

Intégration de données

Médiation

Contexte général

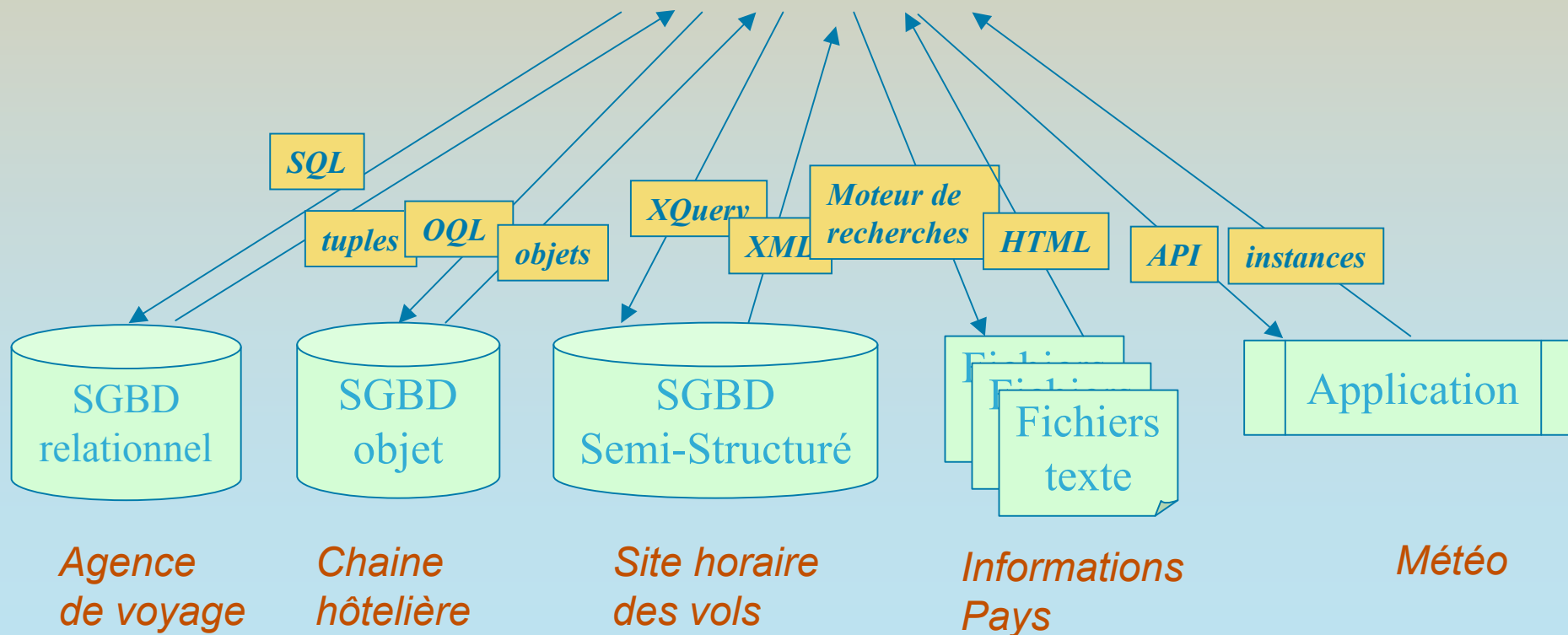
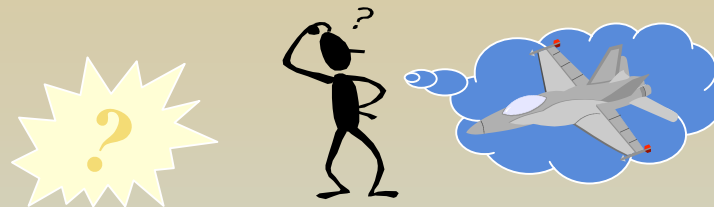
- Sources d'information **nombreuses et diverses**
(SGBD-R ou O ou OR, Pages Web, Données semi-structurées, *etc.*)
- Mode de consultation différents
 - différentes façon d'interroger, *c.a.d* formuler une requête
 - différentes manières pour la sources de répondre, *c.a.d* présenter un résultat
 - exemple :
 - pages web avec URL
 - SGBD-R avec requête SQL
- Interaction avec la source par des méthodes d'accès.
 - Différents protocoles de communication (ODBC, JDBC, IIOP)
 - Différentes interfaces (interface de programmation, interface graphique, *etc.*)

Intégration - Définitions

- Combiner des données de différentes bases
 - collection de données (wrapping)
 - combiner données et générer de nouvelles vues sur les données (mediation)
- Problème : hétérogénéité
 - accès, représentation, content

Exemple

Chercher où passer les vacances



Motivations

- Objectif fondamentaux
 - l'intégration intelligente des informations
 - l'exploitation des sources existantes
- Il faut traiter de :
 - l'hétérogénéité des sources
 - la distribution des sources

Hétérogénéité

- Indépendante de la distribution physique des données à travers un réseau
(on peut avoir un système distribué et homogène)
- Un système est **homogène** si le logiciel qui manipule les données est identique pour toutes les sources et si toutes les données ont le même format (ou modèle)
- Un système **hétérogène** est un système qui n'est pas homogène.
 - Hétérogénéité des schémas
 - Hétérogénéité des données

Hétérogénéité des données

- Modèle « logique »
 - Typages
(ex: adresse a pour type varchar2(64) sous Oracle, string sous PostgreSQL)
 - Structures
(ex: dans une source, personne a pour attributs nom, prenom et age, dans une autre nom, adresse et no_secu)
 - même nom pour désigner des entités différentes
 - noms différents pour désigner la même entité
- Modèle physique
 - Langage de requêtes (ex: SQL, OQL, CGI-BIN)
 - Restitution du résultat (ex: tuples, page Web)

Hétérogénéité des modèles

Source 1: SGBDR

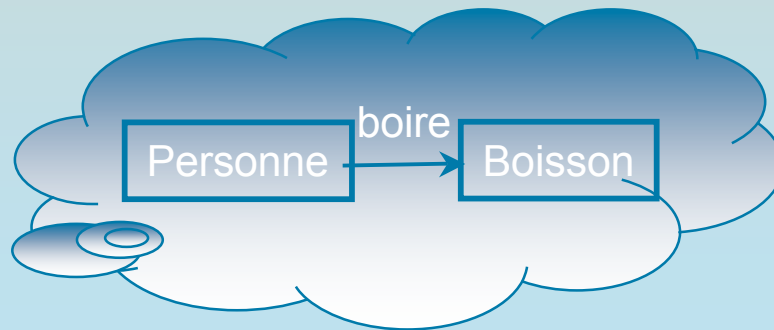
Buveurs	Nom	DateN	Pays	Type
Vins	NV	Cru	Mill	Degre

