

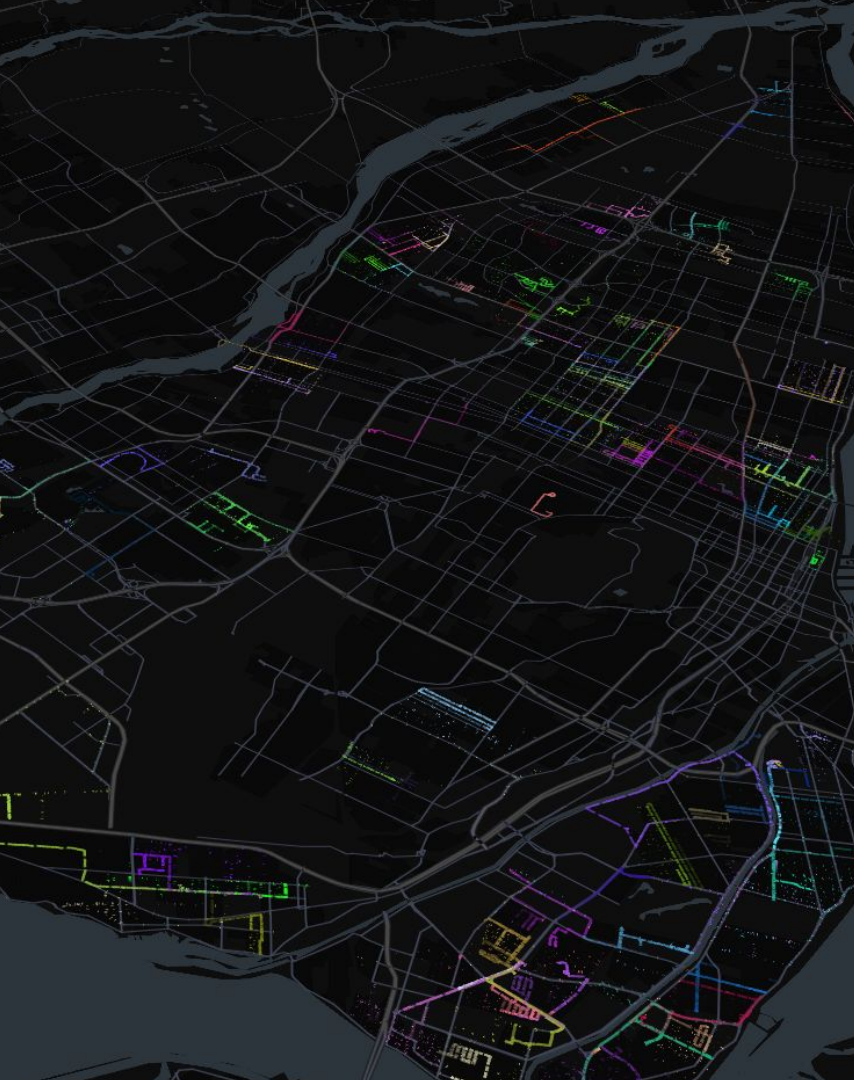


**GEO 7630**

**Intégration et visualisation de données  
géographiques**

**Semaine 3 -**

**Intégration et visualisation des données vectorielles**



# Objectifs

## 1ere partie

- Données vectorielles
- Concepts d'intégration et de visualisation
- Défis d'intégration et de visualisation
- Bonnes pratiques et exemples
- Revue des outils d'intégration
- Les tendances et les développements futurs
- TP2

## 2eme partie

- Atelier prise en main des outils du cours

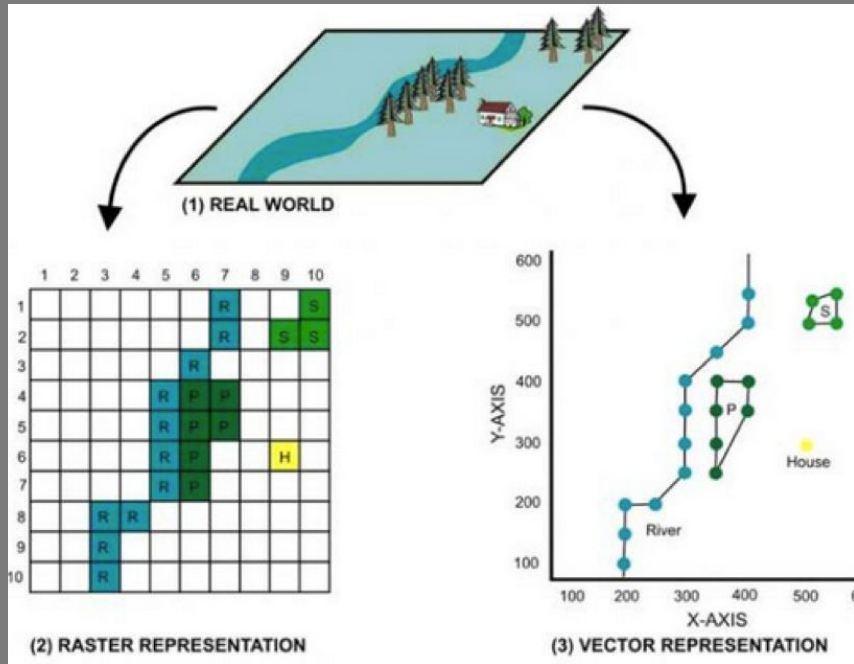
## Laboratoire

- Améliorer la densité des arbres dans les parcs

# Qu'est ce qu'une tuile vectorielle ?



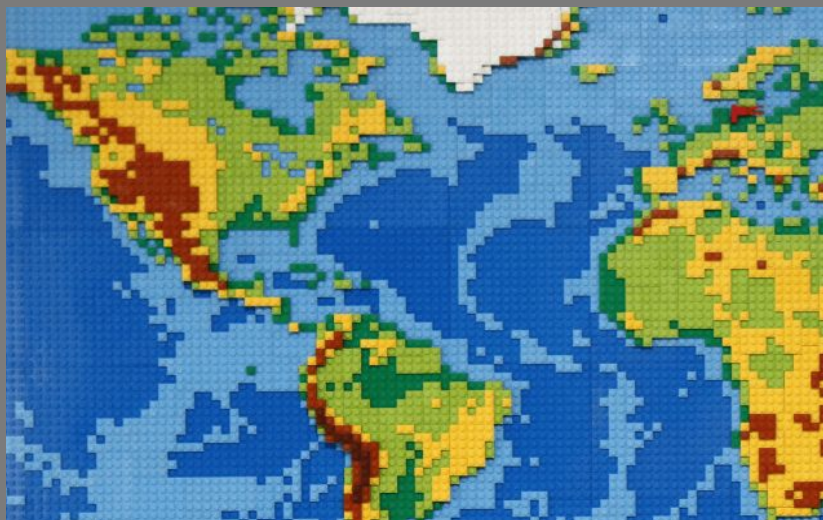
# Les tuiles vectorielles



Une tuile vectorielle est une **représentation numérique** des données géographiques qui permet de **stocker** des informations de cartographie de manière vectorielle plutôt que matricielle (raster)

Les tuiles vectorielles sont souvent utilisées pour **afficher des cartes interactives en ligne**, car elles peuvent être facilement zoomées et déplacées **sans perdre de qualité**.





# Chronologie des tuiles vectorielles

- 1980-1990

Premier paradigme scientifique sur le besoin de stocker et afficher les données vectorielles de manière efficace.

- 2005

Première preuve de concept open source

- 2010

Les développeurs utilisent des technologies telles que l'API JavaScript de Google Maps

- 2015-2020

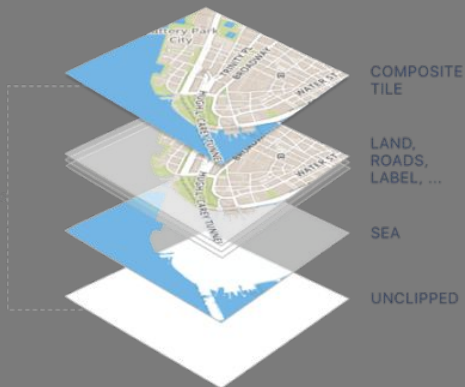
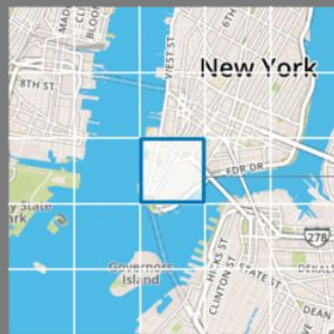
Grand boom de l'utilisation, et intégration dans les suites logiciels comme moyen de diffusion

- 2020-2024

- Standard OGC
- Utilisation du WebGL

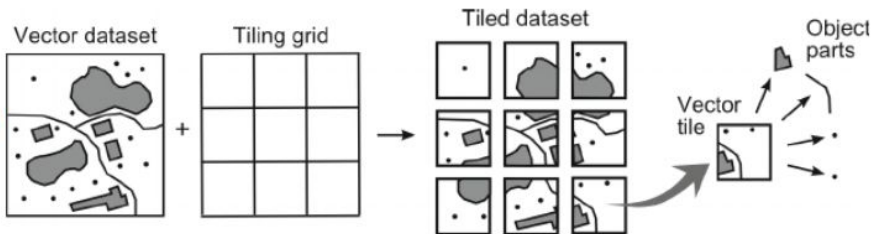
ONE MAP, MULTIPLE TILES

ONE TILE

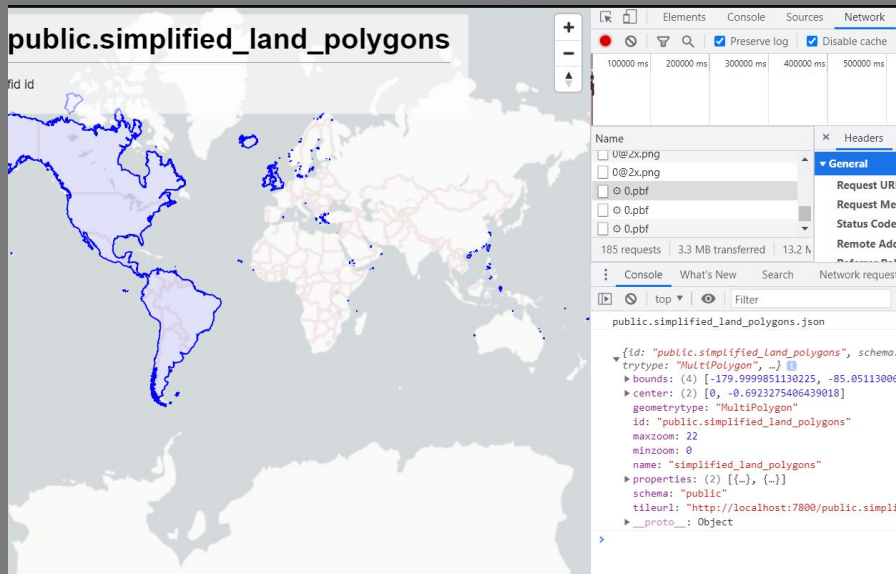


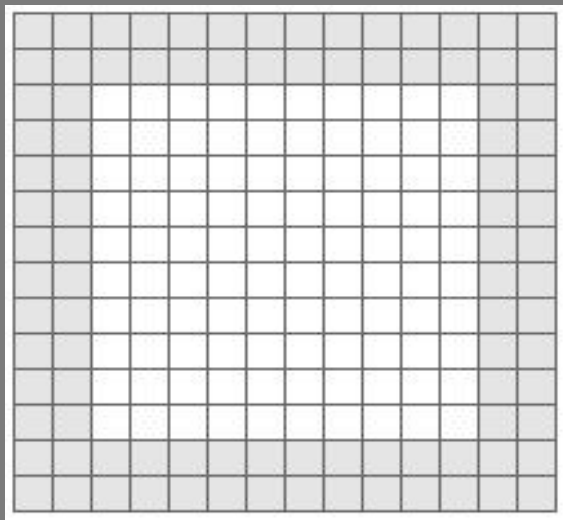
# Les tuiles vectorielles

Comment fonctionnent les tuiles vectorielles?



