

# TransitCrunch

Projet de valorisation des données GTFS temps réel de la STM

Projet #035

Version :

Date d'émission : 2019-04-22

Date de révision : 2019-04-22

---

## Rapport technique final

---

### Auteurs

Francis De Vizio  
Mathieu Poulin  
Martin Satakem Mbougho  
Simon Thériault

### Présenté à

Alain April



De Vizio, Poulin, Satakem Mbougho, Thériault 2019



**Attribution - Partage dans les Mêmes Conditions  
CC BY-SA**

Cette licence permet aux autres de remixer, arranger, et adapter votre œuvre, même à des fins commerciales, tant qu'on vous accorde le mérite en citant votre nom et qu'on diffuse les nouvelles créations selon des conditions identiques. Cette licence est souvent comparée aux licences de logiciels libres, "open source" ou "copyleft". Toutes les nouvelles œuvres basées sur les vôtres auront la même licence, et toute œuvre dérivée pourra être utilisée même à des fins commerciales. C'est la licence utilisée par Wikipédia ; elle est recommandée pour des œuvres qui pourraient bénéficier de l'incorporation de contenu depuis Wikipédia et d'autres projets sous licence similaire.

## RÉSUMÉ

Les sociétés de transport collectif comme la STM offrent désormais des données ouvertes qui permettent de connaître en temps réel les positions et les temps de passage des véhicules. Le projet Transitcrunch vise à élaborer un système pour archiver, traiter et analyser ces données pour les rendre utiles au public, par exemple au moyen d'indicateurs de performance du réseau ou encore de prédictions à partir de données historiques.

# Table des matières

<b>Historique des révisions</b>	<b>6</b>
<b>Introduction</b>	<b>7</b>
Contexte	7
Objectif du projet	7
Portée	8
Exclusion du projet	8
Définitions, acronymes et abréviations	8
Références	9
<b>Positionnement</b>	<b>10</b>
Énoncé du problème	10
Positionnement du produit	11
<b>Descriptions des intervenants et des utilisateurs</b>	<b>12</b>
Résumé des intervenants	12
Résumé des utilisateurs	12
Environnement utilisateur	12
Principaux besoins des intervenants et utilisateurs	13
<b>Vue d'ensemble du produit</b>	<b>15</b>
Perspective du produit	15
Principaux avantages	15
Hypothèses et dépendances	16
Licences et installation	16
<b>Caractéristiques du produit</b>	<b>17</b>
CAR01 – Afficher la position des véhicules d'un réseau de transport en temps réel sur une carte.	17
CAR02 – Visualiser l'évolution de l'offre de service	18
CAR03 - Calculer des statistiques sur les objets du système	19
CAR04 – Mise à jour et transformation des données	20
CAR05 – Vue d'ensemble de la performance actuelle d'un réseau	21
CAR06 – Vue d'ensemble de la performance historique d'un réseau	22
CAR07 – Afficher une approximation de l'heure à laquelle le prochain véhicule arrivera à partir des données temps réel	23
CAR08 – Prédiction des temps de parcours en temps réel	24
<b>Caractéristiques non fonctionnelles</b>	<b>25</b>
Exigences de performance	25
Exigences de sécurité	25
Exigences de disponibilité	25

<b>Contraintes</b>	<b>25</b>
<b>Facteurs qualité pour l'environnement de production</b>	<b>25</b>
<b>Attributs des caractéristiques</b>	<b>26</b>
<b>Échéancier récapitulatif du projet</b>	<b>27</b>
<b>Méthodologie de travail</b>	<b>28</b>
11.1 Gestion	28
11.2 Outils	28
11.3 Technologies	29
<b>Conclusion et recommandations</b>	<b>31</b>
<b>Annexes</b>	<b>33</b>
Annexe A - Schéma relationnel	33
Annexe B - Attributs des caractéristiques	34
Annexe C - Description des composants techniques du projet	35
1. Base de données	35
2. API REST	35
3. Application Web	40
4. Module de mise à jour des données	40
Annexe D - Statistiques/métriques	41
Annexe E - URLs de l'application web (front-end)	43
Annexe F - Cycle de vie GTFS	45
Annexe G - Maquette de l'interface	46
Annexe H - Architecture modulaire de l'API	47
Annexe I - Architecture d'un module de l'API	47

## Historique des révisions

Date	Version	Description	Auteur
2019-01-14	0.1	Document initial. Objectifs, termes, énoncé du problème, intervenants.	Simon Thériault
2019-01-23	0.2	Besoins, positionnement, contraintes, annexe Description des composants techniques du projet.	Simon Thériault
2019-01-29	0.3	Caractéristiques	Simon Thériault
2019-01-30	0.4	Caractéristiques non-fonctionnelles	Mathieu Poulin
2019-02-01	0.5	Ajout des annexes D, E, F, G	Mathieu Poulin
2019-04-13	0.6	Ajout de diagrammes pour les caractéristiques	Simon Thériault
2019-04-16	0.7	Ajout de diagrammes pour les caractéristiques et ajout des facteurs qualité pour l'environnement de production.	Simon Thériault
2019-04-21	0.8	Complétion de l'annexe C : API	Francis De Vizio
2019-04-22	1.0	Version Finale	Équipe

# 1. Introduction

## 1.1 Contexte

Ce document présente une analyse du positionnement, des caractéristiques de haut niveau et des contraintes d'un projet de fin d'études consistant au développement d'un système pour l'exploitation des données ouvertes de géolocalisation d'autobus publics.

Ces données ouvertes sont offertes dans le format GTFS temps réel par plusieurs sociétés de transport en commun à travers le monde et présentent l'état du réseau de transport (habituellement les autobus) au moment où la requête est effectuée.

GTFS temps-réel est une extension de la spécification GTFS, qui a originalement été développée par Google pour que les sociétés de transport puissent ouvrir au public leurs données d'horaires. Il est donc nécessaire d'utiliser ces deux sources de données complémentaires.

Ce document présente les éléments du logiciel du point de vue d'une partie prenante orientée affaires et d'un utilisateur du produit final.

## 1.2 Objectif du projet

L'objectif du projet est d'exploiter la nouvelle API temps-réel de la STM pour offrir une vue sur l'état actuel et historique du réseau de transport, notamment au moyen d'indicateurs de performance. L'historique de ces métriques pourrait également servir à prédire la performance future d'un réseau de transport en commun. Les tâches envisagées pour ce projet sont:

- Définir un modèle de données et mettre en place un système de base de données.
- Implémenter un client pour le téléchargement automatisé des données vers la base de données.
- Élaborer une API REST qui exposera des fonctionnalités d'analyse et d'exploration des données récoltées.

- Créer un tableau de bord web qui pourra présenter les données d'analyse à un usager sous la forme de tableaux, cartes et graphiques.

### 1.3 Portée

Ce document se limite à documenter le cheminement du projet de son initiation jusqu'à sa livraison.

### 1.4 Exclusion du projet

Les points suivants ne feront pas partie du projet :

- L'intégration des données provenant de l'extérieur de la région de Montréal.
- L'API développée ne sera qu'en lecture seulement pour ses utilisateurs. Aucune donnée ne pourra être ajoutée ou modifiée via l'API. Un module séparé fait le traitement et la mise à jour des données utilisées par l'API.
- Fonctionnalités de planification d'itinéraires. De nombreuses applications existent déjà pour ce cas d'utilisation.

### 1.5 Définitions, acronymes et abréviations

Terme	Définition
GTFS	General Transit Feed Specification. Spécification développée par Google pour que les sociétés de transport puissent s'interfacer avec leur produit Google Maps. Elle est maintenant utilisée par d'autres applications qui ne sont pas nécessairement développées par Google. Une extension pour les données temps réel existe, nommée GTFS-RT.
STM	Société de Transport de Montréal. Le projet utilisera les données GTFS de ce réseau durant le développement pour tester le logiciel.
PFE	Projet de fin d'études
API	Application Programming Interface / Interface de programmation. Définit une méthode de communication avec un système. Sert souvent de façade pour des utilisateurs externes au système.
REST	Representational State Transfer. Style architectural pour des services

	Web, tel que des API.
TransitCrunch	Nom du système détaillé par ce présent document.

