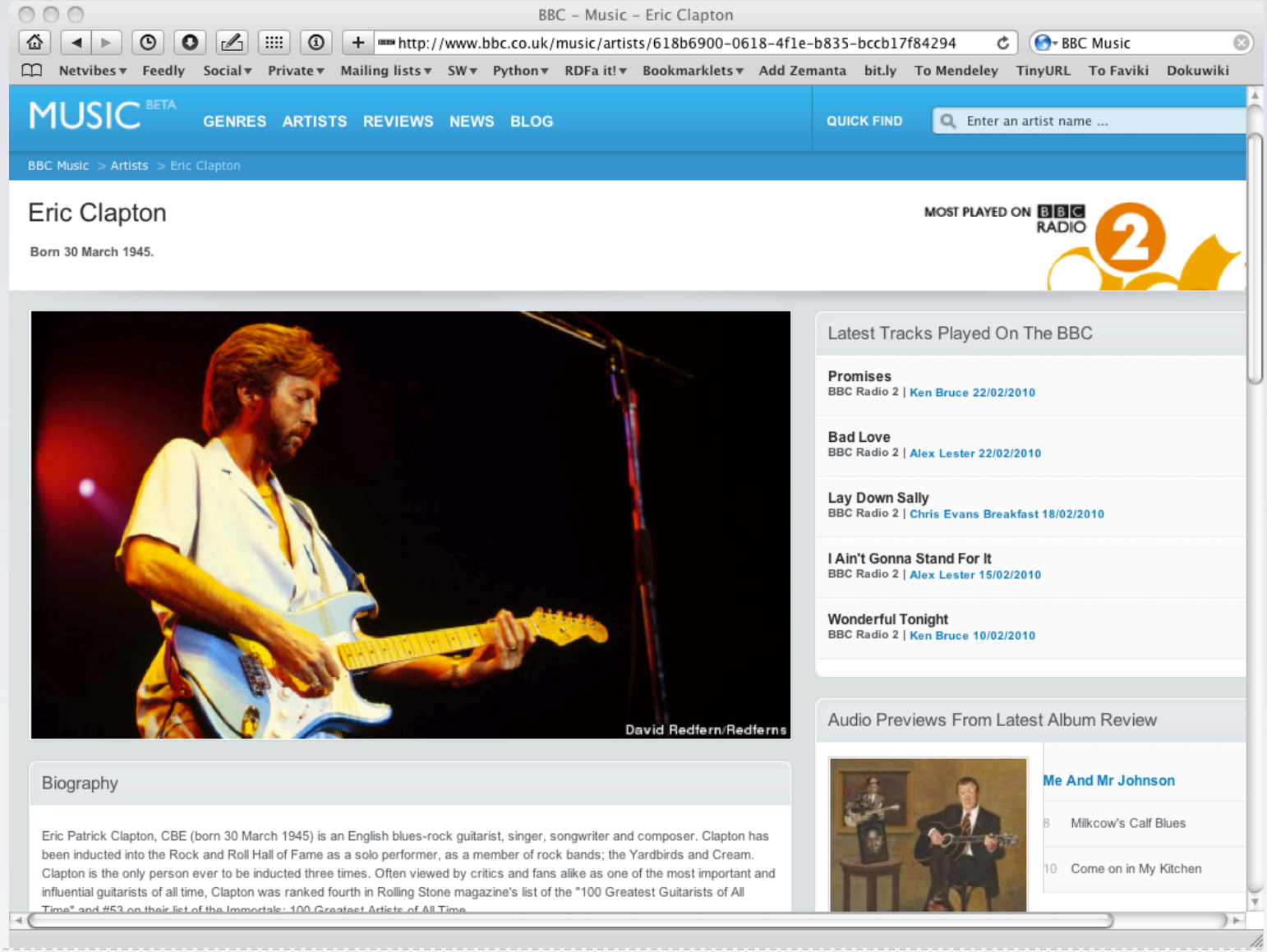
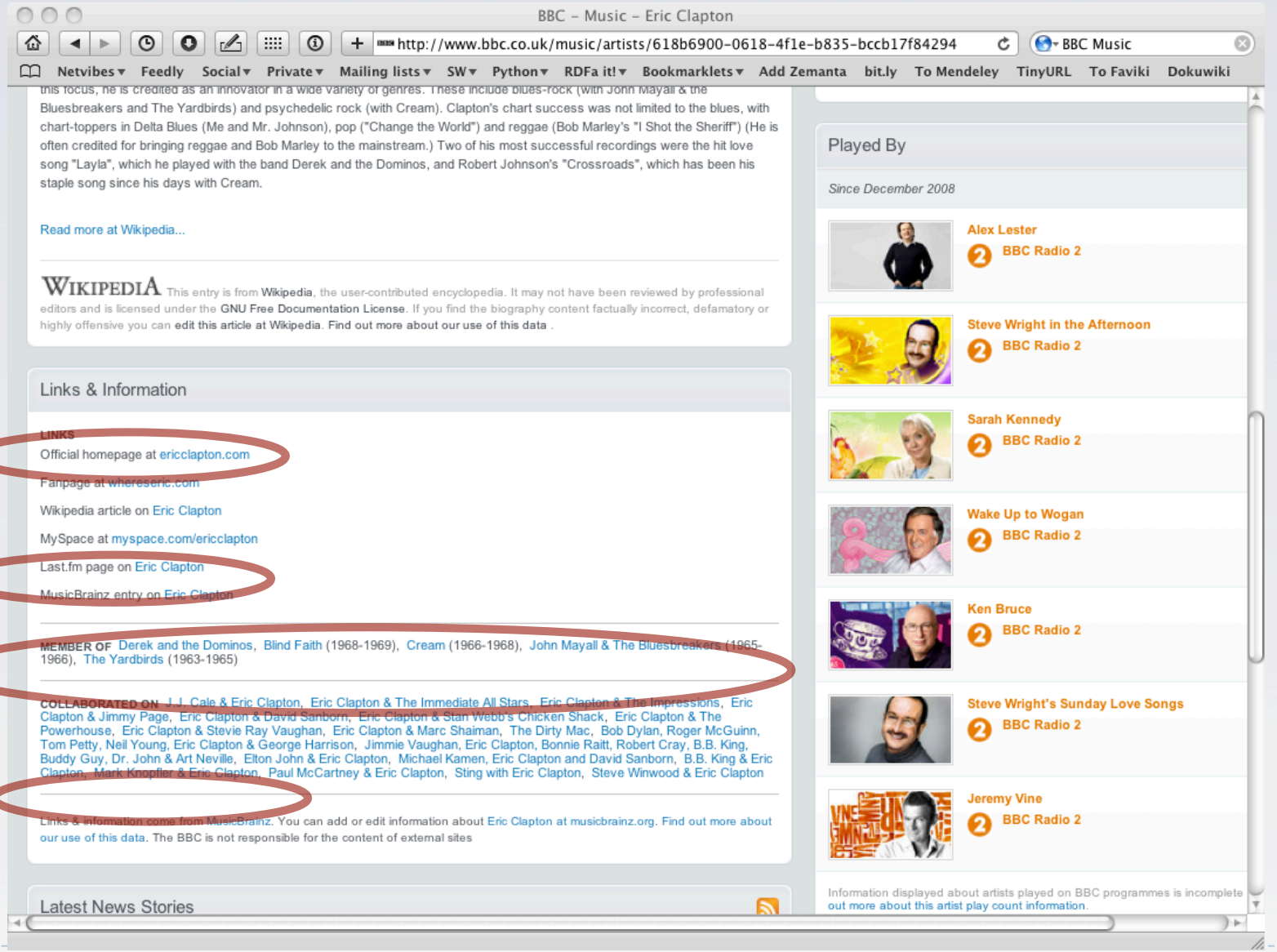
Introduction to Semantic Web and RDF

RDF, Linked Data workshop at DANS
The Hague, 28th July, 2010,
Ivan Herman, W3C

The Music site of the BBC



The Music site of the BBC



How to build such a site 1.

- Site editors roam the Web for new facts
  - may discover further links while roaming
- They update the site manually
- And the site gets soon out-of-date

How to build such a site 2.

- Editors roam the Web for new data published on Web sites
- "Scrape" the sites with a program to extract the information
  - Ie, write some code to incorporate the new data
- Easily get out of date again…

How to build such a site 3.

