Intégration de données

Dan VODISLAV

Université de Cergy-Pontoise

Master Informatique M2

Plan

- Introduction
- Architectures d’intégration de données
- Schémas d’intégration
- Mappings
- Traitement des requêtes
- Exemple de système d’intégration

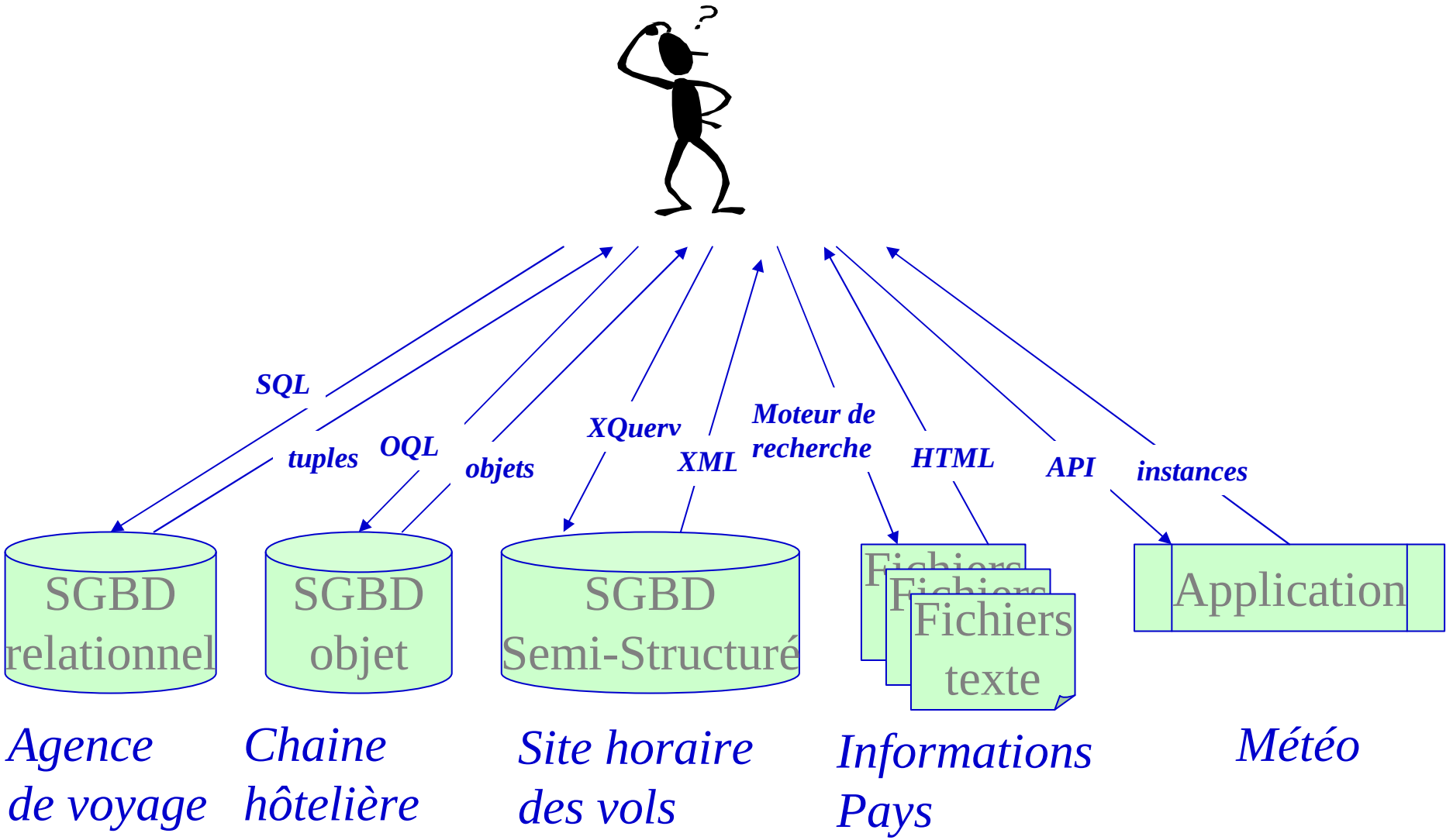
Intégration de données

- Contexte
  - Sources d'information nombreuses et variées
    - SGBD relationnels/XML, pages Web HTML, LDAP, tableurs, fichiers, applications, formulaires, services web, …
  - 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 ces données comme si elles constituaient une seule base de données homogène

Objectif

- Plus particulièrement, ***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)

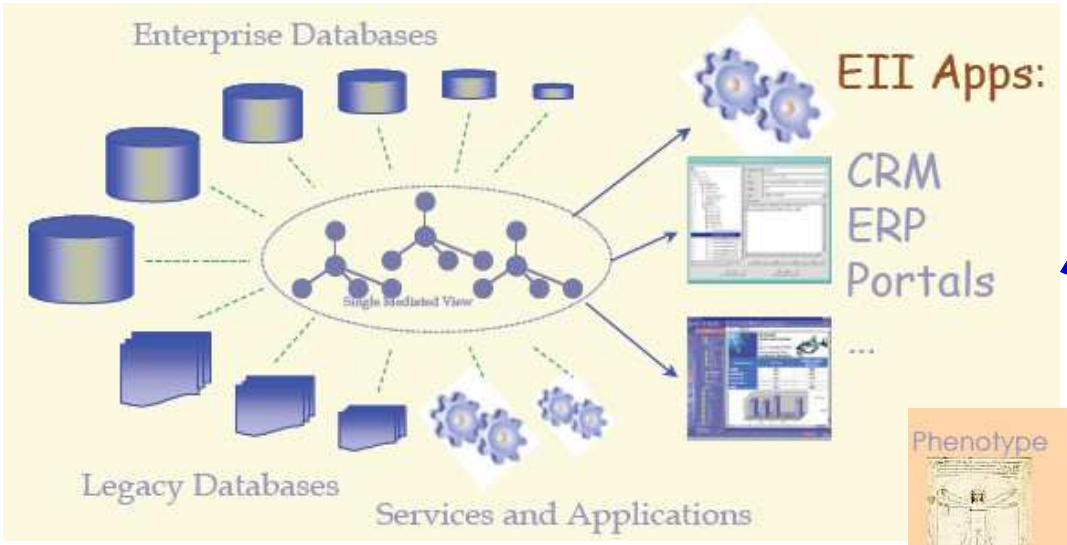
Exemple


SQL
tuples
OQL
objets
XQuery
XML
Moteur de recherche
HTML
API
instances
SGBD relationnel
SGBD objet
SGBD Semi-Structuré
Fichiers texte
Application
Agence de voyage
Chaine hôtelière
Site horaire des vols
Informations Pays
Météo

Enjeux

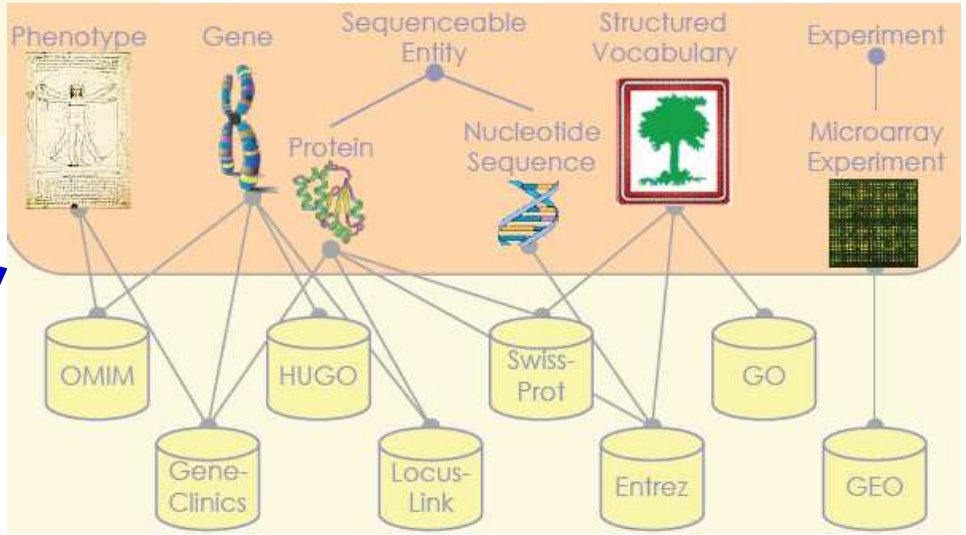
- Dans l'entreprise
  - Données dispersées dans une grande variété de sources hétérogènes:
    - *internes* à l'entreprise (protégées)
    - *externes*, chez des fournisseurs, des partenaires ou des clients
  - Objectif « *business intégration* »: accès *efficace*, *facile* et *sûr* à ces données
  - Études:
    - IBM: « pour 1$ dépensé pour une application, 5-9$ sont dépensés pour assurer son intégration »
    - Gartner: « plus de 40% des budgets IT sont dépensés en intégration »
    - Morgan Stanley: « l'intégration de données est devenue la priorité n°1 des entreprises avant le e-business et le CRM »
- 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



