

Université Toulouse III - Paul Sabatier

Cahier des charges fonctionnel

SOIVD – Système optimisé d'intégration virtuelle de données

Membres du groupe :

- ERREBIAI Chaimae
- KEBIRI Issam Eddine : kebiri.isam.dine@gmail.com
- GHORBEL Ahmed Amine : ghorbelahmedaminelfet@gmail.com
- HADDOU khalid : khalidhaddou99@gmail.com
- ABAKAR Issa : issaabakar17@gmail.com

Responsable pédagogique :

- SHAOYI Yin : shaoyi.yin@irit.fr

Plan :

1. Présentation.....	1
a- Projet	1
b- Contexte	1
c- Solution proposée	1
d- Objectif	1
2. Conception	2
a. Architecture globale	2
b. Modélisation UML.....	2
c. Fonctionnalités	3
d. Bases de données.....	4
3. Développement.....	5
a. Environnement de développement.....	5
b. Implémentation de fonctionnalités	5
4. Gestion de projet	6

1. Présentation

a- Projet

Création d'un Système Optimisé d'Intégration Virtuelle De Données - SOIVD pour mieux comprendre le contexte actuel de crise d'énergies.

b- Contexte

- Évolution vers de grosses quantités de données (Big Data)
- Plusieurs sources d'informations (SGBD relationnels, fichiers, applications, pages Web ...)
- Des données de plus en plus hétérogènes et interfaces d'accès variées (langages d'interrogation, modèle de données, interfaces d'appel ...)

c- Solution proposée

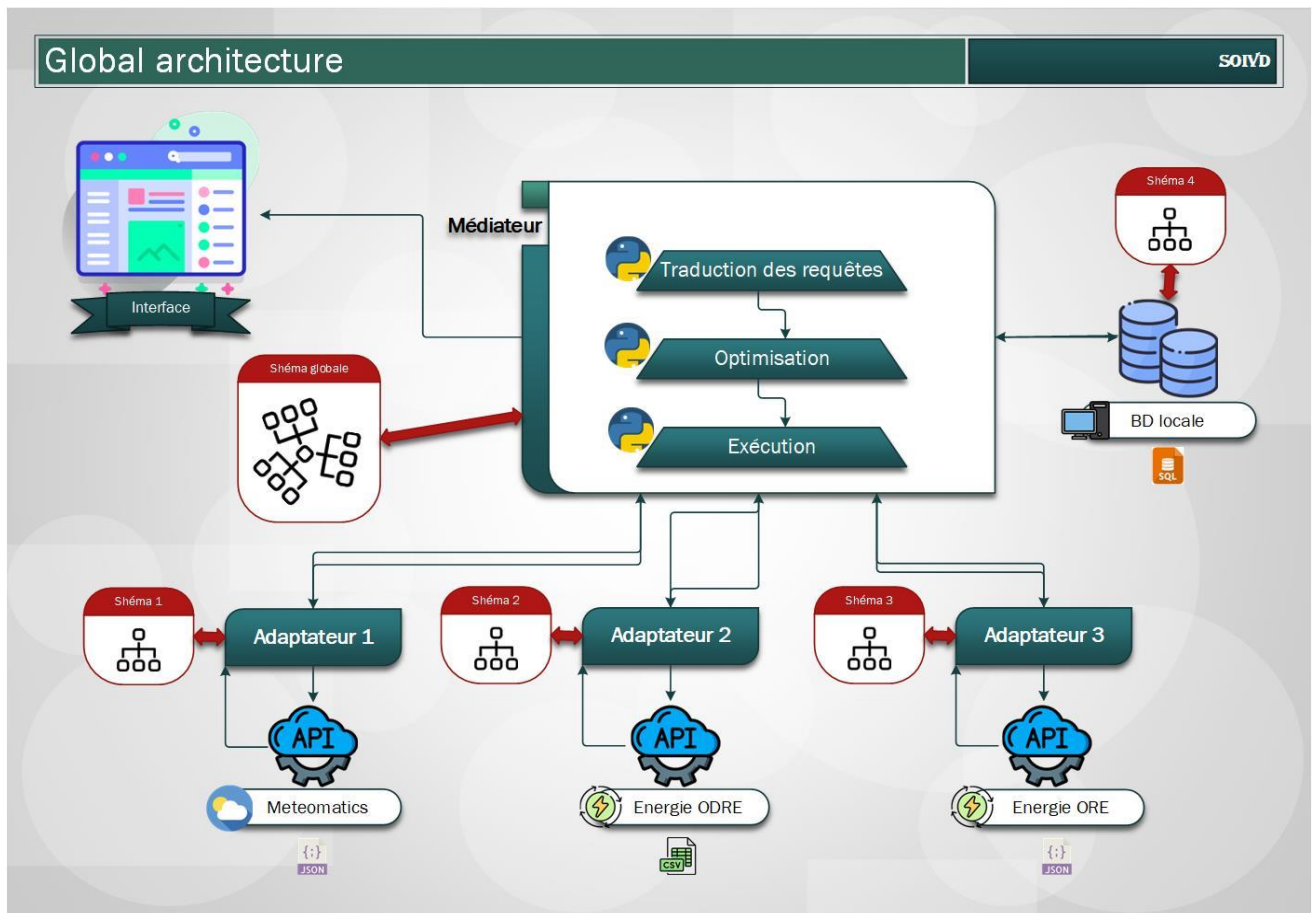
Un Système Optimisé d'Intégration Virtuelle De Données - SOIVD avec une architecture médiateur-adaptateur et une capacité d'intégration de plusieurs sources de données hétérogènes.

d- Objectif

Un accès uniforme à des sources multiples, autonomes et hétérogènes et structurés.

