

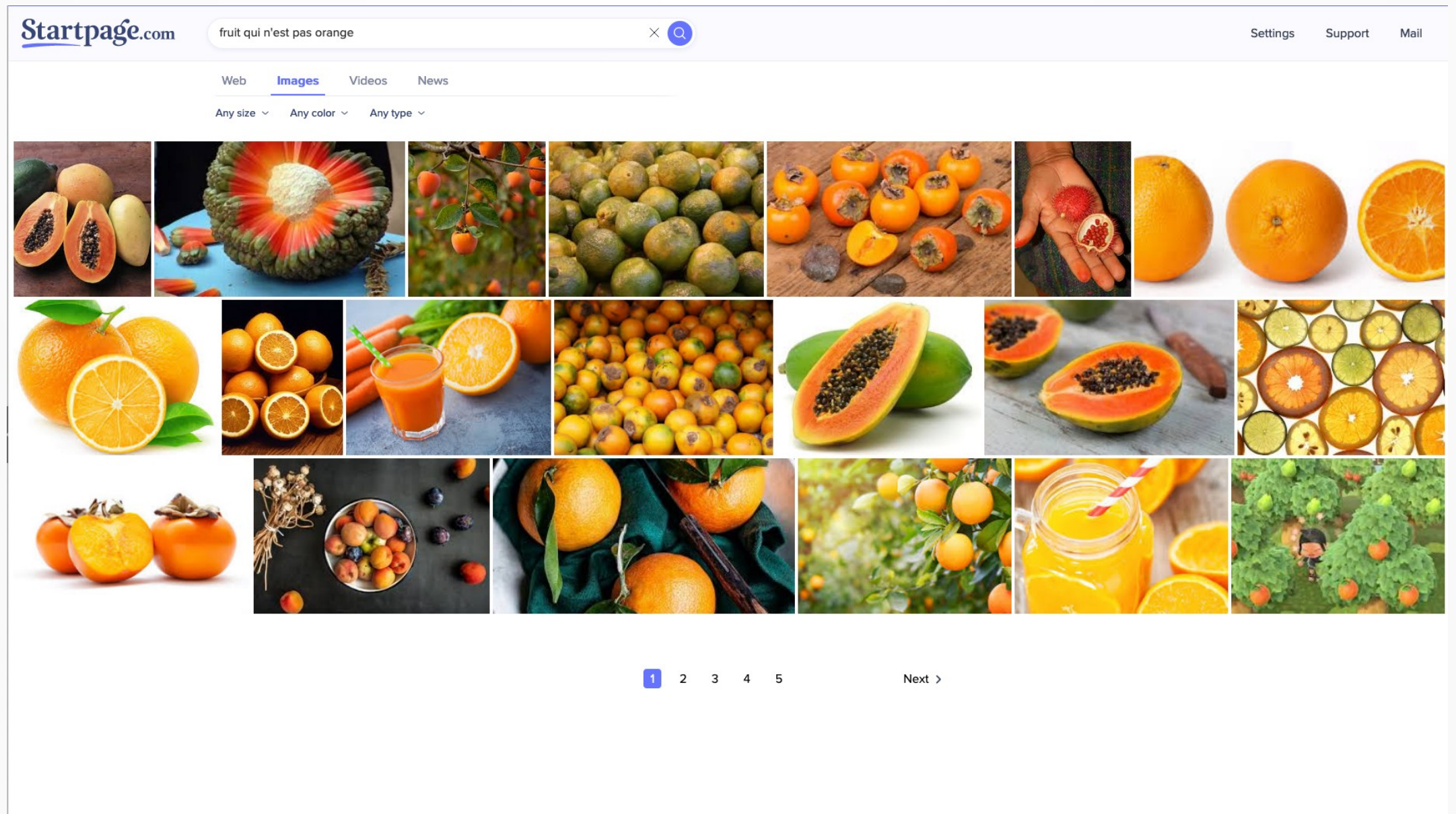
Le web sémantique

RDF : Resource Description Framework

Le web après google...

- Toutes les villes de plus de 10 000 habitants à moins de 75 km de Lille.
- Toutes les entreprises qui recrutent des ingénieurs en informatique à moins de 30 km de chez moi.
- Nom d'un fruit qui n'est pas orange.

Le web après google...