## 1.6 Références

Nom des références	Site des références
Spécification GTFS temps réel	<a href="https://developers.google.com/transit/gtfs-realtime/">https://developers.google.com/transit/gtfs-realtime/</a>
Spécification GTFS statique	<a href="https://developers.google.com/transit/gtfs/">https://developers.google.com/transit/gtfs/</a>
Portail développeurs de la STM	<a href="https://developpeurs.stm.info/fr/">https://developpeurs.stm.info/fr/</a>

## 2. Positionnement

### 2.1 Énoncé du problème

Le problème	<p>Les sociétés de transports publient des données ouvertes, qui sont sous-utilisées, sous-exploitées.</p> <p>Il est difficile pour le public de savoir si le service s'améliore ou se détériore.</p> <p>Les applications existantes présentent des estimés en temps réel qui ne tiennent pas compte de l'historique ni des conditions actuelles sur le réseau.</p> <p>est de ne pas savoir si un autobus est à l'heure, ou s'il y a des pannes sur le réseau</p>
Affecte	Les usagers d'un réseau de transport en commun et le public en général.
dont l'impact est	Une incertitude quant à la performance réelle du réseau de transport, notamment sa ponctualité.
Une bonne solution serait	<p>Archiver les données en temps réel et créer un tableau de bord pour rendre disponibles et utiles les données.</p> <p>Utiliser l'historique de données pour mesurer l'évolution de la performance du réseau et permettre de prédire la performance en temps réel.</p> <p>Ultimement avoir la position du prochain autobus ou une estimation de l'heure à laquelle il devrait arriver à un arrêt désigné.</p>

## 2.2 Positionnement du produit

Pour	Pour le public et en particulier les usagers d'un réseau de transport en commun
Dont le besoin est	<p>Obtenir des données sur la performance du service de transport.</p> <p>Les clients des réseaux de transport ne sont jamais certains si leur autobus (ou autre véhicule) est à l'heure, en avance, ou en retard. Les utilisateurs veulent mieux planifier leurs voyages en fonction de la fiabilité.</p>
TransitCrunch	Est un système permettant de voir la position des véhicules d'un réseau de transport, en temps réel. Il permet également d'accéder à des analyses, statistiques, et estimations basées sur les données historiques du réseau. Et par conséquent de prendre des décisions informées.
Qui	<p>Permet de :</p> <ul style="list-style-type: none"> <li>● voir en temps réel la position des véhicules sur le réseau;</li> <li>● voir une approximation de l'heure à laquelle le prochain véhicule arrivera;</li> <li>● obtenir un tableau de bord du trafic</li> </ul>
Contrairement à	<p>Le système actuel de plusieurs réseaux de transport ne présente que des données statiques (p. ex. horaires planifiés).</p> <p>Les outils existants qui sont soit propriétaires ou ciblés vers les développeurs.</p>
Notre produit logiciel	<ul style="list-style-type: none"> <li>● Doit fournir les données en temps réel</li> <li>● Doit fournir une interface conviviale</li> </ul>

### 3. Descriptions des intervenants et des utilisateurs

#### 3.1 Résumé des intervenants

Nom	Description	Responsabilités
Alain April	Professeur à l'ÉTS	Superviseur et responsable du projet
Francis De Vizio	Étudiant à l'ÉTS	Développeur sur le projet
Mathieu Poulin	Étudiant à l'ÉTS	Initiateur du projet, Chef d'équipe, Développeur sur le projet
Martin Satakem Mbougho	Étudiant à l'ÉTS	Développeur sur le projet
Simon Thériault	Étudiant à l'ÉTS	Développeur sur le projet

#### 3.2 Résumé des utilisateurs

Nom	Description	Responsabilités	Intervenant
Grand public / utilisateurs du transport en commun	Clients d'une agence de transport utilisant les véhicules (autobus, etc.) du réseau.	Utiliser l'application.	Partie prenante utilisant le logiciel.
Agences de transport	Agences de transport ayant ouvert des API GTFS temps-réel.	Maintenir leur API GTFS temps-réel.	TransitCrunch utilise les API de ces agences de transport.
Membres de l'équipe et Alain April	Équipe responsable du développement du projet.	Effectuer la conception et le développement du projet.	Parties prenantes développant le système et évaluateur du projet.

#### 3.3 Environnement utilisateur

- Les fonctionnalités seront accessibles via une application web. L'interface de cette application s'inspirera du Material Design de Google.

- L'application ciblera d'abord un environnement Desktop, mais dans la mesure du possible, un maximum de fonctionnalités seront aussi utilisables via des appareils mobiles.
- L'application sera multilingue (anglais et français).

### 3.4 Principaux besoins des intervenants et utilisateurs

Besoin	Priorité	Préoccupations	Solution actuelle	Solution proposée
B01 - Planifier un voyage en autobus	1	Ne pas rater son autobus!	Horaires statiques	Afficher la position du prochain autobus en temps réel sur une carte.
B02 - Consulter les horaires pour savoir quand le prochain autobus arrivera	2	- Les autobus ne sont pas toujours à l'heure. - Ils peuvent avoir une avance ou un retard.	Horaires statiques	Proposer une approximation de l'heure à laquelle le prochain autobus arrivera à partir des données temps réel.
B03 - Connaître l'état du service	2	Savoir si un autobus fait l'objet d'un accident, d'un retard, d'une annulation de trajet, etc.	API d'état du service (informations données très limitées) (pour certaines agences seulement).	À partir de la position temps réel et de l'approximation du temps d'arrivée, l'utilisateur peut savoir ce qui arrive à son autobus.
B04 - Archiver les données en temps réel	1	Accéder à l'historique des données	Aucune	Télécharger à intervalle fréquent les données à partir des API d'agences de transport et les entreposer dans une base de donnée.
B05 - Connaître des statistiques sur une agence ou une ligne	3	Savoir si une agence ou une ligne est fiable par rapport à une autre.	Aucune	Avec les données historiques, le système peut calculer des statistiques sur n'importe quel objet du système (véhicule, arrêt, ligne, agence, etc.)
B06 - Vue d'ensemble de la performance actuelle du réseau	1	Savoir si un événement majeur peut retarder ou	API d'état du service (informations	À l'aide d'une carte représentant les positions temps réel des véhicules

		annuler un trajet d'autobus.	données très limitées) (pour certaines agences seulement).	du réseau, l'utilisateur peut voir la performance complète du réseau: si le trafic ou un incident retarde le service, par exemple.
B07 - Vue d'ensemble de la performance historique du réseau	2	Voir si la performance du réseau se dégrade ou s'améliore avec le temps.	Aucune	Avec les données historiques GTFS de l'agence, le système est en mesure de fournir des statistiques sur l'efficacité d'une agence ou d'une ligne.
B08 - Visualiser l'évolution de l'offre de service	3	Calculer des statistiques et des événements importants représentant l'offre de service.	Aucune	À partir des données historiques, le système peut montrer l'ajout de nouveaux véhicules, et de nouvelles lignes au réseau.
B09 - Prédire les temps de parcours en temps réel	3	Avoir des prédictions fiables.	Les solutions actuelles ne tiennent pas compte de l'historique, ni de l'état actuel du réseau	Utiliser les données historiques de temps de parcours, les données météo, etc. pour faire des prédictions.
B10 - Avoir accès à une interface multilingue	3	Augmenter l'accessibilité de l'application.	Plusieurs solutions présentent des interfaces multilingues	Présenter une interface où les langues françaises et anglaises sont disponibles.

## 4. Vue d'ensemble du produit

### 4.1 Perspective du produit

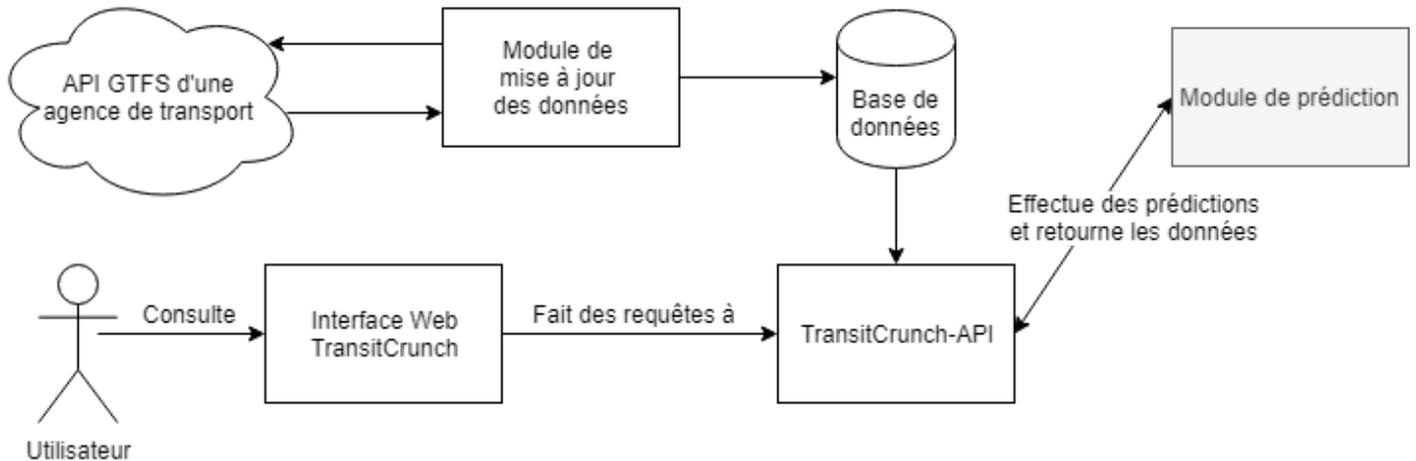


Figure - Vue d'ensemble de l'architecture du système

### 4.2 Principaux avantages

Besoins / Caractéristiques	Caractéristiques correspondantes
B01 - Planifier un voyage en autobus	CAR01, CAR05, CAR08
B02 - Consulter les horaires pour savoir quand le prochain autobus arrivera	CAR02, CAR05, CAR08
B03 - Connaître l'état du service	CAR01, CAR02
B04 - Archiver les données en temps réel	CAR04
B05 - Connaître des statistiques sur une agence ou une ligne	CAR03, CAR04
B06 - Vue d'ensemble de la performance actuelle du réseau	CAR04, CAR05, CAR08
B07 - Vue d'ensemble de la performance historique du réseau	CAR04, CAR06, CAR07
B08 - Visualiser l'évolution de l'offre de services	CAR03, CAR04, CAR07
B09 - Prédire les temps de parcours en temps réel	CAR03, CAR04, CAR08

### **4.3 Hypothèses et dépendances**

1. Le système ne consomme que des données ouvertes / publiques.
2. Le système est dépendant des API des agences de transport. Si une agence décide de fermer son API GTFS, les données ne seront plus disponibles dans le logiciel.