2. Conception

a. Architecture globale



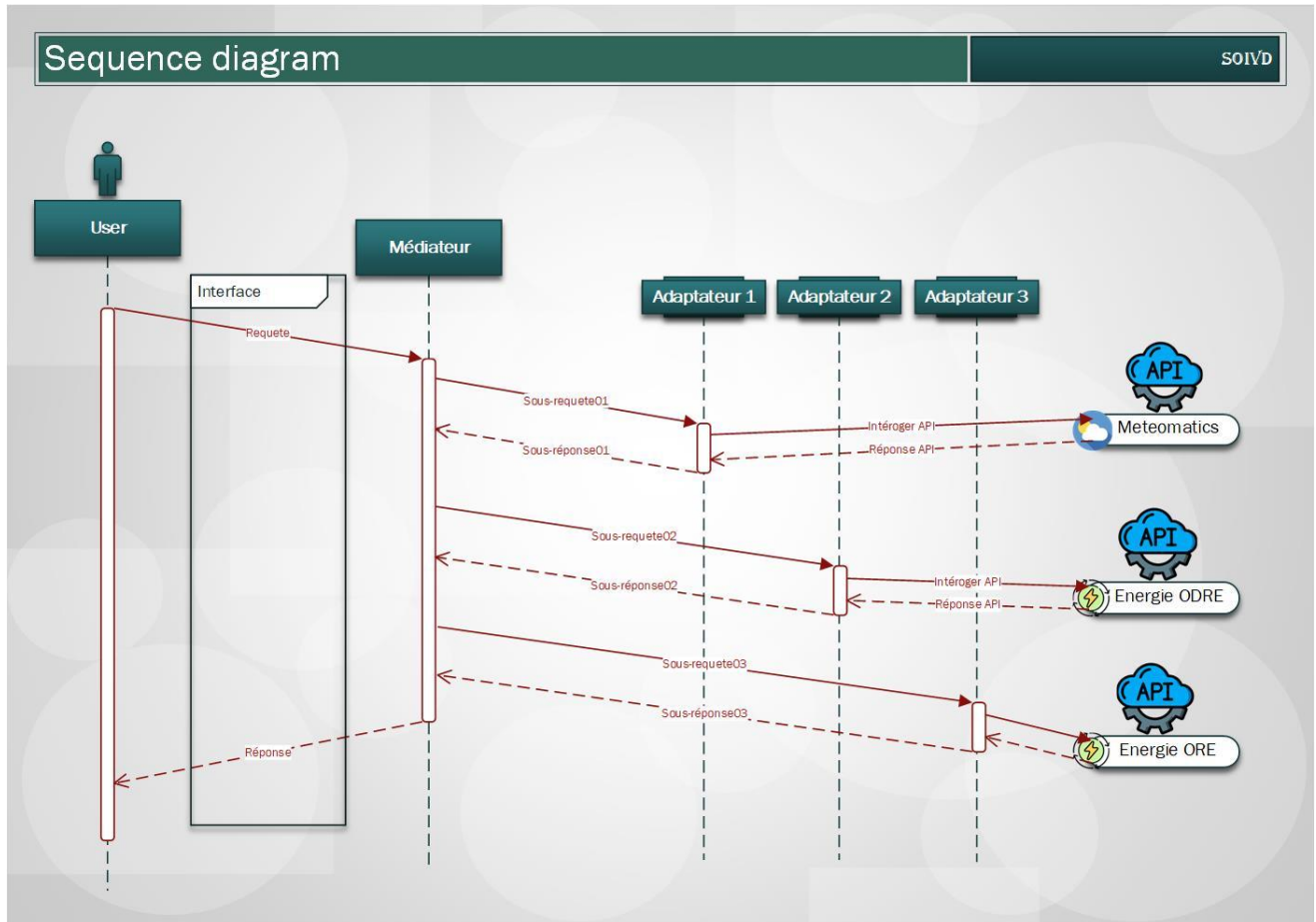
b. Modélisation UML

- Modélisation des quatre schémas de bases de données :

- BD 1 - API Meteomatics
- BD 2 - API Energie ODRE
- BD 3 - API Energie ODE

- BD 4 - BD locale

- Diagramme de séquence



c. Fonctionnalités

- Interrogation du SOIVD
 - Décomposition des requêtes en sous-requêtes
 - Traduction / mapping des sous-requêtes
 - Envois des sous-requêtes réécrites à l'adaptateur associé à chaque source
 - L'adaptateur interroge les sources de données
 - Récupérer les réponses des requêtes et les fusionner
 - Afficher les résultats dans une interface
- # Optionnel : Optimiser les requêtes des BDs

d. Bases de données

Choix des BDs

Description des BDs choisies

...

3. Développement

a. Environnement de développement

Python + framework flask

Bibliothèque : requests / JSON / contextlib.closing/ beautifulsoup/ pandas/

API

SQL + JSON + CSV

Interface : Swiger / HTML / CSS

Outils de gestion de versions et dépôt du code : Git / GitHub (gestion centralisée)

b. Implémentation de fonctionnalités

[Médiateur / adaptateur]

1- Implémenter l'interface swagger

2- Trois adaptateurs extract_from_API

4- Fonction decompose_requests

5- Fonction fusion_reponses

6- Dictionnaire mapping

7- Optimisation pandasql

4. Gestion de projet

Méthode agile SCRUM