

Développement et architecture d'une application web pour localiser les entreprises près de vous selon le mode de transport : le géolocalisateur d'entreprises de Lanaudière

Development and architecture of a web application to locate businesses near you by mode of transportation: The géolocalisateur d'entreprises de lanaudière

Philippe Apparicio, David Maignan and Jérémy Gelb

Volume 45, Number 1, 2022

URI: <https://id.erudit.org/iderudit/1088298ar>

DOI: <https://doi.org/10.7202/1088298ar>

[See table of contents](#)

Publisher(s)

Canadian Regional Science Association / Association canadienne des sciences régionales

ISSN

0705-4580 (print)

1925-2218 (digital)

[Explore this journal](#)

Cite this article

Apparicio, P., Maignan, D. & Gelb, J. (2022). Développement et architecture d'une application web pour localiser les entreprises près de vous selon le mode de transport : le géolocalisateur d'entreprises de Lanaudière. *Canadian Journal of Regional Science / Revue canadienne des sciences régionales*, 45(1), 5–15.
<https://doi.org/10.7202/1088298ar>

Article abstract

Over the past few years, Quebec, as elsewhere, has seen a proliferation of web-based mapping applications produced by various research teams. Designed as knowledge transfer tools, these customized applications are often based on free, opensource solutions. Such an approach to deploy map related applications offers great flexibility both in terms of application design and the implementation of cartographic, data mining and navigation functions. The aim of this article is primarily methodological. It aims to describe the deployment of a new web application, the Géolocalisateur d'entreprises de Lanaudière. To this end, the spatial data used, the innovative microservices-based architecture and the application's functionalities are extensively discussed.

DÉVELOPPEMENT ET ARCHITECTURE D'UNE APPLICATION WEB POUR LOCALISER LES ENTREPRISES PRÈS DE VOUS SELON LE MODE DE TRANSPORT : LE GÉOLOCALISATEUR D'ENTREPRISES DE LANAUDIÈRE

Philippe Apparicio, David Maignan, Jérémy Gelb

Philippe Apparicio

Laboratoire d'équité environnementale
Institut national
de la recherche scientifique,
Centre Urbanisation Culture et société
385, rue Sherbrooke Est, Montréal, QC
H2X 1E3
philippe.apparicio@ucs.inrs.ca

David Maignan

Laboratoire d'équité environnementale
Institut national
de la recherche scientifique,
Centre Urbanisation Culture et société
385, rue Sherbrooke Est, Montréal, QC
H2X 1E3

Jérémy Gelb

Laboratoire d'équité environnementale
Institut national
de la recherche scientifique,
Centre Urbanisation Culture et société
385, rue Sherbrooke Est, Montréal, QC
H2X 1E3

Soumis : 2021-05-28

Accepté : 2021-10-01

Résumé : Depuis plusieurs années, on assiste au Québec comme ailleurs à une multiplication des applications cartographiques sur Internet, produites par plusieurs équipes de recherche. Conçues comme des outils de transfert des connaissances, ces applications sur mesure sont souvent basées sur des solutions libres et gratuites. Une telle approche de déploiement d'application cartographique offre une grande flexibilité tant du point de vue du design de l'application que de l'implémentation de fonctions de cartographie, d'exploration de données et navigation. L'objectif de cet article est avant tout méthodologique. Il vise à décrire le déploiement d'une nouvelle application Web, soit *le Géolocalisateur d'entreprises de Lanaudière*. Pour ce faire, les données spatiales utilisées, l'architecture innovante basée sur des microservices et les fonctionnalités de l'application sont largement discutées.

Mots clés : cartographie, application web, micro-service, OpenTripPlanner, accessibilité, économie régionale, emploi

Development and architecture of a web application to locate businesses near you by mode of transportation: the géolocalisateur d'entreprises de lanauidière

Abstract : Over the past few years, Quebec, as elsewhere, has seen a proliferation of web-based mapping applications produced by various research teams. Designed as knowledge transfer tools, these customized applications are often based on free, open-source solutions. Such an approach to deploy map related applications offers great flexibility both in terms of application design and the implementation of cartographic, data mining and navigation functions. The aim of this article is primarily methodological. It aims to describe the deployment of a new web application, the *Géolocalisateur d'entreprises de Lanaudière*. To this end, the spatial data used, the innovative microservices-based architecture and the application's functionalities are extensively discussed.

Keywords : cartography, web application, microservice, OpenTripPlanner, accessibility, regional economy, employment

INTRODUCTION

Depuis les travaux pionniers de Michael Peterson (1995, 2005, 2008a), la cartographie interactive sur Internet est en renouvellement constant du fait des avancées technologiques et nouvelles solutions cartographiques (Apparicio & Petkevitch, 2006; Kraak & Brown, 2003; Peterson, 2008b). Il existe aujourd'hui deux principales approches pour déployer une application cartographique ou un atlas sur Internet.

La première est basée sur des solutions cartographiques proposant des gabarits (*templates*) qui ne nécessitent aucune ou peu de lignes de code pour être déployée. Leur fonctionnement est principalement modulaire, c'est-à-dire que l'application cartographique est composée de plusieurs blocs accolés les uns aux autres. Les solutions les plus connues et utilisées actuellement sont probablement celles proposées par Esri (ArcGIS Dashboard et ArcGIS Web AppBuilder). Ce type de solution et d'approche présente trois avantages majeurs : l'accès à des fonctionnalités de cartographie interactive et d'exploration des données préconçues, un hébergement des données spatiales grandement facilité (soit sur un serveur de données comme avec ArcGIS Server, soit avec une solution nuagique comme avec ArcGIS Online), la possibilité de publier des cartes directement avec un logiciel de SIG bureautique (comme avec ArcGIS Pro), le tout ne nécessitant pour ainsi dire aucune compétence en programmation. Bien que ces solutions permettent un déploiement rapide d'applications cartographiques, elles présentent plusieurs inconvénients notables. Premièrement, ces solutions sont pour la plupart payantes : le plus souvent des licences annuelles auxquelles s'ajoutent des coûts de maintenance et des mises à jour des serveurs cartographiques, et ce, même si l'application cartographique n'évolue pas dans le temps. Deuxièmement, puisqu'elles reposent sur des gabarits, il en résulte une forme de standardisation de la production cartographique sur Internet ou autrement dit, une certaine forme d'uniformisation des applications cartographiques. Par exemple, à la première utilisation de l'application, des usager·ère·s aguerri·e·s — cartographes, géographes ou géomaticien·ne·s — reconnaîtra rapidement qu'elle est déployée avec une solution d'Esri ou MapBox par exemple. Troisièmement, étant standardisées, elles offrent peu de flexibilité relativement au design de l'application et aux fonctionnalités de cartographie ou d'analyse spatiale.

La deuxième approche consiste à créer une application cartographique sur mesure qui repose le plus souvent sur des solutions libres et gratuites tant du côté serveur (par exemple Mapserver, GeoServer, QgisServer, Geodjango, Geomajas) que du côté client (OpenLayers, Leaflet, Turf, etc). Il est aussi possible d'intégrer, via leurs API, des services externes, souvent payants, de géocodage, d'itinéraire, de fonds cartographiques (entre autres MapBox, Walk Score, Azure Maps, Google Maps), etc. L'avantage principal de cette approche est qu'elle offre une très grande flexibilité tant du point de vue des fonctionnalités que du design de l'application; autrement dit, l'application a une identité qui lui est propre. Par contre, elle nécessite un développement Web conséquent tant du côté serveur que du côté client, reposant sur de solides compétences en programmation. Dépendamment de la complexité de l'application, son déploiement peut être ainsi très chronophage, mais aussi coûteux. Il est à noter que ces deux approches ne sont pas nécessairement exclusives. À titre d'exemple, l'API ArcGIS pour JavaScript permet de combiner à la fois les avantages d'une installation serveur standardisée et une grande flexibilité côté client.