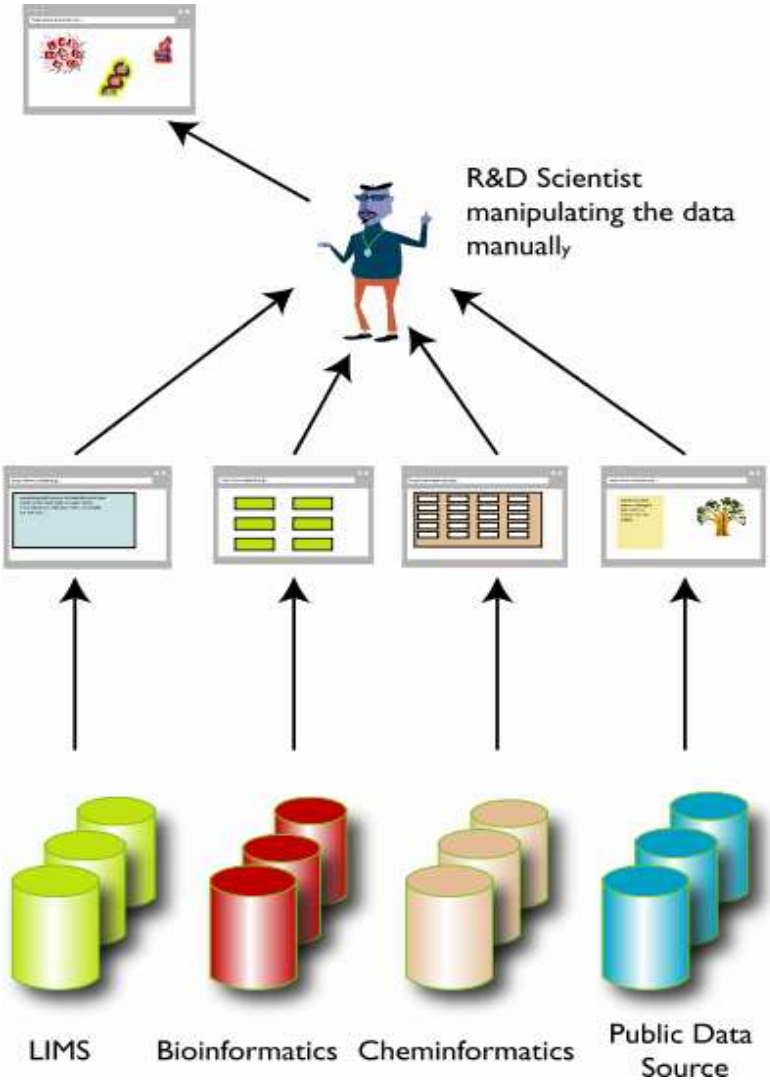
Données d'entreprise

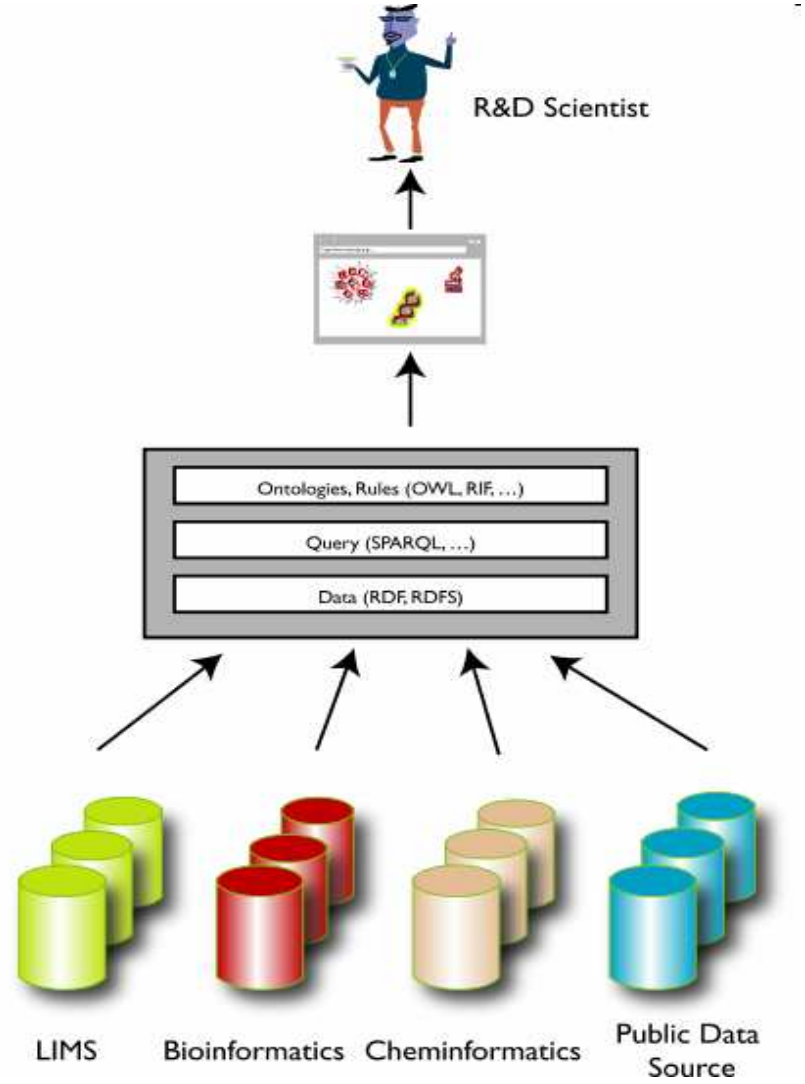
Données scientifiques



+ **le Web !** → centaines de milliers de sources de données

La différence





Caractéristiques des sources de données

- … qui rendent l'intégration de données difficile
  - → *Distribution*
  - → *Autonomie*
  - → *Hétérogénéité*

Distribution

- Les données sont stockées sur des supports répartis géographiquement
  - Caractéristique importante: *l'échelle*
- Avantages
  - Disponibilité: ne tombent pas en panne en même temps
  - Temps d'accès: partage de la charge, parallélisme
- Problèmes
  - Les temps de communication
  - Localisation des sources contenant les données pertinentes
  - Hétérogénéité en termes de puissance de traitement et de charge
  - Les sources peuvent être temporairement indisponibles

Autonomie

- Conception : les sources décident de leur propre
  - modèle de données,
  - langage d'interrogation,
  - sémantique des données.
- Communication : les sources décident quand et comment répondre aux questions d'autres sources
- Exécution : les sources décident de l'ordre d'exécution des transactions locales ou des opérations externes
  - Peu ou pas d'informations fournies sur les détails internes d'exécution
- Association des sources :
  - connexion et déconnexion des sources
  - partage de données et des fonctions

Hétérogénéité

- Concerne les données, les modèles, les langages, …
- Système homogène :
  - même logiciel gérant les données sur tous les sites
  - même modèle de données / langage d'accès
  - même univers de discours / sémantique
- Système hétérogène : qui n'est pas homogène sur au moins un critère
  - Divers niveaux et degrés d'hétérogénéité
- Aussi: hétérogénéité de la *puissance* / des *capacités de traitement* des sites

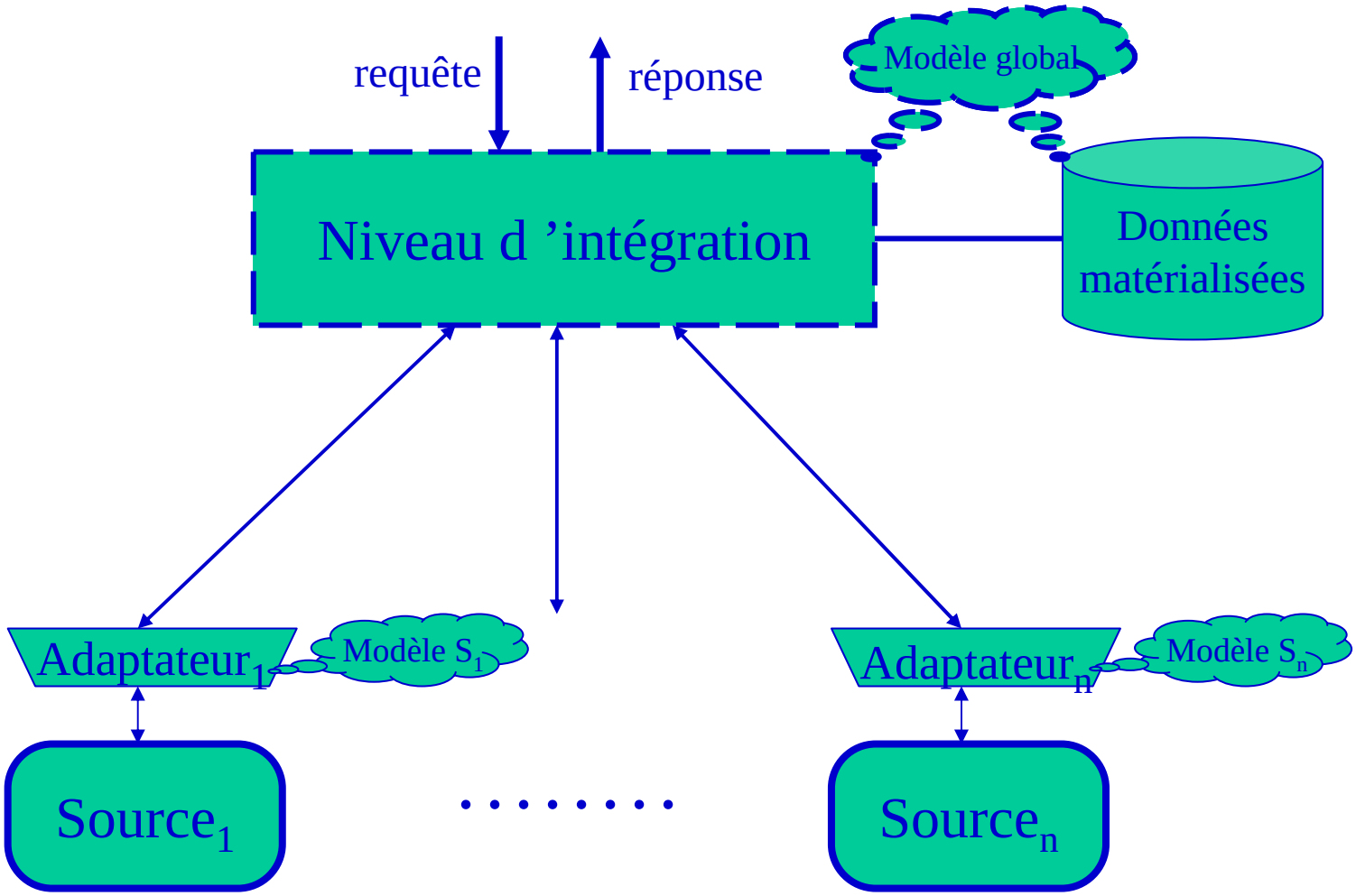
Hétérogénéité de données

- Sémantique
  - Signification, interprétation ou utilisation différente de la même donnée ou relation entre données
  - Types de relations sémantiques: identité, équivalence, compatibilité, incompatibilité

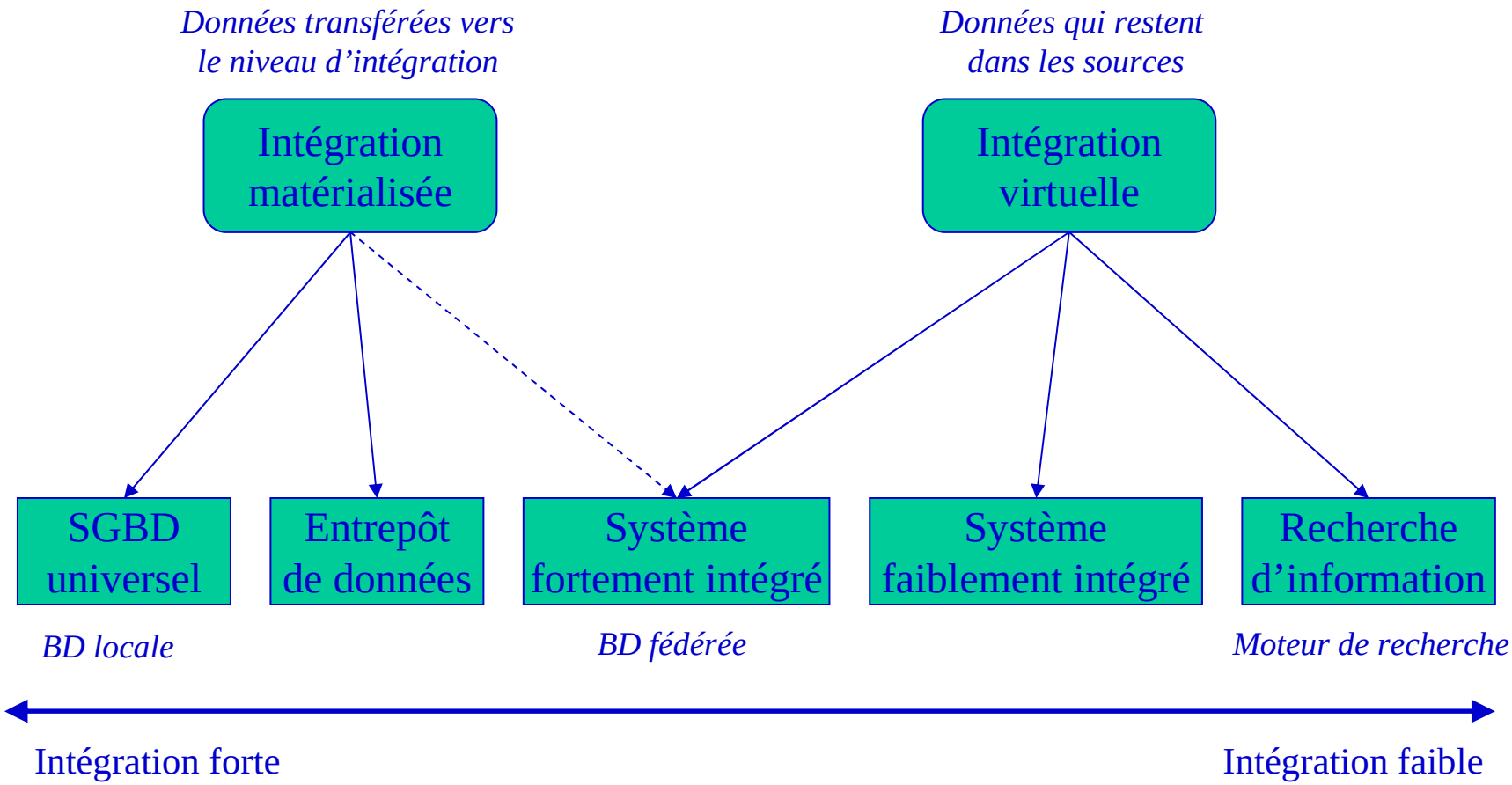
  Ex. même classe avec des extensions différentes, hiérarchies de généralisation différentes
- Structurelle
  - Représentation différente des mêmes concepts dans des bases différentes
  - Conflits de noms, types de données, attributs, unités

