

Chapitre 2: Les bases de données réparties

Module: Bases de données avancées
Master STIC (Semestre 1)
Département Informatique, Université de Guelma
202**1**/202**2**

Plan

Partie 1: Bases de Données Réparties

Partie 2: Bases de données répliquées

Partie 1:

Bases de données réparties

Bases de données centralisées vs. distribuées

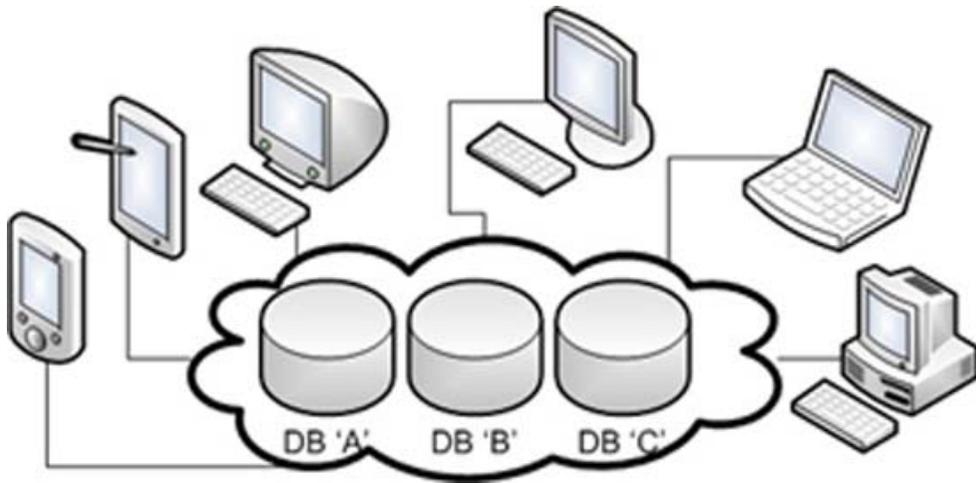


Figure 2: Bases de données distribuées.

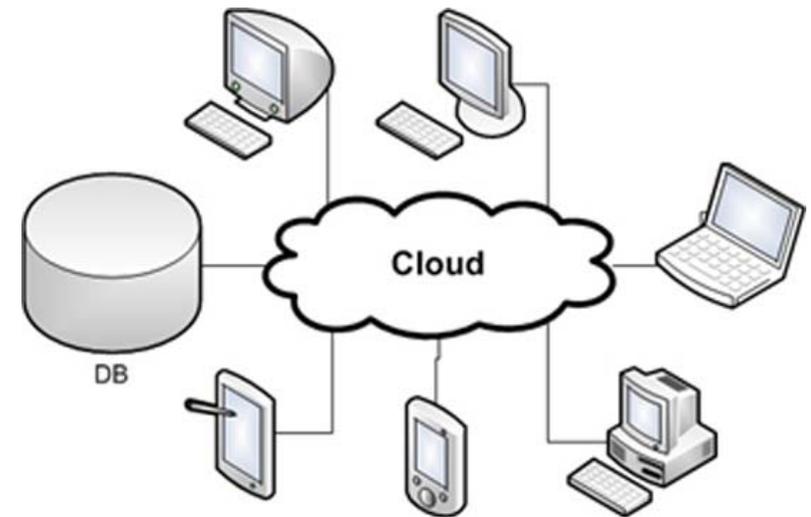


Figure 1: Bases de données centralisées.

Définitions

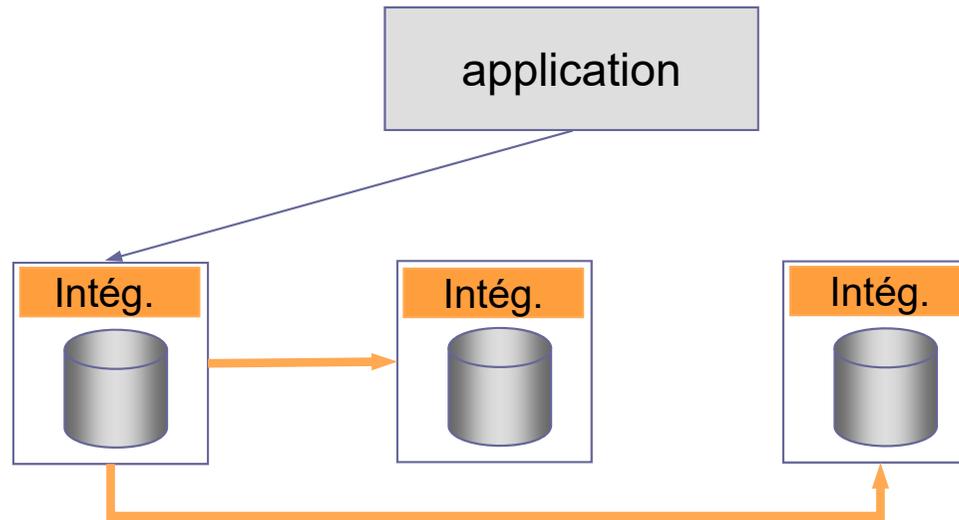
➤ Base de données répartie (BDR)

- Ensemble de bases localisées sur différents sites, perçues par l'utilisateur comme une base unique

➤ Niveaux de schémas

- Chaque base possède son **schéma local**
- Le schéma de la base répartie constitue le **schéma global**
 - Il assure la transparence à la localisation des données
 - Il permet des recompositions de tables par union/jointure
 - il n'y a pas de base globale physique correspondant à ce schéma

