



# **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

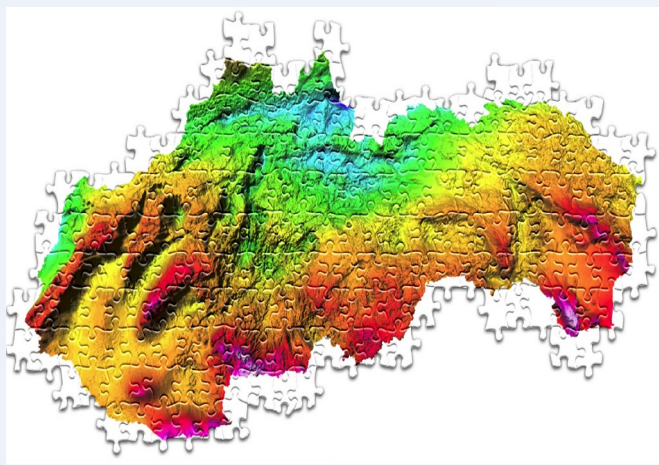


*R. Yahiaoui  
P. Christian  
D. Maurice  
A. Schellenberger*



# Intro

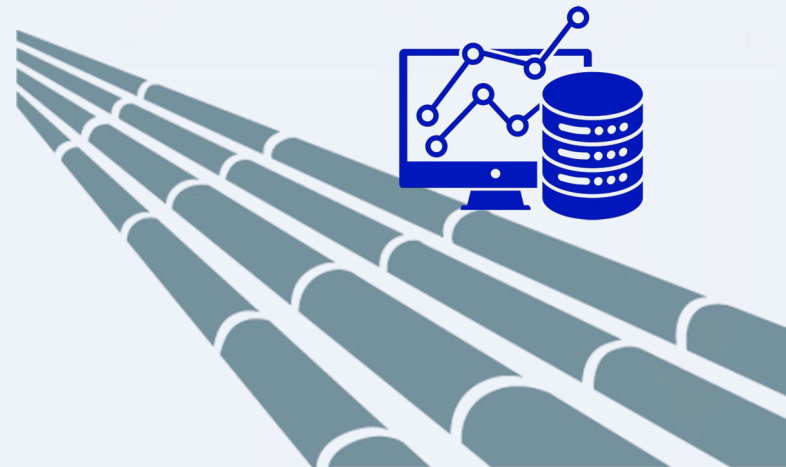
- Infrastructure nationale "Analyse et Expérimentation sur les Écosystèmes"
- Offre à la communauté scientifique des plateformes d'expérimentation, de modélisation et des BDD ( dont celles des SOERE )
- Mettre en place une **interopérabilité** basée sur les technos du **web sémantique**



Hétérogénéité



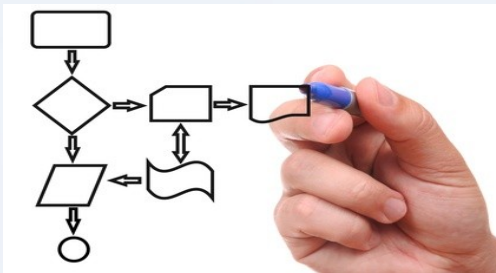
Défi



Automatisation

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

# Processus d'annotation et production de données sémantiques



Définitions



Architecture Générale



Démarche suivie



Aspects Fonctionnels

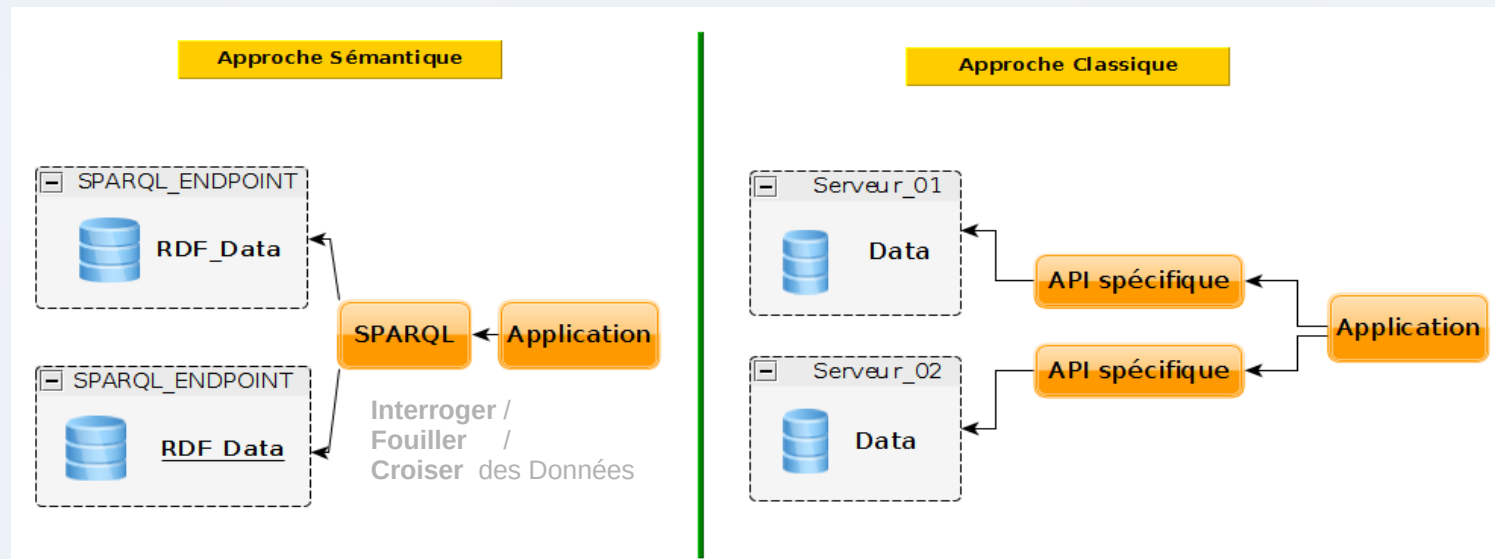


Algorithme / Code





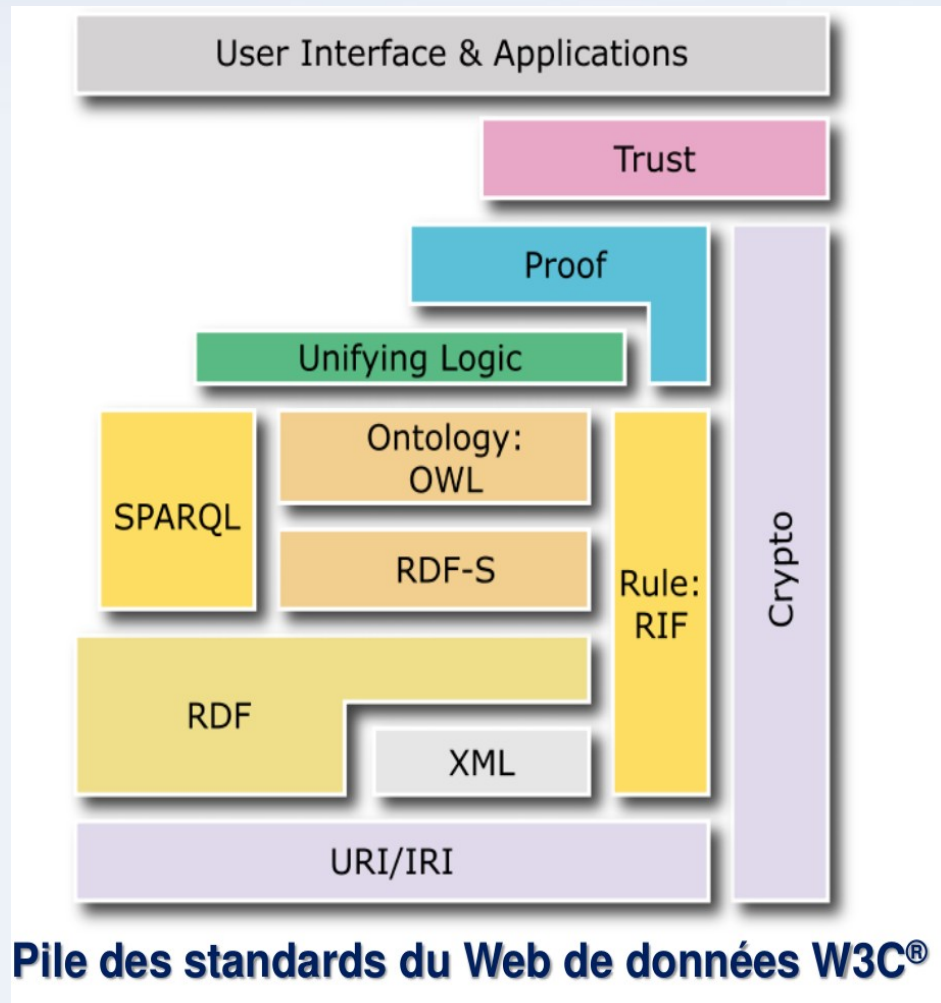
Le **Web sémantique**, ou toile sémantique, est un mouvement collaboratif mené par le World Wide Web Consortium (W3C) qui [favorise des méthodes communes pour échanger des données sur Internet pour accéder simplement..](#) ( Wikipedia )







## 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

**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

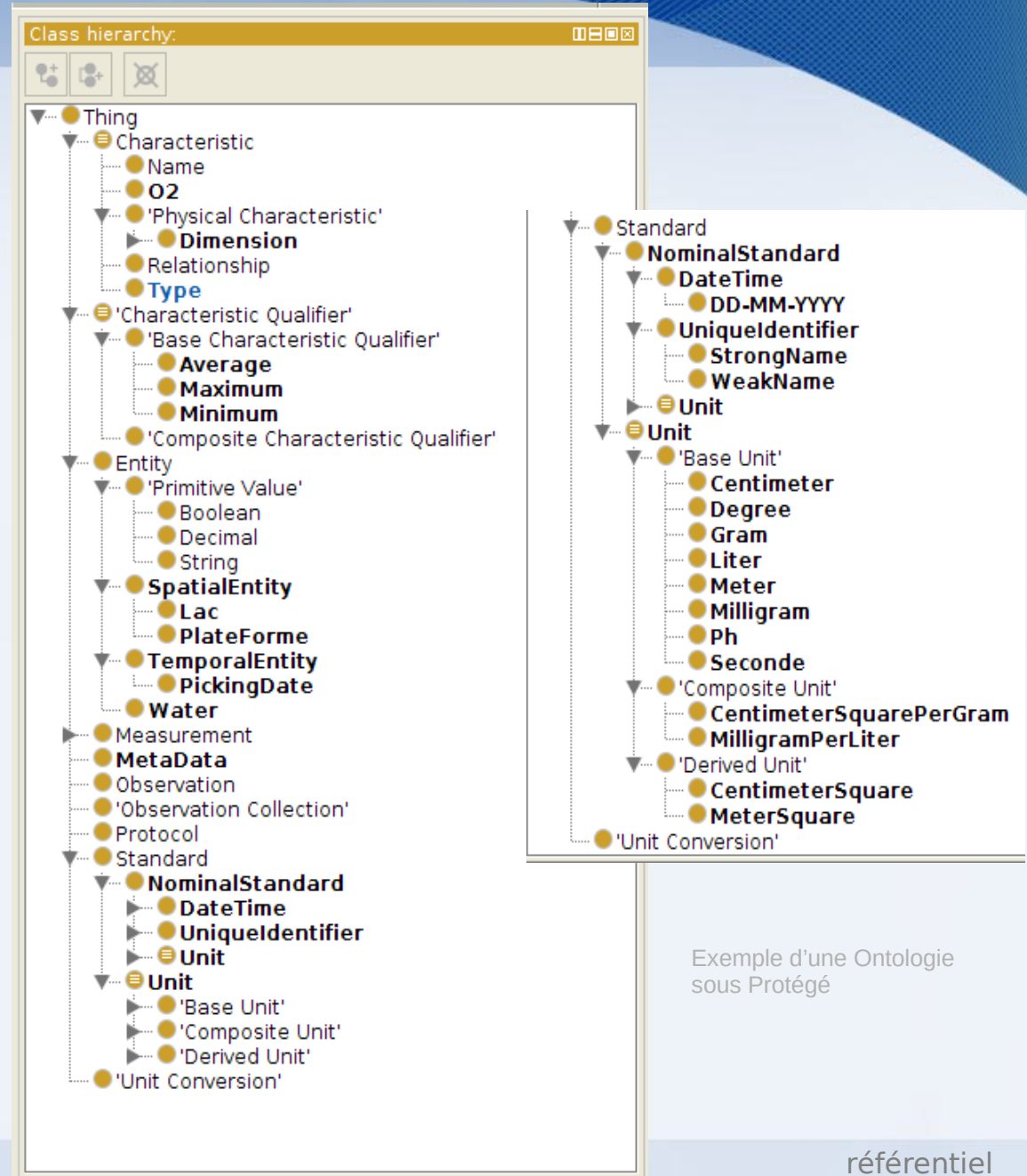
## Ontologie

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....

