# UTILISER OPENSTREETMAP GRÂCE À OVERPASS

Julien Fastré

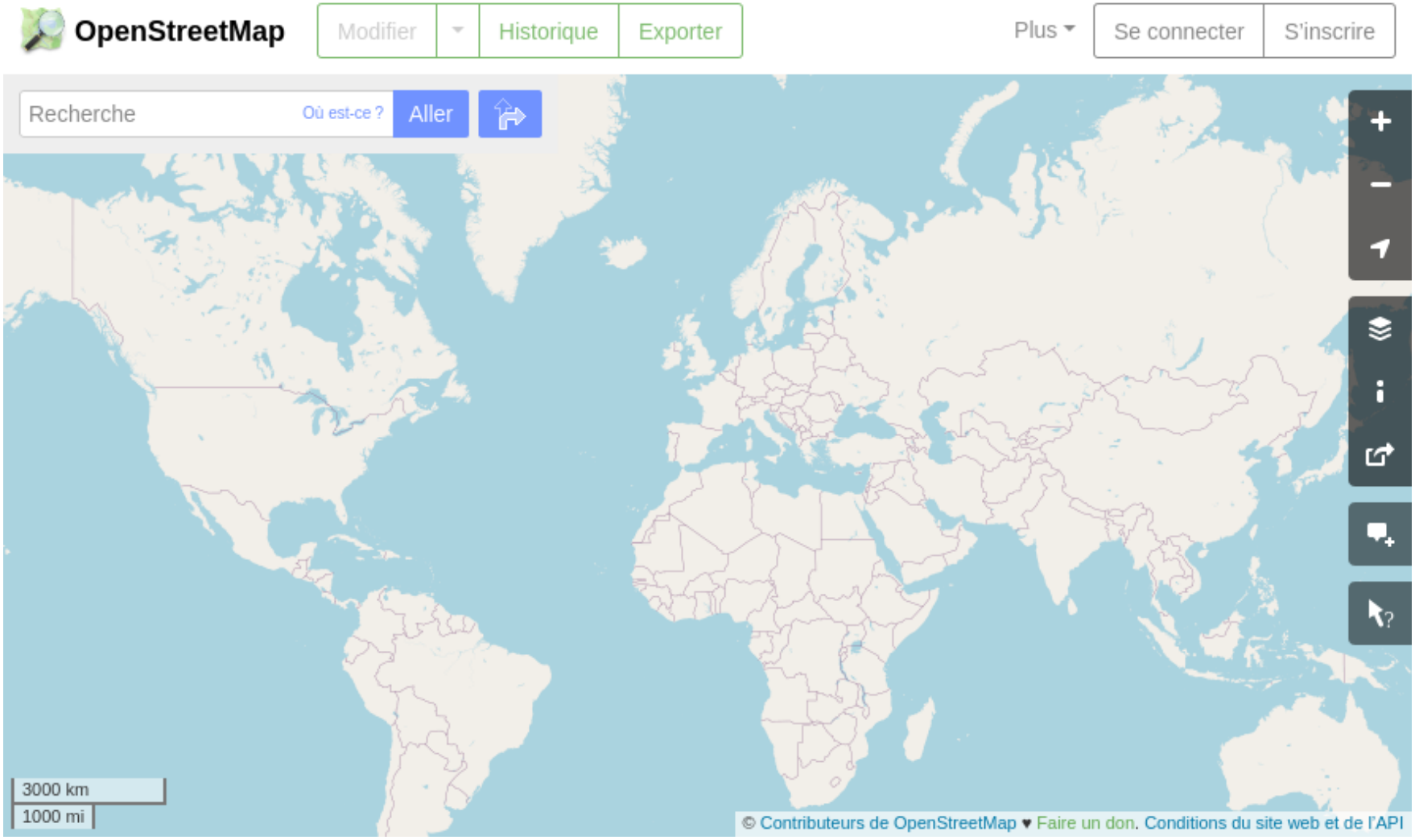
10 février 2021

# SOMMAIRE

1. OSM, c’est quoi ?
2. OSM: où sont les données ?
3. Overpass API
4. Rappel du modèle de données OSM
5. Présentation du langage Overpass et requêtes sympas

# OSM, C'EST QUOI ?

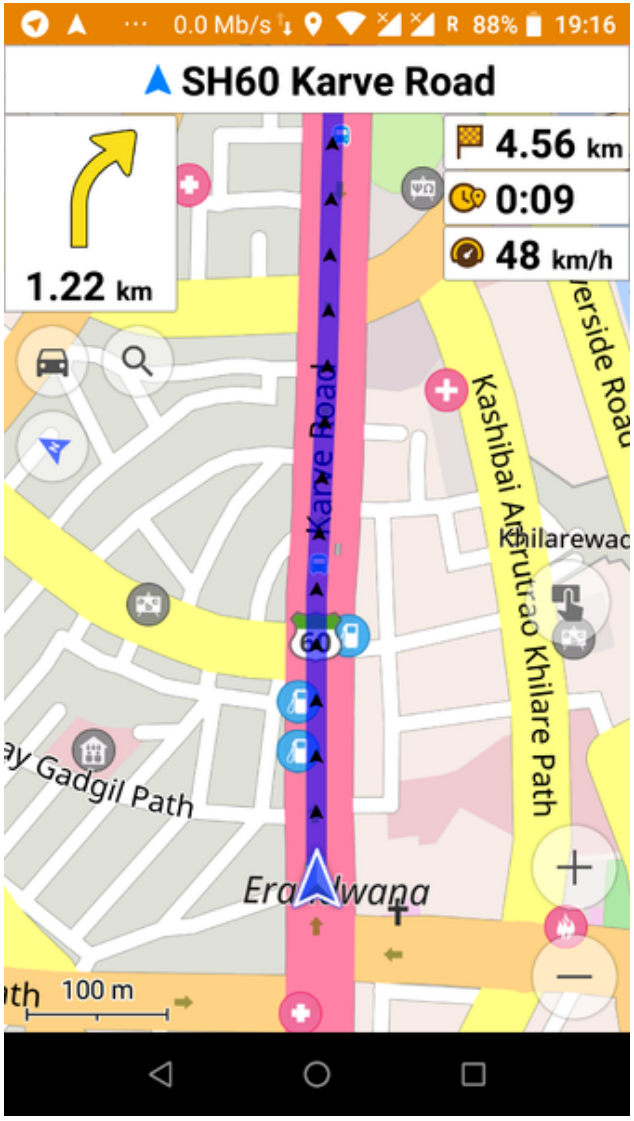
# OSM, C'EST UNE CARTE ?



