**Document de spécifications des interfaces**

Date de création : **07/10/2014**

Version : **0.2 du 17/04/2015**

SpÉcifications des interfaces

**Sommaire**

1 Historique des modifications 4

2 Documents applicables 5

3 Introduction 6

4 Partie 1 7

4.1 Termes et définitions 7

4.1.1 Dictionnaire des termes 7

4.1.2 Illustration des termes sur un exemple 9

4.2 Les acteurs 15

4.2.1 Le client du méta-système 15

4.3 Le périmètre fonctionnel du méta-système 15

4.3.1 Les frontières fonctionnelles du service de recherche d’itinéraire distribué 15

4.3.2 Les frontières fonctionnelles du service de recherche de localité 18

4.3.3 L’objectif du service 18

4.4 Les composants internes 19

4.4.1 La base des métadonnées 19

4.4.2 Le système de synchronisation des métadonnées 19

4.4.3 L’aiguilleur 20

4.5 Les composants externes 20

4.6 Les interfaces entre les composants 20

4.6.1 La base des métadonnées 20

4.6.2 Le système de synchronisation des métadonnées 20

4.6.3 L’aiguilleur 21

4.6.4 Vue d’ensemble des interfaces 21

4.7 Principaux cas d’utilisation 21

5 Partie 2 25

5.1 Interfaces DRI et FRI 25

5.1.1 Structures de données échangées 26

5.1.2 Demi interface fournisseur de recherche d’itinéraire et de localité (FRI) 26

5.1.3 Demi interface demandeur de recherche d’itinéraire et de localité (DRI) 29

5.1.4 Comportement dynamique de la demi-interface FRI 30

5.1.5 Comportement dynamique de la demi-interface DRI 36

5.1.6 Conditions de fonctionnement 37

5.2 Interface langage de requêtes sur les métadonnées 37

5.2.1 Le schéma de la base 37

5.2.2 Contrainte d’optimisation d’accès aux arrêts 40

5.2.3 Contraintes de mise en fonctionnement 40

5.3 Interface générique de SIM 40

5.3.1 Descriptions générales des services de l’interface 41

5.3.2 Service de recherche d’itinéraire détaillée 1-n 42

5.3.3 Service de recherche d’itinéraire non détaillée n-m 45

5.3.4 Collecte des arrêts 47

5.3.5 Capacités du SIM 48

6 Partie 3 49

6.1 Point d’accès fonctionnel 49

6.1.1 Accès au méta-système (FRI, DRI) 49

6.1.2 Accès à la base des métadonnées 49

6.1.3 Accès à l’interface générique de SIM 49

6.2 Structures échangées 49

6.2.1 Les structures d’échange des interfaces du méta-système (FRI, DRI) 50

6.2.2 Les structures d’échange des interfaces génériques de SIM 50

6.2.3 Les structures d’échange de l’interface de lecture écriture de la base des métadonnées 50

7 Liste des figures 51

# Historique des modifications

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date de publication | Description des changements | Auteur |
| 0.1 | 07/10/2014 | Création du document | Marc FLORISSON |
| 0.2 | 15/04/2015 | Modification et validation du document | Marc FLORISSON  Stephan SIMART |

# Documents applicables

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Titre | Contenu | Date | Version | Auteur |
| Dossier des définitions XSD des structures d’échange | https://github.com/CanalTP-Cityway/documentation/xsd/V1.6 : Dossier contenant l’ensemble des schémas XSD utilisés pour les interfaces :   * MisProtocol.xsd * MisPlanTrip.xsd * MisPlanSummedUpTrip.xsd * MisLocationTime.xsd * MisCollectStop.xsd * MisCapabilities.xsd * PlanTrip.xsd * SearchPoints.xsd * Protocol.xsd * Site.xsd * Itinerary.xsd * ItineraryRequest.xsd * ProtocolFramework.xsd * Répertoire gml des schémas GML * Répertoire netex des schémas NETEX | 17/04/15 | 1.6 | Marc FLORISSON,  Stephan SIMART |
| pdf-data-exchange-apii-sim.zip  Documents PDF de présentation des schémas des structures d’échange | https://github.com/CanalTP-Cityway/documentation/xsd/V1.6/pdf :  Archive contenant les documents PDF qui présentent les structures d’échanges des interfaces :   * PlanTrip.pdf * SearchPoints.pdf * MisPlanSummedUpTrip.pdf * MisPlanTrip.pdf * MisCollectStop.pdf * MisCapabilities.pdf | 17/04/15 | 1.6 | Marc FLORISSON |

# Introduction

Le document décrit les interfaces nécessaires à la mise en place de l’architecture d’un système de calcul d’itinéraire réparti.

Le document spécifie les différentes composantes de l’architecture en restant à un niveau suffisamment général pour ne pas se lier à quelque question d’implémentation que ce soit.

Le document se compose de 3 parties :

* la première partie définit les termes et concepts principaux, spécifie les fonctionnalités de l’architecture, identifie les composants logiciels de l’architecture et décrit les principaux cas d’usage
* la deuxième partie spécifie les interfaces entre les composantes logicielles
* la troisième partie spécifie les points d’accès fonctionnels et les messages échanges à travers les interfaces

# Partie 1

Ce chapitre commence une définir un ensemble de termes permettant d’illustrer l’objectif de l’architecture d’un système de recherche d’itinéraire distribué.

Le chapitre détaille ensuite le périmètre fonctionnel de l’architecture puis identifie les composants fonctionnels de l’architecture.

Le chapitre se termine avec une présentation des principaux cas d’usage.

## Termes et définitions

### Dictionnaire des termes

* Service de recherche d’itinéraire distribué
  + C’est est un service de recherche d’itinéraire particulier. Ce service renvoie des itinéraires composés, la composition des itinéraires résulte de la répartition du calcul entre les différents SIM.
* Recherche de localité
  + Ce service consiste à proposer des localités (qui peuvent être des arrêts de transport en commun, des points d’intérêt ou des adresses) qui satisfont à un critère de sélection qui peut être : un texte ou une zone géographique.
* Méta-système
  + C’est le système qui fournit les services de calcul d’itinéraire distribué et de recherche de localité.
* Métadonnées
  + Ce sont les données qui décrivent :
    - Les SIM connectés au méta-système et toutes les propriétés nécessaires à l’utilisation de leur interface générique de SIM.
    - Les transitions entre les arrêts des SIM connectés au méta-système
* SIM (système d’information multimodal)
  + On désigne par SIM, tout web-service autonome de calcul d’itinéraire mis en place par une autorité organisatrice ou un transporteur quels qu’ils soient
  + Par exemple
    - Le web-service sous-jacent au site [www.vialsace.eu](http://www.vialsace.eu) sera appelé SIM Alsace
    - Le web-service sous-jacent au site [www.simplicim-lorraine.eu](http://www.simplicim-lorraine.eu) sera appelé SIM Lorraine
    - Le web-service sous-jacent au site [www.ter-sncf.com](http://www.ter-sncf.com) sera appelé SIM SNCF
  + Tout web-service de calcul d’itinéraire est pris en compte de façon neutre. Il n’y aura pas de distinction entre un SIM « longue-distance » et un SIM « urbain ». Seules les capacités et la couverture géographique sont discriminantes.
  + Il en résulte que plusieurs SIM « longue-distance » peuvent être intégrés au projet et mis en concurrence pour chaque calcul d’itinéraire
* SIM connecté
  + Il s’agit d’un SIM connecté au méta-système et décrit dans les méta-données
  + Un SIM connecté implémente l’interface générique de SIM
* Trace
  + Une « trace » correspond à une succession de SIM qui vérifie un certain nombre de propriétés en fonction de la requête de recherche d’itinéraire distribuée :
    - Le point de départ de la requête est situé dans le premier SIM de la succession.
    - Le point d’arrivée de la requête est situé dans le dernier SIM de la succession.
    - Pour tous les SIM de la succession, il existe au moins une transition entre un arrêt du SIM et un arrêt du SIM qui précède dans la succession.
  + Pour chaque requête d’itinéraires, plusieurs traces seront possibles. Ce qui induit que chaque SIM peut se retrouvé en position de SIM « intermédiaire » de façon contextuelle.
  + Par exemple, pour une demande de Charleville-Mézières à Strasbourg, on pourra imaginer
    - Un itinéraire composé des SIM Champagne-Ardenne > Lorraine > Alsace
    - Un itinéraire composé des SIM Champagne-Ardenne > SNCF > Alsace
    - Un itinéraire composé des SIM Champagne-Ardenne > aérien > Alsace
  + Chaque trace est donc composée de :
    - SIM d’origine
    - SIM via (facultatif dans le cas d’une trace transfrontalière ou mono-SIM)
    - SIM de destination (facultatif dans le cas d’une trace mono-SIM)
* Transition
  + C’est une relation entre un arrêt de départ et un arrêt d’arrivée, chacun des arrêts étant issus de SIM distincts mais suffisamment proches pour permettre une correspondance au cours d’un itinéraire composé.
  + C’est une notion purement géographique, caractérisée par :
    - L’arrêt de départ et son SIM d’appartenance
    - L’arrêt d’arrivée et son SIM d’appartenance
    - La durée de transition standard, éventuellement décomposé en durée de marche à pied et durée d’attente minimale (cette dernière correspondant à la tolérance d’exécution de l’offre)
    - Une durée de transition orientée « PMR »
  + On pourra envisager ultérieurement d’étendre le concept avec les ID de la base arrêt nationale, avec le cheminement dans le pôle d’échange
* Point de transition
  + Un point de transition est un arrêt sur lequel est défini une transition.
* Interface générique de SIM
  + Cette interface permet l’échange d’information entre l’aiguilleur et les SIM
  + Elle doit être exposée par tous les SIM qui sont intégrés au projet
* Traducteur
  + Composant interne à chaque SIM permettant de transformer sa propre interface au format de l’interface générique interne attendue par l’aiguilleur
  + Ce composant, spécifique à chaque calculateur d’itinéraire, est de la responsabilité du SIM
  + Il devra en particulier
    - Gérer les rabattements spécifiques au départ/à l’arrivée
    - Présenter la feuille de route complète
    - Par exemple, un SIM aérien doit fournir les durées d’embarquement dans ses réponses.
* Rabattement
  + Le rabattement correspond à un déplacement qui s’effectue dans un mode de transport vélo ou voiture (un mode « non transport en commun ») et qui permet de relier un arrêt de l’offre de transport en commun.
  + Les emplacements prévus pour le stationnement (parc relais, parking à vélo) facilitent ce type de déplacement qui utilise la multi-modalité voiture, vélo, transport en commun.
  + Les rabattements peuvent s’effectuer en début ou fin d’itinéraire.
* Itinéraire composé
  + Un itinéraire composé est un itinéraire constitué de d’itinéraires partiels reliés par des transitions. Chaque itinéraire partiel correspond à un résultat de recherche d’itinéraire proposé par un SIM particulier.
  + L’itinéraire composé met en œuvre plusieurs SIM et ne peut pas être fourni par un SIM unique. L’objectif du projet est d’adresser ce type d’itinéraire.
* Transfrontalier
  + Un itinéraire transfrontalier est un itinéraire qui part d’un SIM 1 et arrive dans un SIM 2, les SIM 1 et 2 partageant une frontière (entre 2 départements ou 2 régions ou 2 pays). Par exemple la région Lorraine et la région Alsace sont transfrontalières.
  + Ces cas d’itinéraires sont les plus complexes à adresser, les 2 SIM contenant souvent l’offre de transport en double, et les points de transitions étant les plus nombreux.

### Illustration des termes sur un exemple

L’exemple utilisé met en jeu 5 SIM sur un territoire.

