SERVIER

CentraleSupélec

PharmaLink – Lier publications et médicaments par l'ingénierie des données

TEST TECHNIQUE – PYTHON ET DATA ENGINEERING

Marilyne HU

Mise en situation : une démarche orientée production

Nous souhaitons construire un pipeline dédié à la production pour référencer les médicaments à partir des publications scientifiques.

1. **Les données à disposition :**
   - drugs.csv : les données médicaments
   - pubmed.csv : les publications PubMed
   - clinical_trials.csv et clinical_trials.json : les publications scientifiques en lien avec les tests cliniques
2. **Une volumétrie faible :** une quinzaine de lignes par table de données.
3. **Recherche de la modularité du code, une rédaction favorisant le travail en équipe et l'évolution vers le Big Data.**

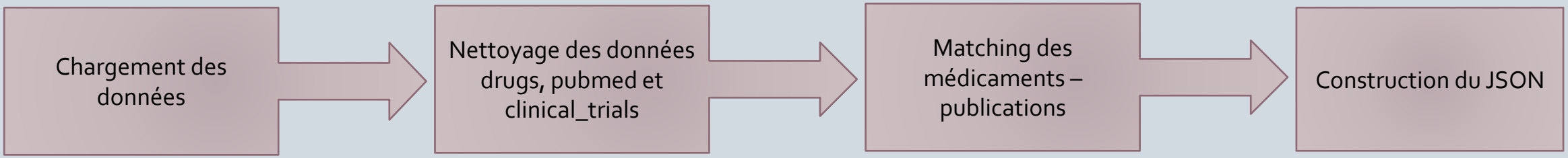
Une architecture propre : une pratique métier et modulable

```
techincal_test_servier/
│
├── __others__/        # Dossier sauvegardant les documents de présentation, etc.
│
├── raw_data/              # Données sources (CSV/JSON)
│   ├── drugs.csv
│   ├── pubmed.csv
│   ├── pubmed.json
│   └── clinical_trials.csv
│
├── output_data/           # Dossier de sortie du fichier JSON final
│   └── drug_output.json
│
├── src/                   # Code source modulaire
│   ├── __init__.py
│   ├── loader.py          # Fonctions de chargement des fichiers
│   ├── processor.py       # Matching entre médicaments et publications
│   ├── graph_builder.py   # Construction du graphe final et sauvegarde
│   └── utils.py           # Fonctions utilitaires (nettoyage, encodage, etc.)
│
├── dags/                    # Dossier spécial pour Airflow
│   ├── __init__.py
│   └── dag_drug_pipeline.py  # Le DAG Airflow
│
├── main.py                # Script principal orchestrant toutes les étapes
│
├── journal_insight.py     # Traitement ad-hoc
│
├── environment.yml       # Dépendances du projet
│
└── README.md              # Documentation du projet
```

1. Un répertoire de données brut : *raw_data/*
2. Une boîte à outils de fonctions sources permettant un code modulable : *src/*
3. Une recette de cuisine bien rédigée et organiser : *main.py*
4. Un répertoire de données de sortie : *output_data/*
5. Automatisation de notre recette de cuisine comme la gestion d'erreur, des dépendances et de la flexibilité : *dags/*
6. Un traitement ad-hoc testant l'output de notre pipeline : *journal_insight.py*

