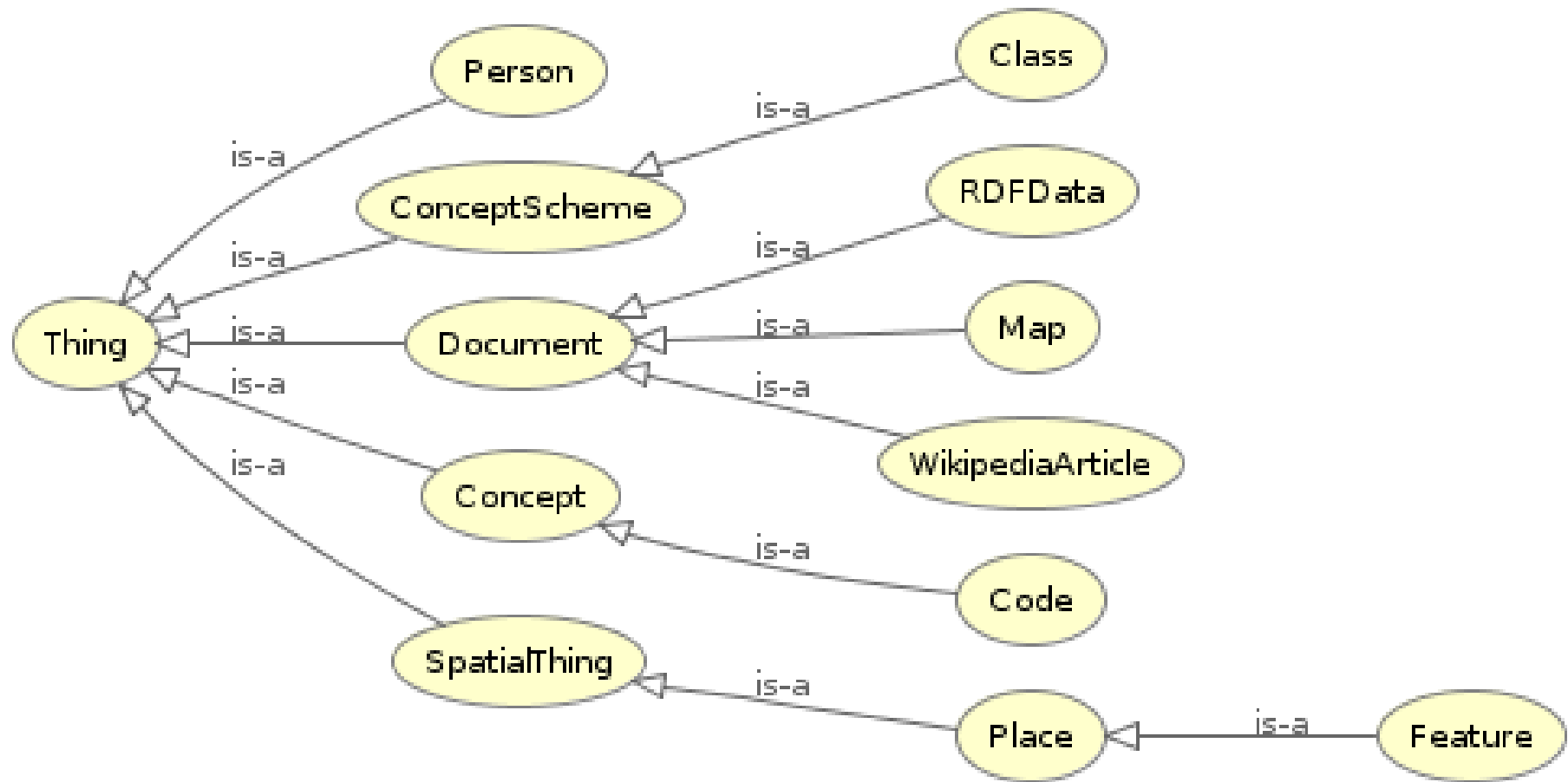
- **Value space:** the set of geometry values defined in the WKT and GML standard that is a subset of the powerset of  $\mathbb{R}^2$  and  $\mathbb{R}^3$ .
- **Lexical-to-value mapping:** takes into account that the vector-based model is used for representing geometries.
- The datatype **strdf:geometry** is the union of the datatypes **strdf:WKT** and **strdf:GML**.

