



**HAL**  
open science

## Développement d'un WEB -SIG pour UnderMap : service dédié aux réseaux enterrés

Karl Nelson Tayou

► **To cite this version:**

Karl Nelson Tayou. Développement d'un WEB -SIG pour UnderMap : service dédié aux réseaux enterrés. Sciences de l'environnement. 2019. dumas-03383158

**HAL Id: dumas-03383158**

**<https://dumas.ccsd.cnrs.fr/dumas-03383158>**

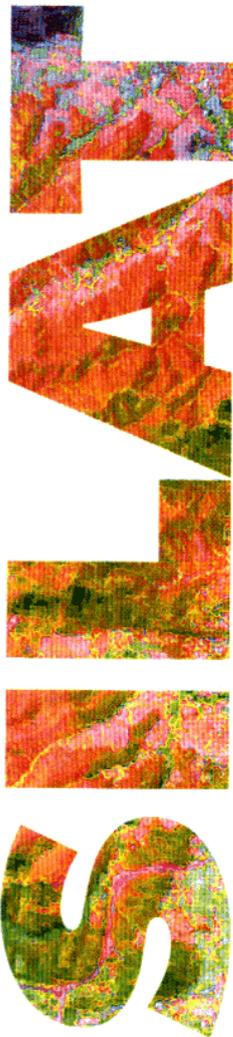
Submitted on 18 Oct 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

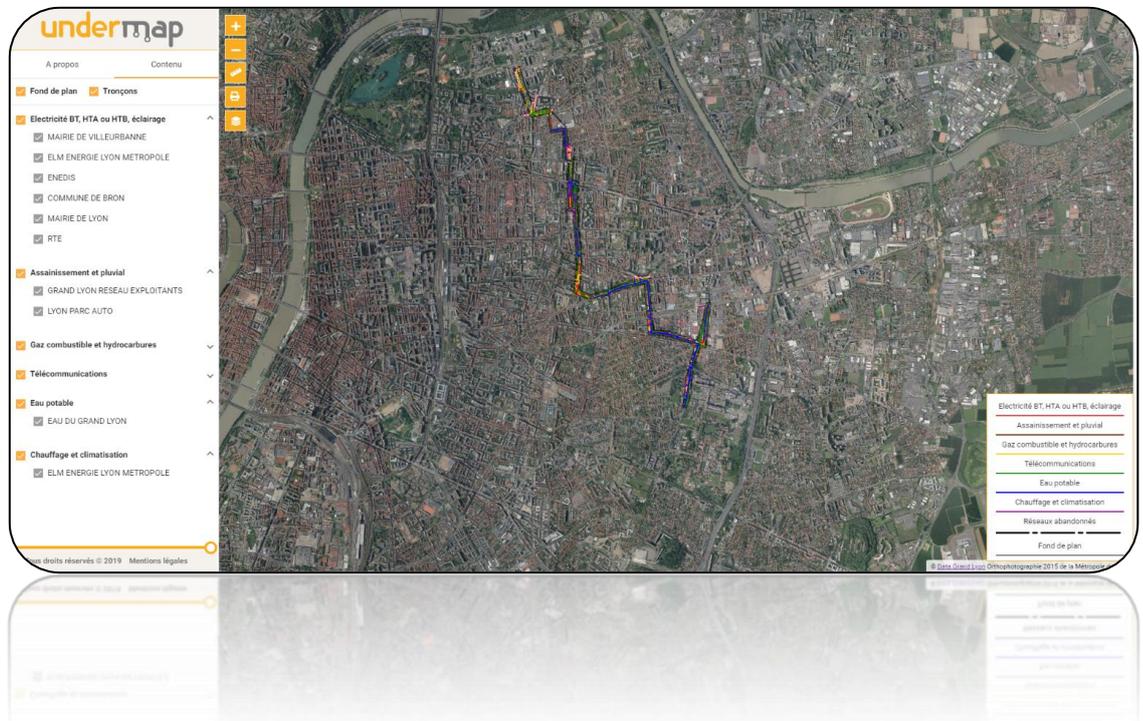


Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0  
International License



# Développement d'un WEB - SIG pour UnderMap : service dédié aux réseaux enterrés

*Karl Nelson TAYOU*



MANAGER  
DE  
PROJETS  
GEOMATIQUES

Directeur du projet : Loic ELSHOLZ (Futurmap)  
Rapporteurs : Lucile SAUTOT (AGROPARISTECH)  
Nicolas MOYROUD (IRSTEA)

Octobre 2019



L'établissement AgroParisTech diffuse ce mémoire et se dégage de toute responsabilité sur le contenu proposé.

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition.

L'obligation de citation et de référencement lors de l'utilisation de ce document doit être respectée selon la licence Creative Commons apposée

Ce document est soumis aux conditions d'utilisation "Paternité – Pas d'utilisation commerciale – Pas de modification 4.0 France disponible en ligne : <https://creativecommons.org/licenses/by-nc-nd/4.0/>



# Table des matières

Table des Illustrations .....	iii
Table des tableaux .....	iv
1 Introduction .....	1
1.1 Présentation de l'entreprise .....	1
1.2 Contexte de l'étude.....	1
1.3 Objectif de l'étude et structure du projet.....	2
2 UNDERMAP : service digital de plan de synthèse de réseaux .....	3
2.1 Description du service .....	3
2.2 Méthode de production des livrables.....	4
2.2.1 Réception et préparation des données du client.....	4
2.2.2 Production et contrôle.....	5
2.2.3 Génération des livrables .....	7
3 Développement d'un SIG visant à déployer des plans sur le WEB .....	8
3.1 Diffusion des données par des flux consommables sur le Web via un serveur cartographique	9
3.1.1 Etat de l'art des serveurs cartographiques .....	9
3.1.2 Installation et mise en œuvre de QGIS SERVER.....	11
3.1.3 Utilisation de QGIS SERVEUR.....	12
3.2 Mise en place d'un système de pré-calcul des données pour diffusion rapide .....	13
3.2.1 Pyramide d'images .....	14
3.2.2 Création d'une pyramide d'images .....	14
3.2.3 Utilisation d'une pyramide d'images .....	16
3.3 Visualisation et interaction avec les données sur le WEB.....	17
3.3.1 Le menu des couches disponibles.....	18
3.3.2 Le bloc carte .....	18
3.3.3 Le menu d'impression .....	20
4 Intégration du processus de déploiement du Web-SIG dans la chaîne de production.....	21
4.1 Création automatique des fichiers de configuration .....	21

4.1.1	Fichier de configuration JSON utile à l'application Web.....	21
4.1.2	Premier fichier de configuration du serveur de pré calcul.....	22
4.1.3	Deuxième fichier de configuration du serveur de pré calcul.....	23
4.2	Déploiement optimisé d'un plan en ligne.....	25
5	Cas pratiques et analyse.....	28
5.1	Projet de tramway de 7 kilomètres à Lyon.....	28
5.2	Projet de faible envergure avec fond de plan cadastral .....	30
6	Conclusion générale et perspectives.....	32
7	Remerciements .....	34
8	Bibliographie .....	35
9	Glossaire.....	36
10	Annexes.....	37
	Résumé.....	49
	Abstract .....	49

## Table des Illustrations

Figure 1:le plan de synthèse de réseaux réunit plusieurs plans de réseaux en un seul plan unique .....	3
Figure 2:architecture générale d'un projet UNDERMAP dans QGIS .....	5
Figure 3:Exemple de fiche à remplir lors d'une vectorisation d'un réseau qui a des caractéristiques spécifiques (Réseau électrique – caractéristique spécifique BT (basse tension) .....	6
Figure 4:Exemple de fiche à remplir lors d'une vectorisation .....	6
Figure 5:Fonctionnement général d'un WEB-SIG .....	8
Figure 6:architecture d'un projet UNDERMAP .....	12
Figure 7:illustration d'une pyramide (source IGN) .....	14
Figure 8: Stockage de la donnée pré calculée .....	14
Figure 9: Le schéma définitif du fonctionnement du Web-SIG UNDERMAP .....	16
Figure 10: Menu des réseaux sur l'application WEB-SIG.....	18
Figure 11:Menu d'impression de plan sur le WEB-SIG .....	20
Figure 12:image avec une densité de réseaux moyenne, 4 KO .....	23
Figure 13:image avec une faible densité de réseaux, 2.47KO.....	23
Figure 14: Réseau avec le style de production (QGIS) .....	25
Figure 15: réseau avec le style d'OPENLAYERS .....	25
Figure 16: Schéma de fonctionnement du FTP d'UNDERMAP .....	26
Figure 17: Interface principale du FTP qui liste tous les projets disponibles.....	27
Figure 18:Actions possible sur un projet.....	27
Figure 19:Log ou suivi du pré calcul d'un projet sur le FTP .....	27
Figure 20:Interface pour uploader un projet UNDERMAP .....	27
Figure 21:Fond de plan du projet superposé à l'image aérienne de la ville de Lyon.....	29
Figure 22:Fond de plan et réseaux du projet .....	29
Figure 23:Image aérienne de la ville de Lyon, fond de plan et réseaux du projet.....	29
Figure 24: vue globale du projet.....	30
Figure 25:espace client sur la plateforme UNDERMAP.....	31
Figure 26: visualisation interactive du plan dans l'espace client .....	31

## Table des tableaux

Tableau 1:tableau de comparaisons des serveurs cartographiques.....	10
Tableau 2 : estimation du nombre d'images pré calculées par niveau de zoom sur 40 000m <sup>2</sup> .....	24

# 1 Introduction

## 1.1 Présentation de l'entreprise

La société FUTURMAP est une jeune entreprise spécialisée dans le domaine du traitement de données géographiques. Fondée en 2013 à Madagascar par M. Luc STRAUB, ingénieur géomètre-topographe, celle-ci a connu une croissance particulièrement rapide, et compte désormais plus de deux cent cinquante collaborateurs.

Les prestations proposées par la société incluent principalement la réalisation de plans 2D et 3D (plans de corps de rue, plans d'intérieurs, coupes de bâtiments, façades), la modélisation 3D et le BIM (*Building Information Modeling*), ainsi que des travaux de vectorisation. La majeure partie de ces activités repose sur le traitement de nuages de points issus de scanners laser, mobiles ou statiques.

Le service de production de l'entreprise est partagé en plusieurs cellules, chacune spécialisée dans un domaine d'activité, et dirigée par un Chef de Projet.

C'est au sein du département UNDERMAP, spécialisé dans le domaine des réseaux et des nouvelles technologies, que s'est déroulé le présent stage sous la direction de M. Loïc ELSHOLZ.

## 1.2 Contexte de l'étude

UNDERMAP, service concerné par l'étude, est un projet lancé par FUTURMAP en début d'année 2018.

Il s'agit principalement d'un service digital de plans de synthèse de réseaux à destination des maîtres d'ouvrage, maîtres d'œuvre et entreprises de travaux publics désirant mieux anticiper leurs chantiers.

Par le biais de l'interface numérique en ligne [www.undermap.me](http://www.undermap.me), l'utilisateur définit une zone de projet et dépose les plans de réseaux communiqués par les exploitants.

FUTURMAP mobilise alors ses équipes internes de dessinateurs pour produire la synthèse sous forme de plan ordonné et adapté à l'utilisateur, au format souhaité.

Le livrable est alors disponible au téléchargement par l'utilisateur directement sur la plateforme digitale.

En parallèle, un outil WEB lié à une application mobile permet la visualisation des plans de synthèse ainsi que d'autres fonctionnalités relatives à la gestion des problématiques chantier et à la communication entre ses différents acteurs. Avant la réalisation de ce stage, l'aspect visuel de ces applications était basé sur Google Maps et ses formats donnés (KML). Ce choix de technologie, pratique de prime abord, limite grandement les possibilités d'exploitation des plans, l'interopérabilité entre les différentes sources de données, et induit des coûts supplémentaires.

### 1.3 Objectif de l'étude et structure du projet

L'intention générale de l'étude consiste à mettre en place une infrastructure permettant de visualiser et d'interagir avec les données concernant les plans de synthèse sur le WEB et sur mobile.

Ces processus doivent impérativement se greffer à la chaîne de production existante, en tenant compte des procédures, logiciels et formats utilisés, afin d'affecter au minimum les rendements actuels du département.

Il s'agit également de considérer la variation importante d'envergure des projets traités par le département (d'une centaine de mètre à plusieurs kilomètres de voirie), qui influence de manière critique le poids des données et la superficie de visualisation, et ainsi la fluidité de navigation sur les outils à mettre en place.

Au terme du stage, l'objectif est d'incorporer l'application sur la plateforme UNDERMAP existante, afin d'être utilisée au quotidien par la clientèle de FUTURMAP.

Au cours de l'étude, un plugin sous QGIS relatif à la chaîne de production des plans a également été maintenu et largement amélioré.

Dans un premier temps, nous présenterons le service digital UNDERMAP, sa chaîne de production et ses différents livrables concernant les plans de synthèse de réseaux.

Nous évoquerons dans un deuxième temps les différentes étapes de mise en place d'une infrastructure visant à déployer des plans sur le WEB, à savoir :

- Diffusion des données par des flux consommables sur le WEB via un serveur cartographique : choix du serveur et configuration
- Mise en place d'un système de pré-calcul des données pour diffusion rapide : intérêt de pré-calculer les données et mise en place du système
- Visualisation et interaction avec les données sur le WEB : exploitation des flux de données mis en place et interaction avec l'utilisateur sur le WEB

Nous montrerons dans un dernier temps l'intégration du processus de déploiement WEB-SIG dans la chaîne de production, avant d'établir notre conclusion.

## 2 UNDERMAP : service digital de plan de synthèse de réseaux

### 2.1 Description du service

UNDERMAP est un service digital de plans de synthèse de réseaux enterrés à destination des maîtres d'ouvrage, maîtres d'œuvre et entreprises de travaux publics désirant mieux anticiper leurs chantiers.

Il répond au besoin exprimé par les dessinateurs, les maîtres d'œuvre, la maîtrise d'ouvrage, les porteurs de projet d'aménagements et les entreprises générales de disposer rapidement d'un plan numérique unique et standardisé réunissant l'ensemble des réseaux existants tels que connus par leurs exploitants au démarrage d'un projet.

Les livrables sont donc des plans en PDF, ou des fichiers DAO et SIG.

