Rappel sur les Transactions

Didier DONSEZ

Université Joseph Fourier (Grenoble 1)

IMAG/LSR/ADELE

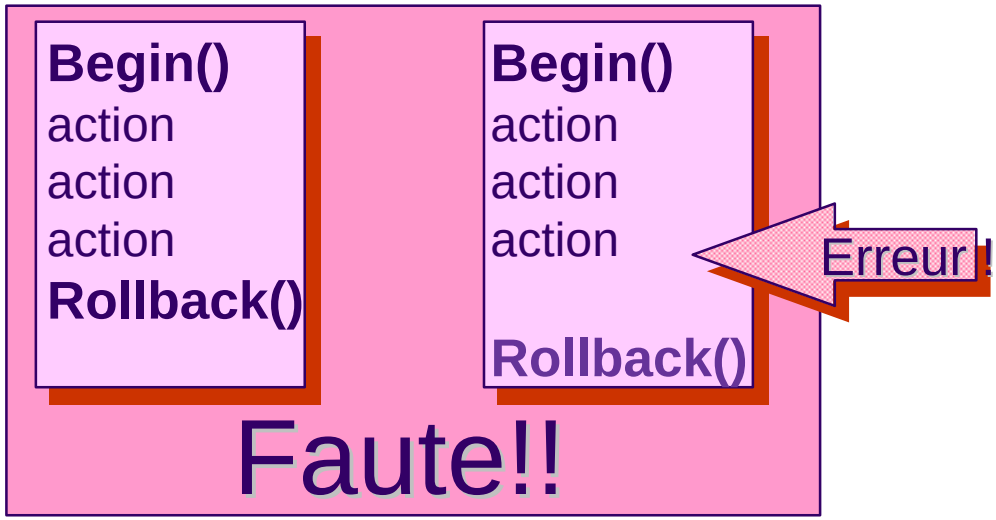
Didier.Donsez@imag.fr

Agenda

- Notion de Transaction
- Propriétés ACID
- Contrôle de concurrence
- Reprise sur Panne
- Transactions distribués
- Moniteur Transactionnel
- Modèle avancés

Rappel Fiabilisation
Notion de Transaction (Jim Gray)

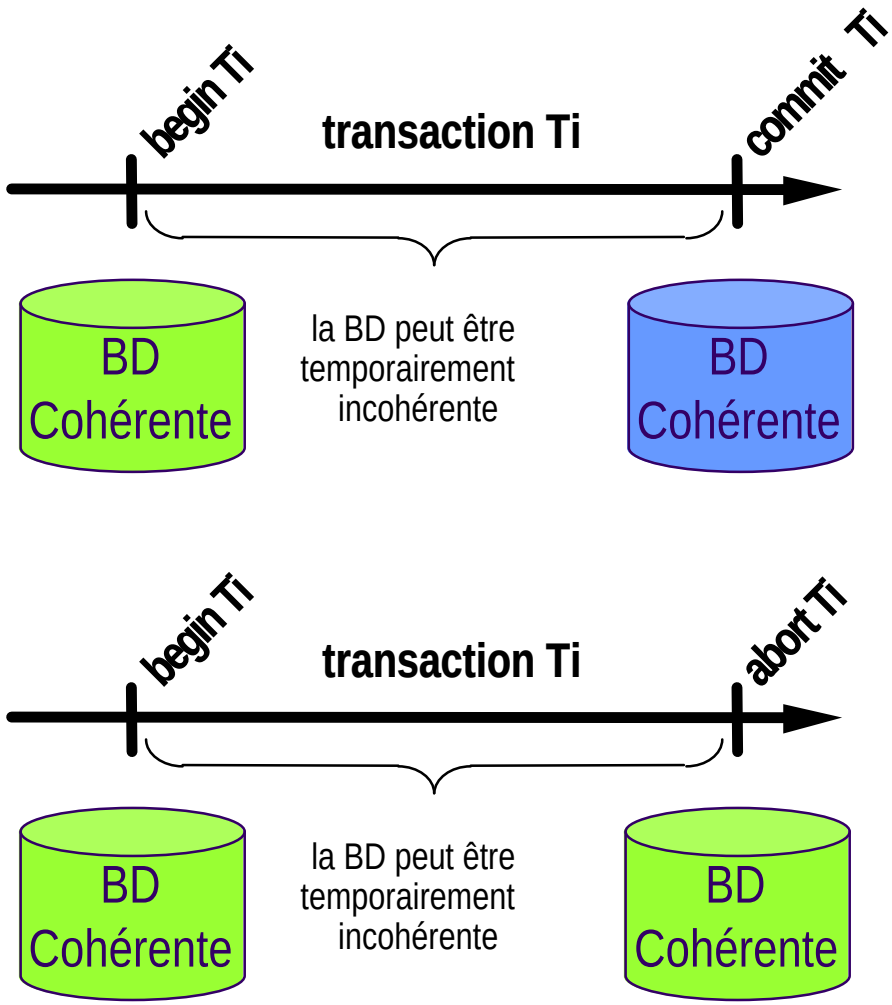
- **Pour le développeur**
  - une série d'actions delimitées par Begin et Commit/Abort.
- **Un modèle simple de panne**
  - seulement 2 devenir



Rappel Fiabilisation
Notion de Transaction

Propriétés ACID

- **A**tomicité
  - tout ou rien
- **C**onsistance
  - cohérence sémantique
- **I**solation
  - pas de propagation de résultats non validés
- **D**urabilité
  - persistance des effets validés



