CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Bases de données réparties et fédérées

Février 2002

René J. Chevance

Contenu

- Définitions
- Exemple de BD répartie
- Répartition des données
- Répartition - Fédération
- Fédération de BD
  - Quelques cas de conflits
- Traduction des schémas
- Architecture de référence
- Accès aux BD multiples
  - API commune
  - FAP commun avec passerelles
  - FAP commun supporté par les SGBD
- Niveaux de transparence à la localisation
  - Client/Multibases :
    - RDA, DRDA, SQL-CLI, ODBC
  - Vues réparties
  - SGBD répartis
- Quelques problèmes des BD réparties et fédérées
- Partitionnement et placement des données
  - Quelques règles pour le partitionnement
  - Expédition de données et Expédition de Fonction
  - Recherche du partitionnement idéal
- Optimisation des requêtes réparties
- Réplication dans les BD
- Un aperçu sur les SGBD du commerce
  - Un peu d'histoire : Ingres/STAR
  - IBM DB2, Informix, Oracle, Sybase
- Évaluation des SGBD répartis
- Bibliographie

Définitions

- **SGBD réparti ou SGBD distribué (Distributed DBMS)**
  - **Système gérant une collection de BD logiquement reliées, réparties sur différents sites en fournissant un moyen d'accès rendant la distribution transparente**
- **Base de donnée fédérée - *a priori* hétérogène (Federated BD)**
  - **Plusieurs BD hétérogènes capables d'interopérer *via* une vue commune (modèle commun)**
- **Multibase**
  - **Plusieurs BD (hétérogènes ou non) capables d'interopérer sans une vue commune (absence de modèle commun)**

Pourquoi des BD réparties ou fédérées?

- **Réparties :**
  - **Amélioration des performances (placer les traitements à l'endroit où se trouvent les données)**
  - **Disponibilité en raison de l'existence de plusieurs copies**
  - **Maintient d'une vision unique de la base de données malgré la répartition**
- **Fédération :**
  - **Donner aux utilisateurs une vue unique des données implémentées sur plusieurs systèmes *a priori* hétérogènes (plates-formes et SGBD)**
  - **Cas typique rencontré lors de la concentration d'entreprises : faire cohabiter les différents systèmes tout en leur permettant d'interopérer**

Exemple de BD répartie

Schéma global entité - relation

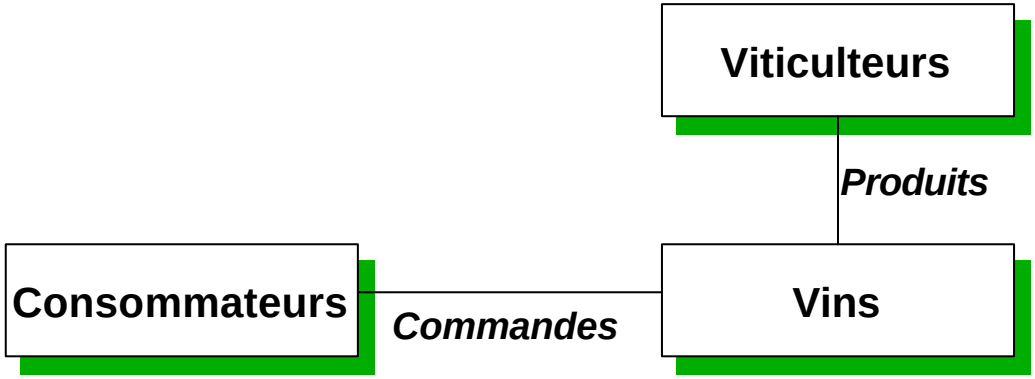
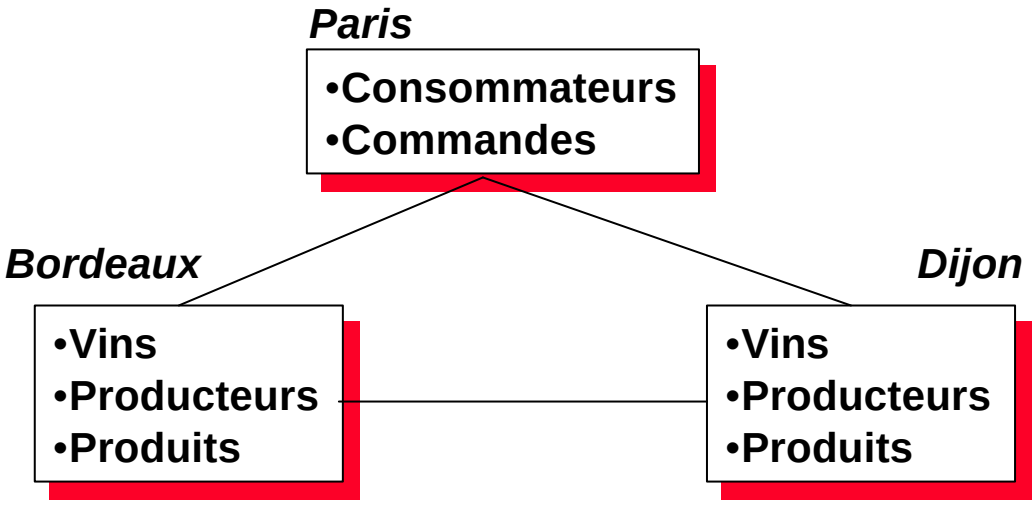
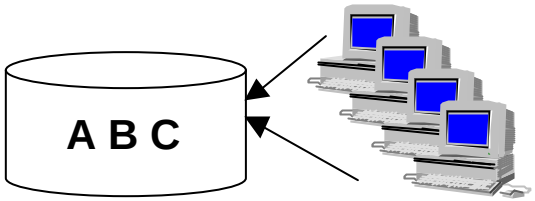


Schéma des données réparties



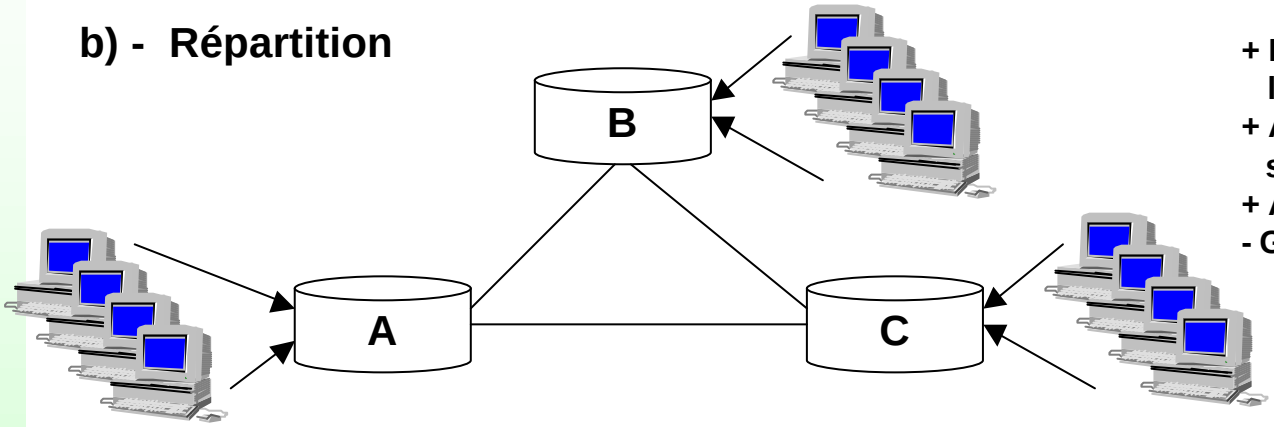
Répartition des données

a) - Cas centralisé



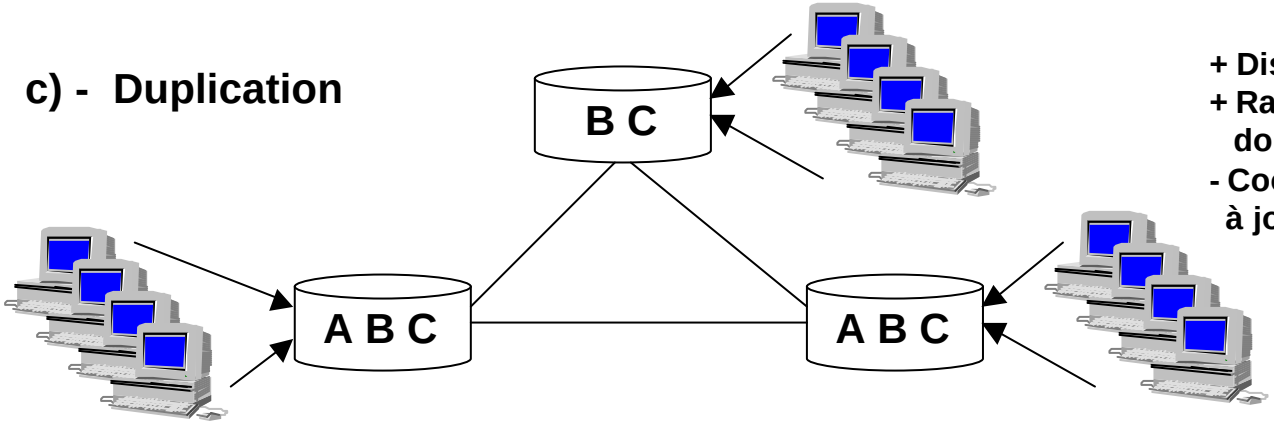
+ Égalité des accès
+ Facilité de gestion
- Contention sur la BD

b) - Répartition



+ Rapidité d'accès au données locales
+ Autonomie locale de chaque site
+ Accès possible aux autres sites
- Gestion globale de la BD

c) - Duplication

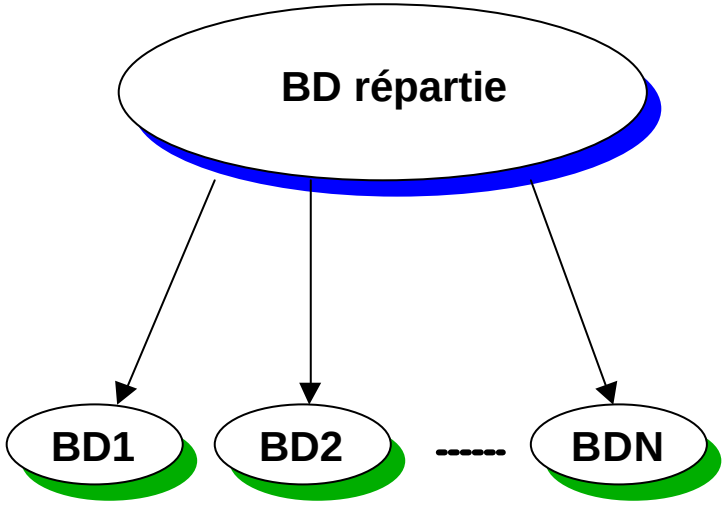


+ Disponibilité des données
+ Rapidité d 'accès aux données locales
- Coordination des mises à jour

Répartition - Fédération

Approche descendante

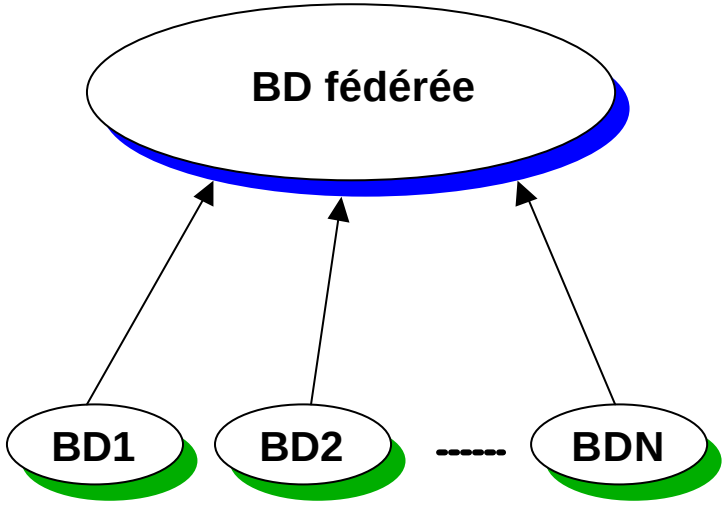
- Conception d'une BD répartie



- Maîtrise de la complexité de la répartition (fragmentation, duplication, placement)
- Définition des schémas locaux à partir du schéma global

Approche ascendante

- Intégration/fédération de BD existantes



- Maîtrise de l'hétérogénéité sémantique (BD) et syntaxique (SGBD, communications,….)
- Maîtrise de l'intégration des schémas locaux pour créer un schéma global

Contenu

- Définitions
- Exemple de BD répartie
- Répartition des données
- Répartition - Fédération
- → Fédération de BD
  - Quelques cas de conflits
- Traduction des schémas
- Architecture de référence
- Accès aux BD multiples
  - API commune
  - FAP commun avec passerelles
  - FAP commun supporté par les SGBD
- Niveaux de transparence à la localisation
  - Client/Multibases :
    - RDA, DRDA, SQL-CLI, ODBC
  - Vues réparties
  - SGBD répartis
- Quelques problèmes des BD réparties et fédérées
- Partitionnement et placement des données
  - Quelques règles pour le partitionnement
  - Expédition de données et Expédition de Fonction
  - Recherche du partitionnement idéal
- Optimisation des requêtes réparties
- Réplication dans les BD
- Un aperçu sur les SGBD du commerce
  - Un peu d'histoire : Ingres/STAR
  - IBM DB2, Informix, Oracle, Sybase
- Évaluation des SGBD répartis
- Bibliographie

Fédération de BD

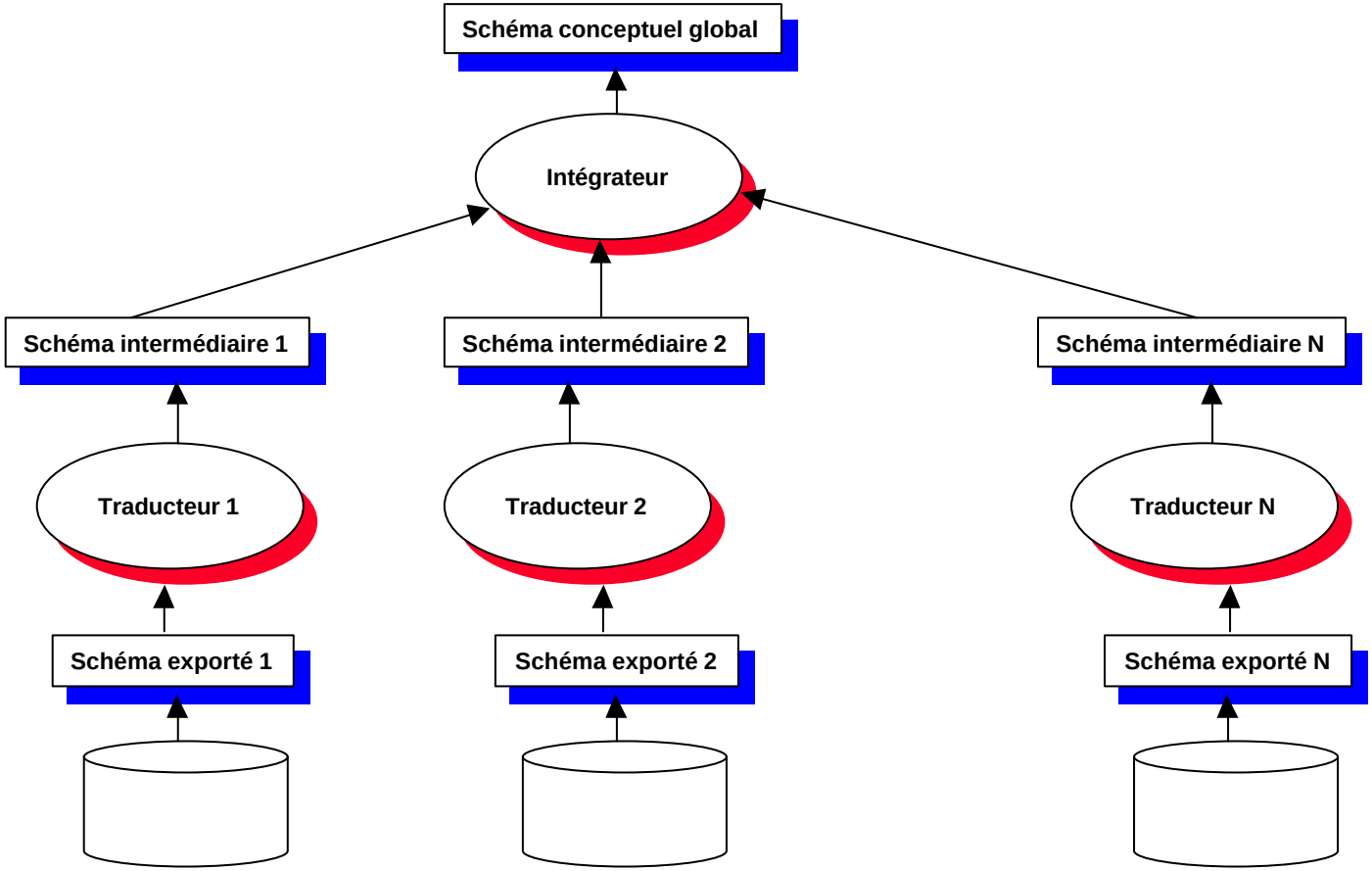
- **Procédure d'intégration :**
  - **Traitement de l'hétérogénéité sémantique**
  - **Traduction des schémas (résolution de l'hétérogénéité syntaxique)**
  - **Intégration des schémas**
- **Hétérogénéité sémantique**
  - **Origine : Résulte des conceptions indépendantes des différentes BD**
  - **Effet : Désaccord sur la signification des données**
  - **Solution : Analyse sémantique comparée des données préalable à la fédération souvent groupée avec la phase de traduction**

Fédération de BD (2)

- **Traduction des schémas (résolution de l'hétérogénéité syntaxique)**
  - **Origine : utilisation de modèles différents dans les BD composantes**
  - **Effet : nécessite des traductions de tous les modèles vers tous les modèles**
  - **Solution : traduction de tous les schémas dans un modèle commun (dit canonique ou pivot)**
  - **Problématique :**
    - **Le modèle canonique doit avoir un pouvoir de modélisation ^{3} à ceux des modèles des BD composantes**
    - **Nécessité de compléter sémantiquement des modèles de BD composantes qui seraient trop pauvres**
  - **Choix du modèle canonique :**
    - **Entité - Association et Relationnel**
    - **Objet**

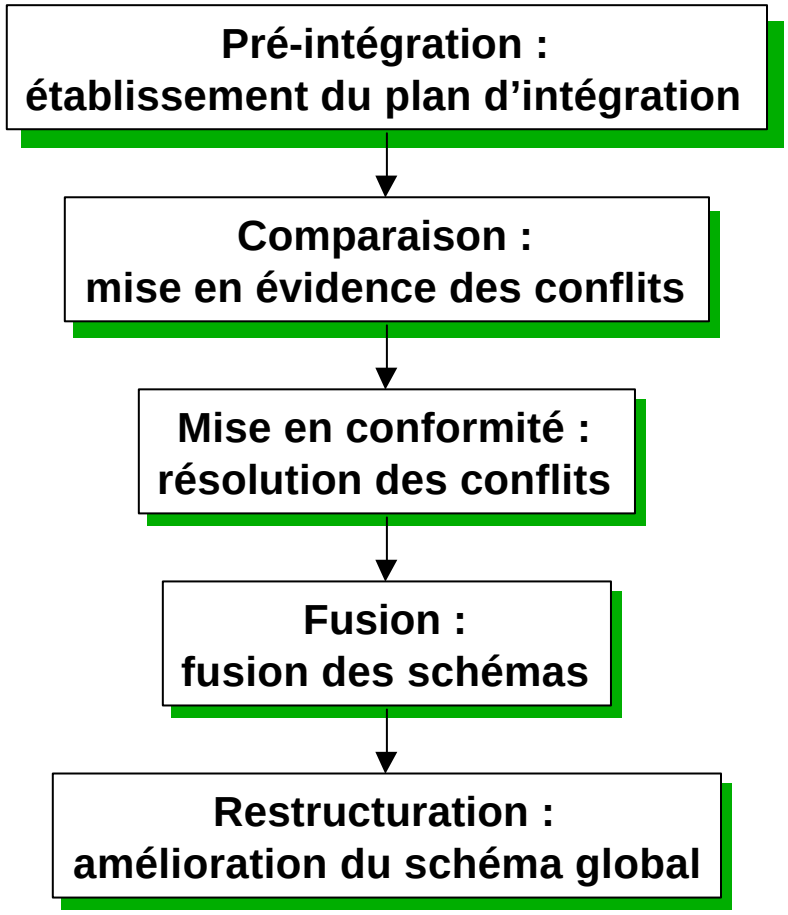
Fédération de BD (3)