Source 2: Repository XML

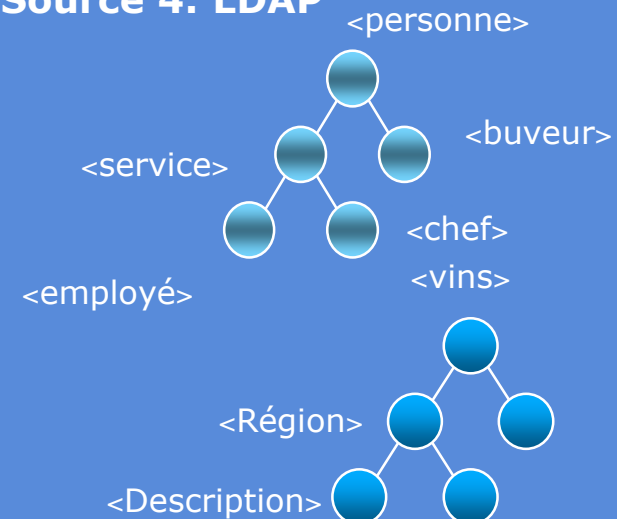
```

<!ELEMENT Vin (Cru, Degre, Description+)>
<!ATTLIST Vin nv CDATA #IMPLIED>
<!ELEMENT Buveur (Nom, Place, Date, Type)>
<!ATTLIST Buveur nb CDATA #IMPLIED>
<!ELEMENT Catalogue (Vin, Offre, Publicité?)+> ...
    
```

Source 3: WEB

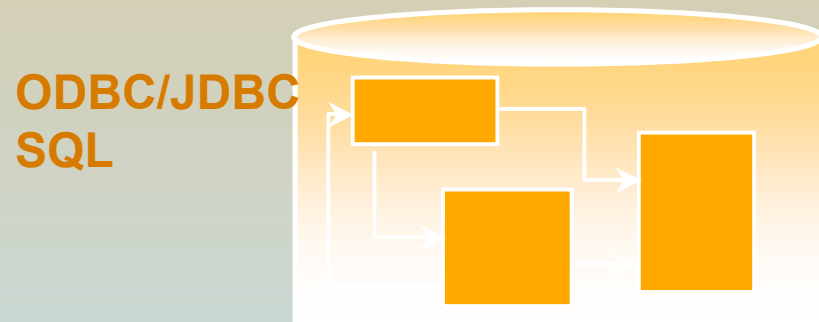


Source 4: LDAP

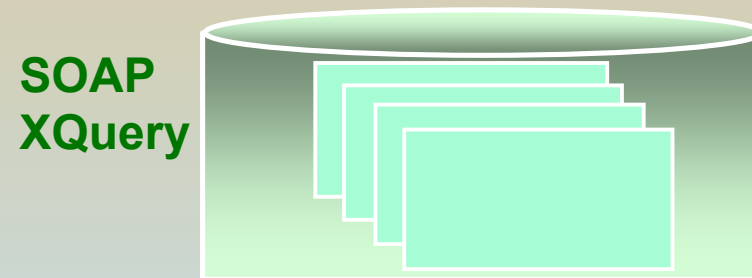


Hétérogénéité des langages

Source 1: RDBMS



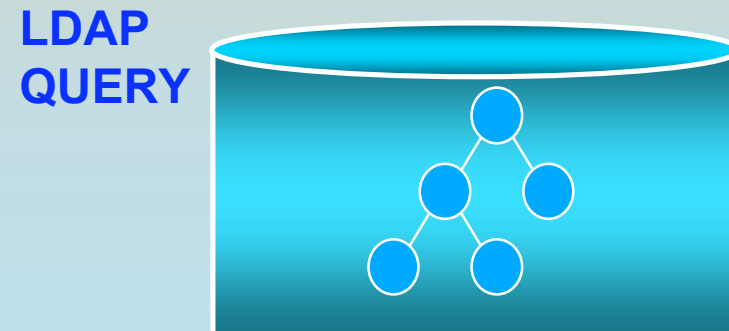
Source 2: XML Repository



Source 3: WEB



Source 4: LDAP



Distribution des données

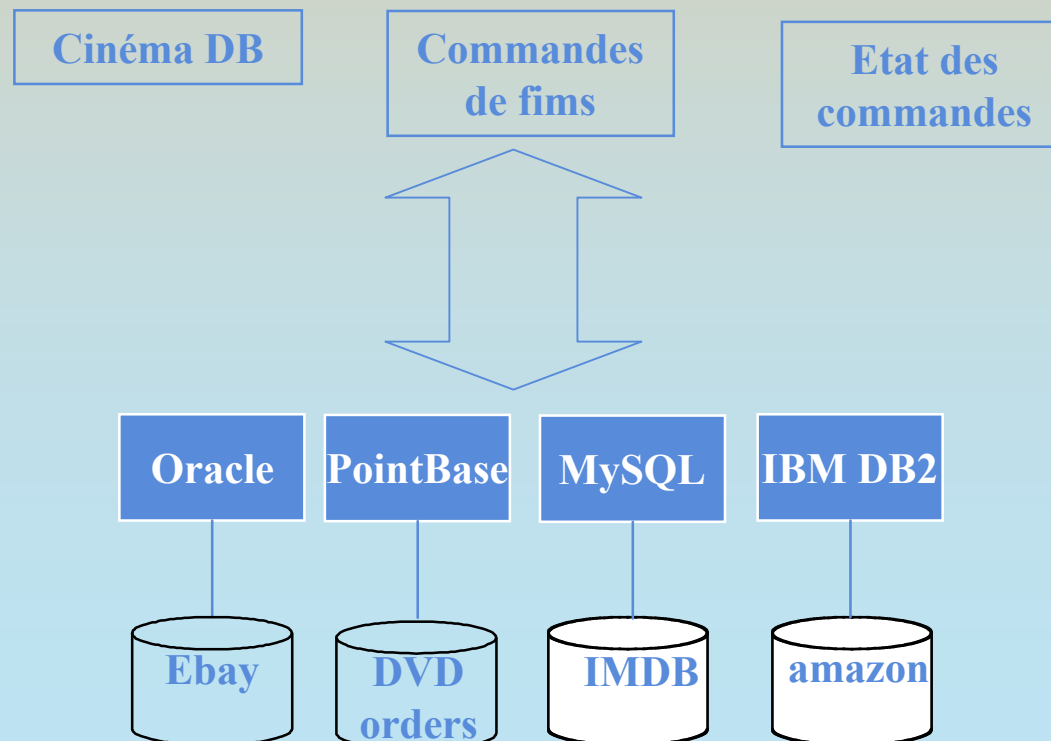
- Localiser quelle source fournira la donnée demandée
- Sources de puissance différentes (temps d'exécution)
- Sources de puissance d'interrogation différentes
- Sources indisponibles temporairement

Systeme interoperable

- Propriétés fondamentales nécessaires à tout système interoperable [Sheth *et* Larson 1990]:
 - **distribution** : les données gérées par le système proviennent de plusieurs sources.
Chaque source met une partie de ses données à disposition des autres participants
 - **hétérogénéité** : chaque source choisie et conçue indépendamment des autres (matériel, système d'exploitation, communication, performance, langages, schémas)
 - **autonomie** : une source participant à un système interoperable doit fonctionner comme avant sa participation.

