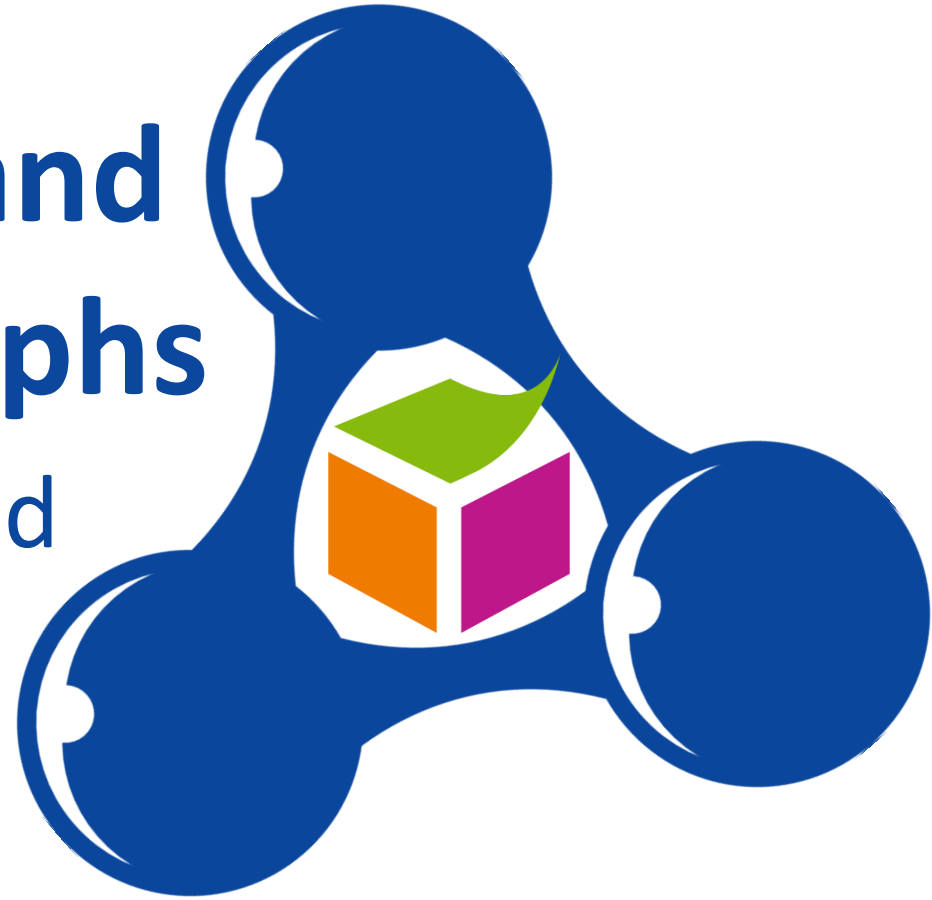


# Semantic Web and Linked Data Graphs

or how to link data and  
schemas on the web

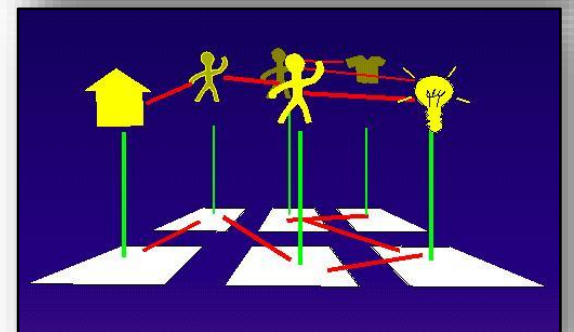
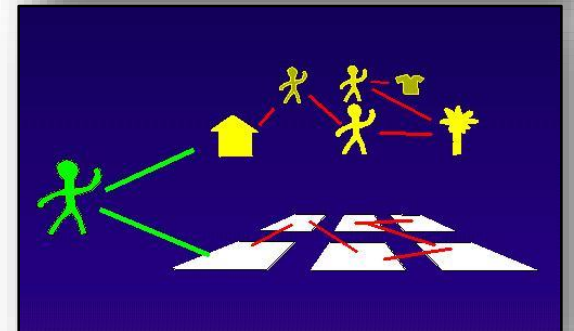
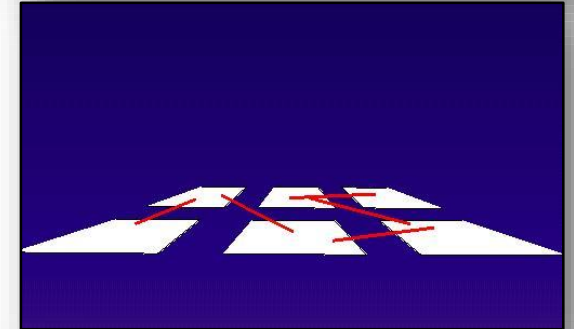
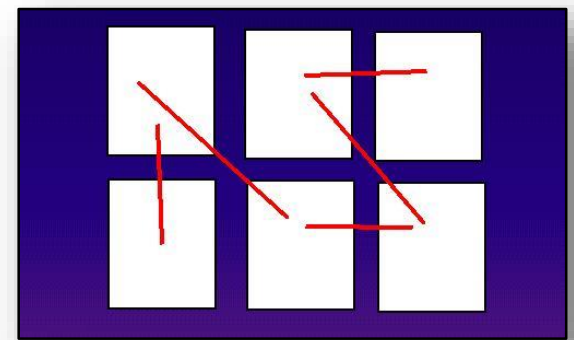


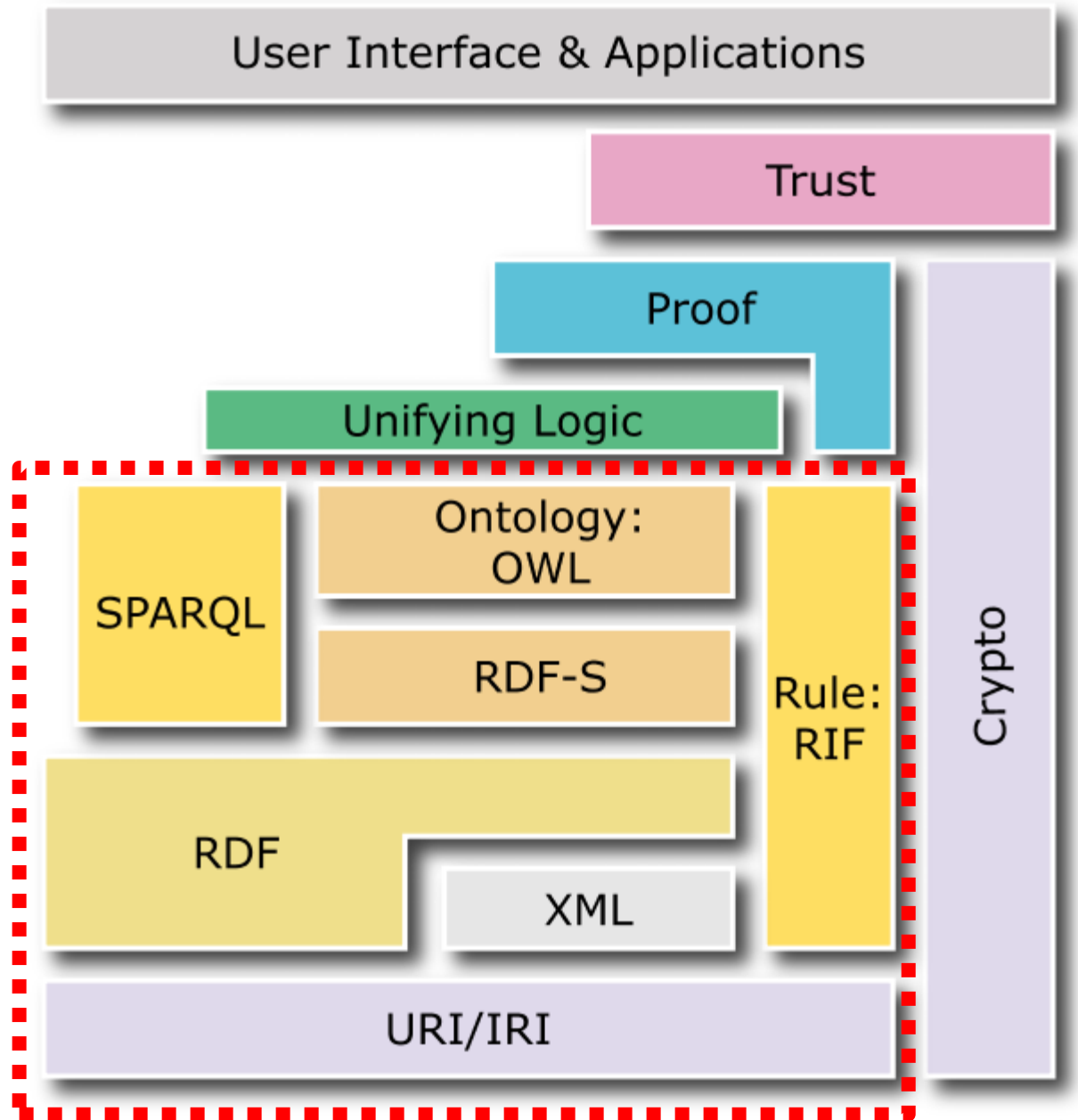
Fabien Gandon, <http://fabien.info>, @fabien\_gandon

# semantic web

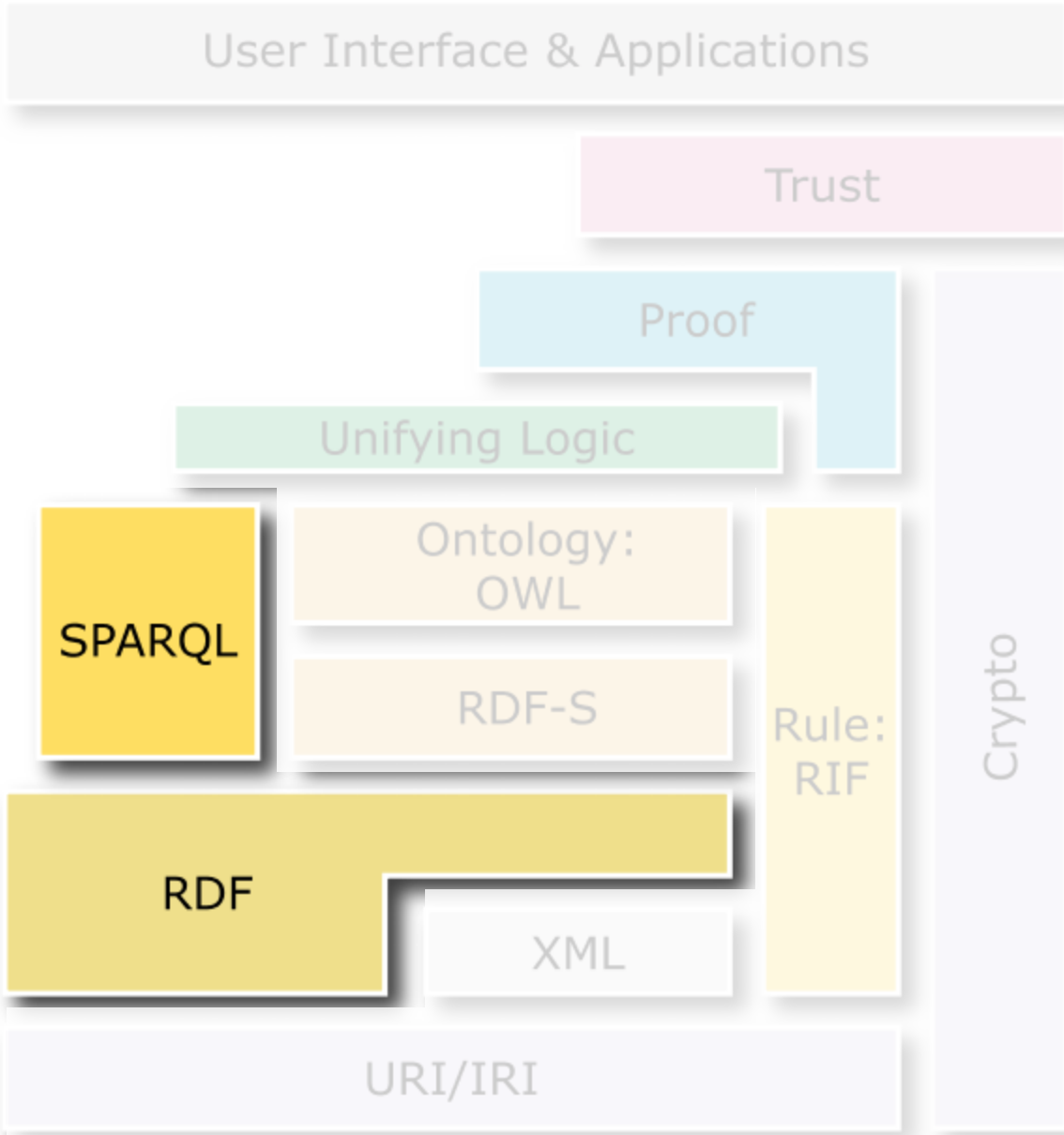
mentioned by Tim BL

in **1994** at WWW





**A WEB OF LINKED DATA**



**RDF** stands for

**Resource:** *pages, dogs, ideas...*

*everything that can have a URI*

**Description:** *attributes, features, and  
relations of the resources*

**Framework:** *model, languages and  
syntaxes for these descriptions*

**R****D****F** is a triple model *i.e.* every piece of knowledge is broken down into

( **s**ubject , **p**redicate , **o**bject )



***doc.html has for author Fabien  
and has for theme Music***

**doc.html** has for author **Fabien**  
**doc.html** has for theme **Music**

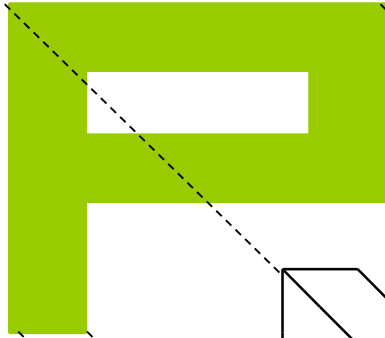


( **doc.html** , **author** , **Fabien** )

