# Déploiement, gestion et exploitation d'entrepôts de données *via* R

Pour en savoir plus: https://docs.google.com/document/d/1oqf_j4PDmyeJyETK6sqRxZzMjTRIqeJ6NuI86kov6vo/edit

Paul Taconet (IRD), Julien Barde (IRD), Emmanuel Blondel (consultant géomaticien indépendant)


SIST 18
Guyancourt , 28 juin 2018

paul.taconet@ird.fr, julien.barde@ird.fr, emmanuel.blondel1@gmail.com

# Pêcheries thonières, données, science

Un cas d'école pour illustrer les besoins : les pêcheries thonières tropicales françaises

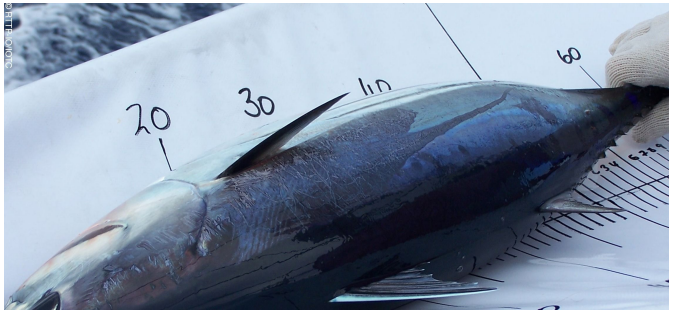




Figure: Type d'information se trouvant dans les bases de données de l'Ob7 ( © Chloé Dalleau)

Des données **hétérogènes**, **confidentielles**, gérées dans des bases de données SQL **complexes**

=> Chaque extraction requiert **l'intervention** du gestionnaire de bases de données

=> **Frein** à l'exploitation des données et la science plus globalement

# Les données au 21ème siècle

## Des constats...

Nombreuses

Complexes

Hétérogènes

Spatialisées


## … et des demandes

Transparentes

Analyses reproductibles

Ouvertes

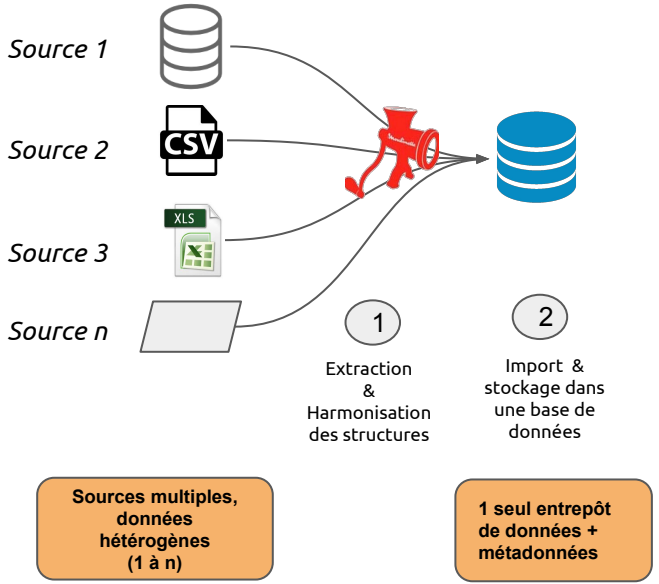
Trouvables

Accessibles

Inter - opérables

# Les entrepôts de données

Définition: “Bases de données utilisées pour collecter, ordonner, journaliser et stocker des informations provenant de bases de données opérationnelles et fournir ainsi un socle à l'aide à la décision en entreprise.” (Wikipedia)




**Bénéfices des entrepôts de données** :

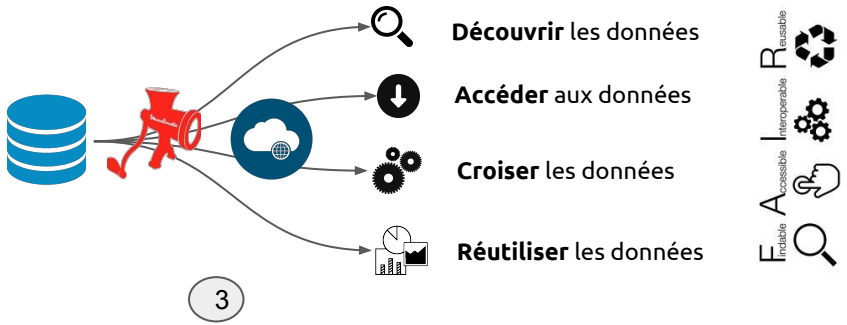
-> **Homogénéisation** des données
=> facilitation de leur croisement

