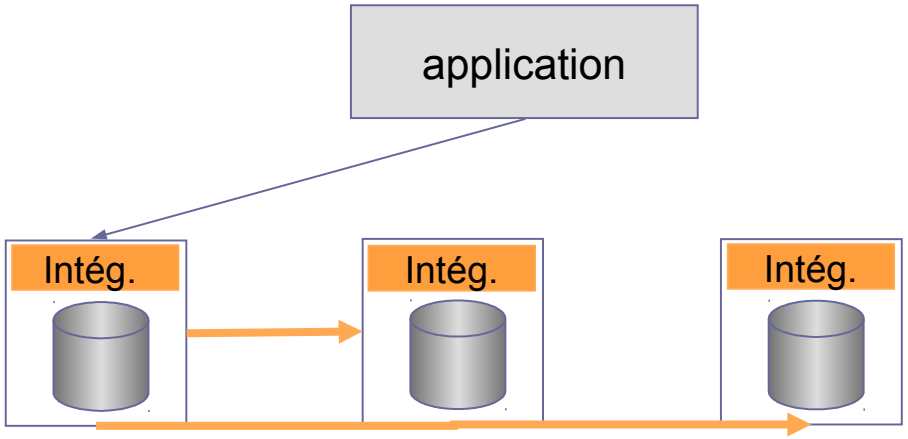
Bases de Données Réparties

1. Définition
2. Architectures
3. Conception de BDR
4. Traitement des requêtes
5. Transaction répartie
6. Passerelles avec autres SGBD

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 cohérence et de la confidentialité

Evaluation de l'approche BDR

- Avantages
  - extensibilité
  - partage des données hétérogènes et réparties
  - performances
  - disponibilité des données
- Inconvénients
  - administration complexe
  - distribution du contrôle

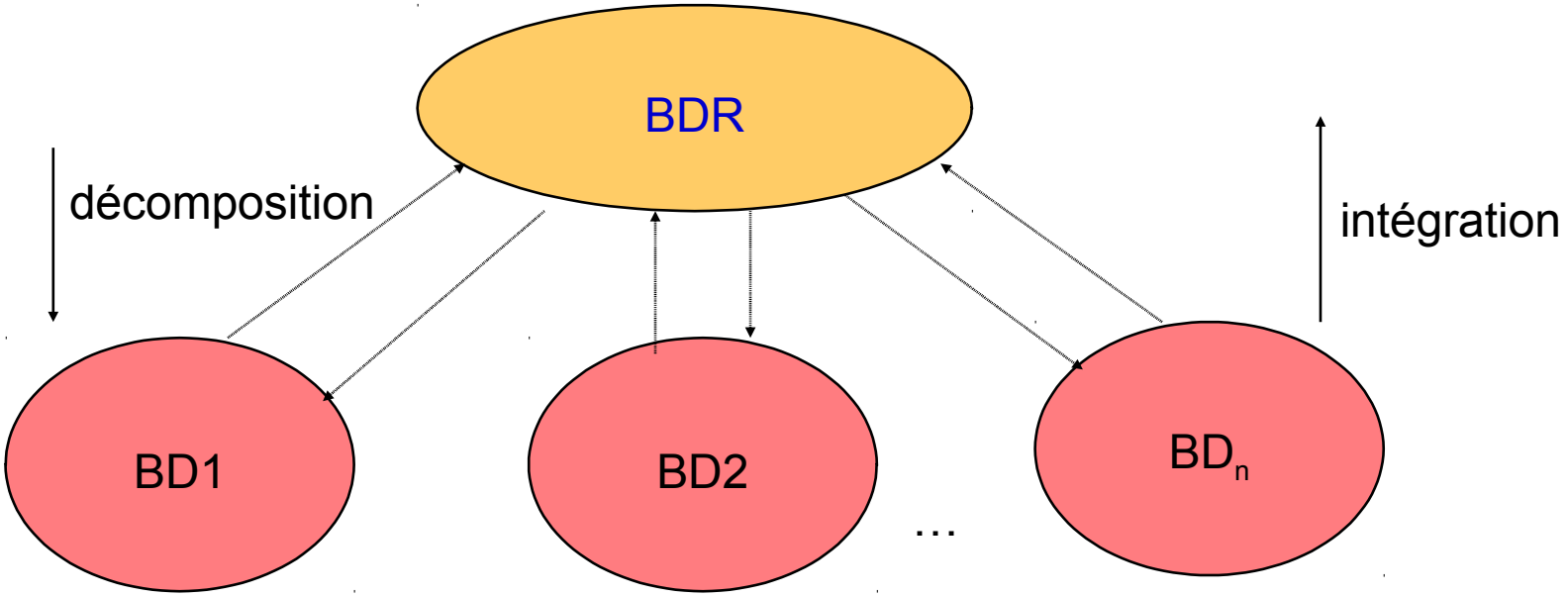
Constituants du schéma global

- 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)

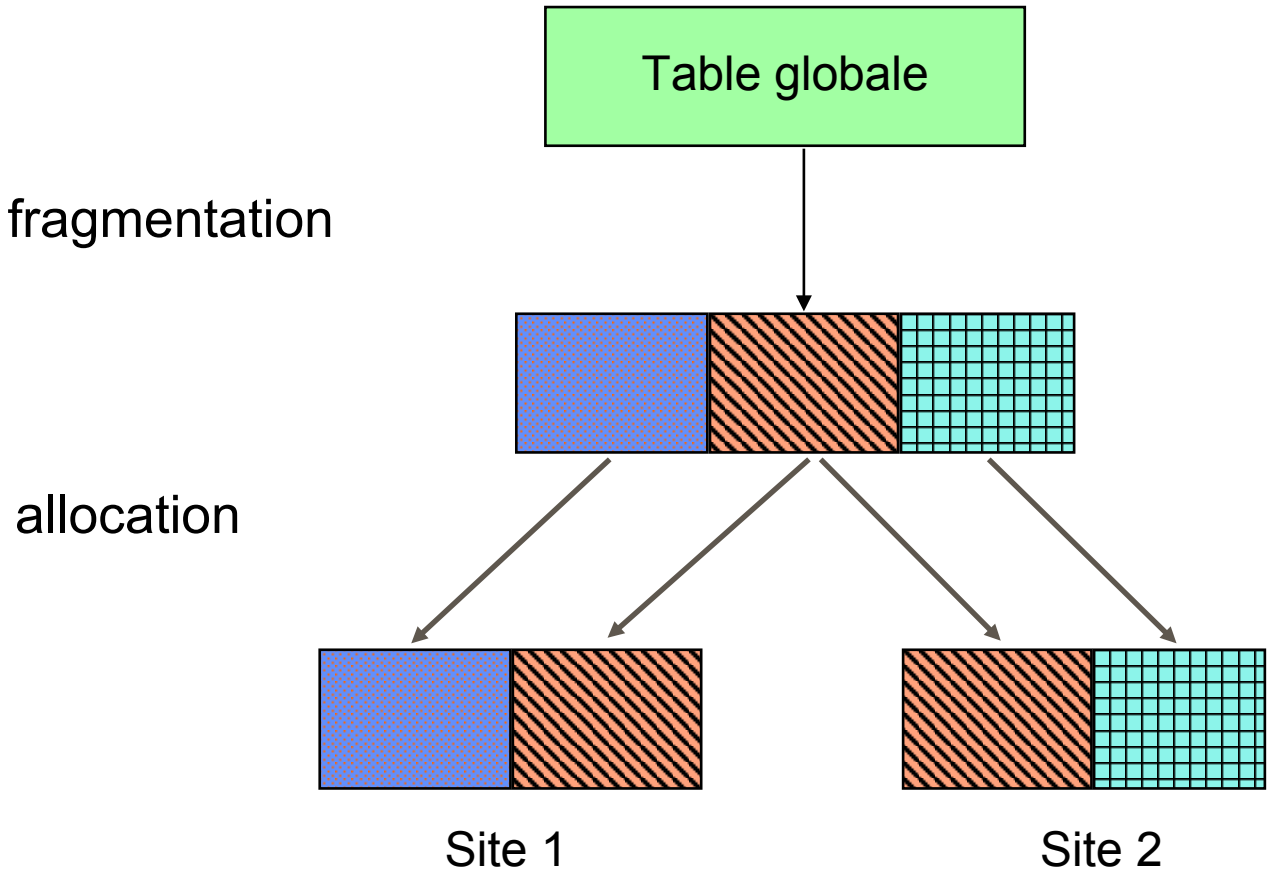
Exemple de schéma global

- Schéma conceptuel global
  - Client (nclient, nom, ville)
  - Cde (ncde, nclient, produit, qté)
- Schéma de placement
  - Client = Client1 @ Site1 U Client1 @ 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
- Conception guidée par des heuristiques

Fragmentation horizontale

- Fragments définis par sélection
  - Client1 = Client where ville = "Bafang"
  - Client2 = Client where ville ≠ "Bafang"

Reconstruction

Client = Client1 U Client2

Client

nclient	nom	ville
C 1 C 2 C 3 C 4	Kamga Essame Essame Bello	Bafang Douala Bafang Ebolowa

Client1

nclient	nom	ville
C 1 C 3	Kamga Essame	Bafang Bafang

Client2

nclient	nom	ville
C 2 C 4	Essame Bello	Douala Ebolowa

Fragmentation horizontale dérivée

Fragments définis par jointure

Cde1 = Cde where
Cde.nclient = Client1.nclient

Cde2 = Cde where
Cde.nclient = Client2.nclient

Reconstruction

Cde = Cde1 U Cde2

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

Cde2

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

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
- Utile si forte affinité d'attributs