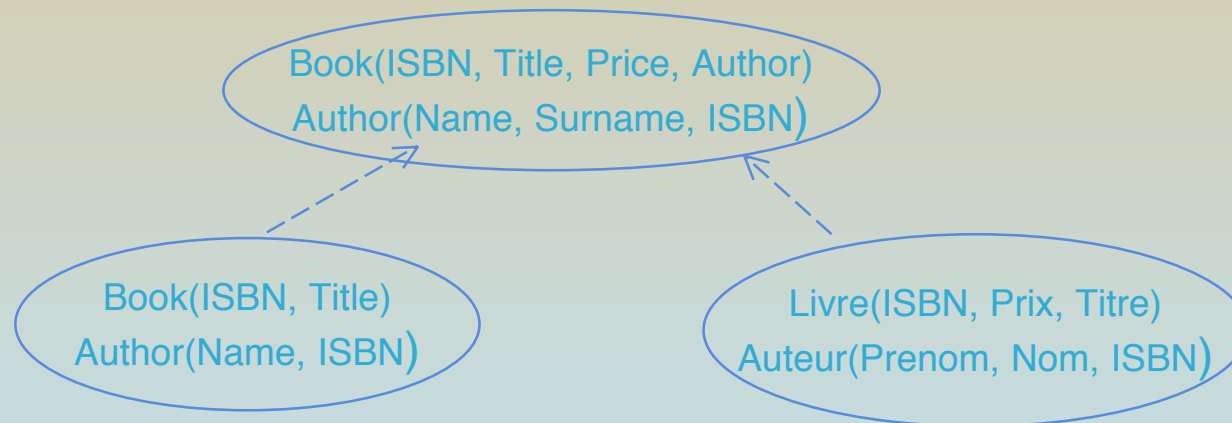
Intégration

- **BUT :**
 - fournir un accès uniforme à de multiples hétérogènes sources d'information
 - Plus que de l'échange simple (e.g., ASCII, EDI, XML)
 - Vieux problème, difficultés, solutions partielles



Intégration

Approche ancienne (1) Intégration manuelle, globale



- Fusionner manuellement diverses bases en une nouvelle base globale
- *Perte de l'autonomie locale*
- *Solution statique*
- *Ne permet pas de passer à l'échelle*

Intégration

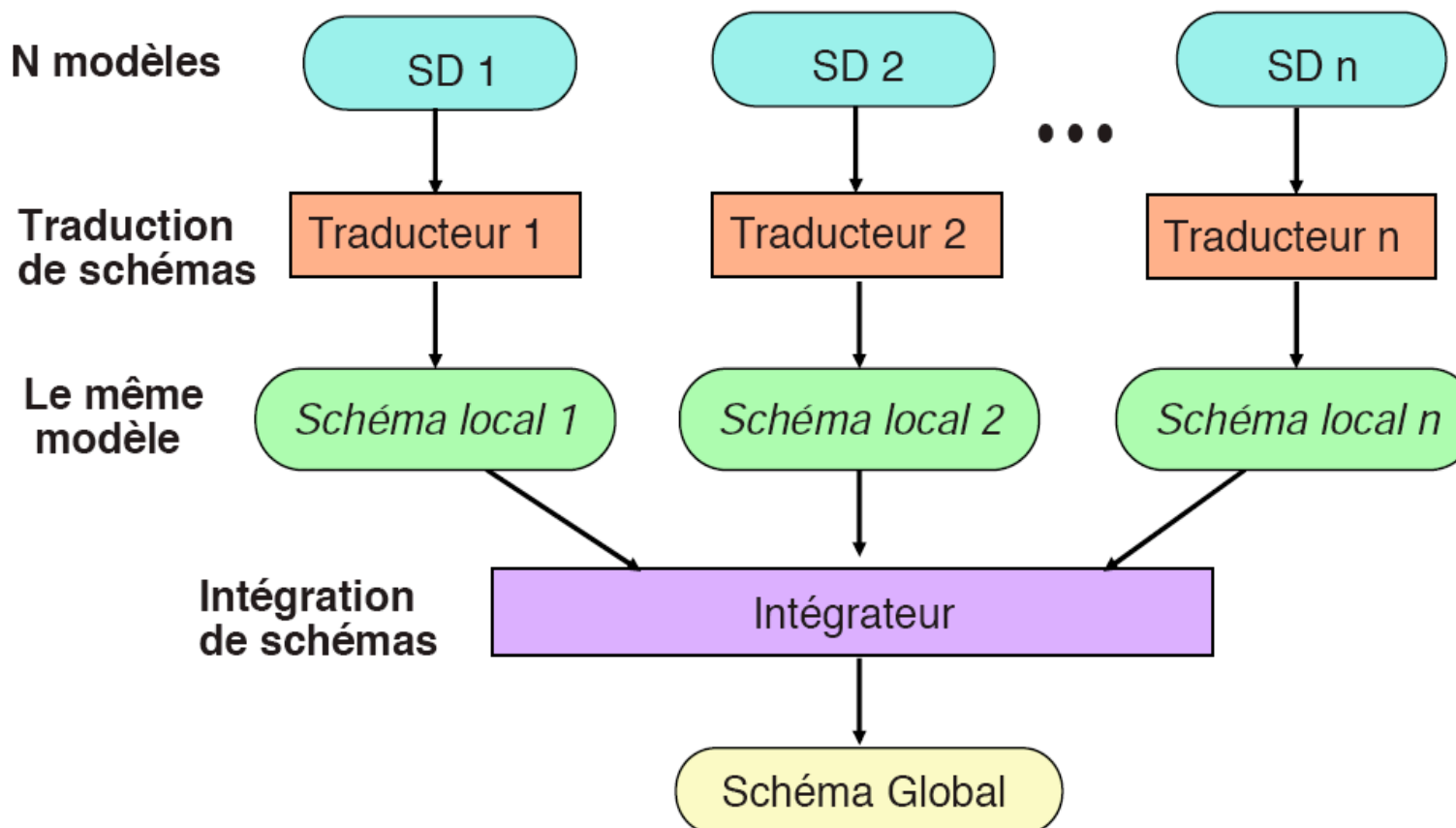
Approche ancienne (2) Langage multibase

- Pas de schéma intégré
- Langage (e.g., MSQL) utilisé pour intégrer les sources d'information
- Exemple :

```
Use S1, S2
Select Titre
From S1.Book, S2.Livre
Where S1.Book.ISBN = S2.Livre.ISBN
```

- Pas de transparence (on doit connaître *toutes les bases!*)
- A la Charge des (experts) utilisateurs
- Requêtes Globales soumises aux changements locaux

Intégration de bases de données



Modèles d'intégration

- Modèle relationnel
 - structures de données simples et régulières
- Modèle objet
 - structures de données complexes et régulières
- Modèle semi-structuré (XML)
 - structures de données complexes et irrégulières
 - pas de schéma obligatoire

Intégration de schémas

- 1. pré-intégration
 - identification des éléments reliés et établissement des règles de conversion (e.g. 1 pouce = 2,54 cm)
- 2. comparaison
 - identification des conflits de noms (synonymes et homonymes) et des conflits structurels (types, clés)
- 3. mise en conformité
 - résolution des conflits de noms et des conflits structurels (changements de types, de clés)
- 4. fusion et restructuration
 - fusion des schémas intermédiaires pour créer le schéma intégré

Intégration en relationnel

Emp = Emp@Site1 U Emp@Site2

prenom	nom	ville	tel.
null	P. Dupont	Paris	0140...
Anne	Martin	Nantes	null
null	A. Martin	Nantes	0235...
Jean	Smith	Lille	null

Emp@Site1

nom	ville	tel.
P. Dupont	Paris	0140...
A. Martin	Nantes	0235...

