

nemo

ontology & conceptual
modeling research group



Linked Data

Vítor E. Silva Souza

[vitorsouza@inf.ufes.br]

<http://www.inf.ufes.br/~vitorsouza>

Department of Informatics

Federal University of Espírito Santo (Ufes),

Vitória, ES – Brazil

License for use and distribution

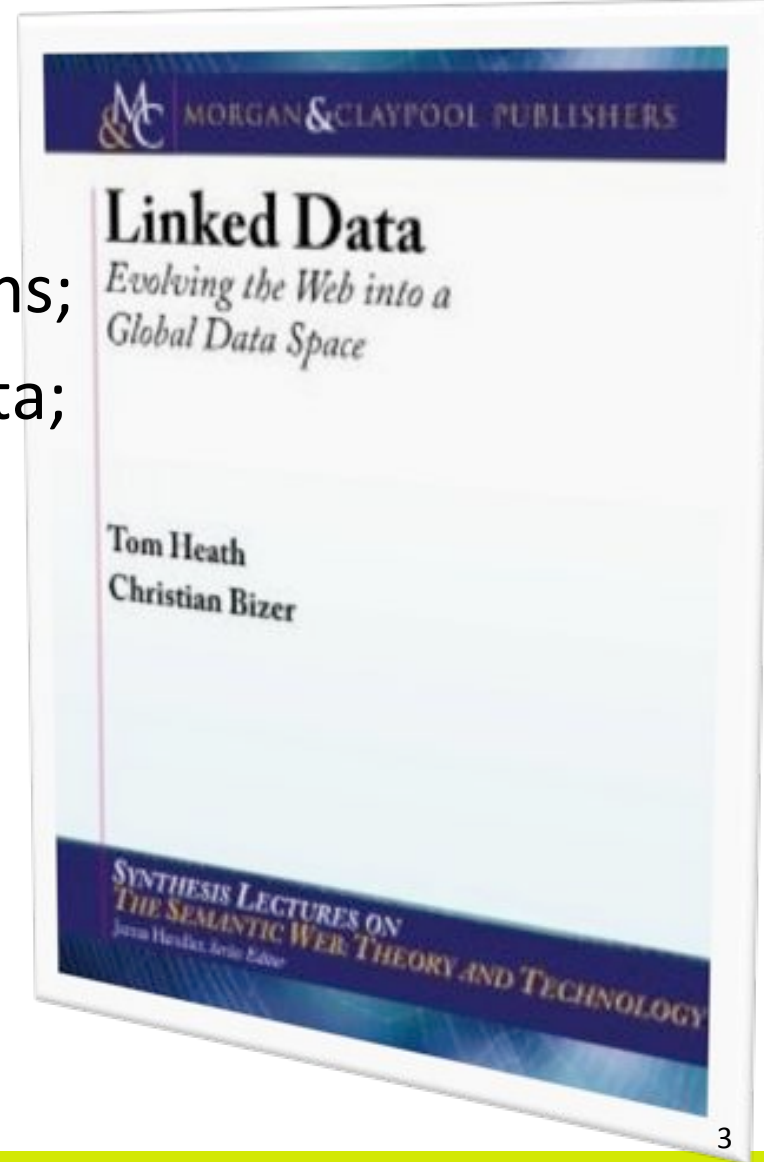
- This material is licensed under the Creative Commons license Attribution-ShareAlike 4.0 International;
- You are free to (for any purpose, even commercially):
 - Share: copy and redistribute the material in any medium or format;
 - Adapt: remix, transform, and build upon the material;
- Under the following terms:
 - Attribution: you must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use;
 - ShareAlike: if you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.



More information can be found at:
<http://creativecommons.org/licenses/by-sa/4.0/>



- Introduction;
- Principles of linked data;
- The Web of Data;
- Linked data design considerations;
- Recipes for publishing linked data;
- Consuming linked data.



<http://linkeddatabook.com>

- New data getting published every day;
- Consuming this data can be beneficial:
 - Amazon: product data available to third parties, creating an eco-system of affiliates;
 - Google / Yahoo!: consume data from various websites and provide better search results;
 - Human Genome Project: cooperation by exchanging research data between scientists;
 - theyworkforyou.com: UK voters can readily assess the performance of elected representatives.

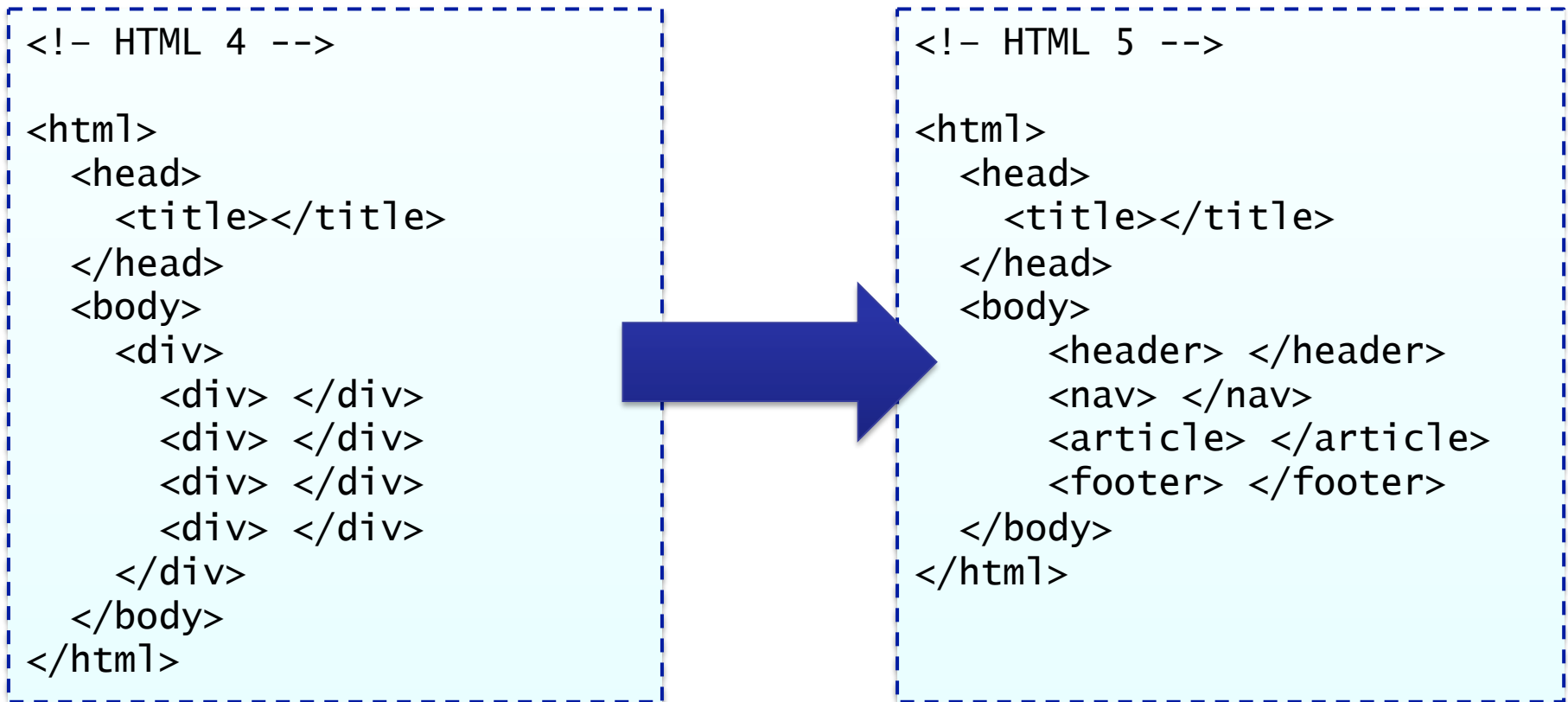
- How best to provide access to data so it can be most easily reused?
- How to enable the discovery of relevant data within the multitude of available data sets?
- How to enable applications to integrate data from large numbers of formerly unknown data sources?



Linked Data

“A set of principles and technologies that harness the ethos and infrastructure of the Web to enable data sharing and reuse on a massive scale.”

- Structure:
 - HTML structures text, not data;
 - But even HTML is beginning to be more structured...



- Earlier proposals for structuring data on the Web:
 - Microformats ([.org](http://microformats.org)): small sets of types and attributes, limited expression (relationships);
 - Web APIs (programmableweb.com): XML, JSON, REST services. No standards, integration efforts;
- XML, JSON, etc. lack something that HTML has since its conception: **hyperlinks!**

Humans click,
software crawl.

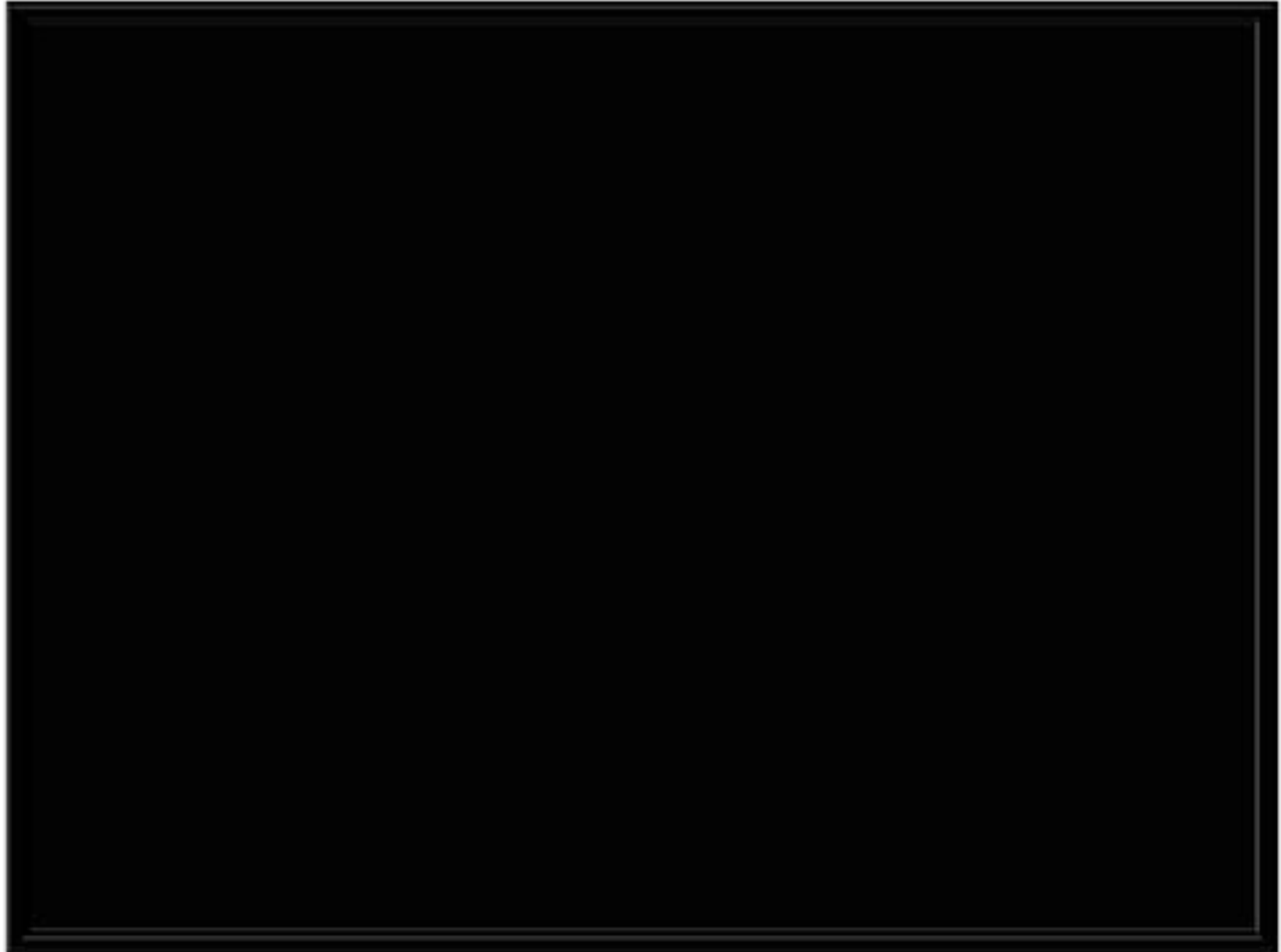


- Resource Description Framework (w3.org/RDF);
- Flexible way of describing things and how they relate to other things;
 - E.g.: a book (described in API 1) is for sale at a physical bookstore (from API 2), which is located at a city (described in API 3).
- Follow HTML principles, but:
 - **RDF links things, not documents:** connect the book, the bookstore and the city, not their web pages;
 - **RDF links are typed:** not *book link bookstore link city*, but *book forSaleIn bookstore locatedIn city*.

- As more people/organizations adopt this idea (data providers, app developers), we grow the Web of Data, or the Semantic Web;
- The concepts also apply to intranets (private webs) for, e.g., interoperability, data integration, etc. in enterprise environments.

“Just as hyperlinks in the classic Web connect documents into a single global information space, Linked Data enables links to the set between items in different data sources and therefore connect these sources into a single global data space. The use of Web standards and a common data model make it possible to implement generic applications that operate over the complete data space. This is the essence of Linked Data.”

Linked Open Data (LOD)



Source: <http://en.wikipedia.org/wiki/File:Linked-open-data-Europeana-video.ogv>

Linked Data

PRINCIPLES OF LINKED DATA

- Use URIs (Universal Resource Identifiers) as names for things;
- Use HTTP URIs, so that people can look up those names;
- When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL);
- Include links to other URIs, so that they can discover more things.

In other words, apply the general architecture of the WWW to the task of sharing structured data.



Tim Berners-Lee,
again.

- The document / syntactic Web:
 - URIs are global, unique IDs;
 - HTTP provides universal access;
 - HTML is a widely used content format;
 - Hyperlinks connect different documents.

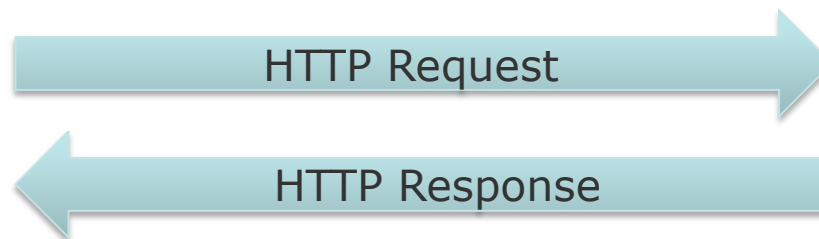
+ simple

+ decentralized

+ open

= architectural sweet spot

Syntactic	Semantic
Documents have URIs	Things (concepts) have URIs
HTTP as access mechanism	Uses same mechanism, allowing URIs to be looked up
HTML as standard format (important factor for web scale)	RDF as standard format
HTML links connect documents and are un-typed	RDF links connect anything and are typed (livesAt and worksAt between Person and Place)
Forms a global information space (interconnected documents)	Forms a global data space (interconnected concepts)





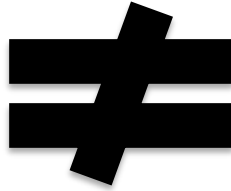
- URI = http:// <server> / <path>
- The URI mechanism is:
 - Simple: protocol + server address + file path;
 - Globally unique: server name is registered;
 - Decentralized: anyone with a server/website;
 - Provides the meaning for accessing the resource.

Open your Web browser and navigate to
<http://biglynx.co.uk/people/matt-briggs>

- First of all:



Concept
(Matt Briggs)

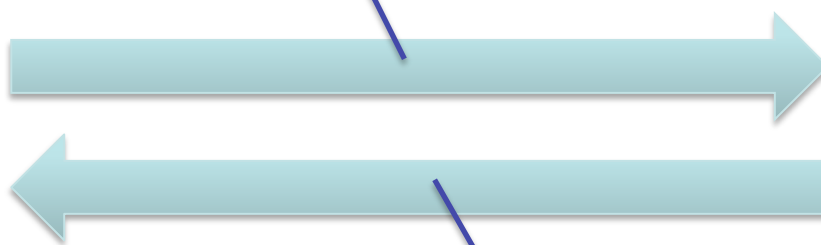


Web Document that describes
the concept (matt-briggs.rdf)

- HTTP allows content negotiation (client states preferred content type);
- Two main dereferencing strategies:
 - 303 URIs;
 - Hash URIs.

Dereferencing with 303 URIs (1)

```
GET /people/matt-briggs HTTP/1.1
Host: biglynx.co.uk
Accept: text/html;q=0.5, application/rdf+xml
```



```
1 HTTP/1.1 303 See Other
2 Location: http://biglynx.co.uk/people/matt-briggs.rdf
3 Vary: Accept
```



Matt Briggs



matt-briggs



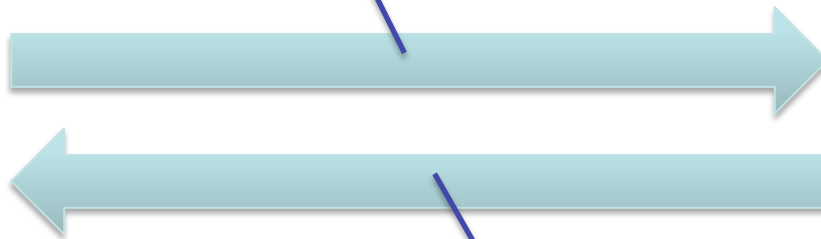
matt-briggs.rdf



matt-briggs.html

Dereferencing with 303 URIs (2)

```
GET /people/matt-briggs.rdf HTTP/1.1
Host: biglynx.co.uk
Accept: text/html;q=0.5, application/rdf+xml
```



```
HTTP/1.1 200 OK
Content-Type: application/rdf+xml

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF ...
```



Matt Briggs



matt-briggs



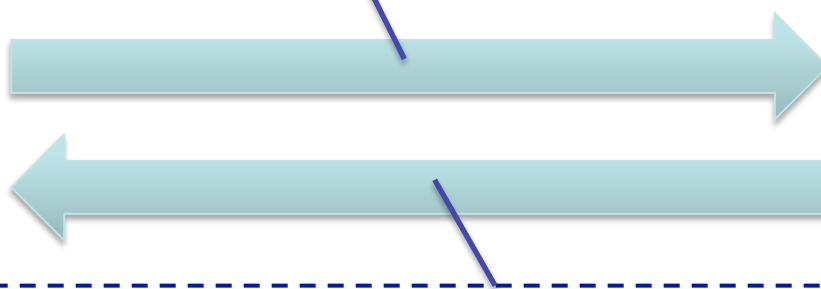
matt-briggs.rdf



matt-briggs.html

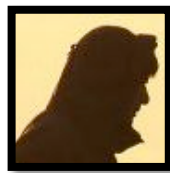
Dereferencing with hash URIs

```
GET /staff HTTP/1.1  
Host: biglynx.co.uk  
Accept: application/rdf+xml
```

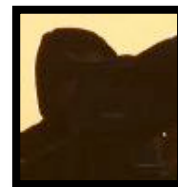


```
HTTP/1.1 200 OK  
Content-Type: application/rdf+xml  
  
<?xml version="1.0" encoding="UTF-8"?>  
<rdf:RDF ...
```

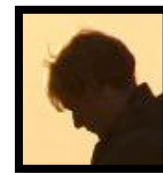
Client gets RDF
with entire
staff, looks for
#matt-briggs



Matt Briggs



Linda Meyer



Scott Miller



staff (.rdf)

- Hash does less HTTP round-trips:
 - If you're interested in the social connection among staff, one request vs. several;
- 303 downloads less RDF content:
 - Think of Amazon and its thousands of products!
- The approaches could be combined.

