

geOcom 2024 Migration vers geOrchestra 19.06.2024





Jean-Michel Crepel

DevOps pour geOrchestra à Chambéry





- → Plateforme open-source de partage de données géospatiales développée par GeoSolutions et largement diffusée dans le monde (de même que Boundless, Kartoza,...)
- Technologies utilisées :
 - Django (Python)
 - OpenLayers
 - Autres libs JS
 - GeoServer
 - PostgreSQL (PostGIS)

- Interopérabilité:
 - Supporte export ISO19139
 - API RESTful
 - WMS, WFS, WCS





Structure utilisée

Concepts - 3 types de métadonnées :

- → Layers
- Documents
- → Cartes

Forte dépendance entre les données et métadonnées :

→ Puisqu'elles sont stockées dans GS et base

Pourquoi je vous raconte tout ça

Migration depuis GeoNode vers geOrchestra pour un client

Vieille version 2.10.1 (~ nov 2019) ⇒ pas de mise à jour possible



Migration données GeoServer

Récupération et édition des fichiers du datadir GeoServer :

- → Renommage de l'entrepôt nommé "geonode" (dossier à renommer)
- → Renommage global du mot clé "geonode" : sed -e "s#geonode#nomducatalogue#g" -i \$(grep "geonode" * -r -l)
- → Changement de tous les liens/URLs fichier

Les couches sont bien renseignées C'était la partie "facile" de la migration.



Migration métadonnées

1^{er} essai

Moissonnage CSW:

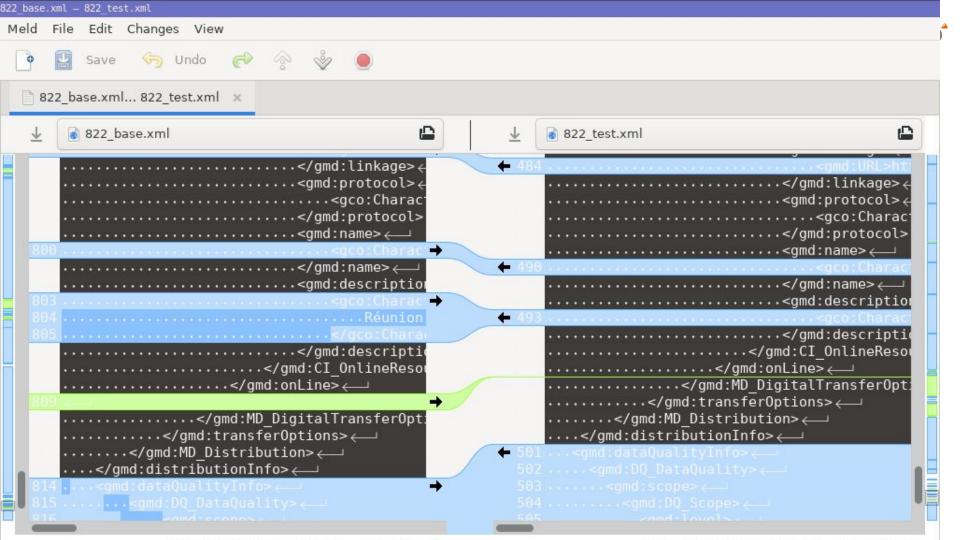
- Génération de fiches non conformes
- → Beaucoup d'informations manquantes (Mapping incomplet)
- → Format pas satisfaisant

Cette méthode n'a pas été retenue

Migration métadonnées

2e essai: à la main

- 1. Télécharger une fiche via l'interface web GeoNode (format ISO with XSL)
- 2. Importer dans GeoNetwork
- 3. Éditer via l'interface GeoNetwork
- 4. Exporter au format XML
- 5. Regarder les différences



Migration des fiches automatiques

1. Récupération de la liste des fiches (API GeoNode) en format json

```
/api/layers/?limit=1000&offset=0
/api/documents/?limit=1000&offset=0&elementsetname=full
/api/maps/?limit=1000&offset=0
```

Z. Téléchargement (API GeoNode)

```
Fiche /showmetadata/xsl/[internal_id]
Feature /layers/geonode:[layergsname]/feature_catalogue
```

Vue détaillée /catalogue/csw_to_extra_format/[uuid-meta]/[title].txt

- 3. Modification (avec regex)
- 4. Upload sur GeoNetwork (via l'api)



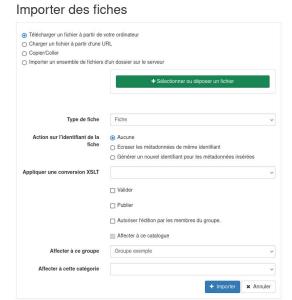
Migration des fiches automatiques

1. Exporter au format MEF (zip).

Pour ajouter tous les documents/imagettes joints aux fiches "simplement"



2. Puis réimporter le zip sur GeoNetwork



Quelques problèmes rencontrés

- 1. Documentation inexistante
- 2. Swagger GeoNetwork ne fonctionne pas
- 3. API GeoNetwork difficile à dompter
- 4. API REST GeoServer librairies avec peu de fonctionnalités

Qu'est-ce que ce travail a généré?

- → Plein plein script (un chantier ~2000 lignes) en Bash et Python
- → Découverte des API dans geOrchestra/GeoNode
- → Envie de publier mon travail sous forme de librairie Python/Script pour de futures utilisations (travail en cours)
- → Éditer de la métadonnée



















Plan

- Présentation GeoNode
- → Structure meta/data
- → Pourquoi cette migration
- → Migration des données (geoserver)
- → Migration des métadonnées
 - Solution non fonctionnelle (Moissonnage)
 - Solution remontage de manche
- → Retour sur investissement (réflexion de librairie simple pour aller discuter)

Plan1

- → Qu'est-ce que c'est geonode
- → Gestion metadonne/donne très connecté
- → Fonctionnement
- → Migration vers geOrchestra data, metadata
- → Réflexion sur l'automatisation action métier (metadata
- Metadata process
- → Gs datadir