eBISS summer school 2014 in Berlin

From the Web of Data
To the Linked Data

The Web of Data, understanding the technological keys and publishing Linked Data

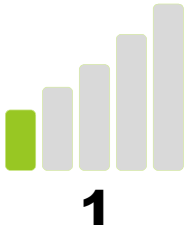
Gabriel KEPEKLIAN
R&D director
Atos Integration France

gabriel.kepeklian@atos.net

Atos

- The Data, open or not, always raise the problems of heterogeneity: many data formats, many metadata schema descriptions, complexity to mix... Using the Web of Data technologies, these problems may be overcome. Semantic Web technologies allow to move from raw data to linked data after elevation and interlinking phases. However, lifting raw Data to Linked Data is far from being straightforward. We will see the whole process and the results of a concrete implementation.

1. The **basic principles** of the Web of Data: from raw and heterogeneous Data to the lingua franca of the Web of Data
2. The **RDF** semantic and its main syntaxes
3. **Handling and querying** Data (triple store, end point, SPARQL)
4. Understanding the basics of **knowledge modeling** (ontology, OWL)
5. The **lifting and linking** of Data in order to publish 5 Stars Data (ex. DataLift plateform).



The basic principles of the Web of Data

from raw and heterogeneous Data to the lingua franca of the Web of Data

1. The **basic principles** of the Web of Data: from raw and heterogeneous Data to the lingua franca of the Web of Data
2. The **RDF** semantic and its main syntaxes
3. **Handling and querying** Data (triple store, end point, SPARQL)
4. Understanding the basics of **knowledge modeling** (ontology, OWL)
5. The **lifting and linking** of Data in order to publish 5 stars Data (DataLift).

Data

Datasets

Sources of data

Data stores

A data, alone,
has no value…

From data to information
From information to knowledge

$\wedge$	0	1
0	0	0
1	0	1

$\vee$	0	1
0	0	1
1	1	1

$\rightarrow$	0	1
0	1	1
1	0	1

$\oplus$	0	1
0	0	1
1	1	0

Figure 1. Truth tables

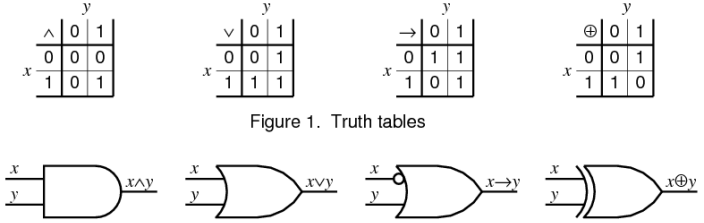


Figure 2. Logic gates

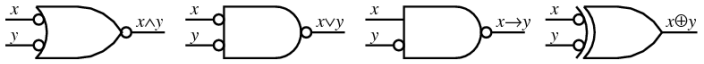
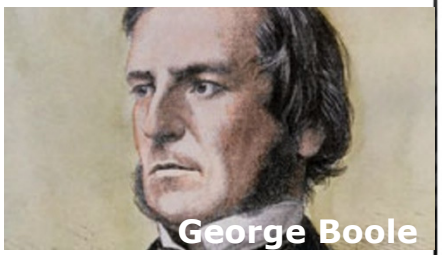


Figure 3. De Morgan equivalents

Figure 4. Venn diagrams



***54·43.** $\vdash :. \alpha, \beta \in 1 . \supset : \alpha \cap \beta = \Lambda . \equiv . \alpha \cup \beta \in 2$

Dem.

$\vdash . *54·26 . \supset \vdash :. \alpha = \iota' x . \beta = \iota' y . \supset : \alpha \cup \beta \in 2 . \equiv . x \neq y .$

[*51·231] $\equiv . \iota' x \cap \iota' y = \Lambda .$

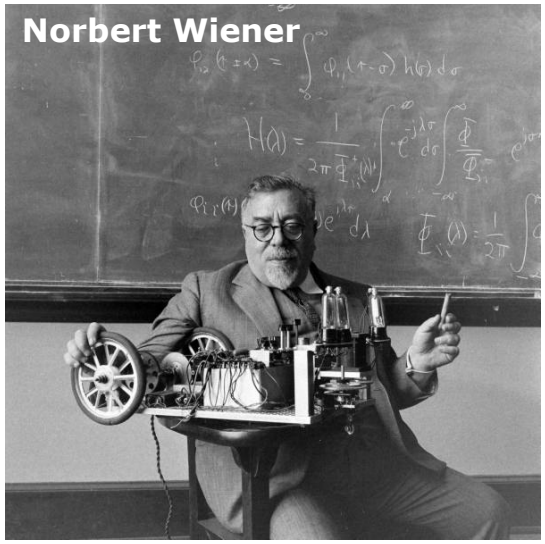
[*13·12] $\equiv . \alpha \cap \beta = \Lambda$ (1)

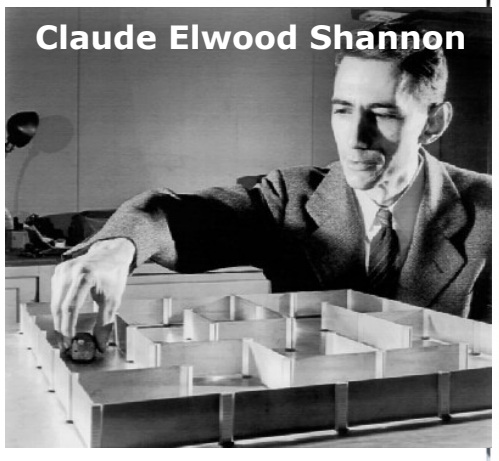
$\vdash . (1) . *11·11·35 . \supset$

$\vdash :. (\exists x, y) . \alpha = \iota' x . \beta = \iota' y . \supset : \alpha \cup \beta \in 2 . \equiv . \alpha \cap \beta = \Lambda$ (2)

$\vdash . (2) . *11·54 . *52·1 . \supset \vdash .$ Prop

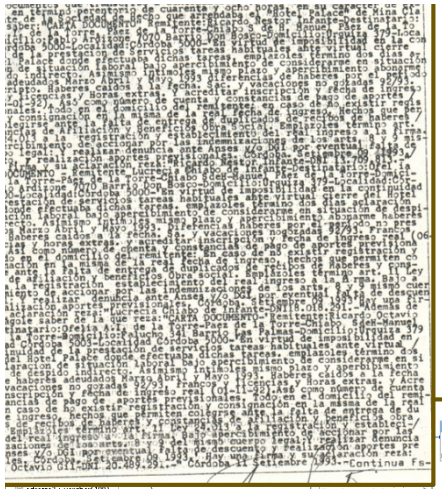
From this proposition it will follow, when arithmetical addition has been defined, that $1 + 1 = 2$.

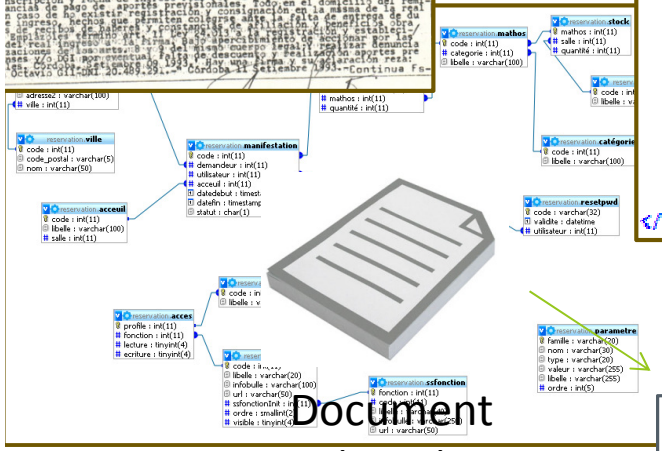





WEB 3.0
2007, 2008, ...

The Data

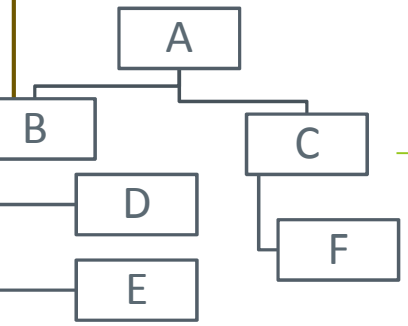




Document
(CMS)

```
<Books>
    <Book ISBN="0553212419">
        <title>Sherlock Holmes: Complete Novels...
        <author>Sir Arthur Conan Doyle</author>
    </Book>
    <Book ISBN="0743273567">
        <title>The Great Gatsby</title>
        <author>F. Scott Fitzgerald</author>
    </Book>
    <Book ISBN="0684826976">
        <title>Undaunted Courage</title>
        <author>Stephen E. Ambrose</author>
    </Book>
    <Book ISBN="0743203178">
        <title>Nothing Like It In the World</title>
        <author>Stephen E. Ambrose</author>
    </Book>
</Books>
```

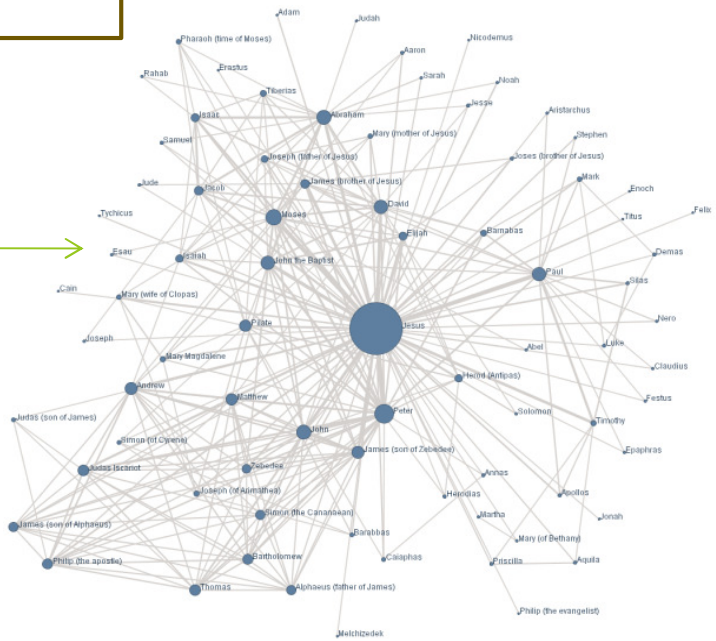
	animal	hair	feathers	eggs	milk	airborne	aquatic	predator	toothed
1	String	String	String	String	String	String	String	String	String
2	aardvark	true	false	false	true	false	false	true	true
3	antelope	true	false	false	true	false	false	false	true
4	bass	false	false	true	false	false	true	true	true
		true	false	false	true	false	false	true	true
		true	false	false	true	false	false	true	true
		true	false	false	true	false	false	false	true
		true	false	false	true	false	false	false	true
		false	false	true	false	false	true	false	true
		false	false	true	false	false	true	true	true
		true	false	false	true	false	false	false	true
		true	false	false	true	false	false	true	true
		false	true	true	false	true	false	false	false



Hierarchy
(XML)

	x	y	z
a	1	2	4
b	0	9	6
c	9	0	7

Data base
(SQL)



Graph
(RDF)

- Simple information
  - The name of Voltaire
  - The number of legs of a spider
  - The fact that the door is closed
- A grouping of data, document
  - A biography, a text, a newspaper
  - A table of values
  - A web page
- A collection of data
  - A list of composers
  - A set of test scores
  - The names of the months of the year

Do we talk about data without giving them meaning?

The elementary data

Many limitations

- Research possibilities (lexical)
- The presentation is summary
- The freshness of data
- The reliability of data
- The comparison of available sources
- The evaluation of sources
- Etc.

There is always much more data than we think

Temps et récit, tome 1 [Poche]

Paul Ricoeur (Auteur)

(2 commentaires client) J'aime (0)

Prix conseillé : ~~EUR 8,50~~

Prix : EUR 8,07 LIVRAISON GRATUITE En savoir plus.

Économisez : EUR 0,43 (5%)

En stock.

Expédié et vendu par **Amazon.fr**. Emballage cadeau disponible.

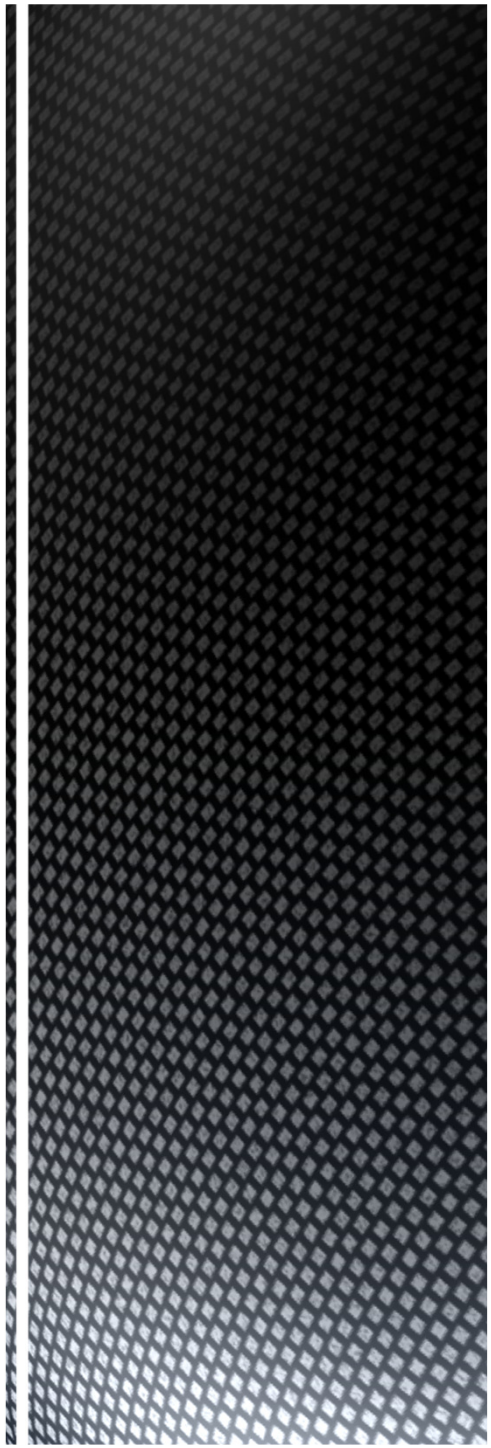
Plus que 8 ex (réapprovisionnement en cours). Commandez vite !

Voulez-vous le faire livrer le mercredi 22 juin ?
Commandez-le dans les 14 h et 53 min et choisissez la **livraison Éclair** sur votre bon de commande. En savoir plus.

9 neufs à partir de EUR 8,07 **5 d'occasion** à partir de EUR 2,90

- The book is titled "Temps et récit"
- The book was writen by Paul Ricoeur
- The discount is 5%
- ETC.

Understanding,
knowledge?

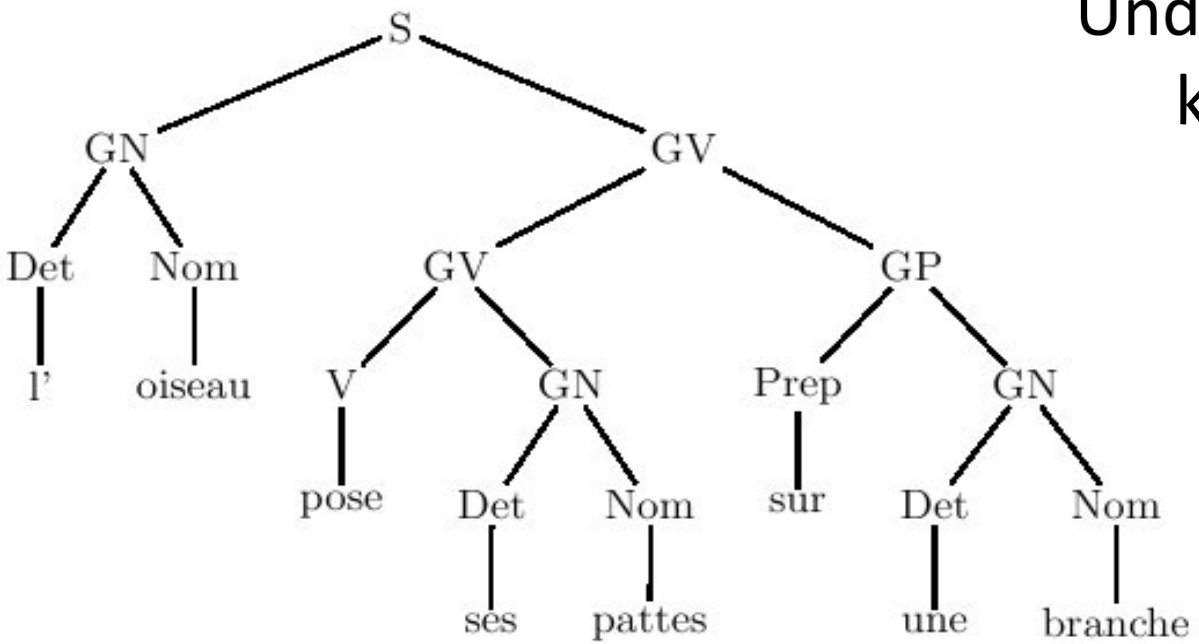


Structuring data

Representing informations

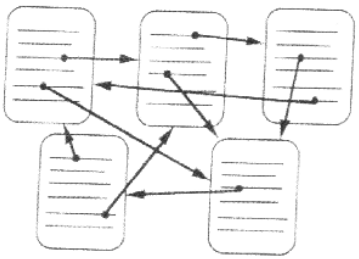
They are ways to gain in understanding and knowledge.

Understanding, knowledge?



WEB = A technological mean to make available, to link and to share **documents** on a network of connected **machines**.

The four building blocks of the Web

A protocole	A name (identification's mecanism)	A principle	A langague
			
HTTP	URL/URI	L'hypertexte	HTML

WEB = A technological mean to make available, to link and to share **ressources** on a network of connected **devices**.

