



Observatoire déchets - Alimentation de tableaux de bord avec des flux WFS

--

geOcom, le 18 juin 2024



Ce projet est cofinancé par
l'Union européenne avec le
Fonds européen de développe-
ment régional (FEDER)



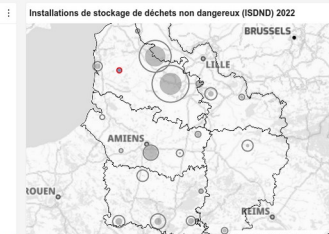
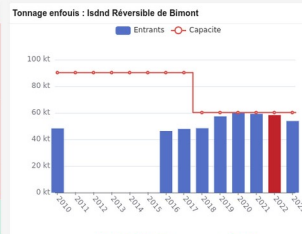
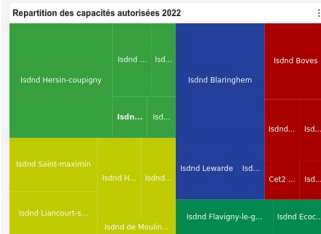
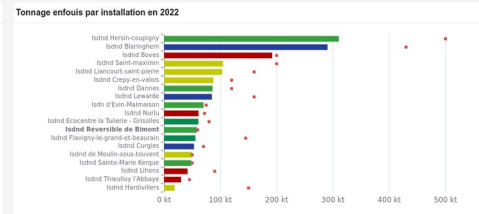
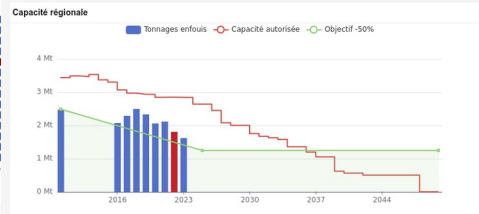
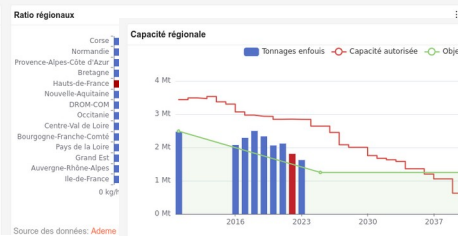
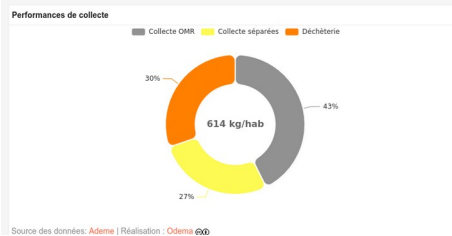
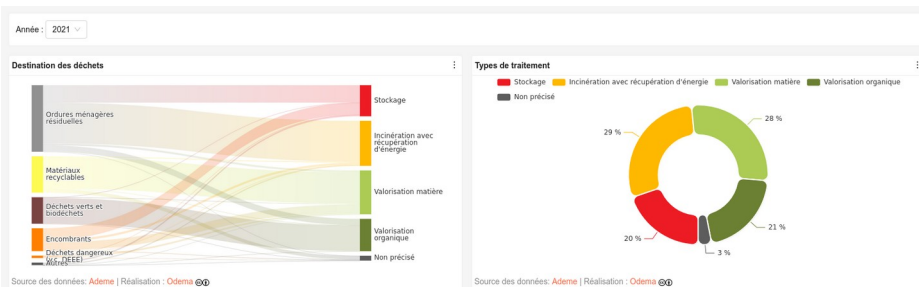
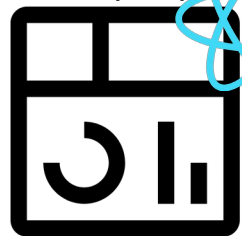


TABLEAU DE BORD ODEMA

Architecture générale

Application web
React (JS)



Carto &
dataviz



WFS



[geo2france.fr/
geoserver](https://geo2france.fr/geoserver)

