

# Accès aux données basé sur des connaissances

**Marie-Laure Mugnier**

[marie-laure.mugnier@umontpellier.fr](mailto:marie-laure.mugnier@umontpellier.fr)

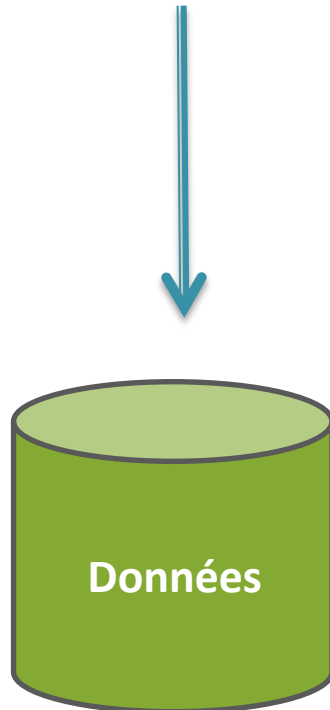
[mugnier@lirmm.fr](mailto:mugnier@lirmm.fr)



Requête (**SQL**, SPARQL, MongoDB, ...)

« trouver les patients atteints d'une maladie des poumons liée à une bactérie »

??



**Table Patients**

ID_Patient	Diagnostic
P	« légionellose (pneumophila) »
...	...

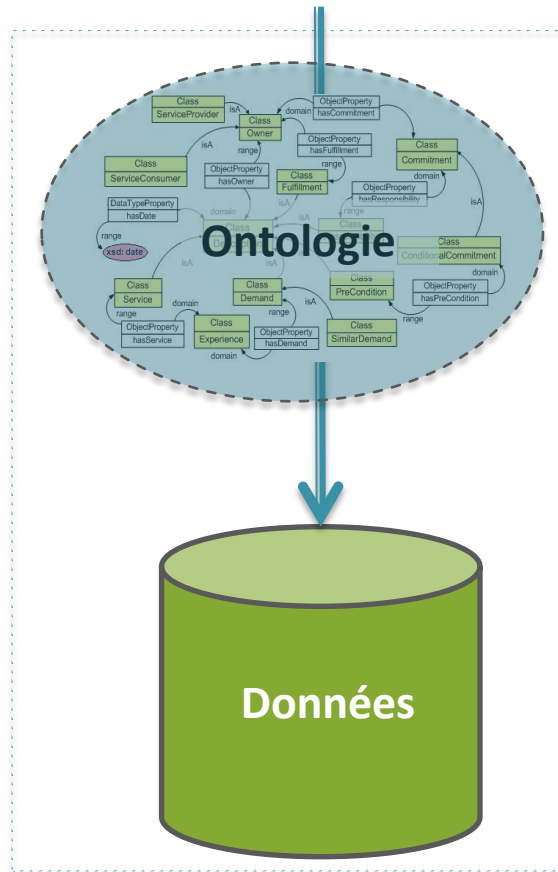
Base de données (**relationnelle**, RDF, JSON, ...)

On ne peut retrouver que les informations explicitement stockées

# INTERROGATION DE DONNÉES MÉDIATISÉE PAR DES CONNAISSANCES

## Requête

« trouver les patients atteint d'une maladie des poumons liée à une bactérie »



Base de connaissances

- Une **légionellose** est une pneumonie bactérienne
- Une pneumonie bactérienne est une pneumonie et une maladie bactérienne
- Une pneumonie est une **maladie des poumons**
- Une maladie bactérienne est une maladie causée par une **bactérie**

