Intégration de données

Dan VODISLAV

CY Cergy Paris Université

Master M2 Recherche SIC

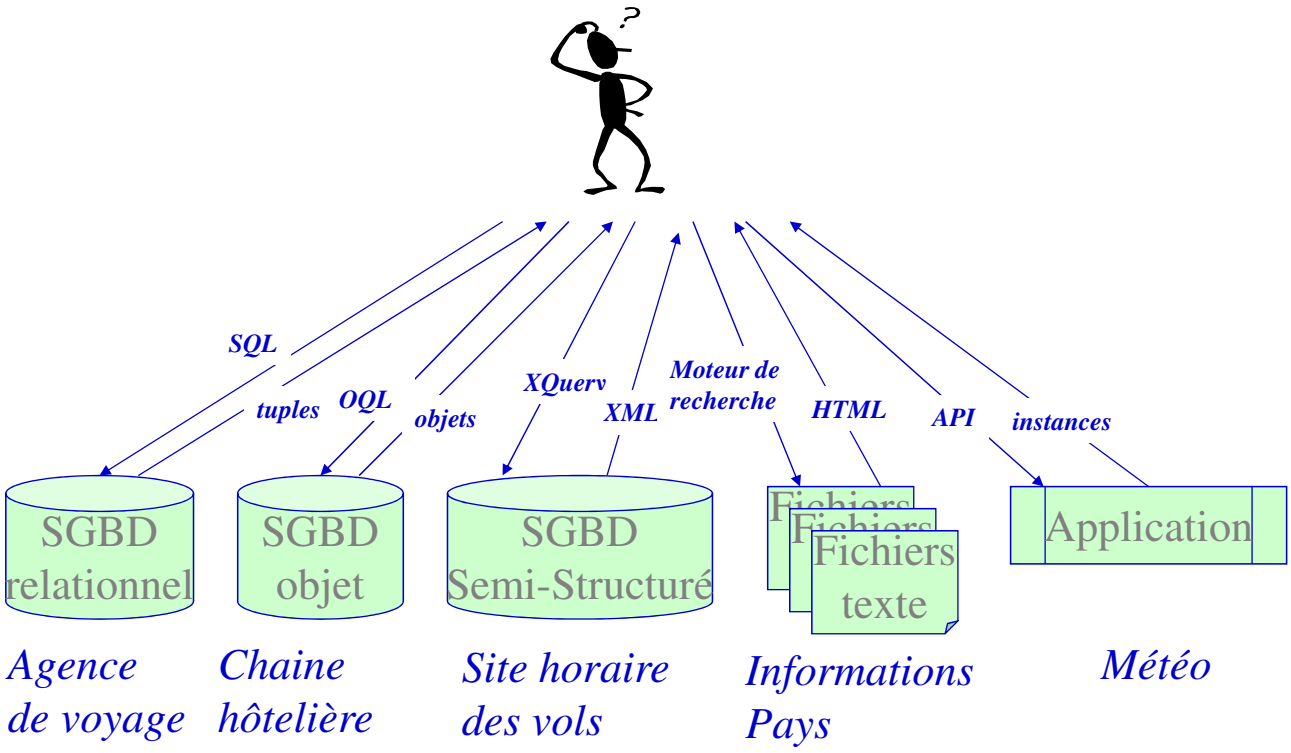
Plan

- Objectifs, principes, enjeux, applications
- Architectures d'intégration de données
  - Médiateurs et entrepôts
  - Traitement des requêtes
- Schémas d'intégration
  - Global-as-view
  - Local-as-view

Intégration de données

- Contexte
  - Sources d'information nombreuses et variées
    - SGBD relationnels/XML, pages HTML, LDAP, tableurs, fichiers, applications, …
  - Interfaces d'accès variées
    - Langages d'interrogation: SQL, XPath, XQuery, URL, …
    - Modèle de données: relationnel, XML, HTML, tableurs
    - Protocoles de communication: JDBC, ODBC, SOAP, HTTP
    - Interfaces d'appel: ligne de commande, API, formulaire, interface graphique
- *Objectif général* : utiliser plusieurs sources comme si elles constituaient une seule base de données homogène → ***l'intégration de données*** doit fournir
  - *un accès* (requêtes, éventuellement mises-à-jour)
  - *uniforme* (comme si c'était une seule BD homogène)
  - *à des sources* (pas seulement des BD)
  - *multiples* (déjà deux est un problème)
  - *autonomes* (sans affecter leur comportement, indépendant des autres sources ou du système d'intégration)
  - *hétérogènes* (différents modèles de données, schémas)
  - *structurées* (ou semi-structurées)

