Le Web Sémantique

Damien Leprovost

Laboratoire LIMICS
Inserm – UPMC – Paris 13
http://www.damien-leprovost.fr


Objectifs du cours

- **Comprendre ce qu'est le Web Sémantique**
- **Comment publier des données sémantiques**
- **Comment interroger des données sémantiques**
- **. . . et partir explorer le Web Sémantique !**

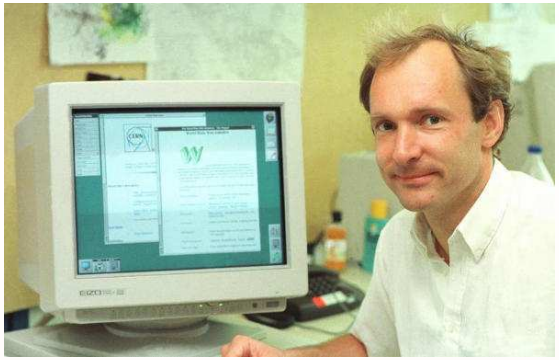
Plan du cours

Plan du cours

1. Un Web de Données
   - Une brève histoire du World Wide Web
   - Notion de ressource
   - Les Standards du Web
   - Linked Open Data

Une brève histoire du World Wide Web

- Le Web
  - Un réseau de document liés
  - Proposé par Tim-Berners-Lee, 1989
  - Documents liés par hypertexte (Ted Nelson, 1965)
  - Identifiés par leur emplacement : le schéma URL



`http://www.w3.org/Consortium/mission.html`

Une brève histoire du World Wide Web

- Le Web 2.0
  - Des interactions entre utilisateurs
  - Un Web social et participatif
  - New technologies
    - AJAX, Applications Internet Riches, Wikis, ...
  - Terme popularisé Tim O'Reilly, 2004
  - Volume exponentiel de contenu publié !

Une brève histoire du World Wide Web

- Naissance du Web Sémantique
  - Un Web de Données
  - « *A common framework that allows data to be shared and reused* » (W3C, 2011)
  - Pouvant être traité par des machines
  - Sémantique formelle, raisonnement, preuves
  - Notion de ressource
  - Extension du Web actuel
  - Basé sur des normes (*standards*) du W3C : recommandations

Plan du cours

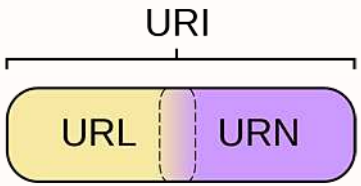
1. Un Web de Données
   - Une brève histoire du World Wide Web
   - Notion de ressource
   - Les Standards du Web
   - Linked Open Data

Notion de ressource

- Sur le « Web 1.0 », l'élément principal était le document
- Une **URL** identifie où un document est sur le Web
  - Uniform Resource Locator
- De nos jours, le Web référence plus que ses propres documents
  - Exemple : objets du monde réel, concepts abstraits, ...

Notion de ressource : URI/IRI

- **URI** : Uniform Resource Identifier
- Identifier sur le Web, toute ressource existante
- Deux spécialisations connues : URL et URN
  - Identifie une ressource soit par un emplacement, soit par un nom, soit par les deux



- **IRI** : *Internationalized Resource Identifier*
  - Utilisant Unicode pour autoriser tout caractère

https://www.مثال.tn/網址

Un Web de Données

Utilisation des URI/IRI sur le Web Sémantique :

- Utilisation d'URI HTTP pour suivre une ressource
  ex : `http://dbpedia.org/resource/Eiffel_Tower`
- Quand une URI est suivie, des données à propos de la ressources sont renvoyées
  format dépendant de la requête
- Ces données devraient contenir des liens vers d'autres données
  (structure Web hypertexte)

Plan du cours

1. Un Web de Données
   - Une brève histoire du World Wide Web
   - Notion de ressource
   - Les Standards du Web
   - Linked Open Data

Le World Wide Web Consortium

- 1994 : création du World Wide Web Consortium

- Les membres du consortium sont des entreprises, des organismes à but non-lucratif, des universités, des organismes gouvernementaux et des particuliers.
- Objet : Harmoniser le Web
- Par le développement et la promotion de normes, appelées Recommendations

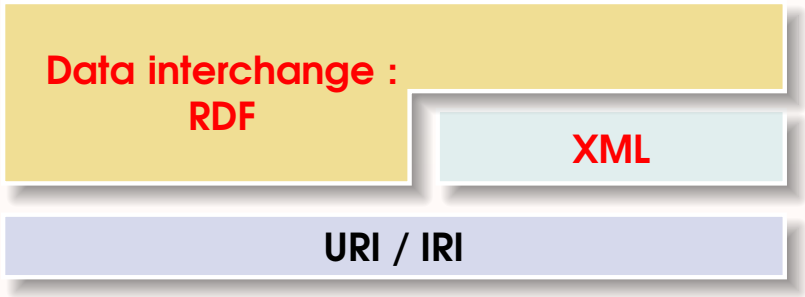
The Semantic Web Stack [1]

- Identification **URI / IRI**

1. Tim Berners-Lee, 2008.

The Semantic Web Stack [1]

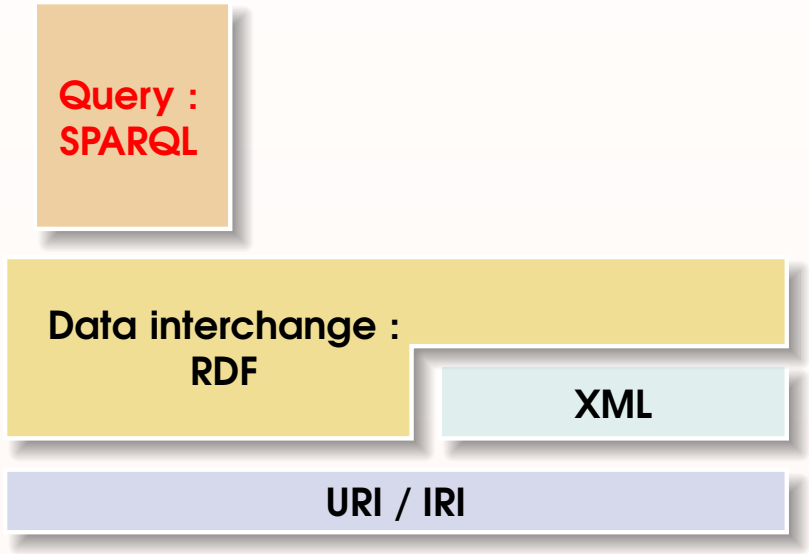
- Représentation
- Identification



1. Tim Berners-Lee, 2008.

The Semantic Web Stack [1]

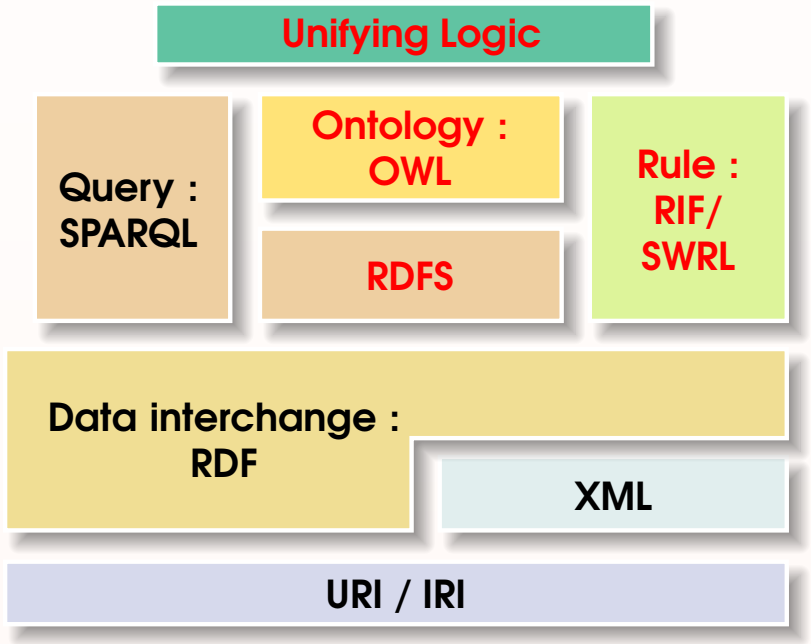
- Requêtes
- Représentation
- Identification



1. Tim Berners-Lee, 2008.

The Semantic Web Stack[1]

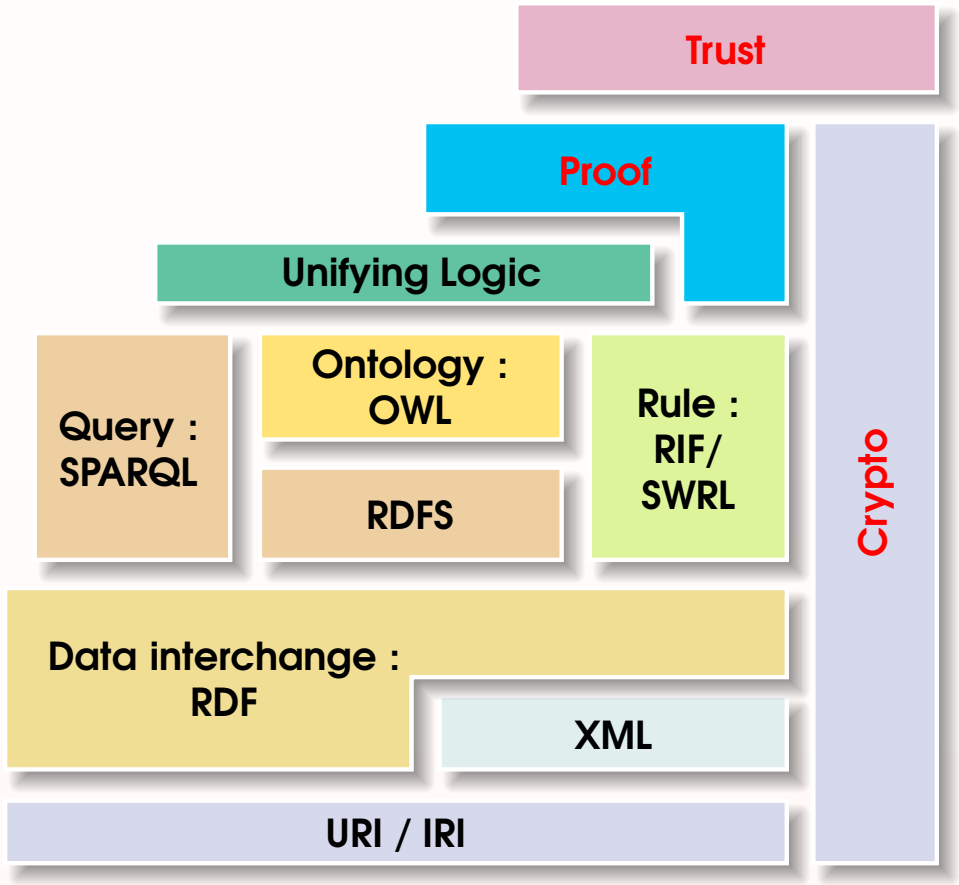
- Raisonnement
- Requêtes
- Représentation
- Identification