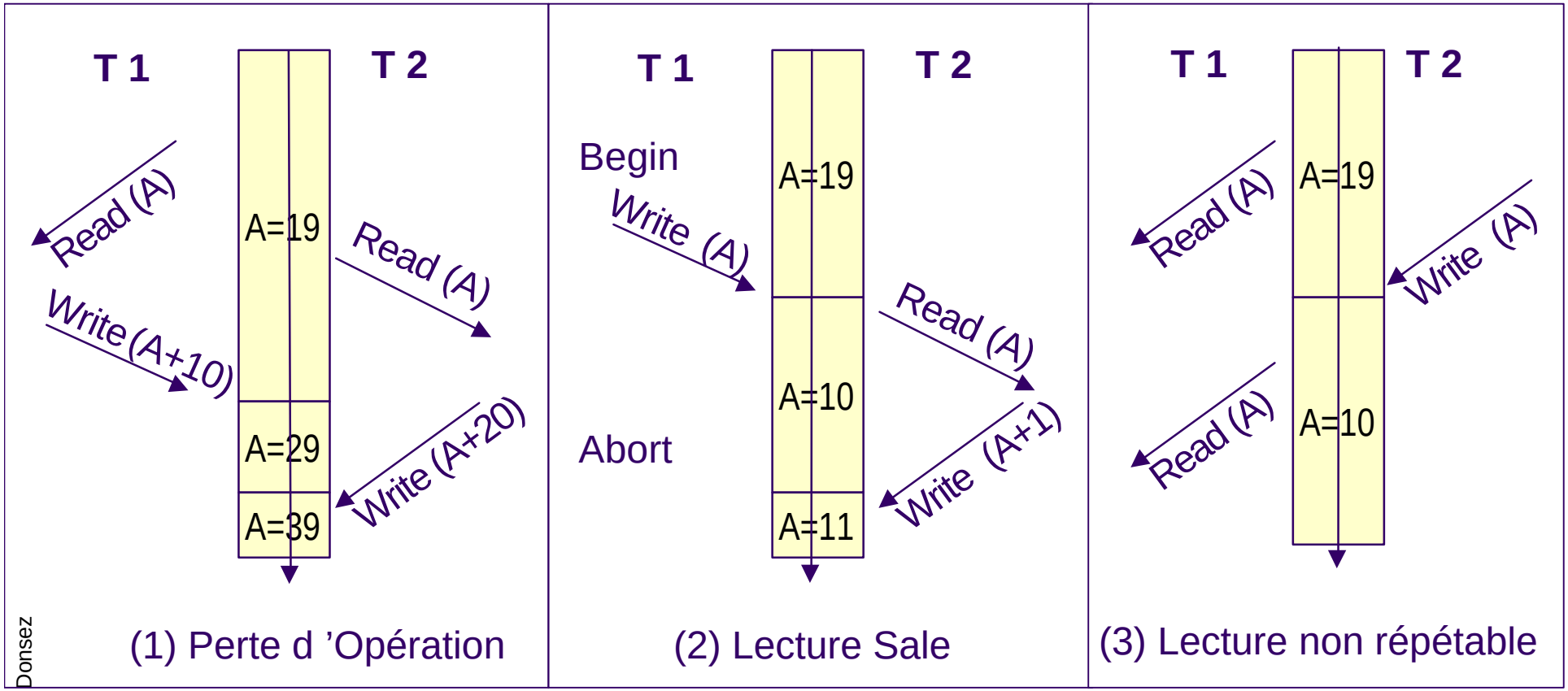
Usage des propriétés ACID

- **Systèmes d ’Information**
  - Bases de Données
  - MOM : Message Posting / Delevery
- **Systèmes d ’exploitation**
  - Systèmes de fichiers
  - Bases de registre
  - Installation/Déploiement logicielle (DLL, Composants)

Le contrôle de concurrence (I)

Notion de cohérence

➔ **Une exécution concurrente non contrôlée de plusieurs transactions crée des incohérences :**



Contrôle de Concurrence

- **Exécution sérielle**
  - T1 puis T2 puis T3 puis T4
  - T2 puis T1 puis T3 puis T4
  - …
- **Sérialisabilité**
  - entrelacement des actions des transactions tel que le résultat est équivalent à celui d ’une des exécutions sérielles
- **Méthodes**
  - Pessimiste : conflits probables
  - Optimiste : conflits peu probables

Exemple de Méthodes Pessimistes : le 2PL

- **Principe du Verrouillage à 2 Phases**
  - N lecteurs XOR 1 seul écrivain
- **A chaque accès à une donnée, un verrou est posé**

Compatibilité	Lecture	Ecriture
Lecture	OUI	NON
Ecriture	NON	NON

- **Les verrous sont tous relâchés à la fin de la transaction**
- **Inconvénient : risque d 'interbloquage (deadlock)**
- **Largement implanté par les SGBDs**

D'autres techniques

- **Estampillage (Pessimiste)**
  - Principes : utilisation de 2 estampilles L et M
  - Avantage :
    - Facile à mettre en œuvre, pas d 'interblocage
  - Inconvénients :
    - Fixe inutilement des dépendances entre les transactions
    - Famine: une transaction peut redémarrer indéfiniment
- **MVCC (Multi Version Concurrency Control)**
  - Estampillage et versions en lecture
    - Garantit la terminaison des transactions en lecture seule
- **Certification (optimiste)**
  - on laisse les transactions se dérouler, on contrôle à la fin
  - Avantage : test simple d'intersection d'ensembles
  - Inconvénient : effondrement si trop de conflits
- **Contrôle de concurrence sémantique (BD OO)**
  - Matrice de compatibilité sur des opérations de haut niveau
    - Exemple: ajout, retrait d'un élément dans un FIFO, …
- **Version et CheckIn-CheckOut**

Remarque

➔ **La cohérence forte (sérialisable) n'est pas toujours souhaitable !!!!**

- Blocage des transactions limite le débit transactionnel !

➢ Tableau de bord avec des données « scintillantes »

- Lecture non répétable

➔ **4 Degrés d'isolation (ISOLATION_LEVEL) en SQL**

Niveaux d'Isolation	Lecture Sale	Lecture Non Répétable	Fantômes
READ_UNCOMMITED	oui	oui	oui
READ_COMMITED	non	oui	oui
REPEATABLE_READ	non	non	oui
SERIALIZABLE	non	non	non

➢ + transactions read-only

La Reprise sur Panne (A,D)

Types de Pannes

- **Abandon d ’une transaction**
  - Peut être dû à un pb d’accès concurrent
- **Panne Système**
  - Tous les travaux sont brutalement interrompus (perte du contenu de la RAM)
- **Panne du support de reprise**
  - Il faut utiliser un support de sauvegarde

Exemple : La Journalisation (1/2)

- **Une Transaction est une séquence d'actions**
- **Chaque action peut :**
  - Modifier la base de données
  - envoyer un message
  - interagir sur les E/S
- **Chaque action doit laisser une trace dans le journal**
- **Pour défaire ou refaire une transaction il suffit de lire le journal**

Les journaux (2/2)

Enregistrement dans un journal

Plusieurs Variantes

- Journal des Images Avant / Journal des Images Après
- Journal Différentiel
  - N'enregistre que la différence entre l'image avant et l'image après
- Journal d 'Actions sémantiques (opération et opérandes)
  - Exemple : faire un closeup de 3 sec dans la video #987 au timestamp 1234

