

10/03/2009



# Paris JUG

[www.parisjug.org](http://www.parisjug.org)

[www.parisjug.org](http://www.parisjug.org)





10/03/2009

# Les technologies du Web Sémantique

De TBL au Sudoku

Alexandre Bertails  
Atos Origin  
Open Source Center  
alexandre@bertails.org





Métadonnées

RDFS

Web sémantique

Taxinomie

Ontologie

Intelligence artificielle

Bases de données

Triples

Web



URI

SPARQL

RDFa

SKOS

Thésaurus

Dublin Core

RDF/XML

Graphes

OWL

NTriples

RDF

TAL

Logique de description

Vocabulaires contrôlés



**Nature du Web sémantique**

Concept  
Activité du W3C  
Ensemble de technologies  
Extension du Web

**Gourou du Web Sémantique**

Tim Berners-Lee

**Premières « working draft » au W3C**

2 octobre 1997

**Acte de naissance officielle**

Article de Tim Berners-Lee, Ora Lassila et James Hendler dans la revue *Scientific American* en 2001

**Rapport avec la « sémantique »**

Presque aucun...

**Synonymes**

Web of data, Web de données, linked data, Web 3.0



Tim Berners-Lee, père fondateur du Web et du Web Sémantique, W3C

« The Semantic Web is a **web of data**, in some ways like a **global database** »

« The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation. »

« [Le Web Sémantique] offrira une **interopérabilité inégalée**, c'est-à-dire une **capacité à faire partager l'accès aux données** qui n'existe pas aujourd'hui. »



Robert Shimp, Vice-président division « Global technology business », Oracle

« But **the second big trend then is to decouple the data from the application** or the application services, so that in that sense what you can do is write your application or **create services independent of the data sources** they have to deal with, which comes full circle back to having a virtual layer between application services and data. The application can go out and find whatever data sources are best to use for that particular question. **That's what semantic technology provides for enterprise information management.** »



David Beckett, Software architect, Yahoo!

« The reason I got involved with the Semantic Web was...**I wanted control of my data.** »



Pablo Castro, Technical Lead  
Microsoft Corporation

Orthogonal to single-data source web applications and their architecture, a new class of application has also emerged on the web, called mashups. Mashups are front-ends that **aggregate and combine data that is available in a "pure data form" on the web.** [ADO.NET Data Services *aka* Astoria]

IMM introduces [...] the Resource Description Framework (RDF) and Web Ontology Language (OWL) [...]. This allows companies to **add nuance and intelligence to media management beyond what is possible with traditional metadata.** [Interactive Media Manager]



Gautier Poupeau, Consultant architecte de données  
Atos Origin - Enterprise Information Management

« Libérez la donnée, ne la confiez pas à un informaticien ! »

« Dans un système d'information, où se situe la valeur de l'organisation : dans l'information qui le compose ou dans l'application qui manipule l'information ? »



**R**esource

d'une ressource

**D**escription

de Description

**F**ramework

Cadre/modèle



Les trois composants de

RDF

sont :

- les URI pour identifier ;
- les triplets pour exprimer ;
- les graphes pour relier



**1- Prenez des choses/concepts/entités du monde réel et placez les dans le monde numérique, en les nommant et en les identifiant par des URIs**

**Signifié  
dans le monde réel**

Tim Berners-Lee

Une personne

L'article « Semantic Web » de 2001

est

Un texte

**Signifiant  
dans le monde numérique**

<http://www.w3.org/People/Berners-Lee/card#i>

<http://xmlns.com/foaf/0.1/Person>

<http://www.sciam.com/article.cfm?id=the-semantic-web>

<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>

<http://purl.org/dc/dcmitype/Text>

**Puisque ces choses/concepts/entités sont identifiés par une URI,  
ils sont assimilables à des ressources (RFC 3986)**



## 2- Exprimez des relations entre ces ressources sous la forme de triplets

Structure d'un triplet RDF = Structure d'une phrase simple

Sujet Verbe Complément = (Sujet, Prédicat, Objet)

L'article « Semantic Web » de 2001 est un texte

(<http://www.sciam.com/article.cfm?id=the-semantic-web>, <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>, <http://purl.org/dc/dcmitype/Text>)

L'article Semantic Web de 2001 a pour créateur Tim Berners-Lee

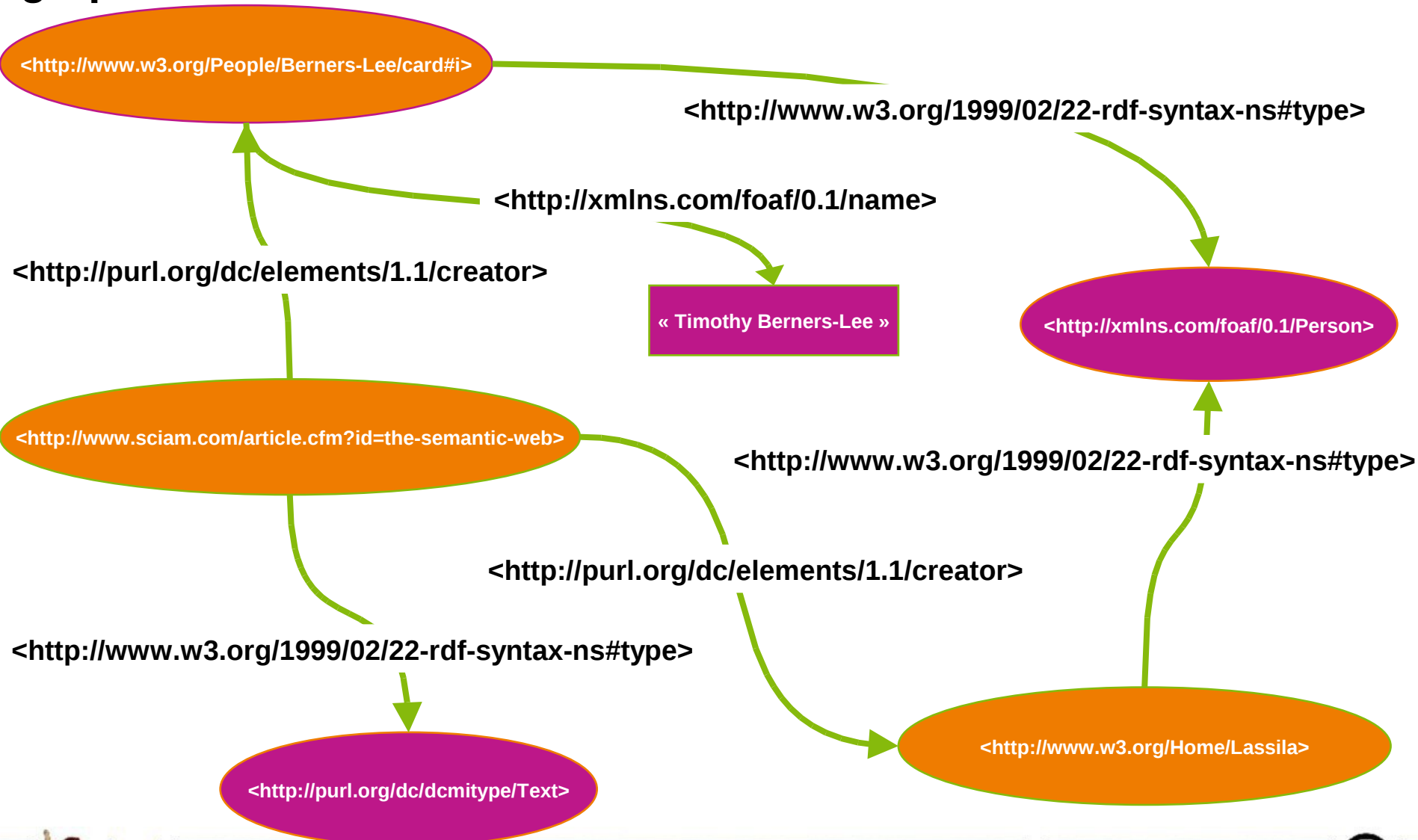
(<http://www.sciam.com/article.cfm?id=the-semantic-web>, <http://purl.org/dc/elements/1.1/creator>, <http://www.w3.org/People/Berners-Lee/card#i>)

Tim Berners-Lee est une personne

(<http://www.w3.org/People/Berners-Lee/card#i>, <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>, <http://xmlns.com/foaf/0.1/Person>)



