Semantic Web technologies: An introduction to Artificial Intelligence on the Web

PENS project meeting @ Middlesex Univ. London (Feb 2019), day 1

Salvador Sánchez Alonso

This talk in PENS

- Course: Digital Business Transformation

III. Course Matrix Contents

Week	Main Topics / Chapters	Learning Hours	Intended Learning Outcome (s)
1	Introduction	3	N/A
2	Digital Transformation : How Technology Changes Business In industry and service, in the government, in the Bank, and Omni-channel Biz & Demat	6	LO.#-1,2
3 4	The Mechanics of Disruption and their effect on collaborative work and infrastructure	12	LO.#-2,3,4
5	Agility and DevOps	6	LO.#-4,5
6 7	Artificial Intelligence & Knowledge management	12	LO.#4,5
8 9	Cloud Computing and API for Bi-Modal IT	12	LO.#-4,5
10 11	IoT, Big Data and Data analytics	12	LO.#-4,5
12	Platforms in a Digital Economy	9	LO.#-4,5
13	Cybersecurity, privacy and ethics.	6	LO.#-3,5,8
14 15	Projects* fulfillment and presentation	50 for fulfillment along the semester 12 for presentations	LO.#-6,7,8
Total Learning Hours	140		

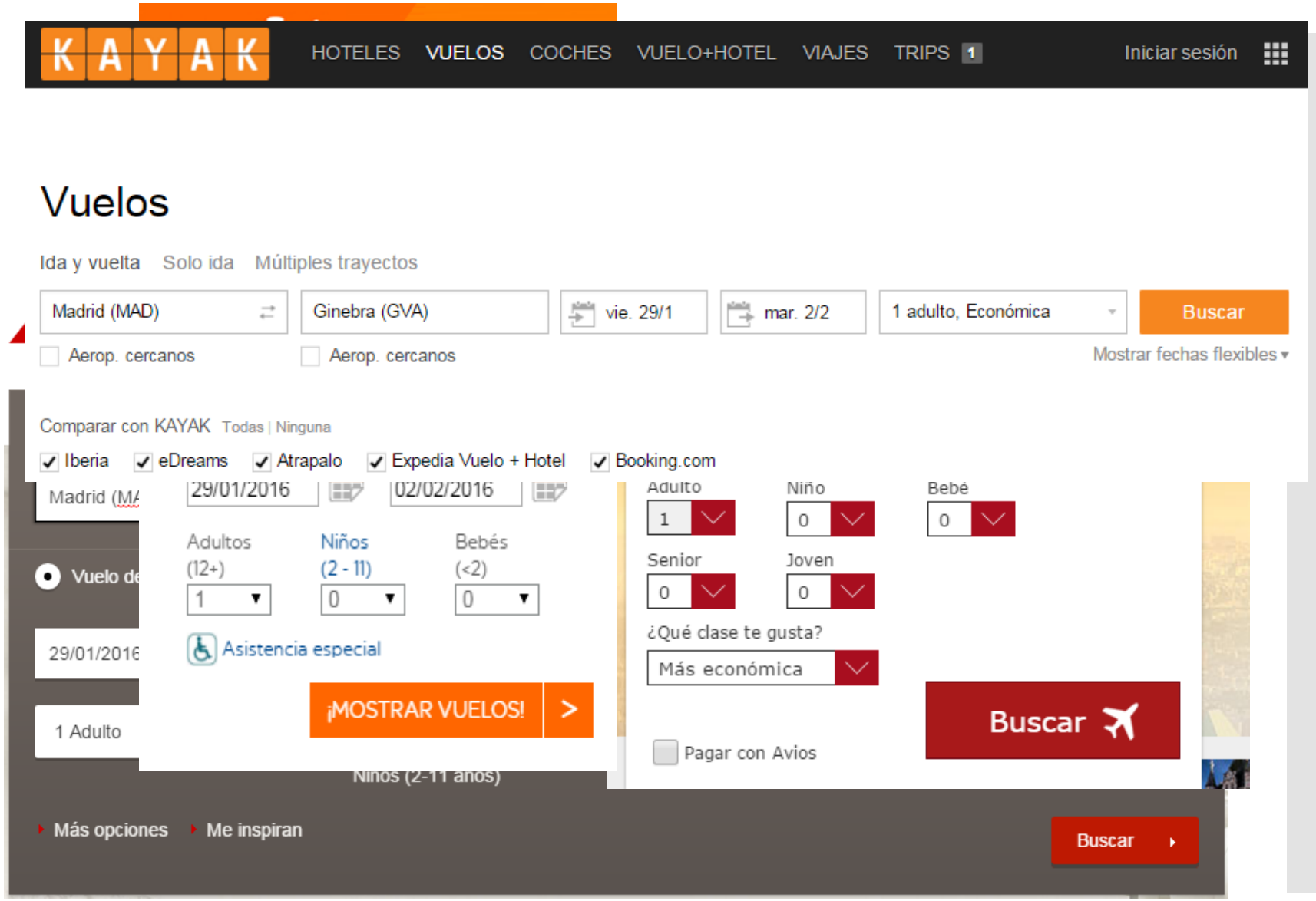
Agenda

- Introduction: Current Web use case
- The Semantic Web and the Web of Data
- Introduction to the Semantic Web technologies:
  - RDF
  - SPARQL
  - Vocabularies
  - Interlinking

Current Web: use case

- We need to travel to Geneva for work
- We will need flight, accommodation, transfers…
- Price comparison
- Check availability and compatibility among different alternatives

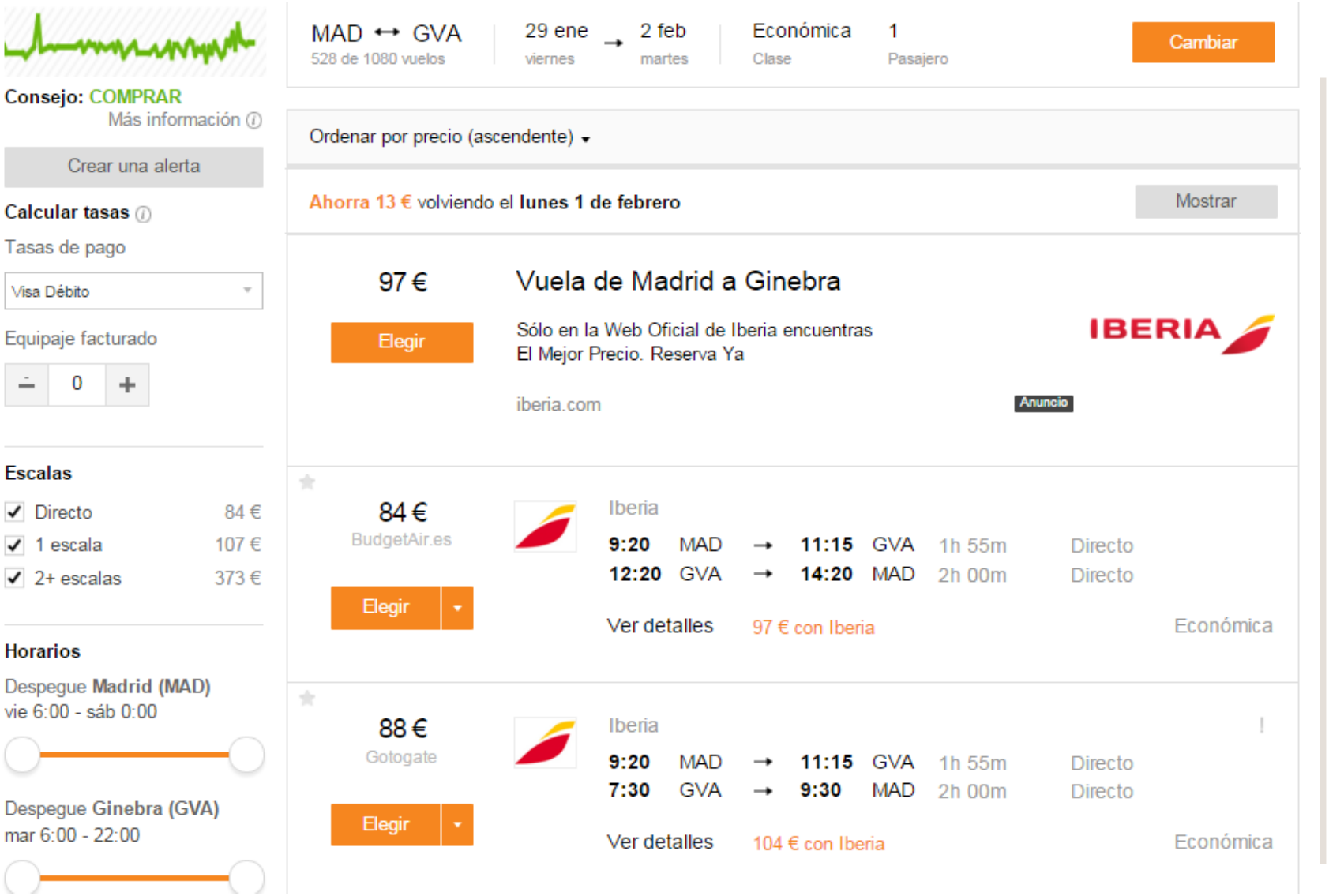
Current web: use case



Current Web: use case

- Summing up: Search flight data in the current Web …
  - Browse different websites
  - Different ways to enter the search data
  - Each page shows different data (date, hours, duration, scales, company, flight number, operated by …)
  - Complemented with different services (luggage, insurance, loyalty programs …).
  - Different format
  - In different languages…

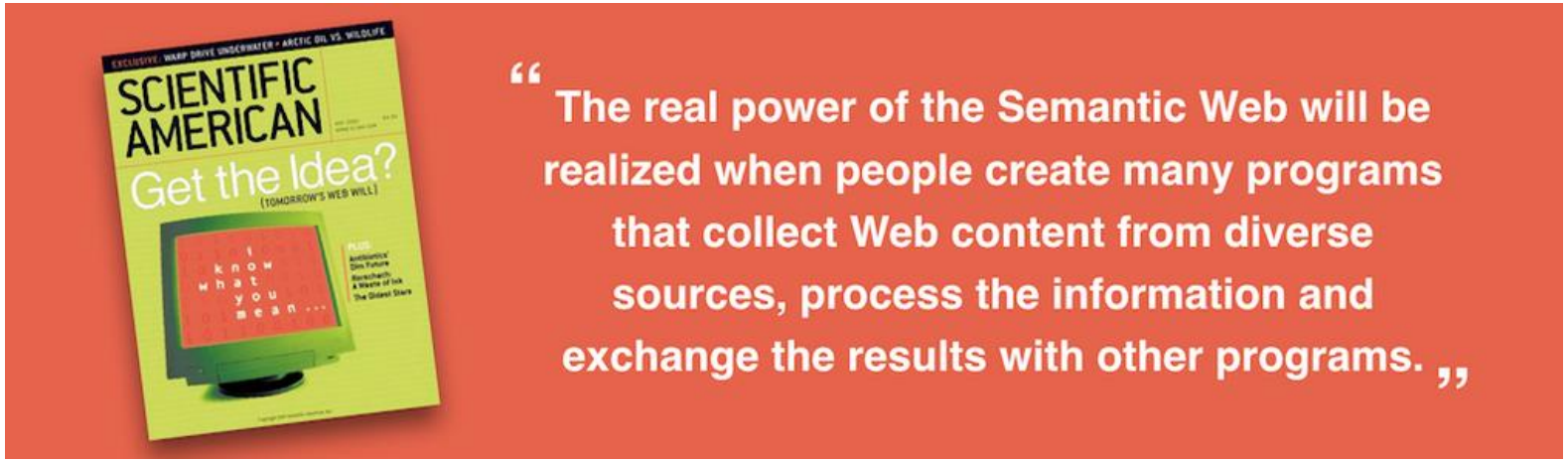
Current Web: use case



Current Web: use case

- Once you have the flight …
  - Search for accommodation (location, price, regime …)
  - Restaurants
  - Car rental
  - Rental of conference rooms, weather conditions …
- For all these, we will face similar issues: time consuming search, different languages and formats, etc.