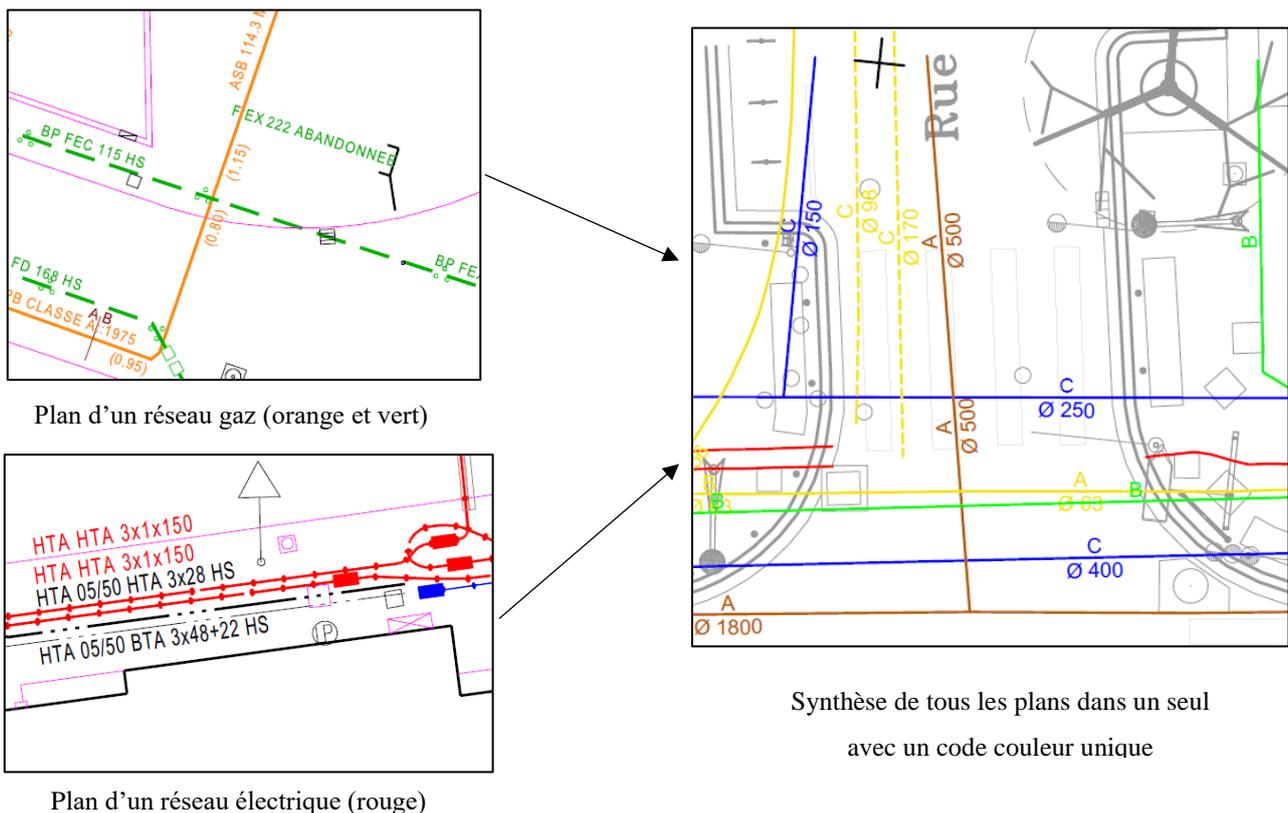


Figure 1: le plan de synthèse de réseaux réunit plusieurs plans de réseaux en un seul plan unique

Une fois les plans réalisés, le client peut créer des comptes utilisateurs répartis en équipes. Ces utilisateurs pourront annoter, commenter ou géo-référencer des images sur les plans directement depuis le terrain via l'application mobile, ou depuis le bureau via l'applicatif WEB.

## 2.2 Méthode de production des livrables

Le plan de synthèse de réseaux est produit en SIG, puis intégré dans un logiciel de DAO pour répondre aux besoins des clients plus habitués à ce type de logiciel.

Le logiciel de SIG utilisé est QGIS. Un plugin y a été développé pour faciliter le traitement effectué par l'équipe de production, et automatiser certaines tâches. Tout au long du stage, nous avons pu maintenir ce plugin et le faire évoluer de sa version 1 à sa version 3 actuelle (cf. Annexe 1 fonctionnalités du plugin).

De la réception d'une demande jusqu'à la livraison, on retrouve six étapes principales :

### 2.2.1 Réception et préparation des données du client

Le client transmet via la plateforme en ligne les données de son projet, à savoir :

- L'emprise de son chantier : c'est la zone géographique où il souhaite la prestation de service. Il peut s'agir d'une rue ou d'un ensemble de rues.

Il prépare alors soit un fichier vectoriel « kml » de sa zone pour l'importer dans la plateforme ou il la dessine sur une image aérienne directement sur la plateforme lorsqu'il passe sa commande.

- Les récépissés des DT-DICT au format PDF : Ce sont des plans (DAO) en PDF que le prestataire a recueilli auprès de tous les exploitants qui interviennent dans sa zone d'intérêt. Il les téléverse sur la plateforme en passant sa commande.
- Le fond de plan s'il en possède un : Il s'agit du PCRS ou d'un plan topographique de la zone. S'il n'en a pas, on extrait le cadastre de cette zone. Le fond de plan est utilisé pour géo référencer les récépissés (plans) des DT-DICT.

Lors de la préparation il est question de passer des données du client à un projet QGIS, où l'équipe de production pourra géo-référencer et vectoriser les plans PDF. Cette tâche est à la charge du chef de projet.

Tout d'abord, les données du client sont classées dans un dossier avec une architecture définie (cf. Annexe 5) en suivant une procédure disponible en Annexe 6.

Ensuite, le plugin UNDERMAP du logiciel QGIS va automatiquement lire le dossier et créer des couches regroupées en plusieurs catégories :

- Fond de plan : contient le fond de plan du client et son emprise ;

- **RSX** : Contient les couches vectorielles vides pour chaque exploitant. C'est dans ces couches que seront vectorisés les réseaux des exploitants. Dans ces couches, le plugin créé des champs bien précis et applique une symbologie générique au produit UNDERMAP
- **TIF** : Contient un groupe pour chaque exploitant. C'est ici que seront mises les images géo-référencées de chaque exploitant.

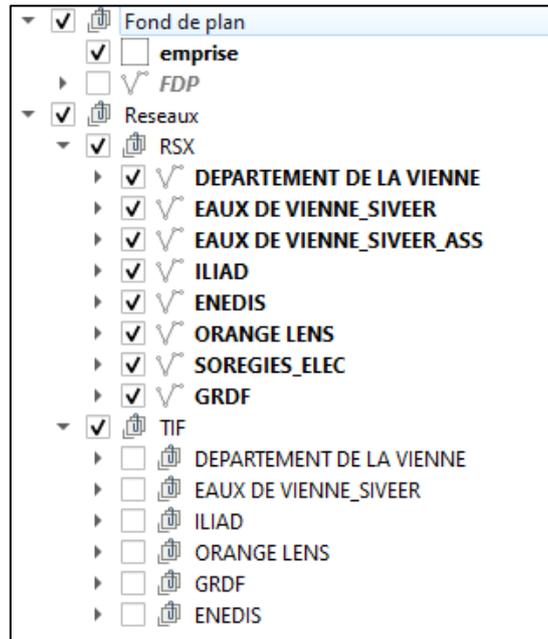


Figure 2: architecture générale d'un projet UNDERMAP dans QGIS

### 2.2.2 Production et contrôle

L'équipe de production a pour mission dans un premier temps de géo-référencer les plans PDF à partir du fond de plan et ensuite de vectoriser tous les réseaux se trouvant sur ces images géo-référencées.

- **Géo-référencement** : grâce au plugin, les écarts moyens et maximum de chaque géo-référencement sont enregistrés pour un contrôle qualité qui interviendra ensuite et sera réalisé par un assistant qualité.

- Vectorisation : le plugin assiste l'opérateur pour remplir les champs, en lui proposant des listes déroulantes ou des cases à cocher pour tous les champs, sauf celui du diamètre qui est rempli manuellement.

The dialog box titled 'EAU ET FORCE - Attributs d'entités' has a close button in the top right. Below the title bar is a section labeled 'ACTIONS'. The fields are:
 

- Exploitant: EAU ET FORCE
- Réseau: Eau potable (dropdown menu)
- Diamètre (mm): 20 (text input with a clear button)
- Classe: C (dropdown menu)
- Abandon:

 At the bottom right are 'OK' and 'Annuler' buttons.

Figure 4: Exemple de fiche à remplir lors d'une vectorisation

The dialog box titled 'ENEDIS - Attributs d'entités' has a close button in the top right. Below the title bar is a section labeled 'ACTIONS'. The fields are:
 

- Exploitant: ENEDIS
- Réseau: Electricité BT, HTA ou HTB, éclairage (dropdown menu)
- Type d'électricité: BT (dropdown menu)
- Diamètre (mm): 20 (text input with a clear button)
- Classe: B (dropdown menu)
- Abandon:

 At the bottom right are 'OK' and 'Annuler' buttons.

Figure 3: Exemple de fiche à remplir lors d'une vectorisation d'un réseau qui a des caractéristiques spécifiques (Réseau électrique – caractéristique spécifique BT (basse tension))

Le champ « Classe » représentant le niveau de précision du réseau est une liste déroulante contenant les valeurs A, B et C.

Le champ « Abandon » pour déterminer si un réseau est abandonné ou non est représenté sous forme de case à cocher

Le champ « Exploitant » est rempli automatiquement grâce au nom de la couche

Le champ « Réseau » est une liste déroulante entre les différents types de réseaux existants. Lorsqu'il s'agit d'un réseau qui possède des caractéristiques spécifiques, un nouveau champ avec une liste déroulante de valeurs correspondantes à ces caractéristiques apparaît.

Le contrôle intervient ensuite pour vérifier l'exactitude des champs remplis et des tracés.

### 2.2.3 Génération des livrables

Cette étape est entièrement automatisée par le biais du plugin.

La vectorisation se faisant sur plusieurs PDF et à la main, il arrive qu'on ait un même réseau divisé en plusieurs entités. Un plugin a été fait pour fusionner des entités ayant les mêmes caractéristiques et qui se réunissent seulement à leur sommets. Cela optimise le dessin et donne la possibilité d'isoler ou de sélectionner un réseau entièrement en un seul clic.

Une fois le réseau « propre », le plugin permet d'exporter les différents livrables :

- Un fichier Excel résumant toute la production : la longueur totale de réseau pour chaque exploitant, selon ses caractéristiques, et la précision des géo-référencements
- Les faisceaux d'incertitude des réseaux (« buffers ») : il s'agit des couches polygonales représentant l'emprise d'incertitude liée à la classe de précision des réseaux
- Les réseaux sous format GEOJSON afin d'être transformés en fichier DAO sous AUTOCAD, où seront par ailleurs extraites les impressions des plans de synthèse en format PDF

Le but de la présente étude est de créer un nouveau livrable : la visualisation et l'exploitation de ces plans sur le WEB.

### 3 Développement d'un SIG visant à déployer des plans sur le WEB

Pour visualiser les données géographiques sur le WEB, trois grands ensembles sont nécessaires :

- Les données SIG en entrée: il s'agit des données rasters ou vecteurs qui sont stockées soit en BD soit en dépôts de fichiers. Dans notre cas, ce sont des données vecteur des réseaux.
- Le serveur cartographique : il va consommer les données SIG pour les distribuer en flux de données vers un visualiseur (WEB ou mobile) sous forme d'images ou de vecteurs.
- Un visualiseur qui recevra les flux du serveur cartographique, appliquera ou non un style sur les données et les affichera sur un écran.

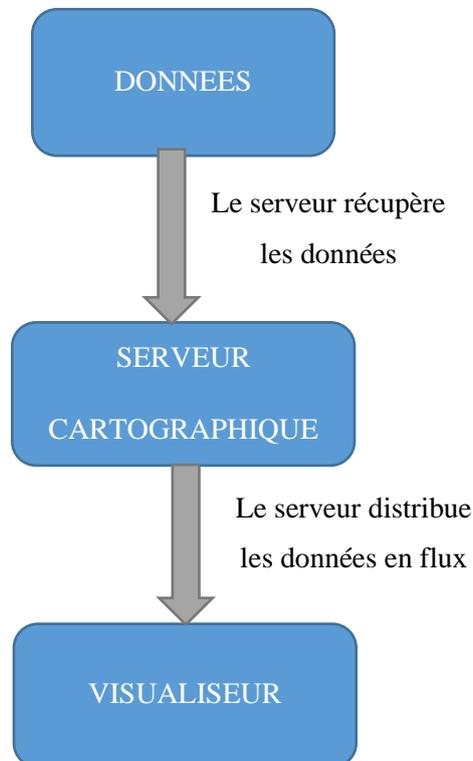


Figure 5: Fonctionnement général d'un WEB-SIG

Nous avons expliqué dans la première partie toute la chaîne de production des données qui représente le premier groupe. Dans la suite, nous nous intéresserons aux deux autres groupes à savoir le serveur cartographique et le visualiseur.

### 3.1 Diffusion des données par des flux consommables sur le Web via un serveur cartographique

#### 3.1.1 Etat de l'art des serveurs cartographiques

Les serveurs cartographiques diffusent des flux de données géographiques encore appelés Web services géographiques normés par l'OGC. Selon le type de données disponibles en entrée (vecteur ou raster), on peut distinguer deux types de fonctionnement :

- Lorsqu'il consomme des rasters : le serveur va appliquer un style par défaut comme QGIS le fait à l'ouverture d'un fichier raster, ou un style qu'on aura à définir; Ensuite, il va renvoyer des images sur l'emprise demandée par le visualiseur.
- Lorsqu'il consomme des vecteurs : dans ce cas, il peut soit délivrer des vecteurs en retour, soit des rasters au visualiseur. Il appliquera un style par défaut sur les données, ou un style défini lorsqu'il délivrera des images.

Mais lorsqu'il délivre en vecteur, aucun style ne peut être appliqué coté serveur, car il renvoie des données brutes sous format vectoriel (GEOJSON, GML etc...). C'est au visualiseur d'y appliquer un style avant affichage.

L'OGC définit principalement deux types de flux de données selon le type de données délivrées :

- Le WMS : lorsque ce sont des images qui sont diffusées. Il possède trois méthodes **GetCapabilities**, **GetMap** et **GetFeatureInfo** qui permettent respectivement d'obtenir les métadonnées du serveur (couches disponibles, projections associées, auteur...), des images d'une couche sur une emprise voulue, et les informations attributaires d'un point grâce à ces coordonnées en pixels sur une image.
- Le WFS : lorsque les vecteurs sont délivrés. Il possède en général 5 méthodes **GetCapabilities**, **DescribeFeatureType**, **GetFeature**, **Transaction** et **LockFeature** qui permettent respectivement de connaître les capacités du serveur (savoir quelles méthodes il supporte), la structure de chaque entité susceptible d'être fournie par le serveur, de délivrer les données sur une emprise voulue, de modifier les données et de bloquer les objets lors d'une modification.

Les serveurs cartographiques les plus utilisés sont :

**MAPSERVER** : C'est le plus vieux et le plus performant (première version en 1994, dernière version en septembre 2019). Il est la référence en matière de cartographie sur internet.