### **4.4 Licences et installation**

Code: Closed source. Le code pourrait être mis sous une license open-source à une date ultérieure.

Documents: Creative Commons (BY-SA)

## 5. Caractéristiques du produit

Les caractéristiques principales du produit sont décrites ci-dessous.

### 5.1 CAR01 – Afficher la position des véhicules d'un réseau de transport en temps réel sur une carte.

Une carte montre la position temps réel des véhicules du réseau de transport choisi par l'utilisateur. Celui-ci peut avoir plus d'information sur un véhicule en cliquant sur ce véhicule.

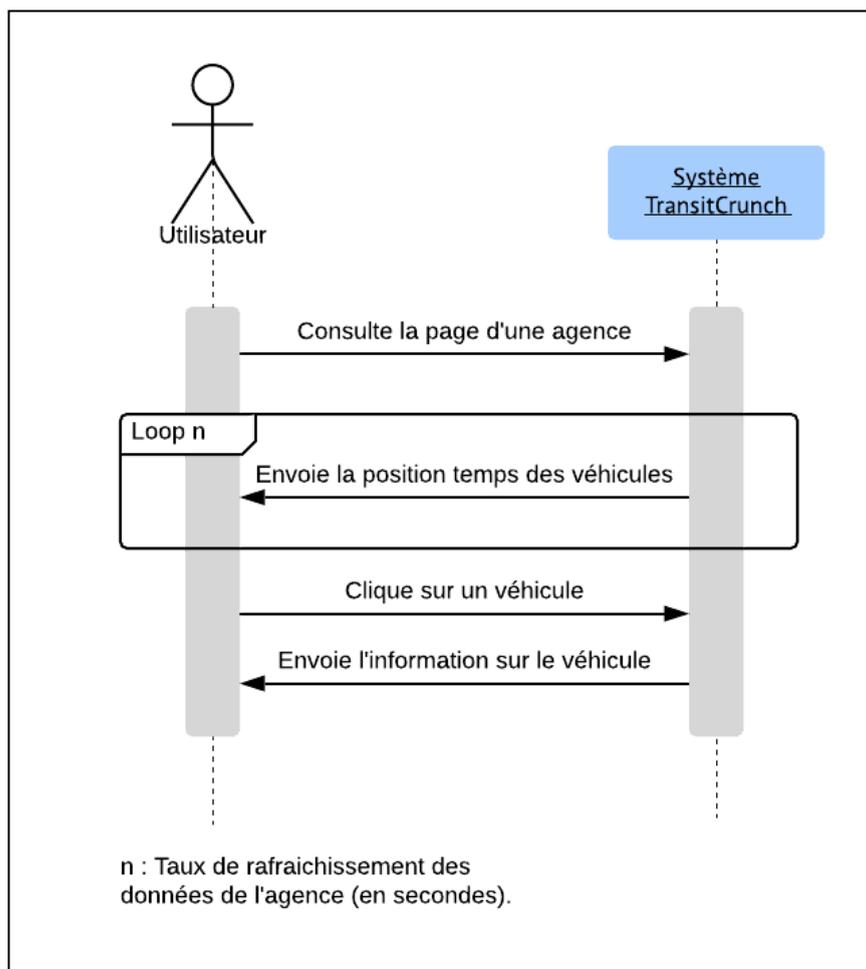


Figure - Diagramme de séquence pour CAR01: affichage de la position actuelle des véhicules d'un réseau

## 5.2 CAR02 – Visualiser l'évolution de l'offre de service

À partir des données historiques, le système peut montrer l'évolution de l'offre d'une agence: l'influence qu'a eu l'ajout de nouveaux véhicules ou de nouvelles lignes au réseau, par exemple. L'utilisateur sélectionne deux dates, et le système montre ce qui est arrivé sur le réseau de transport dans cet intervalle de temps sous forme de statistiques et d'événements importants. Certains événements et statistiques peuvent être précalculés. D'autres devront être calculés au moment de la requête.

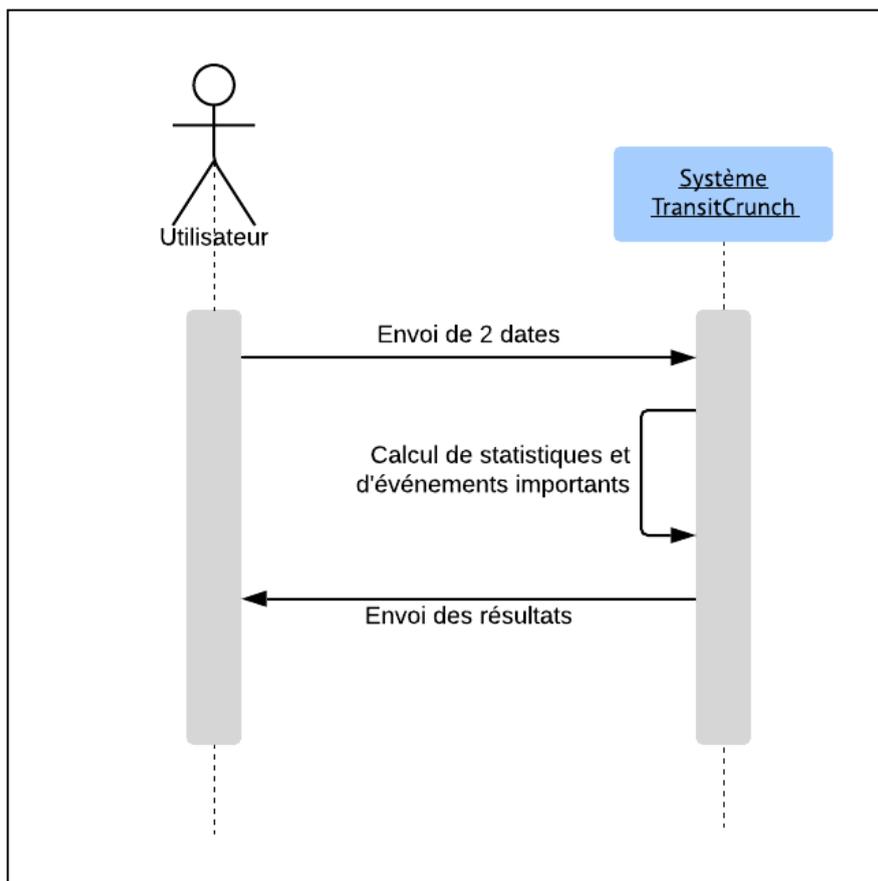


Figure - Diagramme de séquence pour CAR02: Visualisation de l'évolution de l'offre de service

### 5.3 CAR03 - Calculer des statistiques sur les objets du système

À partir des données historiques, le système peut calculer des statistiques pour chaque agence, ligne, arrêt, trajet et véhicule. Des exemples de statistiques possibles sont données à l'annexe F.

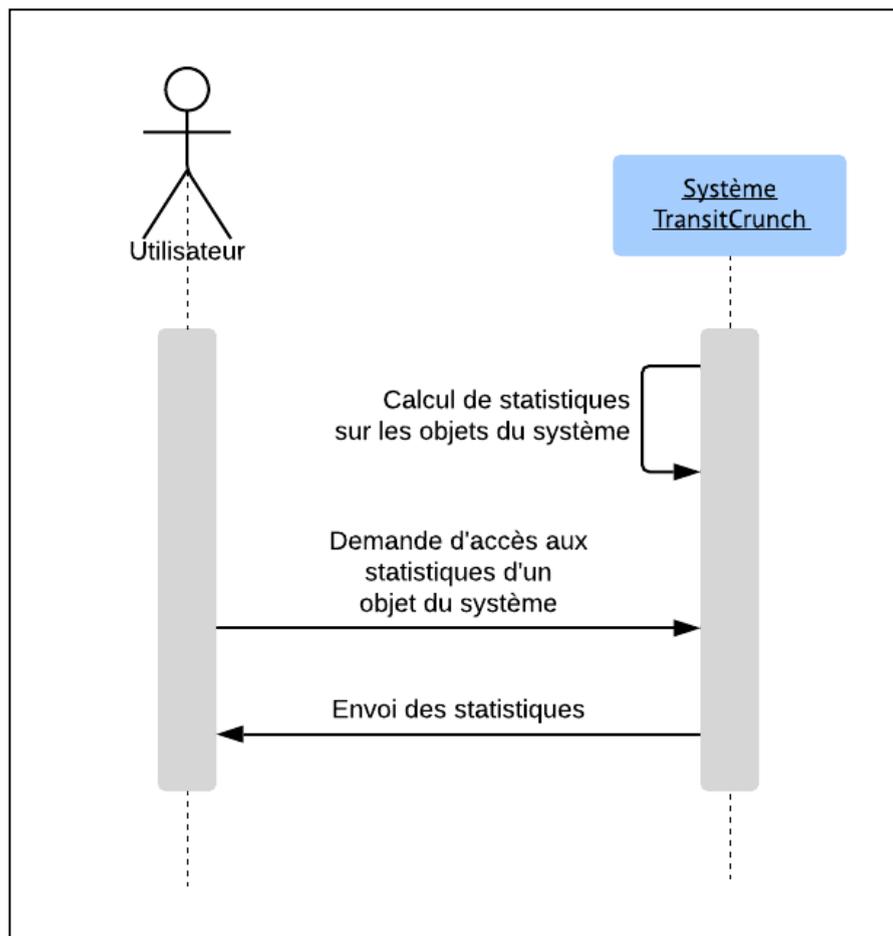


Figure - Diagramme de séquence pour CAR03 : calcul des statistiques sur les objets du système

## 5.4 CAR04 – Mise à jour et transformation des données

Extraction des données à partir de l'API GTFS d'une agence. Transformation des données dans un format plus utile pour le système. Entreposage dans une base de données. Ce processus est répété à un intervalle régulier qui dépend de l'API appelé.

