

watſ up

CAMILLE CÉLESTE COVAREL

DWWM · décembre 25

Watts Up ?

Watts Up est une plateforme de réservation de bornes de recharge pour les usagers de véhicules électriques.

Lineup



Jonathan Dias

Dev FullStack / Scrum Master
orienté front-end



Camille Céleste Covarel

Dev FullStack / Product Owner /
Tech Lead
orienté back-end

Pour quels besoins ?

En tant que visiteur

Découvrir le service

Voir les stations disponibles

En tant que client

Réserver une borne

Se géolocaliser

Avoir son espace personnel

Contacter le service client

En tant qu'administrateur

Gérer les stations / bornes

Gérer les clients

On travaille pour quoi ?

Créer un MVP à présenter à la
société fictive GeoCode

Stack technique

FrontEnd

Vite.js
Maplibre
TanStack
Query

BackEnd

Node.js
Express
Sequelize
+ cli

BDD

Postgresql
Postgis

Déploiement

Docker
Github
GHCR
Github action
Nginx

Maquettage de l'app



The image displays five wireframe prototypes of a mobile application interface, each with a yellow header containing a location icon and the word "GEOCODE". Below the header, the slogan "Notre super slogan" is present.

- Screen 1: Map View**

A map showing several locations with blue and red location markers. Labeled streets include "Bd de Strasbourg", "All. Jean Jaurès", "E DE JSE", and "SAINT-GEORGES". A search bar at the top has a magnifying glass icon. A bottom navigation bar contains icons for battery, calendar, user, email, and help.
- Screen 2: Station Details**

Shows a station named "Robert" located at "12 Rue du grand Robert 31200 ROBERT". It lists four charging bays with their respective prices and times:
 - Bay 1: ₣10 | 4:4
 - Bay 2: ₣20 | 2:4
 - Bay 3: ₣30 | 3:4
 - Bay 4: ₣40 | 0:4Buttons for "Réserver" and "Annuler" are at the bottom, along with a bottom navigation bar.
- Screen 3: Reservation Confirmation**

Text: "Merci de verifier vos informations avant de finaliser la réservation." (Please verify your information before finalizing the reservation.)

Détails de la réservation
Station : Robert
Adresse : 12 Rue du grand robert 31200
ROBERT
Borne :
 - Type de prise
 - 40 kwCréneau de réservation : 16h30 - 17h00
Paiement : CB

Buttons: "Valider ma réservation" (dark grey) and "Annuler ma réservation" (light grey).
- Screen 4: Confirmation and Next Steps**

A large checkmark icon is centered. Text: "Votre réservation à bien été prise en compte !"
"Vous allez recevoir par e-mail toutes les informations sur votre réservations."
"Pour gerer vos reservations :"
A button "Voir mes réservations" is at the bottom.
- Screen 5: Reservation History**