- Les tuiles vectorielles sont divisées en petites sections appelées tuiles, chacune contenant une partie des données géographiques.
- Lorsque l'utilisateur effectue un zoom ou un déplacement sur la carte, **seules les tuiles nécessaires sont chargées et affichées**, améliorant ainsi les **performances** et la rapidité de chargement.





# Les tuiles vectorielles

## Encodage des géométries

Pour **encoder** des informations géographiques dans une tuile vectorielle, un outil doit convertir les coordonnées géographiques, telles que la **latitude et la longitude**, en **coordonnées de grille de tuiles vectorielles**.

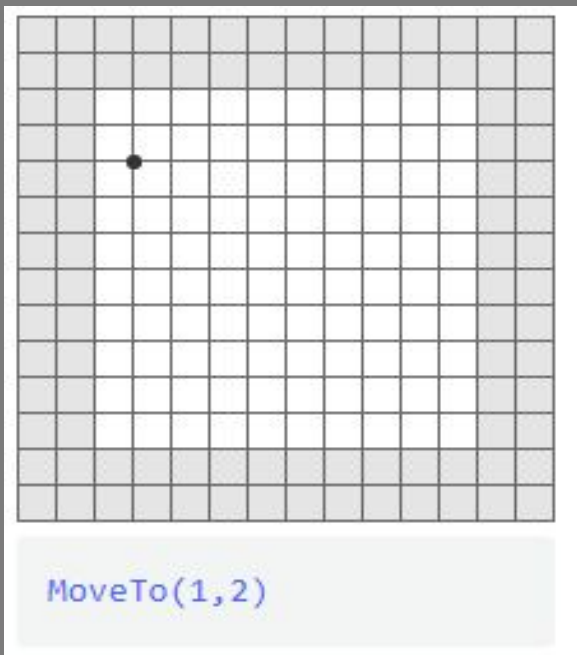
Les tuiles vectorielles ne contiennent aucun concept d'information géographique. Ils encodent des points, des lignes et des polygones sous **forme de paires x/y** par rapport au haut à gauche de la grille de manière droite vers le bas.

# Les tuiles vectorielles

## Encodage des géométries

Pour **encoder** des informations géographiques dans une tuile vectorielle, un outil doit convertir les coordonnées géographiques, telles que la **latitude et la longitude**, en **coordonnées de grille de tuiles vectorielles**.

Les tuiles vectorielles ne contiennent aucun concept d'information géographique. Ils encodent des points, des lignes et des polygones sous **forme de paires x/y** par rapport au haut à gauche de la grille de manière droite vers le bas.



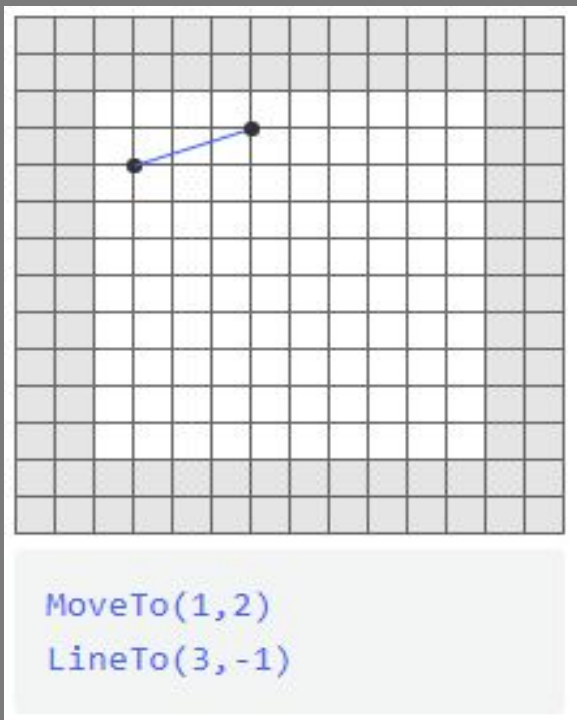


# Les tuiles vectorielles

## Encodage des géométries

Pour **encoder** des informations géographiques dans une tuile vectorielle, un outil doit convertir les coordonnées géographiques, telles que la **latitude et la longitude**, en **coordonnées de grille de tuiles vectorielles**.

Les tuiles vectorielles ne contiennent aucun concept d'information géographique. Ils encodent des points, des lignes et des polygones sous **forme de paires x/y** par rapport au haut à gauche de la grille de manière droite vers le bas.

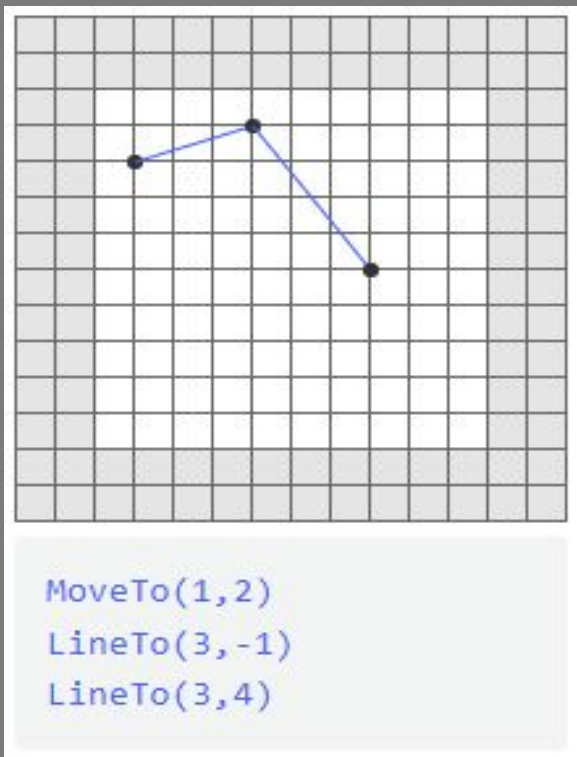


# Les tuiles vectorielles

## Encodage des géométries

Pour **encoder** des informations géographiques dans une tuile vectorielle, un outil doit convertir les coordonnées géographiques, telles que la **latitude et la longitude**, en **coordonnées de grille de tuiles vectorielles**.

Les tuiles vectorielles ne contiennent aucun concept d'information géographique. Ils encodent des points, des lignes et des polygones sous **forme de paires x/y** par rapport au haut à gauche de la grille de manière droite vers le bas.

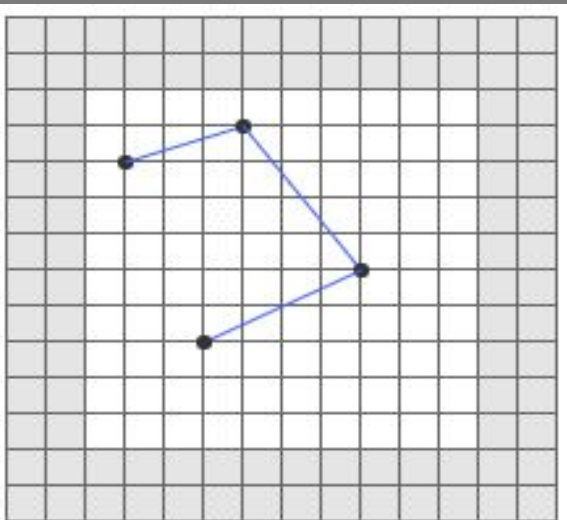


# Les tuiles vectorielles

## Encodage des géométries

Pour **encoder** des informations géographiques dans une tuile vectorielle, un outil doit convertir les coordonnées géographiques, telles que la **latitude et la longitude**, en **coordonnées de grille de tuiles vectorielles**.

Les tuiles vectorielles ne contiennent aucun concept d'information géographique. Ils encodent des points, des lignes et des polygones sous **forme de paires x/y** par rapport au haut à gauche de la grille de manière droite vers le bas.



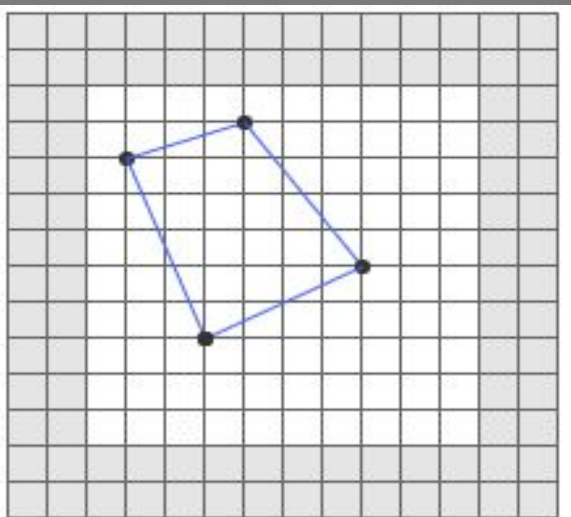
```
MoveTo(1,2)
LineTo(3,-1)
LineTo(3,4)
LineTo(-4,2)
```

# Les tuiles vectorielles

## Encodage des géométries

Pour **encoder** des informations géographiques dans une tuile vectorielle, un outil doit convertir les coordonnées géographiques, telles que la **latitude et la longitude**, en **coordonnées de grille de tuiles vectorielles**.

Les tuiles vectorielles ne contiennent aucun concept d'information géographique. Ils encodent des points, des lignes et des polygones sous **forme de paires x/y** par rapport au haut à gauche de la grille de manière droite vers le bas.



```
MoveTo(1,2)
LineTo(3,-1)
LineTo(3,4)
LineTo(-4,2)
ClosePath()
```

```

00000000 00 00 00 0d - length in bytes of the BlobHeader in network-byte order
00000000 0a - S 1 'type'
00000000 09 - length 9 bytes
00000000 4f 53 4d 48 65 61 64 65 72 - "OSMHeader"
00000000 18 - V 3 'datasize'
00000010 7c - 124 bytes long
00000010 10 - V 2 'raw_size'
00000010 71 - 113 bytes long
00000010 1a - S 3 'zlib_data'
00000010 78 - length 120 bytes

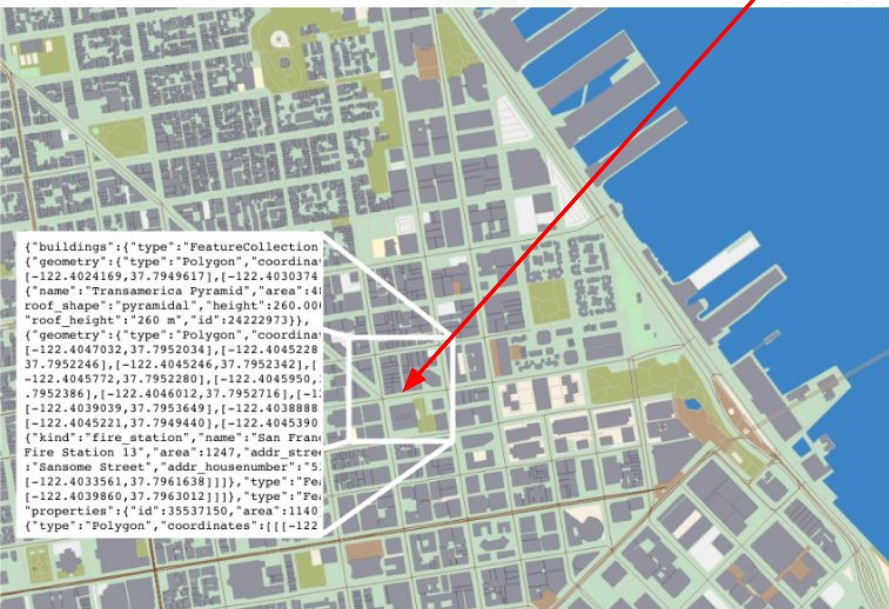
```

