

Projet Smart-Réno



Fiabiliser, Professionnaliser, Valoriser la Rénovation Energétique
Deuxième volet : Industrialiser les processus de rénovation et professionnaliser les acteurs

Rapport volet 2.2

Auteur : Boris Brangeon (boris.brangeon@plateforme-tipee.com) – 14/12/2020

Relecteur : Cécile Jolas (cecile.jolas@plateforme-tipee.com) – 06/01/2020



TABLE DES MATIERES

Projet Smart-Réno	1
Rapport volet 2.2	1
1. Introduction	3
2. Avant-propos : introduction aux technologies utilisées dans la plateforme SaaS	6
3. Géolocalisation de l'habitation	7
4. Identification de l'existant	10
5. Reconnaissance des composants de façade	15
5.1. Lexique utilisé dans la source Classification, Détection d'objets et Segmentation d'images :	15
5.2. Algorithme de réseaux de neurones pour la détection d'objets : YOLOv3	16
5.3. Labélisation / Annotation des images / Apprentissage / Prédictions	17
5.4. Segmentation sémantique	19
5.5. Estimation des surfaces des composants	21
5.6. Orientations multiples et interface utilisateur	22
6. Proposition de modèle 2D/3D de l'existant	22
7. Diagnostic énergétique de l'existant	24
8. Conclusion	26

TABLE DES FIGURES

Figure 1 Industrialisation des processus de rénovation.	3
Figure 2 Schéma explicatif des relations entre le back-end et le front-end.	7
Figure 3 Interface de géolocalisation.	8
Figure 4 Proposition des adresses par auto complétion.	9
Figure 5 Découpage de la commune.	9
Figure 6 Découpage des parcelles cadastrales.	10
Figure 7 Proposition des empreintes au sol des habitations.	10
Figure 8 Définitions des attributs d'altitude sur un bâtiment (page 55 de DC_BDTopo_3-0.pdf)	12
Figure 9 Hauteur du bâtiment dan BD Topo.	12
Figure 10 Exemple de la fonction f(dmatto).	13
Figure 11 Répartition des pourcentages par type de matériaux et par région en 2016 pour les murs	13
Figure 12 Répartition des pourcentages par type de matériaux et par région en 2016 pour la toiture	14
Figure 13 Code couleur concernant la fiabilité des données.	14
Figure 14 Illustration des différentes méthodes de détection d'objet dans une image.	16
Figure 15 Bibliographie non exhaustive des réseaux de neurones profond utilisé dans la reconnaissance d'objets	16
Figure 16 Architecture du réseau de neurone YOLOv3	17
Figure 17 Labélisation des images dans labeling.	18
Figure 18 Résultats de la détection des composants de façade avec YoloV3.	18
Figure 19 Annotations dans le logiciel CVAT.	19
Figure 20 Quelques chiffres.	20
Figure 21 Résultats de la prédiction des composants de façade avec DeepLabV3+.	20
Figure 22 Image originale avant post traitement.	21
Figure 23 Succession de méthodes de traitement d'images.	21
Figure 24 Estimation des surfaces des composants.	21
Figure 25 Interface de chargement des photos de façade.	22
Figure 26 Exemple de requête Overpass.	23
Figure 27 Représentation du bâtiment en .OBJ à partir d'une requête Overpass.	23

1. Introduction

Le projet Smart Réno a pour finalité d'aider à l'accélération des actes de rénovation des bâtiments résidentiels individuels en France. L'activité de rénovation des bâtiments peut présenter des défauts importants, empêchant le passage à l'acte ; atomisation de l'offre, fluctuation importante des coûts, prescriptions personnalisées inexistantes...

Il est donc important, pour inciter les clients à engager des processus de rénovation, d'améliorer leur confiance en professionnalisant les filières engagées à travers une architecture de services.

Dans le volet 2 du projet, nous nous concentrons sur le développement d'une chaîne d'outils qui, en s'appuyant sur des moyens récents tels que le BIM*, les données issues d'objets connectés ou de compteurs intelligents, les méthodes d'analyse d'images et toutes les ressources du Net, permettront la génération automatique de services liés à la rénovation : diagnostic, analyse d'impact de gestes de rénovation, métrés, devis, ... le tout centré sur le cas des maisons individuelles.

L'ensemble du volet 2 se décompose en 5 tâches :

1. Données exploitables ;
2. Diagnostic énergétique automatisé des bâtiments existants ;
3. Outils d'évaluation des apports des solutions de rénovation ;
4. Construction automatique de métrés et de devis ;
5. Construction de plans de travaux et de plans de contrôle de la qualité.

Cette chaîne d'outils se présente sous la forme d'un applicatif de type "Proof-of-Concept" (POC), c'est à dire une interface en ligne permettant d'aider à la préconisation de stratégies optimales de rénovation énergétique.

Ce POC est constitué de briques méthodologiques et/ou calculatoires élémentaires interchangeableables. Le périmètre d'étude restant la rénovation énergétique de maisons individuelles¹. La Figure 1 propose une décomposition des processus d'industrialisation de la rénovation, qui constitue la feuille de route de développement de l'applicatif.

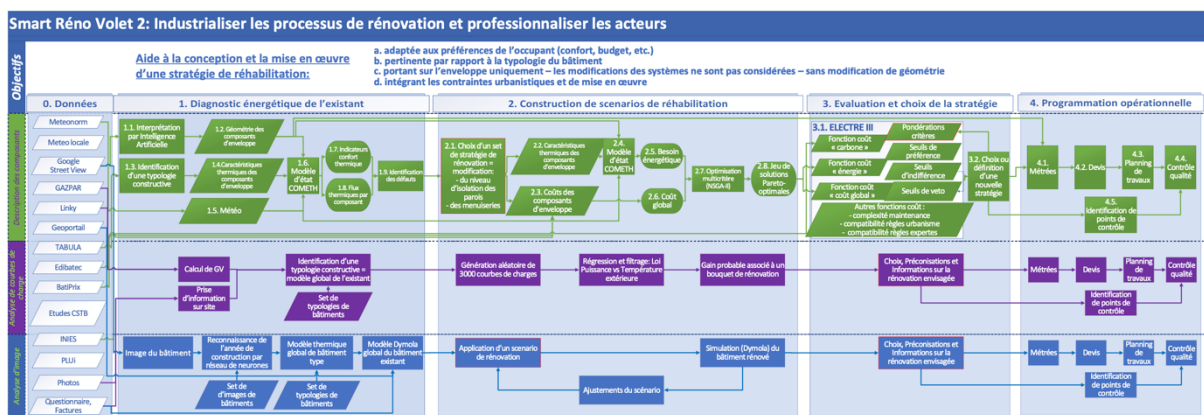
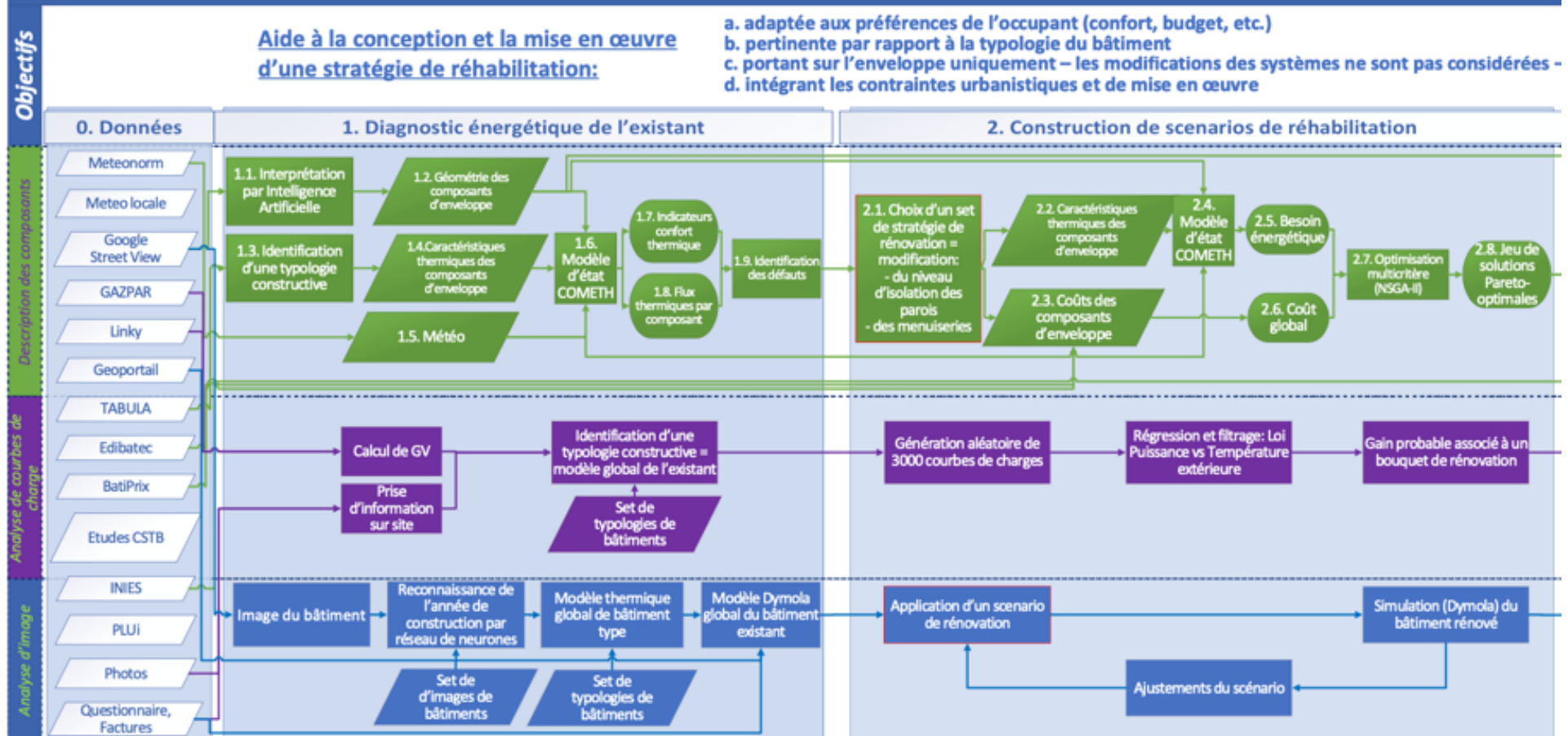