- RDF is a standard. It's all about standards!
- The RDF data model is very simple and tailored for the Web architecture;
- RDF can be serialized to different formats:
 - RDF/XML;
 - RDFa (embedded in HTML);
 - Turtle;
 - N-Triples;
 - Etc.

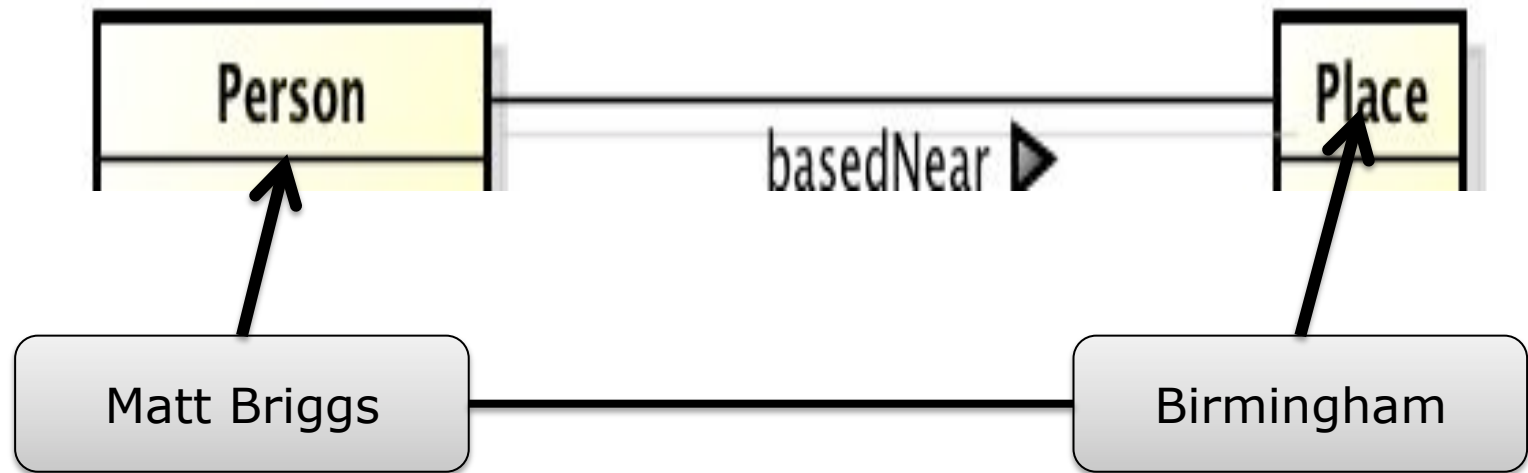


- Node-and-arc-labeled directed graphs;
- Resources are described by a number of triples:

(subject	predicate	object)
----------	-----------	---------

URI (the resource)	Type of relation between subject and object (also URI)	A literal value (string, number, etc.) or URI of another resource.
http://biglynx.co.uk/ people/matt-briggs	http://xmlns.com/foaf/ 0.1/name	Matt Briggs
http://biglynx.co.uk/ people/matt-briggs	http://xmlns.com/foaf/ 0.1/based_near	http://dbpedia.org/ resource/Birmingham

The RDF data model



<http://biglynx.co.uk/people/matt-briggs>

<http://xmlns.com/foaf/0.1/name>

Matt Briggs

<http://biglynx.co.uk/people/matt-briggs>

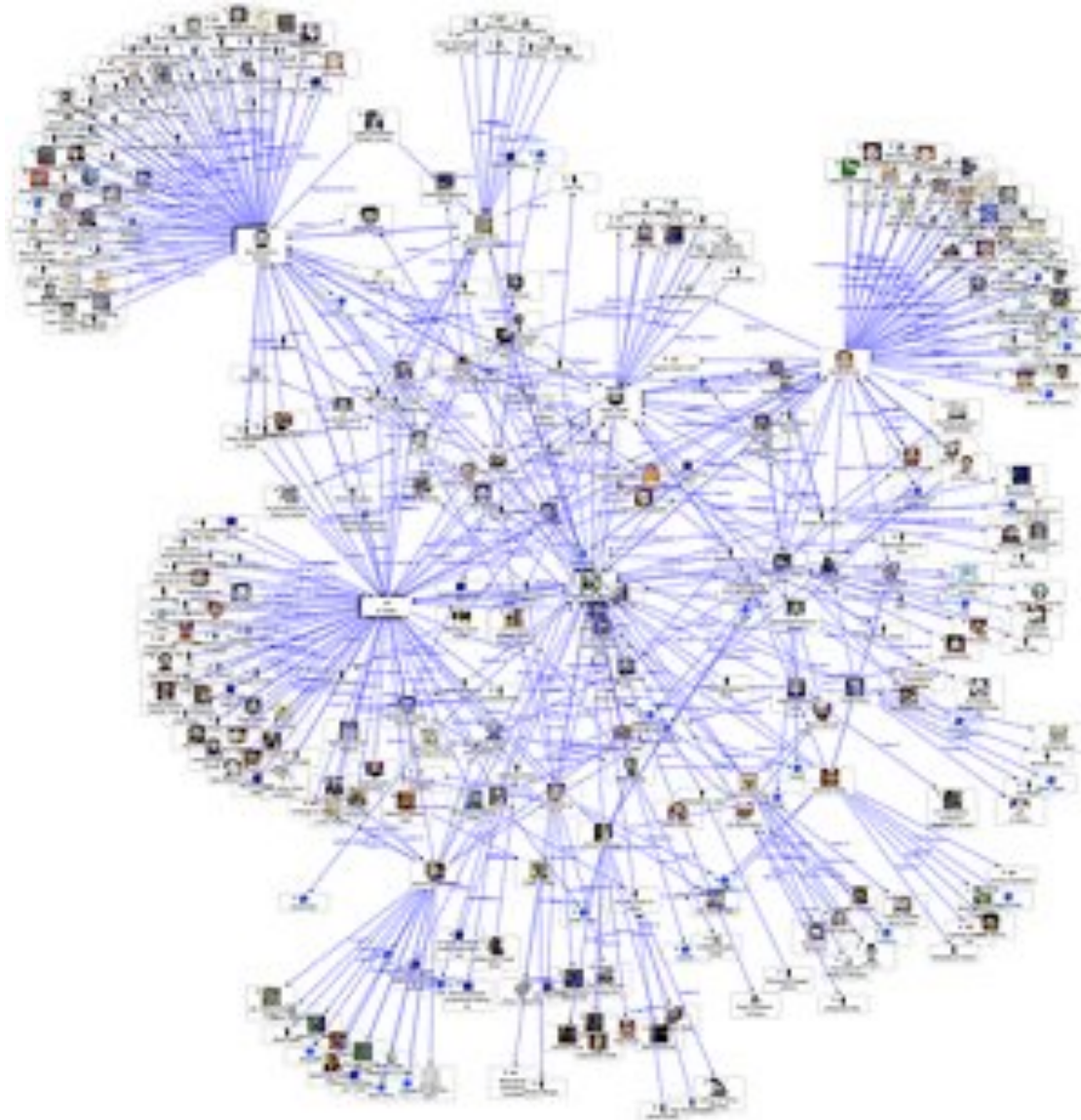
http://xmlns.com/foaf/0.1/based_near

<http://dbpedia.org/resource/Birmingham>

- When the object is an RDF literal (string, number, etc.);
 - E.g.: a person's name or date of birth, etc.;
- Literals can be:
 - Plain: a string with optional language tag (you can have the same property in many languages);
 - Typed: a string combined with a datatype URI from the XML Schema datatype specification.

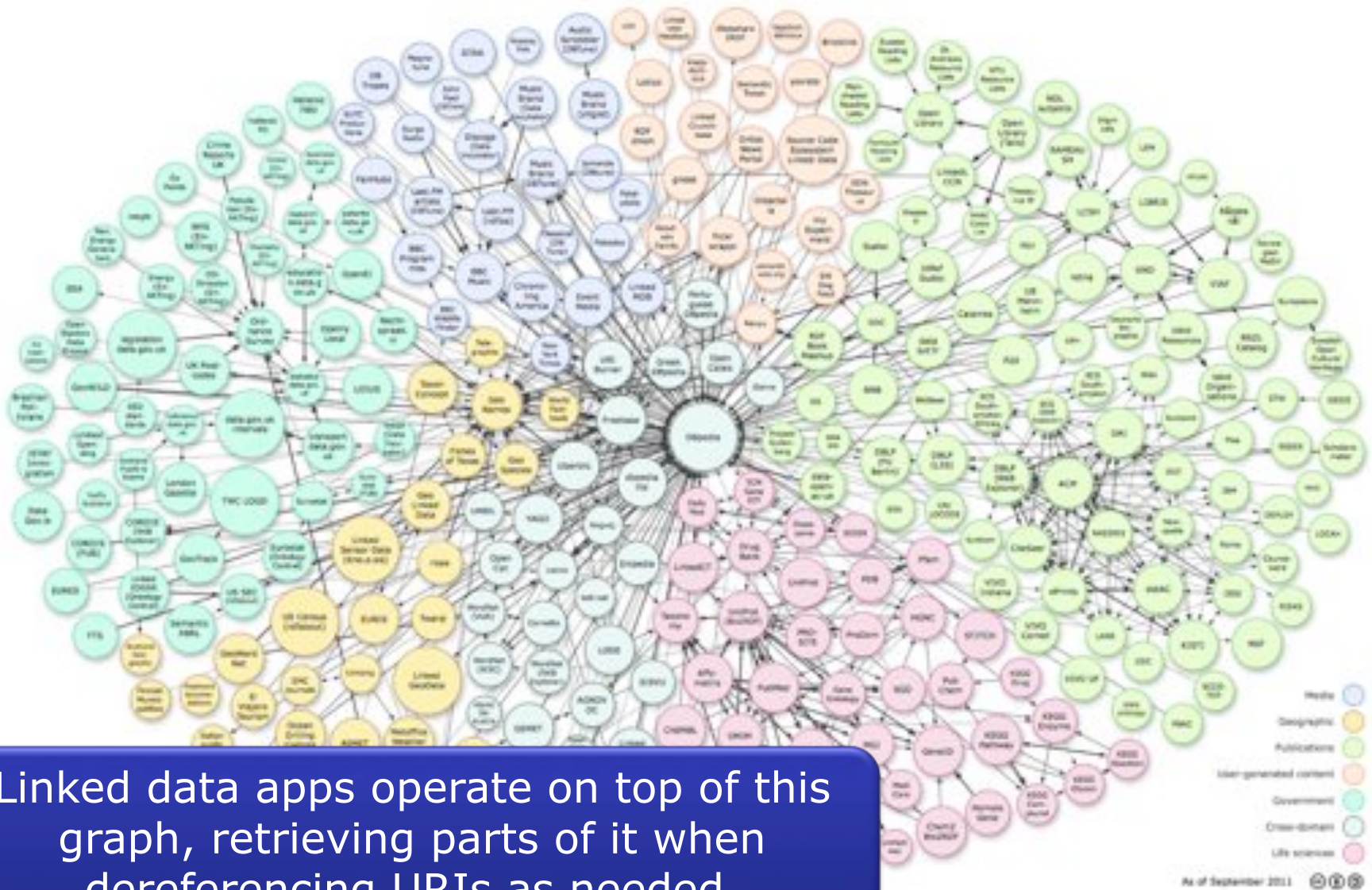
- When the object is a URI, describing the relation between two resources:
 - E.g.: a person's place of birth, a person's friend, etc.
- Links can be:
 - Internal: same source / namespace;
 - External: different source / namespace.

Triples form a graph of things (objects)



- A set of predicates (and their URIs) available for use;
- Example: FOAF (<http://xmlns.com/foaf/spec/>):
 - Classes: Agent, Document, Group, Image, LabelProperty, OnlineAccount, OnlineChatAccount, OnlineEcommerceAccount, OnlineGamingAccount, Organization, **Person**, PersonalProfileDocument, Project
 - Properties: account, accountName, accountServiceHomepage, age, aimChatID, based_near, birthday, currentProject, depiction, depicts, dnaChecksum, familyName, family_name, firstName, focus, fundedBy, geekcode, gender, givenName, givenname, holdsAccount, homepage, icqChatID, img, interest, isPrimaryTopicOf, jabberID, **knows**, lastName, logo, made, maker, mbox, mbox_sha1sum, member, membershipClass, msnChatID, myersBriggs, **name**, nick, openid, page, pastProject, phone, plan, primaryTopic, publications, schoolHomepage, sha1, skypeID, status, surname, theme, thumbnail, tipjar, title, topic, topic_interest, weblog, workInfoHomepage, workplaceHomepage, yahooChatID

Vocabs & data sets form the LOD cloud



Linked data apps operate on top of this graph, retrieving parts of it when dereferencing URIs as needed.

Source: <http://lod-cloud.net>

- URIs can be used in a global scale to refer to anybody or anything;
- Clients can start from one triple and explore the entire data space as desired;
- You can link data from different sources;
- Triples can be merged into a single graph, delimiting the scope as desired and mixing terms from different vocabularies;
- It allows you to use as much or as little structure as desired. More structure can come by combining RDF with other standards like RDFS and OWL.

Not all RDF is good...

- Features that did not gain widespread adoption and are best avoided:
 - RDF reification: difficult to use with SPARQL;
 - RDF collections and containers: use multiple triples with same predicate, unless the relative order of items is important;
 - Blank nodes (without URI): cannot be referenced from outside, so not a good idea.

Since it's better to avoid them,
I didn't bother to learn them...

- RDF is not a **data format**, but a **data model** describing resources as triples;
- To publish it, we must serialize it:
 - In advance (static data set);
 - On demand (dynamic data set).
- Most popular formats:
 - RDF/XML (W3C standard);
 - RDFa (W3C standard);
 - Turtle;
 - N-Triples;
 - RDF/JSON.

- Widely used standard;
- Difficult for humans to read/write;
- Tool support can help.

```
<?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/">

  <rdf:Description rdf:about="http://biglynx.co.uk/people/dave-smith">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
    <foaf:name>Dave Smith</foaf:name>
  </rdf:Description>

</rdf:RDF>
```

- Embeds RDF in HTML pages;
- Good if that's what you need.

```
<!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:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/">
<head>
  <meta http-equiv="Content-Type" content="application/xhtml+xml;
    charset=UTF-8"/>
  <title>Profile Page for Dave Smith</title>
</head>
<body>
  <div about="http://biglynx.co.uk/people#dave-smith"
    typeof="foaf:Person">
    <span property="foaf:name">Dave Smith</span>
  </div>
</body>
</html>
```

- Plain text, no XML;
- More readable by humans;
- Less structured to a computer.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
```

```
<http://biglynx.co.uk/people/dave-smith>  
  rdf:type foaf:Person ;  
  foaf:name "Dave Smith" .
```

- Subset of Turtle (no namespaces or shorthands);
- Redundancy generates larger files;
- On the other hand, each line is complete on itself:
 - It can be parsed a line at a time;
 - One can load large files part by part;
 - Larger size can be remedied with compression.
- N-Triples is the *de facto* standard for exchanging large dumps of Linked Data.

```
<http://biglynx.co.uk/people/dave-smith> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .  
  
<http://biglynx.co.uk/people/dave-smith> <http://xmlns.com/foaf/0.1/name> "Dave Smith" .
```

- Work in progress:
 - <https://dvcs.w3.org/hg/rdf/raw-file/default/rdf-json/index.html>
 - Editor's draft dated November 2013;
- Highly desirable, as the number of programming languages with native JSON supports grow.

```
{  
  "http://example.org/about" : {  
    "http://purl.org/dc/terms/title" : [ { "value" : "My Website",  
                                           "type" : "literal",  
                                           "lang" : "en" } ]  
  }  
}
```

- In our RDF models:
 - Usually, the subject “belongs to us”, i.e., to our own data set / namespace;
 - But often the predicate and object are external:

External predicate == reuse of existing vocabularies.
External objects == building the global data space.

- Types of link:
 - Relationship links;
 - Identity links;
 - Vocabulary links.

- Enable a data set to refer to resources in another data set (and so on...):

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

<http://biglynx.co.uk/people/dave-smith>
  rdf:type foaf:Person ;
  foaf:name "Dave Smith" ;
  foaf:based_near <http://sws.geonames.org/3333125/> ;
  foaf:based_near <http://dbpedia.org/resource/Birmingham> ;
  foaf:topic_interest
    <http://dbpedia.org/resource/Wildlife_photography> ;
  foaf:knows <http://dbpedia.org/resource/David_Attenborough> .
```

- Different people/data sets may talk about the same entities (famous people, places, etc.);
- sameAs links can tell us that different resources (in different data sets usually) talk about the same thing:
 - Representing different opinions;
 - Keeping traceability (a kind of citation);
 - Avoiding a central points of failure.

```
<http://www.dave-smith.eg.uk#me> <http://www.w3.org/2002/07/owl#sameAs> <http://biglynx.co.uk/people/dave-smith> .
```

Imagine GeoNames having to check if URIs already existed for their 8 million locations!

OWL semantics treat RDF statements as facts. In the Semantic Web, however, we should treat them as claims!

- When publishing data, one should try to integrate with existing data sets;
- Two-fold approach to dealing with the heterogeneous data representation:
 - On one hand, try to avoid heterogeneity by reusing terms from existing vocabularies;
 - On the other hand, try to deal with heterogeneity by making data sets as self-descriptive as possible.
- Use a “follow-your-nose” type of integration: URIs can be dereferenced in order to gain more knowledge.

- Defining “small/medium enterprises” in the BigLynx data set, relating it to terms from other vocabularies:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix co: <http://biglynx.co.uk/vocab/sme#> .

