



GEO 7630

**Intégration et visualisation de données
géographiques**

Semaine 10 -

**Principes, Architectures et Services Géospatiaux
pour le Développement d'Applications de
Webmapping**



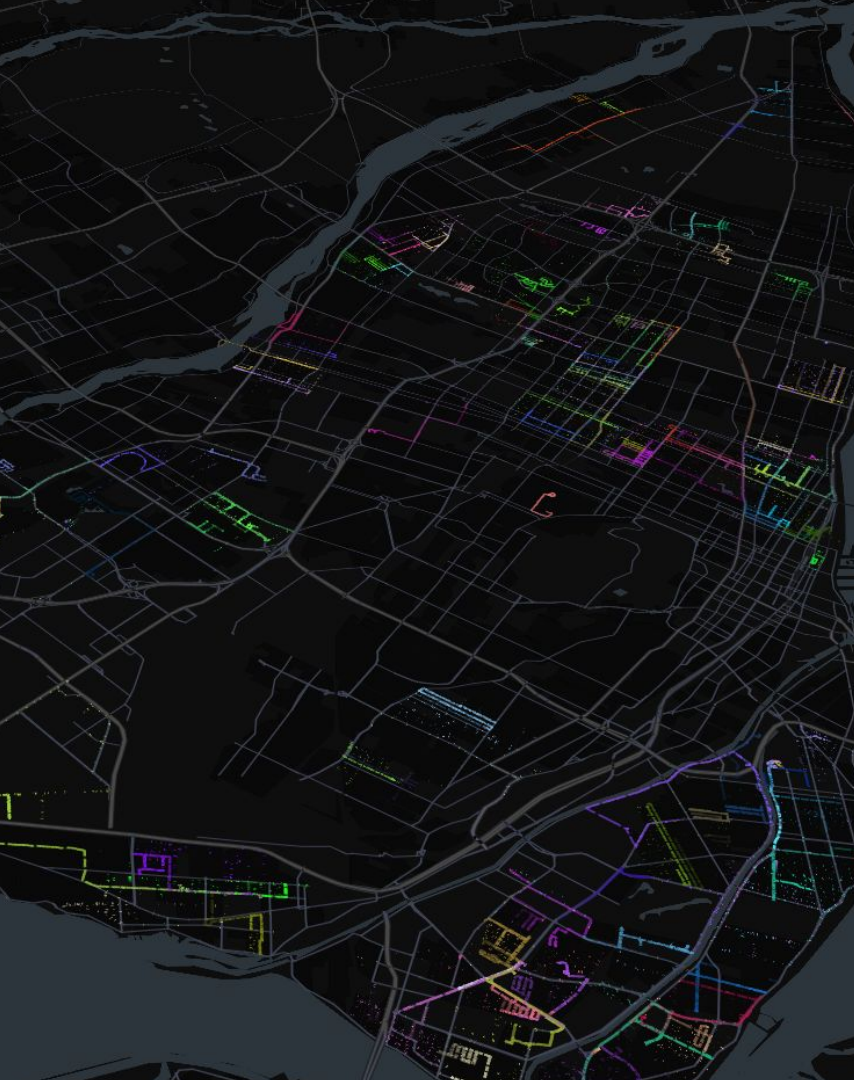
Objectifs du cours

Principes, Architectures et Services Géospatiaux pour le Développement d'Applications de Webmapping

- Développement d'application sur le web
- Principes et bases du webmapping
- Architecture d'une application web open source
- Les normes Open Geospatial Consortium (OGC)
- Notions de services web géospatiaux, WMTS, WFS, VTS ...
- Diffusion des données géospatiales avec GeoServer
- Notions de services web géospatiaux

Laboratoire:

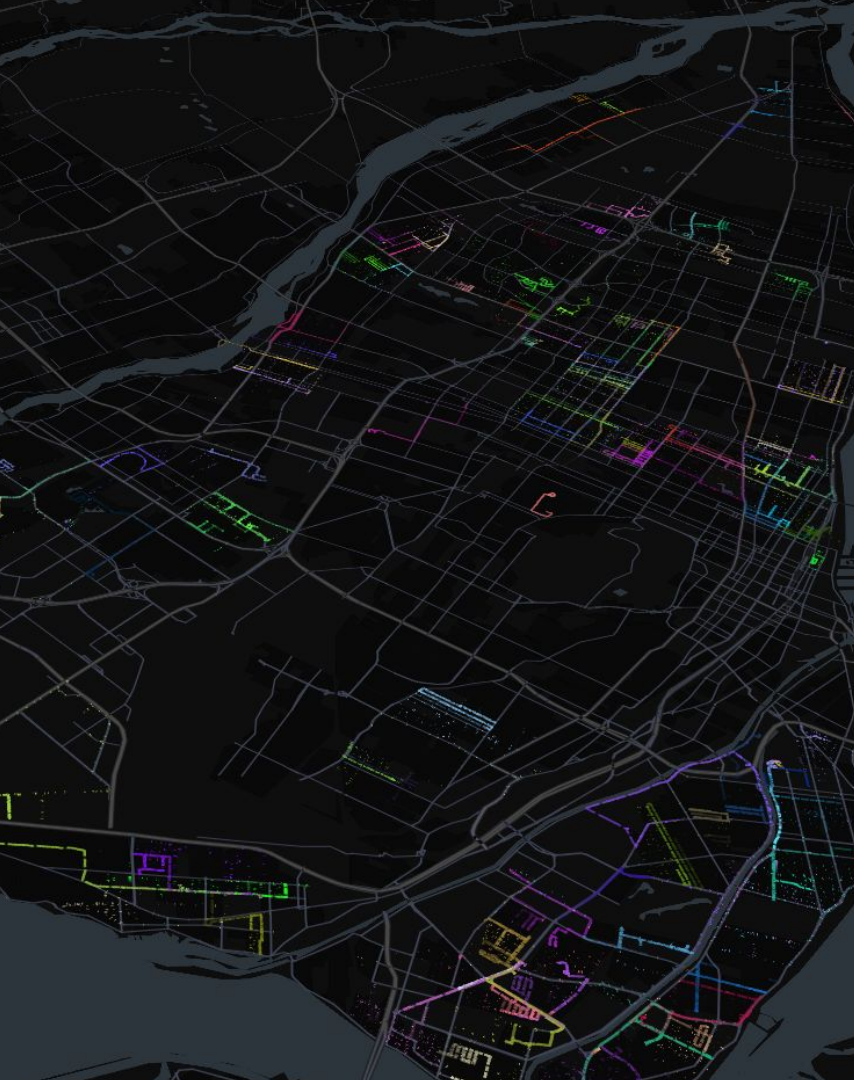
- Diffusion et visualisation de données avec Geoserver et Maplibre



Architecture d'une application web open source

Architecture d'une application web open source:

- L'architecture client-serveur
- La modélisation de données pour le webmapping
- Les bibliothèques de cartographie côté client
- La gestion des requêtes côté serveur
- Les outils de développement d'application web open source (Eclipse, NetBeans)



L'architecture client-serveur

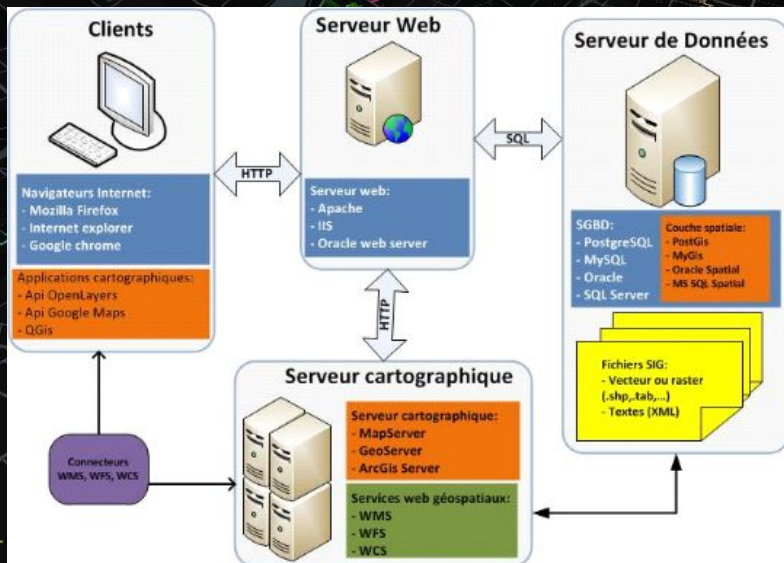
Introduction à l'architecture client-serveur

- Les composants d'une architecture client-serveur
- Les avantages et les inconvénients de l'architecture client-serveur
- Les protocoles de communication client-serveur (HTTP, TCP/IP, WebSocket)
- Les exemples d'application de l'architecture client-serveur

L'architecture client-serveur

Les composants d'une architecture client-serveur

- Le client
- Le serveur
- Le protocole
- La couche de données
- La couche de traitement
- La couche de sécurité



L'architecture client-serveur



Le client :

Interface utilisateur qui permet d'accéder aux services offerts par le serveur.

Le client peut être une application installée sur une machine, un navigateur web ou une application mobile.

L'architecture client-serveur

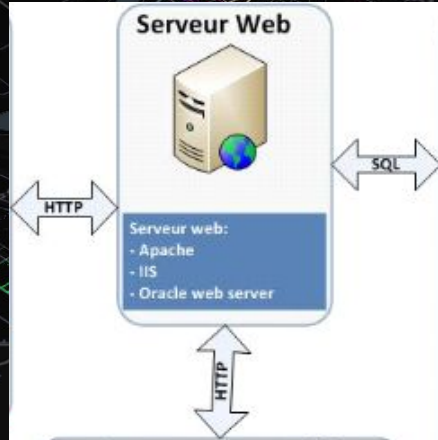


Le serveur :

Entité qui fournit des services aux clients.

Le serveur peut être une machine physique ou virtuelle, sur laquelle des logiciels et des services sont exécutés.

L'architecture client-serveur

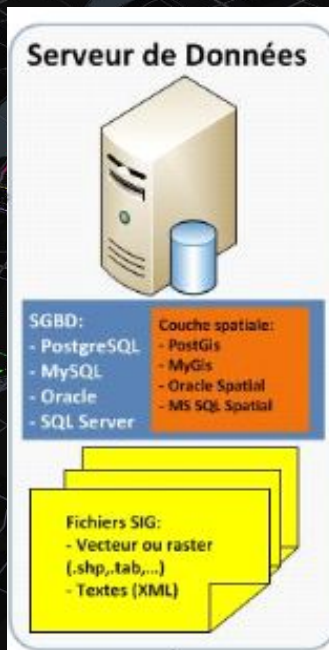


Le protocole :

Ensemble de règles et de formats qui permettent à un client de communiquer avec un serveur.

Les protocoles courants pour les architectures client-serveur sont HTTP, FTP, TCP/IP, WebSocket, etc.