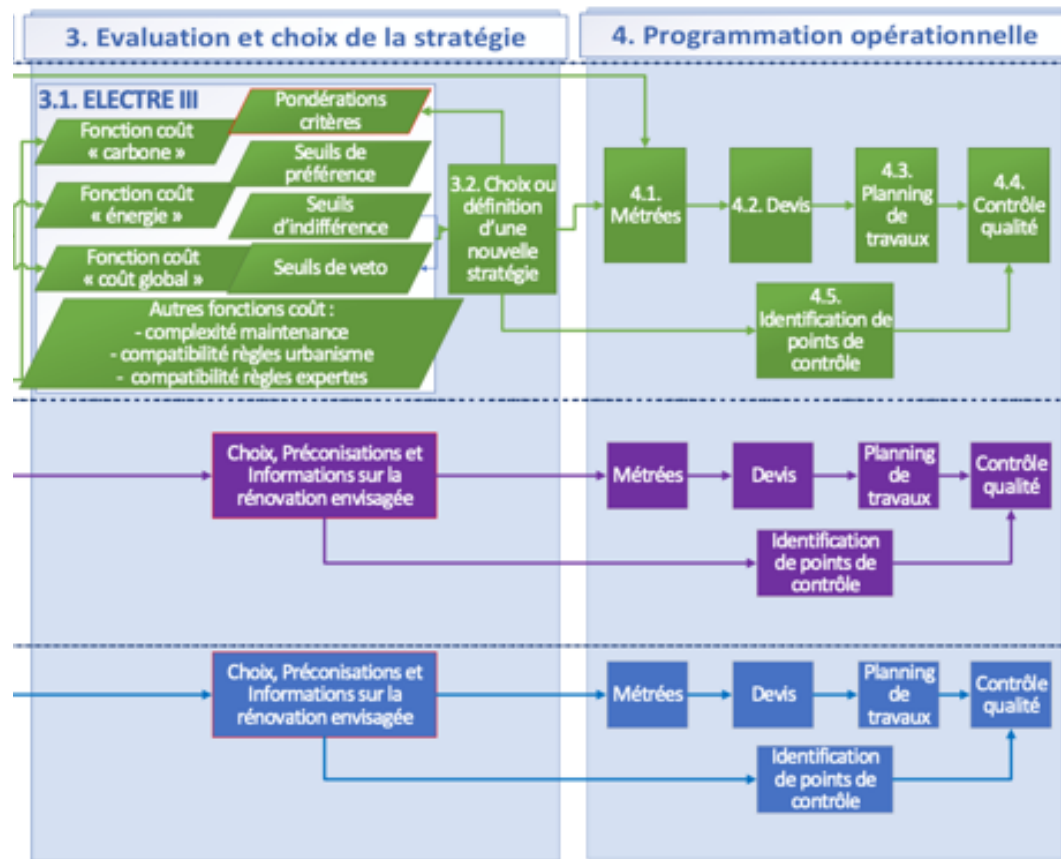
Figure 1 Industrialisation des processus de rénovation.

¹ L'application dans un second temps aux immeubles de logements collectifs n'est cependant pas exclue.

Smart Réno Volet 2: Industrialiser les processus de rénovation et professionnaliser les acteurs



sans modification de géométrie



L'interface en ligne est une plateforme *software as a service* ([SaaS](#)) ou logiciel en tant que service. C'est un modèle d'exploitation des logiciels dans lequel ceux-ci sont installés sur des serveurs distants plutôt que sur la machine de l'utilisateur. L'utilisation de solutions logicielles en tant que service en entreprise permet un meilleur contrôle des charges techniques. L'ensemble des solutions techniques étant délocalisées le coût devient fixe, généralement fonction du nombre de personnes utilisant la solution SaaS.

Un avantage manifeste pour les entreprises est la rapidité de déploiement lorsque le logiciel SaaS correspond exactement au besoin (et qu'il ne nécessite aucune adaptation). Les solutions SaaS étant déjà préexistantes, le temps de déploiement est extrêmement faible. Néanmoins, un des inconvénients majeurs, lors de la mise en place de solutions SaaS, est que les données relatives à l'entreprise cliente sont généralement stockées sur les serveurs du prestataire fournissant la solution. Lorsqu'il s'agit de données sensibles ou confidentielles, l'entreprise est obligée de prendre des dispositions contractuelles avec le fournisseur (RGPD par exemple).

La plateforme SaaS développée par Tipee dans le cadre du volet 2 de Smart-Réno se décompose en dix tâches déroulées par les utilisateurs :

1. Une géolocalisation de l'habitation ;
2. Une identification de l'existant ;
3. Une reconnaissance des composants de façade ;
4. Une proposition de modèle 2D/3D de l'existant ;
5. Un diagnostic énergétique de l'existant ;
6. Une construction des scénarios de réhabilitation ;
7. Une évaluation et choix de la stratégie de réhabilitation ;
8. Une construction automatique de métrés et de devis ;
9. Une construction des plans de travaux et de plans de contrôle de la qualité ;
10. Un bilan global.

Le document a pour but de présenter la plateforme qui est le fil conducteur des travaux menés au travers des tâches 1 et 2 du volet 2. Un autre document présentera les tâches 3 à 5.

2. Avant-propos : introduction aux technologies utilisées dans la plateforme SaaS

La plateforme SaaS est composée d'un front-end et d'un back-end. Les interfaces sur lesquelles interagiront les utilisateurs sont considérées comme du front-end. En effet, il s'agit de la partie visible de l'application, destinée à être manipulée par un tiers. Le front-end se compose généralement d'un client web : [ReactJS](#), [PHP](#), [HTML](#) qui communique de manière sécurisée avec le back-end.

Une interface bien qu'ergonomique et design reste une coquille vide s'il n'y a pas de back-end. En accord avec un cahier des charges, le back-end va exécuter les fonctionnalités de la plateforme (lecture/écriture des données, exécution de services, etc.). Le back-end se compose généralement de trois éléments :

- Un serveur web : [Nginx](#), [Apache](#)
- Une application web service : [Flask](#), PHP
- Une base de données pour organiser les données : [Mongodb](#), [PostgreSQL](#)

La Figure 2 présente les relations entre les différentes technologies nécessaire au fonctionnement de la plateforme SaaS.

On retrouve le front-end développé avec le Framework ReactJS – [Redux](#). Celui-ci envoie des requêtes HTTP et reçoit des réponses HTTP au format JavaScript Object Notation (JSON), qui est un format léger d'échange de données, au serveur WEB (Nginx).

Le serveur Web transmet, filtre ces différentes informations au web service API RESTful (Flask). API REST (Representational State Transfer Application Program Interface) est un style architectural qui permet aux logiciels de communiquer entre eux sur un réseau ou sur un même appareil. Le plus souvent les développeurs utilisent des API REST pour créer des services web. Souvent appelés services web RESTful, REST utilise des méthodes http (GET, POST, DELETE) pour récupérer et publier des données entre un périphérique client et un serveur. La sécurité de l'API concerne le transfert des données via des API connectées à Internet.

Pour la délégation des accès, il existe une norme ouverte nommée OAuth (Open Authorization). Elle permet aux utilisateurs d'accorder à des tiers l'accès à des ressources web sans avoir à partager leurs mots de passe.

Suivant la méthode HTTP utilisée, le service web communique soit directement ou par un intermédiaire (liste de tâches à effectuer de manière synchrone ou asynchrone : [Celery - Distributed Task Queue](#)) avec les différents services suivants :

1. Base de données (Mongodb, PostgreSQL) ;
2. Exécution de scripts ([Python](#)) ;

Pour finir, le service web récupère les différentes réponses au client (front-end) qui interprète les données.

