

# Atelier des 21 et 22 janvier 2020

# **Présents**

- Camille MONCHICOURT (PN Écrins),
- Lise LE LANN (PN Écrins),
- Amandine SAHL (PN Cévennes),
- Michaël VIADERE (OpenIG),
- Timothée DE MONTETY (Makina Corpus),
- Jean-Etienne CASTAGNEDE (Makina Corpus),
- Emmanuelle HELLY (Makina Corpus), partiellement,
- Bastien POTIRON (Makina Corpus), partiellement,
- Gaël UTARD (Makina Corpus),
- Anne MONTEILS (Makina Corpus), pour l'introduction et la conclusion

# Jour 1 - Discussion ouverte sur les différents points techniques

Support pour guider les débats : Fichier « MakinaCorpus atelier technique.pdf »

# Maintenance générale

- Il est nécessaire de mettre à jour les dépendances pour corriger les failles de sécurité, corriger les bugs, et profiter de nouvelles fonctionnalités.
- La mise à jour des distributions (et donc de Python/PostgreSQL/PostGIS/GDAL) force également la mise à jour des dépendances.
- Les dépendances ont des dépendances entres elles. Si on tire un fil on tire la pelote.
- Le passage à Python 3 a été fait in extremis (le 26/11/19 pour une deadline au 31/12/19) sur fonds propres par Makina Corpus et a nécessité plusieurs mois de travail (stage de Timothée).
- La version de Django actuellement utilisée par Geotrek (1.11) est une LTS mais qui prend fin en mars 2020.
- Les versions 2.0 et 2.1 sont non LTS et ne sont déjà plus supportées.
- Une MR est en cours pour passer à la 2.0 mais il faut passer directement à la versions 2.2 LTS rapidement.
  - Consensus sur la nécessité de mettre à jour les dépendances
  - Accord sur l'idée de supporter les 2 dernières versions d'Ubuntu LTS ce qui nécessitera un petit peu de travail en avril 2020
  - Accord pour consacrer ~5jh au passage en Django 2.2 LTS

# Leaflet vs MapboxGL

- Il y a débat sur la pertinence de garder Leaflet ou de passer à MapboxGL.
- Camille s'inquiète du côté « propriétaire » de MapboxGL qui semble porté uniquement par une société commerciale. Mais il faut différencier la partie génération de tuiles proposée par

Mapbox qui est effectivement propriétaire (mais des alternatives libre existent) et la partie affichage qui est sous licence libre, au même titre que Leaflet qui a également été promu par la même société commerciale.

- La problématique de performance de chargement des pages peut se régler en tuilant les géométries. Cela peut se faire avec du GeoJSON et Leaflet et cela n'impose pas le recours aux tuiles MVT et à MapboxGL.
- On peut s'attendre à ce que MapboxGL soit plus performant que Leaflet, mais si on tuile avec Leaflet cette différence sera minime (moins de x 2), alors que la différence entre Leaflet tuilé et Leaflet non tuilé sera énorme (x 100 ou plus).
- Le tuilage sous Leaflet nécessite probablement moins d'un sprint (20jh). Il faudrait passer 1jh pour étudier les implications et découper différentes tâches pour avoir plus de visibilité. Il sera aussi nécessaire de mettre à jour les différents plugins Leaflet utilisés de Leaflet 0.7 à 1.6 ce qui pourrait nécessiter environ 5jh.
- Le passage à Mapbox est beaucoup plus incertain car il faudra ré-écrire une majeure partie du code javascript de Geotrek et refaire tous les plugins. Or, si Makina Corpus a une bonne expertise dans l'utilisation de MapboxGL, elle n'en a pas dans le développement de plugins.
  - Conserver Leaflet semble donc la meilleure solution pour l'instant
  - Par contre, dans le cadre d'une réécriture complète de Geotrek, MapboxGL s'imposerait certainement