L'architecture client-serveur



La couche de données :

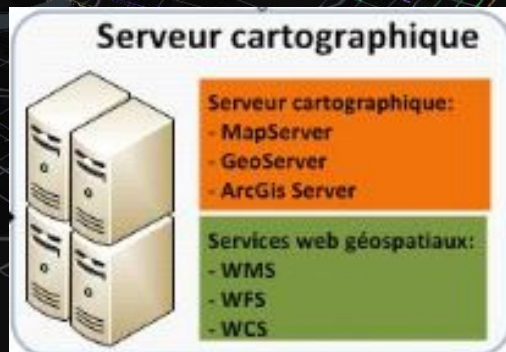
Couche qui gère l'accès et la manipulation des données côté serveur.

Cette couche peut inclure des bases de données, des services web, des API, etc.

L'architecture client-serveur

La couche de traitement :

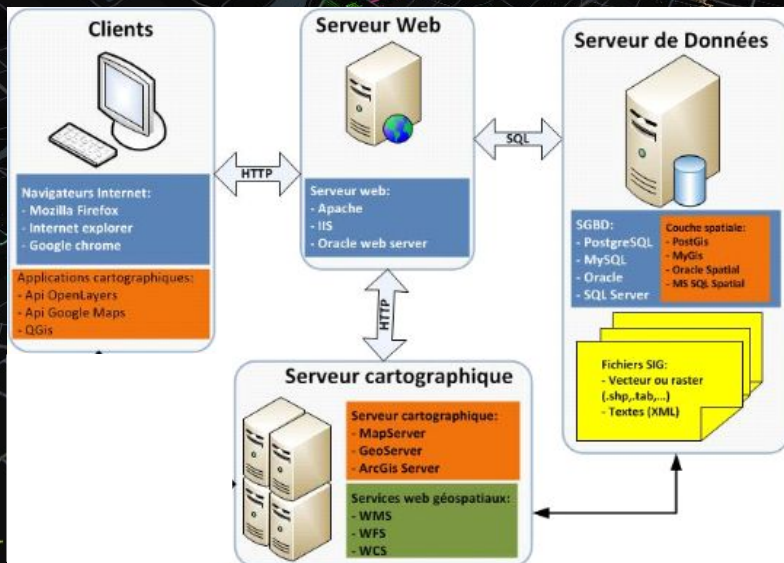
Couche qui gère les opérations de traitement côté client, par exemple la validation des données, les opérations de calcul, etc.



L'architecture client-serveur

La couche de sécurité :

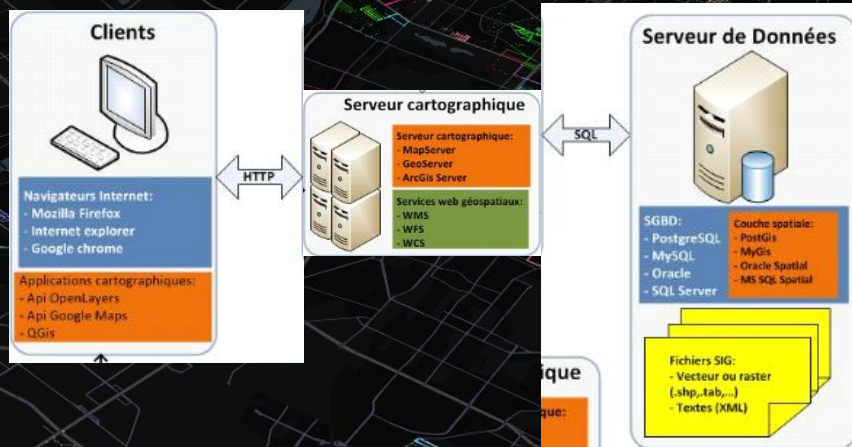
Couche qui garantit la sécurité des échanges entre le client et le serveur, en protégeant les données, les communications, les accès, etc.



L'architecture client-serveur

Les composants d'une architecture client-serveur

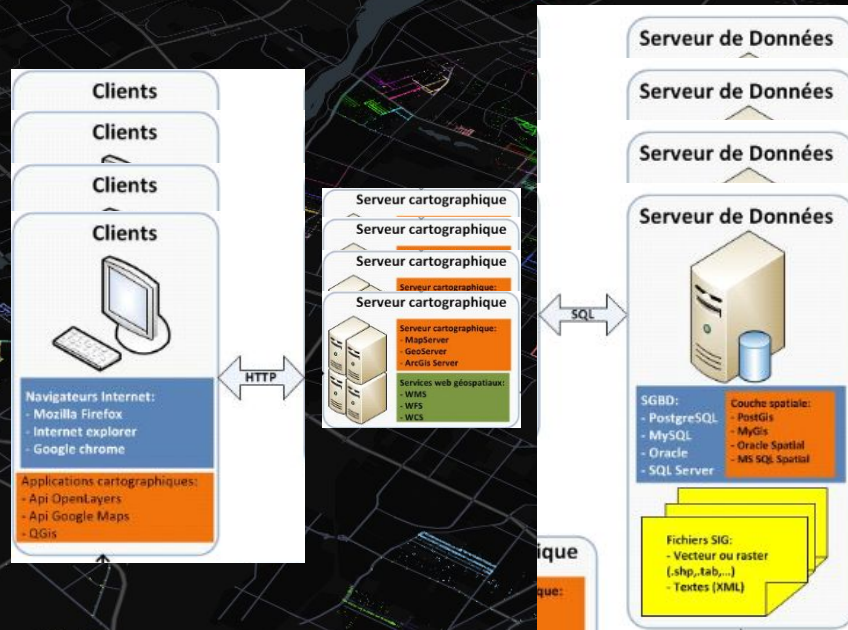
Dans le cadre du cours nous allons abstraire la couche **SERVEUR WEB** pour connecter un serveur cartographique multifonctions directement à nos bases de données



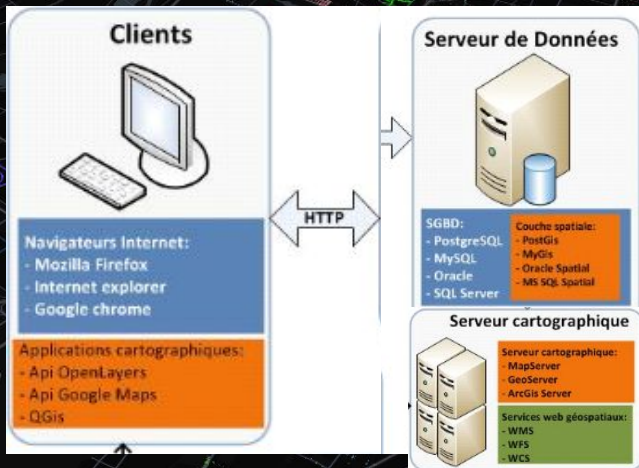
Les avantages et les inconvénients de l'architecture client-serveur

Avantages :

- La séparation
- La centralisation
- La communication
- La flexibilité et la scalabilité



Les avantages et les inconvénients de l'architecture client-serveur



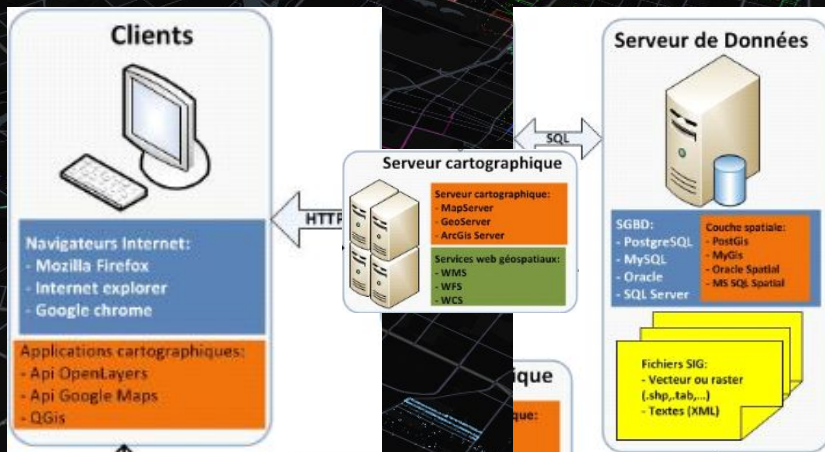
Avantages :

La séparation entre la logique métier et l'interface utilisateur facilite la maintenance, l'évolution et la réutilisation des composants.

Les avantages et les inconvénients de l'architecture client-serveur

Avantages :

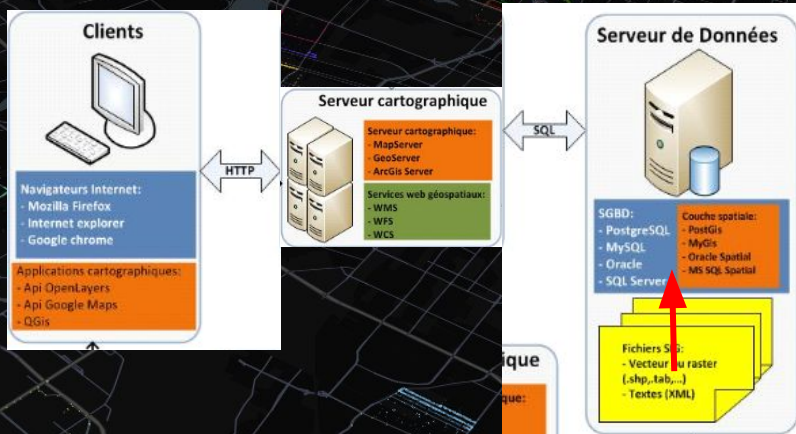
La séparation entre la logique métier et l'interface utilisateur facilite la maintenance, l'évolution et la réutilisation des composants.