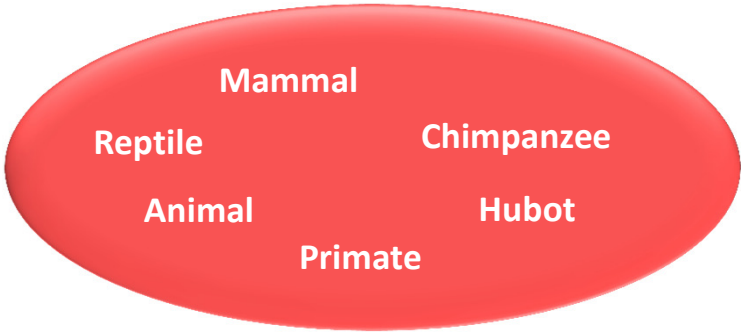
Describing the data

Classification schemes

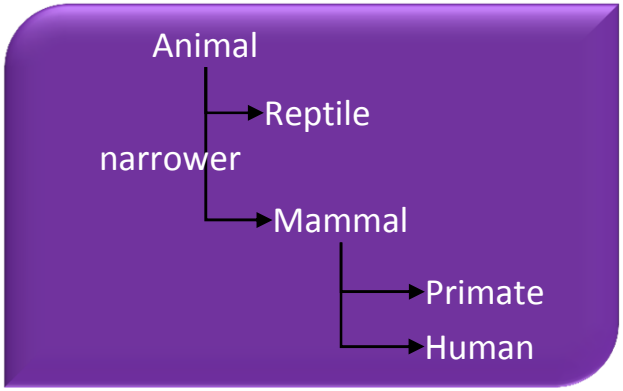
Formats to represent them

(Major) Knowledge Organization's Systems

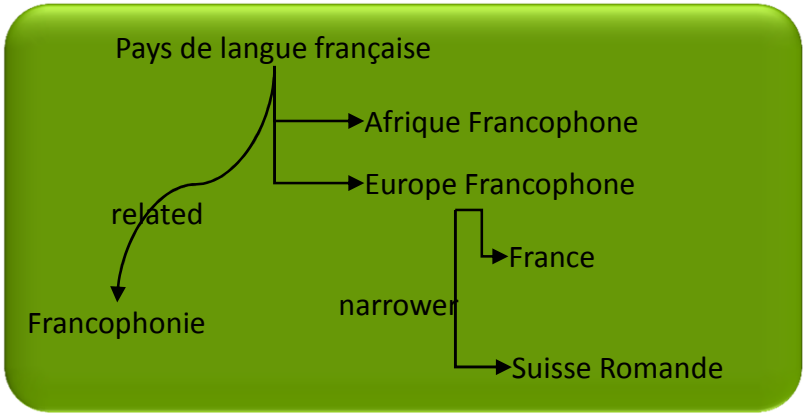
Controlled vocabulary



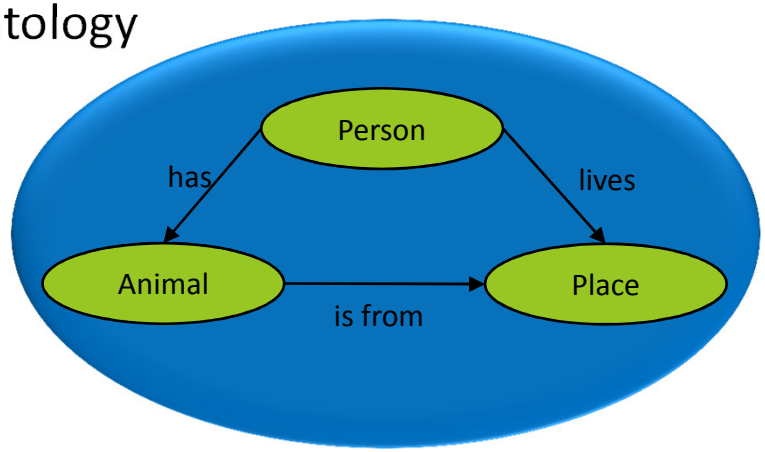
Taxonomy



Thesaurus



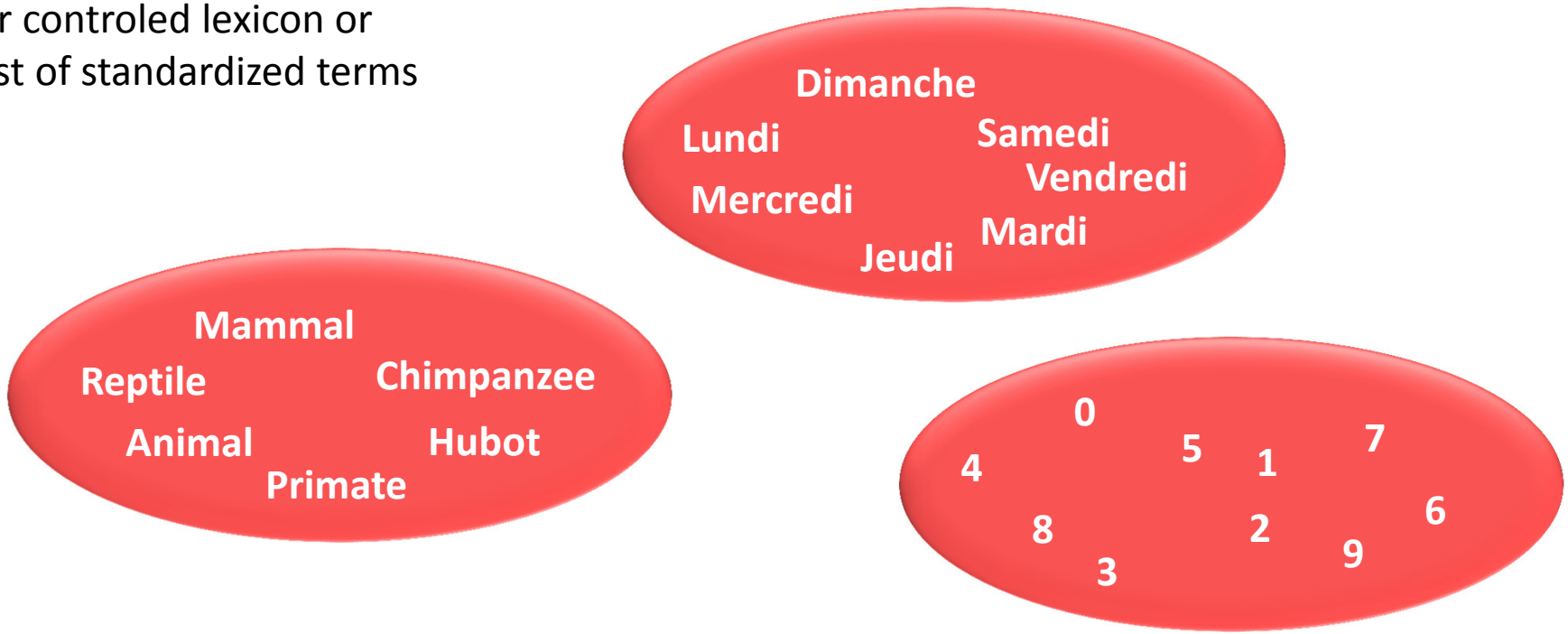
Ontology



Controlled vocabulary

Controlled vocabulary

or controled lexicon or
list of standardized terms



A controlled vocabulary is a set of recognized terms, fixed, unalterable, standardized and validated by a group (a community of practices) used to index and analyze contents and search for information in a defined area of information.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="http://purl.org/gem/instance/level/" targetNamespace="http://purl.org/gem/instance/level/"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
    <xs:annotation>
      <xs:documentation xml:lang="en">GEM controlled vocabulary for designating the education level of the
      audience for a resource. Created in 1996 by the GEM Consortium.</xs:documentation>
    </xs:annotation>
    <xs:simpleType name="Level">
      <xs:union>
        <xs:simpleType>
          <xs:restriction base="xs:Name">
            <xs:enumeration value="Preschool education">
            <xs:enumeration value="Kindergarten">
            <xs:enumeration value="Grade 1">
            <xs:enumeration value="Grade 2">
            <xs:enumeration value="Grade 11">
            <xs:enumeration value="Grade 12">
            <xs:enumeration value="Adult/continuing education">
            <xs:enumeration value="Higher education">
            <xs:enumeration value="Vocational education">
          </xs:restriction>
        </xs:simpleType>
      </xs:union>
    </xs:simpleType>
  </xs:schema>
```

Controlled vocabulary

an XML sample

GEM educationLevel Vocabulary (XML)

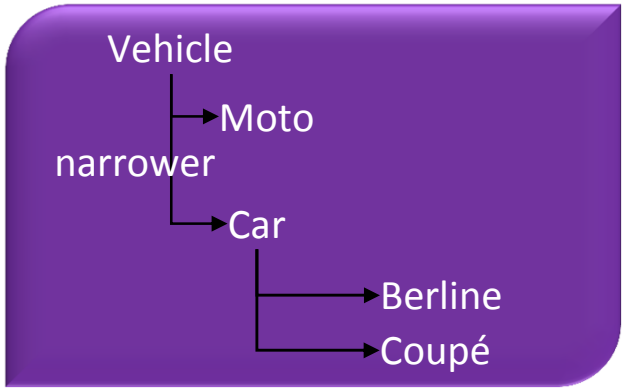
Controlled vocabulary

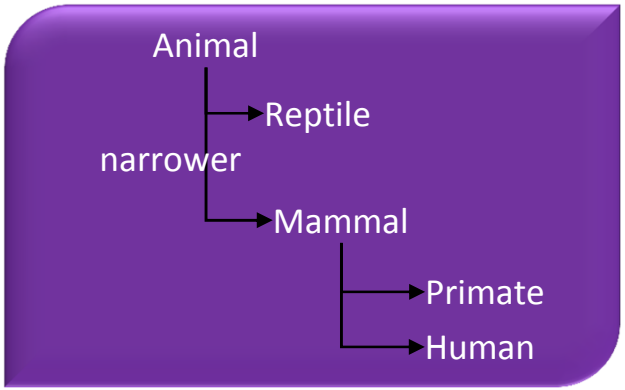
an JSON sample


```
{"countries":[
        {"country":{"country_id":"AD","country_name":"ANDORRA"}},
        {"country":{"country_id":"AE","country_name":"UNITED ARAB EMIRATES"}},
        {"country":{"country_id":"AF","country_name":"AFGHANISTAN"}},
        {"country":{"country_id":"AG","country_name":"ANTIGUA AND BARBUDA"}},
        {"country":{"country_id":"AI","country_name":"ANGUILLA"}},
        {"country":{"country_id":"AL","country_name":"ALBANIA"}},
        {"country":{"country_id":"AM","country_name":"ARMENIA"}},
        {"country":{"country_id":"AN","country_name":"NETHERLANDS ANTILLES"}},
        {"country":{"country_id":"AO","country_name":"ANGOLA"}},
        {"country":{"country_id":"AQ","country_name":"ANTARCTICA"}},
        {"country":{"country_id":"AR","country_name":"ARGENTINA"}},
        {"country":{"country_id":"AS","country_name":"AMERICAN SAMOA"}},
        {"country":{"country_id":"AT","country_name":"AUSTRIA"}},
Etc.
        {"country":{"country_id":"ZM","country_name":"ZAMBIA"}},
        {"country":{"country_id":"ZW","country_name":"ZIMBABWE"}}]}
```


countries.json

Taxonomy

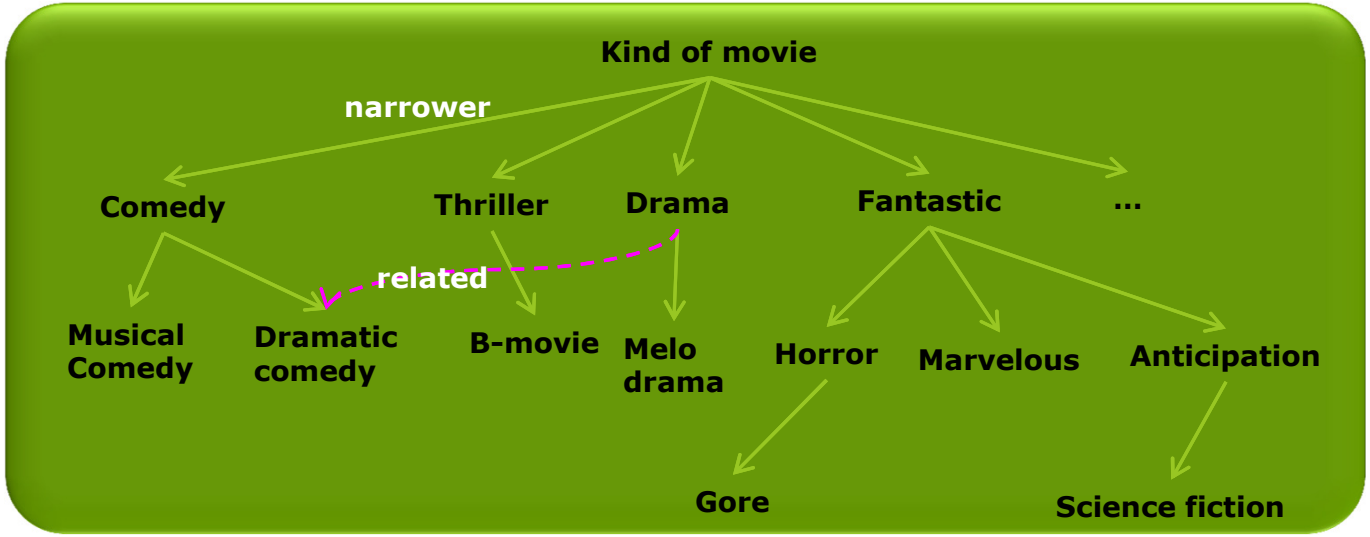


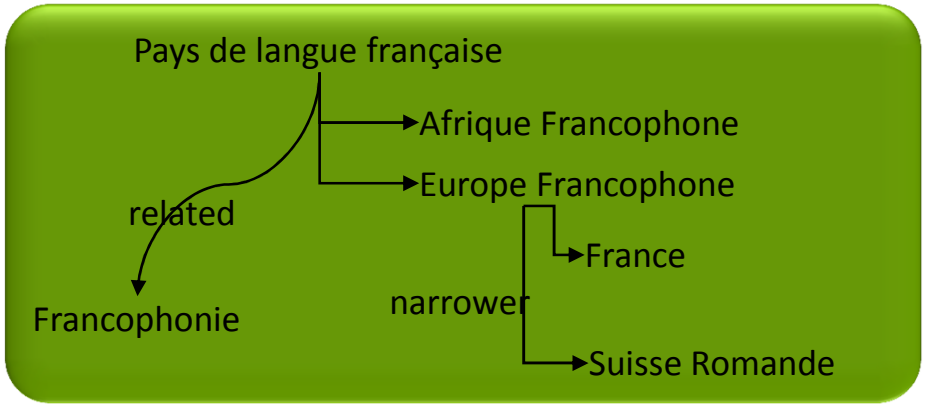


Taxonomy

A taxonomy is a classification method in a hierarchical structured and scalable architecture of information.

Thesaurus





Thesaurus

A thesaurus is like a taxonomy with more information about each concept including alternative and preferred terms. It may contain links to related concepts.

Source http://wiki.univ-paris5.fr/wiki

Thesaurus

- It is a set of standard terms based on a hierarchical structure.
- The terms are conceptually organized and interconnected by semantic relations.
- The thesaurus is alphabetically organized, it forms an alphabetical list of standardized terms for
  - content analysis,
  - classification
  - and indexing documents of information
- In many cases, the thesaurus also proposes a definition of the terms used.

Source : R. Troncy

Relations

- BT : Broader Term
- NT : Narrower Term
- TT : Top Term
- RT : Related Term (other than BT, NT, TT, etc.)

Thesaurus sample

United Nations Educational, Scientific and Cultural Organization

UNESCO Thesaurus

Hierarchical display

List of the seven major subject domains, broken down into microthesauri which allow you to gain a quick overview of the subject matter.

> Select a field | > Then select a microthesaurus | Search | Need help?

- > Select a field
- 1 Education
- 2 Science
- 3 Culture
- 4 Social and human sciences
- 5 Information and communicatio
- 6 Politics, law and economics
- 7 Countries and country groupin

ogue (UNESDOC) directly from the thesaurus.

scriptor to obtain the records indexed with that term in unesdoc/unesbib

Hierarchical display

List of the seven major subject domains, broken down into microthesauri which allow you to gain a quick overview of the subject matter.

2 Science | > Then select a microthesaurus | Search | Need help?

- > Then select a microthesaurus
- 2.05 Scientific approach
- 2.10 Science and research management
- 2.15 Mathematics and statistics
- 2.20 Physical sciences
- 2.25 Chemical sciences
- 2.30 Space sciences
- 2.35 Earth sciences
- 2.40 Geography and oceanography
- 2.45 Meteorology
- 2.50 Hydrology
- 2.55 Environmental sciences and engineering
- 2.60 Pollution, disasters and safety
- 2.65 Natural resources
- 2.70 Biology
- 2.75 Natural sciences
- 2.80 Medical sciences
- 2.85 Pathology

You can launch a query in the online catalogu

Click on the number next to the desired descriptor

verning Bodies | Director-General | Secretariat | us

UNESDOC@lerts | Information centres | Library

Terms of use | Contact us | s

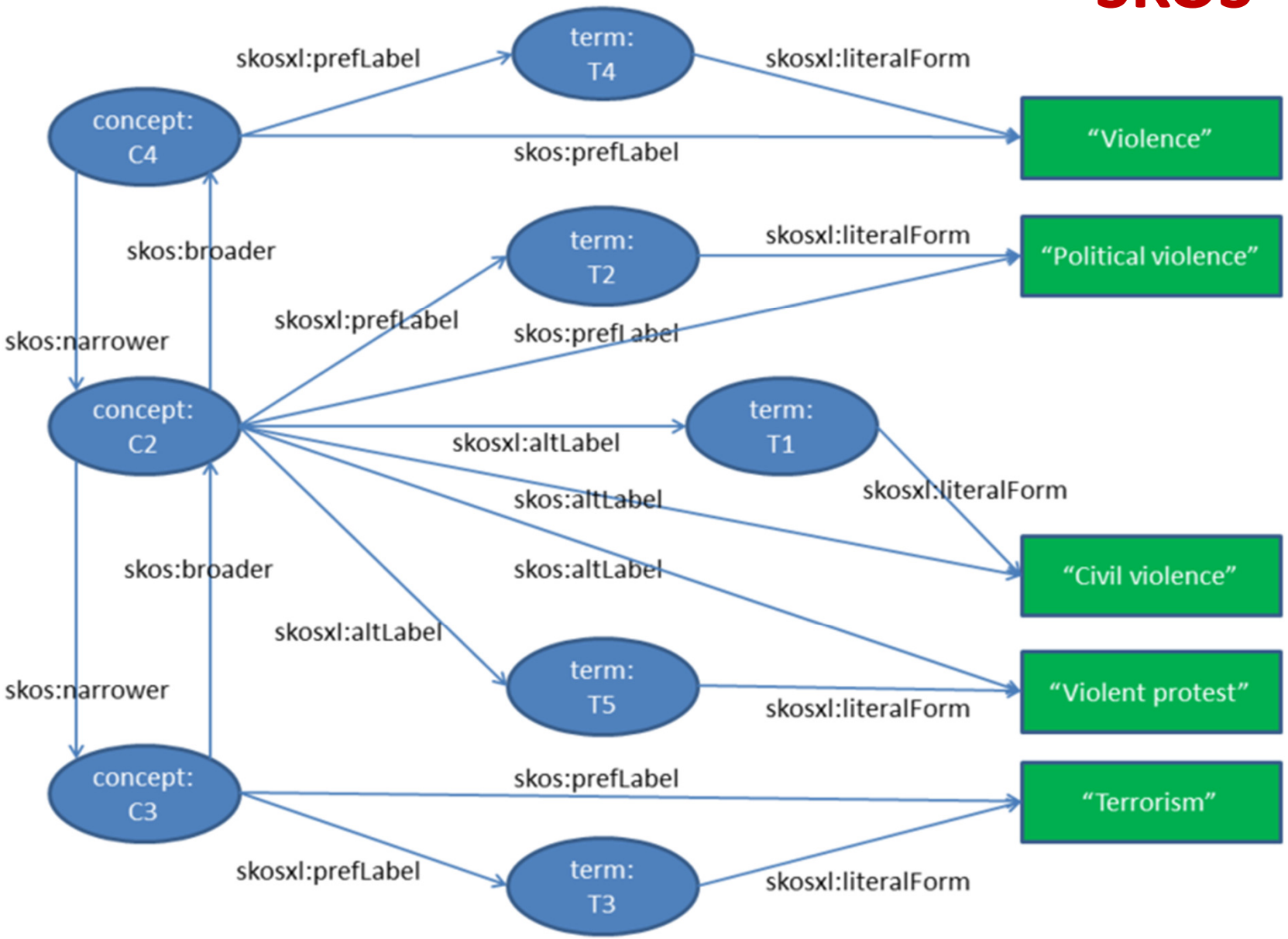
http://databases.unesco.org/thesfr/

Thesaurus sample

Une notice complète contient le microthésaurus (**MT**) auquel le terme appartient; la version anglaise, espagnole et russe; une note explicative le cas échéant (**NE**); les relations entre les termes (**EMP:EP; TG:TS; TA**)

Simple Knowledge Organization System - W3C 2009

SKOS



SKOS

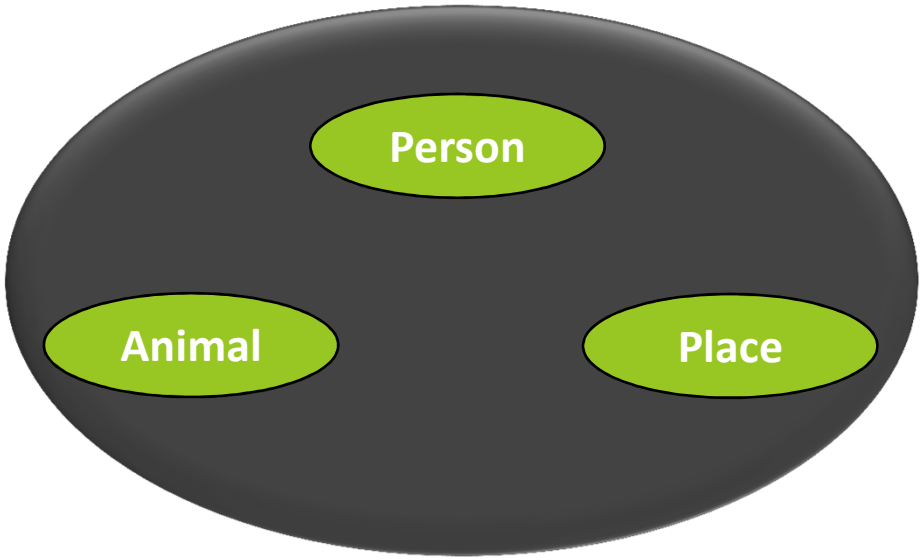
```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix skosxl: <http://www.w3.org/2008/05/skos-xl#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix con: <http://example.org/id/concept/polthes/> .
@prefix term: <http://example.org/id/term/polthes/> .

term:T1 a skosxl:Label;
       skosxl:literalForm "Civil violence"@en.

con:C1 a skos:Concept;
      skos:prefLabel "Civil violence"@en;
      skosxl:prefLabel term:T1;
      skos:altLabel "Political violence"@en;
      skosxl:altLabel term:T2;
      skos:altLabel "Violent protest"@en;
      skosxl:altLabel term:T5;
      skos:broader con:C4;
      skos:narrower con:C3;
      dcterms:replaces con:C2 .
(…)

term:T3 a skosxl:Label;
       skosxl:literalForm "Terrorism"@en.

con:C4 a skos:Concept;
      skos:prefLabel "Violence"@en;
      skosxl:prefLabel term:C4;
      skos:narrower con:C1 .