Si x est causé par y alors x **est lié à** y

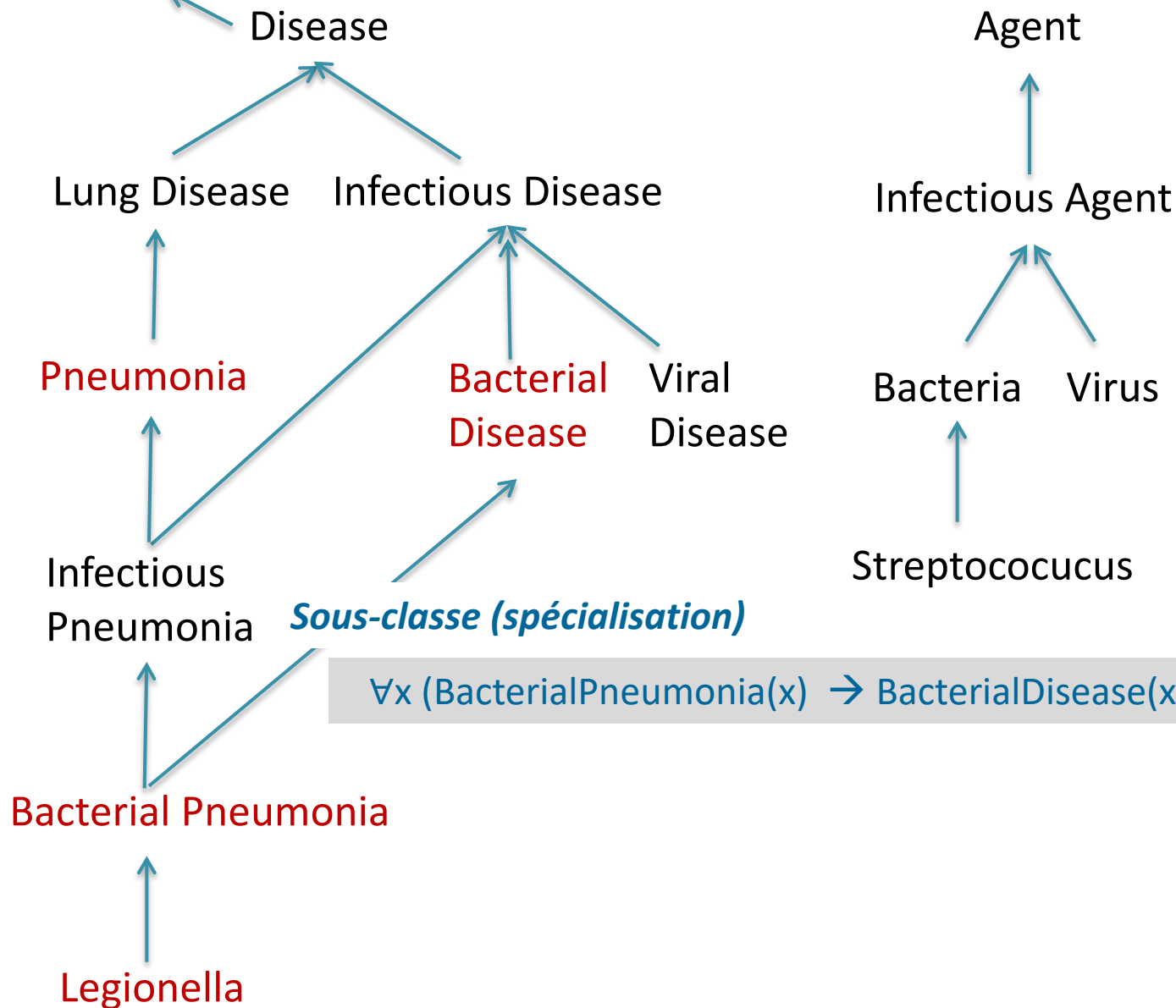
Si le **diagnostic** pour un **patient** x contient une maladie y alors x **est atteint de** y

Table Patients

ID_Patient	Diagnostic
P	« légionellose (pneumophila) »
...	...

