

WEB DES DONNÉES

1. *Web des données et vers un Web de données liées*
2. *Modèle de données RDF*
3. *Schémas RDFS*
4. *Le langage de requête SPARQL*

Vers une description Sémantique

- RDF permet de définir des graphes étiquetés, en utilisant des ressources du web.
- Mais ne permet pas de définir :
 - *Les propriétés autorisées sur un type de ressources*
 - *Les valeurs autorisées pour une propriété*
 - *Les liens entre les types de ressources (généralisation/spécialisation)*
- Besoin d'un vocabulaire = modèle de données représentatif d'un ensemble de concepts dans un domaine, ainsi que des relations entre ces concepts

Vers une description Sémantique

- Comment donner le sens des « étiquettes » du graphe RDF,
- Donner la sémantique des vocabulaires permet d'améliorer l'indexation des contenus en ligne, un traitement plus efficace des requêtes, des réponses plus pertinentes, et une meilleure interopérabilité des systèmes;
- Outils: les ontologies. Elles permettent aussi un enrichissement des données en utilisant l'inférence.
- Ontologies « légères »: RDFS

Vers une description Sémantique

- La voiture1 est un véhicule; autocar1 est un autocar; les concessionnaires ont des véhicules et le concessionnaire1 a le véhicule2

- En RDF:

ex:voiture1 rdf:type ex:vehicule

ex:autocar1 rdf:type ex:autocar

ex:concess co:host ex:vehicule

ex:concessionnaire1 co:host ex:vehicule2

Graphe ne permet pas de déterminer que autocar est un véhicule.

→ Ajouter un autocar est un véhicule: ex:autocar rdfs:subClassOf ex:vehicule

RDFS (RDF Schema)

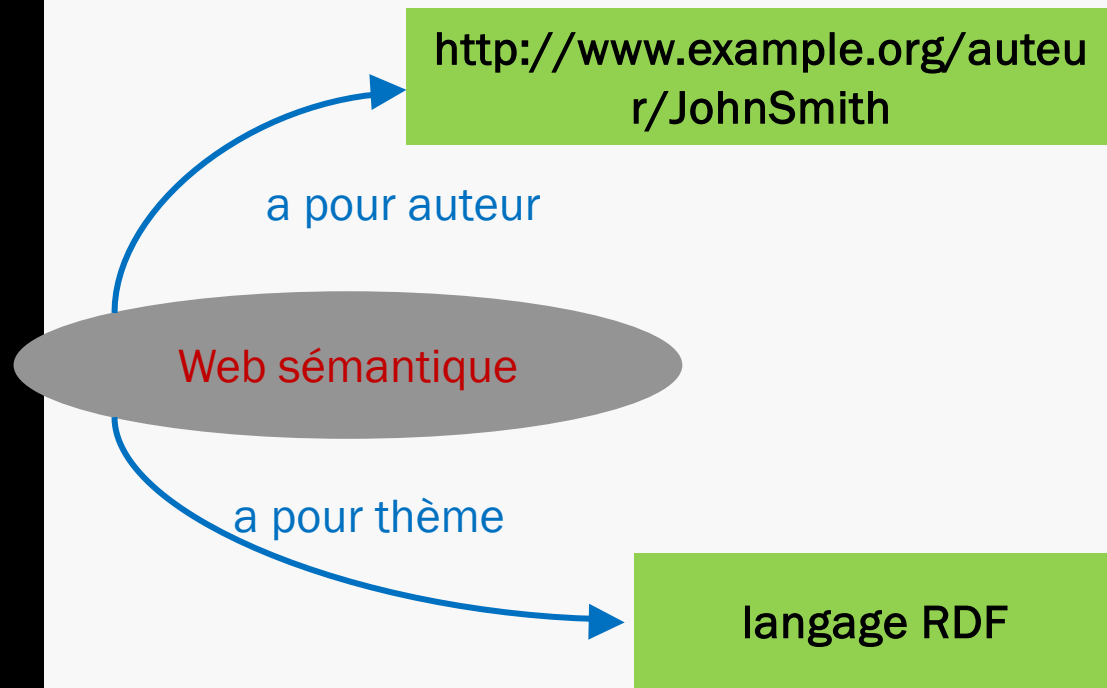
- RDF propose le langage RDF Schema qui permet de définir les caractéristiques sémantiques des données RDF:
 - *des classes (types) et des instances (ressources).*
 - *une relation de sous-classe (héritage)*
 - *Propriétés (attributs)*
 - *le typage des prédicats : domaine (types de ressources sur lesquels peut porter la propriété), co-domaine (valeurs autorisées pour la propriété).*