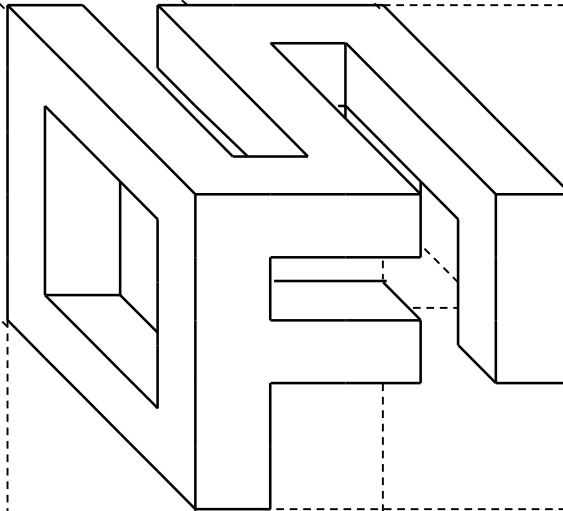
( **doc.html** , **theme** , **Music** )

( **subject** , **predicate** , **object** )

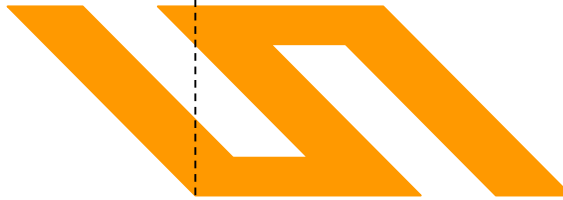
**Predicate**



**Object**



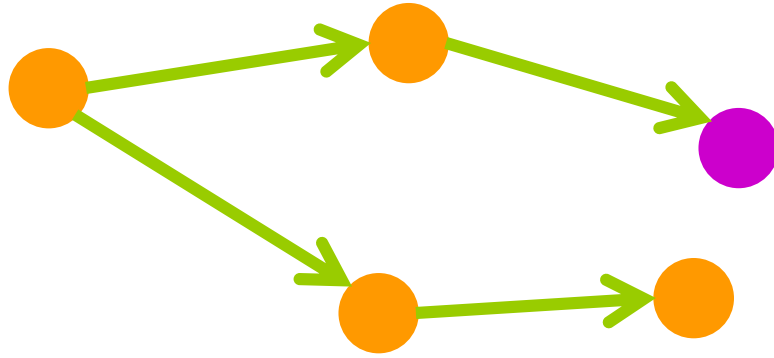
**Subject**



**a triple**  
the RDF atom



**RDF** is also a graph model  
to link the **descriptions** of resources



**RDF** triples can be seen as arcs  
of a graph (**vertex**, **edge**, **vertex**)

( doc.html , author , Fabien )  
( doc.html , theme , Music )

**Fabien**

**author**

**doc.html**

**theme**

**Music**



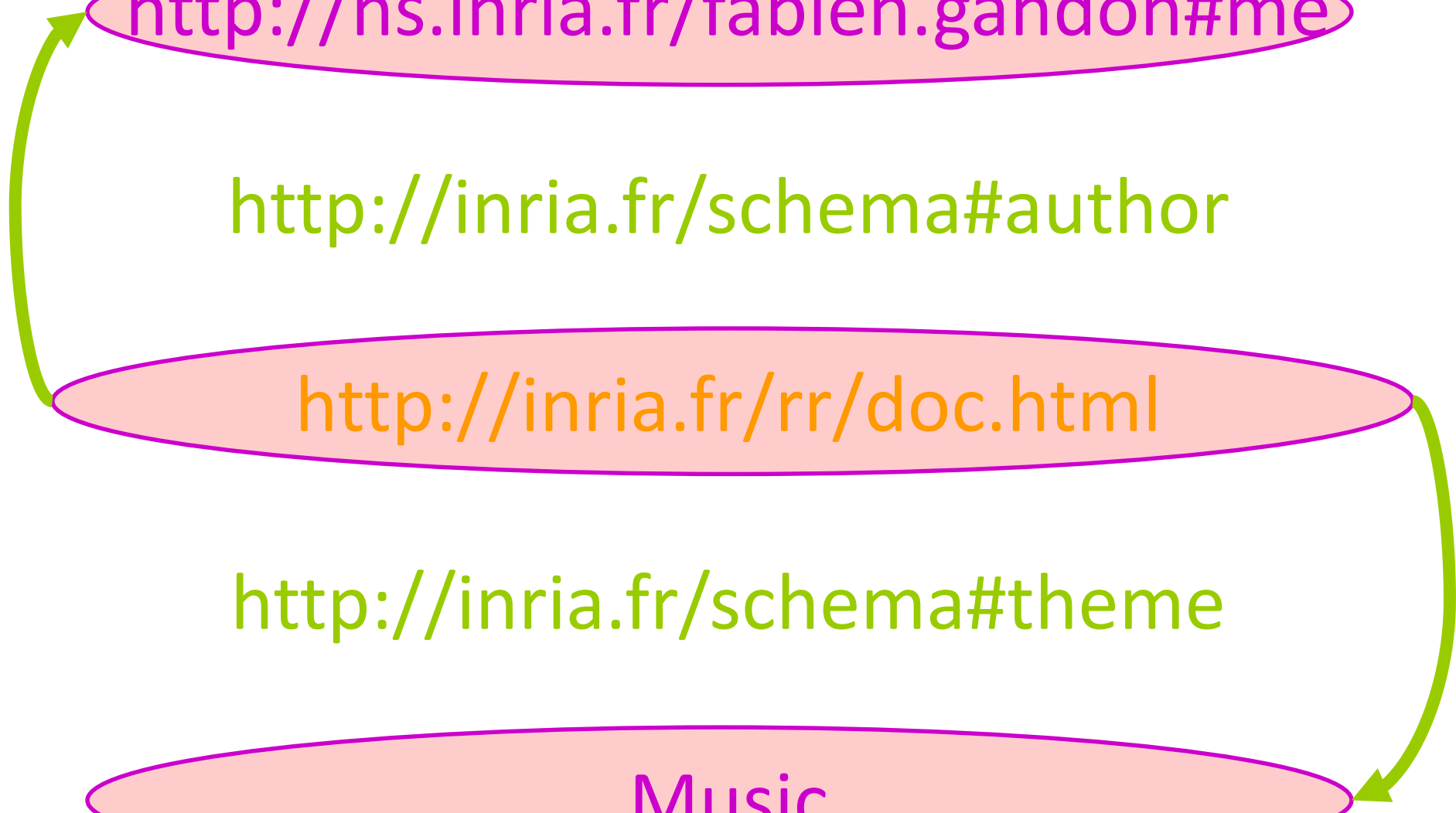
<http://ns.inria.fr/fabien.gandon#me>

<http://inria.fr/schema#author>

<http://inria.fr/rr/doc.html>

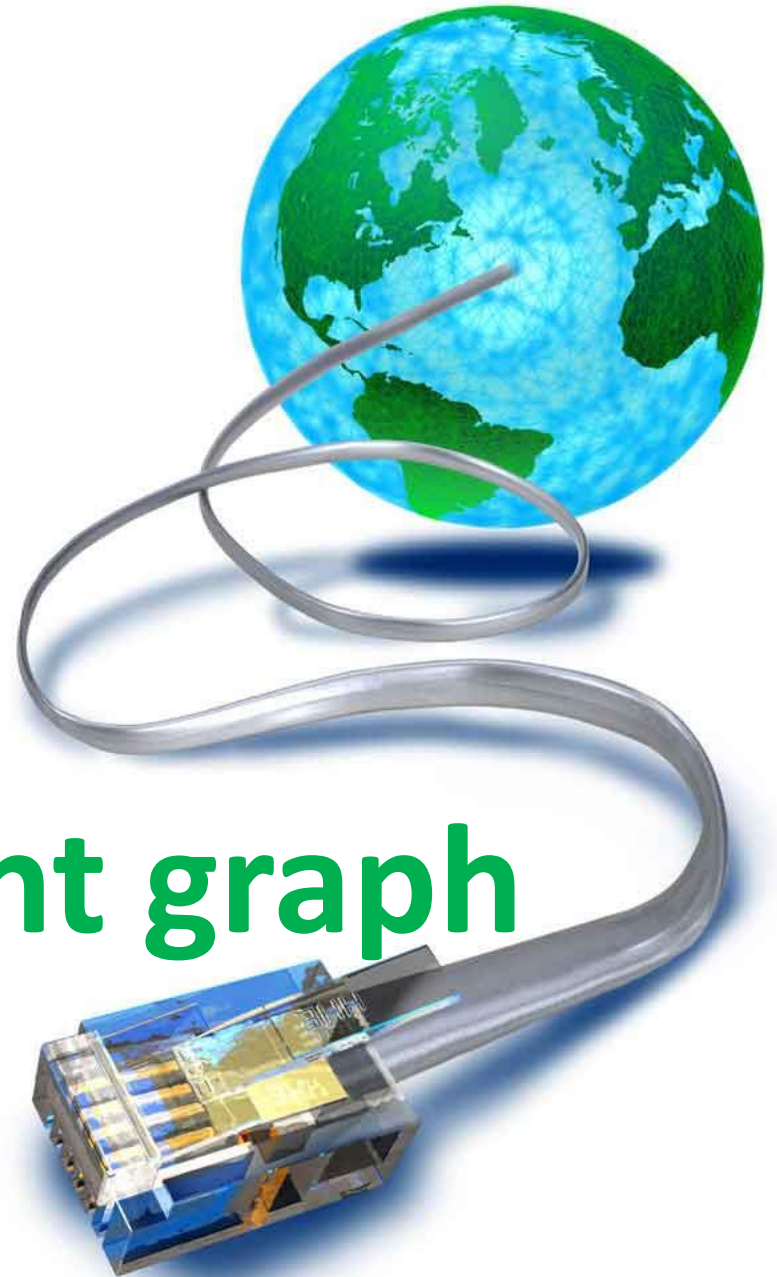
<http://inria.fr/schema#theme>

Music



open and link data in a

**global giant graph**





in **RDF** values of properties can also be literals i.e. strings of characters

( doc.html , author , Fabien )  
( doc.html , theme , "Music" )

<http://ns.inria.fr/fabien.gandon#me>

<http://inria.fr/schema#author>

<http://inria.fr/rr/doc.html>

<http://inria.fr/schema#theme>

"Music"

< RDF /> has an XML syntax

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-
  rdf-syntax-ns#"
  xmlns:inria="http://inria.fr/schema#" >

<rdf:Description
  rdf:about="http://inria.fr/rr/doc.html">
  <inria:author rdf:resource=
    "http://ns.inria.fr/fabien.gandon#me"/>
  <inria:theme>Music</inria:theme>
</rdf:Description>

</rdf:RDF>
```

**R****D****F** has other syntaxes  
(Turtle, JSON, Triple)

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

<http://inria.fr/rr/doc.html>
  inria:author
    <http://ns.inria.fr/fabien.gandon#me> ;
  inria:theme "Music" .
```

# writing rules for RDF triples

- the subject is always a resource (never a *literal*)
- properties are binary relations and their types are identified by URIs
- the value is a resource or a literal





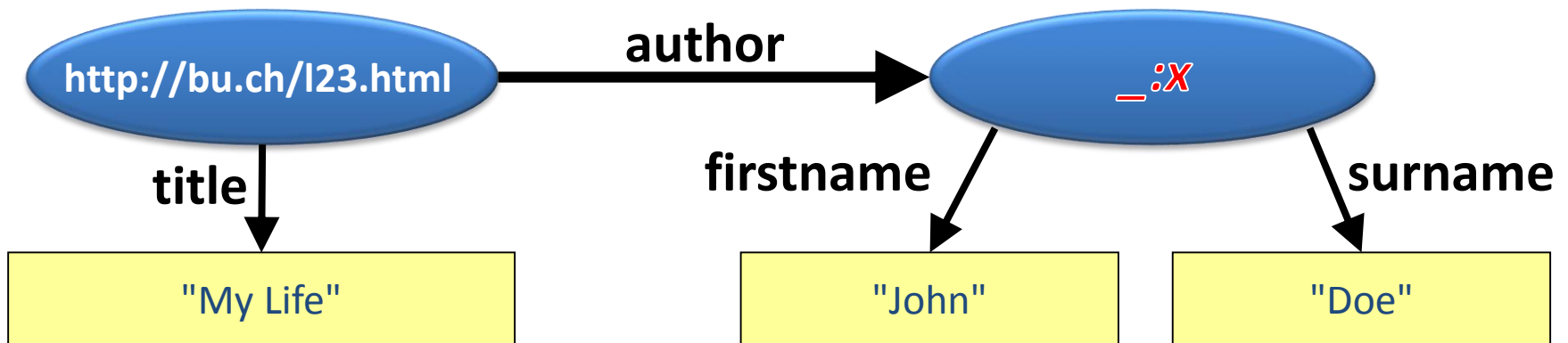
# blank nodes (bnodes)

handy anonymous nodes (existential quantification)

there exist a resource such that...  $\{ \exists r ; \dots \}$

```
<rdf:Description rdf:about="http://bu.ch/123.html" >
  <author>
    <rdf:Description>
      <surname>Doe</surname>
      <firstname>John</firstname>
    </rdf:Description>
  </author>
  <title>My Life</title>
</rdf:Description>
```