Emp@Site2

prenom	nomF	ville
Anne	Martin	Nantes
Jean	Smith	Lille

- Problèmes: renommage et introduction de valeurs nulles

Interrogation en relationnel

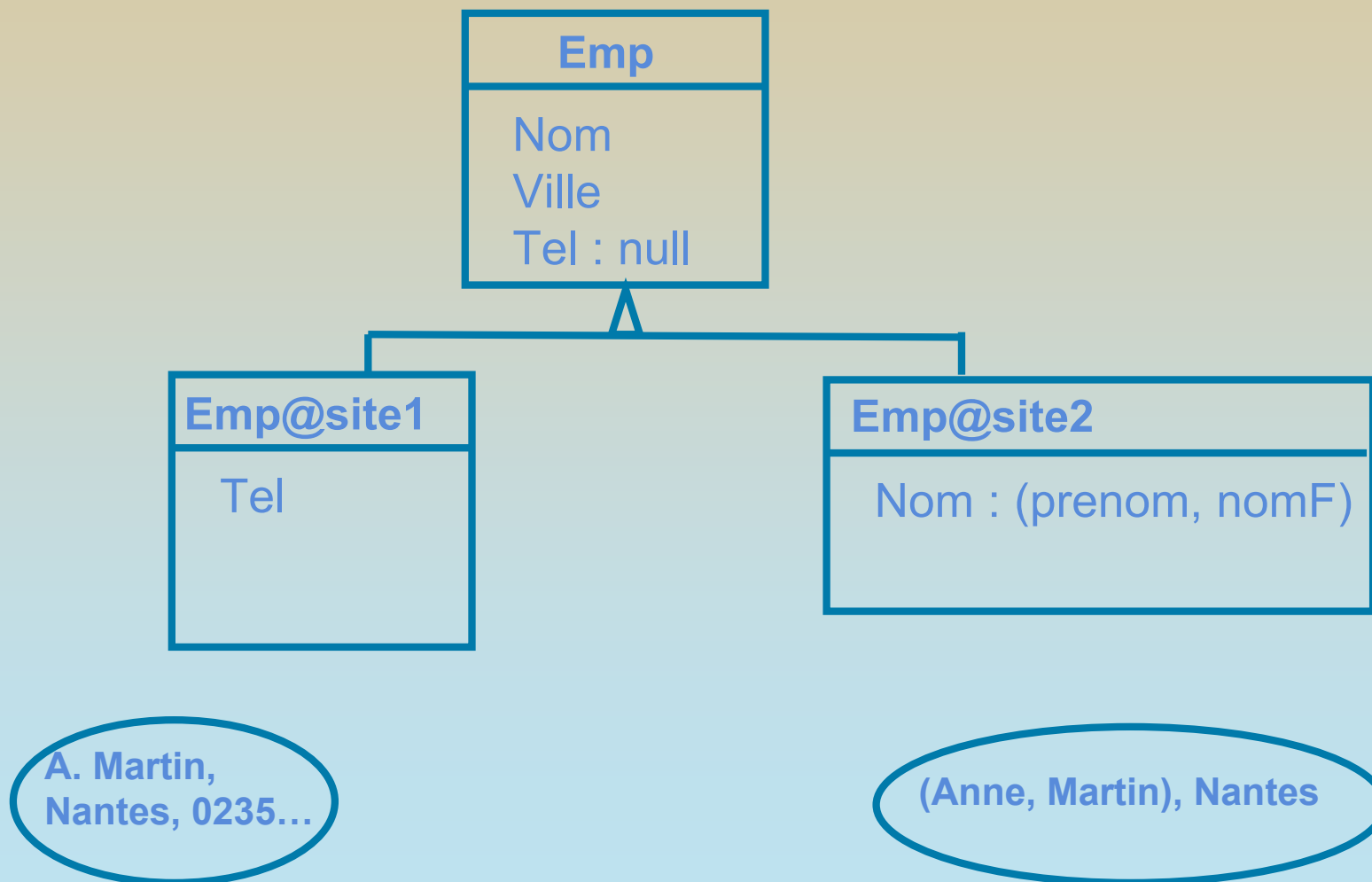
Select prenom, nom, tel
From EMP
Where ville=« Nantes »

prenom	nom	ville	tel.
null	P. Dupont	Paris	0140...
Anne	Martin	Nantes	null
null	A. Martin	Nantes	0235...
Jean	Smith	Lille	null



prenom	nom	tel.
Anne	Martin	null
null	A. Martin	0235...

Intégration en objet



Interrogation en objet

```
Select nom, tel  
From EMP  
Where ville=« Nantes »
```

