

Ontologies et

Web Sémantique

Décrire avec RDF – Suite : RDF-Schema

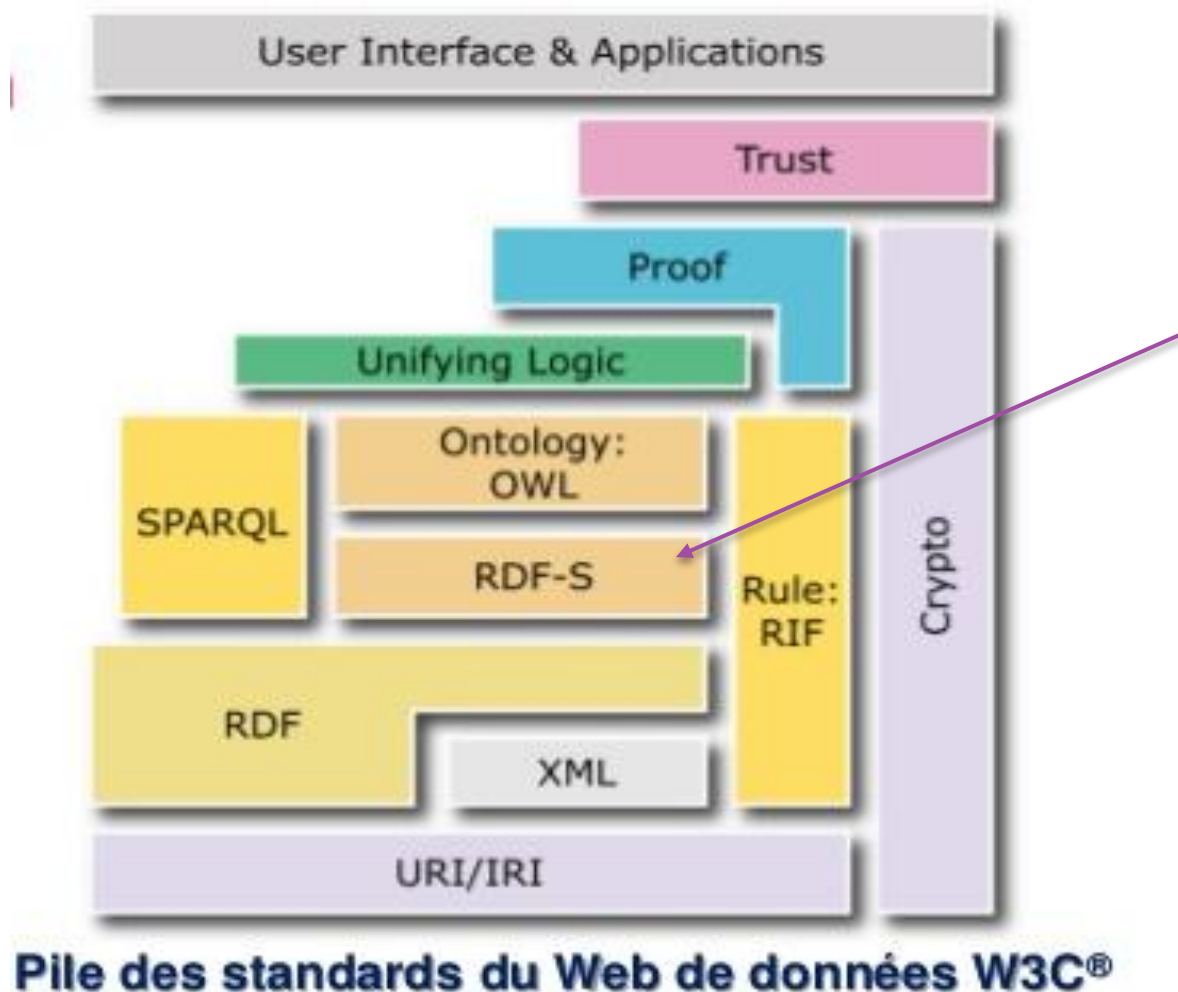
Plan du cours

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.
5. **RDF Schema - RDFS**

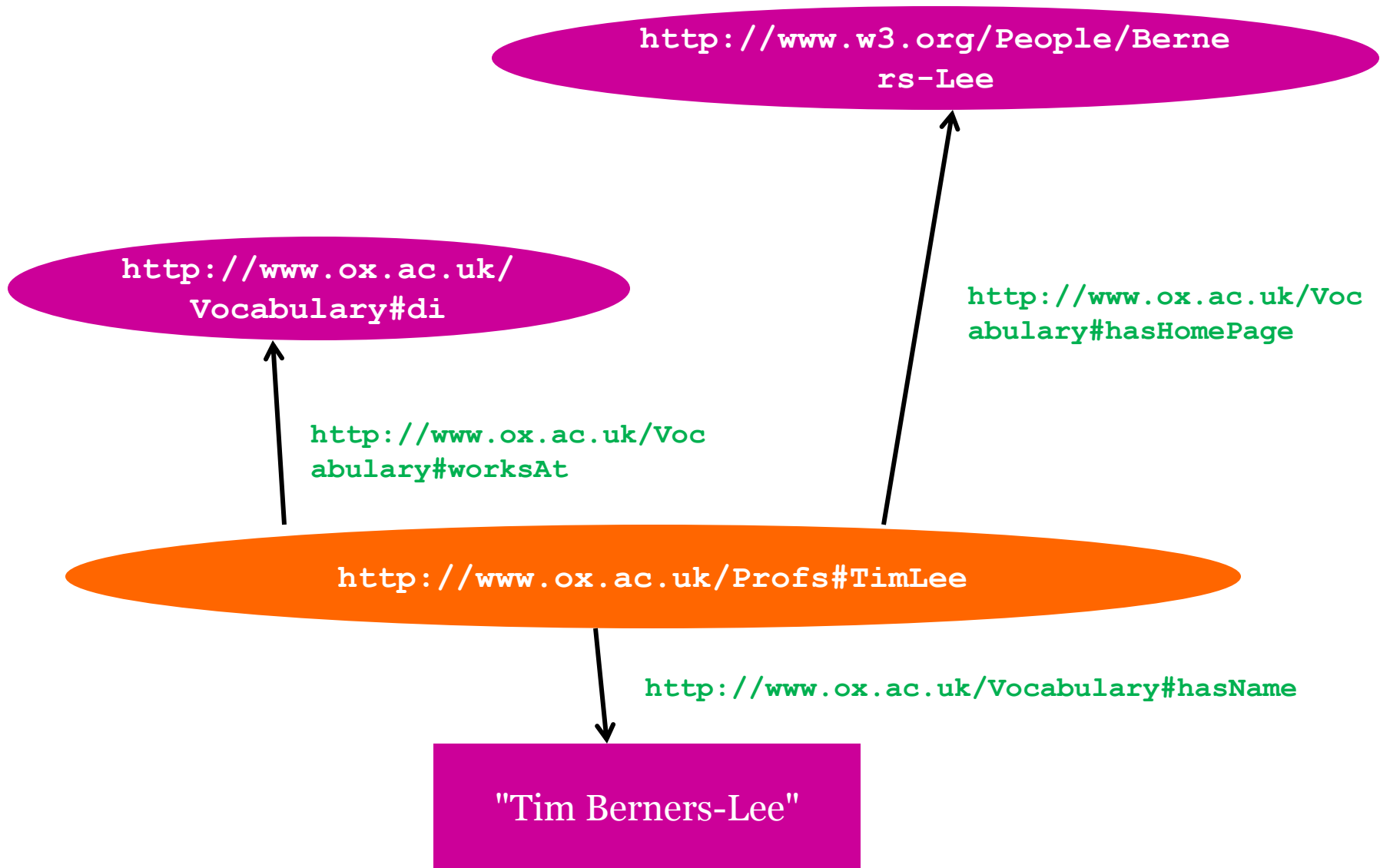
Plan du cours

1. RDFS, définitions
2. RDF Vs. RDFS
3. Définitions de classes et propriétés en RDFS
4. Règles d'inférence
5. Limites de RDFS

La pile des Standards - Layer Cake



RDF - Triplets et Graphes



RDF Schema - RDFS

- **Limitation RDF** - permet de définir des **graphes étiquetés** - Graphes RDF, en utilisant des ressources du web **sans vraiment de sémantique**.
 - Pour donner de la sémantique des étiquettes du graphe RDF, il faut des **vocabulaires plus riches**.
 - Ces vocabulaires permettront :
 - d'améliorer l'indexation des contenus en ligne,
 - un traitement plus efficace des requêtes,
 - des réponses plus pertinentes, et une meilleure interopérabilité des systèmes.
 - Pour formaliser cette **sémantique** on utilisera des **ontologies**.
 - Les ontologies permettent aussi un enrichissement des données en utilisant de l'inférence.
- => RDFS** (RDF-Schema), extension de RDF, permet de construire des ontologies légères basées sur RDF.

RDF Schema - RDFS

`http://www.ox.ac.uk/Profs#TimLee`



`http://www.ox.ac.uk/Vocabulary#hasName`

`https://dbpedia.org/page/Tim_Berners-Lee`

Syntaxiquement juste,
sémantiquement faux

RDF Schema - RDFS

`http://www.ox.ac.uk/Profs#TimLee`



`http://www.ox.ac.uk/Vocabulary#hasName`

`https://dbpedia.org/page/Tim_Berners-Lee`

Syntaxiquement juste,
sémantiquement faux

`http://www.ox.ac.uk/Profs#TimLee`



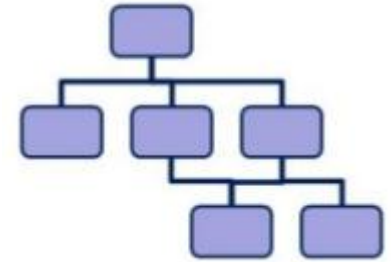
`http://www.ox.ac.uk/Vocabulary#hasName`

"Tim Berners-Lee"

RDFS va nous permettre
d'indiquer que *hasName*
n'accepte qu'un littéral comme
objet (co-domaine).