1. Tim Berners-Lee, 2008.

The Semantic Web Stack [1]

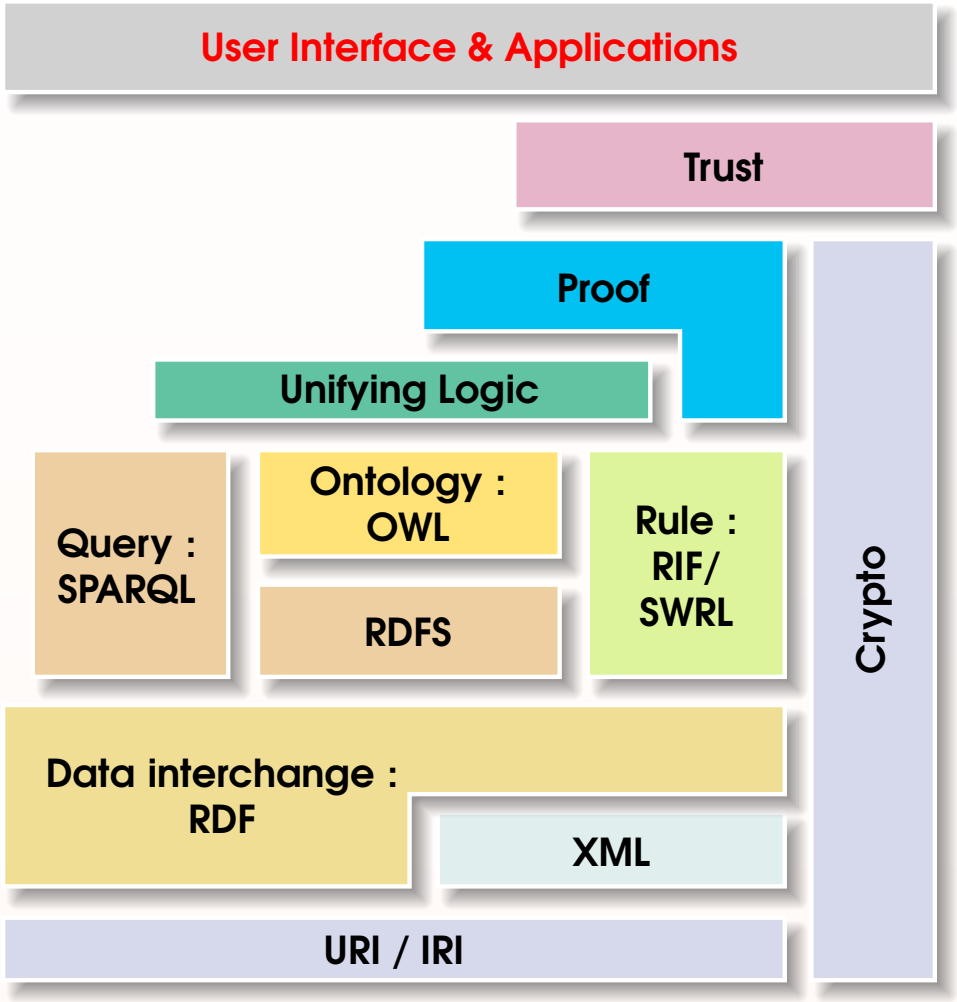
- Confiance
- Raisonnement
- Requêtes
- Représentation
- Identification



1. Tim Berners-Lee, 2008.

The Semantic Web Stack[1]

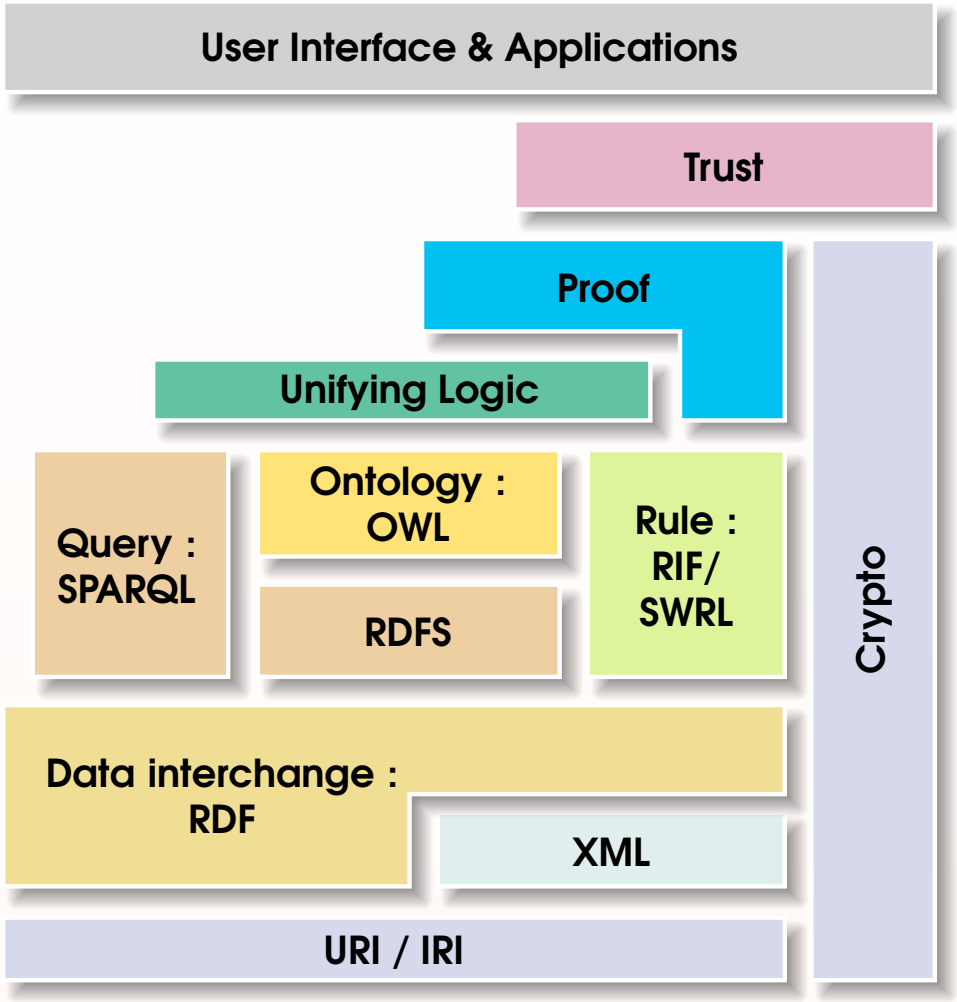
- Interaction
- Confiance
- Raisonnement
- Requêtes
- Représentation
- Identification



1. Tim Berners-Lee, 2008.

The Semantic Web Stack [1]

- Interaction
- Confiance
- Raisonnement
- Requêtes
- Représentation
- Identification



1. Tim Berners-Lee, 2008.

Plan du cours

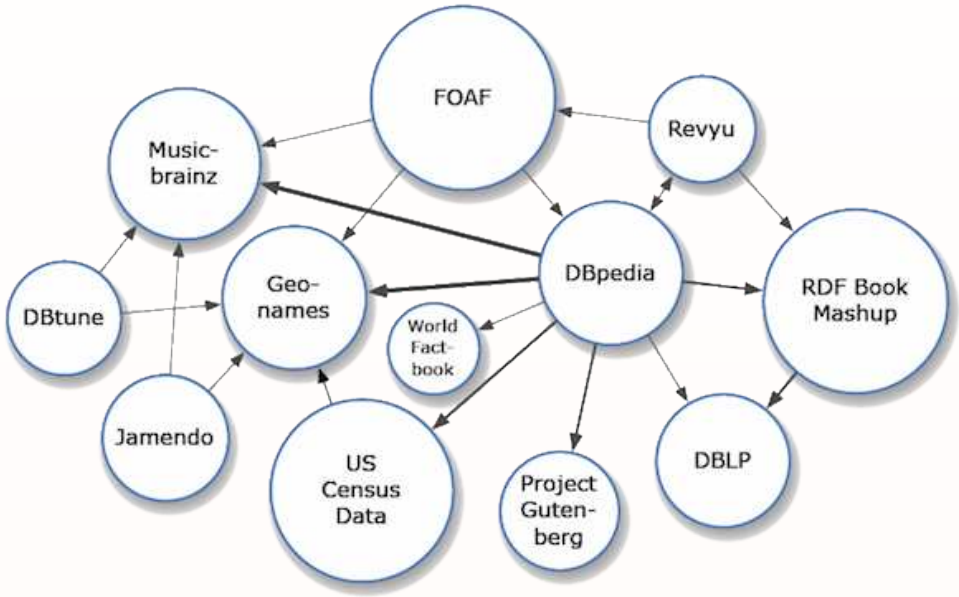
1. Un Web de Données
   - Une brève histoire du World Wide Web
   - Notion de ressource
   - Les Standards du Web
   - Linked Open Data

Linked Open Data

- Des jeux de données ouvertes sont publiées sur le Web
- *Lier les Données Ouvertes* tend à regrouper toutes ces sources de données dans un unique graphe RDF
- Rendu possible par la notion de Linked Open Data (données ouvertes et liées)

Linked Open Data [2]

Mai 2007 :



