Intégration de Données

- Concepts Fondamentaux -

M2 MIAGE IPM

jiefu.song@ut-capitole.fr

UNIVERSITÉ TOULOUSE CAPITOLE

FACULTÉ D'INFORMATIQUE

Membre de l'Université Toulouse Capitole

Plan

- **Contexte**
- **Intégration de données**
  - Sources
  - Approches
  - Architectures
  - Methodes

Pour les examens :
- Partie CM sous forme de QCM
- SQL sous forme de contrôle intégré ou exos à rendre

Talend :
Exo à rendre

Contexte

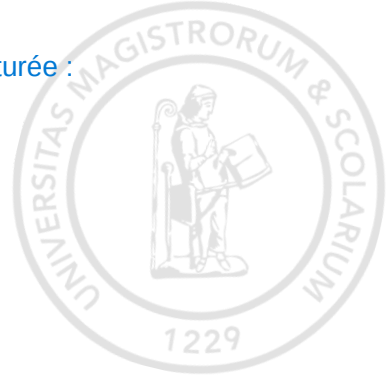
- Omniprésence des outils informatique → prolifération des données
- Dans l'entreprise

  Besoin de plus en plus d'informations => enrichi par des données externes

  - *Data-Driven Companies* : utilise des technologies de gestion et d'analyse de données pour orienter son organisation
  - Données : non seulement un produit pur informatique mais également actif stratégique à valoriser
- Pour le grand public
  - Accès simple, rapide et efficace aux informations disponibles sur le web
- Mais avant tout, une gestion efficace des données…

Contexte

Données semi-structurée :



- Gestion de données traditionnelle Felbes ? Open Data
  - Bases de données homogènes (modèle/schéma uniques)
  - Architecture centralisée ou distribuée, transparente au niveau logique
- A l'heure actuelle : vers un mélange de données en provenance de multiples sources
  - Sources d'information nombreuses et variées
    - SGBD relationnels, SGBD NoSQL, Documents/fichiers, Open Data, Linked Data
  - Interfaces d'accès variées
    - Modèle de données: relationnel, clé/valeur, graphe
    - Langages d'interrogation : SQL, Cypher, Sparql, …
    - Protocoles de communication: JDBC, ODBC, HTTP, …

Intégration compliquée

Contexte

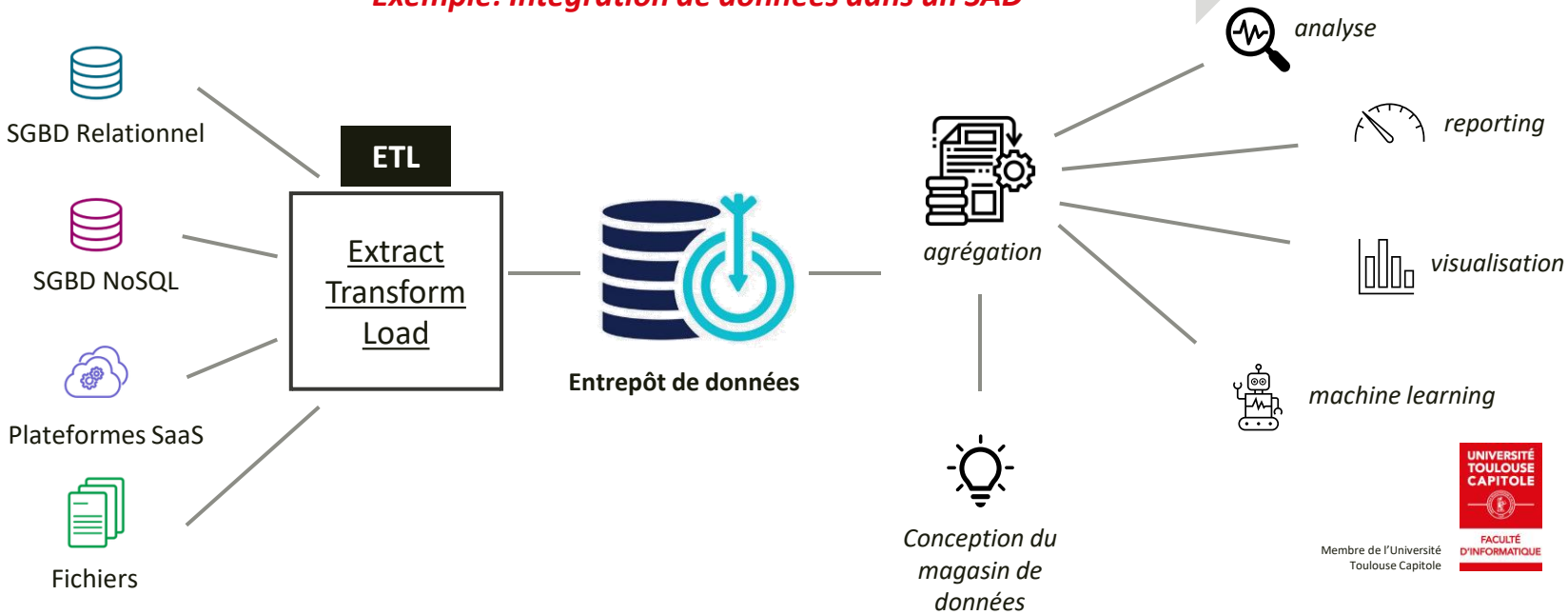
Gestion de données

Exploitation / Traitement

Gris car on n'a pas la main dessus

Sources de données variées | Intégration des données | Solution « Data » | Traitement/Dev spécifique | Mise en valeur

Exemple: Intégration de données dans un SAD



INTÉGRATION DE DONNÉES

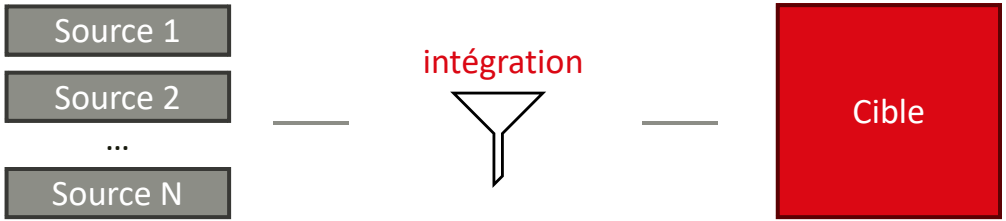
Sources

UNIVERSITÉ TOULOUSE CAPITOLE
FACULTÉ D'INFORMATIQUE
Membre de l'Université Toulouse Capitole