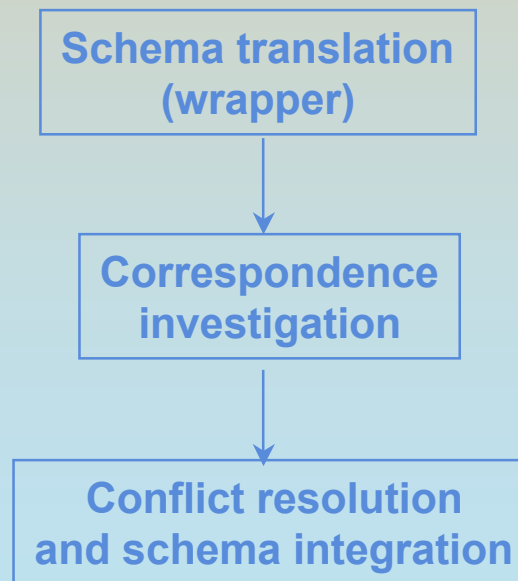


A. Martin, 0235...

(Anne, Martin), null

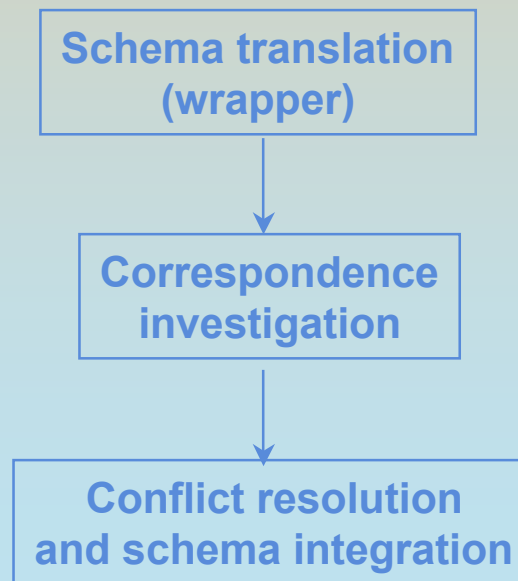
tégration de schémas

- Méthodologie Standard

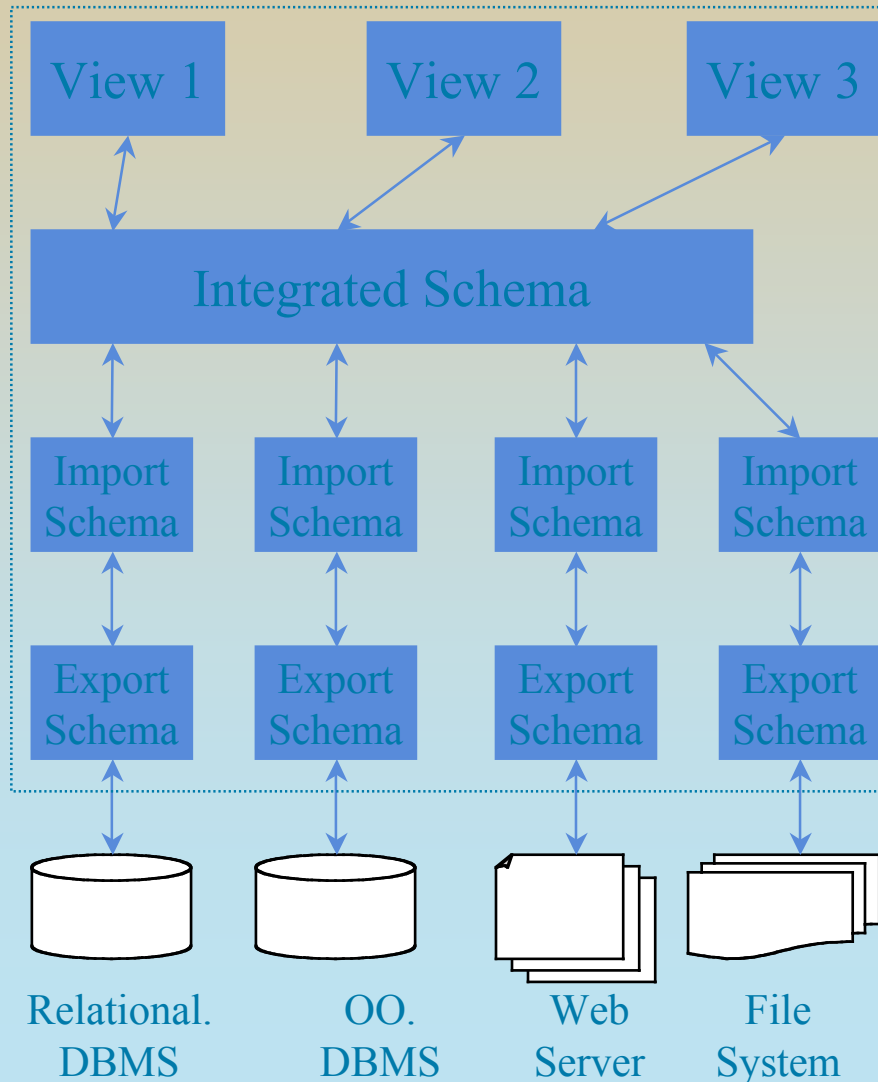


Intégration de schémas

- Méthodologie Standard

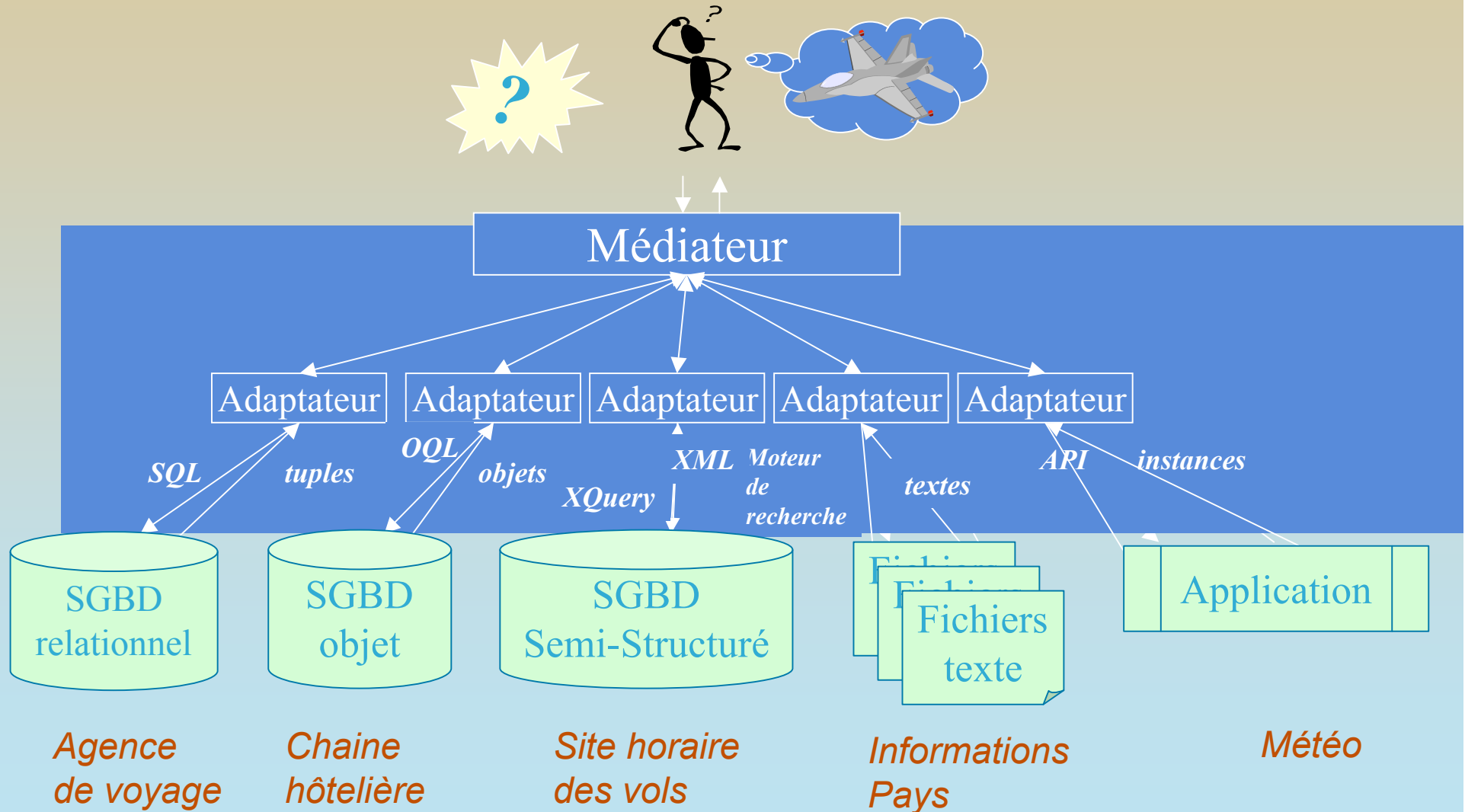


Bases fédérées



- DBMS intégrés avec un schéma intégré
- Architecture 5 niveaux
- Indépendance des données

