

Vector Tiles : concepts et applications

camptocamp 

INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS

Table des matières

1. Présentation des *Vector Tiles*

- Principe
- Innovations et limites

2. Cas d'utilisation

3. Implémentations

4. Questions / réponses



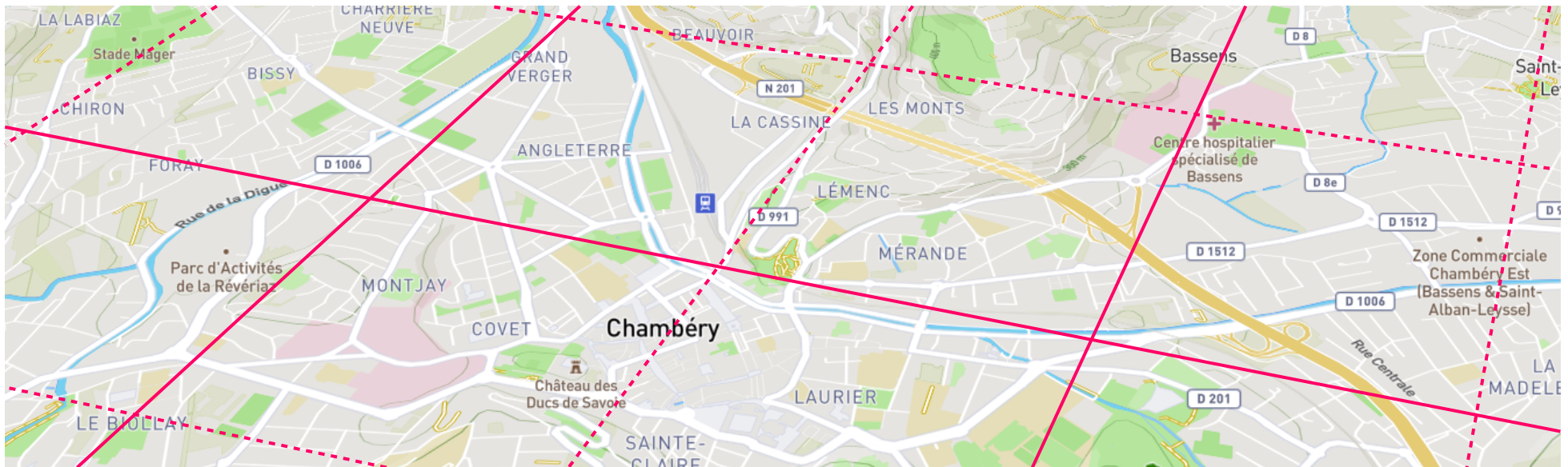
1. Présentation des *Vector Tiles*



1. Présentation

Concepts généraux

- « *Vector Tiles* » : jeux de données vectérielles (*features*) tuilées
- Optimisé pour la visualisation sur internet
- Déporte le travail de rendu sur le client



1. Présentation

Aspects innovants

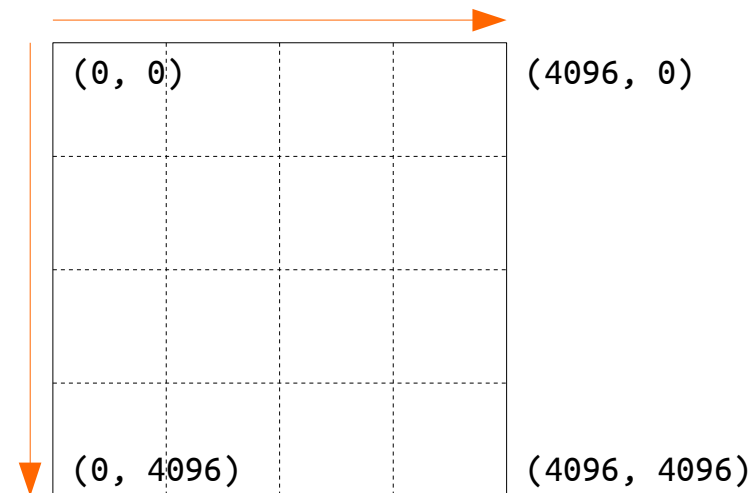
- Mise à contribution de l'augmentation des **capacités de rendu**
- Séparation des **données** et de la **présentation**
- Réduction de la consommation de **bande passante**
- Meilleure qualité de rendu



1. Présentation

Format des données

- Pas (encore) de standard OGC
- Format majoritaire : **Mapbox Vector Tiles** (.mvt)
 - Binaire, compressé par PBF (*protocol buffers*)
 - Web Mercator
 - Coordonnées relatives, en pixel :



- Autres : TopoJSON, GeoJSON...



1. Présentation

Format des données

- Style indépendant des données
- Format majoritaire : **Mapbox Style**
 - Éditeurs graphiques (Mapbox Studio / Maputnik)
 - Nombreuses possibilités
 - Défini en JSON
- Le client a liberté totale sur le style d'affichage



1. Présentation

Exemple : édition de style

The screenshot displays the Maputnik v1.2.0 web application interface. The top navigation bar includes links for Open, Export, Sources, Style Settings, Inspect Mode, and Help. The left sidebar shows a list of layers, with 'landcover_wood' selected. The main panel is divided into three sections: Layer, Filter, and JSON Editor.