Les avantages et les inconvénients de l'architecture client-serveur

Avantages :

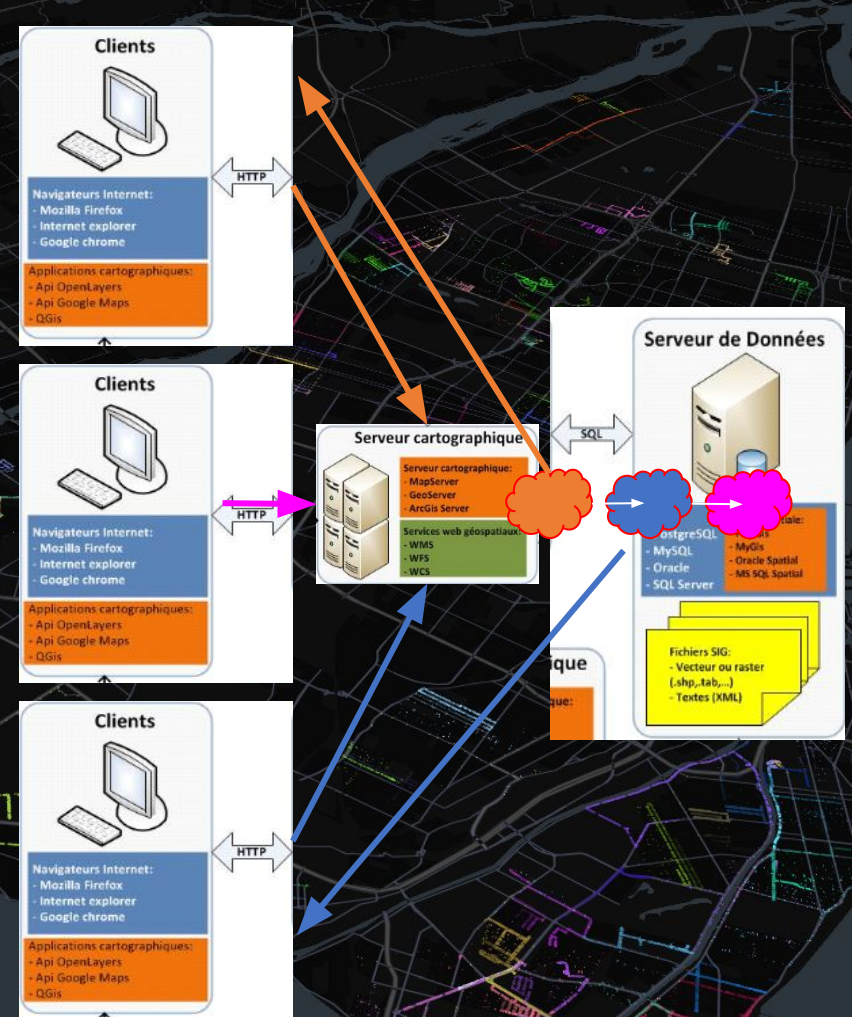
La centralisation des données et des services permet une meilleure gestion de la sécurité, de la disponibilité et de la performance.



Les avantages et les inconvénients de l'architecture client-serveur

Avantages :

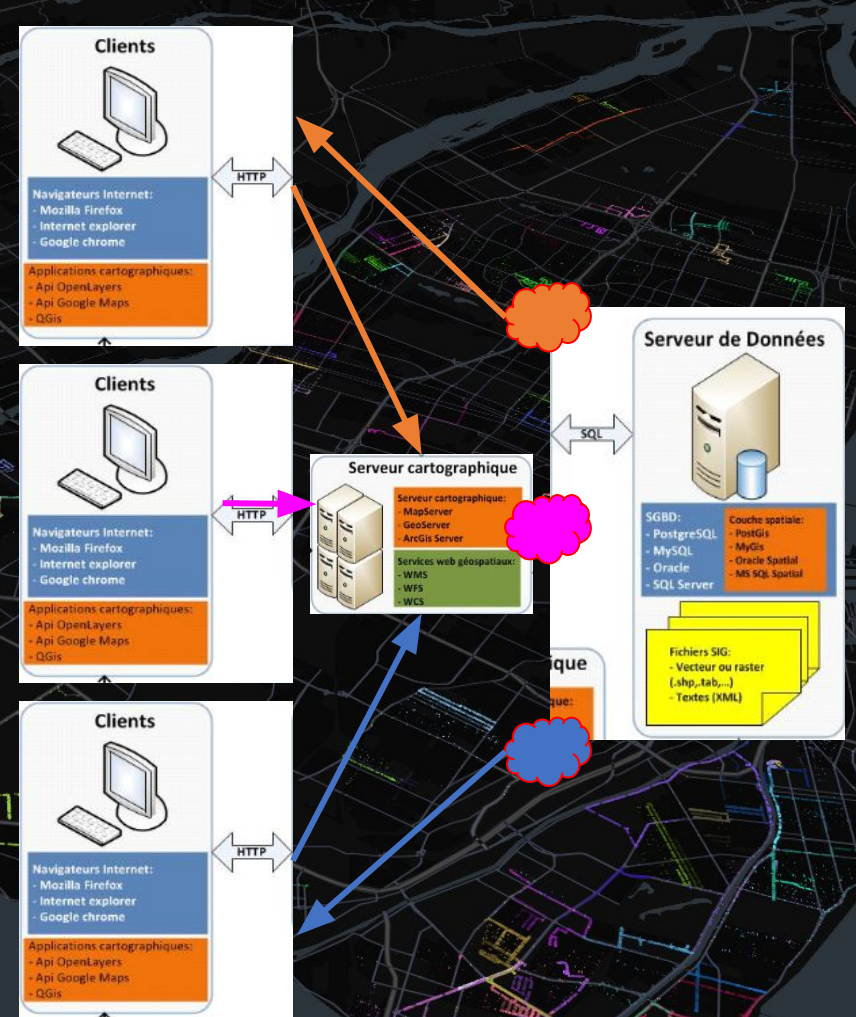
La communication entre le client et le serveur peut se faire de manière asynchrone, ce qui permet de rendre les applications plus réactives et plus performantes.



Les avantages et les inconvénients de l'architecture client-serveur

Avantages :

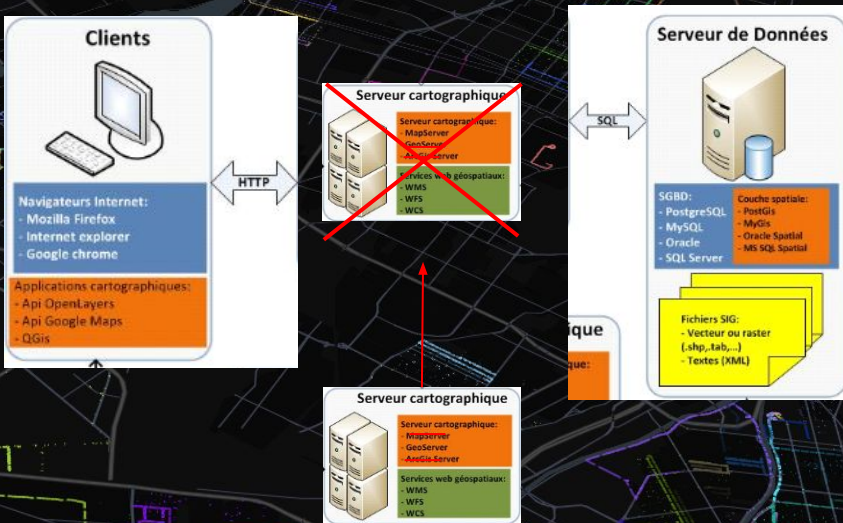
La communication entre le client et le serveur peut se faire de manière asynchrone, ce qui permet de rendre les applications plus réactives et plus performantes.



Les avantages et les inconvénients de l'architecture client-serveur

Avantages :

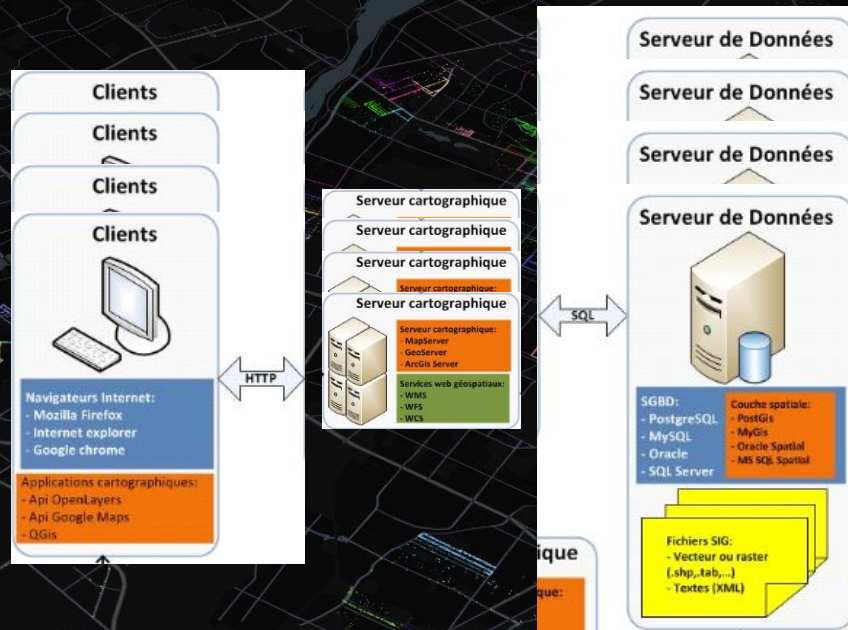
La flexibilité et la scalabilité de l'architecture client-serveur permettent de faire évoluer facilement l'application en ajoutant ou en supprimant des serveurs selon les besoins.



Les avantages et les inconvénients de l'architecture client-serveur

Avantages :

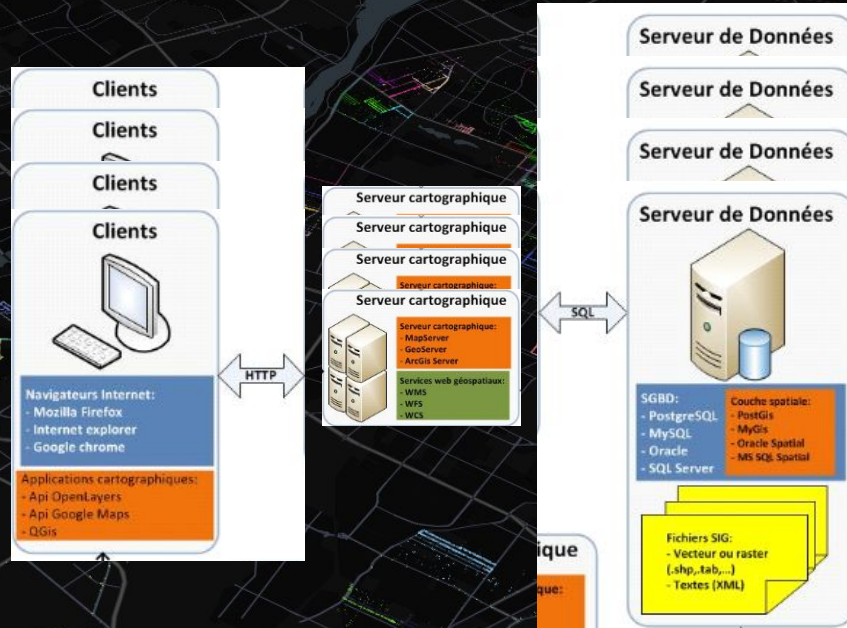
La flexibilité et la scalabilité de l'architecture client-serveur permettent de faire évoluer facilement l'application en ajoutant ou en supprimant des serveurs selon les besoins.