RDF Schema - RDFS

- **rdf:type** : pour distinguer le type des ressources décrites.
- Que représente réellement ?
- Comment enrichir, organiser et hiérarchiser les ressources RDF ?
- Comment donner un **sens** et une signification aux informations stockées sous forme de triplets RDF ?
- Pour le faire, un vocabulaire a été proposé. Un **vocabulaire de métadonnées**.
- **RDF Schema** – RDFS. - RDF Vocabulary Language.
- Le préfix pour tous les éléments du vocabulaire RDFS : **rdfs**

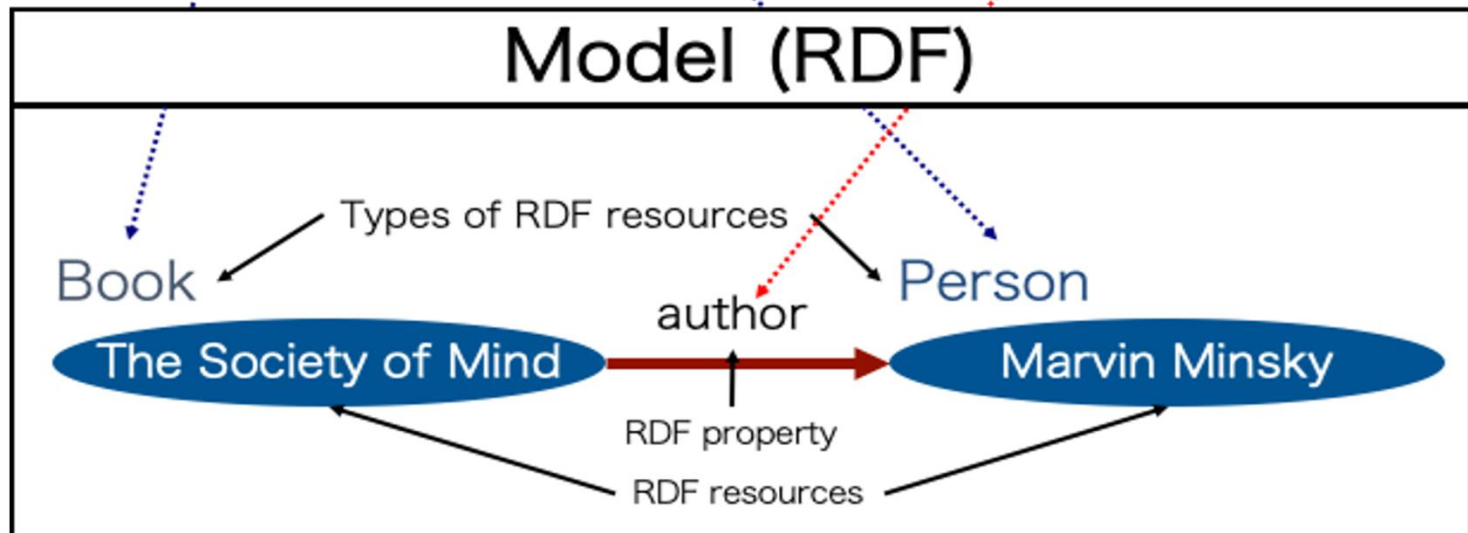


<http://www.w3.org/2000/01/rdf-schema#>

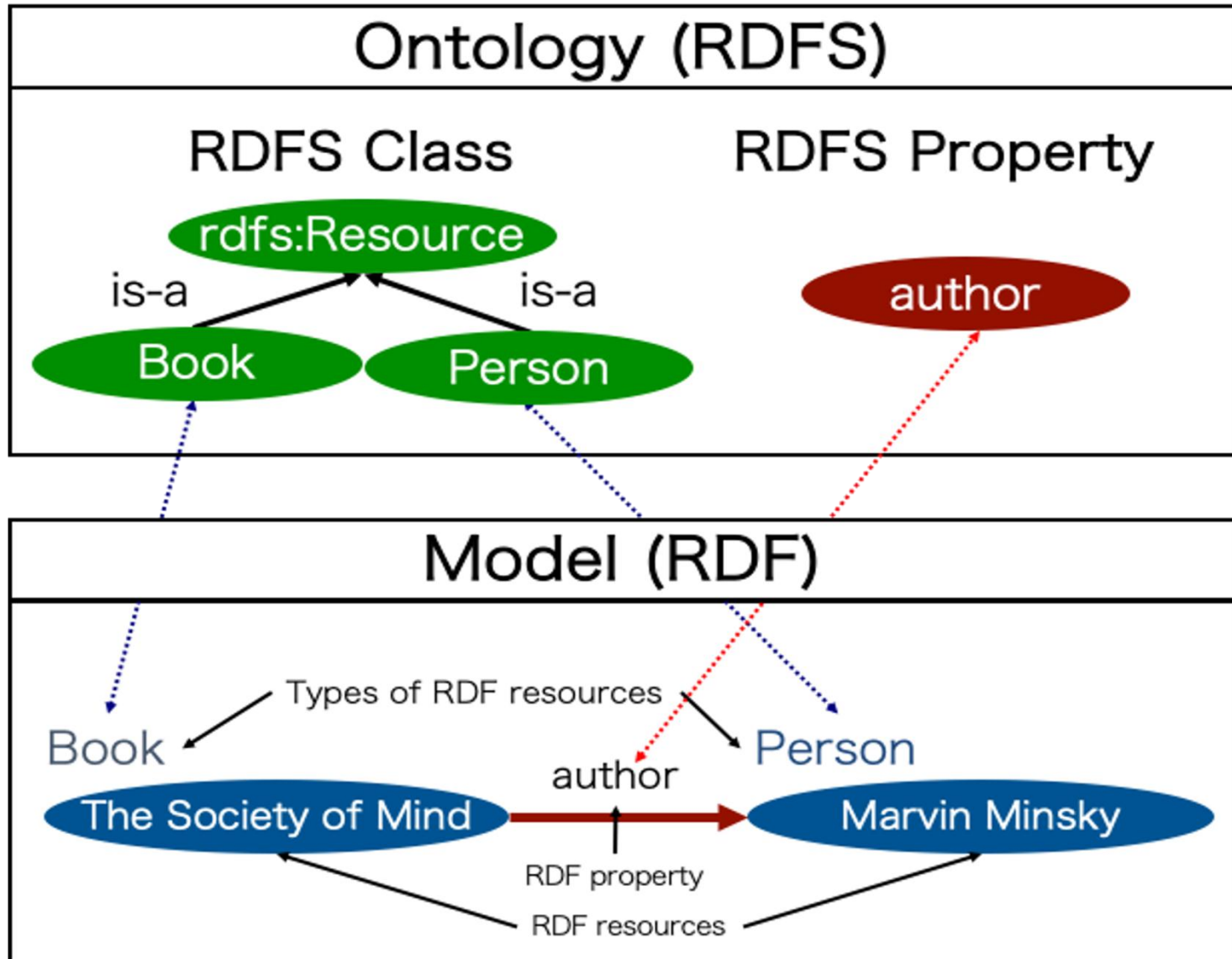
RDF Schema - RDFS

- Recommandation du W3C depuis 2004.
- Permet de définir des vocabulaires RDF, en nommant :
 - des **classes**
 - des **propriétés**
 - des **relations de sous-classe**
 - des **relations de sous-propriété**
 - le **typage des prédicats** : domaine, co-domaine (range), etc.
- Permet de définir une **organisation hiérarchique** des classes et des propriétés.
- RDFS est donc un premier langage de définition d'ontologie.
- RDFS a une expressivité réduite, permet des inférences simples (par rapport à d'autres langage de définition d'ontologie comme OWL).
- *RDFS étend RDF à la description d'ontologies (légères et simples).*

