



# **Processus d'annotation sémantique**

## **pour favoriser l'interopérabilité autour des données de**

## **biodiversité au sein de l'infrastructure AnaEE-France**

A. Chanzy, E. Aivayan, N. Beudez, C. Callou, P. Clastre, M. El-Hamadry,  
L. Greiveldinger, B. Jaillet, F. Lafolie, A. Léturgie, A. Maire, C. Martin, D.  
Maurice, N. Moitrier, G. Monet, H. Raynal, A. Schellenberger, R. Yahiaoui

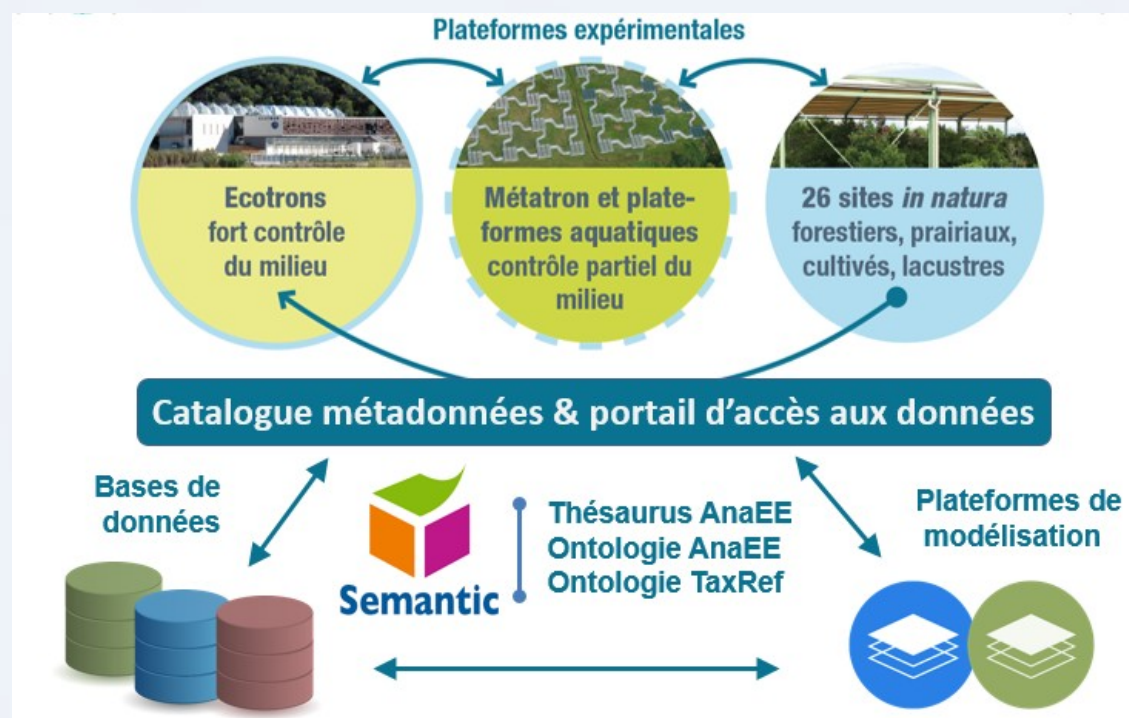


*R. Yahiaoui  
P. Christian  
D. Maurice  
A. Schellenberger*



# Intro

- Infrastructure nationale "Analyse et Expérimentation sur les Écosystèmes"
- Offre à la communauté scientifique des plateformes d'expérimentation, de modélisation et des BDD ( dont celles des SOERE )

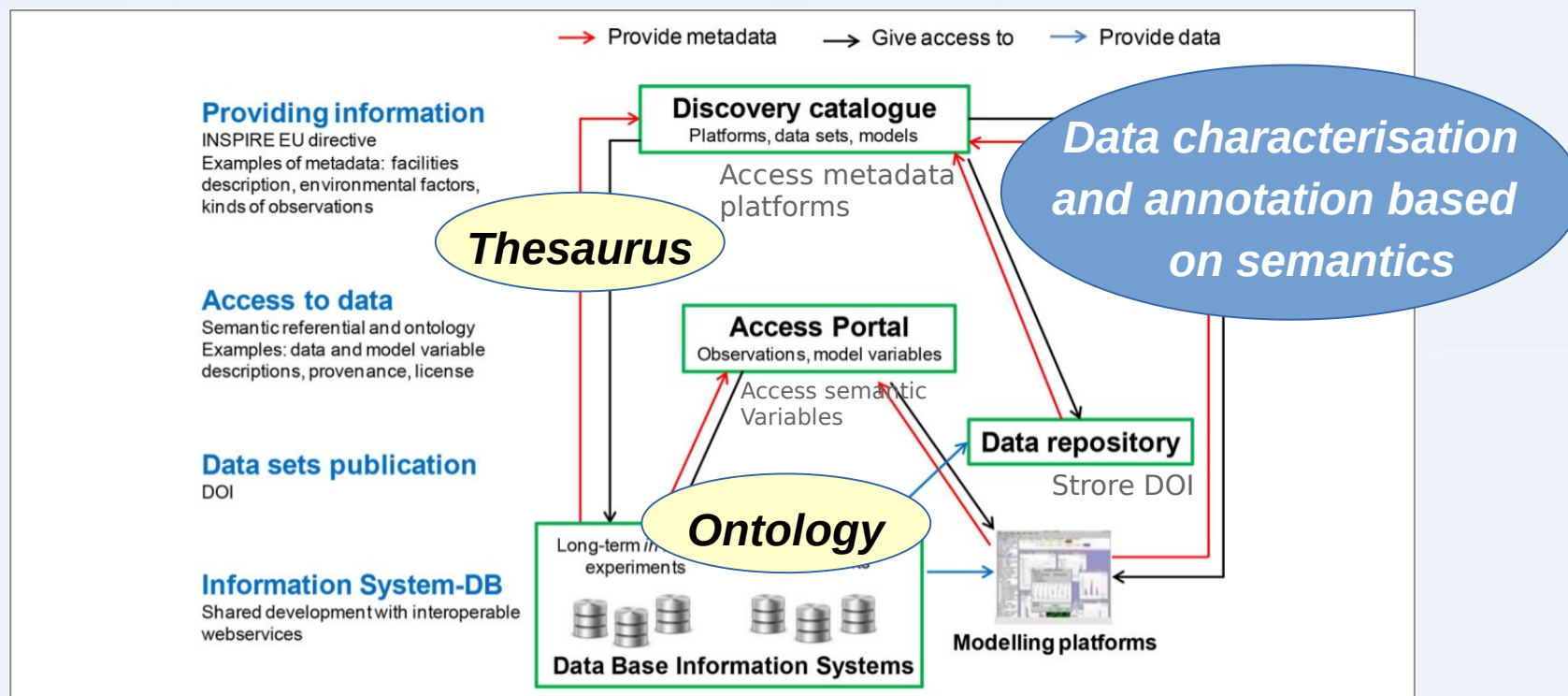


Hétérogénéité ⇒ Interopérable



# Intro

## AnaEE, Un Système d'Information distribué ...



Diverses DB

Concepts différents / Mêmes concepts décrits de différentes façons

⇒ Pour aligner ces concepts ( les homogénéiser ) et donc les rendre interoperable..

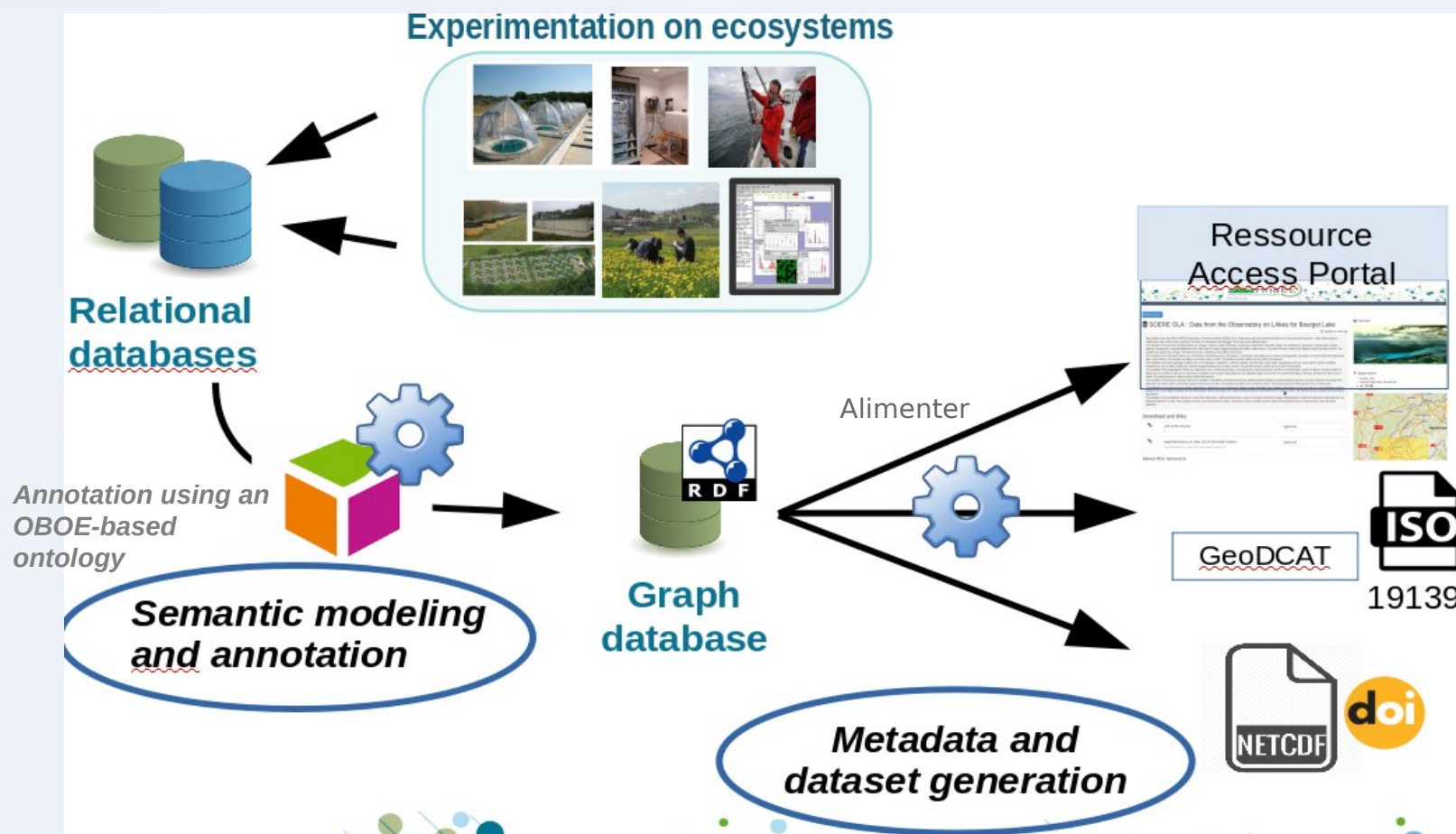
⇒ Approche choisie : **web Sémantique**





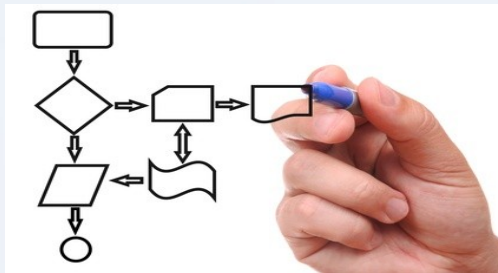
# Intro

Un flux de gestion des données/métadonnées





Et c'est justement ce qui fera l'objet de cette présentation...