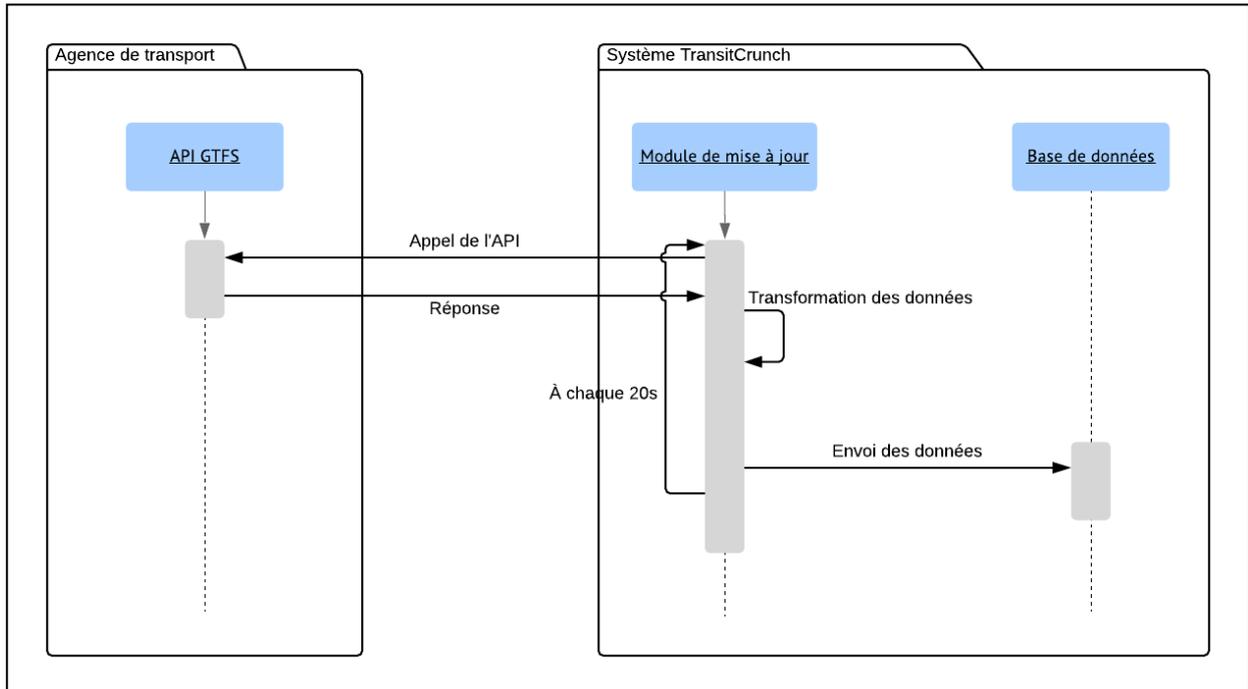


Figure - Diagramme de séquence pour la mise à jour et la transformation des données (CAR04)

## 5.5 CAR05 – Vue d'ensemble de la performance actuelle d'un réseau

Avec les données temps-réel GTFS, le système est en mesure de donner une vue d'ensemble sur la performance actuelle du réseau d'une agence de transport. La vue consiste de cartes, statistiques, et diagrammes montrant la performance. Il est donc possible de voir rapidement s'il y a un problème dans un secteur.

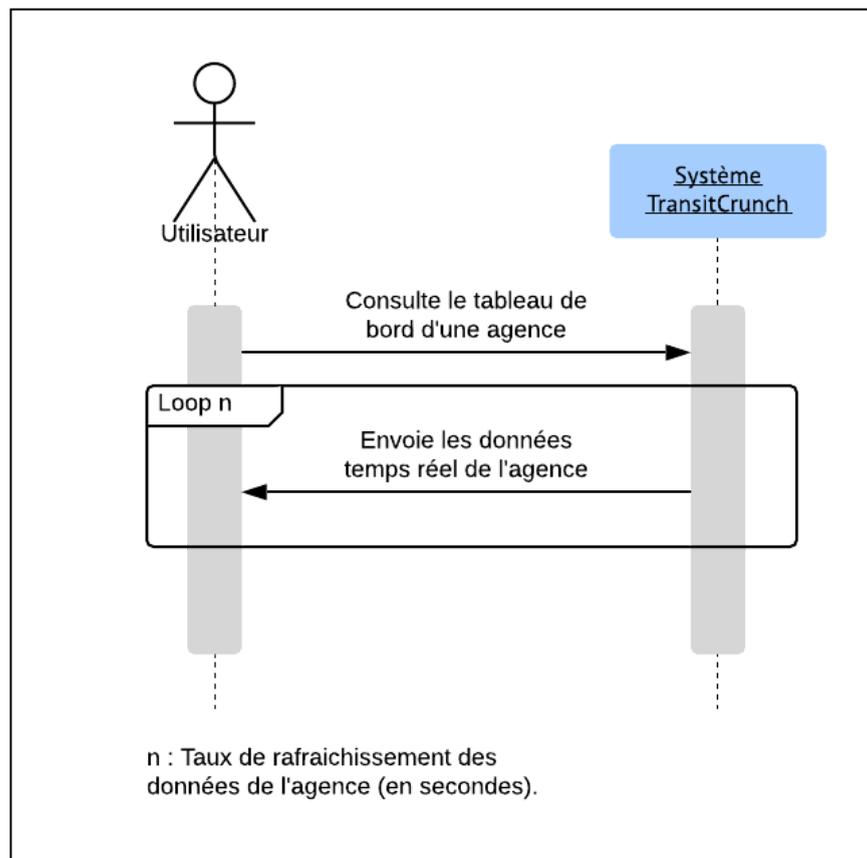


Figure - Diagramme de séquence pour la vue d'ensemble de la performance actuelle d'un réseau (CAR05)

## 5.6 CAR06 – Vue d’ensemble de la performance historique d’un réseau

Avec les données historiques GTFS, le système peut calculer des statistiques et produire des graphiques permettant de montrer une vue d’ensemble de la performance d’une agence à travers le temps.

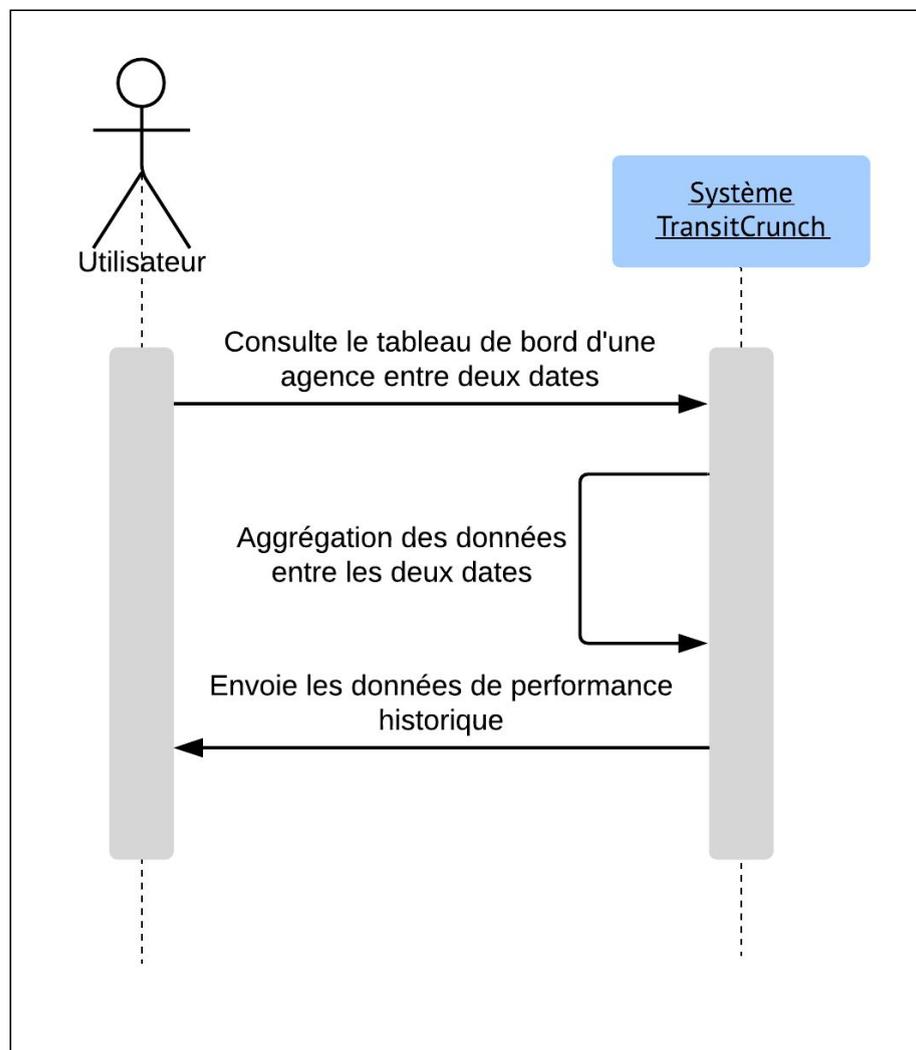


Figure - Diagramme de séquence pour CAR06 : Vue d’ensemble de la performance historique d’un réseau

## 5.7 CAR07 – Afficher une approximation de l’heure à laquelle le prochain véhicule arrivera à partir des données temps réel

Lorsqu’un utilisateur consulte les données d’un des arrêts d’une route, une prédiction de l’heure à laquelle le prochain véhicule arrivera lui est affichée. La prédiction est calculée à partir de la distance entre l’arrêt et le prochain véhicule, de l’historique de cet endroit particulier, et des conditions actuelles du réseau.