**QGIS SERVER** : Celui qui a le vent en poupe. De manière très simple, il permet de mettre les cartes éditées sur QGIS DESKTOP en ligne. Depuis 2012, il délivre la majorité des services du standard OGC, et est en perpétuelle évolution. Il utilise le même moteur de rendu que celui de QGIS DESKTOP, c'est-à-dire que les fichiers de styles entre les deux logiciels sont compatibles. Il évolue en même temps que le logiciel bureautique QGIS. Avec lui, chaque projet QGIS est également un projet QGIS SERVER.



**GEOSEVER** : Conçu pour l'interopérabilité, il publie les données de toutes les sources principales de données spatiales utilisant le standard OGC. Sa dernière version date de septembre 2019.



Tableau 1: tableau de comparaisons des serveurs cartographiques

	<b>MAPSERVER</b>	<b>QGIS SERVER</b>	<b>GEOSEVER</b>
Facilité à mettre en place	Moyen	Moyen	Facile
Compatible avec les fichiers standards de l'OGC	Oui	Oui	Oui
Propose des Web services aux standards de l'OGC	Oui	Oui	Oui
Bibliothèque de style	A sa propre nomenclature. Moyen	Fichier de style du logiciel bureautique. Facile	Utilise les fichiers SLD (XML). Difficile
Paramétrage du serveur pour un projet	A partir d'un fichier texte. Difficile	A partir du logiciel bureautique. Facile	A partir d'une interface graphique. Moyen
Paramétrage du cartouche d'impression	-	Dans le logiciel bureautique	-
Mise à jour régulière (dernière version)	Septembre 2019	Septembre 2019	Septembre 2019
Automatisation de création de projets dans le serveur	Difficile	Moyen	Difficile
Coût	Gratuit	Gratuit	Gratuit

Au vu du tableau ci-dessus, du fait que la production est effectuée sur QGIS avec des styles et cartouche déjà définis que l'entreprise souhaite conserver sur le Web, et qu'il est de surcroît assez facile

d'automatiser la création des projets, le choix a naturellement été porté sur le serveur cartographique de QGIS.

### 3.1.2 Installation et mise en œuvre de QGIS SERVER

Cette étape a été effectuée sur un ordinateur avec le système d'exploitation UBUNTU version 16.04.

Il s'agit d'ajouter les sources du dépôt de QGIS dans la liste des dépôts de l'ordinateur, c'est-à-dire dans le fichier `/etc/apt/sources.list` en collant dans ce fichier les deux lignes suivantes :

```
deb https://qgis.org/ubuntu-ltr xenial main
deb-src https://qgis.org/ubuntu-ltr xenial main
```

Ensuite, on entre dans l'ordre les commandes suivantes qui vont respectivement rafraîchir la liste des dépôts et installer QGIS SERVEUR :

```
sudo apt-get update
sudo apt-get install qgis-serve
```

Pour terminer, il faut que l'on puisse accéder au serveur cartographique depuis un navigateur internet. Pour cela, il suffit de mettre en place une configuration HTTP dans le même ordinateur. Dans le répertoire `/etc/apache2/sites-available`, on crée un fichier nommé `qgis.demo.conf` avec le contenu ci-dessous :

```
<VirtualHost *:80>
    ServerAdmin Webmaster@localhost
    ServerName qgis.demo
    DocumentRoot /var/www/html

    # Longer timeout for WPS... default = 40
    FcgidIOTimeout 120
    FcgidInitialEnv QGIS_SERVER_LOG_FILE /var/log/qgis/qgisserver.log
    FcgidInitialEnv QGIS_SERVER_LOG_LEVEL 0

    FcgidInitialEnv QGIS_DEBUG 1

    # Tell QGIS Server instances to use a specific display number
    FcgidInitialEnv DISPLAY ":99"
    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin/">
```

```
AllowOverride None

Options +ExecCGI -MultiViews -SymLinksIfOwnerMatch

Order allow,deny

Allow from all

Require all granted

</Directory>

<IfModule mod_fcgid.c>
FcgidMaxRequestLen 26214400
FcgidConnectTimeout 60
</IfModule>

</VirtualHost>
```

Ensuite, on entre dans l'ordre les commandes suivantes, pour activer le fichier que l'on vient de créer :

```
sudo a2enmod fcgid
sudo a2ensite qgis.demo
sudo service apache2 restart
```

Le serveur cartographique est désormais disponible à l'adresse « [http://qgis.demo/cgi-bin/qgis\\_mapserv.fcgi](http://qgis.demo/cgi-bin/qgis_mapserv.fcgi) ».

### 3.1.3 Utilisation de QGIS SERVEUR

Pour un projet UNDERMAP dont le projet QGIS est nommé « `psr_rue_du_pont` », situé dans le dossier `/var/www/` et avec l'architecture suivante :



Figure 6: architecture d'un projet UNDERMAP

Toutes les couches du projet sont accessibles par les Web services :

- WMS : pour avoir le lien WMS d'une couche, il suffit d'entrer le lien suivant en remplaçant «nom\_de\_la\_couche» par le nom d'une couche du projet : [http://qgis.demo/cgi-bin/qgis\\_mapserv.fcgi?SERVICE=WMS&VERSION=1.3.0&map=var/www/psr\\_rue\\_du\\_pont.qgs&&LAYERS=nom\\_de\\_la\\_couche](http://qgis.demo/cgi-bin/qgis_mapserv.fcgi?SERVICE=WMS&VERSION=1.3.0&map=var/www/psr_rue_du_pont.qgs&&LAYERS=nom_de_la_couche) .

Exemple, pour la couche ORANGE, ce sera : [http://qgis.demo/cgi-bin/qgis\\_mapserv.fcgi?SERVICE=WMS&VERSION=1.3.0&map=var/www/psr\\_rue\\_du\\_pont.qgs&&LAYERS=ORANGE](http://qgis.demo/cgi-bin/qgis_mapserv.fcgi?SERVICE=WMS&VERSION=1.3.0&map=var/www/psr_rue_du_pont.qgs&&LAYERS=ORANGE)

- WFS : pour avoir le lien WFS d'une couche uniquement vecteur, il suffit d'entrer le lien suivant en remplaçant «nom\_de\_la\_couche» par le nom d'une couche du projet : [http://qgis.demo/cgi-bin/qgis\\_mapserv.fcgi?SERVICE=WFS&VERSION=1.3.0&map=var/www/psr\\_rue\\_du\\_pont.qgs&&LAYERS=nom\\_de\\_la\\_couche](http://qgis.demo/cgi-bin/qgis_mapserv.fcgi?SERVICE=WFS&VERSION=1.3.0&map=var/www/psr_rue_du_pont.qgs&&LAYERS=nom_de_la_couche) .

Exemple, pour la couche ORANGE, ce sera : [http://qgis.demo/cgi-bin/qgis\\_mapserv.fcgi?SERVICE=WFS&VERSION=1.3.0&map=var/www/psr\\_rue\\_du\\_pont.qgs&&LAYERS=ORANGE](http://qgis.demo/cgi-bin/qgis_mapserv.fcgi?SERVICE=WFS&VERSION=1.3.0&map=var/www/psr_rue_du_pont.qgs&&LAYERS=ORANGE)

### 3.2 Mise en place d'un système de pré-calcul des données pour diffusion rapide

Lors de la visualisation d'un projet UNDERMAP qui contient un nombre  $n$  de réseaux, à chaque animation (déplacement ou zoom) de la carte, le serveur cartographique devra calculer les données pour ces  $n$  réseaux pour le rendu, sachant que ce nombre varie entre 4 et 30 réseaux, avec une moyenne de l'ordre de 12.

Ce calcul de données est facile et rapide lorsqu'il est en mode WFS (rend des vecteurs), mais il peut devenir très consommateur de ressources et lent lorsqu'il doit convertir les données vecteur en images avant d'effectuer le rendu (WMS) ; Cette problématique peut rendre la visualisation très désagréable.

Ainsi, pour répondre au besoin de l'entreprise qui souhaite à la fois :

- Conserver sa charte graphique de production sur le Web, ce qui implique d'utiliser un mode qui rend des images, parce que l'on pourra ainsi styliser les données en les transformant en images.
- Obtenir une navigation la plus fluide possible

Il est donc nécessaire de pré calculer ces images, afin que celles-ci ne soient plus à calculer lors d'un appel du visualiseur, mais simplement à rendre.

Ce pré calcul revient à calculer ce qu'on appelle une « pyramide d'images ». Nous allons définir cette pyramide, puis décrire comment la créer et enfin comment l'utiliser.

### 3.2.1 Pyramide d'images

Il s'agit d'un maillage rectangulaire régulier de l'emprise d'une donnée. Une image est associée à chaque maille (appelée aussi tuile). Pour pouvoir représenter cette emprise à différents niveaux d'échelle (ou niveaux de zoom), des maillages de plus en plus fins sont définis pour la zone. Le cache d'images ainsi tuilées apparaît donc comme une pyramide dont chaque niveau de zoom représente un étage.

Les images résultantes peuvent être compressées et stockées dans une base de données SQLITE, qui aura 4 colonnes permettant de trouver une image :

- Zoom\_level : le niveau de zoom de l'image (étage)
- Tile\_column : la colonne de l'image
- Tile\_row : la ligne de l'image
- Tile\_data : l'image

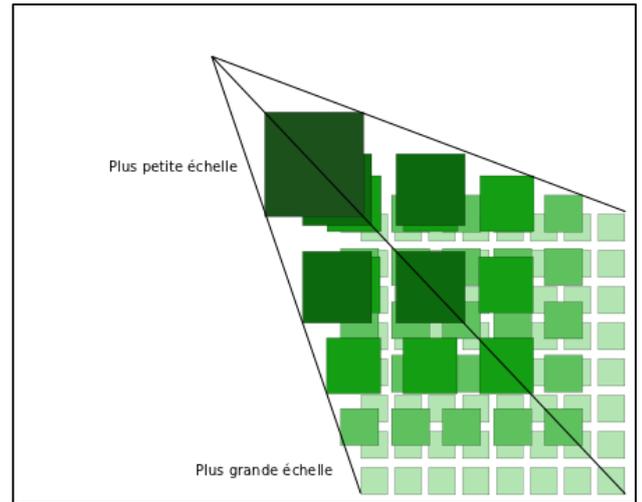


Figure 7: illustration d'une pyramide (source IGN)

zoom_level	tile_column	tile_row	tile_data
Filtre	Filtre	Filtre	Filtre
19	269267	187035	BLOB
20	538532	374068	BLOB
20	538534	374069	BLOB
20	538535	374069	BLOB
21	1077068	748140	BLOB
21	1077069	748140	BLOB
21	1077070	748140	BLOB
21	1077071	748140	BLOB

Figure 8: Stockage de la donnée pré calculée

### 3.2.2 Création d'une pyramide d'images

MapProxy est un logiciel Open Source permettant de pré calculer des données géo-spatiales. Il peut consommer des Web services WMS pour effectuer les pré calculs dans un premier temps et pouvoir les rendre ensuite sur le Web de manière très rapide. On parle alors du Web service WMTS.



Ce logiciel est facile à installer sur un ordinateur sous UBUNTU, qui possède déjà le gestionnaire de paquets python PIP. Il suffit d'entrer la commande :

```
pip install MapProxy
```

Une fois installé, il faut le configurer avec deux fichiers YAML, pour qu'il puisse pré calculer et rendre les tuiles sous le protocole WMTS :

- Premier fichier de configuration : il donne au logiciel le nom et la source de données pour chaque couche que l'on souhaite pré calculer. Il comporte plusieurs blocs, dont les principaux sont :
  - Sources : liste les sources de données que l'on souhaite pré calculer, avec un nom pour chaque source. Il faut spécifier le lien du serveur cartographique où il ira chercher la donnée (url), le mode souhaité de récupération de données (type), le système de coordonnées de la donnée en entrée (srs), l'emprise de toute la donnée (bbox) et le nom de la couche dans le serveur cartographique (layers)
  - Caches : liste de caches définissant comment stocker chaque source de donnée listée dans le bloc source. On lui spécifie le chemin du dossier où il devra stocker les images après pré calcul (filename), le type de format qu'on souhaite utiliser pour le stockage (type) et le nom de la source de données qu'on souhaite stocker (sources)
  - Layers : liste de toutes les couches qu'il pourra délivrer en Web service sur le WEB. On lui renseigne le nom du cache où il ira chercher la donnée pré calculée avant de la rendre (source), et un titre qui sera l'identifiant de la couche
- Deuxième fichier de configuration, dit de « seeding » : il s'allie au premier fichier pour indiquer au logiciel comment il va pré calculer les données. Il décrit au logiciel sur quelle emprise, quels niveaux de zoom, dans quelle projection et sous quel format il doit pré calculer et stocker les images. Il est composé de deux blocs :
  - Coverages : liste des emprises sur lesquelles on souhaite pré calculer des données. Il suffit de spécifier l'emprise (bbox) et le système de projection utilisé dans les coordonnées de cette emprise (srs)
  - Seeds : liste de chaque opération de pré calcul qu'on souhaite faire. Pour chaque opération, il faut préciser le nom du cache à utiliser, celui-ci devant correspondre à l'un des caches définis dans le premier fichier (caches), la zone sur laquelle on veut pré calculer, qui doit correspondre à une des emprises définies dans le bloc « Coverage » et enfin définir les différents niveaux de zoom sur lesquels on souhaite effectuer l'opération de pré calcul (levels)

Une fois les deux fichiers créés, il faut lancer toutes les opérations de pré calcul avec la commande :

```
mapproxy-seed -f premier_fichier.yaml -s deuxième_fichier.yaml
```

Les opérations peuvent prendre plus ou moins de temps en fonction de l'emprise et des niveaux de zoom où elles doivent être effectuées. Nous verrons dans partie d'automatisation les niveaux de zoom choisis pour le projet UNDERMAP.

### 3.2.3 Utilisation d'une pyramide d'images

Une fois le pré calcul terminé et le logiciel MapProxy lancé avec la commande suivante :

```
mapproxy-util serve-develop premier_fichier.yaml
```

Le serveur de tuiles est alors accessible sur le lien <http://localhost:8080/>.