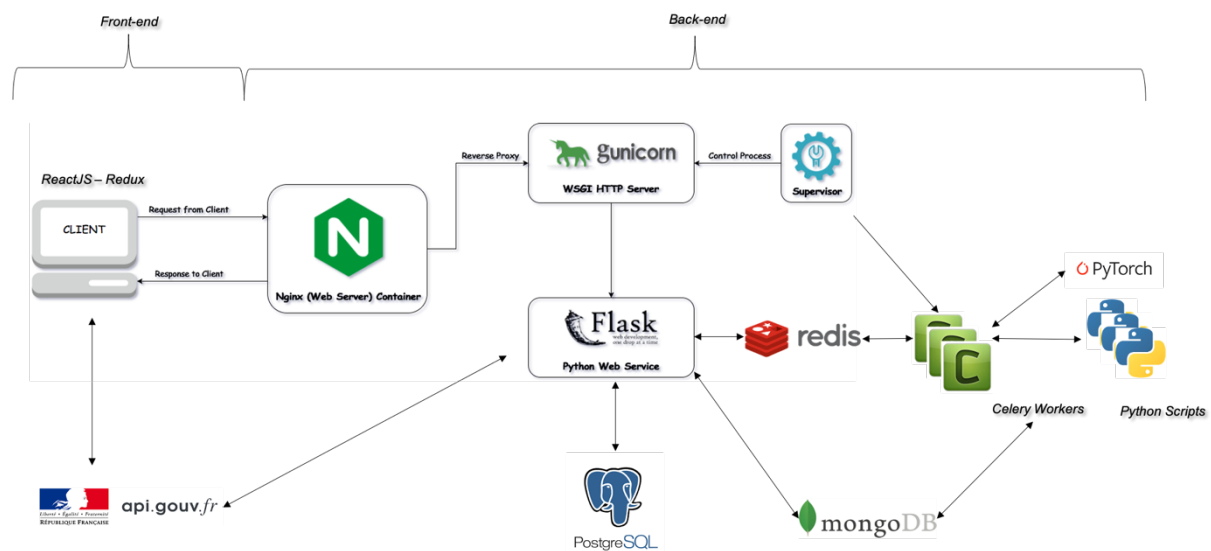


Figure 2 Schéma explicatif des relations entre le back-end et le front-end.

3. Géolocalisation de l'habitation

Les données de géolocalisation et de découpages géographiques sont issues de la plateforme ouverte des données publiques françaises data.gouv.fr qui est une [plateforme de diffusion de données publiques](#) (« *open data* ») de l'[État français](#).

L'interface utilisateur propose une carte basée sur [OpenStreetMap](#) et une barre de recherche d'adresses qui communique avec l'API gouvernementale (<https://geo.api.gouv.fr/adresse>).

Le découpage des communes est basé sur l'API <https://geo.api.gouv.fr/decoupage-administratif/communes>.

Les figures suivantes illustrent le parcours utilisateur. (Voir Figure 3, Figure 4, Figure 5, Figure 6 et Figure 7).

Explication du parcours utilisateur concernant la géolocalisation de son habitation :

1. L'utilisateur recherche l'adresse de son logement et valide une adresse ;
2. L'interface géolocalise le logement et propose le découpage de la commune ;
3. L'utilisateur doit sélectionner et valider sa parcelle cadastrale ;
4. Par la suite, la plateforme SaaS recherche les bâtiments disponibles dans la base de données BDTOPO V3 ;
5. L'utilisateur sélectionne et valide son bâtiment.

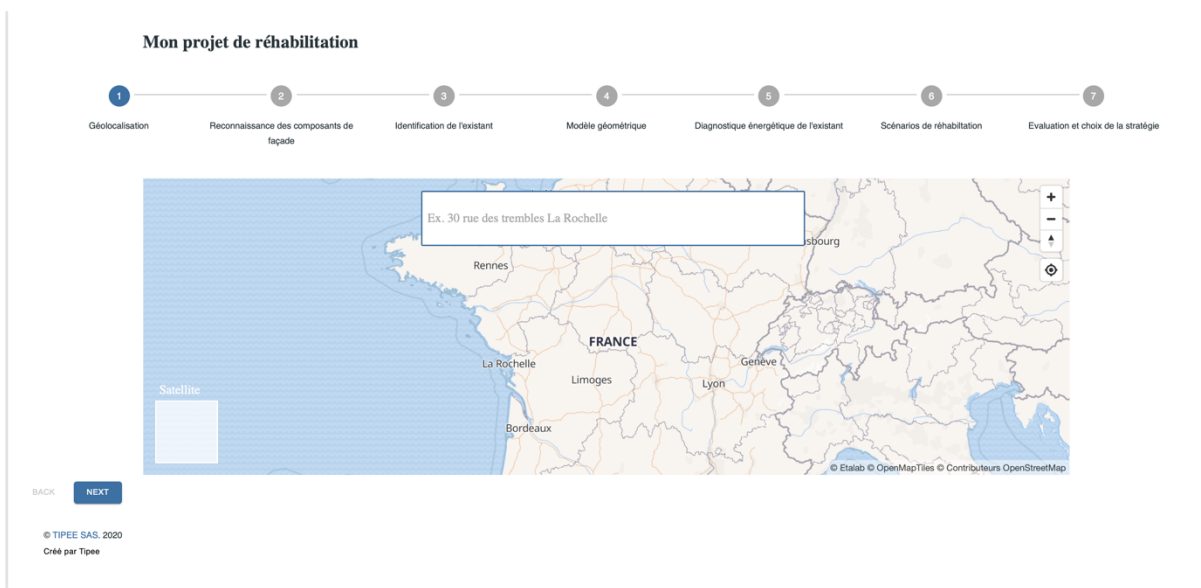


Figure 3 Interface de géolocalisation.

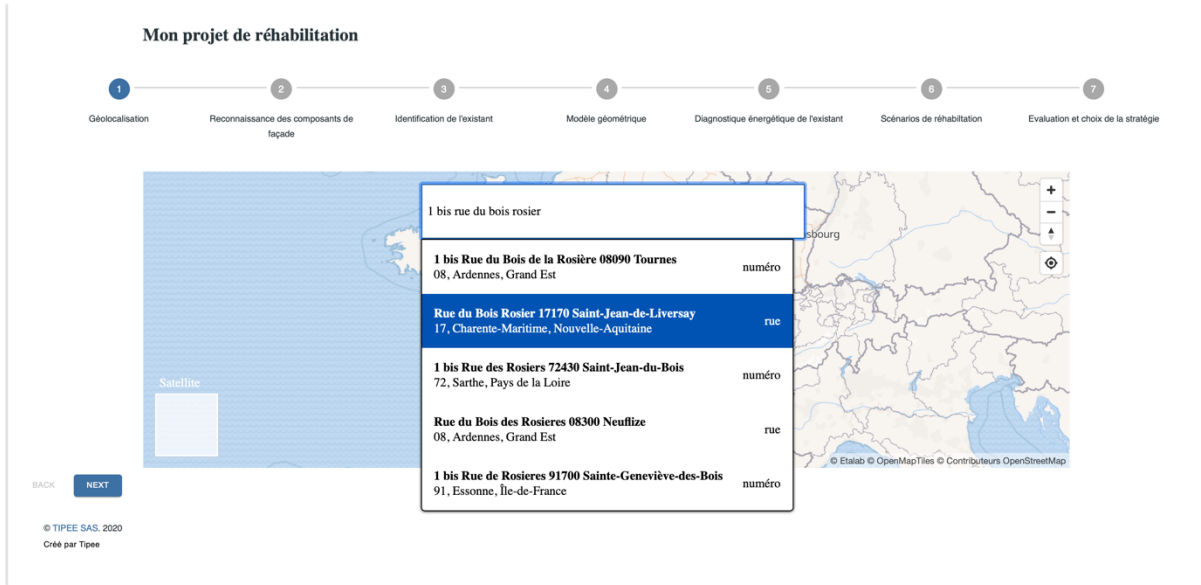


Figure 4 Proposition des adresses par auto complétion.

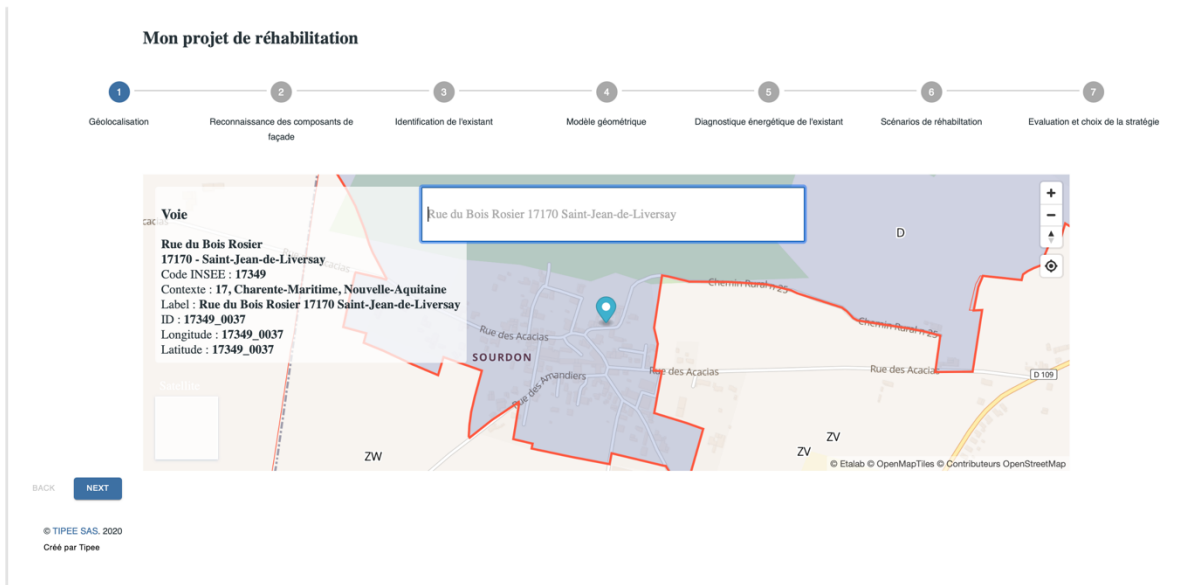


Figure 5 Découpage de la commune.