/!\ FC

Intégration de données

- **Objectif : utiliser les données multi-source, comme si elles étaient dans une seule source homogène**



- **Cette intégration doit fournir**
  - – un accès (requêtes d'intégration, éventuellement de mise-à-jour) Récupération / Modif / Suppression
  - – uniforme (comme si c'était une seule BD homogène) Indépendant de plusieurs schémas
  - – à des sources multiples (pas seulement des BD) Plusieurs sources de données
  - – autonomes (sans affecter leur comportement) Pas de perturbation
  - – hétérogènes (différents modèles de données, schémas, instances) Modèle, schéma, protocole

FC

Intégration de données

- **Sources de données**
  - 3 caractéristiques principaux qui rendent l'intégration de données difficile
    - Distribution
    - Autonomie
    - Hétérogénéité

Intégration de données

- **1 Distribution**
  - Données stockées sur des supports répartis physiquement et géographiquement
    - Enjeu important : monter à l'échelle (scalability)
  - Avantages
    - Disponibilité: tous les sites ne tombent pas en panne en même temps
    - Temps d'accès: partage de la charge, parallélisme
  - Problèmes
    - Temps de communication
    - Identification des sources contenant les données pertinentes
    - Hétérogénéité en termes de puissance de traitement
    - Indisponibilité de certaines sources à un moment ou à un autre

Intégration de données

- **2. Autonomie**
  - Conception : les sources décident de leur propre
    - modèle de données,
    - langage d'interrogation,
  - Communication : les sources décident quand et comment répondre à une demande de données
  - Exécution : les sources décident de l'ordre d'exécution des transactions locales ou des opérations externes
    - Peu ou pas d'informations (feedback) fournies sur les détails internes d'exécution
      Si trop de temps pris pour le traitement / risque de ne pas recevoir de réponse

Intégration de données

- **3. Hétérogénéité**
  - Sémantique : java / java / java
  - Syntaxique : ce n’est pas terrible / ce n’est pas terrible
  - Schéma/structure : first name + last name / full name
  - Technique : différents logiciels gérant les données

INTÉGRATION DE DONNÉES

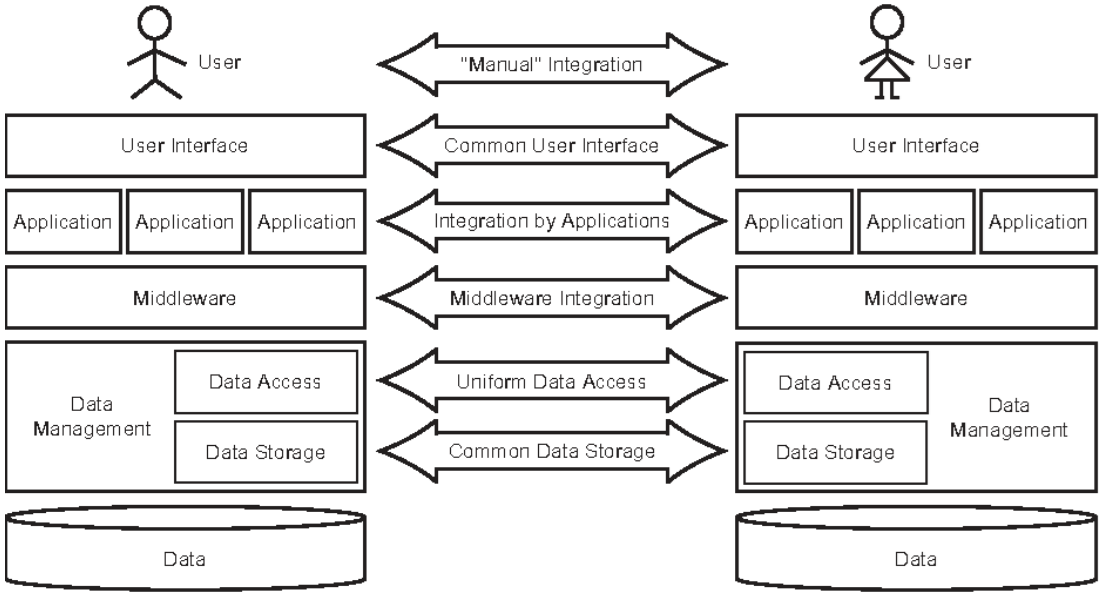
Approches

UNIVERSITÉ TOULOUSE CAPITOLE
FACULTÉ D'INFORMATIQUE

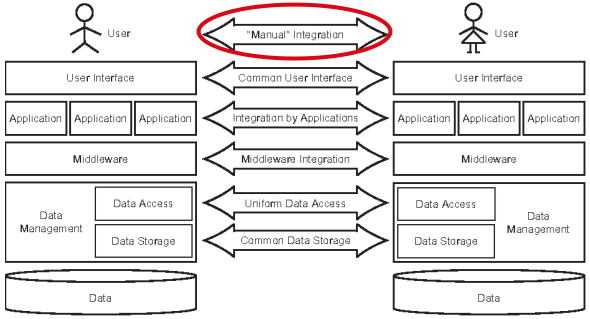
Membre de l'Université Toulouse Capitole

