

Semantic Web - 3: standards and languages for knowledge representation

N. Aussenac-Gilles

IRIT- CNRS

aussenac@irit.fr

MELODI group

<http://www.irit.fr/-Equipe-MELODI->



Further readings

- **Foundations of Semantic Web Technologies**

Pascal Hitzler, Sebastian Rudolph, Markus Kroetzsch

Coll. : Textbooks in Computing. Chapman & Hall/CRC (2009)

- More tutorials and slides

- <http://www.semantic-web-book.org/page/Slides>

- <http://www-sop.inria.fr/acacia/cours/iut2007/sparql-rule.pdf>

- Survey récents

- <https://arxiv.org/pdf/2002.00388v2.pdf>

A Survey on Knowledge Graphs: Representation, Acquisition and Applications

- <https://arxiv.org/pdf/2003.02320.pdf>

Knowledge Graphs (132 pages)

En attendant les TP

- Dbpedia <http://fr.dbpedia.org/>
- Installer le logiciel Protégé :
<http://protege.stanford.edu/>

Motivations for a new web

- A vision: the www
- New needs: towards a semantic web
- New means: the semantic web program



A semantic web: foundational acts

- From Tim Beners-Lee's vision at the 1994 WWW conference
<http://www.w3.org/Talks/WWW94Tim>
- The Semantic Web by Tim Berners-Lee, James Hendler and Ora Lassila, Scientific American, may 2001

<http://websemantique.org/ScientificAmericanMai2001>

The semantic web statement

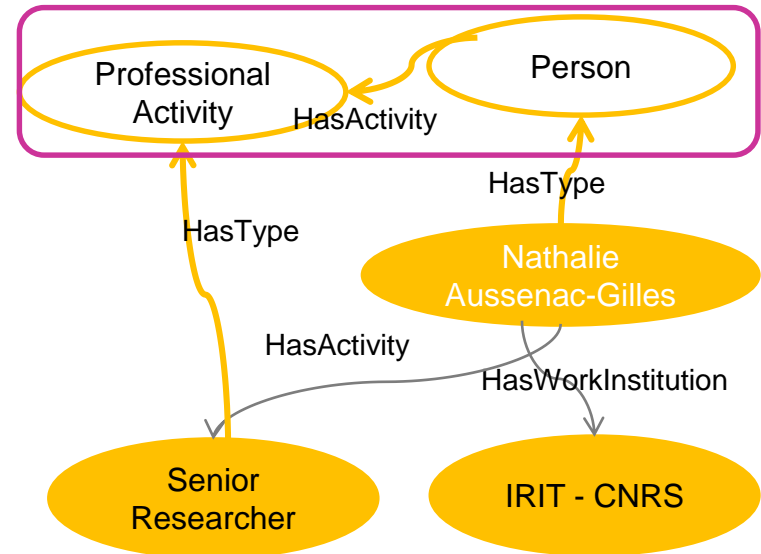
“The Semantic Web is an **extension of the current web** in which information is given well-defined meaning, better **enabling computers and people to work in cooperation**. The mix of content on the web has been shifting from exclusively human-oriented content to more and more **data content**.

The Semantic Web brings to the web the idea of having data defined and linked in a way that it can be used for **more effective discovery, automation, integration, and reuse across various applications**. For the web to reach its full potential, it must evolve into a Semantic Web, providing a universally accessible platform that allows data to be shared and **processed by automated tools as well as by people.**”

W3C Semantic Web Activity Statement

The semantic web program

- Continuity with the www
 - Semantics for computers
 - Open and decentralised
 - Extension of the existing web
- Knowledge representation
 - AI and logic based languages
 - Open world hypothesis
 - Flexibility and decentralisation
- Ontologies and knowledge graphs
 - Shared and consensual formal vocabularies
 - Public on the web
 - Axioms and inferences



The semantic web program

■ Formal Agents

- Web programs that combine, aggregate, share data and make decisions
- Even “things” can become nodes of the web and behave like agents - > IoT

■ Dynamicity, knowledge evolution

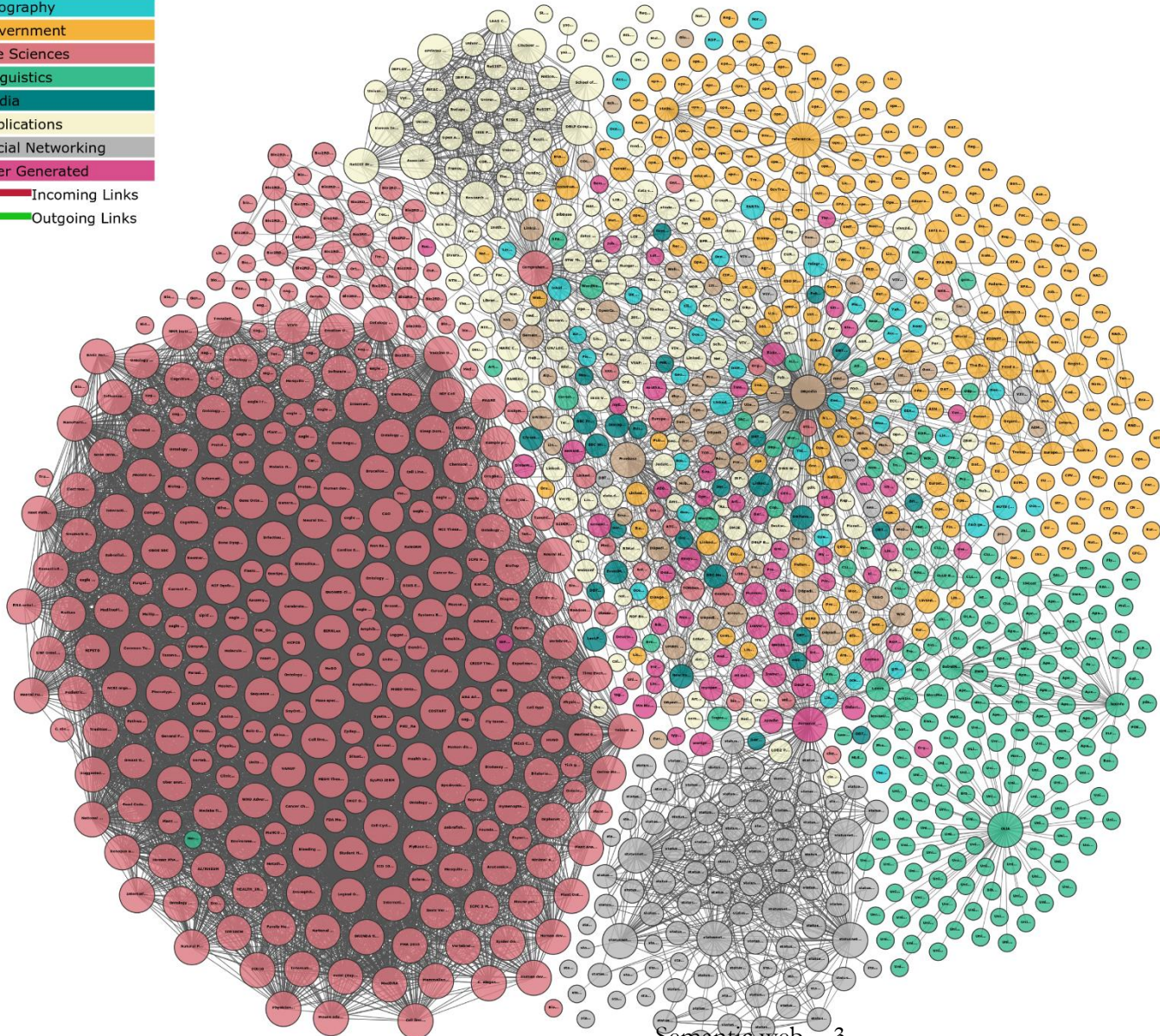
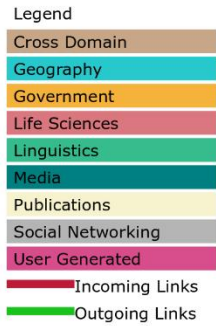
- Anyone can feed the semantic web
- Redundant information may appear
- Need for identifying relations between similar entities (data **mapping**, ontology **alignment**)

The semantic web program: how?