**<http://bu.ch/123.html>**  
**author**  
**[surname "Doe" ;**  
 **firstname "John" . ] ;**  
**title "My Life" .**



# XML schema datatypes & literals

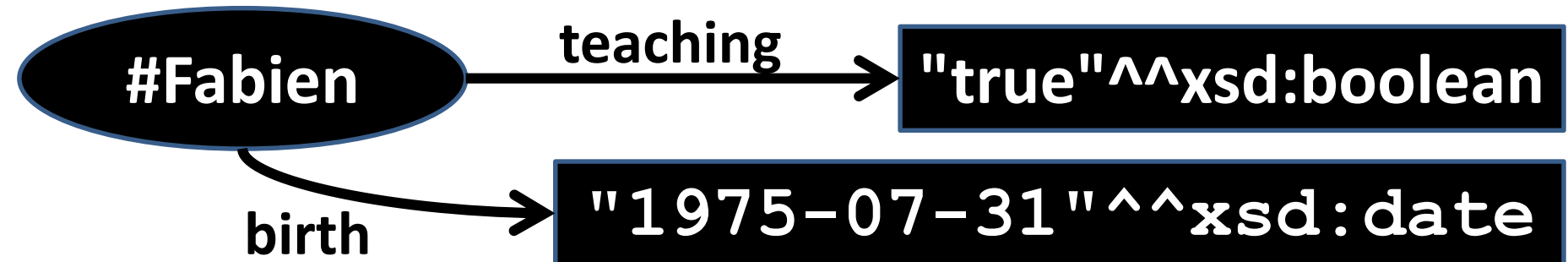
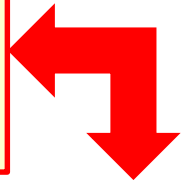
standard literals are xsd:string

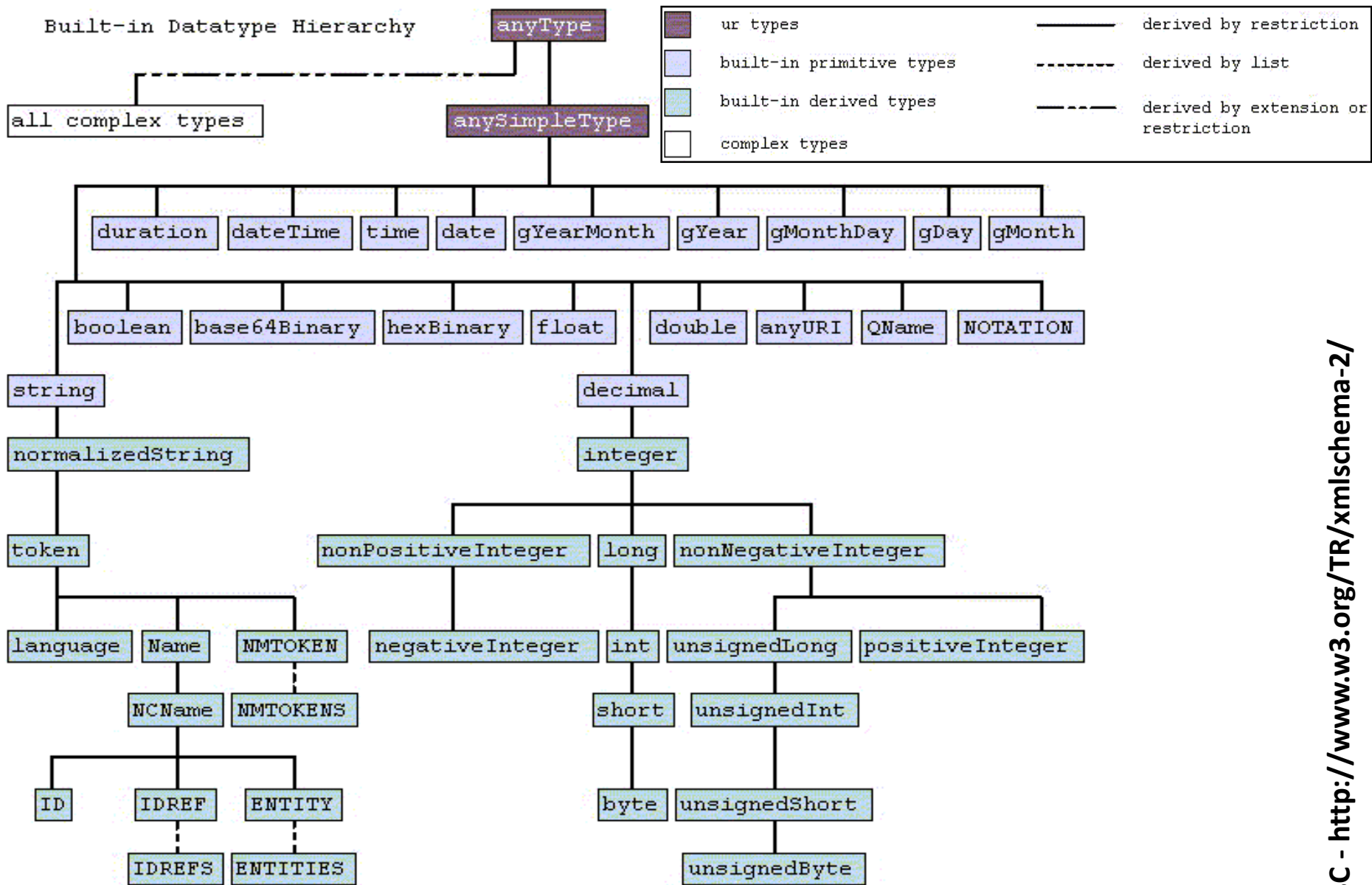
type literals with **datatypes from XML Schema**

```
<rdf:Description rdf:about="#Fabien">  
  <teaching rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean">  
    true</teaching>  
  <birth rdf:datatype="http://www.w3.org/2001/XMLSchema#date">  
    1975-07-31</birth>  
</rdf:Description/>
```



```
#Fabien teaching "true"^^xsd:boolean ;  
      birth "1975-07-31"^^xsd:date .
```





# XML Schema datatypes

```
<Book>
  <title xml:lang='fr'>Seigneur des anneaux</title>
  <title xml:lang='en'>Lord of the rings</title>
</Book>
```

```
<Book> title "Seigneur des anneaux"@fr ;
        title "Lord of the rings"@en .
```

literals with languages and without are disjoint

"Fabien" ≠ "Fabien"@en ≠ "Fabien"@fr

# language

# typing resources

using URIs to identify the types

```
<urn://~fgandon> rdf:type <http://www.inria.fr/schema#Person>
```

a resource can have several types

```
<urn://~fgandon> rdf:type <http://www.inria.fr/schema#Person>
```

```
<urn://~fgandon> rdf:type <http://www.inria.fr/schema#Researcher>
```

```
<urn://~fgandon> rdf:type <http://www.mit.edu/schema#Lecturer>
```

```
<rdf:Description rdf:about="urn://~fgandon">  
  <rdf:type rdf:resource="http://www.inria.fr/schema#Person" />  
  <name>Fabien</name>  
</rdf:Description>
```

```
<in:Person rdf:about="urn://~fgandon">  
  <name>Fabien</name>  
</in:Person>
```

```
<urn://~fgandon>  
  a in:Person ;  
  name "Fabien" .
```

# question:



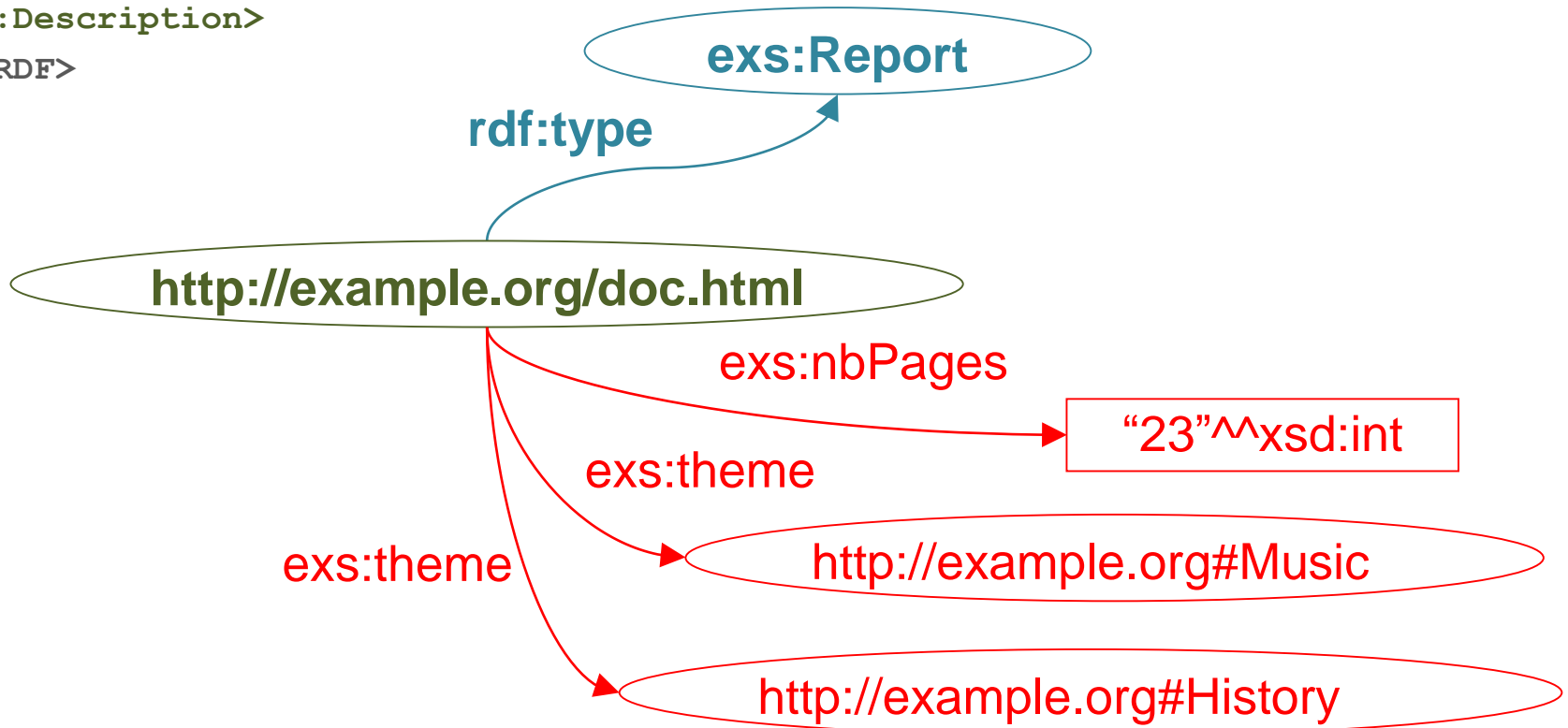
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:exs="http://example.org/schema#">
  <rdf:Description rdf:about="http://example.org/doc.html">
    <rdf:type rdf:resource="http://example.org/schema#Report"/>
    <exs:theme rdf:resource="http://example.org#Music"/>
    <exs:theme rdf:resource="http://example.org#History"/>
    <exs:nbPages rdf:datatype="http://www.w3.org/2001/XMLSchema#int">23</exs:nbPages>
  </rdf:Description>
</rdf:RDF>
```

# meaning ?



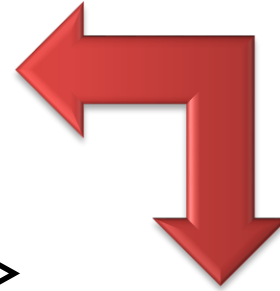
# question:

```
<?xml version="1.0"?>  
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:exs="http://example.org/schema#">  
  <rdf:Description rdf:about="http://example.org/doc.html">  
    <rdf:type rdf:resource="http://example.org/schema#Report"/>  
    <exs:theme rdf:resource="http://example.org#Music"/>  
    <exs:theme rdf:resource="http://example.org#History"/>  
    <exs:nbPages rdf:datatype="http://www.w3.org/2001/XMLSchema#int">23</exs:nbPages>  
  </rdf:Description>  
</rdf:RDF>
```



# bags = unordered groups

```
<rdf:Description rdf:about="#">
  <author>
    <rdf:Bag>
      <rdf:li>Ivan Herman</rdf:li>
      <rdf:li>Fabien Gandon</rdf:li>
    </rdf:Bag>
  </author>
</rdf:Description>
```



```
<#> author [
  a rdf:Bag ;
  rdf:li "Ivan Herman" ;
  rdf:li "Fabien Gandon" . ] .
```



```
<#> author _:a
_:a rdf:_1 "Ivan Herman"
_:a rdf:_2 "Fabien Gandon"
```





# sequence

ordered group of resources or literals

```
<rdf:Description rdf:about="#partition">
  <contains>
    <rdf:Seq>
      <rdf:li rdf:about="#C"/>
      <rdf:li rdf:about="#C"/>
      <rdf:li rdf:about="#C"/>
      <rdf:li rdf:about="#D"/>
      <rdf:li rdf:about="#E"/>
    </rdf:Seq>
  </contains>
</rdf:Description>
```

```
<partition>
  contains [
    a rdf:Seq ;
    rdf:li "C" ;
    rdf:li "C" ;
    rdf:li "C" ;
    rdf:li "D" ;
    rdf:li "E" .
  ] .
```



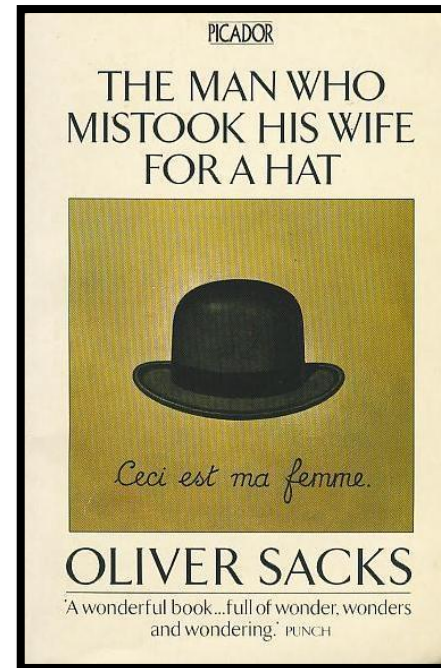
# alternatives

e.g. title of a book in different languages

