

Ontologies et

Web Sémantique

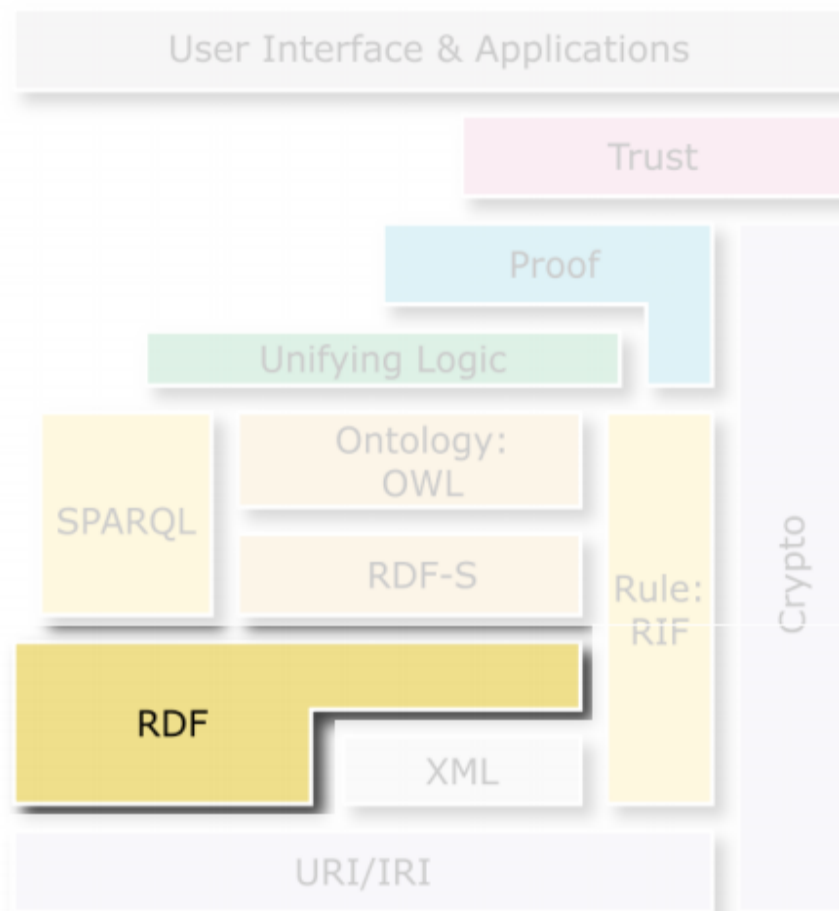
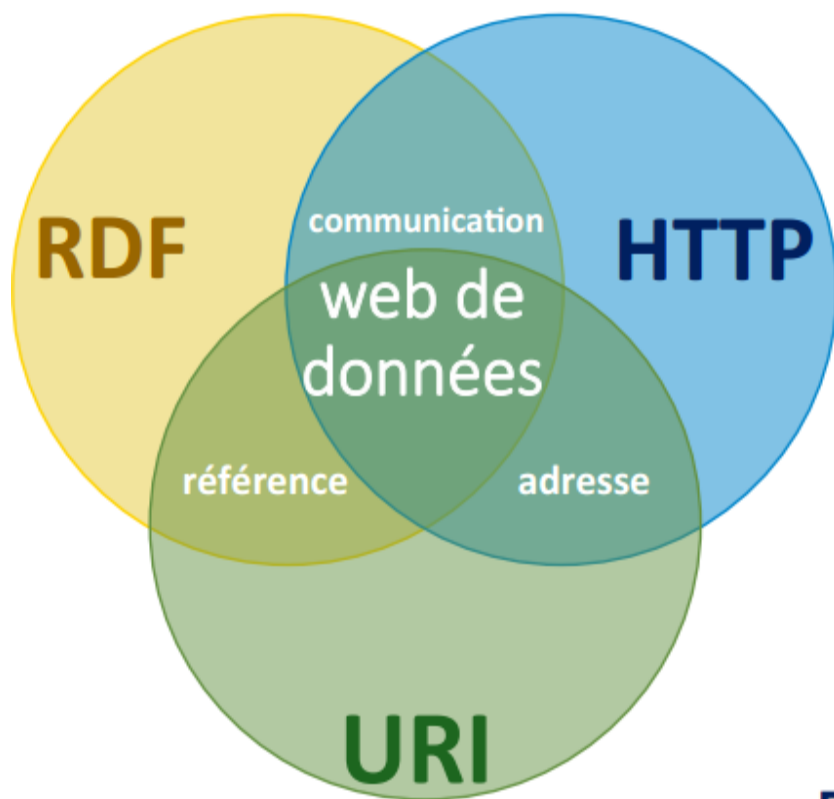
Décrire avec RDF

Plan du cours

1. RDF, c'est quoi ?
2. Syntaxe abstraite : triplets et graphes
3. Syntaxe de sérialisation : N-Triples, Turtle, RDF/XML
4. Valeurs, types, conteneurs, collections, etc.