# Examples of publicly available linked geospatial data

---

- Geonames
- Greek Administrative Geography
- Corine Land Use / Land Cover
- Burnt Area Products

# Geonames



# Geonames

Map center : N 48° 12' 21" E 16° 27' 49"

[google earth](#) [taqzania](#) [mapquest](#)

Map center : N 48° 12' 21" E 16° 27' 49"

Map Satellite Hybrid Terrain

searching for "Vienna"

[GeoNames](#) [Wikipedia](#)

**features**

- ☒ city, village, ...
- ☒ mountain, hill, rock, ...
- ☒ stream, lake, ...
- ☒ country, state, region, ...
- ☒ parks, area, ...
- ☒ road, railroad
- ☒ spot, building, farm
- ☒ forest, heath, ...
- ☒ undersea

Map data ©2012 GeoBasis-DE/BKG (©2009), Google, Tele Atlas

# Geonames

---

**gn:2761333**

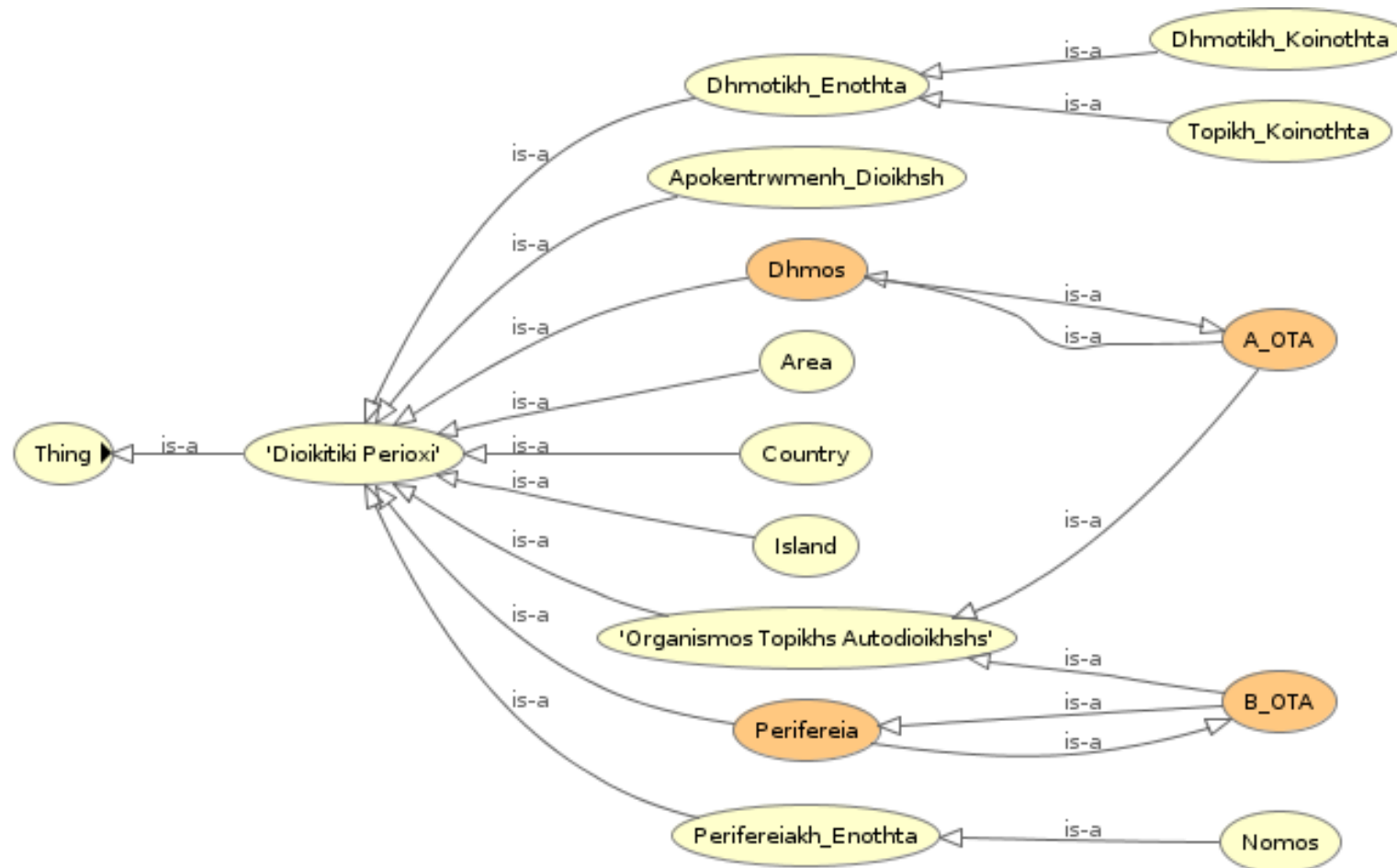
```
rdf:type geonames:Feature;  
geonames:officialName "Vienna"@en;  
geonames:name "Politischer Bezirk Wien (Stadt)";  
geonames:countryCode "AT";  
wgs84_pos:lat "48.2066";  
wgs84_pos:long "16.37341".  
geonames:parentCountry gn:2782113;
```