(…)
```

Ontology

Ontology

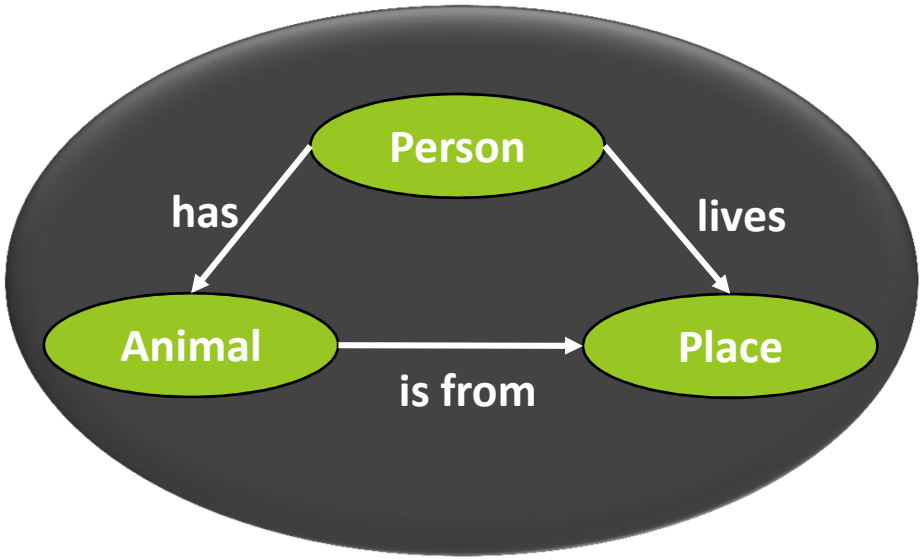
An ontology is a model of a domain of knowledge, describing the **concepts** within that domain

Ontologies are often also called vocabularies.



Ontology

An ontology is a model of a domain of knowledge, describing the **concepts** within that domain and the **relations** between concepts in the domain.



Ontologies are often also called vocabularies.

Studer et al (1998) distinguish foor types of ontology.

Ontology

- **Representation ontologies**
  - don't bellong to any field
  - organize the primitives of the logical theory (Charlet 2002)

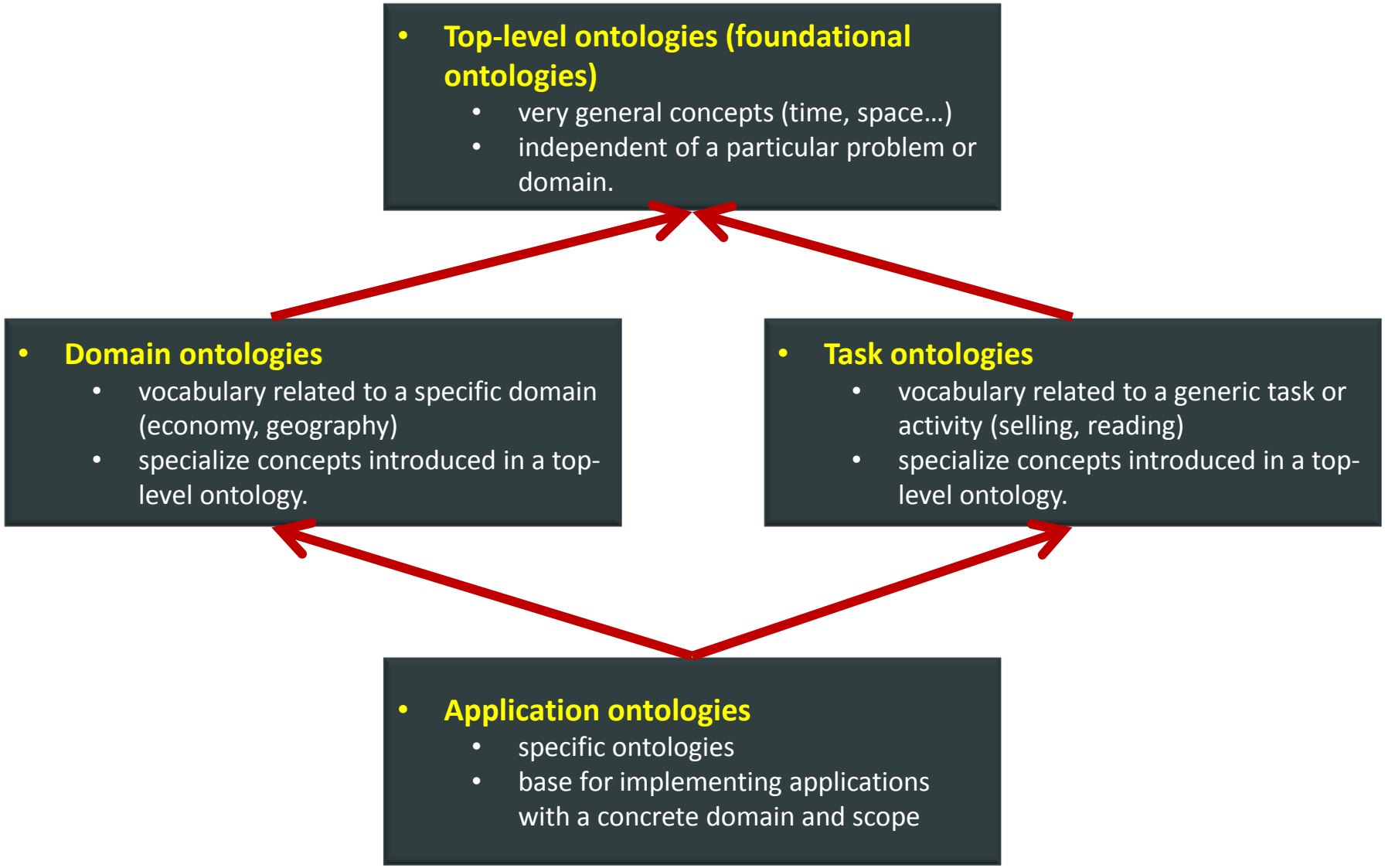
- **Generic ontologies (abstraction)**
  - abstract enough to be valid regardless of the field of study
  - upper ontology

- **Task ontologies (problem solving)**
  - provide concepts modeling a generic activity
  - explaini the role played by each concept in reasoning

- **Application ontologies**
  - double specialisation : modelise a specifique activity in a given domain

A classification system that uses the subject of conceptualization as a main criterion has been introduced by Guarino (1998).

Ontology



Ontology

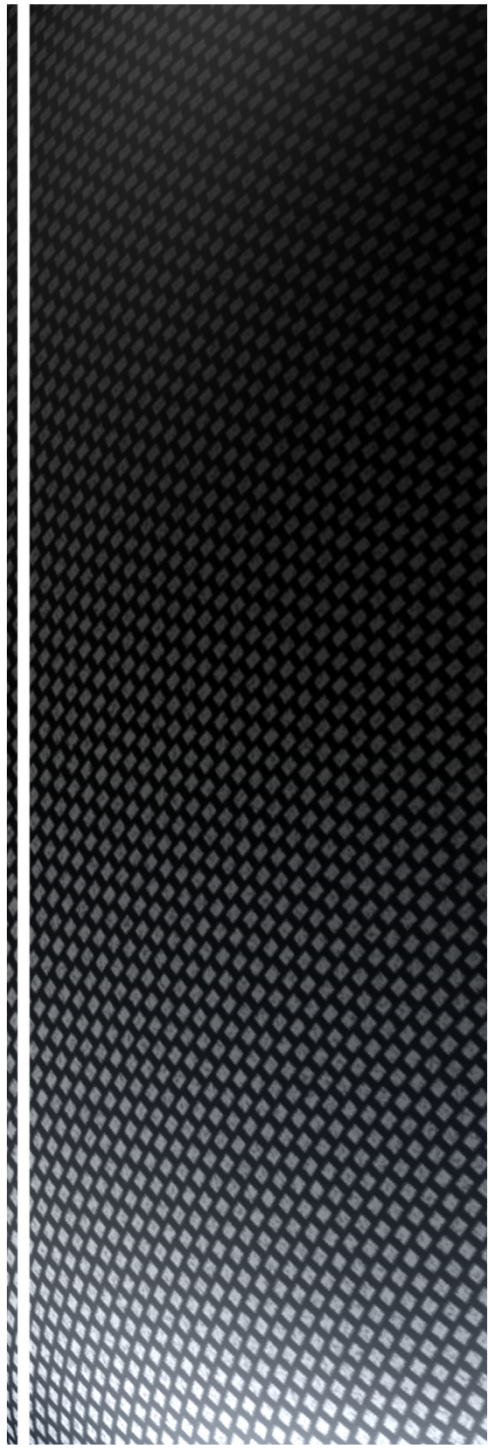
Concepts in relation

Ontology

- The term comes from philosophy
- A study of what ***is***

Continuum : data, information, knowledge

The knowledge enables the production of new data, informations, knowledges : inference



The RDF semantic and its main syntaxes

1. The **basic principles** of the Web of Data: from raw and heterogeneous Data to the lingua franca of the Web of Data
2. The **RDF** semantic and its main syntaxes
3. **Handling and querying** Data (triple store, end point, SPARQL)
4. Understanding the basics of **knowledge modeling** (ontology, OWL)
5. The **lifting and linking** of Data in order to publish 5 stars Data (DataLift).

We can decompose…

- Data
  - A base element, an atom, a quantity

16

- Information
  - A data that agreements structure, for example a unit of measure

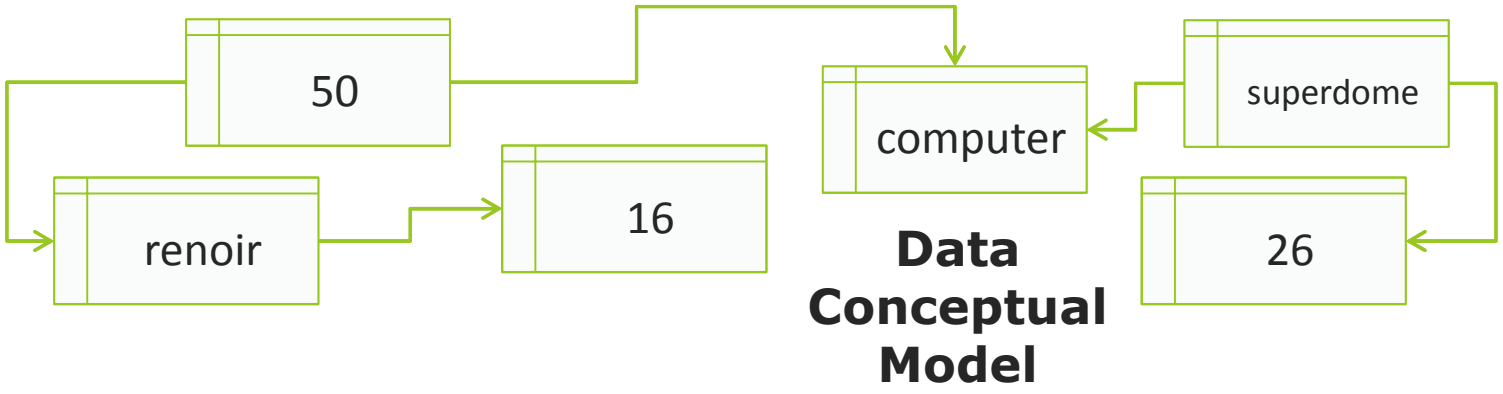
T= 16ºC

- Knowledge
  - A (re)usable and interpretable information with context

The superdome 50 is in the Renoir room whose temperature is 16°C and the air conditioning is on, is an HP computer that the manufacturer guarantees to be operational until 26ºC.

Rewriting complex knowledge

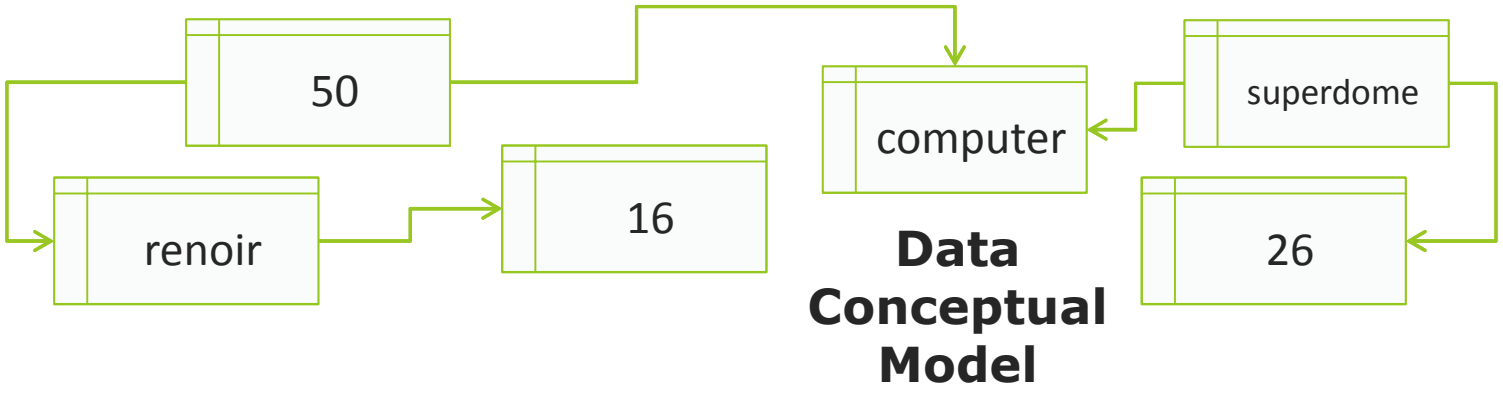
The superdome 50 is in the Renoir room whose temperature is 16°C and the air conditioning is on, is an HP computer that the manufacturer guarantees to be operational until 26°C.



Rewriting complex knowledge

The superdome 50 is in the Renoir room whose temperature is 16°C and the air conditioning is on, is an HP computer that the manufacturer guarantees to be operational until 26°C.

- **The rewriting**
  - **The superdome 50 is in the Renoir room**
  - **The Renoir room has the temperature 16°C**
  - **The air conditioning of the Renoir room is on**
  - **The superdome 50 is an HP computer**
  - **The superdome computer is guaranted operational by HP until 26°C**



Rewriting complex knowledge

The superdome 50 is in the Renoir room whose temperature is 16°C and the air conditioning is on, is an HP computer that the manufacturer guarantees to be operational until 26°C.

- **The rewriting**
  - **The superdome 50 is in the Renoir room**
  - **The Renoir room has the temperature 16°C**
  - **The air conditioning of the Renoir room is on**
  - **The superdome 50 is an HP computer**
  - **The superdome computers are guaranted operational by HP until 26°C**

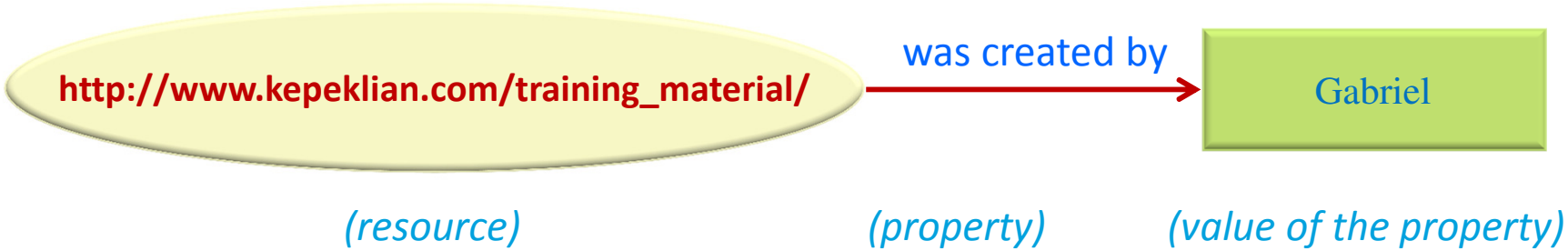
These are regularly formed sentences

Subject	Predicat	Object
The superdome 50	is in	the Renoir room
The Renoir room	has the temperature of	16°C
The air conditioning of the Renoir room	is	on
The superdome 50	is	HP computer
The superdomes computers	are guaranted operationed by HP until	26°c

- A shared grammar is needed to ensure that metadata are interpreted consistently
  - A framework to declare properties and relationships of items in the Web
  - A basic model to declare resources:
    - **Resource**: anything with an URI
    - **Description**: states the properties of the resource using terms named by URIs
    - **Framework**: a common model or grammar for declarations
  - Uses XML as a serialisation syntax

RDF = Resource Description Framework

RDF =
Resource Description Framework