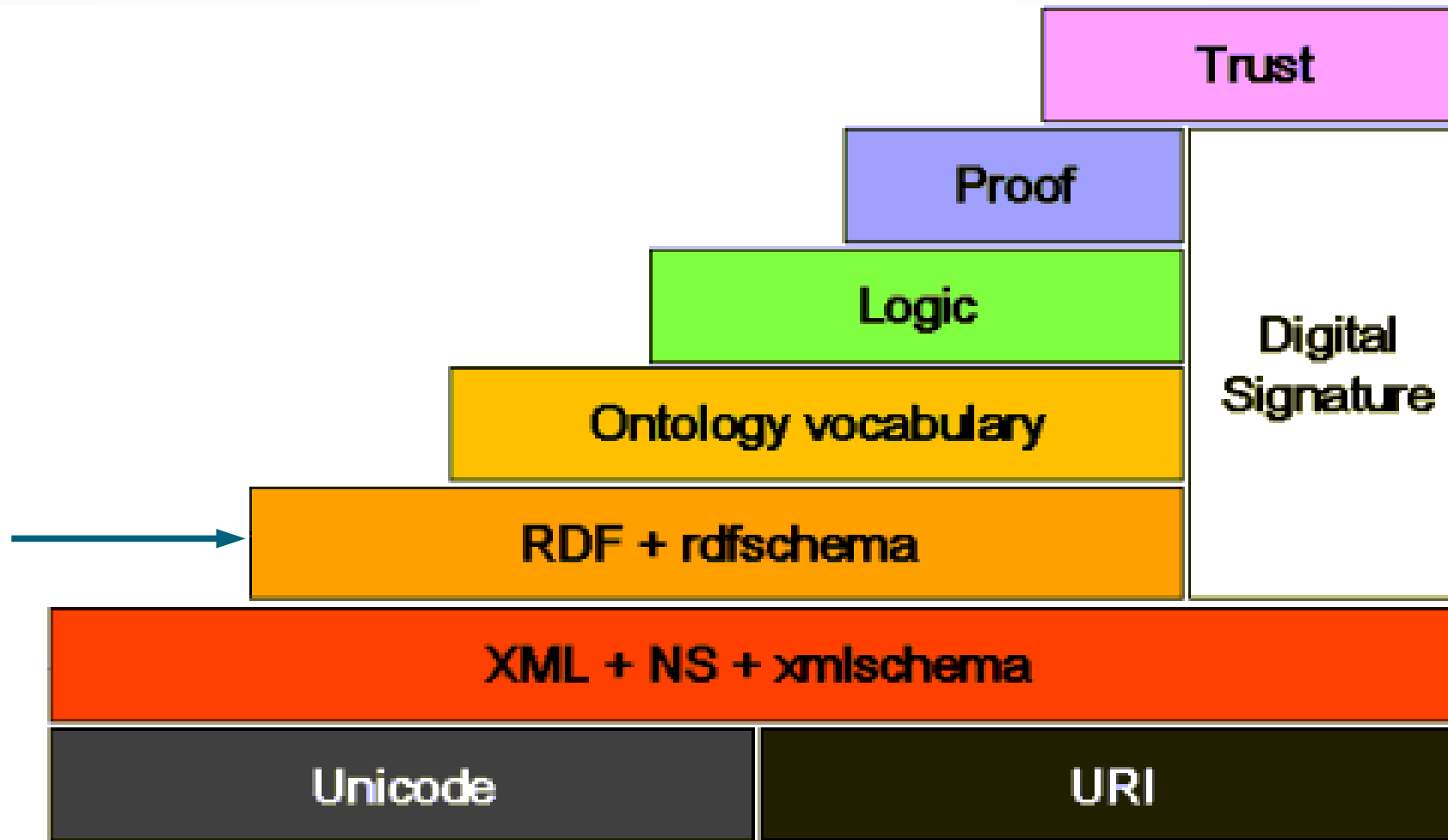
Le web sémantique

- 2001, idée popularisée par Tim Berners-Lee
- Ajouter du sens aux contenus web: description formelle :
 - Page qui traite d'un sujet
 - Événement
 - Coordonnées
 - ...
- Permettre aux machines de naviguer sur le web
- Toile hypertexte vs toile des données (*data web*)

Le web sémantique

- Linked Data : indiquer les liens sémantiques entre les ressources
- **Langage formel : RDF**
- Recherche de contenus plus simple et plus précise
- Mais nécessite un travail en amont par les créateurs de contenu