**gn:2782113**

```
geonames:name "Austria";  
geonames:altName "Republic of Austria"@en,  
"Republik Österreich"@de,  
"Αυστρία"@el.
```

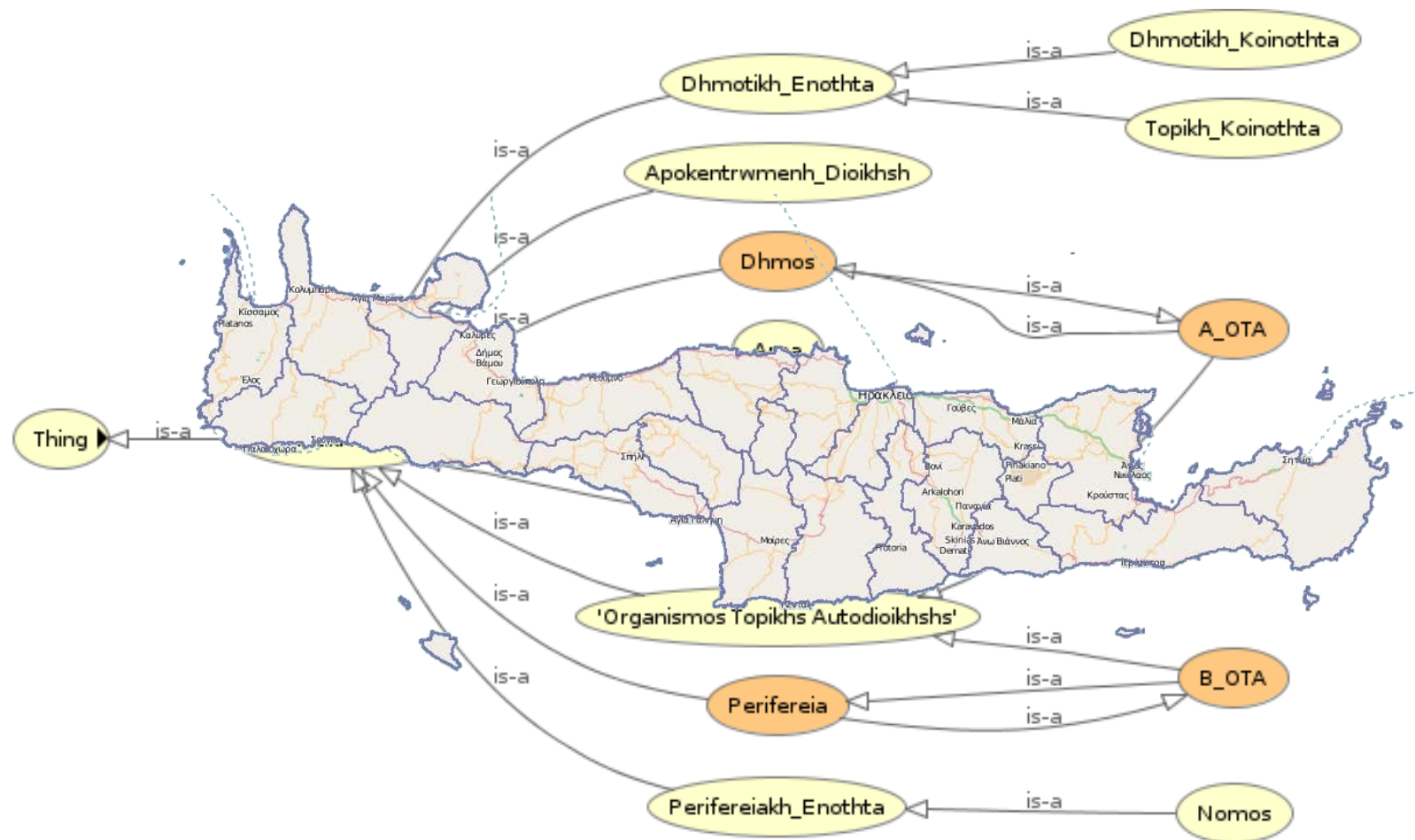
# Greek Administrative Geography

## Kallikrates ontology



# Greek Administrative Geography

## Kallikrates ontology



# Greek Administrative Geography

```
gag:Olympia
  rdf:type gag:Community;
  geonames:name "Ancient Olympia";
  gag:population "184"^^xsd:int;
  strdf:hasGeometry "POLYGON
  (((25.37 35.34,...)))"^^strdf:WKT.
```

```
gag:OlympiaBorough
  rdf:type gag:Borough;
  rdfs:label "Borough of
  Ancient Olympia".
```

