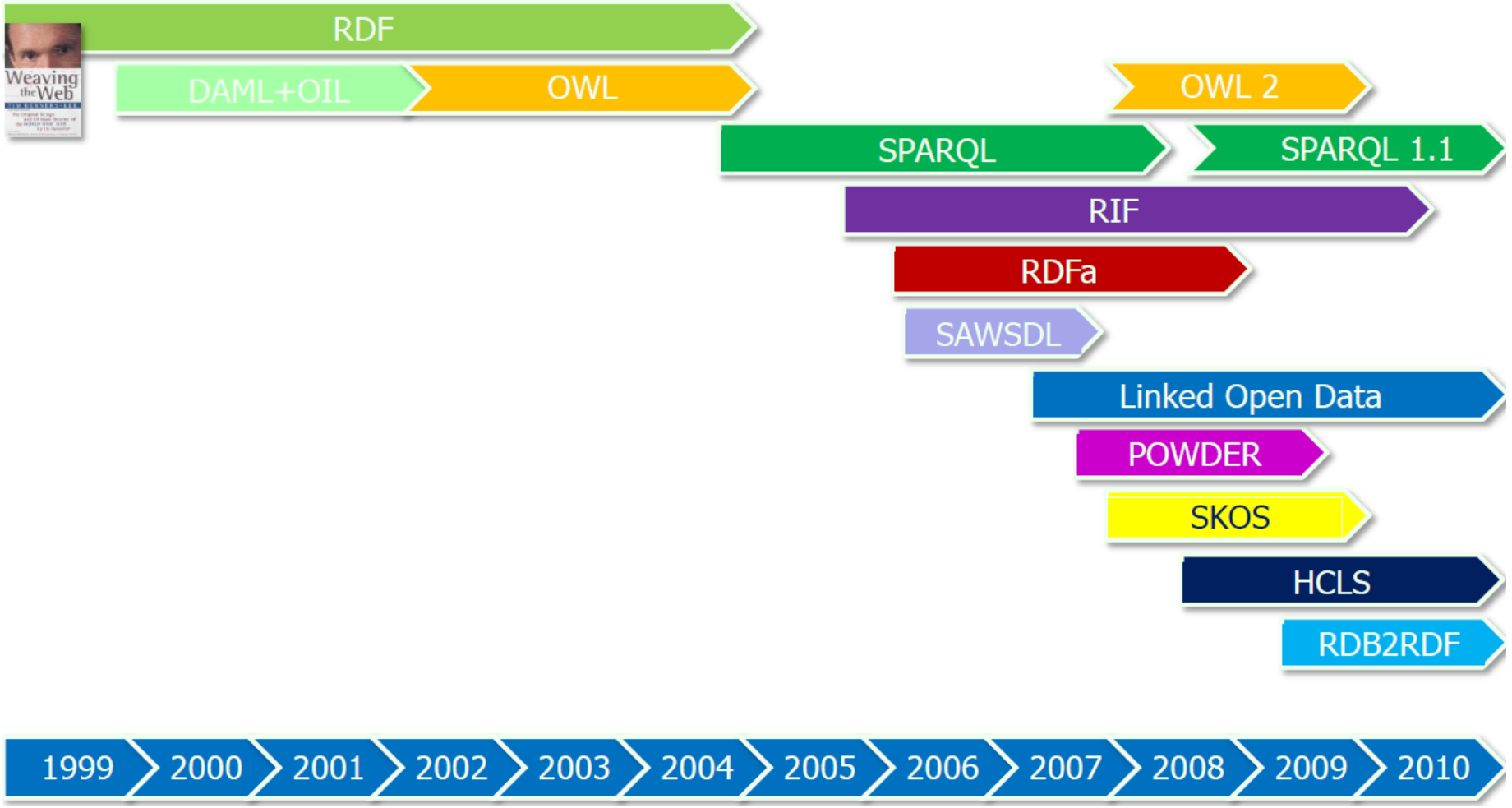
SPARQL - Query Language for RDF

Fulvio Corno, Laura Farinetti
Politecnico di Torino
Dipartimento di Automatica e Informatica
e-Lite Research Group – http://elite.polito.it

The “new” Semantic Web vision

- To make data machine processable, we need:
  - Unambiguous names for resources (that may also bind data to real world objects): URIs
  - A common data model to access, connect, describe the resources: RDF
  - Access to that data: SPARQL
  - Define common vocabularies: RDFS, OWL, SKOS
  - Reasoning logics: OWL, Rules
- “SPARQL will make a huge difference” (Tim Berners-Lee, May 2006)

The Semantic Web timeline



SPARQL basics

SPARQL

- Queries are very important for distributed RDF data
  - Complex queries into the RDF data are often necessary
  - E.g.: “give me the (a,b) pair of resources, for which there is an x such that (x parent a) and (b brother x) holds” (i.e., return the uncles)
- This is the goal of SPARQL (Query Language for RDF)

SPARQL

- SPARQL 1.0: W3C Recommendation January 15th, 2008
- SPARQL 1.1: W3C Working Draft January 5th, 2012
- SPARQL queries RDF graphs
  - An RDF graph is a set of triples
- SPARQL can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware

SPARQL and RDF

- It is the triples that matter, not the serialization
  - RDF/XML is the W3C recommendation but it not a good choice because it allows multiple ways to encode the same graph
- SPARQL uses the Turtle syntax, an N-Triples extension

Turtle - Terse RDF Triple Language

N-Triples ⊂ Turtle ⊂ N3

- A serialization format for RDF
- A subset of Tim Berners-Lee and Dan Connolly's Notation 3 (N3) language
  - Unlike full N3, doesn't go beyond RDF's graph model
- A superset of the minimal N-Triples format
- Turtle has no official status with any standards organization, but has become popular amongst Semantic Web developers as a human-friendly alternative to RDF/XML

“Triple” or “Turtle” notation

“Triple” or “Turtle” notation

```
<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/2000/10/swap/pim/contact#fullName>
"Eric Miller" .

<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/2000/10/swap/pim/contact#mailbox>
<mailto:em@w3.org> .

<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/2000/10/swap/pim/contact#personalTitle>
"Dr." .

<http://www.w3.org/People/EM/contact#me>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://www.w3.org/2000/10/swap/pim/contact#Person> .
```

“Triple” or “Turtle” notation (abbreviated)

```
w3people:EM#me contact:fullName "Eric Miller" .

w3people:EM#me contact:mailbox <mailto:em@w3.org> .

w3people:EM#me contact:personalTitle "Dr." .

w3people:EM#me rdf:type contact:Person .
```

Turtle - Terse RDF Triple Language

- Plain text syntax for RDF
  - Based on Unicode
- Mechanisms for namespace abbreviation
- Allows grouping of triples according to subject
- Shortcuts for collections

Turtle - Terse RDF Triple Language

- Simple triple:
  subject predicate object .

```
:john rdf:label "John" .
```

- Grouping triples:
  subject predicate object ; predicate object ...

```
:john
  rdf:label "John" ;
  rdf:type ex:Person ;
  ex:homePage http://example.org/johnspage/ .
```

Prefixes

- Mechanism for namespace abbreviation

```
@prefix abbr: <URI>
```

- Example:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

- Default:

```
@prefix : <URI>
```

- Example:

```
@prefix : <http://example.org/myOntology#>
```

Identifiers

- URIs: <URI>

```
http://www.w3.org/1999/02/22-rdf-syntax-ns#
```

- Qnames (Qualified names)
  - `namespace-abbr?:localname`

```
rdf:type
dc:title
:hasName
```

- Literals
  - `"string"(@lang)?(^^type)?`

```
"John"
"Hello"@en-GB
"1.4"^^xsd:decimal
```

Blank nodes

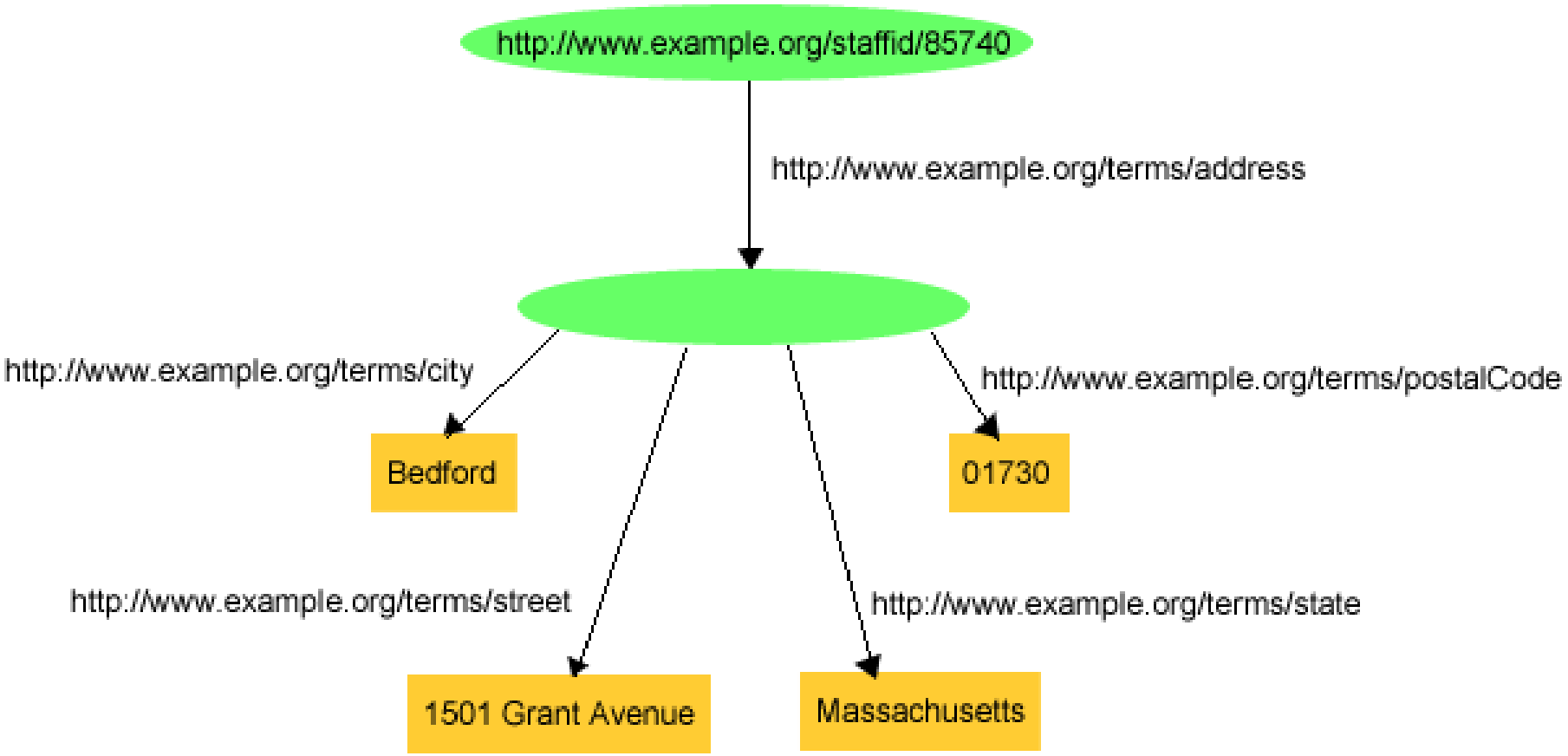
- Simple blank node:
  - `[] or _:x`

```
:john ex:hasFather [] .
:john ex:hasFather _:x .
```

- Blank node as subject:
  - `[ predicate object ; predicate object ... ] .`

```
[ ex:hasName "John"] .
[ ex:authorOf :lotr ;
  ex:hasName "Tolkien"] .
```

Blank nodes


http://www.example.org/staffid/85740
http://www.example.org/terms/address
http://www.example.org/terms/city
http://www.example.org/terms/postalCode
Bedford
01730
http://www.example.org/terms/street
http://www.example.org/terms/state
1501 Grant Avenue
Massachusetts

Collections

- (object1 ... objectn)

```
:doc1 ex:hasAuthor (:john :mary) .
```

- Short for

```
:doc1 ex:hasAuthor
   [ rdf:first :john;
     rdf:rest [ rdf:first :mary;
     rdf:rest rdf:nil ]
   ] .
```

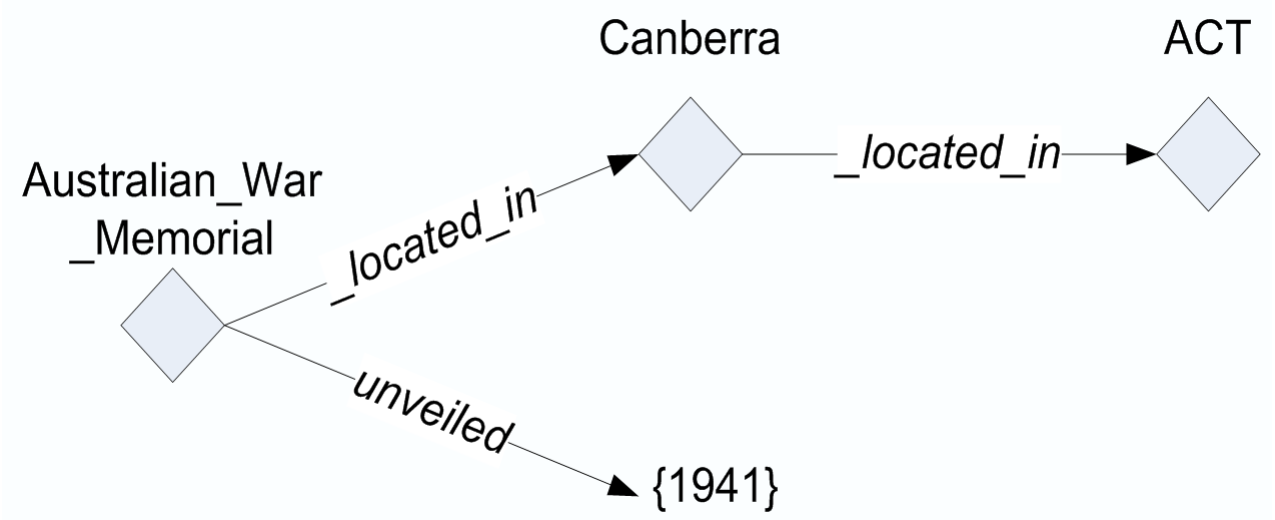
Example

```
@prefix rdf: http://www.w3.org/1999/02/22-rdf-syntaxns# .
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/#> .

<http://www.w3.org/TR/rdf-syntax-grammar>
  dc:title "RDF/XML Syntax Specification (Revised)" ;
  :editor [
    :fullName "Dave Beckett";
    :homePage <http://purl.org/net/dajobe/>
  ] .
```

RDF Triplestores

- Basically, databases for triples



- Triplestores do not only store triples, but allow to extract the “interesting” triples, via SPARQL queries

Comparison

Relational database

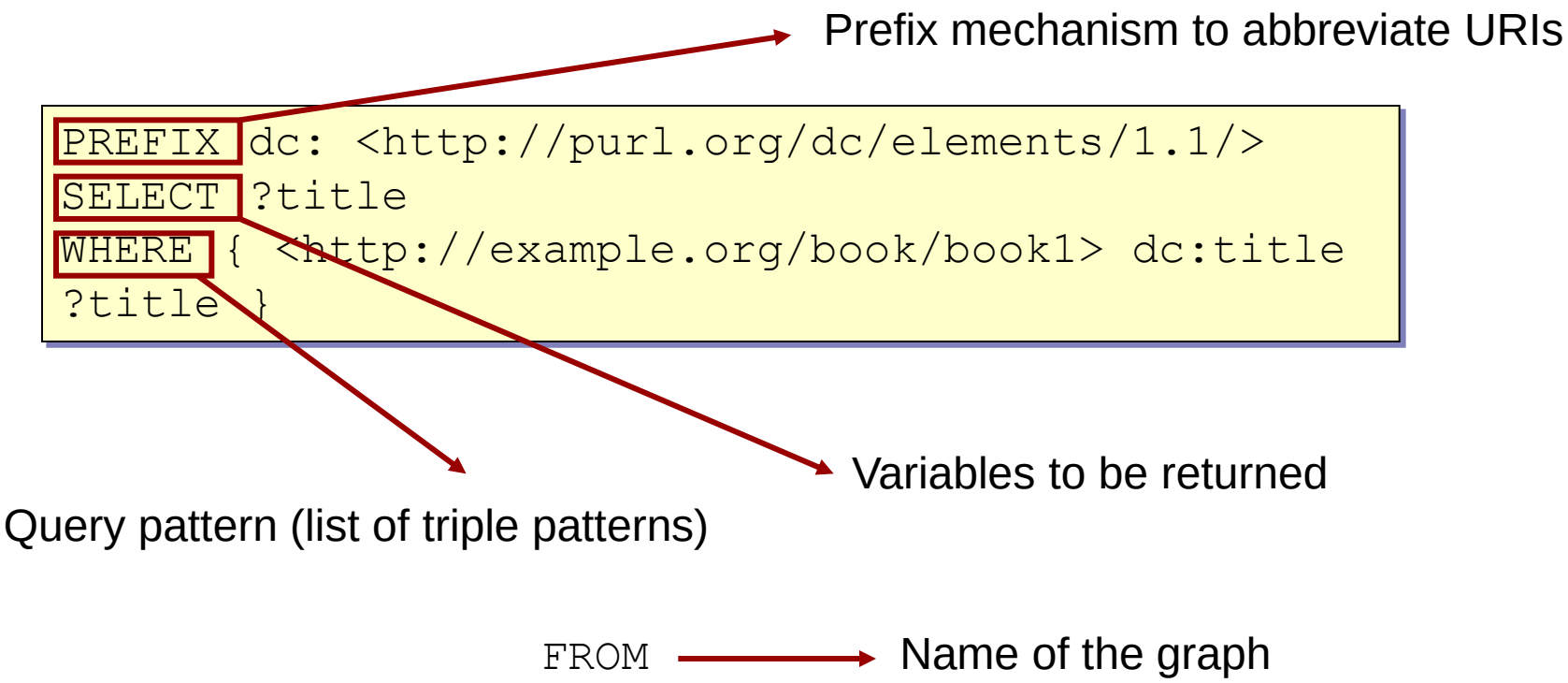
- Data model
  - Relational data (tables)
- Data instances
  - Records in tables
- Query support
  - SQL
- Indexing mechanisms
  - Optimized for evaluating queries as relational expressions

RDF Triplestore

- Data model
  - RDF graphs
- Data instances
  - RDF triples
- Query support
  - SPARQL
- Indexing mechanisms
  - Optimized for evaluating
  - Queries as graph patterns

SPARQL

- Uses SQL-like syntax



SELECT

- Variables selection
- Variables: `?string`

```
?x
?title
?name
```

- Syntax: `SELECT` $var_1,\ldots,var_n$

```
SELECT ?name
SELECT ?x,?title
```

WHERE

- Graph patterns to match
- Set of triples

```
{ (subject predicate object .)* }
```

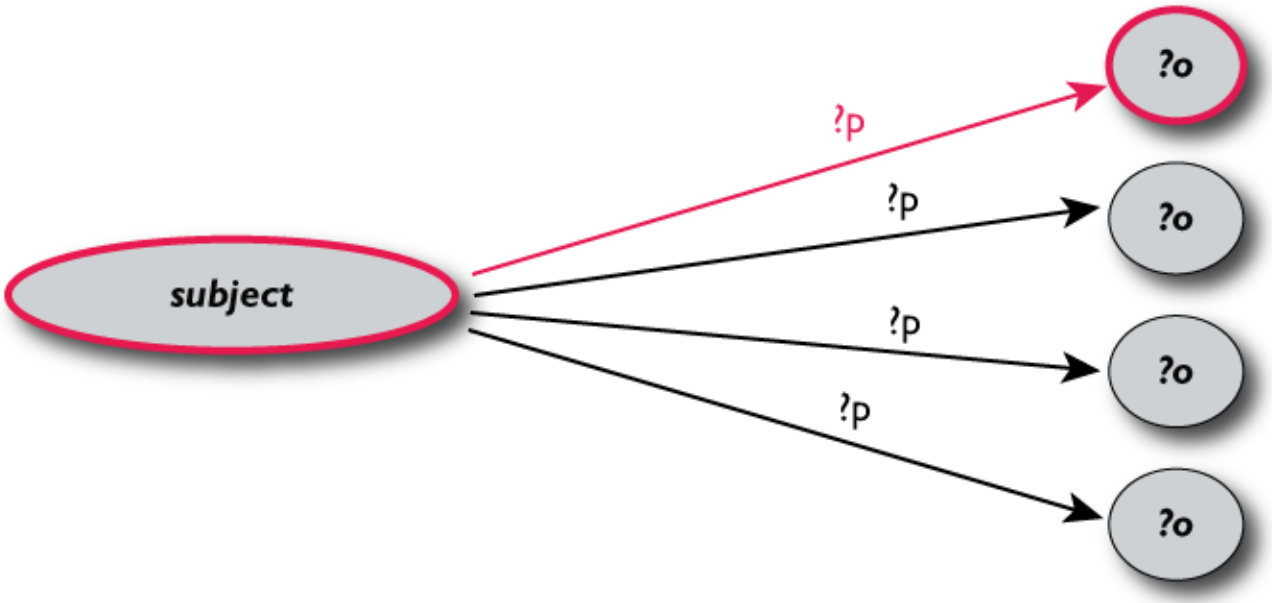
- Subject: URI, QName, Blank node, Literal, Variable
- Predicate: URI, QName, Blank node, Variable
- Object: URI, QName, Blank node, Literal, Variable

Graph patterns

- The pattern contains unbound symbols
- By binding the symbols (if possible), subgraphs of the RDF graph are selected
- If there is such a selection, the query returns the bound resources

Graph patterns

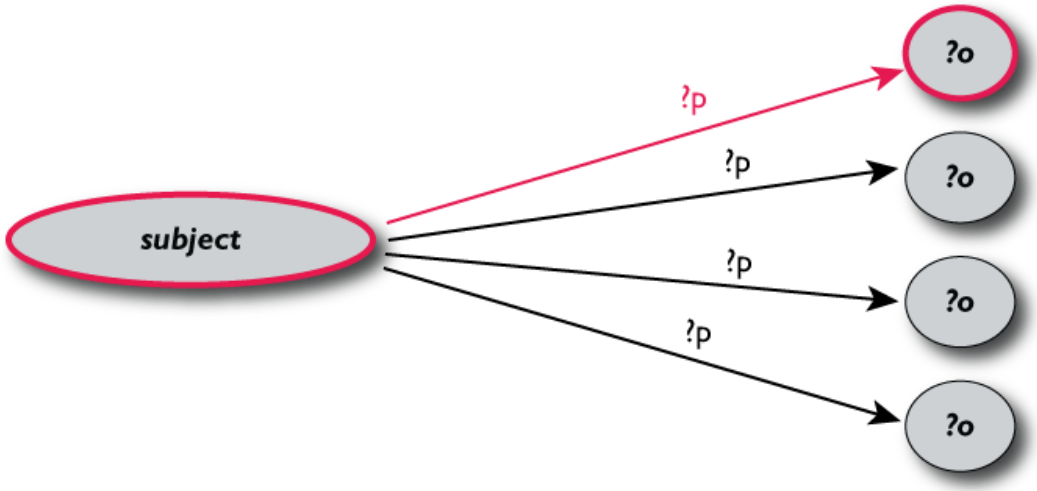
- E.g.: (subject,?p,?o)
  - ?p and ?o are “unknowns”