### **Django-Mapentity**

- La séparation Mapentity/Geotrek complique le développement
- La promesse était de pouvoir utiliser Mapentity dans d'autres projets mais cela ne s'est jamais fait
- On a donc les inconvénients sans les avantages
  - Camille comprend notre point de vue mais ne semble pas pouvoir se résoudre à abandonner l'idée de disposer à travers Geotrek d'un WebSIG
  - Mener la logique Mapentity à terme nécessiterait des moyens financiers supplémentaires, et c'est plutôt Terralego qui répond à cette problématique aujourd'hui.
  - Accord sur l'idée d'intégrer Mapentity dans Geotrek-admin

#### Tests automatisés

- Très bonne couverture de test backend (95%)
- CI fonctionnelle
- Manque sur la partie front/intégration
  - Pas de tâche particulière sur les tests mais faire le nécessaire sur chacune des tâches pour conserver la couverture de test

#### **Installation**

• Le script bash (install.sh) actuel fait le job mais pourrait être amélioré sur de nombreux points

- On peut aussi imaginer utiliser Docker ou faire des paquets Debian
- Configuration dans l'interface web
  - Consacrer 5 jh à améliorer le script, en particulier pour séparer les cibles installation, mise à jour, ou simple reconfiguration (dans ce dernier cas la procédure n'est pas claire).
  - OK pour que Makina utilise Docker mais ça ne doit pas devenir la seule méthode publique officielle d'installation. Il faut conserver le script d'installation.
  - Non pour le packaging Debian
  - Non pour le support d'autres distributions qu'Ubuntu.
  - Configuration dans l'interface web non prioritaire

#### Modularité

- Rendre l'activation et la désactivation des modules plus facile.
- OK mais l'intérêt de Geotrek c'est l'aspect métier qui interconnecterles modules, le code est donc interdépendant.
- Permettre d'ajouter ou supprimer des champs
  - On peut se contenter de masquer un module mais c'est loin d'être satisfaisant si les tables restent en base.
  - Il faudrait proposer un système de modules optionnels dont le code n'est pas dans le dépôt principal (ex. module plongée demandé par le CD30)
  - L'ajout de champs personnalisés (stockés en json) serait une fonctionnalité intéressante

# Temps enregistrement itinéraire

- Il y a un timeout sur l'enregistrement des longs itinéraires (>50 km).
- C'est lié aux calculs d'altimétrie.
- Des solutions ont été proposées sur Github mais jamais mises en œuvre.
- C'est ce souci qu'il faut régler en priorité car c'est plus qu'un problème de performance/volumétrie, c'est un bug bloquant.
  - A régler en priorité en commençant par regarder si on peut optimiser le SQL (supprimer une éventuelle double boucle, récupérer tous Z dans une requête SQL unique plutôt que d'itérer)
  - Vérifier si la performances dépend du pas du MNT. Il y a débat entre les devs sur ce point.

#### Volumétrie

- Les temps d'affichage des pages du CRUD deviennent très longs, jusqu'à rendre l'affichage impossible lorsqu'on augmente le nombre de tronçons.
- La limite actuelle pour un fonctionnement fluide est de l'ordre du millier de tronçons
- La limite actuelle pour afficher les pages est de l'ordre de la dizaine de milliers de tronçons (après une attente de plusieurs minutes !)

- La raison en est que l'ensemble des géométries sont requêtées en base, sérialisées, transférées sur le réseau, désérialisées et transformées en DOM même lorsque l'on zoome sur une petite zone.
- Il faut donc tuiler les géométries pour que les performances ne soient plus dépendantes de la volumétrie.
- Il faut également paginer la liste de objets, simplifier les géométriques sur les faibles zooms, améliorer la gestion de caches
  - OK pour passer 1j sur le 1er sprint à détailler les impacts, découpe en soustâches et chiffrer les sous-tâches
  - On ne touche pas au routage côté front pour l'instant. Le graphe de routage pose forcément un problème de volumétrie, mais cela pourra être réglé dans un second temps si nécessaire.
  - OK pour supprimer l'interaction carte->liste. Par contre il faut conserver l'interaction liste->carte