Approches

- **L'intégration de données peut avoir lieu à différents niveaux**

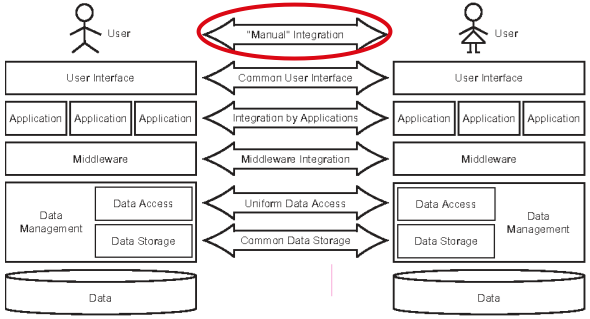


Approches



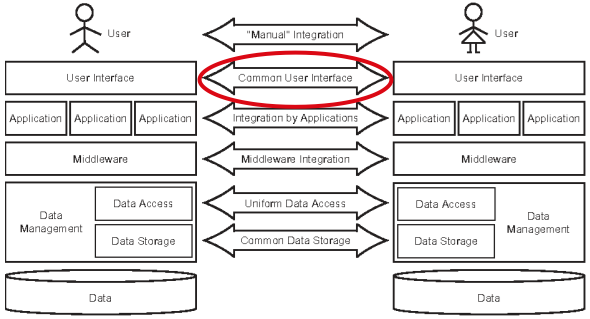
- **Intégration Manuelle**
  - Pour une petite organisation avec peu de données, un Data Manager peut effectuer manuellement toutes les phases de l'intégration
    - interagir avec toutes les sources pertinentes
    - collecter manuellement les données Les collecter, les comprendre
    - les nettoyer et les intégrer manuellement pour fournir des informations utiles Pour faire des correspondances entre les données
  - Connaissances sur l'emplacement, la représentation logique des données et la sémantique des données

Approches



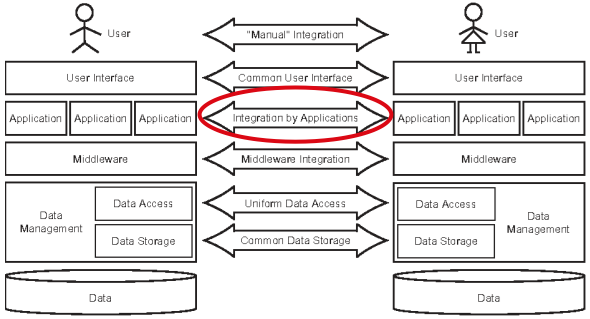
- **Intégration Manuelle**
  - Avantages :
    - Coûts réduits : seule une petite quantité de sources de données est intégrée.
    - Plus de liberté : l'utilisateur a un contrôle total sur l'intégration.
  - Inconvénients :
    - Moins d'accès : un développeur doit implémenter manuellement chaque intégration.
    - Difficulté technique : les utilisateurs doivent gérer différentes interfaces utilisateur et langages de requête.
    - Difficulté à l'échelle : l'expansion pour des projets plus importants nécessite des modifications manuelles du code pour chaque intégration, ce qui prend du temps.
  - Résumé : idéal pour des cas ponctuels, mais devient rapidement peu pratique pour des intégrations complexes ou récurrentes en raison du processus manuel fastidieux.

Approches



- **Interfaces d'Utilisateur Commune (Common User Interfaces)**
  - ≈ Intégration manuelle, mais avec une interface commune (un navigateur web, une interface graphique dédiée, etc.)
  - Données toujours présentées séparément
  - Homogénéisation et intégration manuelle
  - Même avantages & inconvénients que l'intégration manuelle
  - Par exemple : recherche sur Google…

Approches

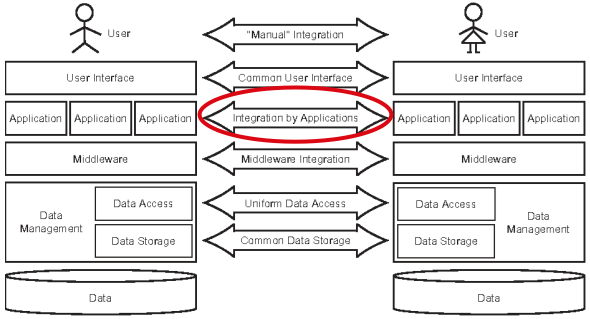


- **Intégration par applications (Entreprise Application Integration)**
  - Les applications logicielles localisent, extraient, nettoient et intègrent des données provenant de sources disparates, et elles retournent des résultats intégrés aux utilisateurs
  - Avantages :

L'utilisateur n'est pas au courant qu'il y a eu un échange de données => transparence de l'exécution

    - Processus simplifiés : Une application effectue automatiquement tout le travail.
    - Échange d'informations facilité : L'application permet de transférer des données multi-sources de manière transparente.
    - Moins de ressources utilisées : Étant donné que la majeure partie du processus est automatisée, les data managers peuvent se consacrer à d'autres projets.

Approches

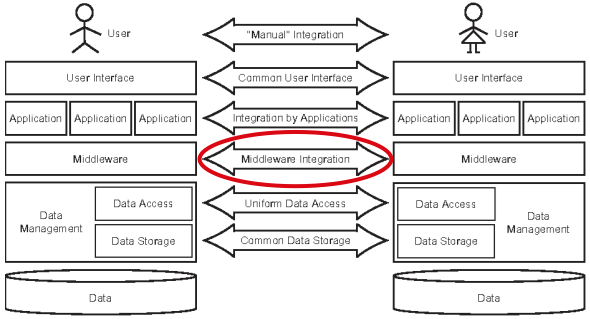


- **Intégration par applications (Entreprise Application Integration)**
  - Inconvénients :
    - Accès limité : Cette technique nécessite des connaissances techniques spéciales et un data manager dédié pour superviser le déploiement et la maintenance de l'application.

      Pas de contrôle sur les protocoles de communication
    - Résultats incohérents : L'approche n'est pas standardisée et varie selon les entreprises qui proposent ce service.