Fonctions d'un SGBD réparti



- Rend la répartition (ou distribution) *transparente*
 - dictionnaire des données réparties
 - traitement des requêtes réparties
 - gestion de transactions réparties
 - gestion de la confidentialité

Evaluation de l'approche BDR

➤ Avantages

- extensibilité
- partage des données réparties
- performances
- disponibilité des données

➤ Inconvénients

- administration complexe
- distribution du contrôle

Constituants du schéma global

➤ schéma conceptuel global

- donne la description globale et unifiée de toutes les données de la BDR (e.g., des relations globales)
- indépendance à la répartition

➤ schéma de placement

- règles de correspondance avec les données locales
- indépendance à la localisation, la fragmentation et la duplication

➤ Le schéma global fait partie du dictionnaire de la BDR et peut être conçu comme une BDR (dupliqué ou fragmenté)

Exemple de schéma global

➤ Schéma conceptuel global

Client(nclient, nom, ville)

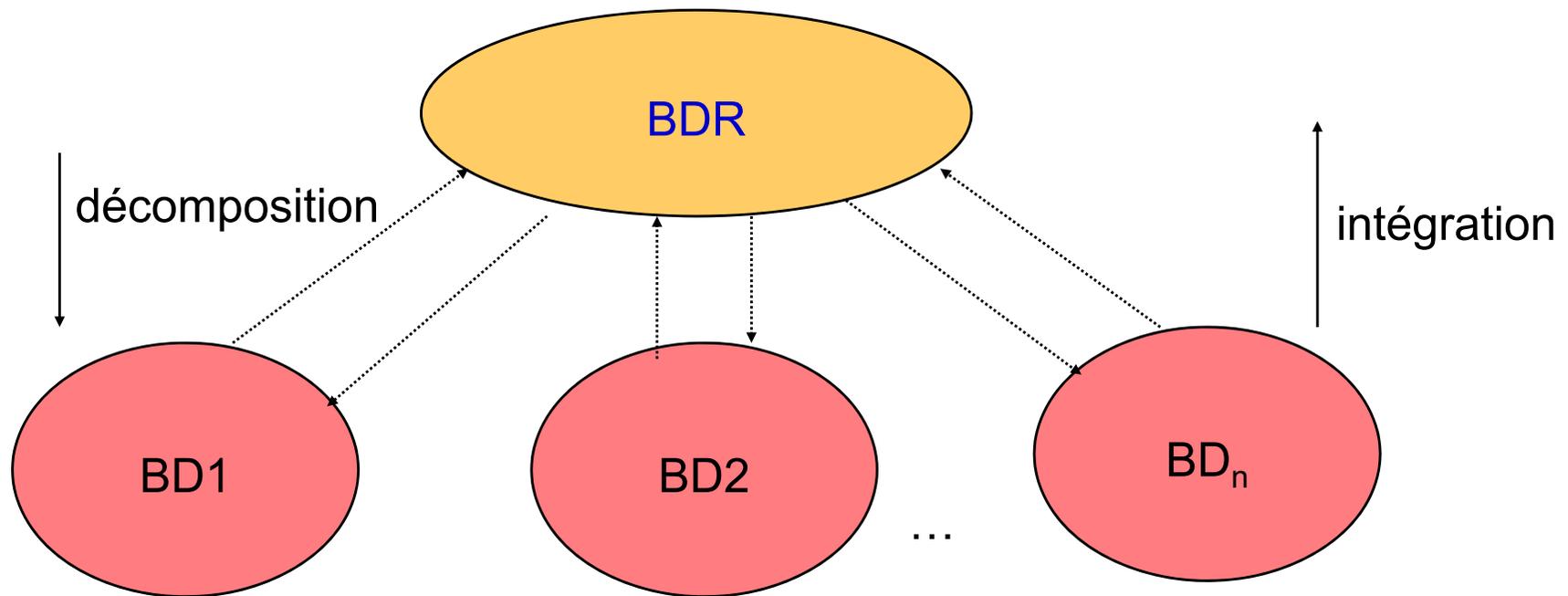
Cde (ncde, nclient, produit, qté)

