

# Atelier technique

Groupement de commande – Lot 1

21 et 22 janvier 2020



# Planning

- **Journée 1 – mardi 21/01/20**

- 9h : lancement des deux journées
- 9h30 → 17h30 : atelier technique « Architecture et performances »
- 17h30 : bilan journée 1

- **Journée 2 – mercredi 22/01/20**

- 9h : atelier technique « MCD Outdoor »
- 14h : finalisation « Architecture et performances » + organisation et suite à donner projet (issues Github, priorités, sprints de développement...)
- 17h30 : bilan journée 2

- *Déjeuners prévus chez Makina Corpus*



# Sommaire

- Maintenance
- Install/conf/MàJ
- Volumétrie/perfs
- Seg° dynamique
- Modélisation
- Partage de données
- Améliorations



# Maintenance



# Dépendances

- Sécurité / Bugfix / Nouvelles fonctionnalités
- Liens avec l'OS
  - Python/Postgresql/Postgis/GDAL
- Inter-dépendances
  - Si on tire un fil on tire la pelote



# Python

- Python 2.7 est mort le 31/12/2019
- Passage en Python 3.5+ le 26/11/2019
- Plusieurs mois de travail (stage de Tim)
- Passer en Python 3.6+ pour les f-strings ?  
[0~]
  - Xenial=3.5    Bionic=3.6    Focal=3.7



# Django

- Django 1.11 LTS (actuelle) → mars 2020
- Django 2.0 (MR en cours) → ~~mars 2019~~
- Django 2.1 → ~~novembre 2019~~ [5j-]
- Django 2.2 LTS → mars 2022 [5j-]
- Puis versions mineures tous les 8 mois
- Ou versions majeures (LTS) tous les 2 ans



# Ubuntu

- Support actuel :
  - 16.04 Xenial
  - 18.04 Bionic
- En avril 2020
  - Ajouter le support de 20.04 Focal [5j-]





# Leaflet

- Passage à MapboxGL ? [∞]
- Passage de 0.7 à 1.6 [5j+ x 2]
  - Une dizaine de dépendances pas forcément maintenues
  - Quelques milliers de lignes js potentiellement impactées
  - Packaging à (re)faire



# Leaflet

- Leaflet.Label → remplacé par L.Tooltip (intégré à Leaflet)
- Leaflet.Spin → ?
- Leaflet.Snap → OK ? Porté pour Leaflet 1.x mais semble pas mal bugé
- Leaflet.draw → OK mais avec breaking changes, passage à Leaflet.editable ?
- Leaflet.MeasureControl → Non mais PR à valider
- Leaflet.OverIntent → Non
- Leaflet.LayerIndex → Non, utilisé ?
- Leaflet.FileLayer → OK
- Leaflet.groupedlayercontrol → ?, fork d'un projet non maintenu
- Leaflet.GeometryUtil → ?
- mapentity/leaflet-datatables
- mapentity/leaflet.enumeration
- mapentity/leaflet.fullscreen
- mapentity/leaflet.informationcontrol
- mapentity/leaflet-objectslayer
- mapentity/leaflet-serializers
- geotrek/.../\*.js



# Autres libs JS

- A faire d'un seul bloc (interdépendances)  
[2j+]
  - jQuery 1.7 → 3.4
  - Bootstrap 2.3 → 4.4
  - Datatables 1.9 → 1.10



# Mapentity

- Complique le développement
  - Où est le code ? Dans Mapentity ou dans Geotrek ?
  - Doubles releases
- Jamais utilisé pour un autre projet
- Découpage à posteriori, le couplage reste fort
- Fusionner Mapentity dans Geotrek [1j]



# Tests automatisés

- Geotrek-admin couvert à 95 %
  - Avec et sans la segmentation dynamique
  - Avec et sans Docker
- Mapentity couvert à 85 % [5j]
  - Passer à CircleCI et Codecov
  - Augmenter la couverture
- Autres dépendances + ou - testées



# Installation, configuration, mise à jour



# Script install.sh

- Fait le job mais...
- Mauvaise gestion des erreurs
- Peut casser une installation qui marchait quand on veut changer un setting
- Éditeur de texte pour les options de base
- Limité à Ubuntu
- Mise à jour pas si simple



# Améliorer install.sh [5j]

- Utiliser invoke
- Télécharger automatiquement le code
- Prendre en compte la mise à jour
- Séparer installation, mise à jour, et changement de conf





# Docker [2j]

- Ne dépend plus de multiples téléchargements externes
  - Plus fiable et plus rapide
- Marche sur toutes les distributions
- Intégrer postgresql et/ou nginx ?
- Script d'install ?



# Packaging [5j]

- Debian/Ubuntu et/ou Redhat/Centos
- Simple apt install
- Mise à jour automatique

*Les différentes solutions ne sont pas exclusives*



# Configuration web [5j]

- Proposer un maximum d'options de configuration dans l'interface web
- Evaluer et choisir parmi <https://djangopackages.org/grids/g/live-setting/>



# Modularité

- Rendre plus simple l'activation/désactivation de modules ?
  - Question des dépendances
- Modules contribués ?
- Ajout de champs personnalisés ?