# Processus d'annotation et production de données sémantiques



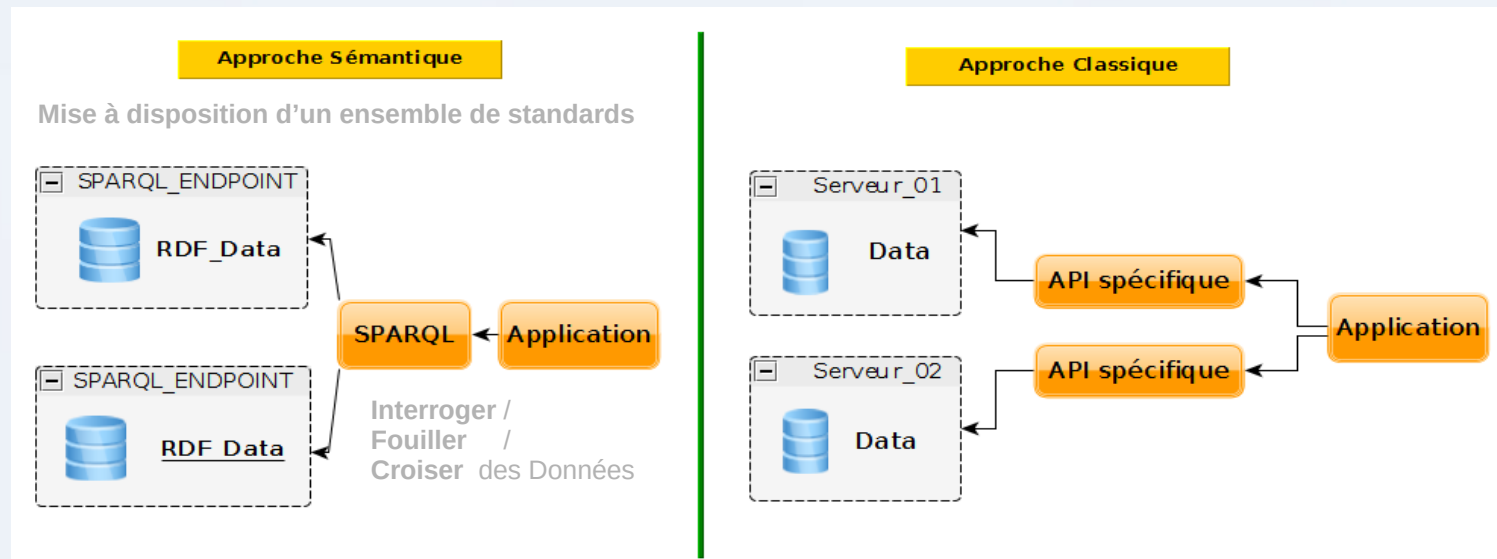
Définitions  
Démarche suivie



Aspects Fonctionnels   
Algorithme / Code 

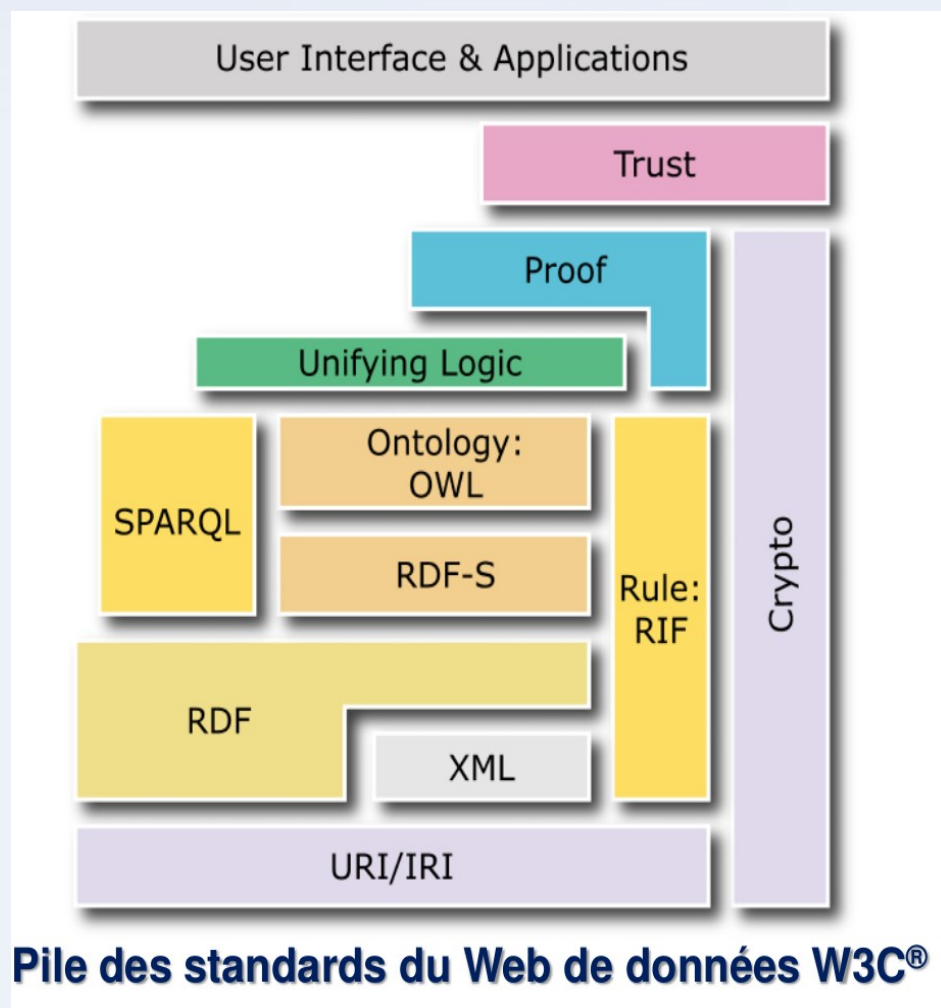


Le **Web sémantique**, ou toile sémantique, est un mouvement collaboratif mené par le World Wide Web Consortium (W3C) qui [favorise des méthodes communes pour échanger des données sur Internet pour accéder simplement..](#) ( Wikipedia )





## Pile de Standardisation



**URI - IRI** : Identifier n'importe quel objet du monde sur le web

**RDF** Moyen pour représenter les ressources et de leur associer des descriptions ( structurées )

**SPARQL** Interroger / Fouiller / Croiser des Données

**RDFS** Vocabulaire pour décrire des ontologies légères

**OWL** Vocabulaire pour décrire des ontologies plus poussées

Un père est un homme qui a au moins un enfant  
( MINACARDINALITY ... )

**Traçabilité** et **vérification** des données afin de les valider.

**Interaction** : Faciliter l'interaction des utilisateurs avec les données



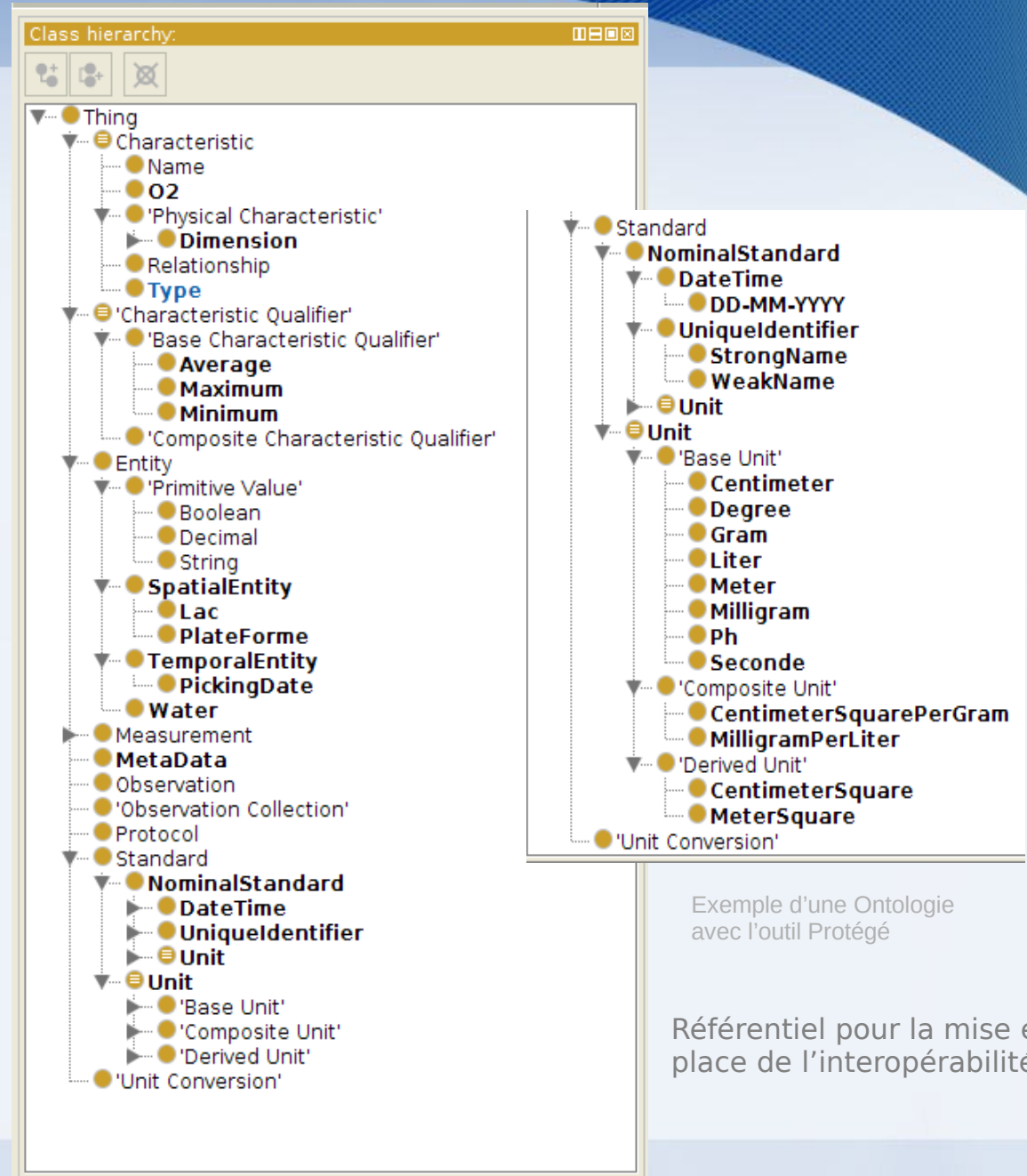
## Ontologie

Une Ontologie est un **réseau sémantique** regroupant un ensemble de **concepts décrivant un domaine**. Ces concepts sont **liés les aux autres** par des relations hiérarchiques d'une part, et sémantiques d'autres part.

Restriction, cardinalité des propriétés, symétrie, transitivité, inversement fonctionnel, intersection, union, disjonctions....

## Mais aussi Thésaurus

Liste structurée et hiérarchisée des termes d'un domaine du savoir plus ou moins large. Chacun des mots est relié à d'autres par divers types de relations hiérarchiques et/ou associatives

