



**Cerema**

Centre d'études et d'expertise sur les risques,  
l'environnement, la mobilité et l'aménagement

# Plateforme TEMPUS pour le calcul d'itinéraires

## Architecture et algorithmes

Aurélié Bousquet, CEREMA – Direction Territoriale Centre-Est  
Romain Billot, IFSTTAR, Laboratoire Ingénierie Circulation Transports

# Principes et objectifs

Outil développé conjointement par

- l'IFSTTAR
- le CEREMA
- Oslandia

Test d'algorithmes de calcul d'itinéraires multimodaux

- Mise en œuvre facilitée sur différents territoires et réseaux ;
- Comparaison :
  - de métriques d'efficacité des algorithmes : temps de calcul, nombre d'itérations, ...
  - des solutions générées par les algorithmes : validation des méthodes, évaluation de la qualité des solutions pour les méthodes approchées.

# Principes et objectifs

« Framework » C++ pour

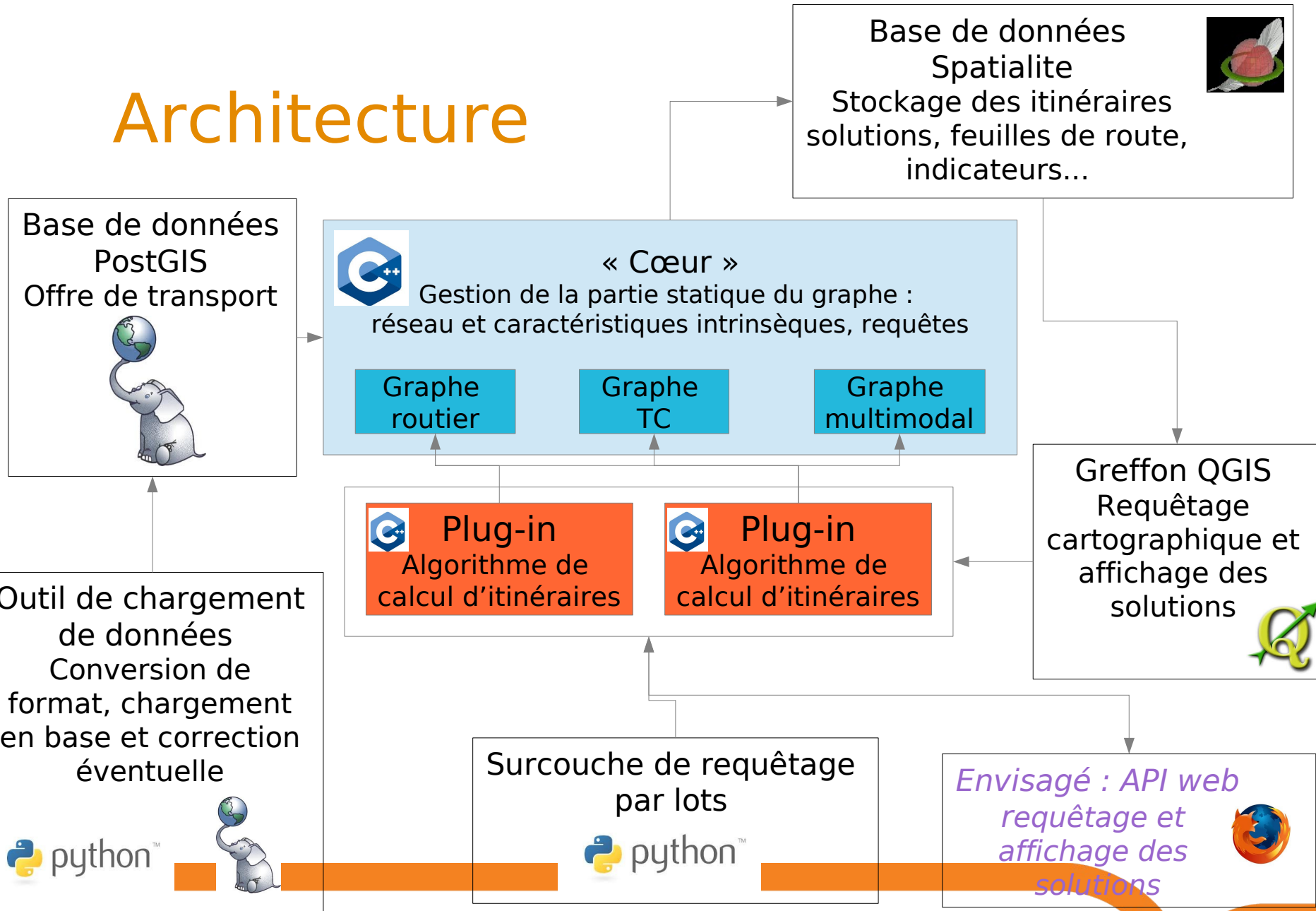
- la construction et la manipulation de graphes représentant des réseaux de transport ;
- le calcul d'itinéraires (algorithmes d'étiquetage et autres...) ;
- la manipulation et la présentation des solutions (concepts d'itinéraires, de feuilles de route, ...).