- Add knowledge to the web
 - Propose new standards for web knowledge representations
 - Collect structured data
- New applications
 - Semantic web services
 - New communication protocols, trust management
- Facilitate interoperability
 - Of heterogeneous data and vocabularies

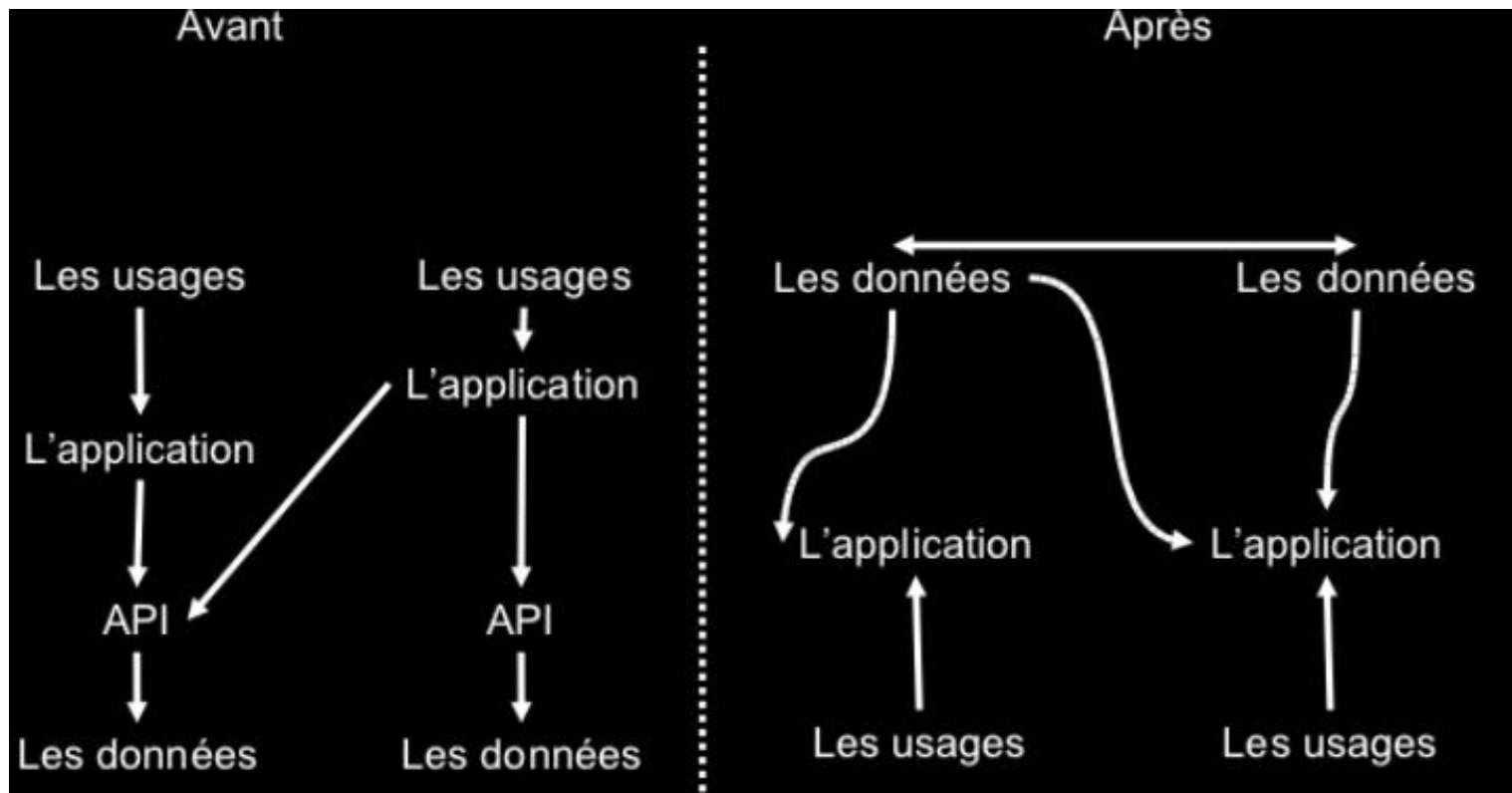


Knowledge in the LOD cloud



The semantic web program: how?

- A new paradigm for web applications



The semantic web: who will do it? (2020)

- Adding knowledge has a cost
 - Build ontologies
 - Annotate pages
 - Identify user's needs
- Who will pay for it?
 - Authors
 - Users
 - Web managers
 - Developers of innovating applications

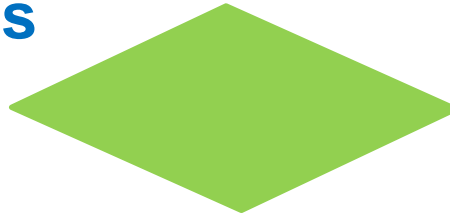
The semantic web: who will do it? (2020)

- Adding knowledge has a cost
 - Build ontologies: ML + human
 - Browse the web, collect pages with search engines
 - Information extraction, entity linking and NLP > automatic and usage oriented semantic annotation
- Who will pay for it?
 - Scientific community and standardization groups build resources
 - Developers of innovating applications use them

Building blocs of the semantic web

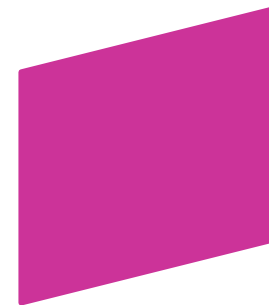
Applications

- Annotation
- Alignment



Groundings

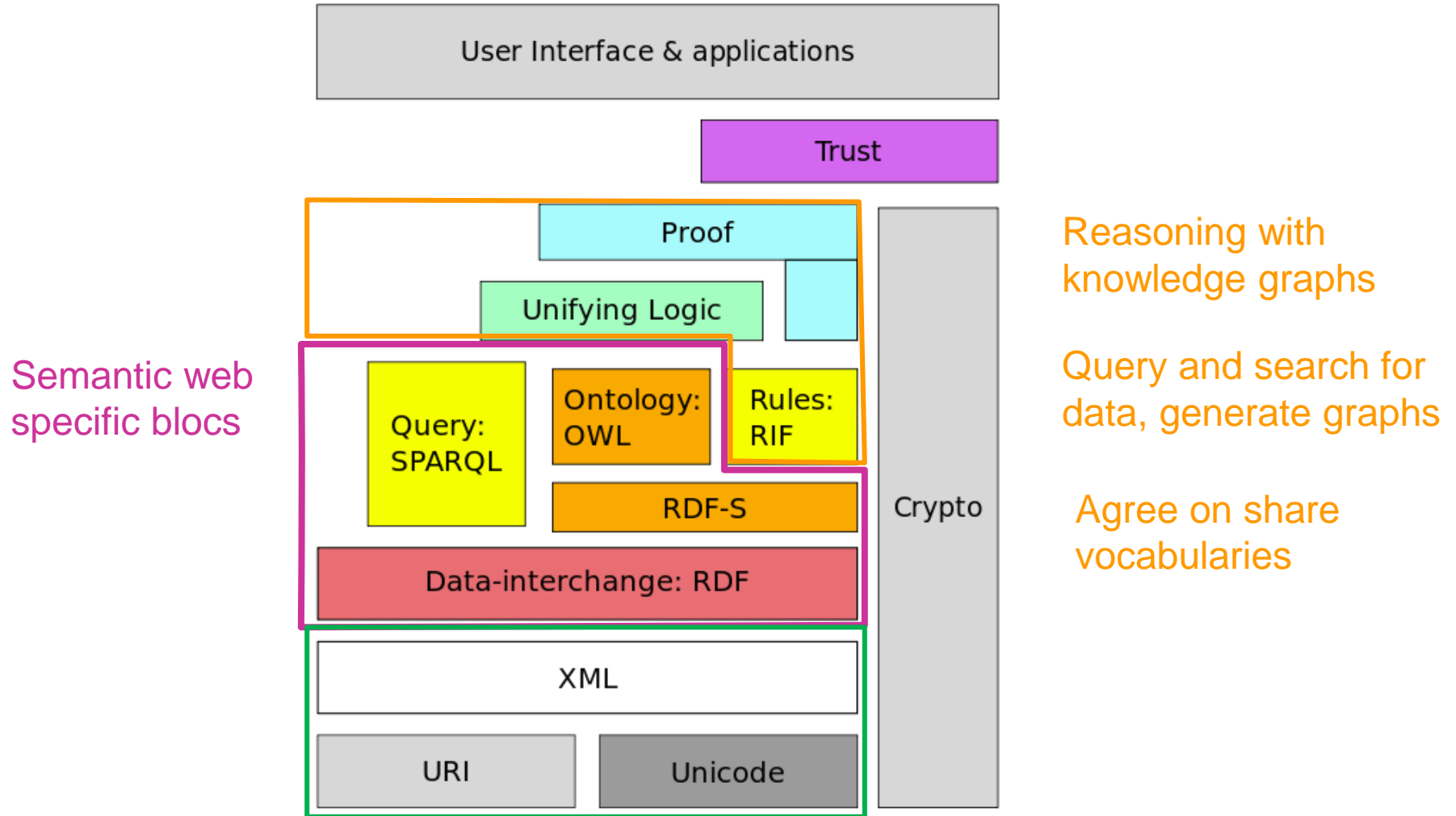
- Hypertexts and XML
- Description logic
- Formal ontology