Rappel - RDF : Décrire et échanger

RDF: modèle de base



Pile des standards du Web de données W

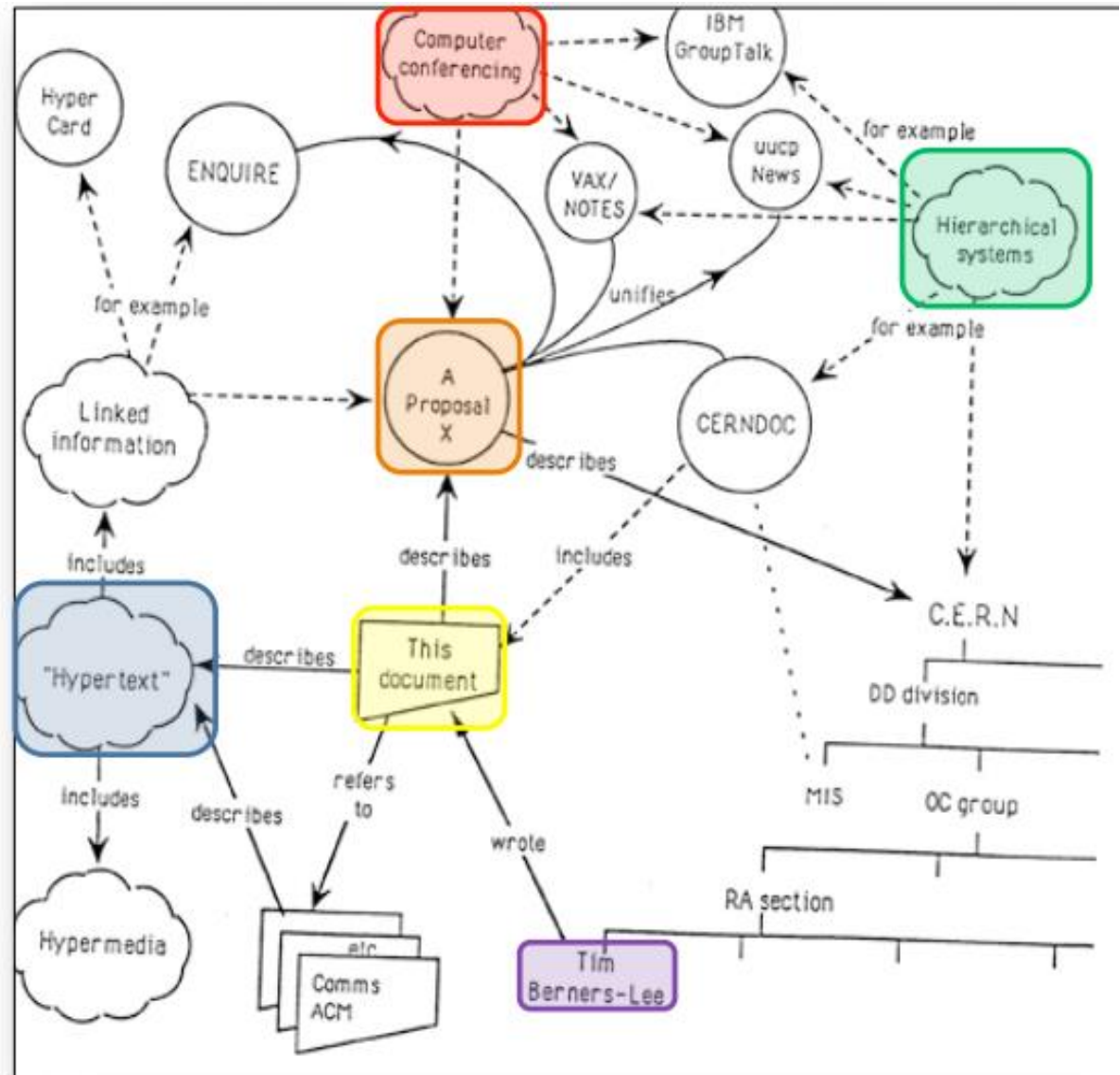
Rappel - Toile de ressources



Tim Berners-Lee



Information Management: A Proposal
Tim Berners-Lee, CERN, March 1989



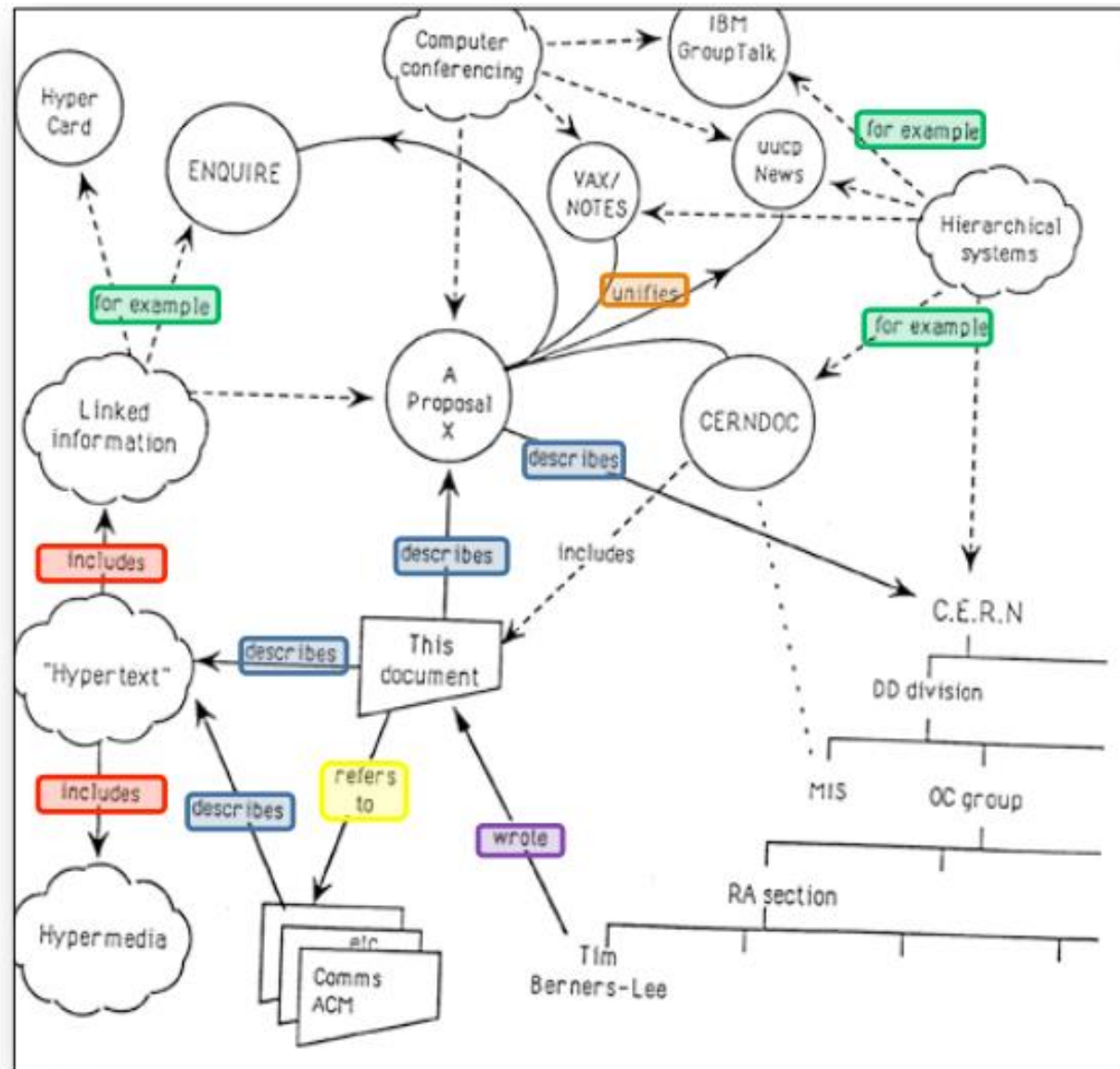
Rappel - Différents liens



Tim Berners-Lee



Information Management: A Proposal
Tim Berners-Lee, CERN, March 1989



RDF, c'est quoi ?

Resource : toute entité que l'on veut décrire sur le web mais qui n'est pas nécessairement accessible sur le web.

Description : attributs, caractéristiques, et relations entre ressources .

Framework : modèle de données et syntaxes pour ces descriptions.

- But : permettre que les informations sur les ressources soient manipulées par des applications, plutôt que d'être simplement affichées aux utilisateurs.
- Ouvert et extensible.

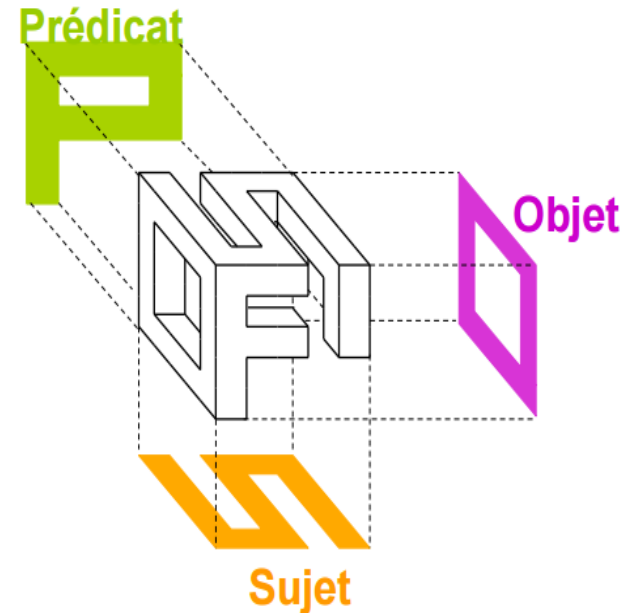


Syntaxe abstraite - Triplets et Graphes

RDF décompose les descriptions en collection de **triplets** :

(**Sujet**, **Prédictat**, **Objet**)

- Le triplet est l'atome de connaissance.
- Le **sujet** peut être une URI ou un nœud vide.
- Le **prédictat** = propriété, toujours une URI.
- L'**objet** = valeur propriété, URI, ou littéral, ou vide.



Syntaxe abstraite - Triplets et Graphes

RDF décompose les descriptions en collection de **triplets** :

(**Sujet**, **Prédictat**, **Objet**)

Exemple : Décrire une personne (URI: <http://www.ox.ac.uk/Profs#TimBernersLee>) qui s'appelle Tim Berners-Lee, qui travaille au département Informatique de l'université d'Oxford, et dont la page personnelle se trouve à l'URL suivante : <https://www.w3.org/People/Berners-Lee/>

Syntaxe abstraite - Triplets et Graphes

RDF décompose les descriptions en collection de **triplets** :

(**Sujet**, **Prédictat**, **Objet**)

Exemple : Décrire une personne qui s'appelle Tim Berners-Lee, qui travaille au département Informatique de l'université d'Oxford, et dont la page personnelle se trouve à l'URI suivante : <https://www.w3.org/People/Berners-Lee/>

(**Personne**, **s'appelle**, **Tim Berners-Lee**)

(**Personne**, **travaille à**, **département Informatique d'Oxford**)

(**Personne**, **a page personnelle**, <https://www.w3.org/People/Berners-Lee/>)

Syntaxe abstraite - Triplets et Graphes

Exemple :

(**Personne**, s'appelle, **Tim Berners-Lee**)

(**Personne**, travaille à, **département Informatique d'Oxford**)

(**Personne**, a page personnelle, <https://www.w3.org/People/Berners-Lee/>)

1 – Extraire les entités référées par la description – Sujet/Objet:

- ✓ La personne décrite
- ✓ Le nom de la personne
- ✓ Le lieu de travail
- ✓ La page personnelle

Syntaxe abstraite - Triplets et Graphes

Exemple :

(**Personne**, s'appelle, **Tim Berners-Lee**)

(**Personne**, travaille à, **département Informatique d'Oxford**)

(**Personne**, a page personnelle, <https://www.w3.org/People/Berners-Lee/>)

2 – Remplacer ces entités par des URI/Littéral – Sujet/Objet :

- ✓ La personne décrite `http://www.ox.ac.uk/Profs#TimBernersLee`
- ✓ Le nom de la personne `"Tim Berners-Lee"`
- ✓ Le lieu de travail `http://www.ox.ac.uk/Vocabulary#di`
- ✓ La page personnelle `https://www.w3.org/People/Berners-Lee/`

Syntaxe abstraite - Triplets et Graphes

Exemple :

(**Personne**, s'appelle, **Tim Berners-Lee**)

(**Personne**, travaille à, **département Informatique d'Oxford**)

(**Personne**, a page personnelle, <https://www.w3.org/People/Berners-Lee/>)

3 – Définir des URIs pour les relations entre ces entités - Prédicats:

- ✓ S'appelle `http://www.ox.ac.uk/Vocabulary#hasName`
- ✓ Travaille à `http://www.ox.ac.uk/Vocabulary#worksAt`
- ✓ A page personnelle `http://www.ox.ac.uk/Vocabulary#hasHomePage`

Syntaxe abstraite - Triplets et Graphes

Exemple :

(**Personne**, s'appelle, **Tim Berners-Lee**)

(**Personne**, travaille à, **département Informatique d'Oxford**)

(**Personne**, a page personnelle, <https://www.w3.org/People/Berners-Lee/>)

3 – Les triplets deviennent :

```
( http://www.ox.ac.uk/Profs#TimBernersLee ,  
  http://www.ox.ac.uk/Vocabulary#hasName ,  
  "Tim Berners-Lee")
```

```
( http://www.ox.ac.uk/Profs#TimBernersLee ,  
  http://www.ox.ac.uk/Vocabulary#worksAt ,  
  http://www.ox.ac.uk/Vocabulary#di)
```

```
( http://www.ox.ac.uk/Profs#TimBernersLee ,  
  http://www.ox.ac.uk/Vocabulary#hasHomePage ,  
  http://www.w3.org/People/Berners-Lee)
```

Syntaxe abstraite - Triplets et Graphes

- **Avantages** à utiliser les URI pour désigner les ressources et les propriétés :
 - ✓ Désambiguïser les désignations utilisées et de permettre à plusieurs applications de partager le même vocabulaire tout en évitant les conflits de noms.
 - ✓ Utilisées les propriétés comme des ressources.
 - ✓ Pouvoir donc ajouter des triplets RDF qui fournissent des informations sur ces propriétés.
- Les URI ont la forme `http://www...`, ne signifie pas qu'il s'agisse d'une URL correspondant à un document auquel on peut accéder sur le web.
- Une convention utilisée pour uniformiser les URI.

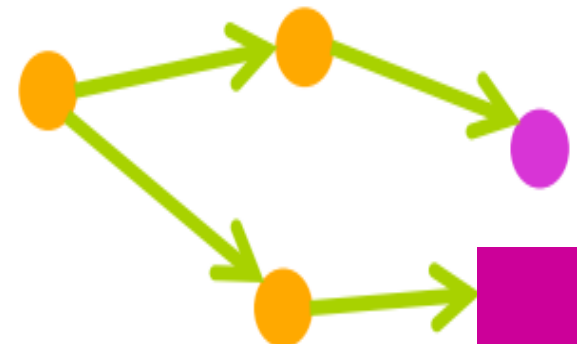
Syntaxe abstraite - Triplets et Graphes

➤ Référence URI - URIref

- ✓ Plus précisément, les ressources sont identifiées par des **URIrefs**.
- ✓ URIref = URI + **identificateur de fragment**.
- ✓ Exemple : <http://www.ox.ac.uk/Profs#TimBernersLee>
- ✓ URIrefs peuvent être relatives ou absolues.
 - URIref absolue fait référence à une ressource indépendamment du contexte dans lequel l'URIref a apparue.
 - URIref relative est une forme raccourcie d'une URIref absolue. Ex : **TimBernersLee** dans <http://www.ox.ac.uk/Profs#TimBernersLee>.