- Editors roam the Web for new data via API-s
- Understand those…
  - input, output arguments, datatypes used, etc
- Write some code to incorporate the new data
- Easily get out of date again…

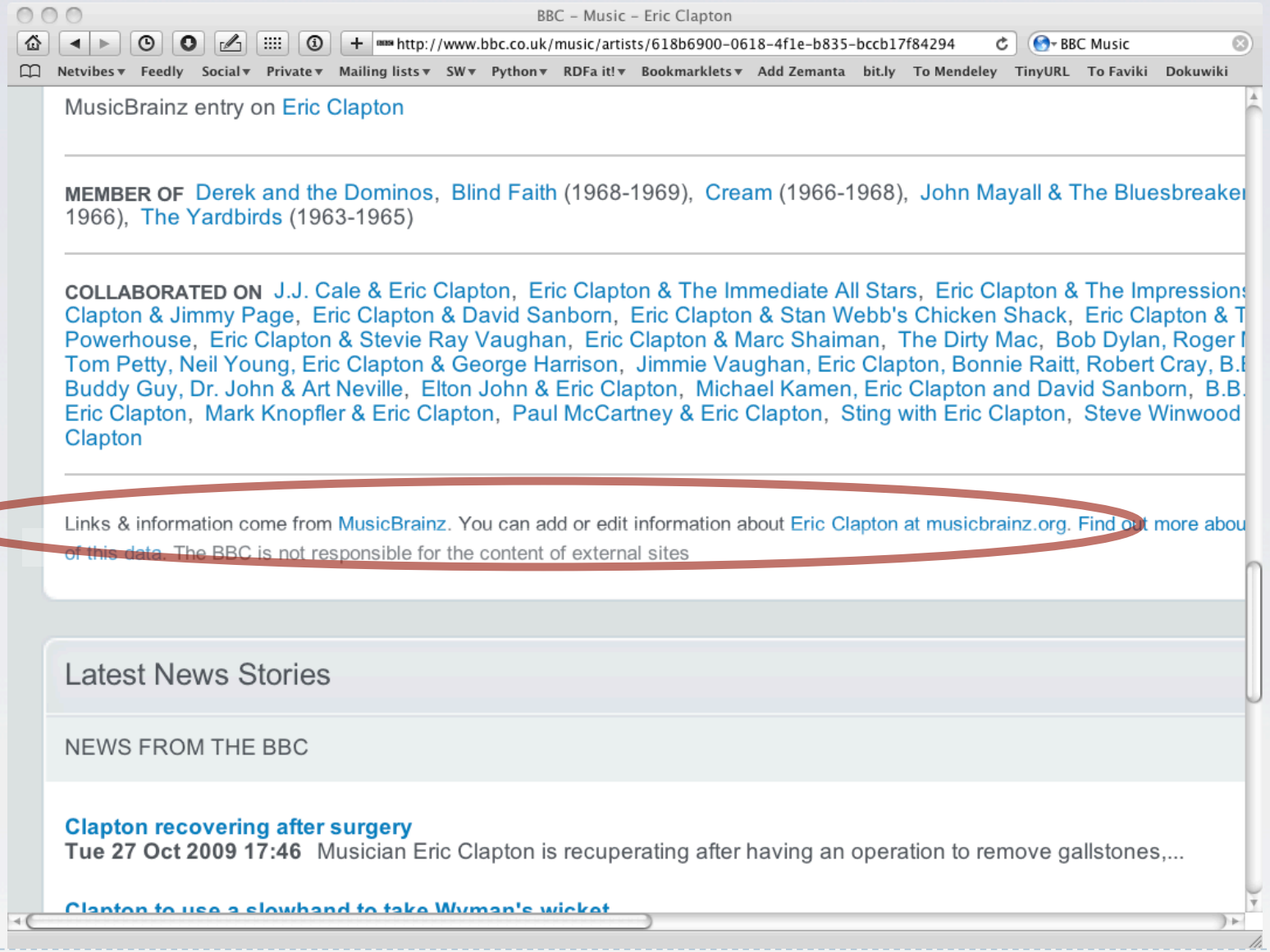
The choice of the BBC

- Use external, public datasets
  - Wikipedia, MusicBrainz, …
- They are available *as data*
  - not API-s or hidden on a Web site
  - data can be extracted using, eg, HTTP requests or standard queries

In short…

- Use the Web of Data as a Content Management System
- Use the community at large as content editors

And this is no secret...



Data on the Web

- There are more an more data on the Web
  - government data, health related data, general knowledge, company information, flight information, restaurants,…
- More and more applications rely on the availability of that data

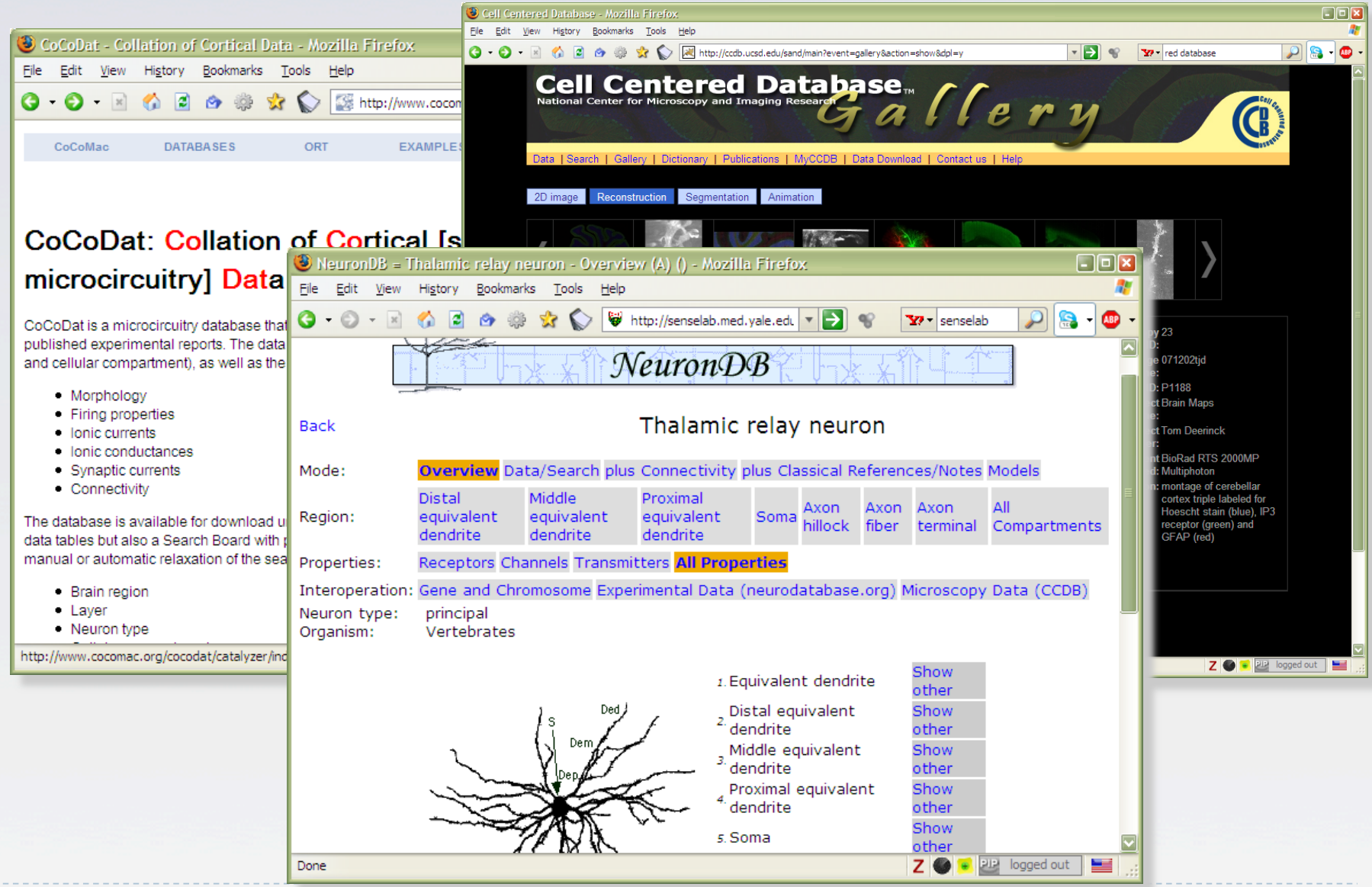
But… data are often in isolation, “silos”

Photo credit Alex (ajagendorf25), Flickr

Imagine…

- A "Web" where
  - documents are available for download on the Internet
  - but there would be no hyperlinks among them

And the problem *is* real…



Data on the Web is not enough…

- We need a proper infrastructure for a real *Web of Data*
  - data is available on the Web
    - accessible via standard Web technologies
  - data are interlinked over the Web
  - ie, data can be *integrated* over the Web
- This is where Semantic Web technologies come in

