Université Joseph Fourier
GRENOBLE
UFR IM²AG

WEB DES DONNÉES
DONNÉES LIÉES (LINKED DATA)

une introduction

Philippe GENOUD – Danielle ZIEBELIN - LIG-Steamer
Prenom.Nom@imag.fr

- Certaines « diapositives » de ce cours sont issues de différentes présentations:

  - Web des données*: Données Ouvertes et Liées* – Ph. GENOUD
    GdR MAGIS – Ecole de Géomatique - 29 septembre au 3 Octobre 2014 – Sète

  - Web des données: *Les Principes- Les Standards du W3C* – Ph. GENOUD
    Journée Interopérabilité et Innovation – IGN-BRGM-OGC - 7 Octobre 2014 - Paris

  - What Is This Thing Called Linked Data ? – Ph. GENOUD, M. ATENCIA, J. DAVID
    ACM-DocEng2015 Tutorial – 9 Septembre 2015 – Lausanne

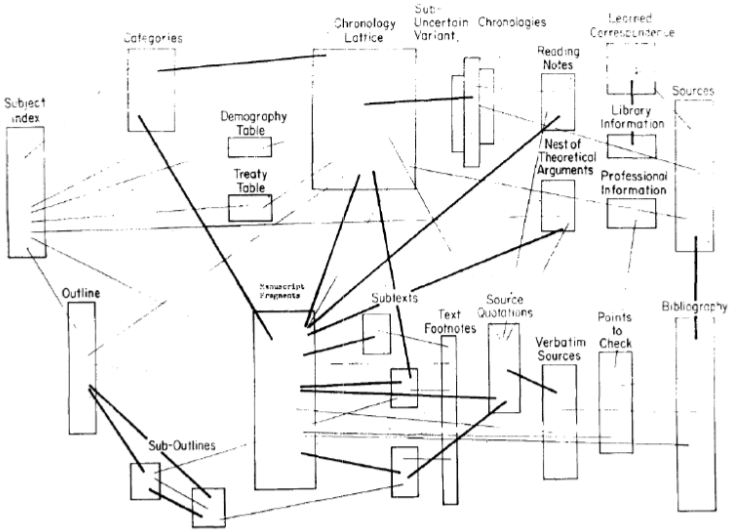
  - Cours Web Semantique et Ontologies : M2P Génie Informatique UFR IM2AG - UJF

WEB Foundations*

1. Link together digital documents

Ted Nelson

Hypertexte - Hypermedia

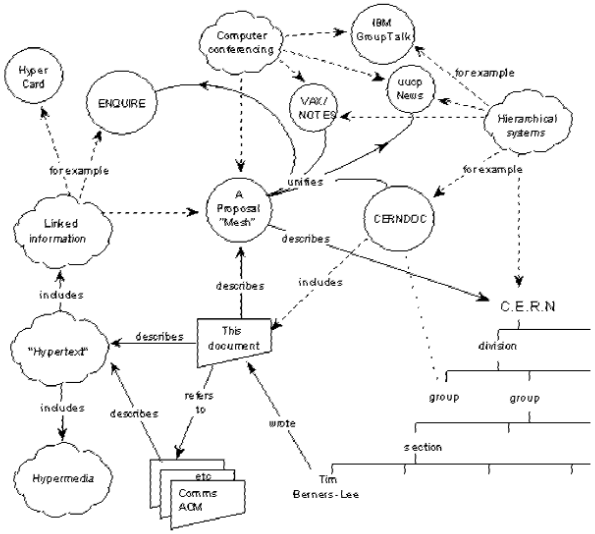


Complex information processing: a file structure for the complex, the changing and the indeterminate, T. H. Nelson, ACM, 1965

2. Link through the network

Tim Berners-Lee

Identify and link over the internet



Information Management: A Proposal
Tim Berners-Lee, CERN, March 1989, May 1990

*Slide Inspired from MOOC : Web sémantique et Web de données – Fabien Gandon – FUN 2015

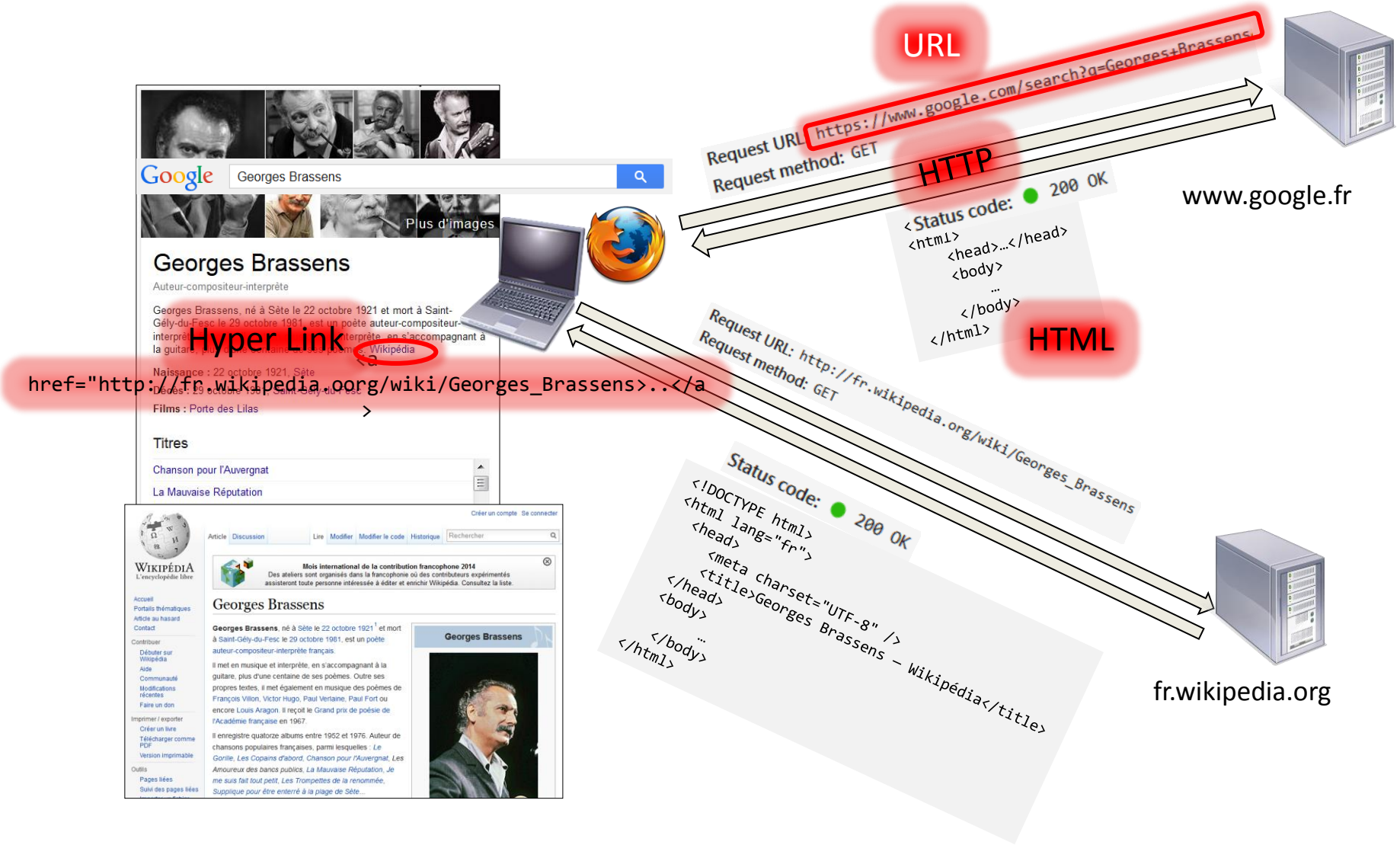
The WEB: a Success-Story

- WEB : a single global information space combining simplicity with decentralization and openness,

- Success of the Web is based on an distributed architecture built on a small set of simple **standards**

  - a globally unique identification mechanism: Uniform Resource Identifiers (URIs)
  - universal access mechanism: Hypertext Transfer Protocol (HTTP) protocol
  - a widely used content format : Hypertext Markup Language (HTML)
  - possibility of setting hyperlinks between Web documents that may reside on different Web servers

WEB architecture



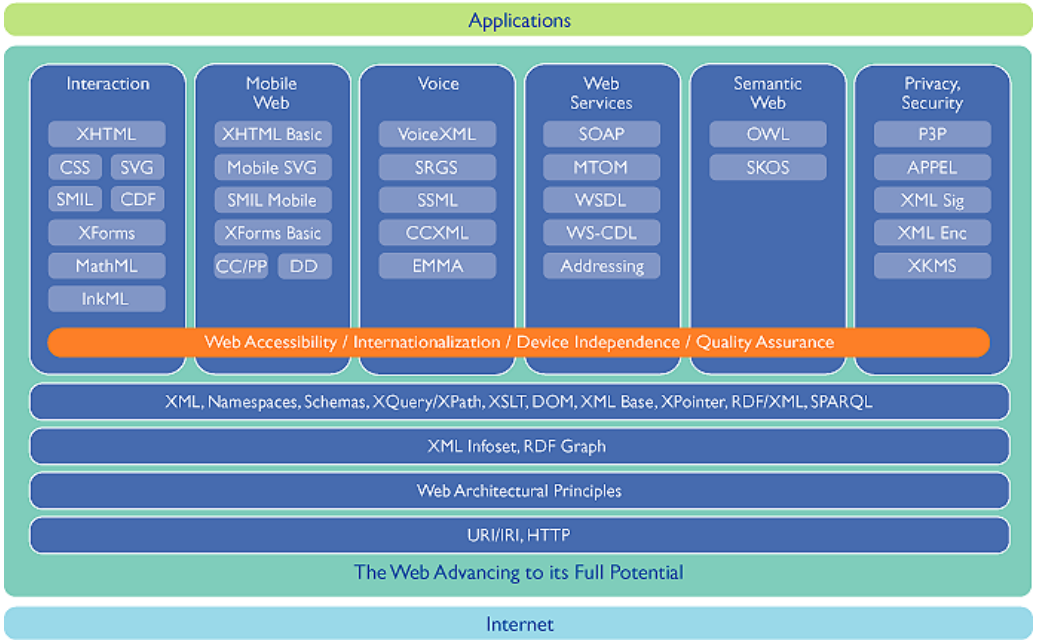
Web standardization

- 1994 creation of the W3C
- International consortium, around 400 Members
- Standardization of core Web technologies, publication of guidelines, technical notes, etc.

Based on a transparent and public Process

http://www.w3.org/2013/dd-epasorg.htm#%288%29

Standardization activities (2008)



Web evolution

Web 1.0

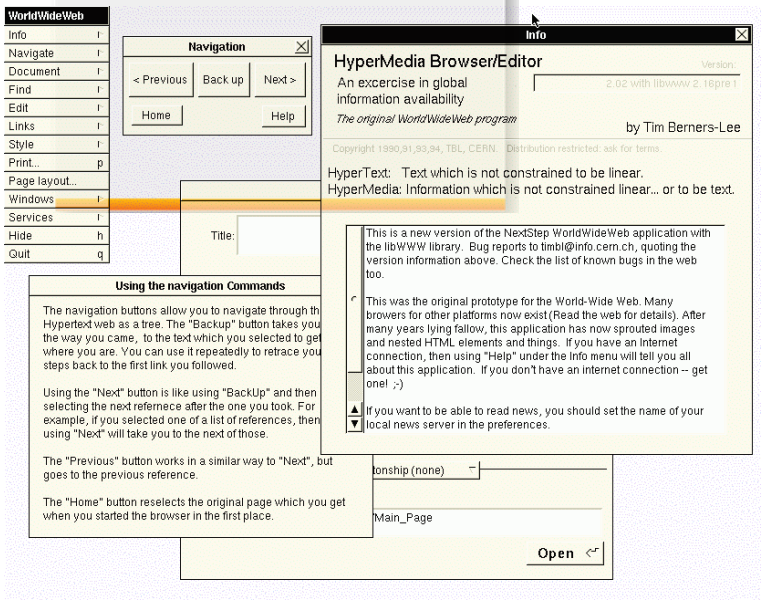
Web 2.0

Web 3.0

Web evolution

Web 1.0 *

- Utilisateur passif
- Essentiellement pro
- Réservé aux experts
- Guerre des navigateurs
- Contenu statique
- Formulaires
- Recherche par mots-clés



Web 2.0 *

Web 3.0

*d'après le cours Technologie du Web – A. Hombiat - Licence Professionnelle Études Statistiques et Systèmes d'Information Géographique (ESSIG) – UPMF - 2015

Web evolution

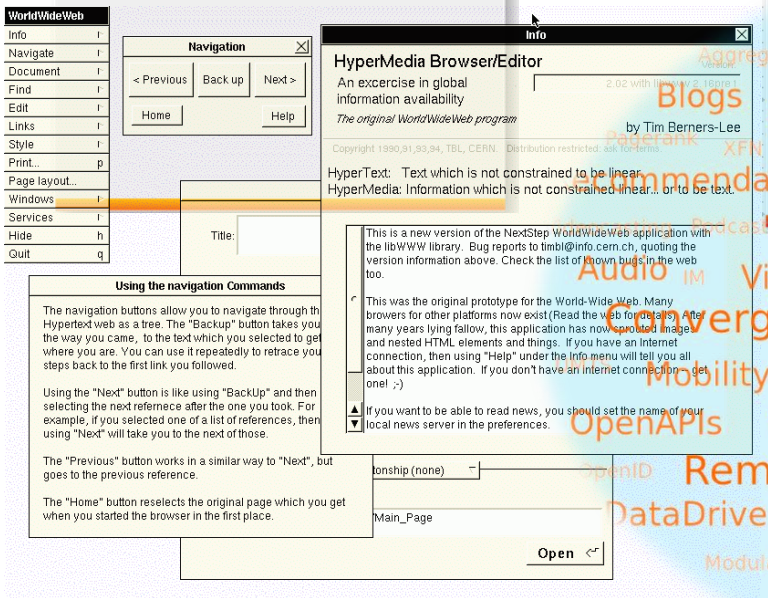
Web 1.0 *

- Utilisateur passif
- Essentiellement pro
- Réservé aux experts
- Guerre des navigateurs
- Contenu statique
- Formulaires
- Recherche par mots-clés

Web 2.0 *

- Utilisateur acteur (Blogs, Wikis, Réseaux sociaux, Podcasts, etc.)
- Ouvert aux profanes (CMS)
- Syndication du contenu (RSS, Atom)
- Tags et folksonomies
- Contenu dynamique
- Collaboration et partage
- Recherche par mots-clés

Web 3.0



*d'après le cours Technologie du Web – A. Hombiat - Licence Professionnelle Études Statistiques et Systèmes d'Information Géographique (ESSIG) – UPMF - 2015

Web evolution

Web 1.0 *

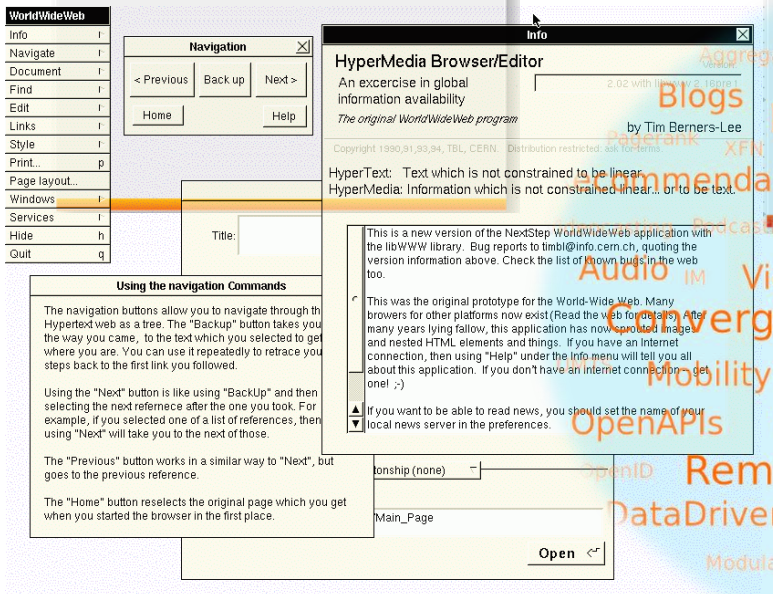
- Utilisateur passif
- Essentiellement pro
- Réservé aux experts
- Guerre des navigateurs
- Contenu statique
- Formulaires
- Recherche par mots-clés

Web 2.0 *

- Utilisateur acteur (Blogs, Wikis, Réseaux sociaux, Podcasts, etc.)
- Ouvert aux profanes (CMS)
- Syndication du contenu (RSS, Atom)
- Tags et folksonomies
- Contenu dynamique
- Collaboration et partage
- Recherche par mots-clés

Web 3.0

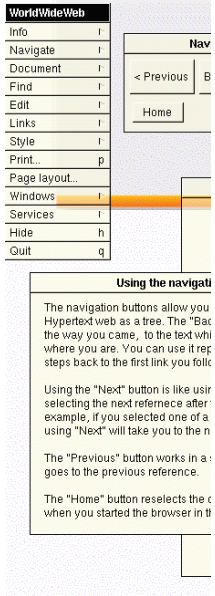
?

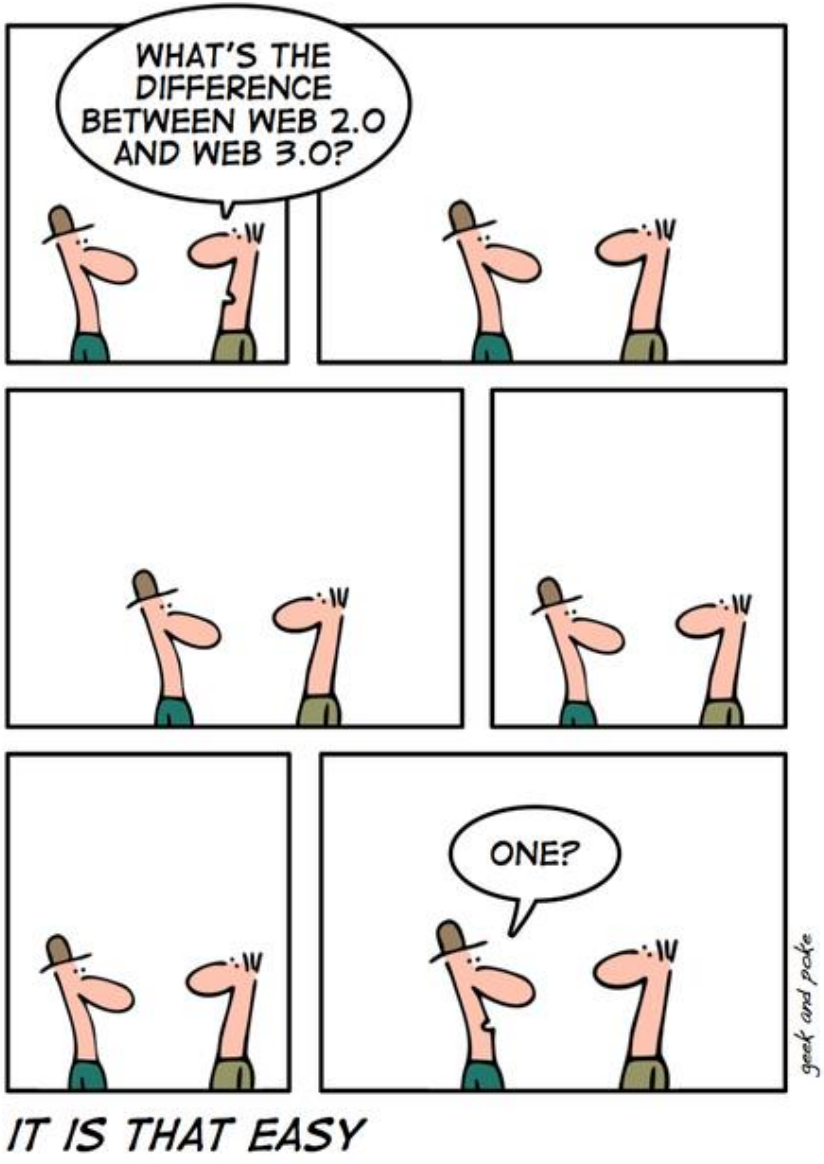


*d'après le cours Technologie du Web – A. Hombiat - Licence Professionnelle Études Statistiques et Systèmes d'Information Géographique (ESSIG) – UPMF - 2015

Web evolution

- Utilisat
- Essenti
- Réserv
- Guerre
- Conten
- Formul
- Recher





ogs,
ux,
enu
tage
clés

*

Web 3.0

?

*d'après le cou

Web evolution

Web 1.0 *

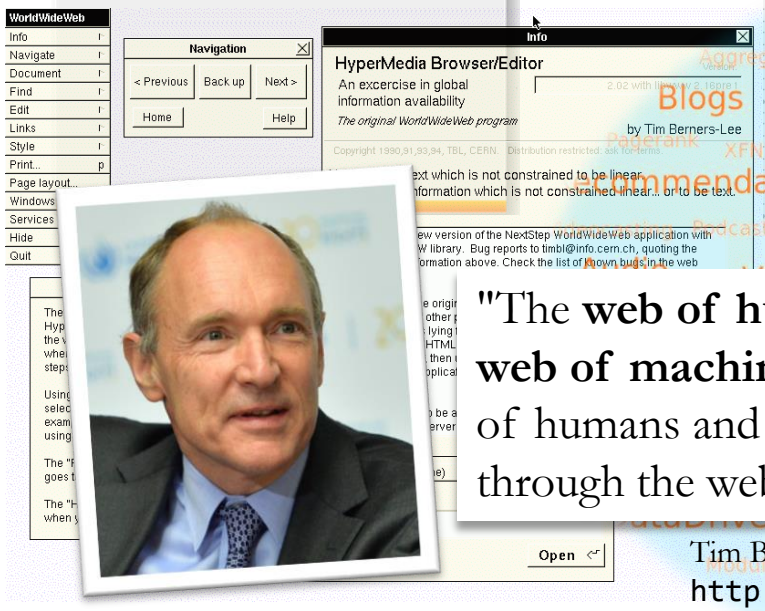
- Utilisateur passif
- Essentiellement pro
- Réservé aux experts
- Guerre des navigateurs
- Contenu statique
- Formulaires
- Recherche par mots-clés

Web 2.0 *

- Utilisateur acteur (Blogs, Wikis, Réseaux sociaux, Podcasts, etc.)
- Ouvert aux profanes (CMS)
- Syndication du contenu (RSS, Atom)
- Tags et folksonomies
- Contenu dynamique
- Collaboration et partage
- Recherche par mots-clés

Web 3.0

?



"The **web of human-readable document** is being merged with a **web of machine understandable data**. The potential of the mixture of humans and **machines** working together and communication through the web could be immense."

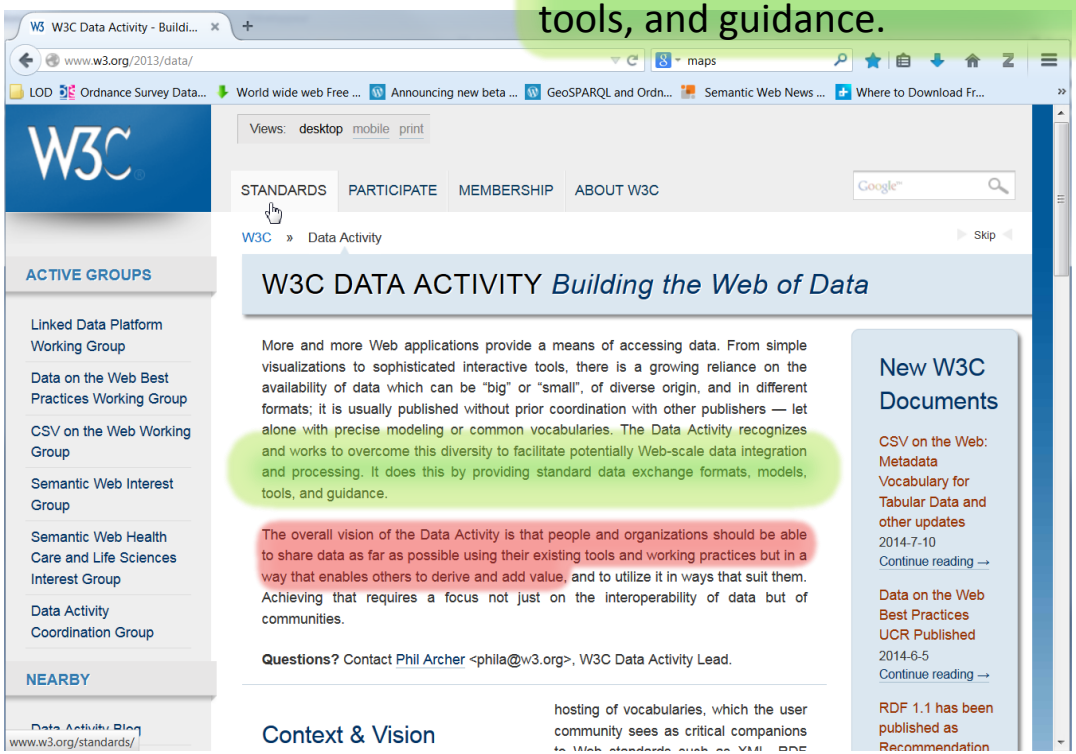
Tim Berners-Lee, The World Wide Web: A very short personal history, May 1998
http://www.w3.org/People/Berners-Lee/ShortHistory.html

*d'après le cours Technologie du Web – A. Hombiat - Licence Professionnelle Études Statistiques et Systèmes d'Information Géographique (ESSIG) – UPMF - 2015

From Web of Documents to Web of Data

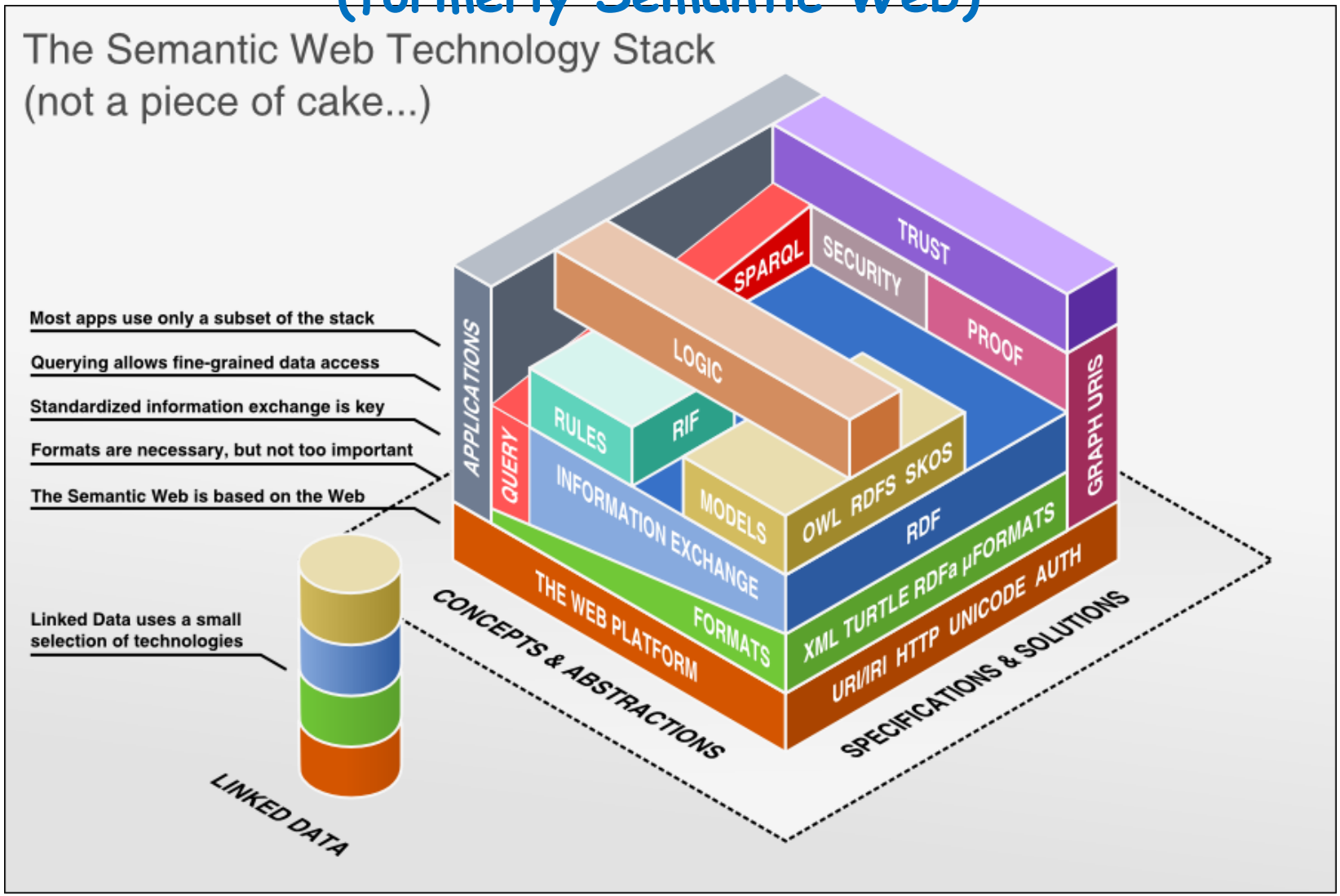
- Web can be seen as a (very) large distributed database (knowledge base) of information accessible to machines.

The Data Activity recognizes and works to overcome this diversity to facilitate potentially **Web-scale data integration and processing**. It does this by providing **standard** data exchange formats, models, tools, and guidance.



people and organizations should be able to share data as far as possible using their existing tools and working practices **but in a way that enables others to derive and add value, and to utilize it in ways that suit them**

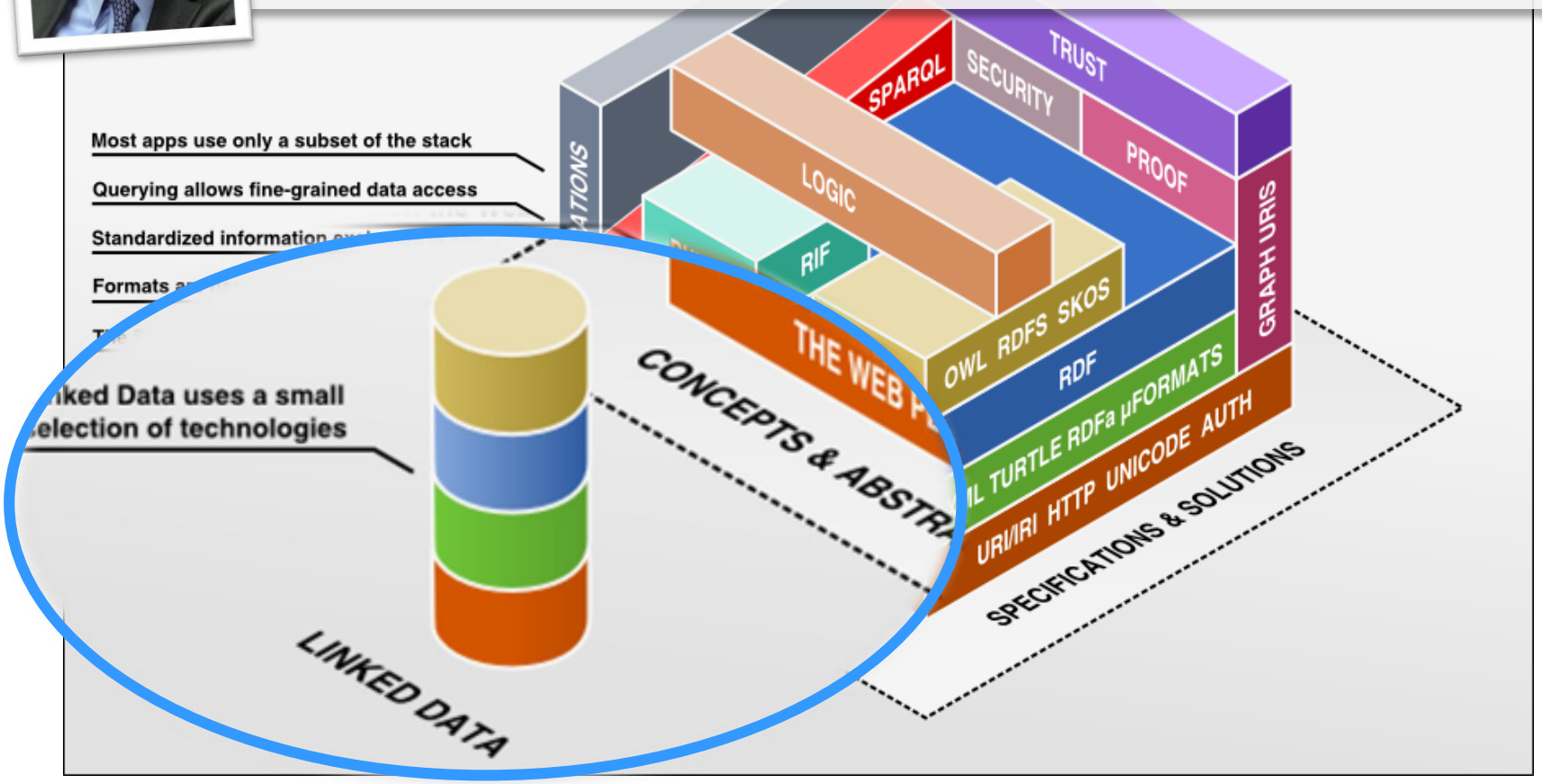
Standards for Web of Data
(formerly Semantic Web)



http://www.bnode.org/blog/tag/layer%20cake (Benjamin Nowack)

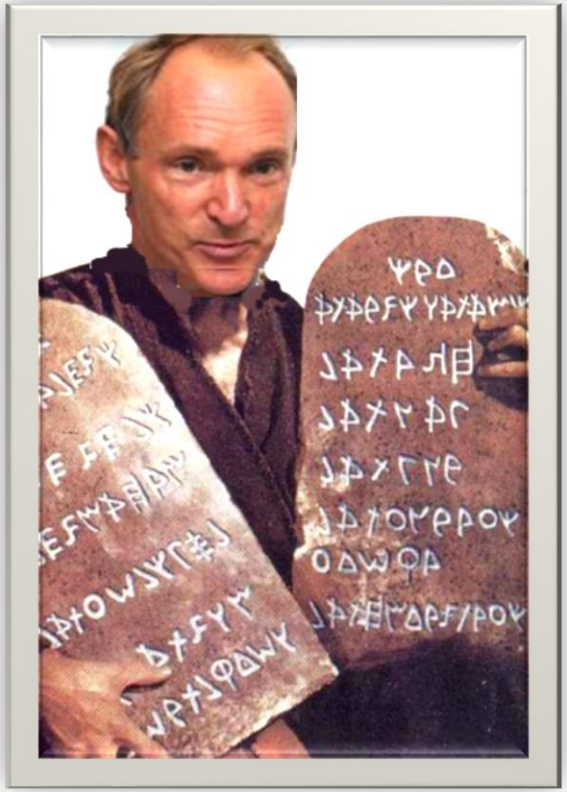
Linked Data

"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."
Tim Berners-Lee - 2006 http://www.w3.org/DesignIssues/LinkedData.html



http://www.bnode.org/blog/tag/layer%20cake (Benjamin Nowack)

Linked Data Principles



Tim Berners-Lee
http://www.w3.org/DesignIssues/LinkedData.html

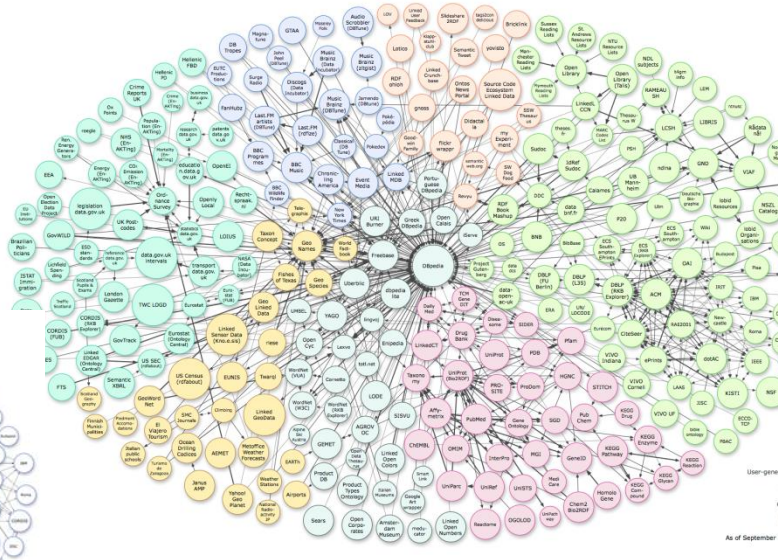
a set of best practices for publishing and interlinking structured data on the Web

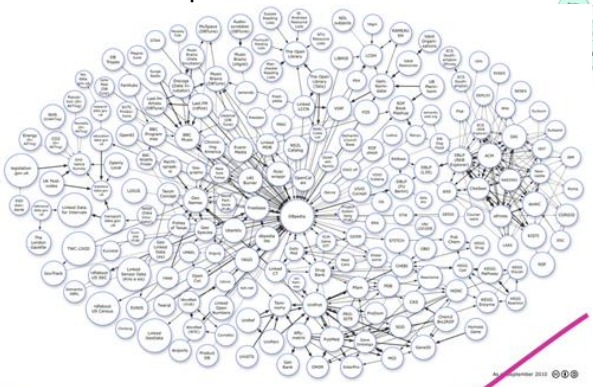
Basic idea: to apply the general architecture of the World Wide Web to the task of sharing structured data on global scale.

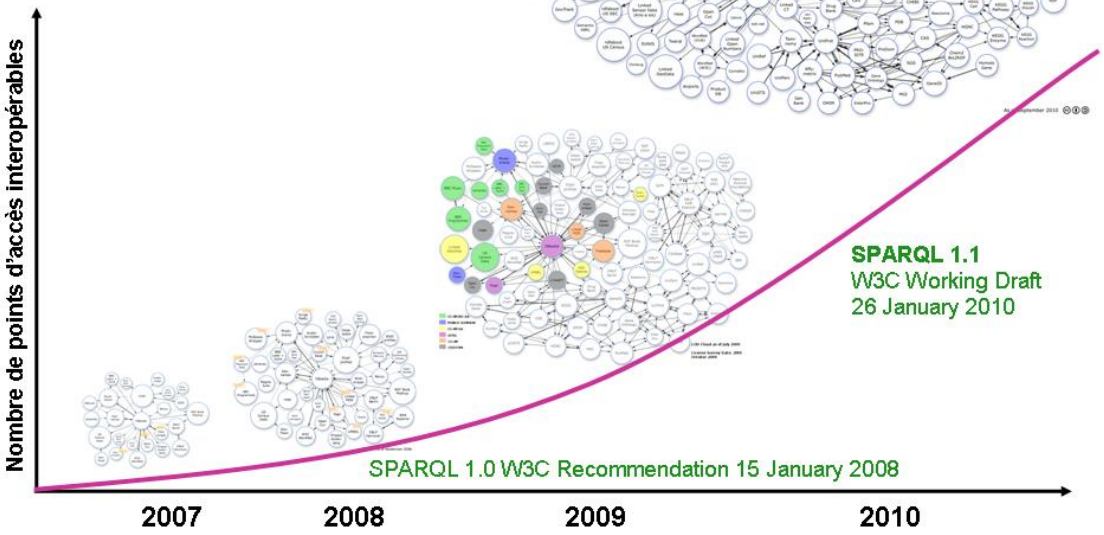
1. Use **URIs** as names for things.
2. Use **HTTP URIs**, so that people can look up those names.
3. When someone looks up a URI, **provide useful information, using the standards** (RDF, SPARQL).
4. Include **links** to other URIs, so that they **can discover more things**.

Linked Open Data Cloud

- Linking open data project
  - goals:
    - Use RDF to “expose” open data sets
    - Create RDF links between these datasets
    - If possible, deploy SPARQL endpoints







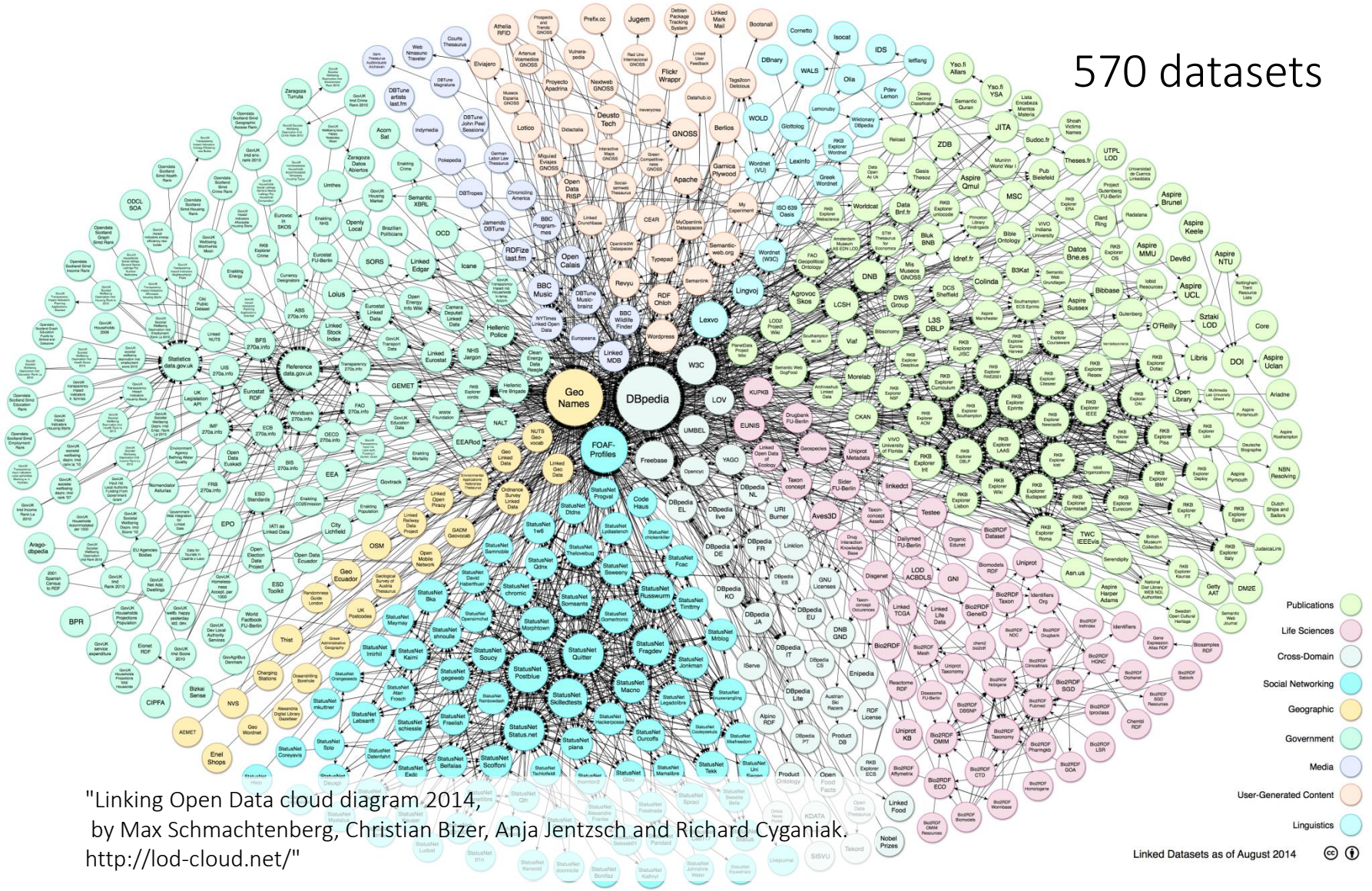
September 2011

295 datasets
31 Billion RDF triples
interconnected by
595 Million of RDF relations

`http://lod-cloud.net`

Linked Open Data Cloud

570 datasets

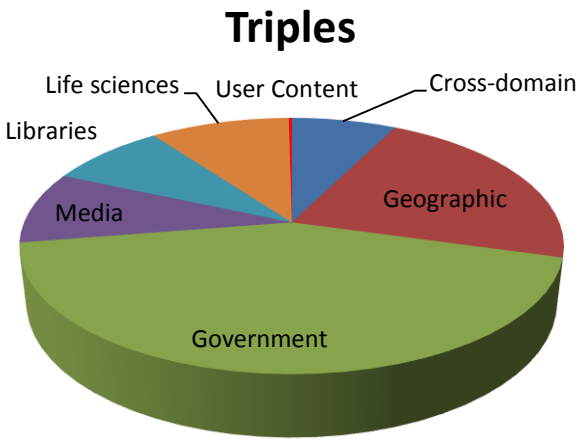


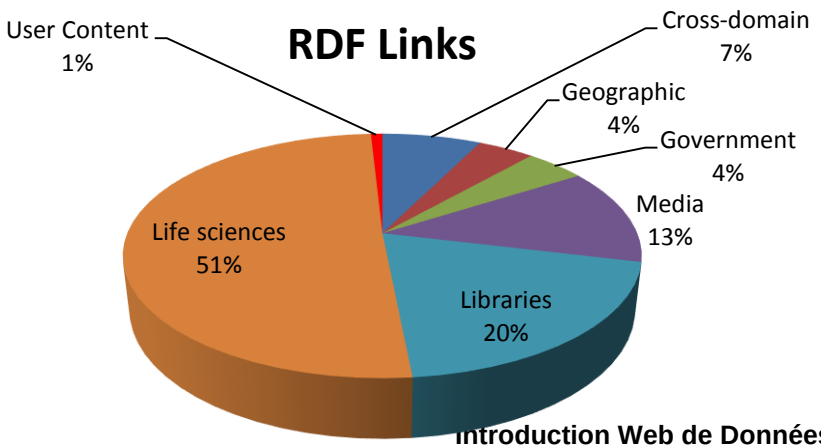
"Linking Open Data cloud diagram 2014,
by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak.
http://lod-cloud.net/"

Linked Open Data Cloud

Domain	Data Sets	Triples	Percent	RDF Links	Percent
Cross-domain	20	1,999,085,950	7.42	29,105,638	7.36
Geographic	16	5,904,980,833	21.93	16,589,086	4.19
Government	25	11,613,525,437	43.12	17,658,869	4.46
Media	26	2,453,898,811	9.11	50,374,304	12.74
Libraries	67	2,237,435,732	8.31	77,951,898	19.71
Life sciences	42	2,664,119,184	9.89	200,417,873	50.67
User Content	7	57,463,756	0.21	3,402,228	0.86
(2011 September)	203	26,930,509,703		395,499,896	

http://lod-cloud.net/state





Linked Open Data Cloud

Domain	Data Sets	Triples
Cross-domain	20	1,999,085
Geographic	16	5,904,980
Government	25	11,613,525
Media	26	2,453,898
Libraries	67	2,237,435
Life sciences	42	2,664,119
User Content	7	57,463
	203	26,930,509

(2011, September)

Datasets by topical domain.