➤ Schéma de placement

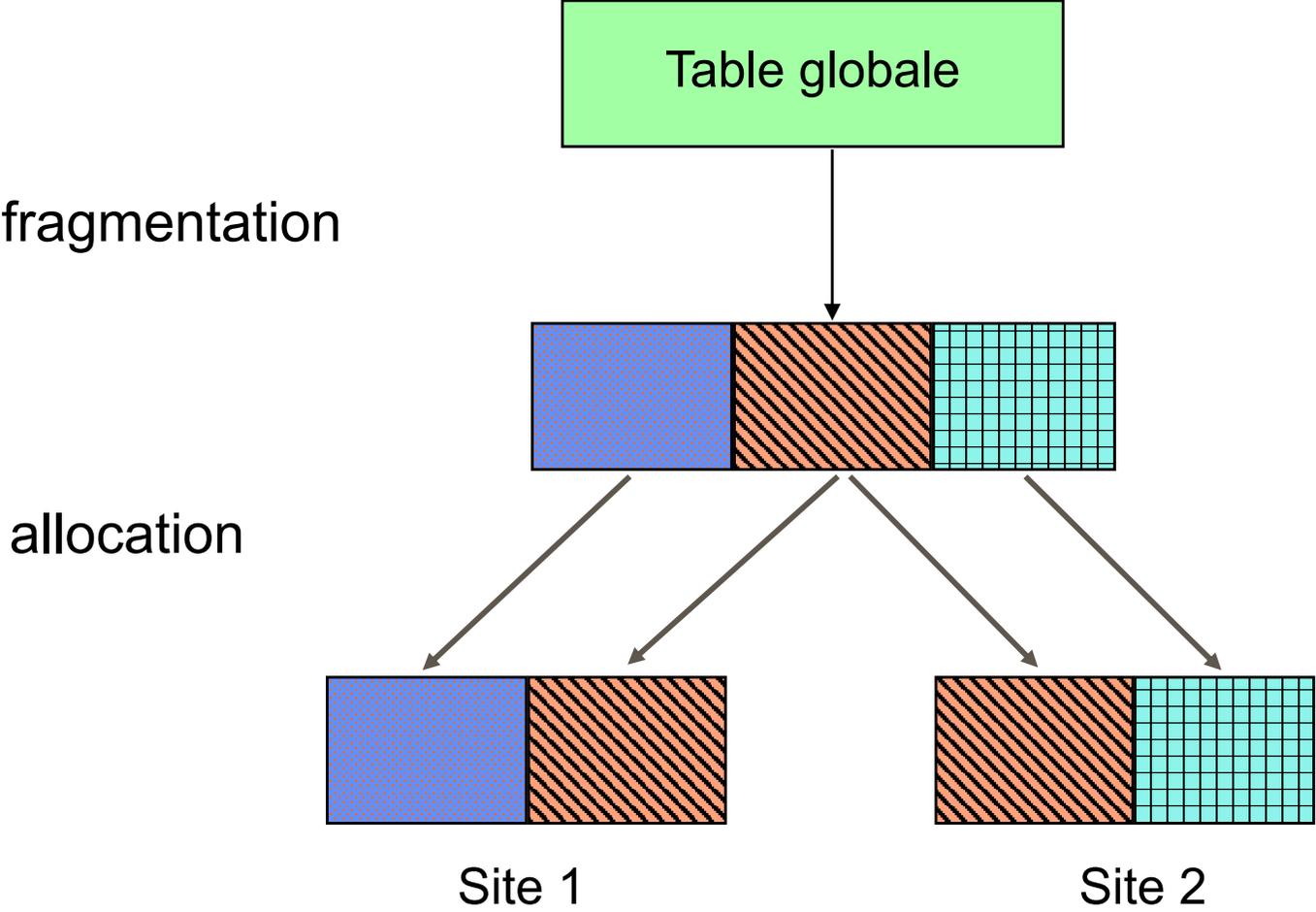
Client = Client1 @ Site1 U Client2 @ Site2

Cde = Cde @ Site3

Conception des bases réparties



Conception par décomposition



Objectifs de la décomposition

➤ fragmentation

- trois types : horizontale, verticale, mixte
- performances en favorisant les accès locaux
- équilibrer la charge de travail entre les sites (parallélisme)

➤ duplication (ou réplication)

- favoriser les accès locaux
- augmenter la disponibilité des données

Fragmentation horizontale

➤ Fragments définis par sélection

- Client1 = Client where ville = "Paris"
- Client2 = Client where ville ≠ "Paris"

Reconstruction

Client = Client1 U Client2

Client

nclient	nom	ville
C 1	Dupont	Paris
C 2	Martin	Lyon
C 3	Martin	Paris
C 4	Smith	Lille

Client1

nclient	nom	ville
C 1	Dupont	Paris
C 3	Martin	Paris

Client2

nclient	nom	ville
C 2	Martin	Lyon
C 4	Smith	Lille

Fragmentation verticale

➤ Fragments définis par projection

- Cde1 = Cde (ncde, nclient)
- Cde2 = Cde (ncde, produit, qté)

➤ Reconstruction

- Cde = [ncde, nclient, produit, qté] where Cde1.ncde = Cde2.ncde

Cde

ncde	nclient	produit	qté
D 1	C 1	P 1	10
D 2	C 1	P 2	20
D 3	C 2	P 3	5
D 4	C 4	P 4	10

Cde1

ncde	nclient
D 1	C 1
D 2	C 1
D 3	C 2
D 4	C 4

Cde2

ncde	produit	qté
D 1	P 1	10
D 2	P 2	20
D 3	P 3	5
D 4	P 4	10

Allocation des fragments aux sites

➤ Non-dupliquée

- partitionnée : chaque fragment réside sur un seul site

➤ Dupliquée

- chaque fragment sur un ou plusieurs sites
- maintien de la cohérence des copies multiples

➤ Règle intuitive:

- si le ratio est $[\text{lectures}/\text{màj}] > 1$, la duplication est avantageuse

Exemple d'allocation de fragments

Client1

nclient	nom	ville
C 1	Dupont	Paris
C 3	Martin	Paris

Client2

nclient	nom	ville
C 2	Martin	Lyon
C 4	Smith	Lille

Cde1

ncde	client	produit	qté
D 1	C 1	P 1	10
D 2	C 1	P 2	20

Site 1

Cde2

ncde	client	produit	qté
D 3	C 2	P 3	5
D 4	C 4	P 4	10

Site 2

Evaluation de requêtes réparties

Requête sur tables globales



Schéma
de fragmentation



A light blue oval with a black border containing the text "Schéma de fragmentation".