-> **Centralisation** des données
=> facilitation de leur accès

-> **Anonymisation** des données

# Les données FAIR

Définition: "La notion de FAIR data (ou Fair data) recouvre les manières de construire, stocker, présenter ou publier des données de manière à permettre que la donnée soit « trouvable, accessible, interopérable et réutilisable »" (Wikipedia)







**Bénéfices des données FAIR** :

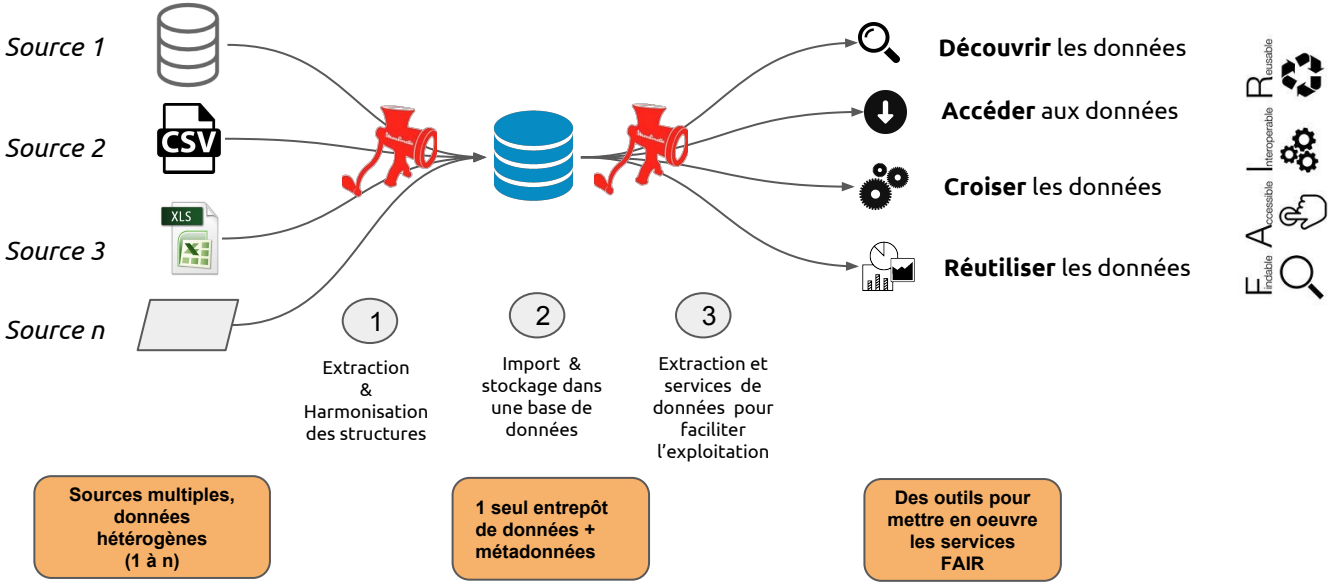
-> Données **trouvables**

-> Données **accessibles**

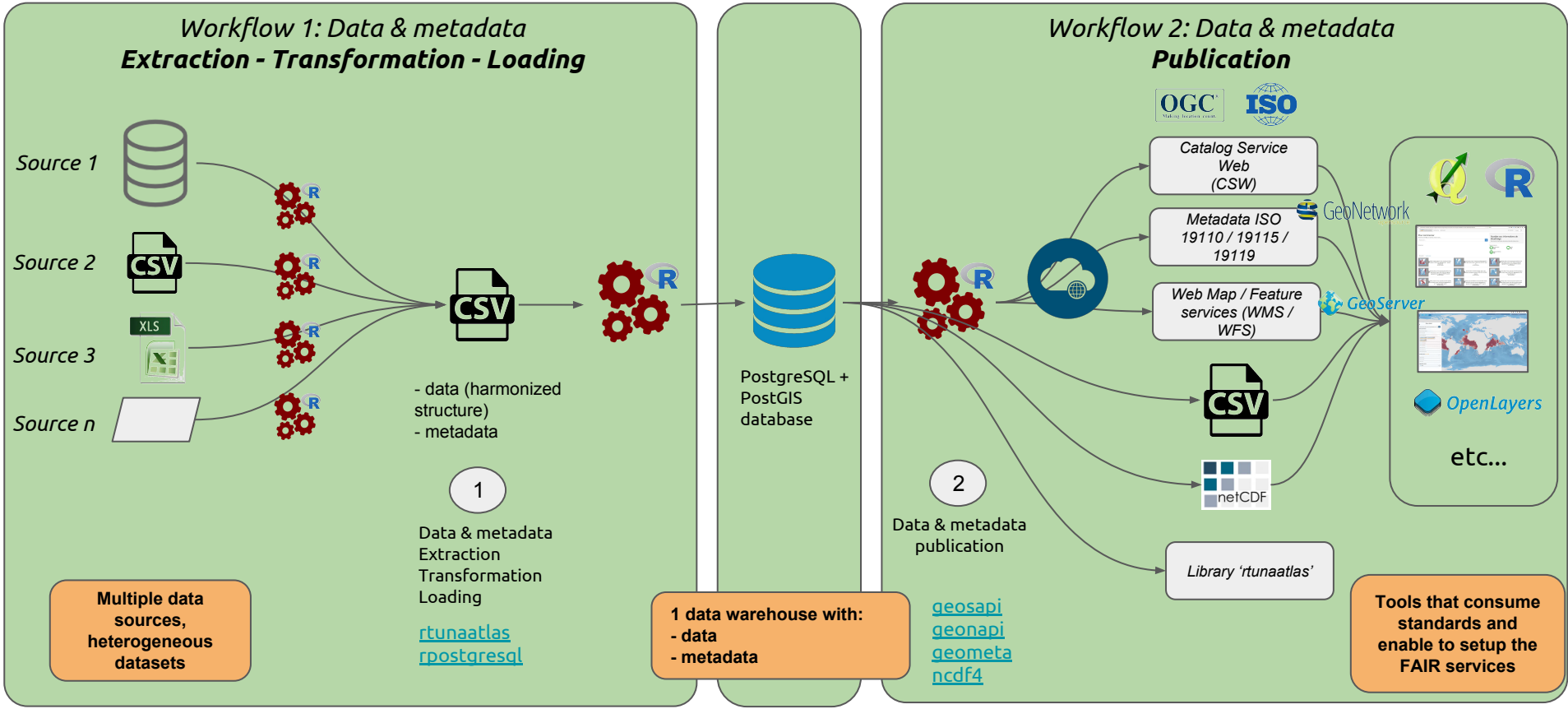
-> Données **interopérables**

-> Données **réutilisables**

# Les entrepôts de données + les données FAIR



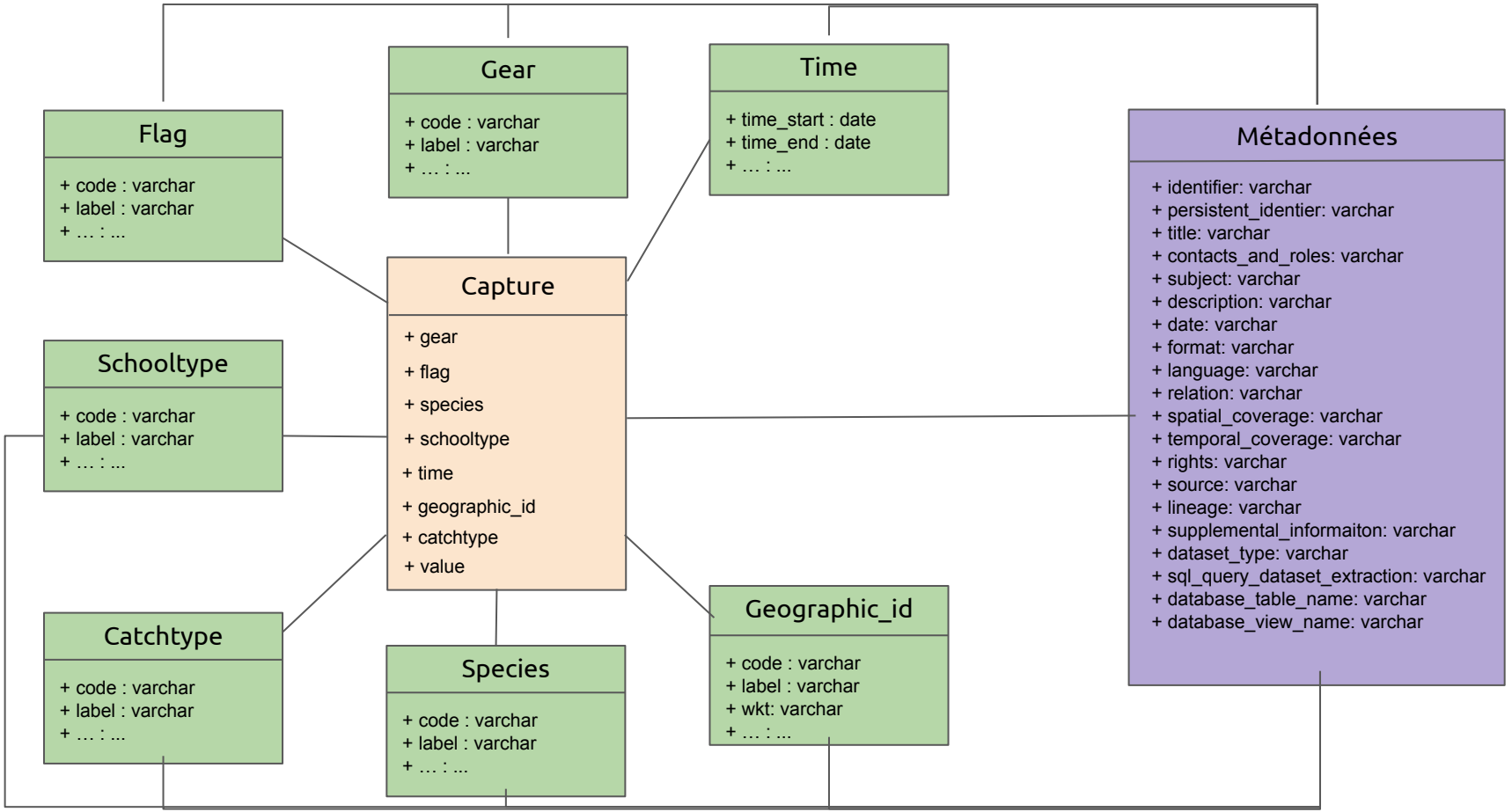
# Le workflow : des jeux de données hétérogènes aux données FAIR