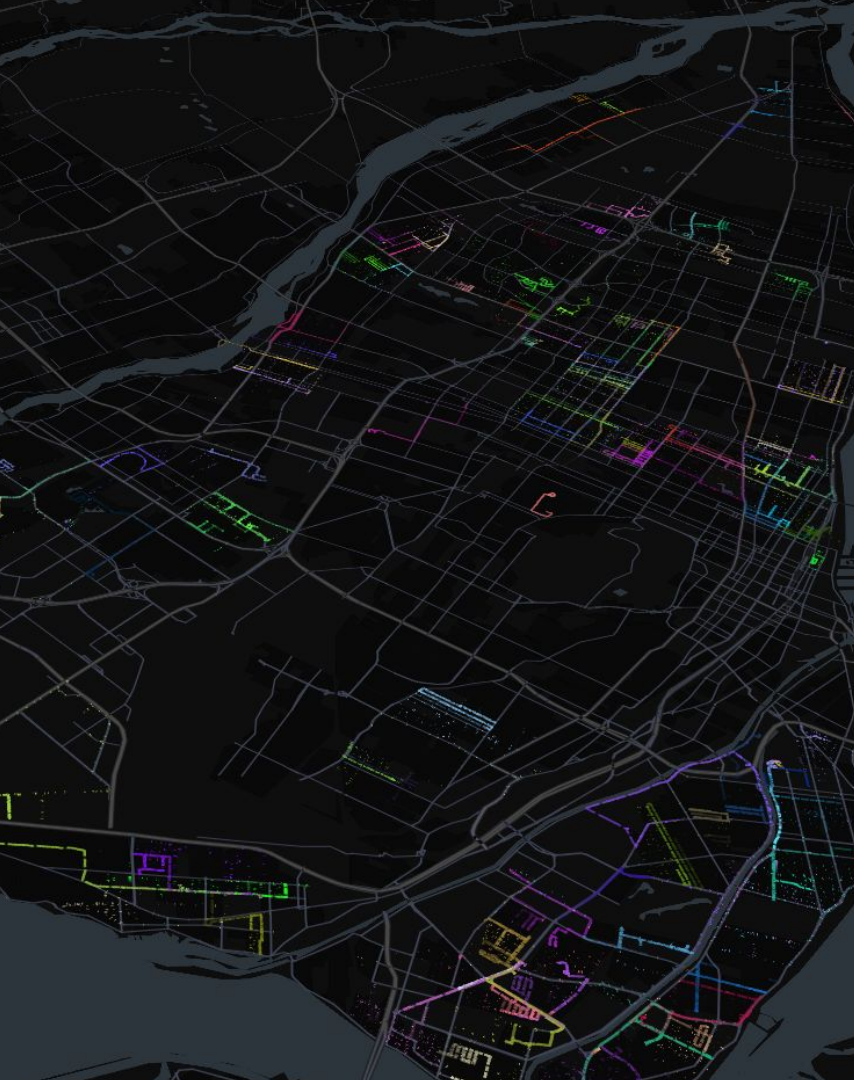


Les avantages et les inconvénients de l'architecture client-serveur

Inconvénients :

- La dépendance
- La complexité
- Les temps de latence
- Les problèmes de sécurité

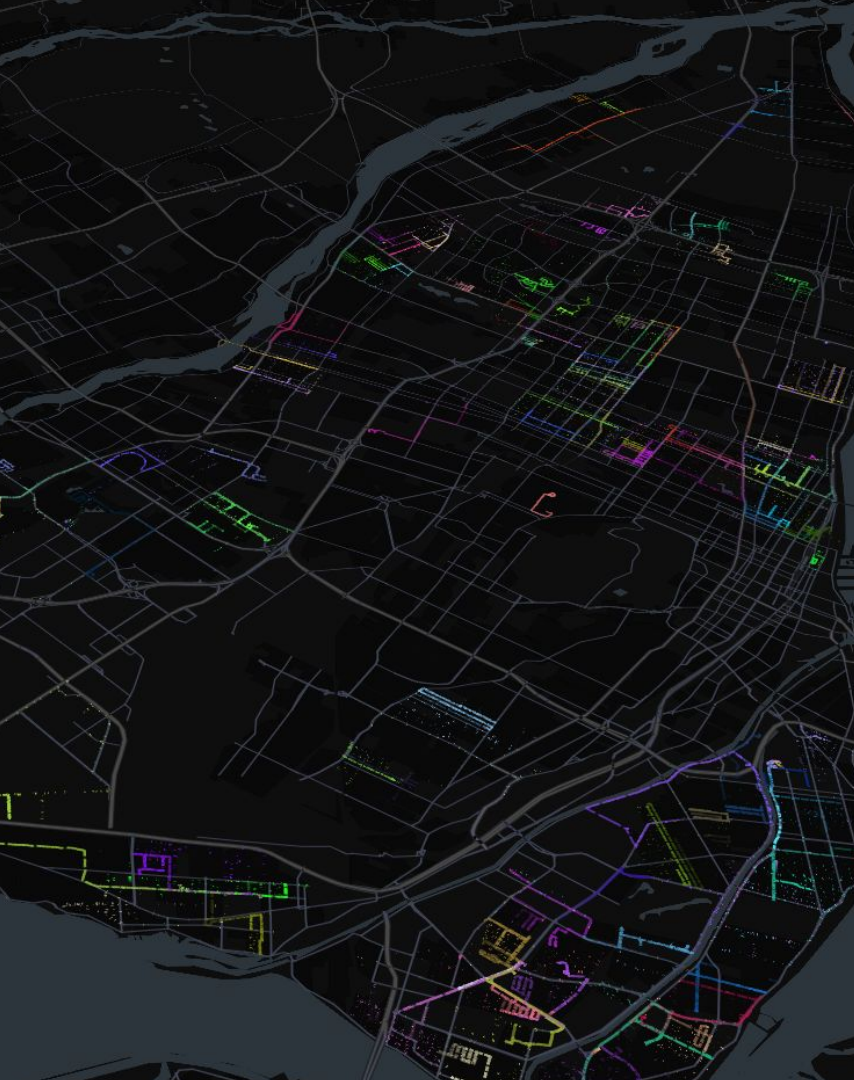




Les avantages et les inconvénients de l'architecture client-serveur

Inconvénients :

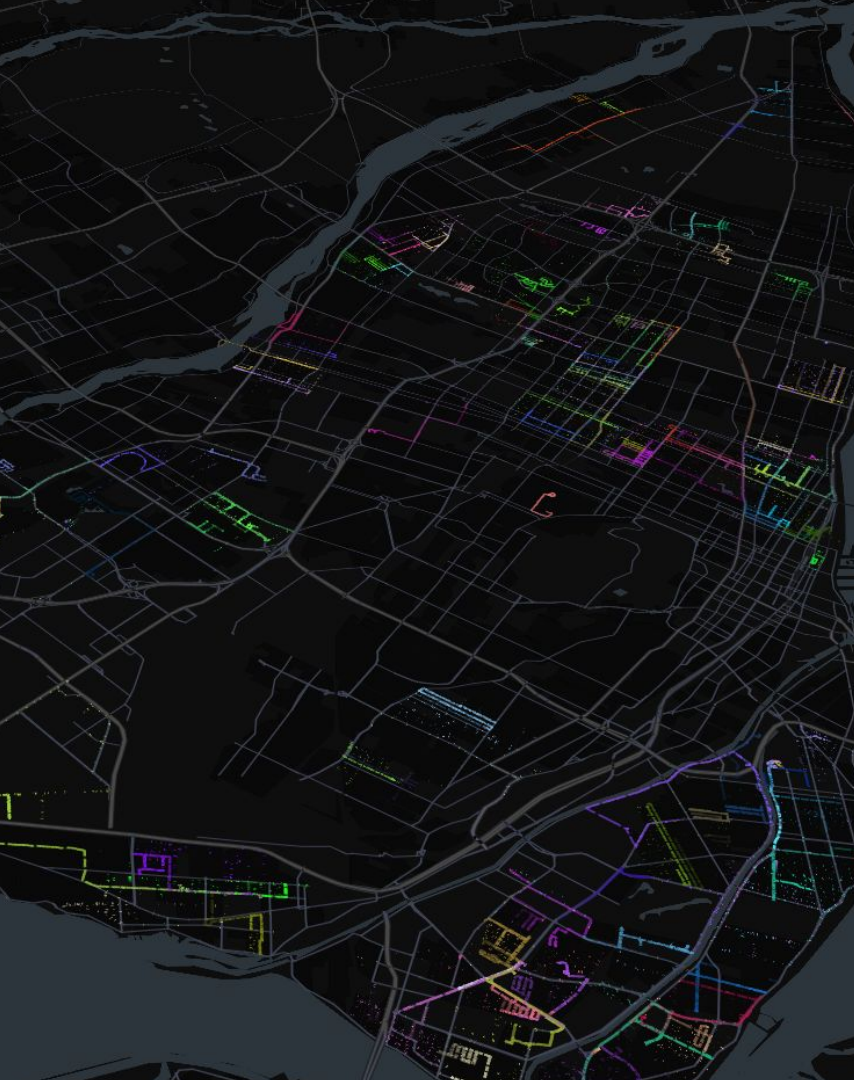
La dépendance entre le client et le serveur peut rendre l'application moins tolérante aux pannes, car la panne d'un composant peut entraîner l'indisponibilité de l'ensemble du système.



Les avantages et les inconvénients de l'architecture client-serveur

Inconvénients :

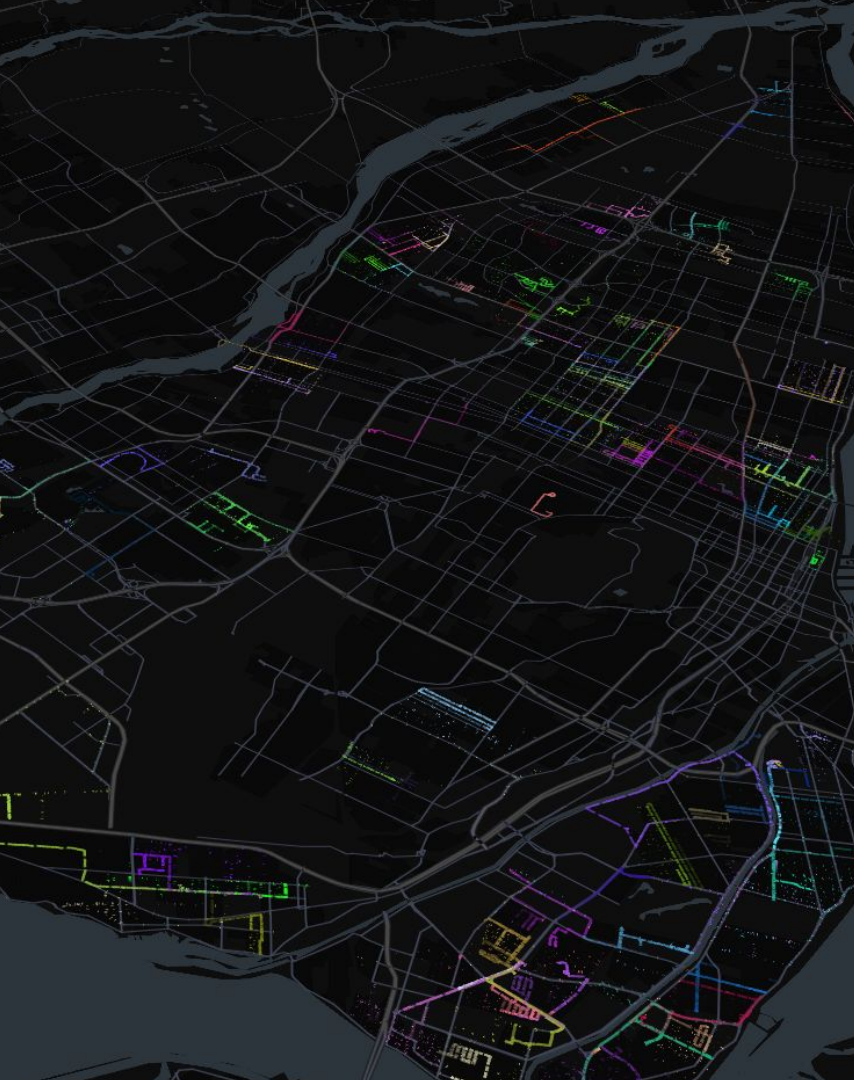
La complexité de l'architecture peut rendre la mise en place et la maintenance plus coûteuses.