Requête sur fragments

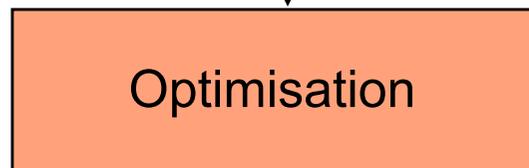


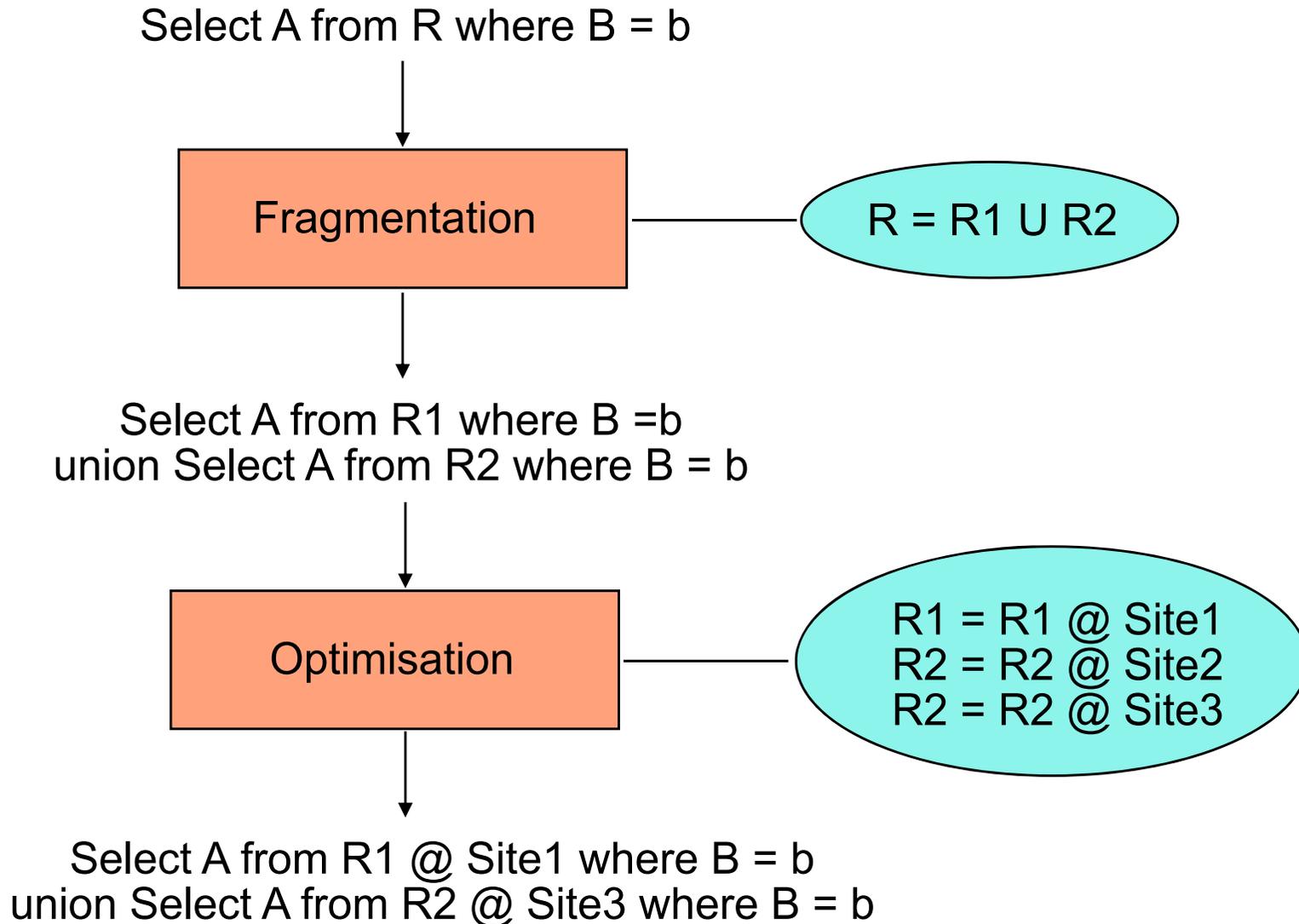
Schéma
d'allocation



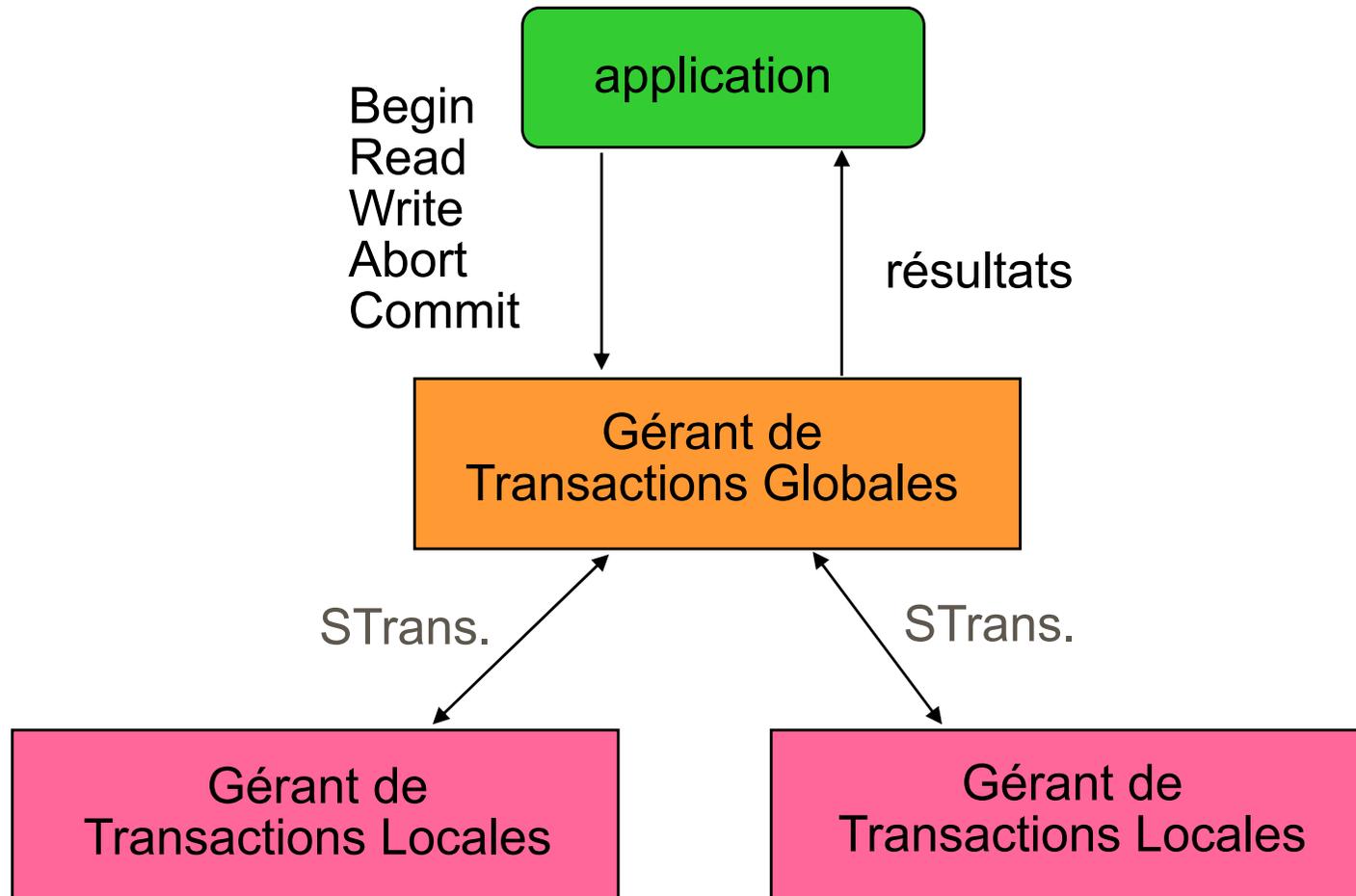
A light blue oval with a black border containing the text "Schéma d'allocation".