Topic	Datasets	%
Government	183	18.05%
Publications	96	9.47%
Life sciences	83	8.19%
User-generated content	48	4.73%
Cross-domain	41	4.04%
Media	22	2.17%
Geographic	21	2.07%
Social web	520	51.28%
Total	1014	

(2014, August)

State of the LOD Cloud 2014

Version 0.4, 08/30/2014

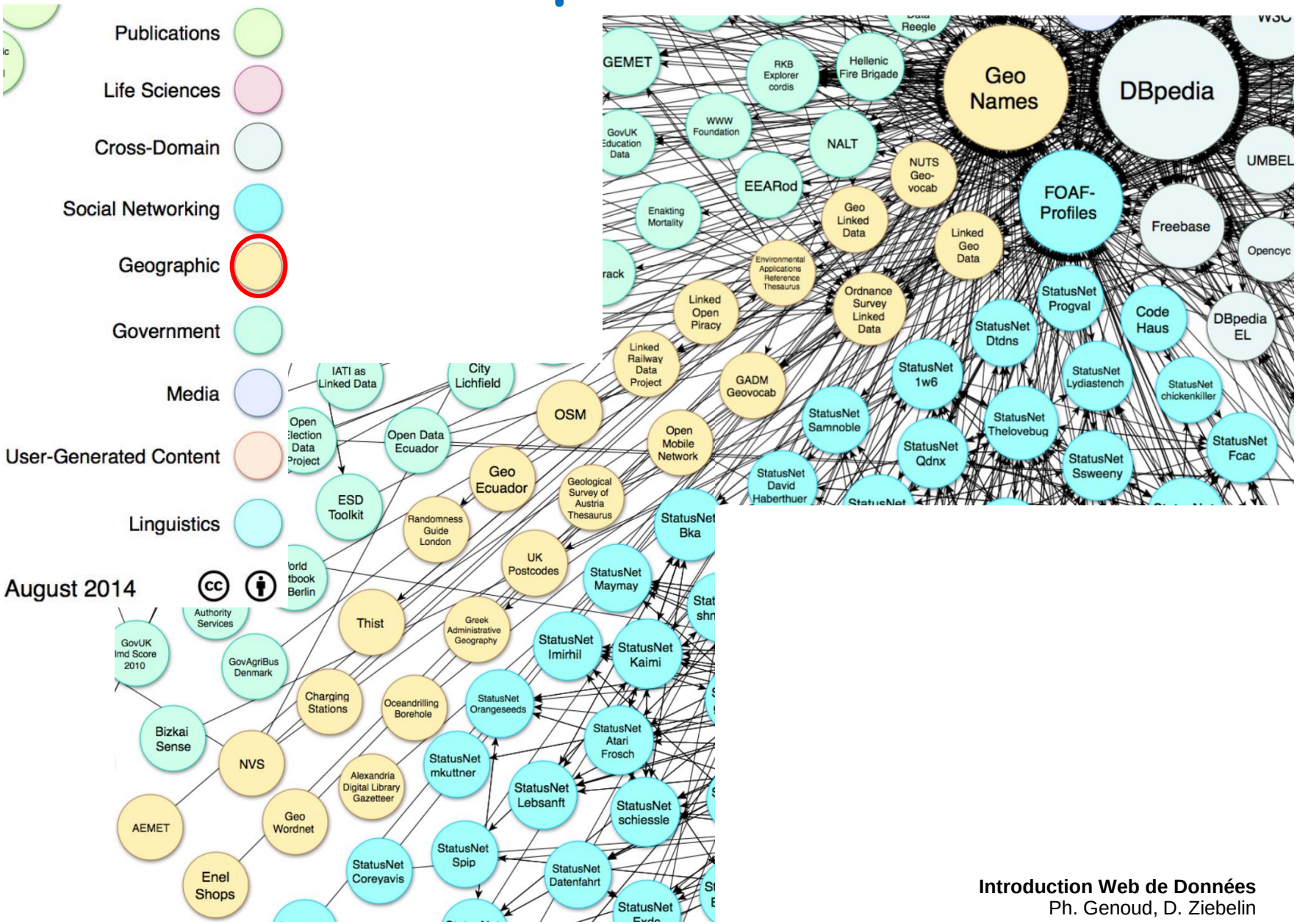
This document provides statistics about the structure and content of the crawlab
Linked Data sources implement the Linked Data best practices.

This document updates the findings of the original State of the LOD Cloud re
publishers themselves via the datahub.io Linked Data catalog. This report is base
the ISWC2014 paper Adoption of the Linked Data Best Practices in Different To
document links the statistics to the Mannheim Linked Data catalog and enables th

Contents

http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/

Linked Open Data Cloud

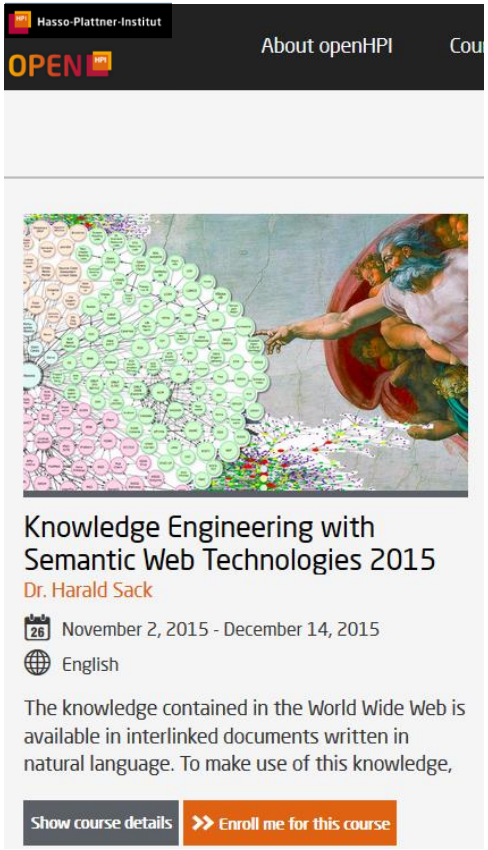


Course Objectives

- To present the common framework the Semantic Web provides to allow data to be shared and reused across application, enterprise, and community boundaries.
- To introduce the core technologies for data exchange and querying : RDF and SPARQL.
- To introduce to knowledge representation with ontology languages : RDFS - OWL

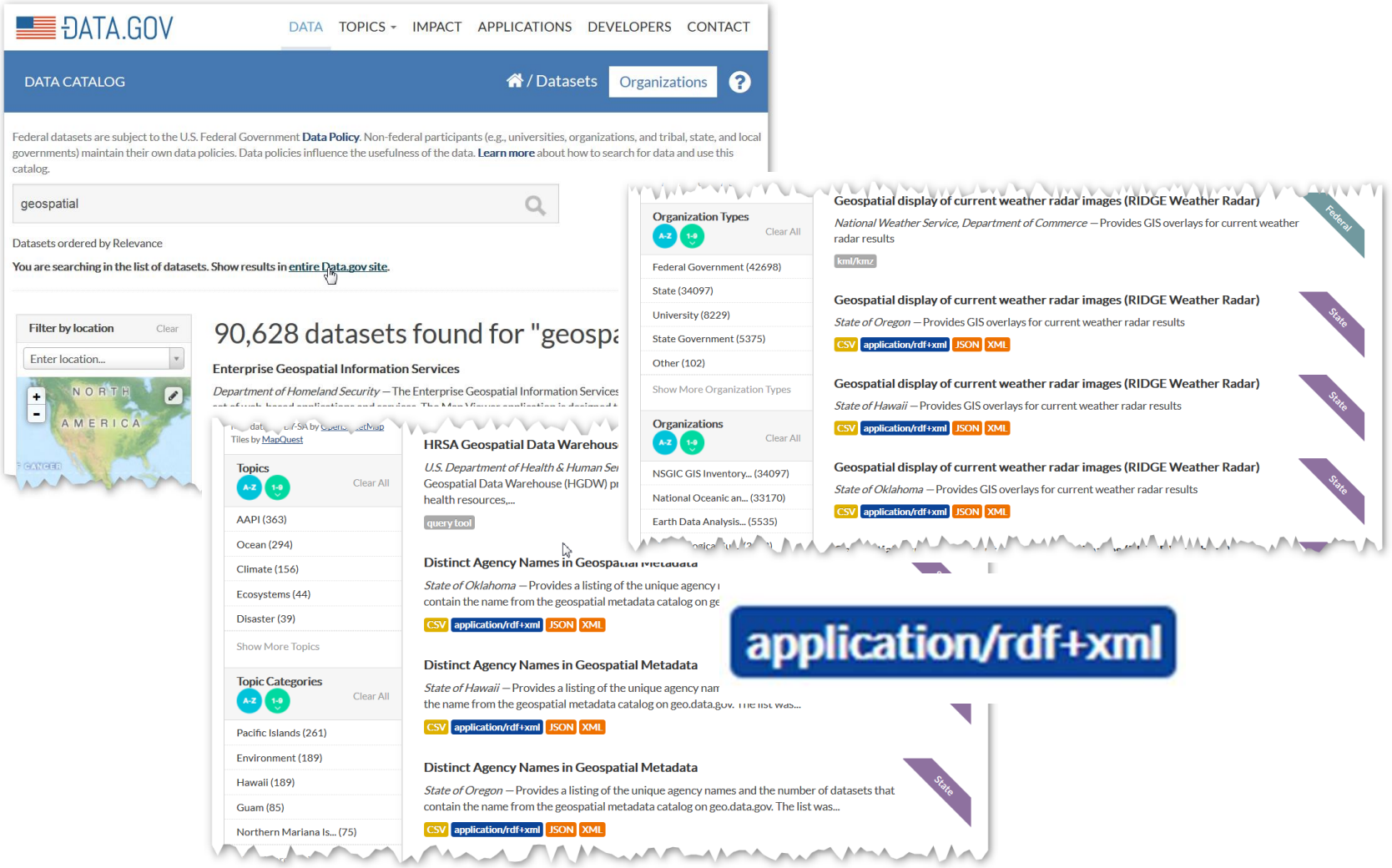
Pourquoi ce cours ?

D'autres le font pourquoi pas nous ?



Pourquoi ce cours ?

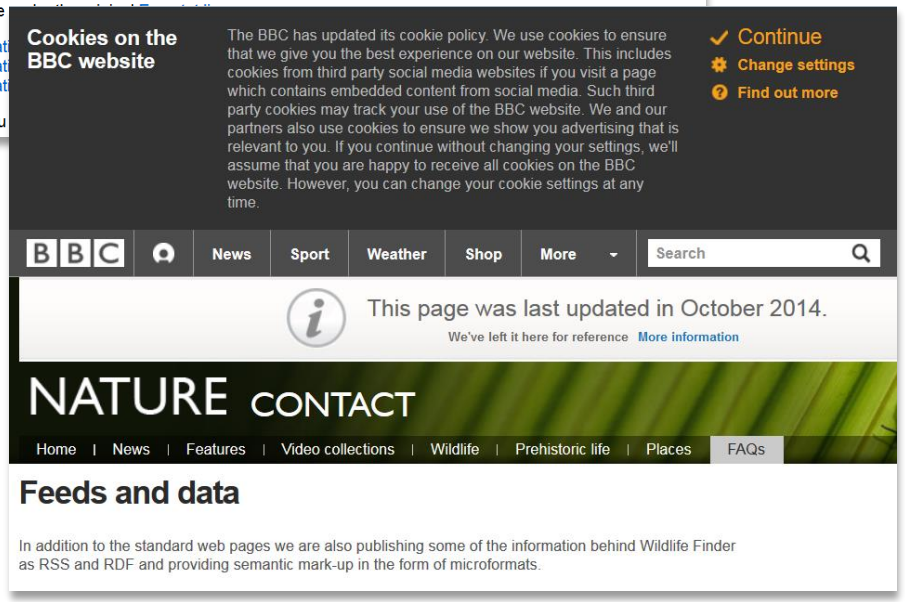
Ces technologies sont en cours d'adoption



http://www.insee.fr/fr/methodes/default.asp?page=xml/xml.htm

http://eurostat.linked-statistics.org/

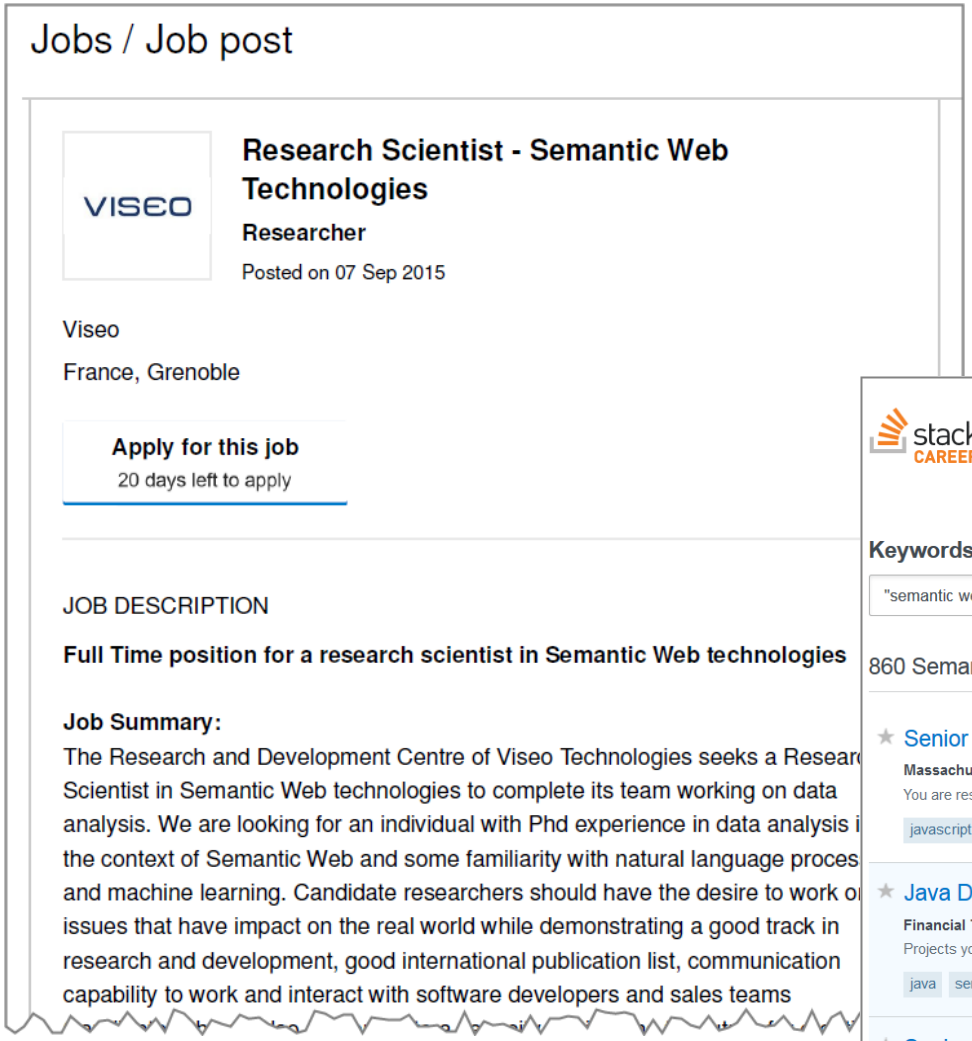
http://www.bbc.co.uk/nature/feedsanddata



. . .

Pourquoi ce cours ?

Il y a même des boulôts



Jobs / Job post

VISEO

Research Scientist - Semantic Web Technologies

Researcher

Posted on 07 Sep 2015

Viseo

France, Grenoble

Apply for this job

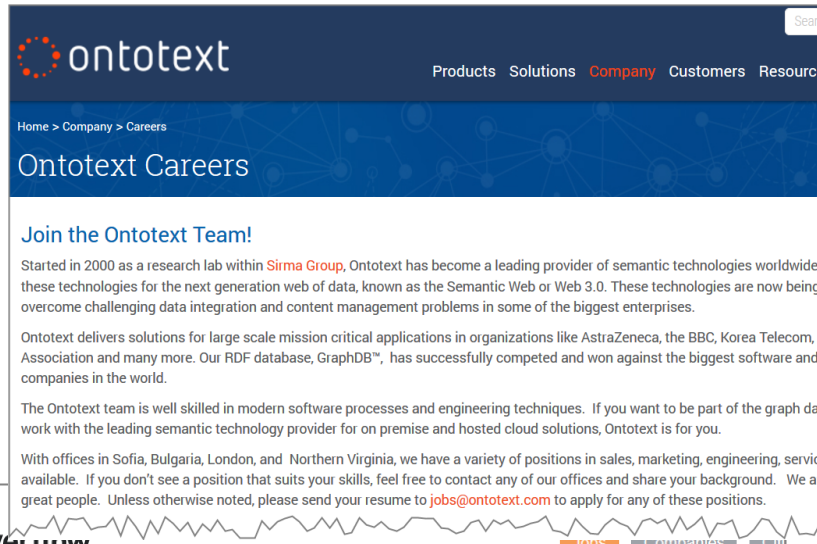
20 days left to apply

JOB DESCRIPTION

Full Time position for a research scientist in Semantic Web technologies

Job Summary:

The Research and Development Centre of Viseo Technologies seeks a Researc
Scientist in Semantic Web technologies to complete its team working on data
analysis. We are looking for an individual with Phd experience in data analysis i
the context of Semantic Web and some familiarity with natural language proces
and machine learning. Candidate researchers should have the desire to work o
issues that have impact on the real world while demonstrating a good track in
research and development, good international publication list, communication
capability to work and interact with software developers and sales teams



ontotext

Products Solutions Company Customers Resourc

Home > Company > Careers

Ontotext Careers

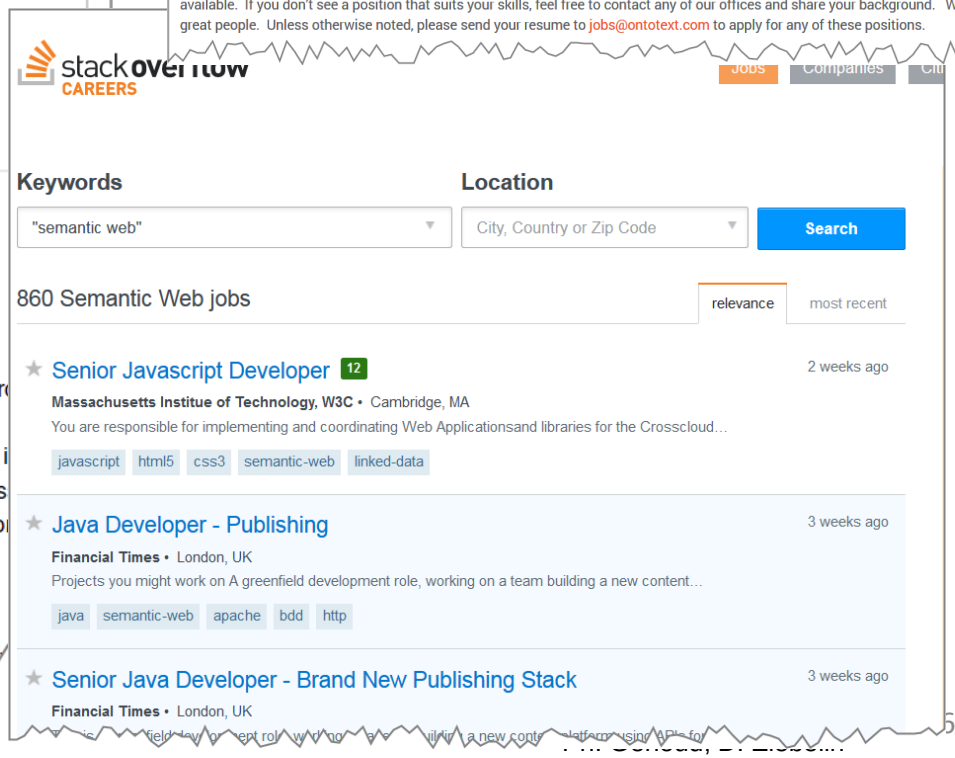
Join the Ontotext Team!

Started in 2000 as a research lab within Sirma Group, Ontotext has become a leading provider of semantic technologies worldwide these technologies for the next generation web of data, known as the Semantic Web or Web 3.0. These technologies are now being overcome challenging data integration and content management problems in some of the biggest enterprises.

Ontotext delivers solutions for large scale mission critical applications in organizations like AstraZeneca, the BBC, Korea Telecom, Association and many more. Our RDF database, GraphDB™, has successfully competed and won against the biggest software and companies in the world.

The Ontotext team is well skilled in modern software processes and engineering techniques. If you want to be part of the graph da work with the leading semantic technology provider for on premise and hosted cloud solutions, Ontotext is for you.

With offices in Sofia, Bulgaria, London, and Northern Virginia, we have a variety of positions in sales, marketing, engineering, servic available. If you don't see a position that suits your skills, feel free to contact any of our offices and share your background. We a great people. Unless otherwise noted, please send your resume to jobs@ontotext.com to apply for any of these positions.



stackoverflow CAREERS

Jobs Companies

Keywords

"semantic web"

Location

City, Country or Zip Code

Search

860 Semantic Web jobs

relevance most recent

Senior Javascript Developer 12 — 2 weeks ago

Massachusetts Institue of Technology, W3C • Cambridge, MA

You are responsible for implementing and coordinating Web Applicationsand libraries for the Crosscloud…

javascript html5 css3 semantic-web linked-data

Java Developer - Publishing — 3 weeks ago

Financial Times • London, UK

Projects you might work on A greenfield development role, working on a team building a new content…

java semantic-web apache bdd http

Senior Java Developer - Brand New Publishing Stack — 3 weeks ago

Financial Times • London, UK

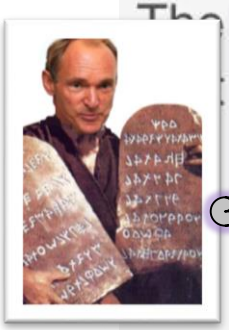
Course Outline

- Introduction
- Distributing Data on the web with RDF
  - Naming the Data : URIs (Uniform Resources Identifiers)
  - The RDF Data model
- Querying Linked Data with SPARQL
- Semantic modelling
  - RDFS
  - OWL
- From Open Data to Linked Open Data
- Conclusion

Outline

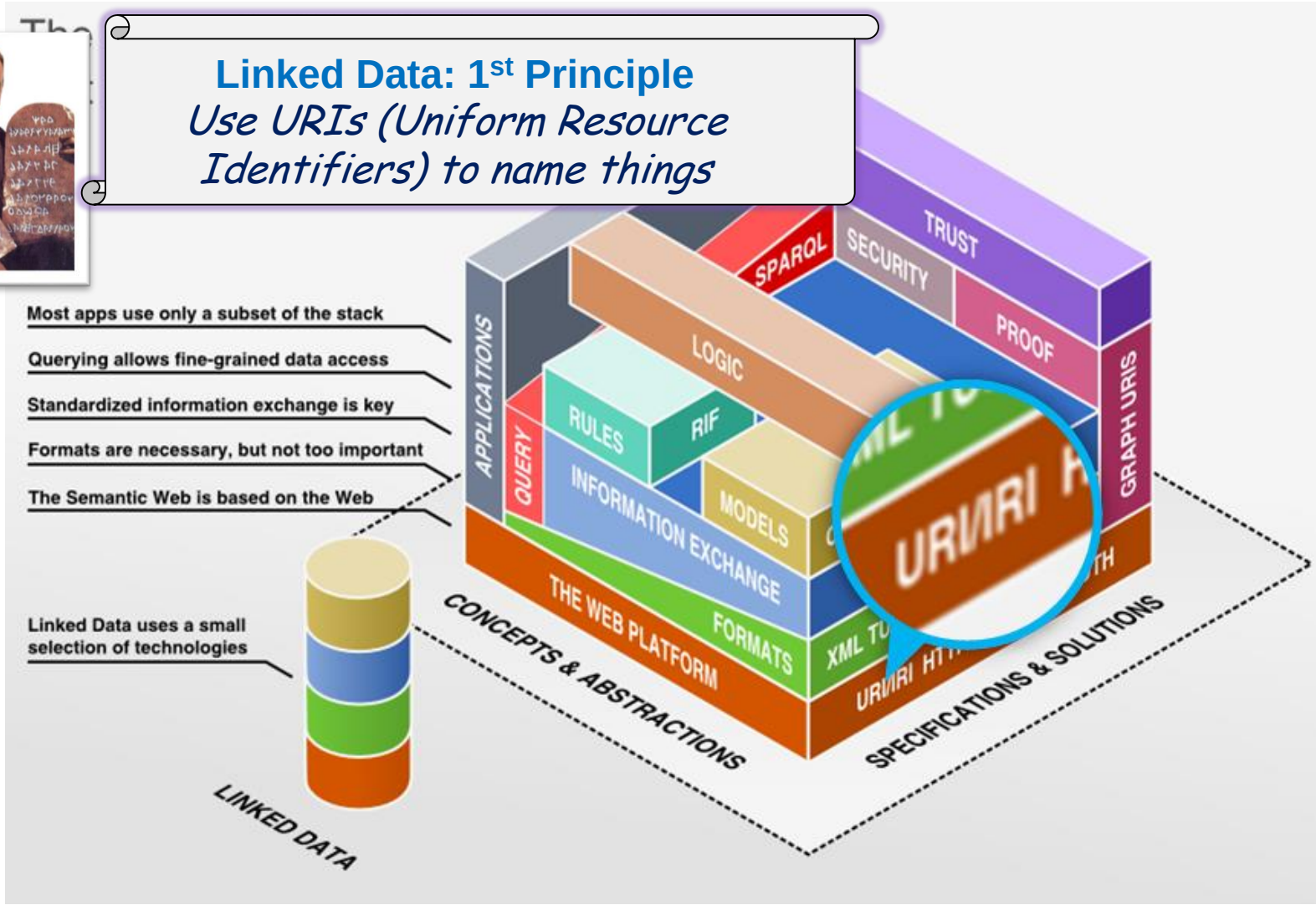
- Introduction
- Distributing Data on the web with RDF
  - Naming the Data : URIs (Uniform Resources Identifiers)
  - The RDF Data model
- Querying Linked Data with SPARQL
- Semantic modelling
  - RDFS
  - OWL
- From Open Data to Linked Open Data
- Conclusion

Uniform Resource Identifiers (URIs)



Linked Data: 1st Principle

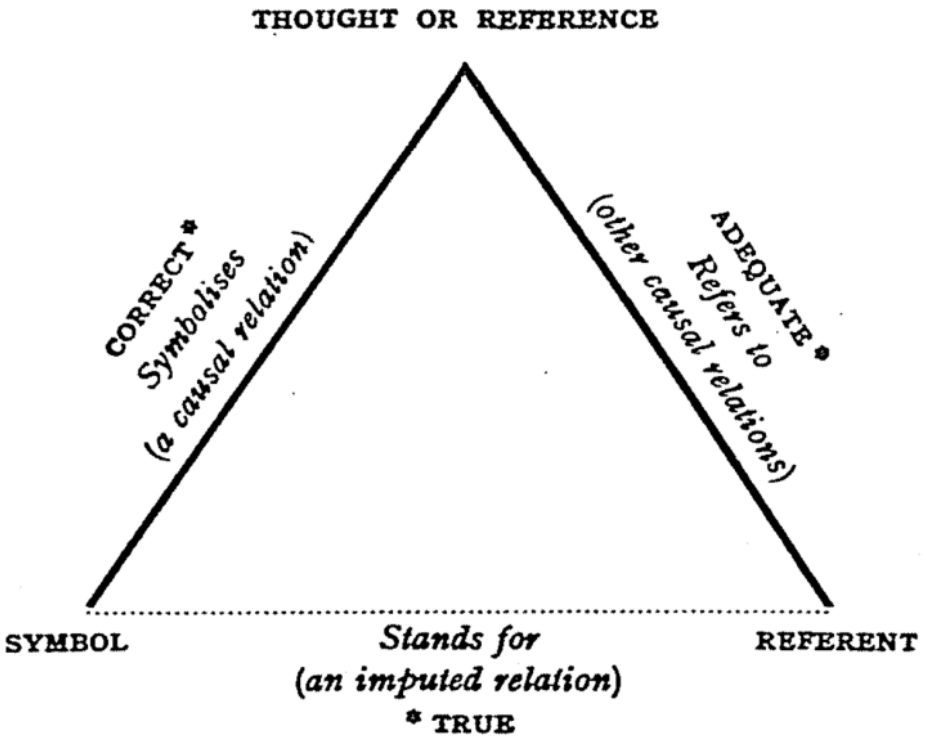
Use URIs (Uniform Resource Identifiers) to name things



http://www.bnode.org/blog/tag/layer%20cake (Benjamin Nowack)

(Semiotics

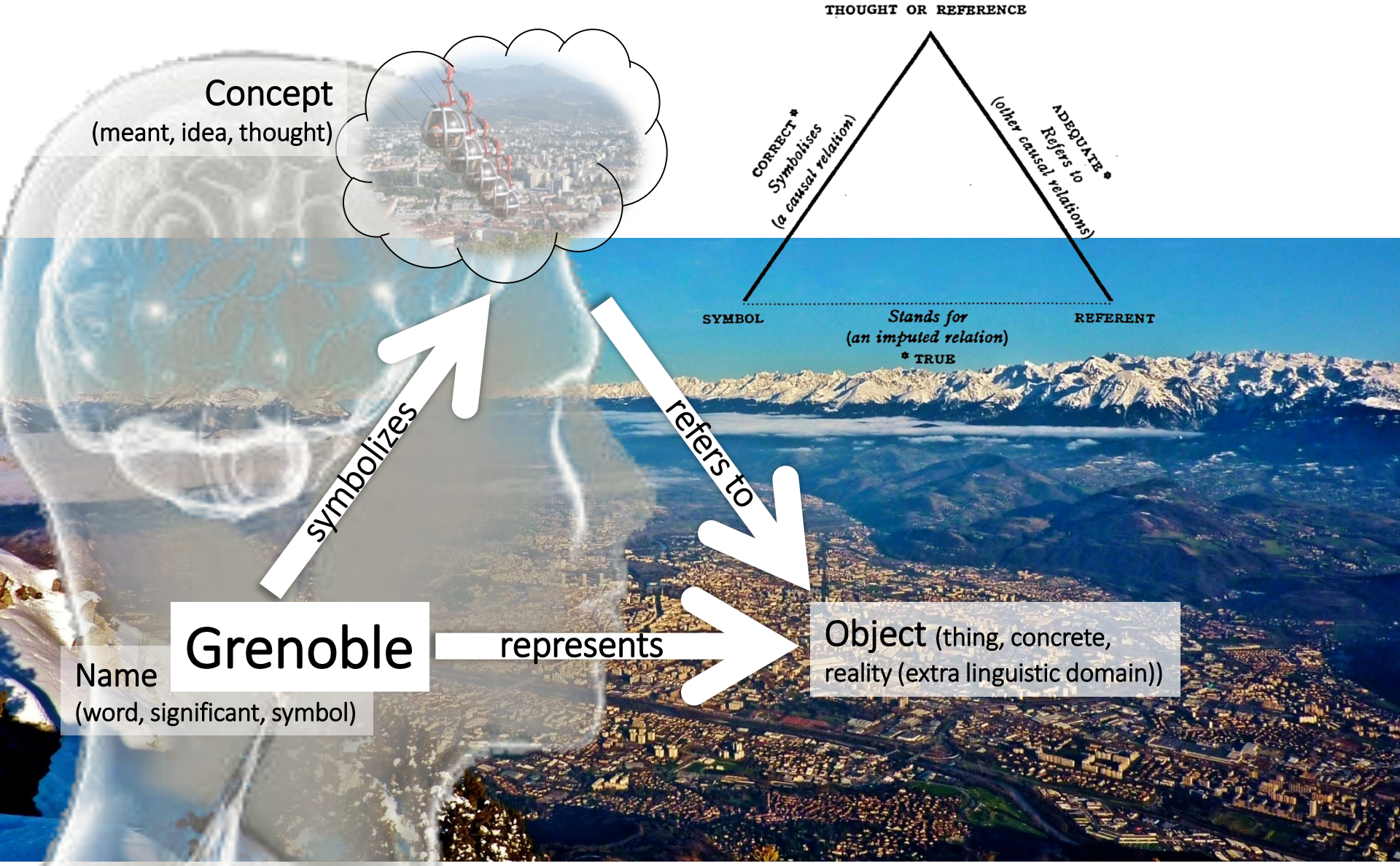
- **Semiotics** : The science of communication studied through the interpretation of signs and symbols as they operate in various fields, esp. language. Oxford English Dictionary (2003).



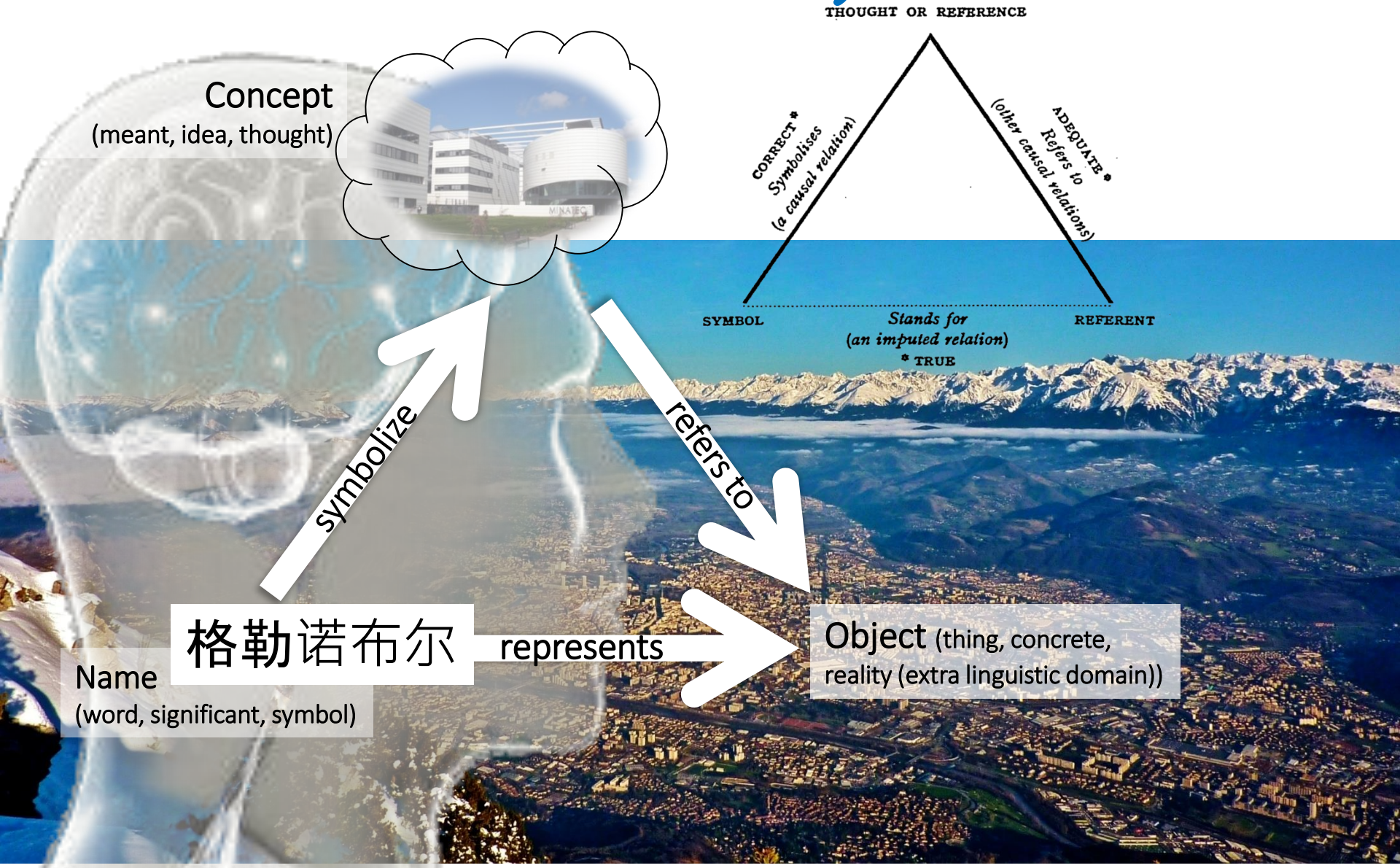
Semiotic Triangle by Odgen & Richard

([*The Meaning of Meaning*](#) -A Study of the Influence of Language upon Thought and of the Science of Symbolism, 1923)

Semiotics


Concept
(meant, idea, thought)
THOUGHT OR REFERENCE
CORRECT *
Symbolises
(a causal relation)
ADEQUATE *
Refers to
(other causal relations)
SYMBOL
Stands for
(an imputed relation)
* TRUE
REFERENT
symbolizes
refers to
Grenoble
represents
Name
(word, significant, symbol)
Object (thing, concrete, reality (extra linguistic domain))

Semiotics)


Concept
(meant, idea, thought)
THOUGHT OR REFERENCE
CORRECT * Symbolises (a causal relation)
ADEQUATE * Refers to (other causal relations)
SYMBOL
Stands for (an imputed relation)
* TRUE
REFERENT
symbolize
refers to
格勒诺布尔
represents
Name
(word, significant, symbol)
Object (thing, concrete, reality (extra linguistic domain))

URI: definition

- "In computing, a **Uniform Resource Identifier (URI)** is a string of characters used to identify the name of a resource. Such identification enables interaction with representations of the resource over a network, typically the World Wide Web, using specific protocols. Schemes specifying a concrete syntax and associated protocols define each URI." (see RFC 3986)
  https://en.wikipedia.org/wiki/Uniform_resource_identifier

Knowledge Engineering with Semantic Web Technologies , Dr. Harald Sack, Hasso-Plattner-Institut, Universität Potsdam

URI: syntax

- URI : generic syntax

scheme ":" ["//" authority "/"] [path] ["?" query] ["#" fragment]

- **scheme:** http, ftp, mailto, …
- **authority:** [userinfo@]host[:port]
  - **userinfo:** authentication section e.g: username:password
  - **host:** domain name, IP address
  - **port:** port number, ex: 80 for HTTP standard port
- **path:** a sequence of segments separated by slashes, e.g. : a path in the hierarchical file system of the HTTP server.
- **query:** a query string of non-hierarchical data. (e.g: a sequence of attribute–value pairs separated by a delimiter (&) for HTTP requests)
- **fragment:** a fragment identifier providing direction to a secondary resource (e.g.: anchor id in a HTML document)

Internationalized Resource Identifier (RFC 3987): extension to support Universal Character Set (Unicode/ISO 10646))

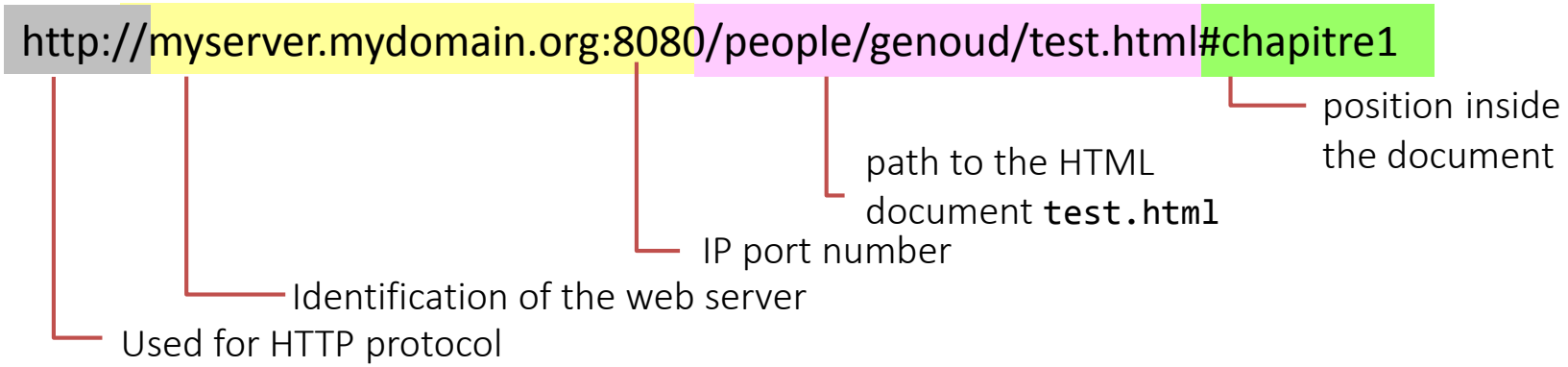
http://fa.dbpedia.org/resource/ژرژ_برسنس

URI: examples

- URI : generic syntax

scheme ":" ["//" authority "/"] [path] ["?" query] ["#" fragment]

example :

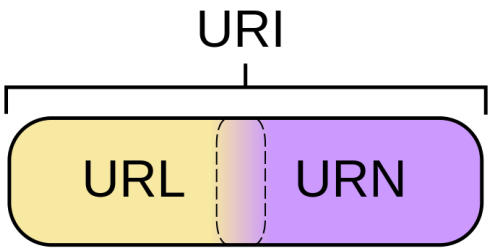


other URIs examples:

ftp://server.example.com/foo
mailto:person@example.fr
urn:isbn:978-0553283686

URI - URL - URN

- an URI/IRI **doesn't necessarily** identifies a resource that is resolvable on the web

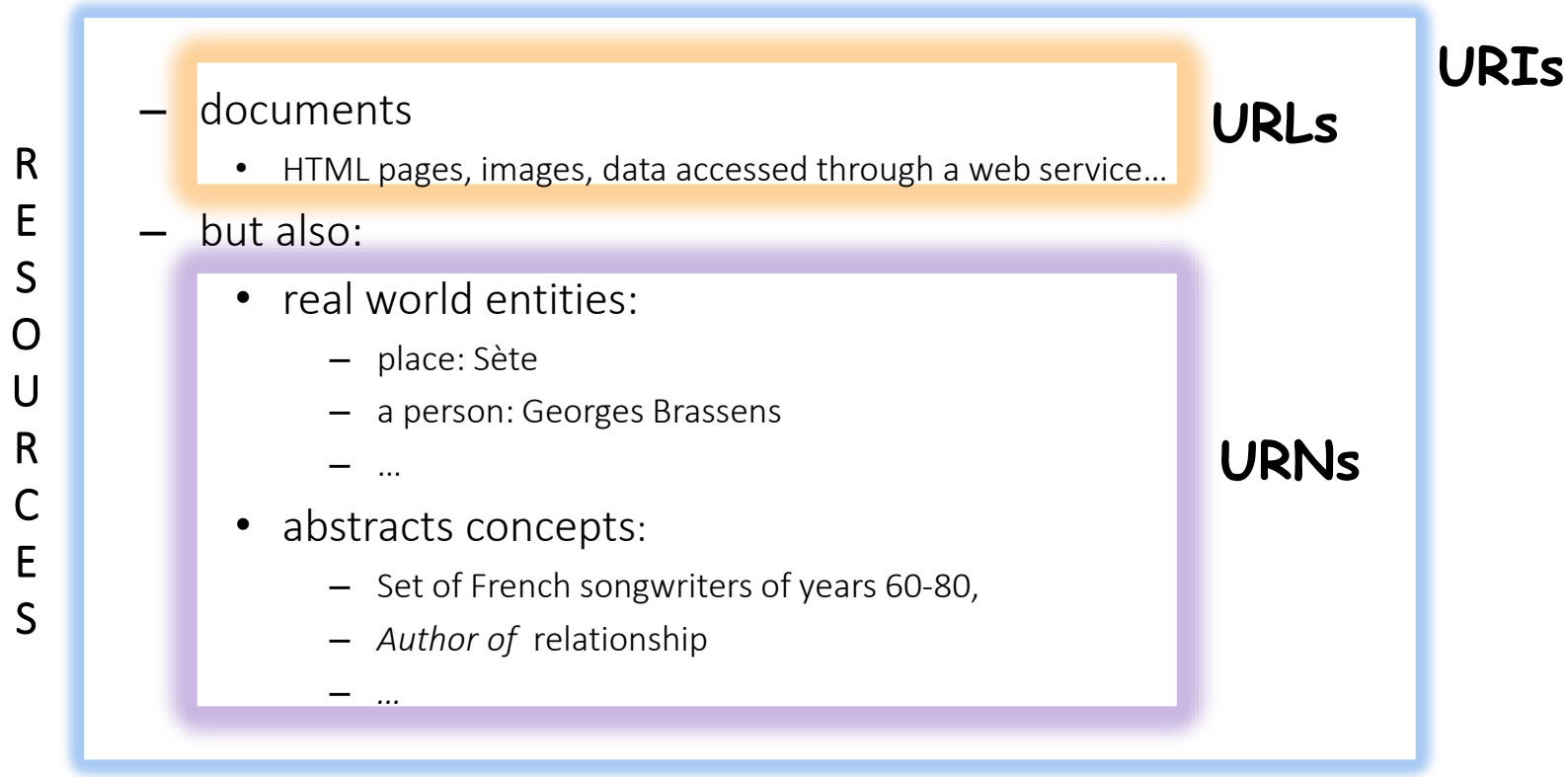


- Address (Locator)
  - **U**niform **R**esource **L**ocator (RFC 1738)
  - Tells where and how a resource can be found in the internet
  - *Can change during the life cycle of a resource*
- Identity (Name)
  - **U**niform **R**esource **N**ame (RFC 2141)
  - identifies a resource by name in a particular namespace. A URN can be used to talk about a resource without implying its location or how to access it.
  - *Remains unchanged during life cycle of the resource*

URIs, URLs, and URNs: Clarifications and Recommendations 1.0
Report from the joint W3C/IETF URI Planning Interest Group- W3C Note 21 September 2001
http://www.w3.org/TR/uri-clarification/

Resources in the Web of data

- Data describe elements (things) for a domain of interest through their properties and relationships.
- Theses elements can be:



- All these elements are resources identified by an URI

Examples of resources about Georges Brassens available in the web

Traditional web

Documents identified by URLs

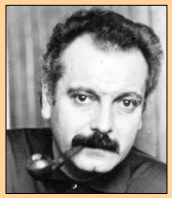
DBpedia

XML file containing structured data about Georges Brassens
http://dbpedia.org/data/Georges_Brassens.xml

Web page (HTML file)

http://fr.wikipedia.org/wiki/Georges_Brassens

Picture (jpeg file)



http://culturetheque.org.uk/media/item/17545/800/brassens.jpg

Video (mp4 file)
https://www.youtube.com/watch?v=rslShTbqNbo

Examples of resources about Georges Brassens available in the web

Traditional web

Documents

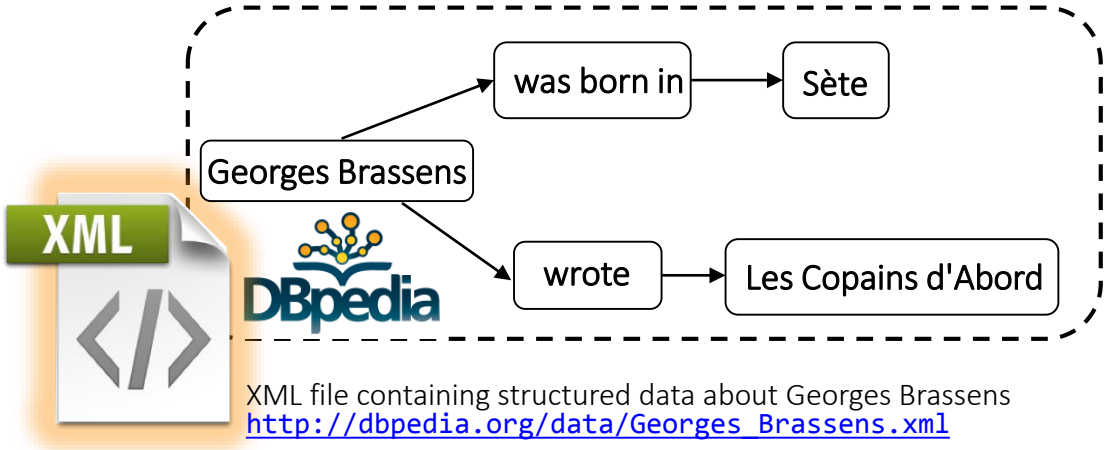
identified by URLs

Web of Data

Real world entities

Abstract concepts

identified by URNs

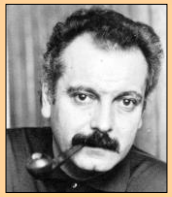


XML file containing structured data about Georges Brassens
http://dbpedia.org/data/Georges_Brassens.xml

Web page (HTML file)

http://fr.wikipedia.org/wiki/Georges_Brassens

Picture (jpeg file)



http://culturetheque.org.uk/media/item/17545/800/brassens.jpg

Video
(mp4 file)
https://www.youtube.com/watch?v=rslShTbqNbo

Examples of resources about Georges Brassens available in the web

Traditional web

Documents

identified by URLs

Web of Data

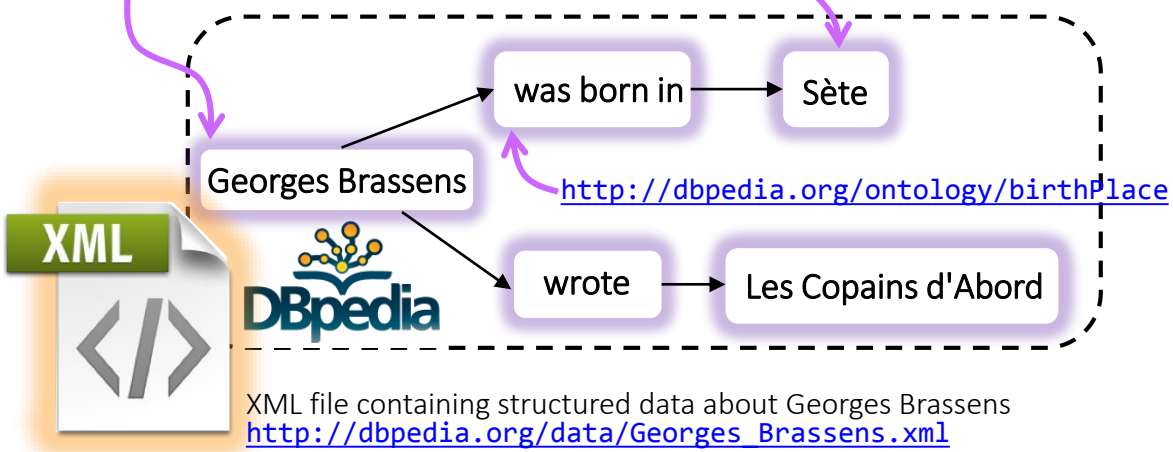
Real world entities

Abstract concepts

identified by URNs

http://dbpedia.org/resource/Georges_Brassens
DBpedia resource representing Georges Brassens

http://dbpedia.org/resource/Sète

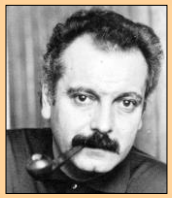


XML file containing structured data about Georges Brassens
http://dbpedia.org/data/Georges_Brassens.xml

Web page (HTML file)

http://fr.wikipedia.org/wiki/Georges_Brassens

Picture (jpeg file)



http://culturetheque.org.uk/media/item/17545/800/brassens.jpg

Video (mp4 file)
https://www.youtube.com/watch?v=rslShTbqNbo

Uniform Resource Identifiers (URIs)

represents

represents

DBpedia

GeoNames

`http://dbpedia.org/resource/grenoble`

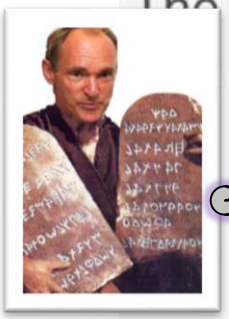
`http://sws.geonames.org/3014728`

The name (URI) that represents the city of Grenoble in DBpedia

The name (URI) that represents the city of Grenoble in GeoNames

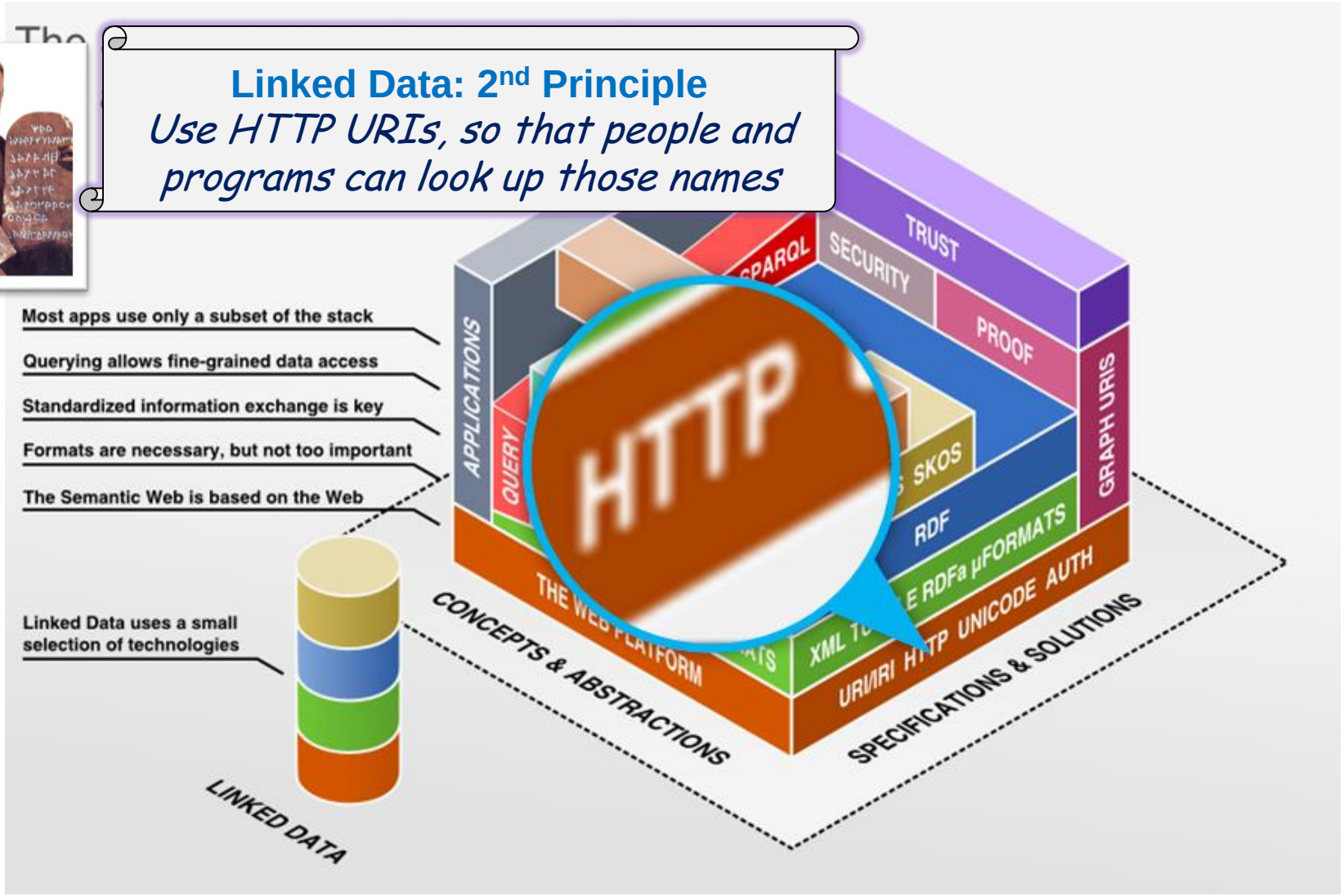
Different URIs (URNs) in different namespaces can represents the same thing

HTTP URIs



Linked Data: 2nd Principle

Use HTTP URIs, so that people and programs can look up those names



http://www.bnode.org/blog/tag/layer%20cake (Benjamin Nowack)

HTTP URIs

- HTTP (Hyper Text Transfer Protocol) protocol is the Web's universal access mechanism.
- HTTP URIs make good names for two reasons*:
  - They provide a simple way to create globally unique names in a decentralized fashion, as every owner of a domain name, or delegate of the domain name owner, may create new URI references.
  - They serve not just as a name but also as a means of accessing information describing the identified
    - HTTP clients can **dereference** (i.e., look up) the URI using the HTTP protocol and retrieve a description of the resource that is identified by the URI.