Les avantages et les inconvénients de l'architecture client-serveur

Inconvénients :

Les temps de latence liés aux échanges entre le client et le serveur peuvent ralentir l'application, en particulier si le réseau est de mauvaise qualité.



Les avantages et les inconvénients de l'architecture client-serveur

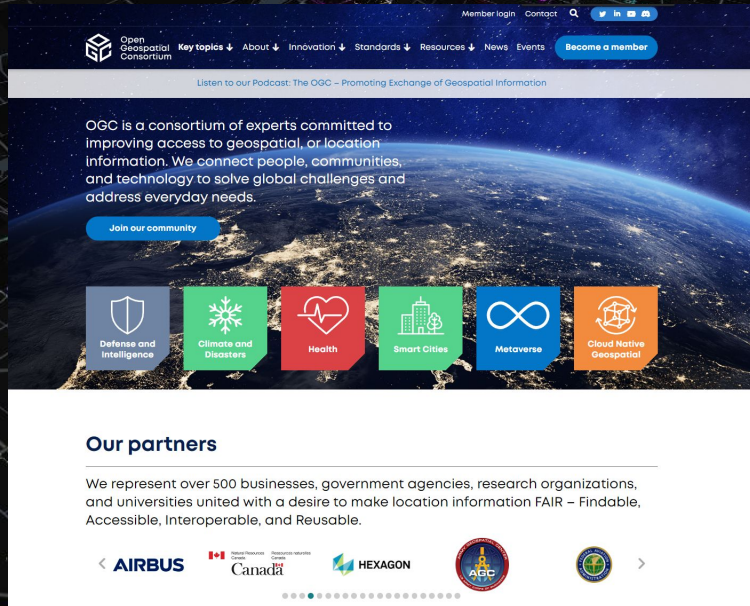
Inconvénients :

Les problèmes de sécurité peuvent être plus complexes à gérer dans une architecture client-serveur, en particulier si les données sensibles sont stockées côté client.

Rétro / Pause



Les services web géospatiaux normés Open Geospatial Consortium (OGC)



- Introduction aux normes OGC
- Les normes de services web géospatiaux (WMS, WMTS, WFS, WCS, WPS)
- Les normes de catalogage de données géospatiales (CS-W, CatMD)
- Les normes d'interopérabilité pour les systèmes d'information géographique (GeoPackage, GeoJSON)
- Les outils et ressources pour la mise en œuvre des normes OGC

Les services web géospatiaux normés Open Geospatial Consortium (OGC)

L'Open Geospatial Consortium (OGC) est une organisation internationale qui développe et promeut des normes pour les données géospatiales et les services web, favorisant l'interopérabilité et l'accessibilité des informations géospatiales.

[Home](#) / [Standards](#)

Standards

OGC Standards are the glue to geospatial information interoperability, and are used by thousands of organizations across the globe and represented in millions of lines of code. They are backed by international organizations, used in proposals, and implemented to speed up the process of innovation.

Any Schemas (XSD, JSON Schema, etc) that support an approved Implementation Standard can be found in the official [OGC Schema Repository](#).



Implementation Standards are different from the Abstract Specification. They are written for a more technical audience and detail the interface structure between software components.

An interface specification is considered to be at the implementation level of detail if, when implemented by two different software engineers in ignorance of each other, the resulting components plug and play with each other at that interface.

Document Title (click to view/download)	OGC Doc.#	Editor	Publication Date
 OGC API - Tiles - Part 1: Core	20-057	Joan Masó, Jérôme Jacovella-St- Louis	2022-11-10

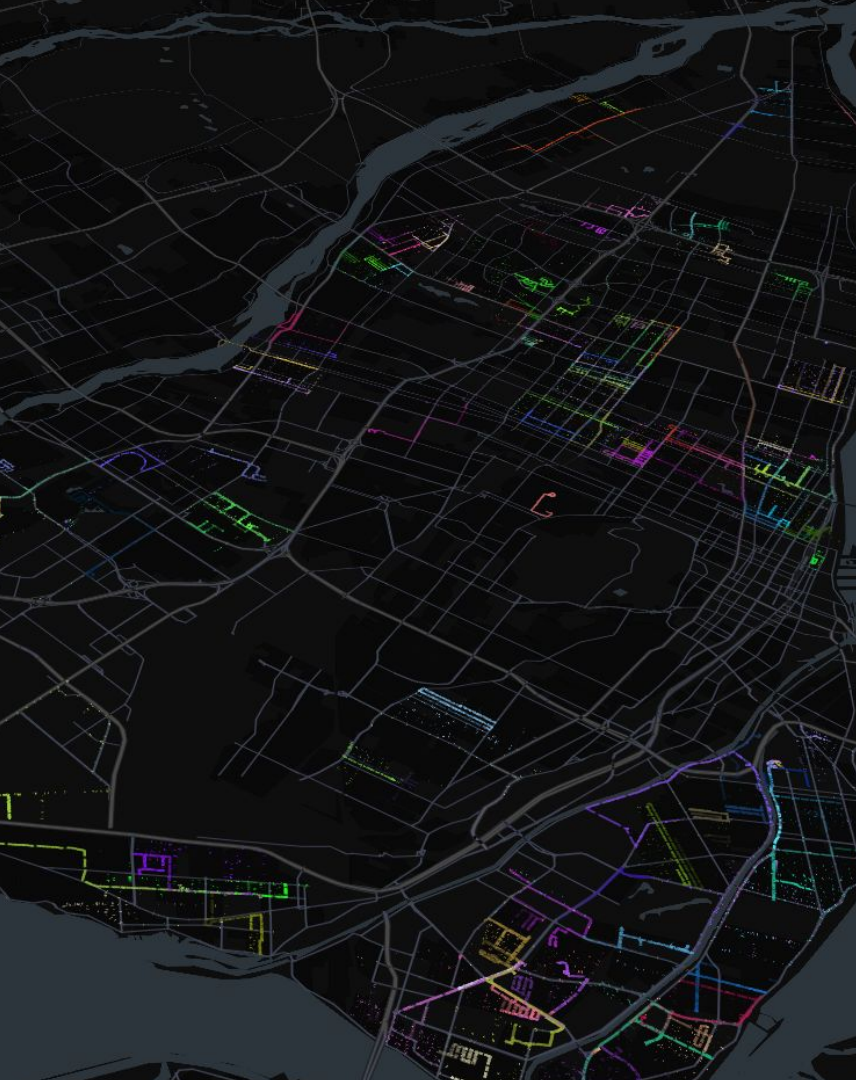
OGC API – Tiles is a standard defining building blocks for creating Web APIs that support the retrieval of geospatial information as t...[Click to continue reading](#)

OGC Standards

- 3D Tiles
- 3dP
- ARML2.0
- Cat: ebRIM App Profile: Earth Observation Products
- Catalogue Service
- CDB
- CityGML
- CityJSON
- Coordinate Transformation
- EO-GeoJSON
- Filter Encoding
- GML in JPEG 2000
- GeoAPI
- GeoPackage
- GeoPose
- GeoSciML
- GeoSPARQL
- Geography Markup Language
- GeoRSS
- Geospatial eXtensible Access Control Markup Language (GeoXACML)
- Geospatial User Feedback (GUF)
- GeoTiff
- GroundwaterML
- HDF5
- I3S
- Indoor Mapping Data Format (IMDF)
- IndoorGML
- KML
- LandInfra/InfraGML
- LAS
- Location Services (OpenLS)
- Moving Features
- NetCDF
- Observations and Measurements
- OGC API – EDR
- OGC API – Features
- OGC API – Processes

[Open Geospatial Consortium](#)

UQÀM



Les services web géospatiaux normés Open Geospatial Consortium (OGC)


Les normes de services web géospatiaux (WMS, WMTS, WFS, WCS, WPS ...) :

Les normes de services web géospatiaux définies par l'Open Geospatial Consortium (OGC) sont des spécifications pour la publication, la découverte et l'accès aux données géospatiales via des services web standardisés.

Web Map Service (WMS) :

Un client peut demander une carte à un serveur WMS, qui répond en générant une image cartographique.

Le WMS permet aux clients d'afficher des cartes sur une variété de plateformes, y compris les navigateurs web, les logiciels SIG et les applications mobiles.