Problématique de la distribution

Validation d'une Transaction

Centralisée

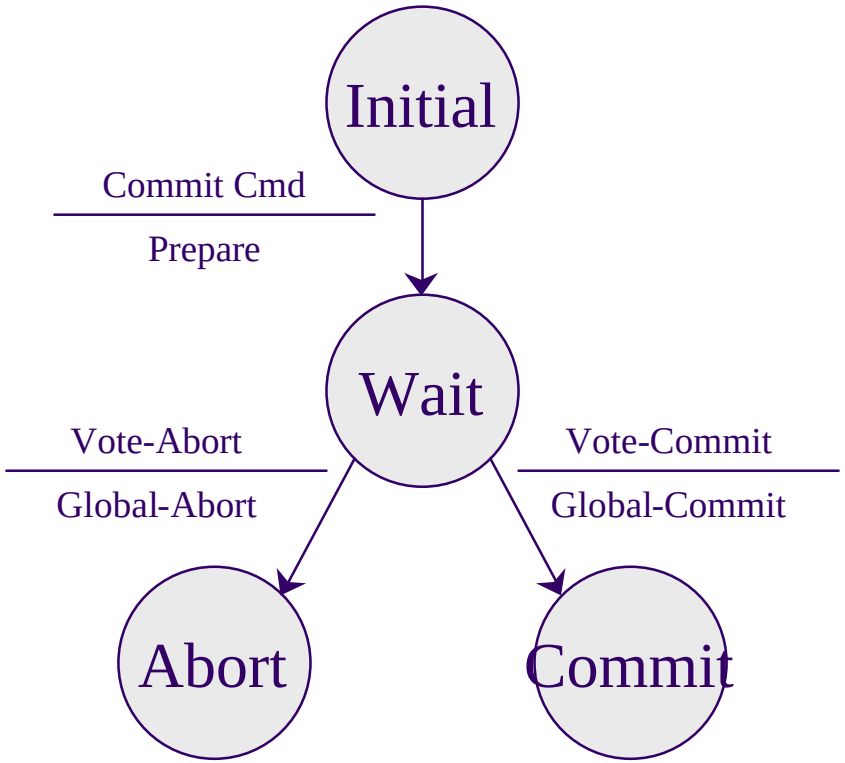
- l'application et les données sont sur la même machine
- Panne facile à traiter
  - Validation à 1 phase

Distribuée

- l'application et les données sont sur 2 à N machines
- Il faut que tout les participants prennent la même décision
- Panne (partielle) difficile à traiter
  - Validation à 2 phases (2PC : Two Phases Commit)
    Classique , Abort implicite, Commit implicite
  - Validation à 3 phases (3PC : Three Phases Commit)

Validation à 2 Phases

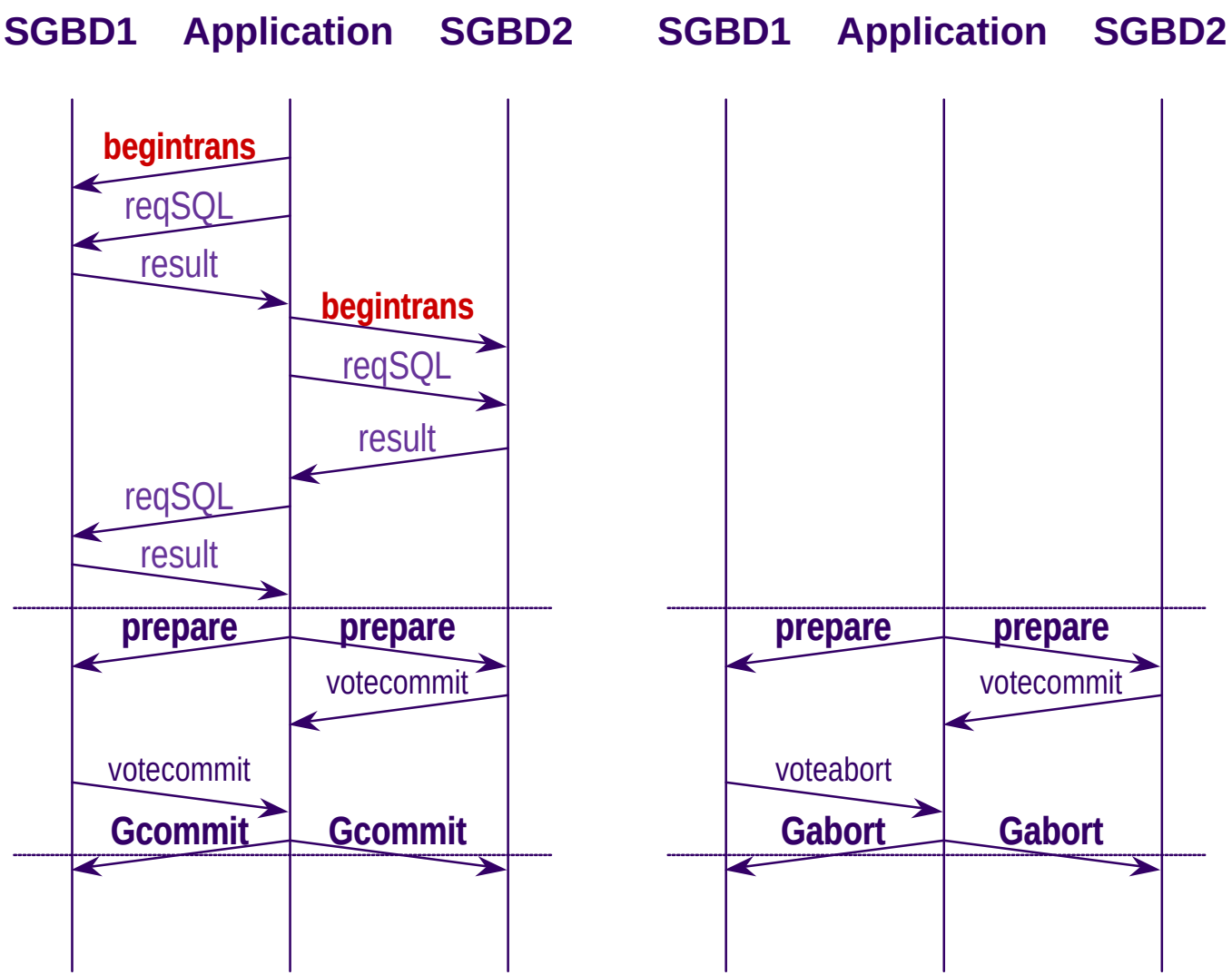
➔Coordinateur



➔Participant(s)