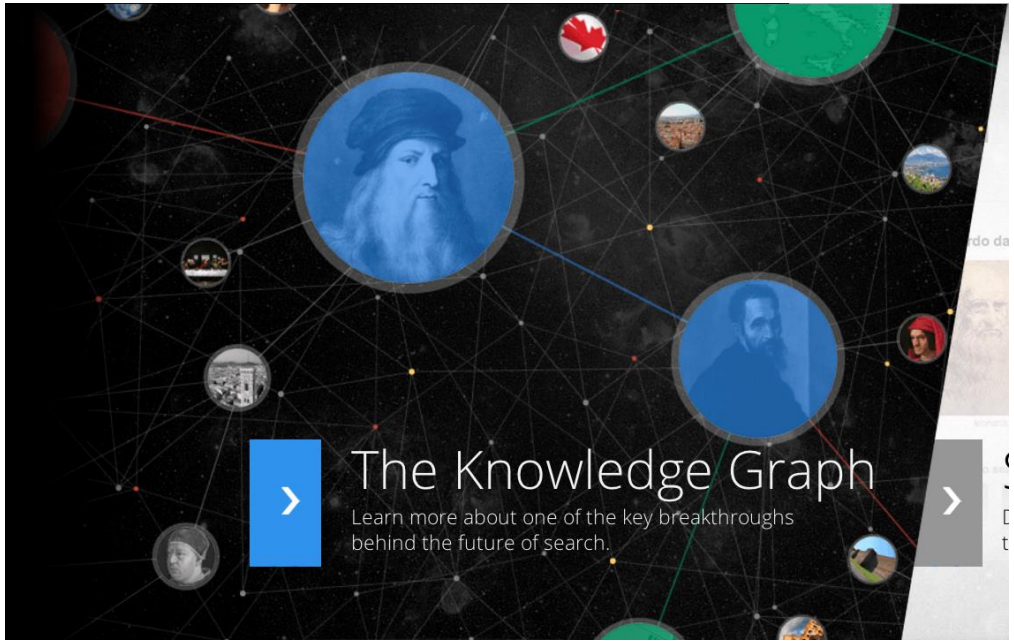
The Semantic Web



Berners Lee's vision (2001, Scientific American)

The Semantic Web today

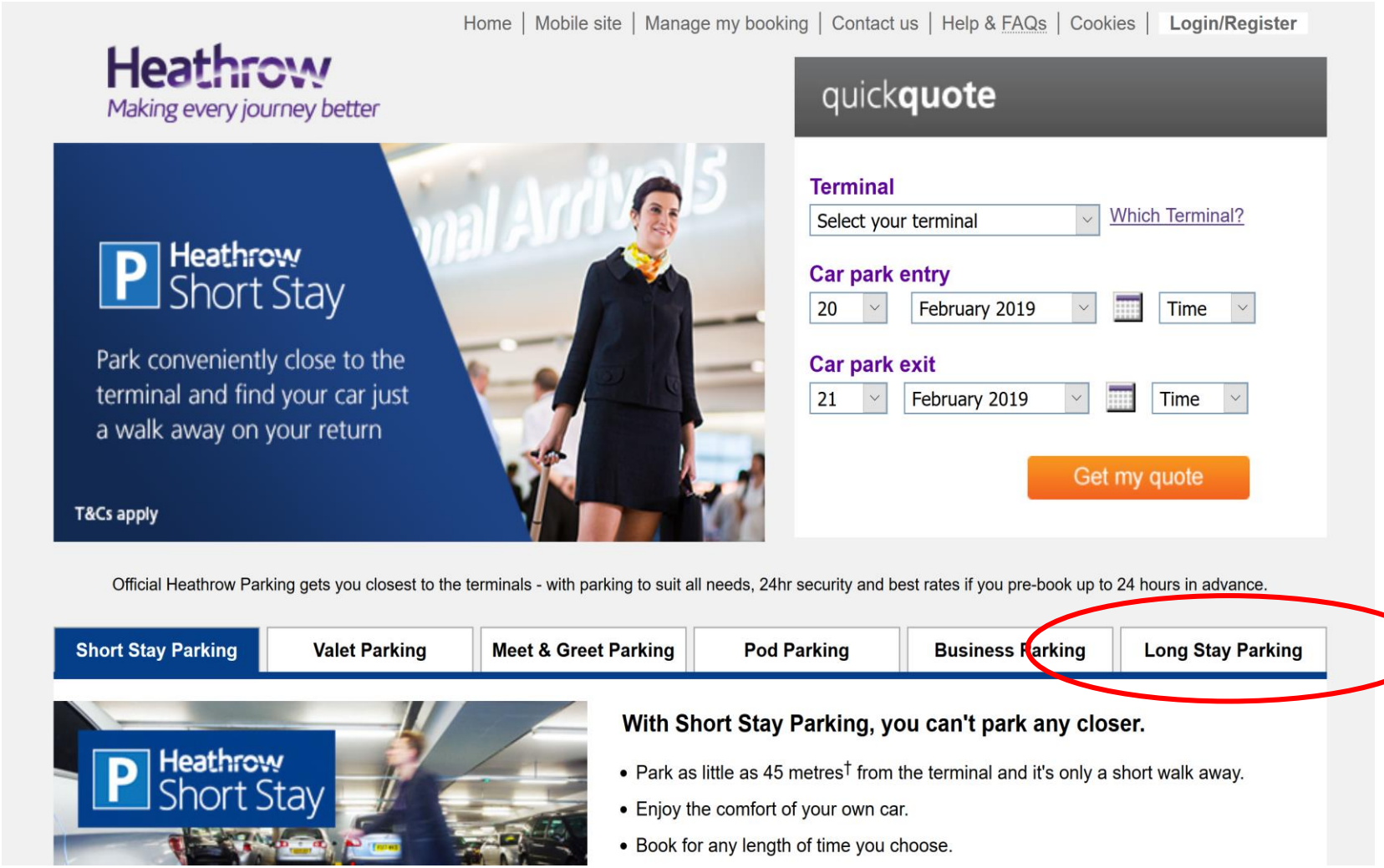
- Ontologies
- Web of data
  - RDF
  - SPARQL
  - Linked open data
  - Vocabularies
- Wikidata



The Web of data

- Web model whose objective is to make structured data directly accessible ("raw data")
  - Not through its inclusion in documents such as HTML pages, PDF reports, etc.
- Concept opposed to the Web of documents
- A new Web of resources, where software applications will become first class "citizens".

Machine oriented Web



Machine oriented Web

A software detects that there is a link to another page, but it does not know what relation it has with the current page nor does it know what the other page is.

Linked data

- Set of good practices to publish and connect data on the Web in a structured way.
- Basic pillars of the model:
  - Representation of information in the form of **RDF triples,** which allows publishing structured data on the Web
  - Use of RDF links with **dereferenceable URIs** to connect data from different sources

http://es.dbpedia.org/resource/UK

Information silos

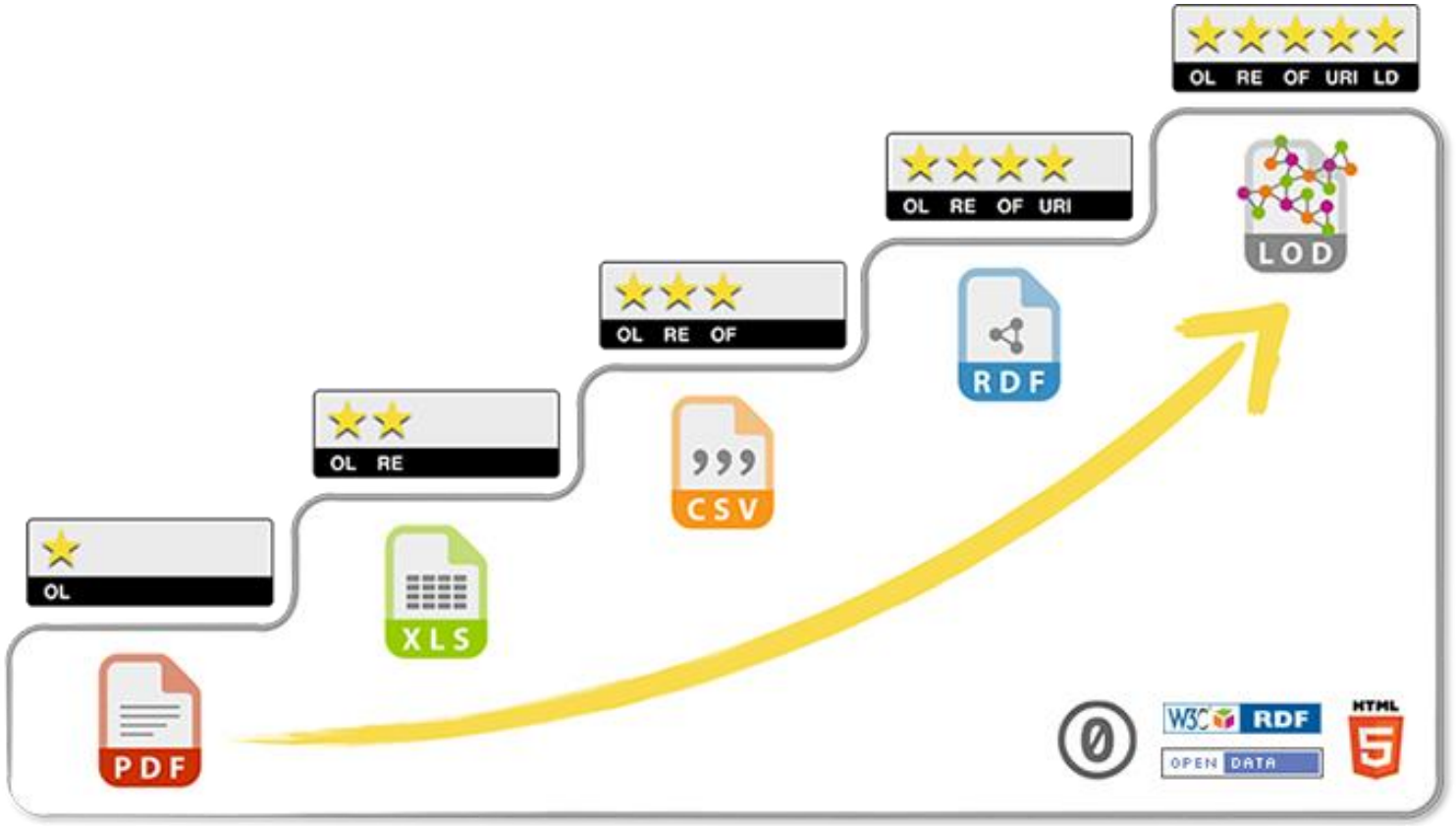
- **SILO**: Information management system unable to communicate with other systems.
- Isolated because…
  - Information owners or managers don’t perceive enough benefits in sharing the information it stores
  - It is thought that the information may not be useful for other systems or users.
  - Other causes

Open data

- Movement/Philosophy aimed at making available certain types of data free and without restrictions for everyone (no copyright, no patents, etc.)
- Open data + use of the Linked Data model = **linked open data** (LOD)
- Recent efforts of different countries to promote free access to data (governments, universities and libraries leading)

5 stars model

5 stars model



http://5stardata.info/

Linked data model principles

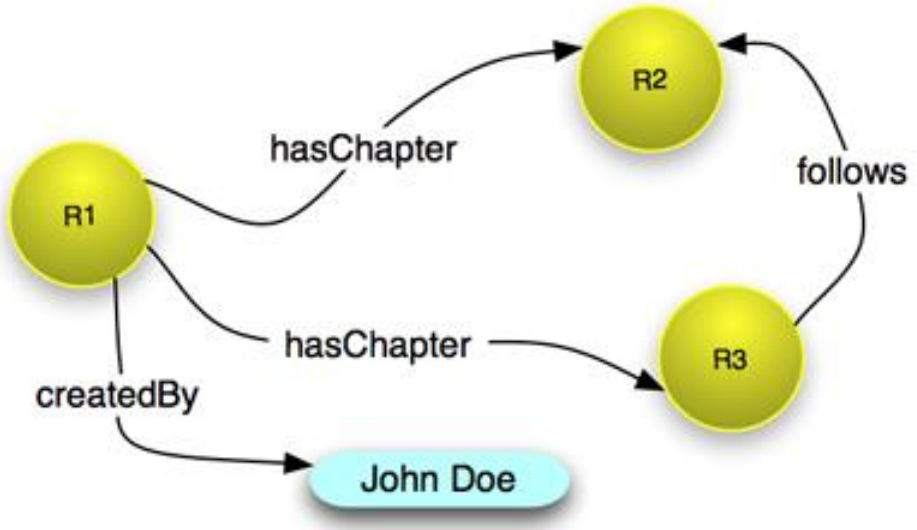
- Use URIs to identify anything
- Make use of dereferenceable URIs for both people and software agents can search, find and retrieve resources.
- Provide useful information about resources, making use of standards (RDF, vocabularies, SPARQL endpoints).
- Include links to other (sometimes external) URIs to facilitate the discovery of new resources.