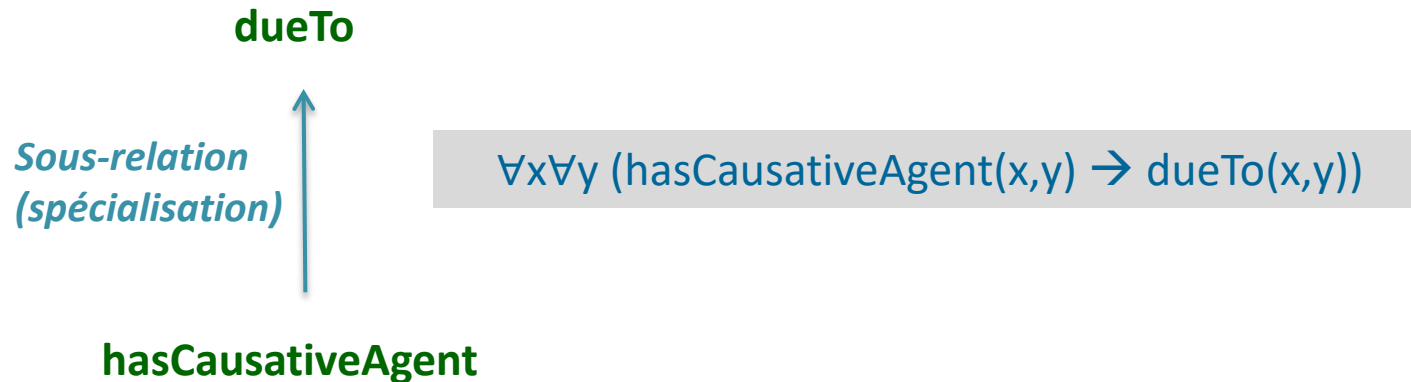
# HIÉRARCHIE DE CLASSES (CONCEPTS) - SNOMED CT

Disorder



# HIÉRARCHIES DE RELATIONS

On parle ici des relations qui peuvent exister **entre des instances de concepts**



Toute relation est munie d'une **signature** : chaque argument est typé par une classe

**hasCausativeAgent** :     **Argument 1 : Disease**  
                                 **Argument 2 : Organism**

$\forall x \forall y (\text{hasCausativeAgent}(x,y) \rightarrow \text{Disease}(x) \wedge \text{Organism}(y))$

Pour une relation **binaire** (RDFS, OWL) :

**Domaine** : type du 1<sup>er</sup> argument (« **domain** »)

**Co-domaine** : type du 2<sup>ème</sup> argument (« **range** »)

# QUELS LANGAGES POUR EXPRIMER DES ONTOLOGIES ?

## Langages très simples (raisonnements faciles à mettre en œuvre)

Hiérarchies de classes

Hiérarchies de relations binaires (aussi appelées « propriétés »)

Signatures de ces relations (« domain » et « range »)

→ **RDF Schema (RDFS)**

## Langages plus expressifs

**Logiques de description**

(OWL dans le Web Sémantique)

**Langages à base de règles**

Datalog, Règles existentielles, ...

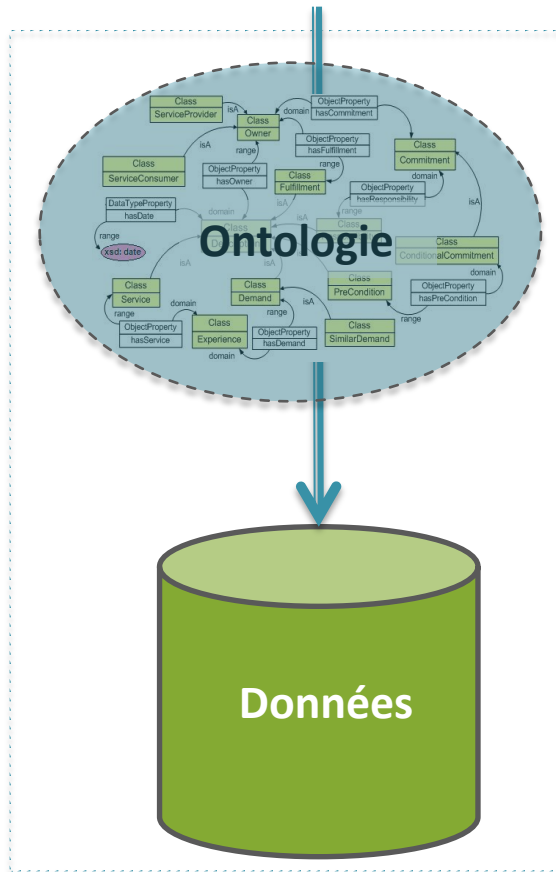
D'un point de vue abstrait, une ontologie peut être vue comme un ensemble de formules logiques de la forme :

