

INRAE



GraphQL pour l'Intégration de Données JSON et RDF pour OpenSILEX

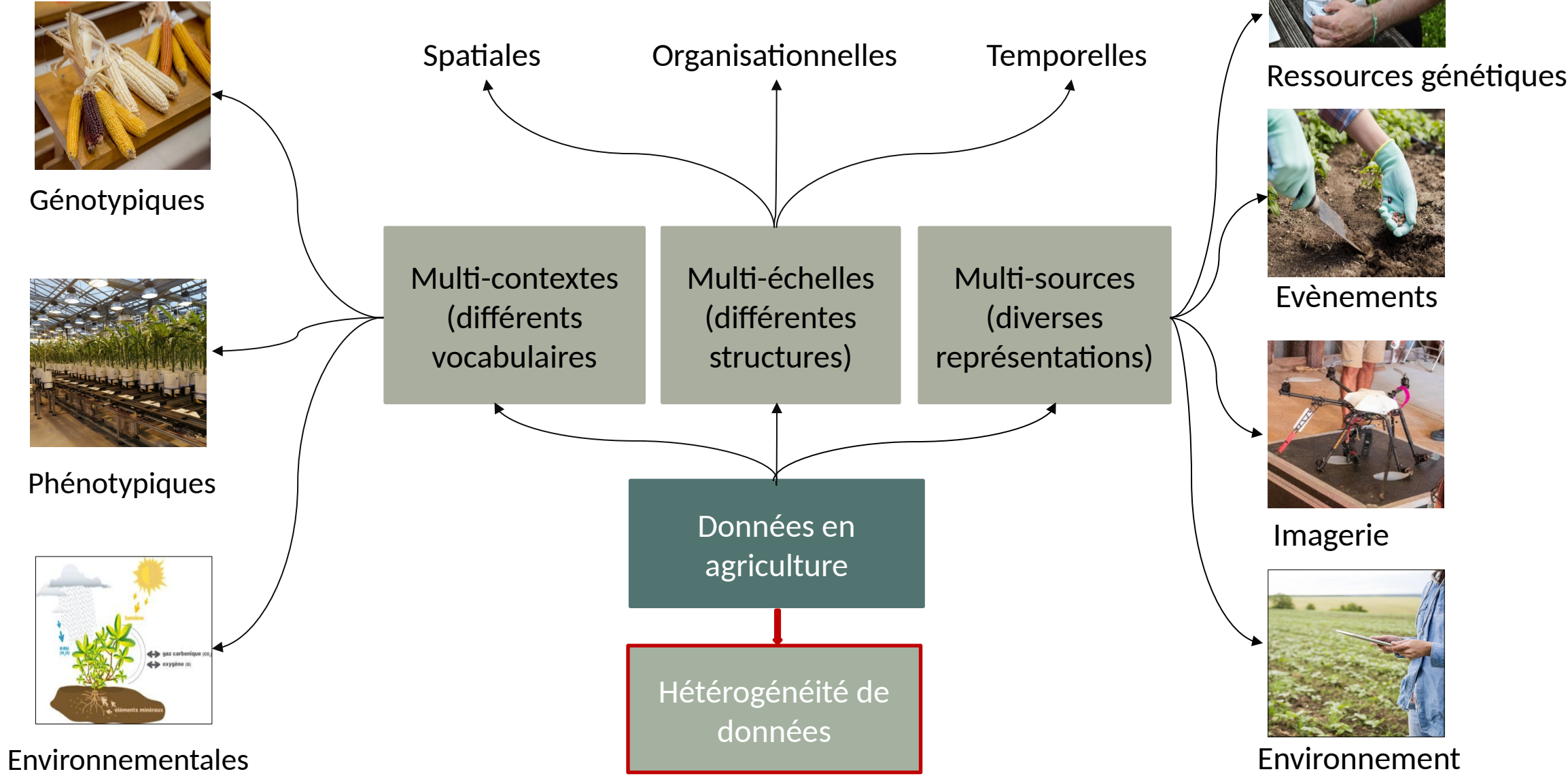
Contact :

sarra.abidri@inrae.fr

Contenu

-
- 1 Contexte & Problématique.
 - 2 Solution d'intégration (Service GraphQL).
 - 3 Apports.
 - 4 Gestion des droits d'accès.
 - 5 Cas d'usage (Illustration en Python).
 - 6 Conclusion & perspectives.