- **Intégration des schémas**



- **Procédure :**
  - **Identifier les éléments de base qui sont liés**
  - **Choisir la représentation la plus adéquate pour le schéma global**
  - **Intégrer les éléments des schémas intermédiaires**

Démarche d'intégration



Fédération de BD (5)

- **Démarche d'intégration**
  - **Pré-intégration :**
    - **Mise en évidence des dépendances induites par les schémas**
    - **Définitions des équivalences entre domaines**
    - **Convention de désignation**
  - **Comparaison ou analyse - mise en évidence des conflits :**
    - **de désignation (homonymie, synonymie)**
    - **structurels**
    - **de domaine**
    - **de contraintes**
    - **....**
  - **Mise en conformité : résolution des conflits**
    - **renommage pour les conflits de noms**
    - **étude au cas par cas pour les conflits structurels**
  - **Fusion des schémas - Qualités recherchées :**
    - **complétude (pas de perte d'information)**
    - **minimalité (absence de redondance)**
    - **clarté**
  - **Restructuration - Amélioration du schéma global**
    - **pour l'essentiel recherche de clarté sans remise en cause des qualités recherchées**

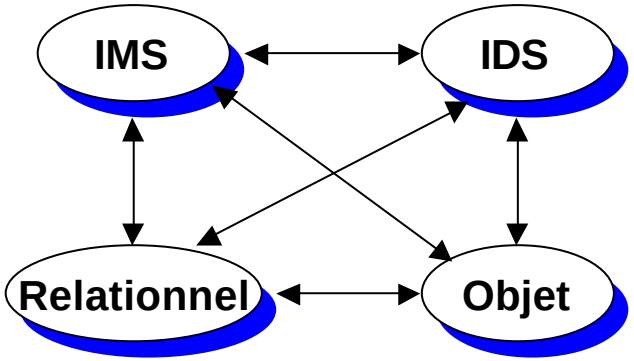
Quelques cas de conflits

- **Conflits d'attributs**
  - **Conflit de nom Þ renommage**
  - **Conflit de type Þ conversion**
- **Attribut sans équivalent dans l'autre relation**
  - **Attribut optionnel Þ valeur nulle**
  - **Attribut indispensable Þ relation auxiliaire**
- **Conflit de relation**
  - **Conflit multi-attribut : un attribut correspond à plusieurs dans l'autre relation (ex. adresse et N°, rue, code, ville) Þ utilisation d'un calcul sur les attributs (ex. extraction)**
  - **Conflit de clé**
    - **pas la même clé Þ changement de clé**
    - **la clé d'une des relations composantes n'est pas une clé générale :**
      - **Þ génération d'une nouvelle clé par ajout d'un élément (ex. nom de commune pas déterminant au niveau national Þ ajout du numéro de département au nom de la commune pour créer la nouvelle clé)**
- **….**

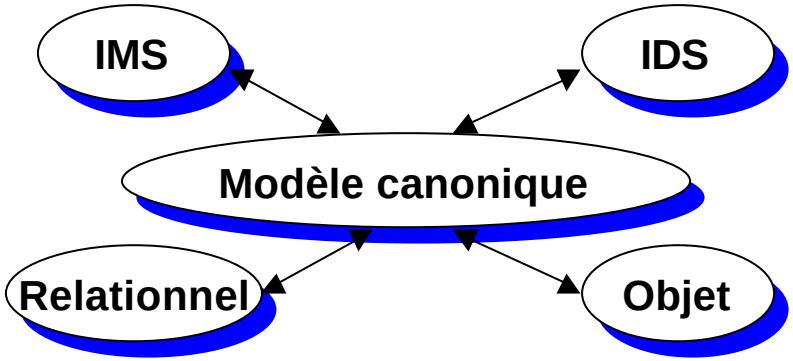
Contenu

- Définitions
- Exemple de BD répartie
- Répartition des données
- Répartition - Fédération
- Fédération de BD
  - Quelques cas de conflits
- ➔ Traduction des schémas
- Architecture de référence
- Accès aux BD multiples
  - API commune
  - FAP commun avec passerelles
  - FAP commun supporté par les SGBD
- Niveaux de transparence à la localisation
  - Client/Multibases :
    - RDA, DRDA, SQL-CLI, ODBC
  - Vues réparties
  - SGBD répartis
- Quelques problèmes des BD réparties et fédérées
- Partitionnement et placement des données
  - Quelques règles pour le partitionnement
  - Expédition de données et Expédition de Fonction
  - Recherche du partitionnement idéal
- Optimisation des requêtes réparties
- Réplication dans les BD
- Un aperçu sur les SGBD du commerce
  - Un peu d'histoire : Ingres/STAR
  - IBM DB2, Informix, Oracle, Sybase
- Évaluation des SGBD répartis
- Bibliographie

Traduction des schémas



N x (N - 1) / 2 traducteurs



N traducteurs

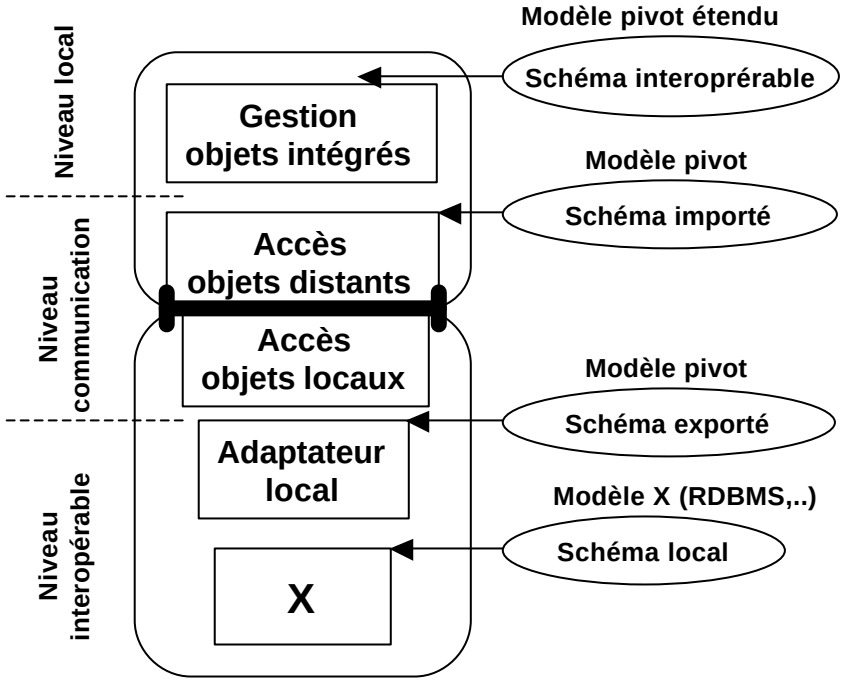
- **Passage par un modèle canonique :**
  - **Chaque site possède un traducteur local/canonique**
  - **Chaque traducteur réalise 3 conversions :**
    - **schéma local ⇘ schéma équivalent en modèle canonique**
    - **données locales ⇘ données équivalentes en modèle canonique**
    - **requêtes en langage du modèle canonique ⇘ requêtes équivalentes en modèle local**

+ Développement d'un seul traducteur par SGBD
+ Simplification de la modélisation
+ Transparence

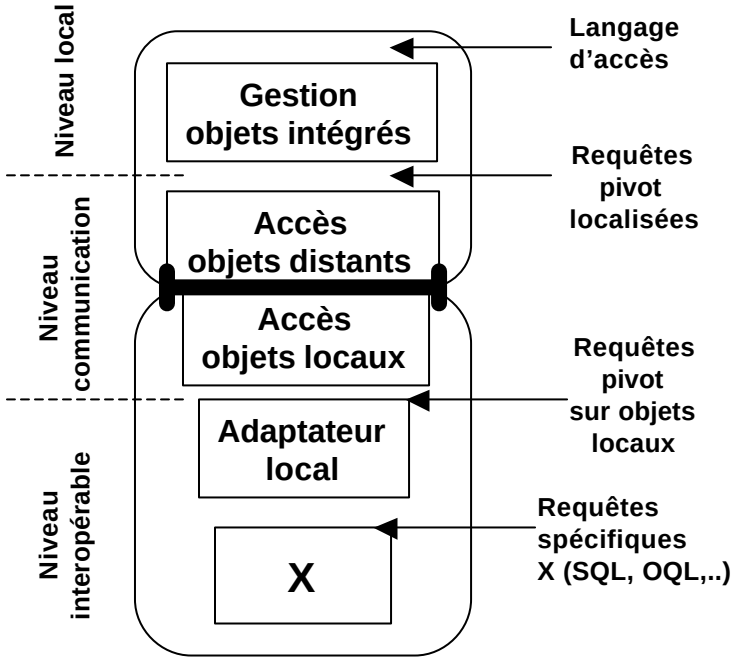
- Difficulté de définir un modèle canonique aussi riche que les modèles locaux
- Temps de réponse accru pour les interrogations locales

Architecture de référence

- **Organisation des schémas**



- **Niveau des langages**

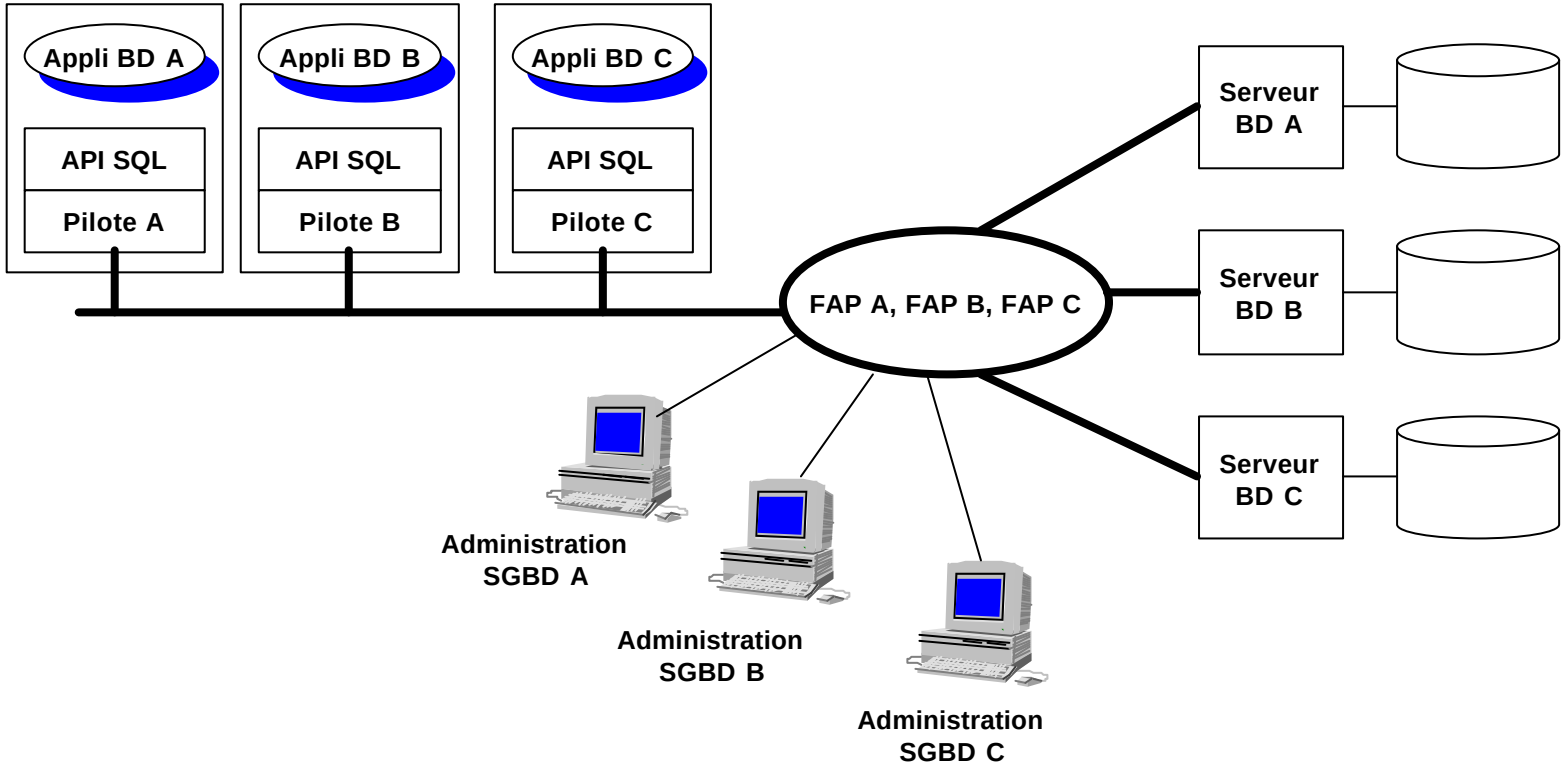


Contenu

- Définitions
- Exemple de BD répartie
- Répartition des données
- Répartition - Fédération
- Fédération de BD
  - Quelques cas de conflits
- Traduction des schémas
- Architecture de référence
- ➔ Accès aux BD multiples
  - API commune
  - FAP commun avec passerelles
  - FAP commun supporté par les SGBD
- Niveaux de transparence à la localisation
  - Client/Multibases :
    - RDA, DRDA, SQL-CLI, ODBC
  - Vues réparties
  - SGBD répartis
- Quelques problèmes des BD réparties et fédérées
- Partitionnement et placement des données
  - Quelques règles pour le partitionnement
  - Expédition de données et Expédition de Fonction
  - Recherche du partitionnement idéal
- Optimisation des requêtes réparties
- Réplication dans les BD
- Un aperçu sur les SGBD du commerce
  - Un peu d'histoire : Ingres/STAR
  - IBM DB2, Informix, Oracle, Sybase
- Évaluation des SGBD répartis
- Bibliographie

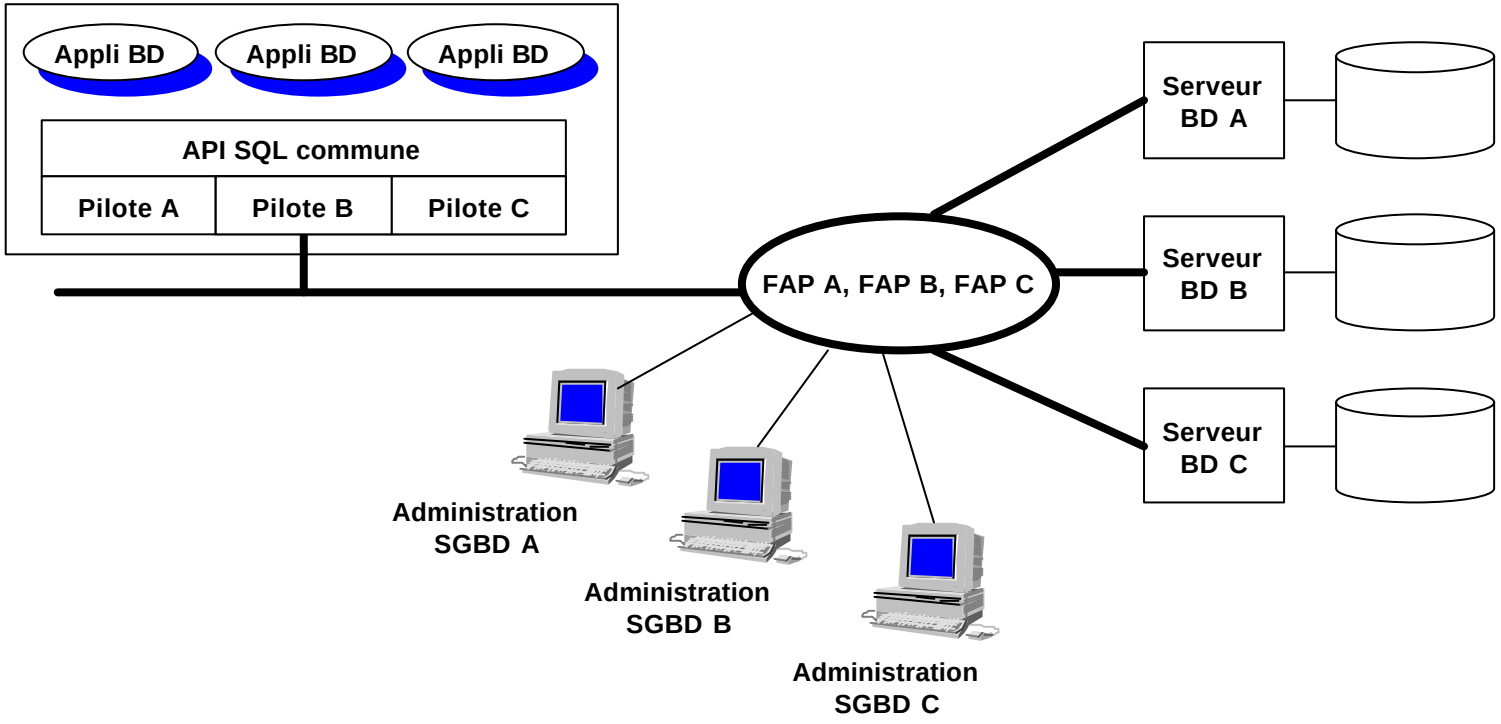
Accès aux BD multiples

- **Éditeurs multiples - la situation la plus difficile :**



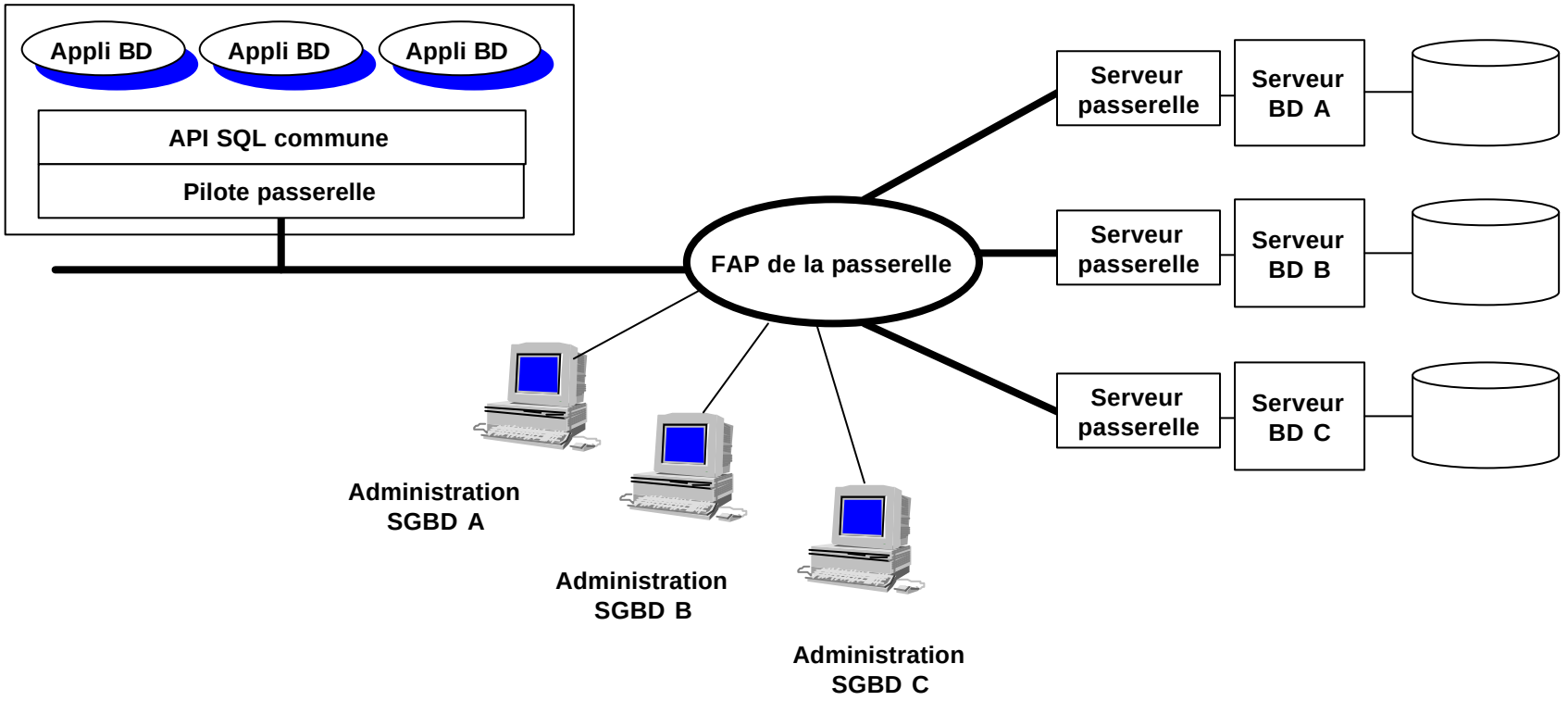
Accès aux BD multiples (2)

