Miscellanea

LES PRINCIPES FONDAMENTAUX DE LA SCIENCE DES DONNÉES

11. L’ingénierie des données

Contexte

L'un des défis de la science des données : mettre de grandes quantités de données dans des formats pouvant être **lus** par des algorithmes.

L'**ingénierie des données** est liée au traitement de ces données.

Après le traitement, les scientifiques des données développent des **preuves de concept** ; les ingénieurs IA/AA les traduisent en **modèles déployables**.

L'ingénierie des données existe depuis un certain ; avec l'essor du "**cloud computing**", l'expertise dans ce domaine devient aussi recherchée que celle en analyse de données (du moins, dans certains cercles).

Rôles et responsabilités (reprise)

Ingénieurs en données (ID)

- recevoir des données d'une source
- structurer, distribuer et stocker les données dans des lacs et des entrepôts de données
- créer des outils et des modèles de données que les SD utilisent

Ingénieurs AA

- déployer de modèles de données
- combler les écarts entre ID et SD
- faire passer des idées de validation de concept à grande échelle

Scientifiques des données

- recevoir des données procurées/fournies par l'ID
- extraire la valeur des données
- construire des modèles prédictifs de preuve de concept
- mesurer et améliorer les résultats
- construire des modèles analytiques

Rôles et responsabilités (reprise)

Dans les petites organisations, l'ingénierie et la science des données sont généralement **regroupées** dans sous un même toit.

Les grandes entreprises disposent d'ingénieurs de données **spécialisés**, qui construisent des **pipelines de données** et gèrent des **entrepôts de données** (en les alimentant en données et en créant des schémas de table pour assurer le suivi des données stockées).

En général, ID ≠ SD.

Les pipelines de données

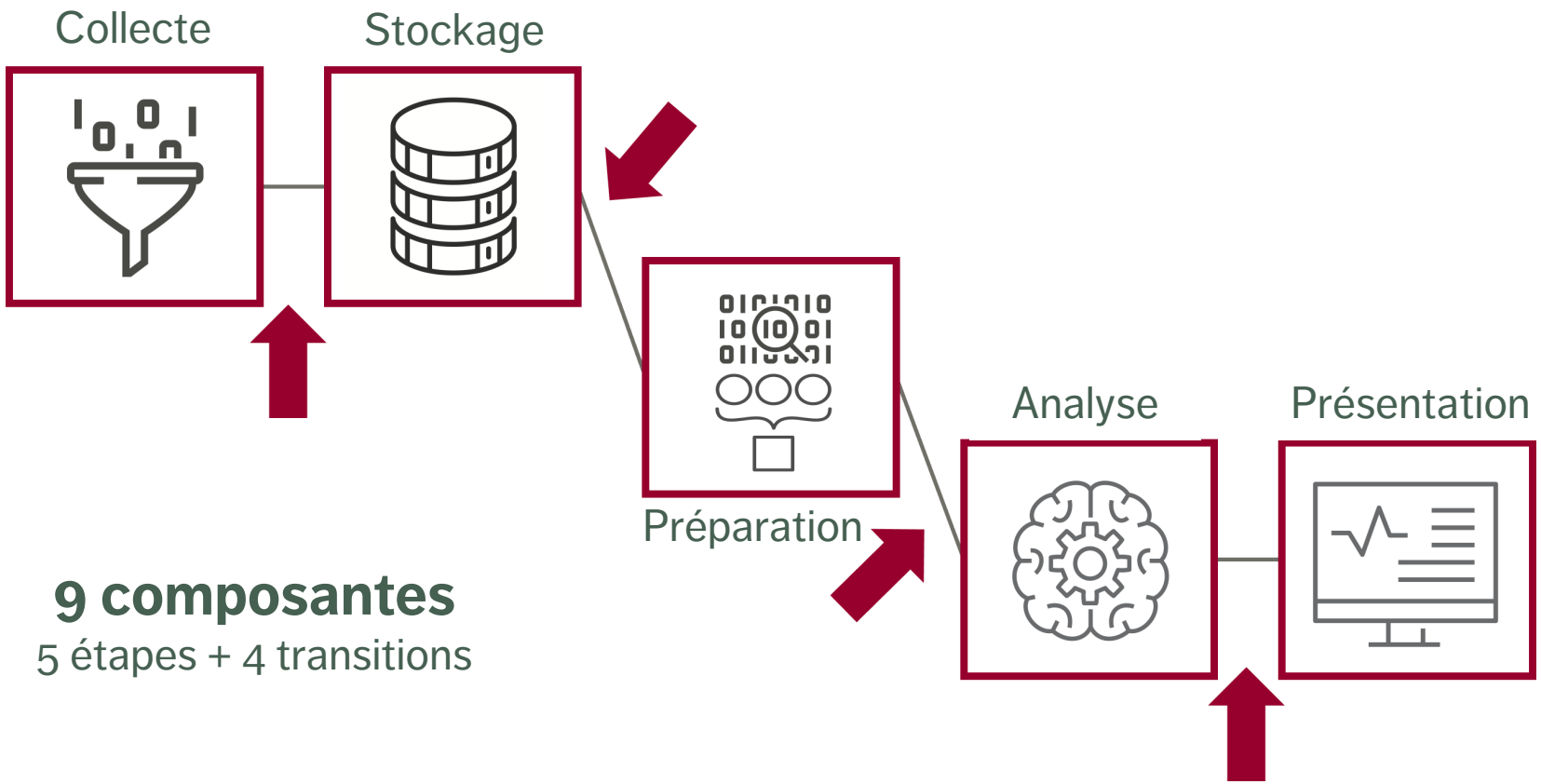
Ingénierie des **données**

- les opérations qui créent des **interfaces** et des **mécanismes** pour le flux/l'accès à l'information
- mise en place d'une **infrastructure de données**, préparation des données pour une analyse plus poussée par des SD

Les données peuvent provenir de nombreuses **sources** (et types de sources), et dans une variété de formats et de tailles.