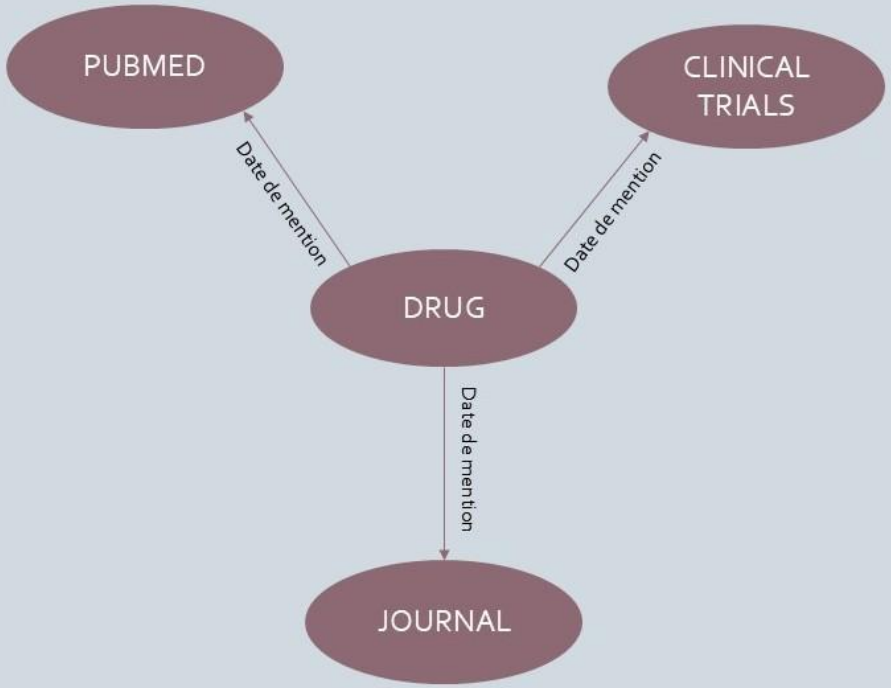
Une recette linéaire facilitant un DAG : *main.py*

1. Chargement des données : *load_data()*
2. Nettoyage des données de publications de test clinique : *clean_clinicals_trials()*
3. Nettoyage des données médicaments : *clean_drugs()*
4. Nettoyage des données d'article PubMed : *clean_pubmed()*
5. Matching des médicaments avec les publications qui le mentionne : *get_links()*
6. Construction du JSON final : *build_json()*

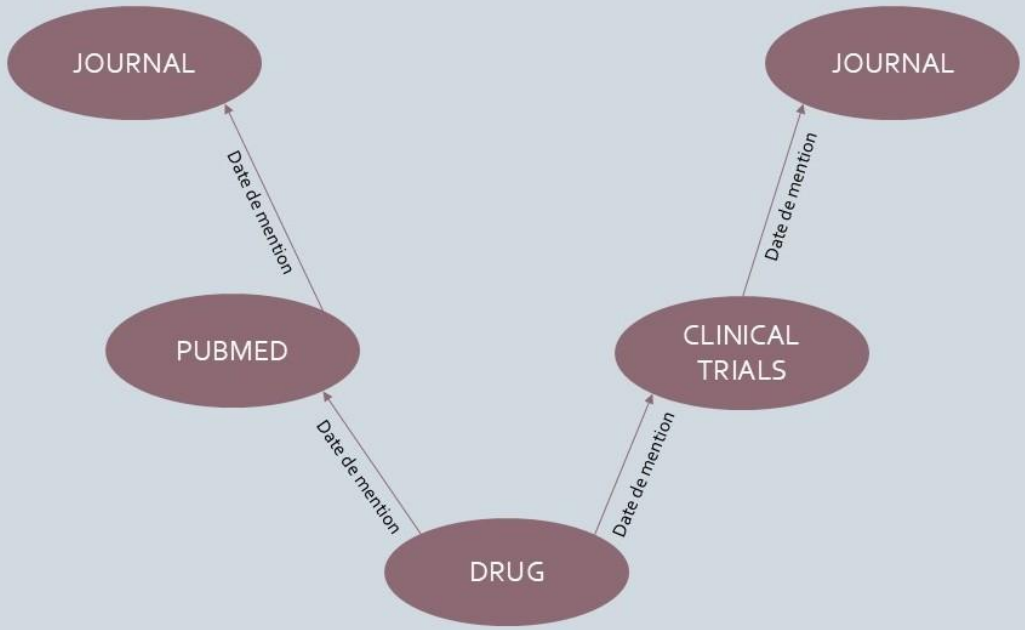


Deux structures de liaison entre les médicaments et les publications : le référencement du journal

1. LIEN DIRECTE AVEC LE MÉDICAMENT



2. LIEN INDIRECTE AVEC LE MÉDICAMENT



Donnant à deux structures de JSON différentes ...