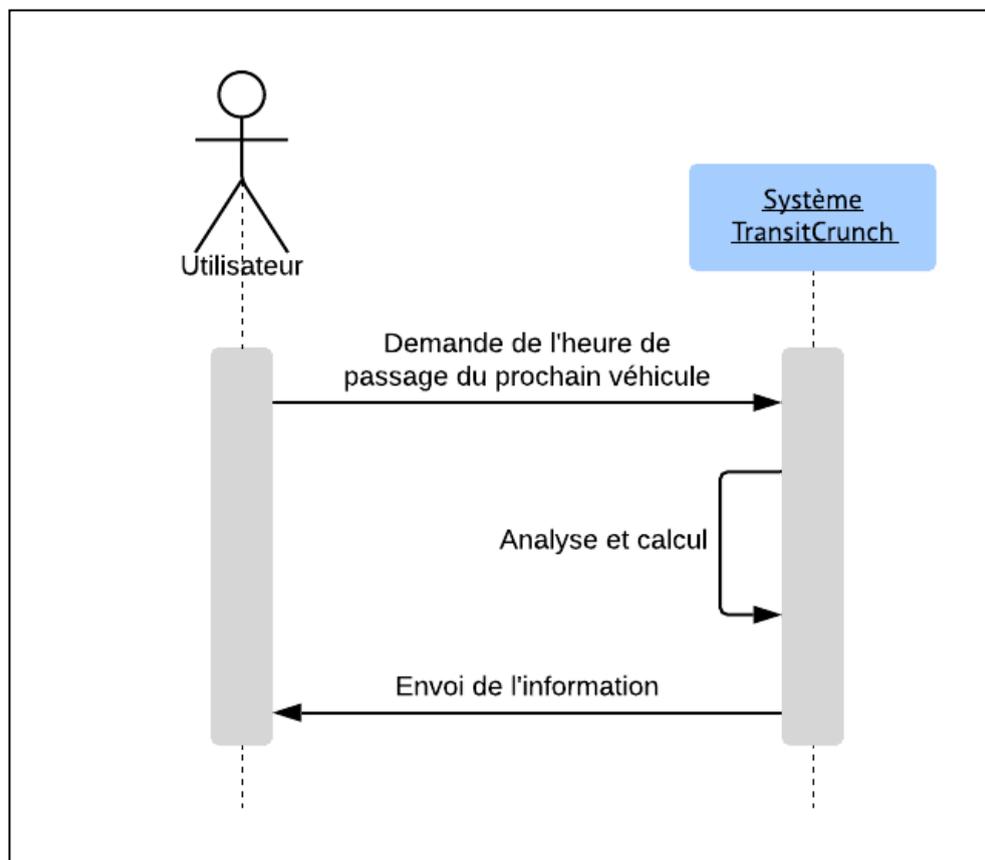


Figure - Diagramme de séquence pour CAR07 : Approximation de l’heure de passage du prochain véhicule

## 5.8 CAR08 – Prédiction des temps de parcours en temps réel

Avec les temps de parcours historiques et les données météorologiques (provenant d'une API externe), le système peut faire des prédictions sur les temps de parcours des véhicules en temps réel. Le système utilisera des algorithmes d'apprentissage machine pour estimer ces résultats. Ces prédictions s'appliquent spécifiquement aux routes et aux trajets, ce qui permet de modéliser les temps de parcours du réseau complet.

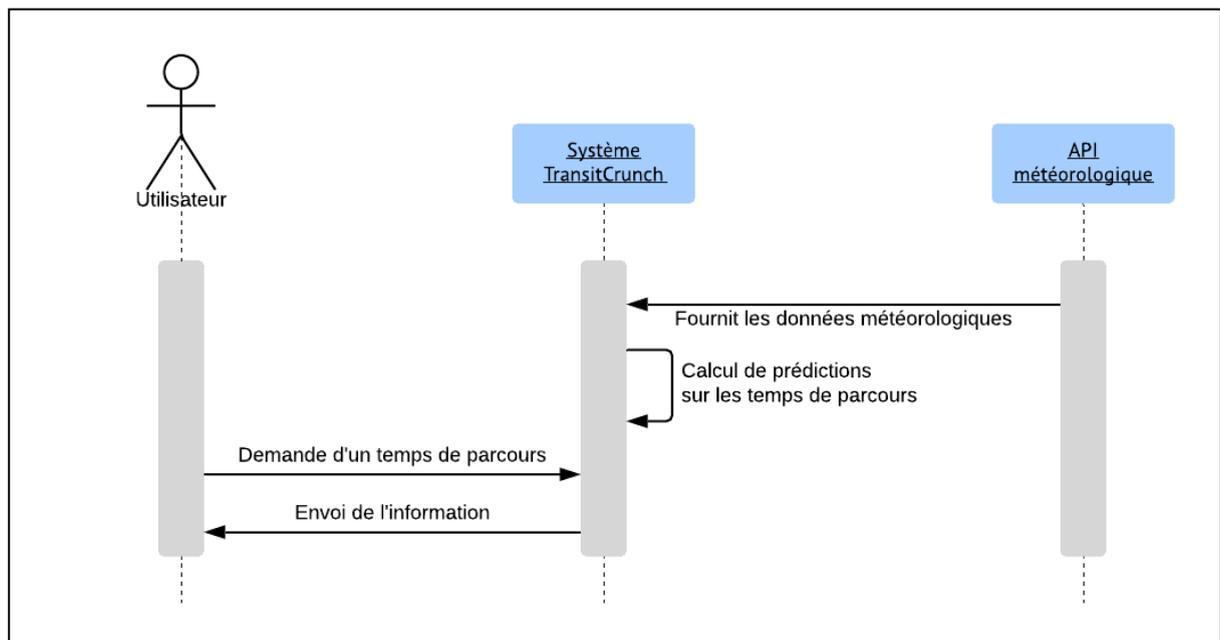


Figure : Diagramme de séquence pour CAR08 - Prédiction des temps de parcours en temps réel

## **6. Caractéristiques non fonctionnelles**

### **6.1 Exigences de performance**

- Le délai entre l'ingestion des données en temps réel et leur présentation doit être inférieur à l'intervalle de consommation de l'API de données ouvertes.

### **6.2 Exigences de sécurité**

- Afin de minimiser les coûts d'opération, des mécanismes doivent être envisagés pour éviter la surutilisation des services back-end par le public.

### **6.3 Exigences de disponibilité**

- Afin d'assurer la continuité des données historiques, l'ingestion des données GTFS temps réel doit se faire sans interruption.

## **7. Contraintes**

C01 - L'intervalle de rafraîchissement des données temps réel est dépendant des API des agences de transport. Ceci est très probablement différent pour chaque agence de transport.

C02 - Les réseaux de transport supportés par le système doivent donner un accès à un API GTFS temps réel.

C03 - L'interface web doit être disponible en français et en anglais.

## **8. Facteurs qualité pour l'environnement de production**

Comme les données accumulées sont d'une grande taille, l'environnement de production doit avoir accès à un espace de stockage important. Il a été calculé que les données temps réel accumulées au cours d'un an ont une taille d'un peu plus d'un téraoctet.

En raison de cette grande taille de données, un serveur et une base de données performantes sont requises pour l'environnement de production.

## 9. Attributs des caractéristiques

Caractéristiques	État	Bénéfice	Effort	Risque
CAR01 - Afficher la position des véhicules d'un réseau de transport en temps réel sur une carte	Complété	Élevé	Bas	Bas
CAR02 - Visualiser l'évolution de l'offre de service	Proposé	Élevé	Moyen	Bas
CAR03 - Calculer des statistiques sur les objets du système	Complété	Moyen	Moyen	Bas
CAR04 - Mise à jour et transformation des données	Complété	Élevé	Haut	Élevé
CAR05 - Vue d'ensemble de la performance actuelle d'un réseau	Proposé	Moyen	Bas	Moyen
CAR06 - Vue d'ensemble de la performance historique d'un réseau	Complété	Moyen	Moyen	Moyen
CAR07 - Afficher une approximation de l'heure à laquelle le prochain véhicule arrivera à partir des données temps réel	Annulé	Élevé	Haut	Élevé
CAR08 - Prédiction des temps de parcours en temps réel	Annulé	Élevé	Haut	Élevé

## 10. Échéancier récapitulatif du projet

<b>Itération</b>	<b>Livrable</b>	<b>Date de début estimée</b>	<b>Date de fin estimée</b>
1	Prototype à données statiques	2019/01/16	2019/02/14
2	Service d'acquisition des données	2019/02/12	2019/03/01
3	API TransitCrunch	2019/02/20	2019/03/15
4	Intégration des données en temps réel dans l'application web	2019/03/15	2019/03/30
5	Fonctionnalités avancées (prédiction)	2019/03/20	2019/04/11

# 11. Méthodologie de travail

## 11.1 Gestion

Une fois les technologies cibles identifiées, la première étape a été de créer un document d'outillage afin que les membres de l'équipe utilisent tous les mêmes outils tout au long du projet.