Technologies

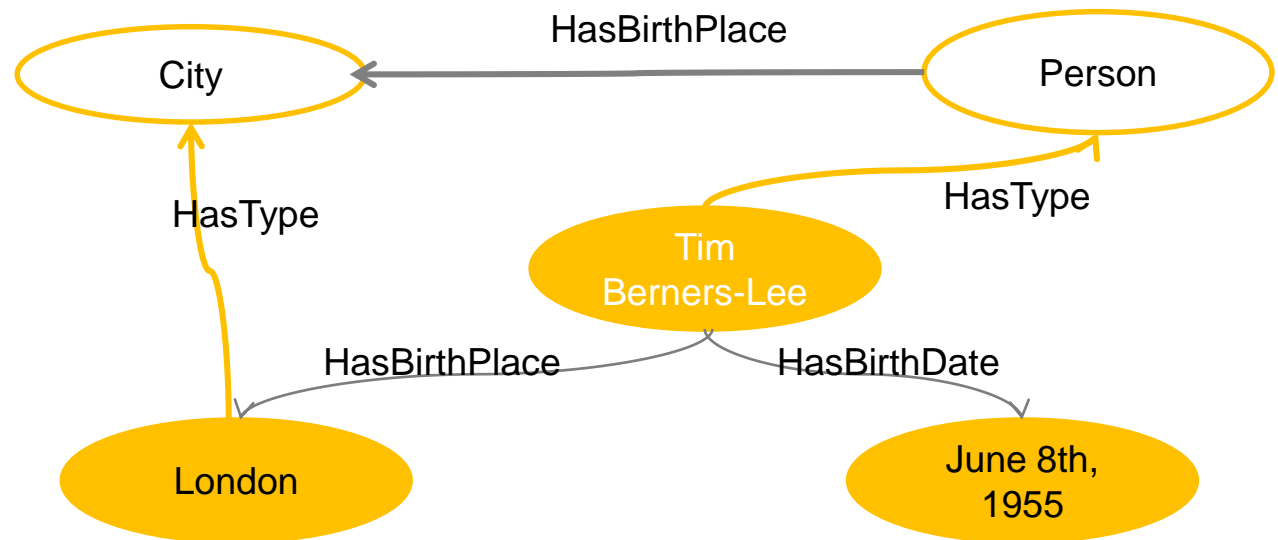
- Languages
- Models: ontologies, vocabularies, Linked Data

The semantic web layer cake (2006)



Motivations and design options

- A web linked resources -> uniform handling of resources, URI
- Sharing data and “meaning” -> type definitions
- Connecting data to documents -> XML compliant
- AAA -> open representation, modularity

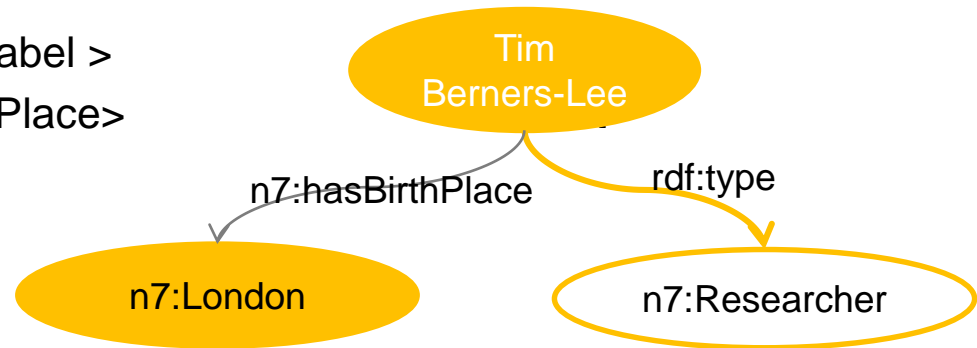


Motivation: semantic search on the web

“Tim Berners Lee is a researcher”

```
<rdf:Description id=n7:Tb>  
  <rdfs:label> “Tim Berners Lee” </ rdfs:label >  
  <hasBirthPlace> n7:London </hasBirthPlace>  
</rdf:Description >  
n7:Tb rdf:type n7:Researcher
```

n7 = namespace



“I am looking for persons born in London”

```
SELECT ?x WHERE {  
  ?x rdf:type n7:Person .  
  ?x n7:hasBirthPlace #London ..  
}
```

Motivation: semantic search on the web

“Tim Berners Lee is a researcher”

n7 = namespace

```
< rdf:Description rdf:ID = n7:Tb >  
  < rdfs:label > “Tim Berners Lee” </ rdfs:label >  
  < birthPlace > n7:London </ birthPlace >  
</rdf:Description>  
n7:Tb rdf:type n7:Researcher
```

RDF

“I am looking for persons”

```
SELECT ?x WHERE {?x rdf:type n7:Person. }
```

SPARQL

```
n7:Tb rdf:type n7:Person
```

Ou

```
n7:Researcher rdfs:subClassof n7:Person
```

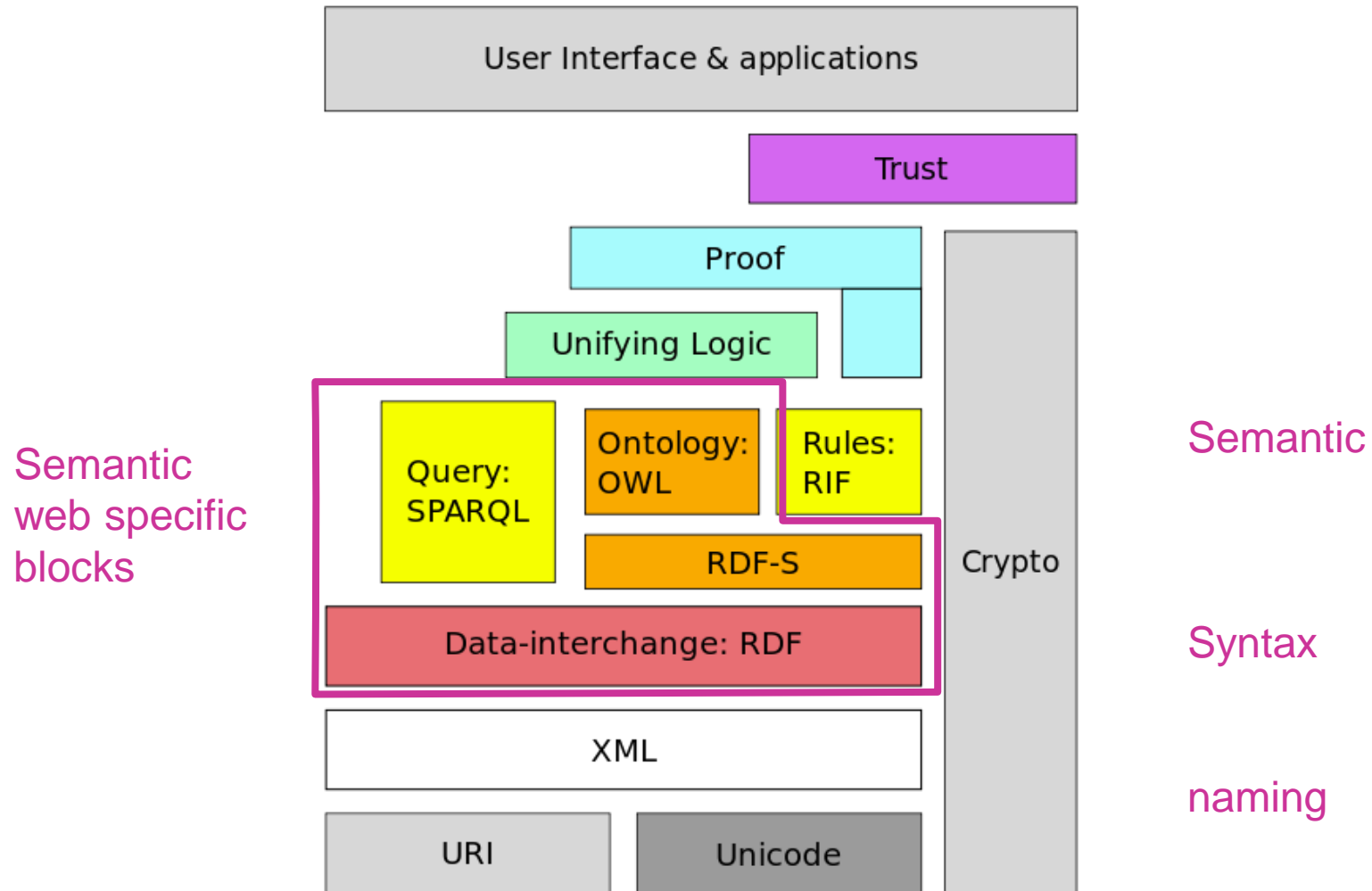
RDFs

```
∀?x ?y (?x rdf:type ?y and ?y rdfs:subClassof ?z)  
→ ?x rdf:type ?z
```

Semantics of subsumption

```
n7:Tb rdf:type n7:Person
```

The Semantic Web layer cake (2006)



- RDF Building Blocks and Turtle Syntax
- Model Theory For RDF
- RDF Schema
- RDFS Entailment
- Shortcomings of RDF