* Tom Heath and Christian Bizer (2011)
Linked Data: Evolving the Web into a Global Data Space (1st edition).
Synthesis Lectures on the Semantic Web: Theory and Technology, 1:1, 1-136. Morgan & Claypool.
http://linkeddatabook.com/editions/1.0/

HTTP URIs

- Where HTTP URIs identify real-world objects or abstract concepts, it is essential to **not confuse** the objects or concepts themselves with the Web documents that describe them.
  - A real world object or abstract concept can have different representations
  - It allows separate statements to be made about an object and about a document that describes that object.

URI of DBpedia resource representing Georges Brassens

http://dbpedia.org/resource/Georges_Brassens

a URN
but not
a URL

http://dbpedia.org/page/Georges_Brassens

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa
<html xmlns="http://www.w3.org/1999/xhtml"
    xmlns:dbpprop="http://dbpedia.org/property
    xmlns:foaf="http://xmlns.com/foaf/0.1/"
    version="XHTML+RDFa 1.0" xml:lang="en">

<!-- header -->
<head profile="http://www.w3.org/1999/xhtml/v
    <title>About: Georges Brassens</title>
    <link rel="alternate" type="application/rdf+xml" href="http://dbpe
    <link rel="alternate" type="text/rdf+n3" href="http://dbpedia.org/
    <link rel="alternate" type="application/json+rdf" href="http://dbp
    <link rel="alternate" type="application/json" href="http://dbpedia
    <link rel="alternate" type="application/atom+xml" href="http://dbpe
    <link rel="alternate" type="text/plain" href="http://dbpedia.org/da
    <link rel="alternate" href="http://dbpedia.org/sparql?default-graph

    <link rel="alternate" href="http://dbpedia.org/sparql?default-graph

    <link rel="alternate" href="http://dbpedia.org/sparql?default-graph
    <link rel="alternate" href="http://dbpedia.org/sparql?default-graph
    <link rel="alternate" href="http://dbpedia.org/sparql?default-graph
    <link rel="timegate" type="text/html" href="http://mementoarchive.la
    <link rel="stylesheet" type="text/css" href="/statics/style.css" />
```

URL of HTML Page
Description intended to be read by humans

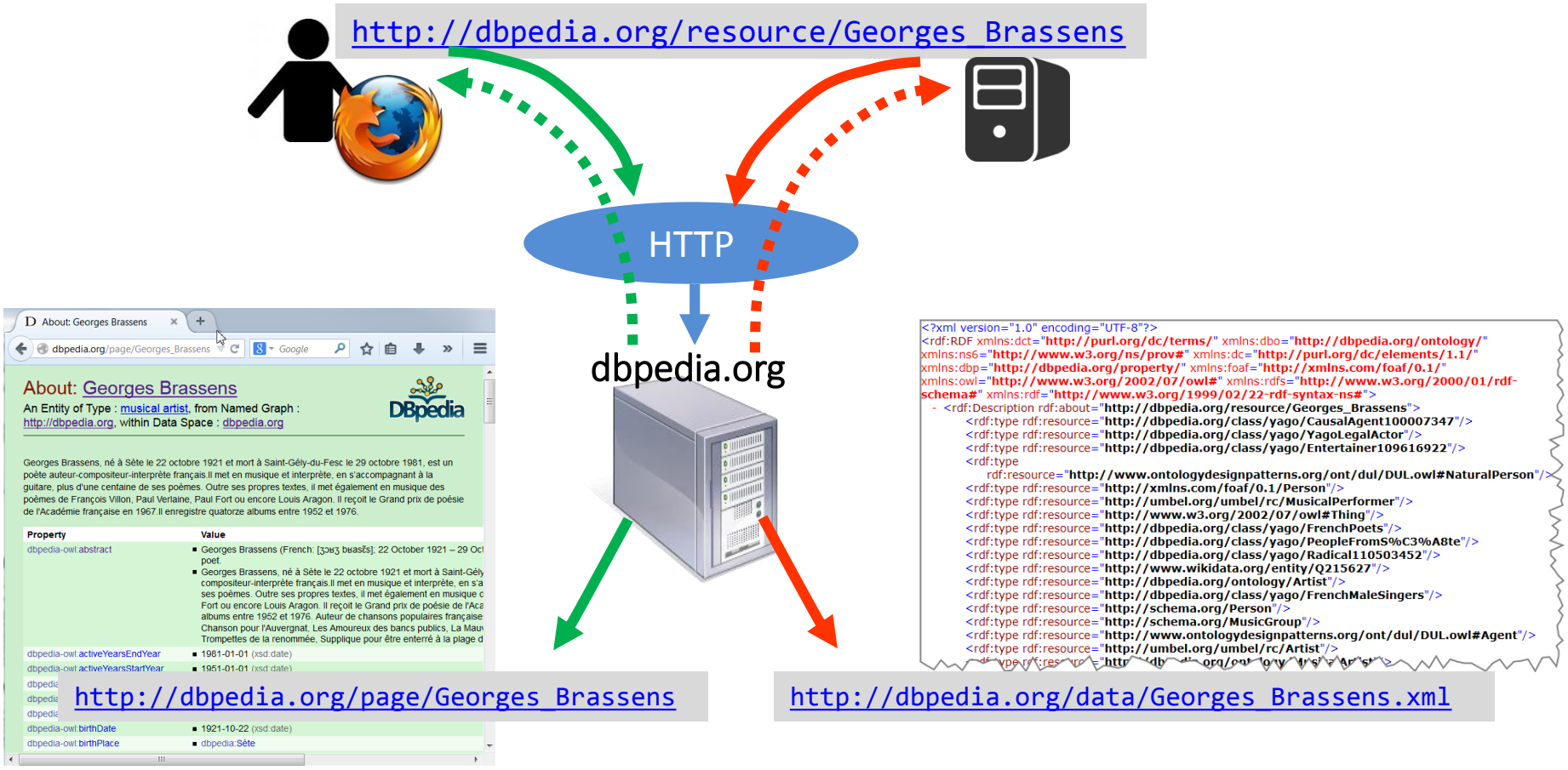
http://dbpedia.org/data/Georges_Brassens.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:dct="http://purl.org/dc/terms/" xmlns:d
xmlns:ns6="http://www.w3.org/ns/prov#" xmlns:dc="h
xmlns:dbp="http://dbpedia.org/property/" xmlns:foaf="
xmlns:owl="http://www.w3.org/2002/07/owl#" xmlns:
schema#" xmlns:rdf="http://www.w3.org/1999/02/22
 - <rdf:Description rdf:about="http://dbpedia.org/resou
      <rdf:type rdf:resource="http://dbpedia.org/class/yago/CausalAgent100007347"/>
      <rdf:type rdf:resource="http://dbpedia.org/class/yago/YagoLegalActor"/>
      <rdf:type rdf:resource="http://dbpedia.org/class/yago/Entertainer109616922"/>
      <rdf:type
        rdf:resource="http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#NaturalPerson"/>
      <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
      <rdf:type rdf:resource="http://umbel.org/umbel/rc/MusicalPerformer"/>
      <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
      <rdf:type rdf:resource="http://dbpedia.org/class/yago/FrenchPoets"/>
      <rdf:type rdf:resource="http://dbpedia.org/class/yago/PeopleFromS%C3%A8te"/>
      <rdf:type rdf:resource="http://dbpedia.org/class/yago/Radical110503452"/>
      <rdf:type rdf:resource="http://www.wikidata.org/entity/Q215627"/>
      <rdf:type rdf:resource="http://dbpedia.org/ontology/Artist"/>
      <rdf:type rdf:resource="http://dbpedia.org/class/yago/FrenchMaleSingers"/>
      <rdf:type rdf:resource="http://schema.org/Person"/>
      <rdf:type rdf:resource="http://schema.org/MusicGroup"/>
      <rdf:type rdf:resource="http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#Agent"/>
      <rdf:type rdf:resource="http://umbel.org/umbel/rc/Artist"/>
```

URL of RDF/XML document
Description intended for consumption by machines

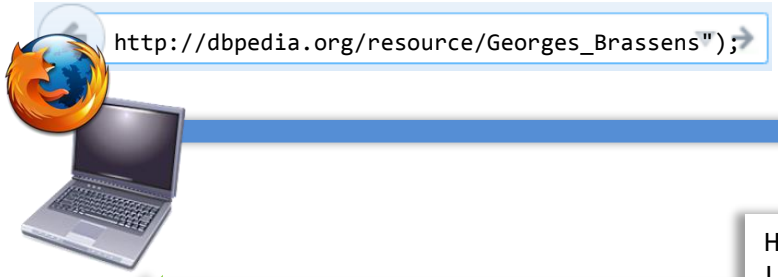
Making URIs Dereferenceable
Content negotiation (303 URIs)

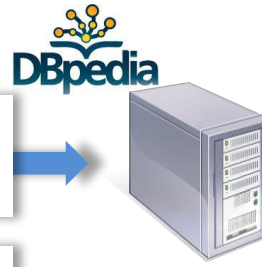
- The same URI can be used to retrieve different representations.



Making URIs Dereferenceable
Content negotiation (303 URIs)

- Contents negotiation uses HTTP headers to a retrieve the resource description

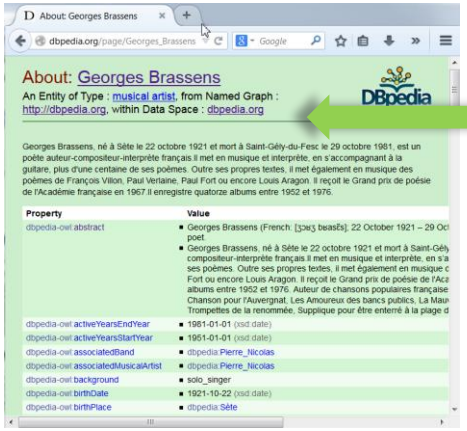





```
GET /resource/Georges_Brassens HTTP/1.1
Host: dbpedia.org
Accept: text/html
```

```
HTTP/1.1 303 See Other
Location: http://dbpedia.org/page/Georges_Brassens
Vary: Accept
```

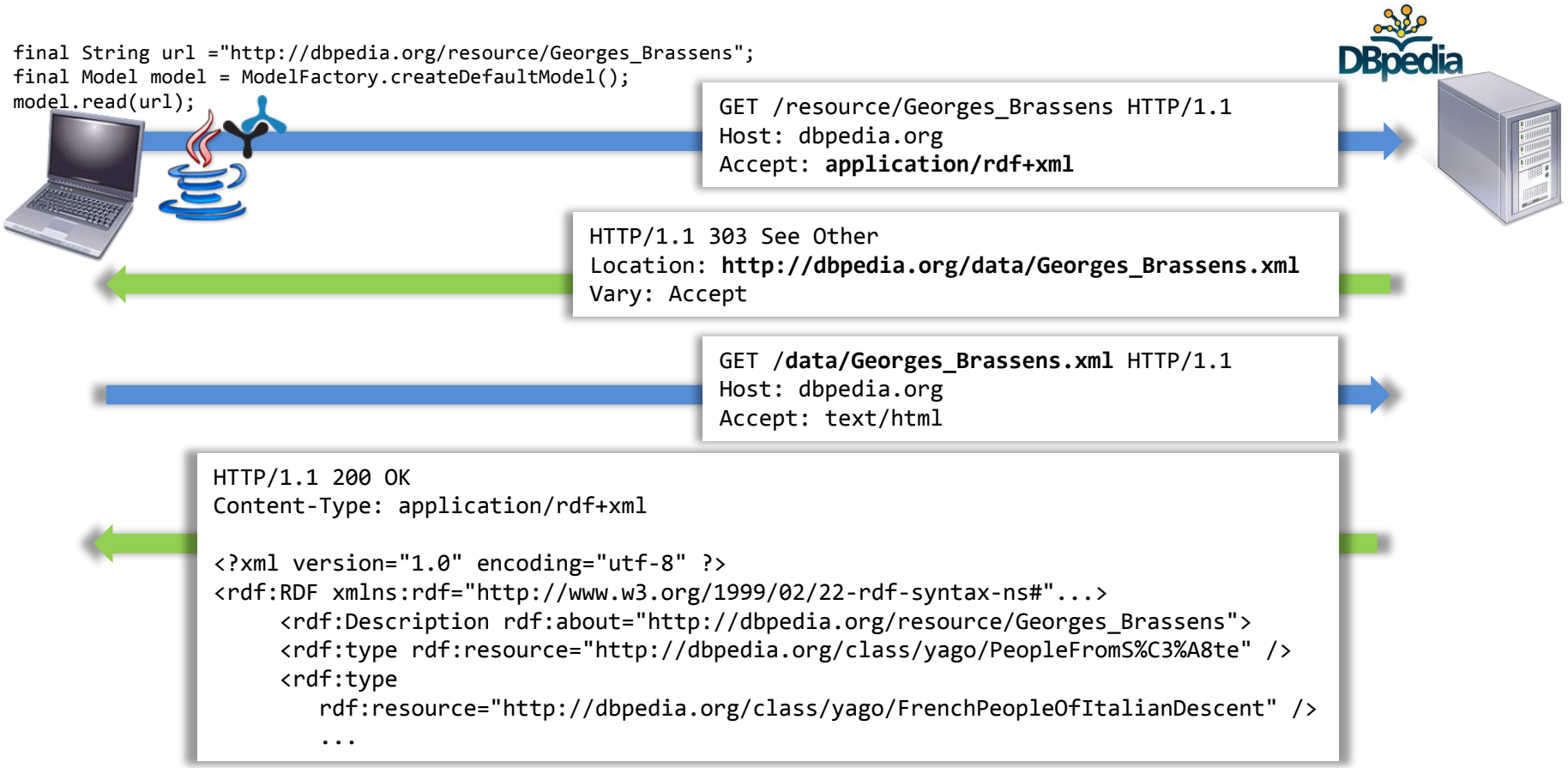
```
GET /page/Georges_Brassens HTTP/1.1
Host: dbpedia.org
Accept: text/html
```




```
HTTP/1.1 200 OK
Content-Type: text/html

<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE html>
   <head>
      <title>About: Georges Brassens</title>
      ...
   </head>
   <body onload="init();">
      <div id="header">
      <h1 id="title">About: Georges Brassens</h1>
       ...
```

Making URIs Dereferenceable
Content negotiation (303 URIs)

- Contents negotiation uses HTTP headers to a retrieve the resource description



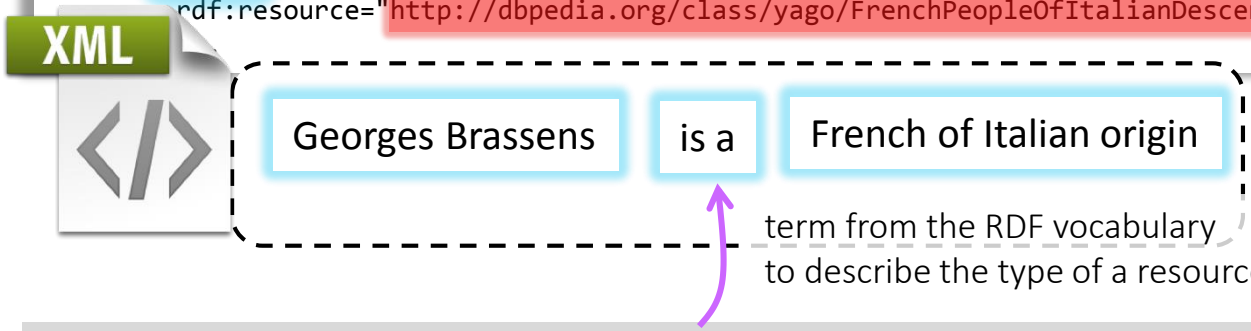
Making URIs Dereferenceable
Hash URIs

- An other way to identify real world objects or abstract concepts without creating ambiguity with the document that contains it 's description is to use hash URIs.

Example of a hash URI used by Dbpedia RDF description of Georges Brassens

http://dbpedia.org/data/Georges_Brassens.xml

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

<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"...>
     <rdf:Description rdf:about="http://dbpedia.org/resource/Georges_Brassens">
     <rdf:type rdf:resource="http://dbpedia.org/class/yago/PeopleFromS%C3%A8te" />
     <rdf:type
          rdf:resource="http://dbpedia.org/class/yago/FrenchPeopleOfItalianDescent" />
```



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

base | fragment identifier

Making URIs Dereferenceable
Hash URIs

When a client wants to retrieve a hash URI, the HTTP protocol requires the fragment part to be stripped off before requesting the URI from the server

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

URL the RDF/XML document containing a description of the whole RDF vocabulary

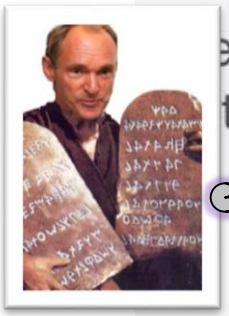

```
HTTP/1.1 200 OK
Content-Type: application/rdf+xml

<?xml version="1.0"?>
<rdf:RDF   xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
   xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
   xmlns:owl="http://www.w3.org/2002/07/owl#"
   xmlns:dc="http://purl.org/dc/elements/1.1/">

...
<rdf:Property rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#type">
  <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
  <rdfs:label>type</rdfs:label>
  <rdfs:comment>The subject is an instance of a class.</rdfs:comment>
  <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  <rdfs:domain rdf:resource="http://www.w3.org/2000/01/rdfschema#Resource"/>
</rdf:Property>
...
<rdfs:Class rdf:about="http://www.w3.org/1999/02/22-rdf-syntax-ns#Bag">
     <rdfs:isDefinedBy rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#"/>
     <rdfs:label>Bag</rdfs:label>
     <rdfs:comment>The class of unordered containers.</rdfs:comment>
```

Client is in charge of extracting the information associated to the fragment

Outline

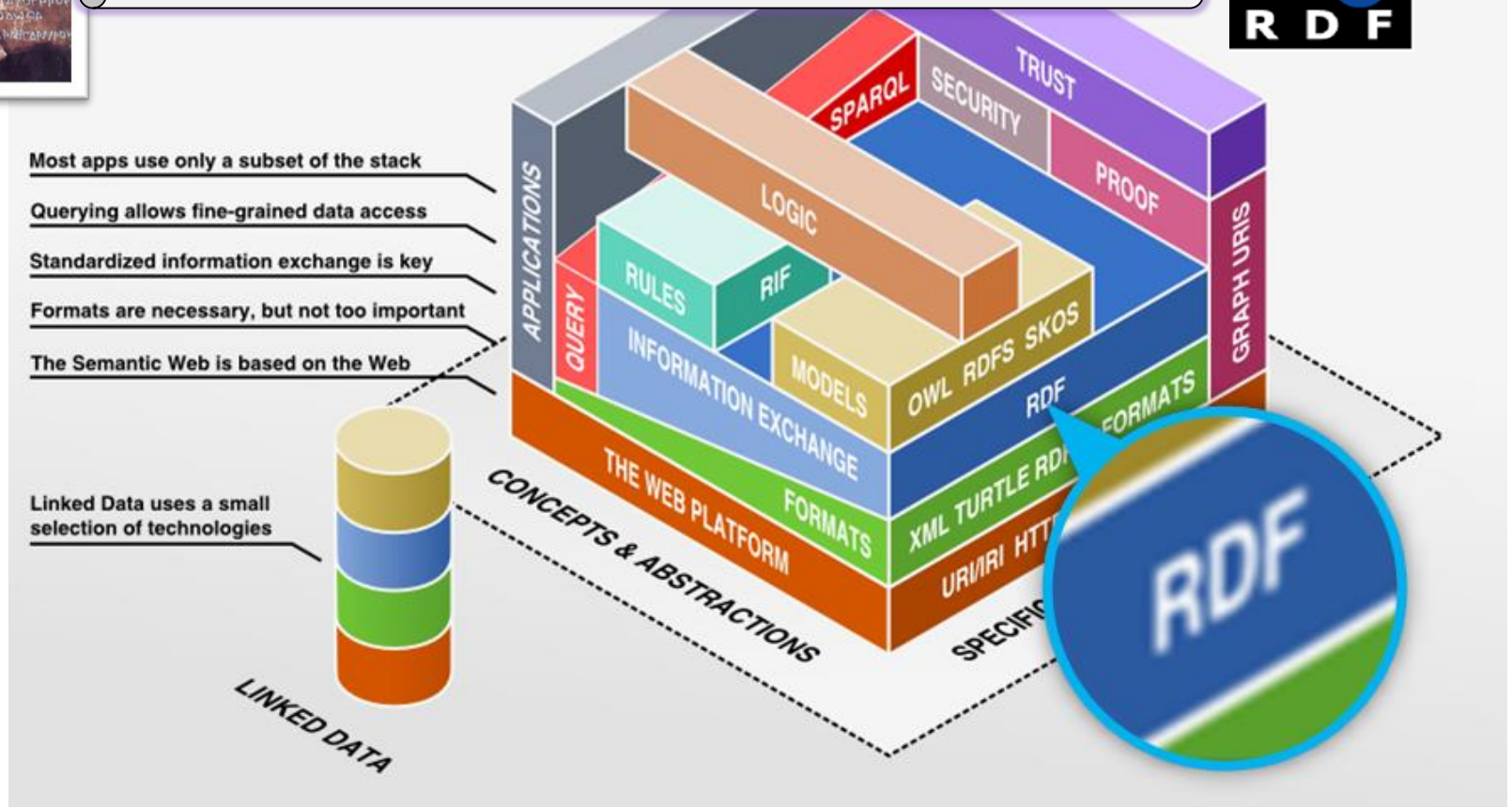
- "Theoretical" Session (morning)
  - Introduction
  - Distributing Data on the web with RDF
    - Naming the Data : URIs (Uniform Resources Identifiers)
    - The RDF Data model
  - Querying Linked Data with SPARQL
  - Semantic modelling
    - RDFS
    - OWL
  - From Open Data to Linked Open Data
  - Conclusion
- Hands-on session (afternoon)
  - From a CSV file to linked data
  - Querying linked data (SPARQL)

Resource Description Framework (RDF)



Linked Data: 3rd Principle

*When someone looks up a URI, provide useful information, using the standards (**RDF**, SPARQL).*



http://www.bnode.org/blog/tag/layer%20cake (Benjamin Nowack)

Resource Description Framework

- RDF Resource Description Framework
  - a framework for describing resources on the web

> "The Resource Description Framework (RDF) is a framework for representing information in the Web." [1]

  - is designed to be read and understood by computers
  - RDF is a part of the W3C's Semantic Web Activity
    - became a W3C recommendation 10. February 2004
    - Updated February 2014 (RDF 1.1)

http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/ [1]

http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/ [2]

Resource ?

- *"The Resource Description Framework (RDF) is a language for* ***representing information about resources in the World Wide Web****. It is particularly intended for representing metadata about Web resources, such as the title, author, and modification date of a Web page, copyright and licensing information about a Web document, or the availability schedule for some shared resource.* ***However, by generalizing the concept of a "Web resource", RDF can also be used to represent information about things that can be identified on the Web, even when they cannot be directly retrieved on the Web****."*

  http://www.w3.org/TR/rdf-primer/

- *"To publish data on the Web, the* ***items in a domain of interest must first be identified****. These are the things whose properties and relationships will be described in the data, and may include Web documents as well as real-world entities and abstract concepts. As Linked Data builds directly on Web architecture , the Web architecture term* ***resource*** *is used to refer* ***to these things of interest****, which are, in turn, identified by HTTP URIs."*

  Tom Heath, Christian Bizer *: Linked Data: Evolving the Web into a Global DataSpace*

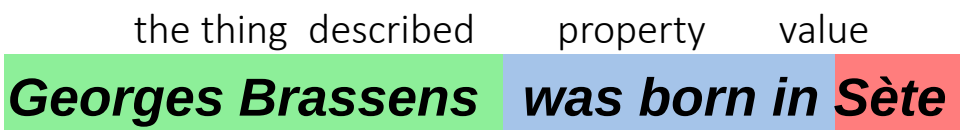
  http://linkeddatabook.com/editions/1.0/

RDF outline

- **RDF Data Model**
- RDF formats
- Blank nodes
- Typed literals
- Resources definition
- RDF an data integration
- Persisting RDF
- References

RDF Data Model

- With RDF, knowledge is represented by a set of assertions (statements)
- All RDF statements follow a simple structure composed of three parts :
  - the thing the statement describes
  - the properties of the thing the statement describes
  - the values of those properties the statement describe

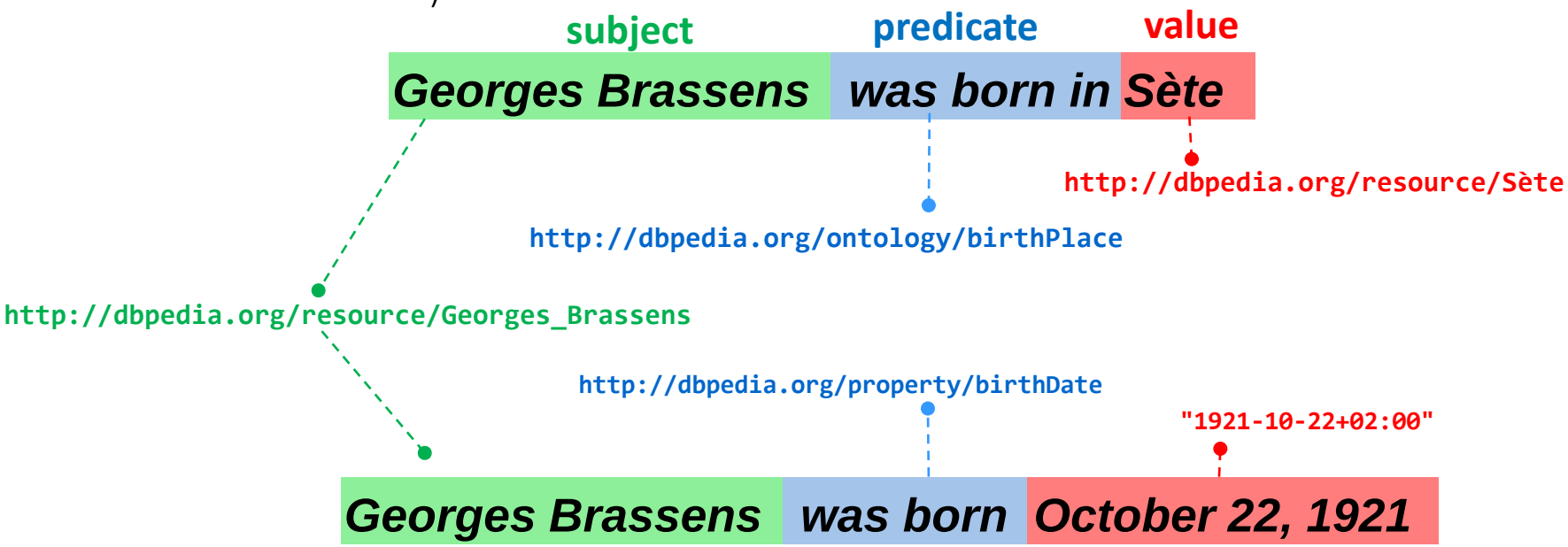


RDF Data Model

- RDF Statements are *triples*

Subject	Predicate	Object
URI	URI	URI/Literal

- the subject and the predicate are resources : RDF uses **URIs** (Universal Resource Identifiers) for **uniquely identifying** them
- object can be a **resource** (URI) or a **literal** (constants that don't have other attributes that describe them)

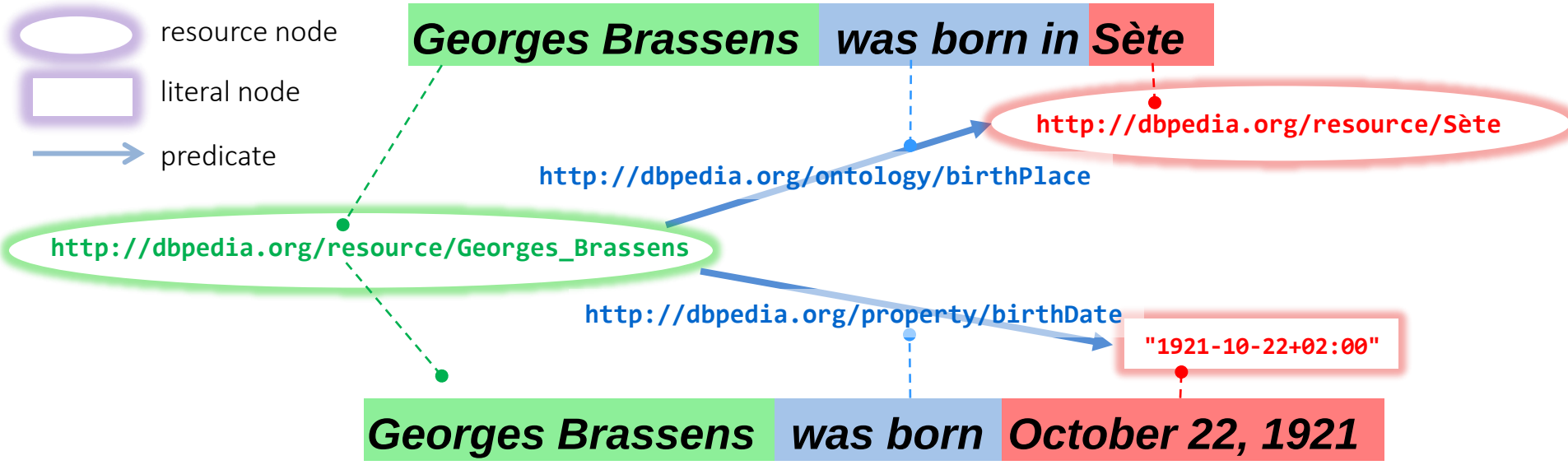


RDF Data Model

- RDF Statements are *triples*

Subject	Predicate	Object
URI	URI	URI/Literal

- RDF data can be viewed as a directed labeled graph
  - subjects and objects are nodes (vertices)
  - predicates are oriented edges (arcs)



RDF outline

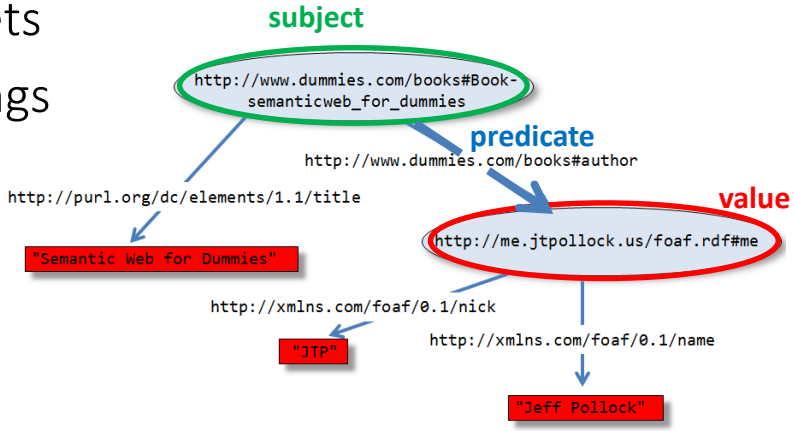
- RDF Model
- RDF formats
- Blank nodes
- Typed literals
- Resources definition
- RDF an data integration
- Persisting RDF
- References

RDF Serializations

- RDF Graphs
  - good for human analysis but unsuitable for application exchange
- RDF serialization
  - provides a way to convert between the abstract model and a concrete format (file or other byte stream)
  - several equally expressive serialization formats
    - XML/RDF (normative (standard) exchange format for serialization)
    - N-Triples
    - Turtle (Terse RDF Triple Language)
    - N3 (Notation3)
    - RDF/JSON
    - RDFa

RDF Serialization - N-Triples

- the simplest notation
  - each line of output represents a single statement followed by '.'
  - resources (subject, predicate, resource object) are expressed as absolute URI enclosed in angle brackets
  - object literals are double-quoted strings



example.nt subject

value

predicate

```
<http://www.dummies.com/books#Book-semanticweb_for_dummies>
    <http://www.dummies.com/books#author> <http://me.jtpollock.us/foaf.rdf#me>.
<http://www.dummies.com/books#Book-semanticweb_for_dummies>
    <http://purl.org/dc/elements/1.1/title> "Semantic Web for Dummies".
<http://me.jtpollock.us/foaf.rdf#me> <http://xmlns.com/foaf/0.1/name> "Jeff Pollock".
<http://me.jtpollock.us/foaf.rdf#me> <http://xmlns.com/foaf/0.1/nick> "JTP".
```

  - useful when hand-crafting data sets for application testing and debugging
  - … but **verbose** (redundant information takes additional time to transmit and parse)

RDF Serialization – N3 – Turtle

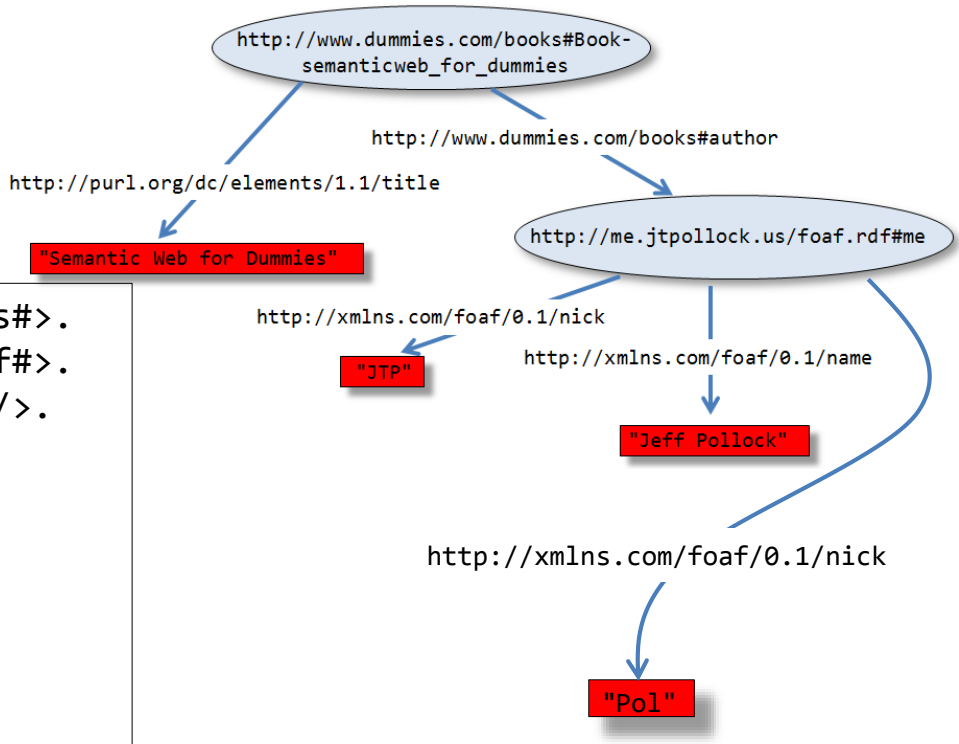
- **Notation3** (N3) more compact format than N-Triples.
  - has several absolute features that go beyond a serialization for RDF models (e.g. support for RDF-based rules).
- **Turtle** (Terse RDF Triple Language)
  - a simplified, RDF-only subset of N3.
- Both condense much of the repetitions of N-Triples
  - URIs can be shortened by using a prefix declared at the beginning of the document

```
<http://www.dummies.com/books#Book-semanticweb_for_dummies>
   <http://www.dummies.com/books#author> <http://me.jtpollock.us/foaf.rdf#me>.
```

```
@prefix swbook: <http://www.dummies.com/books#>.
@prefix jtp: <http://me.jtpollock.us/foaf.rdf#>.

swbook:Book-semanticweb_for_dummies swbook:author jtp:me.
```

RDF Serialization - N3 - Turtle

- possibility to combine multiple statements about the same subject using a semicolon (';')
- possibility to combine multiple statements involving the same subject and predicate using a coma (',')



example.ttl

```
@prefix swbook: <http://www.dummies.com/books#>.
@prefix jtp: <http://me.jtpollock.us/foaf.rdf#>.
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix foaf: <http://xmlns.com/foaf/0.1/>.

swbook:Book-semanticweb_for_dummies
     swbook:author jtp:me;
     dc:title "Semantic Web for Dummies".

jtp:me
    foaf:name "Jeff Pollock";
    foaf:nick "JTP", "Pol" .
```

RDF Serializations : RDF/XML

- RDF/XML , 1rst syntax standardized by W3C (2004)
- Statements about a resource are grouped in a **`rdf:Description`** element
- general form

gives the subject of all statement within the description

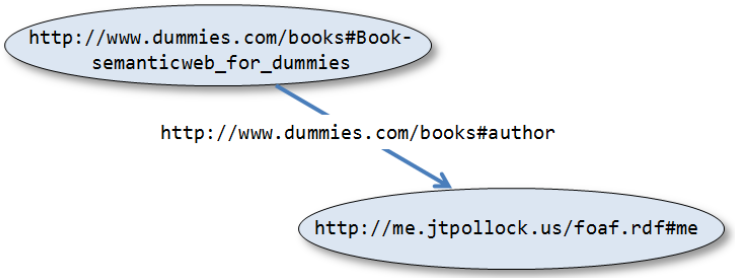
```
<rdf:Description rdf:about="subjectURI">
        <predicate rdf:resource="objectURI"/>
        <predicate>literal value</predicate>
</rdf:Description>
```

name of the internal tags represent a predicate

the object is represented differently depending on whether it is a resource or a literal

RDF Serializations : RDF/XML

- Example



XML declaration (states that's an XML document)

XML namespaces

opening and closing root tag

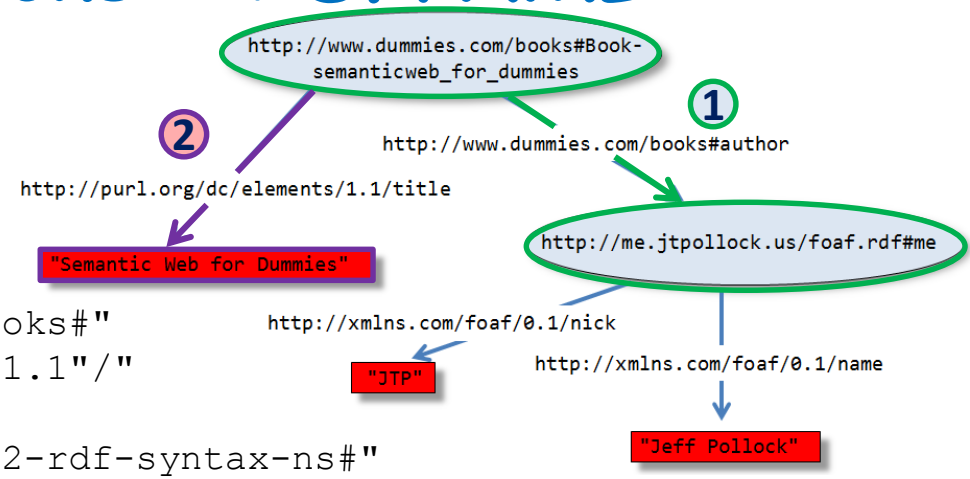
```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:books="http://www.dummies.com/books#">
    <rdf:Description
       rdf:about="http://www.dummies.com/books#Book-semanticweb_for_dummies">
       <books:author
            rdf:resource="http://me.jtpollock.us/foaf.rdf#me" />
    </rdf:Description>
</rdf:RDF>
```

Resources URIs

rdf:DescriptionElement for statements about a resource

RDF Serializations : RDF/XML

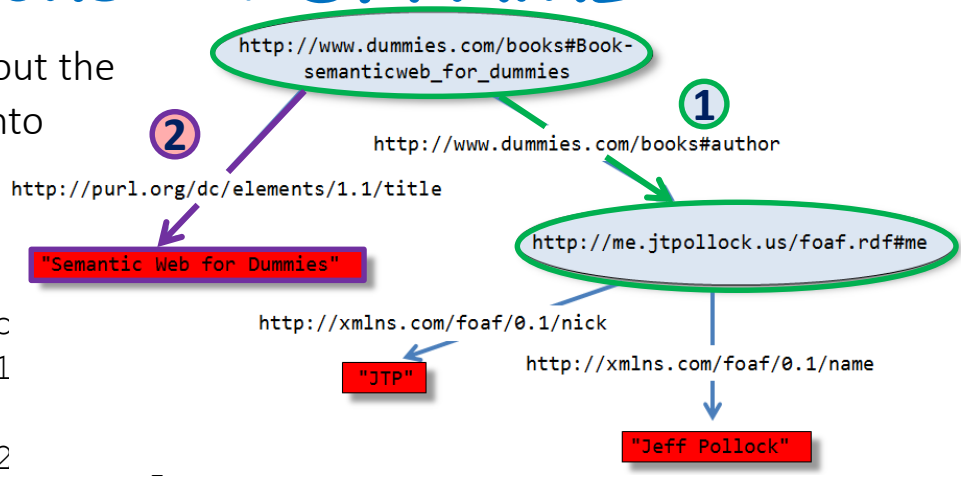
- Example




```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:books="http://www.dummies.com/books#"
    xmlns:dc="http://purl.org/dc/elements/1.1"/"
    xmlns:foaf="http://xmlns.com/foaf"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
    1 <rdf:Description
        rdf:about="http://www.dummies.com/books#Book-semanticweb_for_dummies">
        <books:author
            rdf:resource="http://me.jtpollock.us/foaf.rdf#me" />
      </rdf:Description>
    2 <rdf:Description
        rdf:about="http://www.dummies.com/books#Book-semanticweb_for_dummies">
        <dc:title>Semantic Web for Dumies<dc:title>
      </rdf:Description>

      <rdf:Description
        rdf:about="http://me.jtpollock.us/foaf.rdf#me">
        ...
      </rdf:Description>
      ...
</rdf:RDF>
```

RDF Serializations : RDF/XML

- **simplification:** when multiple descriptions about the same resource possibility to regroup them into one `rdf:Description` element




```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:books="http://www.dummies.com/boc
    xmlns:dc="http://purl.org/dc/elements/1
    xmlns:foaf="http://xmlns.com/foaf"
    xmlns:rdf="http://www.w3.org/1999/02/22
>
```

```
<rdf:Description
   rdf:about="http://www.dummies.com/books#Book-semanticweb_for_dummies">
   <books:author                                                    1
        rdf:resource="http://me.jtpollock.us/foaf.rdf#me" />
   <dc:title>Semantic Web for Dumies<dc:title>  2
</rdf:Description>
```

```
<rdf:Description
   rdf:about="http://me.jtpollock.us/foaf.rdf#me">
   ...
</rdf:Description>
...
</rdf:RDF>
```

RDF Serializations : RDF/XML

- simplification: use XML (DTD) entities to simplify URI attributes.