<http://biglynx.co.uk/vocab/sme#SmallMediumEnterprise>
  rdf:type rdfs:Class ;
  rdfs:label "Small or Medium-sized Enterprise" ;
  rdfs:subClassOf <http://dbpedia.org/ontology/Company> ;
  rdfs:subClassOf <http://umbel.org/umbel/sc/Business> ;
  rdfs:subClassOf
    <http://sw.opencyc.org/concept/Mx4rvVjQNpwpEbGdrcN5Y29ycA> ;
  rdfs:subClassOf <http://rdf.freebase.com/ns/m/0qb7t> .
```

1. Search for terms in widely used vocabularies;
2. For the terms that are not found, define proprietary vocabulary;
3. If later you discover a vocabulary that defines the term, link them:
 - OWL: sameAs, equivalentClass, equivalentProperty;
 - RDFS: subClassOf, subPropertyOf;
 - SKOS: broadMatch, narrowMatch.

SKOS = Simple Knowledge
Organization System

- The Web of Data is, like the WWW, built on standards;
- Structured, interlinked data forms a global data space;
- We should use linked data instead of other (proprietary) formats because:
 - The data model is unified (triples), designed for global data sharing;
 - The access mechanism is standardized (HTTP is universal);
 - Discovery is based on hyperlinks (URIs). New data sources can be discovered at runtime;
 - Data is self-descriptive (shared vocabularies).

- ★ Data available on the web, whatever format, open license;
- ★★ Data available in machine-readable formats;
- ★★★ As before, but using non-proprietary formats;
- ★★★★ As before, but using W3C open standards so people can link to it;
- ★★★★★ As before, plus having outgoing links to other people's data sets.



Linked Data

THE WEB OF DATA

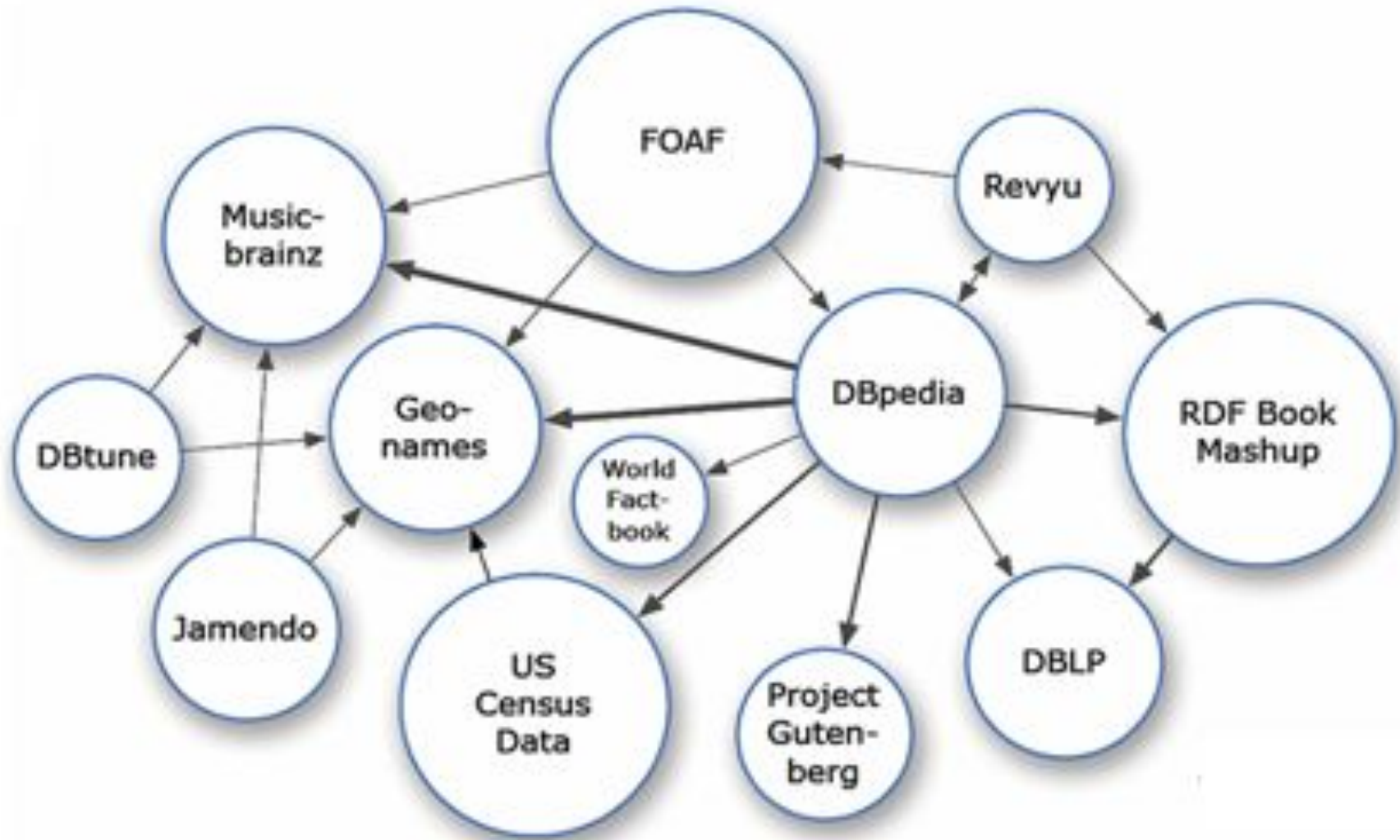
- A giant global graph, billions of RDF statements;
- Numerous sources, all sorts of topics;
- An additional layer to the document Web, tightly interwoven with it, with many of the same properties:
 - Generic (any type of data);
 - Anyone can publish;
 - Can represent disagreements/contradictions;
 - Entities are connected, allowing apps to discover;
 - Publishers are not constrained in their choice of vocabulary;
 - Data is self-describing, dereferenceable over HTTP.

- Origins:
 - The Semantic Web research community;
 - In particular, the W3C Linking Open Data (LOD) project (January 2007).
- Goals of the W3C LOD project:
 - Identify existing open source data sets;
 - Convert them to RDF and publish.

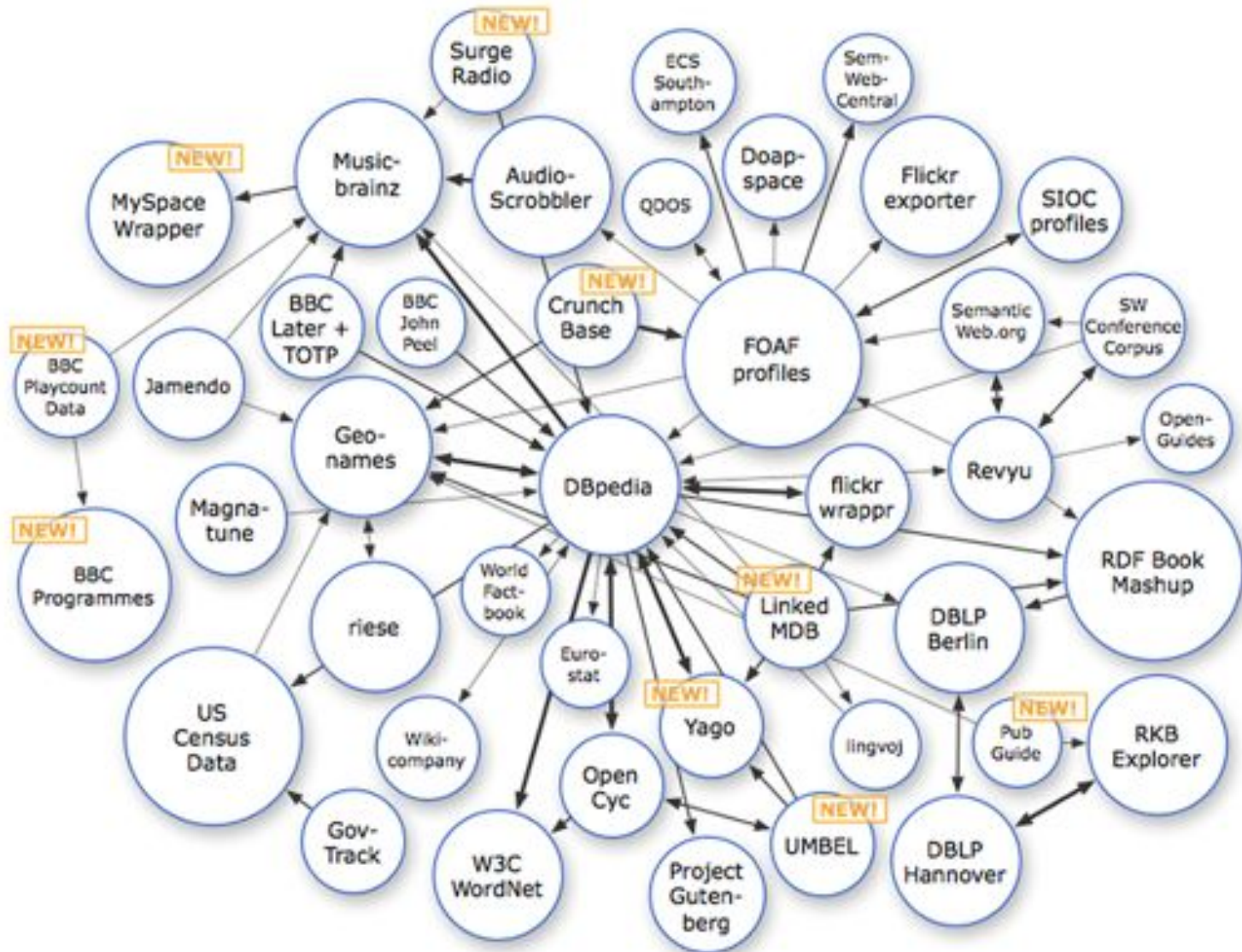


<http://esw.w3.org/topic/SweoIG/TaskForces/CommunityProjects/LinkingOpenData>

Growth of the LOD cloud (2007)

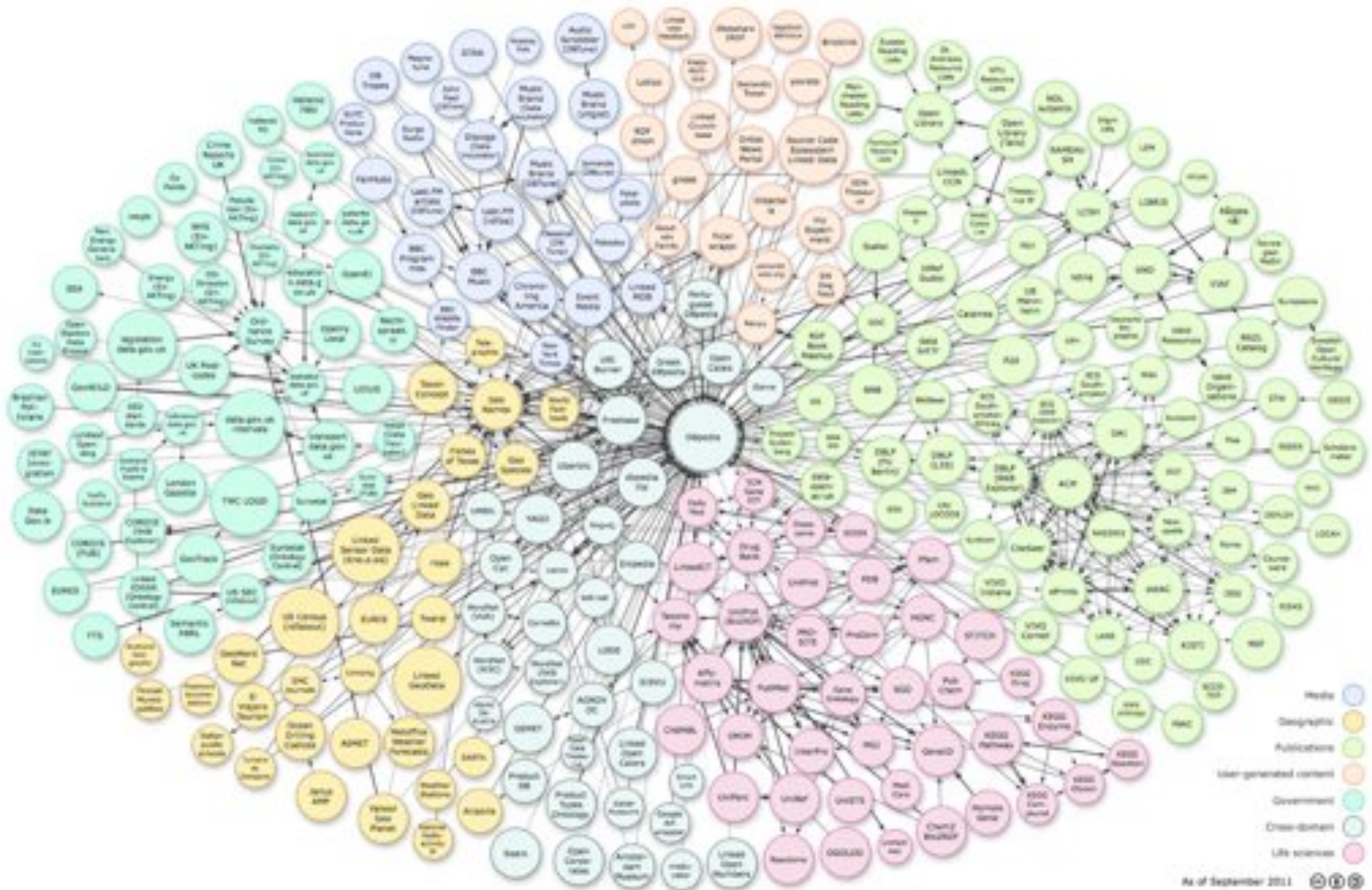


Growth of the LOD cloud (2008)





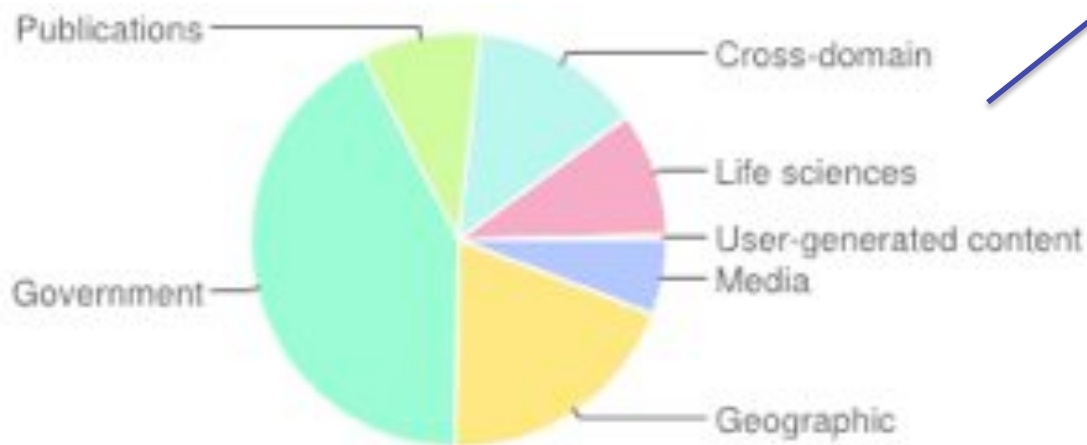
Growth of the LOD cloud (2011)



Source: <http://lod-cloud.net>

Topology of the Web of Data (2011)

Domain	Datasets	Triples	Out-links
Media	25	1.841.852.061	50.440.705
Geographic	31	6.145.532.484	35.812.328
Government	49	13.315.009.400	19.343.519
Publications	87	2.950.720.693	139.925.218
Cross-domain	41	4.184.635.715	63.183.065
Life sciences	41	3.036.336.004	191.844.090
User-generated content	20	134.127.413	3.449.143
	295	31.634.213.770	503.998.829



- One of the first categories;
- Helps connect single-domain datasets into one global data space (the WoD), avoiding data islands;
- Example: DBPedia ([.org](http://dbpedia.org))
 - A crowd-sourced community effort to extract structured information from Wikipedia and make this information available on the Web;
 - Information extracted especially from “info boxes”.

<http://en.wikipedia.org/wiki/Birmingham>



<http://dbpedia.org/page/Birmingham>



- Example: Freebase ([.com](http://freebase.com)):
 - A community-curated database of well-known people, places and things;
 - Editable, openly-licensed;
 - Populated through user contributions, data imports from Wikipedia and Geonames;
 - Linked with DBPedia.



2,506,735,867

Facts
(and counting)

43,905,156

Topics
(and counting)

- Can often interconnect different sets;
- Example: Geonames ([.org](http://www.geonames.org)):
 - Open license;
 - 8 million locations;
- Example: LinkedGeoData ([.org](http://www.linkedgeo.org)):
 - Data form the OpenStreetMap ([.org](http://www.openstreetmap.org)) project;
 - 350 million spacial features;
- Both link to DBPedia.



- BBC:
 - <http://www.bbc.co.uk/ontologies> (programs, music, wildlife, food, sports, politics, etc.);
 - http://www.bbc.co.uk/blogs/legacy/bbcinternet/2010/07/bbc_world_cup_2010_dynamic_sem.html
- The New York Times:
 - <http://data.nytimes.com>
- Thomsom Reuters:
 - <http://www.opencalais.com>
- MusicBrainz:
 - <http://musicbrainz.org>

- Transparency, society participation, etc.
- Examples:
 - Australia: <http://data.gov.au>;
 - New Zealand: <https://data.govt.nz>;
 - The UK: <http://data.gov.uk>;
 - The US: <http://www.data.gov>;
 - Brazil: <http://dados.gov.br>;
- The W3C formed an eGovernment Interest Group:
 - http://www.w3.org/egov/wiki/Main_Page

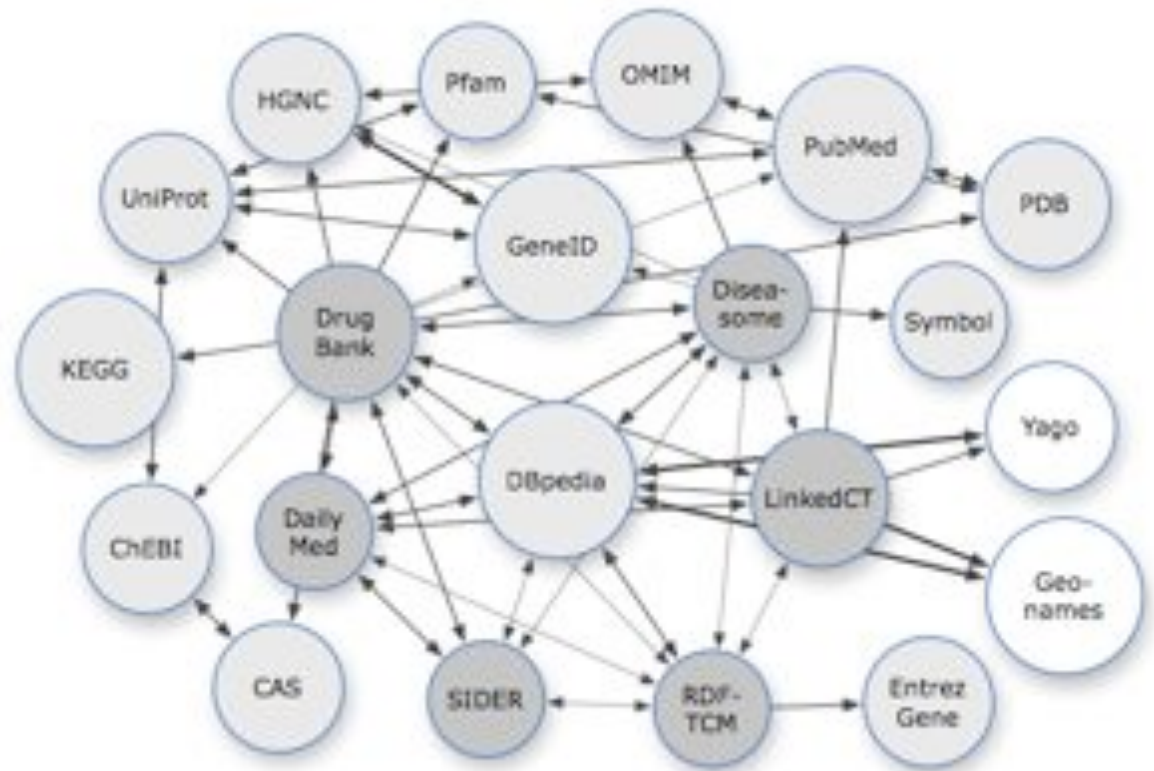
- Integration of library catalogs globally;
- Examples:
 - The U.S. Library of Congress;
 - German National Library of Economics;
 - Swedish National Union Catalogue;
 - The Open Library ([.org](http://openlibrary.org)) – connected to ProductDB;



- Scholarly articles (academic publications):
 - DBLP ([.l3s.de](http://dblp.org));
 - RKBExplorer ([.com](http://rkbexplorer.com));
 - Semantic Web Conference (Dog Food) Corpus:
<http://data.semanticweb.org>
- W3C Library Linked Data Incubator Group:
<http://www.w3.org/2005/Incubator/lld/>



- Biology, genomic, chemistry, drugs, medicine, etc.
- Examples: Bio2Rdf ([.org](http://bio2rdf.org)), the Gene Ontology ([.org](http://geneontology.org));
- W3C Linking Open Drug Data:
<http://www.w3.org/wiki/HCLSIG/LODD>



- The [RDF Book Mashup](#):
 - Integrates Amazon, Google and Yahoo data sources;
 - information about books, their authors, reviews, and online bookstores.
- [GoodRelations](#):
 - Vocabulary for products/services details;
 - Used by Google, Yahoo!, BestBuy, Sears, KMart, ...
- ProductDB ([.org](#)):
 - Aims to be the World's most comprehensive and open source of product data;
 - A page for every product in the world, interlinked.

- [Flickr wrapper](#): extends DBPedia with RDF link to photos posted on flickr;
- Revyu ([.com](#)): user reviews and ratings (on anything);
- Faviki ([.com](#)): annotate Web content with URIs;
- Some wikis are now using [Semantic MediaWiki](#), which annotates pages with RDF;
- Facebook ([.com](#)) and IMDB ([.com](#)) uses the Open Graph Protocol ([.org](#));
- Drupal ([.org](#)) CMS version 7 enables description of entities in RDFa.

The Web of Data is growing...

- Linked data started with academics and enthusiasts;
- Today, more and more businesses and government are interested;
- We can expect great things in the future!

Linked Data

LINKED DATA DESIGN CONSIDERATIONS

- When preparing data to publish, we should take some design considerations into account;
 - How one shapes and structures data to fit neatly in the Web?
- Related to the Linked Data principles:
 - Naming things with URIs;
 - Describing things with RDF;
 - Making links to other data sets.

- Naming things with URIs means a Web server should respond with data when the URI is dereferenced;
 - We should aim to create “cool” (stable, working) URIs;
1. Keep out of namespaces you don't control:
 - Otherwise, you can't dereference it!
 - Use sameAs.
 2. Abstract away from implementation details:
 - Machines change name, technologies change;
 - Your URI should keep the same.

- Uncool URIs:
 - <http://www.imdb.com/title/tt0057012/#film> (only the people at imdb.com can dereference this!);
 - <http://tiger.biglynx.co.uk/people/dave-smith> (the machine “tiger” might change name, or die);
 - <http://biglynx.co.uk:8080/people.php?id=dave-smith&format=rdf> (the port might change, you might want to switch from PHP to something else).
- It doesn't mean you can't use these technologies:
 - Redirectors like `mod_rewrite` in Apache can help.

- When possible, use natural keys within URIs:

```
<http://biglynx.co.uk/vocab/sme#SmallMediumEnterprise>  
  rdfs:subClassOf <http://dbpedia.org/ontology/Company> ;  
  rdfs:subClassOf <http://umbel.org/umbel/sc/Business> ;  
  rdfs:subClassOf  
    <http://sw.opencyc.org/concept/Mx4rvVjQNpwpEbGdrcN5Y29ycA> ;  
  rdfs:subClassOf <http://rdf.freebase.com/ns/m/0qb7t> .
```

- Example – representing people:
 - Name + surname: very readable, but may change;
 - The persistence ID / DB PK: not readable;
 - The tax code: a good compromise?

- Entities can have 3 URIs:
 1. For the object itself;
 2. For an HTML description of the object;
 3. For an RDF/XML description of the object.
- When dereferenced, URL 1 should redirect to URL 2 or URL 3 depending if human or machine, respectively.



Matt Briggs



matt-briggs



matt-briggs.rdf



matt-briggs.html

- Separate by path (e.g.: DBPedia):
 1. <http://dbpedia.org/resource/Object>
 2. <http://dbpedia.org/page/Object>
 3. <http://dbpedia.org/data/Object>
- Disadvantage: not very visually distinct.

- Separate by subdomain:
 1. `http://id.biglynx.co.uk/Object`
 2. `http://pages.biglynx.co.uk/Object`
 3. `http://data.biglynx.co.uk/Object`

- Separate by extension (e.g., BigLynx):
 1. <http://biglynx.co.uk/people/dave-smith>
 2. <http://biglynx.co.uk/people/dave-smith.html>
 3. <http://biglynx.co.uk/people/dave-smith.rdf>

- URIs are cool, stuff is being dereferenced, OK!
- Now, what info to provide when people look it up?
 1. Data properties (triples with literals for objects);
 2. Object properties (triples with URIs for objects);
 3. Incoming links from other resources;
 4. Triples describing related resources;
 5. Meta-data on the description itself;
 6. Triples about the broader data set.

1 & 2 - Literal triples and outgoing links

- Triples with the resource as subject;
- Describe the resource;
- Prefer using more widely supported predicates (rdfs:label, foaf:name, rdfs:comment, etc.)

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rel: <http://purl.org/vocab/relationship/> .

<http://biglynx.co.uk/people/dave-smith>
  rdf:type foaf:Person ;
  foaf:name "Dave Smith" ;
  foaf:based_near <http://sws.geonames.org/3333125/> ;
  foaf:based_near <http://dbpedia.org/resource/Birmingham> ;
  foaf:topic_interest
    <http://dbpedia.org/resource/Wildlife_photography> ;
  foaf:knows <http://dbpedia.org/resource/David_Attenborough> ;
  rel:employerOf <http://biglynx.co.uk/people/matt-briggs> .
```

- Useful for finding other resources (navigate back);
- Sometimes infeasible (imagine DBPedia!);
- Sometimes redundant (bi-directional relations):
employerOf / employedBy below.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rel: <http://purl.org/vocab/relationship/> .

<http://biglynx.co.uk/people/dave-smith>
  rdf:type foaf:Person ;
  <!-- ... -->
  rel:employerOf <http://biglynx.co.uk/people/matt-briggs> .

<http://biglynx.co.uk/people/dave-smith.rdf>
  foaf:primaryTopic <http://biglynx.co.uk/people/dave-smith> .

<http://biglynx.co.uk/people/matt-briggs>
  rel:employedBy <http://biglynx.co.uk/people/dave-smith> .
```

- If X is related to Y, have a copy of some triples from Y.rdf in X.rdf;
- Its use is controversial:
 - Some think it can avoid lots of dereferencing;
 - Some think that it creates redundancy that applications will have to handle;
 - I think it creates a maintenance issue.
- Should be decided based on intended uses of the data.

Meta-triples (meta-meta-data?)

- Triples that describe the description (i.e., the RDF file);
- Can be very useful;
- Are often overlooked.



Source: <http://xkcd.com/917/>

- About the RDF document that describes a resource:
 - Who's the author?
 - How current is the data?
 - What's the license to use?
- There are two primary mechanisms:
 - Semantic Sitemaps;
 - Vocabulary of Interlinked Datasets (voID).

- Extension to the Sitemaps protocol for crawlers:
 - sitemap.xml document with basic website info;
- Semantic extensions include:
 - Label and URI for the dataset;
 - Sample URIs from the dataset;
 - SPARQL endpoints for data dumps;
 - Etc.

Semantic Sitemaps for BigLynx

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset
  xmlns="http://www.sitemaps.org/schemas/sitemap/0.9"
  xmlns:sc="http://sw.deri.org/2007/07/sitemapextension/scschema.xsd">

  <sc:dataset>
    <sc:datasetLabel>Big Lynx People Data Set</sc:datasetLabel>
    <sc:datasetURI>http://biglynx.co.uk/datasets/people
                                </sc:datasetURI>
    <sc:linkedDataPrefix>http://biglynx.co.uk/people/
                                </sc:linkedDataPrefix>
    <sc:sampleURI>http://biglynx.co.uk/people/dave-smith
                                </sc:sampleURI>
    <sc:sampleURI>http://biglynx.co.uk/people/matt-briggs
                                </sc:sampleURI>
    <sc:sparqlEndpointLocation>http://biglynx.co.uk/sparql
                                </sc:sparqlEndpointLocation>
    <sc:dataDumpLocation>http://biglynx.co.uk/dumps/people.rdf.gz
                                </sc:dataDumpLocation>
    <changefreq>monthly</changefreq>
  </sc:dataset>
</urlset>
```

- De facto standard;
- Represents the metadata in RDF itself:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix dcterms: <http://purl.org/dc/terms/> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix rel: <http://purl.org/vocab/relationship/> .
```

The document

```
<http://biglynx.co.uk/people/dave-smith.rdf>  
  foaf:primaryTopic <http://biglynx.co.uk/people/dave-smith> ;  
  rdf:type foaf:PersonalProfileDocument ;  
  rdfs:label "Dave Smith's Personal Profile in RDF" ;  
  dcterms:creator <http://biglynx.co.uk/people/nelly-jones> .
```

```
<http://biglynx.co.uk/people/dave-smith>  
  rdf:type foaf:Person ;  
  <!-- ... -->  
  rel:employerOf <http://biglynx.co.uk/people/matt-briggs> .
```

The person

- Provenance = where the data came from;
- Very important;
- RDF triples describing the document in which the original data is contained;
- Options:
 - Dublin Core (widely used);
 - Open Provenance Model (more expressive, W3C);
 - NG4J (for digital signatures).

```
<rdf:Description>  
  <dc:title>Internet Ethics</dc:title>  
  <dc:creator>Duncan Langford</dc:creator>  
  <dc:format>Book</dc:format>  
  <dc:identifier>ISBN 0333776267</dc:identifier>  
</rdf:Description>
```

- Absence of license \neq free to use. Default is ©;
- If terms of data reuse are unclear, some people/organizations might be cautious about it;
- Importance of licenses and waivers.

Are facts copyrightable? CC applies on “creative works”. Facts don’t seem to be “creative works”.



- Example: a post in Big Lynx blog is copyrightable, so an RDF document that describes it can denote its license:

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix sioc: <http://rdfs.org/sioc/ns#> .
@prefix cc: <http://creativecommons.org/ns#> .

<http://biglynx.co.uk/blog/making-pacific-sharks>
  rdf:type sioc:Post ;
  dcterms:title "Making Pacific Sharks" ;
  dcterms:date "2010-11-10T16:34:15Z"^^xsd:dateTime ;
  sioc:has_container <http://biglynx.co.uk/blog/> ;
  sioc:has_creator <http://biglynx.co.uk/people/matt-briggs> ;
  foaf:isPrimaryTopicOf
    <http://biglynx.co.uk/blog/making-pacific-sharks.rdf> ;
  foaf:isPrimaryTopicOf
    <http://biglynx.co.uk/blog/making-pacific-sharks.html> ;
  cc:license <http://creativecommons.org/licenses/by-sa/3.0/> .
```

- RDF documents with facts are not really copyrightable;
- We could still add waivers to define expectations on how we think the data should be used;
- E.g., we'd like to be credited as the source of the data when it's reused and republished.

```
<http://biglynx.co.uk/people/dave-smith.rdf>
  foaf:primaryTopic <http://biglynx.co.uk/people/dave-smith> ;
  rdf:type foaf:PersonalProfileDocument ;
  rdfs:label "Dave Smith's Personal Profile in RDF" ;
  dcterms:creator <http://biglynx.co.uk/people/nelly-jones> ;
  dcterms:publisher <http://biglynx.co.uk/company.rdf#company> ;
  dcterms:date "2010-11-05"^^xsd:date ;
  dcterms:isPartOf <http://biglynx.co.uk/datasets/people> ;
  vv:waiver <http://www.opendatacommons.org/odc-public-domain-
dedication-and-licence/> ;
  vv:norms <http://www.opendatacommons.org/norms/odc-by-sa/> .
```

Design considerations checklist

- Name resources with cool URIs; ✓
- Describe resources and their documents; ✓
- Next:
 - Which vocabularies should I use?
 - Which data sets should I link to?



- RDF = subject, predicate, object (generic);
- Domain specific terms to describe classes of things in the world:
 - SKOS: for expressing conceptual hierarchies (taxonomies);
 - RDFS & OWL (RDFS++): for describing conceptual models in terms of classes and their properties.
- Example:
 - We create an RDFS document about the class Dog and their properties (hasColor, hasName, etc.);
 - People create RDF documents about their dogs.

- Language for describing lightweight ontologies in RDF;
- Literally, the schema of RDF;
- Two namespaces: **rdfs:** (e.g., rdfs:Class) and **rdf:** (e.g., rdf:Property).

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix owl: <http://www.w3.org/2002/07/owl#> .  
@prefix dcterms: <http://purl.org/dc/terms/> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix cc: <http://creativecommons.org/ns#> .  
@prefix prod: <http://biglynx.co.uk/vocab/productions#> .
```

```
prod:Production  
  rdf:type rdfs:Class;  
  rdfs:label "Production";  
  rdfs:comment "the class of all productions".
```

```
prod:Director
```

```
  rdf:type rdfs:Class;  
  rdfs:label "Director";  
  rdfs:comment "the class of all directors";  
  rdfs:subClassOf foaf:Person.
```

```
prod:director
```

```
  rdf:type owl:ObjectProperty;  
  rdfs:label "director";  
  rdfs:comment "the director of the production";  
  rdfs:domain prod:Production;  
  rdfs:range prod:Director;  
  owl:inverseOf prod:directed.
```

```
prod:directed
```

```
  rdf:type owl:ObjectProperty;  
  rdfs:label "directed";  
  rdfs:comment "the production that has been directed";  
  rdfs:domain prod:Director;  
  rdfs:range prod:Production;  
  rdfs:subPropertyOf foaf:made.
```

Using our own RDF schema

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix prod: <http://biglynx.co.uk/vocab/productions#> .

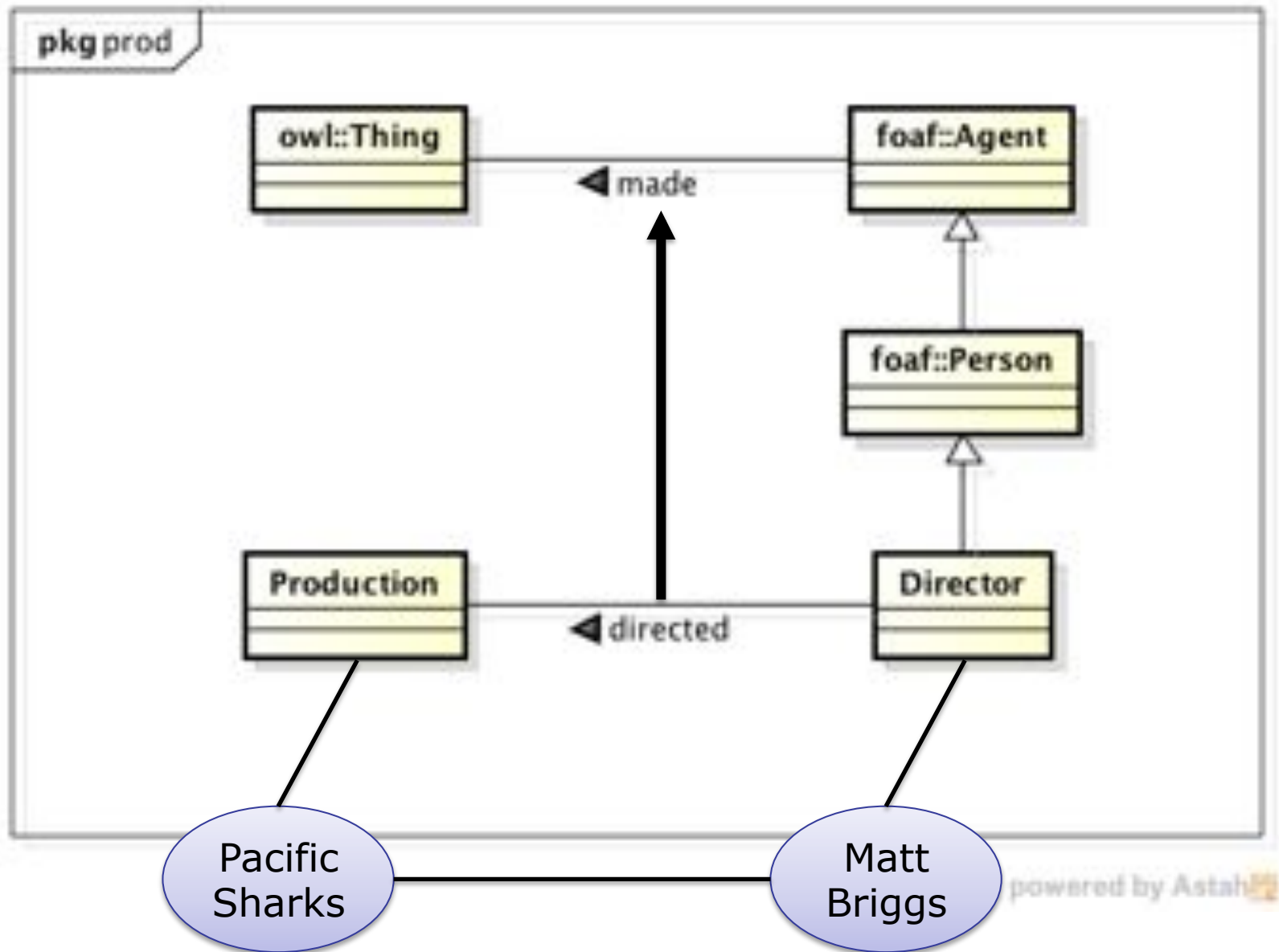
<http://biglynx.co.uk/people/matt-briggs>
  rdf:type
    foaf:Person ,
    prod:Director ;
  foaf:name "Matt Briggs" ;
  foaf:based_near <http://sws.geonames.org/3333125/> ;
  foaf:based_near <http://dbpedia.org/resource/Birmingham> ;
  foaf:topic_interest
    <http://dbpedia.org/resource/Wildlife_photography> ;
  prod:directed <http://biglynx.co.uk/productions/pacific-sharks> .
```

This is just an example. A real dataset should reuse some media vocabulary (e.g., BBC's)

- Recommended to guide potential users of your ontologies:
 - `rdfs:label` = human-readable name for the resource;
 - `rdfs:comment` = human-readable description for the resource.

- Facts about resources can be inferred about their classes' properties:
 - `rdfs:subClassOf`: inheritance (as in OO);
 - `rdfs:subPropertyOf`: inheritance for object properties (associations between classes);
 - `rdfs:domain` = the origin of the association;
 - `rdfs:range` = the destination of the association.