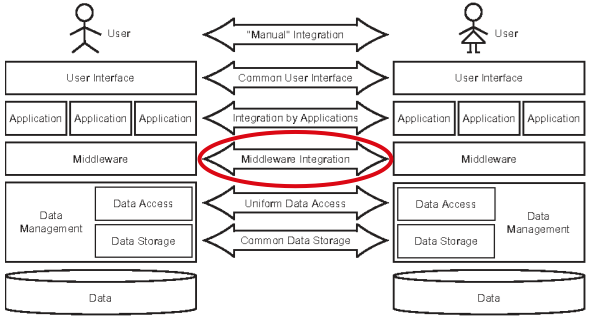
      Si dans une partie ou sous partie des dépendances apparaissent
    - Configuration compliquée : Des connaissances métiers et techniques sont requises.
    - Gestion difficile des données : l'intégrité des données n'est pas toujours garantie entre les sources. Les applications deviennent de plus en plus lourdes à mesure que le nombre d'interfaces système et de formats de données à homogénéiser et à intégrer augmente.
  - Résumé
    - Travaille avec de multiples sources de données - sur site et dans le cloud
    - Adaptée à un nombre limité d'applications.

Approches



- **Intégration par middleware** Facilite les échanges avec les applications
  - Middleware qui collecte des données à partir de différentes sources, les normalise et les stocke dans le jeu de données résultant.
  - Adopté lorsque l'entreprise souhaite intégrer des données des *legacy systems* hérités aux systèmes modernes.
  - Avantages :
    - Meilleure diffusion des données : Le middleware effectue automatiquement l'intégration de la même manière à chaque fois.
    - Accès plus facile entre les systèmes : Le middleware est programmé pour faciliter la communication entre les systèmes au sein d'un réseau.

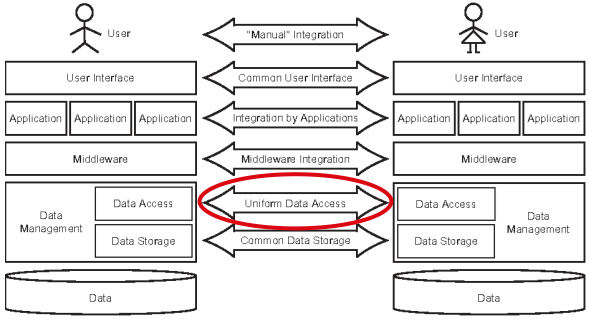
Approches



- **Intégration par middleware**
  - Inconvénients :
    - Moins d'accès : Le middleware doit être déployé et maintenu par un développeur ayant des connaissances techniques.
    - Fonctionnalité limitée : Le middleware ne peut fonctionner qu'avec certains systèmes.
  - Résumé :
    - Pour les entreprises qui travaillent aussi bien avec des *legacy systems* qu'avec des systèmes modernes, le middleware est idéal,
    - mais il s'agit principalement d'un outil de communication : capacités limitées en matière d'analyse de données.

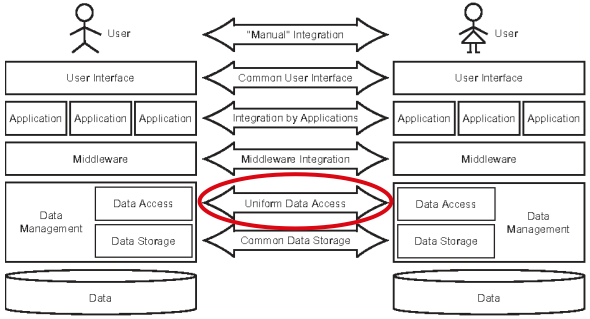
Il s'agit d'un outil de communication et non d'agrégation de données

Approches



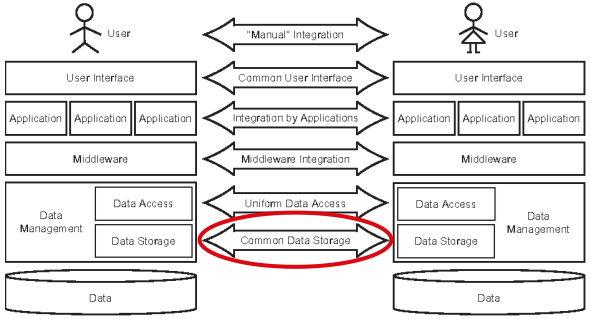
- **Accès uniforme aux données : intégration virtuelle/logique des données**
  - Une technique qui récupère et affiche uniformément les données, tout en les laissant dans leur source d'origine.
  - Cette technique crée uniquement une vue unifiée qui représente les données intégrées.
  - Avantages :
    - Moins de besoins en stockage : Il n'est pas nécessaire de créer un emplacement distinct pour stocker les données.
    - Accès aux données simplifié : Cette approche fonctionne bien avec de multiples systèmes et sources de données.
    - Vue simplifiée des données : Cette technique crée une apparence uniforme des données pour l'utilisateur final.

Approches



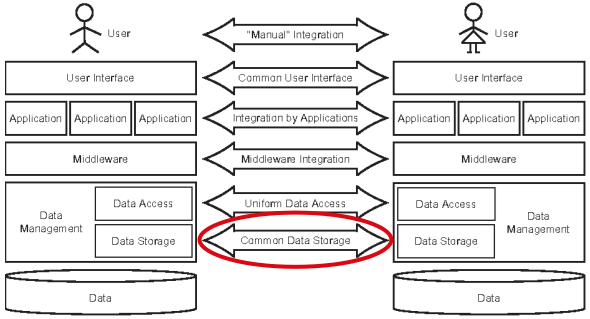
- **Accès uniforme aux données : intégration virtuelle/logique des données**
  - Inconvénients : Problème de sources distribuées
    - Défis liés à l'intégrité des données : L'accès à autant de sources peut compromettre l'intégrité des données.
    - Systèmes surchargés : Les systèmes hôtes de données ne sont généralement pas conçus pour gérer la quantité et la fréquence des demandes de données dans ce processus.
    - L'accès aux données, leur homogénéisation et leur intégration doivent être effectués à l'exécution, mais cela prend du temps.
  - Résumé :
    - Pour les entreprises ayant besoin d'accéder à plusieurs systèmes disparates, il s'agit d'une approche optimale.
    - Si la demande de données n'est pas trop contraignante pour le système hôte, cette approche peut fournir des informations sans le coût de la création d'une copie des données sources.