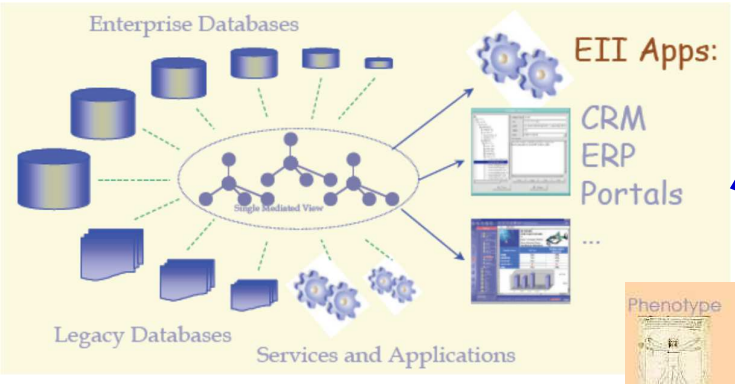
Exemple



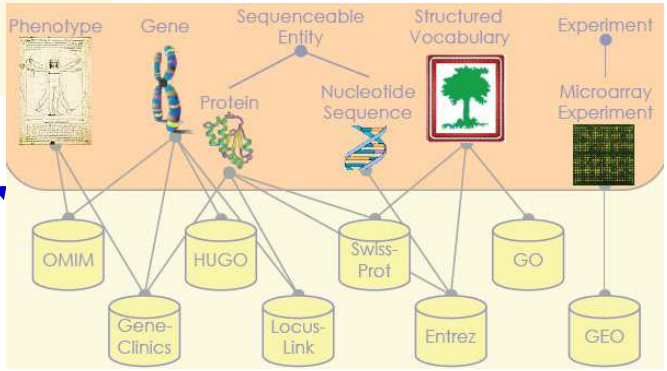
Enjeux

- Dans l'entreprise
  - Données dispersées dans une grande variété de sources hétérogènes:
    - *internes* à l'entreprise (protégées)
    - *externes*, chez des fournisseurs, des partenaires ou des clients
  - Objectif « *business intégration* »: accès *efficace*, *facile* et *sûr* à ces données
  - Études: une partie très importante des budgets IT sont dépensés en intégration
- Grand public
  - Accès simple, rapide et efficace aux informations disponibles sur le web
    - Texte/HTML, images, vidéo, XML, fils RSS, cartes
    - Le web caché, services web
  - Commerce électronique: comparateurs de prix, intégration de magasins en ligne