The BigLynx production vocabulary in UML



- Extends the expressivity of RDFS;
 - owl:equivalentClass
 - owl:equivalentProperty } Mappings between terms from different data sets
- owl:InverseFunctionalProperty = domain is unique;
- owl:inverseOf = a property is the inverse of another.

- Avoid reinventing the wheel;
- Some applications may already be “tuned” to existing vocabularies, avoiding further pre-processing;
- Example vocabularies with common types:
 - **Dublin Core Metadata Initiative (DCMI)**: general metadata attributes such as title, creator, date and subject;
 - **Friend-of-a-Friend (FOAF)**: terms for describing people, their activities and their relations to other people and objects.



- **Semantically-Interlinked Online Communities (SIOC)**: aspects of online community sites, such as users, posts and forums;
- **Description of a Project (DOAP)**: software projects, particularly those that are Open Source;
- **Music Ontology**: various aspects related to music, such as artists, albums, tracks, performances and arrangements;
- **Programmes Ontology**: programs such as TV and radio broadcasts;
- **Good Relations Ontology**: products, services and other aspects relevant to e-commerce applications;

- **Creative Commons (CC)**: copyright licenses in RDF;
- **Bibliographic Ontology (BIBO)**: citations and bibliographic references (e.g., quotes, books, articles);
- **OAI Object Reuse and Exchange vocabulary**: resource aggregations such as different editions of a document or its internal structure;
- **Review Vocabulary**: reviews and ratings, as are often applied to products and services;
- **Basic Geo (WGS84)**: latitude and longitude for describing geographically-located things.

- When something is **really** new, the new term should at least be mapped (related) to existing ones:
 - A BigLynx director is a FOAF person;
- Consider the *Materialize Inferences pattern*:
 - Redundantly describe the resource's classes:

```
<http://biglynx.co.uk/people/matt-briggs>  
  rdf:type  
    foaf:Person ,  
    prod:Director .
```

- The inference Director -> Person could be made by a reasoner, but not all applications use reasoning.

Selecting vocabularies to use

- There's no definitive directory for searching vocabs;
- Good starting places:
 - Swoogle ([.umbc.edu](http://umbc.edu));
 - Sindice (.com);
 - Linked Open Vocabularies (lov.okfn.org);
 - The LOD cloud diagram (lod-cloud.net);
 - Sig.ma.



Linked Open Vocabularies (LOV)



- **Usage and uptake:** is the vocab in widespread usage?
Will it make my data set more accessible to apps?
- **Maintenance and governance:** is it actively maintained? When / on what basis are updates made?
- **Coverage:** does it cover enough of the data set to justify adopting its terms and ontological commitments?
- **Expressivity:** is the degree of expressivity appropriate for my application scenario?



- Supplement existing vocabs (with RDFS/OWL) rather than reinventing one from scratch;
- Use only namespaces you control;
- Apply the linked data principles (we saw before);
- Provide documentation with `rdfs:label` / `comment`;
- Only define things that matter, don't over-specify the vocabulary for no good reason.

neologism

<http://neologism.deri.ie>



open source



protégé

<http://protege.stanford.edu>



TopBraid Composer™
Maestro Edition

<http://www.topquadrant.com/tools/IDE-topbraid-composer-maestro-edition/>

- Related resources should be linked;
- Crawlers should be able to go from one to the other;
- Even better if external data sources link to yours, but you may need to convince them:
 - Is your data set novel?
 - Is something now achievable and it wasn't before?
 - What's the cost to create links to you?

(You could write the links and send them ready for them. It has been done successfully with DBPedia)

- Advantages:
 - Can be dereferenced, opening more information;
 - Can further link to other vocabs, to the LOD cloud...
- Things to consider, however:
 - What is the value of the data in the target data set?
 - To what extent does this add value to your data set?
 - Is the target data set and its namespace under stable ownership and active maintenance?
 - Are the URIs in the data set stable?
 - Are there ongoing links to other data set so that applications can tap into a network of interconnected data sources?

The same applies to reusing vocabs!

- Using foaf:knows links you to FOAF!
- Consider:
 - How widely is the predicate already used for linking by other data sources?
 - Is the vocabulary well maintained and properly published with dereferenceable URIs?

- Manual linking typically done for small/static data sets:
 - Search the resource to link (cf. slide 100);
 - Check the URI of the object, not its document;
 - Add the link to your data set.
- Doesn't scale for large data sets:
 - E.g., link DBPedia's 413.000 places to Geonames;
- Inspiration from DB/ontology matching communities: record linkage problem.

- Key-based approach:
 - Datasets contain common identifiers: usually inverse functional properties, could also be part of the URI;
 - Then use simple pattern-based algorithms;
 - E.g.: ProductDB has URIs by GTIN (Global Trade Item Number) and ISBN;
- Similarity-based approach:
 - No common ID, must compare multiple properties as well as interlinked entities with complex algorithm;
 - E.g.: DBPedia x Geonames – names, long/lat, country name, population, etc.
 - Some tools help: Silk, Limes, RiMOM, idMash, ...

Don't forget maintenance!

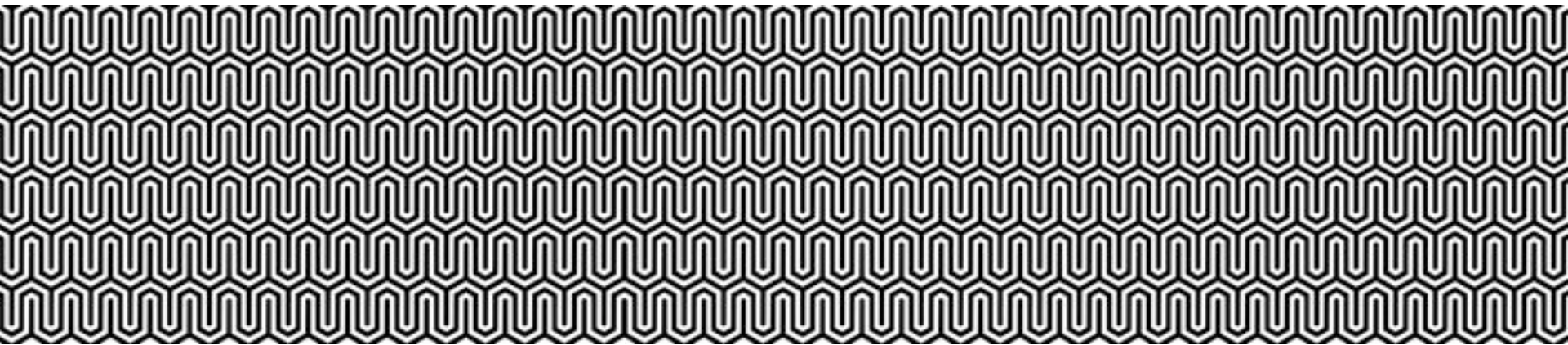
- If URIs die or change, we should fix them;
- DSNotify (<http://www.cibiv.at/~niko/dsnotify/>): tool that monitors data sources and notify about changes.

A W3C task force maintains a list of tools for matching and maintenance: <http://www.w3.org/wiki/TaskForces/CommunityProjects/LinkingOpenData/EquivalenceMining>

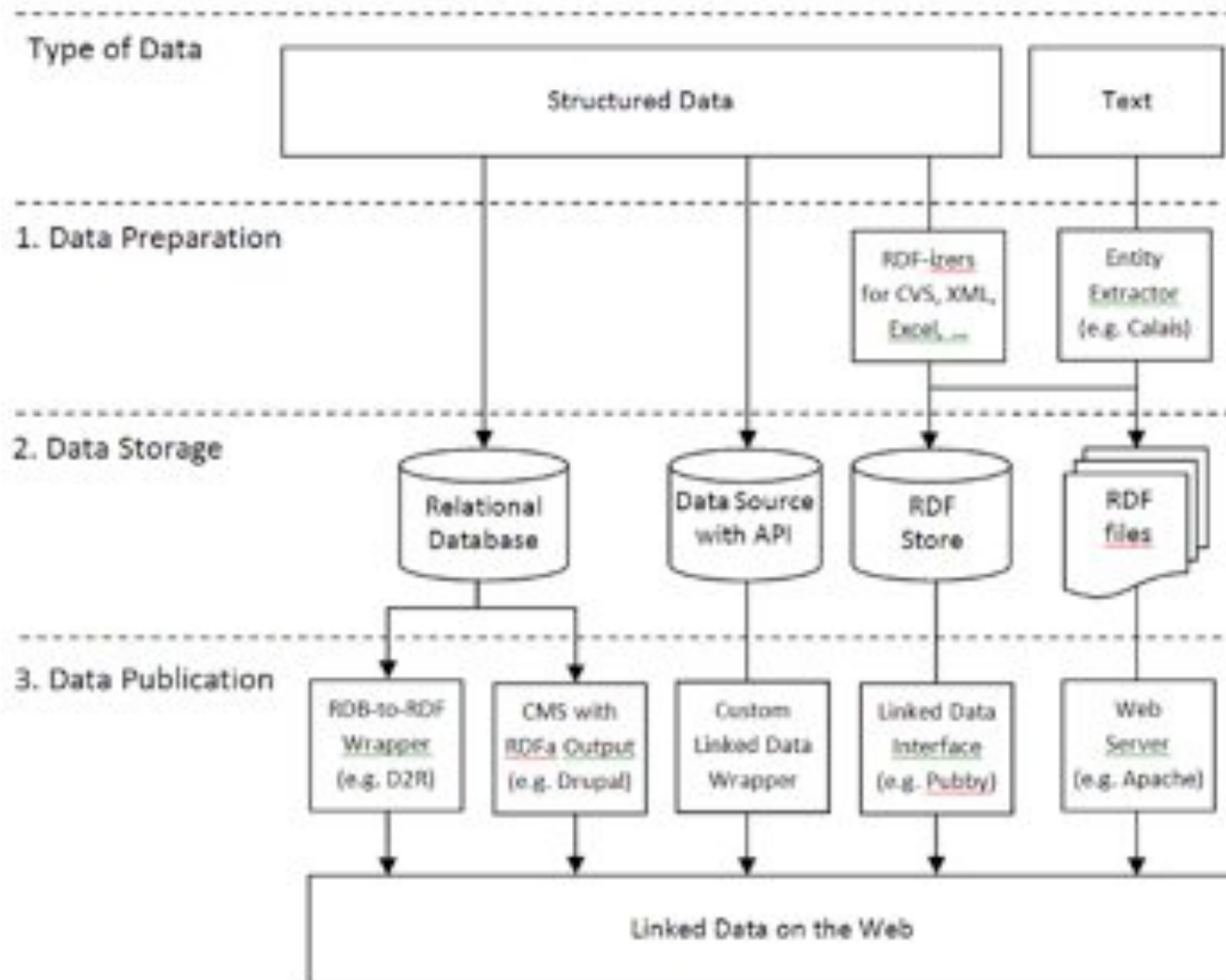
Linked Data

RECIPES FOR PUBLISHING LINKED DATA

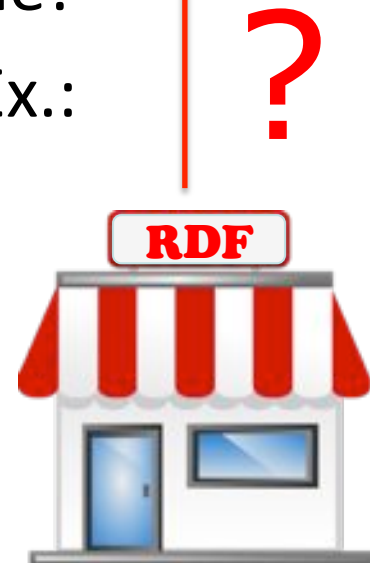
- There are many common patterns for publishing LD;
- LD complements, rather than replaces, existing infrastructure;
- These patterns build on the design considerations we talked about earlier.



Workflows of common LD pub. patterns



- **From “query-able” infrastructure:** use a RDBMS-to-RDF wrapper or create your own custom wrapper (especially when accessing data behind an API);
- **From static structured data (CSV, XML, DB dumps):** perform conversion to RDF (files or store).
Tools available: w3.org/wiki/ConverterToRdf;
- **If data already in RDF:** use a web server, done!
- **From plain text:** use an LD entity extractor. Ex.:
 - Calais (opencalais.com);
 - DBPedia Spotlight (spotlight.dbpedia.org).



- A database built and optimized especially for the storage and retrieval of triples (subj-pred-obj);
- Some examples:
 - AllegroGraph (franz.com/agraph/allegrograph/);
 - Jena (jena.apache.org);
 - Virtuoso (virtuoso.openlinksw.com);
 - SparkleDB (sparkledb.net);
 - RDFLib (github.com/RDFLib/rdfliib);
 - OpenRDF/Sesame (openrdf.org);
 - Many more (see en.wikipedia.org/wiki/Triplestore).

- How much data needs to be served?
 - RDF files or triple store?
 - Split large RDF files to avoid waste of bandwidth (stores usually deliver one “file” per entity);
- How often does the data change?
 - Not much: static files can suit you well;
 - Otherwise, use some kind of store.

Recipe: using static RDF/XML files

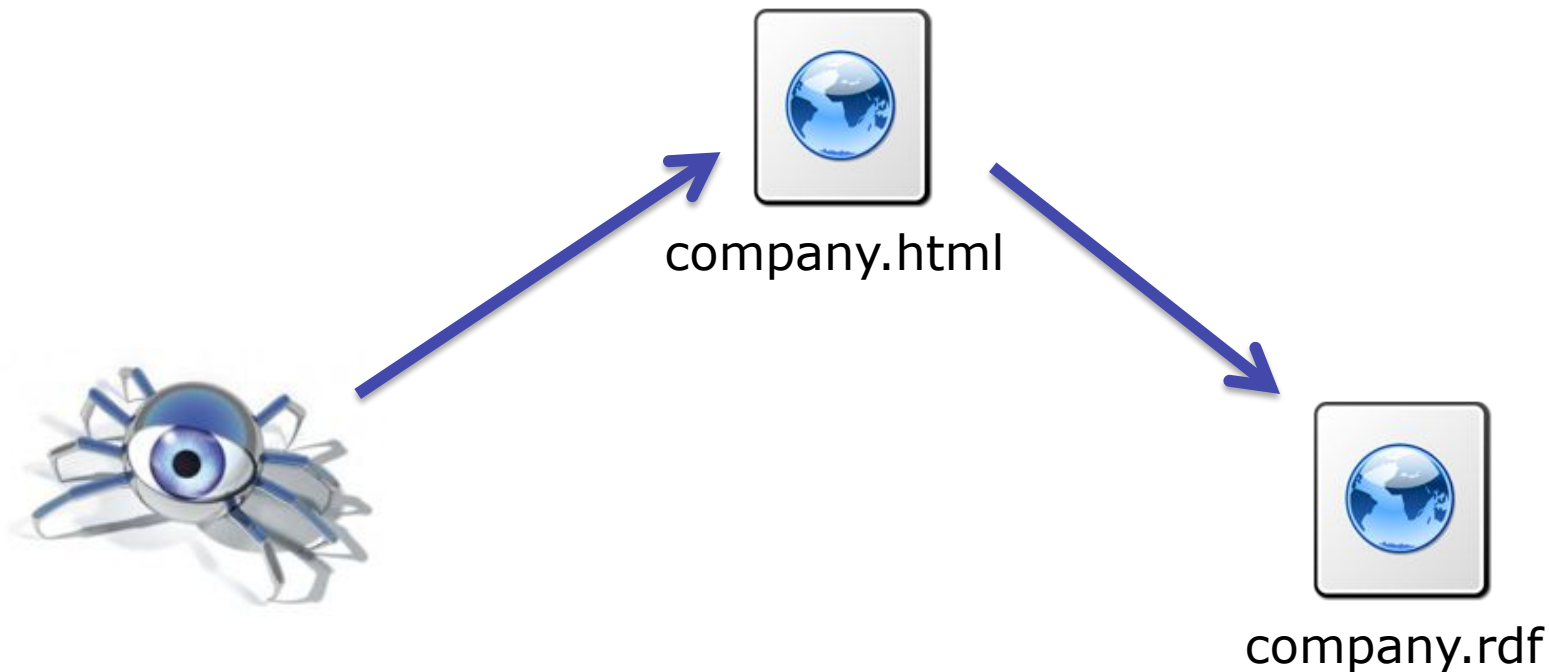
- Prefer RDF/XML over Turtle/N-Triples because of tool support from parsers;
- Files can be created manually or with a tool;
- Then all you need is a web server that is able to serve application/rdf+xml:

```
# In Apache Web Server:  
AddType application/rdf+xml .rdf  
AddType text/n3; charset=utf-8 .n3  
AddType text/turtle; charset=utf-8 .ttl
```

- See, e.g., Big Lynx: biglynx.co.uk/company.rdf and other documents.

- Use the *Autodiscovery* pattern:

```
<link rel="alternate" type="application/rdf+xml" href="company.rdf">
```



Source: <http://patterns.dataincubator.org/book/autodiscovery.html>

RDF embedded in HTML files (RDFa)

```
<?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 xml:lang="en" version="XHTML+RDFa 1.0"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:sme="http://biglynx.co.uk/vocab/sme#">
<head>
  <title>About Big Lynx Productions Ltd</title>
  <meta property="dcterms:title"
    content="About Big Lynx Productions Ltd" />
  <meta property="dcterms:creator" content="Nelly Jones" />
  <link rel="rdf:type" href="foaf:Document" />
  <link rel="foaf:topic" href="#company" />
</head>
<body>
  <h1 about="#company" typeof="sme:SmallMediumEnterprise"
    property="foaf:name" rel="foaf:based_near" resource="http://
    sws.geonames.org/3333125/">Big Lynx Productions Ltd</h1>
```

```
<div about="#company" property="dcterms:description">Big Lynx
  Productions Ltd is an independent television production company
  based near Birmingham, UK, and recognised worldwide for its
  pioneering wildlife documentaries</div>
<h2>Teams</h2>
  <ul about="#company">
    <li rel="sme:hasTeam">
      <div about="http://biglynx.co.uk/teams/management"
        typeof="sme:Team">
        <a href="http://biglynx.co.uk/teams/management"
          property="rdfs:label">The Big Lynx Management Team</a>
        <span rel="sme:isTeamOf" resource="#company"></span>
        <span rel="sme:leader" resource="http://biglynx.co.uk/
          people/dave-smith"></span>
      </div>
    </li>
    <li rel="sme:hasTeam">
      <div about="http://biglynx.co.uk/teams/production"
        typeof="sme:Team">
        <!-- ... -->
```

- You can use a RDFa distiller and parser to get:

```
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix sme: <http://biglynx.co.uk/vocab/sme#> .

<http://biglynx.co.uk/company.html> a <foaf:Document> ;
  dcterms:creator "Nelly Jones"@en ;
  dcterms:title "About Big Lynx Productions Ltd"@en ;
  foaf:topic <http://biglynx.co.uk/company.html#company> .

<http://biglynx.co.uk/company.html#company>
  a sme:SmallMediumEnterprise ;
  sme:hasTeam
    <http://biglynx.co.uk/teams/management>,
    <http://biglynx.co.uk/teams/production>,
    <http://biglynx.co.uk/teams/web> ;

...
```



Community

[Community Home](#)[Getting Involved](#)[Chat](#)[Mailing Lists](#)[Member Directory](#)[Forum](#)

RDF/RDFa (D7)

Last updated August 5, 2013.