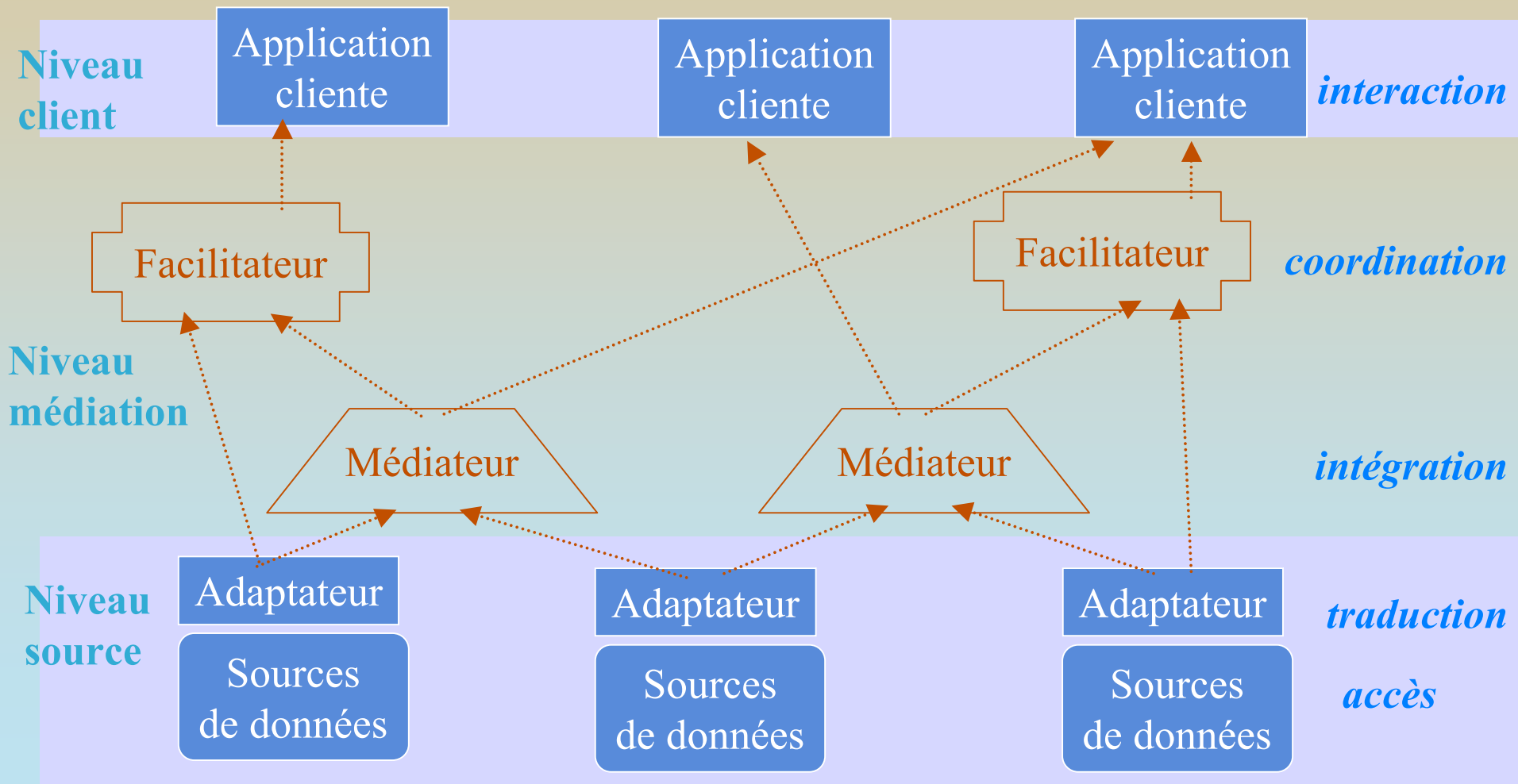
Architecture de médiation



Avantages des architectures de médiation

- **accès intégré** par API et portail Web
- **transparence** à la localisation des données pour les applications
- **disponibilité** accrue des données en cas de pannes des serveurs
- support de l'**hétérogénéité** des sources

Architecture de médiation DARPA I3



Sources de données

- Une source de données peut être décrite par :
 - **localisation**
 - référence du site (URL, IP:Port, annuaire LDAP)
 - protocole de communication (TCP/IP, IPX, AppleTalk)
 - moyen d'accès (ODBC, JDBC, API)
 - support (pages Web, SGBD)
 - **type de données** qu'elle gère
(structuré (SGBD-R, SGBD-O), semi-structuré (XML, OEM), non-structuré (images, multimédia, textes))
 - **possibilité d'interrogation**
(SQL, OQL, propriétaire, moteur de recherche web)
 - **format des résultats**
(XML, HTML, ResultSet (tuples), OEM, textes)

Communication médiateur/adaptateur

- Pour faciliter le travail d'intégration, on définit
 - un langage commun dans lequel le médiateur interrogera les adaptateurs
 - un format de résultat commun dans lequel les adaptateurs répondront au médiateur.
- Ce langage et format de résultat communs peuvent être propriétaires ou standardisés

Adaptateur (Wrapper)

- L'adaptateur (Wrapper) s'occupe de l'hétérogénéité des sources. C'est un “traducteur”.
 - Traduction du langage de requête commun en langage de requête natif (propre à la source)
 - Traduction des résultats natifs en résultats au format commun

Médiateur

- Le **médiateur** s'occupe de la distribution des sources
 - Localisation des sources
 - Décomposition des requêtes en requête adaptée pour chacune des sources
 - Envoi des requêtes aux sources
 - Recomposition des différents résultats provenant de chacune des sources