 Copy code

```
<layer queryable="1" opaque="0">
  <Name>resolutions/V</Name>
  <Title>V</Title>
  <Abstract>
    <Text></Text>
  </Abstract>
  <Keywords>
    <Keyword>WCS</Keyword>
    <Keyword>ImageJ</Keyword>
    <Keyword>V</Keyword>
  </Keywords>
  <SRU>EP50:4326</SRU>
  <!--WKT definition of this CRS:
  EPSG:84
  DATUM["World Geodetic System 1984",
    SPHEROID["WGS 84", 6378137.0, 296.1572187563, AUTHORITY["EPSG","7030"]],
    AUTHORITY["EPSG","6326"]],
    PRIME["Greenwich", 0.0, AUTHORITY["EPSG","8901"]],
    UNIT["degree", 0.017453292519943295],
    AXIS["Geodetic longitude", EAST],
    AXIS["Geodetic latitude", NORTH],
    AUTHORITY["EPSG","4326"]]>-->
    <LatLonBoundingBox minx="4.9375" miny="44.96875" maxx="14.9375" maxy="50.96875"/>
    <BoundingBox SRS="EPSG:4326" minx="4.9375" miny="44.96875" maxx="14.9375" maxy="50.96875"/>
    <Dimension name="time" units="ISO8601"/>
    <Dimension name="elevation" units="EPSG:5030" unitSymbol="m"/>
    <Dimension name="UPDATE" units="" />
    <Dimension name="FILEDATE" units="" />
    <Extent name="time" default="current">2013-03-01T00:00:00.000Z,2013-03-01T01:00:00.000Z,2013-03-01T02:00:00.000Z,2013-03-01T03:00:00.000Z,2013-03-01T04:00:00.000Z/<Extent>
    <Extent name="elevation" default="10.0">10.0,35.0,75.0,125.0,175.0,225.0,350.0,450.0,550.0,700.0,900.0,1250.0,1750.0,2500.0/<Extent>
    <Extent name="UPDATE" default="2013-04-08T07:10:20.000Z">2013-04-08T07:10:20.000Z,2013-04-08T07:11:18.000Z,2013-04-08T07:12:10.000Z,2013-04-08T07:13:00.000Z,2013-04-08T07:14:00.000Z/<Extent>
    <Extent name="FILEDATE" default="2013-03-01T00:00:00.000Z">2013-03-01T00:00:00.000Z,2013-03-01T01:00:00.000Z,2013-03-01T02:00:00.000Z,2013-03-01T03:00:00.000Z,2013-03-01T04:00:00.000Z/<Extent>
  </Style>
  <Name>raster</Name>
  <Title>Default Raster</Title>
  <Abstract>A sample style that draws a raster, good for displaying imagery</Abstract>
  <LegendURL width="20" height="20">
    <Format>image/png</Format>
    <OnlineResource xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="simple" xlink:href="http://localhost:8080/geoserver/home?format=png" />
  </LegendURL>
</Style>
</Layer>
```

javascript

Copy code

```
const url = "https://img.nj.gov/imagerywms/Natural2015?" +  
  "bbox={bbox-epsg-3857}" +  
  "&format=image/png" +  
  "&service=WMS" +  
  "&version=1.1.1" +  
  "&request=GetMap" +  
  "&srs=EPSG:3857" +  
  "&transparent=true" +  
  "&width=256" +  
  "&height=256" +  
  "&layers=Natural2015";
```



Les normes de services web géospatiaux (WMS, WMTS, WFS, WCS, WPS)

Web Map Service (WMS) :

URL + Paramètres :

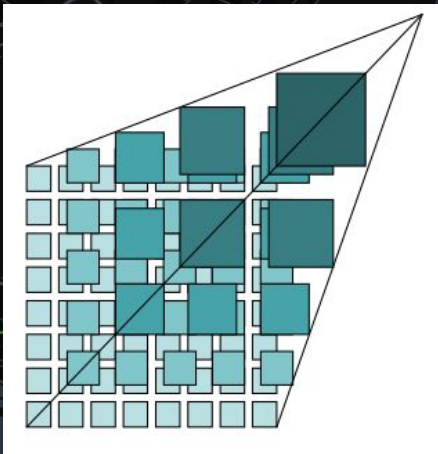
- bbox={bbox-epsg-3857}&
- format=image/png&
- service=WMS&
- version=1.1.1&
- request=GetMap&
- srs=EPSG:3857&
- transparent=true&
- width=256&
- height=256&
- layers=Natural2015

[Add a WMS source | MapLibre GL](#)

javascript

Copy code

```
const url = "http://[votre_serveur_geoserver]/gwc/service/wmts?" +  
  "SERVICE=WMTS" +  
  "&REQUEST=GetTile" +  
  "&VERSION=1.0.0" +  
  "&LAYER=[nom_de_votre_couche]" +  
  "&STYLE=default" +  
  "&TILEMATRIXSET=[nom_de_votre_jeu_de_matrices_de_tuiles]" +  
  "&TILEMATRIX=[identifiant_de_votre_matrice_de_tuiles]" +  
  "&TILEROW=[numéro_de_ligne_de_la_tuile]" +  
  "&TILECOL=[numéro_de_colonne_de_la_tuile]" +  
  "&FORMAT=[format_de_l'image]";
```



Les normes de services web géospatiaux (WMS, WMTS, WFS, WCS, WPS)

Web Map Tile Service (WMTS) :

Norme WMTS qui permet de fournir des cartes pré-générées sous forme de tuiles, qui peuvent être stockées en cache côté client pour une utilisation ultérieure.

Le WMTS améliore la performance et la convivialité des applications en permettant un accès plus rapide aux cartes.

javascript

Copy code

```
const url = "http://localhost:8080/geoserver/gwc/service/wmts?" +  
  "layer=public:list_parcel_hobart_4326" +  
  "&tilematrixset=EPSG:900913" +  
  "&Service=WMTS" +  
  "&Request=GetTile" +  
  "&Version=1.0.0" +  
  "&Format=application/x-protobuf;type=mapbox-vector" +  
  "&TileMatrix=EPSG:900913:{z}" +  
  "&TileCol={x}" +  
  "&TileRow={y}";
```



Les normes de services web géospatiaux (WMS, WMTS, WFS, WCS, WPS)

Web Map Tile Service (WMTS) :

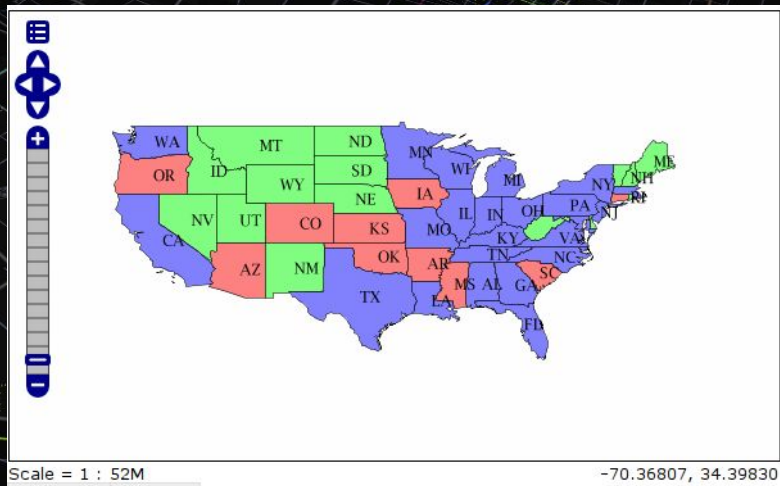
Norme WMTS qui permet de fournir des cartes pré-générées sous forme de tuiles, qui peuvent être stockées en cache côté client pour une utilisation ultérieure.

Le WMTS améliore la performance et la convivialité des applications en permettant un accès plus rapide aux cartes.

javascript

Copy code

```
const url = "http://[votre_serveur_geoserver]/wfs?" +  
  "service=WFS" +  
  "&version=1.1.0" +  
  "&request=GetFeature" +  
  "&typeName=[nom_de_votre_couche]" +  
  "&outputFormat=[format_de_sortie]" +  
  "&maxFeatures=[nombre_maximum_de_caractéristiques_a_retourner]" +  
  "&CQL_FILTER=[filtre_de_caractéristiques]";
```



Les normes de services web géospatiaux (WMS, WMTS, WFS, WCS, WPS)

Web Feature Service (WFS)

Norme pour la publication et l'accès à des données géographiques vectorielles, telles que des points, des lignes ou des polygones.

Le WFS permet aux clients d'interroger et de récupérer des données géospatiales pour une utilisation dans des applications telles que la planification urbaine, l'analyse environnementale et la gestion de l'eau.

Les normes de services web géospatiaux (WMS, WMTS, WFS, WCS, WPS)

```
const baseUrl = "http://localhost:8080/geoserver/geo7630/ows?"
const serviceType = "service=WFS"
const version = "version=1.0.0"
const request = "request=GetFeature"
const typeName = "typeName=geo7630:analyse_bixi_par_station"
const outputFormat = "outputFormat=application/json"

const WFSUrl = baseUrl + '&' + serviceType + '&' + version + '&' + request + '&' + typeName

map.addSource( 'wfs-test-source', {
  type: 'geojson',
  data: WFSUrl
})
```



- Web Feature Service (WFS)
- BaseURL
- ServiceType
- Version
- Request
- TypeName
- Output

[Add a GeoJSON line | MapLibre GL JS Docs](#)

Les normes de services web géospatiaux (WMS, WMTS, WFS, WCS, WPS)

```
map.addSource( 'wfs-test-source', {  
  type: 'vector',  
  url: 'http://localhost:8080/vector-tile-server/tileJSON.json'  
})
```

```
json Copy code  
{  
  "tilejson": "2.2.0",  
  "name": "My Tileset",  
  "tiles": [  
    "https://example.com/tiles/{z}/{x}/{y}.png"  
  ],  
  "minzoom": 0,  
  "maxzoom": 10,  
  "bounds": [  
    -180,  
    -85,  
    180,  
    85  
  ],  
  "attribution": "© OpenStreetMap contributors"  
}
```

Vector tiles (*Bonus (non OGC)*)

2 type de sources

TileJSON config file (url)

Tiles (tiles)

Les normes de services web géospatiaux (WMS, WMTS, WFS, WCS, WPS)

Vector tiles *(Bonus (non OGC))*

2 type de sources

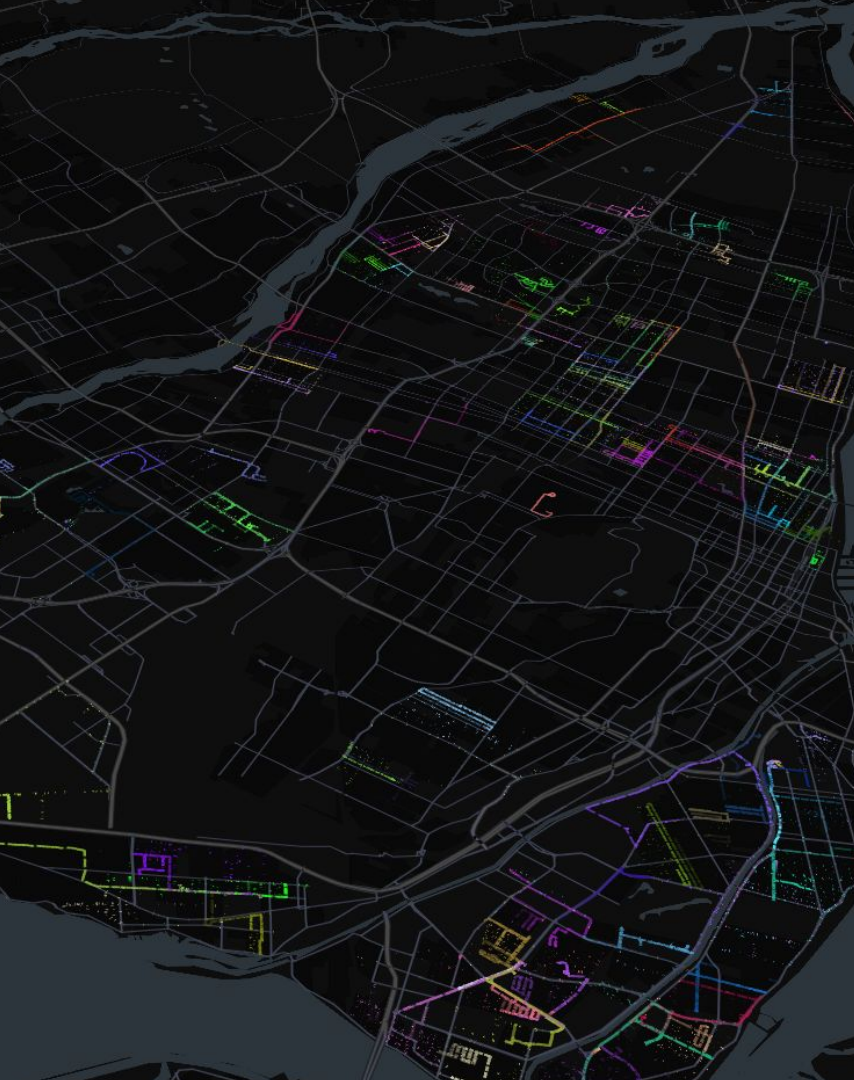
TileJSON config file (url)