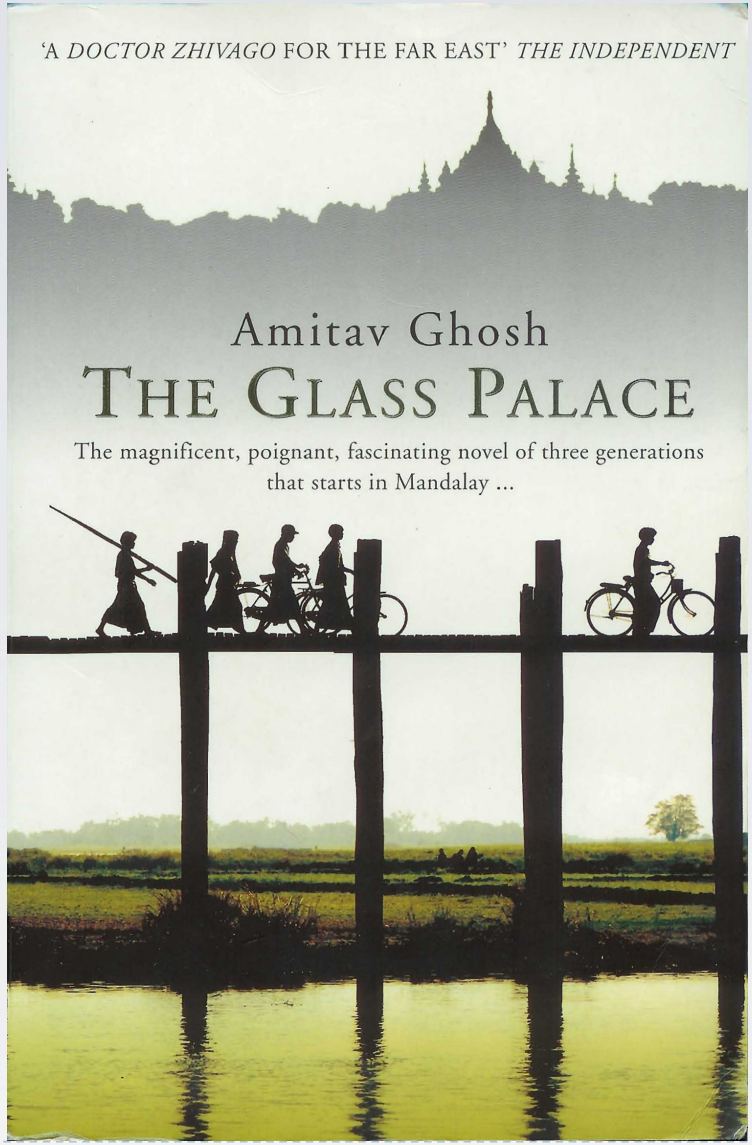
In what follows…

- We will use a simplistic example to introduce the main Semantic Web concepts

The rough structure of data integration

- Map the various data onto an abstract data representation
  - make the data independent of its internal representation…
- Merge the resulting representations
- Start making queries on the whole!
  - queries not possible on the individual data sets

We start with a book...



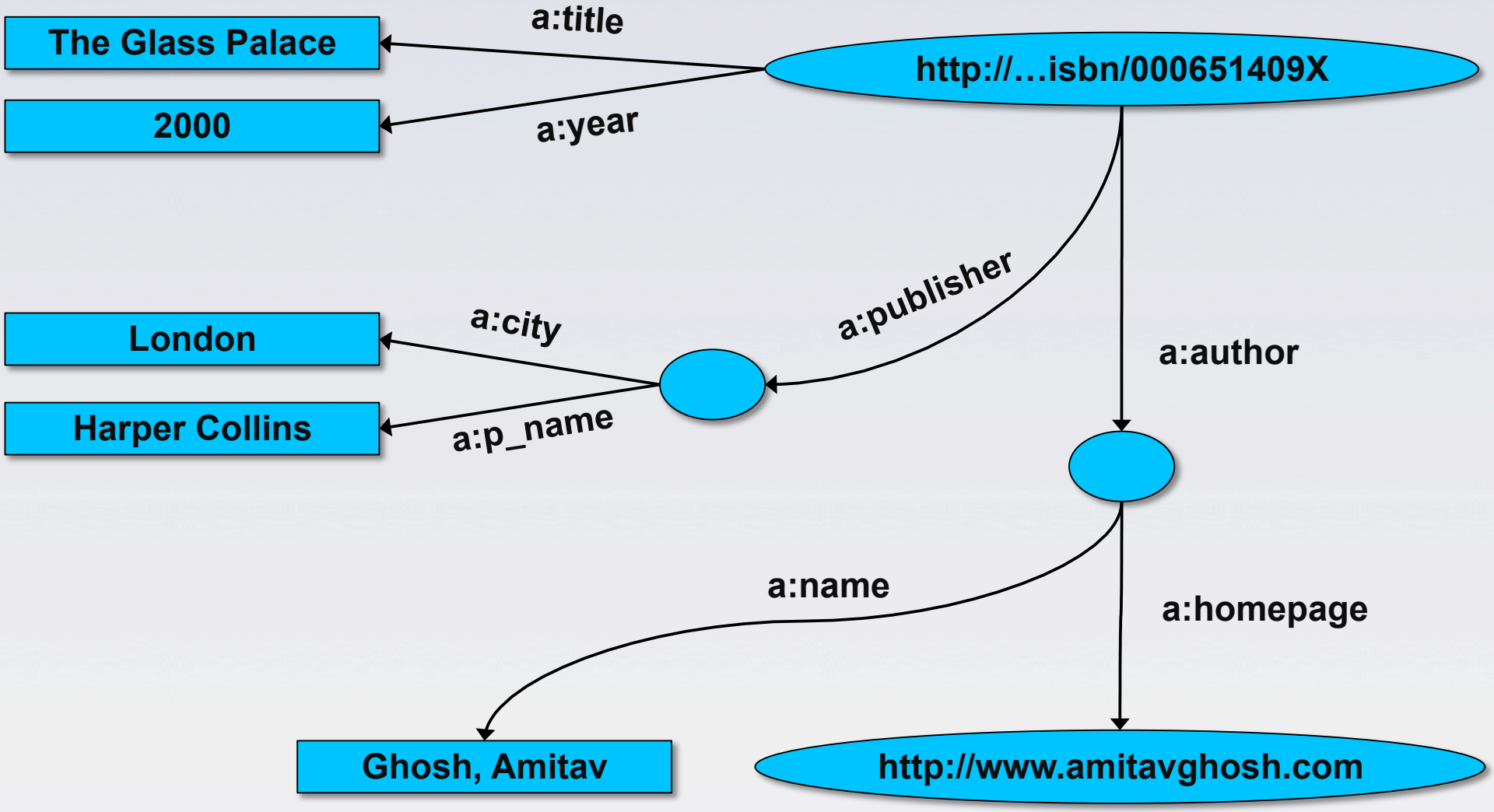
A simplified bookstore data (dataset "A")

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

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

ID	Publisher's name	City
id_qpr	Harper Collins	London

1st: export your data as a set of *relations*



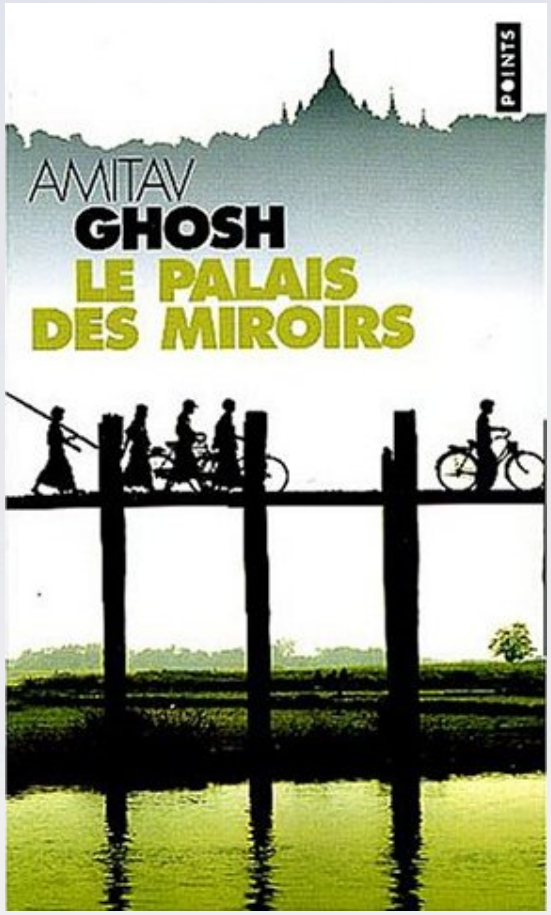
Some notes on the exporting the data

- Relations form a graph
  - the nodes refer to the "real" data or contain some literal
  - how the graph is represented in machine is immaterial for now

Some notes on the exporting the data

- Data export does not necessarily mean physical conversion of the data
  - relations can be generated on-the-fly at query time
    - via SQL "bridges"
    - scraping HTML pages
    - extracting data from Excel sheets
    - etc.
- One can export part of the data