## 3- Représentez et reliez les triples sous forme de graphes orientés





Sérialiser du

RDF

ou

les grammaires du Web Sémantique



```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix dc: <http://purl.org/dc/terms/> .
<http://www.w3.org/People/Berners-Lee/card#i> rdf:type foaf:Person ;
    foaf:name "Timothy Berners-Lee" ;
    dc:creator <http://www.w3.org> ;
    foaf:nick "timbl" .
```



```
<rdf:RDF
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/terms/"

  <rdf:Description rdf:about="http://www.w3.org/People/Berners-Lee/card#i">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
    <foaf:name>Timothy Berners-Lee</foaf:name>
    <dc:creator rdf:resource="http://www.w3.org"/>
    <foaf:nick>timbl</foaf:nick>
  </rdf:Description>

</rdf:RDF>
```



```
<rdf:RDF
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/terms/"

  <foaf:Person rdf:about="http://www.w3.org/People/Berners-Lee/card#i"
    foaf:name="Timothy Berners-Lee">
    <dc:creator rdf:resource="http://www.w3.org"/>
    <foaf:nick>timbl</foaf:nick>
  </foaf:Person>

</rdf:RDF>
```



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+RDFa 1.0//EN"
"http://www.w3.org/MarkUp/DTD/xhtml-rdfa-1.dtd">
```

```
<html
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:dc="http://purl.org/dc/terms/"
  <head>
    <title>Description de Timbl en XHTML + RDFa</title>
  </head>
  <body>
    <div typeof="foaf:person" about="http://www.w3.org/People/Berners-Lee/card#i">
      Le <span rel="dc:creator" href="http://www.w3.org">créateur du W3C</span>
      a pour nom <span property="foaf:name">Timothy Berners-Lee</span>
      et pour surnom <span property="foaf:nick">Timbl</span>.
    </div>
  </body>
</html>
```



# RDFS et OWL

Créer et partager un vocabulaire commun

à l'échelle du Web

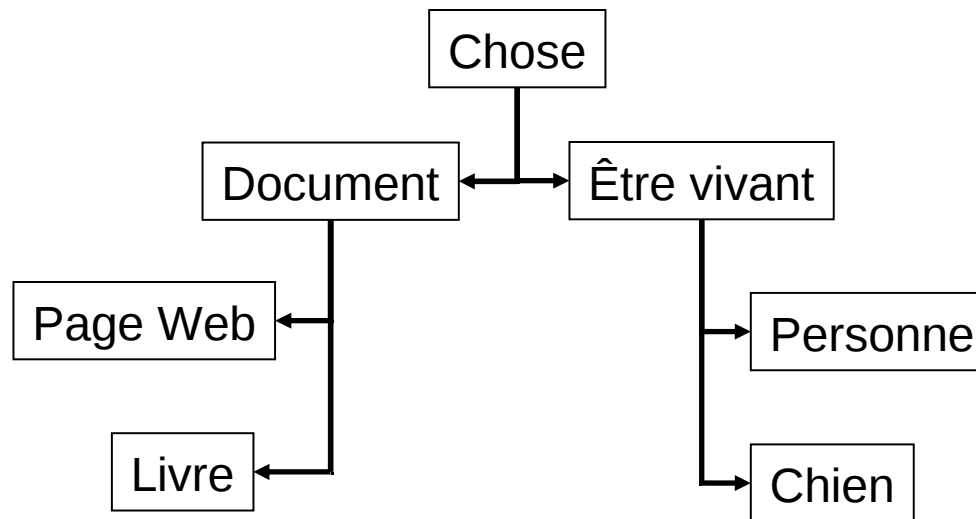


Dans notre monde, chaque chose a une nature, un type.

Exemples :

- Tim Berners-Lee est une personne ;
- Une personne est un être vivant ;
- <http://www.bnf.fr> est une page Web
- Une page Web est un document
- ...

Dans le monde RDF, la nature d'une chose/ressource est une « classe »



En RDF, une ressource appartient toujours à une classe.

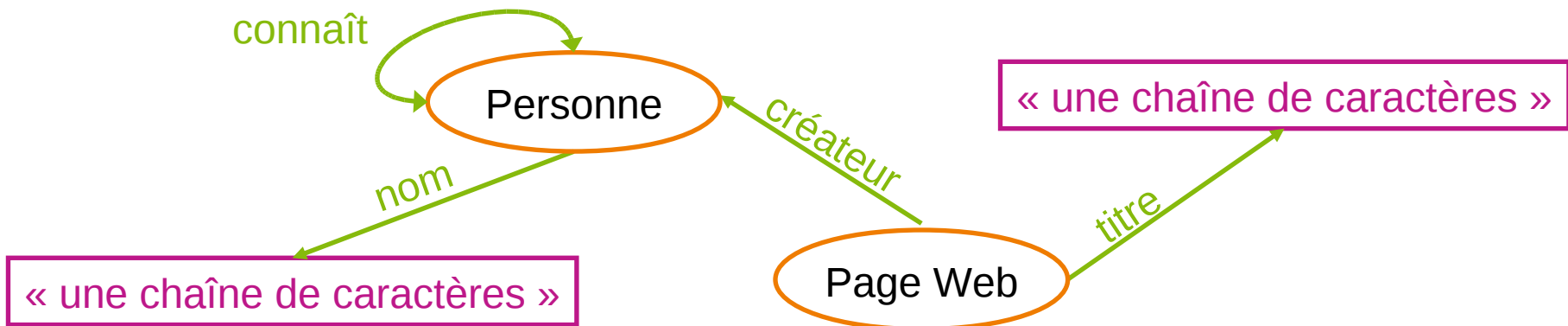


Dans notre monde, chaque type de choses possède des caractéristiques.

Exemples :

- une personne a un nom ;
- une personne connaît d'autres personnes ;
- une page Web a un titre ;
- une page Web a un créateur ;
- ...

Dans le monde RDF, ces caractéristiques sont des propriétés.



En RDF, un prédicat est une propriété définie dans un vocabulaire.



Dans notre monde, chaque caractéristique et chaque type possèdent une logique interne.

Exemple :

- Si un homme A est le frère d'un homme B, alors l'homme B est le frère de la personne A ;
- ...

Dans le monde RDF, cette logique est exprimée clairement dans le vocabulaire.



Est frère de est une propriété symétrique



À partir de RDF et OWL, la logique nous permet d'inférer (*i.e.* déduire) de nouvelles informations

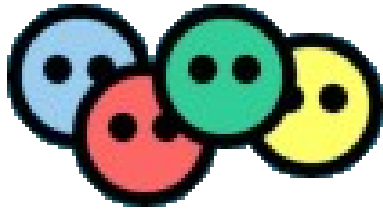


J'ai défini :

- des classes
- des propriétés
- une logique

## J'ai créé une ontologie !





FOAF



Dublin Core



Basic Geo (WGS84 lat/long)





# SPARQL

SPARQL Protocol and RDF Query Language

Interroger, accéder, transporter  
les données en RDF

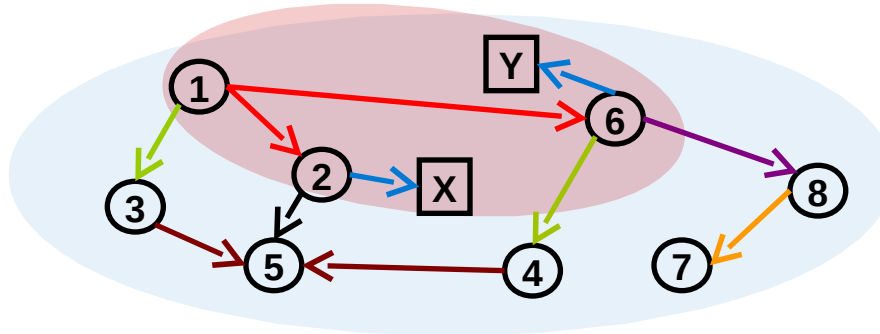
Un langage de requêtes

Un protocole d'accès

Un format XML pour les résultats



Soit le graphe suivant enregistré dans une base de données RDF (un triple store) :



SPARQL permet d'extraire un sous-ensemble de ce graphe par l'expression de contraintes sous la forme d'équations

## Exemple :

Je cherche les ressources liées à 1 par prédicat « rouge » et la chaîne de caractères liée à ces ressources par le prédicat « bleu » :

1. Les ressources liées à 1 par le prédicat « rouge » (`<1> <rouge> ?resources`)
2. La chaîne de caractères liée à ces ressources par le prédicat « bleu » (`?resources <bleu> ?string`)



Dans dbpedia.org,

je veux connaître l'URI et le lieu de naissance de Barack Obama :

Barack Obama est né à un endroit

un endroit a pour nom ???