Approches



- **Stockage de données commun : intégration physique des données**
  - Une approche qui récupère et affiche uniformément les données, mais qui fait également une copie des données et les stocke.
  - Contrairement à l'accès uniforme, cela implique la création et le stockage d'une copie des données dans une cible.
  - Avantages : Les systèmes d'hôte ne s'occupent pas de l'intégration
    - Fournit un accès rapide aux données.
    - Allège la charge : le système hôte ne gère pas en permanence les requêtes de données.
    - Restitution uniforme des données
    - Amélioration de la performance d'analyse, même en cas des requêtes complexes

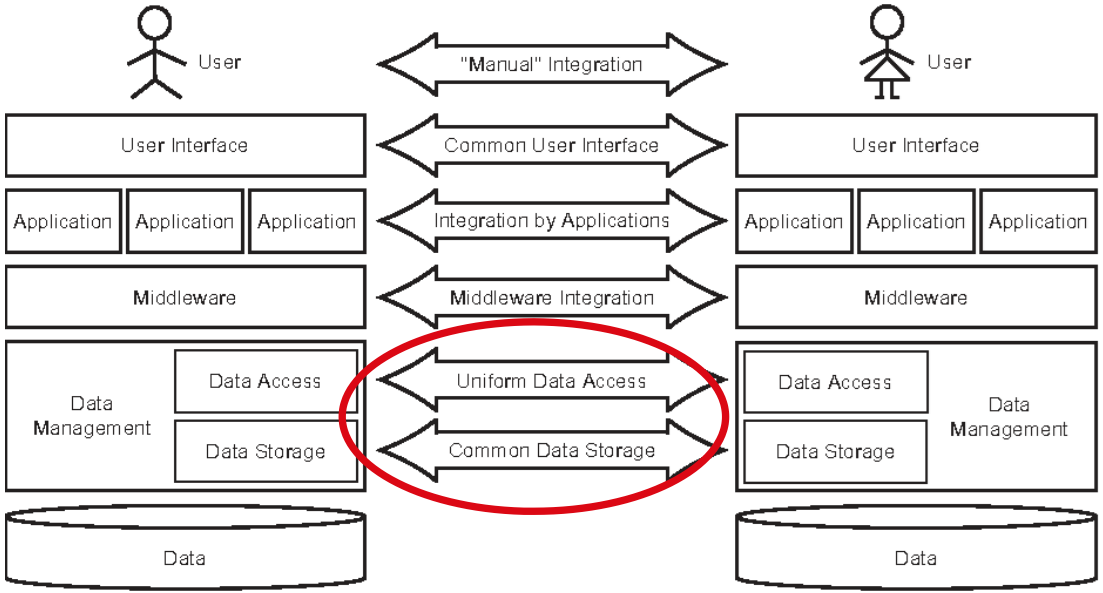
Approches



- **Stockage de données commun : intégration physique des données**
  - Inconvénients :
    - Coûts de stockage accrus : la création d'une copie des données implique de trouver un endroit où les stocker et de payer pour cet espace.
    - Coûts de maintenance plus élevés : l'orchestration de cette approche nécessite des experts techniques pour mettre en place l'intégration, la superviser et la maintenir.
  - Résumé : Niveau au plus proche des données. Interrogation sophistiquée des données.
    - L'approche d'intégration la plus sophistiquée.
    - Elle permet les requêtes les plus sophistiquées, et cette sophistication peut conduire à des insights plus profonds.

Approches

- **Dans le cadre de notre cours :**



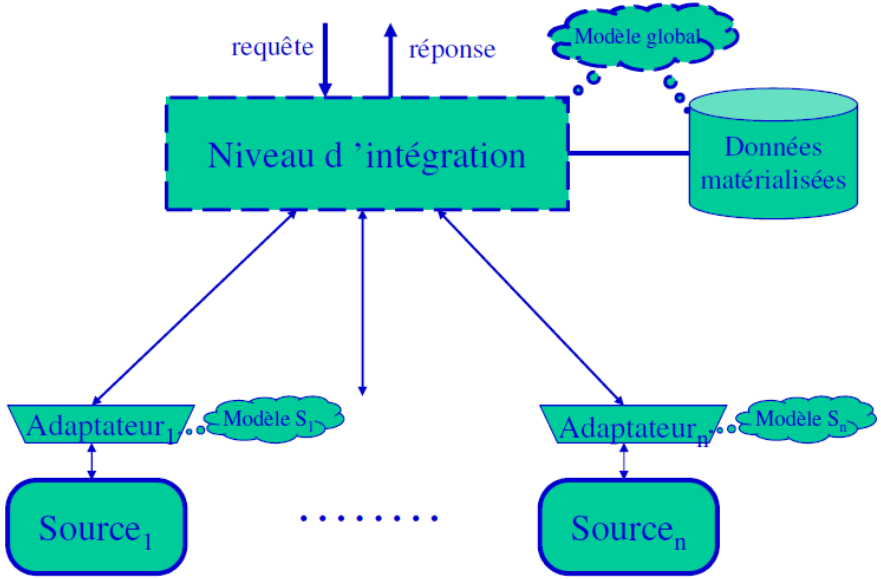
INTÉGRATION DE DONNÉES

Architectures

UNIVERSITÉ TOULOUSE CAPITOLE
FACULTÉ D'INFORMATIQUE
Membre de l'Université Toulouse Capitole