Les 4 SIM (SIM 0, SIM 1, SIM 2, SIM 3) couvrent chacun une partie seulement de ce territoire.

Le SIM LD correspond à un SIM longue distance.

|  |  |
| --- | --- |
| Figure Territoire et SIM en jeu | Le SIM 0 est entièrement couvert par le SIM 1 (par exemple : le SIM 0 est urbain et le SIM 1 est régional).  Le départ se situe en SIM 1 ou SIM 0 et l’arrivée en SIM 2. |

La matrice des points de transition calculée permet d’identifier les correspondances suivantes :

|  | **SIM 0** | **SIM 1** | **SIM 2** | **SIM 3** | **SIM LD** |
| --- | --- | --- | --- | --- | --- |
| SIM 0 |  | K à S |  |  |  |
| SIM 1 | K à S |  | I | B, D | A |
| SIM 2 |  | I |  | J, F | H, G |
| SIM 3 |  | B, D | J, F |  | C, E, F |
| SIM LD |  | A | H, G | C, E, F |  |

On obtient alors le graphe de trace suivant :

|  |  |
| --- | --- |
| Figure Traces des SIM | Les traces suivantes vont être calculées :   * SIM 1 vers SIM 2 en direct * SIM 1 > SIM LD > SIM 2 * SIM 1 > SIM 3 > SIM2 * SIM 0 > SIM 1 > SIM 2   Les traces suivantes ne seront pas calculées :   * SIM 1 > SIM LD > SIM 3 > SIM 2 * SIM 1 > SIM 3 > SIM LD > SIM 2 * SIM 0 > SIM 1 > SIM 2 > SIM 3 * etc |

|  |
| --- |
| Trace 1 : SIM 1 > SIM 2 |
| Récupération des points de transition : uniquement le point I |
| Calculs locaux    Calcul local SIM 1 : de départ vers I  Calcul local SIM 2 : de I vers arrivée |
| Reconstitution du trajet complet de la trace 1 |

|  |
| --- |
| Trace 2 : SIM 1 > SIM LD > SIM 2 |
| Récupération des points de transition    Points de transition au départ : A  Points de transition à l’arrivée : H et G |
| Calculs locaux    Calcul local sim LD : meilleur trajet entre A et H/G  Calcul local sim 1 : départ vers A  Calcul local sim 2 : meilleur trajet entre H/G et arrivée |
| Reconstitution du trajet complet de la trace 2 |

|  |
| --- |
| Trace 3 : SIM 1 > SIM 3 > SIM2 |
| Récupération des points de transition    Points de transition au départ : B et D  Points de transition à l’arrivée : J et F |
| Calculs locaux    Calcul local sim 3 : meilleur trajet entre B/D et J/F  Calcul local sim 1 : meilleur trajet entre départ et B/D  Calcul local sim 2 : meilleur trajet entre D/F et arrivée |
| Reconstitution du trajet complet de la trace 3 |

|  |
| --- |
| Trace 4 : SIM 0 > SIM 1 > SIM 2 |
| Récupération des points de transition    Points de transition au départ : K à S  Points de transition à l’arrivée : I |
| Calculs locaux    Calcul local sim 1 : meilleur trajet entre K/…/S et I  Calcul local sim 1 : meilleur trajet entre départ et K/L/M/…/S  Calcul local sim 2 : meilleur trajet entre I et arrivée |
| Reconstitution du trajet complet de la trace 4 |

|  |
| --- |
| Synthèse des résultats |
|  |

## Les acteurs

### Le client du méta-système

Tout dispositif permettant d’adresser des requêtes aux services de l’interface du méta-système peut être vu comme un « client du méta-système ».

Le choix du protocole d’implémentation des interfaces est donc important par rapport aux contraintes que ce choix peut imposer. Autrement dit, le protocole choisi fixe le spectre des clients possibles du méta-système.

C’est le seul acteur du méta-système.

## Le périmètre fonctionnel du méta-système

Le périmètre fonctionnel du méta-système est couvert par 2 services :

* le service de recherche d’itinéraire distribué
* le service de recherche de localité

### Les frontières fonctionnelles du service de recherche d’itinéraire distribué

#### Objectif du service

Le but de ce service est de répondre à une recherche d’itinéraire entre 2 localités situées sur des SIM connectés différents.

Pour cela, le service identifie les SIM où se situent les localités de la recherche et calcule l’ensemble des traces possibles.

Le service retourne autant d’itinéraires composés qu’il existe de trace pour lesquelles une solution existe.

Pour une trace donnée, l’itinéraire composé est optimal:

* par rapport aux itinéraires proposés par les SIM de la trace
* par rapport aux transitions entre SIM de la trace

En s’appuyant strictement sur les itinéraires proposés par les SIM de la trace, l’itinéraire composé intègre les réglages de chaque SIM qui peuvent privilégier telle ou telle solution de déplacement sur leur territoire.

#### Durée d’exécution du service

Pour obtenir un itinéraire composé, le service interroge les SIM de la trace selon une même interface (l’interface générique de SIM). Pour autant, le temps de réponse des SIM varie :

* selon le nombre de transitions pris en compte par rapport à la trace,
* selon les performances de la solution technique mise en place derrière l’interface générique de SIM
* selon le niveau de sollicitation et la capacité de l’infrastructure derrière l’interface générique de SIM

Par conséquent, le temps de calcul nécessaire pour obtenir une solution sur 2 traces différentes n’est pas identique, même si les traces comptent le même nombre de SIM et à plus forte raison si le nombre de SIM est différent.

#### L’interactivité du service

Supposons qu’un client du méta-système interroge le service de recherche d’itinéraire entre 2 localités pour lesquelles il existe n traces.

Pour chacune des n traces où un itinéraire composé existe, le client du méta-système reçoit :

* une notification d’existence de solution dès celle-ci est avérée pour la trace considérée et des informations partielles sur cette solution
* la solution détaillée dès que celle-ci est obtenue

Le client du méta-système reçoit progressivement les solutions détaillées comme les notifications d’existence.

Entre 2 traces, les temps de calcul sont variables, le client du méta-système peut recevoir

* Les notifications d’existence d’abord (mais à des moments différents)
* Puis les solutions détailles (mais à des moments différents)

Ou bien la notification d’existence et la solution détaillée de la 1° trace puis la notification d’existence et la solution détaillée de la 2° trace.

#### Les différentes offres de transport prises en compte

Le service est conçu pour permettre d’intégrer les offres de transport des modes TC planifiés traditionnels (bus, car, métro, train, avion, TODO compléter) ainsi que des offres plus spécifiques

* TAD :
  + Ce type d’offre ne peut pas être demandé spécifiquement dans la requête. Néanmoins l’itinéraire composé intègre ce type d’offre dès lors que l’itinéraire optimal sur un SIM recoure au TAD.
  + Les informations concernant le TAD se réduisent à quelques informations de réservation.
* Autopartage
  + Comme pour le TAD, ce type d’offre ne peut pas être demandé spécifiquement dans la requête. Néanmoins l’itinéraire composé intègre ce type d’offre dès lors que l’itinéraire optimal sur un SIM recoure à l’autopartage.
  + Les informations concernant l’autopartage se réduisent à quelques informations comme l’url du site d’autopartage.
* Covoiturage
  + Comme pour le TAD, ce type d’offre ne peut pas être demandé spécifiquement dans la requête. Néanmoins l’itinéraire composé intègre ce type d’offre dès lors que l’itinéraire optimal sur un SIM recoure au covoiturage.
  + Les informations concernant le covoiturage se réduisent à quelques informations comme l’url du site de covoiturage.
* Vélopartage
  + Comme pour le TAD, ce type d’offre ne peut pas être demandé spécifiquement dans la requête. Néanmoins l’itinéraire composé intègre ce type d’offre dès lors que l’itinéraire optimal sur un SIM recoure au vélopartage.
  + Les informations concernant le vélopartage se réduisent à quelques informations comme l’url du site de vélopartage.

#### Le rabattement selon des modes hors Transport en Commun

Le service de recherche d’itinéraire permet également d’élargir la recherche aux solutions avec rabattement au début ou à la fin de l’itinéraire composé.

Pour cela, la requête au service précise le mode de rabattement (voiture, vélo) et si le rabattement est possible en début ou en fin de parcours.

La requête ne précise pas d’autre critère de rabattement tels que la présence de parking, la distance maximum, la vitesse de déplacement moyen dans le mode vélo par exemple.

Bien que le service d’itinéraire permette d’étendre la recherche au rabattement, l’itinéraire composé ne prend réellement en compte ce critère qu’à la condition où le premier (ou le dernier) SIM de la trace prend en compte ce critère.

#### Le bilan carbone

Le service renseigne un bilan carbone sur l’itinéraire composé à condition que tous les SIM de la trace associée aient renseignés leur bilan carbone également.

#### Le temps réel et calcul prédictif

Le service fournit un itinéraire composé à partir des réponses des SIM. L’itinéraire composé reporte donc les informations fournies par les SIM qui sont en mesure de gérer le temps réel ou bien le calcul prédictif.

La prise en compte du temps réel est réelle si le SIM qui l’intègre se trouve en début de trace. Dans le cas où un SIM est après le 1er SIM de la trace, les capacités temps-réel du SIM n’ont plus d’intérêt.

Le calcul prédictif peut présenter un intérêt plus général, même si le SIM qui s’appuie sur un calcul prédictif n’est pas le premier de la trace. Tout dépend de l’heure de départ choisie et la durée de parcours des SIM qui précèdent dans la trace.

#### L’accessibilité

Le service permet également de restreindre la recherche d’itinéraire en reportant ce critère dans les échanges avec les SIM.

Le critère d’accessibilité est binaire et représente un niveau d’accessibilité global PMR (donc un niveau exigeant d’accessibilité).

### Les frontières fonctionnelles du service de recherche de localité

### L’objectif du service

Le but de ce service est de fournir les localités qui peuvent servir de point de départ ou d’arrivée du service de recherche d’itinéraire distribué.

D’une part ce service de recherche doit s’utiliser sans connaître les SIM connectés et les territoires qu’ils recouvrent. La recherche doit pouvoir s’appuyer selon des critères généraux indépendants des SIM connectés.

D’autre part le service doit offrir une recherche de localité très générale qui puisse répondre à la diversité des significations que recouvre la notion de localité pour le grand public.

En effet une localité peut se décliner selon une typologie assez riche, les types de localité principaux sont les suivants

* Les adresses dans les territoires des SIM connectés
  + Au delà de ces territoires, le service de recherche d’itinéraire n’est plus en mesure de répondre
* Les arrêts des offres de transport des SIM connectés
  + A priori, le service peut adresser l’ensemble des arrêts quelque soit leur typologie définie sur le SIM connecté : potelet, quai, zone d’échange, arrêt commercial
* Les points d’intérêt localisés dans les territoires des SIM connectés

#### Périmètre fonctionnel minimal

Néanmoins le service n’impose pas de prendre en compte tous les types de localités possibles.

Pour fournir une recherche d’itinéraire distribuée de type « porte à porte », un service de recherche de localité limité aux adresses est suffisant.

En principe, il est nécessaire de disposer d’une base de données de voirie pour offrir un tel service.