Semantic Web tower (Berners-Lee)



RDF : Resource Description Framework

- <http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>
- W3C en 1997, version 1.1 en 2014
- Associer des méta-informations à des ressources :
 - Page web
 - Item sur une page web
 - Document multimédia
 - ...
- On peut créer ses propres annotations
- On peut matérialiser les liens entre les ressources
- Framework très généraliste (trop? → ajout de schémas)

RDF et la notion de triplet

- Toutes les ressources peuvent être décrites formellement sous forme de triplets
- **sujet** *verbe* complément
- **sujet** *prédicat* objet
- **ressource** *propriété* valeur
- **Totor le castor** *s'intéresse à* Web sémantique
- **:totor** *foaf:interest* w3c:semWeb
- L'ensemble des triplets forme un graphe
- Toutes les ressources peuvent être sujet, prédicat ou objet

Exemple 1 : les fruits

- La pomme est un fruit
- La pomme peut être de couleur rouge
- La pomme peut être de couleur verte
- La pomme peut être de couleur jaune
- La pomme pousse en Europe
- La pomme pousse en Amérique
- ...

Exemple 1 : les fruits

- Un légume-fruit est un légume
- Un légume-fruit est un fruit
- *Solanum lycopersicum* est un légume-fruit
- *Solanum lycopersicum* est synonyme de tomate
- La tomate peut être de couleur rouge
- La tomate peut être de couleur noire
- La tomate peut être de couleur jaune
- ...

Exemple 1 : les fruits

- Le poivron est un légume-fruit
- Le poivron peut être de couleur rouge
- Le poivron peut être de couleur verte
- Le poivron peut être de couleur orange
- ...
- Le poivron est-il un fruit qui n'est pas orange ?

Syntaxes RDF

- Plusieurs syntaxes !
 - Notation 3
 - Turtle (sous-ensemble de Notation 3)
 - N-triples (sous-ensemble de turtle)
 - RDF/XML
 - JSON-LD
 - Etc.