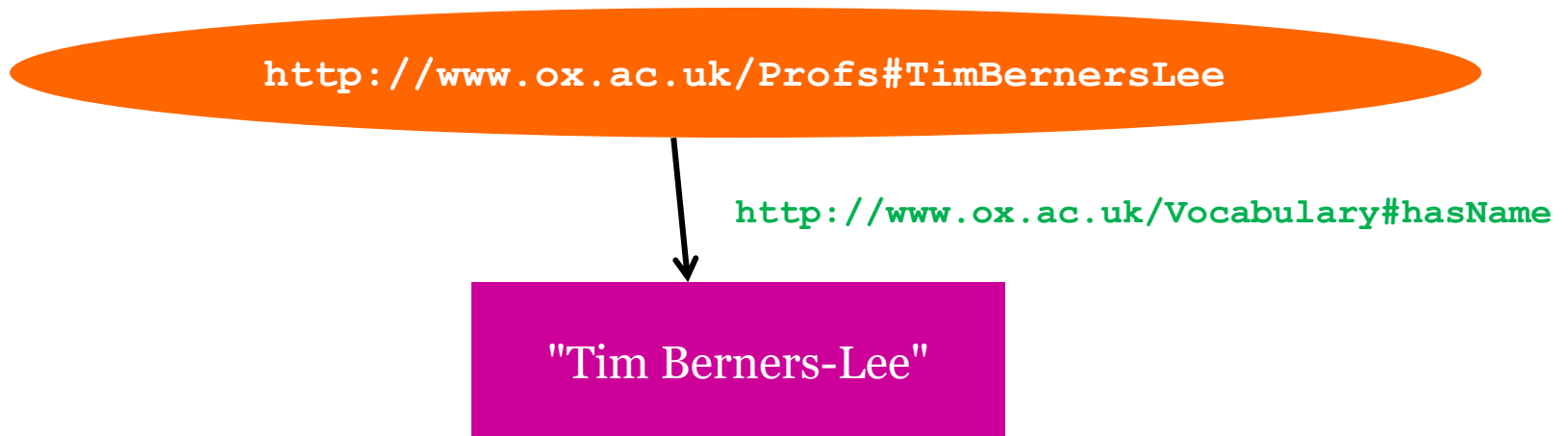
Syntaxe abstraite - Triplets et Graphes

- Une collection de triplets forme un **graphe**.
- **(Sujet, Prédicat, Objet)** => **(Noeud, Arc, Noeud)**
- Deux types de nœuds.
- **URI** représentés par une ellipse et **Littéral** représenté par un rectangle.



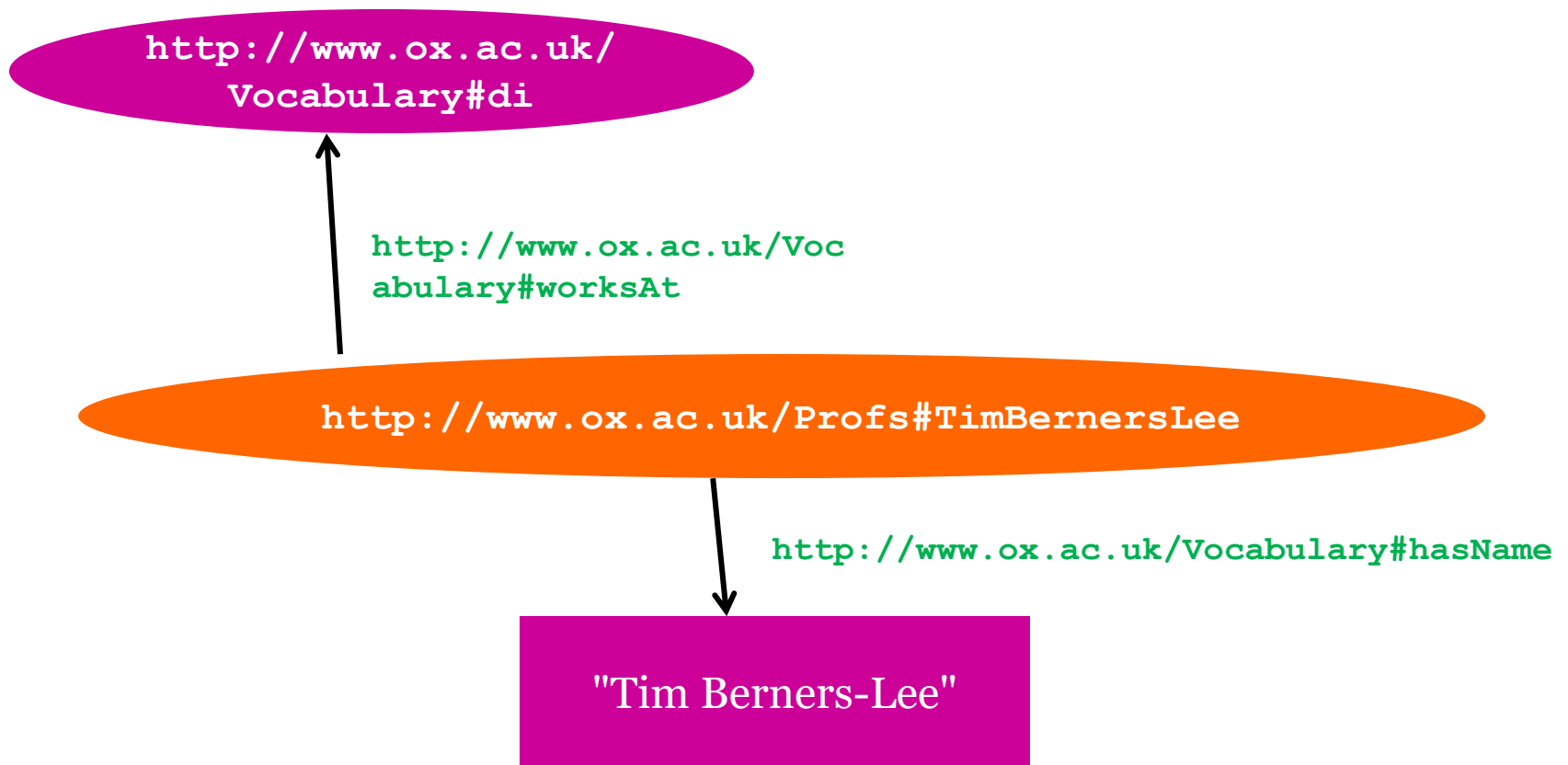
Syntaxe abstraite - Triplets et Graphes

Exemple :



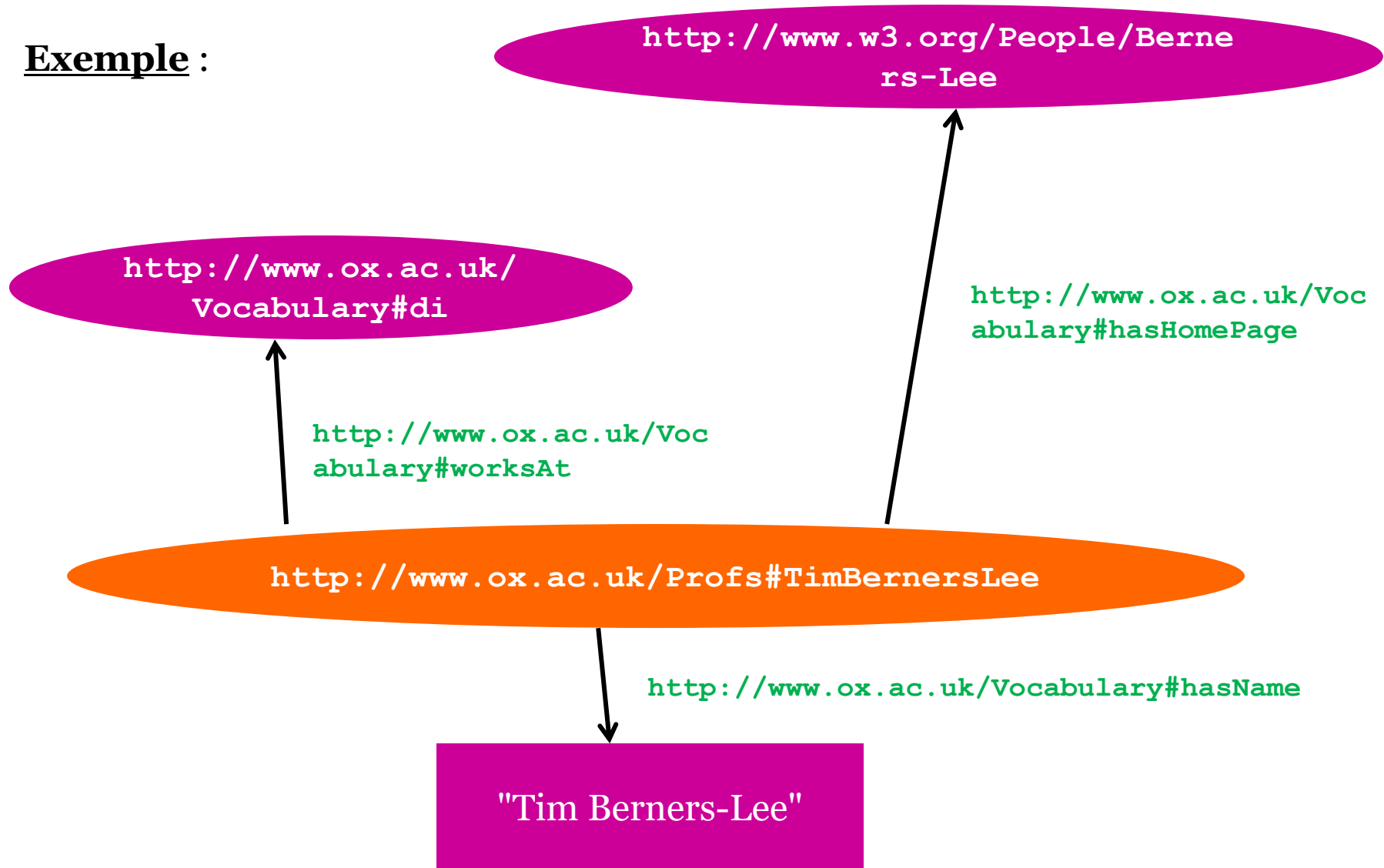
Syntaxe abstraite - Triplets et Graphes

Exemple :



Syntaxe abstraite - Triplets et Graphes

Exemple :



Syntaxe abstraite - Triplets et Graphes

- Deux types de **Littéral** dans un graphe.

Un **littéral simple** constitué d'une chaîne de caractères, appelée *forme lexicale*, et facultativement d'un attribut indiquant la langue.