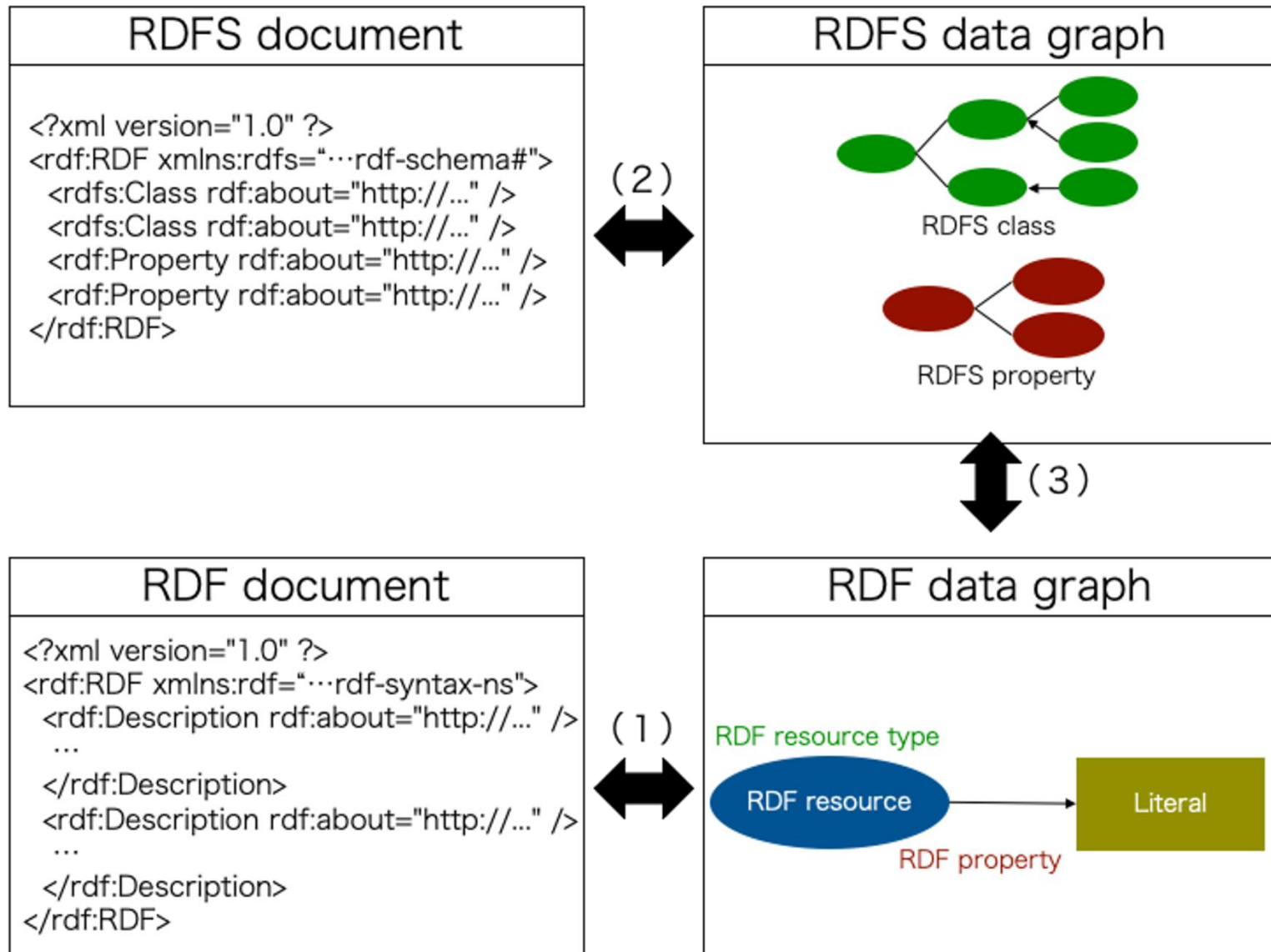
RDF Vs. RDFS



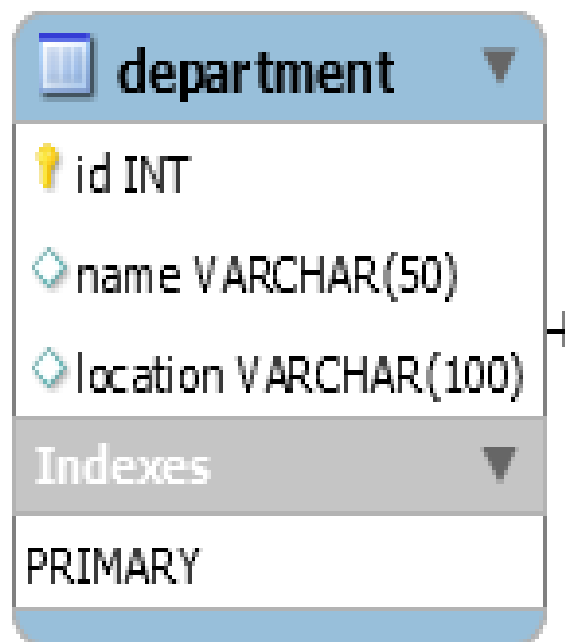
RDF Vs. RDFS



RDF Vs. RDFS



RDF Vs. RDFS

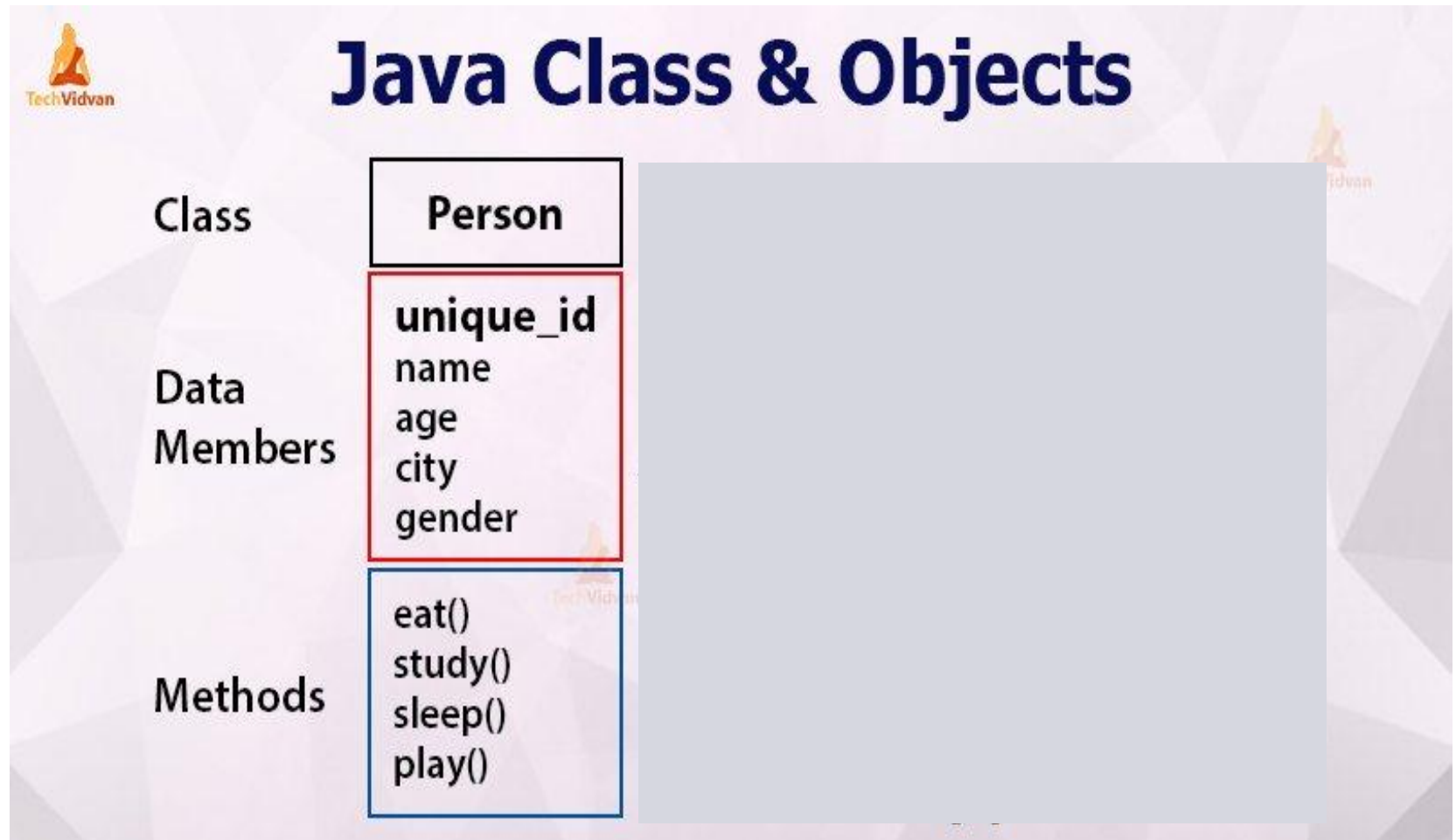


RDF Vs. RDFS

department
id INT
name VARCHAR(50)
location VARCHAR(100)
Indexes
PRIMARY

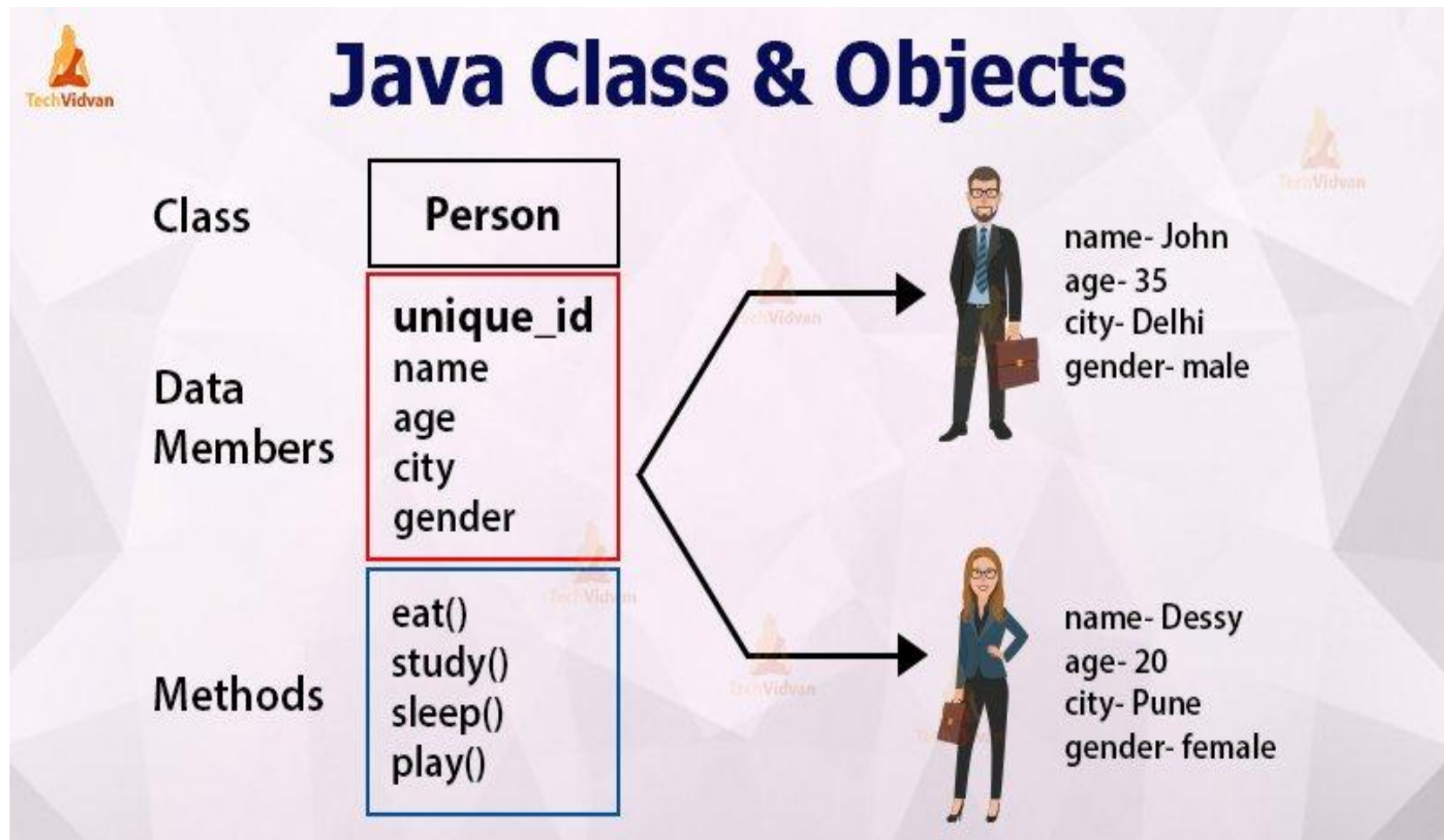
ID	name	location
001	Informatique	FSSA
002	Math	FSSA
003	Biologie	SNV