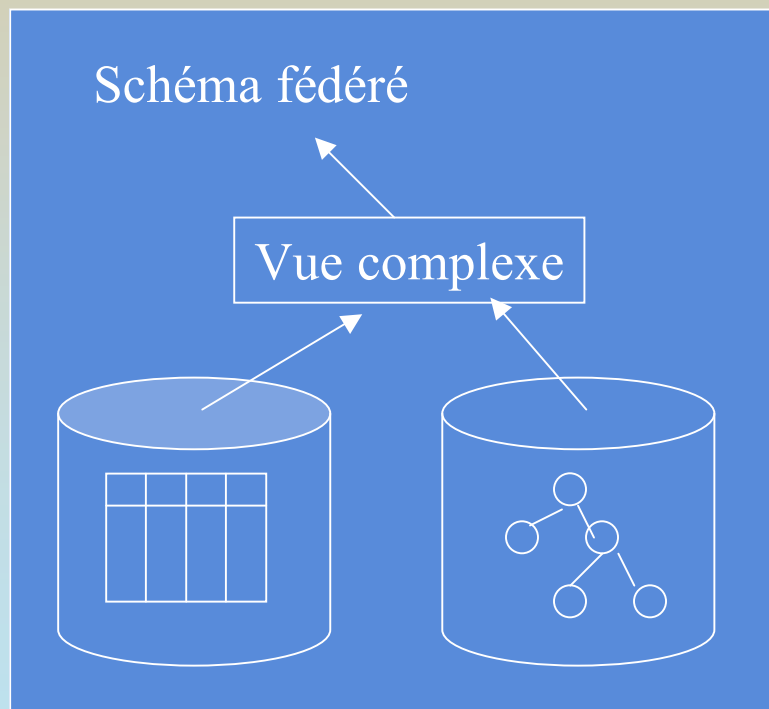
Intégration des schémas

- Comporte différents aspects
 - Comparaison de schéma
 - Unification de schéma
 - Fusion de schéma
- Difficultés
 - Conflit de niveau d'abstraction
 - Conflit de définition de classes
 - Conflit de divergence schématique

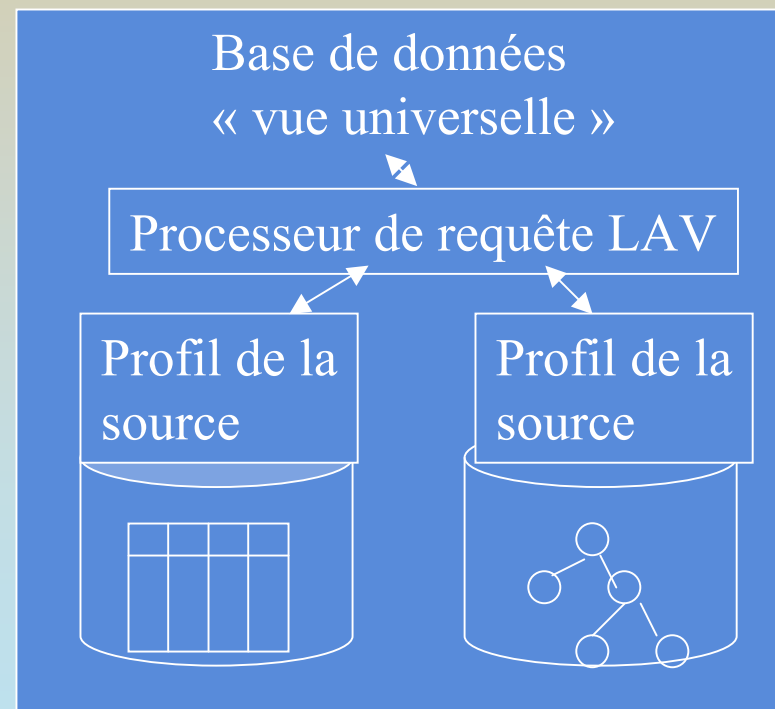
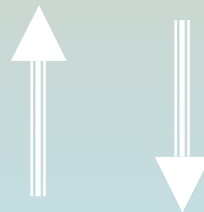
Architecture GAV et LAV

- GAV : Global as View
 - Schéma global défini comme une vue intégrante sur schémas locaux
 - Approche ascendante depuis les sources vers le médiateur
- LAV : Local As View
 - Chaque source locale est définie comme une vue locale du schéma global
 - Approche descendante depuis le médiateur vers les sources

Décomposition versus Recomposition



GAV



LAV

Traitement des requêtes

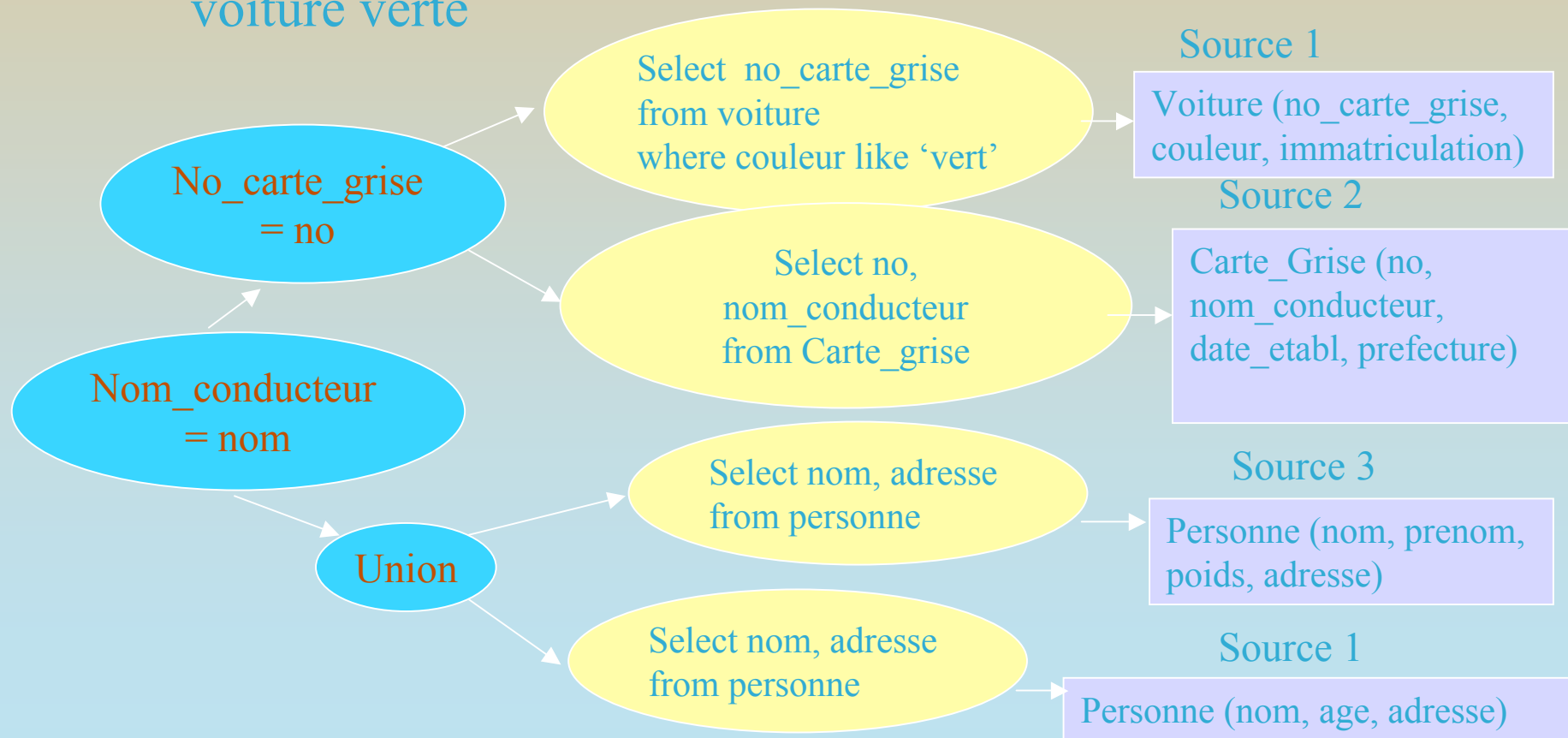
- Analyse syntaxique et sémantique
- Décomposition des requêtes
- Exécution des requêtes sur les sources
 - transformation de la requête en langage commun vers le langage de la source
 - transformation du résultat au format de la source vers le format commun
- Recomposition des résultats
 - combinaison des résultats locaux
 - requêtes de recombinaison sur système global ou un des sites composants.