  Ex. structure XML différente pour un même concept, précision différente pour un type numérique

Architecture générale d'intégration



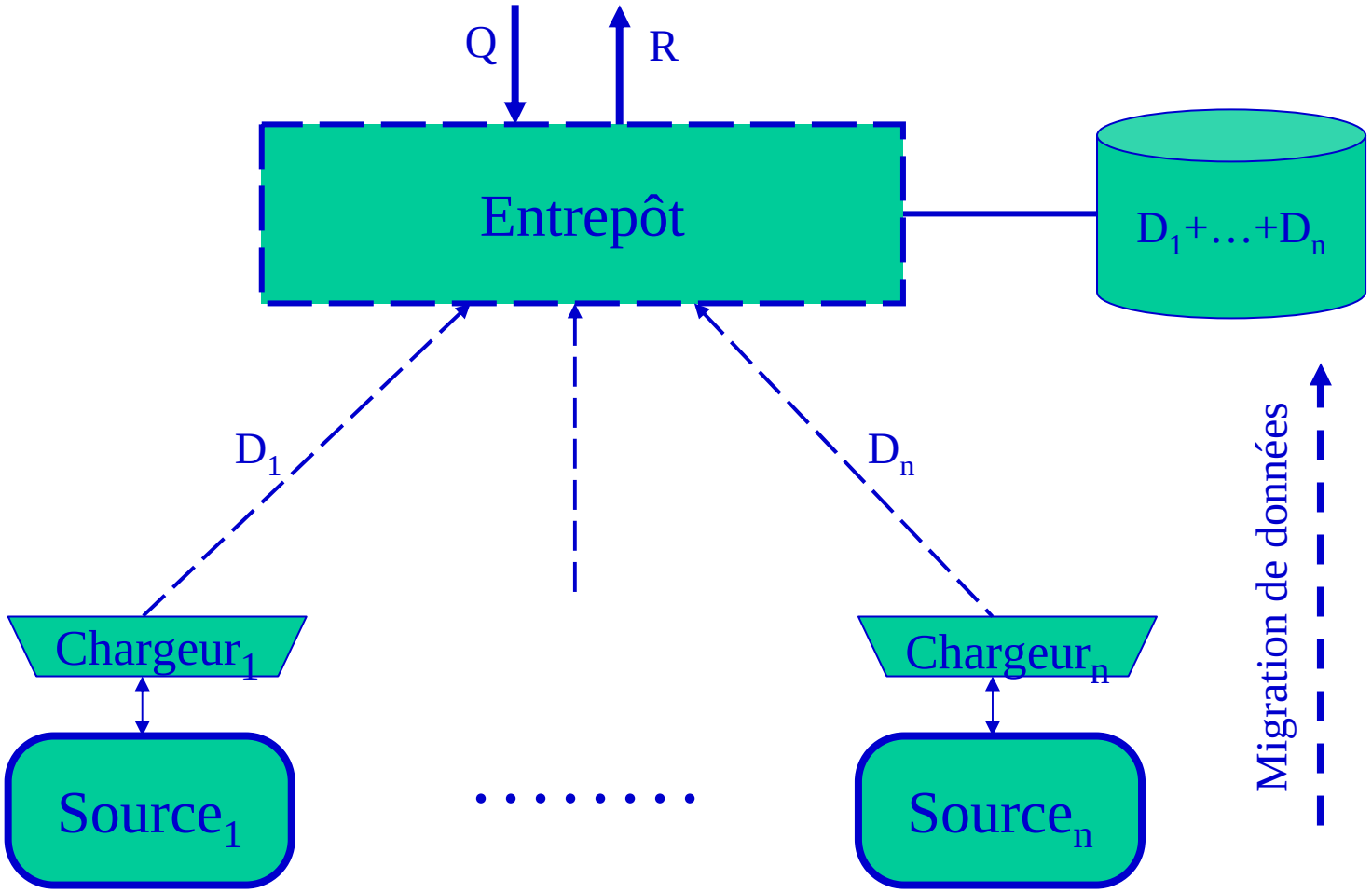
Degré d'intégration des données



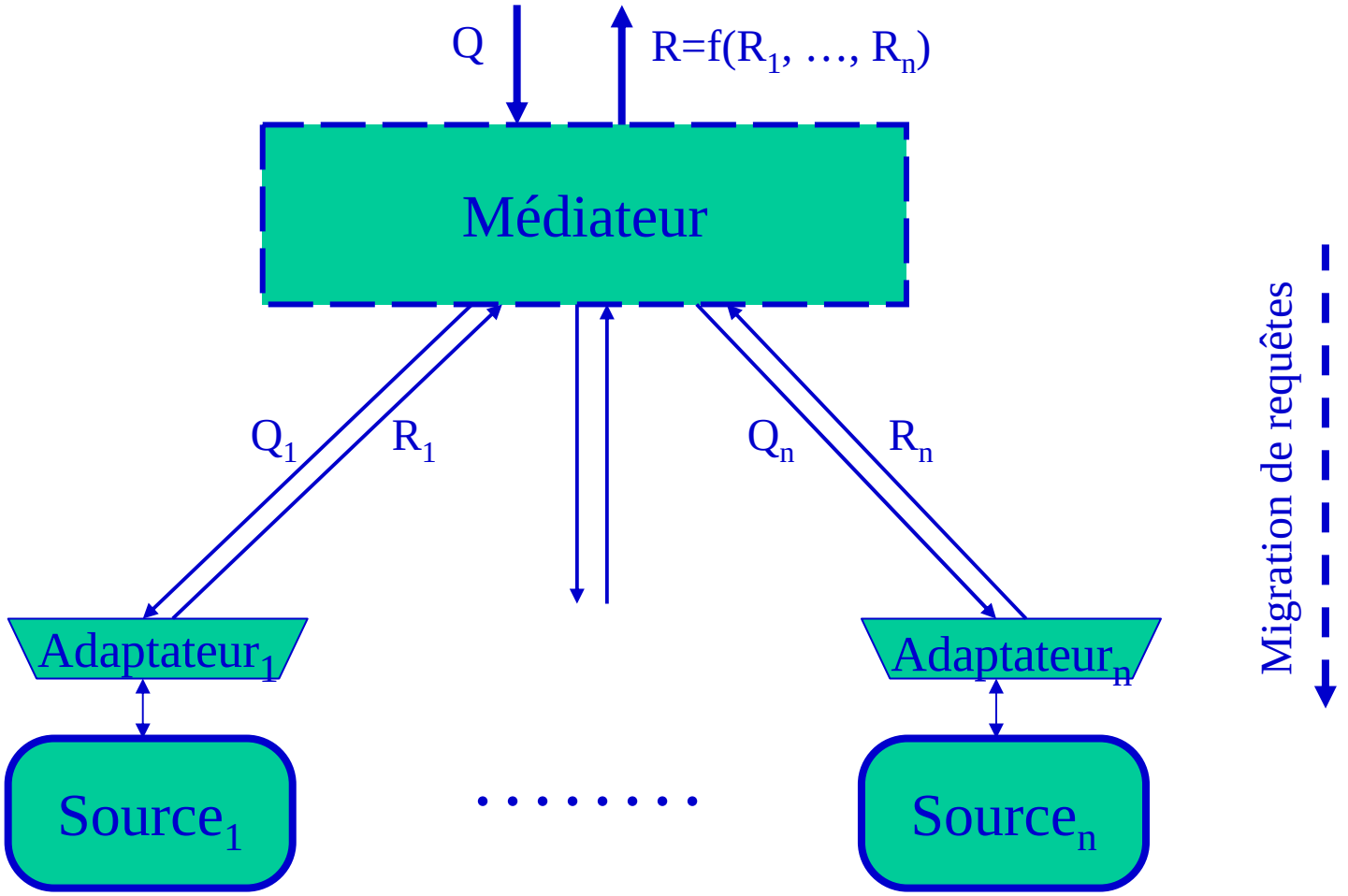
Intégration matérialisée et virtuelle

- 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?

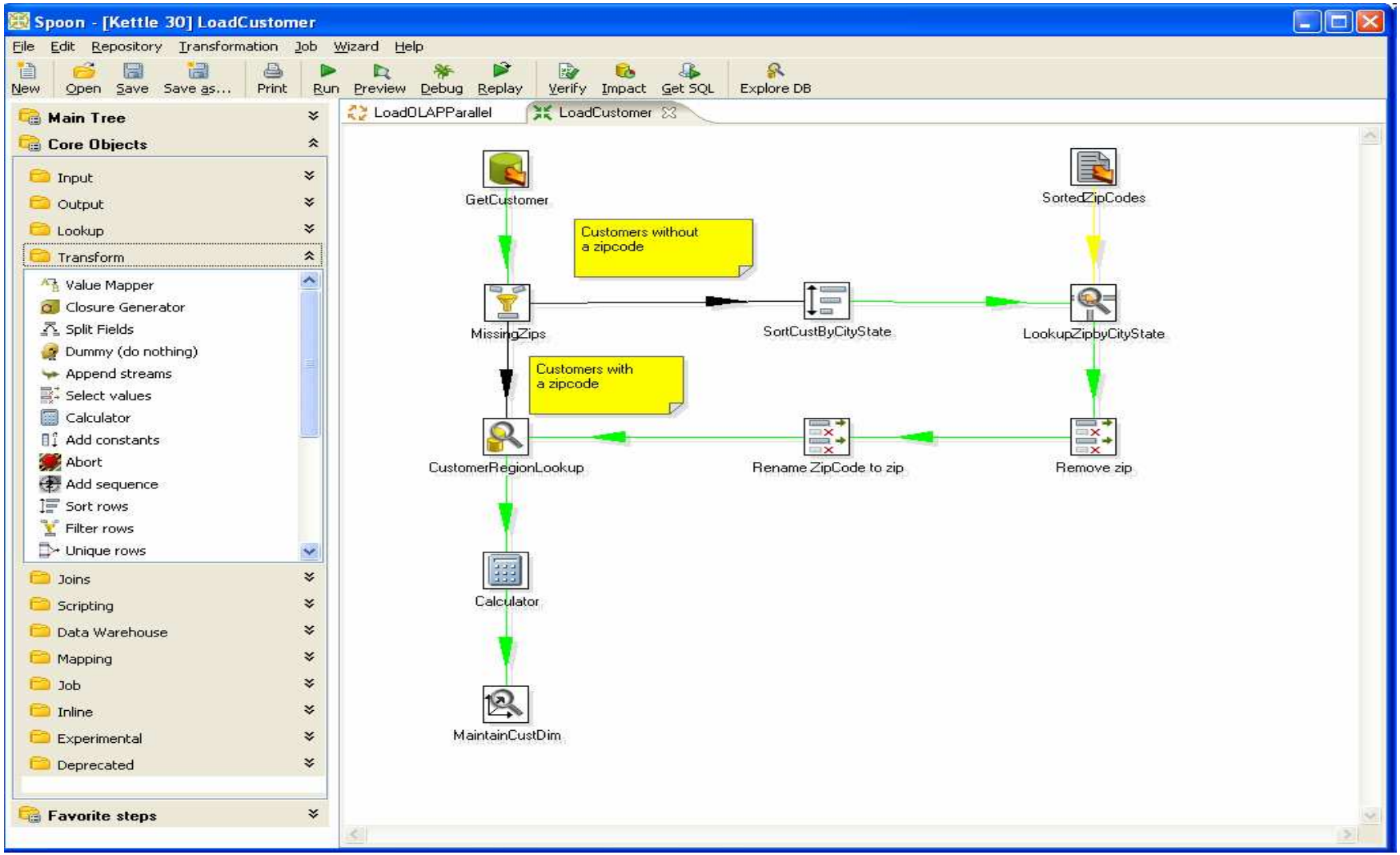
- Médiateur : accès direct aux sources
  - approche « paresseuse », pas de matérialisation
  - migration de requêtes vers les sources
  - *avantages* : données toujours fraîches, plus facile d'ajouter de nouvelles sources, plus grande échelle, distribution de l'effort
  - *inconvénients* : performances,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

