

Récupération d'informations sur un CV

Le projet a pour but de rendre plus pertinentes les recherches d'informations des CV sur le site web Beauté Job.

L'objectif est de créer un programme permettant de récupérer les informations d'une personne ayant déposé son CV afin d'avoir un système de recherche plus performant.

Le langage PHP est choisi pour des raisons internes (meilleure expertise), mais aussi car le site Beauté Job est principalement codé en PHP, donc cela permettra de mieux intégrer le programme de recherche.

Premier temps :

Avec l'utilisation de Tesseract OCR, nous voulions récupérer le texte à partir d'un fichier JPG/JPEG ou PNG. —————>

L'extension Parcer permettait de faire le même travail que Tesseract mais pour les fichiers PDF.

Limites :

- Le texte récupéré n'est pas toujours bon (mot(s)/lettre(s) manquant(e)(s), date(s)).
- Informations inexactes.
- Réponse non structurée.
- Les fichiers ne contenant pas de texte écrit ne sont pas traités.

Second temps :

L'utilisation de l'intelligence artificielle, Mistral ai.

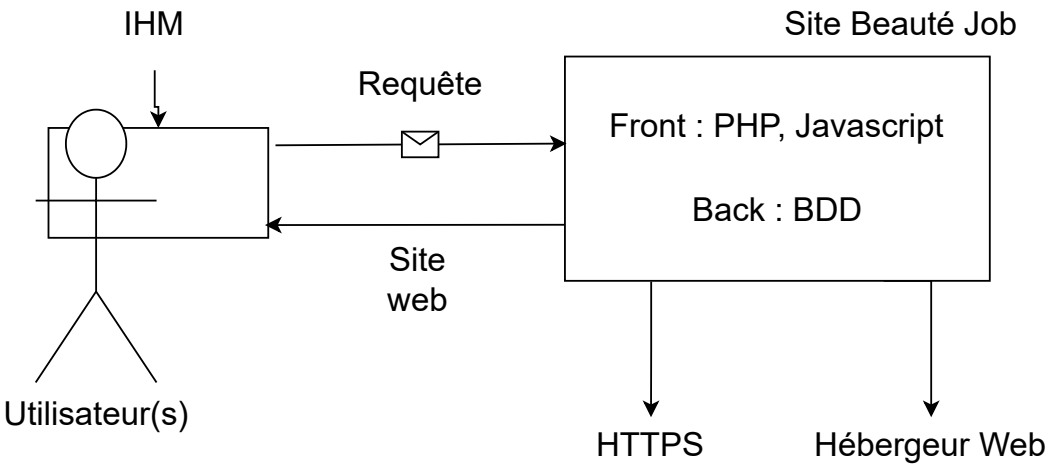
Avec l'API de Mistral, il est possible en PHP de donner un fichier, quel que soit le type de format, à Mistral pour en extraire les informations et de les mettre en forme pour avoir un affichage claire du CV déposé. —————>

Limites :