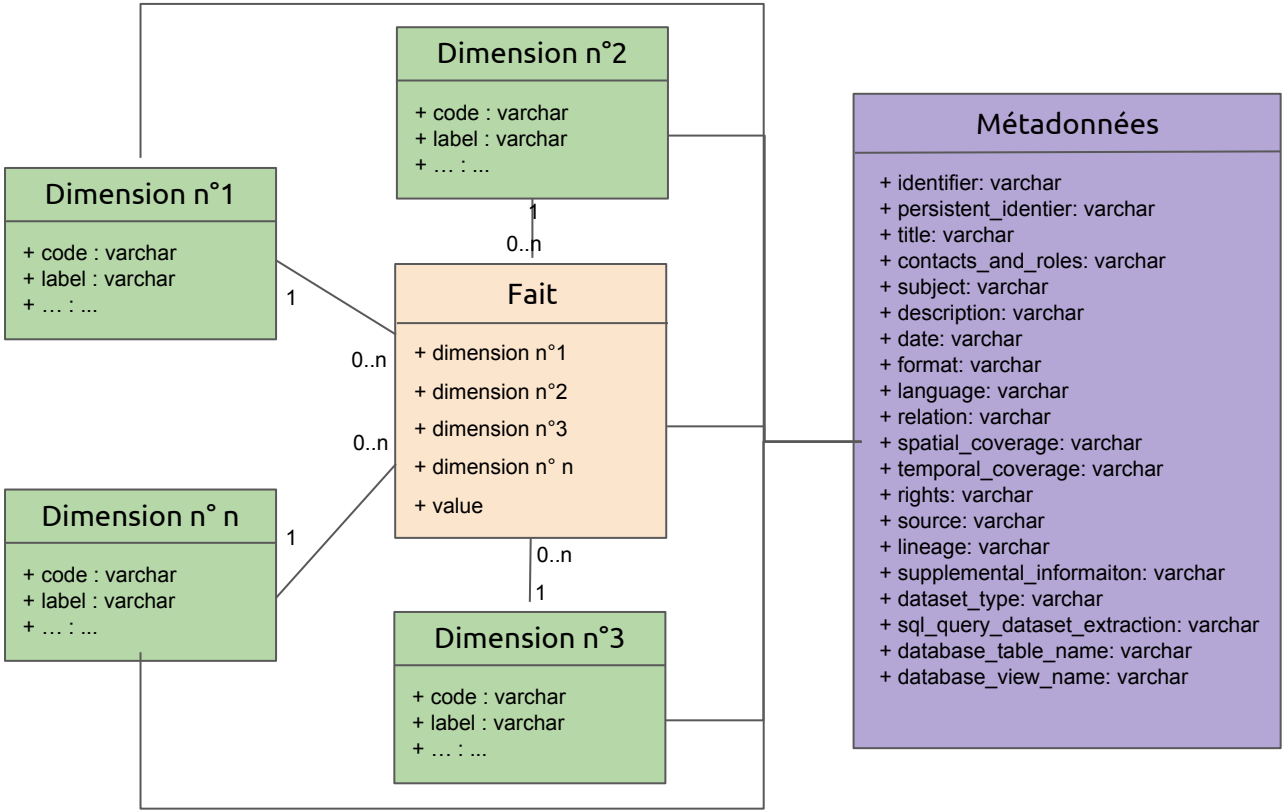
# Un exemple de données en entrée : des captures de thons

