# I) Développement Python et DevOps

**Objectif** : Réaliser un code clair et proprement structuré. Mettre en avant les éléments considérés comme essentiels pour du code utilisable dans un environnement de production. Mettre l'accent sur vos connaissances en conception de jobs de manipulation de données ainsi que les bonnes pratiques python.

## 1. Les données

Vous avez à votre disposition les 4 fichiers de données suivants :

**drugs.csv** : contient les noms de drugs (des médicaments) avec un id (atccode) et un nom (drug)
**pubmed.csv** : contient des titres d'articles PubMed (title) associés à un journal (journal) à une date donnée (date) ainsi qu'un id (id)
**pubmed.json** : même structure que pubmed.csv mais en format JSON
**clinical_trials.csv** : contient des publications scientifiques avec un titre (scientific_title), un id (id), un journal (journal) et une date (date).

## 2. Le travail à réaliser

L'objectif est de construire une data pipeline permettant de traiter les données définies dans la partie précédente afin de générer le résultat décrit dans la partie 3.

Pour ce faire, vous devez mettre en place un projet en python organisé de la manière qui vous paraît la plus pertinente pour résoudre ce problème. Nous attendons que vous identifiiez une structure de projet et une séparation des étapes nécessaires qui permettent de mettre en évidence vos connaissances autour du développement de jobs data en python.

Il faudra essayer de considérer les hypothèses de travail suivantes :

- Certaines étapes de votre data pipeline pourraient être réutilisées par d'autres data pipelines
- Votre code doit respecter les pratiques que vous mettriez en place dans un cadre professionnel au sein d'une équipe de plusieurs personnes

Nous laissons volontairement un cadre assez libre pour voir votre manière de structurer un projet, de rédiger votre code et de mettre en place les éléments qui vous semblent essentiels dans un projet d'équipe. **Si vous n'avez pas le temps ou la possibilité de mettre en place certains aspects que vous considérez comme importants, précisez-nous ces aspects et comment vous les auriez mis en place. Ne vous sentez pas obliger de faire les nice to have pour réussir le test.** N'hésitez pas à argumenter votre proposition et les choix que vous faites si nécessaire. Il n'est pas attendu ni mesuré vos connaissances en big data sur des frameworks comme Spark, Kafka ni sur des connaissances en data science avec des solutions de NLP avancé.

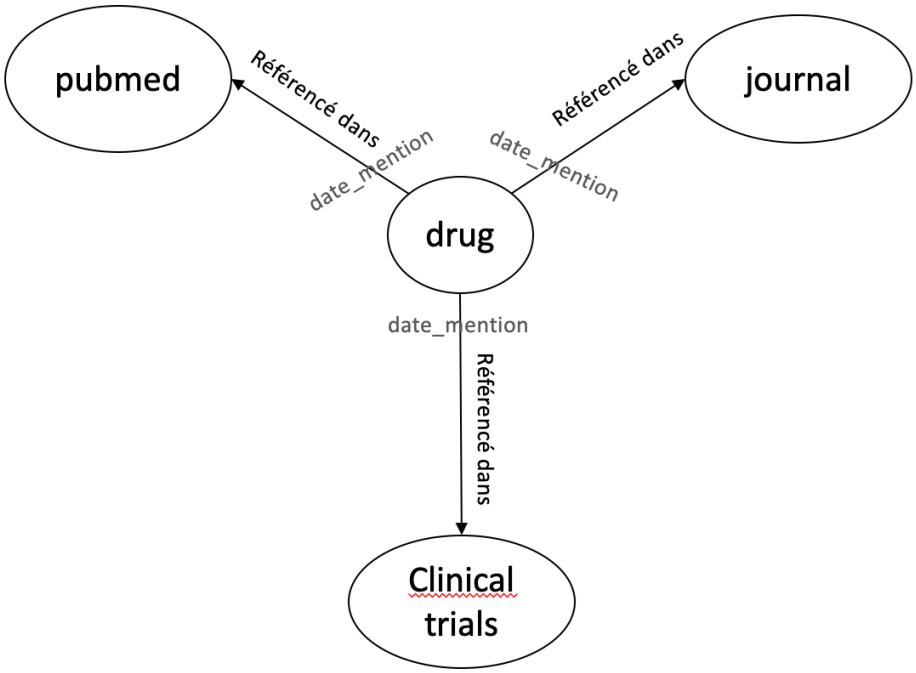
***Hint** : Vous pouvez sanitize les phrases en éliminant tous ce qui match la Regex "(\\x.{2})+"*
***Hint2** : Pour convertir une phrase en une liste de mots vous pouvez utiliser « from nltk.tokenize import RegexpTokenizer*
*RegexpTokenizer('\ \w+', gaps = True) »*
*Filtrer les stopwords de la phrase est seulement un nice to have.*
***Hint3** : Pour sanitize le JSON vous pouvez utiliser la lib JsonComment*

## 3. Data pipeline

Votre data pipeline doit produire en sortie un ou plusieurs fichiers JSON qui représente un graphe de liaison entre les différents médicaments et leurs mentions respectives dans les différentes publications PubMed, les différentes publications scientifiques et enfin les journaux avec la date associée à chacune de ces mentions. La représentation ci-dessous permet de visualiser ce qui est attendu. Il peut y avoir plusieurs manières de modéliser cet output et vous pouvez justifier votre vision :

**Règles de gestion :**

- Un drug est considéré comme mentionné dans un article PubMed ou un essai clinique s'il est mentionné dans le titre de la publication.
- Un drug est considéré comme mentionné par un journal s'il est mentionné dans une publication émise par ce journal.




## 4. API

Vous devez aussi mettre en place (hors de la data pipeline, vous pouvez considérer que c'est une partie connexe) une API permettant de répondre aux problématiques suivantes :

- Rafraichir la sortie du data pipeline suite à un call de l'API. Par exemple, le user ferait cette action suite l'ajout de nouvelles données ayant le même format. Le traitement incrémental seulement des nouvelles données est un Nice to Have.
- *Récupérer le nom du journal qui mentionne le plus de médicaments différents (Nice to Have)*

***Hint** : Pour faire l'API vous pouvez utiliser une Azure (Durable) Function. Pour rafraichir le JSON vous pouvez lancer le data pipeline grâce à la lib Python pour ACI.*

## 5. Le rendu

Vous pouvez partager votre proposition sur un repo git hébergé chez le fournisseur de votre choix (GitHub, Azure DevOps ou autre). Le code doit être capable de run sur des services Azure. L'utilisation de containers est requis.

## 6. Pour aller plus loin

Par retour de mail (ou directement sur le repo git si vous le souhaitez), vous pouvez répondre aux questions suivantes (ne nécessite pas d'implémentation dans votre projet) :

1. Quels sont les éléments à considérer pour faire évoluer votre code afin qu'il puisse être totalement industrialisé ?
2. Que devrait-on mettre autour en termes de DevOps & SRE pour avoir une production et maintenance efficace ?
3. Même question sur l'évolutivité sachant plusieurs nouveaux traitements ad-hoc et sources de données devront être ajouté ?