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 [lectures/màj] > 1, la duplication est avantageuse

Exemple d'allocation de fragments

Client1

nclient	nom	ville
C 1 C 3	Kamga Essame	Bafang Bafang

Client2

nclient	nom	ville
C 2 C 4	Essame Bello	Douala Ebolowa

Cde1

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

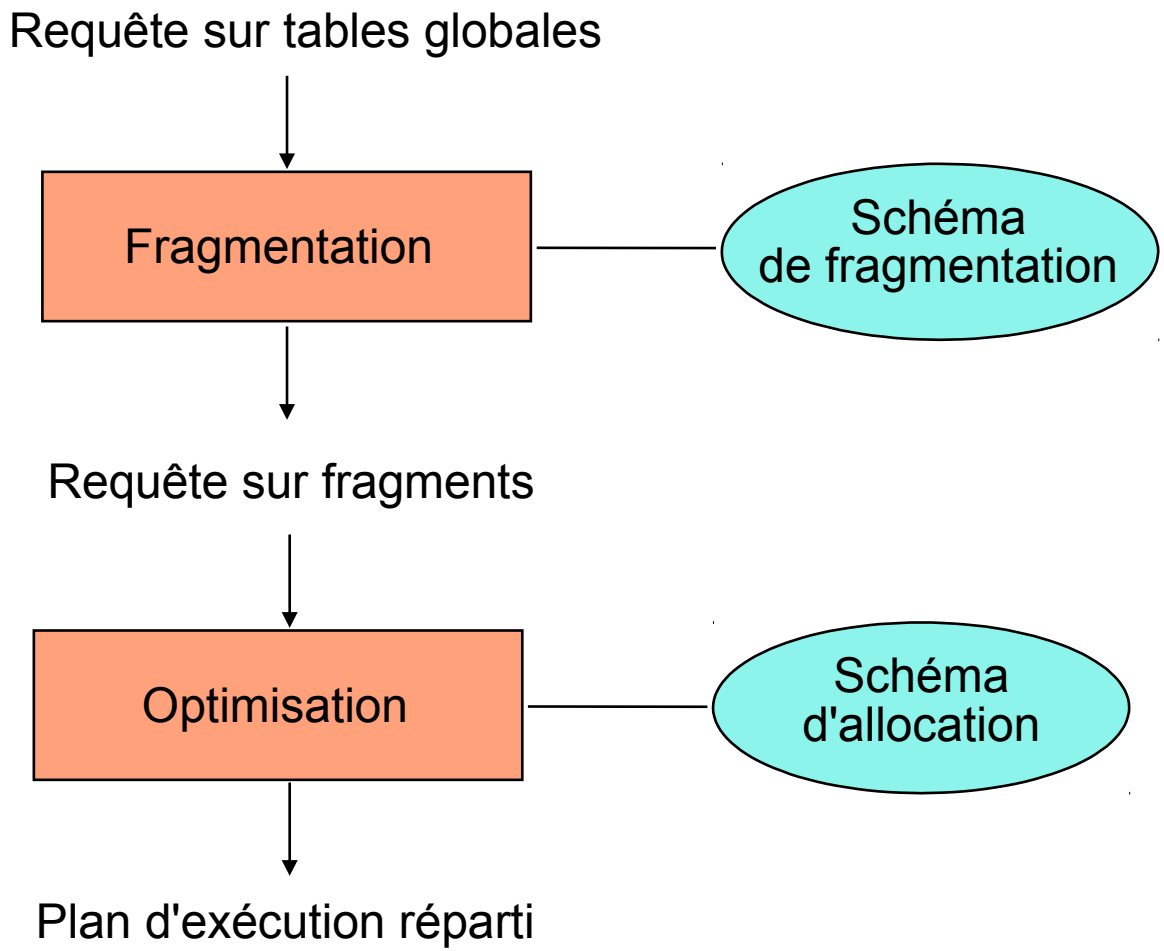
Site 1

Cde2

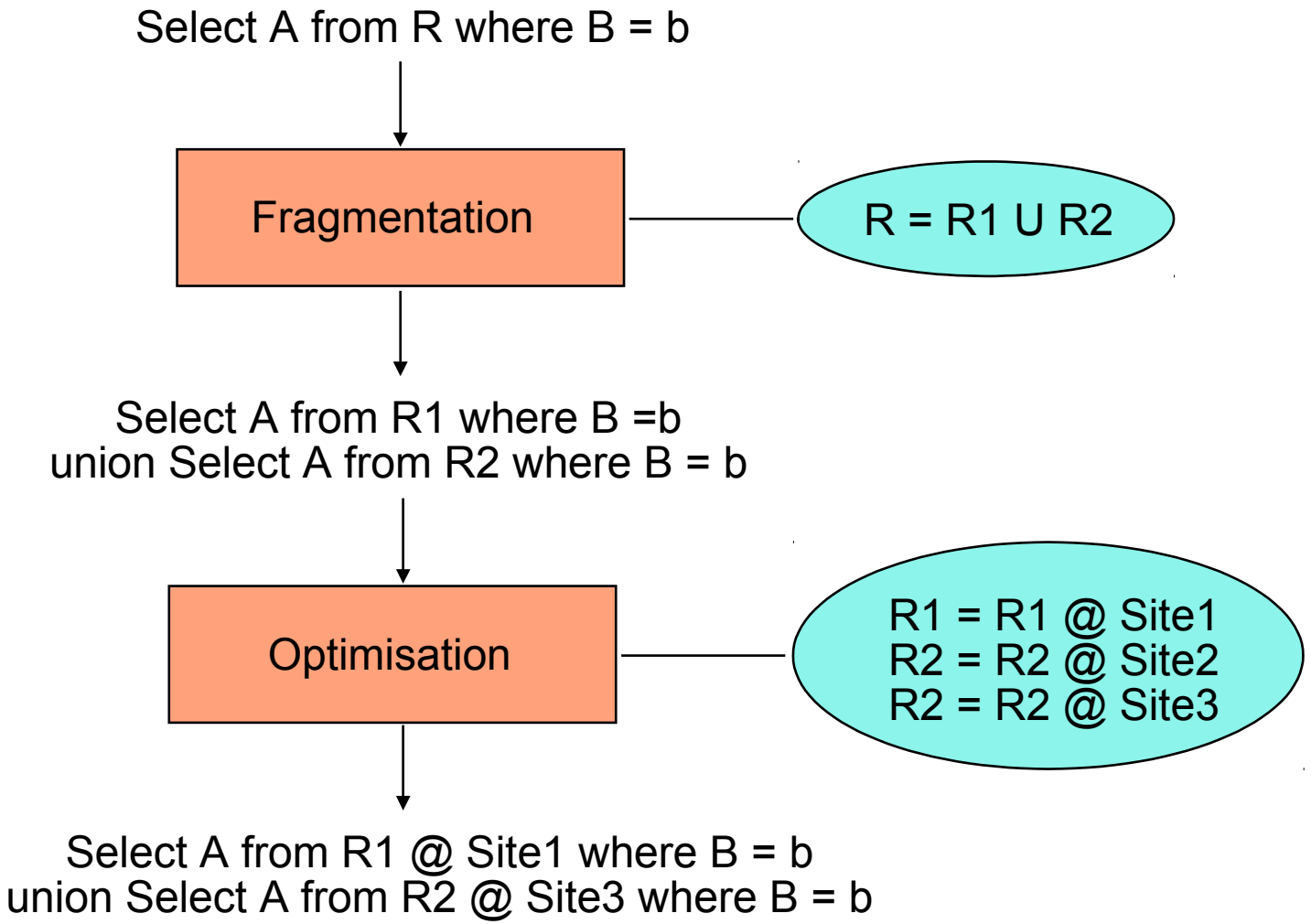
ncde	client	produit	qté
D 3 D 4	C 2 C 4	P 3 P 4	5 10