Mais pas de méthodes: uniquement des descriptions

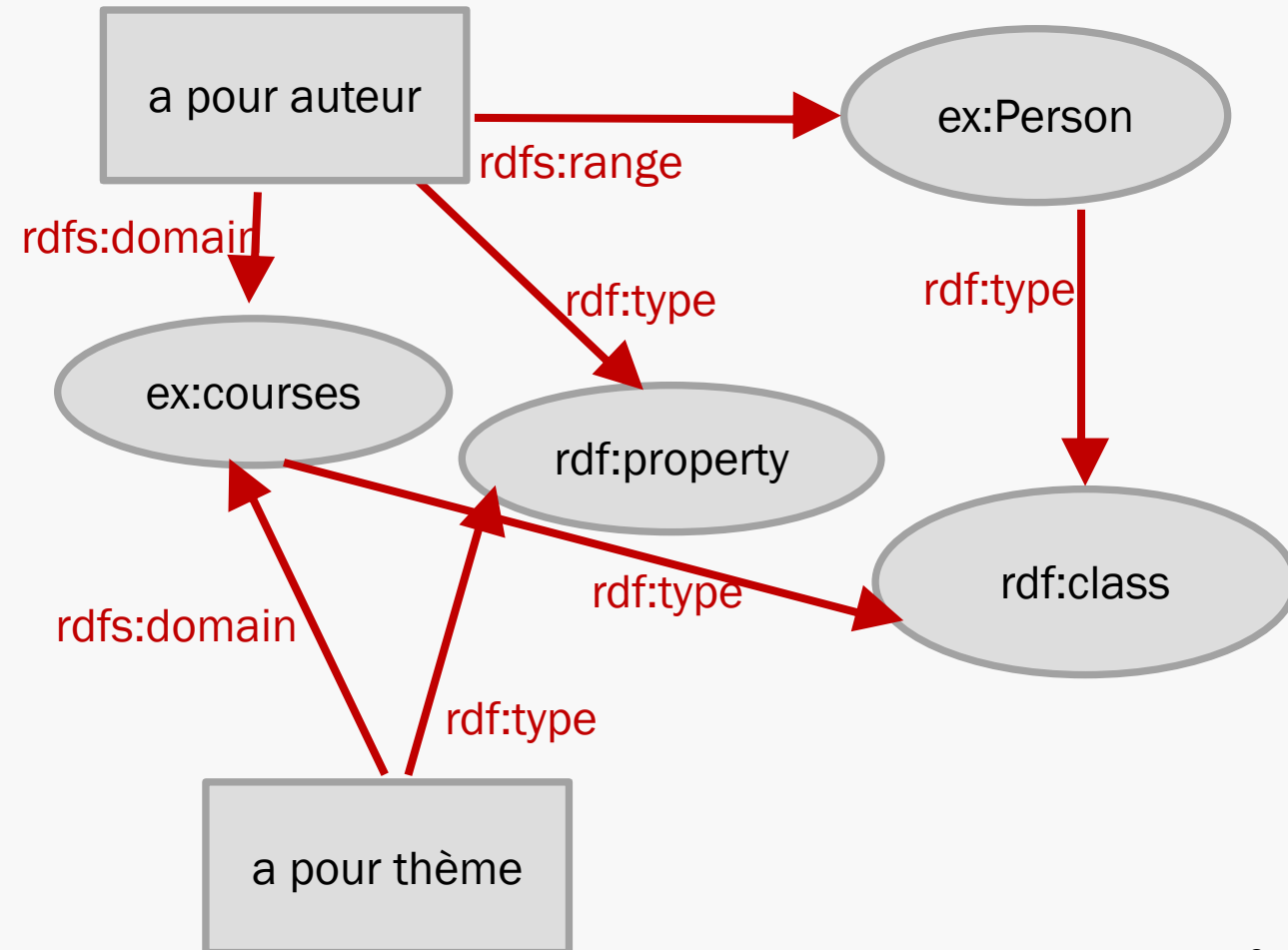
- La relation entre une instance et sa classe est déclarée via la propriété **type**.
- Une ressource peut avoir plusieurs types.
- Un RDF schema est représenté en RDF sous la forme de triplets
 - *Toute application qui gère RDF peut gérer RDF Schema*
 - *RDFS s'écrit en RDF*
 - *RDFS s'interroge en SPARQL*

RDF/RDFS

RDF



RDFS



RDFS

Exemple

- en RDFS : « la *propriété* `ex:author` a pour *domaine* la classe `ex:Courses`, et pour *co-domaine* la classe `ex:Person` »

- En Java:

La classe `ex:Courses` a un attribut `ex:author` de type `ex:Person`

RDFS

- URI de l'espace de nommage RDFS: <http://www.w3.org/2000/01/rdf-schema#>
- Comprend 15 primitives décrites dans le langage RDF et qui étendent le langage RDF.
- Primitives identifiées par des URIs : concaténation de l'URI et de l'espace de nommage RDFS: <http://www.w3.org/2000/01/rdf-schema#> et de son nom local:

Exemple: le nom qualifié `rdfs:literal` identifie la classe de tous les littéraux dont l'URI étend <http://www.w3.org/2000/01/rdf-schema#>`Literal`

Les classes

- **Classe:** type (catégorie) regroupant plusieurs instances (ressources) ayant des caractéristiques communes:
- Une classe est une ressource
 - *Une classe est identifiée par un URI.*
 - *Pour préciser qu'un URI est une classe:*
rdf:type rdfs:Class