Content negotiation

- HTTP mechanism that allows to serve different versions of a given document using the same URI
- The user side determines which version is more appropriate for the current needs
- Accesing this URI http://es.dbpedia.org/resource/Madrid
  - Will be served as http://dbpedia.org/data/Madrid.rdf (machine)
  - Or as: http://dbpedia.org/page/Madrid (human)

RDF

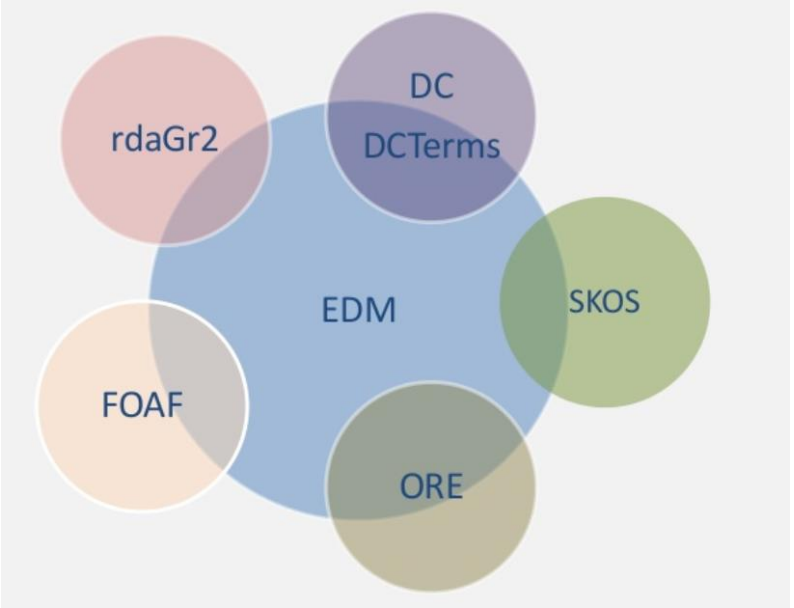
Standard model to represent knowledge on the Web of Data



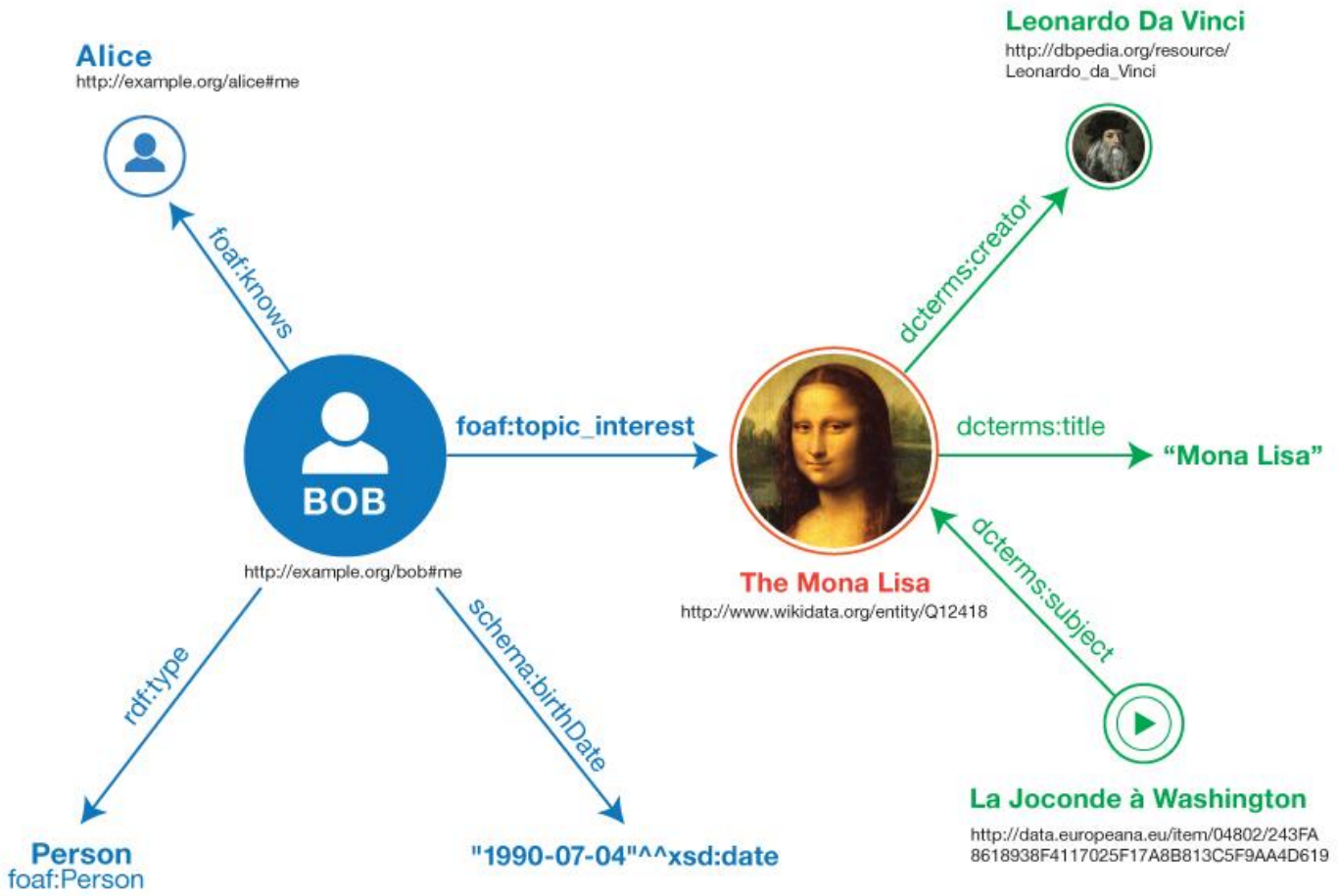
Subject	Predicate	Object
R1	hasChapter	R2
R1	hasChapter	R3
R3	follows	R2
R1	createdBy	"John Doe"

Vocabularies

- Many properties are common over many domains
- It is advisable to use vocabularies didicated to describe specific behaviours or types of relationships
- Defining our own terms makes sense only if existing vocabularies do not cover the terms requested



RDF


Alice
http://example.org/alice#me
foaf:knows
BOB
http://example.org/bob#me
foaf:topic_interest
rdf:type
Person
foaf:Person
schema:birthDate
"1990-07-04"^^xsd:date
Leonardo Da Vinci
http://dbpedia.org/resource/
Leonardo_da_Vinci
dcterms:creator
dcterms:title
"Mona Lisa"
The Mona Lisa
http://www.wikidata.org/entity/Q12418
dcterms:subject
La Joconde à Washington
http://data.europeana.eu/item/04802/243FA
8618938F4117025F17A8B813C5F9AA4D619

RDF

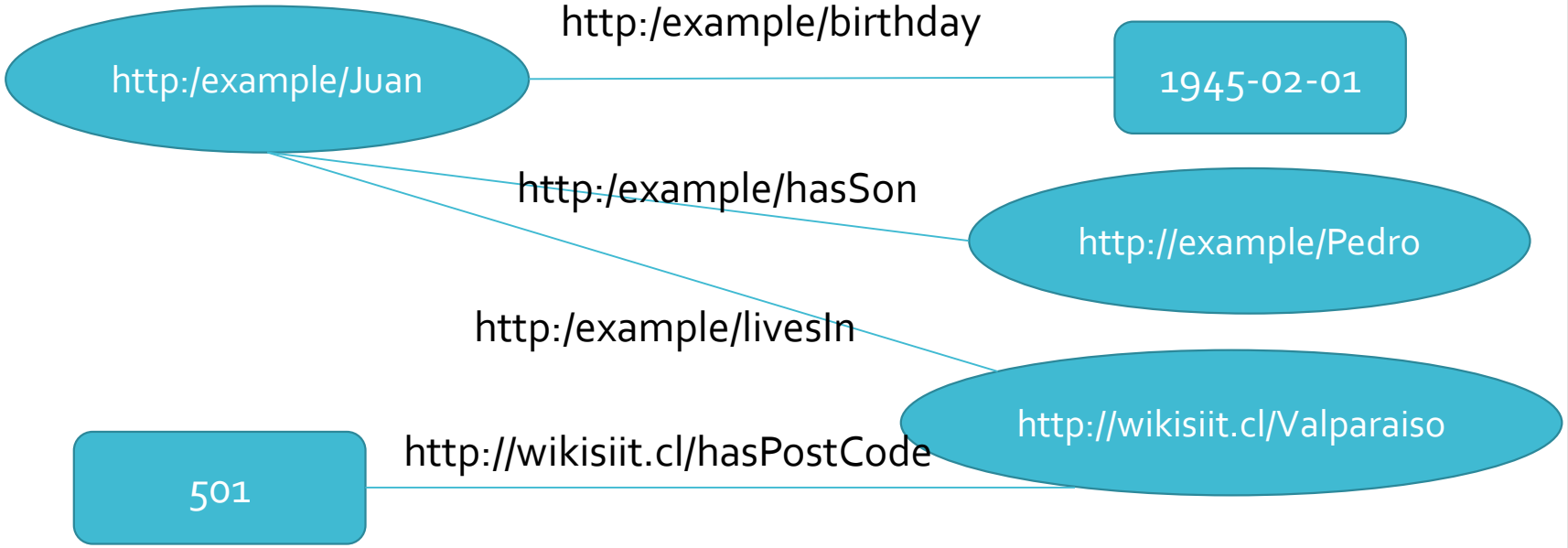
[subject, predicate, object]

[Bob **knows** Alice]

In RDF:

```
<http://Bob.example.org/bob#me>
        <http://xmlns.com/foaf/0.1/knows>
                    <http://Bob.example.org/alice#me>
```

RDF: Example


http:/example/Juan
http:/example/birthday
1945-02-01
http:/example/hasSon
http://example/Pedro
http:/example/livesIn
http://wikisiit.cl/Valparaiso
501
http://wikisiit.cl/hasPostCode

Serialization

- Graphs are represented in documents using a serialization syntax: RDF/XML, N3, Turtle, N Triples…
- RDF/XML:
  - Element `rdf:RDF` encapsulates the whole document
  - `rdf:Description` encapsulates resource descriptions
  - New elements inside `rdf:Description` for properties

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.1/">
  <rdf:Description rdf:about="http://en.wikipedia.org/wiki/Tony_Benn">
    <dc:title>Tony Benn</dc:title>
    <dc:publisher>Wikipedia</dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

Notation3 (N3)

- Human friendly, easy to read, serialization
- Full expressive power

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.

<http://en.wikipedia.org/wiki/Tony_Benn>
  dc:title "Tony Benn";
  dc:publisher "Wikipedia".
```

N-Triples

- Based on lines (one line, one triple)
- Subset of N3 to produce results of e.g. search operations
- Prefix-less, oriented to process thousands (or millions) or triples (easy to parse)

```
<http://www.w3.org/2001/sw/RDFCore/ntriples/> <http://purl.org/dc/elements/1.1/creator> "Dave Beckett" .
<http://www.w3.org/2001/sw/RDFCore/ntriples/> <http://purl.org/dc/elements/1.1/creator> "Art Barstow" .
<http://www.w3.org/2001/sw/RDFCore/ntriples/> <http://purl.org/dc/elements/1.1/publisher> <http://www.w3.org/> .
```

Turtle

- Simplified RDF representation
- no-XML
- More compact and legible, aimed at humans

```
@prefix myNamespace: <http://example/> .
@prefix wikisiit: <http://wikisiit.cl/> .


myNamespace:book1   myNamespace:nacimiento   1945-02-01 ;
                    myNamespace:tieneHijo   myNamespace:Pedro ;
                    myNamespace:viveEn   http://wikisiit.cl/Valparaiso .

http://wikisiit.cl/Valparaiso   wikisiit:tieneCut   501.
```

RDF: Namespaces

- Namespaces allow to simplify xml documents
- Example:
  - Defining the prefix "xmlns:rdf"

    xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  - Will later allow to use `rdf:property` instead of:

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

RDF: Prefixes (Turtle)

- When using vocabularies, we need either use the full URI of items, like this:

```
<http://Angela.ejemplo.org>
    a   <http://xmlns.com/foaf/0.1/Person> ;
    <http://xmlns.com/foaf/0.1/name> "Angela Heifetz"
```

- Alternatively, we can declare and use a prefix:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
<http://Angela.ejemplo.org>
        a   foaf:Person ;
        foaf:name "Angela Heifetz" .
```

Publishing the data

- **Prepare** the data.
  - Likely to imply a transformation to RDF
- **Store** them in a database (relational, RDF native…)
- **Publication** and provision of access to third parties
- … then the data can be queried / accessed

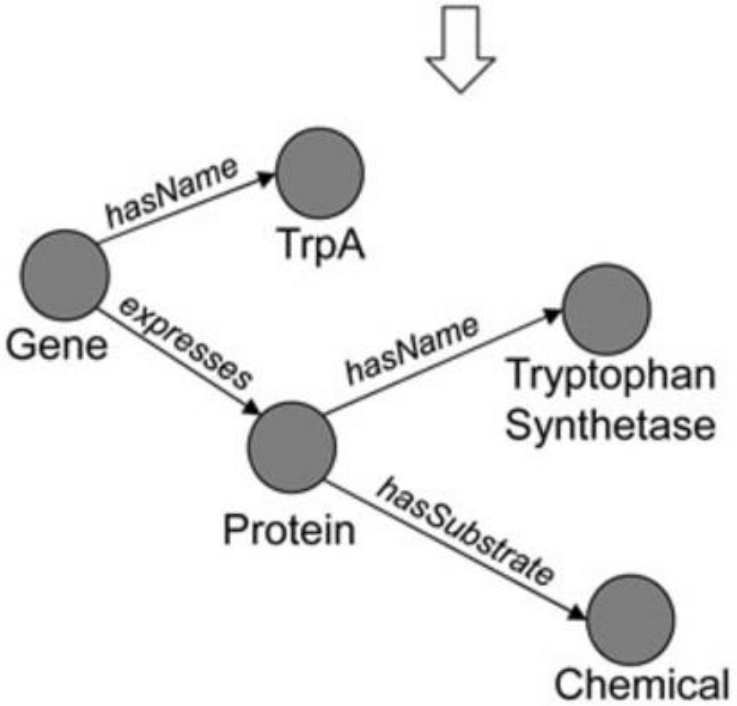
SPARQL

- Query language specifically oriented to operate on triples-oriented databases (RDF graphs)
- Considered one of the key technologies in the **Semantic Web**
- Before querying an RDF graph, a **SPARQL endpoint must exist.**
- **SparQL queries match** a query pattern with the stored knowledge and return the results.