```
PREFIX dbpedia:<http://dbpedia.org/resource/>
PREFIX dbpedia-owl:<http://dbpedia.org/ontology/>
PREFIX property:<http://dbpedia.org/property/>

SELECT ?place ?placename

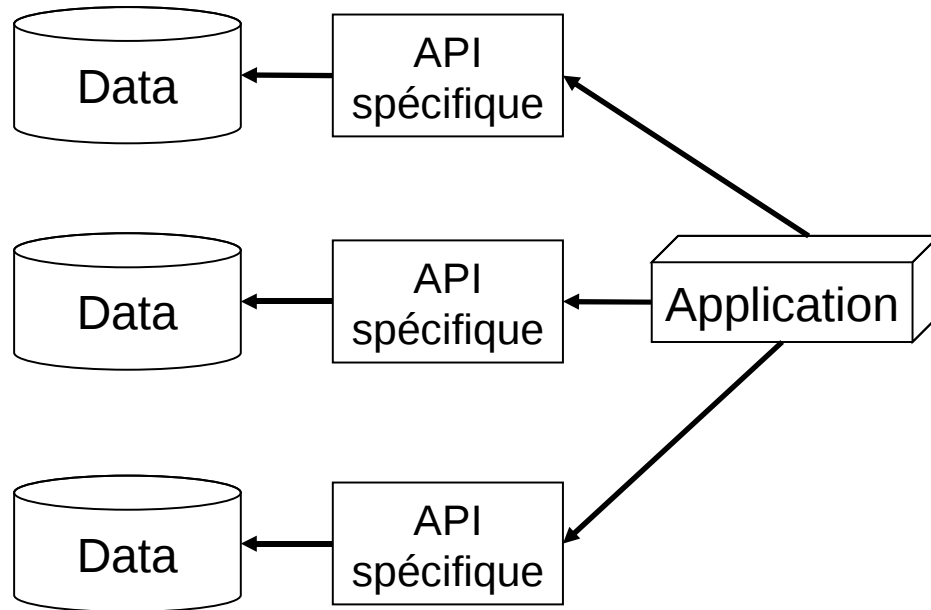
WHERE {
    dbpedia:Barack_Obama dbpedia-owl:birthplace ?place .
    ?place property:name ?placename .
}
```



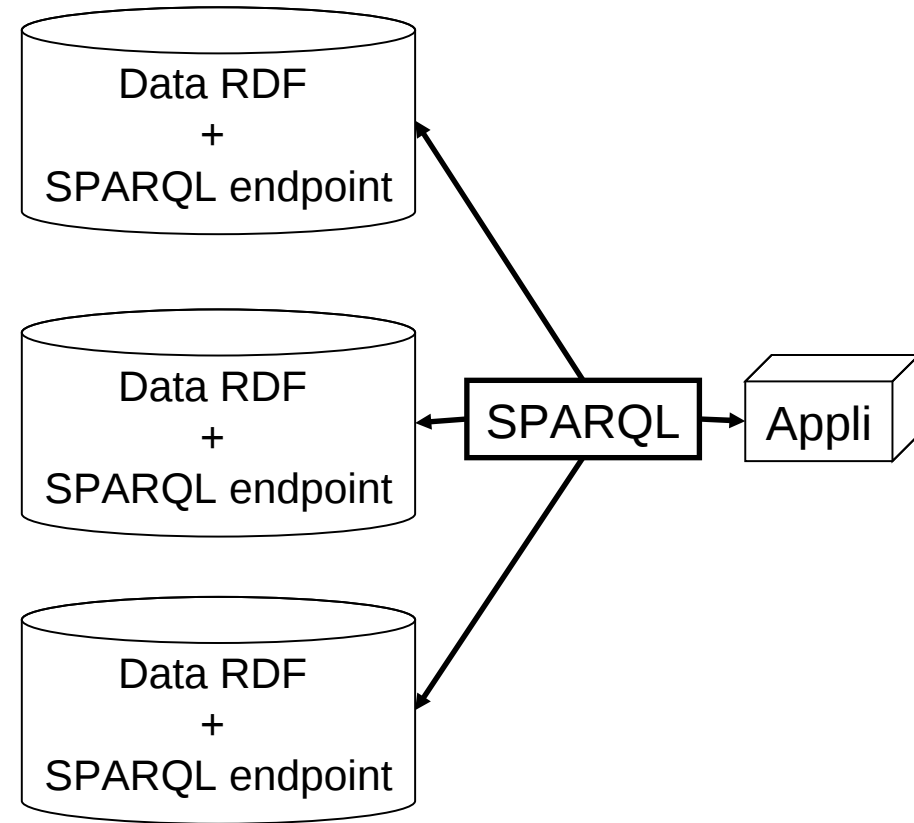
```
@POST
@Produces({"application/sparql-results+xml", "application/rdf+xml"})
public String sparqlEndPoint(
    @FormParam("query") String query,
    @FormParam("default-graph-uri") List<String> defaultGraphUri,
    @FormParam("named-graph-uri") List<String> namedGraphUri);
```



## Sans RDF ni SPARQL



## Avec RDF et SPARQL



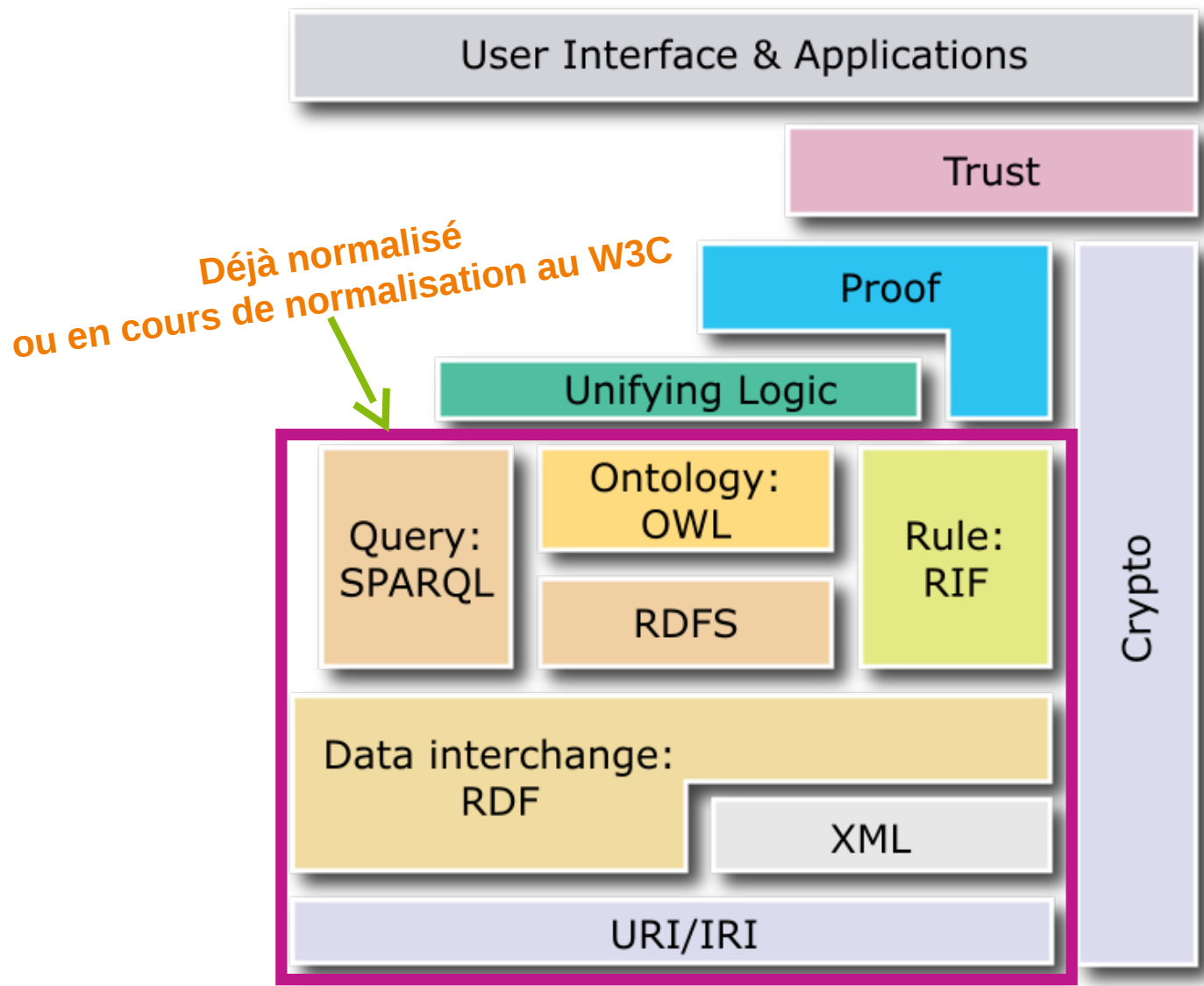


## Comparaison du modèle **RDF**

avec d'autres modèles  
d'organisation de l'information,

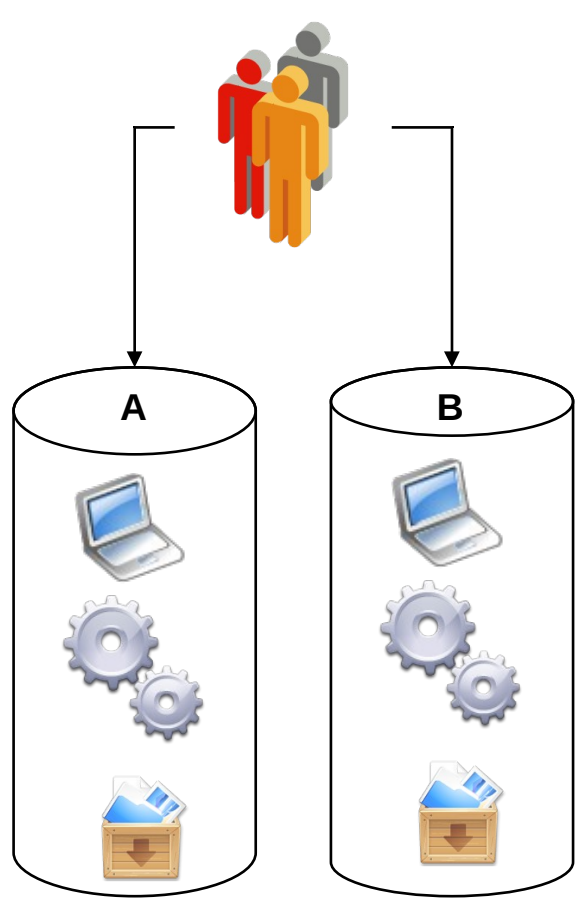
**XML**

**SGBD**

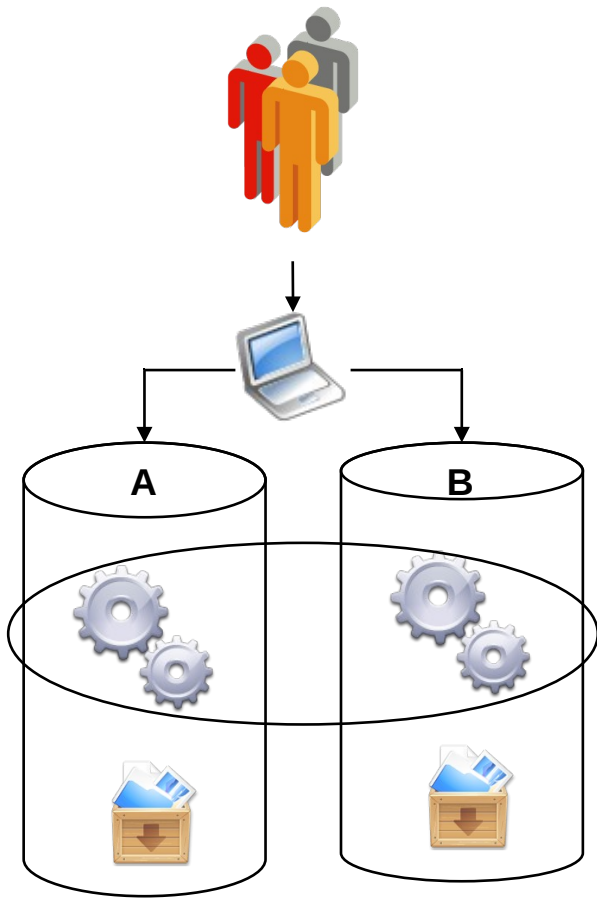




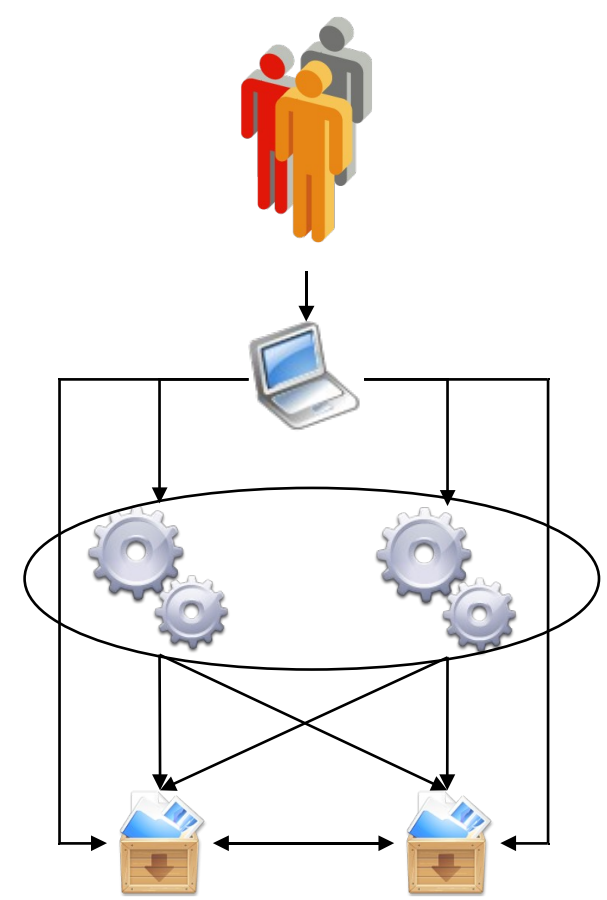
- Créer des vocabulaires/ontologies avec le logiciel [Protege](#)
- Mettre les données au format RDF
  - Créer des données RDF avec un éditeur RDF ([Morla](#) ou [Top Braid](#) composer) ou un éditeur XML
  - Transformer des données de XML vers RDF/XML avec XSL
  - Transformer une base de données relationnelle en RDF avec [D2R server](#)
- Stocker les données dans un triple store RDF
  - Triple store natif : [Mulgara](#), [AllegroGraph](#), [BigOWLIM](#)
  - BDR paramétrée : [Virtuoso](#), [ARC](#), [Oracle 11g](#), [Sesame](#), [3store](#)
  - Column store : [Cstore](#), [Heart](#), [BigData](#)
- Exploiter les données en RDF
  - En Java (triples) : [Jena](#), [Sesame](#), [Trialex SCB](#), [RDF2Go](#)
  - En Java (ORM) : [Topaz](#), [RDFReactor](#), [So\(m\)mer](#), [Elmo](#), [jenabean](#)
  - En PHP : [RAP](#), [ARC](#) (généraliste)
  - En C : [Redland](#)
  - En Python : [RDFlib](#)
  - En Ruby : [ActiveRDF](#)



Silos applicatifs indépendants et non connectés



Silos de services



Indépendance aux trois niveaux (applicatif, service, données)



## Comparaison du modèle **RDF**

avec d'autres modèles  
d'organisation de l'information,

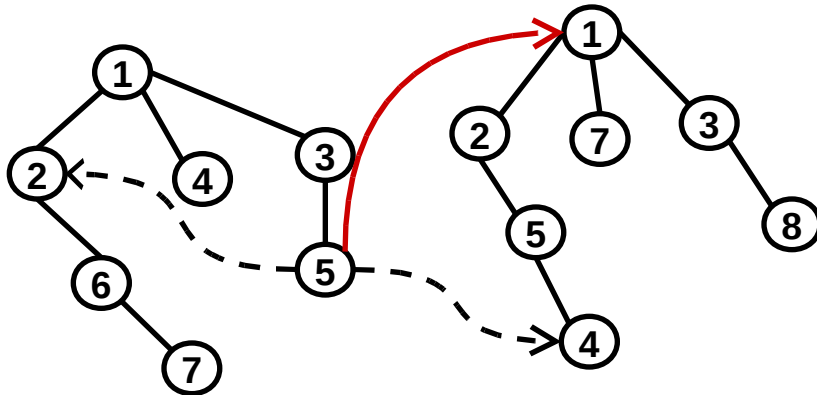
**XML**

**SGBD**



VS.

## XML : un modèle d'arbre

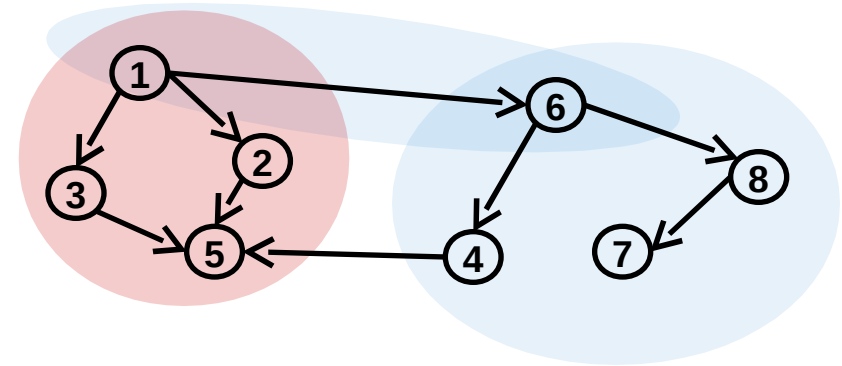


### Description structurée

- La structure est pensée pour le contexte du **document** (interopérabilité très complexe)