Exemple : "Bonjour" ou "Bonjour"@fr

Un **littéral typé** est formé d'une chaîne de caractères (qui constitue la forme lexicale) et d'une URI indiquant un type qui sera utilisé pour décoder cette chaîne.

Exemple : "10"^^http://www.w3.org/2001/XMLSchema#integer

Syntaxe abstraite - Triplets et Graphes

➤ Remarques :

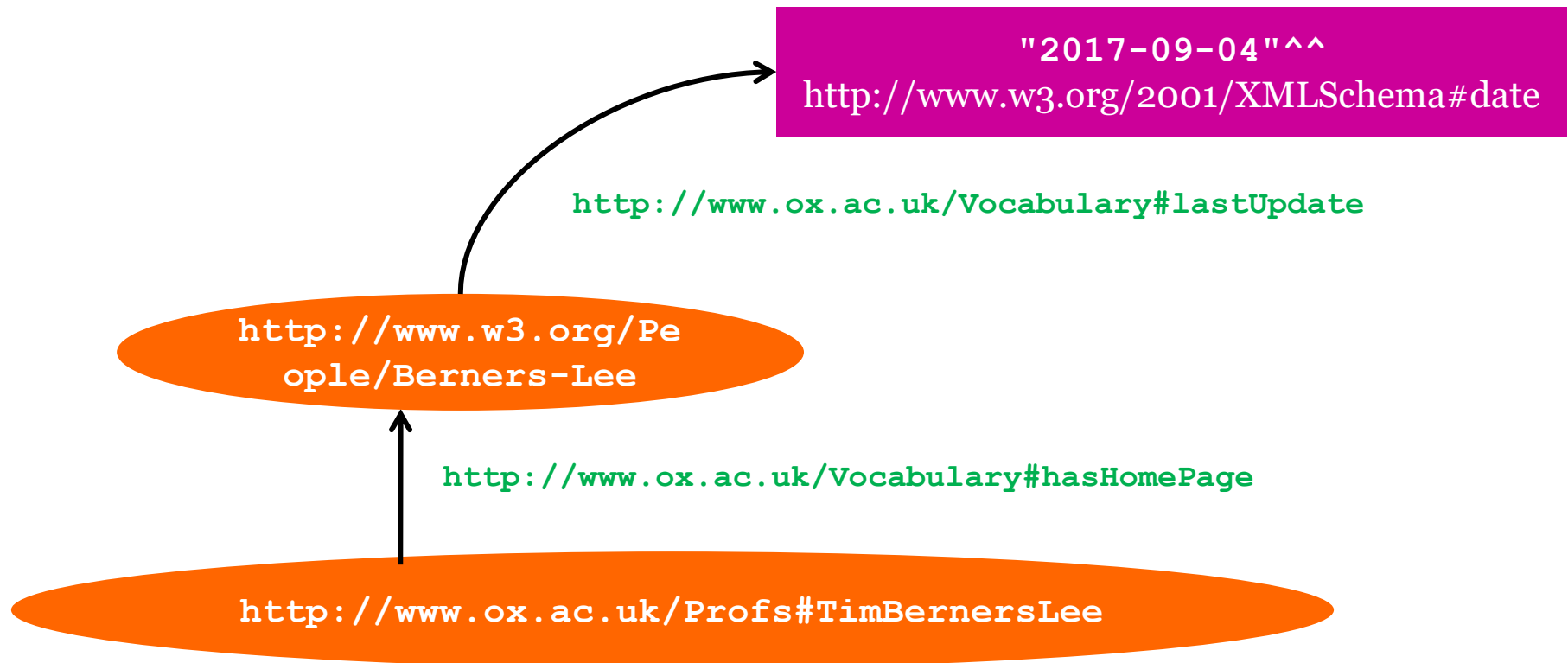
- ✓ RDF n'a pas de types prédéfinis.
- ✓ Il faudra utiliser une ressource externe à RDF pour interpréter un littéral typé. Exemple : XML Schema
- ✓ Rien n'empêche une application de définir ses propres types.



Syntaxe abstraite - Triplets et Graphes

- **Exemple** : Indiquer que la dernière date de mise à jour de la page personnelle Tim Berners-Lee est le 04 septembre 2017.

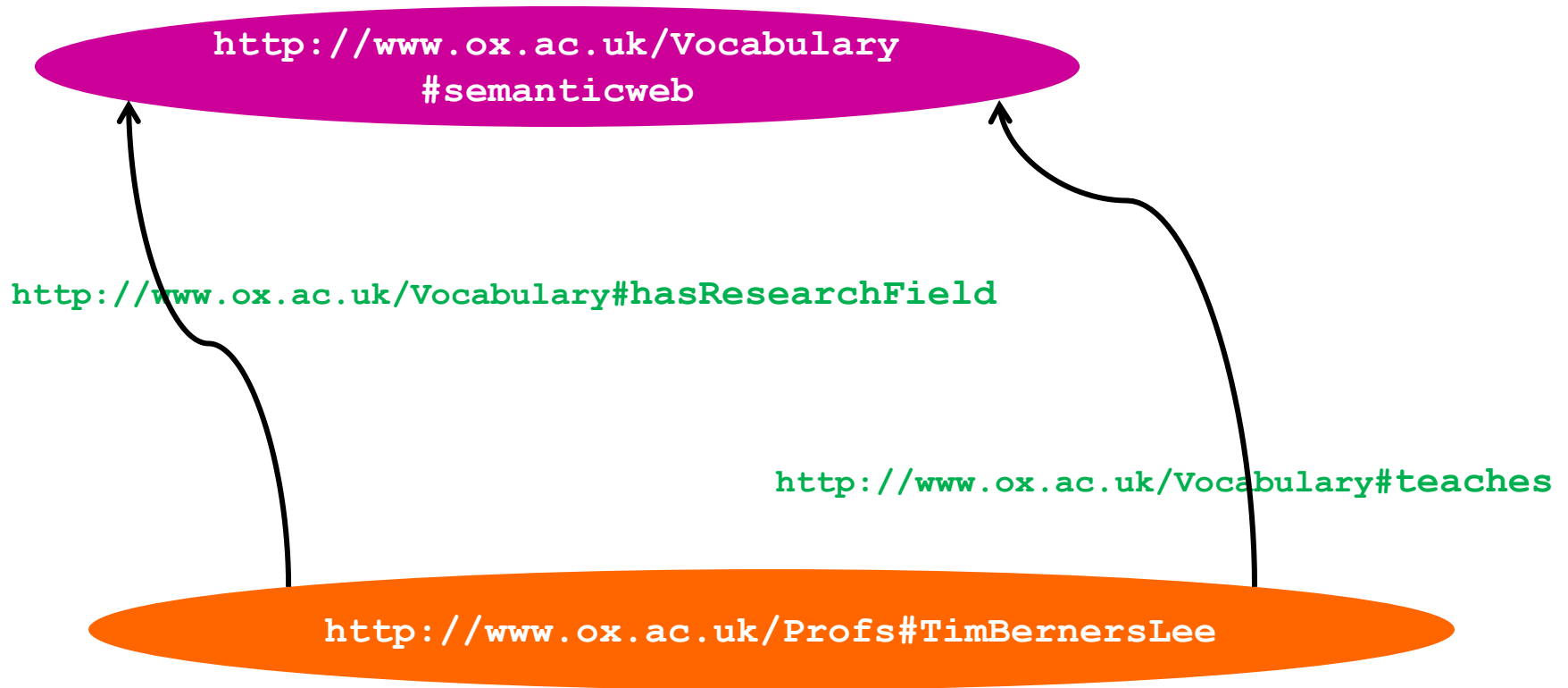
=> Utiliser le type date de XML Schema, plutôt qu'une chaîne de caractères par souci d'opérations de comparaison.



Syntaxe abstraite - Triplets et Graphes

➤ Caractéristiques :

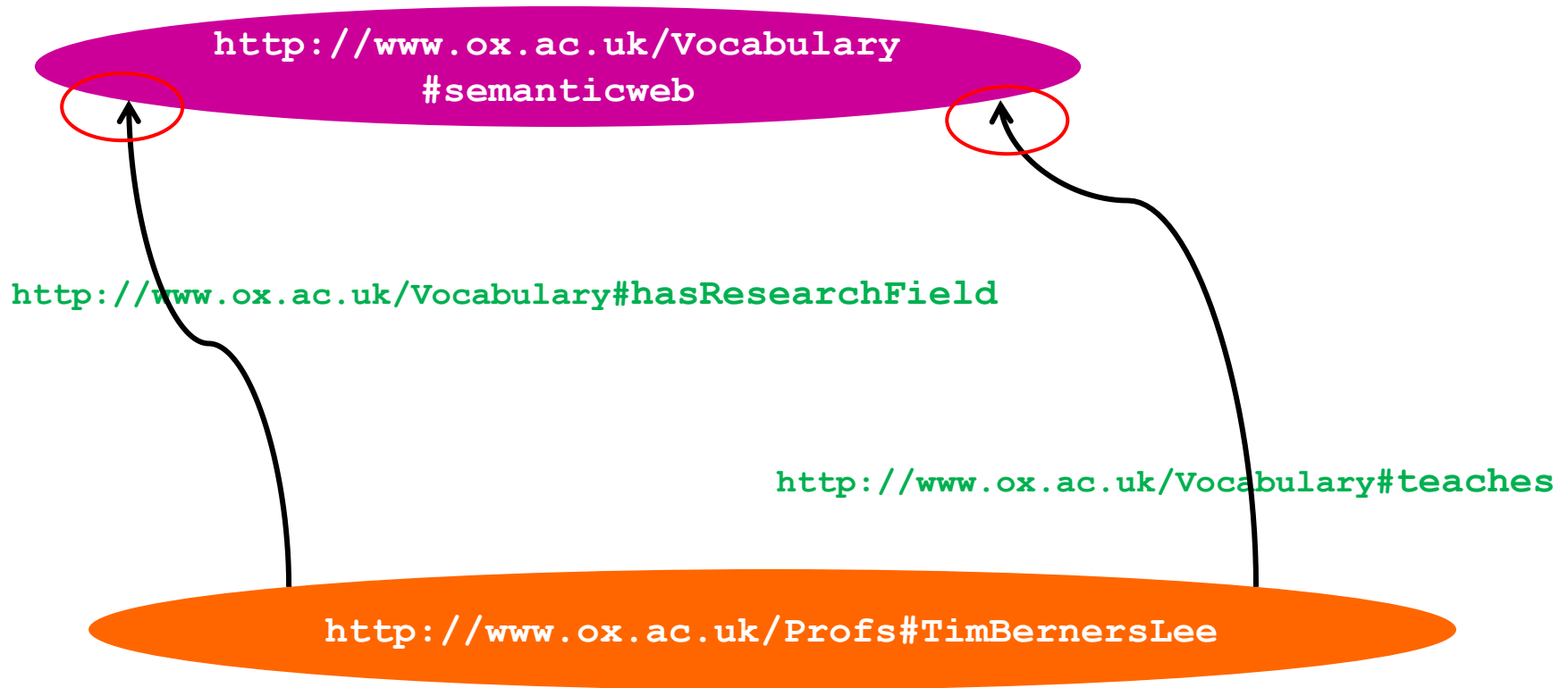
1 - Le graphe RDF est **multi-graphe** : plusieurs arcs entre mêmes nœuds.