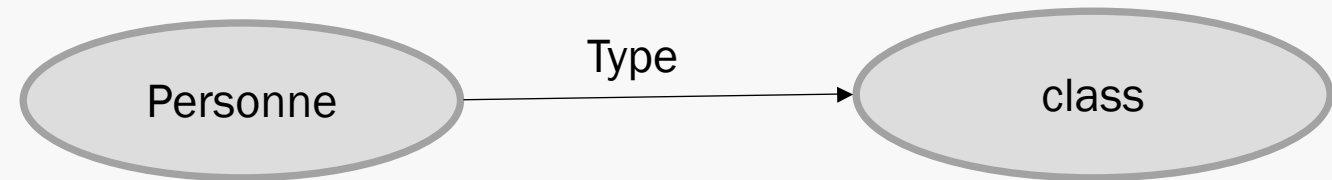
Exemple: ex:Person rdf:type rdfs:class

- *Pour préciser qu'un URI est **une instance d'une classe:***

Exemple: ex:Mathieu rdf:type ex:Person

Les classes

- Exemple: `Personne rdf:type rdfs:class`



- XML

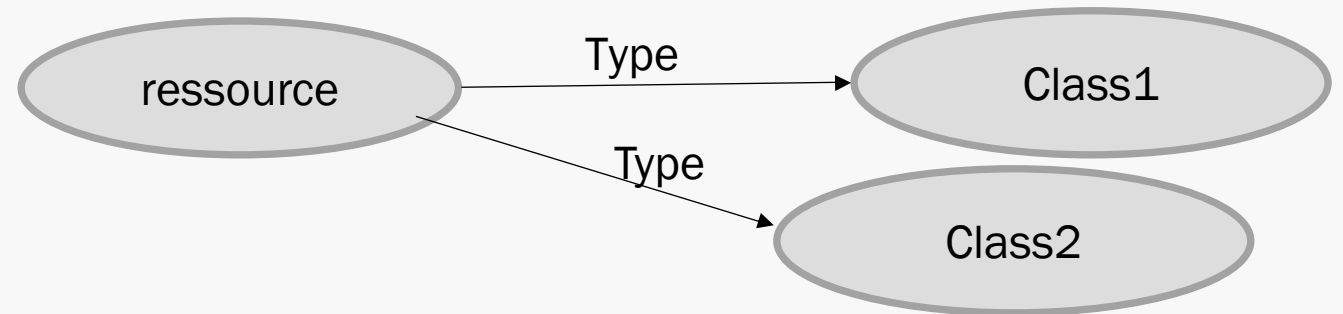
```
<rdf:Description rdf:ID="Personne">  
<rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>  
</rdf:Description>
```

- Version simplifiée XML : `<rdfs:Class rdf:ID="Personne"/>`

Les classes: multi-instanciation

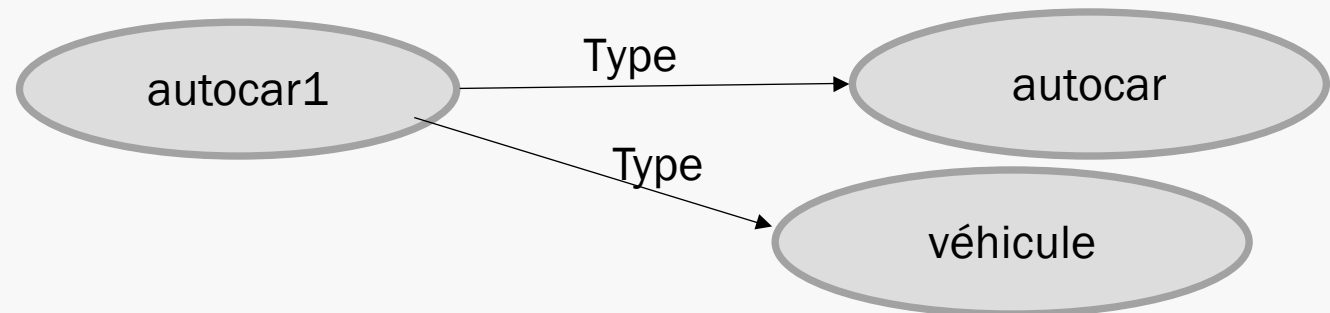
- Possibilité pour une ressource d'avoir plusieurs types de classe
- Permet de ne pas avoir à déclarer une nouvelle classe