Applications



Données d'entreprise



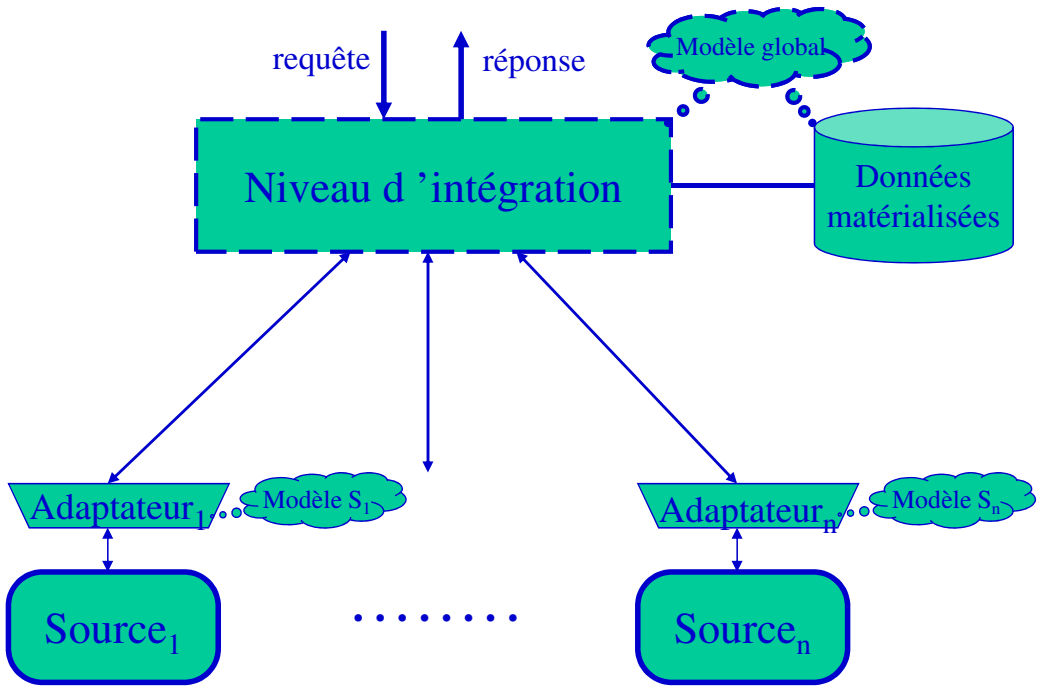
Données d'une communauté

+ **le Web !** → moteurs de recherche, agrégateurs

Caractéristiques des sources de données

- … qui rendent l'intégration de données difficile
  - → *Distribution*
    - Répartition géographique des sources sur le réseau
    - Échelle
  - → *Autonomie*
    - Les sources décident de ce qu'elles partagent, comment et quand
  - → *Hétérogénéité*
    - De format, de structure, de mode d'accès, de capacité de traitement