SPARQL example

Triples in an RDF database

ID	subject	predicate	object
1	<Gene>	<hasName>	TrpA
2	<Gene>	<expresses>	<Protein>
3	<Protein>	<hasName>	Tryptophan Synthetase
4	<Protein>	<hasSubstrate>	<Chemical>



<RDF graph>

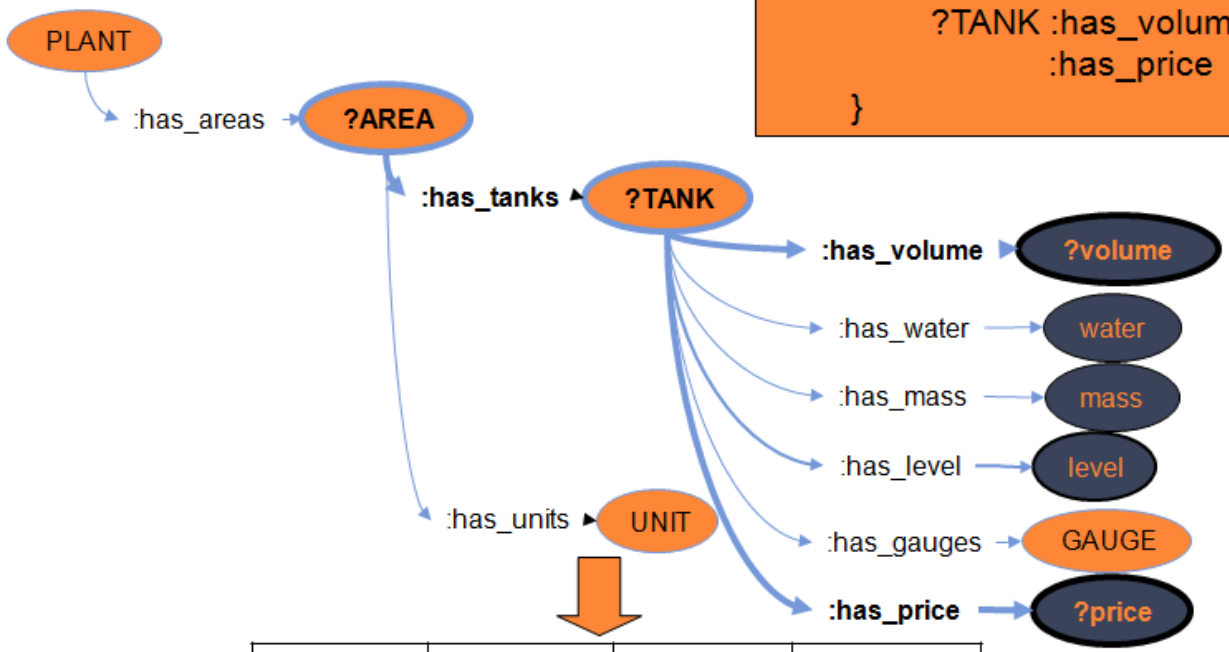
```
SELECT  ?x  WHERE
{
  ?x <hasName> "Tryptophan Synthetase".
  ?x <hasSubstrate> <Chemical>
}
```

<An example of SPARQL>

⇨ Answer: **<Protein>**

SPARQL example

```
SELECT ?AREA ?TANK ?volume ?price
WHERE {
      ?AREA :has_tanks    ?TANK .
      ?TANK :has_volume ?volume ;
            :has_price    ?price
}
```



AREA	TANK	volume	price
:AREA1	:TANK101	42^^:ft3	56^^:$
:AREA1	:TANK102	86^^:m3	83^^:$

SPARQL query pattern

- A SparQL query compares a pattern (**query pattern**) with the knowledge in the graph.
- **Triple pattern**: RDF incomplete triple, missing one or more elements (variables).

Tripleta RDF:

sujeto	predicado	objeto
Maria	esAmigoDe	Juan

Patrón de tripleta:

sujeto	predicado	objeto
Maria	esAmigoDe	?x

SPARQL patterns

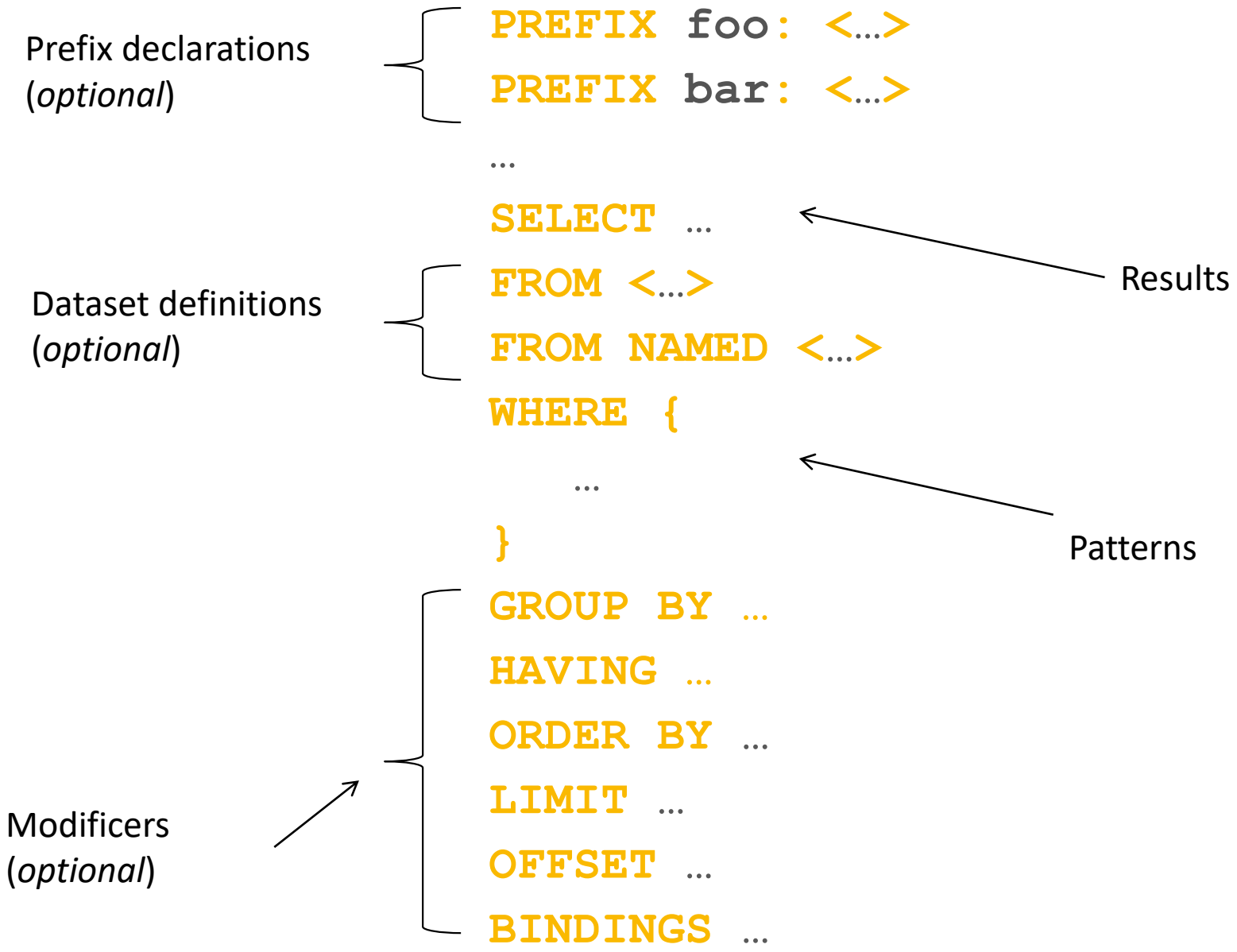
- Patterns can be simple or groups (separated by ".")

```
?x  hasName  ?name .
?x  age      ?age .
```

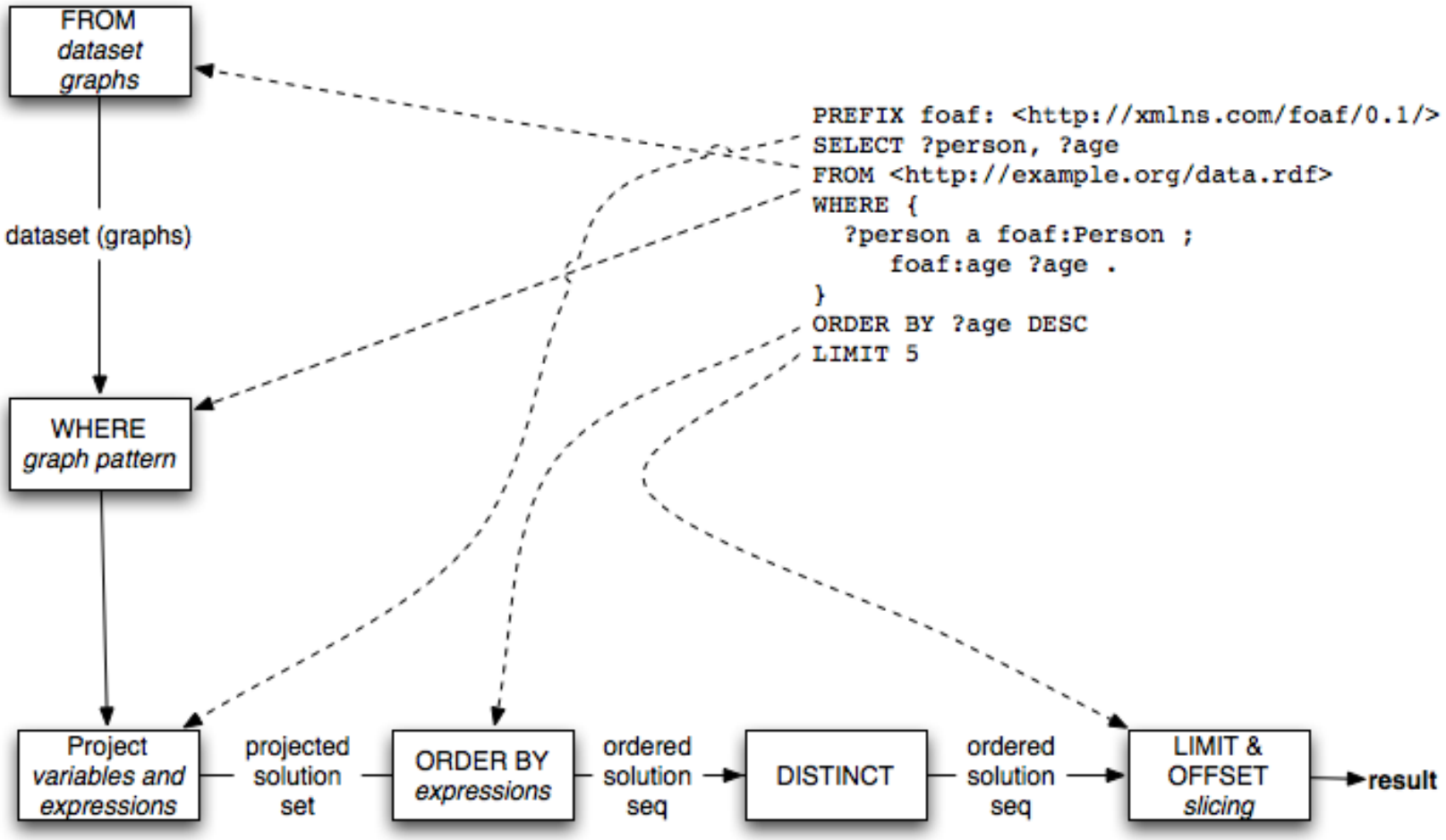
- If the subject is common, this can be shortened using ";"

```
?x  hasName  ?name ;
    age      ?age .
```