## 11.2 Outils

Les items suivants sont les principaux outils utilisés tout au long de l'élaboration du projet.

### Google Drive & Apps

Google Drive a été utilisé pour le partage de la documentation servant d'intrant au projet, la rédaction de la documentation propre au projet même, ainsi qu'à l'entreposage des données GTFS temps-réel téléchargées de la STM.

### Trello

La plate-forme Trello a été utilisée pour planification, l'assignation ainsi que le suivi des tâches au sein de l'équipe.

### Slack

La plate-forme de collaboration Slack a été utilisé pour la communication entre les membres de l'équipe, ainsi que pour recevoir des notifications lorsqu'une nouvelle version de l'application est déployée

### Chocolatey

Chocolatey est un gestionnaire de paquets pour Windows; il a été utilisé par l'équipe pour installer les autres applications de développement comme *nvm*, *yarn* et *docker*.

### Visual Studio Code

Visual Studio Code est un éditeur de code *open source* qui supporte une panoplie de langages et l'ajout de plug-ins. Il a été l'éditeur choisit pour coder les applications du projet.

## **Docker**

Docker est un outil de conteneurs logiciels permettant de démarrer des applications ou des services dans un état spécifique à l'aide de chaînes de commandes lors du démarrage du conteneur. Cet outil a été utilisé pour construire les bases de données de test.

## **Lucidchart**

Lucidchart est une application web offrant des services d'élaboration de graphiques. Les diagrammes de ce document ont été construits avec cet outil.

## **GitHub**

GitHub est un répertoire centralisé pour la gestion de versions avec le standard git. Deux répertoires git ont été utilisés au cours du projet. Un monorépertoire contenant les modules front-end et back-end tous deux écrits en TypeScript, ainsi qu'un deuxième répertoire pour le module de mise à jour des données écrit en Python.

## **11.3 Technologies**

Les items suivants sont les principales technologies qui ont été utilisées pour construire le projet.

### **Angular**

Angular est un cadre de développement d'applications web supportant TypeScript et offrant la séparation du code en modules.

### **NX**

Nx est un outil de gestion et de développement d'applications *full-stack*. Il permet d'organiser un projet complet dans un seul dépôt (*mono-repo*). Le projet a initialement été monté avec cet outil.

### **NestJS**

NestJS est un cadre de Node.js permettant le développement d'applications côté serveur suivant les pratiques d'Angular. Il permet la séparation du code en composants et support TypeScript.

## **TypeORM**

TypeORM est un ORM supportant TypeScript et compatible avec plusieurs cadres, dont Node.js et par conséquent, NestJS. Ce dernier a des extensions particulières permettant une intégration par module de TypeORM.

## **Material Design**

Material Design est un langage de conception faisant abstraction de styles et schémas permettant une interface composée de thèmes et dispositions.

## 12. Conclusion et recommandations

Les objectifs de ce projet étaient ambitieux. D'une part, il fallait composer avec un grand volume de données provenant de plusieurs sources différentes. Un autre défi était la gestion de l'équipe elle-même et l'apprentissage de nouveaux outils et technologies.

Le projet a permis à l'équipe de réaliser les fonctionnalités d'archivage sans problème. L'équipe a pu progresser sur l'avancement du concept du traitement des données temps réel. Un bon rythme de travail a été trouvé dans le dernier mois du projet.

Au niveau de la gestion du projet même, les tâches auraient gagné à être mieux définies et plus petites. En effet, plutôt que de concevoir l'API et le front-end en parallèle dans leur entièreté, il aurait été mieux d'établir un chemin critique pour s'assurer de livrer une première itération de l'API plus tôt. Cela nous aurait permis de mieux aligner les itérations futures sur ce qui pouvait être réalisé à l'intérieur de l'échéancier. Nous aurions également dû tenir des rencontres de travail plus souvent que celles seulement effectuées dans le dernier mois du projet.

En somme, le projet réalisé a permis d'élaborer différents composants d'un système de valorisation de données GTFS-realtime:

- un composant d'acquisition de données à partir de l'API d'une agence de transport
- un module de visualisation des données actuelles

Le prototype d'un système qui permet la visualisation de données GTFS historiques collectées à partir de l'API de la STM a pu être partiellement livré. Les fonctionnalités d'analyse de données demeurent cependant à l'état d'ébauche.

Finalement, le tout est une bonne base pour poursuivre le projet et explorer les améliorations et changements suivants:

- généralisation des données GTFS pour de multiples agences
- nettoyage des données téléchargées afin d'éliminer les doublons selon l'implémentation GTFS de chaque agence
- mise en cache des données back-end afin d'améliorer la performance
- un format de base de données NoSQL, qui se prête bien aux fichiers d'objets GTFS la technologie

- Elasticsearch pour les recherches dans les téraoctets hypothétiques des données GTFS historiques

# Annexes

## Annexe A - Schéma relationnel

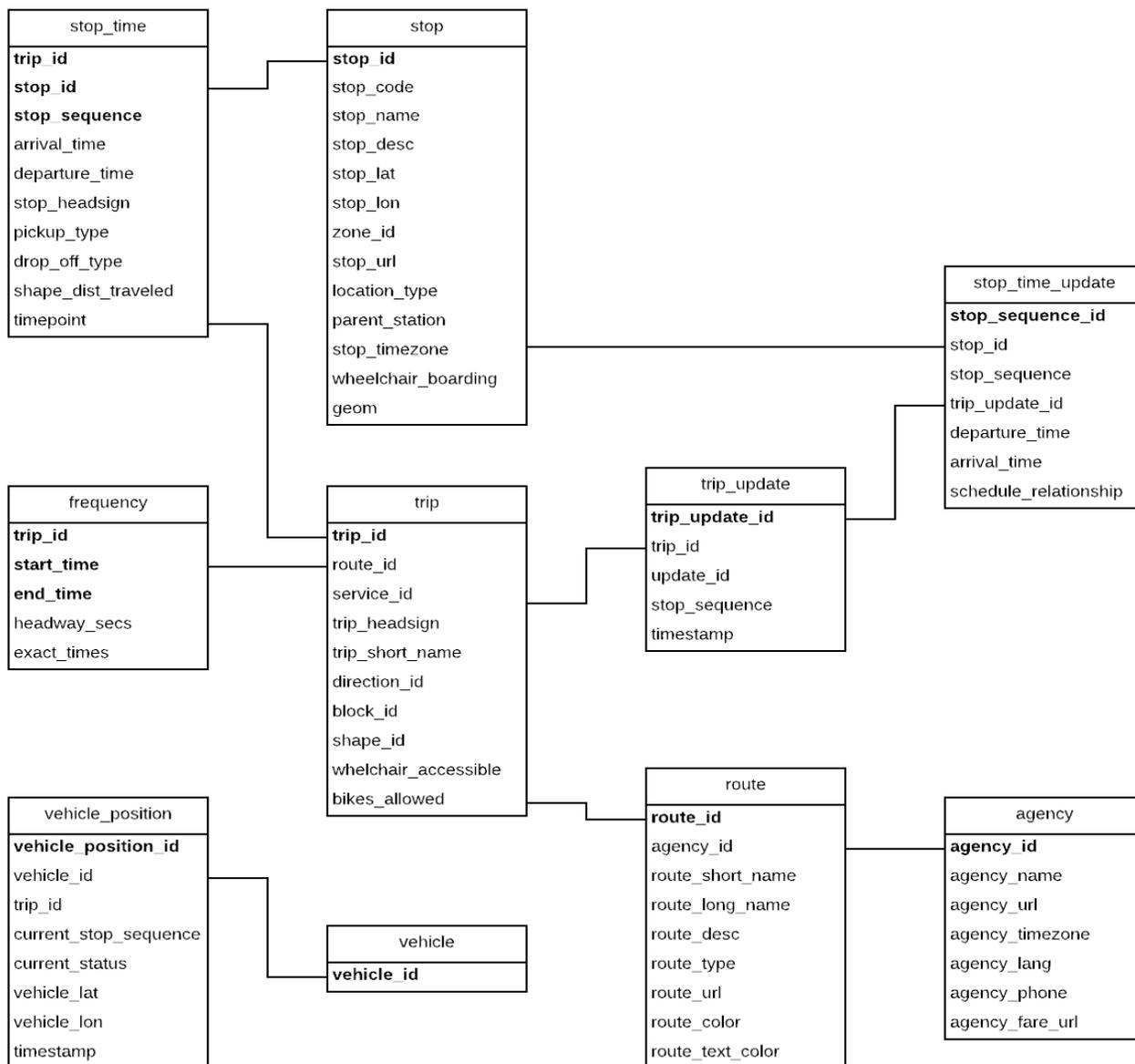


Figure : Schéma relationnel de la base de données

## Annexe B - Attributs des caractéristiques

Cette section présente la définition des attributs que nous avons associés aux différentes caractéristiques du prototype logiciel.

### État

Proposé	La caractéristique est proposée.
Approuvé	La caractéristique a été approuvée par les parties prenantes.
Complété	La caractéristique est complétée et incluse dans le système.
Annulé	La caractéristique a été annulée.

### Bénéfice

Faible	La caractéristique apporte un bénéfice faible au système. La valeur ajoutée n'est pas nécessaire au bon fonctionnement du système.
Moyen	La valeur ajoutée de la caractéristique n'est pas critique au bon fonctionnement du système.
Élevé	La valeur ajoutée de la caractéristique est très élevée ou essentielle au système.