```
<?xml version="1.0"?>
<RDF xmlns="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
        <Description about="http://www.kepeklian.com/training_material/">
                <Creator>Gabriel</Creator>
        </Description>
</RDF>
```

- That declaration can be read as:
  - *The training material identified by* **http://www.kepeklian.com/training_material/** *was* ***created*** *by* ***Gabriel***

Handling and querying Data (triple store, end point, SPARQL)

1. The **basic principles** of the Web of Data: from raw and heterogeneous Data to the lingua franca of the Web of Data
2. The **RDF** semantic and its main syntaxes
3. **Handling and querying** Data (triple store, end point, SPARQL)
4. Understanding the basics of **knowledge modeling** (ontology, OWL)
5. The **lifting and linking** of Data in order to publish 5 stars Data (DataLift).

The limits of the CSV files

The data structure limit

Separation between the data structure and the data itself

idcity	label-fr	Label.de	label-en	longitude	latitude
1	Athènes	Athen	Athens	23.716667	37.966667
2	Pékin	Pekin	Beijing	116.400002	39.900002
3	Londres	London	London	-0.121137	51.515161

The data structure limit

Separation between the data structure and the data itself

Idcity	label-fr	Label.de	label-en	longitude	latitude
1	Athènes	Athen	Athens	23.716667	37.966667
2	Pékin	Pekin	Beijing	116.400002	39.900002
3	Londres	London	London	-0.121137	51.515161

If we extract the data from the file, we have to extract also the structure to understand each data

The data structure limit

Separation between the data structure and the data itself

Idcity	label-fr	Label.de	label-en	longitude	latitude
1	Athènes	Athen	Athens	23.716667	37.966667
2	Pékin	Pekin	Beijing	116.400002	39.900002
3	Londres	London	London	-0.121137	51.515161

1

longitude

23.716667

23.716667

Alone :

no mining

no value

longitude

Alone :

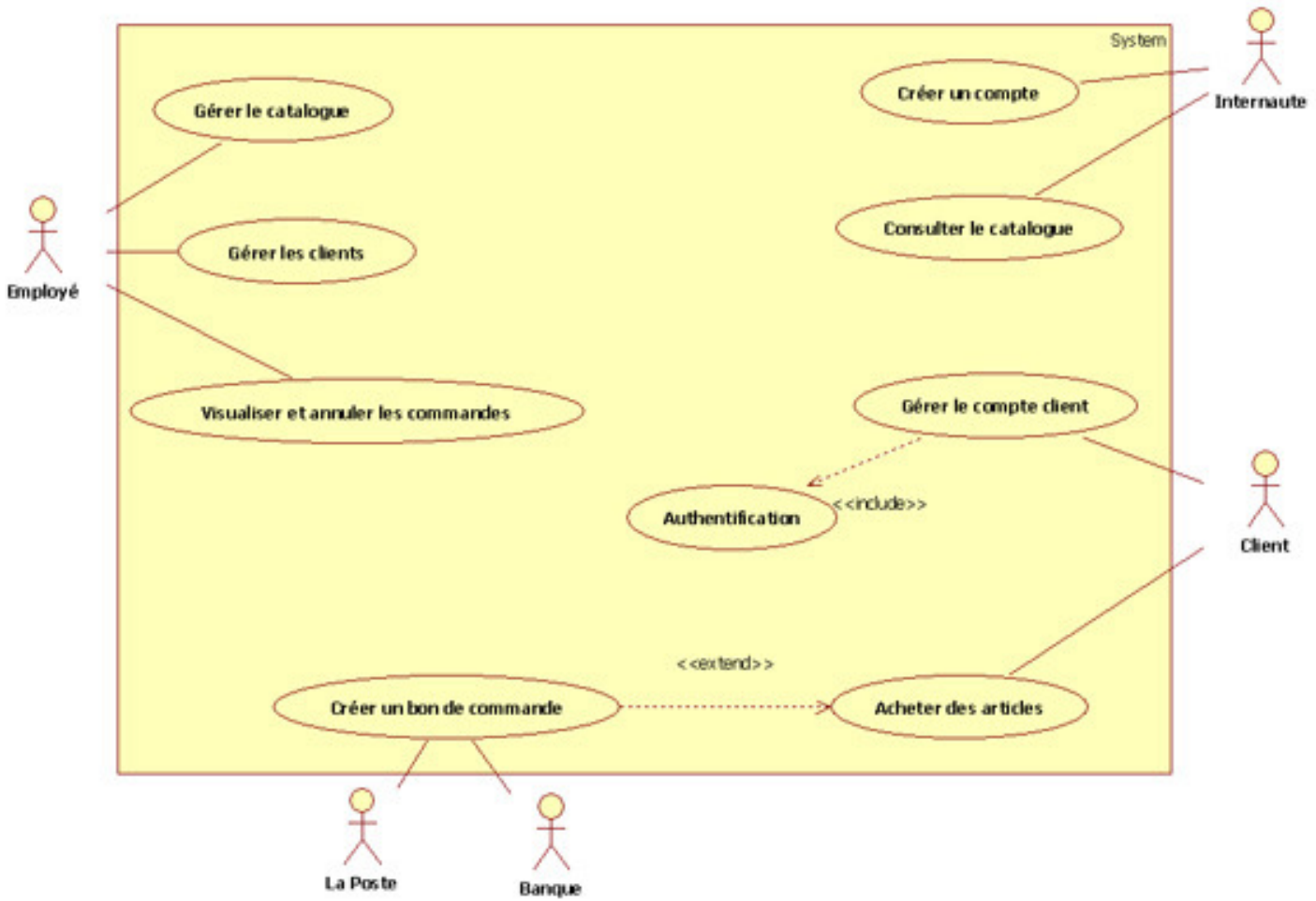
no mining

no value

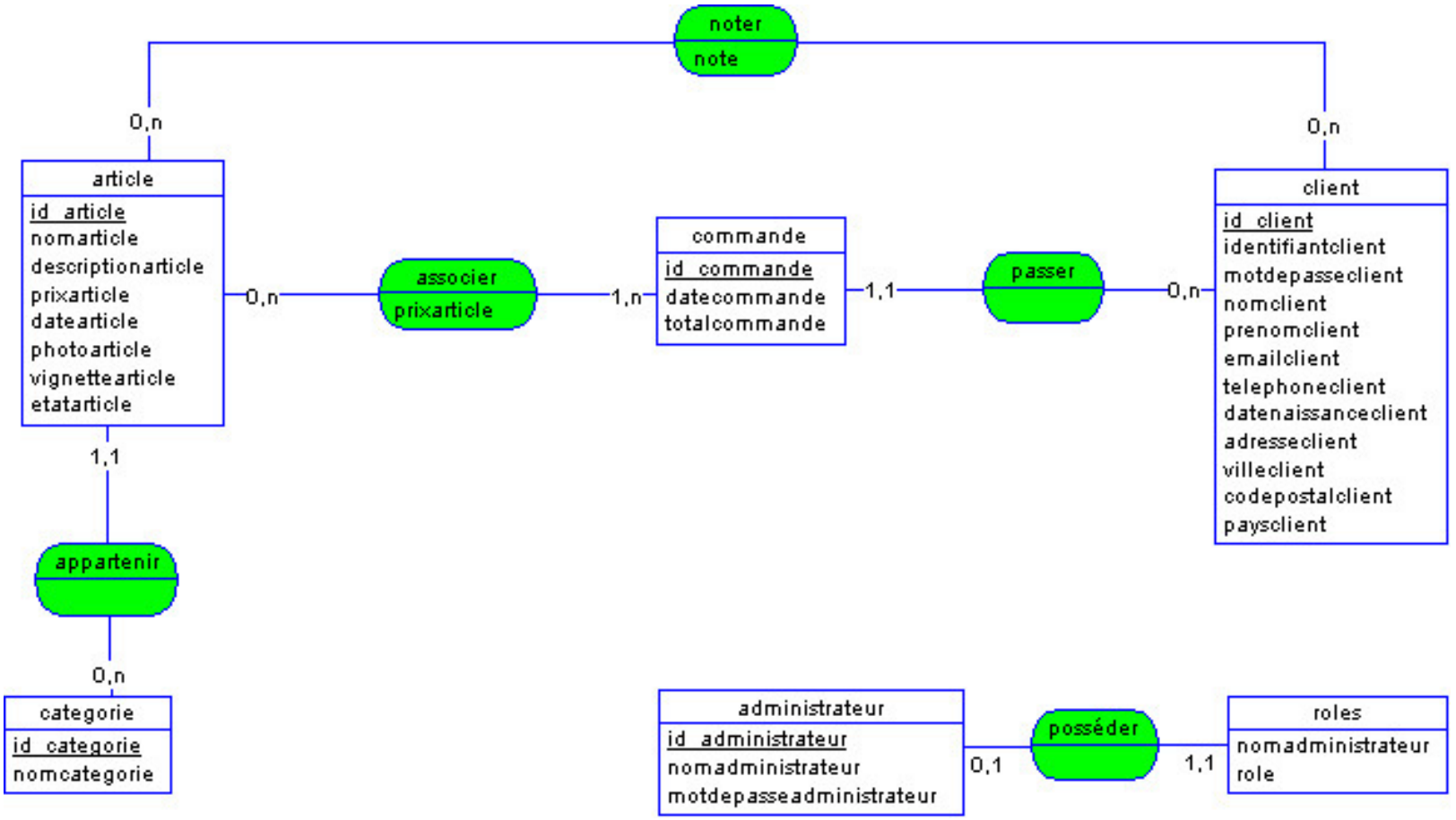
If we extract the data from the file, we have to extract also the structure to understand each data

The limits of the relationnal model

UML


System
Gérer le catalogue
Gérer les clients
Visualiser et annuler les commandes
Créer un compte
Consulter le catalogue
Gérer le compte client
Authentification
<<include>>
<<extend>>
Créer un bon de commande
Acheter des articles
Employé
Internaute
Client
La Poste
Banque

CDM


noter
note
0,n
0,n
article
id_article
nomarticle
descriptionarticle
prixarticle
datearticle
photoarticle
vignettearticle
etatarticle
0,n
associer
prixarticle
1,n
commande
id_commande
datecommande
totalcommande
1,1
passer
0,n
client
id_client
identifiantclient
motdepasseclient
nomclient
prenomclient
emailclient
telephoneclient
datenaissanceclient
adresseclient
villeclient
codepostalclient
paysclient
1,1
appartenir
0,n
categorie
id_categorie
nomcategorie
administrateur
id_administrateur
nomadministrateur
motdepasseadministrateur
0,1
posséder
1,1
roles
nomadministrateur
role

Here is a classic relational model describing the olympic games (a part of)

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

We will now observe the induced problems

A - The relational limits – the data structure

1 – Separation between the data structure and the data itself

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

A - The relational limits – the data structure

1 – Separation between the data structure and the data itself

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

If we extract the data from the database, we have to extract also the structure to understand each data

A - The relational limits – the data structure

1 – Separation between the data structure and the data itself

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

A data applies to a record, because it is associated with a field.
The relationship is induced by the structure of the table.

A - The relational limits – the data structure

1 – Separation between the data structure and the data itself

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

The data are not independent from each other.
They are understandable in the context of the database, of a record and of a field.

A - The relational limits – the data structure

2 – The structure of a database is rigid

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

A - The relational limits – the data structure

2 – The structure of a database is rigid

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

If a data is missing for a field in a record, a "NULL" value is added but it is fictional.

A - The relational limits – the data structure

2 – The structure of a database is rigid

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	Id atheletes repr. 2	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	2	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

If a data of the same nature is duplicate in a record, it is necessary to create another field or ...

A - The relational limits – the data structure

2 – The structure of a database is rigid

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	Id atheletes repr. 2	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	2	11028	One World one Dream

idjo	idAthleteRepresentant
2	1
2	2

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

… or a table

A - The relational limits – the data structure

2 – The structure of a database is rigid

idcity	label-fr	label-en	label-cn	longitude	latitude
1	Athènes	Athens	NULL	23.716667	37.966667
2	Pékin	Beijing	北京市	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

To manage multilingualism, we have to create fields for each language, while the meaning of the field is exactly the same, or create a specific table.

A - The relational limits – the data structure

3 – The relations between two tables are induced

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

A - The relational limits – the data structure

3 – The relations between two tables are induced

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

The relation between data from two tables is induced by the use of common identifiers called foreign keys. The nature of the relationship is not clearly expressed either in the structure or in the data.

A - The relational limits – the data structure

3 – The relations between two tables are induced

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

The extraction of a database does not highlight these relations. We have to retrieve data from different tables to maintain the relationship.

A - The relational limits – the data structure

4 – The identifier of a record is a data like all other

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

A - The relational limits – the data structure

4 – The identifier of a record is a data like all other

idcity	label-fr	label-en	longitude	latitude
1	Athènes	Athens	23.716667	37.966667
2	Pékin	Beijing	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

The identifier of a record does not have a standardized form.
It is independent from the database or from the table (so from the structure).

B - The relational limits - Data interoperability

1 - The identifiers are local and specific to a database

idcity	label-fr	label-en
1	Athènes	Athens
2	Pékin	Beijing

idcity	longitude	latitude
A	23.716667	37.966667
B	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

It is not possible to identify natively two equivalent resources between two different databases.

B - The relational limits - Data interoperability

2 - The names of fields are specific to a database

idcity	label-fr	label-en
1	Athènes	Athens
2	Pékin	Beijing

idcity	Name	longitude	latitude
A	Athènes	23.716667	37.966667
B	Pékin	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

The structure of a database is local. There are no standards for namming properties and to assign them to a normalization of a particular data type.

B - The relational limits - Data interoperability

3 - The database structure is not based on any inheritance mechanism

idcity	label-fr	label-en
1	Athènes	Athens
2	Pékin	Beijing

idcity	Name	longitude	latitude
A	Athènes	23.716667	37.966667
B	Pékin	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

Is that table a specialisation of the description of a person?

Is that table a specialisation of the description of an event?

It is not possible to connect a table to a generic model of local or external description which may inherit the characteristics. It requires to construct a CDM from scratch almost.

B - The relational limits - Data interoperability

4 - There is no standard representation for exchanging RDB on a network.

idcity	label-fr	label-en
1	Athènes	Athens
2	Pékin	Beijing

idcity	Name	longitude	latitude
A	Athènes	23.716667	37.966667
B	Pékin	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

The extraction of a database is specific for each database, there is no standard syntax for exchanging data in a database and merge with another base.

B - The relational limits - Data interoperability

5 - There is no standard way to query a BDR directly on the Web.

idcity	label-fr	label-en
1	Athènes	Athens
2	Pékin	Beijing

idcity	Name	longitude	latitude
A	Athènes	23.716667	37.966667
B	Pékin	116.400002	39.900002

id	id city	year	opened Ceremony Date	closed Ceremony Date	id athletes Representant	athletes Number	motto
1	1	1896	1896/04/06	1896/04/15	NULL	241	NULL
2	2	2008	2008/08/08	2008/08/24	1	11028	One World one Dream

idathlete	label-fr	label-en	label-cn	birthdate	deathdate
1	Zhang Yining	Zhang Yining	張怡寧	1981/10/05	NULL
2	John Doe	John Doe	NULL	1982/07/08	NULL

The extraction of a database is specific for each database, there is no standard syntax for exchanging data in a database and merge with another base.

The RDB is no more the good technology

- We have to handle more and more data
- We have to handle more and more heterogeneous data
- We have to handle more and more faster
- We have to handle more and more in a non intrusive manner

- How do we best handle data in such a context?

The official semantic web activity page says: "The Semantic Web provides a common framework that allows data to be shared and reused across application, enterprise, and community boundaries".

Triple Stores

Three architecture principles

In-Memory Stores

Temporary storage of triples in memory

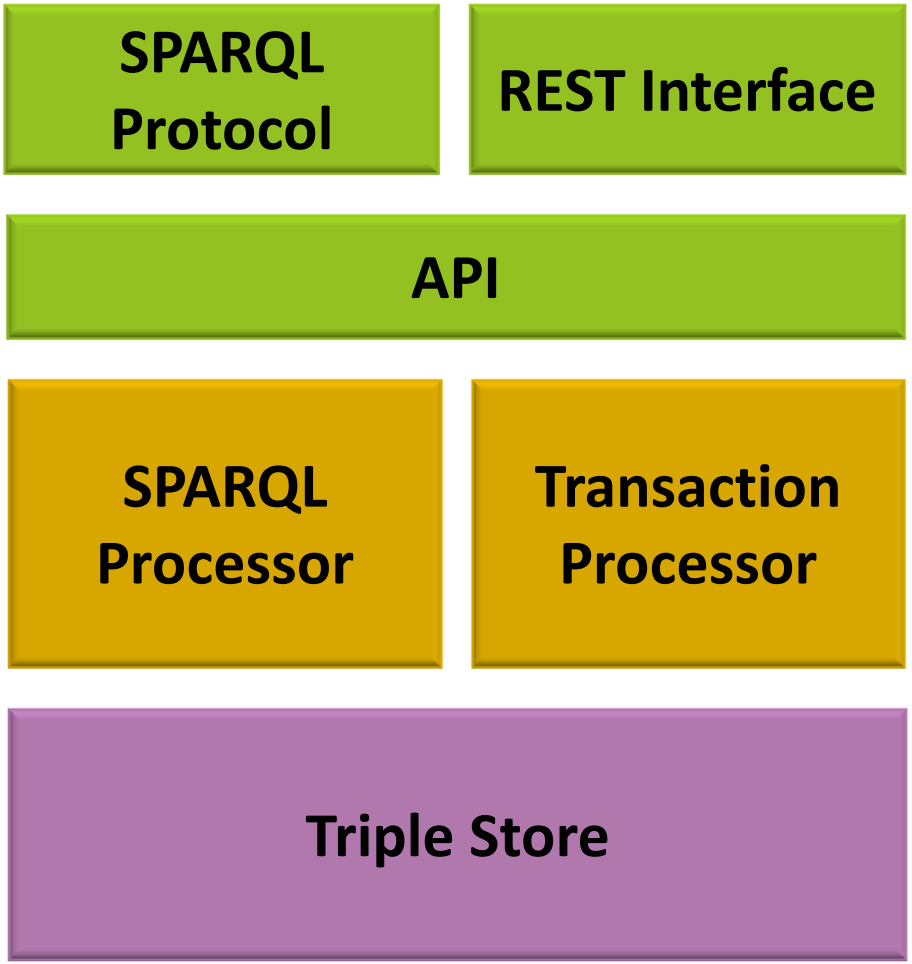
Native Stores

Persistent storage of triples in native stores with their own storage implementation

Non-Native Stores

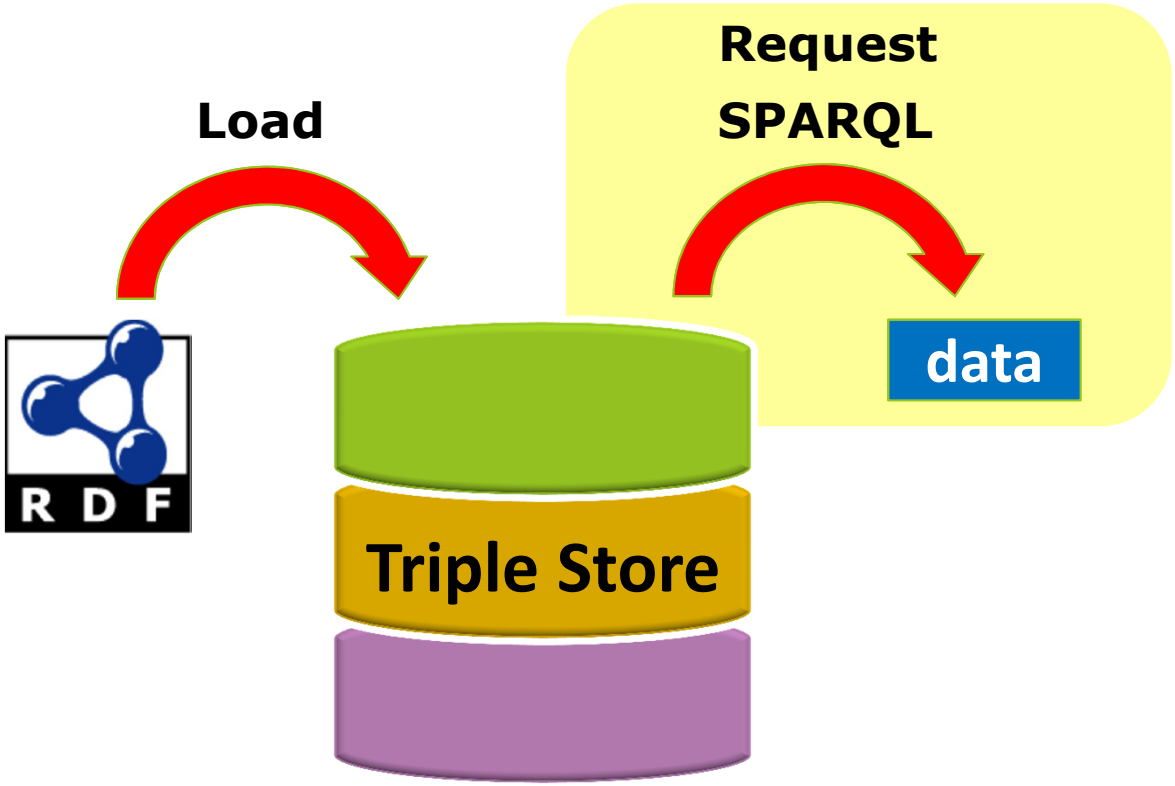
Persistent storage of triples on top of third-party databases

Triple Stores



- select, ask : query
- construct
- describe : export graph
- Insert, delete

SPARQL endpoint



SPARQL Endpoints Status

http://sparqles.okfn.org/

- SPARQL Endpoints Status tool monitors
  - the availability,
  - the performance,
  - the interoperability and
  - the discoverability
- of SPARQL Endpoints registered in Datahub.


SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }
SELECT ?o WHERE { ?s dbpedia2:blackboard ?o. }

SPARQL

SPARQL Protocol and RDF Query Language

Query, Access, Transformation

Data in RDF

A query language

An access protocol

An XML format for the results

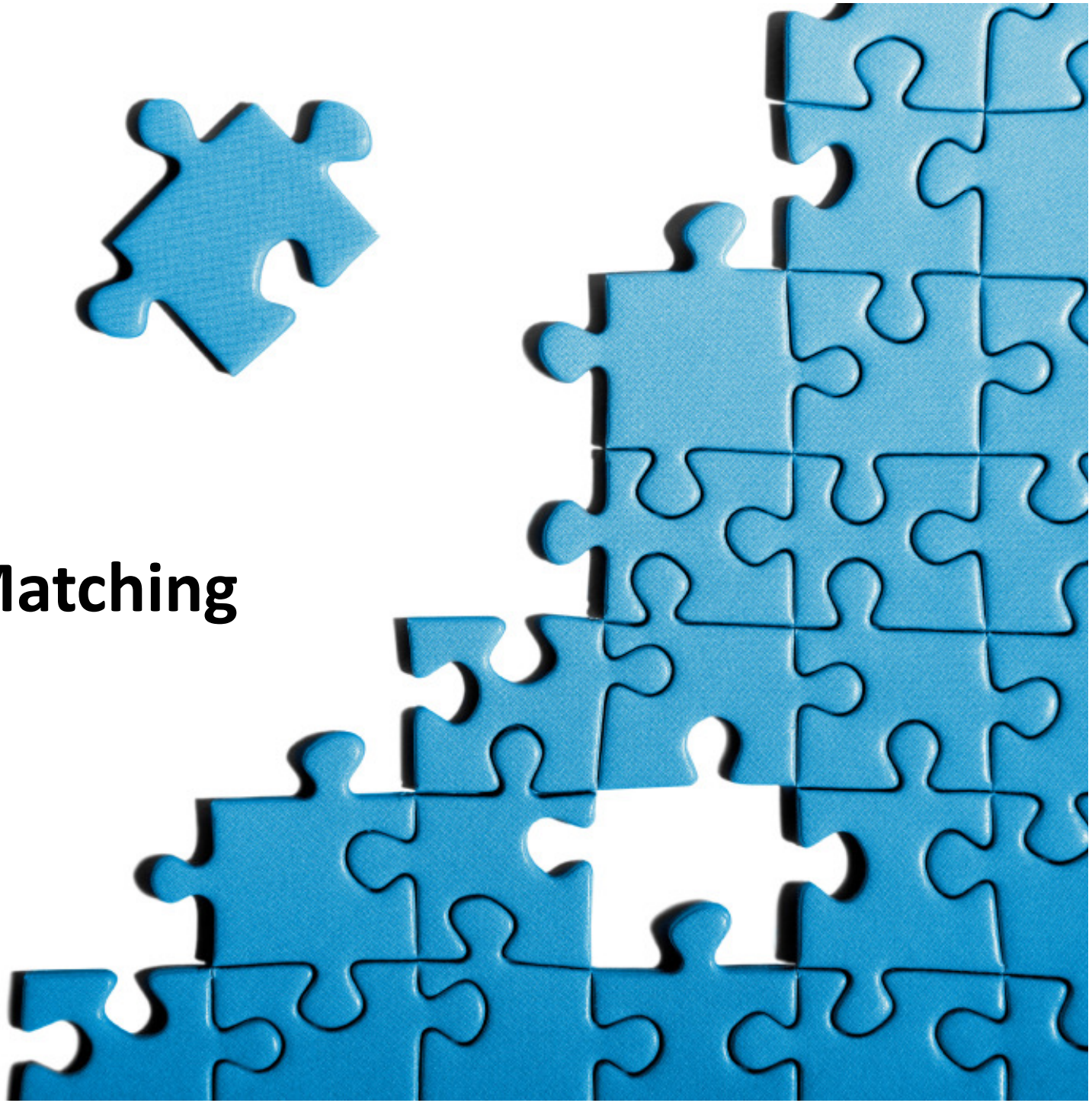
Introduction
requesting the RDF data

- **RDF** (Resource Description Framework)
  - a flexible and extensible way to represent the information of the Web resources
  - a graph model intended to describe formally Web resources and their metadata, and to allow automatic processing of such descriptions.

- A document is structured as a set of RDF triples.

- An RDF triple is an association (subject, predicate, object)
  - The **subject** is the resource being described
  - The **predicate** is a type of property applicable to this resource
  - The **object** represents a data or another resource : it is the value of the property.

Pattern Matching



- SPARQL (SPARQL Protocol and RDF Query Language)

- Only 4 reading possibilities

  - **SELECT** query
    - Extracts the data from a SPARQL endpoint, the results are in an **array**.

  - **ASK** query
    - Requests a SPARQL endpoint to obtain a **boolean answer**.

  - **CONSTRUCT** query
    - Extracts the data from a SPARQL endpoint in order to transform them in **triples** (RDF format).

  - **DESCRIBE** query
    - Use to extract a part of the RDF **graph** in a silo. Returns all triples containing all URIs that meet the requirements of pattern matching specified in the query. URIs can be present as subject or object of the triples.

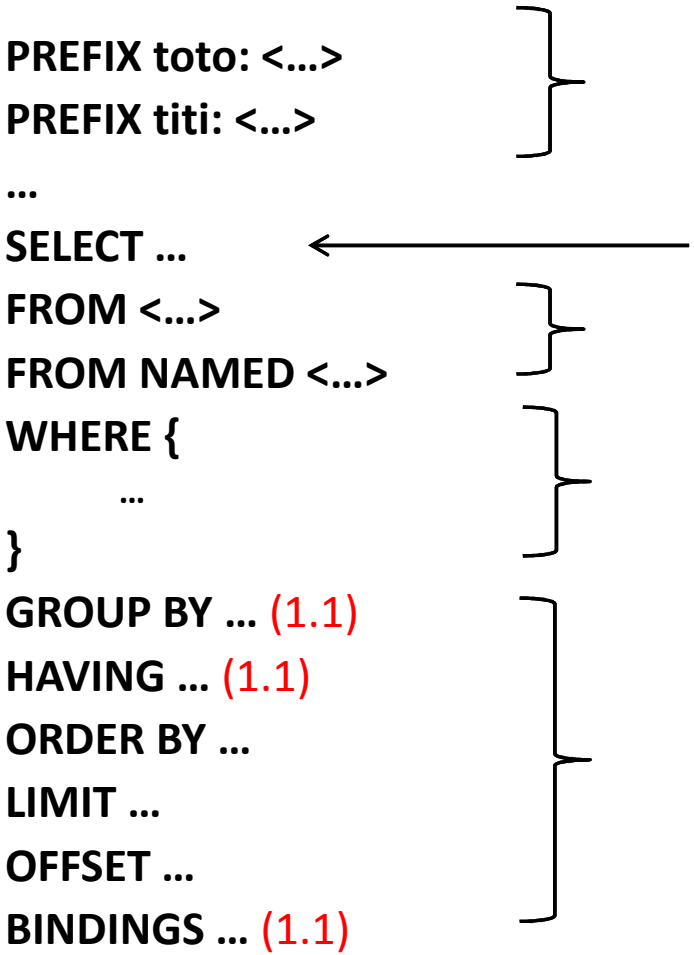
W3C : 2008,
SPARQL 1.0

- SPARQL 1.1 looks more and more SQL
- It allows read operations and write data without the need to use or know specific languages: RDF or RDFS, OWL ...

- Two new functions:
  - insert Data
    - Insertion of new triplets
  - Delete Data
  - Remove selected triplets

W3C : 2013, SPARQL 1.1

SELECT, anatomy of a query



Prefix declaration (optional)
No space between the prefix name and the semi colon

Clause of definition of the results

To define the data set (optional)

Conditions to follow

Modifiers of the results (optional)

I want to know the URI and the name of the persons who know Jean Dupond, and I search in his FOAF profile.

Jean Dupond knows persons.
These persons are named ???.

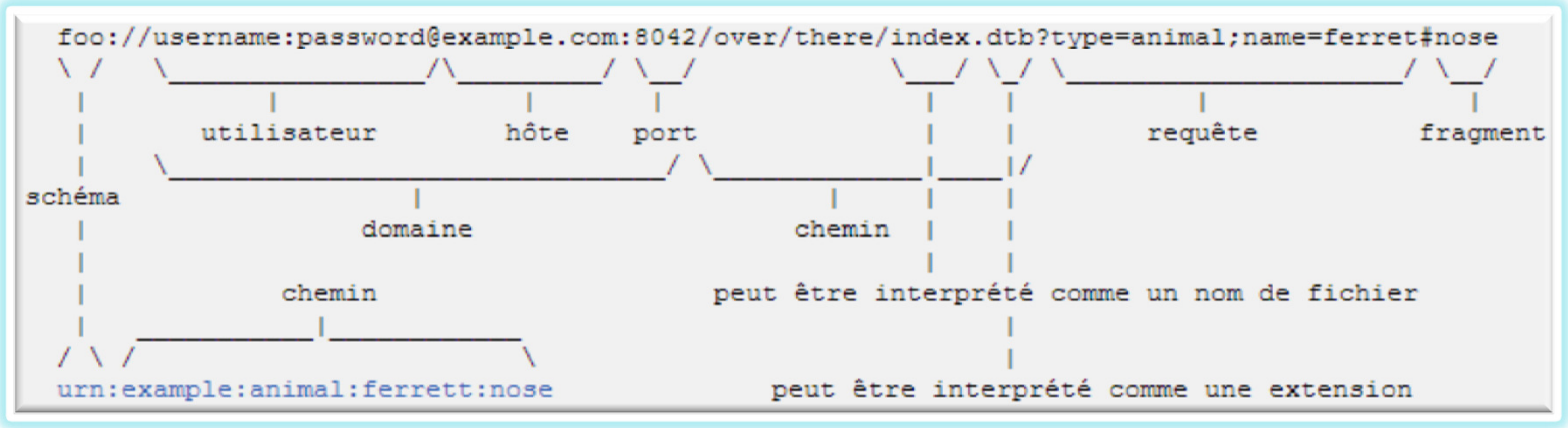
```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>

SELECT ?persons ?name

FROM <http://www.jean_dupond.com/foaf.rdf>

WHERE {

        <http://www.jean_dupond.com/foaf.rdf/i> foaf:knows ?persons.
        ?persons foaf:name ?name.

}
```

URI

RFC 4935

When we need to name the things

- ftp://ftp.is.co.za/rfc/rfc1808.txt
- **http://www.dila.premier-ministre.gouv.fr/qui-sommes-nous/dila.html**
- ldap://[2001:db8::7]/c=GB?objectClass?one
- **mailto:thomas.saint-aubin@dila.gouv.fr**
- news:comp.infosystems.www.servers.unix
- **tel:+1-816-555-1212**
- telnet://192.0.2.16:80/
- urn:oasis:names:specification:docbook:dtd:xml:4.1.2



PREFIX

- **PREFIX... they rewrite**

```
PREFIX rdf: <http://xmlns.com/foaf/0.1/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
```

- **Where can we find others?**

LITERALS

- *Plain literals*
  - **“un exemple de litéral”**
- *Plain literals with the mentions of a country*
  - **“bonjour les amis”@fr**
- *Typed literals*
  - **“123”^^xsd:integer**
- *Shortcuts*
  - **true** ➔ “true”^^xsd:boolean
  - **3** ➔ “3”^^xsd:integer
  - **4.2** ➔ “4.2”^^xsd:decimal
  - **30/06/2014** ➔ “2014-06-30”^^xsd:date
  - **30/06/2014 12:00** ➔ “2014-06-30T12:00:00+02:00”^^xsd:datetime

VARIABLES

- *Variables:*
  - **?var1 ?encoreUneVar $etUneAutre**

- Never start with a digit
- Case sensitive
- Without space
- Meaningful because they are used to name columns (SPARQL 1.1 introducts AS to give the possibility to change the column name)

FILTER

- To complete the way to find triples according to a given search, SPARQL offers a mechanism of filters.
- A filter is expressed in a line placed in the mask of the graph.
- Example:

```
?s1 ?p1 ?o1 .
?s2 ?p2 ?o2 .
FILTER ( expression )
```

- The filter removes the solutions where the expression is false.

FROM

FROM <url> is used to identify the content in the default graph.

```
SELECT (COUNT(*) as ?n)
FROM <http://localhost:9091/project/kiosques/source/kiosques-ouverts-csv-rdf-1>
WHERE { ?s ?p ?o }
```

It is equivalent to the folling request in a named graph.

```
PREFIX g: <http://localhost:9091/project/kiosques/source/kiosques-ouverts-csv-rdf-1>
SELECT (COUNT(*) as ?n)
WHERE { GRAPH g: { ?s ?p ?o } }
```

FROM NAMED

```
<http://grapha.com> = { <a1> <p> <a2> . }
<http://graphb.com> = { <b1> <p> <b2> . }
<http://graphc.com> = { <c1> <p> <c2> . }
<http://graphd.com> = { <d1> <p> <d2> . }
```

```
SELECT ?s WHERE { ?s <p> ?o }
➔ <a1>, <b1> <c1>, <d1>

FROM <http://grapha.com>
SELECT ?s WHERE { ?s <p> ?o }
➔ <a1>.

FROM NAMED <http://grapha.com>
SELECT ?s WHERE { ?s <p> ?o }
➔ rien
```

```
FROM <http://grapha.com>
FROM <http://graphb.com>
FROM NAMED <http://graphc.com>
FROM NAMED <http://graphd.com>
SELECT ?s WHERE { ?s <p> ?o }
➔ <a1>, <b1>.

FROM <http://grapha.com>
FROM <http://graphb.com>
FROM NAMED <http://graphc.com>
FROM NAMED <http://graphd.com>
SELECT ?s WHERE { GRAPH ?g { ?s <p> ?o }}
➔ <c1>, <d1>.
```

THE COMPONENTS OF A QUERY

- **Prolog**
  - PREFIX Definition

```
PREFIX g: <http://localhost:9091/project/kiosques/source/kiosques-ouverts-csv-rdf-1>
SELECT *
WHERE { GRAPH g: { ?s ?p ?o } }
```

- **Result**
  - SELECT, DESCRIBE, CONSTRUCT or ASK

```
PREFIX g: <http://localhost:9091/project/kiosques/source/kiosques-ouverts-csv-rdf-1>
SELECT *
WHERE { GRAPH g: { ?s ?p ?o } }
```

- **Specification of the data set**

```
SELECT *
FROM <http://localhost:9091/project/kiosques/source/kiosques-ouverts-csv-rdf-1>
WHERE { ?s ?p ?o }
```

THE COMPONENTS OF A QUERY

- **Pattern**
  - The things we are looking for

```
PREFIX g: <http://localhost:9091/project/kiosques/source/kiosques-ouverts-csv-rdf-1>
SELECT ?s
WHERE { GRAPH g: { ?s ?p ?o } }
```

- **Modifiers**
  - ORDER BY, LIMIT, OFFSET

```
PREFIX g: <http://localhost:9091/project/kiosques/source/kiosques-ouverts-csv-rdf-1>
SELECT *
WHERE { GRAPH g: { ?s ?p ?o } }
ORDER BY ?s
OFFSET 20
LIMIT 50
```

BIND

Data

```
@prefix dc:
<http://purl.org/dc/elements/1.1/> .
@prefix :      <http://example.org/book/>
.
@prefix ns:    <http://example.org/ns#> .