Étant largement considéré comme un outil de transferts des connaissances, il n'est pas étonnant qu'on assiste depuis plusieurs années à une multiplication des applications cartographiques produites par des équipes de recherche. Le Québec n'échappe pas à cette tendance avec de nombreuses applications ancrées dans des champs

et thèmes et de recherche variés. Citons entre autres les atlas du Réseau de recherche et de connaissances relatives aux peuples autochtones (DIALOG) (Lévesque et al., 2019a, 2019b), un atlas pour comparer l'exposition individuelle aux pollutions atmosphérique et sonore selon le mode de transport (Apparicio et al., 2018), une application sur l'environnement et les aléas climatiques (Barrette et al., 2020), des atlas numériques sur le marché du travail (Apparicio et al., 2014; Apparicio et al., 2019), ou encore l'application Mobiliscope permettant d'explorer la mobilité et l'évolution de la ségrégation sociale durant la journée dans 49 villes françaises, mais aussi six au Québec (Vallée et al., 2020).

L'article est avant tout guidé par un objectif méthodologique et s'inscrit ainsi dans la lignée de travaux récents proposant de nouvelles approches pour déployer des applications cartographiques sur Internet (entre autres Apparicio et al., 2019; Apparicio & Petkevitch, 2006; Costantino et al., 2020; Price et al., 2021). Plus spécifiquement, il s'agira de décrire l'application cartographique du *Géolocalisateur d'entreprises de Lanaudière* (Apparicio et al., 2021). L'originalité de l'application est d'intégrer des mesures d'accessibilité — plus spécifiquement des isochrones — basées sur quatre modes de transport différents. À notre connaissance, seule l'étude récente de Price et al. (2021) propose une solution web intégrant des mesures d'accessibilité calculées pour plusieurs modes de transport.

Dans un premier temps, nous décrirons l'objectif ayant guidé le développement de l'application et les données spatiales nécessaires à son fonctionnement. Dans un second temps, le déploiement de l'atlas — son architecture basée sur plusieurs microservices, les librairies et langages utilisés — sera décrit en détail. Dans une troisième section, nous présenterons les différentes fonctionnalités de cartographie et d'exploration des données implémentées, ainsi qu'un mode d'emploi de l'application. Finalement, nous discuterons des limites de l'application à la fois en termes méthodologiques et d'utilisation et identifierons des pistes d'amélioration ou d'extension de l'application.

OBJECTIF DE L'APPLICATION ET DONNÉES SPATIALES UTILISÉES

Bref retour sur l'accessibilité aux opportunités d'emploi

Depuis les travaux pionniers de Kain (1968; 1992) sur l'hypothèse du mauvais appariement spatial, plusieurs études ont tenté de mesurer les opportunités spatiales d'emploi (Apparicio et al., 2014, p. 19). On assiste alors à un élargissement de la notion initial du *spatial mismatch* avec les notions de *skill mismatch* (Houston, 2005; Manacorda & Petrongolo, 1999; Pastor & Marcelli, 2000), *modal mismatch* (Grengs, 2010; Ong & Miller, 2005; Patacchini & Zenou, 2005; Sanchez, 1999; Taylor & Ong, 1995) et plus récemment de *time mismatch* (Boisjoly & El-Geneidy, 2016). Concrètement, cela signifie que pour une personne résidant à une localisation donnée, ses opportunités spatiales d'emploi dans son environnement immédiat vont varier en fonction de trois critères :

- des types d'emploi offerts, soit l'adéquation entre ses compétences et celles requises pour les emplois, renvoyant ainsi à la notion d'employabilité. Il convient alors d'évaluer les opportunités en fonction des secteurs d'activité et/ou des professions.
- des modes de transport puisque l'accessibilité aux emplois — soit le nombre d'emplois disponibles à n minutes — va varier significativement en fonction du mode de transport utilisé (marche, vélo, automobile, transport en commun).
- du moment de la journée, particulièrement pour des déplacements en transport en commun.

Objectif du géolocalisateur d'entreprises de Lanaudière et utilisateurs potentiels

L'objectif principal du *Géolocalisateur d'entreprises de Lanaudière* est de fournir une liste d'entreprises présentes dans différentes zones de desserte délimitées autour d'une localisation géographique et à partir d'un mode de transport : à pied (10, 15, 20, 25, 30 minutes), à vélo (10, 15, 20, 30, 45 minutes), en automobile (10, 15, 20, 30, 45 minutes) et en transport en commun (15, 30, 45, 60, 75 minutes). Pour ce faire, l'utilisateur·trice de l'atlas doit au préalable sélectionner : 1) une adresse ou un code postal, 2) un à trois secteurs d'activité¹ ou une seule profession².

Par conséquent, l'atlas est destiné à deux types d'utilisateur·trice·s. D'une part, il constitue un outil intéressant pour soutenir la recherche d'emploi. En effet, les personnes en recherche d'un emploi peuvent l'utiliser pour repérer les entreprises localisées à proximité de chez eux, et ce, dans un secteur d'activité donné ou reliées à une profession donnée; et donc celles potentiellement susceptibles d'embaucher du personnel dans les secteurs ou professions sélectionnées. D'autre part, l'application peut être utilisée par diverses organisations ou entreprises de Lanaudière souhaitant approfondir leurs connaissances sur la distribution spatiale des activités économiques dans la région.

Les données spatiales nécessaires au fonctionnement de l'atlas

L'atlas repose sur l'utilisation de plusieurs bases de données spatiales, relatives à trois dimensions : 1) les entreprises, les bureaux de Services Québec et les professions, 2) les adresses et codes postaux, et 3) les réseaux de transport. Concernant la première dimension, quatre bases de données nous ont été fournies par notre partenaire (Services Québec Lanaudière) :

- Une liste des entreprises d'au moins cinq employé·e·s localisées dans la région de Lanaudière (n = 5446) provenant du Répertoire des établissements d'Emploi-Québec (IMT en ligne) avec toute une série d'informations pertinentes : nom de l'entreprise, adresse, téléphone, site Web lorsque disponible, taille de l'entreprise³, code SCIAN à six chiffres, secteur d'activité et coordonnées géographiques.
- Une liste des sept bureaux de Services Québec (anciennement dénommés Centre local d'emploi) avec leur territoire de desserte.
- Une liste des 398 professions (CNP à quatre chiffres) comprenant plus de 50 personnes occupées dans la région de Lanaudière (issue du recensement de 2016 de Statistique Canada).
- Une matrice entre les professions (CNP-4) et les secteurs d'activité (SCIAN 4) où les cellules indiquent le nombre de personnes occupées (issue du recensement de 2016 de Statistique Canada). À partir de cette matrice, il est alors possible d'associer chaque profession (CNP-4) avec plusieurs secteurs d'activité (SCIAN 4). En accord avec notre partenaire, le critère d'association retenu est que la part des personnes occupées de la région de Lanaudière dans les secteurs d'activité représente au moins 2% du total des personnes occupées de chaque profession. Nous verrons qu'il est alors possible de sélectionner les entreprises de la région pour une profession donnée sur la base de ce critère.