Contexte & Problématique

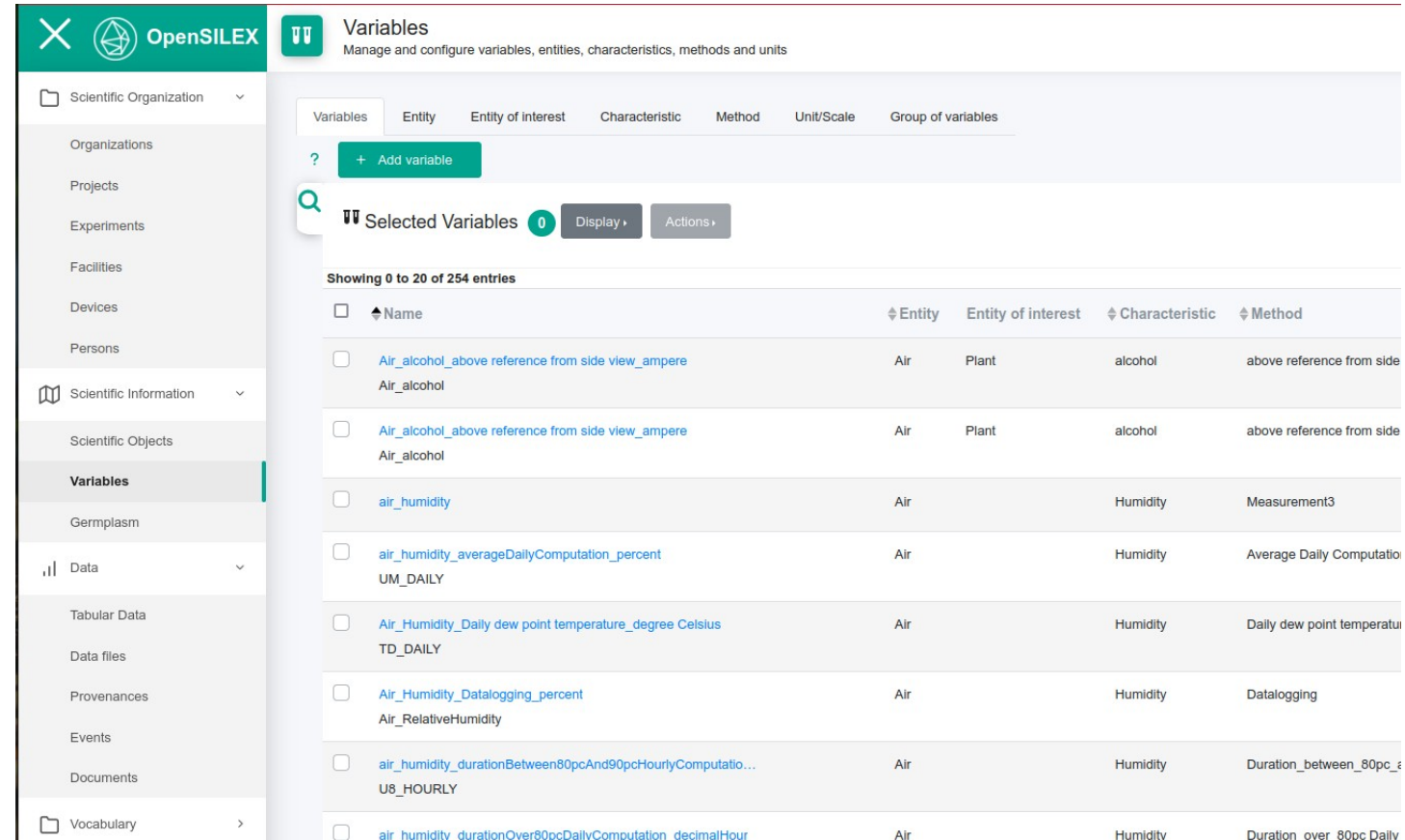


Comment regrouper ces données hétérogènes ?

Systeme d'Information pour l' experimentation

Un Framework open source dédié à la gestion des données expérimentales agronomiques et environnementales.

Méthodes et outils de structuration des données hétérogènes, massives et multi sources.



Variables
Manage and configure variables, entities, characteristics, methods and units

Variables Entity Entity of interest Characteristic Method Unit/Scale Group of variables