ressource rdf:type Class1
ressource rdf:type Class2

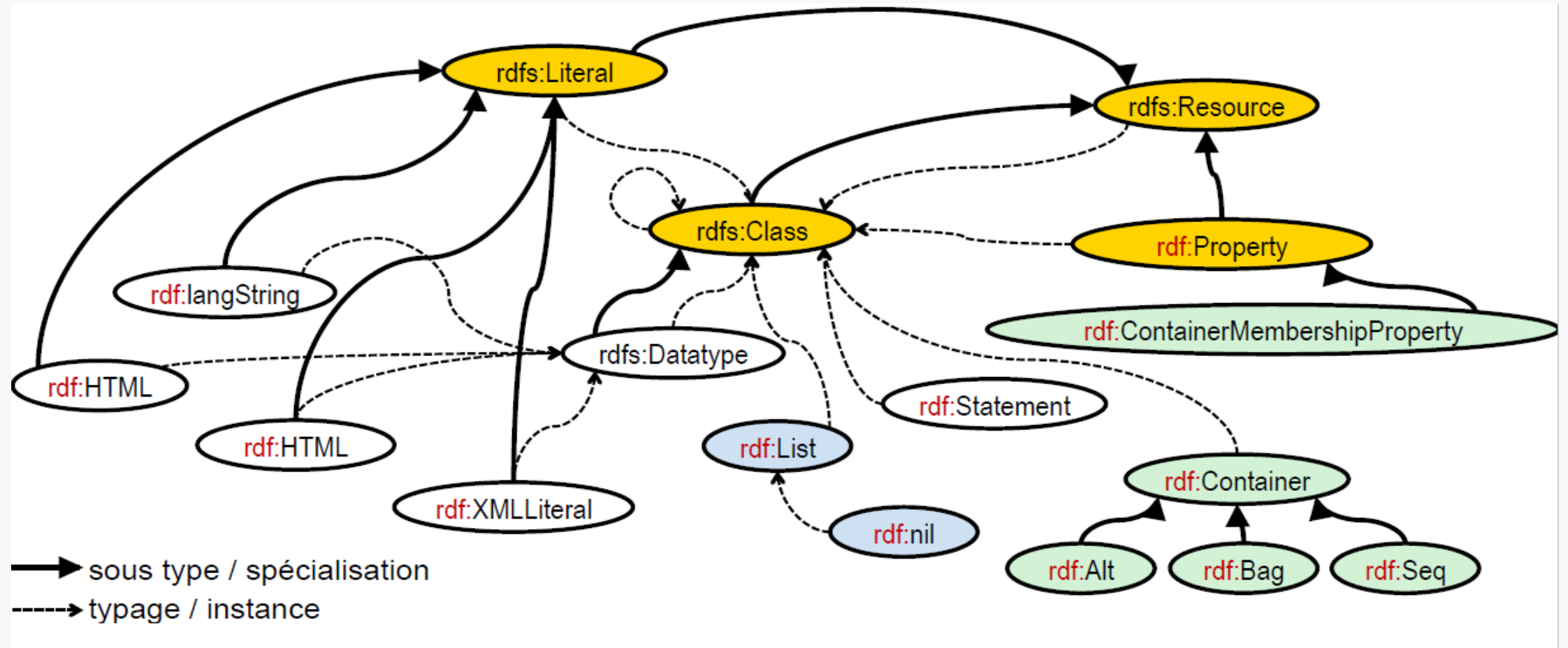


■ Exemple

(autocar1, type, autocar)
(autocar1, type, véhicule)



Graphe des classes



Quelques classes de RDF/RDFS

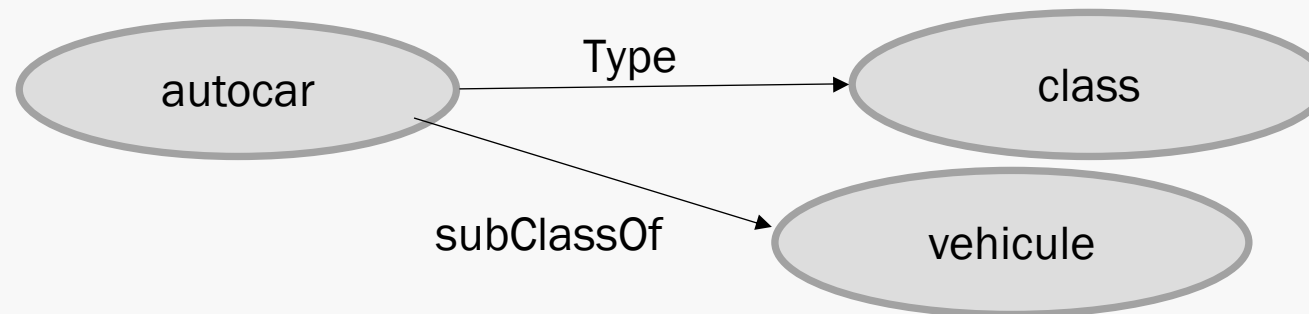
- **rdfs:Resource**: une classe pour déclarer une ressource (tout)
- **rdfs: class** déclare une ressource de type classe, qui peut être utilisée ensuite par d'autres ressources;
- **rdfs:literal**: une classe pour les littéraux (texte)
- **rdfs:Datatype** est la classe de tous les types de données
- Toute **instance** de **rdfs:Datatype** est sous-classe de **rdfs:Literal**
- **rdf:Property** une classe pour les propriétés
-