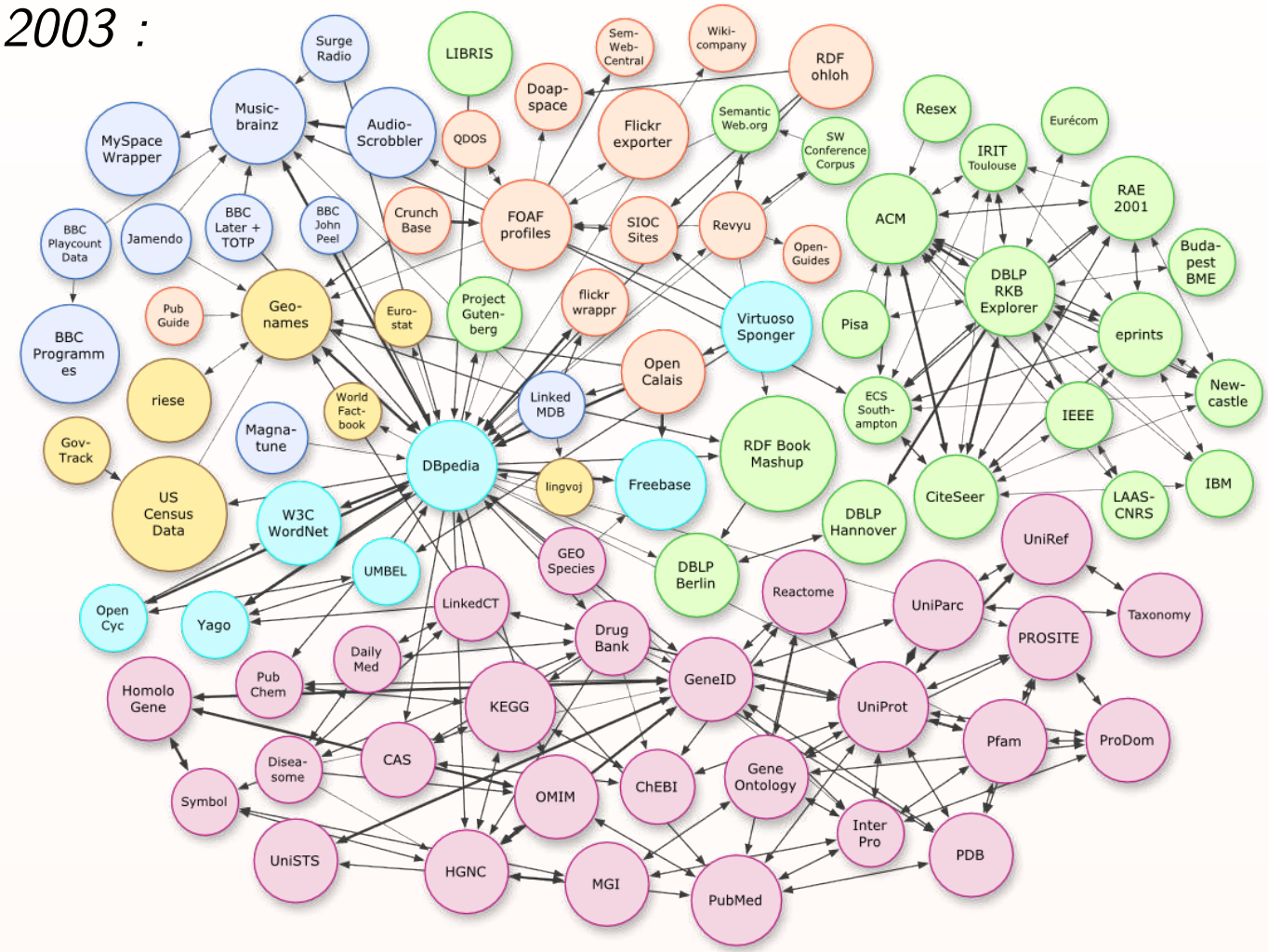
Tout jeu de données dans ce graphe possède :

- Plus de 1000 triplets
- Plus de 50 liens vers d'autres jeux du graphe
- Toutes ses données accessibles

2. Linking Open Data cloud diagram : `http://lod-cloud.net`

Linked Open Data [2]

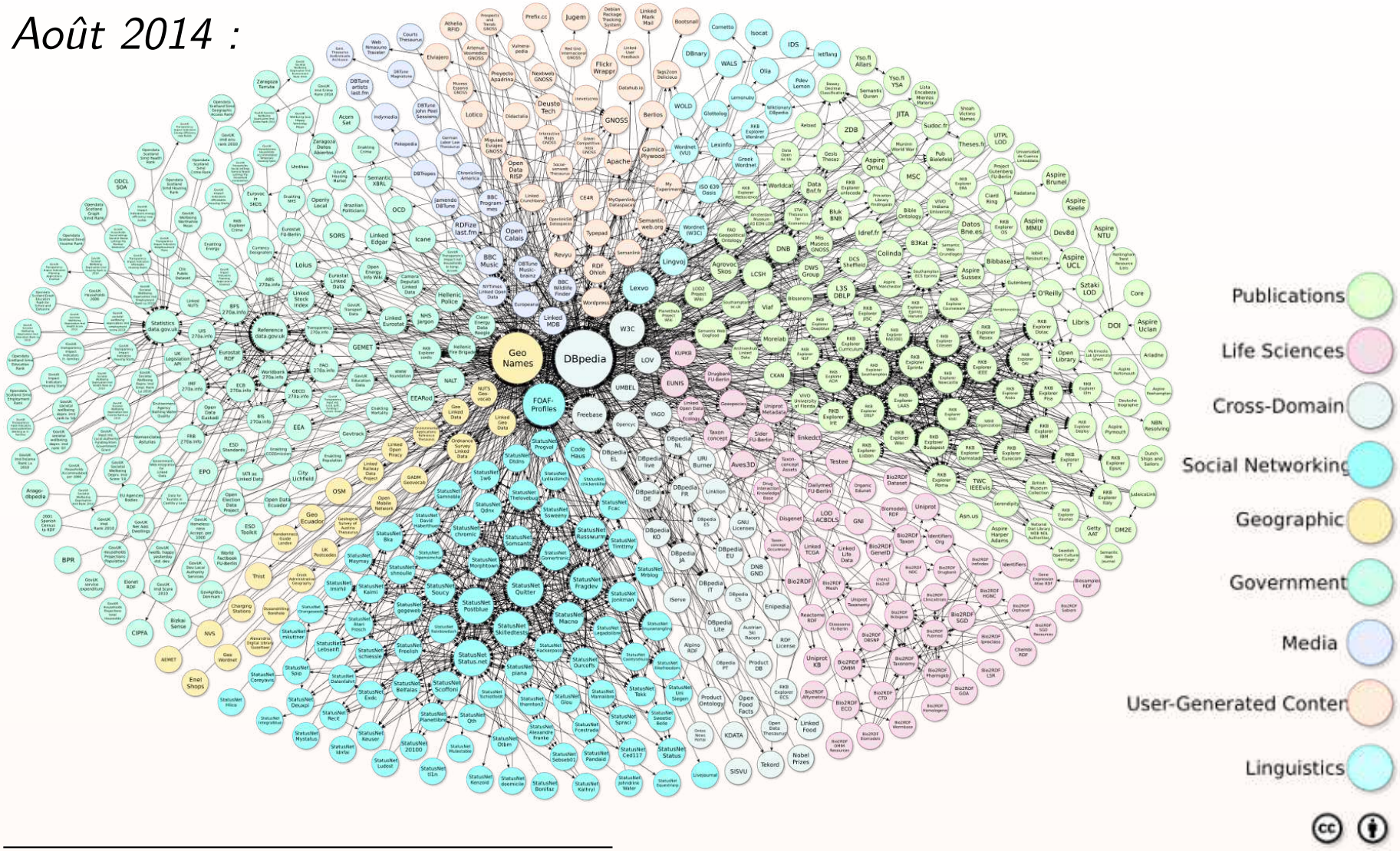
Mars 2003 :



2. Linking Open Data cloud diagram : `http://lod-cloud.net`

Linked Open Data [2]

Août 2014 :

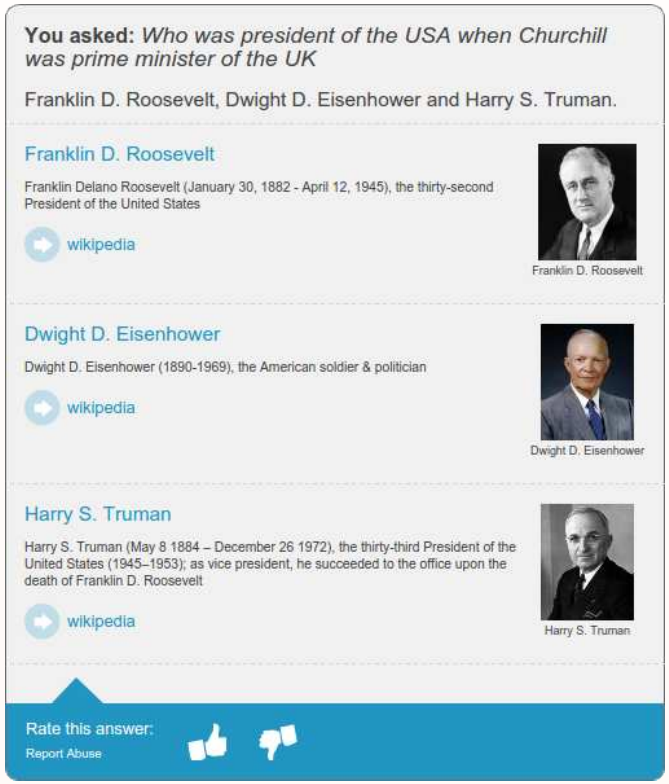


2. Linking Open Data cloud diagram : `http://lod-cloud.net`

Exemples

- Quelques jeux de données populaires
  - DBpedia : annotation RDF de pages Wikipedia (4.58M things)
  - Geonames : informations géographiques (8M lieux)

- Quelques applications
  - Evi : Moteur de réponses (*answer engine*) sémantique
  - OpenCalais : Annotation sémantique de contenu
  - Callimachus : Création d'Application Internet Riches en RDF



Plan du cours

Présentation

Resource **D**escription **F**ramework

- Représenter sur le web toute connaissance
- Décrire des informations à propos de ressources
- Basé sur des URI / IRI
- Exprimés par des assertions ou déclarations (*statements*)

Plan du cours

Représentation de données

- Les descriptions sont représentés comme des triplets

(sujet, prédicat, objet)

