

Présentation du Projet Web Sémantique

Niveau: 5 TWIN

Année universitaire : 2025-2026



- **Objectifs**
- **Architecture générale du projet**
- **Technologies et outils**
- **Planning détaillé**
- **Evaluation**
- **Présentation des thématiques**



Durée: 21 heures

Spécificités: Equipe: 5 - 6 membres

Maîtrise de concepts de bases du Web sémantique, Jena, développement Web

SUSTAINABLE DEVELOPMENT GOALS

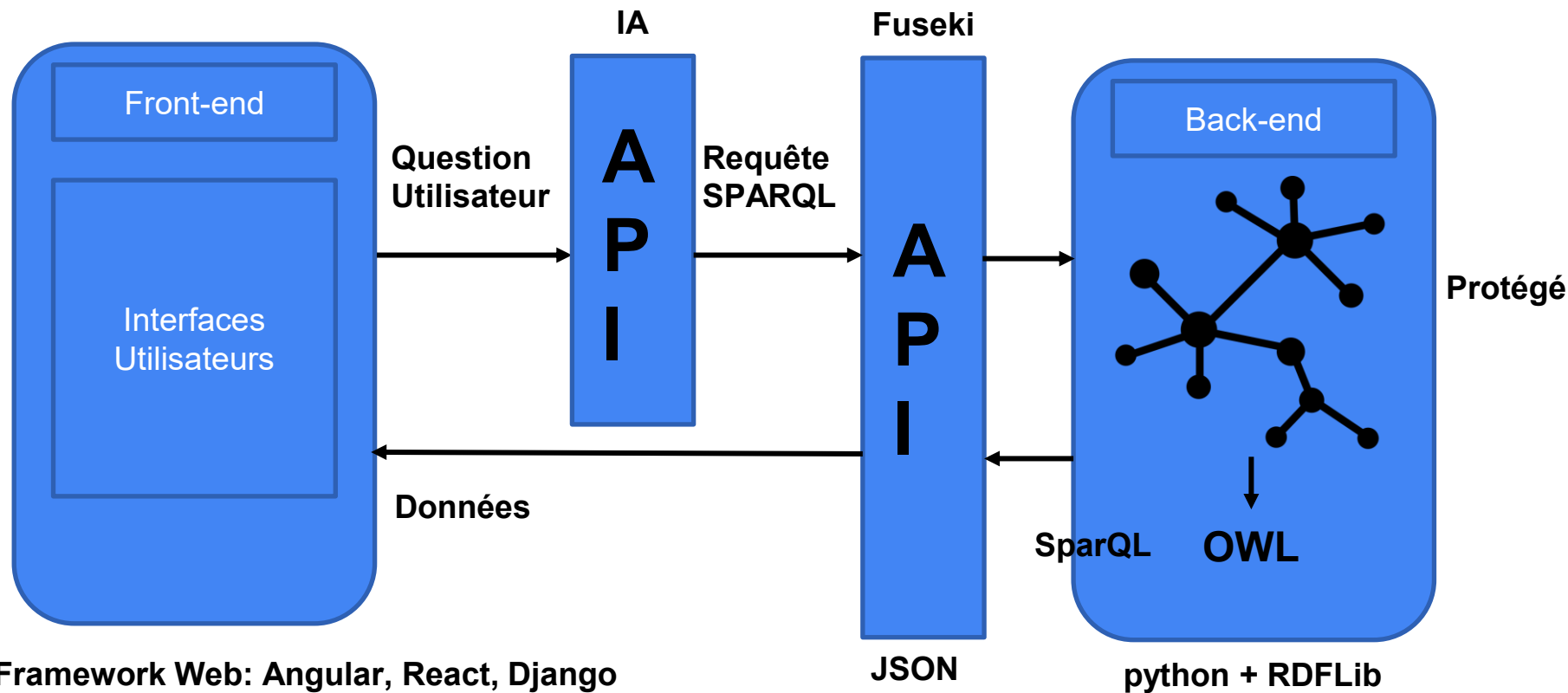




- **Modélisation de l'ontologie** : Créer une ontologie bien conçue qui représente de manière précise les concepts, les relations et les entités du domaine choisi par l'équipe. (Protégé)
- **Validation de l'ontologie**: Une fois l'ontologie construite, sa **validation** est essentielle. Nous utiliserons le **raisonneur HermiT**, intégré à Protégé, pour vérifier sa cohérence logique et garantir l'absence de contradictions.
- **Intégration de données** : Intégrer des données existantes dans l'ontologie, en les transformant en instances OWL conformes à la structure de l'ontologie. (OWL)
- **Interface utilisateur conviviale** : Concevoir une interface utilisateur conviviale qui permet aux utilisateurs de naviguer efficacement dans les données sémantiques et d'interagir avec l'application de manière intuitive. (Framework Web)