### Temps de synchronisation

- Améliorer la gestion du cache. Il n'est pas nécessaire de re-générer systématiquement les PDF ou le DEM si rien n'a changé
- Optimiser l'API (éviter les requêtes SQL à chaque objet)
- Paralléliser
  - Mettre en cache les PDF et le DEM
  - Invalider ce cache globalement lorsqu'il y a une modif, par exemple sur le nom d'un niveau de difficulté
  - Non prioritaire tant que l'on a pas défini l'API avec le nouveau Geotrek-rando codé par BAM
  - Préférence de Camille pour une API dynamique (les clients Geotrek-rando iraient requêter directement les vues Django plutôt que des fichiers statiques mis à jour 1 fois/24h)
  - Il manque la possibilité de différencier plusieurs templates de PDF et plusieurs meta pour les différents portails d'un même Geotrek-admin

# **Export CSV/Shapefile**

- Souvent très long et timeout
- Possibilité de faire générer directement le fichier par PostgreSQL ?
  - Pas de retours utilisateur. Ne semble pas poser problème. Probablement parce que les uns ne font aucun export et les autres le font directement depuis la base SQL

# **Imports (parsers)**

- Souvent très long (plus de 24h pour Champagne Ardennes!)
- Gros téléchargements et stockage de photos alors qu'il serait possible de ne garder que le lien
  - Il manque le multilingue

# Découplage backend/frontend

- Création d'une SPA en utilisant un framework JS
- Meilleure réactivité, aspect plus moderne
- Coût difficile à évaluer
- Serait le choix évident si on partait d'une page blanche, mais aucun problème mis en avant actuellement ne justifie ce changement
  - La question n'est pas forcément de faire ce changement sur le Geotrek actuel mais plutôt de savoir si on ne reprend pas tout de zéro pour profiter entre autres des dernières technologies telles que les SPA.

# Réseau topologique

- En réalité le stockage n'est pas topologique
- Du coup, quand on modifie le réseau, on casse généralement des connexions entre les tronçons alors que justement le but des topologies c'est que le modèle garantit la connexion entre les tronçons via les nœuds
- Le système de trigger pour couper les tronçons est assez fragile (bugs du aux triggers en cascade qui sont mal maîtrisés)
- Il n'est pas possible de représenter un croisement dénivelé (pont)
- PostGIS propose un module topology qu'on pourrait utiliser mais le choix contraire avait été fait au démarrage de Geotrek
  - Camille n'est pas chaud pour se lancer dans ce changement qui lui semble très incertain
  - Selon l'avancement du projet, une étude pourra être conduite sur le passage à un vrai modèle topologique

# 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/linéaires, ne se comporte pas mieux qu'un buffer
- Ne gère que les liens avec ou via le réseau, pas les liens entre ponctuels par exemple
- Proposition de découpler le réseau des itinéraires
- Avec maintien d'une table de liaison entre les couches
- Le système pour alimenter et maintenir cette table serait souple et configurable, manuel et/ou automatique
- Un système d'alerte aiderait à maintenir la cohérence entre le réseau et les itinéraires en signalant par exemple quelles sont les portions d'itinéraire hors réseau
- Une forme de POC a été implémenté sur Géocompostelle
  - La remise en cause de la segmentation dynamique fait consensus
  - L'idée du découplage semble intéresser Camille et Amandine

- Ça simplifierait aussi la saisie mobile hors connexion, ou les échanges de données entre Geotrek-admin, ou le réimport du référentiel de tronçons qui est impossible actuellement
- Mais ils ont besoin d'y voir plus clair
- OK pour passer 2jh de réflexion lors du 1er sprint pour formaliser une proposition