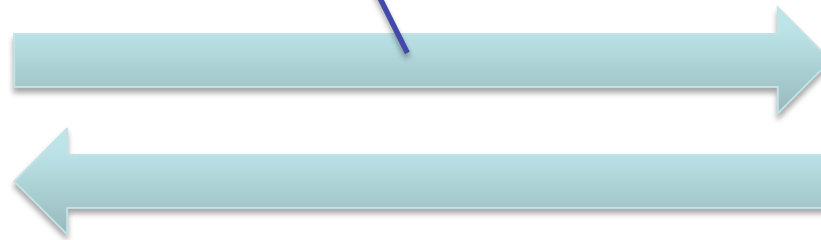
RDF/RDFa is major new functionality in Drupal 7 which describes Drupal entities (nodes, user, comments, terms) and their relationship in a format machines can understand. Mappings are defined between local Drupal entities and widely used vocabularies like Dublin Core, FOAF, SIOC, SKOS, etc. These mappings are stored in arrays and can be defined via programming in the modules. They can also be altered the same way as forms, links, etc.

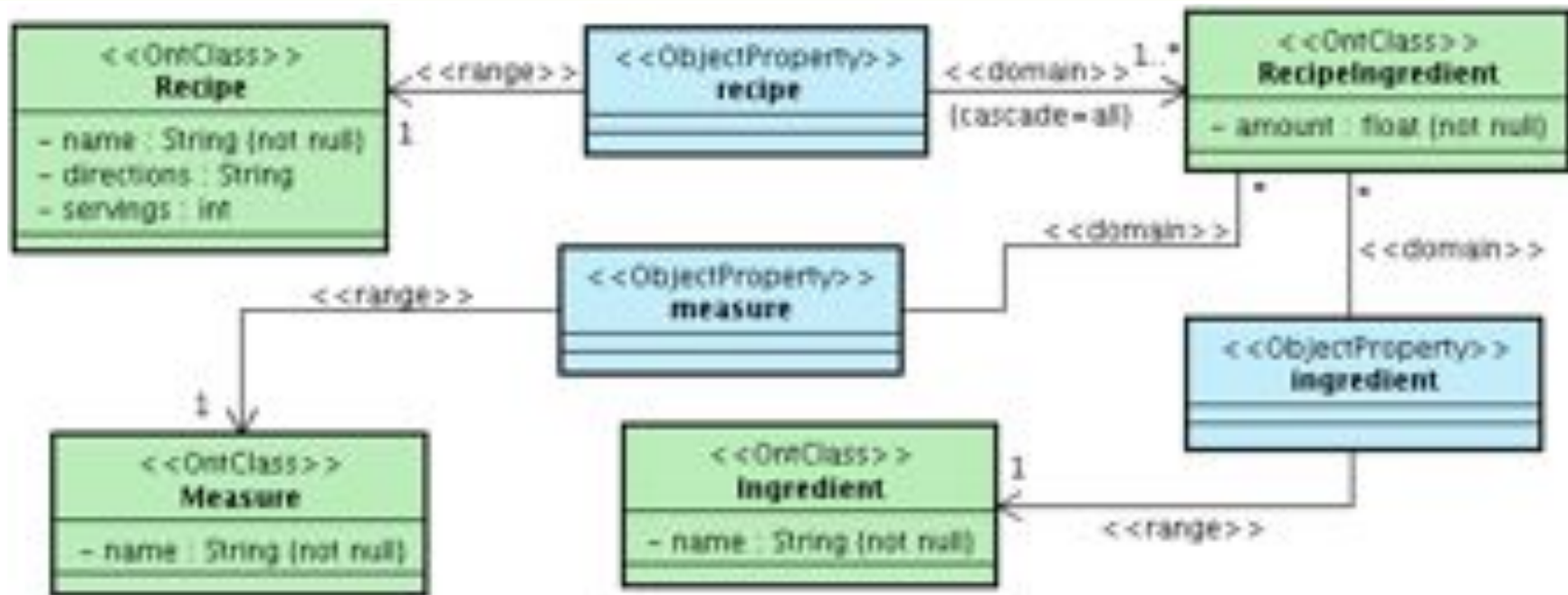
Source: <https://drupal.org/node/574624>

Recipe: using custom server-side code

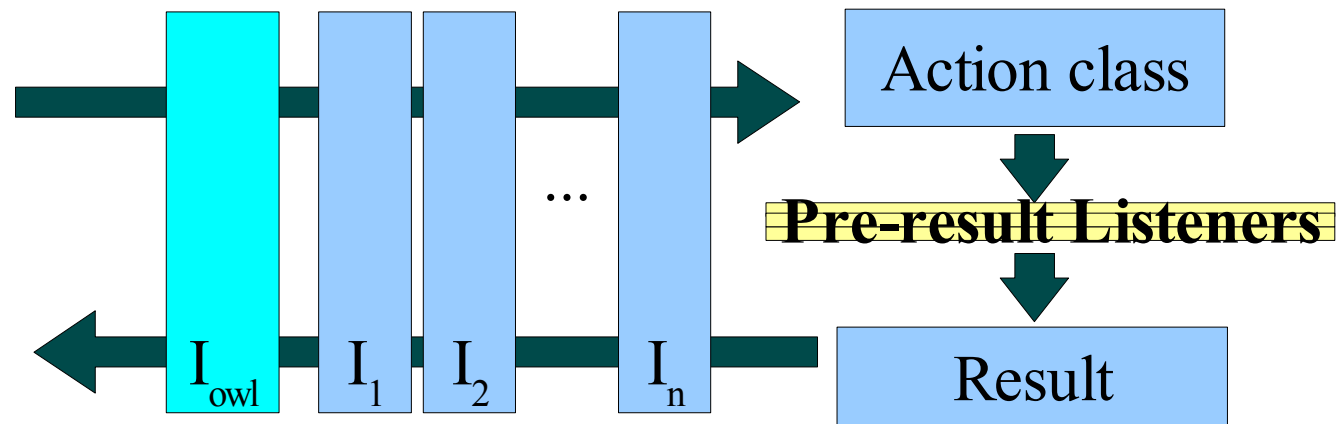
- Server (WebApp) decides when to serve HTML, RDF or do a 303 redirect (see slide 18);
- Clients can specify what they want in the “Accept:” HTTP header:

```
GET /people/matt-briggs HTTP/1.1  
Host: biglynx.co.uk  
Accept: text/html;q=0.5, application/rdf+xml
```

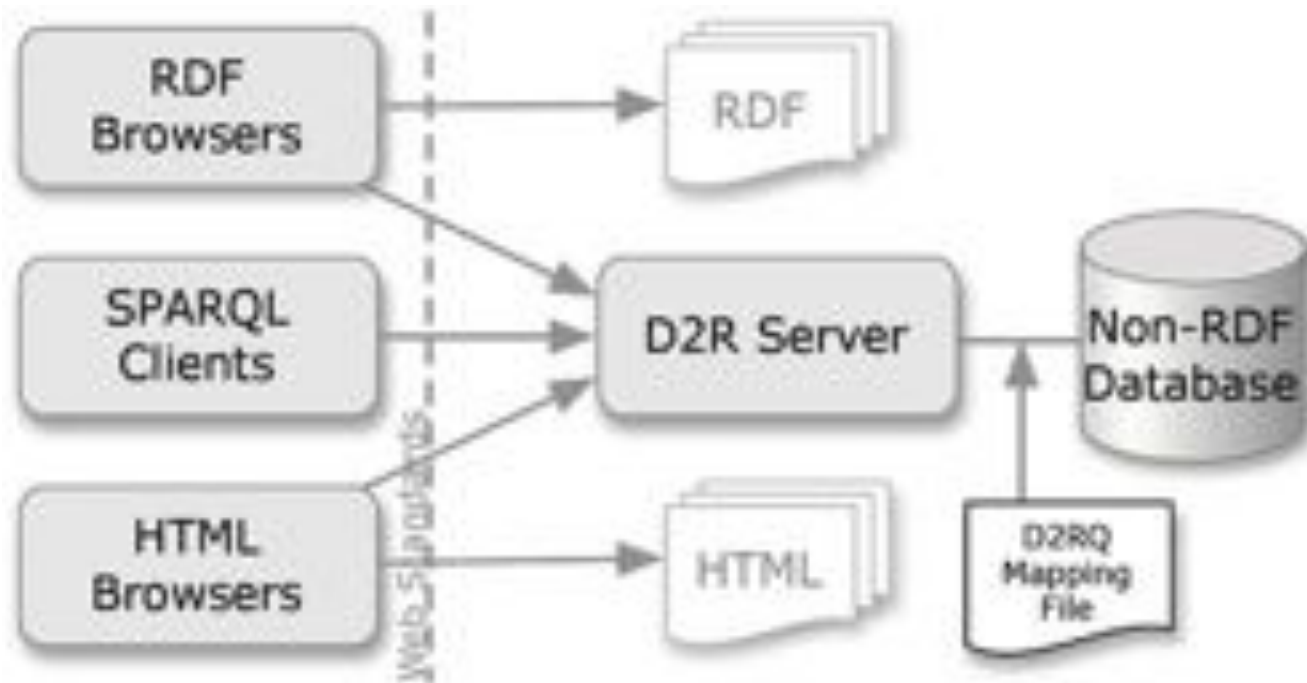





Client (Web
Browser)



- For legacy systems that you don't want to touch the source, you can do DB -> RDF/OWL directly:



See: <http://d2rq.org/d2r-server>

Alternatives:

- Virtuoso RDF Views ([wiki](#));
- Triplify ([.org](#)).

1. Download and install the server software;
2. Have D2R Server auto-generate a D2RQ mapping from your DB schema;
3. Customize the mapping by replacing auto-generated terms with well-known vocabularies;
4. Set RDF links pointing at external data sources;
5. Set RDF links from other data sets you own to this new data set so crawlers can find it;
6. Publicize your data source in other ways (as previously discussed)...

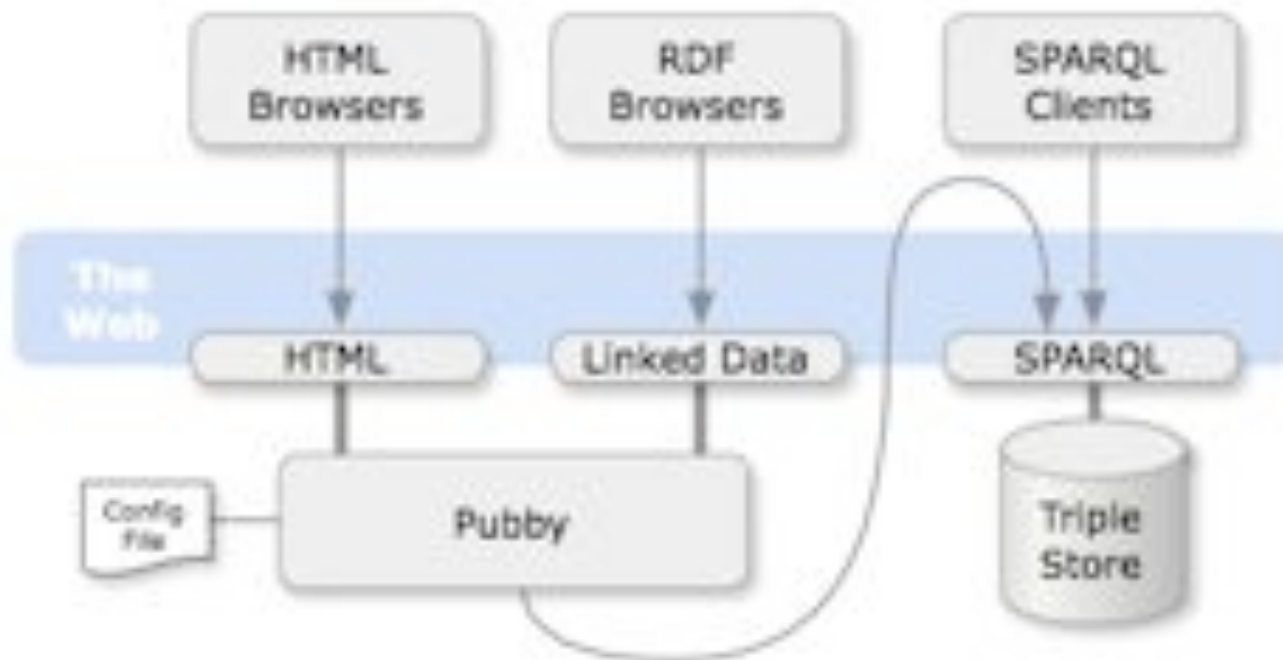
The W3C RDB2RDF Working Group is working on a standard language for this mapping.

D2R mapping example

```
map:Lugar a d2rq:ClassMap;
  d2rq:uriPattern "Lugar/@@Lugar.id@";
  d2rq:class lugar:Lugar;
  ...
map:Lugar_name a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:Lugar;
  d2rq:property rdfs:label;
  d2rq:propertyDefinitionLabel "Lugar name";
  d2rq:column "Lugar.name";
  ...
map:Lugar_country_code a d2rq:PropertyBridge;
  d2rq:belongsToClassMap map:Lugar;
  d2rq:property lugar:codigo_pais;
  d2rq:propertyDefinitionLabel "Lugar country_code";
  d2rq:column "geoname_raw.country_code";
  d2rq:join "Lugar.geoname_raw_id => geoname_raw.geonameid";
  ...
```

Recipe: from triplestores

- Usually, a Linked Data interface is provided;
- Otherwise, you can use Pubby, a Linked Data frontend for SPARQL endpoints:
 - <http://wifo5-03.informatik.uni-mannheim.de/pubby/>



- Sites like Amazon, Google, Twitter, Facebook, etc. expose data via Web APIs (programmableweb.com);
- Web APIs are not standard: heterogeneous query interfaces and formats (XML, JSON, Atom, ...);
- We can develop wrappers around them:
 1. Assign URIs to different resources;
 2. When one of these URIs is looked up, convert the request into an API-specific request;
 3. Convert the API-specific results to RDF and return.
- Make sure that: (a) you create adequate outlinks; (b) you have the rights to expose the data.

- At the end of the day, the primary means of publishing LD is to make URIs dereferenceable:
 - Allows the follow-your-nose style of data discovery.
- In addition, one could also provide:
 - RDF data set dumps for data replication;
 - SPARQL endpoints for querying.
- Then, applications can choose the method that suits them best.

- When publishing we should also check if data and infrastructure adhere to the LD best practices;
- Does the RDF data convey the intended info?
 - Serialize it as N-Triples and read it! Do they make sense?
- Is the document syntactically correct?
 - The W3C validator (w3.org/RDF/Validator/) can test this, plus provide a N-Triples visualization as a bonus!
 - Jena's [Eyeball](#) can perform a more in-depth analysis.

- Vapour (idi.fundacionctic.org/vapour): dereferences URIs and provides details on the HTTP lookup;
- RDF:Alerts (swse.deri.org/RDFAlerts/): dereferences not only URIs, but also vocab predicates, checking domain, range and data type restrictions;
- Sindice Inspector (inspector.sindice.com): visualize and validate RDF, HTML with microformats/RDFa. Performs reasoning and checks for common errors;
- 303 redirects can be verified with cURL (see [tutorial](#));
- Browser extensions that modify the HTTP header (e.g. [Live HTTP Headers](#) and [Modify Headers](#) for Firefox) can aid in testing a server's response.

- Can the data set be fully navigated (reach all resources, go back and forth, etc.) with an LD browser?
 - Tabulator (w3.org/2005/ajar/tab): test if RDF files are too large, consistency of inheritance relations;
 - Marbles (mes.github.io/marbles/): uses 2 second time-out, tests if response is too slow;
 - Check out other browsers at the LOD Browser Switch: <http://browse.semanticweb.org>.

1. Does your data set link to other data sets?
2. Do you provide provenance metadata?
3. Do you provide licensing metadata?
4. Do you use terms from widely deployed vocabularies?
5. Are the URIs of proprietary vocabulary terms dereferenceable?
6. Do you map proprietary vocabulary terms to other vocabularies?
7. Do you provide data set-level metadata?
8. Do you refer to additional access methods?

Check out: <http://lod-cloud.net/state/#best-practices>

Linked Data

CONSUMING LINKED DATA

- Now, it's part of the global data space (the WoD). It's time to consume it!
- Applications can exploit Linked Data properties:
 - Standardized data representation and access;
 - Openness of the Web of Data.

The application you build today may, in the future, consume data that is not published yet...

- Let's see some example applications:
 - Generic applications;
 - Domain-specific applications.
- For further example, refer to:
 - W3C SWEO (Semantic Web Education and Outreach) Interest Group applications page:
<http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData/Applications>
 - W3C SW Case Studies and use cases:
<http://www.w3.org/2001/sw/sweo/public/UseCases/>

- Can process data from any domain;
- Two basic types:
 - Linked Data browsers;
 - Linked Data search engines.

Linked Data browsers

- We just talked about them in slide 132!

The Tabulator



VisiNav



Zitgist

OPENLINK
SOFTWARE
Data Explorer

Disco

new
Graphite

new
EXPLORATOR

```
<rdf:RDF>
  <!-- Ontology head
  - <owl:Ontology rdf:al
    <swivt:creationDat
    <owl:imports rdf:r
  </owl:Ontology>
```

triplr
ntriples

triplr
turtle

triplr
html

triplr
json

triplr
rdf

- Can navigate between data sets following URIs like a Web browser follows links to navigate documents;
- Try the Graphite (.ecs.soton.ac.uk/browser/) browser:
 - Look up `http://dbpedia.org/resource/Charlottesville,_Virginia`;
 - Follow `dbo:hometown`
`dbpedia:Dave_Matthews_Band`;
 - Follow `dbo:artist`
`dbpedia:Crash_(Dave_Matthews_Band_album)`;
 - Find out the songs of this album.

- Already talked about them too, slide 100!
- They crawl LD from the WoD, following RDF links;
- Provide query capabilities over aggregated data;
- Integrate data from thousands of data sources.



Linked Open Vocabularies (LOV)



- LD engines are richer than normal search engines:
 - Filter search result by class of resource;
 - Summary views;
 - Choose which data sets to consider;
 - Etc.

“Give me the URL of all blog who are written by people that Tim Berners-Lee knows”

Linked Data search engine:

- A person resource;
- Another person resource;
- Etc.