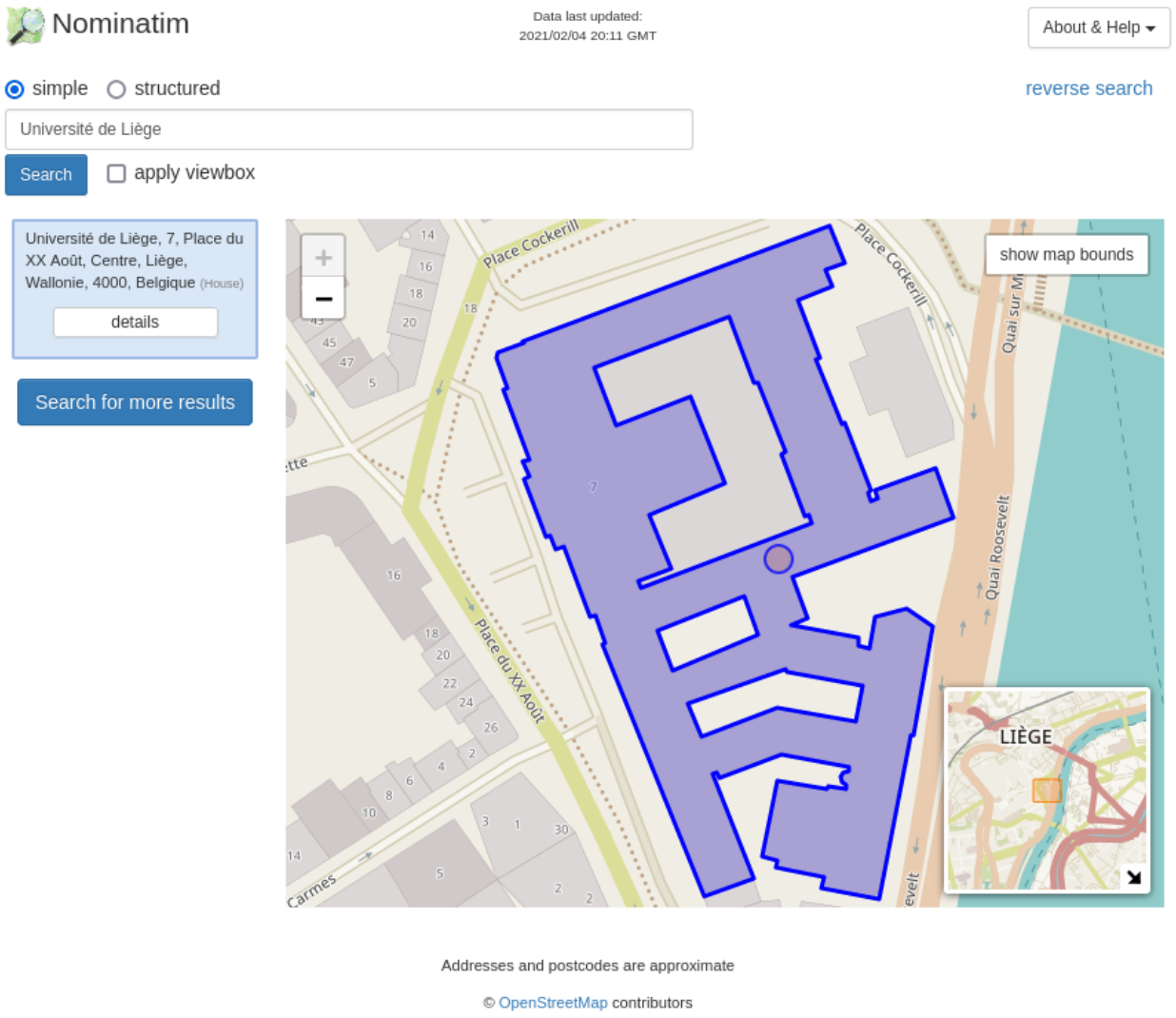
# BASE DE DONNÉE GÉOGRAPHIQUE OUVERTE

Permet de générer des cartes…

# … de calculer des itinéraires …

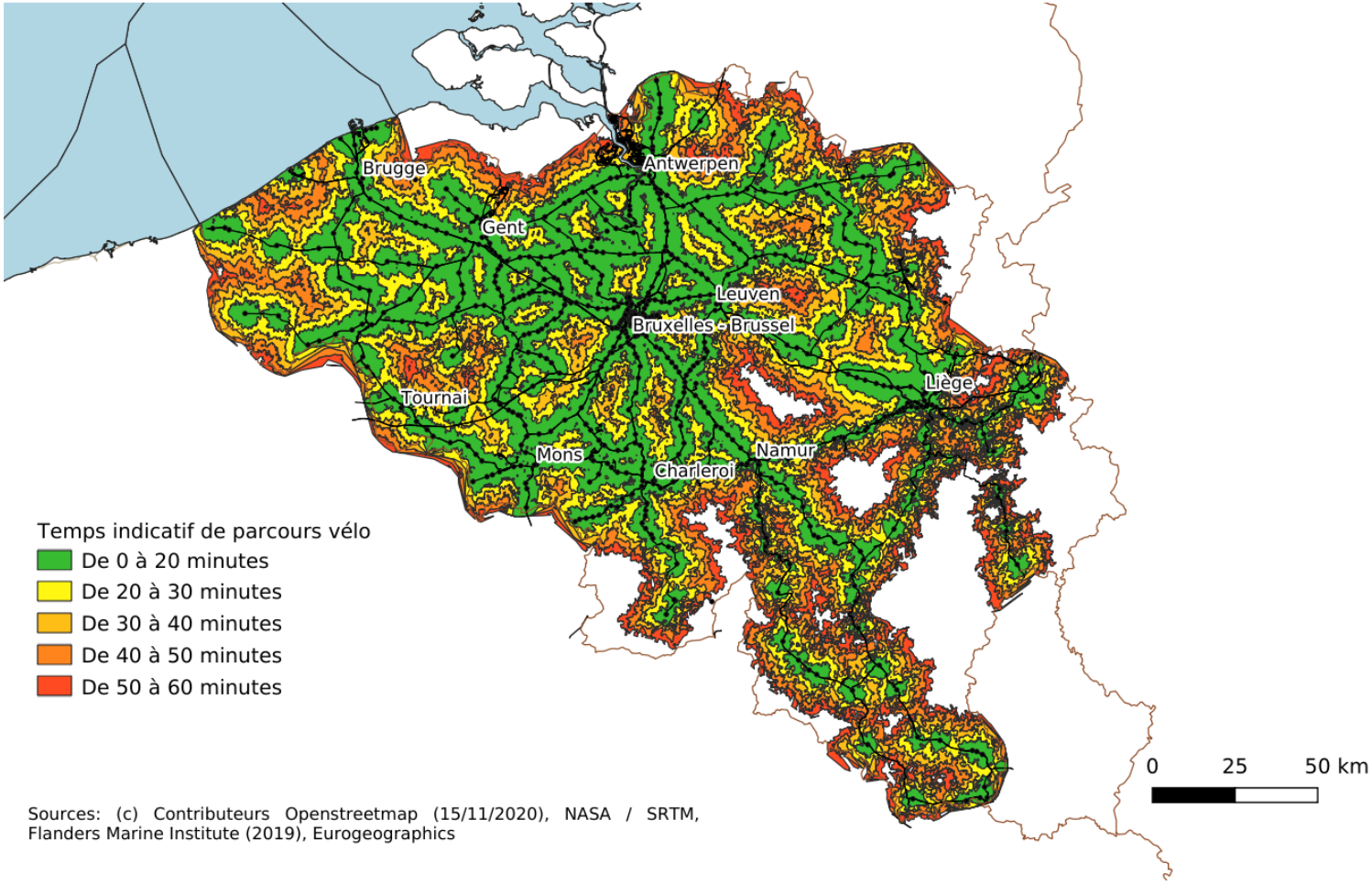


# … de géocoder / reverse geocoder …



# … d'analyser le territoire …

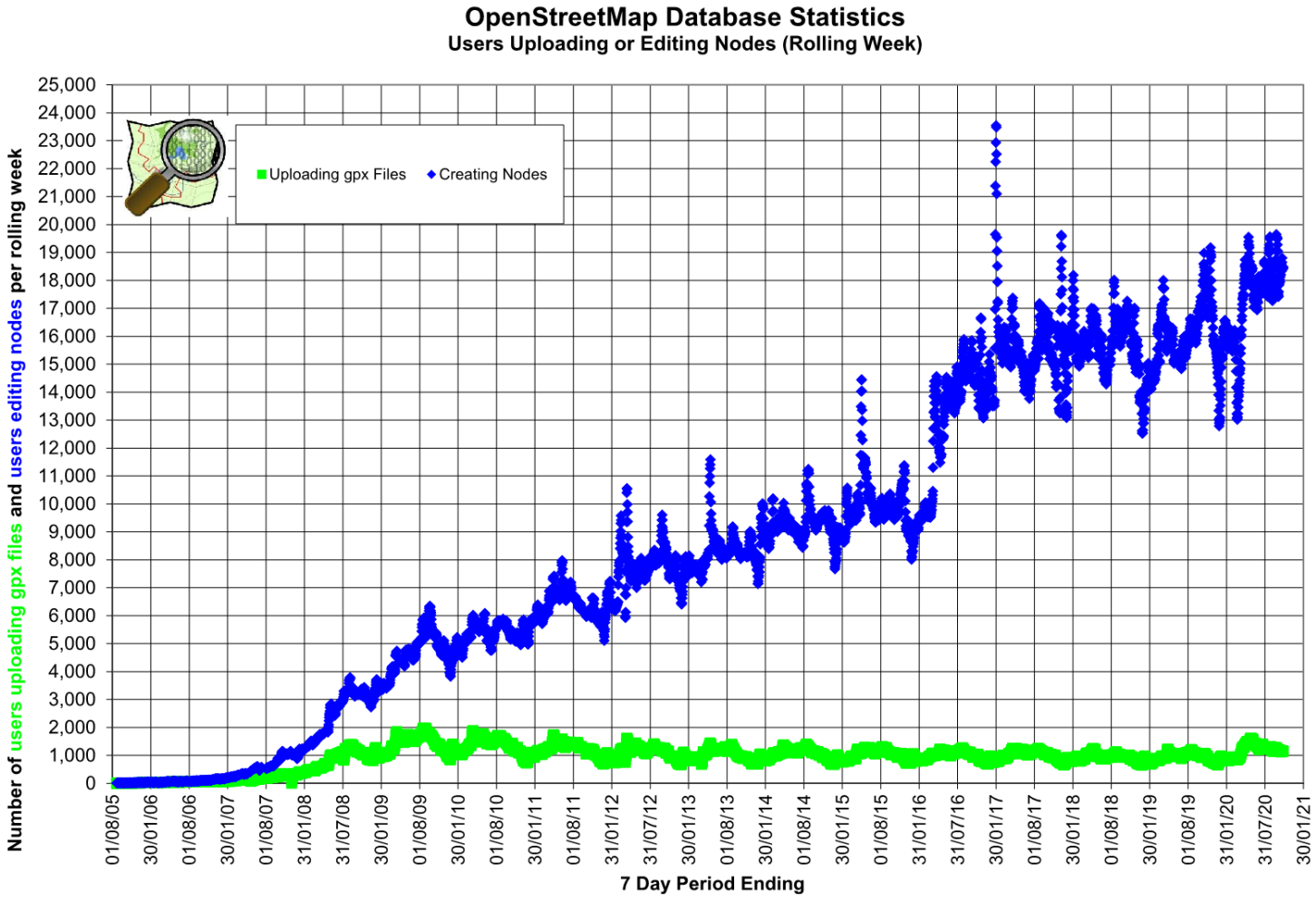
## Temps indicatifs de parcours vélo depuis les gares belges




Sources: (c) Contributeurs Openstreetmap (15/11/2020), NASA / SRTM, Flanders Marine Institute (2019), Eurogeographics

# OPENSTREETMAP, C’EST QUI ET COMMENT ?

# CONTRIBUTEURS & COMMUNAUTÉS



# ORGANISATIONS

- OSM Foundation
- Local Chapters

  En Belgique: osmbe
- Humanitarian OSM Team (HOT)

# OUTILS DE COORDINATION

- Wiki
- Mailing-lists
- Vote
- Messagerie (matrix, salon osm.be)