| flag | gear | geographic_identifier | time_start | time_end | species | catchtype | schooltype | unit | value |
|---|---|---|---|---|---|---|---|---|---|
| BLZ | PS | 5402000 | 2009-08-01 | 2009-09-01 | BET | C | fs | MT | 9.51 |
| BLZ | PS | 5402000 | 2009-08-01 | 2009-09-01 | YFT | C | fs | MT | 98.58 |
| BLZ | PS | 5202006 | 2009-09-01 | 2009-10-01 | BET | C | fd | MT | 0.38 |
| BLZ | PS | 5202006 | 2009-09-01 | 2009-10-01 | SKJ | C | fd | MT | 15.76 |
| BLZ | PS | 5202006 | 2009-09-01 | 2009-10-01 | YFT | C | fd | MT | 2.65 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |

# Implémentation des captures dans l'entrepôt



# L'entrepôt de données : modèle conceptuel




- 1 à ∞ faits
- 1 à ∞ dimensions
- 1 fait relié à 1 à ∞ dimensions

Exemple d'implémentation : [modèle logique de l'entrepôt de données sur les pêcheries thonières mondiales]

# Déployer le modèle physique de l'entrepôt

-> Préalable: Une base de données PostgreSQL + PostGIS vierge déployée sur un serveur

```
# Paramètres de connexion au serveur de BDD
db_name = "mon_nom_de_bdd"
db_host = "mon_hote"
db_admin_name = "mon_nom_admin"
db_admin_password = "mon_mpd_admin"
db_read_name = "mon_nom_usage"

# Liste des dimensions qui composeront l'entrepôt (noms libres)
db_dimensions <- "area,catchtype,unit,flag,gear,schooltype,sex,sizeclass,species,time,source"

# Liste des faits (variables) (noms libres) et des dimensions associées à chacun des faits
db_variables_and_associated_dimension <-
"catch=schooltype,species,time,area,gear,flag,catchtype,unit,source
@effort=schooltype,time,area,gear,flag,unit,source
@catch_at_size=schooltype,species,time,area,gear,flag,catchtype,sex,unit,sizeclass,source"

# Déploiement du modèle physique, généré dynamiquement en fonction des faits et dimensions
deploy_database_model_function(db_name,db_host,db_admin_name,db_read_name,db_admin_password,db_
dimensions,db_variables_and_associated_dimensions)
```

-> Code R disponible ici
-> Plus d'infos (documentation) ici

# Charger l'entrepôt

-> Préalables:

- Le **fait** (i.e. la variable) à charger en tant qu'objet R de type **data.frame correctement structuré** (i.e. nom des colonnes)
- Les **métadonnées** associées en tant qu'objet R de type **data.frame correctement structuré**

```
# Connexion à l'entrepôt de données
con = dbConnect(dbDriver("PostgreSQL"), dbname= "mon_nom_de_bdd", user="mon_nom_admin",
password="mon_mpd_admin", host="mon_hote")

# Lecture du jeu de données à charger (csv)
df_to_load = read.csv( "lien_vers_mon_jeu_de_donnees_a_charger.csv" )
head(df_to_load)

# Lecture des métadonnées associées (csv)
df_metadata = read.csv( "lien_vers_les_metadonnees_de_mon_jeu_de_donnees_a_charger.csv" )
head(df_metadata)

# Chargement du fait & des métadonnées dans l'entrepôt
load_raw_dataset_in_db(con_admin,df_to_load,df_metadata)
```

-> Code R disponible ici

-> Plus d'infos (documentation) ici

# Tout-en-un, et en séquence : le workflow ETL

-> Préalable: Remplir le fichier CSV de *métadonnées et paramétrisation*
- Modèle (template) de fichier disponible ici
- Exemple de fichier rempli disponible ici

Une fois rempli, le WF extrait, transforme et charge tour à tour tous les jeux de données décrits dans le fichier de *métadonnées et paramétrisation*

