

# VisePoss

## Projet SIG3

Zwahlen Charlène et Buchser Jérémy  
26.01.2022



# Introduction- idée de base

- Aide au positionnement des stations (points fixes de la MO)
- Générer des **profils** à l'aide d'un modèle numérique 3D (modèle numérique de surface)
- Projet orienté dans le cas pratique des **visées optiques** avec instruments de mesures
- Deux fonctionnalités principales :
  - Station avec multi-visées
  - Visée simple

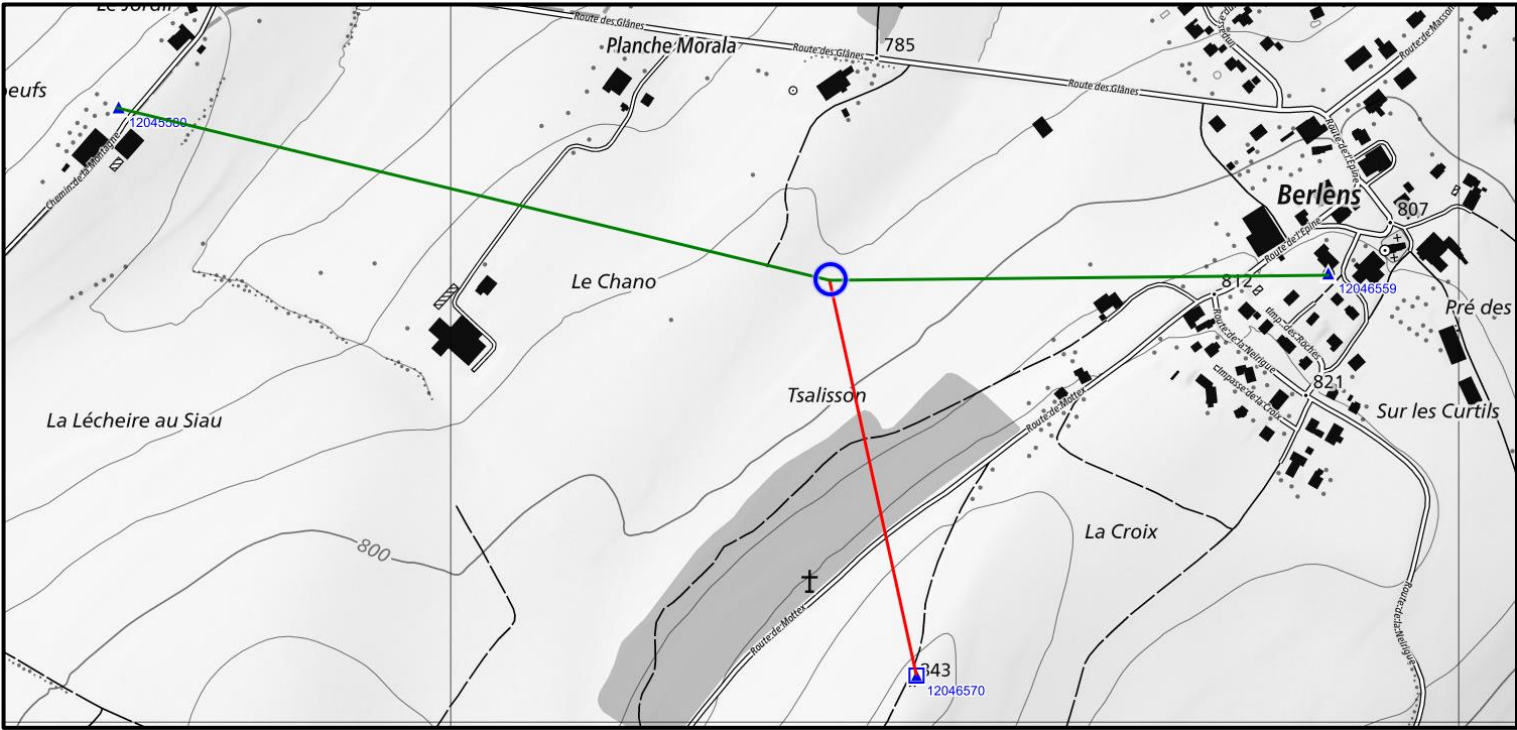
} *Finalement aucune spécification pour la visée simple*
- Communiquer avec une base de données

Fonctionnalité multi-visées

Hauteur d'instrument: 1.100 m

Hauteur de signal: 1.800 m

Distance de recherche: 2000 m



Coordonnée de la station: 2562376.0, 1171437.0

+ Enregistrer la station

→ Rapport PDF

	Numéro de point	Distance	Dénivelé	Est	Nord	Altitude	Visée possible
⊙	12046570	400.35	52.8	2562462.817	1171045.533	843.592	Non
⊙	12046559	495.03	43.2	2562871.255	1171444.291	834.027	Oui
⊙	12045580	727.39	-16.7	2561671.209	1171609.007	774.130	Oui

# Liste des fonctionnalités

Création d'une interface de base	✓	
Carte	✓	Plusieurs fonds de base
Renseigner hauteur cible et instrument	✓	
Rayon limite de recherche	✓	
Positionnement de notre station sur la carte	✓	
Récupération des données nécessaires	✓	
Points fixes, MNS	✓	
Analyse de visibilité	✓	
Tableau récapitulatif	✓	
Enregistrer la station	✓	Définition du nom de la station
Affichage des profils de visées	✓	
Exporter les stations (fichier .csv)	✓	
Importer des points avec fichier local	✗	Manque de temps
Impression rapport PDF	✗	Manque de temps
<i>Filtrer type des points à utiliser (PFP1-2-3)</i>	✓	
<i>Modifier hauteur de cibles et instrument</i>	✗	Manque de temps
<i>Mise à jour automatique des visées possibles</i>	✗	Limité par le temps des requêtes serveur (~6 sec pour 100m)
<i>Mise à jour du rapport</i>	✗	Pas de rapport PDF
<i>Supprimer des visées</i>	✗	Pas utile sans rapport
<i>Affichage temps-réel des mesures possibles</i>	✗	Limité par le temps des requêtes serveur (~6 sec pour 100m)
<i>Elements bonus</i>		

# Préparation des données de base

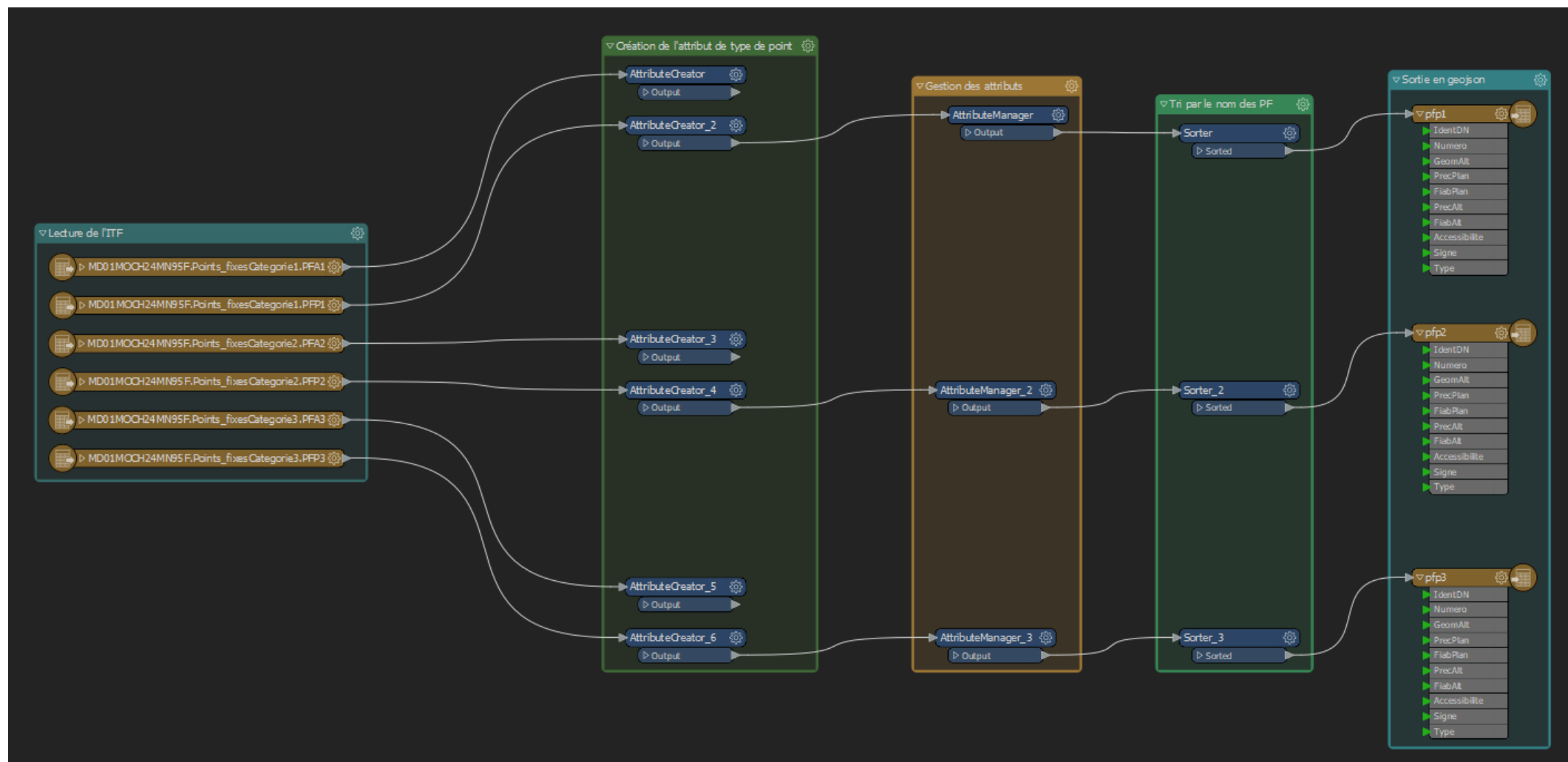
## MNS



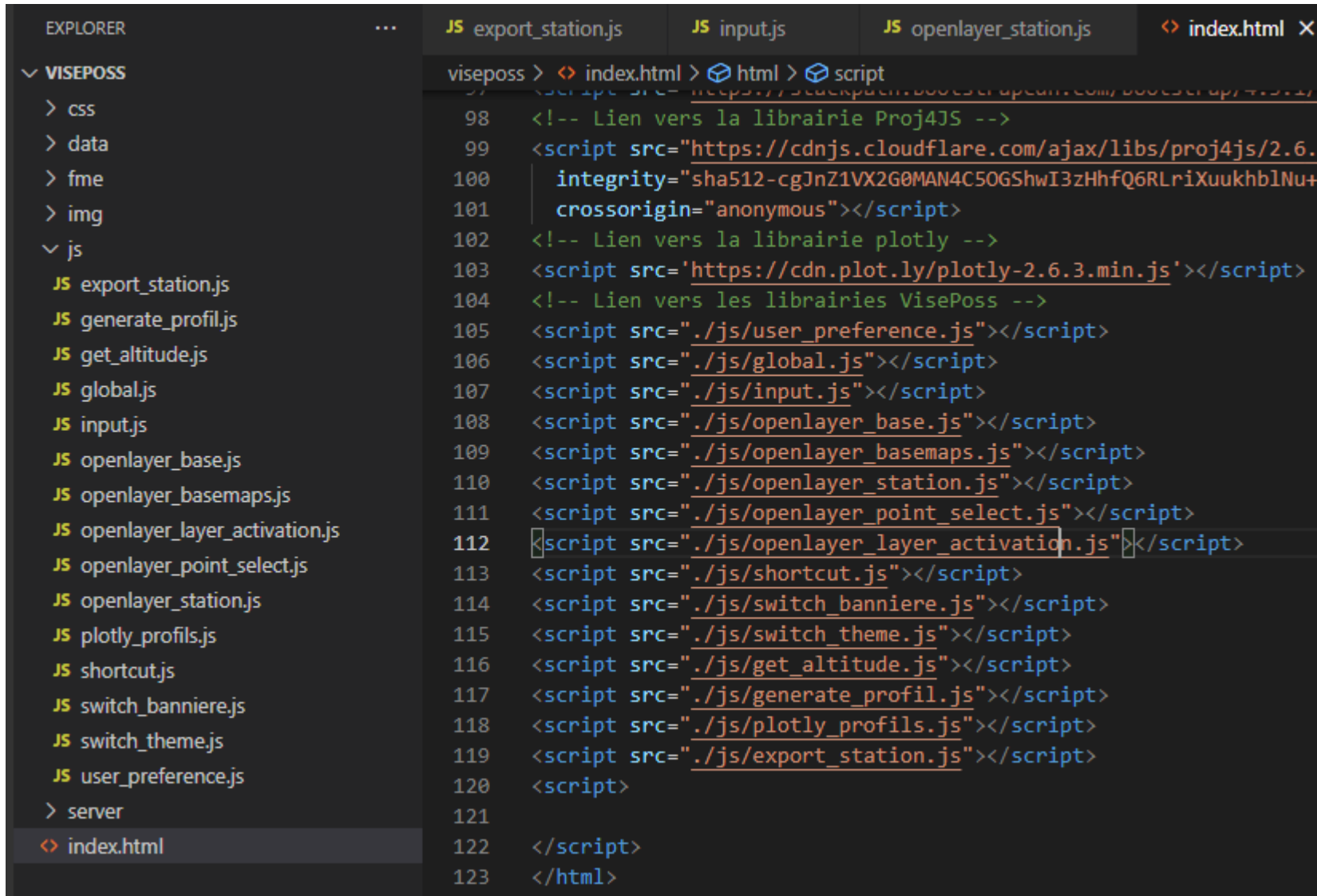
## PFP



# Préparation des données de base



# Démarche suivie (1)



The screenshot shows a code editor with a file explorer on the left and a code editor on the right. The file explorer shows a project named 'VISEPOSS' with a 'js' folder containing various JavaScript files. The code editor shows the content of 'index.html', which includes a series of script tags loading various JavaScript files from the 'js' folder and from external sources like Cloudflare and Plotly.

```
EXPLORER
VISEPOSS
  > css
  > data
  > fme
  > img
  > js
    JS export_station.js
    JS generate_profil.js
    JS get_altitude.js
    JS global.js
    JS input.js
    JS openlayer_base.js
    JS openlayer_basemaps.js
    JS openlayer_layer_activation.js
    JS openlayer_point_select.js
    JS openlayer_station.js
    JS plotly_profil.js
    JS shortcut.js
    JS switch_banniere.js
    JS switch_theme.js
    JS user_preference.js
  > server
  <> index.html

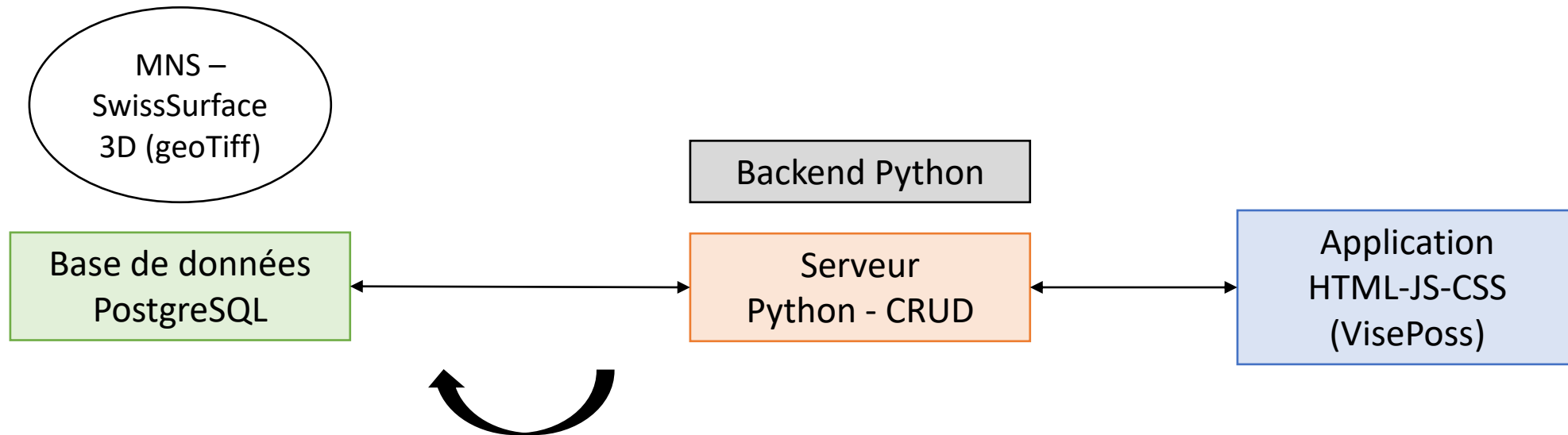
JS export_station.js
JS input.js
JS openlayer_station.js
<> index.html X

viseposs > <> index.html > html > script
97 <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.1/js/
98 <!-- Lien vers la librairie Proj4JS -->
99 <script src="https://cdnjs.cloudflare.com/ajax/libs/proj4js/2.6.2
100   integrity="sha512-cgJnZ1VX2G0MAN4C5OGShwI3zHhfQ6RLriXuukhblNu+T
101   crossorigin="anonymous"></script>
102 <!-- Lien vers la librairie plotly -->
103 <script src='https://cdn.plot.ly/plotly-2.6.3.min.js'></script>
104 <!-- Lien vers les librairies VisePoss -->
105 <script src="./js/user_preference.js"></script>
106 <script src="./js/global.js"></script>
107 <script src="./js/input.js"></script>
108 <script src="./js/openlayer_base.js"></script>
109 <script src="./js/openlayer_basemaps.js"></script>
110 <script src="./js/openlayer_station.js"></script>
111 <script src="./js/openlayer_point_select.js"></script>
112 <script src="./js/openlayer_layer_activation.js"></script>
113 <script src="./js/shortcut.js"></script>
114 <script src="./js/switch_banniere.js"></script>
115 <script src="./js/switch_theme.js"></script>
116 <script src="./js/get_altitude.js"></script>
117 <script src="./js/generate_profil.js"></script>
118 <script src="./js/plotly_profil.js"></script>
119 <script src="./js/export_station.js"></script>
120 <script>
121
122 </script>
123 </html>
```

- Création d'un fichier javascript pour chaque fonctionnalité
- Ajout de ces fichiers dans index.html

# Démarche suivie (2)

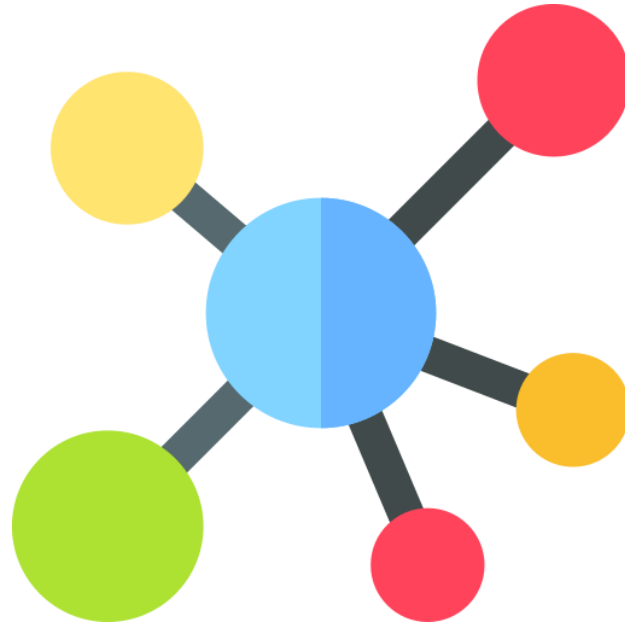
Utilisation de python-CRUD pour générer les profils de visée



```
# Exécution du SQL dans la BD
self.cursor.execute("""WITH line AS
-- From an arbitrary line
(SELECT 'SRID=2056;LINESTRING ("""+e_sta+" "+n_sta+" "+e_vise+" "+n_vise+""')::geometry AS geom),
linemesure AS
-- Add a mesure dimension to extract steps
(SELECT ST_AddMeasure(line.geom, 0, ST_Length(line.geom)) as linem,
generate_series(0, ST_Length(line.geom)::int, 1) as i
FROM line),
points2d AS
(SELECT ST_GeometryN(ST_LocateAlong(linem, i), 1) AS geom FROM linemesure),
cells AS
-- Get DEM elevation for each
(SELECT p.geom AS geom, ST_Value(data.test_raster.rast, 1, p.geom) AS val
FROM data.test_raster, points2d p
WHERE ST_Intersects(data.test_raster.rast, p.geom)),
-- Instantiate 3D points
pointsprofil AS
(SELECT val AS pt FROM cells)
-- Build 3D line from 3D points
SELECT pt FROM pointsprofil;""")
```



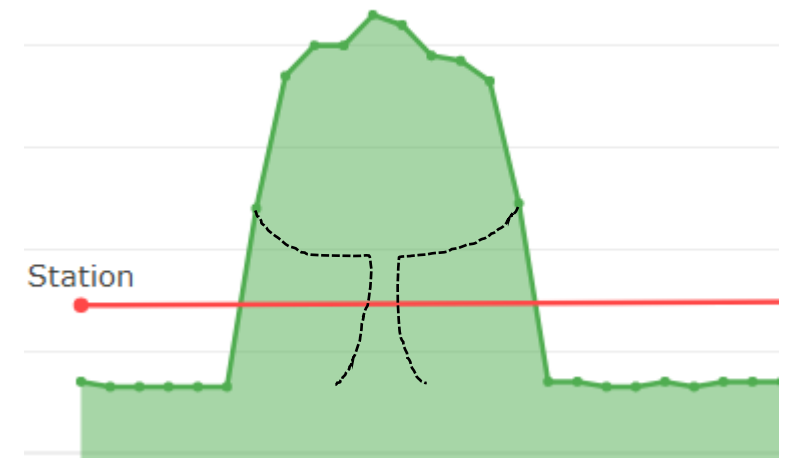
# Démonstration



# Pistes d'amélioration

- Importer des points avec un fichier local
- Impression d'un rapport en PDF
- Résolution adaptative du profil selon la distance entre les 2 points
- Mettre nos données de points fixes dans PG + Python-CRUD pour chargement dans OL
- Ajouter, dans la liste déroulante des points visés, si la visée est possible ou non
- Affinage des données de base (MNS : arbres)

PFP3 - 1588	✓
PFP3 - 1042	✗
PFP3 - 1066	✗
PFP3 - 1164	✓



# Questions?