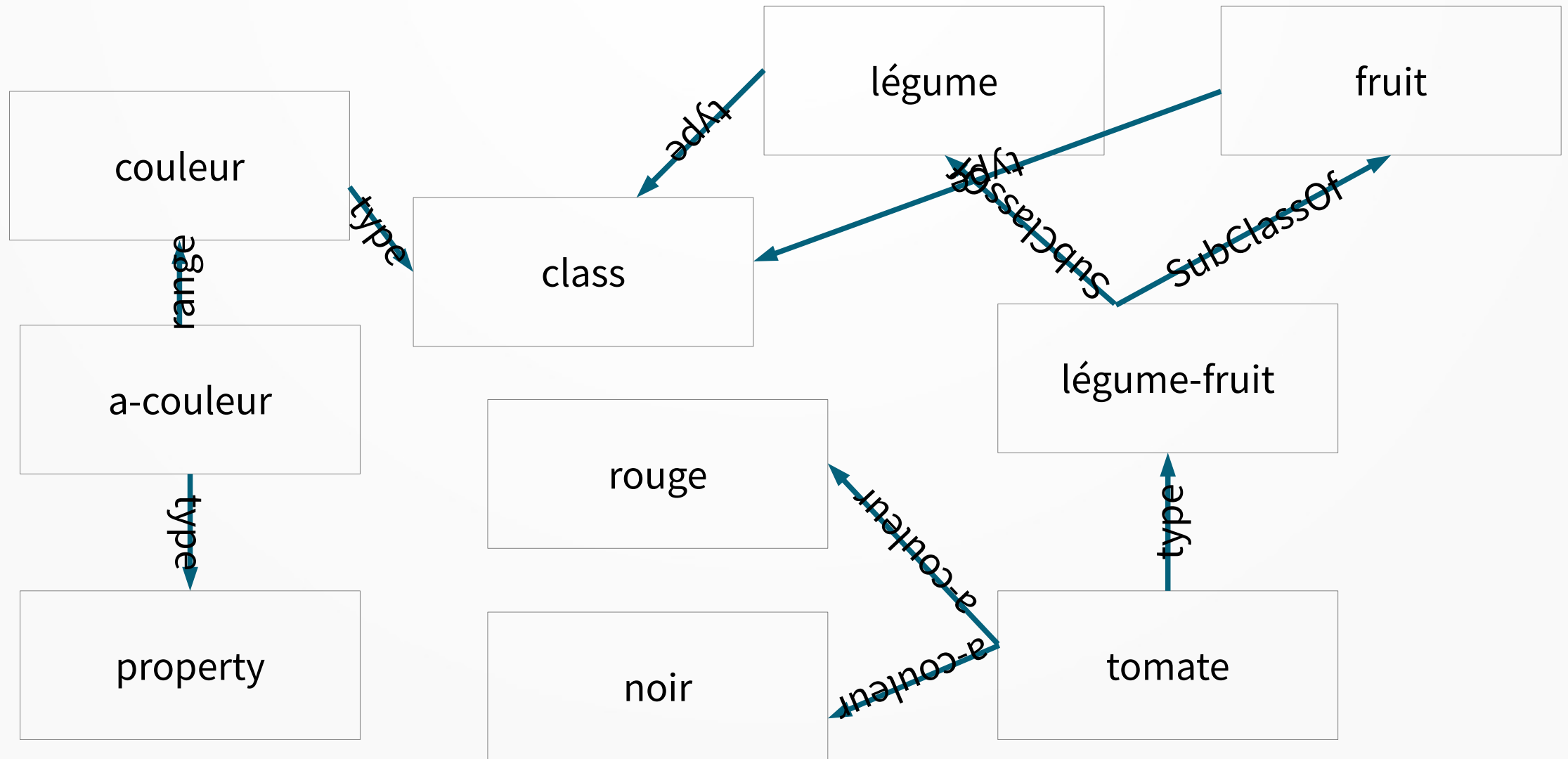
N-triples

- sujet prédicat objet .
- sujet propriété valeur .
- tomate est-un légume-fruit .
- légume-fruit est-un légume .
- légume-fruit est-un fruit .
- tomate couleur rouge .
- tomate couleur noir .
- ...

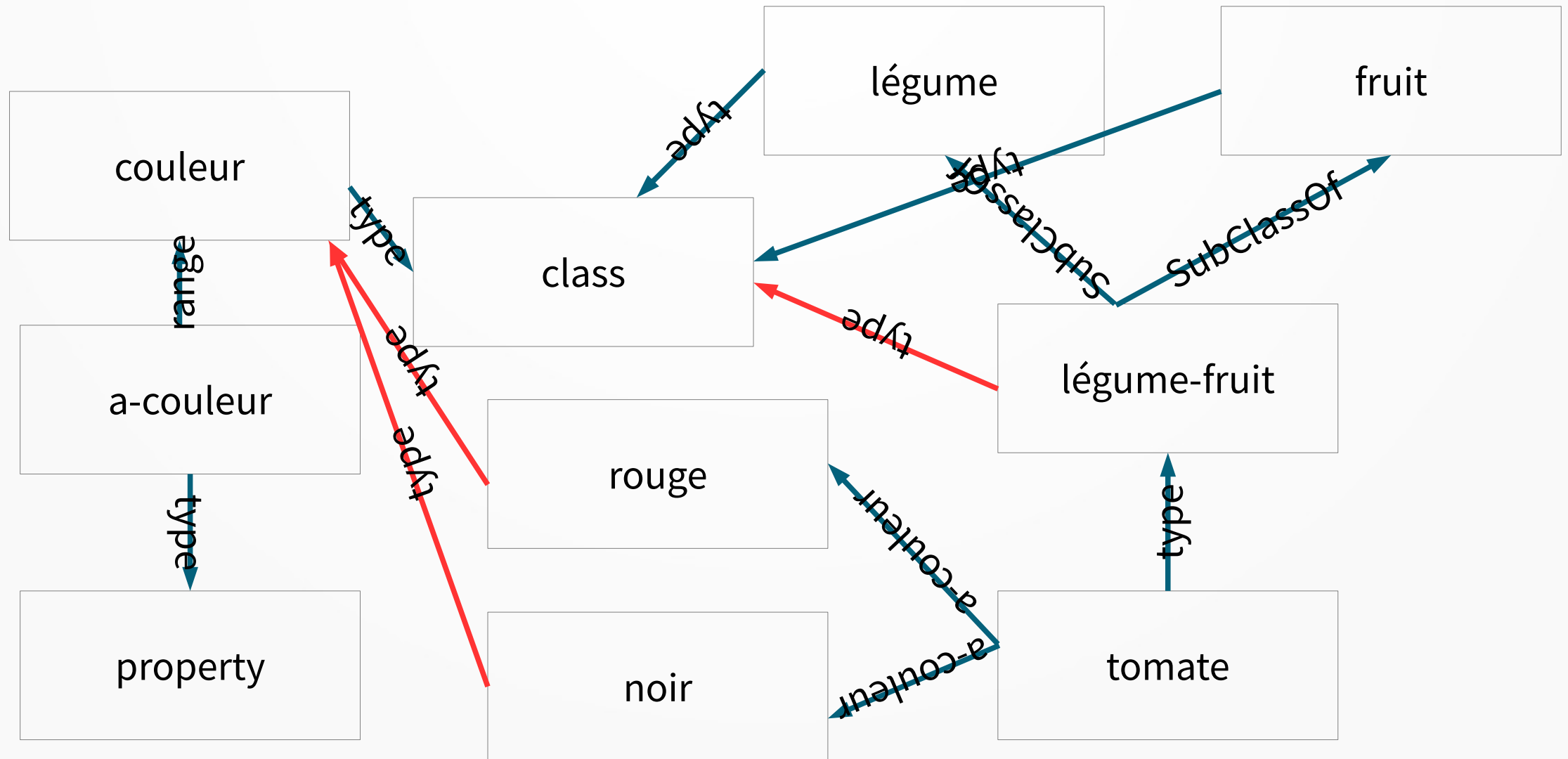
Les triplets forment un graphe

```
légume type class .  
fruit type class .  
légume-fruit subClassOf légume .  
légume-fruit subClassOf fruit .  
tomate type légume-fruit .  
a-couleur type property .  
a-couleur range couleur .  
couleur type class .  
tomate a-couleur rouge .  
tomate a-couleur noir .
```