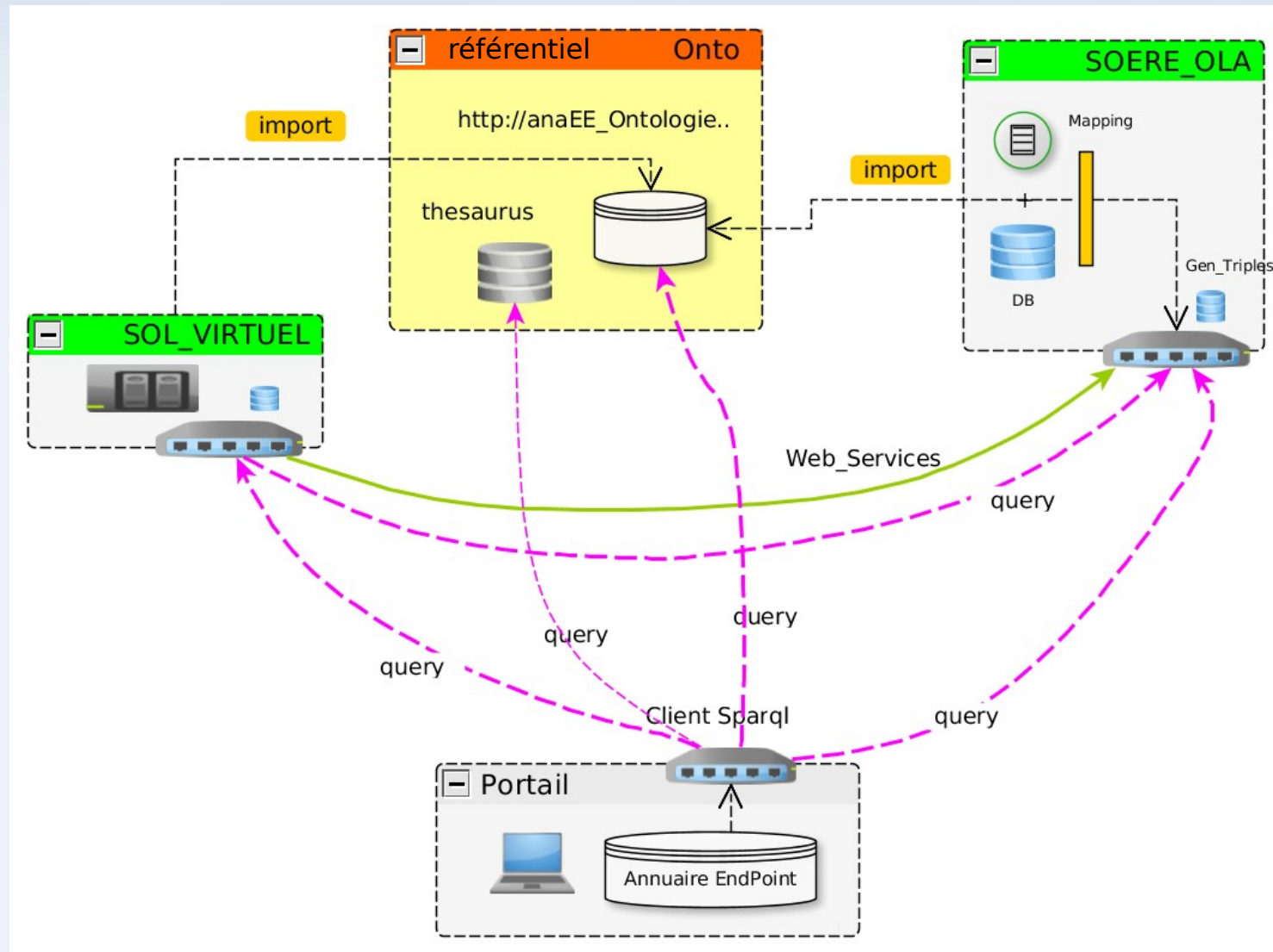
## Thésaurus

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



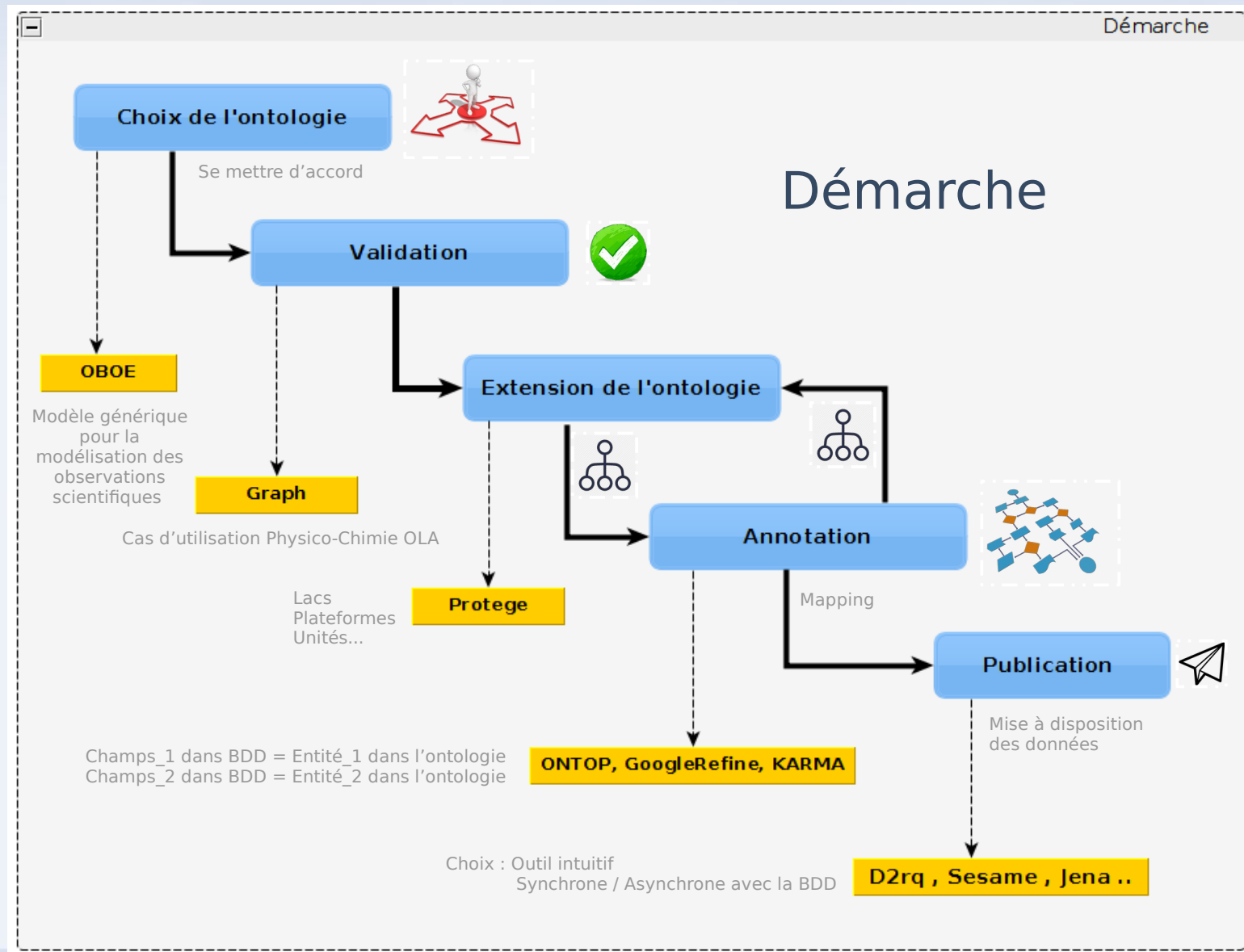
Exemple d'une Ontologie sous Protégé

# Schéma d'architecture Générale



Pour faire cela, chaque S.I s'est appuyé une démarche particulière

# 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

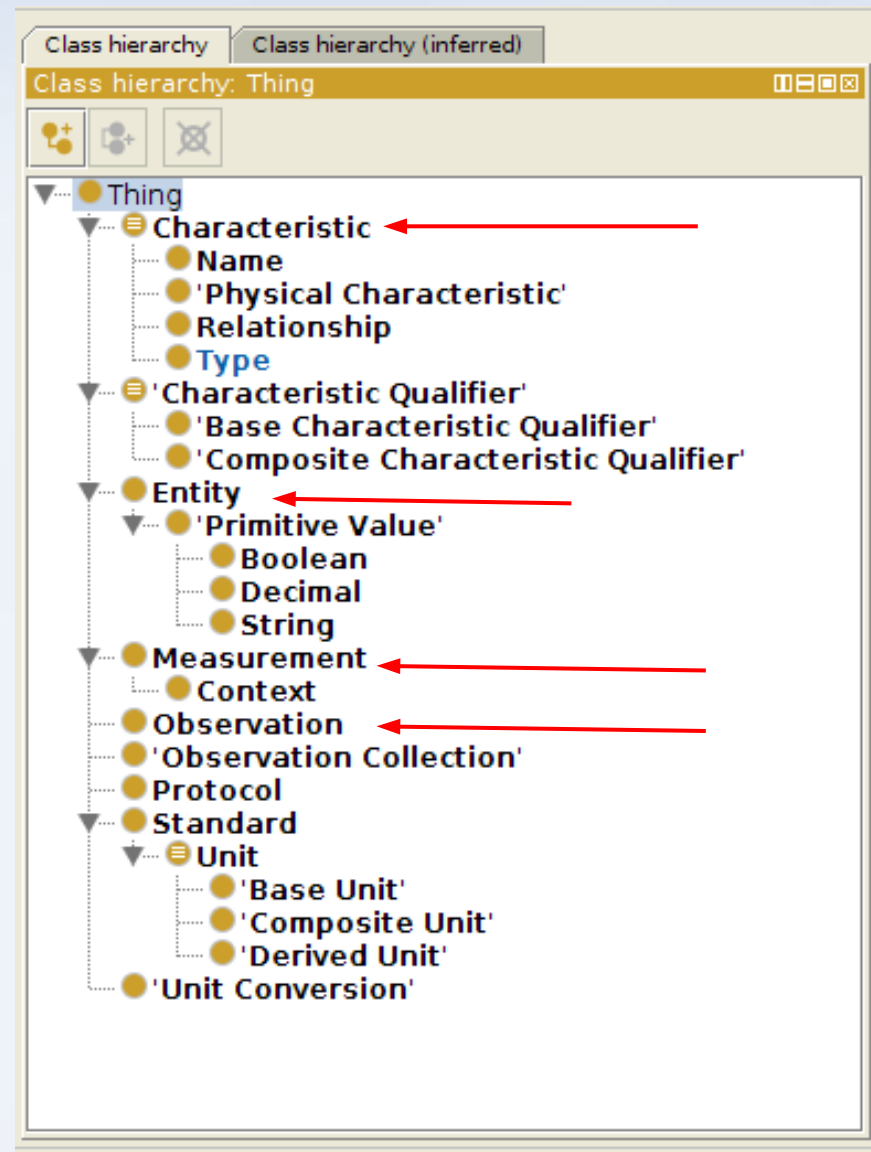
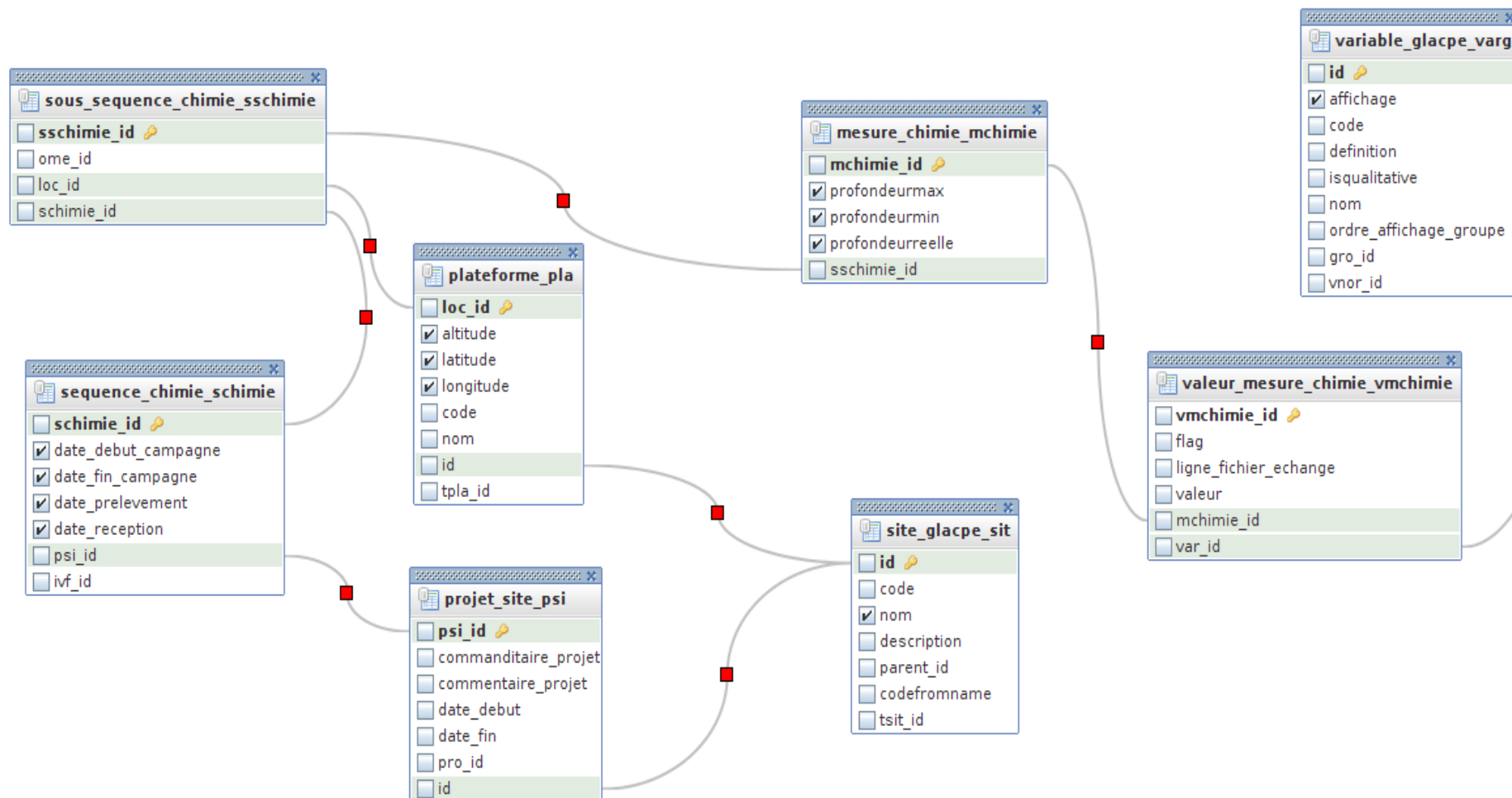