Pour accéder aux images tuilées d'une couche nommée orange, en WMTS, il suffit d'entrer l'url suivante où z, tileCol et tileRow sont respectivement le niveau de zoom, la colonne et la ligne de l'image souhaitée :

```
http://localhost:8080/orange/z/tileCol/tileRow.png
```

Le schéma définitif du fonctionnement de l'infrastructure devient :

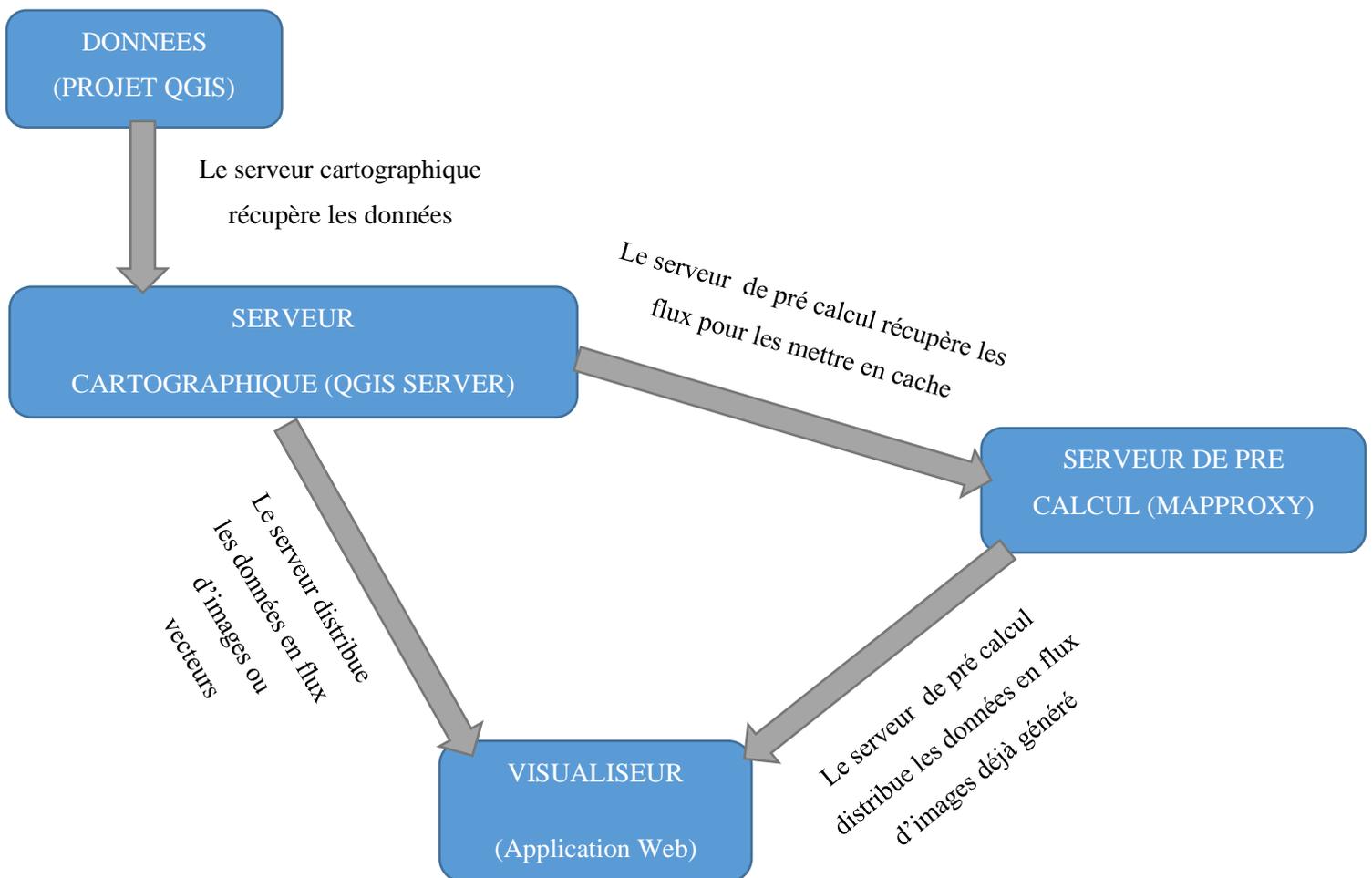


Figure 9: Le schéma définitif du fonctionnement du Web-SIG UNDERMAP

### 3.3 Visualisation et interaction avec les données sur le WEB

Il s'agit de mettre en place une page Web où l'on pourra manipuler la carte, interagir avec celle-ci et effectuer des impressions de cartes. Cette page consommera tous les Web services déjà mis en place précédemment.

L'application contient trois grandes parties :

- Le menu des couches disponibles : c'est la liste des réseaux disponibles, groupé par type de réseau. On peut afficher/masquer une couche de la carte.
- Le bloc carte : l'espace où l'on peut naviguer sur la carte et obtenir les caractéristiques d'une couche (ici les réseaux) en survolant.
- Un menu d'impression : pour imprimer une partie de la carte de la manière souhaitée. On peut renseigner un titre, une description, une échelle, un angle etc.

Pour construire l'application, on a besoin d'utiliser des langages Web tel que :

- Le HTML : pour afficher des éléments (textes et images) sur l'écran de l'utilisateur
- CSS : pour styliser l'interface afin qu'elle soit agréable à utiliser
- JavaScript : pour assurer l'interaction entre l'utilisateur et l'application. C'est grâce à ce langage que l'on pourra appeler les Web services et les interpréter.

A partir de ces langages, une bibliothèque a été créée pour permettre d'effectuer des requêtes sur des serveurs cartographiques, interpréter leurs réponses et les afficher sous forme de carte : OPENLAYERS.

Dans la suite, nous allons voir comment sont réalisées les trois parties de l'application avec l'aide de la bibliothèque OPENLAYERS.

### 3.3.1 Le menu des couches disponibles

Il a pour simple but de montrer tous les exploitants ayant des réseaux sur le projet, regroupé en type de réseau.

Pour ce faire, un fichier JSON est écrit à la fin de la production, afin de donner toutes les caractéristiques du projet à l'application.

Pour chaque réseau dans ce fichier, on dispose es informations suivantes :

- Le nom du réseau
- L'emprise du réseau
- L'identifiant du réseau dans le serveur cartographique et le serveur de tuiles
- Le type de réseau

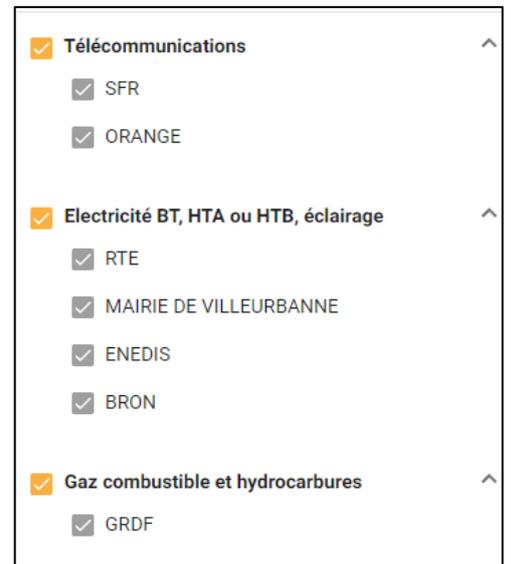


Figure 10: Menu des réseaux sur l'application WEB-SIG

Outre lister les réseaux, ce fichier fournit également à l'application Web :

- Le lien du projet dans le serveur cartographique
- L'emprise du projet, pour que l'application zoom sur le projet en question.
- L'identifiant du fond de plan et de l'emprise dans le serveur cartographique

Grâce à ce fichier de configuration, l'application Web liste tous les réseaux et zoom sur le projet dès son ouverture.

### 3.3.2 Le bloc carte

Il résulte de la superposition d'un fond de carte (image aérienne ou plan de ville) et des réseaux.

Pour créer cette superposition, il faut d'abord créer un conteneur de cartes avant d'y ajouter des couches qui s'y superposeront :

Dans le HTML :

```
<div id= "conteneur_de_carte" > </div>
```

Dans le JavaScript avec la bibliothèque OPENLAYERS :

```
var map = new Map({  
  target: 'conteneur_de_carte',
```

```
layers: [],  
});
```

En ce qui concerne le fond de carte, le choix s'est orienté vers des images aériennes de très haute résolution (inférieur à 30cm) qui permettent d'aller à des échelles très basses (inférieur à 1/200), afin de pouvoir apprécier le positionnement des réseaux sur l'image.

Une rapide comparaison a montré que les images Google s'arrêtent à l'échelle 1/500 dans les grandes villes, et peut s'arrêter à l'échelle 1/1000 sur d'autres zones. L'IGN commence à pixéliser ses images à environ 1/500 et conserve une bonne qualité sur tout le territoire Français. A noter également que les images des agglomérations en open data sont de meilleures qualité ; elles commencent souvent à pixéliser aux environs du 1/100.

Le choix a donc été porté sur un couple de solutions :

- Ortho photo d'une agglomération lorsque le projet est situé dans une zone où celle-ci est disponible en open data
- Ortho photo de l'IGN lorsque le projet se trouve dans une zone qui n'en possède pas

En ce qui concerne l'ajout des couches de réseaux dans la carte, l'application récupère les liens et identifiants de chaque réseau dans le fichier de configuration et les ajoute au conteneur de carte.

Pour des raisons d'optimisation du couple stockage et de fluidité de navigation qui seront expliquées dans la partie suivante, les réseaux sont affichés comme suit :

- Jusqu'au niveau de zoom 19, les images pré calculées sont affichées
- A partir du zoom 20, ce sont les données vecteur qui prennent le relai.

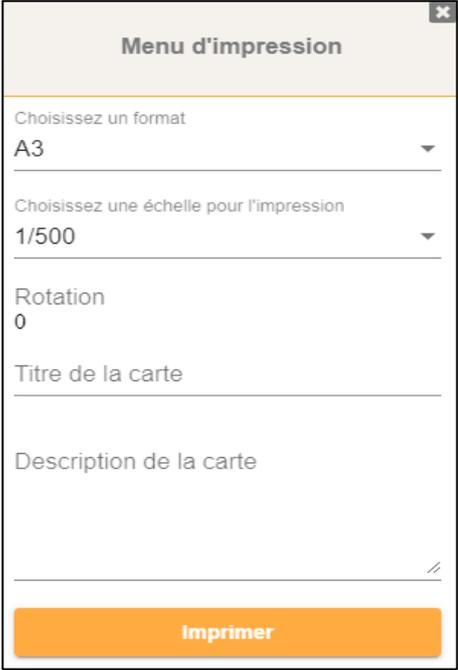
### 3.3.3 Le menu d'impression

Il permet d'entrer des paramètres nécessaires à l'impression d'une carte. Toutes ces informations seront envoyées au serveur cartographique, qui ensuite produira la carte au format PDF.

Pour envoyer ces informations, il faut utiliser des mots clés reconnus par le serveur :

- **EXTENT** : l'emprise de la carte à imprimer
- **ROTATION** : la rotation souhaitée
- **SCALE** : l'échelle à laquelle on souhaite imprimer
- **LAYERS** : liste des couches qu'on souhaite imprimer
- **TEMPLATE** : le nom du composeur qu'on souhaite utiliser pour imprimer. Ce composeur doit être préalablement créé et sauvegarder dans le projet QGIS

Grâce au composeur que nous avons développé, nous avons la possibilité d'ajouter les mots clés que l'on souhaite ; par exemple, un mot clé pour un titre, une description et un format de papier (A4, A5 etc.)



The image shows a screenshot of the 'Menu d'impression' (Print Menu) dialog box in QGIS. The dialog has a title bar with the text 'Menu d'impression' and a close button. Below the title bar, there are several input fields and a button. The first field is 'Choisissez un format' (Choose a format) with a dropdown menu showing 'A3'. The second field is 'Choisissez une échelle pour l'impression' (Choose a scale for printing) with a dropdown menu showing '1/500'. The third field is 'Rotation' with a text input field showing '0'. The fourth field is 'Titre de la carte' (Map title) with a text input field. The fifth field is 'Description de la carte' (Map description) with a text input field. At the bottom of the dialog, there is an orange button labeled 'Imprimer' (Print).

Figure 11: Menu d'impression de plan sur le WEB-SIG

## 4 Intégration du processus de déploiement du Web-SIG dans la chaîne de production

Etant donné que l'entreprise livre plusieurs projets par jour, il est primordial que l'ajout d'un nouveau livrable dans la chaîne de production soit automatisé afin de maintenir les rendements de l'équipe de production.

Cette automatisation passe par deux grandes étapes :

- La création des trois fichiers de configuration indispensables pour chaque projet :
  - Le fichier de configuration JSON utile à l'application Web
  - Le premier fichier de configuration YAML utile au serveur de pré calcul
  - Le deuxième fichier de configuration YAML utile au serveur de pré calcul
- L'ordinateur hébergeant tous ces serveurs (cartographique et de pré calcul) et l'application Web étant distant (serveur Web), il faut pouvoir uploader le projet avec ses fichiers de configuration dans les répertoires adéquats de manière rapide. De plus, une fois uploadés, il faut lancer les opérations de pré calcul.

Le but final est d'obtenir une URL pour chaque projet qui a la forme : [http://sig.undermap.me/identifiant du projet](http://sig.undermap.me/identifiant_du_projet), où « identifiant\_du\_projet » est un identifiant unique permettant d'identifier un projet.

### 4.1 Création automatique des fichiers de configuration

Pour la réalisation de cette étape, nous avons développé un plugin QGIS. En un clic, les 3 fichiers de configuration sont générés.

#### 4.1.1 Fichier de configuration JSON utile à l'application Web

Ce fichier contient l'architecture de l'application Web. Grâce à lui, l'application peut connaître :

- L'image aérienne à afficher : si le projet est dans une ville possédant une ortho photo en open data, il l'utilisera ; Sinon, il utilisera l'ortho photo de l'IGN
- Sur quelle zone ouvrir l'application de manière automatique (grâce à l'emprise du projet)
- Tous les types de réseaux présents

- Tous les réseaux présents et les types de réseau auxquels ils appartiennent
- Le lien du projet dans le serveur cartographique
- Le lien du projet dans le serveur de pré calcul
- L'identifiant d'une couche (réseau) : c'est avec cet identifiant, qu'il pourra trouver les données de la couche dans le serveur cartographique et de pré calcul

Toutes ces informations sont recueillies dans le projet QGIS de la production par le biais de la bibliothèque python de QGIS : PYQGIS. Elles sont ensuite écrites dans le fichier de configuration avec le langage informatique python. Ce fichier a comme nom l'identifiant du projet.

Par exemple :

- Pour connaître l'emprise du projet :

```
emprise = layer.extent() ;// layer étant la couche de l'emprise du projet
```

- Pour connaître tous les réseaux, on recherche toutes les couches du groupe RSX :

```
Tous_les_reseaux = QgsProject.layerTreeRoot().findGroup('RSX').findLayers()
```

- Pour connaître l'identifiant et le nom des réseaux : on fait une boucle dans les couches du groupe RSX, et pour chaque réseau on récupère son identifiant et son nom :

```
for layer in QgsProject.layerTreeRoot().findGroup('RSX').findLayers():
    identifiant = layer.id()
    nom = layer.name()
```

Un exemple de fichier est disponible en Annexe 4.