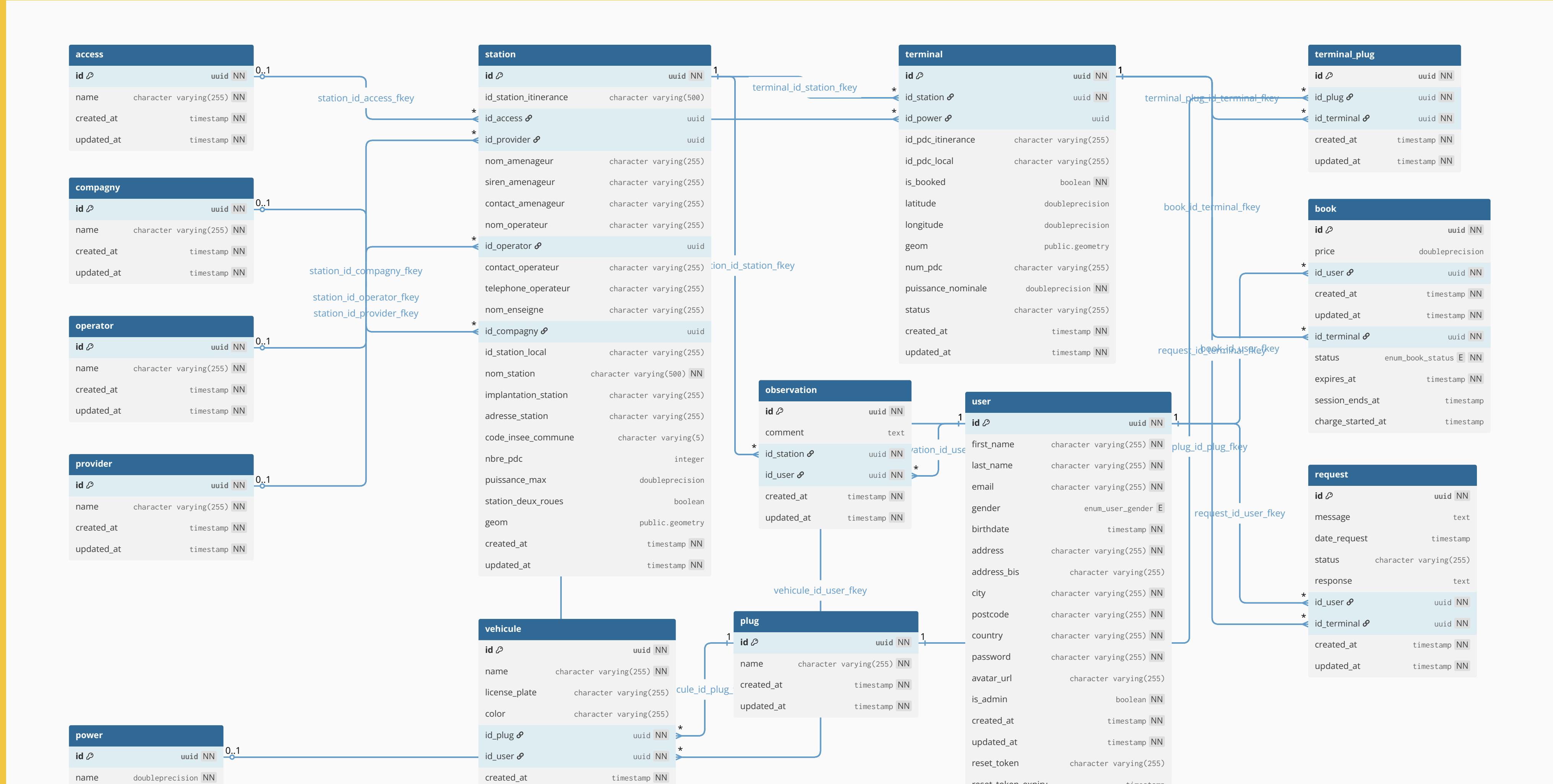
Section title: "Réservation en cours".

Détails de la réservation (with a close button "X")
Station : Robert
Adresse : 12 Rue du grand robert 31200
ROBERT
Borne :
 - Type de prise
 - 40 kwCréneau de réservation : 16h30 - 17h00
Paiement : CB

Section title: "Historique de réservations"
List items: "12 Juin 2025", "06 Juin 2025", and "03 Juin 2025". Each item has a dropdown arrow icon.

Bottom navigation bar.

Modélisation BDD



page 17
du dossier
de projet

PostgreSQL + PostGIS

Après analyse de la data, veille et discussion avec Tristan Laroye, PostGIS a été une évidence.

Addon spécialisé dans les coordonnées géospatiales, PostGIS nous a permis de déployer des tuiles vectorielles.

MongoDB + GeoJSON

Requêtes géospatiales natives mais très différent niveau structure.

Elasticsearch Geo

Performant mais overkill et lourd

PostgreSQL + PostGIS

Migration nécessaire mais tuiles vectorielles donc très bonne performance à peu de charge.

Import CSV, Le défi

Mon gros défi sur ce projet. Importer plus de 170k lignes (35.5k stations, 136k bornes).