Néanmoins, ce type de service étant d’un intérêt très général, il existe déjà un ensemble de services disponibles sous forme d’API WEB et selon différentes licences. Parmi les services les plus répandus sur le marché :

* https://developers.google.com/maps/documentation/geocoding/
* <http://www.postcodeanywhere.co.uk/>
* http://nominatim.openstreetmap.org/search
* <https://www.usps.com/business/web-tools-apis/address-information.htm>

#### La fonction de score

Le service de recherche de localité nécessite d’établir :

* un filtre des localités à partir des critères fournis en entrée
  + seules les localités pertinentes par rapport aux critères fournis au service sont retournées
* un classement parmi toutes les localités sélectionnées
  + les localités sont retournées par ordre décroissant de pertinence

La fonction de score est une fonction qui associe un nombre à chaque localité et ensemble de critères fournis. La fonction de score s’applique aux localités quelque soit leur typologie (potelet de bus, zone d’échange, adresses, POI).

La fonction de score évalue donc la pertinence d’une localité par rapport aux critères de recherche.

En général la fonction de score prend en compte toutes les propriétés d’une localité : son nom, tout le découpage administratif (commune, département, région, etc…).

La fonction de score peut également prendre en compte différentes langues.

En effet, selon la langue, les localités portent des noms différents : Londres, London, Naples, Napoli, etc…

La fonction de score peut également prendre en compte la pertinence phonétique.

## Les composants internes

L’adjectif interne signifie que les composants présentés dans ce chapitre sont du ressort de la maitrise d’œuvre qui aurait en charge la réalisation du méta-système.

### La base des métadonnées

La base des métadonnées est le composant qui stocke de manière persistante les données suivantes :

* les SIM connectés et les moyens d’interroger leur interface générique de SIM
* les arrêts des SIM connectés
* les transitions entre arrêts de SIM connectés différents

La base prend en compte le critère d’accessibilité dans la définition des transitions.

### Le système de synchronisation des métadonnées

C’est le composant qui assure le service de mise à jour des métadonnées par rapport à l’offre des SIM connectés.

Ce service est assuré de manière périodique en fonction du rythme de renouvellement des offres de transport sur les différents SIM connectés.

Le service intègre la planification de synchronisation des données.

Au niveau fonctionnel, l’hypothèse est faite que les SIM connectés conservent la même offre de transport durant la tranche horaire de disponibilité de l’interface générique des SIM.

Il en résulte que la fréquence la plus grande d’exécution du service de synchronisation est une fois par jour.

### L’aiguilleur

C’est le composant qui implémente les interfaces externes du méta-système.

Au niveau de la recherche d’itinéraire c’est ce composant

* qui calcule les traces de SIM à partir des métadonnées
* qui décompose la requête de recherche en une succession de requêtes élémentaires adressées aux SIM connectés. De ce point de vue, ce composant « distribue » le calcul en recherches « localisées » sur les territoires de SIM connectés.

## Les composants externes

L’adjectif externe signifie que les composants présentés dans ce chapitre ne sont pas du ressort de la maitrise d’œuvre qui aurait en charge la réalisation ou l’hébergement du méta-système.

En l’occurrence les composants externes se limitent aux SIM connectés. Or les SIM existent indépendamment du méta-système et font l’objet d’un marché spécifique entre une collectivité et un prestataire.

Néanmoins pour être accessibles du méta-système, les SIM connectés doivent implémenter une « interface générique » de SIM.

Cette interface est décrite au § 5.3 Interface générique de SIM.

## Les interfaces entre les composants

### La base des métadonnées

La base expose une interface de lecture/écriture aux seuls composants internes.

### Le système de synchronisation des métadonnées

Le composant utilise :

* l’interface de lecture/écriture de la base des méta-données
* l’interface générique des SIM exposée par l’ensemble des SIM connectés

Le composant est autonome vis à vis des autres composants internes.

Le composant n’expose pas d’interface externe non plus.

Même si l’implémentation de ce composant nécessite une interface avec son environnement de déploiement, cette interface pourra être adaptée librement en fonction du choix d’environnement et d’hébergement.

L’interface de ce composant n’entre donc pas dans le cadre de ce document.

### L’aiguilleur

L’aiguilleur utilise :

* La base des métadonnées
* L’interface générique de SIM des SIM connectés

L’aiguilleur publie l’interface des 2 services du méta-système :

* le service de recherche de localité
* le service de recherche d’itinéraire distribué

### Vue d’ensemble des interfaces

Le schéma ci-dessous présente les différents composants et leurs interfaces.

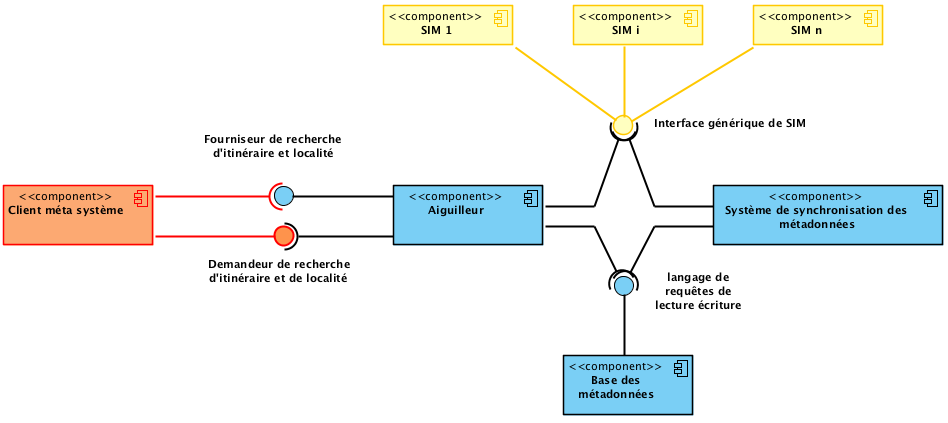


Figure Diagramme de composants

## Principaux cas d’utilisation

Il existe 2 principaux cas d’utilisation :

* L’un pour trouver une localité
* L’autre pour trouver un itinéraire

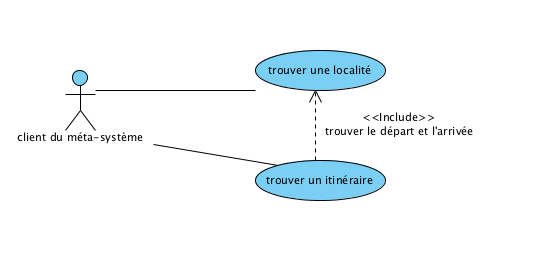


Figure Principaux cas d'usage

Les tableaux ci-dessous décrivent en détail les cas d’usage.

|  |  |
| --- | --- |
| Nom | « Trouver une localité » |
| Acteur | Le client du méta-système |
| Description | L’acteur interroge le méta-système en fournissant :   * une chaine de caractères * une zone géographique (optionnel) * un type de localité (optionnel) * une limite sur la quantité de réponses retournées (optionnel) |
| Scénario nominal | L’acteur interroge le méta-système en fournissant une chaine de caractère « lyon ».  Le méta-système retourne une liste de localités dont une partie des propriétés contient la chaine de caractères fournie.   * Les localités qui sont des adresses ont une partie des informations d’adresse qui contient la chaine fournie. Il peut s’agir de la voirie (rue de Lyon), de la commune (Lyon), du code postal. * Les localités qui sont des arrêts ont un nom qui contient la chaine fournie. * Les localités qui sont des POI ont un nom qui contient la chaine fournie. |
| Scénario alternatif | C’est le même scénario que le scénario nominal mais avec une chaine qui comporte une faute d’orthographe, par exemple « monparnasse ».  Le méta-système retourne une liste de localités dont une partie des propriétés contient est « voisine » la chaine de caractères proche.  Ce scénario permet de géré une approximation phonétique. |
| Scénario d’exception | C’est le même scénario que le scénario nominal seulement la chaine fournie correspond à trop grand nombre de localités possibles. Par exemple si la chaine fournie est « rue ».  Le méta-système retourne une liste vide. |
| Scénario d’exception | C’est le même scénario que le scénario nominal seulement la chaine fournie ne correspond à aucune localité connue. Par exemple si la chaine fournie est « zrtp ».  Le méta-système retourne une liste vide. |
| Post-conditions | La liste de proposition est triée en fonction du score.  La liste de proposition ne retourne que le type de localités demandées, tous les types si rien n’est précisé.  La taille de la liste est limitée en fonction du paramètre fourni |

|  |  |
| --- | --- |
| Nom | « Trouver une itinéraire » |
| Acteur | Le client du méta-système |
| Description | L’acteur interroge d’abord le méta-système pour sélectionner un point de départ et un point d’arrivée puis demande une recherche d’itinéraire au méta-système. |
| Scénario nominal | 1. L’acteur déroule le scénario nominal pour « trouver une localité » de départ et d’arrivée.  2. L’acteur interroge ensuite le méta-système en précisant :   * une position géographique de départ * une position géographique d’arrivée * un horaire de départ et une date * une sélection de modes de transport   3. Le méta-système identifie les SIM où se situent les points de départ et d’arrivée, soit les SIMO et SIMD.  4. Le méta-système identifie les traces t1 et t2 permettant de relier SIMO à SIMD.  5. Le méta-système établit l’existence d’un itinéraire composé I1 sur t1 et notifie cette existence à l’acteur en fournissant certaines informations sur l’itinéraire composé I1.  6. Le méta-système établit l’existence d’un itinéraire composé I2 sur t2 et notifie cette existence à l’acteur en fournissant certaines informations sur l’itinéraire composé I2.  7. Le méta-système obtient tout le détail de l’itinéraire composé I1 et notifie tout le détail de I1 à l’acteur.  8. Le méta-système obtient tout le détail de l’itinéraire composé I2 et notifie tout le détail de I2 à l’acteur.  9. Fin du scénario |
| Scénario alternatif 1 | Même scénario que le scénario nominal sauf aux étapes 6, 7 qui sont interverties.  6. Le méta-système obtient tout le détail de l’itinéraire composé I1 et notifie tout le détail de I1 à l’acteur.  7. Le méta-système établit l’existence d’un itinéraire composé I2 sur t2 et notifie cette existence à l’acteur en fournissant certaines informations sur l’itinéraire composé I2. |
| Scénario alternatif 2 | Ce scénario est une simplification du scénario nominal qui se produit lorsqu’une seule trace convient. Jusqu’à l’étape 3, ce scénario est identique au scénario nominal.  4. Le méta-système identifie la seule trace t1 qui relie SIMO à SIMD.  5. Le méta-système établit l’existence d’un itinéraire composé I1 sur t1 et notifie cette existence à l’acteur en fournissant certaines informations sur l’itinéraire composé I1.  7. Le méta-système obtient tout le détail de l’itinéraire composé I1 et notifie tout le détail de I1 à l’acteur.  9. Fin du scénario |
| Scénario alternatif 3 | Ce scénario est une généralisation du scénario nominal et du scénario alternatif 1 avec n traces, n > 2. |
| Scénario alternatif 4 | Même scénario que le scénario nominal jusqu’à l’étape 5.  6. L’acteur demande au méta-système l’interruption de la recherche d’itinéraire distribuée.  7. Fin du scénario. |
| Scénario d’exception 1 | Même scénario que le scénario nominal jusqu’à l’étape 1.  2. L’acteur renseigne une requête invalide en fournissant un mode de transport inconnu.  3. Le méta-système renvoie une réponse qui indique le paramètre en erreur.  4. Fin du scénario |
| Scénario d’exception 2 | Même scénario que le scénario nominal jusqu’à l’étape 2.  3. Le méta-système détermine que les positions de départ et d’arrivée sont en dehors des SIM connectés.  4. Le méta-système renvoie une réponse sans solution en précisant si le départ, l’arrivée ou les 2 sont en dehors des SIM connectés.  5. Fin du scénario |
| Scénario d’exception 3 | Même scénario que le scénario nominal jusqu’à l’étape 3.  3. Le méta-système détermine qu’il n’existe aucune trace de SIMO à SIMD  4. Le méta-système renvoie une réponse sans solution.  5. Fin du scénario |
| Scénario d’exception 4 | Même scénario que le scénario nominal jusqu’à l’étape 4.  C’est ce scénario qui se produit lorsque la base des métadonnées n’a pas été synchronisée depuis longtemps.  4. Le méta-système détermine qu’il n’existe de solution ni avec t1 ni avec t2.  5. Le méta-système renvoie une réponse sans solution.  6. Fin du scénario |
| Post-conditions | Les itinéraires composés sont optimaux relativement:   * à la trace considérée * aux réponses fournies par les SIM de la trace   L’optimisation garantit :   * soit d’arriver au plus tôt partant à partir de l’horaire demandé et en respectant les autres critères * soit de partir le plus tard possible pour arriver avant l’horaire d’arrivée demandé et en respectant les autres critères |

# Partie 2

Ce chapitre spécifie les interfaces attendues sur les composants du système (internes ou externes). La spécification précise l’ensemble des services de chaque interface.

