BRAINOMICS
A management system for exploring and merging heterogeneous brain mapping data based on CubicWeb

Vincent Michel

Logilab

CrEDIBLE 2013 - 3/10/2013

Plan

Logilab

- **computer science and knowledge management** ;
- Founded in 2000 ;
- 20 experts in IT technologies ;
- Public and private clients (CEA, EDF R&D, EADS, Arcelor Mittal, etc.).

Knowledge management :

- `http://collections.musees-haute-normandie.fr/collections/`
- `http://data.bnf.fr/`

Brainomics

Goals

- Software solution for integration of neuroimaging and genomic data ;
- Conception/optimization (GPU) of algorithms for analysing these data ;

Collaborative R&D project

- **NeuroSpin laboratory of CEA**
- Supelec ;
- UMR 894 of INSERM ;
- UMR CNRS 8203 of IGR ;
- **Logilab ;**
- Alliance Services Plus (AS+) ;
- Keosys.

This work was supported by grants from the French National Reseach Agency (ANR GENIM ; ANR-10-BLAN-0128) and (ANR IA BRAINOMICS ; ANR-10-BINF-04).

Context

Brain mapping data

- Large datasets for brain mapping :
  - `http://openfmri.org/data-sets`
  - `http://fcon_1000.projects.nitrc.org/indi/abide/`
- Neuroimaging + clinical data + genetics data ;

Brain mapping databases

- Neuroimaging and genomics databases are dedicated to their own field of research ;
  - *XNAT* - Neuroimaging ;
  - *BASE* - Genetics ;
  - *SHANOIR* - Neuroimaging ;

Plan

What is Cubicweb

A semantic open-source web framework written in Python

An efficient knowledge management system

- **Entity-relationship data-model** ;
- **RQL (Relational Query Language)** ;
- **Separate query and display** (HTML UI, JSON, RDF, CSV,...) ;
- **Conform to the Semantic Web standards** ;
- **Fine-grained security system** coupled to the data model definition ;
- **Migration mechanisms** control model version and ensure data integrity ;
- **Industrial use** : large databases, many users, security, logging ;

Used in production environments since 2005 ; LGPL since 2008 ;

`http://www.cubicweb.org/`

Logilab

What Cubicweb is not

Cubicweb is not

- **a pure Web application framework** - Web/HTML interface is only one possible output ;
- **a triple store** - data is structured ;
- **a CMS** - allows complex business data modeling ;

Cubicweb is a framework

Used to build applications, with reusable components called *cubes* :

- *data model* : persons, addressbook, billing...
- *displays (a.k.a views)* : d3js, workflow, maps, threejs...
- *full applications* : forge, intranet, erp...
- *open databases* : dbpedia, diseasome, pubmed...

Overview of the framework

Well established core technologies : SQL, Python, HTML5, Javascript ;

Application

- Based on a relational database (e.g. PostgreSQL) for storing information ;
- Web server for HTTP access to the data ;
- Integration with existing LDAP for high-level access management ;

Code - written in Python

- *Schema* - define the data model.
- *Business Logic* - allow to increase the logic of the data beyond the scope of the data model, using specific functions and adapters.
- *Views* - specific display rules, from HTML to binary content.

Logilab

CubicWeb and Semantics

Not using a triple store does not mean "not semantic web compliant"

CubicWeb conforms to the Semantic Web standards

- One entity = an unique URI ;
- One request = an unique URI :
  `http://localhost:8080/?rql=MYRQLREQUEST&vid=MYVIEWID`
- HTTP content negociation (HTML, RDF, JSON, etc.) ;
- Import/export to/from RDF data, based on a specific mapping :

```
xy.add_equivalence('Person given_name',
                   'foaf:givenName')
```

Use RDF as a standard I/O format for data, but stick to relational database for storage.

Visualization and interactions

Cubicweb *views*

- A view is applied on the result of a query ;
- The same result may be visualized using different views ;
- Views are selected based on the types of the resulting data ;

Exploring the data

- Many different possible views ;
- Auto-completion RQL query form ;
- Filtering facets ;
- Using data in scripts with the URL vid/rql parameters ;

And also : *forms*, *widgets*, ...