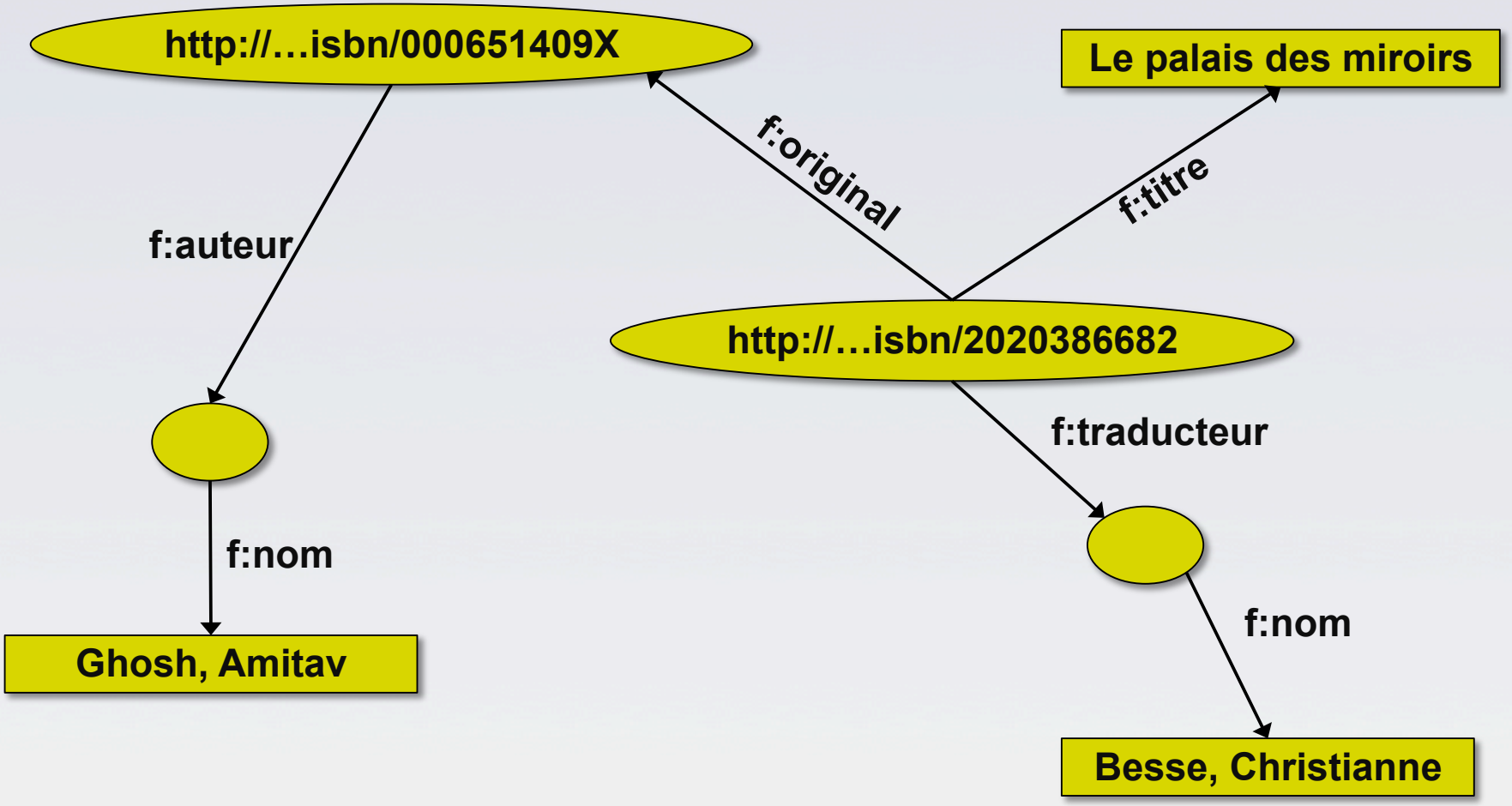
Same book in French…



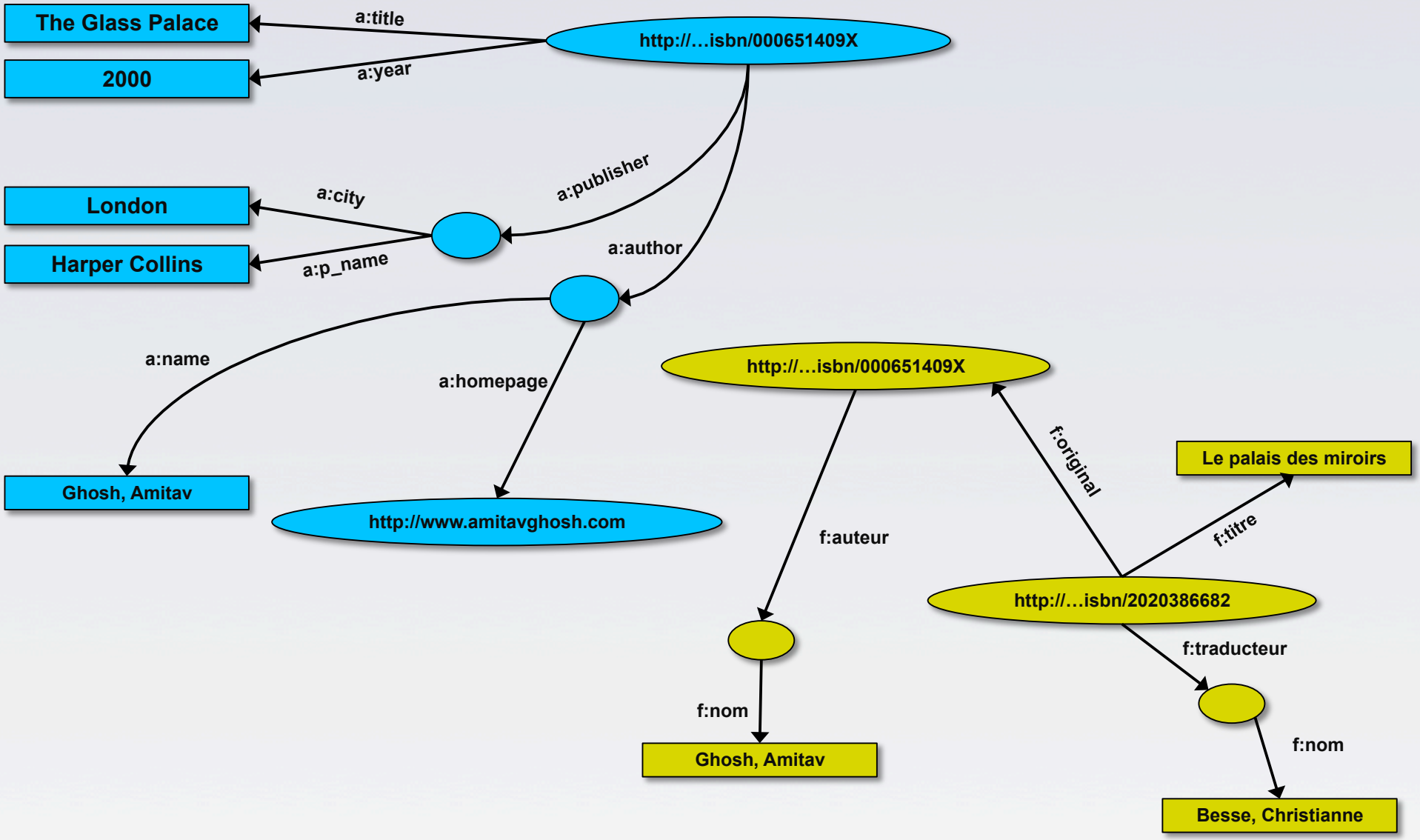
Another bookstore data (dataset “F”)