# OUTILS D'ÉDITION

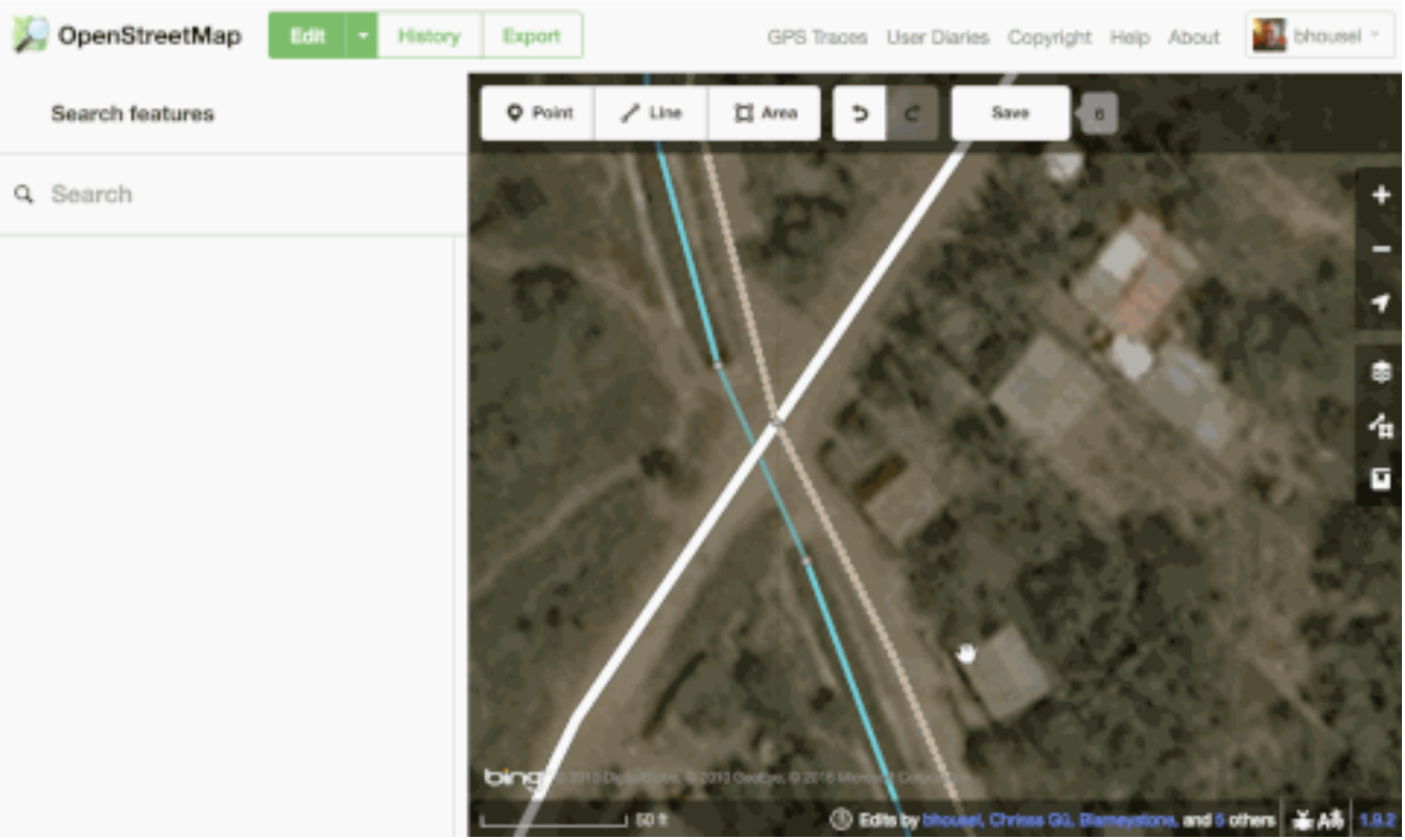
- iD
- JOSM
- Plotlatch
- Vespucci

Gamification sur Android: StreetComplete

# QUALITÉ DES DONNÉES

- Osmose
- OSM Inspector
- …

# OSM, MODIFIABLE PAR “N’IMPORTE QUI”



# LICENSE ODBL

## Open Database License

Vous êtes libres de:

- **Partager** la base de donnée;
- **Créer** des œuvres à partir des données;
- **Adapter** ou modifier la base de donnée;

Aussi longtemps que:

- **attribution** des auteurs;
- **BDD** adaptée sous licence compatible…
- …et **sans DRM**;

Les œuvres créées à partir de la base de donnée peuvent être distribuées sous une licence plus restrictive ou propriétaire.

**Exemples**: cartes, applications, etc.

# OÙ SONT LES DONNÉES ?

# DES ÉDITIONS À CHAQUE MINUTE...

# LES DONNÉES BRUTES

- fichiers planet.osm
  - dump hebdomadaire (disponible aussi sur torrent);
  - format `.pbf` ou XML (`.osm`);
- minutely diff
  - changesets effectués chaque minutes

# DES EXTRAITS GÉOGRAPHIQUES

- Téléchargement depuis l'api

  Surface limitée

- https://download.bbbike.org/osm/

  Sélection d'une aire spécifique sur la carte.

- download.geofabrik.de

  Exports par pays / région. Mises à jour quotidienne.

# ZOOM SUR GEOFABRIK

Formats:

- `.osm` (xml)
- `.pbf` (protocol buffer)

Possibilité de s'authentifier => accès à l'username et userid

# COMMENT LES EXPLOITER ?

1. Osmosis (Java, en mode “light-maintenance mode”)

   Permet un filtre par bbox ou par polygone, l’import dans une DB

2. imposm et osm2pgsql

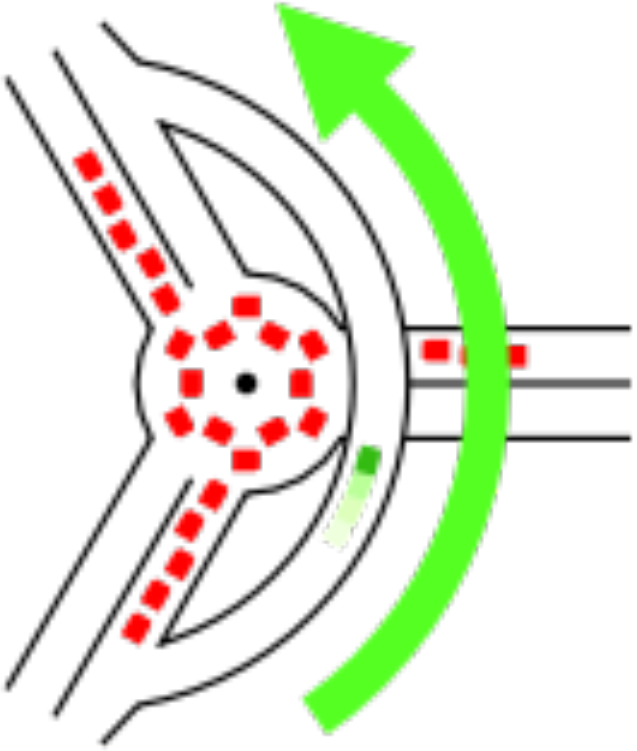
   Import et mise à jour d’une BDD Postgis, orienté rendu carto.

# CONCLUSION...

- Gros volume de données;
- Difficulté de réaliser des scripts;
- (et on n’a pas encore évoqué le modèle de données…)

**Question: n’y a-t-il pas plus simple ?**

# OVERPASS API



Overpass
API

# A QUOI ÇA SERT ?

- Lecture des données OSM;
- Optimisé pour la requête;
- Langage de requête dédié;
- Permet de travailler avec les données historiques;
- Licence AGPLv3.0;
- Plusieurs instances publiques:

  https://overpass-api.de

# LANGAGES DE REQUÊTES

- Soit en XML;
- Soit Overpass-QL

  Langage procédural, impératif.

```
[out:json][timeout:900];
area["name"="Liège"]->.sa;
nw["amenity"="restaurant"](area.sa);
out;
```

# RÉSULTATS


```
{
  "version": 0.6,
  "generator": "Overpass API 0.7.56.8 7d656e78",
  "osm3s": {
    "timestamp_osm_base": "2021-02-05T14:16:03Z",
    "timestamp_areas_base": "2021-02-05T13:44:03Z",
    "copyright": "The data included in this document is from
        www.openstreetmap.org. The data is made available under
        ODbL."
  },
  "elements": [

{
  "type": "node",
  "id": 254279248,
```

