GISE HUB | Geospatial Information Science & Engineering

OGC Stack Winter School

15–31 Dec, 2022

Venue: IIIT Hyderabad

INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY
HYDERABAD

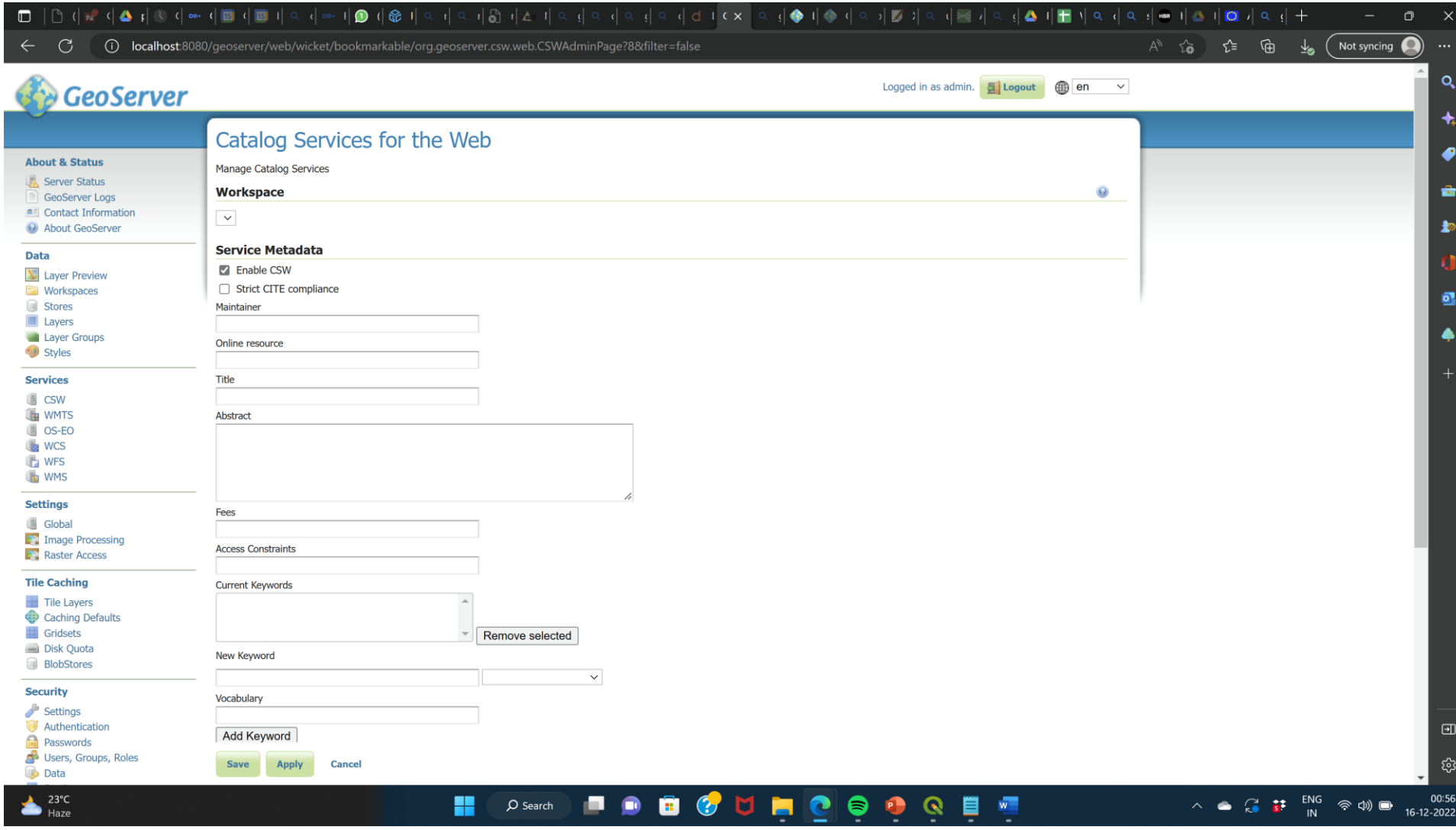
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
ज्ञानम् परमम् ध्येयम्

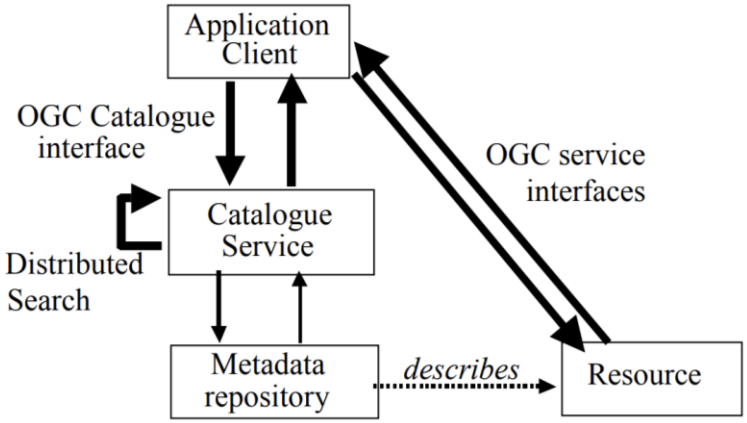
IIT Tirupati Navavishkār I-Hub Foundation

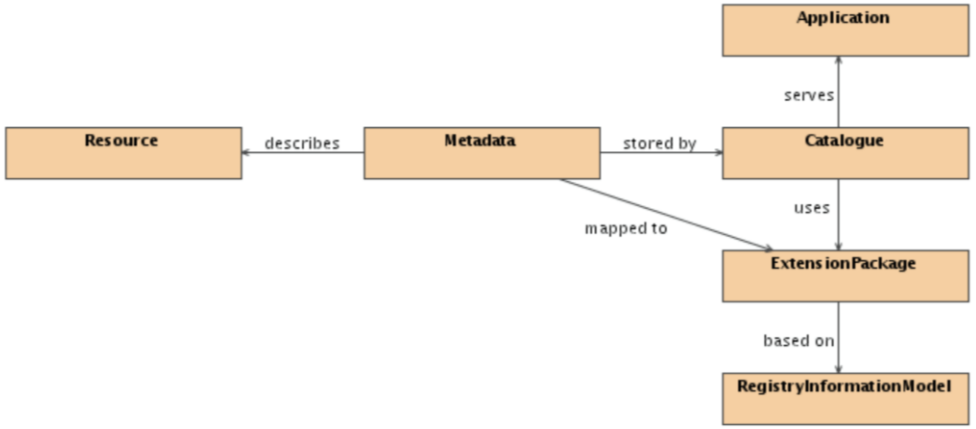
Today

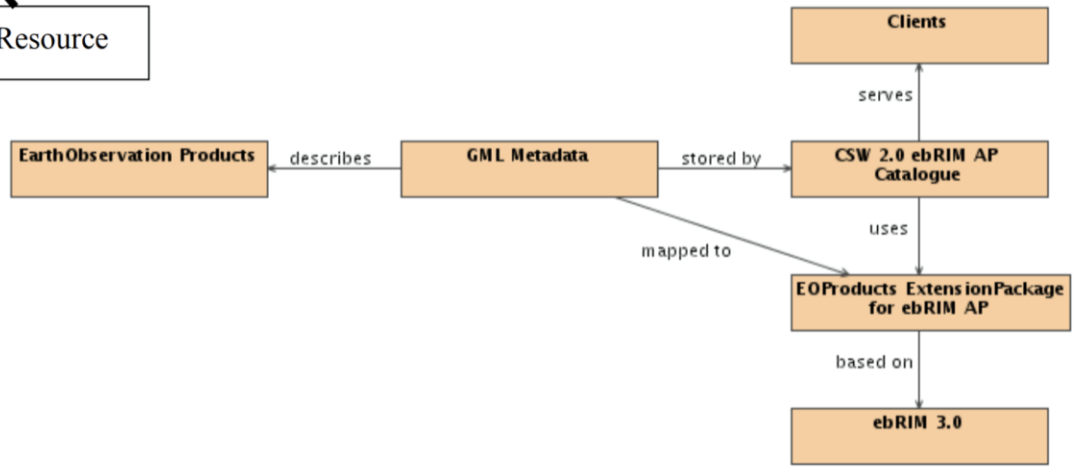
0930-1000	Recap	Participants			
1000-1130	Catalog Services and Publsihing	Sumit Sen	Slides from Sumit		
1130-1145	Tea				
1145-1300	Open Search and GeoSPARQL	Sumit Sen	Slides from Sumit	GeoSPARQL	
1300-1400	Lunch	-	-		
1400-1530	GeoRSS	Sumit Sen	Slides from Sumit		
1530-1545	Tea				
1545-1630	Try Out GeoRSS				
1630-1730	OGC Standards FAIR	Scott Simmons	Online Talk	meet.google.com/fou-wqng-fid	