Site 2

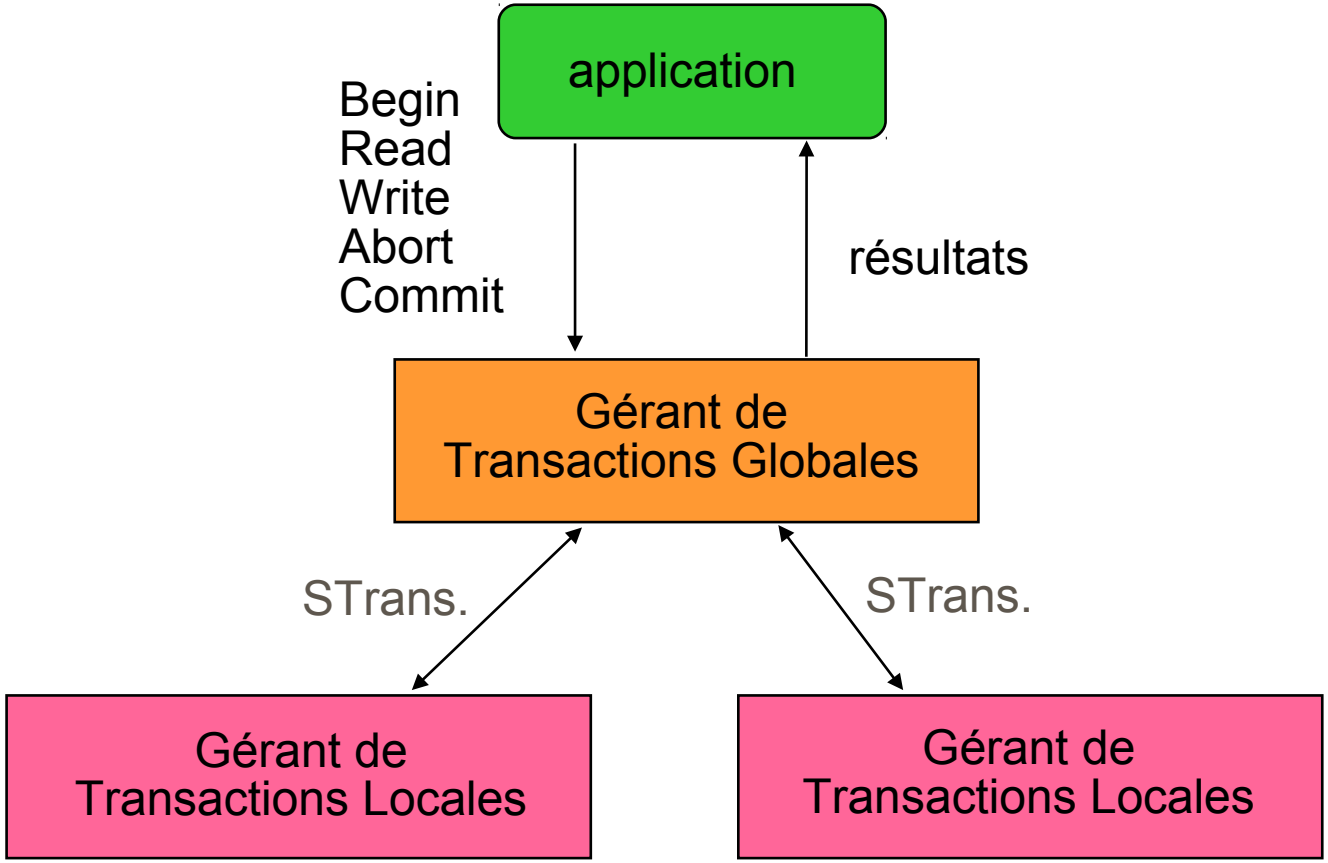
Evaluation de requêtes réparties



Exemple d'évaluation simple



Notion de Transaction Répartie

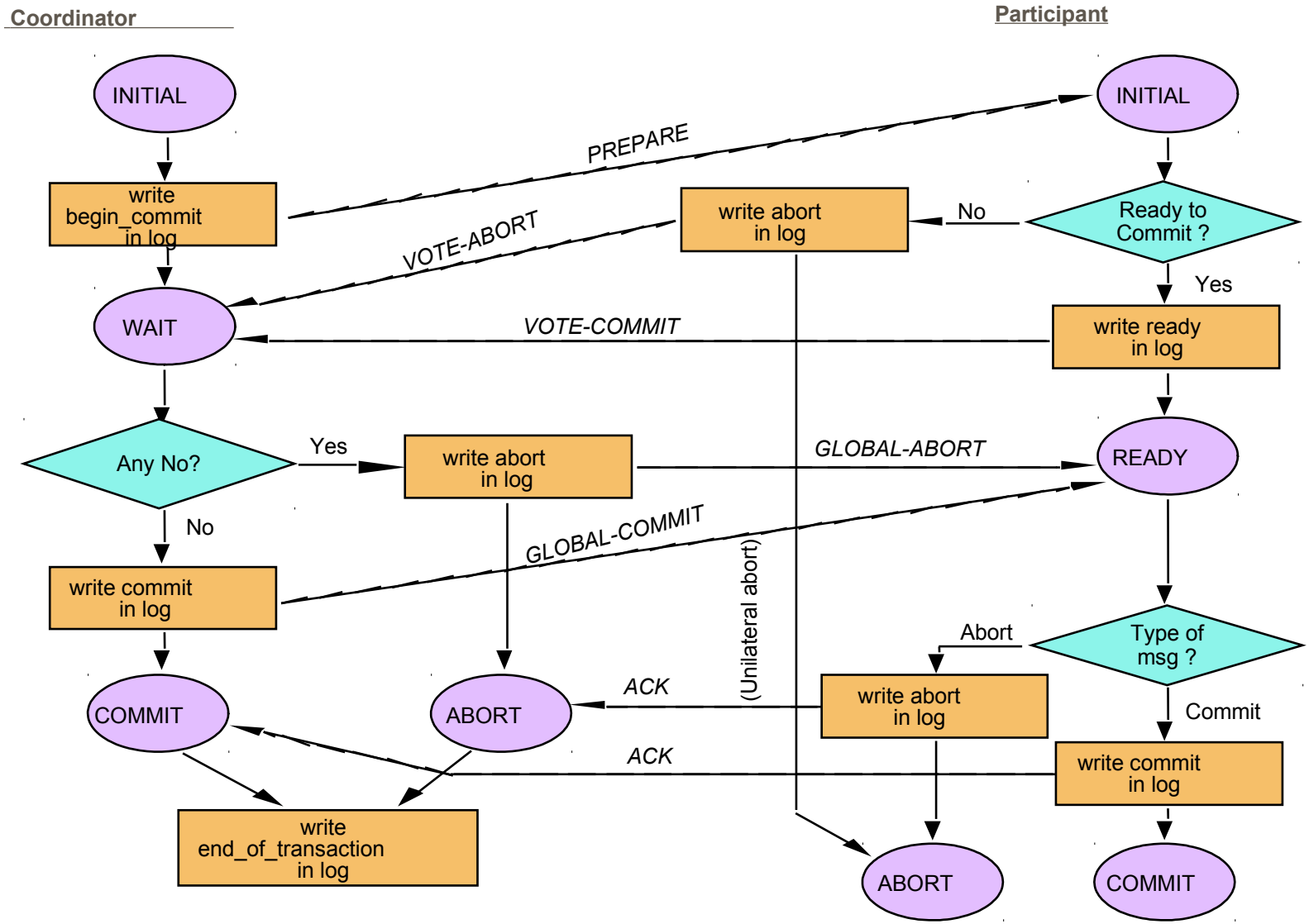


Protocole de validation en 2 étapes

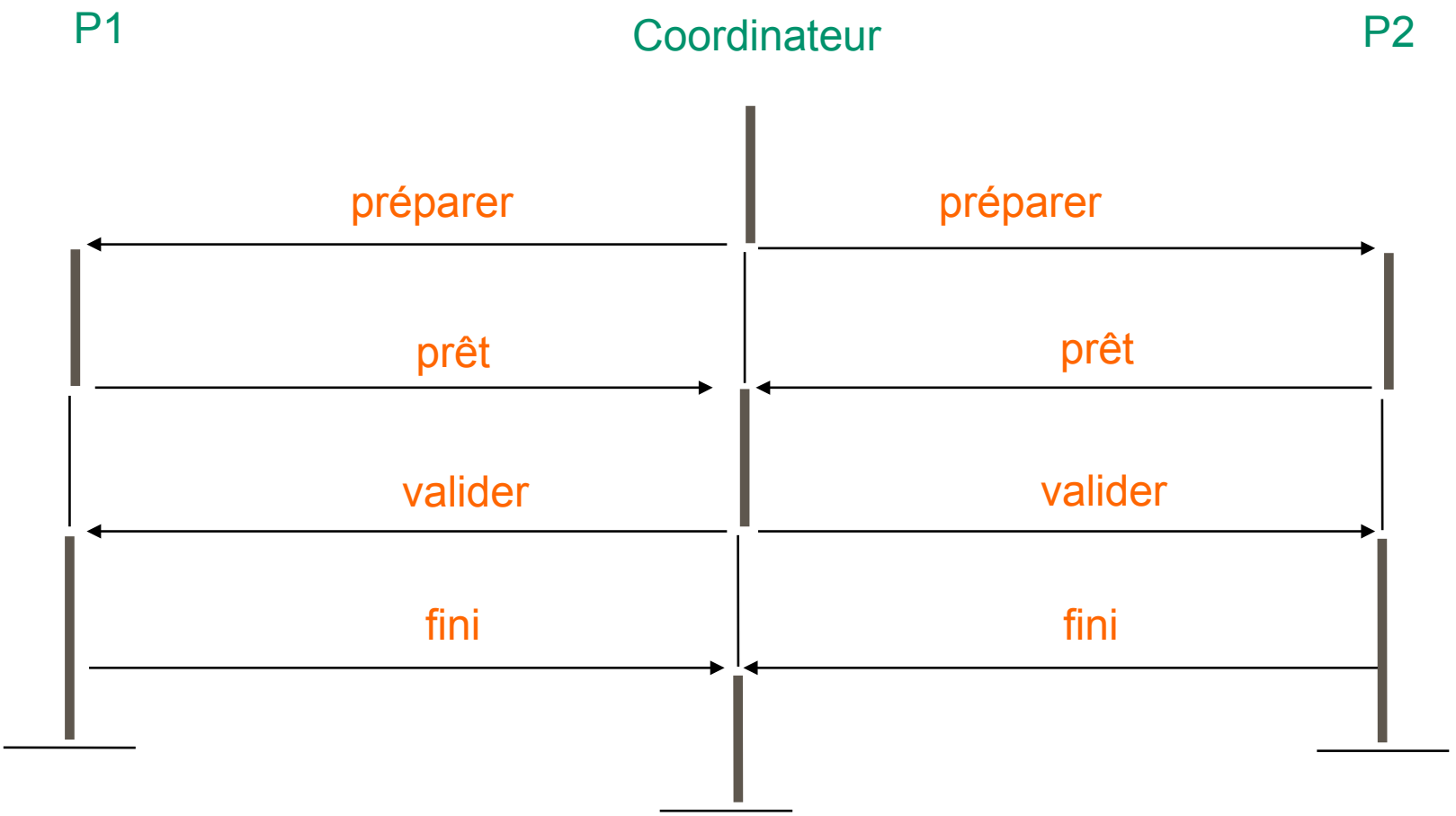
- **Objectif** : Exécuter la commande COMMIT pour une transaction répartie
  - Phase 1 : Préparer à écrire les résultats des mises-à-jour dans la BD
  - Phase 2 : Ecrire ces résultats dans la BD