Transformer tout cela en un processus que les SD peuvent utiliser et dont ils peuvent tirer du sens est connu sous le nom de **construction d'un pipeline de données**.

Les pipelines de données



9 composantes

5 étapes + 4 transitions

Les pipelines de données

Principal défi en matière d'ingénierie des données :

- construire un pipeline qui **s'exécute en temps réel** (ou presque) **à chaque fois qu'il est sollicité**
- afin que les utilisateurs obtiennent des **informations actualisées** avec des **délais minimaux**

Les pipelines conceptuels sont transmis aux ingénieurs AA pour le **déploiement** et la **production**. Certains des travaux entourant cette tâche comprennent :

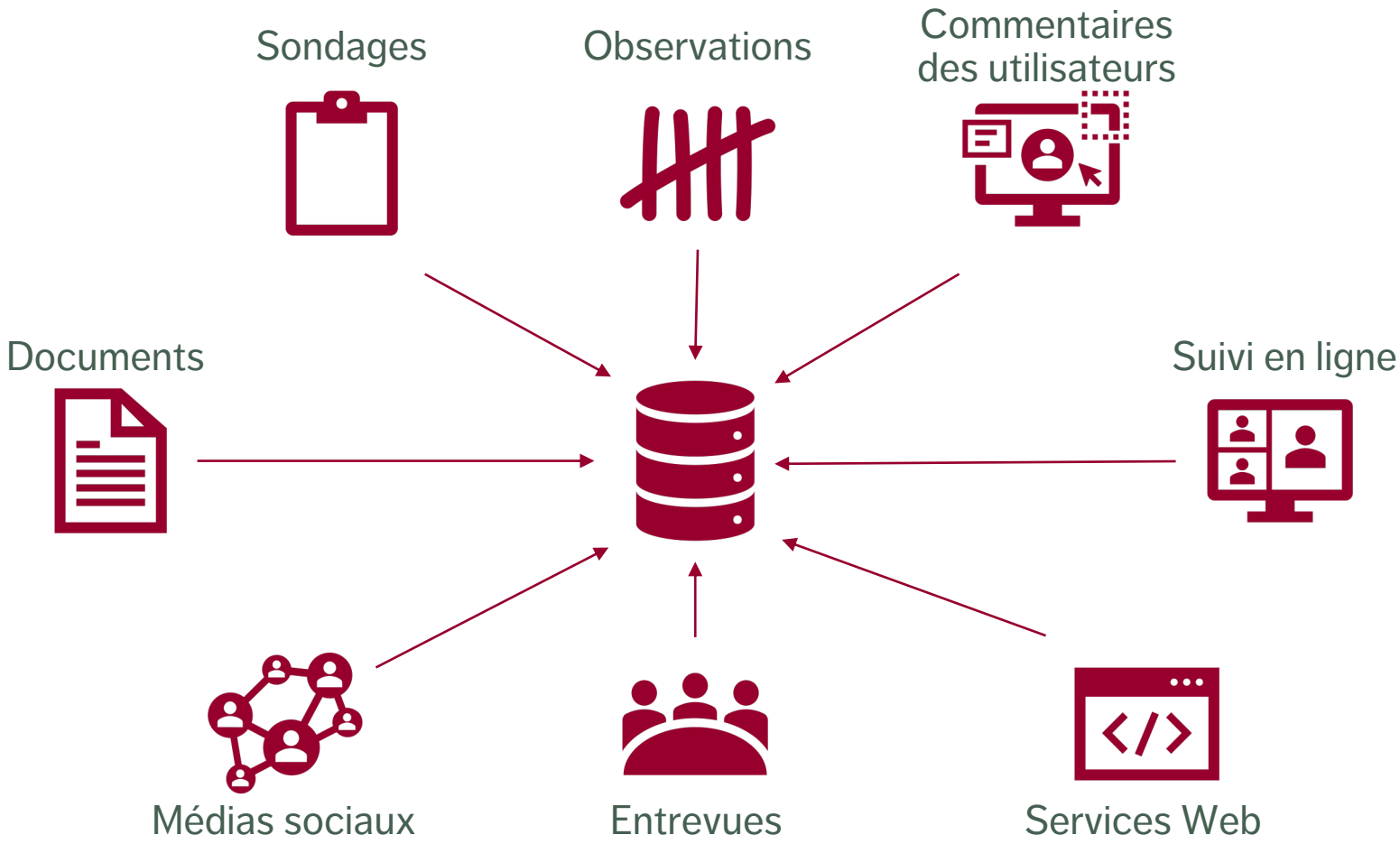
- contrôles de la qualité des données
- optimisation de la performance des requêtes
- la création d'un écosystème d'intégration/livraison continue pour les changements de modèles
- ingéstion des données provenant de diverses sources dans le modèle de données
- transfert des techniques d'AA et de SD aux systèmes distribués

Les pipelines de données

Thèmes communs (opérations/framework/tâches/sources) pour les étapes du pipeline :

- **collecte de données :** applications, applications mobiles, microservices, dispositifs de l'Internet des objets (IoT), sites web, instrumentation, journalisation, capteurs, données externes, contenu généré par l'utilisateur, etc.
- **le stockage des données :** Gestion des données de référence (MDM), entrepôt, lac de données, etc.
- **intégration/préparation des données :** ETL, intégration de données en flux, etc.
- **analyse des données :** apprentissage automatique, analyse prédictive, tests A/B, expériences, intelligence artificielle (IA), apprentissage profond, etc.
- **livraison et présentations :** tableaux de bord, rapports, microservices, notifications push, email, SMS, etc.