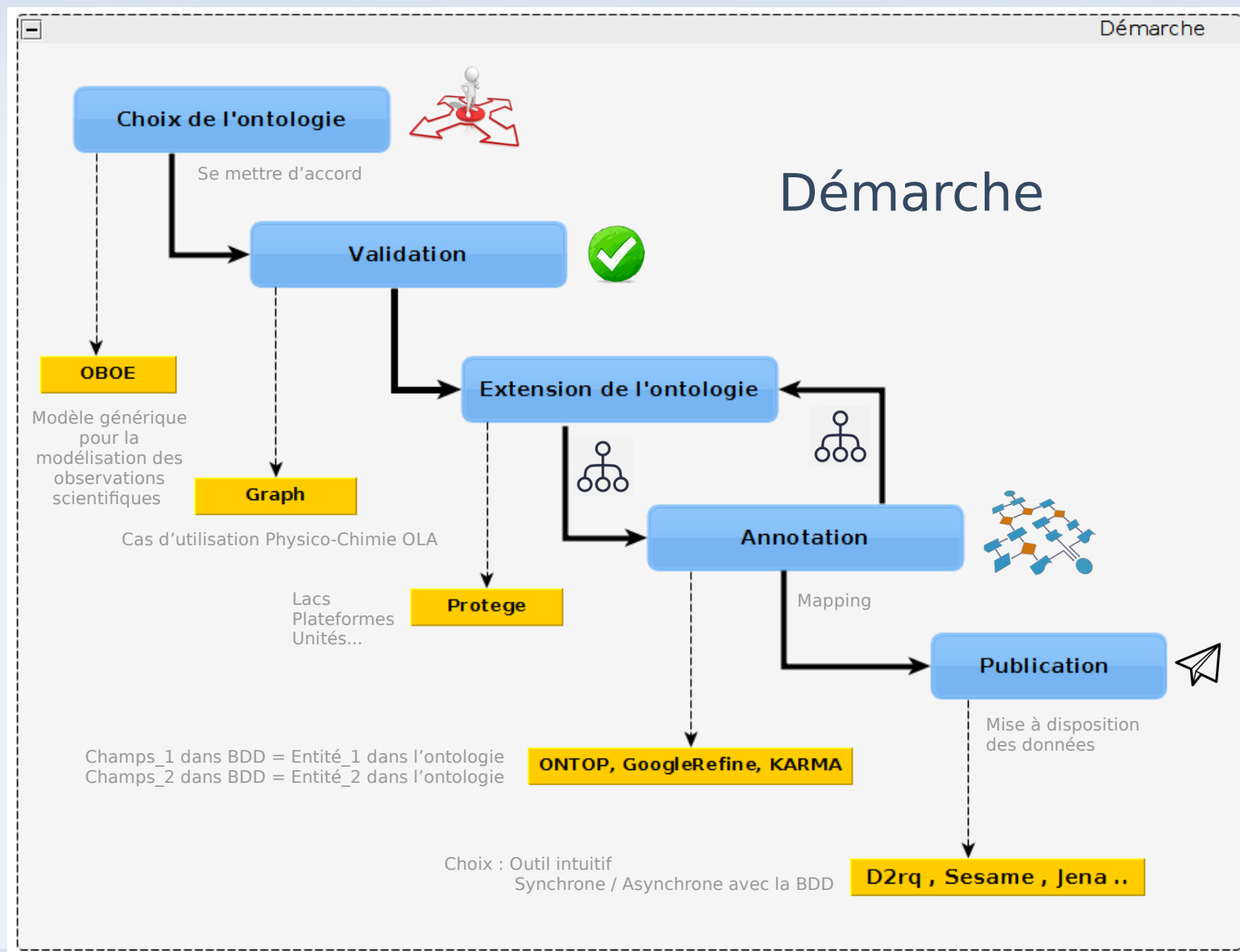


Exemple d'une Ontologie avec l'outil Protégé

Référentiel pour la mise en place de l'interopérabilité

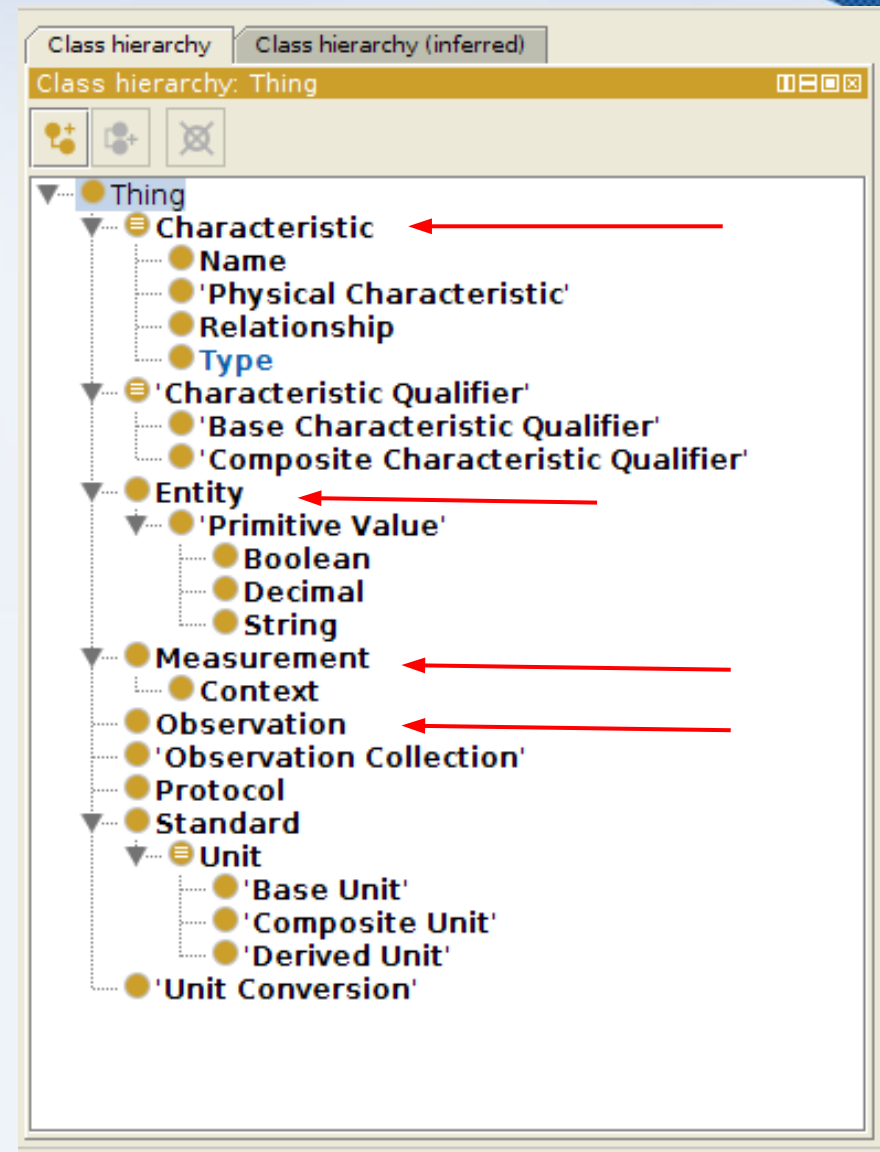
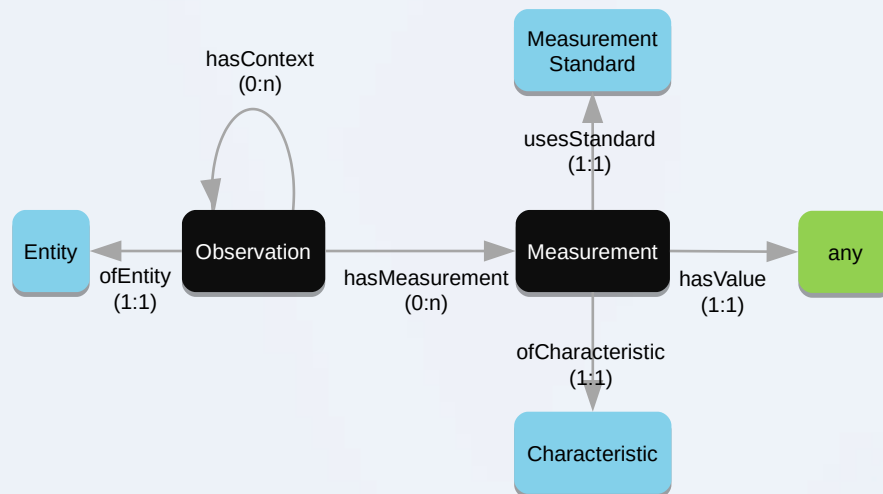


# Mise en place de l'interopérabilité - AnaEE-F



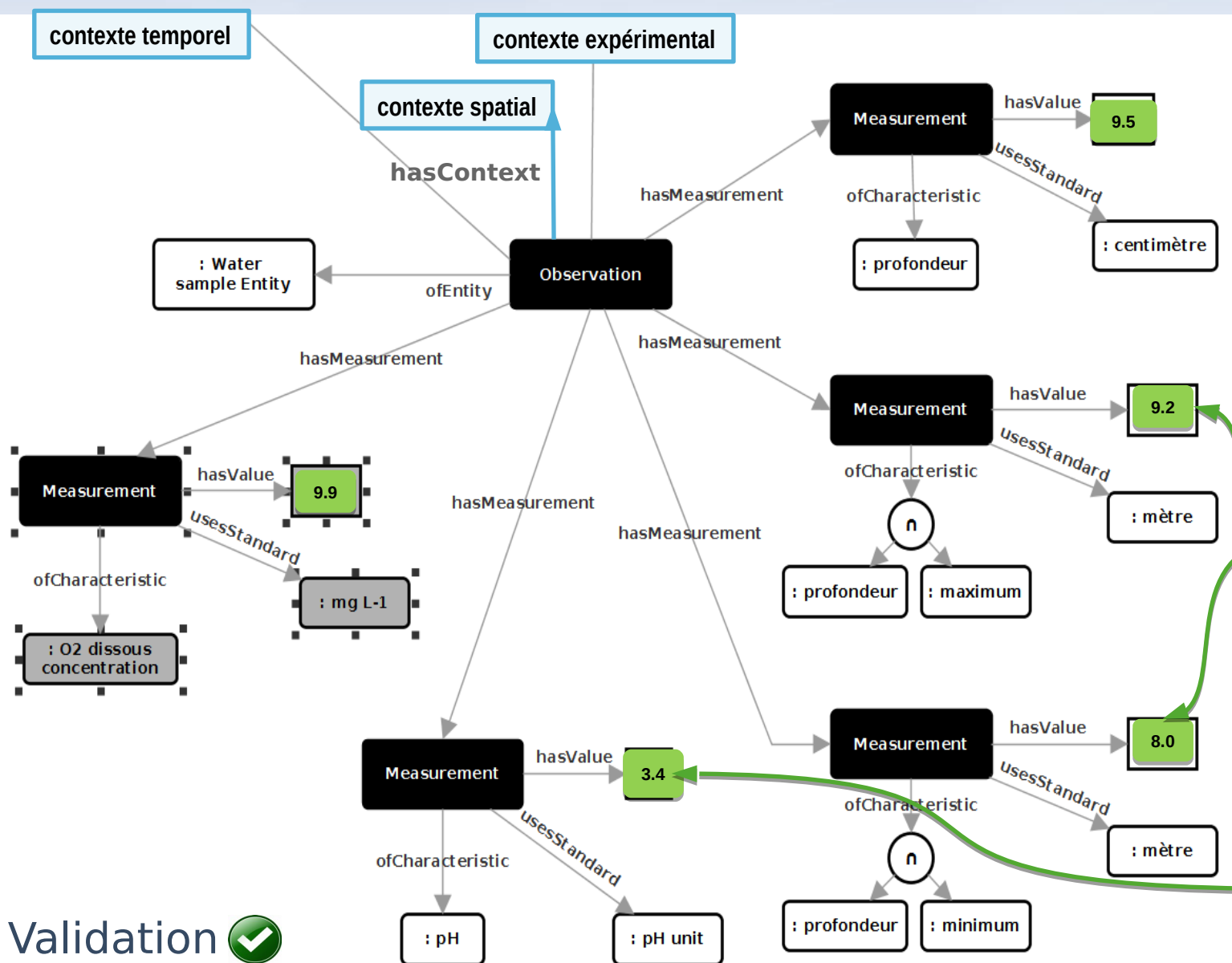
## OBOE

Ontologie conçue comme étant un modèle générique pour la modélisation et la représentation des observations scientifiques



# Validation

Modélisation sous forme de graphe. Cas d'utilisation : **Physico-Chimie**



mesure_chimie_mchimie	
<input type="checkbox"/>	mchimie_id
<input checked="" type="checkbox"/>	profondeurmax
<input checked="" type="checkbox"/>	profondeurmin
<input checked="" type="checkbox"/>	profondeurreelle
<input type="checkbox"/>	sschimie_id

valeur_mesure_chimie_vmchimie	
<input type="checkbox"/>	vmchimie_id
<input type="checkbox"/>	flag
<input type="checkbox"/>	ligne_fichier_echange
<input checked="" type="checkbox"/>	valeur
<input type="checkbox"/>	mchimie_id
<input type="checkbox"/>	var_id

Validation

## Extension OBOE

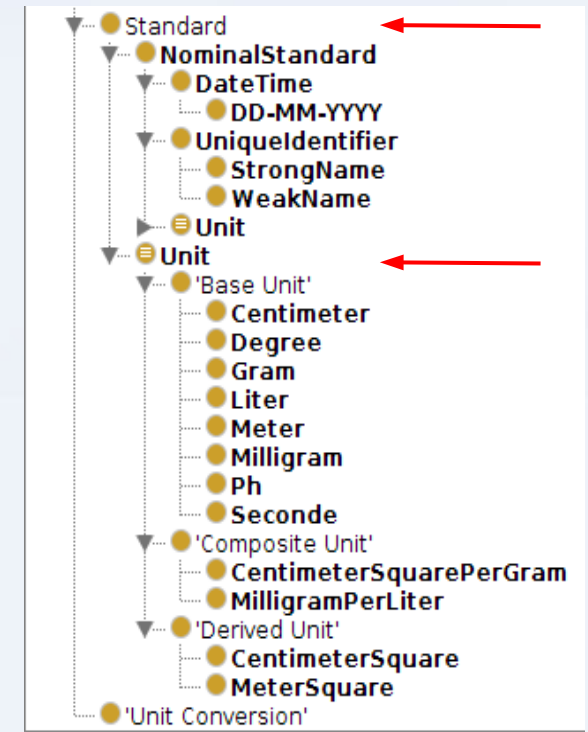
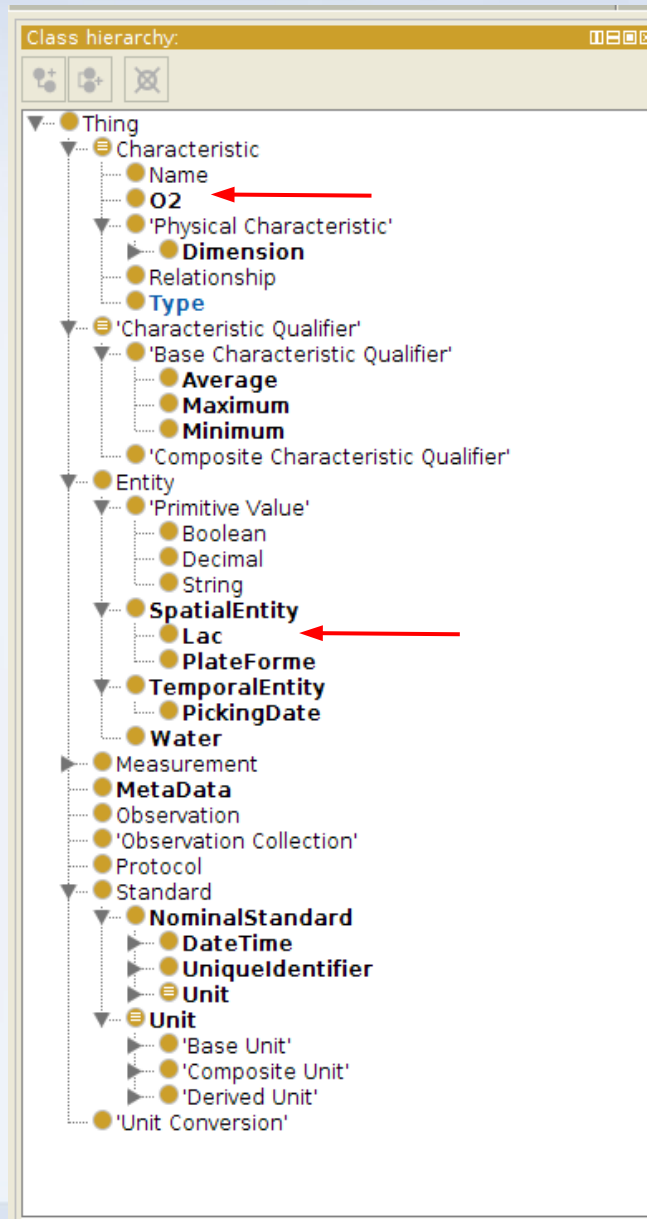
OBOE-CORE

+

'Thésaurus'