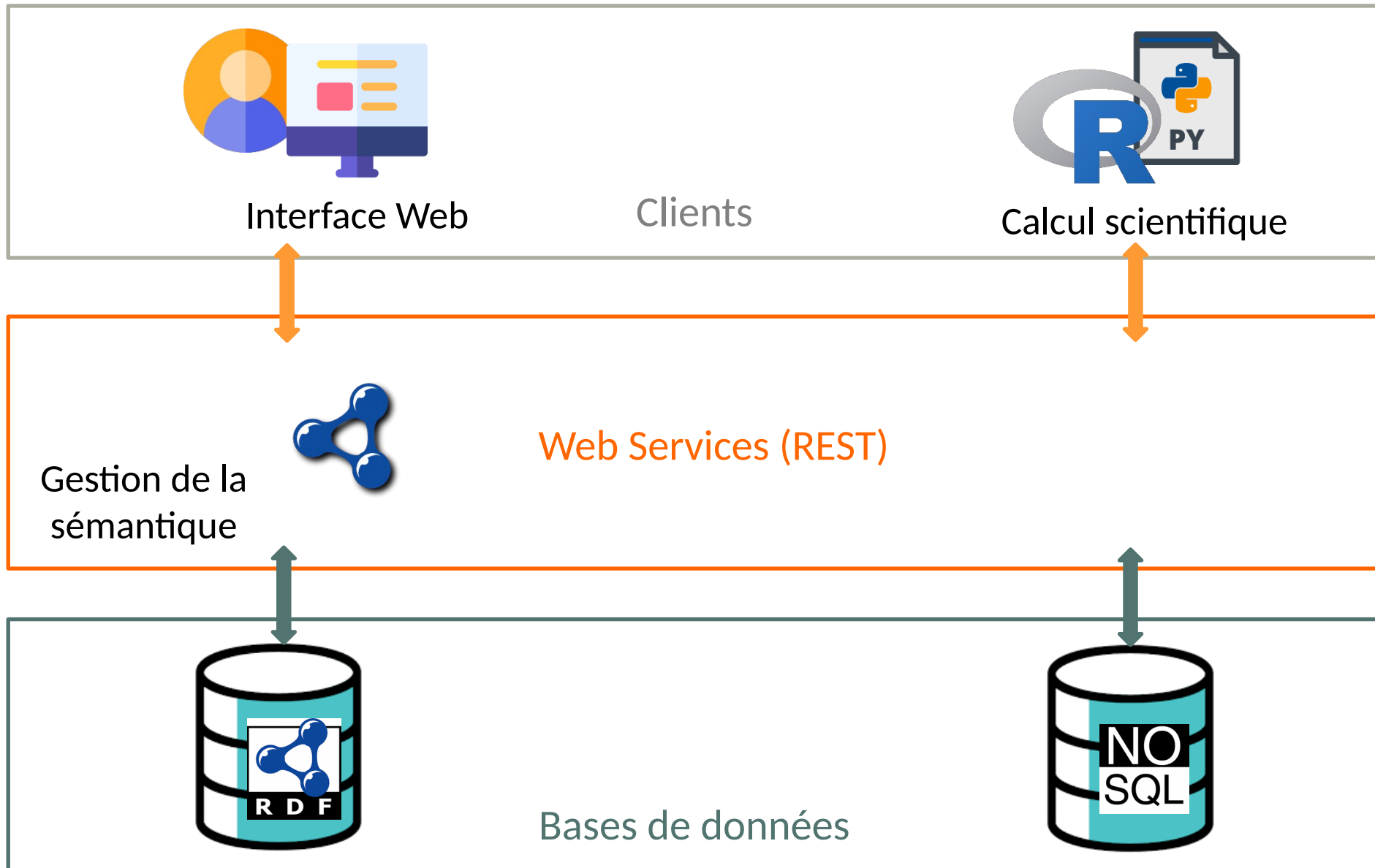
+ Add variable

Selected Variables 0 Display Actions

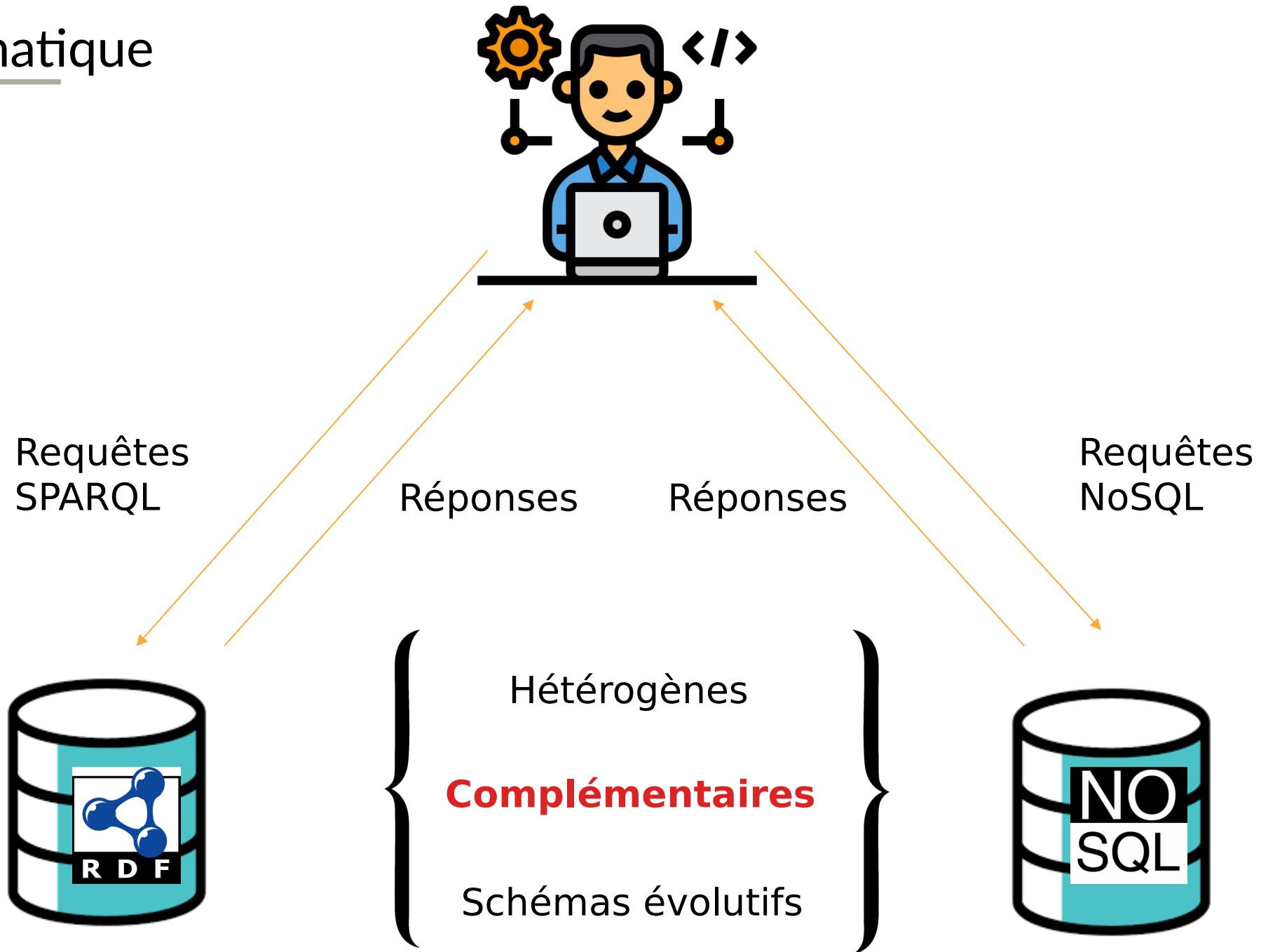
Showing 0 to 20 of 254 entries

<input type="checkbox"/>	Name	Entity	Entity of interest	Characteristic	Method
<input type="checkbox"/>	Air_alcohol_above reference from side view_ampere Air_alcohol	Air	Plant	alcohol	above reference from side
<input type="checkbox"/>	Air_alcohol_above reference from side view_ampere Air_alcohol	Air	Plant	alcohol	above reference from side
<input type="checkbox"/>	air_humidity	Air		Humidity	Measurement3
<input type="checkbox"/>	air_humidity_averageDailyComputation_percent UM_DAILY	Air		Humidity	Average Daily Computatio
<input type="checkbox"/>	Air_Humidity_Daily dew point temperature_degree Celsius TD_DAILY	Air		Humidity	Daily dew point temperatu
<input type="checkbox"/>	Air_Humidity_Datalogging_percent Air_RelativeHumidity	Air		Humidity	Datalogging
<input type="checkbox"/>	air_humidity_durationBetween80pcAnd90pcHourlyComputatio... U8_HOURLY	Air		Humidity	Duration_between_80pc_ε
<input type="checkbox"/>	air humidity durationOver80pcDailyComputation decimalHour	Air		Humidity	Duration over 80pc Daily