Graphe associé



Déductions possibles



N-triples

```
<uri-sujet> <uri-prédictat> <uri-objet> .  
<uri-sujet> <uri-prédictat> "littéral" .  
<uri-sujet> <uri-prédictat> "littéral"^^<uri-type> .  
<uri-sujet> <uri-prédictat> "littéral"@lang .  
<uri-sujet> <uri-prédictat> _:anonyme .  
_:anonyme <uri-predicat> <uri-objet> .  
# Commentaire
```

N-triples

`<http://www.example.org/#totor>`

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

`<https://schema.org/Person> .`

`<http://www.example.org/#totor>`

`<https://schema.org/name>`

`"Totor le Castor" .`

N-triples

- En RDF, les prédicats/propriétés sont des ressources aussi (réification) :

```
<https://schema.org/name>
```

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

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

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

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

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

N-triples : les littéraux

```
<http://www.example.org/#totor>  
<http://xmlns.com/foaf/0.1/age>  
"23"^^<http://www.w3.org/2001/XMLSchema#integer> .
```

```
<http://www.example.org/#totor>  
<https://schema.org/name>  
"Totor the Castor"@en .
```

N-triples : blank nodes

- Noeuds vides : nœuds anonymes
- Pas associés à une ressource identifiée via une URI

`<http://www.example.org/#totor>`

`<https://schema.org/knowsAbout>`

`_:ws.`

- `_:ws <https://schema.org/name> "Web Sémantique".`

`_:ws`

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

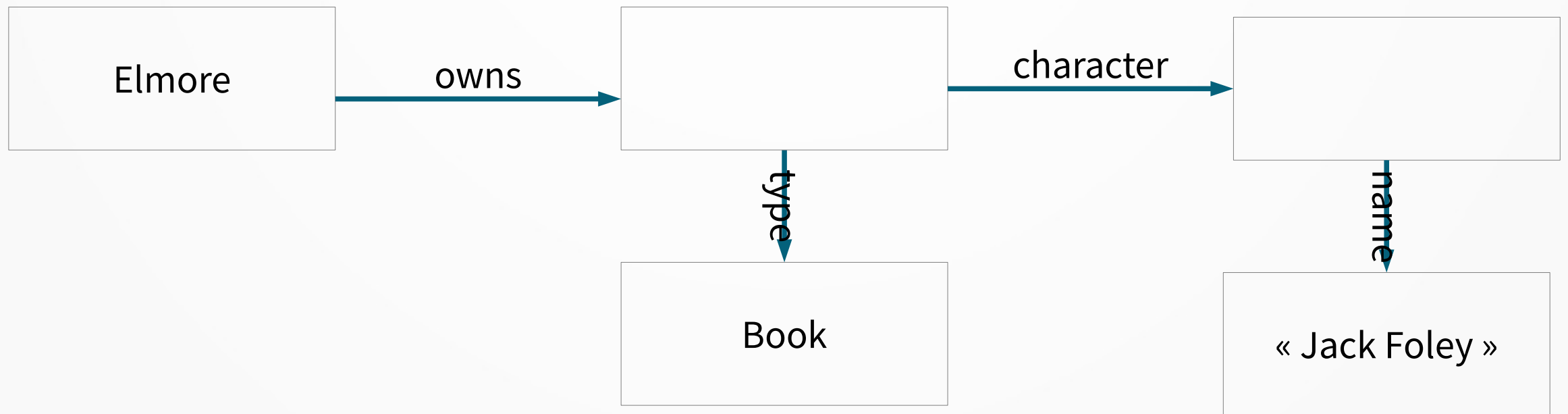
`<http://www.example.org/#Matière>.`

N-triples : blank nodes

- Le livre dont Elmore Leonard est l'auteur et qui a un personnage appelé « Jack Foley » :
- Elmore possède book1.
- book1 est un livre.
- book1 a pour personnage jf.
- jf a pour nom « Jack Foley ».

```
<...#elmore> <...#owns> _:book1 .  
_:book1 <...#type> <.../Book> .  
_:book1 <.../character> _:jf .  
_:jf <.../name> "Jack Foley" .
```

N-triples : blank nodes



N-triples

- Avantages :
 - Notion sous-jacente de triplets évidente
 - Très simple à écrire
 - Très simple à *parser* (script shell)
 - Grammaire de quelques lignes
- Inconvénients :
 - Peut être très verbeux

Turtle

- Surensemble de N-triples
- Possibilité de créer des préfixes (même idée que les namespaces XML)
- Virgules et point-virgules pour éviter les répétitions

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

```
@prefix schema: <https://schema.org/> .
```

```
@prefix ex: <http://www.example.org/> .
```

```
ex:totor rdf:type schema:Person ;
```

```
    schema:name "Totor le Castor" ,
```

```
                "Totor the Castor"@en .
```

```
schema:name rdf:type rdf:property .
```

Turtle

- Préfixe local : « : »

```
@prefix : <https://schema.org/>.
```

```
ex:totor rdf:type :Person .
```

```
@prefix : <#>. # Le document courant
```

```
:totor rdf:type schema:Person .
```

Turtle : « a »

```
:totor rdf:type schema:Person ;  
      schema:name "Totor le Castor" ,  
                  "Totor the Castor"@en .  
schema:name rdf:type rdf:property .
```

```
:totor a schema:Person ;  
      schema:name "Totor le Castor" ,  
                  "Totor the Castor"@en .  
schema:name a rdf:property .
```

Turtle : littéraux sans guillemets et ^^

- Entiers, décimaux, réels
- Listes : (a, b, c)
- Booléens
- ...

Turtle

- Noeuds anonymes : []
« Totor connaît quelque chose qui
 - s'appelle « web sémantique »
 - Est une « chose » au sens de schema.org »@prefix...

```
:totor schema:knowsAbout  
[  
  schema:name "Web Sémantique" ;  
  rdf:type schema:Thing  
] .
```

Turtle : nœuds anonymes

- Le livre dont Elmore Leonard est l'auteur et qui a un personnage appelé « Jack Foley » :
- N-triples :

```
<...#elmore> <.../owns> _:book1 .
```

```
_:book1 <...#type> <.../Book> .
```

```
_:book1 <.../character> _:jf .
```

```
_:jf <.../name> "Jack Foley" .
```

Turtle : nœuds anonymes

- Le livre dont Elmore Leonard est l'auteur et qui a un personnage appelé « Jack Foley » :
- Turtle :

```
:elmore schema:owns
[
  a schema:Book ;
  schema:character
    [
      schema:name "Jack Foley"
    ]
] .
```

RDF et les listes : first, rest et nil

- Une liste en Turtle :

(:x, :y, :z)

- Sucre syntaxique pour :

[

 daml:first :x ;

 daml:rest [

 daml:first :y ;

 daml:rest [

 daml:first :z ;

 daml:rest daml:nil]]] .