Schéma : Cas **Physico-Chimie** du SOERE **OLA**

## Données SOERE OLA

	A	D	E	F	G	H	I	J	K	L
1										
2		MG/L	MG/L	MG/L	MG/L	MG/L	MG/L	MG/L	MG/L	MG/L
3		Na	K	SO4	Cl	Al	Ba	Fe	Li	M
28	BKY	1,7	1,0	9,30	2,64	0,990	0,0341	1,130	0,0030	0,020
29	BKZ	232,0	10,8	68,70	404,00	0,110	0,0303	0,644	0,0140	0,149
30	BKAA	3,3	0,3	2,20	12,20	0,020	0,0181	0,230	0,0005	0,020
31	BKAB	2,7	0,7	0,10	8,46	0,010	0,0292	0,068	0,0010	0,000
32	BKAC	22,1	3,1	13,40	72,80	1,230	0,1250	4,940	0,0090	0,220
33	BKAD	1,9	0,7	34,40	3,91	0,220	0,0159	0,508	0,0020	0,000
34	BKAE	3,5	0,5	32,90	17,10	0,010	0,0139	0,049	0,0005	0,000
35		0,9	0,3	5,00	2,52	0,005	0,0126	0,034	0,0010	0,000
36	BKAG	1,1	0,2	9,40	2,35	0,010	0,0103	0,075	0,0005	0,000
37	BKAH	0,4	0,3	1,60	0,84	0,010	0,0072	0,054	0,0005	0,020
38	BKAI	0,5	0,4	4,70	1,22	0,170	0,0061	0,304	0,0010	0,020
39	BKAI	1,1	0,1	26,00	1,00	0,030	0,0206	0,080	0,0010	0,000

Site

Observation

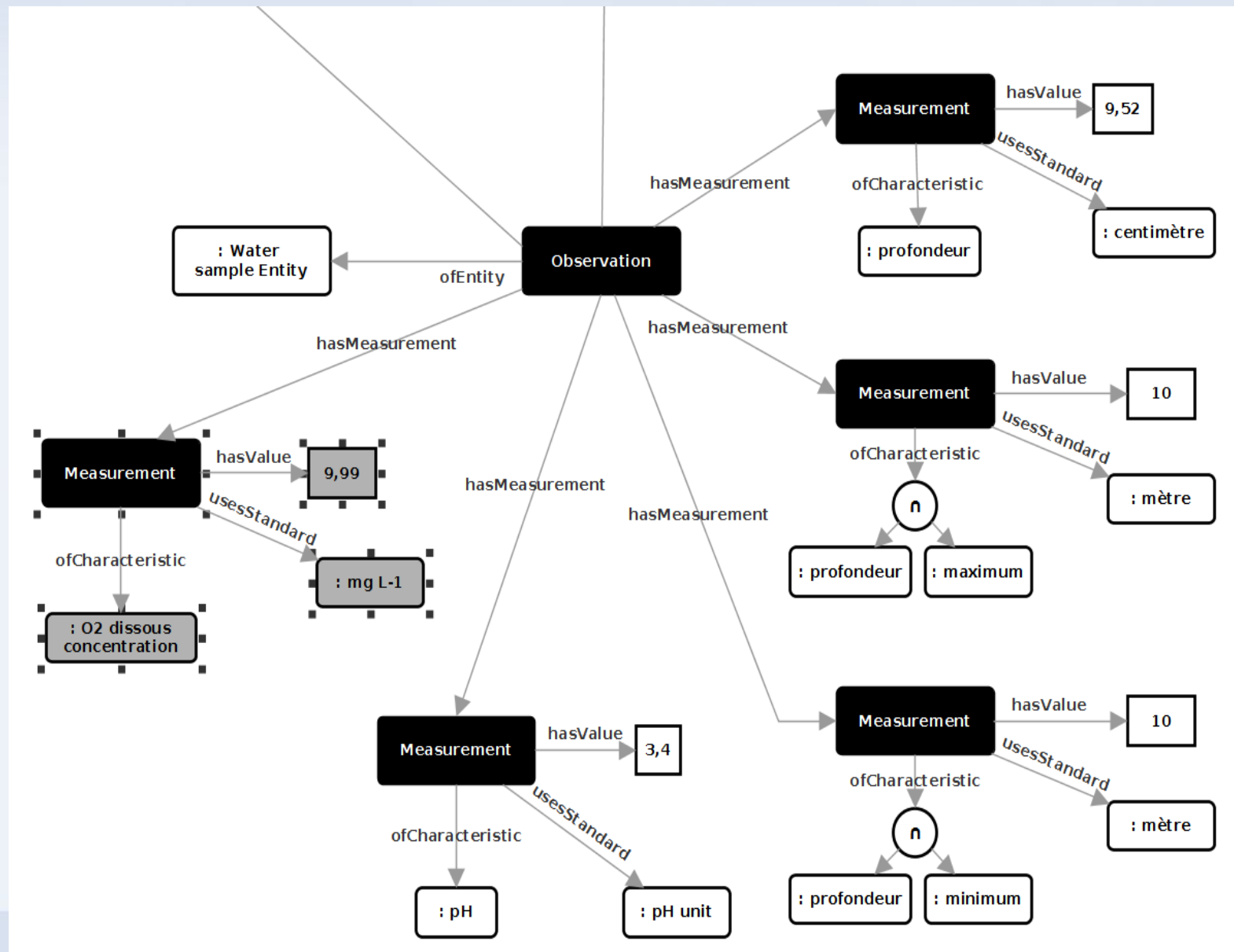
	AA	AB	AC	AD	AE	AF	
		uS	Celsius	m asl			L=Lake
	pH	COND	TEMP	ELEV	LAT	LONG	P=Pond
50	7,6	48,0	6,0	2	74 30.69N	121 41.08W	L
50	8,1	1160,0	8,0	0	74 27.82N	122 34.55W	P
0	7,6	83,0	5,0	8	74 21.46N	124 33.92W	P
0	8,1	89,0	7,0	20	74 08.10N	124 12.52W	P
0	8,4	333,0	8,0	0	72 21.13N	125 24.43W	P
0	7,8	137,0	3,0	122	71 43.79N	123 28.94W	L
00	8,4	216,0	7,5	169			
50	7,8	109,0	3,5	175			
00	7,9	105,0	8,0	105	72 39.96N	119 56.11W	P
0	7,7	41,0	3,0	131	73 35.57N	119 35.01W	L
00	7,9	65,0	4,0	137	73 20.79N	116 46.23W	L
0	8,5	137,0	7,0	195	73 29.11N	115 41.15W	P

PH

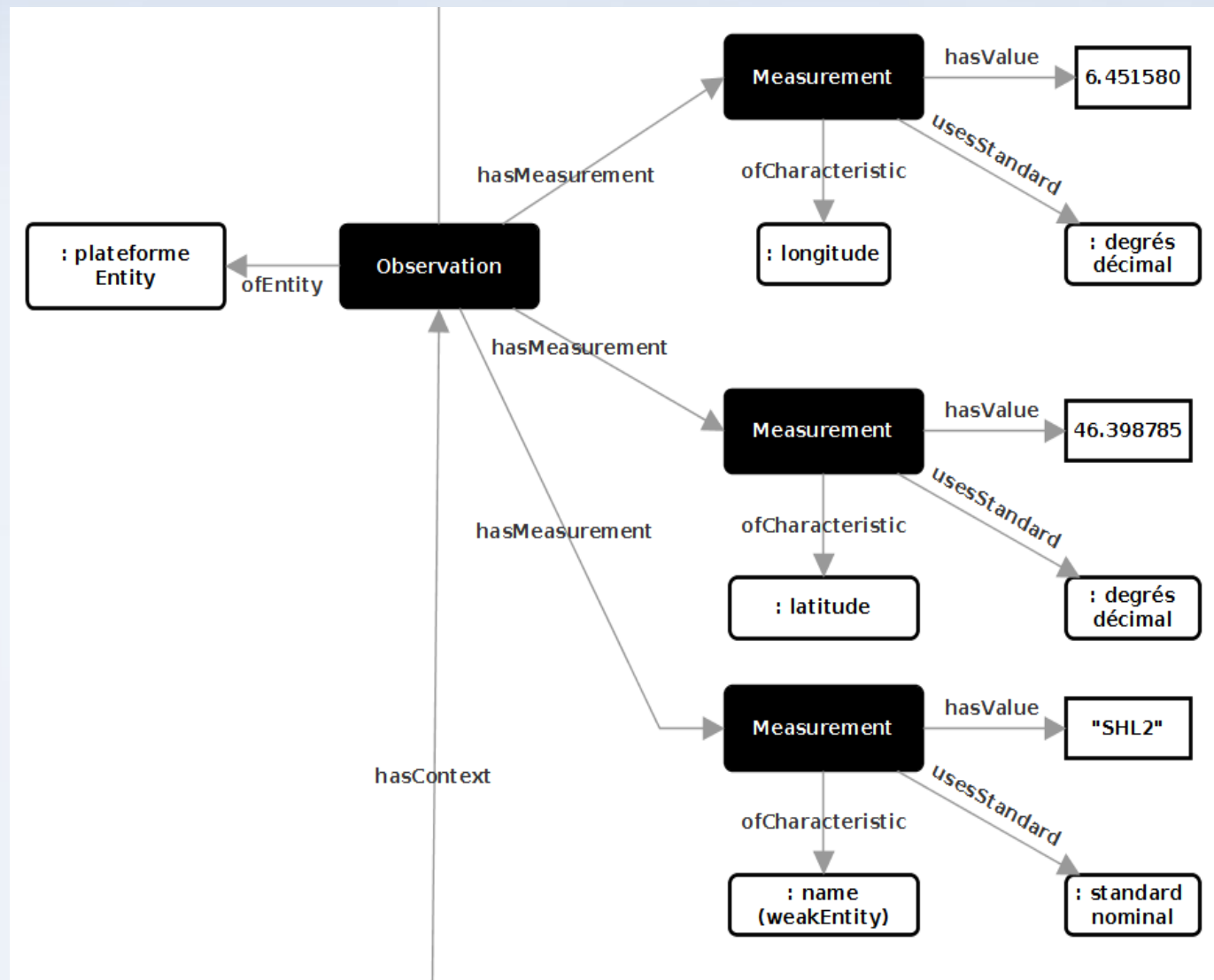
Measurement

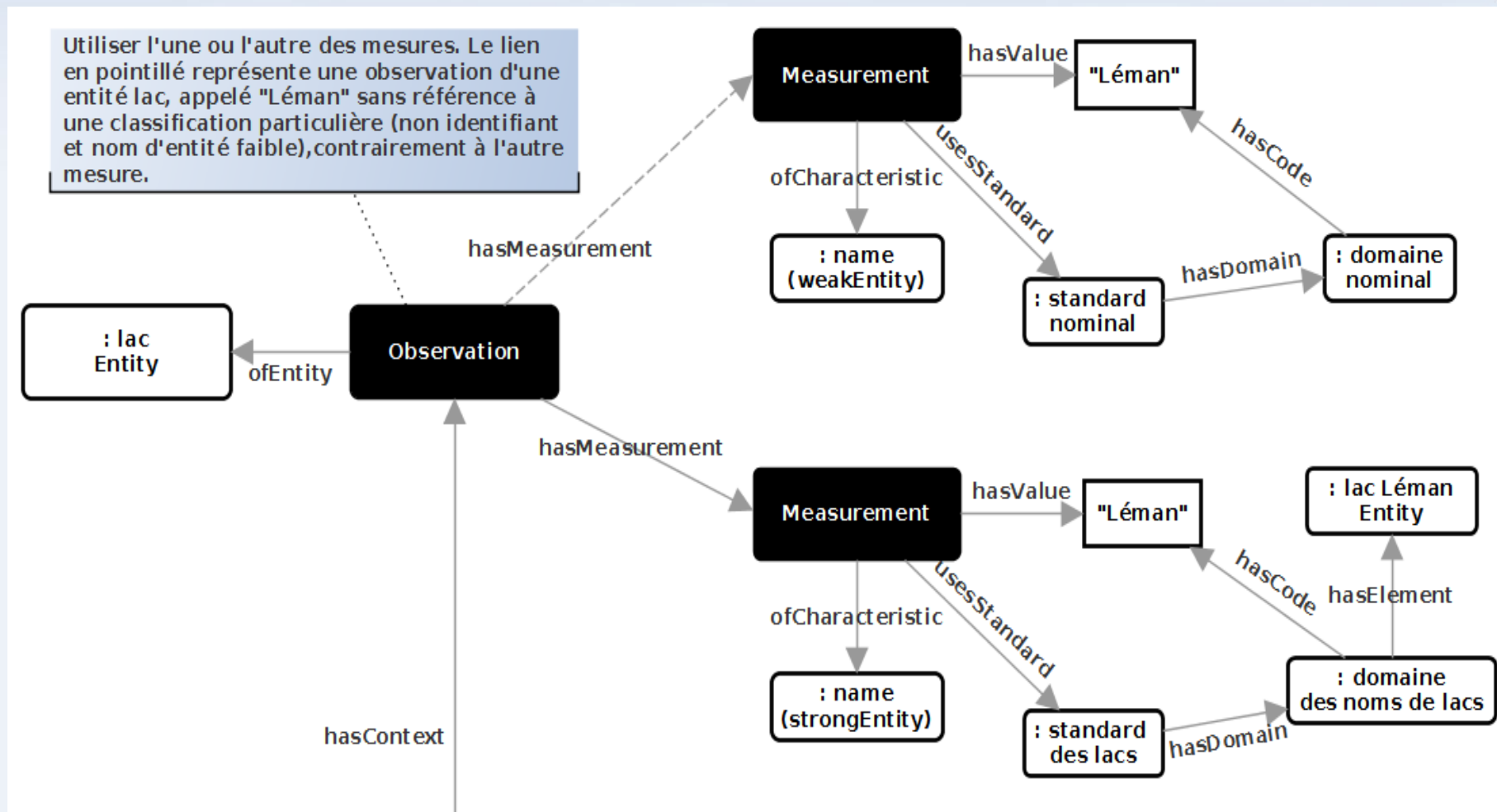
Location

pas suffisant.. !