Concernant la localisation des points d'origine à partir desquelles sont délimitées les différentes isochrones, nous avons utilisé deux sources de données : les produits d'adresse Québec AQ *géobâti* et AQ *géopostal*⁴. La première source comprend les localisations précises des adresses (points placés sur les bâtiments) tandis que la seconde comprend celles des codes postaux à six caractères, et ce, pour l'ensemble du Québec. À partir de ces deux sources, nous avons extrait les adresses (n = 227 044) et codes postaux (n = 10 683) situés sur le territoire de la région de Lanaudière. Il est à noter que nous aurions aussi pu utiliser un service de géocodage externe, comme l'API *Geocoding* de Google Maps. Cette option n'a pas été retenue pour deux raisons. Premièrement, ce service est payant ; les prix variant selon le nombre de requêtes effectuées chaque mois. Deuxièmement, cela aurait nécessité de délimiter les zones de desserte à la volée sur le serveur, ce qui aurait ralenti considérablement l'application Web.

Le calcul des isochrones selon les quatre modes de transport (marche, vélo, automobile, transport en commun) repose sur deux jeux de données : 1) les données géographiques de la cartographie collaborative libre OpenStreetMap (OSM) (OpenStreetMap Contributors, 2021) qui comprennent des informations sur les réseaux routiers et cyclables, 2) et les données sur les réseaux de transport en commun au format GTFS⁵. Les données OSM (fichier osm) ont été extraites avec le *package* *osmdata* (Padgham et al., 2017) dans R (R Core Team, 2021) (encadré 1). Il est à noter que cette opération peut être aussi facilement réalisée dans QGIS (QGIS.org, 2021) avec l'extension QuickOSM (Trimaille, 2021). Quant aux données GTFS des agences de transport du CRT Lanaudière et des MRC de L'Assomption et les Moulins, elles ont été extraites du site OpenMobilityData! (<https://transitfeeds.com/>). Une fois, ces données téléchargées (*osm* et *gtfs*), il est alors possible de créer un graphe avec l'application JAVA gratuite OpenTripPlanner⁶ qui permet de calculer des trajets et des isochrones selon le mode de transport⁷.

Encadré 1. Extraction des données d'OpenStreetMap dans R

```
# Chargement des paquets
library(sf) # pour manipuler des données spatiales
library(osmdata) # pour télécharger des données OpenStreetMap

# Importation de la couche géographique de la région de Lanaudière
# au format shapefile
Lanaudiere <- st_read("../data /RegLanaudiere.shp")
# Zone tampon de 20 km autour de la région
Lanaudiere20km <- st_buffer(Lanaudiere, dist=20000)
# Transformation dans la projection WSG84 (EPSG4326)
Lanaudiere20km <- st_transform(Lanaudiere20km, crs="+init=epsg:4326")
# Délimitation de l'emprise de la région
Emprise <- st_bbox(Lanaudiere20km)
# Extraction du fichier OSM
SpatialQuery <- opq(bbox = Emprise)
osmdata_xml(SpatialQuery, file="../Lanaudiere.osm")
```

¹ Les 41 secteurs d'activité ont été créés par Emploi Québec et constituent des regroupements de codes SCIAN (Système de classification des industries de l'Amérique du Nord) à deux, trois ou quatre chiffres. Pour une description détaillée de ces secteurs (regroupements), on pourra consulter la page suivante du site de l'IMT en ligne. La liste et définition de ces 41 secteurs est aussi disponible dans le glossaire du *Géolocalisateur d'entreprises de Lanaudière* (<https://www.geolocalisateur-lanaudiere.ucs.inrs.ca/glossaire>).

² Nous avons retenu les professions à quatre chiffres de la classification nationale des professions (CNP) comprenant au moins 50 personnes occupées dans la région de la Lanaudière.

³ Variable ordinale avec les catégories suivantes : 5 à 19 employé·e·s, 20 à 19, 50 à 99, 100 à 199, 200 à 499, 500 à 999 et 1000 et plus.

⁴ Les descriptions de ces deux produits sont disponibles aux adresses suivantes : <https://adressesquebec.gouv.qc.ca/aqgeobati.asp> et <https://adressesquebec.gouv.qc.ca/aqgeopostal.asp>

⁵ Succinctement, les fichiers GTFS (General Transit Feed Specification) constituent un format proposé par Google Maps permettant de décrire l'organisation d'un réseau de transport en commun (horaires et lignes de bus, métro, trains, localisation géographique des arrêts) à travers différents fichiers (*agency, calendar, routes, stops, stop_times, trips*, etc.). Ce format est aujourd'hui largement connu et utilisé. La plupart des agences de transport mettent en accès libre ces données qui sont mises à jour habituellement une à deux fois par année.

⁶ <https://www.opentripplanner.org/>

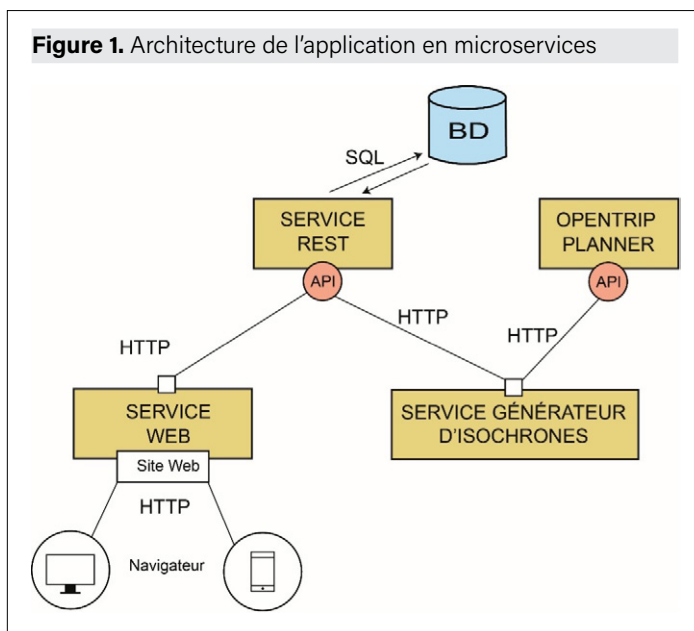
⁷ Notez qu'il est très facile d'utiliser l'application OpenTripPlanner à partir de Python et aussi R (Morgan, Young, Lovelace et Hama, 2019).

Une architecture du projet basée sur des microservices

L'application repose sur une architecture basée sur des microservices, popularisée par Martin Fowler (2014) et de plus en plus utilisée aujourd'hui, notamment par les compagnies Netflix et Amazon. Comme son nom l'indique, ce type d'architecture comprend une suite de services indépendants qui s'exécutent dans leurs propres processus et qui communiquent entre eux avec des mécanismes légers, souvent avec une API et le protocole HTTP. Nous avons choisi ce type d'architecture puisqu'elle présente plusieurs avantages notables pour le déploiement d'une application Web cartographique :

- Ce découplage offre plus de flexibilité concernant les choix technologiques. En effet, puisque les modules sont indépendants, ils peuvent être développés dans des langages différents (Java, Python, C, R, etc.). Autrement dit, chaque module peut être écrit dans le langage le plus adapté aux fonctionnalités qu'il doit remplir. En guise d'exemple, tel que décrit en détail ultérieurement, certains modules du projet sont développés en JavaScript, d'autres en Python et en JAVA. Il serait aussi possible d'ajouter un nouveau module avec R ou encore Matlab.
- Contrairement à une application monolithique, elle impose un fort découplage entre les modules du projet. Cela facilite l'organisation du projet puisque les modules peuvent être développés en parallèle par des personnes ou équipes avec des compétences différentes.
- La maintenance, la mise à jour et la mise à l'échelle d'un service sont facilitées puisqu'il nécessite de réviser que le service en lui-même, et non l'ensemble de l'application.