- Étudiés en détail ultérieurement dans ce cours
- L'approche la plus populaire d'intégration de données
  - Gros avantage: performances
  - Autre gros avantage: 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'ETL : Kettle

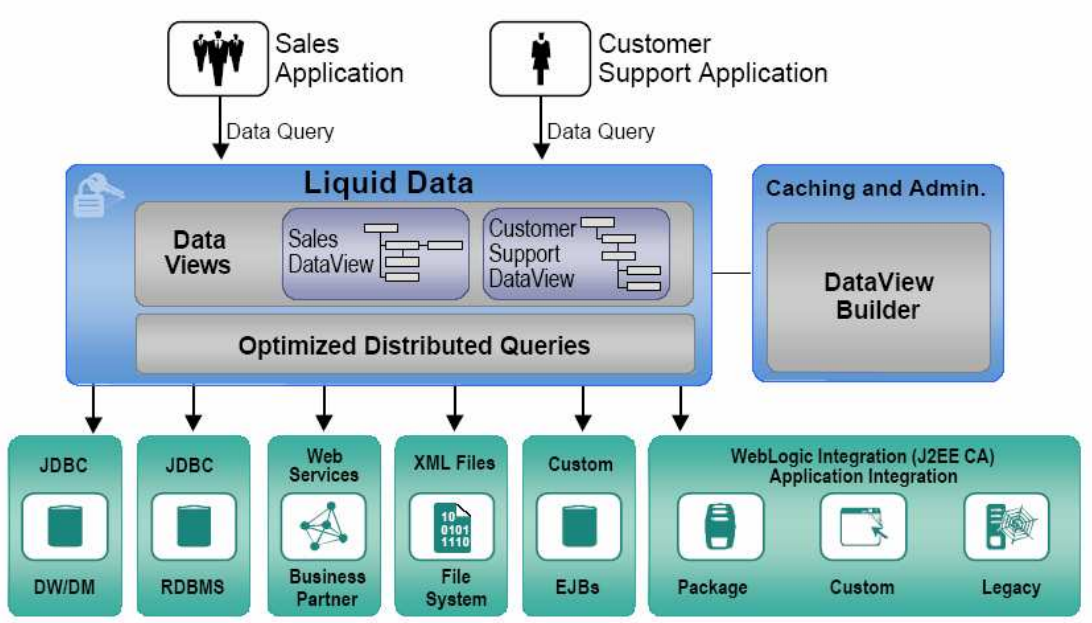


Médiateurs

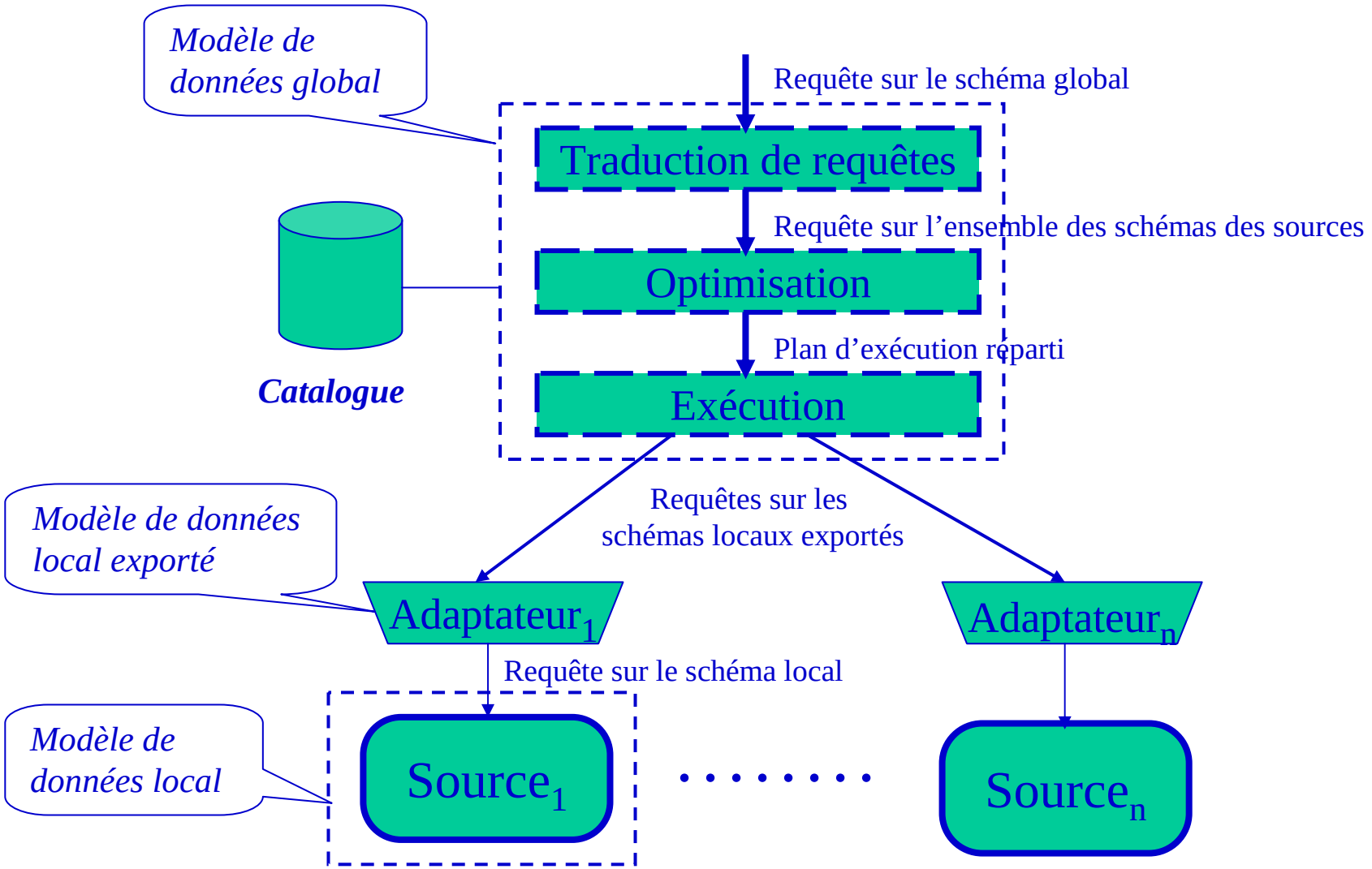
- Bien que moins utilisés en pratique, ils ont plus de potentiel
  - Meilleur passage à l'échelle
  - Acceptent mieux les changements dynamiques (nouvelles sources)

  → mieux adaptés à l'intégration de sources web

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



Architecture plus détaillée



Catalogue

- Le catalogue du médiateur contient des meta-informations:
  - le schéma global
  - les schémas externes des sources tels qu'ils sont exportés
  - des mappings entre le schéma global et celui des sources
  - des éléments de description des sources, utilisées au traitement des requêtes
    - sur le contenu, les contraintes, la complétude des sources
    - sur les capacités de traitement
    - sur la fiabilité, le temps d'accès, le débit réseau
  - des statistiques sur les données

Adaptateur

- Fonctions
  - Traduit le schéma d'une source en termes du schéma global
  - Traduit les requêtes du médiateur en termes compréhensibles par les sources
  - Traduit les résultats renvoyés par la source en termes du schéma global
- Deux façons de voir les choses
  - *Adaptateur spécifique à la source*: créé à l'initiative de la source afin d'exporter un modèle mieux adapté à l'intégration
    - Avantage: exploite au mieux les possibilités de la source
    - Inconvénient: le modèle exporté est indépendant du médiateur, il n'est pas forcément le mieux adapté à un médiateur donné

    Ex. une BD relationnelle qui exporte un modèle XML
  - *Adaptateur spécifique au médiateur*: créé à l'initiative du médiateur, afin d'adapter le contenu de la source au modèle global d'intégration
    - Quand on n'a pas d'accès aux fonctionnalités internes de la source

    Ex. extraction de données structurées (XML) d'une page/site web sur un thème donné

Fonctionnalités des médiateurs

- Adaptation des données sources
  - Traitement de l'hétérogénéité des systèmes
  - Traduction des modèles locaux en modèle global
  - Interrogation des sources via le modèle global
- Conception d'un schéma de médiation
  - Schéma exprimé dans le modèle global
  - Possibilité d'inclure des contraintes d'intégrité
- Expression de mapping
  - Vue qui exprime le lien entre le schéma global et les schémas exportés par les sources
- Exécution de requêtes
  - Optimisation et exécution de requêtes sur le schéma intégré
  - Décomposition en sous-requêtes sur les vues locales

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 schéma global et les schémas des sources?
- Modèle de données global
  - *Relationnel*: mieux maîtrisé
  - *XML*: plus riche et flexible
  - *Sémantique* (ontologie): intégration sémantique
  - *Mixte*

Définition de la vue d'intégration

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

  →Mappings 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)

    $M = V(S_1, \ldots, S_n)$
- Avantages
  - approche naturelle
  - la traduction de requêtes se fait facilement
    - « expansion » de la requête dans la vue
- 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