- **Coordinateur** : composant système d'un site qui applique le protocole
- **Participant** : composant système d'un autre site qui participe dans l'exécution de la transaction

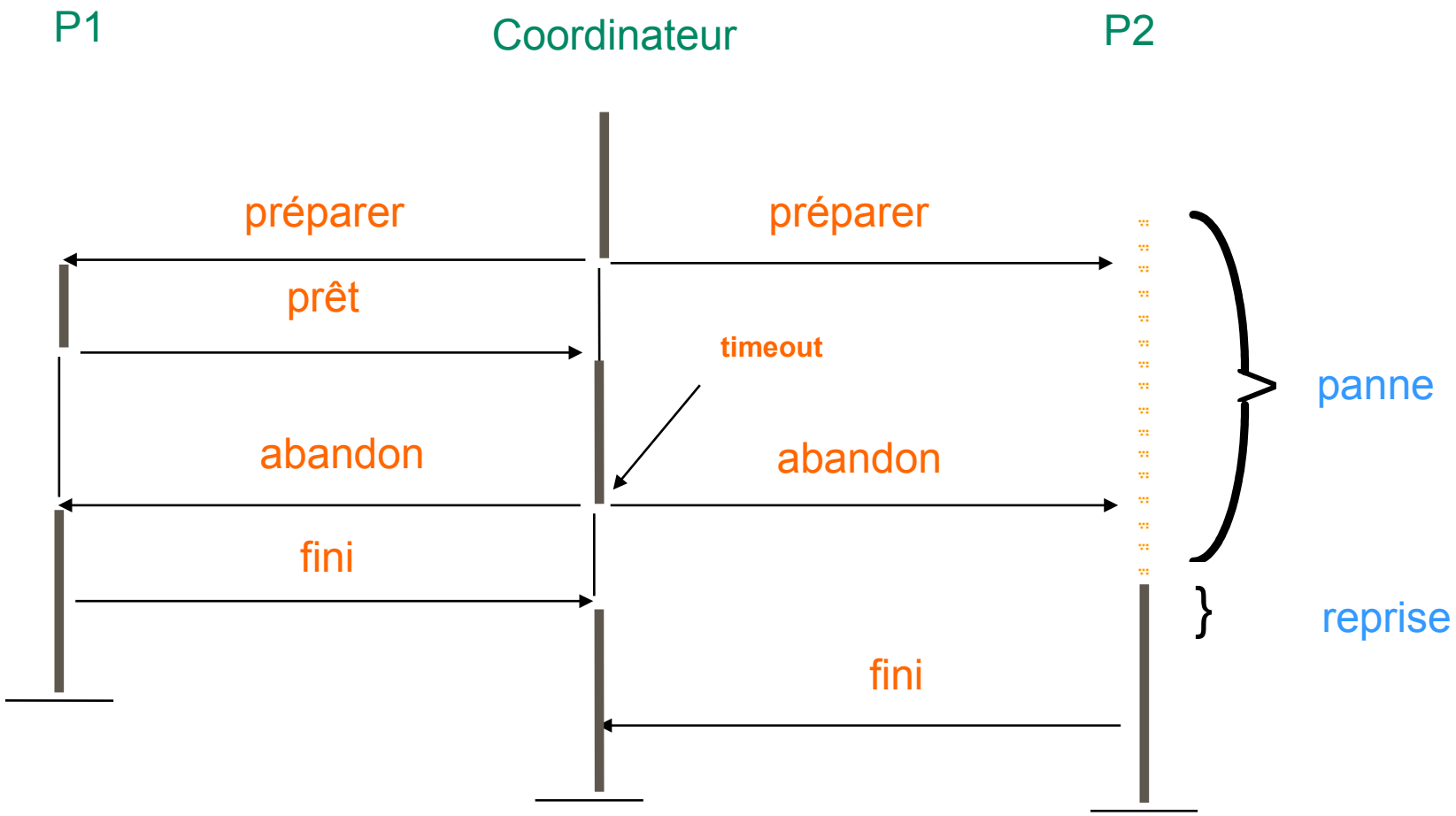
Etude de cas de défaillances



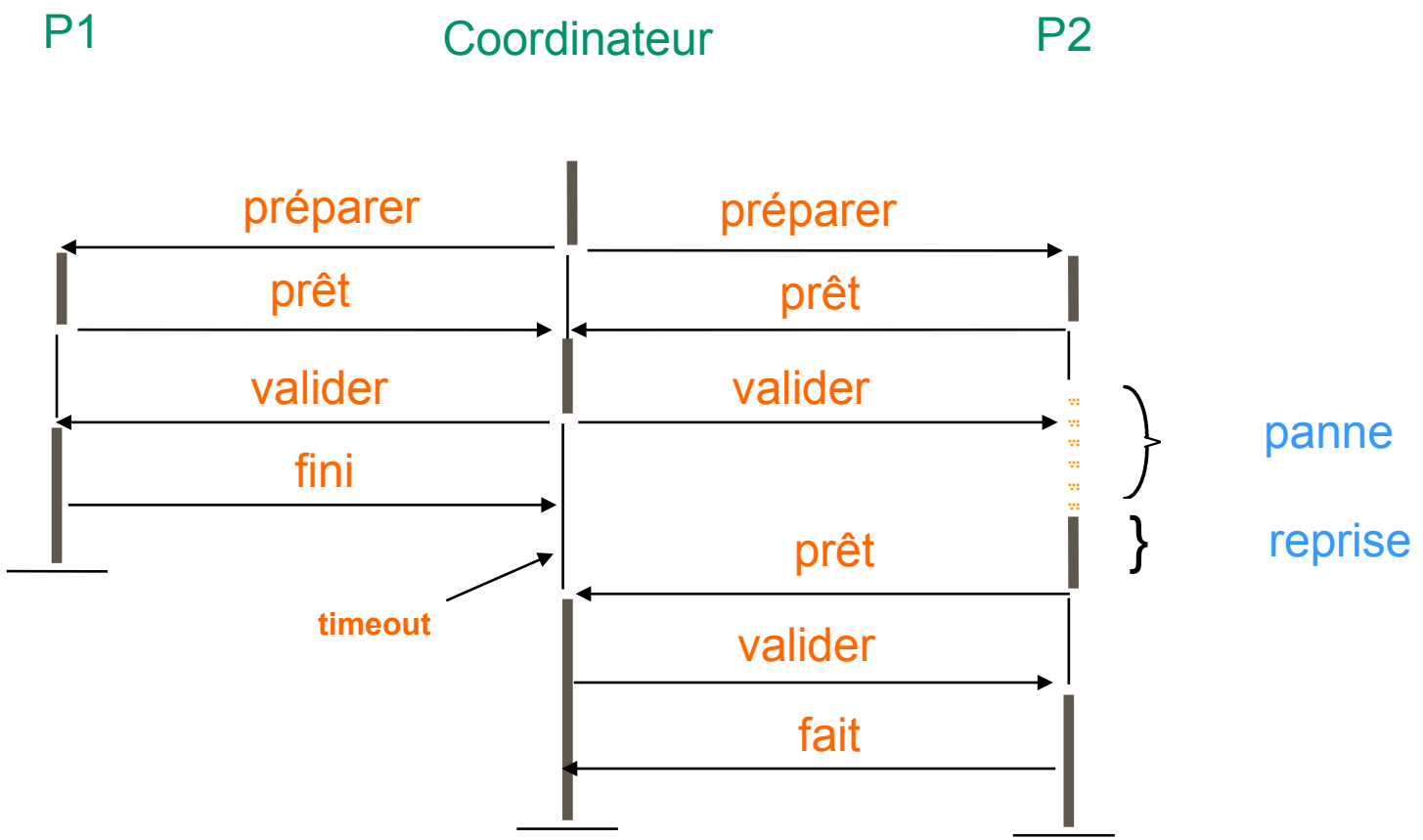
Validation normale



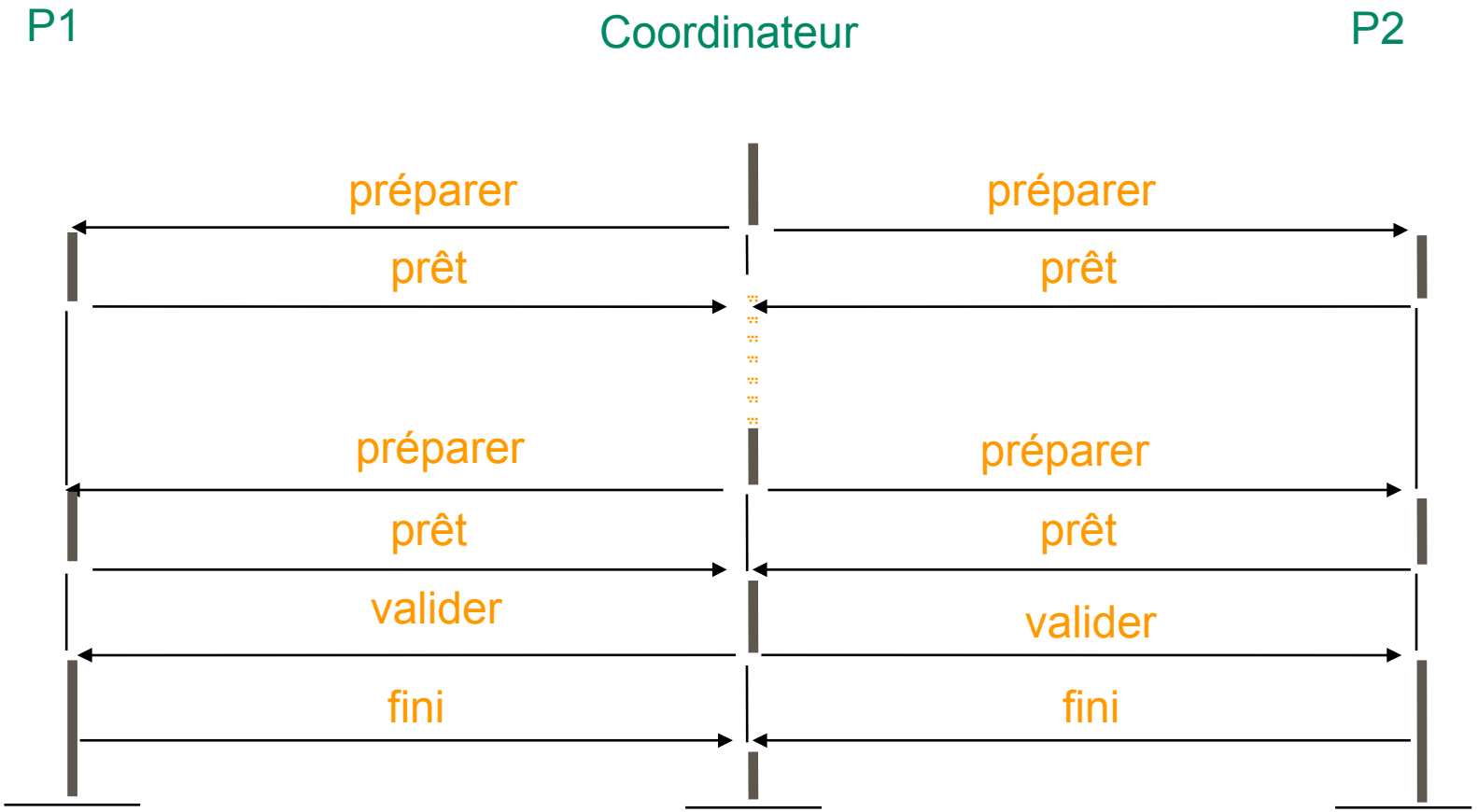