Pour ce projet, nous avons créé quatre microservices (figure 1). Le premier, le module central, est un service REST (*Representational State Transfer*) dont le rôle est simplement de fournir des données aux autres services (écrit en Python). Le deuxième service est l'application gratuite OpenTripPlanner à partir de laquelle sont calculées les isochrones (écrite en Java). Il alimente notamment le troisième service qui convertit les isochrones au format GeoJSON (écrit en JavaScript), soit un format ouvert de données spatiales. Le dernier service est le site web qui publie des pages HTML (écrit en JavaScript). Dans les sous-sections suivantes, chaque microservice est décrit en détail, avec notamment les fonctionnalités et les librairies utilisées.



Le microservice REST

Ce service, développé avec le langage Python et la librairie FlaskRestPlus (Haustant, 2019), joue le rôle d'interface avec la base de données : il envoie les requêtes SQL à la base de données et renvoie les résultats au format JSON aux autres microservices. Il est à noter que deux types de bases de données sont utilisées : 1) SQLite (Hipp, 2020; Owens & Allen, 2010) pour le développement et les tests unitaires, 2) PostgreSQL 10 (The PostgreSQL Global Development Group, 2021) pour le serveur en production. Cela explique que nous utilisons la librairie SQLAlchemy (Bayer, 2012) permettant d'écrire des requêtes SQL standardisées pour les deux types de bases de données. Le modèle conceptuel de la base de données est présenté à la figure 2, avec notamment les tables pour les adresses ($n = 227\,044$), les codes postaux ($n = 10\,683$), les entreprises ($n = 5446$), les professions ($n = 398$), les bureaux de Services Québec ($n = 7$), et les isochrones ($n = 856\,761$).

Le microservice Opentripplanner

Le processus Opentripplanner est un service JAVA qui lance un serveur Web. La communication se fait donc par le protocole http, c'est-à-dire l'envoi de requêtes GET paramétrées (encadré 2). Une fois le calcul de l'isochrone effectué, il renvoie le résultat au format GeoJson.

Encadré 2. Exemple de requête GET envoyé au microservice OpenTripPlanner en JavaScript

```

// Chargement de la librairie axios
const axios = require("axios");

// modes = ["WALK", "BICYCLE", "CAR", "WALK,TRANSIT"]

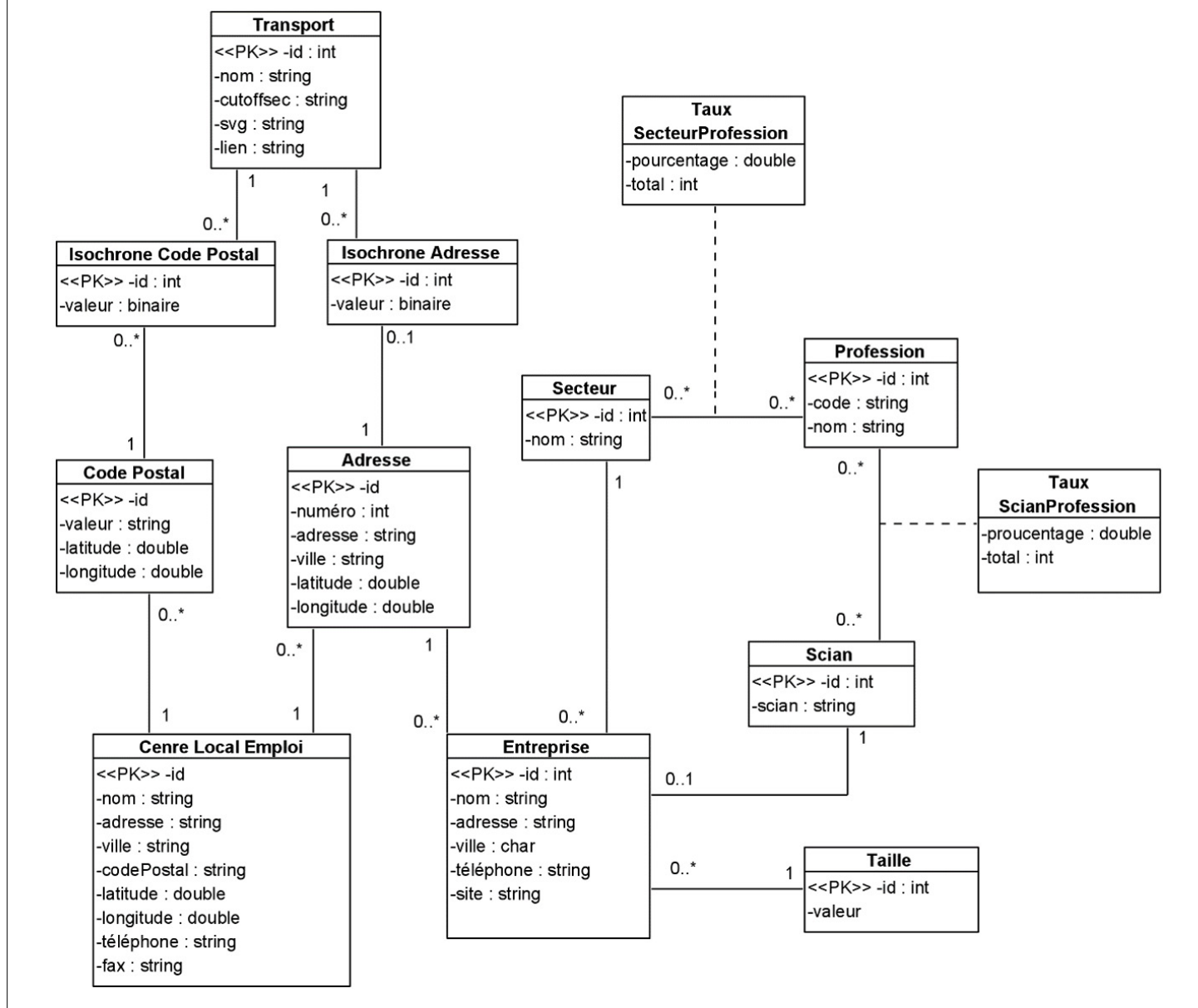
// Requête http vers le serveur OTP pour le transport en commun
const otp = {
  fetchTransit: async function (data) {
    const latLng = `${data['latitude']},${data['longitude']}`
    const param = {
      "fromPlace": latLng,
      "mode": "WALK,TRANSIT",
      "date": "01-25-2020",
      "time": "08:00am",
      "maxWalkDistance": 1500,
      "maxPreTransitTime": 30 * 60,
      "precisionMeters": 75,
      "cutoffSec": data['time'],
    }
    const response = await axios.get("http://127.0.0.1:8801/otp/routers/current/isochrone", {
      params: param
    })
    return response
  }
}

```

Le microservice générateur d'isochrones

Tel que mentionné plus haut, les isochrones pour les quatre modes de transport sont calculées avec l'application Java gratuite OpenTripPlanner. Deux options étaient alors envisageables : 1) créer à la volée sur le serveur les isochrones une fois une localisation et un mode de transport sélectionnés ; 2) créer en amont toutes les isochrones et les stocker dans la base de données PostgreSQL. Puisque le temps de génération des isochrones peut dépasser dix secondes — particulièrement pour celles de 45 minutes en automobile —, nous avons opté pour la seconde option afin d'offrir une expérience de navigation plus rapide pour les usagers de l'application Web.