#### 4.1.2 Premier fichier de configuration du serveur de pré calcul

Il liste toutes les couches avec leurs sources respectives et indique au serveur comment mettre en cache (stocker) les données résultant du pré calcul de chaque couche.

Une source de données pour une couche se définit par :

- Le lien vers le projet du serveur cartographique qui la contient
- Son identifiant dans ce projet
- Le Web service par lequel on souhaite que la donnée soit récupérée avant d'être pré calculée : dans notre cas, il s'agit du WMS car il permet de conserver le style des données

Un cache d'une donnée est quant à lui défini par :

- Sa source : une source parmi la liste des sources
- Son format de stockage : nous avons utilisé le format MBTILES. C'est une base de données SQLITE qui stocke toutes les images pré calculées d'une donnée
- Le dossier dans lequel on souhaite enregistrer le cache

Enfin, une couche se définit par :

- Son cache : un cache parmi la liste des caches. Lorsque cette couche sera demandée par le visualiseur, le serveur ira immédiatement lire le cache et rendra l'image demandée.
- Un titre pour cette donnée dans le serveur

Toutes ces informations sont recueillies avec la bibliothèque python de QGIS, et sont ensuite écrites dans un fichier avec le langage informatique python.

Un exemple de fichier est disponible en Annexe 2

#### 4.1.3 Deuxième fichier de configuration du serveur de pré calcul

Il spécifie quelle couche parmi celles définies dans le premier fichier on souhaite pré calculer. Ceci se fait par des opérations que l'on donne au serveur. Chaque opération est donc une commande qui indique au serveur quelle couche il va pré calculer.

Outre définir la couche que l'on souhaite pré calculer, il permet de définir de surcroît les niveaux de zooms souhaités.

Ce sont ces niveaux de zooms couplés à l'emprise de la couche qui détermineront le temps que l'opération prendra et le poids des données résultantes.

Le poids d'une image dépend de la présence plus ou moins importante de réseaux sur celle-ci. Ce poids peut varier entre 1 KO et 8 KO :

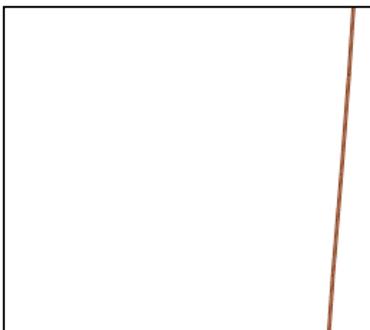


Figure 13: image avec une faible densité de réseaux, 2.47KO

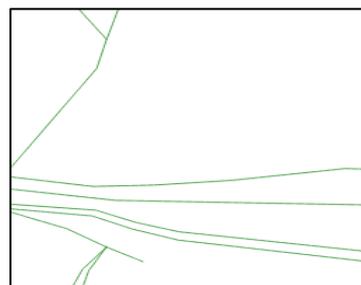


Figure 12: image avec une densité de réseaux moyenne, 4 KO

D'autre part, plus on s'oriente vers d'importants niveaux de zooms, plus on a d'images de stockage. Pour une surface d'environ 40 000 m<sup>2</sup>, on obtient les résultats suivant pour une couche:

Tableau 2 : estimation du nombre d'images pré calculées par niveau de zoom sur 40 000m<sup>2</sup>

Niveau de zoom	Nombre d'images
16	48
17	64
18	160
19	576
20	1904
21	6336
22	23552
23	94208

Pour des raisons d'estimation rapide, on a considéré qu'une image pèse en moyenne 3 KO. Nous ne compterons pas les niveaux inférieurs à 16 car ils sont peu significatifs (peu d'images produites, i.e. moins de 30)

L'estimation en terme de couple poids / temps de calcul par couche, nous donne:

- Option 1 : du niveau 16 à 23, on obtient 380 MO pour 11 heures de calcul.
- Option 2 : du niveau 16 à 21, on obtient 27 MO pour 44 minutes de calcul.
- Option 3 : du niveau 16 à 19, on obtient 2,55 MO pour une minute de calcul.

En prenant en compte qu'il est nécessaire de multiplier ces chiffres par le nombre de réseaux d'un projet (variant entre 4 et 30).

Après plusieurs tests en concert avec l'entreprise et certains clients, l'option 3 a été choisie. Il a été décidé que :

- Jusqu'au niveau de zoom 19, on affiche les images tuilées : on conserve une bonne fluidité et le même style que celui de la production.
- A partir du niveau 20, on affiche les données brutes (vecteurs) : étant donné que l'on couvre de petites surfaces, les données ne seront pas volumineuses ce qui nous permettra de garder une fluidité acceptable. En contrepartie, on perd le style de la production. Pour pallier ce problème de style, nous avons créé un style similaire à celui de la production grâce à OPENLAYERS.

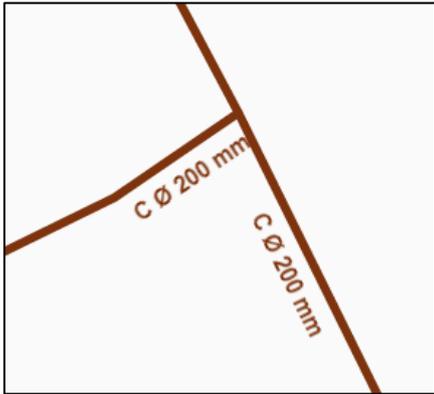


Figure 15: réseau avec le style d'OPENLAYERS

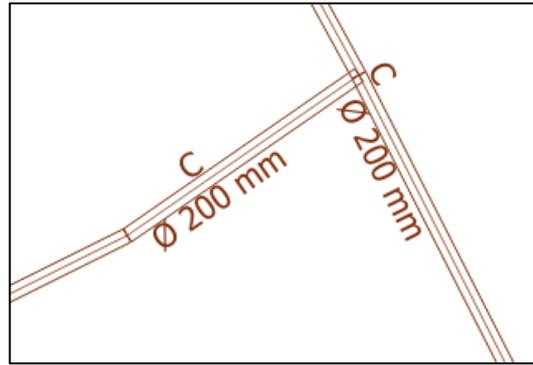


Figure 14: Réseau avec le style de production (QGIS)

Un exemple de fichier est disponible en

Annexe 3.

#### 4.2 Déploiement optimisé d'un plan en ligne

Une fois les fichiers de configuration générés, il faut les uploader avec les données du projet dans le serveur en ligne.

Il est à noter que chaque projet possède un identifiant unique.

Cette automatisation demande à définir certaines règles dans l'ordinateur distant :

- Avoir un dossier unique où l'application Web pourra aller chercher le fichier de configuration dont il a besoin : grâce à l'identifiant du projet, il pourra trouver ce fichier qui est nommé avec ce même identifiant.
- Avoir un dossier unique où QGIS SERVER pourra trouver le projet QGIS correspondant au projet : grâce à l'identifiant du projet, QGIS SERVER pourra trouver le projet QGIS car il porte l'identifiant du projet.
- Un dossier unique où le serveur de pré calcul pourra trouver les deux fichiers de configuration liés au projet : à partir de l'identifiant du projet, il peut trouver :
  - Le premier fichier car ce fichier est nommé par la formule : **identifiant\_du\_projet\_config.yaml**
  - Le deuxième fichier car ce fichier est nommé par la formule : **identifiant\_du\_projet\_seed.yaml**

Pour que ces règles soit respectées, il faut que l'identifiant de chaque projet soit bel et bien unique, au risque sinon d'écraser d'autres fichiers.

Afin d'uploader les fichiers tout en garantissant ces règles, nous avons créé un mini FTP personnalisé sur un site Web. Il permet de :

- Uploader en une fois tous les fichiers d'un projet tout en vérifiant la nomenclature et la structure de tous les fichiers.
- Vérifier que le projet qu'on uploade a un identifiant qui n'existe pas encore.
- Lancer et permettre de suivre l'évolution du pré calcul

Le fonctionnement de ce FTP est expliqué dans le diagramme ci-dessous :

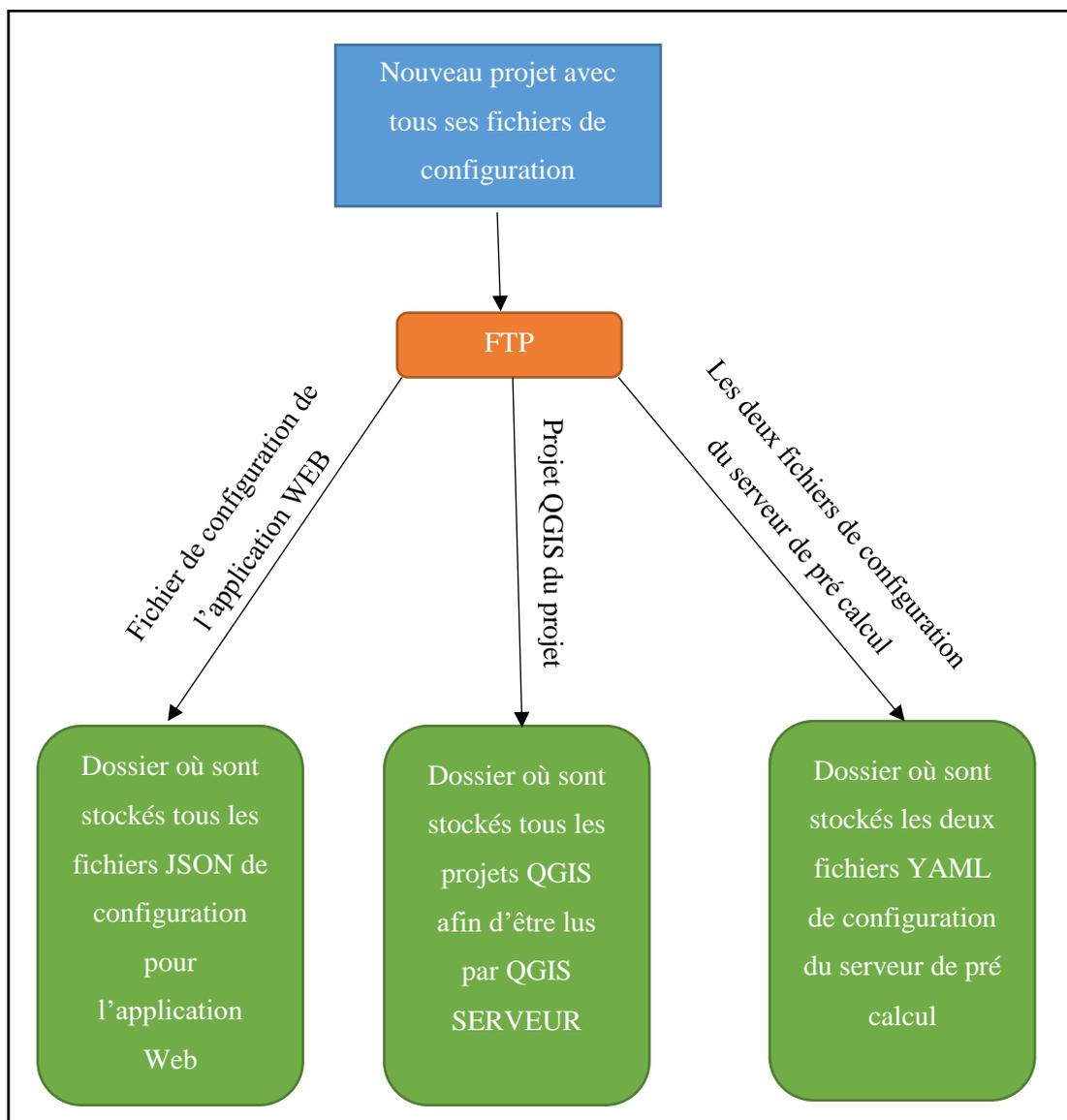


Figure 16: Schéma de fonctionnement du FTP d'UNDERMAP

Une fois tous les fichiers transférés, le FTP lancera automatiquement les opérations de pré calcul. L'évolution du pré calcul peut être consultée depuis le FTP.

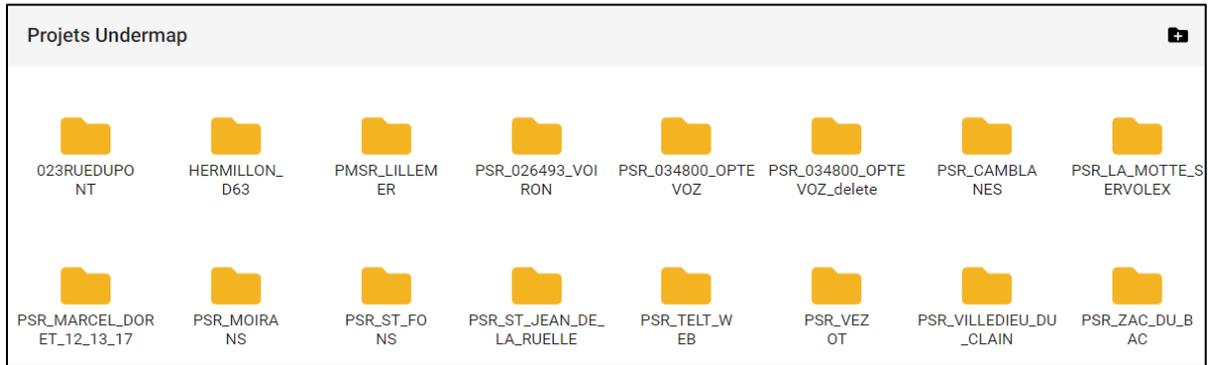


Figure 17: Interface principale du FTP qui liste tous les projets disponibles

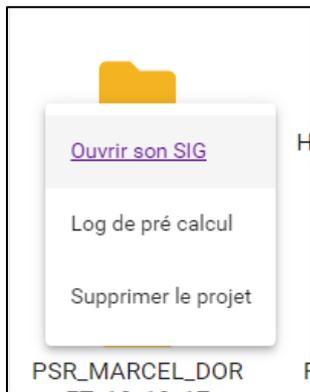


Figure 18: Actions possible sur un projet

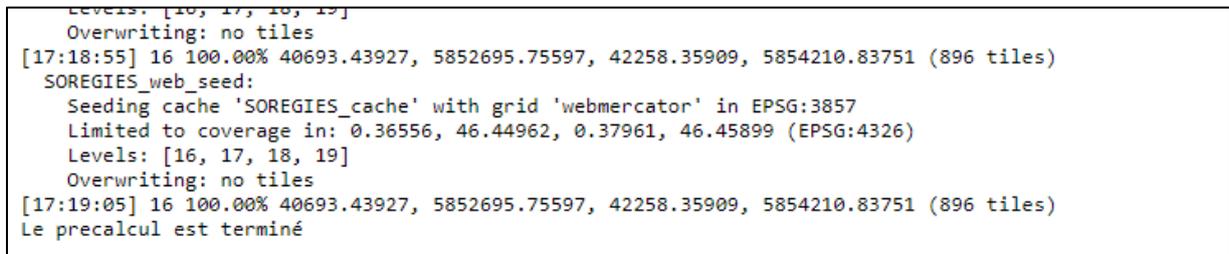


Figure 19: Log ou suivi du pré calcul d'un projet sur le FTP



Figure 20: Interface pour uploader un projet UNDERMAP

## 5 Cas pratiques et analyse

Plus de cinquante projets ont été réalisés jusqu'ici. Nous présenterons deux retours d'expérience sur deux projets pertinents :

- Le plus grand des projets traités : projet pour le bureau d'études EGIS concernant l'extension du tramway T6 de Lyon
- Un petit projet d'un client particulier : il s'agit d'un client ayant en moyenne 6 projets par semaine, sans fond de plan. La production utilise pour cette typologie de projet le cadastre en tant que fond de plan.