-> Code R disponible ici
-> Plus d'infos (documentation) ici

# Accéder aux données

```
# Connexion à l'entrepôt de données
con = dbConnect(dbDriver("PostgreSQL"), dbname= "mon_nom_de_bdd", user="mon_nom_admin",
password="mon_mpd_admin", host="mon_hote")

# Lecture de la table de métadonnées de l'entrepôt
datawarehouse_datasets_metadata <- list_metadata_datasets(con)

# Extraction d'un jeu de données stocké dans l'entrepôt
metadata_dataset_to_extract <-  list_metadata_datasets(con,
identifier="identifier_du_jeu_de_donnee_a_extraire" )

dataset_as_data_frame <-  extract_dataset(con,metadata_dataset_to_extract)
```

-> Code R disponible ici
-> Plus d'infos (documentation) ici

# Déployer les services FAIR

*Quels services ?*

**Pour chaque jeu de données stocké dans l'entrepôt :**

- Génération de **fiches de métadonnées** standards (xml):
  - **ISO 19115** : Métadonnées générales (titre, description, contacts, etc.) ;
  - **ISO 19110** : Structure du jeu de données (noms de colonnes, valeurs, etc.) ;
  - **ISO 19119** : Services disponibles (WMS/WFS, etc.)

- Insertion des métadonnées dans un **catalogue Geonetwork**

- Extraction au format **csv** et chargement sur un serveur http,

- Extraction au format **NetCDF** et chargement sur un serveur Thredds,

- Publication des couches **WMS / WFS** dans Geoserver & Thredds (WMS,WCS, OPeNDAP)

-> Utilisation des librairies R:

- développées par Emmanuel Blondel: geometa, geonapi, geosapi (Librairies pour respectivement la gestion des métadonnées géographiques , l'interfaçage Geonetwork, l'interfaçage Geoserver)
- ncdf4...

# Déployer les services FAIR

***Comment faire ?***

-> Préalables: Les serveurs installés: Geonetwork, Geoserver, Thredds

1) Paramétrer le fichier de config (json) (ci dessous: extrait des paramètres)

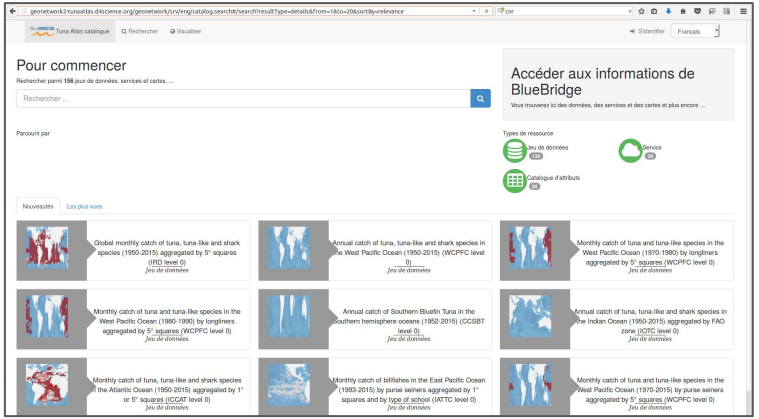
```
"geoserver": {
        "url": "http://geoserver-french-tunaatlas.d4science.org/geoserver",
        "user": "admin",
        "pwd": "***"
        },
"geonetwork": {
        "url": "http://geonetwork-french-tunaatlas.d4science.org/geonetwork/",
        "user": "admin",
        "pwd": "****",
        "version": "3.0.4"
        },
 "actions": {
        "data_wms_wfs":  true,
        "data_csv": true,
        "data_netcdf": false,
        "metadata_iso_19115": true,
        "metadata_iso_19110": true,
        "main": "write_Dublin_Core_metadata"
  }
```

2) Renseigner les jeux de données à publier dans le fichier main (SELECT * FROM metadata.metadata)