Tiles (tiles)

```
map.addSource('mapillary', {  
  'type': 'vector',  
  'tiles': [  
    'https://tiles.mapillary.com/maps/vtp/mly1_public/2/{z}/{x}/{y}.pbf'  
  ],  
  'minzoom': 6,  
  'maxzoom': 14  
});
```

```
map.addSource('mapillary', {  
  'type': 'vector',  
  'tiles': [  
    'https://tiles.mapillary.com/maps/vtp/mly1_public/2/{z}/{x}/{y}.mvt'  
  ],  
  'minzoom': 6,  
  'maxzoom': 14  
});
```

[Add a third party vector tile source | MapLibre GL](#)



Les normes de services web géospatiaux (WMS, WMTS, WFS, WCS, WPS)

Web Coverage Service (WCS)

Norme pour la publication et l'accès à des données géospatiales sous forme de données couvrant des surfaces continues, telles que des images satellite, des modèles numériques d'élévation et des données météorologiques.

Le WCS permet aux clients d'interroger et de récupérer des données géospatiales à haute résolution pour une utilisation dans des applications telles que la surveillance de l'environnement et la gestion des ressources naturelles.



Les normes de services web géospatiaux (WMS, WMTS, WFS, WCS, WPS)

Le Web Processing Service (WPS)

Norme permet aux clients d'envoyer des requêtes de traitement géospatial à un serveur et de recevoir des résultats en retour.

Le WPS est utilisé pour automatiser des tâches de traitement géospatial telles que la création de zones tampons, la classification d'images et l'analyse des données spatiales.

Rétro / Pause





Les normes de catalogage de données géospatiales (CS-W, CatMD)

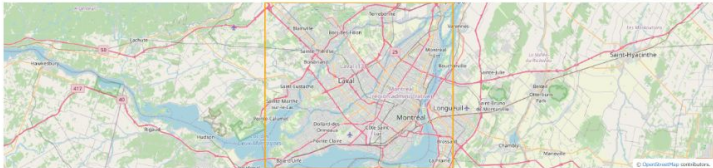
Les normes de catalogage de données géospatiales définissent la manière dont les métadonnées doivent être stockées, organisées et échangées entre les différents systèmes d'information géographique et les services web géospatiaux.

Réseaux d'eaux pluviaux et eaux usées

Finalisé

Inventaire des réseaux d'eaux pluviaux et usées sur l'île de Montréal

Découvrir les données



Contraintes d'accès et d'utilisation

Format

shape file

geodatabase

PDF

JPEG

Ces données sont fournies pour les fins exclusives de consultation ou de recherche dans le cadre de travaux municipaux à la Ville de Montréal, elles ne doivent pas être utilisées à d'autres fins.

- Contraintes d'accès : Être employé ou à contrat pour la Ville de Montréal, et avoir obtenu expressément les droits d'accès.

- Contraintes d'utilisation: Pour des fins de consultation et de recherche.

Informations techniques

Création 09-11-2003

Mots-clés

Langue

Type de représentation

Fréquence de mise à jour Hebdomadaire

Thème

Catégories

Échelle

Système de coordonnées

Accessoire

Basin de Retention

Chambre

Inspection

Puisard

Raccord

Regard

Relié terrain

Segment

français

Vecteur

Télécommunication, approvisionnement et énergie

1:1000

EPSG:32188

Généalogie

Mise à jour servant au plan d'intervention, Info excavation, maximo.

Contact pour la ressource

Point de contact

28 SIRR - Division De La Géomatique

Les normes de catalogage de données géospatiales (CS-W, CatMD)

Le Catalog Service for Web (CS-W) est une norme OGC qui définit un protocole standardisé pour la recherche et la découverte de métadonnées géospatiales dans un catalogue.

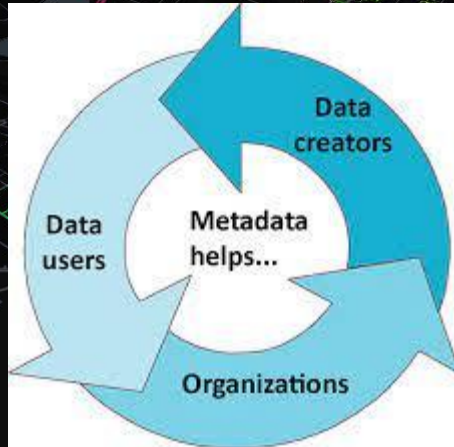
UQÀM



Les normes de catalogage de données géospatiales (CS-W, CatMD)

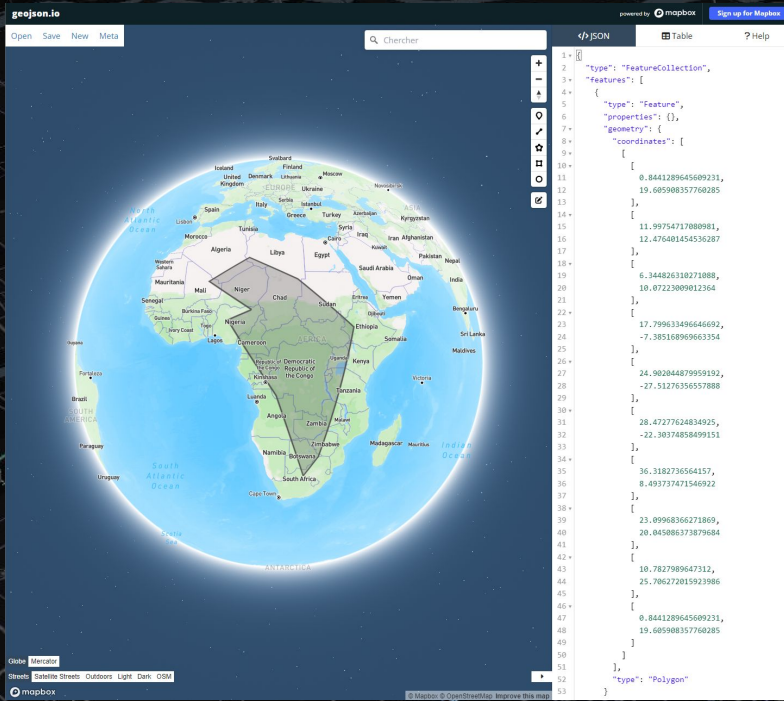
Le Catalogue des métadonnées (CatMD) est une norme OGC qui fournit un modèle de métadonnées géospatiales pour décrire les données géospatiales stockées dans un catalogue.

Les normes de catalogage de données géospatiales (CS-W, CatMD)



Ces normes de catalogage sont essentielles pour faciliter l'interopérabilité et l'échange de données entre les différents SIG et les services web géospatiaux.

Rétro / Pause



[GeoJSON.io](https://geojson.io)

Les normes d'interopérabilité pour les systèmes d'information géographique (GeoPackage, GeoJSON)

Le GeoJSON

Norme OGC qui spécifie un format de données géospatiales basé sur le format de données JSON.

Il permet de stocker des données géospatiales sous forme de fichiers texte, ce qui facilite l'échange de données entre différents SIG et applications web.

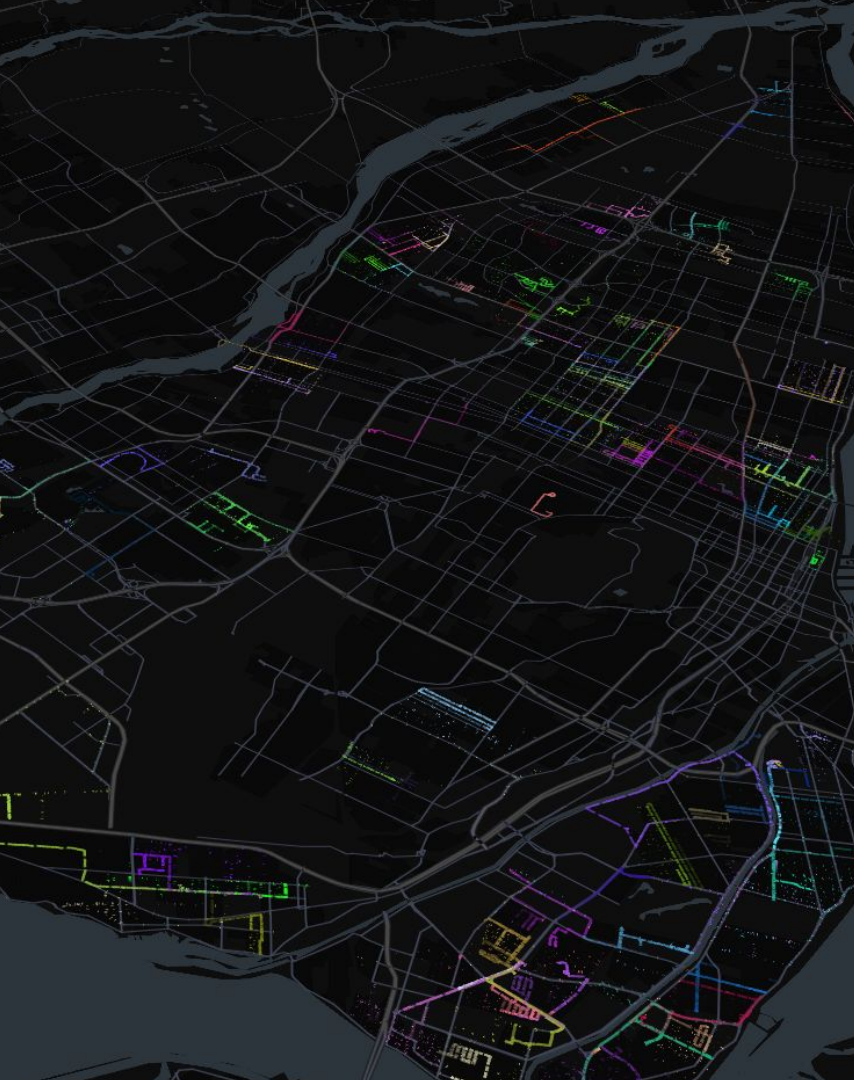
Le GeoJSON peut être utilisé pour représenter différents types de données géospatiales, tels que les points, les lignes, les polygones, les multi-géométries et les collections de géométries.