Classes RDF/RDFS

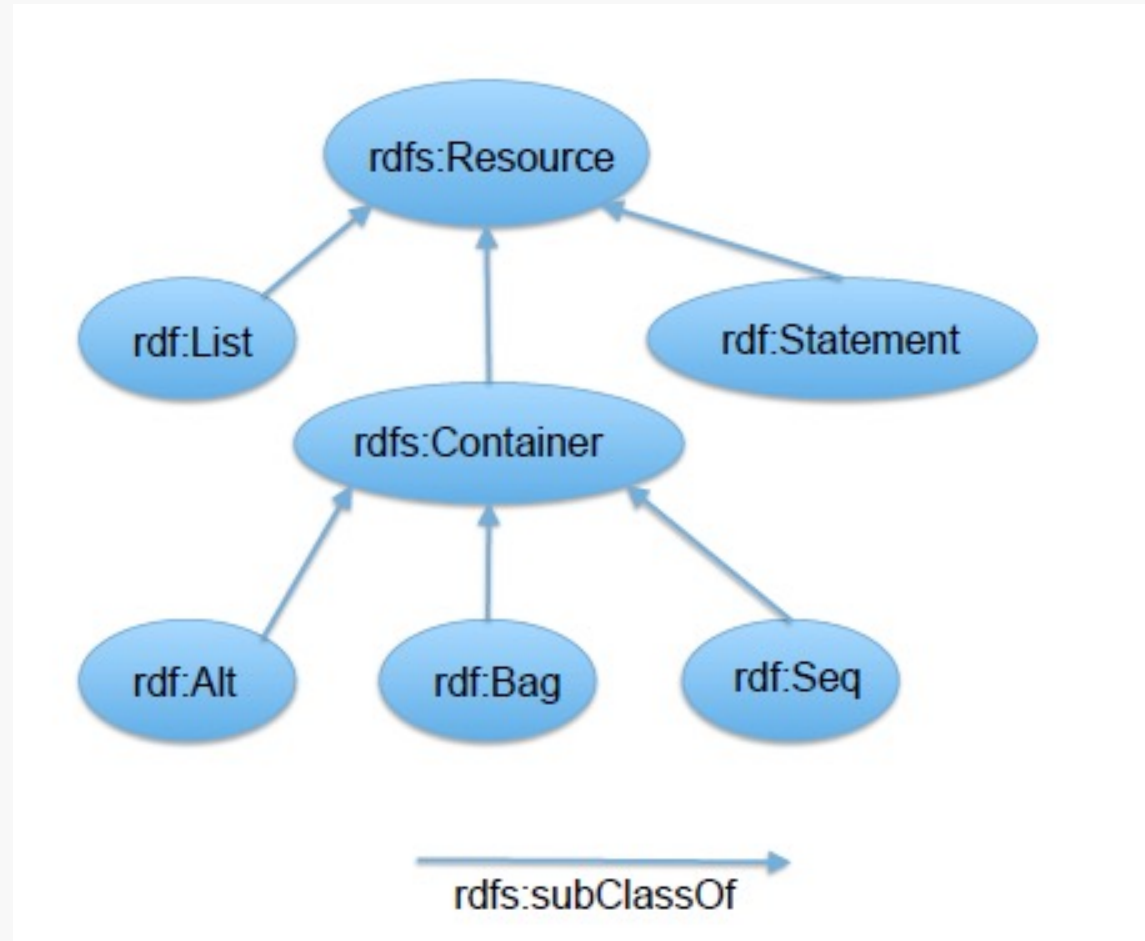
Classe	Commentaire
<code>rdfs:Resource</code>	Tout est ressource
<code>rdfs:Literal</code>	Donc du texte
<code>rdf:langString</code>	Du texte avec indication de langue (@fr)
<code>rdf:HTML</code>	Littéral HTML
<code>rdf:XMLLiteral</code>	Littéral XML
<code>rdfs:Class</code>	Une classe
<code>rdf:Property</code>	Une propriété
<code>rdfs:Datatype</code>	Un type de données
<code>rdf:Statement</code>	Un triplet
<code>rdf:Bag</code>	Un multi-ensemble
<code>rdf:Seq</code>	Une séquence
<code>rdf:Alt</code>	Une alternative
<code>rdfs:Container</code>	Un container
<code>rdfs:ContainerMembershipProperty</code>	<code>rdf:_1</code> , <code>rdf:_2</code> , ..., <code>rdfs:member</code>
<code>rdf:List</code>	Une liste (<code>rdf:nil</code> est instance de <code>rdf:List</code>)

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
 - Si C sous-classe de C' alors toutes les instances de C sont instances de C'
 - par inférence sur le Schéma
 - par transitivité
- Exemple: Un autocar est un véhicule: `ex:autocar` *rdfs:subClassOf* `ex:vehicule`



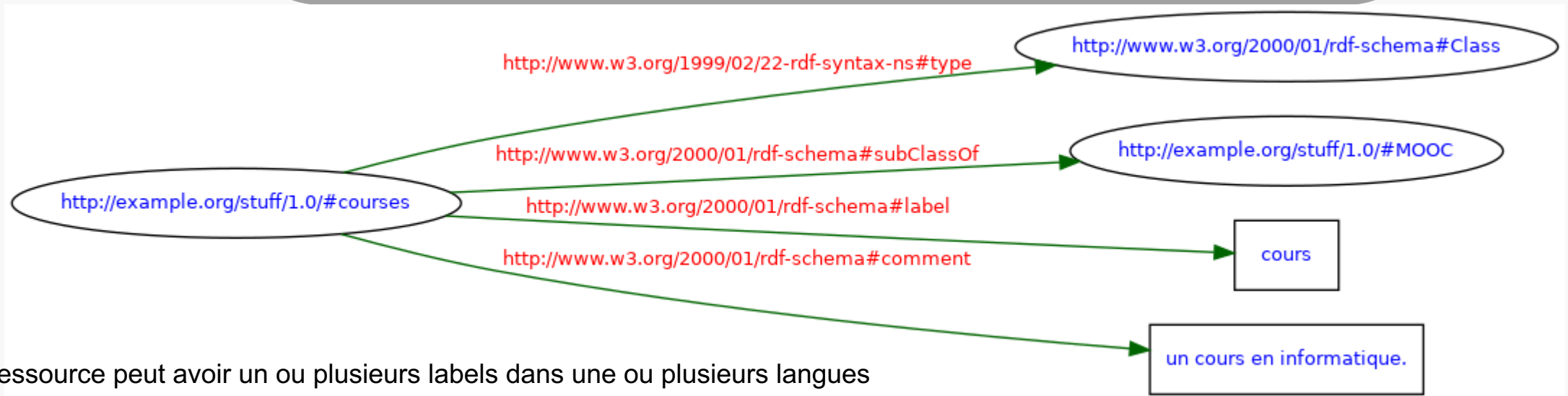
Les sous-classes: rdfs:subClassOf



Les classes

■ Exemple

```
<rdf:RDF xml:base="http://example.org/stuff/1.0/"  
  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:ID="courses">  
  
    <rdfs:subClassOf rdf:resource="#MOOC"/>  
  
    <rdfs:label xml:lang='fr'>cours</rdfs:label>  
    <rdfs:comment>un cours en informatique.</rdfs:comment>  
  </rdfs:Class>  
</rdf:RDF>
```



Toute ressource peut avoir un ou plusieurs labels dans une ou plusieurs langues

Les propriétés

- Est identifiée par une URI
- Pour préciser qu'un URI est une propriété: il faut écrire que cette ressource a pour *rdf:type rdf:Property*

Exemple : ex:vehicule rdf:type rdfs:Class

ex:aPourMarque rdf:type rdf:Property

- La relation de sous-propriété entre 2 propriétés est définie par *rdfs:subPropertyOf*.