=

Ontologie AnaEE-F





Data		
Nom	employeur	HomePage
MichelGagnon	poly	<a href="http://www.professeurs.polymtl.ca/michel.gagnon">http://www.professeurs.polymtl.ca/michel.gagnon</a>

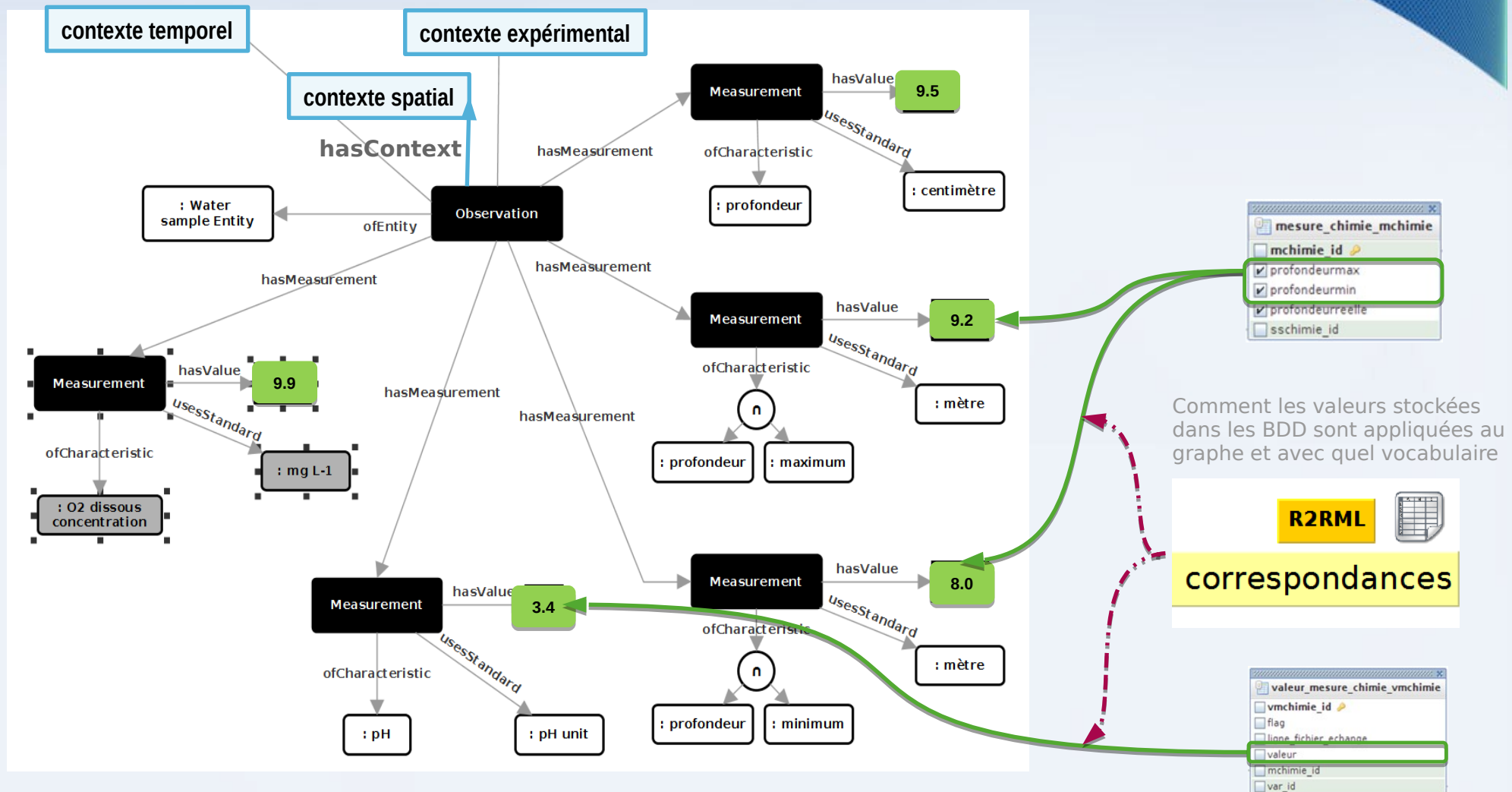


N-Triples

```
#s1 :Nom "MichelGagnon"
#s1 :Employeur "poly"
#s1 :HomePage "http://www.professeurs.polymtl.ca/michel.gagnon"
```

Le Sujet représente la ressource ,  
 Le Prédicat représente une propriété applicable sur la ressource  
 L'objet représente une données ou une autre ressource

# Annotation [ Transformation RDB - RDF ( 2/2 ) ]



Maintenant qu'on sait ce qui doit être fait.. il ne reste plus qu'à trouver les bons outils...

# Inventaire des outils sémantiques ( Sparql Endpoint )

( Phase de Publication Des Données )



## TripleStore

### \* Sesame



- Robustesse : **K.O**
- Scaling out : **K.O**
- Performance : **ERR**

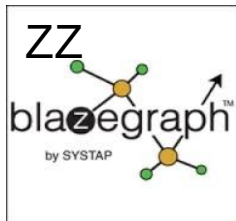
### \* Sol-RDF



- Robustesse : **OK**
- Scaling out : **OK**
- Performance :

REST

### \* BlazeGraph



- Robustesse : **OK**
- Scaling out : **OK \***
- Performance : **OK**

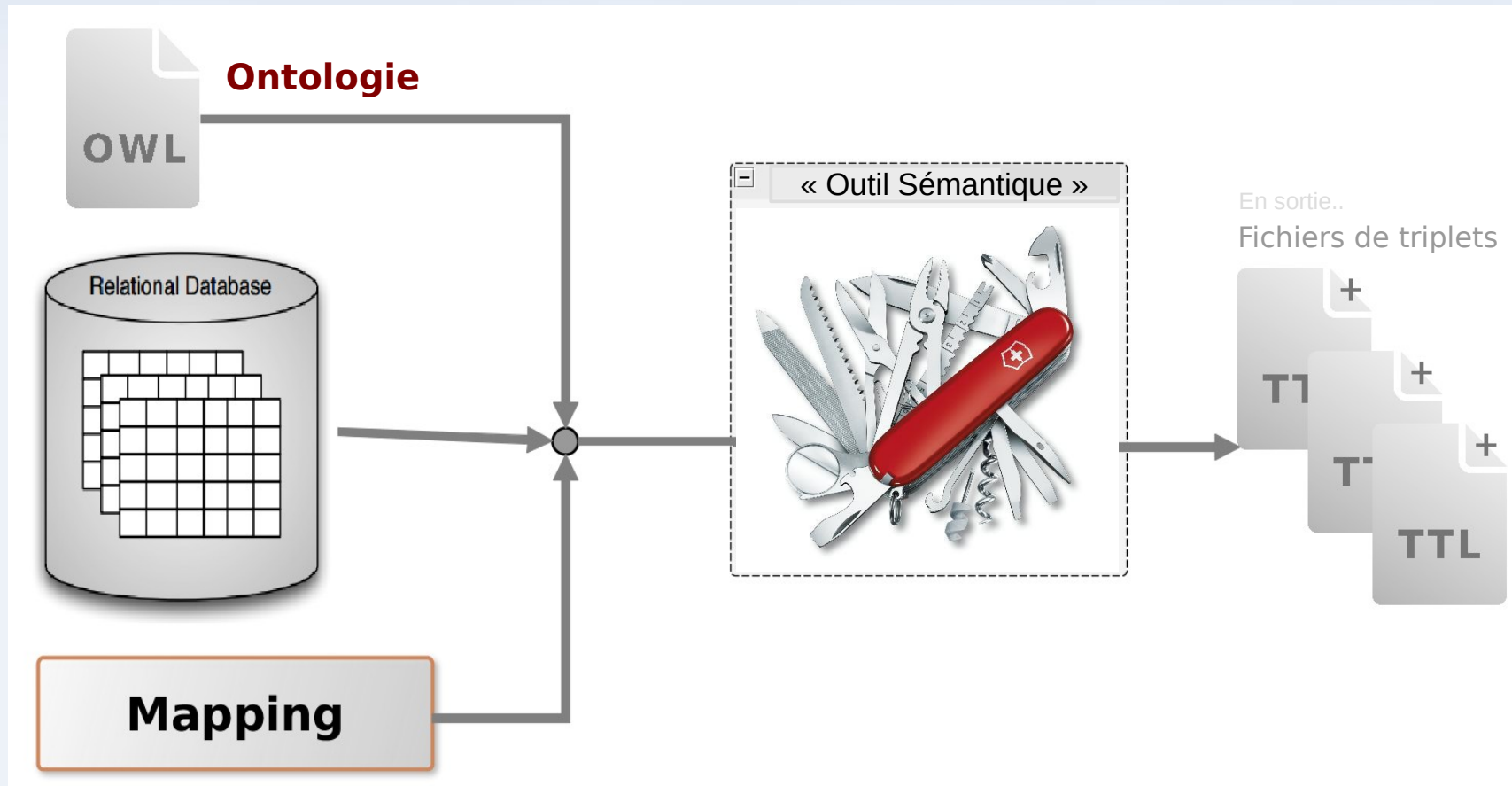
### \* Corese



- Robustesse : **OK**
- Scaling out :
- Performance : **OK**

## Bases de données orientées Graphes ( Structure plus généralisée que celles des triplestores )

## Inventaire des outils sémantiques ( Phase d'annotation )



Comment les données  
relationnelles sont transformées  
en données sémantique...



## Inventaire des outils sémantiques ( Phase d'annotation )



## Outils de transformation à la volée



```
d2rq:column      "Conferences.Name"
d2rq:condition   "PAPER.PDF IS NOT NULL"
d2rq:join        "Papers.Conference => Conferences.ConfID"
```



```
SELECT ID, VALUE FROM measurements
```

## - Transformation à la volée des requêtes

SPARQL ⇒ SQL ( Génération RDF à partir des BD-R )



- Mapping non Intuitif (Spécialement pour ceux qui ne manipulent que du SQL)

- ~~Fail Last~~ Erreurs Mapping détectées au Runtime- ~~Pas d'Interface graphique !~~ Projet Externe (AuReli)

## - On-the-fly Ontology-based Data Access



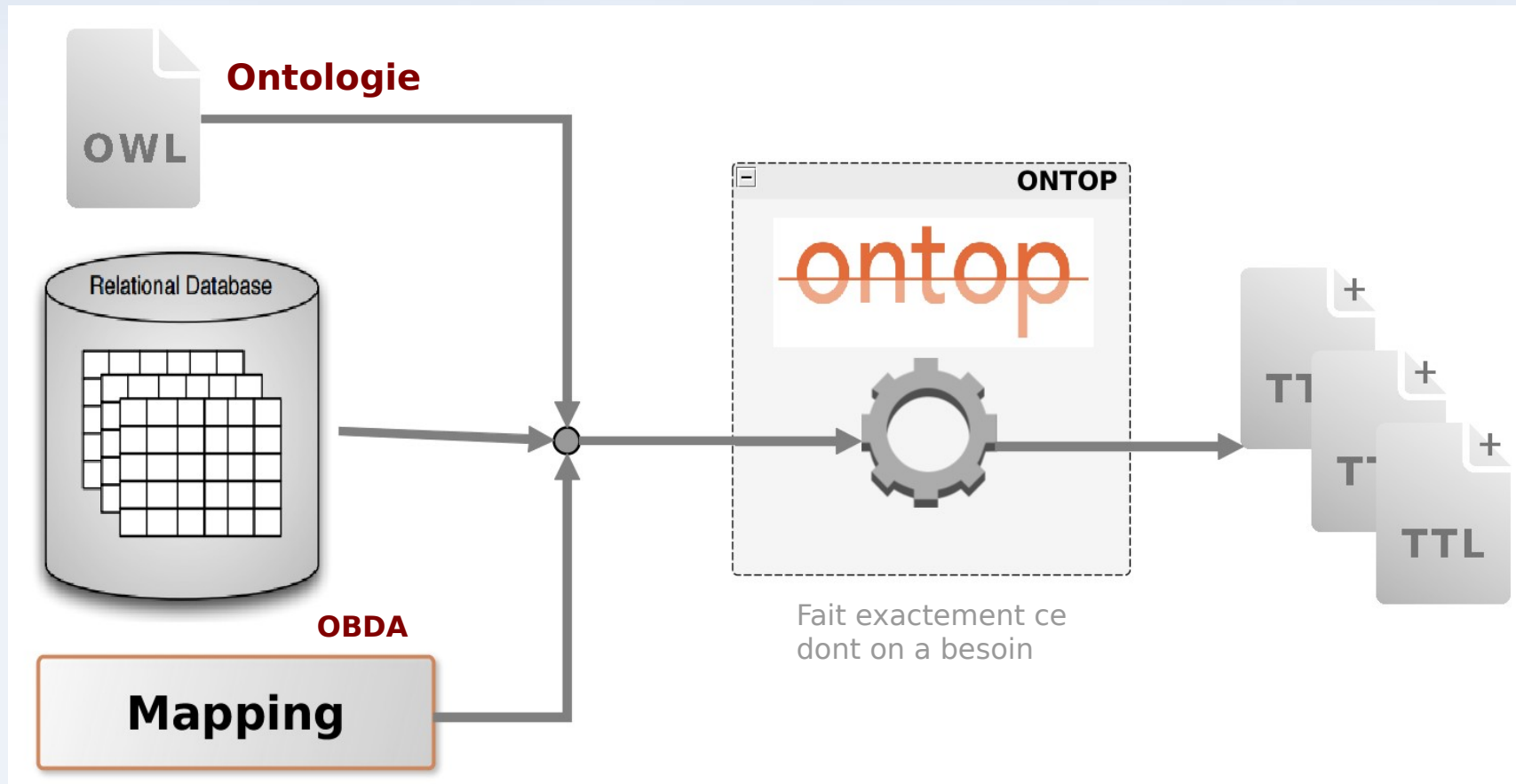
- Mapping Intuitif ( se base sur le SQL )