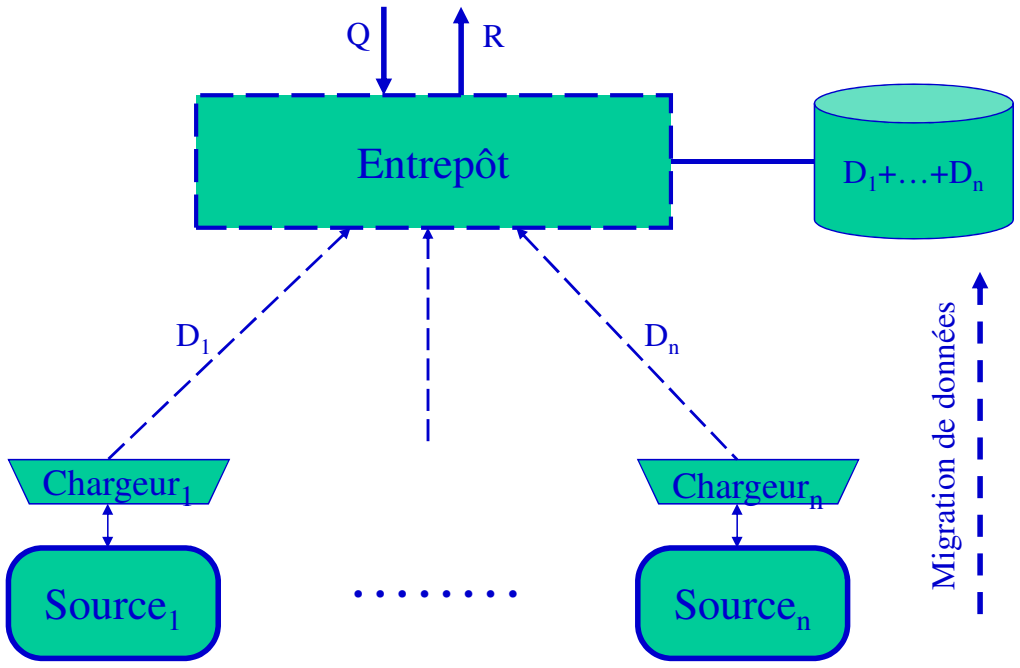
Architecture générale d'intégration



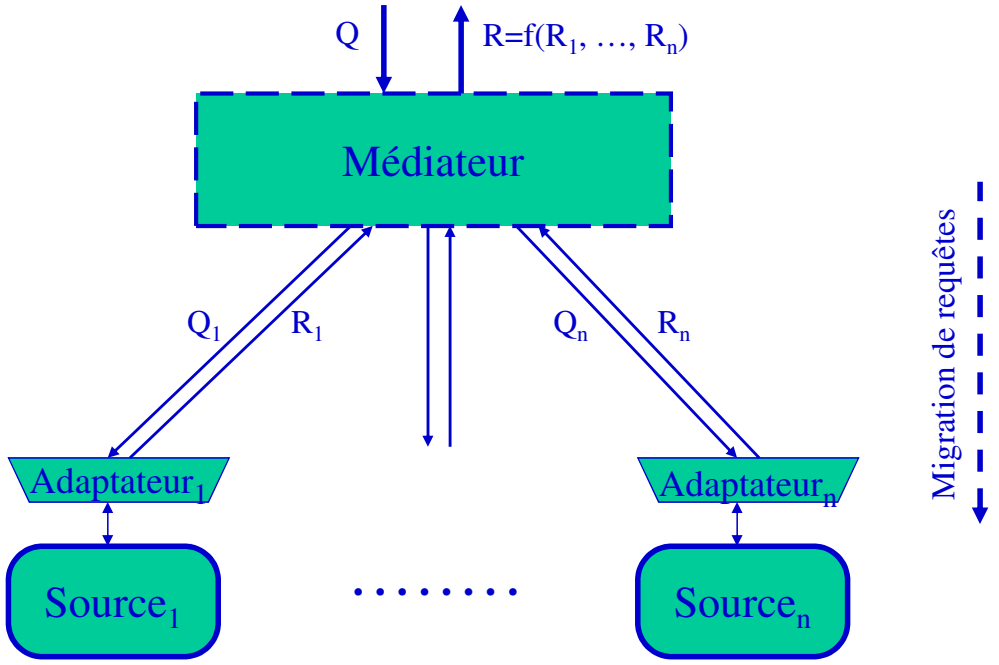
Intégration matérialisée et virtuelle

- Intégration matérialisée → ***entrepôt de données***
  - Les données provenant des sources sont transformées et stockées sur un support spécifique (entrepôt de données).
  - L'interrogation s'effectue comme sur une BD classique
- Intégration virtuelle → ***médiateur***
  - Les données restent dans les sources
  - Les requêtes sont exprimées sur le schéma global, puis décomposées en sous-requêtes sur les sources
  - Les résultats des sources sont combinés pour former le résultat final
- En pratique on peut avoir des architectures intermédiaires, entre ces deux extrêmes

Architecture d'entrepôt



Architecture de médiation



Entrepôt ou médiateur?

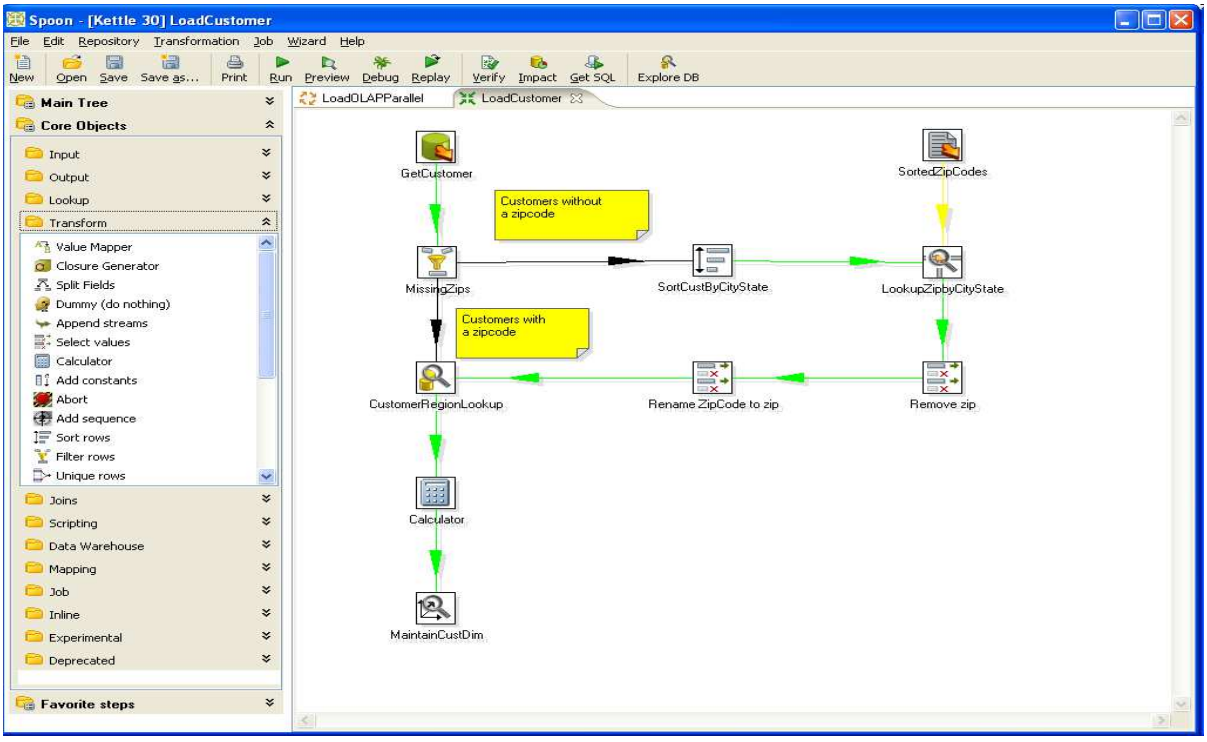
- Médiateur : accès direct aux sources
  - approche « paresseuse », pas de matérialisation
  - migration de requêtes vers les sources
  - *avantages* : données toujours fraîches, plus facile d'ajouter de nouvelles sources, plus grande échelle, distribution de l'effort
  - *inconvénients* : performances, traduction de requêtes, capacités différentes des sources