Architecture OpenSILEX



Problématique



Défis

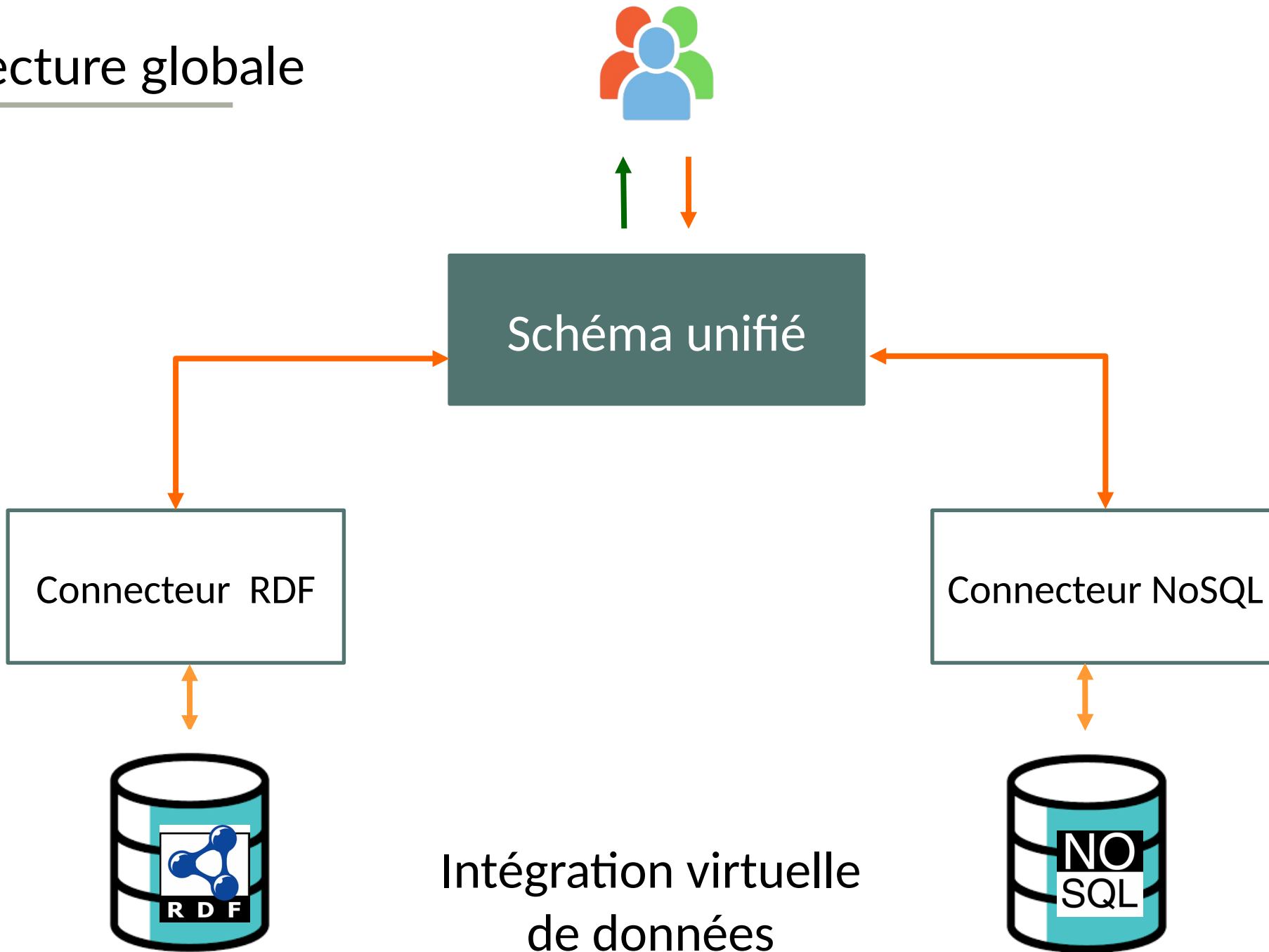
Fournir une vue unifiée et dynamique des sources.

Interroger les données en faisant abstraction de leur source.