- **Recherche sémantique** : Pour la **recherche sémantique**, nous allons au-delà de la simple recherche par mots-clés. Nous utiliserons des **APIs de TALN** pour permettre aux utilisateurs de poser des questions en langage naturel. Le système se chargera alors de traduire ces questions en requêtes **SPARQL** complexes et de fournir des informations précises et pertinentes, exploitant pleinement la richesse de la sémantique de nos données.





- Fuseki est un serveur SPARQL basé sur Apache Jena qui permet de publier des données OWL sur le web et de les interroger à l'aide de requêtes SPARQL.
- Fuseki, en tant que serveur SPARQL, peut vous permettre de transformer les résultats d'une requête SPARQL en format JSON afin de les consommer plus facilement depuis une interface web ou une application.



- Voici comment vous pouvez procéder pour obtenir des résultats JSON à partir de Fuseki :
 1. **Définir une requête SPARQL** : Tout d'abord, vous définissez votre requête SPARQL qui interroge les données RDF stockées dans Fuseki. Par exemple, vous pouvez écrire une requête pour extraire des données spécifiques à partir de votre ensemble de données RDF.
 2. **Exécuter la requête SPARQL** : Vous exécutez la requête SPARQL en utilisant l'interface de requête de Fuseki ou en envoyant la requête via une requête HTTP
 3. **Obtenir les résultats au format JSON** : Pour obtenir les résultats au format JSON, vous devez configurer l'interface de requête de Fuseki pour renvoyer les résultats au format JSON.
 4. **Consommer les résultats** : Les résultats de la requête SPARQL seront renvoyés au format JSON, que vous pouvez ensuite consommer depuis votre interface web ou votre application en utilisant JavaScript, par exemple.



API d'IA pour la Transformation Question-SPARQL

- **Le Problème : Le Langage Humain vs. le Langage des Données**
Les bases de données sémantiques (comme Fuseki) ne comprennent pas le langage naturel (par exemple, "Qui a écrit ce livre ?"). Elles ont besoin de requêtes structurées, comme SPARQL.
- **L'API d'Intelligence Artificielle:** L'API d'IA agit comme un **traducteur intelligent**. Elle prend la question d'un utilisateur en langage naturel et la transforme en une requête SPARQL que le serveur peut exécuter. Vous n'avez pas besoin de créer un modèle d'IA complexe; vous utilisez une API existante.