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

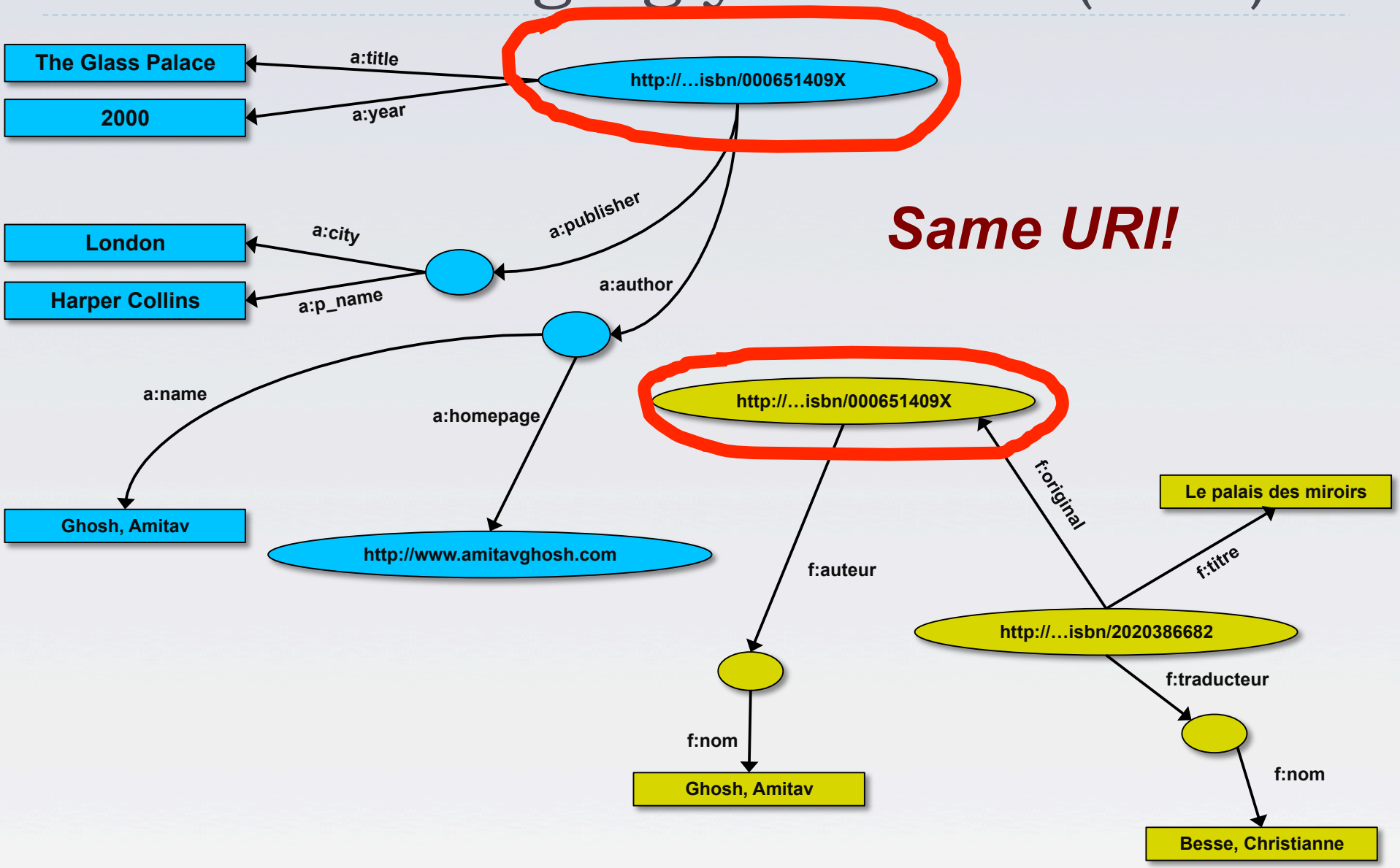
2nd: export your second set of data



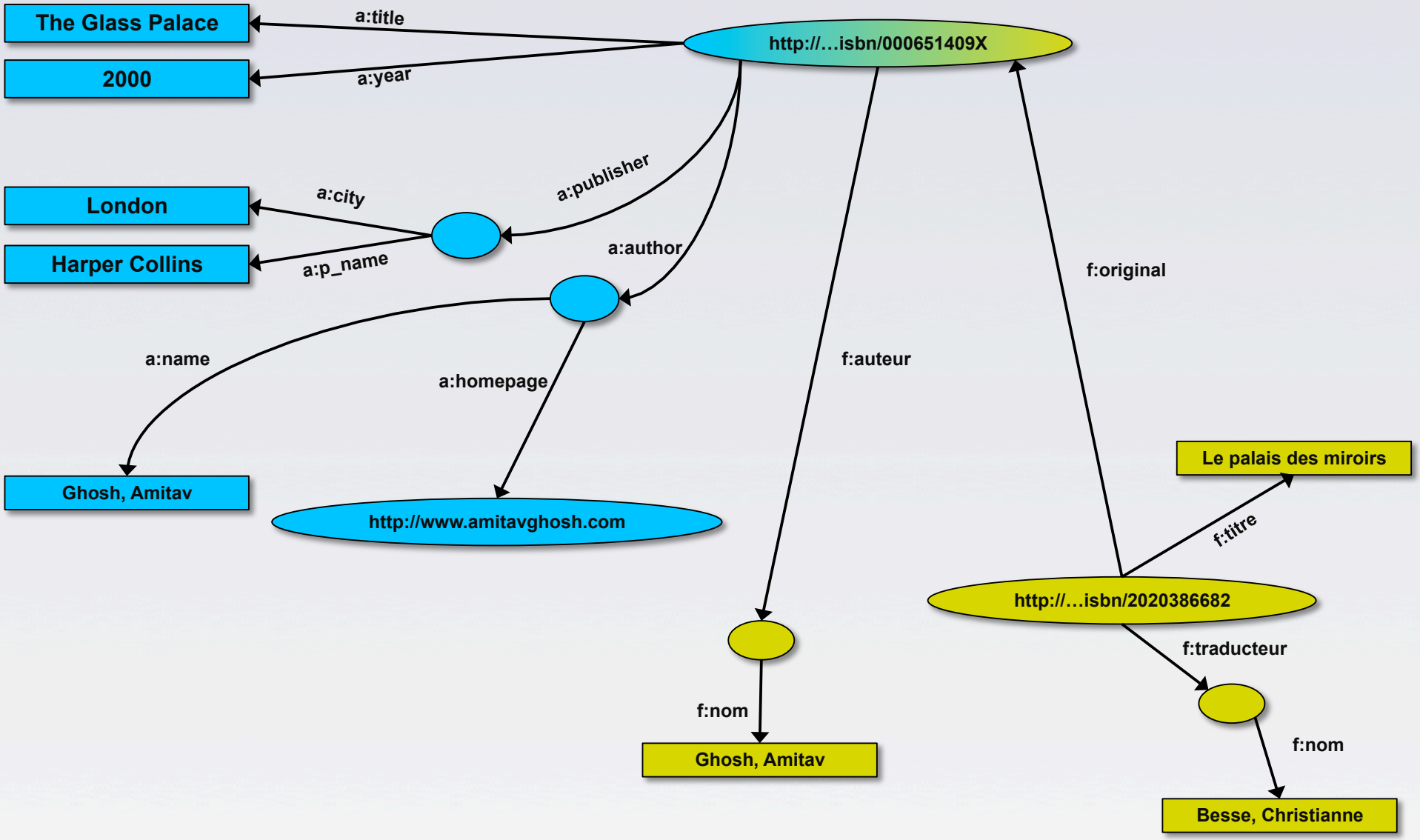
3rd: start merging your data



3rd: start merging your data (cont)

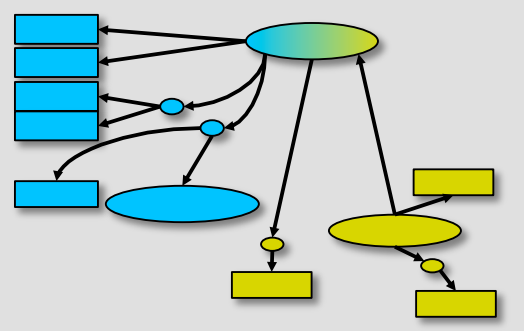


3rd: start merging your data



Start making queries…

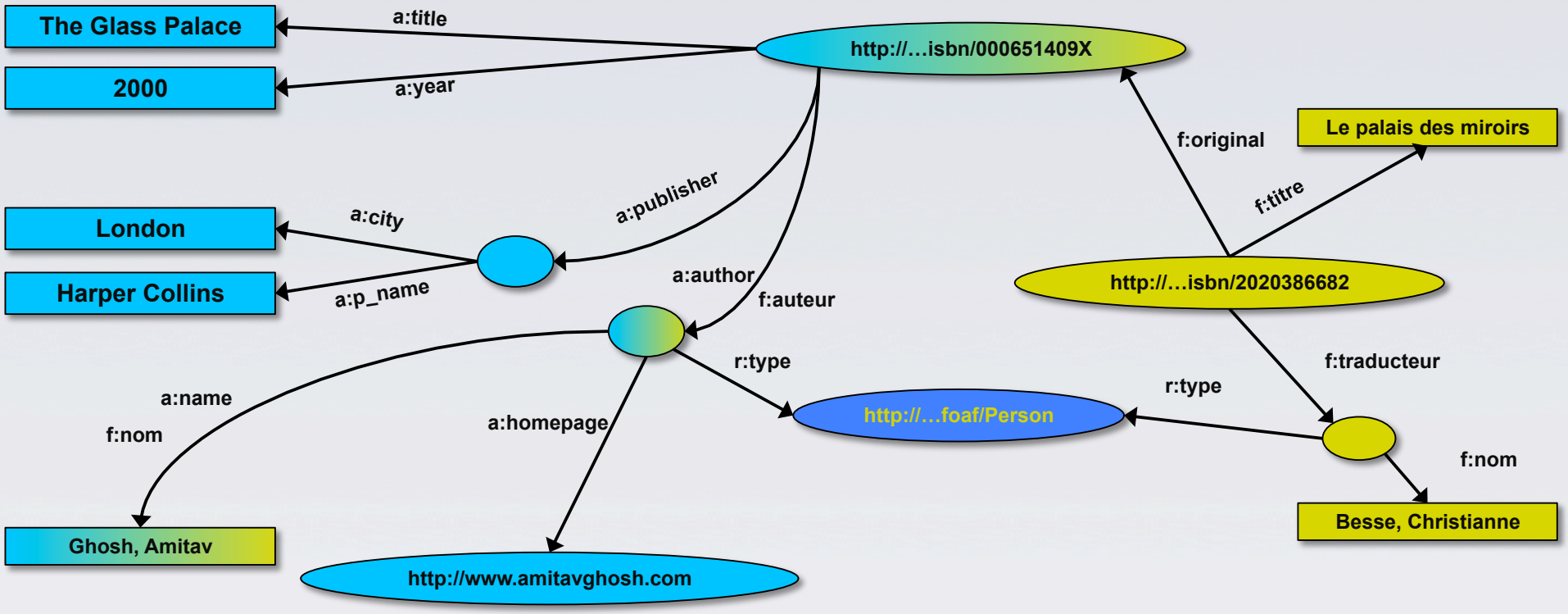
- User of data "F" can now ask queries like:
  - "give me the title of the original"
    - well, … « donnes-moi le titre de l'original »
- This information is not in the dataset "F"…
- …but can be retrieved by merging with dataset "A"!