- ~~GUI intergée à Protegé~~- ~~Support « uniquement » SPARQL 1.0~~

## Inventaire des outils sémantiques ( Phase d'annotation )



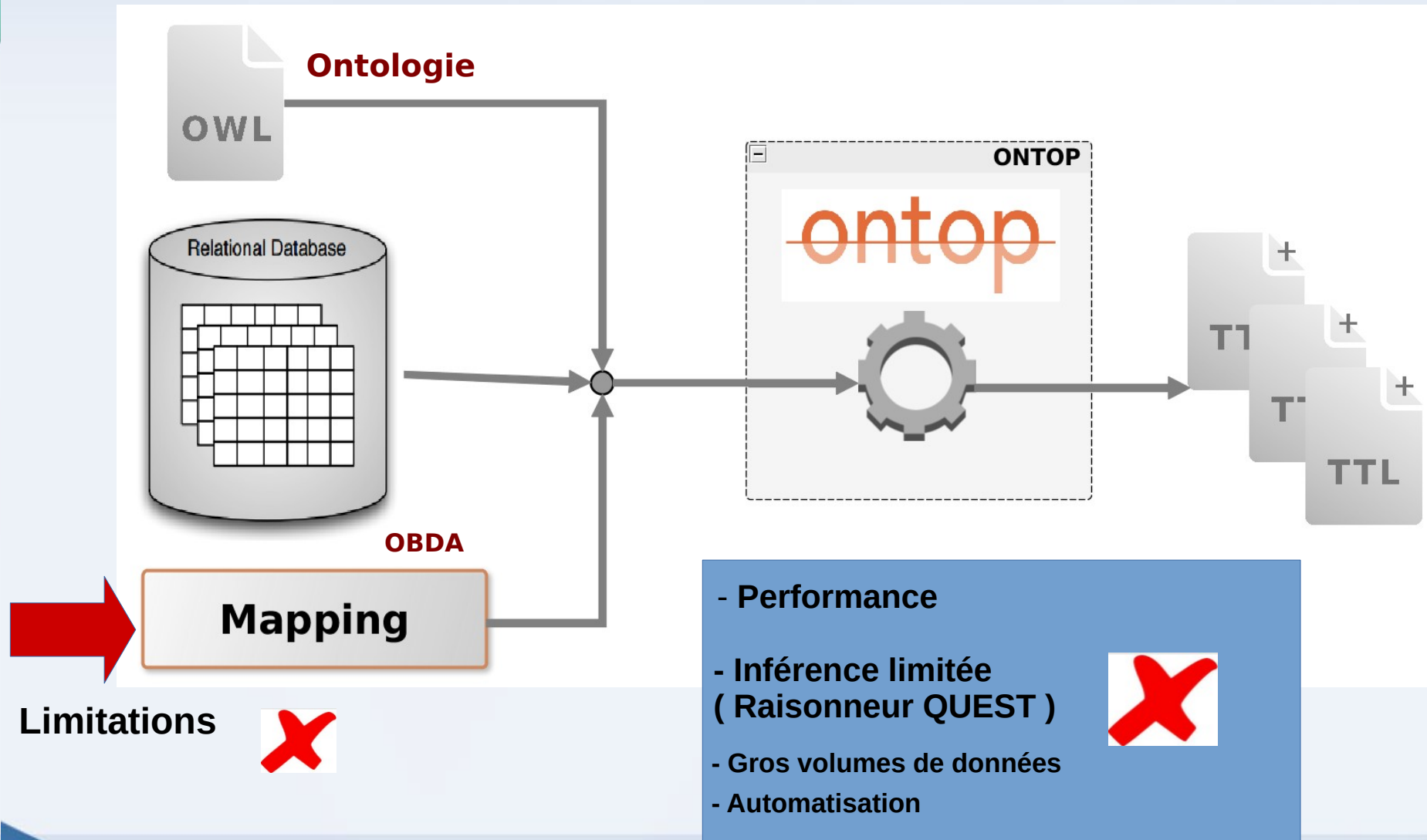
L'outil retenu : **Ontop**



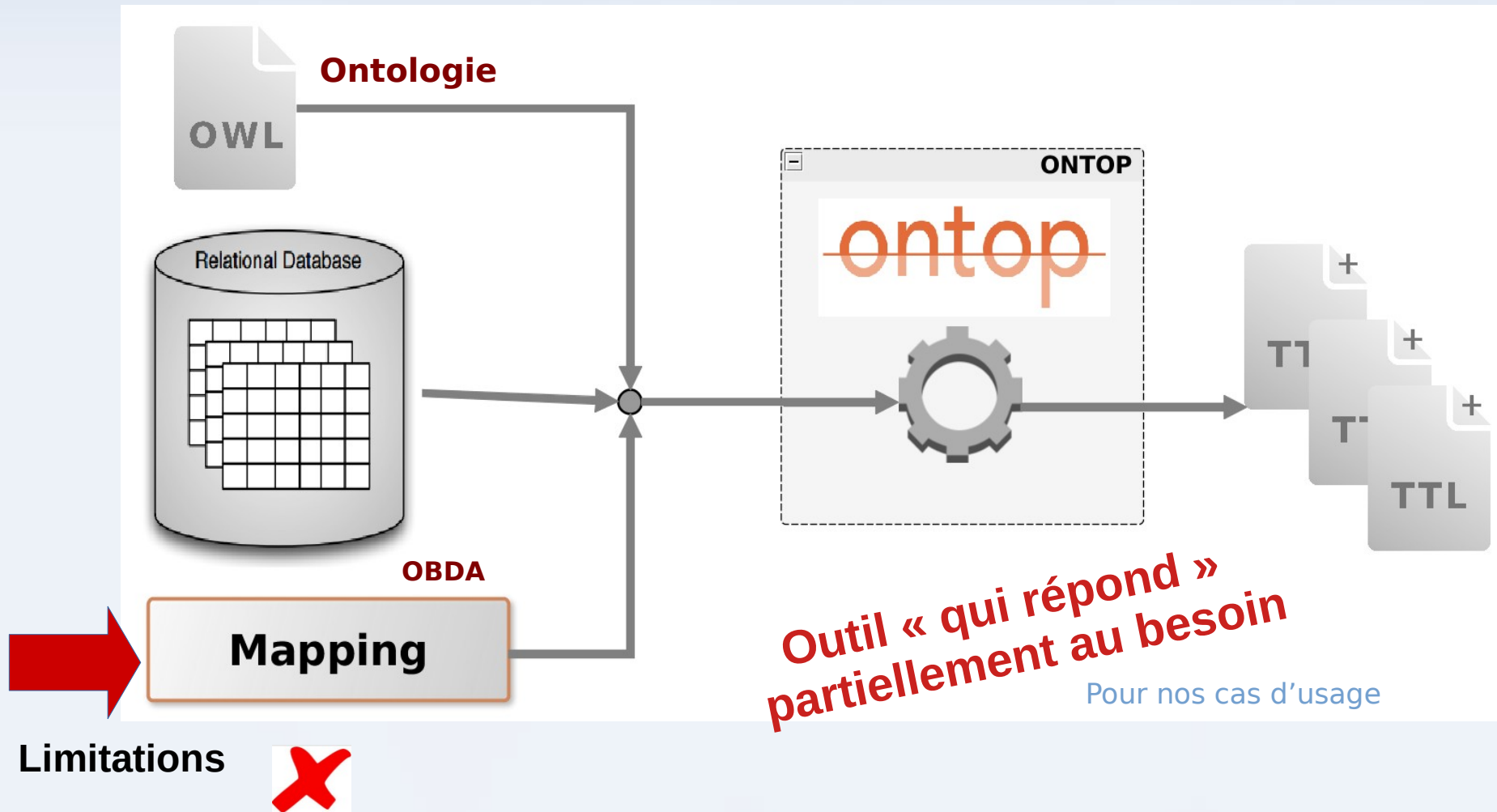
Comment les données  
relationnelles sont transformées  
en données sémantique...

Limitation...

## Inventaire des outils sémantiques ( Phase d'annotation )



# Inventaire des outils sémantiques ( Limitations ) ( Phase d'annotation )



Contournement des limitations..





## Ontop - Protegé

Ontop fourni une interface intergée à Protégé qui facilite la création des mapping.

Protegé : outil open-source pour la création et l'édition des ontologies

3 parties sont distinguées..

The screenshot displays the Ontop-Protégé interface with three main components highlighted:

- Partie DB (1):** The 'Datasource editor' window shows connection parameters for a PostgreSQL database:
  - Connection URL: `jdbc:postgresql://127.0.0.1/ola`
  - Database User: `ryahiaoui`
  - Database Password: `*****`
  - Driver class: `org.postgresql.Driver`
- Partie Target (2):** The 'Edit Mapping' window shows the 'Target (Triples Template)' for the mapping `measurement-ph-water`:
 

```

      :observation
      oboe-core:measurement/ph/water/{vmchimie_id} a oboe-core:Measurement ;
      oboe-core:hasValue {valeur} ; oboe-core:usesStandard :Ph ;
      oboe-core:ofCharacteristic :Ph .
      
```
- Partie Source (3):** The 'Edit Mapping' window also shows the 'Source (SQL Query)' for the mapping:
 

```

      SELECT
      valeur_mesure_chimie_vmchimie.valeur
      FROM
      valeur_mesure_chimie_vmchimie
      INNER JOIN variable_glacpe_varg
      ON valeur_mesure_chimie_vmchimie.var_id = variable_glacpe_varg.id
      WHERE
      variable_glacpe_varg.affichage = 'pH'
      
```

The background shows the Protégé ontology editor with a class hierarchy for 'Measurement' and a data source manager.

Créer les annotation à la main

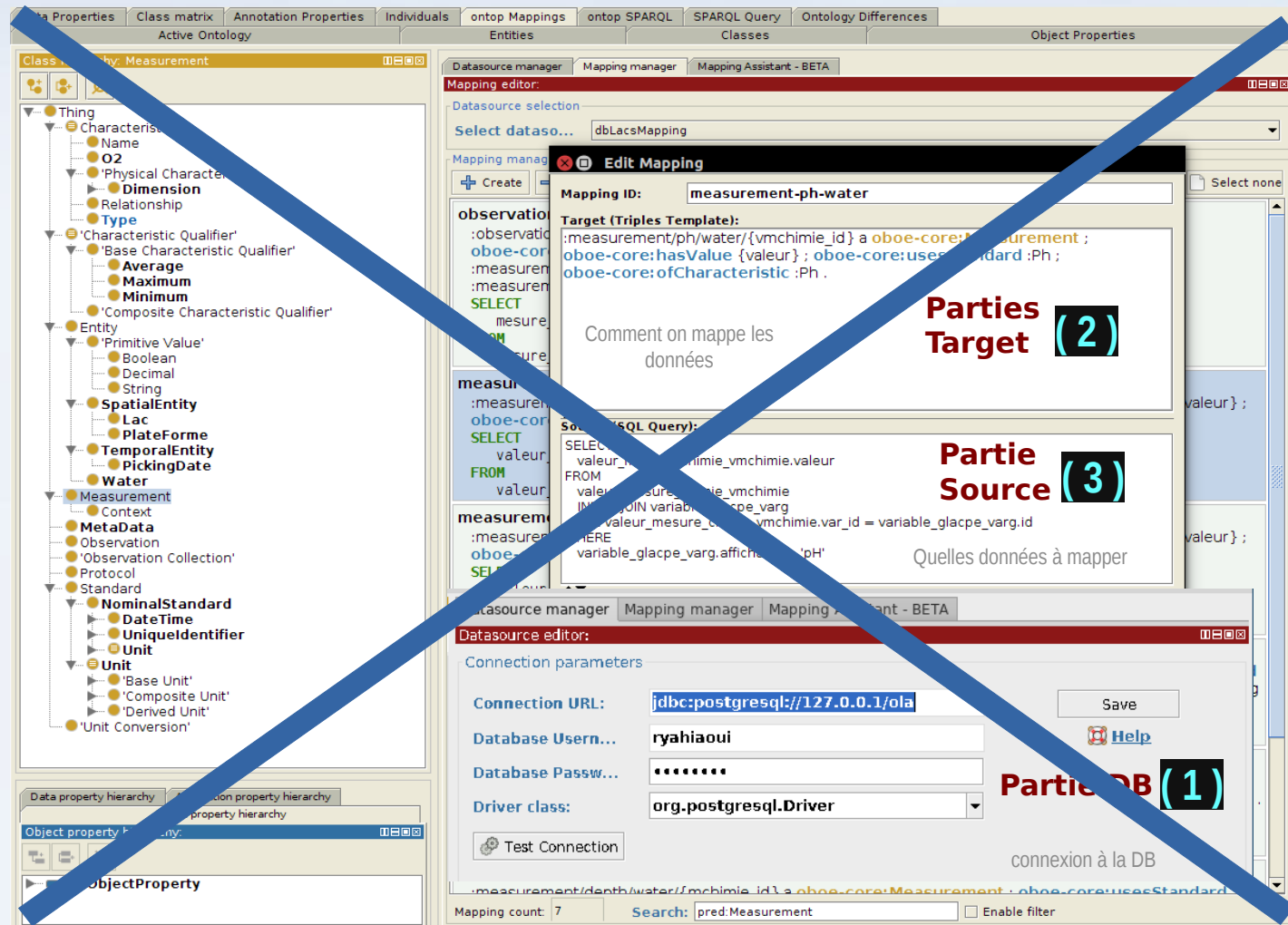


## Ontop - Protegé

Ontop fourni une interface intergée à Protégé qui facilite la création des mapping.