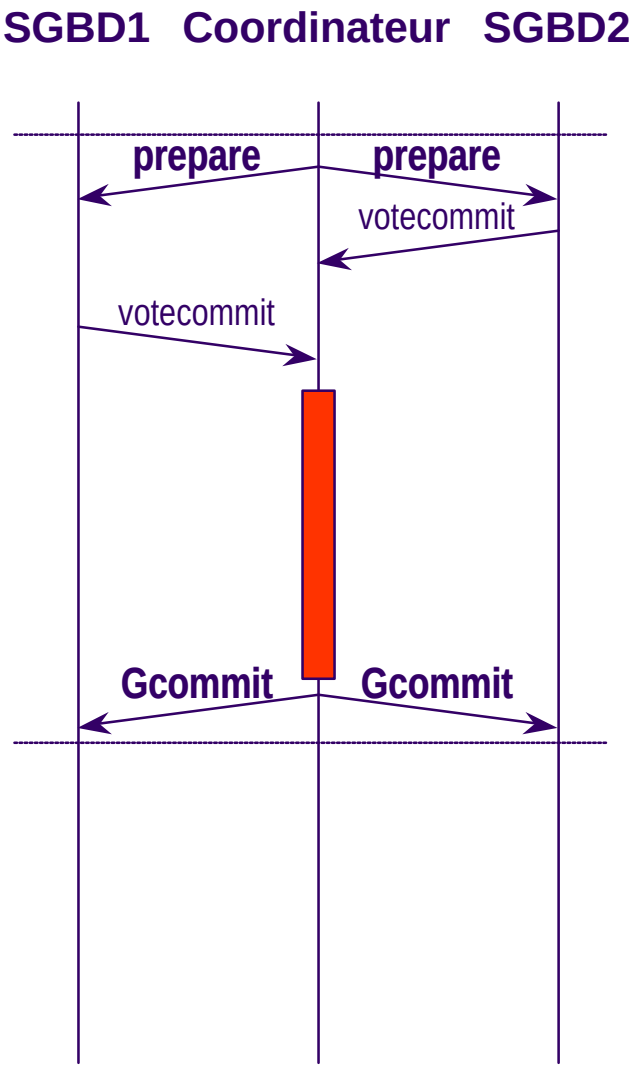
de Dider Donsez, Imag, Univ. J. Fourier

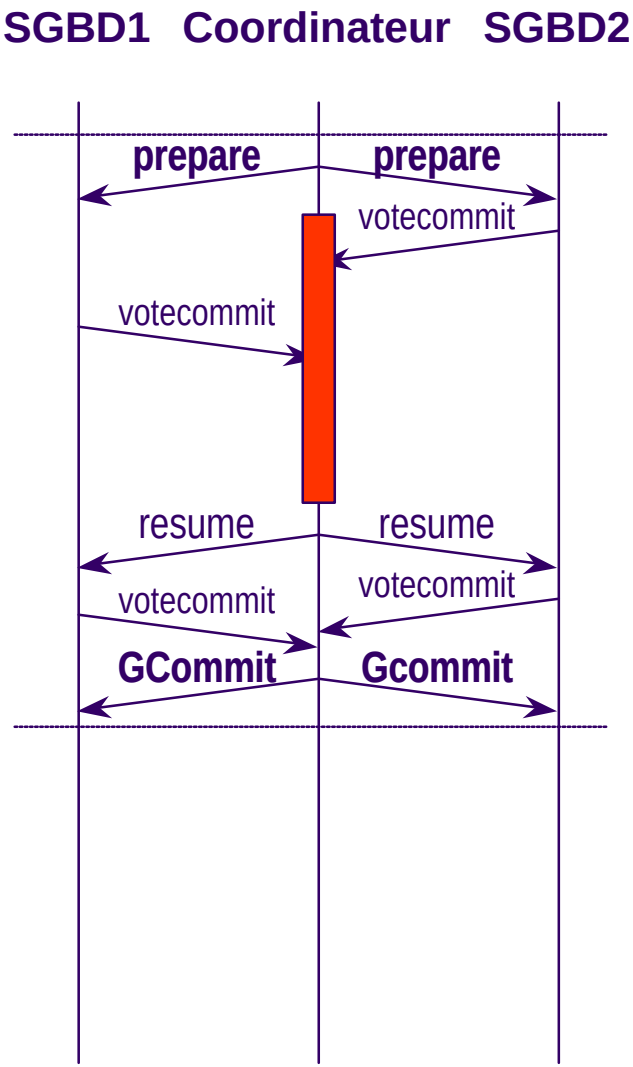
2PC - Hors Panne



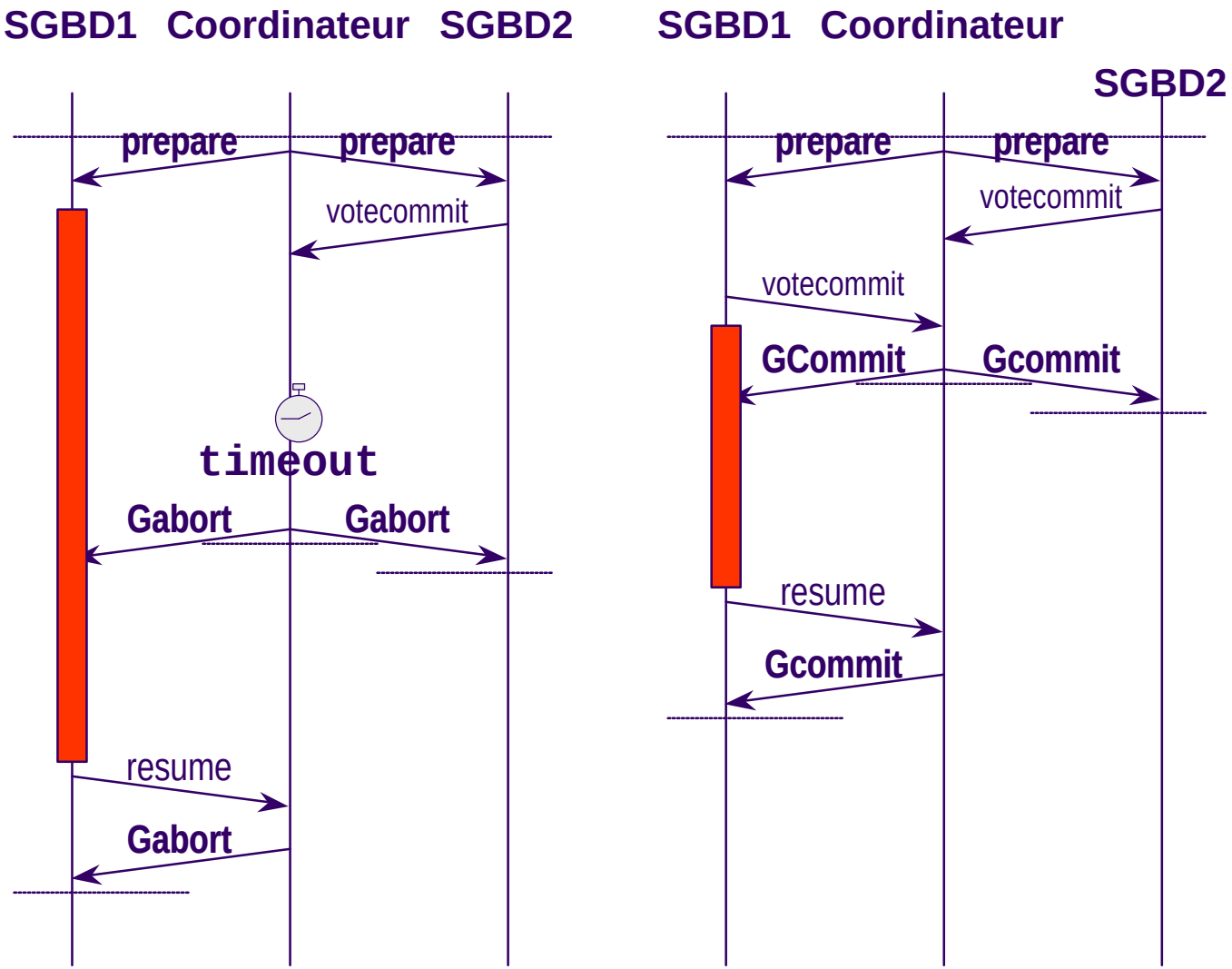
Didier Donsez

2PC - Panne Coordinateur





2PC - Panne Esclave



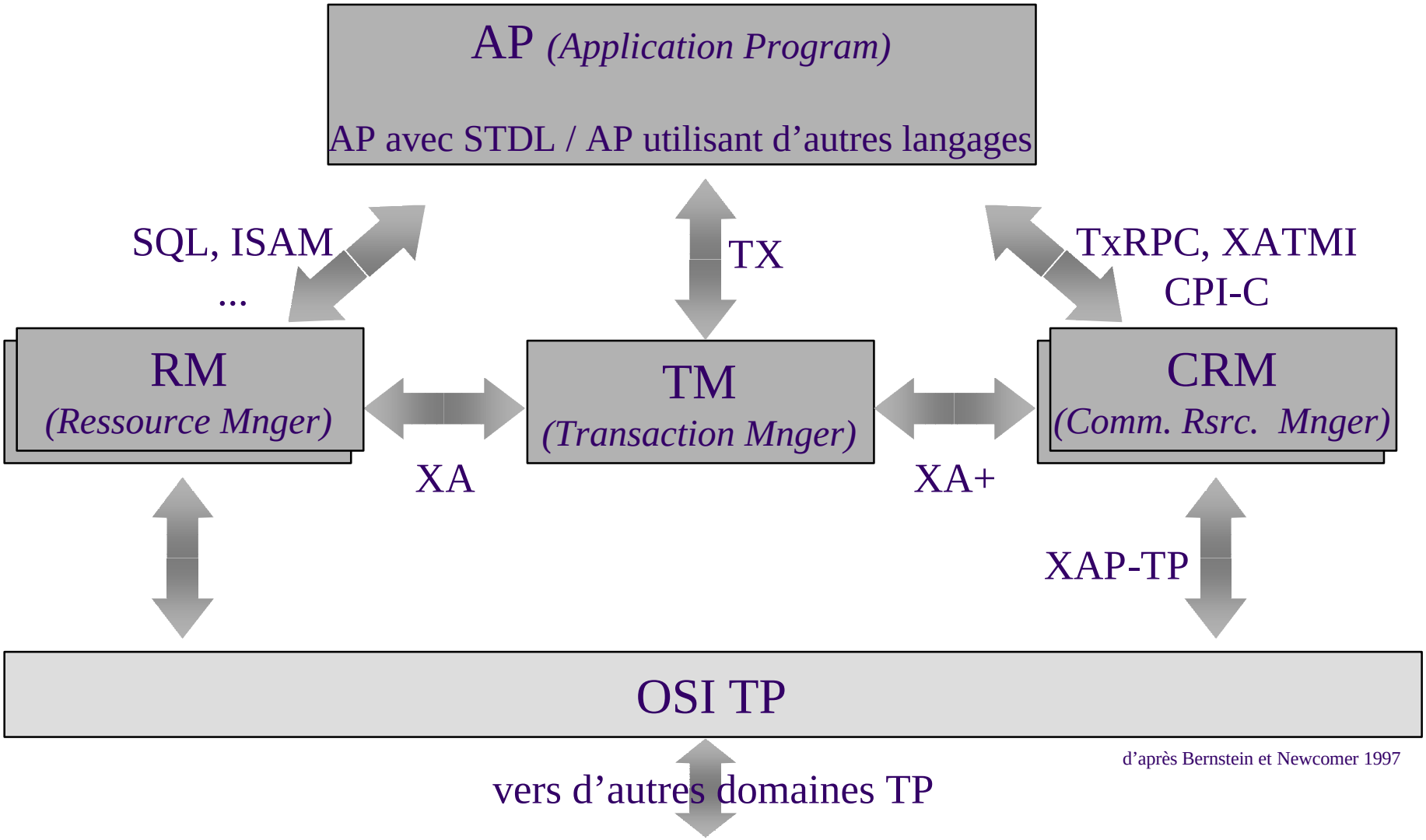
Didier Donsez

Moniteurs Transactionnels

Rappel Fiabilisation
Moniteur Transactionnel *(TP Monitor)*

- **Pilote l'exécution distribuée de transactions globales sur des ressources distribuées**
  - Coordination de la validation (dit à 2 phases)
- **Les Protocoles Standards**
  - X/OPEN DTP (Distributed Transaction Processing)
    - Plusieurs interfaces : TX, XA, CRM, XA+, RM, XAP-TP
  - OSI/TP
  - OMG/OTS (Object Transaction Service)
- **Avantages**
  - Accès hétérogène aux RM (Resource Manager)
  - Haute disponibilité et Hautes Performances
    - SABRE, SOCRATE, NYSE, ATM, …
    - Configurations matérielles logicielles à 700000 transactions TCP/C par minute (02/2002)
  - Equilibre de charge
- **Orienté OLTP (OnLine Transaction Processing)**
  - Traitement « simple », *Thinking time* court, …