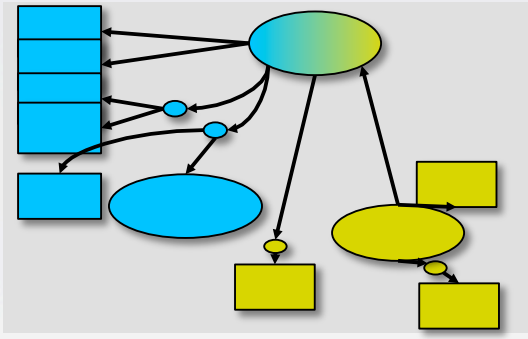


However, more can be achieved…

- We "feel" that a:author and f:auteur should be the same
- But an automatic merge doest not know that!
- Let us add some extra information to the merged data:
  - a:author same as f:auteur
  - both identify a "Person"
  - a term that a community may have already defined:
    - a "Person" is uniquely identified by his/her name and, say, homepage
    - it can be used as a "category" for certain type of resources

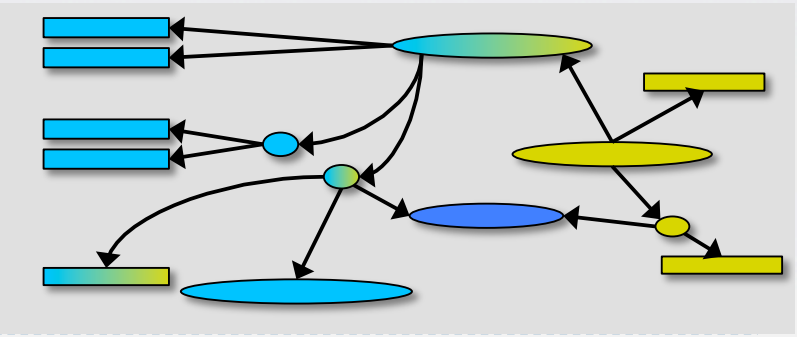
3rd revisited: use the extra knowledge





Start making richer queries!

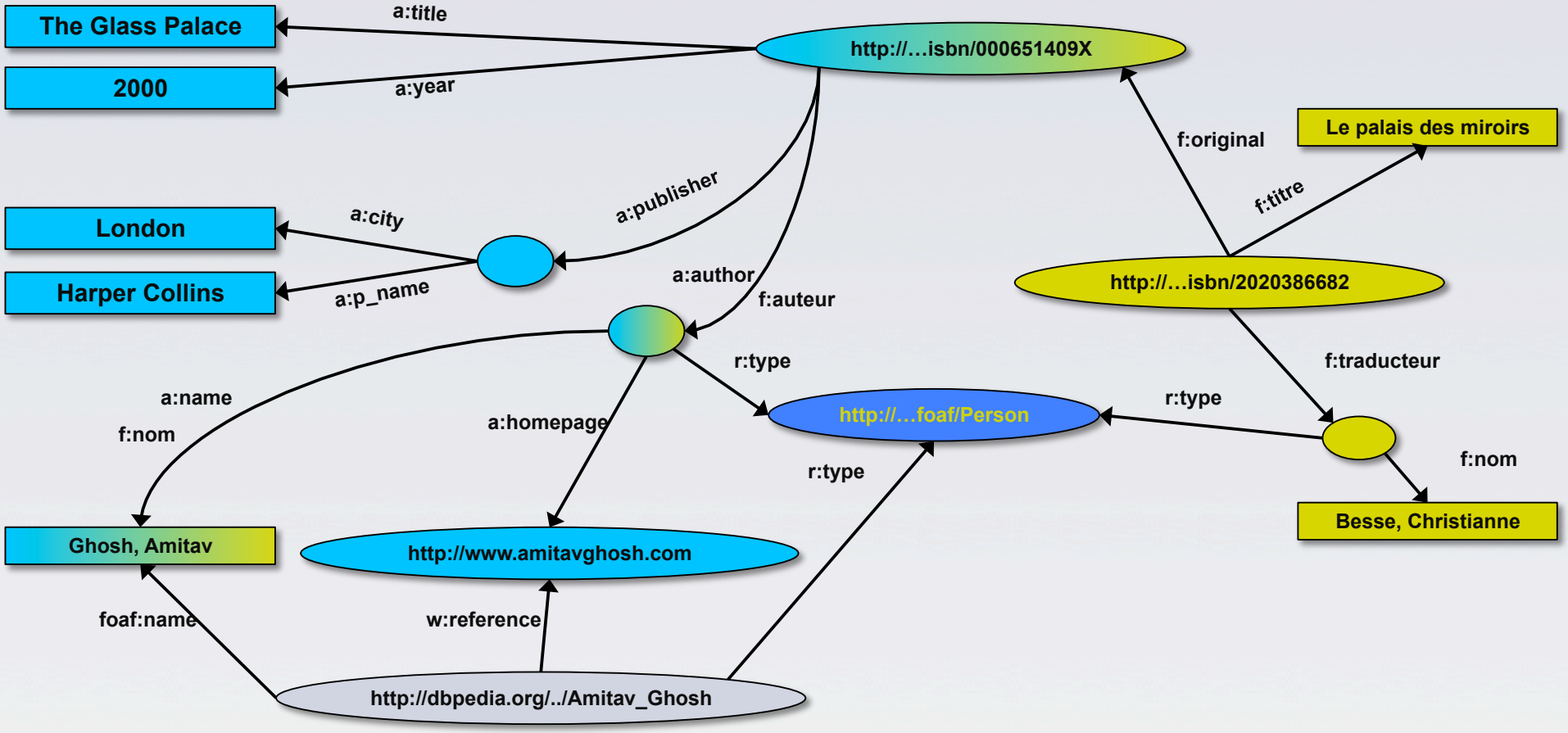
- User of dataset "F" can now query:
  - "donnes-moi la page d'accueil de l'auteur de l'original"
    - well… "give me the home page of the original's 'auteur'"
- The information is not in datasets "F" or "A"…
- …but was made available by:
  - merging datasets "A" and datasets "F"
  - adding three simple extra statements as an extra "glue"