RDF Vs. RDFS



RDFS

RDF Vs. RDFS



RDFS

RDF

Définitions de classes et propriétés en RDFS

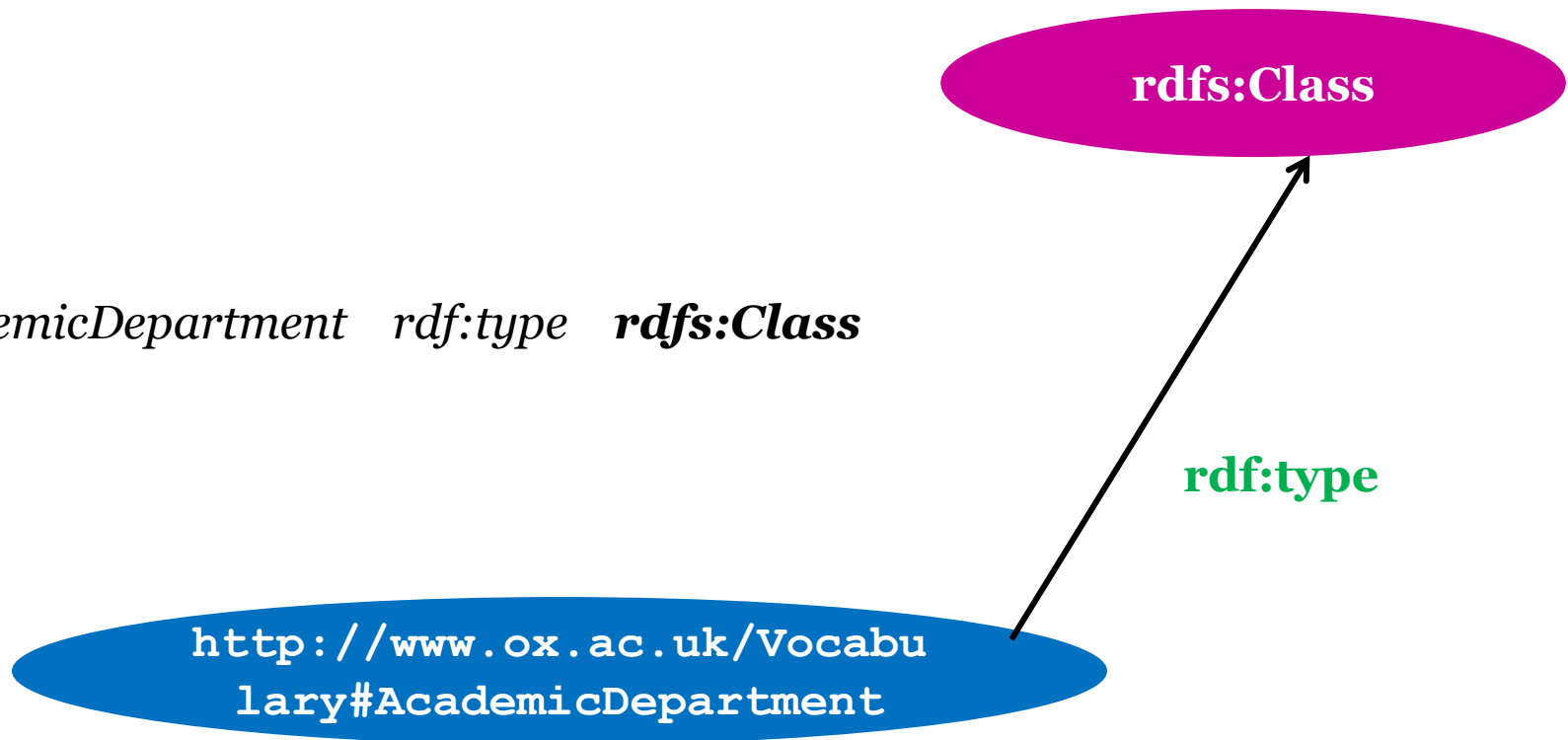
- Par rapport à la définition de classes et propriétés d'un langage de POO comme Java :
- RDFS opte pour une **approche centrée sur les propriétés** :
 - au lieu de définir des classes en donnant leurs propriétés, **on définit des propriétés en donnant leur domaine (sujet) et co-domaine (objet)**.
- Exemple :
 - en Java : « la classe eg:Document a un attribut eg:author de type eg:Person ».
 - en RDFS : « la propriété eg:author a pour domaine la classe eg:Document, et pour co-domaine la classe eg:Person ».
- Une ressource peut avoir plusieurs types.
- Une ressource peut être instance de plusieurs classes (plusieurs **rdf:type** pour une même ressource).

Définitions de classes et propriétés en RDFS

➤ Les Classes

- Les ressources peuvent être **rangées** dans des classes. Une classe est une ressource.
- Déclarations d'une ressource comme classe : **rdfs:Class**

local:AcademicDepartment *rdf:type* **rdfs:Class**



Définitions de classes et propriétés en RDFS

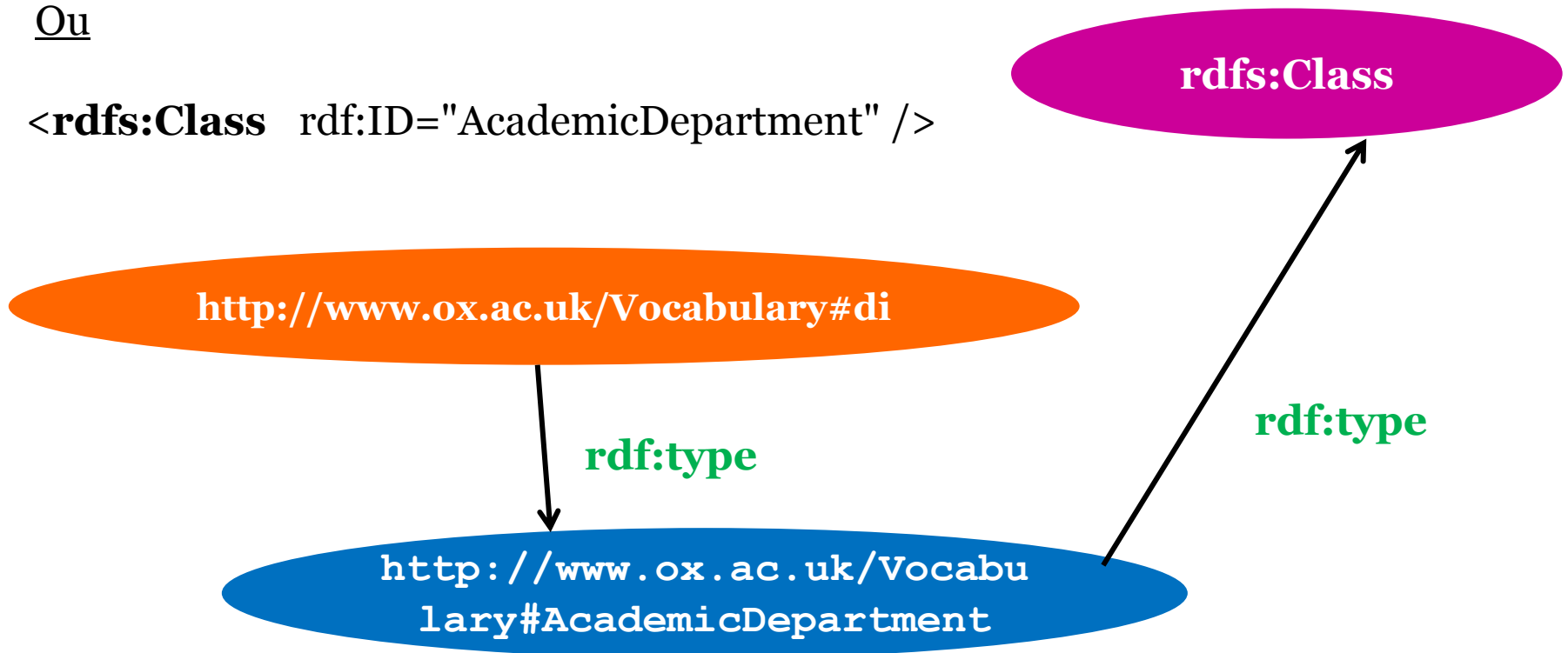
➤ Les Classes

➤ XML :

```
<rdfs:Class rdf:about="http://www.ox.ac.uk/Vocabulary#AcademicDepartment" />
```