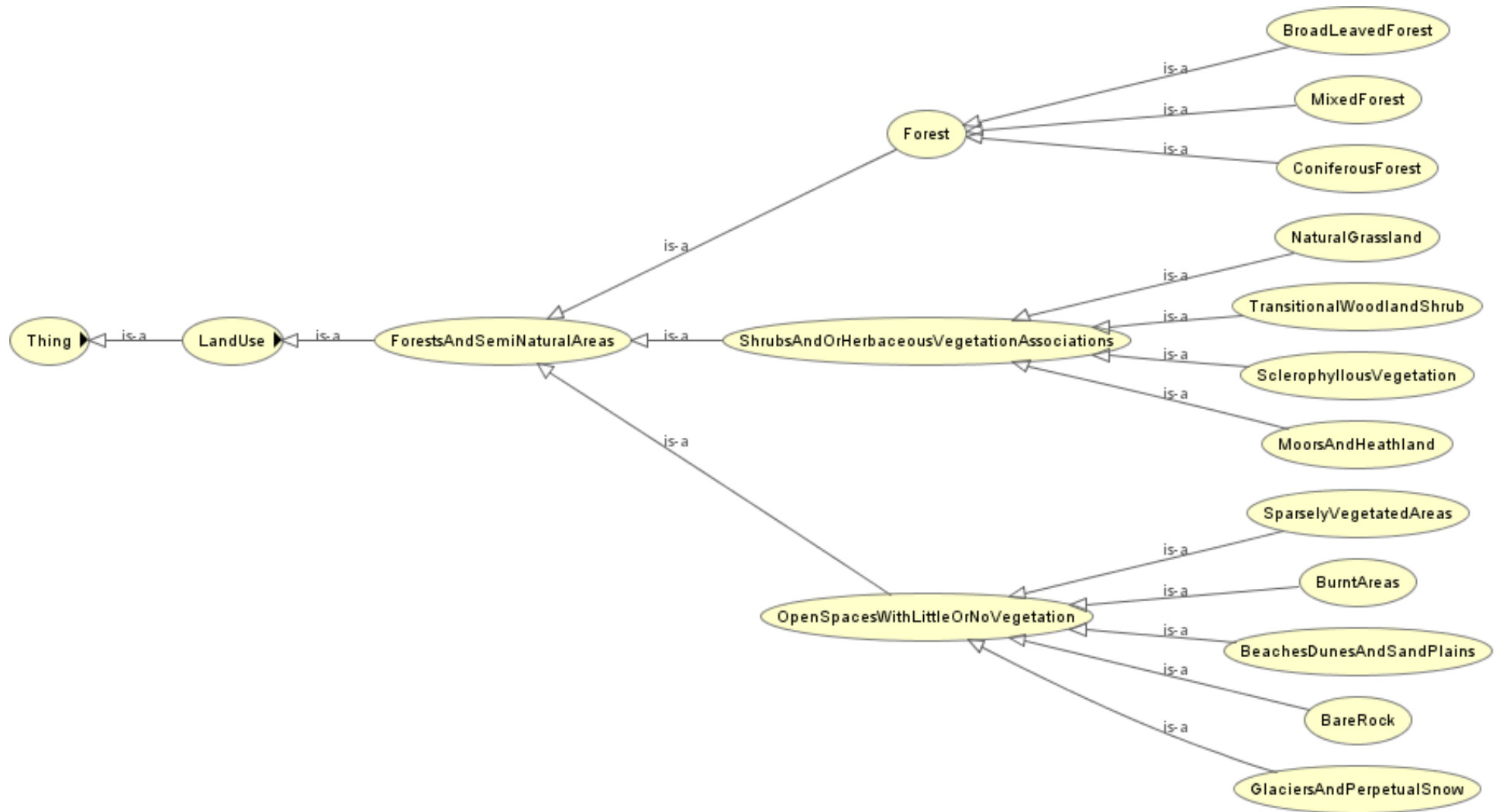
```
gag:OlympiaMunicipality
  rdf:type gag:Municipality;
  rdfs:label "Municipality of
  Ancient Olympia".
```

```
gag:Olympia gag:isPartOf gag:OlympiaBorough .
```

```
gag:OlympiaBorough gag:isPartOf gag:OlympiaMunicipality.
```