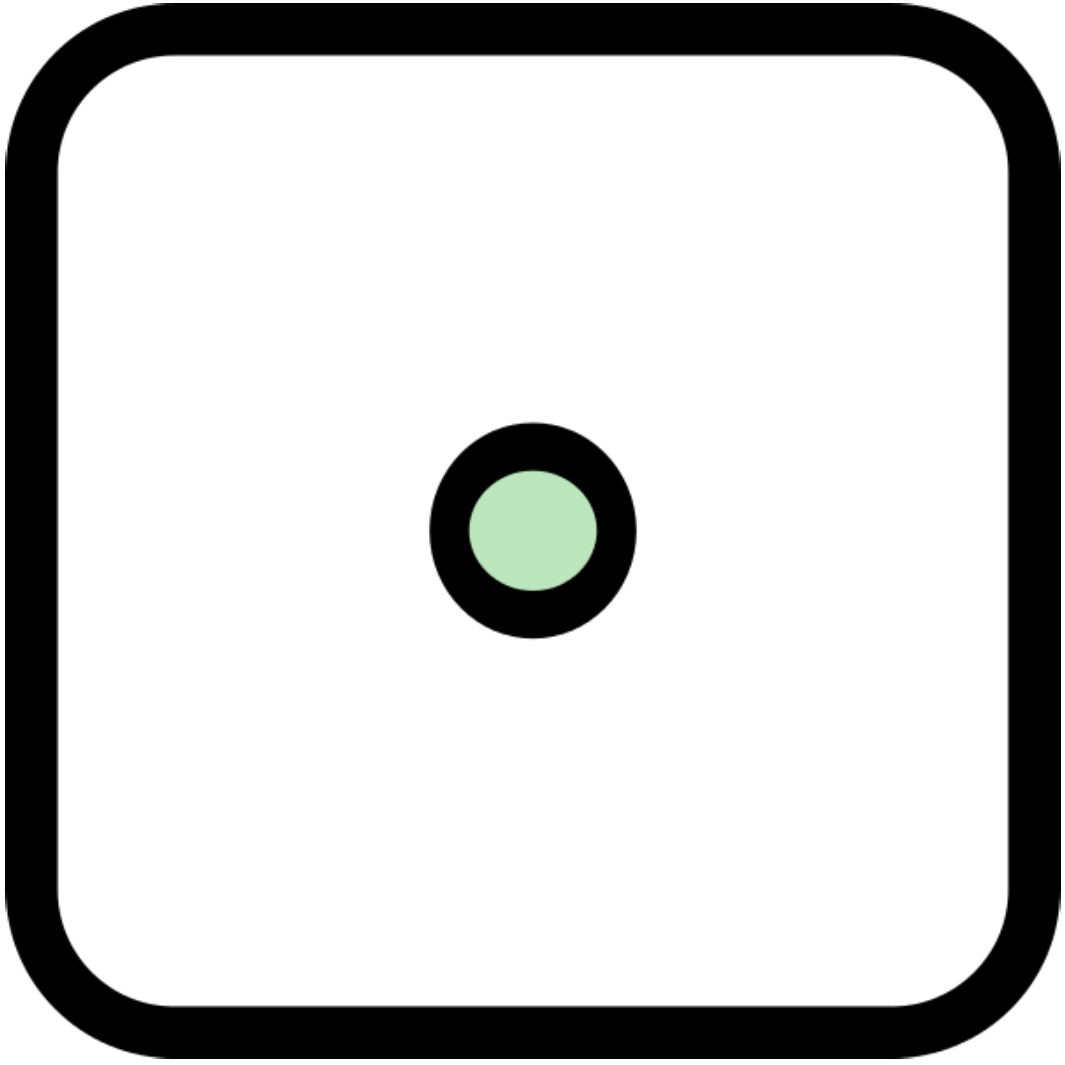
# OVERPASS TURBO

Interface pour construire des requêtes Overpass QL

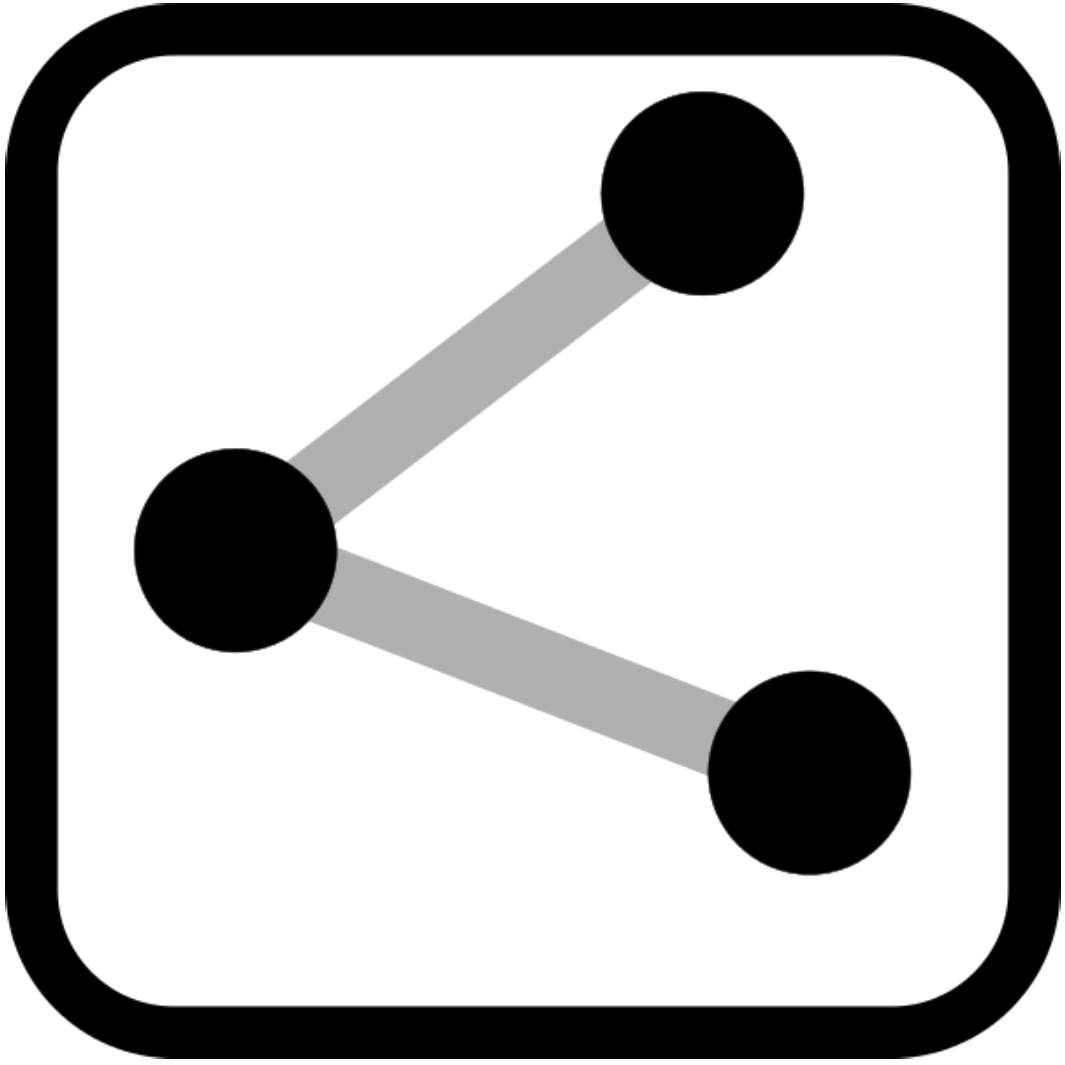
https://overpass-turbo.eu

# RAPPEL DU MODÈLE DE DONNÉES OSM

# AU COMMENCEMENT, IL Y A LE NŒUD

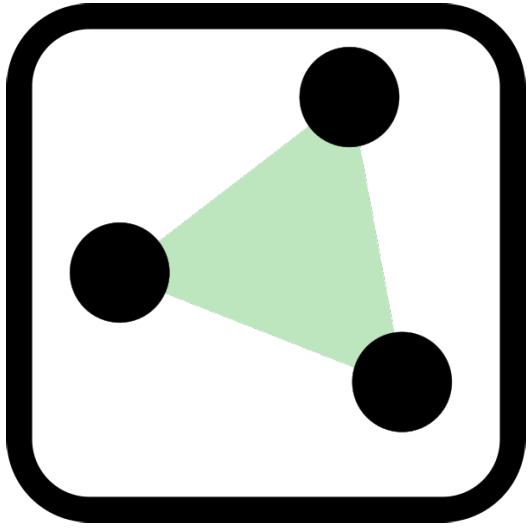