- Relations entre les nœuds implicites
- Navigation dans l'arbre pour passer d'un nœud à un autre
- L'identification des ressources est spécifique au contexte du document

**Idéal pour exprimer la structure d'un document**

## RDF : un modèle de graphes



### Description formelle

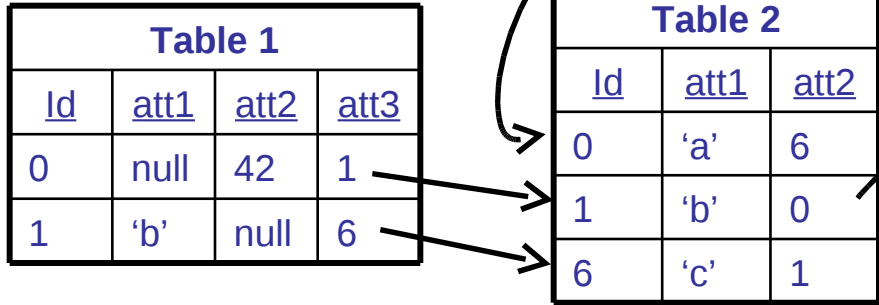
- Les choses sont décrites comme des objets logiques
- Aucune relation n'est implicite
- Plus facile de naviguer dans le graphe si les données sont fortement reliées
- Toutes les ressources sont identifiées par des URI

**Idéal pour décrire des données**



VS

## RDBMS : un modèle de tables

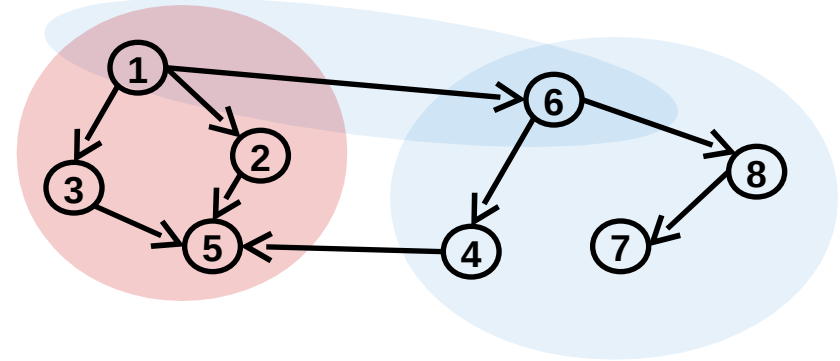


## Description orientée enregistrements

- Le modèle relationnel fait la différence entre les relations internes (attributs) et externes (clés)
- Structure rigide : valeurs absentes explicites, pas de possibilité de répéter un champ pour une notice
- Modèle centralisé : les identifiants d'enregistrements ne sont valables que pour la base de données
- Modèle logique et modèle physique confondu (cf. la forme et l'utilisation de SQL)

**Idéal pour des relations simples, figées et nécessitant un contrôle d'intégrité « physique »**

## RDF : un modèle de graphes



## Description orientée triple pérenne

- Les relations font partie des données
- Chaque triplet est autonome, absolu et pérenne
- Contrôle d'intégrité par inférence
- Modèle distribué
- Séparation entre modèle logique et modèle physique (Cf. RDBMS2RDF et SPARQL)

**Idéal pour décrire des données avec haut niveau d'organisation, réparties et pérennes**



Paris  
JUG

# *Démonstration SPARQL*



[www.parisjug.org](http://www.parisjug.org)

Copyright © 2008 ParisJug. Licence CC – Creative Commons 2.0 France – Paternité – Pas d'Utilisation Commerciale – Partage des Conditions Initiales à l'Identique





Dans dbpedia :

@prefix dbpedia:<<http://dbpedia.org/resource/>>

@prefix dbpedia-owl:<<http://dbpedia.org/ontology/>>

@prefix p:<<http://dbpedia.org/property/>>

Qui est Barack ?

<[http://dbpedia.org/resource/Barack\\_Obama](http://dbpedia.org/resource/Barack_Obama)>

Où est-il né ?

<<http://dbpedia.org/ontology/birthplace>>

Quand est-il né ?

<<http://dbpedia.org/ontology/birthdate>>

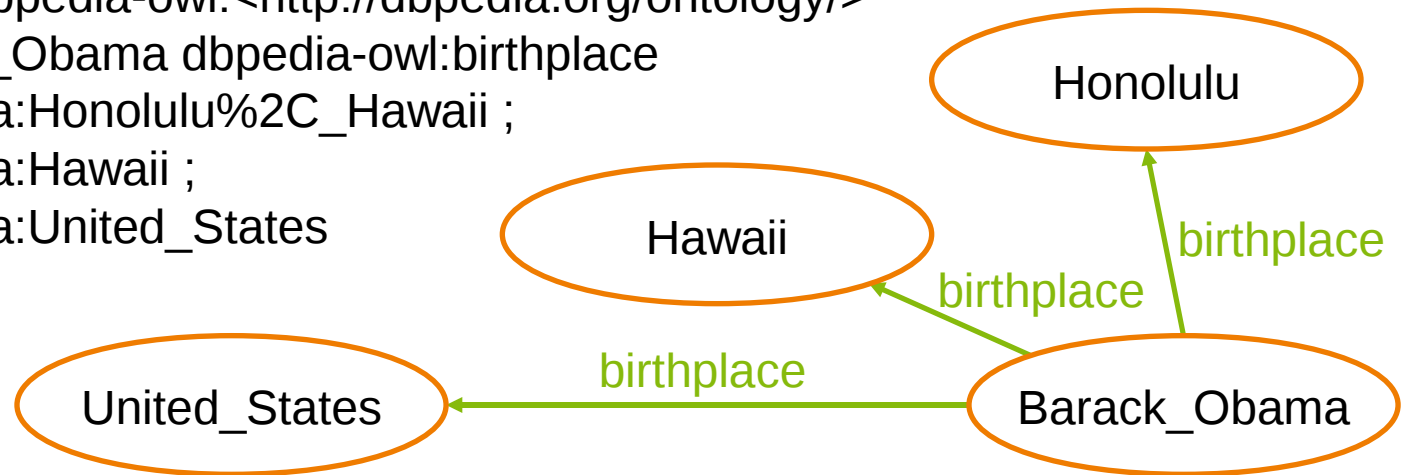
Quelle profession exerce-t-il ?

<<http://dbpedia.org/property/profession>>

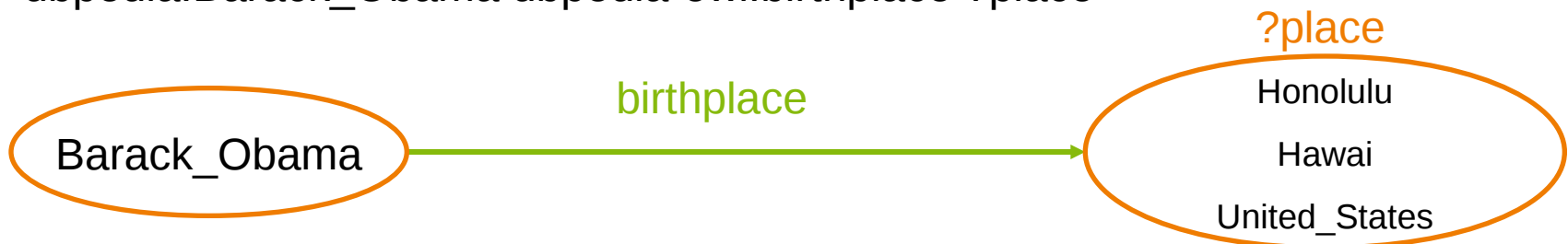


Quelque part dans dbpedia...

