



MINISTÈRES
TRANSITION ÉCOLOGIQUE
AMÉNAGEMENT DU TERRITOIRE
TRANSPORTS
VILLE ET LOGEMENT

*Liberté
Égalité
Fraternité*



HARMONIA: L'IA AU SERVICE DE LA MODÉLISATION DES DONNÉES

Accélérer et fiabiliser la création de modèles sémantiques.

Les défis de la modélisation sémantique

- Complexité technique pour les utilisateurs métiers.
- Tendance à « réinventer la roue » (création de concepts en doublon).
- Manque d'interopérabilité entre les différents systèmes.

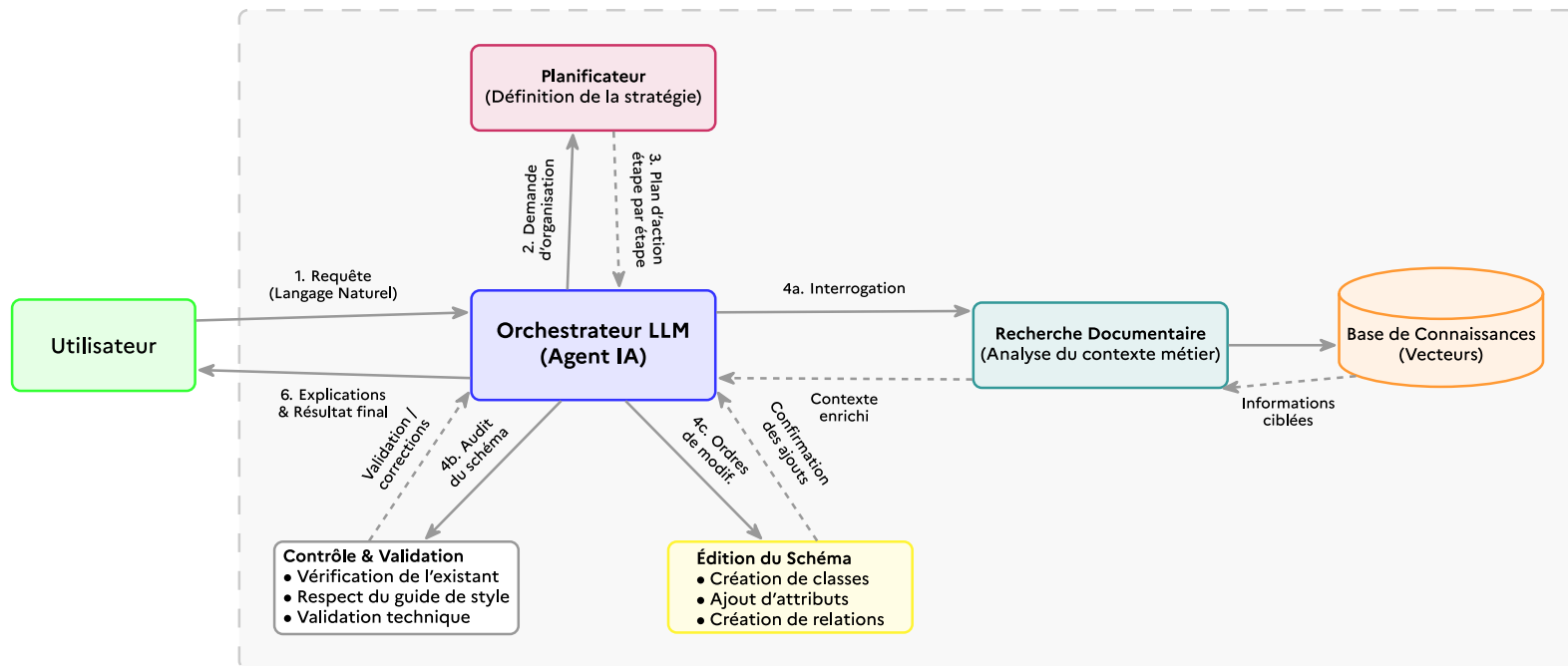
Un assistant IA guidé par les standards

- **Règles d'or** : Réutiliser les standards existants plutôt que recréer.
- **Technologie** : Agent hybride combinant RAG (Génération augmentée par la recherche) pour les standards et Tool Calling pour l'exécution d'actions.
- **Garantie** : Maîtrise totale (*Human-in-the-loop*) : l'agent propose, l'utilisateur valide chaque étape critique.