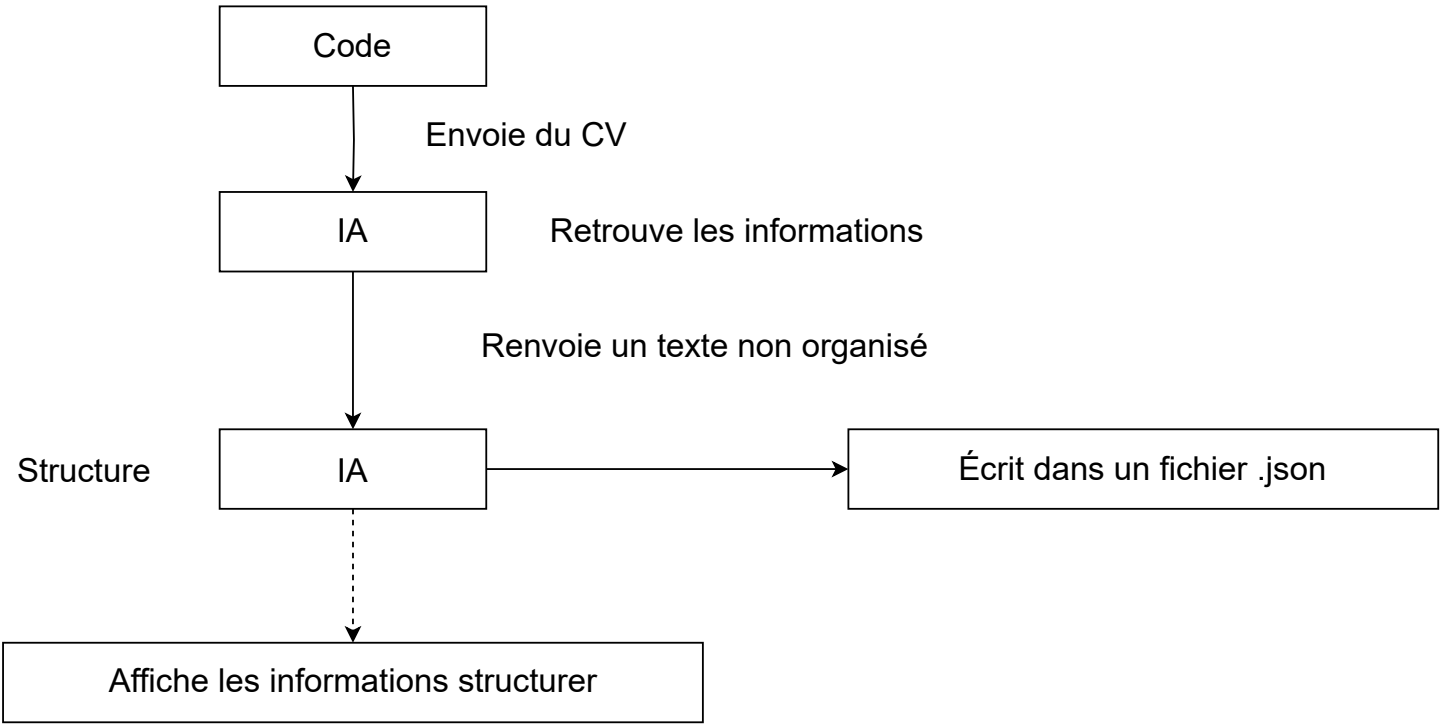
- Peut interpréter et inventer.
- Temps de réponse plus long ou variable.
- Les images peuvent ne pas être correctement traitées.
- Ne parvient pas à détecter les icônes dans les CV.

Pour les besoins de l'entreprise, il est préférable d'utiliser la solution de l'intelligence artificielle. L'utilisation de Parcer va permettre de récupérer le texte non trouvé par l'IA, pour permettre un résultat plus complet.

Schéma du site Beauté Job :



(Ancien) Schéma de l'objectif du code :



Étape 1 :

Création du code permettant d'extraire le texte d'un fichier pdf, png et jpg.

Étape 2 :

Une fois le code fonctionnel, il fallait créer une mise en page du résultat avec l'IA permettant d'avoir une mise en page propre en JSON. Pour cela, il a fallu faire la connexion avec l'API de Mistral avec du curl pour permettre la communication et l'envoi des requêtes, des fichiers et des réponses entre le code et Mistral AI

Mise au point :

Bien que ce moyen de lire, d'extraire et d'écrire les informations d'un CV, ce n'est pas le plus efficace ni le plus pertinent suite aux erreurs d'écriture et au peu d'informations réellement récupérées. De plus, le nombre de formats d'entrée est limité, il aurait donc fallu une fonction pour chaque format de fichier.

Étape 3 :

Avec l'utilisation de curl, il est possible d'envoyer un lien à Mistral pour extraire les informations importantes et les mettre en forme. Mistral utilise sa vision de qu'il a du fichier, ce qui lui permet de lire tout type de fichier et ne se limite pas à l'écriture présente ou non dans le fichier.

Étape 4 :

Pour un meilleur rendu, lisible par tous, un fichier en format JSON se crée en affichant toutes les informations dans le fichier créé et en les regroupant dans un dossier CV_archive.

Étape 5 :

En voyant que plusieurs parties de certains CV n'étaient pas prises en compte. Il a fallu recréer une fonction permettant de récupérer les informations du CV à partir de l'URL avec Parcer, pour ensuite donner le résultat à l'IA avec le prompt, le format attendu et l'URL du CV.