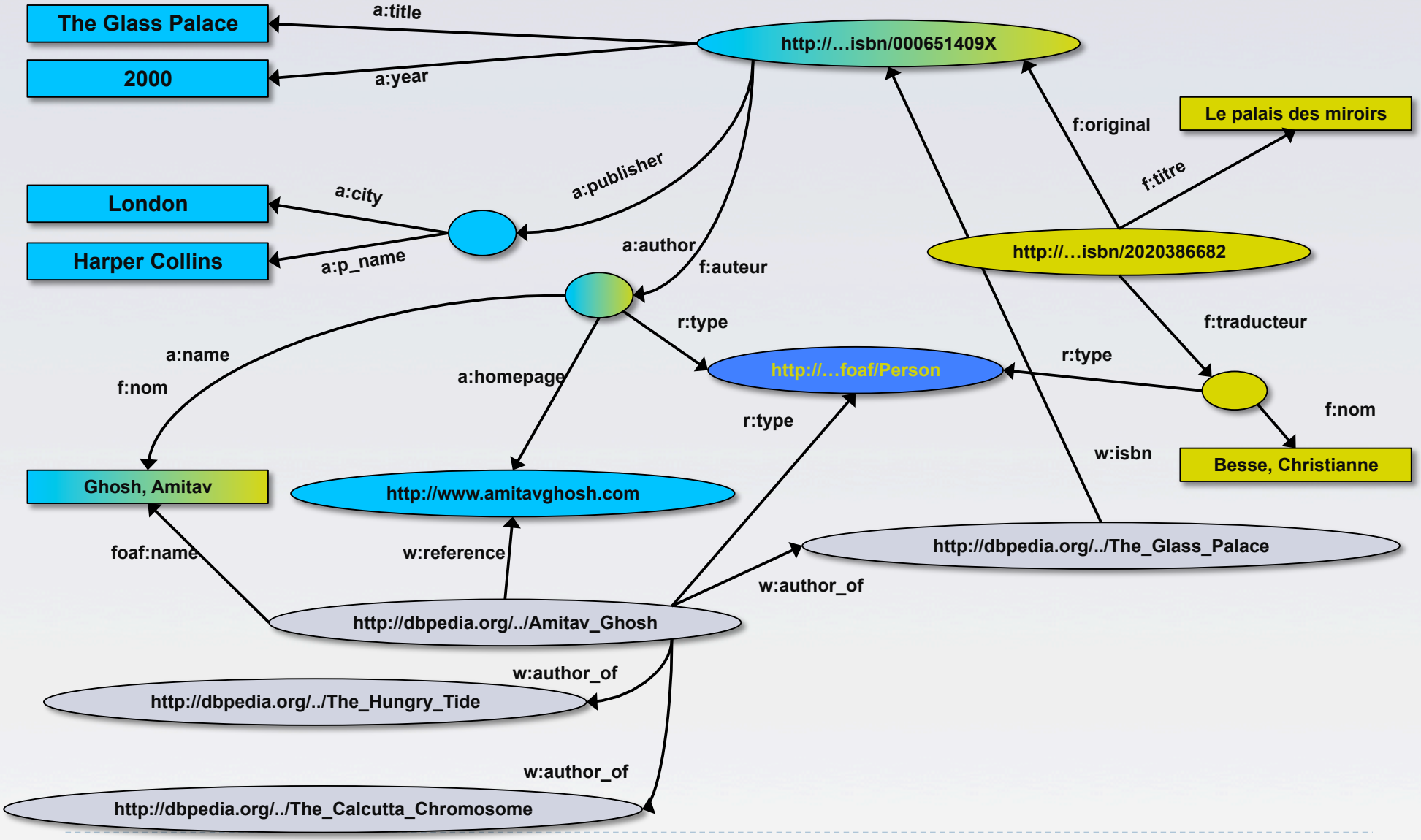
Combine with different datasets

- Using, e.g., the "Person", the dataset can be combined with other sources
- For example, data in Wikipedia can be extracted using dedicated tools
  - e.g., the "dbpedia" project can extract the "infobox" information from Wikipedia already…

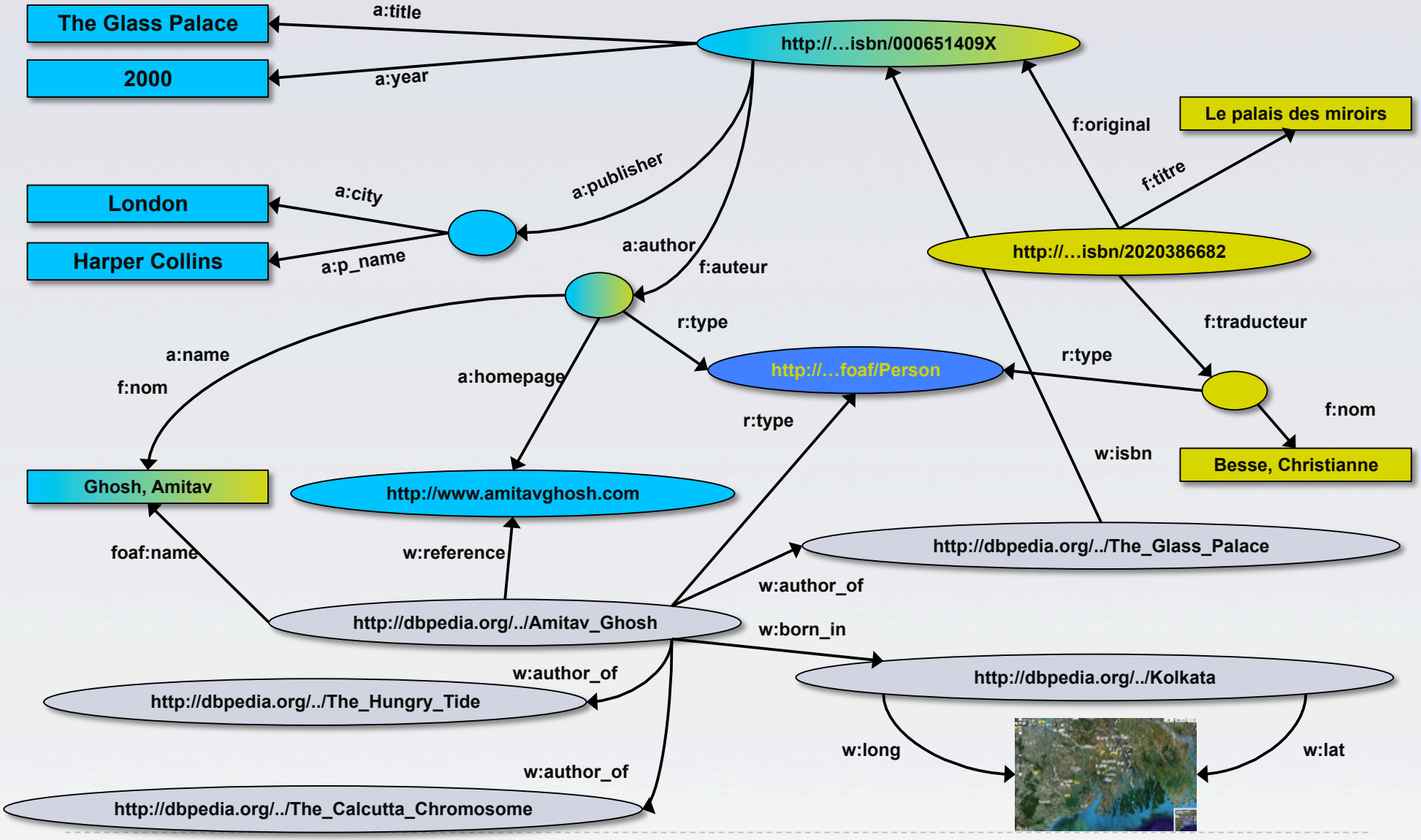
Merge with Wikipedia data



Merge with Wikipedia data



Merge with Wikipedia data



Is that surprising?

- It may look like it but, in fact, it should not be…
- What happened via automatic means is done every day by Web users!
- The difference: a bit of extra rigour so that machines could do this, too

What did we do?

- We combined different datasets that
  - are somewhere on the web
  - are of different formats (mysql, excel sheet, etc)
  - have different names for relations
- We could combine the data because some URI-s were identical (the ISBN-s in this case)

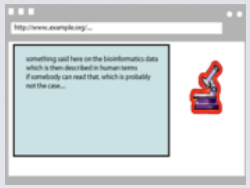
What did we do?

- We could add some simple additional information (the "glue"), also using common terminologies that a community has produced
- As a result, new relations could be found and retrieved

It could become even more powerful

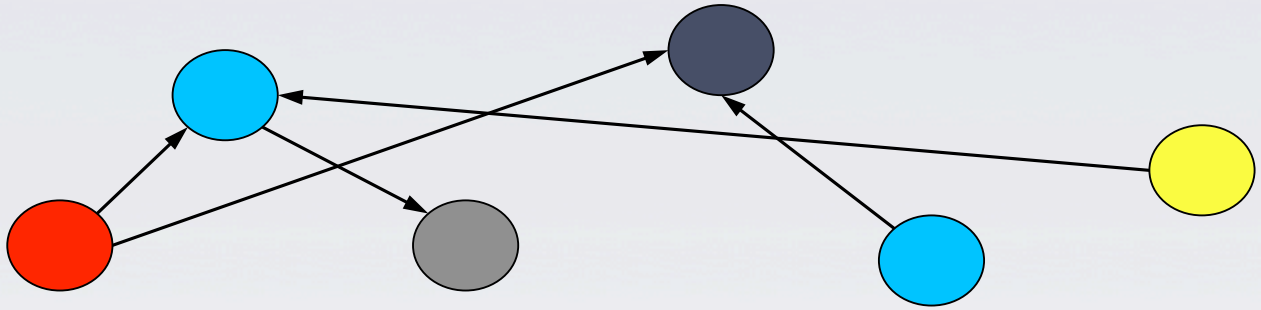
- We could add extra knowledge to the merged datasets
  - e.g., a full classification of various types of library data
  - geographical information
  - etc.
- This is where ontologies, extra rules, etc, come in
  - ontologies/rule sets can be relatively simple and small, or huge, or anything in between…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

What did we do? (cont)



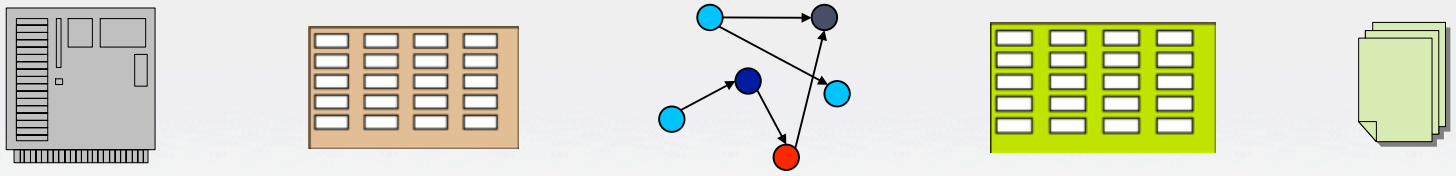
Applications

Manipulate
Query
…



Data represented in abstract format

Map,
Expose,
…



Data in various formats

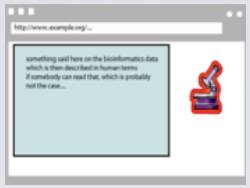
So where is the Semantic Web?

