

Table des matières

[.. Objet du document 2](#_Toc206581906)

[.. Objectifs du projet 2](#_Toc206581907)

[.. Principes d’architecture 2](#_Toc206581908)

[.. Architecture existante 2](#_Toc206581909)

[.. Architectures 2](#_Toc206581910)

[fonctionnel. 2](#_Toc206581911)

[De données 3](#_Toc206581912)

[Technologiques 3](#_Toc206581913)

[.. Justification de l’approche architecturale 3](#_Toc206581914)

[.. Architectures de transition 3](#_Toc206581915)

# .. Objet du document

Ce document permet de modéliser l’architecture qui sera réalisée par les équipes de développement. Au-delà de la modélisation graphique, il s’agit également d’énoncer les principes sur lesquels l’architecture s'appuie, de justifier cette approche architecturale, mais également d’indiquer des transitions si nécessaire.

# .. Objectifs du projet

L’application Your Car Your Way a pour but de permettre à un utilisateur de pouvoir, à la suite d’une authentification, choisir un véhicule de location dans le but de se rendre d’un emplacement et moment X à un emplacement et instant Y. Pour cela, l’utilisateur cherchera dans une interface des véhicules correspondant à sa demande et pourra en réserver un. A l’heure actuelle, plusieurs applicatifs dans plusieurs langues permettent déjà la résolution de cette problématique. Cependant, dans un soucis d’uniformiser l’application, une nouvelle application multilingue centralisera l’ensemble des demande des utilisateurs, peu importe leur langue et emplacement géographique.

# .. Principes d’architecture

Pour réaliser l’application, il va nous falloir deux sections majeures :

* Un backend réalisé avec **Java / Spring Boot** disposant d’un ensemble d’APIs servant à la résolution des requêtes :
  + Une API permettant l’authentification des utilisateurs
  + Une API permettant la recherche et la sélection d’un véhicule
  + Une API servant à la résolution d’une réservation et au référencement du paiement final
  + Une API pour l’assistance litige asynchrone (par mail / messages sur la plateforme)
  + Une API pour le chat lors d’une demande d’assistance immédiate via conseiller
* Un frontend réalisé avec **Angular** servant à l’affichage d’une interface homme-machine adaptée à la langue du navigateur comportant plusieurs modules de sorte à permettre du lazy loading :
  + Module d’authentification
    - Page de connexion / enregistrement
  + Module de visualisation des véhicules
    - Page de référencement des véhicules disponible avec filtres et barre de recherche
  + Module de réservation
    - Page de réalisation d’une réservation
    - Page de paiement
  + Module de gestion utilisateur
    - Page de profil utilisateur servant à en modifier les informations ou à supprimer le compte
    - Page de réinitialisation du mot de passe
  + Module d’assistance litige
    - Page d’assistance regroupant des tickets d’assistance par message / mail et la capacité à demander une assistance instantanée par chat ou via un envoi de mail
    - Page de suivi des messages du chat

En réalisant une séparation des fonctionnalités et leur regroupement dans des sous-ensemble, on peut permettre la mise en place progressive d’une architecture de type micro-service au niveau du backend. Cette architecture permettra un développement facilité des fonctionnalités et leur mise en service plus rapide. Pour éviter d’avoir une sécurité faillible et d’avoir besoin d’un certificat TLS pour chaque micro-service, on procèdera à la mise en place d’une Gateway qui se chargera de rediriger les requêtes HTTPS provenant des clients (extra-serveur) vers les microservices par le protocole HTTP (intra-serveur).

Au niveau de l’application frontend, l’utilisation de plusieurs modules devrait permettre également un développement plus aisé et aussi un chargement plus rapide de l’applicatif au niveau du navigateur. Il serait de bon ton de mettre également en place du SSR (Server-Side Rendering) de sorte à obtenir un référencement plus rigoureux de notre application. L’ajout de SSR permettrait également de traiter en interne la récupération de la langue utilisée par le client dans le but de lui fournir des pages web adaptées à ses besoins directement.