Ou

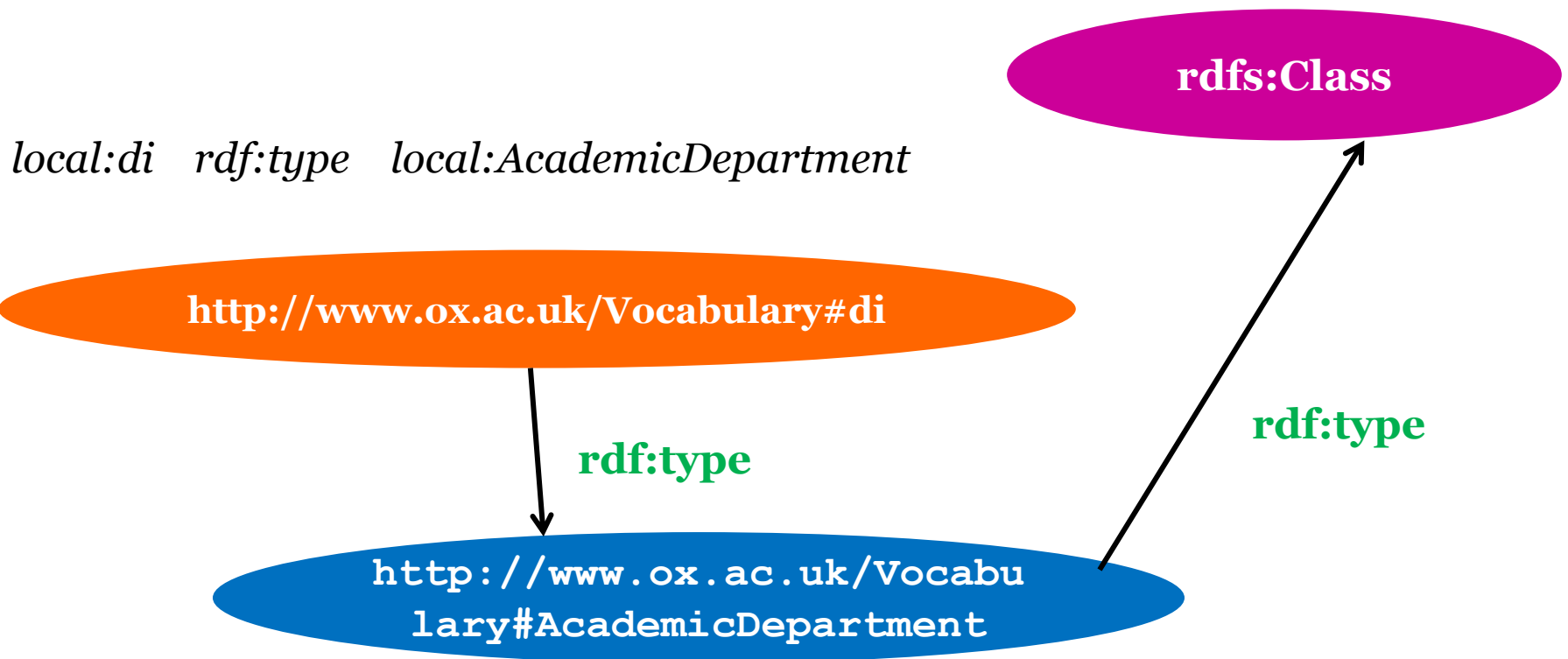
```
<rdfs:Class rdf:ID="AcademicDepartment" />
```



Définitions de classes et propriétés en RDFS

➤ Les Classes

- Les ressources peuvent être **rangées** dans des classes. Une classe est une ressource.
- Déclarations d'une ressource comme une instance de classe :

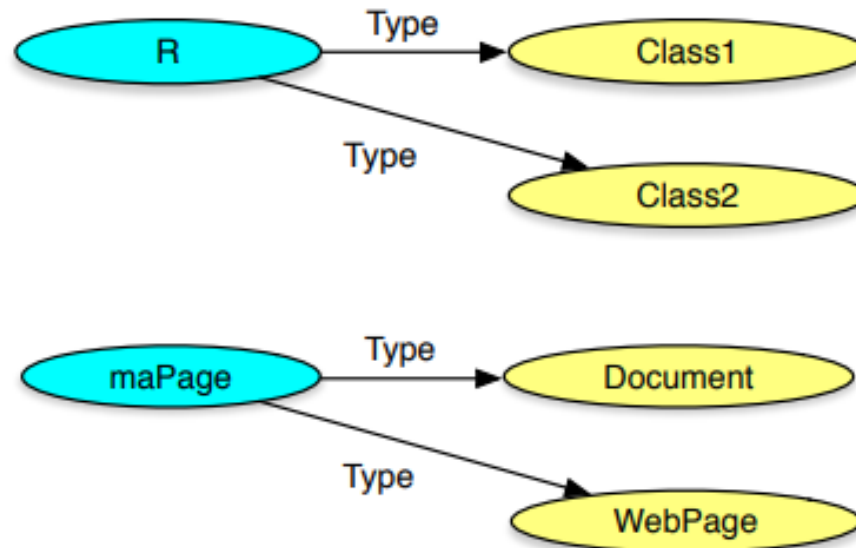


Définitions de classes et propriétés en RDFS

➤ Les Classes

➤ Multi-instanciation

- Possibilité pour une ressource d'avoir plusieurs types de classe.
- `ressource rdf:type Class1` et `ressource rdf:type Class2`

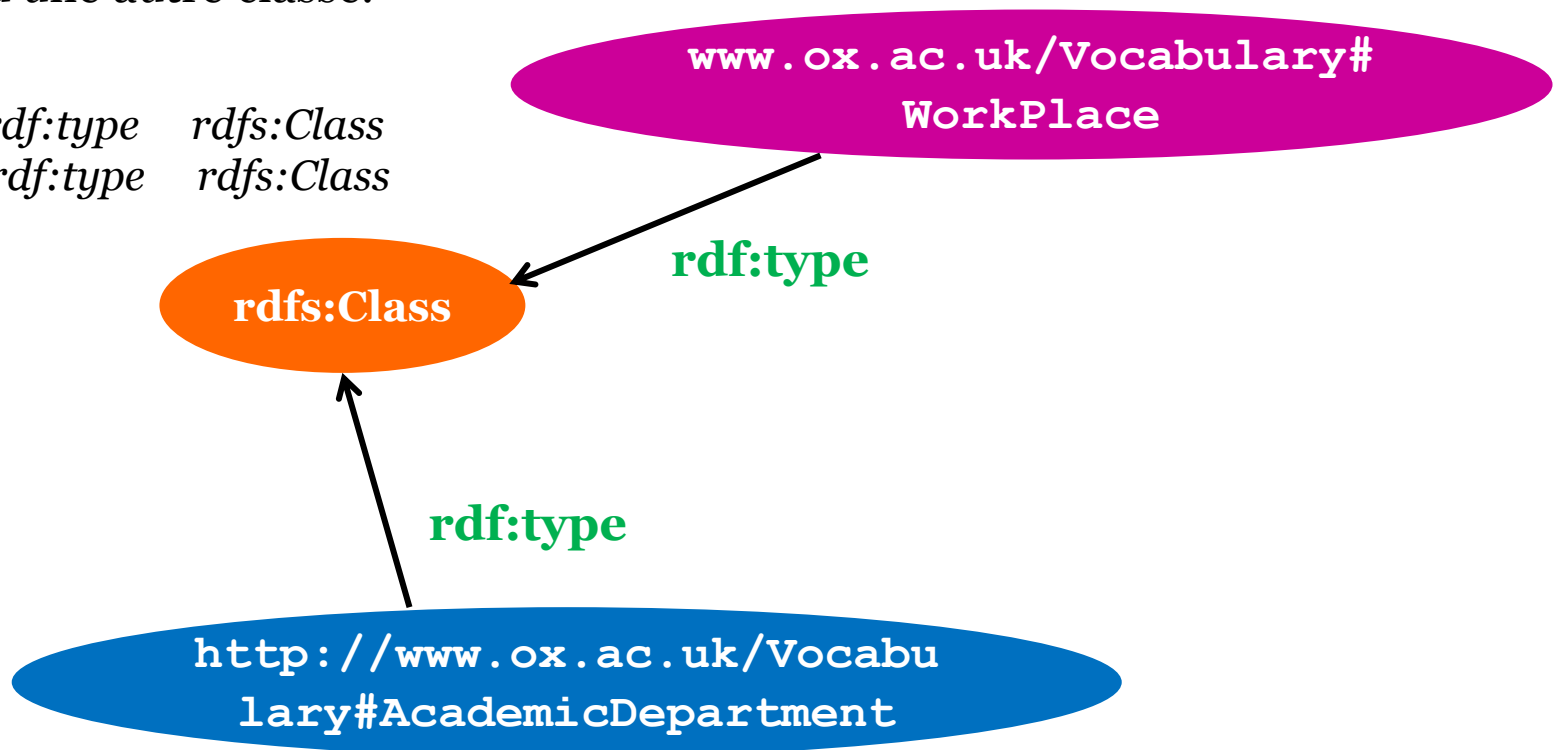


Définitions de classes et propriétés en RDFS

➤ Les Sous-Classes

- Une classe peut être sous-classe d'une ou plusieurs classes.
- On utilise la propriété **rdfs:subClassOf** pour exprimer qu'une classe est sous-classe d'une autre classe.