```
@prefix prefix dbpedia:<http://dbpedia.org/resource/> .  
@prefix prefix dbpedia-owl:<http://dbpedia.org/ontology/>  
dbpedia:Barack_Obama dbpedia-owl:birthplace  
dbpedia:Honolulu%2C_Hawaii;  
dbpedia:Hawaii ;  
dbpedia:United_States
```

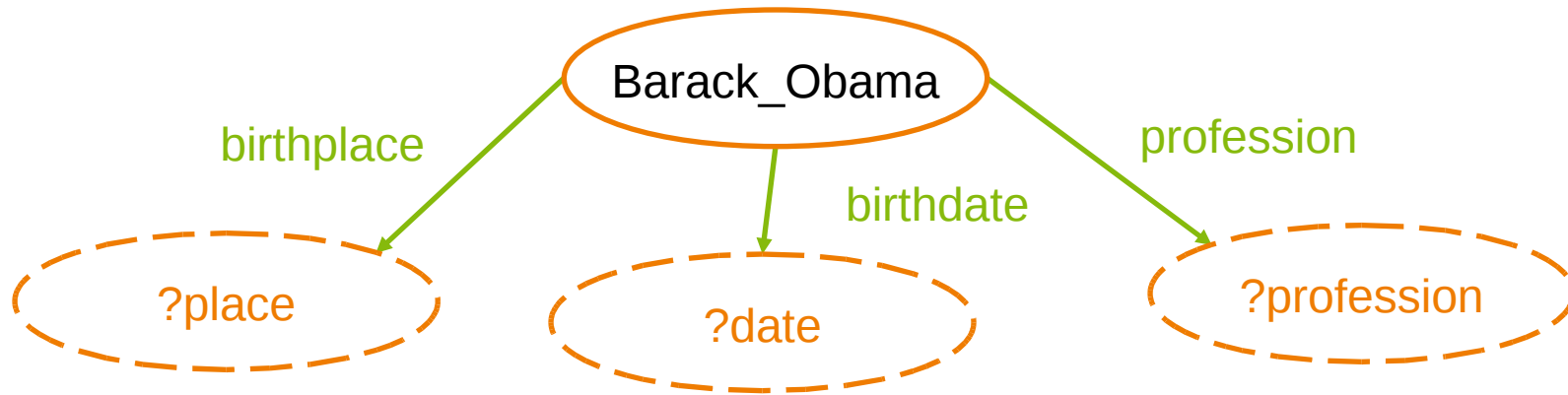


Variabilisons l'ensemble des lieux de naissance :  
dbpedia:Barack\_Obama dbpedia-owl:birthplace ?place

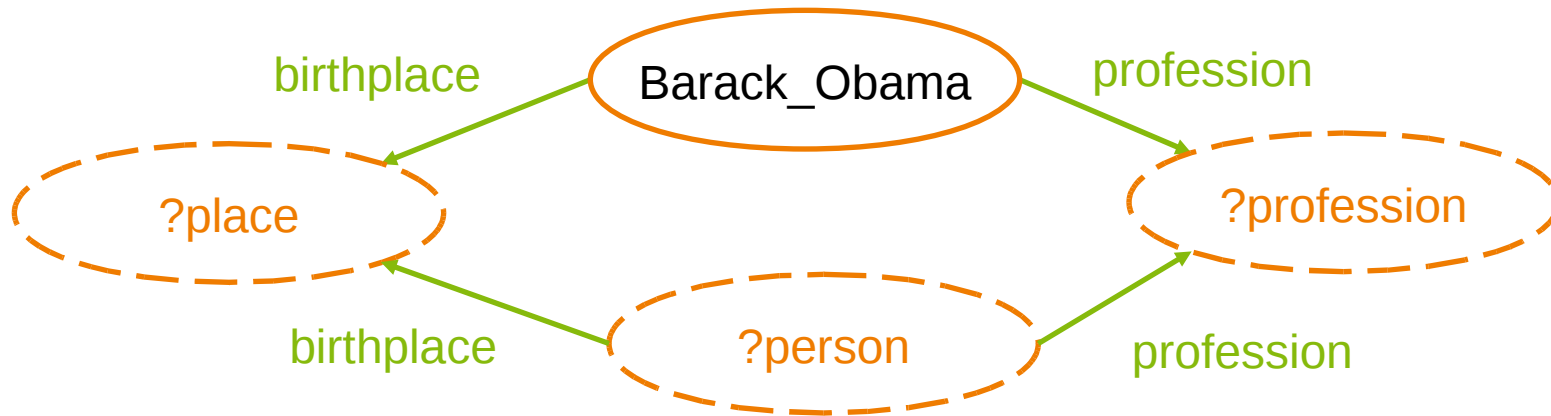




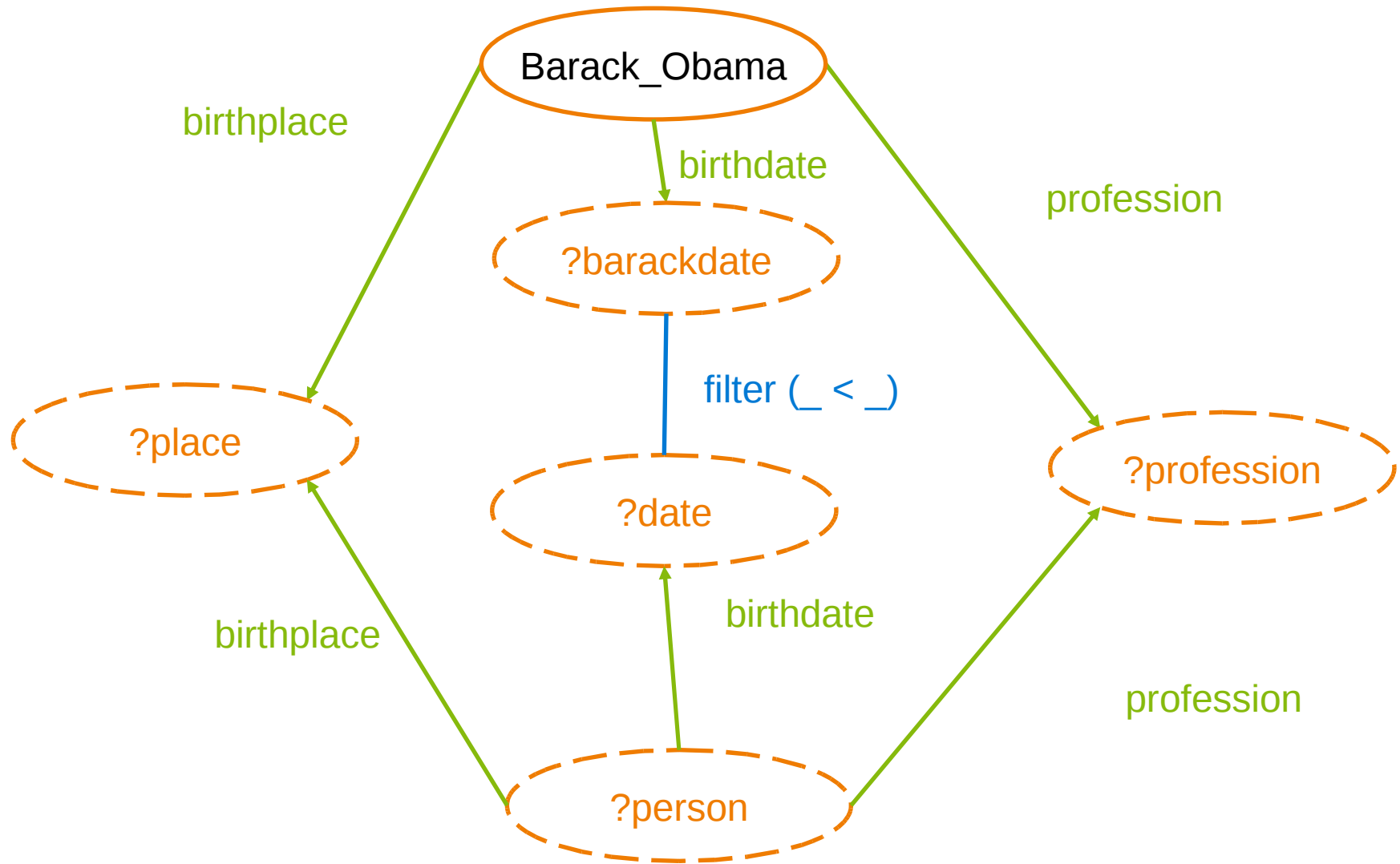
```
PREFIX dbpedia:<http://dbpedia.org/resource/>
PREFIX dbpedia-owl:<http://dbpedia.org/ontology/>
SELECT ?place WHERE {
  dbpedia:Barack_Obama dbpedia-owl:birthplace ?place .
}
```