```
<rdf:Description rdf:about="#book">
  <title>
    <rdf:Alt>
      <rdf:li xml:lang="fr">l'homme qui prenait sa femme
        pour un chapeau</rdf:li>
      <rdf:li xml:lang="en">the man who mistook his wife
        for a hat</rdf:li>
    </rdf:Alt>
  </title>
</rdf:Description>
```

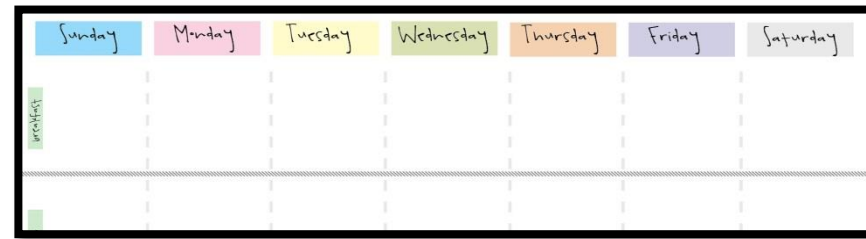


```
<#book>
  title [
    a rdf:Alt ;
    rdf:li "l'homme..."@fr ;
    rdf:li "the man..."@en .
  ] .
```

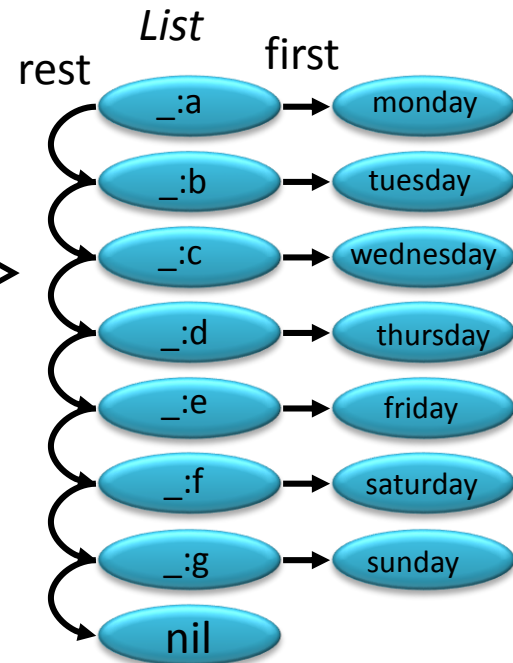


# collection

exhaustive and ordered list



```
<rdf:Description rdf:about="#week">
  <dividedIn rdf:parseType="Collection">
    <rdf:Description rdf:about="#monday"/>
    <rdf:Description rdf:about="#tuesday"/>
    <rdf:Description rdf:about="#wednesday"/>
    <rdf:Description rdf:about="#thursday"/>
    <rdf:Description rdf:about="#friday"/>
    <rdf:Description rdf:about="#saturday"/>
    <rdf:Description rdf:about="#sunday"/>
  </dividedIn>
</rdf:Description>
```



```
<#week> dividedIn
( <#monday> <#tuesday> <#wednesday>
  <#thursday> <#friday> <#saturday> <#sunday>
) .
```



[http://my\\_domain.org/my\\_path/my\\_type](http://my_domain.org/my_path/my_type)

# openmodel

- extensible vocabulary based on URIs
- anyone can say anything about anything



**link**  
to the world

May 2007



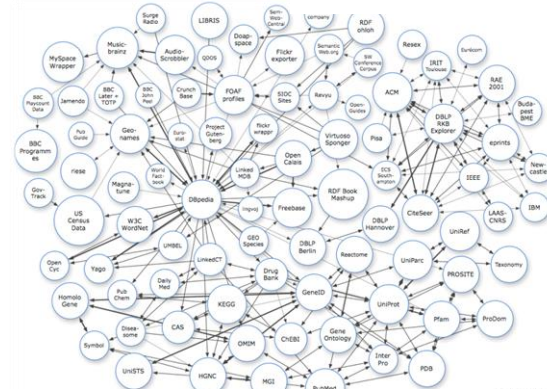
April 2008



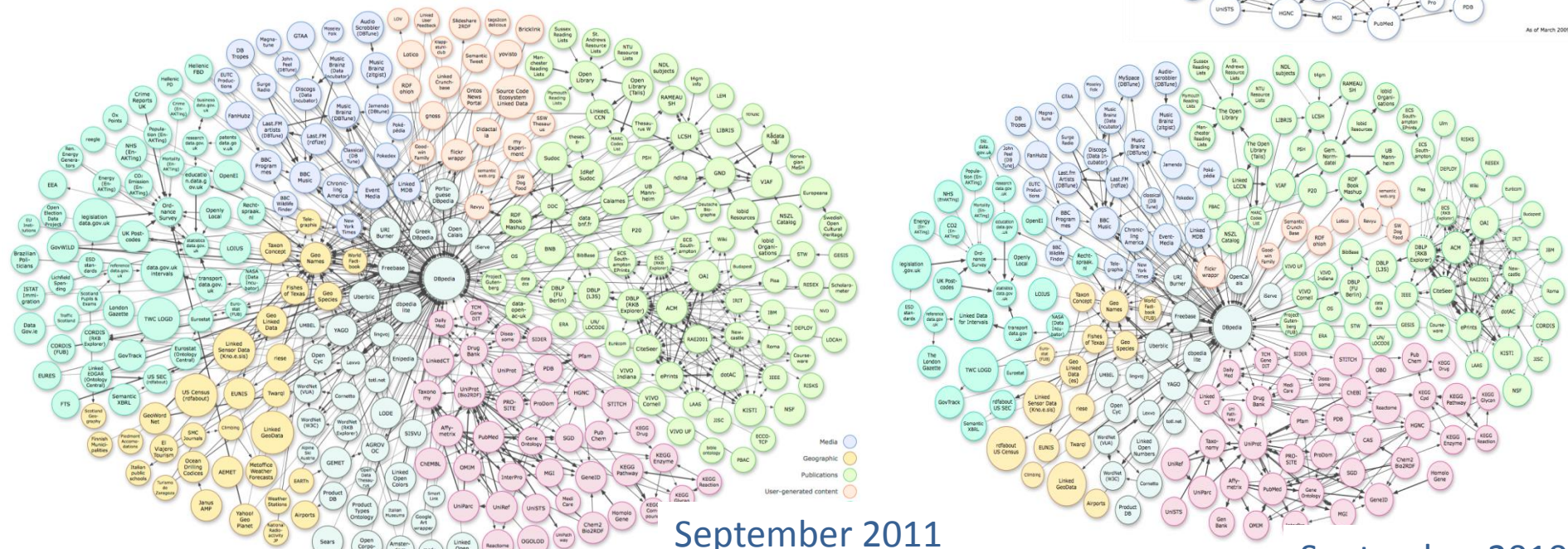
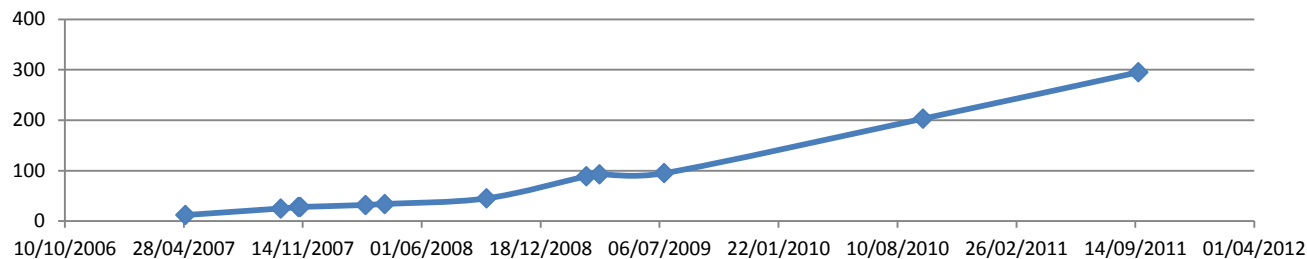
September 2008



March 2009



# Linking Open Data

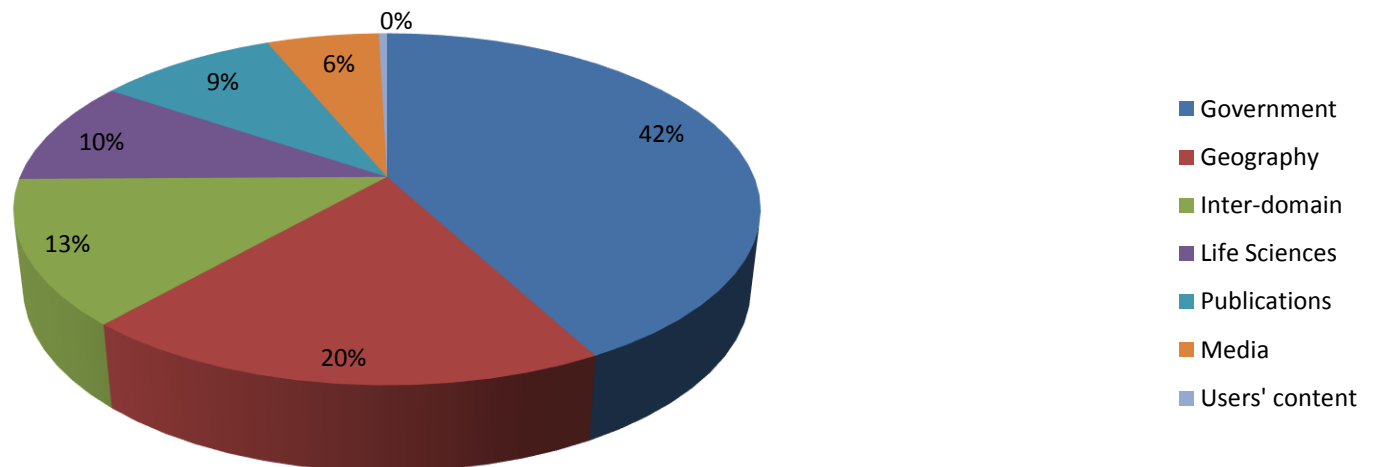


September 2011

September 2010

# thematic content

Domains	Number of datasets	Number of Triples	%	Out links	%
Media	<a href="#">25</a>	1 841 852 061	5,82 %	50 440 705	10,01 %
Geography	<a href="#">31</a>	6145 532 484	19,43 %	35 812 328	7,11 %
Government	<a href="#">49</a>	13 315 009 400	42,09 %	19 343 519	3,84 %
Publications	<a href="#">87</a>	2 950 720 693	9,33 %	139 925 218	27,76 %
Inter-domain	<a href="#">41</a>	4 184 635 715	13,23 %	63 183 065	12,54 %
Life Sciences	<a href="#">41</a>	3 036 336 004	9,60 %	191 844 090	38,06 %
Users' content	<a href="#">20</a>	134 127 413	0,42 %	3 449 143	0,68 %
	<b><a href="#">295</a></b>	<b>31 634 213 770</b>		<b>503 998 829</b>	



# data.ratatouille.com









Wright



Monday 1 February 2010



+ Add more to this page ?

Reset homepage



TOP NEWS STORY

### Top Israeli officers reprimanded

Israel reveals it has reprimanded two top officers over the use of white phosphorus shells in Gaza, its first such admission of errors.

>> More from BBC News

#### More Top Stories

Edit x



Iraqi female bomber kills dozens

US to resume Haiti victim airlift

S Lanka military chiefs dismissed

- Obama to unveil \$3.8tn US budget
- Nigerian oil pipeline sabotaged
- China accuses US over Taiwan deal
- Pratchett would test suicide law
- Beyonce is queen of Grammy Awards

>> MORE TOP STORIES



- Obama to unveil \$3.8tn US budget
- Nigerian oil pipeline sabotaged
- China accuses US over Taiwan deal

#### Sport

Edit x



Live - Transfer deadline day

Cairo delight at African Cup win

Latest updates from first F1 test

- Transfer trio boost Celtic squad
- Ferguson praises 'maturing' Nani
- Terry left to sweat on captaincy
- Geremi completes Newcastle exit
- Benitez dismisses Juventus deal

#### Travel

Edit x

from lonely planet



#### Spotlight

x

>> AEROSPACE



Singapore air show eyes military sales

After a bad year for airlines, manufacturers at the Singapore air show have higher hopes for sales of military hardware than civilian aircraft.

- Gloom ahead of Asia's largest air show
- Planemakers eye depressed cargo market
- 'Manage flights' to cut emissions

#### Business & Money

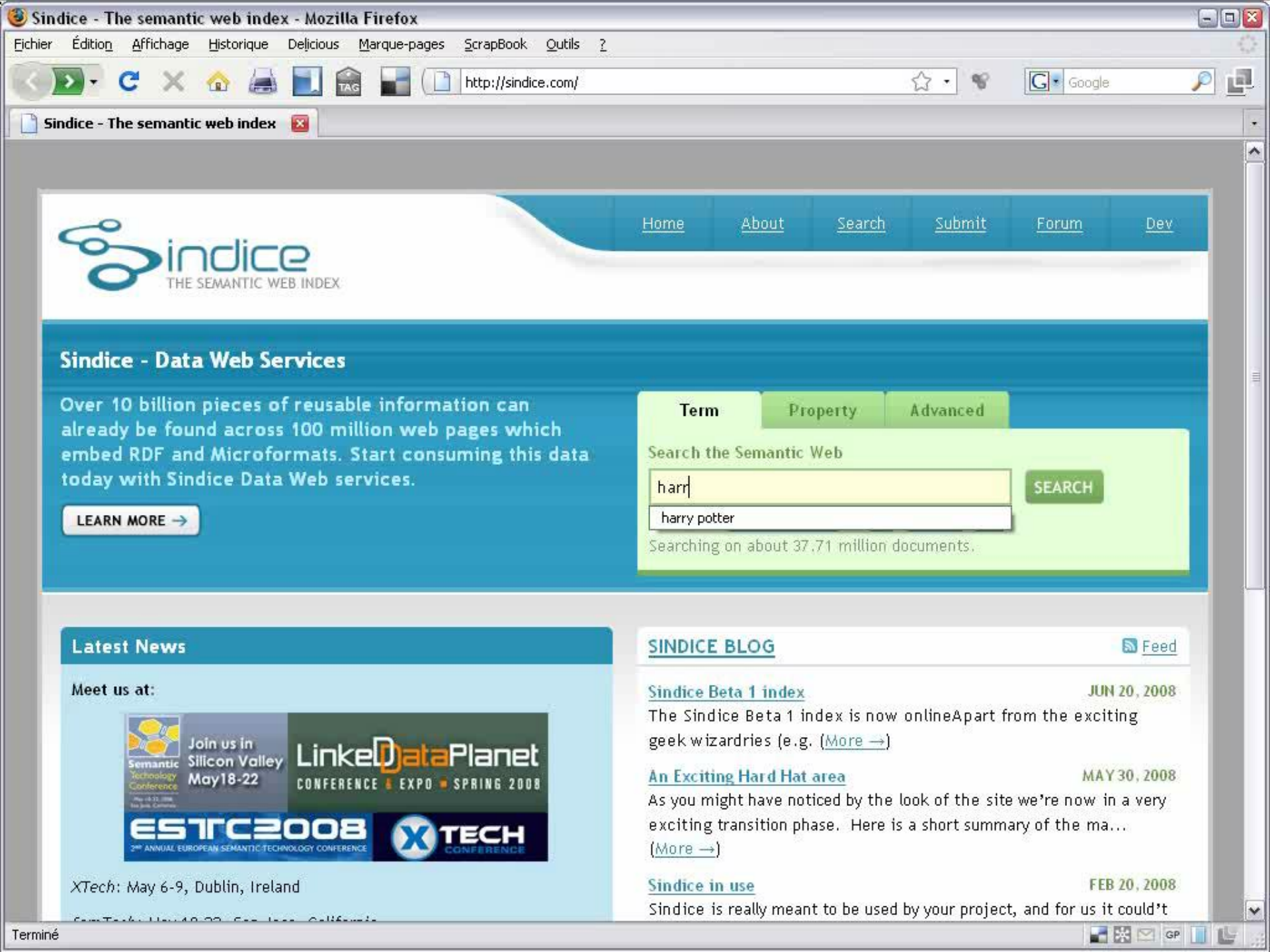
Edit x

>> MARKET DATA MON, 1 FEBRUARY 2010 12:37:42 GMT

Dow Jones	10067.33	▼	-53.13
Nasdaq	2147.35	▼	-31.65
FTSE 100	5188.53	▲	0.01
Dax	5604.08	▼	-4.71
Cac 40	3729.43	▼	-10.03

15 minute delay | Terms and Conditions

>> STERLING EXCHANGE RATES



### Sindice - Data Web Services

Over 10 billion pieces of reusable information can already be found across 100 million web pages which embed RDF and Microformats. Start consuming this data today with Sindice Data Web services.

LEARN MORE →

Term Property Advanced

Search the Semantic Web

harr

harry potter

SEARCH

Searching on about 37.71 million documents.

### Latest News

Meet us at:



XTech: May 6-9, Dublin, Ireland

SemTech: May 18-22, San Jose, California

### SINDICE BLOG

Feed

[Sindice Beta 1 index](#) JUN 20, 2008

The Sindice Beta 1 index is now online Apart from the exciting geek wizardries (e.g. [More](#) →)

[An Exciting Hard Hat area](#) MAY 30, 2008

As you might have noticed by the look of the site we're now in a very exciting transition phase. Here is a short summary of the ma... [More](#) →

[Sindice in use](#) FEB 20, 2008

Sindice is really meant to be used by your project, and for us it could't



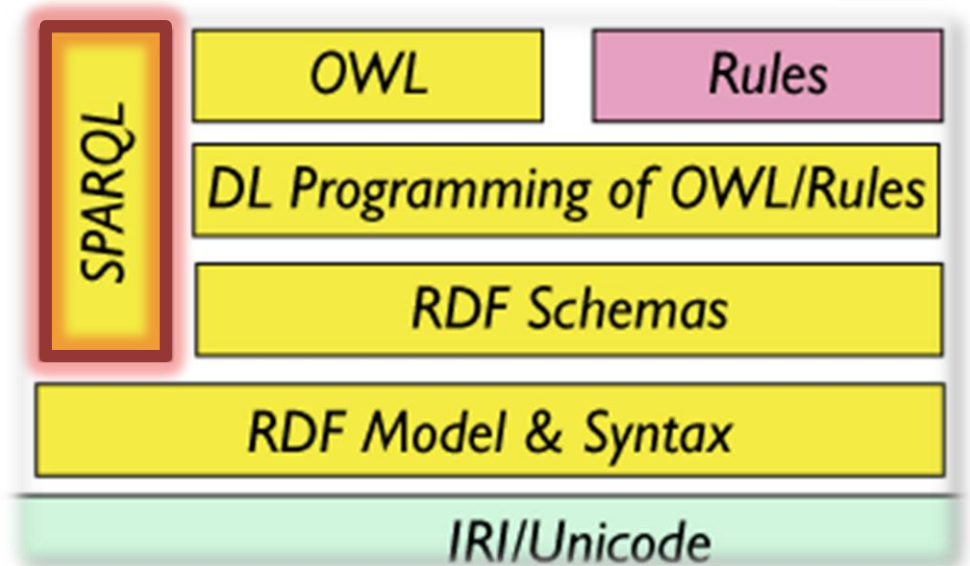
Lappin

# RDF 1.1 ?

- standardize Turtle syntax
- standardize a JSON syntax
- named graphs



**query with SPARQL**  
**SPARQL Protocol and RDF**  
**Query Language**



## SPARQL in 3 parts

part 1: query language

part 2: result format

part 3: access protocol





# SPARQL query

**SELECT** . . .

**FROM** . . .

**WHERE** { . . . }

# example

persons at least 18-year old