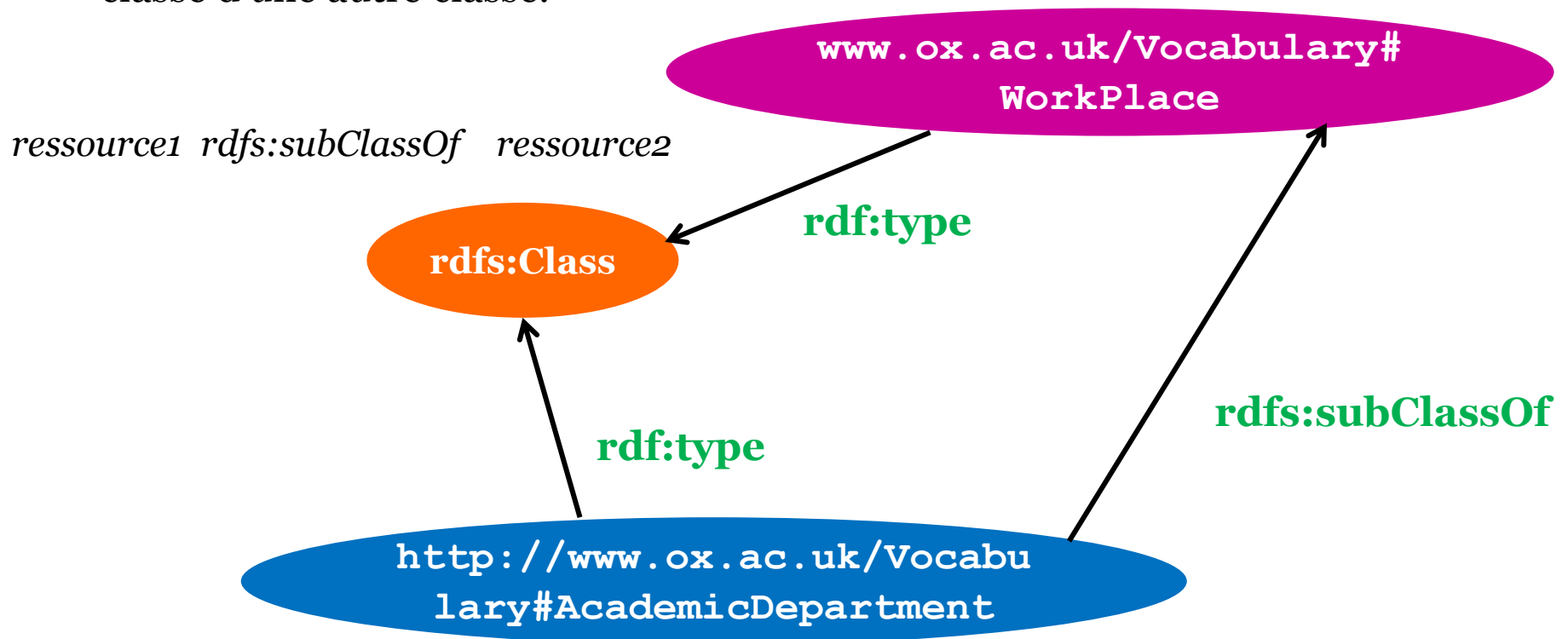
ressource1 rdf:type rdfs:Class
ressource2 rdf:type rdfs:Class



Définitions de classes et propriétés en RDFS

➤ Les Sous-Classes

- Une classe peut être sous-classe d'une ou plusieurs classes.
- On utilise la propriété **rdfs:subClassOf** pour exprimer qu'une classe est sous-classe d'une autre classe.



Définitions de classes et propriétés en RDFS

➤ Les Sous-Classes

- Une classe peut être sous-classe d'une ou plusieurs classes.
- On utilise la propriété **rdfs:subClassOf** pour exprimer qu'une classe est sous-classe d'une autre classe.

XML:

```
<rdfs:Class    rdf:ID="WorkPlace"  />
```

```
<rdfs:Class    rdf:ID="AcademicDepartment"  >
```

```
    <rdfs:subClassOf      rdf:resource="#WorkPlace"  />
```

```
</rdfs:Class>
```

Définitions de classes et propriétés en RDFS

➤ Ressources et Littéraux

- RDFS décrit des ressources, toutes instances de la classe **rdfs:Resource**, y compris rdfs:Resource et rdfs:Class.
- Toutes les primitives du langage sont des instances soit de la classe **rdfs:Class** soit de la classe **rdf:Property**.
- La classe des ressources : **rdfs:Resource**
- La classe de tous les littéraux : **rdfs:Literal**
- La classe de tous les types de données: **rdfs:Datatype**
- **rdf:XMLLiteral**: sous-classe de la classe **rdfs:Literal** et une instance de **rdfs:Datatype** : représente les valeurs codées en XML.

Définitions de classes et propriétés en RDFS

➤ Les Propriétés

- Les ressources ont un(des) type(s). Les propriétés aussi.
- Une propriété est de type : **rdf:Property**, instance de rdfs:Class.

Ex :

```
local:hasName      rdf:type      rdf:Property .
local:hasHomePage  rdf:type      rdf:Property .
local:worksAt      rdf:type      rdf:Property .

rdf:type           rdf:type      rdf:Property .
```

- Hiérarchie des propriétés : **rdfs:subPropertyOf**

Définitions de classes et propriétés en RDFS

➤ Les Propriétés

- RDFS permet de définir la signature de chaque propriété :

`rdfs:domain` **et** **`rdfs:range`**

- Une signature par :
 - ✓ Le *domain* (domaine) : type de la ressource d'où part la propriété (Sujet);
 - ✓ Le *range* (co-domaine ou portée) : type de la ressource vers laquelle pointe la propriété (Objet).

Définitions de classes et propriétés en RDFS

➤ Les Propriétés

- RDFS permet de définir la signature de chaque propriété :

rdfs:domain **et** **rdfs:range**

- Une signature par :
 - ✓ Le *domain* (domaine) : type de la ressource d'où part la propriété (Sujet);
 - ✓ Le *range* (co-domaine ou portée) : type de la ressource vers laquelle pointe la propriété (Objet).

```
<rdf:Property rdf:ID="hasName" >
```

```
  <rdfs:domain rdf:resource="...Vocabulary#StaffMember" />
```

```
  <rdfs:range rdf:resource=".../rdf-schema#Literal" />
```

```
</rdf:Property>
```

Définitions de classes et propriétés en RDFS

➤ Les Propriétés

- RDFS permet de définir la signature de chaque propriété :

rdfs:domain **et** **rdfs:range**

- Une signature par :
 - ✓ Le *domain* (domaine) : type de la ressource d'où part la propriété (Sujet);
 - ✓ Le *range* (co-domaine ou portée) : type de la ressource vers laquelle pointe la propriété (Objet).

```
<rdf:Property rdf:ID="worksAt" >
```

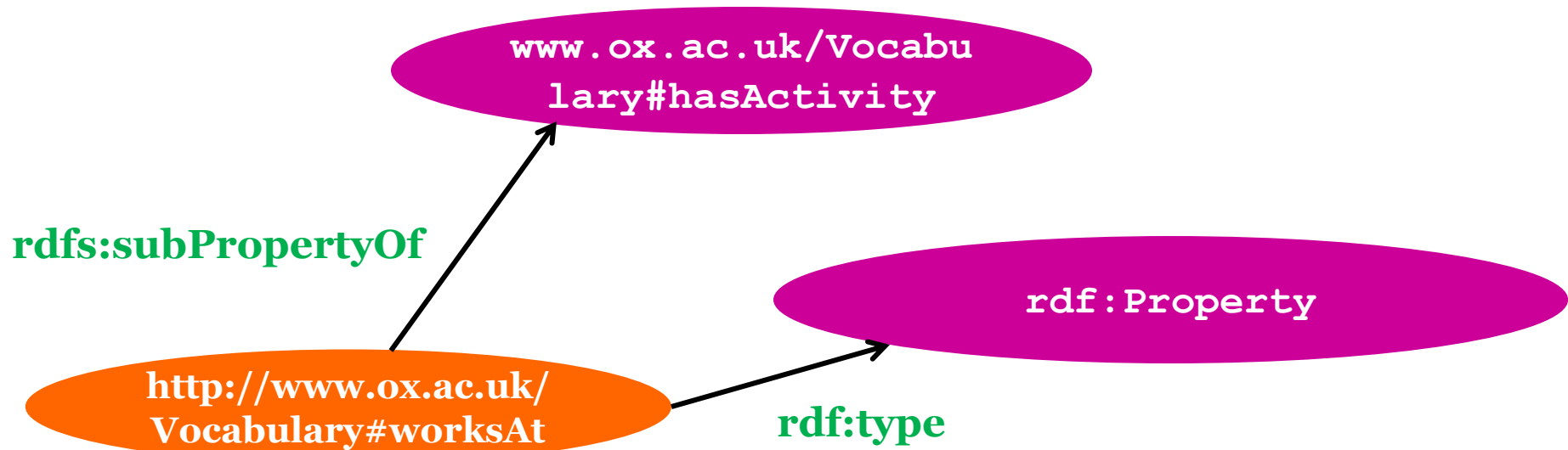
```
  <rdfs:domain  rdf:resource="...Vocabulary#StaffMember" />
```

```
  <rdfs:range   rdf:resource="...Vocabulary#WorkPlace" />
```

```
</rdf:Property>
```