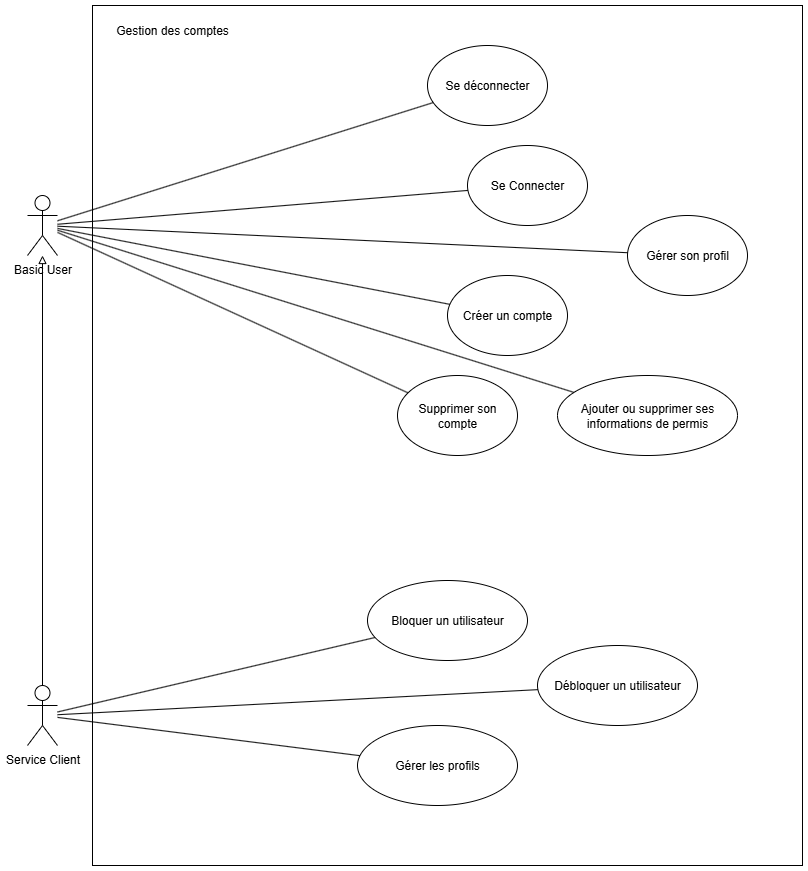
# .. Architecture existante

Le projet ne s'appuie pas sur un produit existant qui serait repris puis mis à jour. Il n’y a donc pas d’architecture existante à définir.