Protegé : outil open-source pour la création et l'édition des ontologies

3 parties sont distinguées..



Créer les annotation à la main



## En coulisse... Ontop manipule des fichiers OBDA ( basé sur le langage R2RML )

( R2RML : recommandation du W3C pour faire du mapping RDB-to-RDF )

```
[PrefixDeclaration]
rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#
oboe-core: http://ecoinformatics.org/oboe/oboe.1.0/oboe-core.owl#
oboe-temporal: http://ecoinformatics.org/oboe/oboe.1.0/oboe-temporal.owl#
xsd: http://www.w3.org/2001/XMLSchema#
: http://www.anaee.fr/ontology/anaee-france_ontology#
oboe-standard: http://ecoinformatics.org/oboe/oboe.1.0/oboe-standards.owl#
oboe-characteristics: http://ecoinformatics.org/oboe/oboe.1.0/oboe-characteristics.owl#
oboe-spatial: http://ecoinformatics.org/oboe/oboe.1.0/oboe-spatial.owl#
oboe-standards: http://ecoinformatics.org/oboe/oboe.1.0/oboe-standards.owl#
rdfs: http://www.w3.org/2000/01/rdf-schema#
```

```
[SourceDeclaration]
sourceUri dbLacsMapping
connectionUrl jdbc:postgresql://127.0.0.1/ola?sendBufferSize=5000
username ryahiaoui
password yahiaoui
driverClass org.postgresql.Driver
```

```
[MappingDeclaration] @collection [[
mappingId (52) ola characteristic depthRelativeToSurface min
target :ola/characteristic/depthRelativeToSurface/min a :DepthRelativeToSurface
oboe-core:hasQualifier :Minimum .
source SELECT id from (values ('1')) s(id) ;
]]
```

Interesting thing ..

The 3 parts that was discussed previously

how can we  
automate the  
generation of  
this kind of the  
mapping files

(1) Informations de  
connexion à la BD

(2)

Comment on mappe les  
données

Quelle donnée à mapper

### 3 Parties importantes

#### \* Partie DB

(1) Informations d'accès  
à la BD Informations

#### \* Partie Target

(2) Comment sont  
mappées les données

Règle : Graphes sont composés  
de nœuds. Chaque nœud non  
terminal est identifié par un URI

Partie Target = URI +  
Syntaxe Turtle

#### \* Partie Source

(3) Quelles sont les données  
concernées par ce mapping

Utilisation de requêtes SQL



### Exemple d'une Syntaxe Turtle ( **Partie Target** )

```
Measurement(61) a                :Measurement ;
:OfCharacteristic :Latitude
:usesStandard    :DecimalDegree
:hasValue        '10'
```



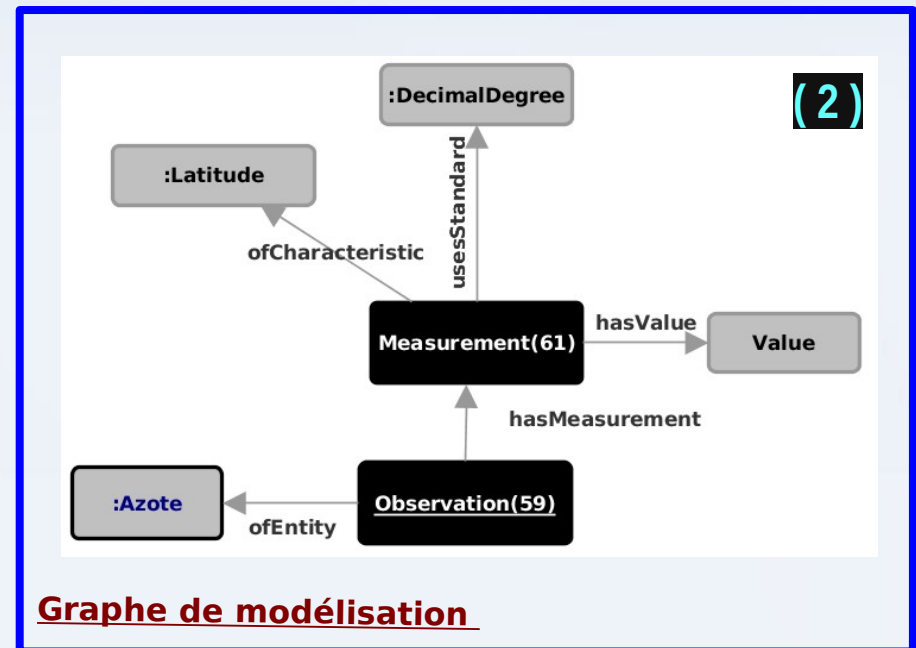
Représentation sous  
forme d'un Graphe

⇒ À Partir d'un graphe de modélisation, on peut générer ( assez simplement ) la **Partie Target** ( décrite dans les fichiers OBDA )

À condition de fournir un URI pour chaque nœud non terminal

Sachant que les graphes sont un outil simple ( car facilement manipulable par les scientifiques ) et en même temps puissants pour faire de la modélisation sémantique..

### **Partie Target**



**Pourquoi ne pas générer les fichiers de mapping ( OBDA ) à partir de ces graphes de modélisation 'sémantique' ( fournis par les scientifiques ) ??**





## Partie Target

Informations de connexion à la BD (1)

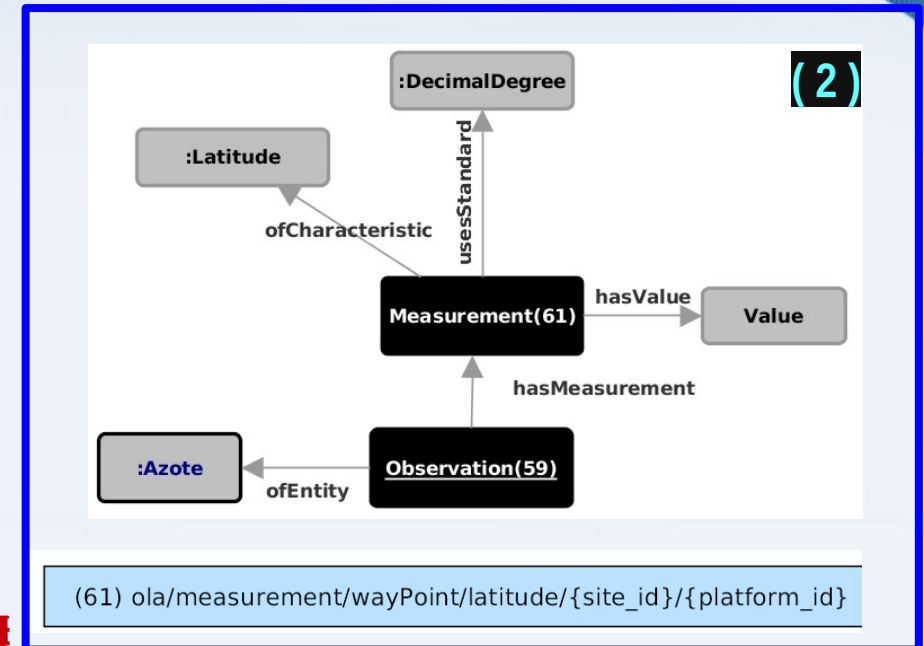
obda-sourceUri : dbLacsMapping

obda-connectionUri : jdbc:postgresql://127.0.0.1/ola?sendBufferSize=5000

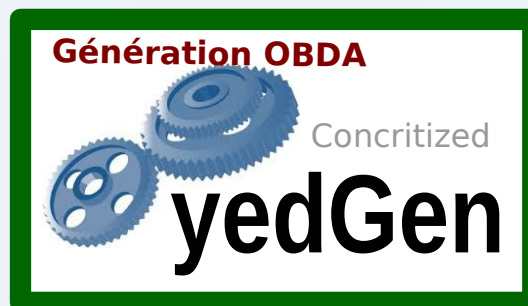
obda-username : ryahiaoui

obda-password : yahiaoui

obda-driverClass : org.postgresql.Driver



YedGen : Outil de  
génération de  
fichiers OBDA à  
partir de graphes  
de modélisation



Assigner une requête SQL pour chaque nœud non terminal

Query\_(61) : SELECT pla.loc\_id AS platform\_id, site.id AS site\_id, pla.latitude AS latitude (3)  
FROM  
public.site\_glacpe\_sit site INNER JOIN public.plateforme\_pla pla ON site.id = pla.id

## Partie Source

C'est ainsi qu'à été résolu le problème de l'automatisation..



## Généricité

Par généricité, on entend l'utilisation d'un **même graphe** pour modéliser plusieurs variables ( renseignées potentiellement dans un fichier CVS )

**Pourquoi ? Parce que ces Variables ont la même structure dans la BD**

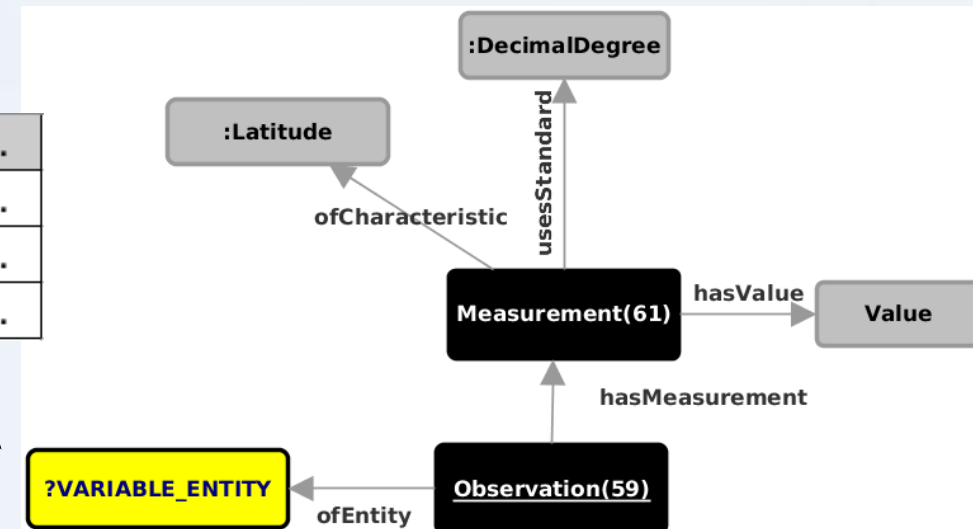
Au lieu d'avoir un graphe par variable, on aura donc un **graphe type** ( désigné pour plusieurs variables ), et à partir de ce graphe type, générer un fichier de mapping ( OBDA ) par variable

Cette généricité concerne le fonctionnement de l'outil yedGen

### Description d'un fichier CSV de variables sémantiques

	AnaEE Standar	Entity	Context	..
1	cumulative rainfall	cumulative rain		..
2	air carbon dioxide	carbon dioxyde	atmosphere,	..
3	atmospheric air sta	air	atmosphere	..

Appliquer sur la variable **VARIABLE\_ENTITY** chaque valeur de la colonne **Entity** du fichier **CSV**. Ce qui nous donne ...

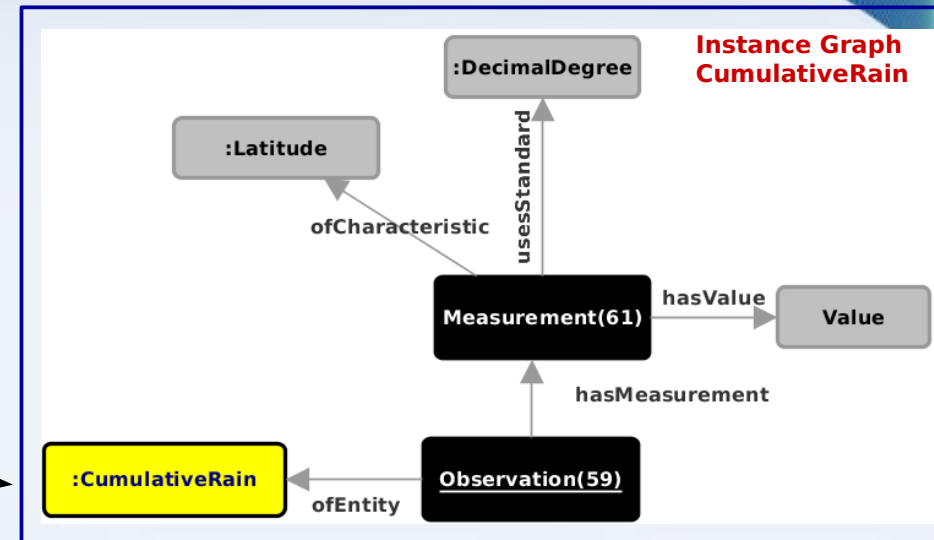




# Généricité

## Description d'un fichier CSV de variables sémantiques

AnaEE Standar	Entity	Context	..
cumulative rainfall	cumulative rain		..
air carbon dioxide	carbon dioxyde	atmosphere	..
atmospheric air sta	air	atmosphere	..