J'ai réalisé cette tâche en suivant les préceptes des transactions ACID

atomicité, cohérence, isolation et durabilité

Traitement ligne par ligne

fs.createReadStream

~ ~ ~ ~ ~

Validation et clean de la data

dataTransformer.ts

~ ~ ~ ~ ~

Consolidation en mémoire

Stations avec bornes pré-validées dans une map

(les relation simples sont gérés par un findOrCreate)

~ ~ ~ ~ ~

Mise en cache

Pour éviter les transactions inutiles, la fonction sait où elle en est

~ ~ ~ ~ ~

5000 lignes or 2000 stations

Une fois atteint ce palier, transaction ACID vers BDD

pages 18 à
19 du dossier
de projet

Transaction & Rollback

Si un terminal ou un autre élément
échoue, la transaction est annulée.
l'erreur est loggée dans une fonction
custom.

Cela évite d'avoir des données
incomplètes et protège la BDD.

page 20
à 21 du
dossier de
projet

Monitoring WebSockets

L'administrateur possède une vision directe sur cet import. Grâce aux WebSockets, il voit en direct :

La progression de l'import

Les erreurs

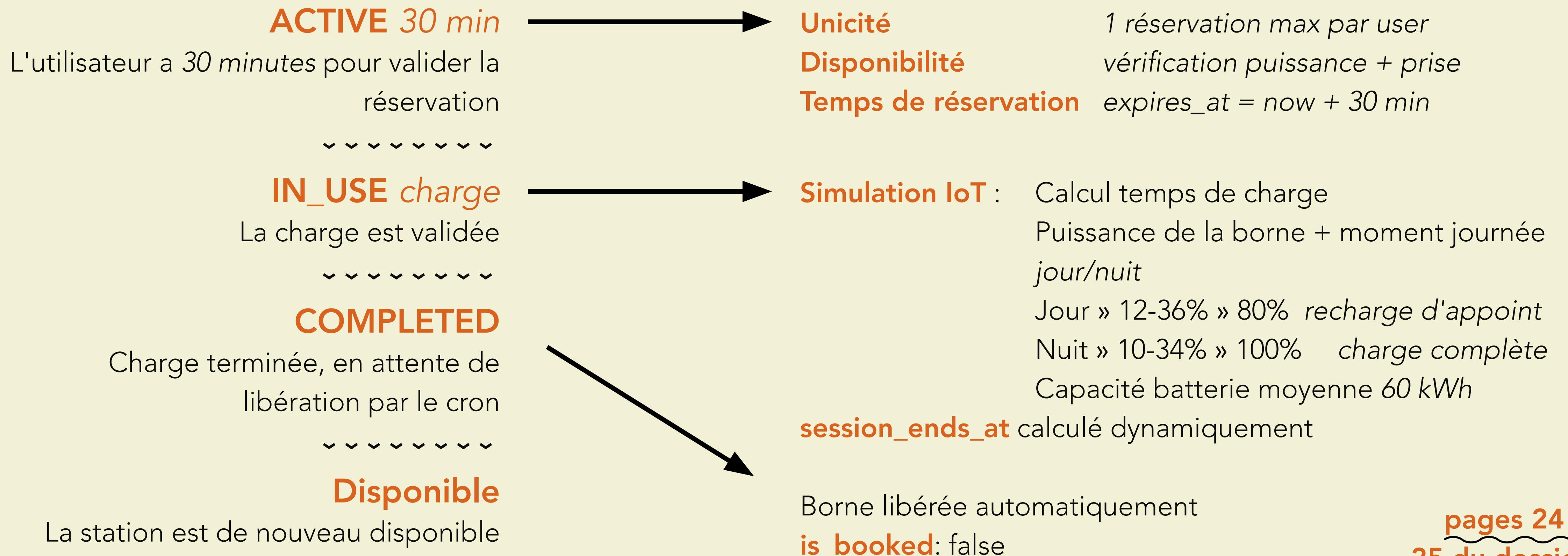
La durée

Le statut final

Une fonction de réhydratation est également présente pour qu'il puisse récupérer les informations à n'importe quel moment.