Layer Section:

- ID: landcover_wood
- Type: Fill
- Source: openmaptiles
- Source Layer: landcover
- Min Zoom: 10
- Max Zoom: 24
- Comments: COMMENTE ...

Filter Section:

Paint properties

- Opacity: 8 0
- Opacity: 12 1
- Color: rgb(220,224,220)
- Available: 2
- Outline color: 1
- Pattern: 2
- Tramplage: 3 0
- Tramplage color: 0
- Tramplage opacity: 2 Map

JSON Editor Section:

```
1 {
2   "id": "landcover_wood",
3   "type": "fill",
4   "source": "openmaptiles",
5   "source-layer": "landcover",
6   "minzoom": 10,
```

The right side of the interface shows a map of a region in France, with various towns and roads visible. The zoom level is set to 11.71.



1. Présentation

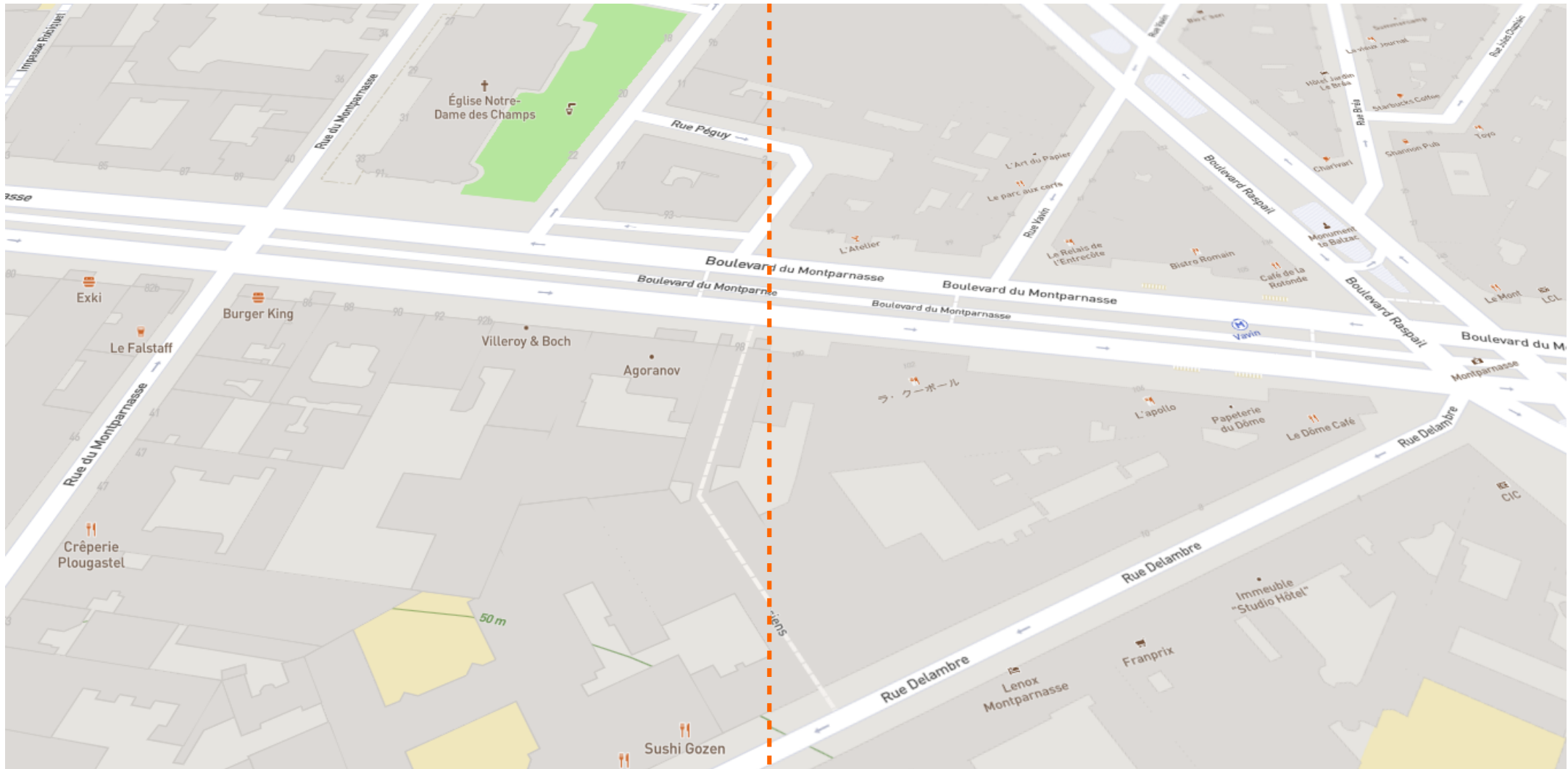
Comparaison avec services OGC

- Pas d'opération *GetCapabilities*, *DescribeFeatureType*
- WFS :
 - Géométrie peut être coupée/simplifiée
 - Données attributaires présentes
 - Équivalence *FeatureType* (WFS) / *Layers* (VT)
 - Pas de transaction
- WMS, WMTS :
 - Aucune image stockée sur le serveur
 - Style non figé par le fournisseur des données
 - Meilleure qualité de rendu