Traitement ad-hoc : recherche du journal mentionnant le plus de médicaments

1. Un traitement indépendant du pipeline.
2. Une fonction robuste permettant de trouver le journal mentionnant le plus de médicaments différents : adapté au graphe de liaison directe et indirecte
3. Le journal « Journal of Emergency Nursing » mentionne le plus de médicaments.
4. Les médicaments mentionnés dans le journal : diphenhydramine et epinephrine

Une évolution vers le Big Data : création d'un DAG

FORME ET MANIPULATION DES DONNÉES INPUT / OUTPUT

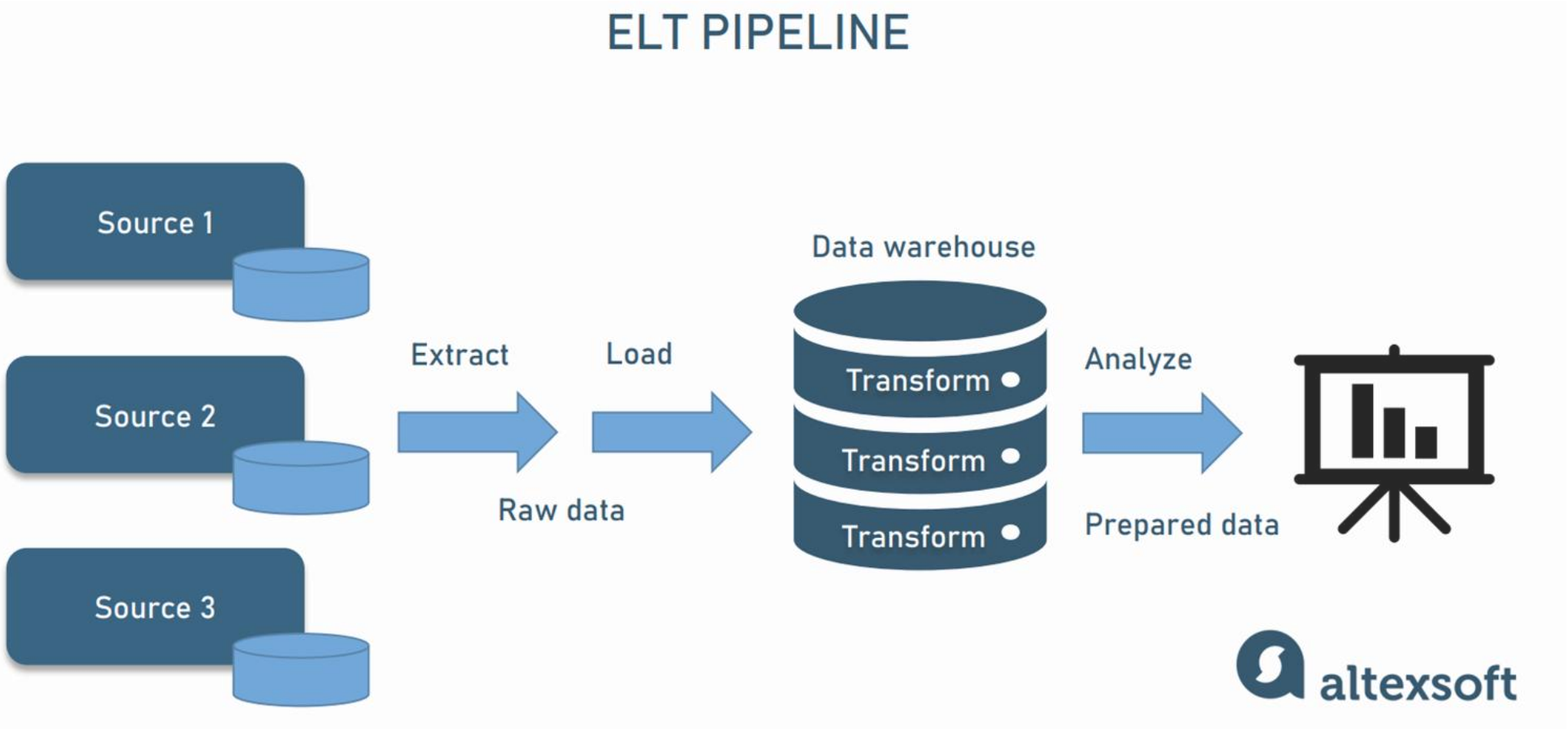
	Petite volumétrie	Grande volumétrie
Format des données	CSV et JSON	Parquet
Librairie de traitement des données	Pandas	Dask

Une combinaison essentielle entre le format Parquet et la librairie Dask :

- Structure de code similaire à Pandas ;
- Temps de lecture optimisé ;
- Traitement parallèle des données.