Lorsque c’est nécessaire, les interfaces sont présentées par en 2 demi-interfaces réparties sur les composants mis en relation.

Le détail des données échangées au travers des interfaces fait l’objet du chapitre suivant.

## Interfaces DRI et FRI

Pour le service de recherche d’itinéraire distribuée, les exigences fonctionnelles (cf 4.3.1.3 L’interactivité du service) nécessitent une interface permettant des notifications asynchrones, que ce soit pour transmettre :

* les itinéraires composés aussitôt que chacun d’eux est connu
* les notifications d’existence d’itinéraire composé aussitôt que chacune d’elles est établie

Les requêtes de recherche d’itinéraire, comme les informations reçues en retour (existence d’itinéraire composé ou détail d’itinéraire composé), correspondent à des événements pour lesquels l’aiguilleur et le client du méta-système sont à la fois émetteurs et récepteurs.

Nous distinguerons les 2 demi interfaces suivantes

* demi interface demandeur de recherche d’itinéraire et de localité
  + Le système qui implémente la demi interface demandeur de recherche d’itinéraire et de localité est noté DRI
* demi interface fournisseur de recherche d’itinéraire et de localité
  + Le système qui implémente la demi interface fournisseur de recherche d’itinéraire et de localité est noté FRI

Le système client de l’interface FRI doit implémenter l’interface DRI.

Le système client de l’interface DRI doit implémenter l’interface FRI.

Les services de la demi interface FRI :

* Service de demande de recherche d’itinéraire
* Service d’annulation de recherche d’itinéraire
* Service de demande de recherche de localité
* Service d’annulation de recherche de localité

Les services de la demi interface DRI

* Service de réception de l’existence d’une solution relative à une demande de recherche d’itinéraire
* Service de réception d’une solution détaillée relative à une demande de recherche d’itinéraire
* Service de réception de la fin de traitement d’une demande de recherche d’itinéraire
* Service de réception des localités qui satisfont le mieux les critères de la recherche de localité

Les composants client du méta-système et aiguilleur implémentent chacune demi-interfaces.

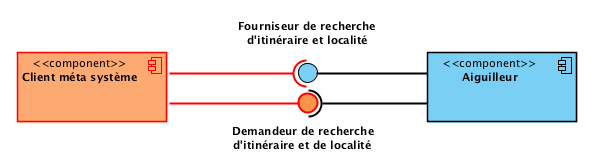


Figure Diagramme de composants interface FRI DRI

Plusieurs solutions existent pour implémenter ce type d’interface. Les questions d’implémentation ne sont pas dans le périmètre de ce document, mais nous pouvons citer à titre d’exemple les protocoles HTTP 1.1 ou WebSocket.

### Structures de données échangées

La question du format d’implémentation des structures de données échangées par ces interfaces est en dehors du périmètre de ce document.

Autrement dit, il est possible de mettre en œuvre les interfaces entre l’aiguilleur et le client du méta-système en utilisant différents format : XML, JSON, CSV, …

Par commodité, le document fait l’hypothèse d’une implémentation en XML afin de pouvoir utiliser le langage XSD pour décrire les propriétés des structures échangées.

Ces mêmes propriétés peuvent néanmoins être appliquées à d’autres formats d’implémentation : JSON, CSV, …

L’accès au détail des structures d’échange est présenté au chapitre 6.2.1 Les structures d’échange des interfaces du méta-système (FRI, DRI).

### Demi interface fournisseur de recherche d’itinéraire et de localité (FRI)

L’aiguilleur implémente l’interface FRI.

#### Service de demande de recherche d’itinéraire

|  |  |
| --- | --- |
| Les entrées | Une structure PlanTripRequest qui décrit les critères de la recherche d’itinéraire.  La structure propose un seul point de départ Departure et un seul point d’arrivée. Ces poinets sont aussi bien des identifiant d’arrêt ou de POI que des positions géographiques.  L’élément MaxTrips indique le nombre maximum de solutions d’itinéraires attendus.  La date et l’heure de départ sont fixés au choix :   * avec l’élément DepartureTime qui signifie une demande pour partir à cet horaire * avec l’élément ArrivalTime qui signifie une demande pour arriver à cet horaire.   L’élément Algorithm contient une valeur énumérée. La fonction de recherche d’itinéraire s’exécute en optimisant un critère propre à chaque valeur énumérée :   * CLASSIC : dans le cas d’une recherche en « départ à », c’est l’horaire d’arrivée au plus tôt qui recherché d’abord avec un départ au plus tard ensuite. Dans le cas d’une recherche en « arrivée à », c’est l’horaire de départ au plus tard qui recherché d’abord avec une arrivée au plus tôt ensuite. * SHORTEST : correspond au minimum de correspondance * FASTEST : correspond à la durée d’itinéraire minimum * MINCHANGES : correspond à la durée minimum de correspondance   Contrairement à l’algorithme de type CLASSIC est ici défini de manière rigoureuse, les autres types d’algorithme nécessitent d’être précisés. Imaginons, une solution notée S ayant une correspondance de moins que celle proposée par l’algorithme CLASSIC notée C. Dans qu’elle mesure la solution S est elle acceptable, si celle-ci part et arrive beaucoup plus tard que la solution S ?  La séquence d’éléments modes liste l’ensemble des modes de transport qui sont autorisés pour la recherche d’itinéraire.  La séquence d’éléments selfDriveConditions représente les possibilités de rabattement qui permettent d’élargir la recherche d’itinéraire à d’autres modes au niveau du début de l’itinéraire ou de la fin de l’itinéraire. Une possibilité de rabattement indique si elle porte sur le début ou la fin de l’itinéraire et les modes de transport (pour lesquels le voyageur se dirige seul) autorisés.  Si cette séquence est vide, les solutions d’itinéraire ne peuvent emprunter que la liste des modes de transport qui figurent dans l’élément modes.  L’élément AccessibilityConstraint est un simple indicateur réduire la recherche aux seuls itinéraires accessibles pour tout type de PMR.  L’élément Language fournit la langue dans laquelle les élements textuels de la réponses sont attendus. La réponse retournées exprime les champs texuels dans cette langue ou la langue par défaut des SIM le cas échéant. |
| Les pré-conditions | Aucune |
| Les sorties | Une structure PlanTripResponse qui   * fournit un identifiant de demande de recherche d’itinéraire avec l’élément RequestId * indique si les données d’entrées sont valides (avec l’élément Status et le code associé) * précise, le cas échéant avec la séquence errors, quelles sont les propriétés invalides dans les données d’entrée |
| Les post-conditions | Si la structure PlanTripRequest est valide, alors le FRI lance un calcul réparti.  Les données de sortie définissent un identifiant de demande de recherche d’itinéraire.  Dans toutes les notifications que le FRI adresse ensuite au DRI, c’est ce même identifiant de demande qui est fourni.  Parmi les différentes conditions d’invalidité des propriétés, les différents cas suivants sont pris en compte :   * la date de la recherche d’itinéraire est en dehors de la période calendaire couverte par le FRI * le départ (ou l’arrivée) est trop éloigné du territoire couvert par le FRI * le départ (ou l’arrivée) référence un arrêt ou un POI inconnu * le départ et l’arrivée référencent un même arrêt, un même POI ou une même * adresse * un ou plusieurs modes de transport sont inconnus ou non gérés par le FRI * un ou plusieurs modes de transport de rabattement sont inconnus ou non gérés par le FRI |

#### Service d’annulation de recherche d’itinéraire

|  |  |
| --- | --- |
| Les entrées | Une structure PlanTripCancellationRequest qui référence une demande de recherche d’itinéraire |
| Les pré-conditions | La structure doit référencer une demande de recherche d’itinéraire par son identifiant contenu dans l’élément RequestId. |
| Les sorties | Une structure d’acquittement PlanTripCancellationResponse qui précise le cas échéant si le calcul d’itinéraire a pu être annulé. |
| Les post-conditions | Si le calcul d’itinéraire a pu être annulé, le FRI n’envoie plus aucune notification relative à la demande de recherche d’itinéraire. |

#### Service de demande de recherche de localité

|  |  |
| --- | --- |
| Les entrées | Une structure SearchPointsRequest qui décrit les critères de la recherche d’itinéraire. |
| Les pré-conditions | Aucune |
| Les sorties | Une structure SearchPointResponse qui   * fournit un identifiant de demande de recherche de localité avec l’élément RequestId * indique si les données d’entrées sont valides * précise le cas échéant quelles sont les propriétés invalides dans les données fournies. |
| Les post-conditions | Si la structure SearchPointsRequest est valide, alors le FRI lance un traitement de recherche.  Les données de sortie définissent un identifiant de demande de recherche de localité.  Dans toutes les notifications que le FRI adresse ensuite au DRI, c’est ce même identifiant de demande qui est fourni.  Parmi les différentes conditions d’invalidité des propriétés, les différents cas suivants sont pris en compte :   * La zone géographique demandée n’est pas incluse dans le territoire couvert par le FRI |

#### Service d’annulation de recherche de localité

|  |  |
| --- | --- |
| Les entrées | Une structure SearchPointsCancellationRequest qui référence une demande de recherche de localité avec l’élément RequestId. |
| Les pré-conditions | La structure doit référencer une demande de recherche de localité par son identifiant. |
| Les sorties | Une structure d’acquittement SearchPointsCancellationResponse qui précise le cas échéant si la recherche a pu être annulée. |
| Les post-conditions | Si la recherche de localité a pu être annulée, le FRI n’envoie plus aucune notification relative à la demande de localité. |

### Demi interface demandeur de recherche d’itinéraire et de localité (DRI)

Le client du méta-système implémente l’interface DRI.