Généricité des développements :

- Conçu comme une extension de la Boost Graph Library (BGL)
- Implémentation des algorithmes par « templates »

Outil multi-plateforme : Windows (installation packagée), Linux

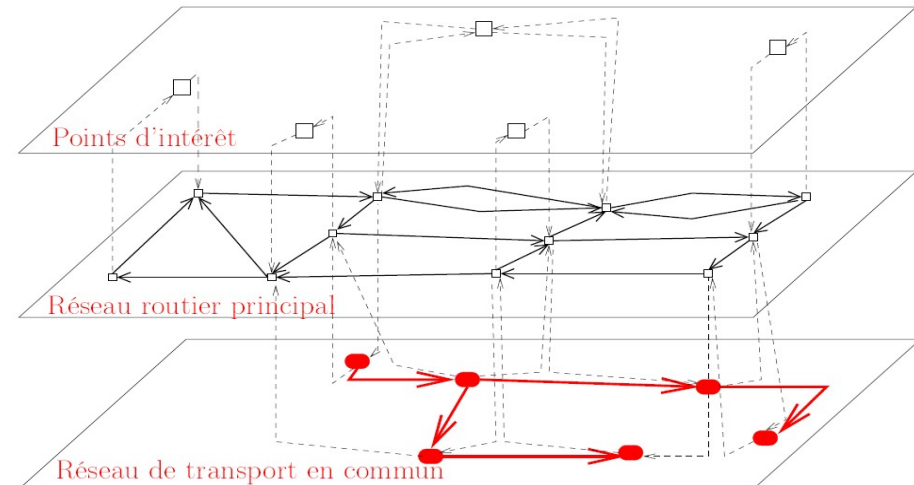
# Architecture



# Architecture du « cœur »

## Graphe routier

- Support des autres réseaux : TC, véhicules partagés, etc.
- Représentation condensée : pas de doublons des parties communes des graphes pour différents modes ;



## N graphes de transport collectif

- Un graphe par réseau (bus, tram, métro, train, autocar, etc.) ;
- Représentation condensée : points d'arrêts et tronçons pouvant être desservi par plusieurs lignes ;

## Points d'intérêt

- Stations de location de véhicules (auto-partage, vélos et voitures libre-service, ...)
- Autres points d'intérêt définis par l'utilisateur

# Architecture des « plug-in »

Structure identique pour tous les plugins

- `post_build` : processus lancé après la construction du graphe, une seule fois
- `pre_process` : processus lancé à chaque requête d'itinéraire avant l'exécution de la requête (mises à jour du graphe, etc.)
- `process` : exécution d'une requête d'itinéraire

Peut interroger :

- le graphe routier
- le graphe TC ou
- le graphe multimodal (graphe routier + graphe(s) TC + points d'intérêt).

Traitements de plusieurs requêtes simultanées, « multi-threading »

# Outil de chargement de données

## Modèle de données « Tempus »

- En partie inspiré par les structures de données proposées par les éditeurs de données et les formats standard
- Prise en compte de la multimodalité

## Chargement des données routières : réseau et caractéristiques, mouvements tournants

- TomTom, Navteq : chargement et conversion au format Tempus
- OSM : chargement et conversion au format Tempus  
+ corrections topologiques : découpage de polylignes aux nœuds, fusion de polylignes trop découpées
- Extension envisagée : données IGN (BD Topo, BD Carto, BD Alti)

# Outil de chargement de données

## Offre de transport collectif

- Chargement automatisé dans les formats
  - GTFS : chargement et conversion au format Tempus
  - Neptune : utilisation de l'outil « Chouette » pour convertir en GTFS
- Correction d'erreurs : utilisation de l'outil « Chouette »

Points d'intérêt géolocalisés par intégration directe d'un fichier shape dans la base PostgreSQL

## Multimodalité

- Création automatique des liens entre le réseau routier, le réseau TC et les points d'intérêt (véhicules partagés...)
- Peut être validé/adapté à la main dans QGIS si besoin



# Interface QGIS

**Couches**

- Tempus\_Roadmap\_1
  - Transfer
  - Roads
  - Public transports
- Tempus\_pin\_points
- road\_section