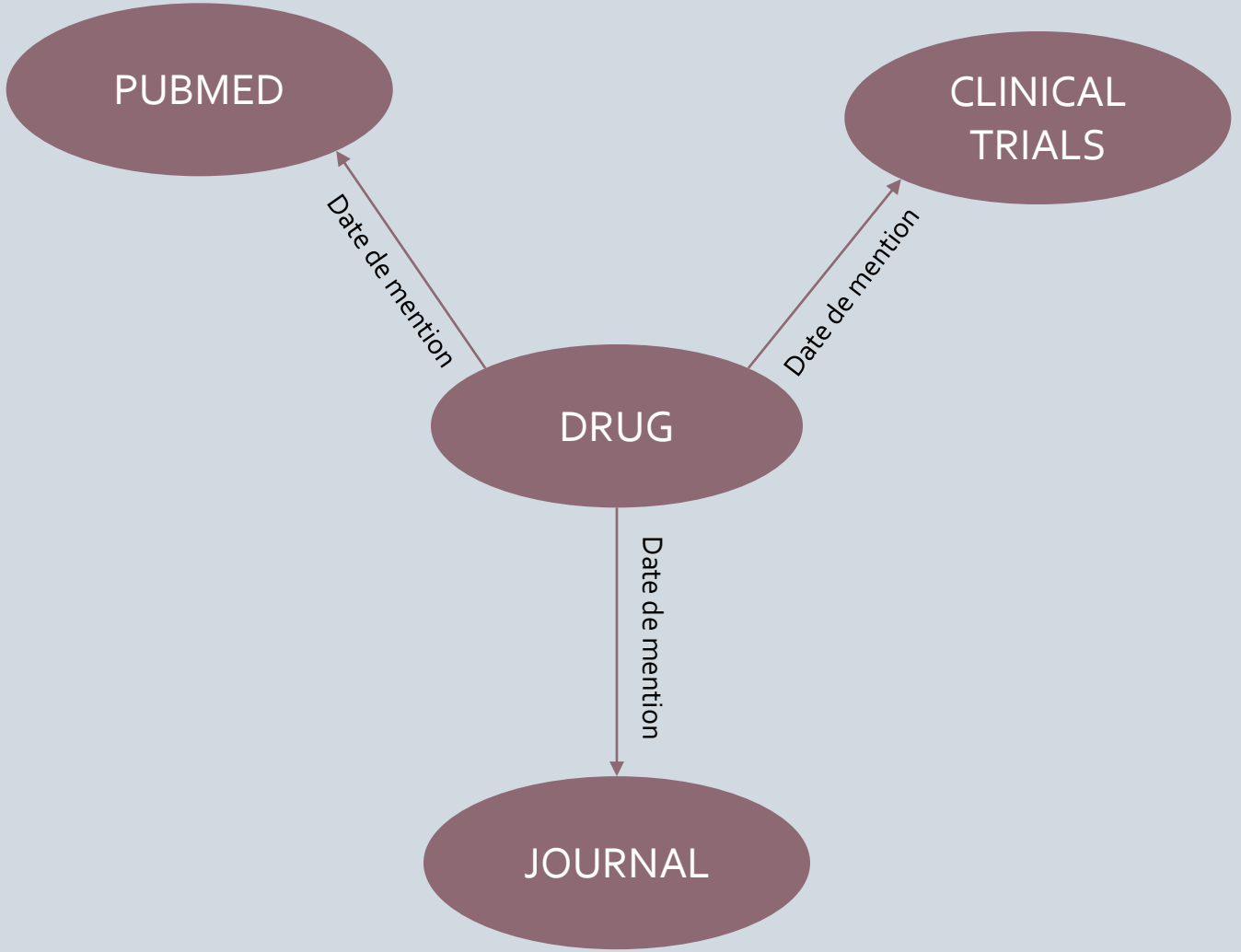
UNE AUTOMATISATION DU PIPELINE : DAG - AIRFLOW

1. Création d'un DAG en utilisant Airflow
2. Plus de flexibilité
3. Relance indépendante des tâches lors des erreurs
4. Une exécution du pipeline automatique