RDF: Resource Description Framework

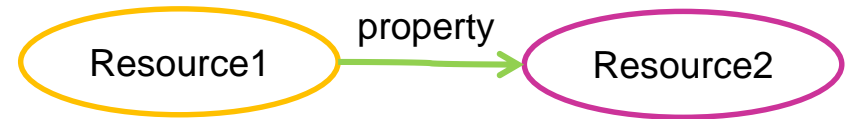


- A meta-data model
 - A descriptive model (1999)
 - A W3C standard (2004)

<http://www.w3.org/RDF/> <http://www.w3.org/TR/rdf-concepts/>
- Universal machine readable data format
- Resources identified by URIs
 - Uniform Resource Identifier
 - URIs identify on the web any Resource (on or outside the web)
 - on the www
 - any object with a clear identity (within the context of a given application)
 - examples: books, cities, humans, publishers, but also
 - relations between those, abstract concepts, etc.
- RDF models refer to namespaces using prefixes
 - URIs that identify sets of resource, schemas, (formal) vocabularies
 - Prefixes: `xmlns:rdf = http://www.w3.org:1999/02/22-rdf-syntax-ns#`

RDF

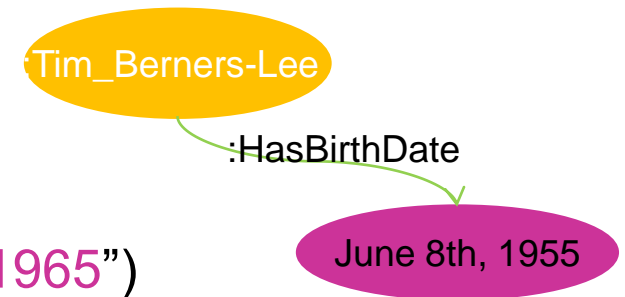
- A model made of triples
(resource1, property, resource2)



- Analogy: simple sentence

(subject, predicate, object)

(:Tim Berners Lee, :hasBirthDate, "June 8th, 1965")



- Building blocks

- Resources
- Properties
- Triples
- Graphs

RDF: a graph model

- A triple is an edge and 2 nodes
(Node1, edgeLabel, Node2)

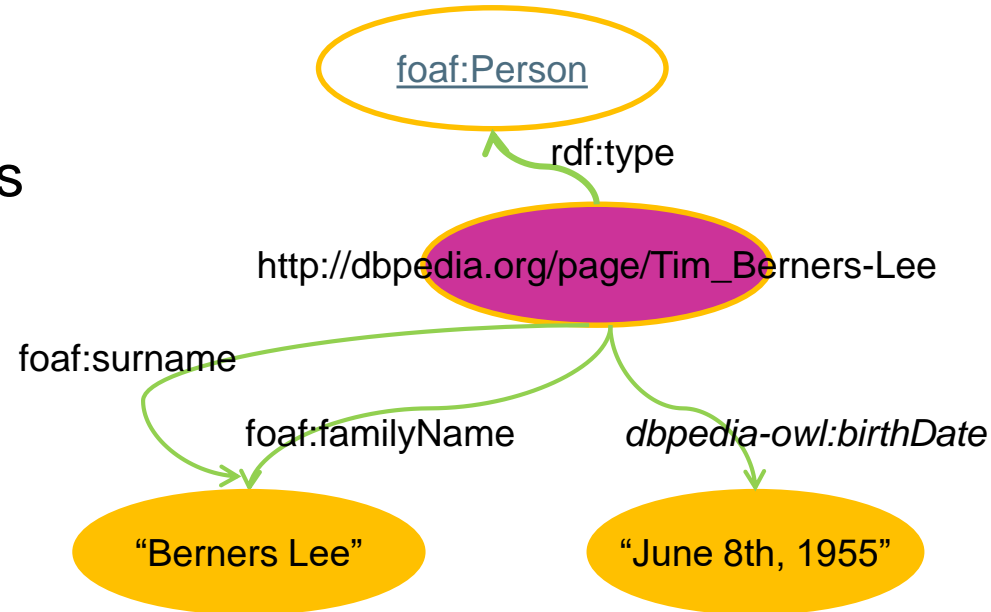
- The RDF graph is
 - a set of triples
 - multigraph
 - directed
 - labelled

- Each resource may come from a different set / web site / with its own XML namespace

`xmlns:foaf=http://xmlns.com/foaf/0.1/`

`foaf:person=http://xmlns.com/foaf/0.1/person`

`xmlns:dbpedia-owl=http://dbpedia.org/resource/classes#`



RDF: nodes

■ URI

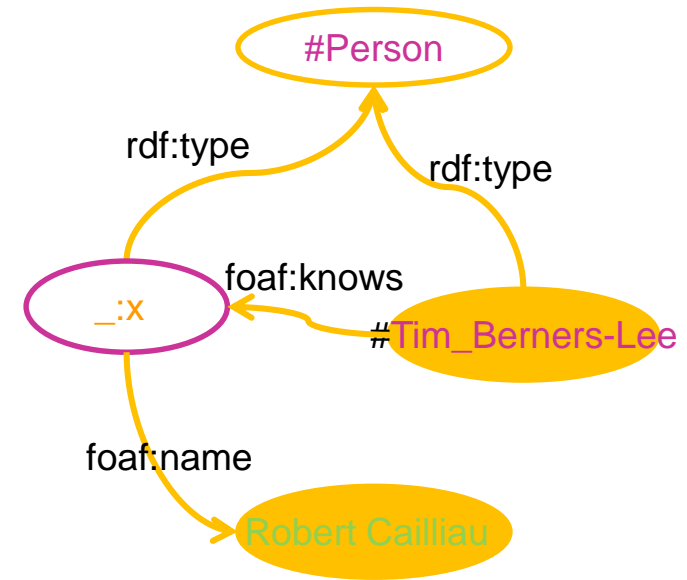
- Define your own http URI
- Use http URIs of web spaces you control
- A URI **denotes** a Resource in the world

■ Literals

- Represent data values of a datatype
- Written as strings
- Interpreted as strings if no datatype is given

■ Blank nodes

- Refer to the existence of an unknown entity
 - `_:x` `rdf:type` `#Person`



RDF: edges

■ properties

- directed
- Labelled (strings)

- Defined in a namespace

- W3C standard (`rdf:type`) or specific (`foaf:name`)

RDF XML syntax: graphs in XML trees

■ Ressources

http://dbpedia.org/page/Tim_Berners-Lee
`foaf:Person`

■ Litterals

"Berners Lee"
"June 8th, 1955"

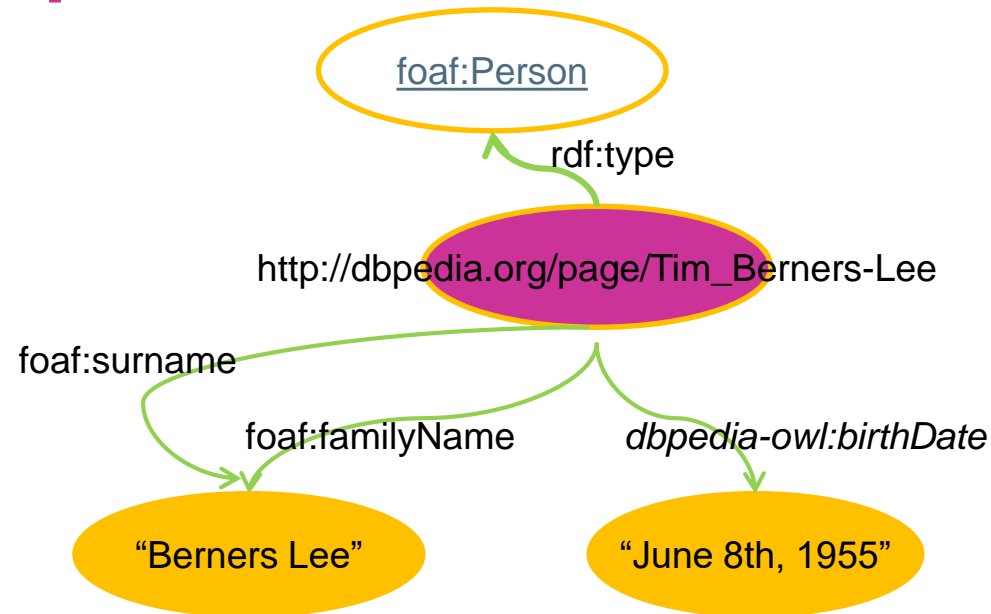
■ Predicates

`rdf:type`, `foaf:surname`
`foaf:familyName`, `dbpedia-owl:birthdate`

■ Triples

(`subject`, `predicate`, `object`)
Resource-URI Predicate-URI Resource or Literal

(http://dbpedia.org/page/Tim_Berners-Lee, `foaf:surname`, "Berners Lee")
(http://dbpedia.org/page/Tim_Berners-Lee, `foaf:familyName`, "Berners Lee")
(http://dbpedia.org/page/Tim_Berners-Lee, `rdf:type`, `foaf:Person`)
(http://dbpedia.org/page/Tim_Berners-Lee, `dbpedia-owl:birthdate`, "June 8th, 1955")



plusieurs vues sur un graphe

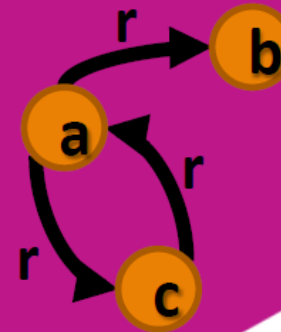
il y a énormément de façons de traiter les données RDF