**Routing - connected**

Connection Plugin Query Performances Roadmap History

Tempus\_Roadmap\_1 Distance 8510.5m

Z (m)

42.0

42.0

Hide elevations

	Direction	Costs
1	Walk on Montée Victor Hugo	Distance: 110.8 m
2	Walk on Montée Victor Hugo	Distance: 17.6 m
3	Walk on Montée Victor Hugo	Distance: 6.0 m
4	Walk on Montée Victor Hugo	Distance: 78.3 m

Coordonnée : 840146,6521536 Echelle 1:40230 Rendu EPSG:2154

# Plug-in de calcul d'itinéraire

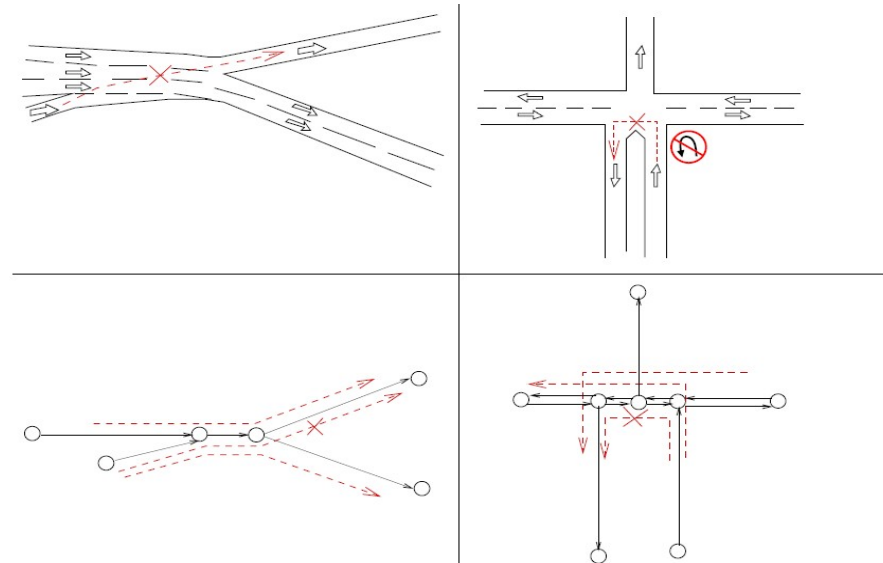
Plugins de base (algorithmes exemples) déjà disponibles sur le dépôt

- Algorithme de Dijkstra minimisant le temps de parcours
  - Routier simple (sans prise en compte des mouvements tournants interdits)
  - Transport Public
- Temps de parcours statiques basés sur une vitesse moyenne par arc

# Plug-in de calcul d'itinéraire

Nouveau plugin multimodal (disponible d'ici l'été 2014)

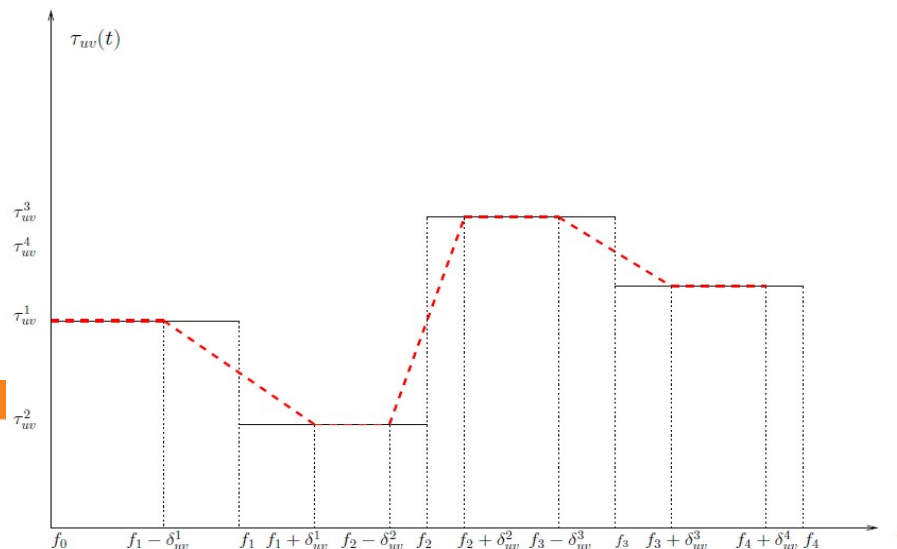
- Algorithme de Dijkstra multimodal minimisant le temps de parcours
  - Modes individuels : marche, voiture, vélo
  - Modes partagés : auto-partage, vélo ou voiture libre-service
  - Modes de transport collectif : définis par horaires ou par fréquences
- Prise en compte de pénalités ou restrictions sur les mouvements tournants sur le réseau routier, distinctes par mode
  - Simples (sur 2 arcs) ou
  - Complexes (sur 3 arcs et plus) :
    - Entrecroisements autoroutiers, demi-tours...
    - Représentation à l'aide d'une structure d'automate