```
PREFIX ex: <http://inria.fr/schema#>
SELECT ?person ?name
WHERE {
  ?person rdf:type ex:Person .
  ?person ex:name ?name .
  ?person ex:age ?age .
  FILTER (?age > 17)
}
```



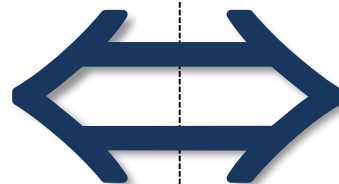
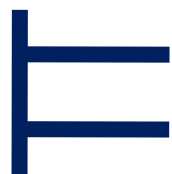
left( $x, y$ )

left( $y, z$ )

right( $z, v$ )

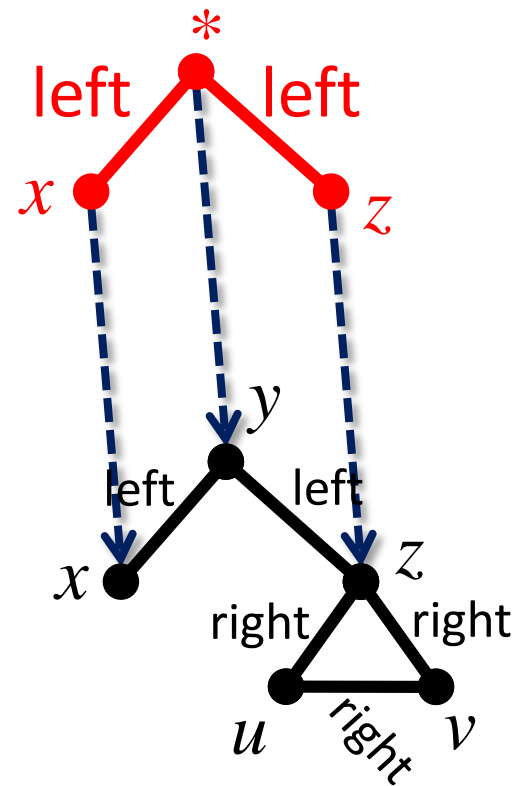
right( $z, u$ )

right( $u, v$ )



left( $x, ?p$ )

left( $?p, z$ )



# graph mapping / projection

classical three clauses:

- Select: clause to select the values to be returned
- Where: triple/graph pattern to match
- Filter: constraints expressed using test functions (XPath 2.0 or external)



# SPARQL triples

- triples and question marks for variables:

```
?x rdf:type ex:Person
```

- graph patterns to match:

```
SELECT ?subject ?property ?value  
WHERE {?subject ?property ?value}
```

- a pattern is, by default, a **conjunction** of triples

```
SELECT ?x WHERE  
{ ?x      rdf:type      ex:Person .  
  ?x      ex:name       ?name . }
```





# question:

- Query:

```
SELECT ?name WHERE {  
  ?x name ?name .  
  ?x email ?email .  
}
```

- Base:

```
_ :a name "Fabien" x2  
_ :b name "Thomas"  
_ :c name "Lincoln"  
_ :d name "Aline"  
_ :b email <mailto:thom@chaka.sn>  
_ :a email <mailto:Fabien.Gandon@inria.fr>  
_ :d email <mailto:avalandre@pachinko.jp>  
_ :a email <mailto:bafien@fabien.info>
```

- Results ?

# prefixes

to use namespaces:

```
PREFIX mit: <http://www.mit.edu#>
```

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

```
SELECT ?student
```

```
WHERE {
```

```
  ?student mit:registeredAt ?x .
```

```
  ?x foaf:homepage <http://www.mit.edu> .
```

```
}
```

Base namespace : **BASE** <...>





# SPARQL result

`failure/ success`

`values found`



# result formats

- a **binding** i.e. list of all the selected values (SELECT) for each answer found;  
(stable XML format ; e.g. for XSLT transformations)
- RDF **sub-graphs** for each answer found  
(RDF/XML format ; e.g. for application integration)
- JSON (eg. ajax web applications)
- CSV/TSV (eg. export)



# example of binding

results for previous query in XML

```
<?xml version="1.0"?>
<sparql xmlns="http://www.w3.org/2005/sparql-results#">
  <head>
    <variable name="student"/>
  </head>
  <results ordered="false" distinct="false">
    <result>
      <binding name="student">
        <uri>http://www.mit.edu/data.rdf#ndieng</uri></binding>
      </result>
      <result>
        <binding name="student">
          <uri>http://www.mit.edu/data.rdf#jdoe</uri></binding>
        </result>
      </results>
    </sparql>
```

# simplified syntax

triples with a common subject:

```
SELECT ?name ?fname
```

```
WHERE {
```

```
  ?x a Person;
```

```
    name ?name ;
```

```
    firstname ?fname ;
```

```
    author ?y . }
```



```
SELECT ?name ?fname
```

```
WHERE {
```

```
  ?x rdf:type Person .
```

```
  ?x name ?name .
```

```
  ?x firstname ?fname .
```

```
  ?x author ?y .
```

```
}
```

list of values

```
?x firstname "Fabien", "Lucien" .
```

blank node

```
[firstname "Fabien"] Or [] firstname "Fabien"
```

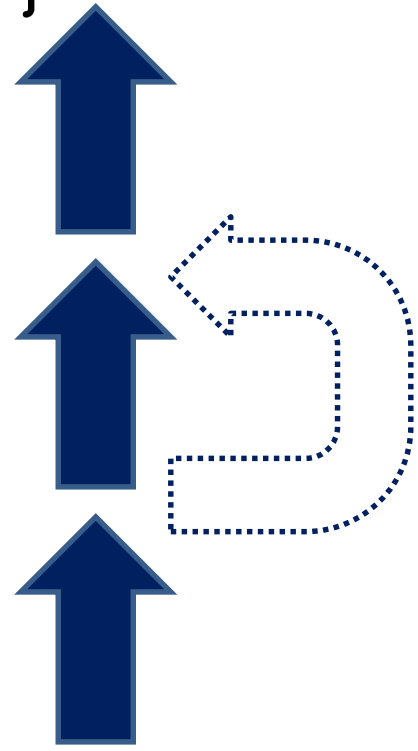
# source

```
PREFIX mit: <http://www.mit.edu#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
FROM http://www.mit.edu/data.rdf
SELECT ?student
WHERE {
  ?student mit:registeredAt ?x .
  ?x foaf:homepage <http://www.mit.edu> .
}
```

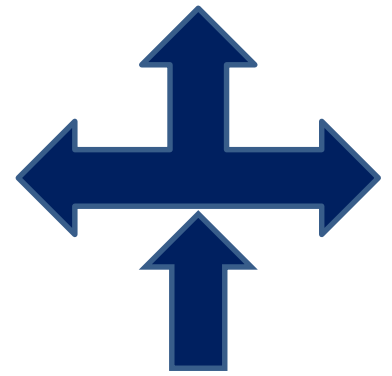
# optional part

```
PREFIX mit: <http://www.mit.edu#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?student ?name
WHERE {
  ?student mit:registeredAt ?x .
  ?x foaf:homepage <http://www.mit.edu> .
  OPTIONAL { ? student foaf:name ?name . }
}
```

possibly unbound



# union



alternative graph patterns

```
PREFIX mit: <http://www.mit.edu#>
```

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

```
SELECT ?student ?name
```

```
WHERE {
```

```
  ?student mit:registeredAt ?x .
```

```
  {
```

```
    {
```

```
      ?x foaf:homepage <http://www.mit.edu> .
```

```
    }
```

```
  UNION
```

```
  {
```

```
    ?x foaf:homepage <www.stanford.edu/> .
```

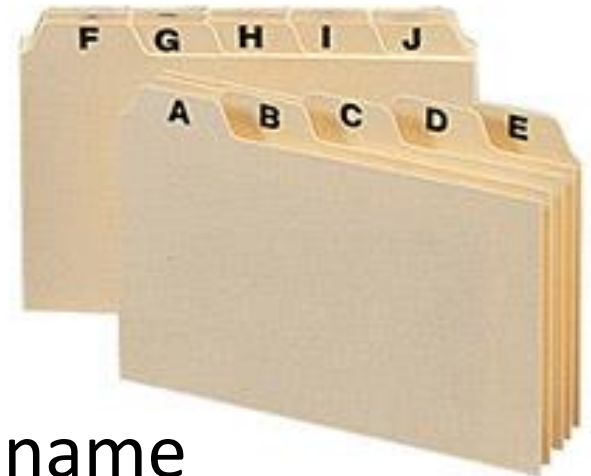
```
  }
```