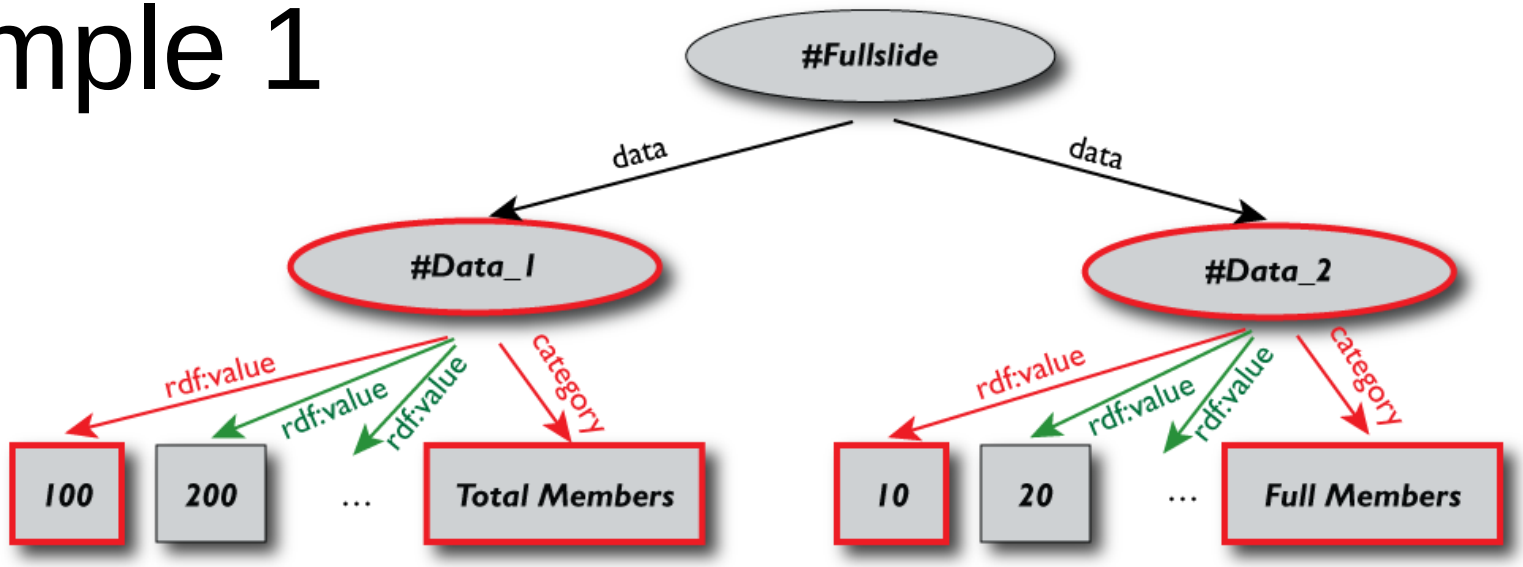
Graph patterns

```
SELECT ?p ?o
WHERE {subject ?p ?o}
```

- The triplets in WHERE define the graph pattern, with ?p and ?o “unbound” symbols
- The query returns a list of matching p,o pairs



Example 1

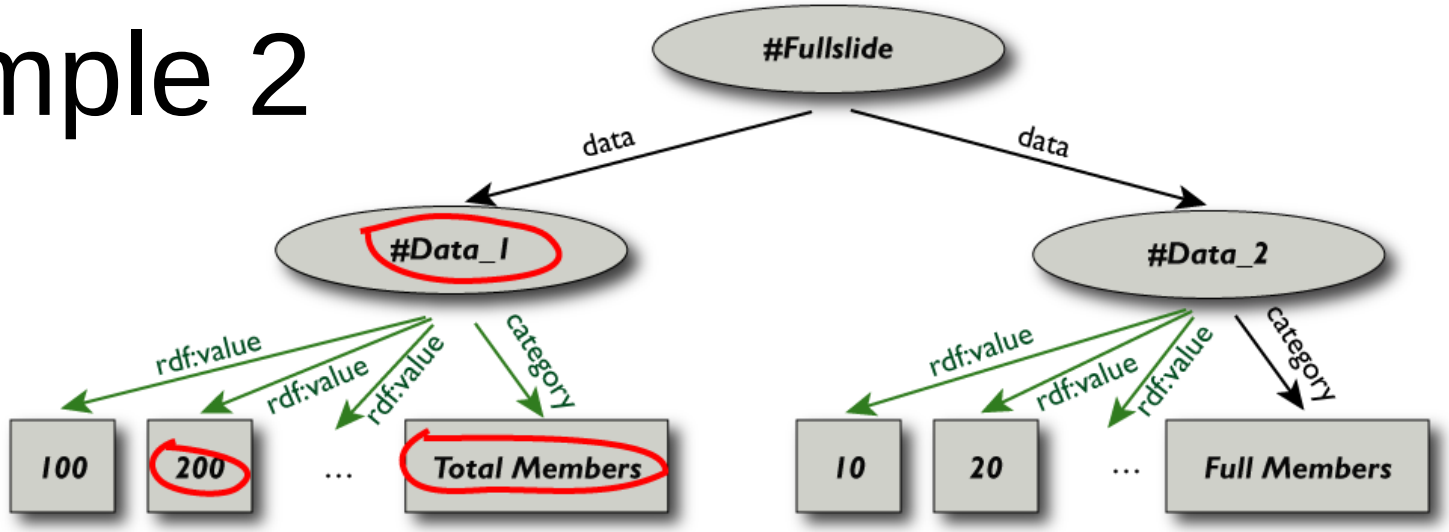



```
SELECT ?cat, ?val
WHERE { ?x rdf:value ?val.
        ?x category ?cat }
```

- Returns:

```
[["Total Members",100],["Total Members",200],…,
["Full Members",10],…]
```

Example 2

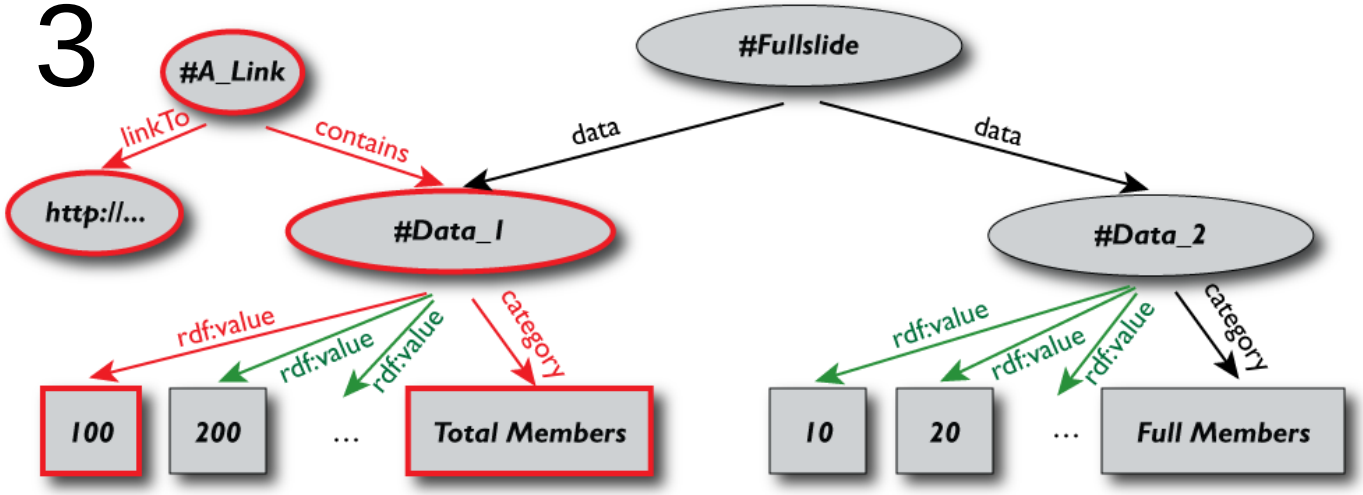



```
SELECT ?cat, ?val
WHERE { ?x rdf:value ?val.
        ?x category ?cat.
       FILTER(?val>=200). }
```

- Returns:

```
[["Total Members",200],…]
```

Example 3

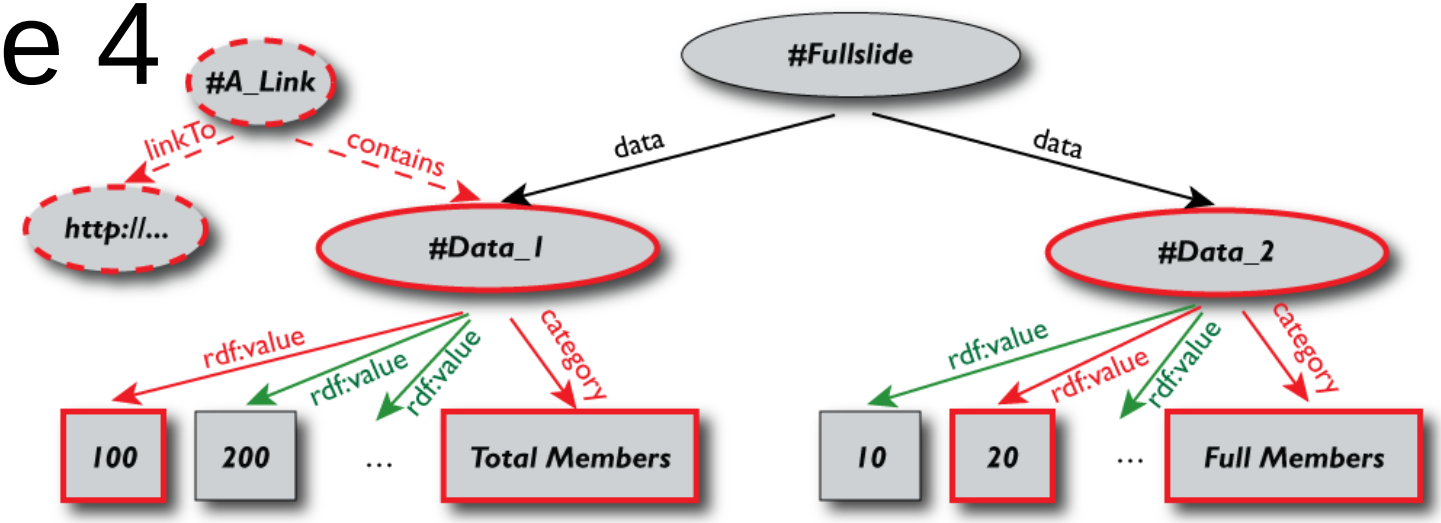



```
SELECT ?cat, ?val, ?uri
WHERE { ?x rdf:value ?val.
        ?x category ?cat.
        ?al contains ?x.
        ?al linkTo ?uri }
```

- Returns:

```
[["Total Members",100,http://...)],…,]
```

Example 4




```
SELECT ?cat, ?val, ?uri
WHERE  { ?x rdf:value ?val.
         ?x category ?cat.
OPTIONAL ?al contains ?x.
         ?al linkTo ?uri }
```

- Returns:

```
[["Total Members",100,http://...], …,
["Full Members",20, ],…,]
```

Other SPARQL Features

- Limit the number of returned results
- Remove duplicates, sort them,…
- Specify several data sources (via URI-s) within the query (essentially, a merge)
- Construct a graph combining a separate pattern and the query results
- Use datatypes and/or language tags when matching a pattern

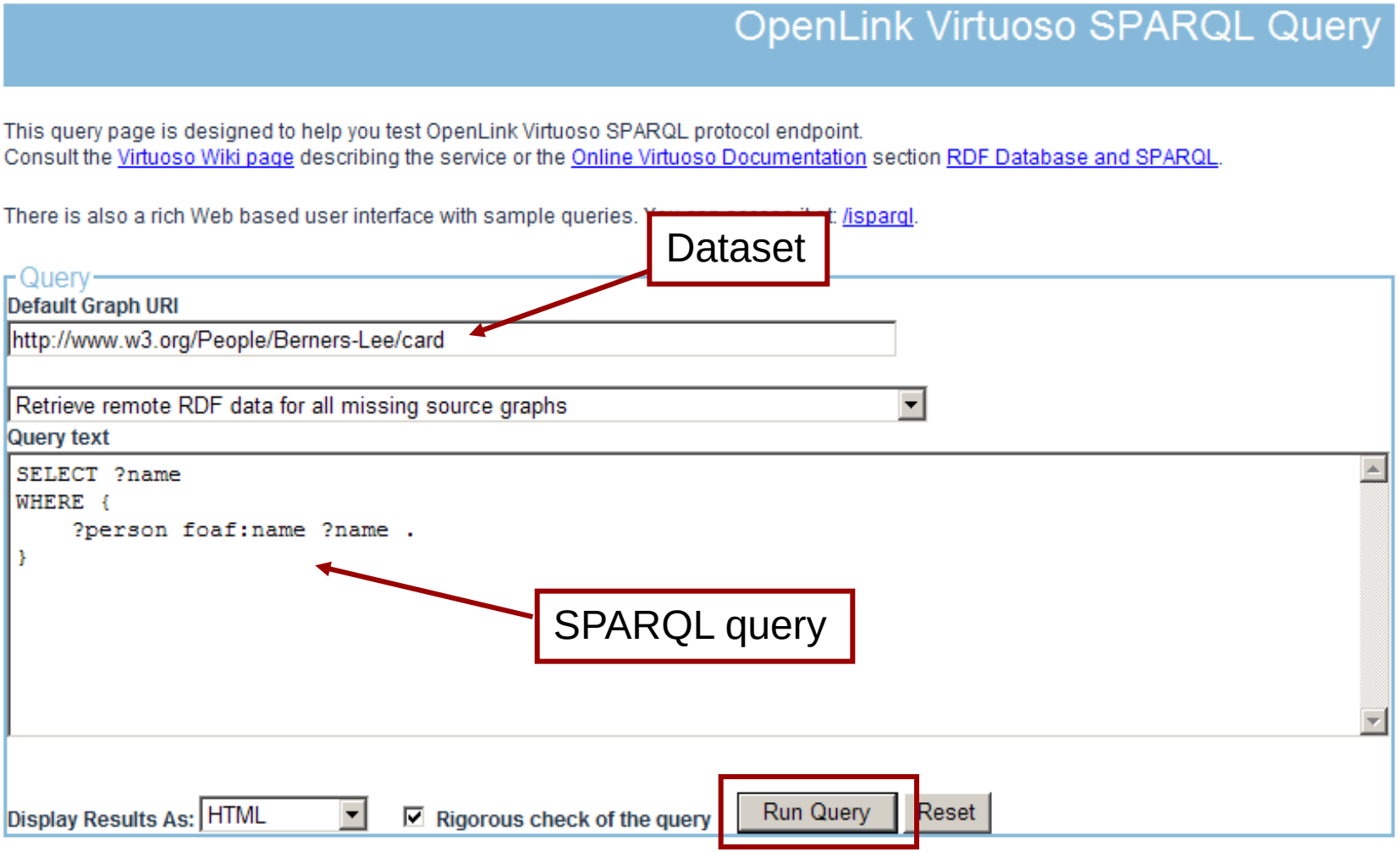
SPARQL use in practice

- Locally, i.e., bound to a programming environments like Jena
  - Jena is a Java framework for building Semantic Web applications; provides an environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine
- Remotely, e.g., over the network or into a database

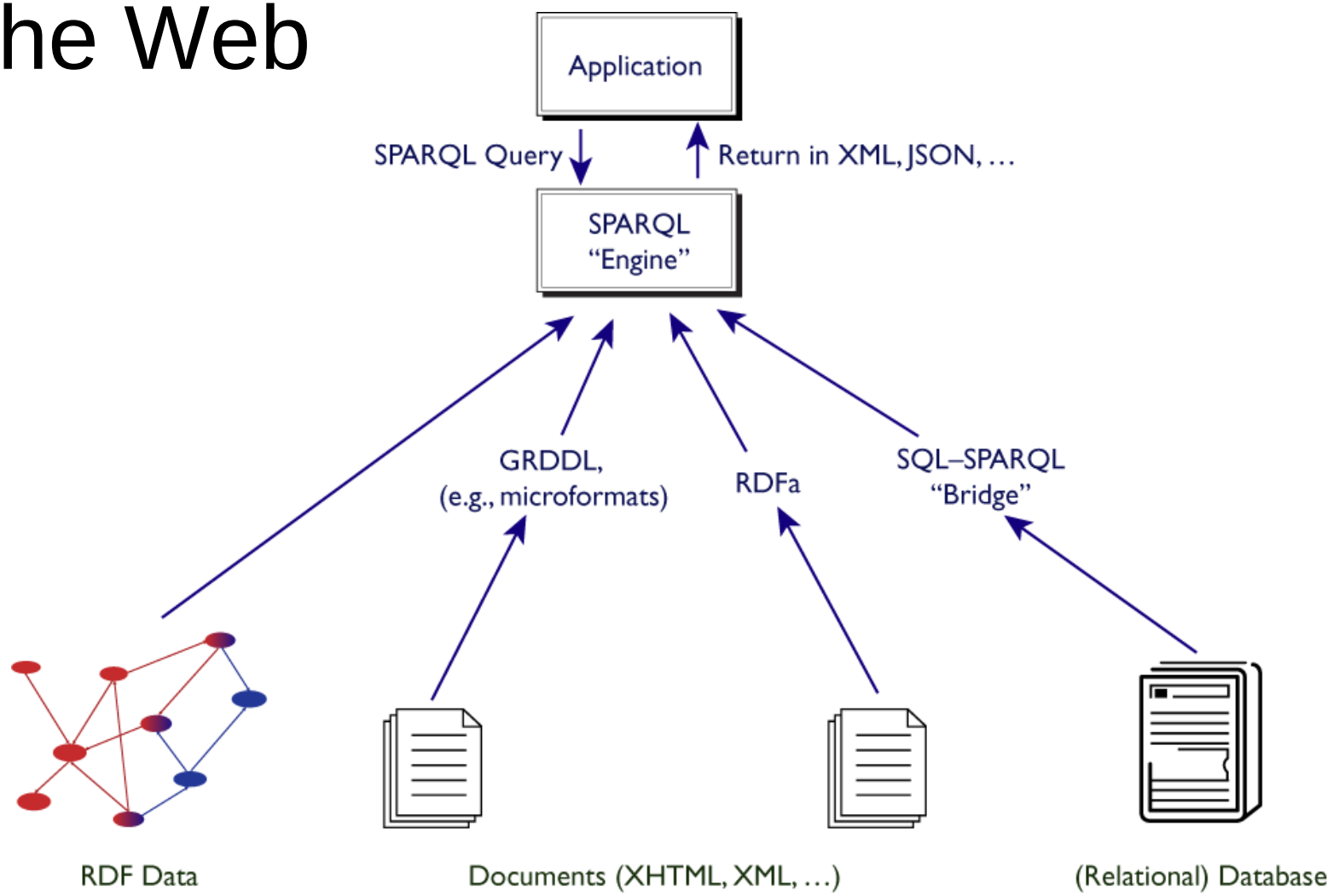
Providing RDF on the Web

- RDF data usually resides in a RDF database (triplestore)
  - …how do we ‘put them out’ on the web?
- SPARQL endpoints (SPARQL query over HTTP)
  - Direct connection to triplestore over HTTP

Example of SPARQL endpoint

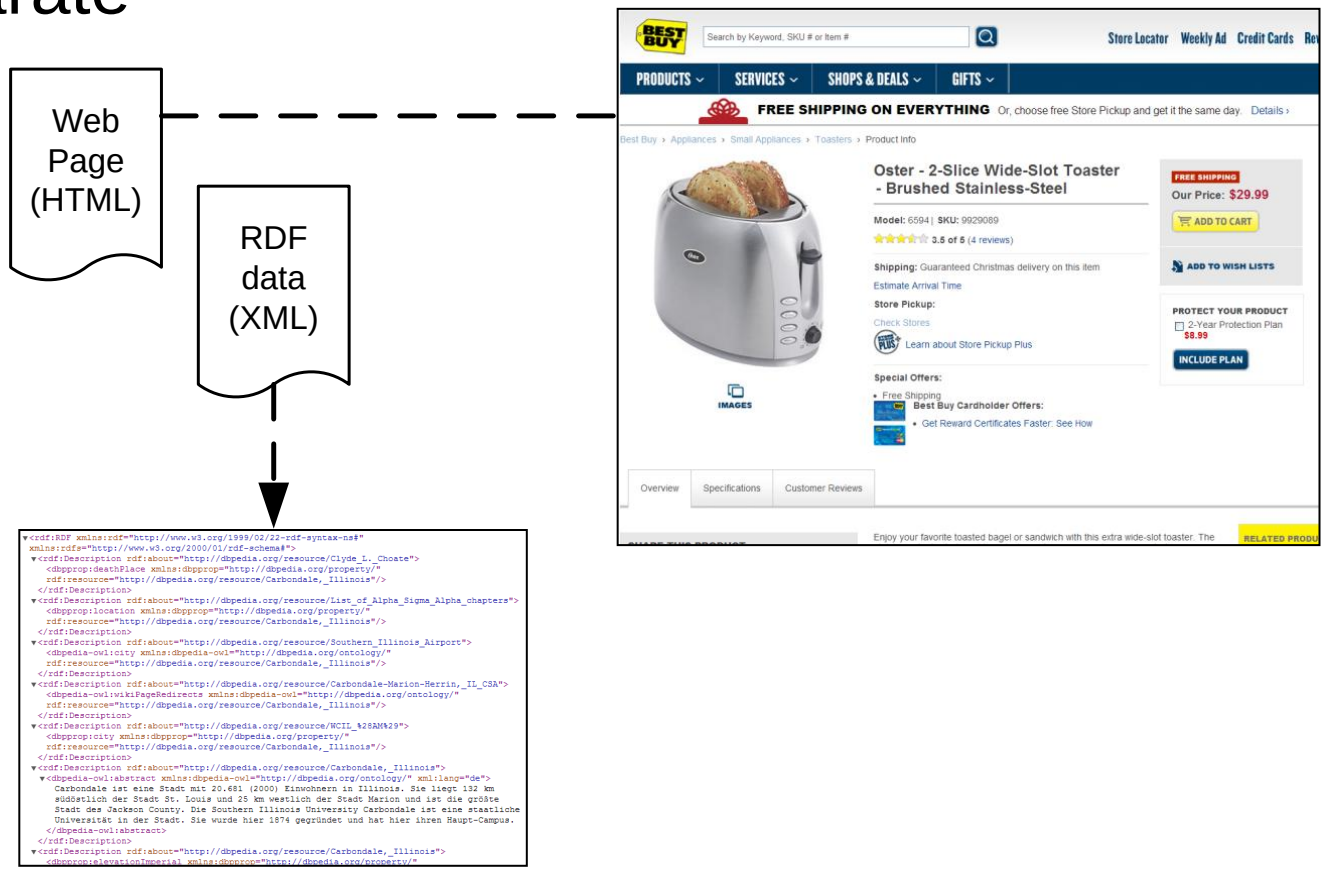


Providing RDF on the Web



Exposing RDF on the web

- Problem: usually HTML content and RDF data are separate

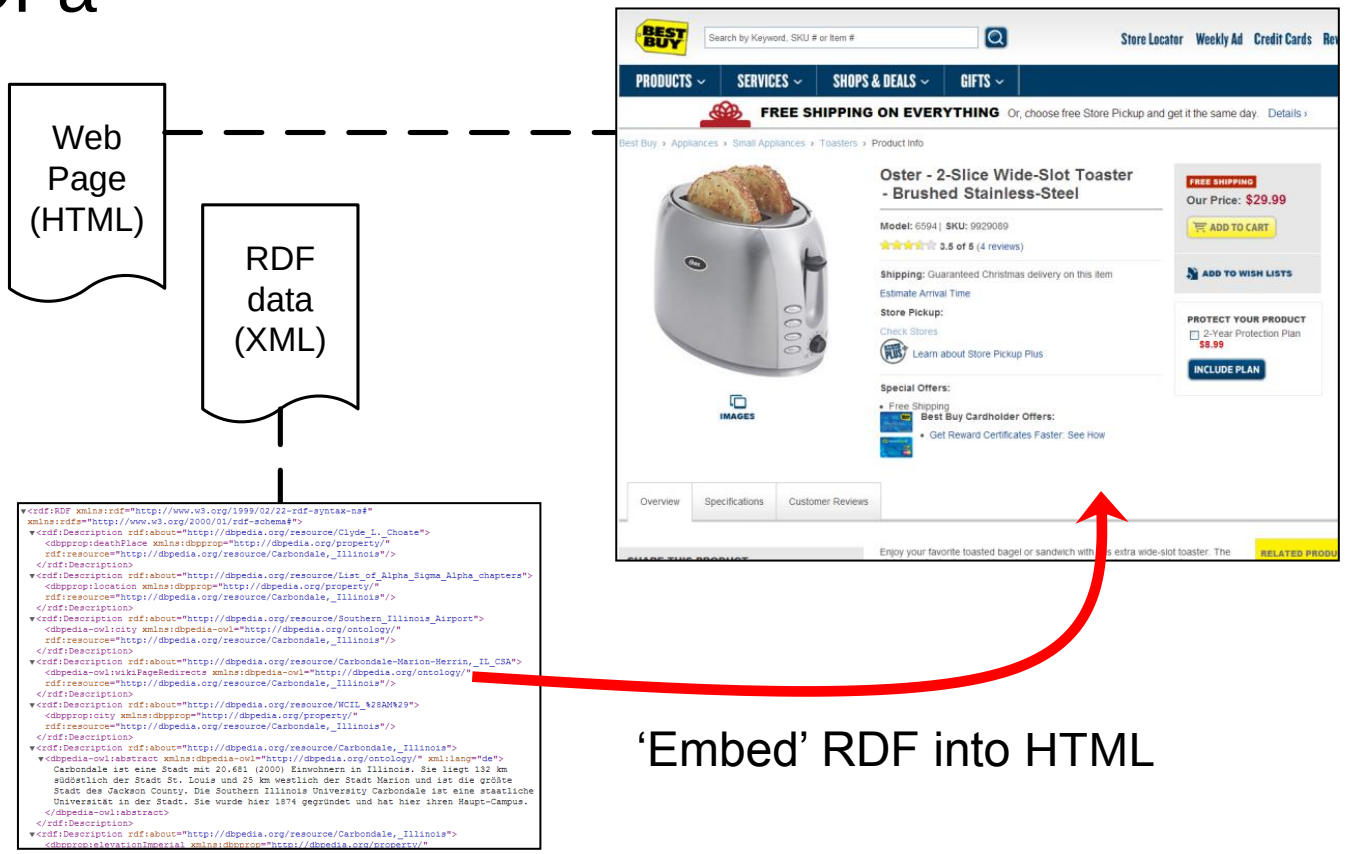


Exposing RDF on the web

- Separate HTML content and RDF data
- Maintenance problem
  - Both need to be managed separately
  - RDF content and web content have much overlap (redundancy)
  - RDF/XML difficult to author: extra overhead
- Verification problem
  - How to differences as content changes?
- Visibility problem
  - Easy to ignore the RDF content (out of sight, out of mind)

