

Thèse de Doctorat de l'Université P & M Curie

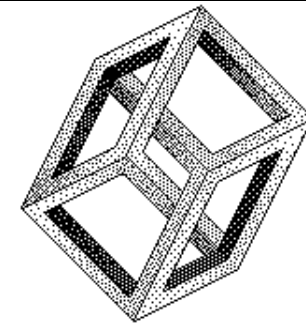
# WEA

Un Gérant d'Objets Persistants  
pour des environnements distribués



Université Pierre et Marie Curie  
Paris VI

Didier Donsez



Laboratoire de Méthodologie et Architecture  
des Systèmes Informatiques

---

# Plan de l'exposé

- les Objectifs de WEA
- le Modèle des Workspaces
- la Modélisation d'architectures de GO
- une application : le Travail Coopératif
- Implémentation
- Perspectives

---

# Rappel

Qu'est ce qu'un objet ?

✓ un état

une donnée simple ou complexe

✓ un comportement

typage

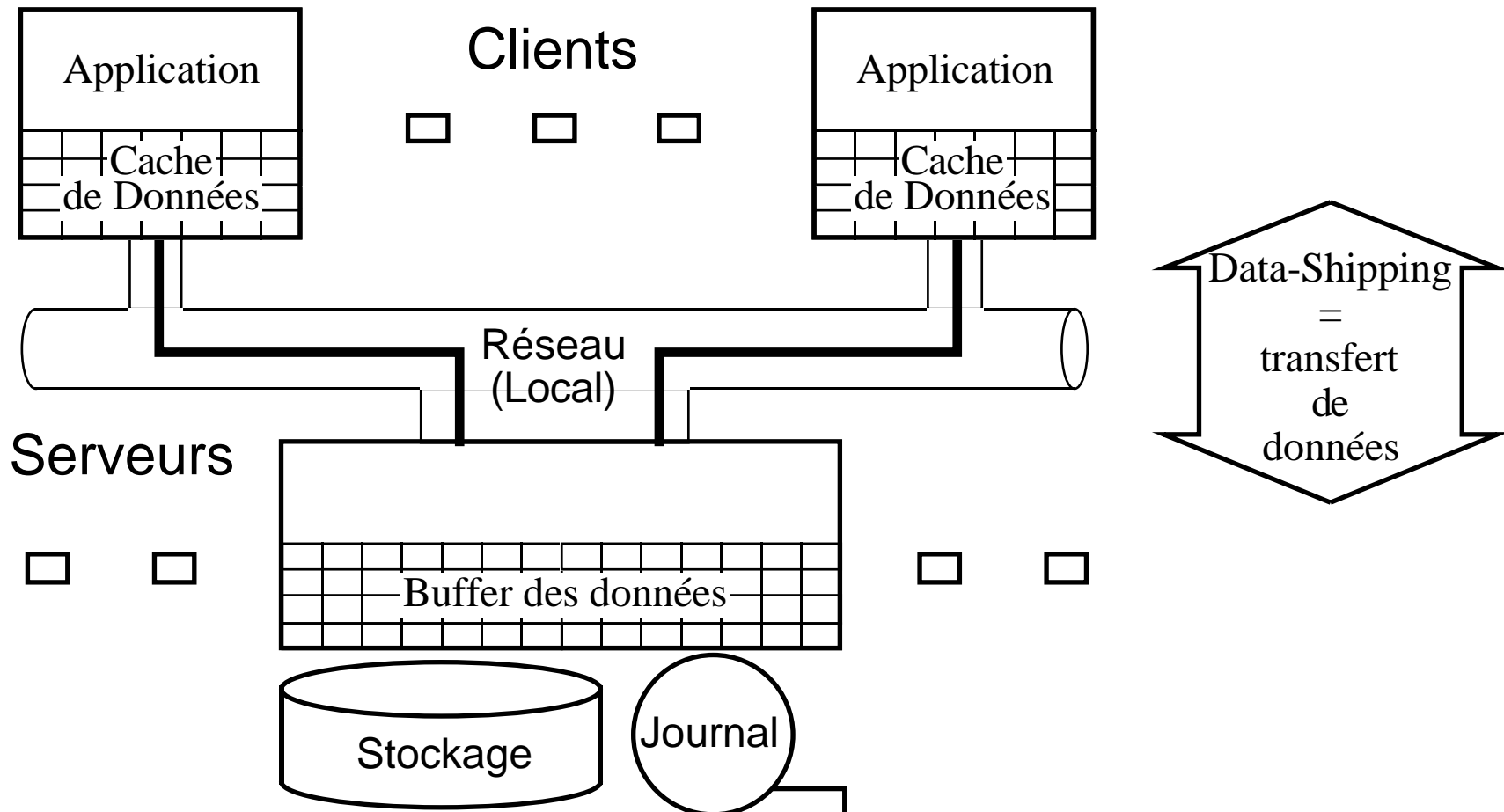
interface d'accès

= un ensemble de fonctions manipulant l'état

Qu'est ce que la persistance ?

