

Geotrek-admin

Groupement de développements

2026





Aperçu

- La Segmentation dynamique
 - Rappel & Fonctionnement actuel
 - Avantages & Limitations
 - Autres besoins
 - Évolutions futures
 - Travaux entrepris

- ⚡ Performances & Cartographie
 - Chantiers passés
 - Chantier actuel
 - Chantiers futurs ?

À propos

- Qui sommes-nous ?
 -  l'équipe chargée des développements de cette partie
 - Justine Fricou (*Développeuse*)
 - Jean-Etienne Castagnede (*Responsable technique*)



La segmentation dynamique





Rappel

- Réseau filaire :
 - *Géométrie de certains objets basée sur un réseau de tronçons.*
- Topologie stricte :
 - *Tronçons coupés obligatoirement à leurs intersections.*

Le fonctionnement actuel

À l'installation d'un Geotrek-admin, il faut choisir son mode :

-  Sans segmentation dynamique ↪ *Géométries classiques*
-  Avec segmentation dynamique ↪ *Géométries basées sur les tronçons*

Ainsi, un linéaire sera :

- Soit une *Ligne* (enchaînement de points -> **Sans segmentation dynamique**)
- Soit une *Topologie* (enchaînement de portions de tronçons -> **Avec segmentation dynamique**)

Les avantages de ce fonctionnement

Qualité et cohérence :

- Des données calées géographiquement sur un référentiel unique.

Liens automatiques :






- Connexion stockée en base entre les objets via les tronçons partagés.

Ses problématiques

Malgré ses avantages, les problématiques sont multiples :

- **Rigidité** : Supprimer un tronçon peut casser les linéaires passant dessus.
- **Blocages** : Impossibilité d'importer directement des linéaires (GPX / Shape).
- **Pertinence** : Liens automatiques entre objets manquant parfois de souplesse métier.
- Etc.

Et de nombreux autres besoins

-  **Cycle de vie** : Mettre à jour facilement son référentiel de tronçons.
-  **Multi-réseaux** : Avoir plusieurs référentiels distincts (vélo, pédestre, etc.).
-  **Stabilité** : Déplacer ou non les objets (POIs, etc.) quand un tronçon est modifié.
-  **Liens** : Repenser le système de connexions logiques entre les objets.
-  **Historisation** : Améliorer le suivi des modifications.
- Etc.



Fusion des modes "avec" et "sans" segmentation dynamique :

Un seul mode pour les gouverner tous...

Dans un même Geotrek-admin, les objets pourront être "couplés" aux tronçons, ou pas :

- ✨ **Liberté** dans les usages (avec ou sans tronçons)
- 🤝 **Cohérence** entre les instances (moins de différences d'architecture)
- 🛡️ **Sécurité** : Plus aucun linéaire "cassé" lors des modifications
- 🛠️ **Maintenabilité** : Une base de code unifiée et simplifiée

Étude des possibilités d'évolution

-  REX par Oslandia : Analyses des gestions du référencement linéaire chez différents acteurs de la géomatique.
-  Travail d'analyse dans Geotrek : - Documentation fine des mécanismes et des *triggers* SQL.
 - Analyse des multiples impacts du scénario de "découplage".

Qu'est-ce qu'une topologie "couplée" ?

C'est une topologie valide qui :

- A au moins une "**path aggregation**" (lien effectif à un tronçon).
- Est **valide**, c'est-à-dire que ses *path aggregations* :
 - Ne contiennent pas de doublons (sur la colonne ordre).
 - Sont positionnées dans le bon ordre chronologique.
 - Sont chacune orientées dans la bonne direction.
 - Forment une *LineString* continue une fois mises bout à bout.

Et une topologie "non couplée" ?

L'objet passe ou reste en mode "déconnecté" :

- S'il n'a **aucune** "path aggregation" associée en base de données.
- **OU** si sa topologie est devenue **invalide** suite à une modification topologique.