```
}
```

```
}
```

# sort, filter and limit answers

```
PREFIX mit: <http://www.mit.edu#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?student ?name
WHERE {
  ?student mit:registeredAt ?x .
  ?x foaf:homepage <http://www.mit.edu> .
  ?student foaf:name ?name .
  ? student foaf:age ?age .
  FILTER (?age > 22)
}
ORDER BY ?name
LIMIT 20
OFFSET 20
```



students older than 22 years sorted by name  
results from number #21 to #40

# operators

- Inside the FILTER:
  - Comparators: `<`, `>`, `=`, `<=`, `>=`, `!=`
  - Tests on variables: `isURI(?x)`, `isBlank(?x)`, `isLiteral(?x)`, `bound(?x)`
  - Regular expression `regex(?x, "A.*")`
  - Attributes and values: `lang()`, `datatype()`, `str()`
  - Casting: `xsd:integer(?x)`
  - External functions and extensions
  - Boolean combinations: `&&`, `||`
- In the where WHERE: `@fr` , `^^xsd:integer`
- In the SELECT: `distinct`



# other functions (v 1.1)

**isNumeric (Val)** test it is a numeric value

**coalesce (val, ..., val)** first valid value

**IRI (Str) / URI (Str)** to build an iri/uri from a string

**BNODE (ID)** to build a blank node

**RAND ()** random value between 0 and 1

**ABS (Val)** absolute value

**CEIL (Val) , FLOOR (Val) , ROUND (Val)**

**NOW ()** today's date

**DAY (Date) , HOURS (Date) , MINUTES (Date) ,**

**MONTH (Date) , SECONDS (Date) ,**

**TIMEZONE (Date) , TZ (Date) , YEAR (Date)**

to access different parts of a date

**MD5 (Val) , SHA1 (Val) , SHA256 (Val) ,**

**SHA384 (Val) , SHA512 (Val)** hash functions

# string / literal functions (v1.1)

<code>STRDT (value, type)</code>	build a typed literal
<code>STRLANG (value, lang)</code>	build a literal with a language
<code>CONCAT (lit1, ..., litn)</code>	concatenate a list of literal
<code>CONTAINS (lit1, lit2), STRSTARTS (lit1, lit2), STRENDS (lit1, lit2)</code>	to test string inclusion
<code>SUBSTR (lit, start [, length])</code>	extract a sub string
<code>ENCODE_FOR_URI (Str)</code>	encodes a string as URI
<code>UCASE (Str), LCASE (Str)</code>	uppercase and lowercase
<code>STRLEN (Str)</code>	length of the string

# Aggregates

`group by + count, sum, min, max, avg, group_concat, or sample`

ex. average scores, grouped by the subject, but only where the mean is greater than 10

```
SELECT (AVG(?score) AS ?average)
WHERE { ?student score ?score . }
GROUP BY ?student
HAVING (AVG(?score) > 10)
```

# question:



```
PREFIX ex: <http://www.example.abc#>
SELECT ?person
WHERE {
    ?person rdf:type ?type .
    FILTER(! ( ?type = ex:Man ))
}
```

# minus

subtract a pattern



```
PREFIX ex: <http://www.example.abc#>
SELECT ?person
WHERE {
  { ?x rdf:type ex:Person }
  minus {?x rdf:type ex:Man}
}
```

# not exist

check the absence of a pattern



```
PREFIX ex: <http://www.example.abc#>
SELECT ?person
WHERE {
  ?x ex:memberOf ?org .
  filter (not exists
    {?y ex:memberOf <Hell>})
}
```

# if... then... else



```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select * where {
  ?x foaf:name ?name ; foaf:age ?age .
  filter (
    if (langMatches( lang(?name), "FR" ),
    ?age>=18, ?age>=21) )
}
```

# test a value is in / not in a list

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select * where {
  ?x foaf:name ?n .
  filter ( ?n in ( "fabien", "olivier",
                  "catherine" ) )
}
```





# values

pre-defined bindings

```
select ?person where {  
  ?person name ?name .  
  VALUES (?name)  
    { "Peter" "Pedro" "Pierre" }  
}
```



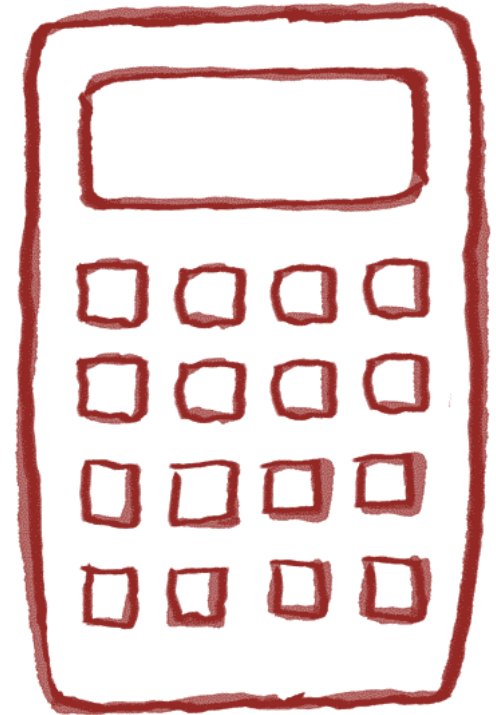
# paths

/ : sequence  
| : alternative  
+ : one or several  
\* : zero or several  
? : optional  
^ : reverse  
! : negation  
{min,max} : length

```
prefix foaf: <http://xmlns.com/foaf/0.1/>
select ?friends_fab where {
  ?x foaf:name "Fabien Gandon" ;
  foaf:knows+ ?friends_fab ;
}
```

# select expression

```
select ?x (year(?date) as ?year)  
where {  
  ?x birthdate ?date .  
}
```



# subquery / nested query



```
select ?name where {  
  {select (max(?age) as ?max)  
    where { ?person age ?age }  
  }  
  ?senior age ?max  
  ?senior name ?name  
}
```

# construct RDF as result

```
PREFIX mit: <http://www.mit.edu#>
```

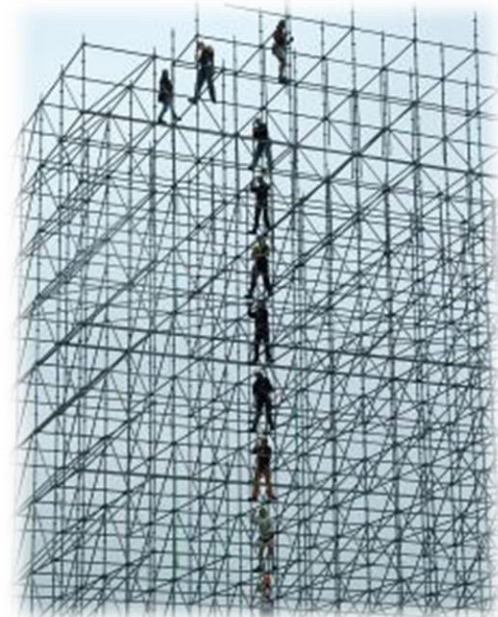
```
PREFIX corp: <http://mycorp.com/schema#>
```

```
CONSTRUCT
```

```
{ ?student rdf:type corp:FuturExecutive . }
```

```
WHERE
```

```
{ ?student rdf:type mit:Student . }
```



# free description



```
PREFIX mit: <http://www.mit.edu#>
```

```
DESCRIBE ?student
```

```
{ ?student rdf:type mit:Student . }
```

or

```
DESCRIBE <...URI...>
```



# **SPARQL** protocol

exchange queries and their  
results through the web



Wright





**e.g. DBpedia**

### SPARQL Explorer for <http://dbpedia.org/sparql>

**SPARQL:**

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX : <http://dbpedia.org/resource/>
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbpedia: <http://dbpedia.org/>
PREFIX skos: <http://www.w3.org/2004/02/skos/core#>
```

Results:

Powered by [OpenLink Virtuoso](#) and [dbpedia](#)

# Test on DBpedia



- Connect to:

<http://dbpedia.org/snorql/> or

<http://fr.dbpedia.org/sparql> or ...

<http://wiki.dbpedia.org/Internationalization/Chapters>

- Query:

```
SELECT * WHERE {  
    ?x rdfs:label "Paris"@fr .  
    ?x ?p ?v .  
}  
LIMIT 10
```



**Partition** x **Ranking**

Nodes Edges

Group Show Pie

**Layout** x

---Choose a layout

<No Properties>

Presets... Reset

Semantic Web Import x Graph x

Driver Request Log Configurations

SPARQL Request

```

1 prefix dbpedia-owl: <http://dbpedia.org/ontology/>
2 prefix dbpedia-pro: <http://dbpedia.org/property/>
3 prefix geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
4 prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
5 prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
6 prefix geonames: <http://sws.geonames.org/>
7 prefix wgs84_pos: <http://www.w3.org/2003/01/geo/wgs84_pos#>
8 prefix gephi: <http://gephi.org/>
9 prefix owl: <http://www.w3.org/2002/07/owl#>
10
11 construct {
12     ?city1 owl:sameLanguage ?city2 .
13     ?city1 gephi:label ?city_name1 ;
14         gephi:longitude ?long1 ;
15         gephi:latitude ?lat1 ;
16         gephi:image ?im1 ;
17         gephi:language ?language ;
18         gephi:populationTotal ?population1 .
19
20     ?city2 gephi:label ?city_name2 ;
21         gephi:longitude ?long2 ;
22         gephi:latitude ?lat2 ;
23         gephi:image ?im2 ;
24         gephi:language ?language ;
25         gephi:populationTotal ?population2 .
26 }
27 where {
28     {
29         service <http://live.dbpedia.org/sparql/> {
  
```

Run

**Context** x

**Nodes:** 765

**Edges:** 2184

Directed Graph

St... x Filters Mul...

Settings

**Network Overview**

- Average Degree
- Avg. Weighted Degree
- Network Diameter
- Graph Density
- HITS
- Modularity
- PageRank
- Erdős Number
- Connected Components

**Node Overview**

- Avg. Clustering Coefficient
- SW Type Statistics
- Clustering Coefficient
- Eigenvector Centrality

**Edge Overview**

- Avg. Path Length
- Neighborhood Overlap, Embeddedness

RDF Request Graph



Lappin



User Interface & Applications

Trust

Proof

Unifying Logic

SPARQL

Ontology:  
OWL

RDF-S

**PUBLISH  
SEMANTICS  
OF SCHEMAS**

RDF

XML

URI/IRI

**RDFS** means **R****D****F** Schema

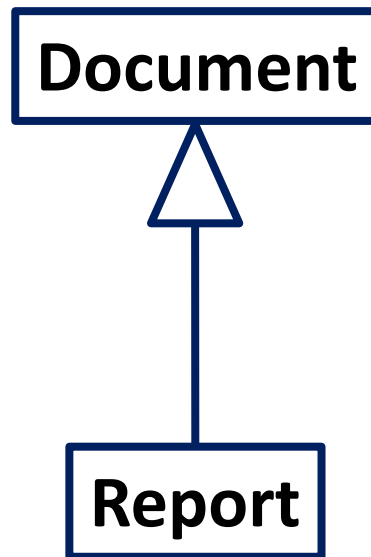




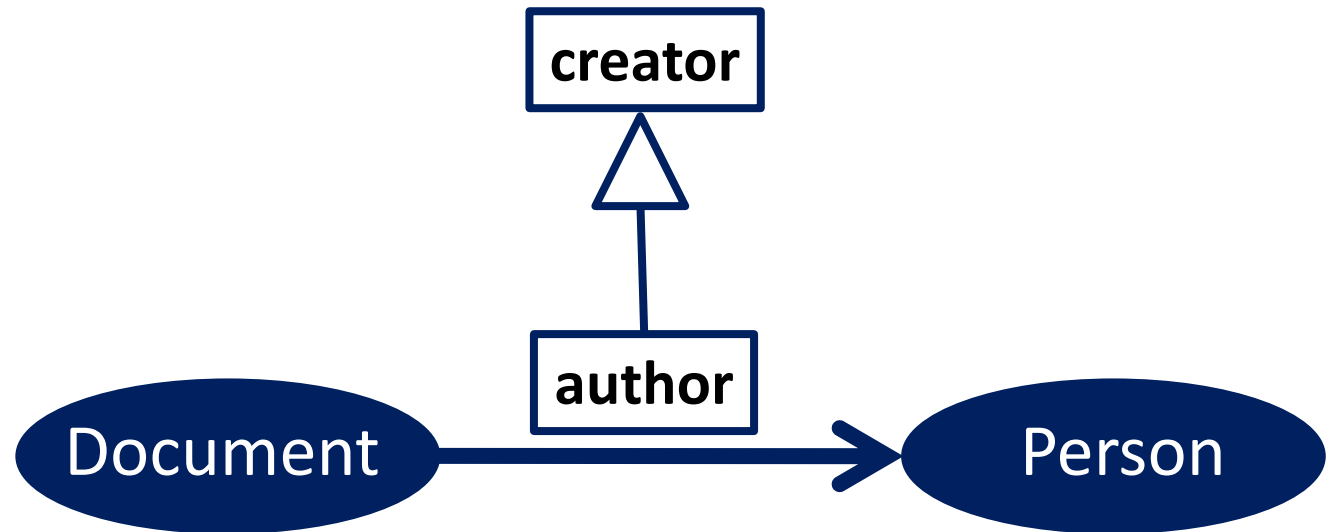
**RDFS** provides primitives to *Write*  
lightweight ontologies

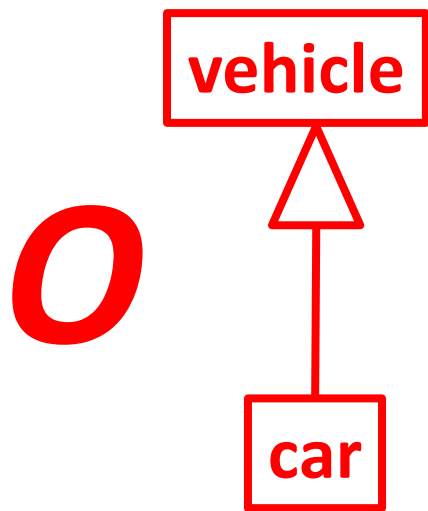


**RDFS** to define classes of resources  
and organize their hierarchy

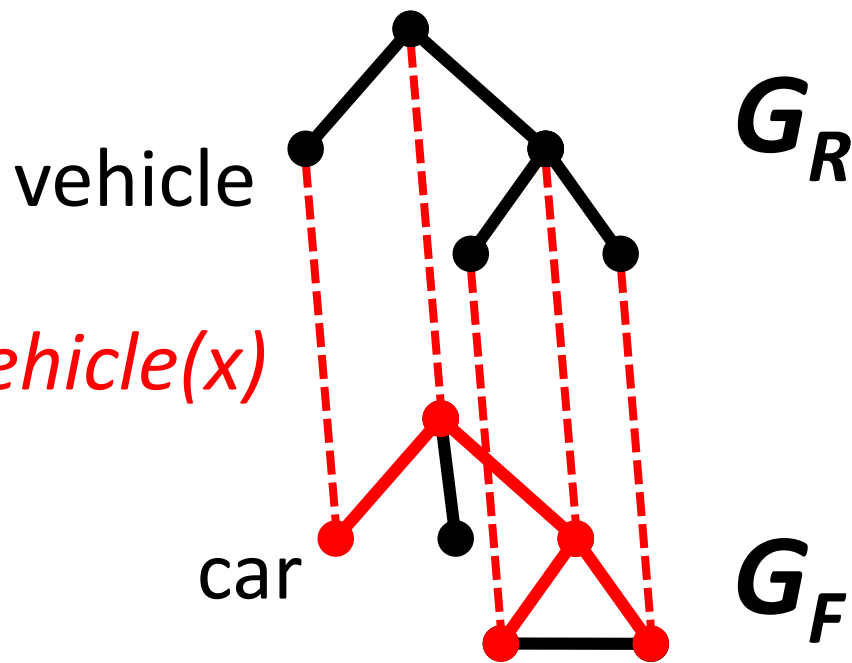


**RDFS** to define relations between resources, their signature and organize their hierarchy





$car(x) \Rightarrow vehicle(x)$



$$F \wedge O \rightarrow R \Leftrightarrow G_F \leq G_R$$

mapping modulo an ontology



# example of RDFS schema