CSW

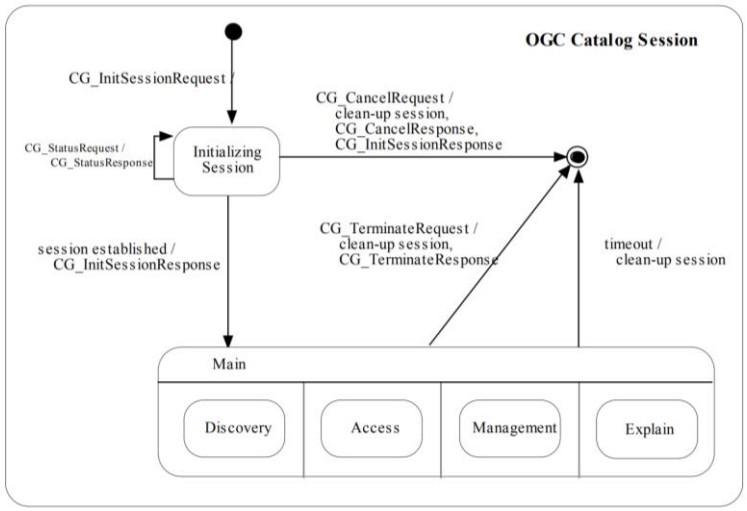

localhost:8080/geoserver/web/wicket/bookmarkable/org.geoserver.csw.web.CSWAdminPage?8&filter=false
Not syncing
GeoServer
Logged in as admin.
Logout
en
About & Status
Server Status
GeoServer Logs
Contact Information
About GeoServer
Data
Layer Preview
Workspaces
Stores
Layers
Layer Groups
Styles
Services
CSW
WMTS
OS-EO
WCS
WFS
WMS
Settings
Global
Image Processing
Raster Access
Tile Caching
Tile Layers
Caching Defaults
Gridsets
Disk Quota
BlobStores
Security
Settings
Authentication
Passwords
Users, Groups, Roles
Data
Catalog Services for the Web
Manage Catalog Services
Workspace
Service Metadata
Enable CSW
Strict CITE compliance
Maintainer
Online resource
Title
Abstract
Fees
Access Constraints
Current Keywords
Remove selected
New Keyword
Vocabulary
Add Keyword
Save
Apply
Cancel
23°C
Haze
Search
ENG
IN
00:56
16-12-2022

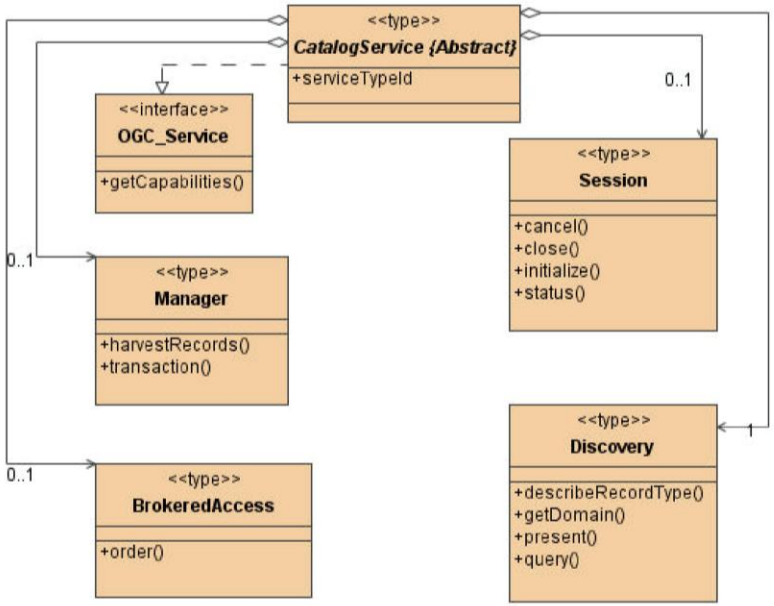






CSW Reference model architecture


OGC Catalog Session
CG_InitSessionRequest /
CG_CancelRequest /
clean-up session,
CG_CancelResponse,
CG_InitSessionResponse
CG_StatusRequest /
CG_StatusResponse
Initializing
Session
session established /
CG_InitSessionResponse
CG_TerminateRequest /
clean-up session,
CG_TerminateResponse
timeout /
clean-up session
Main
Discovery
Access
Management
Explain


<<type>>
CatalogService {Abstract}
+serviceTypeId
0..1
<<interface>>
OGC_Service
+getCapabilities()
<<type>>
Session
+cancel()
+close()
+initialize()
+status()
0..1
<<type>>
Manager
+harvestRecords()
+transaction()
0..1
<<type>>
BrokeredAccess
+order()
<<type>>
Discovery
+describeRecordType()
+getDomain()
+present()
+query()
1

OpenGIS® Catalogue Service Specification

- Document number 07-006r1
- Current version is 2.0.2
- CS/W is a protocol binding defined in 07-006r1 (others include Z39.50 and CORBA)
- CS/W uses HTTP as the distributed computing platform (DCP)
  - The basic interaction model is request/response
  - Asynchronous requests are also supported
- The CS/W API is patterned after the Web Feature Service (WFS)

OpenGIS® Catalogue Service Specification

- Query languages
  - Supports the Filter Encoding Specification for specifying query predicates (with extensions)
  - Supports Common Query Language (CQL); similar to SQL where clause
  - Profiles may define other languages (e.g. XPath)

Components of CS/W

- Defines a standard API for Creating, Updating, Deleting and Querying catalogue records
  - API can be implemented on top of existing servers as well (Z39.50)
- Service requests may be encoded in XML or as Keyword-Value pair
- The specification support HTTP method POST and GET and describes how to use SOAP (basically message literal)
- A specific information model is not defined (i.e. agnostic)

Components of CS/W (Cont.)

- It is expected that profiles will be defined to support specific catalogue information model
  - Current AP's include: FGDC, ISO19119/ISO19115 and ebRIM
  - OGC has recently decided to make ebRIM the preferred catalogue information model ... but more about that in the next presentation
- The OGC® Catalogue Service Specification defines a set of common queryables and returnables
  - The intent is to support CROSS-PROFILE query interoperability
  - All CS/W implementation must support the core queryables / returnables
  - Typically referred to as csw:Record

CS/W Operations

- The following operations are defined for the CS/W:
  - **GetCapabilities**: provides service metadata
  - **DescribeRecord**: allows clients to get a schema description of the catalogue's IM
  - **GetDomain**: allows clients to discover the runtime value space for API parameters as well as other element within the IM
  - **GetRecords** : the primary method for querying the catalogue
    - Supports distributed query
  - **GetRecordById**: convenience request for getting records using their Id

CS/W Operations (con.t)

- **Transaction**: the primary method for creating, updating and deleting catalogue records (PUSH)
- **Harvest**: allows the catalogue service to retrieve web-accessible metadata and register it in the catalogue
  - Analogous to Transaction but performs a PULL rather that a PUSH
  - Supports periodic re-Harvesting of the resource

CS/W Operations - example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecords service="CSW" version="2.0.2" maxRecords="1000"
outputSchema="http://www.opengis.net/cat/csw/2.0.2" outputFormat="application/xml">
```

CS/W Operations - example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecords service="CSW" version="2.0.2" maxRecords="1000"
outputSchema="http://www.opengis.net/cat/csw/2.0.2" outputFormat="application/xml">
   <Query typeNames="csw:Record">
```

CS/W Operations - example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecords service="CSW" version="2.0.2" maxRecords="1000"
outputSchema="http://www.opengis.net/cat/csw/2.0.2" outputFormat="application/xml">
   <Query typeNames=”csw:Record”>
      <Constraint version="1.2.0”>
         <ogc:Filter>
```

OGC®

CS/W Operations - example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecords service="CSW" version="2.0.2" maxRecords="1000"
outputSchema="http://www.opengis.net/cat/csw/2.0.2" outputFormat="application/xml">
   <Query typeNames=”csw:Record”>
      <Constraint version="1.2.0”>
         <ogc:Filter>
            <ogc:And>
```

OGC®

CS/W Operations - example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecords service="CSW" version="2.0.2" maxRecords="1000"
outputSchema="http://www.opengis.net/cat/csw/2.0.2" outputFormat="application/xml">
   <Query typeNames=”csw:Record>
      <Constraint version="1.2.0”>
         <ogc:Filter>
            <ogc:And>
               <ogc:PropertyIsLike escape="" singleChar="_" wildCard="%">
                  <ogc:PropertyName>/Record/AnyText</ogc:PropertyName>
                  <ogc:Literal>%seismic%</ogc:Literal>
               </ogc:PropertyIsLike>
```

CS/W Operations - example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecords service="CSW" version="2.0.2" maxRecords="1000"
outputSchema="http://www.opengis.net/cat/csw/2.0.2" outputFormat="application/xml">
   <Query typeNames=”csw:Record>
      <Constraint version="1.2.0”>
         <ogc:Filter>
            <ogc:And>
               <ogc:PropertyIsLike escape="" singleChar="_" wildCard="%">
                  <ogc:PropertyName>/Record/AnyText</ogc:PropertyName>
                  <ogc:Literal>%seismic%</ogc:Literal>
               </ogc:PropertyIsLike>
               <ogc:T_During>
                  <ogc:PropertyName>/Record/dct:modified</ogc:PropertyName>
                  <gml:TimePeriod>
                      <gml:begin>1976-01-01</gml:begin>
                      <gml:end>1976-12-31</gml:end>
                  <gml:TimePeriod>
               </ogc:T_During>
```

CS/W Operations - example

```
<ogc:Or>
   <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>/Record/dc:type</ogc:PropertyName>
      <ogc:Literal>urn::csw-ObjectType:FeatureType</ogc:Literal>
   </ogc:PropertyIsEqualTo>
   <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>/Record/dc:Type</ogc:PropertyName>
      <ogc:Literal>urn:csw-ObjectType:WMS_Layer</ogc:Literal>
   </ogc:PropertyIsEqualTo>
</ogc:Or>
```

CS/W Operations - example

```
<ogc:Or>
   <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>/Record/dc:type</ogc:PropertyName>
      <ogc:Literal>urn::csw-ObjectType:FeatureType</ogc:Literal>
   </ogc:PropertyIsEqualTo>
   <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>/Record/dc:Type</ogc:PropertyName>
      <ogc:Literal>urn:csw-ObjectType:WMS_Layer</ogc:Literal>
   </ogc:PropertyIsEqualTo>
</ogc:Or>
<ogc:classifiedAs scope="broad">
   <ogc:TypeName>Record</ogc:TypeName>
   <ogc:Scheme>urn:gcmd</ogc:Scheme>
   <ogc:Node>urn:gcmd:oceans:seafloor:topography</ogc:Node>
</ogc:classifiedAs>
```