Syntaxe abstraite - Triplets et Graphes

➤ Caractéristiques :

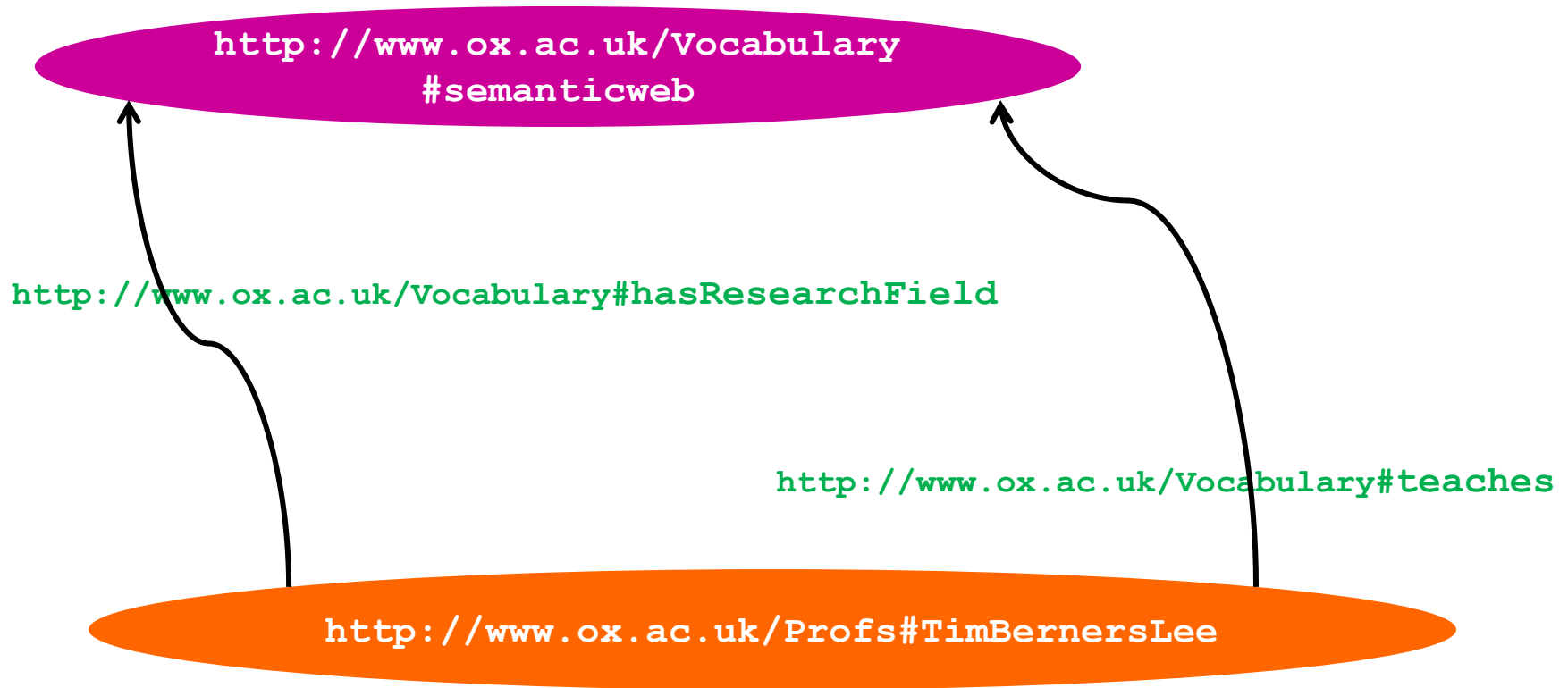
2 - Le graphe RDF est **orienté** : les arcs ont un sens (sujet vers objet).



Syntaxe abstraite - Triplets et Graphes

➤ Caractéristiques :

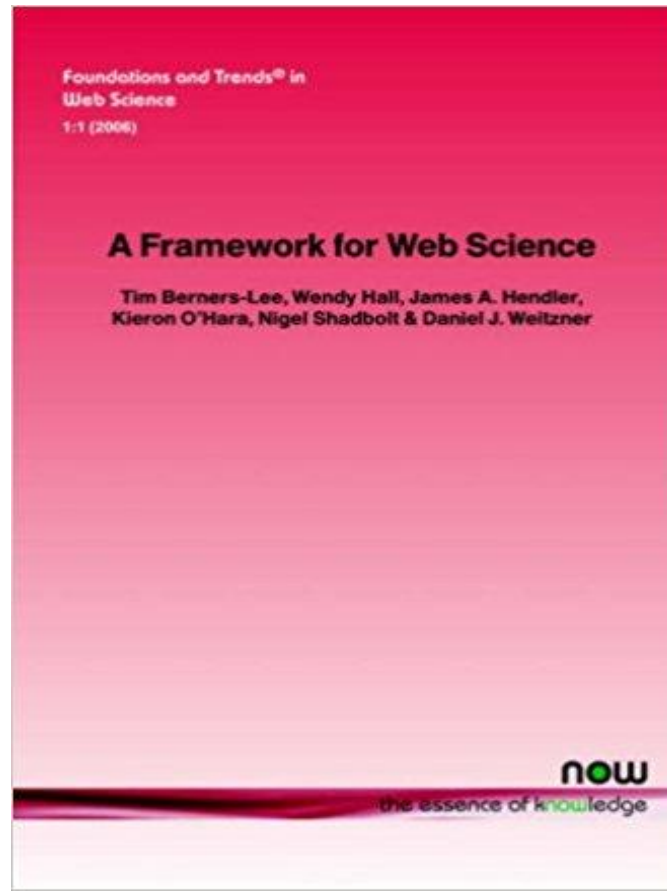
3 - Le graphe RDF est **étiqueté** : les nœuds et les arcs ont des étiquettes (URI/Littéral).



Syntaxe abstraite - Triplets et Graphes

➤ Caractéristiques :

4 - Le graphe RDF est **modèle extensible**: Possibilité d'union et de fusion de graphes



Tim Berners-Lee

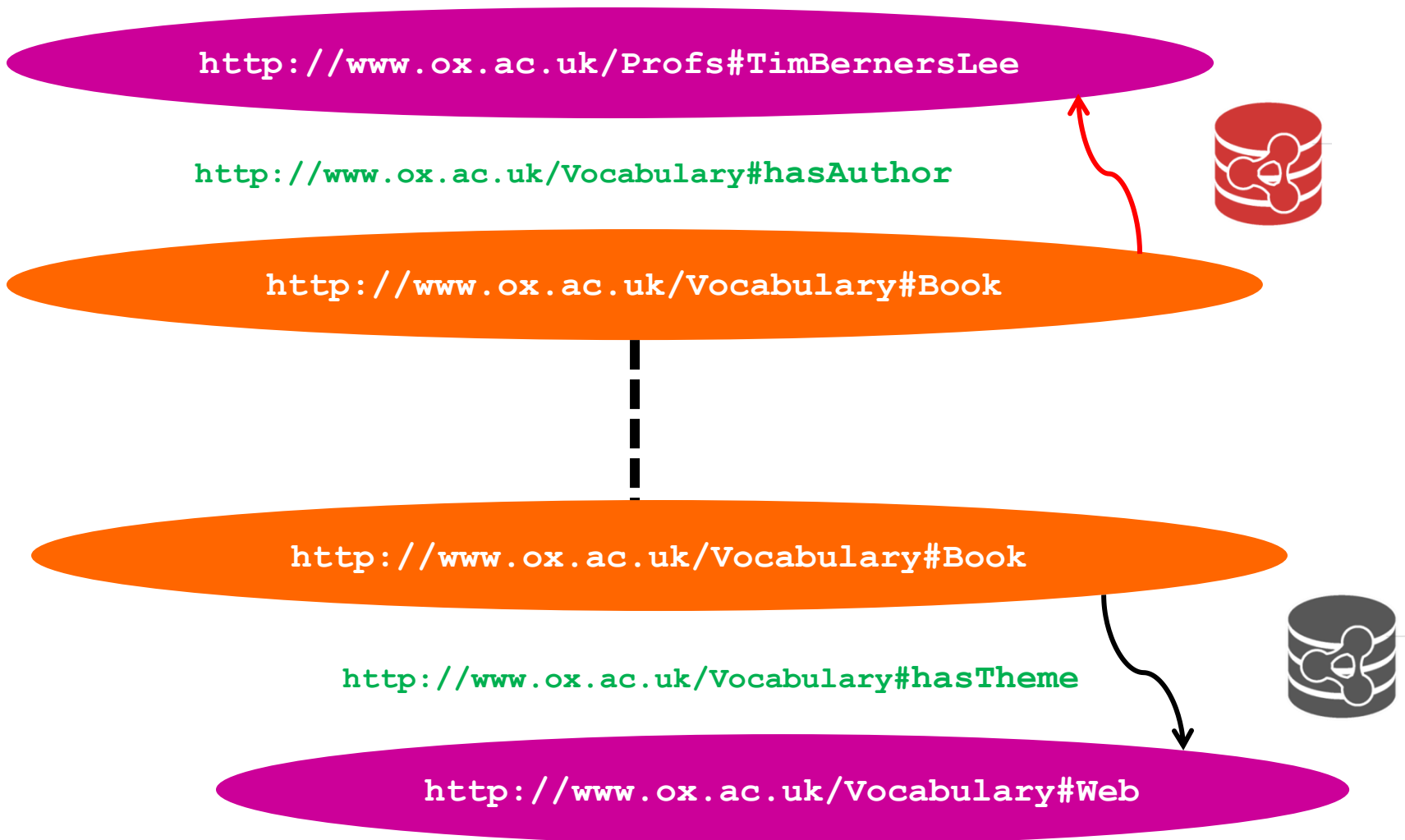


Wendy Hall



Syntaxe abstraite - Triplets et Graphes

Le graphe RDF est **modèle extensible**: Possibilité d'union et de fusion de graphes



Syntaxe abstraite - Triplets et Graphes

- Caractéristiques :
- ✓ Le graphe RDF est **modèle extensible**: Possibilité d'union et de fusion de graphes

This is the idea of GGG, the Giant Global Graph

Tim Berners-Lee: From World Wide Web to Giant Global Graph

By Dan Farber | November 22, 2007, 7:44am PST

Summary

Updated: On this Thanksgiving morning in the U.S., the Facebook Beacon storm continues to rage (Techmeme). It's simply growing pains for the social graph. In fact, the social graph (which Mark Zuckerberg defines as the network of connections between people) reached a new stage of legitimacy or recognition today with a post by Tim Berners-Lee, [...]