--- compressed section:

```

00000010 78 9c e3 92 e2 b8 70 eb da 0c 7b |.q.xx....p...{
00000020 81 0b 7b 7a ff 39 49 34 3c 5c bb bd 9f 59 a1 61 |..{z.9I4<...Y.a
00000030 ce a2 df 5d cc 4a 7c fe c5 b9 c1 c9 19 a9 b9 89 |...].J|.....
00000040 ba 61 06 7a 66 4a 5c 2e a9 79 c5 a9 7e f9 29 a9 |.a.zfJ\..y..-.)|.
00000050 c5 4d 8c fc c1 7e 8e 01 c1 1e fe 21 ba 45 46 26 |.M...-.....!..EF&
00000060 96 16 26 5d 8c 2a 19 25 25 05 56 fa fa e5 e5 e5 |..&].*.*%.V.....
00000070 7a f9 05 40 a5 25 45 a9 a9 25 b9 89 05 7a f9 45 |z...@.tE..t...z.E
00000080 e9 fa 89 05 99 fa 40 43 00 c0 94 29 0c

```



```

{"buildings":{"type":"FeatureCollection",
{"geometry":{"type":"Polygon","coordinates":
[[-122.4024169,37.7949617],[[-122.4030374
{"name":"Transamerica Pyramid","area":41
"roof_shape":"pyramidal","height":260.00
"roof_height":260 m","id":"24222973}}},
{"geometry":{"type":"Polygon","coordinates":
[[-122.4047032,37.7952034],[[-122.4045228
37.7952246],[[-122.4045246,37.7952342],[
-122.4045772,37.7952280],[[-122.4045950,
.7952386],[[-122.4046012,37.7952716],[[-1
-122.4039039,37.7953649],[[-122.4038888
[-122.4045221,37.7949440],[[-122.4045390
{"kind":"fire station","name":"San Fran
Fire Station 13","area":1247,"addr_stre
"Sansone Street","addr_housenumber":5
[-122.4033561,37.7961638]]]], "type":"Pe
[-122.4039860,37.7963012]]]], "type":"Pe
"properties":{"id":"35537150","area":1140
"type":"Polygon","coordinates":[[-122

```

# Les tuiles vectorielles

## Formats

.pbf (Protocolbuffer Binary Format) (OSM)  
 .mvt (Mapbox Vector Tile)

La principale différence entre les deux est que .pbf est un format générique utilisé pour stocker des données vectorielles, tandis que .mvt est un format spécifique développé par Mapbox

En général, les fichiers .mvt sont plus compacts et plus performants que les fichiers .pbf car ils sont optimisés pour une utilisation avec les bibliothèques de cartographie de Mapbox.

[Vector tile specification](#)

UQÀM

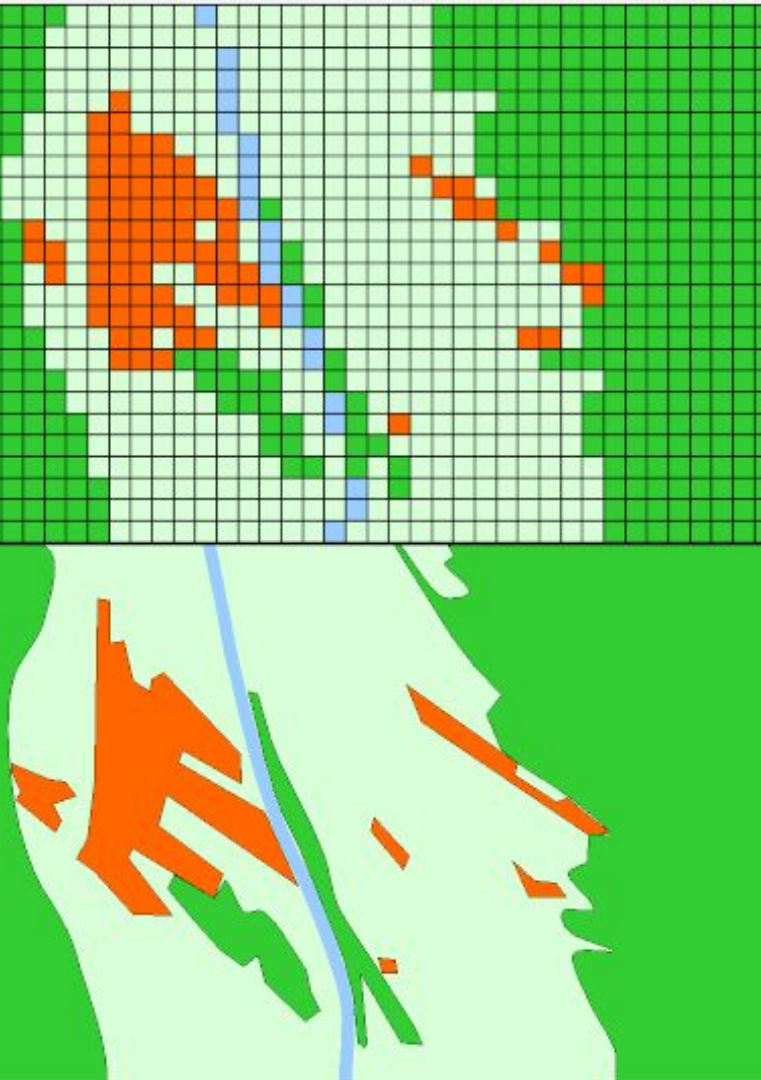




# Les tuiles vectorielles

## Avantages des tuiles vectorielles

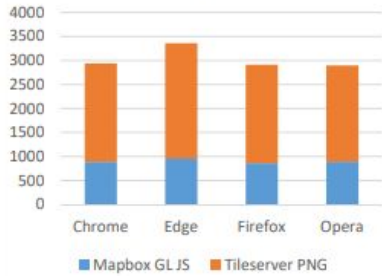
- **Qualité d'image élevée** lors du zoom et du déplacement
- **Chargement rapide** des données
- Capacité à **afficher** des données dynamiques en **temps réel**
- **Stylisation** à la volée
- **Interrogation** des données à la volée



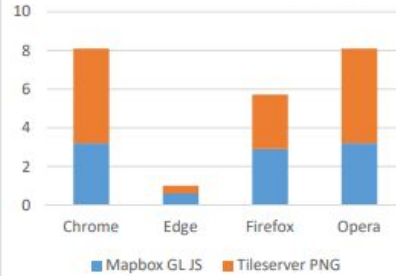
# Quelles sont les différences avec les tuiles matricielles ?

- Le volume d'une tuile
- L'information stockée
- La simplification
- La résolution
- L'interactivité
- Données vivantes
- Style dynamique
- Temps de processing

Cross-browser testing  
on loading time (ms)

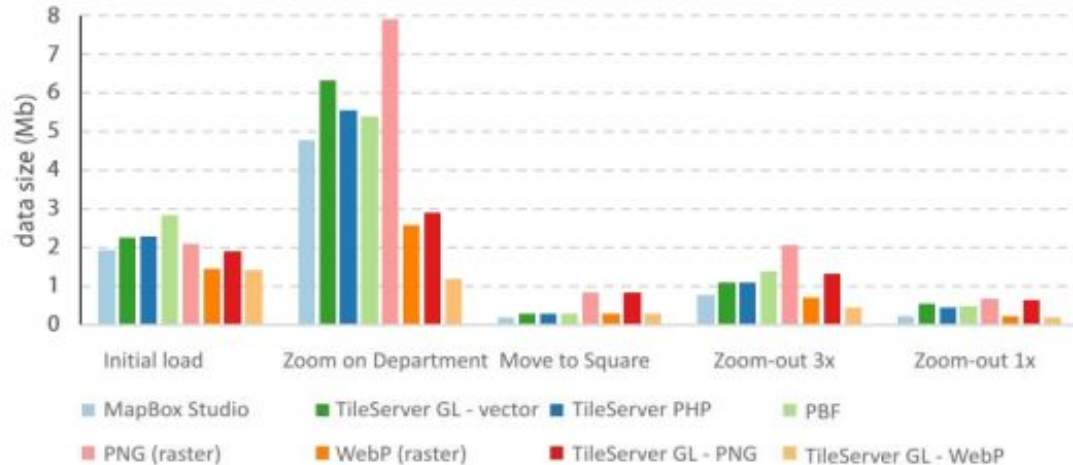


Cross-browser testing  
on transferred data (MB)



3 x plus rapide à charger dans le web

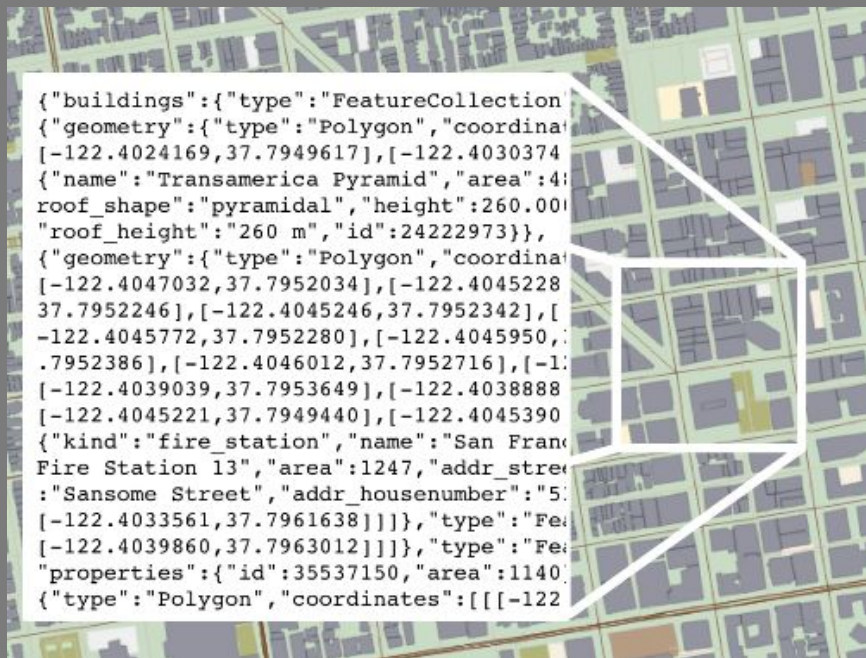
Size of downloaded data  
fast Internet connection



25% à 50% plus léger

[Performance Testing on Vector vs. Raster Map Tiles—Comparative Study on Load Metrics](#)