Exposing RDF on the web

- Solution: embed RDF into web content using RDFa

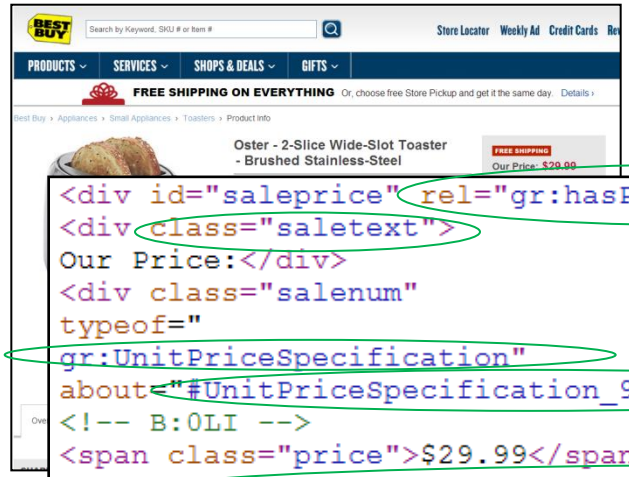


RDFa

- W3C Recommendation (October, 2008)
- Set of extensions to XHTML that allows to annotate XHTML markup with semantics
- Uses attributes from XHTML meta and link elements, and generalizes them so that they are usable on all elements
- A simple mapping is defined so that RDF triples may be extracted

Exposing RDF on the web

- RDFa: Resource Description Framework-in-attributes




```
<div id="saleprice" rel="gr:hasPriceSpecification">
<div class="saletext">
Our Price:</div>
<div class="salenum"
typeof="
gr:UnitPriceSpecification"
about="#UnitPriceSpecification_9929089_sale">
<!-- B:0LI -->
<span class="price">$29.99</span>
<span property="gr:hasCurrencyValue" datatype="xsd:float" content="29.99"></span>
<span property="gr:hasCurrency" datatype="xsd:string" content="USD"></span>
<span property="gr:hasUnitOfMeasurement" datatype="xsd:string" content="C62"></span>
<span property="gr:valueAddedTaxIncluded" datatype="xsd:boolean" content="true"></span>
<!-- E:0LI -->
</div>
</div>
```

- Extra (RDFa) markup is ignored by web browsers

XHTML + RDFa example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
   "http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml"
   xmlns:foaf="http://xmlns.com/foaf/0.1/"
   xmlns:dc="http://purl.org/dc/elements/1.1/"
   version="XHTML+RDFa 1.0" xml:lang="en">
  <head>
    <title>John's Home Page</title>
    <base href="http://example.org/john-d/" />
    <meta property="dc:creator" content="Jonathan Doe" />
  </head>
  <body>
    <h1>John's Home Page</h1>
    <p>My name is <span property="foaf:nick">John D</span> and I like
      <a href="http://www.neubauten.org/" rel="foaf:interest"
      xml:lang="de">Einstürzende Neubauten</a>. </p>
    <p> My <span rel="foaf:interest" resource="urn:ISBN:0752820907">
      favorite book</span> is the inspiring
      <span about="urn:ISBN:0752820907"><cite property="dc:title">
      Weaving the Web</cite> by <span property="dc:creator">Tim
      Berners-Lee</span></span> </p>
  </body>
</html>
```

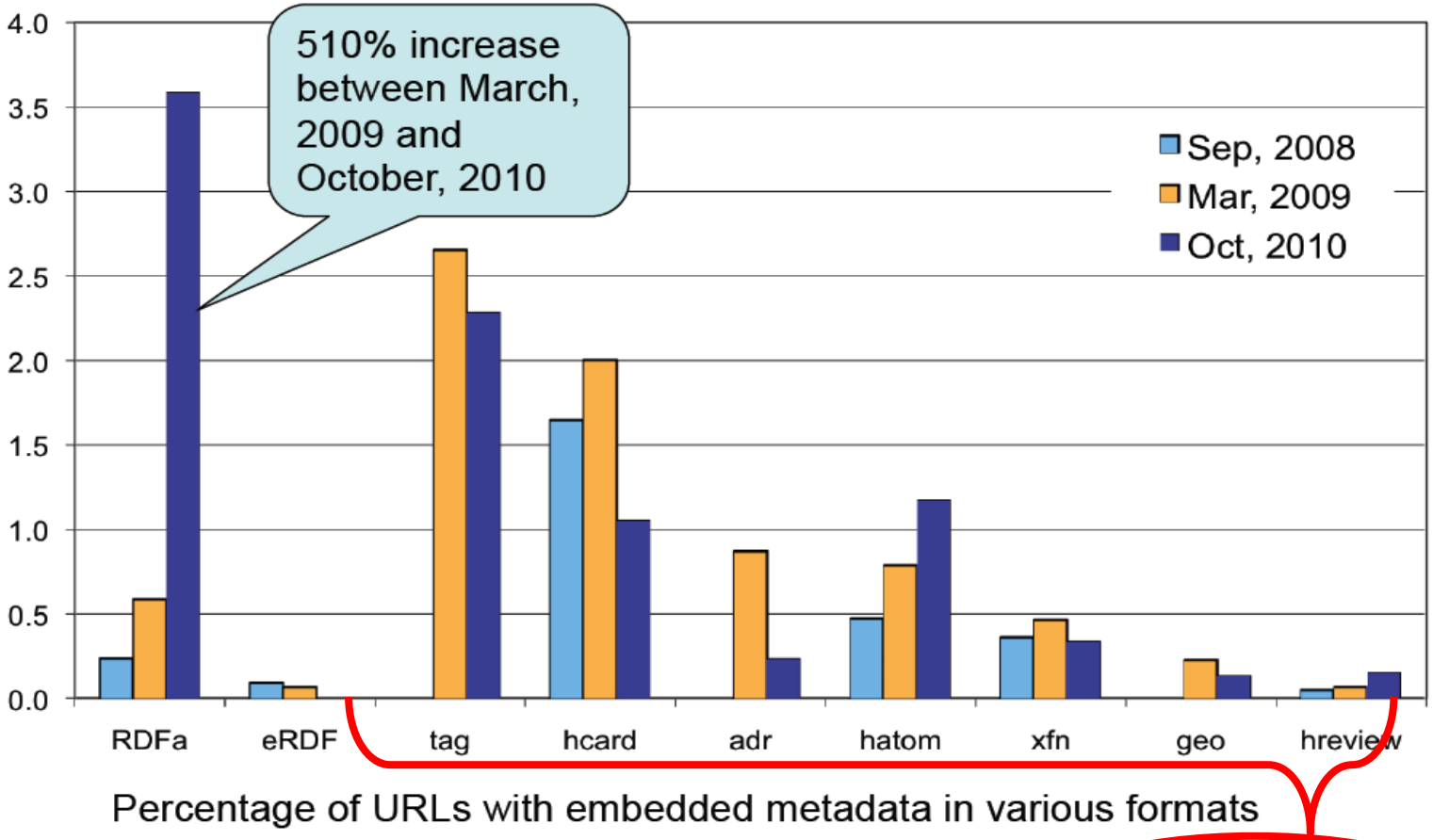
Automatic conversion to RDF/XML

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
     xmlns:foaf="http://xmlns.com/foaf/0.1/"
     xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://example.org/john-d/">
    <dc:creator xml:lang="en">Jonathan Doe</dc:creator>
    <foaf:nick xml:lang="en">John D</foaf:nick>
    <foaf:interest rdf:resource="http://www.neubauten.org/"/>
    <foaf:interest>
      <rdf:Description rdf:about="urn:ISBN:0752820907">
        <dc:creator xml:lang="en">Tim Berners-Lee</dc:creator>
        <dc:title xml:lang="en">Weaving the Web</dc:title>
      </rdf:Description>
    </foaf:interest>
  </rdf:Description>
</rdf:RDF>
```

RDFa annotations

- Less than 5% of web pages have RDFa annotations (Google, 2010)
- However, many organizations already publish or consume RDFa
  - Google, Yahoo
  - Facebook, MySpace, LinkedIn
  - Best Buy, Tesco, O'Reilly
  - SlideShare, Digg
  - WhiteHouse.gov, Library of Congress, UK government
  - Newsweek, BBC

RDFa is not the only solution …



Source: Peter Mika (Yahoo!), RDFa, 2011

microformats

Rich snippets

- Several solutions for embedding semantic data in Web
- Three syntaxes known (by Google) as “rich snippets”
  - Microformats
  - RDFa
  - HTML microdata
- All three are supported by Google, while microdata is the “recommended” syntax

First came microformats

- Microformats emerged around 2005
- Some key principles
  - Start by solving simple, specific problems
  - Design for humans first, machines second
- Wide deployment
  - Used on billions of Web pages
- Formats exist for marking up atom feeds, calendars, addresses and contact info, geo-location, multimedia, news, products, recipes, reviews, resumes, social relationships, etc.

Microformats example

```
<div class="vcard">
  <a class="fn org url"
     href="http://www.commerce.net/">CommerceNet</a>
  <div class="adr">
    <span class="type">Work</span>:
    <div class="street-address">169 University Avenue</div>
    <span class="locality">Palo Alto</span>,
    <abbr class="region"
      title="California">CA</abbr>  
    <span class="postal-code">94301</span>
    <div class="country-name">USA</div>
  </div>
  <div class="tel">
    <span class="type">Work</span> +1-650-289-4040
  </div>
  <div>Email:
    <span class="email">info@commerce.net</span>
  </div>
</div>
```

Then came RDFa

- RDFa aims to bridge the gap between human oriented HTML and machine-oriented RDF documents
- Provides XHTML attributes to indicate machine understandable information
- Uses the RDF data model, and Semantic Web vocabularies directly

RDFa example

```
<div typeof="foaf:Person"
             xmlns:foaf="http://xmlns.com/foaf/0.1/">
<p property="foaf:name">Alice Birpemswick</p>
<p>Email: <a rel="foaf:mbox"
   href="mailto:alice@example.com">alice@example.com</a>
</p>
<p>Phone: <a rel="foaf:phone" href="tel:+1-617-555-7332">
   +1 617.555.7332</a>
</p>
</div>
```

Last but not least, microdata

- Microdata syntax is based on nested groups of name-value pairs
- HTML microdata specification includes
  - An unambiguous parsing model
  - An algorithm to convert microdata to RDF
- Compatible with the Semantic Web via mappings

Microdata syntax

- Microdata properties
  - Annotate an item with text-valued properties using the “itemprop” attribute

```
<div itemscope>
  <p>My name is <span itemprop="name">Daniel</span>.</p>
</div>
```

Microdata syntax

- Multiple values are ok
  - As in RDF, you can have two properties, for the same item (subject) with the same value (object)

```
<div itemscope>
  <p>Flavors in my favorite ice cream:</p>
  <ul>
    <li itemprop="flavor">Lemon sorbet</li>
    <li itemprop="flavor">Apricot sorbet</li>
  </ul>
</div>
```

Microdata syntax

- Item types
  - Correspond to classes in RDF

```
<section itemscope
         itemtype="http://example.org/animals#cat">
  <h1 itemprop="name">Hedral</h1>
  <p itemprop="desc">Hedral is a male american
     domestic shorthair, with a fluffy black fur with
     white paws and belly.</p>
  <img itemprop="img" src="hedral.jpeg" alt=""
       title="Hedral, age 18 months">
</section>
```

Microdata syntax

- Global IDs
  - Items may be given global identifiers, which are URLs
  - They may be, but do not need to be Semantic Web URIs

```
<dl itemscope
  itemtype="http://vocab.example.net/book"
  itemid="urn:isbn:0-330-34032-8">
  <dt>Title</dt>
    <dd itemprop="title">The Reality Dysfunction</dd>
  <dt>Author</dt>
    <dd itemprop="author">Peter F. Hamilton</dd>
  <dt>Publication date</dt>
    <dd><time itemprop="pubdate" datetime="1996-01-26">
        26 January 1996</time></dd>
</dl>
```

Schema.org

- Schema.org is one of a number of microdata vocabularies
  - It is a shared collection of microdata schemas for use by webmasters
- Includes a type hierarchy, like an RDFS schema
  - Starts with top-level Thing (which has four properties: name, description, url, and image) and DataType types
  - More specific types share properties with broader types; for example, a Place is a more specific type of Thing, and a TouristAttraction is a more specific type of Place
  - More specific items inherit the properties of their parent

Example

```
<div itemscope itemtype="http://schema.org/Movie">
  <h1 itemprop="name"&g;Avatar</h1>
  <div itemprop="director" itemscope
    itemtype="http://schema.org/Person"> Director:
    <span itemprop="name">James Cameron</span>(born
    <span itemprop="birthDate">August 16, 1954)</span>
  </div>
  <span itemprop="genre">Science fiction</span>
  <a href="../movies/avatar-theatrical-trailer.html"
    itemprop="trailer">Trailer</a>
</div>
```

Current schema.org types

Schema.org full hierarchy

- **Thing**: description, image, name, url
  - **CreativeWork**: about, accountablePerson, aggregateRating, alternativeHeadline, associatedMedia, audio, author, award, awards, comment, contentLocation, contentRating, contributor, copyrightHolder, copyrightYear, creator, dateCreated, dateModified, datePublished, discussionUrl, editor, encoding, encodings, genre, headline, inLanguage, interactionCount, isFamilyFriendly, keywords, mentions, offers, provider, publisher, publishingPrinciples, review, reviews, sourceOrganization, text, thumbnailUrl, version, video
    - **Article**: articleBody, articleSection, wordCount
      - **BlogPosting**
      - **NewsArticle**: dateline, printColumn, printEdition, printPage, printSection
      - **ScholarlyArticle**
    - **Blog**: blogPost, blogPosts
    - **Book**: bookEdition, bookFormat, illustrator, isbn, numberOfPages
    - **Comment**
    - **ItemList**: itemListElement, itemListOrder
    - **Map**
    - **MediaObject**: associatedArticle, bitrate, contentSize, contentUrl, duration, embedUrl, encodesCreativeWork, encodingFormat, expires, height, interactionCount, offers, playerType, regionsAllowed, requiresSubscription, uploadDate, width
      - **AudioObject**: transcript
      - **ImageObject**: caption, exifData, representativeOfPage, thumbnail
      - **MusicVideoObject**
      - **VideoObject**: caption, productionCompany, thumbnail, transcript, videoFrameSize, videoQuality
    - **Movie**: actor, actors, director, duration, musicBy, producer, productionCompany, trailer

http://www.schema.org/docs/full.html

Item examples

Thing

The most generic type of item.

Property	Expected Type	Description
Properties from Thing		
description	Text	A short description of the item.
image	URL	URL of an image of the item.
name	Text	The name of the item.
url	URL	URL of the item.

More specific types

- CreativeWork
- Event
- Intangible
- Organization
- Person
- Place
- Product

Thing > Product

A product is anything that is made available for sale—for example, a pair of shoes, a concert ticket, or a car.

Property	Expected Type	Description
Properties from Thing		
description	Text	A short description of the item.
image	URL	URL of an image of the item.
name	Text	The name of the item.
url	URL	URL of the item.
Properties from Product		
aggregateRating	AggregateRating	The overall rating, based on a collection of reviews or ratings, of the item.
brand	Organization	The brand of the product.
manufacturer	Organization	The manufacturer of the product.
model	Text	The model of the product.
offers	Offer	An offer to sell this item—for example, an offer to sell a product, the DVD of a movie, or tickets to an event.
productID	Text	The product identifier, such as ISBN. For example: `<meta itemprop='productID' content='isbn:123-456-789'/>`.
review	Review	A review of the item.
reviews	Review	Review of the item (legacy spelling; see singular form, review).

SPARQL use in practice

- Where to find meaningful RDF data to search?
- The Linked Data Project

The Linked Data Project

The Linked Data Project

- A fundamental prerequisite of the Semantic Web is the existence of large amounts of meaningfully interlinked RDF data on the Web
- “To make the Semantic Web a reality, it is necessary to have a large volume of data available on the Web in a standard, reachable and manageable format. In addition the relationships among data also need to be made available. This collection of interrelated data on the Web can also be referred to as Linked Data. Linked Data lies at the heart of the Semantic Web: large scale integration of, and reasoning on, data on the Web.” (W3C)

The Linked Data Project

- Linked Data is about using the Web to connect related data that wasn't previously linked, or using the Web to lower the barriers to linking data currently linked using other methods
- Linked Data is a set of principles that allows publishing, querying and browsing of RDF data, distributed across different servers

The Linked Data Project

- Community effort to make various open datasets available on the Web as RDF and to set RDF links between data items from different datasets
- The datasets are published according to the Linked Data principles and can therefore be crawled by Semantic Web search engines and navigated using Semantic Web browsers
- Supported by W3C
- Began early 2007
  - http://linkeddata.org/home

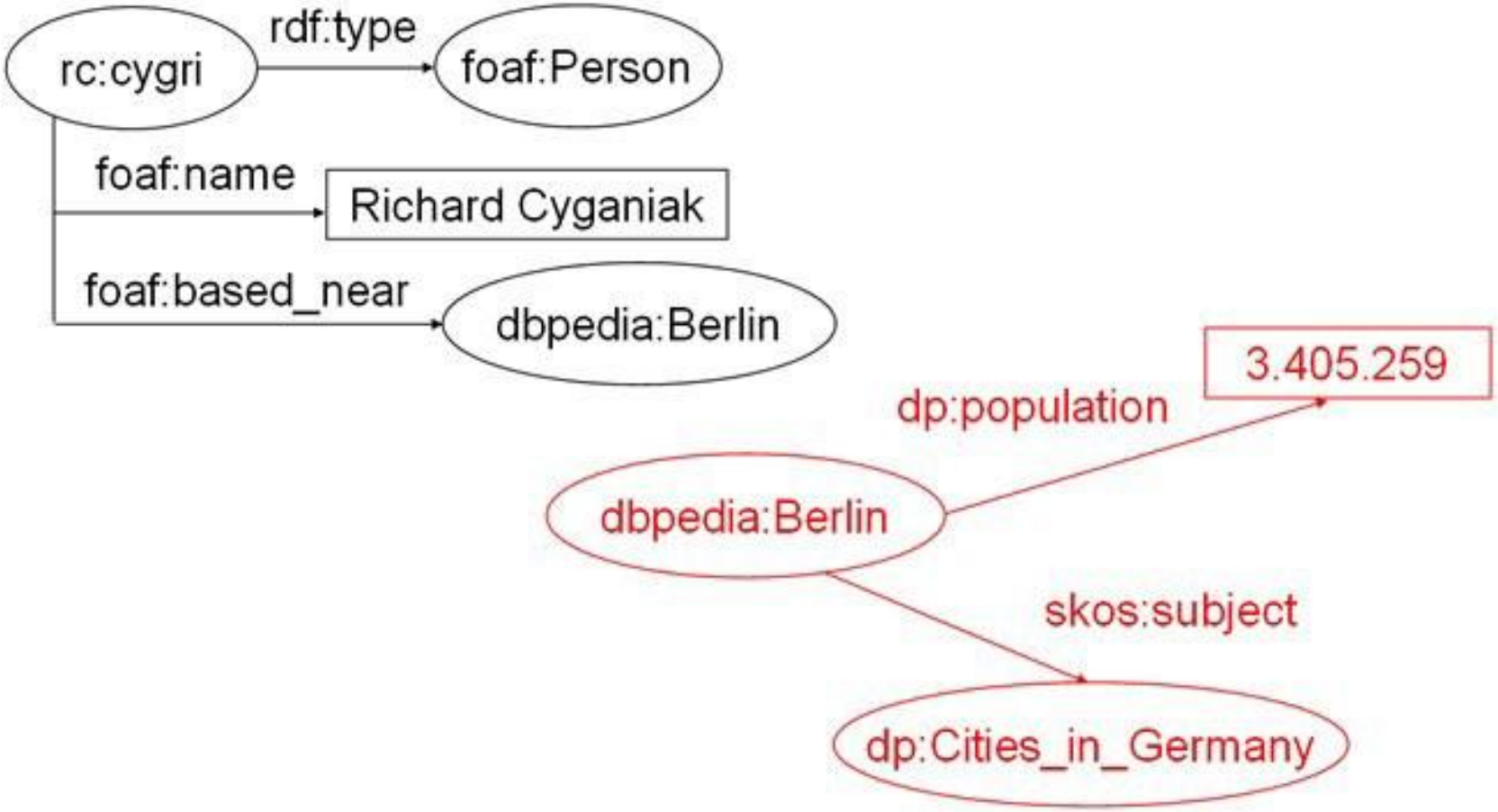
The Web of Documents

- Analogy: a global filesystem
- Designed for human consumption
- Primary objects: documents
- Links between documents (or sub-parts)
- Degree of structure in objects: fairly low
- Semantics of content and links: implicit