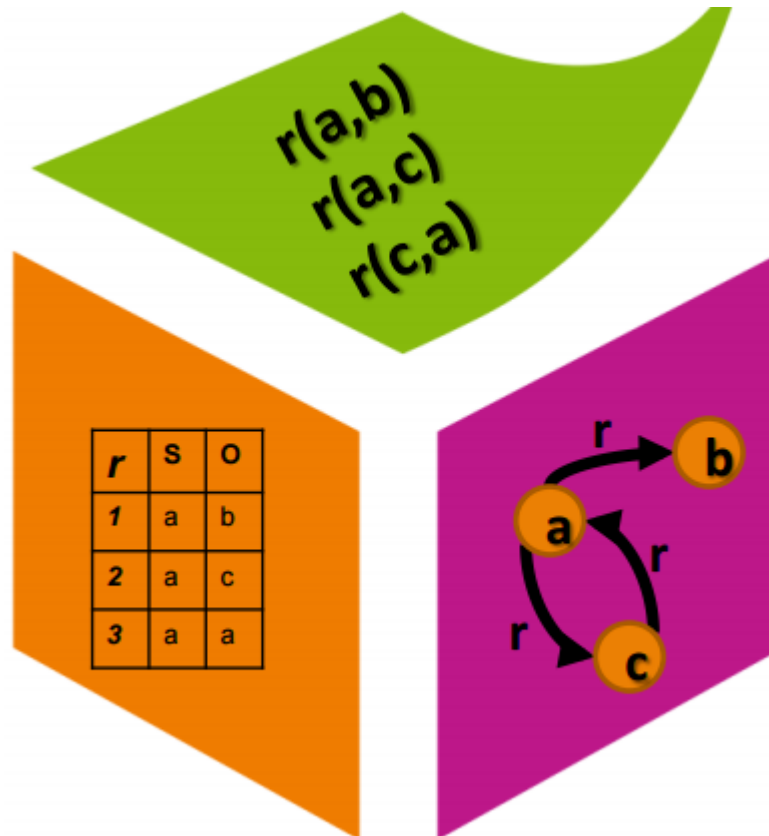
Updated: On this Thanksgiving morning in the U.S., the Facebook Beacon storm continues to rage (Techmeme). It's simply growing pains for the social graph. In fact, the social graph (which Mark Zuckerberg defines as the network of connections between people) reached a new stage of legitimacy or recognition today with a post by Tim Berners-Lee, who invented the World Wide Web in 1989.

In the post, he broadens the scope of the social graph, linking it to the Semantic Web, coming up with the Giant Global Graph. Nova Spivack of semantic Web startup Radar Networks has defined the semantic graph as "connecting people, places, companies, products, Web pages, videos, photos and turning it into semantic Web content."

Syntaxe abstraite - Triplets et Graphes

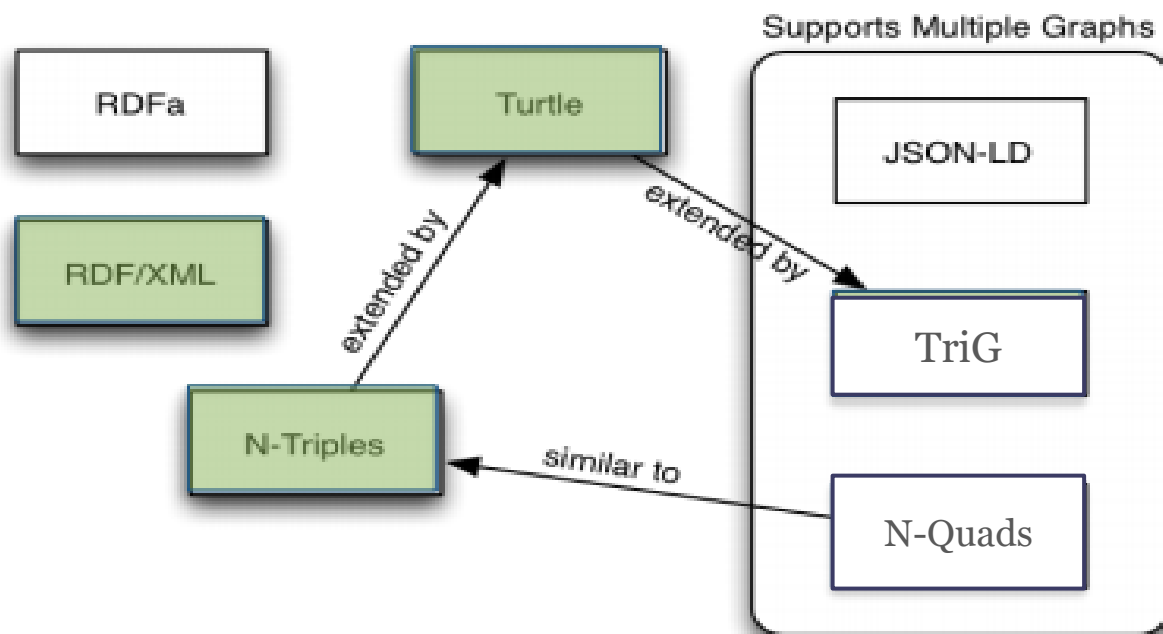
➤ Caractéristiques :

5 - Plusieurs vues sur un graphe RDF. Plusieurs façons de traitement.



Syntaxe de Sérialisation

- Les triplets et les graphes RDF peuvent être écrits dans plusieurs syntaxes de notation et de représentation :



Syntaxe de S rialisation - N-Triples

1. N-Triples

- Le graphe RDF est repr sent  par une collection de triplets de la forme :

Sujet Pr dicat Objet .

- Si URI :

<Sujet> <Pr dicat> <Objet> .

- Si litt ral : le repr senter directement sans modification.

- Si n ud vide : **_:ID**

- ID : identifiant unique permettant de r f rer au m me n ud vide dans deux triplets diff rents lorsqu'il est impliqu  dans plus d'une relation.

Syntaxe de Sérialisation - N-Triples

1. N-Triples

➤ Exemple:

{
 Sujet hasName Objet .
 Sujet worksAt Objet .
 Sujet hasHomePage Objet .
}

<http://www.ox.ac.uk/Profs#TimBernersLee>

<http://www.ox.ac.uk/Vocabulary#hasName> "Tim Berners-Lee" .

<http://www.ox.ac.uk/Profs#TimBernersLee>

<http://www.ox.ac.uk/Vocabulary#worksAt>

<http://www.ox.ac.uk/Vocabulary#di> .

<http://www.ox.ac.uk/Profs#TimBernersLee>

<http://www.ox.ac.uk/Vocabulary#hasHomePage>

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

Syntaxe de Sérialisation - N-Triples

1. N-Triples

➤ Exemple : Si **Sujet** était un nœud vide :

```
_:res123 <http://www.ox.ac.uk/Vocabulary#hasName> "Tim Berners-Lee" .
```

```
_:res123 <http://www.ox.ac.uk/Vocabulary#worksAt>  
        <http://www.ox.ac.uk/Vocabulary#di> .
```

```
_:res123 <http://www.ox.ac.uk/Vocabulary#hasHomePage>  
        < http://www.w3.org/People/Berners-Lee> .
```

Syntaxe de S rialisation - Turtle

2. Turtle - N3

- Le graphe RDF est repr sent  de mani re flexible et lisible.
- Une notation similaire N-Triples sauf qu'elle permet de d finir des pr fixes.
- Pr fixe : **@prefix nom_prefix : <URI> .**

Syntaxe de S rialisation - Turtle

2. Turtle - N3

➤ Exemple :

```
<http://www.ox.ac.uk/Profs#TimBernersLee>  
<http://www.ox.ac.uk/Vocabulary#hasName>  
<http://www.ox.ac.uk/Vocabulary#worksAt>  
<http://www.ox.ac.uk/Vocabulary#hasHomePage>
```

```
@prefix local: <http://www.ox.ac.uk/Vocabulary#> .
```

```
@prefix prof: <http://www.ox.ac.uk/Profs#> .
```

Syntaxe de S rialisation - Turtle

2. Turtle - N3

➤ Exemple:

```
<http://www.ox.ac.uk/Profs#TimBernersLee>  
<http://www.ox.ac.uk/Vocabulary#hasName>  
<http://www.ox.ac.uk/Vocabulary#worksAt>  
<http://www.ox.ac.uk/Vocabulary#hasHomePage>
```

```
@prefix local: <http://www.ox.ac.uk/Vocabulary#> .
```

```
@prefix prof: <http://www.ox.ac.uk/Profs#> .
```

```
prof:TimBernersLee local:hasName "Tim Berners-Lee" .
```

Syntaxe de S rialisation - Turtle

2. Turtle - N3

➤ Exemple:

```
<http://www.ox.ac.uk/Profs#TimBernersLee>  
<http://www.ox.ac.uk/Vocabulary#hasName>  
<http://www.ox.ac.uk/Vocabulary#worksAt>  
<http://www.ox.ac.uk/Vocabulary#hasHomePage>
```

```
@prefix local: <http://www.ox.ac.uk/Vocabulary#> .
```

```
@prefix prof: <http://www.ox.ac.uk/Profs#> .
```

```
prof:TimBernersLee local:hasName "Tim Berners-Lee" .
```

```
prof:TimBernersLee local:worksAt local:di .
```

Syntaxe de S rialisation - Turtle

2. Turtle - N3

➤ Exemple:

```
<http://www.ox.ac.uk/Profs#TimBernersLee>  
<http://www.ox.ac.uk/Vocabulary#hasName>  
<http://www.ox.ac.uk/Vocabulary#worksAt>  
<http://www.ox.ac.uk/Vocabulary#hasHomePage>
```

```
@prefix local: <http://www.ox.ac.uk/Vocabulary#> .
```

```
@prefix prof: <http://www.ox.ac.uk/Profs#> .
```

```
prof:TimBernersLee local:hasName "Tim Berners-Lee" .
```

```
prof:TimBernersLee local:worksAt local:di .
```

```
prof:TimBernersLee local:hasHomePage  
<http://www.w3.org/People/Berners-Lee> .
```