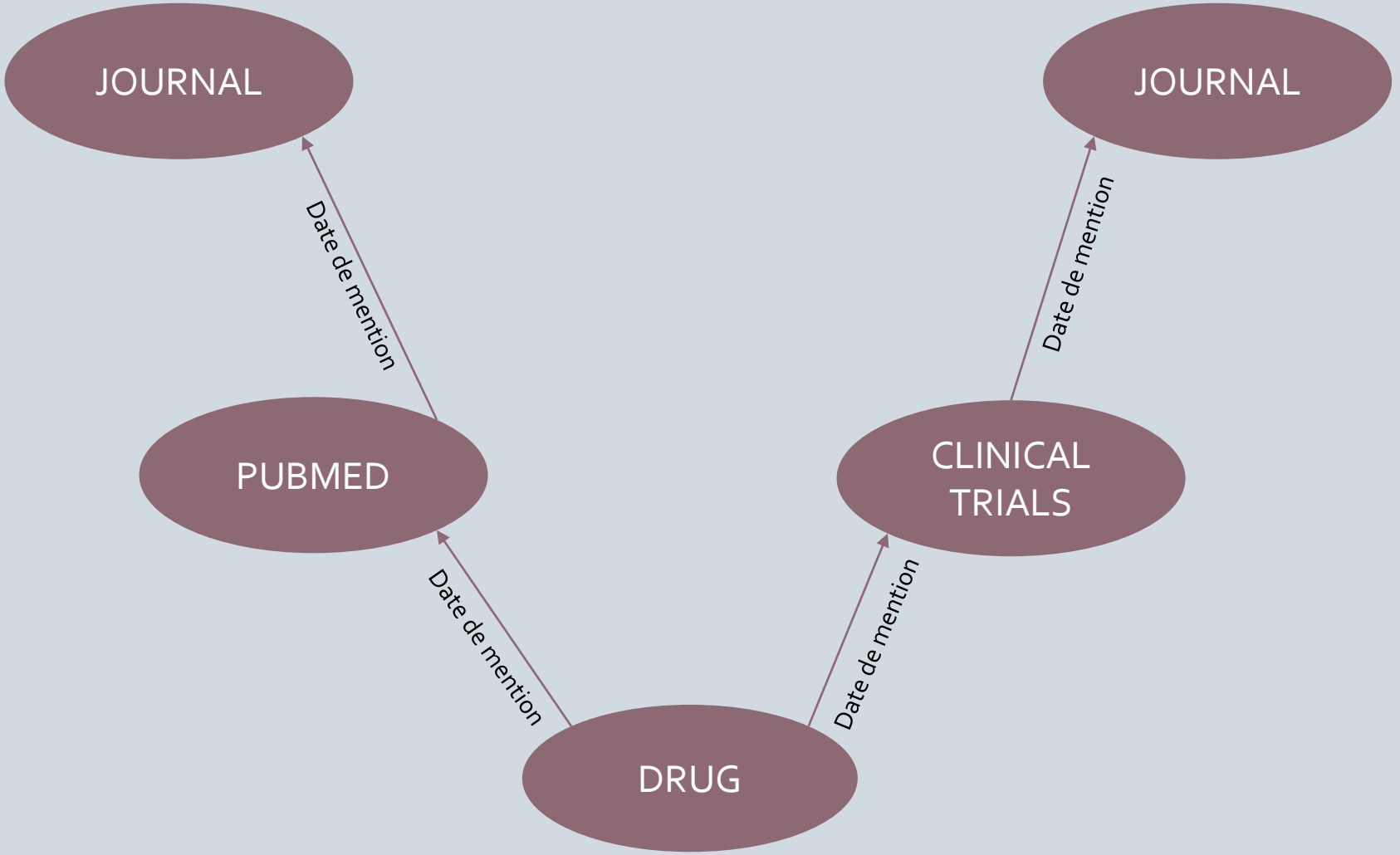

ELT PIPELINE
Source 1
Source 2
Source 3
Extract
Load
Raw data
Data warehouse
Transform
Transform
Transform
Analyze
Prepared data
altexsoft

Merci !

ANNEXE 1


PUBMED
CLINICAL TRIALS
Date de mention
Date de mention
DRUG
Date de mention
JOURNAL

ANNEXE 2


JOURNAL
JOURNAL
Date de mention
Date de mention
PUBMED
CLINICAL TRIALS
Date de mention
Date de mention
DRUG