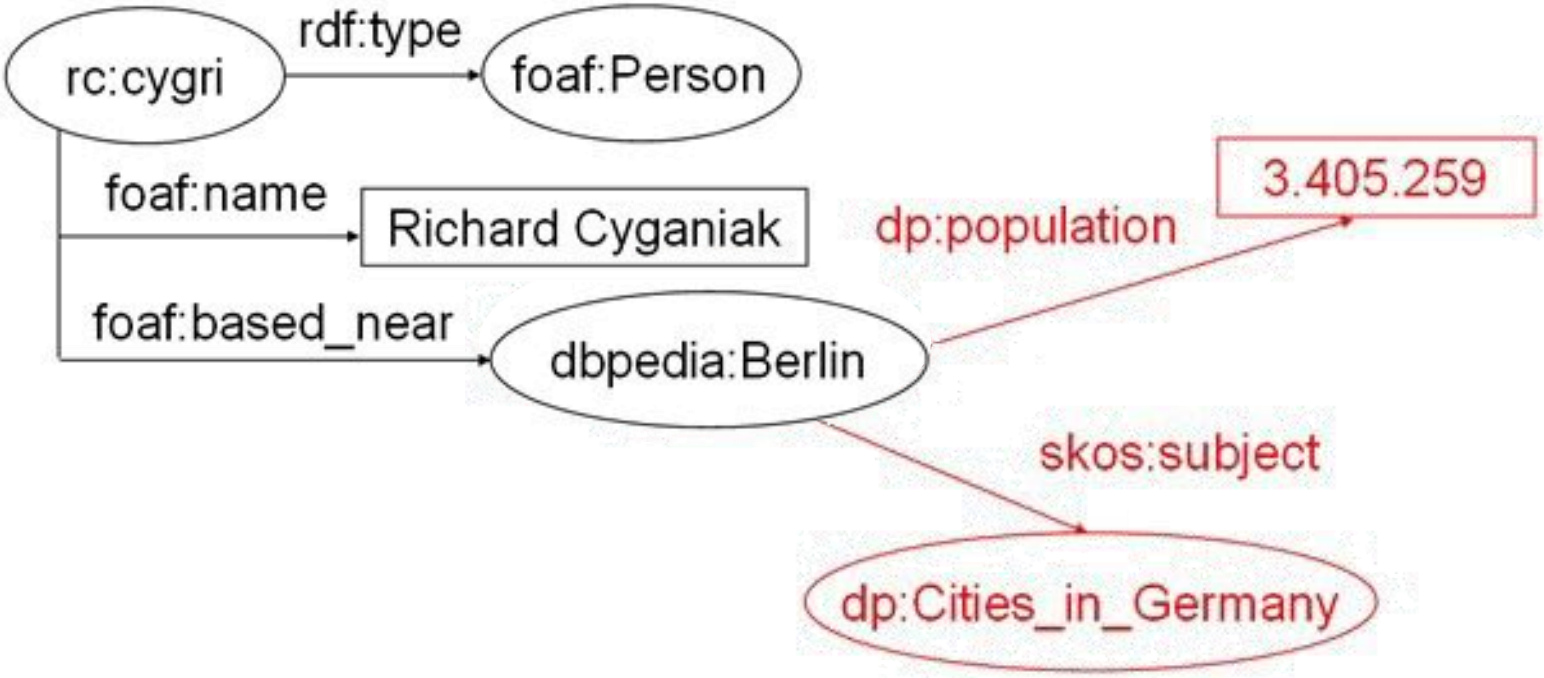
The Web of Linked Data

- Analogy: a global database
- Designed for machines first, humans later
- Primary objects: things (or descriptions of things)
- Links between things
- Degree of structure in (descriptions of) things: high
- Semantics of content and links: explicit

Linked Data example



Linked Data example

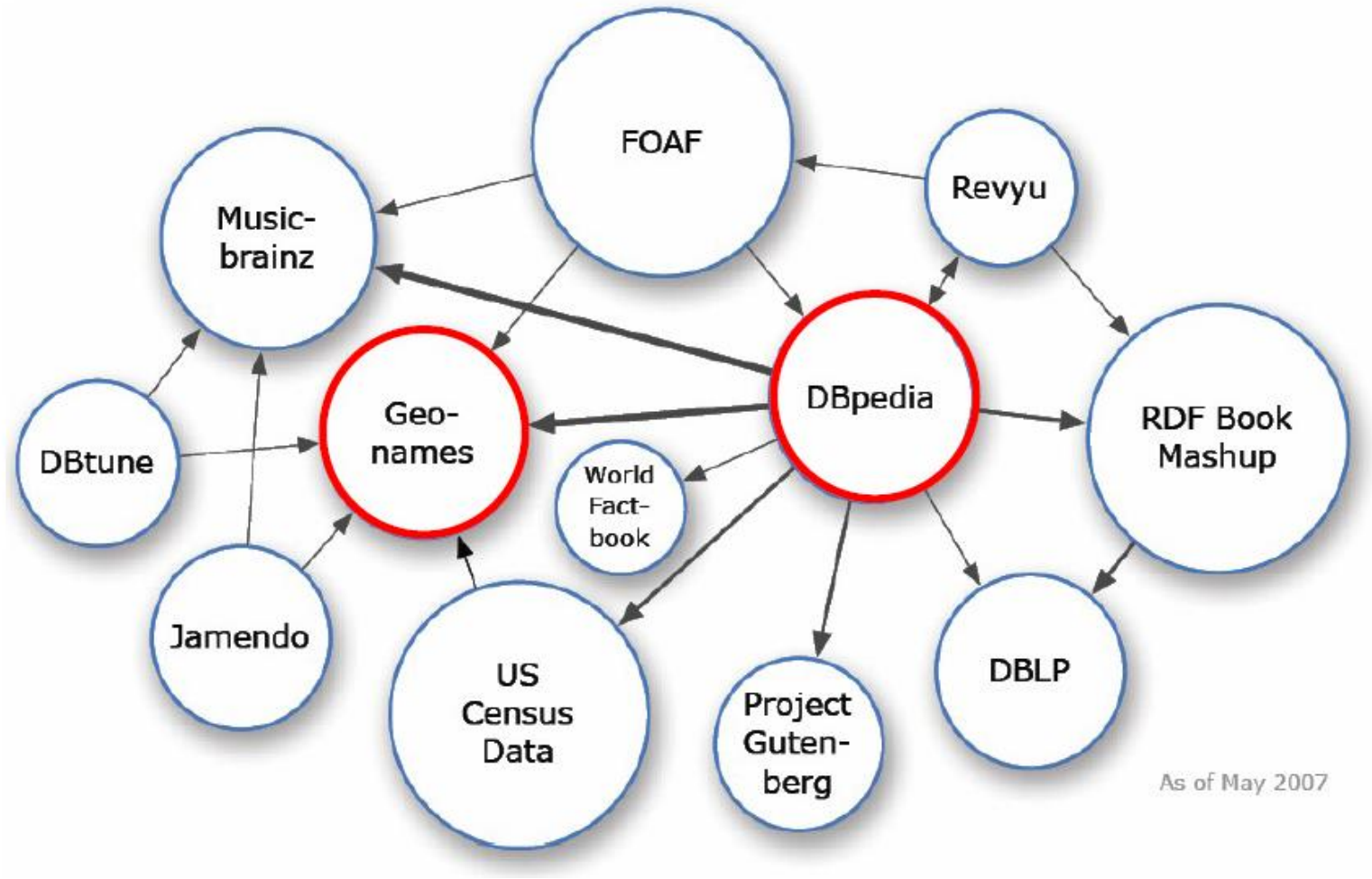


Why publish Linked Data?

- Ease of discovery
- Ease of consumption
  - Standards-based data sharing
- Reduced redundancy
- Added value
  - Build ecosystems around your data/content

Linked Open Data cloud

May 2007



DBpedia

- DBpedia is a community effort to extract structured information from Wikipedia and to make this information available on the Web
- DBpedia allows to ask sophisticated queries against Wikipedia, and to link other data sets on the Web to Wikipedia data

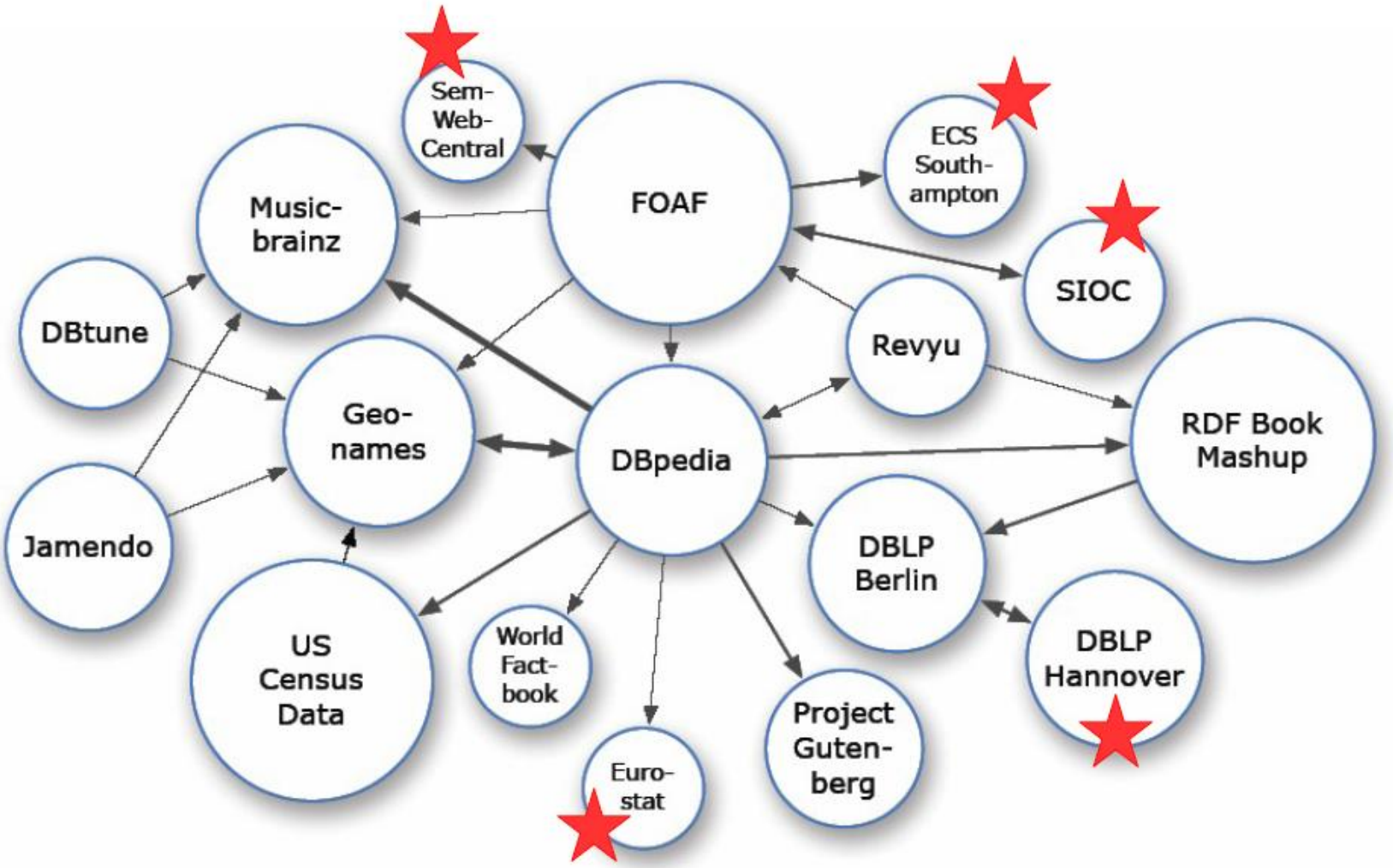
GeoNames

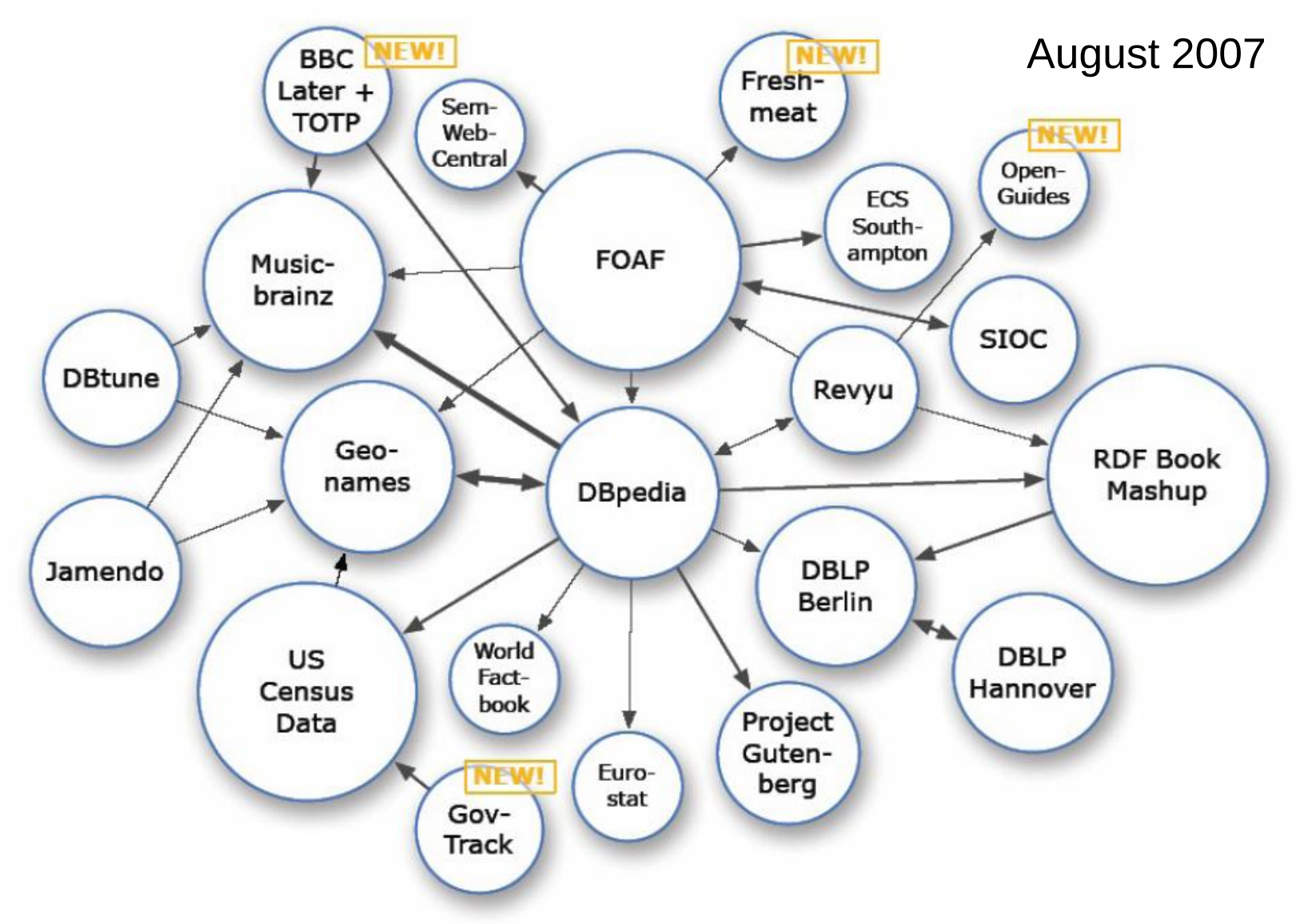
- GeoNames is a geographical database that contains over eight million geographical names
- Available for download free of charge under a creative commons attribution license

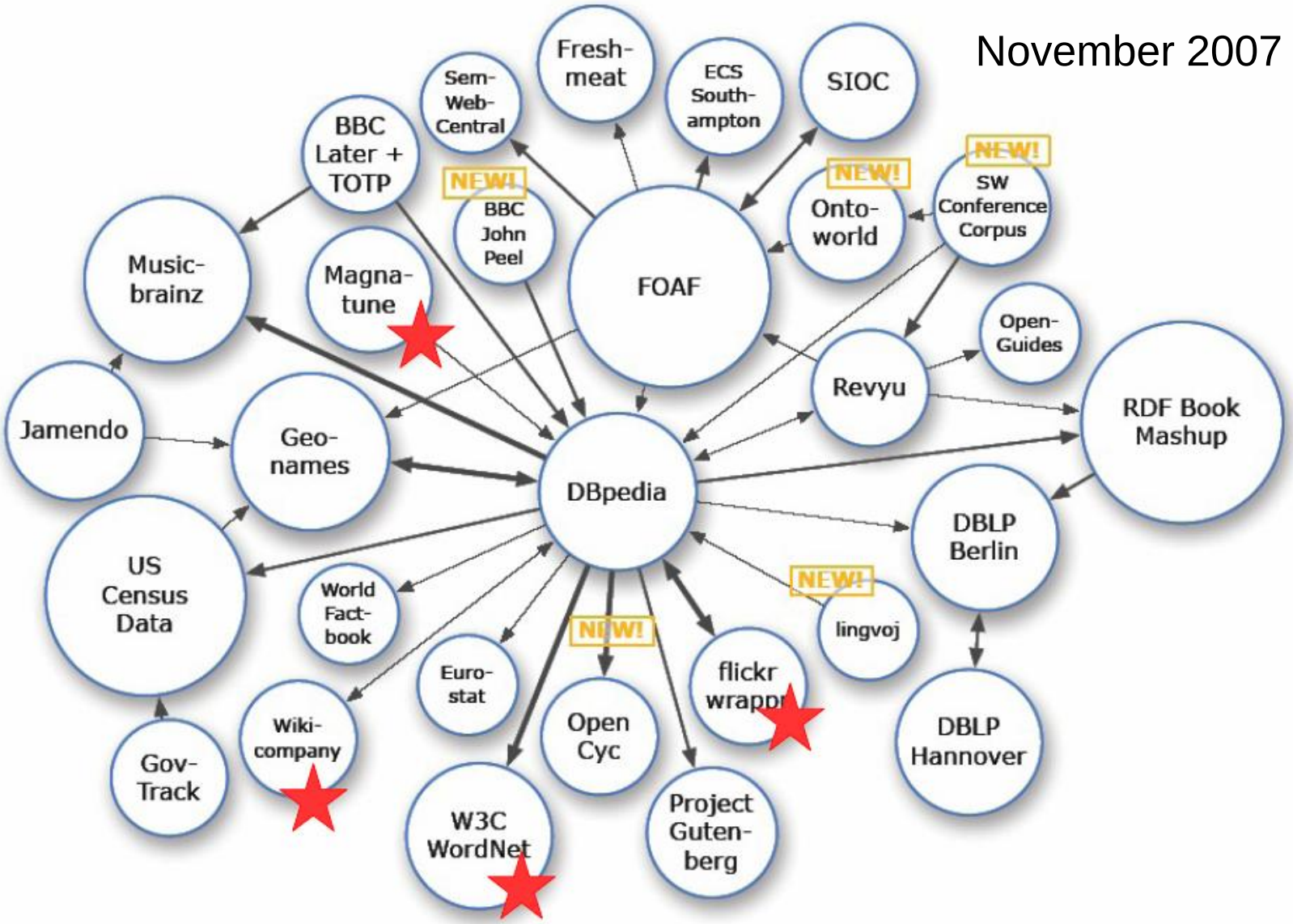
Main contributors

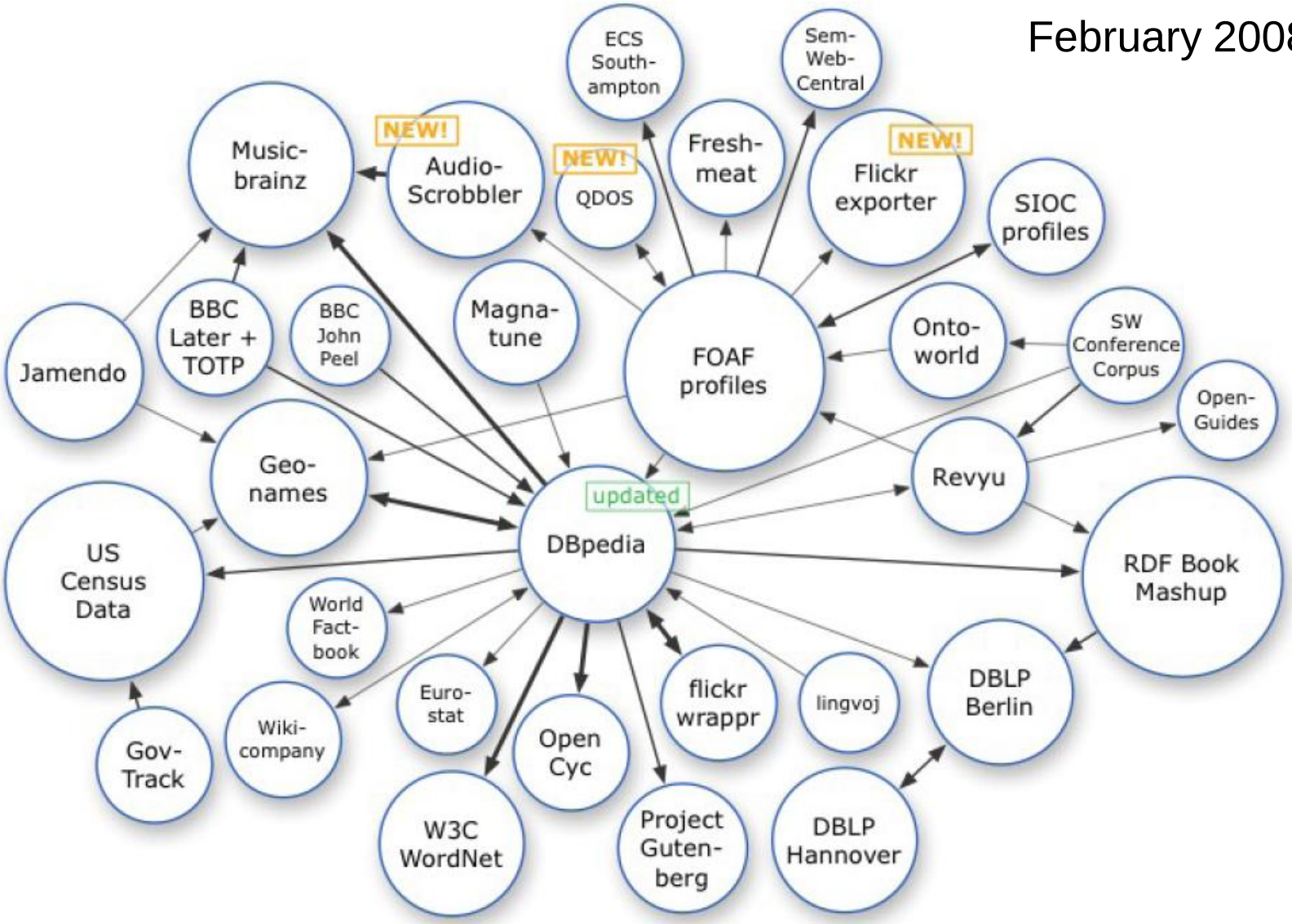
- **DBLP** Computer science bibliography
  - Richard Cyganiak, Chris Bizer (FU Berlin)
- **DBpedia** Structured information from Wikipedia
  - Universität Leipzig, FU Berlin, OpenLink
- **DBtune, Jamendo** Creative Commons music repositories
  - Yves Raimond (University of London)
- **Geonames** World-wide geographical database
  - Bernard Vatant (Mondeca), Marc Wick (Geonames)
- **Musicbrainz** Music and artist database
  - Frederick Giasson, Kingsley Idehen (Zitgist)
- **Project Gutenberg** Literary works in the public domain
  - Piet Hensel, Hans Butschalowsky (FU Berlin)
- **Revyu** Community reviews about anything
  - Tom Heath, Enrico Motta (Open University)
- **RDF Book Mashup** Books from the Amazon API
  - Tobias Gauß, Chris Bizer (FU Berlin)
- **US Census Data** Statistical information about the U.S.
  - Josh Tauberer (University of Pennsylvania), OpenLink
- **World Factbook** Country statistics, compiled by CIA
  - Piet Hensel, Hans Butschalowsky (FU Berlin)