# DEUX NŒUDS FORMENT UN WAY



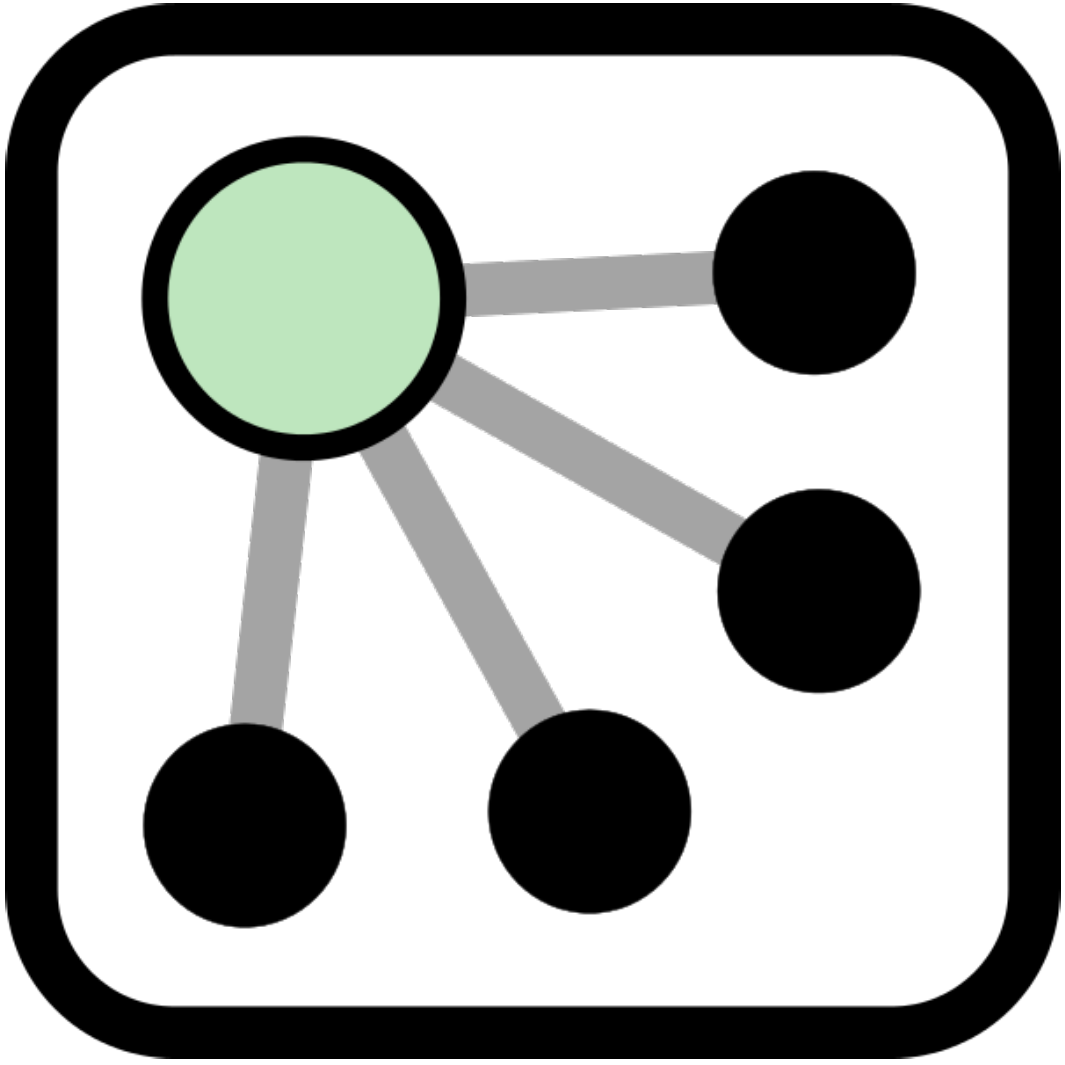
# UN WAY FERMÉ ? UN AREA

Si le premier et le dernier nœud d’un way sont identiques, le way est considéré comme un area (mais, dans le modèle, il s’agit d’un way).



(Coût en calcul important)