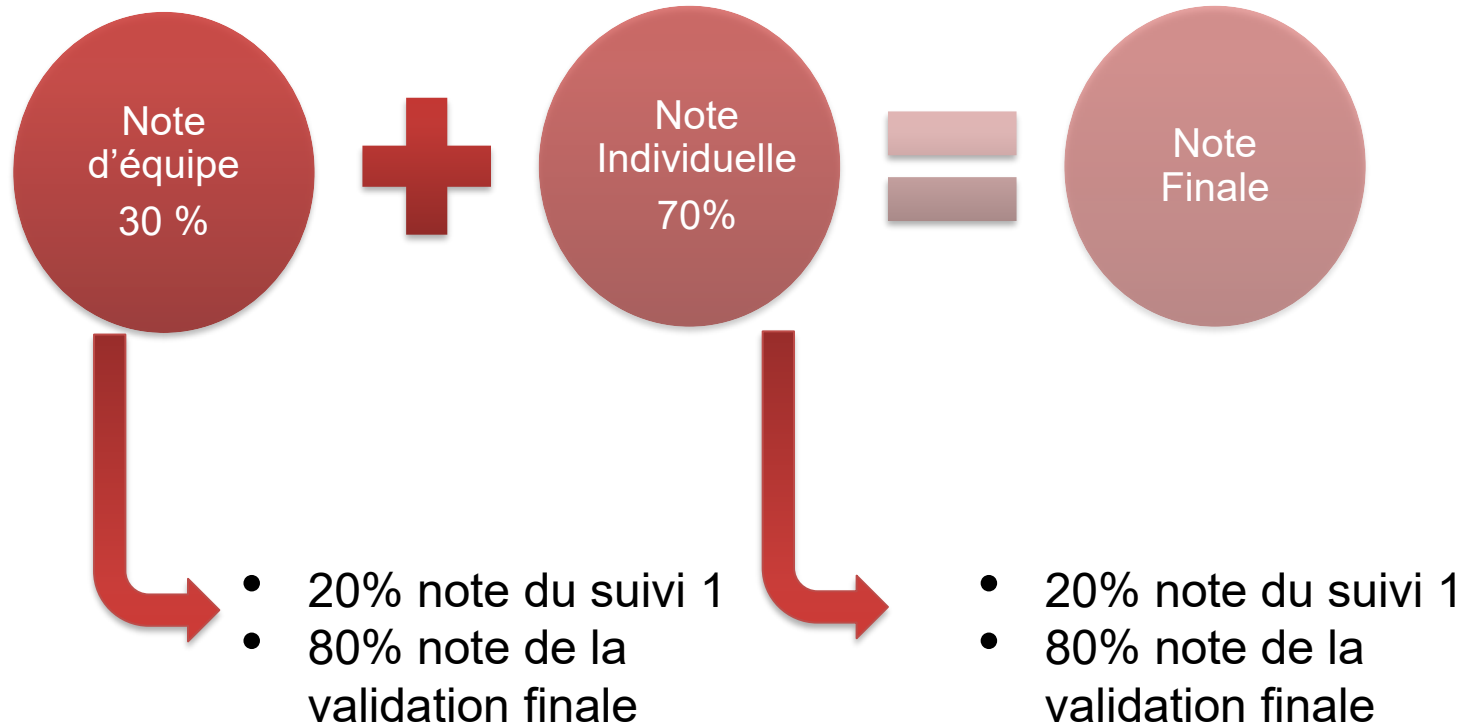


API d'IA pour la Transformation Question-SPARQL

- **SpaCy**, une bibliothèque Python de traitement du langage naturel, peut être configurée pour cette tâche. Des services cloud comme l'API de **Google Cloud AI** ou d'**OpenAI** offrent également des solutions prêtes à l'emploi.
- **Exemple**
 - **Question de l'Utilisateur:** "Qui est l'auteur du livre 'Le Petit Prince' ? »
 - **Analyse par l'API d'IA :** L'API analyse la phrase et identifie :
 - L'entité : "Le Petit Prince"
 - La relation : "auteur"
 - Le type de question : Recherche de personne ("Qui")
 - **Construction de la Requête SPARQL :** L'API utilise les informations extraites pour générer la requête SPARQL.



- Semaine 2
 - Présentation du Projet
 - Thématiques
 - Composition des équipe
 - Choix de sujet
- Semaine 4
 - Suivi noté: Validation des ontologies
- Semaine 8
 - Validation finale : Intégration finale sur une seule machine





- Si la Note Individuelle $< 10 \rightarrow$ la Note d'équipe n'est pas prise en compte dans le calcul de la note finale (Note d'équipe = Note Individuelle)
- Si la Note Individuelle - Note équipe $> 3 \rightarrow$ les pourcentages sont 60% Note individuelle et 40% Note équipe (afin de valoriser le travail en équipe)



Smart City et Mobilité :

Exemple : Plateforme sémantique pour la gestion du trafic et la promotion de la mobilité urbaine douce (vélos, transports en commun).

Gestion des Déchets :

Exemple : Application de traçabilité sémantique pour l'économie circulaire et la valorisation des déchets industriels.

Nutrition et Bien-être :

Exemple : Système de recommandation sémantique pour une alimentation durable et un mode de vie sain, adapté à la santé individuelle.

Tourisme Éco-responsable

Exemple : Développer une application de voyage sémantique qui recommande des activités touristiques locales et des hébergements qui ont une faible empreinte carbone.

Santé

Exemple : Système sémantique d'aide à la décision pour la prescription de traitements à faible impact environnemental. L'idée est de créer une ontologie qui modélise les maladies, leurs symptômes et les traitements existants, mais en y ajoutant des données sur l'impact environnemental de chaque traitement.