Solution N°1 : API (SQL) commune



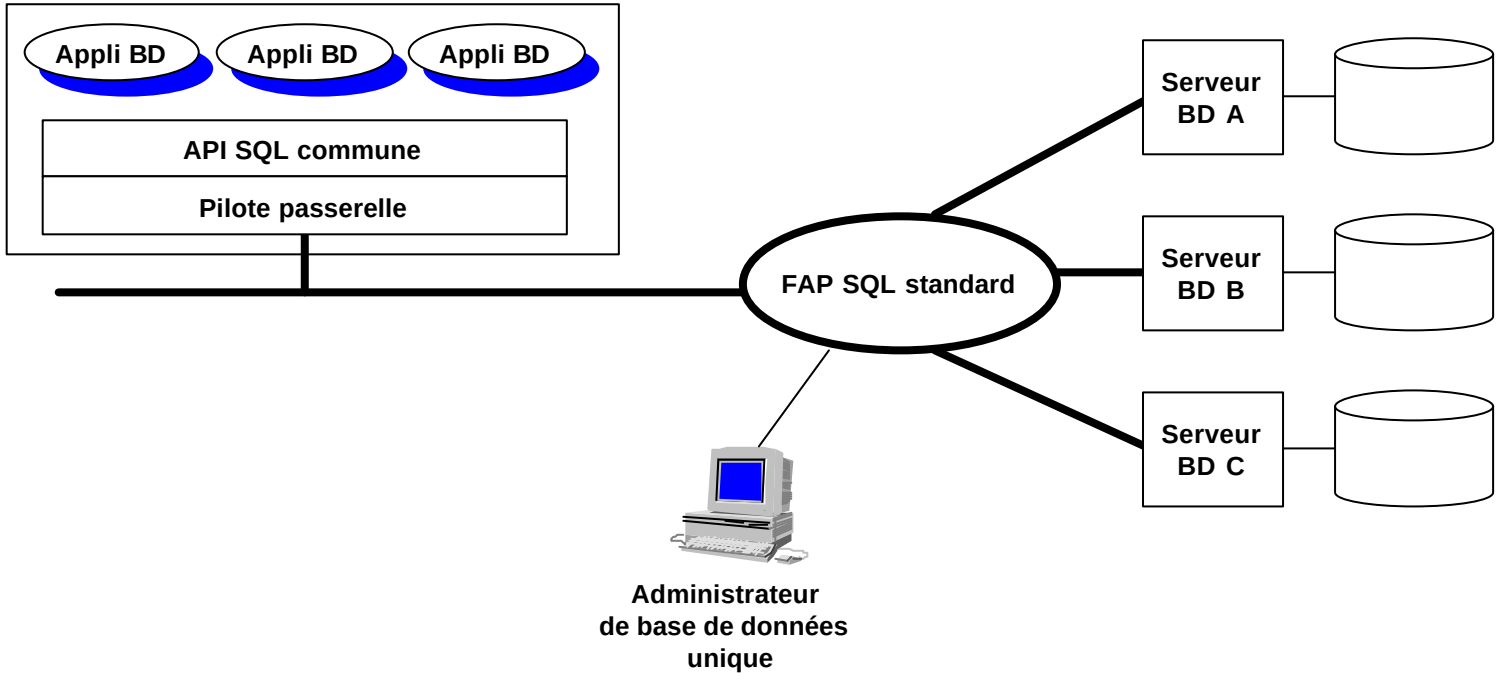
Accès aux BD multiples (3)

Solution N°2 : FAP commun avec passerelles



Accès aux BD multiples (4)

- **Solution N°3 (idéale) : FAP commun implémenté par les fournisseurs de SGBD**



Contenu

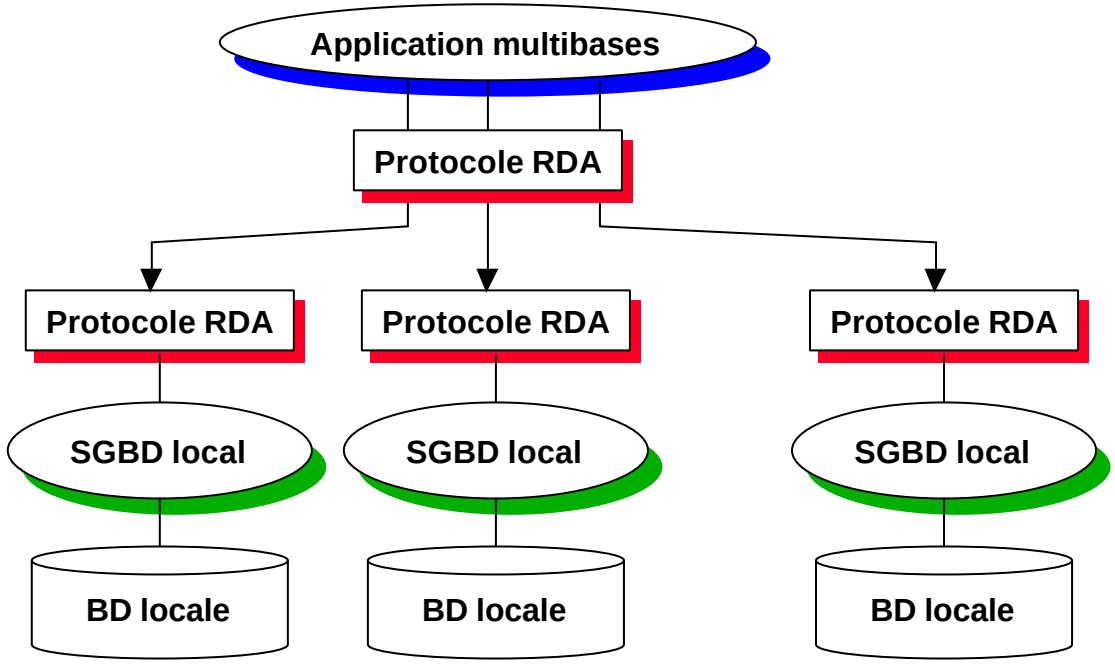
- Définitions
- Exemple de BD répartie
- Répartition des données
- Répartition - Fédération
- Fédération de BD
  - Quelques cas de conflits
- Traduction des schémas
- Architecture de référence
- Accès aux BD multiples
  - API commune
  - FAP commun avec passerelles
  - FAP commun supporté par les SGBD
- ➔ Niveaux de transparence à la localisation
  - Client/Multibases :
    - RDA, DRDA, SQL-CLI, ODBC
  - Vues réparties
  - SGBD répartis
- Quelques problèmes des BD réparties et fédérées
- Partitionnement et placement des données
  - Quelques règles pour le partitionnement
  - Expédition de données et Expédition de Fonction
  - Recherche du partitionnement idéal
- Optimisation des requêtes réparties
- Réplication dans les BD
- Un aperçu sur les SGBD du commerce
  - Un peu d'histoire : Ingres/STAR
  - IBM DB2, Informix, Oracle, Sybase
- Évaluation des SGBD répartis
- Bibliographie

Niveaux de transparence à la localisation

- **Trois types d'accès :**
  - **Client / Multibases :**
    - **RDA (Remote Data Access) - Standard ISO**
    - **DRDA (Distributed Relational Database Architecture) d'IBM (semble être en voie de disparition)**
    - **SQL-CLI (Call Level Interface) de l'Open Group**
    - **ODBC (Open Database Connectivity) de Microsoft**
    - **JDBC (Java Database Connection) de SUN**
  - **Vues réparties (sur BD fédérées)**
  - **SGBD réparti**

RDA - Remote Data Access

- **Les usagers connaissent la localisation**
- **Si une jointure est nécessaire, elle doit être réalisée par l'application**



RDA - Remote Data Access (2)

- **Exemple**
  - **Site 1 : Cartes grises**

    Personne (N° personne, nom, prénom, adresse, …)
    Voiture (N° véhicule, marque, type, …)
    Conducteur (N° personne, N° véhicule, NB_accidents,..)
  - **Site 2 : Base SAMU**

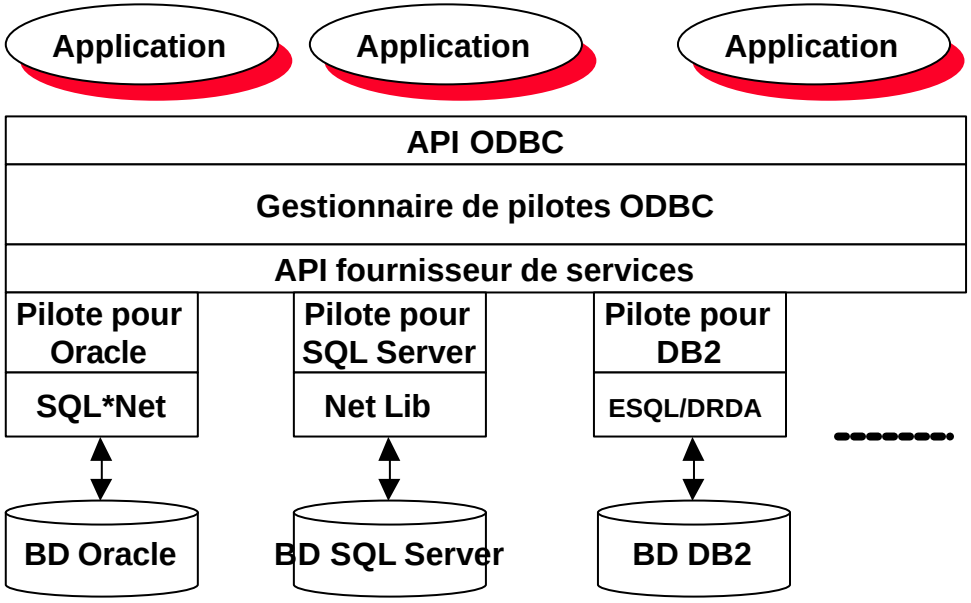
    Accident (N° accident, date, departement, N° véhicule, N° personne, …)
    Blessé (N° accident, N° personne, gravité, ….)
  - **Site 3 : requête**

    Liste des blessés graves dans une voiture de marque xxx et de type yyy dans région parisienne
    - **Requête en centralisé :**
      - **SELECT P.nom,P.prénom FROM Personne P, Blessé B, Accident A, Voiture V**
        WHERE P.N° personne = B.N° personne
        AND B.gravité > « commotion »
        AND B.N° accident = A.N° accident
        AND A.N° véhicule = V.N° véhicule
        AND V. marque = « xxx »
        AND V. type = « yyy»
        AND A.département IN (75, 78 , 91, 92 ,93, 94, 95)
- **Solution RDA**
  - **Requête sur site 1 : SELECT N° véhicule FROM Voiture WHERE marque = « xxx » AND type = « yyy » INTO temp1**
  - **Requête sur site 1 : SELECT * FROM Personne INTO temp2**
  - **Requête sur site 2 : SELECT B.N° personne, A.N° véhicule FROM Blessé B, Accident A**
    WHERE B.gravité > « commotion » AND B.N° accident = A.N° accident
    AND A.département IN (75, 78 , 91, 92 ,93, 94, 95) INTO temp3
- **Conclusion**
  - **Il est nécessaire d'envoyer 3 requêtes pour seulement 2 sites**
  - **La totalité de la relation Personne doit être transférée**
  - **L'intégration du résultat final doit être faite par l'application :**

    SELECT P.nom, P.prénom FROM temp2 P, temp3 B, temp1 V
    WHERE P.N° personne = B.N° personne
    AND B.N° véhicule = V.N° véhicule
- **Difficultés : Programmation et adhésion de l'industrie des SGBD au protocole RDA**

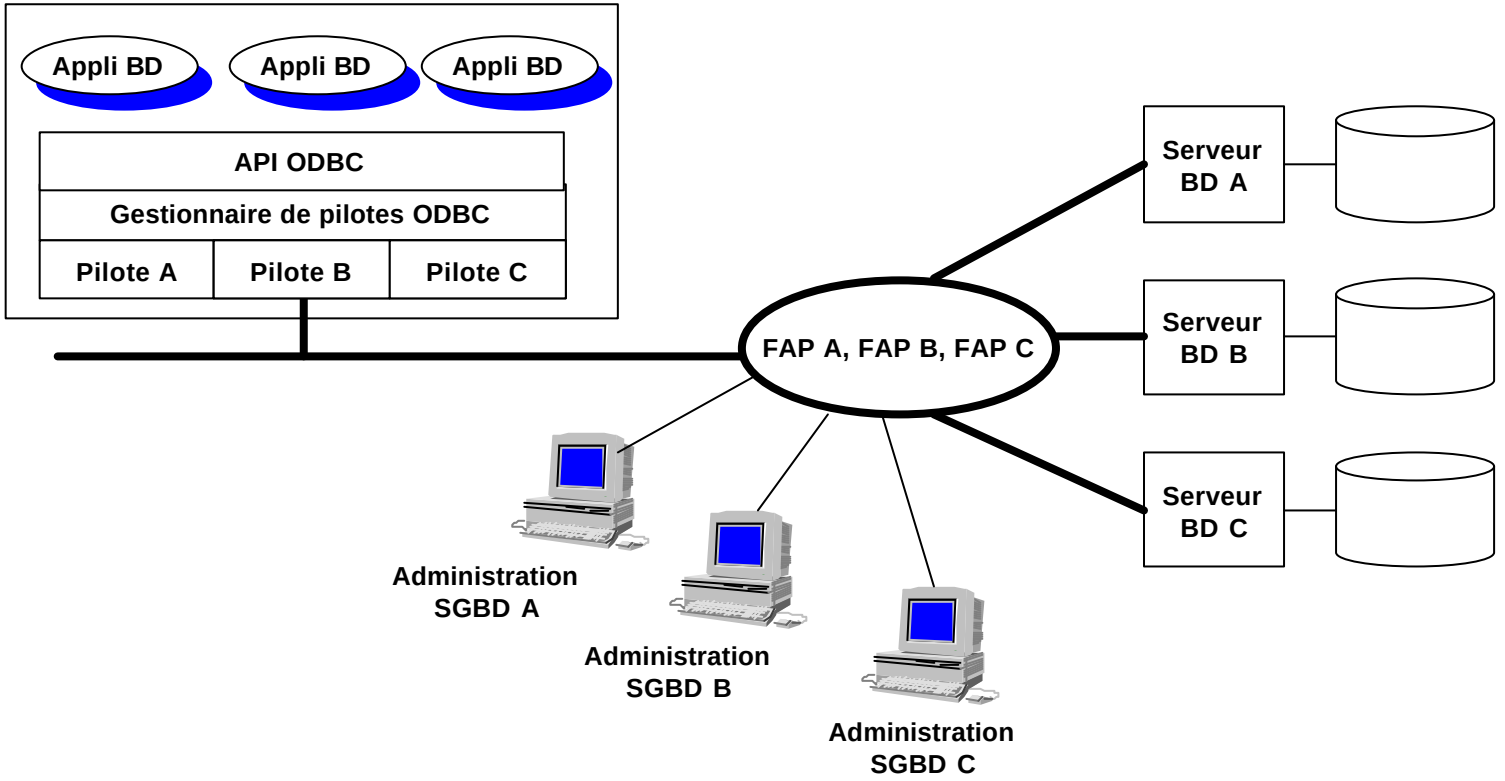
SQL-CLI

- **Formation, en 1988, d'un consortium (le SAG pour SQL Access Group) regroupant 44 éditeurs de SGBD avec pour objectif de définir :**
  - **un standard d'interopérabilité entre clients et SGBD;**
  - **une interface (CLI Call Level Interface) définissant un ensemble d'API (Application Programming Interface) communes pour les différents SGBD.**
- **Exemple : Composants de la CLI ODBC de Microsoft**



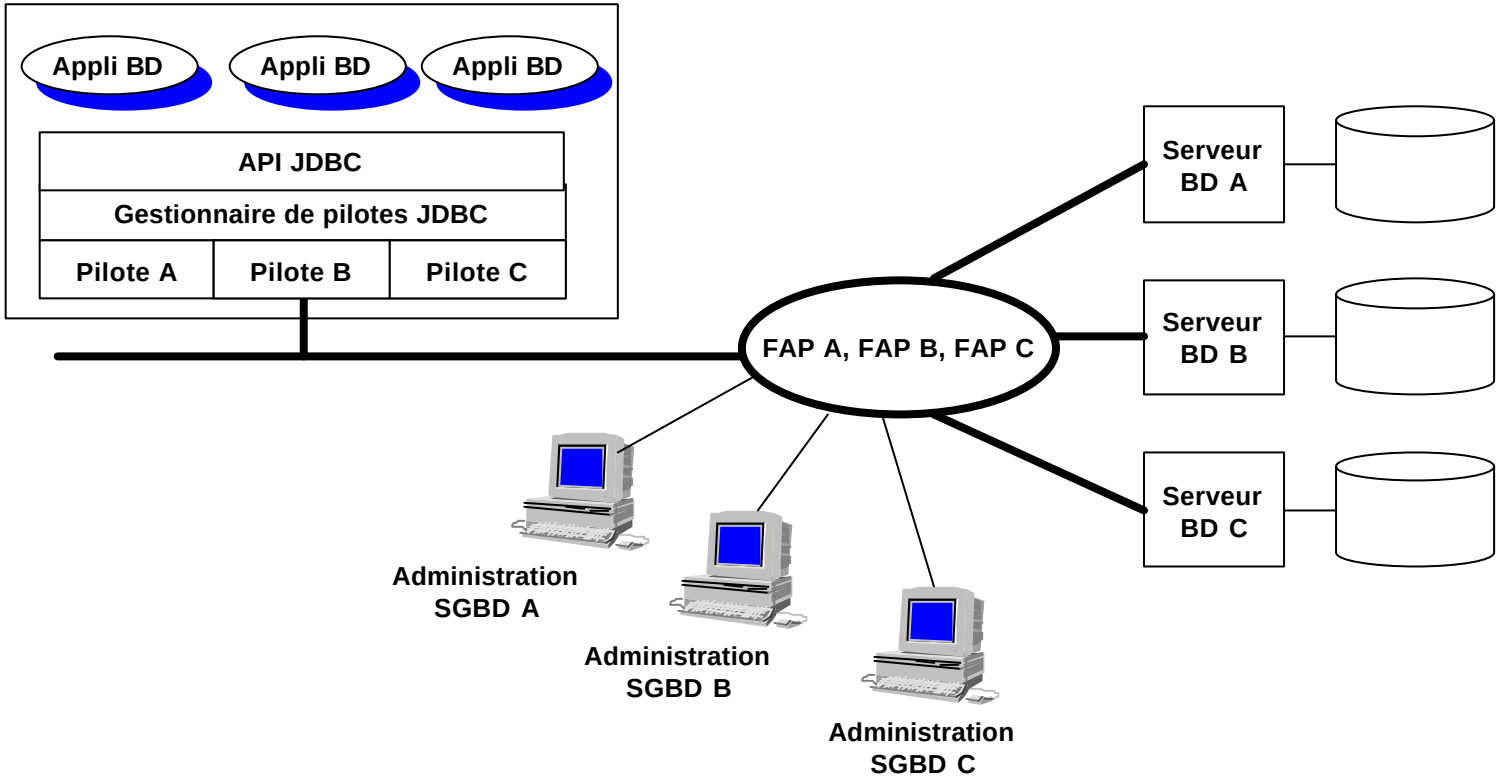
ODBC - Open Database Connectivity

- **Spécification contrôlée par Microsoft et supportée par les principaux fournisseurs de SGBD**
- **Difficulté : niveau de SQL supporté, développement des pilotes, ..**



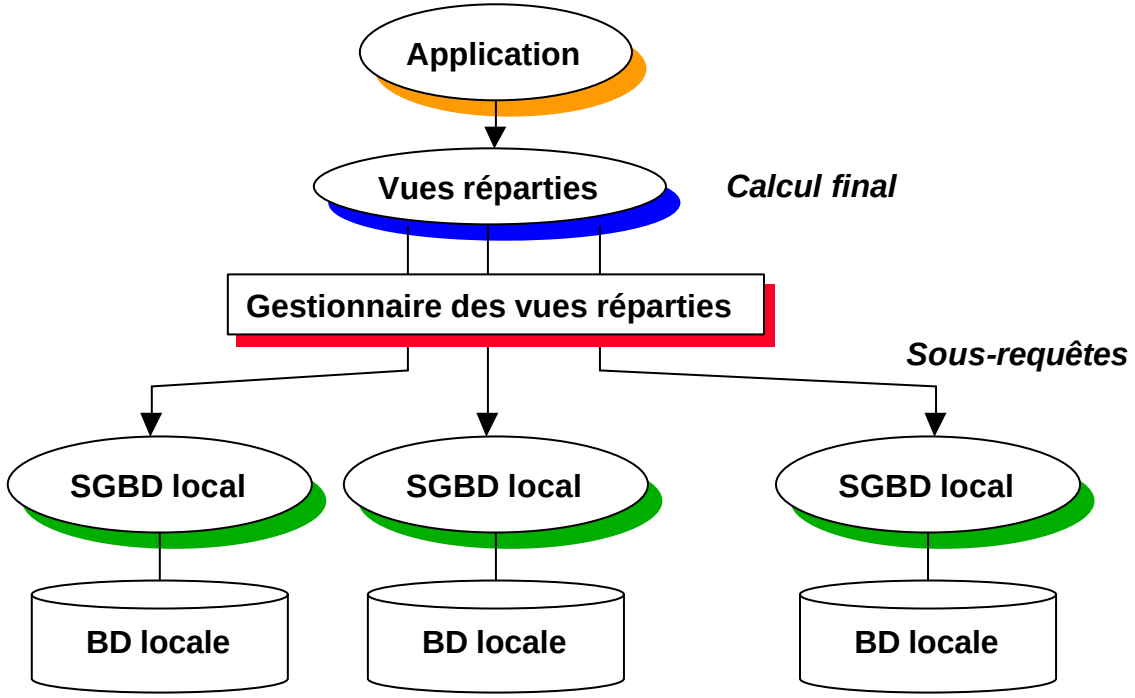
JDBC - Open Database Connection

- **Spécification commune à Sun et à différents fournisseurs de SGBD**
- **Difficulté : risque potentiel d'intrusion dans des systèmes par l'intermédiaire du code mobile (byte-code)**