Syntaxe de S rialisation - Turtle

2. Turtle - N3

➤ Exemple: Combiner en une seule description

```
@prefix local: <http://www.ox.ac.uk/Vocabulary#> .
```

```
@prefix prof: <http://www.ox.ac.uk/Profs#> .
```

```
prof:TimBernersLee local:hasName "Tim Berners-Lee" .
```

```
prof:TimBernersLee local:worksAt local:di .
```

```
prof:TimBernersLee local:hasHomePage  
<http://www.w3.org/People/Berners-Lee> .
```

Syntaxe de S rialisation - Turtle

2. Turtle - N3

➤ Exemple: Combiner en une seule description

```
@prefix local: <http://www.ox.ac.uk/Vocabulary#> .
```

```
@prefix prof: <http://www.ox.ac.uk/Profs#> .
```

```
prof:TimBernersLee
```

```
local:hasName "Tim Berners-Lee" ;
```

```
local:worksAt local:di ;
```

```
local:hasHomePage <http://www.w3.org/People/Berners-Lee> .
```


Syntaxe de S rialisation - Turtle

2. Turtle - N3

- Exemple: Cas URI de la personne est un n ud vide

```
@prefix  local: <http://www.ox.ac.uk/Vocabulary#> .
```

```
[ ]
```

```
local:hasName "Tim Berners-Lee" ;
```

```
local:worksAt local:di ;
```

```
local:hasHomePage <http://www.w3.org/People/Berners-Lee> .
```

Syntaxe de S rialisation - Turtle

2. Turtle - N3

➤ Exemple: Cas URI de la personne est un n ud vide

ou :

```
@prefix  local: <http://www.ox.ac.uk/Vocabulary#> .
```

```
[  local:hasName  "Tim Berners-Lee"  ;  
  local:worksAt  local:di  ;  
  local:hasHomePage  <http://www.w3.org/People/Berners-Lee>  ] .
```

Syntaxe de Sérialisation - Turtle

2. Turtle - N3

➤ Exemple: Nœud vide comme objet d'un triplet

```
@prefix  local: <http://www.ox.ac.uk/Vocabulary#> .
```

```
@prefix  prof: <http://www.ox.ac.uk/Profs#> .
```

```
prof:WendyHall  local:knows
```

```
    [  local:hasName  "Tim Berners-Lee"  ;
```

```
      local:worksAt  local:di  ;
```

```
      local:hasHomePage  <http://www.w3.org/People/Berners-Lee> ] .
```

Web Sémantique - Outils

Rhizomik – RDF to SVG Form <https://rdf2svg.redefer.rhizomik.net/>



Rhizomik **ReDeFer**

RDF to SVG

Render RDF URL as SVG

GET /render?url=URL

URL ...pointing to RDF content

⚙ Render as SVG

Examples

- [mpeg7.rdf](#)
- [foaf.rdf](#)
- [test.nt](#)
- [roberto.ttl](#)
- [emotions&cognitionontology.owl](#)
- [mooc-sample.ttl](#)

Map RDF to SVG

POST /render

Content-Type: text/turtle | application/rdf+xml | application/n-triples

Turtle (.ttl)

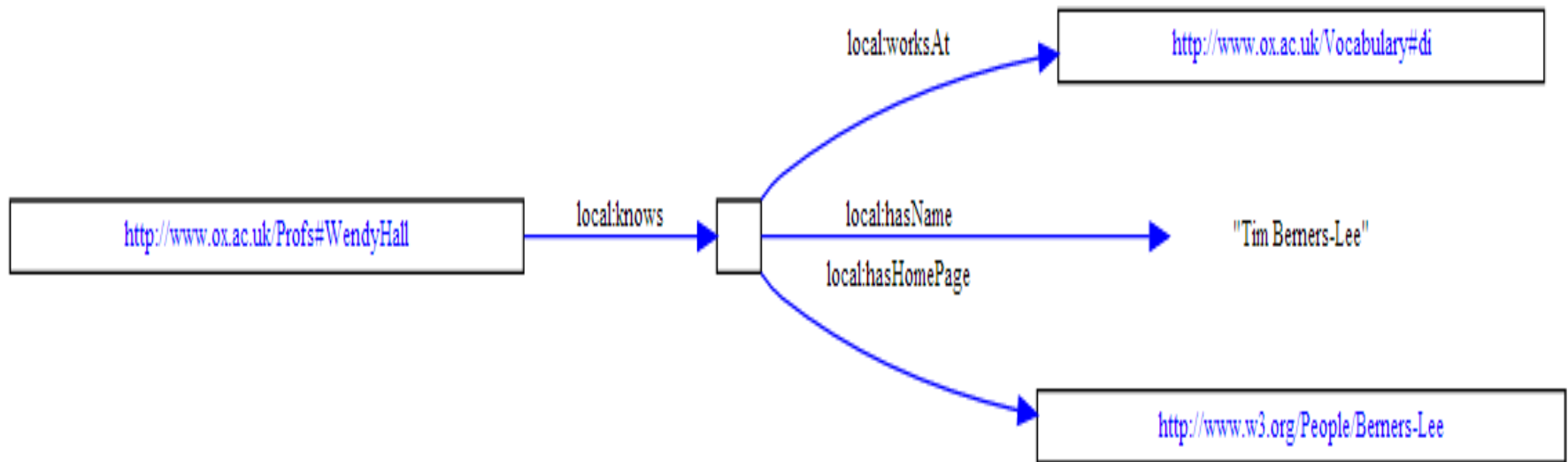
```
@prefix local: <http://www.ox.ac.uk/Vocabulary#> .  
@prefix prof: <http://www.ox.ac.uk/Profs#> .
```

```
prof:WendyHall local:knows  
[ local:hasName "Tim Berners-Lee" ;  
  local:worksAt local:di ;
```

Web Sémantique - Outils

Rhizomik – RDF to SVG Form

<https://rdf2svg.redefer.rhizomik.net/>



Powered by <http://rhizomik.net>

Syntaxe de Sérialisation - RDF/XML

3. RDF/XML (Cours 3)

➤ Exemple: **Racine**

```
<?xml version="1.0"?>
```

```
<rdf:RDF>
```

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

Syntaxe de Sérialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
```

```
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
```

namespace

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

Syntaxe de Sérialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
```

```
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
```

```
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
```

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

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

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

Descriptions de
ressources

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


Syntaxe de S rialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
```

```
  .... >
```

```
    <rdf:Description      rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

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

```
    <rdf:Description      rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

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

```
    <rdf:Description      rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

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

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

Syntaxe de S rialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
```

```
  .... >
```

```
    <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

```
      <local:hasName>Tim Berners-Lee</local:hasName>
```

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

```
    <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

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

```
    <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

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

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

Syntaxe de Sérialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
```

```
  .... >
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

```
    <local:hasName>Tim Berners-Lee</local:hasName>
```

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

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

```
    <local:worksAt rdf:resource="http://www.ox.ac.uk/Vocabulary#di"/>
```

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

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

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

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

Syntaxe de Sérialisation - RDF/XML

```
<?xml version="1.0"?>
```

```
<rdf:RDF
```

```
  xmlns:local="http://www.ox.ac.uk/Vocabulary#"
```

```
  .... >
```

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

```
    <local:hasName>Tim Berners-Lee</local:hasName>
```

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

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

```
    <local:worksAt rdf:resource="http://www.ox.ac.uk/Vocabulary#di"/>
```

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

```
  <rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >
```

```
    <local:hasHomePage rdf:resource="http://www.w3.org/People/Berners-Lee"/>
```

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

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

Syntaxe de Sérialisation - RDF/XML

Abréviation 1 : Combiner

```
<rdf:Description    rdf:about="http://www.ox.ac.uk/Profs#TimLee" >  
  <local:hasName>Tim Berners-Lee</local:hasName>  
  <local:worksAt rdf:resource=" http://www.ox.ac.uk/Vocabulary#di"/>  
  <local:hasHomePage rdf:resource=" http://www.w3.org/People/Berners-Lee"/>  
</rdf:Description>
```

Syntaxe de Sérialisation - RDF/XML

Abréviation 2 : Préfixes des ressources

```
<!DOCTYPE rdf:RDF [  
    <!ENTITY local "http://www.polymtl.ca/Vocabulary#">  
    <!ENTITY prof "http://www.polymtl.ca/Profs#">  
    ]>  
  
<rdf:RDF ... >  
    <rdf:Description rdf:about="&prof;TimLee" >  
        <local:hasName>Tim Berners-Lee</local:hasName>  
        <local:worksAt rdf:resource="&local;di"/>  
        <local:hasHomePage rdf:resource="http://www.w3.org/People/Berners-Lee"/>  
    </rdf:Description>  
  
</rdf:RDF>
```

Syntaxe de S rialisation - RDF/XML

Abr viation 3 : URI de base et **rdf:ID**

<rdf:RDF

xmlns:local="http://www.ox.ac.uk/Vocabulary#"

xml:base="http://www.ox.ac.uk/Profs"

.... >

<rdf:Description **rdf:ID**="TimLee" >

<local:hasName>Tim Berners-Lee</local:hasName>

<local:worksAt **rdf:resource**="http://www.ox.ac.uk/Vocabulary#di"/>

<local:hasHomePage **rdf:resource**="http://www.w3.org/People/Berners-Lee"/>

</rdf:Description>

</rdf:RDF>

Syntaxe de S rialisation - RDF/XML

N ud vide ne peut  tre que le sujet
ou l'objet, jamais une propri t 

Repr sentation des n uds vides

`<rdf:Description rdf:nodeID="id"> </rdf:Description >`

Exemple :

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<!ENTITY local "http://www.polymtl.ca/Vocabulary#">
    <!ENTITY prof "http://www.polymtl.ca/Profs#">]>

<rdf:RDF ... >
  <rdf:Description rdf:nodeID="id01">
    <local:hasName>Tim BernersLee</local:hasName>
    <local:worksAt rdf:resource=" &local;di" />
    <local:hasHomePage rdf:resource="http://www.w3.org/People/Berners-Lee" />
  </rdf:Description>
</rdf:RDF>
```


Syntaxe de S rialisation - RDF/XML

N ud vide ne peut  tre que le sujet
ou l'objet, jamais une propri t 

Repr sentation des n uds vides

`<rdf:Description rdf:nodeID="id"> </rdf:Description >`

Exemple :

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<!ENTITY local "http://www.polymtl.ca/Vocabulary#">
    <!ENTITY prof "http://www.polymtl.ca/Profs#">]>
<rdf:RDF ... >
  <rdf:Description rdf:nodeID="id01">
    <local:hasName>Tim BernersLee</local:hasName>
  </rdf:Description>
  <rdf:Description rdf:nodeID="id01">
    <local:worksAt rdf:resource=" &local;di" />
  </rdf:Description>
</rdf:RDF>
```