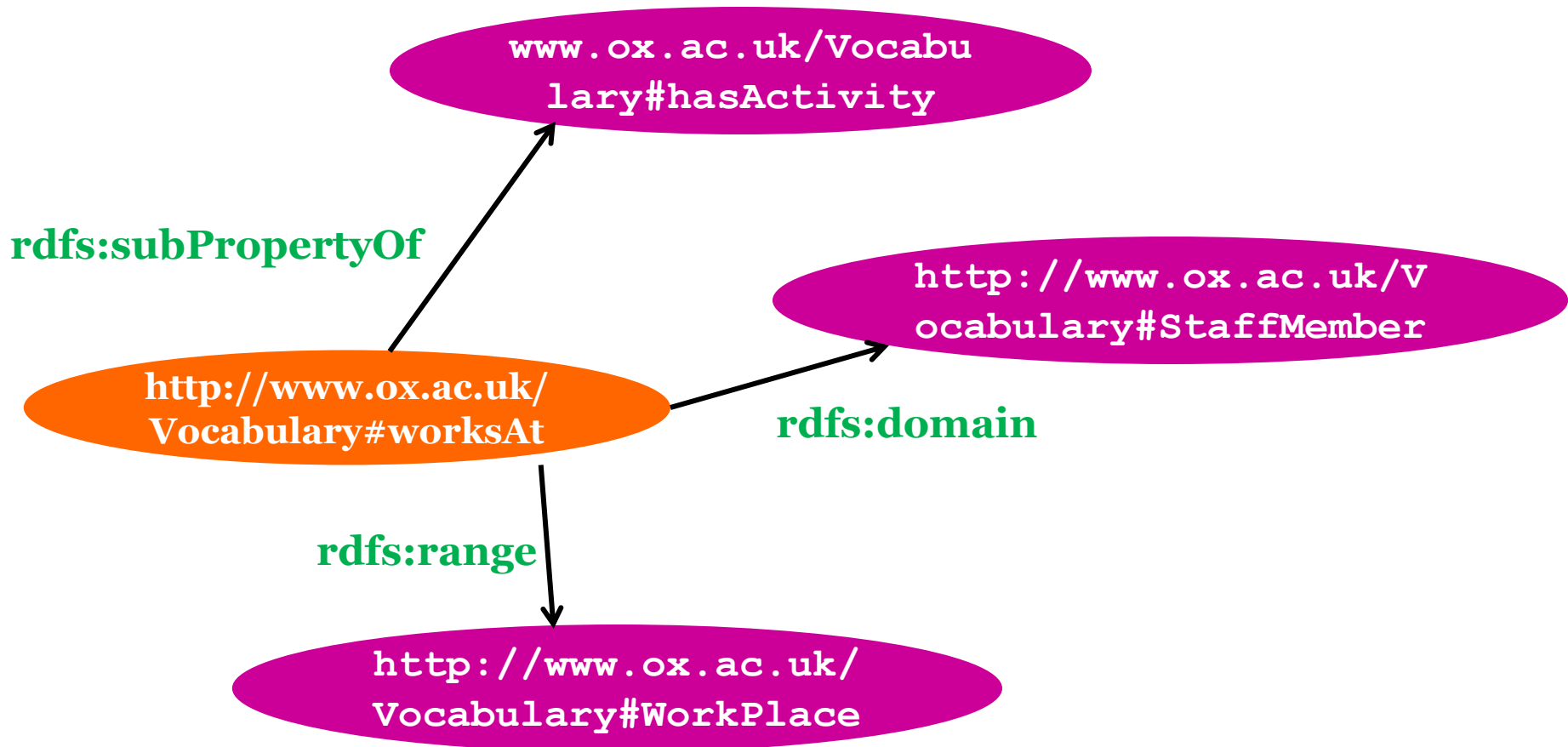
Définitions de classes et propriétés en RDFS

➤ Les Propriétés - Exemple



Définitions de classes et propriétés en RDFS

➤ Les Propriétés - Exemple



Définitions de classes et propriétés en RDFS

➤ Propriétés particulières

- RDFS permet de nommer une ressource avec du texte (en langue naturelle) compréhensible par un humain :

`rdfs:label`

- RDFS permet de commenter une ressource avec du texte (en langue naturelle) compréhensible par un humain :

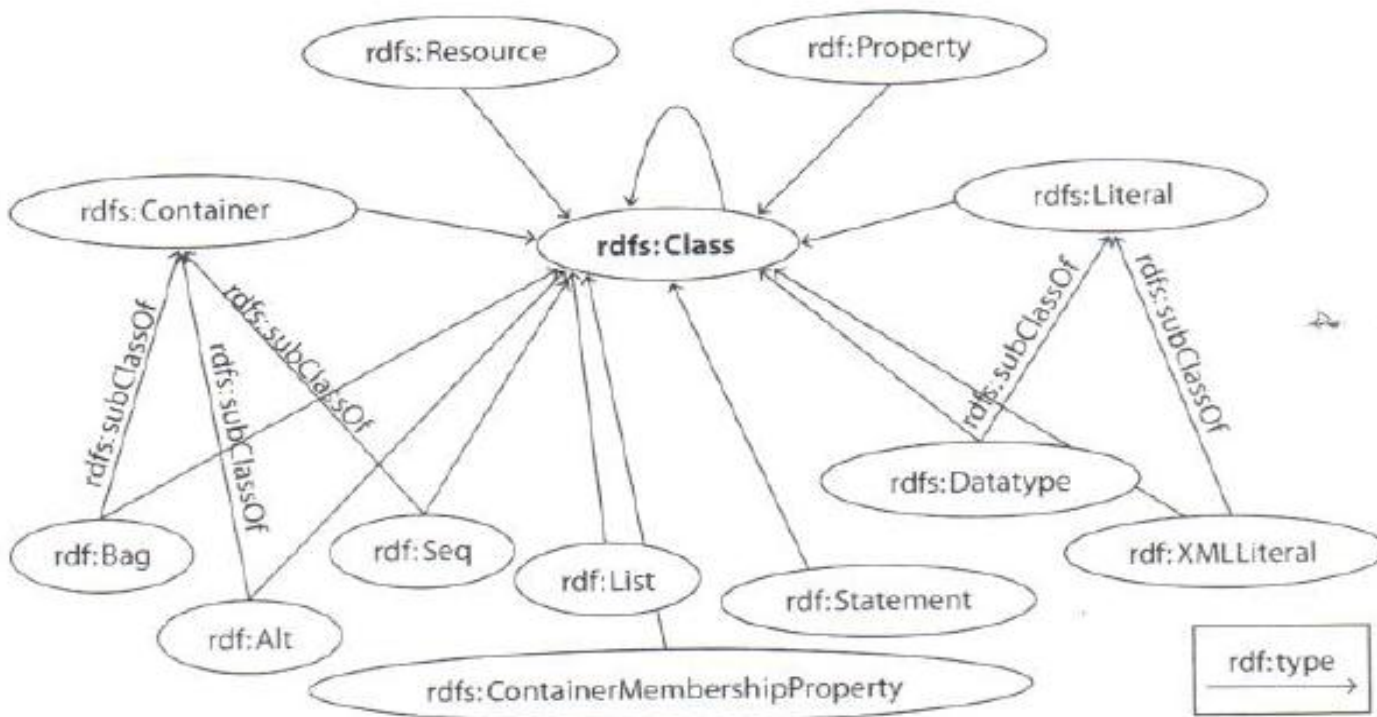
`rdfs:comment`

- RDFS permet de mentionner une ressource contenant d'autres informations sur la ressource sujet :