#### Service de réception de l’existence d’une solution relative à une demande de recherche d’itinéraire

|  |  |
| --- | --- |
| Les entrées | Une structure PlanTripExistenceNotificationResponse qui précise :   * l’identifiant de demande relative à cette notification * un identifiant de solution d’itinéraire composé * un horaire d’arrivée si la demande est basée sur un critère en « départ à » ou un horaire de départ si la demande est basée sur un critère en « partir à ». |
| Les pré-conditions | Aucune |
| Les sorties | Aucune |
| Les post-conditions | Aucune |

#### Service de réception d’une solution détaillée relative à une demande de recherche d’itinéraire

|  |  |
| --- | --- |
| Les entrées | Une structure PlanTripNotificationResponse qui précise :   * l’identifiant de demande relative à cette notification * un identifiant de solution d’itinéraire composé * tout le détail de l’itinéraire composé |
| Les pré-conditions | Aucune |
| Les sorties | Aucune |
| Les post-conditions | Aucune |

#### Service de réception de la fin de traitement d’une demande de recherche d’itinéraire

|  |  |
| --- | --- |
| Les entrées | Une structure EndingSearch qui précise :   * l’identifiant de demande relative à cette notification |
| Les pré-conditions | Aucune |
| Les sorties | Aucune |
| Les post-conditions | Aucune |

#### Service de réception d’une liste de localités

|  |  |
| --- | --- |
| Les entrées | Une structure SearchPointsNotificationResponse qui précise :   * un identifiant de demande de recherche de localité * une liste de localités qui satisfont les critères géographiques, de type de localité et de taille maximum de la liste   Les localités sont décrites en détail et leurs propriétés obtiennent les meilleurs scores possibles. |
| Les pré-conditions | Aucune |
| Les sorties | Aucune |
| Les post-conditions | Aucune |

### Comportement dynamique de la demi-interface FRI

Outre les éléments de spécification qui précèdent concernant la demi-interface FRI, celle-ci doit respecter des contraintes au niveau de son comportement dynamique vis à vis de la demi-interface DRI.

Le comportement dynamique de la demi-interface FRI se définit à partir d’un diagramme d’état.

#### Diagramme relatif à une recherche d’itinéraire

Le diagramme déroule toutes les étapes que suit le FRI pour traiter une demande de recherche d’itinéraire.

##### Règles de notation

Le diagramme définit des états simples et des états complexes.

Les états complexes représentent une sous machine à états finis elle même composée d’états simples ou complexes. Les états simples sont associés à un numéro, et les états complexes à une lettre.

Les transitions qui représentent un appel à un service du FRI sont accompagnées d’un label écrit en rouge.

Les transitions qui représentent un appel à un service du DRI sont accompagnées d’un label écrit en bleu.

|  |  |
| --- | --- |
| Réception PlanTripRequest | désigne un appel du service de demande de recherche d’itinéraire du FRI |
| Envoi PlanTripResponse | désigne la réponse du service de demande de recherche d’itinéraire du FRI |
| Réception PlanTripCancellationRequest | désigne un appel au service demande d’annulation de recherche d’itinéraire du FRI |
| Envoi PlanTripCancellationResponse | désigne une réponse du service demande d’annulation de recherche d’itinéraire du FRI |
| Envoi PlanTripExistenceNotificationResponse | désigne un appel du FRI au service DRI de réception de l’existence d’une solution relative à une demande de recherche d’itinéraire |
| Envoi PlanTripNotificationResponse | désigne un appel du FRI au service DRI de réception d’une solution relative à une demande de recherche d’itinéraire |
| Envoi EndingSearch | désigne un appel du FRI au service DRI de réception de fin de traitement d’une demande de recherche d’itinéraire |

Dans les diagrammes, « n » représente le nombre de traces que le FRI a déterminé par rapport à la demande de recherche d’itinéraire.

##### Le diagramme d’état général du FRI

Le diagramme d’état s’applique au FRI.

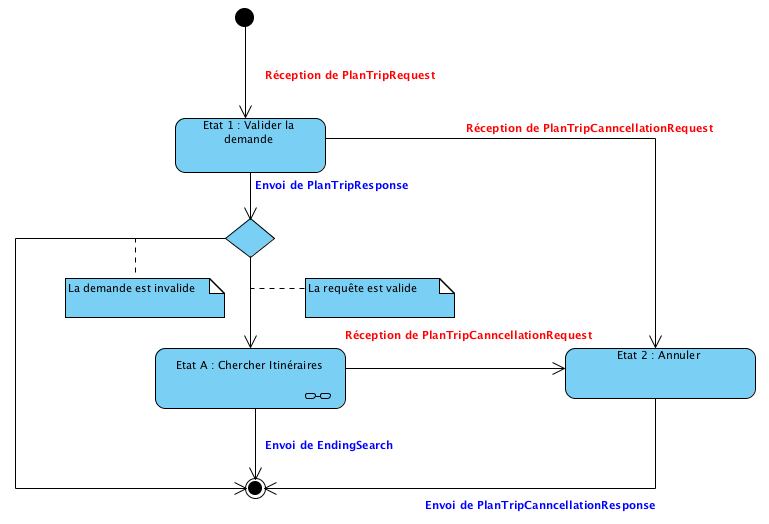


Figure Diagramme d'état général de recherche d'itinéraire

Les différents états du diagramme sont énoncés dans le tableau ci-dessous

|  |  |  |
| --- | --- | --- |
| Etat | Condition d’entrée dans l’état | Condition de sortie de l’état |
| Etat 1 : Valider la demande | Le FRI reçoit une demande de recherche d’itinéraire | Soit le FRI a terminé de valider la demande,  Soit le FRI reçoit une demande d’annulation |
| Etat 2 : Annuler | Le FRI reçoit une demande d’annulation | Le FRI a annulé la demande |
| Etat A : Rechercher itinéraire | Le FRI a vérifié que la demande est valide | Soit le FRI a terminé la recherche d’itinéraire  Soit le FRI reçoit une demande d’annulation |

Ci-dessous sont listées les contraintes qui s’appliquent aux données échangées à travers les interfaces du FRI ou du DRI.

Le diagramme ne prend en compte que les demandes d’annulation qui référence l’identifiant de la demande de recherche d’itinéraire en cours, i.e. l’identifiant fourni par le FRI en retour de la demande de recherche d’itinéraire. Les autres demandes d’annulation n’ont aucun effet. [C1]

Lorsque le FRI appelle le service de réception de la fin de traitement d’une recherche d’itinéraire du DRI, le FRI fournit le même identifiant de requête que celui fourni en retour de la demande de recherche d’itinéraire. [C2]

##### Le sous-diagramme d’état propre à l’état A

Ce diagramme dépend du nombre de traces que le FRI a déterminé en fonction des critères de la demande de recherche d’itinéraire.

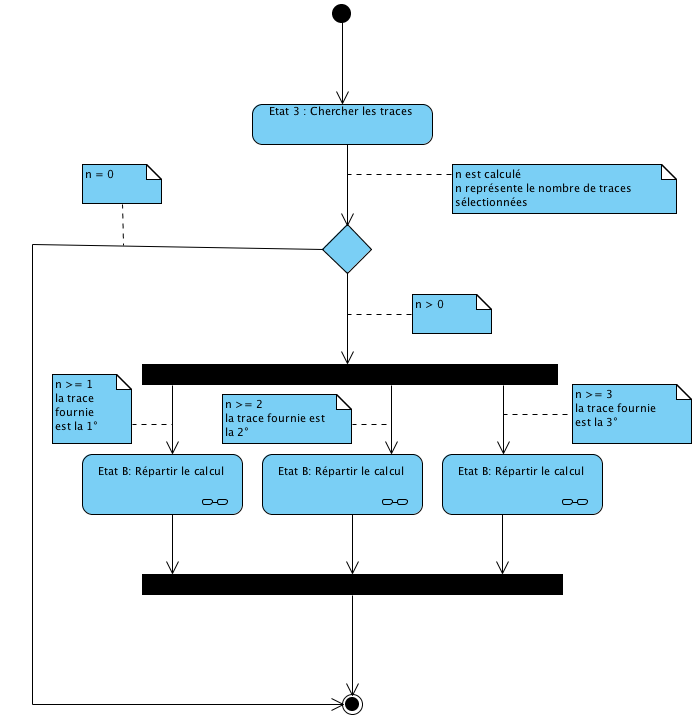


Figure Sous-diagramme propre à l'état A

Chaque trace conduit à un traitement asynchrone de recherche d’itinéraire composé.

Le diagramme se termine :

* lorsque le nombre de trace est nul
* ou lorsque tous les traitements asynchrones sont terminés

Les différents états du diagramme sont énoncés dans le tableau ci-dessous

|  |  |  |
| --- | --- | --- |
| Etat | Condition d’entrée dans l’état | Condition de sortie de l’état |
| Etat 3 : Rechercher les traces | Le FRI a validé la demande de recherche d’itinéraire | Le nombre n de traces possibles est calculé, avec  n qui représente le nombre de traces de SIM possibles |
| Etat B : Répartir le calcul | Le FRI a vérifié que la demande est valide | Le FRI a terminé la recherche d’itinéraire |

##### Le sous-diagramme d’état propre à l’état B

Ce diagramme est relatif à un traitement asynchrone de recherche d’itinéraire composé pour une trace donnée.

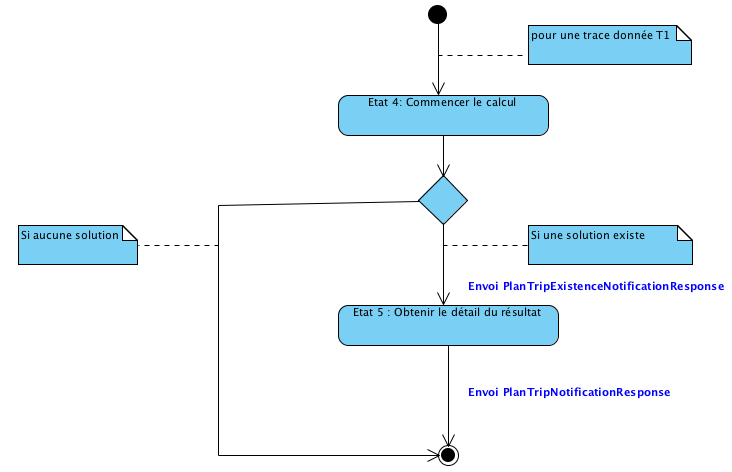


Figure Sous-diagramme propre à l'état B

Les différents états du diagramme sont énoncés dans le tableau ci-dessous

|  |  |  |
| --- | --- | --- |
| Etat | Condition d’entrée dans l’état | Condition de sortie de l’état |
| Etat 4: Commencer le calcul | Le traitement asynchrone est démarré par le FRI à partir d’une trace donnée | Soit le traitement asynchrone a déterminé l’existence d’une solution d’itinéraire composé  Soit le traitement asynchrone a détermine qu’il n’existe pas de solution |
| Etat 5 : Obtenir le détail du résultat | Le traitement asynchrone a déterminé l’existence d’une solution d’itinéraire composé | Le traitement asynchrone a déterminé tout le détail d’un itinéraire composé |

Ci-dessous sont listées les contraintes qui s’appliquent aux données échangées à travers les interfaces du FRI ou du DRI.

Lorsque le FRI appelle le service DRI de réception de l’existence d’une solution relative à une recherche d’itinéraire, le FRI fournit :

* un même identifiant de demande que celui en retour de la demande de recherche d’itinéraire [C3]
* un identifiant de solution différent des autres identifiants de solutions qui sont issues de cette demande à partir des autres traces [C4]
* la solution dont il est question correspond à l’itinéraire composé optimal par rapport à la trace et aux critères de la demande [C5]

Lorsque le FRI appelle le service DRI de réception d’une solution relative à une recherche d’itinéraire, le FRI fournit le détail de l’itinéraire composé dont l’existence a été notifiée préalablement, le FRI fournit donc :

* le même identifiant de demande que pour la notification d’existence [C6]
* le même identifiant de solution que pour la notification d’existence [C7]

#### Diagramme relatif à une recherche de localité

Le diagramme déroule toutes les étapes que suit le FRI pour traiter une demande de recherche de localité.

##### Règle de notation

Ce sont les mêmes que pour le diagramme de recherche d’itinéraire.