La collecte de données



ETC – Extraire

Extraire | Transformer | Charger

ETC – Transformer

Structure

Types de données

Agrégation

Nettoyage

Rejoindre

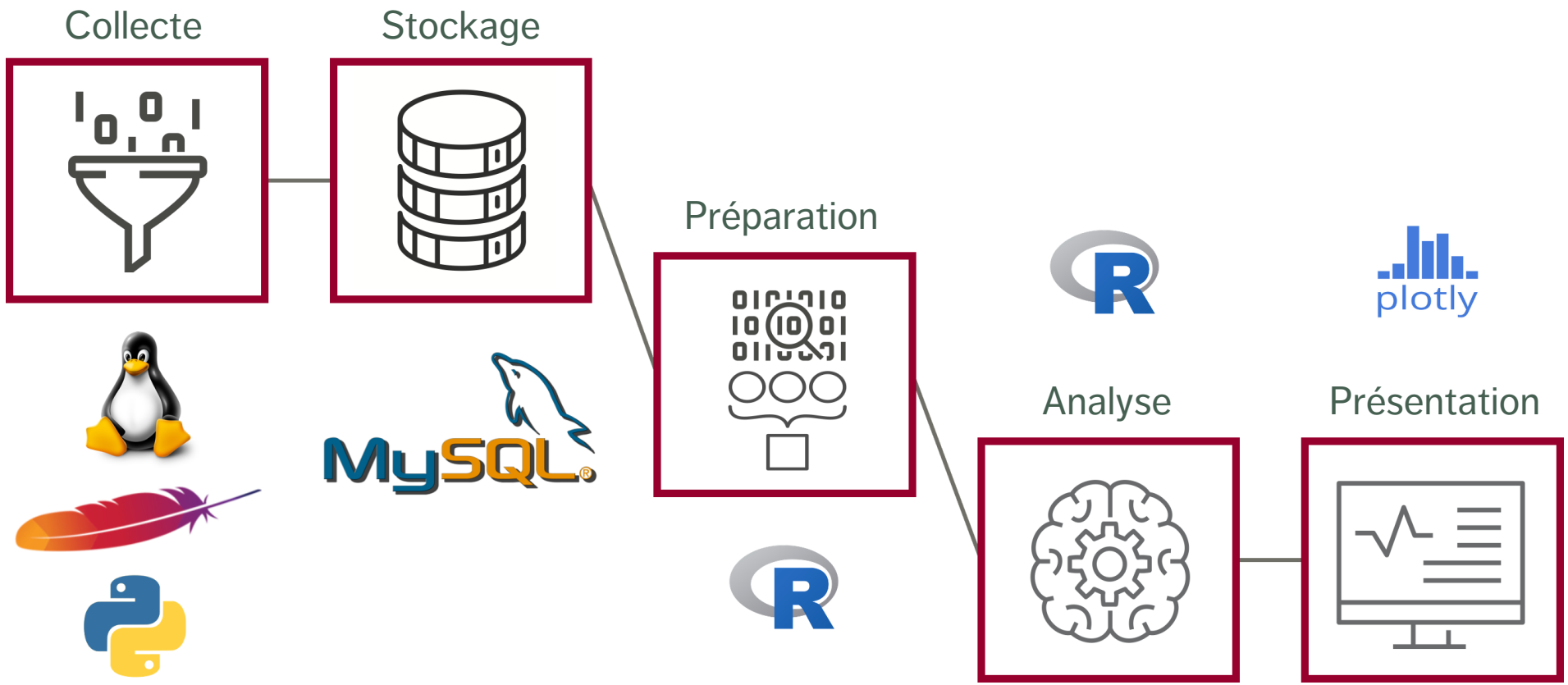
Regroupement

Extraire | **Transformer** | Charger

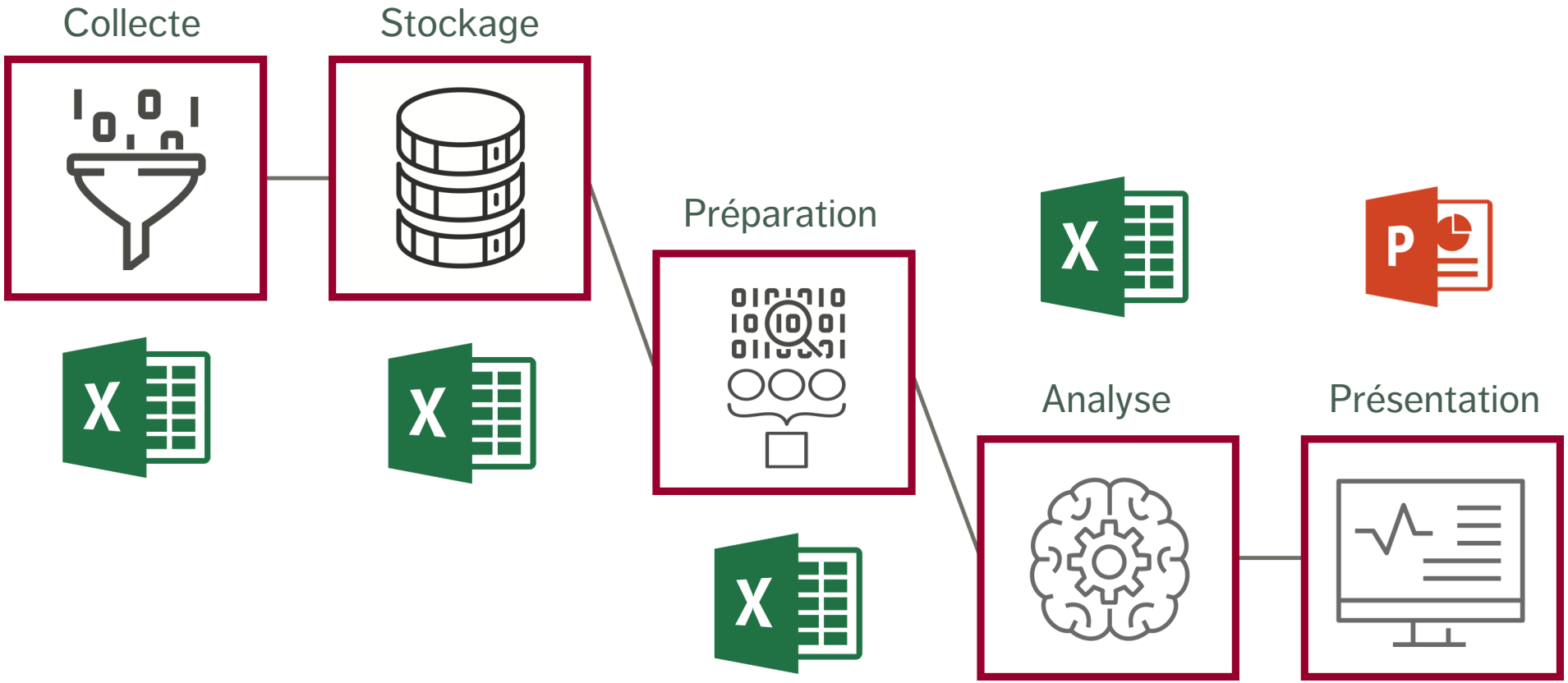
ETC – Transformer

Extraire | Transformer | **Charger**

Un pipeline de données “Open Source”



Un pipeline de données GdC (?)



Les outils de pipelines de données

Les pipelines permettent aux utilisateurs de diviser les tâches importantes en une série de petites étapes séquentielles, ce qui peut aider à **optimiser** chaque étape.

E.g., si vous utilisez TensorFlow pour la composante d'analyse d'un pipeline DL qui consiste en un seul grand script, **tout**, de la collecte des données à la présentation, doit utiliser TensorFlow, ce qui peut ne pas être optimal.

Les outils de pipeline de données sélectionnent le meilleur cadre/ langage pour chaque composante/ tâche du pipeline :

- Luigi (Spotify)
- Airflow (AirBnB)
- scikit-learn
- pandas/tidyverse
- etc.

Les outils d'ingénierie des données

Il est peu probable qu'un ID puisse maîtriser tous les outils d'ingénierie de données possibles, mais les équipes ID ont une plus grande **couverture :**

- **bases de données analytiques** (Big Query, Redshift, Synapse, etc.)
- **ETC** (Spark, Databricks, DataFlow, DataPrep, etc.)