Brainomics

Brings together brain imaging and genetics data

A solution based on CubicWeb

- Modeling of Scans, Questionnaires, Genomics results, Behavioural results, Subjects and Studies information, ...
- Can deal with large volumes (> 10000 subjects) ;
- Tested with several datasets (openfmri, abide, imagen, localizer) ;
- Specific views : ZIP, XCEDE XML, CSV ;

Open source solution

`http://www.brainomics.net/demo`

Plan

Data model and Schema

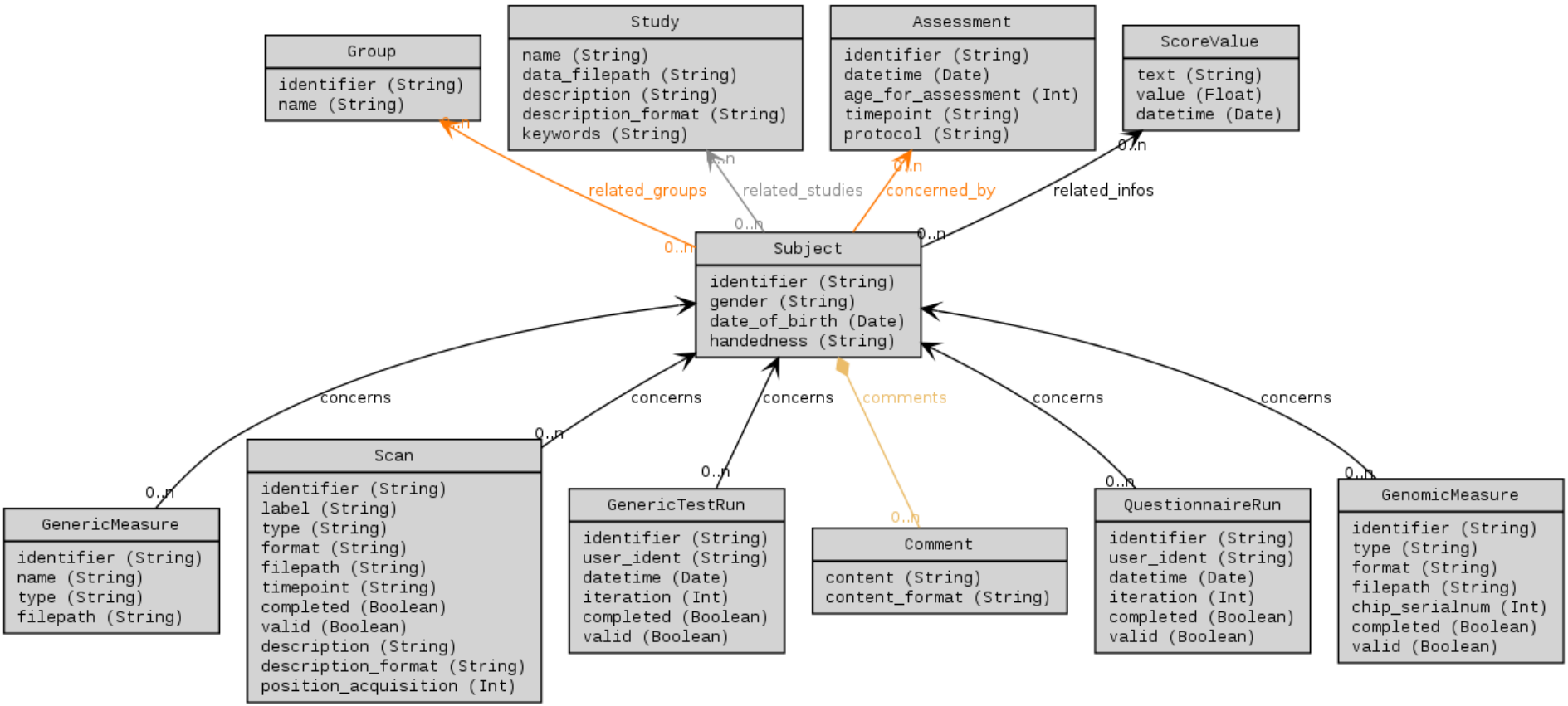
CubicWeb schema

- Based on a Python library : `http://www.logilab.org/project/yams`
- Defined in a Python file (*schema.py*) ;
- Allow to create entity types, relations, constraints ;
- Security can be tightly included in the data model ;

```
class Subject(EntityType):
    identifier = String(required=True,indexed=True,maxsize=64)
    gender = String(vocabulary=('male', 'female', 'unknown'))
    date_of_birth = Date()
    ...

class related_studies(RelationType):
    subject = 'Subject'
    object = 'Study'
    cardinality = '1*'
```