Panne d'un participant avant Prêt



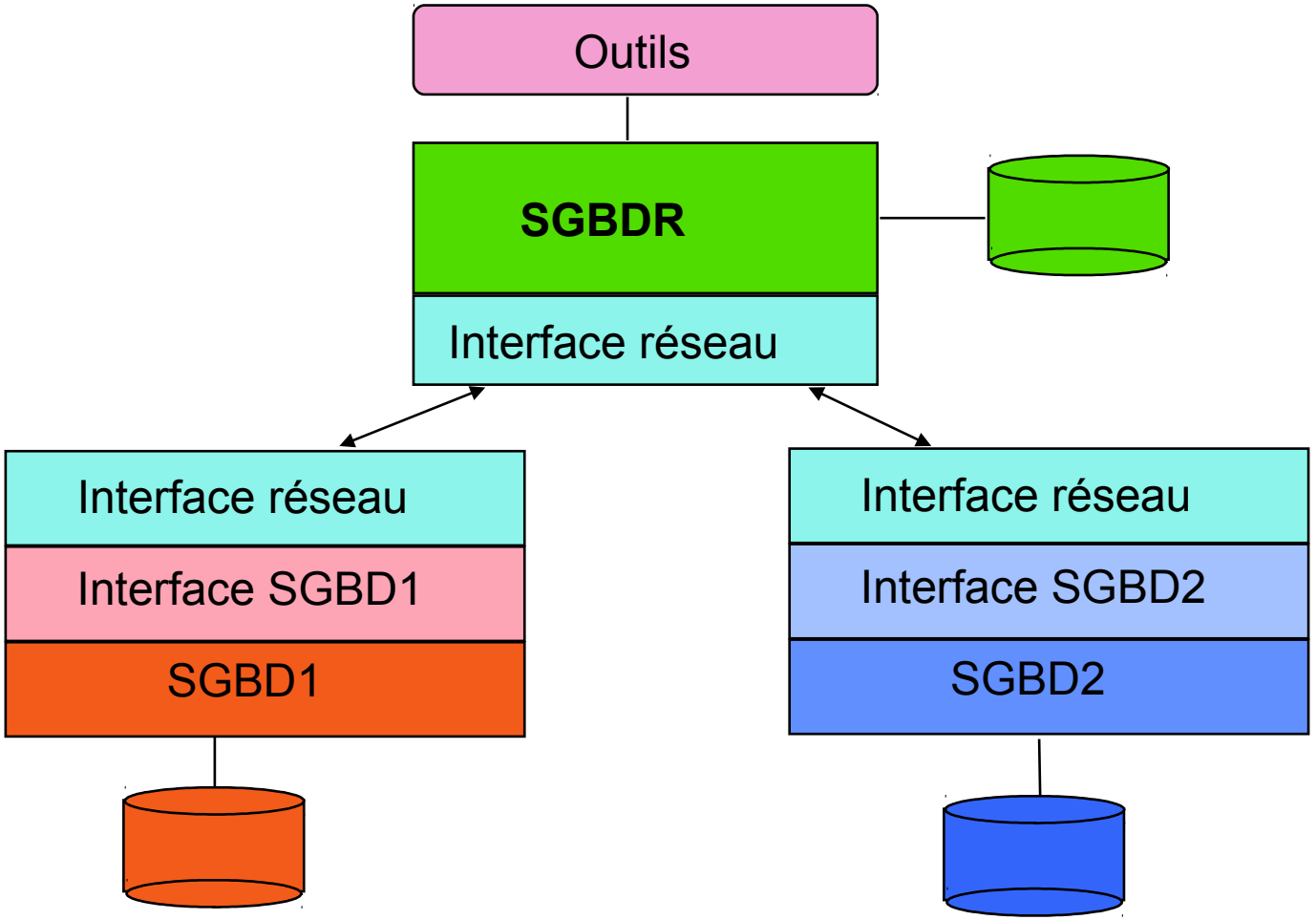
Panne d'un participant après Prêt



Panne du coordinateur



SGBD réparti hétérogène



Produits

- SGBD relationnels
  - Oracle, DB2, SQL Server 2000, Sybase, Informix
- VirtualDB (Enterworks)
  - basé sur GemStone, vue objet des tables
- Open Database Exchange (B2Systems)