```
<rdf:RDF xml:base = "http://inria.fr/2005/humans.rdfs"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns      = "http://www.w3.org/2000/01/rdf-schema#">
  <Class rdf:ID="Man">
    [ <subClassOf rdf:resource="#Person"/>
      <subClassOf rdf:resource="#Male"/>
      <label xml:lang="en">man</label>
      <comment xml:lang="en">an adult male person</comment>
    </Class>
```



**<Man> a Class ; subClassOf <Person>, <Male> .**

# example of RDFS properties

```
<rdf:RDF xml:base = "http://inria.fr/2005/humans.rdfs"
  xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
  xmlns = "http://www.w3.org/2000/01/rdf-schema#"
  <rdf:Property rdf:ID="hasMother">
    [ <subPropertyOf rdf:resource="#hasParent"/>
      <range rdf:resource="#Female"/>
      <domain rdf:resource="#Human"/>
      <label xml:lang="en">has for mother</label>
      <comment xml:lang="en">to have for parent a female.
      </comment>
    ]
  </rdf:Property>
```



```
<hasMother> a rdfs:Property ;
  subPropertyOf <hasParent> ;
  range <Female> ; domain <Human> .
```

```
<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="http://inria.fr/2005/humans.rdfs#"
  xml:base="http://inria.fr/2005/humans.rdfs-instances" >
```

```
<rdf:Description rdf:ID="Lucas">
  <rdf:type
rdf:resource="http://inria.fr/2005/humans.rdfs#Man"/>
  <hasMother rdf:resource="#Laura"/>
</rdf:Description>
```

```
<Man rdf:ID="Lucas">
  <hasMother rdf:resource="#Laura"/>
</Man>
```



**<Luca> a Man; hasMother <Laura> .**

example of RDF using this schema

# rdfs:label

a resource may have one or more labels in one or more natural language

```
<rdf:Property rdf:ID='name'>
  <rdfs:domain rdf:resource='Person' />
  <rdfs:range rdf:resource='&rdfs;Literal' />
  <rdfs:label xml:lang='fr'>nom</rdfs:label>
  <rdfs:label xml:lang='fr'>nom de famille</rdfs:label>
  <rdfs:label xml:lang='en'>name</rdfs:label>
</rdf:Property>
```




```
<name> a rdfs:Property ;
  range rdfs:Literal ; domain <Person> ;
  label "nom"@fr, "nom de famille"@fr, "name"@en .
```



# rdfs:comment & rdfs:seeAlso

comments provide definitions and explanations in natural language


```
<rdfs:Class rdf:about='#Woman' >
  <rdfs:subClassOf rdf:resource="#Person" />
  <rdfs:comment xml:lang='fr' >une personne adulte du
    sexe féminin</rdfs:comment>
  <rdfs:comment xml:lang='en' >a female adult person
</rdfs:comment>
</rdfs:Class>
```



```
<Woman> a rdfs:Class ; rdfs:subClassOf <Person> ;
  rdfs:comment "adult femal person"@en ;
  rdfs:comment "une adulte de sexe féminin"@fr .
```

see also...

```
<rdfs:Class rdf:about='#Man' >
  <rdfs:seeAlso rdf:resource='#Woman' />
</rdfs:Class>
```



```
<Man> a rdfs:Class ; rdfs:seeAlso <Woman> .
```



# CORESE/ KGRAM

Corese/KGRAM GUI - v2.4.2.6 - 2010-11-11

File Edit Engine Debug ?

System +

Loaded files: Debug Reload

Logs:

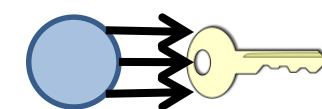
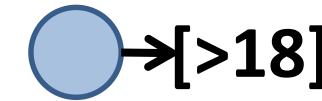
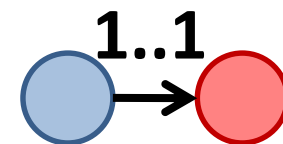
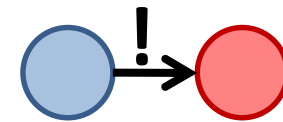
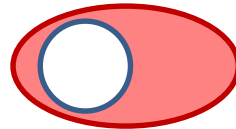
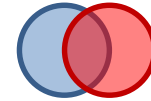
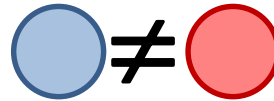
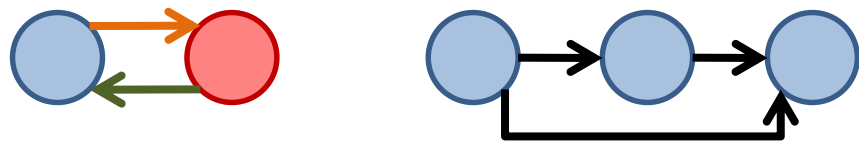
```
reset...  
RDFS/Rules : 0.00 s  
Load Triples : 488  
Load Relations : 423  
Load Concepts : 135  
  
done.
```



**OWL** provides **additional**  
**primitives** for  
**heavyweight ontologies**



# OWL in one...



union

disjunction

intersection

complement

restriction

cardinality

equivalence

enumeration

value restrict.

disjoint union

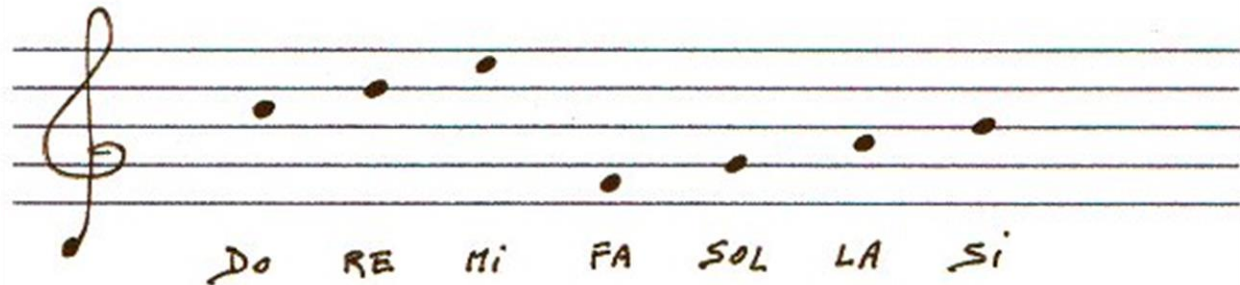
keys

...

# enumerated class {a,b,c,d,e}

define a class by providing all its members

```
<owl:Class rdf:id="EyeColor">  
  <owl:oneOf rdf:parseType="Collection">  
    <owl:Thing rdf:ID="Blue"/>  
    <owl:Thing rdf:ID="Green"/>  
    <owl:Thing rdf:ID="Brown"/>  
    <owl:Thing rdf:ID="Black"/>  
  </owl:oneOf>  
</owl:Class>
```



# classes defined by union



of other classes

```
<owl:Class>  
  <owl:unionOf rdf:parseType="Collection">  
    <owl:Class rdf:about="#Person"/>  
    <owl:Class rdf:about="#Group"/>  
  </owl:unionOf>  
</owl:Class>
```

# classes defined by intersection

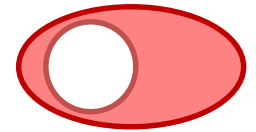
of other classes

```
<owl:Class rdf:ID="Man">  
  <owl:intersectionOf rdf:parseType="Collection">  
    <owl:Class rdf:about="#Male"/>  
    <owl:Class rdf:about="#Person"/>  
  </owl:intersectionOf>  
</owl:Class>
```

# complement and disjunction

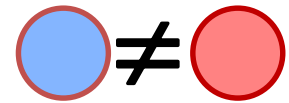
## complement class

```
<owl:Class rdf:ID="Male">  
  <owl:complementOf rdf:resource="#Female"/>  
</owl:Class>
```



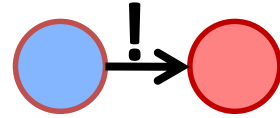
## declare a disjunction

```
<owl:Class rdf:ID="Square">  
  <owl:disjointWith rdf:resource="#Round"/>  
</owl:Class>
```



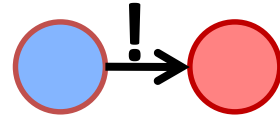


# restriction on all values



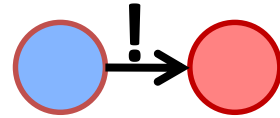
```
<owl:Class rdf:ID="Herbivore">
  <subClassOf rdf:resource="#Animal" />
  <subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#eats" />
      <owl:allValuesFrom rdf:resource="#Plant" />
    </owl:Restriction>
  </subClassOf>
</owl:Class>
```

# restriction on some values



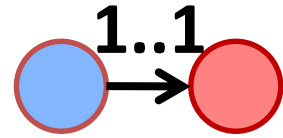
```
<owl:Class rdf:ID="Sportive">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#hobby" />
      <owl:someValuesFrom rdf:resource="#Sport" />
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

# restriction to an exact value



```
<owl:Class rdf:ID="Bike">  
  <subClassOf>  
    <owl:Restriction>  
      <owl:onProperty rdf:resource="#nbWheels" />  
      <owl:hasValue>2</owl:hasValue>  
    </owl:Restriction>  
  </subClassOf>  
</owl:Class>
```

# restriction on cardinality



how many times a property is used for a same subject but with different values

- Constraints: minimum, maximum, exact number
- Exemple

```
<owl:Class rdf:ID="Person">
  <subClassOf>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#name" />
      <owl:maxCardinality>1</owl:maxCardinality>
    </owl:Restriction>
  </subClassOf>
</owl:Class>
```

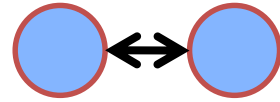
# types of properties

- `ObjectProperty` are relations between resources only  
e.g. `hasParent(#thomas,#stephan)`
- `DatatypeProperty` have a literal value possibly typed  
ex: `hasAge(#thomas,16^^xsd:int)`
- `AnnotationProperty` are ignored in inferences and used for documentation and extensions

# algebraic properties

- **Symmetric** property,  $xRy \Rightarrow yRx$

```
<owl:SymmetricProperty rdf:ID="hasSpouse" />
```

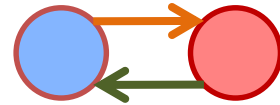


- **Inverse** property,  $xR_1y \Leftrightarrow yR_2x$

```
<rdf:Property rdf:ID="hasChild">
```

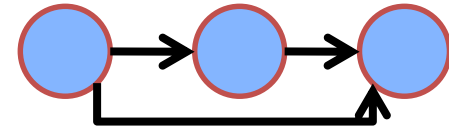
```
  <owl:inverseOf rdf:resource="#hasParent"/>
```

```
</rdf:Property>
```



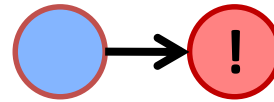
- **Transitive** property,  $xRy \ \& \ yRz \Rightarrow xRz$

```
<owl:TransitiveProperty rdf:ID="hasAncestor" />
```



- **Functional** property,  $xRy \ \& \ xRz \Rightarrow y=z$

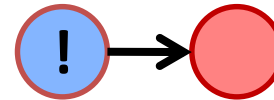
```
<owl:FunctionalProperty rdf:ID="hasMother" />
```



