OpenSearch implementation status

Jérôme Gasperi

WGISS #37
Cocoa Beach, Florida - USA - April 15th, 2014
OpenSearch enabled data sources

- **Kalideos**
- **SPOT 5**
- **PEPS**
- **Sentinels 1, 2, 3**
- **SPOT World Heritage**
- **SPOT 4, 5**
- **POSTEL**
- **Biophysical products**
- **Spirit**
- **SPOT 5**
- **Landsat**
- **5, 7, 8**
- **Hydroweb**
- **Lake and rivers heights**
- **Take5**
- **SPOT 4**
- **Landsat**
- **5, 6, 7, 8**

- **Disasters Charter catalog**

*Available as of 04.2014*

*Planned for end of 2014*
OpenSearch capabilities are provided by RESTo
RESTo - restful semantic search tool for geospatial is an implementation of OGC13-026 OpenSearch Extension for Earth Observation
Architecture

(*) Collections can be stored within RESTo database or in external databases

RESTo

RESTo backend

resto db

uses

PostgreSQL
PostGIS
hstore

External services

EO collections *

RESTo modules

Core
Gazetteer
Collection Manager
Query Storage
Query Analyzer
Resource Manager
More to come...

RESTo frontend

mapshup
javascript

iTag
PHP

PostGIS

hstore

PostgreSQL

JavaScript

External services

EO collections *

API modules

More to come...

External services
R E S T O

restful responsive reliable
restful
<table>
<thead>
<tr>
<th>Method</th>
<th>Route</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>/</td>
<td>List all collections</td>
</tr>
<tr>
<td>POST</td>
<td>/</td>
<td>Create a new collection</td>
</tr>
<tr>
<td>GET</td>
<td>/collection/$describe</td>
<td>Describe collection OpenSearch service</td>
</tr>
<tr>
<td>GET</td>
<td>/collection</td>
<td>Search collection</td>
</tr>
<tr>
<td>POST</td>
<td>/collection</td>
<td>Insert a resource within collection</td>
</tr>
<tr>
<td>DELETE</td>
<td>/collection</td>
<td>Delete collection</td>
</tr>
<tr>
<td>PUT</td>
<td>/collection</td>
<td>Update collection</td>
</tr>
<tr>
<td>GET</td>
<td>/collection/identifier</td>
<td>Show resource metadata</td>
</tr>
<tr>
<td>GET</td>
<td>/collection/identifier/$download</td>
<td>Download resource product</td>
</tr>
</tbody>
</table>
responsive
Desktops
Tablets
Mobiles

0 to 500 px
501 to 959 px
960 to > 1280 px
SPOT DATABASE
New products retrieved every 3 hours from ADS catalog

SEARCH
Time period of 1 month within a 10x10 km² box

INGEST
Per product for a ~5000 products ingestion

Order of magnitude compute on a Dual Core 2.6 GHz | 4 Go RAM | HDD 500 To
Search
GET

RESTo

Ingest
POST
During ingestion process, resources are automatically tagged with location and land use

github.com/jjrom/itag
Tag this footprint with continent, country and Land use

http://goo.gl/WtbcbR
Additionally, **conditional ingestion rules** can be defined at the collection level to provide specific **tags**

*Add tags* #mh370,#plane,#malaysianairline
to resources acquired between 2014, march 8th and 2014, april 14th in the south of the Indian Ocean

http://goo.gl/W8VIPV
Search

GET

RESTo
RESTo provides **semantic search capabilities**. It uses a Query Analyzer to translate natural language query into a set of EO OpenSearch parameters.
Query Analyzer goodies

Multilingual - current languages are EN, FR, IT and DE
Synonyms supported (e.g. unit «m» is «m», «meter» or «meters»)
Each collection can define its own dedicated keywords
Automatic typing error correction using similarity
Embed a Gazetteer containing ~9 000 000 toponyms
Example

« Images of urban area in the US acquired in the last 10 days with less than 5% of cloud cover »
Example

<table>
<thead>
<tr>
<th>keyword</th>
<th>location</th>
<th>date</th>
<th>acquisition parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Images of urban area in the US acquired in the last 10 days with less than 5% of cloud cover</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
1. Search parameters are derived from Natural Language query

2. Each search result has an «human readable url» that can be indexed by web crawler (i.e. google robots)

3. Keywords on resources are links to search requests: they can be indexed by web crawler—and so on

http://goo.gl/GvMEHj
http://mapshup.info/resto