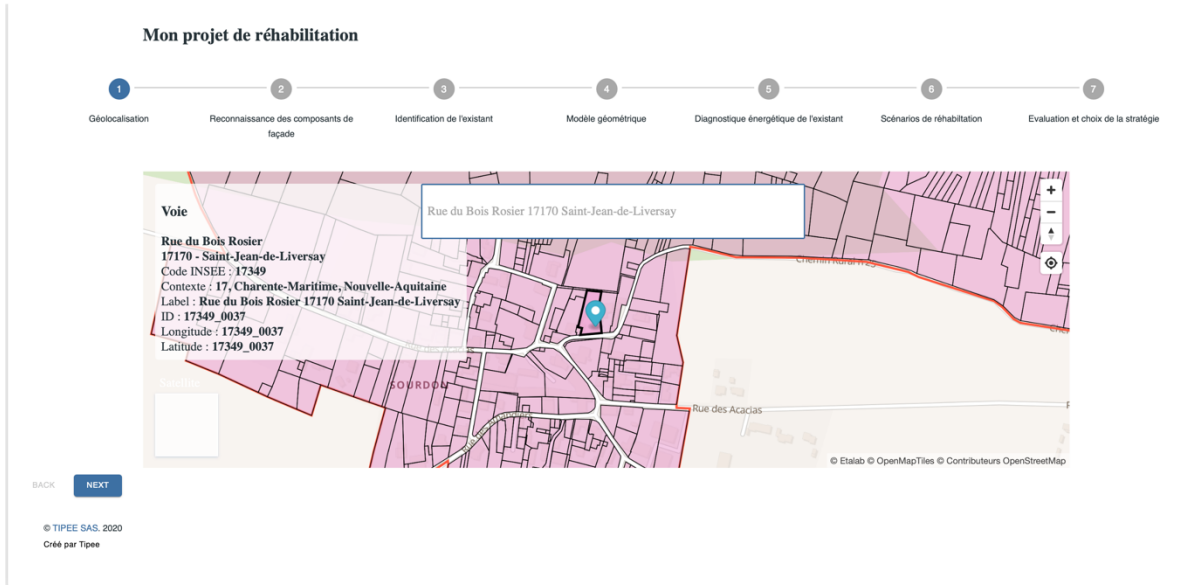


Figure 6 Découpage des parcelles cadastrales.

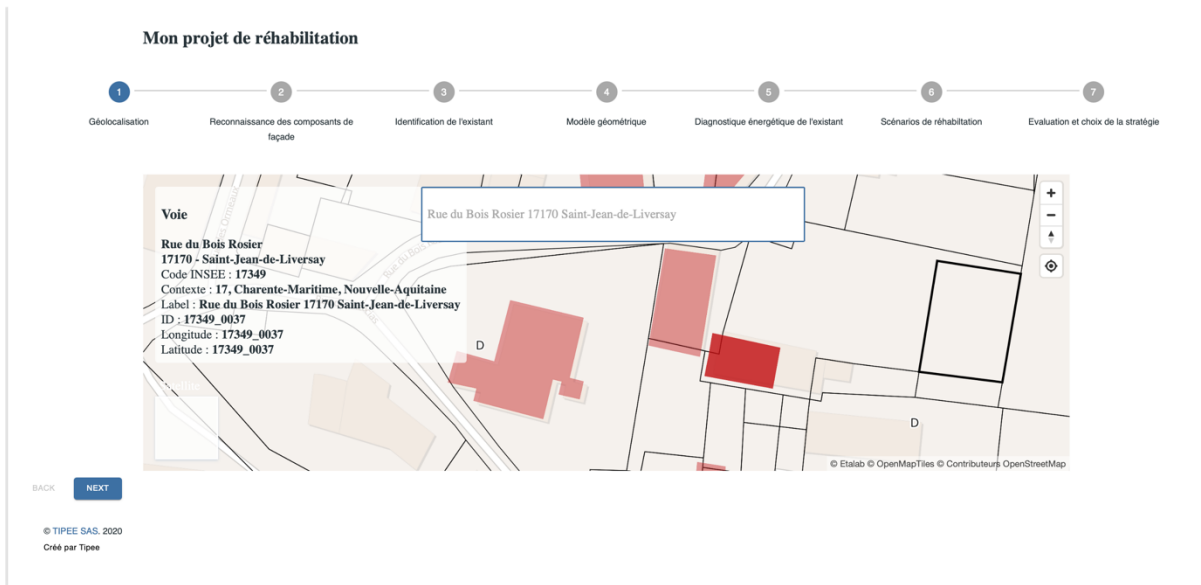


Figure 7 Proposition des empreintes au sol des habitations.

4. Identification de l'existant

Les données du bâtiment dans son état existant sont obtenues essentiellement à partir de la base de données BD Topo V3 au format PostgreSQL.




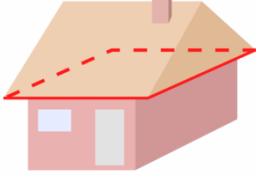


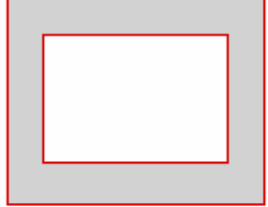



La BD Topo est une base de données vectorielles 2D et 3D disponible depuis 2007 sur l'ensemble du territoire français. D'une précision métrique, elle permet une exploitation jusqu'à une échelle de 1/10 000 voire 1/5 000. Elle propose des [méta-données](#) liées aux bâtiments.

A compter du 1^{er} janvier 2021, toutes les données publiques (vecteur et ortho-images) de l'IGN seront libres et accessibles gratuitement pour tous.

Concernant l'origine géométrique du bâtiment, elle peut provenir de différentes sources (page 55 de [DC BDTOPO 3-0.pdf](#)) :

1. Cadastre : la géométrie du bâtiment provient de fichiers cadastraux, parfois recalés par l'IGN. La saisie des contours est faite au niveau du mur.
2. Imagerie aérienne : la géométrie du bâtiment provient de la numérisation d'images aériennes. La saisie des contours est faite "à la gouttière" : la surface de l'objet bâtiments ne reflète pas rigoureusement l'emprise au sol au sens cadastral.
3. Autre : le mode de saisie de la géométrie du bâtiment est inconnu ou non identifié comme provenant du Cadastre ou d'une saisie sur images aériennes.

Le tableau suivant reprend la définition de la modélisation géométrique des bâtiments dans BD Topo (page 56 de [DC BDTOPO 3-0.pdf](#)).

MONDE REEL ET MODELISATION	MODELISATION GEOMETRIQUE
 <p><i>Bâtiments (immeubles contigus) issu de la BD Parcelaire®</i></p>	<p>Perspective :</p>  <p>Vue de dessus :</p> 
 <p><i>Bâtiment (maison individuelle) issu de la photogrammétrie</i></p>	
 <p><i>Cas d'un corps de ferme avec cour intérieure</i></p>	
 <p><i>Bâtiment (immeubles contigus) issu de la photogrammétrie</i></p>	<p>Perspective :</p>  <p>Vue de dessus :</p> 

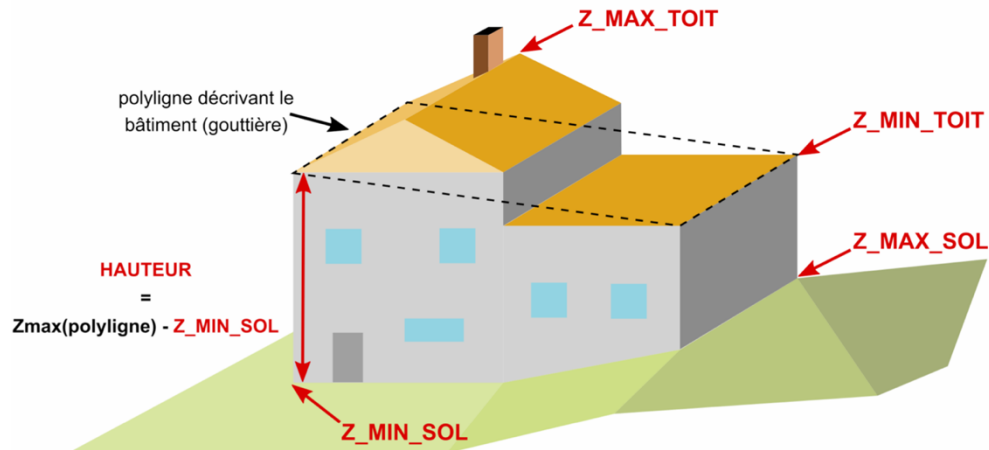


Figure 8 Définitions des attributs d'altitude sur un bâtiment (page 55 de DC_BDTOPO_3-0.pdf)

La hauteur des bâtiments (page 67 de [DC_BDTOPO_3-0.pdf](#)) est mesurée entre le sol et le point haut de la gouttière (altitude maximum de la poly-ligne décrivant le bâtiment), exprimée en mètres.

En cas de pente dans le terrain sur lequel est construit le bâtiment, le point le plus bas du sol est retenu. Par "gouttière", il faut entendre bord du toit ou aplomb du mur si le toit est plat. Si la gouttière n'est pas horizontale, on retient le point le plus haut.