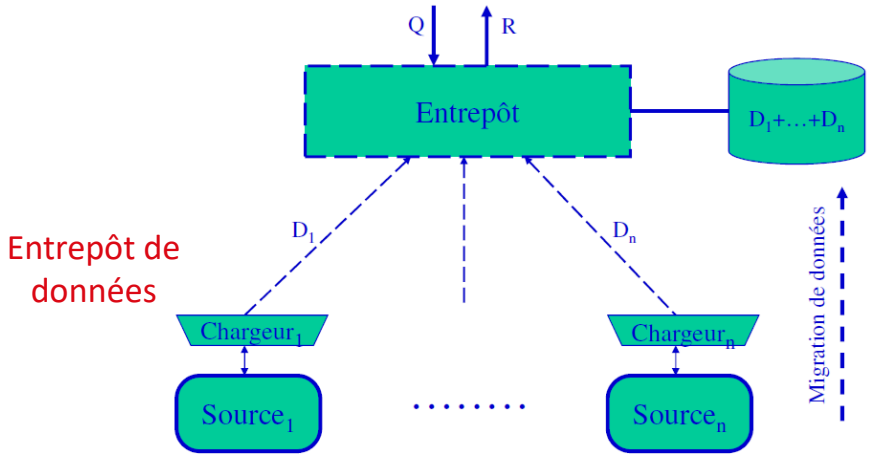
Architectures

- **Architecture générale**



Architectures

- **Architecture pour l'intégration physique**
  - Les données provenant des sources sont transformées et stockées sur un support spécifique (par exemple, un entrepôt de données).
  - L'interrogation s'effectue comme sur une BD classique



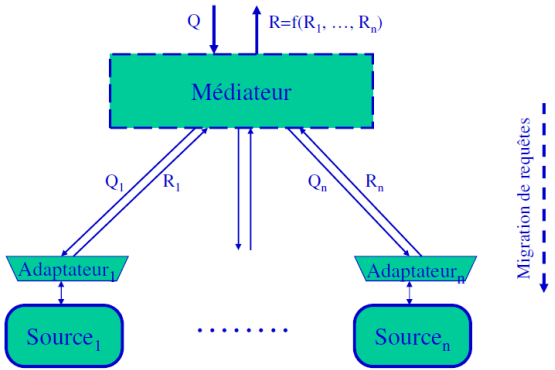
Architectures

- **Architecture pour l'intégration physique**
  - matérialisation des sources au niveau du modèle global
  - migration de données vers l'entrepôt
  - avantages
    - Performances
    - personnalisation des données (nettoyage, filtrage)
    - versionning
  - inconvénients
    - données pas toujours fraîches, cohérence
    - gestion des mises-à-jour
    - gestion de gros volumes de données

Architectures

- **Architecture pour l'intégration virtuelle**
  - Les données restent dans les sources
  - Les requêtes sont exprimées sur un schéma global, puis décomposées en sous-requêtes sur les sources
  - Les résultats des sources sont combinés pour former le résultat final

mediator/wrapper



Comparable à une vue dans Oracle

Où a lieu l'uniformisation des données ? Dans le médiateur.

Architectures

- **Architecture pour l'intégration virtuelle** Au lieu d'une migration de données => migration de la requête de données
  - approche « paresseuse », pas de matérialisation
  - migration de requêtes vers les sources
  - avantages
    - données toujours fraîches,
    - plus facile d'ajouter de nouvelles sources
    - plus grande échelle
    - distribution de l'effort
  - Inconvénients
    - Performance
    - traduction de requêtes
    - Capacités différentes des sources

La traduction de requêtes se fait par le Data Management

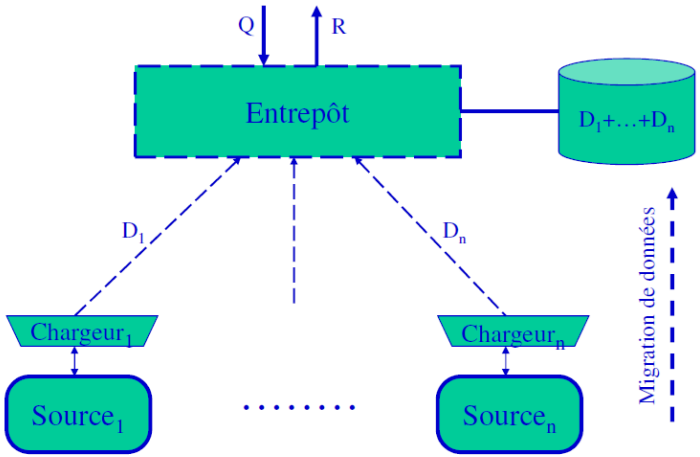
INTÉGRATION DE DONNÉES

Méthodes

UNIVERSITÉ TOULOUSE CAPITOLE
FACULTÉ D'INFORMATIQUE
Membre de l'Université Toulouse Capitole

Stockage de données commun

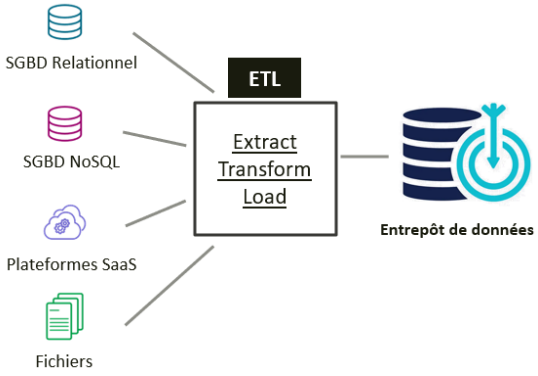
- **Intégration physique via un entrepôt de données**
  - L'approche la plus populaire d'intégration de données



Stockage de données commun

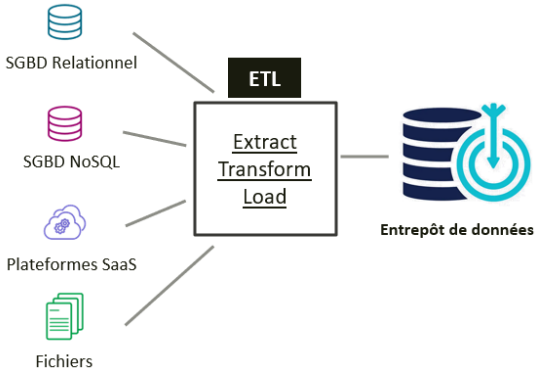
- **Intégration physique via un entrepôt de données**
  - Avantage
    - Performance
    - Contrôle plus facile de l'hétérogénéité des données
  - Utilisation pour les Systèmes d'Aide à la Décision OLAP => Il y a un entrepôt de données
  - Transformation de données pour alimenter l'entrepôt
    - Chargeurs = systèmes ETL (« Extract, Transform, Load »)
    - Outils graphiques pour définir des flots de traitements/transformations (e.g., Talend) autrement appelé pipeline
    - Une fois le flot de traitement défini : appliqué au contenu des sources