- **moteurs de calcul évolutifs** (GKE, AKS, EC2, DataProc, etc.)
- **orchestration de processus** (AirFlow/Cloud Composer, Bat, Azure Data Factory, etc.)
- **déploiement et mise à l'échelle de plateforme** (Terraform, outils personnalisés, etc.)
- **outils de visualisation** (Power BI, Tableau, Google Data Studio, D3.js, ggplot2, etc.)
- **programmation** (tidyverse, numpy, pandas, matplotlib, scikit-learn, scipy, Spark, Scala, Java, SQL, T-SQL, H-SQL, PL/SQL, etc.)

La gouvernance des données

La gouvernance des données englobe :

- les **personnes** ;
- les **processus**, et
- les **technologie de l'information**

Onl'utilise pour créer un traitement **cohérent/approprié** des données d'une organisation à travers l'entreprise.

Elle fournit la base, la stratégie, et la structure pour garantir que les données sont gérées comme un **actif** et transformées en informations **significatives**.

La gouv. des données

Objectifs :

- création d'une culture de données libre service
- établir des règles internes pour leur utilisation
- mettre en œuvre les exigences de conformité
- améliorer les communications
- augmenter la valeur des données
- réduire les coûts associés aux données
- gérer continuellement les risques
- assurer une existence continue

Lectures conseillées

L'ingénierie des données

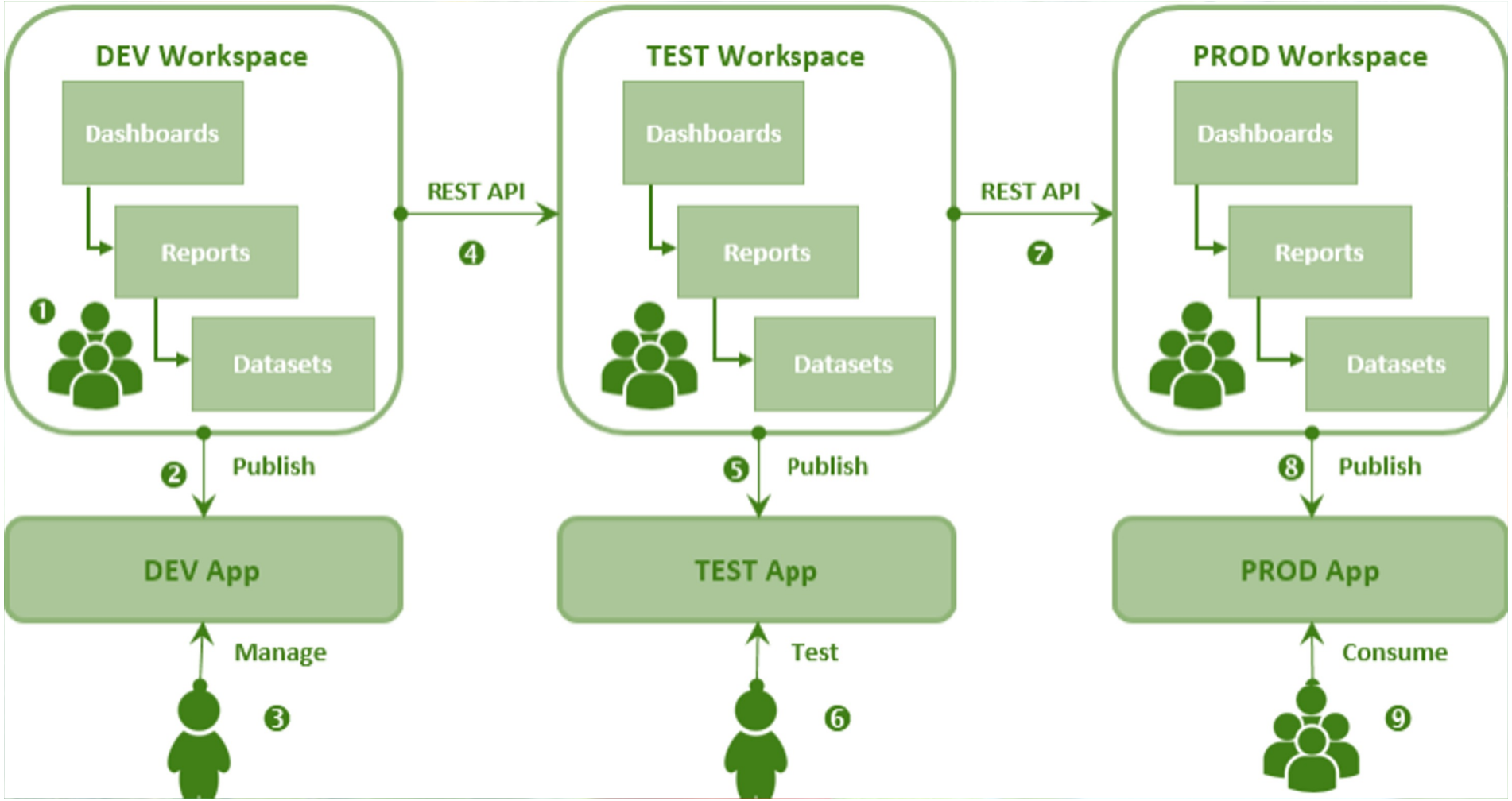
Data Understanding, Data Analysis, Data Science
Volume 2: Fundamentals of Data Insight