```
<?xml version="1.0"?>
<rdf:RDF
   xmlns:books="http://wwwdummies.com/books#"
   xmlns:dc="http://purl.org/dc/elements/1.1"/"
   xmlns:foaf="http://xmlns.com/foaf"
   xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>

      <rdf:Description
         rdf:about="http://www.dummies.com/books#Book-semanticweb_for_dummies">
         <books:author
              rdf:resource="http://me.jtpollock.us/foaf.rdf#me" />
         <dc:title>Semantic Web for Dumies<dc:title>
      </rdf:Description>

       <rdf:Description
          rdf:about="http://me.jtpollock.us/foaf.rdf#me">
          ...
       </rdf:Description>
       ...
</rdf:RDF>
```

RDF Serializations : RDF/XML

- simplification: use XML (DTD) entities to simplify URI attributes.

http://www.w3schools.com/dtd/default.asp

- a **DTD** (Document Type Definition) defines the legal building blocks of an XML document → the document structure with a list of legal elements and attributes.

```
<!DOCTYPE NEWSPAPER [

<!ELEMENT NEWSPAPER (ARTICLE+)>
<!ELEMENT ARTICLE (HEADLINE,BYLINE,LEAD,BODY,NOTES)>
<!ELEMENT HEADLINE (#PCDATA)>
<!ELEMENT BYLINE (#PCDATA)>
<!ELEMENT LEAD (#PCDATA)>
<!ELEMENT BODY (#PCDATA)>
<!ELEMENT NOTES (#PCDATA)>

<!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>
<!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>
<!ATTLIST ARTICLE DATE CDATA #IMPLIED>
<!ATTLIST ARTICLE EDITION CDATA #IMPLIED>

]>
```

- **Entities :** variables used to define shortcuts to standard text or special characters.

entity declaration

```
<!ENTITY entity-name "entity-value">
```

entity usage

```
&entity-name;
```

example

```
<!ENTITY writer "Donald Duck.">
<!ENTITY copyright "Copyright W3Schools.">

<author>&writer;&copyright;</author>
```

RDF Serializations : RDF/XML

- simplification: use XML (DTD) entities to simplify URI attributes.

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF
   [ <!ENTITY books "http://www.dummies.com/books#">
     <!ENTITY myfoaf "http://me.jtpollock.us/foaf.rdf#" >
   ]
>
<rdf:RDF
    xmlns:books="http://www.dummies.com/books#"
    xmlns:dc="http://purl.org/dc/elements/1.1"/"
    xmlns:foaf="http://xmlns.com/foaf"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
        <rdf:Description
           rdf:about="&books;Book-semanticweb_for_dummies">
           <books:author
                rdf:resource="&myfoaf;me">
           <dc:title>Semantic Web for Dumies<dc:title>
        </rdf:Description>
        <rdf:Description
           rdf:about="&myfoaf;me">
           ...
        </rdf:Description>
        ...
</rdf:RDF>
```

RDF outline

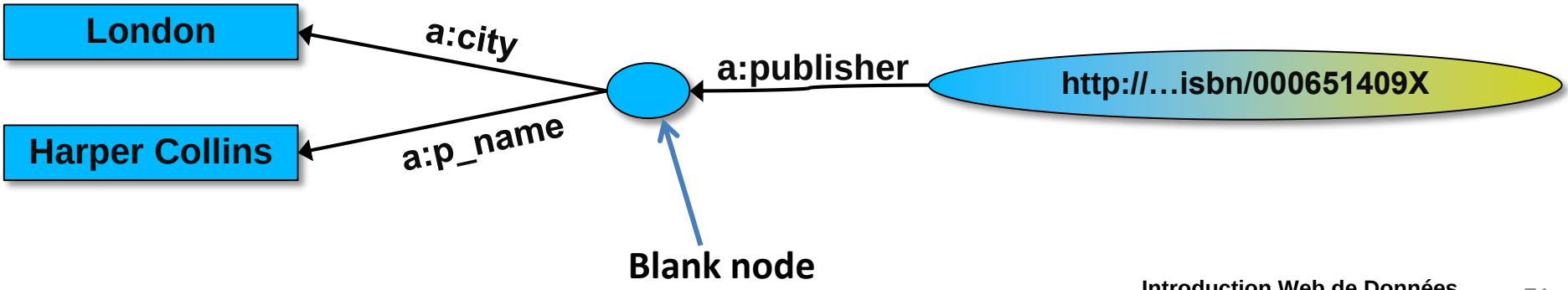
- RDF Model
- RDF formats
- Blank nodes
- Typed literals
- Resources definition
- RDF an data integration
- Persisting RDF
- References

RDF Blank nodes

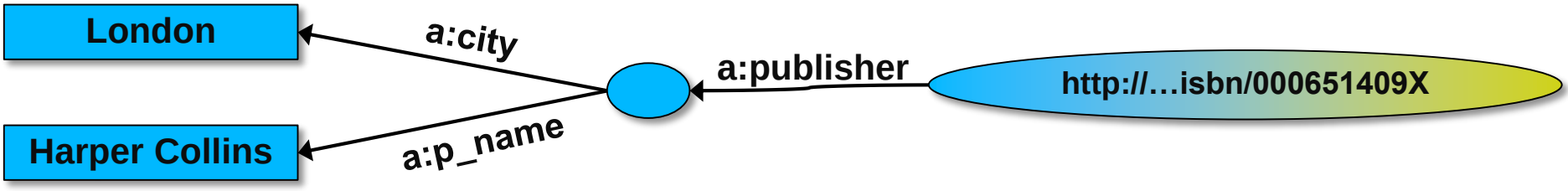
- there are some situations where you don't know the URI of the thing you would like to reference or there is no identifier available
- but not having a URI for the item doesn't means you can't talk about it

→ RDF provides anonymous (blank) nodes

Consider the following statement: "the publisher is a «thing» that has a name and an address" … what is the URI of «thing»?

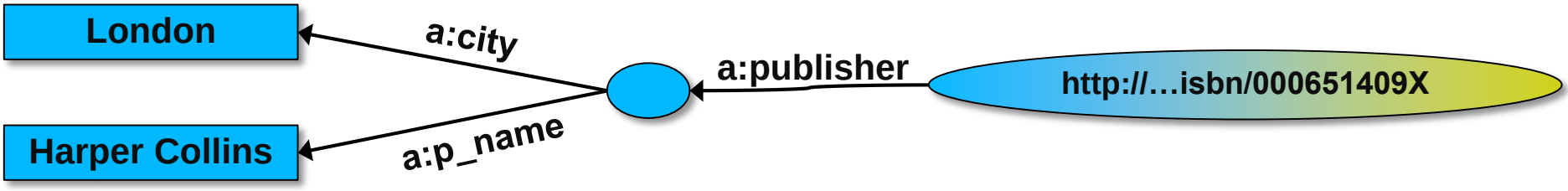


Blank nodes - RDF/XML




```
<rdf:Description rdf:about="http://…/isbn/000651409X">
   <a:publisher rdf:nodeID="A234"/>
</rdf:Description>
<rdf:Description rdf:nodeID="A234">
   <a:p_name>HarpersCollins</a:p_name>
   <a:city>London</a:city>
</rdf:Description>
```

Blank nodes - N3 - Turtle



dereferenceable blank node

```
<http://…/isbn/2020386682> a:publisher _:A234.
_:A234 a:p_name "HarpersCollins";
       a:city  "London".
```

Anonymous blank node

```
<http://…/isbn/000651409X> a:publisher [
    a:p_name "HarpersCollins";
    a:city   "London".
].
```

RDF outline

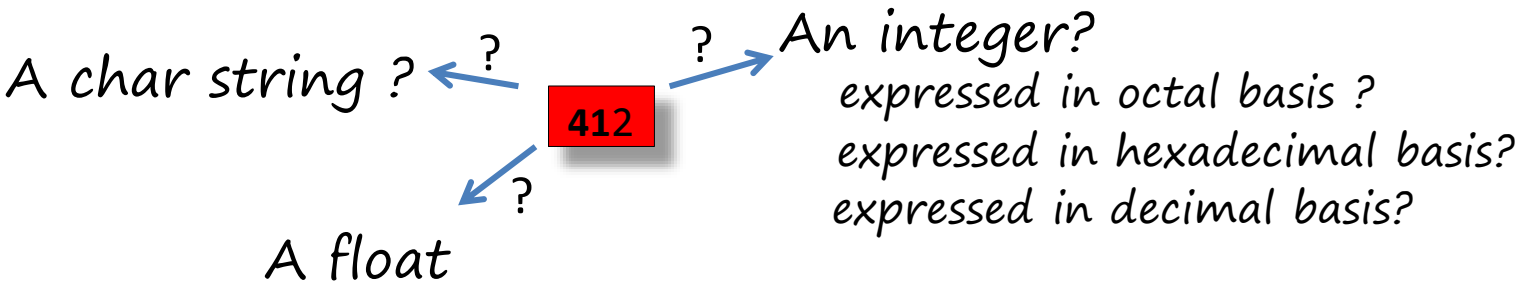
- RDF Model
- RDF formats
- Blank nodes
- Typed literals
- Resources definition
- RDF an data integration
- Persisting RDF
- References

Typed Literals

- Literal are not resources : they are values

When looking at that description a human can easily realize that 412 is an integer.

But what about a computer program ?



You must provide **some context** if you intend to use the value in any other way than to just view it on a web page → **typed literals**

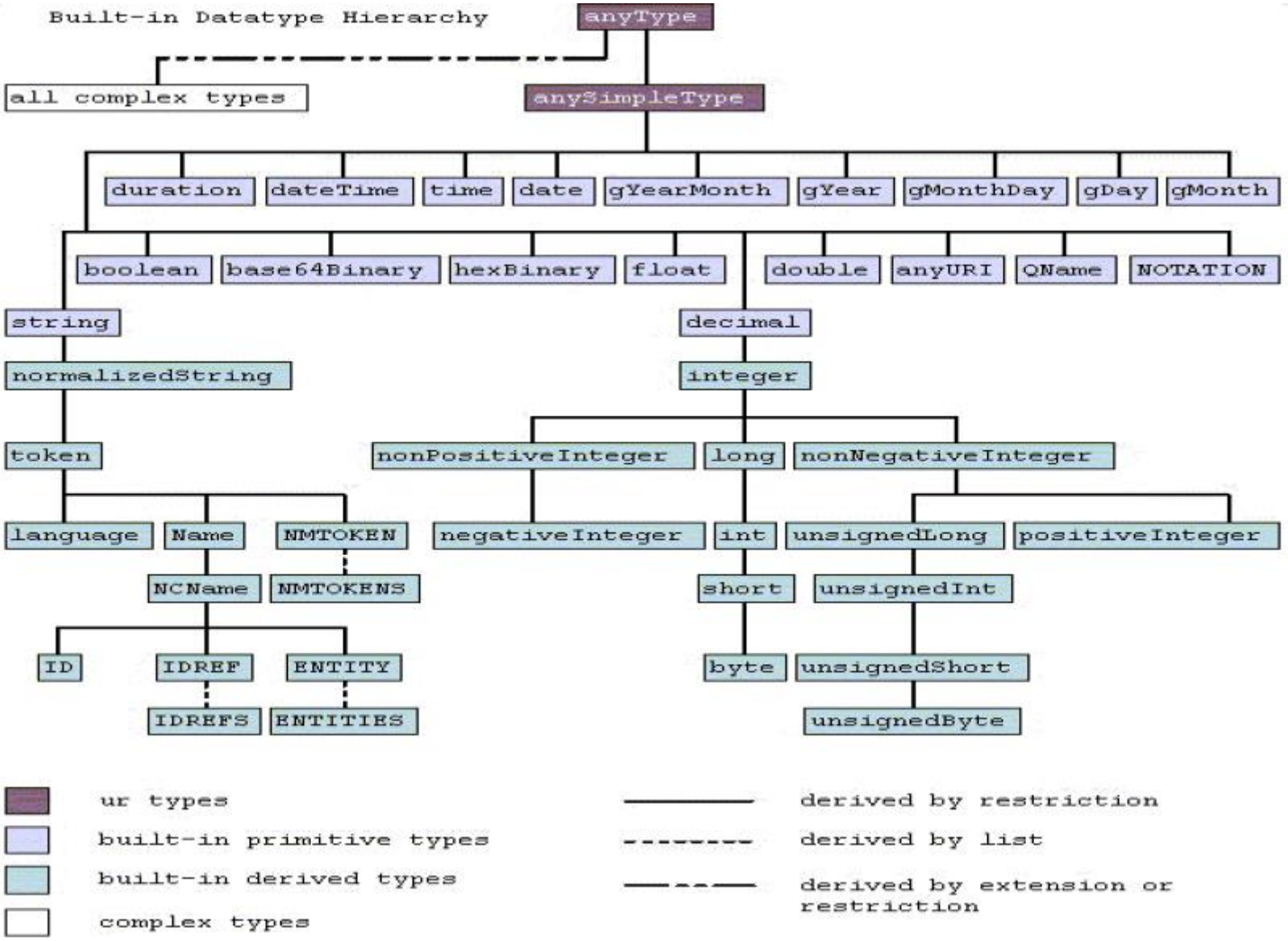
Typed Literals

- How to define a type (datatype) ?
  - **value space** : the set of values represented by the type
    - - e.g. an integer interval , dates, ….
  - **lexical space** : the set of char strings defining the representations of the values
    - eg. dates : `yy-mm-dd` or `dd-mm-yy`
  - a **mapping** between the lexical space and the value space
    - associating a concrete value with each eligible literal

Typed Literals : XSD

- To associate type to literals RDF uses XSD (**X**ML **S**chema **D**efinition)
  - W3C recommendations :
    - http://www.w3.org/TR/xmlschema-2/
    - http://www.w3.org/TR/rdf-mt/
  - XSD defines a predefined datatype hierarchy (see next slide)
    - primitive types (string, float, decimal, etc.)
    - derived types (integer, long, etc.)
  - new types can be defined by derivation
    - restriction
    - lists
    - union
    - extension

Typed Literals: XSD



Typed Literals: XSD

- examples of definition of new data types

new type derived by restrictions

```
<xsd:schema ...>
  <xsd:simpleType name="humanAge">
    <xsd:restriction base="integer">
     <xsd:minInclusive value="0">
     <xsd:maxExclusive value="150">
    </xsd:restriction>
  </xsd:simpleType>
  ...
</xsd:schema>
```

the "super" type

constraints to express the restriction

new type derived by list

```
<simpleType name="listOfFloat">
  <list itemType="float"/>
</simpleType>
```

type of the list elements

+ constraints about the list
`length, maxLength, minLength`

Typed Literals : XSD

new type derived by union and extension

```
<xsd:simpleType name="fontsize">
  <xsd:union>
    <xsd:simpleType>
       <xsd:restriction  base="xsd:positiveInteger">
         <xsd:minInclusive  value="8"/>
         <xsd:maxInclusive  value="72"/>
       </xsd:restriction>
    </xsd:simpleType>
    <xsd:simpleType>
       <xsd:restriction  base="xsd:NMTOKEN">
         <xsd:enumeration  value="small"/>
         <xsd:enumeration  value="medium"/>
         <xsd:enumeration  value="large"/>
       </xsd:restriction>
    </xsd:simpleType>
  </xsd:union>
</xsd:simpleType>
```

type defined by union

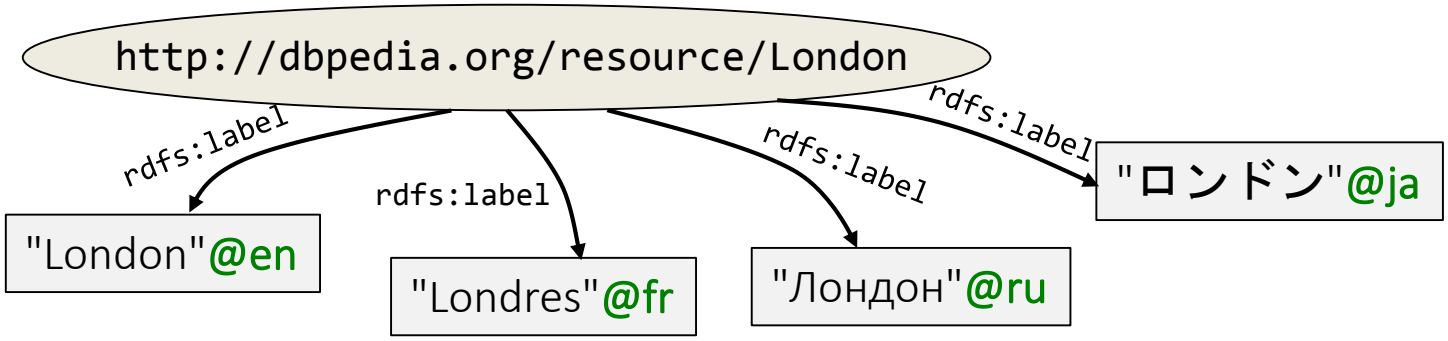
type defined by restrictions

type defined by extension

fontsize : `8-72` or `small`, `medium`, `large`

Localized textual literals

- A language can be associated to textual literals



RDF/XML

```
<rdf:RDF (…)>
<rdf:Description rdf:about="http://dbpedia.org/resource/London">
      <rdfs:label xml:lang='en'>London</rdfs:label>
      <rdfs:label xml:lang='fr'>Londres</rdfs:label>
      ...
</rdf:Description>
</rdf:RDF>
```

Turtle

```
@prefix rdfs:< http://www.w3.org/2000/01/rdf-schema#>.

<http://dbpedia.org/resource/London>
        rdfs:label "Londres"@fr, ... , "London"@en.
```

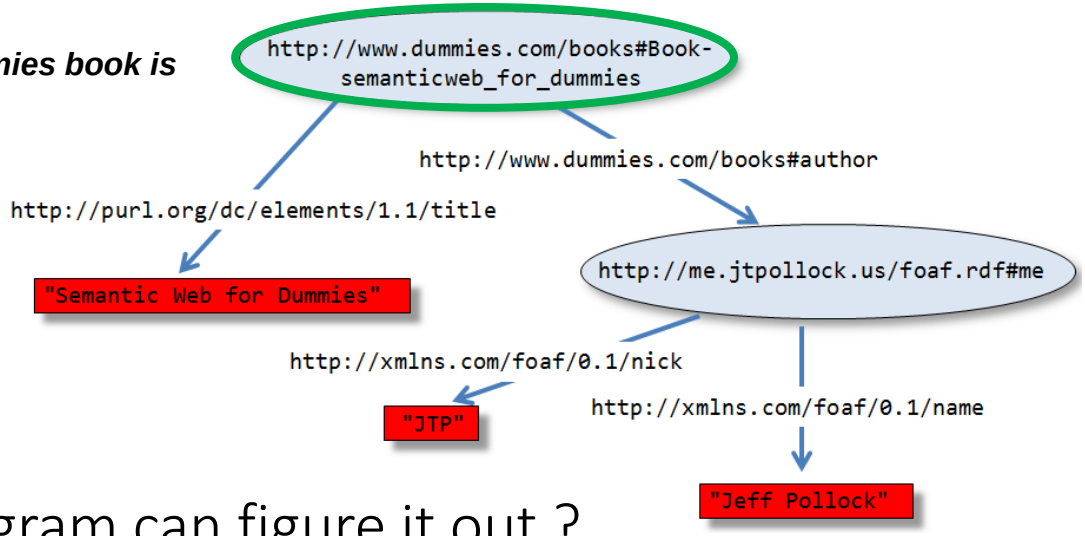
RDF outline

- RDF Model
- RDF formats
- Blank nodes
- Typed literals
- **Resources definition**
- RDF an data integration
- Persisting RDF
- References

Identifying the type of a resource

- the same way literals can be typed, it's possible to associate a type to a resource

The Semantic Web For Dummies book is authored by Jeff Pollock

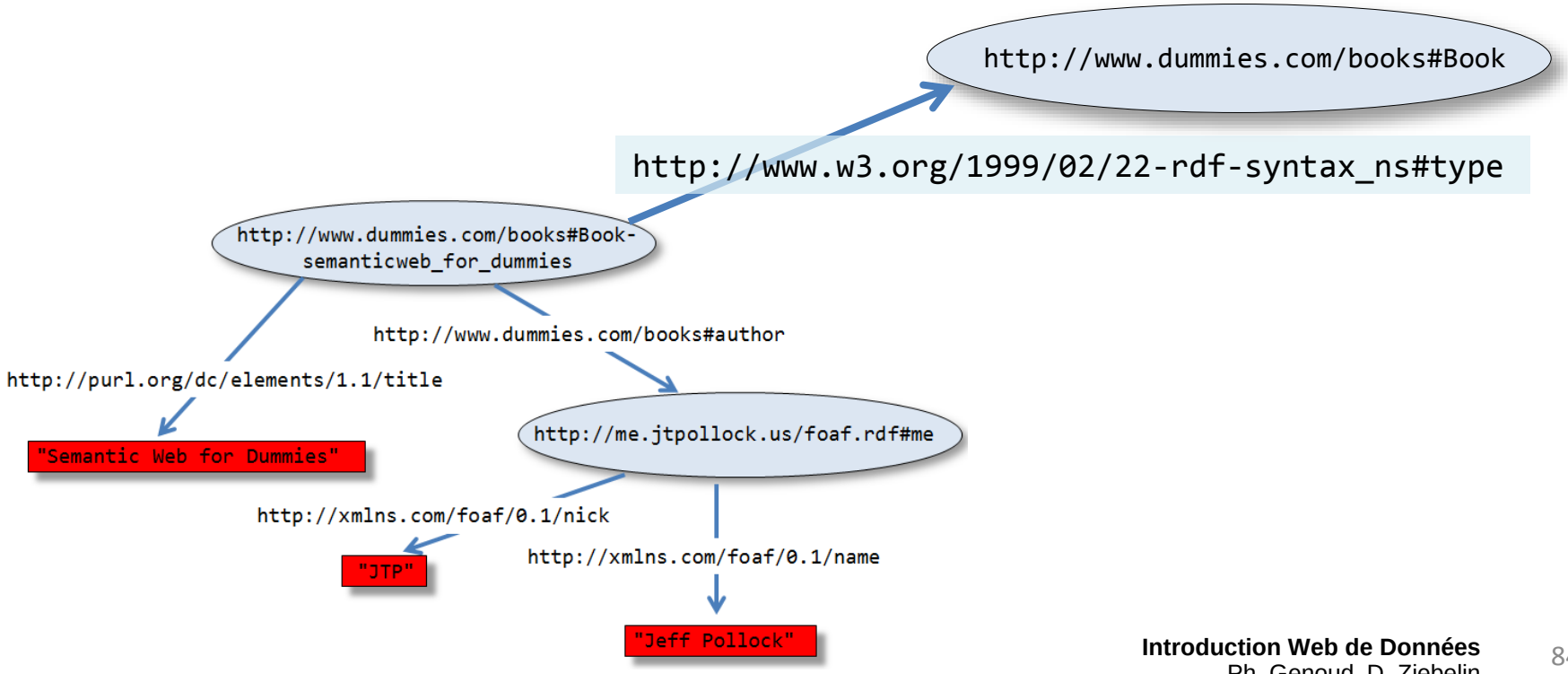


We know it's a book…
but how a computer program can figure it out ?

To solve the problem of classifying resources a way the software can understand, RDF vocabulary has a predefined predicate : **`rdf:type`**

Identifying the type of a resource

- `rdf:type` predicate's semantics
  - the value of this predicate is a resource and represents a class of things
  - the subject of this predicate is also an instance of that class



Identifying the type of a resource

- N3 - Turtle

```
@prefix swbook: <http://www.dummies.com/books#>.

swbook:Book-semanticweb_for_dummies
        swbook:author <http://me.jtpollock.us/foaf.rdf#me>;
        a swbook:Book.
```

shortcut for rdf:type

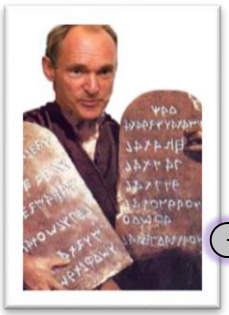
- XML/RDF

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:books="http://www.dummies.com/books#">
   <rdf:Description
      rdf:about="http://www.dummies.com/books#Book-semanticweb_for_dummies">
      <rdf:type rdf:resource="http://www.dummies.com/books#Book"/>
      <books:author
           rdf:resource="http://me.jtpollock.us/foaf.rdf#me" />
   </rdf:Description>
</rdf:RDF>
```

RDF outline

- RDF Model
- RDF formats
- Blank nodes
- Typed literals
- Resources definition
- RDF an data integration
- Persisting RDF
- References

RDF and Data Integration



Linked Data : 4th Principle

Include links to other URIs, so that they can discover more things.

DBpedia

`http://dbpedia.org/resource/grenoble`

GeoNames

`http://sws.geonames.org/6454071`

`dbpedia:Stendhal`

`dbpedia-owl:birthPlace`

`dbpedia:Grenoble`

`dbpedia-owl:elevation`

`"398"^^xsd:double`

`geonames:3014728`

`insee:38185`

INSEE

`gn:parentFeature`

`gn:name`

`geonames:6454071`

`"Grenoble"`

RDF and Data Integration

adaptation of presentations by Ivan Herman (W3C) ivan@w3.org) at
Semantic Technology Conferences 2009 et 2011
(San Jose, CA. USA, June, 2009) (San Francisco, CA. USA, June, 20
http://www.w3.org/2009/Talks/0615-SanJose-tutorial-IH/
http://www.w3.org/2011/Talks/0606-SemTech-Tut-IH/

- Dataset "A": a simplified bookstore data base

BOOKS

ID	Author	Title	Publisher	Year
ISBN 0-00-6511409-X	id_xyz	The Glass Palace	id_qpr	2000

FK

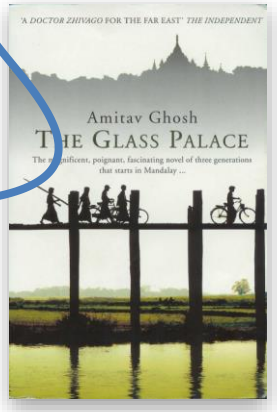
FK

AUTHORS

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

PUBLISHERS

ID	Publisher's name	City
id_qpr	Harper Collins	London



RDF and Data Integration

BOOKS

ID	Author	Title	Publisher	Year
ISBN 0-00-6511409-X	id_xyz	The Glass Palace	id_qpr	2000

FK FK

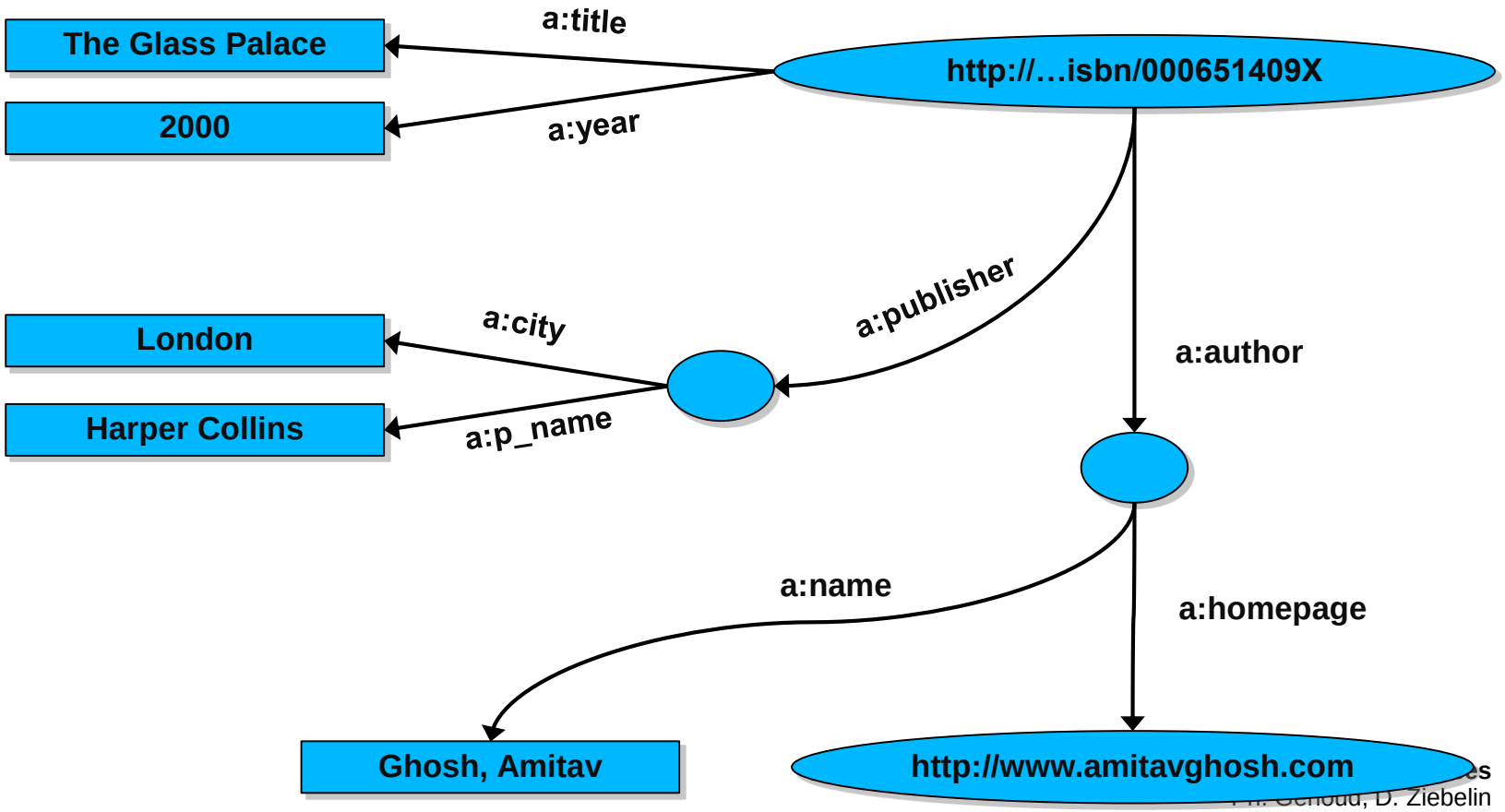
AUTHORS

ID	Name	Homepage
id_xyz	Ghosh, Amitav	http://www.amitavghosh.com

PUBLISHERS

ID	Publisher's name	City
id_qpr	Harper Collins	London

- 1st: export your data as a RDF graph



RDF and Data Integration

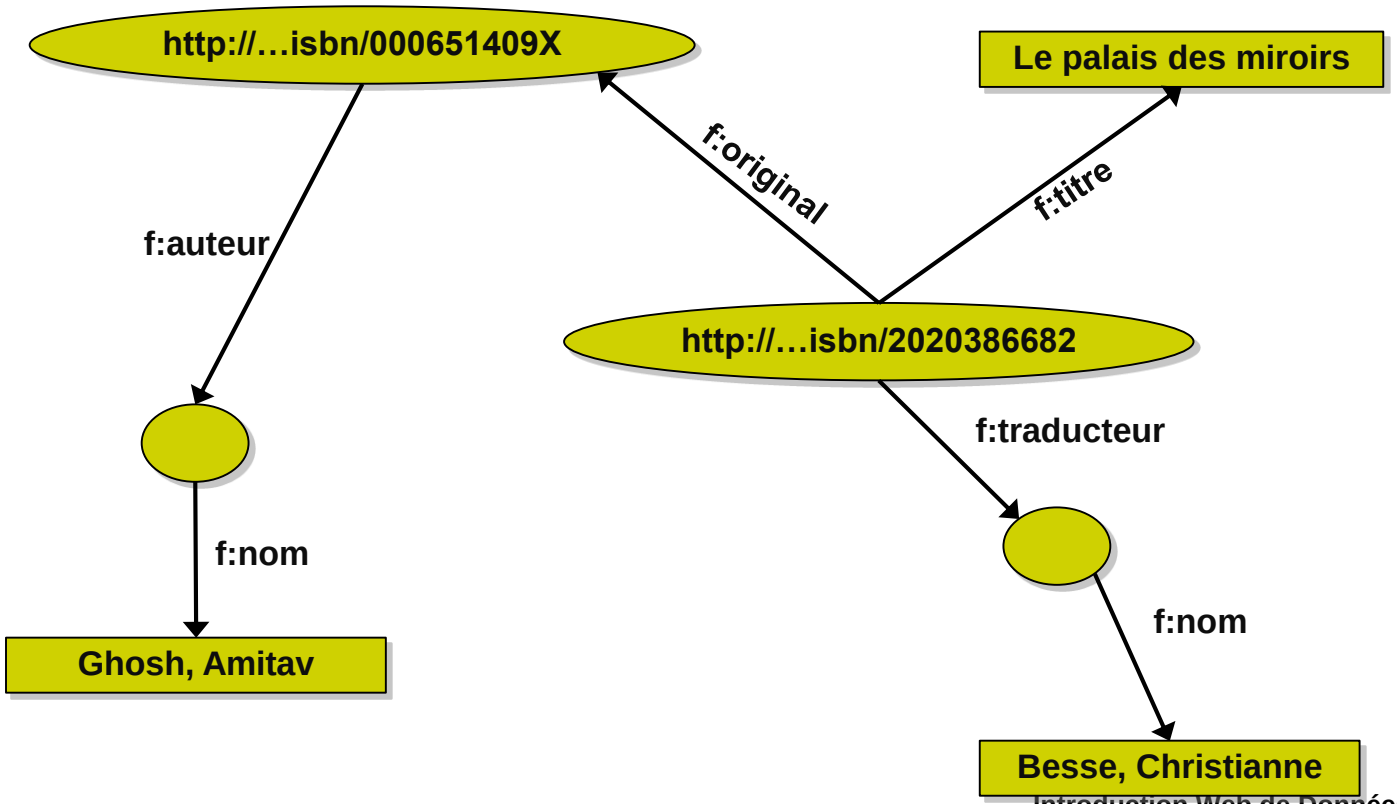
- Another dataset "F" : a google docs spreadsheet bookstore data

	A	B	C	D
1	**ID**	**Titre**	**Traducteur**	**Original**
2	ISBN 2020286682	Le Palais des Miroirs	$A12$	ISBN 0-00-6511409-X
3				
4				
5				
6	**ID**	**Auteur**		
7	ISBN 0-00-6511409-X	$A11$		
8				
9				
10	**Nom**			
11	Ghosh, Amitav			
12	Besse, Christianne			

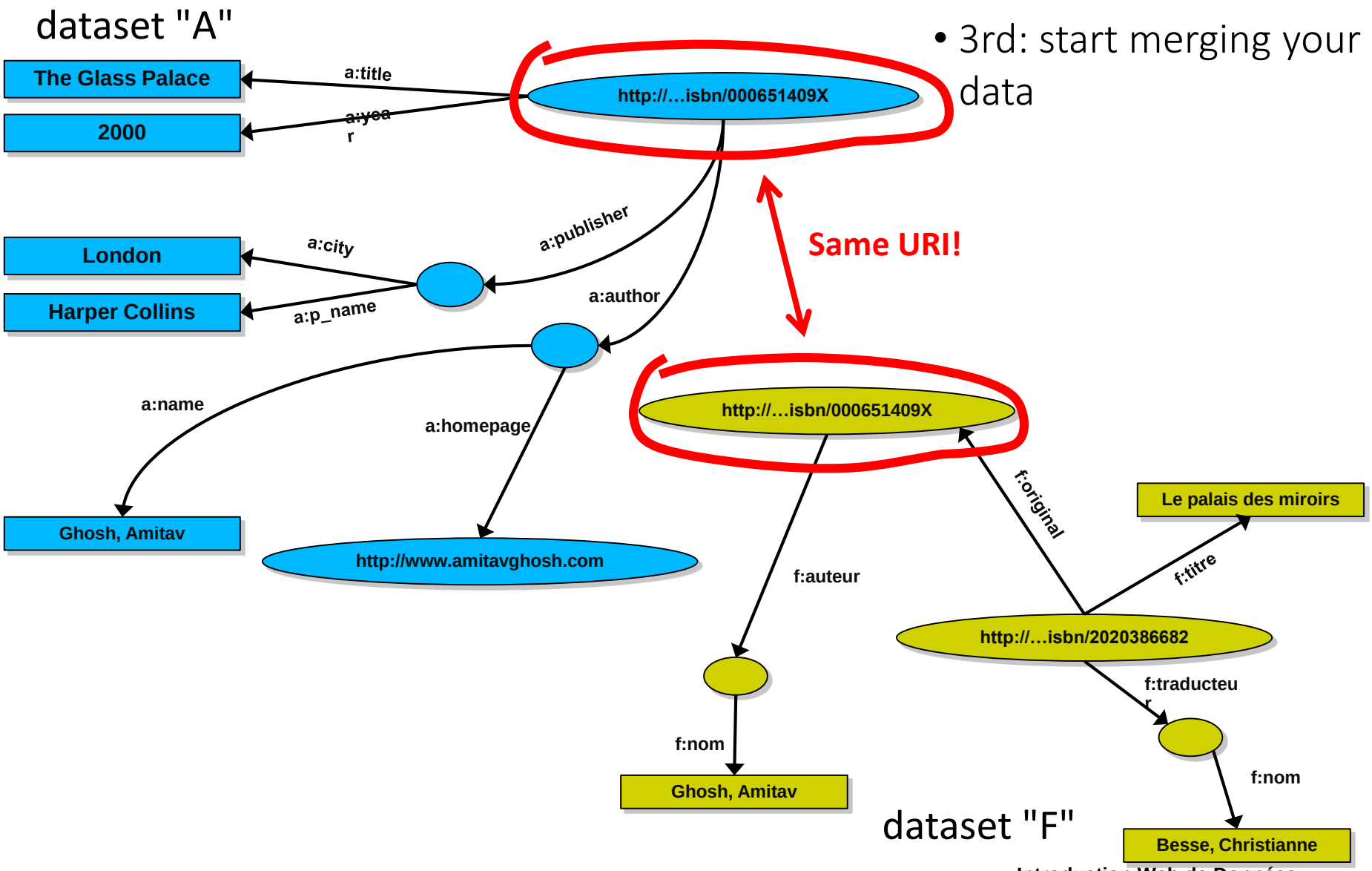
RDF and Data Integration

	A	B	C	D
1	**ID**	**Titre**	**Traducteur**	**Original**
2	ISBN 2020286682	Le Palais des Miroirs	$A12$	ISBN 0-00-6511409-X
3				
4				
5				
6	**ID**	**Auteur**		
7	ISBN 0-00-6511409-X	$A11$		
8				
9				
10	**Nom**			
11	Ghosh, Amitav			
12	Besse, Christianne			

- 2nd: export your second set of data to another RDF graph

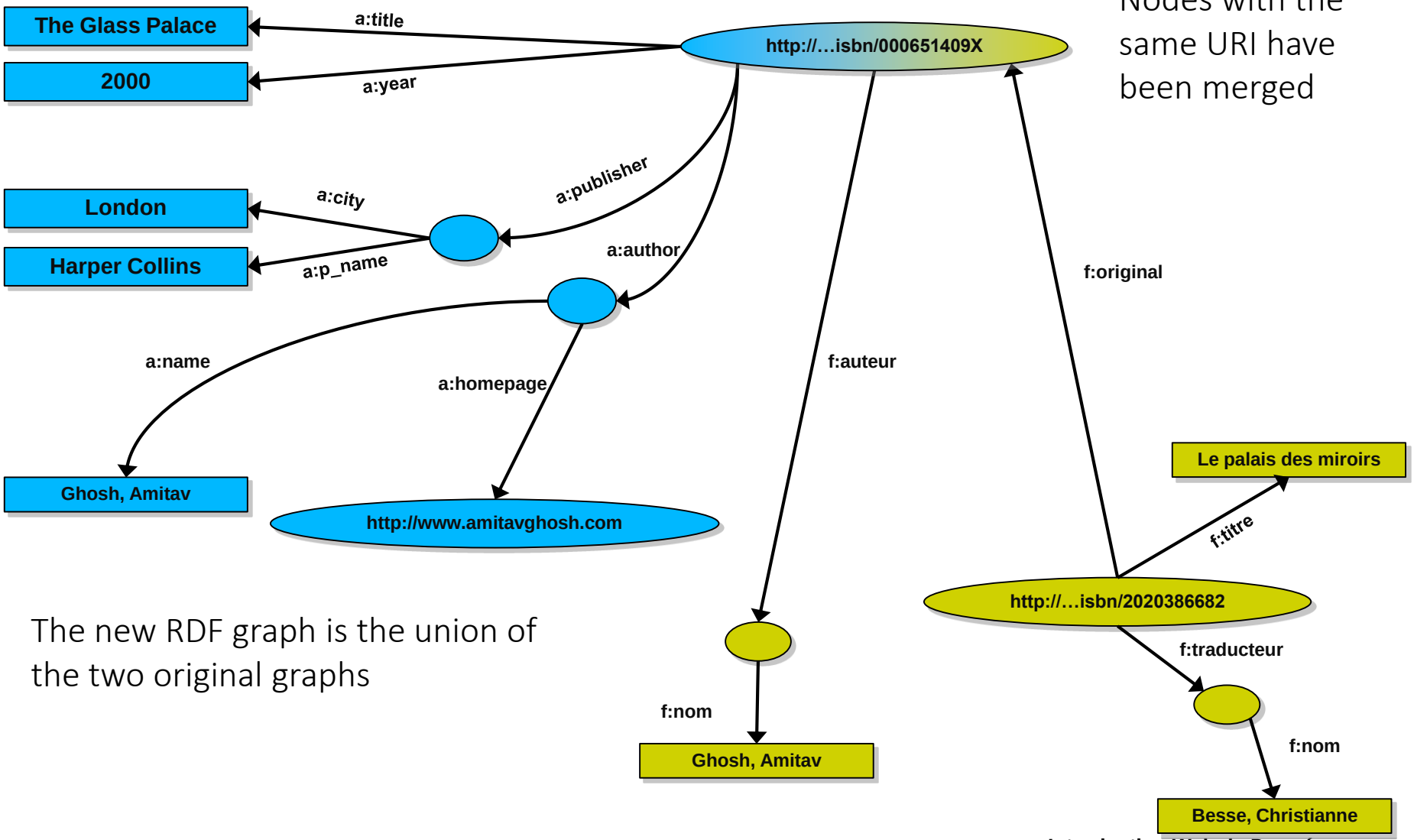


RDF and Data Integration



- 3rd: start merging your data

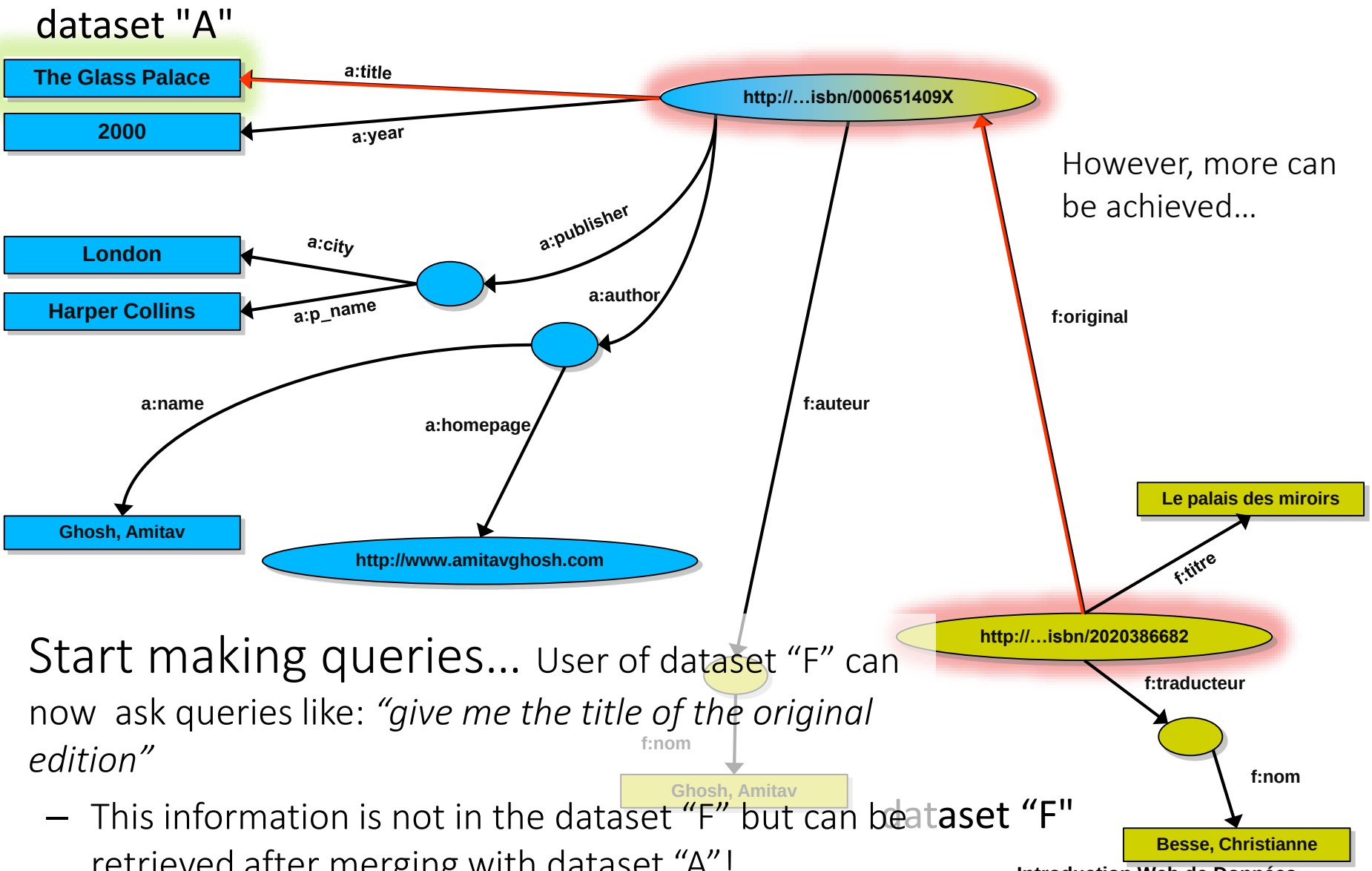
RDF and Data Integration



Nodes with the same URI have been merged

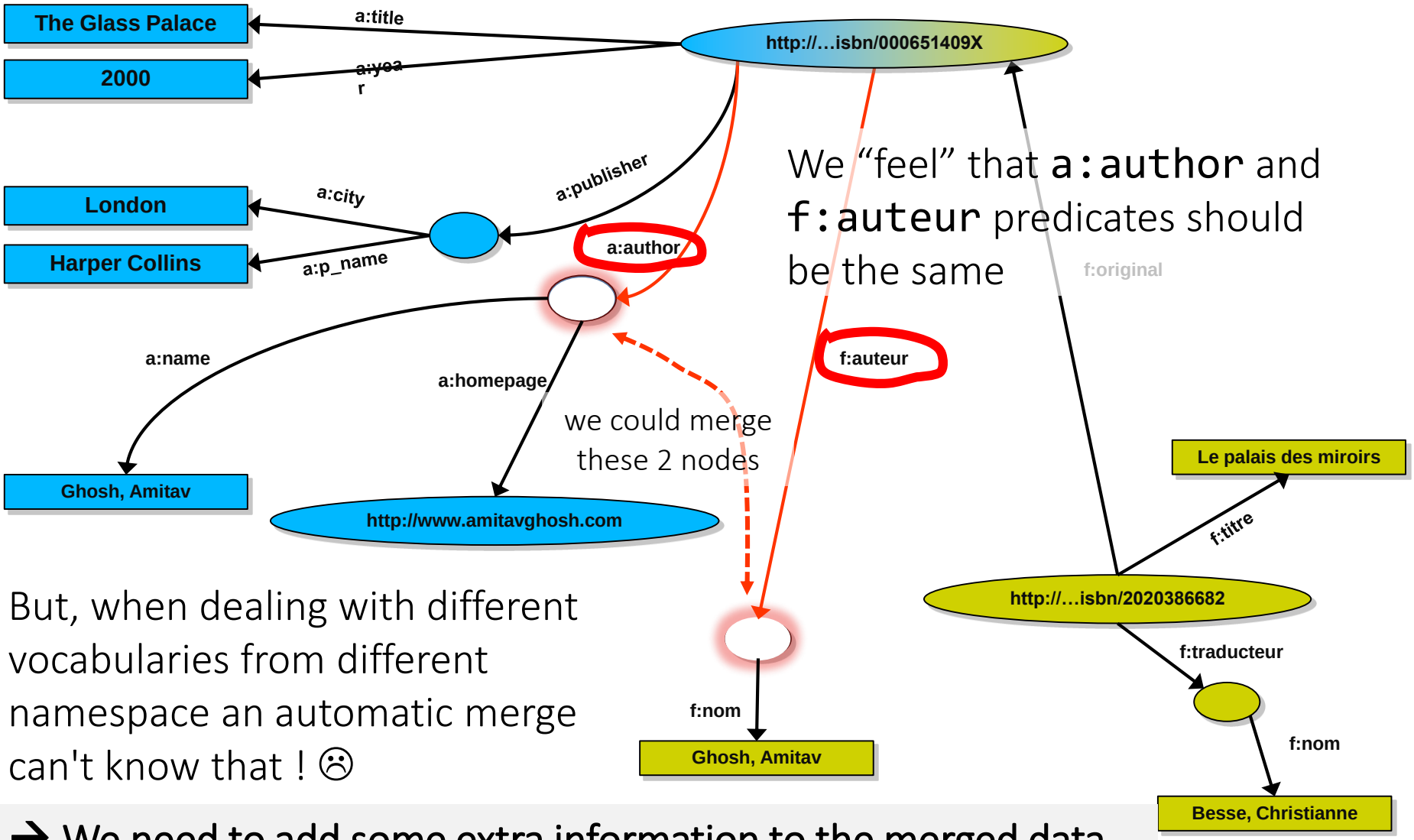
The new RDF graph is the union of the two original graphs

RDF and Data Integration



- Start making queries… User of dataset "F" can now ask queries like: *"give me the title of the original edition"*
  - This information is not in the dataset "F" but can be retrieved after merging with dataset "A"!

RDF and Data Integration



But, when dealing with different vocabularies from different namespace an automatic merge can't know that ! ☹

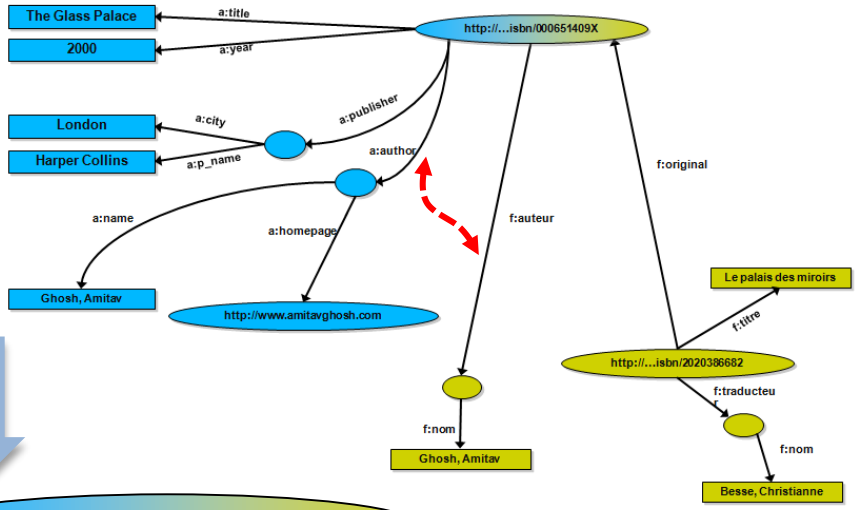
→ We need to add some extra information to the merged data

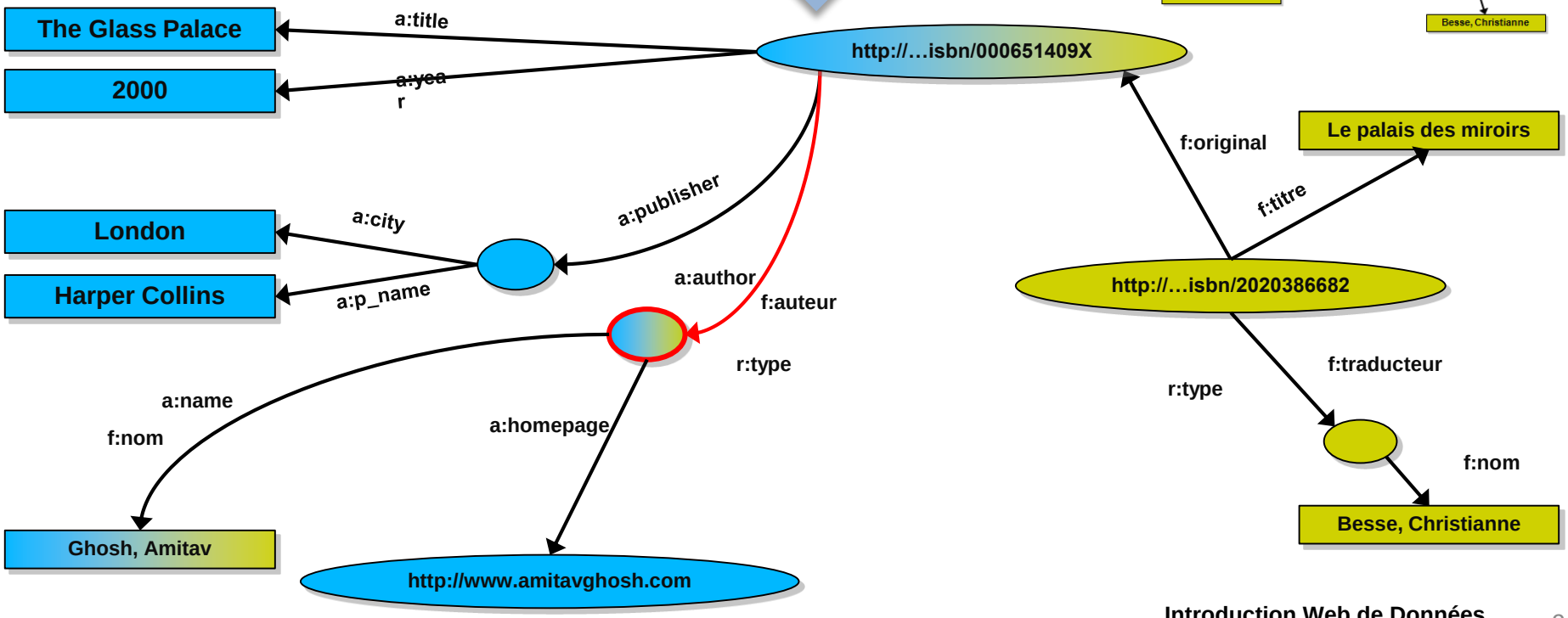
RDF and Data Integration

`a:author` and `f:auteur` are URIs identifying resources in different namespaces. We can add RDF statement about them in our RDF graph

```
a:author owl:equivalentProperty f:auteur
```

The well defined meaning (semantic) of this property allows to performs the merging



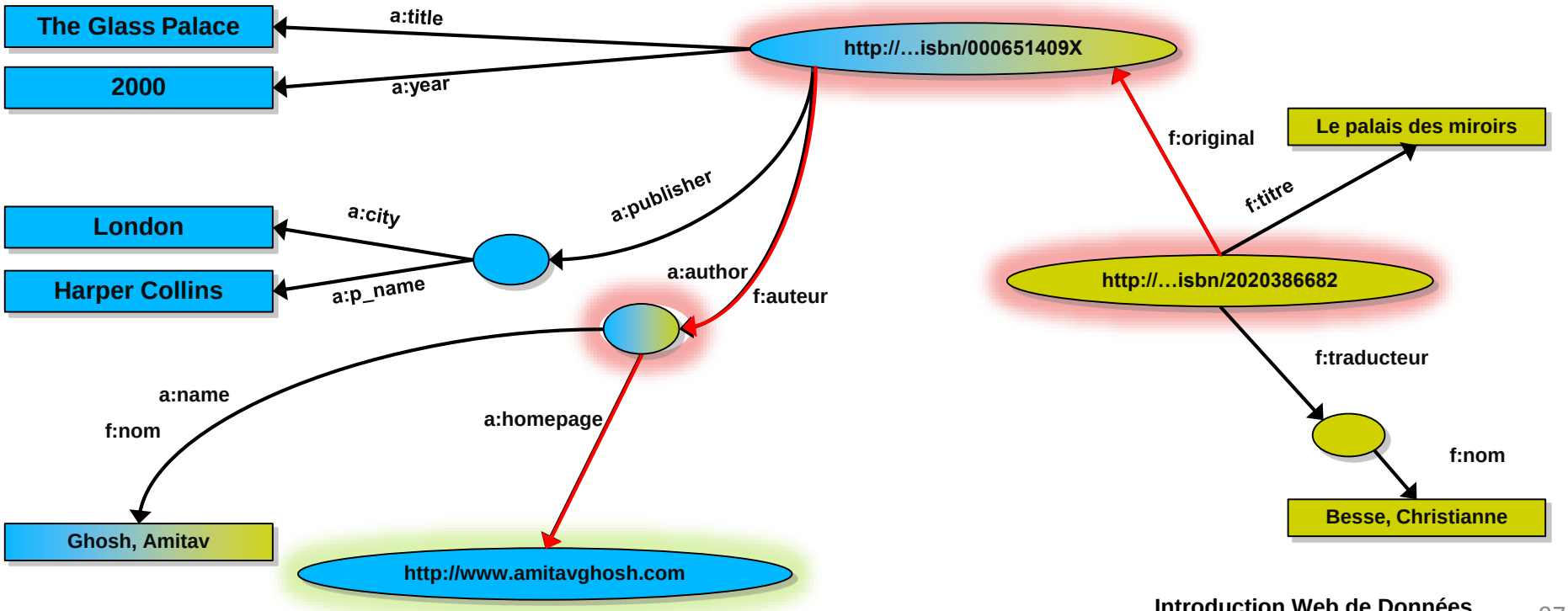


RDF and Data Integration

- By merging datasets “A” and datasets “F”
- By adding simple extra statements (owl:sameAs) as an extra “glue”
- It’s now possible to make richer queries

“donnes-moi la page d’accueil de l’auteur de l’édition originale”

“give me the home page of the original’s ‘auteur’”



Examples of Links between datasets

represents

represents

DBpedia

GeoNames

`http://dbpedia.org/resource/grenoble`

`http://sws.geonames.org/3014728`

The name (URI) that represents the city of Grenoble in DBpedia

The name (URI) that represents the city of Grenoble in GeoNames

Different URIs (URNs) in different namespaces can represents the same thing

Examples of Links between datasets

In the RDF representation of this Dbpedia resource there is a triple **that links it** to Geonames

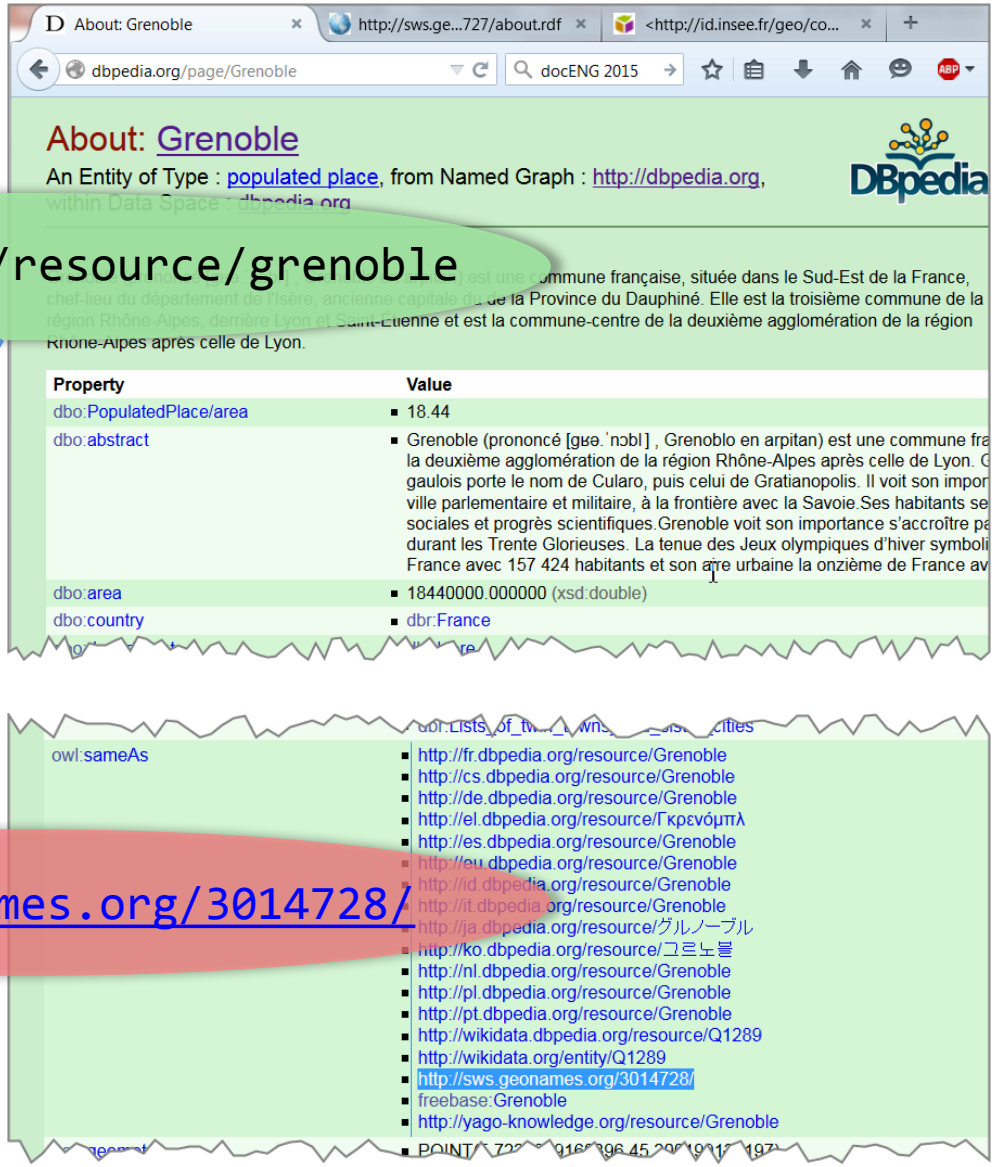
subject

`http://dbpedia.org/resource/grenoble`

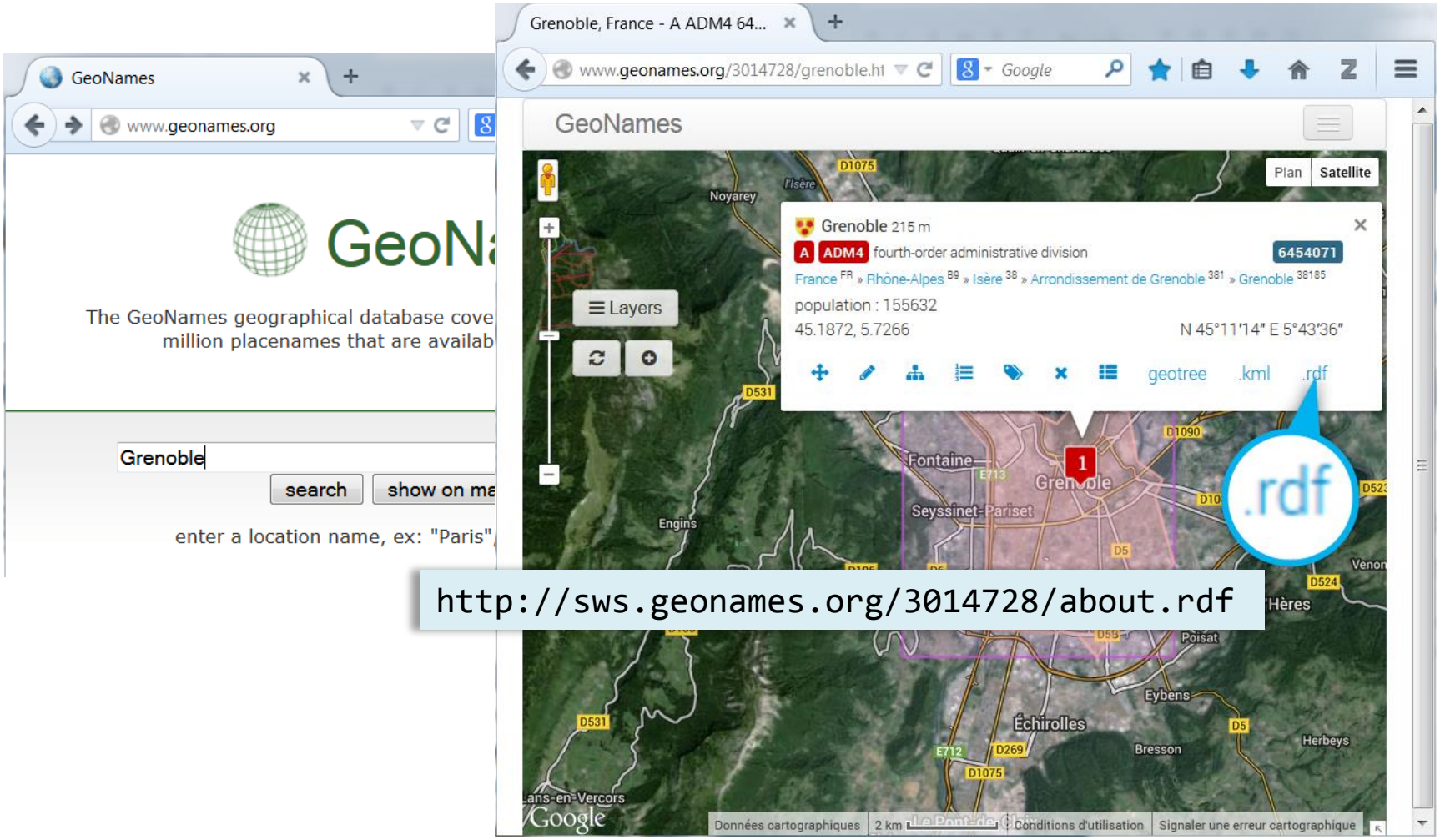
`owl:sameAs`

object

`http://sws.geonames.org/3014728/`



Examples of Links between datasets



Examples of Links between datasets

RDF description of Grenoble's parent feature in Geonames

`http://sws.geonames.org/6454071/about.rdf`

```
<rdf:RDF>
 −<gn:Feature rdf:about="http://sws.geonames.org/6
    <rdfs:isDefinedBy rdf:resource="http://sws.geonames.org/6454071/about.rdf"/>
    <gn:name>Grenoble</gn:name>
    <gn:officialName xml:lang="de">Grenoble</gn:officialName>
    <gn:officialName xml:lang="en">Grenoble</gn:officialName>
    <gn:officialName xml:lang="fr">Grenoble</gn:officialName>
    <gn:featureClass rdf:resource="http://www.geonames.org/ontology#A"/>
    <gn:featureCode rdf:resource="http://www.geonames.org/ontology#A.ADM4"/>
    <gn:countryCode>FR</gn:countryCode>
    <gn:population>155632</gn:population>
    <wgs84_pos:lat>45.1872</wgs84_pos:lat>
    <wgs84_pos:long>5.7266</wgs84_pos:long>
    <wgs84_pos:alt>215</wgs84_pos:alt>
    <gn:parentFeature rdf:resource="http://sws.geonames.org/3014727/"/>
    <gn:parentCountry rdf:resource="http://sws.geonames.org/3017382/"/>
    <gn:parentADM1 rdf:resource="http://sws.geonames.org/2983751/"/>
    <gn:parentADM2 rdf:resource="http://sws.geonames.org/3012715/"/>
    <gn:parentADM3 rdf:resource="http://sws.geonames.org/3014727/"/>
    <gn:childrenFeatures rdf:resource="http://sws.geonames.org/6454071/contains.rdf"/>
    <gn:locationMap rdf:resource="http://www.geonames.org/6454071/grenoble.html"/>
    <gn:wikipediaArticle rdf:resource="http://ru.wikipedia.org/wiki/%D0%93%D1%80%D
    %D1%8C"/>
    <owl:sameAs rdf:resource="http://id.insee.fr/geo/commune/38185"/>
  </gn:Feature>
 −<foaf:Document rdf:about="http://sws.geonames.org/6454071/about.rdf">
    <foaf:primaryTopic rdf:resource="http://sws.geonames.org/6454071/"/>
    <cc:license rdf:resource="http://creativecommons.org/licenses/by/3.0/"/>
    <cc:attributionURL rdf:resource="http://sws.geonames.org/6454071/"/>
    <cc:attributionName rdf:datatype="http://www.w3.org/2001/XMLSchema#s
    <dcterms:created rdf:datatype="http://www.w3.org/2001/XMLSchema#date"
    <dcterms:modified rdf:datatype="http://www.w3.org/2001/XMLSchema#date">2014-07-18</dcterms:modified>
  </foaf:Document>
</rdf:RDF>
```

Grenoble's parent feature URI in Geonames

`http://sws.geonames.org/6454071/`

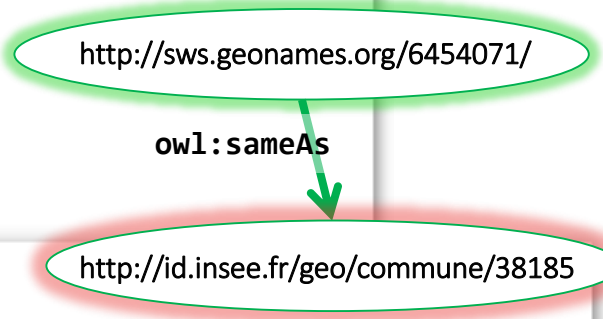
```
Préfixes:
gn:          "http://www.geonames.org/ontology#"
wgs84_pos:   "http://www.w3.org/2003/01/geo/wgs84_pos#"
PREFIX foaf: "http://xmlns.com/foaf/0.1/"
PREFIX cc:   "http://creativecommons.org/ns#"
```

Examples of Links between datasets

```
<gn:childrenFeatures rdf:resource="http://sws.geonames.org/6454071/contains.rdf"/>
<gn:locationMap rdf:resource="http://www.geonames.org/6454071/grenoble.html"/>
<gn:wikipediaArticle rdf:resource="http://ru.wikipedia.org/wiki/%D0%93%D1%80%D0%B5%D0%BD%D0%BE
%D0%B1%D0%BB%D1%8C"/>
<owl:sameAs rdf:resource="http://id.insee.fr/geo/commune/38185"/>
</gn:Feature>
<foaf:Document rdf:about="http://sw
<foaf:primaryTopic rdf:resource=
<cc:license rdf:resource="http://creativecommons.org/licenses/by/3.0/"/>
<cc:attributionURL rdf:resource="http://sws.geonames.org/6454071/"/>
<cc:attributionName rdf:datatype="http://www.w3.org
</rdf:
```

Un lien **owl:sameAs** permet de relier les deux jeux de données

Geonames resource for Grenoble



INSEE resource for Grenoble

DataLift

Institut national de la statistique et des études économiques

Description du nœud <http://id.insee.fr/geo/commune/38185>

Rechercher Texte dans Toutes les colonnes

Sujet	Prédicat	Objet
http://id.insee.fr/geo/commune/38185	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://rdf.insee.fr/def/geo#Commune
http://id.insee.fr/geo/commune/38185	http://rdf.insee.fr/def/geo#codeCommune	"38185"
http://id.insee.fr/geo/commune/38185	http://rdf.insee.fr/def/geo#codeINSEE	"38185"
http://id.insee.fr/geo/commune/38185	http://rdf.insee.fr/def/geo#nom	"Grenoble"
http://id.insee.fr/geo/commune/38185	http://rdf.insee.fr/def/geo#subdivisionDe	http://id.insee.fr/geo/arrondissement/381
http://id.insee.fr/geo/commune/38185	http://www.w3.org/2002/07/owl#sameAs	http://data.ign.fr/id/geofla/commune/38185

Examples of Links between datasets

Prefixes

```
dbpedia:      "http://dbpedia.org/resource/"
dbpedia-owl:  "http://dbpedia.org/ontology/"
geonames:     "http://sws.geonames.org/"
gn:           "http://www.geonames.org/ontology#"
insee:        "http://id.insee.fr/geo/commune/"
```

GeoNames

http://dbpedia.org/resource/grenoble

http://sws.geonames.org/3014728/

dbpedia:Stendhal

geonames:3014728

dbpedia-owl:birthPlace

gn:parentFeature

gn:name

owl:sameAs

dbpedia:Grenoble

"Grenoble"

INSEE

dbpedia-owl:elevation

geonames:6454071

insee:38185

"398"^^xsd:double

owl:sameAs

It could become even more powerful

- We could add extra knowledge to the merged datasets
  - geographical information
  - a full classification of various types of library data (novel, fiction, travel, history…)
  - etc.
- This is where ontologies, extra rules, etc, come in
  - ontologies/rule sets can be relatively simple and small, or huge, or anything in between…
- Even more powerful queries can be asked as a result

RDF outline

- RDF Model
- RDF formats
- Blank nodes
- Typed literals
- Resources definition
- RDF an data integration
- Persisting RDF
- References

Persisting RDF Data
Database systems

- Mapping RDF-relational databes
  - W3C RDB2RDF Working Group published two recommendations (september 2012)
    - R2RML: RDB to RDF Mapping Language, http://www.w3.org/TR/r2rml/
    - A Direct Mapping of Relational Data to RDF, http://www.w3.org/TR/rdb-direct-mapping/
  - DR2Q Accessing Relational Databases as Virtual RDF Graphs http://d2rq.org/

- Save triples into Relational Database
  - Various strategies: 1 giant table for alls triplet → hexastore (create indexes for all possible combinaisons: spo, pos, osp, sop, pso, ops)
  - Building an Efficient RDF Store Over a Relational Database (Mihaela A. Bornea et al., SIGMOD '13 Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data) https://cs.uwaterloo.ca/~gweddell/cs848/papers/Bornea.pdf

Persisting RDF Data
RDF Triple Stores

- W3C maintains a list of triplestores
  - http://www.w3.org/wiki/SemanticWebTools#RDF_Triple_Store_Systems
- Commercial:
  - Open Link Virtuoso - http://virtuoso.openlinksw.com
  - AllegroGraph - http://www.franz.com/agraph/allegrograph/
  - Ontotext GraphDB (SwiftOWLIM) : http://www.ontotext.com/products/ontotext-graphdb-owlim/
  - …
- Open source
  - Apache Jena - http://jena.apache.org
  - Sesame - http://www.openrdf.org
  - Parliament – http://parliament.semwebcentral.org
  - …

RDF outline

- RDF Model
- RDF formats
- Blank nodes
- Typed literals
- Resources definition
- RDF an data integration
- Persisting RDF
- References

References

- RDF is part of W3C Semantic Web W3C activity

> "The Resource Description Framework (RDF) is a framework for representing information in the Web." [1]

  - W3C recommendation February 2004 (RDF 1.0)
  - Updaded February 2014 (RDF 1.1)

RDF Working Group

Recommandations

- RDF 1.1 Concepts and Abstract Syntax [1] } RDF model
- RDF 1.1 Semantics
- JSON-LD 1.0
- JSON-LD 1.0 Processing Algorithms and API
- RDF 1.1 Turtle
- RDF 1.1 TriG
- RDF 1.1 N-Triples
- RDF 1.1 N-Quads
- RDF 1.1 XML Syntax

} RDF serialization formats

Notes

- RDF 1.1 Primer [2]
- What's new in RDF 1.1
- RDF 1.1: On Semantics of RDF Datasets
- RDF 1.1 Test Cases
- RDF 1.1 JSON Alternate Serialization (RDF/JSON)

RDFa Working Group

Recommandations

- RDFa Core 1.1 - Second Edition
- XHTML+RDFa 1.1 - Second Edition
- HTML+RDFa 1.1
- RDFa Lite 1.1

Notes

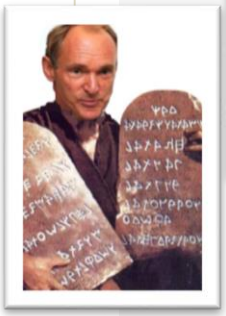
- RDFa 1.1 Primer - Second Edition [3]
- Linked Data Glossary [4]
- HTML Data Guide

[1] http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/
[2] http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/
[3] http://www.w3.org/TR/2013/NOTE-rdfa-primer-20130822/
[4] http://www.w3.org/TR/2013/NOTE-ld-glossary-20130627/

Outline

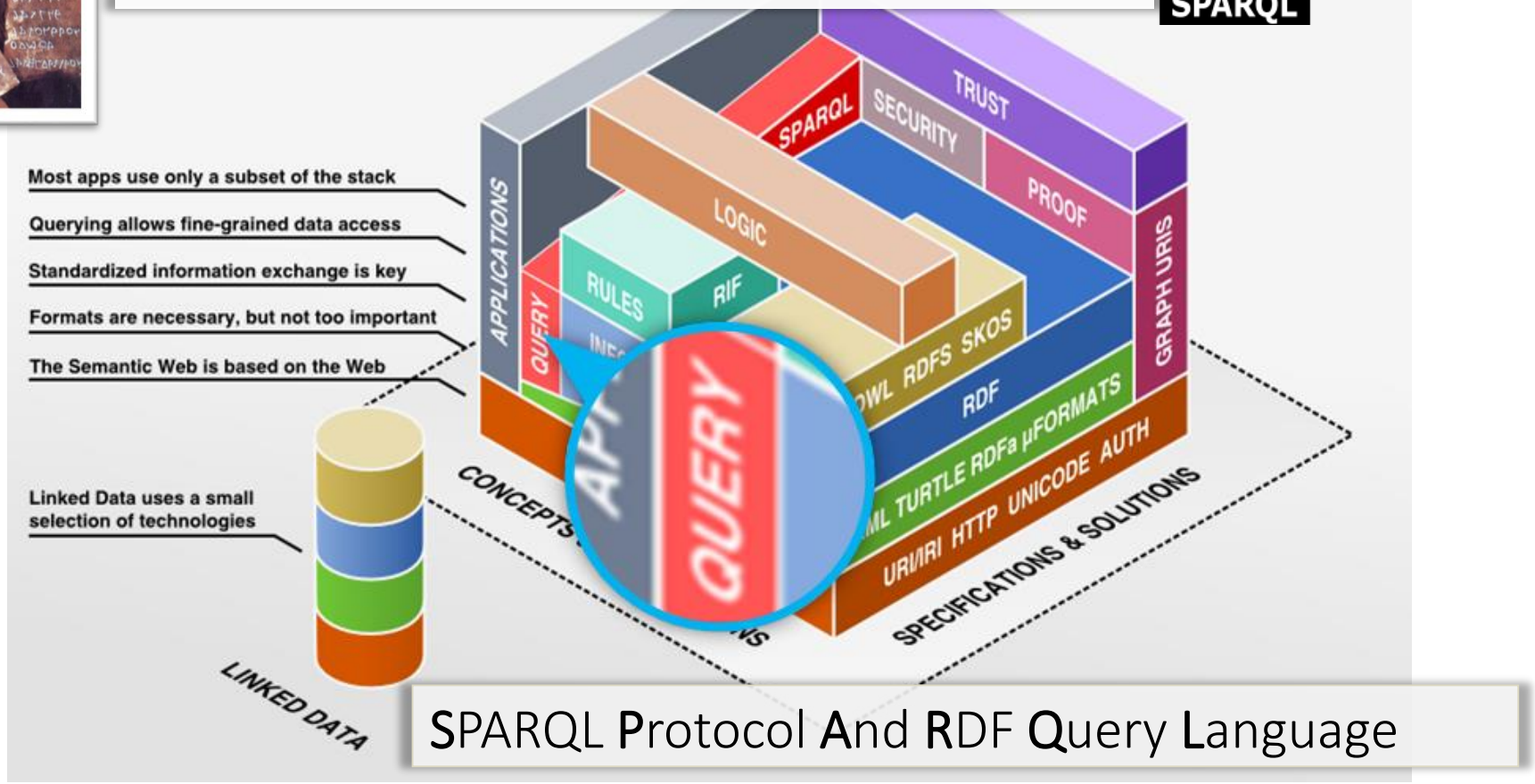
- "Theoretical" Session (morning)
  - Introduction
  - Distributing Data on the web with RDF
    - Naming the Data : URIs (Uniform Resources Identifiers)
    - The RDF Data model
  - Querying Linked Data with SPARQL
  - Semantic modelling
    - RDFS
    - OWL
  - From Open Data to Linked Open Data
  - Conclusion
- Hands-on session (afternoon)
  - From a CSV file to linked data
  - Querying linked data (SPARQL)

Querying Linked Data with SPARQL



Linked Data: 3rd Principle

*When someone looks up a URI, provide useful information, using the standards (RDF, **SPARQL**).*



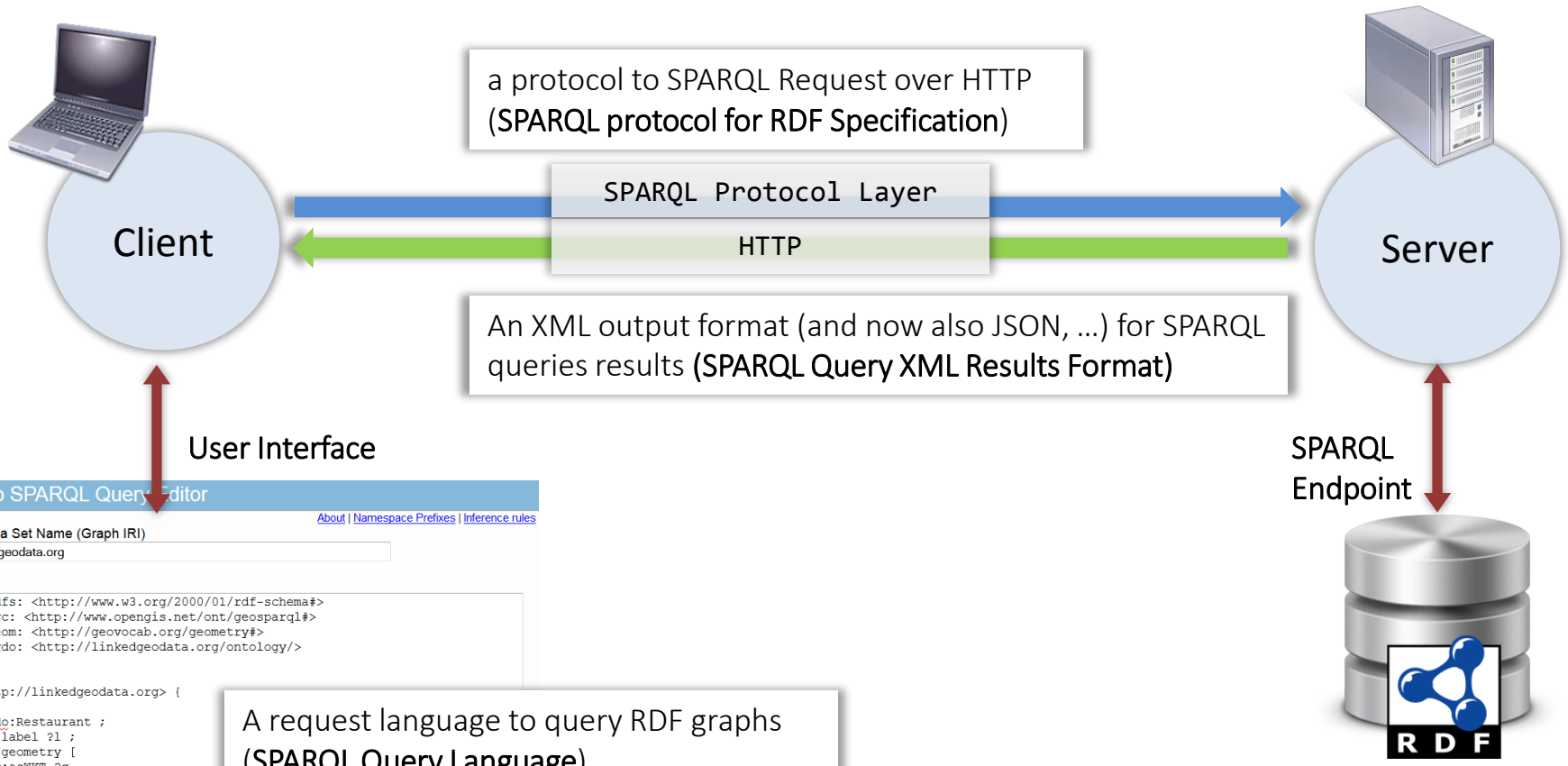
SPARQL Protocol And RDF Query Language

SPARQL : introduction

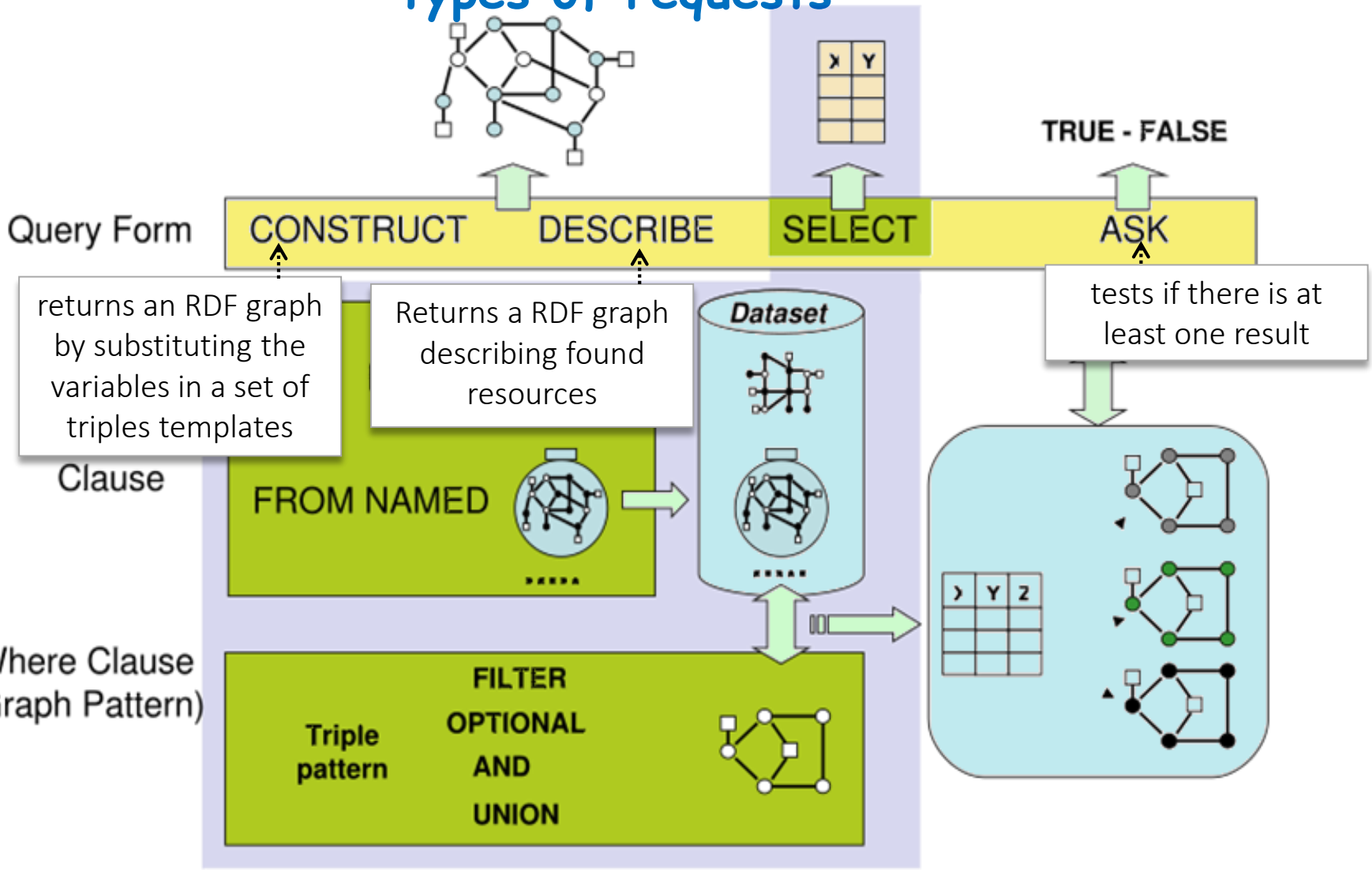
- RDF (Resource Description Framework)
  - Flexible and extensible way to represent information about resources of the web
- SPARQL (SPARQL Protocol And RDF Query Language)
  - A W3C standard
    - SPARQL 1.0 recommendation - January 2008,
    - SPARQL 1.1 recommendation – March 2013 http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/
  - a request language to access a RDF graph (SPARQL Query Language Specification) inspired from SQL
  - a protocol to submit request through HTTP GET, HTTP POST or SOAP (SPARQL protocol for RDF Specification)
  - an XML format for the results (SPARQL Query XML Results Format), and now JSON

SPARQL in action with LinkedGeoData

SPARQL Protocol And RDF Query Language
overview



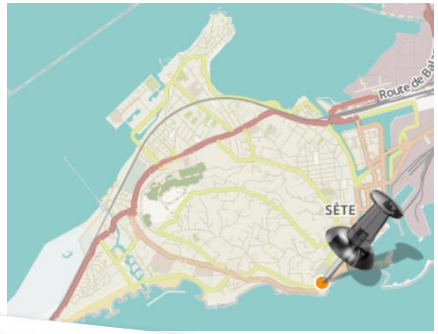
SPARQL Query Language
types of requests



from: Pérez, Arenas and Gutierrez, Chapter 1: On the Semantics of SPARQL, Semantic Web Information Management: A Model Based Perspective, Springer 2010

SPARQL Query Language
Select

- find all the restaurants that are less than 1km from Fort Saint-Pierre (Théâtre de la Mer – Sète)



```
Prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Prefix ogc:  <http://www.opengis.net/ont/geosparql#>
Prefix geom: <http://geovocab.org/geometry#>
Prefix lgdo: <http://linkedgeodata.org/ontology/>

Select ?s, ?l From <http://linkedgeodata.org>
Where
{
     ?s    a lgdo:Restaurant ;
           rdfs:label ?l ;
           geom:geometry [
              ogc:asWKT ?g
           ] .

     Filter(bif:st_intersects (?g, bif:st_point (3.692764, 43.393794), 1)) .
}
```

SPARQL Query Language

- SPARQL based on :
  - RDF serialization with Turtle
  - Graph Pattern Matching

```
Prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Prefix ogc:  <http://www.opengis.net/ont/geosparql#>
Prefix geom: <http://geovocab.org/geometry#>
Prefix lgdo: <http://linkedgeodata.org/ontology/>

Select ?s, ?l From <http://linkedgeodata.org>
Where
{
      ?s    a lgdo:Restaurant ;
            rdfs:label ?l ;
            geom:geometry [
               ogc:asWKT ?g
            ] .

      Filter(bif:st_intersects (?g, bif:st_point (3.692764, 43.393794), 1)) .
}
```

variable

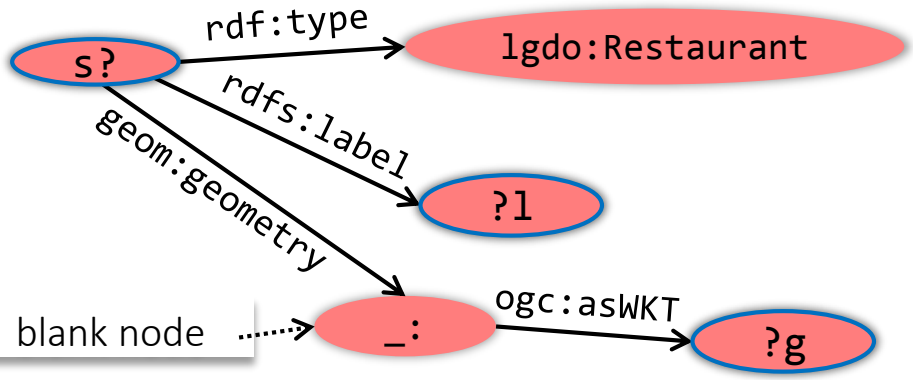
Graph Pattern: RDF triple containing one or more variables at any position (subject, predicate, object)

SPARQL Query Language
Select

- Graph patterns can be combined to construct complex (conjunctive) requests.

```
Prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Prefix ogc:  <http://www.opengis.net/ont/geosparql#>
Prefix geom: <http://geovocab.org/geometry#>
Prefix lgdo: <http://linkedgeodata.org/ontology/>

Select ?s, ?l From <http://linkedgeodata.org>
Where
{
      ?s    a lgdo:Restaurant ;
            rdfs:label ?l ;
            geom:geometry [
               ogc:asWKT ?g
            ] .

      Filter(bif:st_intersects (?g, bif:st_point (3.692764, 43.393794), 1)) .
}
```

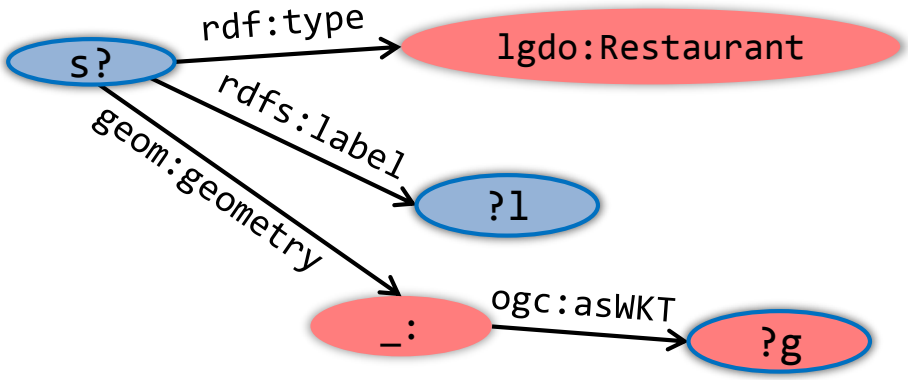
variable



SPARQL Query Language
Select

- The SELECT clause indicates which variables to consider in the result
- * all the variables (like SQL)

```
Prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Prefix ogc:  <http://www.opengis.net/ont/geosparql#>
Prefix geom: <http://geovocab.org/geometry#>
Prefix lgdo: <http://linkedgeodata.org/ontology/>

Select ?s, ?l From <http://linkedgeodata.org>
Where
{
     ?s    a lgdo:Restaurant ;
           rdfs:label ?l ;
           geom:geometry [
              ogc:asWKT ?g
           ] .