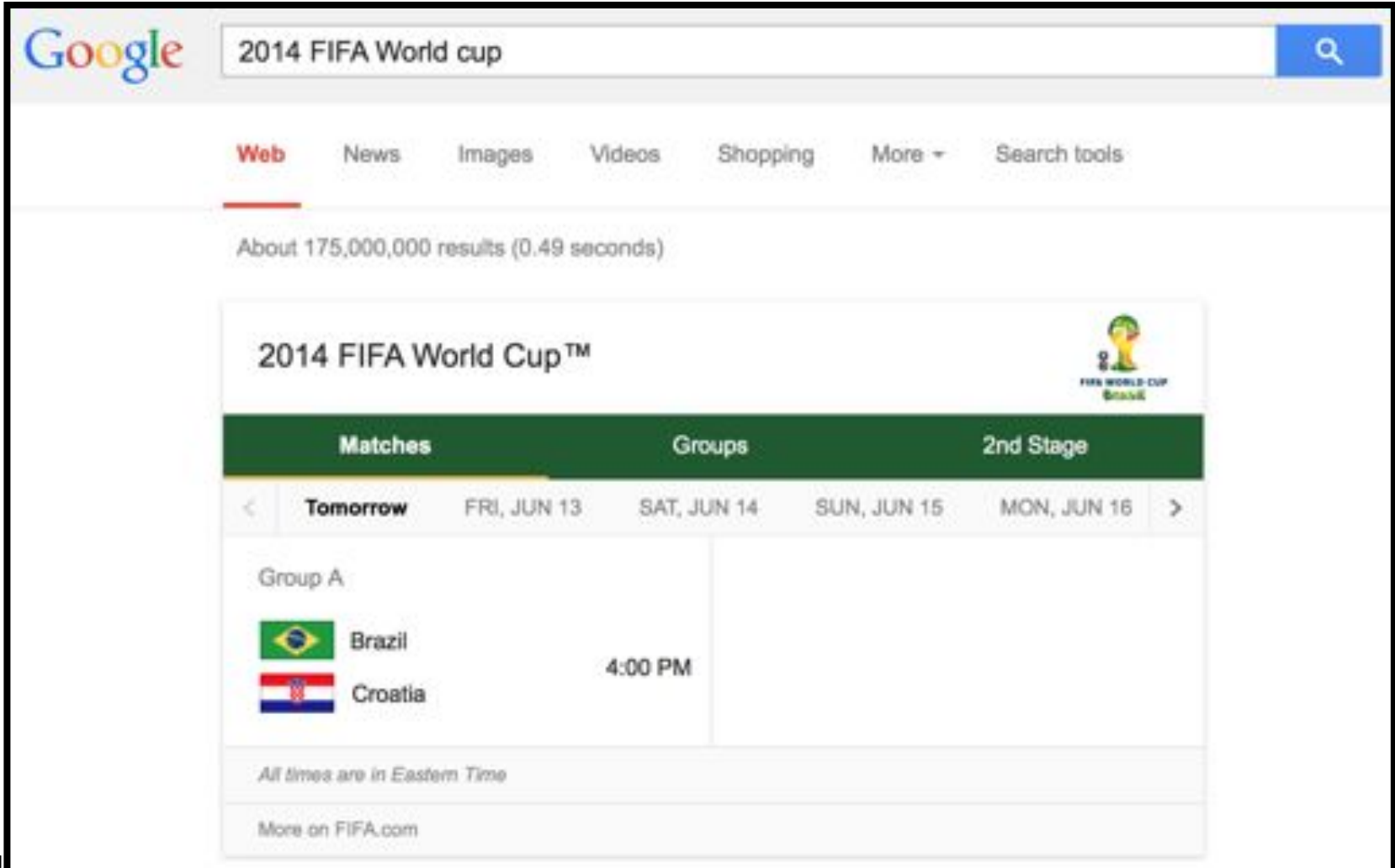
X

“Vanilla” search engine:



- Some page about TBL;
- Some other page about TBL;
- Etc.

Traditional search engines also use LD

- Google provides “rich snippets” for people, organizations, products, recipes, events, music, etc.



Google search results for "2014 FIFA World Cup". The search bar shows the query and a magnifying glass icon. Below the search bar are tabs for Web, News, Images, Videos, Shopping, More, and Search tools. The results show "About 175,000,000 results (0.49 seconds)". The main result is for the "2014 FIFA World Cup™" with a trophy icon. Below this is a table showing match details for Group A.

Matches		Groups	2nd Stage
<	Tomorrow	FRI, JUN 13	SAT, JUN 14
Group A			
	Brazil		
	Croatia	4:00 PM	

All times are in Eastern Time

More on FIFA.com

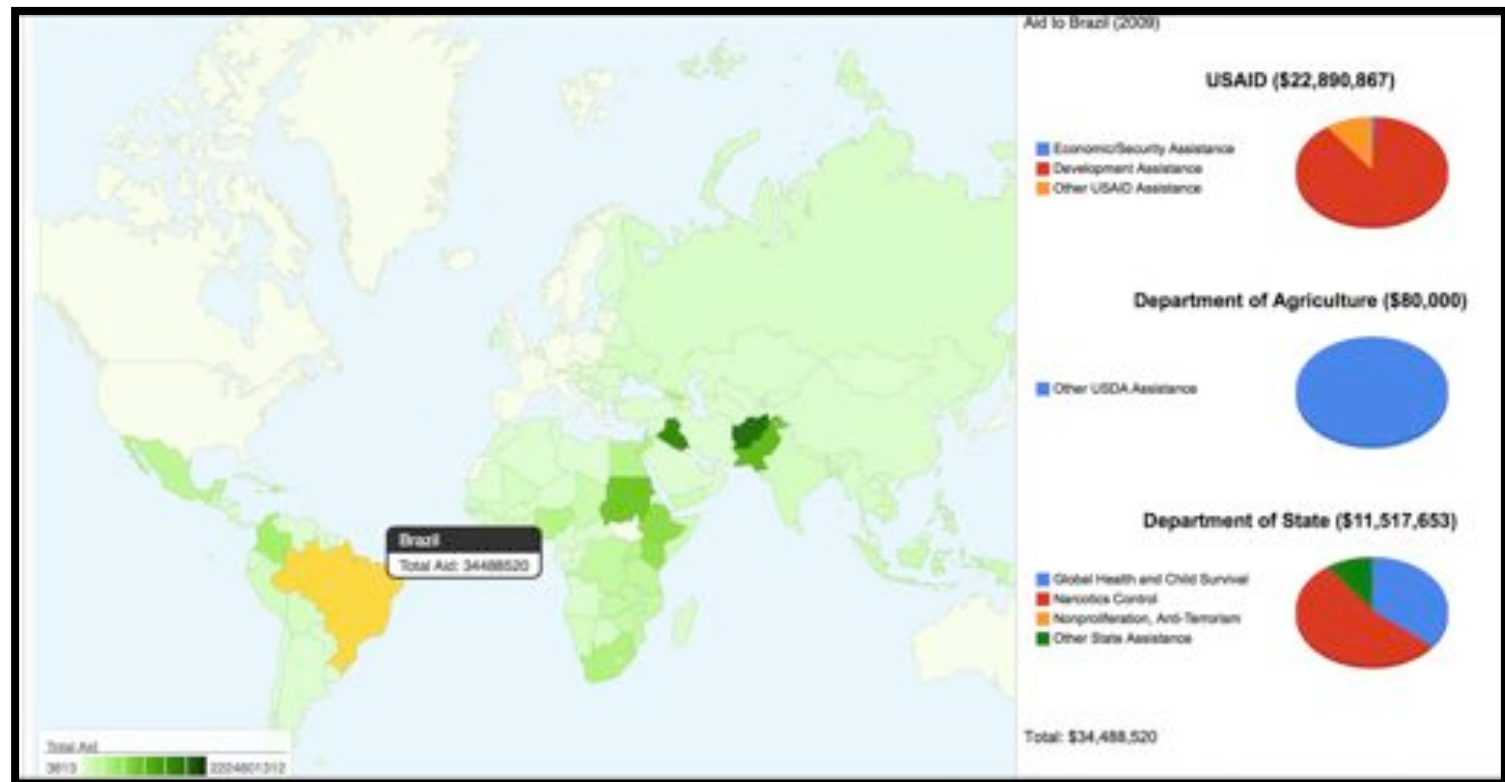
Traditional search engines also use LD

- Google provides “rich snippets” for people, organizations, products, recipes, events, music, etc.



The screenshot shows a Google search interface. The search bar contains the text "what is the birth date of barack obama". Below the search bar, the "Web" tab is selected. The search results show "About 1,360,000 results (0.59 seconds)". The main result is a rich snippet for Barack Obama's birth date, "August 4, 1961 (age 52 years)", with the subtext "Barack Obama, Date of birth". To the right of this snippet is a portrait of Barack Obama. Below the main snippet, there are three smaller snippets for "George Washington" (February 22, 1732), "Vladimir Putin" (October 7, 1952), and "Abraham Lincoln" (February 12, 1809). On the right side of the search results, there is a section titled "Barack Obama" with "4,340,780 followers on Google+", a "Follow" button, and a brief biography: "Barack Hussein Obama II is the 44th and States, and the first African American to ho". Below the biography, it lists: "Born: August 4, 1961 (age 52), Honolulu," "Full name: Barack Hussein Obama II", "Nationality: American", "Spouse: Michelle Obama (m. 1992)", "Education: Harvard Law School (1988–19)", and "Parents: Ann Dunham, Barack Obama Sr."

- U.S. Global Foreign Aid Mashup: <http://data-gov.tw.rpi.edu/demo/USForeignAid/demo-1554.html>;
 - Pulls spending data from USAID, Dep. of Agriculture and Dep. of State:

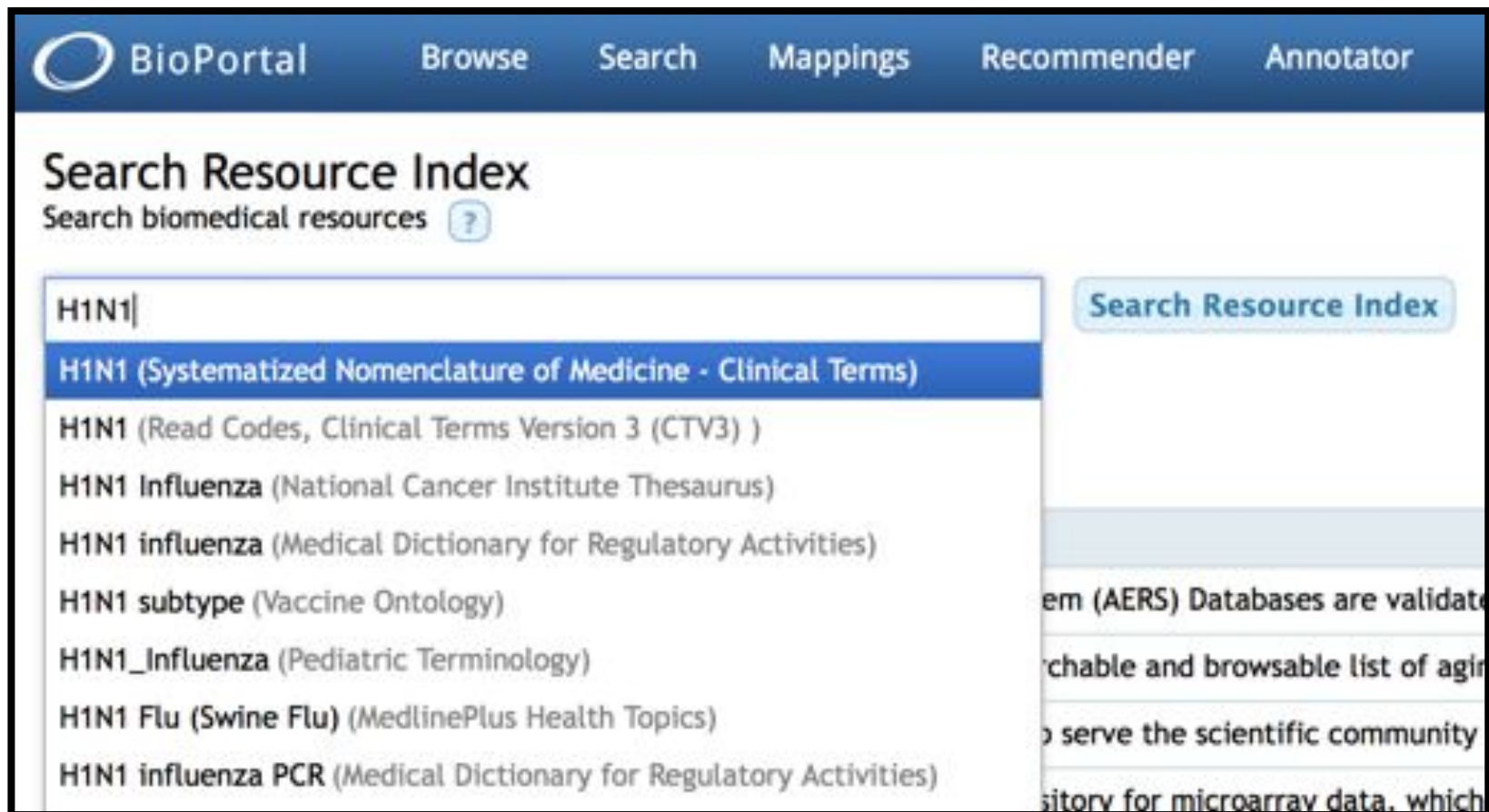


- DBPedia Mobile (wiki.dbpedia.org/DBpediaMobile):
 - Helps tourists explore a city;
 - Location-centric mashup of nearby locations;
 - Allows users to publish check-ins, pictures and reviews as Linked Data.

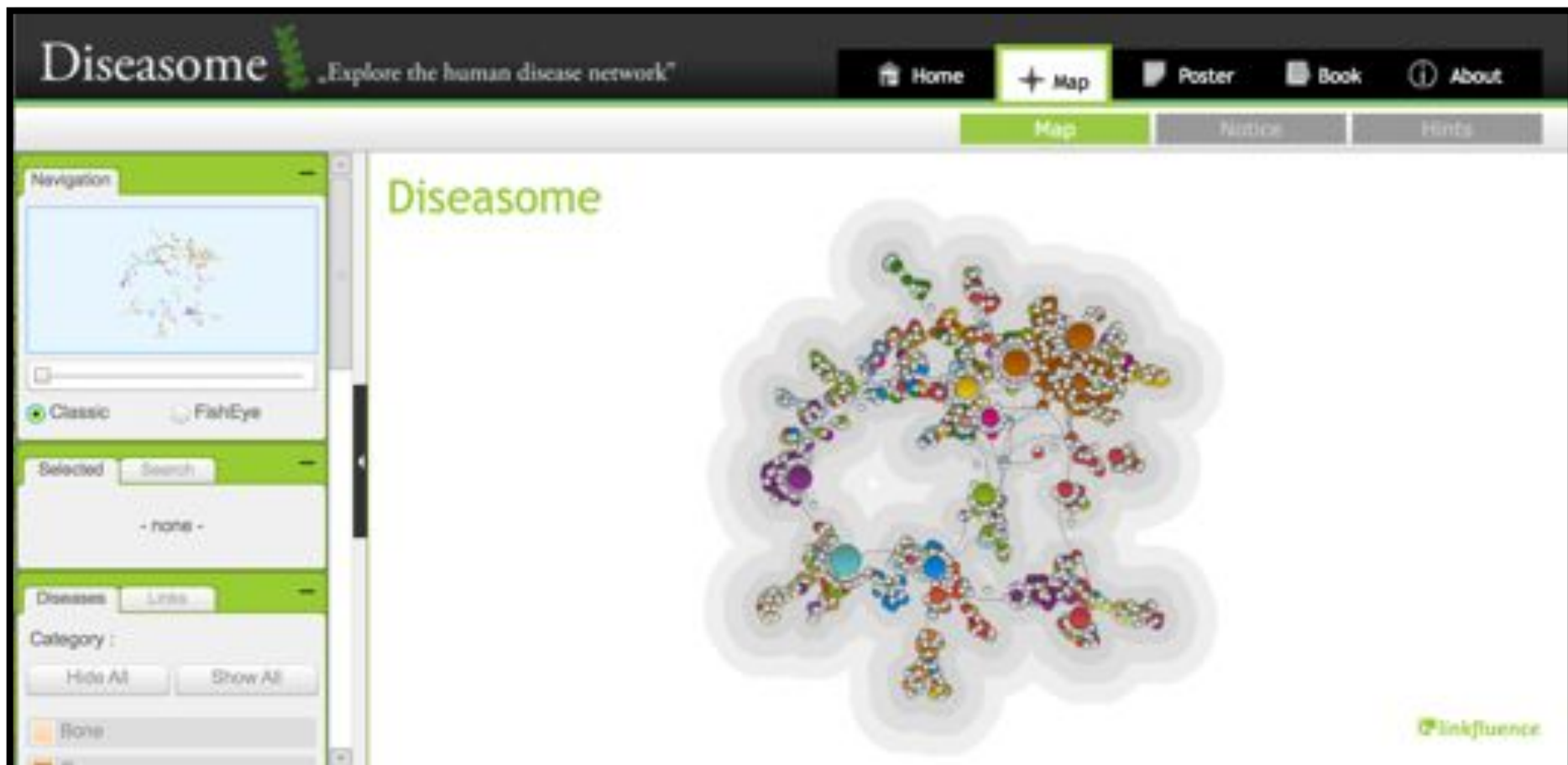


Domain-specific LD applications

- BioPortal: <http://bioportal.bioontology.org>;
- Life sciences application that lets users explore biomedical resources:



- Diseasome: <http://diseasome.eu>;
- Combines various life science sources to generate “a network of disorders and disease genes”:



Domain-specific LD applications

- Faviki: <http://www.faviki.com>;
- Social bookmarking that lets you use Wikipedia concepts as tags:



- Example from the book: augment the Big Lynx website with WoD data about the places where employees live;
 1. Discover data sources with info about the cities;
 2. Download the data and store it in a local RDF store along with provenance meta-data;
 3. Retrieve information to be displayed using SPARQL.

- LDSpider:
 - <https://code.google.com/p/ldspider/>;
 - Crawls the WoD and store triples locally;
- Jena TDB:
 - <http://openjena.org/TDB>;
 - A database of triples (triplestore);
- SPARQL/Update:
 - <http://www.w3.org/TR/sparql11-update/>;
 - Update language for RDF graphs.

```
java -jar ldspider.jar  
-u "http://dbpedia.org/resource/Birmingham"  
-b 5 10000  
-follow "http://www.w3.org/2002/07/owl/sameAs"  
-oe "http://localhost:2020/update/service"
```

- -u: where to look;
- -b: crawling limit;
- -follow: type of predicate to follow;
- -oe: where to store the triples (SPARQL/Update endpoint).