17. Data Engineering and Data Management
    17.1 Background and Context
    17.2 Data Engineering
    - Data Pipelines
    - Automatic Deployment and Operations
    - Scheduled Pipelines and Workflows
    - Data Engineering Tools

Exercices

L’ingénierie des données

1. À quoi ressemble votre pipeline de science des données (ou celui de votre organisation) ? Pourrait-il être amélioré ?
2. Identifiez des cas où vous avez rencontré des problèmes liés à la disponibilité, la facilité d’utilisation, la cohérence, l'intégrité, la qualité, la sécurité ou la fiabilité des données.
3. Complétez tous les exercices précédents que vous n'avez pas eu l'occasion de terminer.



12. La gestion des données

Quelques concepts fondamentaux

Les données et les **connaissances** doivent être structurées de manière à pouvoir être :

- stockées et accessibles
- modifiables et ajoutables
- extraites utilement et efficacement (extraire - transformer - charger)
- exploitées par des **humains** et des **ordinateurs** (programmes, bots, IA)

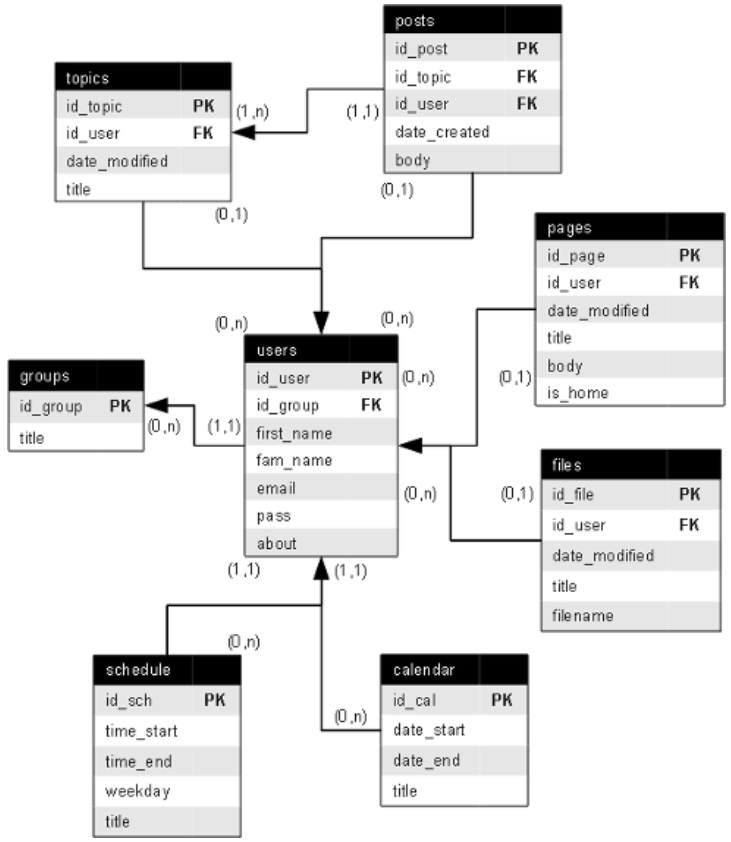
La modélisation des données

Les modèles de données sont des descriptions **abstraites/logiques** d'un système, utilisant des termes qui sont implémentables en tant que structure d'un type de logiciel de gestion des données.