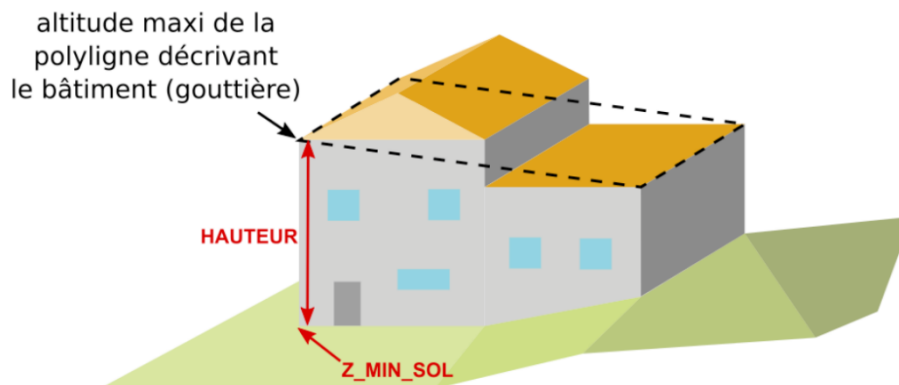


Figure 9 Hauteur du bâtiment dans BD Topo.

Les attributs informationnels présents dans BD TOPO concernant les **matériaux des murs, des toits, le nombre d'étages** et certaines valeurs d'**Usage1** et **Usage2** sont issus des fichiers fonciers, et plus particulièrement des fichiers fonciers 'MAJIC' (Mise À Jour des Informations Cadastreales) produits par la DGFIP, ensuite antonymies par le CEREMA.

Les valeurs calculées sont issues de fonctions (f) utilisant les variables suivantes :

1. Nombre d'étages = f(dnbniv, dniv)
2. Date d'apparition = f(jannath) ;
3. Usage 1 / Usage 2 = f(dteloc, ccoaff, fonction BDTopo) ;
4. Nombre de logements = f(dteloc, cconlc, ccoaff, dnudes) ;
5. Matériaux des murs = f(dmatgm) ;
6. Matériaux des toits = f(dmatto) (voir Figure 10).

code	valeur	Description	Observation	2009	2011	2012	2013	2014	2015	2016
0	INDETERMINE			✓	✓	✓	✓	✓	✓	✓
1	TUILES			✓	✓	✓	✓	✓	✓	✓
2	ARDOISES			✓	✓	✓	✓	✓	✓	✓
3	ZINC ALUMINIUM			✓	✓	✓	✓	✓	✓	✓
4	BETON			✓	✓	✓	✓	✓	✓	✓
9	AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			
00	INDETERMINE			✓	✓	✓	✓	✓	✓	✓
01	TUILES			✓	✓	✓	✓	✓	✓	✓
02	ARDOISES			✓	✓	✓	✓	✓	✓	✓
03	ZINC ALUMINIUM			✓	✓	✓	✓	✓	✓	✓
04	BETON			✓	✓	✓	✓	✓	✓	✓
09	AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			
10	TUILES			✓	✓	✓	✓	✓	✓	✓
11	TUILES			✓	✓	✓	✓	✓	✓	✓
12	ARDOISES - TUILES			✓	✓	✓	✓	✓	✓	✓
13	TUILES - ZINC ALUMINIUM			✓	✓	✓	✓	✓	✓	✓
14	BETON - TUILES			✓	✓	✓	✓	✓	✓	✓
19	TUILES - AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			
20	ARDOISES			✓	✓	✓	✓	✓	✓	✓
21	ARDOISES - TUILES			✓	✓	✓	✓	✓	✓	✓
22	ARDOISES			✓	✓	✓	✓	✓	✓	✓
23	ARDOISES - ZINC ALUMINIUM			✓	✓	✓	✓	✓	✓	✓
24	ARDOISES - BETON			✓	✓	✓	✓	✓	✓	✓
29	ARDOISES - AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			
30	ZINC ALUMINIUM			✓	✓	✓	✓	✓	✓	✓
31	TUILES - ZINC ALUMINIUM			✓	✓	✓	✓	✓	✓	✓
32	ARDOISES - ZINC ALUMINIUM			✓	✓	✓	✓	✓	✓	✓
33	ZINC ALUMINIUM			✓	✓	✓	✓	✓	✓	✓
34	BETON - ZINC ALUMINIUM			✓	✓	✓	✓	✓	✓	✓
39	ZINC ALUMINIUM - AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			
40	BETON			✓	✓	✓	✓	✓	✓	✓
41	BETON - TUILES			✓	✓	✓	✓	✓	✓	✓
42	ARDOISES - BETON			✓	✓	✓	✓	✓	✓	✓
43	BETON - ZINC ALUMINIUM			✓	✓	✓	✓	✓	✓	✓
44	BETON			✓	✓	✓	✓	✓	✓	✓
49	BETON - AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			
90	AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			
91	TUILES - AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			
92	ARDOISES - AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			
93	ZINC ALUMINIUM - AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			
94	BETON - AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			
99	AUTRES		n'existe pas dans la doc	✓	✓	✓	✓			

Figure 10 Exemple de la fonction $f(dmatto)$.

	Pierre	Meulière	Béton	Brique	Aggloméré	Bois	Autres	Non Renseigné
Auvergne-Rhone-Alpes	23%	13%	26%	3%	13%	2%	7%	13%
Bourgogne-Franche-Comte	35%	8%	16%	8%	20%	2%	3%	8%
Bretagne	26%	9%	13%	7%	29%	1%	5%	9%
Centre-Val de Loire	25%	10%	18%	13%	18%	1%	9%	6%
Corse	35%	1%	9%	3%	31%	1%	1%	19%
Grand-Est	18%	9%	18%	20%	16%	2%	4%	12%
Hauts-de-France	5%	2%	13%	51%	9%	1%	7%	11%
Ile-de-France	14%	9%	27%	16%	7%	1%	6%	20%
Normandie	16%	5%	17%	16%	25%	3%	8%	10%
Nouvelle-Aquitaine	27%	9%	11%	22%	16%	2%	3%	9%
Occitanie	23%	7%	11%	27%	13%	1%	4%	12%
Pays de la Loire	23%	5%	17%	7%	32%	1%	4%	10%
Provence-Alpes-Cote d'Azur	22%	5%	18%	5%	28%	1%	3%	19%
Departements d'Outre Mer	4%	0%	20%	7%	28%	9%	7%	25%
Total France	20%	8%	18%	16%	17%	2%	5%	13%

Figure 11 Répartition des pourcentages par type de matériaux et par région en 2016 pour les murs

	Tuiles	Ardoises	Zinc Aluminium	Béton	Autres	Non Renseigné
Auvergne-Rhone-Alpes	59%	4%	4%	12%	9%	13%
Bourgogne-Franche-Comte	72%	4%	2%	7%	6%	8%
Bretagne	1%	75%	5%	4%	7%	9%
Centre-Val de Loire	48%	29%	2%	7%	8%	6%
Corse	60%	9%	1%	9%	3%	19%
Grand-Est	66%	4%	3%	6%	9%	12%
Hauts-de-France	59%	11%	3%	7%	10%	11%
Ile-de-France	33%	5%	13%	21%	7%	20%
Normandie	36%	33%	2%	7%	10%	10%
Nouvelle-Aquitaine	75%	6%	2%	4%	4%	9%
Occitanie	70%	6%	1%	5%	6%	12%
Pays de la Loire	40%	38%	2%	5%	5%	10%
Provence-Alpes-Cote d'Azur	57%	1%	2%	16%	6%	19%
Departements d'Outre Mer	2%	0%	20%	9%	44%	25%
Total France	51%	13%	5%	10%	8%	13%

Figure 12 Répartition des pourcentages par type de matériaux et par région en 2016 pour la toiture

Les spécifications de qualité d'origine sont disponibles dans la [documentation attributaire des fichiers fonciers](#). Concernant la qualité et l'exhaustivité des fichiers sources, la documentation du CEREMA ne donne pas d'indicateurs chiffrés, mais une estimation de la fiabilité sous forme de smileys :

Dictionnaire qualité des fichiers MAJIC :

Code couleur	Fiabilité
😊😊😊	Ces données sont très fiables, exhaustives, et ne nécessitent pas d'être confrontées à des données terrains.
😊😊	Ces données sont par construction très fiables, mais peuvent parfois présenter des décalages par rapport à une confrontation terrain (données déclaratives). Ces décalages peuvent être considérés comme non significatifs à une échelle communale.
😊	Ces données sont fiables et utilisables mais peuvent ne pas être exhaustives ou bien présenter des limites d'utilisation.
😞	Ces données peuvent présenter des soucis de réactualisation dans leur modalités les plus fines. Elles peuvent cependant être utilisées grâce à des regroupements, ou des précautions particulières.
😞	Ces données doivent être considérées comme indicatives et leur fiabilité comme incertaine. Les variables ne peuvent être à nouveau croisées avec d'autres données. Il n'est pas conseillé d'utiliser ces données.

Figure 13 Code couleur concernant la fiabilité des données.

En conclusion, la géolocalisation du bâtiment a permis de rapatrier et enrichir le modèle thermique en données.

A savoir :

1. L'empreinte au sol du bâtiment ;
2. La hauteur du bâtiment ;
3. Le nombre d'étage du bâtiment ;
4. La nature des matériaux composant les murs ainsi que la toiture ;
5. La zone climatique du bâtiment ;
6. L'orientation du bâtiment ;
7. La représentation simplifiée d'un modèle 3D du bâtiment.

La partie suivante présente la méthode mis en place pour la reconnaissance des composants de façade afin d'en déduire le pourcentage de surface des menuiseries par façade ainsi que la part des parois opaques.

5. Reconnaissance des composants de façade

L'objectif de cette tâche est de détecter les composants de façade à partir d'une soumission d'images de la part de l'utilisateur. Les composants à détecter sont les suivants :

1. Le contexte ;
2. La façade ;
3. Les menuiseries ;
4. Les portes ;
5. La porte de garage ;
6. La toiture avec comble ;
7. La toiture terrasse ;
8. Les volets.