Les propriétés

- On peut définir le type du sujet (**domaine**) et/ou de l'objet (**co-domaine**) d'une propriété :

- ***rdfs:domain** : domaine d'une propriété*
- ***rdfs:range** : co-domaine d'une propriété*

- Exemple:

ex:habite rdfs:domain ex:Personne

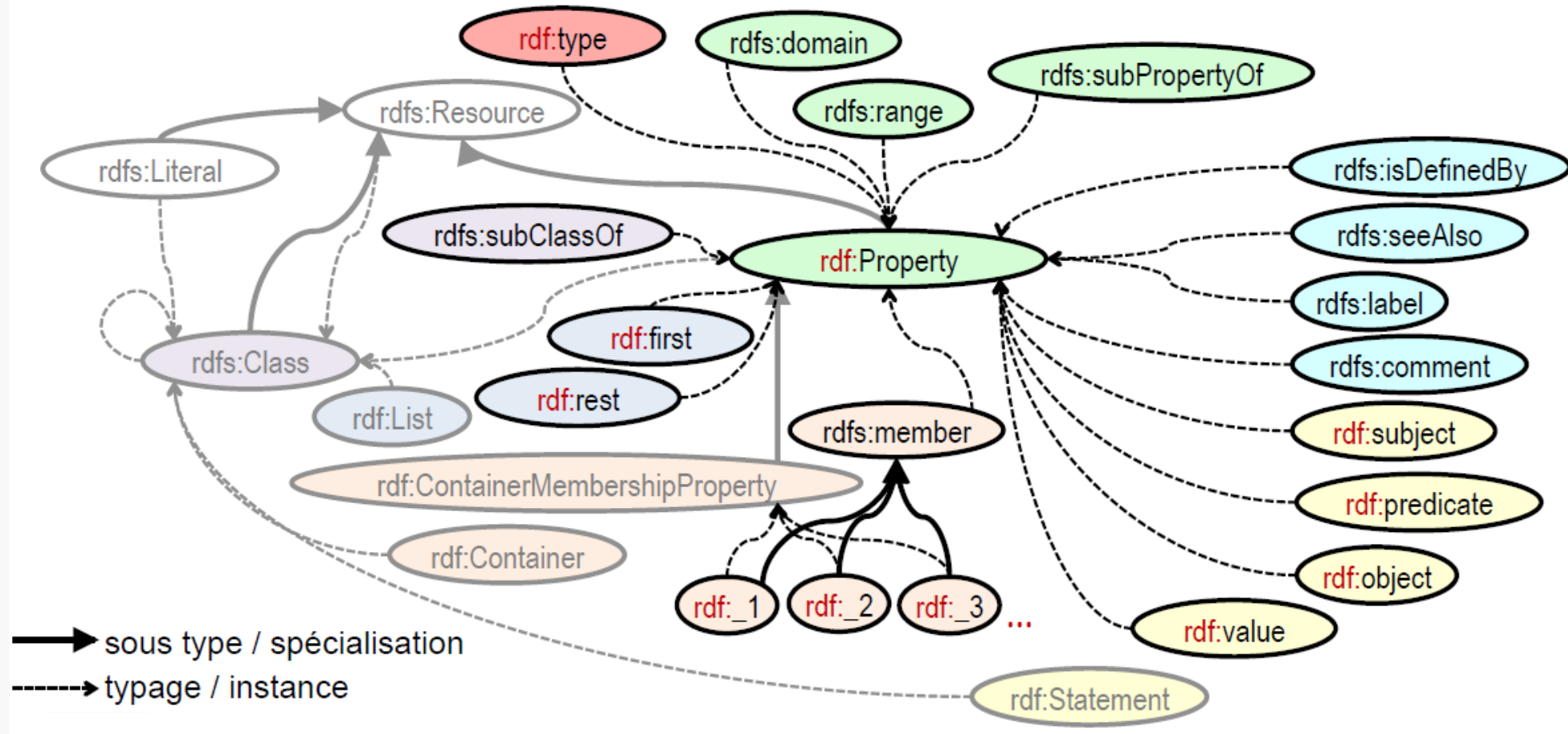
ex:habite rdfs:range ex:Ville

- Le domaine ou co-domaine d'une propriété peut contenir plusieurs classes.

ex:habite rdfs:range ex:Departement

- Une **propriété** peut avoir **plusieurs domaines** et **plusieurs co-domaines**

Graphe des propriétés



Quelques propriétés de RDF/RDFS

- **rdf:type:** indique qu'une ressource sujet est une instance d'une classe donnée
- **rdfs:domain:** permet de déclarer le domaine, c'est-à-dire les classes dans lesquelles une propriété prend ses sujets.
- **rdfs:range** déclare la portée c'est-à-dire les classes dans lesquelles une propriété prend ses valeurs.
- **rdfs:subClassOf** pour hiérarchiser les classes
-