Plan d'exécution réparti

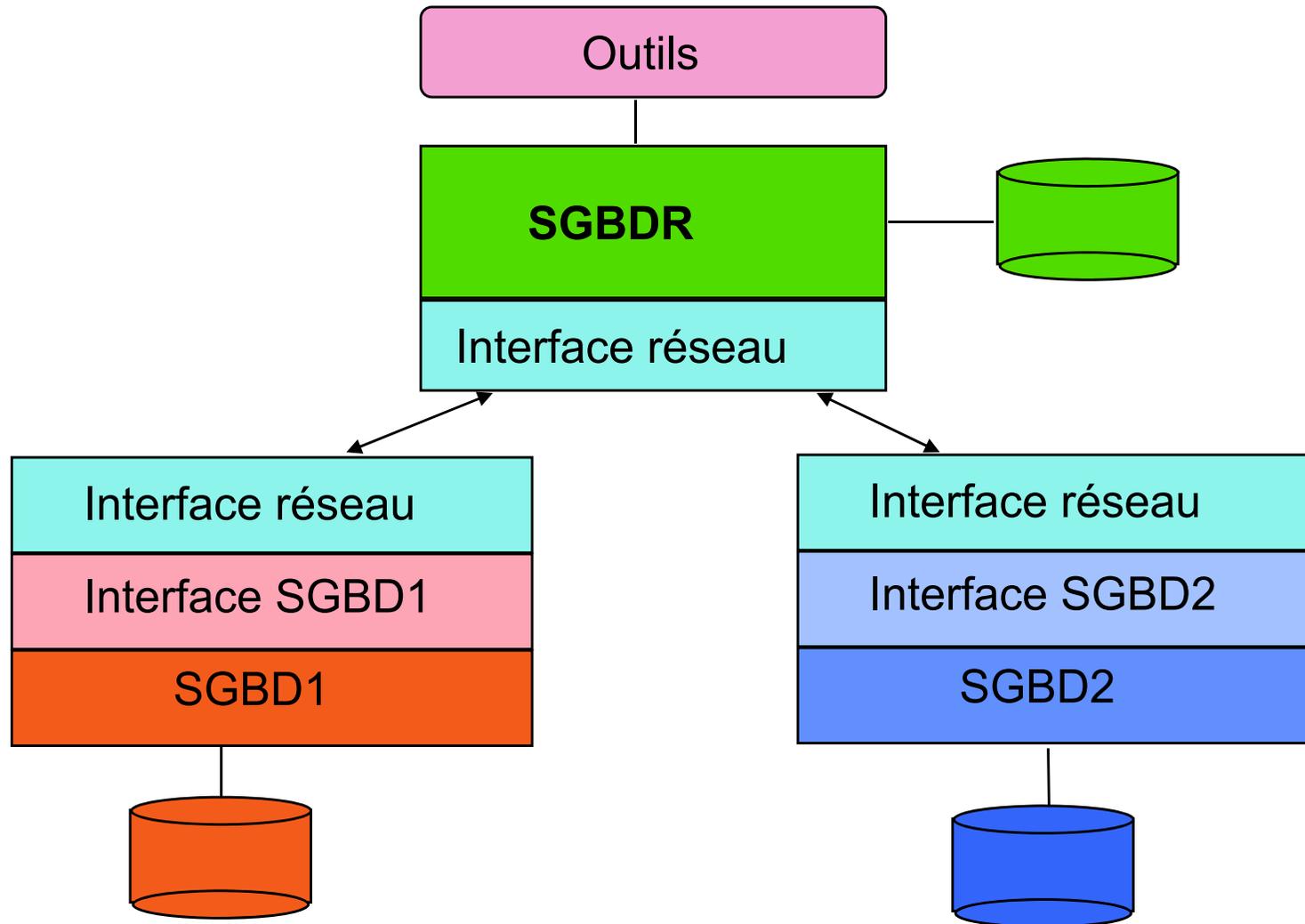
Exemple d'évaluation simple



Notion de Transaction Répartie



SGBD réparti hétérogène



Oracle

➤ SGBD Oracle

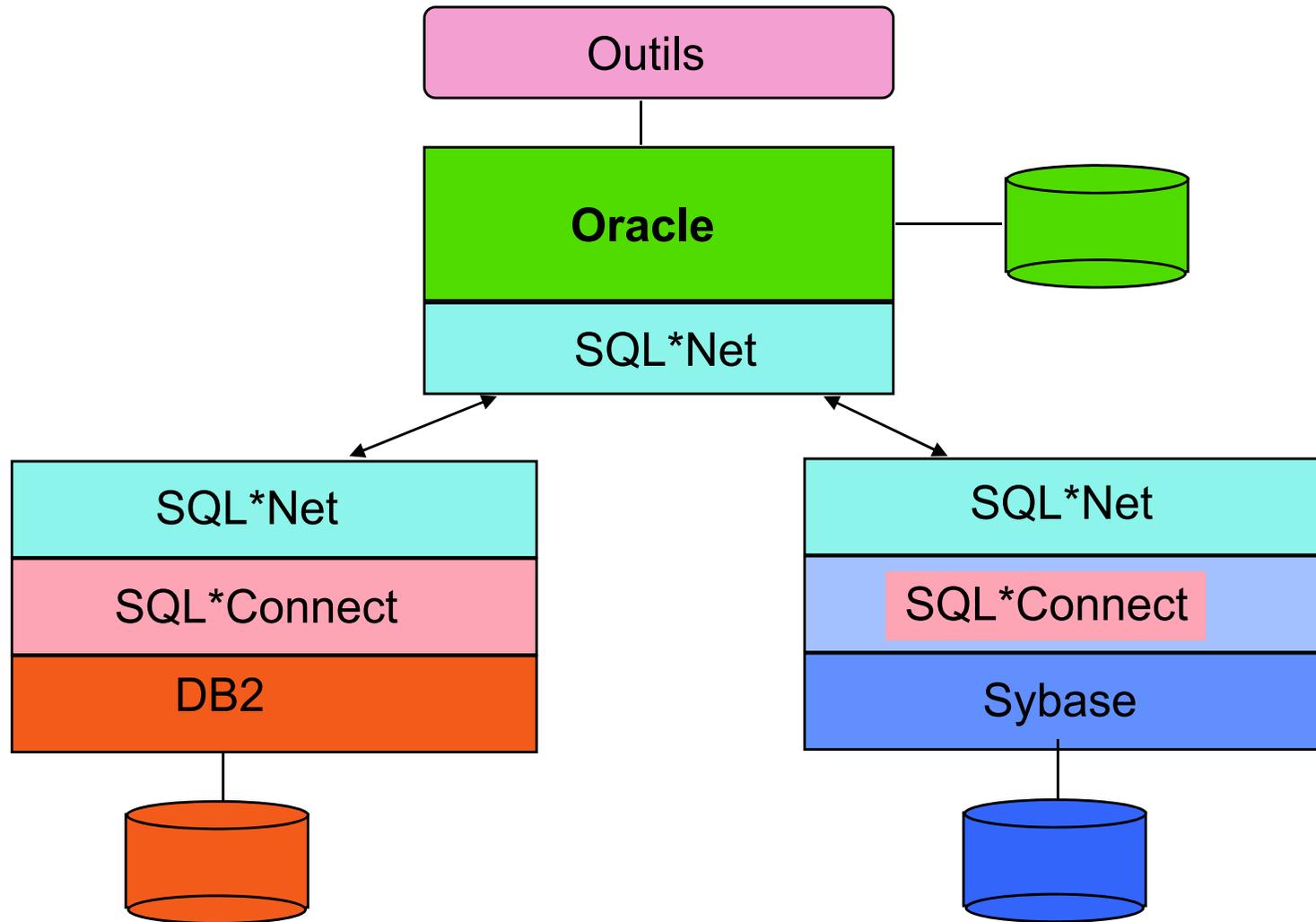
- gestion du dictionnaire de la BDR

➤ SQL*Net

- transparence au réseau
- connexion client-serveur, login à distance automatique
- évaluation de requêtes réparties
- validation en deux étapes et réplication

➤ SQL*Connect : passerelle vers les bases non-Oracle

Oracle/Star : architecture



Partie 2:

Bases de données répliquées

Définitions

➤ Réplica ou copie de données

- Fragment horizontal ou vertical d'une table stockée dans une base de données qui est copiée et transféré vers une autre base de données
- L'original est appelé la copie primaire et les copies sont appelées copies secondaires

➤ Transparence

- Les applications clientes croient à l'existence d'une seule copie des données qu'ils manipulent