Table de BDI

<i>r</i>	s	o
1	a	b
2	a	c
3	a	a

$r(a,b)$
 $r(a,c)$
 $r(c,a)$

Prédicats logiques



Graphes étiquetés

RDF XML syntax: root of the XML tree

Namespaces are properties of the RDF tag

```
<rdf:RDF  
  xmlns:rdf=http://www.w3.org:1999/02/22-rdf-syntax-ns#  
  xmlns:foaf= http://xmlns.com/foaf/0.1/  
  xmlns:dbpedia-owl= http://dbpedia.org/resource/classes#  
  xmlns:n7=http://www.irit.fr/ontologies/n7# >
```



```
<rdf:Description  
  rdf:about=''http://dbpedia.org/page/Tim_Berners-Lee''>  
  <foaf:surname> Berners Lee </foaf:surname>  
  <foaf:familyname> Berners Lee </foaf:familyname>  
  <n7:worksWith  
    rdf:resource=''http://fr.dbpedia.org/resource/Vint_Cerf'' />  
</rdf:Description>
```

```
</rdf:RDF>
```

RDF vocabulary

- Resource description

```
<rdf:Description rdf:about= "http://dbpedia.org/page/Tim_Berners-Lee">
```

```
...
```

```
</rdf:Description>
```

OR

```
<rdf:Description rdf:ID= "TB" />
```

- Absolute or relative URI identifier

- absolute

http://dbpedia.org/page/Tim_Berners-Lee

<http://www.irit.fr/ontologies/n7#TB>

- Relative

`xmlns:n7=http://www.irit.fr/ontologies/n7#`

`n7:TB :TB or #TB(local)`

- Give the description an identifier

- `rdf:about` : the resource needs to have a global identifier

- `rdf:ID` : the resource has a local identifier

RDF vocabulary : triples

```
<rdf:Description
  rdf:about='http://dbpedia.org/page/Tim_Berners-Lee'>
  <foaf:surname> Berners Lee </foaf:surname>
  <foaf:familyname> Berners Lee </foaf:familyname>
  <n7:worksWith
    rdf:resource='http://fr.dbpedia.org/resource/Vint_Cerf' />
</rdf:Description>
```

Litteral

Resource in absolute form

```
<n7:worksWith> dbpedia-fr:Vint_Cerf </n7:worksWith>
```

RDF –XML: syntactic variation

■ Resource reference: `rdf:resource`

```
xmlns:n7="http://www.irit.fr/exemple#>
<rdf:Description rdf:about="http://www.irit.fr/exemple#WebPage-nag">
  <dc:title> Page web de Nathalie Aussenac-Gilles</dc:title>
  <dc:author rdf:resource= "n7:Nathalie" />
  <n7:uri> rdf:resource= "http://www.irit.fr/~Nathalie.Aussenac-Gilles"
/>
</rdf:Description>
```

■ Description imbrication

```
<rdf:Description rdf:about="http://www.irit.fr/exemple#WebPage-nag">
  <dc:title> Page web de Nathalie Aussenac-Gilles</dc:title>
  <dc:author> <rdf:Description
    rdf:about='http://www.irit.fr/exemple#Nathalie'>
    <foaf:name> Nathalie </foaf:name>
    </rdf:Description>
  </dc:author>
</rdf:Description>
```

RDF serialization in XML

■ RDF/XML

- ❑ Widely used
- ❑ But not easily readable for humans



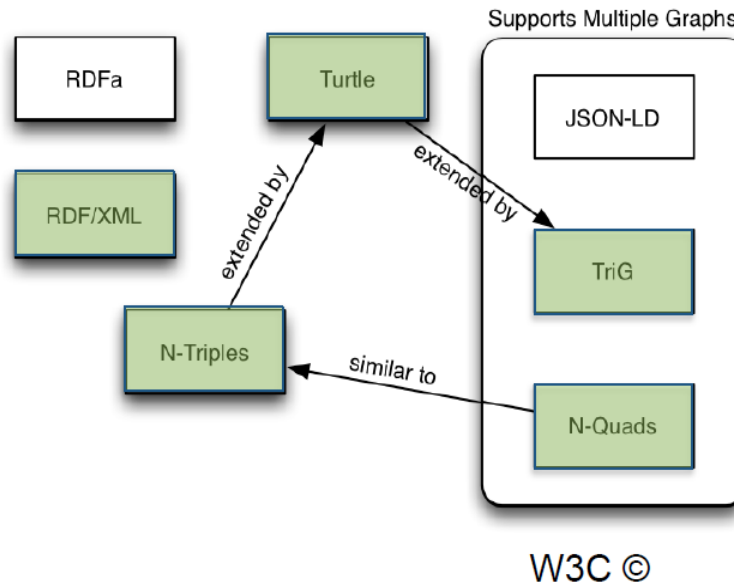
```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/elements/1.1/"
  xml:base="http://www.irit.fr/exemple"
  xmlns:n7="http://www.irit.fr/exemple#">

  <rdf:Description rdf:about="http://www.irit.fr/exemple#WebPageNag">
    <dc:title>Page web de Nathalie Aussenac-Gilles</dc:title>
    <dc:author>Nathalie Aussenac-Gilles</dc:author>
    <dc:publisher rdf:resource=" http://www.irit.fr/ " />
    <n7:URI rdf:resource="http://www.irit.fr/~Nathalie.Aussenac-Gilles" />
  </rdf:Description>

  <rdf:Description rdf:about="http://www.irit.fr/exemple#WebPageIRIT">
    <dc:title>Institut de Recherche en Informatique de Toulouse</dc:title>
    <n7:URI rdf:resource="http://www.irit.fr/~Nathalie.Aussenac-Gilles" />
  </rdf:Description>
</rdf:RDF>
```

RDF

a une syntaxe historique en XML et d'autres syntaxes : Turtle, TriG, JSON-LD, N-Triples, N-Quads



RDF: various possible notations

- RDF/XML syntax

- Easy syntaxes

- N-Triples

- TURTLE

- N3



Simple, verbose

Compact, complex

RDF: N3 or N-Triple notation

- Simple representation
- Easier to read by humans

Subject predicate object .

- Easily uploaded and read by programs

```
<http://www.irit.fr/exemple#WebPageNag>          URI
    dc:title "Page web de Nathalie Aussenac-Gilles" Literal
<http://www.irit.fr/exemple#WebPageIRIT> dc:title "Institut de
Recherche en Informatique de Toulouse" . End of Triple
```

RDF: Turtle notation

- Simple and **compact** representation
- Easier to read by humans

[illegible]

- Name spaces are declared as @prefix

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
 @prefix dc: <http://purl.org/elements/1.1/>.

Name space

URI

List of predicates -properties

List of values

End of Triple

URI

Literal

RDF:type, un prédicat particulier

■ rdf:type

prédicat réservé en RDF

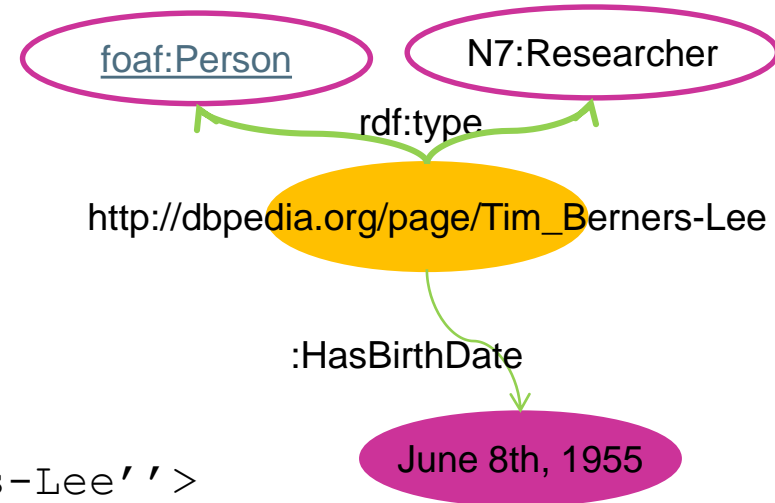
définit une ressource comme type d'une autre

■ Syntaxe RDF- XML

```
<rdf:Description rdf:about=
  ``http://dbpedia.org/page/Tim_Berners-Lee``>
  <rdf:type foaf:Person />
  <rdf:type n7:Researcher />
  <n7:hasBirthDate> "June 8th, 1965" </n7:hasBirthdate>
</rdf:Description>
```

■ RDF Turtle : **rdf:type** devient **a**

```
<http://dbpedia.org/page/Tim_Berners-Lee> a foaf:Person,
n7:Researcher .
```