Exemple 1 : Suppression d'un tronçon

Un itinéraire qui passait sur ce tronçon détruit :

1.  **Découplage** : Est noté comme "*topologie invalide*" et "*découplé*" du réseau.
2.  **Préservation** : Conserve sa géométrie brute, affichée correctement sur rando/widget/mobile.
3.  **Résolution** : Peut être recouplé manuellement si besoin par l'administrateur.





Exemple 2 : Import de linéaires

Lors de l'import d'un nouveau tracé linéaire :

-  **Stockage** : Sa géométrie brute est enregistrée en base de données.
-  **Autonomie** : Il est marqué comme n'étant "*pas couplé*" au réseau de tronçons.
-  **Intégrité** : Il n'est pas considéré comme invalide pour autant.
-  **Évolution** : Il pourra être couplé manuellement au réseau si besoin.

 Objectifs de la feuille de route

Objectifs à court terme (Groupement de commandes)

1.  **Hybridation** : Mise en place technique du couplage/découplage.
2.  **Filtrage** : Modification et amélioration du filtre "topologie invalide".
3.  **Alignement** : Possibilité de recoupler manuellement une géométrie.
4.  **Sécurité** : Création d'une nouvelle permission fine : *"Peut tracer hors tronçons"*.



Objectif à moyen terme



Une interface de recouplage assisté :

- **Mode miroir** : Affichage de la géométrie attendue ("correcte").
- **Comparaison** : Superposition avec la géométrie de la topologie devenue invalide.
- **Guidage** : Navigation pas-à-pas entre les points de divergence géométrique.



Objectif long terme



Couplage automatisé ou semi-automatisé :

- **Algorithme** : Détermination automatique d'une topologie théorique à partir de la géométrie de la ligne.
- **Validation** : L'utilisateur vérifie et complète la topologie calculée.
- **Cas d'usages principaux** : Mises à jour massives du référentiel de tronçons et imports de linéaires à la volée.



Évolutions du système de rattachement des objets



Fonctionnement actuel : Avec segmentation dynamique

- **Principe** : Un itinéraire est basé sur des tronçons physiques.
- **Lien** : Des objets (POIs, Services...) sont également rattachés à ces tronçons.
- **Résultat** : Les objets remontant sur la fiche itinéraire sont strictement ceux liés aux mêmes tronçons.

Fonctionnement actuel : Sans segmentation dynamique

- **Principe** : Un itinéraire n'est pas basé sur des tronçons mais sur une ligne indépendante.
- **Lien** : Une zone tampon spatiale (*buffer*) est calculée autour de l'itinéraire.
- **Résultat** : Les objets qui remontent sur la fiche sont ceux situés géographiquement dans cette zone tampon.

Solution à court terme (Groupement de commandes)

Objectif : Aller au plus simple en s'adaptant dynamiquement à la nature de la géométrie :

- **Objets couplés au réseau** ↪ Rattachement précis par l'identifiant des tronçons.
- **Objets non couplés** ↪ Rattachement spatial automatique via zone tampon (*buffer*).

Solution à long terme

Des réflexions globales sont en cours pour repenser entièrement ce système :

-  Instaurer des règles générales fines via des paramètres globaux.
-  Permettre une gestion chirurgicale au cas par cas (inclusion ou exclusion manuelle forcée).

⚡ Améliorations des performances cartographiques





Rappel des précédents travaux

- Optimisation des requêtes SQL
- Mise en cache partielle
- Projection interne vers WGS84 via PostGIS
- Génération des GeoJSON en 1x
- Compression GZip coté Nginx
- Pagination des listes