Représentation de données

- Les descriptions sont représentés comme des triplets

(sujet, prédicat, objet)

- Un jeu de données RDF peut être représenté comme un graphe

Représentation de données

- Les descriptions sont représentés comme des triplets

  (sujet, prédicat, objet)

- Un jeu de données RDF peut être représenté comme un graphe

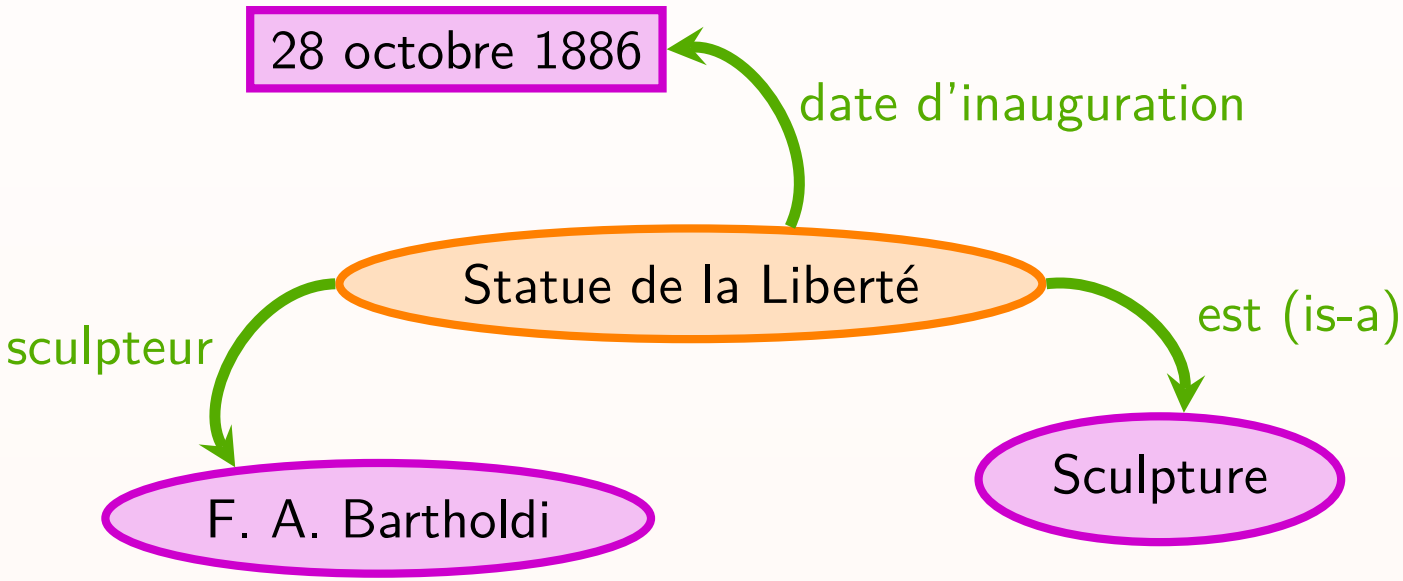
- Le sujet est une **ressource**.
- Le prédicat est une propriété de la ressource.
- L'objet est la valeur de la propriété pour cette ressource ; il peut être une autre ressource, ou un littéral.

Représentation de données : Exemple

- La Statue de la Liberté est une sculpture
- Bartholdi est son sculpteur
- Inauguration le 28 octobre 1886

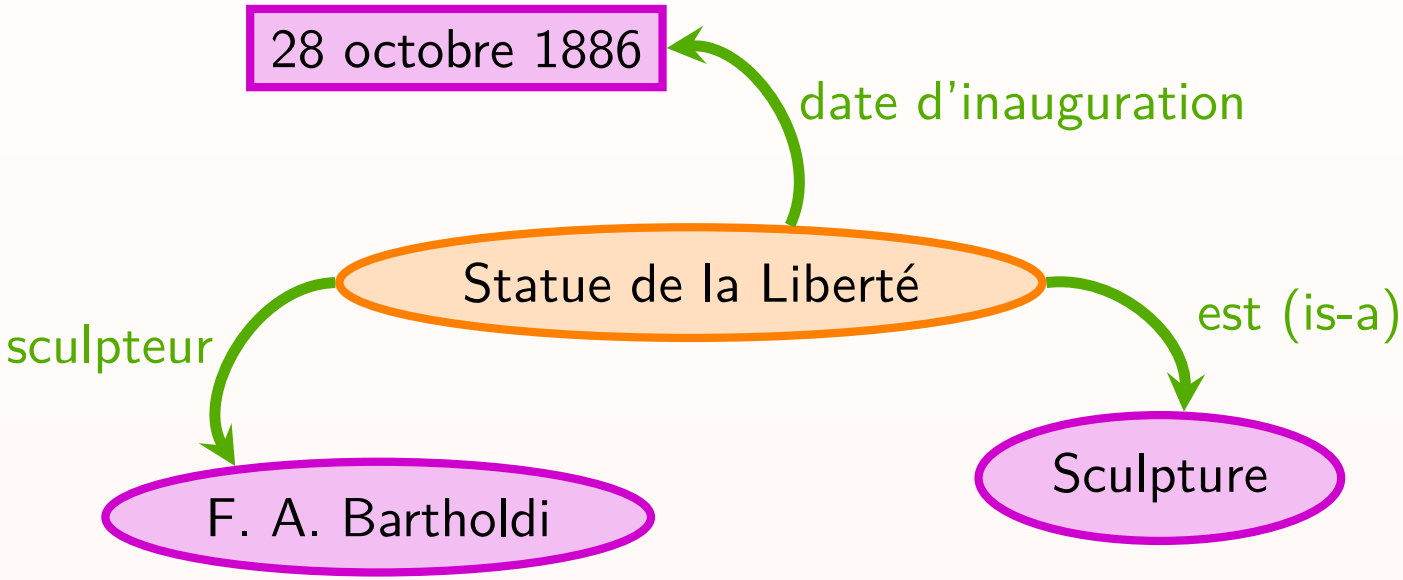
Représentation de données : Exemple

- La Statue de la Liberté est une sculpture
- Bartholdi est son sculpteur
- Inauguration le 28 octobre 1886



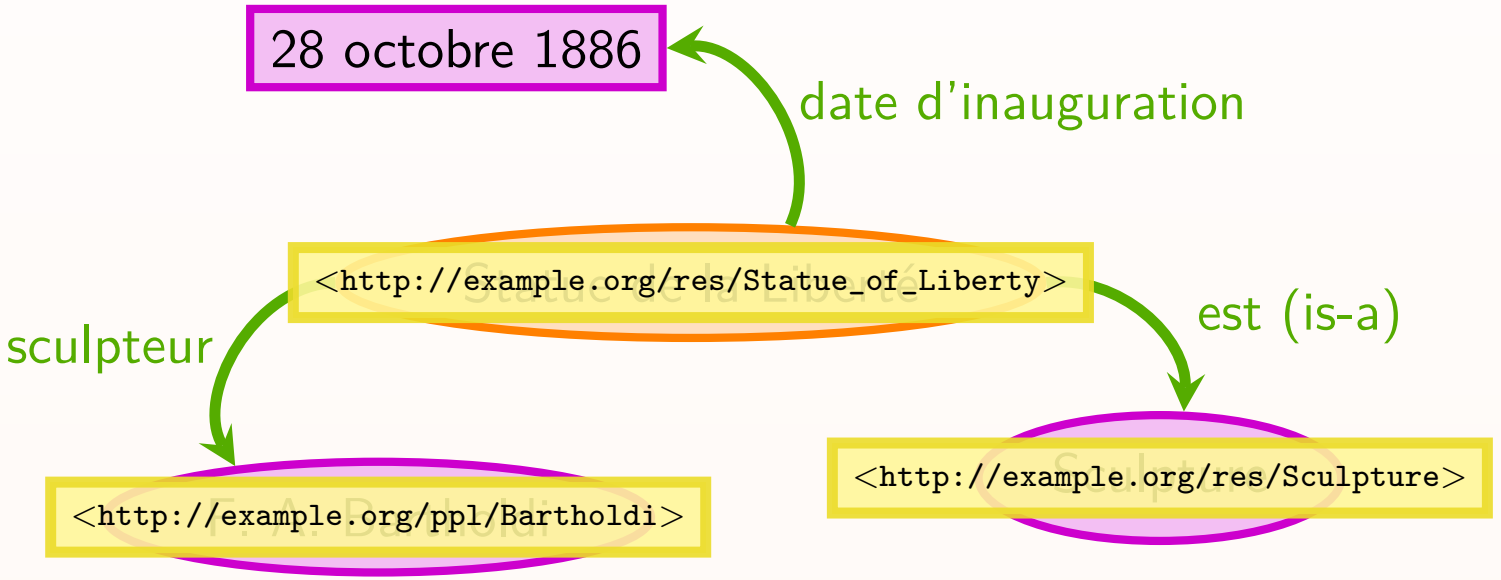
Représentation de données : Exemple

- Chaque nœud ou arc du graphe est labelisé par une URI/IRI ou un littéral
- Les URI/IRI identifient les ressources
- Les littéraux peuvent être des entiers, des chaînes de caractères ou des dates