:book1   dc:title      "SPARQL Tutorial" .
:book1   ns:price      42 .
:book1   ns:discount   0.2 .

:book2   dc:title      "The Semantic Web" .
:book2   ns:price      23 .
:book2   ns:discount   0.25 .
```

Query

```
PREFIX  dc:
<http://purl.org/dc/elements/1.1/>
PREFIX  ns:  <http://example.org/ns#>

SELECT  ?title ?price
{   ?x ns:price ?p .
    ?x ns:discount ?discount
    BIND (?p*(1-?discount) AS ?price)
    FILTER(?price < 20)
    ?x dc:title ?title .
}
```

Another query

```
PREFIX  dc:
<http://purl.org/dc/elements/1.1/>
PREFIX  ns:  <http://example.org/ns#>

SELECT  ?title ?price
{   { ?x ns:price ?p .
      ?x ns:discount ?discount
      BIND (?p*(1-?discount) AS ?price)
    }
    {?x dc:title ?title . }
    FILTER(?price < 20)
}
```

title	price
"The Semantic Web"	17.25

AGGREGATION

Data

```
@prefix : <http://books.example/> .

:org1 :affiliates :auth1, :auth2 .
:auth1 :writesBook :book1, :book2 .
:book1 :price 9 .
:book2 :price 5 .
:auth2 :writesBook :book3 .
:book3 :price 7 .
:org2 :affiliates :auth3 .
:auth3 :writesBook :book4 .
:book4 :price 7 .
```

Query

```
PREFIX : <http://books.example/>
SELECT (SUM(?lprice) AS ?totalPrice)
WHERE {
  ?org :affiliates ?auth .
  ?auth :writesBook ?book .
  ?book :price ?lprice .
}
GROUP BY ?org
HAVING (SUM(?lprice) > 10)
```

totalPrice
21

GROUP BY

- **GROUP BY :** In order to calculate total values for a solution, the solution is first divided into one or more groups, and the total value is calculated for each group.

Query

```
SELECT (AVG(?y) AS ?avg)
WHERE {
  ?a :x ?x ;
     :y ?y .
}
GROUP BY ?x
```

- **NB :** AVG, the average function.

SUB REQUEST

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" .
```

Query

```
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"

This is achieved by first evaluating the internal request

```
SELECT ?y (MIN(?name) AS ?minName)
WHERE {
  ?y :name ?name .
} GROUP BY ?y
```

We obtain the following solutions

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

that are then binded with the results of the external query

y
:bob
:carol

Return a name (the one with the lowest sort order) for all the people that know Alice and have a name.

DUPLICATE RESULTS

Data

```
@prefix  foaf:  <http://xmlns.com/foaf/0.1/> .

_:x    foaf:name    "Alice" .
_:x    foaf:mbox    <mailto:alice@example.com> .

_:y    foaf:name    "Alice" .
_:y    foaf:mbox    <mailto:asmith@example.com> .

_:z    foaf:name    "Alice" .
_:z    foaf:mbox    <mailto:alice.smith@example.com>
.
```

Query

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

DUPLICATE RESULTS

Data

```
@prefix  foaf:  <http://xmlns.com/foaf/0.1/> .

_:x    foaf:name    "Alice" .
_:x    foaf:mbox    <mailto:alice@example.com> .

_:y    foaf:name    "Alice" .
_:y    foaf:mbox    <mailto:asmith@example.com> .

_:z    foaf:name    "Alice" .
_:z    foaf:mbox    <mailto:alice.smith@example.com>
.
```

Query

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

name
"Alice"
"Alice"
"Alice"

DUPLICATE RESULTS

Data

```
@prefix  foaf:  <http://xmlns.com/foaf/0.1/> .

_:x    foaf:name    "Alice" .
_:x    foaf:mbox    <mailto:alice@example.com> .

_:y    foaf:name    "Alice" .
_:y    foaf:mbox    <mailto:asmith@example.com> .

_:z    foaf:name    "Alice" .
_:z    foaf:mbox    <mailto:alice.smith@example.com>
.
```

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

Query

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

name
"Alice"
"Alice"
"Alice"

DUPLICATE RESULTS

Data

```
@prefix  foaf:  <http://xmlns.com/foaf/0.1/> .

_:x    foaf:name    "Alice" .
_:x    foaf:mbox    <mailto:alice@example.com> .

_:y    foaf:name    "Alice" .
_:y    foaf:mbox    <mailto:asmith@example.com> .

_:z    foaf:name    "Alice" .
_:z    foaf:mbox    <mailto:alice.smith@example.com>
.
```

Query

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

name
"Alice"
"Alice"
"Alice"

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

name
"Alice"

TYPES

xsd:integer
xsd:decimal
xsd:float
xsd:double
xsd:string
xsd:boolean
xsd:dateTime

xsd:nonPositiveInteger
xsd:negativeInteger
xsd:long
xsd:int
xsd:short
xsd:byte
xsd:nonNegativeInteger
xsd:unsignedLong
xsd:unsignedInt
xsd:unsignedShort
xsd:unsignedByte
xsd:positiveInteger

TYPES & CASTS

xsd:integer
xsd:decimal
xsd:float
xsd:double
xsd:string
xsd:boolean
xsd:dateTime

xsd:nonPositiveInteger
xsd:negativeInteger
xsd:long
xsd:int
xsd:short
xsd:byte
xsd:nonNegativeInteger
xsd:unsignedLong
xsd:unsignedInt
xsd:unsignedShort
xsd:unsignedByte
xsd:positiveInteger

```
SELECT ?n
WHERE
{
  BIND (xsd:integer("34") AS ?n)
}
```

IF (exp1, exp2, exp3)

?x = 2, ?z = 0,?y unbounded.

IF(?x = 2, "yes", "no") ➔ ???
IF(bound(?y), "yes", "no") ➔ ???
IF(?x=2, "yes", 1/?z) ➔ ???
IF(?x=1, "yes", 1/?z) ➔ ???
IF("2" > 1, "yes", "no") ➔ ???

IF (exp1, exp2, exp3)

?x = 2, ?z = 0,?y unbounded.

IF(?x = 2, "yes", "no")	➔ "yes"
IF(bound(?y), "yes", "no")	➔ "no"
IF(?x=2, "yes", 1/?z)	➔ "yes", and the expression 1/?z is not evaluated
IF(?x=1, "yes", 1/?z)	➔ error
IF("2" > 1, "yes", "no")	➔ error

IF (exp1, exp2, exp3)

?x = 2, ?z = 0,?y unbounded.

IF(?x = 2, "yes", "no") ➔ "yes"
IF(bound(?y), "yes", "no") ➔ "no"
IF(?x=2, "yes", 1/?z) ➔ "yes", and the expression 1/?z is not evaluated
IF(?x=1, "yes", 1/?z) ➔ error
IF("2" > 1, "yes", "no") ➔ error

```
SELECT ?result
WHERE
{
  BIND (IF (3>4, « True", « False") AS ?result)
}
```

Data

```
@prefix foaf:       <http://xmlns.com/foaf/0.1/> .

_:a  foaf:name       "Alice".
_:a  foaf:mbox       <mailto:alice@work.example> .

_:b  foaf:name       "Ms A.".
_:b  foaf:mbox       <mailto:alice@work.example> .
```

name1	name2
"Alice"	"Ms A."
"Ms A."	"Alice"

Query 1

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name1 ?name2
WHERE { ?x foaf:name  ?name1 ;
           foaf:mbox  ?mbox1 .
        ?y foaf:name  ?name2 ;
           foaf:mbox  ?mbox2 .
        FILTER (sameTerm(?mbox1, ?mbox2) && !sameTerm(?name1, ?name2))
      }
```

Query 2

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name1 ?name2
WHERE { ?x foaf:name  ?name1 ;
          foaf:mbox  ?mbox1 .
          ?y foaf:name  ?name2 ;
          foaf:mbox  ?mbox2 .
          FILTER (?mbox1 = ?mbox2 && ?name1 != ?name2)
        }
```

IRI Syntax

Different writing of the same IRI

<http://example.org/book/book1>

BASE <http://example.org/book/>
<book1>

PREFIX book: <http://example.org/book/>
book:book1

Different writing of the same query

```
PREFIX  dc: <http://purl.org/dc/elements/1.1/>
SELECT  ?title
WHERE   { <http://example.org/book/book1> dc:title ?title }
```

```
PREFIX  dc: <http://purl.org/dc/elements/1.1/>
PREFIX  : <http://example.org/book/>
SELECT  $title
WHERE   { :book1  dc:title  $title }
```

```
BASE    <http://example.org/book/>
PREFIX  dc: <http://purl.org/dc/elements/1.1/>
SELECT  $title
WHERE   { <book1>  dc:title  ?title }
```

Triple Syntax

Triple Syntax

```
?x  foaf:name  ?name ;
    foaf:mbox  ?mbox .
                =
?x  foaf:name  ?name .
?x  foaf:mbox  ?mbox .
```

```
?x foaf:nick  "Alice" , "Alice_" .
                =
?x  foaf:nick  "Alice" .
?x  foaf:nick  "Alice_" .
```

```
?x  foaf:name ?name ; foaf:nick  "Alice" , "Alice_" .
                =
?x  foaf:name  ?name .
?x  foaf:nick  "Alice" .
?x  foaf:nick  "Alice_" .
```

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

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

Groups of patterns

```
{   ?x foaf:name ?name .
    ?x foaf:mbox ?mbox .
    FILTER regex(?name, "Smith")
}
```

```
{   FILTER regex(?name, "Smith")
    ?x foaf:name ?name .
    ?x foaf:mbox ?mbox .
}
```

```
{   ?x foaf:name ?name .
    FILTER regex(?name, "Smith")
    ?x foaf:mbox ?mbox .
}
```

FILTER

Data

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

_:a  rdf:type        foaf:Person .
_:a  foaf:name       "Alice" .
_:a  foaf:mbox       <mailto:alice@example.com> .
_:a  foaf:mbox       <mailto:alice@work.example> .
_:b  rdf:type        foaf:Person .
_:b  foaf:name       "Bob" .
```

Query

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

Optional Pattern

name	mbox
"Alice"	<mailto:alice@example.com>
"Alice"	<mailto:alice@work.example>
"Bob"	

```
@prefix dc:   <http://purl.org/dc/elements/1.1/> .
@prefix :     <http://example.org/book/> .
@prefix ns:   <http://example.org/ns#> .

:book1  dc:title  "SPARQL Tutorial" .
:book1  ns:price  42 .
:book2  dc:title  "The Semantic Web" .
:book2  ns:price  23 .
```

title	price
"SPARQL Tutorial"	
"The Semantic Web"	23

```
PREFIX  dc:  <http://purl.org/dc/elements/1.1/>
PREFIX  ns:  <http://example.org/ns#>
SELECT  ?title ?price
WHERE   { ?x dc:title ?title .
          OPTIONAL { ?x ns:price ?price . FILTER (?price < 30) }
        }
```

Test the absence of a pattern

Data

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

:alice  rdf:type   foaf:Person .
:alice  foaf:name  "Alice" .
:bob    rdf:type   foaf:Person .
```

Query

```
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 }
}
```

person
<http://example/bob>

Test the presence of a pattern

Data

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

:alice  rdf:type   foaf:Person .
:alice  foaf:name  "Alice" .
:bob    rdf:type   foaf:Person .
```

Query

```
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 EXISTS { ?person foaf:name ?name }
}
```

person
<http://example/alice>

Minus

Data

```
@prefix :       <http://example/> .
@prefix foaf:   <http://xmlns.com/foaf/0.1/> .

:alice  foaf:givenName "Alice" ;
        foaf:familyName "Smith" .

:bob    foaf:givenName "Bob" ;
        foaf:familyName "Jones" .

:carol  foaf:givenName "Carol" ;
        foaf:familyName "Smith" .
```

Query

```
PREFIX :       <http://example/>
PREFIX foaf:   <http://xmlns.com/foaf/0.1/>

SELECT DISTINCT ?s
WHERE {
   ?s ?p ?o .
   MINUS {
      ?s foaf:givenName "Bob" .
   }
}
```

s
<http://example/carol>
<http://example/alice>

LISTS
IN & NOT IN

```
2 IN (1, 2, 3)                              ???
2 IN ()                                     ???
2 IN (<http://example/iri>, "str", 2.0)     ???
2 IN (1/0, 2)                               ???
2 IN (2, 1/0)                               ???
2 IN (3, 1/0)                               ???
```

```
2 NOT IN (1, 2, 3)                          ???
2 NOT IN ()                                 ???
2 NOT IN (<http://example/iri>, "str", 2.0) ???
2 NOT IN (1/0, 2)                           ???
2 NOT IN (2, 1/0)                           ???
2 NOT IN (3, 1/0)                           ???
```

LISTS
IN & NOT IN

```
2 IN (1, 2, 3)                              true
2 IN ()                                     false
2 IN (<http://example/iri>, "str", 2.0)     true
2 IN (1/0, 2)                               true
2 IN (2, 1/0)                               true
2 IN (3, 1/0)                               error
```

```
2 NOT IN (1, 2, 3)                          false
2 NOT IN ()                                 true
2 NOT IN (<http://example/iri>, "str", 2.0) false
2 NOT IN (1/0, 2)                           false
2 NOT IN (2, 1/0)                           false
2 NOT IN (3, 1/0)                           error
```

STRINGS

- STRSTARTS, STRENDS, CONTAINS, STRBEFORE & STRAFTER

Argument1	Argument2	Compatible?
"abc"	"b"	Oui
"abc"	"b"^^xsd:string	Oui
"abc"^^xsd:string	"b"	Oui
"abc"^^xsd:string	"b"^^xsd:string	Oui
"abc"@en	"b"	Oui
"abc"@en	"b"^^xsd:string	Oui
"abc"@en	"b"@en	Oui
"abc"@fr	"b"@ja	Non
"abc"	"b"@ja	Non
"abc"	"b"@en	Non
"abc"^^xsd:string	"b"@en	Non

```
strStarts("foobar", "foo")                          true
strStarts("foobar"@en, "foo"@en)                    True
strEnds("foobar"^^xsd:string, "bar"^^xsd:string)    true
strEnds("foobar"^^xsd:string, "bar")                True
contains("foobar", "bar"^^xsd:string)               true
contains("foobar"@en, "foo")                        true
strbefore("abc","b")                "a"
strbefore("abc"@en,"bc")            "a"@en
strbefore("abc"@en,"b"@cy)          error
strbefore("abc"^^xsd:string,"")     ""^^xsd:string
strbefore("abc","xyz")              ""
strafter("abc"@en, "z")             ""
strafter("abc"@en, ""@en)           "abc"@en
strafter("abc"@en, "")              "abc"@en
```

STRINGS

STRLEN

```
strlen("chat")                4
strlen("chat"@en)             4
strlen("chat"^^xsd:string)    4
```

CONCAT

```
concat("foo", "bar")                            "foobar"
concat("foo"@en, "bar"@en)                      "foobar"@en
concat("foo"^^xsd:string, "bar"^^xsd:string)    "foobar"^^xsd:string
concat("foo", "bar"^^xsd:string)                "foobar"
concat("foo"@en, "bar")                         "foobar"
concat("foo"@en, "bar"^^xsd:string)             "foobar"
```

SUBSTR

```
substr("foobar", 4)                   "bar"
substr("foobar"@en, 4)                "bar"@en
substr("foobar"^^xsd:string, 4)       "bar"^^xsd:string
substr("foobar", 4, 1)                "b"
substr("foobar"@en, 4, 1)             "b"@en
substr("foobar"^^xsd:string, 4, 1)    "b"^^xsd:string
```

Catogories	Functions / Operators	Examples
Logic & Compaisons	!, &&, ||, =, !=, <, <=, >, >=, IN, NOT IN	?hasPermit || ?age < 25
Conditions (SPARQL 1.1)	EXISTS, NOT EXISTS, IF, COALESCE	NOT EXISTS { ?p foaf:mbox ?email }
Math	+, -, *, /, abs, round, ceil, floor, RAND	?decimal * 10 > ?minPercent
Strings (SPARQL 1.1)	STRLEN, SUBSTR, UCASE, LCASE, STRSTARTS, CONCAT, STRENDS, CONTAINS, STRBEFORE, STRAFTER	STRLEN(?description) < 255
Date/time (SPARQL 1.1)	now, year, month, day, hours, minutes, seconds, timezone, tz	month(now()) < 4
Tests SPARQL	isURI, isBlank, isLiteral, isNumeric, bound	isURI(?person) || !bound(?person)
Constructors (SPARQL 1.1)	URI, BNODE, STRDT, STRLANG, UUID, STRUUID	STRLANG(?text, “en”) = “hello”@en
Accesseurs	str, lang, datatype	lang(?title) = “en”
Hashing (1.1)	MD5, SHA1, SHA256, SHA512	BIND(SHA256(?email) AS ?hash)
Miscellaneous	sameTerm, langMatches, regex, REPLACE	regex(?ssn, “\\d{3}-\\d{2}-\\d{4}”)

Construct

CONSTRUCT

```
CONSTRUCT {
    #the form of triples to be constructed
}
WHERE {
    #the selected constituents
}
```

```
CONSTRUCT {
    ?x  a  foaf:Person .
    ?y  a  foaf:Person .
}
WHERE {
    ?x  foaf:knows  ?y .
}
```

CONSTRUCT with optional data

```
CONSTRUCT {
    ?song dcterms:title ?title.
    ?song music:hasComposer ?composer.
    ?composer foaf:name ?cname.
    ?composer music:birthYear ?y.
}
WHERE {
    ?song dcterms:title ?title.
    OPTIONAL {
        ?song music:hasComposer ?composer.
        ?composer foaf:name ?cname.
        ?composer music:birthYear ?y.
    }
}
```

To learn more ...

SPARQL 1.1 Query Language

W3C Recommendation 21 March 2013

http://www.w3.org/TR/sparql11-query/

Understanding the basics of knowledge modeling (ontology, OWL)

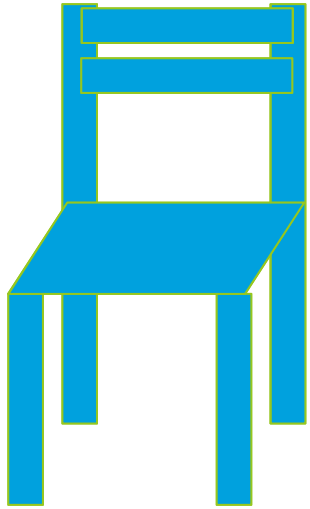
1. The **basic principles** of the Web of Data: from raw and heterogeneous Data to the lingua franca of the Web of Data
2. The **RDF** semantic and its main syntaxes
3. **Handling and querying** Data (triple store, end point, SPARQL)
4. Understanding the basics of **knowledge modeling** (ontology, OWL)
5. The **lifting and linking** of Data in order to publish 5 stars Data (DataLift).

What can we do with an ontology?

- Understand
- Capture
- Model
- Transform
- Integrate contextualized knowledge
- Do
- Share
- Reuse

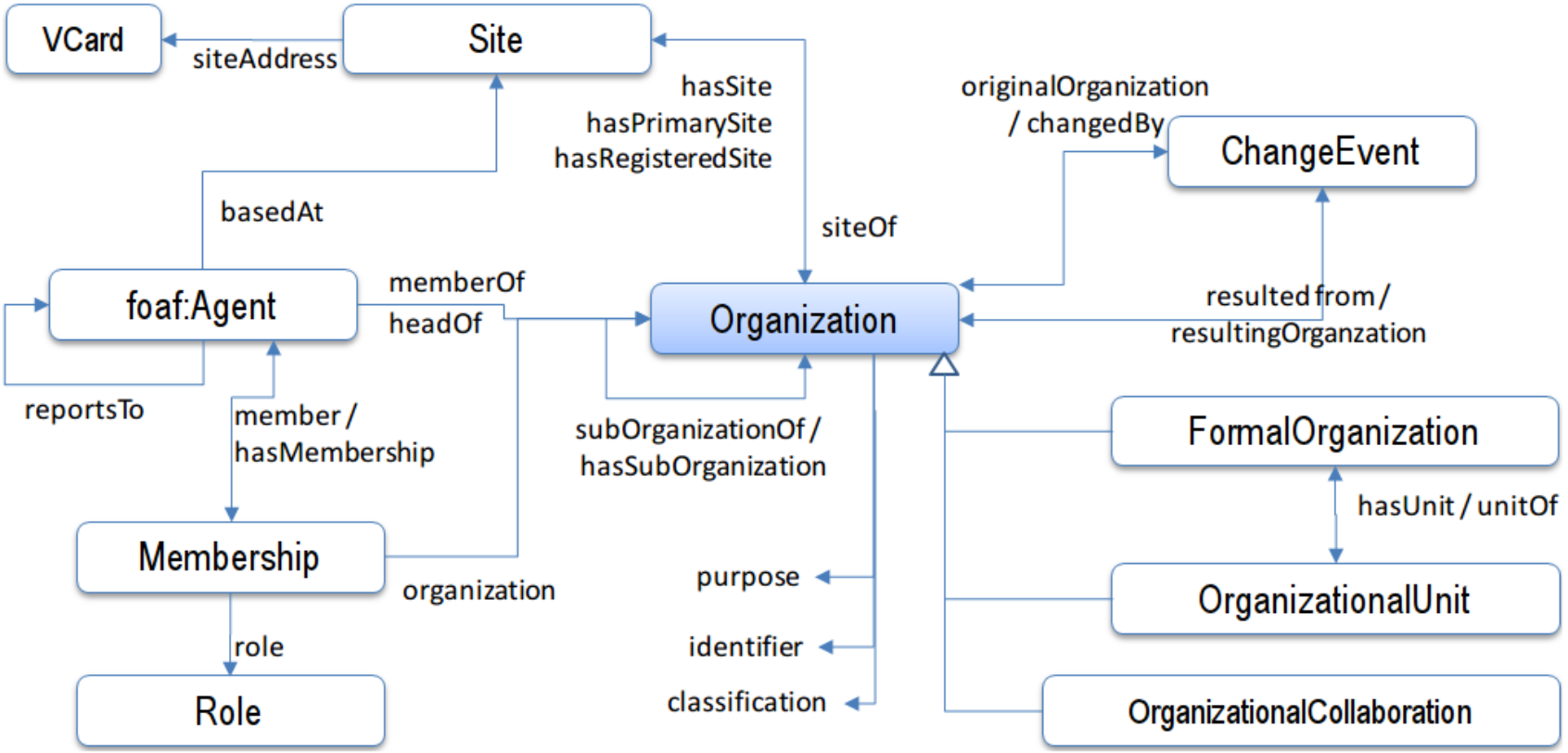
- Ontologies provide a common vocabulary
  - The meaning of the classes and properties is well defined, it allows us to "understand" the data that we exchange.
- Ontologies contain general knowledge, valid for all instances of data
  - they allow us to deduce new facts from the available data