Data model - Subject



Where are the reference models ?

Reference models

Reference models may be implemented as Cubicweb schema

- additional modeling refinements for application features (sortability, readability, ...)
- already existing cubes for some ontologies/taxonomies (e.g. FRBR) ;

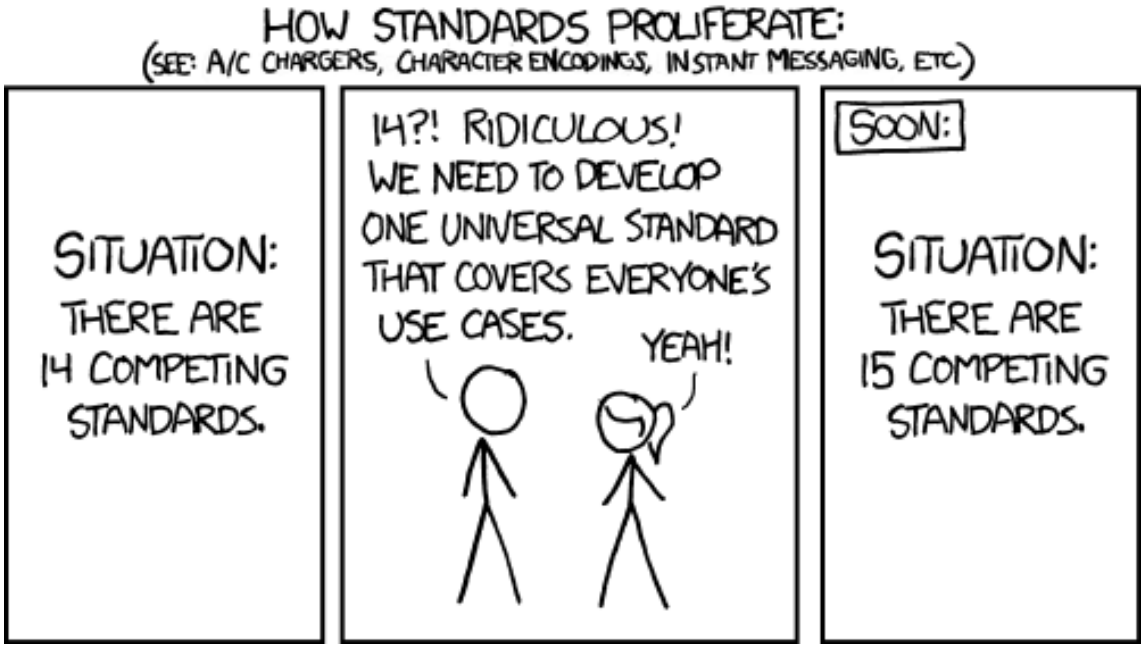
Be pragmatic : if you need a new attribute, add it in the model $\rightarrow$ deal with the reference model in the I/O formats ;

... or you could use your own specific schema

- Ontologies do not exist for all fields ;
- Define your schema and only code the mappings (I/O) to reference models when they go out.

Logilab

Schema evolution



http://xkcd.com/

Standards are moving

- Existing **Migration mechanisms** to stick to the reference models evolution ;
- Easy modification for modeling of other applicative fields ;

Data - Input

Keep the application data model as pivot model → convert the data to this model during the insertion

Datafeed - Periodic input

- an *URL* ;
- some Python logic for integrating/updating information ;
- an interval of synchronization ;

→ used for almost any possible type of data, e.g. RSS feeds, files, RDF data, other CW instances...