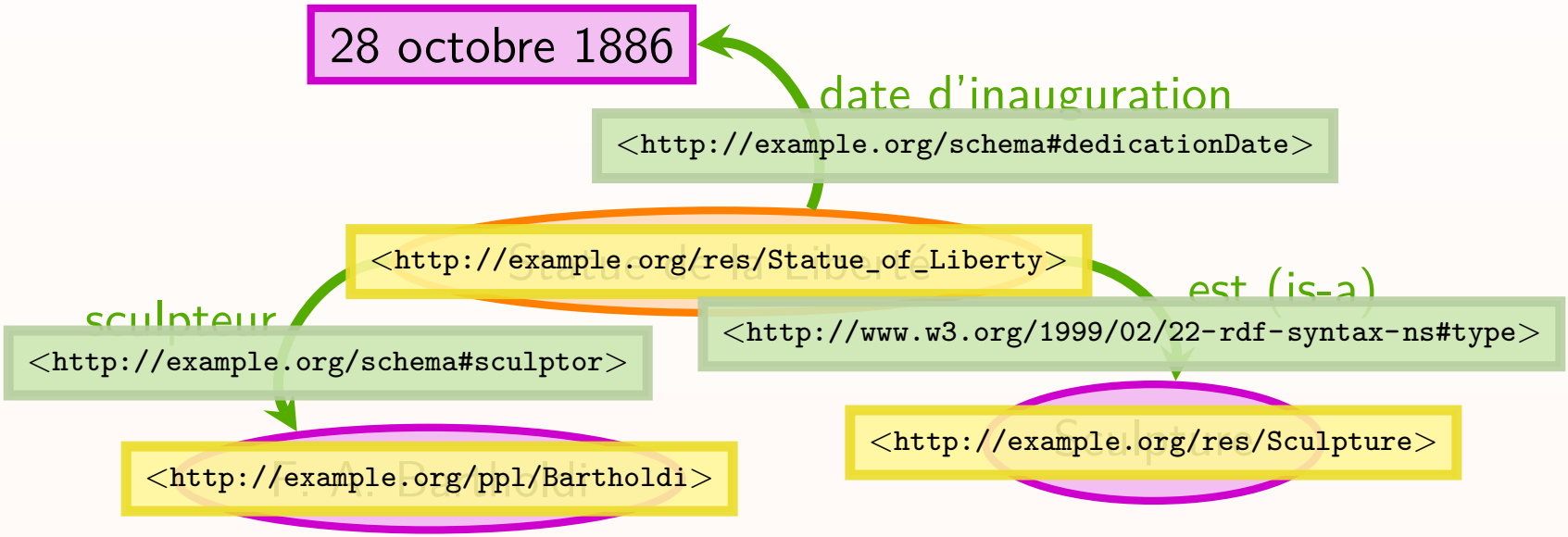
Représentation de données : Exemple

- Chaque nœud ou arc du graphe est labelisé par une URI/IRI ou un littéral
- Les URI/IRI identifient les ressources
- Les littéraux peuvent être des entiers, des chaînes de caractères ou des dates



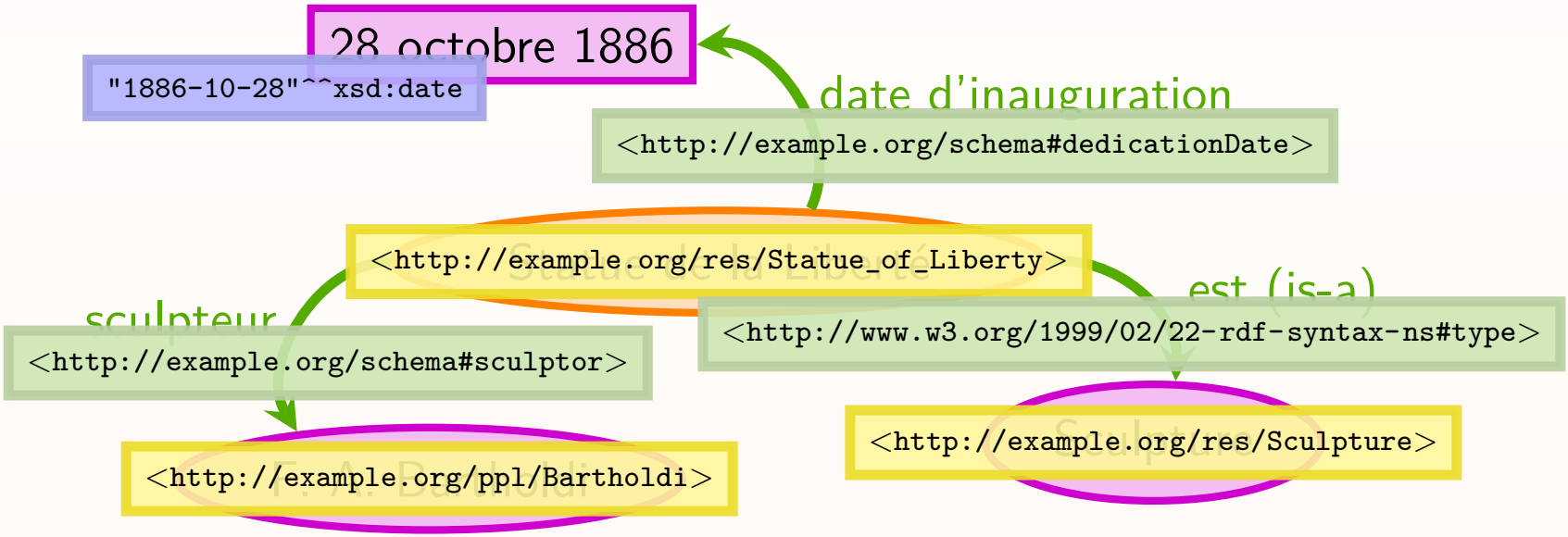
Représentation de données : Exemple

- Chaque nœud ou arc du graphe est labelisé par une URI/IRI ou un littéral
- Les URI/IRI identifient les ressources
- Les littéraux peuvent être des entiers, des chaînes de caractères ou des dates



Représentation de données : Exemple

- Chaque nœud ou arc du graphe est labelisé par une URI/IRI ou un littéral
- Les URI/IRI identifient les ressources
- Les littéraux peuvent être des entiers, des chaînes de caractères ou des dates



Plan du cours

Vocabulaires

- Les URI/IRI identifient tout
- Laquelle dois-je utilisée pour une chose (supposée) existante ?
- Utiliser les vocabulaires RDF : LOV (*Linked Open Vocabularies*)
- Définisent Définition des entités de classe et des prédicats entre entités
- Souvent réunis autour d'un thème
- Facilité d'utilisation, non une obligation !

Quelques vocabulaires

- RDF Schema : éléments de base

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

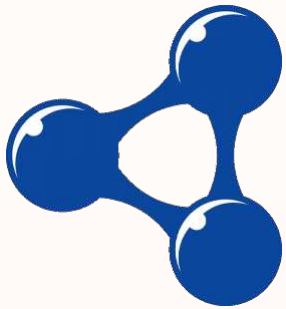
  `rdfs:<http://www.w3.org/2000/01/rdf-schema#>`
  - *Ex. of classes* : Class, Property, Datatype, List
  - *Ex. of propriétés* : type, label, value
- Friend of a Friend : relations sociales

  `foaf:<http://xmlns.com/foaf/0.1/>`
  - *Ex. of classes* : Person, Organization, Project
  - *Ex. of propriétés* : firstName, knows, homepage
- Dublin Core : spécifications documentaires

  `dc:<http://purl.org/dc/elements/1.1>`
  - *Ex. of classes* : BibliographicResource, PhysicalResource, Policy
  - *Ex. of propriétés* : creator, references, title
- SIOC : Semantically-Interlinked Online Communities
- SKOS : Simple Knowledge Organization System
- OWL : Web Ontology Language

Interliens

- Ces vocabulaires fournisent des prorpiétés pour interlier les ressources et les jeux de données
  - `rdfs:seeAlso` pour des informations additionnelles
  - `owl:sameAs` quand deux URI se réfèrent à la même chose
  - `rdfs:isDefinedBy` quand une ressource est définie dans une ressource sujet
- L'héritage permet un nombre illimité de classes et de propriétés
- `seeAlso` : Linked Open Vocabularies (LOV) [3]