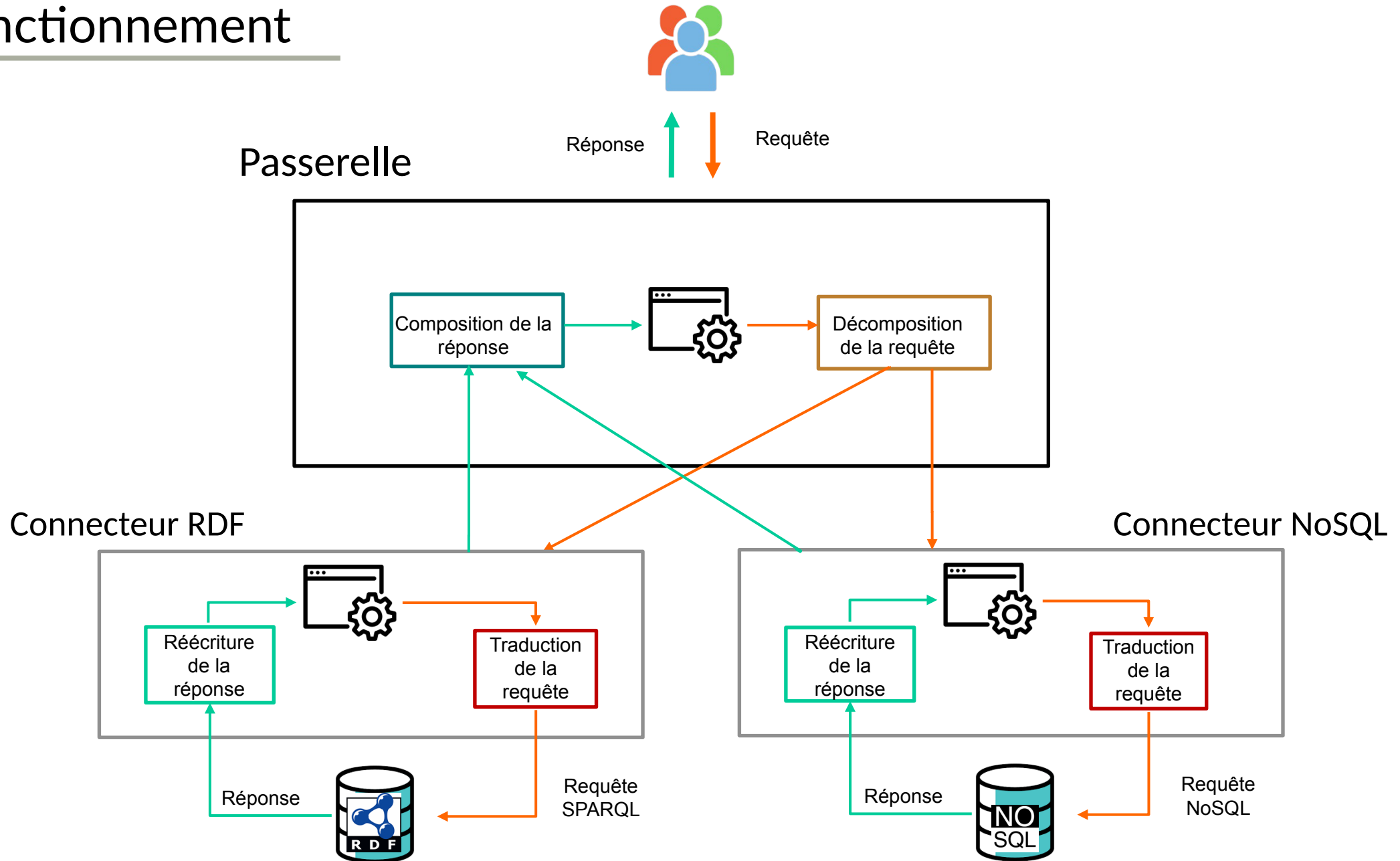
Effectuer des requêtes fédérées sur les sources de données.

Intégrer d'une façon performante les données.

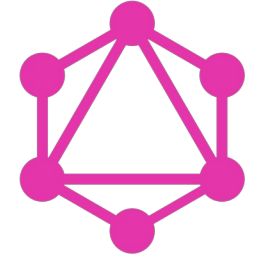
Architecture globale



Fonctionnement



GraphQL

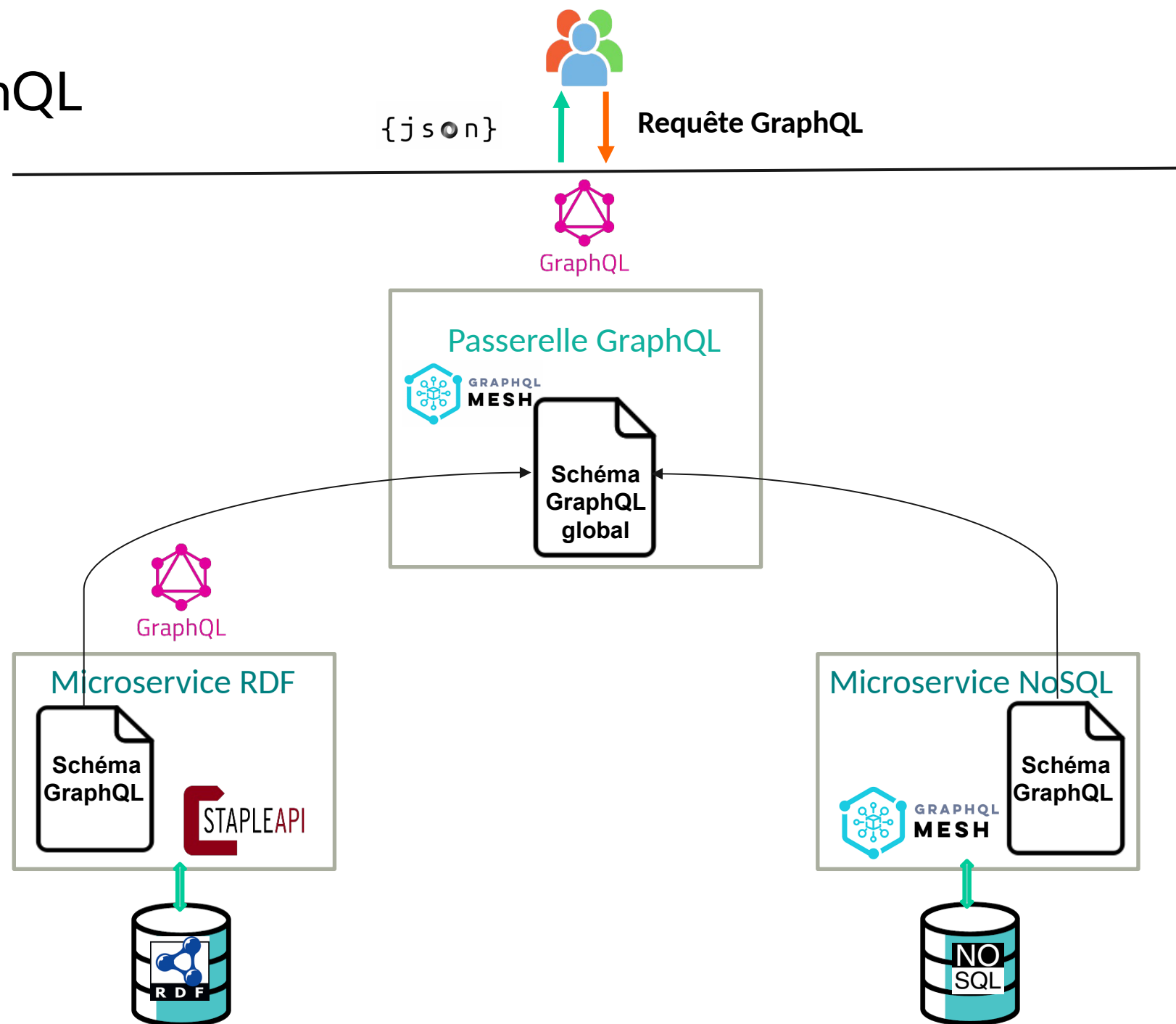


Graph Query Language

GraphQL est un langage de requête et un environnement d'exécution côté serveur pour les interfaces de programmation d'application (API).