Validation

## Extension OBOE

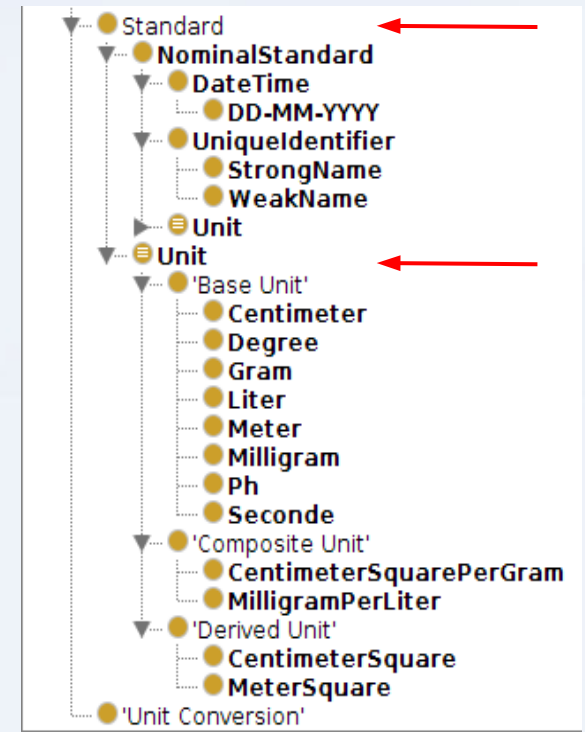
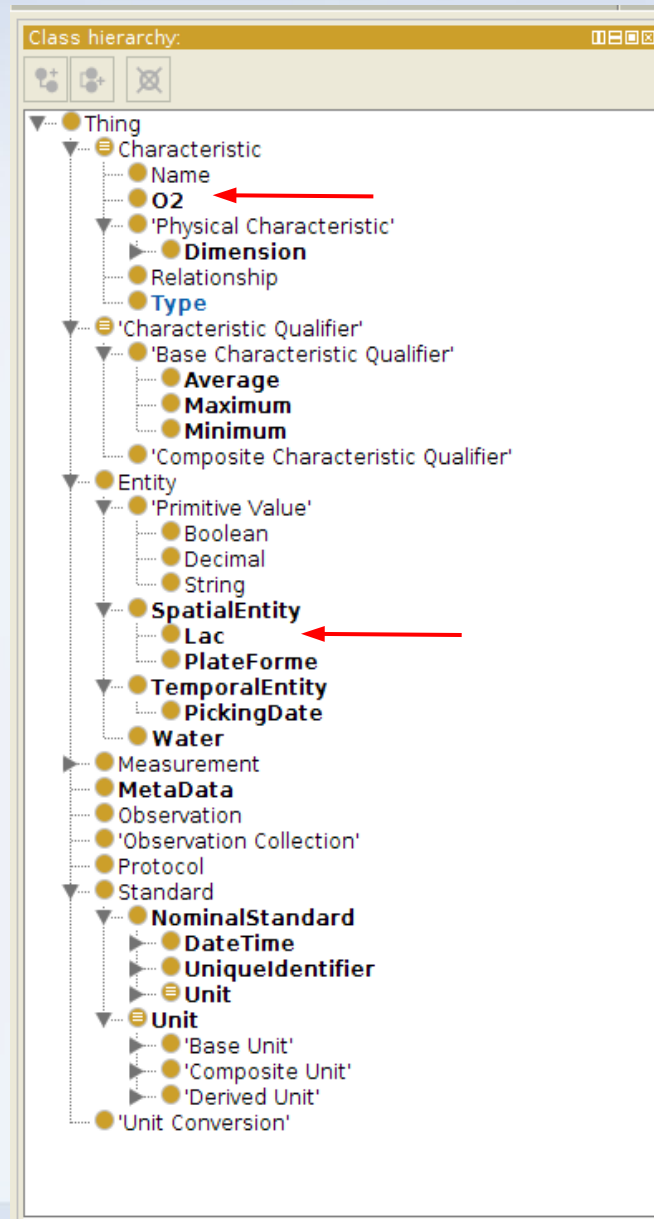
OBOE-CORE

+

' Thésaurus '

=

Ontologie AnaEE-F



Data		
Nom	employeur	HomePage
MichelGagnon	poly	http://www.professeurs.polymtl.ca/michel.gagnon



