

Fluidification des échanges de données entre producteurs, SINP régional et national

Une solution avec l'outil GN2PG

Cendrine HOARAU, Développeuse web & Géomaticienne, CBN Alpin

Julien GIRARD-CLAUDON, Responsable du pôle SI, LPO Auvergne-Rhône-Alpes



Agir pour
la biodiversité



SINP
Système d'Information
de l'Inventaire du Patrimoine naturel



Parcours des données : du local au national

Exemple de la région Auvergne-Rhône-Alpes



SINP
Système d'Information
de l'Inventaire du Patrimoine naturel



**Producteurs
de données**

**Administrateurs
de données**

**Plateforme SINP
régionale**

**Plateforme SINP
nationale**

Réalisation des
observations de
terrain

Agrégation & Validation
par les pôles
thématiques :
FLORE & HABITAT
VERTEBRES
INVERTEBRES

Agrégation et diffusion des
données géolocalisées sur la
biodiversité



Agir pour
la biodiversité

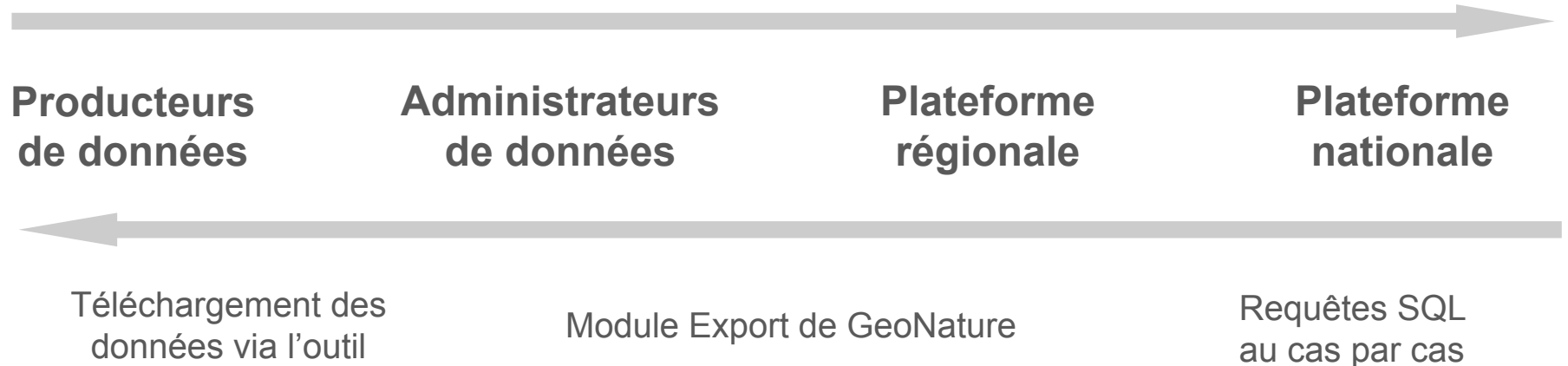


CONSERVATOIRE
BOTANIQUE NATIONAL
ALPIN

Comment sont transférées les données entre producteurs, SINP régional et SINP National ?

Manuellement : Exports CSV + Imports avec scripts bash et python ou Module Import de GeoNature

Automatiquement : GN2PG



Flux de données : limites et solutions

- Chaque structure mets en place sa solution alors que même besoin
- Evolution constante des outils, des référentiels et du matériel informatique
- Modifications de données liées à la validation non retournées systématiquement à l'échelon inférieur (ex : ajout d'UUID, rattachement taxonomique...)
- Gros volume de données

Limites

Moyens techniques et financiers

Augmentation du temps de maintenance

Multiplicité des outils, pas d'obligation quant au renvoi des données modifiées

Augmentation du temps d'exécution des requêtes

Solutions

Mutualisation des outils et/ou pratiques

Partage des connaissances et compétences

Faciliter les flux de retours des modifications effectuées

Espace de stockage performant et suffisant

GN2PG

Une solution pour faciliter les flux de données



Agir pour
la biodiversité



Génèse

- flux de données pour le SINP
- automatisation
- généricité (import dans une base postgresql)
- contraintes :
 - assurer intégrité données (modifications, métadonnées)
 - possibilité de “flux croisés”
- financement initial de la DREAL Auvergne-Rhône-Alpes pour le développement en 2020
- financement complémentaire de l'OFB (optimisation code, tests, dashboard, performances)
- un développement opensource <https://github.com/lpoaura/GN2PG/>



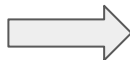
Choix techniques

- Adossé au module Export de GeoNature (utilisation de l'API)
- Programme Python installé côté destination (rien à faire côté source)
- Import automatique des données dans une base PostgreSQL (GeoNature ou autre)

Principes

GeoNature source

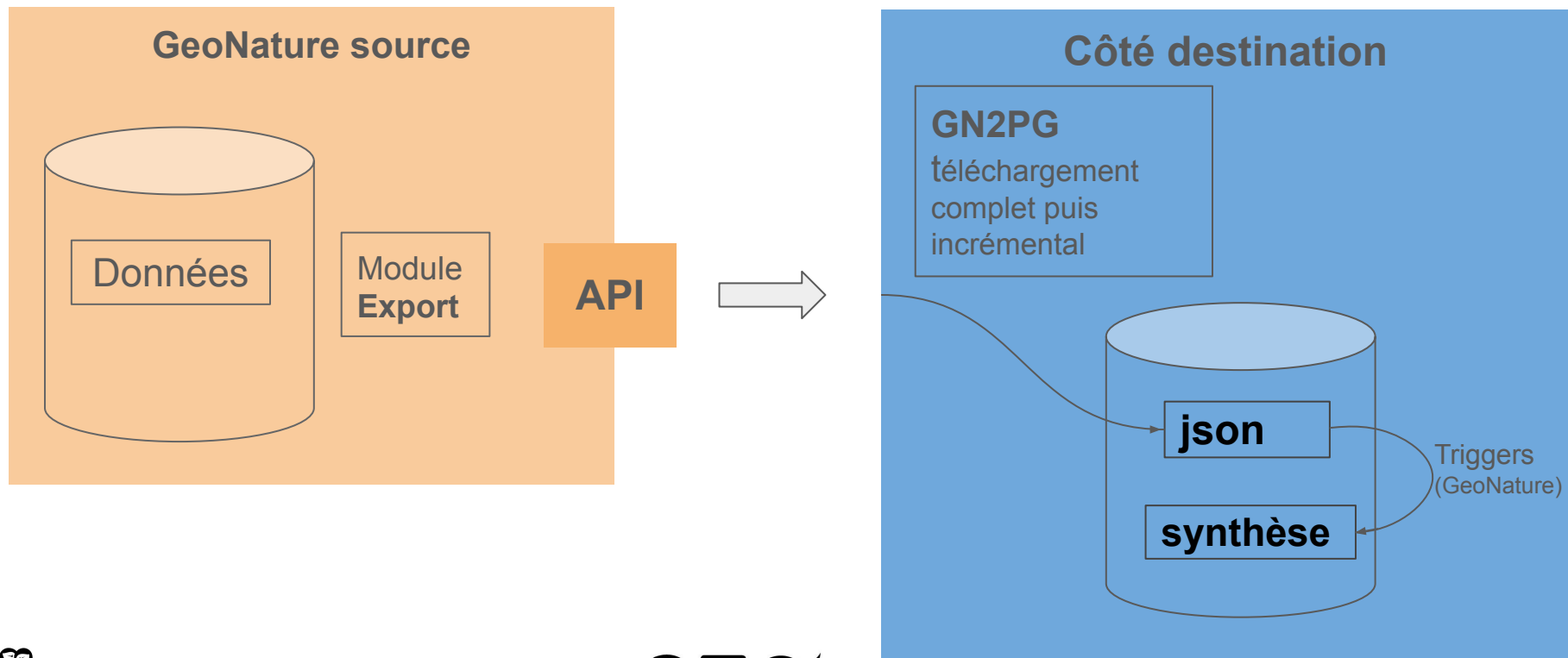
- Table / vue / vue matérialisée
- Ligne de données intégrant les métadonnées
- Triée par id_synthèse
- Possibilité exporter autre types données



Côté destination

- Programme installé sur le serveur
- Configuration de la (des) source(s) de données
- Import initial puis incrémental
- Exécution manuelle ou via une tâche cron
- Peuplement d'une table en json dans un schéma dédié
- Contrainte d'unicité sur source / numéro de donnée

Principes



Côté client

```
gn2pg_import
├── tables 4
│   ├── data_json
│   ├── download_log
│   ├── error_log
│   └── increment_log
```

```
data_json
├── columns 7
│   ├── source varchar
│   ├── controler varchar
│   ├── type varchar
│   ├── id_data integer
│   ├── uuid uuid
│   ├── item jsonb
│   └── update_ts timestamp = now()
```

```
increment_log
├── columns 3
│   ├── source varchar
│   ├── controler varchar
│   └── last_ts timestamp = now()
```

```
error_log
├── columns 6
│   ├── source varchar
│   ├── id_data integer
│   ├── controler varchar
│   ├── last_ts timestamp = now()
│   ├── item jsonb
│   └── error varchar
```

```
{
  "sexe": "6",
  "cd_nom": 65219,
  "ca_data": {
    "desc": "Données d'observation de la faune, de la flore",
    "name": "Données RNR Isles du Drac, Etang de Haute Jar",
    "uuid": "7d73b2db-c82b-4692-9fa5-c94493de8fec",
    "actors": [
      {
        "email": "",
        "identity": {
          "organism_name": "Grenoble Alpes Métropole"
        },
        "type_role": "organism",
        "uuid_actor": "6718d6f8-6db9-46be-b440-c4821e9",
        "cd_nomenclature_actor_role": "1"
      },
      {
        "email": "",
        "identity": {
          "organism_name": "Grenoble Alpes Métropole"
        },
        "type_role": "organism",
        "uuid_actor": "6718d6f8-6db9-46be-b440-c4821e9",
        "cd_nomenclature_actor_role": "8"
      },
      {
        "email": "",
        "identity": {
          "organism_name": "Grenoble Alpes Métropole"
        },
        "type_role": "organism",
        "uuid_actor": "6718d6f8-6db9-46be-b440-c4821e9",
        "cd_nomenclature_actor_role": "6"
      }
    ]
  }
}
```

Intégration à GeoNature (optionnelle)

- Gérée par 2 triggers (create/update et delete)
- Personnalisation à faire en fonction des besoins :
 - choix sur les mises à jour des données
 - valeurs par défaut (ex : validation)
 - ...
- Cohérence avec les référentiels (taxref, nomenclatures, organismes, métadonnées) :
 - ne gère pas les personnalisations de la base source
 - attention aux montées en version GN et nomenclature OLD_XXX

Retour d'expérience LPO AuRA

- Utilisation :
 - depuis 2021
 - 9 sources en import (171 000 données)
 - mise à disposition de 2 exports pour l'ORB (16 millions de données)
 - utilisation en interne entre deux instances (16 millions de données)
- Constat :
 - Rigueur dans la base de données source
 - Pas de prise en compte des modifications des métadonnées
 - Impact du stockage sur data_json (55 Go pour 16 millions de données)
 - Traitements longs sur l'ensemble de la chaîne (API - téléchargement - stockage - triggers) - env. 1 millions données / jour
- Evolutions possibles :
 - traiter les métadonnées séparément



Agir pour
la biodiversité