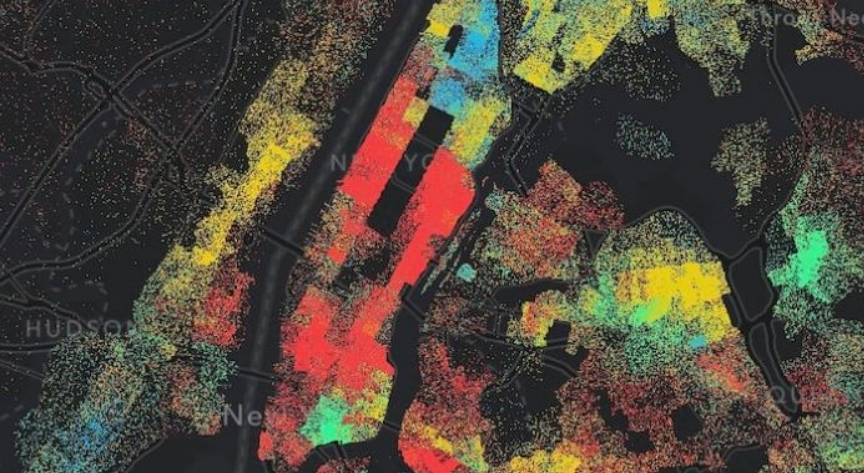
# Les tuiles vectorielles



## L'information stockée

Les **propriétés** et la **géométrie** sont **stockées** sous forme de **protocole binaire** pour minimiser la taille mais aussi faciliter le traitement de l'information par l'ordinateur (bas niveau 0 et 1)

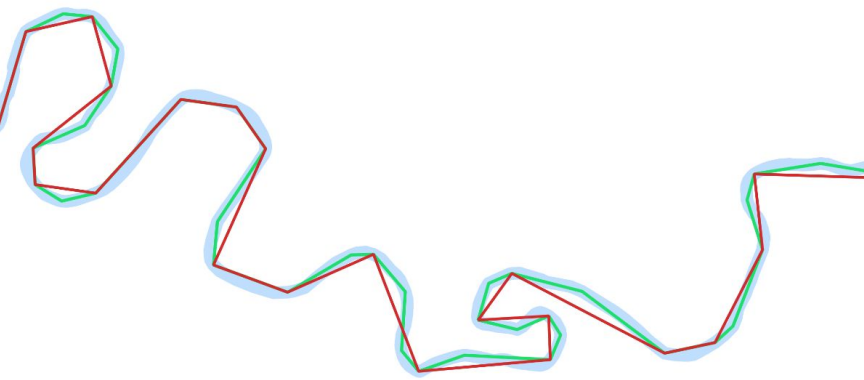




# Les tuiles vectorielles

## Simplification

Une tuile pour ne pas être trop lourde, est simplifiée, allégée par la **réduction de la quantité de vertex** qu'elle contient.

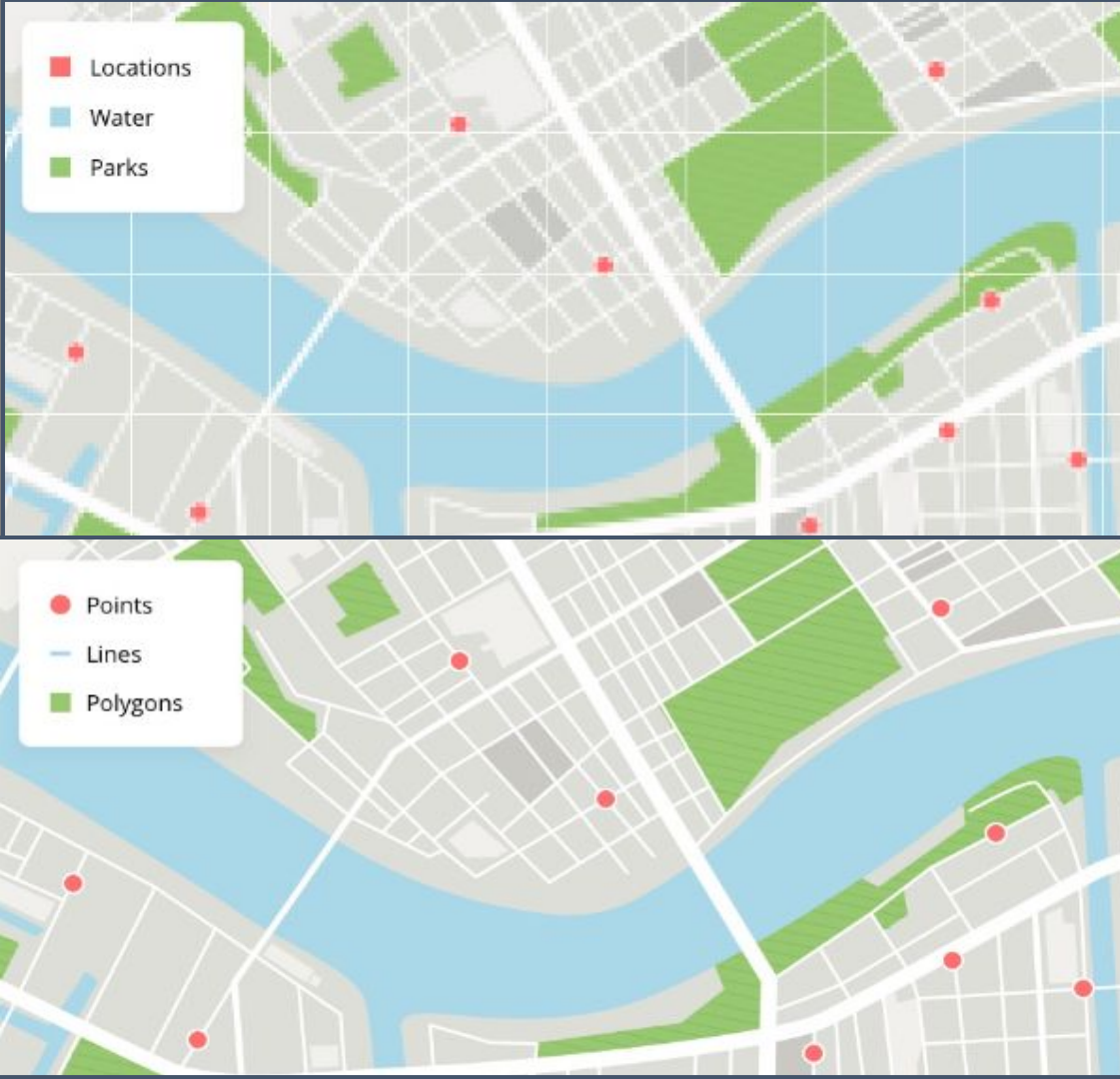


### Legend

- 100m tolerance
- 50m tolerance
- Original

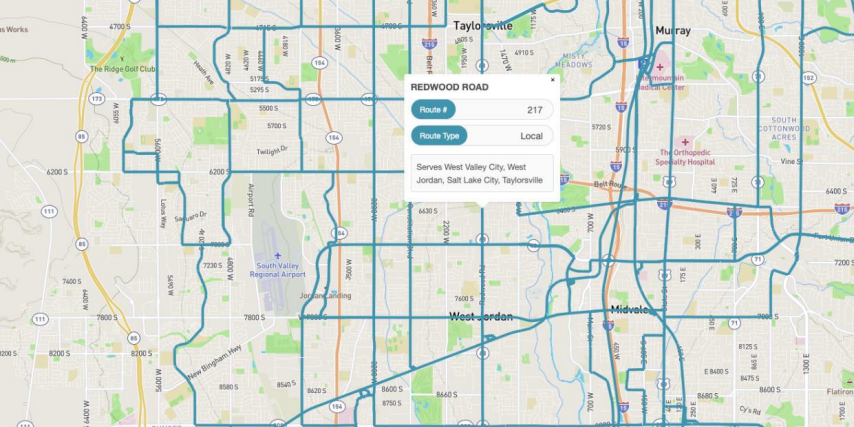
0 200 400 600 800 m





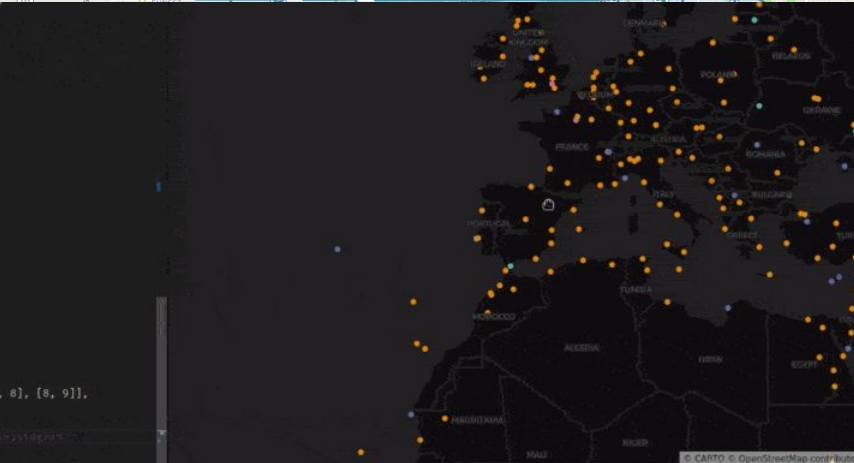
# Résolution

- Interactivité
- Précision
- Tuilage



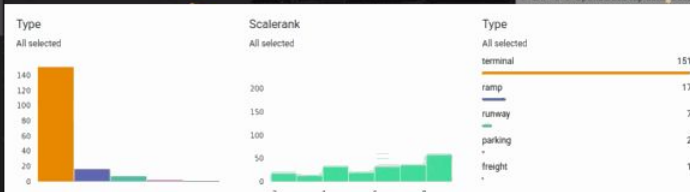
# L'interactivité

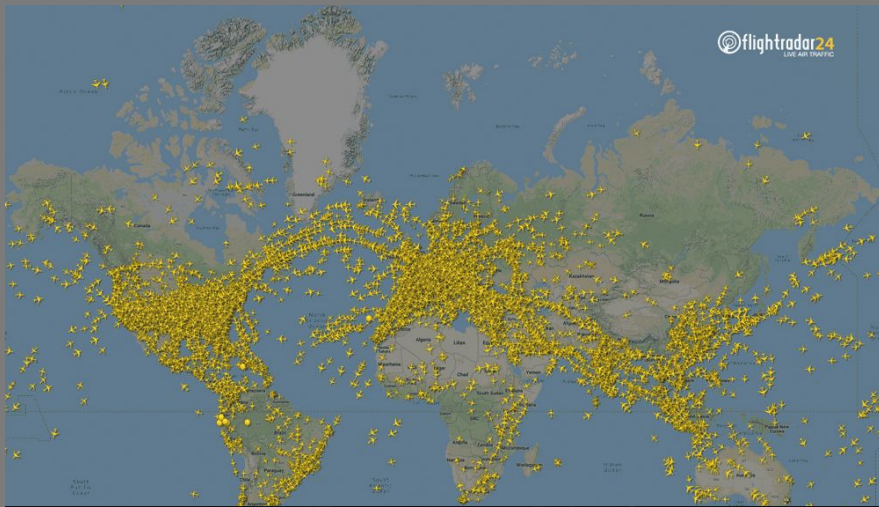
- Elle permet d'interroger les données en temps réel
- De Calculer des statistiques et faire des analyses spatiales en temps réel
- De filtrer les données sur demande
- Tout ça sans jamais avoir à interroger la base de données, tout est fait dans le client