Stores - Bulk loading

Tools similar to ETL

- allow to import huge amount of structured data ;
- principally used for bulk loading ;
- different level of security and data check ;

Data - Output

Design a specific view for outputing in any reference model

- If one reference model change, change the I/O, not the internal model ;
- Avoid data redundancy, if different reference models are based on the same information (e.g. *dc :title*, *foaf :name*) ;

Example of View - Export a Scan in XCEDE format

```
class ScanXcedeItemView(XMLItemView):
    __select__ = XMLItemView.__select__ & is_instance('Scan')
    __regid__ = 'xcede-item'

    def entity_call(self, entity):
        self.w(u'<acquisition ID="%(id)s" projectID="%(p)s"
                'subjectID="%(s)s" visitID="%(a)s" '
                'studyID="%(a)s" episodeID="%(la)s"/>\n'
                 % {'id': entity.identifier,
                    'p': entity.related_study[0].name,
                 ...
```

Data Output - Example of XCEDE

```
-<XCEDE>
    <subject ID="demo_subject_0"/>
    <visit ID="cmap_1827" projectID="Demo study" subjectID="demo_subject_0"/>
    <visit ID="fmri_1827" projectID="Demo study" subjectID="demo_subject_0"/>
    <visit ID="anat_1827" projectID="Demo study" subjectID="demo_subject_0"/>
    <visit ID="genomics_1827" projectID="Demo study" subjectID="demo_subject_0"/>
    <visit ID="vineland_1827" projectID="Demo study" subjectID="demo_subject_0"/>
    <visit ID="ados_1827" projectID="Demo study" subjectID="demo_subject_0"/>
    <acquisition ID="tmap_4_1827" projectID="Demo study" subjectID="demo_subject_0" visitID="cmap_1827"
    studyID="cmap_1827" episodeID="Audio interaction"/>
  -<dataResource xsi:type="dimensionedBinaryDataResource_t">
      <uri offset="0" size="262144">/tmp/demo_data/tmap/1827_4.nii.gz</uri>
      <elementType>float32</elementType>
      <compression>nii.gz</compression>
    -<dimension label="x">
        <size>128</size>
        <spacing>1.5</spacing>
        <direction>1 0 0</direction>
        <units>mm</units>
      </dimension>
```

Data Output - RDF specific case

Existing tools in CubicWeb for RDF mapping

RDF mapping

```
xy.register_prefix('foaf', 'http://xmlns.com/foaf/0.1/')

xy.add_equivalence('Subject', 'foaf:Subject')
xy.add_equivalence('MedicalCenter', 'foaf:Organization')

xy.add_equivalence('Subject given_name', 'foaf:givenName')
xy.add_equivalence('Subject family_name', 'foaf:familyName')

xy.add_equivalence('* same_as *', 'owl:sameAs')
xy.add_equivalence('* see_also *', 'foaf:page')
```

Also possible to plug specific Python functions for non-trivial mapping.

Used for RDF import/export and SPARQL endpoint

Logilab

Plan

RQL - Relational Query Language

Features

- **Similar to W3C's SPARQL**, but less verbose ;
- Supports the basic operations (select, insert, etc.), subquerying, ordering, counting, ...
- Tightly integrated with SQL, but **abstracts the details of the tables and the joins** ;
- Use the schema for data types inference, based on a syntactic analysis of the request.

$\rightarrow$ A query returns a **result set** (a list of results), that can be displayed using specific views.

RQL - Example

Query all the Cmap scans of left-handed male subjects that have a score greater than 4.0 for the "algebre" question of the Localizer questionnary

↓

```
Any SA WHERE S handedness "left", S gender "male",
            X concerns S, A questionnaire_run X,
            A question Q, Q text "algebre", A value > 4,
            SA concerns S, SA is Scan, SA type "c map"
```

and the SQL translation ...

```
SELECT _SA.cw_eid FROM cw_Answer AS _A, cw_Question AS _Q,
cw_QuestionnaireRun AS _X, cw_Scan AS _SA, cw_Subject AS _S
WHERE _S.cw_handedness="left" AND _S.cw_gender"male"
AND _X.cw_concerns=_S.cw_eid
AND _A.cw_questionnaire_run=_X.cw_eid
AND _A.cw_question=_Q.cw_eid AND _Q.cw_text="algebre"
AND _A.cw_value>4 AND _SA.cw_concerns=_S.cw_eid
AND _SA.cw_type="c map"
```