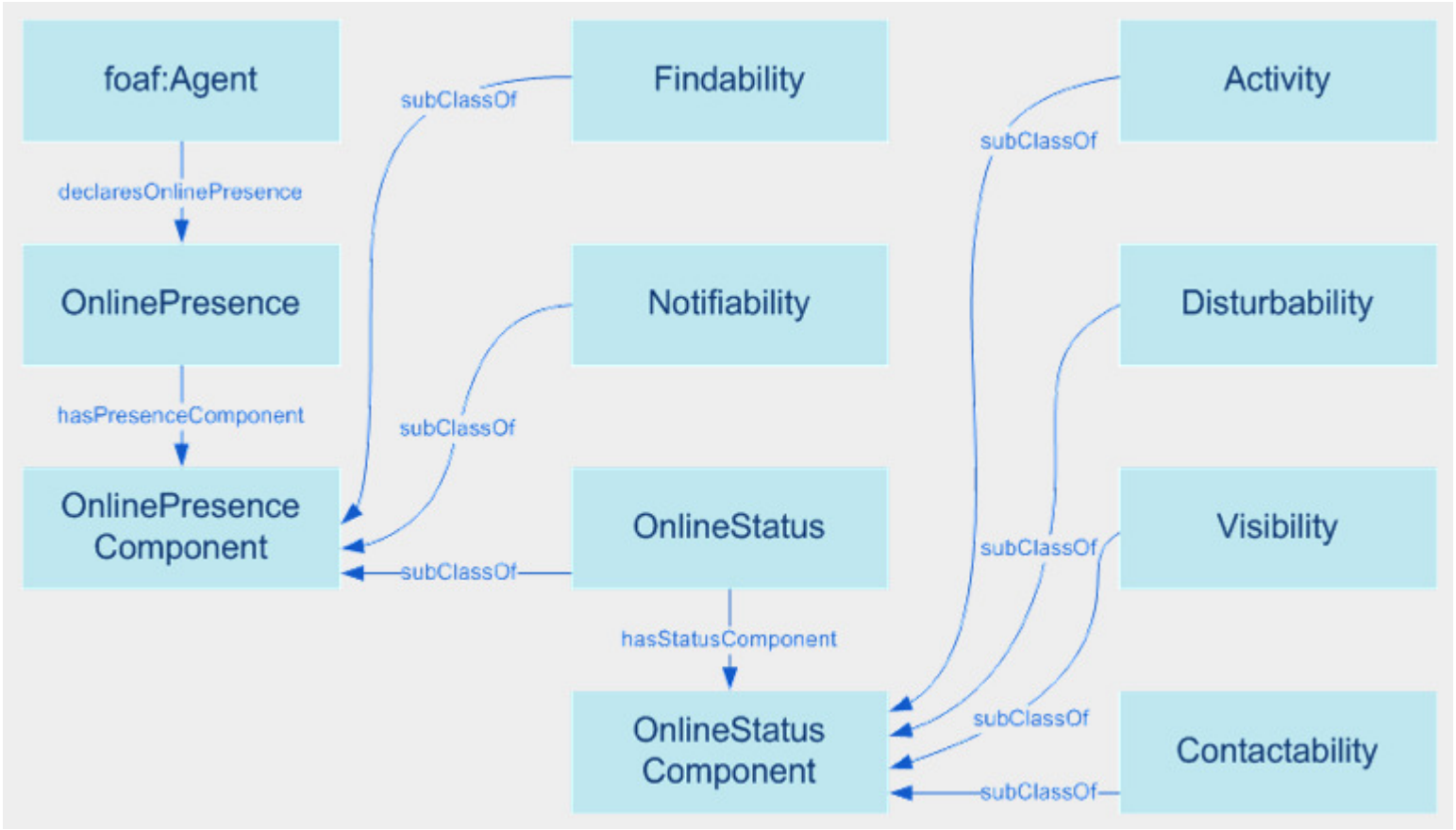
When is it a chair?

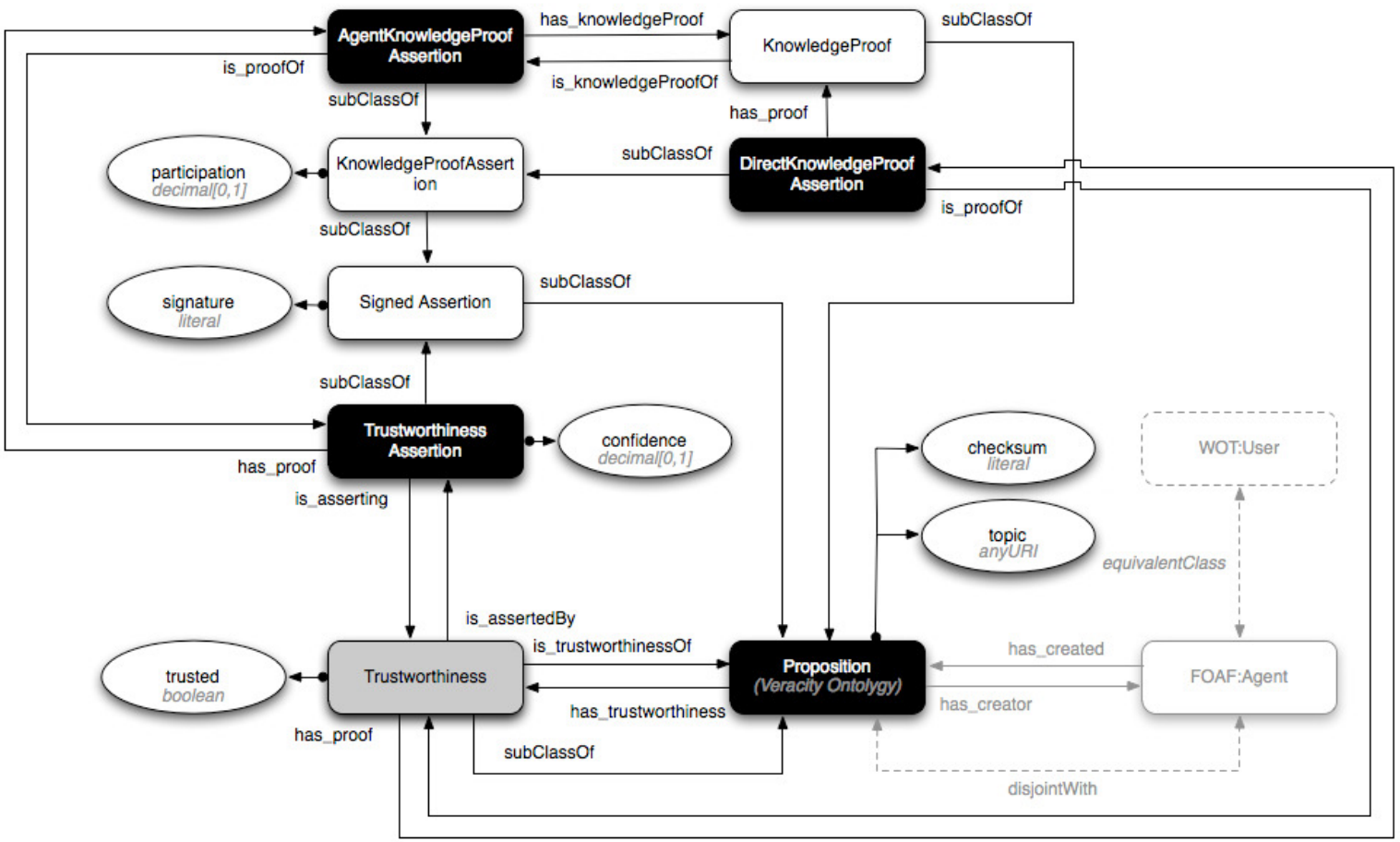


Ontology

Ontology samples


VCard
siteAddress
Site
hasSite
hasPrimarySite
hasRegisteredSite
originalOrganization
/ changedBy
ChangeEvent
basedAt
siteOf
foaf:Agent
memberOf
headOf
Organization
resulted from /
resultingOrganzation
reportsTo
member /
hasMembership
subOrganizationOf /
hasSubOrganization
FormalOrganization
hasUnit / unitOf
Membership
organization
purpose
OrganizationalUnit
role
identifier
classification
Role
OrganizationalCollaboration


foaf:Agent
declaresOnlinePresence
OnlinePresence
hasPresenceComponent
OnlinePresence Component
Findability
subClassOf
Notifiability
subClassOf
OnlineStatus
subClassOf
hasStatusComponent
OnlineStatus Component
Activity
subClassOf
Disturbability
subClassOf
Visibility
subClassOf
Contactability
subClassOf

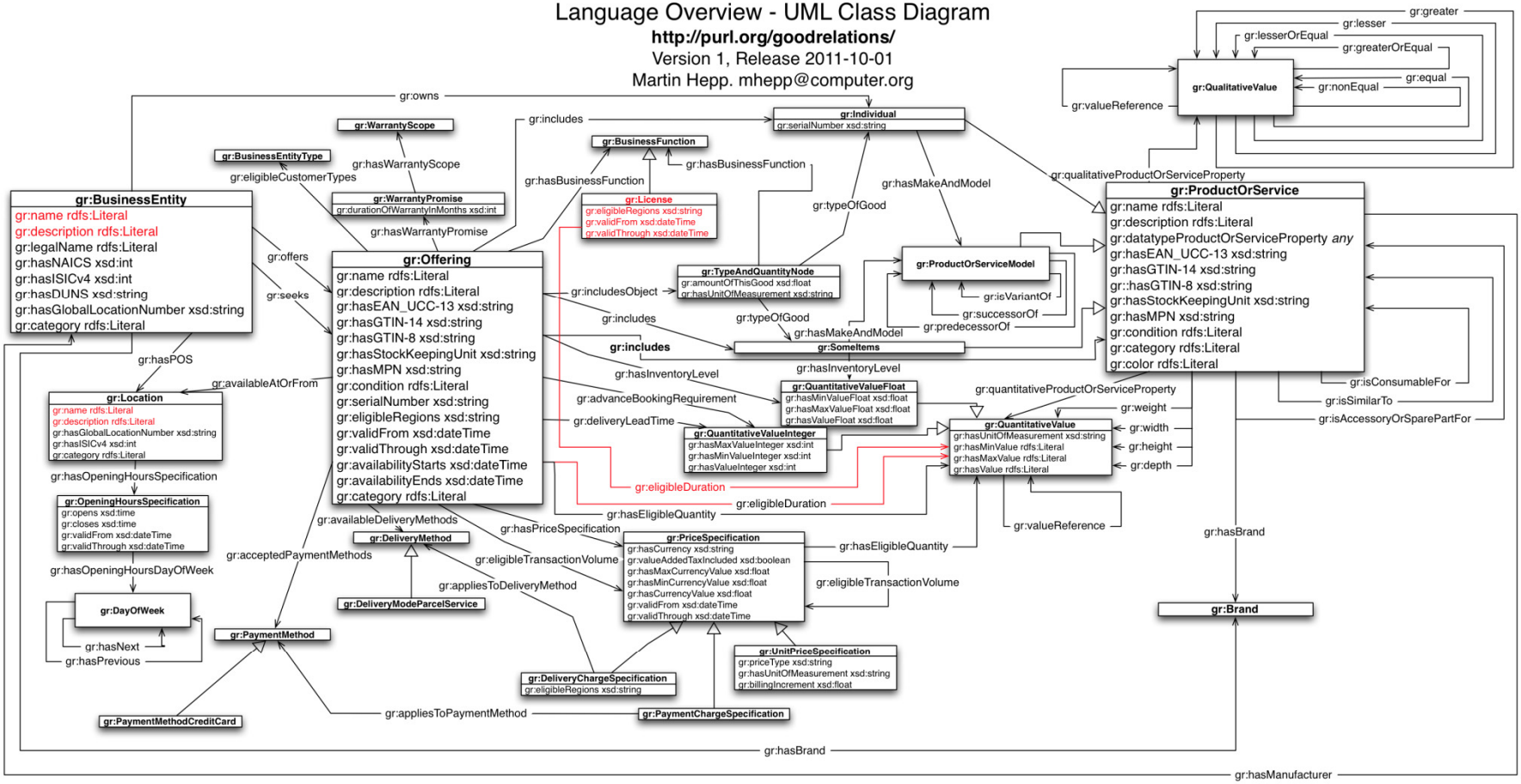

AgentKnowledgeProof Assertion
has_knowledgeProof
KnowledgeProof
subClassOf
is_proofOf
is_knowledgeProofOf
subClassOf
has_proof
participation
decimal[0,1]
KnowledgeProofAssert ion
subClassOf
DirectKnowledgeProof Assertion
is_proofOf
subClassOf
signature
literal
Signed Assertion
subClassOf
subClassOf
Trustworthiness Assertion
confidence
decimal[0,1]
has_proof
is_asserting
checksum
literal
WOT:User
topic
anyURI
equivalentClass
is_assertedBy
is_trustworthinessOf
trusted
boolean
Trustworthiness
Proposition
(Veracity Ontolygy)
has_created
FOAF:Agent
has_trustworthiness
has_creator
has_proof
subClassOf
disjointWith

The GoodRelations Ontology for E-Commerce
Language Overview - UML Class Diagram

http://purl.org/goodrelations/

Version 1, Release 2011-10-01

Martin Hepp. mhepp@computer.org



Notes:

1. The following GoodRelations elements are only shortcuts for simpler annotation or querying. See the documentation at http://purl.org/goodrelations/ for details:
   gr:hasValue (shortcut for setting both hasMinValue and hasMaxValue properties to the same value in one turn)
   gr:hasValueFloat (shortcut for setting both hasMinValueFloat and hasMaxValueFloat properties to the same value in one turn)
   gr:hasValueInteger (shortcut for setting both hasMinValueInteger and hasMaxValueInteger properties to the same value in one turn)
2. The following elements are now deprecated, but you can still use them, e.g. for staying compatible with older data consumers (e.g. Yahoo SearchMonkey):
   gr:ActualProductOrServiceInstance (now gr:Individual)
   gr:ProductOrServicesSomeInstancesPlaceholder (now gr:SomeItems)
   gr:LocationOfSalesOrServiceProvisioning (now gr:Location
3. For the recommended cardinality of attributes, see the GoodRelations Language Reference at http://purl.org/goodrelations/ v1.html.
4. gr:valueReference links can also exist between a gr:QualitativeValue and a gr:QuantitativeValue and vice versa, but this rare case is not shown for readability.
5. gr:name and gr:description can now be attached to any GoodRelations type, but this is not shown here for readability.

Red highlighting indicates elements added or changed in this release.

GoodRelations
The Web Vocabulary for E-Commerce
http://purl.org/goodrelations/

Fragments from the Wine Ontology example in the OWL Guide…

Namespaces

```
<rdf:RDF xmlns =http://www.example.org/wine#
  xmlns:vin =http://www.example.org/wine#
  xmlns:food=http://www.example.org/food#
  xmlns:owl ="http://www.w3.org/2002/07/owl#"
  xmlns:rdf =http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:rdfs=http://www.w3.org/2000/01/rdf-schema#
  xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#">
```

Ontology header

```
<owl:Ontology rdf:about="http://www.example.org/wine">
  <rdfs:comment>An example OWL ontology</rdfs:comment>
  <owl:priorVersion
    rdf:resource="http://www.example.org/wine-2102.owl"/>
  <owl:imports rdf:resource="http://www.example.org/food.owl"/>
  <rdfs:label>Wine Ontology</rdfs:label>
  ...
</owl:Ontology>
```

Simple classes

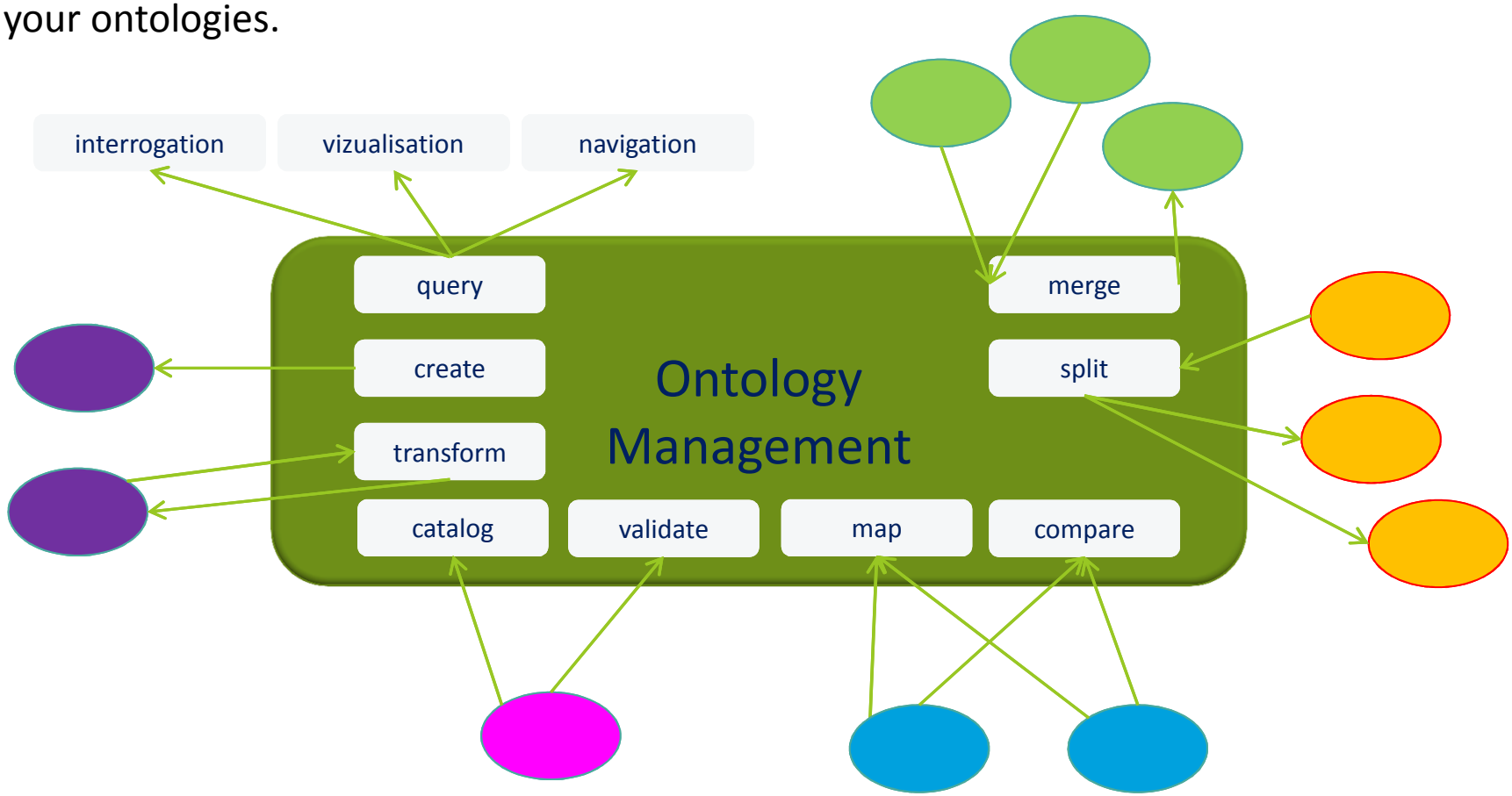
```
<owl:Class rdf:ID="Winery"/>
<owl:Class rdf:ID="Region"/>
<owl:Class rdf:ID="ConsumableThing"/>

<owl:Class rdf:ID="Wine">
  <rdfs:subClassOf rdf:resource="&food;PotableLiquid"/>
  <rdfs:label xml:lang="en">wine</rdfs:label>
  <rdfs:label xml:lang="fr">vin</rdfs:label>
   ...
</owl:Class>
```

Some tools

A small panorama

Ontology Management

- Supporting the ontology data model are a number of processing technologies to manage your ontologies.



Ontology creation

properties

A person **knows** other persons

logic

« is a brother of » is a symetrical property

classes (a value of a property)

A person is a **living been**

Ontology creation is not a uniforme question…

- Tow approches
  - Bottom – Up
  - Top – Down
- Different worlds
  - Unstructured data
  - Semi structured data
  - Strucutured data
- Knowledges at different levels
  - Task, sensor, …
  - Application, system, …
  - Domain, ecosystem, …
  - World, univers, …

Ontology Creation

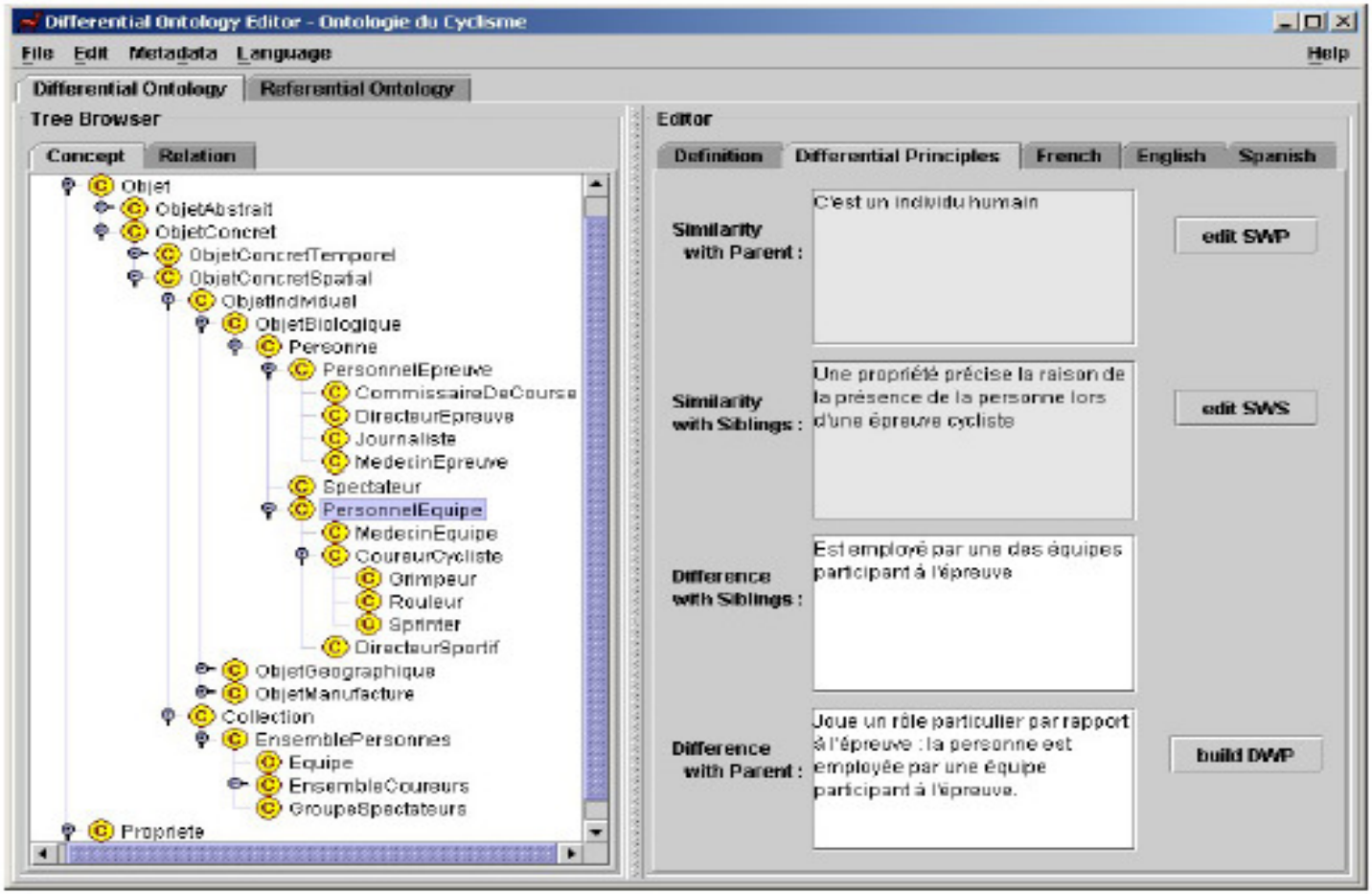
- **From texts**
  - **Terminae (2002)**
    - Manual Conceptualisation
  - **Text2Onto (2005)**
    - Automatic Conceptualisation
  - **OntoGen (2006)**
    - semi-automatic Conceptualisation
  - **Etc.**
- **From data**
  - Generalisation from data analyse
- **From structures**
  - Transformation