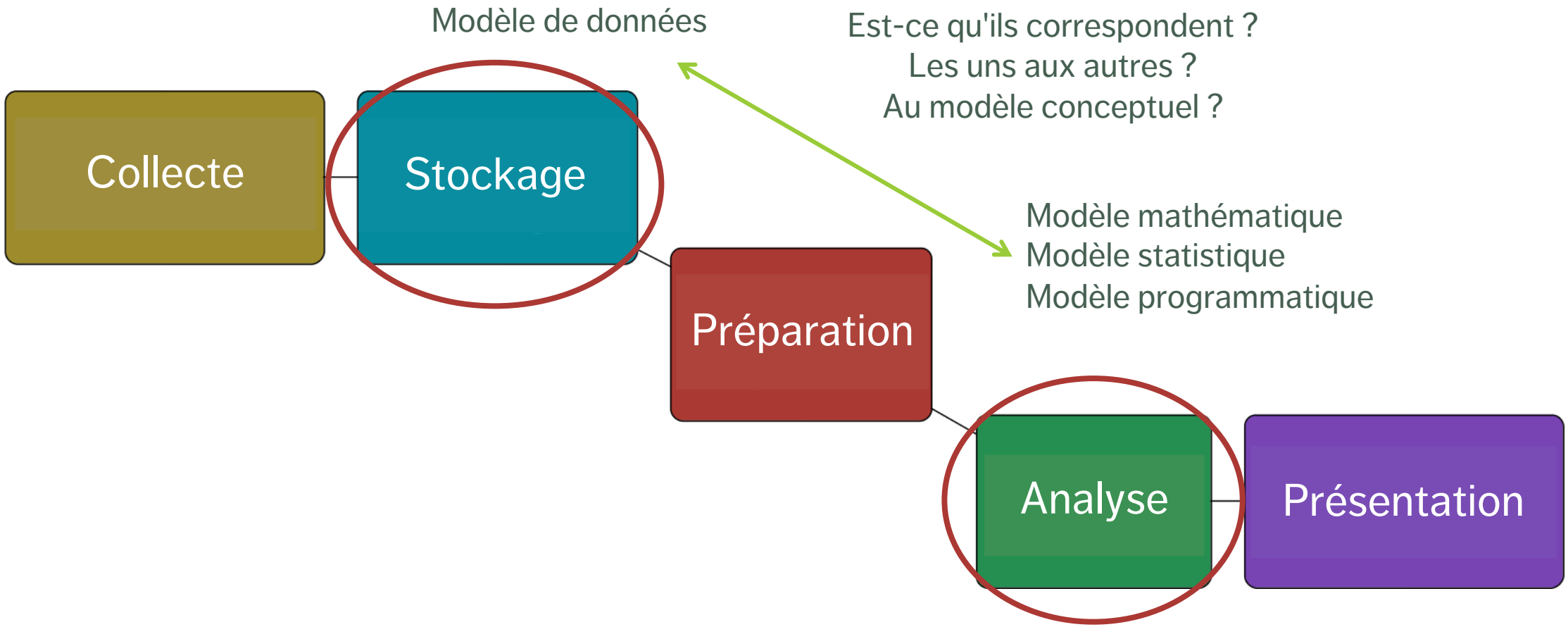
Cela se trouve à mi-chemin entre un **modèle conceptuel** et une **implémentation de banque de données**.

Les données elles-mêmes concernent les **instances** – le modèle, quant à lui, concerne les **types d'objets**.

Une autre option à envisager : les **ontologies**.



Un pipeline de données automatisé



[Representation of Langerhans cells in the *Cell Ontology*, A. M. Masci *et al.*]

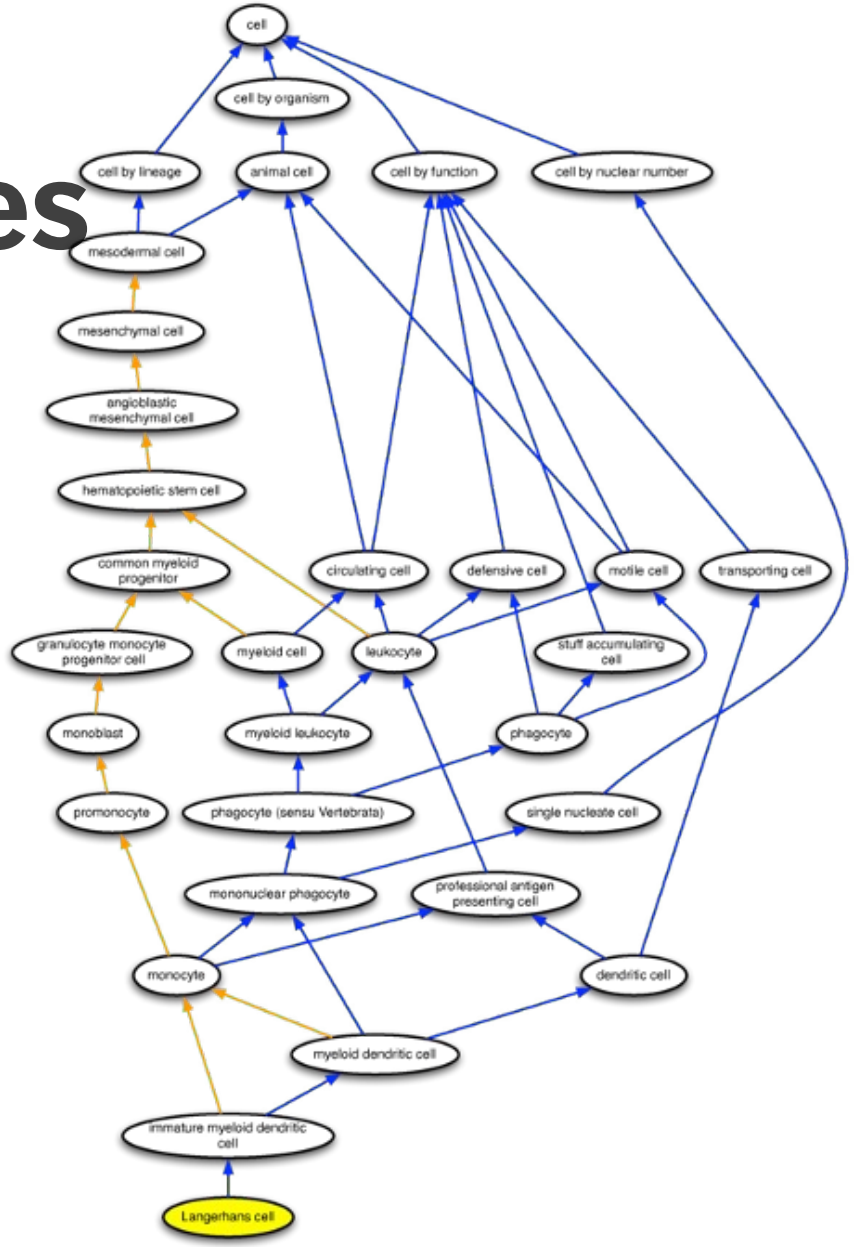
Métadonnées contextuelles

Nous perdons quelque chose lorsque nous passons de notre modèle conceptuel à un modèle de type spécifique – p. ex. le modèle de données ou de connaissances.