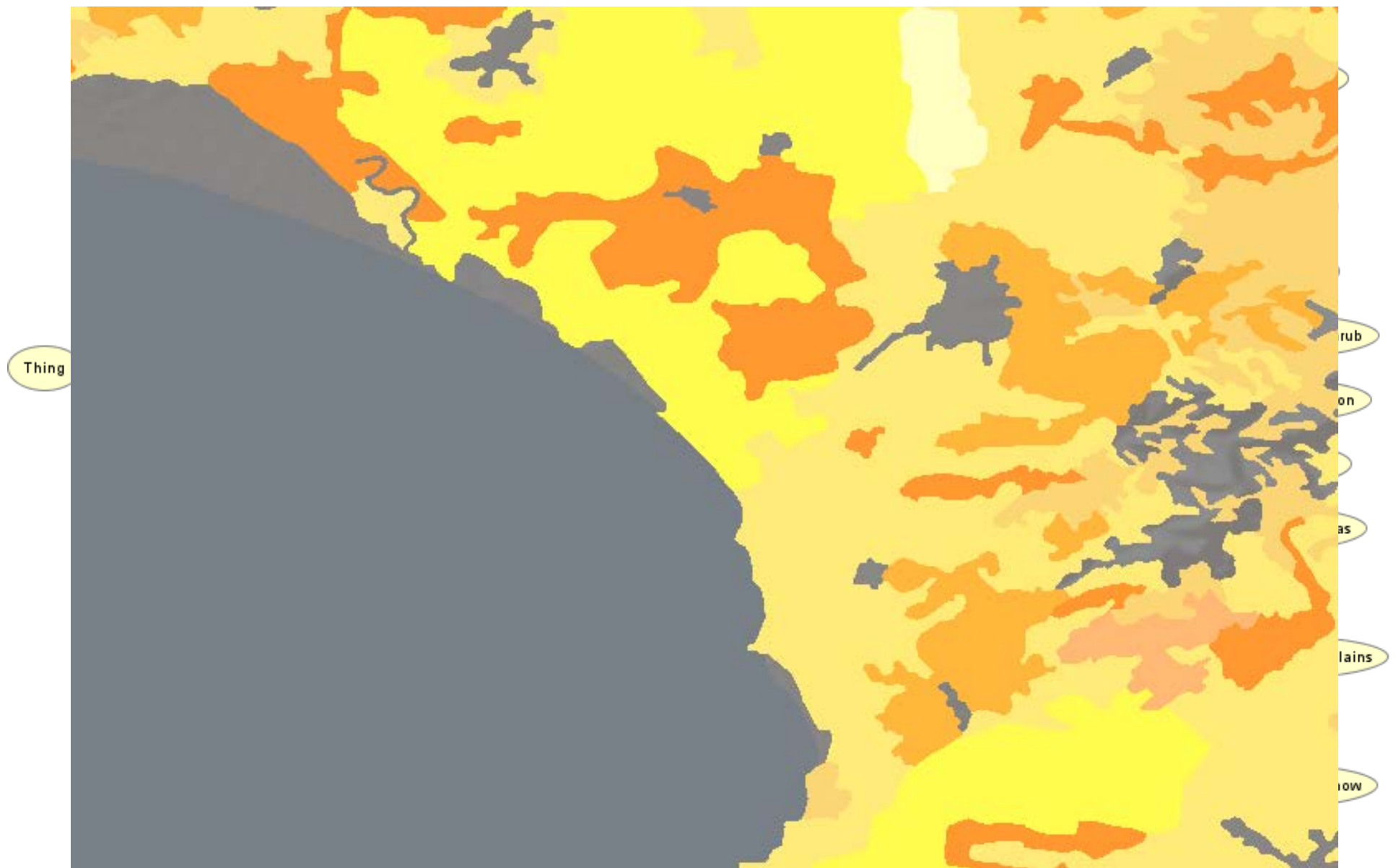


# Corine Land Use / Land Cover



# Corine Land Use / Land Cover

---

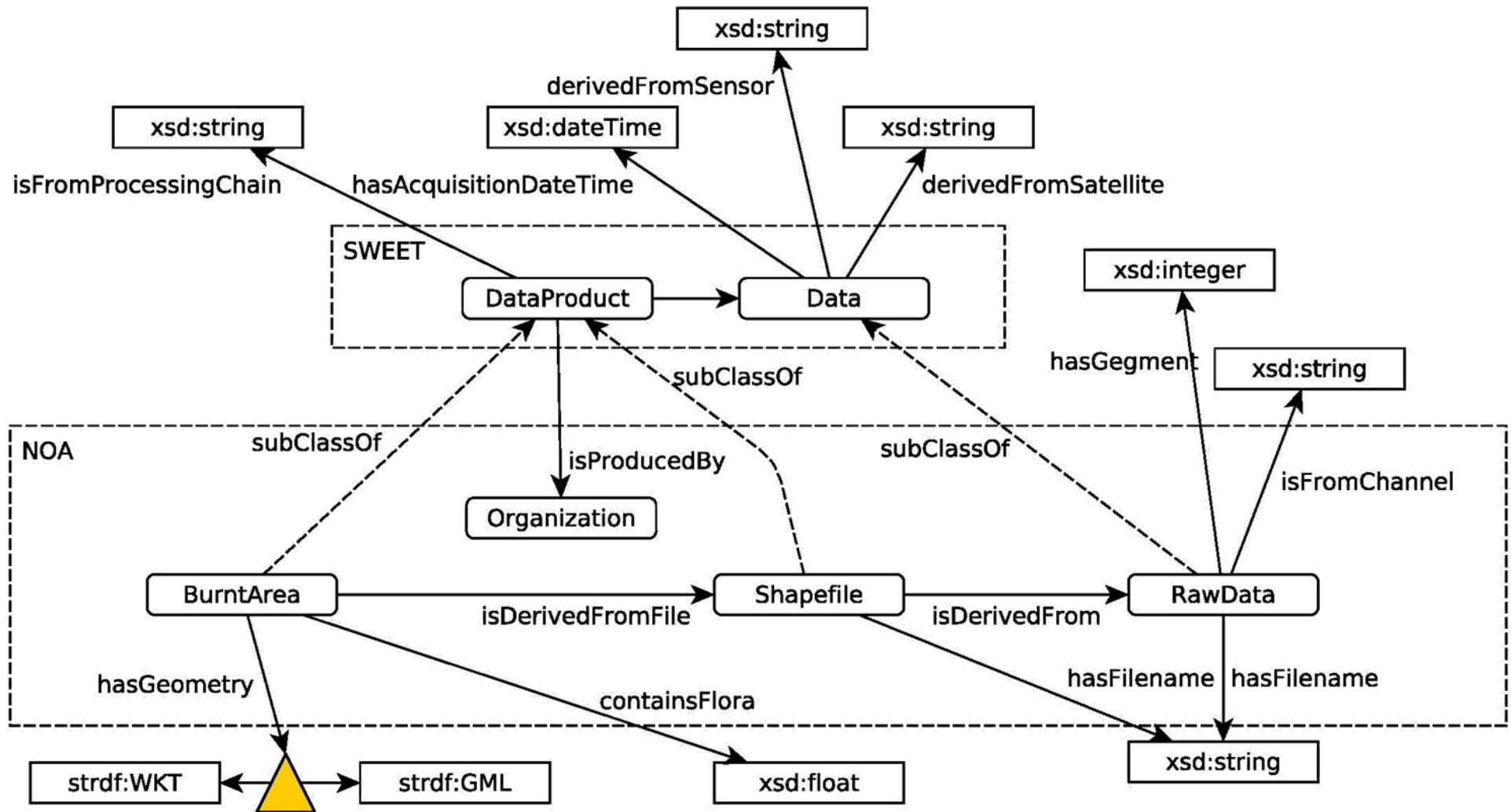


# Corine Land Use / Land Cover

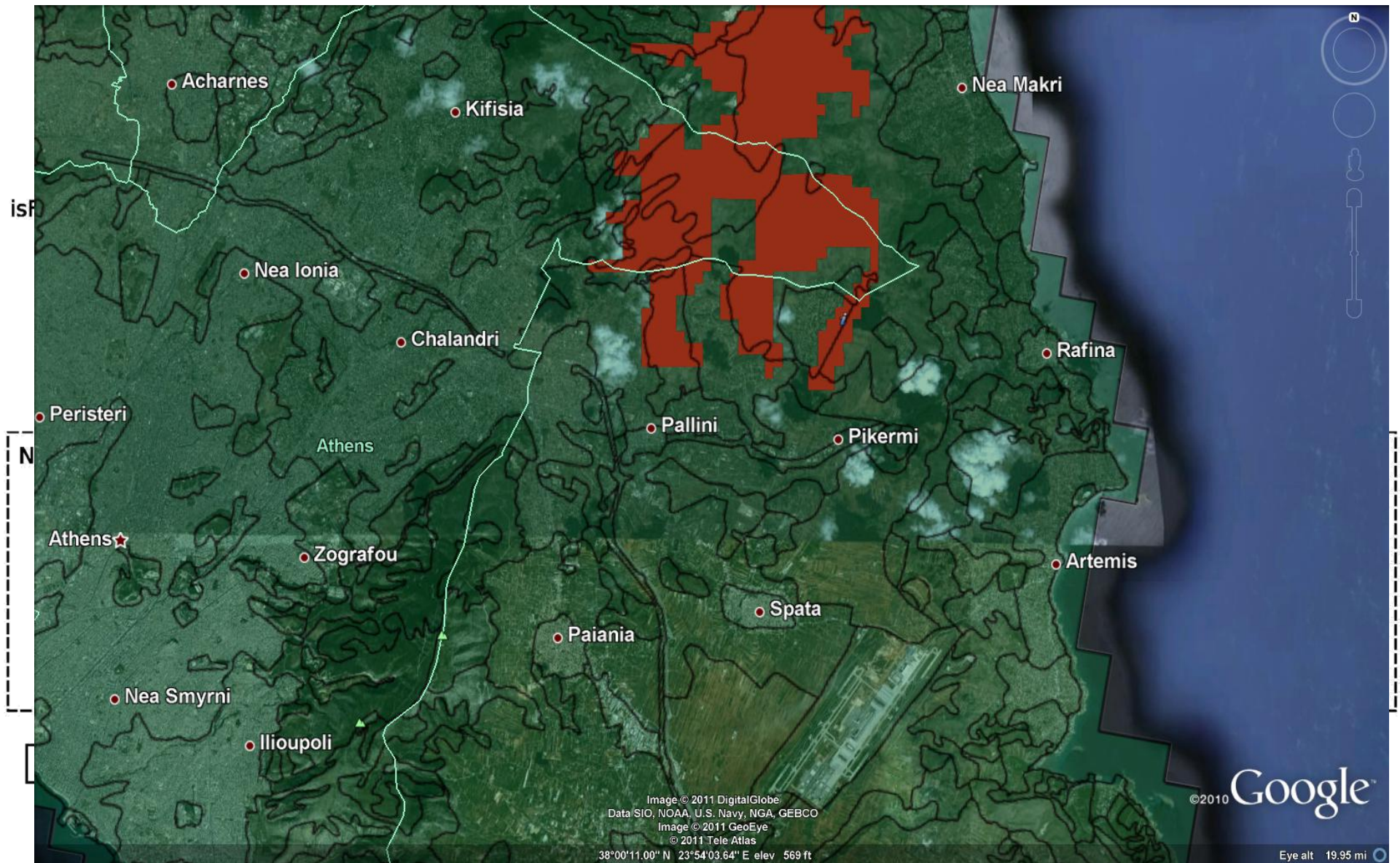
---

```
noa:Area_24015134
  rdf:type noa:Area ;
  noa:hasCode "312"^^xsd:decimal;
  noa:hasID "EU-203497"^^xsd:string;
  noa:hasArea_ha "255.5807904"^^xsd:double;
  strdf:hasGeometry "POLYGON((15.53 62.54,
                                ...))"^^strdf:WKT;
  noa:hasLandUse noa:ConiferousForest
```

# Burnt Area Products



# Burnt Area Products



# Burnt Area Products

---

```
noa:ba_15
  rdf:type noa:BurntArea;
  noa:isProducedByProcessingChain
    "static thresholds"^^xsd:string;
  noa:hasAcquisitionTime