# Plug-in de calcul d'itinéraire

Nouveau plugin multimodal (disponible d'ici l'été 2014)

- Prise en compte de la position des véhicules privés (vélo, voiture) au moment du départ du déplacement
- Prise en compte de temps intermodaux
  - Temps de stationnement,
  - Temps de sécurité entre deux services TC...
- Prise en compte de temps de parcours définis par une fonction de constante par morceaux pour les voitures particulières
  - Processus de lissage pour garantir le respect de la propriété FIFO  
=> maintien de la complexité polynomial du problème de plus court chemin



# Plug-in de calcul d'itinéraire

A venir au second semestre 2014

- Passage du « one-to-one » au « one-to-all » et au « all-to-one »
- Insertion de techniques d'accélération d'algorithmes
  - A\*, ALT, « contraction hierarchies » pour la partie routière
- Ajout de nouveaux critères d'optimisation :
  - Chemin le plus simple (pour les modes individuels)
    - Le moins de changements de direction
    - Le moins de carrefours complexes
  - Chemin le plus confortable
    - Nombre de correspondances
    - Sécurité, paysage et dénivelé pour les modes doux
  - Chemin de coût minimum avec modélisation de systèmes de tarifications complexes :
    - Péage définis gare à gare sur le réseau autoroutier
    - Tarifications zonales des transports collectifs
- Intégration d'un algorithme bi-objectif de base

# Plug-in de calcul d'itinéraire

## A venir au second semestre 2014 (suite)

- Prise en compte de nouvelles contraintes sur le problème d'optimisation
  - Accès Personnes à Mobilité Réduite (données disponibles dans OSM)
  - Limitations sur les critères définis précédemment
- Plugin permettant l'optimisation d'un aller-retour multimodal (algorithme à fixation d'étiquettes proposé par Huguet et al. 2013)
  - Extension envisagée pour une boucle de n déplacements

## A venir en 2015

- Intégration d'indicateurs d'incertitude sur le temps de parcours (Thèse de Raphaël Delhome en cours à l'IFSTTAR)
- Diversification des algorithmes multi-objectif
- Optimisation d'itinéraires en tenant compte de contraintes spécifiques au véhicule électrique (Thèse de Fouad Baouche en cours à l'IFSTTAR)
  - Etat de charge (rechargement en descente)
  - Passage par les bornes de rechargement

# Autres outils open-source de calcul d'itinéraires

OpenTripPlanner - <http://opentripplanner.com/>

- Multimodal avec fonctionnalités avancées (prise en compte des Véhicules en Libre-Service, notion de confort de déplacement pour les vélos...)
- Méthodes d'accélération implémentées
- Développé en Java
- Dispose d'un client web pour le calcul d'itinéraires et la représentation de l'accessibilité
- Interactions avec Tempus à l'étude

OSRM - [http://wiki.openstreetmap.org/wiki/Open\\_Source\\_Routing\\_Machine](http://wiki.openstreetmap.org/wiki/Open_Source_Routing_Machine)

- Routier statique uniquement
- Réputé pour ses très bonnes performances en temps de calcul (basé sur l'algorithme « contraction hierarchies »)
- Développé en C++ mais pas basé sur l'architecture Boost
- Traitement automatisé pour les données OSM uniquement



# Points forts de « Tempus »

Généricité et extensibilité du code C++

Richesse de fonctionnalités à venir, en lien avec les travaux de recherche menés à l'IFSTTAR

- Mouvements tournants complexes
- Politiques tarifaires complexes
- Véhicules électriques
- Incertitude sur le temps de parcours...

Interfaçage direct avec QGIS et PostgreSQL

- Permet d'utiliser les fonctions SIG d'analyse spatiale et des requêtes SQL sur les données

Chargement de données automatisé pour les formats les plus courants



# Merci de votre attention

Dépôt du projet

<https://github.com/Tempus-project/tempus>

Documentation

<http://tempus-project.github.io/tempus/documentation.html>

Contacts

[aurelie.bousquet@cerema.fr](mailto:aurelie.bousquet@cerema.fr)

[romain.billot@ifsttar.fr](mailto:romain.billot@ifsttar.fr)