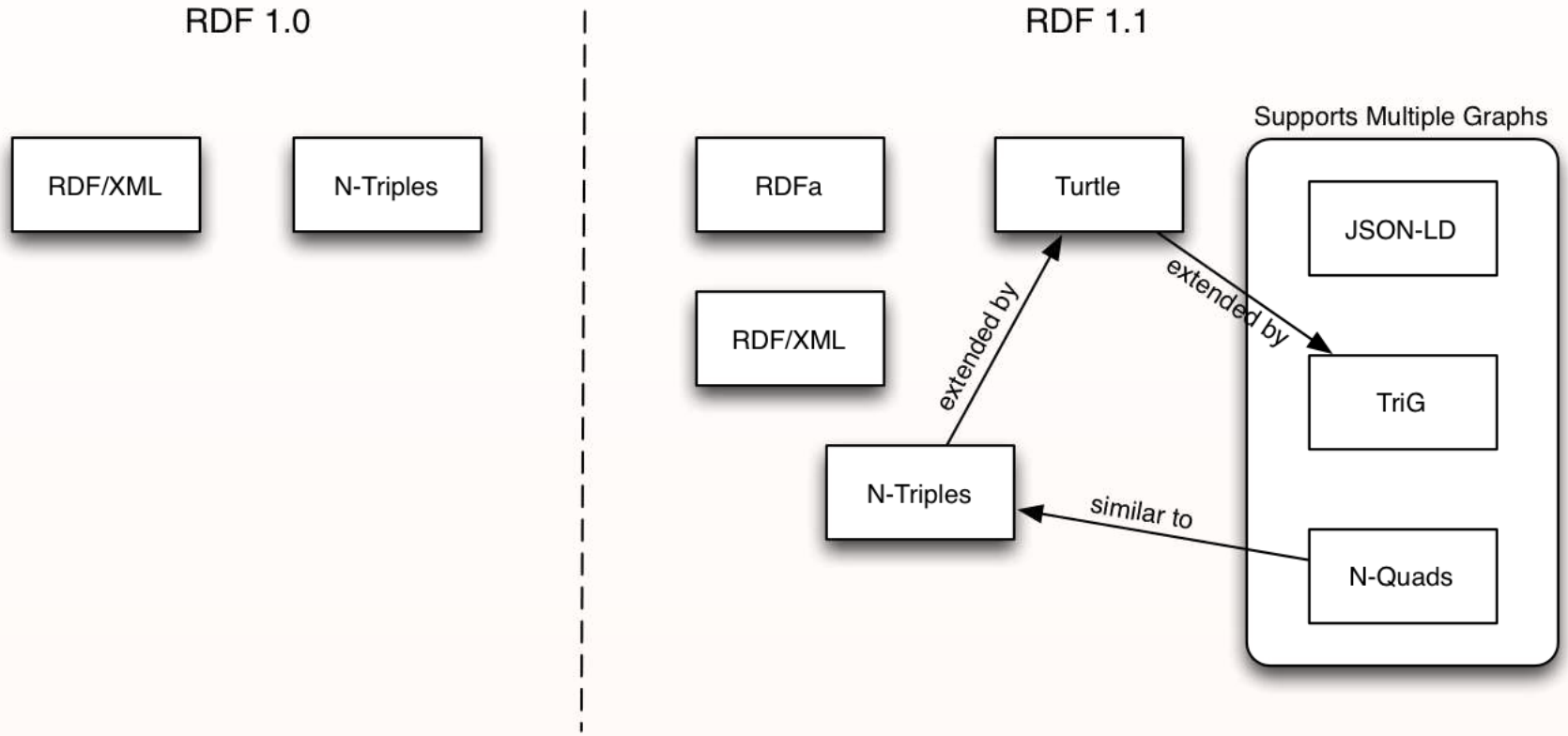


3. `http://lov.okfn.org`

Plan du cours

Sérialisation

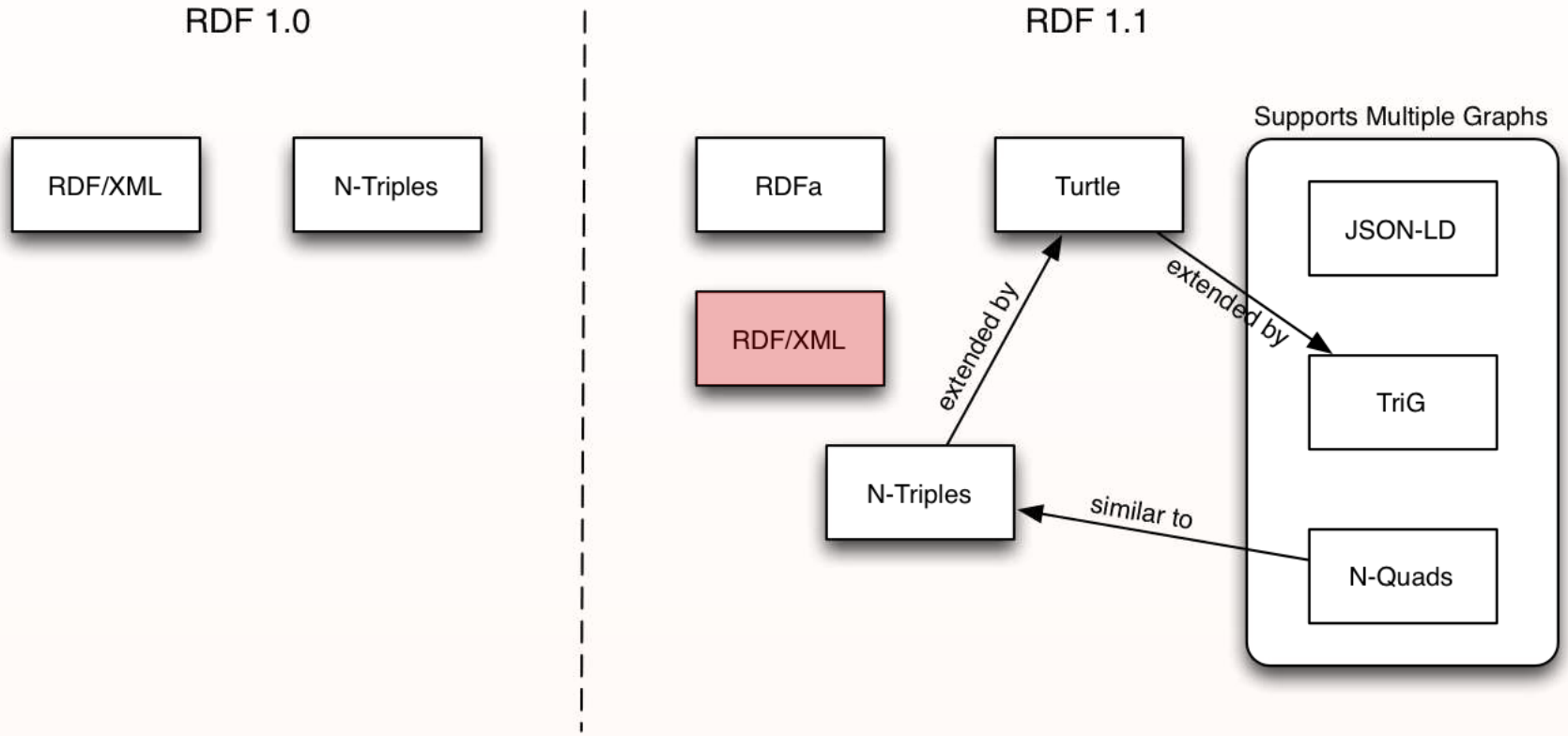
- En quel format écrire du RDF ?
- Sept formats de sérialisations recommandés par le W3C [4].



4. Image depuis `http://www.w3.org/TR/rdf11-new/`

Sérialisation

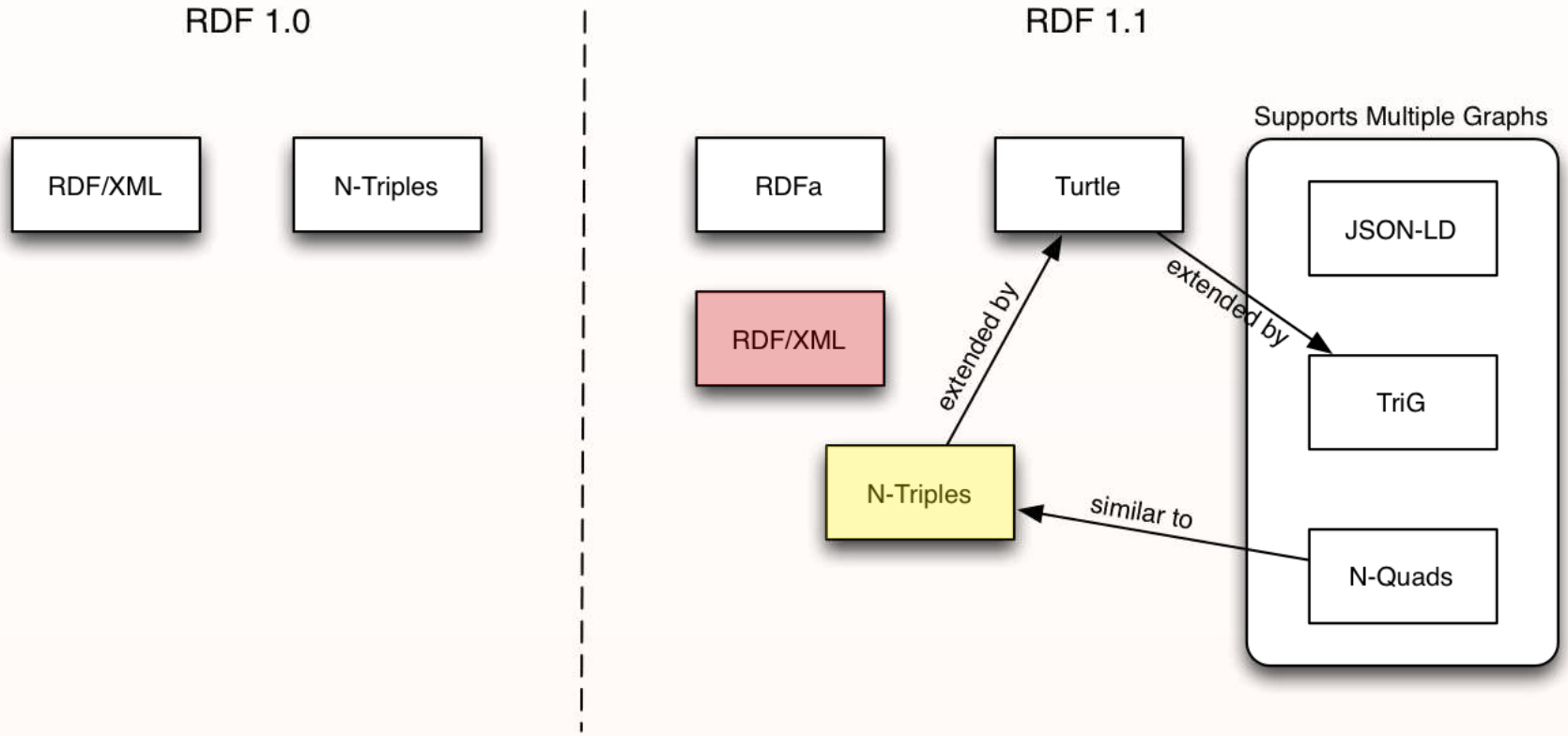
- En quel format écrire du RDF ?
- Sept formats de sérialisations recommandés par le W3C [4].