RQL for Endusers

No perfect dashboard / no perfect search form

↓

Put an expressive query language in the hands of the endusers

Exploring the data model

- Explore the schema

  `http://localhost:8080/schema`

- RQL completion form ;
- Learning RQL by showing the RQL for each page/each filter ;

Deep exploration of the data by the endusers

Logilab

RQL VS SPARQL

Developped in parallel with SPARQL years ago, with a focus on SQL database and support of SET / UPDATE / DELETE.

Why we should support SPARQL ...

- SPARQL is a W3C standard, and a reference in the Semantic Web community.
- To improve interoperability of the CubicWeb application with other Semantic Web technologies.

... and why we don't want to use only SPARQL

- Huge company's internal knowledge on RDBMS (VS Triplestores) ;
- SPARQL is quite verbose, RQL is more intuitive and elegant (*"Syntax matters"*) ;

There exists a basic translation from SPARQL to RQL (only selection queries).

Logilab

Federated databases in CW

PostgreSQL federated databases

Let PostgreSQL do what it does best

- Since PostgreSQL 9.3 ;
- Based on *Foreign Data Wrapper (FDW)* ;
- Could be used to federated queries in CW (WIP) ;

FROM clause (WIP)

Similar to SPARQL *SERVICE*

```
Any P, S WHERE GEN is GenomicMeasure, GEN concerns S,
GEN platform P, P related_snps SN, SN in_gene G, G name GN
WITH P BEING (Any X WHERE X is Paper, X keywords GN)
FROM http://pubmed:8080
```

Eventually, also allow SPARQL subqueries ;

Or use CubicWeb SPARQL endpoint with *SERVICE*.

Plan

Conclusion

Brainomics

- Open source solution to manage brain imaging datasets and associated metadata ;
- Powerful querying and reporting tool, customized for emerging multimodal studies.

Feedback from Brainomics

- Do not store raw data in database ;
- Try to interact with existing reference databases ;
- **Using CubicWeb** :
  - Easy modeling of other applicative fields in the schema (e.g. Histology) ;
  - Security, migrations are already included ;
  - Many different views, and existing API to define your own ;

Future work

Future of Brainomics

- How to transfer large files (>10Go for genotype files) ?
- Need Content Delivery Network (CDN) in CubicWeb ;
- Integration to reference databases (pubmed, refseq, ...) ;

Future of Cubicweb

- Extended support of SPARQL ;
- Finish the work on federated queries ;
- REST support, Python's Web Server Gateway Interface ;
- Better integration with Bootstrap ;

Questions ?

http://www.cubicweb.org/project/cubicweb-brainomics
http://www.brainomics.net/demo/

vincent.michel@logilab.fr

brainomics@logilab.fr

cubicweb@lists.cubicweb.org

RQL/SPARQL - Example

Cities of Île de France with more than 100 000 unhabitants ?

RQL

```
Any X WHERE X region Y, X population > 100000,
Y uri "http://fr.dbpedia.org/resource/Île-de-France"
```

SPARQL

```
select ?ville where {
  ?ville db-owl:region <http://fr.dbpedia.org/resource/Île-de-F
  ?ville rdf:type db-owl:Settlement .
  ?ville db-owl:populationTotal ?population .
  FILTER (?population > 100000)
}
```