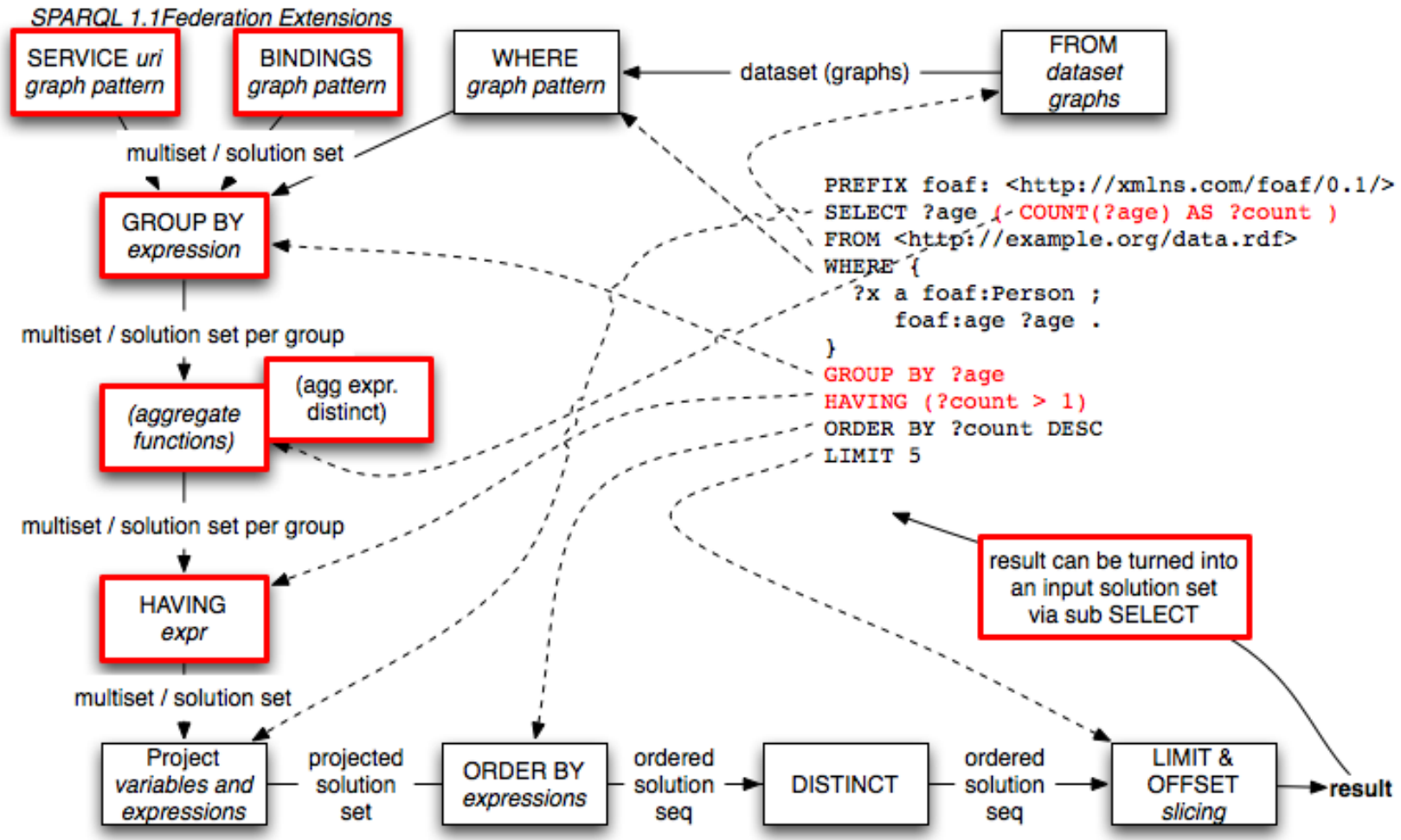
SPARQL: Query structure



SPARQL 1.0: SELECT syntax



SPARQL 1.1: SELECT syntax



SPARQL: Basic elements

URIs

Full URI:

```
<http://this.is.a/full/URI/written#out>
```

Prefix shortened URIs :

```
PREFIX
ex:<http://this.is.a/full/URI/written#>

ex:out
```

Variables

Variables

```
?var1, ?anotherVar, ?and_one_more
```

Triple patterns

Exactly 1 triple:

```
ex:myWidget ex:partNumber "XY24Z1" .
```

Matches one triple:

```
?person foaf:name "Lee Feigenbaum" .
```

Matches many triples:

```
conf:SemTech2009 ?property ?value .
```

Literals

Plain literal:

```
"a plain literal"
```

Plain literal with language tag:

```
"bonjour"@fr
```

Typed literal:

```
"13"^^xsd:integer
```

Comments

```
# Comments start with a '#'
# continue to the end of the
line
```

SPARQL commands

- 4 different commands:
  - **SELECT**: Returns all (or a subset of) variables that match a given search pattern.
  - **CONSTRUCT**: Returns a graph built by substituting variables in a set of triple patterns.
    - Used to transform information, e.g. from foaf to vcard
  - **ASK**: Returns true or false depending on whether there is a match for the given pattern
  - **DESCRIBE**: Describes through a graph the resources found (all triples associated to them, basically).

SPARQL Prefixes and base

The same URI in 3 different equivalent forms:

```
(1)
<http://example.org/book/book1>

(2)
BASE <http://example.org/book/>
<book1>

(3)
PREFIX book: <http://example.org/book/>
book:book1
```

SPARQL exercise

- Access The British National Bibliography SPARQL endpoint here: http://bnb.data.bl.uk/flint-sparql
- List all the information recorded for the writer Peter Johnson:

```
http://bnb.data.bl.uk/id/person/JohnsonPeter
```

SPARQL exercise

SOLUTION(s):

```
SELECT ?p ?o WHERE {
 <http://bnb.data.bl.uk/id/person/JohnsonPeter> ?p ?o
}
```

or

```
DESCRIBE <http://bnb.data.bl.uk/id/person/JohnsonPeter>
```

SPARQL modifiers

- **LIMIT <n>**: Max number of results to display
- **OFFSET <n>**: Discards the first n results

SPARQL modifiers exercise

List 5 books written by Peter Johnson

Hint: use the predicate `dct:creator`

Sintaxis SPARQL

SOLUTION:

```
PREFIX dct: <http://purl.org/dc/terms/>
SELECT ?book WHERE {
   ?book dct:creator
          <http://bnb.data.bl.uk/id/person/JohnsonPeter>
}
LIMIT 5
```

SPARQL modifiers exercise

List the **titles of 5 books** written by Peter Johnson

Hint: use the predicate `dct:title`

SPARQL exercise solution

SOLUTION:

```
PREFIX dct: <http://purl.org/dc/terms/>
SELECT ?title WHERE {
   ?book dct:creator <http://bnb.data.bl.uk/id/person/JohnsonPeter> ;
         dct:title ?title.
}
LIMIT 5
```

SPARQL: Filters

- If we want to apply restrictions, filters must be used.
- **FILTER** : applies a restriction to a query
- **REGEX** <expression> allows to evaluate regular expressions on string literals

SPARQL: Filters examples

- FILTER (?price < 30.5)
- FILTER regex(?title, "^SPARQL")
- FILTER regex(str(?mailbox), "@work.example")
- FILTER (?date > "2005-01-01T00:00:00Z"^^xsd:dateTime)
- FILTER (xsd:integer(?population) > 1000)

SPARQL: Filters example

- Display 15 books whose title starts with “Spain”

SPARQL: Filters exercise solution

SOLUTION:

```
PREFIX dct: <http://purl.org/dc/terms/>
SELECT ?title WHERE {
   ?book
      dct:title ?title;
      FILTER (regex (?title, "^Spain")).
}
LIMIT 15
```

Interlinking

- Linked data model makes full sense when data not previously related are linked
- "Break the silo, free the data!"
- Interlinking means finding structured and persisting information describing entities related to our data

```
<Resource1, owl:sameAs, Resource2>
```

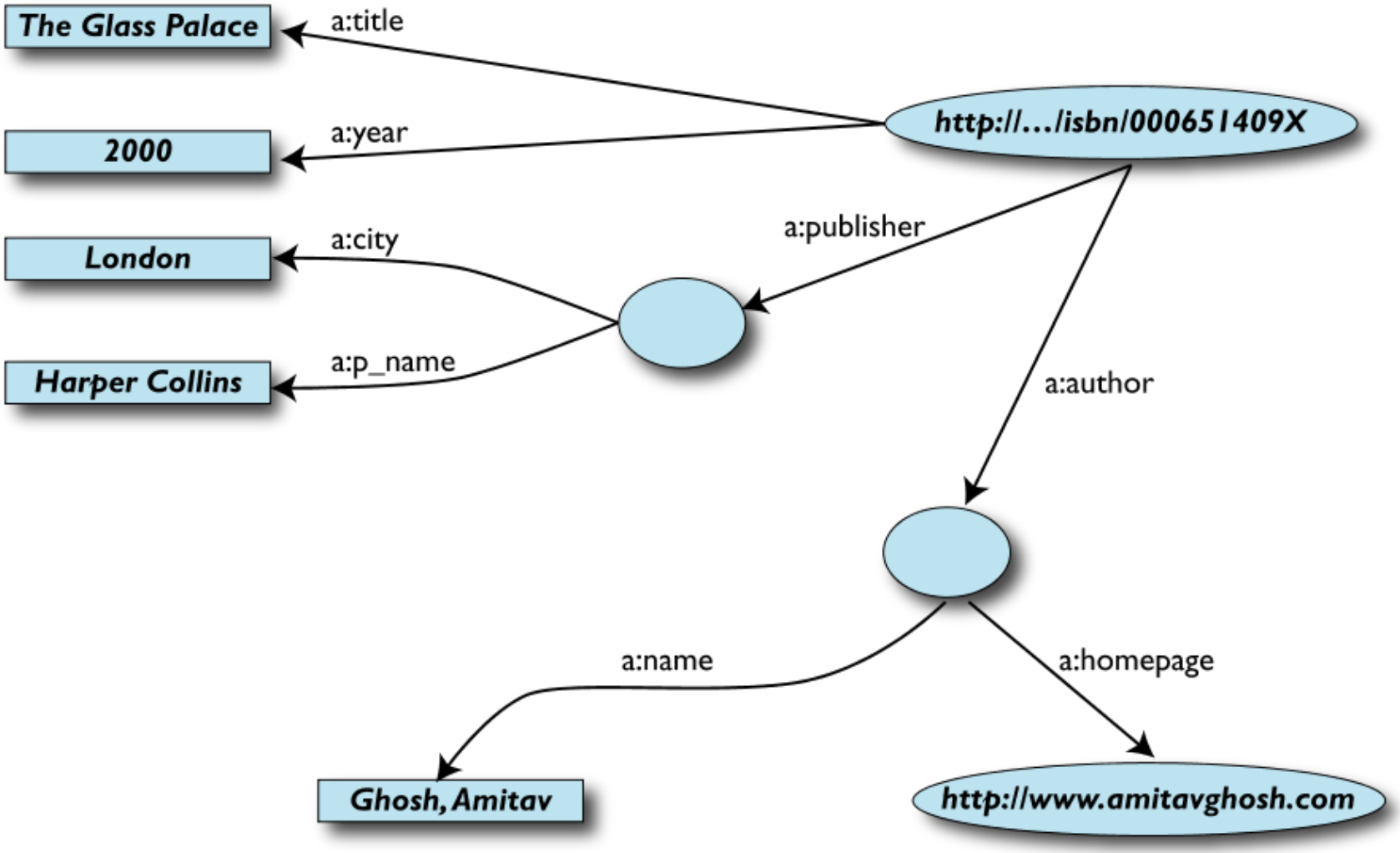
Simplified data from library A

ID	Author	Title	Publisher	Year
ISBN0-00-651409-X	id_xyz	The Glass Palace	id_qpr	2000