# Volumétrie et performances



# Points critiques

- Temps d'affichage des pages du CRUD
- Temps d'enregistrement des longs itinéraires
- Temps de synchronisation
- Temps d'import
- Temps d'export



# Affichage des pages du CRUD

- Un seul fichier pour l'emprise entière
  - C'était OK pour la spec initiale (PNE) mais ça ne passe pas à l'échelle
- Toute la pile est impacté :
  - Requête de la base → Sérialisation geoJSON → Transfert réseau → Désérialisation → Création du DOM



# Affichage des pages du CRUD

- Tuiler les géométries [10j]
  - Pas de support natif dans Leaflet mais des gist existent
- Simplifier les géométries [5j]
- Paginer les listes [5j]
- Mieux gérer le cache (invalidations ciblées) [5j]
- Passer à DRF





# Affichage des pages du CRUD

- Impact sur le routage ?
  - A déplacer côté back avec pgRouting ?
- Supprimer l'interaction carte / liste ?
  - Ne fonctionne pas avec la pagination ou le clustering
  - Proposer 2 vues plutôt qu'une vue partagée ?



# Enregistrement des longs itinéraires

- Temps de calcul de l'altimétrie
- Solutions multiples (à tester et évaluer) [5j]
  - Optimiser le SQL
  - Agréger l'altimétrie des tronçons
  - Calculer en C/Python (numpy ?)
  - Rendre le calcul asynchrone



# Synchronisation

- Améliorer la gestion du cache [2j]
  - PDF générés systématiquement même en l'absence de changements
  - Mais les changements sont compliqués à détecter
- Optimiser l'API [2j]
- Paralléliser [2j]



# Export

- Timeouts fréquents
  - Durée > 5 min, voire beaucoup plus
- Optimiser la génération [?]
- Génération asynchrone [3j]



# Import (parsers)

- Meilleure gestion des images
  - Éviter de télécharger la pleine résolution
  - Parallélisation



# Réactivité générale

- Découplage back/front [∞]
- Utilisation d'un framework JS
  - Angular / React / VueJS / ... ?



# Emprise mondiale [∞]

- Stockage en WGS84
- Projection locale à la volée pour faire les calculs



# Segmentation dynamique





# Réseau topologique

- □ La modélisation n'est pas topologique !
- Le système de découpage des tronçons est
  - Un contournement plutôt qu'une vraie solution
  - Fragile (bugs dans les triggers en cascade)
  - Empêche les croisements dénivelés
- Difficultés à maintenir la connectivité



# Topologies postgis [?]

- Node / edge / face / relation
- Cf. [Paris2011\\_TopologyWithPostGIS\\_2\\_0.pdf](#)
- Nécessite de revoir l'API et l'UX pour l'édition
- Non compatible avec l'ORM Django
- La matrice de routage est plus directe



# Segmentation dynamique

- Les modifications de réseau ont fortement tendance à casser les itinéraires
- Que faire des pk lors d'une modification de tronçon ?
- Pour les associations ponctuels/linéaire, ne se comporte pas mieux qu'un buffer
- Ne gère que les liens avec ou via le réseau



# Découplage

- La géométrie redeviendrait la source primaire
- Maintien d'une table de liaison plus générique [10j]
  - Cache des différents prédicats postgis
  - Quels que soient les types
  - Topologies postgis
  - Auto/manuel + alertes (type osmose)



# Découplage

- 2 modes avec/sans routage comme sur cirkwi
- Réintégration des partie hors réseau dans le réseau ?
- Proposition de recalculer sur le réseau existant ?



# Caractérisation du réseau

- Physique et foncier
- Sans trous ni recouvrements
- Découpage automatique sur le cadastre
- Supprimer les statuts de gestions ?
- Supprimer les attributs sur les tronçons ?



# Modélisation



# Suppression

- Supprimer le NoDeleteMixin ? [1j]
- Revoir les on\_delete [1j]
- Ajouter un mécanisme de rollback ? [5j]
  - Django-reversion ?
  - Quid des FK ayant disparu entre temps ?





# Nommage

- Mettre les noms des champs en anglais dans la base comme dans le code
- Supprimer les schemas
  - Pas supporté par Django
  - Cause des soucis lors des mises à jour (masquage)
  - Il n'est pas interdit d'en créer manuellement



# Edition sous QGIS

- Très grosse contrainte sur le code
  - Fonctionnalités, bugs, temps de dev
- Est-ce réellement nécessaire ?
  - Que manque-t-il à l'UX de Geotrek pour remplacer QGIS ?
- Par contre, la lecture ne pose pas de problème
  - Possibilité de créer des vues



# Partage de données



# API

- Statique vs. Dynamique
  - Performances
  - Sécurité (serveur privé  $\neq$  serveur public)
- Dédiée ou générique
- GraphQL ?



# Échanges entre Geotrek

- OK si maître/esclave
  - Gérer les conflits d'édition est trop compliqué
- Quid de l'intégrité référentielle ?
- Instance unique
  - Multi-portails
  - Multi-structures



# Imports

- Automatiser les imports OSM
  - Du réseau
  - Des zonages (collectivités, cadastre, ...)
- Recalage automatique des traces



# Améliorations



# Améliorations

- Module aménagements
- Module gestion
- Représentation 3D
- Export PDF





# Tableaux de bord

- Intégré / Module / Externe
- Objectifs ?



# UX

- ...



# Outdoor

- Présentation générale
- MCD
  - Aire / site
  - Rando ∈ APN
- UX