Vues réparties

- La transparence à la localisation est assurée par la définition des vues réparties
- Les jointures inter-bases sont exécutées par le système
- Les mises à jour sont supportées au moyen des vues réparties
- Un protocole de validation à 2 phases est supporté



Vues réparties (2)

- **Définition de la vue répartie (sur le site 3)**
  - **CREATE VIEW Accidenté-grave {N°personne, nom, prénom, adresse,gravité, département, N° véhicule, marque, type}**
    AS SELECT P.N°personne, P.nom, P.prénom, P.adresse, B.gravité, A.département,V.N° véhicule, V.marque, V.type
    FROM S1.Personne P, S2.Blessé B, S2.Accident A, S1.Voiture V
    WHERE P.N°personne = B.N°personne
    AND B.gravité > « commotions »
    AND A.N°véhicule = V.N°véhicule
    AND A.N°.accident = B.N°accident

- **Requête sur la vue répartie (sur le site 3) : liste des blessés graves dans une voiture yyy de marque xxx dans la région parisienne**

  - **SELECT N°personne,nom,prénom,adresse**
    FROM Accidenté-grave
    WHERE marque = « xxx »
    AND type = « yyy »
    AND département IN (75, 78, 91, 92, 93, 94, 95)

Vues réparties (3)

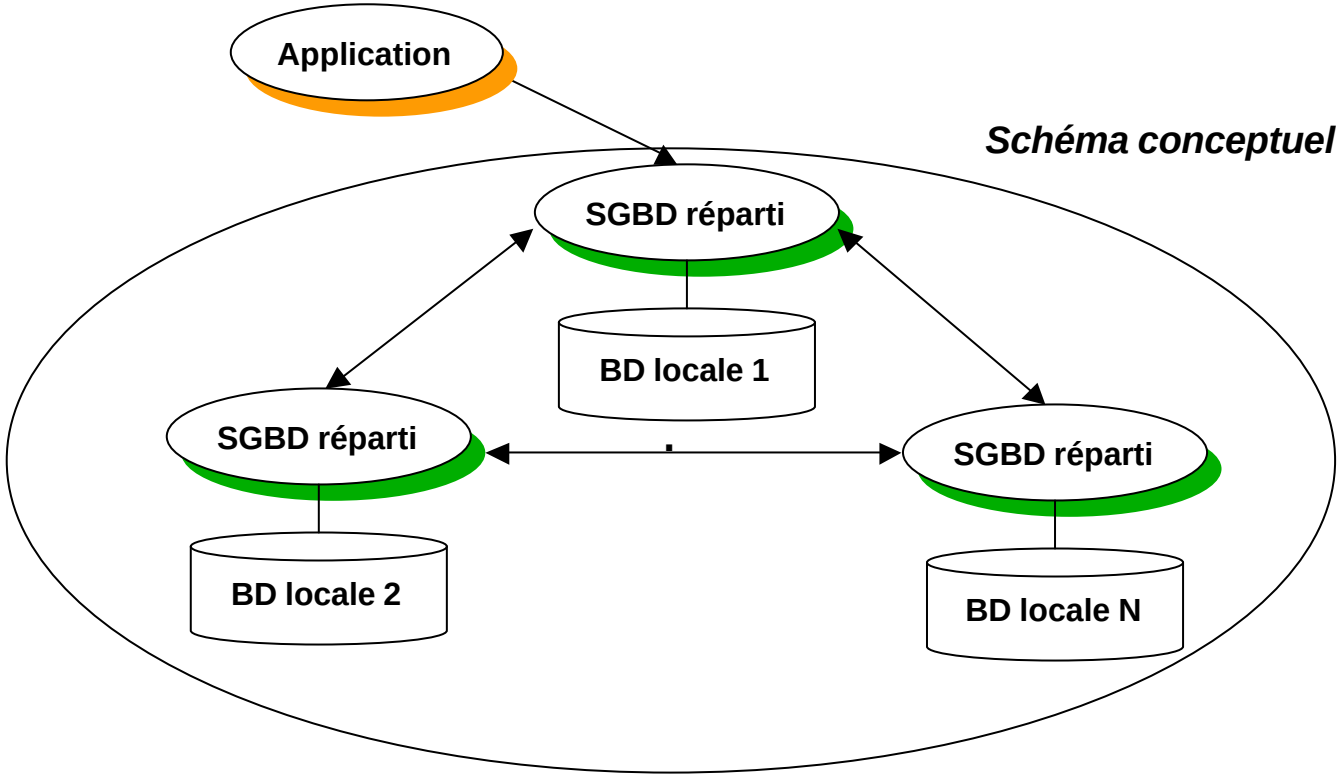
- **Fonctions réalisées par le gestionnaire des vues réparties :**
  - **La transformation de la requête sur les relations de base**
  - **La décomposition de la requête en requêtes mono-site :**
    - **Requête sur site 1 : SELECT N°véhicule FROM Voiture ….**
    - **Requête sur site 1 : SELECT * FROM Personne …**
    - **Requête sur site 2 : SELECT B.N°personne, A.N°véhicule FROM Blessé B, Accident A, ….**
  - **Le contrôle de l'exécution des requêtes**
  - **L'intégration du résultat en effectuant les différentes opérations (dont les jointures)**
- **Conclusion**
  - **Le système apparaît à l'application comme un vrai SGBD réparti**

    mais

  - **Il y a toujours 3 requêtes différentes pour 2 sites**
  - **La totalité de la relation personne doit être transférée**

SGBD réparti

- **La transparence à la localisation est assurée par la définition de la base répartie**
- **Les différentes opérations sont prises en charge par les différents SGBD**
- **Un protocole de validation à 2 phases est supporté**



SGBD réparti (2)

- **Schéma conceptuel de la base :**
  - Personne (N° personne, nom, prénom, adresse, …)
  - Voiture (N° véhicule, marque, type, …)
  - Conducteur (N° personne, N° véhicule, NB_accidents,..)
  - Accident (N° accident, date, département, N° véhicule, N° personne, …)
  - Blessé (N° accident, N° personne, gravité, ….)
- **Implémentation de la base :**
  - **Sites 75, 78, 91, 92, 93, 94, 95**
    - **Bases préfectorales avec Voitures, Conducteur et Personne pour les voitures immatriculées dans le département (Personne, Voiture, Conducteur)**
  - **SAMU : base SAMU de la région parisienne (Accident, Blessé)**
- **La requête « liste des blessés graves dans une voiture type yyy de marque xxx dans la région parisienne » émane d'un site appelé Interrogation**

SGBD réparti (3)

- **Plan d'exécution répartie**
  - **Requête sur site SAMU :**
    - **SELECT B.N°personne, A.N°véhicule FROM Blessé B, Accident A WHERE B.N°accident = A.N°accident AND B.gravité > « commotions » AND A.département IN (75, 78, 91, 92, 93, 94, 95) INTO temp1**
    - **SEND temp1 to S75, S78, S91, S92, S93, S94, S95**
  - **Requêtes sur S75, S78, S91, S92, S93, S94, S95 :**
    - **RECEIVE temp1 FROM SAMU**
    - **SELECT P.nom,P.prénom FROM Personne P, temp1 T, Voiture V WHERE P.N°personne = T.N°personne AND T.N°véhicule = V.N°véhicule AND V.marque = « xxx » AND V.type = « yyy » INTO temp2.i**
    - **SEND temp2.i TO Interrogation**
  - **Requête sur site Interrogation :**
    - **RECEIVE temp2.75 FROM S75**
      ……
      RECEIVE temp2.95 FROM S95
      UNION temp2.75, temp2.78,…..temp2.95
      INTO résultat
- **Conclusion**
  - **Le SGBD réparti a pris en charge tous les problèmes liés à la répartition**
  - **Les transferts sont minimisés :**
    - **seuls les N° des blessés et des véhicules sont transférés**

Contenu

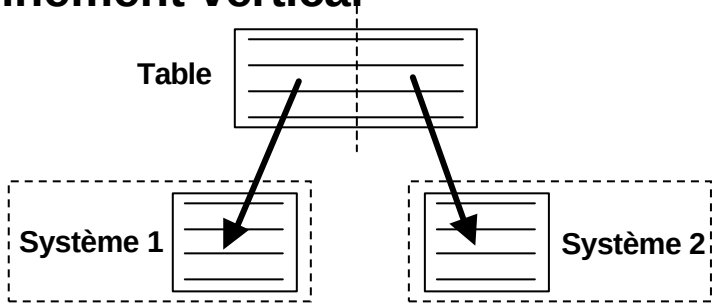
- Définitions
- Exemple de BD répartie
- Répartition des données
- Répartition - Fédération
- Fédération de BD
  - Quelques cas de conflits
- Traduction des schémas
- Architecture de référence
- Accès aux BD multiples
  - API commune
  - FAP commun avec passerelles
  - FAP commun supporté par les SGBD
- Niveaux de transparence à la localisation
  - Client/Multibases :
    - RDA, DRDA, SQL-CLI, ODBC
  - Vues réparties
  - SGBD répartis
- ➔ Quelques problèmes des BD réparties et fédérées
- Partitionnement et placement des données
  - Quelques règles pour le partitionnement
  - Expédition de données et Expédition de Fonction
  - Recherche du partitionnement idéal
- Optimisation des requêtes réparties
- Réplication dans les BD
- Un aperçu sur les SGBD du commerce
  - Un peu d'histoire : Ingres/STAR
  - IBM DB2, Informix, Oracle, Sybase
- Évaluation des SGBD répartis
- Bibliographie

Quelques problèmes des BD réparties et fédérées

- **Validation à deux phases**
  - **Dès qu'une mise à jour s'adresse à plus d'un site ou à plus d'une base sur un même site, il convient d'utiliser le protocole de validation à deux phases**
- **Verrouillage**
  - **Les SGBD utilisent le verrouillage à deux phases pour assurer la sérialisation des transactions (phase d'acquisition des verrous puis phase de relâchement des verrous)**
  - **Un SGBD sait détecter les étreintes fatales locales (détection d'un cycle dans le graphe d'attente)**
  - **Dans le cas distribué, on peut utiliser plusieurs techniques pour traiter le cas des étreintes fatales :**
    - **Prévention = Éviter que le problème ne survienne :**
      - **Technique d'estampillage : dater les transactions et « tuer » les transactions en attente en fonction de leur « âge » :**
        - **Die Wait = « tuer » les transactions demandant des ressources détenues par des transactions plus anciennes et reprendre la transaction « tuée » avec la même estampille**
        - **Wound Wait = « blesser » les transactions en attente de ressources détenues par une plus ancienne, on « tue » la transaction « blessée » si elle demande une ressource détenue par une autre transaction. On reprend la transaction « tuée ».**
    - **Détection :**
      - **Construction d'un graphe global d'attente par union des graphes locaux**
    - **Présomption :**
      - **Abandon des transactions n'ayant pas terminé leur exécution après un certain temps (horloge de garde ou Watch Dog)**

Partitionnement et placement des données

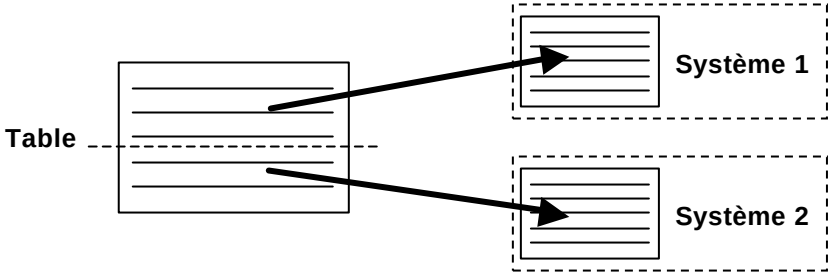
- **Objectifs**
  - **Réduction de la charge (accès aux données, communication, espace de recherche)**
  - **Équilibrage de charge**
  - **Accroissement du travail utile (e.g. effet de cache)**
- **Partitionnement vertical**



• Projection, dont un attribut commun, sur chacun des sites
• Table globale reconstituée par une jointure selon cet attribut

Note : Relation avec l'organisation des applications (minimisation des interactions entre les systèmes et validation à deux phases)

- **Partitionnement horizontal**



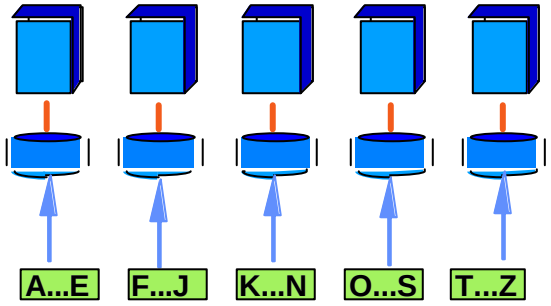
• Sélection, selon des valeurs disjointes d'un critère sur chaque site
• Table globale reconstituée par UNION

Note : Prise en compte de la validation à deux phases par le SGBD

Partitionnement des données (2)

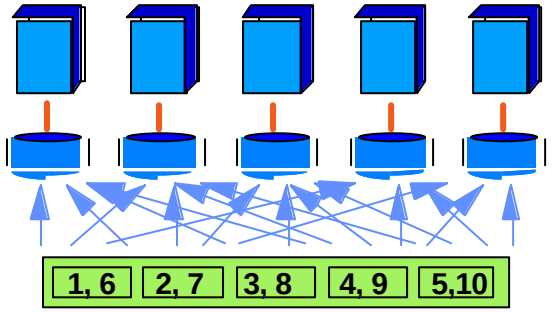
Méthodes de partitionnement horizontal (exemples)

Domaine



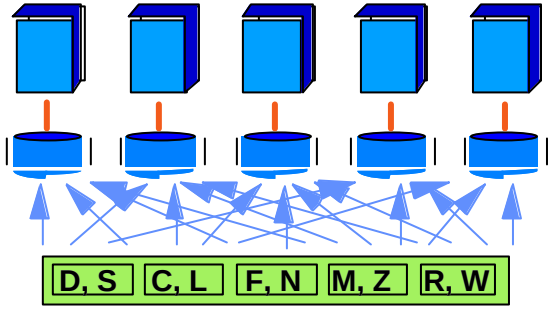
Les données sont réparties en fonction des domaines de valeur des clés

Round robin



Chaque article est rangé dans la partition suivante en séquence

Hash



Un algorithme appliqué à la clé de l'article détermine son numéro de partition

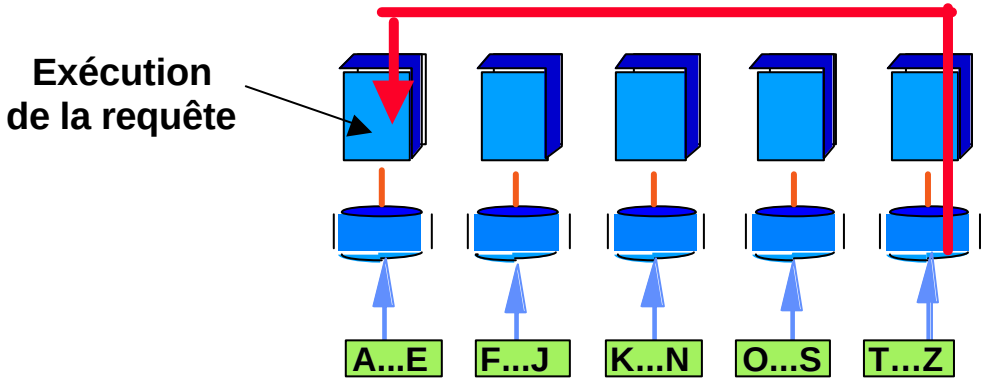
Note : Il existe des méthodes hybrides, ce sont des combinaisons/variations des méthodes de base.

Quelques règles pour le partitionnement

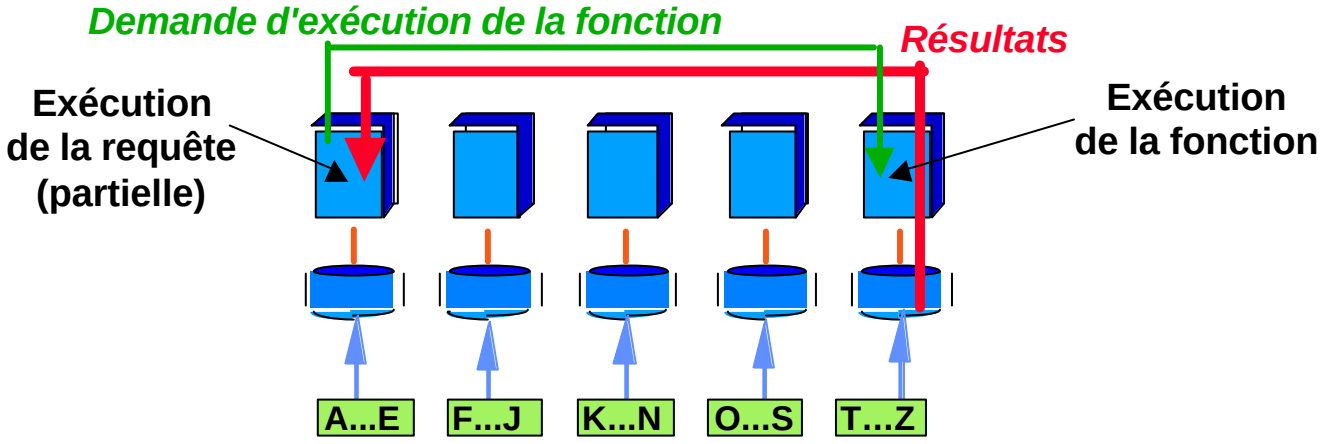
- **Equilibrage des partitions (pour éviter le "data skew")**
- **Quelques caractéristiques des méthodes de base**
  - **Domaine de valeur**
    - **Permet des optimisations**
    - **Risque de déséquilibre des partitions et de la charge**
  - **Round Robin**
    - **Équilibre, par définition, des partitions**
    - **Ne facilite pas la réduction de la charge**
  - **Hash**
    - **Choix de la fonction de hashing**
    - **Pas optimal pour les recherches fondées sur des domaines de valeur**
- **Possibilité de dupliquer les données : problème avec les mises à jour (validation à deux phases)**

Expédition de données et Expédition de Fonction

- **Deux modèles fonctionnels dans le cas d'une architecture de SGBD réparti**
  - **Expédition de données "Data Shipping"**



  - **Expédition de fonction "Function Shipping"**



Recherche du partionnement idéal

- **Exposé :**
  - **Soit une BD répartie implantée sur un ensemble de p sites ($S=\{S_1, S_2,\ldots S_p\}$ et un ensemble de r fragments composant la base ($F=\{F_1, F_2,\ldots F_r\}$)**
  - **Trouver la répartition optimale de F sur S**
- **Formalisation :**
  - **Pour un fragment F_k, R_l et U_l sont, respectivement, les taux d'accès en lecture et en mise à jour depuis le site S_l**
  - **Soit C_{lm}, le coût de communication unitaire de S_l à S_m**
  - **Soit D_l le coût de stockage du fragment F_k sur le site S_l**
- **Problème :**
  - **Trouver l'assignation optimale de F_k $\{X_1, X_2,\ldots X_m\}$ telle que $X_l = 1$ si F_k est assigné sur S_l et 0 sinon et minimisant le coût de la communication et du stockage exprimé par la formule :**

$$\text{coût} = \sum_l \left(\sum_m (X_m \times U_m \times C_{lm} + R_m \times \text{Min}\{C_{lm} | X_m=1\})\right) + \sum_l X_l \times D$$