Origine : L'initiative SEMIC

- **Le SEMIC** : *Semantic Interoperability Community*, initiative de la Commission européenne.
- **Mission** : Harmoniser l'interopérabilité sémantique dans l'espace public européen.
- **Référentiel** : Fournit les guides de style, les modèles et les meilleures pratiques pour la modélisation RDF/OWL.
- **Utilisation** : L'outil intègre nativement ces standards pour garantir que chaque modèle produit est conforme aux directives européennes.

Sous le capot : L'architecture de l'outil



L'interface en trois onglets

Chercher

*Explorer les standards
et ontologies existants*

Recherchez des concepts métier dans des sources indexées (schema.data.gouv.fr, SEMIC, Schema.org, LOV, FIWARE, etc.). Filtrez par source et identifiez ce qui est réutilisable.

Analyser

*Importer un modèle et
dialoguer avec l'IA*

Importez un modèle existant (XMI, TTL, JSON, SQL, texte) et posez des questions sur sa structure, sa complétude ou sa conformité aux standards.

Éditer

*Créer ou modifier un
modèle guidé par l'IA*

Construisez un modèle à partir de zéro ou enrichissez un modèle existant. L'IA suggère classes, attributs et relations, vous gardez le contrôle. Exportez en XMI, TTL, SVG ou PNG.

Chercher

modèles de transport en commun

Chercher

Chercher des modèles

modèles de transport en commun

Chercher

+ LOV

+ INSPIRE

+ FIWARE

+ schema.org

+ schema.data.gouv.fr

+ CNIG

+ SAREF

+ SEMIC

20 RÉSULTAT(S)

PublicTransportRoute.json

Ce document est un schéma JSON (en anglais) définissant le modèle de données « PublicTransportRoute » pour la mobilité urbaine SynchroniCity.

Points clés : - Spécifie les propriétés d'un itinéraire de transport public (code, type, couleur, segments et horaires). - Impose comme champs obligatoires l'identifiant, le type et le mode de transport.

Score: 0.7004 Source: FIWARE ID: 736a47a2...



TransitManagement.json

Ce document est un schéma JSON définissant le modèle de données "TransitManagement" pour un système de transport public.

Points clés : - Standardise les données de suivi des véhicules (position, vitesse, retard) et les informations d'agence. - S'appuie largement sur les spécifications GTFS (Static et Realtime).

Langue principale : Anglais.

Score: 0.5779 Source: FIWARE ID: 2c71b457...



PublicTransportStop.json

Ce document est un schéma JSON (en anglais) définissant un modèle de données générique pour les arrêts de transports publics (PublicTransportStop).

Points clés : - Spécifie des propriétés telles que le code de l'arrêt, l'accessibilité PMR et le type de transport. - Inclut des champs pour l'estimation du nombre de personnes et les horaires d'ouverture.

Score: 0.5461 Source: FIWARE ID: a0c26592...



FareCollectionSystem.json

Ce document est un schéma JSON (en anglais) définissant le modèle de données "FareCollectionSystem" pour les systèmes de collecte de titres de transport public.

Points clés : - Spécifie les propriétés des transactions (tarifs, types de tickets, identifiants de cartes). - Intègre des données de flux (nombre de passagers, niveau d'occupation) et des références GTFS.

Score: 0.5365 Source: FIWARE ID: 81ebfbc2...



Schema_des_donnees_accessibilite_des_ERPs.json

Ce document est une spécification technique (format JSON) définissant le schéma de données pour l'échange d'informations sur l'accessibilité des Établissements Recevant du Public (ERP)

Analyser

Fais une comparaison concise, en te concentrant uniquement sur les grandes différences entre ces trois modèles.