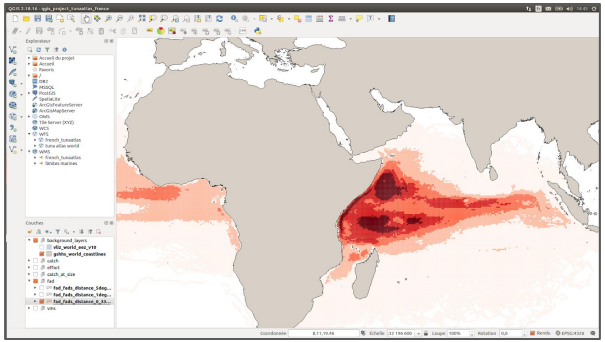
Code disponible ici

# Une IDS une fois le workflow exécuté

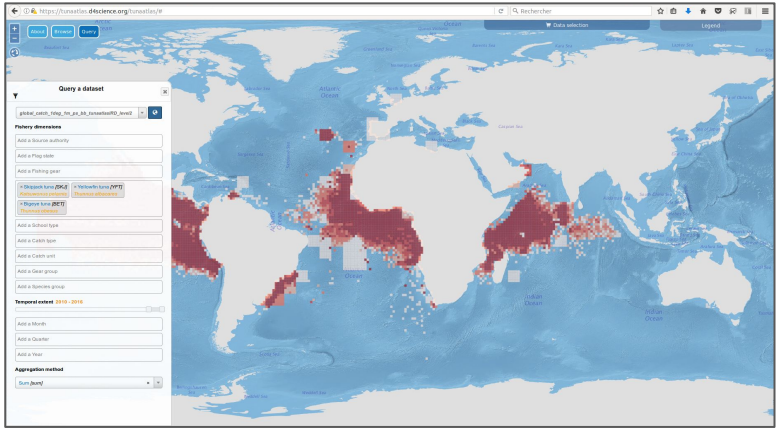
## Le catalogue de données Geonetwork




## Les Données accessibles en ligne sur QGIS



## Le viewer basé sur un catalogue OGC CSW



# Conclusion / Résumé

- Nous avons mis en oeuvre une méthode permettant de :
  - Centraliser des données hétérogènes dans un unique entrepôt de données multidimensionnelles
  - Mettre en oeuvre les services FAIR sur cet entrepôt (mais pas uniquement)

- L'ensemble des codes est développé en R et ouvert

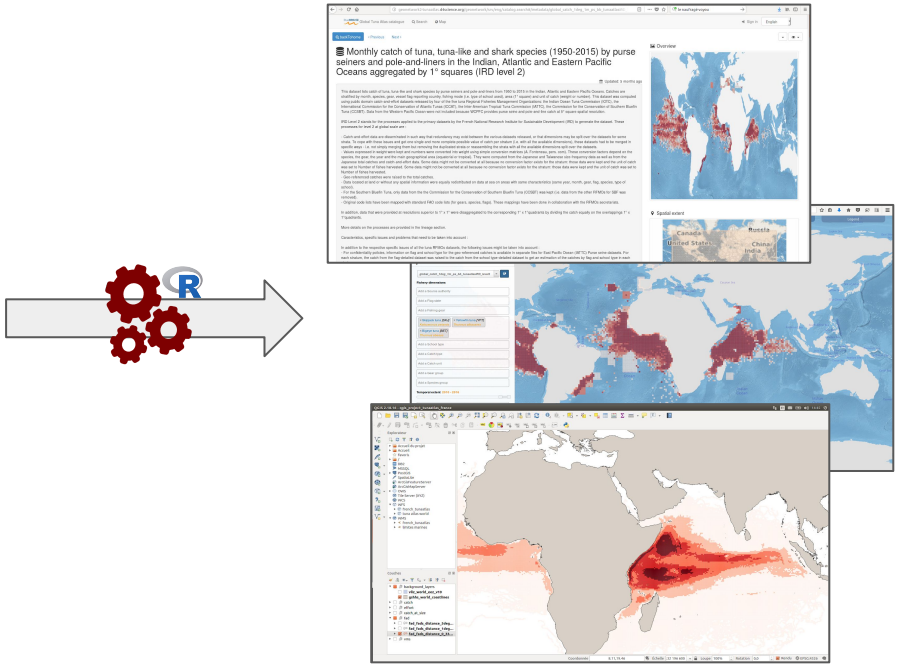
- L'entrepôt est:
  - Est flexible (i.e. adaptable aux variables et dimensions en entrée)
  - Gère différents référentiels
  - Gère les métadonnées

- Le travail restant à la charge de chacun est l'extraction et l'harmonisation des structures des données sources…(le *E* et le *T* de *ETL*)

# Conclusion

De ... → ... à :

| flag | gear | species | schooltype | time_start | time_end | geographic_identifier | catch type | unit | value |
|---|---|---|---|---|---|---|---|---|---|
| USA | LL | UNK | UNK | 1995-09-01 | 1995-09-30 | 6445035 | UNK | NO | 4.00 |
| SYC | LL | UNK | UNK | 2009-10-01 | 2009-10-31 | 6210060 | UNK | MT | 0.05 |
| MYS | LL | UNK | UNK | 2014-07-01 | 2014-07-31 | 6230045 | UNK | MT | 6.35 |
| SYC | LL | UNK | UNK | 2008-11-01 | 2008-11-30 | 6210060 | UNK | MT | 0.05 |
| MYS | LL | UNK | UNK | 2015-09-01 | 2015-09-30 | 6230045 | UNK | MT | 0.82 |
| UNK | LL | UNK | UNK | 2011-05-01 | 2011-05-31 | 6205175 | UNK | MT | 1.68 |
| MYT | LL | UNK | UNK | 2005-06-01 | 2005-06-30 | 6230045 | UNK | MT | 0.04 |