4. Image depuis `http://www.w3.org/TR/rdf11-new/`

Sérialisation

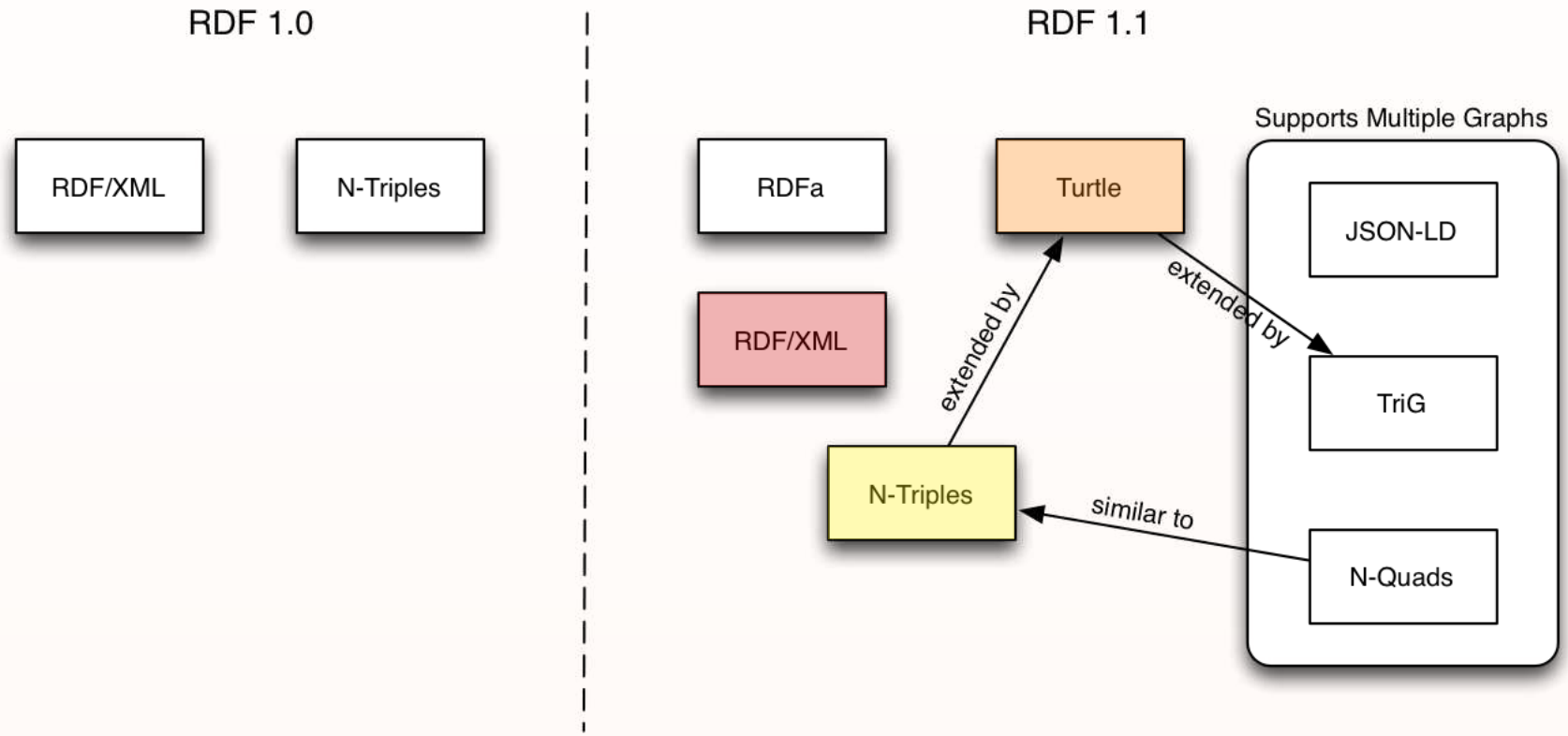
- En quel format écrire du RDF ?
- Sept formats de sérialisations recommandés par le W3C [4].



4. Image depuis `http://www.w3.org/TR/rdf11-new/`

Sérialisation

- En quel format écrire du RDF ?
- Sept formats de sérialisations recommandés par le W3C [4].



4. Image depuis `http://www.w3.org/TR/rdf11-new/`

RDF/XML

- RDF/XML : Première sérialisation RDF existante
- RDF décrit dans un format XML

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:exs="http://example.org/schema#">
<rdf:Description
  rdf:about="http://example.org/res/Statue_of_Liberty">
  <rdf:type rdf:resource="http://example.org/res/Sculpture"/>
  <exs:sculptor rdf:resource="http://example.org/ppl/Bartholdi"/>
  <exs:dedicationDate rdf:datatype="xsd:date">
    1886-10-28
  </exs:dedicationDate>
</rdf:Description>
```

N-Triples

- RDF décrit sous forme de triplets
- `<IRI sujet> <IRI prédicat> <IRI objet ou littéral>`

```
<http://example.org/res/Statue_of_Liberty>
<http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
<http://example.org/res/Sculpture> .

<http://example.org/res/Statue_of_Liberty>
<http://example.org/schema#sculptor>
<http://example.org/ppl/Bartholdi> .

<http://example.org/res/Statue_of_Liberty>
<http://example.org/schema#dedicationDate>
"1886-10-28"^^<http://www.w3.org/2001/XMLSchema#date> .
```

Turtle

- Étend N-Triple
- Ajoute des raccourcis d'écriture pour rendre le code plus lisible

```
BASE <http://example.org/res/>
  PREFIX xsd:<http://www.w3.org/2001/XMLSchema#">
  PREFIX exs:<http://example.org/schema#">
  PREFIX ppl:<http://example.org/ppl/">

Statue_of_Liberty
  a Sculpture ;
  exs:sculptor ppl:Bartholdi ;
  exs:dedicationDate "1886-10-28"^^xsd:date .
```

- **a** représente `rdf:type`
- Le point-virgule lie plusieurs assertions d'un même sujet, la virgule lie plusieurs valeurs d'un même sujet–prédicat.

Plan du cours

Valeurs littérales

- Les littéraux utilisent des types de données du *XML Schema*
- Les plus communs sont les chaînes de caractères, les entiers et les dates

```
BASE <http://example.org/res/>
  PREFIX xsd:<http://www.w3.org/2001/XMLSchema#">
  PREFIX exs:<http://example.org/schema#">

<Statue_of_Liberty>
  exs:dedicationDate "1886-10-28"^^xsd:date ;
  exs:visitation "3.2e6"^^xsd:double ;
  exs:visitationYear "2009"^^xsd:integer ;
  exs:name "Statue of Liberty"^^xsd:string ;
  exs:publicOpened "true"^^xsd:boolean .
```

- En Turtle : nombres, chaînes et booléens peuvent être écrits directement (omission du datatype)

Valeurs littérales

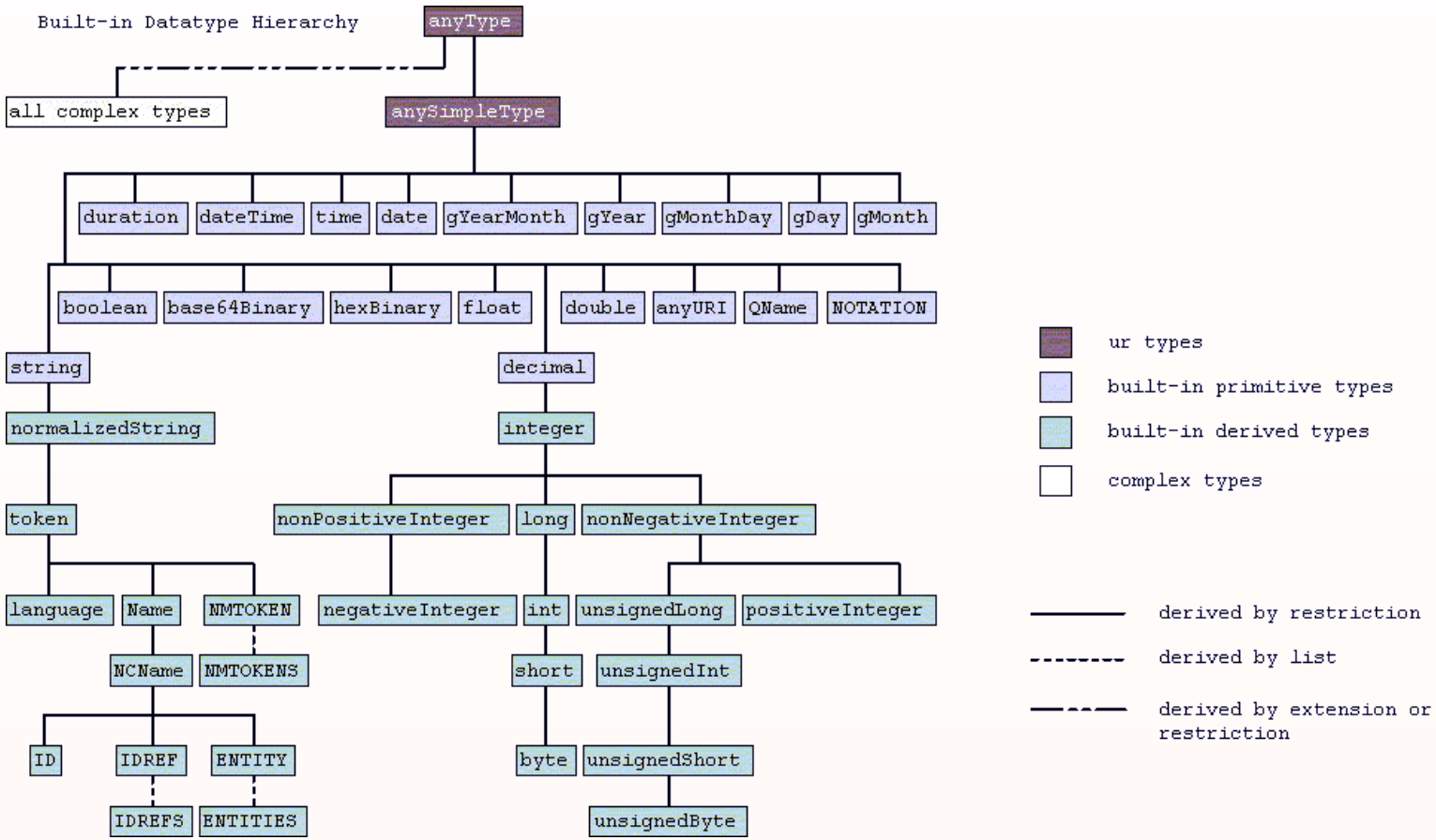
- Les chaînes peuvent être localisées
- `xml:lang` proriété en RDF/XML

```
BASE <http://example.org/res/>
  PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#">

<Statue_of_Liberty>
  rdfs:label "Statue of Liberty"@en ;
  rdfs:label "Statue de la Liberté"@fr ;
  rdfs:label "Freiheitsstatue"@de .
```

Valeurs littérales

- De nombreux types disponibles [5]



5. Image depuis `http://www.w3.org/TR/xmlschema-2/`

Nœud anonyme (Blank node)

- Utilisé pour décrire des informations à propos de ressources anonymes
- Uniquement comme sujet ou objet d'un triplet
- Exemple : *Tobias connaît quelqu'un nommé David.*

```
<Tobias> foaf:knows [ foaf:name "David" ] .
```

- Les nœuds anonymes peuvent être nommés pour être réutilisés

```
<Tobias> foaf:knows _:a1 .
_:a1 foaf:name "David";
  foaf:age 40 .
```

Conteneurs

Décrire les goupes d'éléments

- `rdf:Bag` , un ensemble de valeurs sans ordre spécifique

```
<Statue_of_Liberty> exs:contains [ a rdf:Bag ;
  rdf:li <Torch> ;
  rdf:li <Statue> ;
  rdf:li <Pedestal> . ] .
```

- `rdf:Seq` , une liste ordonnée de valeurs

```
<Statue_of_Liberty> exs:restored [ a rdf:Seq ;
  rdf:li 1938 ;
  rdf:li 1986 ;
  rdf:li 2012 . ] .
```

- `rdf:Alt` , une liste de valeur alternatives