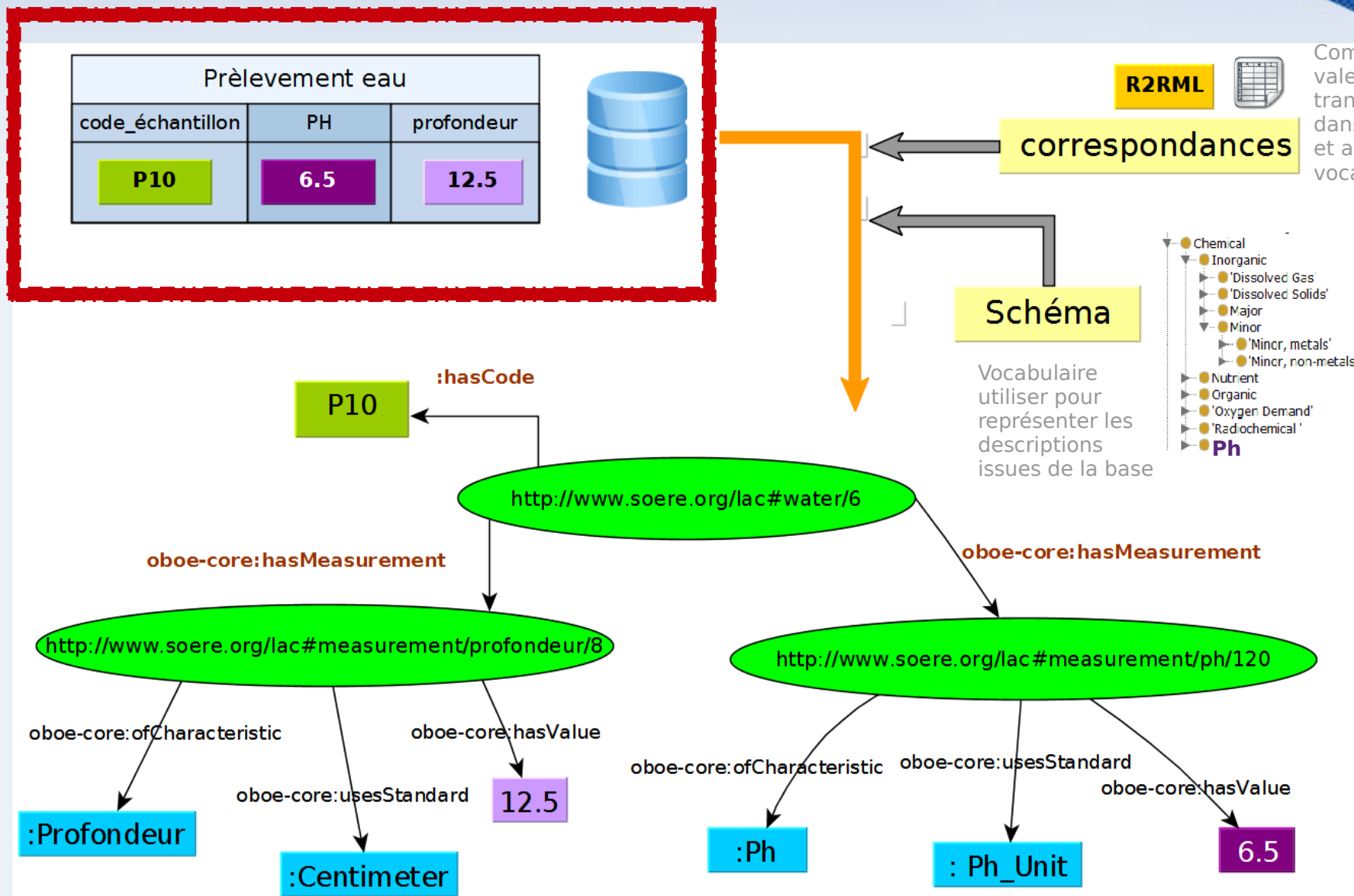
N-Triples

```
#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

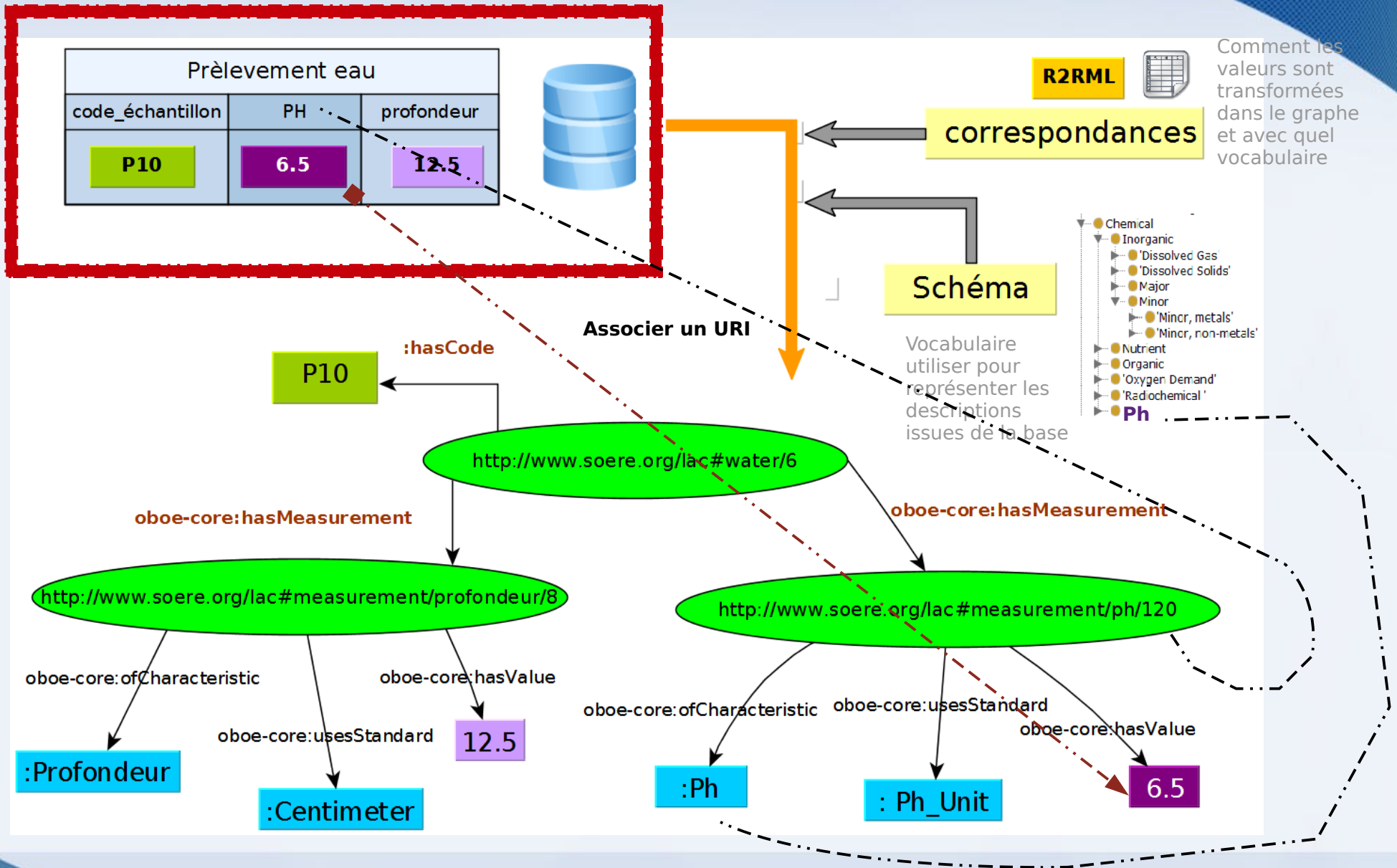


# Annotation [ Transformation RDB - RDF ( 2/2 ) ]

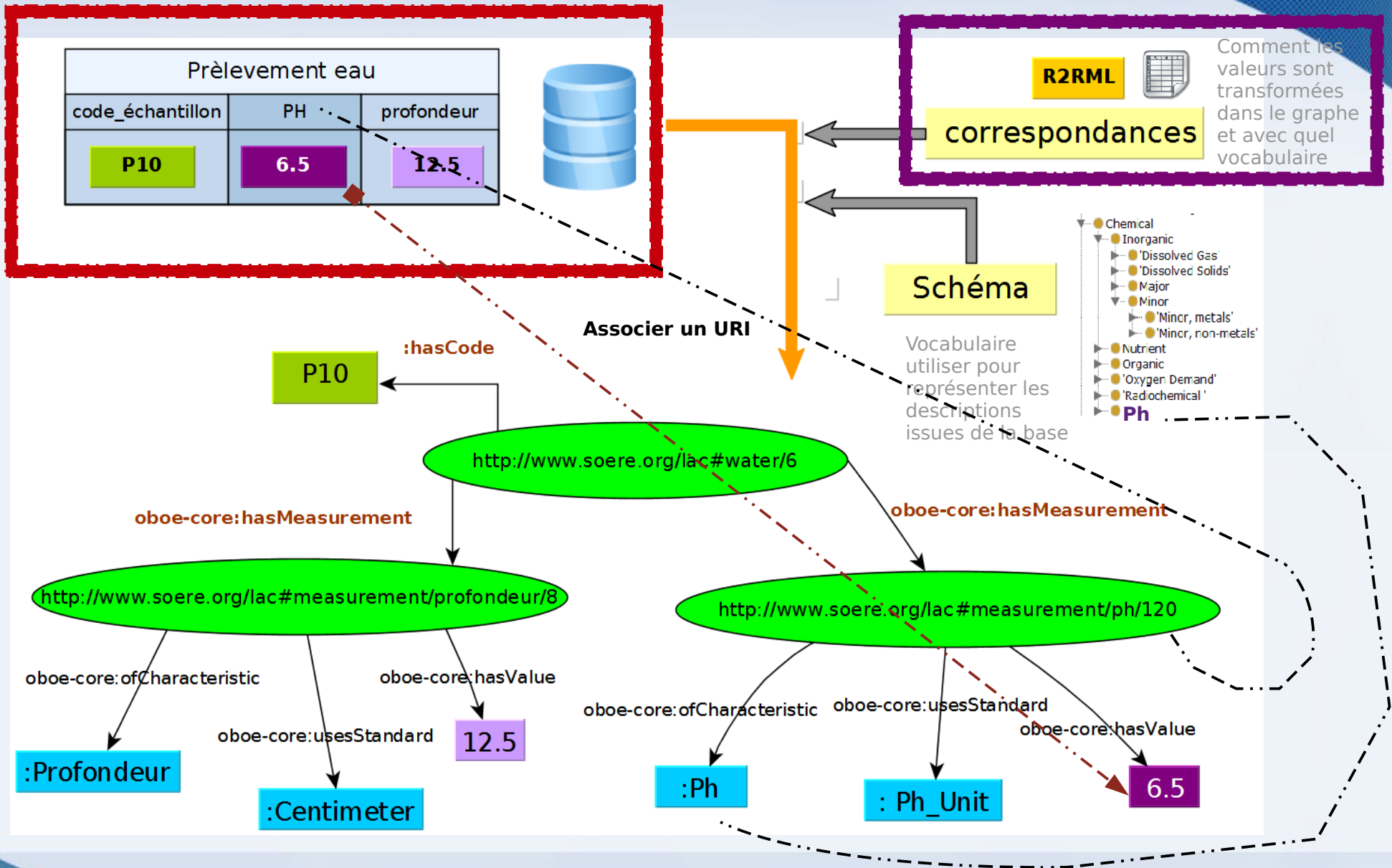


Comment les valeurs sont transformées dans le graphe et avec quel vocabulaire

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



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



# Inventaire des outils sémantiques ( Sparql Endpoint )

( Phase de Publication Des Données )



## TripleStore

### \* Sesame



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

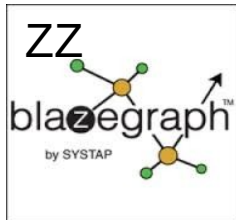
### \* Sol-RDF



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

REST

### \* BlazeGraph



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

### \* Corese



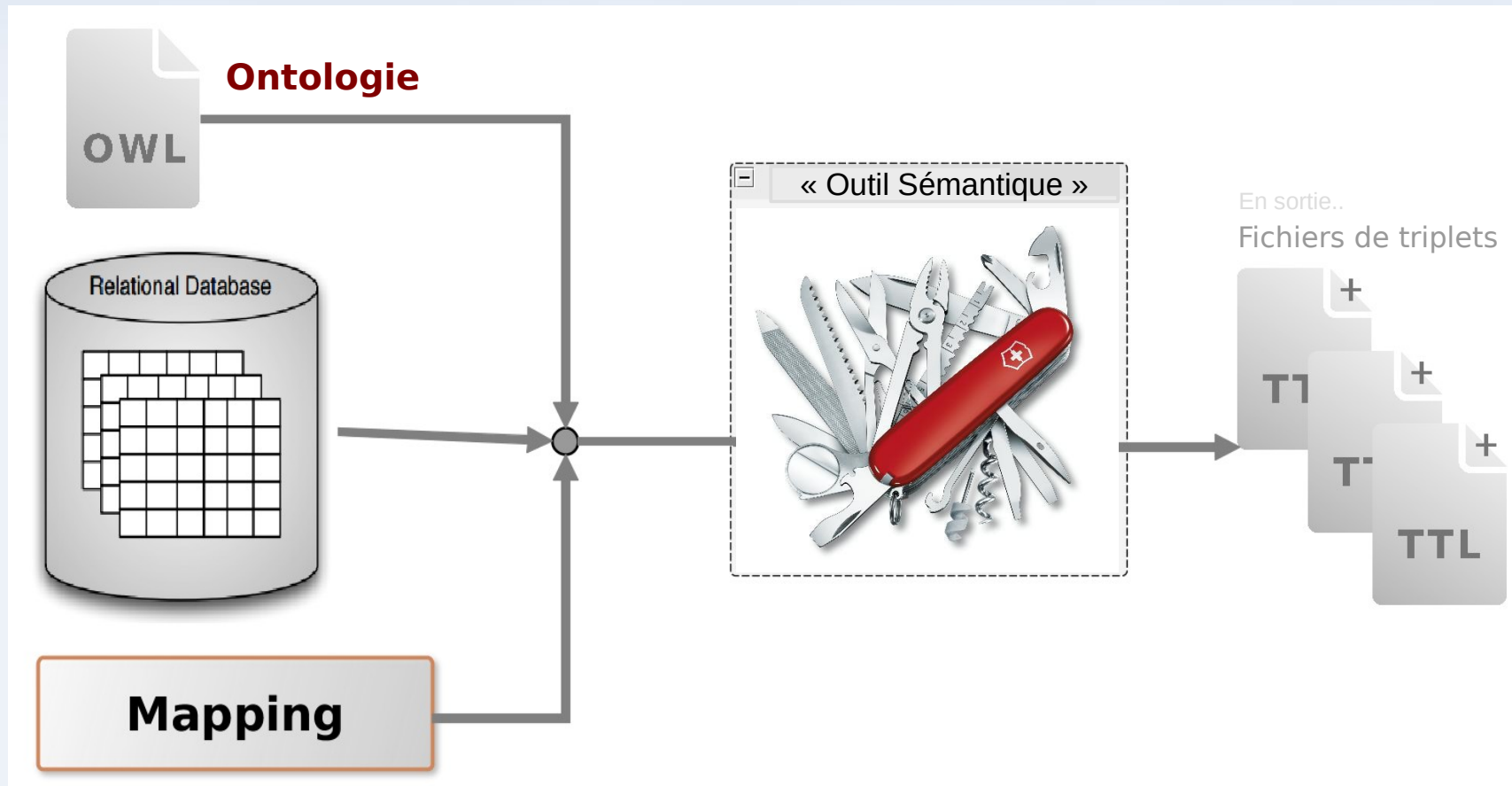
- Robustesse : **OK**
- Scaling out :
- Performance : **OK**

## 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 Protégé



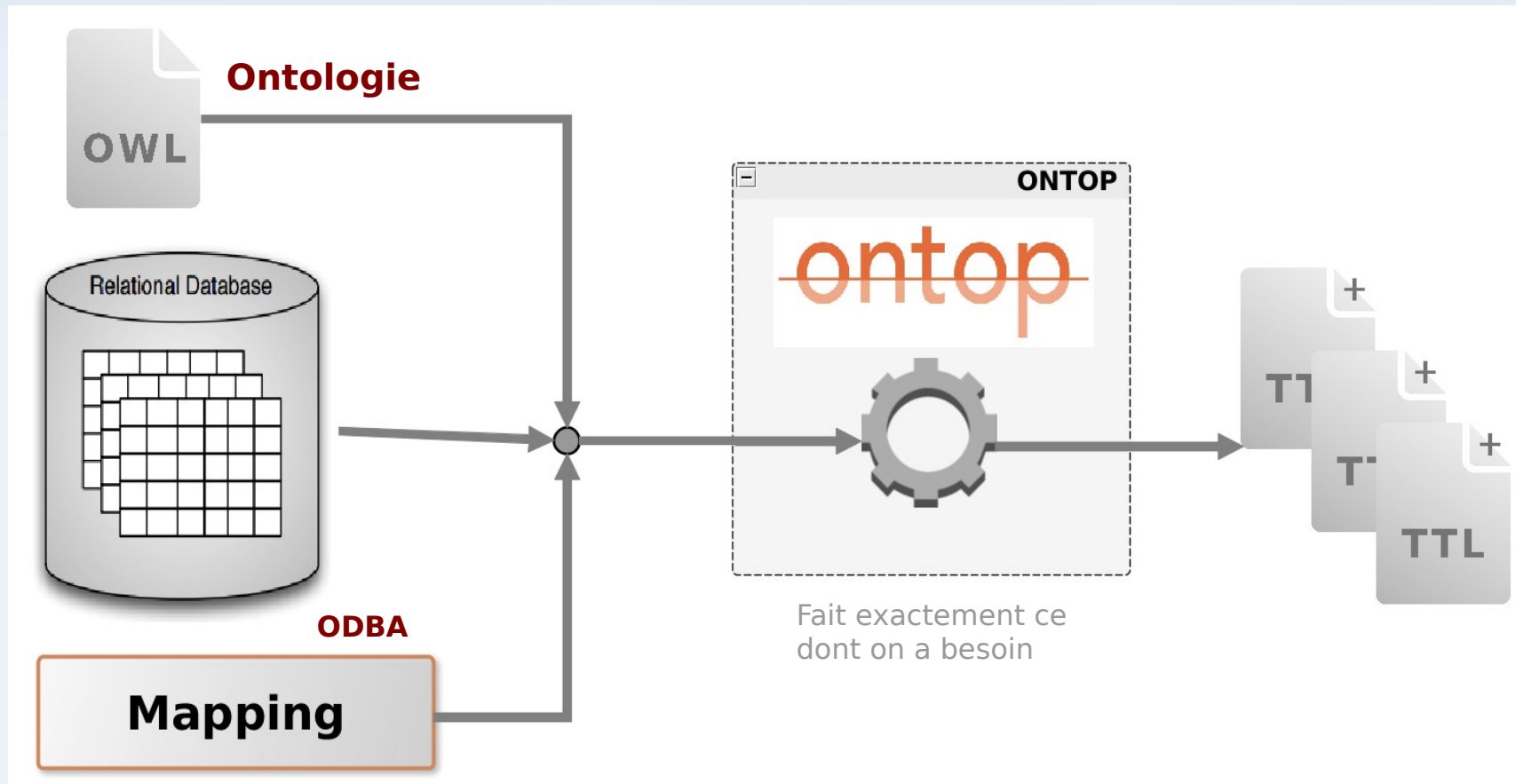
- Support « uniquement » SPARQL 1.0



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



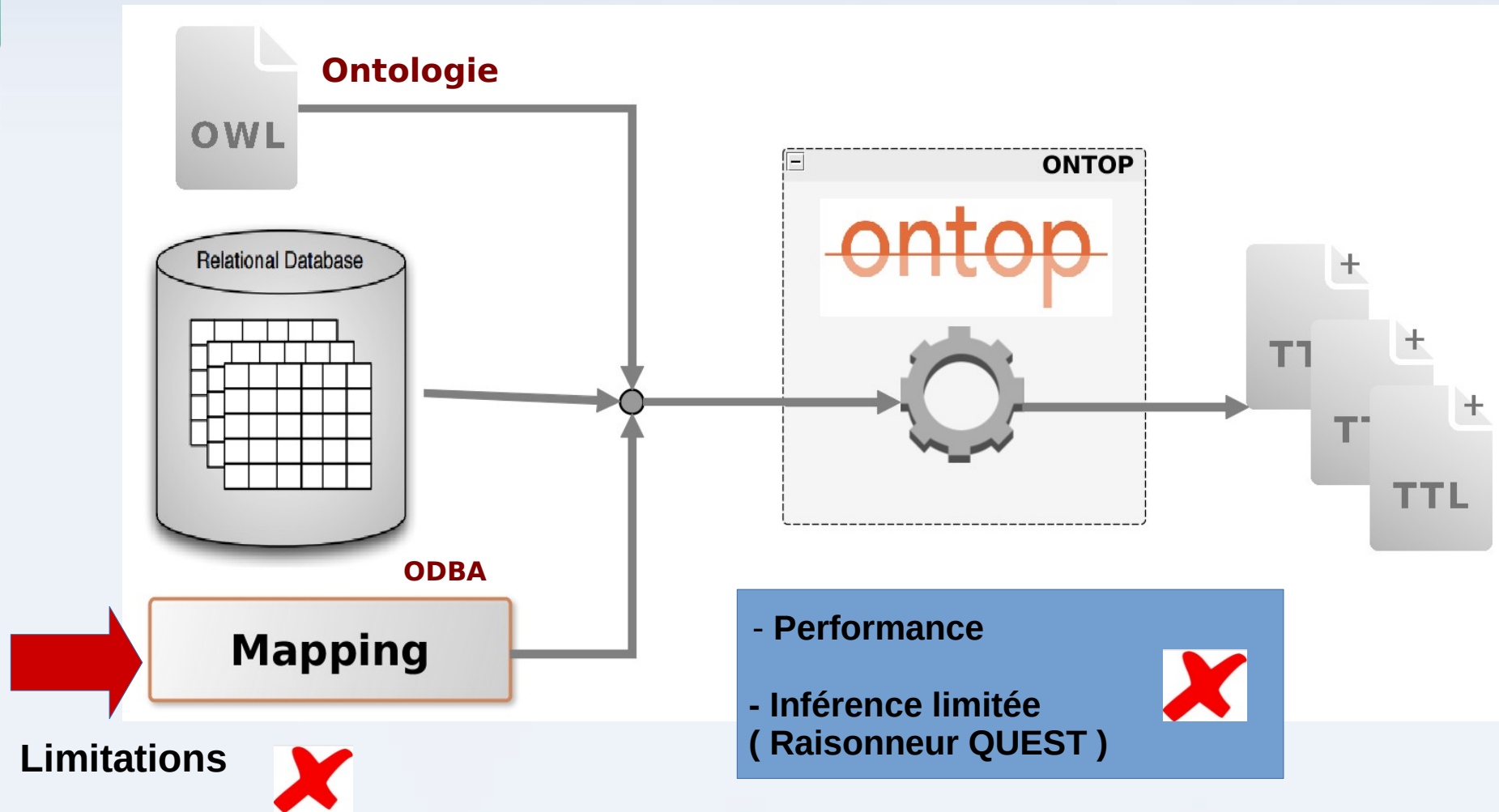
L'outil retenu : Ontop



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

Mais... ?

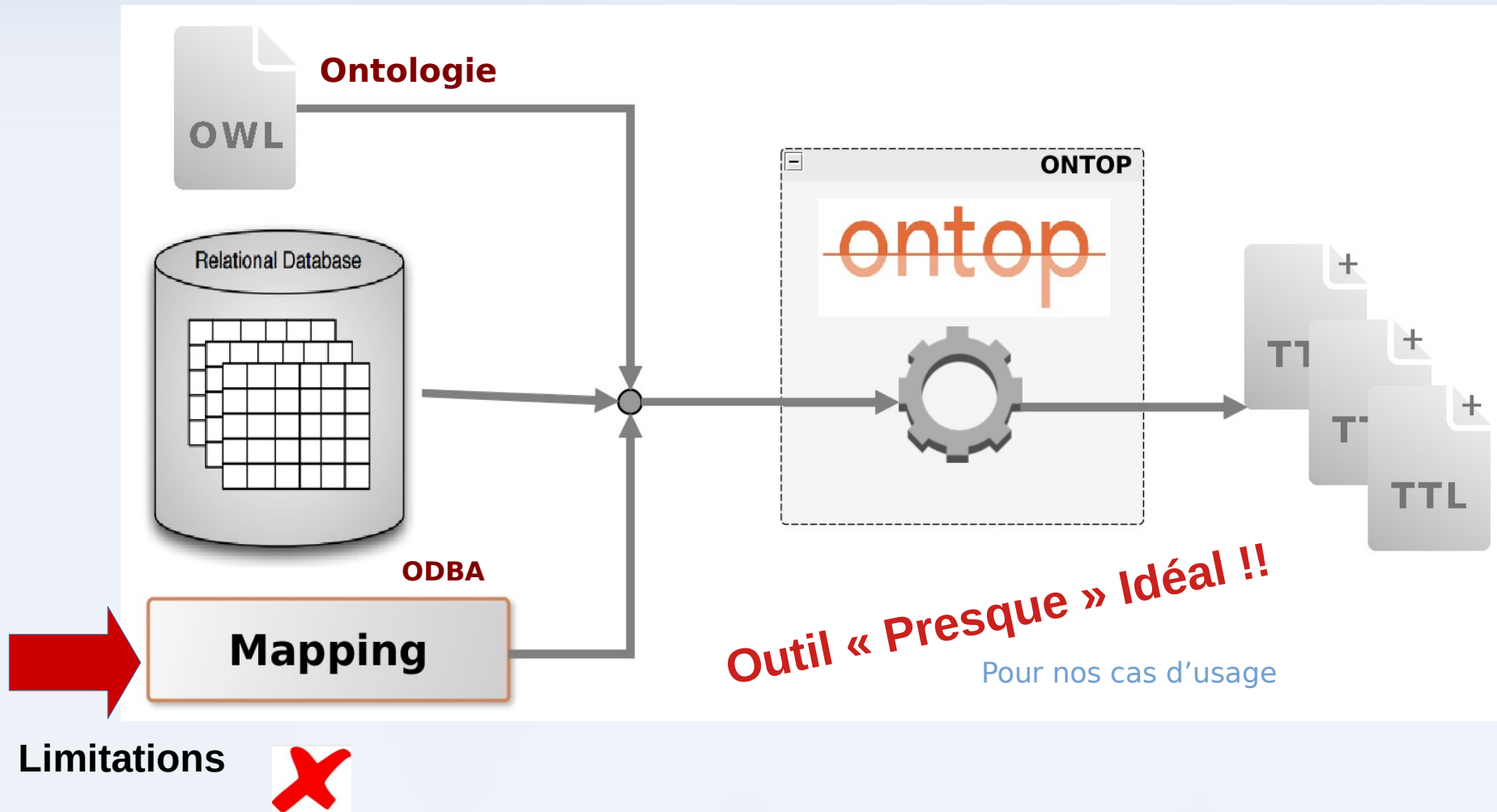
# Inventaire des outils sémantiques ( Phase d'annotation )





# 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`
- Parties Target (2):** The 'Edit Mapping' window shows the 'Target (Triples Template)' section with the following triples:
 