L'intérêt principal de cette reconnaissance automatisée est de déduire de manière approchée les surfaces des menuiseries extérieures et de parois opaques ainsi que le nombre d'étage des habitations afin de constituer le modèle thermique du bâtiment existant.

La reconnaissance des composants de façade est basée sur une méthode d'intelligence artificielle de réseau de neurones profonds (deep-learning).

Les parties suivantes présentent les différentes méthodes et techniques qui ont permis de créer le modèle de reconnaissance de composants de façade.

5.1. Lexique utilisé dans la source Classification, Détection d'objets et Segmentation d'images :

Labeling : Signifie que nous prenons un ensemble de données d'images non étiquetées et que nous ajoutons des balises significatives et informatives indiquant le contenu de cette image.

Bounding boxes : Sont des boîtes entourant des objets (cadres de sélection).

Classification : Fait référence à un type d'étiquetage dans lequel une image / vidéo est associée à certains concepts, dans le but de répondre à la question « Que contient cette image / cette vidéo ? » Une image peut être classée en plusieurs catégories.

Object Détection : Est une technique de vision par ordinateur qui consiste à distinguer les objets d'une image ou d'une vidéo :

- Application de la classification à des objets distincts dans une image / vidéo;
- Utilisation des cadres de sélection (*Bounding boxes*) pour nous indiquer où se trouve chaque objet dans une image / vidéo.

Segmentation : Est un type d'étiquetage dans lequel chaque pixel d'une image est étiqueté avec des concepts donnés. Les modèles de segmentation fournissent le contour exact de l'objet dans une image. C'est-à-dire que des détails pixel par pixel sont fournis pour un objet donné, par opposition aux modèles de classification, où le modèle identifie le contenu d'une image, et aux modèles de détection, qui placent un cadre de sélection autour d'objets spécifiques.

Source : <https://www.clarifai.com/blog/classification-vs-detection-vs-segmentation-models-the-differences-between-them-and-how-each-impact-your-results>

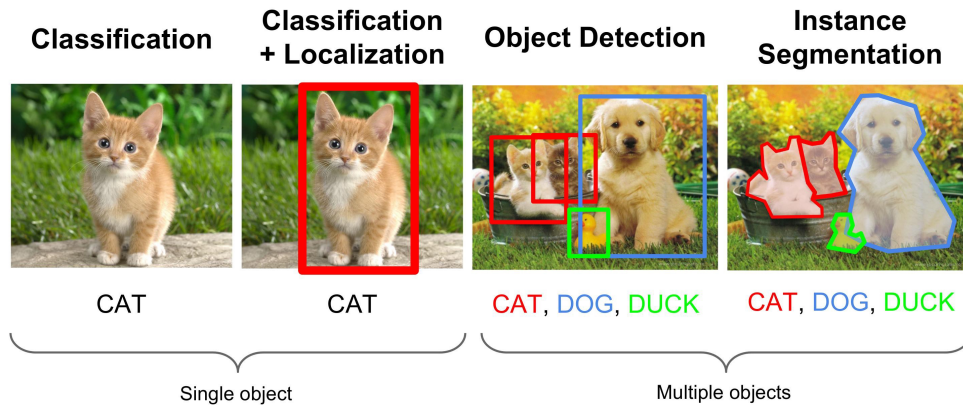


Figure 14 Illustration des différentes méthodes de détection d'objet dans une image.

Modèles	Authors	Exemples
Region-based Convolutional Network (R-CNN)	J.R.R. Uijlings and al. (2012) http://www.huppelen.nl/publications/selectiveSearchDraft.pdf	
Fast Region-based Convolutional Network (Fast R-CNN)	R. Girshick (2015) https://arxiv.org/pdf/1504.08083.pdf	
Region-based Fully Convolutional Network (R-FCN)	J. Dai and al. (2016) https://arxiv.org/pdf/1605.06409.pdf	
You Only Look Once (YOLO, YOLOv2, YOLOv3)	J. Redmon et al., 2016) https://arxiv.org/pdf/1506.02640.pdf	
Neural Architecture Search Net (NASNet)	B. Zoph and Q.V. Le, 2017) https://arxiv.org/pdf/1707.07012.pdf	
Mask Region-based Convolutional Network (Mask R-CNN)	K. He and al. (2017) https://arxiv.org/pdf/1703.06870.pdf	

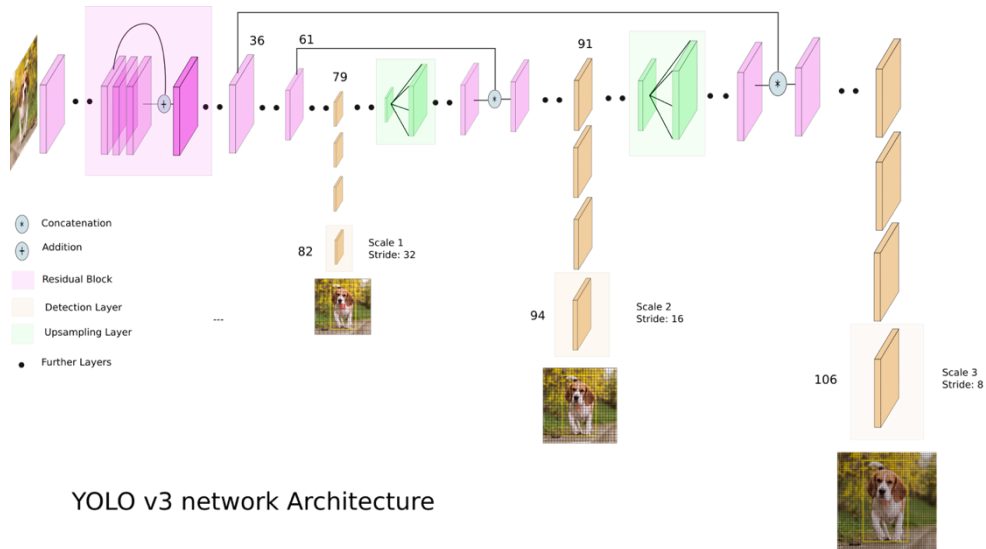
Figure 15 Bibliographie non exhaustive des réseaux de neurones profond utilisé dans la reconnaissance d'objets

5.2. Algorithme de réseaux de neurones pour la détection d'objets : YOLOv3

Yolo, qui veut dire « You Only Look Once », est un réseau de neurones spécialisé dans la détection et l'analyse d'objets dans l'image. Sa grande force est la rapidité : il peut travailler en temps réel (à 45 img/ sec). Le réseau de neurone Yolo est écrit en C (bibliothèque **Darknet**). Une version plus User-Friendly a été développée avec la bibliothèque TensorFlow (framework de machine learning, open source, de Google) : **Darkflow**. Il dépend de **OpenCV** (Open Source Computer Vision Library) qui est une bibliothèque Python spécialisée dans la « vision machine ».

Sources :

1. <http://nxt-marketing.com/ia/reconnaissance-automatique-dimages-en-machine-learning-avec-yolo-python-et-tensorflow/>
2. <https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>



YOLO v3 network Architecture

Figure 16 Architecture du réseau de neurone YOLOv3

5.3. Labélisation / Annotation des images / Apprentissage / Prédictions

Afin de construire la base de connaissance pour le réseau de neurone, il est primordial d'annoter les images de maison individuelle issues d'une base de données de différents sites immobiliers (3589 photos).

Afin d'annoter les images recensées, l'outil labeling (<https://github.com/tzutalin/labelimg>) est employé et permet de définir des cadres de limitation autour des composants de façade (bounding-box).

Une centaine de photos labélisées ont été nécessaire pour l'apprentissage pour le réseau de neurone YoloV3 (voir Figure 17). Les résultats sont présentés Figure 18. Ce premier réseau de neurone permet de détecter les composants de façade. Cependant, il ne permet pas de détourner au pixel près le composant. Il est donc nécessaire de mettre en place une méthode de segmentation sémantique.



Figure 17 Labélisation des images dans labeling.

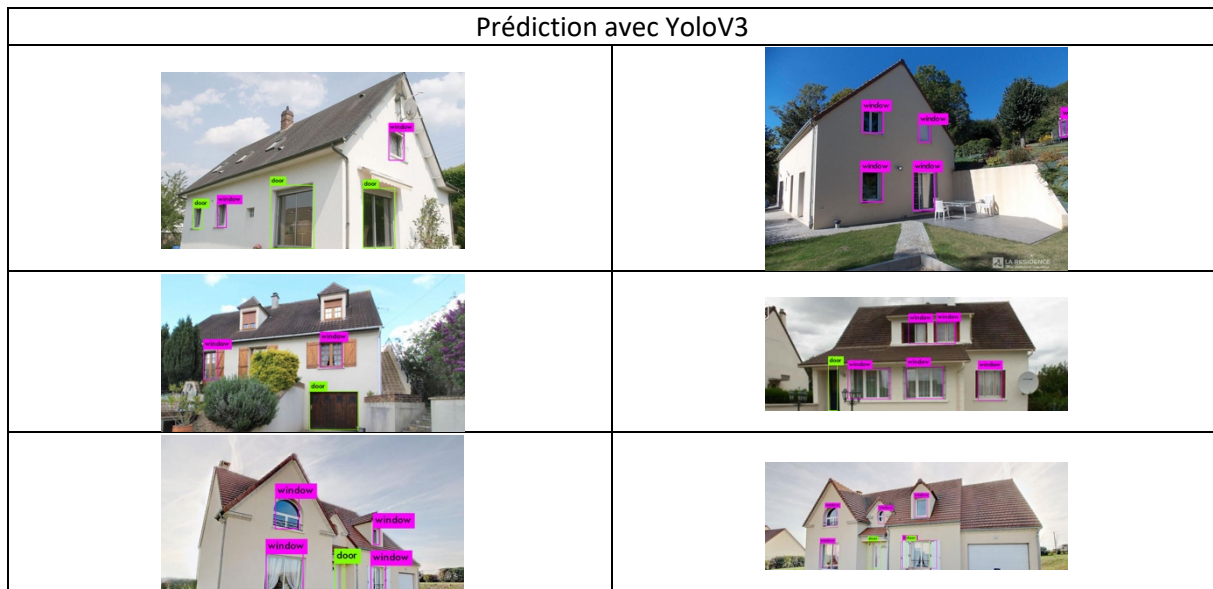


Figure 18 Résultats de la détection des composants de façade avec YoloV3.