```
<Statue_of_Liberty> rdfs:label [ a rdf:Alt ;
  rdf:li "Statue of Liberty"@en ;
  rdf:li "Statue de la Liberté"@fr ;
  rdf:li "Freiheitsstatue"@de . ] .
```

Plan du cours

Présentation

- Acronyme récursif pour
  ***S**PARQL **P**rotocol and **R**DF **Q**uery **L**anguage*

- Langage de requête pour RDF et RDFS
- Utilise filtrage par motif (*pattern matching*) sur le graphe
- Syntaxe Turtle

Présentation

- Quatre type de requêtes
  - **SELECT** Retourne toutes les valeurs qui valident le motif de requête
  - **CONSTRUCT** Construit un graphe RDF depuis les données retournées
  - **DESCRIBE** Demande d'information à propos de quelque chose
  - **ASK** Retourne un booléen pour une question-requête
- Utilisées sur un SPARQL endpoint
- Traditionnellement, chaque appel utilise une clause WHERE pour restreindre la requête

Présentation

- Une requête contient un motif de triplet
- Ce motif peut utiliser :
  - des IRIs
  - des littéraux
  - des variables, préfixées par **?** or **$**
  - des nœuds anonymes
- Utilisant syntaxe et préfixes Turtle

Plan du cours

La requête SELECT

- Forme globale :

[PREFIX <*préfixes*>]

SELECT <*variables retournées*>

[FROM <*sources*>]

WHERE { <*motif de triplet*> }

La requête SELECT

Exemples de requête SELECT

- Retourne toutes les assertions du graphe :

La requête SELECT

Exemples de requête SELECT

- Retourne toutes les assertions du graphe :

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

La requête SELECT

Exemples de requête SELECT

- Retourne toutes les assertions du graphe :

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

- Retourne noms et prénoms de tous les sculpteurs du graphe :

La requête SELECT

Exemples de requête SELECT

- Retourne toutes les assertions du graphe :

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

- Retourne noms et prénoms de tous les sculpteurs du graphe :

```
SELECT ?firstName ?lastName WHERE { _:x
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
   <http://example.org/res/Sculptor> .
 _:x <http://xmlns.com/foaf/0.1/firstName>
   ?firstName .
 _:x <http://xmlns.com/foaf/0.1/lastName>
   ?lastName . }
```

La requête SELECT

Exemples de requête SELECT

- Retourne toutes les assertions du graphe :

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

- Retourne noms et prénoms de tous les sculpteurs du graphe :

```
SELECT ?firstName ?lastName WHERE { _:x
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
   <http://example.org/res/Sculptor> .
 _:x <http://xmlns.com/foaf/0.1/firstName>
   ?firstName .
 _:x <http://xmlns.com/foaf/0.1/lastName>
   ?lastName . }
```

- Retourne toutes les propriétés ayant 2015 comme valeur :

La requête SELECT

Exemples de requête SELECT

- Retourne toutes les assertions du graphe :

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

- Retourne noms et prénoms de tous les sculpteurs du graphe :

```
SELECT ?firstName ?lastName WHERE { _:x
 <http://www.w3.org/1999/02/22-rdf-syntax-ns#type>
   <http://example.org/res/Sculptor> .
 _:x <http://xmlns.com/foaf/0.1/firstName>
   ?firstName .
 _:x <http://xmlns.com/foaf/0.1/lastName>
   ?lastName . }
```

- Retourne toutes les propriétés ayant 2015 comme valeur :

```
SELECT DISTINCT ?prop
WHERE { [] ?prop "2015" }
```

SPARQL est en Turtle !

- Rappel de Turtle :
  - `BASE` et `PREFIX` sont disponibles pour les vocabulaires
  - `a` pour `rdf:type`
  - Factorisation des assertions avec `;` et des valeurs avec `,`
- Réécriture de la seconde requête du slide précédent :

SPARQL est en Turtle !

- Rappel de Turtle :
  - `BASE` et `PREFIX` sont disponibles pour les vocabulaires
  - `a` pour `rdf:type`
  - Factorisation des assertions avec `;` et des valeurs avec `,`
- Réécriture de la seconde requête du slide précédent :

```
BASE <http://example.org/res/>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?firstName ?lastName WHERE
{ [] a Sculptor ;
    foaf:firstName ?firstName ;
    foaf:lastName ?lastName . }
```

Motif optionnels : OPTIONAL

- Une partie du motif n'est pas obligatoire :

```
BASE <http://example.org/res/>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?firstName ?lastName ?homepage WHERE {
_:x a Sculptor ;
    foaf:firstName ?firstName ;
    foaf:lastName ?lastName .
  OPTIONAL { _:x foaf:homepage ?homepage .}
}
```

- ?homepage peut être absent de certains résultats

Union de motifs : UNION

- Union de motifs, similaire à une clause **OR** :

```
BASE <http://example.org/res/>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?firstName ?lastName WHERE {
  {
    _:x a Sculptor ;
  }
  UNION
  {
    _:x a Singer .
  }
  _:x foaf:firstName ?firstName ;
    foaf:lastName ?lastName .
}
```

Exclusion de motifs : MINUS

- Retirer des motifs des résultats :

```
BASE <http://example.org/res/>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?x WHERE {
  ?x a Sculptor .
   MINUS {?x a Jazz_trumpeter }
}
```

- (à titre d'exemple, cette requête retire Jeff Nuttall des résultats.)

Motifs de chemin

- Quelques expressions rationnelles sur les chemins entre les ressources :

```
BASE <http://example.org/res/>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?relation WHERE {
  [] foaf:name "John Doe" ;
    foaf:knows+ ?relation . }
```

- Motifs existants :
  - / séquence
  - + un ou plusieurs
  - ? optionnel
  - ! négation
  - | alternative
  - * zéro or plusieurs
  - ^ inverse

Ordonner et paginer

- Ces clauses fonctionnement comme en SQL
  - `ORDER` Ordonner les résultats avec `ASC` et `DESC`
  - `GROUP BY` Grouper les résultats avec une variable
  - `LIMIT` Limiter le nombre de résultats
  - `OFFSET` Démarrer à une position donnée

Plan du cours

Filtres

- Appliquer des filtres à la clause `WHERE`
- La clause `FILTER` est écrite en XPath

```
BASE <http://example.org/res/>
PREFIX ex:<http://example.org/schema#>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?x ?ach WHERE {
  ?x a Sculptor ;
      ex:achievements ?ach .
   FILTER (xsd:integer(?ach) >= 50)
}
```

- Il est également possible d'utiliser des filtres externes

Filtres

- Tests sur les valeurs :
  - Comparaisons : $<$, $>$, $=$, $<=$, $>=$, $!=$
  - Expressions rationnelles : `regex(?x, ".*rdo")`
  - Tests de typage : `isURI(?x)`, `isBlank(?x)`, `isLiteral(?x)`, `isNumeric(?x)`
  - Tests de chaîne : `contains(?x,"foobar")`, `strstarts(?x,"foo")`, `strends(?x,"bar")`
- Les filtres peuvent être combinés
  - Opérateurs booléens : `&&`, `||`, `!`, `()`

Filtres

- Filtres conditionnels :
  - Structure `if - then - else` :

```
BASE <http://example.org/res/>
PREFIX ex:<http://example.org/schema#>
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?adults WHERE {
  ?adults foaf:age ?age ;
      ex:country ?c .
  FILTER (?c="USA",?age>=21,?age>=18)
}
```

  - Tout filtre peut être nié : `FILTER NOT EXISTS (...)`

Plan du cours

Fonctions

- De nombreuses fonctions peuvent être utilisées dans les clauses `FILTER` et `SELECT`

- Typage : `lang()`, `datatype()`, `str()`, `xsd:integer(?x)`, `strdt(v, t)`, ...
- Chaînes : `strlang(v, l)`, `concat(?x,?y)`, `ucase(?x)`, `lcase(?x)`, `substr(?x,s[,l])`, `strlen(?x)`, ...
- Numérique : `abs(?x)`, `ceil(?x)`, `floor(?x)`, `round(?x)`, `rand()`, ...
- Dates : `year(?x)`, `month(?x)`, `day(?x)`, `hours(?x)`, `timezone(?x)`, `now()`, ...
- Et beaucoup d'autres disponibles[6], appels externes également possibles

6. Voir : `http://www.w3.org/TR/sparql11-query/#expressions`

Plan du cours

Les autres requêtes

- **ASK** Retourne un booléen à une question
  - Renvoie `true` si n'importe quelle donnée valide la requête
  - Sinon renvoie `false`
- **ASK** est une clause `WHERE`

```
BASE <http://example.org/res/>
ASK {?x a Actor, Politician }
```

Les autres requêtes

- **CONSTRUCT** Produit un graphe RDF à partir des données retournées

- Le résultat est un sous-graphe du graphe source

```
PREFIX ppl:<http://example.org/ppl/>
CONSTRUCT { ppl:Bartholdi ?predicate ?object }
WHERE { ppl:Bartholdi ?predicate ?object }
```

Les autres requêtes

- **DESCRIBE** Demande d'information à propos d'une ressource
- Ce qui sera effectivement retrouné dépend de la configuration du serveur

```
DESCRIBE <http://example.org/ppl/Bartholdi>
```

Plan du cours

SPARQL UPDATE

- Bien qu'au delà du cadre de ce cours, il est important de savoir que :

SPARQL peut altérer les données

(sous réserve de droits suffisants)

- `LOAD` Charge des triplets externes
- `INSERT DATA` Ajoute des triplets dans le graphe
- `DELETE DATA` Retire des triplets du graphe
- Une mise à jour (update) peut être réalisé par la combinaison de deux requêtes
  `DELETE DATA` et `INSERT DATA`.

SPARQL UPDATE

```
BASE <http://example.org/res/>
  PREFIX exs:<http://example.org/schema#">
  PREFIX ppl:<http://example.org/ppl/">
DELETE DATA {Statue_of_Liberty exs:sculptor ppl:Bartholdi }
INSERT DATA {Statue_of_Liberty exs:sculptor ppl:Matt_Groening }
```

Conclusion

- Ce cours n'est qu'une introduction à ce qu'est le Web Sémantique
- Partage de connaissance, rendre la connaissance utilisable par des machines
- Requêtes et utilisation de cette connaissance
- ***Be open, be linked !***

Le Web Sémantique

Damien Leprovost

29 novembre 2016

permalink: https://www.damien-leprovost.fr/enseignements/semweb.2016.pdf