#### **NoDeleteMixin**

- Supprimer NoDeleteMixin qui permet de garder en base les objets supprimés pour une éventuelle récupération (champs booléen « supprime »)
- Remplacer par un mécanisme de rollback plus performant tel que Django-reversion ?
- Revoir le choix des on\_delete (cascade, set null ?)
  - OK pour supprimer NoDeleteMixin sans remplacement par autre chose
  - Pas besoin de revoir les on\_delete. La situation actuelle n'est pas si problématique.
  - Supprimer les medias automatiquement

### Nommage

- Passer les noms des champs en anglais dans la base comme dans le code
- Supprimer les schemas PostgreSQL
  - OK pour renommer les champs et les tables
  - OK pour supprimer les schemas multiples mais pas OK pour tout mettre dans le schéma public. Il faut séparer les fonctions PostGIS (dans public) des fonctions Geotrek (dans un schema geotrek)

# **Edition sous QGIS**

- Crée des contraintes sur le code
- Est-ce réellement nécessaire ?
- Par contre, la création de vues en lecture seule ne pose pas de problème
  - La modification du réseau directement en SQL permet de faire des imports et de sélectionner les tronçons visibles ou non dans Geotrek
  - Supprimer cette possibilité ne peut donc se faire que si Geotrek propose des outils d'import adaptés
  - Les sigistes (Amandine) préfèrent bidouiller en SQL que d'appeler des scripts python

#### API

- Statique vs. Dynamique
- Dédiée ou générique
- GraphQL?
  - Camille préfère une API dynamique (sans synchro)

- Il sera cependant nécessaire d'évaluer les conséquences en matière de performances
- Le sujet est temporairement mis de côté en attendant les besoins de BAM pour Geotrek-rando

# Jour 2 - Outdoor

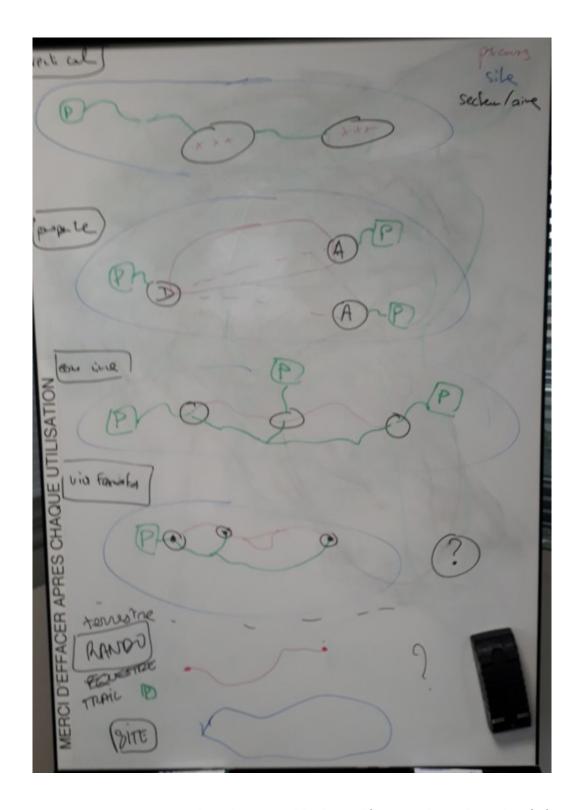
#### **Présentation**

En tant que <u>chargée de mission</u> (<u>https://geotrek.ecrins-parcnational.fr/ressources/outils/2019-06-FP-CDM-Geotrek-Outdoor-Data.pdf</u>)</u>, Lise commence par présenter le sujet. Celui-ci s'inscrit à la fois dans le cadre d'un projet européen au niveau du PNE, et dans le cadre des missions des départements (PDESI = Plan Départementaux des Espaces, Sites et Itinéraires). Il peut également intéresser les fédérations sportives telles que la <u>FFME</u> (<u>https://www.ffme.fr/</u>). L'objectif est de recenser, gérer et valoriser les sites d'activité de pleine nature. Un groupe de travail a été constitué et a produit <u>des documents</u> (<u>https://geotrek.ecrins-parcnational.fr/ressources/gt/07-geotrek-apn/</u>)

