

Atelier

Générer automatiquement des fiches de  
métadonnées ISO19139 valides Inspire  
avec R

SIST 2019 - Toulouse

# Rappels

## Contexte

Besoin de faciliter la création de métadonnées

## Historique

Inspire - geosource - geonetwork

## Normes

ISO 19115/139 - EML ...

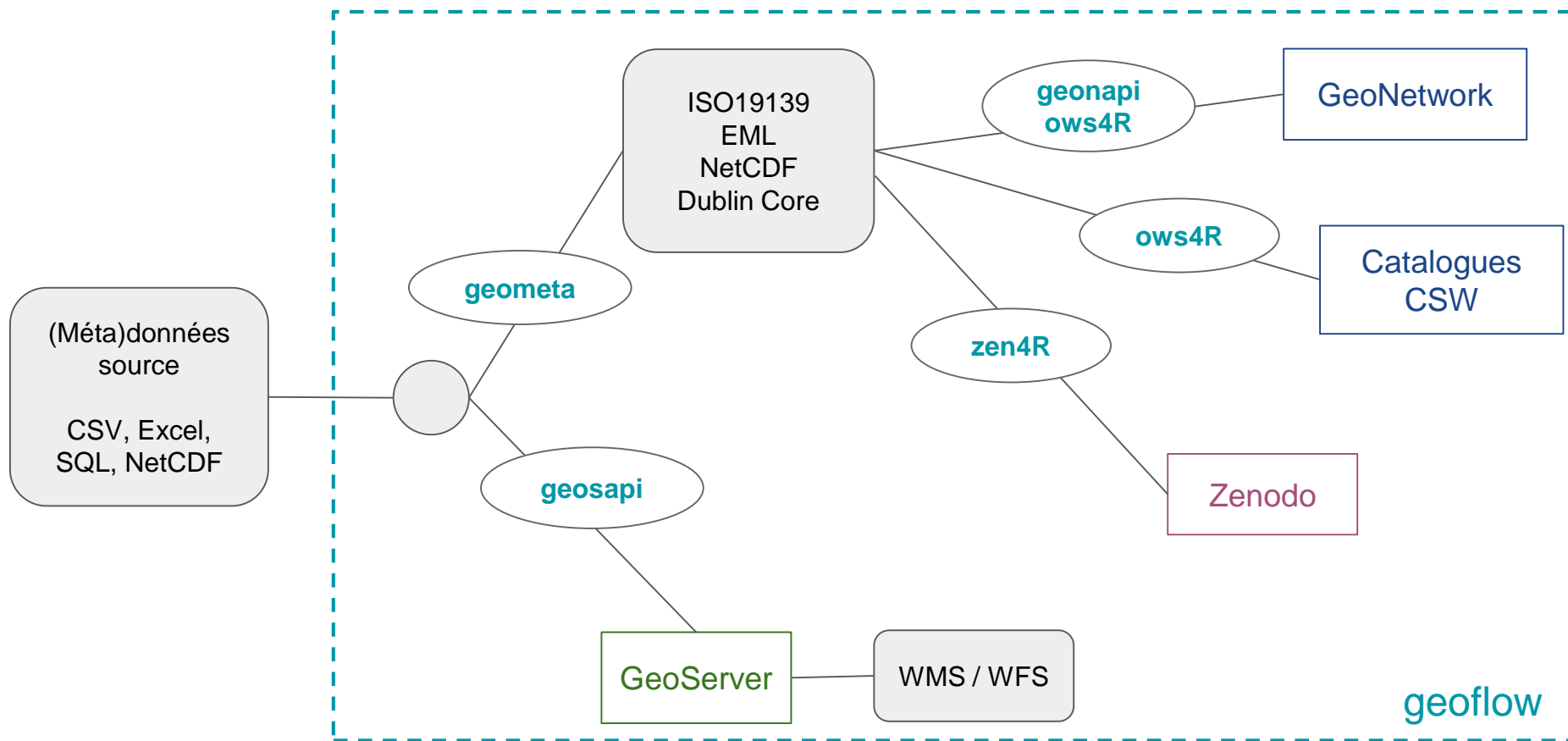
# Présentation de geoflow

## Package R

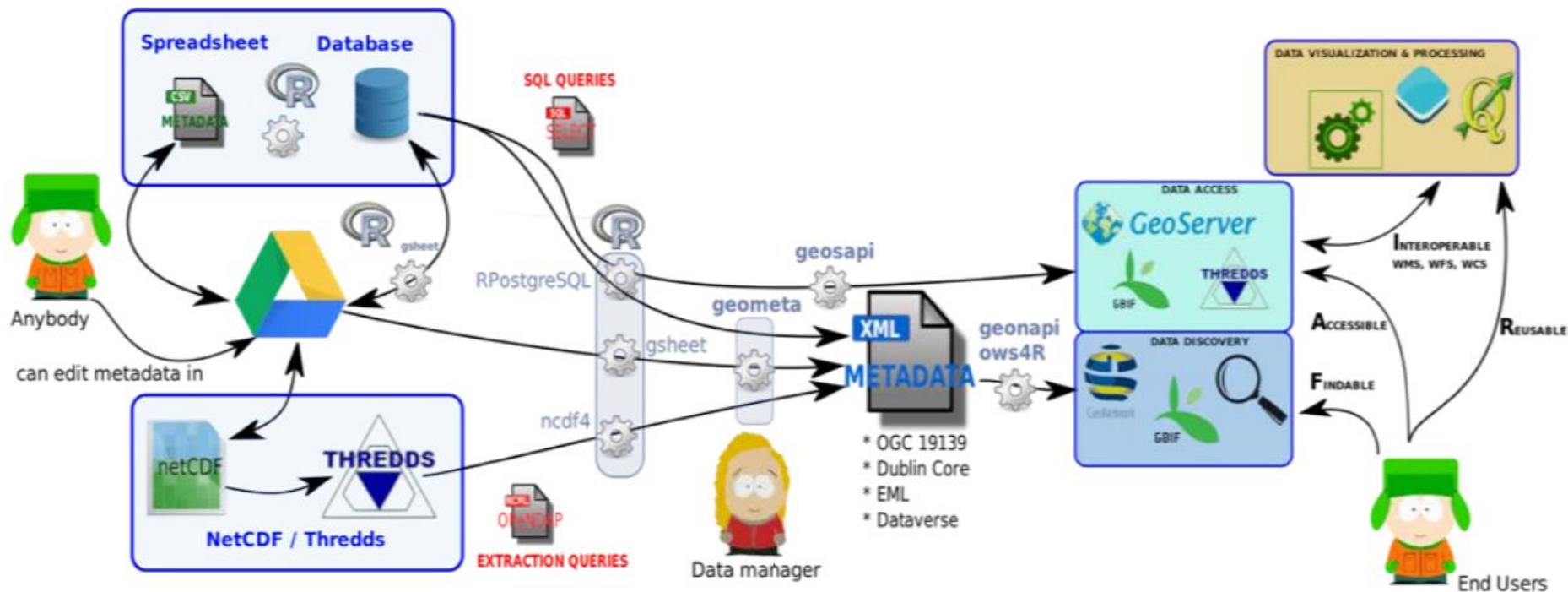
### Orchestration de différentes librairies / API

Lecture des sources	<a href="#">gsheet</a> , <a href="#">RPostgreSQL</a> , <a href="#">ncdf4</a> , ..
Ecriture de l'ISO19139	<a href="#">geometa</a>
Publication de métadonnées en CSW	<a href="#">ows4R</a> (CSW pur) ou <a href="#">geonapi</a> (Geonetwork)
Publication de flux dans Geoserver	<a href="#">geosapi</a>
Publication dans Zenodo	<a href="#">zen4R</a>
Publication dans Dataverse / Metacat	<a href="#">Atom4R</a>

# Principe de fonctionnement simplifié



# Principe de fonctionnement



D'après J. Barde

# Métadonnées source

Différentes sources possibles :

CSV / Excel / Google sheet

BDD

NetCDF ..

Dans geoflow, structure figée :

Pas d'ajout de champs possible (prévoir développements)

Certains champs obligatoires

Documentation complète à venir

[Exemple de travail : google sheet](#)

# Installation de geoflow

## Pré-requis

paquets OS

packages R => installés par geoflow

## Installation : [wiki](#)

### Github

```
require(devtools)
install_github("eblondel/geoflow", dependencies
= c("Depends", "Imports"))
```

### CRAN (à venir)

```
install.packages("geoflow")
```

# Configuration et exécution de geoflow

- 2 fichiers de métadonnées à compléter en amont :
  - description des contacts
  - description des jeux de données
- Configuration du json : [wiki](#) - [vos fichiers json personnalisés :\)](#)

Infos générales

Sources (csv, gsheets, bdd ...)

Connexion aux logiciels

Actions