PublicTransportSto
p

PublicTransportRoute

TransitManagement



Analyser des modèles

fais une comparaison concise, en te concentrant uniquement sur les grandes différences entre ces trois modèles.

Voici une comparaison concise des trois modèles, centrée sur leurs finalités et leurs différences structurelles :

1. PublicTransportStop (Le Point d'Arrêt)

- **Focus** : L'infrastructure physique et statique.
- **Particularité** : Il se concentre sur les caractéristiques de l'arrêt (code de l'arrêt, accessibilité PMR, horaires d'ouverture) et sur l'affluence (estimation du nombre de personnes en attente via des capteurs).
- **Rôle** : C'est le point d'ancrage géographique du réseau.

2. PublicTransportRoute (Le Trajet Théorique)

- **Focus** : La planification et la définition du service.
- **Particularité** : Il modélise la ligne en tant qu'entité logique (code de la route, couleur visuelle, segments de trajet) et son calendrier théorique (jours de la semaine, heures d'ouverture/fermeture).
- **Rôle** : Il définit "ce qui est prévu" (le réseau et ses horaires).

3. TransitManagement (L'Exploitation en Temps Réel)

- **Focus** : Le suivi opérationnel et dynamique.
- **Particularité** : C'est le modèle le plus complexe et le plus riche. Il intègre des données de flux (GTFS Realtime) : position GPS exacte des véhicules, retards (`trip_delay`), vitesse, capacité d'occupation et

Interrogez l'assistant de modélisation...



@ PublicTra... 

@ PublicTra... 

@ TransitMa... 

 Sources



Caractéristiques PublicTransportStop PublicTransportRoute TransitManagement

Analyser des modèles

- **Rôle** : C'est le point d'ancrage géographique du réseau.

2. PublicTransportRoute (Le Trajet Théorique)

- **Focus** : La planification et la définition du service.
- **Particularité** : Il modélise la ligne en tant qu'entité logique (code de la route, couleur visuelle, segments de trajet) et son calendrier théorique (jours de la semaine, heures d'ouverture/fermeture).
- **Rôle** : Il définit "ce qui est prévu" (le réseau et ses horaires).

3. TransitManagement (L'Exploitation en Temps Réel)

- **Focus** : Le suivi opérationnel et dynamique.
- **Particularité** : C'est le modèle le plus complexe et le plus riche. Il intègre des données de flux (GTFS Realtime) : position GPS exacte des véhicules, retards (`trip_delay`), vitesse, capacité d'occupation et mises à jour des temps d'arrivée/départ.
- **Rôle** : Il gère "ce qui se passe réellement" sur le terrain.

Synthèse des différences

Caractéristique	PublicTransportStop	PublicTransportRoute	TransitManagement
Nature	Statique / Infrastructure	Planifiée / Logique	Dynamique / Opérationnelle
Élément central	L'arrêt → Passagers	La ligne → Segments	Le véhicule → Position/État
Indicateur clé	Affluence (<code>peopleCount</code>)	Calendrier (<code>schedule</code>)	Retard et Vitesse (<code>delay</code> , <code>speed</code>)
Temporalité	Permanente	Programmée	Instantanée (Temps réel)



Interrogez l'assistant de modélisation...



PublicTra...  

PublicTra...  

TransitMa...  

 Sources



Éditer

Je veux modéliser un arrêt de transport en commun. Quelles classes et quels attributs me recommandes-tu ? Sois concis dans toutes tes réponses.



Éditer un modèle



Modèle vide
Cliquez sur "Classe" pour commencer.

+ Classe

+ Attribut

+ Relation



Je souhaite modéliser un arrêt de transport en commun à l'aide de deux classes. Quelles classes, quels attributs et quelle relation entre ces deux classes me recommandes-tu ? N'oublie pas de me donner également le type de chaque attribut. Sois concis dans toutes tes réponses.