RDF: types for literals

■ Literals can have a datatype

(#Tim Berners Lee, n7:hasBirthDate, “June 8th, 1965”)

```
<foaf:Person rdf:about="http://dbpedia.org/page/Tim_Berners-Lee">
<ex:hasBirthYear rdf:datatype="&xsd:gYear"> 1965</ex:hasBirthYear>
</foaf:Person>
```

```
<http://dbpedia.org/page/Tim_Berners-Lee> ex:hasBirthDate "1965-06-
08"^^xsd:date
```

■ Datatypes

- denoted by URIs
- can be freely chosen
- frequently: xsd datatypes from XML
- syntax of typed literal: “datavalue”^^datatype-URI

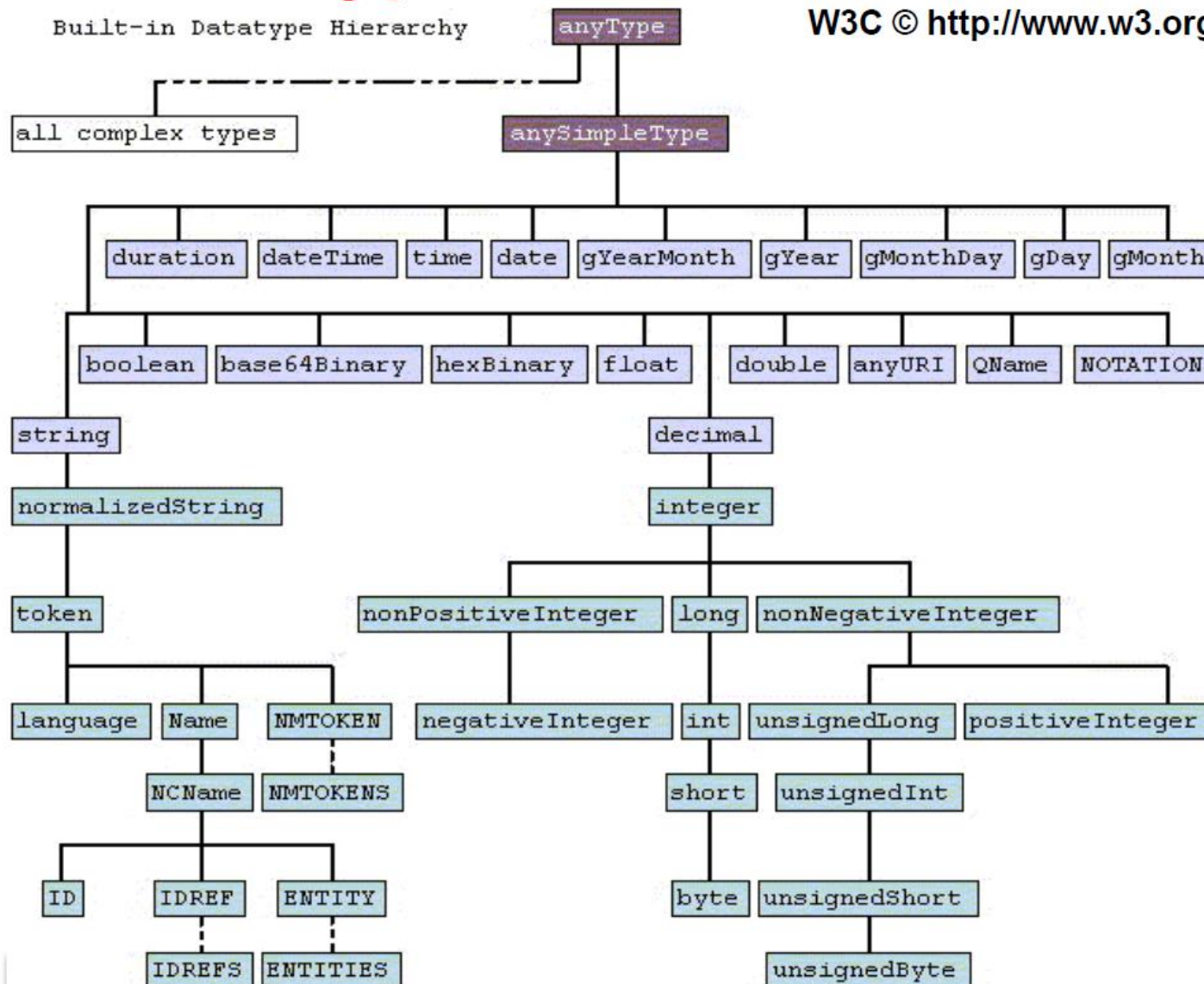
■ Literals can have several datatypes

```
<http://crcpress.com/uri> http://example.org/Name "CRC Press",
"CRC Press"@en ,
"CRC Press"^^xsd:string
```

tous les types de données des schémas

Built-in Datatype Hierarchy

W3C © <http://www.w3.org/TR/xmlschema-2/>



RDF/XML: variants to express types

```
<rdf:Description rdf:about="http://www.irit.fr/~Nathalie.Aussenac-Gilles">  
  <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Document" />  
  <dc:title>Page web de Nathalie Aussenac-Gilles</dc:title>  
  <dc:author>Nathalie Aussenac-Gilles</dc:author>  
  <dc:publisher rdf:resource="http://www.irit.fr/" />  
</rdf:Description>
```

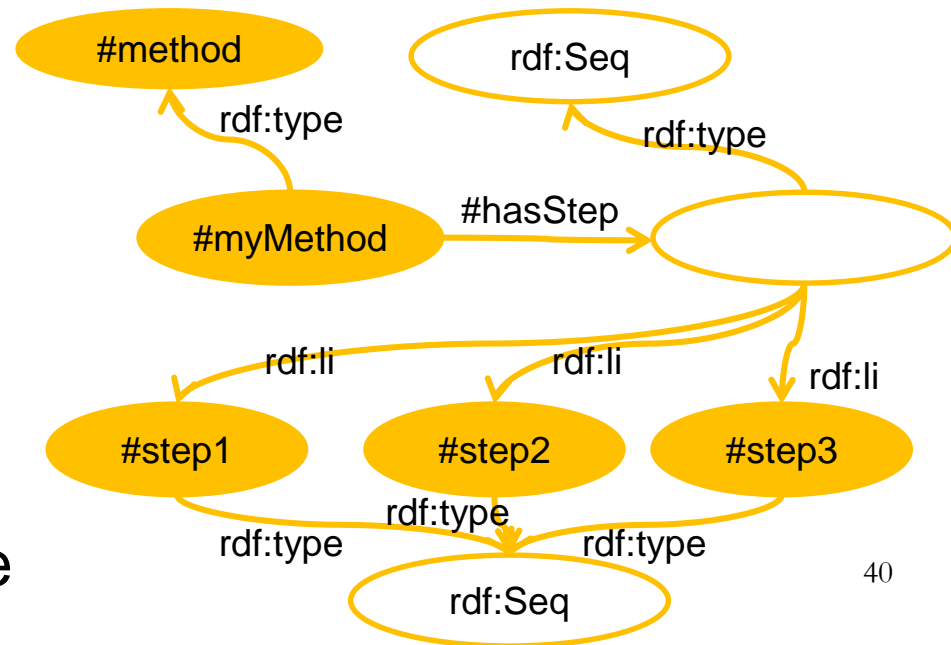


```
... xmlns: foaf ="http://xmlns.com/foaf/0.1/Document" _ ...  
<foaf:Document rdf:about="http://www.irit.fr/~Nathalie.Aussenac-Gilles">  
  <dc:title> Page web de Nathalie Aussenac-Gilles</dc:title>  
  <dc:author> Nathalie Aussenac-Gilles</dc:author>  
  <dc:publisher rdf:resource="http://www.irit.fr/" />  
</foaf:Document>
```

RDF syntax: complex data structures

■ Containers (open lists)