- Entrepôt de données : accès efficace à une copie des données
  - matérialisation des sources au niveau du modèle global
  - migration de données vers l 'entrepôt
  - *avantages* : performances, personnalisation des données (nettoyage, filtrage), versions
  - *inconvénients* : données pas toujours fraîches, cohérence, gestion des mises-à-jour, gestion de gros volumes de données

Entrepôts de données

- L'approche la plus populaire d'intégration de données
  - Gros avantages: performances, contrôle plus facile à réaliser sur l'hétérogénéité des données
- Utilisation pour les systèmes décisionnels OLAP
- Transformation de données pour alimenter l'entrepôt
  - Chargeurs = *systèmes ETL* (« Extract, Transform, Load »)
  - Outils graphiques pour définir *des flots de traitements/transformations*
  - Une fois le flot de traitement défini → appliqué au contenu des sources

Exemple d'interface graphique d'ETL

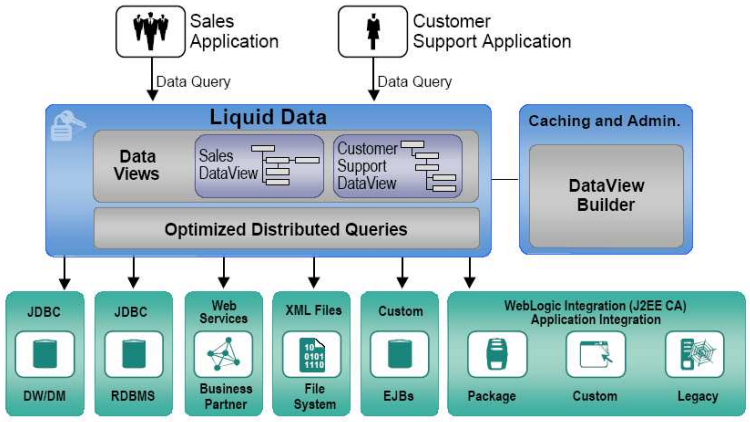


Médiateurs

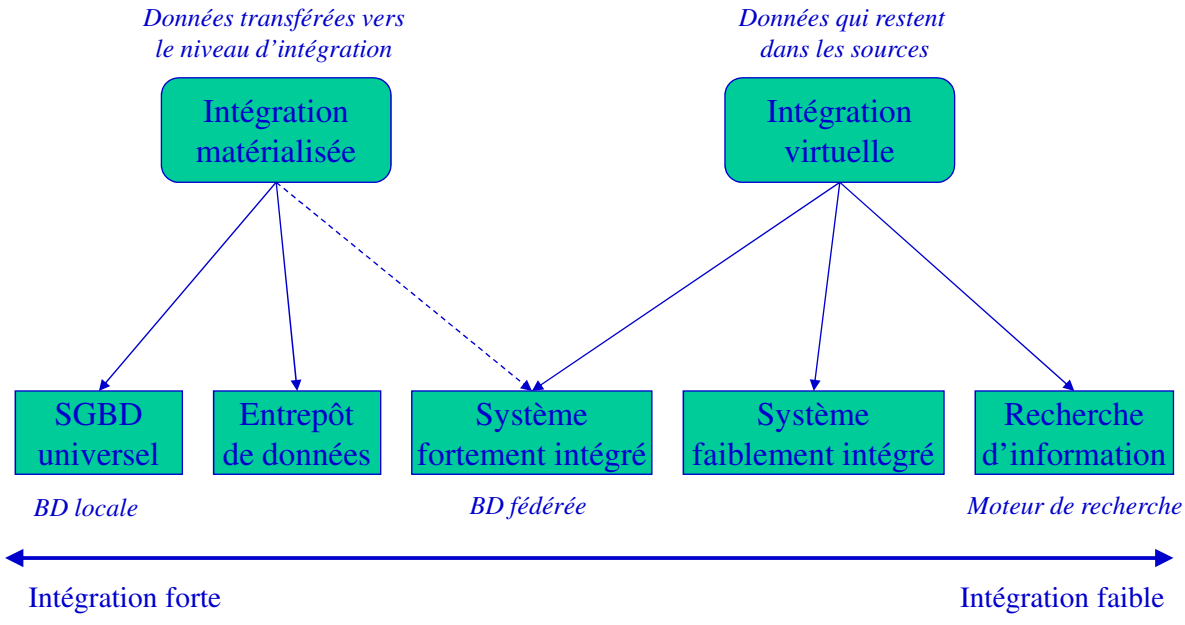
- Bien que moins utilisés en pratique, ils ont plus de potentiel
  - Meilleur passage à l'échelle
  - Acceptent mieux les changements dynamiques (nouvelles sources)