CS/W Operations - example

```
<ogc:Or>
   <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>/Record/dc:type</ogc:PropertyName>
      <ogc:Literal>urn::csw-ObjectType:FeatureType</ogc:Literal>
   </ogc:PropertyIsEqualTo>
   <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>/Record/dc:Type</ogc:PropertyName>
      <ogc:Literal>urn:csw-ObjectType:WMS_Layer</ogc:Literal>
   </ogc:PropertyIsEqualTo>
</ogc:Or>
<ogc:classifiedAs scope="broad">
   <ogc:TypeName>Record</ogc:TypeName>
   <ogc:Scheme>urn:gcmd</ogc:Scheme>
   <ogc:Node>urn:gcmd:oceans:seafloor:topography</ogc:Node>
</ogc:classifiedAs>
<ogc:Contains>
   <ogc:PropertyName>/Record/ows:BoundingBox</ogc:PropertyName>
   <gml:Envelope srsName="EPSG:4326">
      <gml:lowerCorner>-80,30</gml:lowerCorner>
      <gml:upperCorner>-70,40</gml:upperCorner>
   </gml:Envelope>
</ogc:Contains>
```

CS/W Operations - example

```
            </ogc:And>
          </ogc:Filter>
        </Constraint>
    </Query>
</GetRecords>
```

CS/W Operations - example

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecords
  xmlns="http://www.opengis.net/cat/csw/2.0.2"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc"
  service="CSW"
  version="2.0.2"
  maxRecords="1000"
  outputSchema="http://www.opengis.net/cat/csw/2.0.2"
  outputFormat="application/xml">
  <Query typeNames="Record">
    <Constraint version="1.2.0">
      <ogc:Filter>
        <ogc:And>
          <ogc:PropertyIsLike escape="" singleChar="_" wildCard="%">
            <ogc:PropertyName>/Record/AnyText</ogc:PropertyName>
            <ogc:Literal>%seismic%</ogc:Literal>
          </ogc:PropertyIsLike>
          <ogc:T_During>
            <ogc:PropertyName>/Record/dct:modified</ogc:PropertyName>
            <gml:TimePeriod>
              <gml:begin>1976-01-01</gml:begin>
              <gml:end>1976-12-31</gml:end>
            <gml:TimePeriod>
          </ogc:T_During>
          <ogc:Or>
            <ogc:PropertyIsEqualTo>
              <ogc:PropertyName>/Record/dc:type</ogc:PropertyName>
              <ogc:Literal>urn:x-ogc:specification:csw-
ebrim:ObjectType:FeatureType</ogc:Literal>
            </ogc:PropertyIsEqualTo>
            <ogc:PropertyIsEqualTo>
              <ogc:PropertyName>/Record/dc:Type</ogc:PropertyName>
              <ogc:Literal>urn:x-ogc:specification:csw-
ebrim:ObjectType:WMS_Layer</ogc:Literal>
            </ogc:PropertyIsEqualTo>
          </ogc:Or>
          <ogc:classifiedAs scope="broad">
            <ogc:TypeName>Record</ogc:TypeName>
            <ogc:Scheme>urn:gcmd</ogc:Scheme>
            <ogc:Node>urn:gcmd:oceans:seafloor:topography</ogc:Node>
          </ogc:classifiedAs>
          <ogc:Contains>
            <ogc:PropertyName>/Record/ows:BoundingBox</ogc:PropertyName>
            <gml:Envelope srsName="EPSG:4326">
              <gml:lowerCorner>-80,30</gml:lowerCorner>
              <gml:upperCorner>-70,40</gml:upperCorner>
            </gml:Envelope>
          </ogc:Contains>
        </ogc:And>
      </ogc:Filter>
    </Constraint>
  </Query>
</GetRecords>
```

OGC Core Queryables

Name	Definition	Data type
Subject [a]	The topic of the content of the resource [b]	CharacterString
Title [a]	A name given to the resource	CharacterString
Abstract [a]	A summary of the content of the resource	CharacterString
AnyText	A target for full-text search of character data types in a catalogue	CharacterString
Format [a]	The physical or digital manifestation of the resource	CharacterString
Identifier [a]	An unique reference to the record within the catalogue	Identifier
Modified [c]	Date on which the record was created or updated within the catalogue	Date-8601
Type [a]	The nature or genre of the content of the resource. Type can include general categories, genres or aggregation levels of content.	CodeList [f]
BoundingBox [d]	A bounding box for identifying a geographic area of interest	BoundingBox, See Table 2
CRS [e]	Geographic Coordinate Reference System (Authority and ID) for the BoundingBox	Identifier
Association	Complete statement of a one-to-one relationship	Association, See Table 3

OGC Core Returnables

Dublin Core element name	Term used in OGC queryables	Definition	Data type
title	Title	A name given to the resource. Also known as "Name".	CharacterString
creator		An entity primarily responsible for making the content of the resource.	CharacterString
subject	Subject	A topic of the content of the resource. This is a place where a Topic Category or other taxonomy could be applied.	CharacterString
description	Abstract	An account of the content of the resource. This is also known as the "Abstract" in other aspects of OGC, FGDC, and ISO metadata.	CharacterString
publisher		An entity responsible for making the resource available. This would equate to the Distributor in ISO and FGDC metadata.	CharacterString
contributor		An entity responsible for making contributions to the content of the resource.	CharacterString
date	Modified	The date of a creation or update event of the catalogue record.	ISO-8601 date
type	Type	The nature or genre of the content of the resource.	CodeList
format	Format	The physical or digital manifestation of the resource.	CharacterString
identifier	Identifier	A unique reference to the record within the catalogue.	Identifier

OGC Core Returnables

source	Source	A reference to a resource from which the present resource is derived.	CharacterString
language		A language of the intellectual content of the catalogue record.	CharacterString
relation	Association	The name of the relationship that exists between the resource described by this record and a related resource referenced using the *Source* or *dc:source* property.	CodeList
coverage	BoundingBox	The spatial extent or scope of the content of the resource.	
rights		Information about rights held in and over the resource.	CharacterString

CS/W Materialization of Core Properties

- the CS/W protocol binding materializes the OGC Core queryables and returnables as an XML document based on a subset of the Dublin Core metadata elements with some extras
  - Extra #1: bounding box (csw:BoundingBox)
  - Extra #2: pseudo-property AnyText to be used as a target for full text searches
- Brief, Summary and Full element sets defined
  - **Brief**: dc:identifier, dc:title, dc:type, ows:BoundingBox
  - **Summary** Brief + dc:subject, dc:format, dc:relation, dct:modified,
  - dct:abstract, dct:spatial

OGC Core Properties - Example

```
<?xml version="1.0" encoding="ISO-8859-1"?>
  <Record
    xmlns="http://www.opengis.net/cat/csw/2.0.2"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:dct="http://purl.org/dc/terms/"
    xmlns:ows="http://www.opengis.net/ows"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/cat/csw/2.0.2 ../../../csw/2.0.2/record.xsd">
    <dc:identifier>00180e67-b7cf-40a3-861d-b3a09337b195</dc:identifier>
    <dc:title>Image2000 Product 1 (at1) Multispectral</dc:title>
    <dct:modified>2004-10-04 00:00:00</dct:modified>
    <dct:abstract>IMAGE2000 product 1 individual orthorectified scenes. IMAGE2000
was  produced from ETM+ Landsat 7 satellite data and provides a consistent European
coverage of individual orthorectified scenes in national map projection
systems.</dct:abstract>
```

OGC Core Properties – Example (cont.)

```
    <dc:type>dataset</dc:type>
    <dc:subject>imagery</dc:subject>
    <dc:subject>baseMaps</dc:subject>
    <dc:subject>earthCover</dc:subject>
    <dc:format>BIL</dc:format>
    <dc:creator>Vanda Lima</dc:creator>
    <dc:language>en</dc:language>
    <ows:WGS84BoundingBox>
      <ows:LowerCorner>14.05 46.46</ows:LowerCorner>
      <ows:UpperCorner>17.24 48.42</ows:UpperCorner>
    </ows:WGS84BoundingBox>
  </Record>
```

Issues

- No rules for re-Harvest
- Distributed query is not tested very well
- With the IM exposed, interoperability can be a problem in certain cases (e.g. ebRIM AP)
- Request complexity
- ???

Tea Beak

GeoSPARQL Overview

OGC®

Goals for GeoSPARQL

- Work within SPARQL's extensibility framework
- Simple enough for general users but capable enough for GIS professionals
- Accommodate systems based on qualitative spatial reasoning and systems based on quantitative geometries
- Don't re-invent the wheel!

ISO 13249 – SQL/MM

Simple Features
Well Known Text (WKT)
GML
KML
GeoJSON

A Simple Example

Example Data: OGC Sept 2011 TC is located in Boulder

```
:OGCSept2011TC :locatedIn      :Boulder .
:Boulder       :hasPointGeometry [
        ogc:asWKT "POINT(-105.17 40.1)"^^ogc:WKTLiteral ] .
```

Example Query: Find airports within 100 KM of Boulder