#### **MCD**

Certains souhaitent travailler à un niveau très général (sites). D'autres souhaitent inventorier par ex. chaque longueur de chaque voie de chaque secteur de chaque site d'escalade. Un MCD (https://geotrek.ecrins-parcnational.fr/ressources/gt/07-geotrek-apn/2019-08-mcd-outdoor-v6.png) proposé par le groupe de travail va plutôt dans le deuxième sens. Camille Monchicourt semble critique vis à vis de ce MCD et défend une approche qui soit générique.

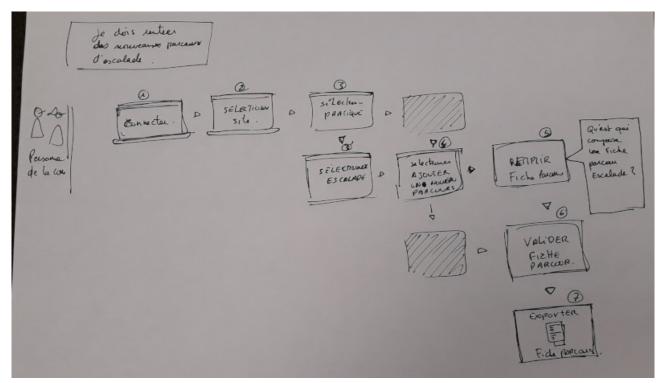
Nous avons donc débattu, en essayant de représenter les différents concepts de chaque activité et de voir ce qui est commun et ce qui est différent :

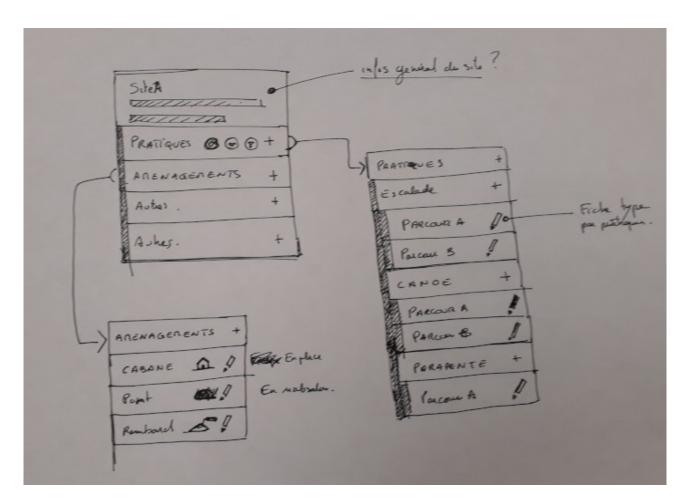


Nous n'avons pas vraiment pu trancher. Il est possible de modéliser quelque chose de générique avec les notions de site et de parcours. Mais cela ne permettra pas de satisfaire certains utilisateurs qui souhaitent aller dans le détail. Inversement une modélisation trop détaillée ne correspondra pas aux autres utilisateurs.

Le groupement de commande convient qu'il doit retravailler sur le sujet et que c'est à lui de trancher sur le niveau de détails attendu.

 ${f UX}$  Mathieu a proposé un parcours utilisateur et différents écrans :