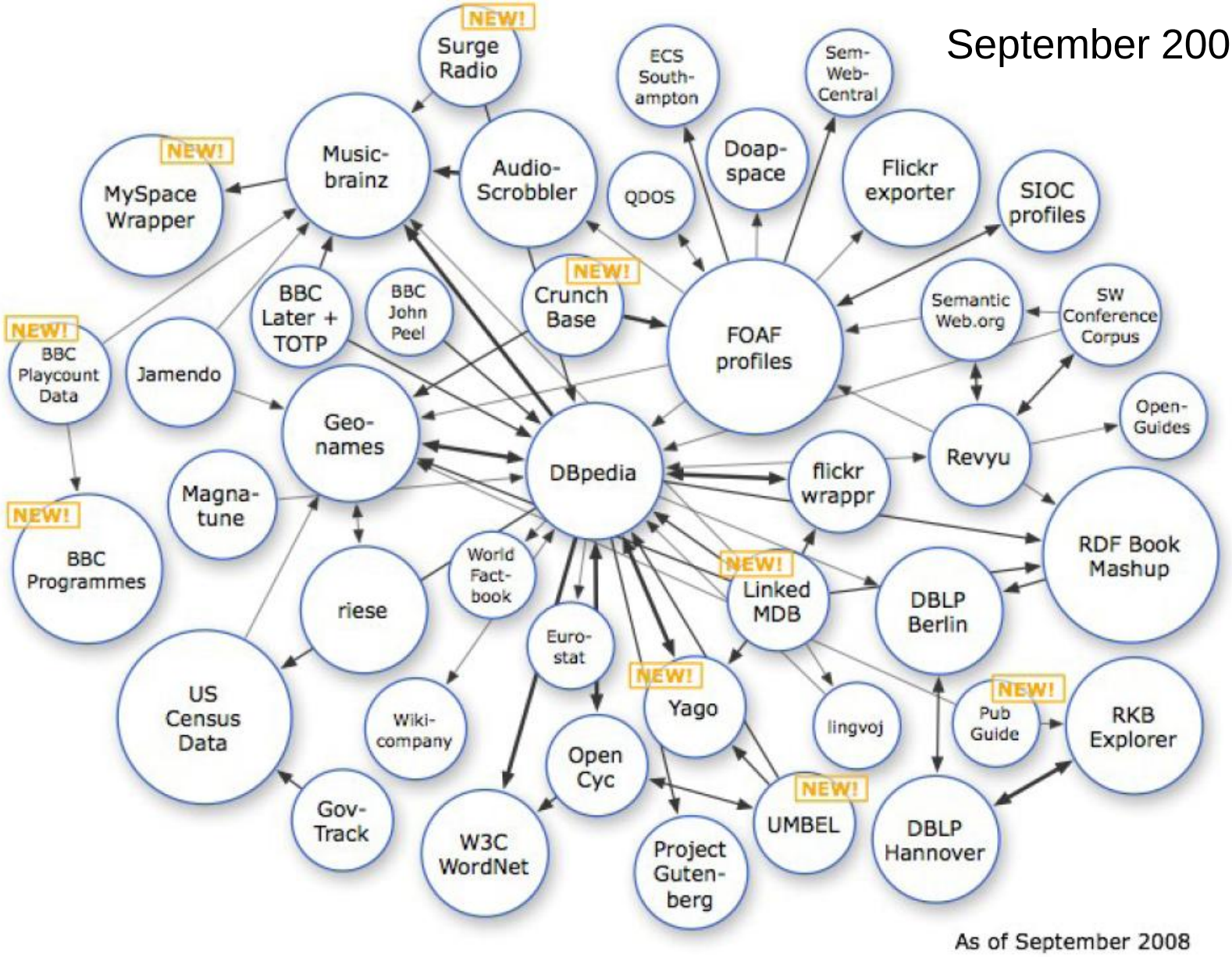
July 2007


Sem-
Web-
Central
FOAF
ECS
South-
ampton
Music-
brainz
SIOC
DBtune
Revyu
Geo-
names
DBpedia
RDF Book
Mashup
Jamendo
DBLP
Berlin
DBLP
Hannover
US
Census
Data
World
Fact-
book
Project
Guten-
berg
Euro-
stat

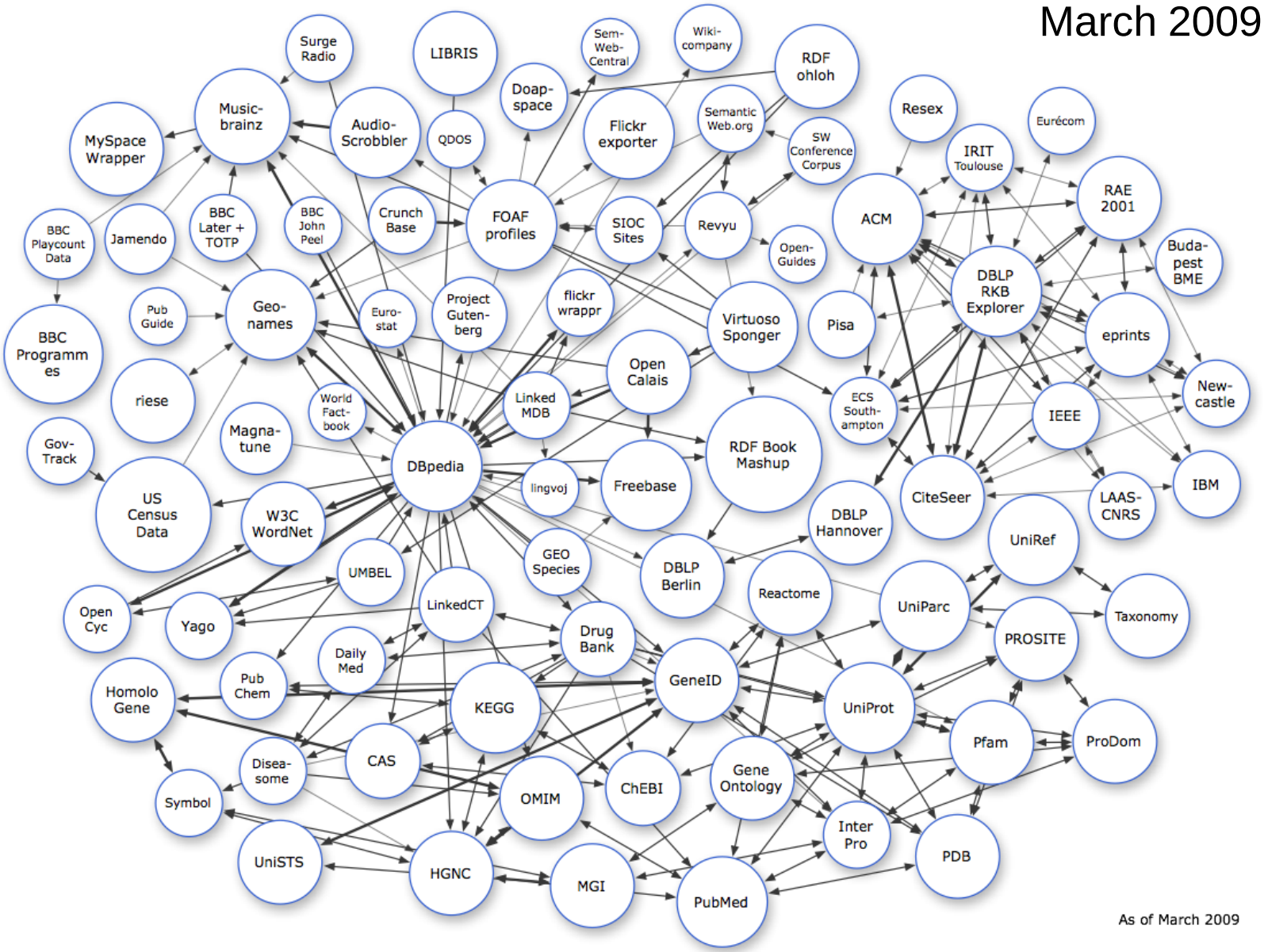

August 2007
BBC Later + TOTP
NEW!
Sem-Web-Central
Fresh-meat
NEW!
Open-Guides
NEW!
ECS South-ampton
Music-brainz
FOAF
SIOC
DBtune
Revyu
Geo-names
DBpedia
RDF Book Mashup
Jamendo
DBLP Berlin
DBLP Hannover
US Census Data
World Fact-book
Project Guten-berg
Euro-stat
NEW!
Gov-Track


November 2007
Fresh-meat
ECS South-ampton
SIOC
Sem-Web-Central
BBC Later + TOTP
NEW!
BBC John Peel
NEW!
Onto-world
NEW!
SW Conference Corpus
Music-brainz
Magna-tune
FOAF
Open-Guides
Revyu
RDF Book Mashup
Jamendo
Geo-names
DBpedia
DBLP Berlin
US Census Data
World Fact-book
NEW!
lingvoj
Euro-stat
NEW!
flickr wrappr
Open Cyc
Wiki-company
DBLP Hannover
Gov-Track
W3C WordNet
Project Guten-berg

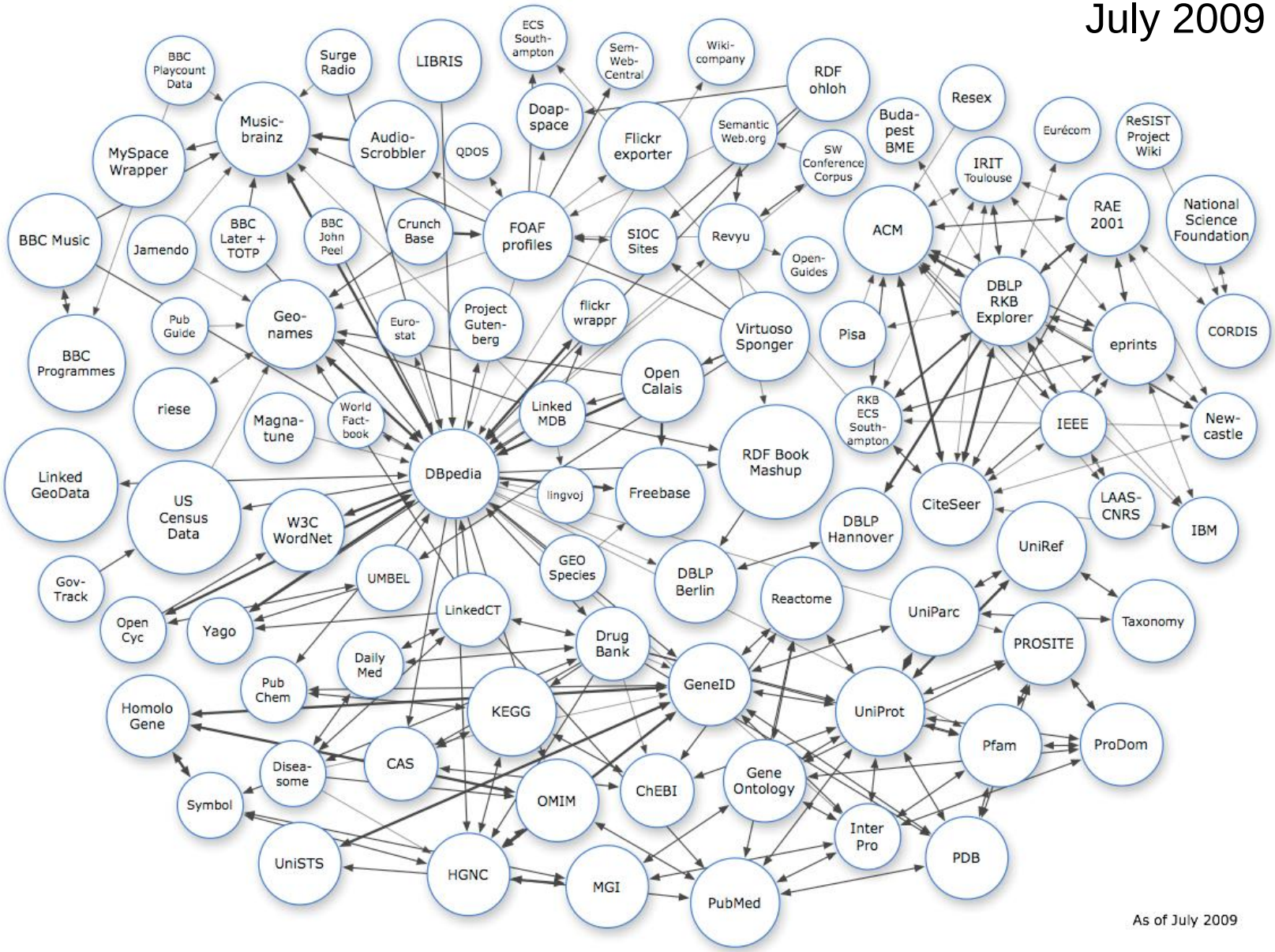

February 2008
ECS South-ampton
Sem-Web-Central
Music-brainz
NEW!
Audio-Scrobbler
NEW!
QDOS
Fresh-meat
NEW!
Flickr exporter
SIOC profiles
BBC Later + TOTP
BBC John Peel
Magna-tune
FOAF profiles
Onto-world
SW Conference Corpus
Jamendo
Open-Guides
Geo-names
updated
Revyu
US Census Data
DBpedia
RDF Book Mashup
World Fact-book
DBLP Berlin
Euro-stat
flickr wrappr
lingvoj
Wiki-company
Open Cyc
Gov-Track
W3C WordNet
Project Guten-berg
DBLP Hannover


September 2008
NEW!
Surge Radio
ECS South-ampton
Sem-Web-Central
Music-brainz
Audio-Scrobbler
Doap-space
Flickr exporter
SIOC profiles
NEW!
MySpace Wrapper
QDOS
NEW!
Crunch Base
FOAF profiles
Semantic Web.org
SW Conference Corpus
BBC Later + TOTP
BBC John Peel
NEW!
BBC Playcount Data
Jamendo
Geo-names
DBpedia
flickr wrappr
Revyu
Open-Guides
Magna-tune
NEW!
BBC Programmes
World Fact-book
RDF Book Mashup
NEW!
Linked MDB
DBLP Berlin
riese
Euro-stat
US Census Data
NEW!
Yago
NEW!
Pub Guide
RKB Explorer
Wiki-company
lingvoj
Open Cyc
Gov-Track
NEW!
UMBEL
W3C WordNet
Project Guten-berg
DBLP Hannover
As of September 2008

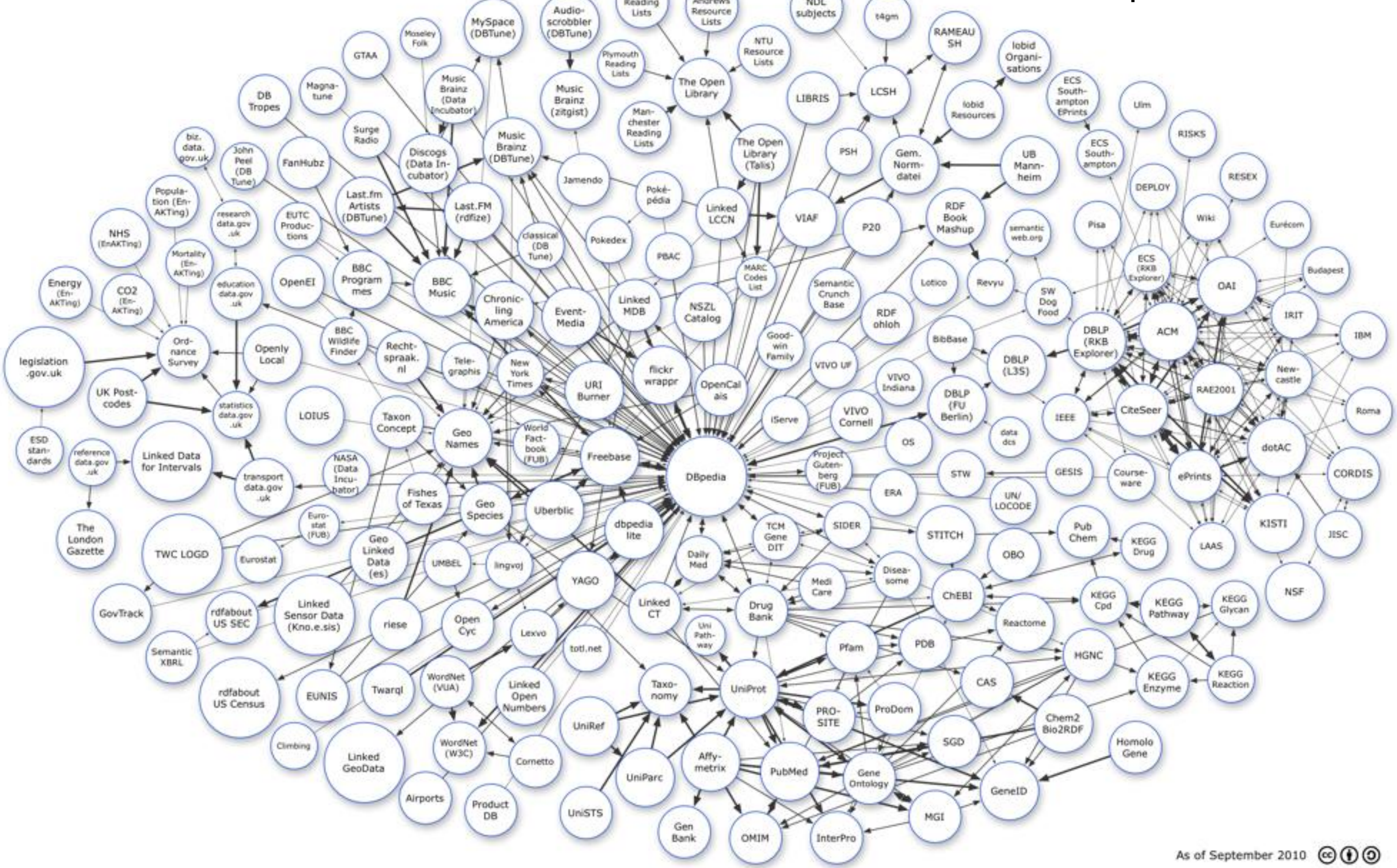
March 2009


Surge Radio
LIBRIS
Sem-Web-Central
Wiki-company
RDF ohloh
Music-brainz
Audio-Scrobbler
Doap-space
Flickr exporter
Semantic Web.org
SW Conference Corpus
Resex
Eurécom
IRIT Toulouse
MySpace Wrapper
QDOS
RAE 2001
BBC Playcount Data
Jamendo
BBC Later + TOTP
BBC John Peel
Crunch Base
FOAF profiles
SIOC Sites
Revyu
Open-Guides
ACM
Buda-pest BME
DBLP RKB Explorer
Pub Guide
Geo-names
Euro-stat
Project Guten-berg
flickr wrappr
Virtuoso Sponger
Pisa
eprints
BBC Programmes
Open Calais
riese
World Fact-book
Linked MDB
ECS South-ampton
IEEE
New-castle
Gov-Track
Magna-tune
DBpedia
RDF Book Mashup
lingvoj
Freebase
CiteSeer
LAAS-CNRS
IBM
US Census Data
W3C WordNet
DBLP Hannover
UniRef
UMBEL
GEO Species
DBLP Berlin
Reactome
UniParc
Taxonomy
Open Cyc
Yago
LinkedCT
Drug Bank
Daily Med
PROSITE
Pub Chem
GeneID
Homolo Gene
KEGG
UniProt
Pfam
ProDom
Disea-some
CAS
Gene Ontology
Symbol
OMIM
ChEBI
Inter Pro
UniSTS
HGNC
MGI
PubMed
PDB
As of March 2009

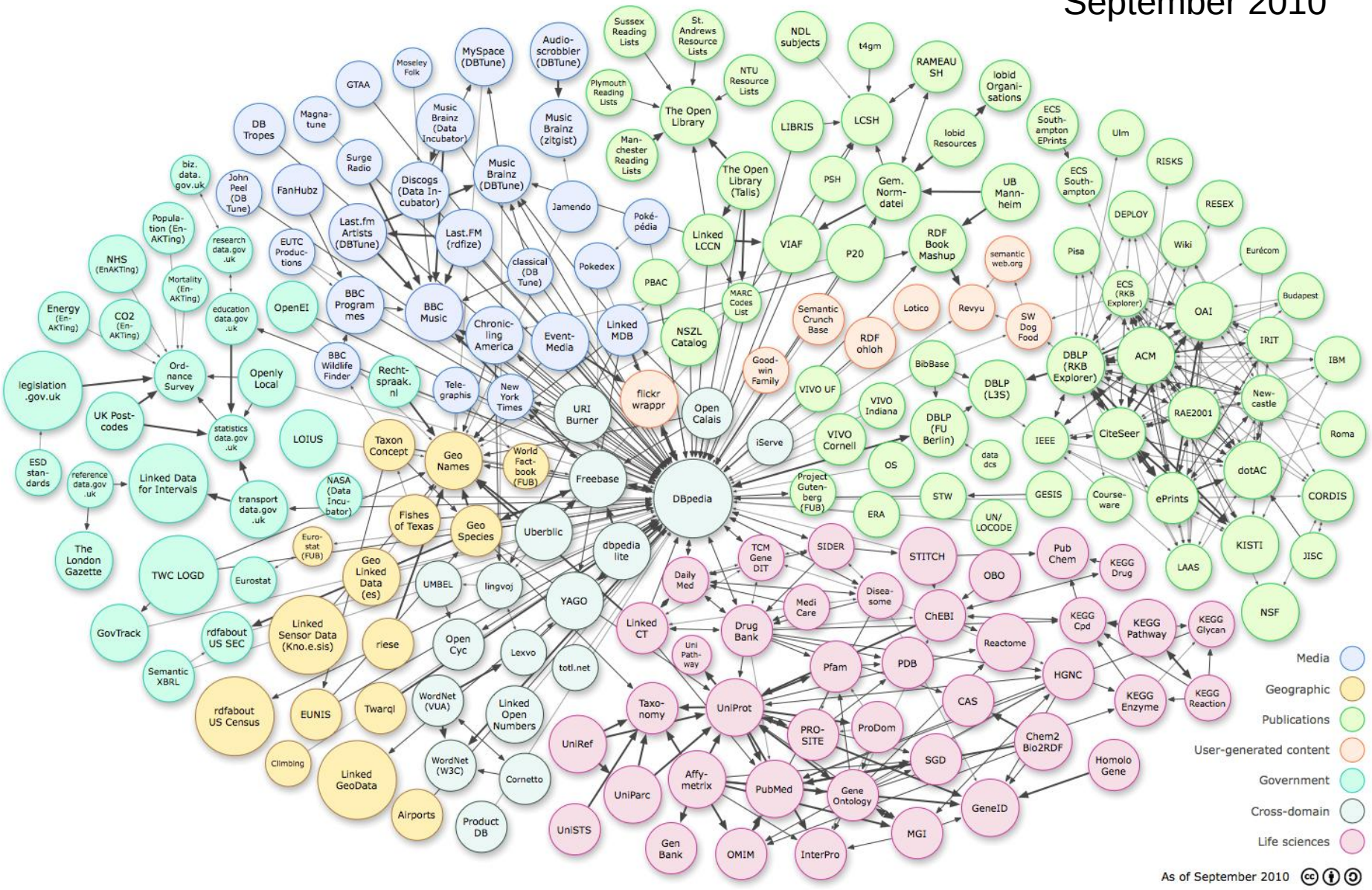
July 2009


BBC Playcount Data
Surge Radio
LIBRIS
ECS South-ampton
Sem-Web-Central
Wiki-company
RDF ohloh
Music-brainz
Audio-Scrobbler
QDOS
Doap-space
Flickr exporter
Semantic Web.org
SW Conference Corpus
Buda-pest BME
Resex
Eurécom
ReSIST Project Wiki
MySpace Wrapper
IRIT Toulouse
RAE 2001
National Science Foundation
BBC Music
Jamendo
BBC Later + TOTP
BBC John Peel
Crunch Base
FOAF profiles
SIOC Sites
Revyu
Open-Guides
ACM
DBLP RKB Explorer
CORDIS
Pub Guide
Geo-names
Euro-stat
Project Guten-berg
flickr wrappr
Virtuoso Sponger
Pisa
eprints
BBC Programmes
riese
Magna-tune
World Fact-book
Linked MDB
Open Calais
RKB ECS South-ampton
IEEE
New-castle
Linked GeoData
US Census Data
DBpedia
lingvoj
Freebase
RDF Book Mashup
CiteSeer
LAAS-CNRS
IBM
W3C WordNet
DBLP Hannover
UniRef
Gov-Track
UMBEL
GEO Species
DBLP Berlin
Reactome
Open Cyc
Yago
LinkedCT
Drug Bank
UniParc
Taxonomy
Daily Med
GeneID
PROSITE
Pub Chem
UniProt
Homolo Gene
KEGG
Pfam
ProDom
Disea-some
CAS
Gene Ontology
Symbol
OMIM
ChEBI
Inter Pro
UniSTS
HGNC
MGI
PubMed
PDB
As of July 2009