```
SELECT ?airport
WHERE { ?airport rdf:type :Airport .
        ?airport :hasPointGeometry [
                 ogc:asWKT ?aPointGeom ]
        FILTER(ogcf:distance(?aPointGeom,
                 "POINT(-105.17 40.1)"^^ogc:WKTLiteral,
                  ogc:km) <= 100) }
```

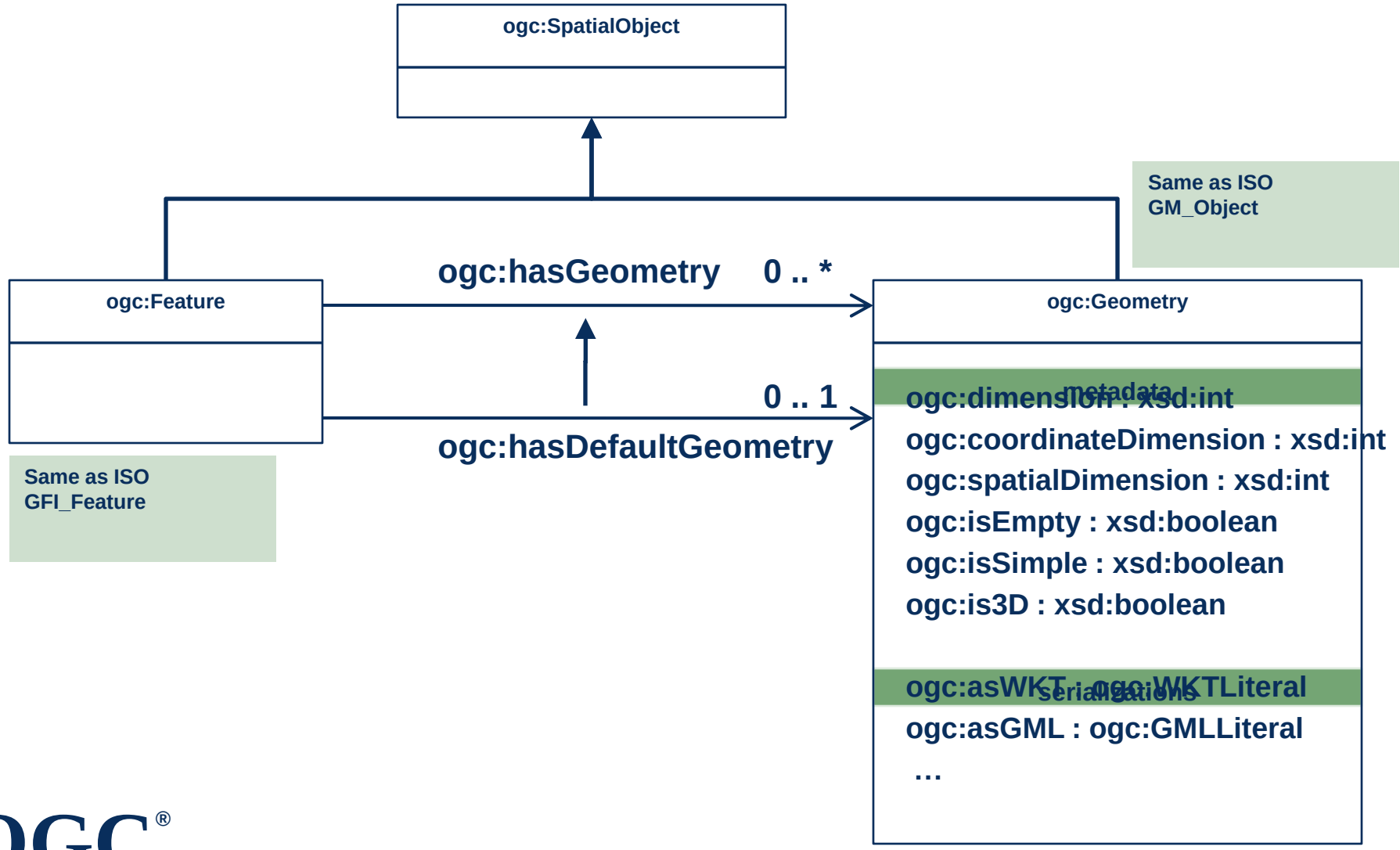
OGC®

What Does GeoSPARQL Give Us?

- Vocabulary for Query Patterns
  - Classes
    - Spatial Object, Feature, Geometry
  - Properties
    - Topological relations
    - Links between features and geometries
  - Datatypes for geometry literals
    - ogc:WKTLiteral, ogc:GMLLiteral
- Query Functions
  - Topological relations, distance, buffer, intersection, …
- Query Rewrite Rules
  - Expand feature-feature query into geometry query
  - Gives a common interface for qualitative and quantitative systems

GeoSPARQL Vocabulary

GeoSPARQL Vocabulary: Basic Classes and Relations



Why Encode Geometry Data as a Literal?

Advantage: single self-contained unit

Consistent way to select geometry information

Find all water bodies that are within 1 km of Route 3

```
SELECT  ?water ?wWKT
WHERE { ?water          rdf:type                            :WaterBody .
        ?water          :hasExactGeometry       ?wGeo .
        ?wGeo                       ogc:asWKT               ?wWKT .
        :Route_3        :hasExactGeometry       ?r3Geo .
        :r3Geo          ogc:asWKT               ?r3WKT .
        FILTER(ogcf:distance(?r3WKT, ?wWKT,…) <= 1000)
}
```

Consistent way to pass geometry information around

OGC®

Details of ogc:WKTLiteral

All RDFS Literals of type ogc:WKTLiteral shall consist of an optional IRI identifying the spatial reference system followed by Simple Features Well Known Text (WKT) describing a geometric value.

```
"<http://www.opengis.net/def/crs/OGC/1.3/CRS84>
 POINT(-122.4192 37.7793)"^^ogc:WKTLiteral
```

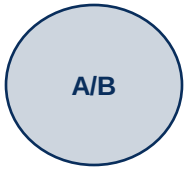
WGS84 longitude – latitude is the default CRS

```
"POINT(-122.4192 37.7793)"^^ogc:WKTLiteral
```

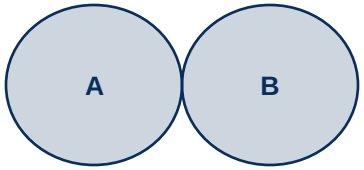
European Petroleum Survey Group (EPSG) maintains a set of CRS identifiers.

OGC®

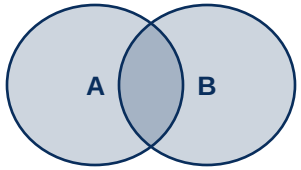
Topological Relations between ogc:SpatialObject



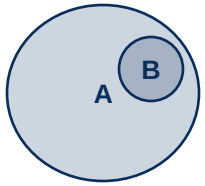
ogc:sfEquals



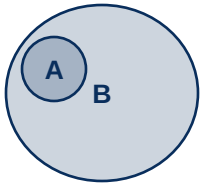
ogc:sfTouches



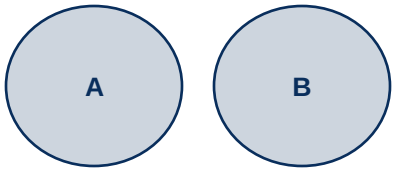
ogc:sfOverlaps



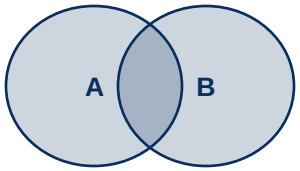
ogc:sfContains



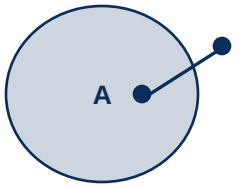
ogc:sfWithin



ogc:sfDisjoint



ogc:sfIntersects



ogc:sfCrosses

Assumes Simple Features
Relation Family

OGC®

RCC8, Egenhofer & Simple Features

Simple Features	Egenhofer	RCC8
equals	equal	EQ
disjoint	disjoint	DC
intersects	¬ disjoint	¬ DC
touches	meet	EC
within	inside+coveredBy	NTPP+TPP
contains	contains+covers	NTPPi+TPPi
overlaps	overlap	PO

Example Data

Meta Information

```
:City           rdfs:subClassOf    ogc:Feature .
:Park           rdfs:subClassOf    ogc:Feature .
:exactGeometry  rdfs:subPropertyOf ogc:hasGeometry .
```

Non-spatial Properties

```
:Nashua                 rdf:type    :City .
:MinesFallsPark         rdf:type    :Park .
:MinesFallsPark         :opened     "1950-03-01"^^xsd:date .
```

Spatial Properties

```
:MinesFallsPark :exactGeometry  :geo1 .
:geo1           rdf:type        ogc:Polygon  .
:geo1           ogc:asWKT       "Polygon((…))"^^ogc:WKTLiteral .

:Nashua         :exactGeometry  :geo2 .
:geo2           rdf:type        ogc:Polygon  .
:geo2           ogc:asWKT       "Polygon((…))"^^ogc:WKTLiteral .

:MinesFallsPark ogc:sfWithin    :Nashua .
```

OGC®

GeoSPARQL Query Functions

OGC®

Non-topological Query Functions

- `ogcf:distance(geom1: ogc:WKTLiteral, geom2: ogc:WKTLiteral, units: xsd:anyURI): xsd:double`
- `ogcf:buffer(geom: ogc:WKTLiteral, radius: xsd:double, units: xsd:anyURI): ogc:WKTLiteral`
- `ogcf:convexHull(geom: ogc:WKTLiteral): ogc:WKTLiteral`
- `ogcf:intersection(geom1: ogc:WKTLiteral, geom2: ogc:WKTLiteral): ogcf:WKTLiteral`
- `ogcf:union(geom1: ogc:WKTLiteral, geom2: ogc:WKTLiteral): ogc:WKTLiteral`
- `ogcf:difference(geom1: ogc:WKTLiteral, geom2: ogc:WKTLiteral): ogcf:WKTLiteral`
- `ogcf:symDifference(geom1: ogc:WKTLiteral, geom2: ogc:WKTLiteral): ogc:WKTLiteral`
- `ogcf:envelope(geom: ogc:WKTLiteral): ogcf:WKTLiteral`
- `ogcf:boundary(geom1: ogc:WKTLiteral): ogc:WKTLiteral`

Topological Query Functions

- `ogcf:relate(geom1: ogc:WKTLiteral, geom2: ogc:WKTLiteral, patternMatrix: xsd:string): xsd:boolean`
- `ogcf:sfEquals(geom1: ogc:WKTLiteral, geom2: ogcf:WKTLiteral): xsd:boolean`
- `ogcf:sfDisjoint(geom1: ogc:WKTLiteral, geom2: ogcf:WKTLiteral): xsd:boolean`
- `ogcf:sfIntersects(geom1: ogc:WKTLiteral, geom2: ogcf:WKTLiteral): xsd:boolean`
- `ogcf:sfTouches(geom1: ogc:WKTLiteral, geom2: ogcf:WKTLiteral): xsd:boolean`
- `ogcf:sfCrosses(geom1: ogc:WKTLiteral, geom2: ogcf:WKTLiteral): xsd:boolean`
- `ogcf:sfWithin(geom1: ogc:WKTLiteral, geom2: ogcf:WKTLiteral): xsd:boolean`
- `ogcf:sfContains(geom1: ogc:WKTLiteral, geom2: ogcf:WKTLiteral): xsd:boolean`
- `ogcf:sfOverlaps(geom1: ogc:WKTLiteral, geom2: ogcf:WKTLiteral): xsd:boolean`

Example Query

Find all land parcels that are within the intersection of :City1 and :District1