Stockage de données commun



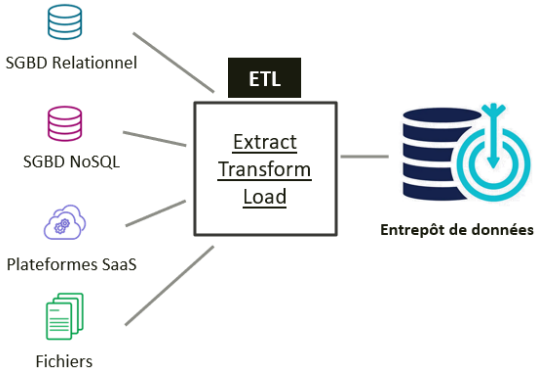
- **ETL – Extract, Transform, Load**
  - Permet la consolidation des données à l'aide des trois opérations suivantes:
    - Extraction: identifier et extraire les données de sources ayant subiune modification depuis la dernière exécution;
    - Transformation: appliquer diverses transformations aux données pour les nettoyer, les intégrer et les agréger; Transformation de données numériques, Transformation d'unité (Kilo => Livres)
    - Chargement: insérer les données transformées dans l'entrepôt et gérer les changements aux données existantes
  - Traite normalement de grandes quantités de données de manière périodique

Stockage de données commun



- **ETL – Extract, Transform, Load**
  - Avantages
    - Optimisé pour la structure de l'entrepôt de données;
    - Peut traiter de grandes quantités de données dans une même exécution (traitement en lot); Batch processing
    - – Permet des transformations complexes et agrégations sur les données; Le temps d'exécution n'est pas très impactant car pas de répercutions sur les données sources
    - La disponibilité d'outils graphique sur le marché permet d'améliorer la productivité;
    - Permet la réutilisation des processus et transformations

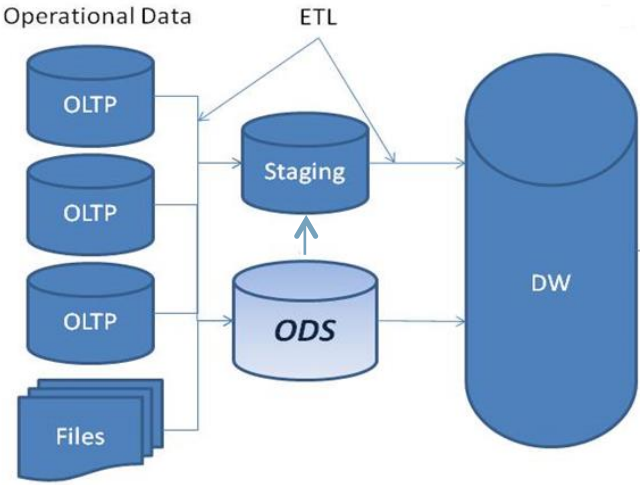
Stockage de données commun



- **ETL – Extract, Transform, Load**
  - Inconvénients:
    - Processus de développement long et coûteux;
    - Gestion des changements nécessaire;
    - Exige de l'espace disque pour effectuer les transformations (staging area);
    - Latence des données entre la source et l'entrepôt; Ne peut pas prendre de décision en temps réel
    - Unidirectionnel (des sources vers l'entrepôt de données).

Stockage de données commun

- **ODS et DSA**
  - Pour passer des sources vers un entrepôt de données, les données suivent un certain nombre d'étapes



/!\ FC + Mindmap

Stockage de données commun

- **ODS et DSA**
  - ODS (Operational Data Store) : une base de données intermédiaire dans laquelle se trouve des extraits des sources hétérogènes de manière temporelle
    - sert de zone d'attente provisoire pour toutes les données qui sont sur le point d'intégrer l'entrepôt de données. Offre un stockage temporaire optionnel
    - ODS offre un stockage
      - non permanente
      - optionnel
    - Quand : traitement d'un grand volume de données provenant **de différentes sources** nécessitant des **transformations complexes** (par exemple, agrégation)

Stockage de données commun

- **ODS et DSA**
  - ODS (Operational Data Store) : une base de données intermédiaire dans laquelle se trouve des extraits des sources hétérogènes de manière temporelle
    - Pourquoi? Réduction du temps d'extraction des données
      - Minimisez le temps d'extraction des données des sources
      - Plate-forme unifiée pour toutes les données sources pour faciliter les processus de transformation :
        - transformation complexe et ne pouvant pas être fait à la volée ou en une seule étape dans la mémoire
        - volume de données important et ne pouvant pas être mis en mémoire. Exécution des traitements facilités
        - données provenant de plusieurs sources à des moments différents.
    - Enregistrer que le différentiel entre rafraichissement
    - Un ODS est généralement exécuté sur un système de gestion de base de données relationnelle ou sur la plateforme Hadoop HaDoop File System

Stockage de données commun

- **ODS et DSA**
  - ODS (Operational Data Store) : une base de données intermédiaire dans laquelle se trouve des extraits des sources hétérogènes de manière temporelle
    - Bonnes pratiques
      - Charger des données brutes dans les tables ODS

        - Projection sur tables sources

        - Sélection temporelle sur les données sources pour « ne conserver que les données fraiches »

        - Transformation mineures : rognage, transformation en majuscules de chaînes de caractères, transformation de types de données
      - Ne pas diviser une table source en plusieurs tables ODS
      - Mettre des dates de création ou de maj
      - Convention de nommage des tables <nom de source> _ <nom de table>