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

ID	Publ. Name	City
id_qpr	Harper Collins	London

Step 1: data modeling



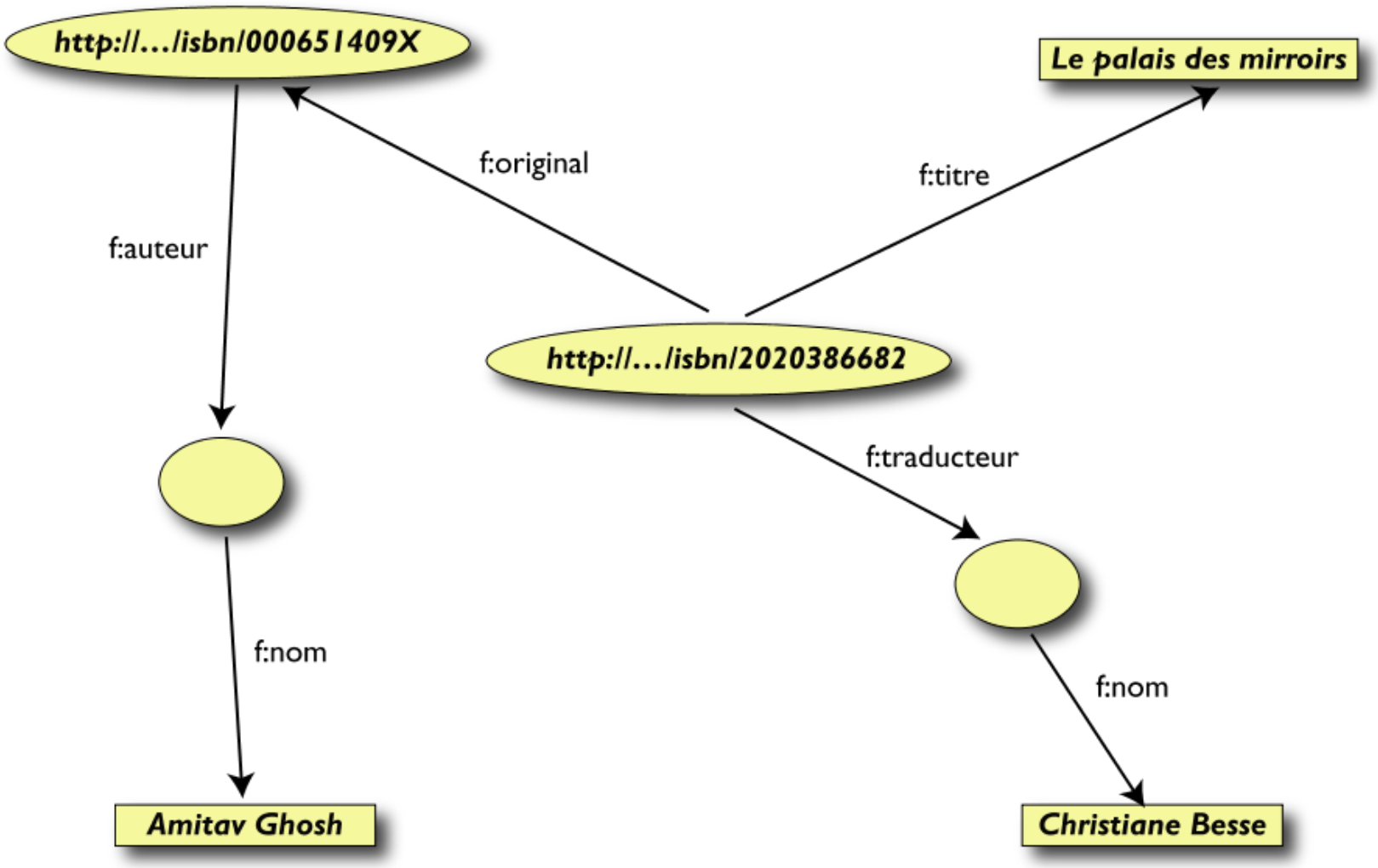
Simplified data from library F

	A	B	D	E
1	**ID**	**Titre**	**Traducteur**	**Original**
2	ISBN0 2020386682	Le Palais des miroirs	A13	ISBN-0-00-651409-X
3				

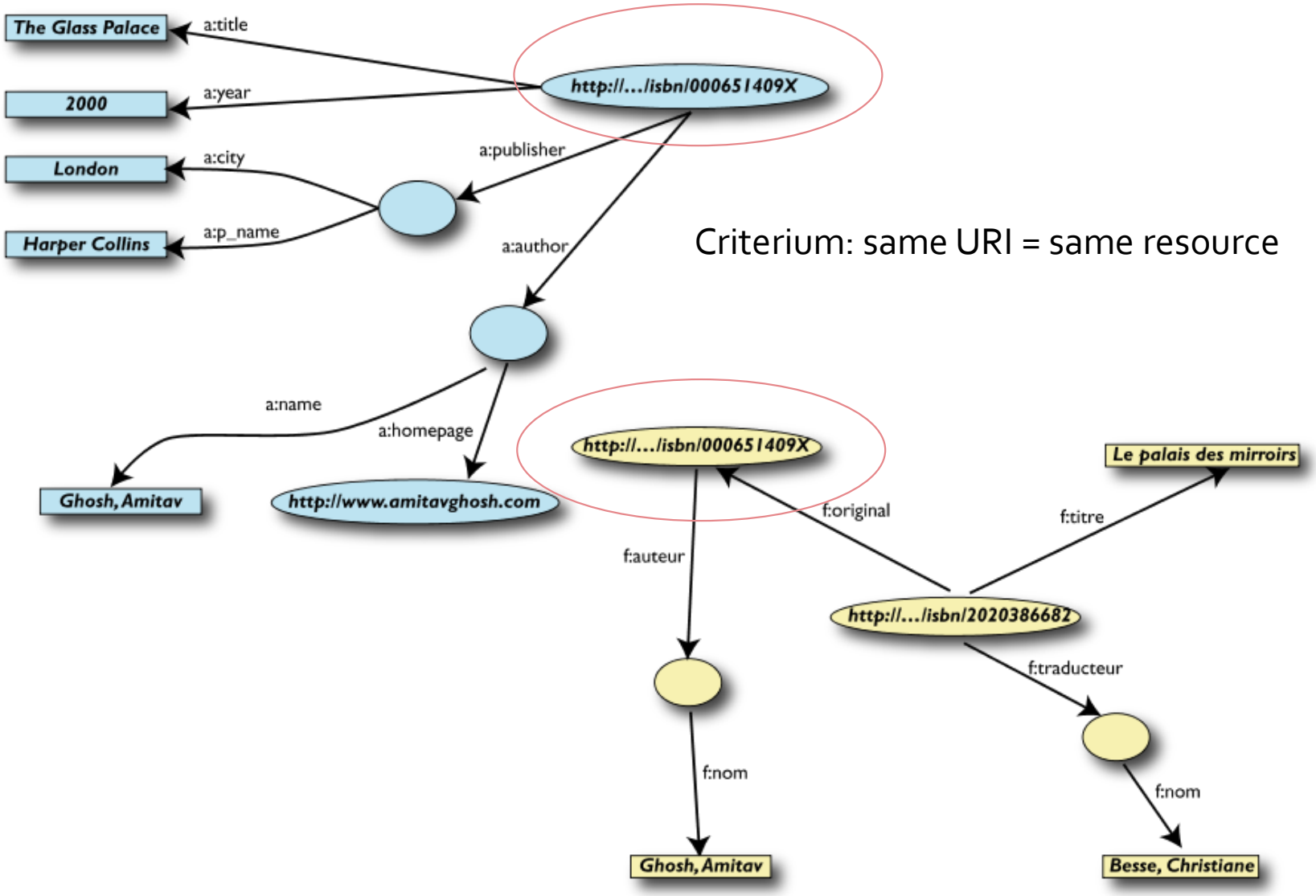
6	**ID**	**Auteur**
7	ISBN-0-00-651409-X	A12