RDF et les listes : first, rest et nil

- Une liste en Turtle :

`(:x, :y, :z)`

- N-triples :

`_:lst1 <...#first> <...#x> .`

`_:lst1 <...#rest> _:lst2 .`

`_:lst2 <...#first> <...#y> .`

`_:lst2 <...#rest> _:lst3 .`

`_:lst3 <...#first> <...#z> .`

`_:lst3 <...#rest> <...#nil> .`

Fun with triples : restrictions

Un légume vert est un légume dont la couleur est forcément le vert

```
légume-vert type légume ,  
    [ type Restriction ;  
      onProperty aCouleur ;  
      allValues vert ] .
```


Réécriture en triplets

```
légume-vert type légume ,  
    [ type Restriction ;  
      onProperty aCouleur ;  
      allValues vert ] .
```

```
légume-vert type légume .  
légume-vert type _restr .  
_restr type Restriction .  
_restr onProperty aCouleur .  
_restr allValues vert .
```

La vraie version...

```
légume-vert type Class ,  
    [ type Restriction ;  
      onProperty aCouleur ;  
      allValuesFrom [type Class ;  
                     oneOf(vert)  
                     ]  
    ] .
```

La vraie version...

```
_restr type Restriction .  
_restr onProperty aCouleur .  
_restr allValuesFrom _vals .  
_vals type Class .  
_vals oneOf _1st .  
_1st first vert .  
_1st rest nil .
```

Turtle

- Avantages :
 - Très « human-friendly » : facile à lire, comprendre et écrire pour un humain
 - Idéal pour écrire des descriptions à la main
 - Relativement facile à *parser*
 - Plus compact que n-triples
- Inconvénients :
 - Plus difficile à *parser* que n-triples
 - Pas une syntaxe XML

Notation3 (ou N3)

- Sur-ensemble de Turtle
- Quelques ajouts

N3 : Équivalence de concepts : =

- Parfois des synonymes : `dc:title` et `ogp:title`
- `dc:title = ogp:title .`
- En Turtle :
- `dc:title owl:sameAs ogp:title .`

Équivalence de concepts

- = est un prédicat comme les autres

`dc:title a rdf:property ;`

`= ogp:title, schema:name .`

- En N-Triples:

`<http://.../title> <http://...#type> <http://...#property> .`

`<http://.../title> <http://...#sameAs> <http://...#title> .`

`<http://.../title> <http://...#sameAs> <https://.../name> .`

Implication et implication inverse

`:Book => :CreativeWork . # Any Book is a CreativeWork`

`:CreativeWork <= :Book .`

`:Book <http://www.w3.org/2000/10/swap/log#implies> :CreativeWork.`

N3 : is... of...

- Sucre syntaxique pour écrire comme en Anglais :
:swag schema:author :elmore .
:elmore **is** schema:author **of** :swag .

objet **is** prédicat **of** sujet .

N3 : keywords

- Pour se débarrasser du « : » initial
- Solution 1/2 : préfixer les mots-clés

@prefix ...

:elmore a schema:Person ;

is schema:author of :swag .

@keywords.

@prefix ...

elmore @a schema:Person ;

@is schema:author @of **swag** .

N3 : keywords

- Solution 2/2 : lister tous les mots-clés utilisés

@keywords a, is, of.

@prefix ...

elmore a schema:Person ;

is schema:author **of swag** .

N3 : chemins (paths)

- Elmore possède quelque chose

- Turtle :

`:elmore :owns [] .`

- N3 :

`elmore!owns .`

`subj!pred . ⇔ subj pred [] .`

N3 : chemins (paths)

- Elmore possède quelque chose qui est un livre

- Turtle :

`:elmore :owns [a :Book] .`

- N3 :

`e!lmore!owns a Book .`

`pred!obj ↔ [pred obj] ↔ [is obj of pred]`

N3 : chemins (paths)

- Elmore possède quelque chose qui a un personnage

- Turtle :

```
:elmore :owns [ :character [] ] .
```

- N3 :

```
elmore!owns!character .
```

N3 : chemins (paths)

- Elmore possède quelque chose qui a un personnage qui s'appelle « Jack Foley »

- Turtle :

```
:elmore :owns [ :character [ :name "Jack Foley" ] ] .
```

- N3 :

```
elmore!owns!character name "Jack Foley" .
```

N3 : chemins inverse

- Il y a quelques chose qui a pour auteur Elmore et qui a pour personnage « Jack Foley »
- Turtle :

```
[ :character "Jack Foley" ] ; :author :elmore .  
[ ] :character "Jack Foley" ; :author :elmore .  
[ :character "Jack Foley" ; :author :elmore ] .
```
- N3 :

```
elmore^author character "Jack Foley" .
```