Une façon de conserver le contexte est de fournir des **métadonnées** (riches, si possible) – des données sur nos données!

Les métadonnées sont essentielles lorsqu'il s'agit de mettre en œuvre des stratégies pour travailler d'un ensemble de données à l'autre.

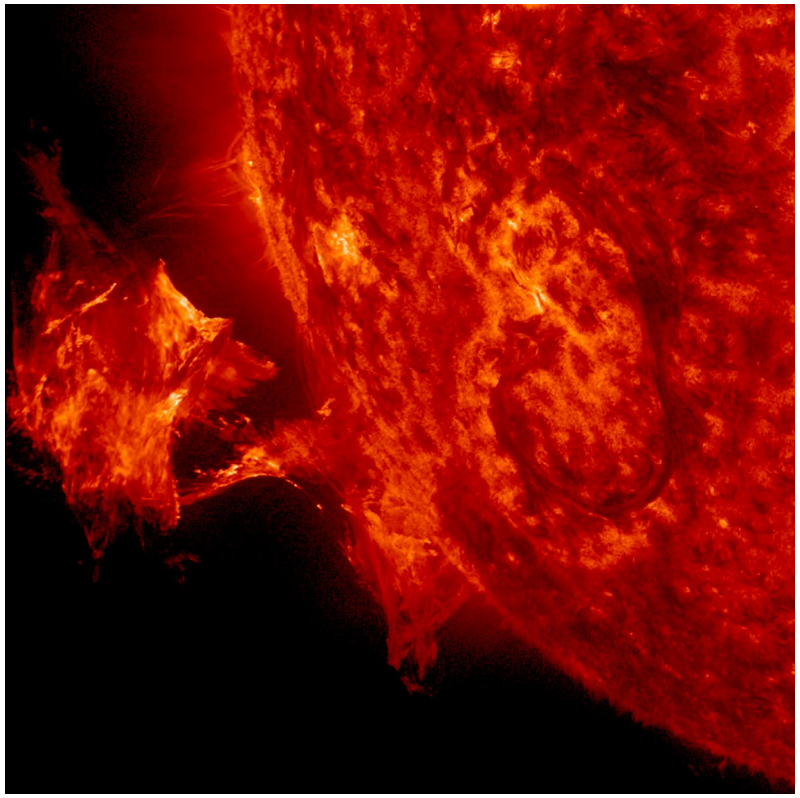
Les **ontologies** peuvent aussi jouer un rôle ici!



Les données (non) structurées

La disponibilité croissante de données non structurées et de grands objets binaires (**blob)** est l'une des principales motivations de certains des nouveaux développements dans les types de bases de données et autres stratégies de stockage de données :

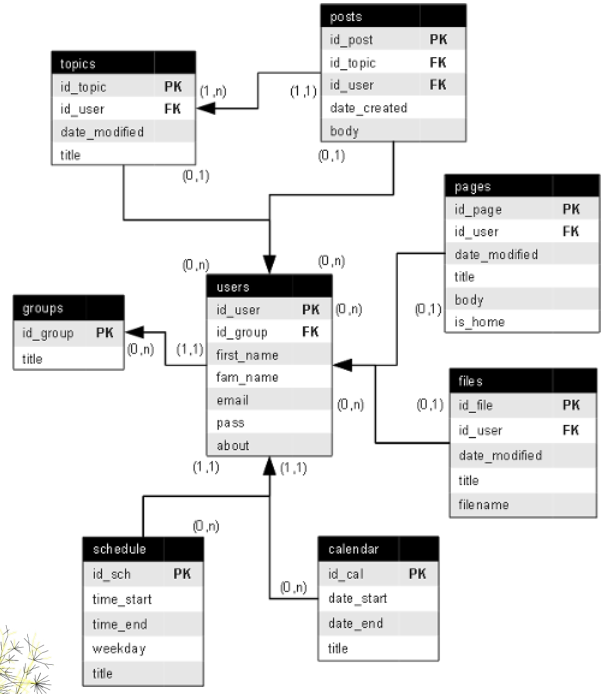
- **données structurées :** étiquetées, organisées, discrètes, selon une structure limitée et prédéfinie
- **données non structurées :** non organisées, pas de modèle de données à structure spécifique prédéfinie
- **données blob :** grand objet binaire – images, audio, multimédia

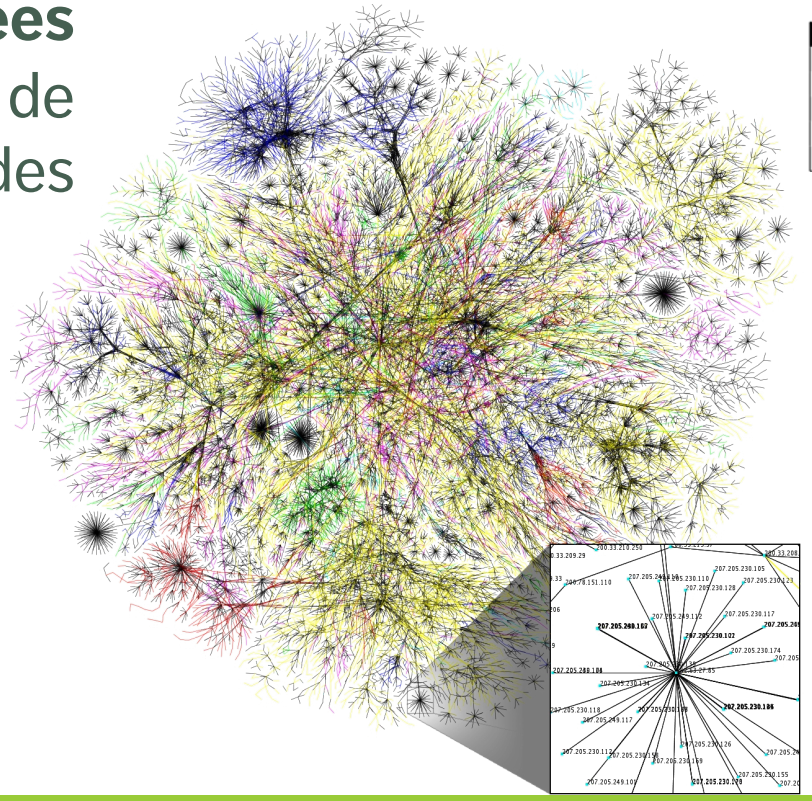


La modélisation des données

Différentes options sont actuellement populaires en termes de **données** fondamentales et de stratégies de modélisation ou de structuration des **connaissances :**