- Schéma (Schema)
- Opérations (Queries et Mutations)
- Champs (Fields)
- Résolveurs (Resolvers)

Passerelle GraphQL





Explorer

query MyQuery

- MG_DataFile_connection
- MG_DataFile_count
- MG_DataFile_dataLoader
- MG_DataFile_dataLoaderMany
- MG_DataFile_findById
- MG_DataFile_findByIds
- MG_DataFile_findMany
- MG_DataFile_findOne
- MG_DataFile_pagination
- MG_Data_connection
- MG_Data_count
- MG_Data_dataLoader
- MG_Data_dataLoaderMany
- MG_Data_findById
- MG_Data_findByIds
- MG_Data_findMany
- MG_Data_findOne
- MG_Data_pagination
- MG_DeviceAttribute_connection
- MG_DeviceAttribute_count
- MG_DeviceAttribute_dataLoader
- MG_DeviceAttribute_dataLoaderMany
- MG_DeviceAttribute_findById
- MG_DeviceAttribute_findByIds
- MG_DeviceAttribute_findMany
- MG_DeviceAttribute_findOne
- MG_DeviceAttribute_pagination
- MG_Geometry_connection
- MG_Geometry_count
- MG_Geometry_dataLoader
- MG_Geometry_dataLoaderMany
- MG_Geometry_findById
- MG_Geometry_findByIds
- MG_Geometry_findMany
- MG_Geometry_findOne
- MG_Geometry_pagination
- MG_GermplasmAttribute_connection
- MG_GermplasmAttribute_count
- MG_GermplasmAttribute_dataLoader

MyQuery MyQuery <untitled> <untitled> MyQuery MyQuery MyQuery MyQuery MyQuery MyQuery MyQuery MyQuery MyQuery MyQuery

```
1 query MyQuery {
2   RDF_ScientificObject(Experience: as_expe, inferred: true) {
3     _id
4     _type
5   }
6 }
```

Execute query (Ctrl-Enter)

```
{
  "data": {
    "RDF_ScientificObject": [
      {
        "_id": "http://www.phenome-fppn.fr/id/scientific-object/so-0000-bc/1",
        "_type": [
          "Plot"
        ]
      },
      {
        "_id": "http://www.phenome-fppn.fr/id/scientific-object/so-test_db_f",
        "_type": [
          "Plot"
        ]
      },
      {
        "_id": "http://www.phenome-fppn.fr/id/scientific-object/so-as-1",
        "_type": [
          "Plot"
        ]
      },
      {
        "_id": "http://www.phenome-fppn.fr/id/scientific-object/so-as-2",
        "_type": [
          "Plot"
        ]
      },
      {
        "_id": "http://www.phenome-fppn.fr/id/scientific-object/so-as-3",
        "_type": [
          "Plot"
        ]
      },
      {
        "_id": "http://www.phenome-fppn.fr/id/scientific-object/so-as-4",
        "_type": [
          "Plot"
        ]
      }
    ]
  }
}
```

Variables Headers

```
1 {
2   "Authorization":
3   "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.eyJpc19hZG1pbmI6Zm
4 }
```

Opérations

[< Schema](#)

Query

```
RDF_ScientificObject(  
  page: Int  
  inferred: Boolean = false  
  filter: RDF_FilterScientificObject  
  source: [RDF_DataSource] = [sparql]  
  limit: Int  
): [RDF_ScientificObject]
```

Get objects of type: ScientificObject

Types

[< Query](#)

RDF_ScientificObject

Search RDF_ScientificObject...

No Description

FIELDS

_id: ID!

The unique identifier of the object

_type(inferred: Boolean = false): [String]

Types of the object

hasReplication: [String]

hasVariety(source: [RDF_DataSource]): [RDF_Variety]

hasCreationDate: [String]

hasDestructionDate: [String]

label: String

hasGermplasm(source: [RDF_DataSource]): [RDF_Germplasm]

hasFactorLevel(source: [RDF_DataSource]): [RDF_FactorLevel]

MG_geometry: [MG_Geometry]!

Requête



```
1 query QueryScientificObjects ($id: [ID]){
2   RDF_ScientificObject(
3     Experience: test_xp_brice,
4     inferred: true,
5     filter:{_id : $id})
6 {
7   _id
8   _type
9   geometry{
10    geometry{
11     type
12     coordinates
13    }
14  }
15 }
16 }
```