Le Modèle DTP de l'X/Open



d'après Bernstein et Newcomer 1997

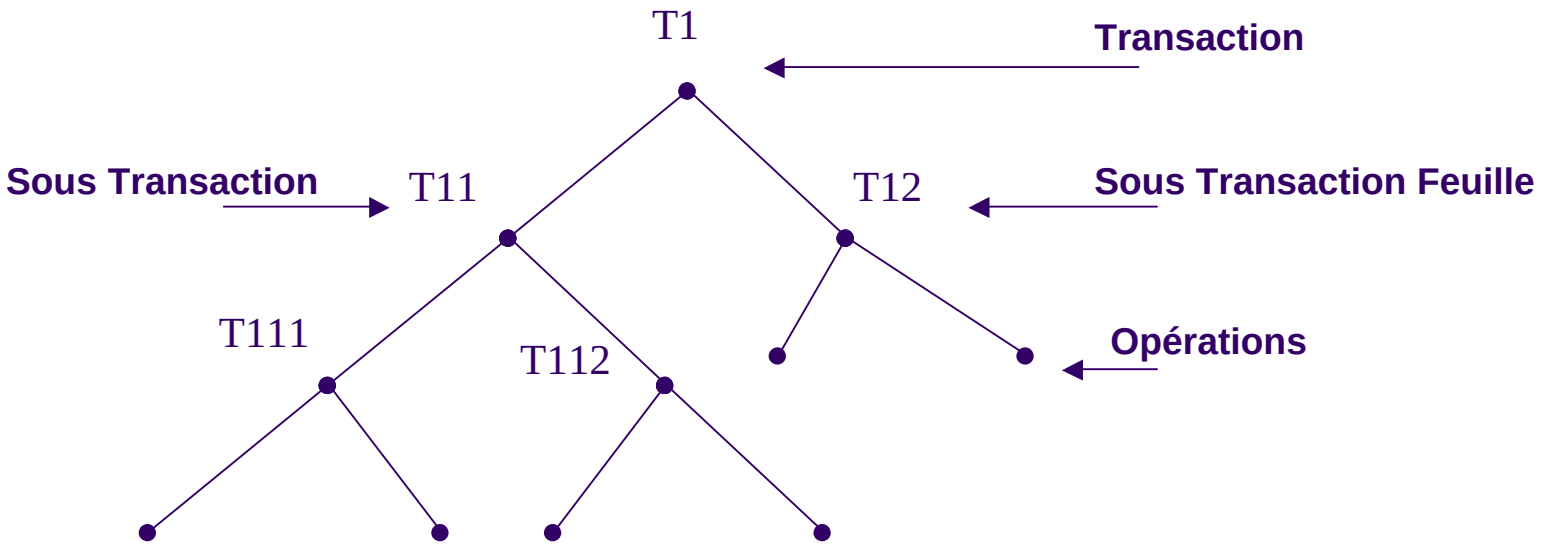
Modèles de Transactions

Les transactions Plates

- **Un seul niveau de transaction**
- **Bien adapté aux transactions en lignes (On Line Transaction Processing : OLTP)**
  - Manipule peu de données
  - De courte durée
  - Des centaines de transactions simultanées
- **Des carences importantes :**
  - Atomicité contraignante sur des transactions très longues
  - Isolation est un frein à l ’exécution de transaction coopérantes

Les Transactions Emboîtées (1)

➔ **La transaction n'est plus monolithique : elle contient des sous transactions**



Les Transactions Emboîtées (2)

➔ Règles de fonctionnement :

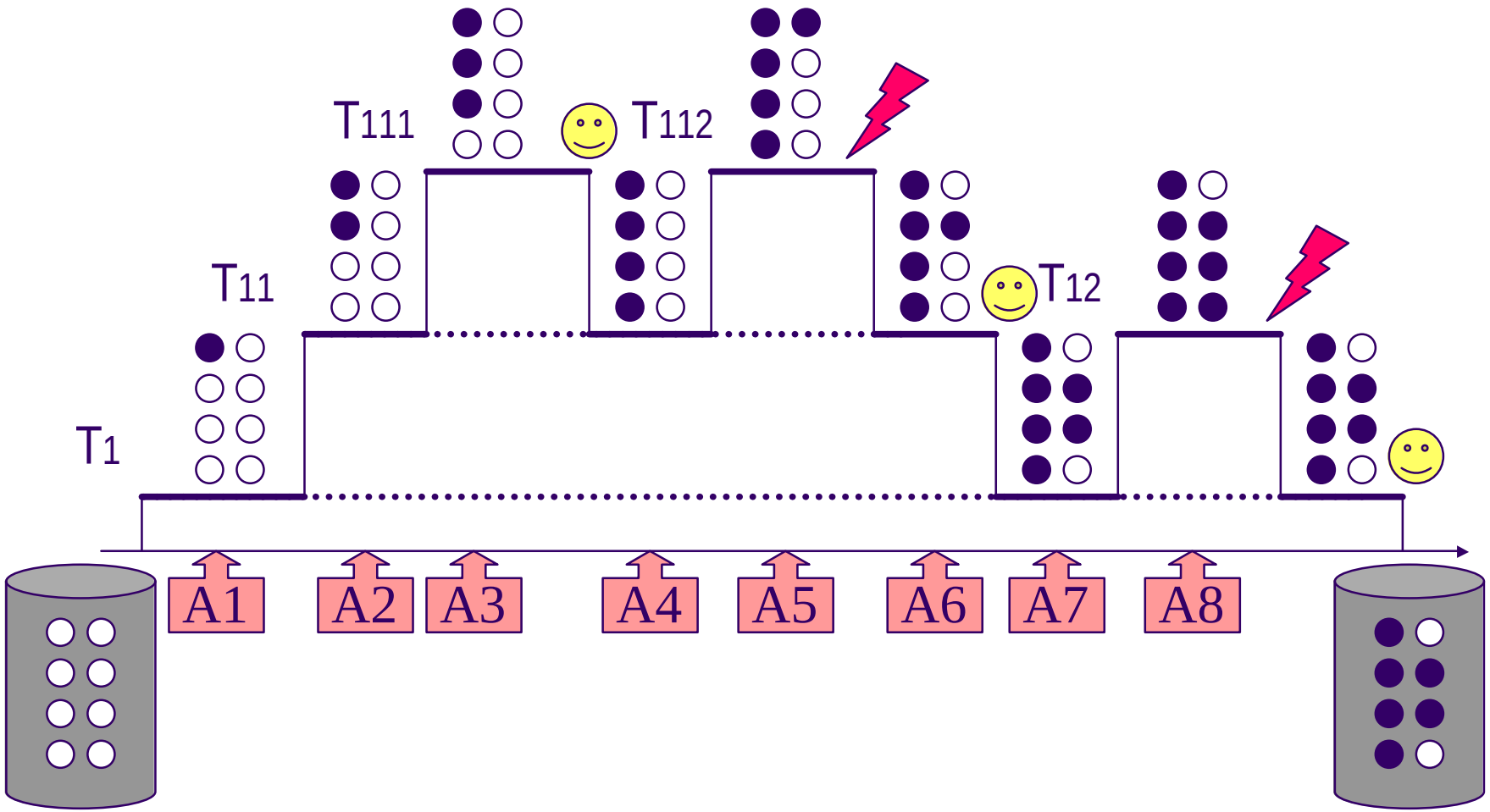
- Une sous transaction démarre après la transaction mère et se termine avant elle