Le diagramme quelles sont les transitions qui sont associées à des appels aux interfaces DRI et FRI.

|  |  |
| --- | --- |
| Réception SearchPointsRequest | désigne un appel du service de demande de recherche de localité du FRI |
| Envoi SearchPointsResponse | désigne la réponse du service de demande de recherche de localité du FRI |
| Réception SearchPointsCancellationRequest | désigne un appel au service demande d’annulation de recherche de localité du FRI |
| Envoi SearchPointsCancellationResponse | désigne une réponse du service demande d’annulation de recherche de localité du FRI |

##### Le diagramme d’état du FRI

Ce diagramme d’état s’applique au FRI.

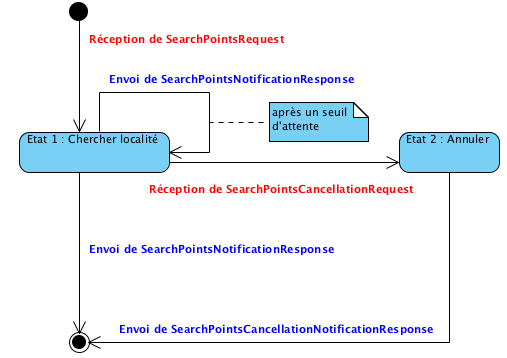


Figure Diagramme d’état de recherche de localités

Les différents états du diagramme sont énoncés dans le tableau ci-dessous

|  |  |  |
| --- | --- | --- |
| Etat | Condition d’entrée dans l’état | Condition de sortie de l’état |
| Etat 1 : Chercher localité | Le FRI reçoit une demande de recherche d’itinéraire  Ou après une temporisation | Soit le FRI renvoie les meilleures localités après un délai maximum de calcul,  Soit le FRI a eu le temps d’exécuter tout le traitement de cherche  Soit le FRI a reçu une demande d’annulation |
| Etat 2 : Annuler | Le FRI reçoit une demande d’annulation | Le FRI a annulé la demande |

Ci-dessous sont listées les contraintes qui s’appliquent aux données échangées à travers les interfaces du FRI ou du DRI.

[C8] Le diagramme ne prend en compte que les demandes d’annulation qui référence l’identifiant de la demande de recherche de localité en cours, i.e. l’identifiant fourni par le FRI en retour de la demande de recherche de localité. Les autres demandes d’annulation n’ont aucun effet.

Lorsque le FRI appelle le service DRI de réception d’une liste de localités, le FRI fournit :

* un même identifiant de demande que celui en retour de la demande de recherche de localité [C9]
* la liste de localité dont il est question correspond aux localités ayant obtenues les meilleures score durant le temps d’exécution de la recherche [C10]

### Comportement dynamique de la demi-interface DRI

Aucune exigence ce porte sur le comportement de cette demi interface.

### Conditions de fonctionnement

Pour que le FRI (l’aiguilleur) produise des résultats pertinents au DRI (le client du méta-système), il faut que la base des métadonnées soit à jour des offres de transport des SIM connectés.

Comme les offres de transport sur le SIM évoluent dans le temps, les services de l’interface générique de SIM renvoient des réponses qui changent aussi dans le temps.

Par exemple certains arrêts disparaissent et d’autres sont créés.

Par conséquent, il faut que la base des métadonnées soit à jour des changements qui peuvent survenir sur les réponses fournies par l’interface générique de SIM (au niveau des services de collecte d’arrêts et de capacités du SIM) de chacun des SIM connectés.

## Interface langage de requêtes sur les métadonnées

L’interface qu’expose la base de données fournit un langage de requêtes de lecture et écriture des métadonnées.

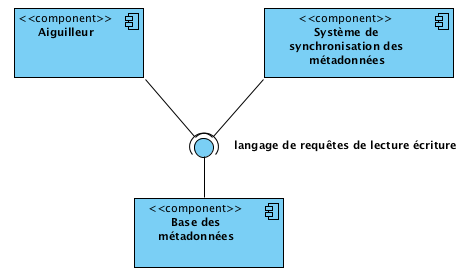


Figure Diagramme de composants : interface langage de requêtes

Au niveau de l’implémentation, il existe une grande variété de base de données.

Par commodité nous supposons, dans la suite du document, que nous utilisons une base relationnelle dont l’interface d’accès définie par :

* le langage SQL
* le schéma de la base

L’aiguilleur comme le système de synchronisation ont besoin d’accéder aux métadonnées :

* l’aiguilleur lit les métadonnées pour traiter les requêtes de recherche d’itinéraire ou de localités (pour ce qui concerne les localités de type arrêt)
* le système de synchronisation lit les métadonnées et les modifie

### Le schéma de la base

Le schéma contient les seules données nécessaires et suffisantes pour réaliser l’ensemble de la solution de manière opérationnelle.

S’agissant du service de recherche de localité, le schéma présenté ici ne permet de traiter que des localités de type « arrêt ». Le schéma doit être étendu pour traiter des localités de type POI ou adresse.

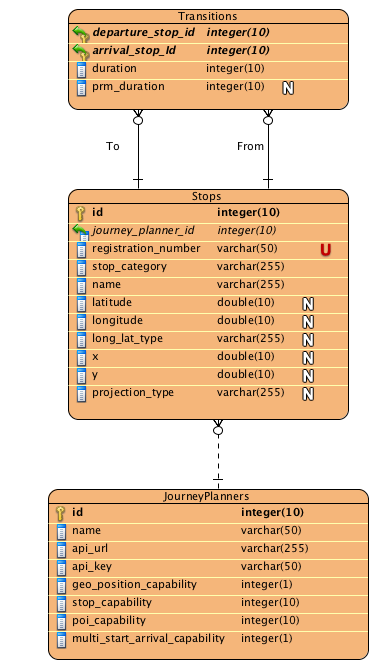


Figure Schéma de la base des métadonnées

Sur cette figure, la contrainte d’unicité correspond au symbole U rouge sur la colonne concernée.

Les colonnes qui peuvent éventuellement contenir des champs non définis sont repérés par le symbole N blanc.

Le schéma définit 3 tables :

* JourneyPlanners
* Stops
* Transitions

|  |  |  |
| --- | --- | --- |
| JourneyPlanners  La table liste tous les SIM connectés au méta-système et fournit pour chacun toutes les informations nécessaires à l’utilisation de leur interface générique de SIM.  Cette table est doit être initialisée avant la mise en service du système et avant ajout d’un nouveau SIM à connecter (cf 5.2.3 Contraintes de mise en fonctionnement).  Le composant de synchronisation des métadonnées est chargé de la mise à jour de certaines propriétés des SIM, ces propriétés proviennent de l’interface générique de SIM. | | |
| Nom de colonne | Contenu | Correspondance avec la réponse du service de « collecte des arrêts » de l’interface générique de SIM |
| id | Identifiant du SIM | Aucune |
| name | Nom du SIM | Aucune |
| api\_url | URL d’accès à l’interface générique du SIM | Aucune |
| api\_key | Clé d’accès à l’interface générique du SIM | Aucune |
| geo\_position\_capablility | Capacité du SIM à traiter des localités de type adresse au niveau de l’interface générique de SIM | Elément GeographicPositionCompliant |
| poi\_capablility | Capacité du SIM à traiter des localités de type POI au niveau de l’interface générique de SIM | Aucune |
| stop\_capablility | Capacité du SIM à traiter des localités de type arrêt au niveau de l’interface générique de SIM | Aucune |
| multi\_start\_arrival\_capability | Quantité maximum de départ ou d’arriver que le SIM sait gérer dans l’interface générique de SIM | élément MultipleStartAndArrivals |

|  |  |  |
| --- | --- | --- |
| Stops  La table liste tous les arrêts des SIM connectés au méta-système.  C’est le composant de synchronisation des métadonnées qui est chargé de remplir cette table.  Excepté la clé primaire (id) et la clé étrangère (journey\_planner\_id), toutes les données de cette table sont toutes issues de l’interface générique de SIM, et plus précisément de la structure de sortie du service de collecte d’arrêt (cf § 5.3.4.2. La structure de donnée en sortie). | | |
| Nom de colonne | Contenu | Correspondance avec la réponse du service de « capacités du SIM » de l’interface générique de SIM |
| id | Identifiant de l’arrêt | Aucune |
| journey\_planner\_id | Référence le SIM auquel appartient l’arrêt | Aucune |
| registration\_number | Identifiant de l’arrêt relatif au SIM dont il est issu | élément PrivateCode de Quay |
| stop\_category | type d’arrêt par rapport à la classification proposée par le profil Netex utilisé dans l’interface générique de SIM | élément QuayType de Quay |
| name | Nom de l’arrêt | élément Name de Quay |
| latitude | Latitude relative au référentiel géodésique | élément Latitude de Quay / Centroid / Location / Latitude |
| longitude | Longitude relative au référentiel géodésique | élément Longitude de Quay / Centroid / Location / Latitude |
| long\_lat\_type | Référentiel géodésique | élément DefaultLocationSystem |

|  |  |
| --- | --- |
| Transitions :  La table liste toutes les transitions qui relient les SIM connectés.  C’est le composant de synchronisation des métadonnées qui est chargé de remplir ou mettre à jour cette table. | |
| Nom de colonne | Contenu |
| departure\_stop\_id | Référence l’arrêt au départ de la transition |
| arrival\_stop\_Id | Référence l’arrêt à l’arrivée de la transition |
| duration | durée de parcours de la transition pour un voyageur ordinaire |
| pmr\_duration | durée de parcours de la transition pour une PMR |

### Contrainte d’optimisation d’accès aux arrêts

Les systèmes clients de cette interface (aussi bien l’aiguilleur que le système de synchronisation) peuvent être soumis à une contrainte liée à l’optimisation des accès aux arrêts dans les méta-données.

Tout dépend du choix d’implémentation sur le client de l’interface.

Selon l’environnement technique du client de l’interface (librairie géographique) et de l’espace mémoire disponible (qui dépend aussi de la quantité d’arrêts dans les méta-données), le client peut :

* gérer lui-même la question de la sélection des arrêts de SIM par rayonnement autour d’une position géographique.
* déléguer le rayonnement au système de méta-données et dans ce cas, la base des méta-données doit être pourvue d’une extension géographique.

L’extension géographique dépend du système de base de données choisi. Mais la plupart des systèmes de base de données disposent de ce genre d’extension.

### Contraintes de mise en fonctionnement

Pour être mise en service, la base des métadonnées doit être initialisée, l’initialisation consiste

* à renseigner la table « JourneyPlanners » en y reportant les propriétés des SIM destinés à être connectés
* puis à exécuter la tache de synchronisation des métadonnées sur ce composant. A l’issue de cette tache, les tables Stops et Transitions sont renseignées

De même pour ajouter un nouveau SIM connecté, il est nécessaire d’ajouter

* un nouvel enregistrement dans la table JourneyPlanners de la base des métadonnées
* d’exécuter la tache de synchronisation des métadonnées sur le composant de synchronisation

## Interface générique de SIM

L’interface générique de SIM fournit tous les services qui permettent :

* de traiter une recherche d’itinéraire distribuée entre plusieurs SIM connectés
* de mettre à jour la base des métadonnées

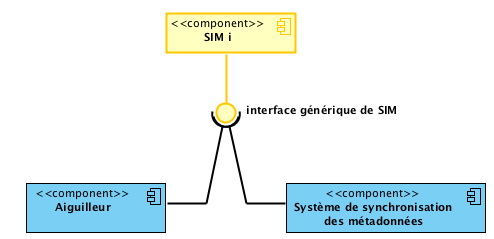


Figure Diagramme de composants: interface générique de SIM

Tous les SIM connectés implémentent cette interface.

Les composants clients de cette interface sont :

* l’aiguilleur
* le système de synchronisation des métadonnées

### Descriptions générales des services de l’interface

#### Structure des données échangées

Comme pour les interfaces DRI et FRI (cf § 5.1.1), plusieurs formats sont possibles pour structurer les échanges.