```

      :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 shows the 'Source (SQL Query)' section with the following query:
 

```

      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'
      
```

Créer les annotation à la main

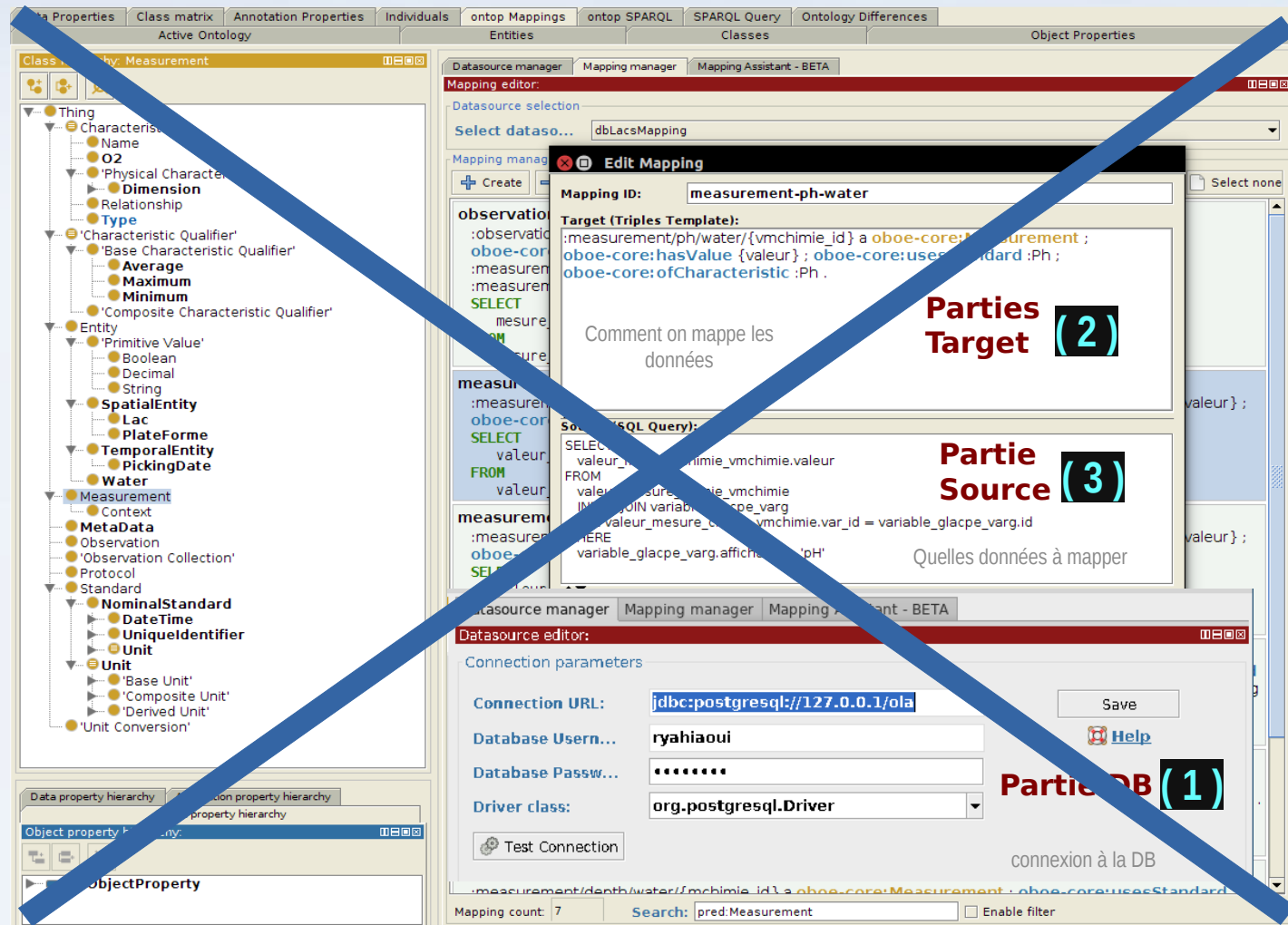


## Ontop - Protegé

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-france.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)

(3)