DOE : Differential Ontology Editor

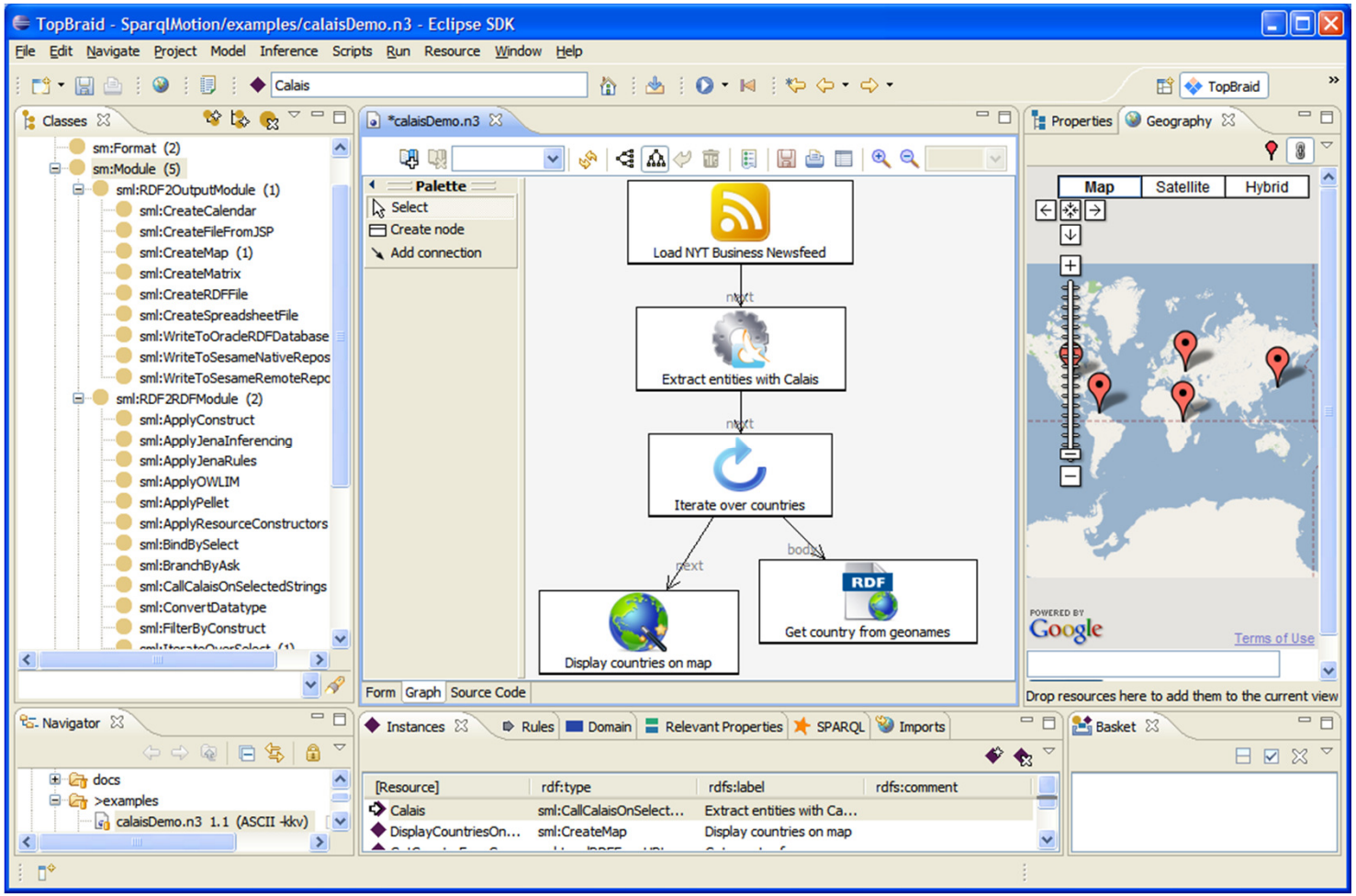
- **DOE** is a simple ontology editor which allows the user to build ontologies according to the methodology proposed by *Bruno Bachimont*.

- The specification process is divided in 3 steps.
  1. The user is invited to build taxonomies of concepts and relations, explicitly justifying the position of each item (notion) in the hierarchy. For each notion, the user builds a definition following 4 principles which come from the *Differential Semantics* theory. Hence, the user has to explicit why a notion is similar but more specific than its parent (2 principles), and why this notion is similar but different from its siblings (2 others principles). The user can also add synonyms and encyclopedic definition in a few languages for all notions.
  2. The two taxonomies are considered from an extensional semantics point of view. The user can augment them with new entities (defined) or add constraints onto the domains of the relations.
  3. The ontology can be translated into a knowledge representation language, which allows to use it in an appropriate ontology-based system or to import it into another ontology-building tool to specify it further:

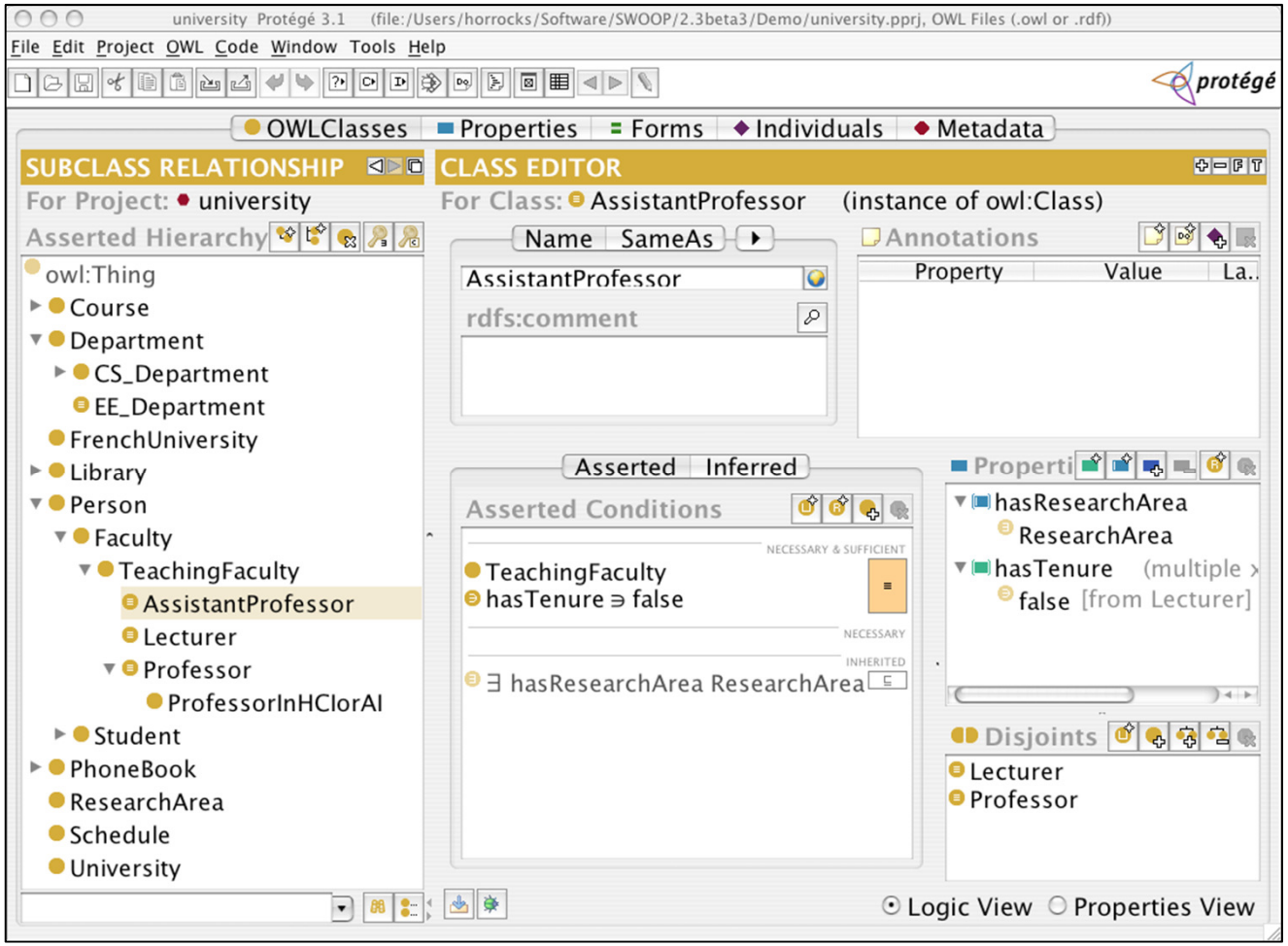
DOE : Differential Ontology Editor

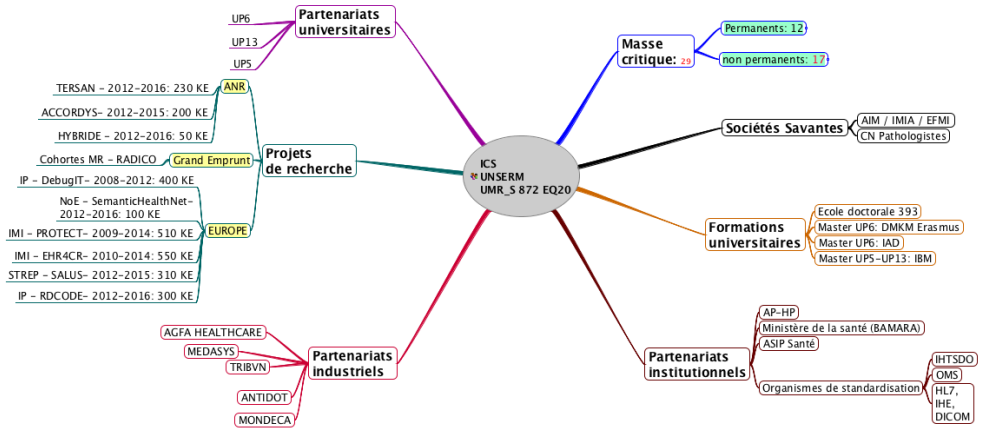


TopBraid composer

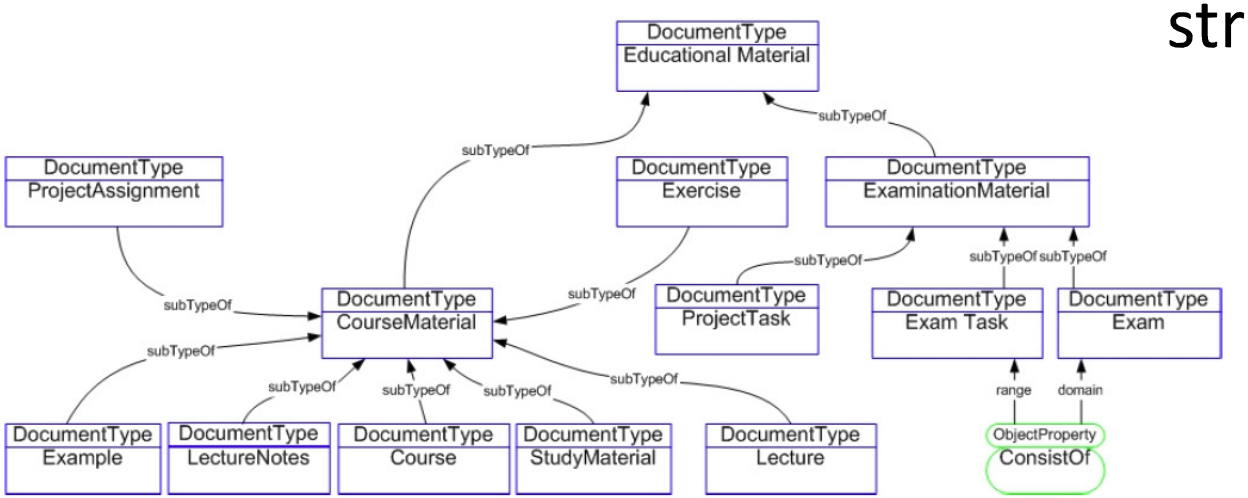


Protégé
An ontology editor





It helps to structure thinking



Oyster - Distributed Ontology Registry

Oyster is a distributed registry that exploits semantic web techniques in order to provide a solution for the management of ontology related metadata in distributed environments.

Swoogle
semantic web search 2007

ontology document term across ontologies

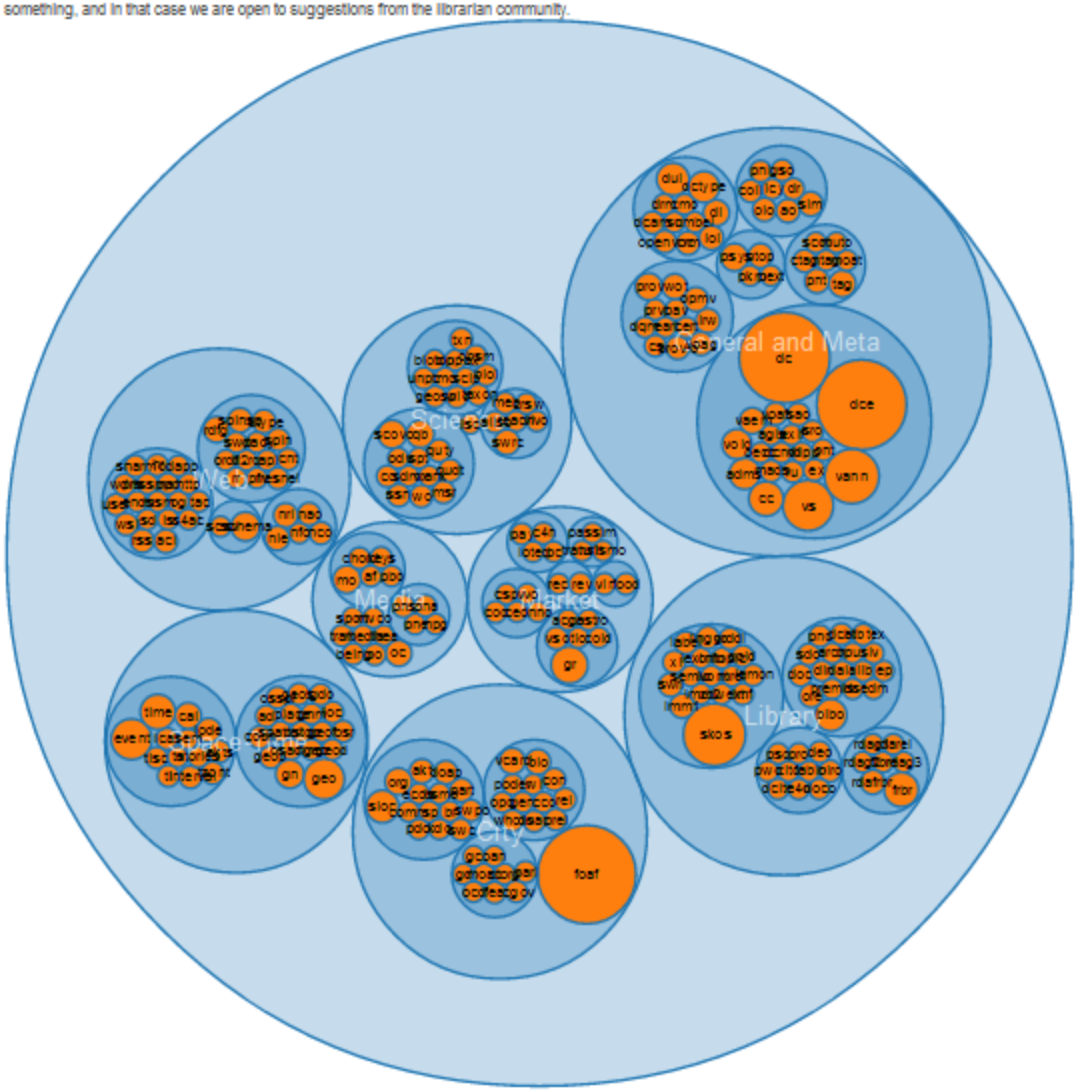
Swoogle Search

Searching over 10,000 ontologies

manual o news o faq o web-service o submit-url o sw-archive o feedback o swoogle2005

Swoogle © 2004-2007, ebiquity group at UMBC
This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License.

Catalog : LOV



DATALIFT

Catalog : LOV

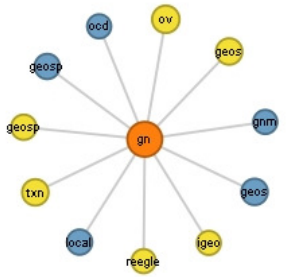
GN - The Geonames Ontology

Metadata:

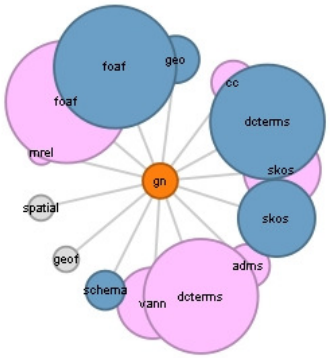
Property	Value
is part of vocabulary space	All > Where & When > Geography
Vocabulary URI	http://www.geonames.org/ontology
Prefix	gn
Namespace URI	http://www.geonames.org/ontology#
Description	The Geonames ontology provides elements of description for geographical features defined in the geonames.org data base @en
Date issued	2006-10-14
Last modified	2012-10-29
Language	Russian, Swedish, English, Bulgarian, Norwegian
Creator	Bernard Vatant
Publisher	Geonames.org
Class number	7
Property number	26
Homepage	http://www.geonames.org/ontology/documentation.html
See also	http://stats.lod2.eu/vocabularies/95
Represented by	format-geonames

Vocabulary links:

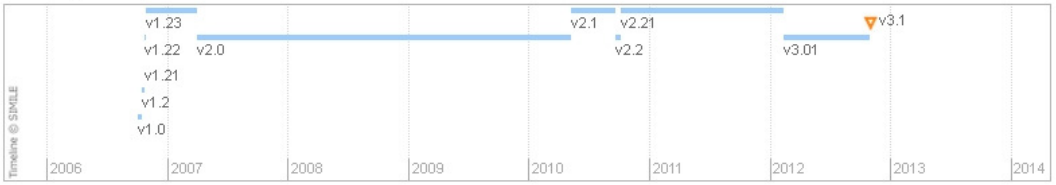
Vocabularies referencing "gn" (11)



Vocabularies referenced by "gn" (14)



Vocabulary history:



Languages to express ontologies

- Two major languages (W3C)
  - **RDFS**
    - extensible language for knowledge representation
    - provides basic elements for the definition of ontologies or vocabularies intended to structure RDF resources
    - V1 in March 1999
    - Final recommendation in February 2004
    - the main components of RDFS are included in OWL , a more expressive language
  - **OWL**
    - knowledge representation built on the RDF model
    - based on the description logic
    - designed as an extension of RDFS
    - V1 in feb. 2004
    - V2 in oct. 2009

Ontology alignment

What is interconnection?

Latitude

Paris **48°51′ 24″ N**

Longitude

Paris **2°21′ 07″ E**

Geographical data

Statistical data

Population

Paris **2.211.297**

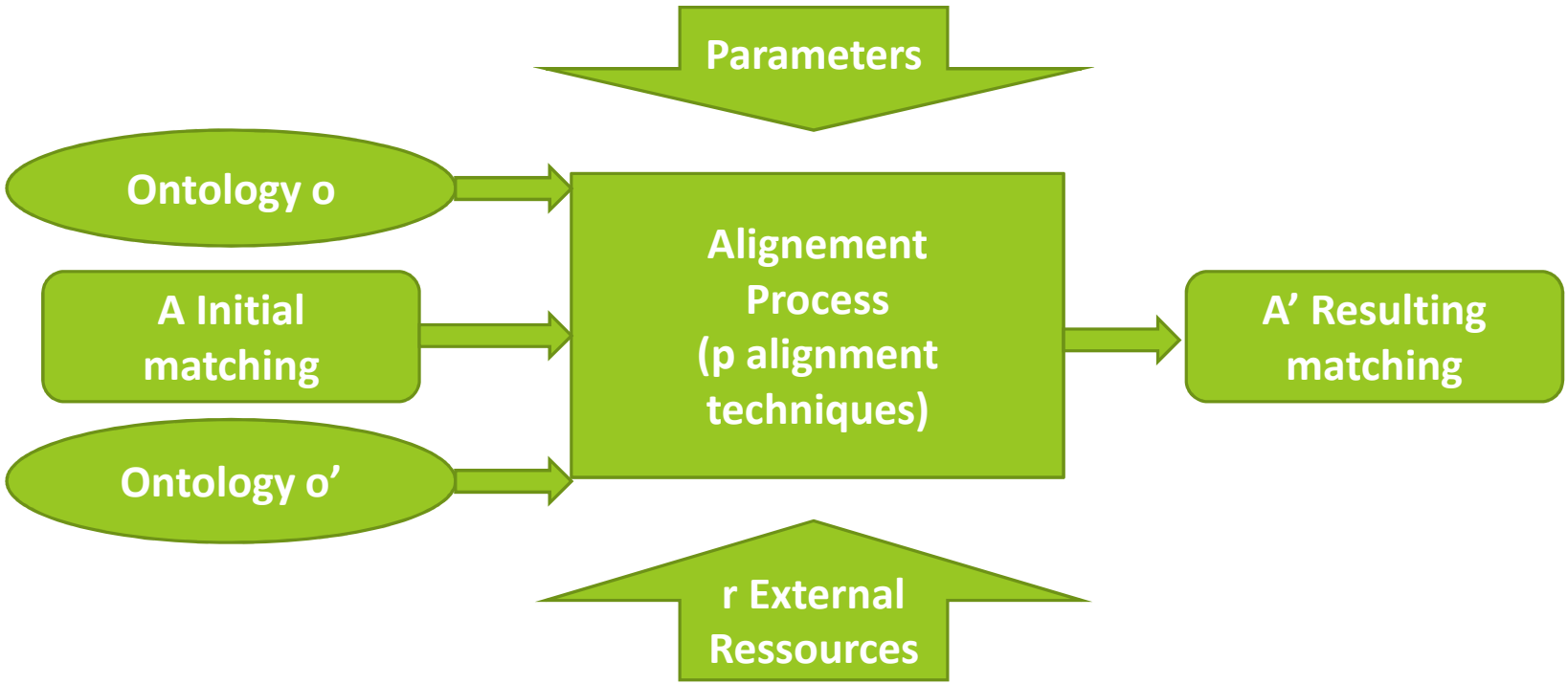
Mutualy enriched

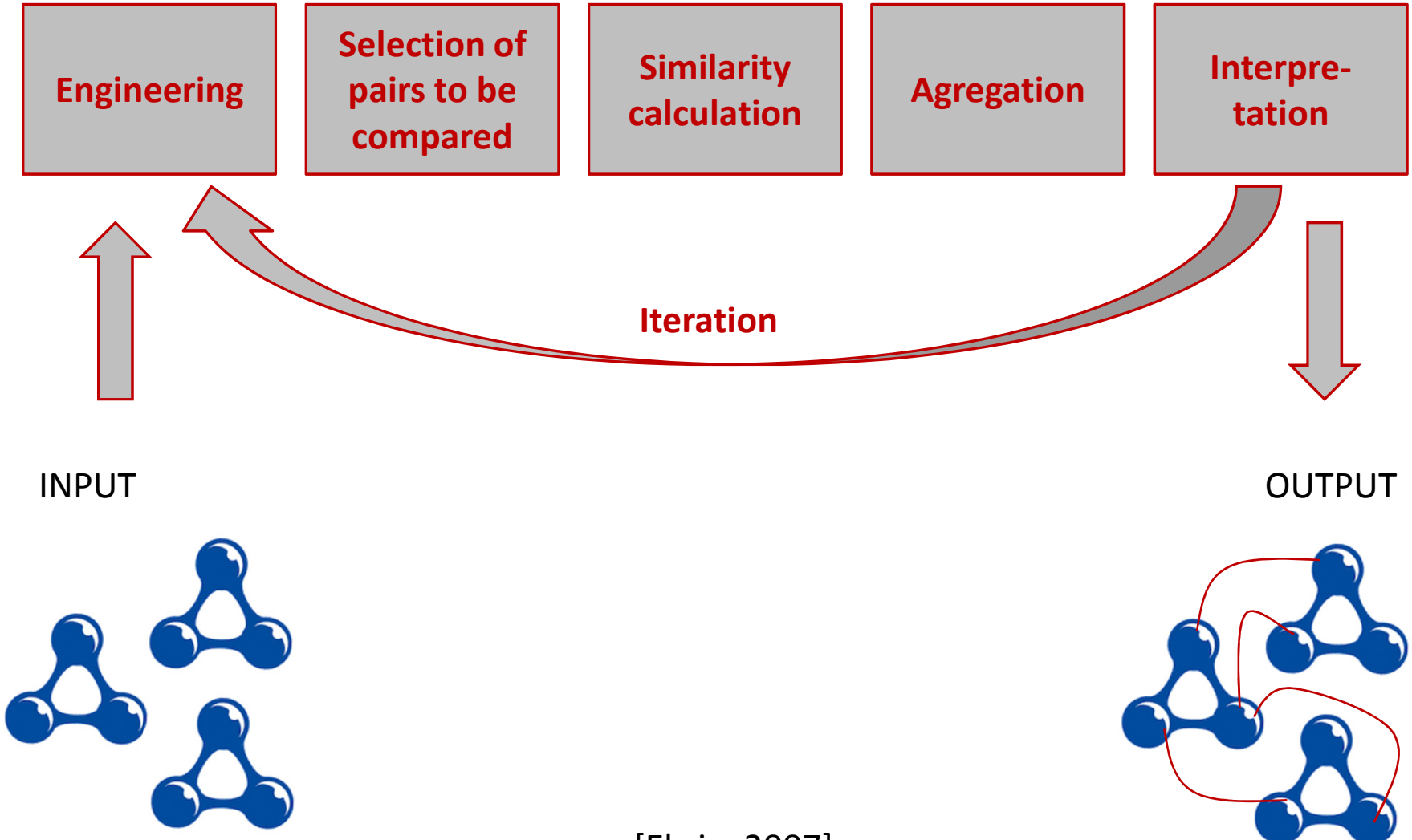
Ontology matching is a function f applied:

- to two ontologies o and o',
- to a set of original matching A,
- from a set of parameters applicable to p alignment techniques implemented in the process
- and a set of external resources r,

and that produces a set of matching A‘ between the two ontologies.
[Euzenat and Shvaiko, 2007]

$$A' = f(o; o'; A; p; r)$$




Engineering
Selection of pairs to be compared
Similarity calculation
Agregation
Interpre-tation
Iteration
INPUT
OUTPUT

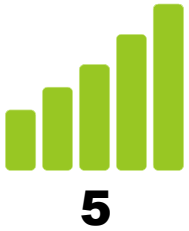
[Ehrig, 2007]

- Jaccard distance [Jaccard, 1901]
- n-grammes technique [Shannon, 1948]
- Levenshtein distance [Levenshtein, 1966]
  - minimal cost of edition operations necessary to transforme
- Lin mesure [Lin 1998]
- etc.

$$Sim_{Lin}(x,y) = 2\,\frac{\sum_{t\in(tri(x)\cap tri(y))} logP(t)}{\sum_{t\in tri(x)} logP(t) + \sum_{t\in tri(y)} logP(t)}$$

For a long time, tools were created.
SILK is an interesting implementation.

http://lod2.eu/Project/Silk.html



The lifting and linking of Data in order to publish 5 stars Data (DataLift)

1. The **basic principles** of the Web of Data: from raw and heterogeneous Data to the lingua franca of the Web of Data
2. The **RDF** semantic and its main syntaxes
3. **Handling and querying** Data (triple store, end point, SPARQL)
4. Understanding the basics of **knowledge modeling** (ontology, OWL)
5. The **lifting and linking** of Data in order to publish 5 stars Data (DataLift).

The 5 stars of TBL

2010: At the "Gov 2.0 Expo" Tim Berners-Lee presented a wide opening became famous data that refers

5 stars

★ Available on the web (whatever format) but with an open licence, to be **Open Data**

★★ Available as **machine-readable structured data** (e.g. excel instead of image scan of a table)

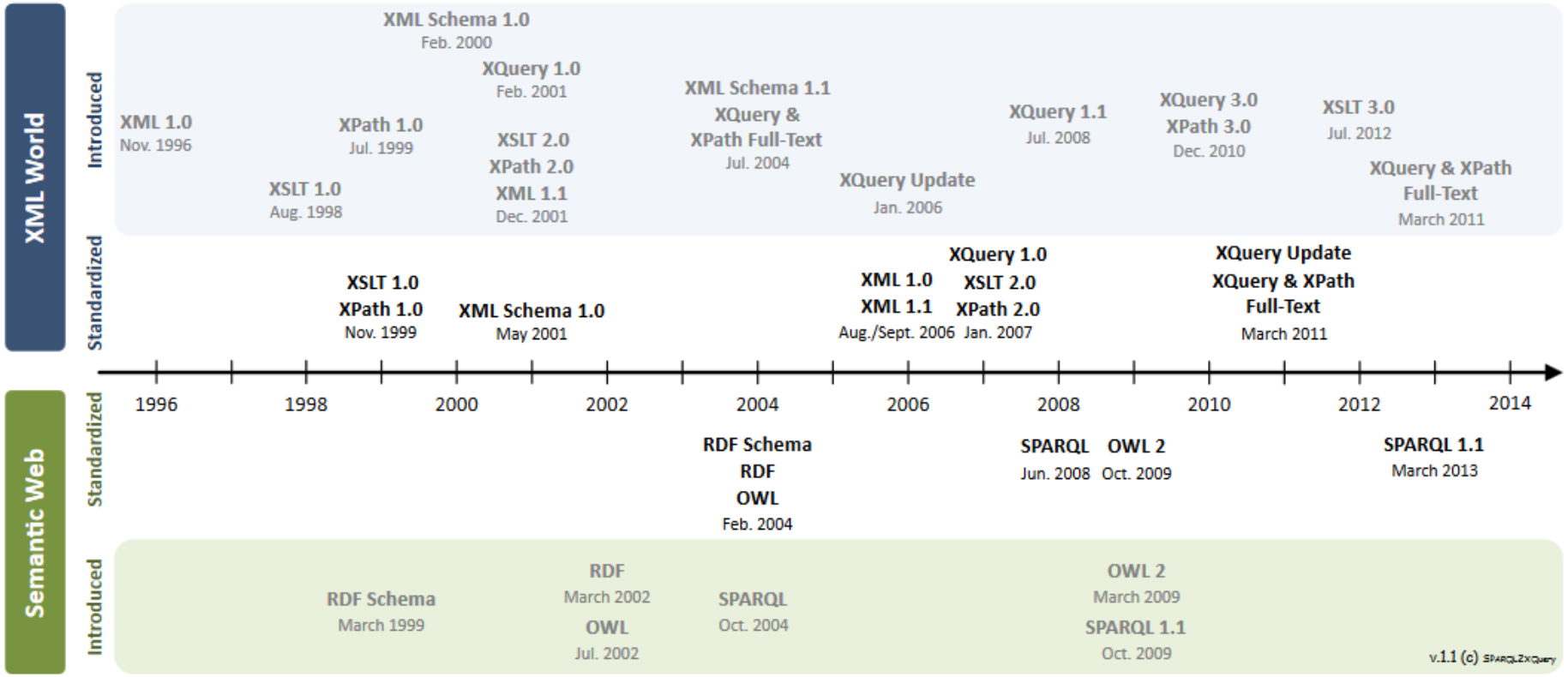
★★★ as (2) plus **non-proprietary format** (e.g. CSV instead of excel)

★★★★ All the above plus, Use **open standards from W3C** (RDF and SPARQL) to identify things, so that people can point at your stuff

★★★★★ All the above, plus: Link your data to other people's data to **provide context**

http://www.w3.org/DesignIssues/LinkedData.html

4 stars – Standards & norms



4 stars – shared vocabularies

Subject	predicate	object
ASIN:B00005014I	dc:type	enregistrement musical
ASIN:B00005014I	dc:format	DVD
ASIN:B00005014I	dc:title	‘Variations Goldberg’ @fr
ASIN:B00005014I	bibo:performer	Glenn Gould
ASIN:B00005014I	mo:composer	Johannn-Sebastian Bach

The predicates have a **declared semantic** in **shared vocabularies** accessible via their URI

dc:	Dublin Core Terms
bibo:	Bibliographic Ontology
mo:	Music Ontology

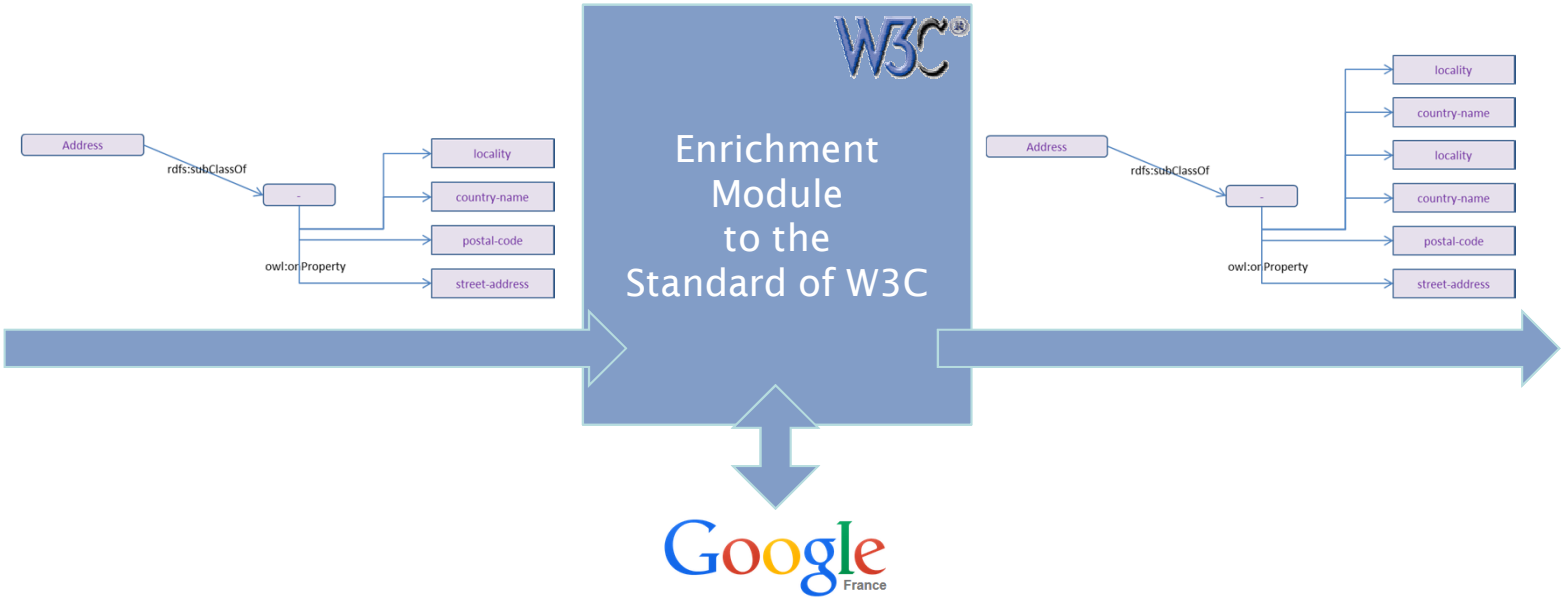
5 stars – shared referentials

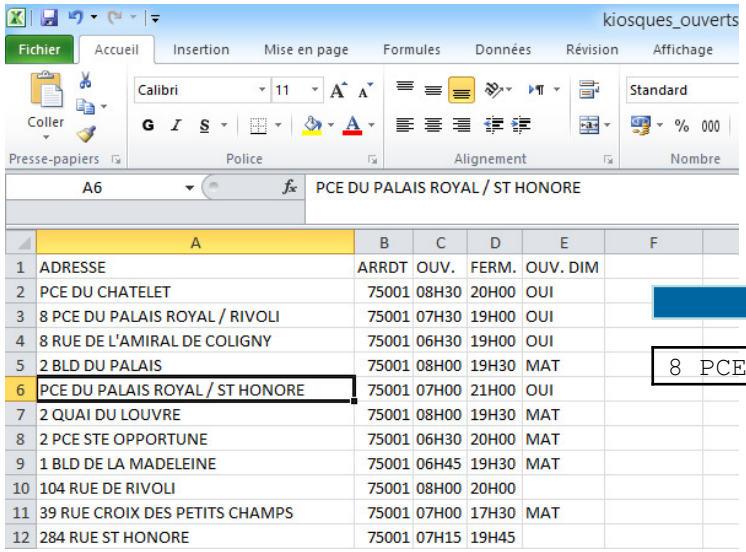
The description uses **entities of reference defined by URI**

subject	**predicate**	**object**
ASIN:B00005014I	dc:type	http://schema.org/MusicRecording
ASIN:B00005014I	dc:format	http://purl.org/spar/fabio/dvd
ASIN:B00005014I	bibo:performer	http://viaf.org/viaf/54148399/
ASIN:B00005014I	mo:composer	http://viaf.org/viaf/12304462/

The data structure does not change with respect to the original description

A demonstration





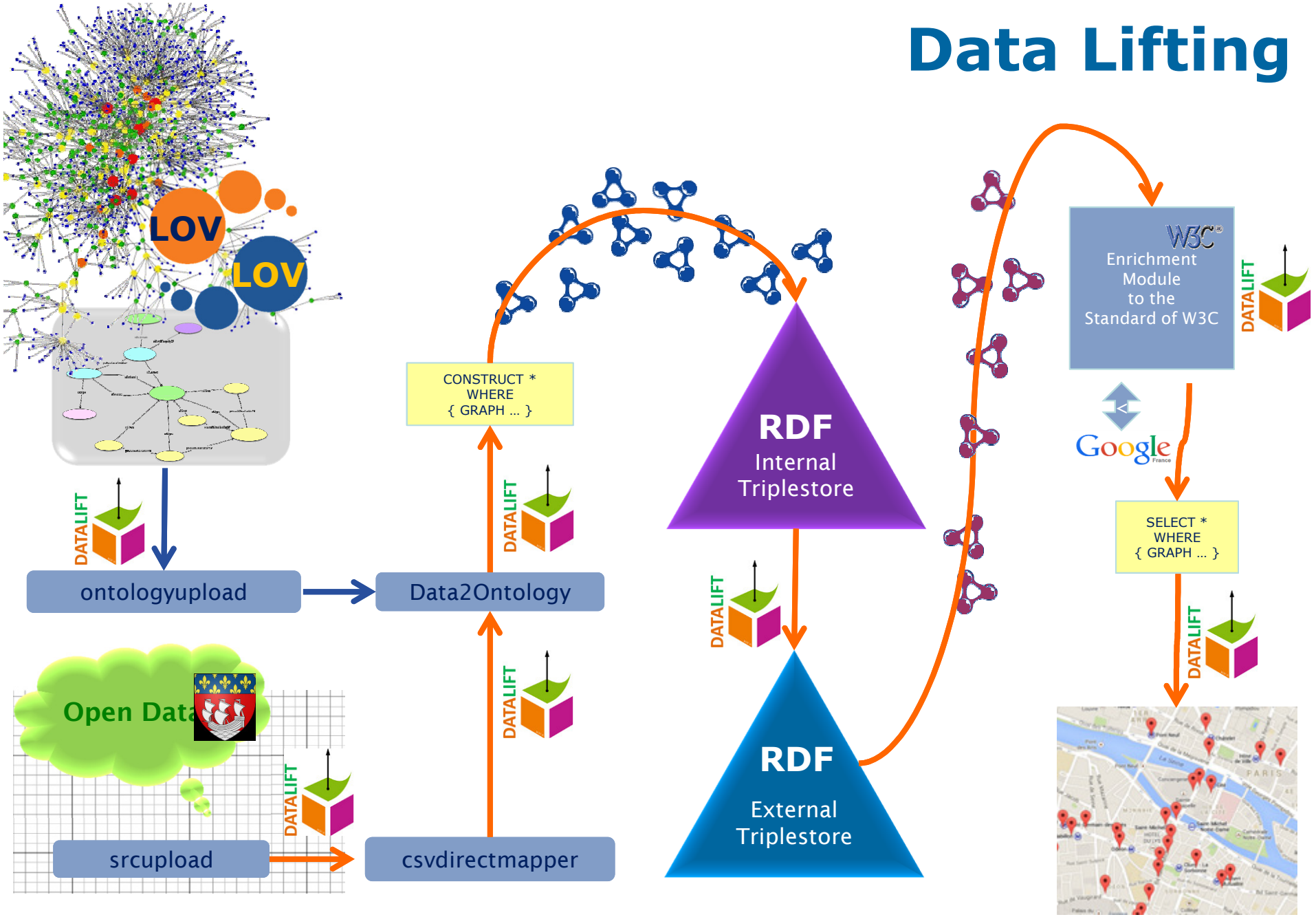
	A	B	C	D	E	F
1	ADRESSE	ARRDT	OUV.	FERM.	OUV. DIM	
2	PCE DU CHATELET	75001	08H30	20H00	OUI	
3	8 PCE DU PALAIS ROYAL / RIVOLI	75001	07H30	19H00	OUI	
4	8 RUE DE L'AMIRAL DE COLIGNY	75001	06H30	19H00	OUI	
5	2 BLD DU PALAIS	75001	08H00	19H30	MAT	
6	PCE DU PALAIS ROYAL / ST HONORE	75001	07H00	21H00	OUI	
7	2 QUAI DU LOUVRE	75001	08H00	19H30	MAT	
8	2 PCE STE OPPORTUNE	75001	06H30	20H00	MAT	
9	1 BLD DE LA MADELEINE	75001	06H45	19H30	MAT	
10	104 RUE DE RIVOLI	75001	08H00	20H00		
11	39 RUE CROIX DES PETITS CHAMPS	75001	07H00	17H30	MAT	
12	284 RUE ST HONORE	75001	07H15	19H45		

The data are « flat »

8 PCE DU PALAIS ROYAL / RIVOLI	75001	07H30	19H00	OUI

And incomplete for geolocalisation

Data Lifting



Using ***Datalift***, the all-in-one linked data production plateform

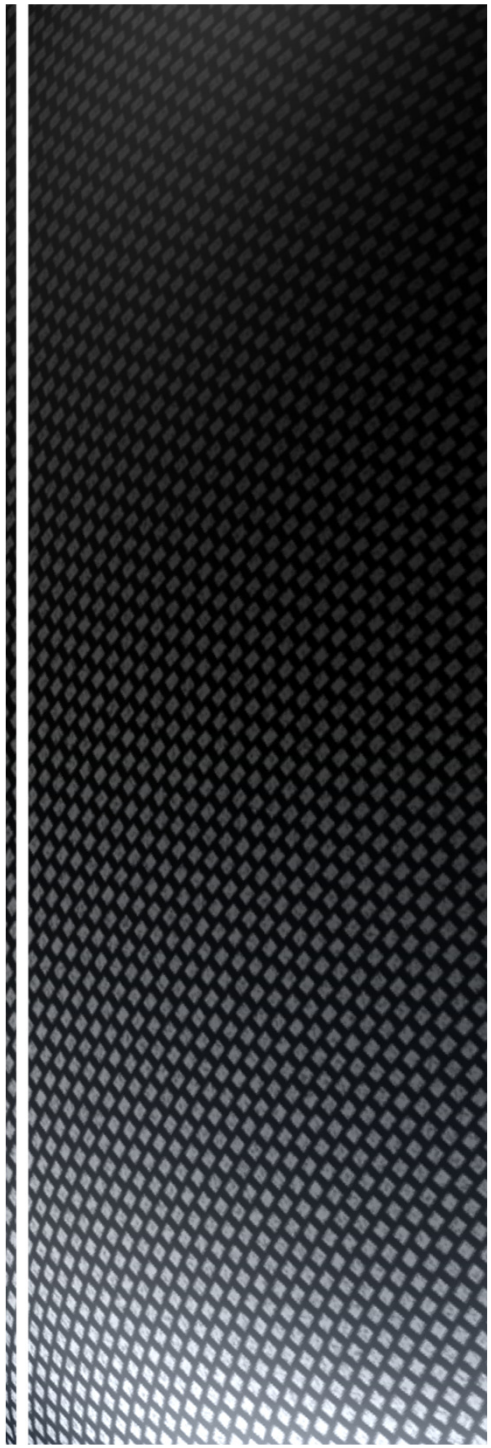
http://www.youtube.com/watch?v=dMvEPpUZJqE

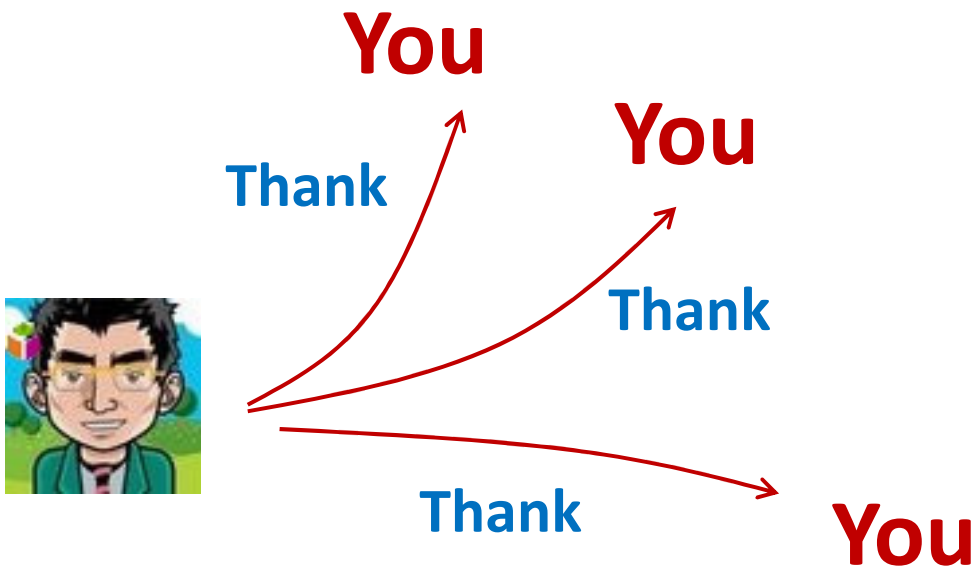
Demo on youtube

Datalift is an open source plateform

www.datalift.org

DataLift is also an association,
which is freely open to all students.





Gabriel KEPEKLIAN
R&D director
Atos Integration France

gabriel.kepeklian@atos.net

Atos