5.4. Segmentation sémantique

C'est un algorithme de deep-learning qui associe une étiquette ou une catégorie à chaque pixel d'une image. Il permet de reconnaître un ensemble de pixels qui forment des catégories distinctes (source : <https://fr.mathworks.com/solutions/image-video-processing/semantic-segmentation.html>).

L'outil CVAT Powerful and efficient Computer Vision Annotation Tool (<https://github.com/opencv/cvat>) a permis de détourer deux cents images selon 8 catégories de composants (voir Figure 19)

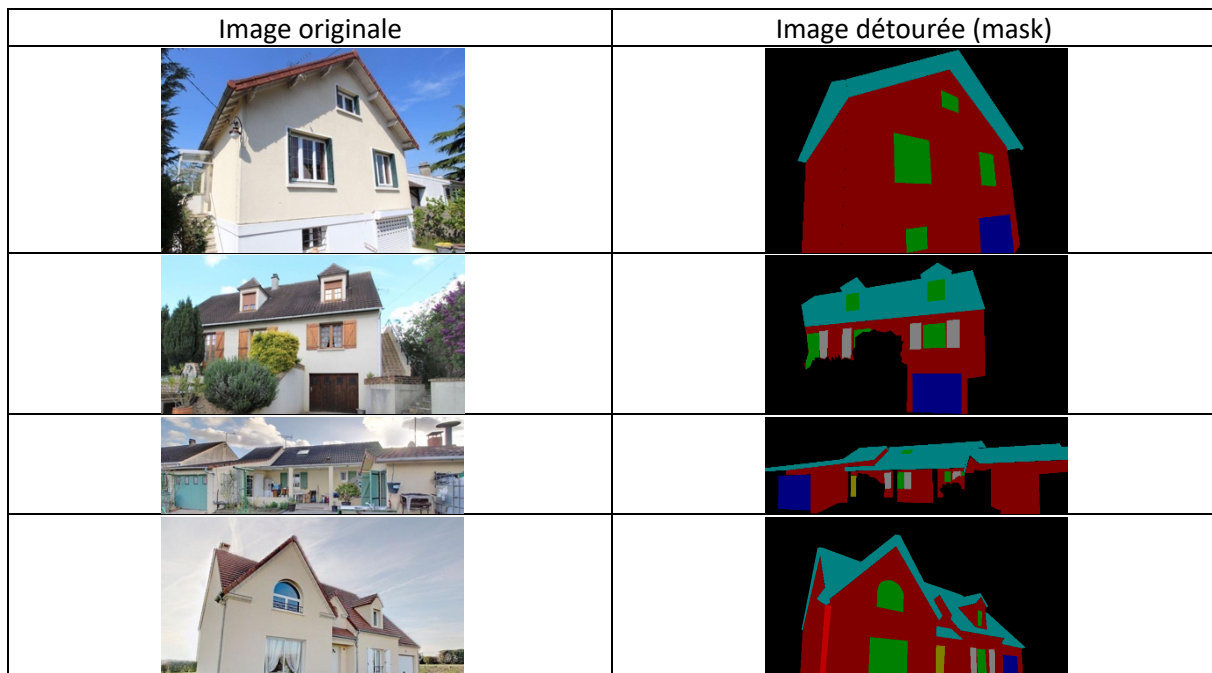


Figure 19 Annotations dans le logiciel CVAT.

Par ailleurs, le réseau de neurone utilisé pour l'apprentissage est basé sur le modèle [DeepLab](#). DeepLab est un modèle de segmentation sémantique de pointe conçu et open source par Google en 2016.

Plusieurs améliorations ont été apportées au modèle depuis => DeepLab V2, DeepLab V3 et le dernier DeepLab V3 +.

Le modèle DeepLab se compose généralement de deux étapes :

1. Phase d'encodage : L'objectif de cette phase est d'extraire des informations essentielles de l'image. Cela se fait en utilisant un réseau neuronal convolutif CNN pré-entraîné (ex : imagenet),
2. Phase de décodage : Les informations extraites en phase de codage sont utilisées ici pour reconstruire la sortie de dimensions appropriées

On trouvera en figure 20 quelques chiffres techniques liés à la labélisation et en figure 21 quelques illustrations de résultats de prédictions.

Nombre d'images annotées	220
Modèle pré-entraîné	Null
Modèle de la GPU	Quadro RTX 6000
Nombre de labels	8
Version CUDA	Driver Version: 430.50 - 10.1
Cœurs NVIDIA Tensor	576
Cœurs de traitement parallèle CUDA	4 608
Bibliothèque logicielle Python	PyTorch (Facebook)
Modèle CNN	DeepLabv3+

Figure 20 Quelques chiffres.


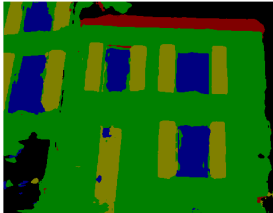



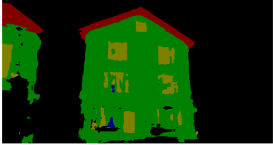






Image originale	Prédiction des composants
	
	
	
	
	
	

Figure 21 Résultats de la prédiction des composants de façade avec DeepLabV3+.

5.5. Estimation des surfaces des composants

A partir de la prédiction des composants de façade issue du réseau de neurone profond, différents post-traitements sont nécessaires. Les traitements sont les suivants :

1. Rectification de l'image : le calcul du point de fuite de l'image permet de rectifier l'image dans un plan orthogonale.
2. Détourage des composants par une succession de différentes méthodes de traitement et d'analyse d'image.



Figure 22 Image originale avant post traitement.

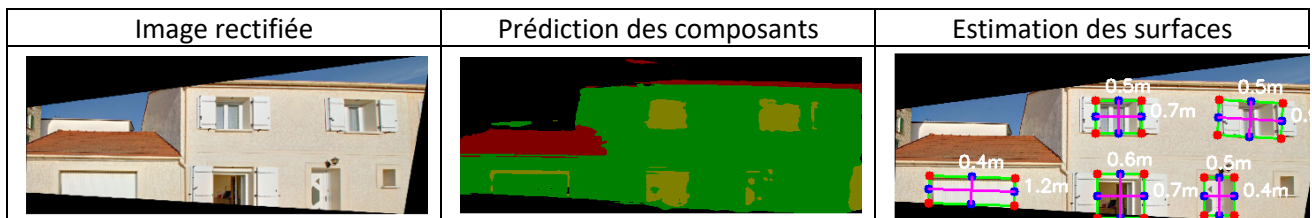


Figure 23 Succession de méthodes de traitement d'images.

En connaissant la hauteur du bâtiment, le service en déduit les différentes surfaces des composants de la façade.

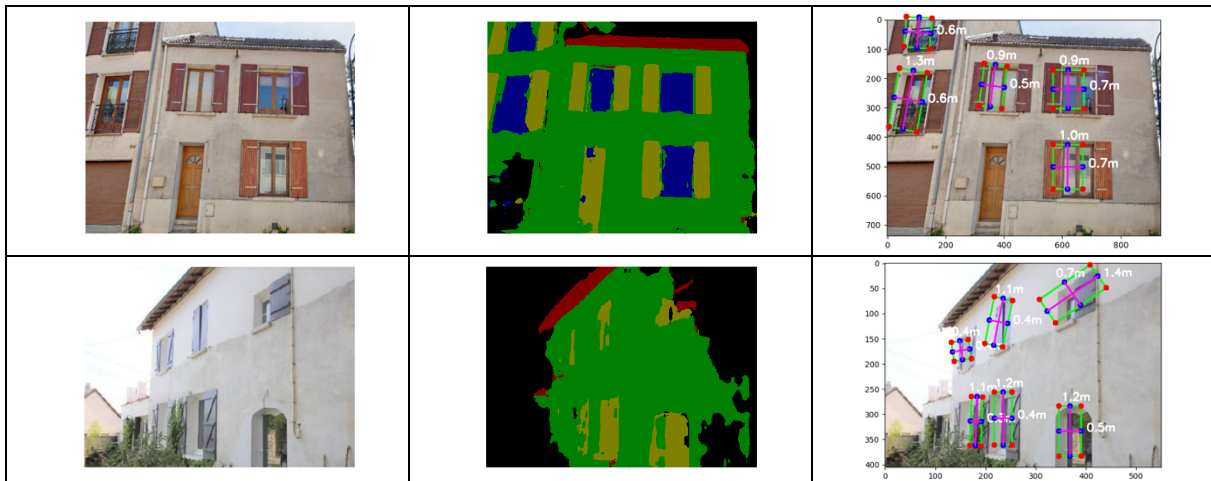


Figure 24 Estimation des surfaces des composants.