- L ’abandon d’une sous transaction entraîne :
  - l ’abandon de ses sous transactions descendantes,
  - pas nécessairement l ’abandon de ses ancêtres

- La validation d’une sous transaction est conditionnée à la validation de sa transaction mère

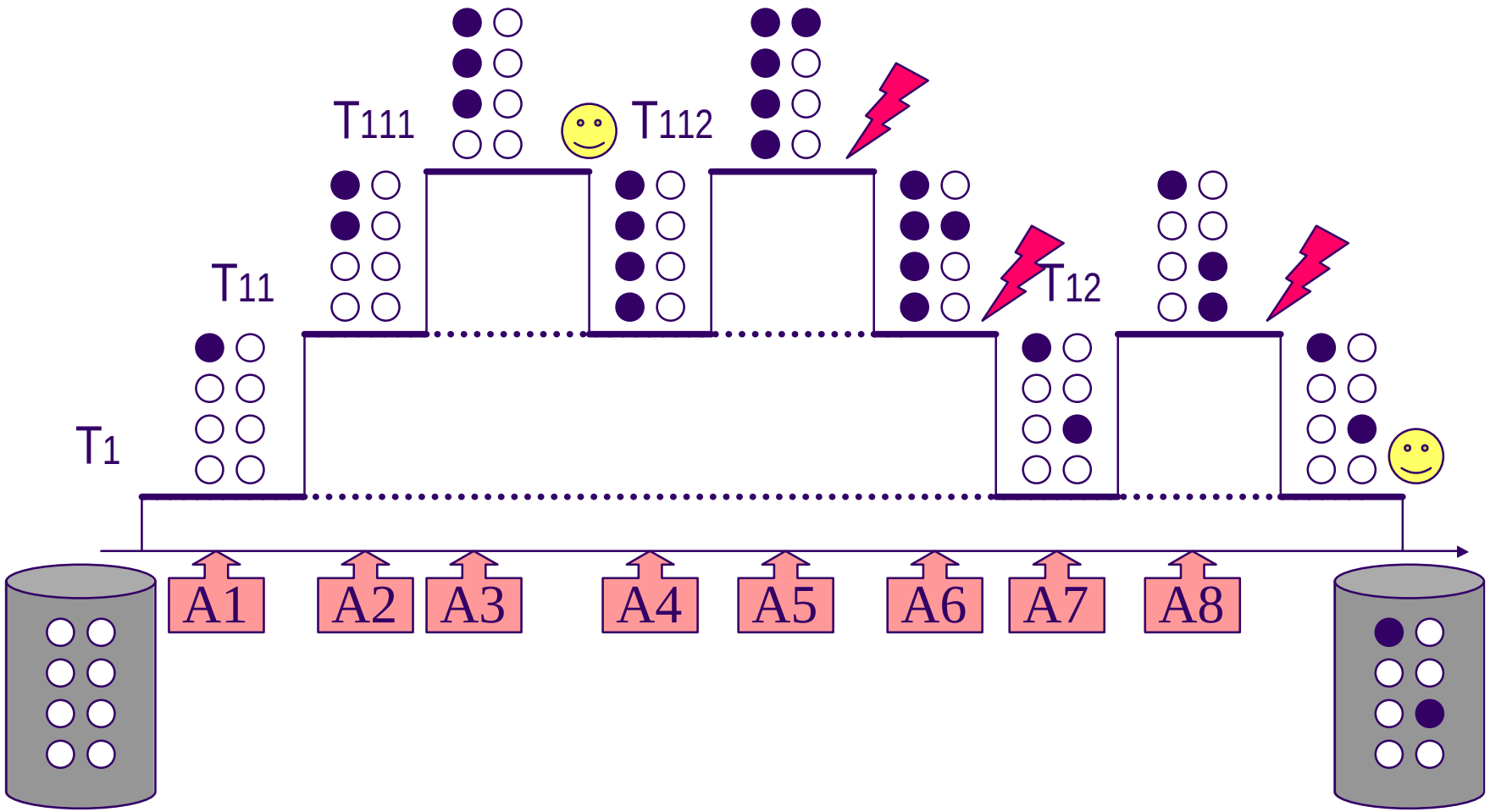
- Une sous transaction est atomique et isolée de toutes les sous transactions qui n ’appartiennent pas à sa descendance