- paires valeur-clé (e.g., JSON)
- triples (e.g., RDF)
- bases de données graphiques
- bases de données relationnelles
- feuilles de calcul





Les mémoires et les bases de données

Base de données relationnelle :

- largement soutenue, bien comprise, fonctionne bien pour de nombreux types de systèmes et de cas d'utilisation. Base toutefois difficile à changer une fois mise en œuvre; ne gère pas bien les liens.

Magasins de clés-valeurs :

- peuvent prendre n'importe quel type de données; nul besoin de beaucoup de renseignements sur la structure ; si vous avez beaucoup de valeurs manquantes, ces mémoires ne prendront pas de place ; peuvent toutefois être désordonnées et mystérieuses; difficile d'y trouver des données.

Bases de données graphiques :

- rapides et intuitives si vous utilisez des données fortement axées sur les liens; pourraient être la seule option si vos données sont ainsi parce que les bases de données traditionnelles peuvent ralentir énormément ; sont souvent trop spécialisées ; pas encore supportées à grande échelle.

Les fichiers “plats” et les feuilles de calcul

Pour :

- très efficace si vous recueillez des données une seule fois, sur un type particulier d’objet
- certains types d’analyse exigent que vous ayez toutes les données en un seul endroit
- facile à lire dans un logiciel et à effectuer des opérations sur l'ensemble des données

Cons :

- très difficile de gérer l'intégrité des données si l'on collecte continuellement des données
- pas idéal pour les données de systèmes impliquant de multiples types d'objets et de relations
- il peut être très difficile d'effectuer des opérations d'interrogation de données

Quelques outils et mots-clés

- MongoDB, ArangoDB
- Magasin de documents
- JSON, YAML
- API, GraphQL
- Données interreliées
- Web sémantique
- Langage d'ontologie Web (OWL)
- Protégé
- SQL, etc.

La mise en œuvre du modèle

Pour mettre en œuvre votre modèle de données/connaissances, il faut avoir accès à un **logiciel de stockage et de gestion des données**.

Cela peut constituer un défi pour les particuliers : ces logiciels fonctionnent généralement sur des **serveurs**.

Les serveurs sont utiles car ils permettent à plusieurs utilisateurs d'accéder **simultanément** à une même base de données, à partir de différents programmes clients, mais il est difficile de "jouer" avec les données.

C'est là que **SQLite** entre en jeu.

Le rôle du logiciel de gestion des données

Les logiciels de gestion des données offrent aux utilisateurs un moyen facile d'interagir avec leurs données.

Il s'agit essentiellement d'une interface entre les **personnes** et les **données**.

Grâce à cette interface, les utilisateurs peuvent :

- ajouter des données à leur collection de données
- extraire des sous-ensembles de données de leur collection en fonction de certains critères
- supprimer ou modifier des données dans leur collection

Un peu de terminologie

Auparavant :

- base de données
- entrepôt de données
- mini-entrepôt de données
- système de gestion des données
- (SQL)

Maintenant :

- lac de données
- bassin de données
- marais de données ?
- cimetière de données ?
- (NoSQL)

De plus en plus : on fait une distinction entre l'**entrepot de données** et le **logiciel de gestion des données**.

Du modèle de données à la mise en œuvre

Une fois que le mode de données (logique) est achevé :

1. **instancier le modèle** dans le logiciel choisi (par exemple, créer des tables dans MySQL)
2. **télécharger/charger les données**
3. **interroger les données :**
   - les bases de données relationnelles traditionnelles utilisent le **langage de requête structuré** (SQL : Structured Query Language)
   - d'autres utilisent des langages de requête différents (AQL, moteurs sémantiques, etc.) ou s'appuient sur des programmes informatiques sur mesure (par exemple, écrits en R, Python)

La gestion des bases de données

Une fois les données collectées, il faut aussi les **gérer**.

Fondamentalement, cela signifie que la base de données doit être **maintenue**, afin que les données soient

- **précises**
- **exactes**
- **cohérentes**
- **complètes**

Ne laissez pas votre lac de données se transformer en marais de données !

Sevvices en nuage (Cloud Services)

1. Stocker de **grandes** quantités de données
2. Exécutez des processus coûteux et avancés en **cliquant sur un bouton**
3. **Flexible** et **évolutif**
4. Permettre le traitement des données **en code bas**

Nuage vs. accès local

Nuage (Cloud)

- sans intervention manuelle
- paiement à la consommation
- propriétaire douteux

Accès local (On-Premise)

- auto-entretenu
- tous les coûts sont absorbés
- sécurité entièrement contrôlée

Lectures conseillées

La gestion des données

Data Understanding, Data Analysis, Data Science
Volume 2: Fundamentals of Data Insight

14. Data Science Basics
    14.5 Getting Insight From Data
    - Structuring and Organizing Data

17. Data Engineering and Data Management
    17.3 Data Management
    - Databases
    - Data Modeling
    - Data Storage

    17.4 Reporting and Deployment
    - Reports and Products
    - Cloud and On-Premise Architecture

Exercices

La gestion des données

1. Votre organisation possède-t-elle des données ? Si oui, sont-elles hébergées localement ou sur le cloud ? Comment y accède-t-on ? Comment sont-elles structurées ?

2. Complétez tous les exercices précédents que vous n'avez pas eu l'occasion de terminer.