$\forall X (\text{condition}[X,..] \rightarrow \text{conclusion}[X,..])$       « règles »

# REVENONS À NOTRE EXEMPLE

## Requête

« trouver les patients atteints d'une maladie des poumons liée à une bactérie »



Base de connaissances

- Une **légionellose** est une pneumonie bactérienne
- Une pneumonie bactérienne est une pneumonie et une maladie bactérienne
- Une pneumonie est une **maladie des poumons**
- Une maladie bactérienne est une maladie causée par une **bactérie**

Si x est causé par y alors x **est lié à** y

Si le **diagnostic** pour un **patient** x contient une maladie y alors x **est atteint de** y

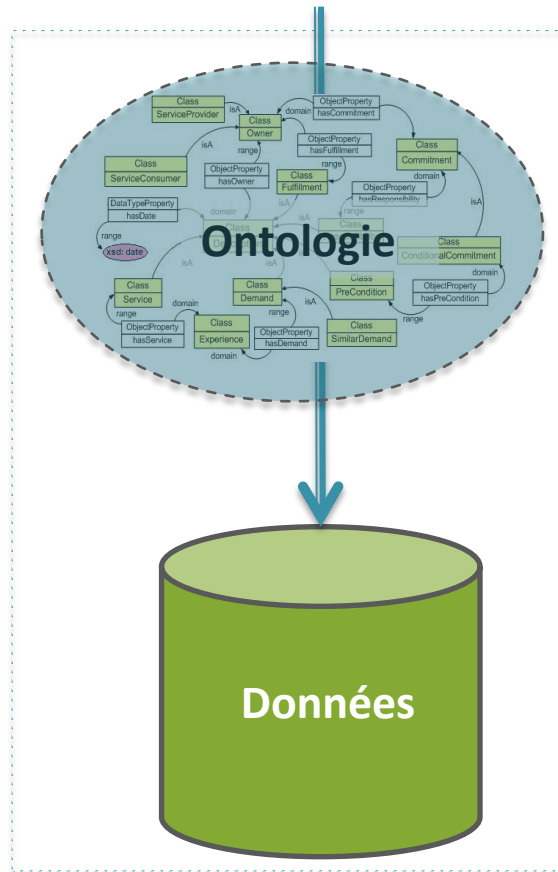
Table Patients

ID_Patient	Diagnostic
P	« légionellose (pneumophila) »
...	...

# INTERROGATION DE DONNÉES MÉDIATISÉE PAR DES CONNAISSANCES

## Exploiter les connaissances du domaine lors de l'accès aux données

### Requête



1- Enrichir le vocabulaire d'interrogation  
L'adapter à l'utilisateur / l'utilisation