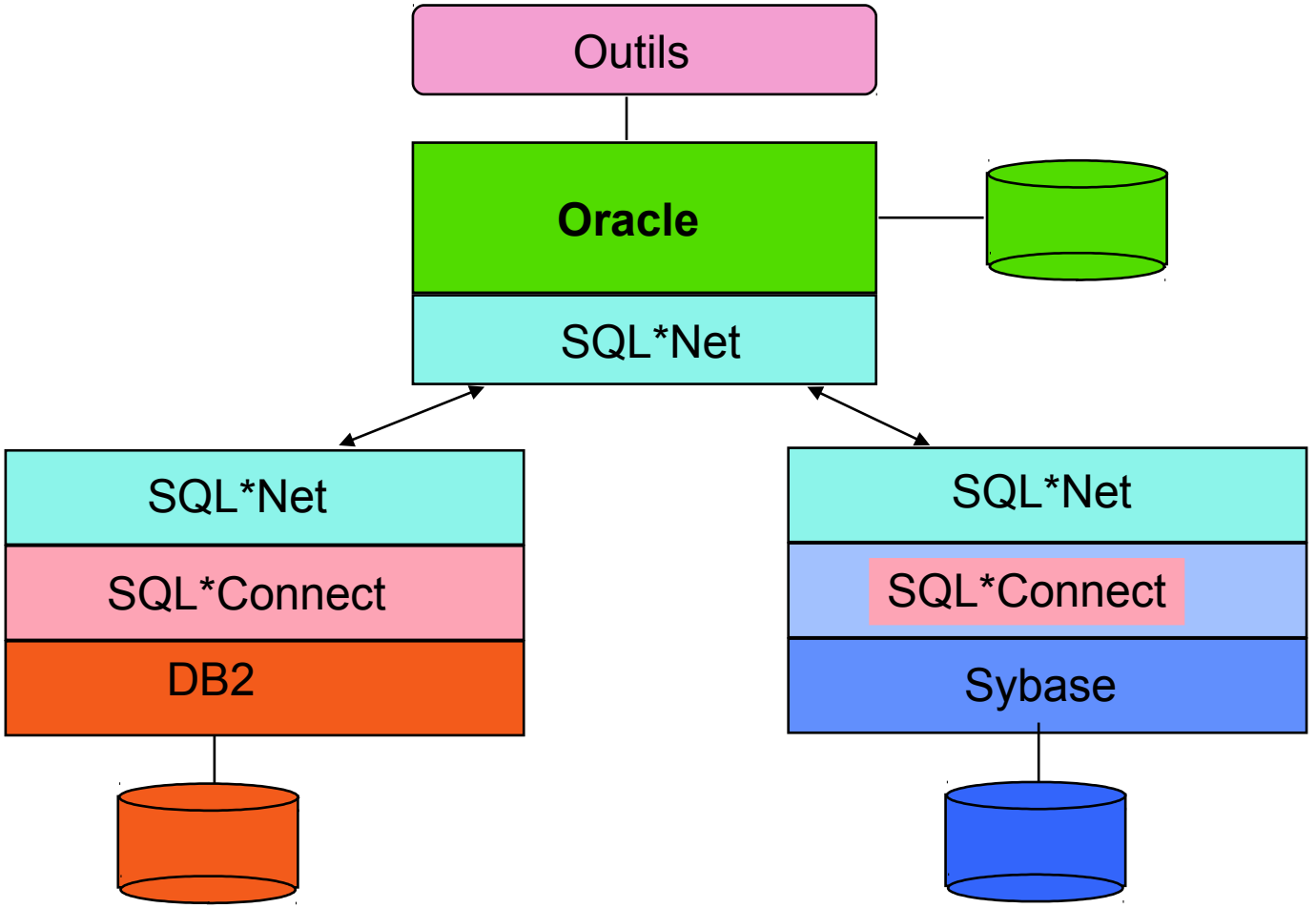
Oracle/Star

- SGBD Oracle
  - gestion du dictionnaire de la BDR
- SQL*Net
  - transparence au réseau
  - connexion client-serveur, loggin à distance automatique
  - évaluation de requêtes réparties
  - validation en deux étapes et réplication
- SQL*Connect : passerelle vers les bases non-Oracle

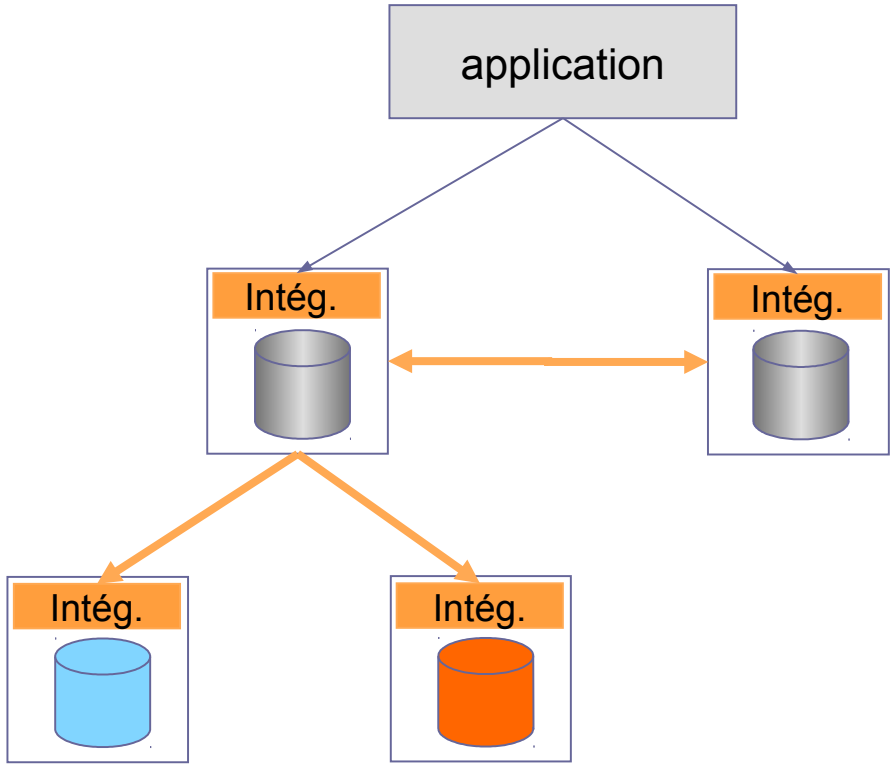
Database link

- Lien à une table dans une BD distante specifié par :
  - nom de lien
  - nom de l'utilisateur et password
  - chaîne de connexion SQL*Net (protocole réseau, nom de site, options, etc…)
- Exemple
  - CREATE DATABASE LINK empParis
  - CONNECT TO patrick
  - IDENTIFIEDBY monPW
  - USING Paris.emp

Oracle/Star : architecture



Architecture finale de l’infastructure



Difficultés des bases réparties

- Choix et maintien des fragments
  - En fonction des besoins des applications
  - Heuristiques basées sur l'affinité d'attributs et le regroupement
- Disponibilité des données
  - Dépend de la robustesse du protocole 2PC; implique une grande fiabilité du réseau et des participants
- Echelle
  - Le nombre de sessions simultanées est limité par l'architecture 2-tiers; grande échelle nécessite un moniteur transactionnel

Fonctionnalités d’intégration BDR

Fonctionnalité	**Réponse BDR**
Définition de vues intégrées	Modèle relationnel – vues par fragmentation et réplication à partir des données locales. Schéma global, droits d’accès, contraintes d’intégrité simples
Langage de manipulation de données	Requêtes SQL de sélection et de mise à jour. Transactions ACID réparties
Interfaces applicatives	Idem SGBD

Règle d’urbanisation BDR

Caractéristiques données sources	Bases de données relationnelles ou sources dotées d’un connecteur adapté (2PC, …) Coopération forte entre sources
Caractéristiques données cibles	Données virtuelles Faibles capacités de transformation Cohérence forte des données Disponibilité des données fragile Bonnes performances d’accès
Coût	Robustesse du réseau et des sources Administration

Bases de données répliquées

1. Intérêt de la réplication
2. Diffusion synchrone et asynchrone
3. Réplication asymétrique
4. Gestion des défaillances
5. Réplication symétrique
6. Conclusions

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 :
  - soit « logique » dans le cas d'une vue
  - soit physique

Les avantages de la réplication

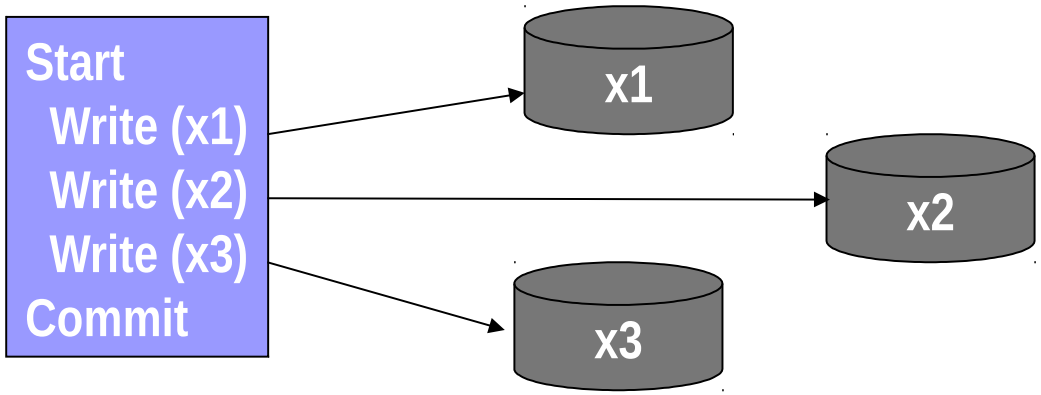
- Amélioration des performances
  - lecture de la copie la plus proche
  - évitement du goulot d'étranglement du serveur unique
- Amélioration de la disponibilité
  - lors d'une panne d'un serveur, on peut se replier sur l'autre
  - Disponibilité = 1 - probabilité_panne^N
    - probabilité de panne = 5% et 2 copies => disponibilité = 99.75%
- Meilleure tolérance aux pannes
  - possibilité de détecter des pannes diffuses

Les problèmes de la réplication

- Convergence
  - les copies doivent être maintenues à jour
  - à un instant donné, elles peuvent être différentes
  - mais elles doivent converger vers un même état cohérent où toutes les mises à jour sont exécutées partout dans le même ordre
- Transparence : le SGBD doit assurer
  - la diffusion et la réconciliation des mises à jour
  - la résistance aux défaillances

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
  - - pas de contrôle de l ’instant de mise-à-jour