N3 : suite

- Beaucoup d'autres choses :
 - Formules
 - Mots-clés `forAll` et `forSome`
 - Etc.
- Permet de décrire aussi des requêtes
- Plus puissant que RDF seul : un document N3 n'est pas forcément traduisible dans une autre syntaxe RDF

RDF/XML

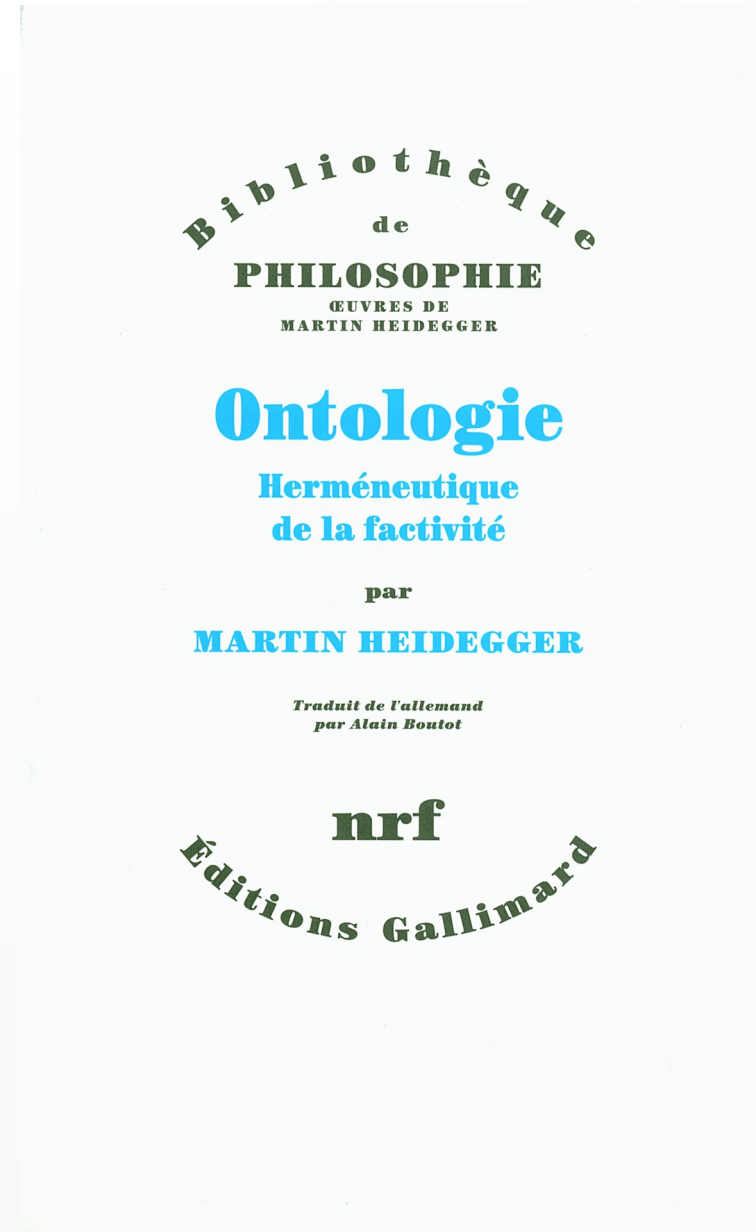
- Une syntaxe XML pour RDF
- Syntaxe historique de RDF

Quelques grammaires RDF...

- RDF est trop généraliste
- Comment raisonner sur un ensemble de triplets RDF ?
- Comment connaître la sémantique associée à un prédicat ?
- Il faut un vocabulaire commun : une ontologie
 - RDF Schema
 - Friend of a friend
 - Dublin core
 - OWL
 - ...

Ontologie (au singulier)

- Truc de philosophes



Les ontologies

- Truc d'informaticiens
- « spécification explicite d'une conceptualisation » [Gruber]
- « Une conceptualisation est une vue abstraite et simplifiée du monde que l'on veut représenter » [Gruber]

[Gruber], *Towards Principles for the Design of Ontologies Used for Knowledge Sharing in Formal Ontology in Conceptual Analysis and Knowledge Representation*, Kluwer Academic Publishers, 1993