CNT - Example



De Didier Donsez, Imag, Univ. J. Fourier

CNT - Example



De Didier Donsez, Imag, Univ. J. Fourier

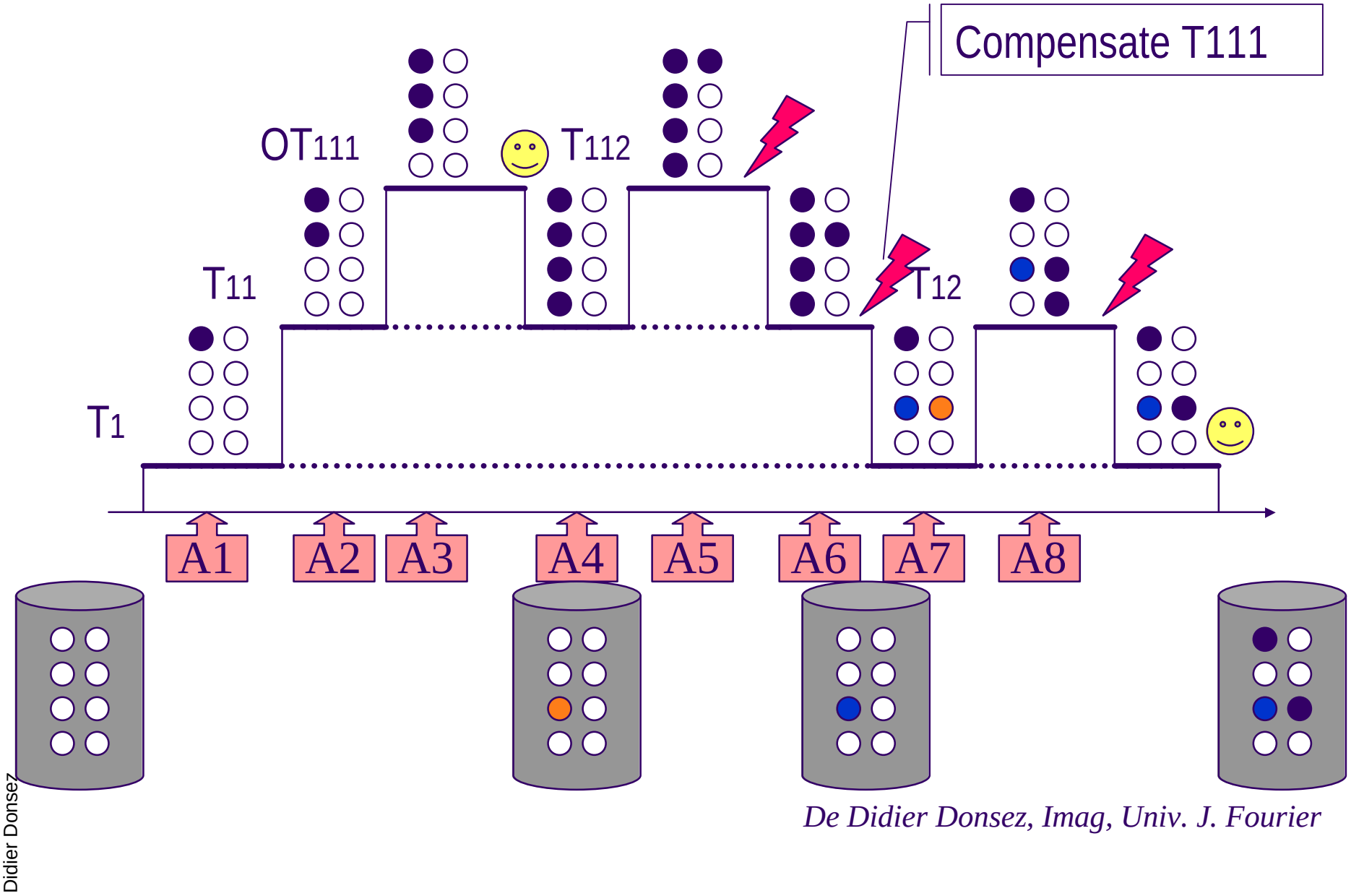
Les transactions emboîtées ouvertes

➔Modèle de transaction multi-niveaux

- ➢Règles de validation des ONT
  - ●Après la validation d'une ONT, toutes les mises à jour sont globallement visibles par les transactions racine.

- ➢Règles d'annulation de la transaction parent
  - ●Une transaction de compensation est déclenchée pour défaire les effets de l'ONT

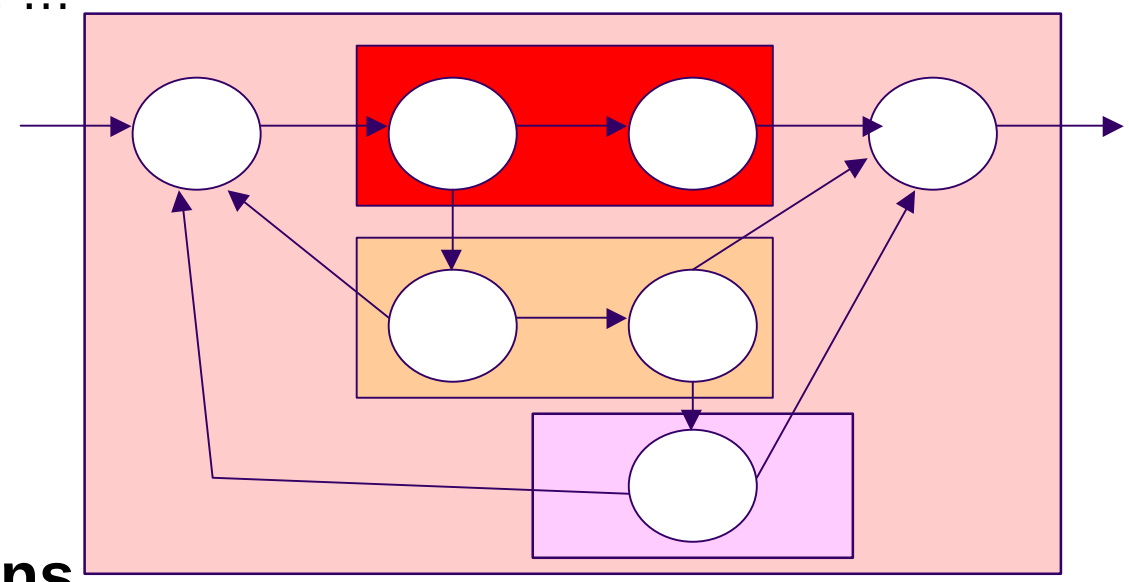
ONT - Example



De Didier Donsez, Imag, Univ. J. Fourier

D'autres modèles de transactions ...

- **Transactions longues**
  - Sagas, Contract, Flex, …
- **Workflow *reliable***



- **Business Transactions**
  - Cohesion et Actions
  - Les esclaves du 2PC negocient un delai de garde pour éviter le blocage par le coordinateur
- **Remarque**
  - ACTA : un modèle formel