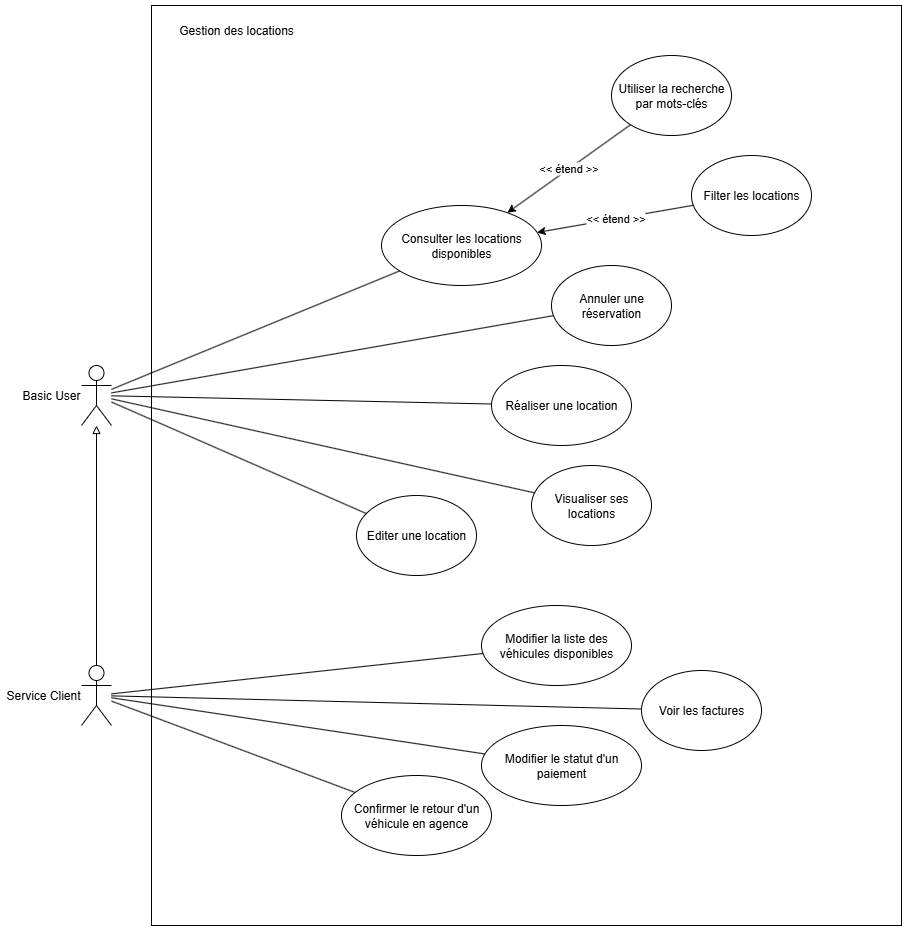
# .. Architectures

## Métier

### Authentification et Gestion des comptes



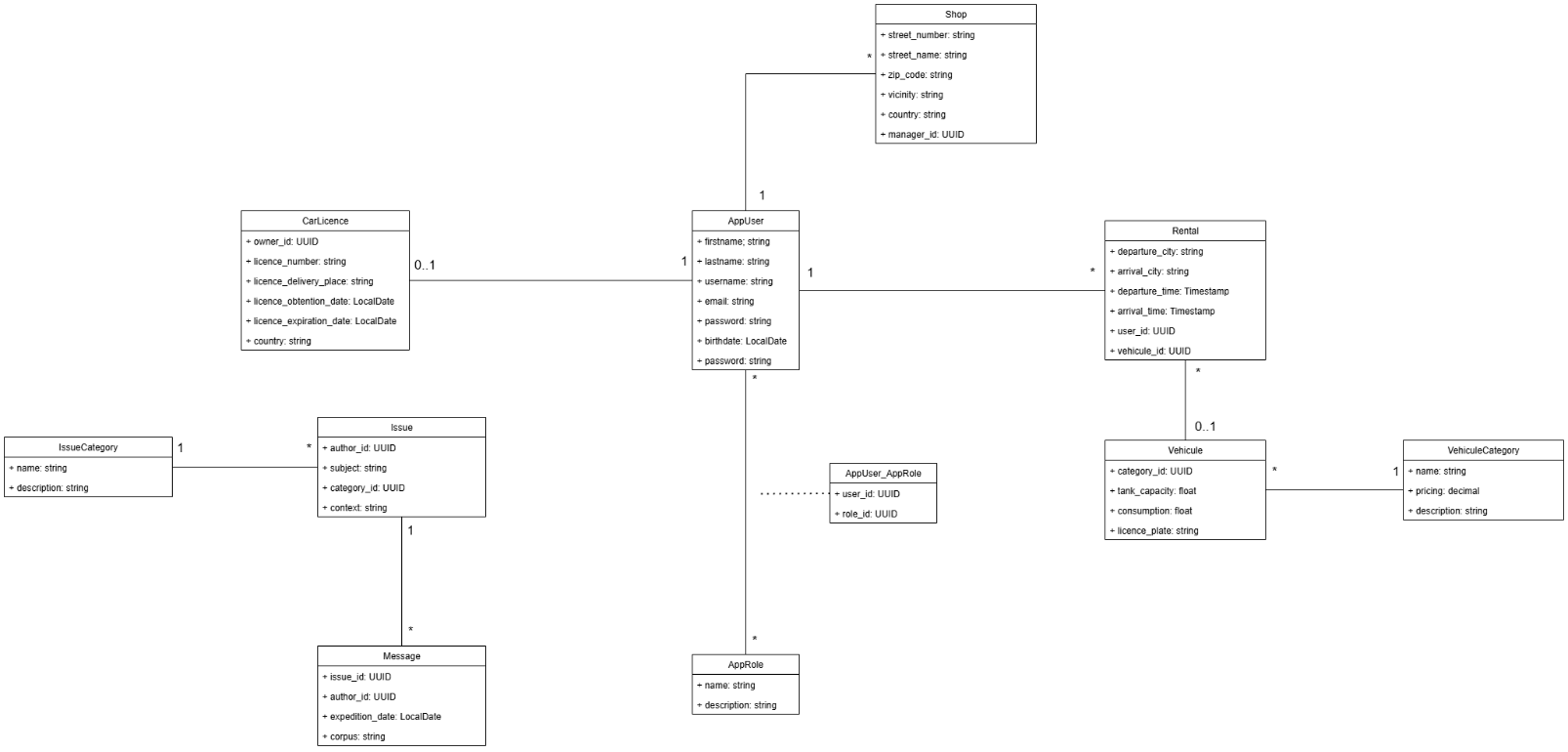
### Gestion des locations



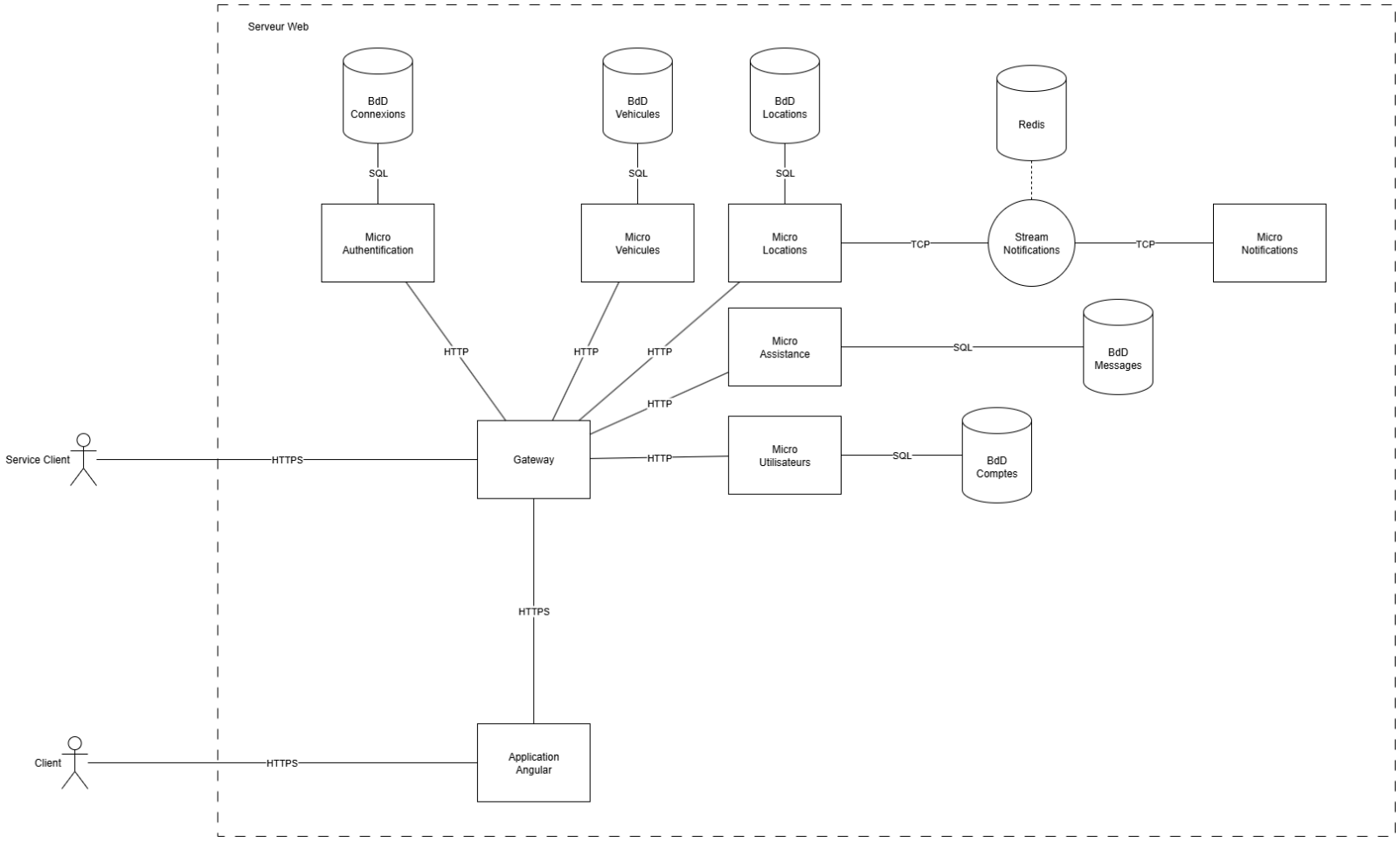
### Assistance et litiges



## De données



## Technologiques



# .. Justification de l’approche architecturale

*[ Explication de la manière dont l’architecture modélisée précédemment répond aux objectifs, respecte les principes d’architecture, et suit les normes et bonnes pratiques reconnues. ]*

# .. Architectures de transition

*[ Cette section est optionnelle – si la mise en œuvre complète de toute l’architecture revêt une forte complexité, la définition d’architectures de transition, pour passer du point de départ à l’architecture finale à travers différentes étapes progressives, peut être envisagée. ]*