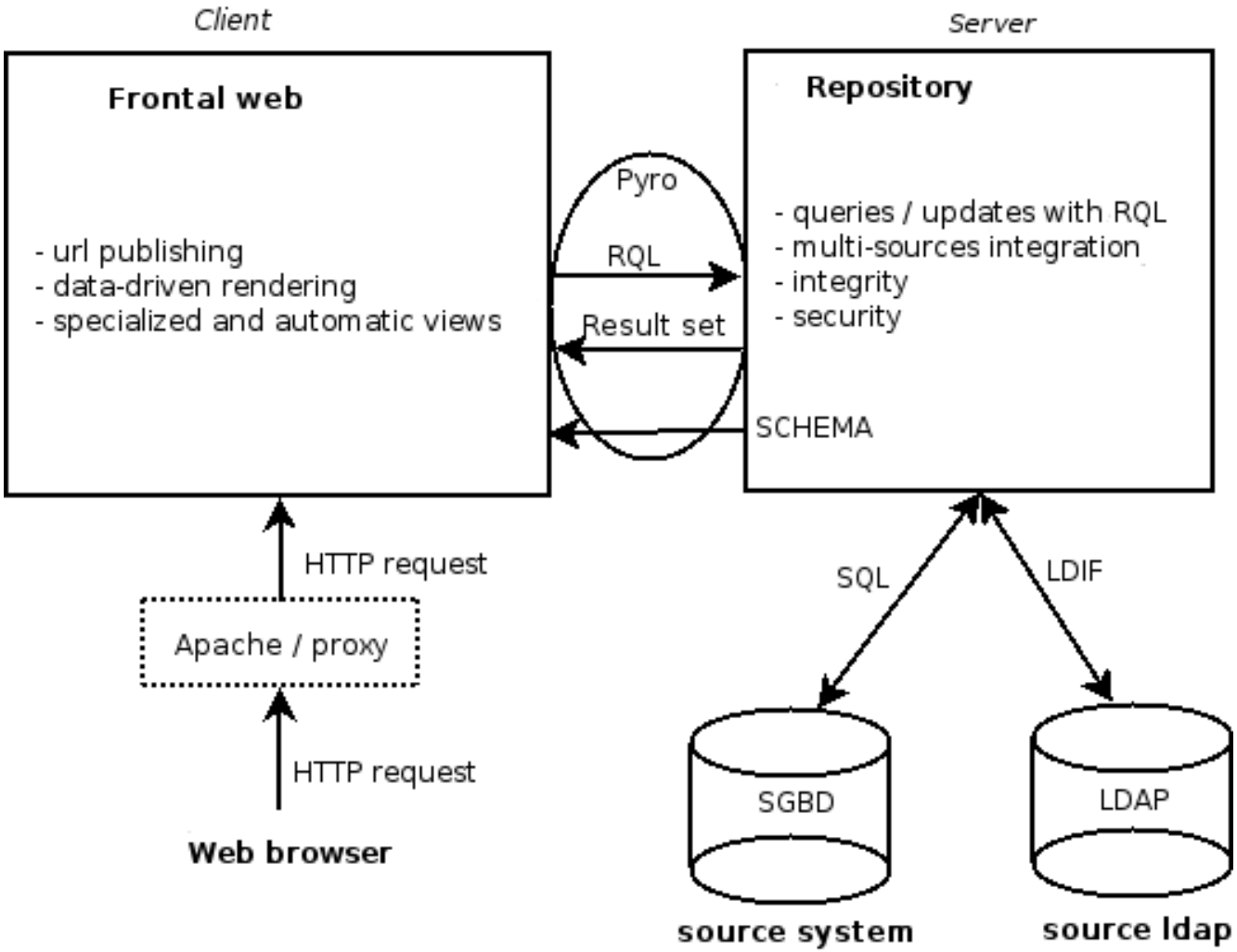
This is NOT big data !

Small databases ...

- **No need to store raw data in the database** ;
- Structured data : metadata (patient informations, ...), choosen scores (some statistical values on genes of interest, ...) ;
- 10 Mo / subject * 10.000 subjects = 100 Go ;
- Classical SQL databases (e.g. PostgreSQL) work greats up to few TB !

http://www.chrisstucchio.com/blog/2013/hadoop_hatred.html
http://www.vitavonni.de/blog/201309/2013092701-big-data-madness-and-reality.html

Global architecture of CubicWeb



What's needed

- Efficient **data model to integrate all the measures** ;
- **Easy access to the relevant information** (query language + UI) ;
- **Import** / **Export** in several formats, for **merging heterogenous studies** ;
- **Adaptable to the evolutions** of various dynamic applicative fields.

CubicWeb, a semantic datamanagement framework

Logilab

Data model - Assessment