  - **Ce problème est NP-complet (ne pouvant pas être résolu efficacement). De plus, il est soumis à des contraintes de limitation (capacité de stockage, de communication, de temps de réponse).**

Contenu

- Définitions
- Exemple de BD répartie
- Répartition des données
- Répartition - Fédération
- Fédération de BD
  - Quelques cas de conflits
- Traduction des schémas
- Architecture de référence
- Accès aux BD multiples
  - API commune
  - FAP commun avec passerelles
  - FAP commun supporté par les SGBD
- Niveaux de transparence à la localisation
  - Client/Multibases :
    - RDA, DRDA, SQL-CLI, ODBC
  - Vues réparties
  - SGBD répartis
- Quelques problèmes des BD réparties et fédérées
- Partitionnement et placement des données
  - Quelques règles pour le partitionnement
  - Expédition de données et Expédition de Fonction
  - Recherche du partitionnement idéal
- ➔ Optimisation des requêtes réparties
- Réplication dans les BD
- Un aperçu sur les SGBD du commerce
  - Un peu d'histoire : Ingres/STAR
  - IBM DB2, Informix, Oracle, Sybase
- Évaluation des SGBD répartis
- Bibliographie

Optimisation des requêtes réparties

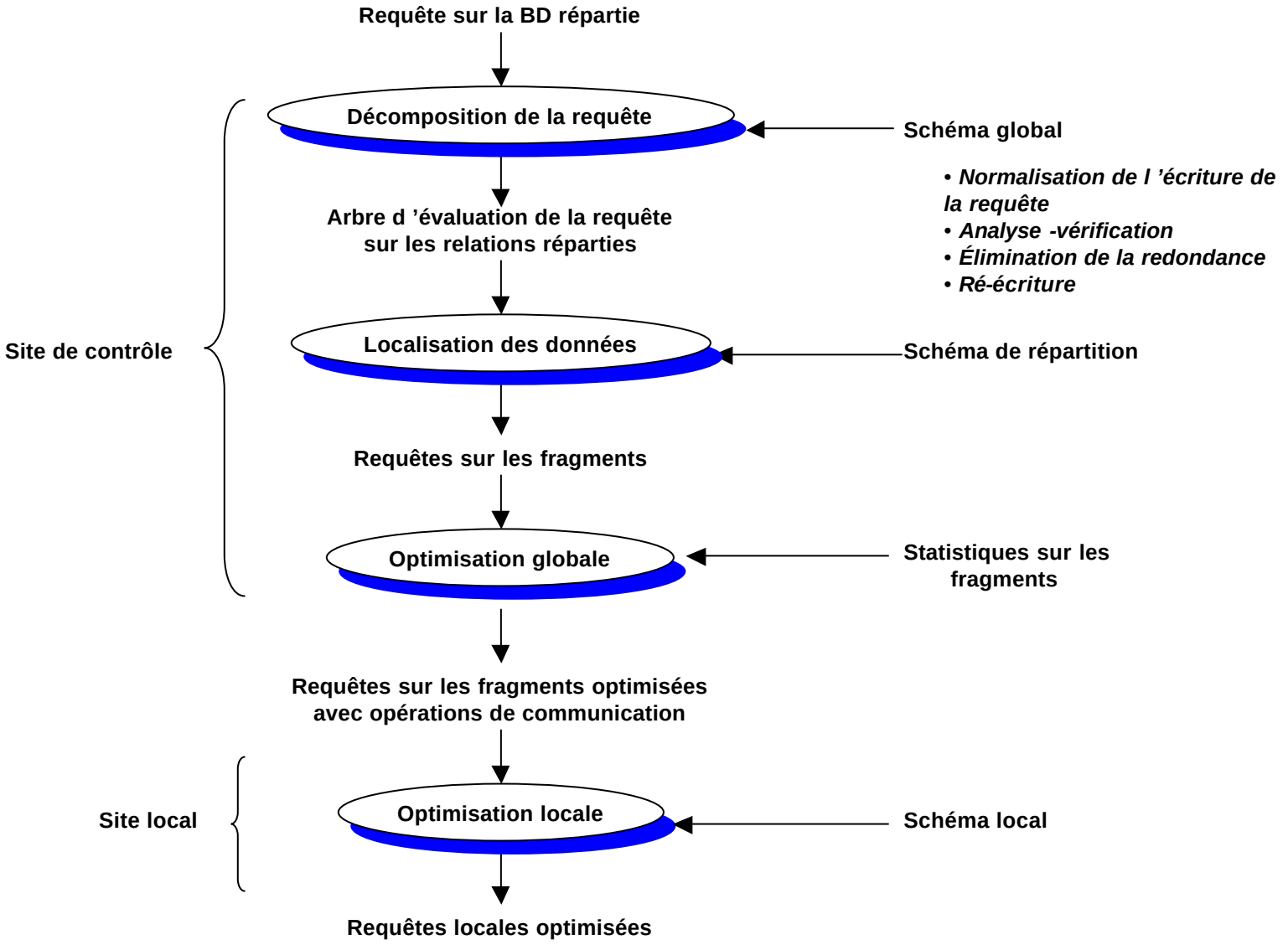
- **Établissement d'un plan d'évaluation optimal**
- **Optimisation d'une fonction de coût ou de temps de réponse de la forme :**

  coût global = a x coût(E/S) + b x coût(Processeur) + c x coût(Communication)
  + d x coût(Transfert des données)

- **Rappel : la compilation d'une requête SQL produit un arbre d'évaluation composé d'un certain nombre d'opérateurs de base :**
  - **Projection : P_X R projection de la relation R sur la liste d 'attributs X**
  - **Sélection : s_P R sélection des tuples de R vérifiant le prédicat P**
  - **Équijointure : R1 $\bowtie_A$ R2 jointure des relations R1 et R2 selon l'attribut A (R1.A=R2.A)**
  - **Produit cartésien : x produit de deux relations**
  - **Union : $\cup$ union de 2 relations**
  - **Intersection : $\cap$ intersection de deux relations**
- **L'optimisation vise à transformer l'arbre d'évaluation en un arbre optimal**

Optimisation des requêtes réparties (2)

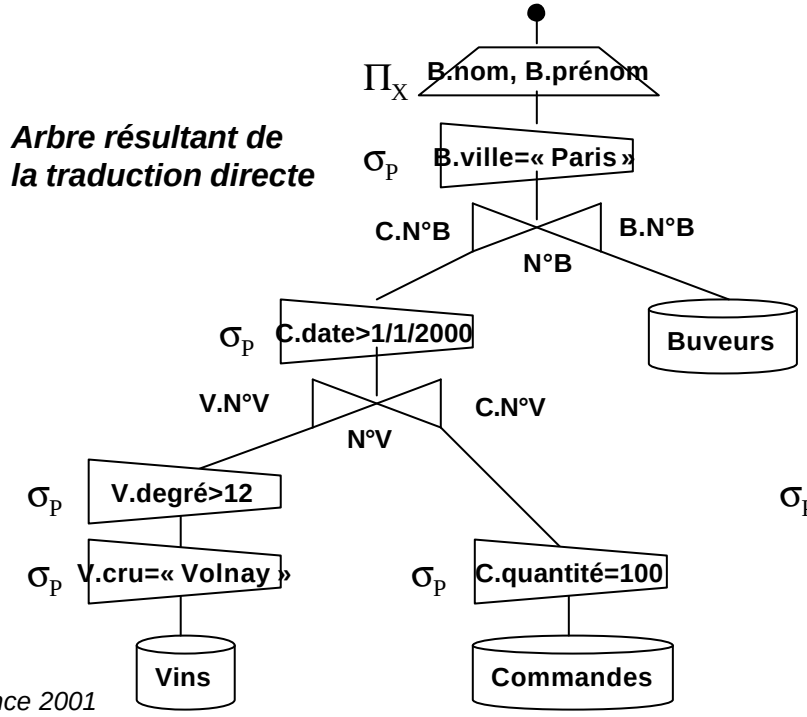
- **Schéma général de traitement et d'optimisation d'une requête répartie**



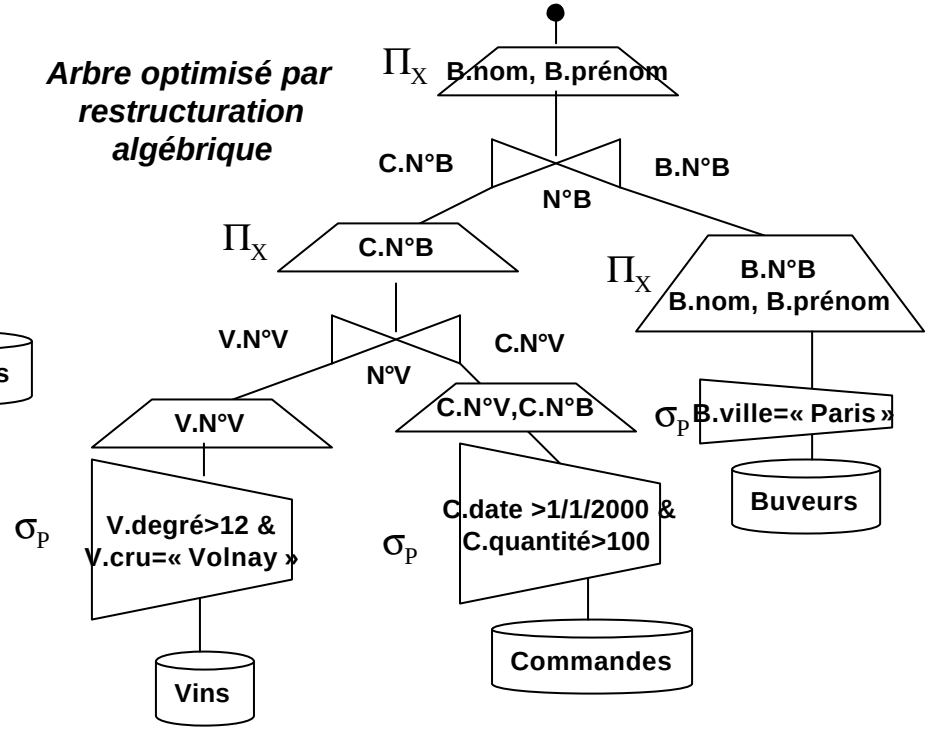
Optimisation des requêtes réparties (3)

- **Exemple - Définition de la base de données :**
  - **B : buveurs (N°B, nom, prénom, ville)**
  - **V : vins (N°V, cru, millésime, degré)**
  - **C : commandes (N°V,N°B, date, quantité)**
- **Exemple de requête de sa traduction et de son optimisation en l'absence de répartition :**
  - **SELECT nom, prénom FROM buveurs (B), vins (V), commandes (C)**
    WHERE V.cru = « Volnay » AND V.degré > 12 AND C.quantité > 100
    AND C.N°V = V.N°V AND C.date >1/1/2000 AND B.N°B = C.N°B
    AND B.ville = « Paris »

Arbre résultant de la traduction directe



Arbre optimisé par restructuration algébrique



Optimisation des requêtes réparties (4)

- **Hypothèse de volume :**
  - **Buveurs (B) : 10 000 tuples**
  - **Vins (V) : 1 000 tuples**
  - **Commandes : 200 000 tuples**
- **Hypothèse de distribution des BD :**
  - **Paris : Buveurs (B)**
  - **Dijon : Vins 1 (V1) restriction N°V<= 400**
    Commandes 1 (C1) restriction N°V<= 400
  - **Bordeaux : Vins 2 (V2) restriction N°V > 400**
    Commandes 2 (C2) restriction N°V>400
- **Requête émise sur le site de Paris :**
  « Noms des buveurs parisiens n'ayant pas commandé en décembre 2000 »
  Sélectivités supposées : 20% de parisiens et 1% n'ayant pas commandé
- **Stratégies :**
  - **Simpliste : transférer C1 et C2 vers Paris (200 000 tuples) et faire C = C1 È C2 et évaluer SELECT B.nom FROM Buveurs (B) WHERE B.ville = « Paris » AND B.N°B NOT IN (SELECT N°B FROM C WHERE C.date>1/12/2000 AND C.date<1/1/2001)**
  - **Améliorée : Transférer vers Dijon et Bordeaux Buveurs.N°B des seuls parisiens (= 2 x 2 000 petits tuples). Évaluer sur les sites de Dijon et Bordeaux : Buveurs.N°B NOT IN (SELECT N°B FROM Ci WHERE Ci.date>1/12/2000 AND Ci.date<1/1/2001)**
    Transférer les résultats vers Paris (= 2 x 20 petits tuples) et faire l'intersection des résultats

Contenu

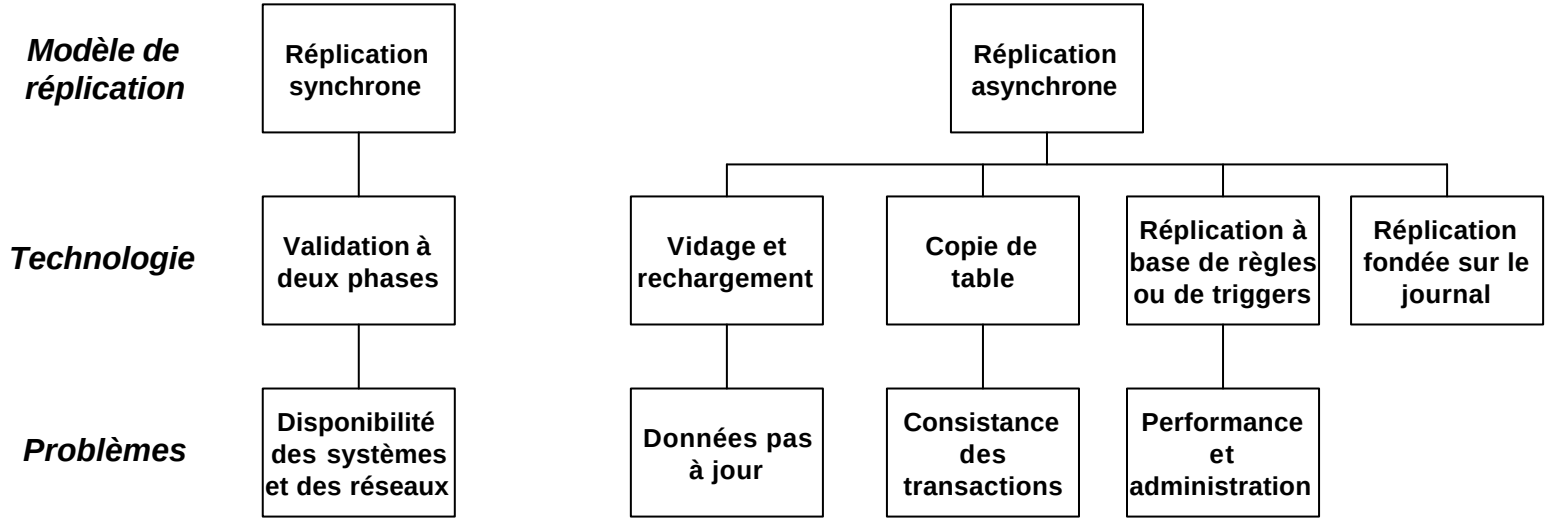
- Définitions
- Exemple de BD répartie
- Répartition des données
- Répartition - Fédération
- Fédération de BD
  - Quelques cas de conflits
- Traduction des schémas
- Architecture de référence
- Accès aux BD multiples
  - API commune
  - FAP commun avec passerelles
  - FAP commun supporté par les SGBD
- Niveaux de transparence à la localisation
  - Client/Multibases :
    - RDA, DRDA, SQL-CLI, ODBC
  - Vues réparties
  - SGBD répartis
- Quelques problèmes des BD réparties et fédérées
- Partitionnement et placement des données
  - Quelques règles pour le partitionnement
  - Expédition de données et Expédition de Fonction
  - Recherche du partitionnement idéal
- Optimisation des requêtes réparties
- ➔ Réplication dans les BD
- Un aperçu sur les SGBD du commerce
  - Un peu d'histoire : Ingres/STAR
  - IBM DB2, Informix, Oracle, Sybase
- Évaluation des SGBD répartis
- Bibliographie

Réplication dans les BD

- **Objectifs de la réplication :**
  - **Amélioration de la disponibilité des données**
  - **Amélioration des performances**
- **Difficultés de la réplication :**
  - **Synchronisation des copies**
  - **Transparence de la gestion**
- **Mise à jour synchrone et asynchrone**
  - **Synchrone :**
    - **+ Maintien de toutes les copies en cohérence**
    - **- Perte de performance du fait de la mise en œuvre de la validation à deux phases**
  - **Asynchrone : mise à jour différées des copies**
    - **+ Incidence minime sur les performances**
    - **- Nécessité de mise à niveau de la copie ou des copies en cas de reprise**

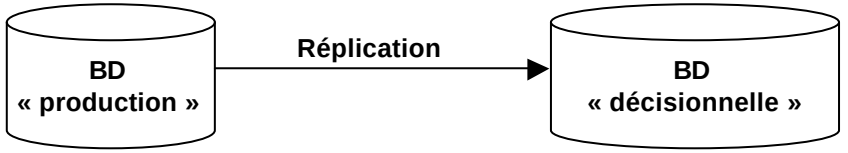
Réplication des données

- **Objectifs de la réplication de données :**
  - **Disponibilité des données**
  - **Respect de l'intégrité des données**
  - **Optimisation des accès**
  - **Administration centralisée**
  - **Gestionnaires de données hétérogènes**
  - **Autonomie locale**
- **Options de réplication de données :**

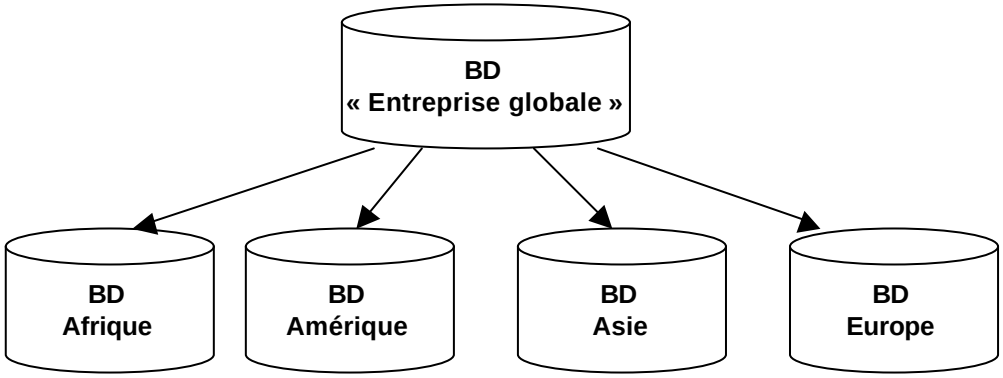


Réplication des données (2)

- **Quelques exemples d'utilisation**
  - **Mouvement d'information (OLTP þ DSS)**