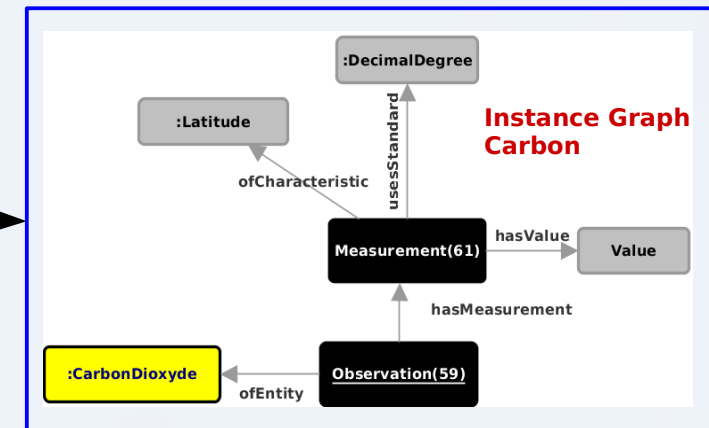




# Généricité

## Description d'un fichier CSV de variables sémantiques

AnaEE Standar	Entity	Context	..
cumulative rainfall	cumulative rain		..
air carbon dioxide	carbon dioxide	atmosphere	..
atmospheric air sta	air	atmosphere	..



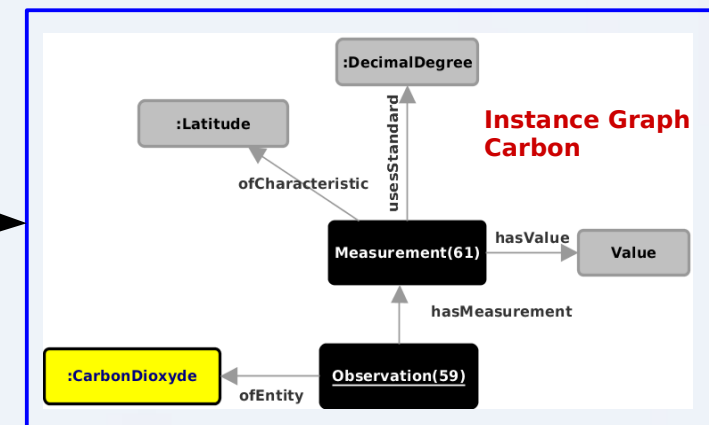
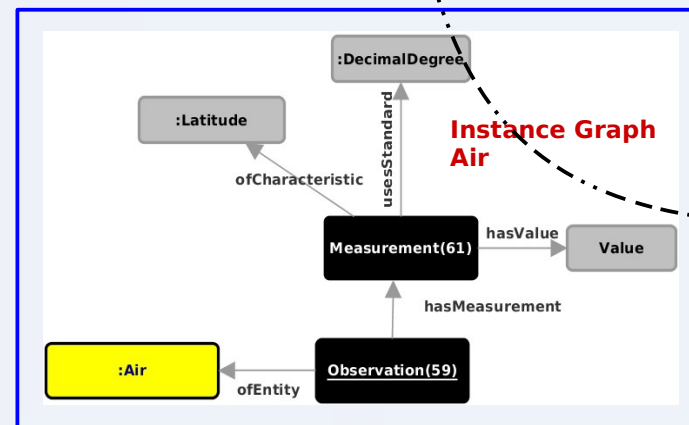
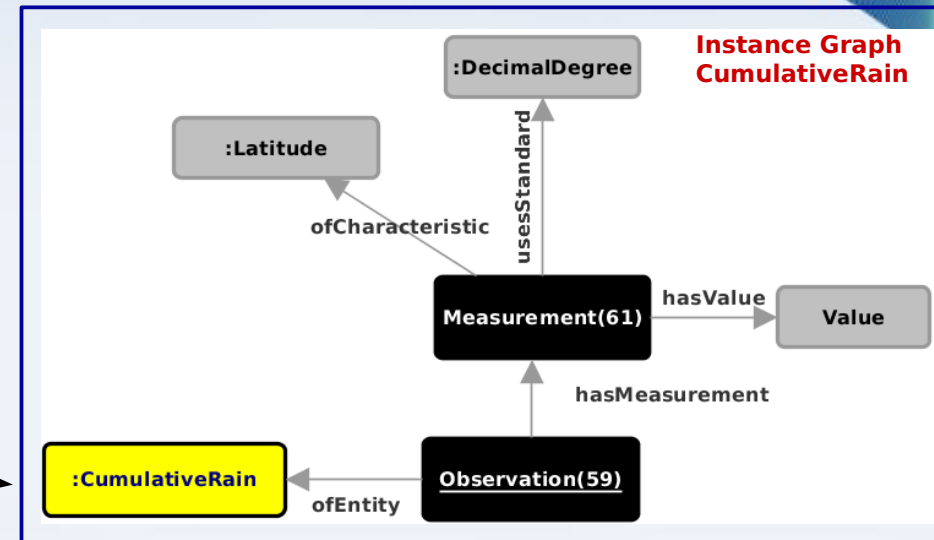




# Généricité

## Description d'un fichier CSV de variables sémantiques

AnaEE Standar	Entity	Context	..
cumulative rainfall	cumulative rain		..
air carbon dioxide	carbon dioxyde	atmosphere	..
atmospheric air sta	air	atmosphere	..



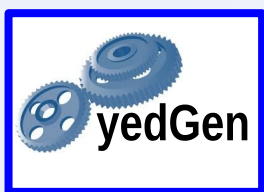


# Généricité

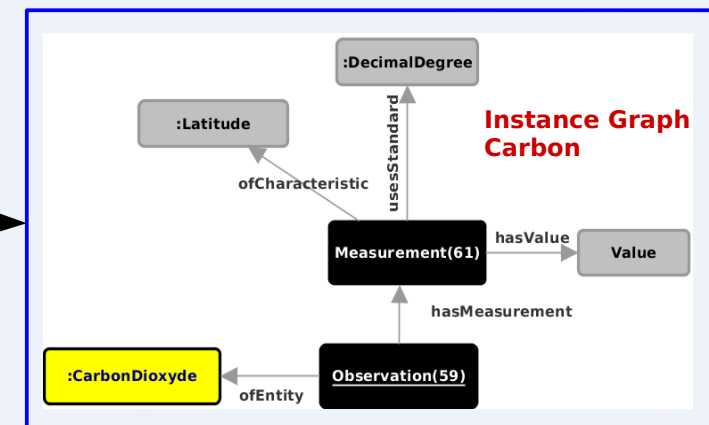
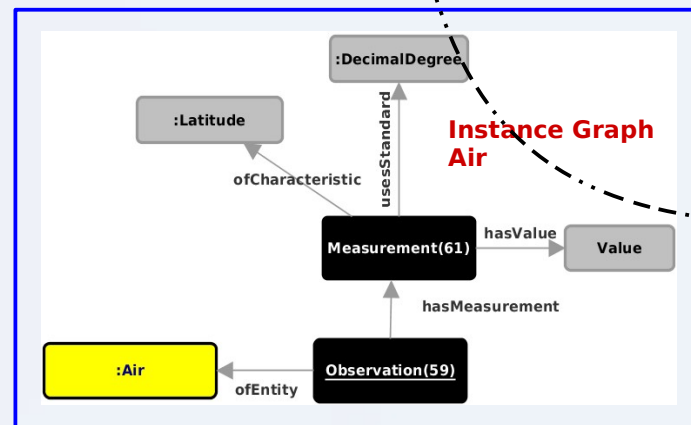
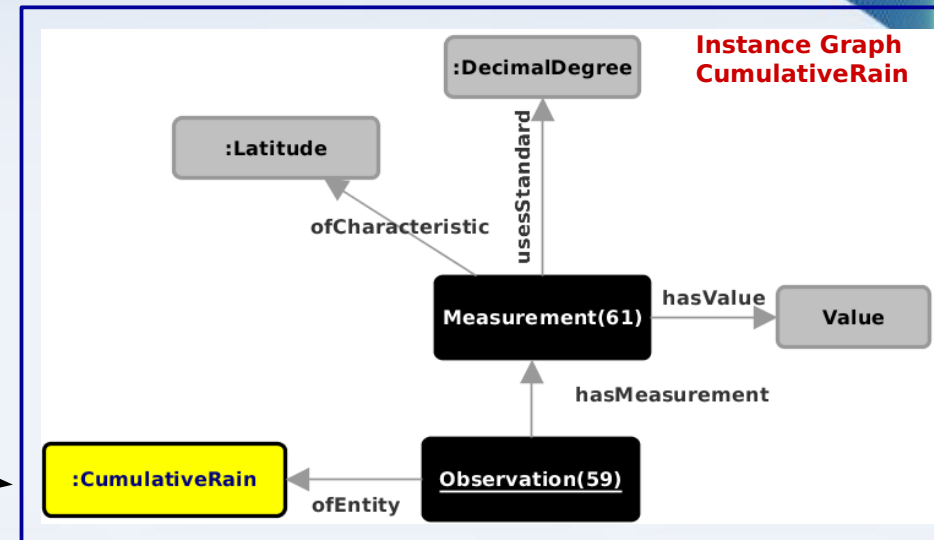
## Description d'un fichier CSV de variables sémantiques

AnaEE Standar	Entity	Context	..
cumulative rainfall	cumulative rain		..
air carbon dioxide	carbon dioxide	atmosphere	..
atmospheric air sta	air	atmosphere	..

Le même process est répété pour chaque ligne du CSV..



C'est ainsi qu'à été approché la problématique de la généricité

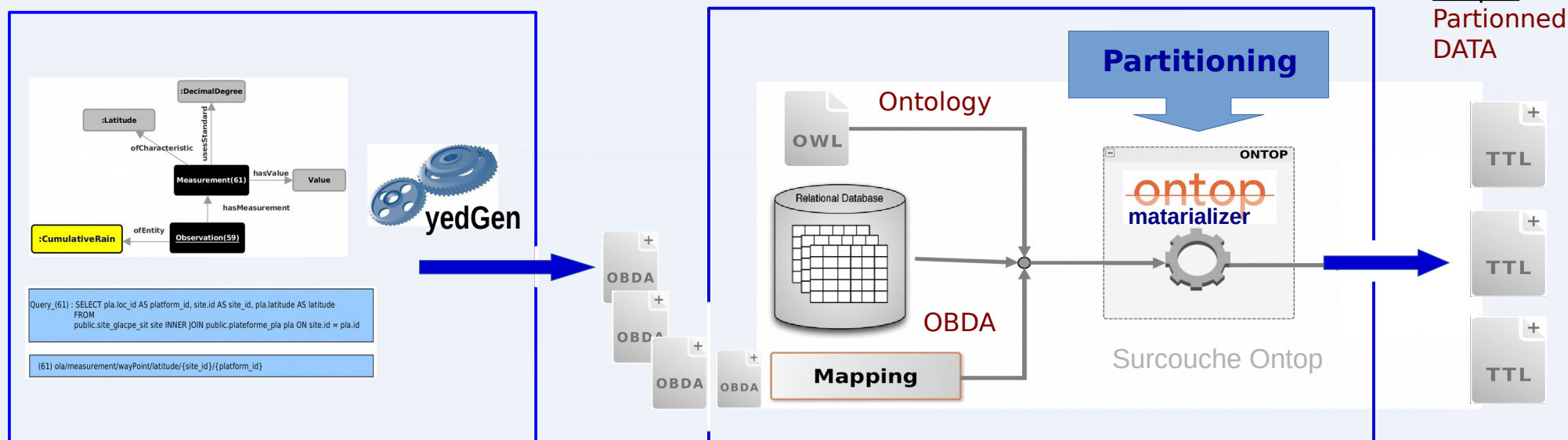




## Gros Fichiers / Grosse Bases de données

Il arrive parfois que le volume de données traité par **ONTOP** et **BlazeGraph** dépasse la capacité mémoire de la machine, dans ce cas, on est confronté à des **Outofmemoryerrors**

Solution : **Volume data Partitioning** → Traitement des données par chunk ( LIMIT/OFFSET )