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'
```



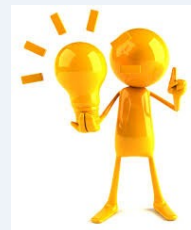
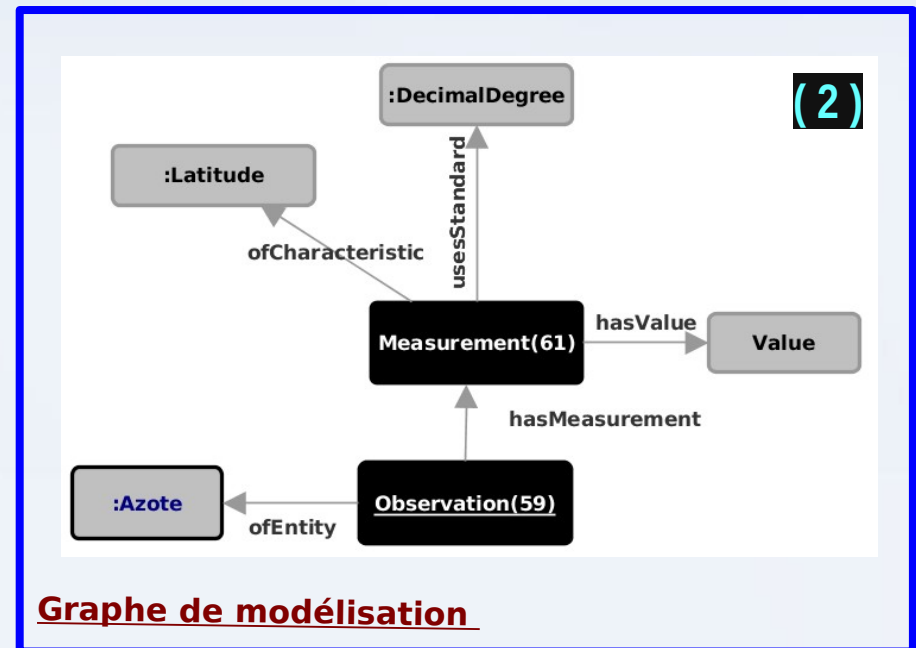
Représentation sous  
forme d'un Graphe

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

À condition de fournir un URI pour chaque  
nœud non terminal

Sachant que les graphes sont un outils simple et en  
même temps puissants pour faire de la modélisation  
sémantique..

**Partie Target**



**Pourquoi ne pas générer les fichiers  
de mapping ( ODBA ) à partir de ces  
graphes de modélisation  
sémantique ??**





## Partie Target

Informations de connexion à la BD (1)

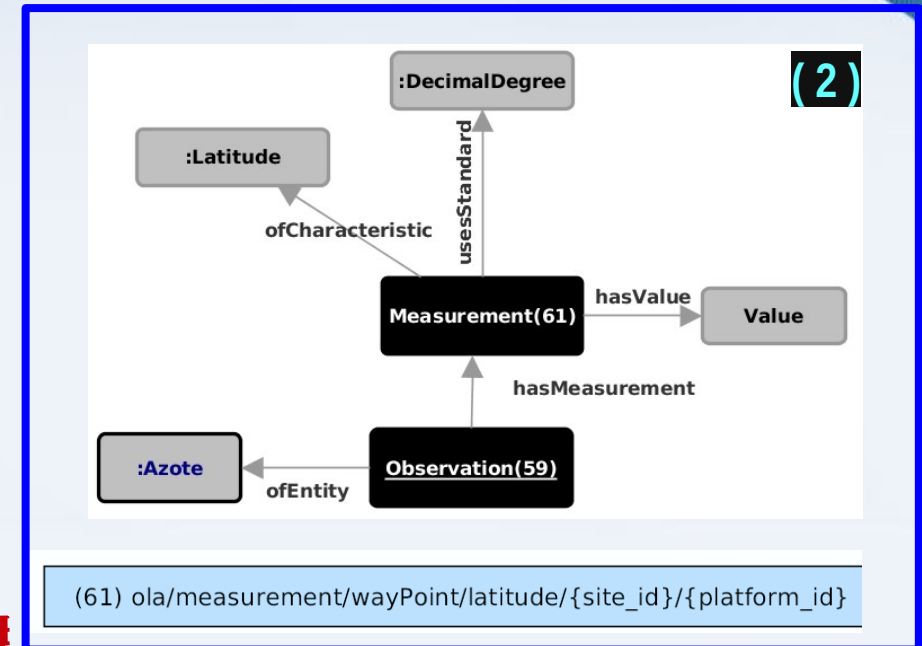
obda-sourceUri : dbLacsMapping

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

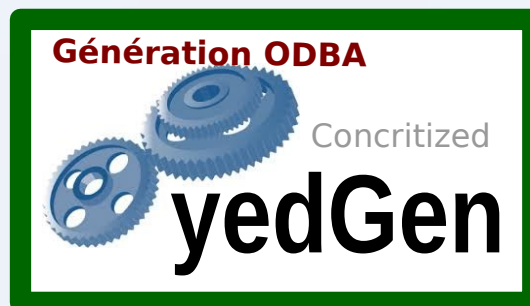
obda-username : ryahiaoui

obda-password : yahiaoui

obda-driverClass : org.postgresql.Driver



YedGen : Outil de  
génération de  
fichiers ODBA 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  
FROM  
public.site\_glacpe\_sit site INNER JOIN public.plateforme\_pla pla ON site.id = pla.id (3)

## Partie Source

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



## Généricité

L'idée derrière cette généricité est d'utiliser 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 ( ODBA ) par variable

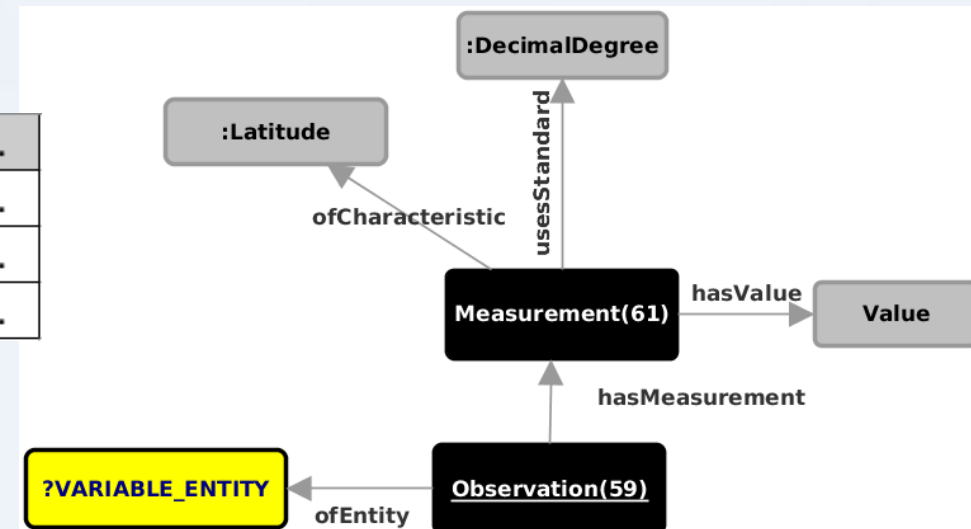
Cette généricité concerne le fonctionnement de l'outil yedGen

### 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 ...

**Graphe Type =**  
**Un graphe pour plusieurs variables**

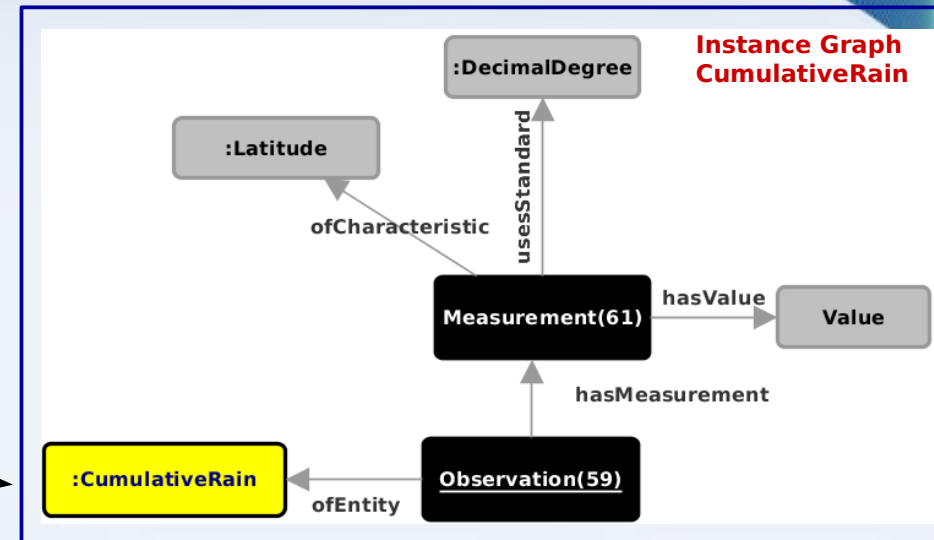




# Généricité

## 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	..

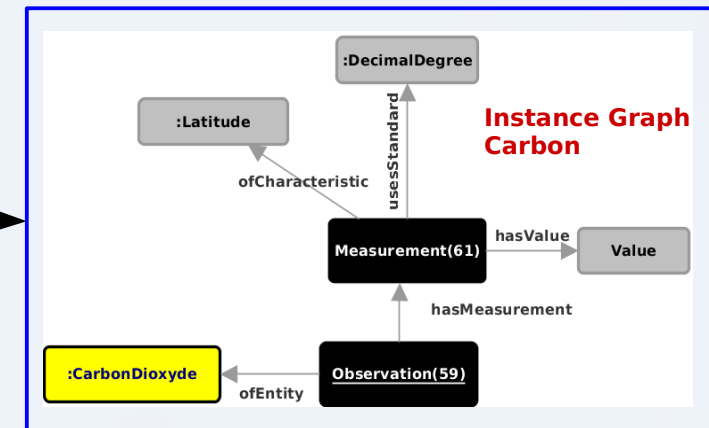




# Généricité

## 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	..

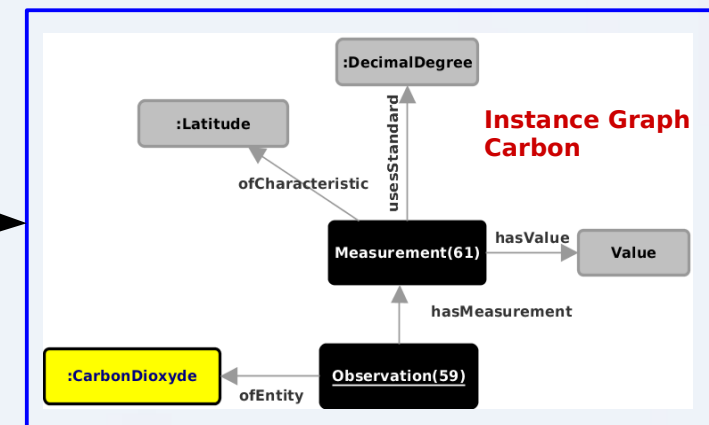
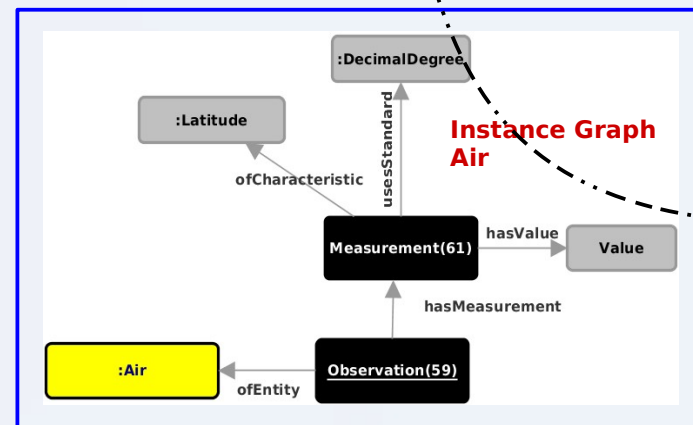
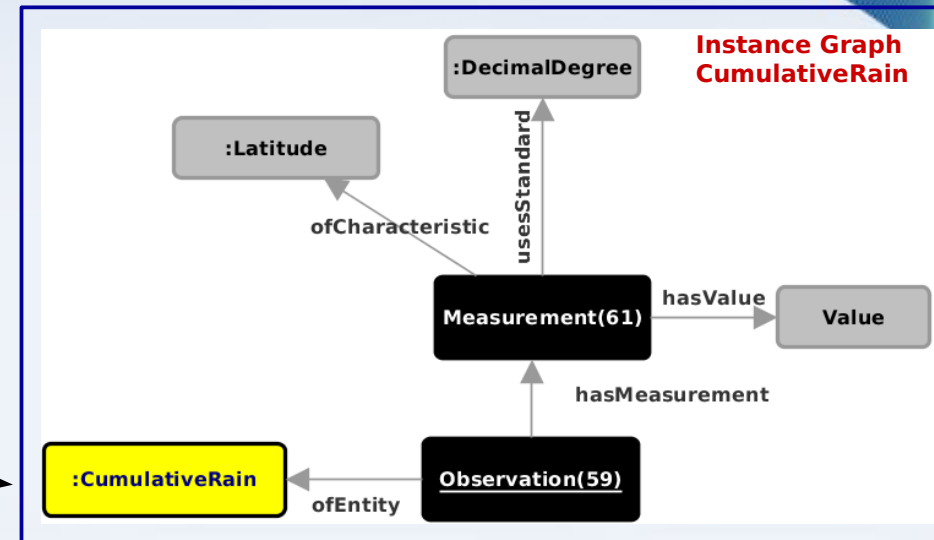