```
PREFIX : <http://my.com/appSchema#>
PREFIX ogc: <http://www.opengis.net/geosparql#>
PREFIX ogcf: <http://www.opengis.net/geosparql/functions#>
PREFIX epsg: <http://www.opengis.net/def/crs/EPSG/0/>

SELECT  ?parcel
WHERE { ?parcel      rdf:type         :Residential .
        ?parcel      :exactGeometry   ?pGeo .
        ?pGeo        ogc:asWKT        ?pWKT .
        :District1   :exactGeometry   ?dGeo .
        ?dGeo        ogc:asWKT        ?dWKT .
        :City1       :extent          ?cGeo .
        ?cGeo        ogc:asWKT        ?cWKT .
        FILTER(ogcf:sfWithin(?pWKT,
                 ogcf:intersection(?dWKT,?cWKT)))}
```

OGC®

Example Query

Find the three closest Mexican restaurants

```
PREFIX : <http://my.com/appSchema#>
PREFIX ogc: <http://www.opengis.net/geosparql#>
PREFIX ogcf: <http://www.opengis.net/geosparql/functions#>
PREFIX epsg: <http://www.opengis.net/def/crs/EPSG/0/>

SELECT  ?restaurant
WHERE { ?restaurant  rdf:type        :Restaurant .
        ?restaurant  :cuisine        :Mexican .
        ?restaurant  :pointGeometry  ?rGeo .
        ?rGeo        ogc:asWKT       ?rWKT }
ORDER BY ASC(ogcf:distance("POINT(…)"^^ogc:WKTLiteral,
                           ?rWKT, ogc:KM))
LIMIT 3
```

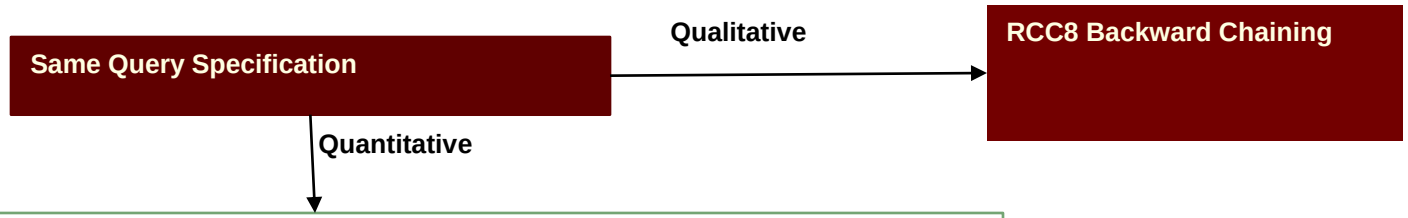
OGC®

GeoSPARQL Query Rewrite Rules

Motivation for Query Rewrite Rules

Find all water bodies within New Hampshire

```
SELECT ?water
WHERE { ?water  rdf:type      :WaterBody .
        ?water  ogc:rcc8Within    :NH }
```




```
SELECT ?water
WHERE { ?water  rdf:type      :WaterBody .
        ?water  ogc:hasDefaultGeometry          ?wGeo .
        ?wGeo   ogc:asWKT                       ?wWKT .
        :NH     ogc:hasDefaultGeometry          ?nGeo .
        ?nGeo   ogc:asWKT                       ?nWKT .
        FILTER(ogcf:rcc8Within(?wWKT, ?nWKT)) }
```

Query Rewrite

Specified with a RIF rule

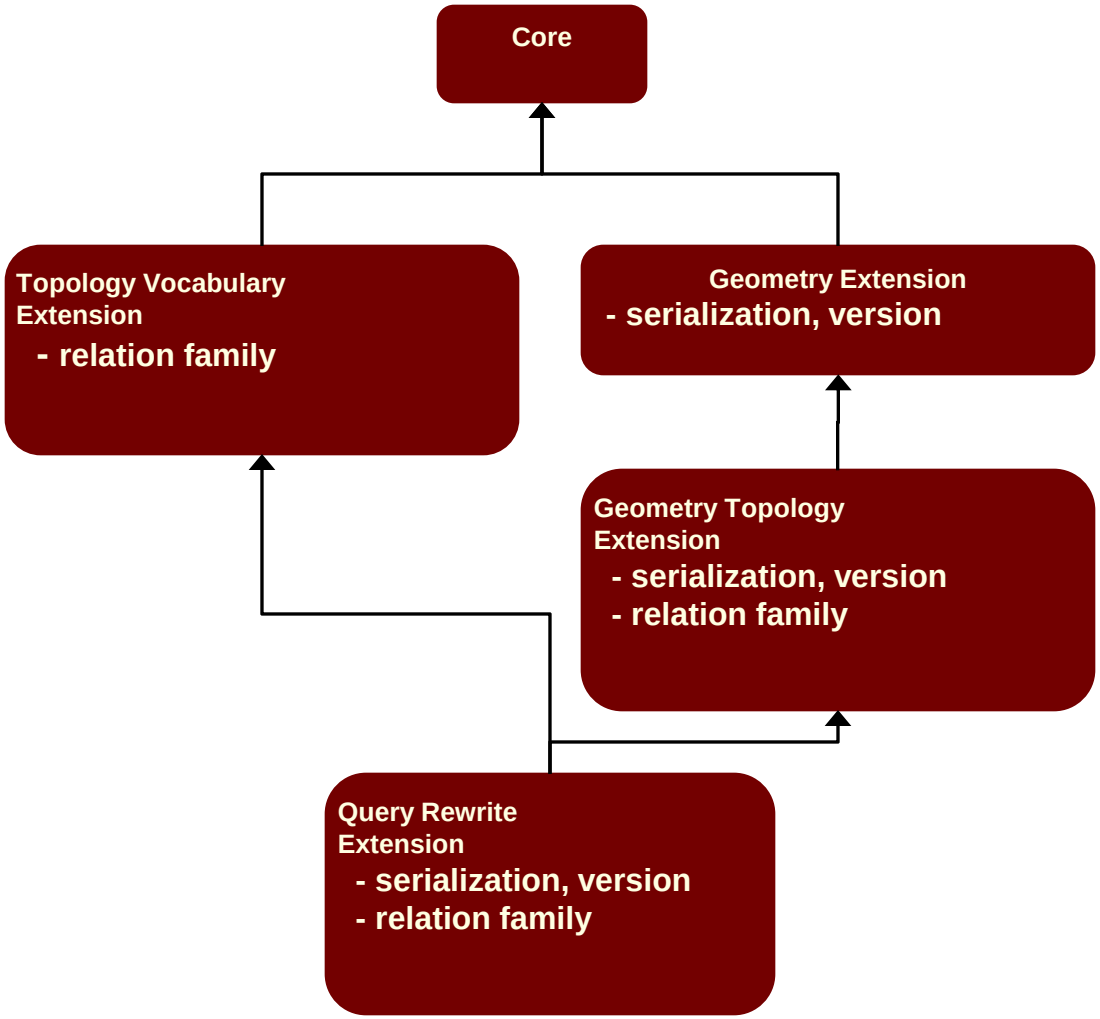
OGC®

Query Rewrite Rules

- Used to compute Feature-Feature spatial relations based on default geometries
- Specified as a collection of RIF rules
- Example: ogcr:sfEquals