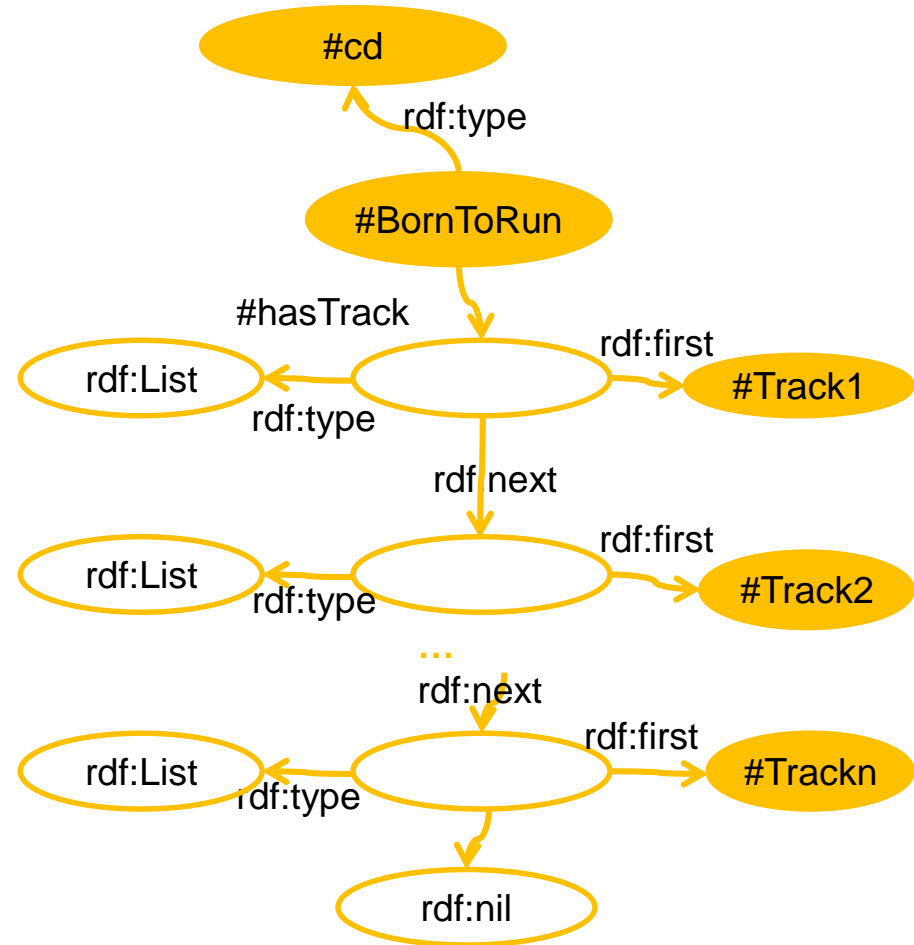
- ❑ `rdf:Bag` non ordered group
- ❑ `rdf:Seq` ordered list
- ❑ `rdf:Alt` set of alternatives or choices (select ONE)
- ❑ `rdf:li` = is member of a container
- ❑ The list of members can be extended



RDF: data structures

■ Collections (closed lists)

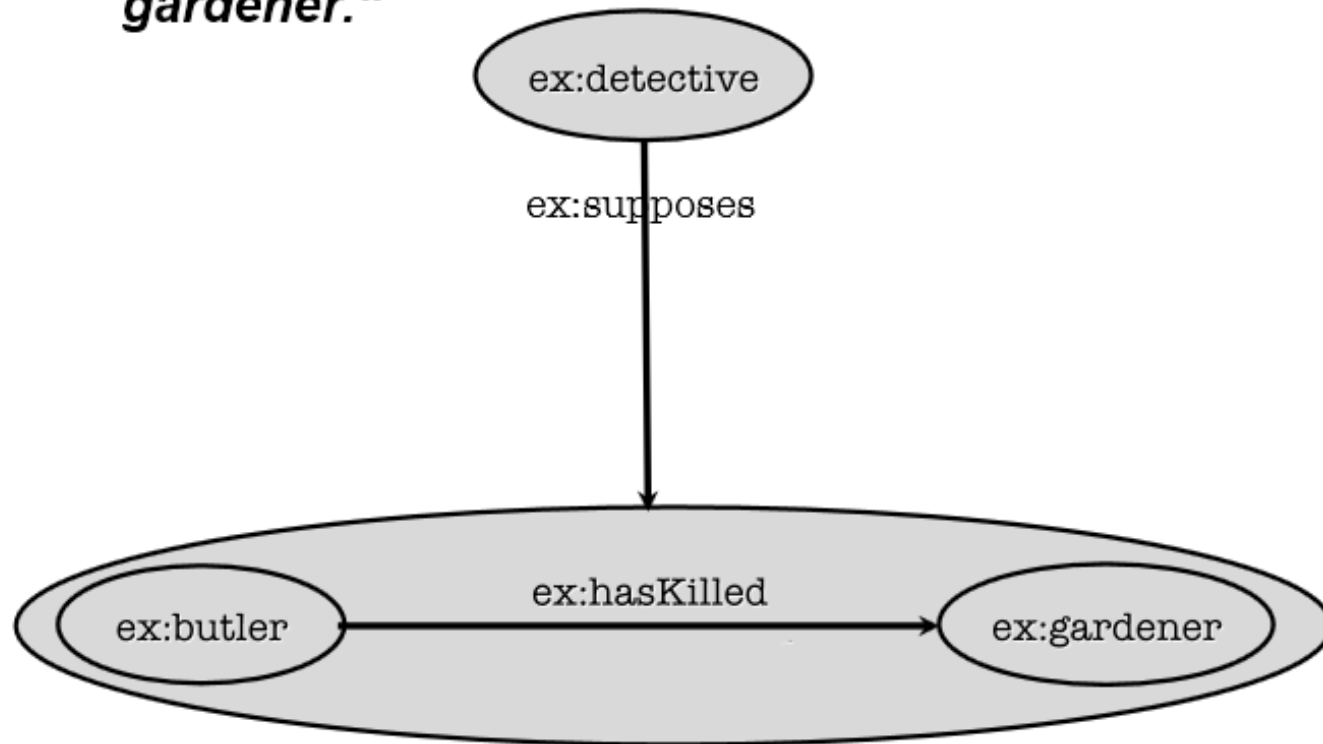
- `rdf:list`
- `rdf:first`, `rdf:rest`, `rdf:nil`
- The collection is closed to the members listed in the definition



RDF data structures

■ Reified triples

- How to model propositions about propositions such as:
„The Detective supposes that the butler killed the gardener.“




<http://www.semantic-web-book.org/w/images/4/40/SWeMoL-Part-1-IJCAI-09.pdf>

RDF data structures

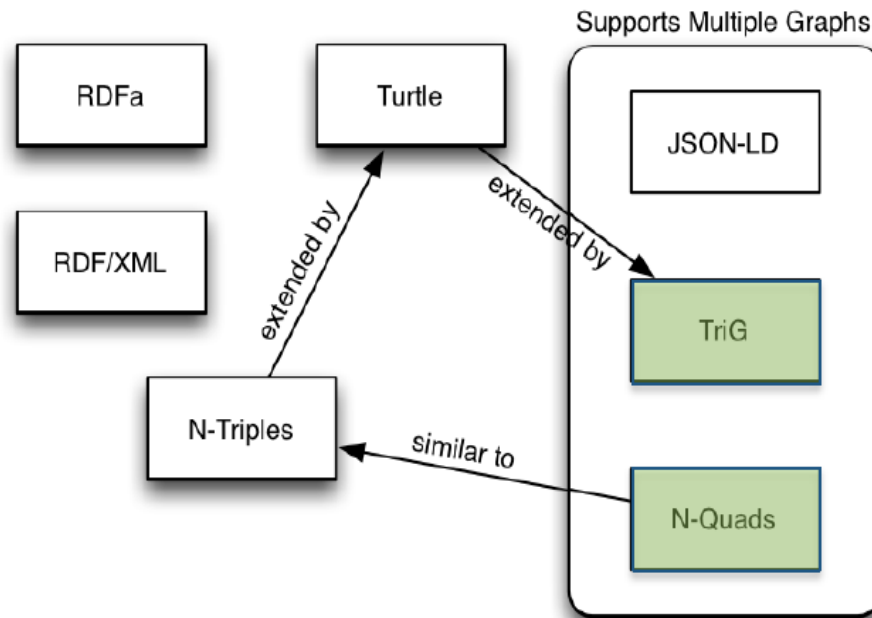
■ Solution: auxiliary node for nested proposition