Plan d'action

- 1 Extraire les concepts existants du modèle 'arret_transport' pour vérifier s'il y a déjà des bases de modélisation pour les arrêts de transport.
- 2 Rechercher des standards de modélisation pour les arrêts de transport en commun (ex: GTFS, Transmodel, NeTEx) afin d'identifier les deux classes les plus pertinentes, leurs attributs et leur relation.

retrieve_documents

nécessite un outil

- 3 Sélectionner les deux classes optimales (ex: Arrêt et PointDAccès ou Arrêt et Ligne), définir leurs attributs avec types de données et la relation sémantique entre elles.
- 4 Pour chaque concept retenu, décider s'il faut réutiliser un URI standard trouvé dans les documents ou suggérer un URI local cohérent avec le namespace de

Interrogez l'assistant de modélisation...

arret_transport

Sources

Éditer un modèle



Modèle vide
Cliquez sur "Classe" pour commencer.

Classe

Attribut

Relation



L'utilisateur

- Présenter la recommandation concise comprenant : les deux classes, les attributs (avec types), la relation et les URIs suggérées.

L'utilisateur a fourni un modèle vide. La recommandation doit être concise et s'appuyer sur des standards de transport. Le plan prévoit une étape de décision sur les URIs pour éviter toute invention arbitraire.

Je cherche des standards pour modéliser un arrêt de transport en comm...

5 RÉSULTAT(S)

PublicTransportStop.json

Ce document est un schéma JSON (en anglais) définissant un modèle de données générique pour les arrêts de transports publics (PublicTransportStop).

Points clés : - Spécifie des propriétés telles que le code de l'arrêt, l'accessibilité PMR et le type de transport. - Inclut des champs pour l'estimation du nombre de personnes et les horaires d'ouverture.

Score: 0,9586 Source: FIWARE ID: #8c26592...

GtfsStopTime.json

Ce document est un schéma JSON (en anglais) définissant un modèle de données générique pour les arrêts de transports publics (GtfsStopTime).

Interrogez l'assistant de modélisation...

arrêt_transport

Sources

Éditer un modèle

Modèle vide
Cliquez sur "Classe" pour commencer.

Classe

Attribut

Relation

+ - ↺

Score: 0.9117 Source: FIWARE ID: 1094eb081...

Sur la base des standards GTFS et des documents récupérés, je vous recommande la modélisation suivante :

1. Classe : Arrêt (stop)

- **Nom** (stop_name) : xsd:string
- **Code** (stop_id) : xsd:string
- **Latitude** (stop_lat) : xsd:decimal
- **Longitude** (stop_lon) : xsd:decimal

2. Classe : Quai (stopTime / Platform)

- **Code Quai** (platform_code) : xsd:string
- **Nom Quai** (platform_name) : xsd:string

Relation :

- **Lien** : Un **Arrêt** possède un ou plusieurs **Quais**.
- **Cardinalité** : 1 Arrêt -> 1..* Quais.

Sources utilisées :

- **gtfsStop.json** : Définition de la classe Arrêt et ses attributs de localisation.
- **gtfsStopTime.json** : Identification de la notion de quai/platforme associée aux arrêts.



Interrogez l'assistant de modélisation...

arrêt_transport ↵

Sources



Éditer

Je souhaite que tu crées un modèle à partir de tes suggestions,
en veillant à réutiliser des URI existantes.



Éditer un modèle



Classe

Attribut

Relation

Modèle vide
Cliquez sur "Classe" pour commencer.



Je souhaite que tu crées un modèle à partir de tes suggestions, en veillant à réutiliser des URI existantes.

Plan d'action

- 1 Analyser l'historique de la conversation pour identifier les suggestions de modélisation précédemment faites à l'utilisateur (classes, attributs, relations et URIs standards suggérées).
- 2 Pour chaque concept suggéré, décider s'il faut réutiliser une URI standard identifiée dans les suggestions ou créer une URI locale cohérente avec le namespace du modèle "arret_transport".
- 3 Si des URIs sont manquantes ou ambiguës pour certains concepts, produire un récapitulatif des URIs trouvées/manquantes et demander la confirmation de l'utilisateur avant toute mutation.
- 4 Une fois les URIs validées, créer séquentiellement les classes, les attributs et les connecteurs dans le modèle cible.

nécessite un outil

Le plan commence par une phase d'extraction des suggestions depuis le contexte. Conformément aux règles de gouvernance des URIs, aucune mutation (add_class, etc.) ne sera effectuée sans que les URIs ne soient explicitement validées ou confirmées par l'utilisateur via

Interrogez l'assistant de modélisation...

arret_transport

Sources



Pour créer le modèle basé sur mes suggestions précédentes, je dois d'abord valider les

Éditer un modèle



+ Classe

+ Attribut

+ Relation

Modèle vide
Cliquez sur "Classe" pour commencer.