# Généricité

## Description d'un fichier CSV de variables sémantiques

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





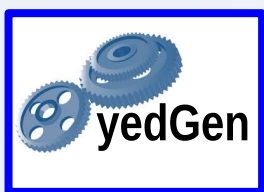


# Généricité

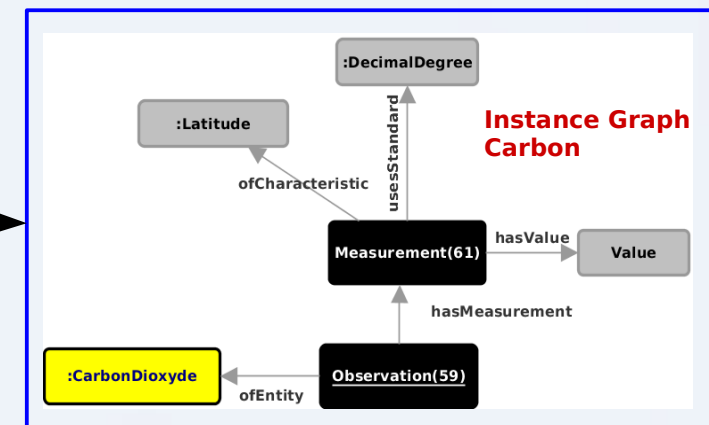
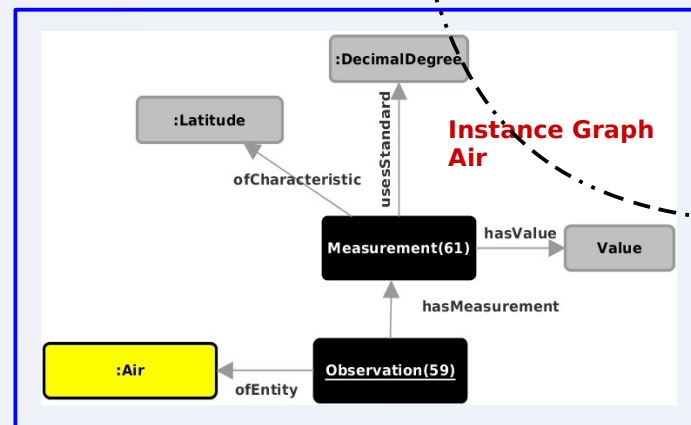
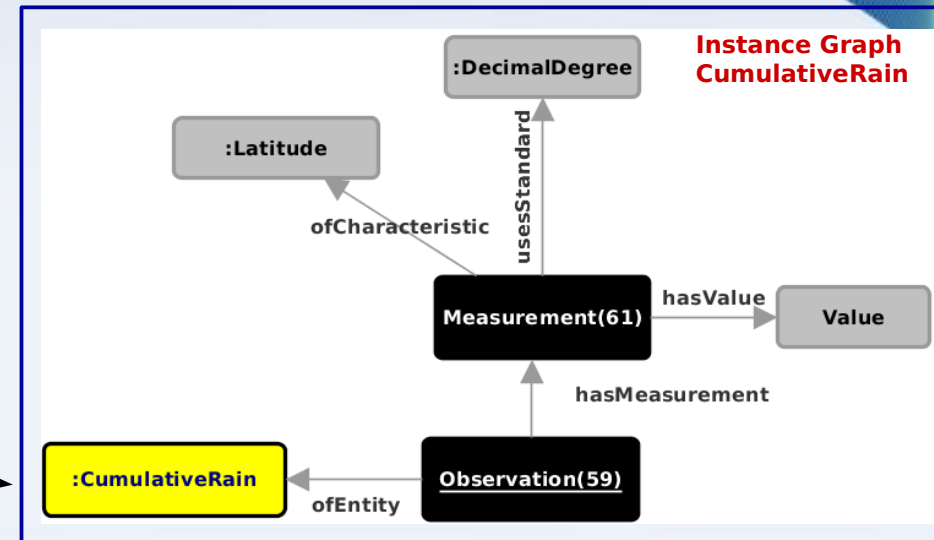
## Description d'un fichier CSV de variables sémantiques

AnaEE Standar	Entity	Context	..
cumulative rainfall	cumulative rain		..
air carbon dioxide	carbon 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

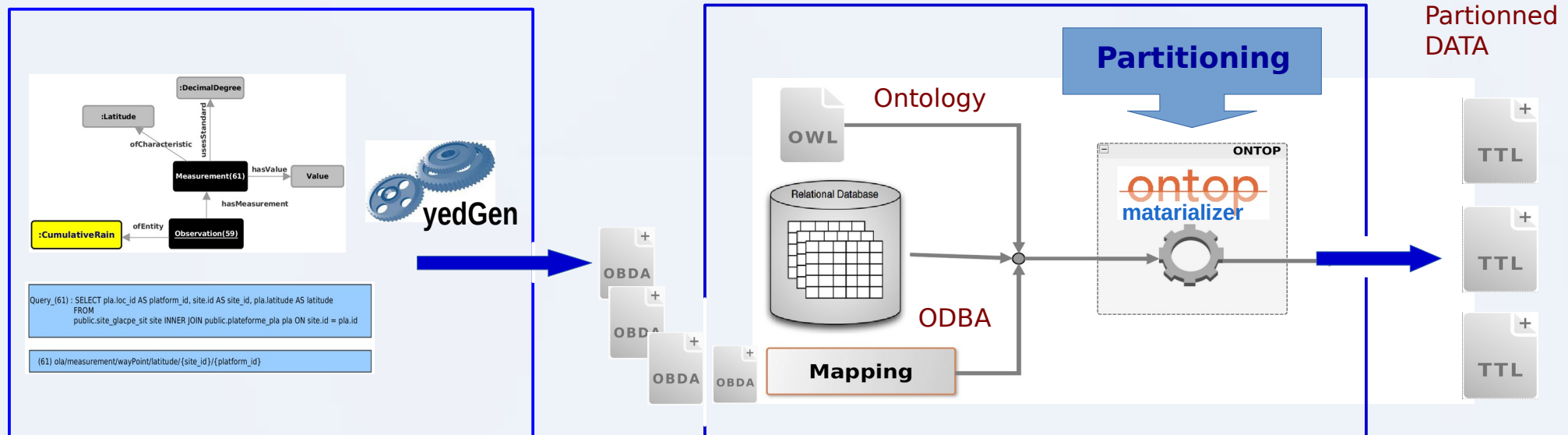




## Gros Fichiers / Grosse Bases de données

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



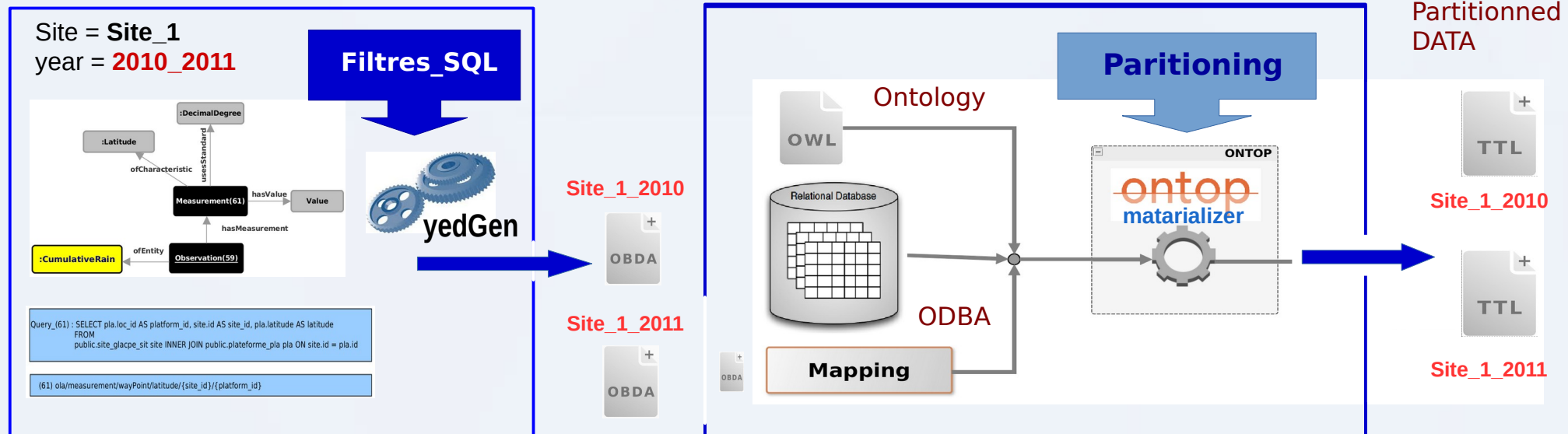
## Filtre sur les données

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

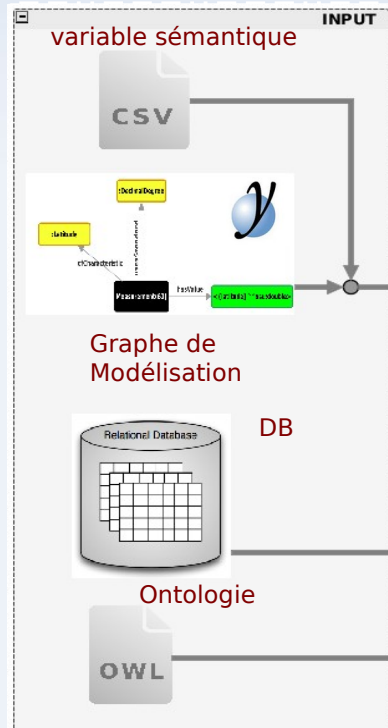
Solution : **Logical data partitioning**

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

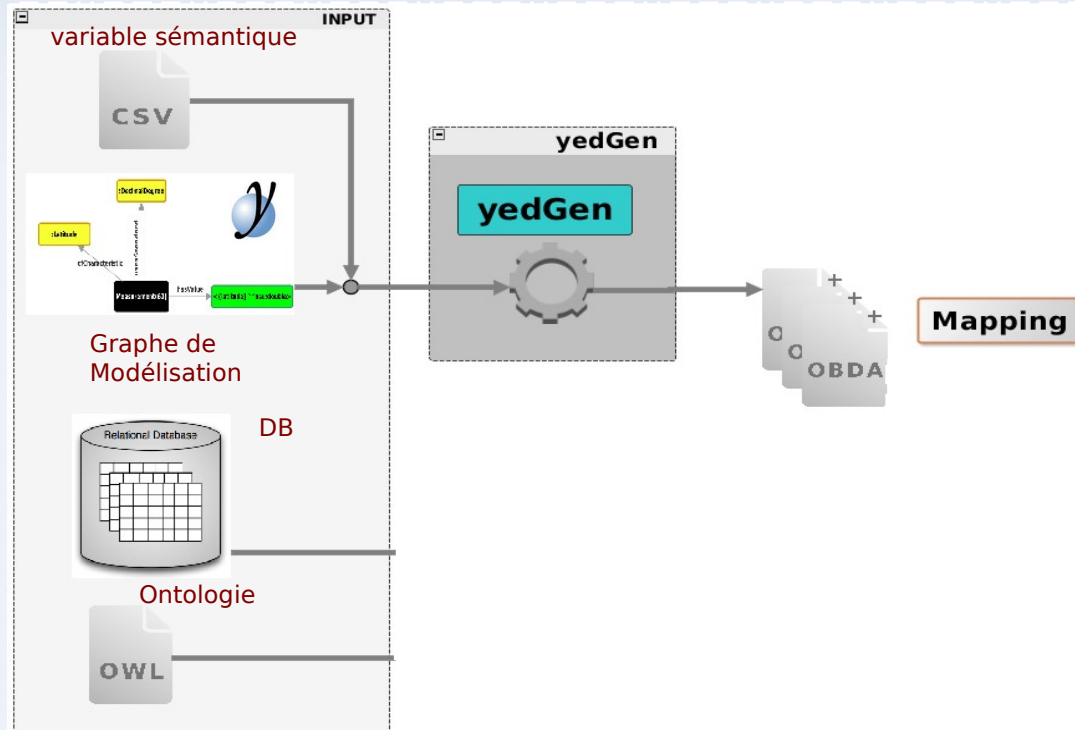
Plus vous filtrez les données, moins vous en avez, plus vous êtes performant



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

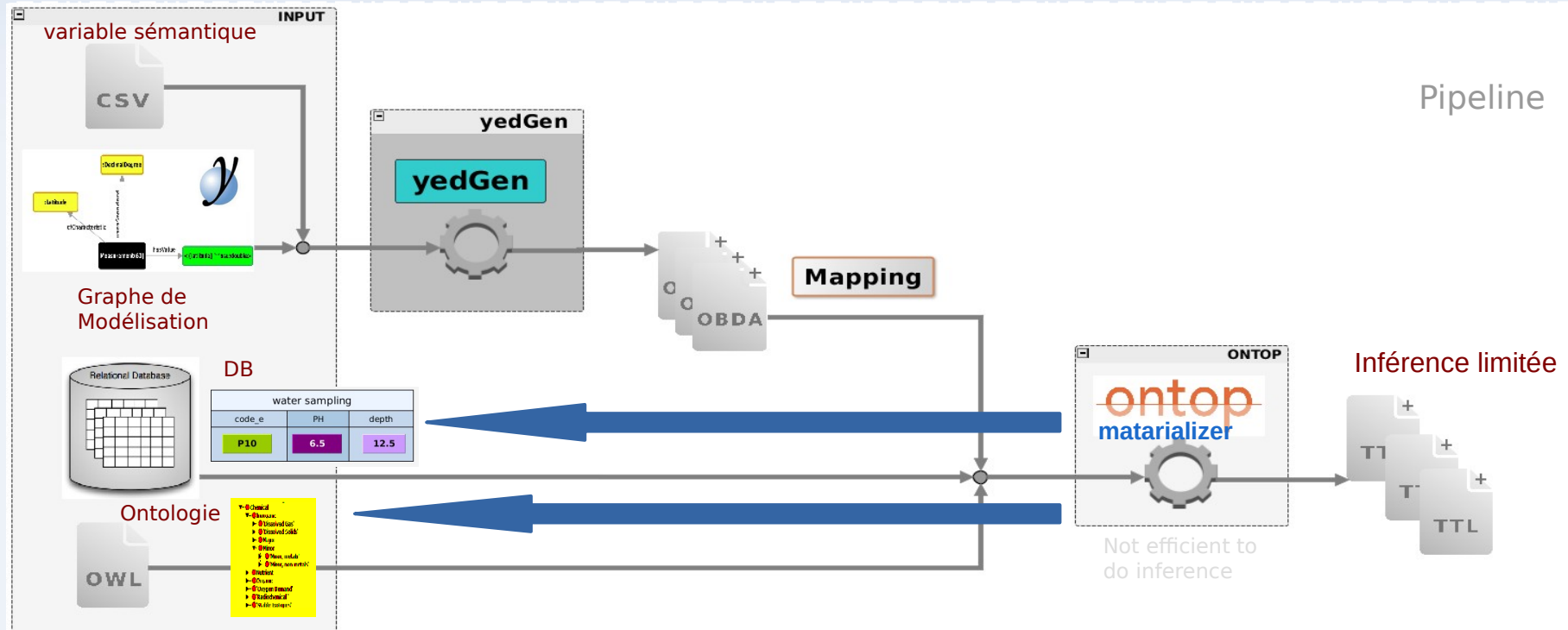


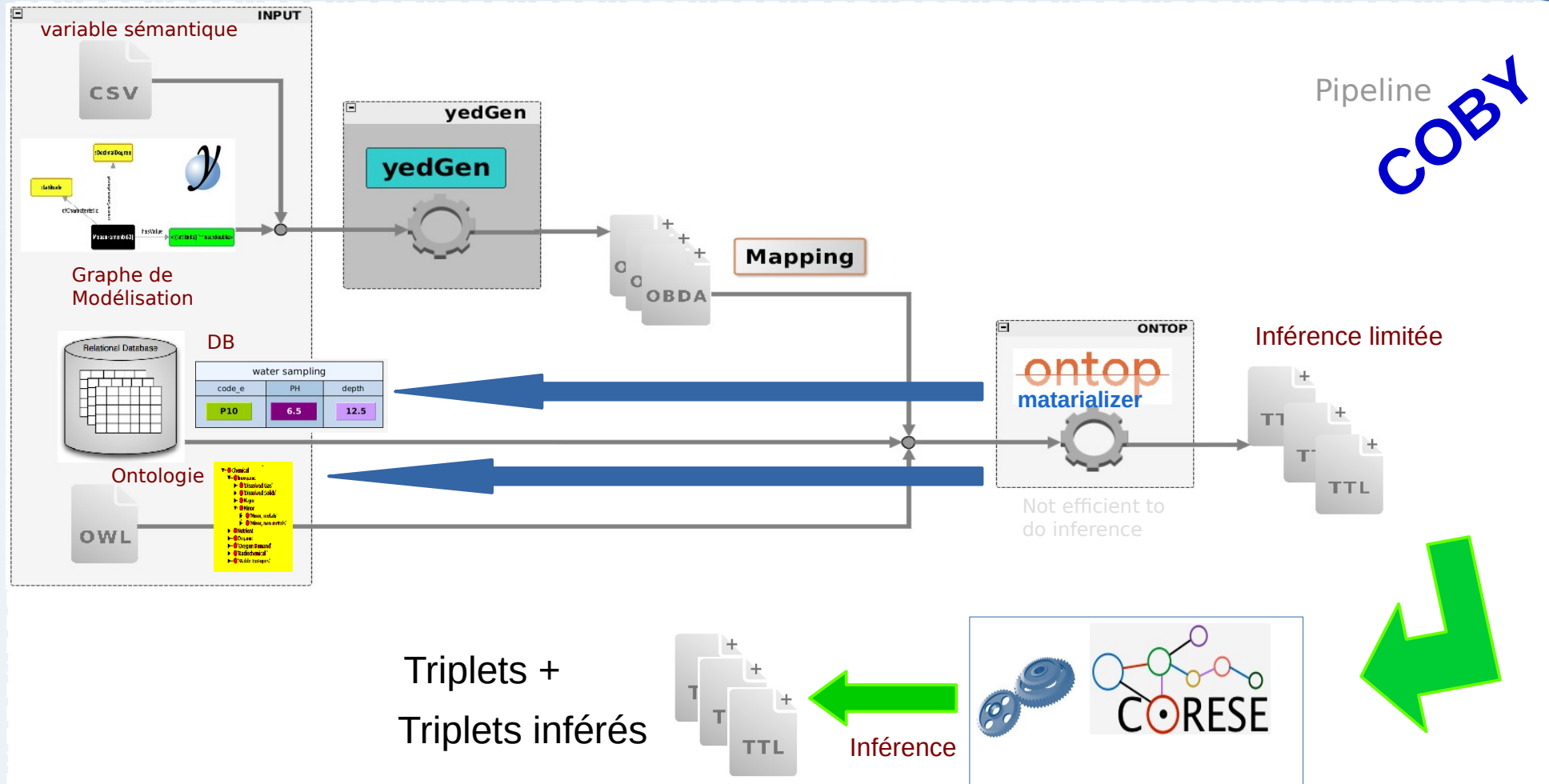
Pipeline



Pipeline



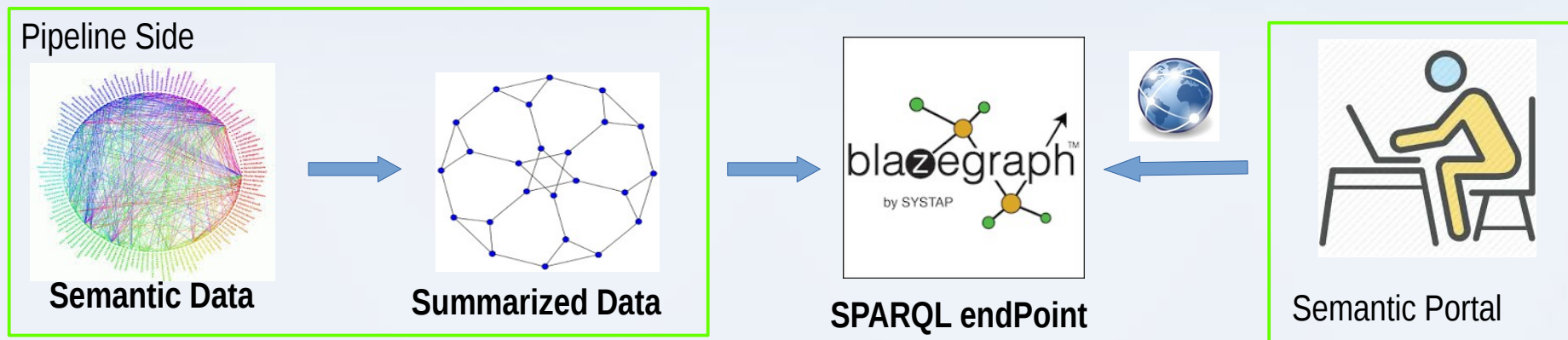




## 1 – Données de synthèse ( sémantiques ) pour le portail Anaee-F

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 fichiers netCDF

Use Case 2



**Objectif :** Production de données sémantiques filtrées ( au format TTL ), qui seront utilisées pour produire des fichiers au format **netCDF**

- **yedGen : Génération instantanée des fichiers de mapping ( ODBA ) à partir des graphes de modélisation**  
⇒ **Passage instantané : Modélisation → Annotation**
- **Image Docker pré-configurée du pipeline + Déployable en un clique**
- **Généricité**
- **Métriques :**

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

* <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 ODBA ( 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