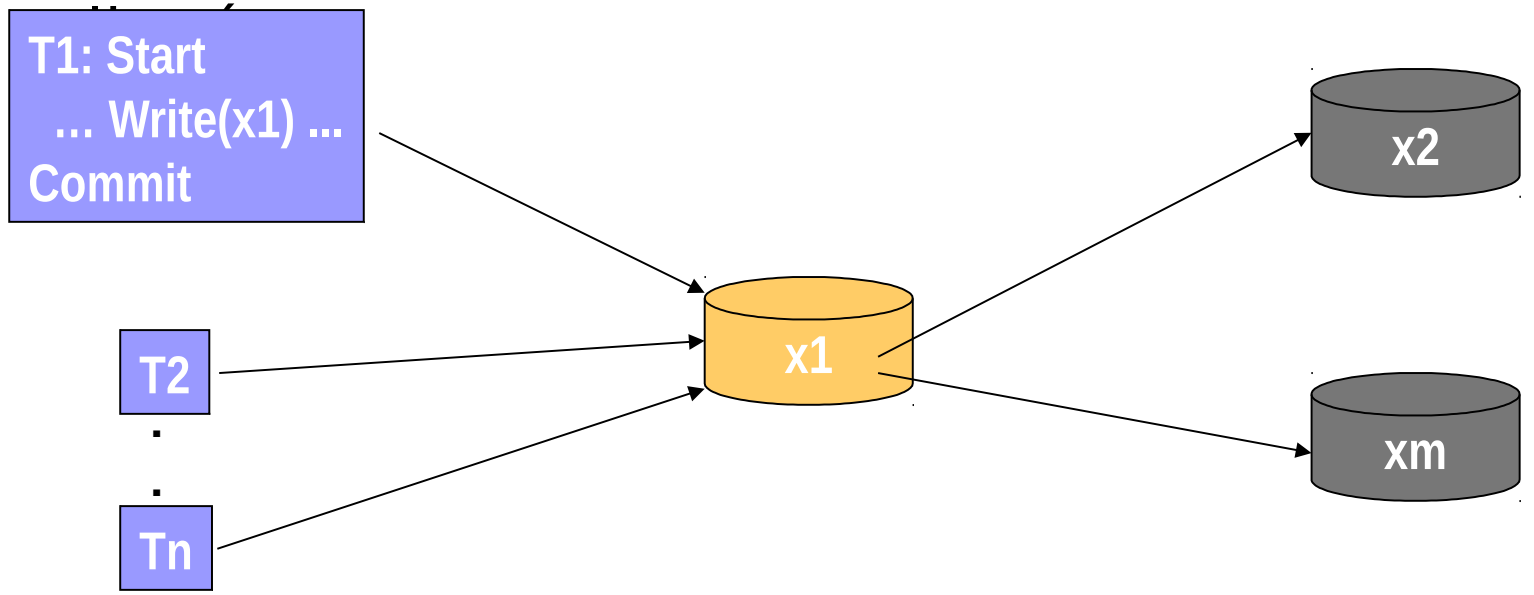


Diffusion asynchrone

- 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

Réplication asymétrique

- Désigner une copie comme *primaire* (“publisher”) ; les transactions ne mettent à jour que cette copie
- les mises à jour de la copie primaire sont envoyées ultérieurement aux copies *secondaires* (“subscribers”) dans l’ordre où elles ont été a

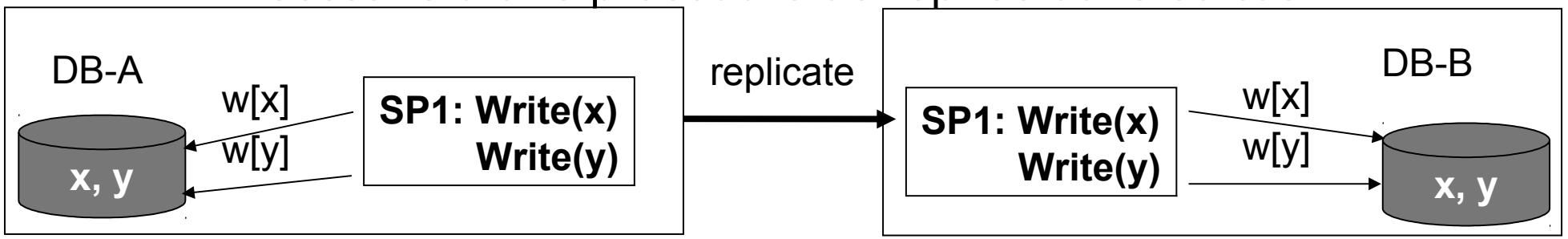


Diffusion asynchrone - asymétrique

- Collecte des mises-à-jour sur la copie primaire via :
  - des triggers (Oracle, Rdb, SQL Server, DB2, ...)
  - Le journal des images après (“log sniffing”) (SQL Server, DB2, Tandem Non-Stop SQL, Sybase Replication Server)
    - Off-line
    - R/W log synchronization
    - administration

Diffusion asynchrone - asymétrique (2)

- Autre technique : diffuser une requête plutôt que les données mises à jour (e.g., stored procedure call)
- Problème : assurer le bon ordonnancement des requêtes
  - Les requêtes peuvent être diffusées de façon synchrone à toutes les copies mais la diffusion est validée même si une la mise à jour sur une copie a échoué
  - nécessité d'une procédure de reprise dans ce cas



Gestion des défaillances de site

- Défaillance d’une copie secondaire - rien à faire
  - Après reprise, appliquer les mises à jour oubliées pendant la panne (déterminées à partir du journal)
  - Si panne trop longue, il est préférable d’obtenir une copie neuve
- Défaillance d’une copie primaire – idem dans les produits

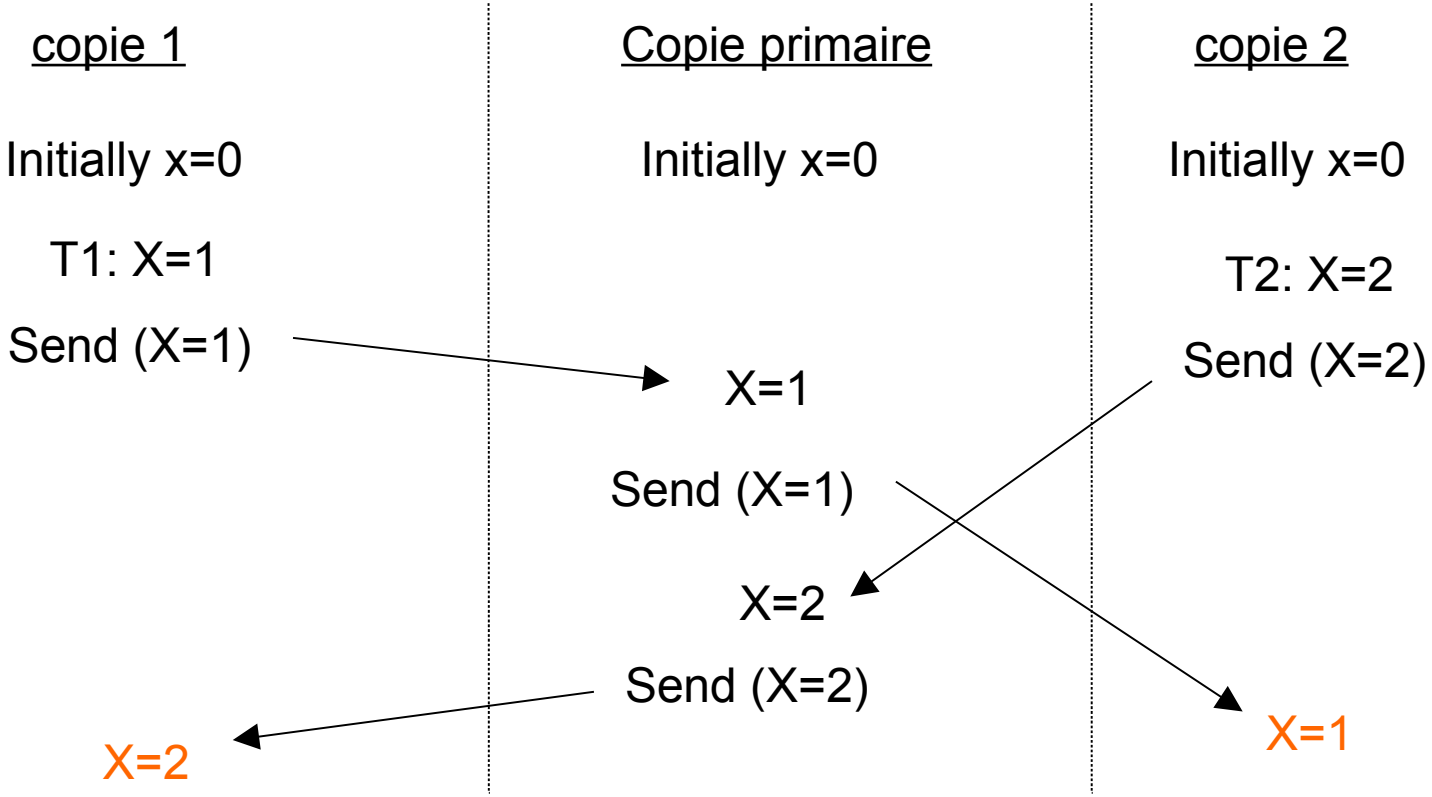
Défaillance des communications

- Les copies secondaires ne peuvent pas distinguer 1 panne de communication d'une panne de site
- Si les secondaires élisent un nouveau primaire et l'ancien primaire est toujours vivant, il y aura un pb de réconciliation ...
- Une solution est qu'une partition du réseau sache qu'elle est la seule à pouvoir fonctionner, mais elle ne peut pas communiquer avec les autres partitions pour le savoir.
  - décision statique : la partition qui possède le primaire gagne
  - solution dynamique : consensus majoritaire

Réplication symétrique

- Certains systèmes doivent fonctionner même s'ils sont partitionnés
  - plusieurs copies sont mises à jour (pas seulement une)
  - les conflits de mise à jour sont détectés après coup
- Exemple classique - portable du commercial déconnecté
  - Customer table (rarement mise à jour); Orders table (insertion)
  - Customer log table (append)
  - les conflits de mise à jour sont rares !
- Méthode :
  - quand une copie se reconnecte au réseau, il y a "échange" :
    - elle envoie ses mises à jour avec la copie primaire
    - la copie primaire lui envoie les mises à jour reçues
  - les mises à jour conflictuelles nécessitent une réconciliation