Nous avons le choix entre l'API de Mistral et celle de OpenAI. Après les deux codes fonctionnels, la comparaison entre les deux Intelligences Artificielles a montré que Mistral avec de meilleures performances tandis que l'API d'OpenAI était plus rapide d'environ 11 secondes.

À noter que le résultat de Parcer est enregistré temporairement, une fois que le code a fini d'être exécuté, le résultat se vide.

Schéma du fonctionnement du site web Beauté Job:

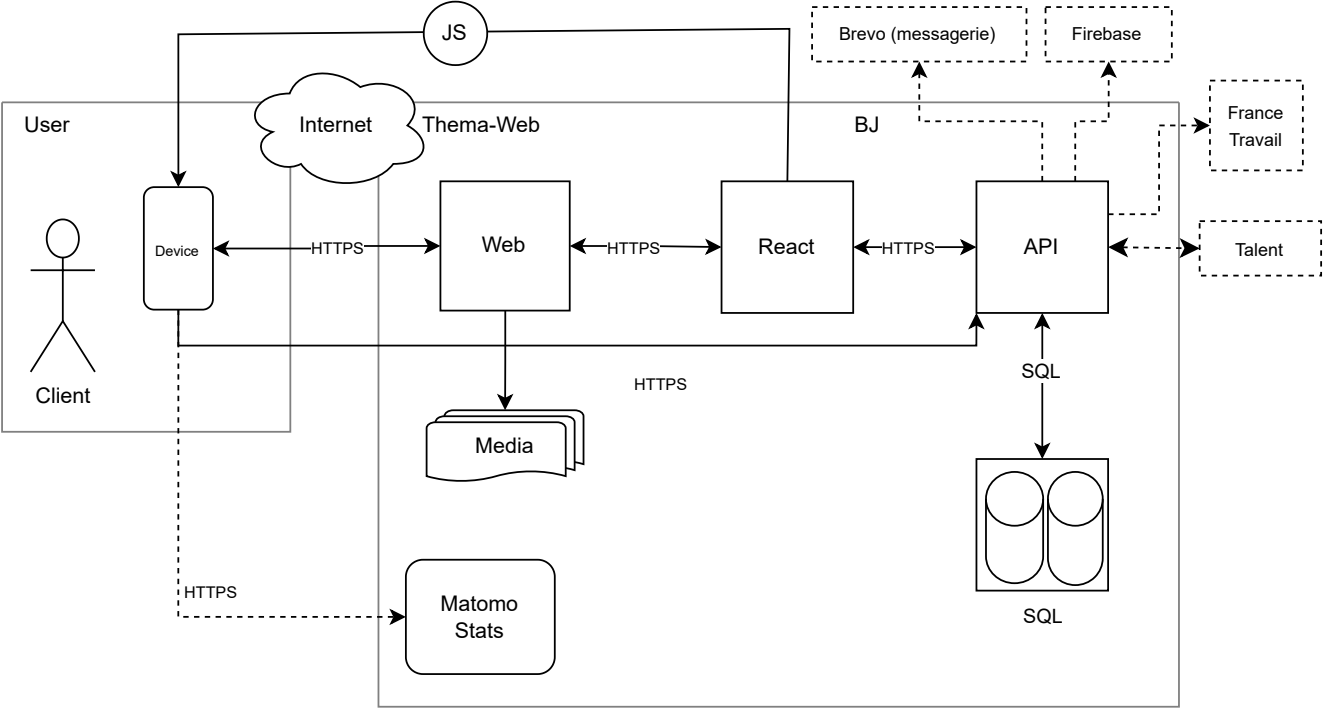


Schéma du rôle du script dans l'API:

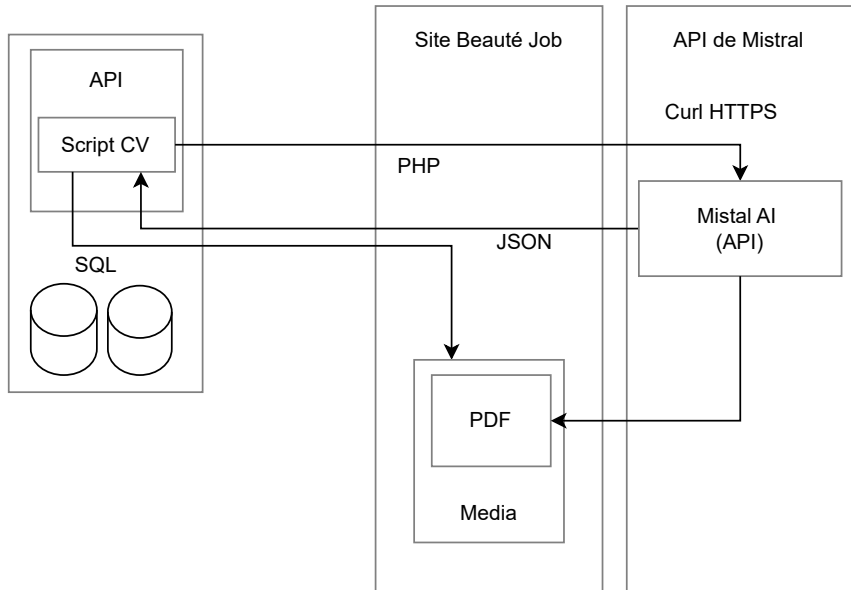


Schéma sur l'exécution du script:

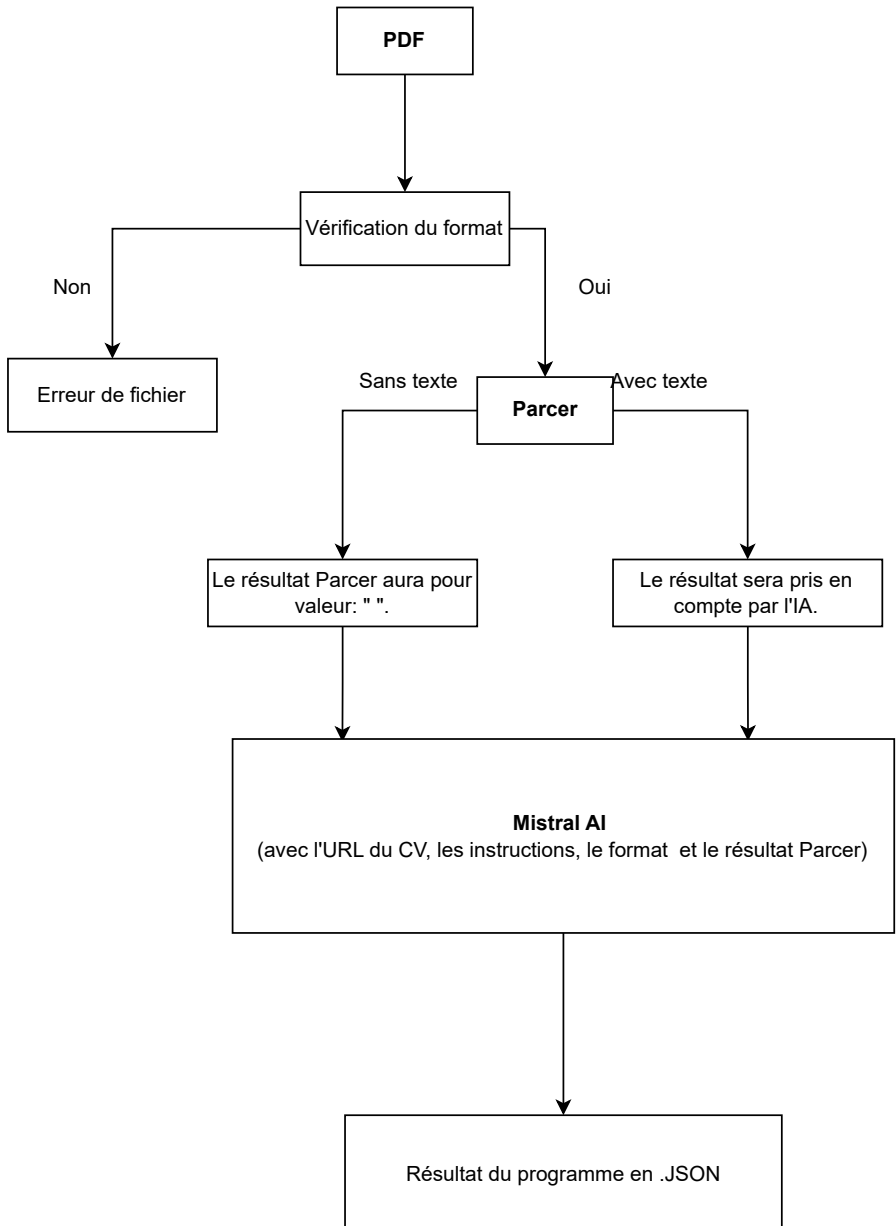


Diagramme du fonctionnement du script:

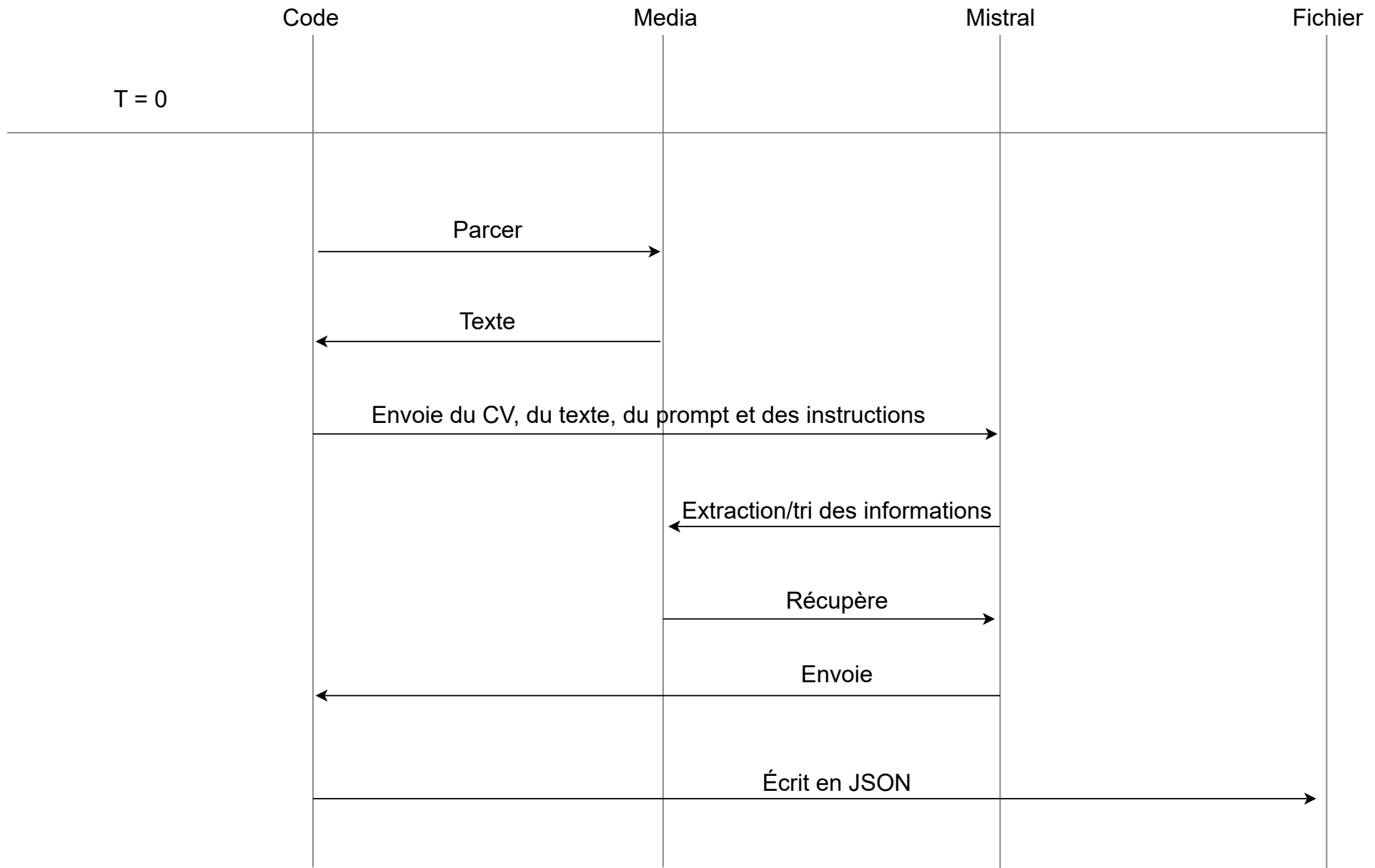
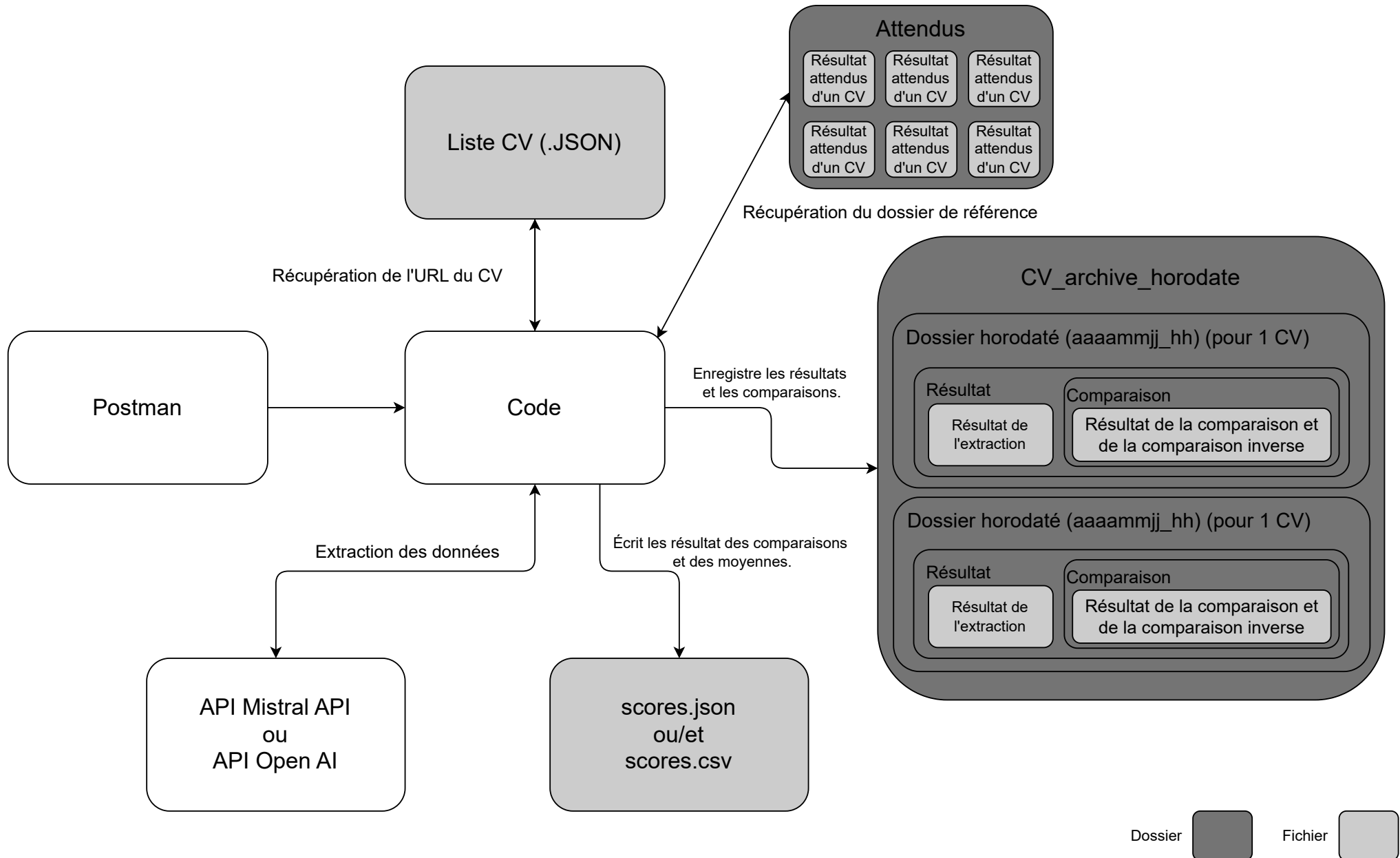


Schéma des tests automatiques (Mistral et OpenAI) :



Résultat des comparaisons entre MistralAI et OpenAI.

Tableau comparatif			
	Pourcentage de similarité moyen	Le temps d'exécution moyen	Nombre de tests
Mistral AI	87,68%	23.003 secondes	90
Open AI	50,94%	11,827 secondes	82