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:ex="http://www.irit.fr /exemple/ex#" >
  <rdf:Description rdf:about=#detective>
    <supposes>
      <rdf:Statement rdf:about=#theory>
        <rdf:subject rdf:resource="ex:butler"/>
        <rdf:predicate rdf:resource="ex:hasKilled"/>
        <rdf:object rdf:resource="ex:gardener"/>
      </rdf:Statement>
    </supposes>
  </rdf:Description>
</rdf:RDF>
```



RDF

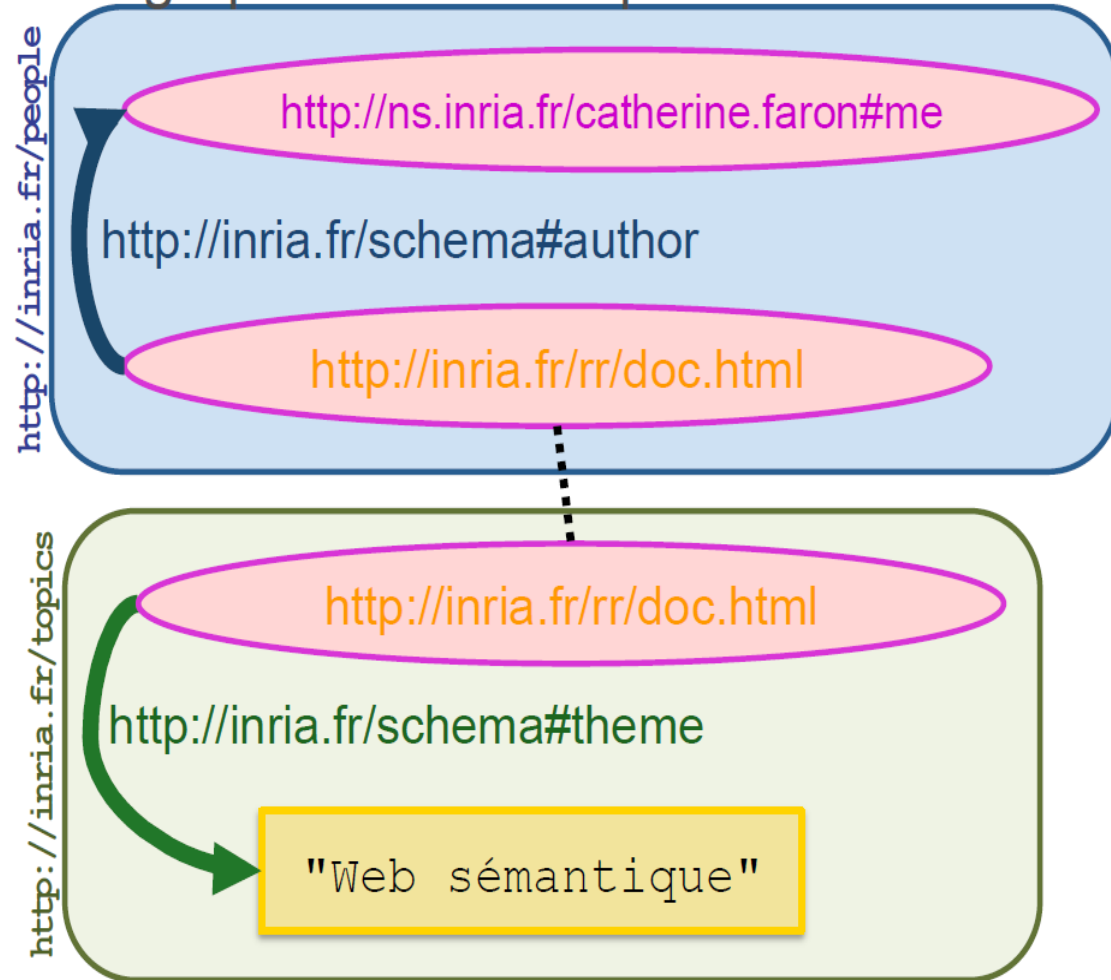
a une syntaxe TriG et une syntaxe N-Quads permettant d'exprimer des contextes



W3C ©

les graphes nommés

grouper des triplets dans des sous-graphes identifiés par des URI



RDF vocabulary

■ Classes

rdf:Description – triple description

rdf:Resource - the class resource, everything

rdf:XMLLiteral - the class of XML literal values

rdf:Property - the class of properties

rdf:Statement - the class of RDF statements

rdf:Alt, **rdf:Bag**, **rdf:Seq** - containers of alternatives, unordered containers, and ordered containers

rdf:List - the class of RDF Lists

rdf:nil - an instance of rdf:List representing the empty list

■ Properties

rdf:type - an instance of rdf:Property used to state that a resource is an instance of a class

rdf:first - the first item in the subject RDF list

rdf:rest - the rest of the subject RDF list after the first item

rdf:value - idiomatic property used for structured values

rdf:subject - the subject of the subject RDF statement

rdf:predicate - the predicate of the subject RDF statement

rdf:object - the object of the subject RDF statement

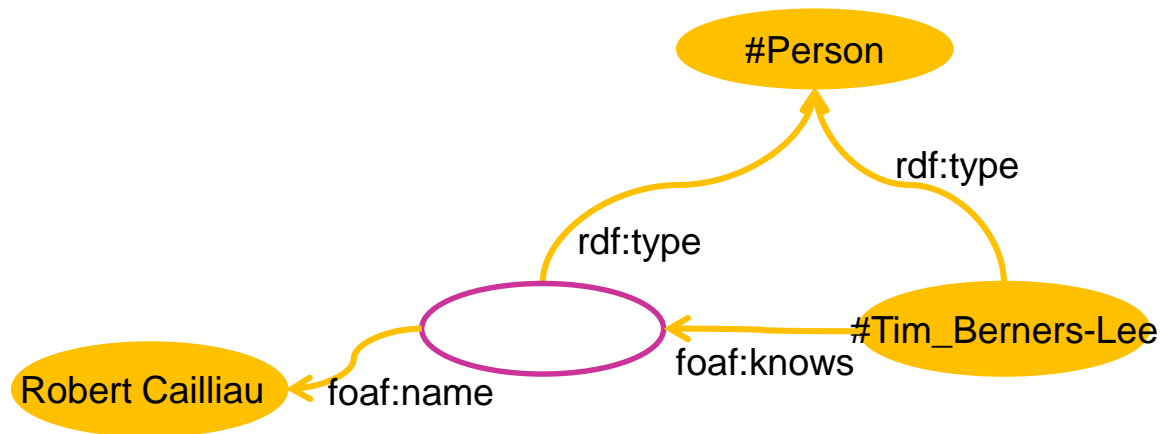
RDF semantics

■ Semantics in model theory

- Triple: $\langle s, p, o \rangle$ is associated an atomic formula $p(o, s)$
- Document: existential closure of the conjunction of atomic formulas

■ Interpretation in predicate logic

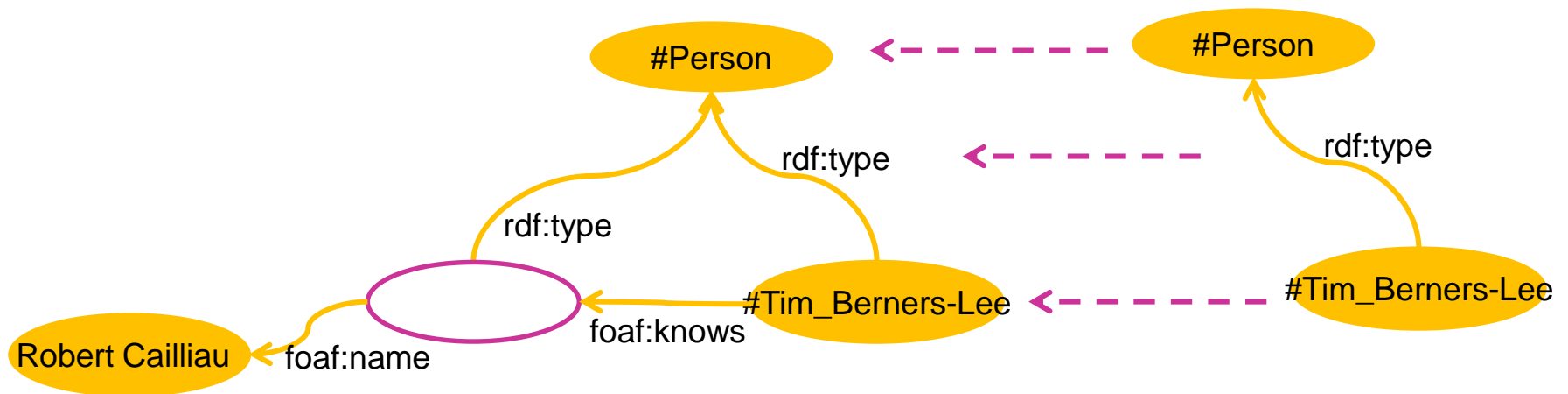
$\exists x, (\text{rdf:type}(\#Tim_Berners-Lee, \#Person) \wedge \text{rdf:type}(x, \#Person) \wedge \text{foaf:knows}(\#Tim_Berners-Lee, x) \wedge \text{foaf:name}(x, \text{"Robert Cailliau"}))$



RDF semantics

■ Reasoning mechanism

- An RDF document is the consequence of another RDF document iff there is a graph homomorphism between them
- Goal = identifies all the consequences of a set of predicates



RDF: strengths and limitations

- compared with DB
 - better manages semi-structured data
 - “universal”
 - Adapted to the web
 - But very verbose, little efficiency for storage and querying
- Semantics
 - Logical model / model theory

RDF and the semantic web

- Difficulties with meta-data
 - May be missing
 - May be conflicting or erroneous
 - Reflect a point of view
 - Refer to local / specific types
- Difficulties to query different structural options

```
<foaf:Person rdf:about="http://dbpedia.org/page/Tim_Berners-Lee">  
  <ex:hasActivity rdf:resource=#researcher"/>  
</foaf:Person>
```

```
<foaf:Person rdf:about="http://dbpedia.org/page/Marie_Curie">  
  rdf:type rdf:resource=#researcher"  
</foaf:Person>
```

```
<ex:researcher rdf:about="http://dbpedia.org/page/Albert_Einstein" />  
which persons are researchers ?
```

From RDF to RDFS

- **Need** to define properties for generic groups of individuals, such as the class of publishers, of organizations, or of persons
- **Solution:**
 - (XML) schema knowledge
 - RDF Schema (RDFS): part of the RDF W3C recommendation
 - Adequate for simple models, hierarchies of classes and instances