+ - ↺

Pour créer le modèle basé sur mes suggestions précédentes, je dois d'abord valider les URI. Conformément aux règles de modélisation, je ne peux pas inventer d'URI standards.

Voici le récapitulatif des concepts à créer et les URI proposées :

URI trouvées (via documents récupérés) :

- **Classe Arrêt** → <http://vocab.gtfs.org/stop> (Source : gtfsStop.json)
- **Attribut Nom** → http://vocab.gtfs.org/stop_name (Source : gtfsStop.json)
- **Attribut Code** → http://vocab.gtfs.org/stop_id (Source : gtfsStop.json)
- **Attribut Latitude** → http://vocab.gtfs.org/stop_lat (Source : gtfsStop.json)
- **Attribut Longitude** → http://vocab.gtfs.org/stop_lon (Source : gtfsStop.json)

URI manquantes (concepts sans URI explicite dans les docs) :

- **Classe Qual** : Concept identifié mais URI non spécifiée.
💡 Suggestion locale : http://example.org/ontology/arret_transport/Qual
- **Attribut Code Qual** : Absent des documents.
💡 Suggestion locale : http://example.org/ontology/arret_transport/platform_code
- **Attribut Nom Qual** : Absent des documents.
💡 Suggestion locale : http://example.org/ontology/arret_transport/platform_name
- **Relation "possède"** (Arrêt → Qual) : Absente des documents.
💡 Suggestion locale : http://example.org/ontology/arret_transport/possedeQual

☑ Confirmez-vous ces URI ou souhaitez-vous me fournir un namespace spécifique (ex: <https://schema.data.gouv.fr/...>) pour les éléments manquants ?

SA

Interrogez l'assistant de modélisation...

arret_transport ↗

🔍 Sources



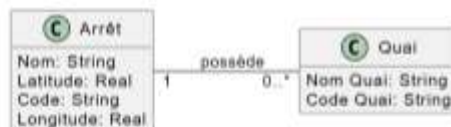
Éditer

Oui, je confirme.



Éditer un modèle

arret_transport



Classe

Attribut

Relation

+ - ↺

Le modèle a été créé avec succès dans le modèle

arret_transport__20260903142151007471. Voici le résumé des éléments ajoutés :

Éléments ajoutés :

- **Classe Arrêt** → <http://vocab.gtfs.org/Stop>
 - Source : [gtfsStop.json](#)
 - **Attribut Nom** → http://vocab.gtfs.org/stop_name (Source : [gtfsStop.json](#))
 - **Attribut Code** → http://vocab.gtfs.org/stop_id (Source : [gtfsStop.json](#))
 - **Attribut Latitude** → http://vocab.gtfs.org/stop_lat (Source : [gtfsStop.json](#))
 - **Attribut Longitude** → http://vocab.gtfs.org/stop_lon (Source : [gtfsStop.json](#))
- **Classe Quai** → http://example.org/ontology/arret_transport/Quai
 - Source : URI locale cohérente
 - **Attribut Code Quai** → http://vocab.gtfs.org/platform_code (Source : [gtfsStop.json](#))
 - **Attribut Nom Quai** → http://example.org/ontology/arret_transport/platform_name (Source : URI locale cohérente)
- **Relation "possède"** (Arrêt → Quai) →

Interrogez l'assistant de modélisation...

arret_transport ↴

Oui, je confirme.

Sources

Les bénéfices clés

- Création rapide et accessible.
- Interopérabilité garantie dès la conception (URIs existantes).
- Modèles valides et prêts à être exportés (XMI, TTL, etc.).

Merci de votre attention. Avez-vous des questions ?