```
(Forall ?f1 ?f2 ?g1 ?g2 ?g1Serial ?g2Serial
  (f1[ogc:sfEquals->?f2] :-
    And
      (?f1[ogc:hasDefaultGeometry->?g1]
       ?f2[ogc:hasDefaultGeometry->?g2]
       ?g1[ogc:asWKT->?g1Serial]
       ?g2[ogc:asWKT->?g2Serial]
       External(ogcf:sfEquals(?g1Serial,?g2Serial)))
  )
)
```

Conformance Classes



Parameters

- Serialization
  - *WKT*
  - *GML*

  Determines geometry classes and geometry literal datatype

- Relation Family
  - *Simple Features*
  - *RCC8*
  - *Egenhofer*

  Determines topology properties and topology functions

OGC®

Summary

- GeoSPARQL Defines:
  - Basic vocabulary, Query functions, Query rewrite rules
- Based on existing OGC/ISO standards
  - WKT, GML, Simple Features, ISO 19107
- Uses SPARQL's built-in extensibility framework
- Modular specification
  - Allows flexibility in implementations
  - Easy to extend
- Accommodates qualitative and quantitative systems
  - Same query specification for qualitative (core + topology vocabulary) and quantitative (all components, incl. query rewrite)

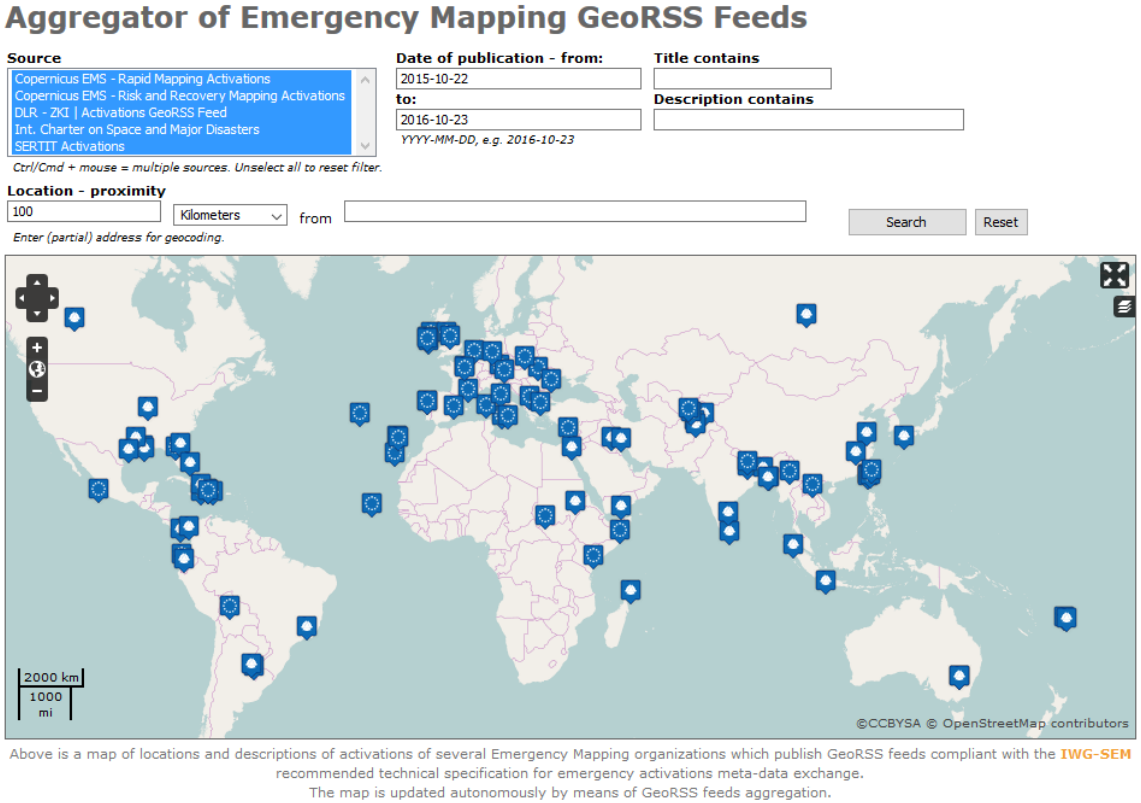
GeoSPARQL Demo

- GeoSPARQL
- Virtuoso GeoSPARQL Demo Server - Virtuoso Universal Server / RDF Data Management - OpenLink Software Community (openlinksw.com)
- GeoSPARQL queries on OSM Data in GraphDB (bobdc.com)

Lunch

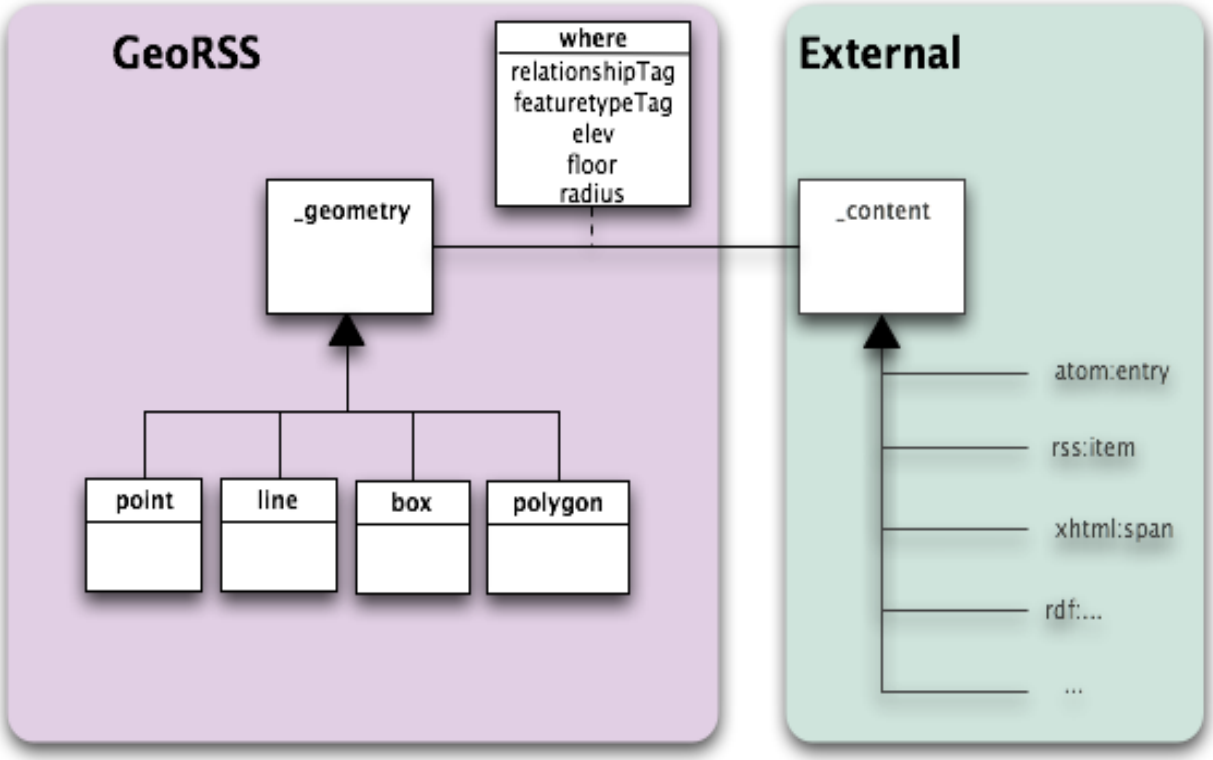
What is GeoRSS?

- GeoRSS is a simple proposal for geo-enabling, or tagging, "really simple syndication" (RSS) feeds with location information. GeoRSS standardizes the way in which "where" is encoded with enough simplicity and descriptive power to satisfy most needs to describe the location of Web content. It is extensible and upwardly-compatible with more sophisticated formats like the OGC GML



GeoRSS Information Model

- In the model, **where** is an association *of* a **geometry** to some **content**. GeoRSS places no constraint on the type of content, nor on its format**.**



Two types of GeoRSS

There are currently two encodings of GeoRSS:

- GeoRSS-Simple is meant as a very lightweight format that developers and users can quickly and easily add to their existing feeds with little effort. It supports basic geometries (point, line, box, polygon) and covers the typical use cases when encoding locations; and
- GeoRSS-GML is a formal GML Application Profile, and supports a greater range of features, notably coordinate reference systems other than latitude/longitude from the World Geodetic System of 1984 (WGS-84).

```
<georss:where>
    <gml:Point>
        <gml:pos>23.256 72.92</gml:pos>
    </gml:Point>
</georss:where>
```

```
<georss:point>23.256 72.92</georss:point>
```

Two types of GeoRSS

- Both GeoRSS Simple and GeoRSS GML (extended) use the same geometry and are therefore compatible. Both GeoRSS formats are designed for use with Atom 1.0, RSS 2.0 and RSS 1.0, although they can be used just as easily in non-RSS XML encodings.
- Perhaps the most powerful advantages of GeoRSS feeds will be seen in the possibilities for geographic search and aggregation. More than just getting feeds for a particular city or zip code, using GeoRSS it will be possible to search with all sorts of geographic criteria. Perhaps all earthquake-related items within a certain radius--delivered to your phone? Or a feed of traffic accident items along a regular daily commute? Once RSS feeds contain geographic location, applications like these and more are possible.

Evidence of implementation

- Mature: First released in 2006
- Hundreds of operational implementations
- Example implementations

Acme: https://acme.com/GeoRSS/about.html
British Geological Survey: http://www.bgs.ac.uk/data/services/georss.html
COPERNICUS: http://emergency.copernicus.eu/mapping/georss-feeds-aggregated#zoom=2&lat=17.34786&lon=33.48229&layers=B00T
Esri: https://doc.arcgis.com/en/arcgis-online/reference/georss.htm
GDAL: http://www.gdal.org/drv_georss.html
GeoNames: http://www.geonames.org/rss-to-georss-converter.html
Google: https://developers.google.com/maps/documentation/javascript/examples/layer-georss
MediaWiki: https://www.mediawiki.org/wiki/Extension:GeoRSS
OpenLayers: http://dev.openlayers.org/examples/georss.html
SAFE FME: https://knowledge.safe.com/articles/602/reading-and-writing-georss-and-rss-feeds.html
Yahoo!: https://developer.yahoo.com/maps/georss/
Worldkit: http://worldkit.org/doc/rss.php

Other information

- Publicly Available: www.georss.org
- License:
  - GeoRSS is currently licensed under a Creative Commons Attribution-ShareAlike 2.5 License. The OGC version shall be available under the same license agreement.
- OGC GeoRSS White Paper
  - http://portal.opengeospatial.org/files/?artifact_id=15755
- Justification Document
  - https://portal.opengeospatial.org/files/?artifact_id=71820&version=1

Demo

Home - International Disasters Charter

https://cgt.disasterscharter.org/en

https://emergency.copernicus.eu/mapping/georss-feeds-aggregated#zoom=2&lat=9.58815&lon=-10.9389&layers=TB00

Create your own RSS feed

- RSS Feed Generator, Create RSS feeds from URL
- GeoNames
- Create a GeoRSS

OpenSearch

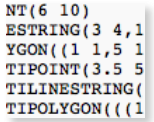
- OpenSearch started in an effort built around Amazon's A9.com and now maintained in a community process at opensearch.org
- Allows syndication of search results
- Search engines have a description document used by client applications

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/">
  <ShortName>Web Search</ShortName>
  <Description>Use Example.com to search the Web.</Description>
  <Tags>example web</Tags>
  <Contact>admin@example.com</Contact>
  <Url type="application/rss+xml"
       template="http://example.com/?q={searchTerms}&amp;pw={startPage?}&amp;format=rss"/>
</OpenSearchDescription>
```