- The Semantic Web provides technologies to make such integration possible!

Details: many different technologies

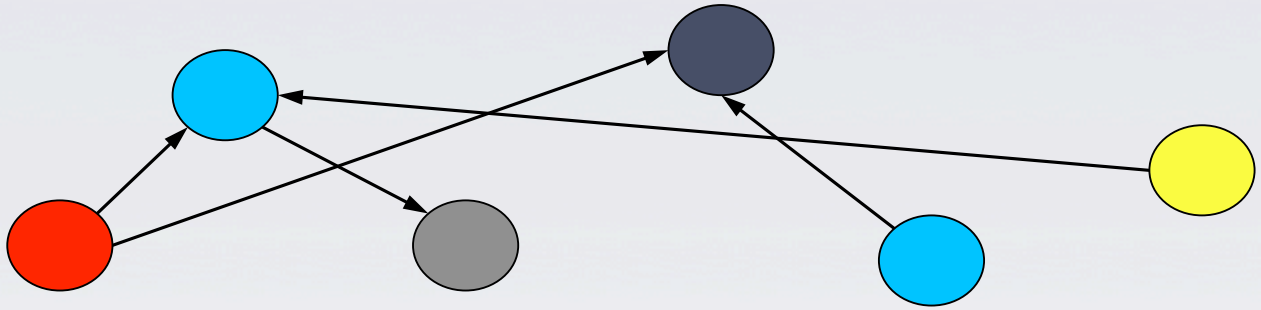
- an abstract model for the relational graphs: RDF
- add/extract RDF information to/from XML, (X) HTML: GRDDL, RDFa
- a query language adapted for graphs: SPARQL
- characterize the relationships and resources: RDFS, OWL, SKOS, Rules
  - applications may choose among the different technologies
- reuse of existing "ontologies" that others have produced (FOAF in our case)

Using these technologies…



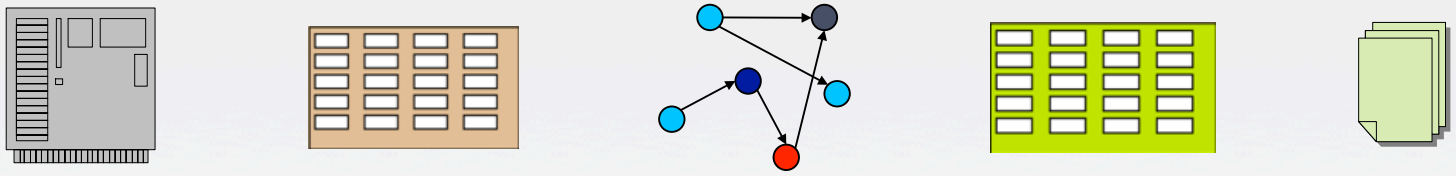
Applications

SPARQL, Inferences …



Data represented in RDF with extra knowledge (RDFS, SKOS, RIF, OWL,…)

RDB ⇔ RDF, GRDL, RDFa, …



Data in various formats

Where are we today (in a nutshell)?

- The technologies are in place, lots of tools around
  - there is always room for improvement, of course
- Large datasets are "published" on the Web, ie, ready for integration with others
- Large number of vocabularies, ontologies, etc, are available in various areas

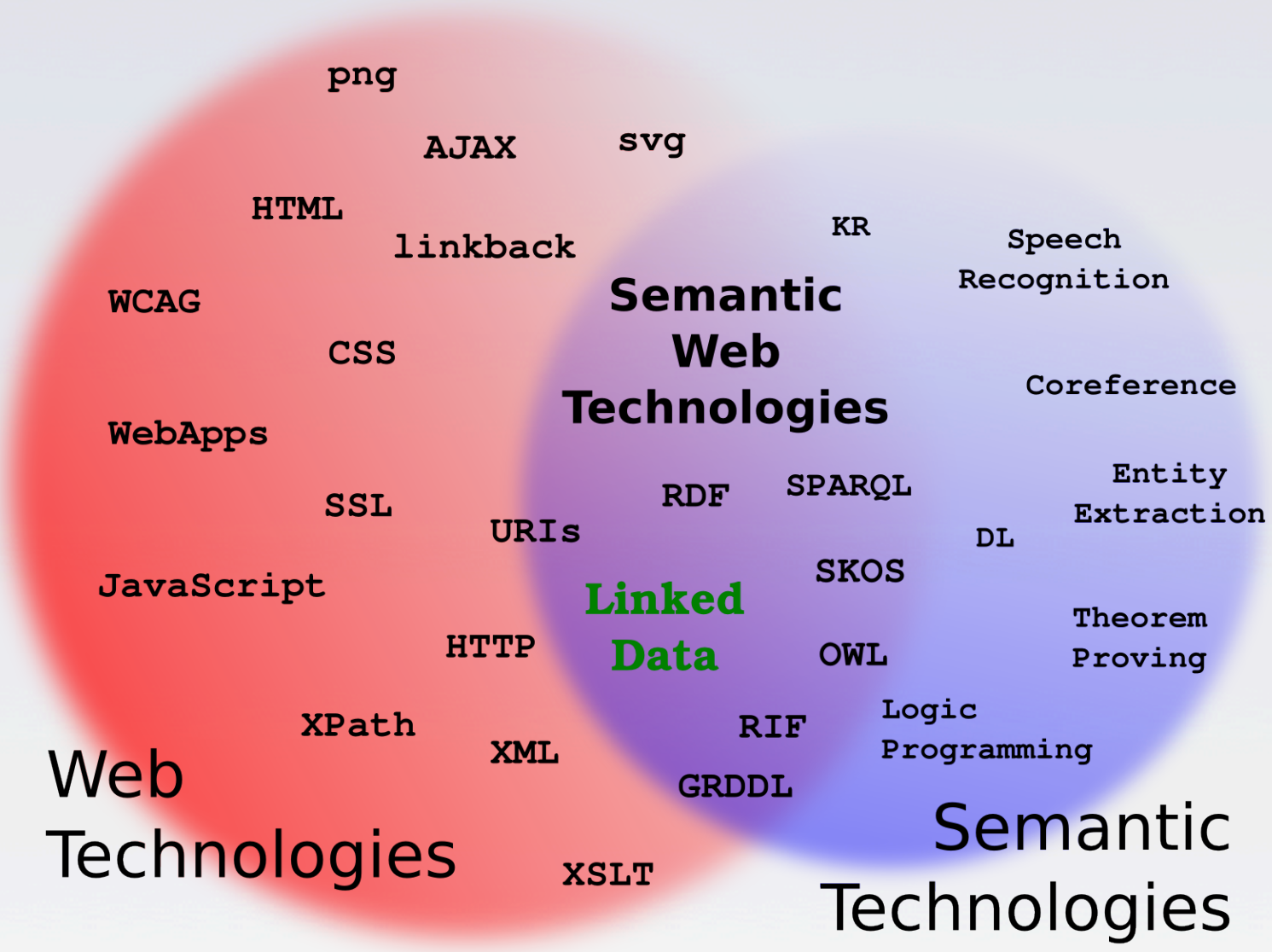
Everything is not rosy, of course…

- Tools have to improve
  - scaling for *very* large datasets
  - quality check for data
  - etc
- There is a lack of knowledgeable experts
  - this makes the initial "step" tedious
  - leads to a lack of understanding of the technology

There are also R&D issues

- What does query/reasoning means on Web scale data?
- How does one incorporate uncertainty information?
- What is the granularity for access control, security, privacy…
- What types of user interfaces should we have for a Web of Data?
- etc.

Fit in the larger landscape…



Courtesy of Sandro Hawke, W3C

Thank you for your attention!

These slides are also available on the Web:

http://www.w3.org/2010/Talks/0728-TheHague-IH/