     Filter(bif:st_intersects (?g, bif:st_point (3.692764, 43.393794), 1)) .
}
```



SPARQL Query Language
Select clause - DataSets

- The RDF data-store service can handle one or more RDF graphs, the SPARQL query is executed against a data set (RDF Dataset) that represents a collection of one or more graphs.

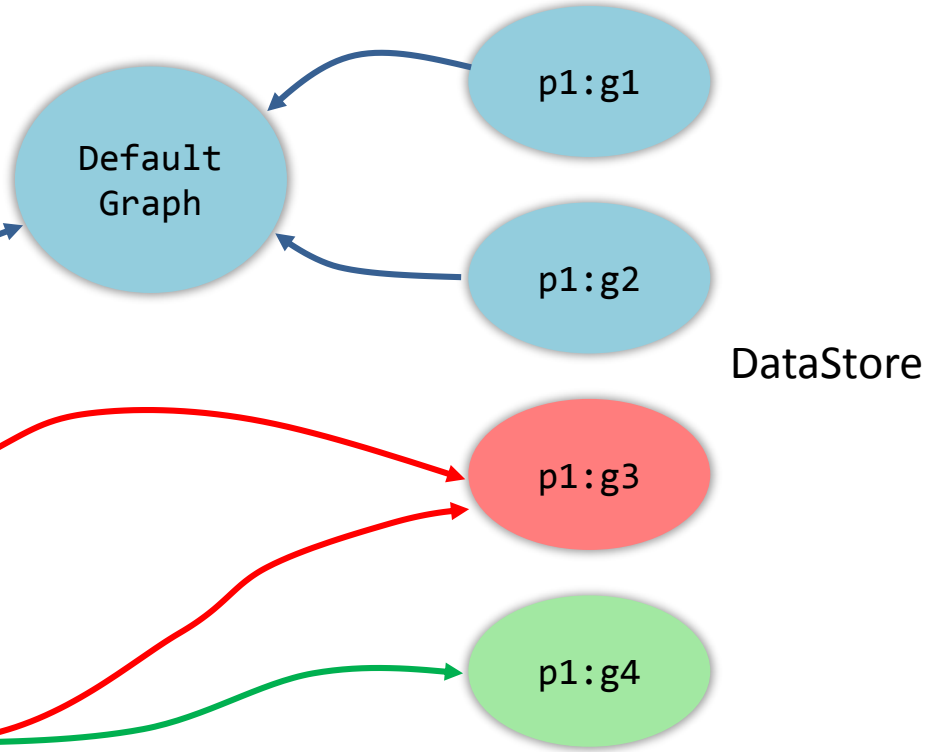
```
Prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Prefix ogc:  <http://www.opengis.net/ont/geosparql#>
Prefix geom: <http://geovocab.org/geometry#>
Prefix lgdo: <http://linkedgeodata.org/ontology/>

Select ?s, ?l From <http://linkedgeodata.org>
Where
{
     ?s    a lgdo:Restaurant ;
           rdfs:label ?l ;
           geom:geometry [
              ogc:asWKT ?g
           ] .

     Filter(bif:st_intersects (?g, bif:st_point (3.692764, 43.393794), 1)) .
}
```

graph on which the query is executed

SPARQL Query Language
Select - DataSets

- a SPARQL request can associate different graph patterns to different named graphs.
  - FROM defines the default graph (this may be merging several graphs)
  - FROM NAMED defines graphs which then can be explicitly named in the WHERE part of the query through the GRAPH keyword

```
PREFIX p1: ...
SELECT ...
FROM p1:g1
FROM p2:g2
FROM NAMED hpi:g3
FROM NAMED hpi:g4
WHERE {
   ......
   GRAPH hpi:g3 {
        ......
   }
   GRAPH ?g {
     ......
   }
}
```

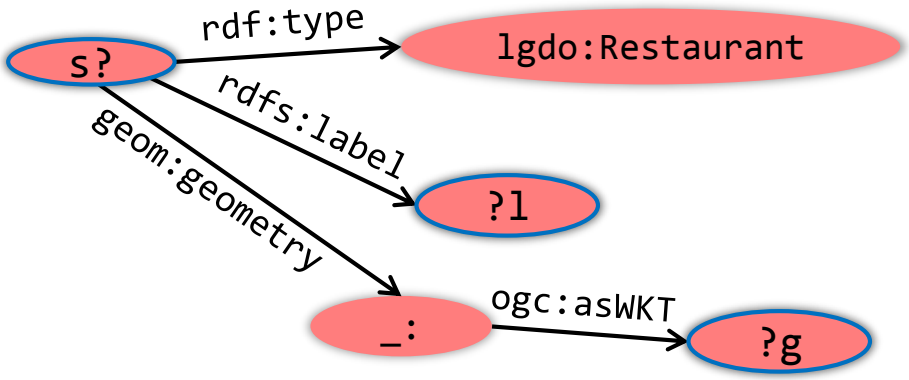


SPARQL Query Language
Select -Filters

- Possibility to filter results
  - A filter allows to constrain solution values

```
Prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
Prefix ogc:  <http://www.opengis.net/ont/geosparql#>
Prefix geom: <http://geovocab.org/geometry#>
Prefix lgdo: <http://linkedgeodata.org/ontology/>

Select ?s, ?l From <http://linkedgeodata.org>
Where
{
      ?s    a lgdo:Restaurant ;
            rdfs:label ?l ;
            geom:geometry [
               ogc:asWKT ?g
            ] .

      Filter(bif:st_intersects (?g, bif:st_point (3.692764, 43.393794), 1)) .
}
```

Graph Pattern



Filter on ?g variable

SPARQL Query Language
Select -Filters

- Filters : allow to restrict the values in a solution
  - Boolean expression the request solutions must satisfy.
  - rich expression language based on Xpath, XQuery and special operators defined by SPARQL.
    see section 11 of SPARQL specification document http://www.w3.org/TR/rdf-sparql-query/#tests
    - Relational operators: **<, >, =, <=, >=, !=**
    - Boolean operators: **&&, ||, !**
    - Arithmetic operator: **+ * - /**
    - Variable binding tests: **isURI(?x), isBlank(?x), isLiteral(?x), bound(?x**)
    - Regular expressions: **regex(?x, "A.*")**
    - Access to attributes/values **lang(), datatype(), str()**
    - Casting ((re-)typing functions) **xsd:integer(?x)**
    - External functions / extensions

SPARQL Query Language
Solution Modifiers

- The set of solution produced by graphs pattern matching can be modified in various ways:
  - Projection - keep only selected variables
  - OFFSET/LIMIT - chop the number solutions (best used with ORDER BY)
    - OFFSET the start index,
    - LIMIT the number of solutions to be returned.
      - Using LIMIT alone useful to ensure not too many solutions are returned, to restrict the effect of some unexpected situation
  - ORDER BY - sorted results
  - DISTINCT - yield only one row for one combination of variables and values.

- The solution modifiers OFFSET/LIMIT and ORDER BY always apply to all result forms

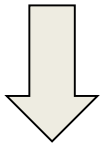
example

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
SELECT DISTINCT * where {
    ?x foaf:name ?name;
       faof:age ?age.
}
ORDER BY ?name DESC(?age)
```

SPARQL Query Language
ASK query

- ASK – ask a boolean query.
  - to verify that there is at least one response.
  - Is there a student over 30 years?

```
PREFIX ufrimag: <http://www.ufrimag.fr#>
ASK {
 ?etudiant ufrimag:inscrit ?x .
 ?x ufrimag:siteweb <http://www.ufrimag.fr> .
 ?etudiant ufrimag:age ?age .
 FILTER (?age > 30)
}
```



the boolean result

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
 <head> … </head>
 <boolean> true </boolean>
</sparql>
```

SPARQL Query Language
CONSTRUCT query

- SELECT returns a flat list of variables bindings
  - the application program is in charge of processing these bindings (often by converting solution tuples into triples and adding them to a RDF graph)

- CONSTRUCT allows you to directly product a RDF graph containing the variables values
  - the WHERE and FILTER clause works the same way as the SELECT form
  - bindings of the variables are inserted into a new graph constructed from template triples specified in the CONSTRUCT clause (which replace the SELECT clause).

SPARQL Query Language
CONSTRUCT query

```
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
PREFIX vcard:   <http://www.w3.org/2001/vcard-rdf/3.0#>

CONSTRUCT { ?x foaf:firstName ?y .
            ?x foaf:lastName ?z .
          }

FROM <vca-db-1rd.rdf>
WHERE
 {        ?x vcard:N ?u .
          ?u vcard:Given ?y .
          ?u vcard:Family ?z.
 }
```

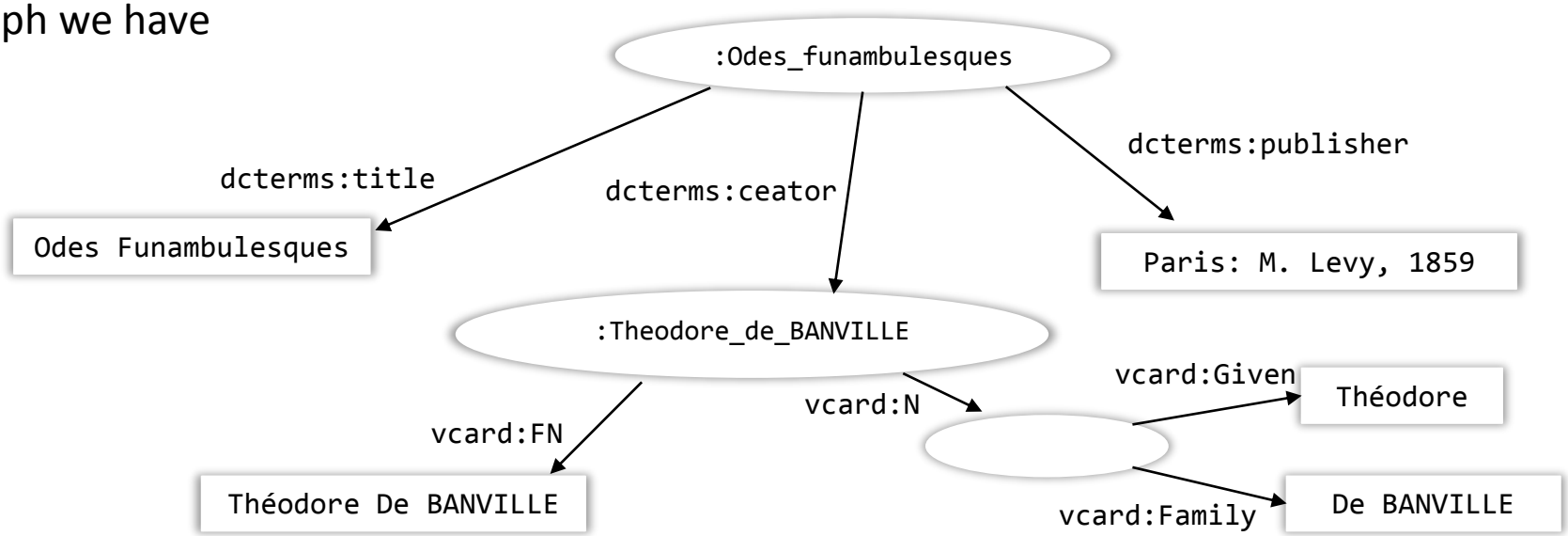
CONSTRUCT avec un graph-gabarit à un seul sujet et deux triplets

```
@prefix foaf:    <http://xmlns.com/foaf/0.1/> .
@prefix vcard:   <http://www.w3.org/2001/vcard-rdf/3.0#> .

<http://somewhere/RebeccaSmith/>
      foaf:firstName  "Rebecca" ;
      foaf:lastName  "Smith" .

<http://somewhere/MattJones/>
      foaf:firstName  "Matthew" ;
      foaf:lastName  "Jones" .

<http://somewhere/SarahJones/>
      foaf:firstName  "Sarah" ;
      foaf:lastName  "Jones" .

<http://somewhere/JohnSmith/>
      foaf:firstName  "John" ;
      foaf:lastName  "Smith" .
```

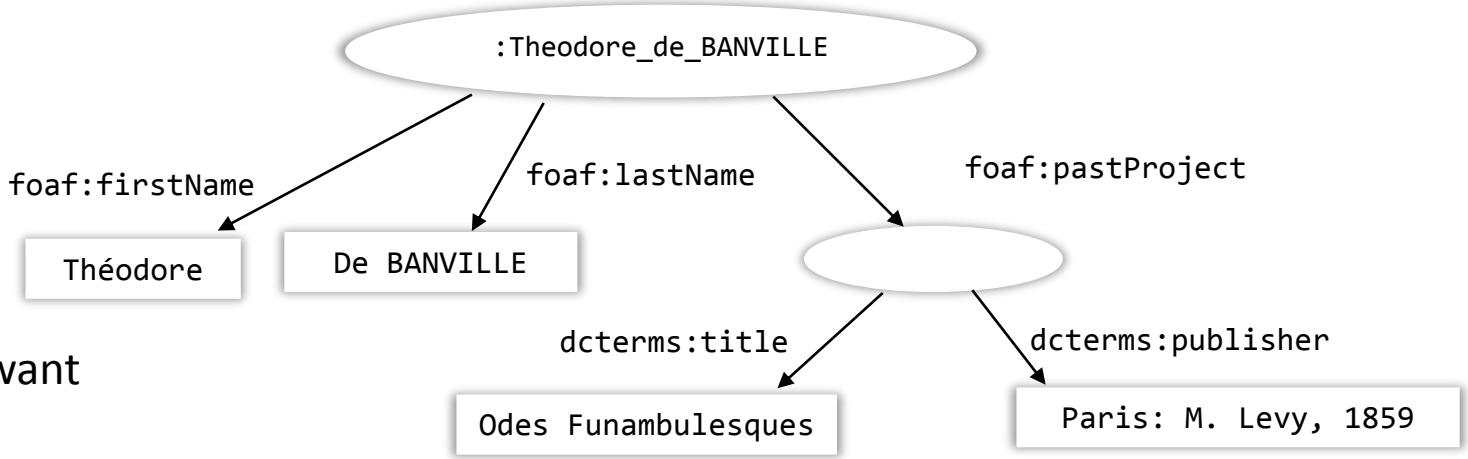
D'après : http://pagesperso-systeme.lip6.fr/Jean-Francois.Perrot/inalco/XML/RDF/SPARQL/IntroSPARQL.html

SPARQL Query Language
CONSTRUCT query

The graph we have



The graph we want



SPARQL Query Language
CONSTRUCT query

```
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
PREFIX dc:      <http://purl.org/dc/elements/1.1/>
PREFIX vcard:   <http://www.w3.org/2001/vcard-rdf/3.0#>

CONSTRUCT { ?auteur foaf:pastProject _:oeuvre .
            _:oeuvre dc:title ?nom .
            _:oeuvre dc:publisher ?editeur .
            ?auteur foaf:firstName ?p .
            ?auteur foaf:lastName ?n .
          }

FROM <http://pagesperso-systeme.lip6.fr/Jean-Francois.Perrot/inalco/XML/RDF/Frantext/RRb.rdf>
WHERE
 {         ?oe  dc:creator ?auteur .
           ?oe  dc:title ?nom .
           ?oe  dc:publisher ?editeur .
           ?auteur  vcard:N ?vc .
           ?vc vcard:Family ?n .
           ?vb vcard:Given ?p .
 }
```

```
@prefix vcard:   <http://www.w3.org/2001/vcard-rdf/3.0#> .
@prefix dc:      <http://purl.org/dc/elements/1.1/> .
@prefix foaf:    <http://xmlns.com/foaf/0.1/> .

<http://inalco/M2-Trad/poetes#Théodore_de_BANVILLE>
      foaf:firstName  "Théodore" ;
      foaf:lastName  "de BANVILLE" ;
      foaf:pastProject
                  [ dc:publisher  "Paris : M. Levy, 1859." ;
                    dc:title      "Odes funambulesques"
                  ] .
```

SPARQL Query Language
DESCRIBE query

- DESCRIBE – Returns an RDF graph, based on what the query processor is configured to return.
  - SPARQL specification says : "the useful information the service has about a resource"
  - in theory this should help you understand the context of the resources returned… but there is no warranty.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX ex: <http://example.com/>
DESCRIBE ex:karen ?friend {
  ex:karen foaf:knows ?friend .
}
```

asks for a description of karen ad her friends

Subject	Predicate	Object	
ex:karen	foaf:knows	ex:alex	×
ex:karen	foaf:name	"Karen"	×
ex:alex	foaf:name	"Alex"	×

```
DESCRIBE <http://example.com/fish> ?x WHERE {
  ?x ?y <http://example.com/fish>
}
```

asks for a description of fish, and any resource directly related to fish.

SPARQL Protocol

Client

SPARQL request over HTTP : SPARQL URI

SPARQL Protocol Layer

HTTP

Serveur

the SPARQL endpoint URL
http://linkedgeodata.org/sparql

SPARQL request (SPARQL URI) has 4 components:

SPARQL Query Language

```
Prefix rdfs: <http
Prefix ogc:  <http
Prefix geom: <http
Prefix lgdo: <http
Select ?s, ?l From
Where
{
     ?s    a lgdo:
           rdfs:la
           geom:ge
              ogc:
           ] .

     Filter(bif:st_intersects (?g, bif:st_point (3.692764, 43.393794), 1)) .
}
```

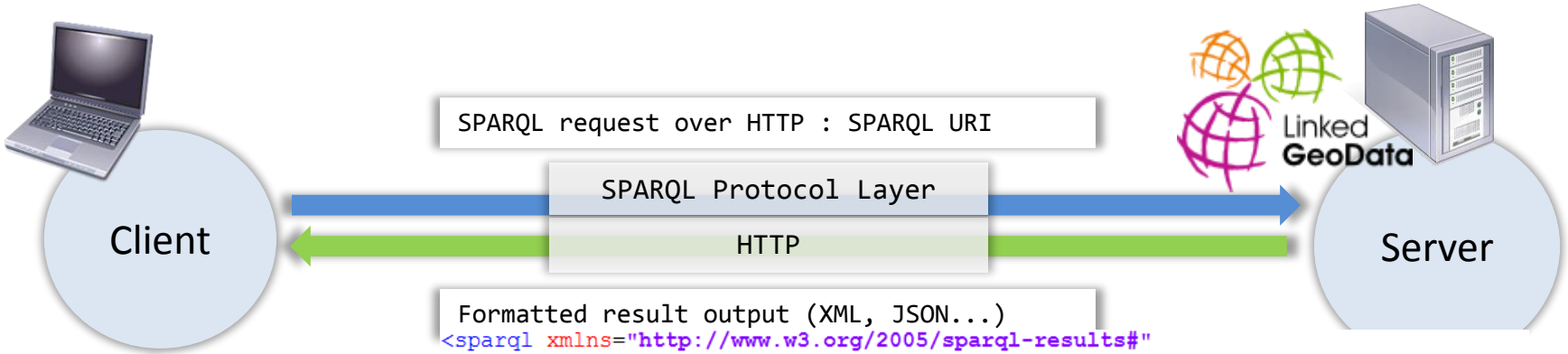
The query string

http://linkedgeodata.org/sparql?**default-graph-uri**=http%3A%2F%2Flinkedgeodata.org
&**query**=Prefix+rdfs%3A+%3Chttp%3A%2F%2Fwww.w3.org%2F2000%2F01%2Frdf-schema%23%3E%0D%0APrefix+ogc%3A+%3Chttp%3A%2F%2Fwww.opengis.net%2Fogc%2Fgeosparql%23%3E%0D%0APrefix+geom%3A+%3Chttp%3A%2F%2Fgeovocab.org%2Fgeometry%23%3E%0D%0APrefix+lgdo%3A+%3Chttp%3A%2F%2Flinkedgeodata.org%2Fontology%2F%3E%0D%0A%0D%0ASelect+*%0D%0AFrom+%3Chttp%3A%2F%2Flinkedgeodata.org%3E+%7B%0D%0A++%3Fs%0D%0A++++a+lgdo%3ARestaurant+%3B%0D%0A++++rdfs%3Alabel+%3Fl+%3B++++%0D%0A++++geom%3Ageometry+%5B%0D%0A++++++ogc%3AasWKT+%3Fg%0D%0A++++%5D+.%0D%0A%0D%0A+++Filter%28bif%3Ast_intersects+%28%3Fg%2C+bif%3Ast_point+%283.692764%2C+43.393794%29%2C+1%29%29+.%0D%0A%7D&**format**=application%2Fsparql-results%2Bxml

The RDF graphs to request (optional)
default-graph-uri=http://linkedgeodata.org
named-graph-uri=...

Output format
format=application/sparql-results+xml
(text/html, json…)

SPARQL : output format (XML)



SPARQL Query Language

```
Prefix rdfs: <http://www.w3.org/2000/01/rdf-s
Prefix ogc:  <http://www.opengis.net/ont/geos
Prefix geom: <http://geovocab.org/geometry#>
Prefix lgdo: <http://linkedgeodata.org/ontolo

Select ?s,
Where
{
     ?s
                ogc:asWKT ?g
          ] .

     Filter(bif:st_intersects (?g, bif:st_poi
}
```

```
<sparql xmlns="http://www.w3.org/2005/sparql-results#"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://www.w3.org/2001/sw/DataAccess/rf1/result2.xsd">
  <head>
    <variable name="s"/>
    <variable name="l"/>
  </head>
  <results distinct="false" ordered="true">
    <result>
      <binding name="s">
        <uri>http://linkedgeodata.org/triplify/node1304006380</uri>
      </binding>
      <binding name="l">
        <literal>L&#39;Hostal</literal>
      </binding>
    </result>
    <result>
      <binding name="s">
        <uri>http://linkedgeodata.org/triplify/node1952213273</uri>
      </binding>
      <binding name="l">
        <literal>L&#39;Ultima</literal>
      </binding>
    </result>
  </results>
```

Variables from the SELECT clause

one **result** element for each row in the result set

binding of one the variables of this result

SPARQL 1.1: new functionalities

- W3C Recommendation, March 2013, 21
  - http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/
- Query
  - New aggregation functions for results (`count`, `min`, `max`, `group by`, etc.)
  - Variables assignment
    - `SELECT (COUNT(DISTINCT ?s)) AS ?num` *number of distinct restaurants*
  - Negation - MINUS
    - `NOT EXIST`, `EXIST` :filtering results depending on whether a graph pattern does or does not match in the context of the query solution being filtered,

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

SELECT ?person
WHERE
{
    ?person rdf:type  foaf:Person .
    FILTER NOT EXISTS { ?person foaf:name ?name }
}
```

Persons who don't have a name

Example from
http://www.w3.org/TR/2013/REC-sparql11-query-20130321/#negation

    - `MINUS` removing solutions related to another pattern.

SPARQL 1.1: new functionalities

- Query…
  - subqueries: possibility to embed SPARQL queries within other queries
    - e.g. for limiting the number of results from some sub-expression within the query
  - subqueries are evaluated logically first, and the results are projected up to the outer query.

Example*: *Return a name (the one with the lowest sort order) for all the people that know Alice and have a name.* • From http://www.w3.org/TR/2013/REC-sparql11-query-20130321/#subqueries

The data

```
@prefix : <http://people.example/> .
:alice :name "Alice", "Alice Foo", "A. Foo" .
:alice :knows :bob, :carol .
:bob :name "Bob", "Bob Bar", "B. Bar" .
:carol :name "Carol", "Carol Baz", "C. Baz" .
```

```
PREFIX : <http://people.example/>
PREFIX : <http://people.example/>
SELECT ?y ?minName
WHERE {
  :alice :knows ?y .
  {
    SELECT ?y (MIN(?name) AS ?minName)
    WHERE {
      ?y :name ?name .
    } GROUP BY ?y
  }
}
```

y	minName
:bob	"B. Bar"
:carol	"C. Baz"

y
:bob
:carol

y	minName
:alice	"A. Foo"
:bob	"B. Bar"
:carol	"C. Baz"

1: inner query evaluation

2: outer query evaluation

3: results of 1. are joined with results of 2.

SPARQL 1.1: new functionalities

- Query…
  - Basic federated queries (`SERVICE`, `BINDING`)
    - To execute requests distributed over different SPARQL endpoints

Example*: *is there anyone among Alice's friends with the same name as the resource identified by the IRI* ***<http://dbpedia.org/resource/Snoopy>*** *at Dbpedia?*

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
    <http://example.org/alice#me> foaf:knows [ foaf:name ?name ] .
    SERVICE <http://dbpedia.org/sparql> {
       <http://dbpedia.org/resource/Snoopy> foaf:name ?name
    }
}
```

names of Alice's friend. This pattern is matched against the local SPARQL service

find out the name of http://dbpedia.org/resource/Snoopy

evaluation of this pattern s delegated to the respective remote SPARQL service **`http://dbpedia.org/sparql`**

* From http://www.w3.org/TR/sparql11-overview/

SPARQL 1.1: new functionalities …

- New serialization formats for request results (JSON…)
- CRUD operations
  - Graph update: INSERT, INSERT DATA, DELETE DATA, DELETE, DELETE WHERE, LOAD, CLEAR)
  - Graph management: CREATE, DROP, COPY, MOVE, ADD
- Entailments
  - RDF, RDFS, OWL, RIF
- …

http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/

Outline

- "Theoretical" Session (morning)
  - Introduction
  - Distributing Data on the web with RDF
    - Naming the Data : URIs (Uniform Resources Identifiers)
    - The RDF Data model
  - Querying Linked Data with SPARQL
  - Semantic modelling
    - RDFS
    - OWL
  - From Open Data to Linked Open Data
  - Conclusion
- Hands-on session (afternoon)
  - From a CSV file to linked data
  - Querying linked data (SPARQL)

RDF limitations

- RDF provides a standard way to express simple statements about resources, using named properties and values…

  But you can't express knowledge about the properties and types of the resources
  - what are the types allowed for resources ?
  - what are the properties allowed for a given type of resource ?
  - what are the allowed values for a given property ?
  - what are the relations between types of resources (generalization/specialization) ?
  - …

Resource Description Framework

- RDF a simple model (labelled graph) for data exchange

`http://dbpedia.org/page/Rin_Tin_Tin`

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

`http://dbpedia.org/class/yago/AnimalActors`

"On the Internet, nobody knows you're a dog."

But…

- Which labels for the nodes and edges??
- How to interpret them ?

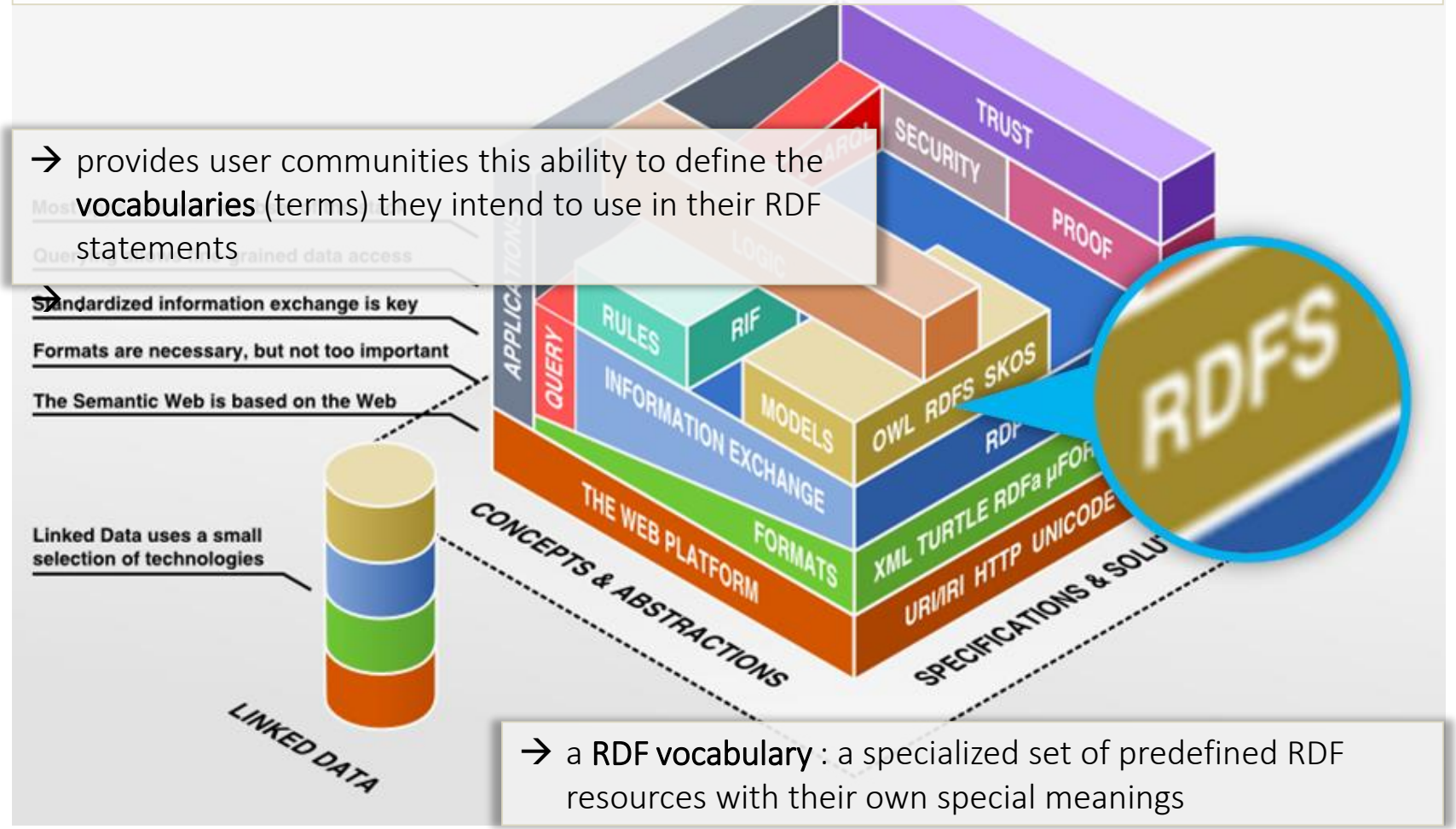
Semantic Web : formally capture some aspects of the meaning of these labels.

RDF Schema (RDFS)

Officially called “RDF Vocabulary Description Language”

“Schema” is retained for historical reasons (XMLmania…)

→ provides user communities this ability to define the **vocabularies** (terms) they intend to use in their RDF statements



→ a **RDF vocabulary** : a specialized set of predefined RDF resources with their own special meanings

RDF Schema (RDF(S))

- Introduction
- RDF(S) Classes
- RDF(S) Properties
- Interpreting RDF(S) Schema Declarations
- More RDF(S) properties
- Conclusion

What is a vocabulary ?

- On the Semantic Web, vocabularies define the concepts and relationships (also referred to as “terms”) used to describe and represent an area of concern. Vocabularies are used to classify the terms that can be used in a particular application, characterize possible relationships, and define possible constraints on using those terms. In practice, vocabularies can be very complex (with several thousands of terms) or very simple (describing one or two concepts only).

- There is no clear division between what is referred to as “**vocabularies**” and “**ontologies**”. The trend is to use the word “ontology” for more complex, and possibly quite formal collection of terms, whereas “vocabulary” is used when such strict formalism is not necessarily used or only in a very loose sense. Vocabularies are the basic building blocks for inference techniques on the Semantic Web.

http://www.w3.org/standards/semanticweb/ontology

RDF Schema - RDF(S)

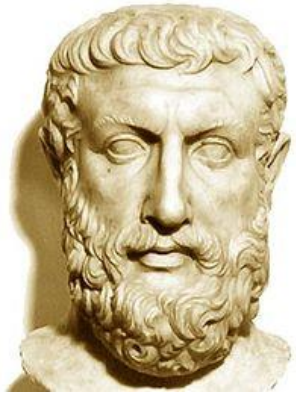
- RDF(S) extends RDF with a *schema vocabulary (**`Resource`**, **`Class`**, **`Property`**, **`subClassOf`**, **`subPropertyOf`**, **`range`**, **`domain`**)* that allows us to define basic vocabulary terms and the relations between them.
  - W3C recommendation RDF Vocabulary Description Language –RDF 1.1 (Feb. 2014) http://www.w3.org/TR/rdf-schema/ (previous rec. RDF 1.0 Feb. 2004)

- A well-defined **semantics** gives "extra meaning" to these particular RDF predicates and resources
  - specifies how terms should be interpreted
  - allows us to draw simple inferences (*entailments*)

- the RDF(S) schema vocabulary is itself provided in the form of an RDF vocabulary
  - resources in the RDF Schema vocabulary have URIrefs with the prefix `http://www.w3.org/2000/01/rdf-schema#` (conventional abbreviation `rdfs:`)

What is an Ontology ?

- "**ontology** is the philosophical study of the nature of being, becoming, existence, or reality, as well as the basic categories of being and their relations. Traditionally listed as a part of the major branch of philosophy known as **metaphysics**, ontology deals with questions concerning what entities exist or can be said to exist, and how such entities can be grouped, related within a hierarchy, and subdivided according to similarities and differences." http://en.wikipedia.org/wiki/Ontology



Parmenides
(c. 515 BCE - c. 460 BCE)

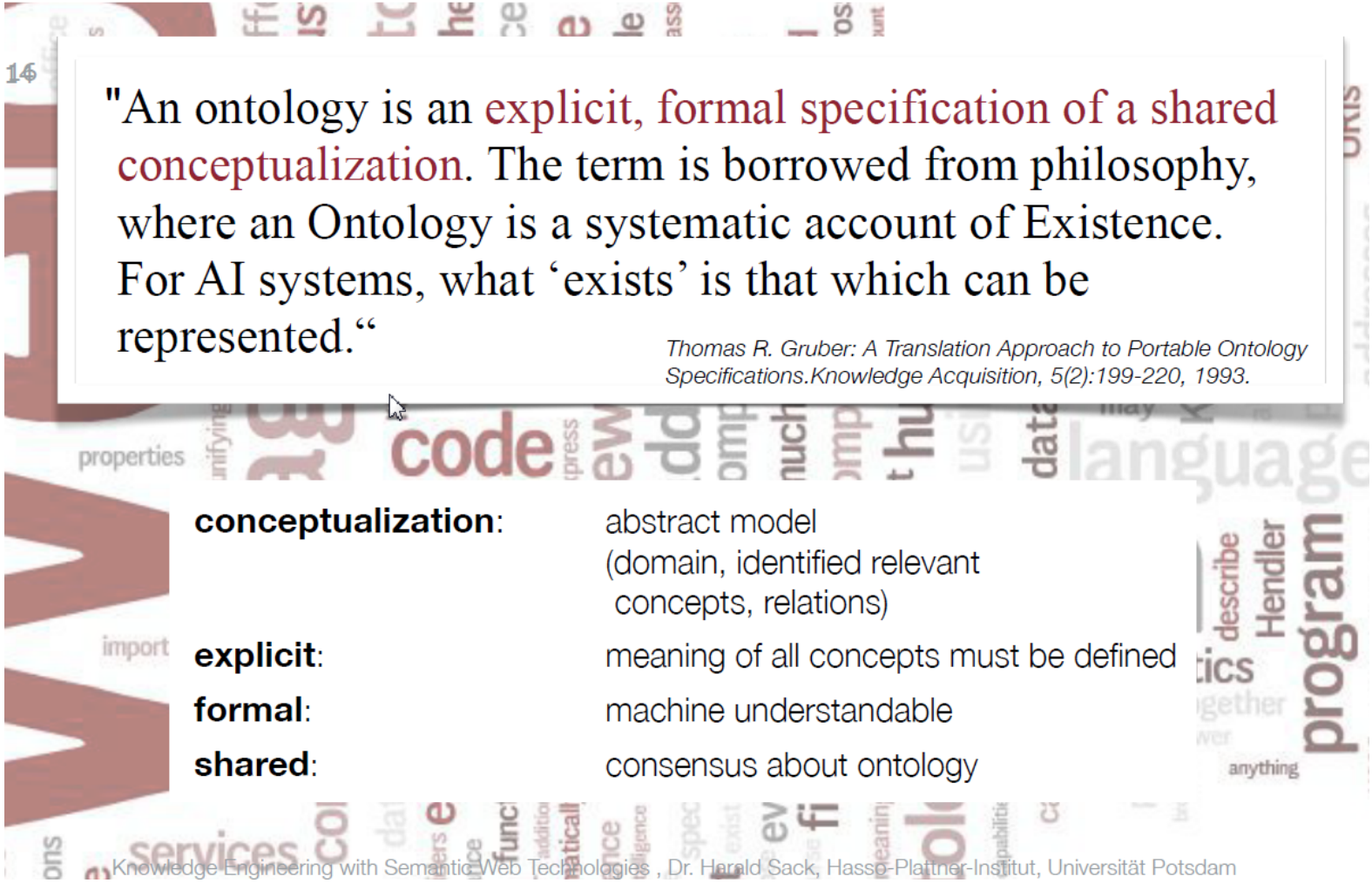
- "In computer science …, an ontology is a formal framework for representing knowledge. This framework names and defines the types, properties, and interrelationships of the entities in a domain of discourse. The entities are conceptualizations (limited abstractions) of phenomena."
  http://en.wikipedia.org/wiki/Ontology_%28information_science%29

Thomas R. Gruber
(1959 -)

An ontology is an explicit specification of a conceptualization. [...] A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose.

Thomas R. Gruber, *Towards Principles for the Design of Ontologies Used for Knowledge Sharing* in Formal Ontology in Conceptual Analysis and Knowledge Representation, Kluwer Academic Publishers, 1993
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.91.6025&rep=rep1&type=pdf

The Concept of Ontology in Computer Science



"An ontology is an explicit, formal specification of a shared conceptualization. The term is borrowed from philosophy, where an Ontology is a systematic account of Existence. For AI systems, what 'exists' is that which can be represented."

Thomas R. Gruber: A Translation Approach to Portable Ontology Specifications.Knowledge Acquisition, 5(2):199-220, 1993.

conceptualization:	abstract model (domain, identified relevant concepts, relations)
explicit:	meaning of all concepts must be defined
formal:	machine understandable
shared:	consensus about ontology

RDF Schema - RDF(S)

- Vocabulary descriptions (schemas) written in the RDF(S) language are legal RDF graphs.
  - any software that can process RDF can also interpret a RDF(S) schema as a legal RDF graph
  - … but must be extended to understand the additional built-in meanings of the RDF Schema terms.

RDF Schema - RDF(S)

- RDF(S) provides a *type system* for RDF
  - similar in some respects to the type systems of object-oriented programming languages such as Java
    - Classes ⇔ Types
    - Attributes ⇔ Properties
    - Instances ⇔ Resources
    - inheritance ⇔ Hierarchical organization of types
  - but RDF Schema is a description language not a programming language : no methods

RDF Schema - RDF(S)

- RDF(S) extends RDF with a *schema vocabulary (`Resource`, `Class`, `Property`, `subClassOf`, `subPropertyOf`, `range`, `domain`…)* that allows us to define basic vocabulary terms and the relations between them.
  - W3C recommendation RDF Vocabulary Description Language http://www.w3.org/TR/rdf-schema/
- the RDF(S) schema vocabulary is itself provided in the form of an RDF vocabulary
  - resources in the RDF Schema vocabulary have URIrefs with the prefix **`http://www.w3.org/2000/01/rdf-schema#`** (conventional prefix `rdfs:`)
  - Semantics gives "extra meaning" to these particular RDF predicates and resources
    - specifies how terms should be interpreted
    - allows us to draw inferences

RDF Schema - RDF(S)

- Vocabulary descriptions (schemas) written in the RDF(S) language are legal RDF graphs.
  - any software that can process RDF can also interpret a RDF(S) schema as a legal RDF graph
  - … but must be extended to understand the additional built-in meanings of the RDF Schema terms.

RDF Schema (RDF(S))

- Introduction
- RDF(S) Classes
- RDF(S) Properties
- Interpreting RDF(S) Schema Declarations
- More RDF(S) properties
- Conclusion

RDF(S) Classes

- **classes** : Sets of objects/instances/individuals sharing certain characteristics
- in RDF(S) a class is any resource having an **rdf:type** property whose value is the resource **rdfs:Class**.

```
ex:MotorVehicle rdf:type rdfs:Class .
```

⇕

```
ex:BlackThing a rdfs:Class .
```

  - **rdf:type** property to indicate that a resource is an instance of a class.

```
rdfs:Class rdf:type rdfs:Class.
```

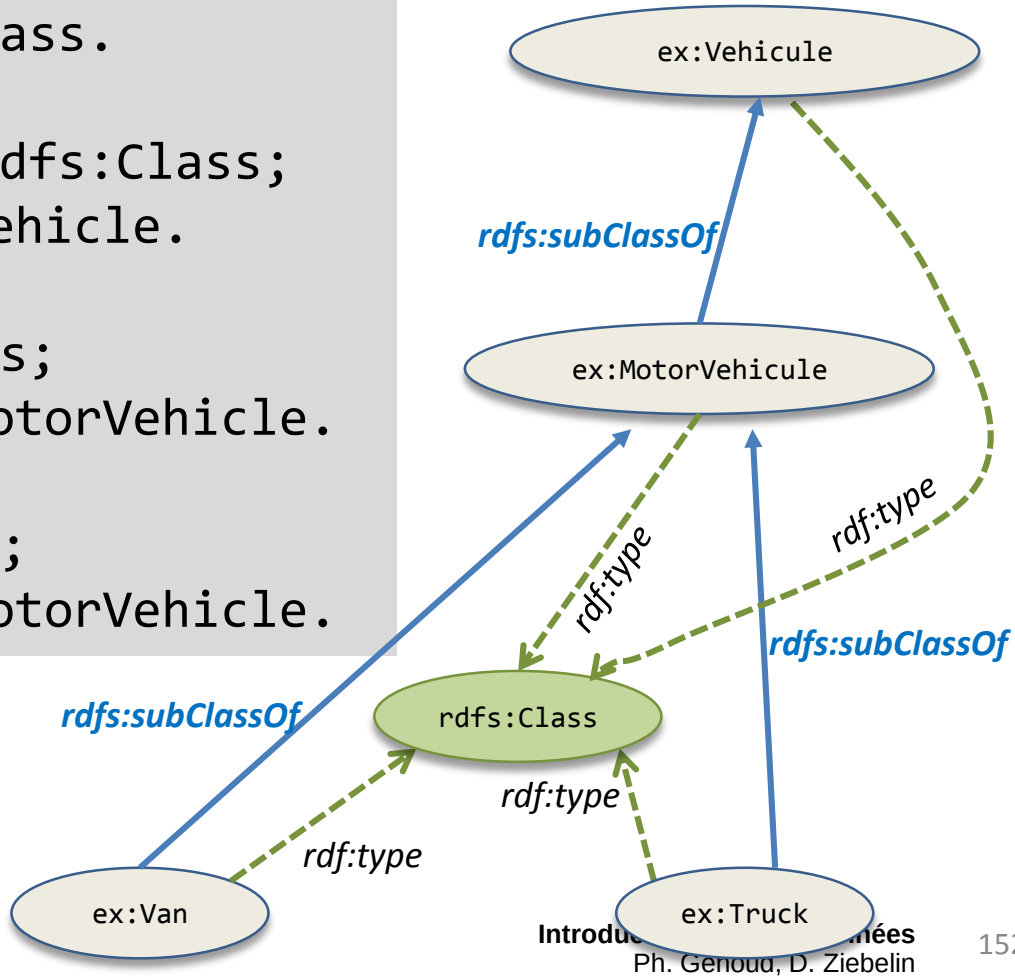
```
exthings:companyCar rdf:type ex:MotorVehicle , ex:BlackThing.
```

A resource may be an instance of more than one class.

RDF(S) Classes

- Specialization relationship between two classes is described using the predefined **rdfs:subClassOf** property

```
ex:Vehicle rdf:type rdfs:Class.

ex:MotorVehicule rdf:type rdfs:Class;
      rdfs:subClassOf ex:Vehicle.

ex:Truck rdf:type rdfs:Class;
      rdfs:subClassOf ex:MotorVehicle.

ex:Van rdf:type  rdfs:Class;
      rdfs:subClassOf ex:MotorVehicle.
```



RDF(S) Classes

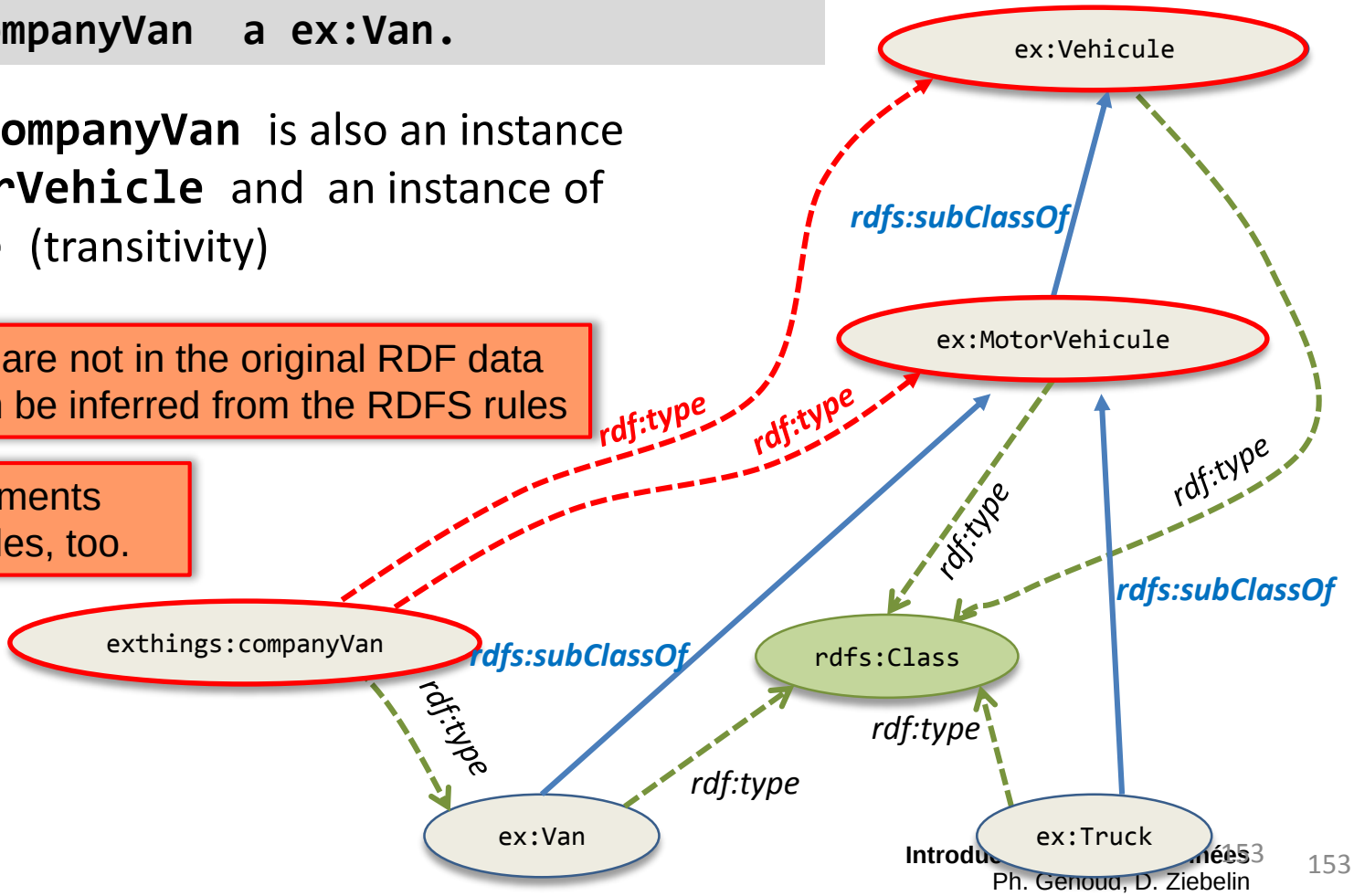
- **subClassOf** meaning : any instance of a subclass is also an instance of all the superclasses

```
exthings:companyVan  a ex:Van.
```

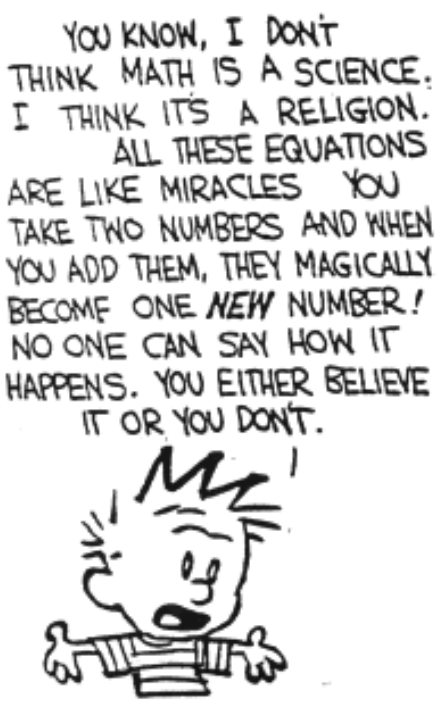
➔ **exthings:companyVan** is also an instance of **ex:MotorVehicle** and an instance of **ex:Vehicle** (transitivity)

Triples are not in the original RDF data but can be inferred from the RDFS rules

RDFS environments return that triples, too.