Le document fait l’hypothèse d’une implémentation en XML pour les mêmes motifs. L’accès au détail des structures d’échange est présenté au chapitre 6.2.2 Les structures d’échange des interfaces génériques de SIM.

#### Les structures en entrée des services

Les structures en entrée sont particulières à chacun des services de l’interface générique des SIM.

Néanmoins dans un souci de cohérence avec l’interface FRI, un groupe d’éléments ItineraryRequestParametersGroup a été défini afin que les données en entrée des services réalisant une recherche d’itinéraire contiennent ce même groupe d’éléments.

Le tableau ci-dessous présente la liste des services réalisant une recherche d’itinéraire:

|  |  |  |
| --- | --- | --- |
| Interface | Service | Structure de donnée en entrée du service |
| FRI | Service de demande de recherche d’itinéraire | PlanTripRequest |
| Interface générique de SIM | Service de recherche d’itinéraire non détaillée n-m | ItineraryRequest |
| Interface générique de SIM | Service de recherche d’itinéraire détaillée 1-n | SumedUpItineraryRequest |

L’intérêt de mutualiser ainsi un groupe d’éléments, c’est de pouvoir véhiculer de manière uniforme les critères de recherches que ce soit

* au niveau des entrées aux services concernés du FRI
* ou au niveau des entrées aux services concernés de l’interface générique des SIM.

#### Les structures en sortie des services et exception

Quelque soit le service, la réponse dispose d’informations sur l’exécution du service.

Ces informations sont rassemblées dans la structure Status définie dans le fichier MisProtocol.xsd.

La structure Status contient en particulier l’élément Code qui est une valeur énumérée issue d’une énumération, cette énumération codifie tous les cas d’exception des différents services.

Le tableau ci-dessous donne la liste de ces codes en précisant à quels services ceux-ci se rapportent.

|  |  |  |
| --- | --- | --- |
| Valeur énumérée | Définition | Services de l’interface générique de SIM susceptibles de produire le code |
| OK | succès de l’exécution du service | Tous les services |
| TOO\_FAR\_POSITION | La structure de donnée en entrée référence une localité qui est trop éloignée du périmètre géographique du SIM | Les 2 services réalisant une recherche d’itinéraire |
| UNKNOWN\_END\_POINT | La structure de donnée en entrée référence une localité (de type arrêt ou POI) ayant un identifiant inconnu | Les 2 services réalisant une recherche d’itinéraire |
| DATE\_OUT\_OF\_SCOPE | La structure de donnée en entrée référence une date de la recherche est trop lointaine | Les 2 services réalisant une recherche d’itinéraire |
| TOO\_MANY\_END\_POINT | La structure de donnée en entrée référence un nombre de localités trop important par rapport aux capacités du SIM. | Les 2 services réalisant une recherche d’itinéraire |
| BAD\_REQUEST | La structure en entrée ne respecte pas le format attendu | Les 2 services réalisant une recherche d’itinéraire |
| INTERNAL\_ERROR | L’exécution du service a échoué pour une raison inconnue | Tous les services |

### Service de recherche d’itinéraire détaillée 1-n

Le service calcule l’itinéraire optimal par aller d’un point de départ vers n points de destination (ou de n points de départ vers une destination) et en fonction des autres critères fournis.

L’itinéraire optimal retourné est complètement détaillé.

Les structures échangées dans ce service sont définies dans le fichier XSD MisPlanTrip.

#### La structure de donnée en entrée

La structure de donnée en entrée correspond à l’élément SumedUpItinerariesRequest.

L’élément SumedUpItinerariesRequest propose plusieurs points de départ (éléments Departure de la séquence departures) et plusieurs points d’arrivée (éléments Arrivals de la séquence arrivals).

Les structures Departure et Arrival modélisent un emplacement géographique (qui peut être un arrêt aussi bien qu’une position géographique d’adresse) assorti d’une durée d’accès. La durée d’accès représente le temps nécessaire au voyageur pour se rendre à pied à cet emplacement.

Si l’élément SumedUpItinerariesRequest définit une séquence « options » non vide, alors l’une des 2 séquences « departures » ou « arrivals » ne contient qu’un seul élément.

Sans cela, le code d’exception BAD\_REQUEST est retourné dans l’élément Status de la réponse.

L’élément Algorithm est défini de la même manière que sur l’interface FRI, cf § 5.1.2.1 Service de demande de recherche d’itinéraire.

Ce n’est pas le but de ce document de spécifier en détail ces différents algorithmes. Dans la suite du document, l’optimisation du calcul sera envisagé seulement avec l’algorithme CLASSIC.

La structure de donnée en entrée peut contenir également des paramètres optionnels.

Lorsque ces paramètres ne sont pas renseignés, le service applique un fonctionnement par défaut présenté ci-dessous.

|  |  |
| --- | --- |
| Paramètre optionnel | Prise en compte en cas d’absence dans la structure de donnée en entrée |
| Algorithm | L’optimisation est faite comme si le paramètre était fixé à la valeur énumérée CLASSIC. |
| modes | La recherche est faite comme si l’ensemble des modes avaient été sélectionnés. |
| selfDriveConditions | Le rabattement n’est pas autorisé. Autrement dit, si le mode vélo (ou voiture) ne figure pas dans le paramètre « modes », alors aucun Leg ne peut être d’un mode vélo (ou voiture). |
| AccessibilityContraint | La recherche d’itinéraire ne tient pas compte de l’accessibilité. C’est équivalent à un paramètre AccessibilityContraint égal à « false ». |
| Language | Les informations textuelles des itinéraires sont exprimées dans la langue par défaut du SIM. |
| options | Pour une recherche d’itinéraire en « départ à »  les itinéraires retournés ne sont pas forcément ceux qui partent le plus tard possible  Pour une recherche d’itinéraire en « arrivée à »  les itinéraires retournés ne sont pas forcément ceux qui arrivent le plus tôt possible |

#### La structure de donnée en sortie

La structure de donnée en sortie correspond à l’élément ItinerariesResponse.

S’il n’existe aucune solution d’itinéraire, l’élément DetailedTrip est absent de la structure de réponse.

L’élément DetailedTrip est une solution d’itinéraire optimale par rapport :

* aux points de départ demandés (Departure)
* aux points d’arrivée demandés (Arrival)
* à l’horaire (DepartureTime, ArrivalTime)
* au critère d’optimisation (Algorithm)
* aux modes de déplacements sélectionnés (modes)
* aux conditions de rabattement autorisées
* à la contrainte éventuelle d’accessibilité

L’élément DetailedTrip présente un itinéraire

* où les modes des lignes de transport en commun empruntées (élément PublicTransportMode des PTRide) figurent tous dans la liste des modes de la structure de donnée en entrée
* où les modes des déplacements hors TC (élément SelfDriveMode des Leg) figurent tous dans la liste des modes de la structure de donnée en entrée (sauf si la structure de donnée en entrée autorise un rabattement avec la séquence SelfDriveConditions)
* où l’horaire de départ est postérieur à l’horaire demandé (dans le cas d’une structure de donnée en entrée ayant un paramètre DepartureTime), le décalage entre l’horaire demandé et l’horaire effectif est supérieur ou égal au temps d’accès au point de départ
* où l’horaire de d’arrivée est antérieur à l’horaire demandé (dans le cas d’une structure de donnée en entrée ayant un paramètre ArrivalTime), le décalage entre l’horaire demandé et l’horaire effectif est supérieur ou égal au temps d’accès au point d’arrivée
* où le choix des déplacements TC (PTRide) et hors TC (Leg) est optimal par rapport au critère Algorithm d’optimisation pour le choix du meilleur itinéraire

L’élément DetailedTrip décrit un itinéraire qui part depuis

* soit l’un des points de départ possibles (cas d’une structure de donnée en entrée avec le paramètre multiDepartures)
* soit le point de départ imposé (cas d’une structure de donnée en entrée avec le paramètre multiArrivals)

L’élément DetailedTrip décrit un itinéraire qui arrive

* soit à l’un des points d’arrivée possibles (cas d’une structure de donnée en entrée avec le paramètre multiArrivals)
* soit au point d’arrivée imposé (cas d’une structure de donnée en entrée avec le paramètre multiDepartures)

Dans le cas où la structure de donnée en entrée autorise le rabattement (élément SelfDriveCondition dans la séquence SelfDriveConditions), le rabattement précise un mode et une position sur l’itinéraire global.

Le tableau ci-dessous exprime de quelle manière le rabattement est pris en compte.

|  |  |  |
| --- | --- | --- |
| Mode de rabattement | Position | Prise en compte |
| vélo | début | Le premier élément Leg de l’itinéraire peut être du mode vélo même si le vélo ne figure pas parmi les modes de la structure de donnée en entrée (paramètres modes) |
| voiture | début | Le premier élément Leg de l’itinéraire peut être du mode voiture même si la voiture ne figure pas parmi les modes de la structure de donnée en entrée (paramètres modes) |
| vélo | fin | Le dernier élément Leg de l’itinéraire peut être du mode vélo même si le vélo ne figure pas parmi les modes de la structure de donnée en entrée (paramètres modes) |
| voiture | fin | Le dernier élément Leg de l’itinéraire peut être du mode voiture même si la voiture ne figure pas parmi les modes de la structure de donnée en entrée (paramètres modes) |

Les informations textuelles fournies par l’élément DetailedTrip sont dans la langue demandée (Language).

#### Composants qui utilisent le service

Seul l’aiguilleur utilise ce service de l’interface générique de SIM.

L’aiguilleur interroge ce service pour l’ensemble des SIM connectés.

### Service de recherche d’itinéraire non détaillée n-m

Le service calcule les meilleurs itinéraires permettant de relier n départs à m arrivées en respectant un certain nombre de critères. Le service renvoie une liste de résumés d’itinéraire (point et horaire départ et point et horaire d’arrivée).

Cette liste contient au plus m ou n, selon que le critère d’horaire est de type « départ à » ou « arrivée à ».

L’implémentation de ce type de service s’apparente aux algorithmes de calcul d’isochrone.

Les structures échangées dans ce service sont définies dans le fichier XSD MisSummedUpTrip.xsd.

#### La structure de donnée en entrée

La structure de donnée en entrée correspond à l’élément SummedUpItinerariesRequest du fichier XSD MisSummedUpTrip.xsd.

L’élément SumedUpItinerariesRequest propose plusieurs points de départ (éléments Departure de la séquence departures) et plusieurs points d’arrivée (éléments Arrivals de la séquence arrivals).

Les structures Departure et Arrival modélisent un emplacement géographique (qui peut être un arrêt aussi bien qu’une position géographique d’adresse) assorti d’une durée d’accès. La durée d’accès représente le temps nécessaire au voyageur pour se rendre à pied à cet emplacement.

Si l’élément SumedUpItinerariesRequest définit une séquence « options » non vide, alors l’une des 2 séquences « departures » ou « arrivals » ne contient qu’un seul élément.

Sans cela, le code d’exception BAD\_REQUEST est retourné dans l’élément Status de la réponse.

L’élément Algorithm est défini de la même manière que sur l’interface FRI, cf § 5.1.2.1 Service de demande de recherche d’itinéraire.

La structure de donnée en entrée peut contenir également des paramètres optionnels.