wiki Note: It is planned to offer a shiny app interface, through geoflow, that will allow configure the workflow in a user-friendly manner (The shiny app will then take care of creating the appropriate JSON configuration file in a transparent way)

- Exécution

```
executeWorkflow('my_json.json')
```



# TP : configuration et exécution de geoflow

1. Compléter les 2 fichiers (ici gsheets):
    - a. [Fichier de description des jeux de données](#)
    - b. [Fichier de contacts](#)
  2. Se connecter au VRE [d4science](#) (SDI\_Lab) - login ORCID possible
  3. Ouvrir Rstudio
  4. Dans la console, définir le répertoire de travail
1. **Optionnel** : Déposer le fichier de configuration 'my\_json.json' dans votre répertoire de travail (commande upload dans fenêtre en bas à droite)
  2. Sinon, utiliser *config\_pnom.json* déjà présent sur votre espace
  3. Editer le json en paramétrant vos services (cf. [exemples fournis](#))

4. `require('geoflow')`

5. Tester “à blanc” le workflow : `initWorkflow('my_json.json')`

6. Exécuter l'unique commande de geoflow dans Rstudio

`executeWorkflow('my_json.json')`

rem : si vous devez répéter la commande, ajouter avant cette commande `rm(list=ls())` pour repartir avec une session nettoyée

# Focus sur geometa

## Fonctionnalités

Lecture / écriture vers / depuis ISO19139

Conversion vers / depuis EML et NetCDF

## Avantages à l'utiliser sans geoflow

Pas de limitation sur le format des métadonnées source (nature et syntaxe des attributs)

Customisation du workflow

## Inconvénient

Nécessite de manipuler du code R - [cf wiki](#)

# Utilisation d'Open Fair Viewer

## Principes de fonctionnement

Connexion du viewer à un catalogue CSW

Lecture des métadonnées

Spatialisation + requêtage des flux WMS/WFS existant

## Installation

code HTML + JS (sur un serveur web) + connexion GN (édition de *main.js*)

[github](#)

## Exemple

<https://tunaatlas.d4science.org/tunaatlas>