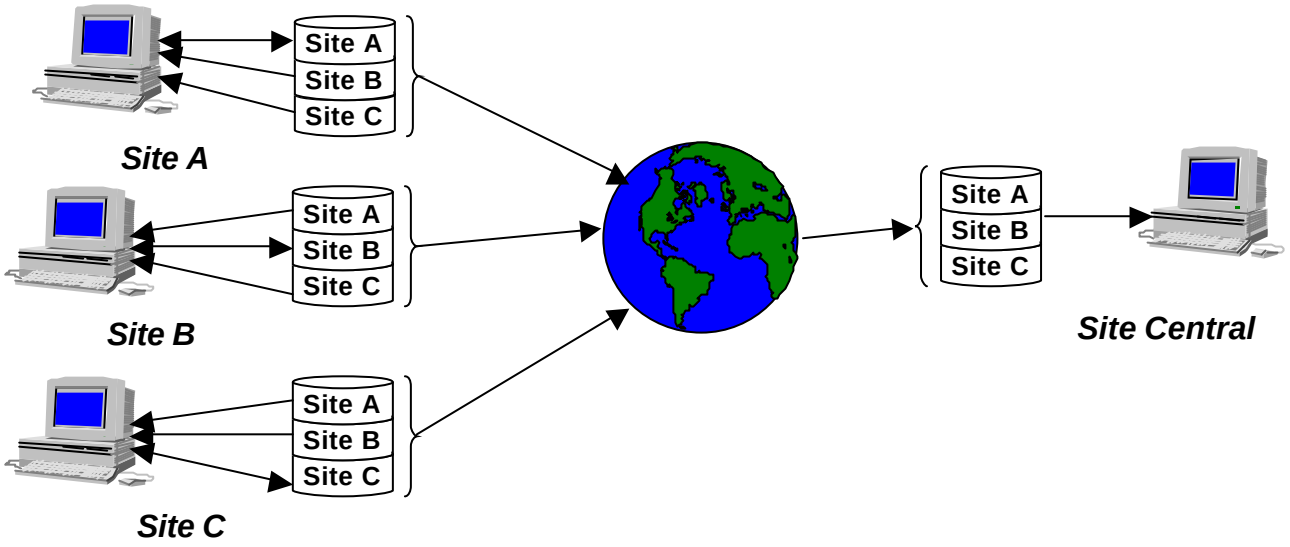


  - **Distribution d'information**

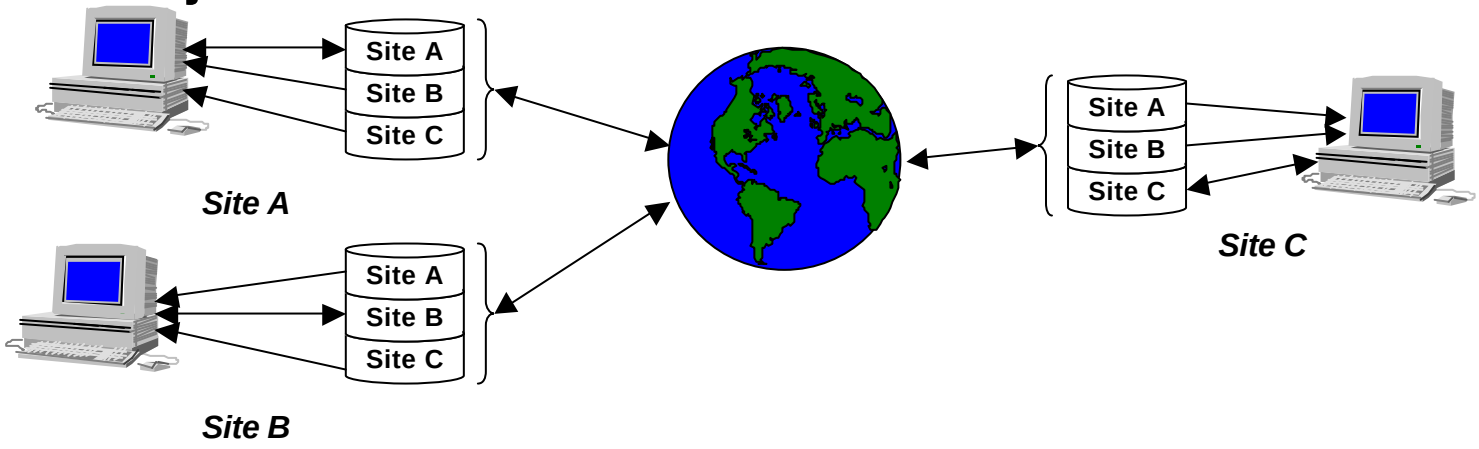


Réplication des données (3)

- **Consolidation d'informations**

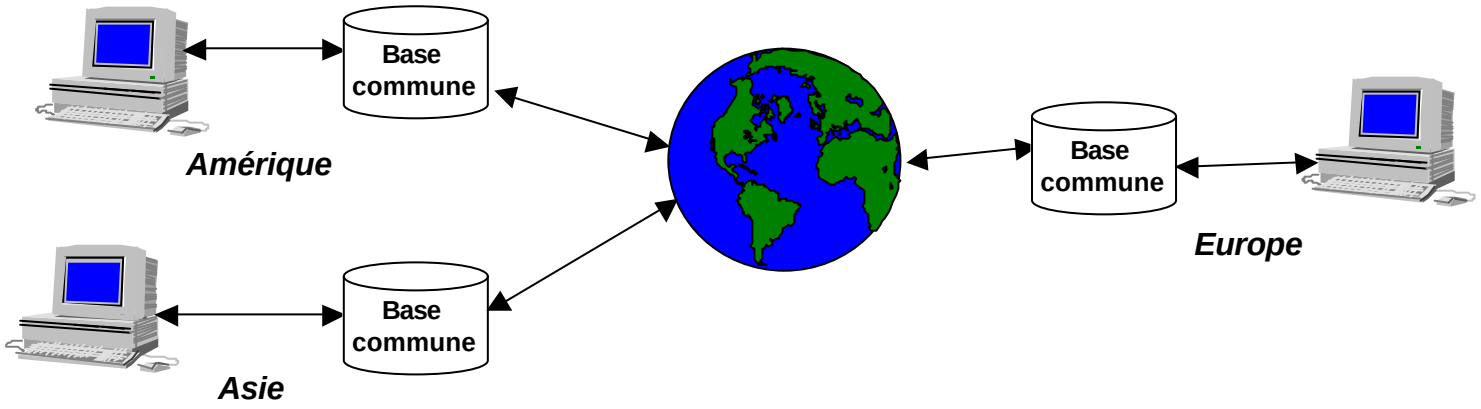


- **Mises à jour sans conflit**



Réplication des données (4)

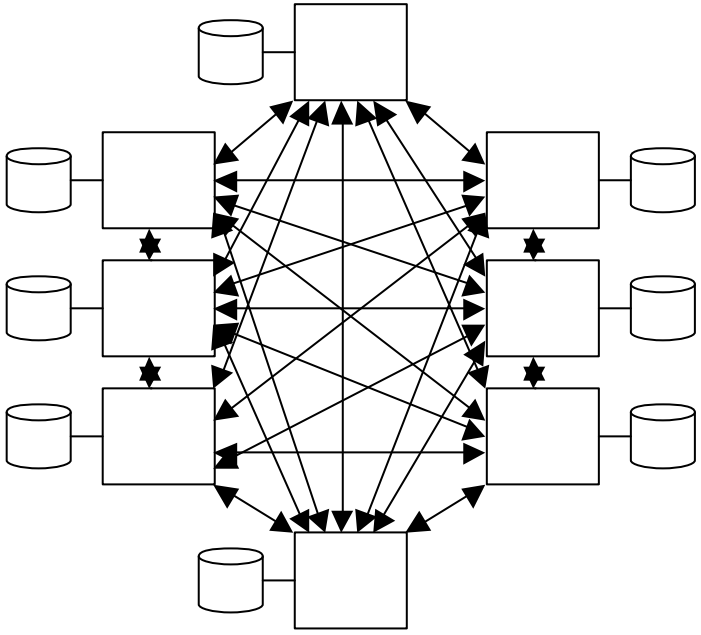
- **Mises à jour avec conflits d'accès**
  - **Exemple : société de logiciel au niveau mondial (travaillant donc 24 x 24 du fait des décalages horaires) maintenant une base de données pour le support de ses clients (erreur connues, corrections, détours,….). La base doit être accessible en permanence et être à jour.**



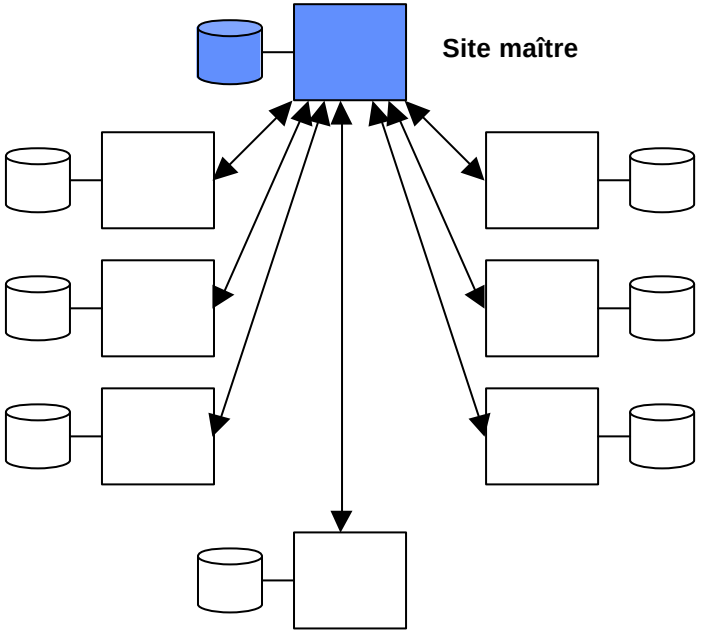
  - **La résolution des problèmes de mise à jour implique la mise en œuvre d'une stratégie particulière (e.g. mise à jour d' une copie maître qui est ensuite propagée)**

Réplication des données (5)

Illustration de stratégies de mise à jour en cas de conflit d'accès



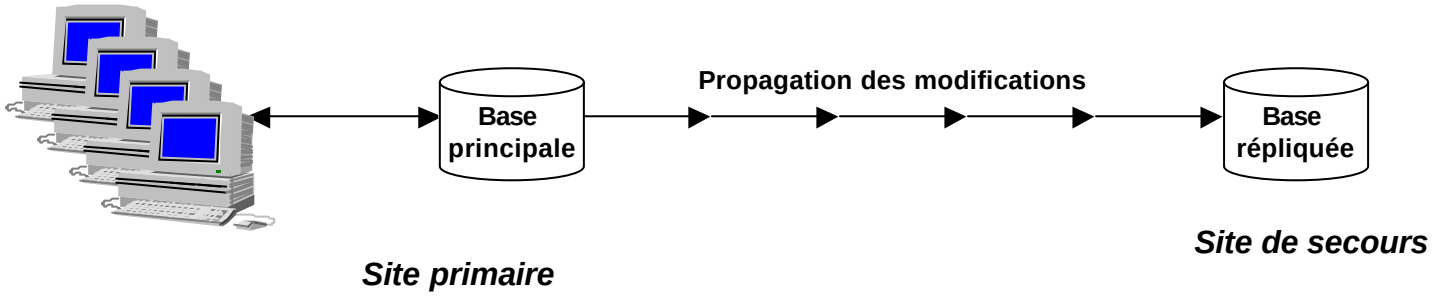
- Difficulté de conception
- Difficulté de reprise après panne
- Impact des mises à jour sur le fonctionnement des nœuds (overhead)



- Mises à jour asynchrones à partir d'un site maître
- Un seul point de référence
- Faible impact des mises à jour sur le fonctionnement des nœuds

Réplication des données (6)

- **Réplication en vue de la résistance aux défaillances (Disaster Recovery)**



  - **La réplication peut être partielle ou totale**
  - **Elle peut se fonder sur une sauvegarde totale périodique (e.g. hebdomadaire) et la copie du journal des transactions à intervalles réguliers. La base répliquée est alors régénérée à partir de la dernière sauvegarde totale et du journal**

Contenu

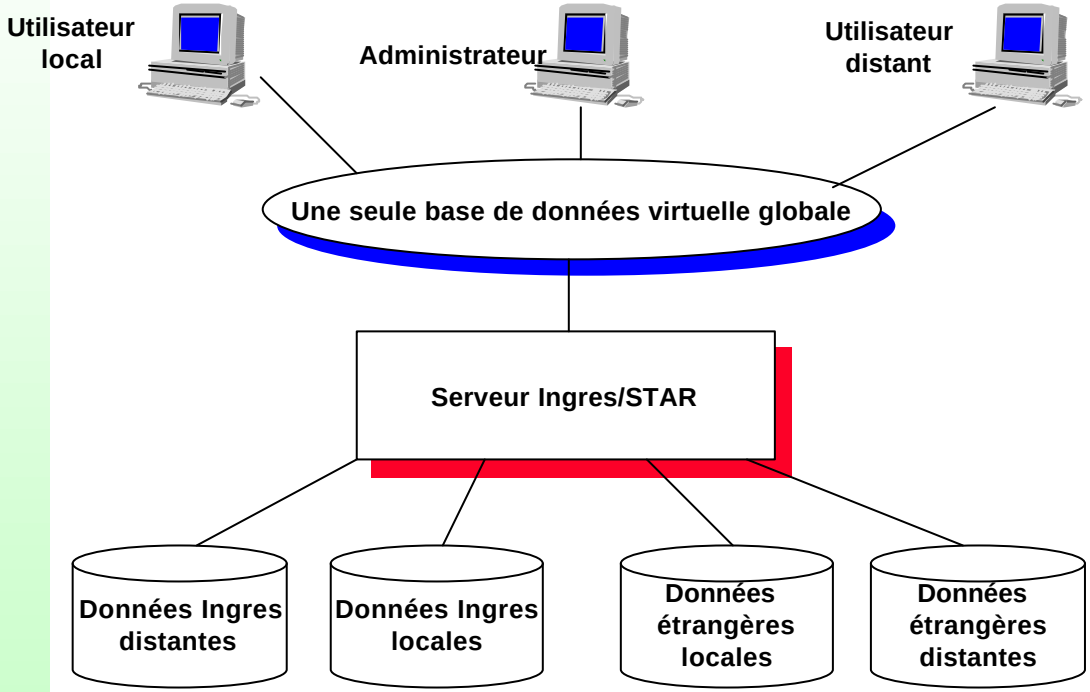
- Définitions
- Exemple de BD répartie
- Répartition des données
- Répartition - Fédération
- Fédération de BD
  - Quelques cas de conflits
- Traduction des schémas
- Architecture de référence
- Accès aux BD multiples
  - API commune
  - FAP commun avec passerelles
  - FAP commun supporté par les SGBD
- Niveaux de transparence à la localisation
  - Client/Multibases :
    - RDA, DRDA, SQL-CLI, ODBC
  - Vues réparties
  - SGBD répartis
- Quelques problèmes des BD réparties et fédérées
- Partitionnement et placement des données
  - Quelques règles pour le partitionnement
  - Expédition de données et Expédition de Fonction
  - Recherche du partitionnement idéal
- Optimisation des requêtes réparties
- Réplication dans les BD
- → Un aperçu sur les SGBD du commerce
  - Un peu d'histoire : Ingres/STAR
  - IBM DB2, Informix, Oracle, Sybase
- Évaluation des SGBD répartis
- Bibliographie

Un aperçu sur les SGBD du commerce

- **A la fin des années 80 et au début des années 90, la plupart des fournisseurs de SGBD avaient des projets voire des produits de versions réparties de leur SGBD et des capacité à fédérer des BD (homogènes et hétérogènes)**
- **Les différentes expériences faites, par les utilisateurs dans le cadre d'applications, avec ces SGBD ont montré leurs limites, en particulier dans un contexte avec mise à jour**
- **Au début des années 2000, les fournisseurs ne mettent plus en avant les aspects distribués.**
- **Les fournisseurs ont évolué vers des solutions à base de réplication des bases de données et des moyens d'accès à des bases de données « étrangères »**
- **A titre d'illustration de cette évolution, Ingres qui proposait la solution la plus élégante et la plus avancée en matière de distribution a renoncé à cette ambition. Ingres a été acquise par CA (Computer Automation). CA ne met pas l'emphase sur l'aspect distribué**
- **Nous allons passer rapidement en revue certaines des solutions proposées par différents fournisseurs de SGBD**

Un peu d'histoire : Ingres/STAR

- **Parmi les différents modèles deSBD distribués, Ingres proposait incontestablement l'un des modèles les plus élaborés**
- **Vision Ingres/STAR**



•Gestion des dictionnaires répartie :

- Dictionnaire global sur site maître
- Caches des informations dans les sites participants
- Droits d'accès répartis

• Exécution des requêtes réparties :

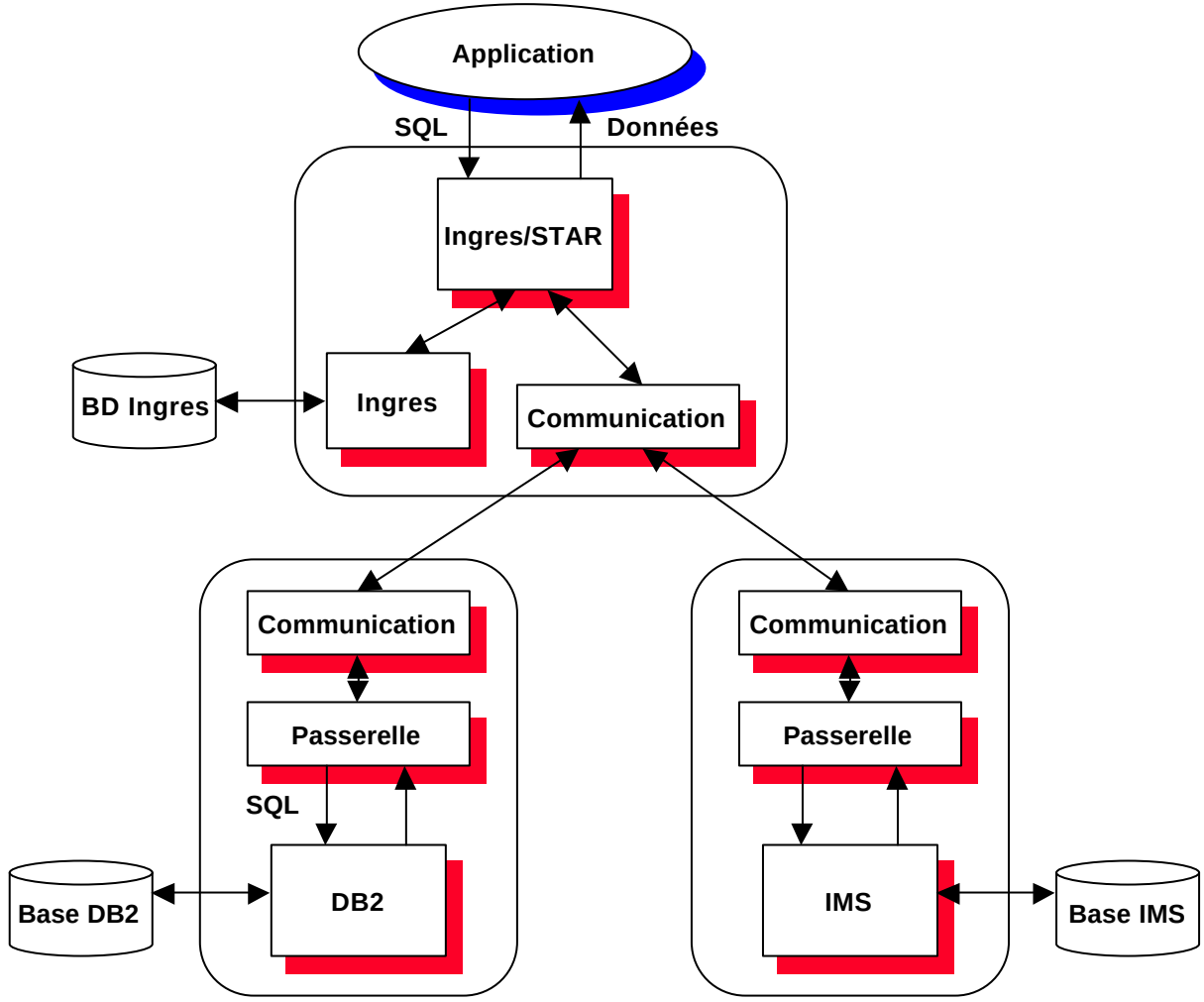
- Requêtes multi-bases en Ingres-SQL ou Open-SQL
- Définition de curseurs multi-sites

• Contrôle des transactions réparties :

- Validation à deux phases
- Détection des étreintes fatales

Un peu d'histoire : Ingres/STAR (2)

Architecture Ingres/STAR

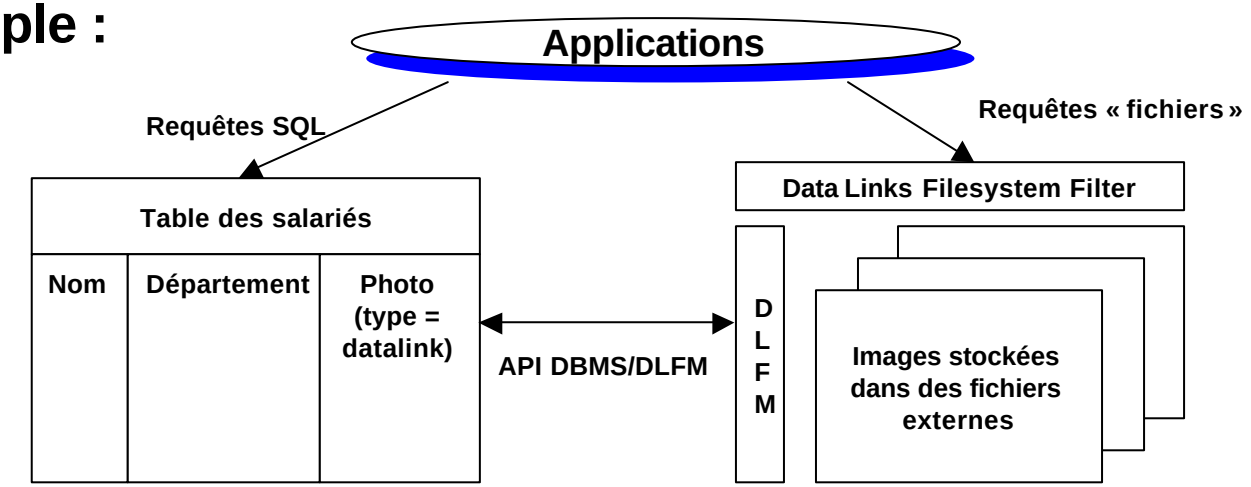


- **Trois produits (entre autres) :**
  - DataLinks
  - DataJoiner
  - DataPropagator