→ Traiter un volume « infini » de données



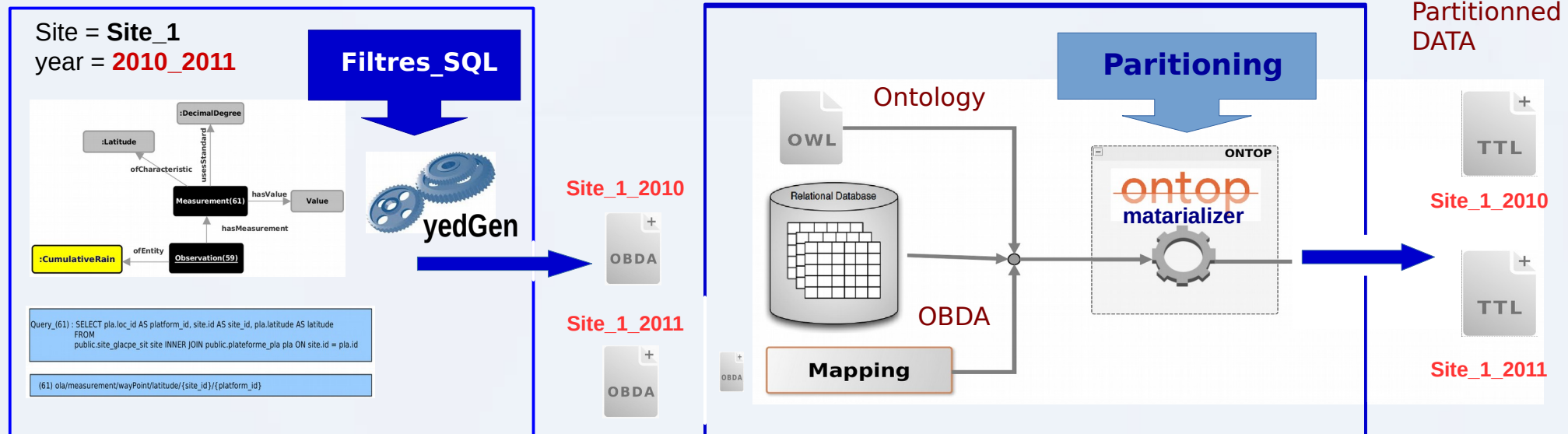
## Filtre sur les données

Pour certains use cases, on a besoin de n'extraire que la donnée dont l'utilisateur a besoin

Solution : **Logical data partitioning**

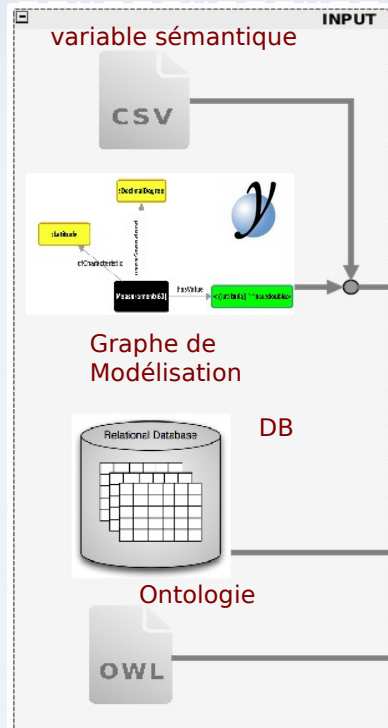
**Exemple** : Générer des données spécifiques à une **variable** particulière, pour un **site** particulier et un intervalle **d'années** précis

Plus vous filtrez les données, moins vous en avez, plus vous êtes performant

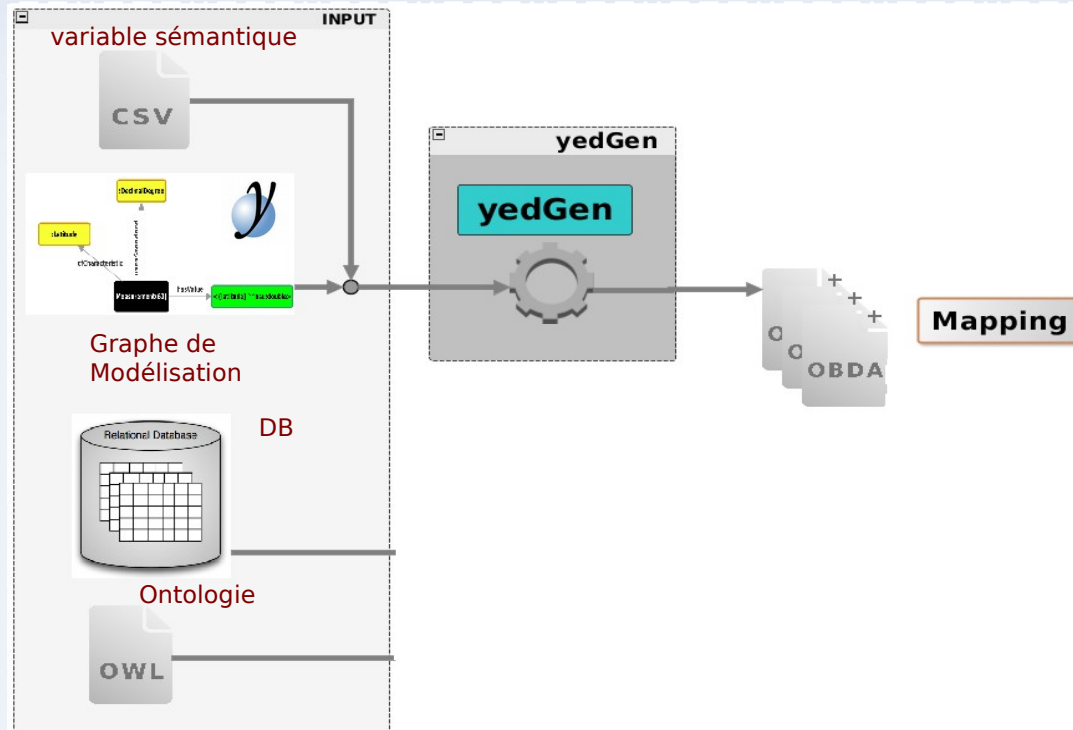


**Remarque** : Volume data partitioning & Logical data partitioning peuvent être **combinés**



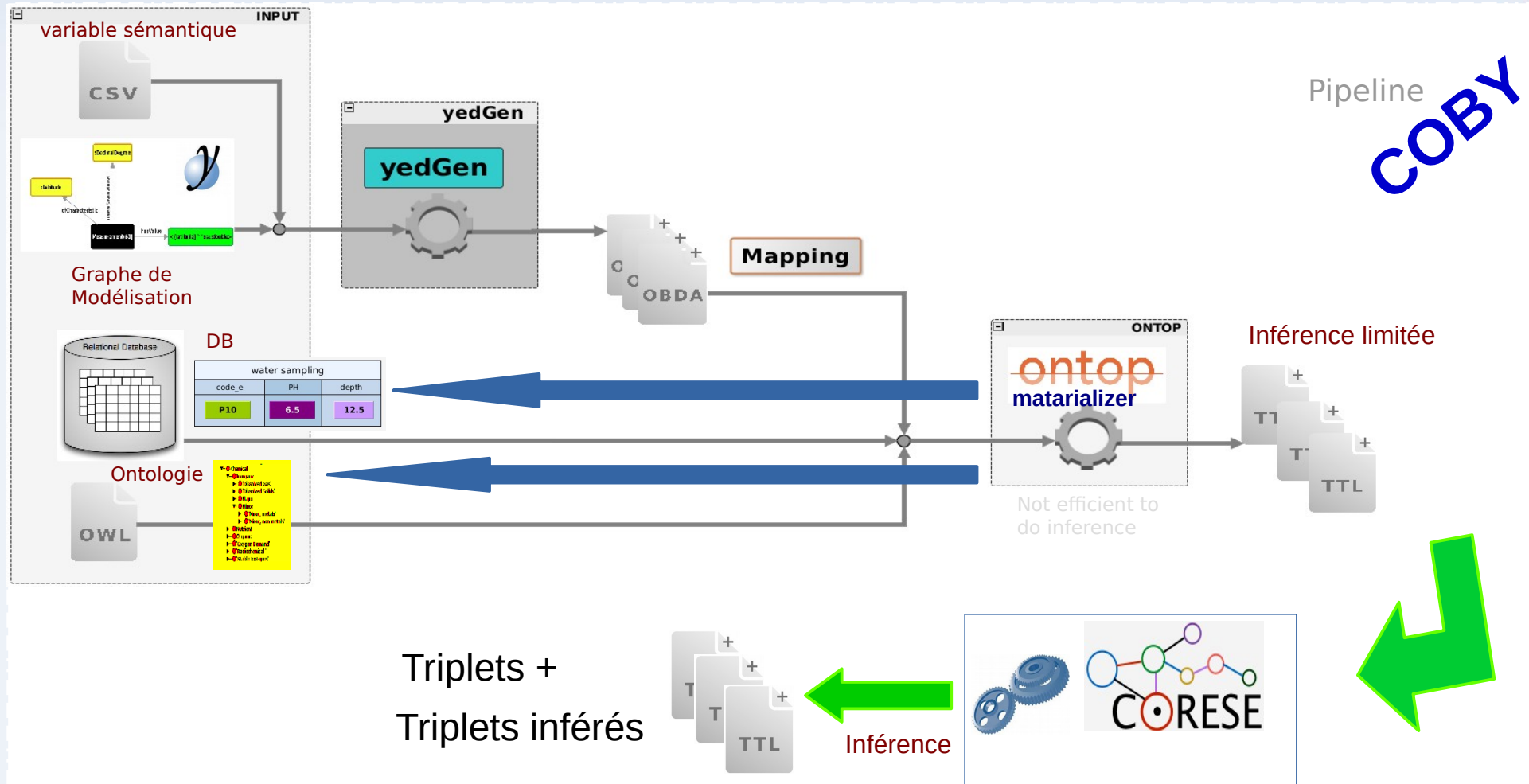


Pipeline



Pipeline



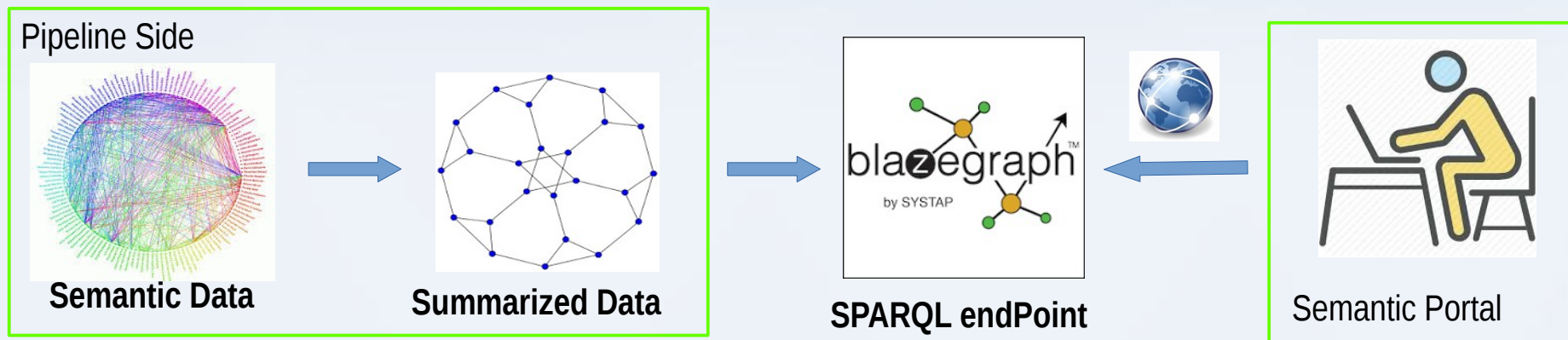




## 1 – Données de synthèse ( sémantiques ) pour le portail Anaee-F

Use Case 1

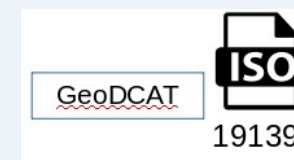
Portal side



**Objectif :** Production de données de synthèse sémantique par le pipeline d'annotation, et publication de ces dernières sur un Sparql-Endpoint directement accessible par le Portail AnaEE-F

## 2 - Production de données + métadonnées

Use Case 2



**Objectif :** Production de données sémantiques filtrées ( au format N-Triples ), qui seront utilisées pour produire des fichiers au format **netCDF** ( jeu de donnée qui descend au grain de la donnée ) / Format **GeoDCAT** ( sémantique ) auquel on applique du XSLT pour le transformer en **ISO-19139**

- **yedGen : Génération instantanée des fichiers de mapping ( OBDA ) à partir des graphes de modélisation**  
⇒ **Passage instantané : Modélisation → Fichiers d'Annotation**
- **Image Docker pré-configurée du pipeline & Déployable en un clique ( en fournissant les graphes + CSV )**
- **Généricité \***
- **Métriques :**

**( Machine test : 17 / 8 Cores / 5 Go Heap / HDD )**

- |                                      |   |                  |               |
|--------------------------------------|---|------------------|---------------|
| * <b>Génération</b> ( Ontop )        | ~ | <b>700.000</b>   | triplets / mn |
| * <b>Inférence</b> ( Moteur Corese ) | ~ | <b>2.600.000</b> | triplets / mn |
| * <b>Chargement</b> ( BlazeGraph )   | ~ | <b>3.000.000</b> | triplets / mn |

## **\*\* À l'échelle des SOERE :**

Modélisation de nouveaux types de données

⇒ Consiste à la création de nouveaux modèles d'annotations pour les variables stockées en base de données en utilisant l'outil **Yed Graph Editor** pour les graphes

## **\*\* À l'échelle du Pipeline :**

- Augmenter les performances en introduisant du traitement distribué ⇒  
[ Technologie Docker\*\* ] → Un Fichier OBDA ( Mapping ) par conteneur Docker )
- Développement d'un PSL ( Pipeline Specific Langage ) ⇒  
Simplification d'écriture des Orchestrateurs ( use cases – écrit actuellement en Bash ! )
- Autres pistes : **Apache RYA** !

**Docker** : Technologie de « conteneurisation »

MERCI DE VOTRE  
ATTENTION