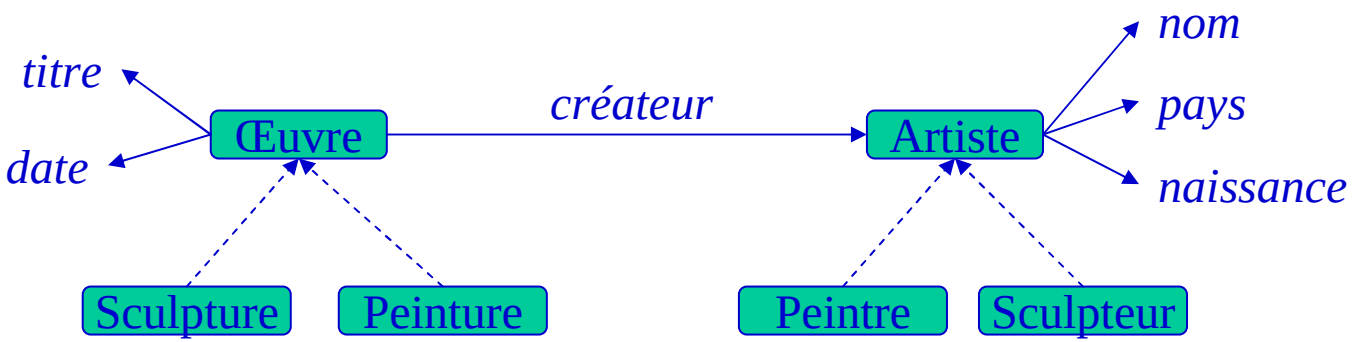
<C>
<nom>NF</nom><prenom>P</prenom> Reste2
</C>**@Inf** AND

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
  - 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)
  - union de toutes les combinaisons valides des sources

    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
- Remarque : dans GAV, les jointures entre sources sont déjà exprimées, dans LAV il faut les déduire

Mappings et vues

- Relation entre le modèle global et le modèle des sources
  - GAV : $\mathbf{M = V(S_1, \ldots, S_n)}$
  - LAV : $\mathbf{S_i \subseteq V_i\ (M)}$
- Vue = domaine + schéma + relation
  - domaine : ensemble de sources: données + modèles/schémas (d'entrée)
  - schéma : modèle de vue (de sortie)
  - relation : correspondance entre le schéma et le domaine = mapping

```
create view V as
select Prof.nom as prof, count(Cours.id) as nbcours
from Cours, Prof
where Prof.dept = "Informatique" , Prof.cours = Cours.id
groupby Prof.nom;
```

Mappings

- Mapping
  - correspondance entre le schéma global et les schémas des sources
  - utilisé pour la traduction des requêtes et la structuration des résultats
- Diversité
  - les schémas : relationnel, XML, orienté-objet, entité-association
  - les mappings : paires d'éléments, fonctions, contraintes, degrés de similarité
- Objectifs contradictoires
  - mappings complexes : puissance d'expression, précision
  - mappings simples : découverte automatique, composition, maintenance simplifiée
- Intégration : sources nombreuses, hétérogènes, ajouts de sources
  - besoin de calcul (semi-)automatique des mappings
  - vue = objet structuré (ensemble de mappings), plus facile à maintenir

$\Rightarrow$ *solutions de compromis, privilégiant les mappings simples*

Exemple de mapping

S_1 : Client — Numéro
— Société
— Nom
— Prénom

S_2 : Acheteur — ID
— Compagnie
— Contact
— Téléphone

M : Client → Acheteur
Client.Numéro → Acheteur.ID
Client.Société → Acheteur.Compagnie
Client.Nom → Acheteur.Contact
Client.Prénom → Acheteur.Contact

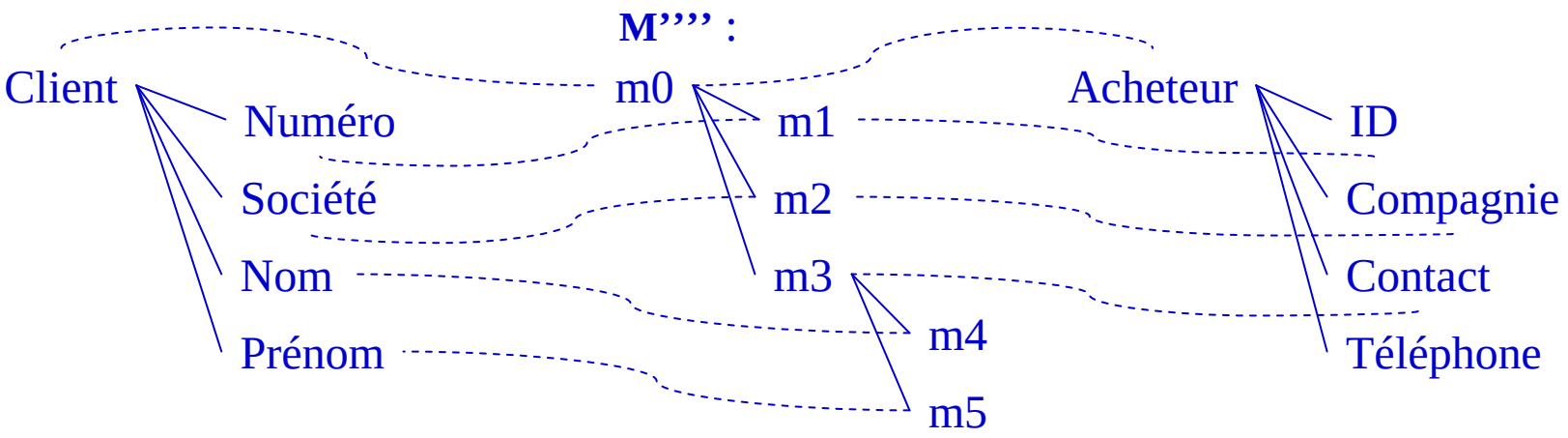
M' : Client → Acheteur
Client.Numéro → Acheteur.ID
Client.Société → Acheteur.Compagnie
concat(Client.Nom, Client.Prénom) →
Acheteur.Contact

M'' : create view Client
select ID as Numéro,
Compagnie as Société,
getNom(Contact) as Nom,
getPrenom(Contact) as Prénom
from Acheteur

M''': create view Acheteur
select Numéro as ID,
Société as Compagnie,
concat(Nom, Prénom) as Contact
from Client

Exemple de mapping (suite)

- Le même, représenté comme un objet, pas comme une correspondance



La difficulté de définir les mappings

- En pratique la création des mappings prend plus de 50% de l'effort d'intégration!
  - La précision des mappings a un impact majeur sur l'intégration
- Les difficultés
  - Hétérogénéité des schémas et des modèles
  - Les schémas ne capturent jamais complètement toute la sémantique souhaitée → il faut la chercher partout, dans les données, commentaires
  - Il faut combiner plusieurs critères, proposer plusieurs variantes avec des degrés de confiance différents
- Technique générale: deux étapes
  - Mise en correspondance d'éléments individuels du schéma («matching»)
  - Utilisation des correspondances individuelles pour définir des mappings
    - Union, jointure, filtrage

Techniques de calcul de mappings

- Utilisation du schéma uniquement
  - *linguistique* : similarité des noms/descriptions des éléments
    - égalité : stricte, forme canonique, synonymes, hypernymes
    - sous-chaînes communes, distance d'édition, similarité phonétique
    - découpage des noms composés, détection d'abréviations
  - *contraintes* : type, domaine, multiplicité, valeur clé, cardinalité relations
  - *réutilisation* : mappings déjà calculés pour des structures qui apparaissent souvent
- Utilisation des instances (données)
  - extraction de caractéristiques du schéma absentes dans sa description
  - calcul de mappings d'instances → généralisés au schéma

Techniques (suite)

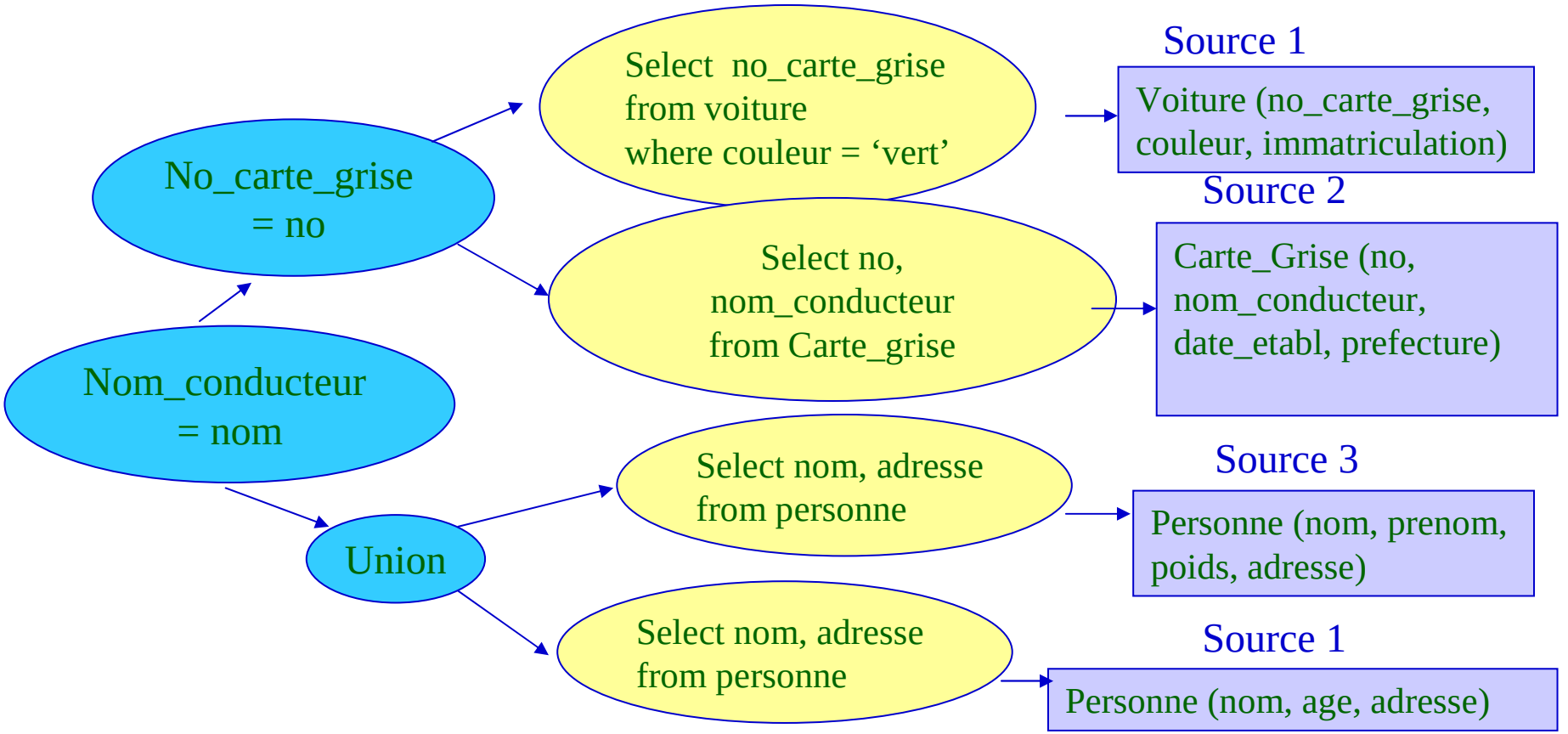
- Combinaison de plusieurs critères
  - *hybride* : algorithme qui combine plusieurs critères
  - *composition* : combinaison flexible de plusieurs algorithmes
    - poids ajustables
    - ordre d ’exécution flexible, un algorithme utilise les résultats d ’un autre
- Intervention de l ’utilisateur
  - décision sur les mappings candidats
  - nouveaux mappings loupés par le système
  - paramétrage fin de la composition d ’algorithmes