11	**Nom**
12	Ghosh, Amitav
13	Besse, Christianne

Step 1b: data modeling

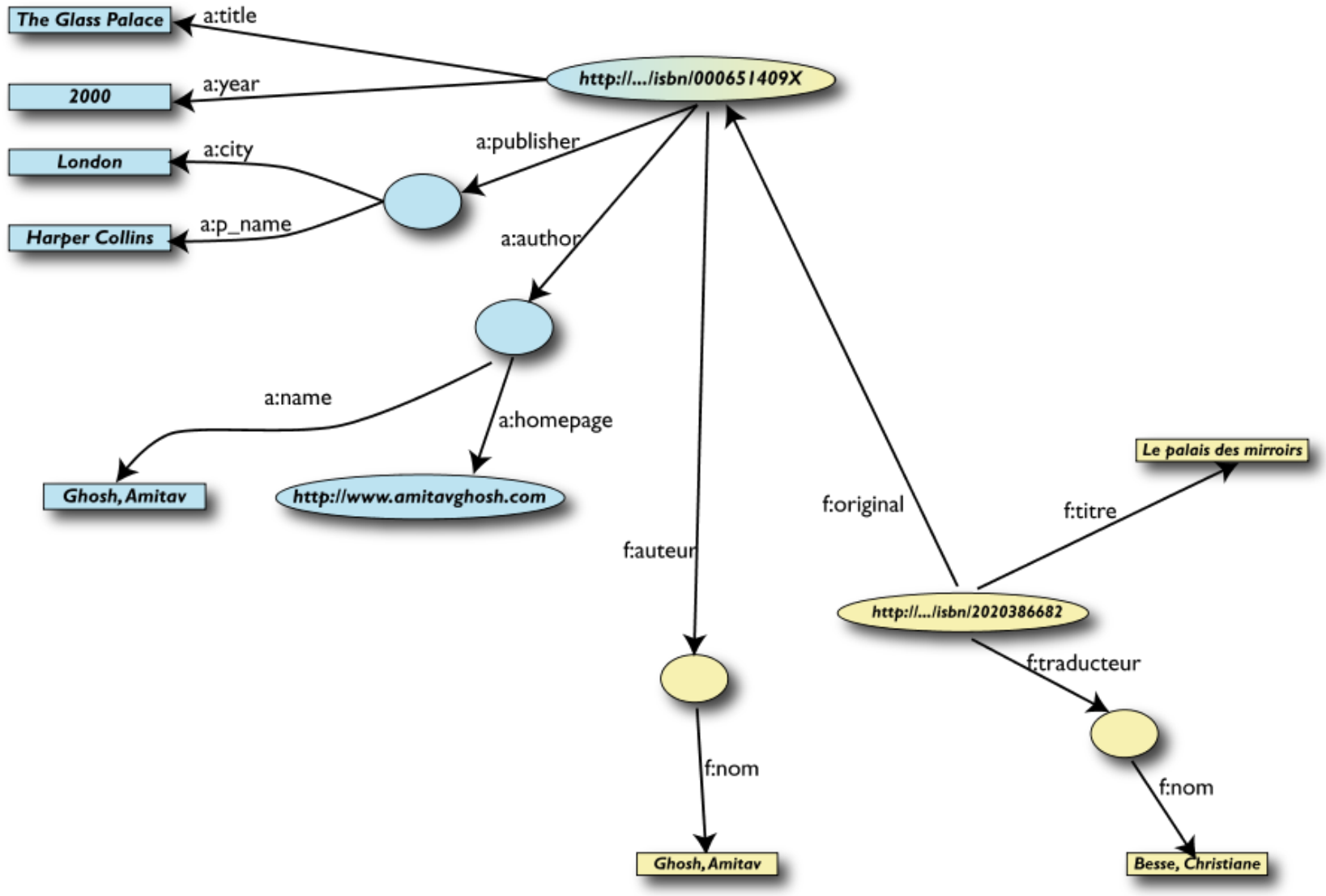


Step 2: merging datasets

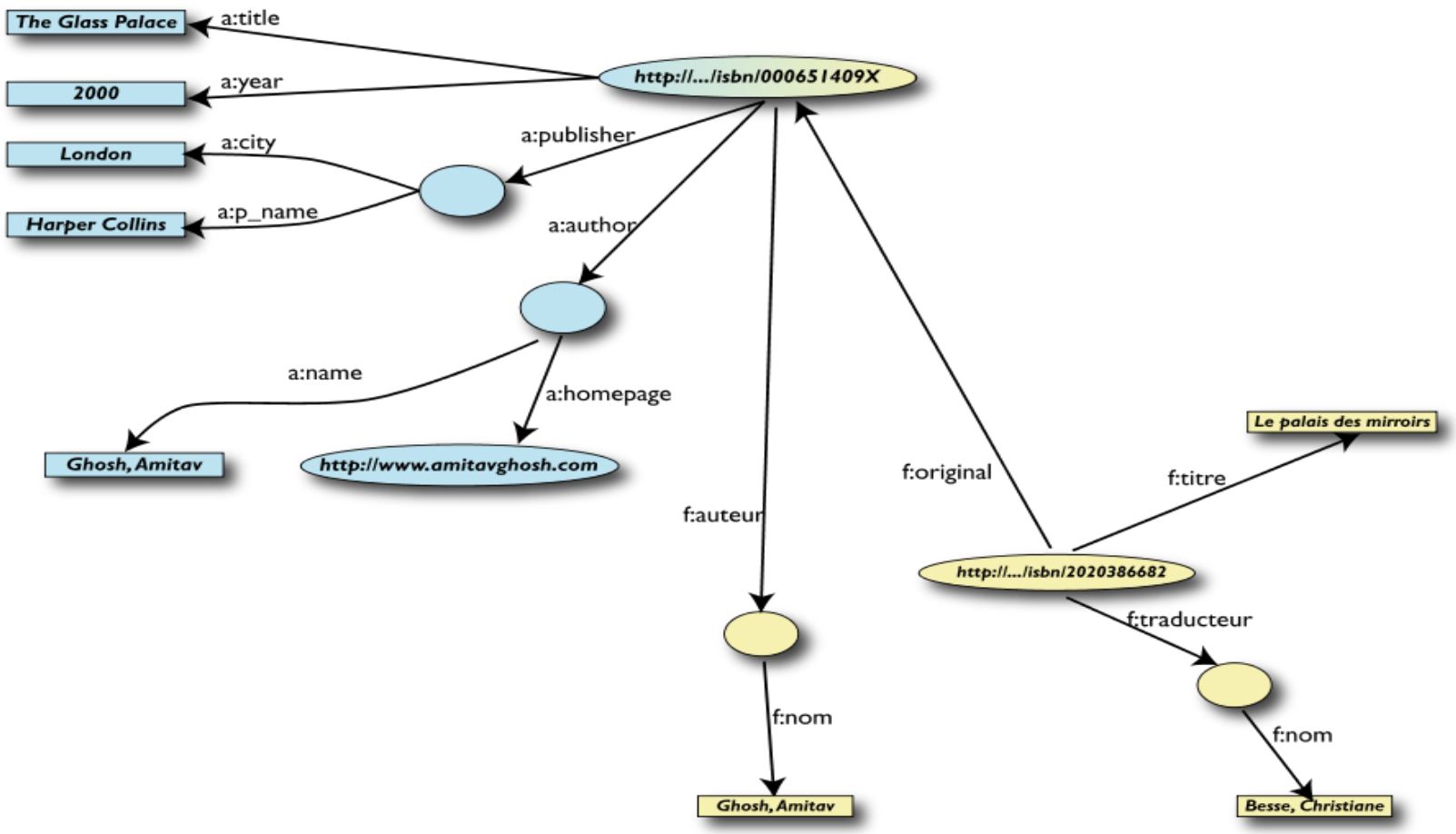


Criterium: same URI = same resource

Step 2: merging datasets

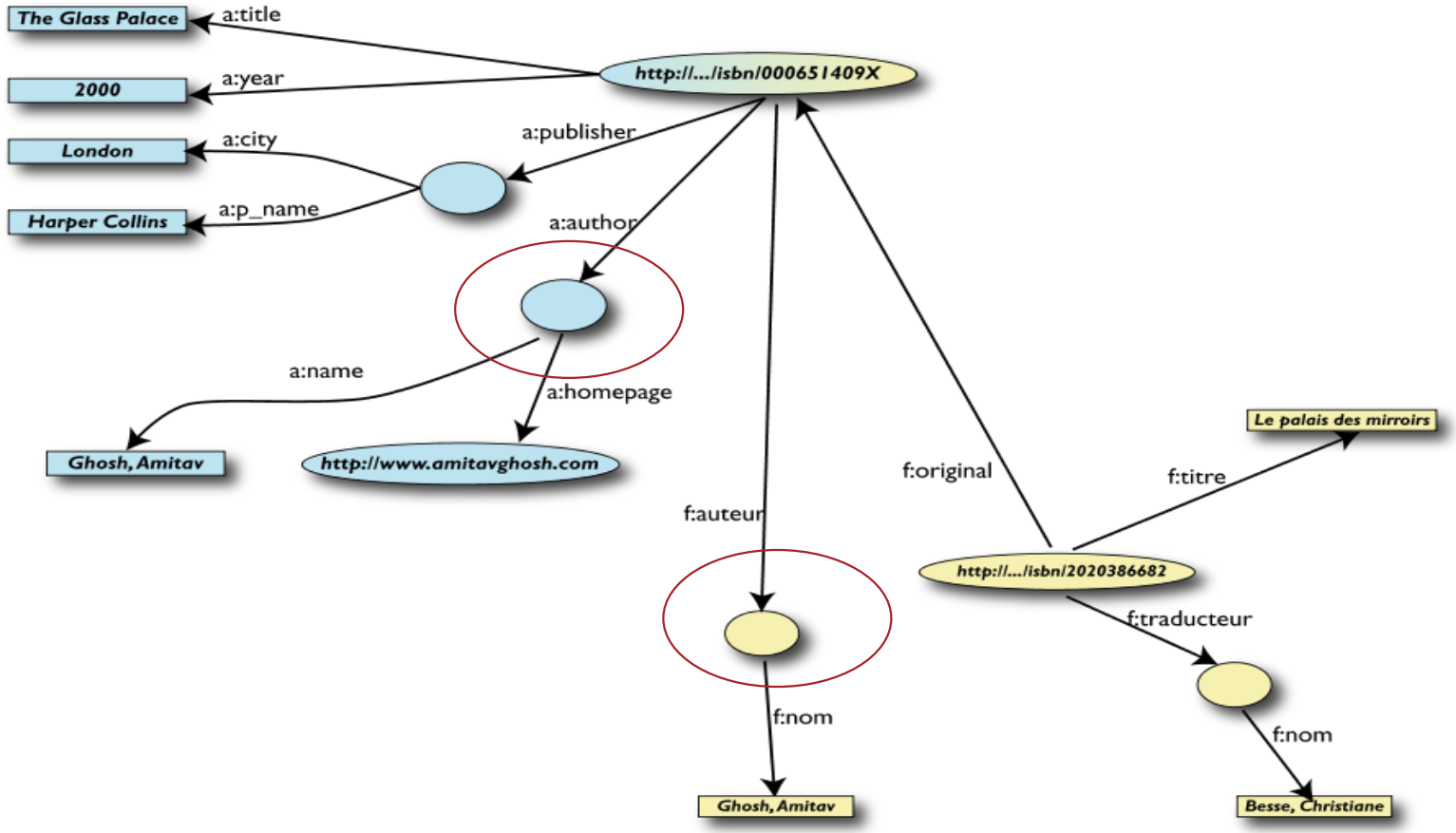


New queries



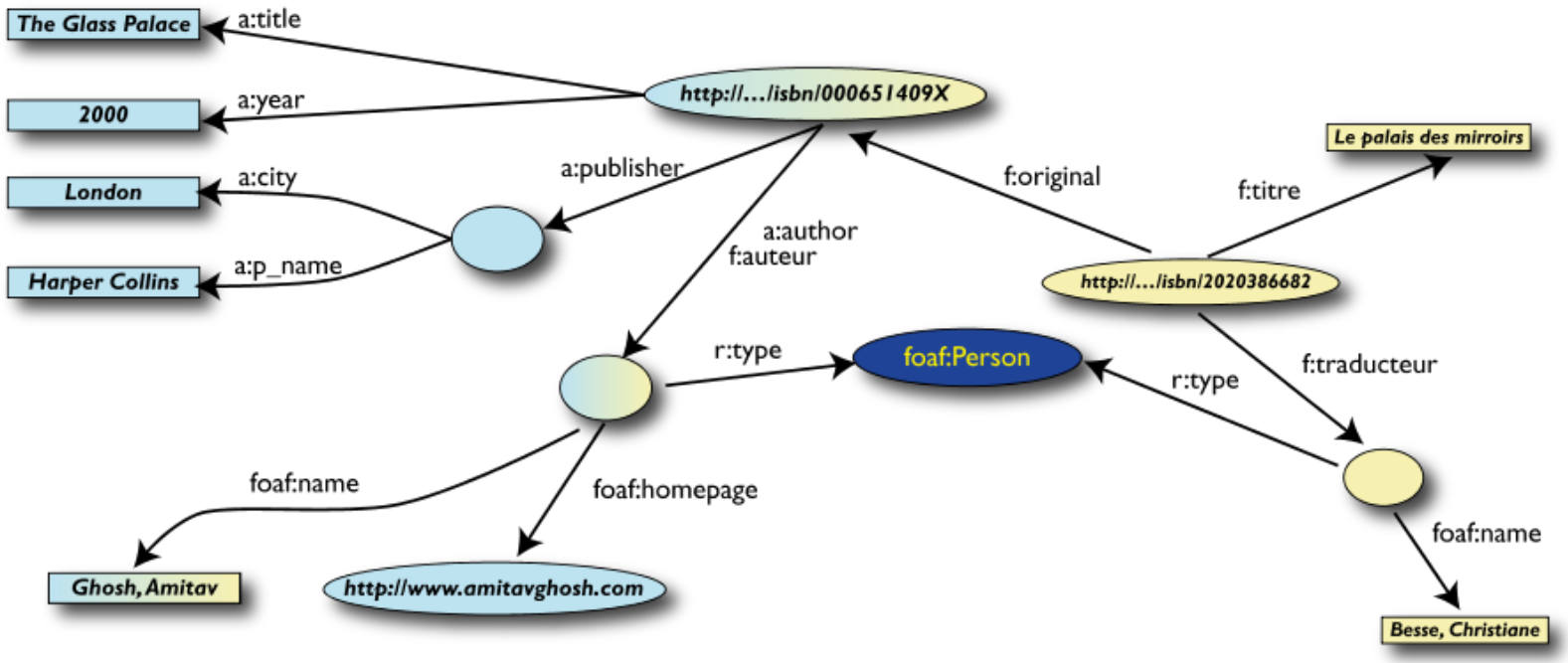
- Users of F can now query their dataset for new information
  "give me the title of the original book"