Stockage de données commun

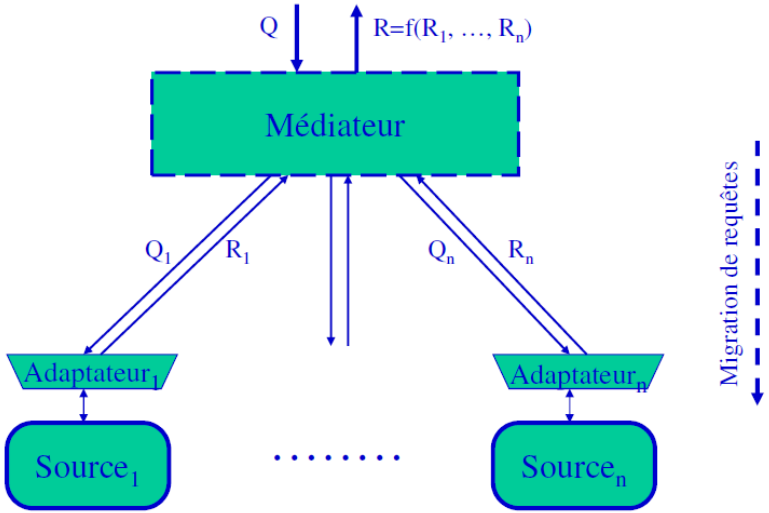
- **ODS et DSA**
  - Staging (Data Staging Area) : zone de stockage temporaire pour les données qui ont été extraites de différentes sources de données brutes (de structures et de formats différents) Cette zone sert à combiner des zones intermédiaires (ODS ou sources de données) afin de les "normaliser" avant de les envoyer à la cible de stockage de données
    - Dans cette zone, les données sont souvent nettoyées, normalisées, enrichies et structurées de manière à faciliter leur traitement ultérieur.
      - Tables avec une structure le plus proche possible du schéma de l'ED
    - Elle sert de zone tampon pour le ***traitement*** des données
      - Stockage temporaire
      - Nettoyage
      - Normalisation
      - Validation selon des règles de qualité
      - Transformation (format, structure, valeurs etc.)

Stockage de données commun

- **ODS et DSA**
  - Staging (Data Staging Area) : zone de stockage temporaire pour les données qui ont été extraites de différentes sources de données brutes (de structures et de formats différents)
    - Quelques exemples d'outils utilisés
      - BD relationnelles « légères » : MySQL, PostgreSQL…
      - Modules dédiés dans des outils ETL : Talend, Informatica…
      - Systèmes de stockage de fichier : Hadoop HDFS, AWS S3, Azure Blob

Accès uniforme aux données

- **Intégration virtuelle via Mediator/Wrapper** Prend de plus en plus d'ampleur en raison du volume de données important
  - moins utilisés en pratique, mais du potentiel à l'ère du Big Data

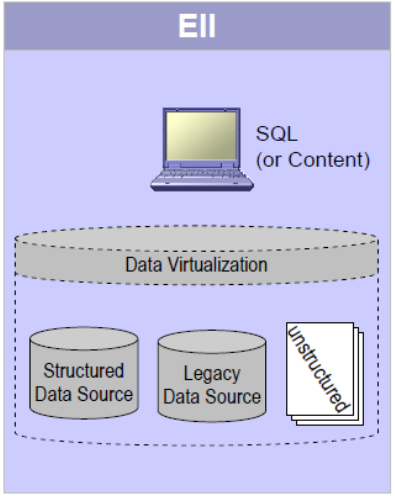


Accès uniforme aux données

- **Intégration virtuelle via Mediator/Wrapper**
  - Avantage
    - Meilleur passage à l'échelle
    - Acceptent mieux les changements dynamiques (nouvelles sources)
  - Mieux adapté à l'intégration de sources Web
  - Permettant de compléter l'intégration physique : approche hybride
  - EII - Enterprise Information Integration

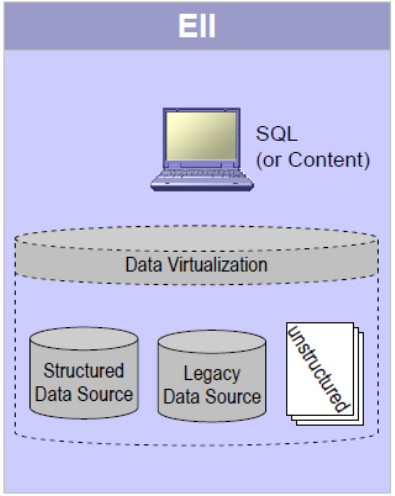
Accès uniforme aux données

Projection, uniformisation,



- **Implémentation : EII – Enterprise Information Integration**
  - Fournit une vue unifiée des données de l'entreprise
  - Les sources de données dispersées sont consolidées à l'aide d'une BD virtuelle, de manière transparente aux applications / utilisateurs
  - Une couche de métadonnées pour définir "comment et où" obtenir les données
  - Permet de consolider uniquement les données utilisées, au moment où elles sont utilisées (source data pulling).

Accès uniforme aux données



- **Implémentation : EII – Enterprise Information Integration**
  - Fonctionnement du « source data pulling » Mediator / Wrapper
    1. attendre une demande
    2. divise la requête (si nécessaire) à travers des systèmes sources hétérogènes
    3. rassemble des ensembles de données
    4. les fusionne ensemble (en s'appuyant sur la couche de métadonnées pour les règles d'intégration
    5. les transmet au demandeur

Accès uniforme aux données

- **Implémentation : EII – Enterprise Information Integration**
  - Avantages:
    - Aucun déplacement de données
    - Accès temps-réel aux données
    - Accès « relationnel » à des sources non-relationnelles
    - Accélère le déploiement de la solution
    - Peut être réutilisé par le système ETL dans une itération future

Ce cours est à mi-chemin entre IDA et la recherche.
LAV <-> GAV ?

Les termes utilisés sont différents en f° des environnements, organismes, ...

Accès uniforme aux données

Garder pour moi le slide, ne pas le diffuser.

- **Implémentation : EII – Enterprise Information Integration**
  - Inconvénients:
    - Requiert la correspondance d'une source à l'autre
    - Surtaxe les système sources
    - Plus limité que l'ETL dans la quantité de données pouvant être traitée
    - Transformations limitées sur les données
    - Peut consommer une grande bande passante du réseau
    - Le traitement en-ligne des données peut cependant entraîner des délais importants