  → mieux adaptés à l'intégration de sources web

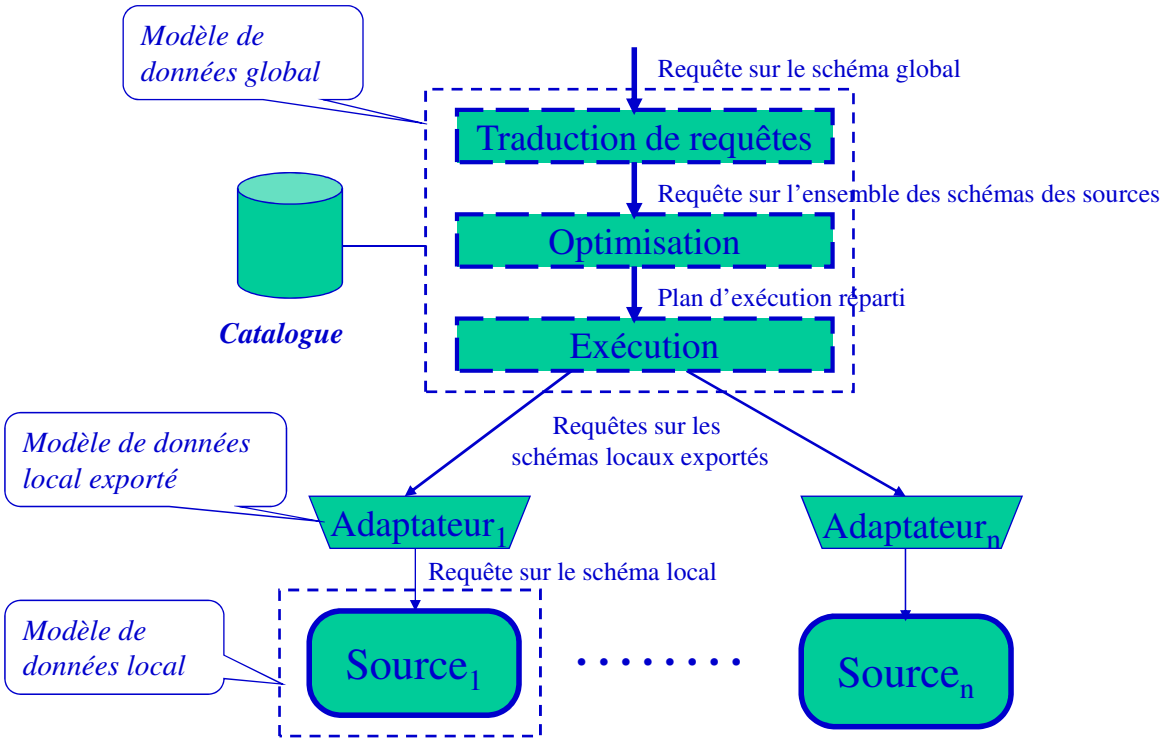
- En entreprise: EII
  « Enterprise Information Integration »
  - Ex. BEA Liquid Data, IBM Websphere Information Integrator