, [8, 9]],

0.000000



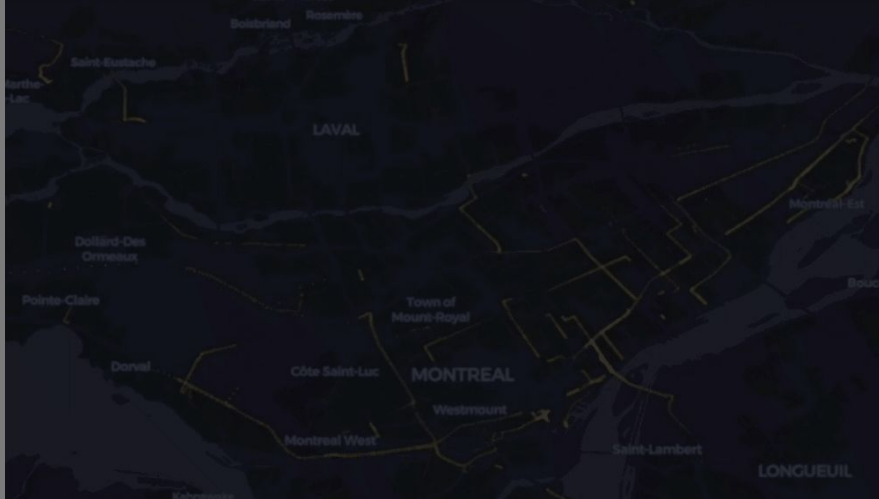


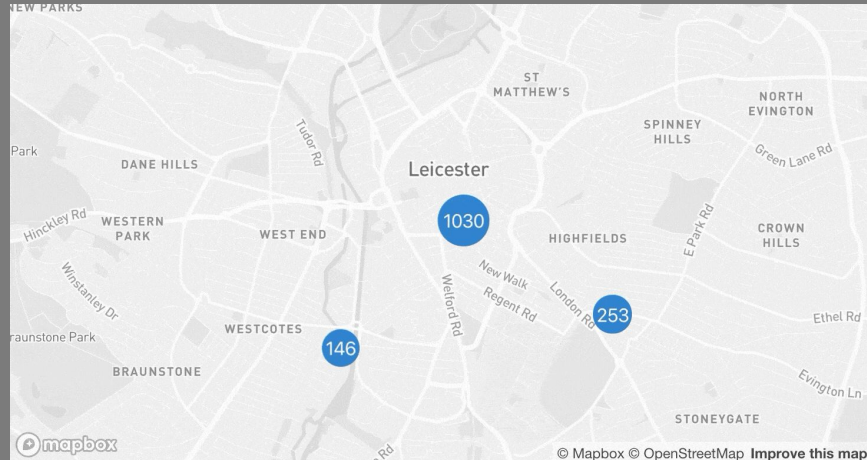
# Données vivantes

Mise à jour des données avec  
intervalles prédéfinies,

Seulement sur les tuiles demandées

Ca ne coute pas cher en terme de  
réseau, de computing et d'affichage





# Style dynamique

On peut faire évoluer le style en fonction du niveau de zoom

On peut changer le style sur demande

On peut appliquer un style en fonction des données qui se trouvent dans la tuile



# Temps de processing

## Tile creation process: Esri basemaps

- Entire world
  - ~ 8hrs on a desktop machine
  - Tiles ~ 13 GB
  - Multiple styles can use the same tiles
- Compared to raster for the entire world
  - ~ many weeks on a server cluster per map style
  - Tiles ~ 20 TB

**Calculer tous les états (zoom+tuilage) pour une pyramide matricielle est beaucoup plus demandant que de calculer une pyramide de vecteurs**





# Les tuiles vectorielles

## Désavantages des tuiles vectorielles

- Le **rendu** se produit **côté client**, ce qui peut créer des **problèmes de performances** sur les appareils lents
- **! Les vecteurs sont généralisés (c'est-à-dire qu'il ne s'agit pas de données brutes), ils peuvent donc ne pas convenir à l'édition !**
- La norme OGC n'est pas encore complétée
- Encore certains produits ne supportent pas encore les VTs



# Les tuiles vectorielles

## Utilisations courantes des tuiles vectorielles

- Cartographie en ligne pour les applications web et mobiles
- Affichage de données géographiques en temps réel
- Outils d'analyse spatiale pour les entreprises et les gouvernements

# Pause



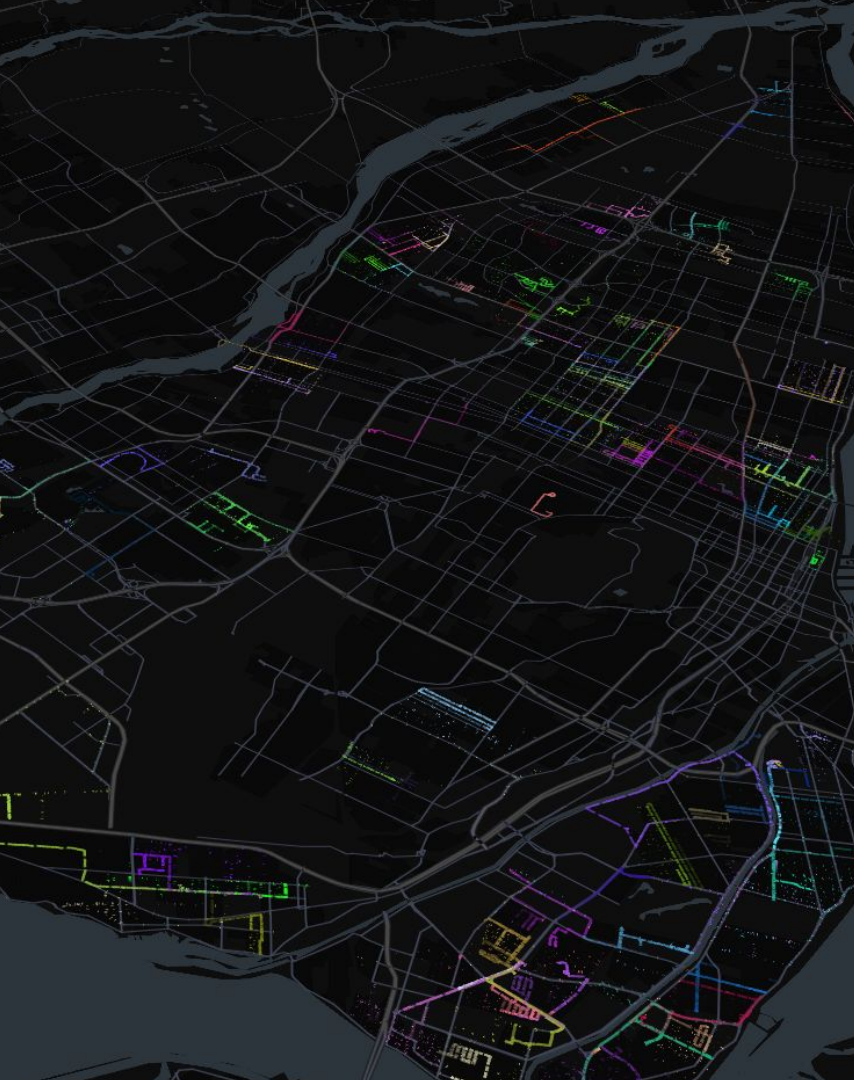


# Les tuiles vectorielles

## Les outils

- **Parsers & Generators**
- **Clients**
- **Applications / Command line tools**
- **CLI Utilities**
- **Servers**
- **Low-level utilities**

[awesome implementations of the  
Mapbox Vector Tile specification](#)



# Les tuiles vectorielles

## Les outils

- **Parsers & Generators**
- Clients
- Applications / Command line tools
- CLI Utilities
- Servers
- Low-level utilities

[awesome implementations of the  
Mapbox Vector Tile specification](#)





# Les tuiles vectorielles

## Les outils

- Parsers & Generators
- **Clients**
- Applications / Command line tools
- CLI Utilities
- Servers
- Low-level utilities

[awesome implementations of the  
Mapbox Vector Tile specification](#)

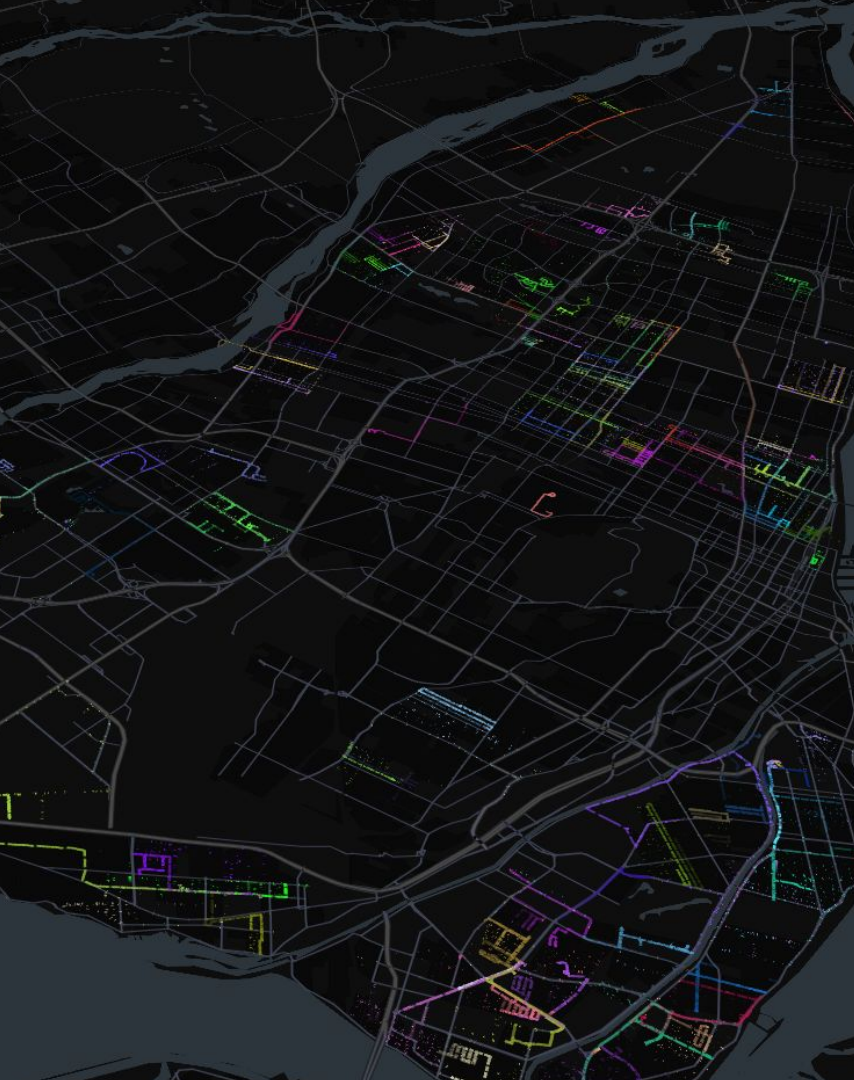


# Les tuiles vectorielles

## Les outils

- Parsers & Generators
- Clients
- **Applications / Command line tools**
- CLI Utilities
- Servers
- Low-level utilities

[awesome implementations of the  
Mapbox Vector Tile specification](#)



# Les tuiles vectorielles

## Les outils

- Parsers & Generators
- Clients
- Applications / Command line tools
- **CLI Utilities**
- Servers
- Low-level utilities

[awesome implementations of the  
Mapbox Vector Tile specification](#)





# Les tuiles vectorielles

## Les outils

- Parsers & Generators
- Clients
- Applications / Command line tools
- CLI Utilities
- **Servers**
- Low-level utilities

[awesome implementations of the  
Mapbox Vector Tile specification](#)



# Les tuiles vectorielles

## Les outils

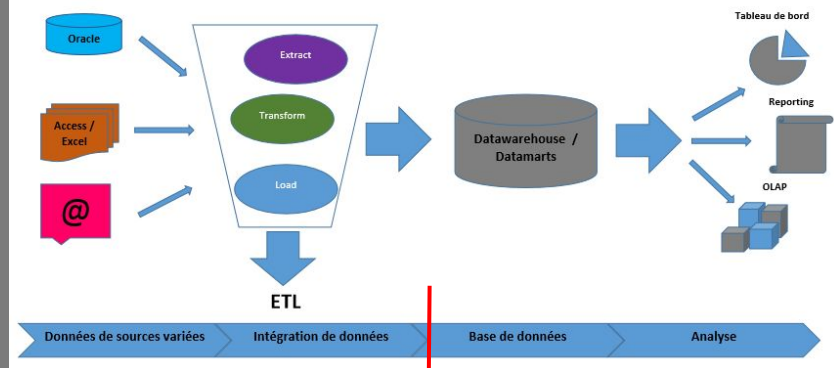
- Parsers & Generators
- Clients
- Applications / Command line tools
- CLI Utilities
- Servers
- **Low-level utilities**

[awesome implementations of the  
Mapbox Vector Tile specification](#)



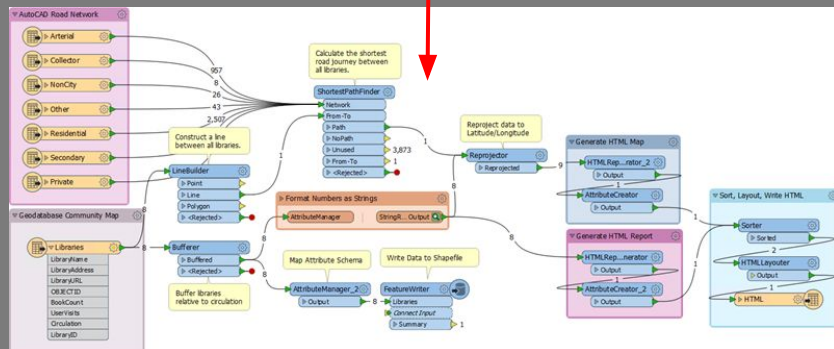
# Pause

## Schéma d'intégration de données



# Concept d'intégration de données vectorielles

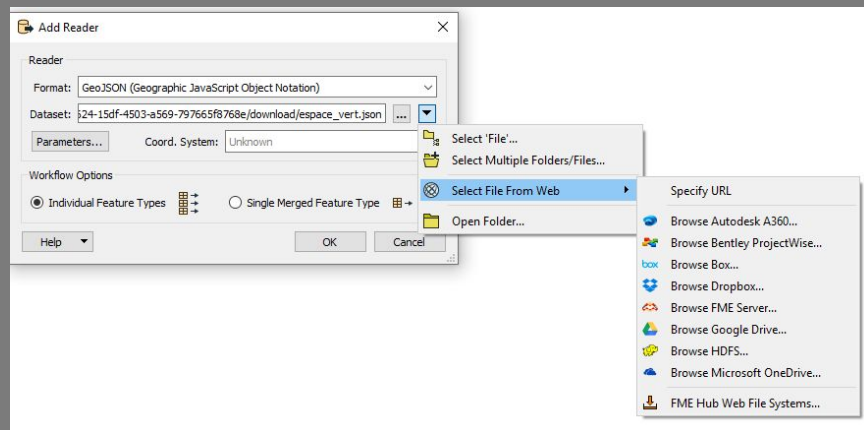
1. La **collecte** et la récupération des données.
2. La **conversion** des données dans un format compatible avec le système de gestion de données géographiques (SIG)
3. Le **traitement**, la **vérification** et la **validation**
4. La mise en place de la **persistance** des données
5. La **diffusion** des données pour créer des cartes, des analyses spatiales, des visualisations et des modèles.





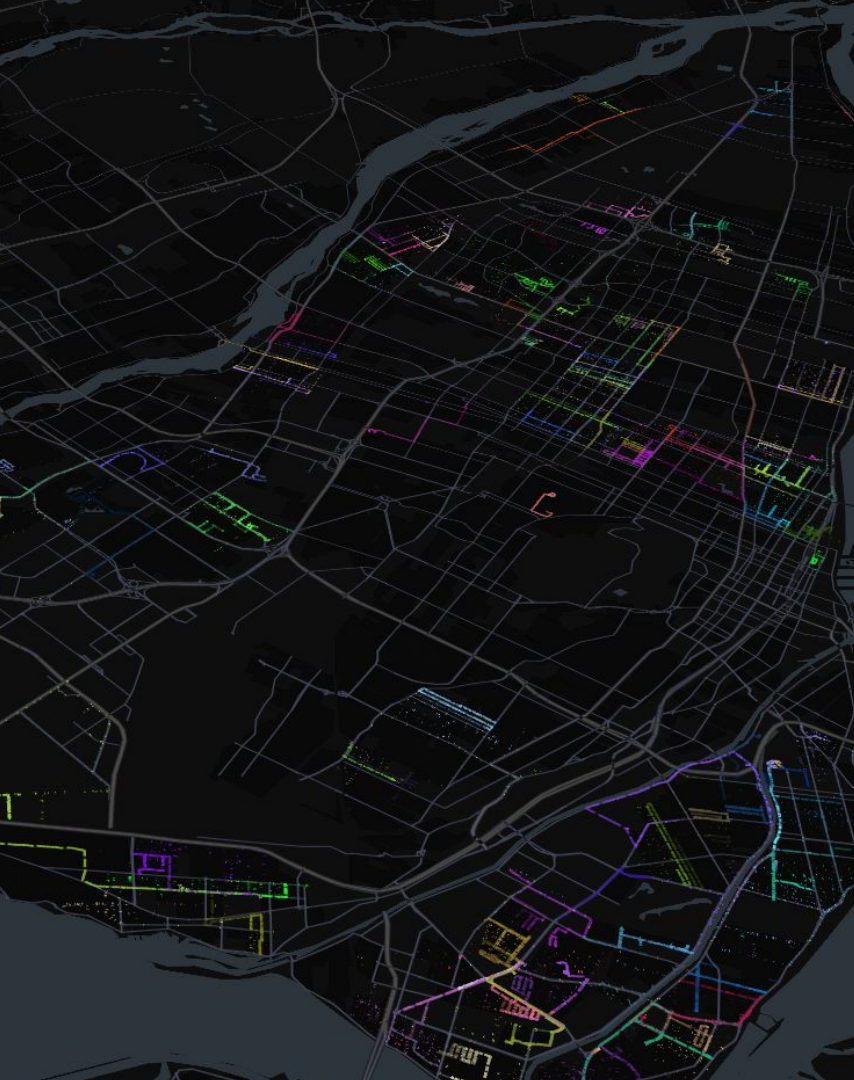
# Concept d'intégration de données vectorielles

La collecte et la récupération des données



- Le paradigme a changé avant nous téléchargeons périodiquement les données dans des fichiers, maintenant on **se connecte directement à la source**
- Fini les fichiers et les versionnements, les téléchargements interminables et le stockage volumineux

[Switch from Shapefile](#)



# Concept d'intégration de données vectorielles

La conversion des données dans un format compatible avec le système de gestion de données géographiques (SIG)

- On a maintenant les moyens de transformer presque tous les formats dans les 2 sens
- On doit uniformiser et normaliser les systèmes pour qu'ils puissent être tous interopérables
- OGC aide à normaliser les pratiques

[Open Geospatial Consortium](https://www.opengeospatial.org/)





# Concept d'intégration de données vectorielles

La vérification et la validation des données

- L'étape la plus importante
- Crédibilité
- Véracité
- Performance
- Précision
- Fiabilité





# Concept d'intégration de données vectorielles

La mise en place de la persistance des données

- Centralisées à une seule place
- Capacité de gestion des permissions et des versions
- Disponibilité
- Mise à l'échelle
- Performant (grâce aux **index**)
- Répliqué
- Fiabilité

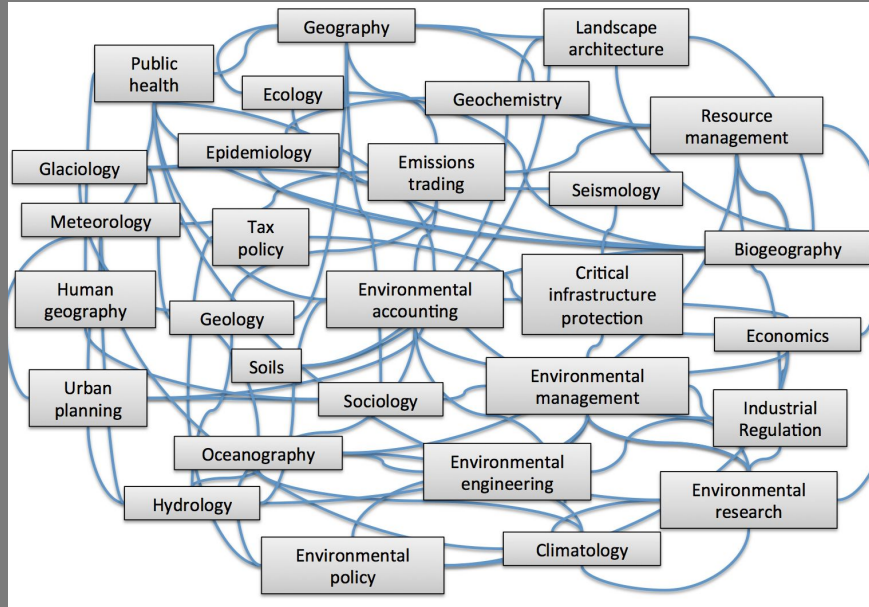


# Concept d'intégration de données vectorielles

L'utilisation par la diffusion des données pour créer des cartes, des analyses spatiales, des visualisations et des modèles

- Diffusion en direct au plus proche de la donnée
- Transparence
- Ville intelligente
- Open source
- Crowdsource
- Découvrabilité accrue

# Défis et enjeux de l'intégration



1. Hétérogénéité des données
2. Incohérence des données
3. Problèmes de méthodologie
4. Problèmes de performance
5. Problèmes de maintenance
6. Problème de sécurité
7. Problème de coût

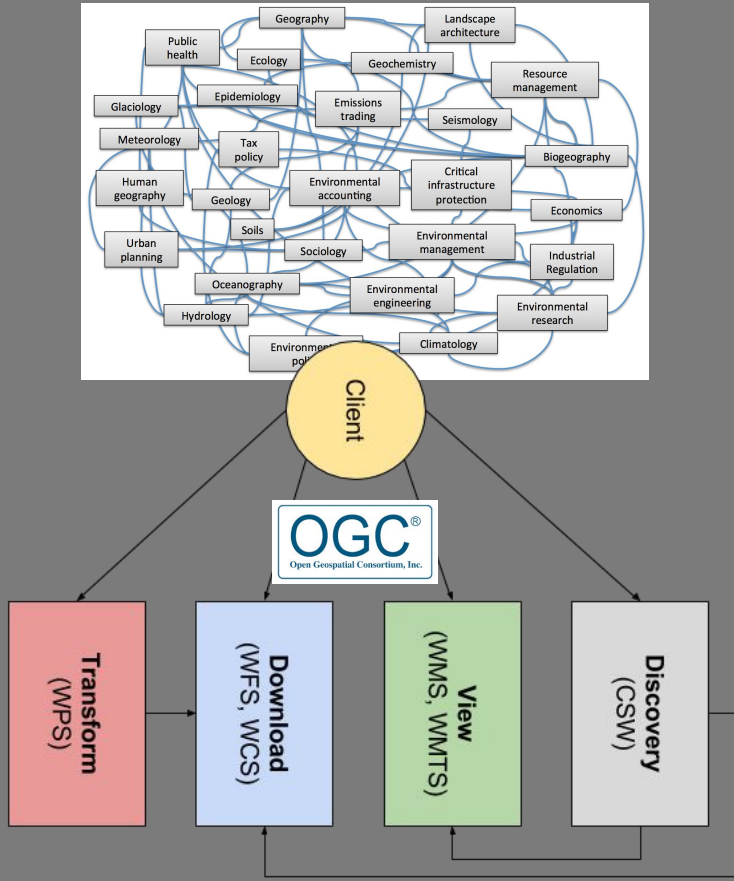
# Défis et enjeux de l'intégration



1. Hétérogénéité des données
2. Incohérence des données
3. Problèmes de méthodologie
4. Problèmes de performance
5. Problèmes de maintenance
6. Problème de sécurité
7. Problème de coût



# Défis et enjeux de l'intégration



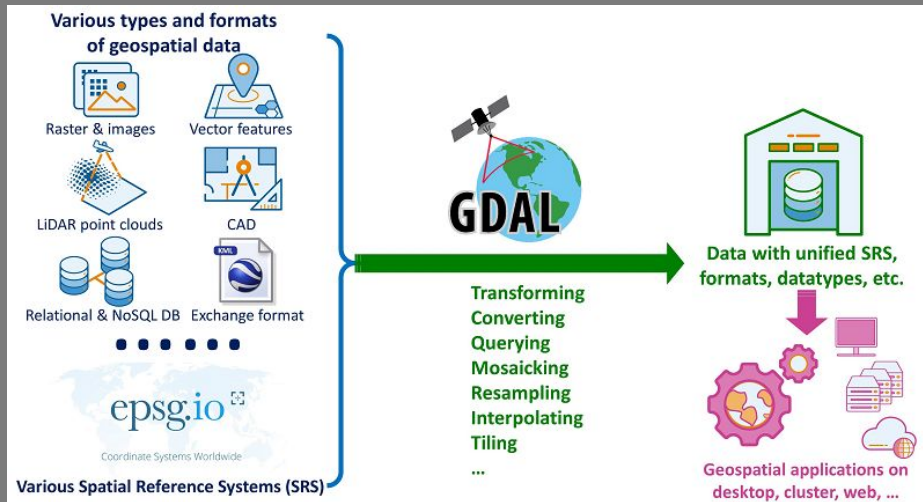
## Hétérogénéité des données

Les données géographiques peuvent provenir de différentes sources, utiliser différents formats et avoir des niveaux de qualité variables. Il peut être difficile de les intégrer de manière cohérente dans un SIG.

C'est pourquoi il faut normaliser les pratiques, centraliser l'intégration et utiliser des standards de type OGC



# Défis et enjeux de l'intégration



## Hétérogénéité des données

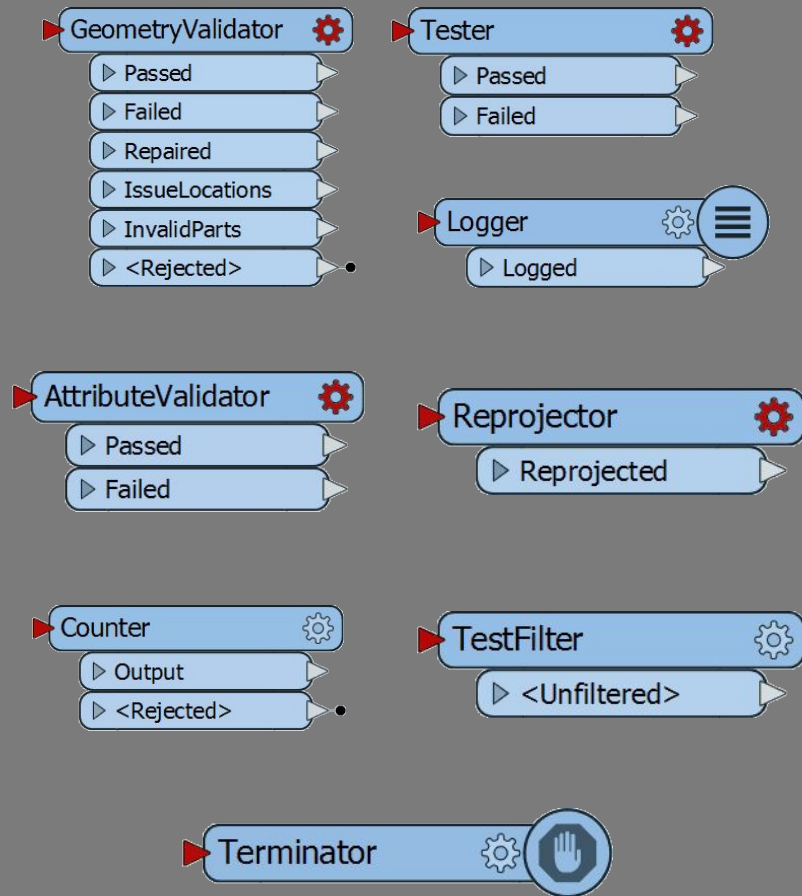
Des outils comme FME, ou les bibliothèques GDAL/OGR, GRASS ... permettent de nous assurer de rendre les données homogènes intrinsèquement mais aussi pour les systèmes, afin de les rendre cohérentes.

[GDAL/OGR - OSGeo](#)

[GDAL](#)

[Software using GDAL](#)

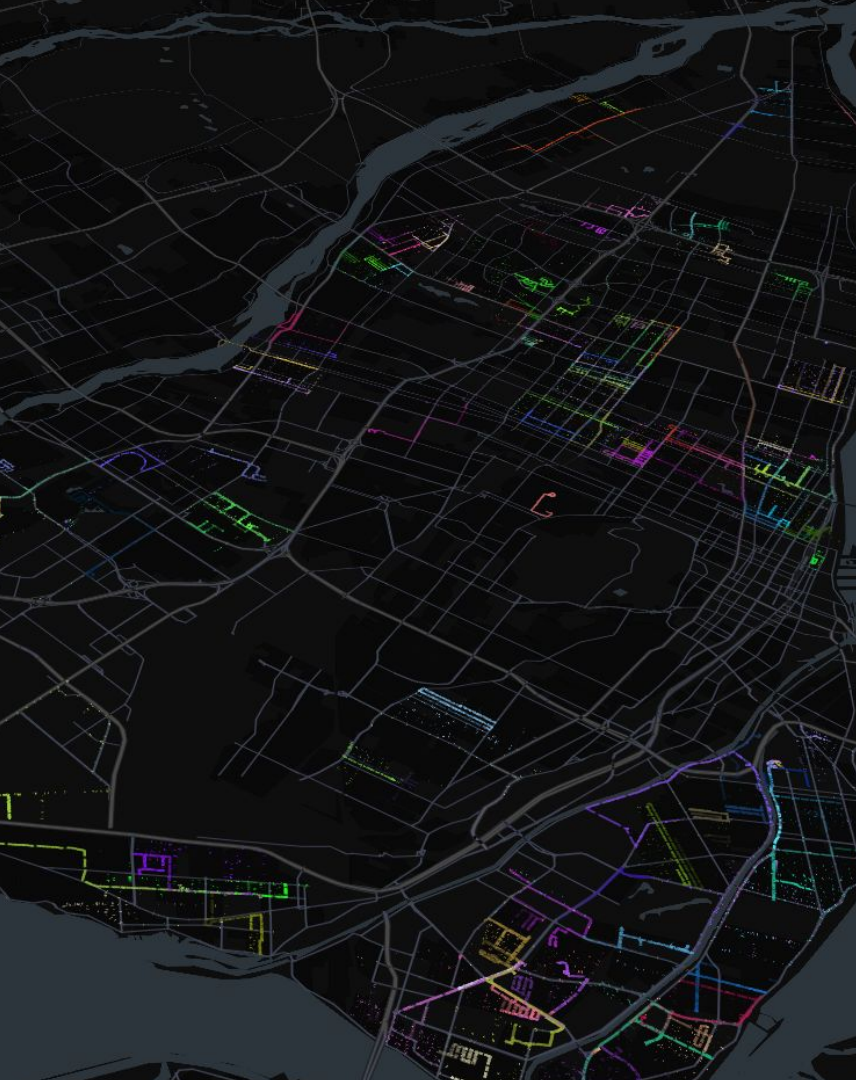
# Défis et enjeux de l'intégration



## Incohérence des données

Les données géographiques peuvent contenir des incohérences, comme des erreurs, des redondances ou des incohérences dans les attributs. Il peut être difficile de détecter et de corriger ces incohérences avant de les intégrer dans un SIG.

C'est pour ça qu'il faut toujours faire des **itérations de reconnaissances**, analyses des jeux de données, en sortant des **statistiques**, ou en créant des boucle de validation d'assurance qualité

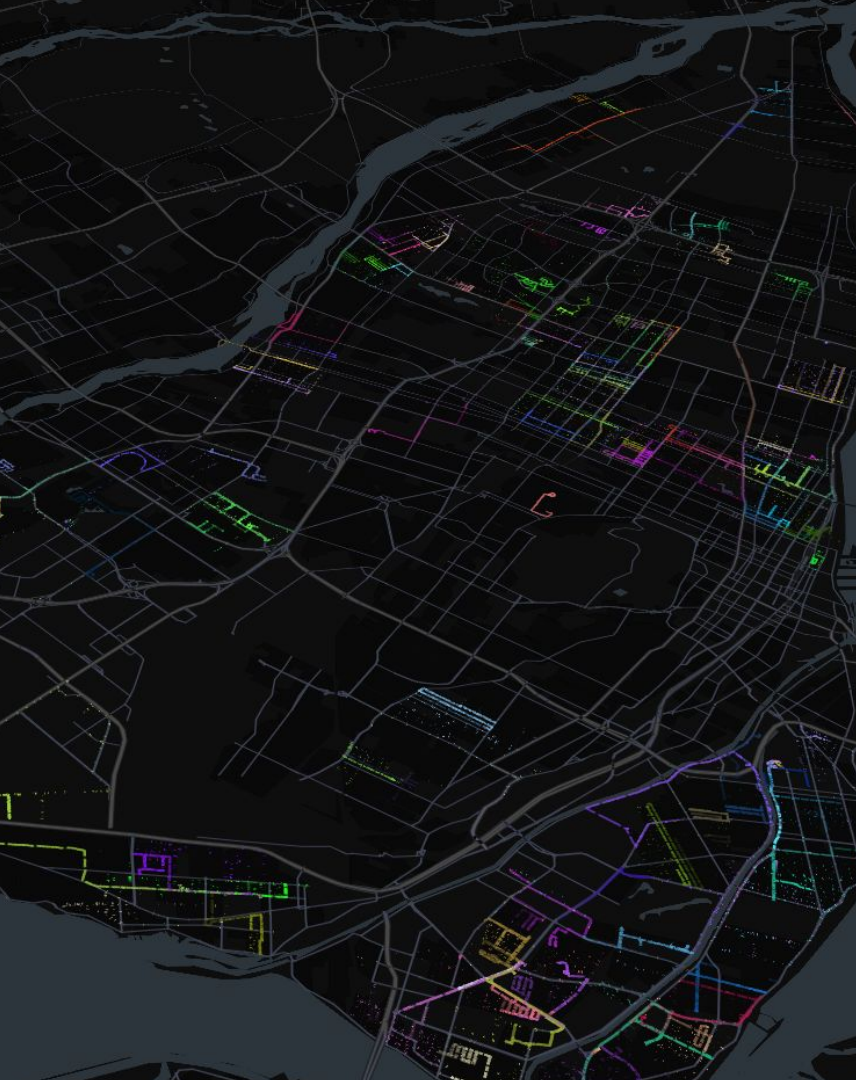


# Défis et enjeux de l'intégration

## Problèmes de méthodologie

Il peut être difficile de définir une méthodologie d'intégration de données qui convient à tous les types de données géographiques et qui est adaptée aux besoins de l'application.