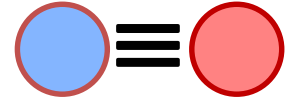
- **Inverse functional** property,  $xRy \ \& \ zRy \Rightarrow x=z$

```
<owl:InverseFunctionalProperty
```

```
  rdf:ID="hasSocialSecurityNumber" />
```



# equivalencies and alignment



- equivalent classes : `owl:equivalentClass`
- equivalent properties: `owl:equivalentProperty`
- identical or different resources:  
`owl:sameAs`, `owl:differentFrom`

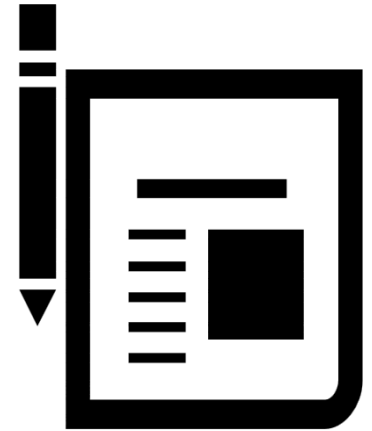
# document the schemas

## description of the ontology

`owl:Ontology`, `owl:imports`, `owl:versionInfo`,  
`owl:priorVersion`, `owl:backwardCompatibleWith`,  
`owl:incompatibleWith`

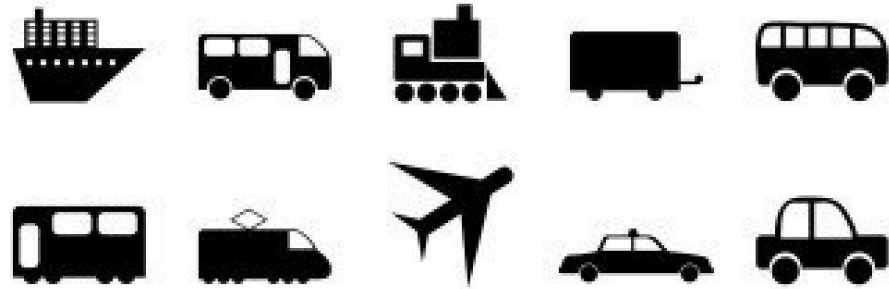
## versions of classes and properties

`owl:DeprecatedClass`, `owl:DeprecatedProperty`





# OWL profiles



**EL:** large numbers of properties and/or classes and polynomial time.

**QL:** large volumes of instance data, and conjunctive query answering using conventional relational database in LOGSPACE

**RL:** scalable reasoning without sacrificing too much expressive power using rule-based reasoning in polynomial time



- Dublin core
- Creative Commons
- FOAF

...



Welcome to LOV, your entry point to the growing ecosystem of linked open vocabularies (RDFS or OWL ontologies) used in the Linked Data Cloud. Here you will find vocabularies **listed** and **individually described** by metadata, **classified** by vocabulary spaces, interlinked using the dedicated vocabulary VOAFA.

You will enjoy querying the LOV dataset either at **vocabulary level** or at **element level**, exploring the vocabulary content using full-text faceted **search**, and finding **metrics** about the use of vocabularies in the Semantic Web.

Not finding your favourite one? Suggest a new vocabulary to add to LOV!

[Read more ...](#)



Search

## Metadata:

Property	Value
Creator	Bernard Vatant, Pierre-Yves Vandenbussche
Publisher	Mondeca
SPARQL Endpoint	<a href="http://lov.okfn.org/endpoint/lov">http://lov.okfn.org/endpoint/lov</a>
RDF dump File	<a href="http://lov.okfn.org/dataset/lov/lov.rdf">http://lov.okfn.org/dataset/lov/lov.rdf</a>
Last modified	2012-09-10
Status	Work in progress



Help us and suggest a new vocabulary !

## Vocabulary spaces (285):

Classifying vocabularies is a brand new challenge for Library Science, and no existing classification scheme seems to fit the need so far. Unless we missed something, and in that case we are open to suggestions from the librarian community.

# *le doggy-bag* *de la présentation*





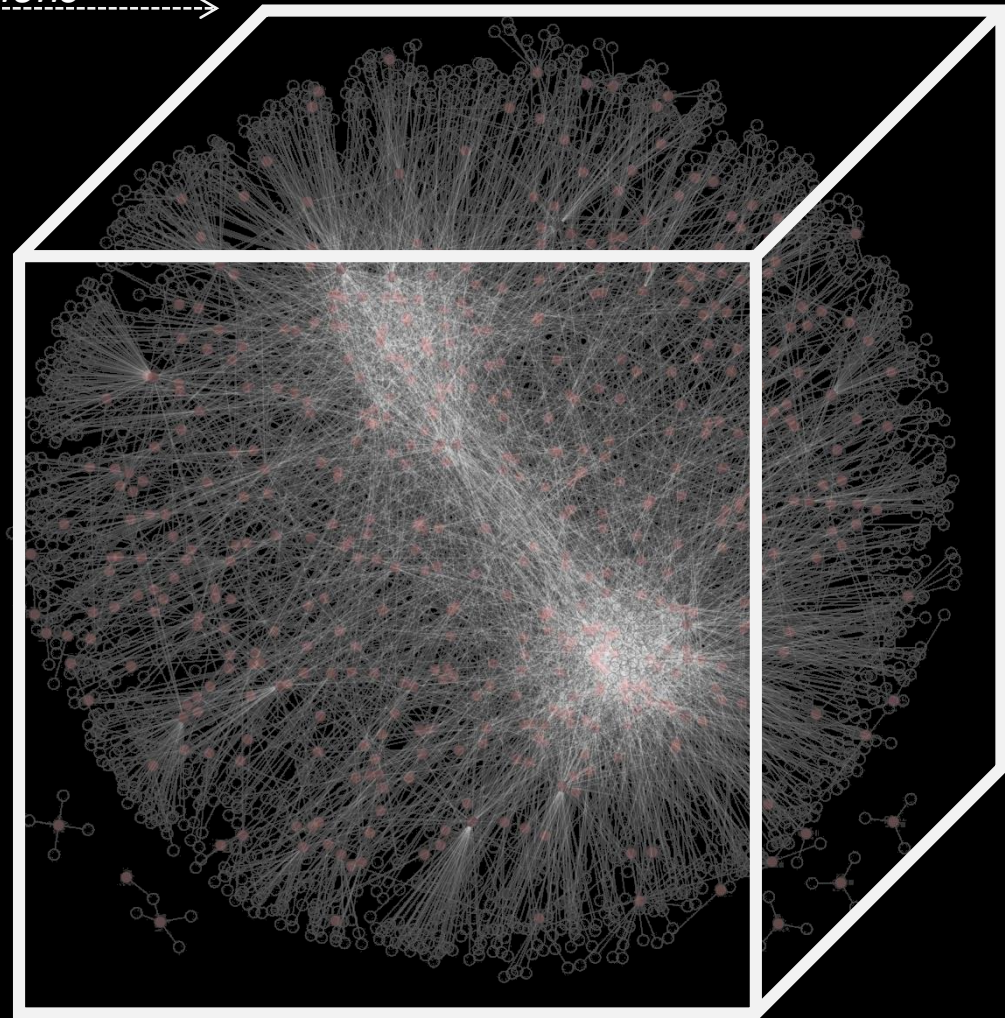
open your data  
to those who could use them

W3C ©

66 FOAF primitives

3 475 908 348 references <sup>(2)</sup>

←----- *x 52 millions* -----→



**“a small tree ruling a big graph”<sup>(1)</sup>**

(1) Franck Van Harmelen, ISWC 2011

(2) Libby Miller, 2009



[C. Welty, ISWC 2007]

“semantic web”

*and not*

“semantic web”



[J. Hendler, ISWC 2007]

“a lightweight ontology  
allows us to do  
lightweight reasoning”

Click to **LOOK INSIDE!**

## The Quark and the Jaguar: Adventures in the Simple and the Complex [Paperback]

[Murray Gell-Mann](#)  (Author)

★★★★☆  (30 customer reviews)

Price: **\$13.59**

**In Stock.**

Ships from and sold by **Amazon.com**.

### Formats

	Amazon Price	New from	Used from
<input checked="" type="checkbox"/> Hardcover	--	\$13.67	\$0.55
<input checked="" type="checkbox"/> Paperback	<b>\$13.59</b>	\$10.01	\$7.84

[Show 4 more formats](#)

### Product Details

**Paperback:** 392 pages

**Publisher:** St. Martin's Griffin (September 15, 1995)

**Language:** English

**ISBN-10:** 0805072535

**ISBN-13:** 978-0805072532

**Product Dimensions:** 9.1 x 6.1 x 1.2 inches

**Shipping Weight:** 1.2 pounds ([View shipping rates and policies](#))

# web 1, 2



Click to **LOOK INSIDE!**

## The Quark and the Jaguar: Adventures in the Simple and the Complex [Paperback]

[Murray Gell-Mann](#)  (Author)

★★★★☆  (30 customer reviews)

Price: **\$13.59**

**In Stock.**

Ships from and sold by **Amazon.com**.

### Formats

	Amazon Price	New from	Used from
<input type="checkbox"/> Hardcover	--	\$13.67	\$0.55
<input checked="" type="checkbox"/> Paperback	<b>\$13.59</b>	\$10.01	\$7.84

[Show 4 more formats](#)

### Product Details

**Paperback:** 392 pages

**Publisher:** St. Martin's Griffin (September 15, 1995)

**Language:** English

**ISBN-10:** 0805072535

**ISBN-13:** 978-0805072532

**Product Dimensions:** 9.1 x 6.1 x 1.2 inches

**Shipping Weight:** 1.2 pounds ([View shipping rates and policies](#))

**Average Customer Review:** ★★★★★  (30 customer reviews)

**Amazon Bestsellers Rank:** #373,365 in Books

### Customer Reviews

30 Reviews

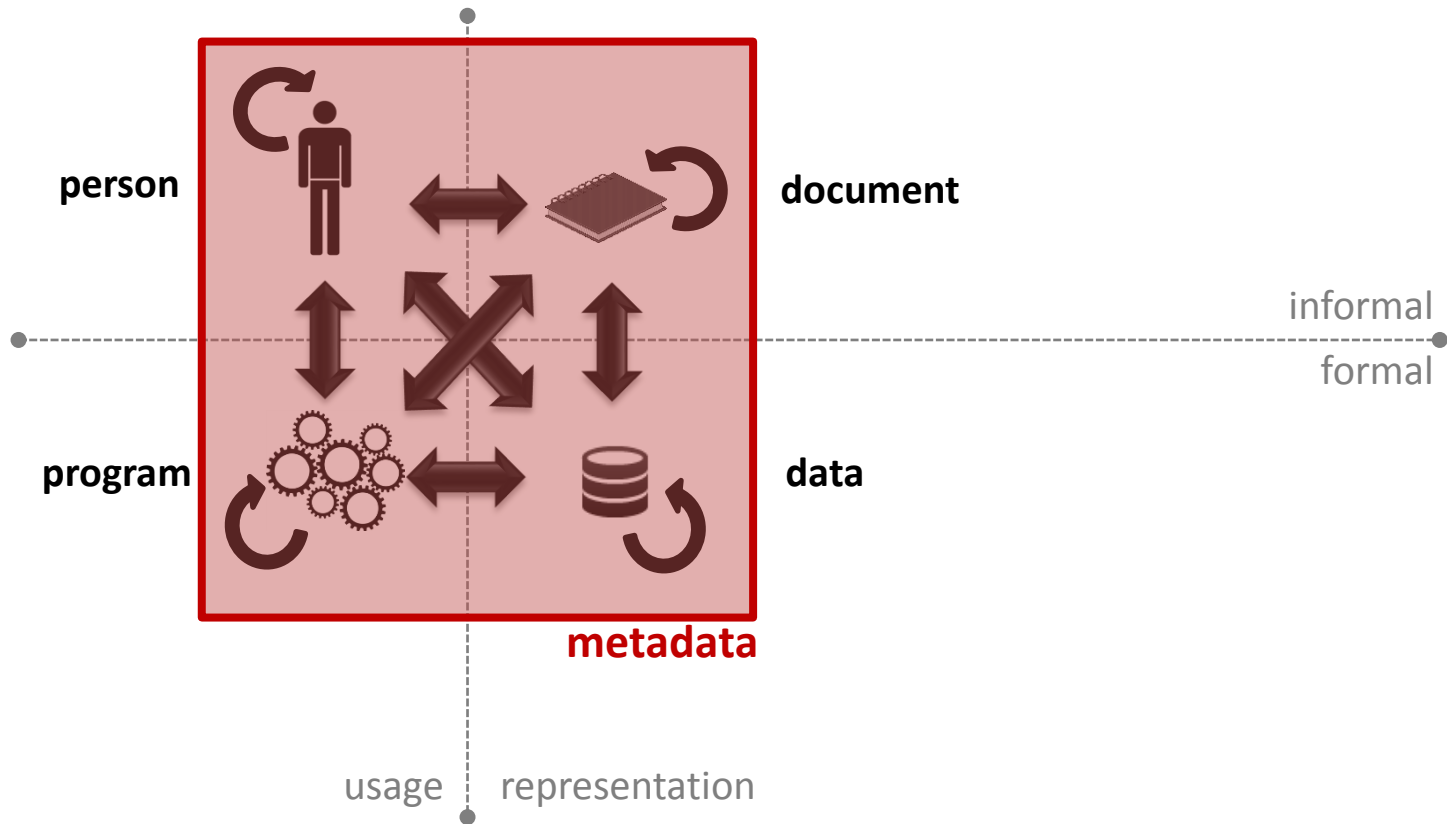
5 star:  (9)  
 4 star:  (8)  
 3 star:  (7)  
 2 star:  (4)  
 1 star:  (2)

**Average Customer Review**

★★★★☆ (30 customer reviews)

# web 1, 2, 3

# one web...



**tomorrow,** he, who controls the metadata,  
controls the web.

Fabien Gandon, <http://fabien.info>, @fabien\_gandon