Lorsque ces paramètres ne sont pas renseignés, le service applique un fonctionnement par défaut présenté ci-dessous.

|  |  |
| --- | --- |
| Paramètre optionnel | Prise en compte en cas d’absence dans la structure de donnée en entrée |
| Algorithm | L’optimisation est faite comme si le paramètre était fixé à la valeur énumérée CLASSIC. |
| modes | La recherche est faite comme si l’ensemble des modes avaient été sélectionnés. |
| selfDriveConditions | Le rabattement n’est pas autorisé. Autrement dit, si le mode vélo (ou voiture) ne figure pas dans le paramètre « modes », alors aucun Leg ne peut être d’un mode vélo (ou voiture). |
| AccessibilityContraint | La recherche d’itinéraire ne tient pas compte de l’accessibilité. C’est équivalent à un paramètre AccessibilityContraint égal à « false ». |
| Language | Les informations textuelles des itinéraires sont exprimées dans la langue par défaut du SIM. |
| options | Pour une recherche d’itinéraire en « départ à »  les itinéraires retournés ne sont pas forcément ceux qui partent le plus tard possible  Pour une recherche d’itinéraire en « arrivée à »  les itinéraires retournés ne sont pas forcément ceux qui arrivent le plus tôt possible |

#### La structure de donnée en sortie

La structure de donnée de sortie (ou réponse) correspond à l’élément SummedUpItinerariesResponse du fichier XSD MisSummedUpTrip.xsd.

La réponse contient une séquence summedUpTrips d’au plus max(n, m) éléments SummedUpTrip :

* en départ à, la séquence compte m éléments au plus
* en arrivée à, la séquence compte n éléments au plus

Il peut en effet ne pas exister de solution pour certains couples (départ, arrivée).

L’élément SummedUpTrip est une description minimaliste d’itinéraires qui indique uniquement

* les points de départ et arrivée et les horaires sur ces points
* le nombre et la durée de correspondance

Chaque élément SummedUpTrip de la séquence summedUpTrips relie un couple (départ, arrivée) spécifique.

L’élément Departure de SummedUpTrip correspond à l’un des points de départ proposés dans la requête, à savoir un élément Departure de la séquence departures.

L’élément Arrival de SummedUpTrip correspond à l’un des points d’arrivée proposés dans la requête, à savoir un élément Arrival de la séquence arrivals.

Les itinéraires SummedUpTrip qui figurent dans la réponse sont choisis de manière à respecter les critères de la requête

* où les modes des lignes de transport en commun empruntées (élément PublicTransportMode des PTRide) figurent tous dans la liste des modes de la requête
* où les modes des déplacements hors TC (élément SelfDriveMode des Leg) figurent tous dans la liste des modes de la requête (sauf si la requête autorise un rabattement avec la séquence SelfDriveConditions)
* où l’horaire de départ est postérieur à l’horaire demandé (dans le cas d’une requête ayant un paramètre DepartureTime), le décalage entre l’horaire demandé et l’horaire effectif est supérieur ou égal au temps d’accès au point de départ
* où l’horaire de d’arrivée est antérieur à l’horaire demandé (dans le cas d’une requête ayant un paramètre ArrivalTime), le décalage entre l’horaire demandé et l’horaire effectif est supérieur ou égal au temps d’accès au point d’arrivée
* où le choix des déplacements TC (PTRide) et hors TC (Leg) est optimal par rapport au critère Algorithm de la requête

En "départ à" (« Departure\_at » renseigné)

* les horaires d’arrivée des itinéraires sont les meilleurs possibles (arrivée au plus tôt possible à leur point de destination)
* les horaires de départ sont postérieurs ou égal à celui de la requête et tiennent compte des durées d’accès aux arrêts de départ
* les itinéraires pris en compte sont compatibles avec l’ensemble des autres critères de la requête : modes de transport (séquence modes),
  + accessibilité (élément AccessibilityContraint),
  + rabattement (séquence selfDriveConditions),
  + critère d’optimisation (élément Algorithm)

En "arrivée à" :

* les horaires de départ des itinéraires sont les meilleurs possibles (départ au plus tard possible de leur point de départ)
* les horaires d’arrivée sont antérieurs ou égal à celui de la requête et tiennent compte des durées d’accès aux arrêts d’arrivée

Dans le cas où la requête définit l’option d’optimisation (DEPARTURE\_ARRIVAL\_OPTIMIZED) et sous réserve que la requête ne propose qu’un seul point de départ ou d’arrivée, la réponse produite garantit

* dans le cas d’une requête en "départ à", que l’horaire d’arrivée de chaque itinéraire est au plus tôt possible et que par rapport à cet horaire d’arrivée, l’horaire de départ est au plus tard possible
* dans le cas d’une requête en "arrivée à", que l’horaire de départ de chaque itinéraire est au plus tard possible et que par rapport à cet horaire de départ, l’horaire d’arrivée est au plus tôt possible

S’il n’existe aucune solution d’itinéraire ou si le service rencontre un cas d’exception, alors la liste summedUpTrips est vide.

#### Composants qui utilisent ce service

Seul l’aiguilleur utilise ce service.

### Collecte des arrêts

Le service liste l’ensemble des arrêts d’un SIM.

Les identifiants associés aux arrêts listés sont tous reconnus par les services de l’interface qui réalisent une recherche d’itinéraire.

Les structures échangées dans ce service sont définies dans le fichier XSD MisCollectStops.xsd.

#### La structure de donnée en entrée

Aucune

#### La structure de donnée en sortie

La structure de donnée de réponse correspond à l’élément StopsResponse.

L’élément StopsResponse contient un élément PublicationDelivery qui respecte le format NETEX.

Cet élément contient essentiellement un SiteFrame.

Les différents arrêts du SIM sont collectés en précisant une typologie (attribut TypeOfPlaceRef). Dans le cas du POC, cette typologie n’est pas utile, puisque le POC ne s’intéresse qu’aux seuls arrêts physiques (les structures « quays »).

Par contre dans le cas d’une alimentation du un référentiel d’arrêts partagé, le service est en mesure de fournir la typologie (lieu d’arrêt monomodal, lieu d’arrêt multimodal, etc…). Le méta-système pourra ainsi utiliser cette typologie pour la renseigner dans la base des métadonnées (colonne Type de la table STOPS) et l’utiliser ensuite pour un calcul « automatisé » des transitions (cf § 5.2.1 Le schéma de la base).

S’agissant du service de recherche de localités au niveau du méta-système, la typologie des arrêts qui sont retournés (en tant que « localité ») correspond à celle qui fourni par les différents SIM à travers le service de sélection de leurs arrêts.

Les identifiants des arrêts qui sont fournis dans ce service, sont ceux qui peuvent être reconnu dans les 2 services de calcul d’itinéraire (recherche de solutions non détaillées et recherche d’une solution détaillée). Au niveau des requêtes à ces services de calcul l’identifiant se place dans l’élément PlaceTypeId.

#### Composants qui utilisent ce service

Seul le système de synchronisation des métadonnées utilise ce service.

### Capacités du SIM

Le service liste les capacités d’un SIM relatives aux services de l’interface qui réalisent des recherches d’itinéraire.

Les structures échangées dans ce service sont définies dans le fichier XSD MisCapabilities.xsd.

#### La structure de donnée en entrée`

Aucune

#### La structure de donnée en sortie

La structure de réponse correspond à l’élément CapabilitiesResponse.

Les éléments MultipleStartsAndArrivals et GeographicPositionCompliant (de la structure de réponse) correspondent au niveau de la méta-base, aux colonnes du même nom dans la table MIS (cf § 5.2.1 Le schéma de la base).

#### Composants qui utilisent ce service

Seul le système de synchronisation des métadonnées utilise ce service.

# Partie 3

## Point d’accès fonctionnel

### Accès au méta-système (FRI, DRI)

Le service de recherche de localité est le point d’accès fonctionnel.

Celui-ci permet de définir les points de départ et d’arrivée, par exemple un arrêt sur SIM et POI sur un autre SIM.

Le service de recherche d’itinéraire est ensuite pleinement accessible.

Néanmoins, le service de recherche d’itinéraire est accessible directement dans une certaine mesure. Les points de départ et d’arrivée peuvent être renseignés comme des positions géographiques sans que cela nécessite l’usage du service de recherche de localité.

### Accès à la base des métadonnées

Au niveau fonctionnel, le langage de consultation et mise à jour donne accès à toutes les données de la base.

### Accès à l’interface générique de SIM

Les services ci-dessous sont accessibles directement et indépendamment :

* service de consultation des capacités du SIM
* service de collecte des arrêts

Les résultats fournis par ces 2 services permettent d’utiliser correctement les 2 autres services :

* service de recherche d’itinéraire non détaillée n x m
* service de recherche d’itinéraire détaillée 1 x n

## Structures échangées

Ayant adopté un format de donnée XML pour décrire les structures de données échangées par les interfaces DRI, FRI et l’interface générique de SIM, la structure de ces données est décrite en XML Schéma.

Les fichiers XML Schéma sont rassemblés dans le Dossier des définitions XSD des structures d’échange. Le tableau ci-dessous classe ces fichiers en fonction de l’interface à laquelle ces fichiers se rapportent.

|  |  |
| --- | --- |
| Interface | Fichier XML Schéma |
| Propres à l’interface générique de SIM | MisProtocol.xsd  MisPlanTrip.xsd  MisPlanSummedUpTrip.xsd  MisLocationTime.xsd  MisCollectStop.xsd  MisCapabilities.xsd |
| Propres aux interfaces DRI, FRI | PlanTrip.xsd  SearchPoints.xsd  Protocol.xsd |
| Commun aux interfaces DRI, FRI et l’interface générique de SIM | Site.xsd  Itinerary.xsd  ItineraryRequest.xsd  ProtocolFramework.xsd |

### Les structures d’échange des interfaces du méta-système (FRI, DRI)

Le document PlanTrip.pdf présente l’ensemble des structures d’échange du service de recherche d’itinéraire au niveau des interfaces DRI et FRI.

Le document SearchPoints.pdf présente l’ensemble des structures d’échange du service de recherche de localité au niveau des interfaces DRI et FRI.

Les documents PlanTrip.pdf et SearchPoints.pdf figurent dans le dossier des Documents PDF de présentation des schémas des structures d’échange.

### Les structures d’échange des interfaces génériques de SIM

Le document MisPlanSummedUpTrip.pdf présente l’ensemble des structures d’échange du service de recherche d’itinéraire non détaillée n x m.

Le document MisPlanTrip.pdf présente l’ensemble des structures d’échange du service de recherche d’itinéraire détaillée 1 x n.

Le document MisCollectStop.pdf présente l’ensemble des structures d’échange du service de collecte d’arrêts.

Le document MisCapabilities.pdf présente l’ensemble des structures d’échange du service de capacités du SIM.

Les documents MisPlanSummedUpTrip.pdf, MisPlanTrip.pdf, MisCollectStop.pdf et MisCapabilities.pdf figurent dans le dossier des Documents PDF de présentation des schémas des structures d’échange.

### Les structures d’échange de l’interface de lecture écriture de la base des métadonnées

L’interface est constituée par un langage de requêtes. Les structures échangées dépendent donc des requêtes utilisées.

# Liste des figures

Figure 1 Territoire et SIM en jeu 9

Figure 2 Traces des SIM 10

Figure 3 Diagramme de composants 21

Figure 4 Principaux cas d'usage 21

Figure 5 Diagramme de composants interface FRI DRI 26

Figure 6 Diagramme d'état général de recherche d'itinéraire 32

Figure 7 Sous-diagramme propre à l'état A 33

Figure 8 Sous-diagramme propre à l'état B 34

Figure 9 Diagramme d’état de recherche de localités 36

Figure 10 Diagramme de composants : interface langage de requêtes 37

Figure 11 Schéma de la base des métadonnées 38

Figure 12 Diagramme de composants: interface générique de SIM 41