1. Présentation

Comparaison tuiles vecteur / *raster*



1. Présentation

Limites

- Interactions **très limitées** avec les données
- **Dégradation** des géométries
- Rendu peut être très **coûteux** pour le client
- A données identiques, **rendu variable** selon implémentation client
- Non adapté pour données *raster*



2. Cas d'utilisation



2. Cas d'utilisation

Mise à jour fréquente de données

- Production *vector tiles* à partir de données sources environ **50x plus rapide** que des tuiles raster
- Si mise en cache, taille sur le disque environ **1000x moindre** que des tuiles raster
- Possibilité de produire les *vector tiles* en flux tendu (pas de mise en cache)



2. Cas d'utilisation

Personnalisation du rendu

- Possibilité d'utiliser **plusieurs styles** à l'affichage (ex. : en fonction du thème)
- Styles **pré-configurés** (ex. : *Mapbox Styles*) ou **spécifiques** à l'application client
- Fonctions de rendu avancées :
 - Perspective / profondeur
 - Regroupement de *features*
 - Détection de collision texte / symbole
 - *Heatmaps*
 - Effets visuels



2. Cas d'utilisation

Publication de données complexes

- Certaines sources peuvent contenir des **millions** de points/géométries
- Possibilité de **simplifier/filtrer** les géométries dans les *vector tiles* afin de visualiser ces données
- Découpage de géométries très étendues (ex. : océans) pour faciliter leur rendu



2. Cas d'utilisation

Manipulations à l'affichage

- Possibilité de **filtrer** les *features* par attribut / type d'objet
- **Isolation** d'un certain type d'objet
- Mise en **surbrillance** d'une *feature* sur plusieurs tuiles
- Inspection de *feature*



3. Implémentations



3. Implémentations

Librairies disponibles

- Génération de tuiles :
 - **Tippecanoe** (Mapbox) : utilise ShapeFile ou GeoJSON en entrée
 - **Tegola** (Go Spatial) : PostGIS en entrée, mise en cache & publication
 - **Geoserver** : simple extension, mise en cache par GeoWebCache
- Publication de tuiles :
 - **Tileserver GL** (Klokan) : Node.js, tuiles vecteur et raster, support WMTS
 - **Tileserver PHP** (Klokan)
- Affichage client :
 - **Mapbox GL** : support complet, natif (Android, iOS, etc.) et Javascript
 - **Openlayers** : support des styles Mapbox, pas de WebGL
 - **Leaflet** : support expérimental



3. Implémentations

Exemple d'infrastructure

- Serveur de génération des tuiles :
 - Données stockées en **GeoJSON**
 - Génération de tuiles à l'aide de **Tippecanoe** (par ex. : CRON job)
 - Stockage des tuiles dans un **cache S3**
- Serveur de publication de carte
 - Construction d'un style avec **Mapputnik**, stockage en JSON sur le serveur
 - Affichage de la carte (style + données) avec **Openlayers** dans une page web

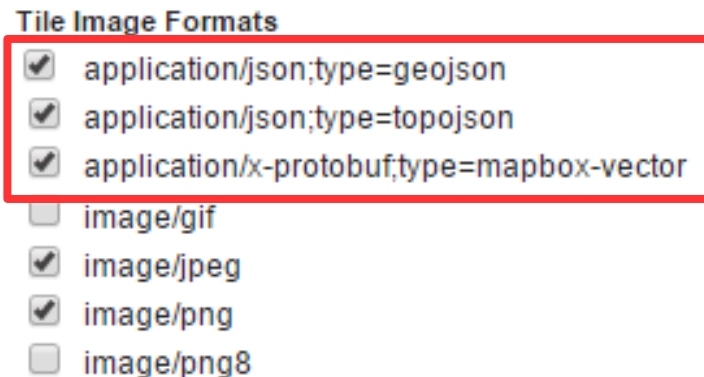


3. Implémentations

Publier des données existantes en VT

■ Geoserver (≥ 2.11) :

- Mise en place **très rapide**
- **Activer les formats** liés aux *vector tiles* sur une source de donnée tuilée :



- Utiliser SLD pour filtrer les géométries (paramètre *scale*)
- Support Mapbox Style (conversion SLD, rendu *raster*)

■ Mapserver (≥ 7.2):

- Idem sauf support MB Style



Questions / réponses



Merci !

Olivier Guyot
développeur géospatial @ **camptocamp**

Email : olivier.guyot@camptocamp.com

Github : [@jahow](#)



to camp 

camp **to** camp

INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS