AIOS

BEST CENTRALESUPELEC

CentraleSupélec

# EBEC Paris Saclay

**Matthieu Annequin,**
**Hicham Bouanani**
**Arnaud Petit**
**Simon Tronchi**

EPSILON TECHNICAL SOLUTIONS

# Our Team

Arnaud Petit

Simon Tronchi

Matthieu Annequin

Hicham Bouanani

# Describe a location in natural language


# API

We use the overpass API, which is an API of OverStreetMap

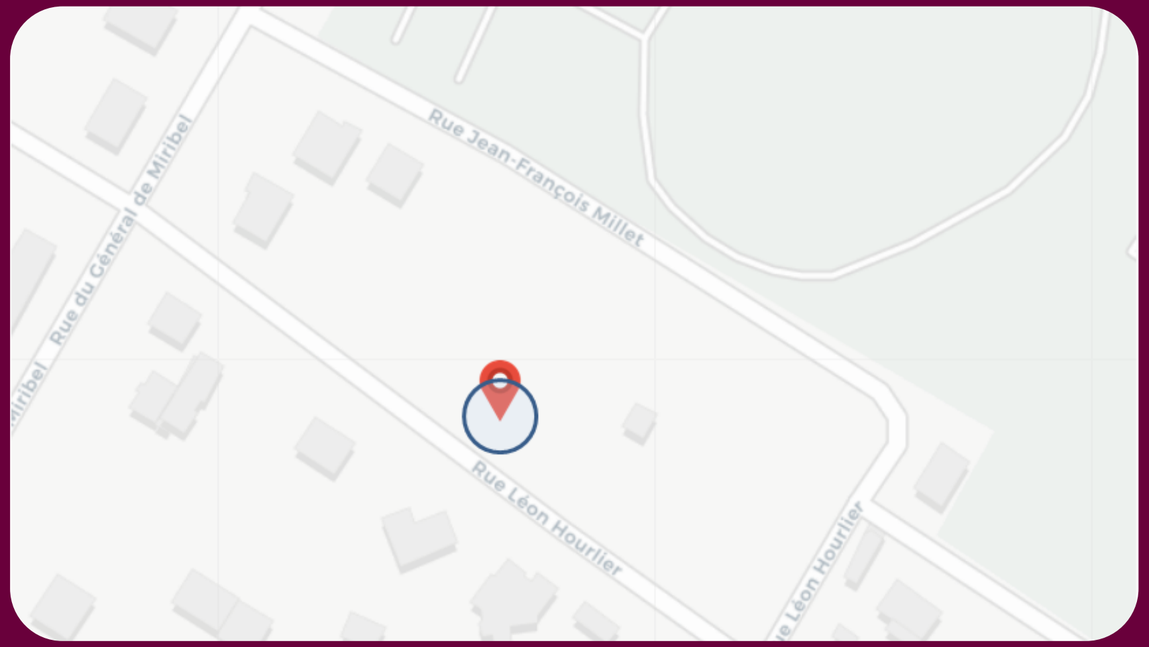
It takes a query and return a .json dataset

# Objectives

- OBJECTIVE 0: FIND THE STREET OF GPS COORDINATES
- OBJECTIVE 1: GET A TEXTUAL DECRIPTION OF THE POSITION OF GEOGRAPHIC POINT USING THE NEAREST CROSSINGS
- OBJECTIVE 2: DESCRIPTION OF SEVERAL POINTS ON A SECTION
- OBJECTIVE 3: DESCRIPTION OF TWO POINTS ON TWO CONSECUTIVE SECTIONS

# Our algorithms

**Binary search of the nearest city and the nearest street**

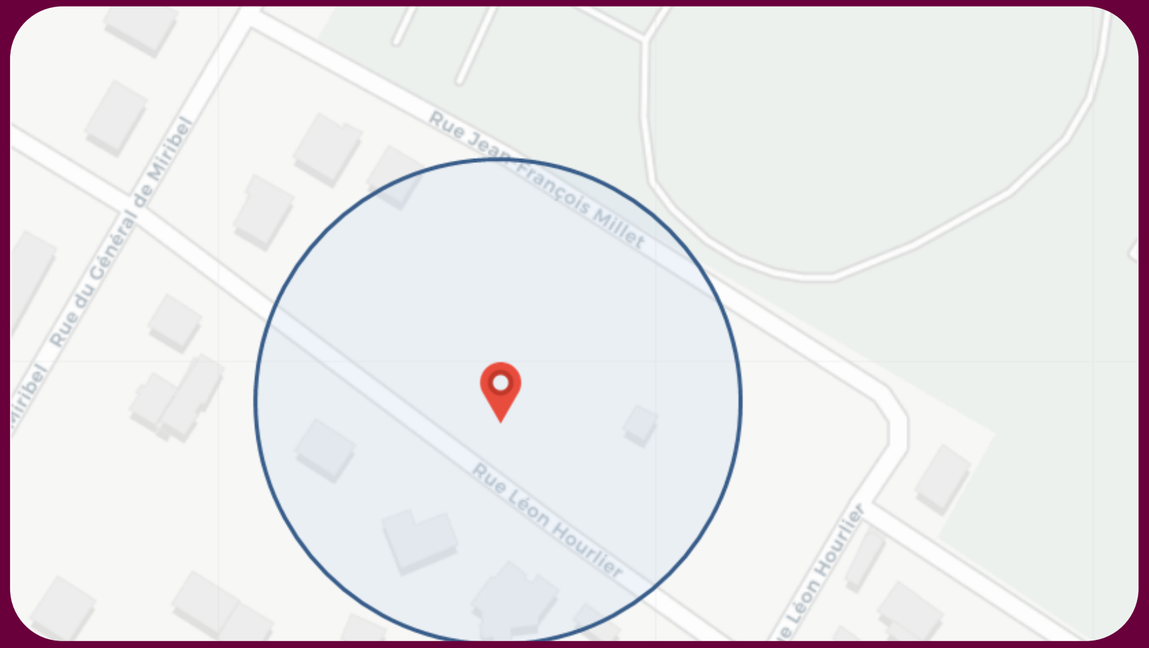



The algorithm uses concentric circles to find the nearest street

It adapts the radius using binary search

# Our algorithms

**Binary search of the nearest city and the nearest street**

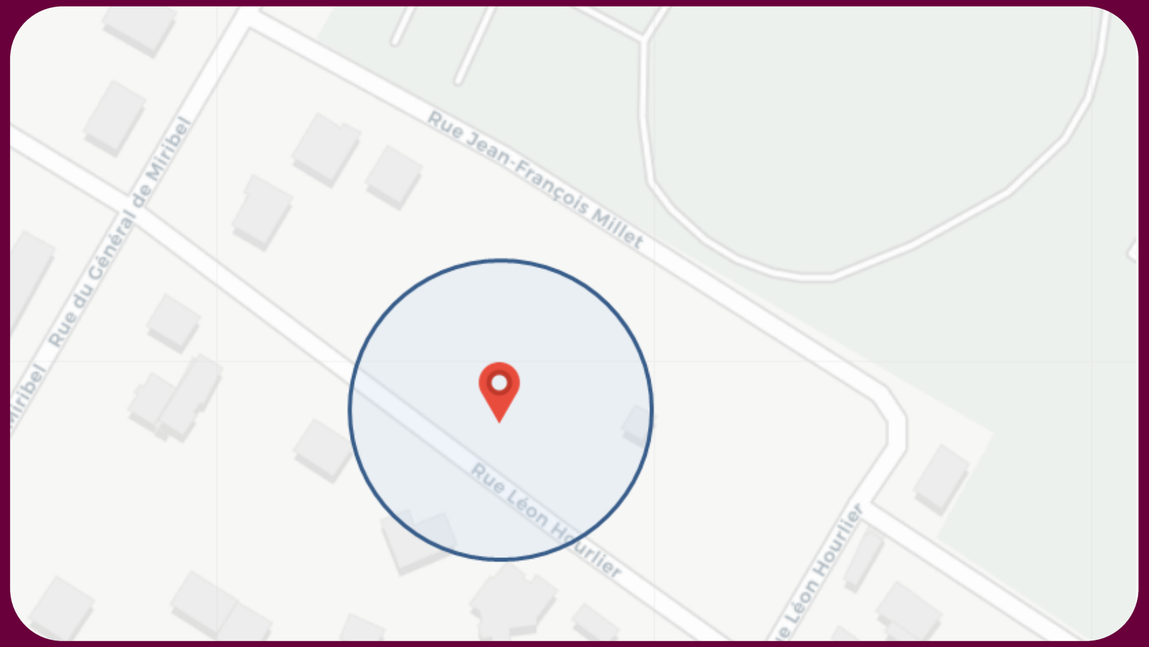



The algorithm uses concentric circles to find the nearest street

It adapts the radius using binary search

# Our algorithms

**Binary search of the nearest city and the nearest street**

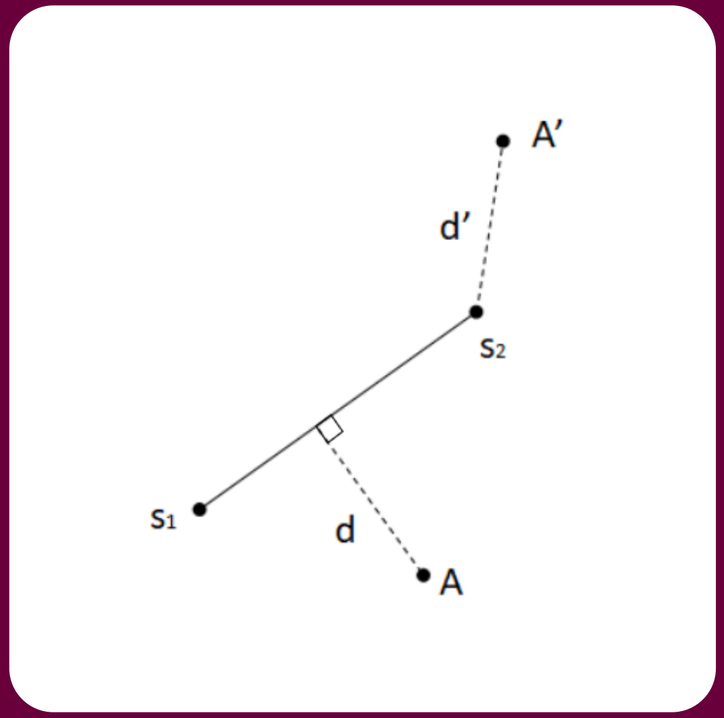


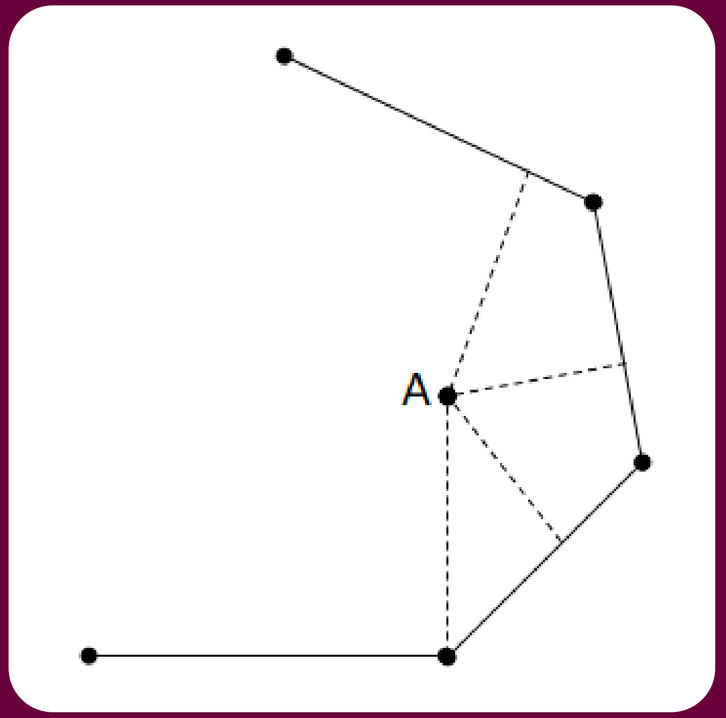

The algorithm uses concentric circles to find the nearest street

It adapts the radius using binary search

# Our algorithms

**Computation of the distance between a point and a segment**



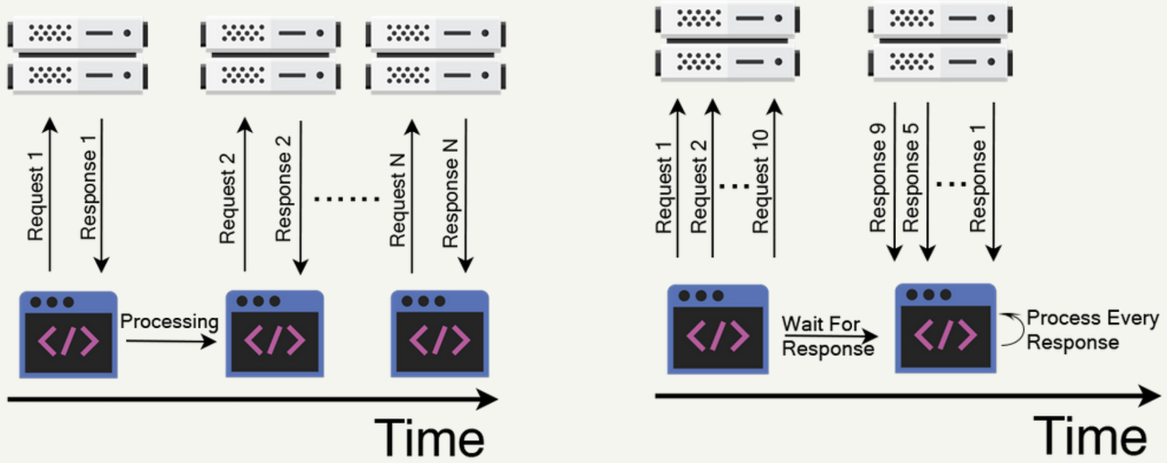




The algorithm finds the minimum distance between a point and a segment, using orthogonal projection if possible

Let n be the number of points,

$$argmin_{i\in\{1,\dots,n-1\}}\left(\frac{||(s_{i+1}-s_i)\wedge(s_i-A')||}{||s_{i+1}-s_i||}\right)$$

# Optimization of the requests: supercharge requests python library with custom methods

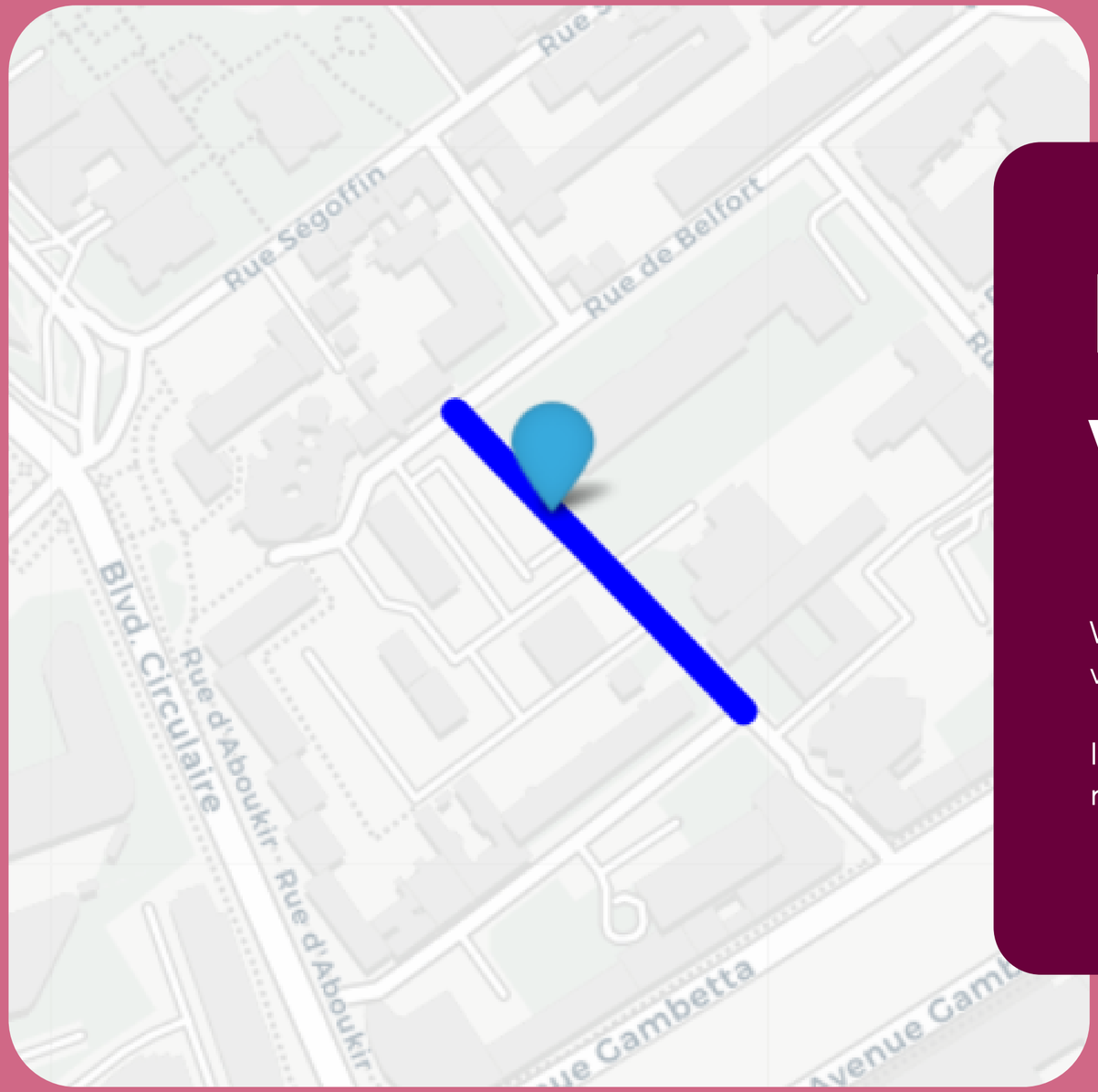



**Async**

Several API requests at once, continue the algotihm when possible while waiting for the responses.

**Cache system**

Store previous requests in a python dictionnary (saved afterward in a raw file) to faster process.




# Data Visualization

We created an algorithm to have a efficient visualization of our result susing the Folium library

It marks the given point and highlights the related road section.

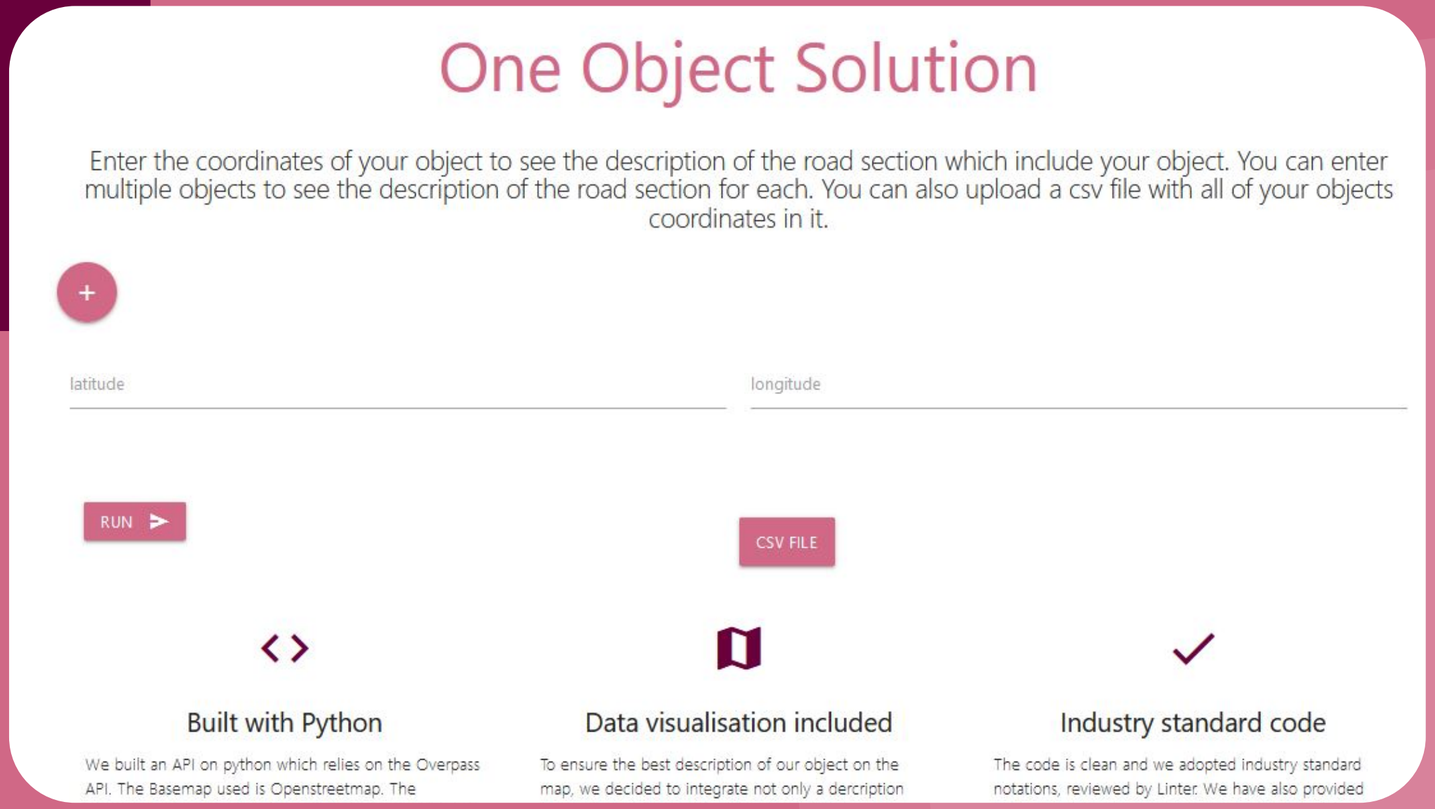
# Our solution

Algorithm : python package.
Web server with python backend.
Deployable on any app platform such as Heroku, debian server...

# Our interface

We create an interface that enables the user to easyly enter coordinates by hand or using a .csv file

It returns the visualization, a table with the result and a sentence in French that gives the location of the tree



# Thank you!


Check our Github repository