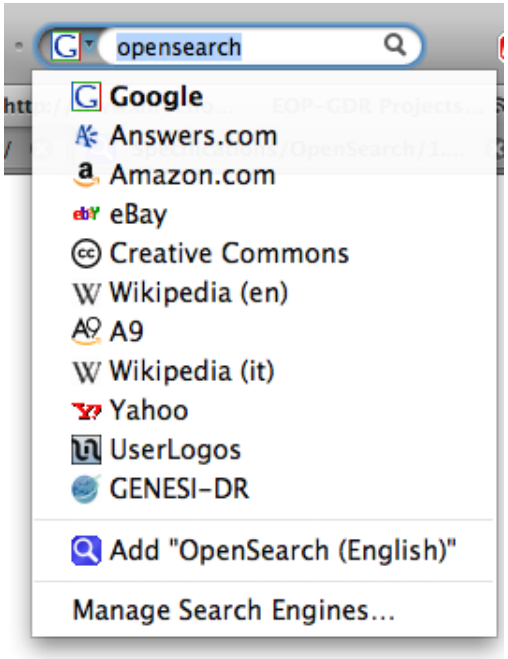
- OASIS Search Web Services group is publishing searchRetrieve Operation with Bindings for SRU 1.2 and OpenSearch http://docs.oasis-open.org/search-ws/

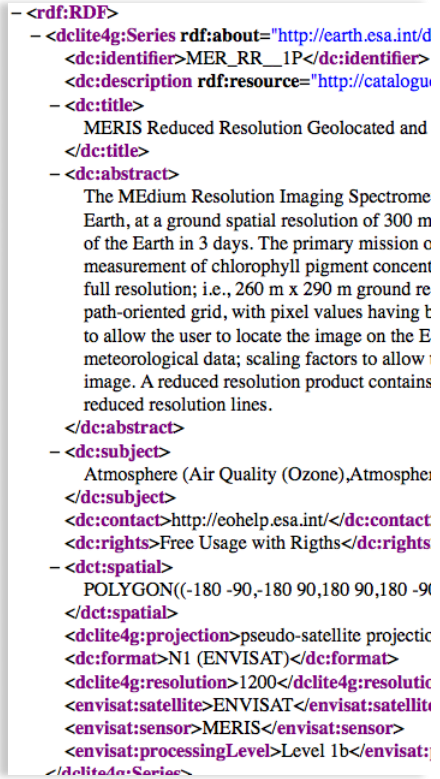
OpenSearch

- OpenSearch is flexible, results can be returned as HTML interfaces or in Atom, XML/RDF, KML, WKT, JSON….



- Directly integrated in Web Browsers
- Includes result's paging





OpenSearch Description

- Provides metadata about the contents along with a set of URL Templates which illustrate the parameters accepted by the service and the variety of output formats in which results can be obtained.
- The OpenSearch request interface is simple, consisting of a description of a HTTP GET request with a series of optional key-value parameters that can be used to constrain the search.

```
<Url type="text/html"
  template="http://example.com/?q={searchTerms?}&amp;pw={startPage
  ?}"/>
```

http://www.google.com/?q=question
http://www.google.com/?q={searchTerms}

Description XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
  <OpenSearchDescription xmlns="http://a9.com/-
  /spec/OpenSearch/1.1/">
        <ShortName>Web Search</ShortName>

        <Description>Use Acme.com to search the Web.</Description>
        <Tags>example web</Tags>
        <Contact>admin@acme.com</Contact>

        <Url type="application/rss+xml”
  template="http://acme.com/?q={searchTerms}&amp;pw={startPage?}&
  amp;format=rss"/>
  </OpenSearchDescription>
```

Description XML Document

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/OpenSearch/1.1/">
    <ShortName>Web Search</ShortName>
    <Description>Use Acme.com to search the Web.</Description>
    <Tags>example web</Tags>
    <Contact>admin@acme.com</Contact>
    <Url type="application/atom+xml”
    template="http://acme.com/?q={searchTerms}&amp;pw={startPage?}&amp;format=atom"/>
    <Url type="application/rss+xml”
    template="http://acme.com/?q={searchTerms}&amp;pw={startPage?}&amp;format=rss"/>
    <Url type="text/html” template="http://acme.com/?q={searchTerms}&amp;pw={startPage?}"/>
    <LongName>Example.com Web Search</LongName>
    <Image height="64" width="64" type="image/png">http://acme.com/websearch.png</Image>
    <Image height="16" width="16”
    type="image/vnd.microsoft.icon">http://acme.com/websearch.ico</Image>
    <Query role="example" searchTerms="cat" />
    <Developer>Acme.com Development Team</Developer>
    <Attribution>Search data Copyright 2005, Acme.com, Inc., All Rights Reserved</Attribution>
    <SyndicationRight>open</SyndicationRight>
    <AdultContent>false</AdultContent><Language>en-us</Language>
    <OutputEncoding>UTF-8</OutputEncoding><InputEncoding>UTF-8</InputEncoding>
</OpenSearchDescription>
```

GeoSpatial Extension

- Specify a series of parameters that can be used to spatially constrain search results.
- Provision is made to filter results by:
  - A bounding box
  - A geometry using Well Known Text
  - Within a certain radius from a given latitude-longitude point
  - Having a certain containment relation (within, overlaps, disjoint) with a geographic constraint
  - Matching a geographic name
- All geographic information is to be expressed using the EPSG 4326 (WGS84)
  - Following the GeoRSS “precedent”

GeoSpatial Extension Elements

- **geo:box** -> Geographic bounding box The box is defined by "west, south, east, north" coordinates of longitude, latitude, in a EPSG:4326 decimal degrees.
- **geo:geometry** -> Geographic area (geometry): The polygon is defined using the Well Known Text standard for geographic shapes, using EPSG:4326 (POINT, LINESTRING, POLYGON, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON)
- **geo:lat, geo:lon, geo:radius** -> The latitude, longitude of a given point with a given radius
- **geo:relation** -> Spatial relation to result set, Character String; One of “overlaps”, “contains”, “disjoint” (default is “overlaps”)
- **geo:name** -> A string describing the location to perform the search

The namespace of the OpenSearch GeoSpatial Extension is:
http://a9.com/-/opensearch/extensions/geo/1.0/

Time Extension Elements

- **time:start** - > Start of the temporal interval to search (RFC-3339)
- **time:end** -> End of the temporal interval to search (RFC-3339)

The namespace of the OpenSearch Time Extension is:
http://a9.com/-/opensearch/extensions/time/1.0/

Geo and Time Extension

- The geospatial extension allow to formulate geospatial requests e.g. point-plus-radius, a bounding box, or a polygon

bbox=**{geo:bbox?}**

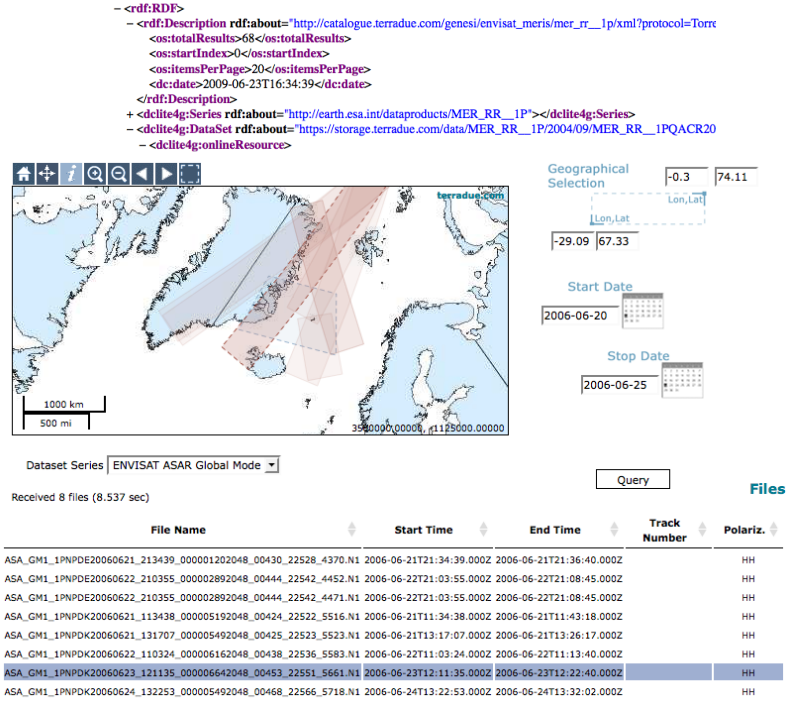
- Together with the Time extension, OpenSearch can specify time start, finish, and slices for searching data.

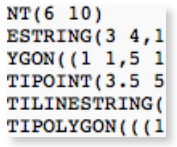
start=**{time:start?}**&stop=**{time:end?}**

<Url type="text/html”
template=”http://example.com/xml/?q={searchTerms?}&
start_date={time:start?}&stop_date={time:end?}&bbox={geo:
box}"/>