Plans d'exécution

- Un plan d'exécution décrit la méthode d'exécution d'une requête. Il est souvent représenté par un arbre algébrique.
- Un arbre algébrique est un arbre où les nœuds sont des opérateurs algébriques et les feuilles les sources de données.
- Il peut exister plusieurs voire une infinité de façon d'exécuter une requête (toutes représentée par des plans d'exécution équivalents). L'ensemble des plans d'exécution permettant de résoudre une requête est appelé espace de recherche.

Décomposition des requêtes

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



Puissance d'interrogation des sources

- Toutes les sources n'ont pas les mêmes possibilités d'interrogation
 - SGBD : possibilité de requêtes souvent complexes
 - Moteur de recherche : par mots-clefs
 - Fichiers : via champ indexé
- Le médiateur ou l'adaptateur de la source doit pallier les déficiences de la source
 - si adaptateur : implémentation complexe de chaque adaptateur, intégration simple au niveau médiateur
 - si médiateur : implémentation simple des adaptateurs, intégration complexe au niveau médiateur. Nécessite une communication des capacités de la source au médiateur.

Optimisation des requêtes

- Stratégies classiques de remontée de projection, restriction, etc.
- Ordonnancement des jointures
- Stratégie sur les jointures inter-sites
 - par interrogation multiples d'une source avec les résultats du premier
 - par boucles imbriqués
 - par tri fusion

Optimisation statique de requêtes hétérogènes

- Ajout de vues transitoires
- Décomposition d'une requête
- Simplification d'une requête
- Prise en compte des capacités réduites des sources
- Parallélisation d'une requête
- →Élaboration d'un plan optimisé

Optimisation dynamique de requêtes hétérogènes

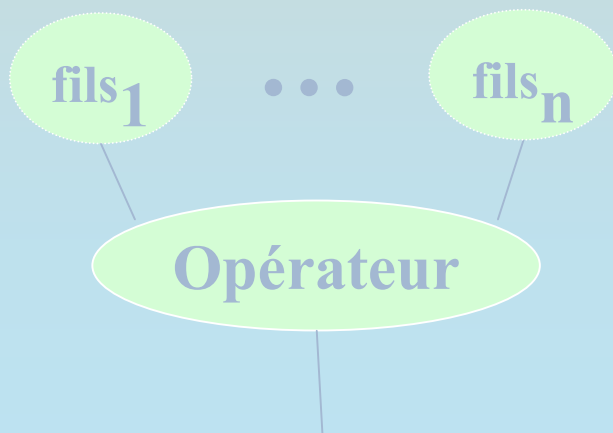
- Reformulation dynamique du plan d'exécution
- Prise en compte de sources indisponibles
- Ordonnancement dynamique des jointures
- Optimisation adaptative

Modèles de coûts

- Un **modèle de coût** permet d'estimer le coût que prendra un plan d'exécution.
- But : choisir parmi tous les plans d'exécution celui de coût minimal pour l'exécution
- S'appuie sur les **statistiques** et des formules de coûts.
- Statistiques :
 - Du système : Système d'exploitation (CPU, E/S), SGBD (taille d'une page, taille cache)
 - Des données : cardinalité d'une collection, sélectivité d'un attribut, etc.

Modèle de coût au niveau médiateur

Coût d'exécution d'une requête
= coût_communication
+ opération_médiateur
+ coûts_sur_les_adaptateurs
+ congestion_du_réseau



Dépend du débit, de la taille des données à transférer

Formule classiques de coût de calcul d'opérateurs en mémoire

$coûts_fils \in [max (coût_fils_i), \Sigma(coût_fils_i)]$
suivant degré de parallélisme

Difficile à gérer : latence et temps d'attente au moment de l'exécution de la requête

Coût sur les adaptateurs

- Les sources sont indépendantes et ne communiquent pas forcément leurs informations de coûts.
- Différentes stratégies permettant d'estimer le coût d'une requête sur une source
 - Estimation analytique
 - Soumission de formules
 - Apprentissage progressif
 - Gestion d'historique

Modèle de coût

Coût d'une architecture de médiation

- **Calibration [PEGASUS]**
 - requêtes types pour calibrer paramètres de la source
 - affinée avec échantillonnage
 - pour données objets [IRO-DB]
- **Historique [HERMES]**
 - s'appuie sur les statistiques des requêtes précédentes
- **Défini par les adaptateurs [GARLIC]**
 - modèle de coût défini séparément pour chaque adaptateur
- **Générique [DISCO]**
 - intégrer modèle de coût des adaptateurs + hiérarchie de coût et coût par défaut pour coût manquant d'un adaptateur

Coût sur données semi-structurées

- **Coût sur modèle semi-structuré dans un entrepôt [LORE]**

Gestion de Cache

- **Cache de pages** : adapté à des SGBD classiques, peu adapté aux autres sources (Web, ou opaques)
- **Cache de tuples** : faisable pour les pages web (proxy). Mais difficile de préciser quels sont les tuples déjà dans le cache.
- **Cache sémantique** : Garder un historique des prédicats de requêtes déjà posées.
 - requête dans le cache local
 - requête complémentaire
 - actualiser le cache

Médiateur existants

- Génération relationnelle (1975-1990)
 - Souvent centré sur un SGBD qui joue le rôle d'un médiateur
 - SDD1, Sirius Delta, R*, Ingres/Star
 - Mermaid, Multibase, MSQ
- Génération relationnelle étendue (1990-2000)
 - Fédère des BD hétérogènes autour de SQL3
 - Objet : OLE-DB, pegasus, IRO-DB
 - XML : Medience Server, Information Integrator (IBM)
- Génération XML XQuery (2000- ...)
 - OLE-DB.NET (Microsoft), Nimble, Xquark Fusion,
 - Liquid Data (BEA), Enosys Software

Conclusion

- Internet s'étend
 - Sources d'information de plus en plus nombreuses
 - Informations de plus en plus hétérogènes
- Médiation de plus en plus nécessaire
 - base de connaissances
 - portails d'information
 - moteur de recherche spécifiques