page 22
du dossier
de projet

Système de réservation



pages 24 à
25 du dossier
de projet

Architecture adaptative

watts UP



Jean Jaurès

16 allée Jean Jaurès 31000 Toulouse FRANCE

Implantation: Parking public
Horaires: 24/7
Accès PMR: Oui
Points de charge: 67

Choisir ma borne

 7.4 kW 60/60	 3.7 kW 7/7
---	--

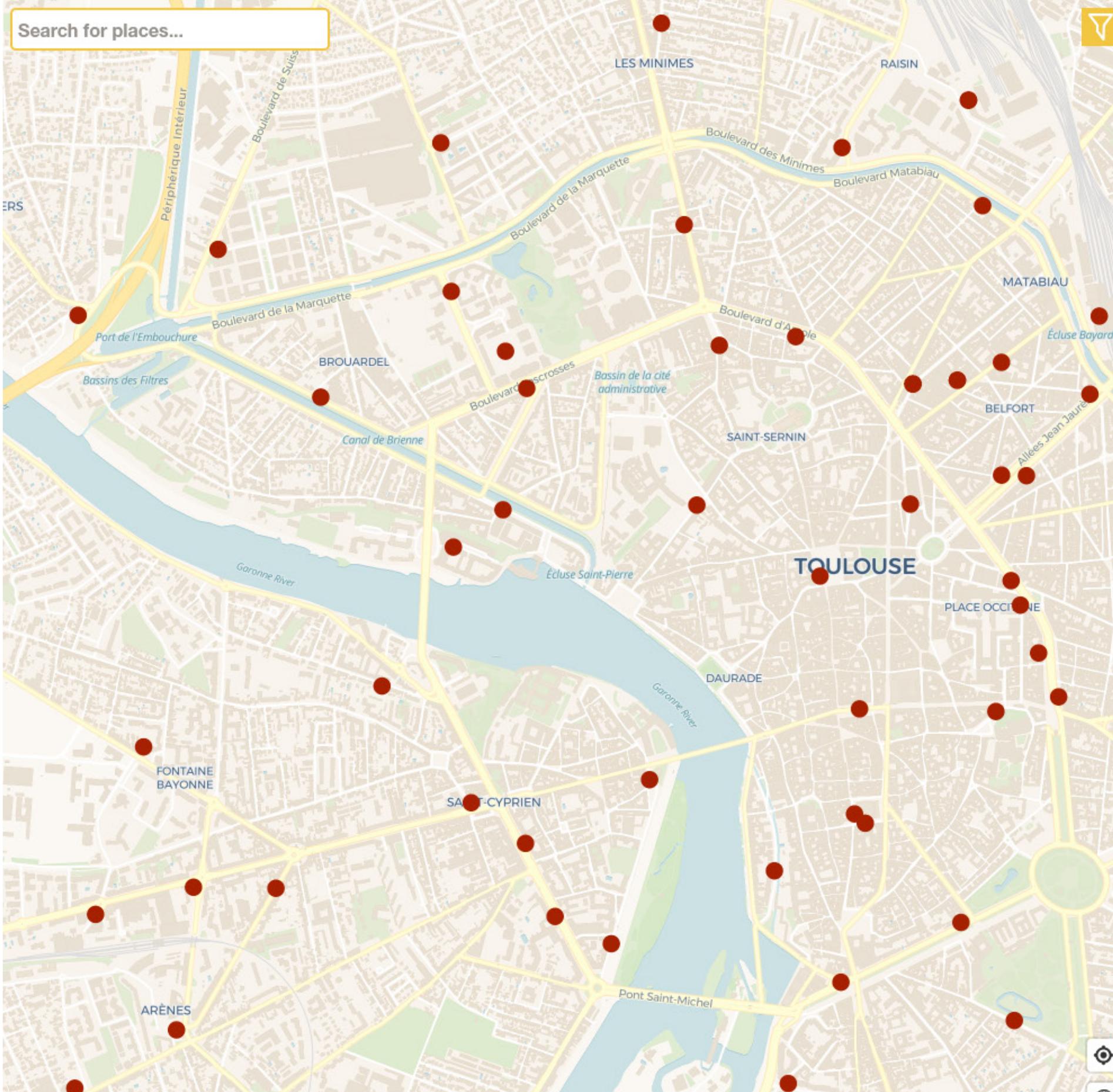
    

watts UP



Carte Mes réservations Assistance Informations Dashboard Admin

Search for places...



Parc Victor Hugo

Place Victor Hugo 31500 Toulouse FRANCE

Implantation: Parking public
Horaires: 24/7
Accès PMR: Oui
Points de charge: 26

Choisir ma borne

 7.4 kW 19/19	 3.7 kW 1/1
 3.7 kW 6/6	

Réserver Retour à la carte Supprimer la station

page 28 à 30 du dossier de projet

Cycle de vie réservation

Pour plus de clarté, les bornes sont agrégées côté client avec `useMemo` sous le format

22 kW - Type 2 - 1/8
puissance - type prise - dispo/total

Les timers sont en temps réel
Avec le status ACTIVE, on a un compte à rebours

Avec le status IN_USE, on a un chronomètre `setInterval` avec l'heure de fin de charge
Simulation IoT

page 32
du dossier
de projet

Feedback utilisateur/erreurs

L'application possède des Toasts et Modales branchés directement sur les mutations de TanStack Query

Les Toasts sont utilisés pour les erreurs et succès

Les Modales sont utilisés pour les actions sensibles avec validation explicite
réservation, annulation

Un autre type de feedback utilisateur concerne les alertes affichées en surimpression

Fichier trop volumineux (max 1 Mo)

pages 32 du dossier de projet

watts up

DÉMO LIVE

watts-up.anamnesis.ovh

Jeux d'essai