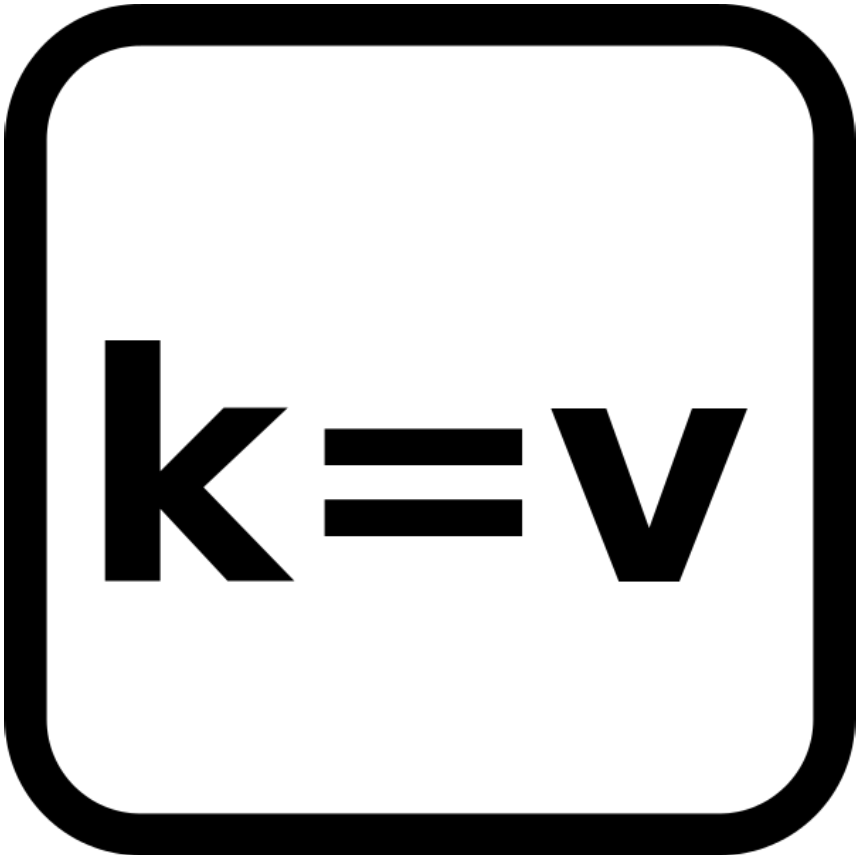
# DES OBJETS ENSEMBLE ? UNE RELATION



Les relations sont utilisées pour, par exemple:

- relier les routes dans un itinéraire:
  - ligne de bus, de chemin de fer, …
  - itinéraire balisé vélo, randonnée pédestre, …
- relier des équipements entre eux:
  - poteau d’arrêt de bus, plate-forme d’attente, point d’arrêt du bus;
- créer des objets complexes: multipolygones

# CHAQUE OBJET A DES ATTRIBUTS: CLÉ/VALEUR



# CHAQUE OBJET A UN IDENTIFIANT UNIQUE

- Les identifiants diffèrent pour les `nodes`, `way`, `relation`;
- Les identifiants sont modifiés en cas de:
  - suppression, et re-création;
  - division d’un objet en deux (par exemple un chemin).

# LES CLÉS/VALEURS SONT DOCUMENTÉES DANS LE WIKI

## La page wiki des restaurants:

# EXEMPLE: PARKING VÉLO

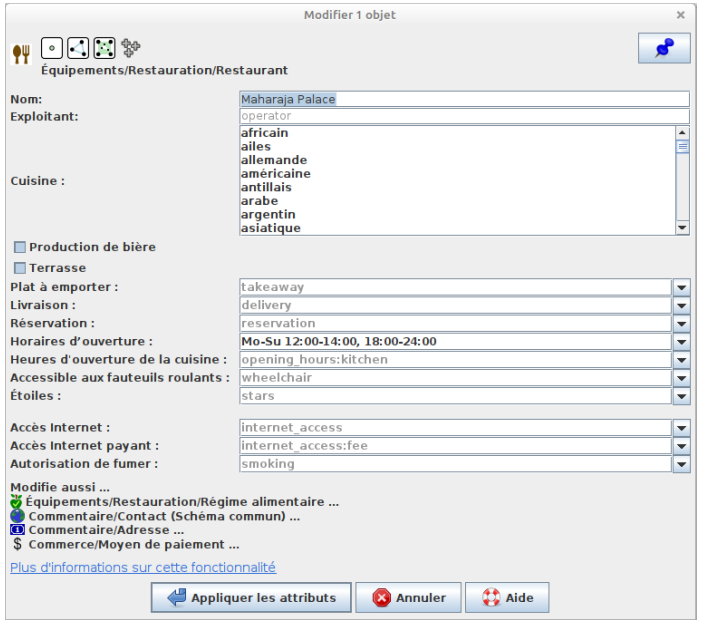
Parking vélo

```
amenity=bicycle_parking

bicycle_parking=stands

capacity=8

operator=Ville de Bruxelles

covered=no
```

# ÉDITION SIMPLIFIÉE DANS LES ÉDITEURS

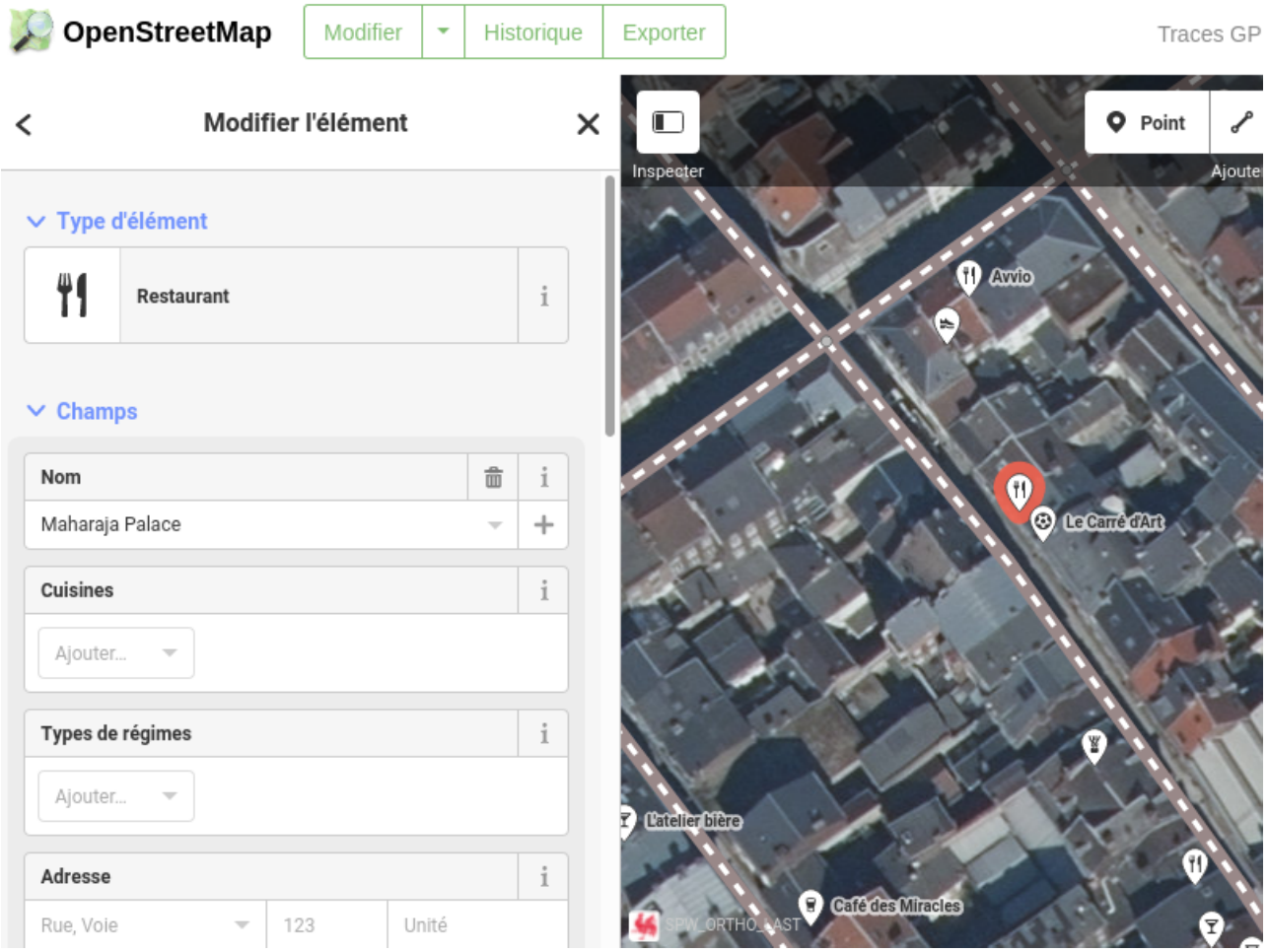
Les tags sont pré-sélectionnés dans les interfaces graphiques.