Traitement des requêtes

- Étapes:
  - Analyse syntaxique et sémantique
  - Décomposition de la requête
  - Exécution des requêtes sur les sources
    - transformation de la requête en langage global vers le langage de la source
    - transformation du résultat au format de la source vers le format global
  - Recomposition des résultats
    - combinaison des résultats locaux
- Plan d'exécution
  - décrit la méthode d'exécution d'une requête
  - souvent représenté par un *arbre algébrique* = arbre où les nœuds sont des *opérateurs algébriques* et les feuilles les *sources de données*
  - il peut exister plusieurs plans d'exécutions équivalents → *espace de recherche*
- Optimisation : choix du meilleur plan dans l'espace de recherche
  - Basée sur un *modèle de coût* et une *stratégie de recherche*

Décomposition des requêtes

- Exemple : chercher l'adresse de tous les propriétaires de voiture verte



Problèmes spécifiques aux médiateurs

- Capacités et puissance variable des sources
  - SGBD : possibilité de requêtes complexes
  - Moteur de recherche : par mots-clefs et similarités
  - Fichiers : via champ indexé

→ Le médiateur ou l'adaptateur de la source doit pallier aux déficiences des sources

- Modèle de coût

  Coût d'exécution d'une requête

  = *coût_opérations_médiateur* (classique dans les BD)

  + *coût_communication* (réseau: nombre d'accès, débit)

  + *coûts_sur_les_adaptateurs* (tenir compte de l'accès parallèle aux sources)

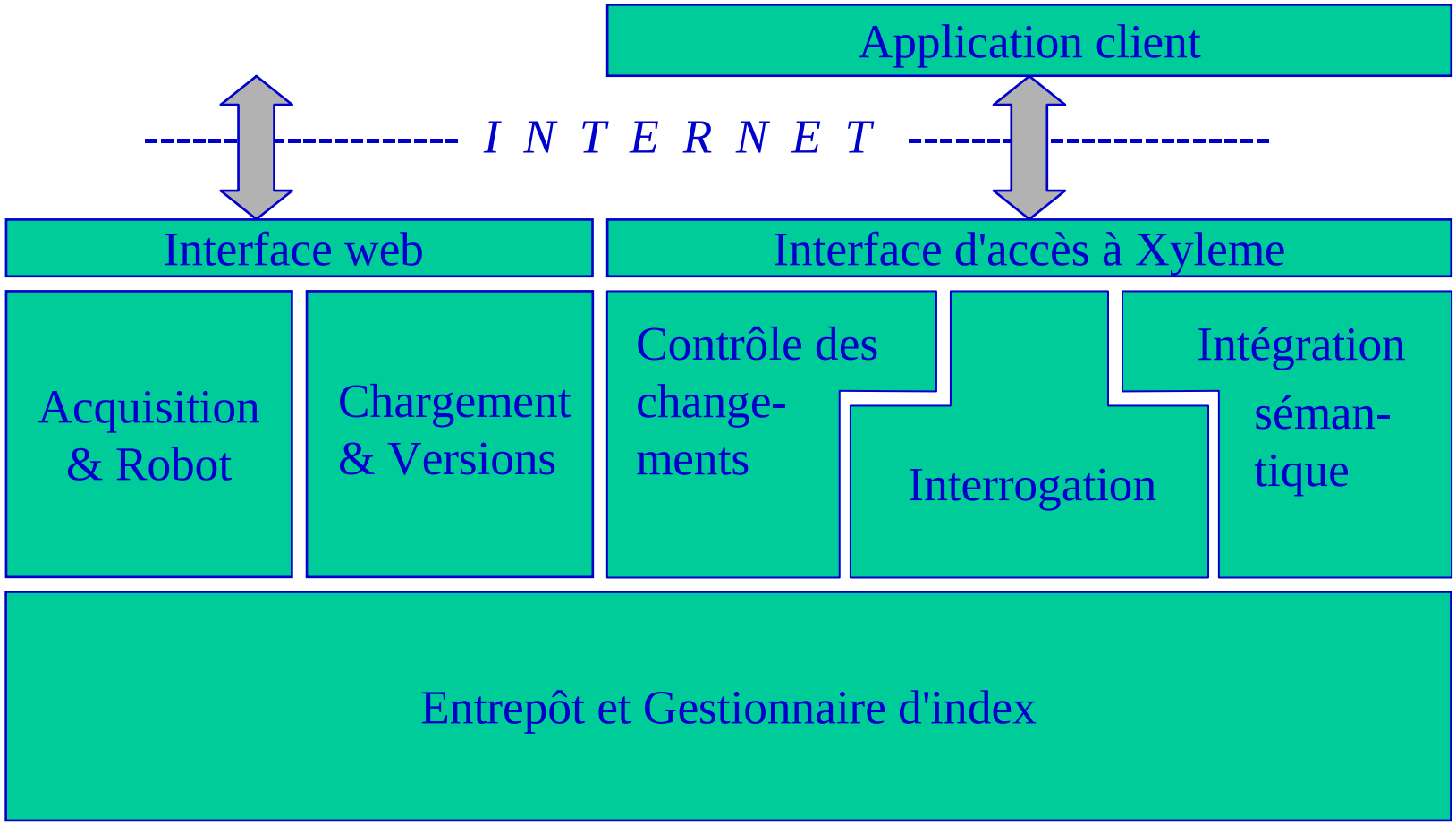
  + *congestion_du_réseau* (difficile à modéliser)

  - Coût adaptateurs/sources: difficile à connaître → apprentissage

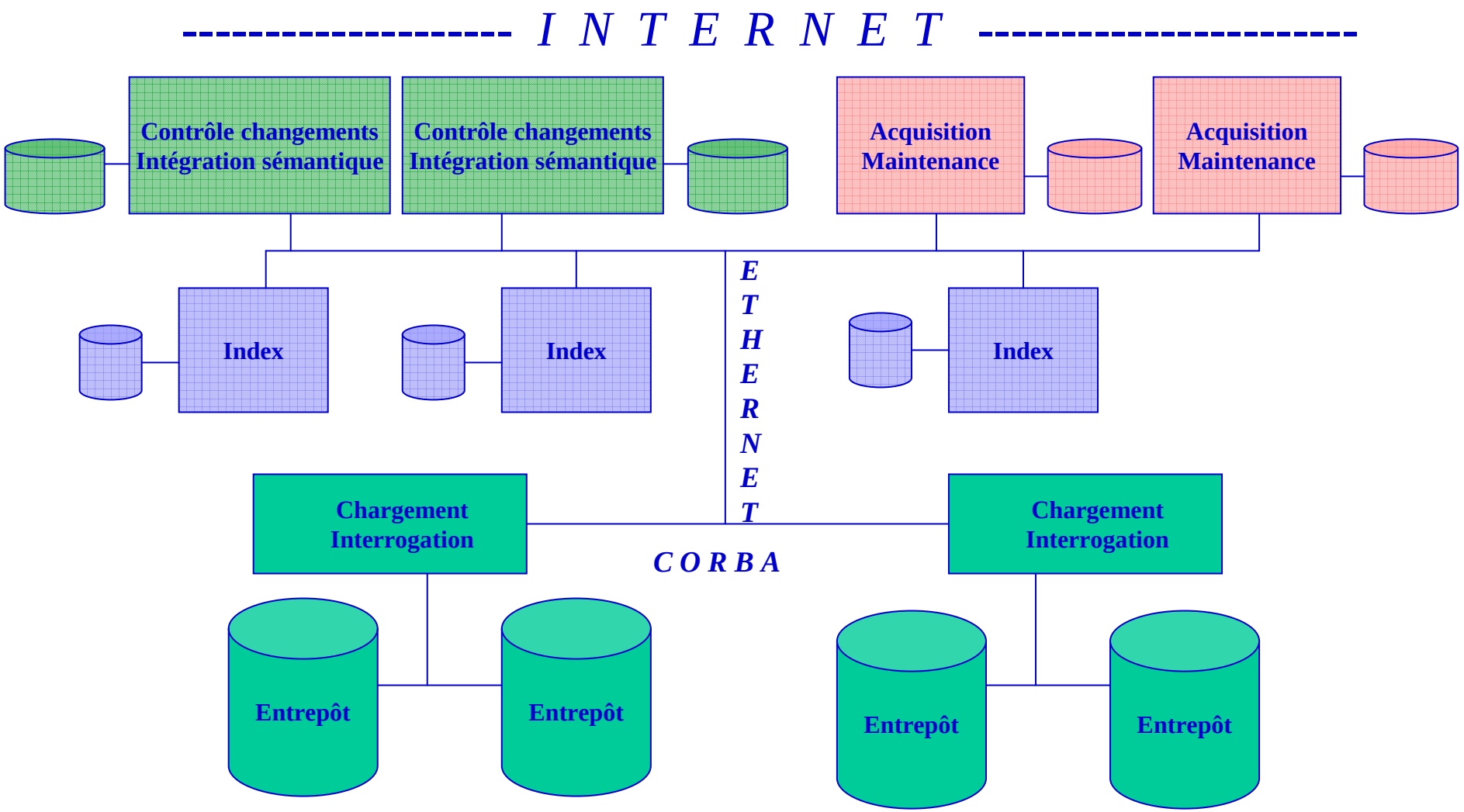
Exemple de système: Xyleme

- Entrepôt de données XML à l'échelle du web
  - projet INRIA (Verso) + LRI, CNAM, Université Mannheim
    - commercialisé par Xyleme SA
  - objectif : stocker et indexer le contenu XML du web
    - offrir divers services autour de ce contenu
    - Google XML
  - problèmes : passage à l'échelle, hétérogénéité, changements
  - architecture distribuée
- Principaux services
  - stockage XML natif, distribué
  - acquisition de documents à partir du web ou en local
  - interrogation basée sur la structure et le langage naturel
  - notification de changements, gestion de versions
  - intégration sémantique à travers des vues