Les avantages de la réplication

➤ Amélioration des performances

- lecture de la copie la plus proche

➤ Amélioration de la disponibilité

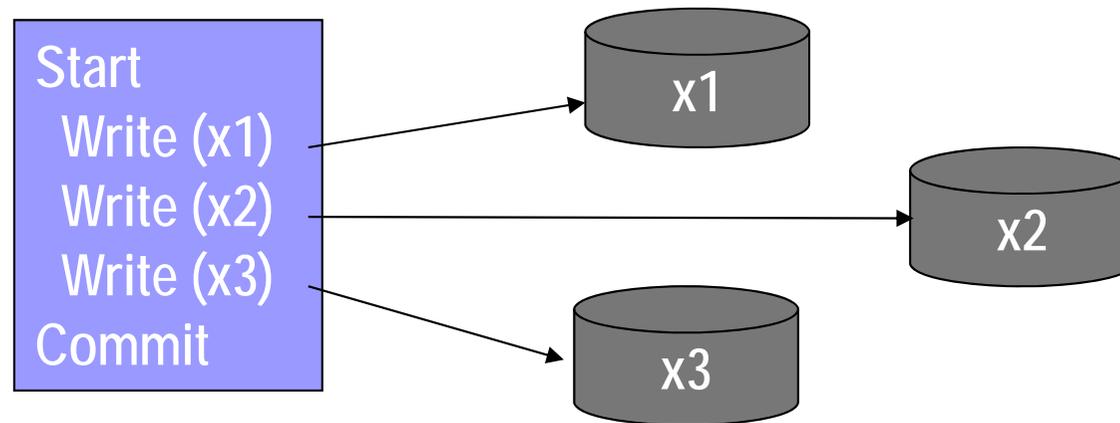
- lors d'une panne d'un serveur, on peut se replier sur l'autre

➤ Meilleure tolérance aux pannes

- possibilité de détecter des pannes diffuses

Diffusion synchrone

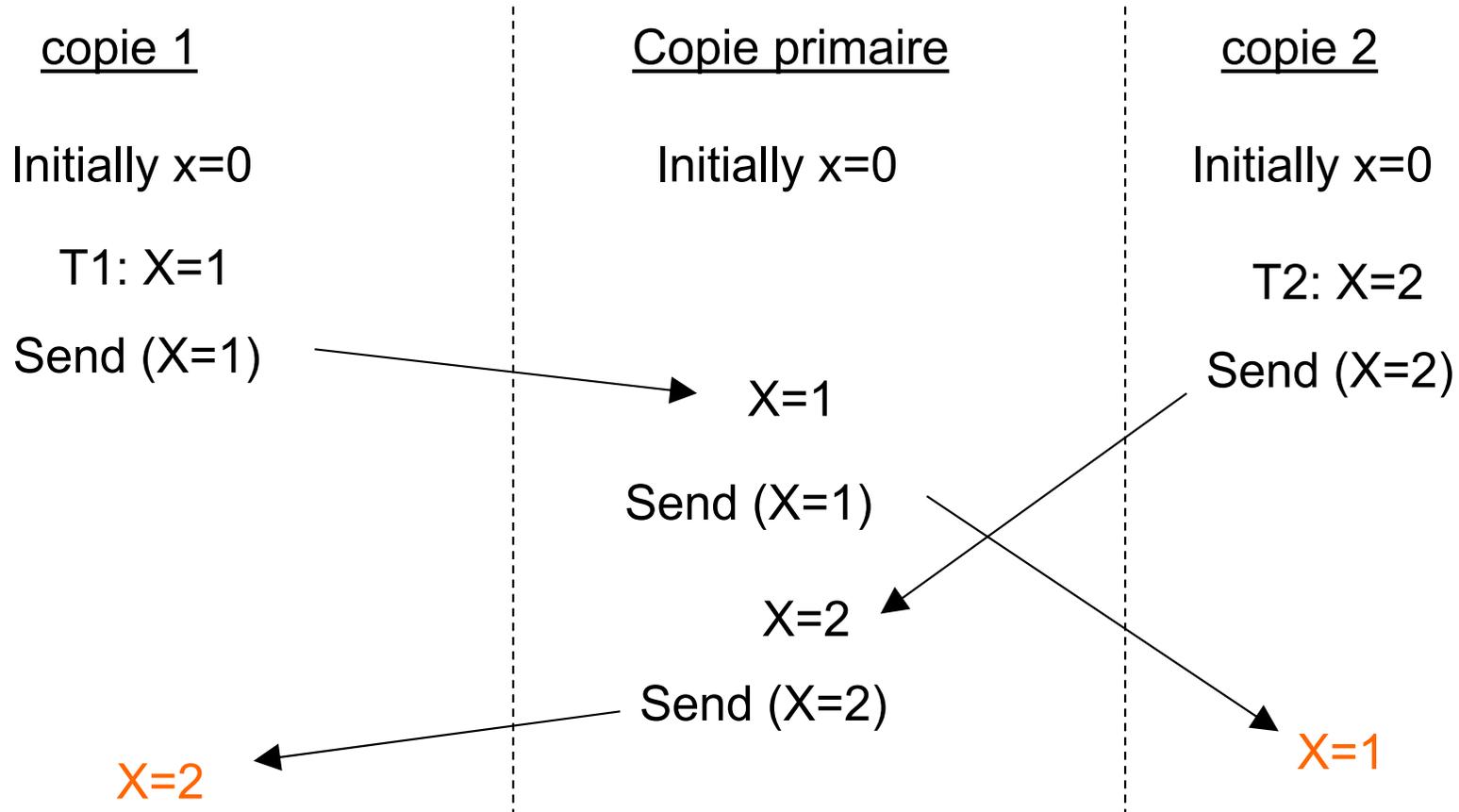
- Une transaction met à jour toutes les copies de toutes les données qu'elle modifie.
 - + mise à jour en temps réel des données
 - trop coûteux pour la plupart des applications