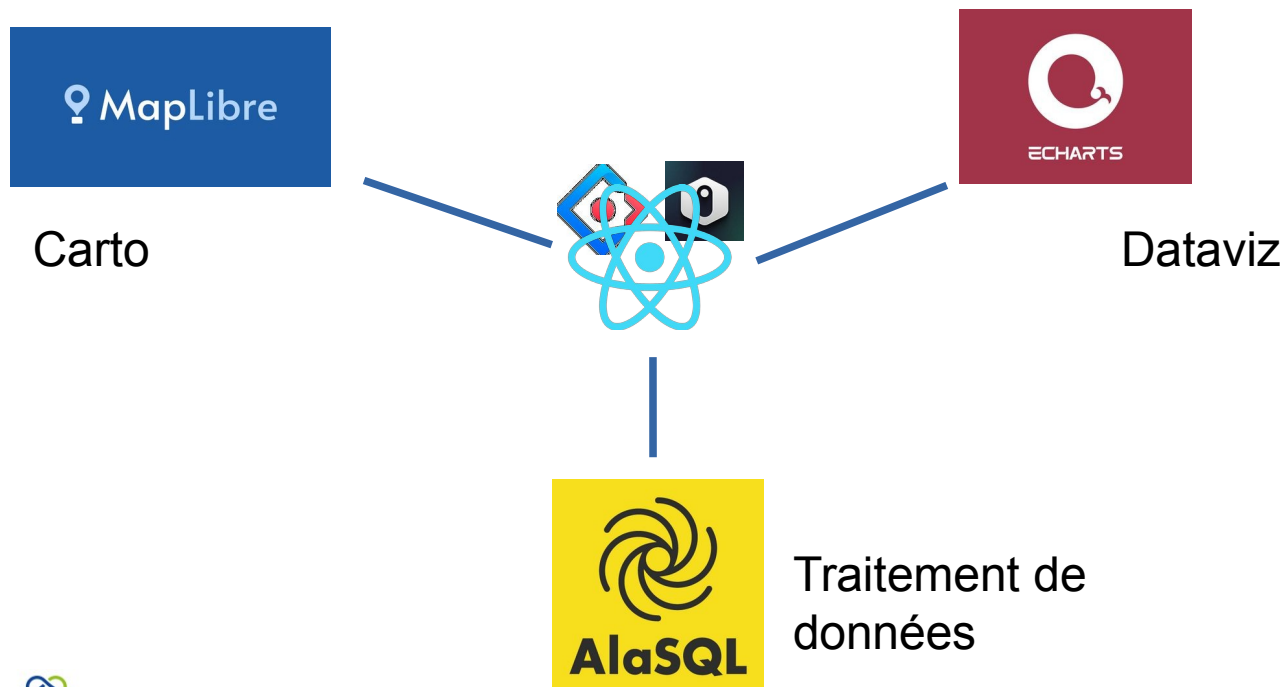
API Datafair

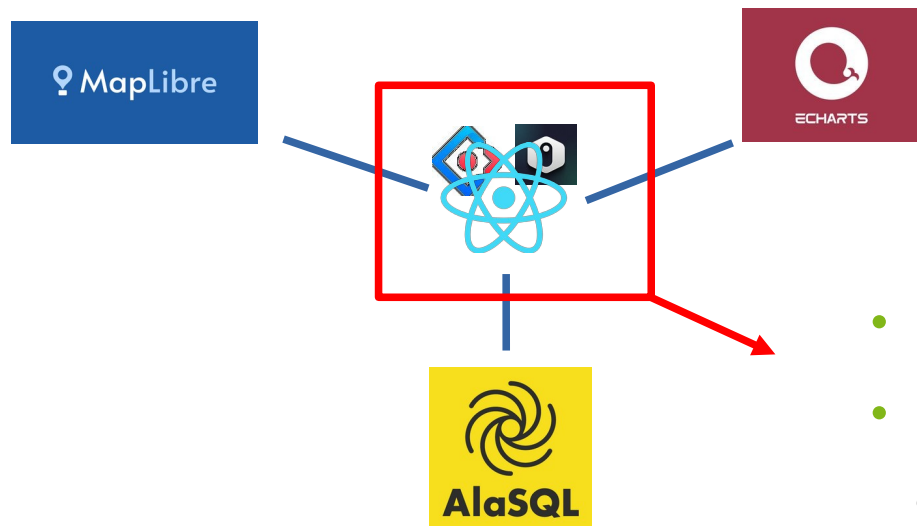


data.ademe.fr

TABLEAU DE BORD ODEMA

Architecture générale







- Connexions aux API
- Fonctionnalités génériques
 - Exports graphique & tabulaire
 - Navigation
 - UX/UI

TABLEAU DE BORD ODEMA

Connecteurs



- Bibliothèque Refine.dev
- Développement de **deux connecteurs** (*Data Provider*) Refine
 - WFS :  <https://github.com/geo2france/refine-wfs>
 - API Datafair :  <https://github.com/geo2france/refine-datafair>

Existe aussi : connecteurs *core* + *community* : GraphQL, REST, etc.

- **Génériciser** le projet → Bibliothèque de composants React pour **équiper les observatoires**.
- **Démocratiser l'accès aux non-développeurs** sans perdre la souplesse
 - Simplifier au maximum l'utilisation de la librairies
 - A ce stade, pas l'ambition d'une interface *no-code*.