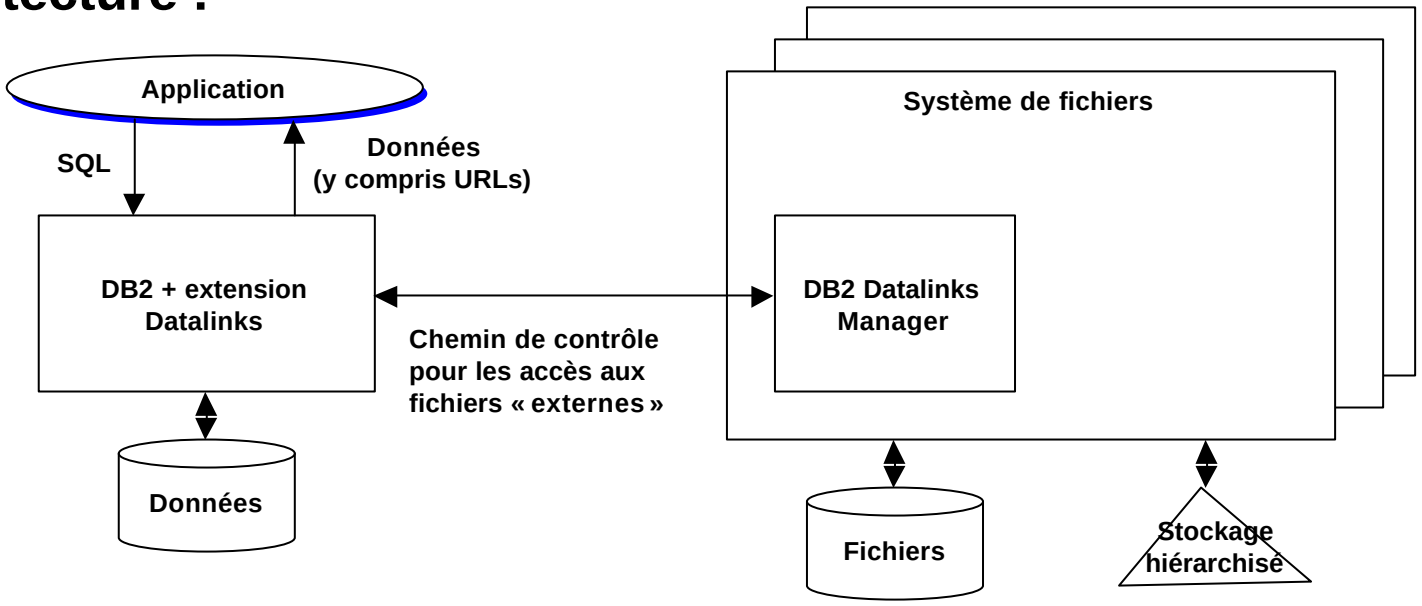
- **DataLinks :**
  - Permet à DB2 d 'accéder à des données stockées indépendamment de DB2
  - Permet différents niveaux de contrôle sur ces données externes :
    - Intégrité référentielle
    - Contrôle d'accès
    - Opérations de sauvegarde et de restauration coordonnées
    - Transactionnel distribué
  - Composants de DataLinks :
    - Nouveau type de données DATALINK (URL - Uniform Resource Locator)
    - DB2 Data Links Manager qui a deux composantes :
      - Data Links File Manager (DLFM)
      - Data Links Filesystem Filter (DLFF)
    - API DBMS/DLFM pour dialoguer avec les Data Links File Managers

IBM DB2 - DataLinks (2)

Exemple :



Architecture :

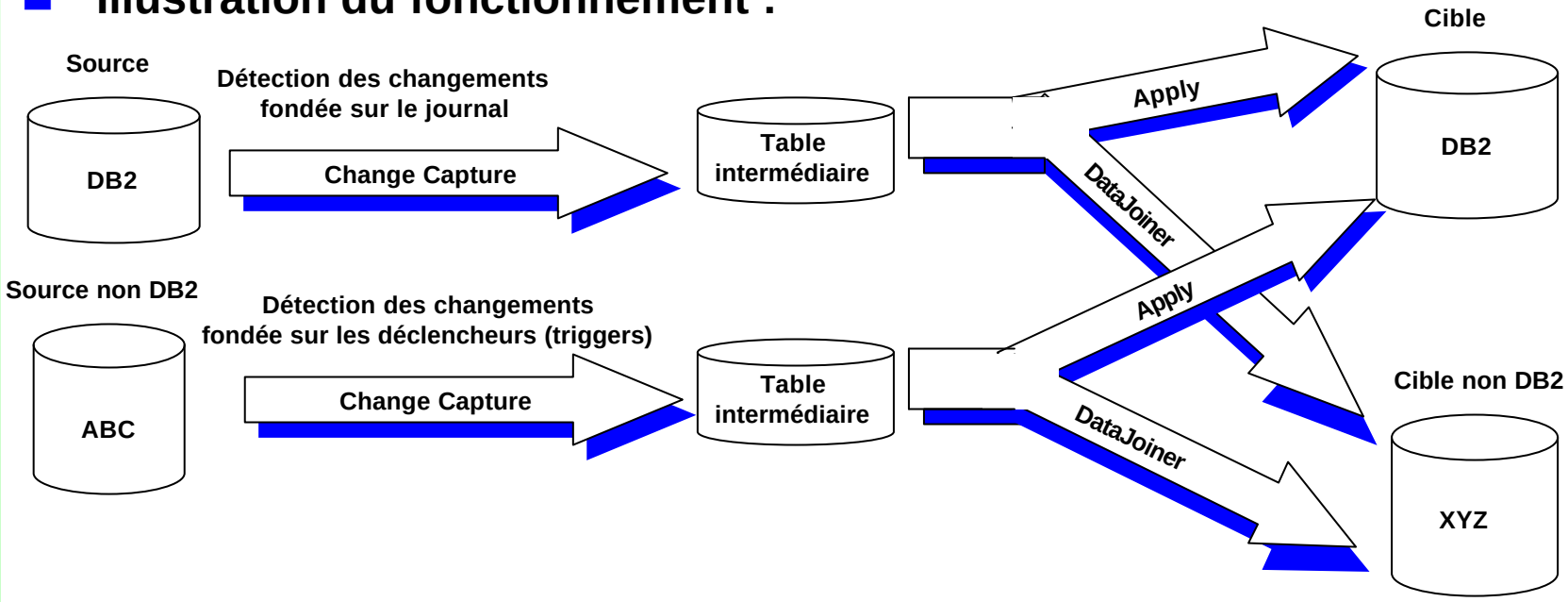


IBM DB2 - DataJoiner (3)

- **Permet l'accès aux données gérées par des SGBD différents (relationnels ou non)**
- **Supporte les fonctions :**
  - **SQL DB2**
  - **Validation à deux phases (XA)**
  - **Procédures stockées**
  - **Définition globale des données (DDL)**
  - **Accès ODBC, JDBC, SQL-CLI**
- **Fonctionne en relation avec les mécanismes de réplication**

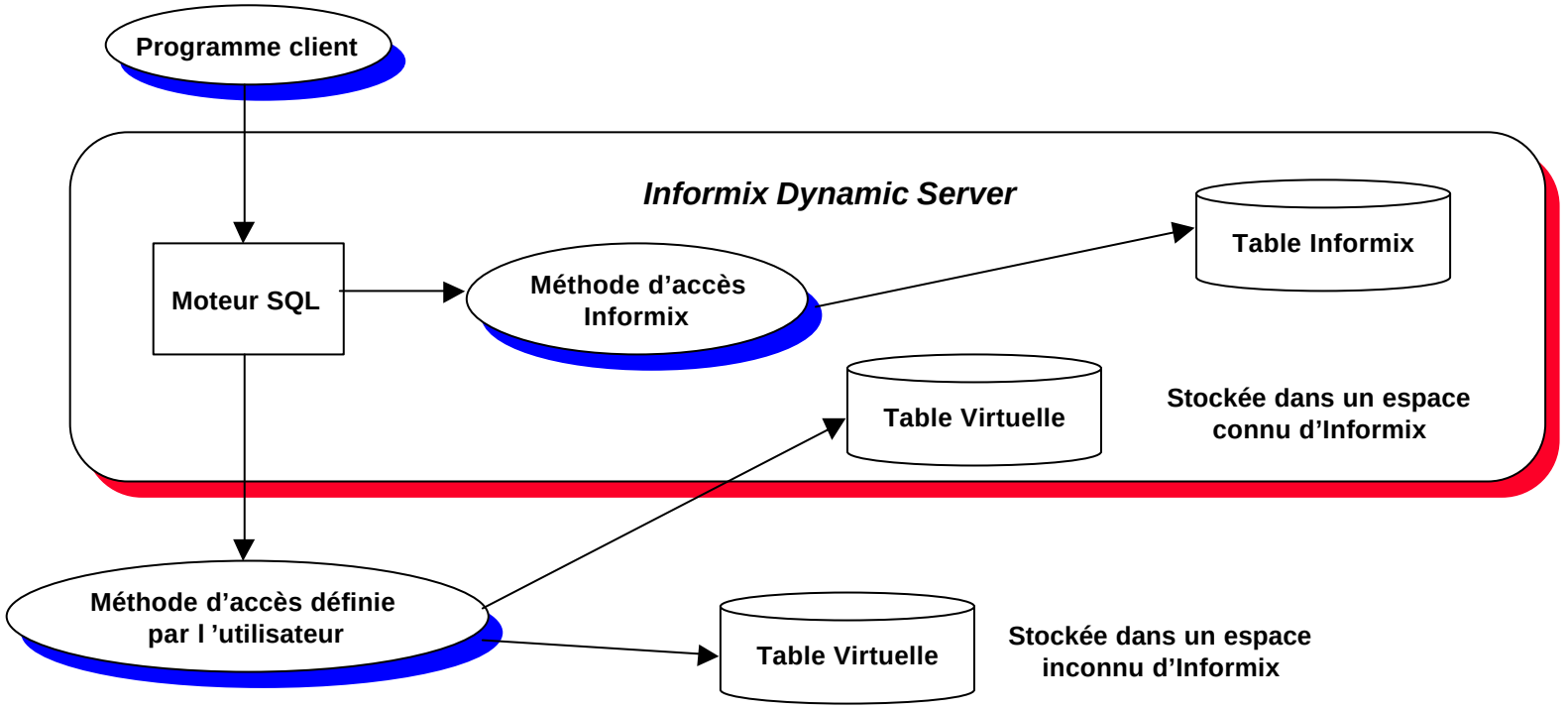
IBM DB2 - DataPropagator

- **DB2 Datapropagator assure la mise à jour de bases de données distribuées (d'une « source » vers des « cibles »)**
- **Les composants de DataPropagator sont :**
  - **Change Capture : détection des changements dans la base « source » et stockage dans des tables intermédiaires (staging tables)**
  - **Apply : prend les données stockées dans les tables intermédiaires et applique les changements aux bases des données « cibles »**
  - **Administration : fonctions permettant de spécifier et de contrôler le processus de réplication**
- **Illustration du fonctionnement :**



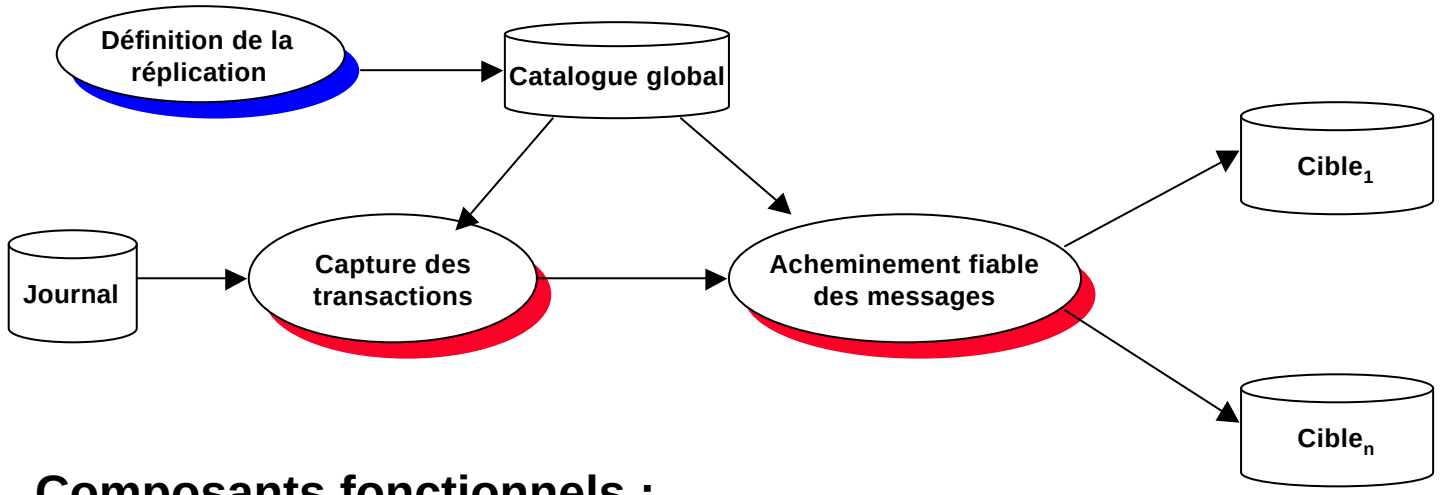
Informix - Virtual Table Interface

- **Virtual Table Interface**
  - **Capacité de définir des méthodes d'accès en complément de celles offertes par Informix**
  - **Ces méthodes d'accès peuvent opérer soit :**
    - **sur des données gérées par Informix**
    - **sur des données non gérées par Informix (pas de service transactionnel ni de sauvegarde/restauration fournit par Informix pour ces données)**



Informix - Enterprise Replication

Architecture



Composants fonctionnels :

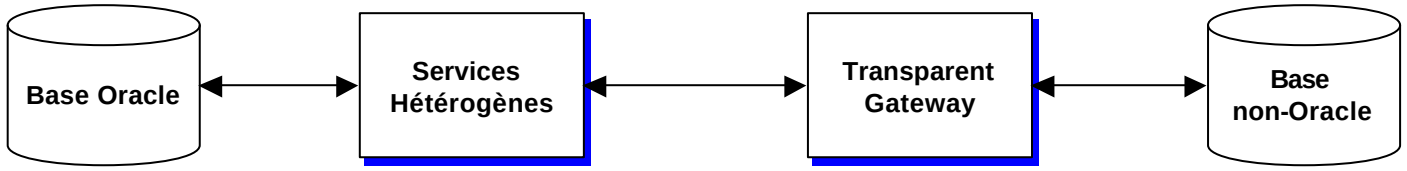
- Configuration et contrôle
- Capture des transactions (via le journal)
- Acheminement des informations (type MOM sécurisé)
- Prise en compte des mises à jour sur site distant (mises à jour suivant une logique transactionnelle)
- Résolution de conflits. Possibilités :
  - Priorité à la dernière modification en date
  - Stratégie définie par programme
  - Ignorance

Oracle - Transparent Gateways

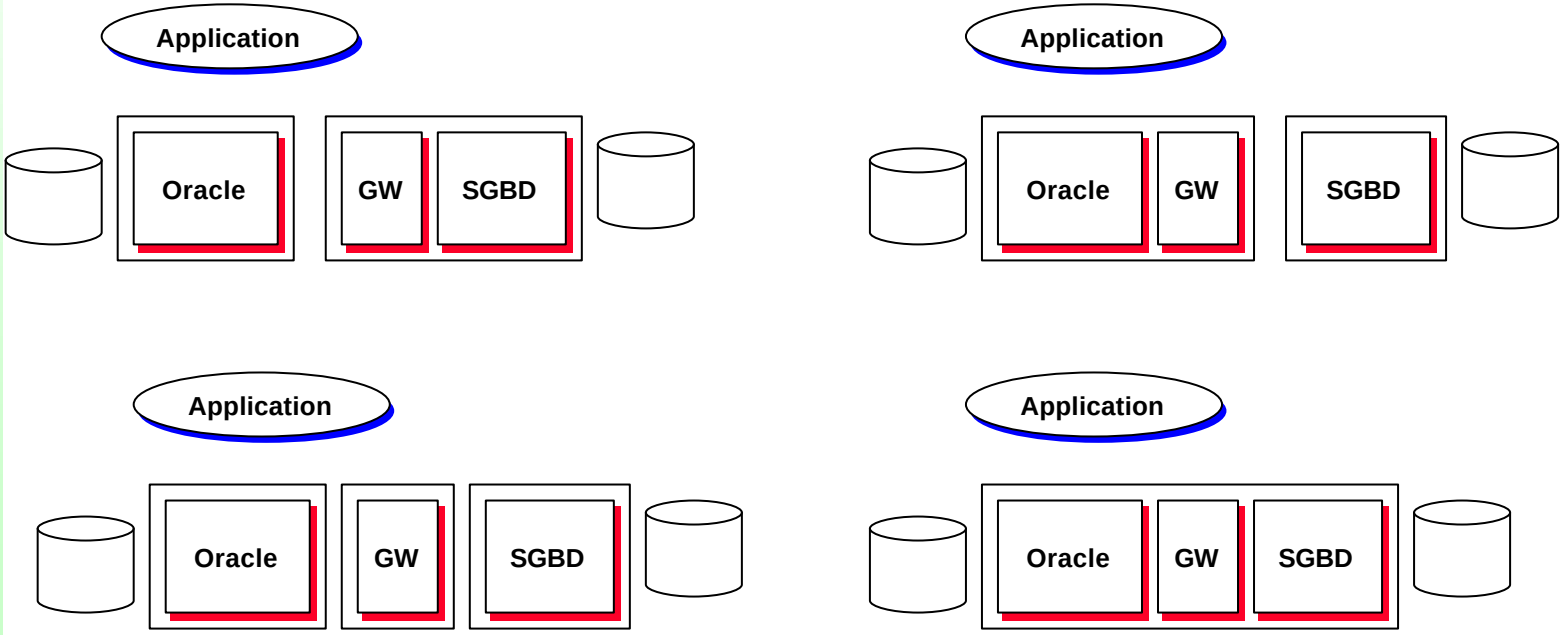
- **Solution Transparent Gateways**
  - **Permet l'accès à de nombreux (40?) gestionnaires de données**
  - **Constituée de deux composants :**
    - **Services hétérogènes (Heterogeneous Services)**
      - **Service transactionnel (validation à deux phases)**
      - **Service SQL**
        - **Traduction SQL Oracle vers SQL non Oracle**
        - **Traduction de dictionnaire de données**
      - **Service procédural (exécution de procédures stockées)**
    - **Transparent Gateway**
      - **SQL and Data Dictionary Translation Information : fournit aux services hétérogènes les informations nécessaires à la traduction**
      - **Traduction des types de données**
      - **Assure la connexion à des systèmes non-Oracle**
      - **Plusieurs possibilités concernant la localisation du Gateway**

Oracle - Transparent Gateways (2)

Architecture générale :