    "2010-08-24T13:00:00"^^xsd:dateTime;

  strdf:hasGeometry "MULTIPOLYGON(((
393801.42 4198827.92, ..., 393008 424131)))";
  <http://www.opengis.net/def/crs/
    EPSG/0/2100>"^^strdf:WKT.
```

# 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
  - `xsd:int strdf:Dimension(strdf:geometry A)`
  - `xsd:string strdf:GeometryType(strdf:geometry A)`
  - `xsd:int strdf:SRID(strdf:geometry A)`
- Get the desired representation of a geometry
  - `xsd:string strdf:AsText(strdf:geometry A)`
  - `strdf:wkb strdf:AsBinary(strdf:geometry A)`
  - `xsd:string strdf:AsGML(strdf:geometry A)`
- Test whether a certain condition holds
  - `xsd:boolean strdf:IsEmpty(strdf:geometry A)`
  - `xsd:boolean strdf:IsSimple(strdf:geometry A)`

# stSPARQL: Geospatial SPARQL 1.1

---

## Functions for testing topological spatial relationships

- **OGC Simple Features Access**

```
xsd:boolean strdf:Equals(strdf:geometry A, strdf:geometry B)
xsd:boolean strdf:Disjoint(strdf:geometry A, strdf:geometry B)
xsd:boolean strdf:Intersects(strdf:geometry A, strdf:geometry B)
xsd:boolean strdf:Touches(strdf:geometry A, strdf:geometry B)
xsd:boolean strdf:Crosses(strdf:geometry A, strdf:geometry B)
xsd:boolean strdf:Within(strdf:geometry A, strdf:geometry B)
xsd:boolean strdf:Contains(strdf:geometry A, strdf:geometry B)
xsd:boolean strdf:Overlaps(strdf:geometry A, strdf:geometry B)
```

```
xsd:boolean strdf:Relate(strdf:geometry A, strdf:geometry B,
                        xsd:string intersectionPatternMatrix)
```