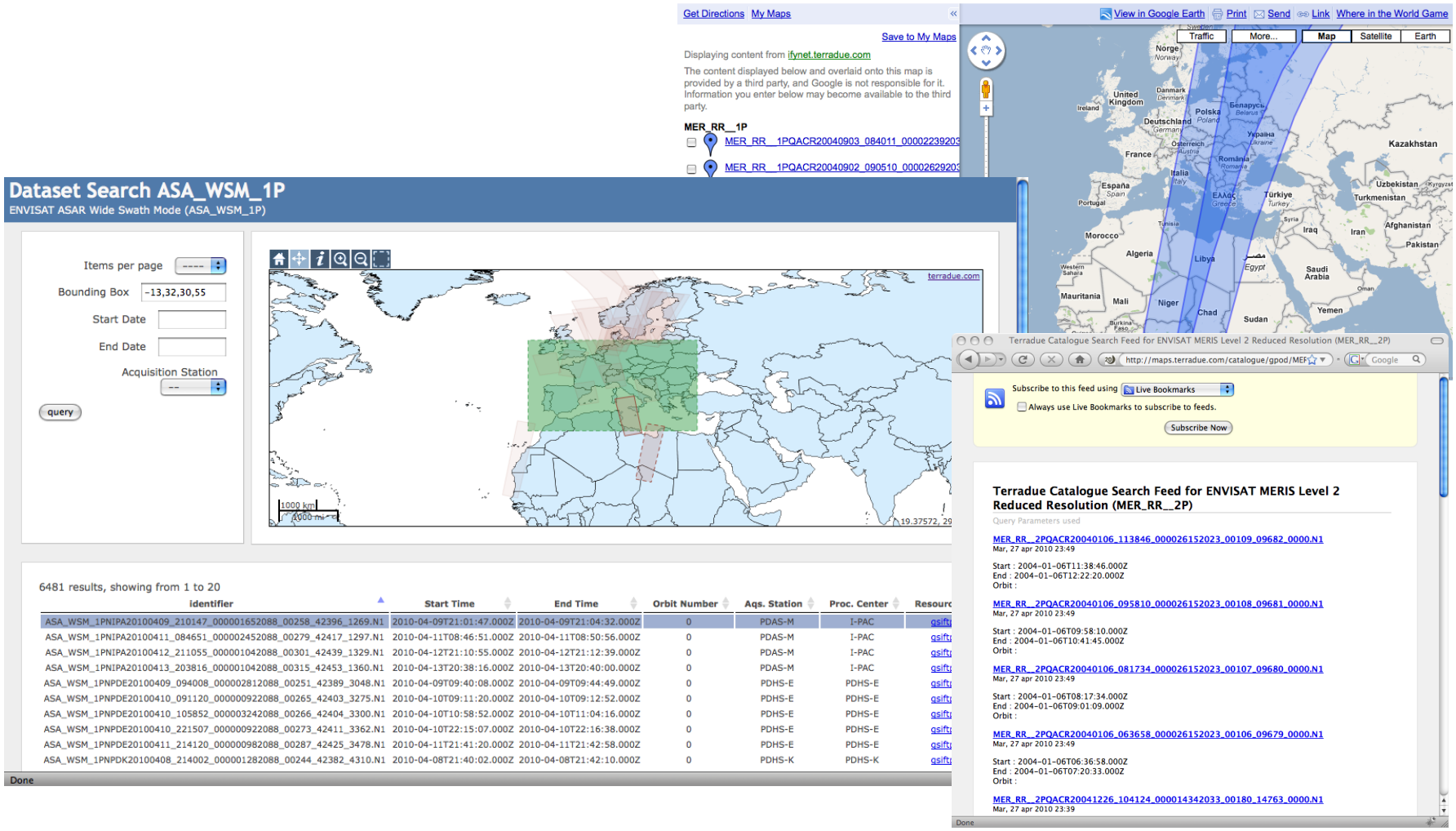
Integration

- REST approach allowing the query and registry of datasets
- Allows the easy definition of custom result types to ease the integration and creation of mashup applications
- Responses as HTML interfaces or in Atom, XML/RDF, KML, WKT, metalink, EOP/HMA

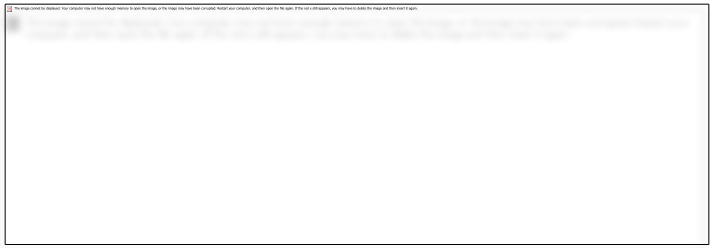




Integration using feeds



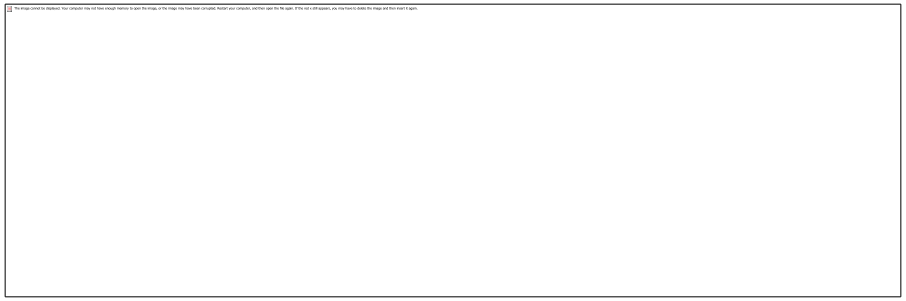
Experience from GENESI-DR



- OpenSearch with geo-temporal extensions provided the necessary flexibility to federate different Earth Science communities
- Search results can be in mass-market formats (e.g. KML, Atom) but also in community specific like the HMA/EOP
- Implementation of gateways demonstrated a future path to access community wide infrastructures (e.g. CNES's SIPAD)
- Follow-up in new GENESI-DEC project focusing on Digital Earth Communities enlarging the GENESI-DR infrastructure in terms of both resources availability and geographical extent

Adoption

- GENESI-DR
  - ESA, NILU, JRC, ENEA, DLR, CNES, INFOTERRA-UK, K-SAT



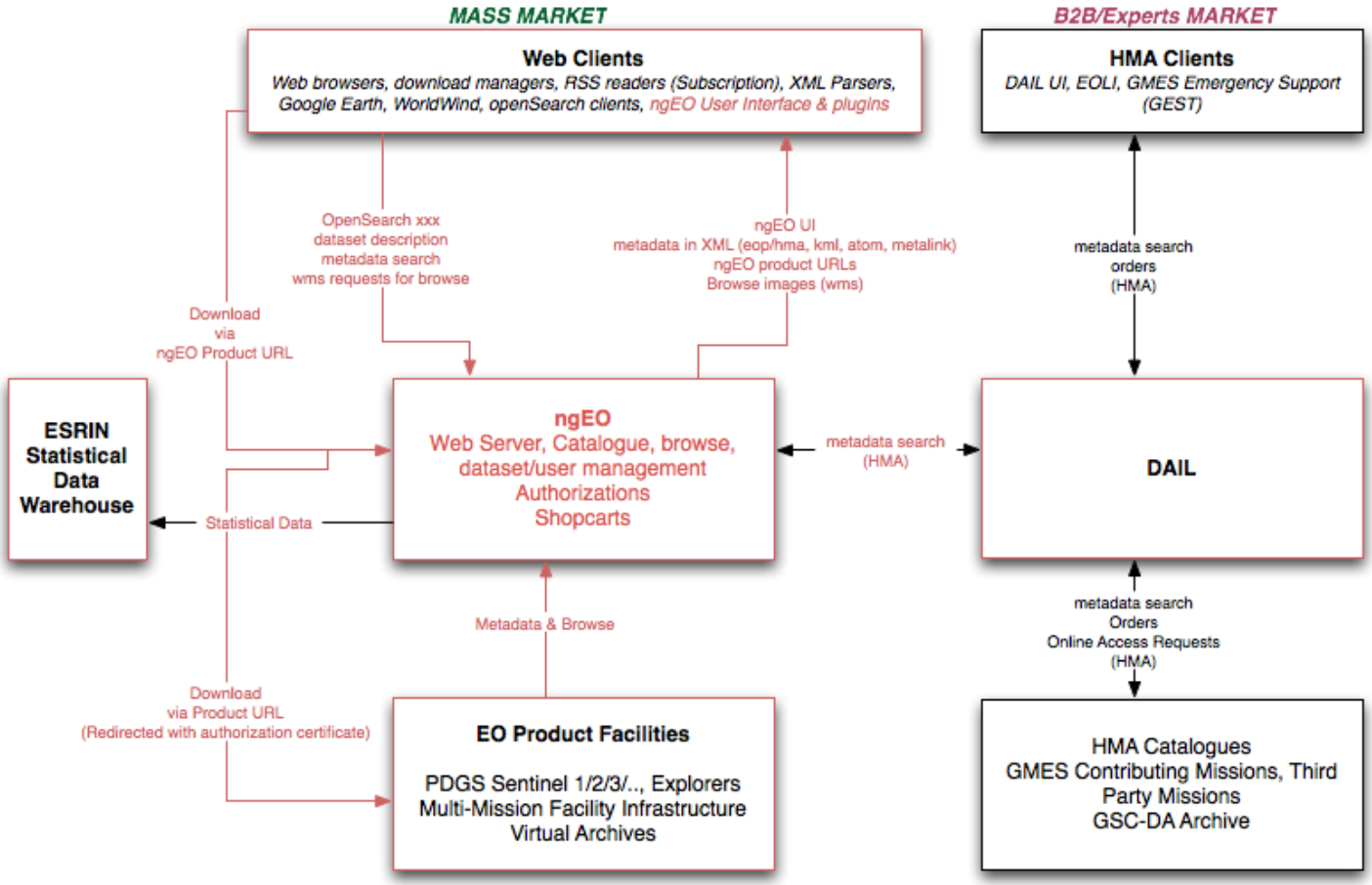
- ESA Grid - G-POD Catalogue (ESA biggest online data repository)

- OGC Web Services Phase 7 (OWS-7)

- Software
  - Geocommons
  - Gi-CAT (ESSI Lab)
  - GeoNetwork (development trunk)
  - OpenLayers (development trunk)

Adoption

- ngEO - ESA next generation of user services for Sentinel ground segment (GMES) seems to adopt it …



CSW 3.0 OpenSearch Vote Nov 2009

- OpenSearch with Geo and Time extensions as the "baseline" query operation for all profiles:
  - the complete CSW 2.0.2 implementation without any additional profiles or extension packages.
  - Supported regardless of profile (or even without a profile).
  - It defines core queryables, query and response message syntax,
  - Allows a client to query any catalog by any profile without having to know the extended profile-specific details
- Requests using HTTP GET and responding with Atom (w/geoRSS) results at a minimum,
- Based on 09-084 “OpenSearch Geospatial Extensions” discussion paper and 08-169 Change Request

CSW 3.0 Part 4 (OGC 10-032) Status

- Final version submitted in pending documents (aligned with the OpenSearch.org)
- Codifies geospatial and temporal extensions in a form compatible with OGC
- Includes:
  - Conformance Classes
  - Mappings for queirables
  - Mappings for response
- OpenSearch is included in annex for reference

FAIR Principles

meet.google.com/fou-wqng-fid