variable

Arguments

Sélection

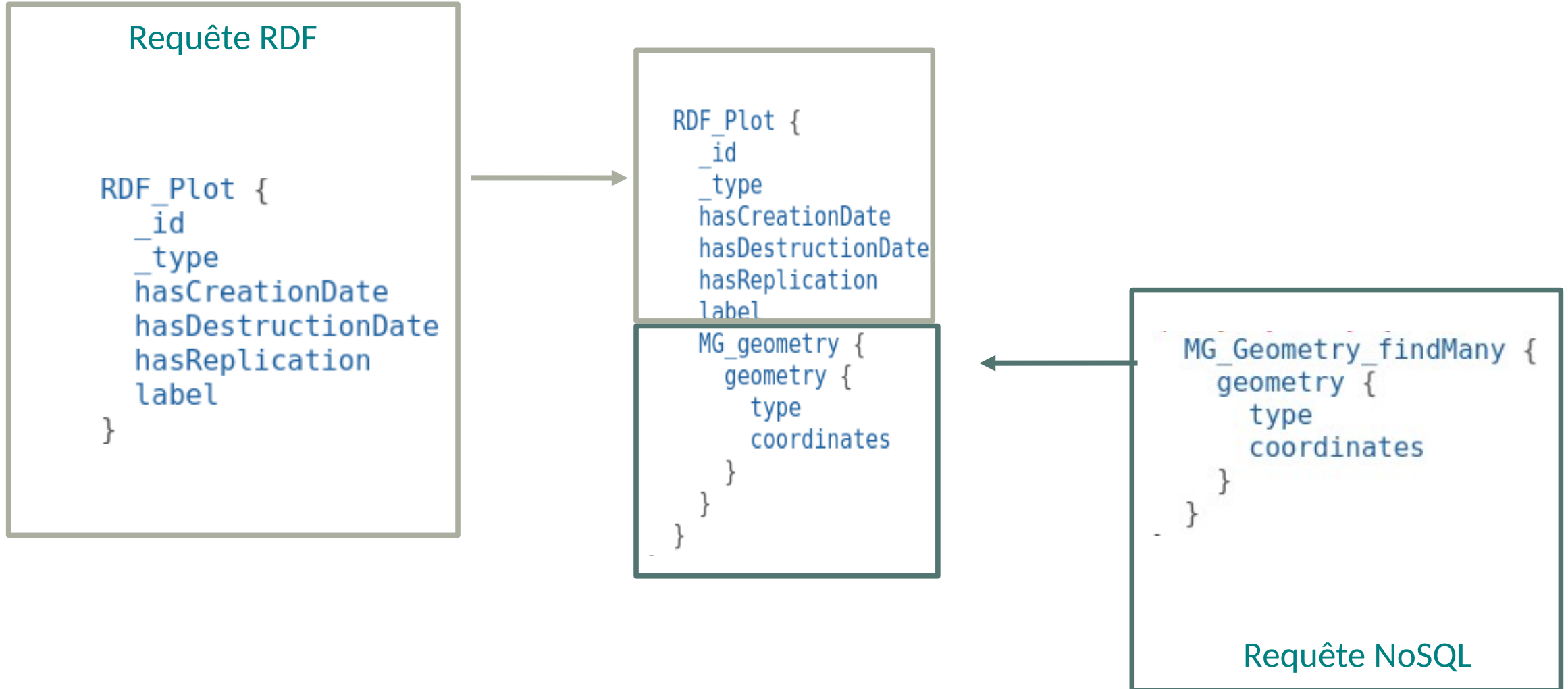
Variables Headers

```
1 {
2   "id" : "http://www.phenome-fppn.fr/id/scientific-object/so-sr"
3 }
```

Résultat



```
▼ {
  ▼ "data": {
    ▼ "RDF_ScientificObject": [
      ▼ {
        "_id": "http://www.phenome-fppn.fr/id/scientific-object/so-sr",
        ▼ "_type": [
          "Plant"
        ],
        ▼ "geometry": [
          ▼ {
            ▼ "geometry": {
              "type": "Point",
              ▼ "coordinates": [
                10,
                10
              ]
            }
          }
        ]
      }
    ]
  }
}
```