Inference: let us be formal…



- The RDF Semantics document has a list of (33) entailment rules:
  - "if such and such triples are in the graph, add this and this"
  - do that recursively until the graph does not change
- The relevant rule for our example:

- Type propagation

```
If:
  C2 rdfs:subClassOf C1.
  x rdf:type C2.
Then add:
  x rdf:type C1.
```

RDFS Class semantics (…)

- Any rdfs:Class is a subclass of rdfs:Resource

```
If:
  C rdf:type rdfs:Class.
 Then add:
  C rdfs:subClassOf rdfs:Resource.
```

- Subsumption Reflexivity

```
If:
  C rdf:type rdfs:Class.
Then add:
  C rdfs:subClassOf C.
```

- Subsumption Transitivity

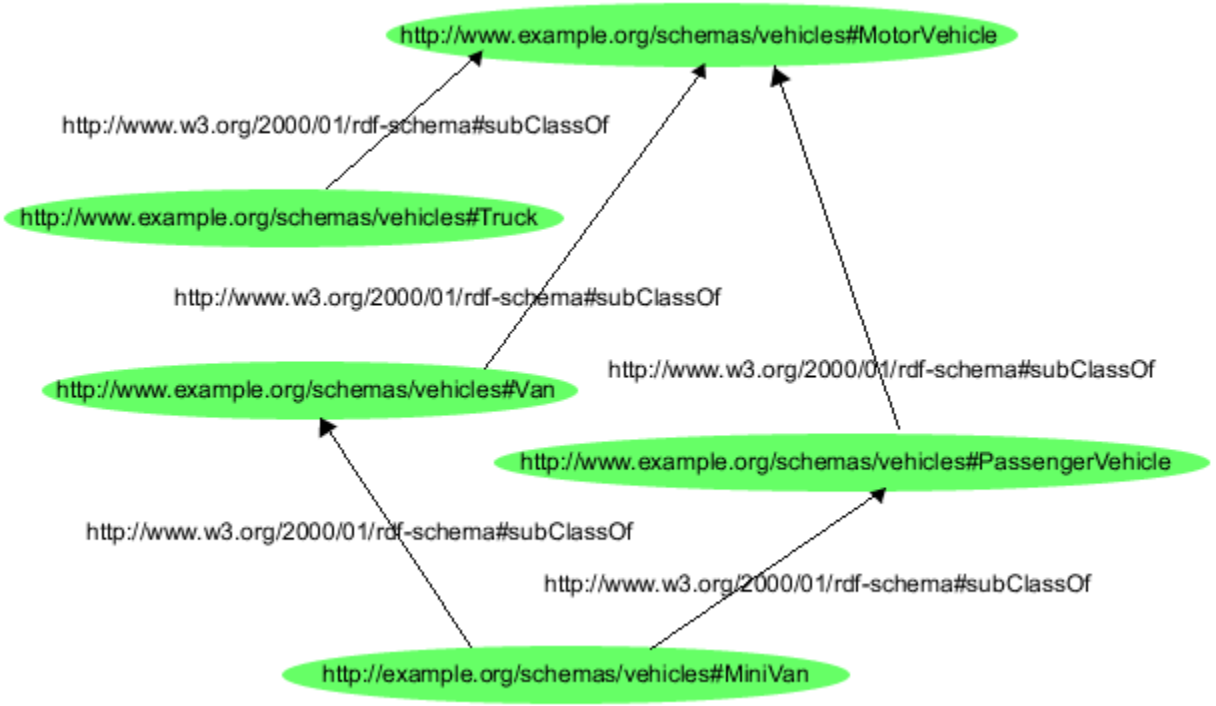
```
If:
  C2 rdfs:subClassOf C1.
  C3 rdfs:subClassOf C2.
Then add:
  C3 rdfs:subClassOf C1.
```

Enables to model the fact that 2 classes contains the same individuals (are extensionally equivalent) by establishing a mutual subclass relationship

```
C1 rdfs:subClassOf C2
C2 rdfs:subClassOf C1
```

RDF(S) Classes

- A class may be a subclass of more than one class




```
ex:MotorVehicle       rdf:type          rdfs:Class .
ex:PassengerVehicle   rdf:type          rdfs:Class ;
                      rdfs:subClassOf   ex:MotorVehicle .
ex:Van                rdf:type          rdfs:Class ,
                      rdfs:subClassOf   ex:MotorVehicle .
ex:Truck              rdf:type          rdfs:Class;
                      rdfs:subClassOf   ex:MotorVehicle .
ex:MiniVan            rdf:type          rdfs:Class ;
                      rdfs:subClassOf   ex:Van, ex:PassengerVehicle .
```

RDF Schema (RDF(S))

- Introduction
- RDF(S) Classes
- RDF(S) Properties
- Interpreting RDF(S) Schema Declarations
- More RDF(S) properties
- Conclusion

RDF(S) Properties

- All properties in RDF are described as instances of class **`rdf:Property`**.

- RDF(S) properties **`rdfs:domain`**, **`rdfs:range`**, and **`rdfs:subPropertyOf`** describe how properties and classes are intended to be used together in RDF data.
  - **`rdfs:domain`** specifies the type of all individuals that are the subject of statements using the described property.
  - **`rdfs:range`** specifies the type of all individuals or the datatype of all literals that are the object of statements using the described property.

RDF(S) Properties

- example of a property with a domain and range

```
ex:Person    rdf:type       rdfs:Class .
ex:Book      rdf:type       rdfs:Class .

ex:author    rdf:type       rdf:Property;
             rdfs:domain    ex:Book;
             rdfs:range     ex:Person .
```

```
ex:SemanticWebForDummies ex:author ex:JeffPollock.
```

```
ex:SemanticWebForDummies rdf:type ex:Book.
ex:JeffPollock rdf:type ex:Person.
```

- example of a property with a datatype range

```
ex:price     rdf:type       rdf:Property;
             rdfs:range     xsd:Integer .
```

RDF(S) Properties

- A property can have zero, one, or more than one range property
  - **0 range property :** nothing is said about the values of the property. Anything (resource or literal) can be used as an object of the property
  - **1 range property :** this says that the objects of the property are instances (literals) of the given class (datatype)
  - **more than 1 range property :** this says that the objects of the property are instances (literals) of **all** classes (datatypes) specified as a range

```
ex:hasMother    rdfs:range    ex:Female ;
                rdfs:range    ex:Person .
```

```
exstaff:frank    ex:hasMother    exstaff:frances .
```

must be both an instance of `ex:Female` and of `ex:Person`.

RDF(S) Properties

- A property can have zero, one, or more than one domain property
  - **0 domain property:** nothing is said about the subjects of the property. Any resource can be used as a subject of the property
  - **1 domain property :** this says that the subjects of the property are instances of the given class
  - **more than 1 domain property** : this says that the subjects of the property are instances of **all** classes specified as a range

```
ex:isMotherOf   rdfs:domain   ex:Female ;
                rdfs:domain   ex:Person .
```

```
exstaff:frances   ex:isMotherOf   exstaff:frank .
```

must be both an instance of `ex:Female` and of `ex:Person`.

RDF(S) Properties

- the predefined **rdfs:subPropertyOf** property allows to describe specialization relationship between two properties

```
property1    rdfs:subPropertyOf    property2.
```

  - property1 is a specialization of property2
  - any two resources related using property1 are implicitly related by property2

```
ex:driver          rdf:type              rdf:Property .
ex:primaryDriver   rdf:type              rdf:Property ;
                   rdfs:subPropertyOf    ex:driver .
```

```
exstaff:fred ex:primaryDriver ex:companyVan
```

```
exstaff:fred ex:driver ex:companyVan
```

RDF(S) Properties

- A property may be a subproperty of zero, one or more properties
- All RDF Schema `rdfs:range` and `rdfs:domain` properties that apply to an RDF property also apply to each of its subproperties.

```
ex:driver           rdf:type            rdf:Property ;
                    rdf:range           ex:Person;
                    rdf:domain          ex:Vehicle.
ex:primaryDriver    rdf:type            rdf:Property ;
                    rdfs:subPropertyOf  ex:driver .
```

```
exstaff:fred ex:primaryDriver ex:companyVan.
```

```
exstaff:fred ex:driver ex:companyVan.
exstaff:fred rdf:type ex:Person.
ex:companyVan rdf:type ex:Vehicle.
```

RDFS Property semantics

- Type inference (domain)

```
If:
  p rdfs:domain d.
  x p y
 Then add:
  x rdf:type d.
```

- Type inference (range)

```
If:
  p rdfs:range r.
  x p y
 Then add:
  y rdf:type r.
```

- Property inheritence

```
If:
  p2 rdfs:subPropertyOf p1.
  x p2 y.
Then add:
  x p1 y.
```

RDF(S) Properties

- by default properties are defined independently of classes descriptions

```
ex:hasParent a rdf:Property.
```

```
ex:Human a rdfs:Class.
```

```
ex:John a            ex:Human;
        ex:hasParent ex:Mary.
```

```
ex:Dog a rdfs:Class.
```

```
ex:Daisy a ex:Dog;
         ex:hasParent ex:Amber.
```

```
ex:Book a rdfs:Class.
```

```
ex:SemwebForDummies a ex:Book;
                    ex:hasParent ex:Jeff.
```

hasParent has a *global scope* an could then be used to describe instances of any class

this flexibility facilitates the use of property in situations that might not have been anticipated in the original description

but properties can be mis-applied in inappropriate situations

RDF(S) properties

- with rdfs:domain and rdfs:range you can "constraint" properties…

```
ex:hasParent a rdf:Property;
             rdfs:domain ex:Human;
             rdfs:range ex:Human.
```

```
ex:John a ex:Human;
        ex:hasParent ex:Mary.
```

- but this introduce inflexibility and should be used with caution

```
ex:Daisy a ex:Dog.
```

```
ex:Mary ex:hasParent ex:Daisy.
```

→ ex:Daisy a Dog **and** a Human ?

any range defined for an RDF property applies to all uses of the property

it is not possible in an RDF schema to define a specific property as having locally-different ranges depending on the class of the resource it is applied to.

RDF(S) Properties

```
ex:LivingOrganism a rdfs:Class.
```

```
ex:Human a rdfs:Class;
         rdfs:subClassOf ex:LivingOrganism.
```

```
ex:Dog   a rdfs:Class;
         rdfs:subClassOf ex:LivingOrganism.
```

```
ex:Cat   a rdfs:Class;
         rdfs:subClassOf ex:LivingOrganism.
```

```
ex:hasParent a rdf:Property;
             rdfs:domain ex:LivingOrganism;
             rdfs:range  ex:LivingOrganism.
```

Not enough :
A Dog can have a Cat as parent

```
ex:hasHumanParent a rdf:Property;
                  rdfs:subPropertyOf ex:hasParent;
                  rdfs:domain exHuman;
                  rdfs:range  ex:Human.
```

Add new properties specializing `hasParent`

```
ex:hasCatParent a rdf:Property;
                rdfs:subPropertyOf ex:hasParent;
                ...
```

RDF Schema (RDF(S))

- Introduction
- RDF(S) Classes
- RDF(S) Properties
- Interpreting RDF(S) Schema Declarations
- More RDF(S) properties
- Conclusion

Interpreting RDF(S) Schema Declarations

- RDF(S) descriptions are not (necessarily) *prescriptive* like OOP languages are.

→ RDF(S) doesn't prescribe how the statements should be used by an application

OOP Language (Java)

```
public Class Person{
   private String name;

   Person(String name) {
        this.name = name;
   }
}
```

every instance of Person has a property (attribute) name and this attribute is unique

this attribute is unique

instance of Person have no other property than name

RDF(S)

```
ex:Person a rdfs:Class.
```

```
ex:hasName a rdf:Property;
        rfs:domain ex:Person;
        rdfs:range xsd:String;
```

the property is not mandatory for instances of Person

```
ex:John a ex:Person;
        ex:hasName "Johnny Defool".
```

```
ex:Jane a rdfs:Person.
```

you can have multiple values

```
ex:John ex:hasName "Johnny".
```

you can have other properties

```
ex:John foaf:knows ex:Mary.
```

Interpreting RDF(S) Schema Declarations

- RDF(S) might be used in different ways according to applications

```
ex:hasParent a rdf:Property;
             rdfs:domain ex:Human;
             rdfs:range  ex:Human.
```

- Constraint
  - a Human does'nt have a parent →application error
  - a Human has a parent p but its type is unknown →application error
  - a Human has a parent p with an other type than Human →application error
- Inferences
  - a Human does'nt have a parent →application adds an unknown (Human) parent
  - a Human has a parent p but its type is unknown →application infers that p is of Human type
  - a Human has a parent p with an other type than Human →application infers that p is of Human type (in addition to other known types)
    - this can lead to inconsistencies that must be detected

RDF Schema (RDF(S))

- Introduction
- RDF(S) Classes
- RDF(S) Properties
- Interpreting RDF(S) Schema Declarations
- More RDF(S) properties
- Conclusion

more RDF(S) properties

- *rdfs:Resource* **rdfs:comment** *rdfs:Literal*
  - used to provide a human-readable description of a resource
- *rdfs:Resource* **rdfs:label** *rdfs:Literal*
  - a human-readable version of a resource's name
- *rdfs:Resource* **rdfs:seeAlso** *rdfs:Resource*
  - used to indicate a resource that might provide additional information about the subject resource
- *rdfs:Resource* **rdfs:isDefinedBy** *rdfs:Resource*
  - to indicate a resource defining the subject resource. This property may be used to indicate an RDF vocabulary in which a resource is described
- …

http://www.w3.org/TR/rdf-schema/

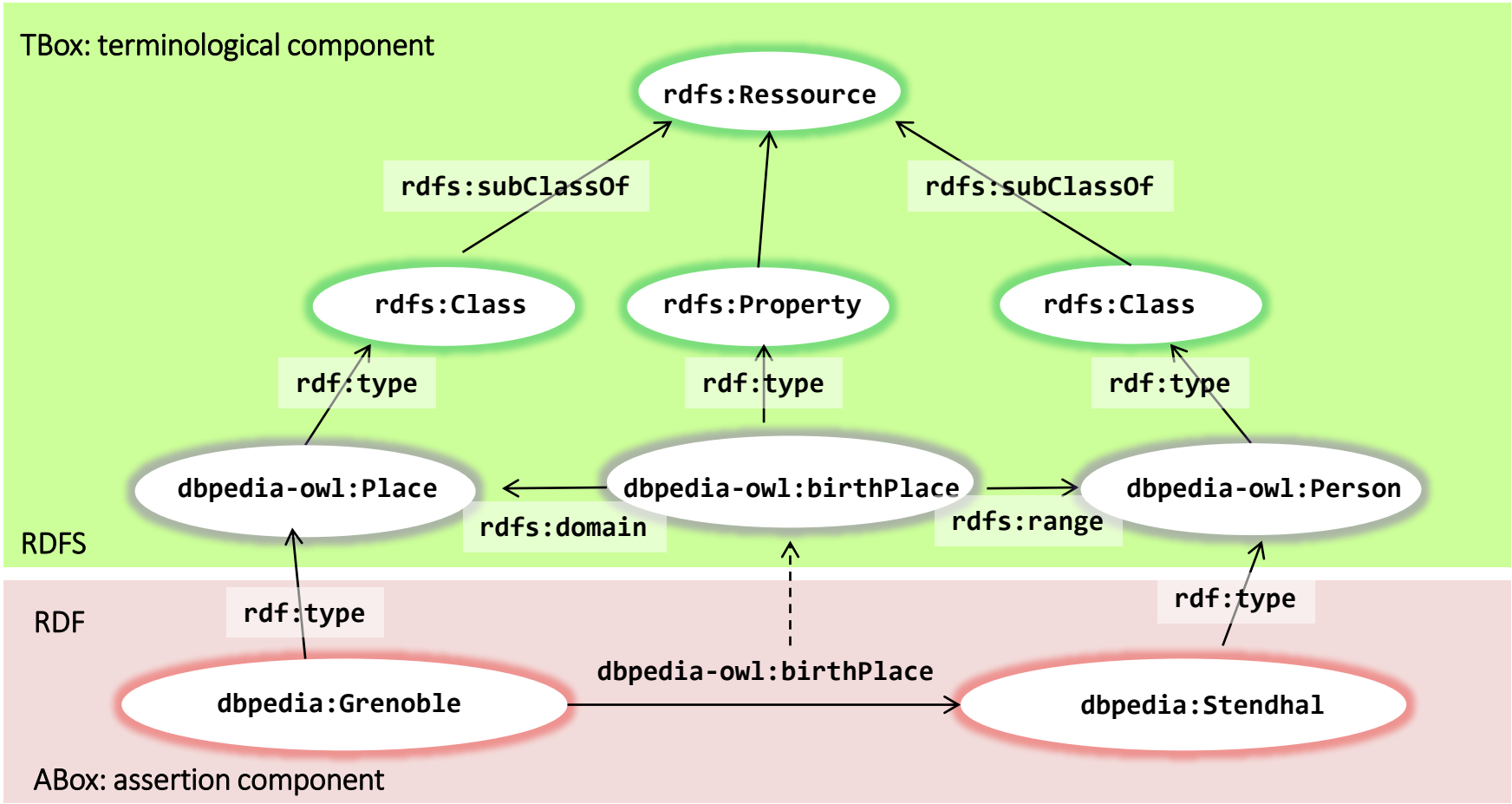
RDF Schema (RDF(S))

- Introduction
- RDF(S) Classes
- RDF(S) Properties
- Interpreting RDF(S) Schema Declarations
- more RDF(S) properties
- Conclusion

RDF Schema (RDFS)

Prefixes

```
dbpedia:      http://dbpedia.org/resource/
dbpedia-owl:  http://dbpedia.org/ontology/
rdfs:         http://www.w3.org/2000/01/rdf-schema#
```



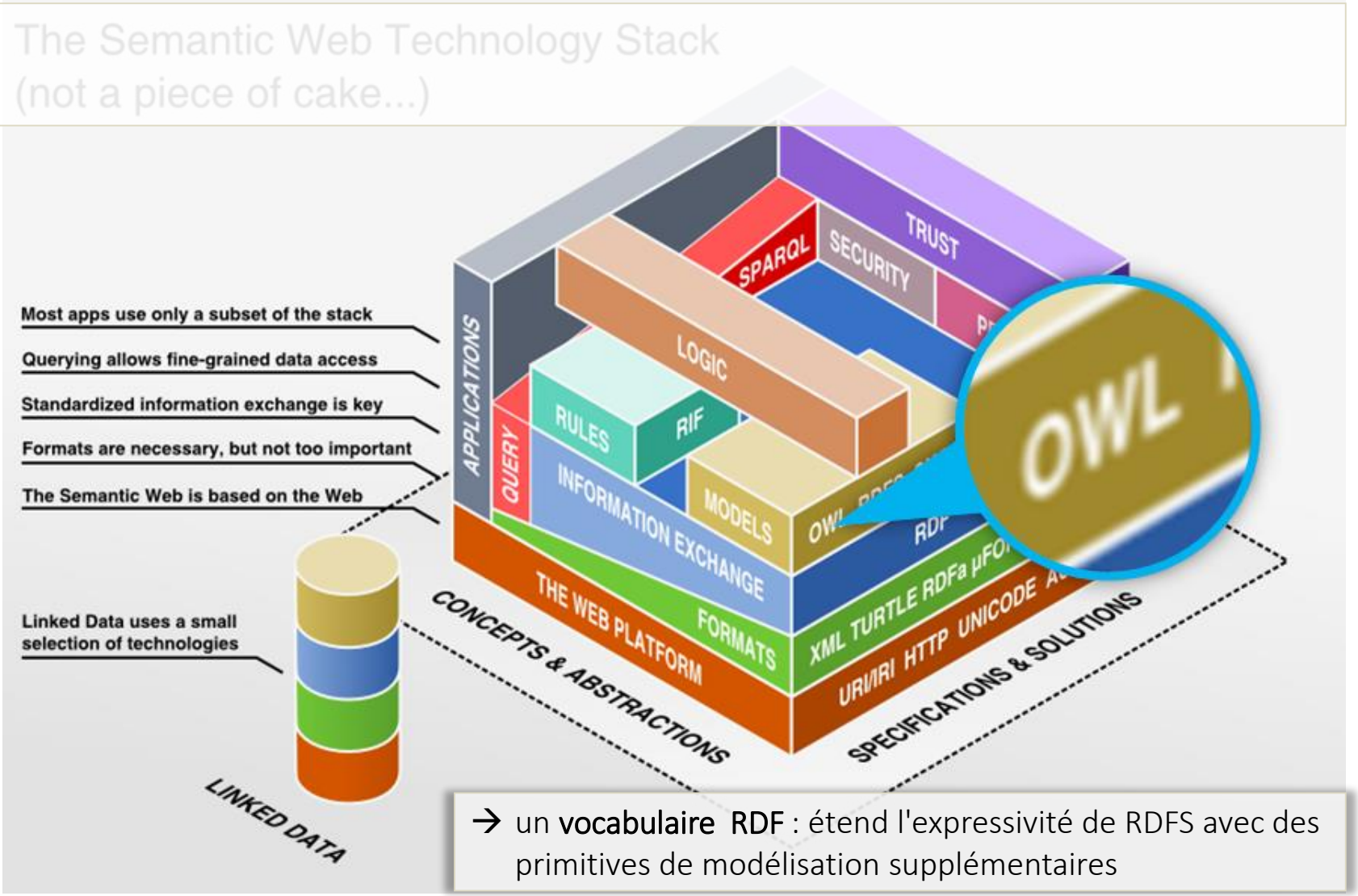
What does RDF(S) give us ?

- RDF
  - A mechanism for publishing data.
  - Single (simple) data model.
  - Syntactic consistency between names (IRIs).
  - Low level integration of data.
  - Mash the graphs together and we're done.
- RDF(S)
  - Ability to use simple schema/vocabularies when describing our resources.
  - Consistent vocabulary use and sharing.
  - Basic inference

Problems with RDF(S)

- RDF(S) is **too weak** to describe resources in sufficient detail
  - No **localized range and domain** constraints
    - Can't say that the range of `hasParent` is Person when applied to `Persons` and `Dog` when applied to `Dogs`
  - No **existence/cardinality** constraints
    - Can't say that all instances of `Person` have a mother that is also a Person, or that Persons have exactly 2 parents
  - No **transitive, inverse** or **symmetrical** properties
    - Can't say that `isPartOf` is a transitive property, that `hasPart` is the inverse of `isPartOf` or that `touches` is symmetrical
  - …

Web Ontology Language (OWL)



→ un vocabulaire RDF : étend l'expressivité de RDFS avec des primitives de modélisation supplémentaires

Web Ontology Language (OWL)

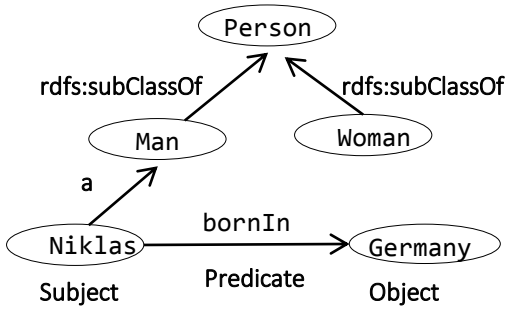
- a W3C standard
  - OWL 1 : W3C recommendation 10 Feb. 2004 http://www.w3.org/TR/owl-features/
  - OWL 2 : W3C recommendation 11 Dec. 2012 http://www.w3.org/TR/owl2-overview/

- OWL vocabulary : a set of primitives described in RDF that extends RDFS vocabulary
  - namespace **http://www.w3.org/2002/07/owl#** ⇔ **owl:**

- Far more expressive than RDFS
  - Classes can be describe by union, intersection, complement, properties restrictions.
  - notions of classes or properties equivalence, resources equality,
  - notions of inverse, symmetric, transitive … properties
  - properties cardinality…
- Formal specification (based on Description Logics)

  → support for automated reasonning

RDFS (RDF Schema):
Vocabulary for defining hierarchical classes and properties (taxonomies)



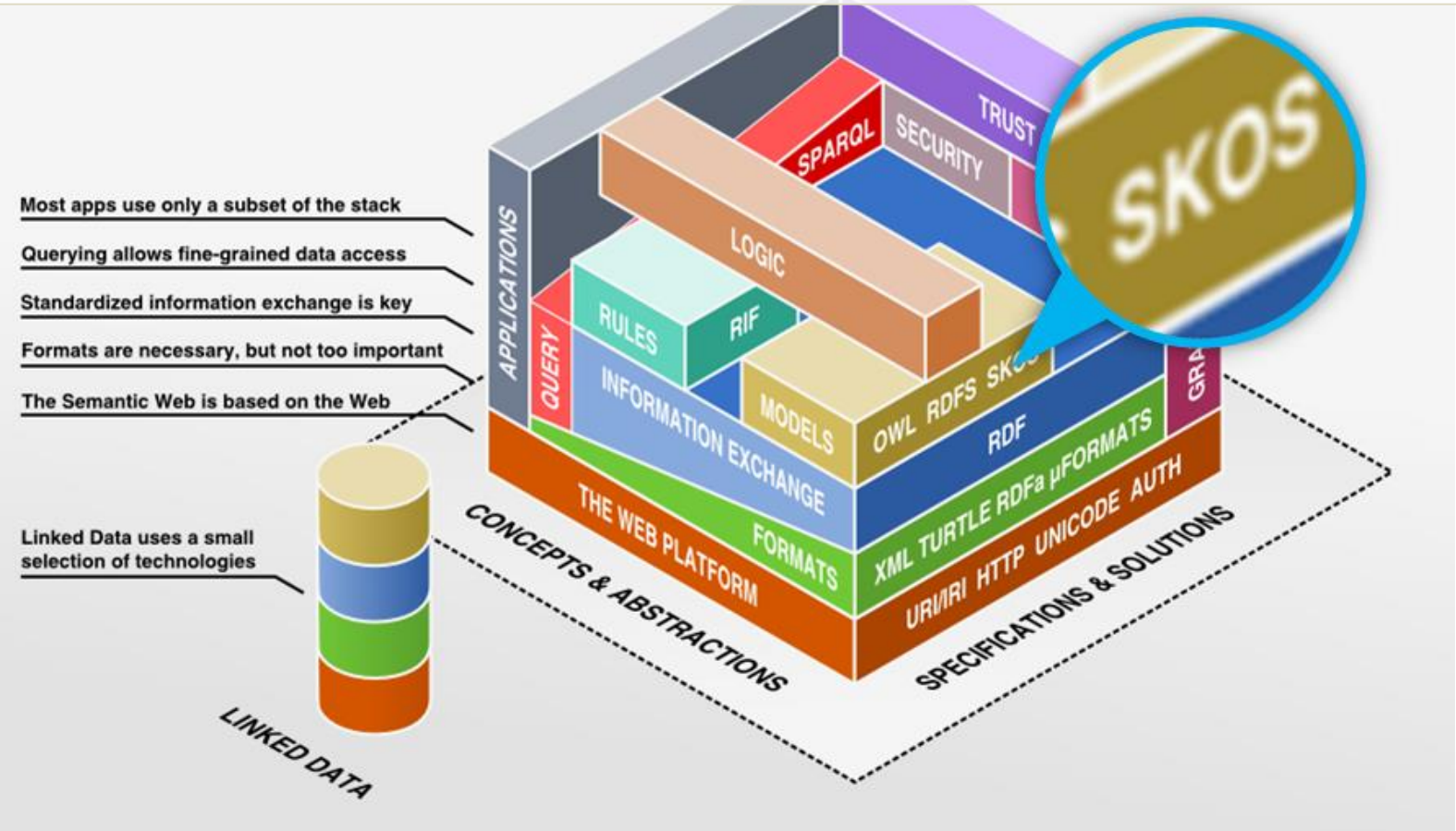
OWL (Web Ontology Language):
Ontology language based on DL (subset of 1rst Order Logic) to describe classes

$\texttt{Person} \equiv \texttt{Man} \cup \texttt{Woman}$
$\texttt{Man} \cap \texttt{Woman} \equiv \varnothing$

178

Simple Knowledge Organization System (SKOS)

modèle commun pour partager et lier (via RDF) sur le web différents systèmes d'organisation de connaissances tels thésaurus, taxinomies, systèmes de classification, système d'index.



Simple Knowledge Organization System (SKOS)

- OWL → Ontologie SKOS → thesaurus

  - *"l'objectif d'un thésaurus est de constituer des vocabulaires normalisés et d'organiser la liste des termes de ces vocabulaire sans forcément les définir, dans le but notamment d'indexer un corpus de ressources documentaires et de faciliter les recherches dans ce corpus"* Le web Sémantique, Fabien Gandon, Catherine Faron-Zucker, Olivier Corby, ed. Dunod 2012

- Hierarchical relationships (d'après http://fr.wikipedia.org/wiki/Thésaurus)
  - Generic term (BT: broader term) , specific term (NT: narrower term).
  - Partitives relationships (whole-part relations) , instantiation relationships (to give examples)
- Associative relationships
  - RT: related term.
- Equivalenct terms
  - …

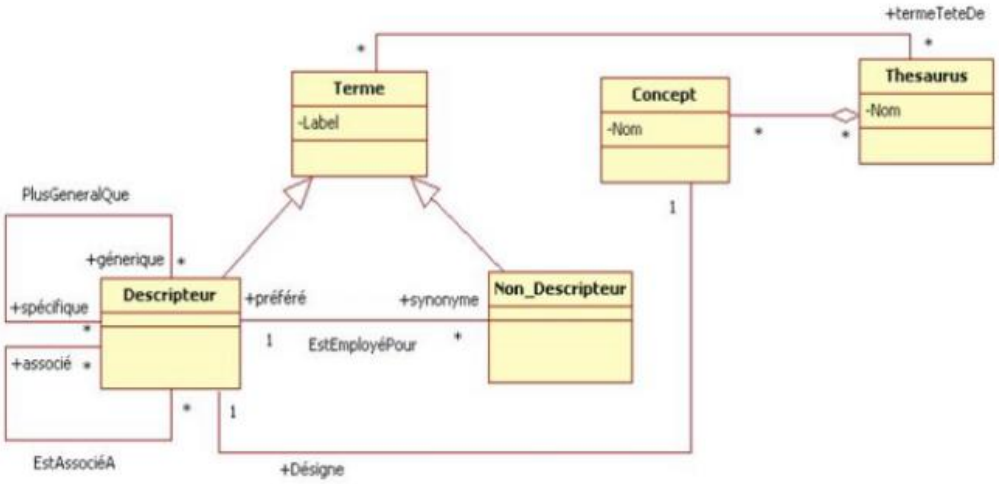
Simple Knowledge Organization System (SKOS)

• W3C Standard (recommandation) :

SKOS Simple Knowledge Organization System Reference (August 2009)
http://www.w3.org/TR/2009/REC-skos-reference-20090818/

"SKOS a common data model for sharing and linking knowledge organization systems via the Web. Many knowledge organization systems, such as thesauri, taxonomies, classification schemes and subject heading systems, share a similar structure, and are used in similar applications. SKOS captures much of this similarity and makes it explicit, to enable data and technology sharing across diverse applications

The SKOS data model provides a standard, low-cost migration path for porting existing knowledge organization systems to the Semantic Web. SKOS also provides a lightweight, intuitive language for developing and sharing new knowledge organization systems. It may be used on its own, or in combination with formal knowledge representation languages such as the Web Ontology language (OWL)"



• Technical notes:

– SKOS Simple Knowledge Organization System Primer http://www.w3.org/TR/2009/NOTE-skos-primer-20090818/

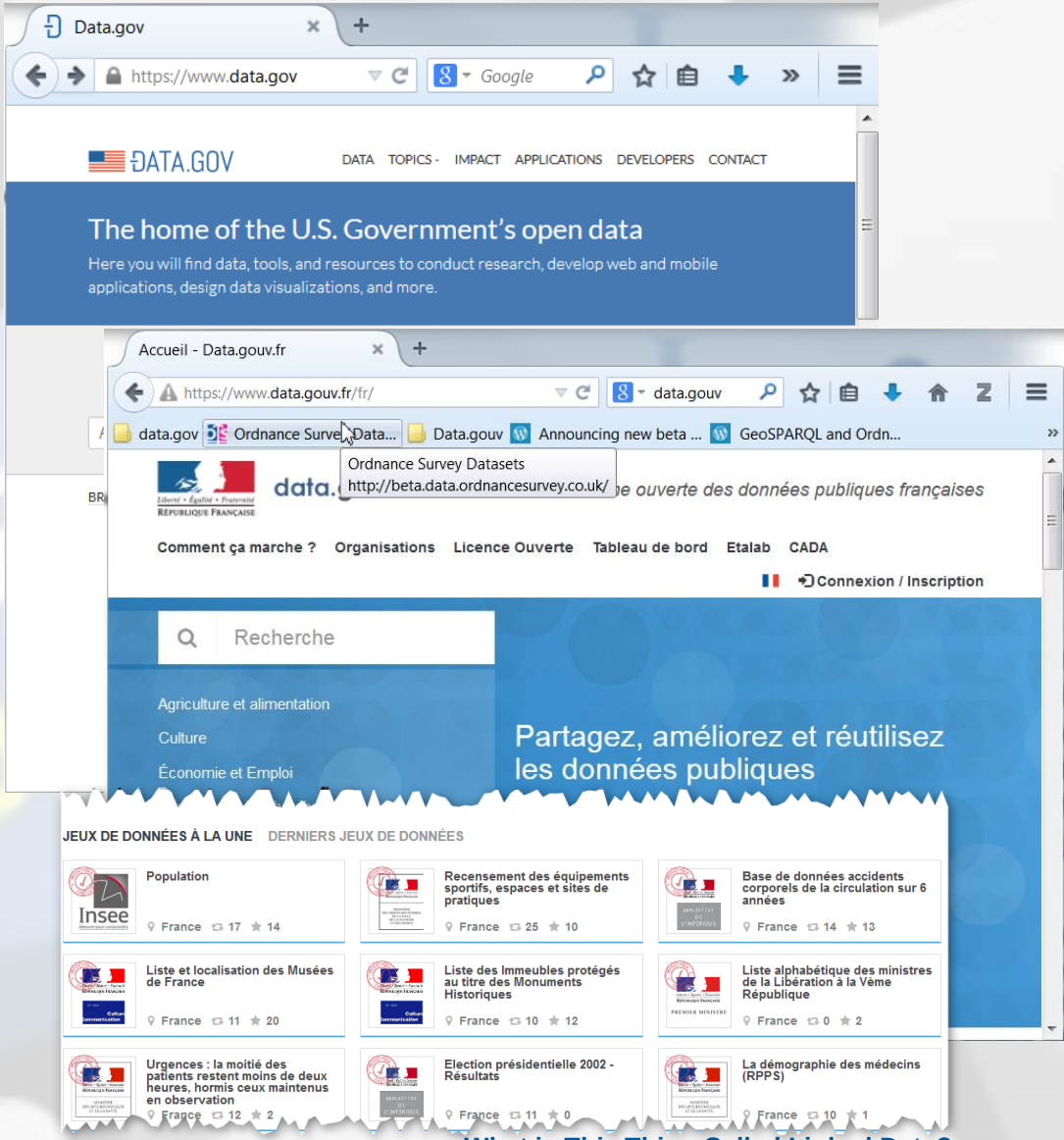
– SKOS Use Cases and Requirements http://www.w3.org/TR/2009/NOTE-skos-ucr-20090818/

Outline

- "Theoretical" Session (morning)
  - Introduction
  - Distributing Data on the web with RDF
    - Naming the Data : URIs (Uniform Resources Identifiers)
    - The RDF Data model
  - Querying Linked Data with SPARQL
  - Semantic modelling
    - RDFS
    - OWL
  - From Open Data to Linked Open Data
  - Conclusion
- Hands-on session (afternoon)
  - From a CSV file to linked data
  - Querying linked data (SPARQL)

Open Data

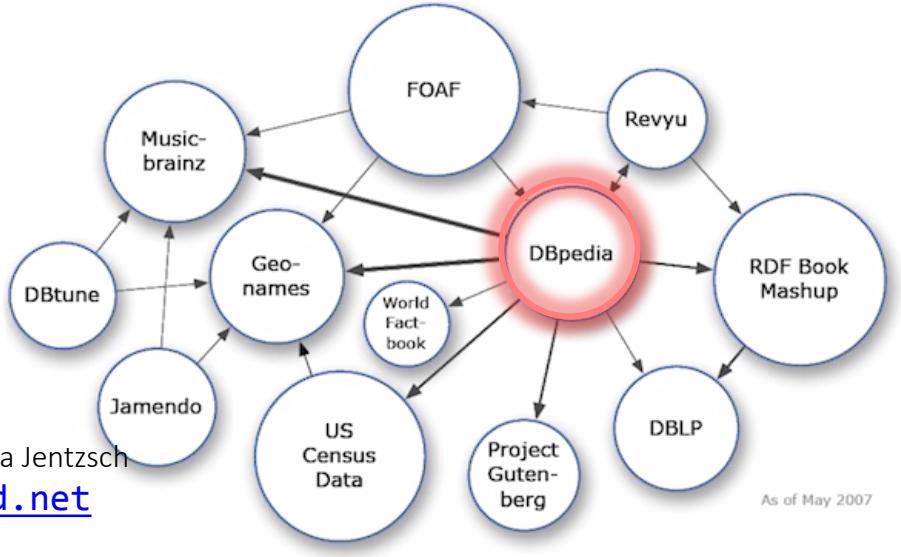
- Open Data Movement
  - *"A piece of content or data is open if anyone is free to use, reuse, and redistribute it — subject only, at most, to the requirement to attribute and/or share-alike."* http://opendefinition.org/
  - an old idea but a recent term gaining popularity
    - with the rise of the Internet and World Wide Web
    - with the launch of open-data government initiatives such as Data.gov (USA), data.gouv.fr (FR)…

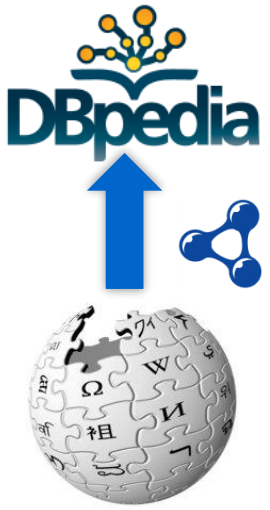


Linked Open Data

The Open Data Movement aims at making data freely available to everyone. There are already various interesting open data sets available on the Web. Examples include Wikipedia, Wikibooks, Geonames, MusicBrainz, WordNet, the DBLP bibliography and many more which are published under Creative Commons or Talis licenses.

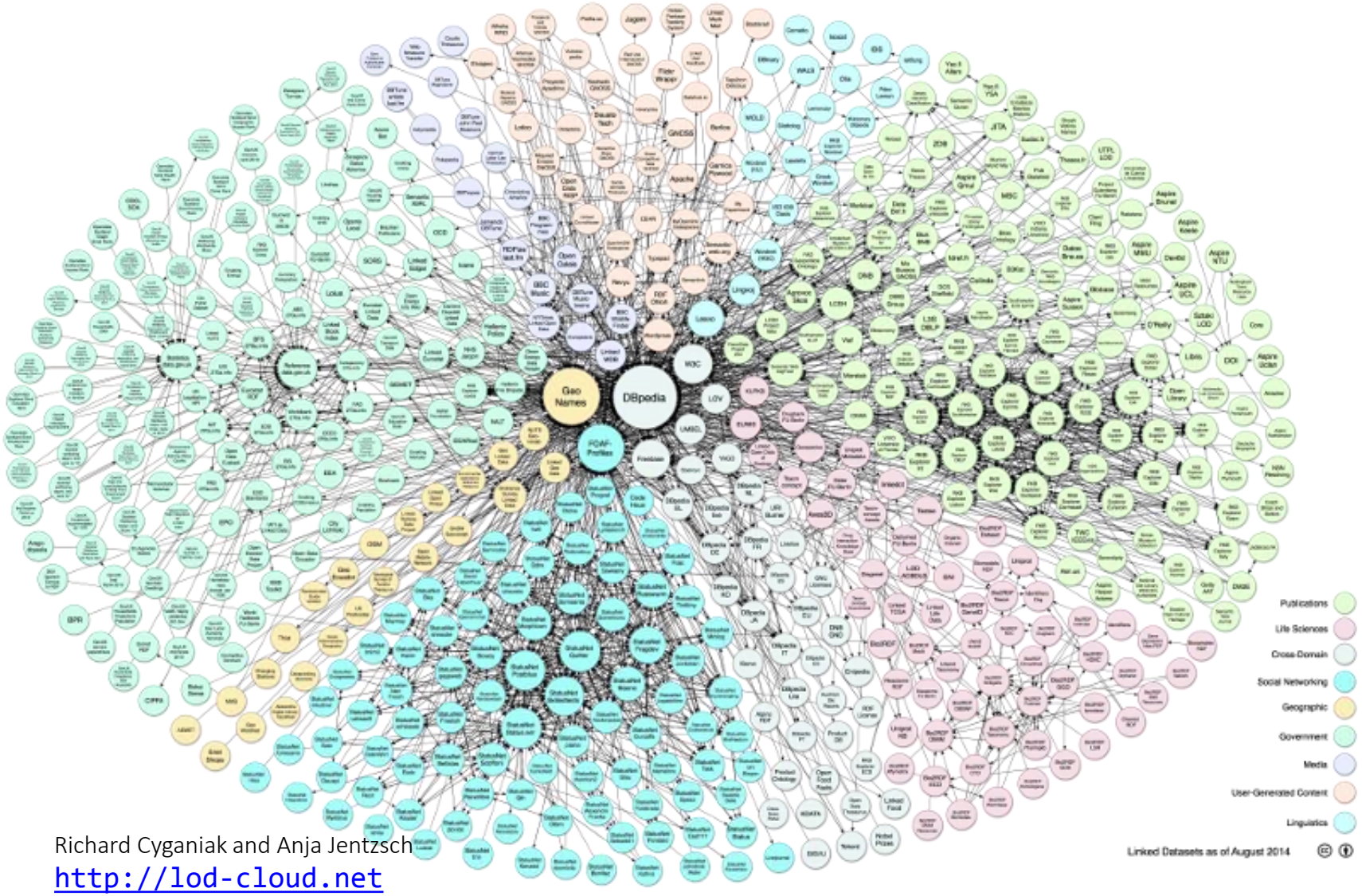
The goal of the W3C SWEO Linking Open Data community project is to extend the Web with a data commons by publishing various open data sets as RDF on the Web and by setting RDF links between data items from different data sources.





Richard Cyganiak and Anja Jentzsch
http://lod-cloud.net

Linked Open Data



Richard Cyganiak and Anja Jentzsch

http://lod-cloud.net

From Open Data to Linked Open Data

- 2006: defines basic principles for publishing Linked Data
- 2010: added a 5 star rating system for Linked Open Data (LOD)

"in order to encourage people -- especially government data owners -- along the road to good linked data…"

Tim Berners-Lee
http://www.w3.org/DesignIssues/LinkedData.html

★ Open Data

make your stuff available on the Web (whatever format)
under an open license

Temperature forecast for Galway, Ireland

Day	Lowest Temperature (°C)
Saturday, 13 November 2010	2
Sunday, 14 November 2010	4
Monday, 15 November 2010	7

Michael Hausenblas http://5stardata.info/

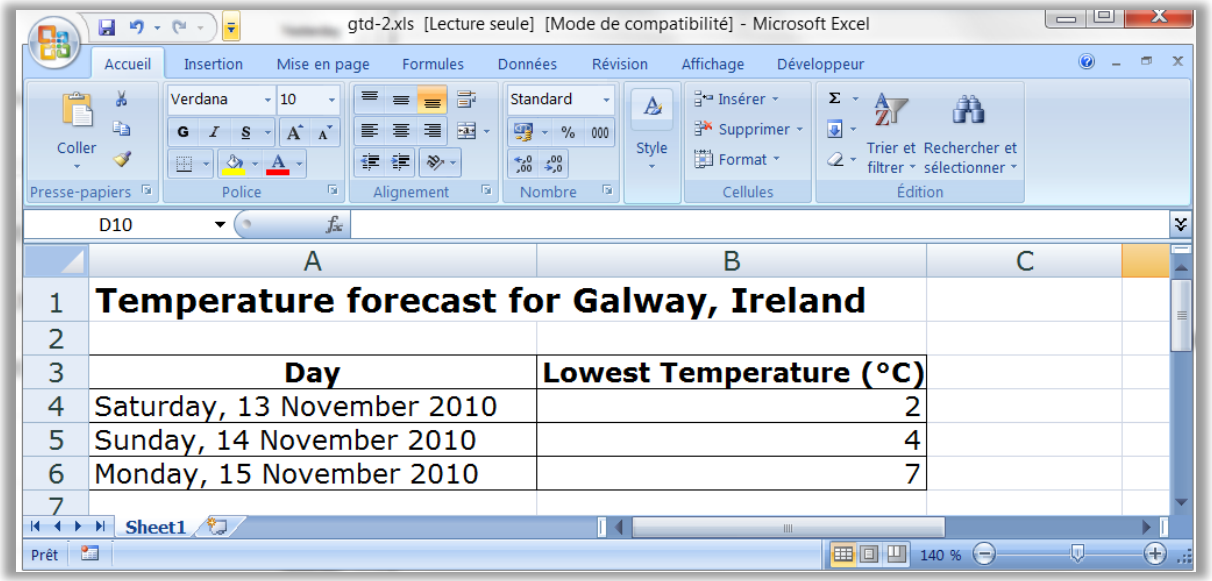
Costs & benefits of ★ Web data

- As a consumer ...
  - ✓ You can look at it.
  - ✓ You can print it.
  - ✓ You can store it locally (on your hard drive or on an USB stick).
  - ✓ You can enter the data into any other system.
  - ✓ You can change the data as you wish.
  - ✓ You can share the data with anyone you like.
- As a publisher ...
  - ✓ It's simple to publish.
  - ✓ You do not have to explain repeatedly to others that they can use your data.

Michael Hausenblas http://5stardata.info/

★★ Open Data

make it available as structured data (e.g., Excel instead of image scan of a table)



Michael Hausenblas http://5stardata.info/

Costs & benefits of ★★ Web data

- As a consumer, you can do all what you can do with ★ Web data and additionally:
  - ✓ You can directly process it with proprietary software to aggregate it, perform calculations, visualise it, etc.
  - ✓ You can export it into another (structured) format.
- As a publisher ...
  - ✓ It's still simple to publish.

★★★ Open Data

use non-proprietary formats (e.g., CSV instead of Excel)

gtd-3.csv

```
1  "Temperature forecast for Galway, Ireland",
2
3  "Day","Lowest Temperature (C)"
4  "Saturday, 13 November 2010",2
5  "Sunday, 14 November 2010",4
6  "Monday, 15 November 2010",7
```

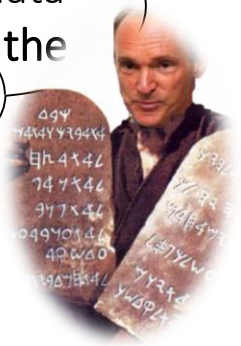
length : 164 lines : 6 | Ln : 1 Col : 1 Sel : 0 | Macintosh | ANSI as UTF-8 | INS

Costs & benefits of ★★★ Web data

- As a consumer, you can do all what you can do with ★ ★Web data and additionally:
  - ✓ You can manipulate the data in any way you like, without being confined by the capabilities of any particular software.
- As a publisher ...
  - ⚠ You might need converters or plug-ins to export the data from the proprietary format.
  - ✓ It's still rather simple to publish.