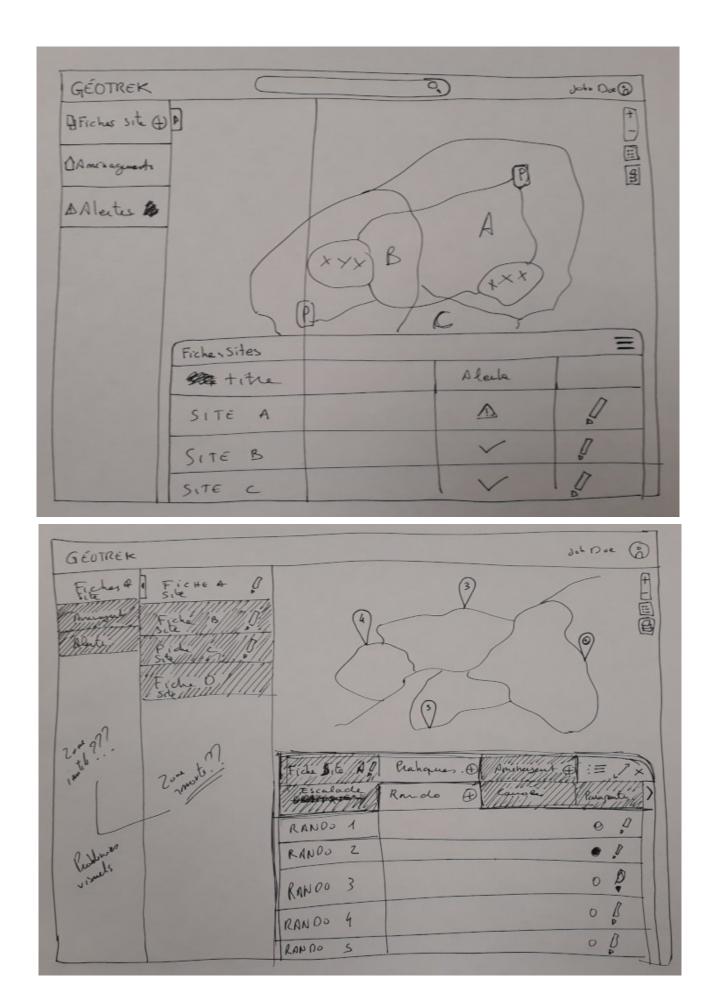








Makina Corpus – Atelier Lot 1 – 21 et 22 janvier 2020 - Compte-rendu



# Jour 2 - Priorisation

Résultat de la priorisation sous forme de post'its issus du travail de la veille :

#### Verticalement

• La première rangée de post'its en haut concerne les aspects purement techniques



• La deuxième rangée de post'its au milieu concerne les nouvelles fonctionnalités (outdoor, API pour BAM),

- La troisième rangée de post'its en bas concerne les améliorations diverses
- L'idée est de consacrer environ 1/3 du temps sur chacune de ces rangées

#### Horizontalement

- La première colonne à gauche est prioritaire
- · La deuxième colonne est ce qui est moins prioritaire
- La troisième colonne est ce qui semble optionnel ou mal défini

# **Premier sprint**

- C'est le cadre en haut à gauche.
- Il est estimé à 20jh (~2 devs sur 2 semaines).
- Il correspond à la première colonne de la première ligne (aspects techniques pour commencer, les plus prioritaires d'abord).
- Il mélange des tâches de dev précises et des tâches de conception qui permettront de préciser le sprint suivant sur les sujets du tuilage et du découplage.
- L'idée est de poursuivre ainsi en pensant non seulement au sprint en cours mais aussi à ce qu'il est nécessaire de préparer pour le sprint suivant.

#### Suivi sur Github

Dans un souci de transparence, l'avancement des tâches sera visible sur Github sous forme d'issues et de pull requests. Le tout rassemblé dans un board : <a href="https://github.com/GeotrekCE/Geotrek-admin/projects/2">https://github.com/GeotrekCE/Geotrek-admin/projects/2</a>

#### Bons de commande

Un bon de commande a été fait au sujet de l'outdoor. Mais il n'est pas nécessaire de commencer par cela. Ce qui est par contre impossible, c'est de ne pas avoir développé les fonctionnalités outdoor à la fin du projet.

# Jour 2 - Présentation de Géocompostelle

Fort intérêt de la part de Camille et Amandine pour l'aspect WebSIG.