Diffusion asynchrone

- Chaque transaction met à jour une seule copie et la mise-à-jour des autres copies est différée (dans d'autres transactions)
- Réplication asymétrique : toutes les transactions mettent à jour la même copie
- Réplication symétrique : les transactions peuvent mettre à jour des copies différentes
 - + mise-à-jour en temps choisi des données
 - + accès aux versions anciennes puis nouvelles
 - l'accès à la dernière version n'est pas garanti

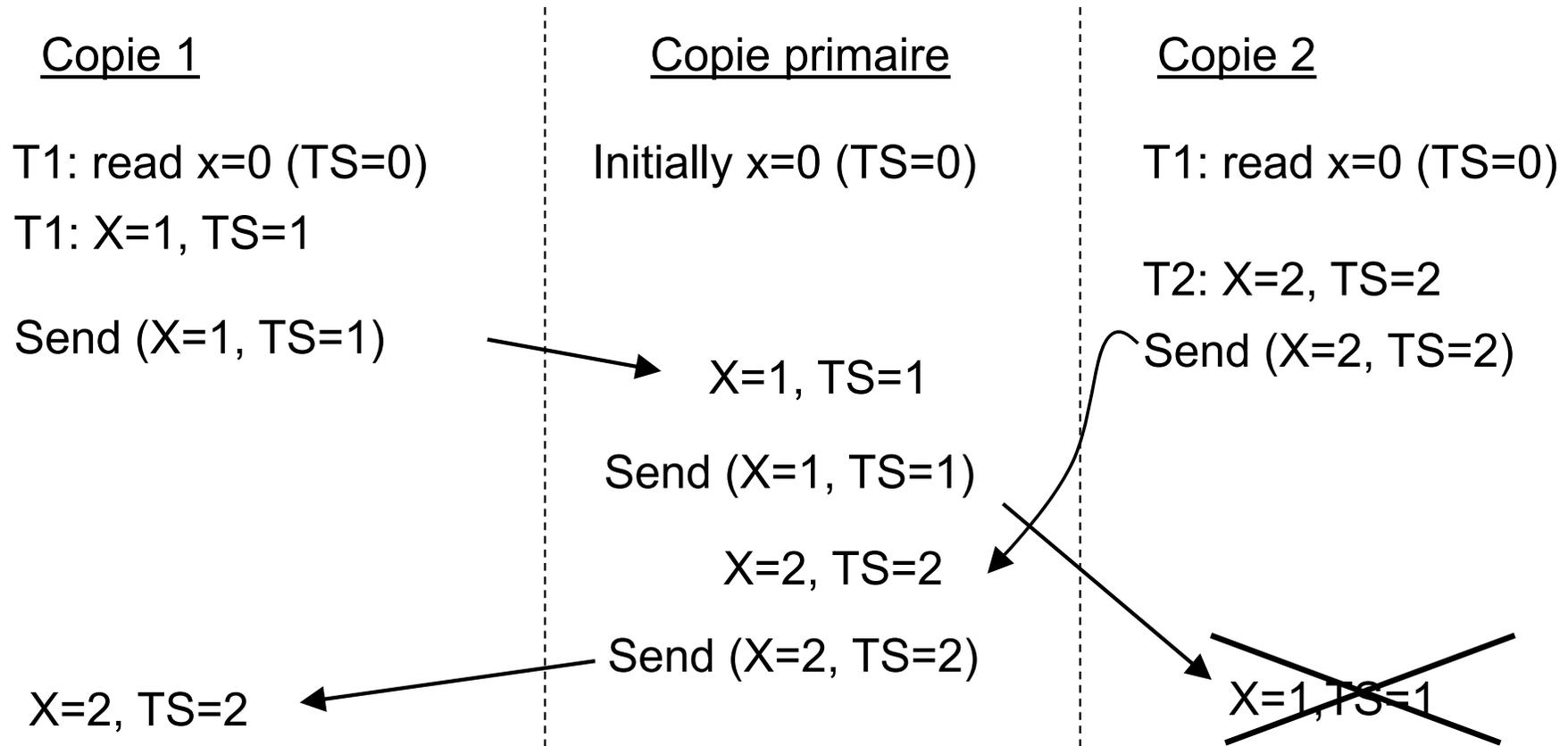
Exemple de mises à jour conflictuelles



La règle d'écriture de Thomas

- Pour assurer que l'état des copies convergent :
 - estampiller chaque record (e.g., id site + local clock)
 - une transaction met à jour un record et son estampille (toujours croissante)
 - Une mise à jour n'est appliquée que si l'estampille de la mise à jour est plus grande que l'estampille de la copie possédée
 - Il suffit de conserver les estampilles pour les records mis à jour récemment
- Tous les produits utilisent une variation de cette règle

La règle de Thomas



Références

- Özsu, M. T., & Valduriez, P. (2011). *Principles of distributed database systems*. Springer Science & Business Media.
- Ceri, S. (2017). *Distributed databases*. Tata McGraw-Hill Education.
- Coronel, C., & Morris, S. (2016). *Database systems: design, implementation, & management*. Cengage Learning.