Figure 2. Schéma conceptuel de la base de données



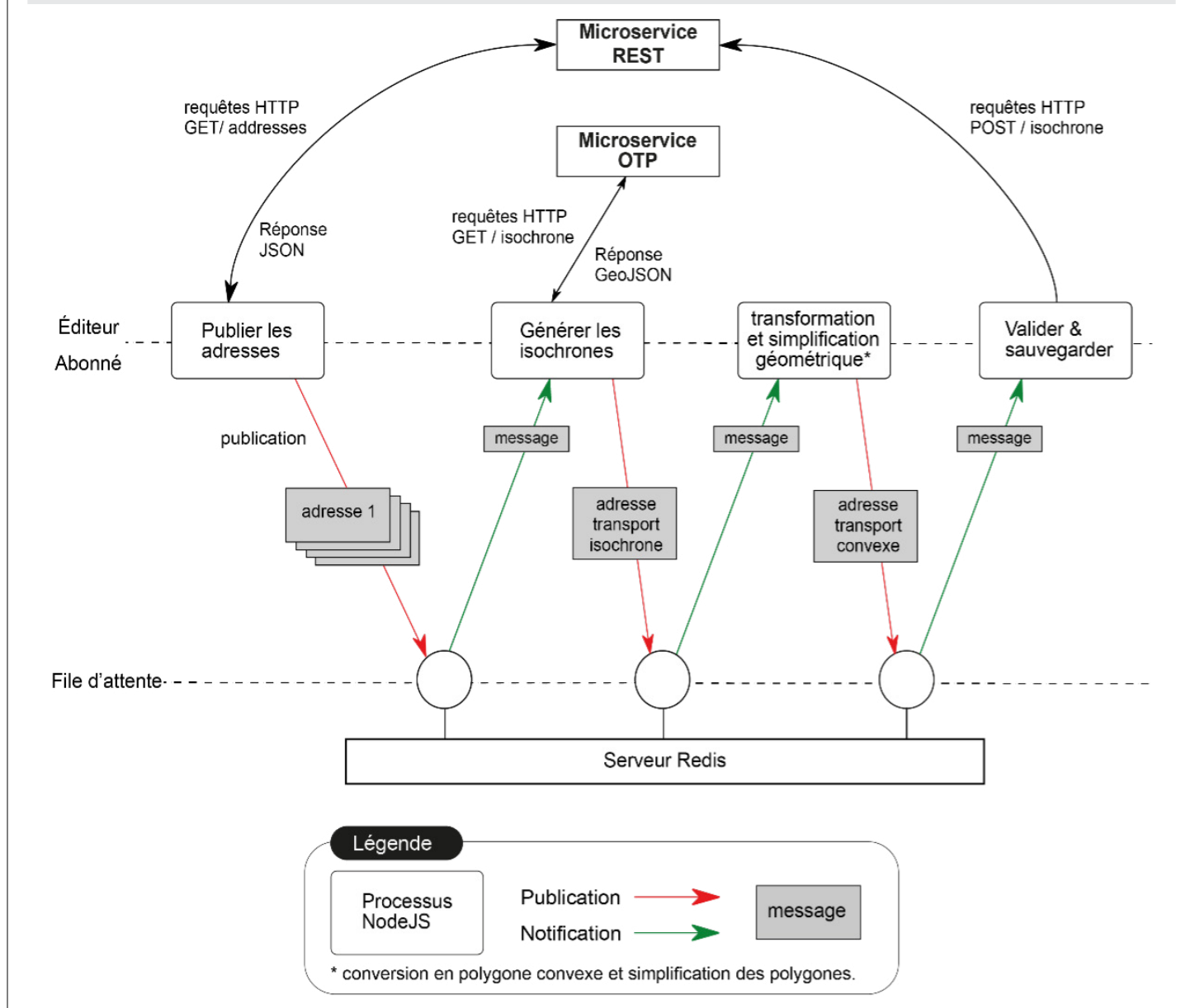
Pour les isochrones en transport en commun, il est à noter que trois critères ont été appliqués. Premièrement, pour une localisation donnée (adresse ou code postal), si aucun arrêt en transport en commun n'est présent à moins de 750 mètres à travers le réseau de rues, aucune isochrone n'est calculée. Deuxièmement, la distance de marche maximale pour un déplacement en transport en commun a été fixée à 1000 mètres. Troisièmement, plusieurs isochrones ont été calculées pour le lundi 18 janvier 2021 à 8 h, 8 h 10, 8 h 20, 8 h 30, 8 h 40 et 8 h 50 ; nous retenons ensuite l'isochrone la plus grande en termes de superficie. Cela explique que certaines isochrones en transport en commun ne sont pas disponibles pour plusieurs adresses et codes postaux. Au final, 4 283 805 ont été calculés soit plus de 90 % des combinaisons possibles (4 modes de transport x 5 isochrones x (227 044 adresses + 10 683 codes postaux) = 4 754 540).

Le microservice calculant les isochrones a été écrit en JavaScript avec la librairie BullJS (Optimalbits, 2021) qui est basée sur une communication interprocessus en publiant des messages sur des files

d'attente (queues FIFO – *first in first out*). Les files d'attente sont hébergées sur un serveur NoSQL Redis (Redis community, 2021). À cette fin, nous avons utilisé trois queues (files d'attente) et quatre processus illustrés à la figure 3. Succinctement, le premier processus récupère les adresses et les codes postaux du service REST (principalement l'identifiant et les coordonnées latitude / longitude de la localisation). Il boucle ensuite sur la liste de ces adresses qu'il envoie une à une et par type de mode de transport (soit quatre messages par localisation). Le deuxième processus se charge d'envoyer les requêtes vers le microservice OTP pour calculer les isochrones et stocke les réponses dans une deuxième queue. Le troisième processus effectue des transformations géométriques – conversion en polygone convexe, simplification des polygones avec la librairie Turf.js (Turf community, 2021) – et les sauvegarde dans une dernière queue. Quant au dernier processus, il valide et compresse l'isochrone geoJSON au format Gzip⁸ et l'envoie vers le service REST pour être stocké dans la base de données PostgreSQL.

⁸ L'étape de compression au format Gzip (Gailly, 2021) est très importante puisqu'elle sauvegarde le fichier geoJSON original du format texte au format binaire. Cela permet un gain d'espace considérable sur la base de données : de 200 à 35 Gb, soit un gain de 82,5%. Aussi, le format binaire est directement envoyé au client (utilisateur), ce qui accélère significativement l'application : le temps de réponse pour l'envoi du fichier passant de 350 à 90 millisecondes.

Figure 3. Le microservice générateur d'isochrones



Cette approche en files d'attente avec BullJS à deux avantages majeurs. Le premier est de permettre une meilleure gestion des ressources informatiques (mémoire et processeur) et une parallélisation des opérations. Chaque processus consomme et publie des messages à des vitesses différentes. Pour une mise à l'échelle, il suffit de lancer plus de processus pour les queues qui se remplissent plus vite. Pour l'étape 1, nous avons utilisé qu'un seul processus. En revanche, pour l'étape 2 (calcul des isochrones), ce fut une dizaine de processus. Un deuxième avantage est la faible consommation de ressources pour chaque processus (mémoire et CPU) qui s'exécute sans impacter le serveur qui héberge le site web et la base de données.

















Le microservice Web

Le service Web a été développé avec la librairie JavaScript React (Walke, 2021) de Facebook qui permet de développer un site selon un modèle d'application web monopage. Le site se comporte comme à une application mobile, c'est-à-dire que la totalité du site est chargée lors de la première visite ; le reste des échanges ne sont que des transferts de données au format JSON avec le protocole AJAX, ce qui rend l'application plus fluide. Du point de vue architectural, une

application REACT est un ensemble de composants indépendants qui forme les pages. Pour la communication intercomposante et la propagation des mises à jour, nous utilisons la librairie JavaScript Redux (Abramov, 2021) qui permet de centraliser les données et de simplifier les rafraîchissements de l'interface. En résumé, le site n'a pas d'application dorsale (*backend*) puisque c'est le service REST qui joue ce rôle.