- To store provenance meta-data, name the graphs that are retrieved by the crawler;
- Then, write triples with the URI of the graph as subject to indicate author, retrieval date, etc.;
- The TriG syntax (www4.wiwiss.fu-berlin.de/bizer/TriG/) extends Turtle to support Named Graphs:

```
<http://localhost/myGraphNumberOne>
{
  biz:JoesPlace rdfs:label "Joe's Noodle Place"@en .
  biz:JoesPlace rev:rating rev:excellent .

  <http://localhost/myGraphNumberOne> dc:creator <http://
www4.wiwiss.fu-berlin.de/is-group/resource/persons/Person4> .
  <http://localhost/myGraphNumberOne> dc:date "2010-12-17"^^xsd:date .
}
```

Result of LDSpider crawl in TriG

```
<http://dbpedia.org/data/Birmingham.xml> {  
  dbpedia:Birmingham rdfs:label "Birmingham"@en .  
  dbpedia:Birmingham rdf:type dbpedia-ont:City .  
  dbpedia:Birmingham dbpedia-ont:thumbnail  
    <http://.../200px-Birmingham_-UK_-Skyline.jpg> .  
  dbpedia:Birmingham dbpedia-ont:elevation "140"^^xsd:double .  
  dbpedia:Birmingham owl:sameAs  
    <http://data.nytimes.com/N35531941558043900331> .  
  dbpedia:Birmingham owl:sameAs <http://sws.geonames.org/3333125/> .  
  dbpedia:Birmingham owl:sameAs  
    <http://rdf.freebase.com/ns/guid.9202...f8000000088c75> .  
}
```

```
<http://sws.geonames.org/3333125/about.rdf> {  
  <http://sws.geonames.org/3333125/> gnames:name  
    "City and Borough of Birmingham" .  
  <http://sws.geonames.org/3333125/> rdf:type gnames:Feature .  
  <http://sws.geonames.org/3333125/> geo:long "-1.89823" .  
  <http://sws.geonames.org/3333125/> geo:lat "52.48048" .  
  <http://sws.geonames.org/3333125/> owl:sameAs  
    <http://www.ordnancesurvey.co.uk/...#birmingham_00cn> .  
}
```

- SPARQL is implemented in most RDF stores;
- Can query single graphs or named graphs:

```
SELECT DISTINCT ?p ?o ?g WHERE {  
  { GRAPH ?g { <http://dbpedia.org/resource/Birmingham> ?p ?o . } }  
  UNION  
  { GRAPH ?g1 { <http://dbpedia.org/resource/Birmingham>  
    <http://www.w3.org/2002/07/owl#sameAs> ?y }  
    GRAPH ?g { ?y ?p ?o }  
  }  
}
```

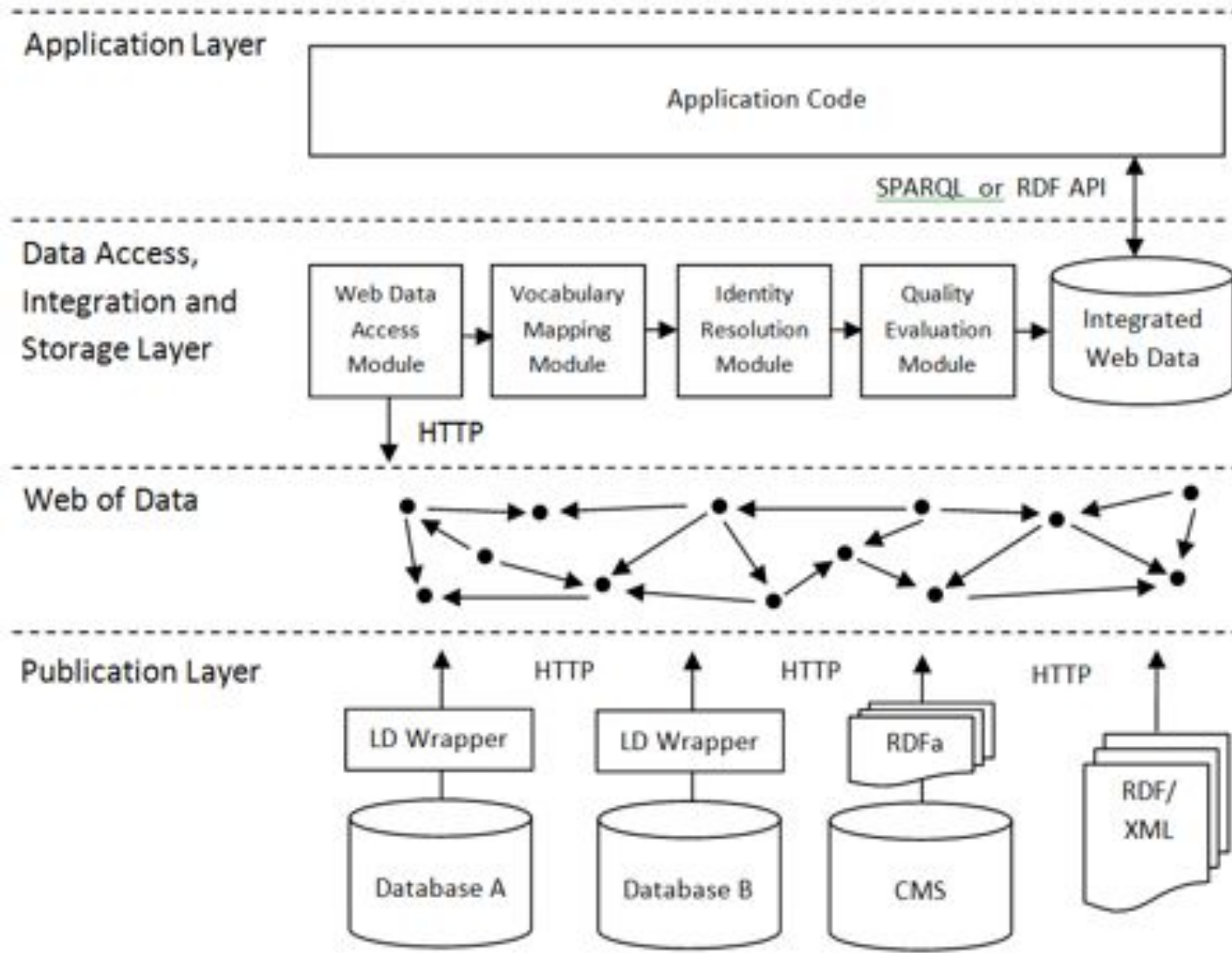
- The small example from before can already highlight some architectural patterns;
- The Crawling Pattern:
 - Crawl the WoD in advance;
 - Integrate/clean the data afterwards;
 - Executes complex queries with reasonable performance over large amounts of data;
 - Disadvantage: data is replicated, may be stale, should re-run the crawler often;

- The On-The-Fly Dereferencing Pattern:
 - Opposite of the crawling pattern;
 - Used by LD browsers;
 - Data is always up-to-date, but complex queries are slow;
- The Query Federation Pattern:
 - Send complex queries to a fixed set of data sources (instead of the entire Web of Data);
 - Can be used if sources provide SPARQL endpoints;
 - JOINS over a large number of data sets is very complex. Works better if this number is small.

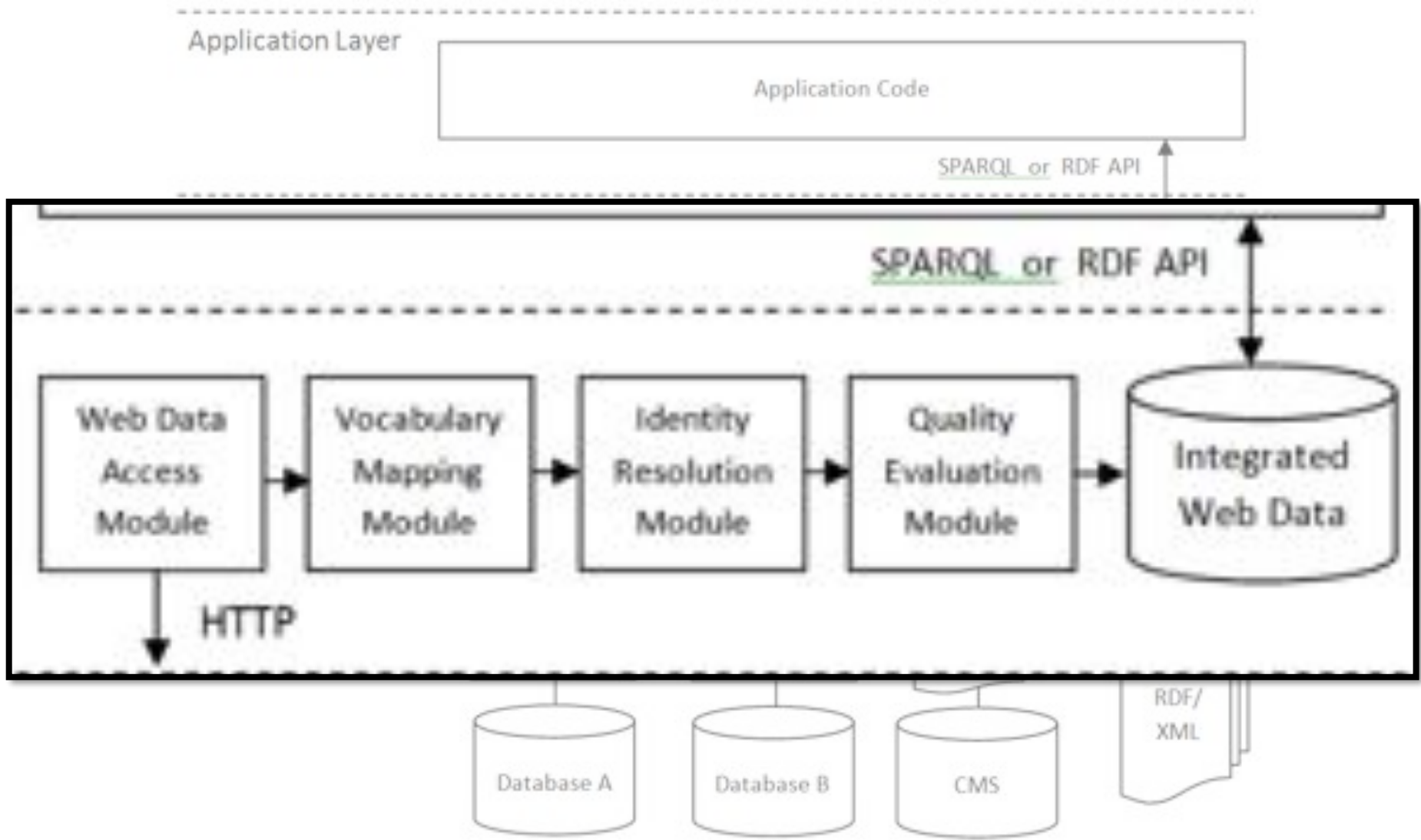
- The decision depends on:
 1. The number of data sources the application intends to use;
 2. The degree of data freshness required;
 3. The required response times for interactions;
 4. The extent to which the application aims to discover new data sources at runtime.

In most cases, probably crawling +
local RDF store are best.

Crawling architectural pattern overview



Crawling architectural pattern overview



1. Web data access: dereferencing URIs or performing SPARQL queries;
2. Vocabulary mapping: translate different vocabulary to a single target schema using vocabulary links;
3. ID resolution: follow sameAs links to resolve identity or use heuristics when links are not present;
4. Provenance tracking: for data cached locally it's important to keep track of where it came from;
5. Data quality assessment: the WoD is open, so treat data as claims (possibly spam!), not facts;
6. Data use: query, retrieve, format for end users...

- LD crawlers to crawl. E.g.: LDSpider;
- LD client libraries to dereference on-the-fly. E.g.: the [SW Client Library](#);
- SPARQL client libraries: to query endpoints. E.g.: [Jena](#);
- Federated SPARQL engines: provide support to [federated queries](#). E.g., [DARQ](#), [SemaPlorer](#), Jena;
- RDFa Tools: to extract from HTML. See rdfa.info/tools/;
- Search engine APIs: to access the data they have crawled. E.g.: [Sindice](#), [sameAs.org](#).

2 – Vocabulary mapping

- Exploit owl:equivalentClass, owl:equivalentProperty and rdfs:subClassOf, rdfs:subPropertyOf;
- More complex transformations (merging resources, splitting properties, normalizing) not in RDFS/OWL;
- Some proposals:
 - [SPARQL++](#);
 - [The Alignment API](#);
 - [Rules Interchange Format \(RIF\)](#);
 - [R2R Framework](#);
 - [RDF Refine](#) (based on [Google Refine](#)).

- Silk Server:
 - Receives RDF instances (e.g., from a crawler);
 - Matches them against a local set of known instances;
 - Discovers duplicate instances;
- Can be used to store only new information:
 - Add resources that are totally new;
 - Updating information on existing resources.

4 – Provenance tracking

- Use Named Graphs, TriG and SPARQL as in the Big Lynx example:

```
<http://dbpedia.org/data/Birmingham.xml> {  
  dbpedia:Birmingham rdfs:label "Birmingham"@en .  
  dbpedia:Birmingham rdf:type dbpedia-ont:City .  
  dbpedia:Birmingham dbpedia-ont:thumbnail  
    <http://.../200px-Birmingham_UK_Skyline.jpg> .  
  dbpedia:Birmingham dbpedia-ont:elevation "140"^^xsd:double .  
  dbpedia:Birmingham owl:sameAs  
    <http://data.nytimes.com/N35531941558043900331> .  
  dbpedia:Birmingham owl:sameAs <http://sws.geonames.org/3333125/> .  
  dbpedia:Birmingham owl:sameAs  
    <http://rdf.freebase.com/ns/guid.9202...f8000000088c75> .  
}  
  
<http://sws.geonames.org/3333125/about.rdf> {  
  <http://sws.geonames.org/3333125/> gnames:name  
    "City and Borough of Birmingham" .  
  <http://sws.geonames.org/3333125/> rdf:type gnames:Feature .  
  <http://sws.geonames.org/3333125/> geo:long "-1.89823" .  
  ...  
}
```

- Some quality assessment heuristics:
 - Content-based: use the information itself as indicator (e.g., outlier detection, SPAM filters, etc.);
 - Context-based: use meta-info (owner, date, etc.) as indicator (e.g., prefer data from manufacturer about a product, ignore data about its competitors);
 - Ratings-based: explicit ratings as indicators (e.g., Sig.ma rates the data they crawled);

- Handling conflicts:
 - Rank data: display all data, but rank them from most to least reliable (search engines do this, page rank);
 - Filter data: display only data that passes a minimum quality requirement (e.g., use the [WIQA framework](#));
 - Fuse data: integrate multiple data items that represent the same resource into one:
 - Consistent, clean, single representation;
 - Challenge is the resolution of data conflicts, which is an active research problem in the DB community.

- A MediaWiki project wiki page proposes a set of criteria to assess the quality of LD sources:

[http://sourceforge.net/apps/mediawiki/trdf/index.php?title=Quality Criteria for Linked Data sources](http://sourceforge.net/apps/mediawiki/trdf/index.php?title=Quality%20Criteria%20for%20Linked%20Data%20sources)

- Tim Berners-Lee “Oh, yeah?” button:

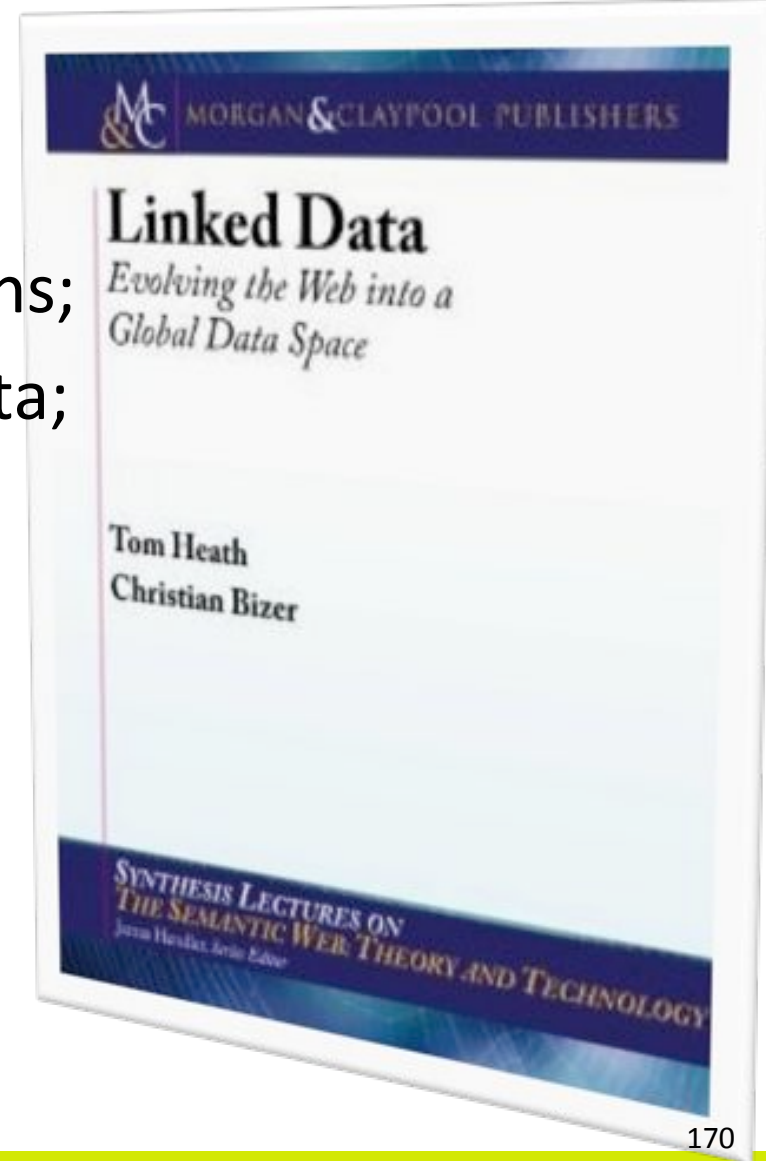
At the toolbar (menu, whatever) associated with a document there is a button marked "Oh, yeah?". You press it when you loses that feeling of trust. It says to the Web, "so how do I know I can trust this information?". The software then goes directly or indirectly back to meta-information about the document, which suggests a number of reasons.

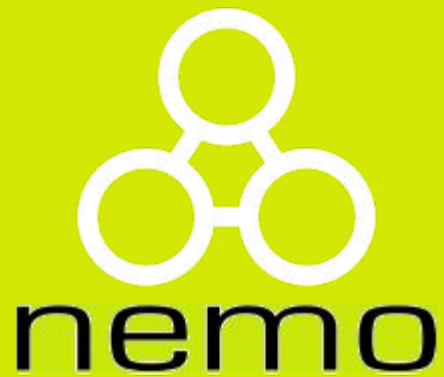
Source: <http://www.w3.org/DesignIssues/UI.html>

- Cache data locally (for better performance):
 - Use a triplestore (cf. slide 114);
 - For performance evaluation, check out the [Berlin SPARQL benchmark](#) or the [W3C one](#);
- Use RDF tools:
 - There's a list in the [W3C SW Wiki](#);
 - An alternative list is [Sweet Tools](#);
 - Widgets for visualizing Web data are provided by the [Information Workbench](#) tool.

- With LD, data providers can help data consumers:
 - Reusing terms from widely used vocabularies;
 - Publishing mappings between terms;
 - Setting RDF links to related resources...
- The WoD promotes dividing the integration effort in an open environment at the cost of quality uncertainty;
- Currently there's still a lot of heterogeneity, but over time more mappings will be provided and applications will generate better query answers.

- Introduction;
- Principles of linked data;
- The Web of Data;
- Linked data design considerations;
- Recipes for publishing linked data;
- Consuming linked data.





[**http://nemo.inf.ufes.br/**](http://nemo.inf.ufes.br/)