Architecture fonctionnelle



Architecture physique distribuée

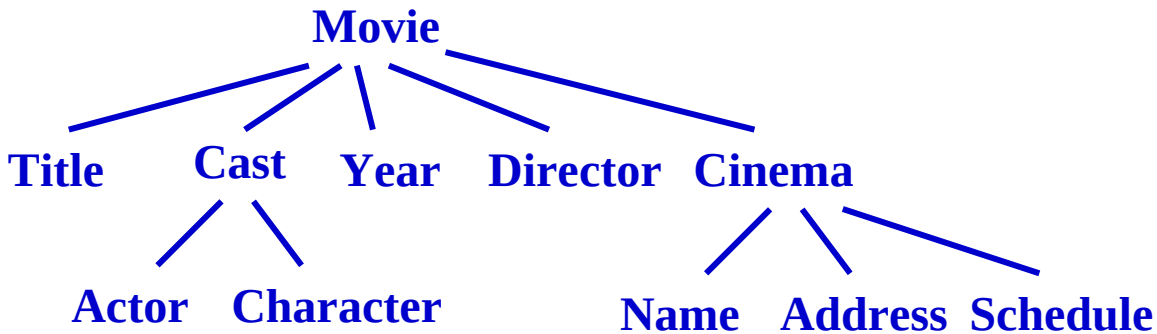


Intégration de données dans Xyleme

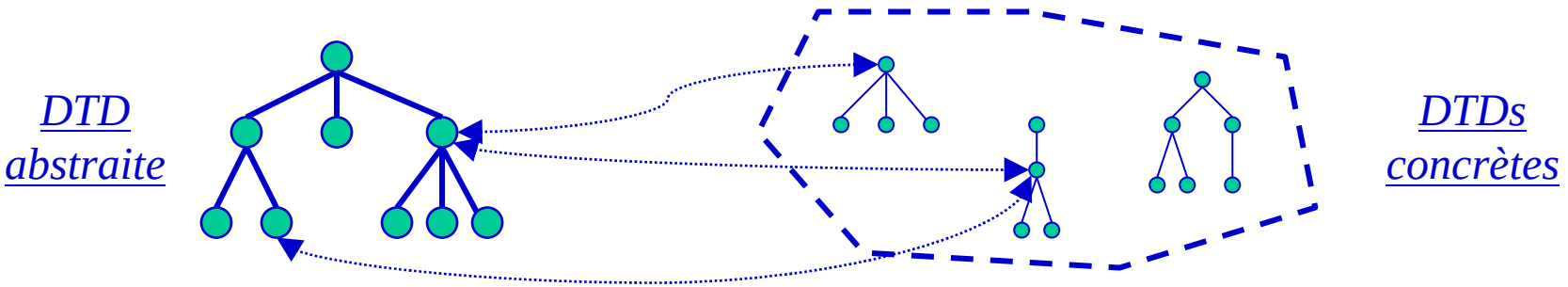
- Besoin
  - XML : liberté dans la définition des structures → hétérogénéité
  - Documents avec contenus similaires mais structures différentes
  - Interrogation :
    - autant de requêtes que de structures de document différentes
    - autant de structures de résultat que de structures de document différentes
- Problématique
  - Offrir un accès unique et homogène à l'entrepôt (requêtes et résultats)
  - ~ Médiation de sources hétérogènes
- Deux approches
  - Intégration à très large échelle (web)
  - Intégration à petite/moyenne échelle (quelques schémas)

Vue à large échelle

- Vue sur le cinéma
  - domaine **Cinema** : clusters {Films, Acteurs, Spectacles}
  - schéma:

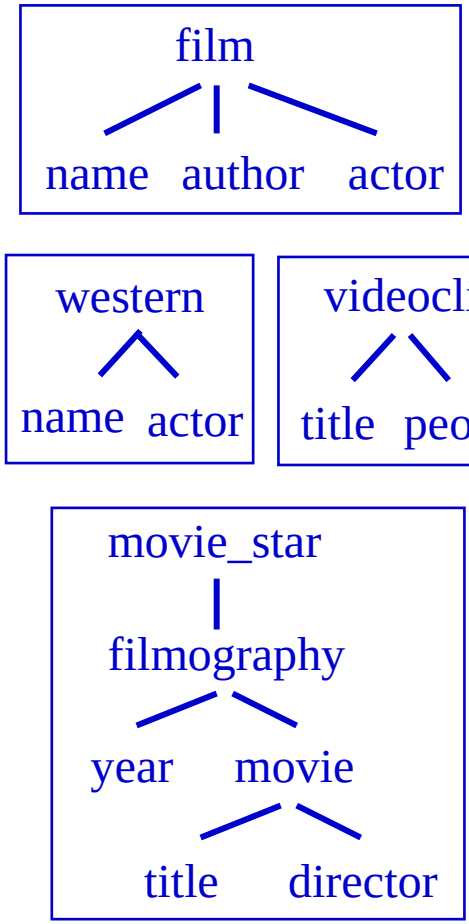


  - définition: mappings



Exemples de mappings

DTDs concrètes



Quelques mappings

Movie ↔ film
↔ western
↔ videoclip
↔ movie_star/filmography/movie

Movie/Title ↔ film/name
↔ western/name
↔ videoclip/title
↔ movie_star/filmography/movie/title

Movie/Director ↔ film/author
↔ videoclip/people
↔ movie_star/filmography/movie/director

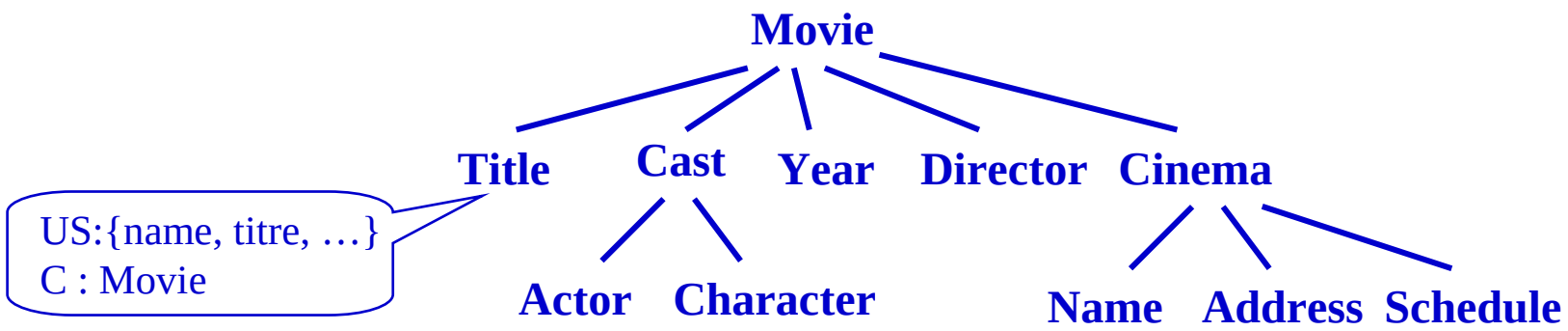
Movie/Cast/Actor ↔ film/actor
↔ western/actor
↔ videoclip/people
↔ movie_star

Génération automatique de mappings

- Absolument nécessaire
  - gestion manuelle possible pour la DTD abstraite, mais impossible pour les mappings
  - nb. de mappings proportionnel au nb. de DTDs concrètes !
- Choix: mappings chemin-à-chemin, plus adaptés à la génération automatique
- Principes de génération automatique
  - similarités entre mots
    - lexicale: racine commune, mots composés, abréviations, etc.
    - sémantique: synonymie, généralisation, etc.
  - contexte d ’interprétation : le chemin

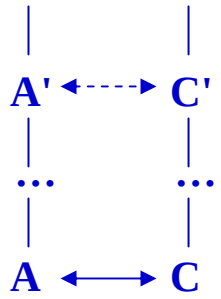
Annotation de la DTD abstraite

- Chaque concept (nœud) de la DTD abstraite
  - unité de sens : l'ensemble des mots "similaires"
    - chaque mot: décomposé, lemmatisé, info langue
  - contexte : nœud ancêtre important pour l 'interprétation



Phases

- Mise en correspondance sémantique
  - mot abstrait ↔ mot concret, à l ’aide de l ’US
  - factorisation des mots qui apparaissent souvent
    - la comparaison de mots est l'opération la plus coûteuse
- Mise en correspondance contextuelle
  - un mapping de mots → plusieurs mappings de chemins
  - le contexte permet d ’éliminer des mappings incorrects
    - si contexte(A)=A’, A ↔ C valide seulement s ’il existe un mapping de A’ vers C', un préfixe (ancêtre) de C



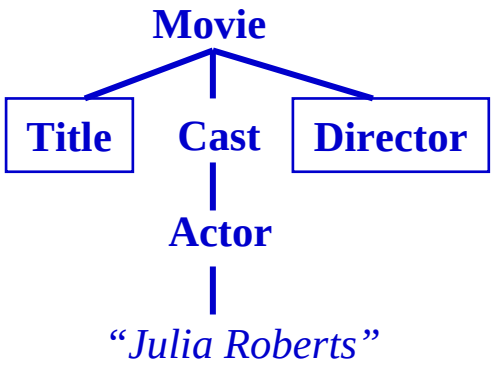
- Validation
  - aide à la découverte de mappings incorrects ou manquants
- Remarques
  - il vaut mieux ne pas perdre de mappings que d’en avoir trop
  - la traduction n’accepte que des combinaisons *valides* de mappings

Interrogation de vues

- Exemple de requête abstraite:

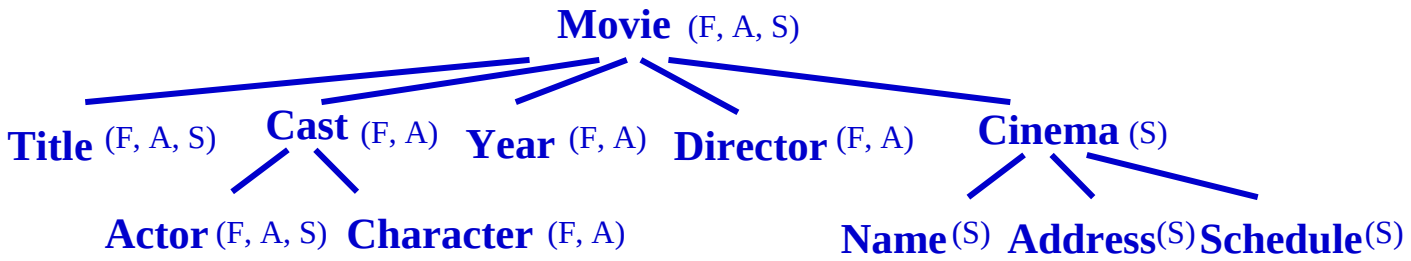
  "Trouver le titre et le metteur en scène des films où joue Julia Roberts"

select *m*/Title, *m*/Director
from *doc* **in *Cinema***, *m* in *doc*/Movie
where *m*/Cast/Actor **contains** "Julia Roberts«



- Distribution:
  - Données (clusters), mappings (par cluster)

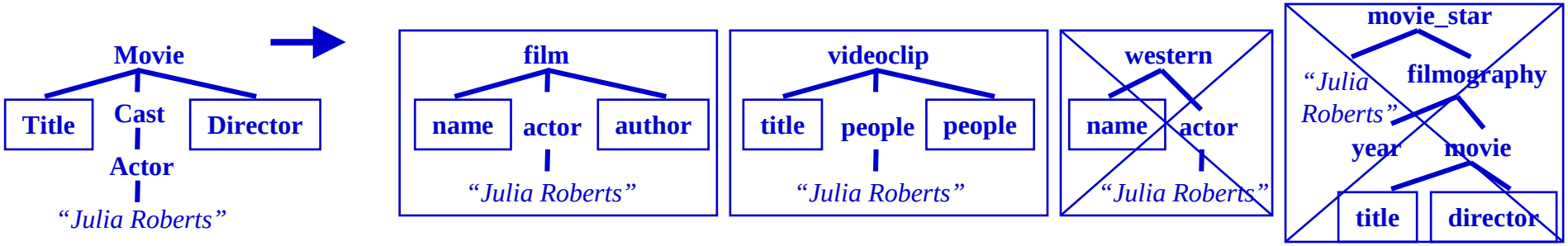
  → distribution en sous-requêtes par cluster



Clusters : F = Films, A = Acteurs, S = Spectacles

Évaluation sur chaque machine

- Traduction en requêtes concrètes à l'aide des mappings
  - Contraintes pour accélérer la traduction (préservation descendance)



- Typage des résultats : utiliser le schéma global

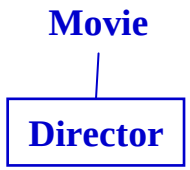
<Result>
  <name>Ocean 's Eleven</name>
  <author>S. Soderbergh</author>
</Result>

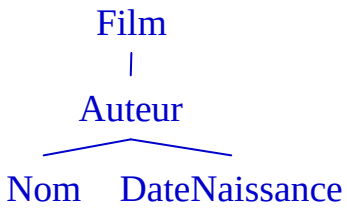
→

<Result>
  <Title>Ocean 's Eleven</Title>
  <Director>S. Soderbergh</Director>
</Result>