5.6. Orientations multiples et interface utilisateur

Concernant les multiples orientations du bâtiment, l'interface incite l'utilisateur de charger autant de photo que le bâtiment a de façades ainsi que l'orientation de la façade.

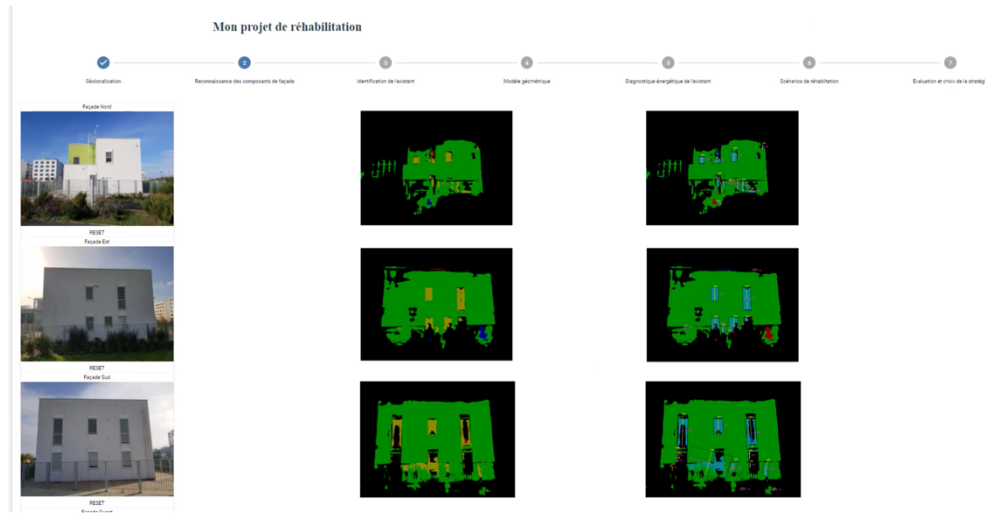


Figure 25 Interface de chargement des photos de façade.

6. Proposition de modèle 2D/3D de l'existant

Dans le cas où le bâtiment n'est pas référencé dans la base de données BD Topo, le service interroge OpenStreetMap avec [Overpass Turbo](#).

L'API Overpass peut être un excellent outil pour la cartographie, car il est très puissant en filtrant les données OSM. Avec Overpass Turbo, il est facile d'exécuter rapidement des requêtes Overpass et d'inspecter les résultats d'une manière conviviale.

Voici quelques exemples requêtes démontrant la praticité cartographique d' « Overpass turbo » :

- Rechercher, sur une grande surface, de (rares) fautes d'orthographe et des erreurs de casse en respectant une convention de nommage.
- Rechercher des POI spéciaux qui ne sont pas dessinés sur la carte.
- Analyser des POI (par exemple, les nœuds "place") pour vérifier s'ils sont uniformément répartis sur de grandes surfaces.
- Afficher spatialement de grandes entités (les frontières, les rivières, les autoroutes complètes, points de connexion, ...) puis les charger [directement](#) dans un éditeur.
- Filtrer les données OSM que l'on souhaite visualiser.

La Figure 26 présente une [requête Overpass](#) qui recherche les bâtiments dans une fenêtre de recherche. La réponse à la requête renvoie des données intrinsèques au bâtiment tel que la hauteur et la nature de la donnée.

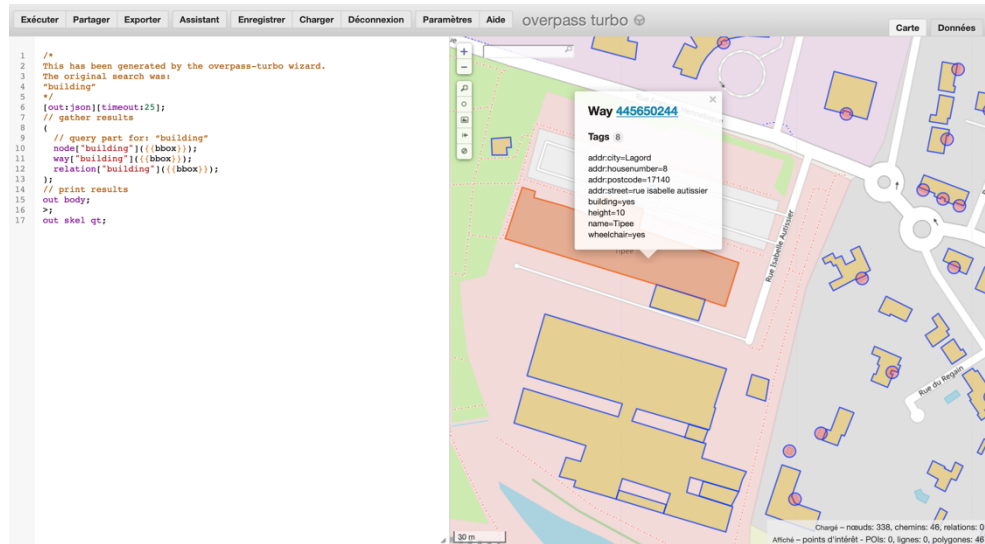


Figure 26 Exemple de requête Overpass.

Un service automatisé est dédié à l'interrogation de BD TOPO et de l'API Overpass Turbo pour permettre la construction du modèle géométrique 3D.

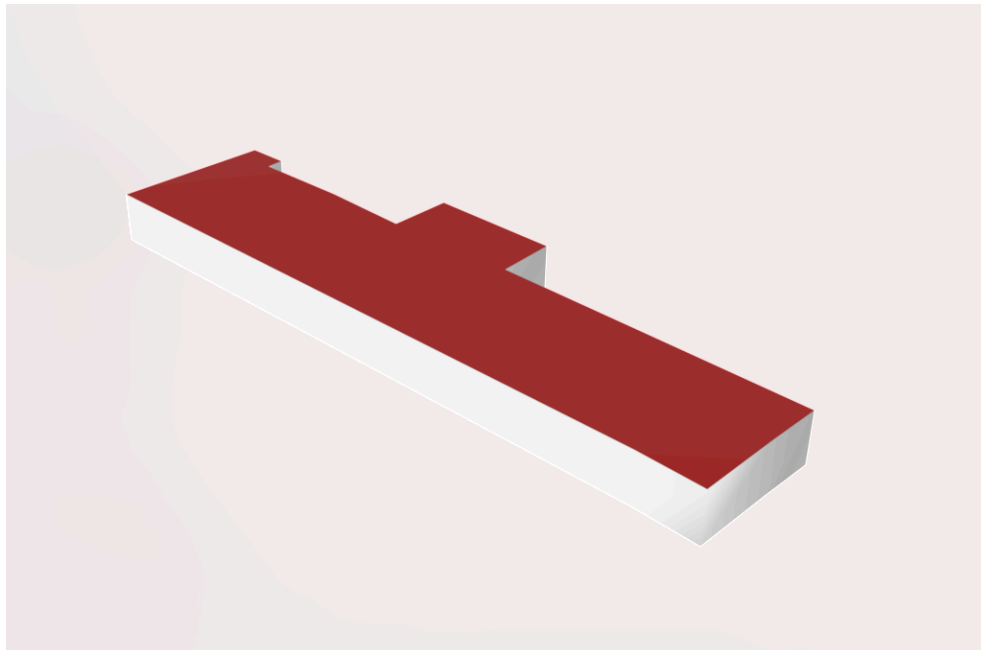


Figure 27 Représentation du bâtiment en .OBJ à partir d'une requête Overpass.

7. Diagnostic énergétique de l'existant

A partir des informations récupérées sur l'enveloppe du bâtiment (chapitre précédent), le service interroge une base de données construite spécifiquement et notamment issue de l'outil en ligne [TABULA](#). TABULA est une méthodologie visant à évaluer les performances énergétiques du parc résidentiel. Elle regroupe des typologies de bâtiments résidentiels et une méthode de calcul pour évaluer les consommations énergétiques pour le chauffage ainsi que leurs réductions par des mesures de rénovation énergétique.

Le service qui évalue le diagnostic énergétique de l'existant récupère essentiellement les données sur les typologies constructives des bâtiments résidentiels à partir d'une période constructive fournie par BD Topo (voir Figure 28). Des valeurs cibles pour les coefficients U des éléments de l'enveloppe thermique sont ainsi déduites.

Par la suite, un modèle thermique est généré au format XML RT2012 et un calcul des besoins énergétiques du bâtiment est effectué avec le moteur de calcul réglementaire du CSTB.

Classe bâtiment		Maison individuelle détachée SFH	Maison individuelle mitoyenne TH	Petit logement collectif (<10 log.) MFH	Grand logement collectif (≥10 log.) AB
Période constructive					
1	Avant 1915				
2	1915 - 1948				
3	1949 - 1967				
4	1968 - 1974				
5	1975 - 1981				
6	1982 - 1989				
7	1990 - 1999				
8	2000 - 2005				
9	2006 - 2012				
10	après 2012				

Figure 28 Typologie des bâtiments résidentiels en France dans le cadre du projet TABULA et EPISCOPE (<https://episcope.eu/welcome/>)

8. Conclusion

Ce document présente la plateforme SAAS qui est le fil conducteur des travaux menés au travers de la tâche données exploitables et la tâche diagnostic énergétique automatisé des bâtiments existants du volet 2. Un autre document présentera la tâche des outils d'évaluation des apports des solutions de rénovation ainsi que la tâche concernant la construction automatique de métrés et de devis.