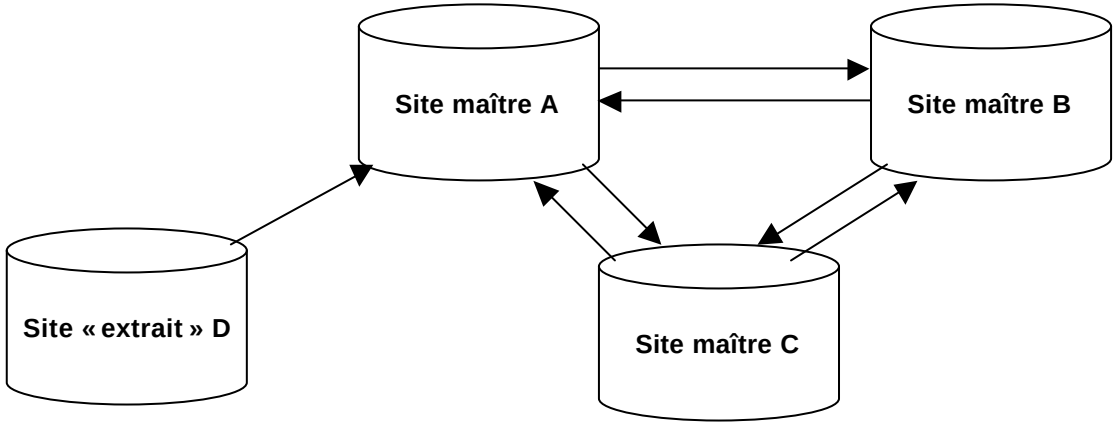


Localisation du Gateway



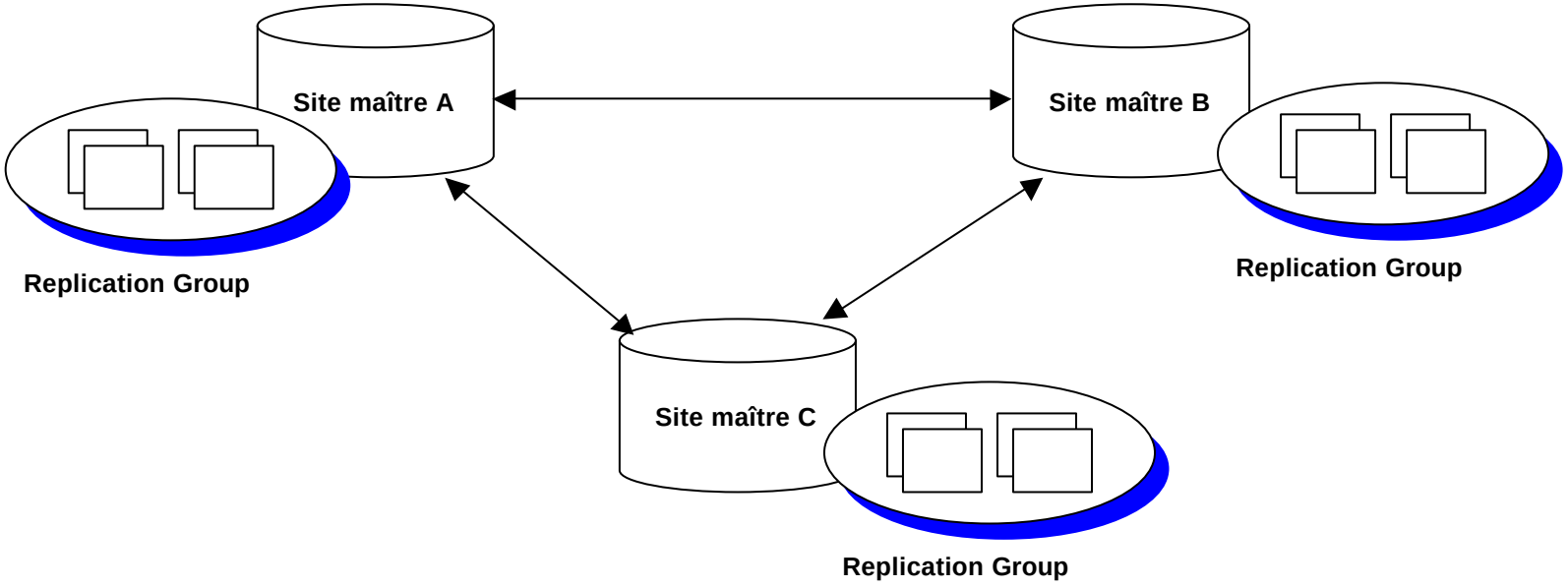
Oracle Database Replication

- **Notions de base :**
  - **Replication Object : objet existant à de multiples exemplaires**
  - **Replication Groups : regroupement logique d'objets répliqués en vue de faciliter l'administration de la réplication**
  - **Replication sites :**
    - **Master Site : contient la copie complète des objets d'un groupe de réplication**
    - **Snapshot Site : supporte des extraits en lecture seule d'un « replication group » ou un extrait susceptible de mise à jour**



Oracle Database Replication (2)

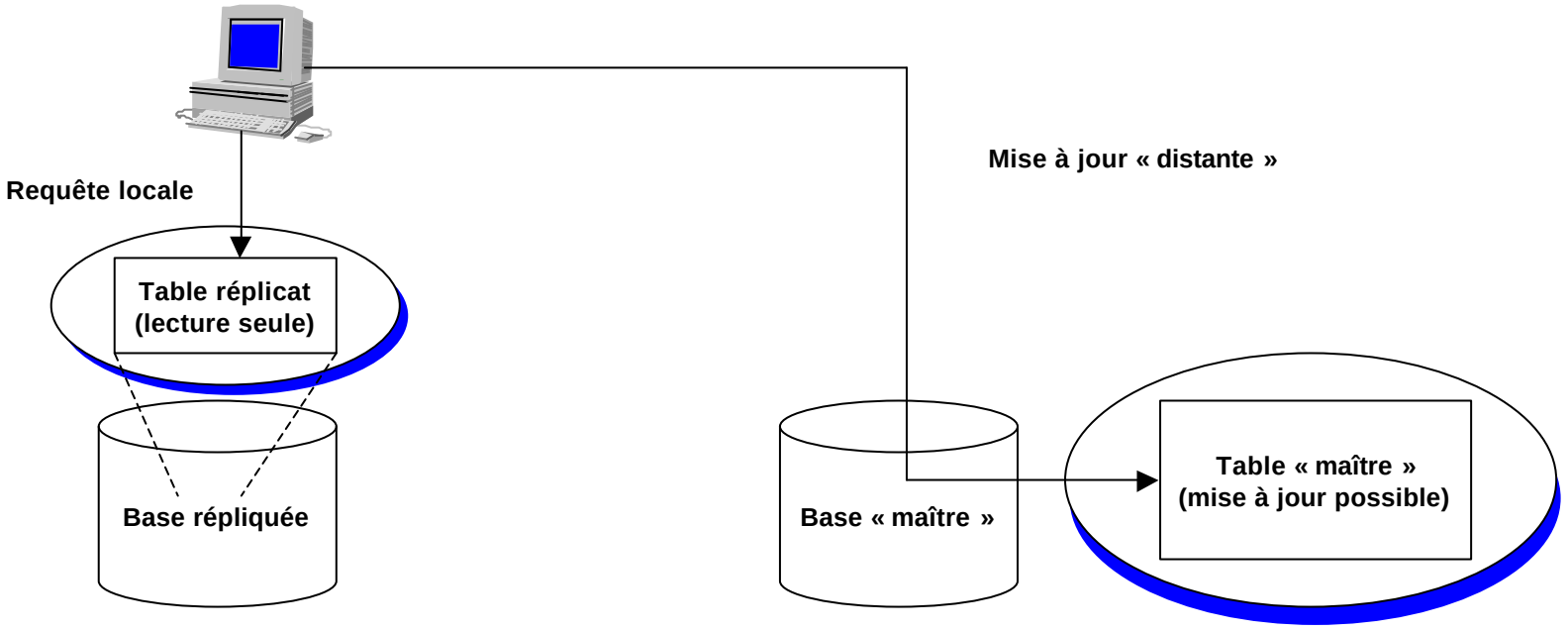
- **Multimaster Replication : Plusieurs sites, dans un mode d'égal à égal (peer-to-peer), gèrent des objets répliqués. Utilisation :**
  - **Recouvrement en cas de défaillance**
  - **Distribution de la charge de travail**



- **Difficulté : synchronisation**

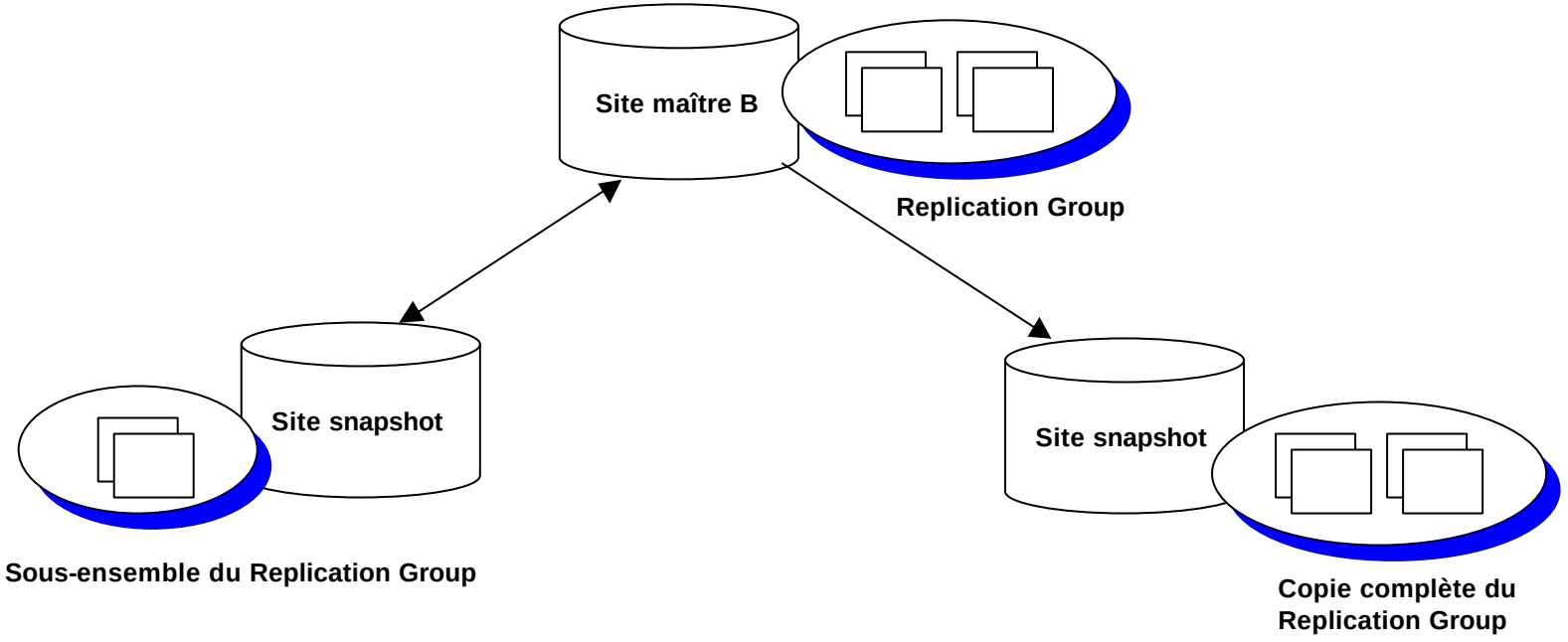
Oracle Database Replication (3)

- **Snapshot Replication :**
  - **Snapshots en lecture seule (Read-Only Snapshots) :**
    - **Réplication complète ou partielle d'une table « maître ». Avantages :**
      - **Les tables « maître » n'appartiennent pas nécessairement au même groupe de réplication**
      - **Les snapshots peuvent résulter de la composition de plusieurs BD)**
      - **Exécution locale de requête (allègement de la charge du site maître)**
    - **En lecture seule ou avec possibilité de mise à jour de la fonction de mise à jour distante (interaction avec le site détenant la base maître)**



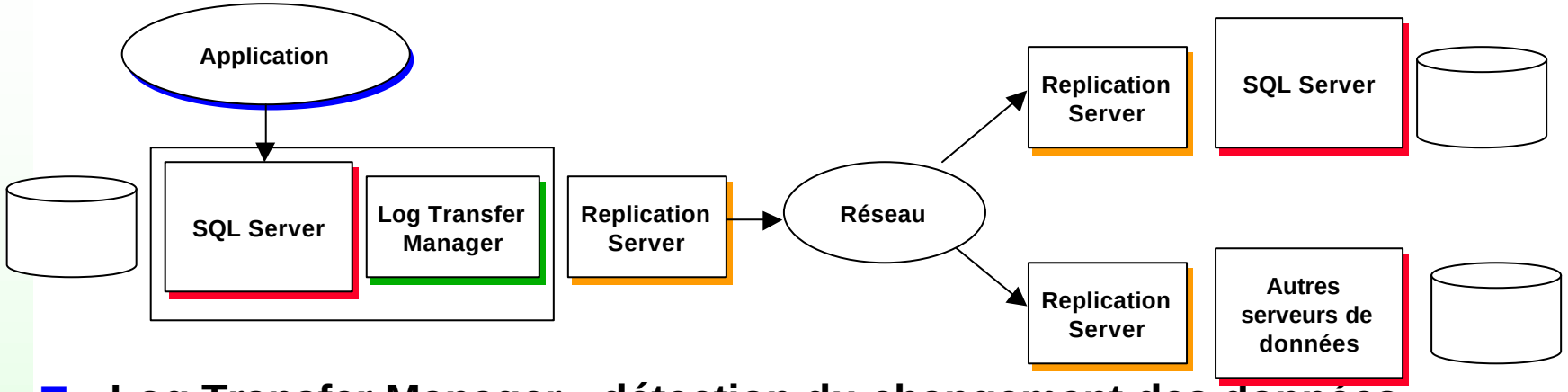
Oracle Database Replication (4)

- **Snapshots avec mise à jour (Updateable Snapshots) :**
  - **Les snapshops supportant la mise à jour ne dépendent que d'une seule table maître et sont mis à jour (refresh) de façon incrémentale ou immédiate**
  - **Les mises à jour sont propagées depuis le snapshot vers la base maître. Les mises à jour peuvent être répercutée vers d'autres sites**
  - **Les snapshots supportant la mise à jour sont remis à jour**
  - **Avantages :**
    - **Permet aux utilisateurs de consulter et de mettre à jour les données (locale) même en cas de déconnexion avec le site maître**
    - **Réduction possible du volume de la base répliquée**



Sybase - Replication Server

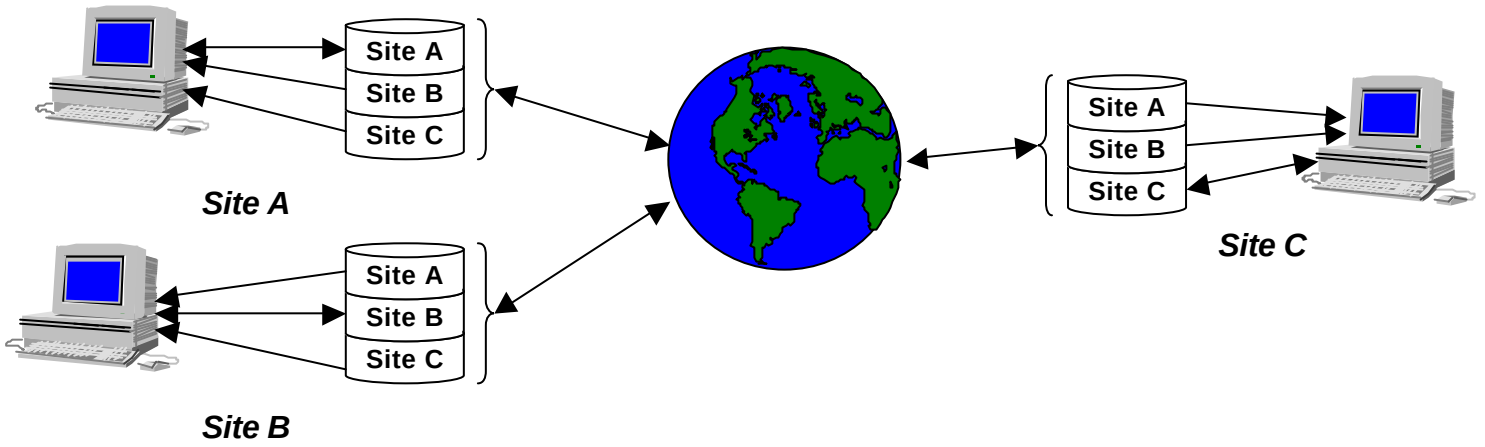
- **Composants du Replication Server**



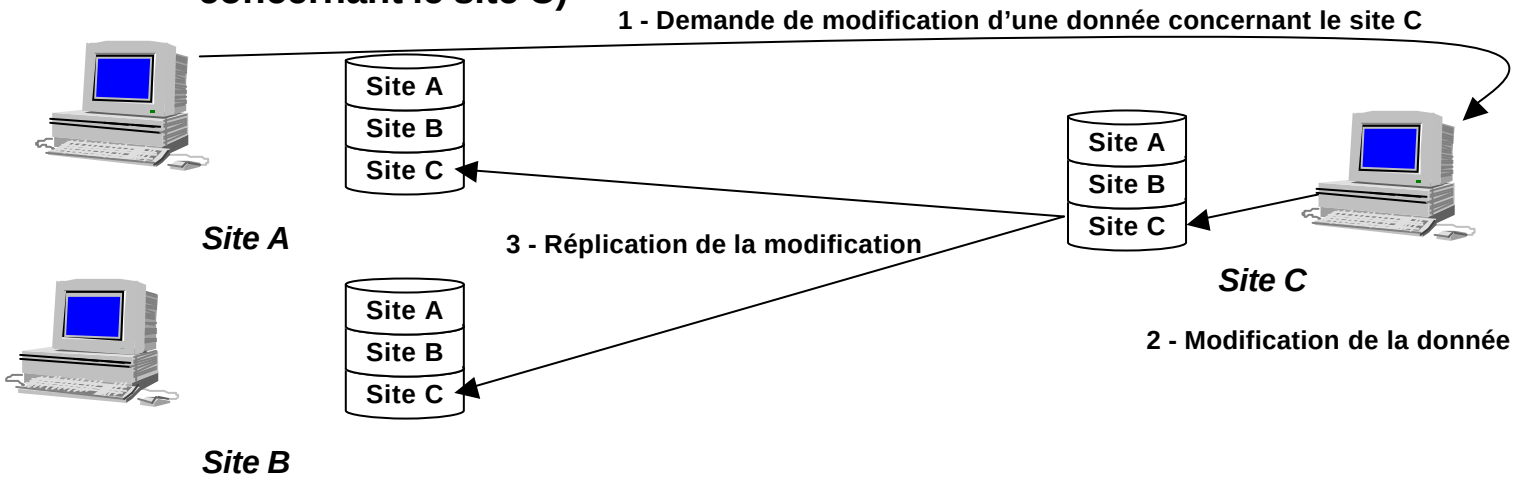
- **Log Transfer Manager : détection du changement des données**
  - **Le LTM (Log Transfer manager) est un thread qui surveille le journal de la base de données à laquelle il est associé**
  - **Le LTM traduit les modifications qu'il détecte sous une forme indépendante compréhensible par le Replication Server : le LTL (Log Transfer Language) qui est une composante du RCL (Replication Control Language)**
  - **Cette interface est publique (ce qui permet d'implémenter le développement des composants nécessaires pour des environnements hétérogènes)**
  - **Utilisation de files d'attente de messages (MOM ou Message Oriented Middleware) pour palier l'indisponibilité des systèmes**

Sybase - Replication Server (2)

- **Quelques exemples :**
  - **Partage d'informations entre sites**
    - **Une règle simple : un seul site fait les mises à jour des données qui le concernent**



    - **Exemple de mise à jour (le site A demande la mise à jour d'une donnée concernant le site C)**



Contenu

- Définitions
- Exemple de BD répartie
- Répartition des données
- Répartition - Fédération
- Fédération de BD
  - Quelques cas de conflits
- Traduction des schémas
- Architecture de référence
- Accès aux BD multiples
  - API commune
  - FAP commun avec passerelles
  - FAP commun supporté par les SGBD
- Niveaux de transparence à la localisation
  - Client/Multibases :
    - RDA, DRDA, SQL-CLI, ODBC
  - Vues réparties
  - SGBD répartis
- Quelques problèmes des BD réparties et fédérées
- Partitionnement et placement des données
  - Quelques règles pour le partitionnement
  - Expédition de données et Expédition de Fonction
  - Recherche du partitionnement idéal
- Optimisation des requêtes réparties
- Réplication dans les BD
- Un aperçu sur les SGBD du commerce
  - Un peu d'histoire : Ingres/STAR
  - IBM DB2, Informix, Oracle, Sybase
- ➔ Évaluation des SGBD répartis
- Bibliographie

Évaluation des SGBD répartis

- **Les 12 + 1 critères de C Date**
  - **0 - Transparence pour l'utilisateur**
  - **1 - Autonomie de chaque site**
  - **2 - Absence de site privilégié**
  - **3 - Continuité de service**
  - **4 - Indépendance vis-à-vis de la localisation**
  - **5 - Indépendance vis-à-vis de la fragmentation**
  - **6 - Indépendance vis-à-vis de la réplication**
  - **7 - Traitement réparti des requêtes**
  - **8 - Gestion répartie des transactions**
  - **9 - Indépendance vis-à-vis du matériel**
  - **10 - Indépendance vis-à-vis du système d'exploitation**
  - **11 - Indépendance vis-à-vis du réseau**
  - **12 - Indépendance vis-à-vis du SGBD**

Bibliographie

- Claude Chrisment, Geneviève Pujolle, Gilles Zurfluh « Bases de données réparties » Les Techniques de l'Ingénieur
- Georges et Olivier Gardarin « Le Client - Serveur » Eyrolles
- Robert Orfali, Dan Harkey, Jerry Edwards « Client/Serveur - Guide de survie » John Wiley, Thomson Publishing
- Serge Miranda, Anne Ruols « Client-Serveur » Eyrolles
- Polycopiés des cours CNAM d'Intégration des Systèmes Client/Serveur des années précédentes par :
  - Béatrice Finance
  - Jean-Pierre Meinadier
  - Jean-Marc Saglio, Yann Viemont
- Sites des principaux fournisseurs de SGBD :
  - http://www.ibm.com
  - http://www.informix.com
  - http://www.oracle.com
  - http://www.sybase.com