Typage des résultats : difficultés

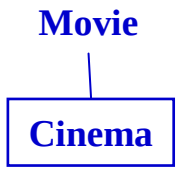
- Projection feuille abstraite

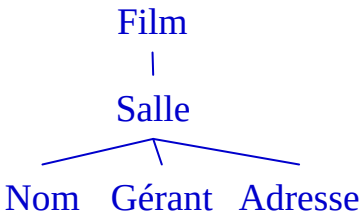


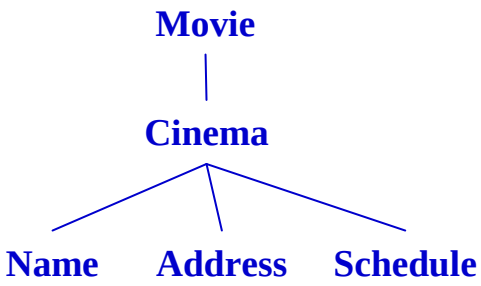



```
…
<Result>
   <Director>
      Nom: S. Kubrick
      DateNaissance: 26/07/1928
   </Director>
</Result>
...
```

- Informations en trop dans la partie concrète








```
…
<Result>
   <Cinema>
      <Name>Rex</Name>
      <Address>1 bd Poissonière</Address>
   </Cinema>
</Result>
...
```

Xyleme: vues à moyenne échelle

- A moyenne échelle les restrictions imposées à l'échelle du web ne sont plus nécessaires
- Objectif: développer des applications au dessus de données hétérogènes
  - Simplifier l'écriture des requêtes
  - Précision: traduction de requêtes sans pertes
- Système XyView
  - Vues de type « relation universelle » → interrogation très simple
  - Niveau intermédiaire de vue pour gérer l'hétérogénéité et les jointures

Exemple de documents hétérogènes

```
<GameResult>
   <WireHeading> ... </WireHeading>
   <Description> Real Madrid 1 - Valencia 0
   </Description>
   <Date> 2004-05-22 </Date>
   <Team>
      <Name> Real Madrid </Name>
      <Scored> 1 </Scored>
      <Scorer><Name> Zidane </Name>
            <Goals> 1 </Goals>
      </Scorer>
   </Team>
   <Team>
      <Name> Valencia </Name>
      <Scored> 0 </Scored>
   </Team>
</GameResult>
```

```
<Result Date="2004-03-15">
   <Summary> France 1 - Spain 1
   </Summary>
   <Scorers>
      <Scorer Goals="1">
         <Name> Zidane </Name>
         <Country> France </Country>
      </Scorer>
      <Scorer Goals="1">
         <Name> Raul </Name>
         <Country> Spain </Country>
      </Scorer>
   </Scorers>
</Result>
```

```
<Encyclopedia>
  <Football>
    <Player><Name> Zidane </Name>
          <Biography>...</Biography>
    </Player>
    ...
  </Football>
  ...
</Encyclopedia>
```

Concepts : match, joueur, date, biographie, …

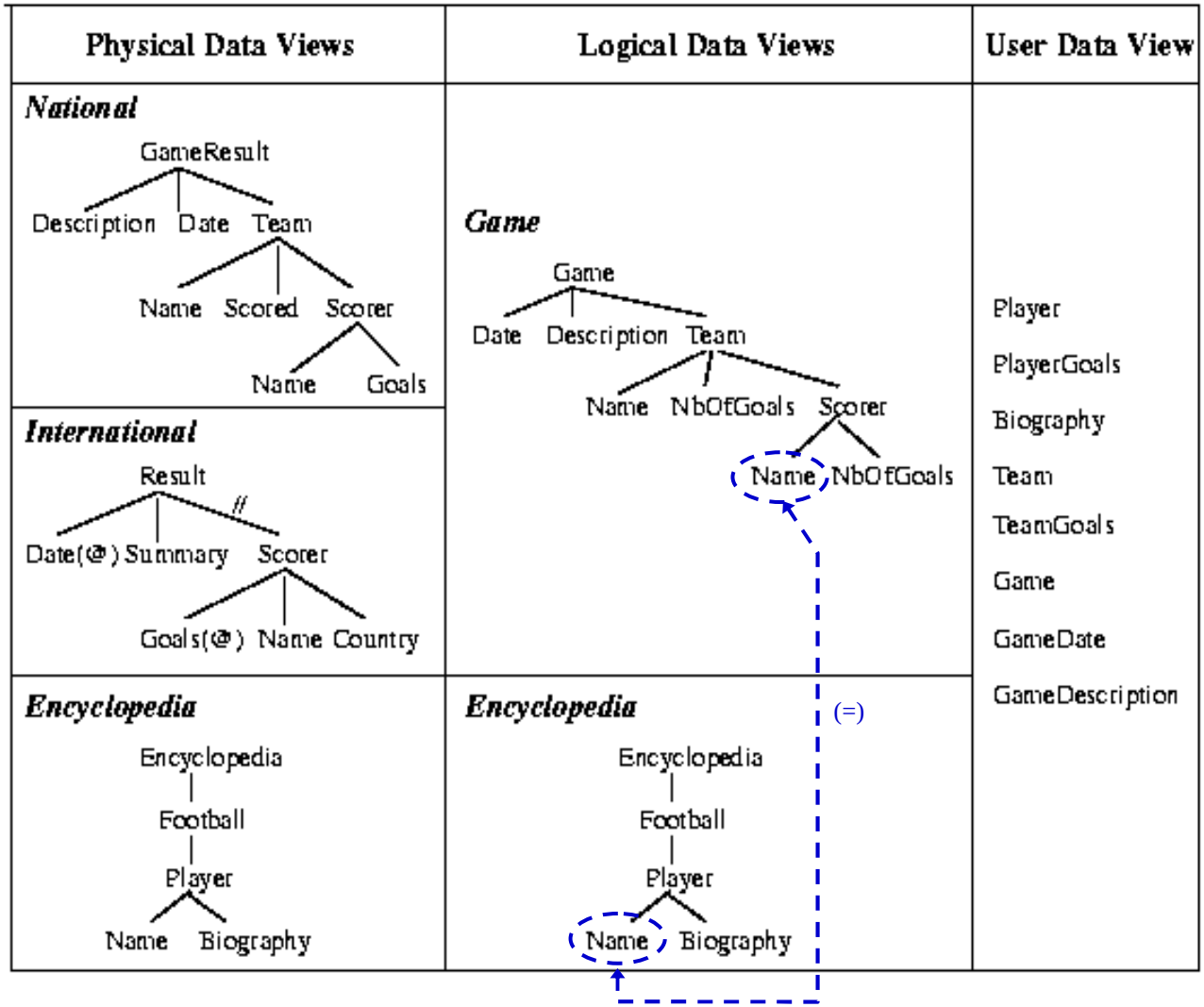
Exemple de requête :

Q : "Les biographies des buteurs des matches du 2004-09-08"

```
union(For $doc1 in collection(NationalURI),
      $var1 in $doc1/GameResult,
      $doc2 in collection(EncyclopediaURI),
      $var2 in $doc2/Encyclopedia/Football/Player,
      $var3 in $var2/Biography
   Where $var1/Date = xs:date('2004-09-08') and
      $var1/Team/Scorer/PlayerName = $var2/Name
   Return string($var3),

   For $doc1 in collection(InternationalURI),
      $var1 in $doc1/Result,
      $doc2 in collection(EncyclopediaURI),
      $var2 in $doc2/Encyclopedia/Football/Player,
      $var3 in $var2/Biography
   Where $var1/@Date = xs:date('2004-09-08') and
      $var1//Player/Name = $var2/Name
   Return string($var3) )
```

Exemple de vue XyView

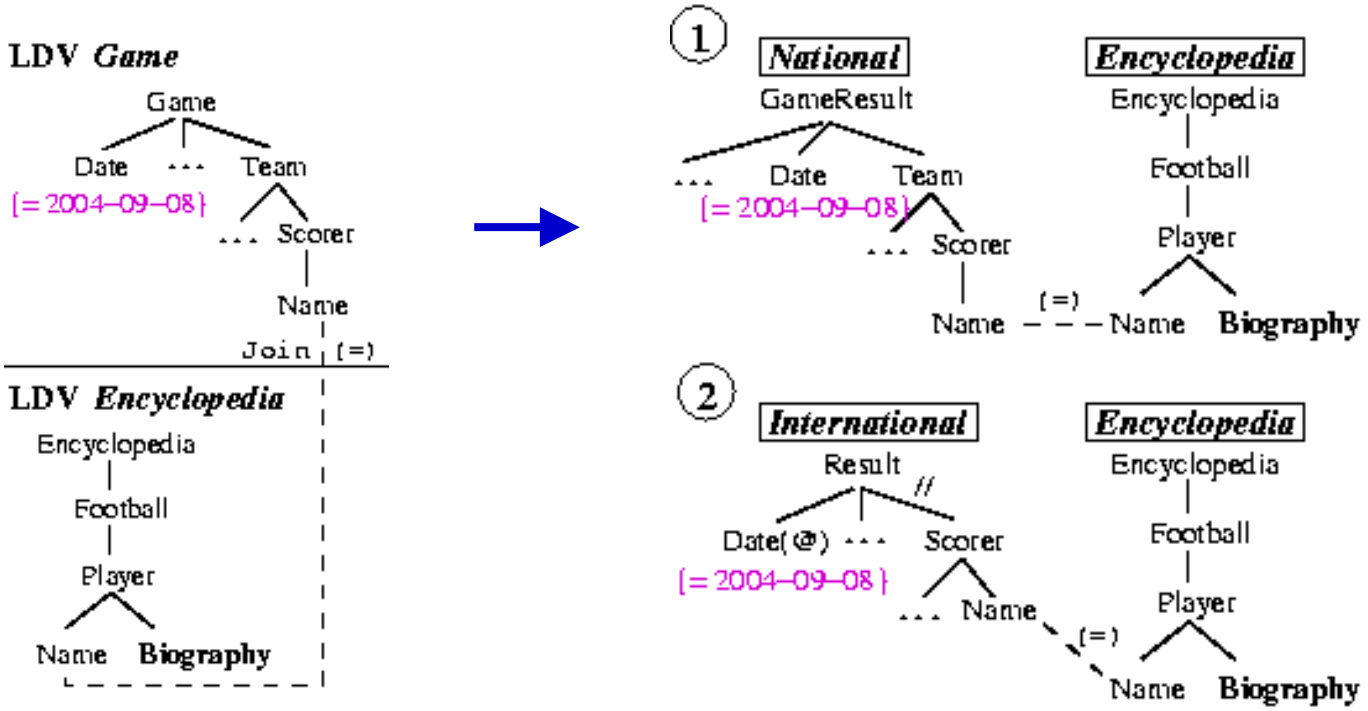


Traduction de requêtes

- Requêtes simples sélection-projection (formulaire)
  - Traduction « global as view » entre les niveaux successifs

Exemple: "Les biographies des buteurs des matches du 2004-09-08"

```
Select Biography
Where  GameDate = 2004-09-08
```



Sources utilisées pour ces transparents

- G. Gardarin, *Médiation de données*, cours Université de Versailles
- A. Doucet, *Intégration de données hétérogènes et réparties*, cours Université Pierre et Marie Curie
- A. Halevy, *Data Integration*, Cours invité à l'Université d'Aalborg