Travaux actuels

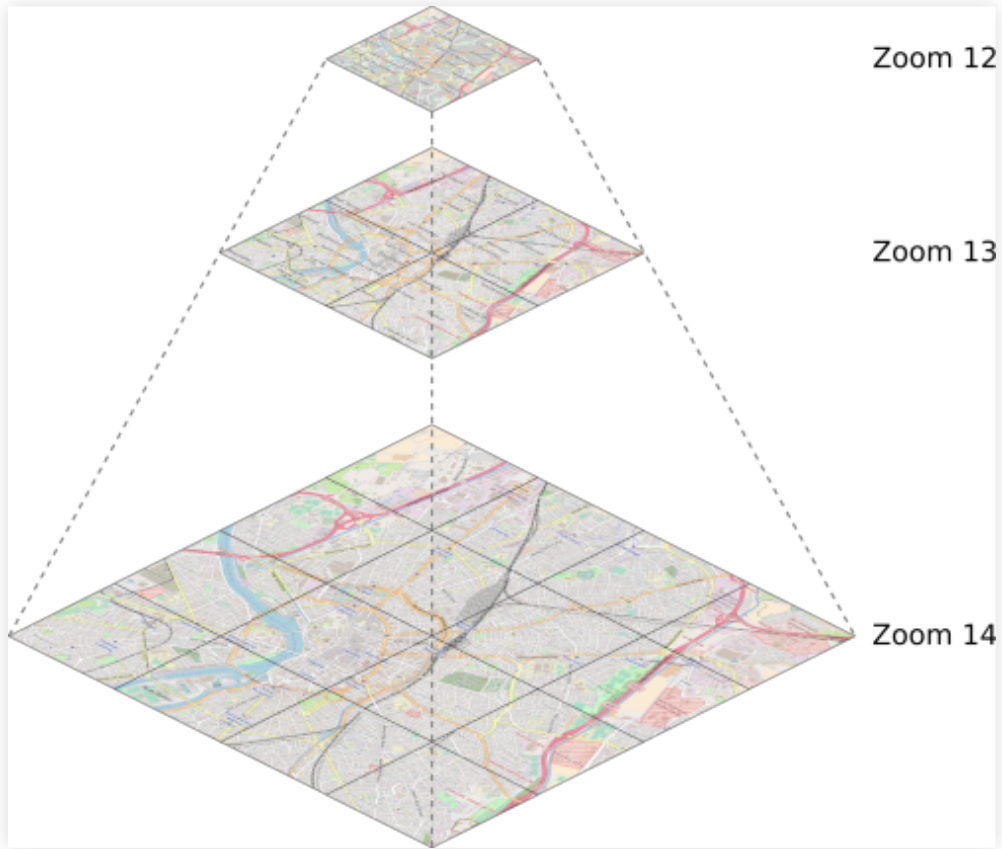
La problématique des grands réseaux

Sur les instances contenant plusieurs dizaines de milliers de tronçons :

-  **Lenteur réseau** : Un seul fichier GeoJSON très lourd contenant l'ensemble des objets génère un long temps de téléchargement.
-  **Surcharge client** : Le navigateur charge et calcule tout en mémoire (même les éléments hors écran), ralentissant l'affichage.



La solution : L'utilisation de tuiles vectorielles (MVT)

- 🗺️ **Grille spatiale** : Les données géographiques sont découpées et segmentées en tuiles standardisées.
- 📦 **Flux optimisé** : Seules les tuiles visibles à l'écran (selon la zone et le niveau de zoom) sont requêtées et chargées.
- 🚀 **Performance** : Allègement drastique de la mémoire requise et affichage instantané.



Modernisation : Passage de Leaflet à MapLibre GL

Le passage aux tuiles vectorielles impose de moderniser le moteur de rendu :

-  **Rendu matériel** : Transition nécessaire pour exploiter la puissance des tuiles vectorielles via le GPU du client.
-  **Capitalisation** : Intégration des résultats concluants du stage de Pascia qui a défriché et validé le sujet.

Mapbox-gl-path & Ergonomie de tracé

Le changement de moteur implique la réécriture complète de l'interface de dessin :

- **Nouveau composant** : Utilisation d'un plugin moderne développé par Makina Corpus adapté à MapLibre.
- **Ergonomie** : Tracé rendu plus fluide et gestion native facilitée des boucles de randonnée.
- **Perspectives fonctionnelles à terme** :
 - Créer des itinéraires semi-couplés (moitié sur tronçons, moitié hors piste).
 - Choisir dynamiquement le référentiel d'accroche (réseau vélo, réseau piéton, etc.).

Demo

Des questions ?

Merci !