Propriétés RDF/RDFS

Propriété	Domaine	Co-domaine
<code>rdf:type</code>	<code>rdfs:Resource</code>	<code>rdfs:Class</code>
<code>rdfs:subClassOf</code>	<code>rdfs:Class</code>	<code>rdfs:Class</code>
<code>rdfs:subPropertyOf</code>	<code>rdf:Property</code>	<code>rdf:Property</code>
<code>rdfs:domain</code>	<code>rdf:Property</code>	<code>rdfs:Class</code>
<code>rdfs:range</code>	<code>rdf:Property</code>	<code>rdfs:Class</code>
<code>rdfs:label</code>	<code>rdfs:Resource</code>	<code>rdfs:Literal</code>
<code>rdfs:comment</code>	<code>rdfs:Resource</code>	<code>rdfs:Literal</code>
<code>rdfs:member</code>	<code>rdfs:Resource</code>	<code>rdfs:Resource</code>
<code>rdf:first</code>	<code>rdf:List</code>	<code>rdfs:Resource</code>
<code>rdf:rest</code>	<code>rdf:List</code>	<code>rdf:List</code>
<code>rdfs:seeAlso</code>	<code>rdfs:Resource</code>	<code>rdfs:Resource</code>
<code>rdfs:isDefinedBy</code>	<code>rdfs:Resource</code>	<code>rdfs:Resource</code>
<code>rdf:value</code>	<code>rdfs:Resource</code>	<code>rdfs:Resource</code>
<code>rdf:subject</code>	<code>rdfs:Statement</code>	<code>rdfs:Resource</code>
<code>rdf:predicate</code>	<code>rdfs:Statement</code>	<code>rdfs:Resource</code>
<code>rdf:object</code>	<code>rdfs:Statement</code>	<code>rdfs:Resource</code>

À ces propriétés, il faut ajouter `rdf:_1`, `rdf:_2`, ... qui sont toutes sous-propriétés de `rdfs:member` et instances de `rdfs:ContainerMembershipProperty`.

Les propriétés

la classe des propriétés est placée dans l'espace de nom RDF car les triplets sont une construction de RDF.

■ Exemple

```
<rdf:RDF xml:base="http://example.org/stuff/1.0/"  
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"  
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
```

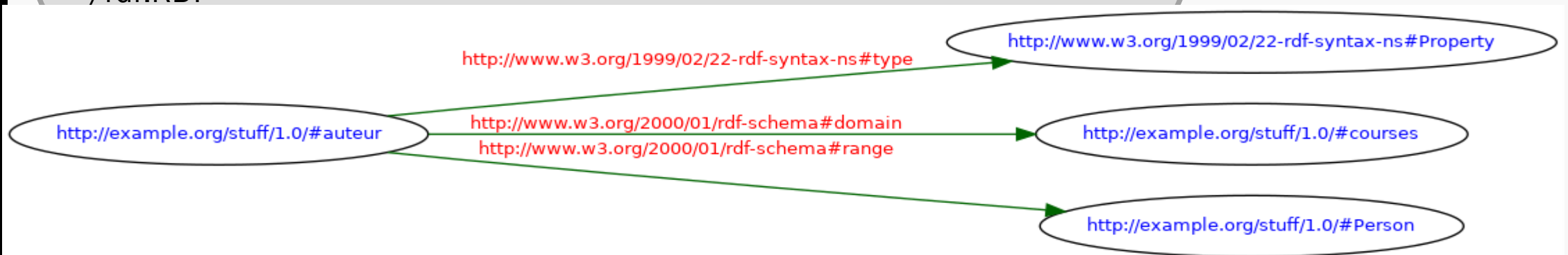
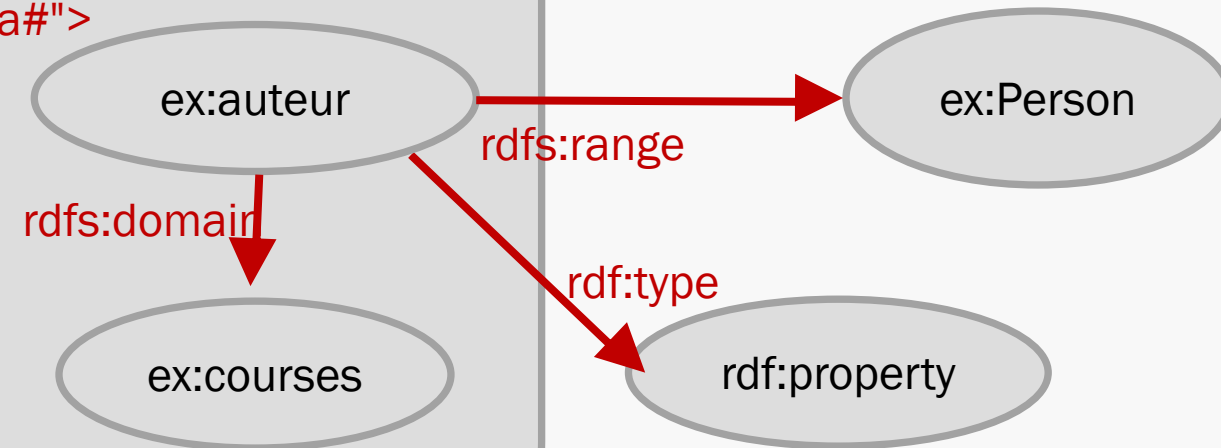
```
<rdf:Property rdf:ID="auteur">
```

```
<rdfs:domain rdf:resource="#courses"/>
```

```
<rdfs:range rdf:resource="#Person"/>
```

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

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



Quelques propriétés de RDF/RDFS

■ Exemple:

la propriété « *travailledans* » relie un sujet qui est une personne à un objet qui est une organisation:

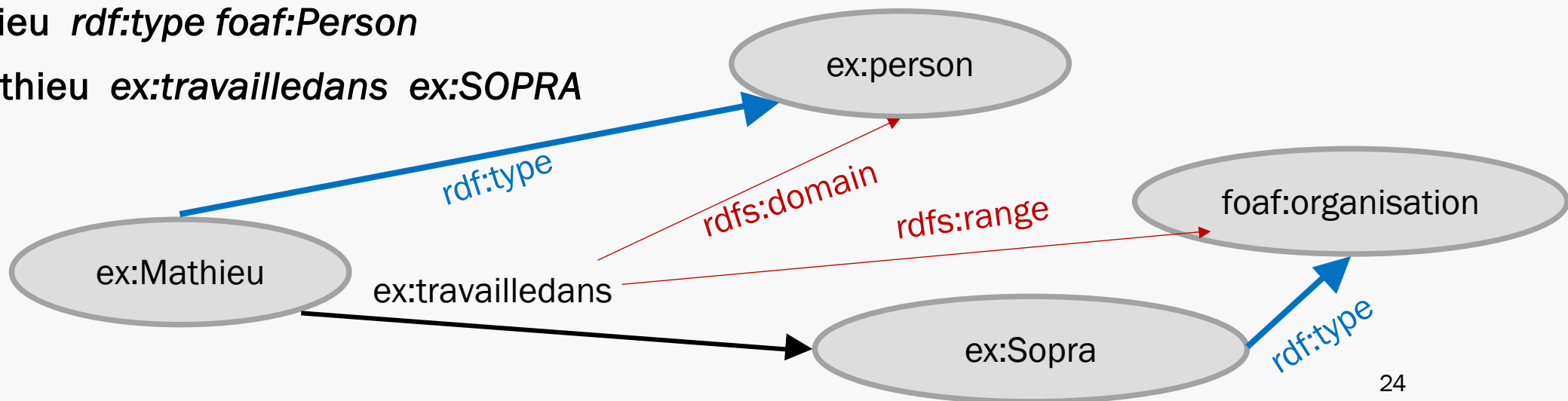
ex:travailledans rdfs:domain foaf:Person

ex:travailledans rdfs:range foaf:Organisation

À partir des déclarations précédentes, la déclaration suivante implique nécessairement que **ex:Mathieu** est une instance de *foaf:Person*, et **ex:SOPRA** une instance de *foaf:Organisation*

ex:Mathieu rdf:type foaf:Person

→ *ex:Mathieu ex:travailledans ex:SOPRA*



Inférence

- 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*

Inférence

Règles basées sur les relations de sous-classes et sous-propriétés

- R1: Si x de type C et $C \subseteq C'$ alors x de type C'

$$\frac{(x, \text{rdf:type}, C) \wedge (C, \text{rdfs:subclassOf}, C')}{(x, \text{rdf:type}, C')}$$

- *Exemple*

(Man, *rdfs:subclassOf*, Person) et (Mathieu, *rdf:type*, Man)
Alors (Mathieu, *rdf:type*, Person)

- R2: Si p1 (x,y) et p1 sous-propriété de p2 Alors p2 (x,y)

$$\frac{(x, p1, y) \wedge (p1, \text{rdfs:subPropertyOf}, p2)}{(x, p2, y)}$$

- *Exemple*

(author, *rdfs:subPropertyOf*, creator) et (John, *author*, WebSemantique)
Alors (John, *creator*, WebSemantique)

Inférence

Règles basées sur les transitivités

- Transitivité des relations de sous-classe:
 - *R3: Si C sous-classe de C' ET C' sous-classe de C''*
Alors C est sous- classe de C''

- Transitivité des relations de sous-propriété:
 - *R4 : Si p sous-propriété de p' ET p' sous-propriété de p''*
Alors p est sous- propriété de p''

Inférence

Règles basées sur les domaines et co-domaines

■ R5:
$$\frac{(x, p, y) \wedge (p, \text{rdfs : domain}, C)}{(x, \text{rdf:type}, C)}$$

Si p propriété de domaine C **ET** p(x,y) **ALORS** x de type C

■ R6:
$$\frac{(x, p, y) \wedge (p, \text{rdfs : range}, C)}{(y, \text{rdf:type}, C)}$$

Si p propriété de co-domaine C **ET** p(x,y) **ALORS** y de type C

D'autres règles : <https://www.w3.org/TR/rdf11-mt/#entailment-rules-informative>

Inférence

■ Exercice: Dédire les inférences faites par RDFS

1. c:creator rdfs:domain c:Person
 ex:John c:creator ex:WebSémantique
 ex:John rdf:type c:Person
2. c:author rdfs:subPropertyOf c:creator
 c:author rdfs:range c:Document
 ex:Mathieu c:author ex:WebDonnees
 ex:WebDonnees rdf:type c:Document
 ex:Mathieu c:creator ex:WebDonnees
 ex:Mathieu rdf:type c:creator

Exemples de schémas/ontologies utilisant RDFS

- Simple Knowledge Organization System (SKOS), un standard W3C pour décrire un thésaurus
- Dublin Core (DC) qui décrit des ressources du Web (e.g images, vidéos)
- Schema.org un ensemble de schémas pour structurer les données du Web et ses domaines populaires (e.g personnes, restaurant, e-commerce, événements)
- Friend of a friend (FOAF): une ontologie pour décrire des personnes, des activités et des relations

Ce qu'il faut retenir de RDFS

- Déclarer et décrire les types de ressources manipulées (appelées classes, ex. les livres, les films, les personnes etc.) et les types de relation entre ces ressources (appelées propriétés, ex : a pour auteur, a pour acteur, a pour titre, etc.)
- → Définir des vocabulaires utilisés dans les graphes RDF et d'en nommer les primitives avec des URI
- **Objectif:** automatiser des déductions (ex. déduire qu'une ressource du type Roman est aussi du type Livre).