`rdfs:seeAlso`

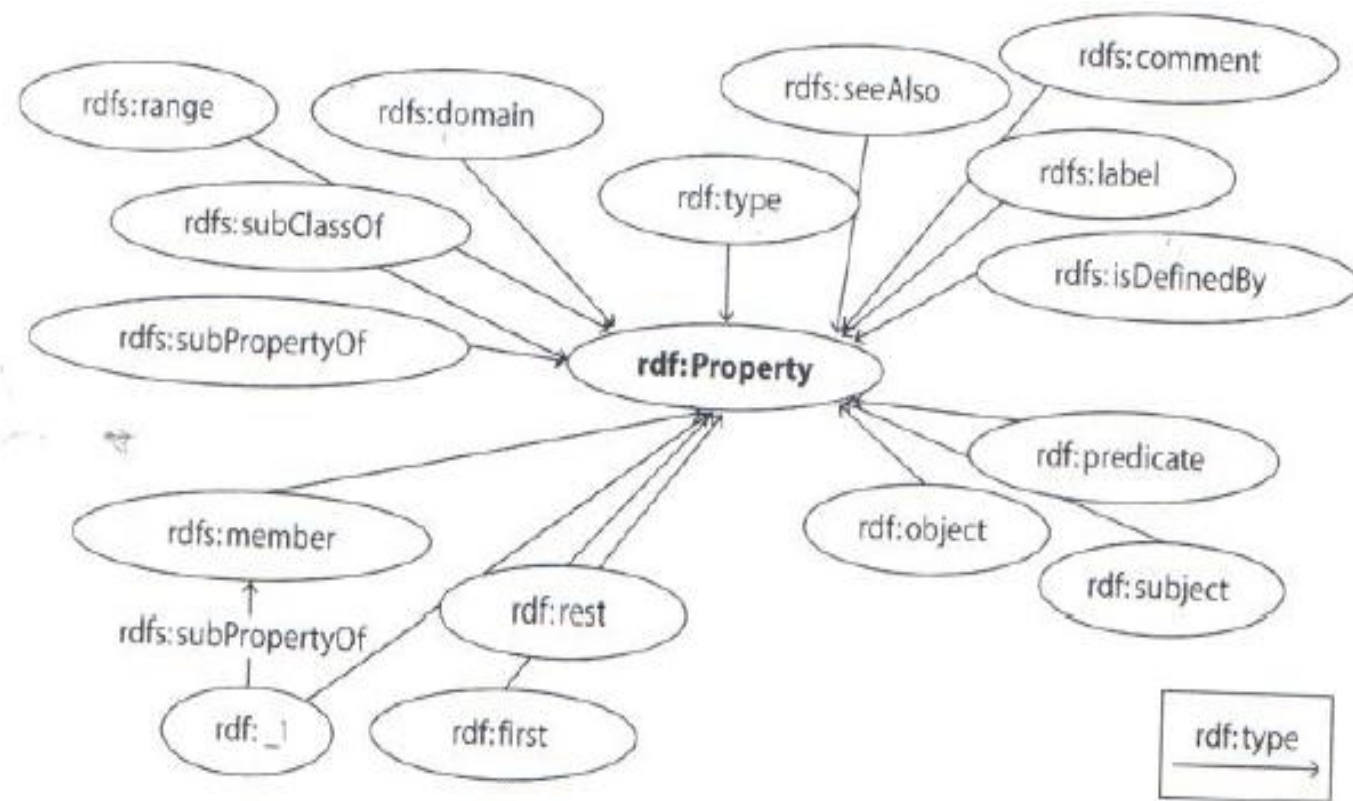
RDF Schema - RDFS

➤ Méta-modèle RDFS : déclaration de classes



RDF Schema - RDFS

➤ Méta-modèle RDFS : déclaration de propriétés



RDF Schema - RDFS

► Méta-modèle RDFS

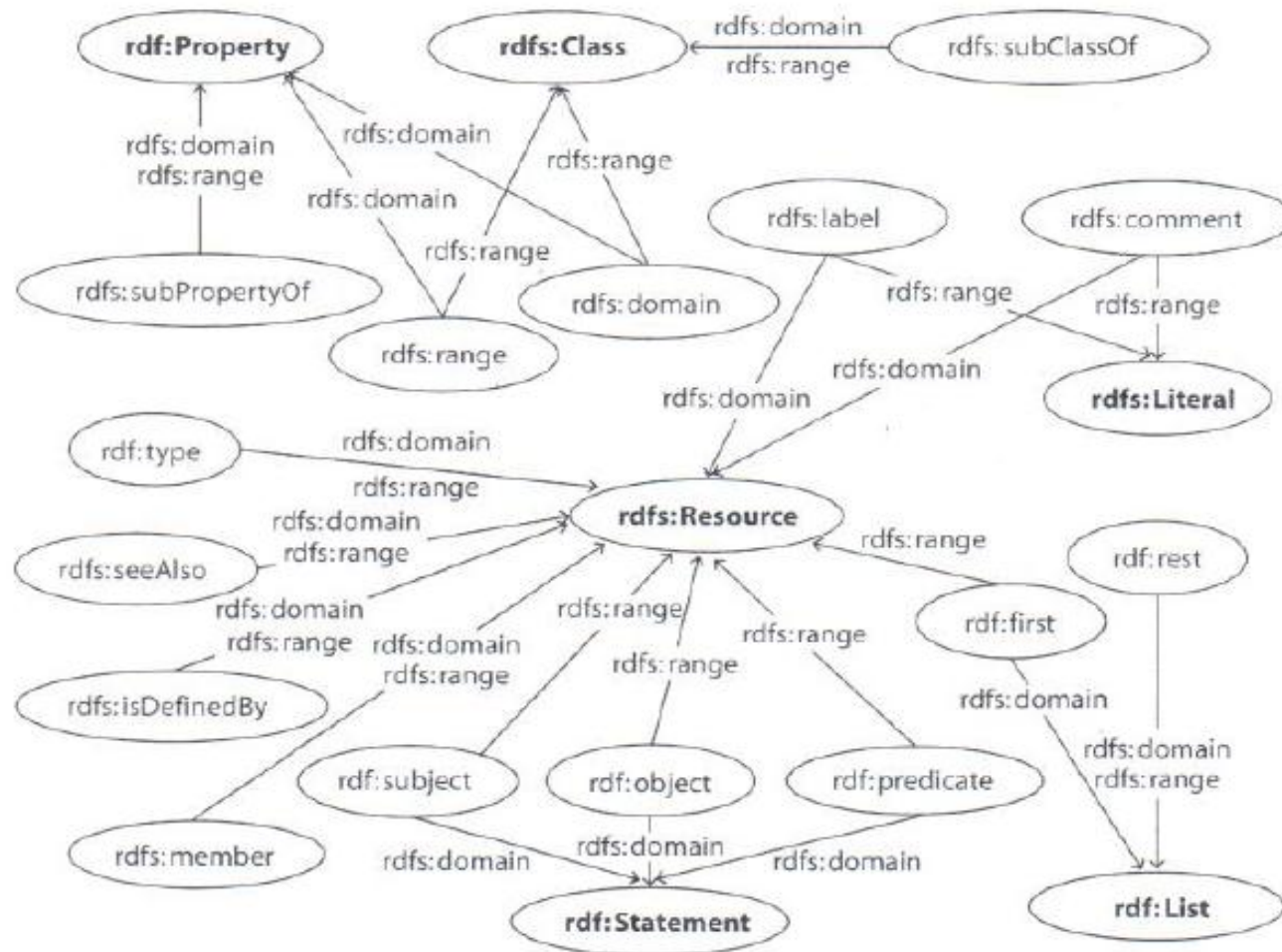


Figure: source : Le web sémantique. F. Gandon, C. Faron-Zucker, O. Corby

RDF Schema - RDFS

➤ Exemples d'un document schéma RDFS :

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">

  <rdfs:Class rdf:about="http://www.labd.org/2015/voitures/schema#Personne">
    <rdfs:comment>La classe personne</rdfs:comment>
  </rdfs:Class>

  <rdfs:Class rdf:about="http://www.labd.org/2015/voitures/schema#Vehicule">
    <rdfs:comment>La classe vehicule</rdfs:comment>
  </rdfs:Class>

  <rdfs:Class rdf:about="http://www.labd.org/2015/voitures/schema#Voiture">
    <rdfs:comment>La classe voiture</rdfs:comment>
    <rdfs:subClassOf rdf:resource="http://www.labd.org/2015/voitures/schema#Vehicule">
  </rdfs:Class>

  <rdf:Property rdf:about="http://www.labd.org/2015/voitures/schema#conducteur">
    <rdfs:range rdf:resource="http://www.labd.org/2015/voitures/schema#Personne"/>
    <rdfs:domain rdf:resource="http://www.labd.org/2015/voitures/schema#Vehicule"/>
  </rdf:Property>

</rdf:RDF>
```

RDF Schema - RDFS

➤ Exemples de schémas utilisant RDFS :

- Dublin Core (DC), qui décrit des ressources du web (e.g., images, vidéos) ou physiques. <http://dublincore.org/>
- Schema.org, un ensemble de schémas pour structurer les données du web et ses domaines populaires (e.g., personnes, restaurants, e-commerce, événements). <http://schema.org/>
- Friend of a Friend (FOAF), pour décrire des personnes, des activités et des relations. <http://www.foaf-project.org/>

Règles d'inférences

- RDFS permet **d'inférer de nouveaux triplets**, à partir de :
 - triplets existants,
 - relations de sous-classe,
 - relations sous-propriété,
 - domaines et co-domaines.

- Différents **types de règles** :
 - Règles basées sur les relations de sous-classes et de sous-propriétés
 - Règles basées sur la transitivité
 - Règles basées sur les domaines et co-domaines

Règles d'inférences

- RDFS permet de déduire des triplets additionnels à partir des triplets existants:
- Règles basées sur la **Transitivité** :

SI $(c_2, \text{subClassOf}, c_1)$
ET $(c_3, \text{subClassOf}, c_2)$
ALORS $(c_3, \text{subClassOf}, c_1)$

SI $(p_2, \text{subPropertyOf}, p_1)$
ET $(p_3, \text{subPropertyOf}, p_2)$
ALORS $(p_3, \text{subPropertyOf}, p_1)$

Règles d'inférences

- RDFS permet de déduire des triplets additionnels à partir des triplets existants:
- Règles basées sur les relations de **sous-classes et de sous-propriétés** :

SI $(c_2, \text{subClassOf}, c_1)$
ET (x, type, c_2)
ALORS (x, type, c_1)

Ex :
SI (Woman, rdfs:subClassOf, Person)
ET (Arya, rdf:type, Woman)

ALORS (Arya, rdf:type, Person)

SI $(p_2, \text{subPropertyOf}, p_1)$
ET (x, p_2, y)
ALORS (x, p_1, y)

Ex:
(author, rdfs:subPropertyOf, creator)
ET (Arya, author, Book123)

ALORS (Arya, creator, Book123)

Règles d'inférences

- RDFS permet de déduire des triplets additionnels à partir des triplets existants:
- Règles basées sur les **Range/Domain** :

SI $(p_1, \text{domain}, c_1)$
ET (x, p_1, y)
ALORS (x, type, c_1)

SI (p_1, range, c_1)
ET (x, p_1, y)
ALORS (y, type, c_1)

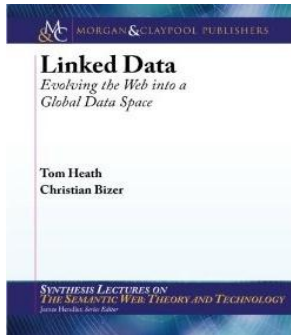
Limites de RDFS

- Puissance **expressive insuffisante**, il manque :
 - Cardinalités (min et max)
 - Décomposition (disjoint, exhaustivité)
 - Axiomes
 - Négation

- **Problèmes** dans RDF/RDFS :
 - Pas de distinction entre classes et instances
 - <Espece, type, Class>
 - <Lion, type, Espece>
 - <Simba, type, Lion>
 - Les propriétés peuvent avoir des propriétés.
 - Pas de distinction entre constructeurs du langage et les termes de l'ontologie.

- ❖ Pour dépasser ces limites, passage à aux **ontologies** et **OWL** (Ontology Web Language).

Références



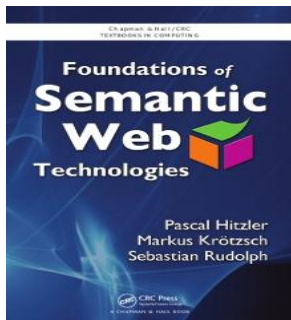
Linked Data: Evolving the Web into a Global Data Space

- ✓ 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>
- Bernard ESPINASSE– Cours - RDFS
 - ✓ <https://pageperso.lis-lab.fr/bernard.espinasse/wp-content/uploads/2021/12/3-Cours-RDFS-BE-4P.pdf>