DESCRIPTION DES FONCTIONNALITÉS ET MODE D'EMPLOI DE L'APPLICATION CARTOGRAPHIQUE

À partir de la figure 4, nous proposons de décrire les éléments (représentés par des chiffres de 1 à 7) et fonctionnalités (identifiées par des lettres a à k) du *Géolocalisateur d'entreprises de Lanaudière*. Comme pour les applications cartographiques développées récemment (Apparicio et al., 2019), concernant l'interface, nous avons largement privilégié le recours à des icônes au détriment du texte afin qu'elle soit la plus intuitive possible et rapide d'utilisation. La description de ces icônes est disponible au tableau 1.

Tableau 1. Description des icônes utilisées dans l'application	
Mode de transport	
	Vélo
	Marche
	Automobile
	Transport en commun
Isochrone	
	10 minutes
	15 minutes
	20 minutes
	25 minutes
	30 minutes
	45 minutes
	60 minutes
	75 minutes
Fonctions de navigation sur la carte	
	Agrandissement de la carte
	Réduction de la carte
	Centrage de la carte sur un élément (localisation ou bureau de Services Québec)
Autres fonctions	
	Mettre une entreprise dans ses favoris
	Renvoi dans un nouvel onglet le trajet entre la localisation et une entreprise dans Google Maps en fonction du mode de transport sélectionné

L'utilisation de l'application repose sur sept principaux composants identifiés par les symboles ❶ à ❷ :

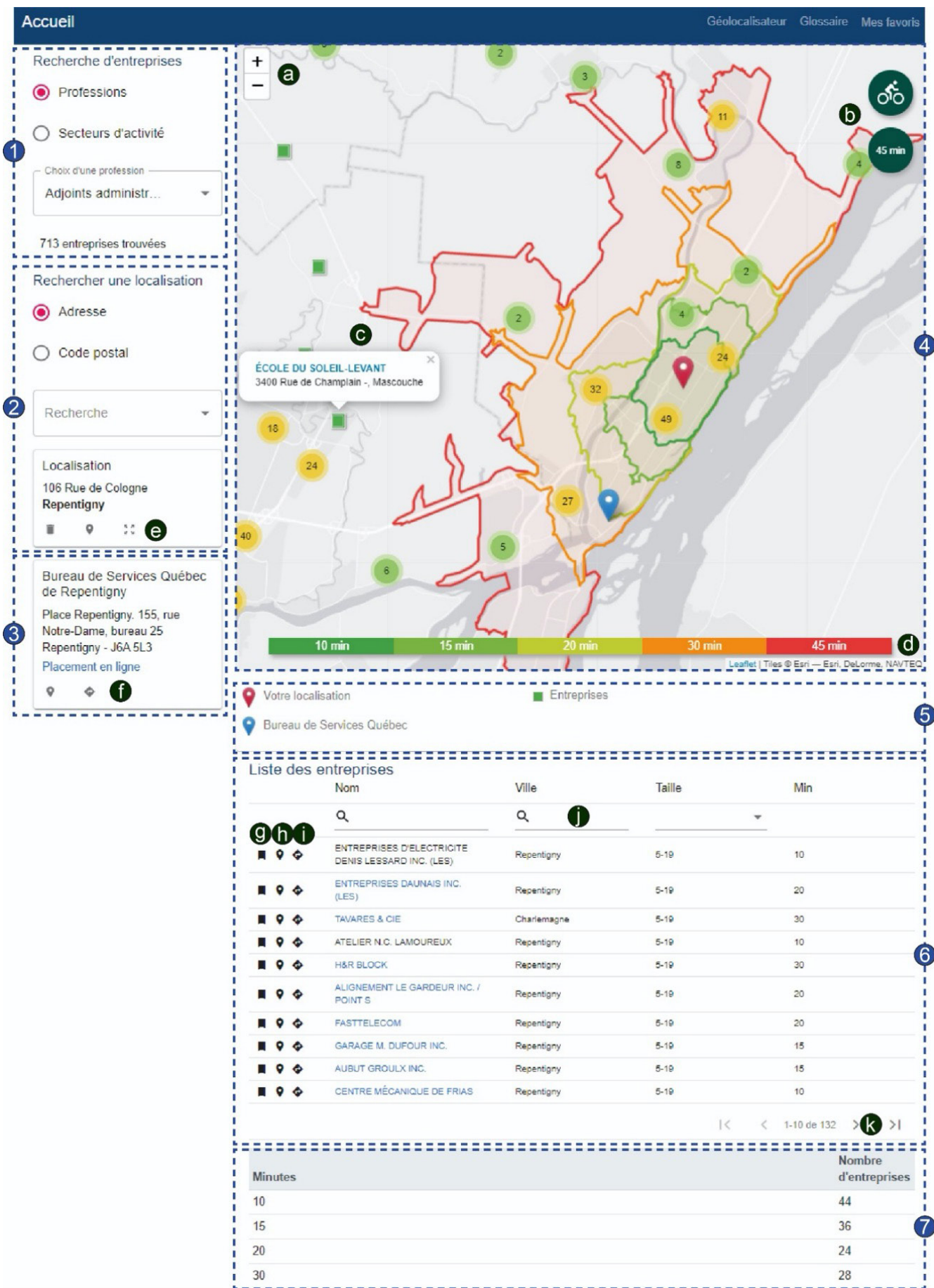
❶ Le premier module permet de sélectionner les entreprises en fonction d'une seule profession ou d'un à trois secteurs d'activité (parmi 27 secteurs) à l'aide d'un menu déroulant dans laquelle la saisie de texte est possible. Par exemple, en tapant «*ingé*», toutes les professions relatives à l'ingénierie seront disponibles. Une fois ce choix réalisé, un court texte dans ce module indique le nombre d'entreprises sélectionnées dans l'ensemble de la région (ici, 713 à la figure 4). Ces entreprises sont automatiquement positionnées sur la carte.

- ❷ Dans le second module, l'utilisateur peut choisir sa localisation à partir d'une adresse ou d'un code postal qu'il peut saisir dans le menu déroulant.
- ❸ Lorsque l'utilisateur a choisi une localisation, le bureau de Services Québec est affiché. Notez que ce bureau n'est pas forcément le plus proche, mais celui qui dessert le territoire du bureau dans laquelle est comprise la localisation.
- ❹ Une fois la sélection d'entreprises et le choix d'une localisation réalisée, les isochrones en fonction du mode de transport sélectionné seront affichées sur la carte.
- ❺ Le cinquième module est tout simplement la légende de la carte.
- ❻ Dans le sixième module est reportée la liste des entreprises comprises dans les différentes isochrones.
- ❼ Dans le dernier module, un tableau récapitulatif comprend les effectifs des entreprises comprises dans chaque isochrone.