Degré d'intégration des données

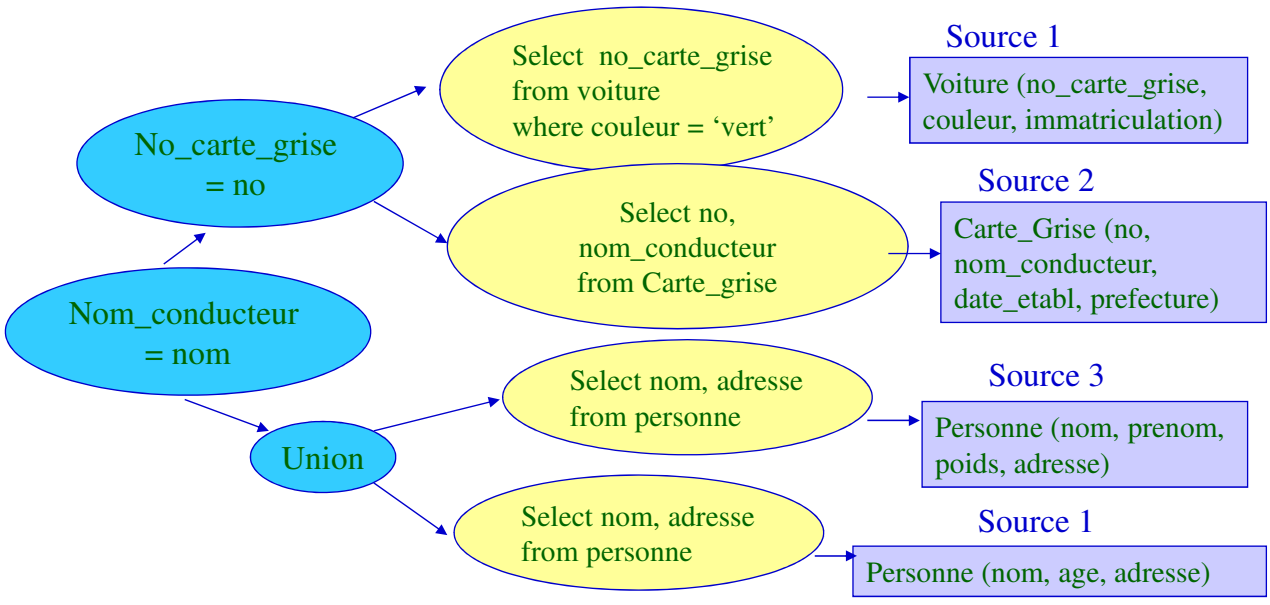


Architecture plus détaillée



Décomposition des requêtes

- Exemple : chercher l'adresse de tous les propriétaires de voitures vertes



Schémas d'intégration

- Problèmes
  - *Intégration de schéma*: comment définir un schéma global d'intégration à partir des schémas des sources?
  - *Fusion de données*: comment rendre compatibles, transformer les données en provenance des sources?
  - *Mappings/vue d'intégration*: comment décrire le lien entre le modèle global et les modèles locaux des sources?

Définition de la vue d'intégration

- Le lien entre schéma global et schémas locaux est défini à travers des vues

  → Mapping entre ces schémas
- Deux façons principales de définir ce lien
  - Le schéma global en fonction des schémas locaux → « ***global as view*** »
    - Approche *ascendante*: on part des sources pour produire le schéma global
  - Les schémas locaux en fonction du schéma global → « ***local as view*** »
    - Approche *descendante*: on fixe le schéma global et on décrit les sources par rapport à ce schéma fixé

« Global-as-View »

- Le modèle global = vue sur les sources
  - élément global = f(éléments des sources)

    $\mathbf{M = V(S_1, \ldots, S_n)}$
- Avantages
  - approche naturelle
  - la traduction de requêtes se fait facilement
- Inconvénients
  - nouvelle source → modification du modèle global
    - il faut considérer l 'interaction de la nouvelle source avec les autres

« Local-as-View »

- Les sources = vues matérialisées du modèle global
  - une source décrit les données du modèle global qu'elle peut fournir
  - élément source = f(éléments modèle global)

    $\mathbf{S_i \subseteq V_i\,(M)}$
- Avantages
  - les sources sont décrites indépendamment les unes des autres
    - très simple de rajouter une nouvelle source
- Inconvénients
  - traduction de requêtes plus complexe

Exemple « Global-as-View »

- TSIMMIS (Stanford)
  - Sources : informations sur les personnes d 'une université
    - **Inf** : BDR avec des employés et des étudiants du département Informatique

      Employé(Nom, Prénom, Titre, Chef)

      Étudiant(Nom, Prénom, Année)
    - **Ann** : Annuaire pour l 'université (nom, département, catégorie, e-mail, …)
  - Médiateur : les personnes du département Informatique
    - nom, catégorie, titre, chef, e-mail, année, ...
  - langage de spécification de médiateur MSL
    - règles : PM :- P_1, …, P_k , avec PM, Pi « patterns »

TSIMMIS : modèle

Adaptateur **Inf**

```
<employe>
   <nom>Dupont</nom>
   <prenom>Michel</prenom>
   <titre>professeur</titre>
   <chef>Jean Martin</chef>
</employe>
<etudiant>
   <nom>Hugo</nom>
   <prenom>Victor</prenom>
   <annee>2</annee>
</etudiant> …
```

Adaptateur **Ann**

```
<personne>
   <nom>Michel Dupont</nom>
   <dept>Informatique</dept>
   <categ>employé</categ>
   <email>md@univ.fr</email>
</personne>
<personne>
   <nom>Zoé Durand</nom>
   <dept>Informatique</dept>
   <categ>étudiant</categ>
   <annee>3</annee>
</personne> ...
```

Médiateur

```
<pers_inf>
   <nom>Michel Dupont</nom>
   <categorie>employé</categorie>
   <titre>professeur</titre>
   <chef>Jean Martin</chef>
   <email>md@univ.fr</email>
</pers_inf> ...
```