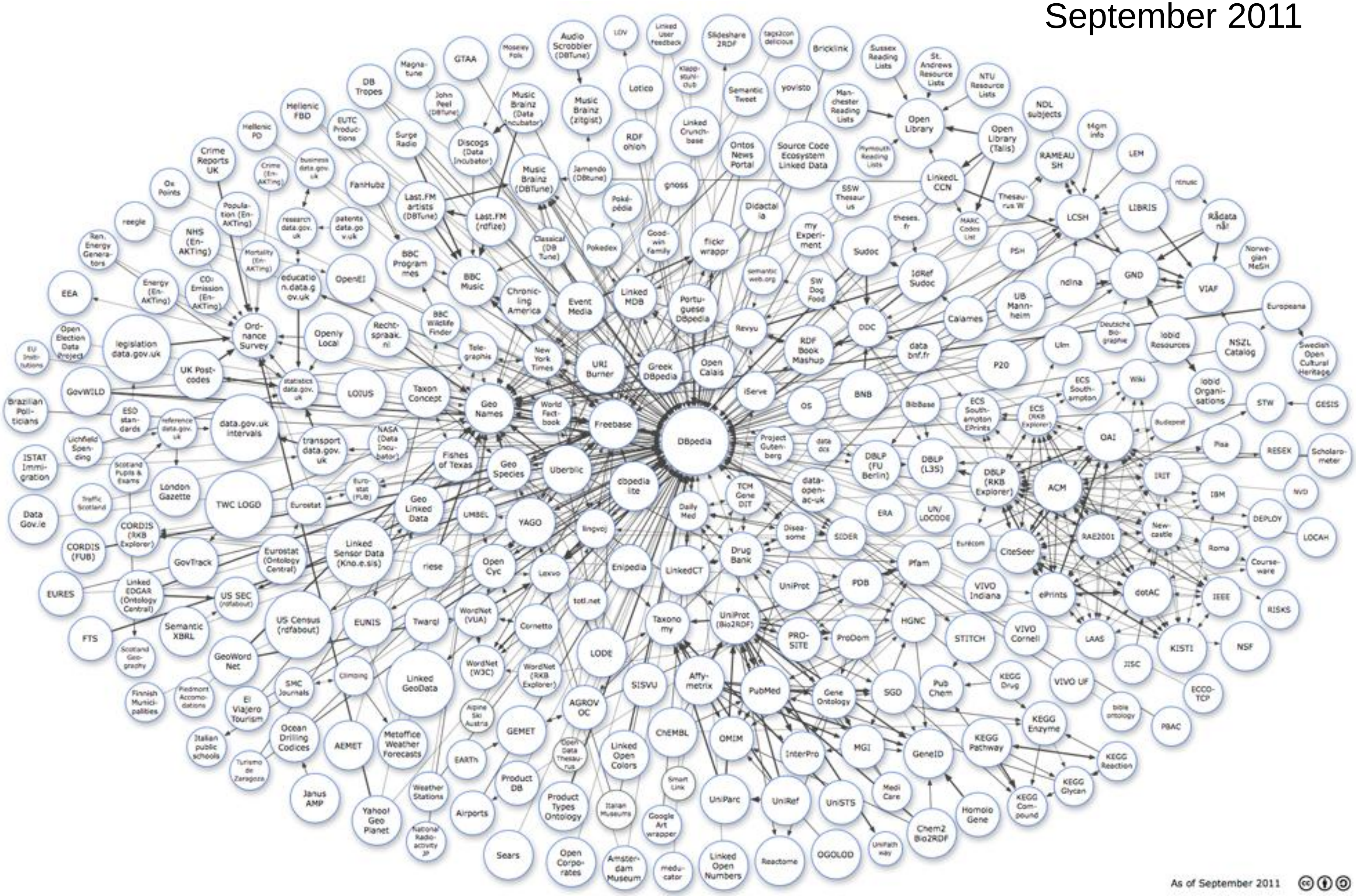
September 2010



September 2010



September 2011



Statistics on datasets

- http://ckan.net/group/lodcloud
- http://www4.wiwiss.fu-berlin.de/lodcloud/

Data set	Size of the data set (number of triples)	Wrapper?	endpoint?	RDF dump?
ACM (RKB)	12,644,052	N	Y	Y
AudioScrobbler	600,000,000	Y	Y	N
BBC John Peel	277,000	N	Y	Y
BBC Later + TOTP (link not responding - 2009-04-01)	10,000	N	Y	Y
BBC Music	>10,000,000	N	N	N
BBC Playcount Data	10,000	N	Y	Y
BBC Programmes	10,000,000	N	N	N
Budapest BME (RKB)	42,064	N	Y	Y
Bio2RDF:Affymetrix	45,560,115	N	Y	Y
Bio2RDF:BioCYC	18,699,622	N	Y	Y
Bio2RDF:EBI:ChEBI	7,376,253	N	Y	Y
Bio2RDF:DBpedia	190,790	N	Y	Y
Bio2RDF:GO	8,188,649	N	Y	Y
Bio2RDF:HGNC	1,208,802	N	Y	Y
Bio2RDF:KEGG:Compound	177,199	N	Y	Y
Bio2RDF:KEGG:Drug	116,822	N	Y	Y

Statistics on links between datasets

Source data set	Type	Target data set	Link count (range)	Link count (actual)
ACM (RKB)	Data Set	DBLP (RKB)	> 100,000	834,923
ACM (RKB)	Data Set	LAAS CNRS (RKB)	> 100	173
ACM (RKB)	Data Set	Newcastle (RKB)	> 100	684
ACM (RKB)	Data Set	eprints (RKB)	> 10,000	59,081
ACM (RKB)	Data Set	IRIT Toulouse (RKB)	> 100	729
ACM (RKB)	Data Set	CiteSeer (RKB)	> 1,000,000	1,768,94
ACM (RKB)	Data Set	Pisa (RKB)	> 100	
ACM (RKB)	Data Set	R e s e x (RKB)	> 100	
ACM (RKB)	Data Set	IBM (RKB)	> 100	116
ACM (RKB)	Data Set	IEEE (RKB)	> 1,000	2,267
ACM (RKB)	Data Set	RAE 2001 (RKB)	> 1,000	4,534
ACM (RKB)	Data Set	ECS Southampton (RKB)	> 1,000	1,358
AudioScrobbler	Wrapper	Musicbrainz	> 100	
AudioScrobbler	Wrapper	FOAF profiles	> 100,000	
BBC John Peel	Data Set	DBpedia	> 1,000	
BBC Later + TOTP	Data Set	DBpedia	> 1,000	

Linked Data success stories

- BBC Music
  - Integrates information from MusicBrainz and Wikipedia for artist/band infopages
  - Information also available in RDF (in addition to web pages)
  - 3rd party applications built on top of the BBC data
  - BBC also contributes data back to the MusicBrainz
- Nytimes
  - Maps its thesaurus of 1 million entity descriptions (people, organisations, places, etc) to Dbpedia and Freebase

Linked Data shopping list

- List of sites/datasets that the “community” would like to see published as Linked Data
  - This list may form the basis for some campaign/action to encourage these data publishers to embrace Linked Data

- http://linkeddata.org/linked-data-shopping-list

The Linked Data principles ("expectations of behavior")

- The Semantic Web isn't just about putting data on the web. It is about making links, so that a person or machine can explore the web of data. With linked data, when you have some of it, you can find other, related, data
- It is the unexpected re-use of information which is the value added by the web

(Tim Berners-Lee)

The Linked Data principles ("expectations of behavior")

- Unambiguous identifiers for objects (resources)
  - Use URIs as names for things
- Use the structure of the web
  - Use HTTP URIs so that people can look up the names
- Make is easy to discover information about an object (resource)
  - When someone lookups a URI, provide useful information
- Link the object (resource) to related objects
  - Include links to other URIs

Link to existing vocabularies

- For describing classes (categories) and properties (relationships), try to re-use existing vocabularies
  - Easier to interoperate if we're talking the same language!
- Many vocabularies/ontologies out there
  - schema.org is a great place to start looking!
  - Vocabs for products (Good Relations), people (FOAF), social media (SIOC), places, events, businesses, e-commerce, music, etc., you name it…
- If nothing relevant, you can create your own, but make sure you…
  - Publish it!
  - Reconcile (map) it to other vocabularies, if you can

Link to other datasets

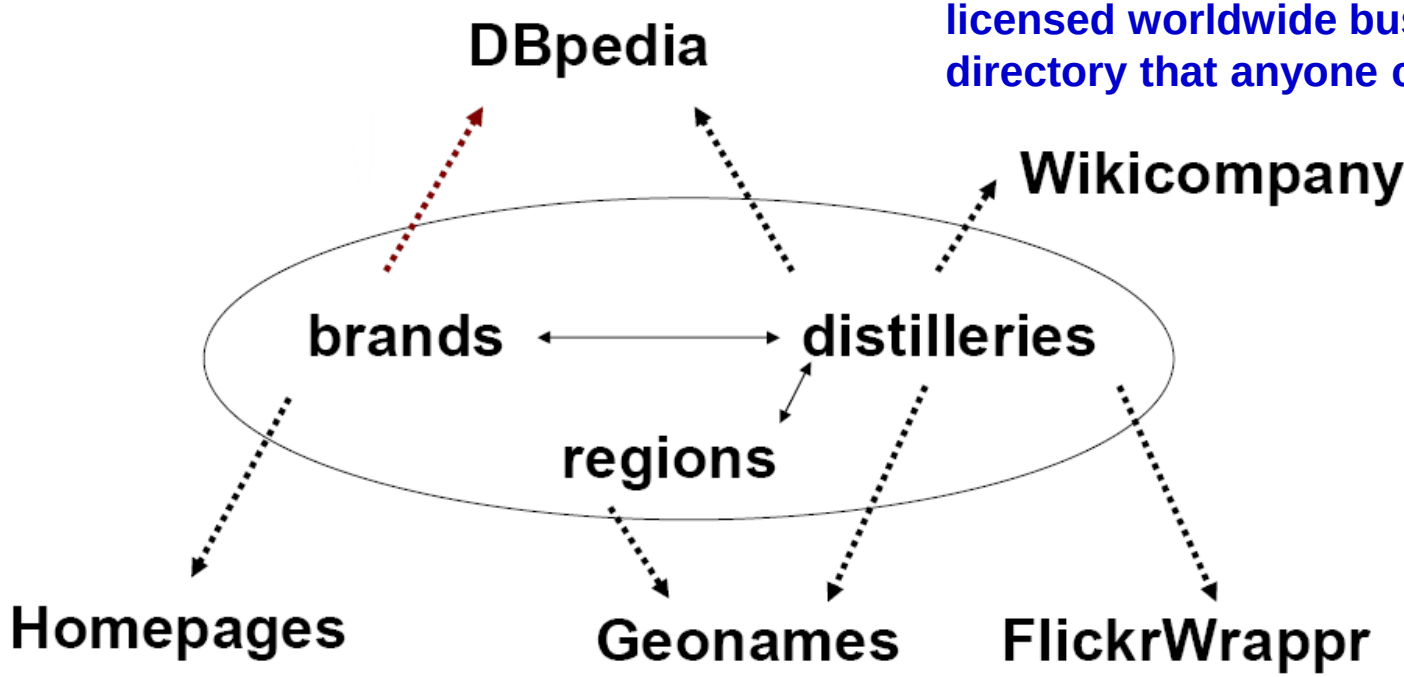
- Popular predicates for linking
  - owl:equivalentClass
  - owl:sameAs
  - rdfs:seeAlso
  - skos:closeMatch
  - skos:exactMatch
  - skos:related
  - foaf:homepage
  - foaf:topic
  - foaf:based_near
  - foaf:maker/foaf:made
  - foaf:page
  - foaf:primaryTopic
- Example:

http://dbpedia.org/resource/Canberra

owl:sameAs

http://rdf.freebase.com/rdf/en.canberra

Link to other Data Sets



(Wikicompany is a free content licensed worldwide business directory that anyone can edit)

(flickr wrappr extends DBpedia with RDF links to photos posted on flickr)

Linked Data – open issues

- Schema diversity & proliferation
- Quality of data is poor
- No kind of consistency is guarantees
- Issues with reliability of data end-points
  - High down-time is not unusual
  - There is no Service Level Agreement provided
- Querying of linked data is slow
  - Data is distributed on the web
  - Even single SPARQL endpoints can be slow
  - Most end-points are experimental/research projects with no resources for quality guarantees
- Licensing issues
  - Majority of datasets carry no explicit open license

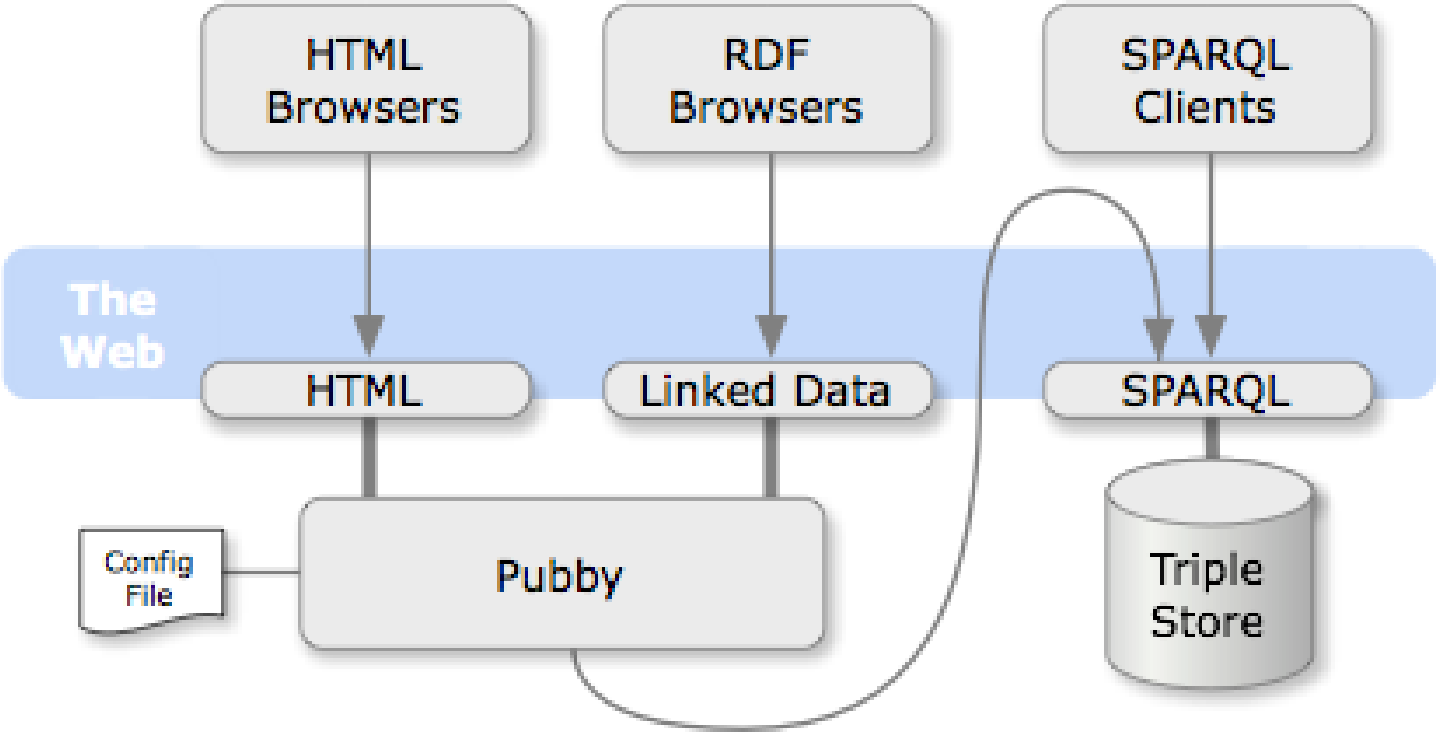
Linked Data tools

- Tools for Publishing Linked Data
  - D2R Server: a tool for publishing relational databases as Linked Data
  - Triplify: transforms relational data into RDF/LinkedData
  - Pubby: a Linked Data frontend for SPARQL endpoints
- Tools for consuming Linked Data
  - Semantic Web Browsers and Client Libraries
  - Semantic Web Search Engines

Pubby

- Many triple stores and other SPARQL endpoints can be accessed only by SPARQL client applications that use the SPARQL protocol
  - It cannot be accessed by the growing variety of Linked Data clients
- Pubby is designed to provide a Linked Data interface to those RDF data sources
- http://www4.wiwiss.fu-berlin.de/pubby/

Pubby

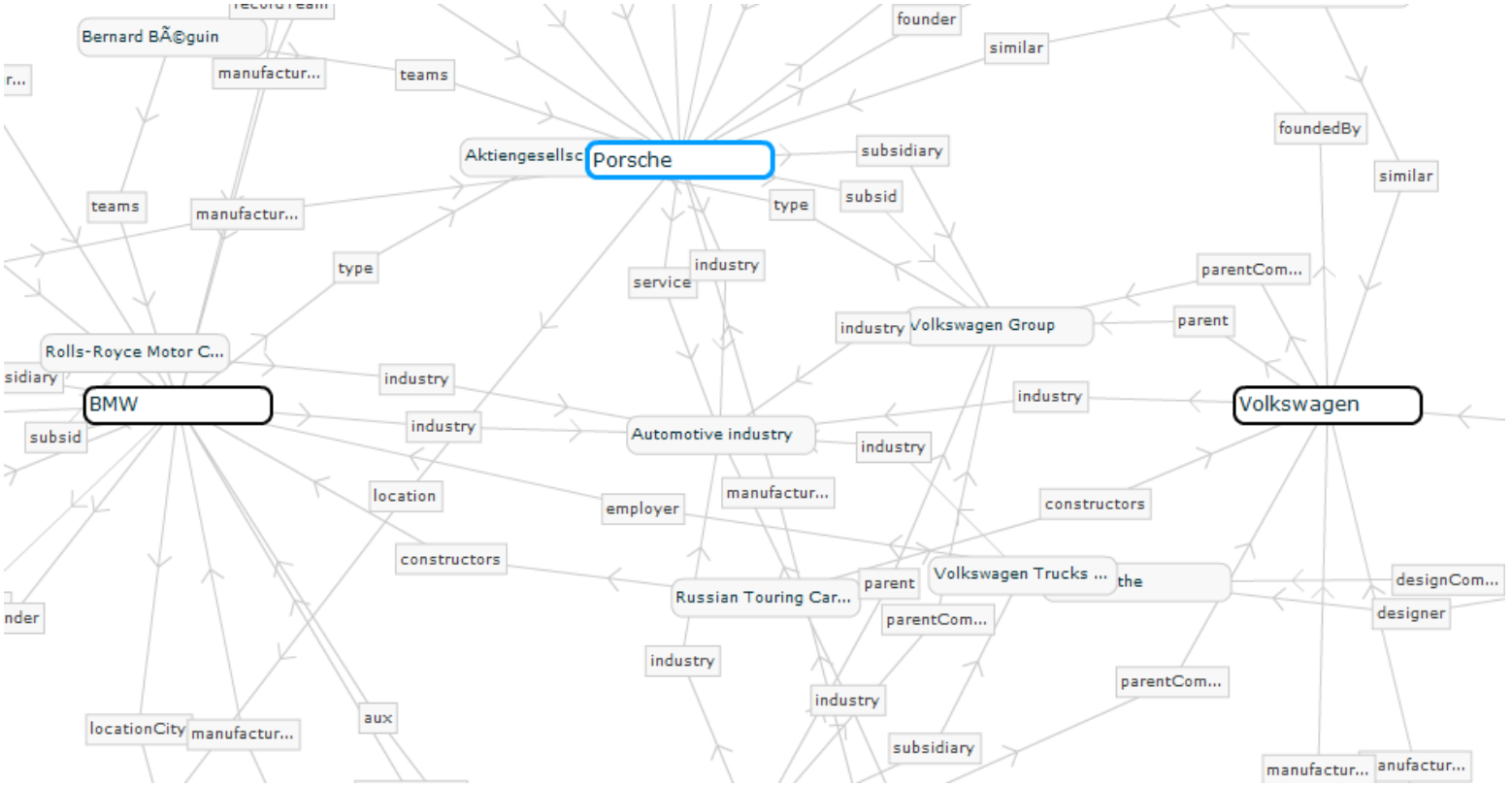

HTML Browsers
RDF Browsers
SPARQL Clients
The Web
HTML
Linked Data
SPARQL
Config File
Pubby
Triple Store

Linked Data browsers – Marbles

- http://marbles.sourceforge.net
- XHTML views of RDF data (SPARQL endpoint), caching, predicate traversal