```
PREFIX dbpedia:<http://dbpedia.org/resource/>
PREFIX dbpedia-owl:<http://dbpedia.org/ontology/>
PREFIX p:<http://dbpedia.org/property/>
SELECT ?place ?date ?profession WHERE {
  dbpedia:Barack_Obama dbpedia-owl:birthplace ?place .
  dbpedia:Barack_Obama dbpedia-owl:birthdate ?date .
  dbpedia:Barack_Obama p:profession ?profession .
}
```

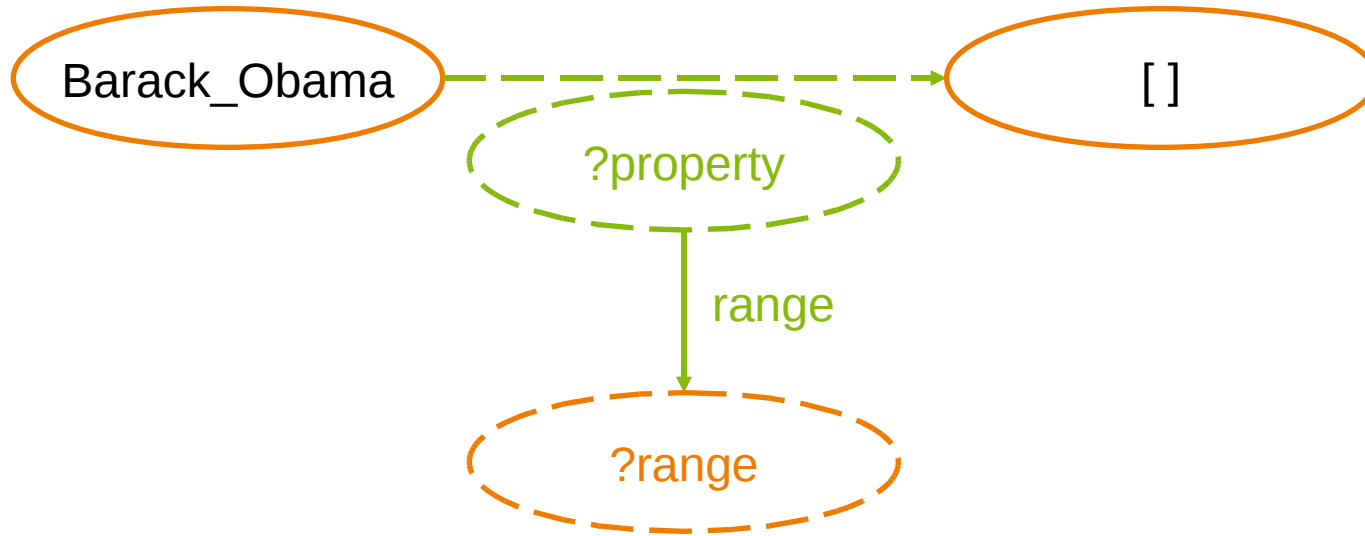