Néanmoins le paradigme méthodologique a changé et on peut désormais faire confiance à des outils d'intégration qui nous permettent d'**uniformiser** la méthode d'intégration, ou du moins de la **centraliser** et de la **documenter**.



# Défis et enjeux de l'intégration

## Problèmes de performance

L'intégration de grandes quantités de données géographiques peut entraîner des problèmes de performance dans un SIG, notamment des temps de chargement et de traitement plus longs.

Encore une fois, le paradigme a changé et nos méthodes de centralisation, de stockage, d'indexation et de diffusion sont devenus très performantes (pratique DevOps)



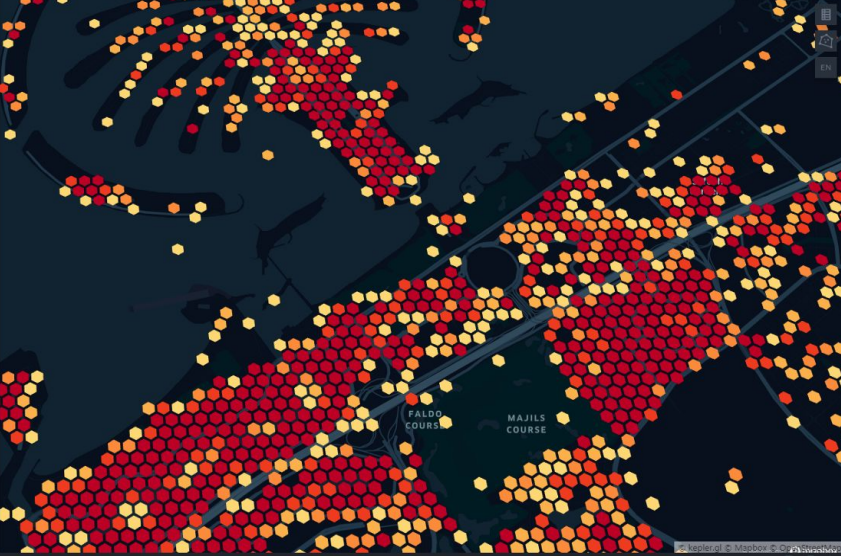


# Défis et enjeux de l'intégration

## Problèmes de maintenance

Une fois les données intégrées dans un SIG, il peut être difficile de les maintenir à jour et de les synchroniser avec les données d'origine

C'est pourquoi il faut centraliser l'information et mettre des "gatekeepers" (gardiens) dans les systèmes. **Logger** le plus possible les endroits problématiques et mettre des **déclencheurs d'alerte**



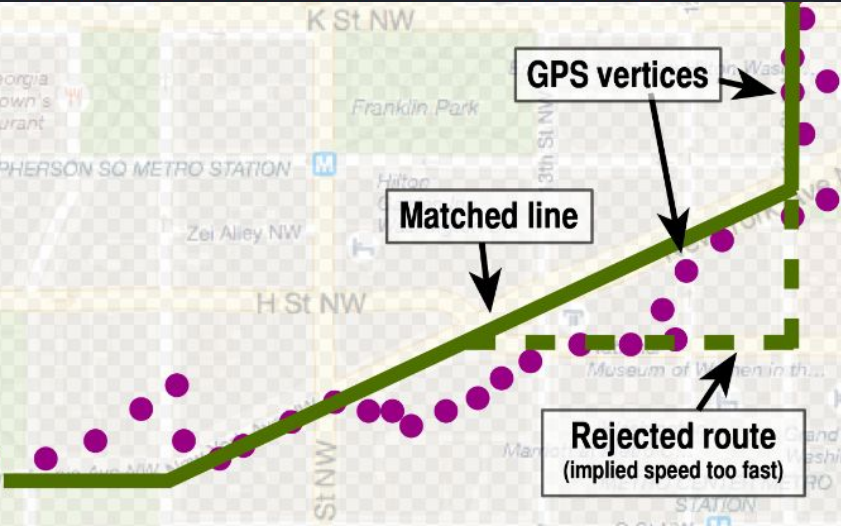
# Défis et enjeux de l'intégration

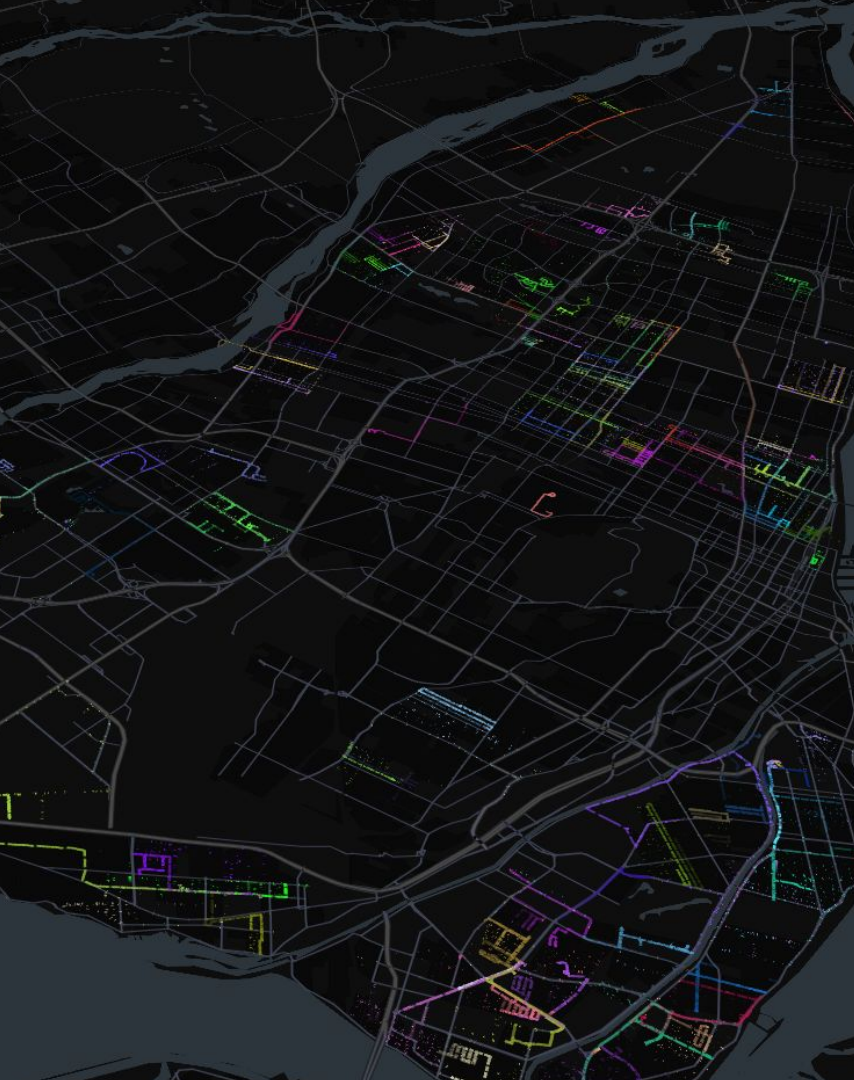
## Problème de sécurité

Les données géographiques peuvent contenir des informations sensibles qui nécessitent une protection spécifique pour éviter les fuites ou les abus.

## Anonymiser les données :

- Simplifier les géométrie
- Fusionner/ agréger des données dans des entités administratives
- Indexer les données
- Map matching





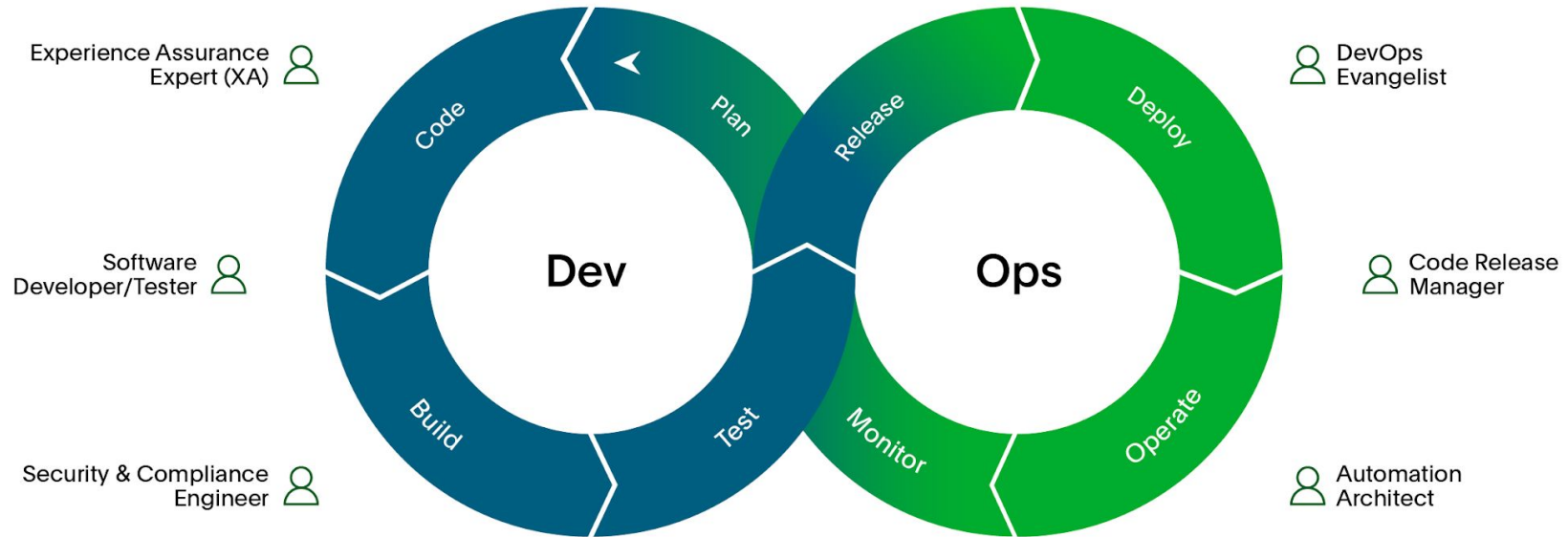
# Défis et enjeux de l'intégration

Problème de coût

L'intégration de données géographiques peut être coûteuse en termes de **personnel**, de matériel et de logiciels.

Pratique DevOps GIS

# 6 essential DevOps roles







# Les tendances et les développements futurs

- DevOps
- Savoir réfléchir et mettre en place l'**architecture complète** d'intégration et de visualisation
- 3D et Lidar sont compatibles avec les VTs
- Ne plus utiliser de fichiers, avec un processus d'intégration automatisé
- Automatisation du déploiement
- Automatisation de réplication ETL
- **Documentation et tests = moins d'efforts et moins de maintenance**