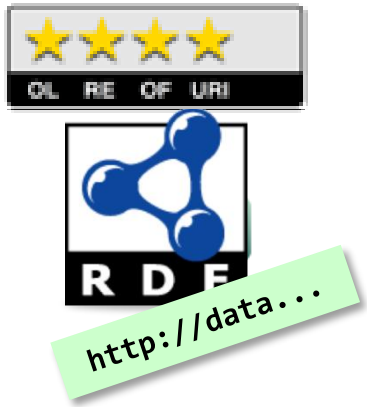
Is ★★★ open data enough ?

Excellent! The data is not only available via the Web but now everyone can use the data easily. **On the other hand, it's still data on the Web and not data in the Web*.**



**Data and the Web – a great many of choices*. Woddiscovery (M. Hausenblas' blog)· 2010-03-01
https://webofdata.wordpress.com/2010/03/01/data-and-the-web-choices/

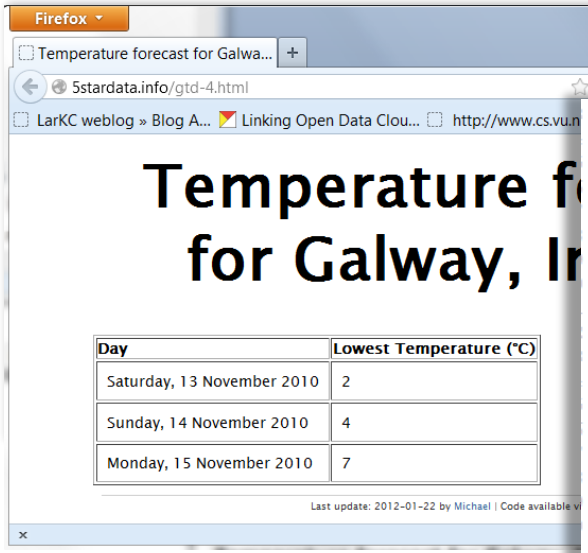
★★★★ Linked data



1. Use URIs as names for things.
2. Use HTTP URIs, so that people can look up those names.
3. When someone looks up a URI, **provide useful information, using the standards**(RDF, SPARQL).

RDFa = RDF in attributes

- to add metadata (RDF annotations) in (X)HTML documents
- use existing attributes (e.g. `href`, `src`) et introduce new ones (`vocab`, `typeof`, `property`, `resource`, `prefix`)




```
<h1 property="dcterms:title">Temperature forecast for Galway, Ireland</h1>

<div id="data" about="#Galway" typeof="meteo:Place">
<table border="1px">
    <tr>
        <th>Day</th>
        <th>Lowest Temperature (&deg;C)</th>
    </tr>
    <tr rel="meteo:forecast" resource="#forecast20101113">
        <td>
            <div about="#forecast20101113">
                <span property="meteo:predicted" content="2010-11-13T00:00:00Z" datatype="xsd:dateTime">Saturda
            </div>
        </td>
        <td rel="meteo:temperature">
            <div about="#temp20101113">
                <span property="meteo:celsius" datatype="xsd:decimal">2</span>
            </div>
        </td>
    </tr>
```

W3C XHTML + RDFa

Michael Hausenblas http://5stardata.info/

RDFa

```
<h1 property="dcterms:title">Temperature forecast for Galway, Ireland</h1>

<div id="data" about="#Galway" typeof="meteo:Place">
<table border="1px">
    <tr>
        <th>Day</th>
        <th>Lowest Temperature (&deg;C)</th>
    </tr>
    <tr rel="meteo:forecast" resource="#forecast20101113">
        <td>
            <div about="#forecast20101113">
                <span property="meteo:predicted" content="2010-11-13T00:00:00Z" datatype="xsd:dateTime">Saturday, 13 November 2010</span>
            </div>
        </td>
        <td rel="meteo:temperature">
            <div about="#temp20101113">
                <span property="meteo:celsius" dat
            </div>
        </td>
    </tr>
```

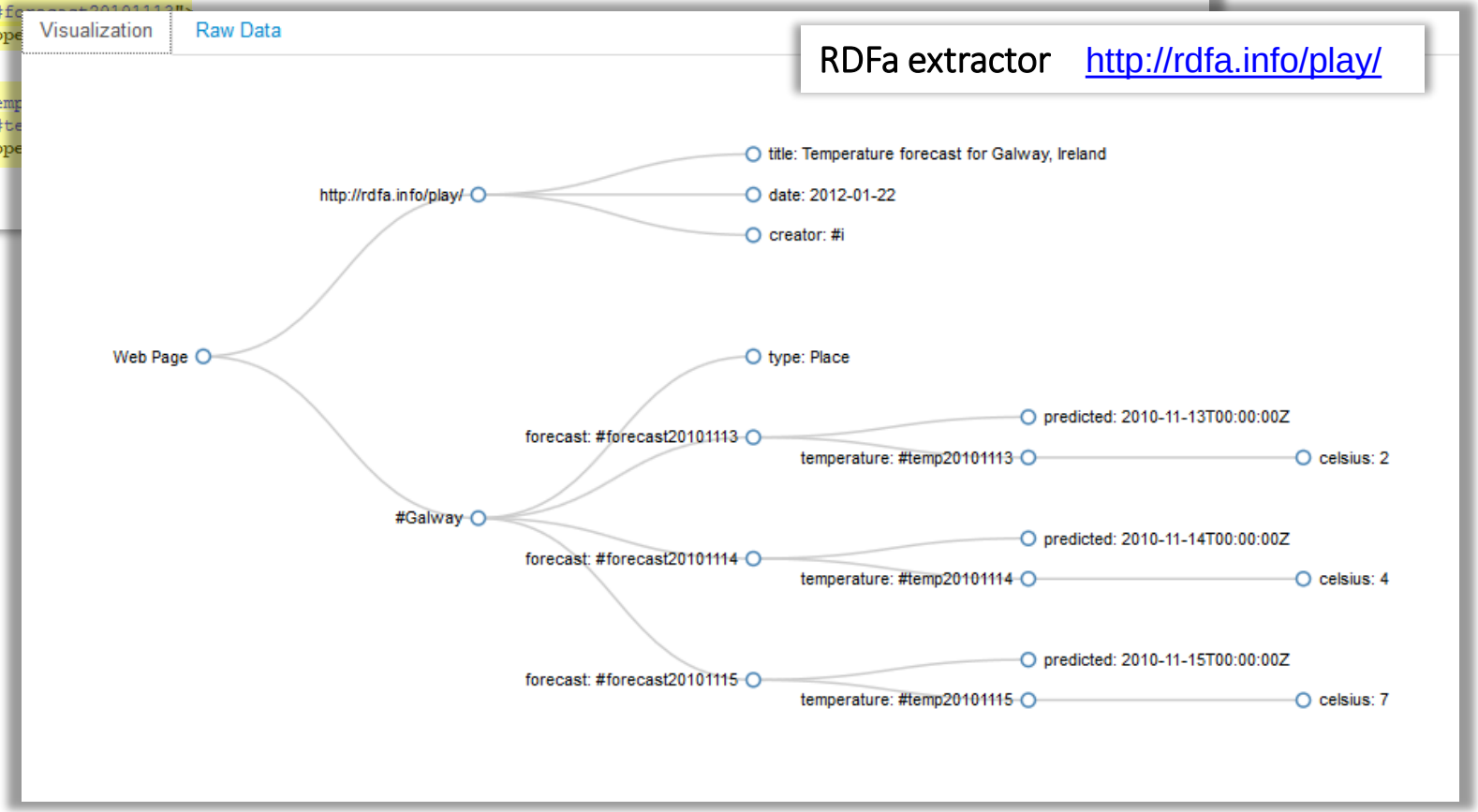
Extracted RDF

```
@prefix dc: <http://purl.org/dc/terms/> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

<http://rdfa.info/play/>
   dc:title "Temperature forecast for Galway, Ireland";
   dc:date "2012-01-22"^^xsd:date;
   dc:creator <http://sw-app.org/mic.xhtml#i> .
<http://rdfa.info/play/#Galway>
   rdf:type <http://purl.org/ns/meteo#Place>;
   <http://purl.org/ns/meteo#forecast> <http://rdfa.info/play/#forecast20101113>;
   <http://purl.org/ns/meteo#forecast> <http://rdfa.info/play/#forecast20101114>;
   <http://purl.org/ns/meteo#forecast> <http://rdfa.info/play/#forecast20101115> .
<http://rdfa.info/play/#forecast20101113>
   <http://purl.org/ns/meteo#predicted> "2010-11-13T00:00:00Z"^^xsd:dateTime;
   <http://purl.org/ns/meteo#temperature> <http://rdfa.info/play/#temp20101113> .
<http://rdfa.info/play/#temp20101113>
   <http://purl.org/ns/meteo#celsius> "2"^^xsd:decimal .
<http://rdfa.info/play/#forecast20101114>
   <http://purl.org/ns/meteo#predicted> "2010-11-14T00:00:00Z"^^xsd:dateTime;
   <http://purl.org/ns/meteo#temperature> <http://rdfa.info/play/#temp20101114> .
<http://rdfa.info/play/#temp20101114>
   <http://purl.org/ns/meteo#celsius> "4"^^xsd:decimal .
<http://rdfa.info/play/#forecast20101115>
   <http://purl.org/ns/meteo#predicted> "2010-11-15T00:00:00Z"^^xsd:dateTime;
   <http://purl.org/ns/meteo#temperature> <http://rdfa.info/play/#temp20101115> .
<http://rdfa.info/play/#temp20101115>
   <http://purl.org/ns/meteo#celsius> "7"^^xsd:decimal .
```

Michael Hausenblas http://5stardata.info/

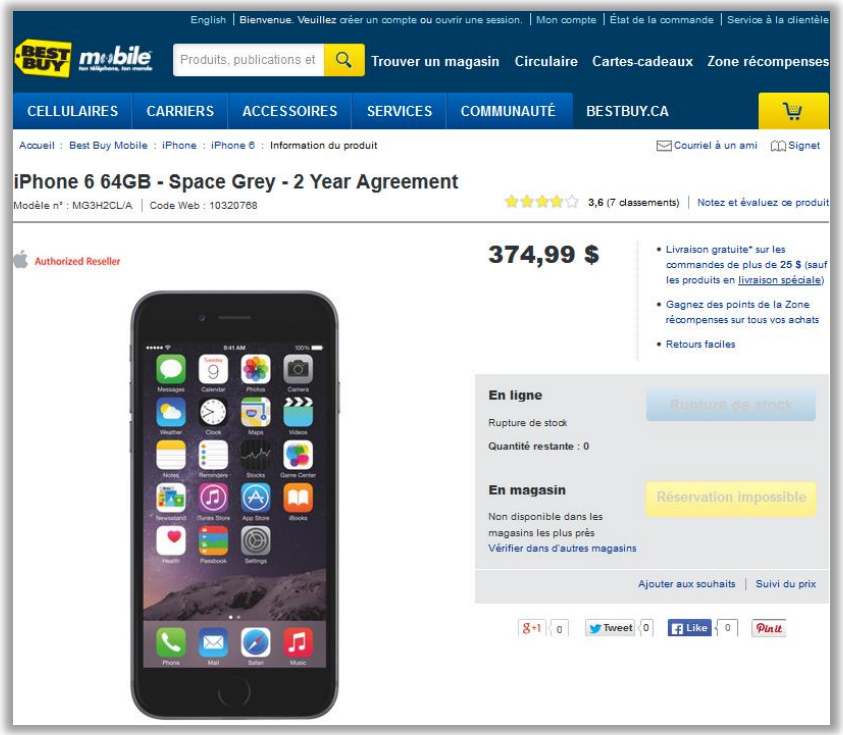
RDFa

```
<h1 property="dcterms:title">Temperature forecast for Galway, Ireland</h1>

<div id="data" about="#Galway" typeof="meteo:Place">
<table border="1px">
    <tr>
        <th>Day</th>
        <th>Lowest Temperature (&deg;C)</th>
    </tr>
    <tr rel="meteo:forecast" resource="#forecast20101113">
        <td>
            <div about="#f
                <span prope
            </div>
        </td>
        <td rel="meteo:temp
            <div about="#te
                <span prope
            </div>
        </td>
    </tr>
```



Michael Hausenblas http://5stardata.info/

RDFa



RDFa used par :

- Google, Yahoo
- Facebook, MySpace, LinkedIn
- Best Buy, O'Reilly …
- Newsweek, BBC
- WhiteHouse.gov, UK government, Library of Congress

…

Costs & benefits of ★★★★ Web data

- As a consumer, you can do all what you can do with ★ ★ ★ Web data and additionally:
  - ✓ You can link to it from any other place (on the Web or locally).
  - ✓ You can bookmark it.
  - ✓ You can reuse parts of the data.
  - ✓ You may be able to reuse existing tools and libraries, even if they only understand parts of the pattern the publisher used.
  - ⚠ Understanding the structure of an RDF "Graph" of data can be more effort than tabular (Excel/CSV) or tree (XML/JSON) data.
  - ✓ You can combine the data safely with other data. URIs are a global scheme so if two things have the same URI then it's intentional, and if so that's well on it's way to being 5 star data!

Michael Hausenblas http://5stardata.info/

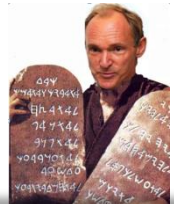
Costs & benefits of ★★★★ Web data

- As a publisher ...
  - ✓ You have fine-granular control over the data items and can optimise their access (load balancing, caching, etc.)
  - ✓ Other data publishers can now link into your data, promoting it to 5 star!
  - ⚠ You typically invest some time slicing and dicing your data.
  - ⚠ You'll need to assign URIs to data items and think about how to represent the data.
  - ⚠ You need to either find existing patterns to reuse or create your own.

Michael Hausenblas http://5stardata.info/

★★★★★ Linked Open Data

4. Include links to other URIs, so that they can discover more things.



```
<h1 property="dcterms:title">Temperature forecast for Galway, Ireland</h1>
<div id="data" about="#Galway" typeof="meteo:Place">
<span rel="owl:sameAs" resource="http://dbpedia.org/resource/Galway"></span>
<table border="1px">
    <tr>
        <th>Day</th>
        <th>
            <div about="#temp">
                Lowest
                <a rel="rdfs:seeAlso" href="
                (<span rel="owl:sameAs" reso
            </div>
        </th>
    </tr>
    <tr rel="meteo:forecast" resource="#fore
        <td>
            <div about="#forecast20101113">
                <span property="meteo:predict
            </div>
        </td>
        <td rel="meteo:temperature">
            <div about="#temp20101113">
                <span property="meteo:celsius
            </div>
```

dbpedia.org/page/Galway

linked geodata

Ontologies and Vocabu... What is URI (Uniform R... Bookmarks

About: Galway

An Entity of Type : populated place, from Named Graph : http://dbpedia.org, within Data Space : dbpedia.org

DBpedia

Galway est une ville de la province de Connacht, dans le Comté de Galway, sur la côte ouest de l'Irlande. Son nom vient de la rivière Corrib (Gaillimh) qui traverse la ville. Elle est également surnommée la « ville des tribus » en référence aux quatorze tribus qui se partageaient la ville à l'époque anglo-normande. La population de la ville est de habitants. L'agglomération de Galway est la quatrième du pays par le nombre d'habitants, après celles de Dublin, de Cork et de Limerick.

Property	Value
dbpedia-owl:abstract	▪ Galway és una ciutat d'Irlanda, capital del comtat de Galway, a la p és una institució local de Galway. Antigament es digué Galway Cor el desembre del 1484 pel rei Ricard III d'Anglaterra. El primer alcald Galway, però revifat el 1937 com a municipi i el 1985 com a municip ▪ Galway (irsky Gaillimh, nebo také An Gaillimh) je město v západoirs rozkládá na západním pobřeží Irska. V irštině je označováno také ja rostoucí irské město. Počet obyvatel v roce 2006 činil rovných 70 0(▪ Galway ist die Hauptstadt der Grafschaft Galway in der Provinz Con ▪ Galway or City of Galway (Cathair na Gaillimhe) is a city on the wes Galway Bay and is surrounded by County Galway. It is the third larg into fourth place behind Limerick. The population of Galway city at t ▪ Galway es la capital del condado de Galway, en Irlanda. La ciudad :

Firefox

Temperature forecast for Galwa...

5stardata.info/gtd-5.html

LarKC weblog » Blog A... Linking Open Da

Tempe
for Ga

Day	Lowest Temperature (°C)
Saturday, 13 November 2010	2
Sunday, 14 November 2010	4
Monday, 15 November 2010	7

Last update: 2012-01-22 by Michael | Code available via GitHub

zotero

Michael Hausenblas http://5stardata.info/

Costs & benefits of ★★★★★ Web data

- As a consumer, you can do all what you can do with ★★★★ Web data and additionally:
  - ✓ You can discover more (related) data while consuming the data.
  - ✓ You can directly learn about the data schema.
  - ⚠ You now have to deal with broken data links, just like 404 errors in web pages.
  - ⚠ Presenting data from an arbitrary link as fact is as risky as letting people include content from any website in your pages. Caution, trust and common sense are all still necessary.

Michael Hausenblas http://5stardata.info/

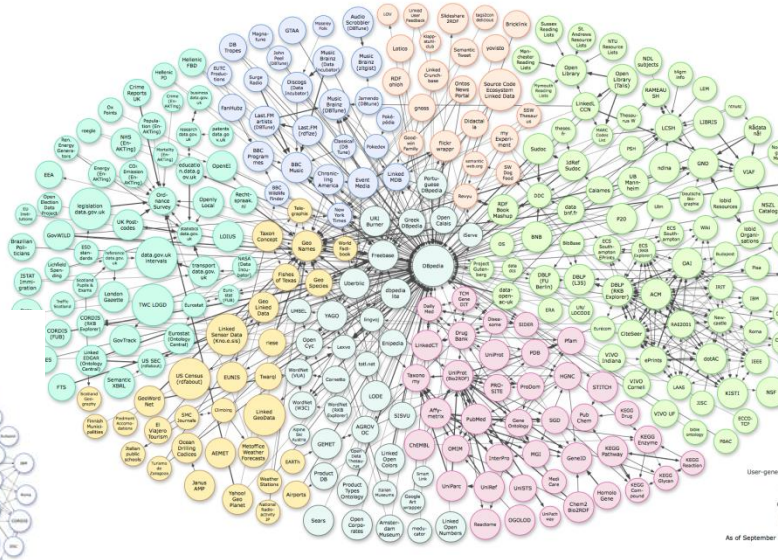
Costs & benefits of ★★★★★ Web data

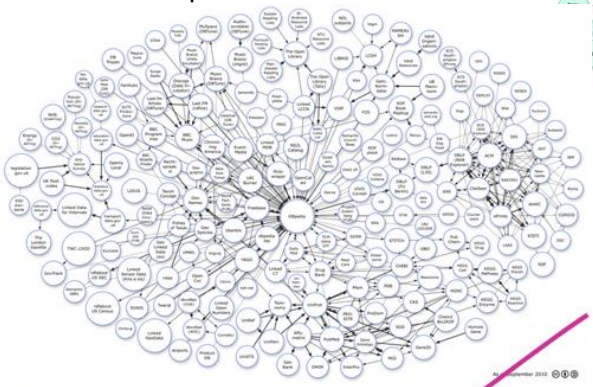
- As a publisher ...
  - ✓ You make your data discoverable.
  - ✓ You increase the value of your data.
  - ✓ Your own organisation will gain the same benefits from the links as the consumers.
  - ⚠ You'll need to invest resources to link your data to other data on the Web.
  - ⚠ You may need to repair broken or incorrect links.

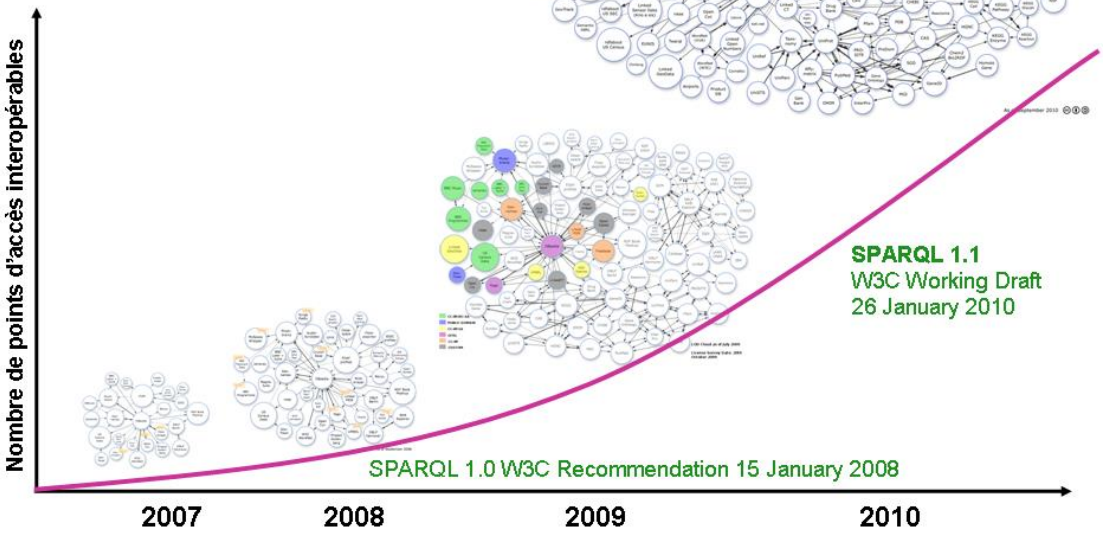
Michael Hausenblas http://5stardata.info/

Linked Open Data Cloud

- Linking open data project
  - goals:
    - Use RDF to “expose” open data sets
    - Create RDF links between these datasets
    - If possible, deploy SPARQL endpoints







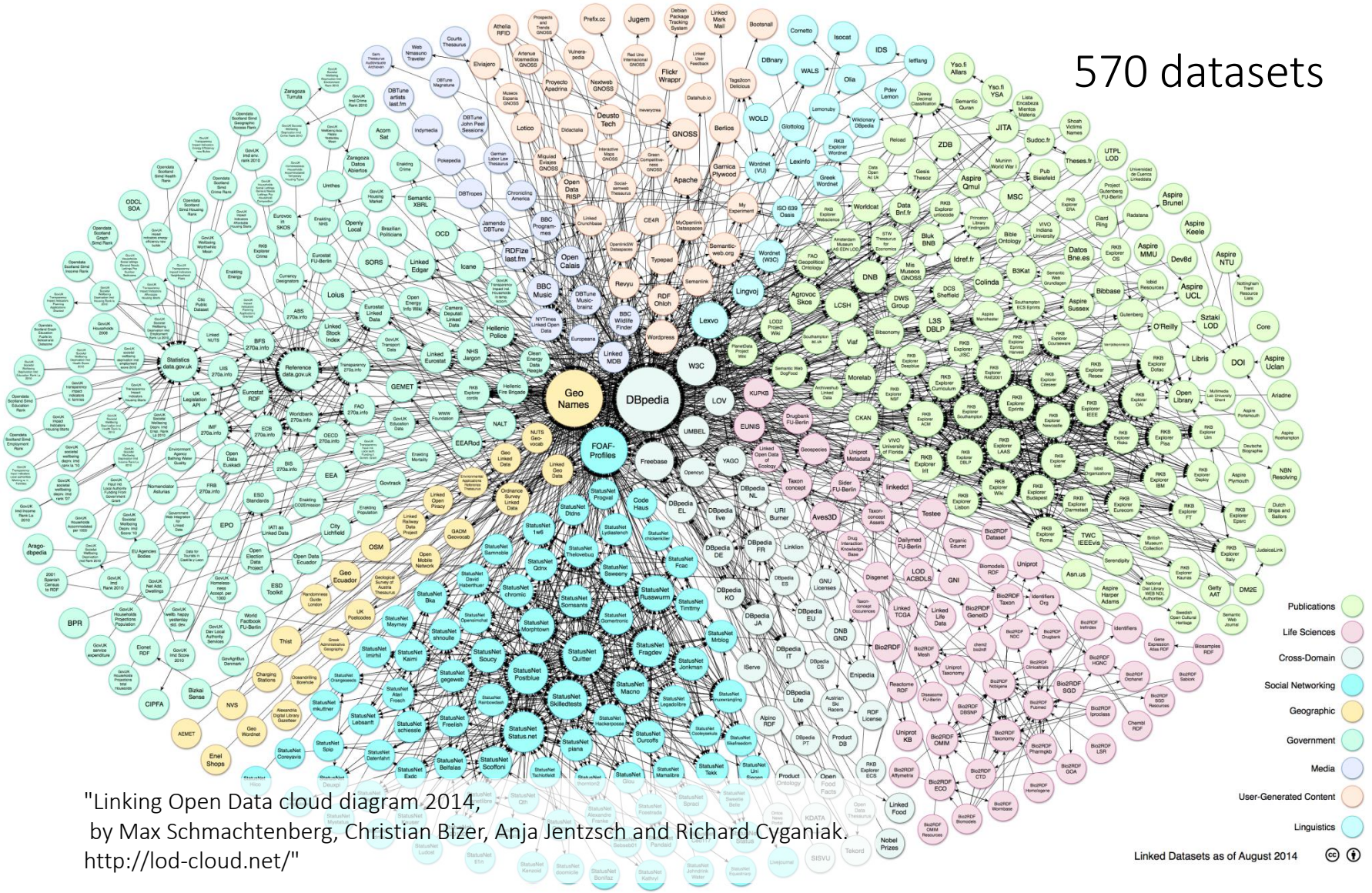
September 2011

295 datasets
31 Billion RDF triples
interconnected by
595 Million of RDF relations

`http://lod-cloud.net`

Linked Open Data Cloud

570 datasets

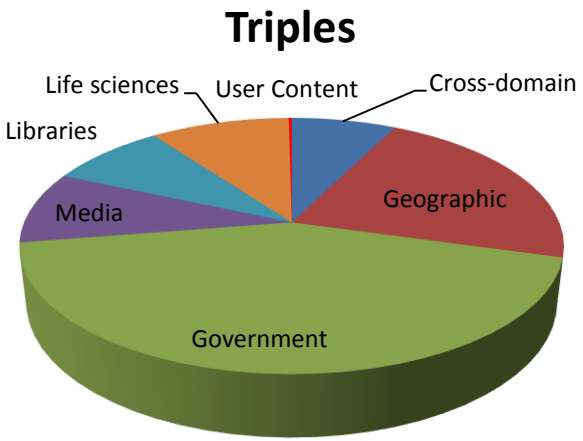


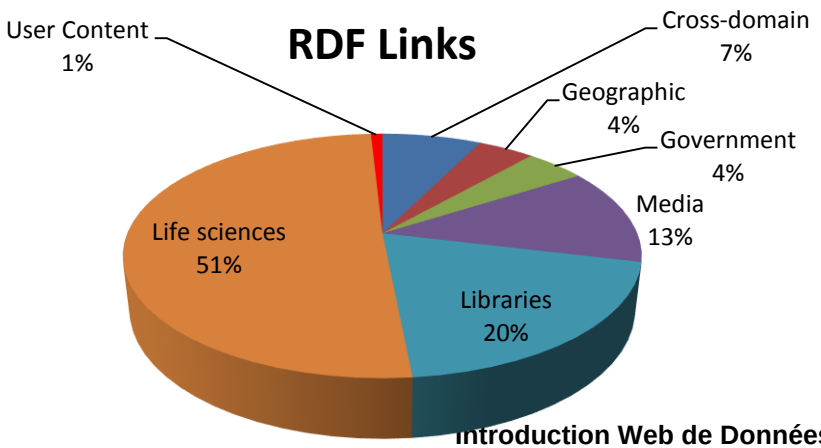
"Linking Open Data cloud diagram 2014,
by Max Schmachtenberg, Christian Bizer, Anja Jentzsch and Richard Cyganiak.
http://lod-cloud.net/"

Linked Open Data Cloud

Domain	Data Sets	Triples	Percent	RDF Links	Percent
Cross-domain	20	1,999,085,950	7.42	29,105,638	7.36
Geographic	16	5,904,980,833	21.93	16,589,086	4.19
Government	25	11,613,525,437	43.12	17,658,869	4.46
Media	26	2,453,898,811	9.11	50,374,304	12.74
Libraries	67	2,237,435,732	8.31	77,951,898	19.71
Life sciences	42	2,664,119,184	9.89	200,417,873	50.67
User Content	7	57,463,756	0.21	3,402,228	0.86
(2011 September)	203	26,930,509,703		395,499,896	

http://lod-cloud.net/state





Linked Open Data Cloud

Domain	Data Sets	Triples
Cross-domain	20	1,999,085
Geographic	16	5,904,980
Government	25	11,613,525
Media	26	2,453,898
Libraries	67	2,237,435
Life sciences	42	2,664,119
User Content	7	57,463
	203	26,930,509

(2011, September)

Datasets by topical domain.

Topic	Datasets	%
Government	183	18.05%
Publications	96	9.47%
Life sciences	83	8.19%
User-generated content	48	4.73%
Cross-domain	41	4.04%
Media	22	2.17%
Geographic	21	2.07%
Social web	520	51.28%
Total	1014	

(2014, August)

State of the LOD Cloud 2014

Version 0.4, 08/30/2014

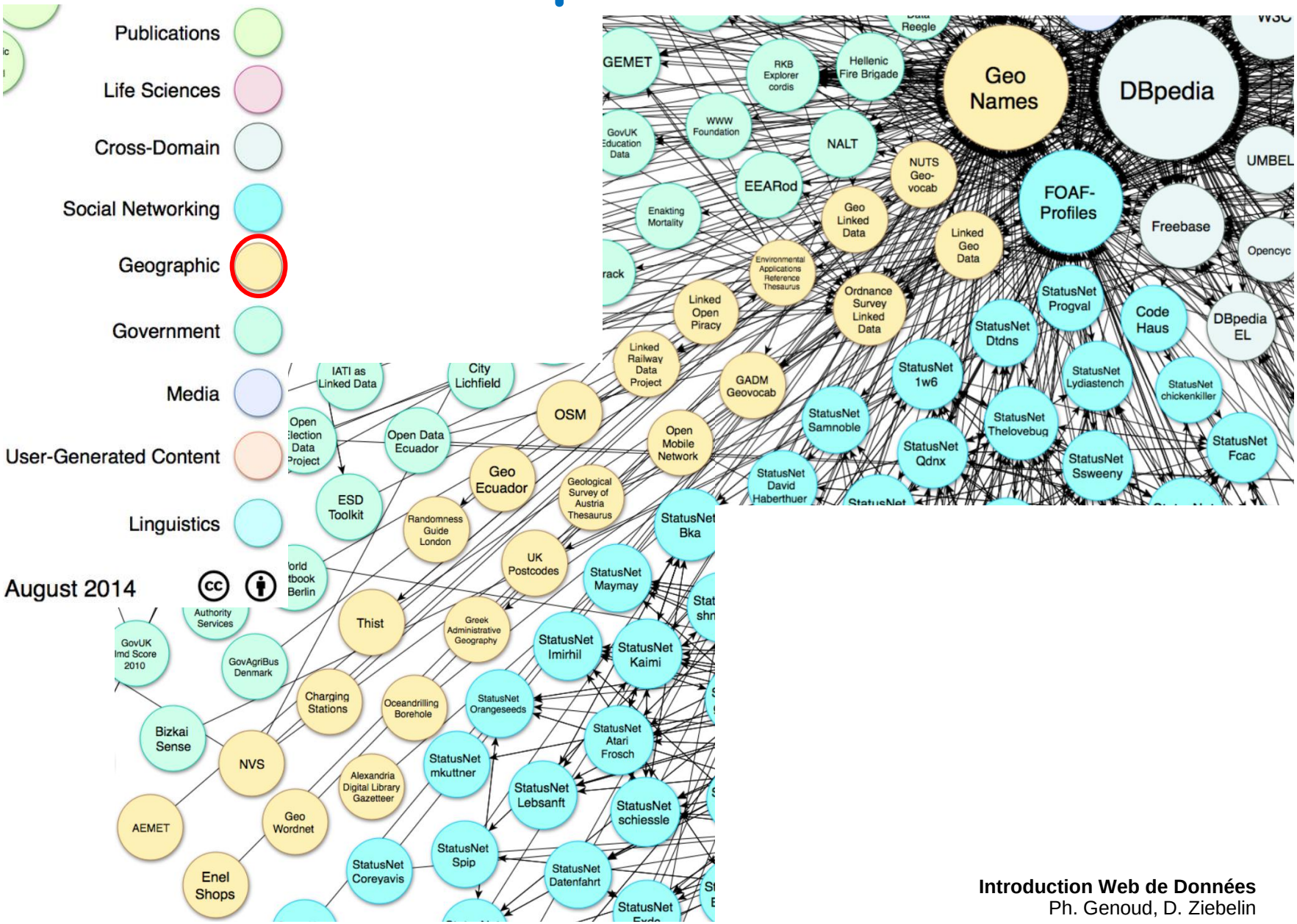
This document provides statistics about the structure and content of the crawlab
Linked Data sources implement the Linked Data best practices.

This document updates the findings of the original State of the LOD Cloud re
publishers themselves via the datahub.io Linked Data catalog. This report is base
the ISWC2014 paper Adoption of the Linked Data Best Practices in Different To
document links the statistics to the Mannheim Linked Data catalog and enables th

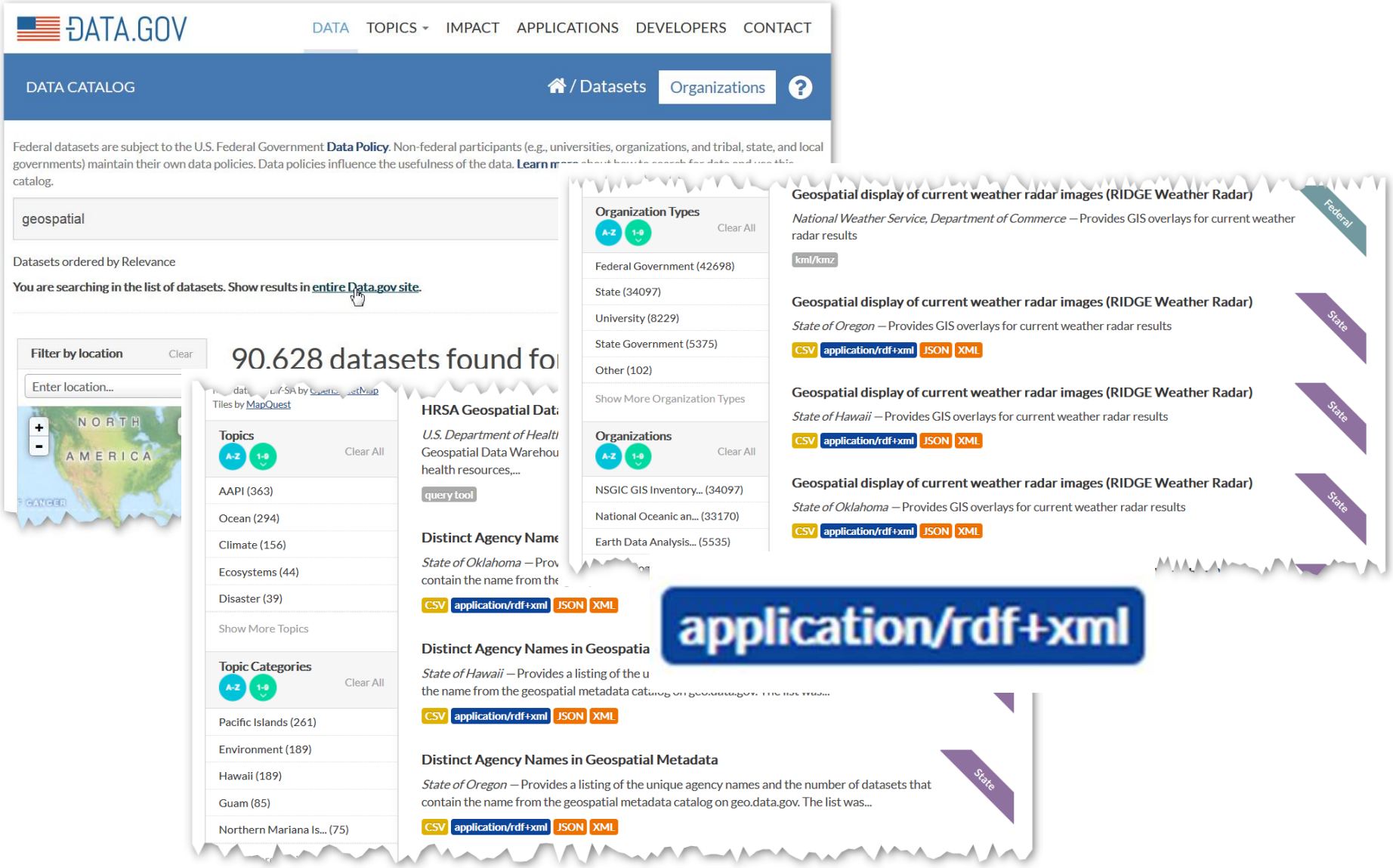
Contents

http://linkeddatacatalog.dws.informatik.uni-mannheim.de/state/

Linked Open Data Cloud



data.gov


DATA.GOV
DATA
TOPICS
IMPACT
APPLICATIONS
DEVELOPERS
CONTACT
DATA CATALOG
/ Datasets
Organizations
Federal datasets are subject to the U.S. Federal Government Data Policy. Non-federal participants (e.g., universities, organizations, and tribal, state, and local governments) maintain their own data policies. Data policies influence the usefulness of the data.
catalog.
geospatial
Datasets ordered by Relevance
You are searching in the list of datasets. Show results in entire Data.gov site.
Filter by location
Clear
Enter location...
NORTH AMERICA
90,628 datasets found fo
Tiles by MapQuest
Topics
Clear All
AAPI (363)
Ocean (294)
Climate (156)
Ecosystems (44)
Disaster (39)
Show More Topics
Topic Categories
Clear All
Pacific Islands (261)
Environment (189)
Hawaii (189)
Guam (85)
Northern Mariana Is... (75)
HRSA Geospatial Dat
U.S. Department of Healt
Geospatial Data Warehou
health resources,...
query tool
Distinct Agency Name
State of Oklahoma — Prov
contain the name from the
CSV application/rdf+xml JSON XML
Distinct Agency Names in Geospatia
State of Hawaii — Provides a listing of the u
the name from the geospatial metadata cata
CSV application/rdf+xml JSON XML
Distinct Agency Names in Geospatial Metadata
State of Oregon — Provides a listing of the unique agency names and the number of datasets that contain the name from the geospatial metadata catalog on geo.data.gov. The list was...
CSV application/rdf+xml JSON XML
State
Organization Types
Clear All
Federal Government (42698)
State (34097)
University (8229)
State Government (5375)
Other (102)
Show More Organization Types
Organizations
Clear All
NSGIC GIS Inventory... (34097)
National Oceanic an... (33170)
Earth Data Analysis... (5535)
Geospatial display of current weather radar images (RIDGE Weather Radar)
National Weather Service, Department of Commerce — Provides GIS overlays for current weather radar results
kml/kmz
Federal
Geospatial display of current weather radar images (RIDGE Weather Radar)
State of Oregon — Provides GIS overlays for current weather radar results
CSV application/rdf+xml JSON XML
State
Geospatial display of current weather radar images (RIDGE Weather Radar)
State of Hawaii — Provides GIS overlays for current weather radar results
CSV application/rdf+xml JSON XML
State
Geospatial display of current weather radar images (RIDGE Weather Radar)
State of Oklahoma — Provides GIS overlays for current weather radar results
CSV application/rdf+xml JSON XML
State
application/rdf+xml

http://www.ordnancesurvey.co.uk/

http://www.ordnancesurvey.co.uk/blog/2013/06/new-linked-data-service-launches/

Ordnance Survey

About this blog Blog

Home / Using GI and maps / New Linked Data service launches

New Linked Data service launches

JUN 03 2013

By Gemma In Using GI and maps

Comments (5)

Gemma works in our Corporate Communications team as our Social Media Manager

Author RSS

We are delighted to launch the next iteration of our Linked Data service today at: http://data.ordnancesurvey.co.uk.

In preparation for this launch we created a beta version, which was designed for you to have a play around, test, and review against your current applications.

We had over 2,000 people test the beta version, thank you for your helpful feedback.

We launched Linked Data in April 2010 as part of the drive to increase innovatic Public Data Public" initiative and have seen a cont government and research. This has allowed us to c

http://data.ordnancesurvey.co.uk/

Ordnance Survey

You are here: linked-data

Ordnance Survey Linked Data Platform

Quick Search: Enter a place name or postcode to view as Linked Data

Ordnance Survey is Great Britain's national mapping agency, providing the most accurate and up-to-date geographic data, relied on by government, business and individuals.

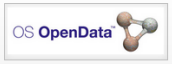
OS OpenData is the opening up of Ordnance Survey data as part of the drive to increase innovation and support the "Making Public Data Public" initiative. As part of this initiative Ordnance Survey has published a number of its products as Linked Data. Linked Data is a growing part of the Web where data is published on the Web and then linked to other published data in much the same way that web pages are interlinked using hypertext.

The term Linked Data is used to describe a method of exposing, sharing, and connecting data via URIs on the Web. To find more Linked Data published as part of this initiative please go to data.gov.uk.

If you are not familiar with Linked Data, OS OpenData products are also available in alternative formats from the OS OpenData website. Ordnance Survey can provide support for the Ordnance Survey OpenData products, but cannot give advice or support on using RDF, SPARQL or SPARQL Endpoints.

Ordnance Survey has published three OS Open Data products as Linked Data: the 1:50 000 Scale Gazetteer, Code-Point Open and the administrative geography for Great Britain taken from Boundary Line. A combined OS Linked Data dataset combines these products into one database to support more flexible data access.

Current Datasets



Ordnance Survey Linked Data
36,773,687 triples

Code-Point Open Linked Data
33,750,456 triples

50K Gazetteer Linked Data
2,368,655 triples

Boundary Line Linked Data
653,433 triples

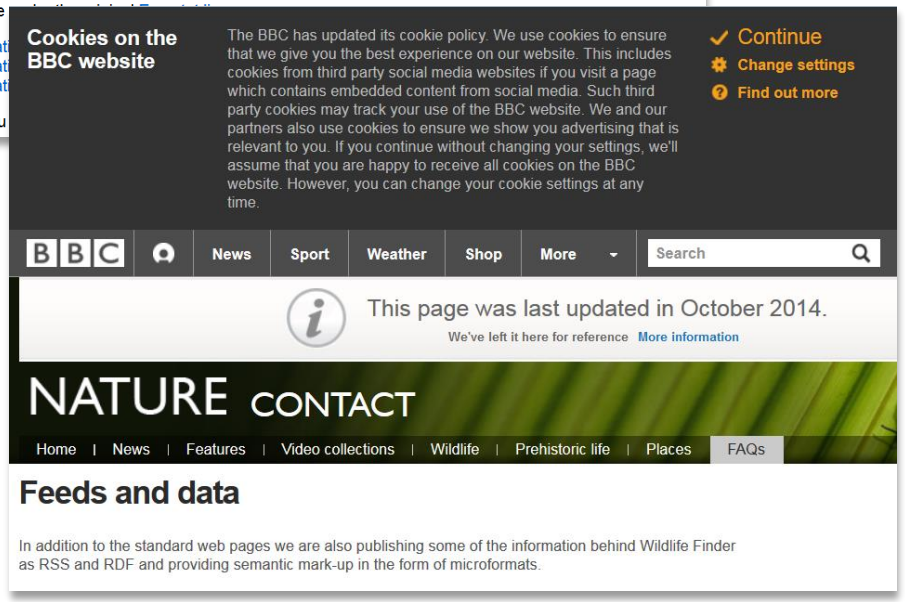
data.gouv.fr

encore un effort !!!

http://www.insee.fr/fr/methodes/default.asp?page=xml/xml.htm

http://eurostat.linked-statistics.org/

http://www.bbc.co.uk/nature/feedsanddata



. . .

Outline

- "Theoretical" Session (morning)
  - Introduction
  - Distributing Data on the web with RDF
    - Naming the Data : URIs (Uniform Resources Identifiers)
    - The RDF Data model
  - Querying Linked Data with SPARQL
  - Semantic modelling
    - RDFS
    - OWL
  - From Open Data to Linked Open Data
  - Conclusion
- Hands-on session (afternoon)
  - From a CSV file to linked data
  - Querying linked data (SPARQL)

The importance of Linked Data

Ivan Herman http://www.w3.org/2009/Talks/0615-SanJose-tutorial-IH/

- It provides a core set of data that Semantic Web applications can build on
  - stable references for “things”,
    - e.g., http://dbpedia.org/resource/Grenoble
  - many many relationships that applications may reuse
    - e.g., the BBC application!
  - a “nucleus” for a larger, semantically enabled Web!
- For many, publishing data may be the first step into the world of Semantic Web

Some things to remember if you publish data

Ivan Herman http://www.w3.org/2009/Talks/0615-SanJose-tutorial-IH/

- Publish your data first, care about sexy user interfaces later!
  - the "raw data" can become useful on its own right and others may use it
  - you can add your added value later by providing nice user access
- If possible, publish your data in RDF but if you cannot, others may help you in conversions
  - trust the community…
- *Add links to other data*. "Just" publishing isn't enough…

Open challenges for LOD

- Contradictions:

  Population of Grenoble
  - DBpedia: **156,659**
  - Freebase: **155,632**

- **Identity crisis**: is the capital of the Roman Empire the *sameAs* the capital of modern Italy?

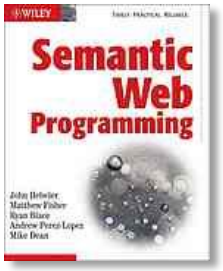
- Data ownership, copyright, access control

StratAG sfi

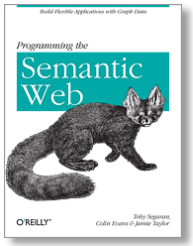
LOD challenges

- Ontological modeling :
  - design patterns
- Reasoning
- Large-scale processing of Linked Data
- Data interlinking
  - Ontologies alignments
- Quality of links and the data
- …

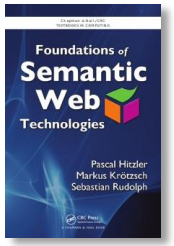
Bibliography



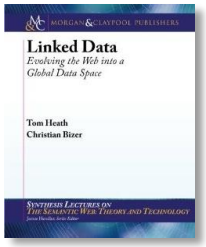
- John Hebeler, Matthew Fisher, Ryan Blace, Andrew Perez-Lopez, Mike Dean, *Semantic Web Programming*, Wiley, (2009)



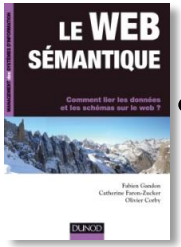
- Toby Segaran, Jamie Taylor, and Colin Evans, *Programming the Semantic Web*, O'Reilly, (2009)



- P. Hitzler, R. Sebastian, and M. Krötzsch: Foundation of Semantic Web Technologies, (2009)



- T. Heath and C. Bizer: *Linked Data: Evolving the Web into a Global Data Space*, (2011)



- F. Gandon, C. Faron-Zucker, O. corby : *Le Web Sémantique* (Dunod 2012)
- ...

W3C wiki page to find more references : http://www.w3.org/2001/sw/wiki/Books

Many tools (not an exhaustive list!)

- Few names:
  - Jena, AllegroGraph, Mulgara, Sesame, flickurl, 4Store, …
  - TopBraid Suite, Virtuoso environment, Falcon, Drupal 7, Redland, Pellet, HermiT …
  - Disco, Oracle 11g, RacerPro, IODT, Ontobroker, OWLIM, Talis Platform, …
  - RDF Gateway, RDFLib, Open Anzo, DartGrid, Zitgist, Ontotext, Protégé, …
  - Thetus publisher, SemanticWorks, SWI-Prolog, RDFStore…
  - …
- Categories:
  - Triple Stores
  - Inference engines
  - Converters
  - Search engines
  - Middleware
  - CMS
  - Semantic Web browsers
  - Development environments
  - Semantic Wikis
  - …

more on http://www.w3.org/2001/sw/wiki/Tools

Merci de votre attention