### 5.1 Projet de tramway de 7 kilomètres à Lyon

Il s'agit d'un ensemble de huit différents projets formant la globalité du plan. Malgré cette division en 8 de la production du livrable, il s'agissait de livrer un seul projet Web.

Pour ce faire, nous avons créé un nouveau plugin qui permet de jumeler plusieurs projets en un seul.

Une fois le projet unique créé, ces caractéristiques sont les suivantes :

- 8 emprises pour une superficie totale à pré calculer de près 9 km<sup>2</sup>
- 7 km de linéaire de voirie
- 6 types de réseaux: électricité, assainissement, gaz, télécommunication, eau potable et chauffage
- Un fond de plan (100 MB) en fichier vectoriel (SHAPEFILE)
- 31 réseaux

Il a donc été nécessaire de pré calculer 31 réseaux ainsi qu'un fond de plan de grande envergure sur une superficie de 9 km<sup>2</sup>. Les prés calculs ont donné pour chaque réseau un cache de 150 MB ainsi que 160 MB pour le fond de plan, soit un total de 4,810 GB.

Se situant dans la ville de Lyon, nous avons utilisé l'image aérienne en open data de la ville qui affiche 6 cm de résolution.

Une impression issue de la plateforme est disponible en Annexe 7. Quelques images du résultat sur l'application ci-dessous :

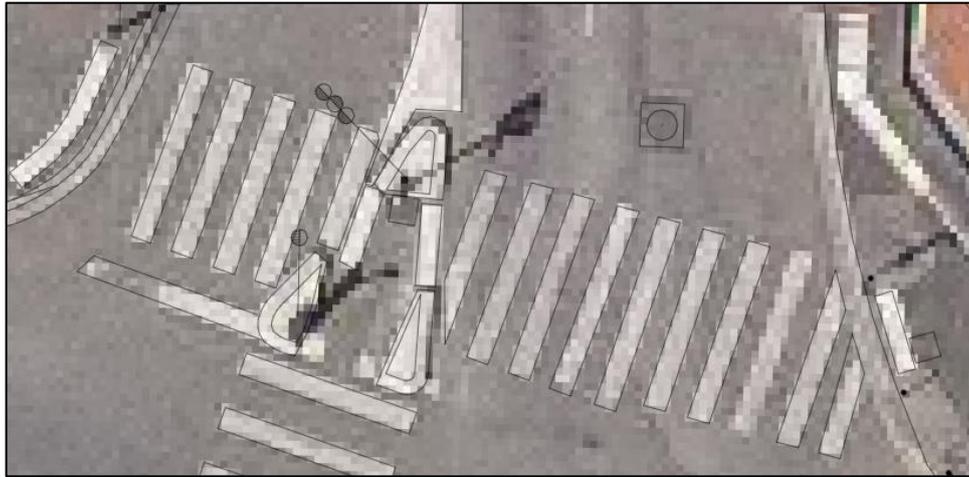


Figure 21: Fond de plan du projet superposé à l'image aérienne de la ville de Lyon

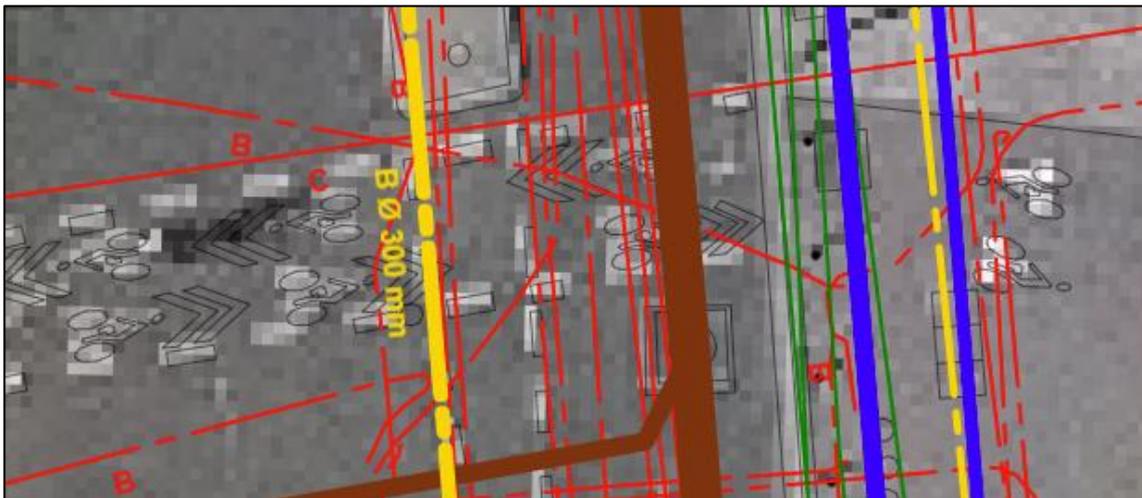


Figure 23: Image aérienne de la ville de Lyon, fond de plan et réseaux du projet



Figure 22: Fond de plan et réseaux du projet

## 5.2 Projet de faible envergure avec fond de plan cadastral

Ce type de projet permet de vérifier que l'infrastructure mise en place est « scalable » : capable de gérer de petits, moyens et grands projets.

Dans le souci d'améliorer toujours la chaîne de production, nous avons développé un plugin qui permet d'extraire le cadastre à partir d'une emprise sur QGIS. Le cadastre Français a d'abord été inséré dans une base de données PostgreSQL/POSTGIS, puis indexé spatialement. Aujourd'hui, l'équipe de production est capable en quelques secondes d'obtenir le cadastre sur une emprise donnée.

Ce projet a pour caractéristiques :

- Une emprise d'environ 0,16 km<sup>2</sup>
- 500 m de long
- 5 types de réseaux: électricité, assainissement, gaz, télécommunication et eau potable
- Un fond de plan (cadastre) qui pèse 140 KB en fichier vectoriel (SHAPEFILE)
- 8 réseaux

Nous avons donc pré calculé 8 réseaux et un fond de plan cadastral sur une superficie de 0,16 km<sup>2</sup>. Les pré calculs ont donné pour chaque réseau un cache de 500 KB et 6 MB pour le fond de plan, soit un total de 10 MB en une minute et demi seulement.



Figure 24: vue globale du projet

Finalement, le client peut visualiser son plan de manière interactive dans son espace client, sur la plateforme Web UNDERMAP, en cliquant sur l'onglet Plan :

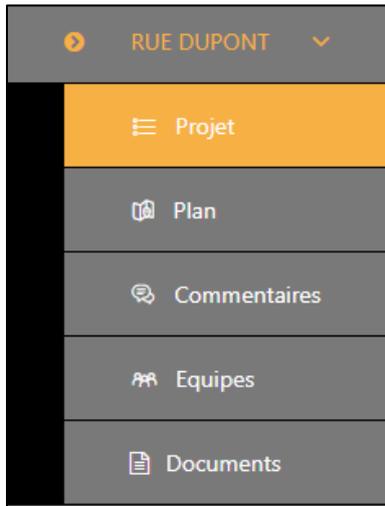


Figure 25: espace client sur la plateforme UNDERMAP

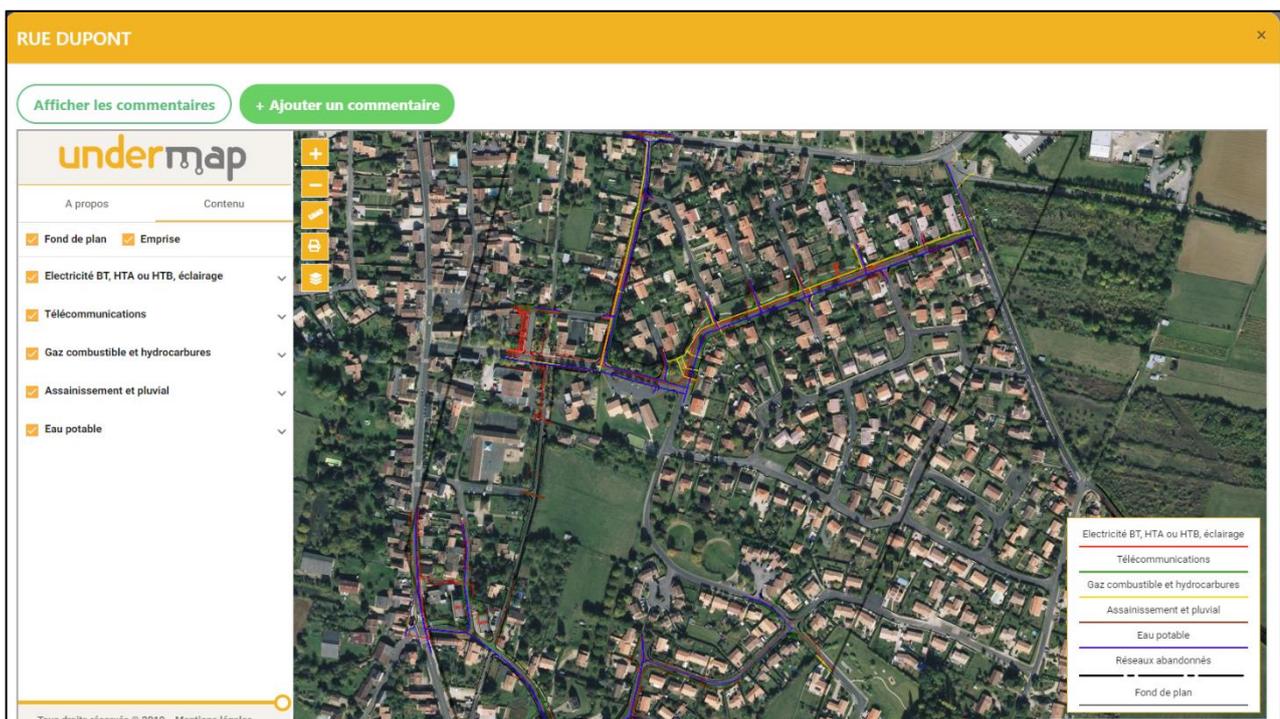


Figure 26: visualisation interactive du plan dans l'espace client

## 6 Conclusion générale et perspectives

Au cours de la présente étude, plusieurs développements ont été réalisés dans le but d'obtenir un WEB-SIG permettant de visualiser et exploiter les plans produits par le service UNDERMAP, tout en assurant la plus grande fluidité dans la chaîne de production.

En premier lieu, nous avons défini le service UNDERMAP et les différentes étapes de production d'un livrable de type « Plan de Synthèse de Réseaux » depuis la commande d'un client jusqu'à la restitution des livrables.

En second lieu, nous avons décrit comment développer un WEB-SIG centré sur les plans de réseaux produits sous QGIS. Pour ce faire, nous avons réalisé un état de l'art concernant les serveurs cartographiques et explicité leur fonctionnement. A l'issue d'une comparaison des différents serveurs cartographiques les plus répandus, nous avons choisi QGIS SERVER qui nous donne la possibilité d'épouser la chaîne de production, déjà optimisée sous QGIS. Outre un serveur cartographique, nous avons mis en place un serveur de pré calcul visant à minimiser le temps de réponse de chargement des données sur le Web.

Grâce à ces deux serveurs, nous avons pu développer un visualiseur de plans sur le Web, composé de trois grandes parties : un menu permettant d'afficher et/ou masquer des réseaux sur la carte, un bloc de carte pour visualiser et interagir avec les données, et un menu d'impression offrant la possibilité d'imprimer des plans sur le Web.

Dans une troisième phase, nous nous sommes intéressés à l'optimisation de la publication d'un projet UNDERMAP sur notre WEB-SIG. Un procédé automatique permettant la création des fichiers de configuration liés à un projet, puis l'upload des données sur un ordinateur distant, a été implémenté. Une fois sur l'ordinateur distant, les différents serveurs et le visualiseur interprètent ces fichiers et publient le projet. Pour assurer la fluidité de la chaîne de production, ce processus est exécutable aisément par le biais d'un plugin.

Finalement, nous avons pu analyser les performances de notre système à travers deux projets particuliers. Il s'agit d'un projet de grande envergure (7 km) concernant l'extension d'un tramway de Lyon, comprenant 31 exploitants de réseaux, et un projet de très faible étendue, ayant pour fond de plan le cadastre.

UNDERMAP, au terme de cette étude, est désormais en mesure de publier ses plans sur le Web, et d'offrir ainsi à ses clients une réelle valeur ajoutée via l'exploitation des données dans ce nouveau type de livrable.

L'entreprise ayant déjà entamé le développement d'une application mobile dédiée aux chantiers concernés par les réseaux enterrés, il s'agit maintenant d'adapter le système développé dans notre étude afin de répondre aux besoins du terrain.

C'est dans ce contexte et par le biais du contrat à durée indéterminée proposé par l'entreprise, qu'il nous sera possible de continuer l'étude et d'atteindre cet objectif final.

## 7 Remerciements

Tout d'abord, je remercie vivement les deux dirigeants de la société FUTURMAP, M. Luc STRAUB et M. Florent MICHELS, qui m'ont accueilli et suivi au sein de leur structure.

Mes remerciements vont ensuite tout particulièrement à mon tuteur de stage, M. Loïc ELSHOLZ, dont la disponibilité, la patience et le concours en général ont très largement contribué à la mise en œuvre de cette étude.

Je tiens également à remercier l'ensemble des collaborateurs de la société pour leur accueil, et disponibilité qui m'ont permis de mener à bien mes études.

Enfin, je remercie ma famille pour son inébranlable soutien, depuis le début de mes études.