Exemple de mises à jour conflictuelles

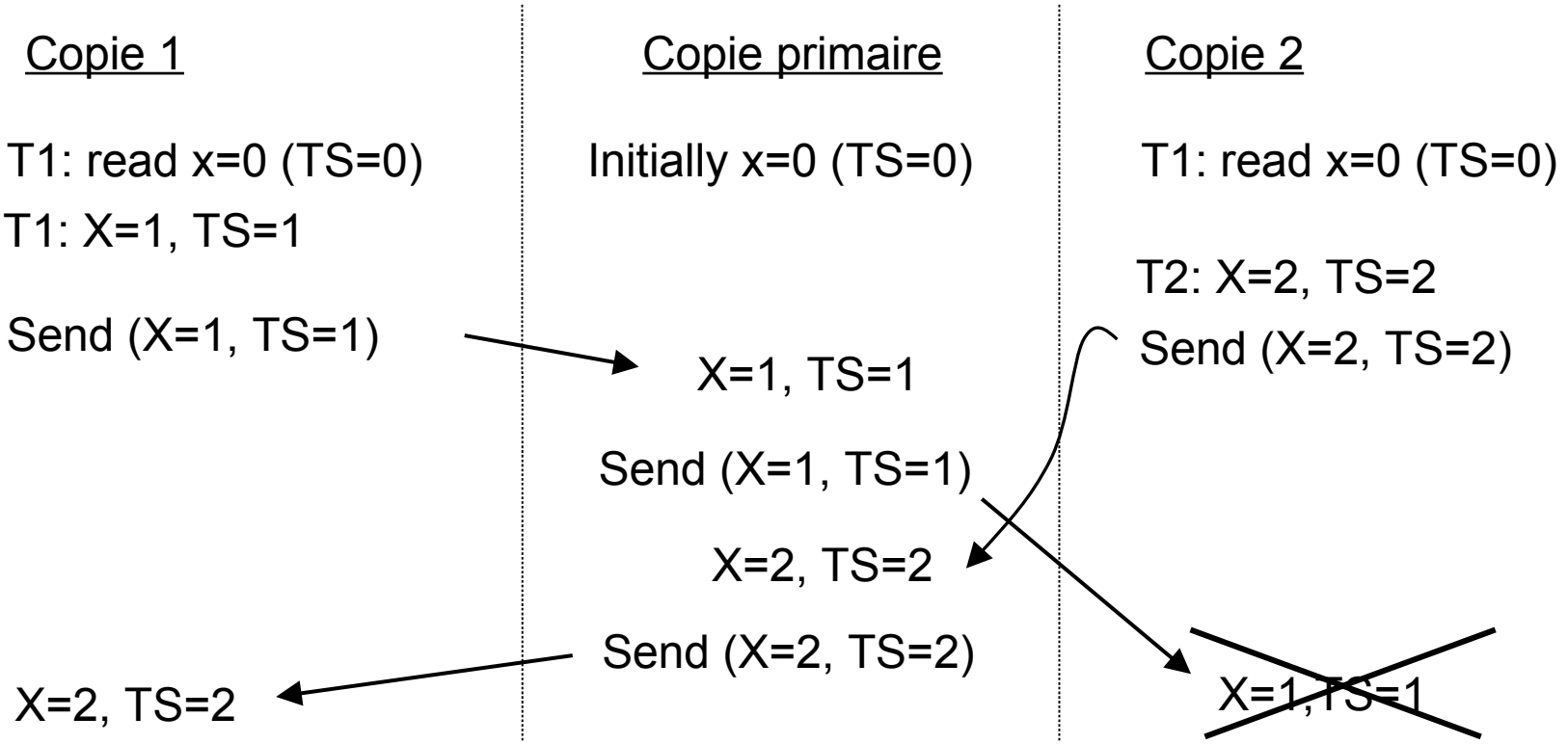


La règle d'écriture de Thomas

- 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 possedée
  - 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 ⇒ Sérialisabilité



- Ni T1 ni T2 ne lisent le résultat l'une de l'autre. Cette exécution n'est pas sérialisable.

Performances de la réplication symétrique

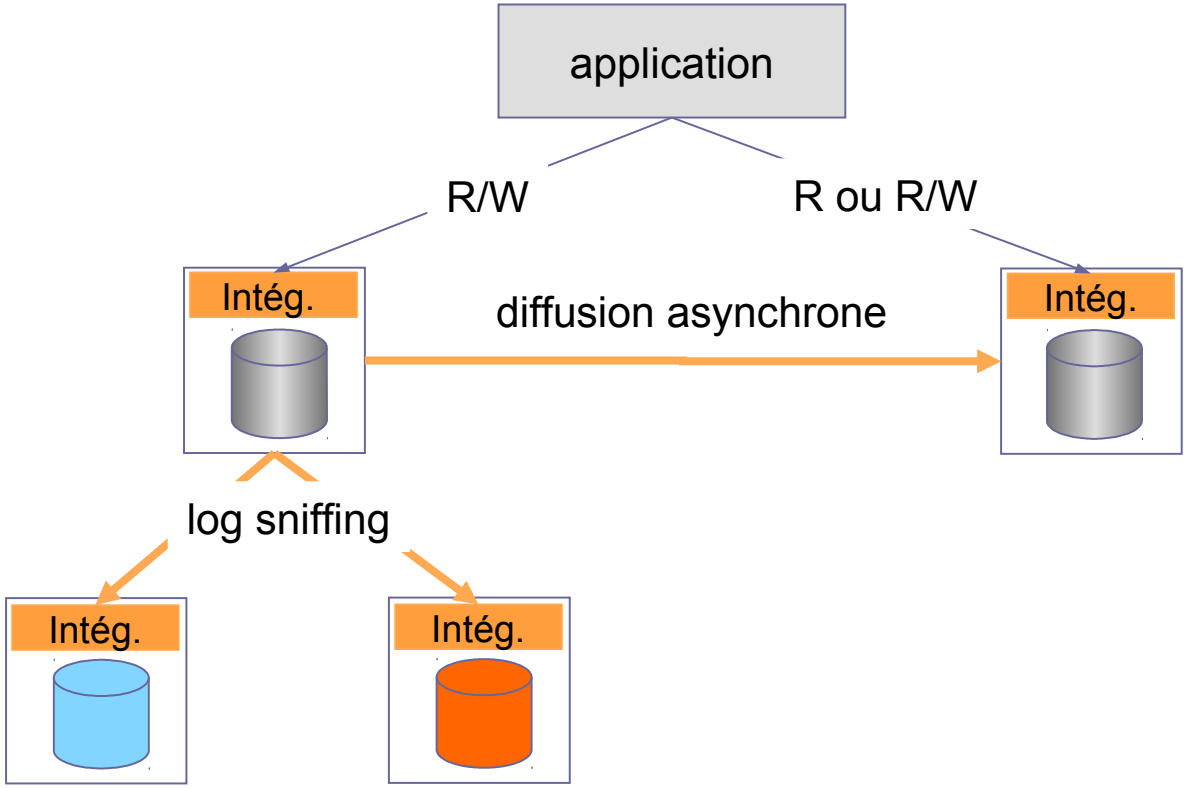
Déconnexions

- Plus une copie est déconnectée et effectue des mises à jour, plus il est probable qu'une réconciliation sera nécessaire

Nombre de copies

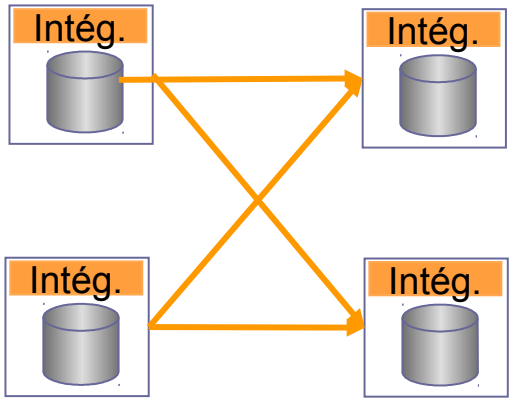
- Le volume de l'activité de propagation de mises à jour augmente avec le nombre de copies : si chaque copie effectue des mises à jour, l'effet sera quadratique

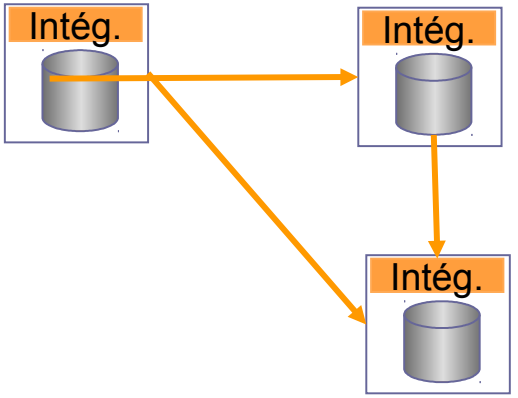
Architecture de l'infrastructure



Difficultés de la réplication

- Maintien du compromis performance – cohérence
  - Diffusion asynchrone
  - Gestion des règles de réconciliation
- Gestion des défaillances
  - Défaillances de réseau et de copies primaires mal gérées; nécessité de solutions applicatives
- Cohérence globale
  - Problèmes potentiels dans certaines configurations





Fonctionnalités d’intégration réplication

Fonctionnalité	**Réponse Réplication**
Définition de vues intégrées	Modèle relationnel – vues par fragmentation horizontale et verticale à partir des copies primaires. Droits d’accès
Langage de manipulation de données	Requêtes SQL de sélection (réplication asymétrique) et de mise à jour (réplication symétrique). Atomicité des mises à jour seulement dans le mode de diffusion synchrone
Interfaces applicatives	Idem SGBD

Règle d’urbanisation réplication

Caractéristiques données sources	SGBD relationnels homogènes Coopération forte entre sources
Caractéristiques données cibles	Données physiques Convergence mais cohérence faible des données – effort d’administration Transformation simples (union, jointure) Bonne disponibilité Bonnes performances d’accès
Coût	réglage performance/cohérence Gestion des défaillances Cohérence globale