<u>Sélection</u>	<u>Réservation</u>	<u>Charge</u>	<u>Double réservation</u>	<u>Mauvaises données</u>
stations madrid	tentative réservation	démarrage charge	seconde réservation	rejet de la ligne + log
~ ~ ~	~ ~ ~	~ ~ ~	~ ~ ~	~ ~ ~
bornes affichées	minuteur 30 minutes	chrono actif	blocage + message	rejetée et loggée
~ ~ ~	~ ~ ~	~ ~ ~	~ ~ ~	~ ~ ~
nominale	nominale	nominale	nominale	nominale

page 35 à 41 du dossier de projet

Veille & Sécurité

Maplibre vs Leaflet

Rendu GPU / Rendu DOM

Meilleure fluidité pour MapLibre

Sequelize vs Prisma vs SQL

Le SQL vanille n'était pas envisageable au vu de l'équipe et du projet. Prisma a un langage propre. Sequelize nous assurait un traitement sécurisé et rapide de l'application.

PostGis

MapLibre

Sequelize

Sequelize-Cli

UUID

JWT httpOnly

Rate Limiting

tuiles vectorielles

rendu GPU performant

anti-injection SQL

migration versionnée

anti-énumération

anti-XSS

anti-brute force

pages 46 à
47 du dossier
de projet

Déploiement & Bilan

GitHub Actions CI

Lance l'intégration quand **PR** ou **Push** sur
branche staging



Technique

L'application est déployée automatiquement en production et est fonctionnelle. La solution est performante.

Build

Tests & images versionnées
(SHA commit)



Apprentissages

Une belle réussite sur le traitement de la data.

La découverte de PostGis et PostgreSQL avec maplibre.

Déploiement CD

21 secrets sont nécessaire pour déployer l'application **sur 2 CTs**



Perspective d'évolution

Ajout de tests pour chaque feature.

Ajout des refresh token et du 2FA.

Mise en conformité RGPD.

etc.

page 42,43
et 50 du
dossier de
projet

l'application est disponible

watſup

MERCI