Spécification MSL du médiateur

```
<pers_inf>
   <nom>N</nom>
   <categorie>C</categorie>
   Reste1 Reste2
</pers_inf>  :-
   <personne>
      <nom>N</nom> <dept>Informatique</dept>
      <categ>C</categ> Reste1
   </personne>@Ann   AND
   <C>
      <nom>NF</nom><prenom>P</prenom> Reste2
   </C>@Inf  AND
   decomp(N, NF, P)
```

TSIMMIS : requêtes

- Exemple de requête
  - trouver toutes les informations sur Michel Dupont

    <pers_inf> <nom>Michel Dupont</nom></pers_inf>@Med

  - substitution des éléments de la requête dans la définition du médiateur

<pers_inf> <nom>Michel Dupont</nom> <categorie>C</categorie> Reste1 Reste2 </pers_inf> **:-**

<personne>
<nom>Michel Dupont</nom> <dept>Informatique</dept> <categ>C</categ> Reste1
</personne>**@Ann** AND

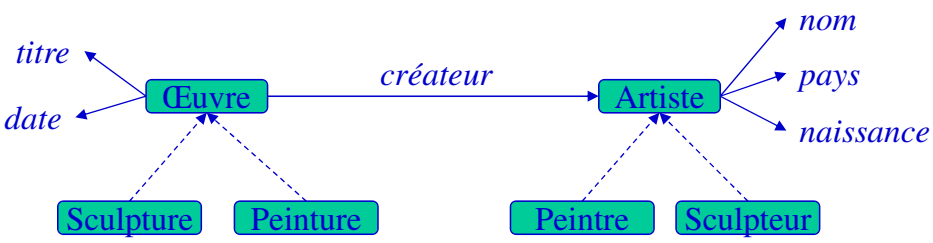
<C>
<nom>NF</nom><prenom>P</prenom> Reste2
</C>**@Inf** AND

decomp("Michel Dupont", NF, P)

  - chaque source répondra à la sous-requête qui la concerne

Exemple « Local-as-View »

- Information Manifold (AT&T)
  - modèle global : de type entité – association, exprimé par des relations
- Exemple de modèle global
  - Œuvre(titre, date, créateur), Artiste(nom, pays, naissance)
  - Sculpture, Peinture < Œuvre (sous-classes de Œuvre)
  - Peintre, Sculpteur < Artiste (sous-classes de Artiste)
    - Sculpture(titre, date, créateur), Peinture(titre, date, créateur),
    - Peintre(nom, pays, naissance), Sculpteur(nom, pays, naissance)



Information Manifold : sources

- Sources : vues sur le modèle global
  - définition = requête conjonctive + inégalités
- Exemple de description de sources

  S_1 : noms/dates naissance des peintres nés après 1800 et les titres/dates de leurs peintures

  $S_1(t, d, n, dn) \subseteq$ Peintre(n, p, dn), Peinture(t, d, n), $dn \geq 1800$

  S_2 : titres/dates des œuvres réalisées avant 1940 et le nom/pays de leurs auteurs

  $S_2(t, d, n, p) \subseteq$ Œuvre(t, d, n), Artiste(n, p, dn), $d \leq 1940$

  S_3 : noms et dates de naissance des sculpteurs français

  $S_3(n, dn) \subseteq$ Sculpteur(n, 'France', dn)

Information Manifold : requêtes

- Requête
  - titre/date des œuvres après 1900 + nom/date naissance de leurs créateurs

    Q(t, d, n, dn) : Œuvre(t, d, n), Artiste(n, p, dn), $d > 1900$
- Algorithme Bucket
  - identifier les sources pour chaque sous-requête (avec vérif. contraintes)
    - Œuvre(t, d, n) : $S_1(t, d, n, dn')$, $S_2(t, d, n, p')$
    - Artiste(n, p?, dn) : $S_1(t', d', n, dn)$, $S_3(n, dn)$ (p inutile, S_2 ne fournit pas dn)
  - union de toutes les combinaisons valides des sources
    - on élimine les combinaisons qui produisent des résultats déjà obtenus par ailleurs

      Q(t, d, n, dn) : $S_1(t, d, n, dn)$, $d > 1900$

      Q(t, d, n, dn) : $S_2(t, d, n, p')$, $S_3(n, dn)$, $d > 1900$

      Q(t, d, n, dn) : $S_2(t, d, n, p')$, $S_1(t', d', n, dn)$, $d > 1900$

      La combinaison S_1-S_3 est plus restrictive que S_1 seule !
- Remarque : dans GAV, les jointures entre sources sont déjà exprimées, dans LAV il faut les déduire