Plusieurs fonctionnalités de cartographie interactive ou d'exploration des données sont implémentées dans l'application et sont identifiées par les symboles a à k :

- a Les fonctions classiques de navigation sur la carte (agrandissement et réduction) qui peuvent être aussi utilisées avec le bouton Ctrl et la molette de la souris.
- b Le choix du mode de transport (à pied, en vélo, en automobile et en transport en commun).
- c Les informations sur une entreprise en cliquant sur sa localisation.
- d Le centrage de la carte sur une des isochrones.
- e Le centrage de la carte sur la localisation choisie par l'utilisateur.
- f Le centrage de la carte sur le bureau de Services Québec. Il est aussi possible d'obtenir le trajet entre la localisation choisie et ce bureau dans GoogleMaps en fonction du mode de transport sélectionné en cliquant sur la seconde icône; le résultat est présenté dans un nouvel onglet.
- g L'ajout d'une entreprise dans les favoris de l'utilisateur. Ces favoris sont sauvegardés par l'intermédiaire d'un fichier témoin (*cookie*) afin qu'ils soient conservés entre deux sessions de navigation. L'utilisateur peut ensuite accéder à la page de ses favoris dans laquelle sont listées dans un tableau les informations sur ses entreprises favorites. Il a alors la possibilité d'imprimer cette liste.
- h Le centrage de la carte sur la localisation d'une entreprise.
- i Le trajet dans GoogleMaps entre la localisation choisie et l'entreprise en fonction du mode de transport sélectionné. Le résultat est présenté dans un nouvel onglet.
- j La possibilité de filtrer et trier le tableau des entreprises en saisissant du texte dans les colonnes *Nom*, *Ville* ou en sélectionnant une catégorie de la variable *taille des entreprises*.
- k La liste des entreprises est paginée dans un tableau comprenant dix entreprises à la fois. Il est donc possible d'afficher des pages suivantes à partir de cette fonctionnalité.

Figure 4. Fonctionnalités et mode d'emploi de l'application



DISCUSSION ET CONCLUSION

Le *Géolocalisateur d'entreprises de Lanaudière* est un outil particulièrement intéressant pour explorer la distribution spatiale des entreprises dans la région de Lanaudière autour d'une localisation géographique, et ce, en fonction de quatre modes de transport (marche, vélo, automobile et transport en commun), d'un à trois secteurs d'activité ou d'une profession (CNP-4). Son architecture repose sur des microservices, soit une approche qui s'est avérée particulièrement efficace à la fois pour la mettre en œuvre et pour la faire évoluer rapidement. Du point de vue géographique, l'application gratuite OpenTriPlanner est particulièrement puissante pour délimiter des isochrones à partir de données ouvertes relatives aux réseaux de transport (données OpenStreetMap pour les réseaux routiers et cyclables et données GTFS pour les réseaux de transport). Aussi, l'utilisation de la librairie JavaScript React pour le développement du service Web a permis de déployer un modèle d'application web monopage qui rend la navigation sur le site particulièrement fluide. Rappelons que, comme son nom l'indique, le territoire d'étude se limite à la région de Lanaudière. Or, les microservices étant développés et fonctionnels, ils pourraient être étendus à d'autres régions du Québec, voire à l'ensemble de la province. À cette fin, le seul enjeu sera d'intégrer les données des adresses et des codes postaux (produits d'Adresses Québec) et des entreprises (répertoire des établissements d'Emploi-Québec) pour l'ensemble de la province.

Comme toute application, le *Géolocalisateur d'entreprises de Lanaudière* comprend plusieurs limites inhérentes aux données sémantiques et spatiales utilisées. Premièrement, le nombre et la localisation des entreprises varient dans le temps. En accord avec notre partenaire, la liste des entreprises sera mise à jour trois fois par année, ce qui nécessite une maintenance du site sur une base régulière. Nous avons signalé que l'application peut constituer un outil particulièrement intéressant pour la recherche d'emploi puisqu'il permet de repérer les entreprises localisées près de chez soi en fonction d'un mode de transport, d'un secteur d'activité ou d'une profession. Toutefois, concernant les données sur les entreprises, aucune information n'indique si elles ont actuellement un ou plusieurs postes à combler. Cela nécessiterait l'accès et l'appariement à d'autres bases de données, soit une opération plus complexe à mettre en œuvre.

Deuxièmement, la sélection des entreprises peut se faire selon 41 secteurs d'activité ou selon une profession donnée (CNP-4). Or, pour une profession donnée, la sélection est réalisée à partir d'une matrice entre les professions (CNP-4) et les secteurs d'activité (SCIAN-4). Bien qu'intéressante, la sélection des entreprises pour une profession est approximative. Autrement dit, il est possible qu'elle renvoie parfois des entreprises dans lesquelles la profession choisie est peu ou non présente.

Troisièmement, les données spatiales relatives aux localisations (adresses et codes postaux) et aux réseaux de transport évoluent aussi dans le temps. Les produits d'adresse Québec AQ *géobâti* et AQ *géopostal* sont aussi mis à jour régulièrement : mensuellement pour les adresses et à chaque trimestre pour les codes postaux (Gouvernement du Québec, 2021, p. 4). Il s'agira alors de développer un autre microservice pour intégrer les nouvelles données d'adresses et de codes postaux et de générer leurs isochrones. Il en va de même pour les données GTFS pour les réseaux de transport qui sont mises à jour habituellement une ou deux fois par année, bien que les changements soient habituellement mineurs (principalement des changements d'horaire et plus rarement l'ajout d'une nouvelle ligne de bus). Là encore, un microservice de mise à jour des réseaux pourrait être écrit avec Python ou R. Il s'agirait alors de récupérer les nouvelles données GTFS et OSM, de reconstruire

le graphe et de relancer les calculs des isochrones, soit une tâche somme toute assez chronophage qu'il convient de paralléliser sur plusieurs ordinateurs.

GLOSSAIRE

API

Application Programming Interface (interface de programmation applicative).

« C'est un ensemble normalisé de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une bibliothèque logicielle ou un service web, le plus souvent accompagnée d'une description qui spécifie comment des programmes consommateurs peuvent se servir des fonctionnalités du programme fournisseur » (https://fr.wikipedia.org/wiki/Interface_de_programmation).

Protocole http

Hypertext Transfer Protocol (protocole de transfert hypertexte).

« [...] Protocole de communication client-serveur développé pour le World Wide Web. HTTPS (avec S pour secured, soit « sécurisé ») est la variante sécurisée par l'usage des protocoles Transport Layer Security (TLS) » (https://fr.wikipedia.org/wiki/Hypertext_Transfer_Protocol). Ce protocole permet l'envoi d'information à partir d'une connexion établie entre deux ordinateurs. Le client envoie une requête HTTP vers un serveur qui après traitement renvoie une réponse contenant l'information demandée par le client.

Service informatique

Dans une architecture orientée service, le terme service fait référence à une fonctionnalité logicielle ou à un ensemble de fonctionnalités logicielles (telles que la récupération d'informations ou l'exécution d'algorithmes) dans le but que différents clients peuvent les réutiliser pour leur propre usage. (https://fr.wikipedia.org/wiki/Architecture_orientée_services).

SQL

Structured Query Language (langage de requête structurée).

« [...] un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles ». (https://fr.wikipedia.org/wiki/Structured_Query_Language).

Financement

Le site Internet du *géolocalisateur d'entreprises de Lanaudière* (<https://www.geolocalisateur-lanaudiere.ucs.inrs.ca/>) a été développé dans le cadre d'une entente de partenariat entre Services Québec Lanaudière et l'Institut national de la recherche scientifique (Centre Urbanisation Culture Société). La rédaction du présent article a par contre été soutenue par la Chaire de recherche du Canada en équité environnementale (950-230813) et le laboratoire d'équité environnementale (<http://laeq.ucs.inrs.ca/>).