Les services web géospatiaux normés Open Geospatial Consortium (OGC)



En utilisant ces normes d'interopérabilité, les utilisateurs peuvent facilement échanger des données géospatiales entre différents SIG et applications de traitement géospatial sans avoir à se soucier des différences de format et de structure des données.

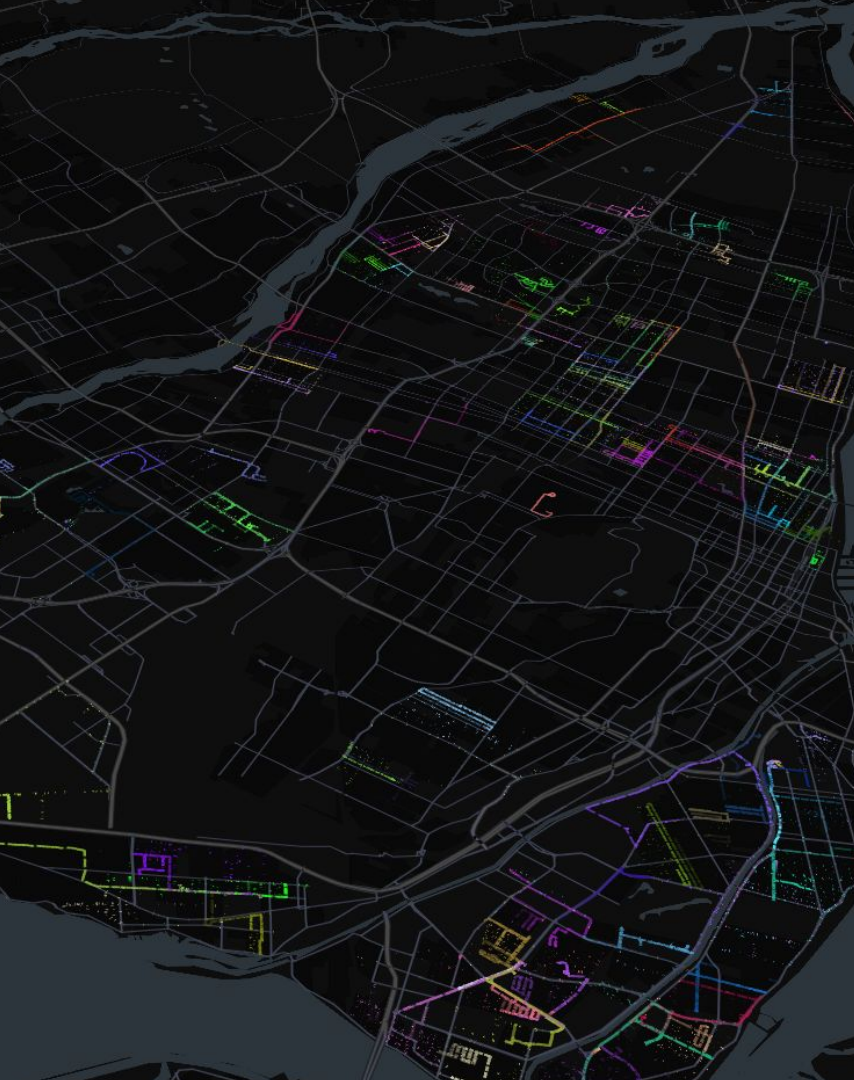
Cela permet également une meilleure intégration des données géospatiales dans les systèmes d'information et une utilisation plus efficace des ressources géospatiales.



Les outils et ressources pour la mise en œuvre des normes OGC

Pour mettre en œuvre les normes de l'Open Geospatial Consortium (OGC), il existe de nombreux outils et ressources disponibles, notamment :

- Les logiciels SIG open source
- Les API géospatiales
- Les implémentations de référence OGC
- Les profils d'implémentation
- Les communautés d'utilisateurs et de développeurs
- Les formations et les tutoriels en ligne



Les outils et ressources pour la mise en œuvre des normes OGC

Les logiciels SIG open source :

Il existe de nombreux logiciels SIG open source qui prennent en charge les normes OGC, tels que QGIS, GeoServer, MapServer, PostGIS, etc.



Les outils et ressources pour la mise en œuvre des normes OGC

Les API géospatiales :

Telles que OpenLayers, Leaflet, Google Maps API, etc., fournissent des outils pour afficher et manipuler des données géospatiales conformément aux normes OGC.

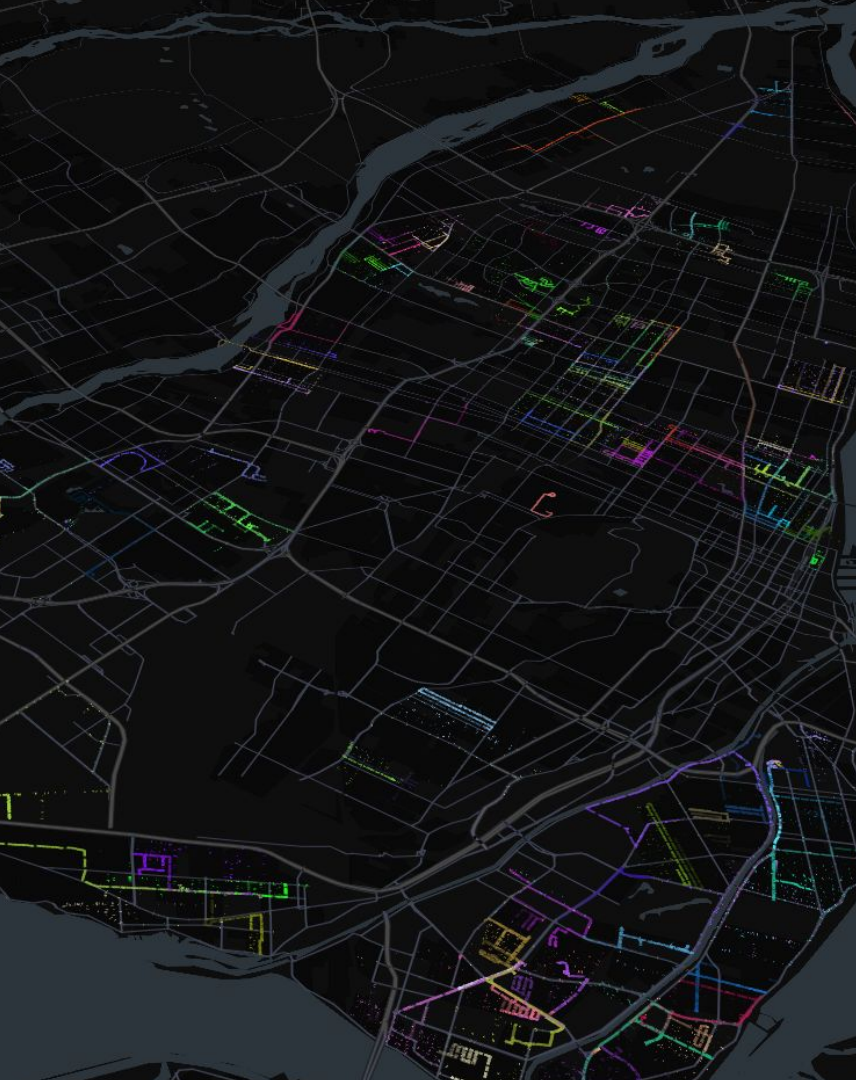


Les outils et ressources pour la mise en œuvre des normes OGC

Les implémentations de référence OG

Logiciels de référence qui implémentent les normes OGC.

Ces implémentations sont disponibles gratuitement et peuvent être utilisées pour tester la compatibilité des implémentations de logiciels SIG avec les normes OGC.



Les outils et ressources pour la mise en œuvre des normes OGC

Les profils d'implémentation

Les profils d'implémentation OGC fournissent des directives pour la mise en œuvre des normes OGC dans des contextes spécifiques, tels que l'environnement INSPIRE en Europe.



Les outils et ressources pour la mise en œuvre des normes OGC

Les communautés d'utilisateurs et de développeurs OGC

Elles fournissent un forum pour discuter de l'implémentation des normes OGC et partager des expériences et des meilleures pratiques.



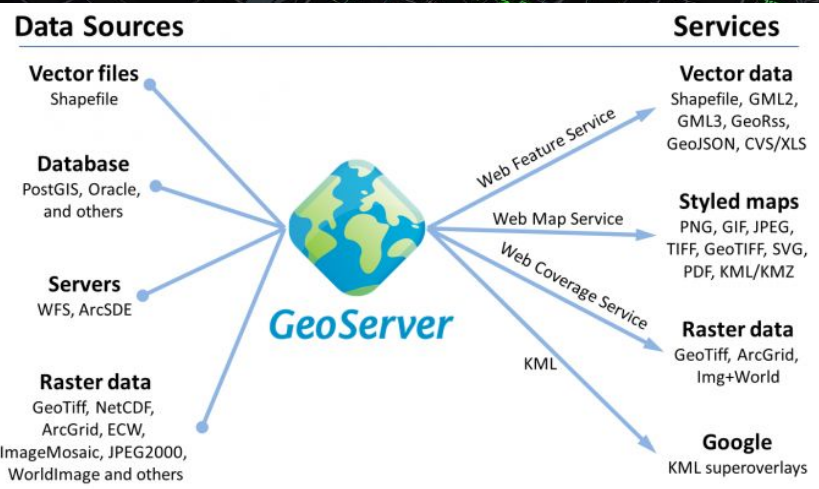
Les outils et ressources pour la mise en œuvre des normes OGC

Les formations et les tutoriels en ligne

Il existe de nombreuses formations et tutoriels en ligne pour apprendre à mettre en œuvre les normes OGC, notamment des cours en ligne gratuits sur la plateforme MOOC ou des ressources payantes proposées par des organismes de formation en géomatique.

Rétro / Pause

Diffusion des données géospatiales avec GeoServer



- Introduction à GeoServer
- Configuration de GeoServer
- La gestion des couches et des groupes de couches
- La publication de données géospatiales avec GeoServer
- La personnalisation de l'apparence de la carte dans GeoServer

[Tutorials — GeoServer 2.24.x User Manual](#)



Diffusion des données géospatiales avec GeoServer

La publication de données géospatiales avec GeoServer

- Création d'un service WMTS
- Création d'un service WFS