## 8 Bibliographie

Le site de GeoServer <http://geoserver.org/> - le 15 avril 2019

Le site de MapServer <https://mapserver.org/> - le 15 avril 2019

Le site sur les Web services géographique de l'IGN <https://geoservices.ign.fr/documentation/geoservices/> - le 16 avril 2019

Le guide d'installation de QGIS SERVER <https://www.qgis.org/fr/site/forusers/alldownloads.html> - le 8 Mai 2019

Le guide d'utilisation de QGIS SERVER [https://docs.qgis.org/3.4/en/docs/training\\_manual/qgis\\_server/index.html](https://docs.qgis.org/3.4/en/docs/training_manual/qgis_server/index.html) - le 8 mai 2019

La documentation du langage informatique python <https://docs.python.org/3/> - le 9 mai 2019

La documentation de la bibliothèque python pour QGIS <https://qgis.org/api/> - le 15 mai 2019

Le guide de MapProxy <https://mapproxy.org/docs/nightly/install.html> - le 4 juin 2019

La documentation du format MBTILES <https://gdal.gloobe.org/gdal/formats/mbtiles.html> - le 5 juin 2019

La documentation du format de fichier YAML <https://yaml.org/> - le 5 juin 2019

La documentation du format JSON <https://www.json.org/> - le 5 juin 2019

Le site de la bibliothèque OPENLAYERS <https://openlayers.org/> - le 26 juin 2019

Le site de l'open data de la ville de Lyon <https://data.beta.grandlyon.com/fr/accueil> - le 27 juin 2019

Le site des Web services géographique de Google <https://cloud.google.com/maps-platform/maps> - le 27 juin 2019

## 9 Glossaire

**CSS** : Cascading Style Sheets. C'est un langage informatique qui décrit la présentation des documents HTML.

**DAO** : Dessin assisté par ordinateur.

**DICT** : déclaration d'intention de commencement de travaux

**DT** : déclaration de projet de travaux

**DXF** : Drawing eXchange Format. C'est un format de fichier créé par la société Autodesk servant à échanger des fichiers DAO ou CAO entre systèmes CAO n'utilisant pas le même format de fichier natif

**FDP** : Fond de plan

**FTP** : File Transfer Protocol. C'est un protocole de communication destiné au partage de fichiers.

**HTML** : Hypertext Markup Language. Langage informatique pour écrire des pages WEB

**JSON** : JavaScript Object Notation. C'est un format de données textuelles dérivé de la notation des objets du langage JavaScript

**KML** : Keyhole Markup Language. C'est un langage fondé sur le formalisme XML et destiné à la gestion de l'affichage de données géospatiales dans les logiciels de SIG

**PYQGIS** : Python pour QGIS

**QGIS** : Logiciel de SIG

**RSX** : Réseaux

**SHP** : ShapeFile

**SIG** : Système d'information géographique

**SQLITE** : C'est une bibliothèque qui propose un moteur de base de données relationnelle accessible par le langage SQL

**WEB** : World Wide Web.

**WEB-SIG** : ce sont des applications qui permettent de visualiser des cartes sur le WEB

**WFS** : WEB Feature Service

**WMS** : WEB Map Service

**WMTS** : WEB Map Tile Service

**XML** : Extensible Markup Language

**YAML**: Yet Another Markup Language. C'est un format de représentation de données par sérialisation Unicode

## 10 Annexes

Annexe 1:Fonctionnalités du plugin UNDERMAP .....	38
Annexe 2:Exemple du premier fichier de configuration du serveur de pré calcul .....	39
Annexe 3:Exemple du deuxième fichier de configuration du serveur de pré calcul .....	40
Annexe 4:Exemple de fichier JSON de configuration de l'application WEB .....	41
Annexe 5: Architecture du dossier d'un projet UNDERMAP .....	43
Annexe 6:Algorithme de la production d'un projet UNDERMAP .....	44
Annexe 7:Exemple d'impression sur la plateforme .....	47

Annexe 1: Fonctionnalités du plugin UNDERMAP

Fonctionnalité	Description	Remarque
Initialiser un projet	Lis le dossier du client pour trouver tous les exploitants de réseaux afin de créer l'arborescence du projet	Etait déjà fait
Ajouter un exploitant	Permet d'ajouter un exploitant (couche) dans un projet	Etait déjà fait
Initialiser un fond de plan	Ajouter le fond de plan du client sous des formats SIG et DAO	Amélioré dans le cadre de la mission
Initialiser une emprise	Ajouter l'emprise du client sous des formats SIG et DAO	Amélioré dans le cadre de la mission
Fusionner des entités	Regroupe les entités ayant les mêmes attributs et se touchant uniquement à leurs extrémités	Conçu dans le cadre de la mission
Exporter les couches en GEOJON	Ces GeoJson seront ensuite intégrés dans un logiciel de DAO	Amélioré dans le cadre de la mission
Exporter le rapport de production	Un fichier Excel qui contiendra toutes les longueurs, les classes et tous les types de réseaux du projet.	Amélioré dans le cadre de la mission
Générer les fichiers de configuration du projet	Génère les 3 fichiers de configuration indispensables pour l'application Web et les serveurs	Conçu dans le cadre de la mission
Importer un DXF dans un réseau	Transformer un fichier de DAO vers SIG	Conçu dans le cadre de la mission
Extraire le cadastre	Extraire le cadastre sur une emprise, et l'ajouter au projet	Conçu dans le cadre de la mission
Exporter le cadastre	Export du cadastre en DAO (DXF) pour qu'il puisse être intégré dans AUTOCAD	Conçu dans le cadre de la mission

*Annexe 2: Exemple du premier fichier de configuration du serveur de pré calcul*

Pour un projet qui a pour identifiant `psr_rue_du_pont`, situé dans le répertoire `/var/www`, qui a un réseau nommé ORANGE, ce fichier ressemble à :

```
sources :
  ORANGE:
    coverage:
      bbox:
        - -55742.46216226926
        - 5585101.314662504
        - -54429.29926984115
        - 5586447.897953028
      srs: EPSG:3857
    req:
      layers: ORANGE
      transparent: true
      url: http://qgis.demo/cgi-bin/qgis_mapserv.fcgi?SERVICE=WMS&VERSION=1.3.0&map=/var/www/psr_rue_du_pont.qgs
    supported_srs:
      - EPSG:4326
      - EPSG:3857
    type: wms
caches:
  ORANGE_cache:
    cache:
      - filename: /var/www/mapproxy/data/ORANGE_cache.mbtiles
      - type: mbtiles
    grids:
      Webmercator
    sources:
      ORANGE
layers:
  name: ORANGE_Web
  sources:
    ORANGE_cache
  title: ORANGE
```

*Annexe 3: Exemple du deuxième fichier de configuration du serveur de pré calcul*

Pour un projet qui a pour identifiant `psr_rue_du_pont`, situé dans le répertoire `/var/www`, qui a un réseau nommé ORANGE, ce fichier ressemble à :

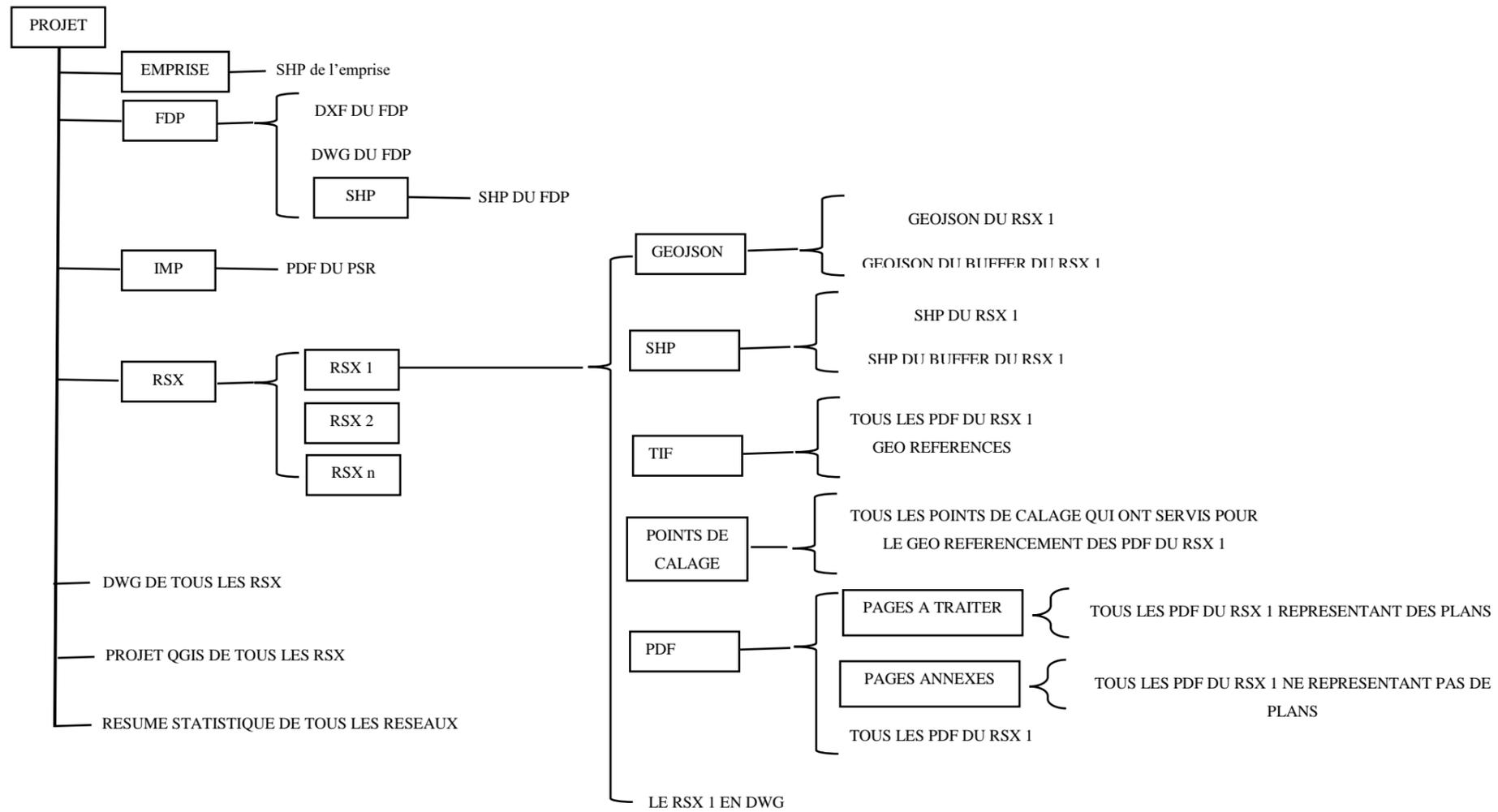
```
layers:
  name: ORANGE_Web
  sources:
    ORANGE_cache
coverages:
  roi:
    bbox:
      - -55742.46216226926
      - 5585101.314662504
      - -54429.29926984115
      - 5586447.897953028
    srs: EPSG:3857
seeds:
  ORANGE_Web_seed:
    caches:
      ORANGE_cache
    coverages:
      roi
    grids:
      Webmercator
    levels:
      from: 16
      to: 19
```

*Annexe 4: Exemple de fichier JSON de configuration de l'application WEB*

```
{
  "prefix_url": "http://xxx.xxx.xxx/023RUEDUPONT_config/",
  "url_qgis_serveur": "http://xxx.xxx.me/cgi-xxx/qgis_mapserv.fcgi?SERVICE=WMS&VERS
  ION=1.3.0&map=/var/www/projet_qgis/023RUEDUPONT/023RUEDUPONT_Web.qgz",
  "identifiant_emprise_wfs": "emprise",
  "reseaux": [
    {
      "nom": "Electricité BT, HTA ou HTB, éclairage",
      "couleur": "#ff0000",
      "reseaux": [
        {
          "nom": "SRD",
          "extent": [
            40814.19037398486,
            5852779.206042461,
            41986.008384831,
            5854032.729889923
          ],
          "type_reseau": "ELEC",
          "identifiant_tms": "SRD_Web",
          "identifiant_wfs": "SRD",
          "identifiant_wms": "SRD_b2290df4_0c10_4e20_b55c_0e828d67343d",
          "z_wmts": 19
        }
      ]
    },
    {
      "nom": "Télécommunications",
      "couleur": "#008000",
      "reseaux": [
        {
          "nom": "DEPARTEMENT DE LA VIENNE",
          "extent": [
            40811.67065675649,
```

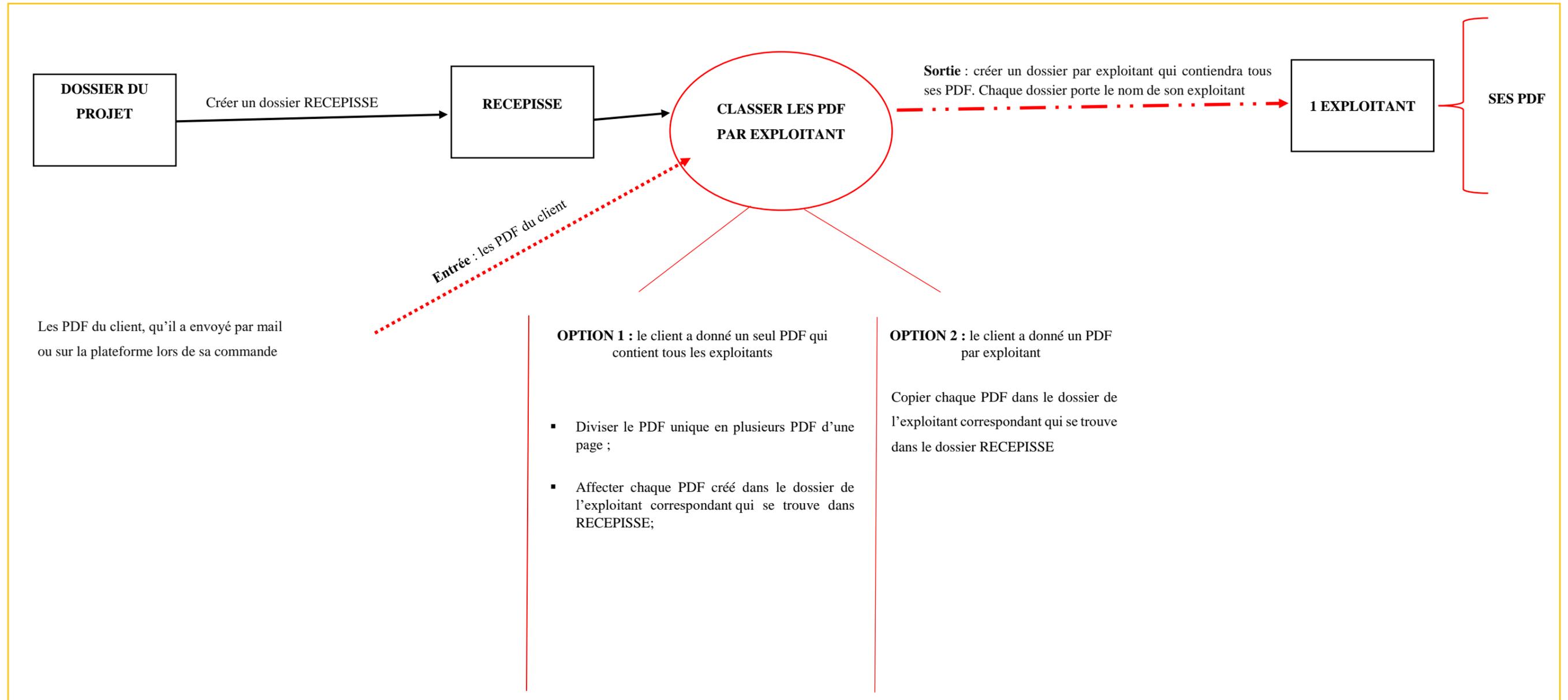
```
        5852763.798612825,  
        41741.144384015104,  
        5854027.03611214  
    ],  
    "type_reseau": "TEL",  
    "identifiant_tms": "DEPARTEMENTDELAVIENNE_Web",  
    "identifiant_wfs": "DEPARTEMENT DE LA VIENNE",  
    "identifiant_wms": "DEPARTEMENT_DE_LA_VIENNE_b62b5720_496d_492  
c_9a33_f44af346229e",  
    "z_wmts": 19  
    }  
    ]  
    }  
],  
"identifiant_fdp_wms": "FDP_VILLEDIEU_6c9c6b5d_a5b9_4053_be3c_9bc4d6a05904",  
"identifiant_fdp_wfs": "FDP_VILLEDIEU",  
"identifiant_fdp_tms": "FDPVILLEDIEU_Web",  
"fdp_z_wmts": 21,  
"extent": [  
    40693.439265379006,  
    5852695.755971294,  
    42258.35908831721,  
    5854210.837511972  
    ]  
}
```