```
prefix dbpedia:<http://dbpedia.org/resource/>
prefix dbpedia-owl:<http://dbpedia.org/ontology/>
prefix p:<http://dbpedia.org/property/>
select ?person where {
  dbpedia:Barack_Obama dbpedia-owl:birthplace ?place .
  dbpedia:Barack_Obama p:profession ?profession .
  ?person dbpedia-owl:birthplace ?place .
  ?person p:profession ?profession
}
```





```
PREFIX dbpedia:<http://dbpedia.org/resource/>
PREFIX dbpedia-owl:<http://dbpedia.org/ontology/>
PREFIX p:<http://dbpedia.org/property/>
SELECT ?person ?date WHERE {
  dbpedia:Barack_Obama dbpedia-owl:birthplace ?place .
  dbpedia:Barack_Obama p:profession ?profession .
  dbpedia:Barack_Obama dbpedia-owl:birthdate ?barackdate .
  ?person dbpedia-owl:birthplace ?place .
  ?person p:profession ?profession .
  ?person dbpedia-owl:birthdate ?date .
  FILTER ( ?date < ?barackdate )
}
```



```
PREFIX dbpedia:<http://dbpedia.org/resource/>  
SELECT DISTINCT ?property ?range WHERE {  
  dbpedia:Barack_Obama ?property [] .  
  OPTIONAL { ?property rdfs:range ?range }  
}
```



```
PREFIX dbpedia:<http://dbpedia.org/resource/>  
CONSTRUCT { [] ?property ?object } WHERE {  
  dbpedia:Barack_Obama ?property ?object .  
}
```



# *Démonstration Sudoku + OWL*



[www.parisjug.org](http://www.parisjug.org)

Copyright © 2008 ParisJug. Licence CC – Creative Commons 2.0 France – Paternité – Pas d'Utilisation Commerciale – Partage des Conditions Initiales à l'Identique





Tous différents

<b>6</b>	<b>8</b>		<b>3</b>	<b>5</b>				
				<b>1</b>			<b>2</b>	
		<b>7</b>		<b>6</b>		<b>9</b>		
<b>1</b>								
<b>3</b>		<b>5</b>		<b>9</b>		<b>1</b>		<b>2</b>
								<b>9</b>
		<b>3</b>		<b>2</b>		<b>8</b>		
	<b>6</b>			<b>7</b>				
				<b>4</b>	<b>5</b>		<b>3</b>	<b>1</b>



Une case du Sudoku devient une classe

```
<owl:Class rdf:about="#C1_11"/>
```

habitée par un individu

```
<sudoku:C1_11 rdf:about="#V1_11"/>
```

Restreint à un ensemble de valeurs

```
<rdf:Description
rdf:about="http://www.w3.org/2002/07/owl#Thing">
  <rdfs:subClassOf><rdf:Description>
    <owl:oneOf rdf:parseType="Collection">
      <rdf:Description rdf:about="#1"/>
      :
      <rdf:Description rdf:about="#9"/>
    </owl:oneOf>
  </rdf:Description></rdfs:subClassOf>
</rdf:Description>
```



OWL ne sait pas tout seul que les URIs sont différentes, on lui dit :

```
<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <rdf:Description rdf:about="#1"/>
    <rdf:Description rdf:about="#2"/>
    <rdf:Description rdf:about="#3"/>
    <rdf:Description rdf:about="#4"/>
    <rdf:Description rdf:about="#5"/>
    <rdf:Description rdf:about="#6"/>
    <rdf:Description rdf:about="#7"/>
    <rdf:Description rdf:about="#8"/>
    <rdf:Description rdf:about="#9"/>
  </owl:distinctMembers>
</owl:AllDifferent>
```



On ajoute les contraintes liées à la nature du Sudoku :