### Effort

Bas	Ce niveau indique que la quantité d'effort requis est en dessous d'une semaine de travail.
Moyen	Ce niveau indique que la quantité d'effort requis est entre une et deux semaines de travail.
Haut	Ce niveau indique que la quantité d'effort requis est d'au moins deux semaines de travail.

### Risque

Faible	La technologie est bien connue, la méthode d'implémentation est peu complexe, ou la conception est simple à effectuer.
Moyen	La technologie est récente ou a peu de documentation, la méthode d'implémentation demande une attention particulière, ou la conception requiert une analyse assez complète.
Élevé	La technologie est peu connue, la méthode d'implémentation est complexe, ou la conception requiert une analyse très complète.

## Annexe C - Description des composants techniques du projet

Cette annexe présente une vue d'ensemble des quatre parties logicielles composant le système TransitCrunch. La base de données, l'API, l'application Web, et le module de mise à jour des données sont présentés. Tous les points de terminaison possibles de l'API y sont documentés.

### 1. Base de données

PostgreSQL, avec l'extension PostGIS

Le schéma de la partie statique est généré à l'aide de l'outil GTFS-Sequelize<sup>1</sup>

### 2. API REST

L'API est basée sur l'architecture REST. Elle ne supporte que les requêtes HTTP GET. Elle est donc en lecture seule.

#### Points de terminaison possibles

Pour chaque endpoint, des filtres pourraient être disponibles :

- Pagination
- Contraintes de date
- Données spécifiques aux points de terminaison

#### Réseaux de transport

```
GET /api/agencies
```

Offre la liste des agences présentement enregistrés dans la base de données.

```
GET /api/agencies/{agencyId}
```

Offre les informations spécifique à une agence particulière.

---

<sup>1</sup> <https://www.npmjs.com/package/gtfs-sequelize>

## Véhicules

```
GET /api/vehicles
```

Offre la liste des véhicules enregistrés dans le système, cette information n'inclut pas le positionnement des véhicules.

Supporte, entre autres, les filtres suivants:

- `agency={agencyId}`
  - Filtre pour une agence en particulier

```
GET /api/vehicles/{vehicleId}
```

Offre les informations spécifiques à un véhicule particulier.

## Positionnement des véhicules

```
GET /api/vehicle-positions
```

Offre des informations sur les véhicules en utilisation sur le réseau au moment de la requête.

Supporte, entre autres, les filtres suivants:

- `agency={agencyId}`
  - Filtre pour une agence en particulier
- `latest=(Date, format Epoch)`
  - Filtre pour les événements les plus récents depuis la date fournie. Date optionnelle.
- `tripId={tripId}`
  - Filtre pour un voyage spécifique
- `vehicleId={vehicleId}`
  - Filtre pour un véhicule spécifique.

```
GET /api/vehicle-positions/{vehiclePositionId}
```

Offre des informations sur un véhicule particulier en utilisation au moment de la requête.

## Lignes

Une ligne est un trajet planifié qui sera emprunté par plusieurs véhicules. C'est une série d'arrêts.

```
GET /api/routes
```

Offre la liste des lignes enregistrés dans le système.

Supporte, entre autres, les filtres suivants:

- `agency={agencyId}`
  - Filtre pour une agence en particulier

```
GET /api/routes/{routeId}
```

Offre de l'information sur une ligne en particulier.

## Arrêts

Un arrêt est un point géographique qui compose une ligne. Un véhicule s'arrête à ce point pour embarquer ou débarquer des passagers. À noter qu'un arrêt peut faire partie de plusieurs lignes à la fois.

```
GET /api/stops
```

Offre la liste des arrêts enregistrés dans le système.

Supporte, entre autres, les filtres suivants:

- `agency={agencyId}`
  - Filtre pour une agence en particulier
- `stopLong={stopLongitude}`
  - Filtre pour une longitude (approximative) particulière
- `stopLat={stopLatitude}`
  - Filtre pour une latitude (approximative) particulière
- `stopId={stopId}`
  - Filtre pour un stop spécifique.

```
GET /api/stops/{stopId}
```

Offre un arrêt en particulier.

## Voyages

Un voyage est le trajet planifié d'un véhicule en particulier pour un moment (habituellement un aller-retour); c'est une série d'arrêts dans le temps.

```
GET /api/trips
```

Offre la liste des voyages enregistrés dans le système.

Supporte, entre autres, les filtres suivants:

- `agency={agencyId}`
  - Filtre pour une agence en particulier

- `routeld={routeld}`
  - Filtre pour une ligne en particulier.

```
GET /api/trips/{tripId}
```

Offre un voyage spécifique.

### Arrêts dans le temps (Stop-time)

Il s'agit d'arrêts dans le temps pour des voyages spécifiques.

```
GET /api/stop-times
```

Offre la liste des arrêts dans le temps enregistrés dans le système.

Supporte, entre autres, les filtres suivants:

- `agency={agencyId}`
  - Filtre pour une agence en particulier.
- `tripId={tripId}`
  - Filtre pour un voyage en particulier
- `stopId={stopId}`
  - Filtre pour un arrêt en particulier
- `stopSequence={stopSequence}`
  - Filtre pour le numéro d'arrêt dans un voyage (1 à X).

```
GET /api/stop-times/{stopTimeId}
```

Offre un arrêt dans le temps spécifique.

### Mises à jour de voyage (Trip update)

Il s'agit d'une mise-à-jour sur les voyages planifiés.

```
GET /api/trip-updates
```

Offre la liste des mises-à-jour de voyages reçues par le système.

Supporte, entre autres, les filtres suivants:

- `agency={agencyId}`
  - Filtre pour une agence en particulier
- `latest=(Date, format Epoch)`
  - Filtre pour les événements les plus récents depuis la date fournie. Date optionnelle.
- `tripId={tripId}`
  - Filtre pour un voyage spécifique
- `vehicleId={vehicleId}`

- Filtre pour un véhicule spécifique.
- feedTimestamp={Date, format Epoch}
  - Filtre pour un feed particulier des données temps-réel.

```
GET /api/trip-updates/{tripUpdateId}
```

Offre une mise-à-jour en particulier.

### Mises à jour d'arrêt dans le temps (Stop-time update)

Il s'agit d'une mise-à-jour d'un arrêt dans le temps. Elle contient un changement d'horaire associé à l'arrêt.

```
GET /api/stop-time-updates
```

Offre la liste des mises-à-jour.

Supporte, entres autre, les filtres suivants:

- agency={agencyId}
  - Filtre pour une agence en particulier
- latest=(Date en Epoch)
  - Filtre pour les événements les plus récents depuis la date fournie. Date optionnelle.
- stopId={stopId}
  - Filtre pour un arrêt en particulier
- stopSequence={stopSequence}
  - Filtre pour le numéro d'arrêt dans un voyage (1 à X).
- feedTimestamp={Date en Epoch}
  - Filtre pour un feed particulier des données temps-réel.

```
GET /api/stop-time-updates/{stopUpdateId}
```

Offre une mise à jour en particulier.

### 3. Application Web

Angular, Leaflet

Cette application web utilise l'API pour exposer les données temps réel, historiques et des statistiques sur le réseau choisi.

### 4. Module de mise à jour des données

Ce module fait une requête à l'API de la STM à un certain intervalle de temps pour actualiser la base de données.

## Annexe D - Statistiques/métriques

Tableau de bord Transitcrunch:

- Grande carte avec:
  - les arrêts
  - les lignes
- Statistiques de base
  - nombre de lignes
  - nombre d'arrêts

Véhicules (vehicles)

- trajets effectués
  - routes effectuées
  - distance parcourue
  - temps de service
  - heatmap
- date d'entrée en service
- disparitions (réparation/retraite)
  - % activité (fiabilité?)
- records
  - vitesse maximale
  - vitesse moyenne
  - heures de service en 1 jour
  - plus grand "stop"

Arrêts (Stops)

- Lignes qui servent l'arrêt
- Catégorie (semaine seulement, nuit seulement?)
- Fréquence de passage
- Fiabilité
- % d'annulation
- % des bus qui y arrêtent

Lignes (Routes)

- Longueur (km)
- temps moyen
- Records
  - vitesse maximale
  - temps maximal
  - temps minimal
  - retard maximal

- Fiabilité
- Variantes
- Arrêts annulés
- Trajets annulés / interrompus
- Fréquence
  - # de trajets par heure/ jour

#### Trajets (trips)

- retard au départ
- retard à l'arrivée
- vitesse maximale
- temps immobilisé

## Annexe E - URLs de l'application web (front-end)

app/agency/network

app/agency/route/trip

app/agency/network

app/agency/buses

----

### **example.com/stm**

- example.com/stm/network
- example.com/stm/routes
- example.com/stm/stops
- example.com/stm/trips
- example.com/stm/vehicles

### Network

example.com/stm/dashboard

example.com/stm/now

example.com/stm/daily

- example.com/stm/daily/2018-12-31

example.com/stm/weekly

example.com/stm/monthly

example.com/stm/dtd

example.com/stm/wtd

example.com/stm/mtd

example.com/stm/ytd

### Routes

example.com/stm/routes

example.com/stm/routes/107

example.com/stm/routes/107/trips

example.com/stm/routes/107/schedule

example.com/stm/routes/107/vehicles

example.com/stm/routes/107/stops

### Vehicles

- example.com/stm/vehicles
- example.com/stm/buses

- [example.com/stm/trains](http://example.com/stm/trains)

[example.com/stm/trips](http://example.com/stm/trips)

[example.com/stm/trips/190328802](http://example.com/stm/trips/190328802)

[example.com/stm/trips/190328802](http://example.com/stm/trips/190328802)