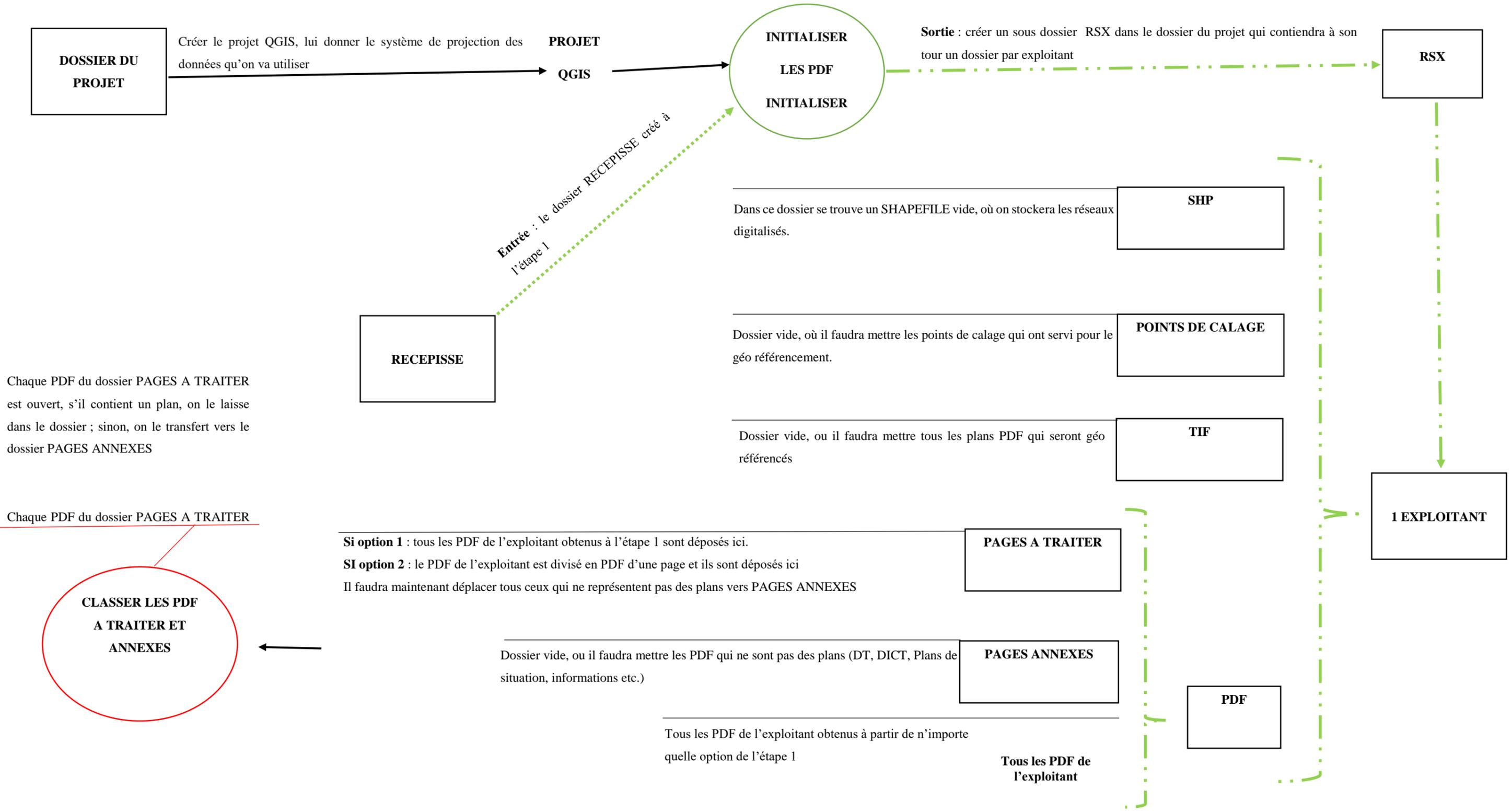
Annexe 5: Architecture du dossier d'un projet UNDERMAP

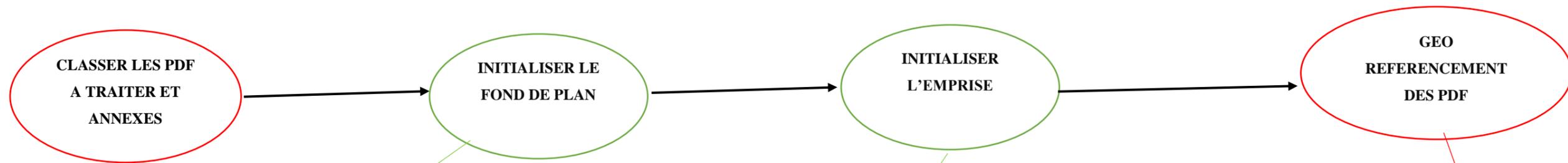


Annexe 6: Algorithme de la production d'un projet UNDERMAP

Les rectangles représentent des dossiers. Les cercles des opérations. La couleur rouge représente des opérations faites manuellement et celles en vert, les opérations faites de manière automatique (le plugin).







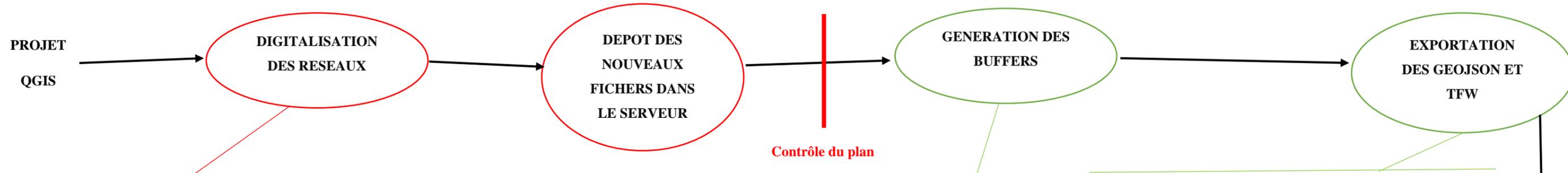
Si le client a fourni un fond de plan, on l'ajoute au projet QGIS. C'est en général un plan DAO vectoriel (DWG ou DGN).

- Plan vectoriel : le fichier est converti en DXF et directement ajouté au projet, un style de QGIS est appliqué à la volée selon le champ « layer ».
- Plan raster : il sera géo référencé avant d'être ajouté au projet QGIS

Cette étape dépend de la manière dont le client a passé sa commande :

- Sur le site WEB : dans ce cas on a un fichier vectoriel qui pourra être ajouté au projet QGIS.
- Par mail : si le fichier transmis par le client est un fichier vectoriel, on l'ajoute directement au projet ; sinon, on digitalisera l'emprise pour l'ajouter au projet QGIS.

Ici, les dessinateurs géo référencent les plans PDF exploitant par exploitant



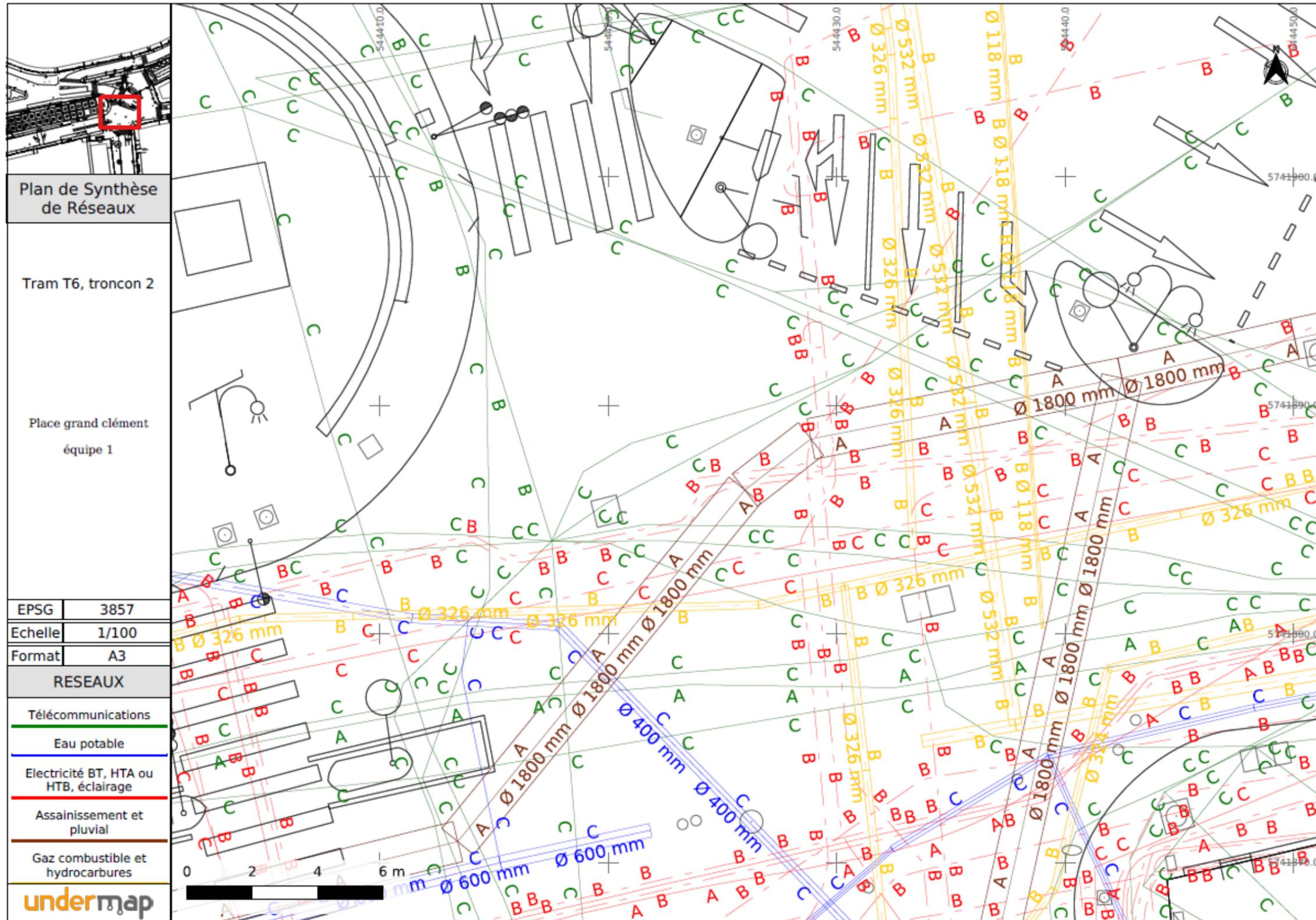
Contrôle du plan

Sur les plans géo référencés, les réseaux sont digitalisés et enregistrés dans les SHAPEFILE se trouvant dans le dossier SHP de chaque exploitant

Sur chaque ligne de réseau, une zone tampon est créée en fonction de son diamètre et de sa classe. Les résultats sont stockés dans le dossier SHP de chaque exploitant.

Chaque donnée SIG est transformée dans un format lisible par AUTOCAD grâce à une routine interne :

- Vecteur : les SHAPEFILE sont transformés en GEOJSON;
- Les rasters (tif) : pour être lus, ils ont besoin d'un fichier de calage, qui définit les dimensions de leurs pixels, le point d'origine et la rotation de l'image (fichier TFW)





## Résumé

Cette étude s'inscrit dans les objectifs du service UNDERMAP de livrer ses plans de synthèse de réseaux via une carte interactive sur le WEB. Cette carte doit d'une part permettre de visualiser de manière pertinente tous les réseaux présents sur l'emprise d'un projet, et d'exploiter les données (obtenir des informations attributaires, afficher/masquer, mesurer, imprimer etc.).

Les serveurs cartographiques permettant de fournir des Web services à un visualiseur de carte, nous avons examiné plusieurs de ces serveurs et choisi finalement QGIS Serveur pour sa compatibilité avec la chaîne de production existante. Ensuite, afin de minimiser les temps de chargement des données sur le Web, nous avons mis sur place un serveur de pré calcul. Pour finir, nous avons implémenté un visualiseur de données, qui interroge et interprète les différents Web services des deux précédents serveurs.

Aujourd'hui, le service UNDERMAP est capable de livrer à ses clients une exploitation de ses plans via WEB-SIG, tout en conservant la fluidité de sa chaîne de production.

**Mots clés :** WEB-SIG, Réseaux, DT-DICT, serveur cartographiques, QGIS, PyQgis, pré calcul de tuiles, chaîne de production.

## Abstract

This study contributes in the accomplishments of the objectives of UNDERMAP service, to deliver its network synthesis plans via an interactive map on the WEB. This map must firstly have the possibility to visualize in a relevant way all the networks present on the footprint of a project, and to exploit the data (obtain attribute information, display / hide, measure, print, etc.).

As the map servers provide Web services to a map viewer, we looked at several of these servers and finally chose QGIS Server for his compatibility with the existing production chain. Then, in order to minimize the loading times of the data on the Web, we set up a pre-calculation server. Finally, we implemented a data viewer, which interrogates and interprets the different Web services of the two previous servers.

Today, the UNDERMAP service is able to deliver to its customers an exploitation of its plans via WEB-SIG, while maintaining the fluidity of its production chain.

**Keywords:** WEB-SIG, network, cartographic server, QGIS, PyqGis, seeding, production line.