# Merci !

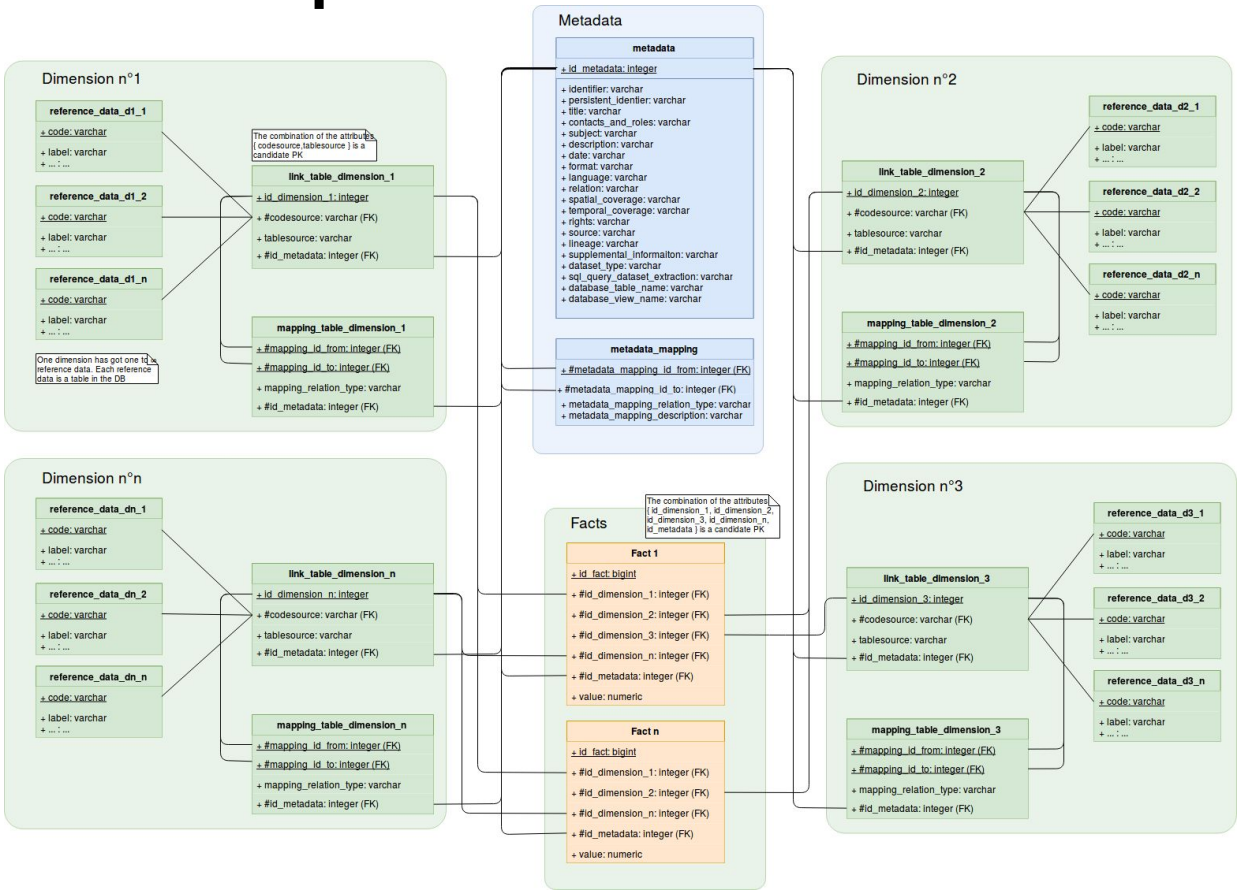


Pour en savoir plus:

Documentation de la méthode:
https://docs.google.com/document/d/1oqf_j4PDmyeJyETK6sqRxZzMjTRIqeJ6NuI86kov6vo/edit?usp=sharing

Exemple d'implémentation: les jeux de données mondiaux sur les pêcheries thonières
https://docs.google.com/document/d/1jxaE4iMiBI1TsG0Qb0siPal_1g_VHgUufCvA9DX009M/edit?usp=sharing

# Annexe: L'entrepôt de données : modèle logique




Exemple d'implémentation : [l'entrepôt de données sur les pêcheries thonières mondiales]