Benchmark des solutions potentielles testées par Geo2France

ESRI
Metabase
Superset

Streamlit
R-shiny
DGE (développement Data Grand Est)

Librairies javascript : Apache echarts, chart.js

NO
CODE

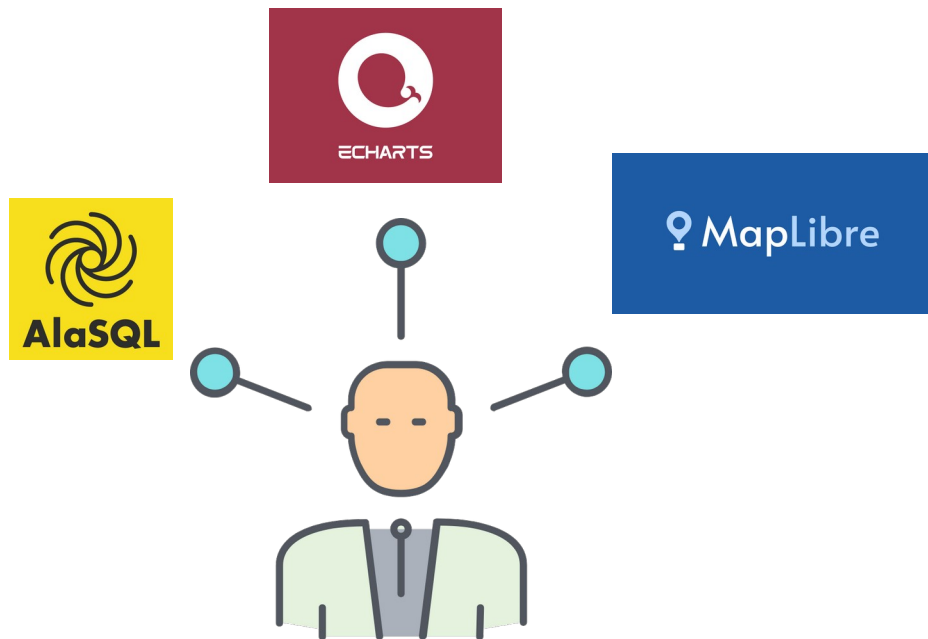
CODE



PERSPECTIVES

Géo2France Compétences nécessaires

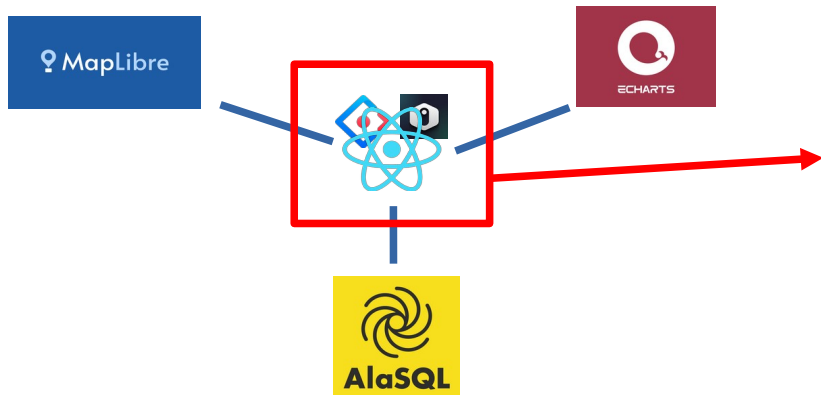
- Pré-requis administrateur :
 - Manipuler du **JSON**
 - Se référer aux documentations **ECharts** et **MapLibre**
 - Avoir des **notions** dans les **langages web**





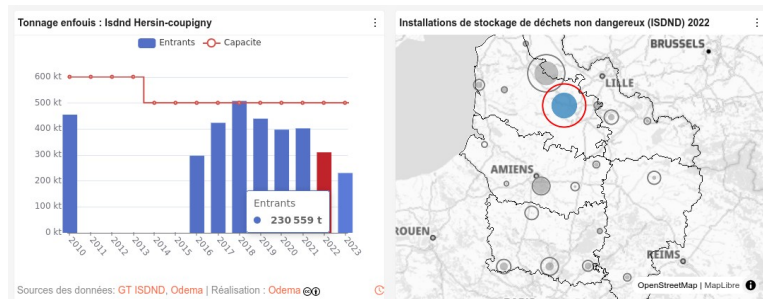
PERSPECTIVES

Géo2France Une bibliothèque de composants



- Rendre cette partie la plus simple et lisible possible
- Composant(s) (= « balise ») permettant de gérer les opérations les plus courantes
 - **Récupération et formatage des données**
 - **Exports** image/data, plein écran, habillage, crédit, etc.

```
<<<<DashboardElement title={`Installations de stockage de déchets non dangereux (ISDND) ${year}`}>>>>  
.....<MapIsdnd data={data_isdnd} year={year} aiot={aiot} onClick={(e:any) => setAiot(e.aiot)} />  
.....</DashboardElement>  
  
.....<DashboardElement title={`Tonnage enfouis : ${isdnd_name}`}>>>>  
.....attributions=[{name: 'GT ISDND'}, {name: 'Odema', url: 'https://www.geo2france.fr/public/odema/'}]]>  
.....<ChartEvolutionISDND data={data_isdnd} year={year} aiot={aiot} onClick={(e:any) => setYear(e.year)} />  
.....</DashboardElement>
```