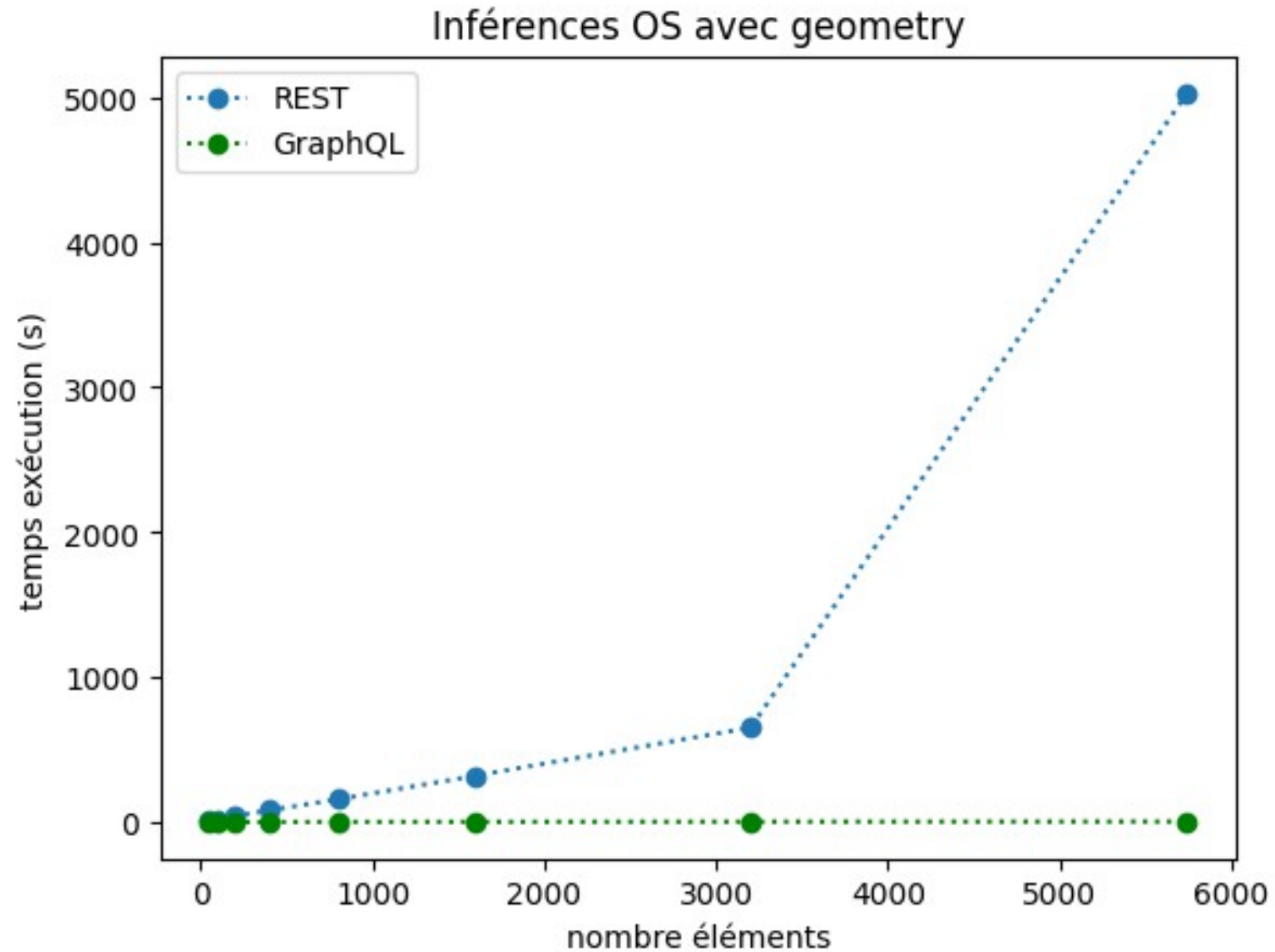
Performances

Requêtes en lecture depuis Python (Temps de réponse)

```
▼ {  
  ▼ RDF_ScientificObject(inferred: true) {  
    _id  
    _type (inferred:true)  
    label  
    hasDestructionDate  
    ▼ hasFactorLevel{  
      _id  
    }  
    ▼ hasGermplasm{  
      _id  
    }  
    ▼ MG_geometry{  
      ▼ geometry{  
        type  
        coordinates  
      }  
    }  
  }  
}
```

Performances

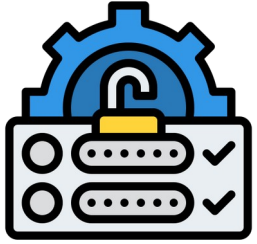
Requêtes en lecture depuis Python (Temps de réponse)



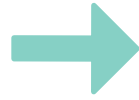
Avantages GraphQL :

- Récupérer les données via un seul point de terminaison d'API.
- Fournir un schéma global de données.
- La souplesse et la facilité d'expression des requêtes.
- La facilité de l'intégration de données de différentes sources.
- Pas d'over-fetching ou d'under-fetching.
- Documentation intégrée.
- Versionnement simplifié.

Gestion des droits d'accès



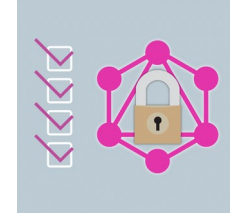
API de sécurité pour
OpenSilex



Jeton d'authentification



Requête avec en-tête
d'autorisation



Service GraphQL



Utilisateur

```
{  
  "is_admin": false,  
  "sub": "dev:id/user/account.guestopensilexorg",  
  "iss": "opensilex",  
  "name": "guest@opensilex.org",  
  "experiments_list": [  
    "dev:id/experiment/test_xp_brice",  
    "dev:id/experiment/xp_isa",  
    "dev:id/experiment/as-expe"  
  ],  
  "exp": 1697555784,  
  "given_name": "Guest",  
  "locale": "en",  
  "iat": 1697553084,  
  "family_name": "Guest",  
  "email": "guest@opensilex.org",  
  "credentials_list": []  
}
```

Variables Headers

```
1 {  
2   "Authorization":  
3   "Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.eyJpc19hZG1pbmIiOiJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.eyJpc19hZG1pbmIiOiJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.eyJpc19hZG1pbmIiOiJ0eXAiOiJKV1QiLCJhbGciOiJSUzUxMiJ9.  
4 }
```

Cas d'usage

Environment dash application

Variable_mesurée

plant_Height_estimation_mm × ▾

Variable_environnementale

Air_Température_Comptage direct_DegresCelcius × ▾

Date Range

03/15/2017 → 08/02/2017

plant_Height VS Air_Température measurements



```
# Rechercher l'historique de position d'un os (os_uri) entre (date_beginning) et (date_end)
{
  historique_positions(
    uri: $os_uri
    dateEnd: $date_end
    dateBeginning: $date_begin
  ) {
    hasEnd {
      inXSDDateTimeStamp
    }
    hasBeginning {
      inXSDDateTimeStamp
    }
    to {
      label
      _id
    }
    from {
      label
      _id
    }
  }
}
"""
```

```
# Rechercher la data de (os_uri) associée à la variable (variable_os) entre (date_beginning) et (date_end)
{
  MG_Data_findMany(
    filter: {
      target: $target,
      variable: $variable,
      _operators: {date: {gte: $date_begin}}, AND: {_operators: {date: {lte: $date_end}}}
    }
    limit: $count_data
  ) {
    date
    value
  }
}
```

```
# Rechercher la data associée à la variable (variable_facility) de chacune des facilities dans lesquelles (os_
# entre la date début et la date de fin de ce déplacement
{
MG_Data_findMany(
  filter: {
    target: $target_facility,
    variable: $result_variable_env_uri,
    _operators: {date: {gte: $begin_move}}, AND: {_operators: {date: {lte: $end_move}}}}
) {
  date
  value
}
}
```


Conclusion et Perspectives

- L'ajout des fonctionnalités d'alerte et de notification sur les changement dans les sources.
- L'utilisation du service pour des visualisations et des analyses complexes.
- Assurer une évolutivité de service selon les besoins de la recherche.
- Ajouter une couche d'intégration au niveau des instances.