## JOSM



# ID

## Édition d'un restaurant à Liège



# REQUÊTES OVERPASS-QL

Quelques possibilités de requêtes.

# REQUÊTES PAR IDENTIFIANT UNIQUE

Relation “Université de Liège” (multipolygone):

```
[out:json][timeout:25];

rel(1376343);

out body geom;
```

Tester sur overpass-turbo

# Sélection par éléments du modèle:

```
//sélection d'un chemin
way(int);

// sélection d'un noeud
node(int) ;

//sélection d'une relation
rel(int) ;
```

# SÉLECTION PAR BBOX

BBOX = Boîte englobante, 4 coordonnées: sud, ouest, nord, est

```
node(44.0918, 6.2356790, 44.09231, 6.2362690);
out skel;
```

Outil pour générer des bbox: http://bboxfinder.com

# SÉLECTION PAR CLÉ/VALEUR

```
[out:json][timeout:900];
area["name"="Liège"]->.sa;

// la parenthèse crée une union des différents éléments
(
  // sélection des nœuds
  node["amenity"="restaurant"](area.sa);
  // sélection des chemins
  way["amenity"="restaurant"](area.sa);
)

out;
```

```
// sélection pour nodes and way
nw["amenity"="restaurant"];

// séleciton pour nodes et relations
nr["amenity"="restaurant"];

// sélection pour way et relations
wr["amenity"="restaurant"];

// sélection des way, relations et nodes
nwr["amenity"="restaurant"];
```

# EXEMPLES DE FILTRES

```
//équivalent
way[highway=pedestrian];
way["highway"="pedestrian"];

// chemins qui n'ont *pas* le tag pedestrian
way[highway!=pedestrian];
// (sélectionne aussi ceux qui ne sont pas des highway)

// sélectionne les chemins qui ont le tag highway
way[highway];

// sélection selon expression régulière
way[name~"^Rue"]; // tout ce qui commence par "Rue"
```

# Combiner des filtres:

```
// gather results
way["amenity"="bicycle_parking"]["bicycle_parking"="stands"]
(area.searchArea);
```

tester

# AREA: DES AIRES PRÉ-ENREGISTRÉES

Les zones administratives sont définies par les limites administratives (très souvent des `multipolygones` dans OSM):

```
// Province de Liège :-(
area["name"="Liège"]->.sa;

// peut être géocodée:
{{geocodeArea:"Liege"}}->.searchArea;
```

# CHERCHER AUTOUR D'UN POINT

Les barbecues proches d'une aire de pique-nique:

```
[out:json][timeout:250];
// fetch area "Province de Liège" to search in
{{geocodeArea:"Province de Liège"}}->.searchArea;

// rassembler aires de pique-nique dans une variable (.a)
(
  node["tourism"="picnic_site"](area.searchArea);
  way["tourism"="picnic_site"](area.searchArea);
)->.a;

// chercher les barbecues non-privés à moins de 500 mètres
(
  node(around.a:500)["amenity"="bbq"]["access"!="private"]
["access"!="no"];
  way(around.a:500)["amenity"="bbq"]["access"!="private"]
```

Tester

# RETROUVER TOUS LES OBJETS TOUCHÉS PAR UN USER

```
// Select nodes last modified by user
node(user:"juminet")({{bbox}});
```

tester

# SORTIE EN CSV

## Liste des musées en province de Liège:

```
[out:csv( name, fee, wheelchair, "addr:housenumber",
"addr:postcode", "addr:street", ::"lat", ::"lon" )];

{{geocodeArea:Province de Liège}}->.searchArea;

(
  // query part for: “tourism=museum”
  node["tourism"="museum"](area.searchArea);
  way["tourism"="museum"](area.searchArea);
  relation["tourism"="museum"](area.searchArea);
);

// sortie: les méta-données et le centre de chaque objet
out meta center;
```

Tester

# Sortie:

```
name    fee wheelchair  addr:housenumber    addr:postcode
addr:street @lat    @lon
Préhistosite de Ramioul                     50.5786075
5.4268272
Musee de Moresnet                       50.7213483  5.9888327
Archéoforum yes limited             50.6452983  5.5732626
Remember 39-45                      50.6688706  5.8840043
Musée de Fléron                     50.6109969  5.6797217
Charbonnage de Blegny-Mine                      50.6863350
5.7226648
Ikob -Internationales Kunstzentrum Ostbelgien           3   4700
Loten   50.6258969  6.0301564
Mö(h)renmuseum                      50.6918624  6.1057144
Töpfereimuseum Raeren                       50.6787456
6.1206606
```

# REQUÊTE AVEC DONNÉES DE STYLE

Uniquement avec Overpass-Turbo:

```
[out:json][timeout:25];
// gather results
(
  way["highway"](area:3081749435);
);
// print results
out body;
>;
out skel qt;


{{style:

way {
  text: surface;
}
```

Tester

```
[out:json][timeout:25];
// gather results
(
  // query part for: “maxspeed"
  way["maxspeed"="20"]({{bbox}});
  relation["maxspeed"="20"]({{bbox}});
  way["maxspeed"="30"]({{bbox}});
  relation["maxspeed"="30"]({{bbox}});
  way["maxspeed"="signals"]({{bbox}});
  relation["maxspeed"="signals"]({{bbox}});
  // query part for: “maxspeed=50”
  way["maxspeed"="50"]({{bbox}});
  relation["maxspeed"="50"]({{bbox}});
  way["maxspeed:forward"="50"]({{bbox}});
  relation["maxspeed:forward"="50"]({{bbox}});
```

# Tester

# REQUÊTE POUR COMPTER DES OBJETS

```
[out:csv(::count)][timeout:900];
area["name"="Liège"]->.sa;
nw["addr:housenumber"](area.sa);
out count;
```

Tester

# PARTAGE AVEC UMAP

Une requête Overpass peut servir de source de donnée sur une carte umap.

Exemple

Requête overpass originale

# WHERE IS THE DOC ?

https://wiki.openstreetmap.org/wiki/Overpass_API

# MERCI DE VOTRE ATTENTION !

Des questions ?

// reveal.js plugins