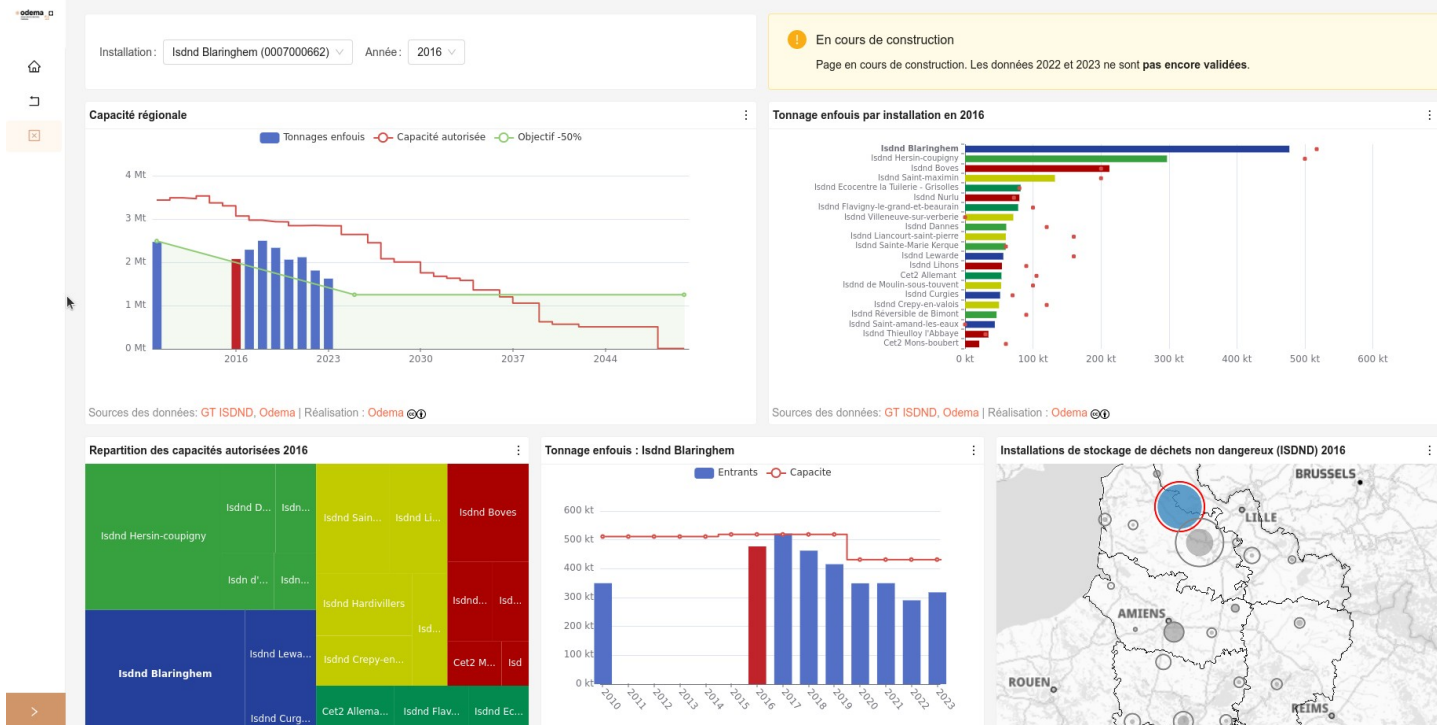


- Haut niveau de personnalisation
- Bonne **articulation data et geodata**
- Nombreuses bibliothèques libres
- *Client-side* : facilité de déploiement
- Utilise la politique de sécurité de geOrchestra
- Diversité d'API supportés
- Ré-utilisation des composants



- Compétences en développement nécessaires
- Nécessite un *backend*
- Temps de développement *relativement* long (p/r à une interface)

DEMONSTRATION



QUESTIONS / RÉPONSES



Merci pour votre attention !



[https://github.com/geo2france/
odema-dashboard](https://github.com/geo2france/odema-dashboard)