Syntaxe de Sérialisation - RDF/XML

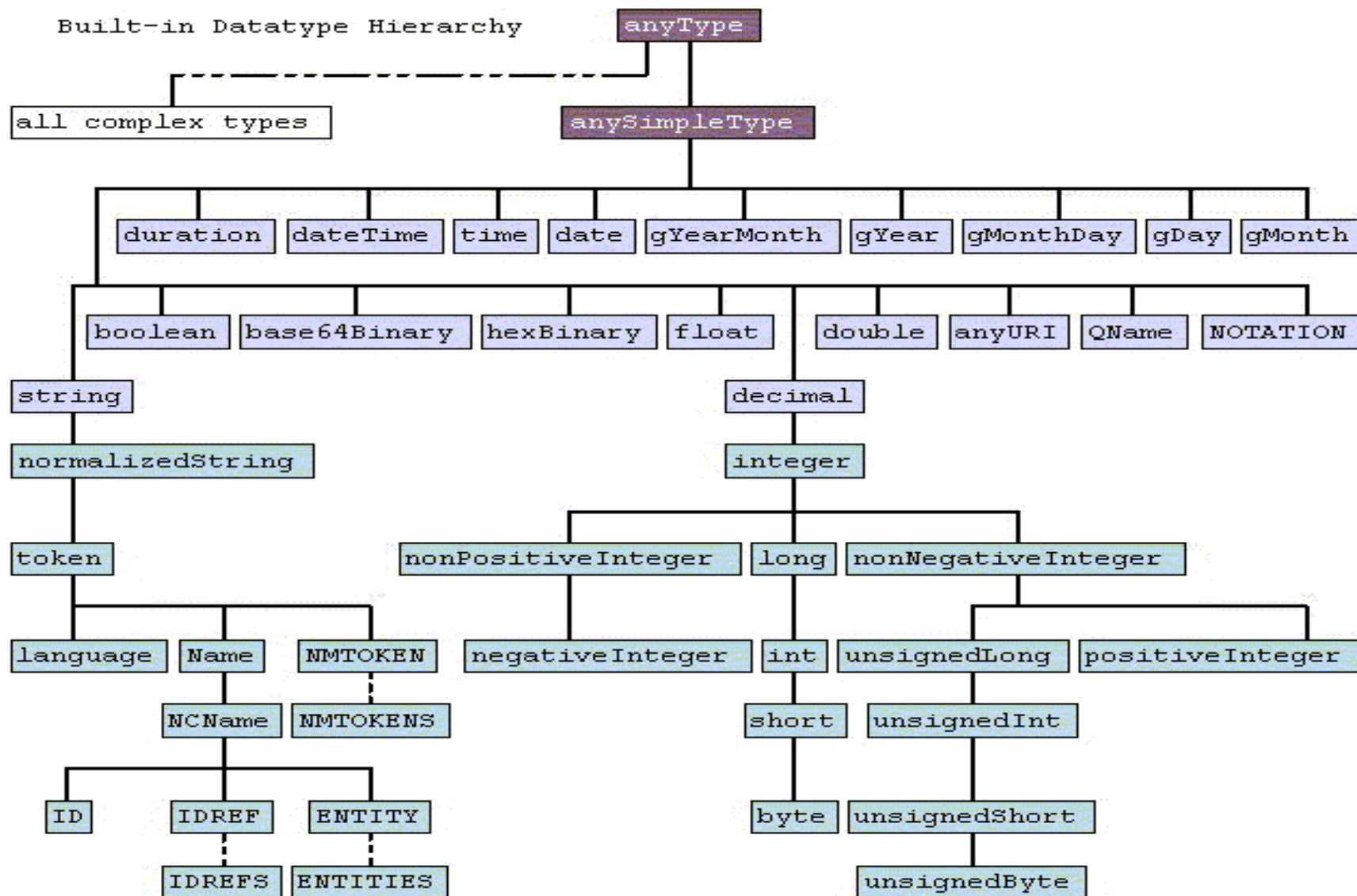
Représentation des Littéraux - Types

- Les littéraux standards sont des chaînes de caractères.
- Les littéraux typés peuvent être de différents types : date, integer, etc.
- « XML schema datatypes » pour typer les littéraux.

<http://www.w3.org/2001/XMLSchema#>

- `rdf:datatype`

Syntaxe de S rialisation - RDF/XML



Syntaxe de Sérialisation - RDF/XML

Représentation des Littéraux - Types

➤ Exemple : date

```
<rdf:Description  rdf:about="http://www.w3.org/People/Berners-Lee">  
  <local:lastUpdate  
    rdf:datatype="http://www.w3.org/2001/XMLSchema#date">  
    2017-09-04  
  </local:lastUpdate>  
</rdf:Description>
```

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

<http://www.ox.ac.uk/Vocabulary#lastUpdate>

"2017-09-04"^^xsd:date

Syntaxe de Sérialisation - RDF/XML

Représentation des Littéraux - Types

➤ Exemple : integer

```
<rdf:Description  rdf:about="http://www.w3.org/People/Berners-Lee">  
  <local:numberWords  
    rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">  
      12345  
    </local:numberWords>  
</rdf:Description>
```

http://www.w3.org/People/Berners-Lee

http://www.ox.ac.uk/Vocabulary#numberWords

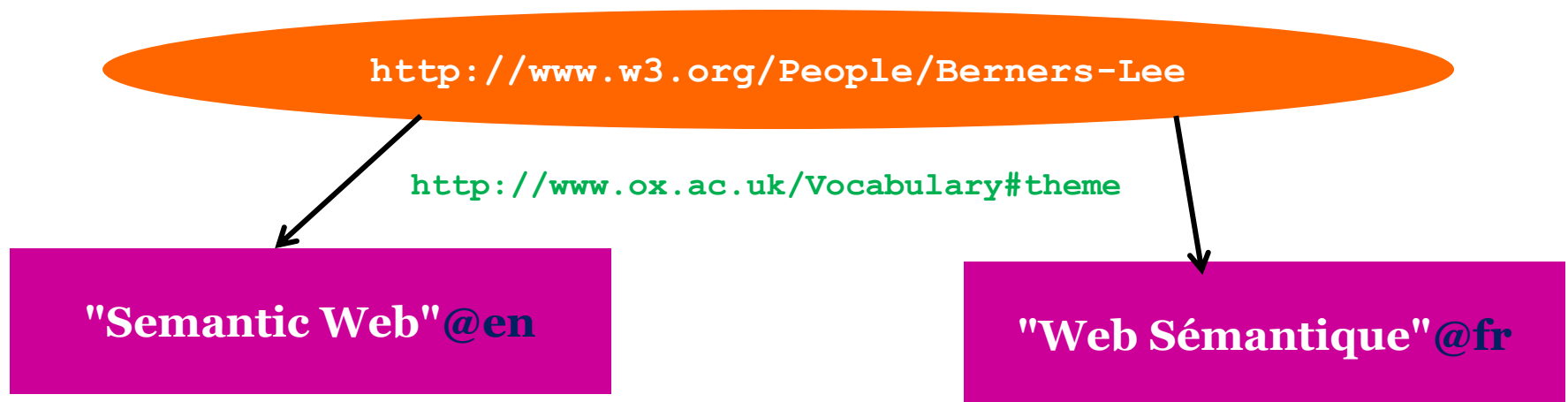
"12345"^^xsd:integer

Syntaxe de Sérialisation - RDF/XML

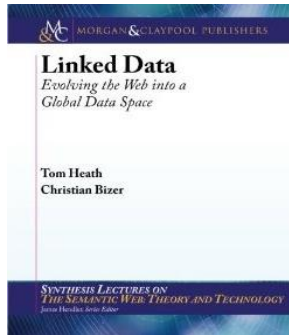
Représentation des Littéraux - Types

- Exemple : les littéraux peuvent être associés à une langue .

```
<rdf:Description  rdf:about="http://www.w3.org/People/Berners-Lee">  
  <local:theme  xml:lang='fr'>Web Sémantique</local:theme>  
  <local:theme  xml:lang='en'>Semantic Web</local:theme>  
</rdf:Description>
```

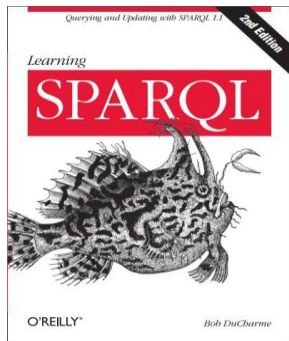


Références



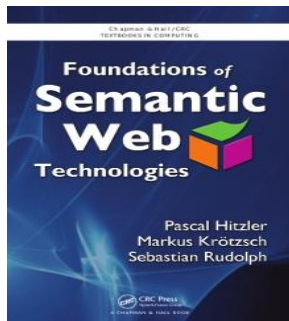
Linked Data: Evolving the Web into a Global Data Space

- ✓ Auteur : Christian Bizer, Tom Heath
- ✓ Éditeur : Morgan & Claypool Publishers
- ✓ Edition : Février 2011 - 136 pages - ISBN 9781608454310



Learning SPARQL : Querying and Updating with SPARQL

- ✓ Auteur : Bob DuCharme
- ✓ Éditeur : O'Reilly Media
- ✓ Edition: Juillet 2013– 386pages -ISBN : 9781449306595



Foundations of Semantic Web Technologies

- ✓ Auteur : Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph
- ✓ Éditeur : CRC Press/Chapman and Hall
- ✓ Edition : 2009 - 455 pages - ISBN : 9781420090505

Références

- W3C – Semantic Web
 - ✓ https://www.w3.org/2001/sw/wiki/Main_Page
- INRIA MOOC - Fabien Gandon– Web Sémantique et Web de Données
 - ✓ https://www.canal-u.tv/producteurs/inria/cours_en_ligne/web_semantique_et_web_de_donnees
- Cours en Ligne- Michel Gagnon– Ontologies et Web Sémantique
 - ✓ <https://moodle.polymtl.ca/course/view.php?id=30>
- Initiation à RDF - W3C
 - ✓ <http://www.yoyodesign.org/doc/w3c/rdf-primer>