*maladie des poumons, être lié à,  
être atteint de ...*

2 - Inférer de nouveaux faits  
non explicitement stockés dans la BD

*P est atteint d'une maladie causée par une  
bactérie, ...*

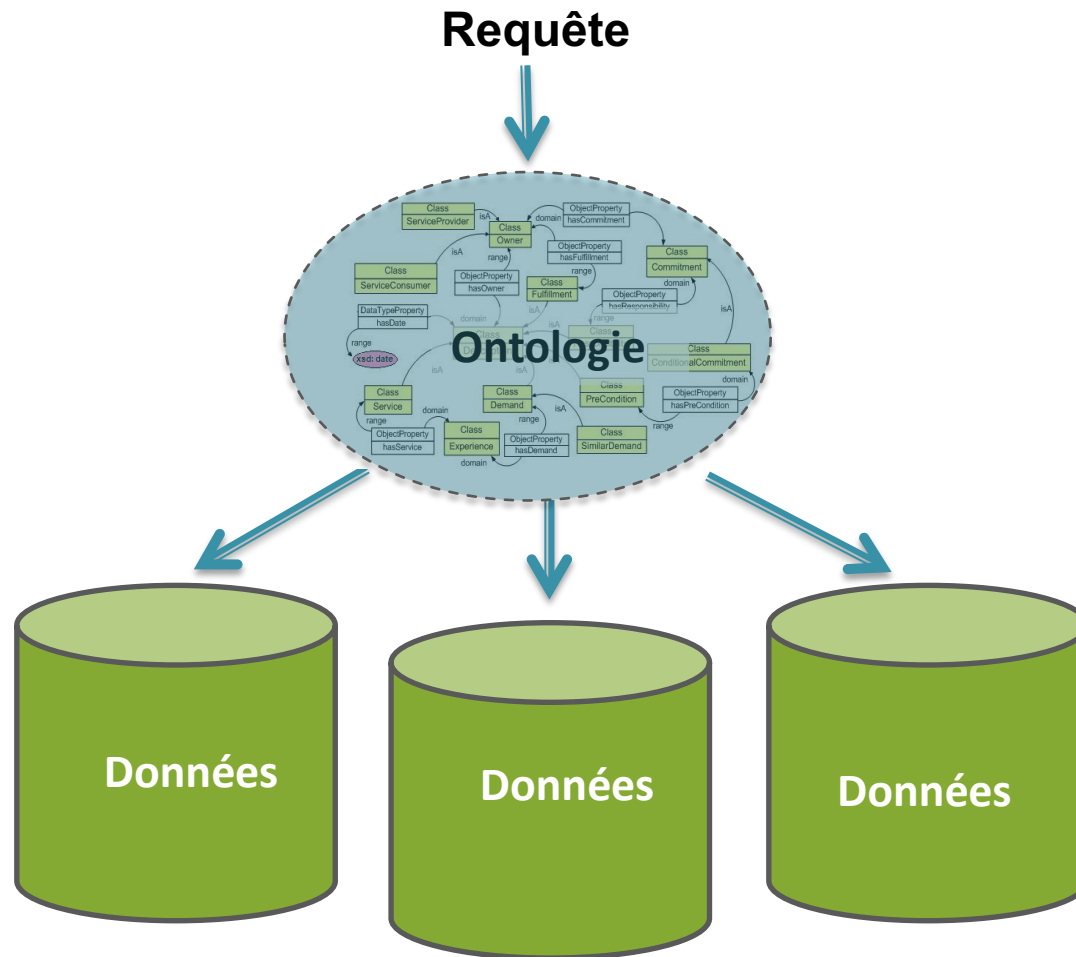
**Base de connaissances**



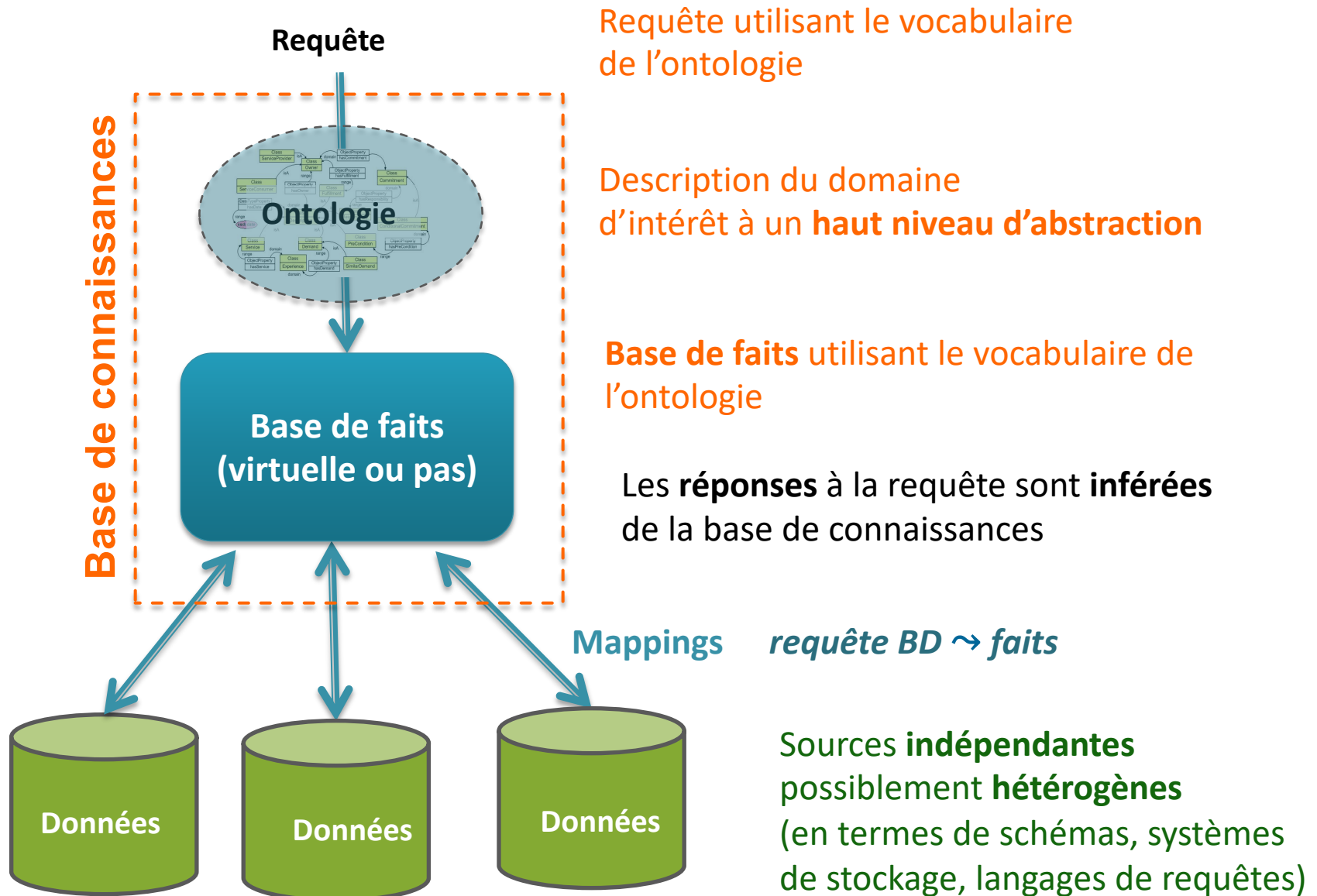
# INTERROGATION DE DONNÉES MÉDIATISÉE PAR DES CONNAISSANCES

## Exploiter les connaissances du domaine lors de l'accès aux données

3 - Fournir une vue unifiée de plusieurs sources de données



# SCHÉMA GÉNÉRAL (ONTOLOGY-BASED DATA ACCESS)



# CONTENU DU COURS

---

## Thématique

- Accès aux données médiatisé par des connaissances
- en utilisant la logique comme formalisme unificateur

- Vue logique des bases de données relationnelles (et de SQL)
- Vue logique de RDFS (et de SPARQL)
- Datalog : un langage de règles logiques
- Mappings entre les données et les connaissances
- Techniques d'interrogation (requêtage)

Mise en pratique dans un TP

Contrôle des connaissances :  
1 contrôle  
1 TP