- **Egenhofer**

- **RCC-8**

# stSPARQL: Geospatial SPARQL 1.1

---

## Spatial analysis functions

- **Construct new geometric objects from existing geometric objects**

```
strdf:geometry strdf:Boundary(strdf:geometry A)
strdf:geometry strdf:Envelope(strdf:geometry A)
strdf:geometry strdf:Intersection(strdf:geometry A, strdf:geometry B)
strdf:geometry strdf:Union(strdf:geometry A, strdf:geometry B)
strdf:geometry strdf:Difference(strdf:geometry A, strdf:geometry B)
strdf:geometry strdf:SymDifference(strdf:geometry A, strdf:geometry B)
strdf:geometry strdf:Buffer(strdf:geometry A, xsd:double distance)
```

- **Spatial metric functions**

```
xsd:float strdf:distance(strdf:geometry A, strdf:geometry B)
xsd:float strdf:area(strdf:geometry A)
```

- **Spatial aggregate functions**

```
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)
```

# 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

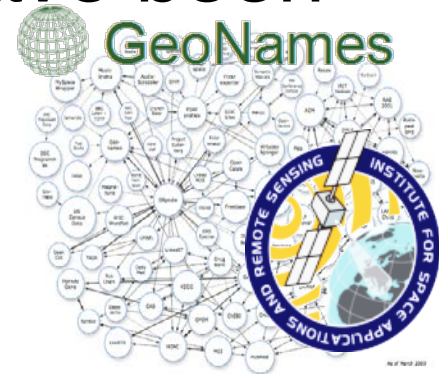
- Functions for testing topological spatial relationships between spatial terms (e.g., `strdf:contains(?G1, strdf:union(?G2, ?G3))`)
- Numeric expressions involving spatial metric functions (e.g., `strdf:area(?G1) ≤ 2*strdf:area(?G2)+1`)
- Boolean combinations

## 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`)

# stSPARQL: An example (1 / 3)

Return the names of communities that have been affected by fires



```
SELECT    ?name
WHERE {
```

```
    ?community rdf:type dbpedia:Community;
                geonames:name ?name;
                strdf:hasGeometry ?comGeom.
```

```
    ?ba  rdf:type noa:BurntArea;
          strdf:hasGeometry ?baGeom.
```

```
    FILTER(strdf:overlap(?comGeom, ?baGeom))
}
```

Spatial  
Function

# stSPARQL: An example (2 / 3)

Find all burnt forests near communities

```
SELECT ?ba ?baGeom  
WHERE {
```

```
  ?r rdf:type noa:Region;  
    strdf:geometry ?rGeom;  
    noa:hasCorineLandCoverUse ?f.  
  ?f rdfs:subClassOf clc:Forests.
```

```
  ?c rdf:type dbpedia:Community;  
    strdf:geometry ?cGeom.
```

```
  ?ba rdf:type noa:BurntArea;  
    strdf:geometry ?baGeom.
```

```
  FILTER( strdf:intersects(?rGeom, ?baGeom) &&  
    strdf:distance(?baGeom, ?cGeom) < 0.02) }
```

Spatial  
Functions



# stSPARQL: An example 3 / 3)

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

```
SELECT ?burntArea  
(strdf:intersection(?baGeom,  
strdf:union(?fGeom))  
AS ?burntForest)
```

Spatial  
Aggregate



```
WHERE {
```

```
?burntArea    rdf:type    noa:BurntArea;  
               strdf:hasGeometry ?baGeom.
```

```
?forest rdf:type noa:Region;  
        noa:hasLandCover noa:coniferousForest;  
        strdf:hasGeometry ?fGeom.
```

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

```
GROUP BY ?burntArea ?baGeom
```

Spatial  
Function

# Conclusions

---

- **Geospatial data in the Semantic Web - stSPARQL**
  - Early works
  - The data model stRDF
  - Examples of publicly available linked geospatial data
  - The query language stSPARQL
  
- **Next topic:** Geospatial data in RDF - GeoSPARQL

# Bibliography

---

[Kolas and Self, 2007]

Kolas, D., Self, T.: *Spatially Augmented Knowledgebase*. In: Proceedings of the 6th International Semantic Web Conference and 2nd Asian Semantic Web Conference (ISWC/ASWC2007). Lecture Notes in Computer Science, vol. 4825, pp. 785-794. Springer Verlag (2007)

[Perry, 2008]

Perry, M.: *A Framework to Support Spatial, Temporal and Thematic Analytics over Semantic Web Data*. Ph.D. thesis, Wright State University (2008)

[Koubarakis and Kyzirakos, 2010]

Koubarakis, M., Kyzirakos, K.: *Modeling and Querying Metadata in the Semantic Sensor Web: The Model stRDF and the Query Language stSPARQL*. In: ESWC. pp. 425-439 (2010)