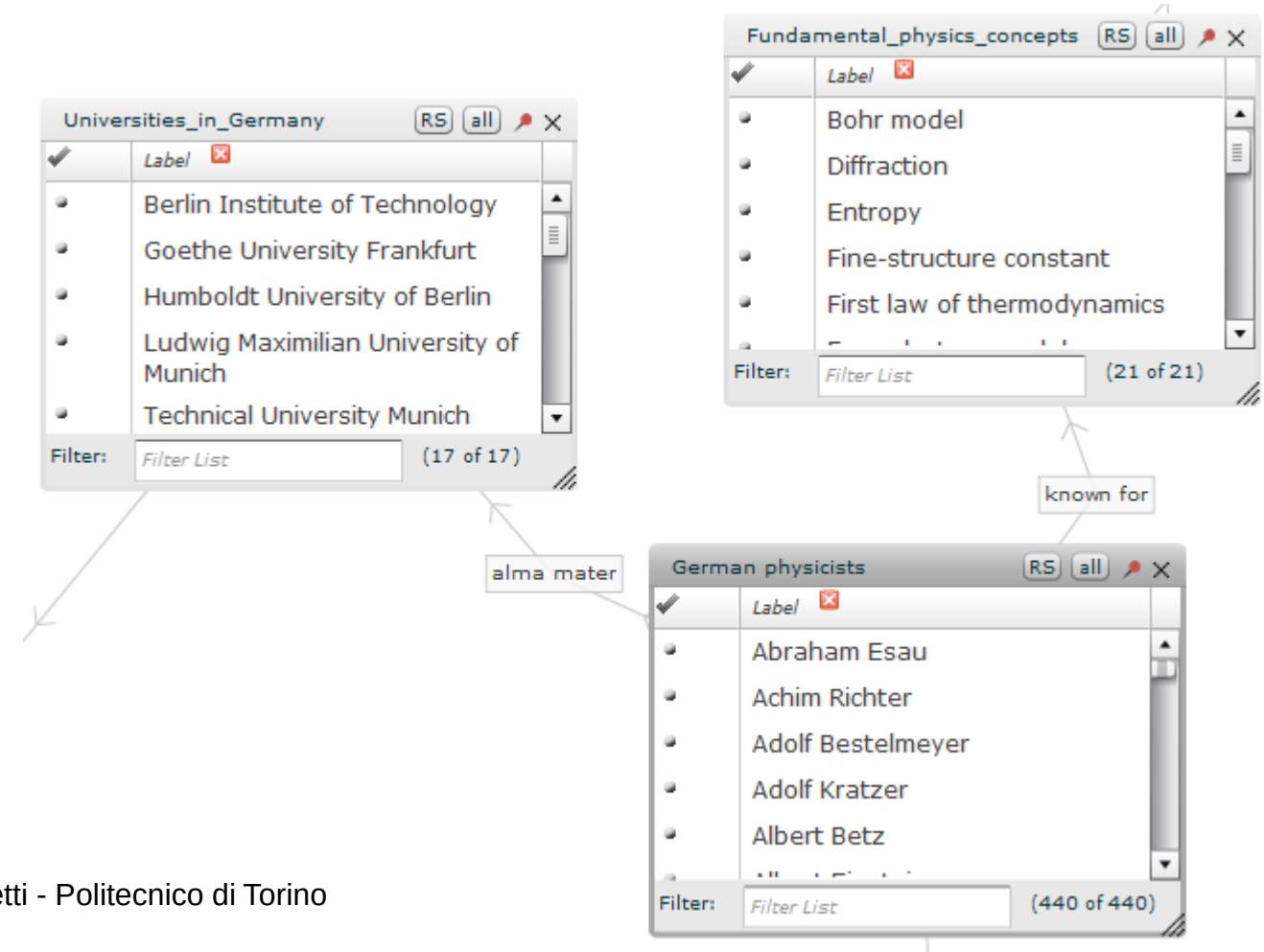
Linked Data browsers – RelFinder

- http://relfinder.dbpedia.org
- Explore & navigate relationships in a RDF graph



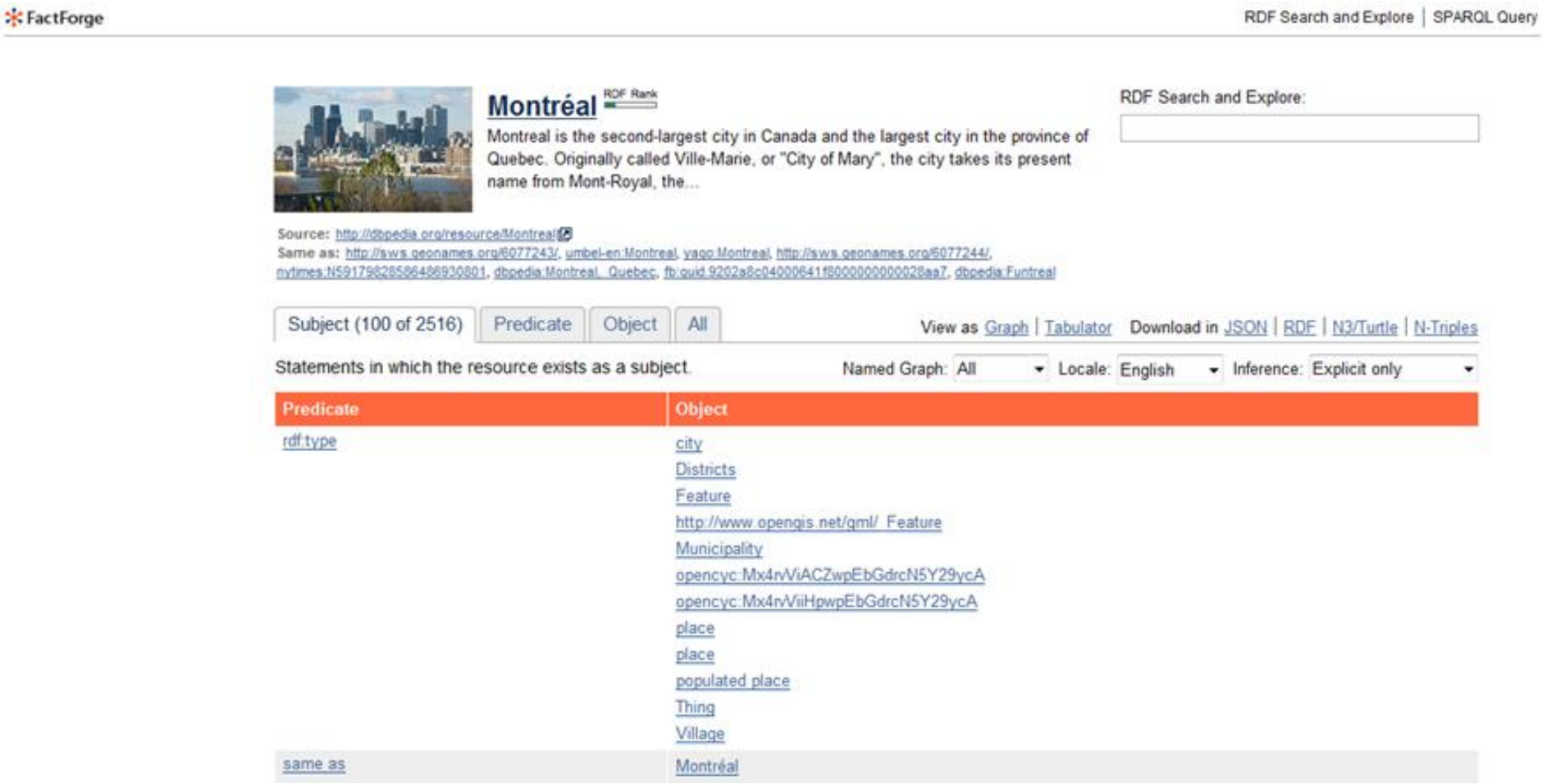
Linked Data browsers – gFacet

- http://semanticweb.org/wiki/GFacet
- Graph based visualisation & faceted filtering of RDF data



Linked Data browsers – Forest

- Front-end to FactForge and LinkedLifeData

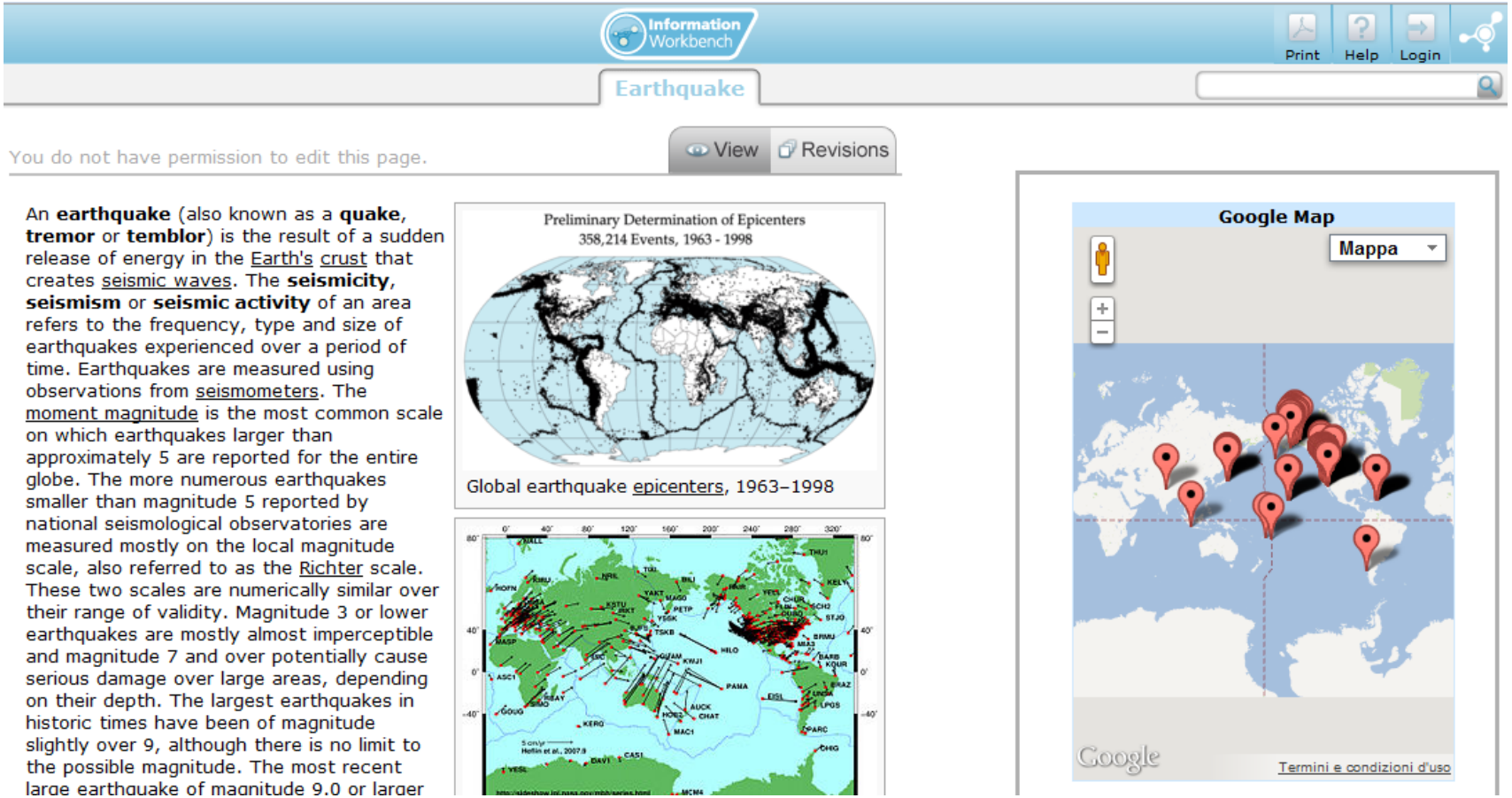


FactForge and LinkedLifeData

- FactForge
  - Integrates some of the most central LOD datasets
  - General-purpose information (not specific to a domain)
  - 1.2B explicit plus 1B inferred statements
  - The largest upper-level knowledge base
  - http://www.FactForge.net/
- LinkedLifeData
  - 25 of the most popular life-science datasets
  - 2.7B explicit and 1.4B inferred triples
  - http://www.LinkedLifeData.com

Linked Data browsers – Information Workbench

- http://iwb.fluidops.com/resource/Help:Start



dati.gov.it
I dati aperti della PA

Governo italiano
Presidenza del Consiglio dei Ministri
Ministro per la pubblica amministrazione e la semplificazione

Home | Cerco i dati | Voglio capire di più | Condivido un dataset | Le App della PA | Notizie

forma dei dati aperti del CNR: it

siglio Nazionale delle Ricerche) con la data.cnr.it, rappresenta una delle più esperienze italiane di apertura delle i in formato Linked data.

I dati ape

Il Governo la dedicata alla stati rilascia dataset – co

5 di 6

più recenti **più lette**

Il progetto open data di INAIL
Apps4Italy: le migliori proposte
I dati aperti del Ministero della Salute
La piattaforma dei dati aperti del CNR: data.cnr.it
Le officine dello stand Open Data a Forum PA

altre notizie...

Cerco i dati

Il catalogo degli open data contiene 531 dataset di 35 Amministrazioni

Click

Voglio capire di più

Come e perché fare open data:

- Definizione
- Vademecum
- Licenza italiana
- Discussione online
- Open data in Italia
- Altri riferimenti utili

Condivido un dataset

Segnalo un insieme di dati della pubblica amministrazione pubblicato in formato aperto

Click

Le applicazioni già disponibili per accedere ai servizi della PA da uno smartphone, suddivise per

- Amministrazioni centrali
- Regioni
- Province
- Comuni

Linked Open Data Italia
www.linkedopendata.it

Liberare i dati a beneficio di tutti

Mi piace 286

Cerca nel sito

HOME | HOME (ENG) | DIVENTA SOCIO | PROGETTI | LINKED DATA | CHI SIAMO | NEWS | PRESS | SLIDES

APPS4ITALY

I vincitori saranno premiati Sabato 19 maggio a Roma, all'interno di Forum PA 2012.

PUBBLICA I TUOI DATI!

Hai un dataset che vuoi rendere disponibile? Vuoi pubblicarlo su linkedopendata.it?

Accettiamo dataset in qualunque settore (governo, scienza, cultura,

Home

Switch to English

Linked Open Data: dati aperti collegati e usabili

Linked Open Data Italia pubblica dati aperti e facilmente accessibili da persone e applicazioni. I data set a disposizione, con licenze aperte e pubblicati in modalità LinkedData, possono essere direttamente interrogati da qualsiasi applicazione indipendentemente da linguaggi di programmazione e tecnologie.

Stiamo lavorando per pubblicare sempre più dati in diversi settori. Pubblica amministrazione, istruzione, infrastrutture e ricerca sono solo alcuni delle potenziali aree in cui l'accesso libero ai dati può portare giovamenti e aprire nuove opportunità. Seguici su twitter e facebook per tenerti informato sui nuovi dataset pubblicati.

Come collaborare?

Usa i dati. Sperimenta e fornisci feedback facendo query sui dati o provando ad utilizzarli in mash-up e applicazioni.
Suggerisci un dataset. Conosci dei dati pubblici poco accessibili a cui vorresti avere

NEWS

- Le implicazioni economiche degli Open Data in Italia: Intervista a Raimondo Iemma, research fellow del NEXA Center for Internet and Society del Politecnic...
- Libro Bianco per il riutilizzo dell'informazione del settore pubblico: E' disponibile per il download in versione beta il Libro Bianco per il riutilizzo dell'informazione ...
- Welcome dati.camera.it: Siamo davvero contenti che, dal 20 dicembre, ci sia dati.camera.it, il portale con cui il parlamento...

SCARICA IL LIBRO

Francesca Di Donato, Lo

http://it.ckan.net/

Benvenuti su CKAN Italia!

Cerca dati

Cerca dataset

CKAN Italia contiene 238 dataset che puoi esplorare, conoscere e scaricare.

Condividi dati

Aggiungi i tuoi dataset, condividili con gli altri e trova altre persone interessate ai tuoi dati.

Crea un dataset »

Collabora

Scopri di più su come lavorare con gli open data con queste risorse:

- GetTheData.org
- DataPatterns.org
- Open Data Manual

Chi altro c'è qui?

EVPSI

Dataset provenienti dal progetto EVPSI (Extracting Value from Public Sector Information), coordinato dal Dipartimento di Scienze Giuridiche di Torino, con il supporto della Regione...

EVPSI ha 52 dataset

Spaghetti Open Data

Dataset segnalati nel gruppo di discussione Spaghetti Open Data.

Spaghetti Open Data ha 6 dataset

Provincia Autonoma di Trento

Si tratta di tutti i siti del dominio .provincia.tn.it che contengono dati che possono essere scaricati

Provincia Autonoma di Trento ha 4 dataset

LinkedOpenData.it

Dataset creati o gestiti dall'Associazione Linked Open Data Italia.

LinkedOpenData.it ha 4 dataset

Torino Open Data

Dataset rilasciati dalla Città di Torino in occasione di Biennale Democrazia 2011.

Torino Open Data ha 2 dataset

dati.piemonte.it

Un gruppo che raccoglie tutti i pacchetti dati presenti sul portale del riuso della Regione Piemonte.

dati.piemonte.it ha 1 dataset

SPARQL syntax

SPARQL query structure

- A SPARQL query includes, in order
  - Prefix declarations, for abbreviating URIs
  - A result clause, identifying what information to return from the query
  - The query pattern, specifying what to query for in the underlying dataset
  - Query modifiers: slicing, ordering, and otherwise rearranging query results

SPARQL query structure

- A SPARQL query includes, in order

```
# prefix declarations
PREFIX foo: <http://example.com/resources/>
...
# result clause
SELECT ...
# query pattern
WHERE {
    ...
}
# query modifiers
ORDER BY ...
```

Dataset: Friend of a Friend (FOAF)

- FOAF is a standard RDF vocabulary for describing people and relationships
- Tim Berners-Lee's FOAF information available at http://www.w3.org/People/Berners-Lee/card

```
@prefix card: <http://www.w3.org/People/Berners-Lee/card#> .
@prefix foaf:  <http://xmlns.com/foaf/0.1/> .
card:i foaf:name "Timothy Berners-Lee" .
<http://bblfish.net/people/henry/card#me>
foaf:name "Henry Story" .
<http://www.cambridgesemantics.com/people/about/lee>
foaf:name "Lee Feigenbaum" .
card:amy foaf:name "Amy van der Hiel" .
...
```

Example 1 – simple triple pattern

- In the graph http://www.w3.org/People/Berners-Lee/card, find all subjects (?person) and objects (?name) linked with the foaf:name predicate.
- Then return all the values of ?name.
- In other words, find all names mentioned in Tim Berners-Lee's FOAF file

```
PREFIX foaf:
<http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
    ?person foaf:name ?name .
}
```

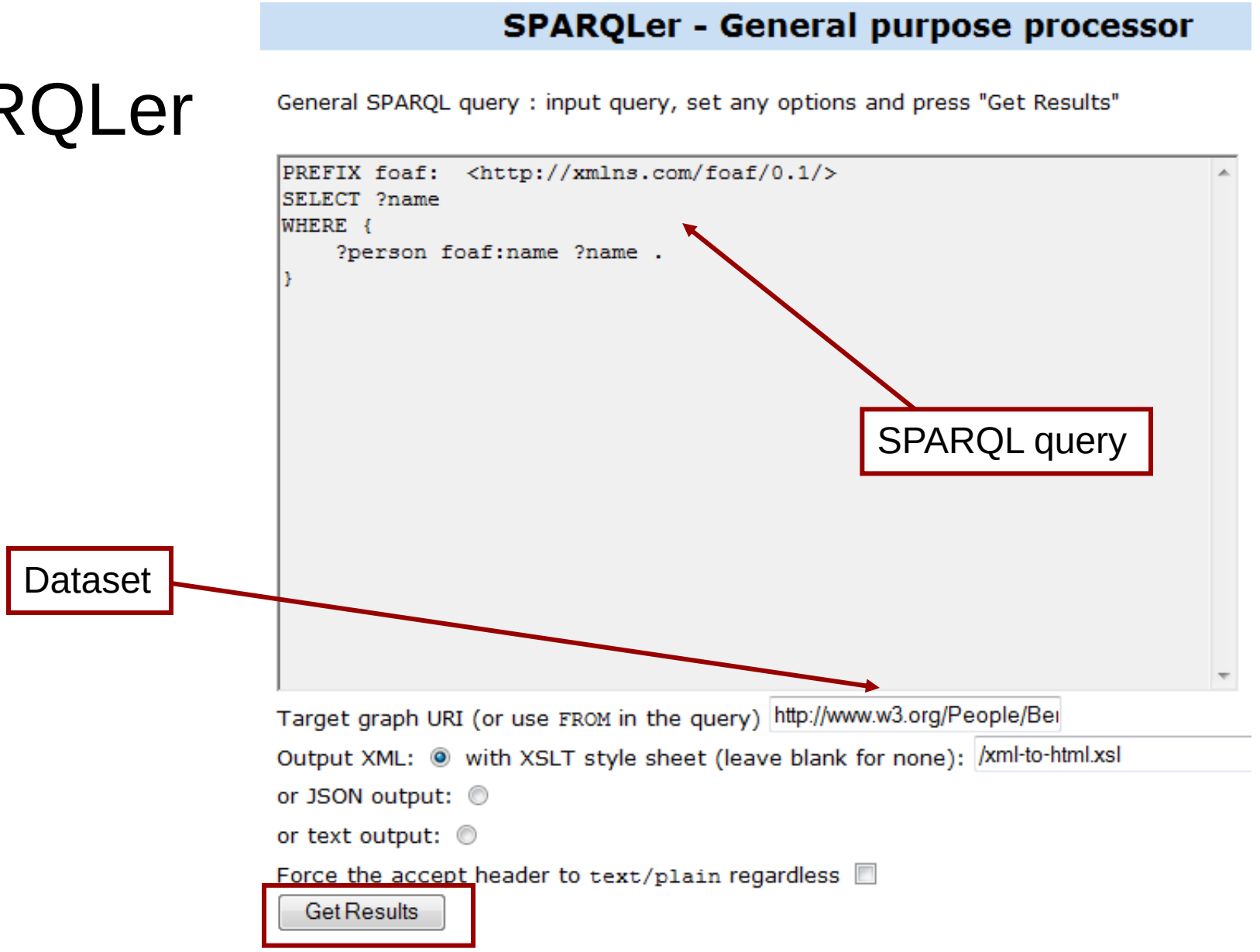
SPARQL endpoints

- Accept queries and returns results via HTTP
  - Generic endpoints queries any Web-accessible RDF data
  - Specific endpoints are hardwired to query against particular datasets
- The results of SPARQL queries can be returned in a variety of formats:
  - XML, JSON, RDF, HTML
  - JSON (JavaScript Object Notation): lightweight computer data interchange format; text-based, human-readable format for representing simple data structures and associative arrays

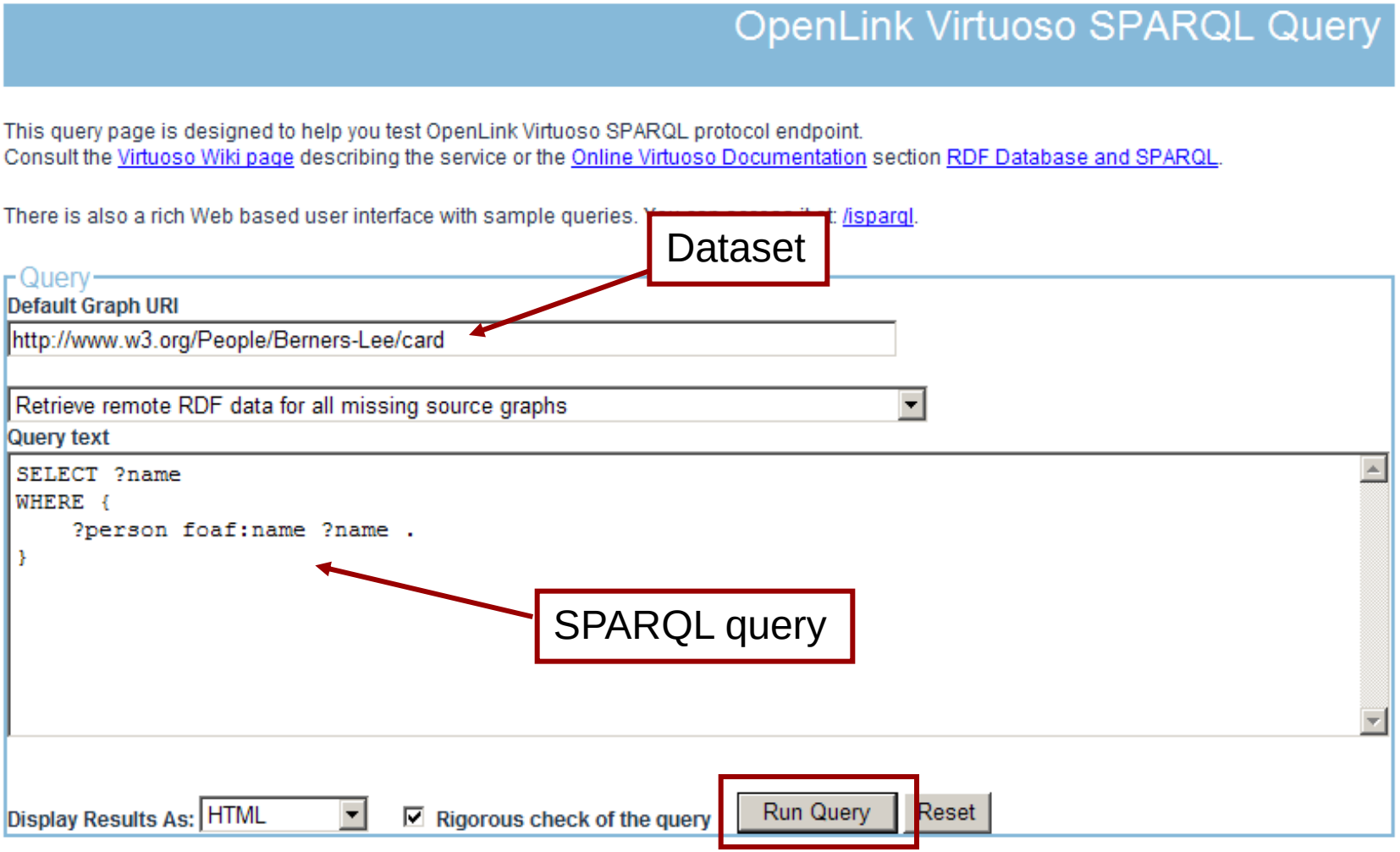
SPARQL endpoints

- This query is for an arbitrary bit of RDF data (Tim Berners-Lee's FOAF file)
- => generic endpoint to run it
- Possible choices
  - SPARQLer - General purpose processor - sparql.org
    - http://sparql.org/sparql.html
  - OpenLink's Virtuoso (Make sure to choose "Retrieve remote RDF data for all missing source graphs")
    - http://bbc.openlinksw.com/sparql/
  - Redland’s Rasqal
    - http://librdf.org/rasqal/

SPARQLer


SPARQLer - General purpose processor
General SPARQL query : input query, set any options and press "Get Results"
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
?person foaf:name ?name .
}
SPARQL query
Dataset
Target graph URI (or use FROM in the query)
http://www.w3.org/People/Be
Output XML: with XSLT style sheet (leave blank for none):
/xml-to-html.xsl
or JSON output:
or text output:
Force the accept header to text/plain regardless
Get Results