Extra: data integration

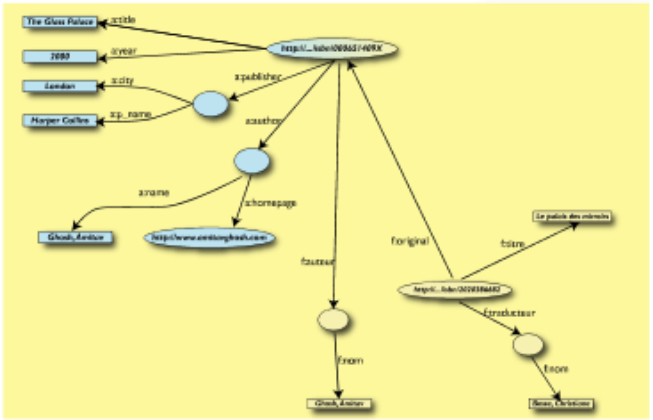


- Identification of repeated information
- Identification of clases of elements

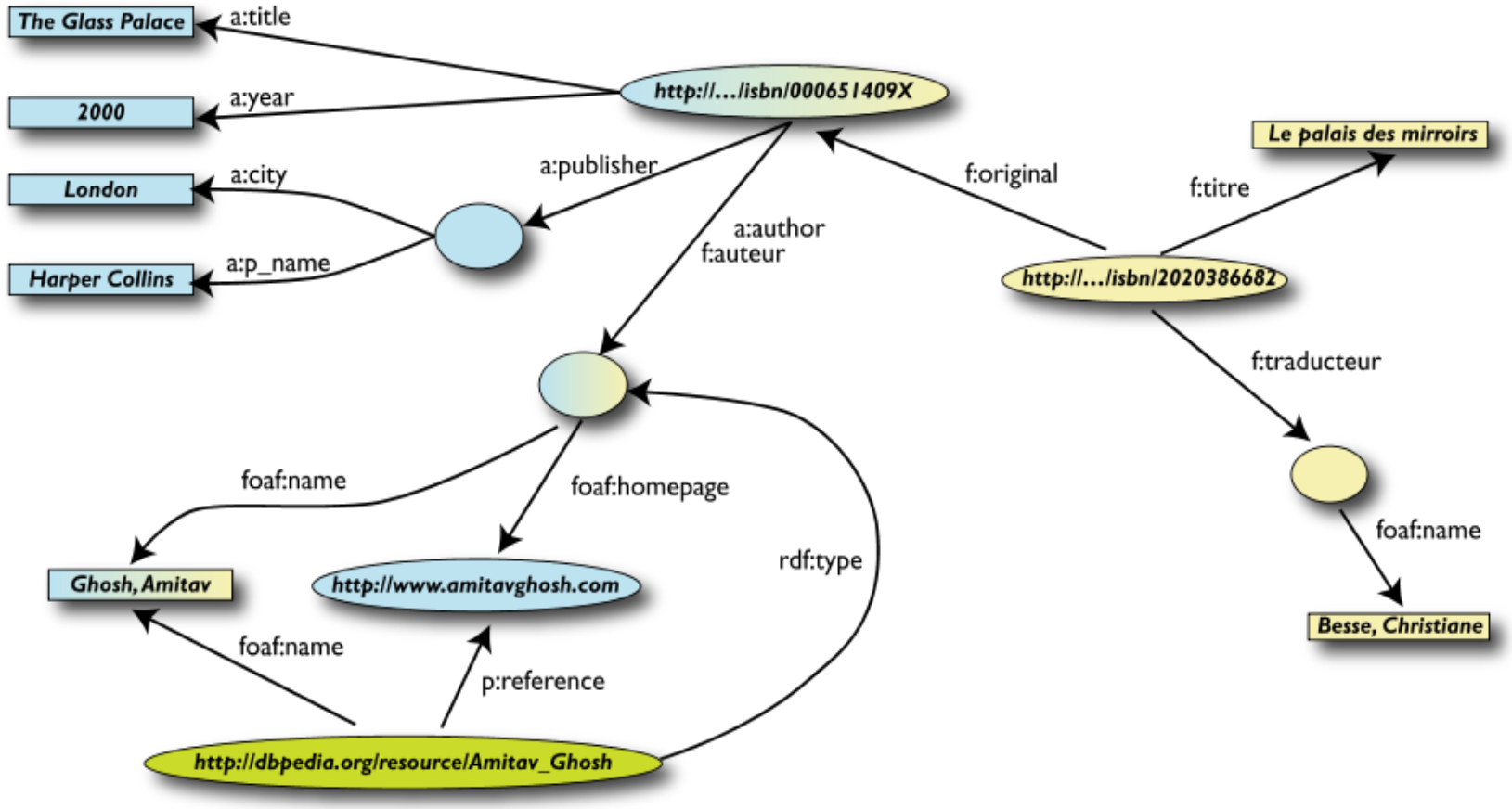
Extra: data integration



- Using "Person" will allow property reuse as well as more interoperability

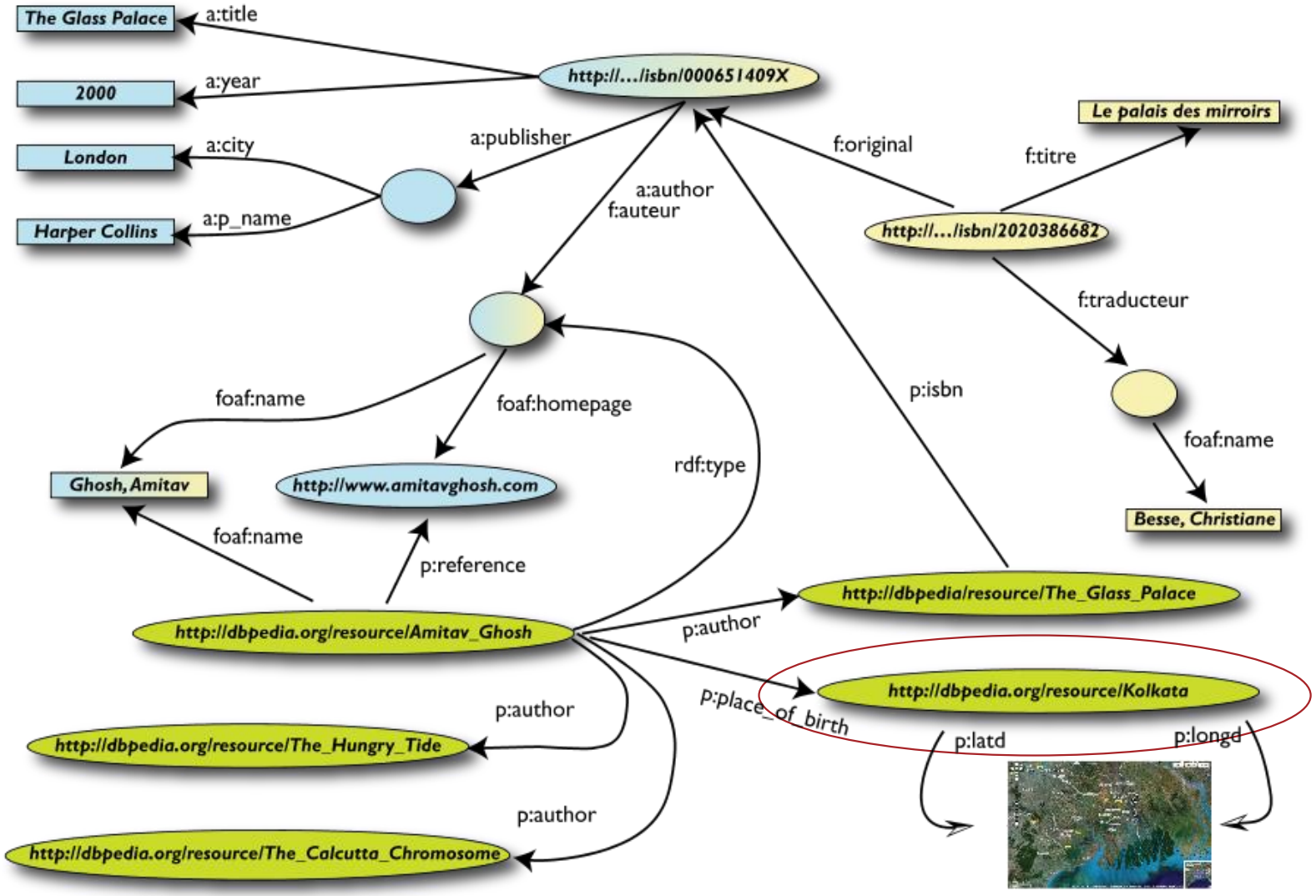


Interlinking to other datasets



- Enriching current information with external sources (e.g. Dbpedia)

Interlinking to other datasets



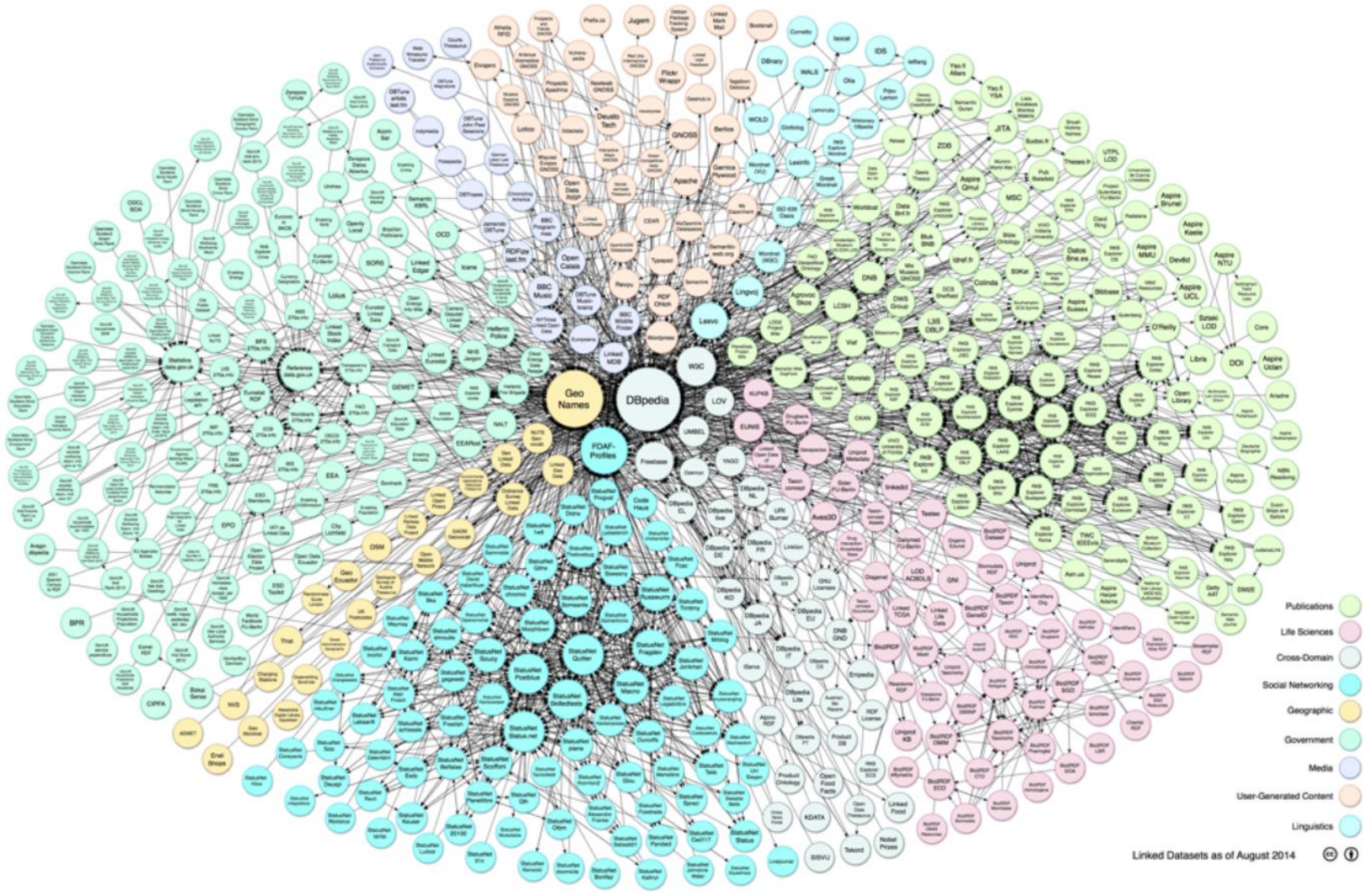
Which data should I link my data to?

- Where can I find datasets that are relevant as a context to my information?
- What is the value of the data in that dataset
- Will linking my data to data in such dataset provide an added value
- Does the external dataset and namespace belong to a estable institution? Is it being periodically maintained?
- Are URIs in the dataset stable? URIs modification seems unlikely?
- Are there other external links allowing applications to navigate away in search for interrelated resources?

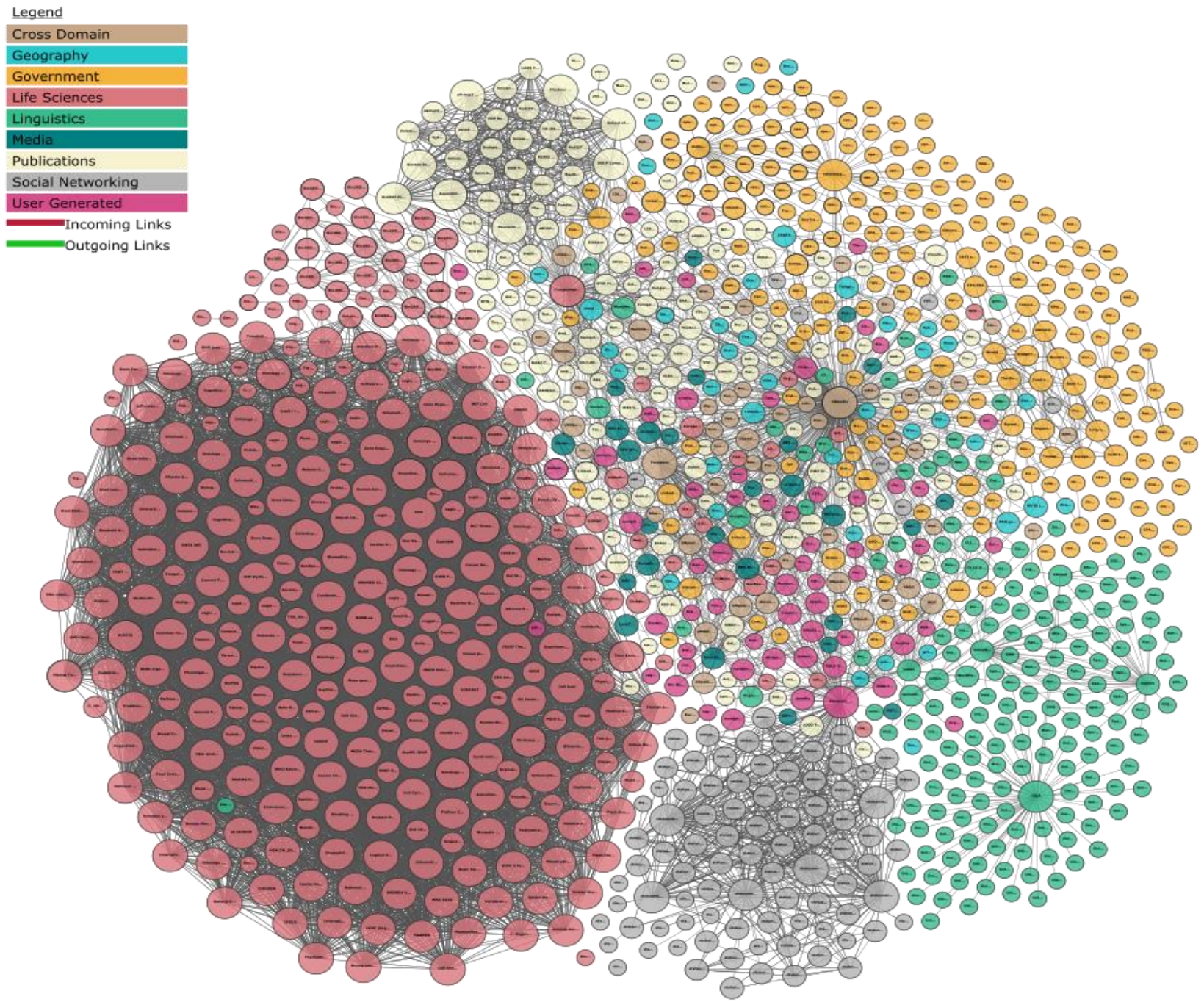
Linked data cloud

- Starts in 2007 as a W3C project "Linking Open Data project"
- **Objetive**: extending the Web with a common aggregate of data
  - (a) by publishing several open datasets in RDF
  - (b) linking RDF data elements stored in different sources
- From inception, the number of data sources / datasets has grown dramatically.

Linke ddata cloud



Linked data cloud



Towards a Web of data

- Steps to make available a dataset:
  - Model data and link them
  - Name resources using URIs
  - Reuse existing vocabularies as much as posible
  - Publish data descriptions for both people and automated systems
  - Transform data into RDF
  - Specify a license
  - Publish and disseminate the new dataset

References

- BERNERS LEE et al. The Semantic Web, Scientific American http://www.scientificamerican.com/article/the-semantic-web/
- HAUSENBLAS, M. 5 Stars Model on Open Government Data. http://5stardata.info/
- W3C RDF Reference: https://www.w3.org/RDF/

Open Data Certificate

- Questionnaire-based evaluation to recognizes Evaluación mediante public open data
- Use of *badges*
- Acknowldegement of "good data"
- Guarantee system is independent of quality
- Evaluation of data sustainability and reliability
- Promotes trust and data reuse
- Promotes good practices

Open Data Certificate

https://certificates.theodi.org/

BRONZE: data is openly licensed, available with no restrictions, accessible and legally reusable.

SILVER: satisfies the Bronze requirements, the data is documented in a machine readable format, reliable and offers ongoing support from the publisher via a dedicated communication channel.

GOLD: satisfies the Silver requirements, is published in an open standard machine readable format, has guaranteed regular updates, offers greater support, documentation, and includes a machine readable rights statement.

PLATINUM: satisfies the Gold requirements, has machine readable provenance documentation, uses unique identifiers in the data, the publisher has a communications team offering support. This is an exceptional example of an information infrastructure.