Virtual Observatory
Paris Data Centre

# Registry interface proposition VOParis

Jonathan Normand, Pierre Le Sidaner
Observatoire de Paris

# Registry actual interface

- All resources are define by xml schema

- Search method on define fields
- Keywordsearch search on (identifier, content/
- description,title, @xsi:type,content/subjec)

Both methods used complex  ADQL1 language over SOAP

Difficulty to query, not all registry respond, very slow and too much verbous

# The Evolution proposed

**Only define the service behavior not the implementation**

- **Rest access using SEARCH method**

**http://<my_url>/search?q=text[&text]**
**Plain text search in the list of fields**

**http://<my_url>/search?q=field:text1[&field2:text2]**
**search on specific field**

- **Return**

**should be the necessary fields with a link to the complete xml ressource**

**format ? XML mandatory, json recommended**

# Implementation for validation

- All classical services CS, SSA, SIA have been ingest in a no-sql database couchdb. With the field of research (capabilities, description, identifier, subject, type).

+ easy to modify because structure is not fixed
+ easy to maintain
+ easy to ingest new resources (index on the fly)

For search method a search engine have been used Elasticsearch (build on top of Apache Lucene).
Really powerful quick and adapted to text search.
Can face increasing of res sources.
Scalable

# Demo on my laptop

- **Look at couchdb content**

**http://127.0.0.1:5984/_utils**

**Some type of query using elastic search**

**Bases**

**http://127.0.0.1:9200/registry/registry/_search?q=stsci***

**Using a output formalisation**

**http://127.0.0.1:9200/registry/registry/_search?q=stsci*%20%20|%20python2.7%20-mjson.tool**