OpenLink's Virtuoso



Example 1 - simple triple pattern

```
PREFIX foaf:
<http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
    ?person foaf:name ?name .
}
```

name
Dan Connolly
Henry Story
Timothy Berners-Lee
Norman Walsh
World Wide Web Consortium
Ralph R. Swick
Daniel Krech
Christoph Bussler
Nicholas Gibbins
Wendy Hall
Nigel Shadbolt
Les Carr
Charles McCathieNevile
Håkon Wium Lie
Peter Szolovits

Example 2 – multiple triple pattern

- Find all people in Tim Berners-Lee's FOAF file that have names and email addresses
- Return each person's URI, name, and email address

- Multiple triple patterns retrieve multiple properties about a particular resource
- SELECT * selects all variables mentioned in the query

```
PREFIX foaf:
<http://xmlns.com/foaf/0.1/>
SELECT *
WHERE {
    ?person foaf:name ?name .
    ?person foaf:mbox ?email .
}
```

Example 2 - multiple triple pattern

person	name	email
http://www.w3.org/People/Connolly/#me	Dan Connolly	mailto:connolly@w3.org
http://www.w3.org/People/Berners-Lee/card#i	Timothy Berners-Lee	mailto:timbl@w3.org
http://www.aaronsw.com/about.xrdf#aaronsw	Aaron Swartz	mailto:me@aaronsw.com
http://www.dajobe.org/foaf.rdf#i	Dave Beckett	mailto:dave@dajobe.org
http://www.w3.org/People/Berners-Lee/card#amy	Amy van der Hiel	mailto:amy@w3.org
http://www.w3.org/People/EM/contact#me	Eric Miller	mailto:em@w3.org
http://www.w3.org/People/karl/karl-foaf.xrdf#me	Karl Dubost	mailto:karl@w3.org
http://www.w3.org/People/Berners-Lee/card#cm	Coralie Mercier	mailto:coralie@w3.org
http://www.w3.org/People/Berners-Lee/card#dj	Dean Jackson	mailto:dean@w3.org
http://www.w3.org/People/Berners-Lee/card#dj	Dean Jackson	mailto:dino@grorg.org
http://www.w3.org/People/Berners-Lee/card#edd	Edd Dumbill	mailto:edd@usefulinc.com
http://www.w3.org/People/Berners-Lee/card#edd	Edd Dumbill	mailto:edd@xml.com
http://www.w3.org/People/Berners-Lee/card#edd	Edd Dumbill	mailto:edd@xmlhack.com
http://swordfish.rdfweb.org/people/libby/rdfweb/webwho.xrdf#me	Libby Miller	mailto:libby.miller@bristol.ac.uk
http://people.csail.mit.edu/lkagal/foaf.rdf#me	Lalana Kagal	mailto:lalana@csail.mit.edu
nodeID://1003907694	Susie Stephens	mailto:susie.stephens@gmail.com

Example 3 – traversing a graph

- Find the homepage of anyone known by Tim Berners-Lee

homepage
http://purl.org/net/eric/
http://www.johnseelybrown.com/
http://www.grorg.org/dean/
http://heddley.com/edd/
http://www.mellon.org/about_foundation/staff/program-area-staff/irafuchs

Example 3 – traversing a graph

```
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
PREFIX card: <http://www.w3.org/People/Berners-Lee/card#>
SELECT ?homepage
FROM <http://www.w3.org/People/Berners-Lee/card>
WHERE {
    card:i foaf:knows ?known .
    ?known foaf:homepage ?homepage .
}
```

- The FROM keyword specifies the target graph in the query
- By using ?known as an object of one triple and the subject of another, it is possible to traverse multiple links in the graph

Dataset: DBPedia

- DBPedia is an RDF version of information from Wikipedia
- Contains data derived from Wikipedia's infoboxes, category hierarchy, article abstracts, and various external links
- Contains over 100 million triples
- Dataset: http://dbpedia.org/sparql/

Example 4 – exploring DBPedia

- Find 15 example concepts in the DBPedia dataset

```
SELECT DISTINCT ?concept
WHERE {
    ?s a ?concept .
} LIMIT 15
```

concept
http://www.w3.org/2004/02/skos/core#Concept
http://dbpedia.org/ontology/MusicalWork
http://dbpedia.org/ontology/Resource
http://dbpedia.org/ontology/Work
http://dbpedia.org/ontology/Album
http://dbpedia.org/ontology/Musical
http://umbel.org/umbel/sc/Product
http://umbel.org/umbel/ac/Artifact
http://dbpedia.org/class/yago/1982Novels
http://dbpedia.org/ontology/Book
http://dbpedia.org/class/yago/EnglishAstronomers
http://dbpedia.org/class/yago/EnglishPoliticians
http://umbel.org/umbel/sc/Astronomer
http://dbpedia.org/class/yago/LivingPeople
http://dbpedia.org/class/yago/AmericanCompetitiveEaters

Example 4 – exploring DBPedia

- LIMIT is a solution modifier that limits the number of rows returned from a query
- SPARQL has two other solution modifiers
  - ORDER BY for sorting query solutions on the value of one or more variables
  - OFFSET, used in conjunction with LIMIT and ORDER BY to take a slice of a sorted solution set (e.g. for paging)
- The SPARQL keyword a is a shortcut for the common predicate rdf:type (class of a resource)
- The DISTINCT modifier eliminates duplicate rows from the query results

Example 5 – basic SPARQL filters

- Find all landlocked countries with a population greater than 15 million

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
    ?country a type:LandlockedCountries ;
             rdfs:label ?country_name ;
             prop:populationEstimate ?population .
    FILTER (?population > 15000000) .
}
```

- FILTER constraints use boolean conditions to filter out unwanted query results
- A semicolon (;) can be used to separate two triple patterns that share the same subject

SPARQL filters

- Conditions on literal values
- Syntax

```
FILTER expression
```

- Examples

```
FILTER (?age > 30)
FILTER isIRI(?x)
FILTER !BOUND(?y)
```

SPARQL filters

- `BOUND(var)`
  - true if var is bound in query answer
  - false, otherwise
  - `!BOUND(var)` enables negation-as-failure
- Testing types
  - `isIRI(A)`: A is an “Internationalized Resource Identifier”
  - `isBLANK(A)`: A is a blank node
  - `isLITERAL(A)`: A is a literal

SPARQL filters

- Comparison between RDF terms

```
A = B
A != B
```

- Comparison between Numeric and Date types

```
A = B
A != B
A <= B
A >= B
A < B
A > B
```

- Boolean AND/OR

```
A && B
A || B
```

- Basic arithmetic

```
A + B
A - B
A * B
A / B
```

Example 5 – basic SPARQL filters

- Note all the translated duplicates in the results
- How can we deal with that?

country_name	population
Afghanistan	31889923
Afghanistan	31889923
□□□□□□□	31889923
Afghanistan	31889923
Afganistan	31889923
Афганистан	31889923
Afghanistan	31889923
Afganistán	31889923
Afganistan	31889923
Afghanistan	31889923
Afeganistão	31889923
□□□	31889923
Afghanistan	31889923
Afghanistan	31889923
Afghanistan	31889923
Ethiopia	78254090
Éthiopie	78254090

Example 6 – SPARQL filters

- Find me all landlocked countries with a population greater than 15 million (revisited), with the highest population country first

```
PREFIX type: <http://dbpedia.org/class/yago/>
PREFIX prop: <http://dbpedia.org/property/>
SELECT ?country_name ?population
WHERE {
    ?country a type:LandlockedCountries ;
             rdfs:label ?country_name ;
             prop:populationEstimate ?population .
    FILTER (?population > 15000000 &&
            langMatches(lang(?country_name), "EN")) .
} ORDER BY DESC(?population)
```

Example 6 – SPARQL filters

- lang extracts a literal's language tag, if any
- langMatches matches a language tag against a language range

country_name	population
Ethiopia	78254090
Afghanistan	31889923
Uganda	30900000
Nepal	29519114
Uzbekistan	27372000
Kazakhstan	15217711

Dataset: Jamendo

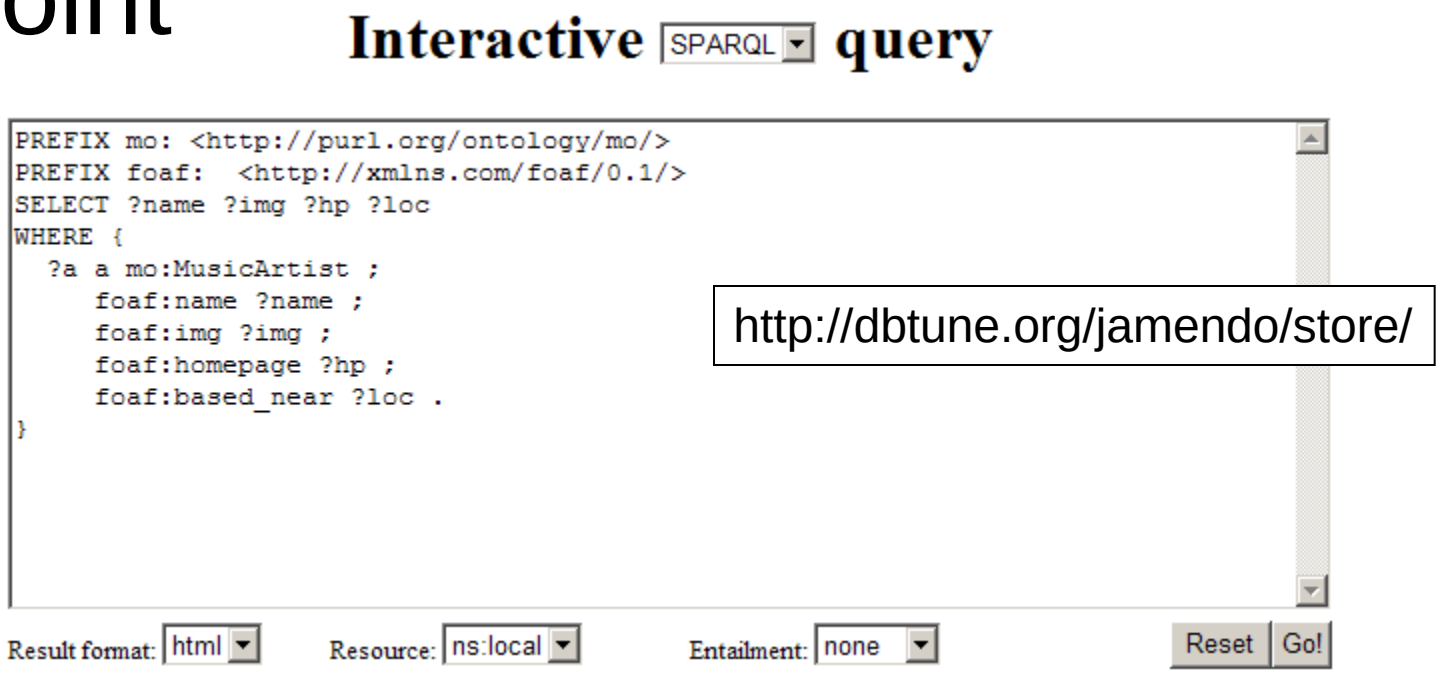
- Jamendo is a community collection of music all freely licensed under Creative Commons licenses
  - http://www.jamendo.com/it/
- DBTune.org hosts a queryable RDF version of information about Jamendo's music collection
  - Data on thousands of artists, tens of thousands of albums, and nearly 100,000 tracks
  - http://dbtune.org/jamendo/store/

Example 7 – the wrong way

- Find all Jamendo artists along with their image, home page, and the location they're near

```
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?name ?img ?hp ?loc
WHERE {
  ?a a mo:MusicArtist ;
     foaf:name ?name ;
     foaf:img ?img ;
     foaf:homepage ?hp ;
     foaf:based_near ?loc .
}
```

Example 7 – DBTune SPARQL endpoint



- Jamendo has information on about 3,500 artists
- Trying the query we only get 2,667 results. What's wrong?

Example 7 – the right way

- Not every artist has an image, homepage, or location!
- OPTIONAL tries to match a graph pattern, but doesn't fail the whole query if the optional match fails
- If an OPTIONAL pattern fails to match for a particular solution, any variables in that pattern remain unbound (no value) for that solution

```
PREFIX mo: <http://purl.org/ontology/mo/>
PREFIX foaf:  <http://xmlns.com/foaf/0.1/>
SELECT ?name ?img ?hp ?loc
WHERE {
  ?a a mo:MusicArtist ;
     foaf:name ?name .
  OPTIONAL { ?a foaf:img ?img }
  OPTIONAL { ?a foaf:homepage ?hp }
  OPTIONAL { ?a foaf:based_near ?loc }
}
```

Dataset: GovTrack

- GovTrack provides SPARQL access to data on the U.S. Congress
- Contains over 13,000,000 triples about legislators, bills, and votes
- http://www.govtrack.us/

Example 8 – querying alternatives

- Find Senate bills that either John McCain or Barack Obama sponsored and the other cosponsored

```
PREFIX bill: <http://www.rdfabout.com/rdf/schema/usbill/>
PREFIX dc:  <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?title ?sponsor ?status
WHERE {
  { ?bill bill:sponsor ?mccain ; bill:cosponsor ?obama . }
      UNION
  { ?bill bill:sponsor ?obama ; bill:cosponsor ?mccain . }
    ?bill a bill:SenateBill ;
          bill:status ?status ;
          bill:sponsor ?sponsor ;
           dc:title ?title .
    ?obama  foaf:name "Barack Obama" .
    ?mccain foaf:name "John McCain" .
}
```

Example 8 – GovTrack specific endpoint

```
PREFIX bill: <http://www.rdfabout.com/rdf/schema/usbill/>
PREFIX dc:  <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?title ?sponsor ?status
WHERE {
  { ?bill bill:sponsor ?mccain ; bill:cosponsor ?obama . }
      UNION
  { ?bill bill:sponsor ?obama ; bill:cosponsor ?mccain . }
```

Run Query Display As: SPARQL XML

http://www.govtrack.us/developers/rdf.xpd

- The UNION keyword forms a disjunction of two graph patterns: solutions to both sides of the UNION are included in the results

RDF datasets

- All queries so far have been against a single graph
- In SPARQL this is known as the default graph
- RDF datasets are composed of a single default graph and zero or more named graphs, identified by a URI
- Named graphs can be specified with one or more FROM NAMED clauses, or they can be hardwired into a particular SPARQL endpoint
- The SPARQL GRAPH keyword allows portions of a query to match against the named graphs in the RDF dataset
- Anything outside a GRAPH clause matches against the default graph

Dataset: semanticweb.org

- data.semanticweb.org hosts RDF data regarding workshops, schedules, and presenters for the International Semantic Web (ISWC) and European Semantic Web Conference (ESWC) series of events
- Presents data via FOAF, SWRC, and iCal ontologies
- The data for each individual ISWC or ESWC event is stored in its own named graph
  - i.e., there is one named graph per conference event contained in this dataset
- http://data.semanticweb.org/

Example 9 – querying named graphs

- Find people who have been involved with at least three ISWC or ESWC conference events

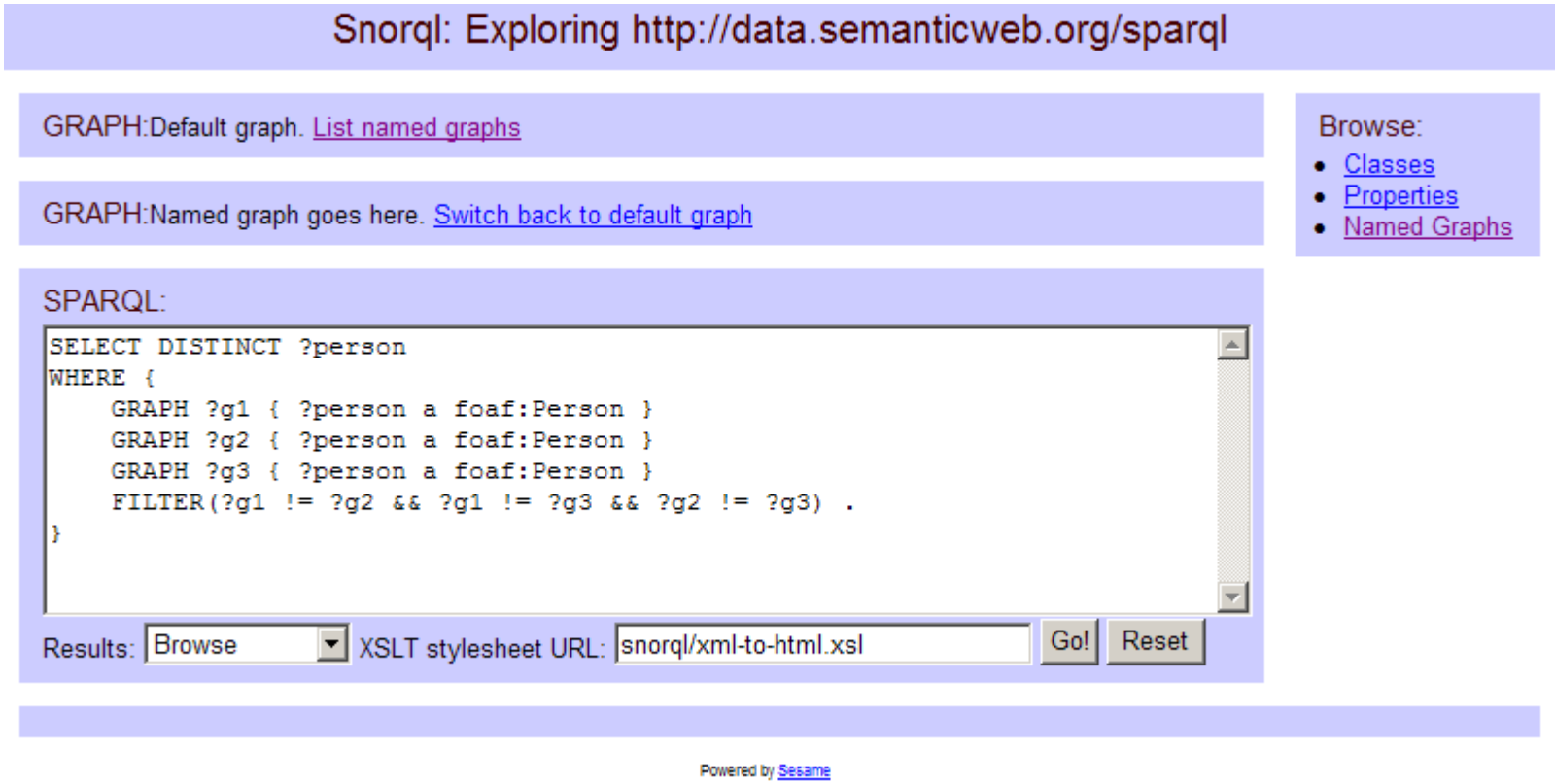
```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT ?person
WHERE {
    GRAPH ?g1 { ?person a foaf:Person }
    GRAPH ?g2 { ?person a foaf:Person }
    GRAPH ?g3 { ?person a foaf:Person }
    FILTER(?g1 != ?g2 && ?g1 != ?g3 && ?g2 != ?g3) .
}
```

Example 9 – querying named graphs

- The GRAPH ?g construct allows a pattern to match against one of the named graphs in the RDF dataset
- The URI of the matching graph is bound to ?g (or whatever variable was actually used)
- The FILTER assures that we're finding a person who occurs in three distinct graphs
- The Web interface used for this SPARQL query defines the foaf: prefix, which is why it is omitted here

Data.semanticweb.org specific SPARQL endpoint

http://data.semanticweb.org/snorql/



SPARQL 1.1 extensions

- Projected expressions
  - Adds the ability for query results to contain values derived from constants, function calls, or other expressions in the SELECT list
- Aggregates
  - Adds the ability to group results and calculate aggregate values (e.g. count, min, max, avg, sum, …)
- Sub-queries
  - Allows one query to the nested within another

SPARQL 1.1 extensions

- Negation
  - Includes improved language syntax for querying negations
- Property paths
  - Adds the ability to query arbitrarily lenght path through a graph via a regular-expression-like syntax known as property paths
- Basic federated query
  - Defines a mechanism for splitting a single query among multipleSPAQL endpoints and combining together the results from each

SPARQL exercise

Exercises - RDF

```
@prefix : <http://example.org/data#> .
@prefix ont: <http://example.org/myOntology#> .
@prefix vcard: <http://www.w3.org/2001/vcard-rdf/3.0#> .

:john
  vcard:FN "John Smith" ;
  vcard:N [
    vcard:Given "John" ;
    vcard:Family "Smith" ] ;
  ont:hasAge 32 ;
  ont:marriedTo :mary .
:mary
  vcard:FN "Mary Smith" ;
  vcard:N [
    vcard:Given "Mary" ;
    vcard:Family "Smith" ] ;
  ont:hasAge 29 .
```

SPARQL query – exercise 1

- Return the full names of all people in the graph

```
PREFIX vCard:
<http://www.w3.org/2001/vcardrdf/3.0#>
SELECT ?fullName
WHERE {?x vCard:FN ?fullName}
```

- Result

```
fullName
===============
“John Smith”
“Mary Smith”
```

SPARQL query – exercise 2

- Return the relation between John and Mary

```
PREFIX : <http://example.org/data#>
SELECT ?p
WHERE { :john ?p :mary }
```

- Result

```
p
=================
<http://example.org/myOntology#marriedTo>
```

SPARQL query – exercise 3

- Return the spouse of a person whose name is John Smith

```
PREFIX vCard:
<http://www.w3.org/2001/vcard-rdf/3.0#>
PREFIX ont: <http://example.org/myOntology#>
SELECT ?y
WHERE {?x vCard:FN "John Smith".
       ?x ont:marriedTo ?y}
```

- Result

```
y
=================
<http://example.org/data#mary>
```

SPARQL query – exercise 4

- Return the name and the first name of all people in the knowledge base

```
PREFIX vCard:
<http://www.w3.org/2001/vcard-rdf/3.0#>
SELECT ?name, ?firstName
WHERE {?x vCard:N ?name .
       ?name vCard:Given ?firstName}
```

- Result

```
    name     firstName
=========================
“John Smith”   "John"
“Mary Smith”   "Mary"
```

SPARQL query – exercise 5

- Return all people over 30 in the knowledge base

```
PREFIX ont: <http://example.org/myOntology#>
SELECT ?x
WHERE {?x ont:hasAge ?age .
       FILTER(?age > 30)}
```

- Result

```
x
==================
<http://example.org/data#john>
```

FROM

- Select RDF graph (= dataset) to be queried
- In case of multiple FROM clauses, graphs are merged
- Example

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
FROM <http://example.org/foaf/aliceFoaf>
WHERE { ?x foaf:name ?name }
```

SPARQL query – exercise 6

- Graph http://example.org/bob

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Bob" .
_:a foaf:mbox <mailto:bob@oldcorp.example.org> .
```

- Graph http://example.org/alice

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Alice" .
_:a foaf:mbox <mailto:alice@work.example> .
```

SPARQL query – exercise 6

- Return the names of people in both graphs

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?src ?name
FROM NAMED <http://example.org/alice>
FROM NAMED <http://example.org/bob>
WHERE
{ GRAPH ?src { ?x foaf:name ?name } }
```

- Result

```
              src                      name
=======================================
<http://example.org/bob>      "Bob"
<http://example.org/alice>    "Alice"
```

References

- W3C, “Introduction to the Semantic Web”
  - http://www.w3.org/2006/Talks/0524-Edinburgh-IH/
- Lee Feigenbaum, “SPARQL By Example”
  - http://www.cambridgesemantics.com/2008/09/sparql-by-example
- Valentina Tamma, “Chapter 4: SPARQL”
  - http://www.csc.liv.ac.uk/~valli/Comp318/PDF/SPARQL.pdf
- Tom Heath, “An Introduction to Linked Data”
  - http://tomheath.com/slides/2009-02-austin-linkeddata-tutorial.pdf
- HTML Microdata
  - http://www.w3.org/TR/microdata
- Schema.org
  - http://schema.org

License