BIBLIOGRAPHIE

- Abramov, D. (2021). *Redux (JavaScript library)*. <https://redux.js.org/>.
- Apparicio, P., Gagnon, M. et Franjul, I. (2014). *Atlas numérique de la zone métropolitaine de l'emploi de Montréal* (édition 2014). Repéré à <http://www.atlas-zme.qc.ca/>

- Apparicio, P., Gelb, J., Lachance, A., Carrier, M., Mathieu, M.-È., Lecot, F. & Kingham, S. (2018). Comparaison de l'exposition individuelle aux pollutions atmosphérique et sonore à Montréal selon le mode de transport utilisé. Repéré à <http://atlaspollutionmtl.ucs.inrs.ca/>
- Apparicio, P., Gelb, J. & Mathieu, M.-È. (2019). Un atlas-web pour comparer l'exposition individuelle aux pollutions atmosphérique et sonore selon le mode de transport. *Cybergeog: European Journal of Geography*.
- Apparicio, P., Lesage-Mann, É., Guemache, A., Gelb, J. & Del Carmen Sanchez-Rojas, L. (2019). Atlas électronique de l'emploi de la région de Lanaudière. Repéré à <http://www.atlas-lanaudiere.ucs.inrs.ca/>
- Apparicio, P., Maignan, D. & Gelb, J. (2021). Géolocalisateur d'entreprises de Lanaudière : localiser des entreprises près de vous. Repéré à <https://www.geolocalisateur-lanaudiere.ucs.inrs.ca/>
- Apparicio, P., Mathon, D., Séguin, A.-M. & Gagnon, M. (2014). *Re-cension des écrits sur la notion de mauvais appariement spatial entre les lieux de résidence et d'emploi*. Montréal: Institut national de la recherche scientifique.
- Apparicio, P. & Petkevitch, V. (2006). Déploiement d'atlas interactifs sur Internet: nouvelles avenues avec le Scalable Vector Graphics, le C# et l'ASP. Net. *Cybergeog: European Journal of Geography*.
- Barrette, N., Vandersmissen, M.-H., Roy, F., Lalonde, B., Tanguay, L.-P. & Dionne, M. (2020). Atlas interactif de la vulnérabilité de la population québécoise aux aléas climatiques. Repéré à <https://atlas-vulnerabilite.ulaval.ca/vague-de-chaaleur/>
- Bayer, M. (2012). SQLAlchemy. Dans A. Brown & G. Wilson (dir.), *The Architecture of Open Source Applications Volume II: Structure, Scale, and a Few More Fearless Hacks*. Mountain View: aosabook.org, <http://aosabook.org/en/sqlalchemy.html>.
- Boisjoly, G. & El-Geneidy, A. (2016). Daily fluctuations in transit and job availability: A comparative assessment of time-sensitive accessibility measures. *Journal of transport geography*, 52: 73-81.
- Costantino, D., Angelini, M., Alfio, V., Claveri, M. & Settembrini, F. (2020). Implementation of a system WebGIS open-source for the protection and sustainable management of rural heritage. *Applied Geomatics*, 12(1): 41-54.
- Fawler, M. (2014). *Microservices: a definition of this new architectural term*. <https://martinfowler.com/articles/microservices.html>.
- Gailly, J.-L. (2021). *GNU Gzip*. <https://www.gnu.org/software/gzip/>.
- Gouvernement du Québec (2021). *Adresses Québec: Guide de l'utilisateur (version 2.60)*. Québec: Ministère de l'Énergie et des Ressources naturelles, Direction de la cartographie topographique, https://adressesquebec.gouv.qc.ca/pdf/guide_utilisateurs.pdf.
- Grengs, J. (2010). Job accessibility and the modal mismatch in Detroit. *Journal of Transport Geography*, 18(1): 42-54.
- Haustant, A. (2019). *Flask-restplus documentation*. <https://readthedocs.org/projects/flask-restplus/downloads/pdf/latest/> et <https://flask-restplus.readthedocs.io/en/stable/>.
- Hipp, D. R. (2020). *SQLite*. <http://sqlite.org/>.
- Houston, D. (2005). Employability, skills mismatch and spatial mismatch in metropolitan labour markets. *Urban Studies*, 42(2): 221-243.
- Kain, J. F. (1968). Housing segregation, negro employment, and metropolitan decentralization. *The Quarterly Journal of Economics*, 82(2): 175-197.
- Kain, J. F. (1992). The spatial mismatch hypothesis: three decades later. *Housing policy debate*, 3(2): 371-460.
- Kraak, J.-M. & Brown, A. (2003). *Web cartography*. CRC Press.
- Lévesque, C., Apparicio, P., Radu, I., Gagnon, M. & Maignan, D. (2019a). Atlas de la population des Premières Nations du Québec inscrite au Registre des Indiens. Repéré à <http://geoatlas.reseaudialog.ca/affaires-indiennes-quebec/comparaison/>
- Lévesque, C., Apparicio, P., Radu, I., Gagnon, M. & Maignan, D. (2019b). Atlas des activités des chasseurs et piégeurs cris de Eeyou Istchee. Repéré à <http://geoatlas.reseaudialog.ca/chasseurs-piegeurs-cris/accueil/>
- Manacorda, M. & Petrongolo, B. (1999). Skill mismatch and unemployment in OECD countries. *Economica*, 66(262): 181-207.
- Morgan, M., Young, M., Lovelace, R. & Hama, L. (2019). OpenTrip-Planner for R. *Journal of Open Source Software*, 4(44): 1926.
- Ong, P. M. & Miller, D. (2005). Spatial and transportation mismatch in Los Angeles. *Journal of Planning Education and Research*, 25(1): 43-56.
- OpenStreetMap Contributors. (2021). OpenStreetMap. www.openstreetmap.org.
- Optimalbits (2021). *Welcome to Bull's Guide*. <https://optimalbits.github.io/bull/>.
- Owens, M. et Allen, G. (2010). *SQLite*. Springer.
- Padgham, M., Lovelace, R., Salmon, M. & Rudis, B. (2017). *osmdata*. *Journal of Open Source Software*, 2(14).
- Pastor, M. J. & Marcelli, E. A. (2000). Men N the hood: skill, spatial, and social mismatch among male workers in Los Angeles county. *Urban Geography*, 21(6): 474-496.
- Patacchini, E. & Zenou, Y. (2005). Spatial mismatch, transport mode and search decisions in England. *Journal of Urban Economics*, 58(1): 62-90.
- Peterson, M. P. (1995). *Interactive and animated cartography*. Prentice Hall.
- Peterson, M. P. (2005). *Maps and the Internet*. Elsevier.
- Peterson, M. P. (2008a). *International perspectives on maps and the Internet*. Springer.
- Peterson, M. P. (2008b). Maps and the Internet: what a mess it is and how to fix it. *Cartographic perspectives*(59): 4-11.
- Price, A., Langford, M. & Higgs, G. (2021). Computing geographical access to services: The design of a client-server solution that incorporates multiple transport modes. *Transactions in GIS*.
- QGIS.org (2021). *QGIS Geographic Information System*. QGIS Association. <http://www.qgis.org>.
- R Core Team (2021). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, <https://www.R-project.org/>.
- Redis community (2021). *Redis documentation*. <https://redis.io/documentation>.
- Sanchez, T. W. (1999). The connection between public transit and employment. *Journal of the American Planning Association*, 65(3), 284-296.
- Taylor, B. D. & Ong, P. M. (1995). Spatial mismatch or automobile mismatch? An examination of race, residence and commuting in US metropolitan areas. *Urban Studies*, 32(9): 1453-1473.

The PostgreSQL Global Development Group (2021). *PostgreSQL 10.16 Documentation*. <https://www.postgresql.org/docs/10/index.html> et <https://www.postgresql.org/files/documentation/pdf/10/postgresql-10-A4.pdf>.

Trimaille, E. (2021). *QuickOSM*. <https://docs.3liz.org/QuickOSM/>.

Turf community (2021). *Turf.js documentation*. <https://turfjs.org/> et <https://github.com/Turfjs>.

Vallée, J., Douet, A., Lecomte, C., Le Roux, G., Commenges, H., Villard, É. & Riot, M. (2020). Mobiliscope: la ville à tout heure. Repéré à <https://mobiliscope.cnrs.fr/>

Walke, J. (2021). *React (JavaScript library)*. <https://reactjs.org/>.