[example.com/stm/vehicles](http://example.com/stm/vehicles)

[example.com/stm/vehicles/28004](http://example.com/stm/vehicles/28004)

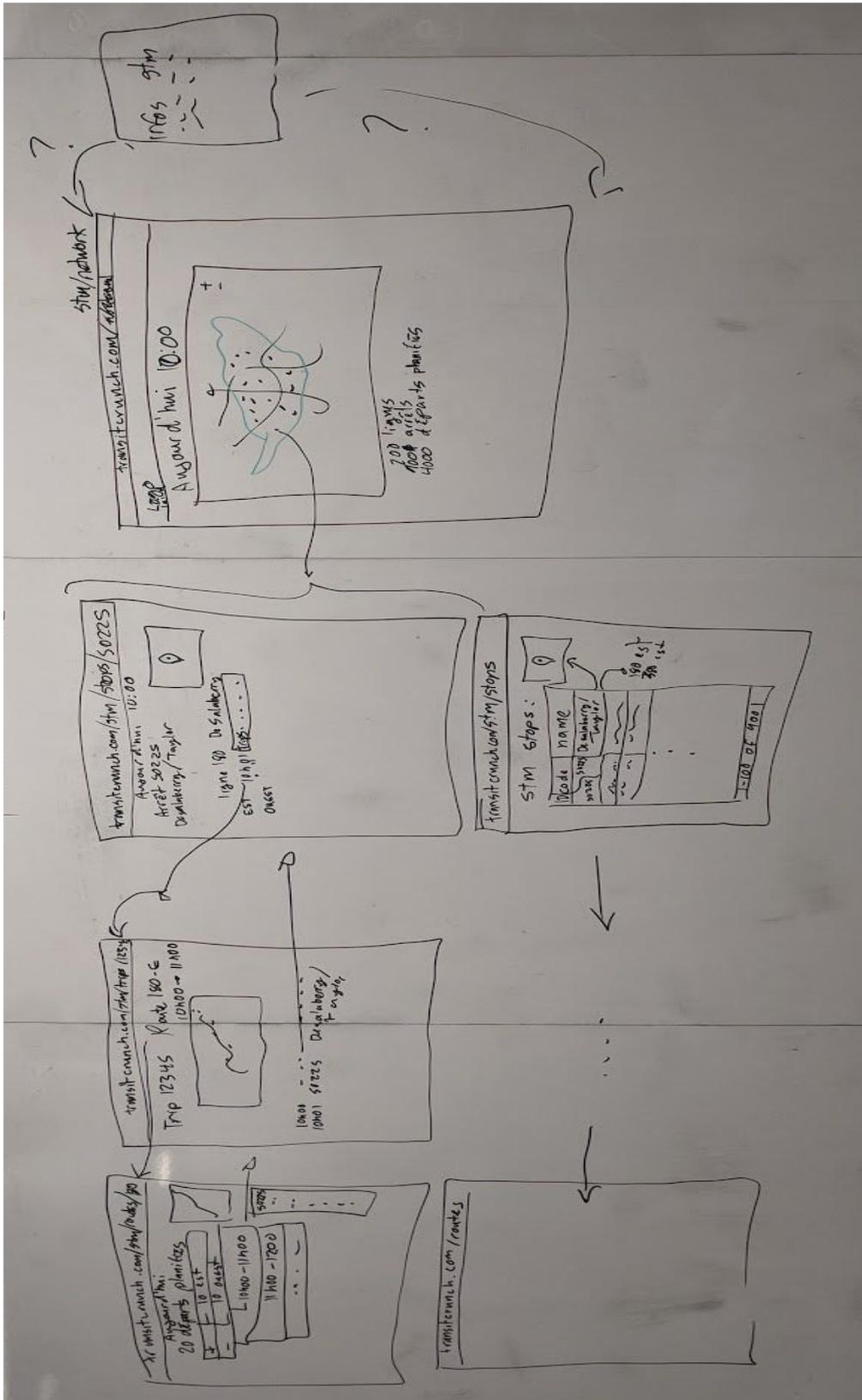
[example.com/stm/vehicles/28004/trips](http://example.com/stm/vehicles/28004/trips)

[example.com/stm/vehicles/28004/routes](http://example.com/stm/vehicles/28004/routes)

## Annexe F - Cycle de vie GTFS

- **GTFS statique**
  - **Fréquences de mises à jour**
    - Comment vérifier si et quand est mis à jour?
  - **Changement des données à travers le temps**
    - Quelles données changent?
    - Quelles données restent les mêmes?
  - **Comment gérer les changements**
    - Changements rétroactifs?
    - ajout/retrait de services?
  - **Autres particularités?**
  
- **flux GTFS-realtime**
  - **Fréquences de mises-à-jour**
    - Limitations de l'API?
      - Intervalle de mise à jour de données publiées?
      - Intervalle d'accès maximum?
    - Donc quel est l'intervalle que nous visons pour notre système?
  
  - **Changement des données à travers le temps**
    - Quelles données changent?
    - Quelles données restent les mêmes?
  - **Autres particularités?**

# Annexe G - Maquette de l'interface



## Annexe H - Architecture modulaire de l'API

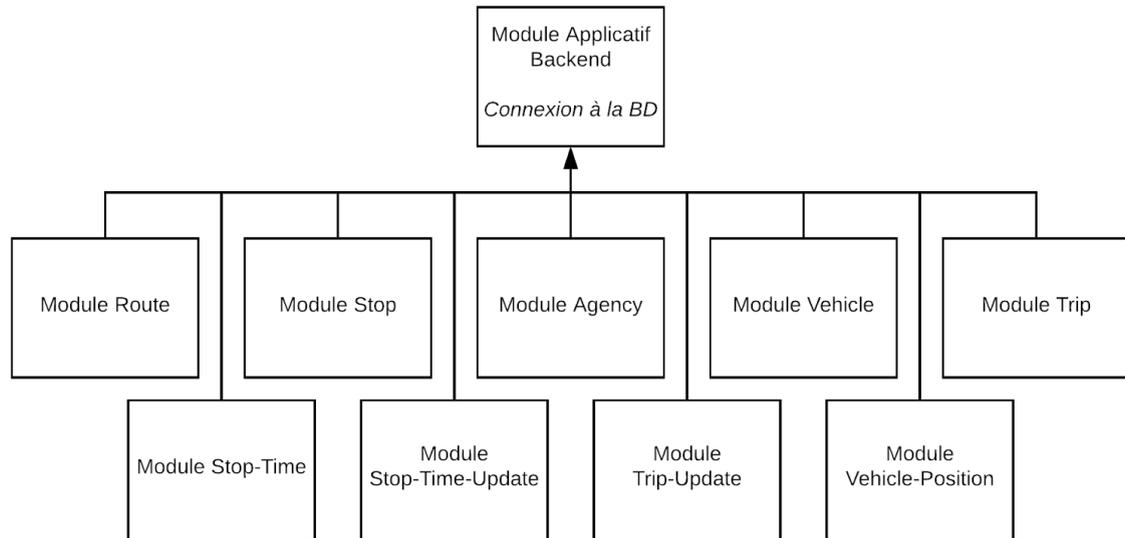


Figure - Schéma des modules de l'API

## Annexe I - Architecture d'un module de l'API

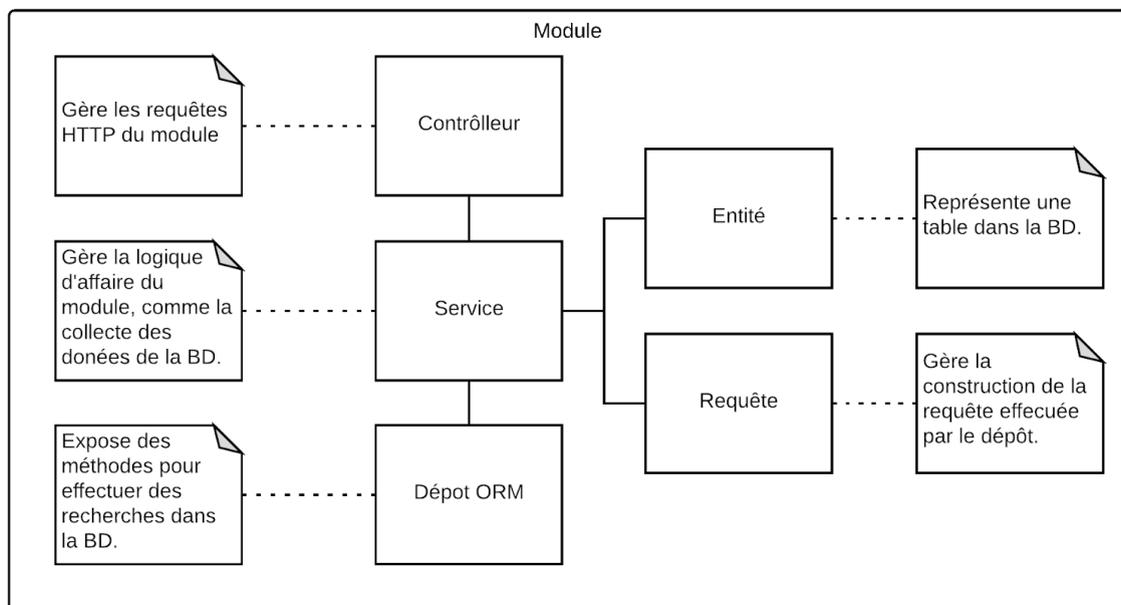


Figure - Schéma d'un module de l'API