```
<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Collection">
    <rdf:Description rdf:about="#V1_11"/>
    <rdf:Description rdf:about="#V1_12"/>
    <rdf:Description rdf:about="#V1_13"/>
    <rdf:Description rdf:about="#V1_21"/>
    <rdf:Description rdf:about="#V1_22"/>
    <rdf:Description rdf:about="#V1_23"/>
    <rdf:Description rdf:about="#V1_31"/>
    <rdf:Description rdf:about="#V1_32"/>
    <rdf:Description rdf:about="#V1_33"/>
  </owl:distinctMembers>
</owl:AllDifferent>
```



On ajoute enfin les contraintes de l'utilisateur :

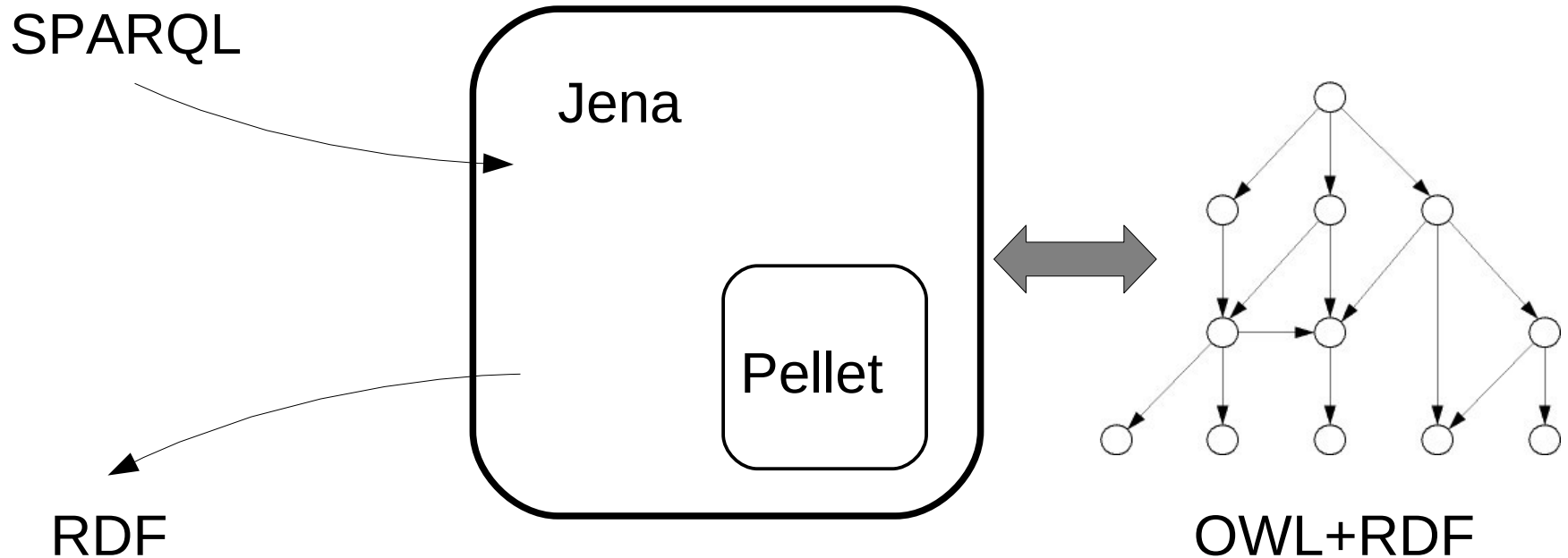
```
<owl:Thing rdf:about="#6">  
  <owl:sameAs rdf:resource="#V1_11"/>  
</owl:Thing>
```

Ici, la première case en haut à gauche est fixée à la valeur 6

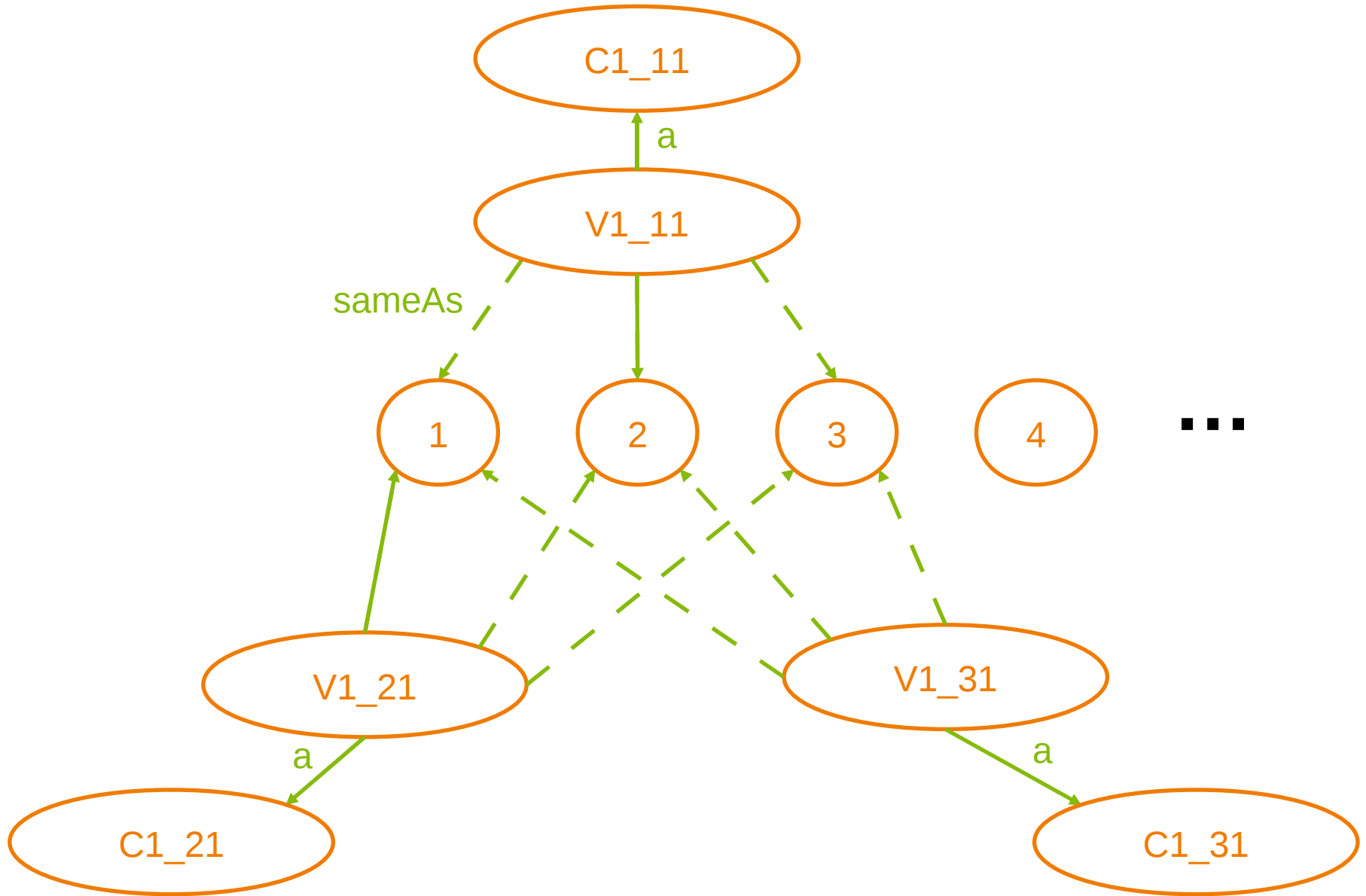


On récupère les résultats avec une simple requête :

```
SELECT ?x1_11 WHERE {  
  ?x1_11 a <http://sudoku.owl#C1_11>  
}
```



# Que fait l'inférence ?



Moi aussi je veux jouer avec !



<http://code.google.com/p/ontosudoku/>



Paris  
JUG

# Questions / Réponses



[www.parisjug.org](http://www.parisjug.org)

Copyright © 2008 ParisJug. Licence CC – Creative Commons 2.0 France – Paternité – Pas d'Utilisation Commerciale – Partage des Conditions Initiales à l'Identique



# Sponsors



# *Merci de votre attention!*



Paternité partagée avec  
Gautier Poupeau  
<http://www.lespetitescases.net/>



[www.parisjug.org](http://www.parisjug.org)