une propriété d'objet d'exister après la terminaison du programme qui la crée

# Architecture d'un Gérant d'Objets Persistants (GOP)

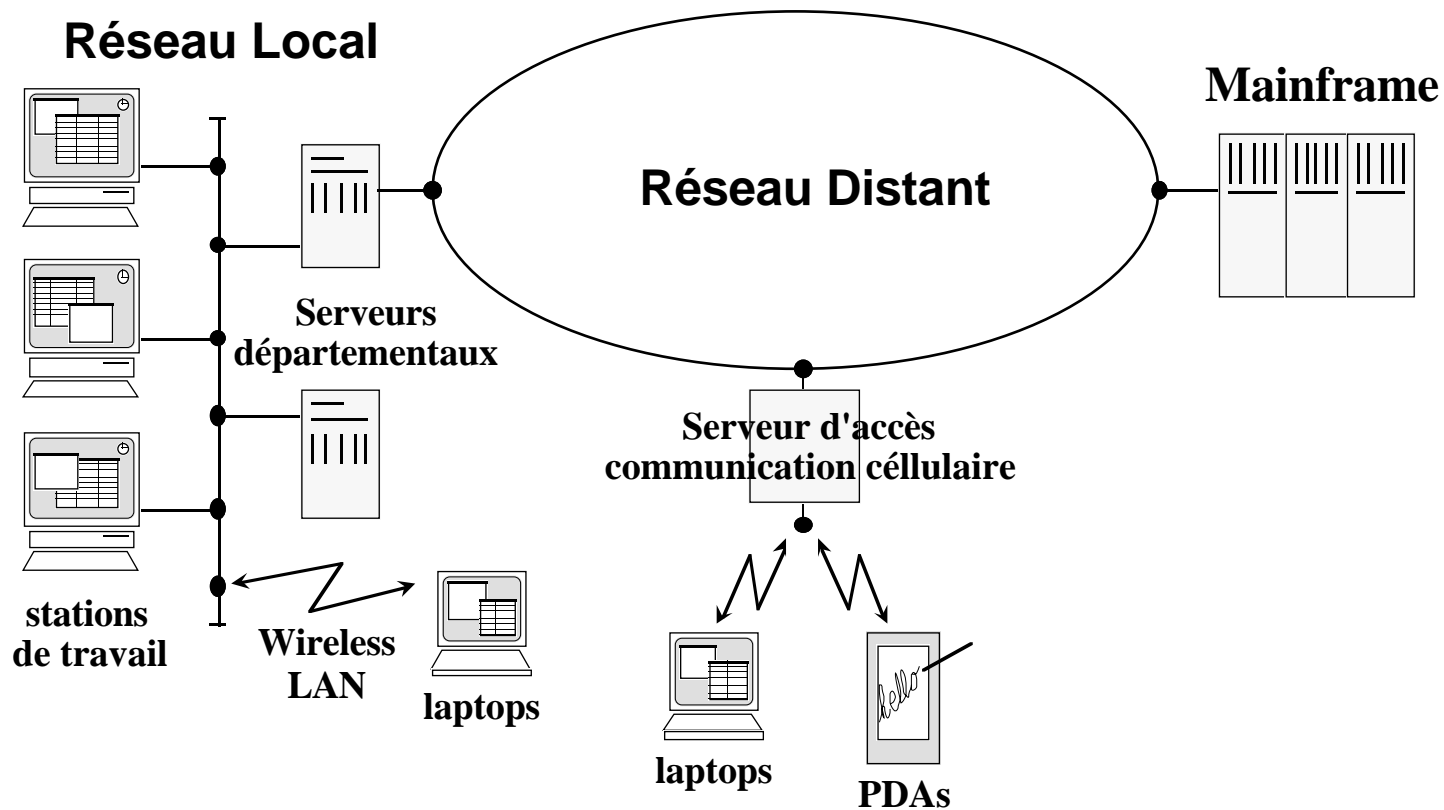


# Objectifs de WEA

## s'adapter à un système informatique hétérogène

Réseaux : débit, coût, fiabilité

Machines : puissance, mémoire, stockage



---

# Objectifs de WEA

## s'adapter aux besoins des applications

### ✓ Data-Shipping

Maximise l'utilisation des ressources des clients

### ✓ Query-Shipping

Sécurité (client non sécurisé)

Déléguer les calculs sur les sites stockant les données

*Informatique mobile*

### ✓ Nouvelles applications

Travail Coopératif

---

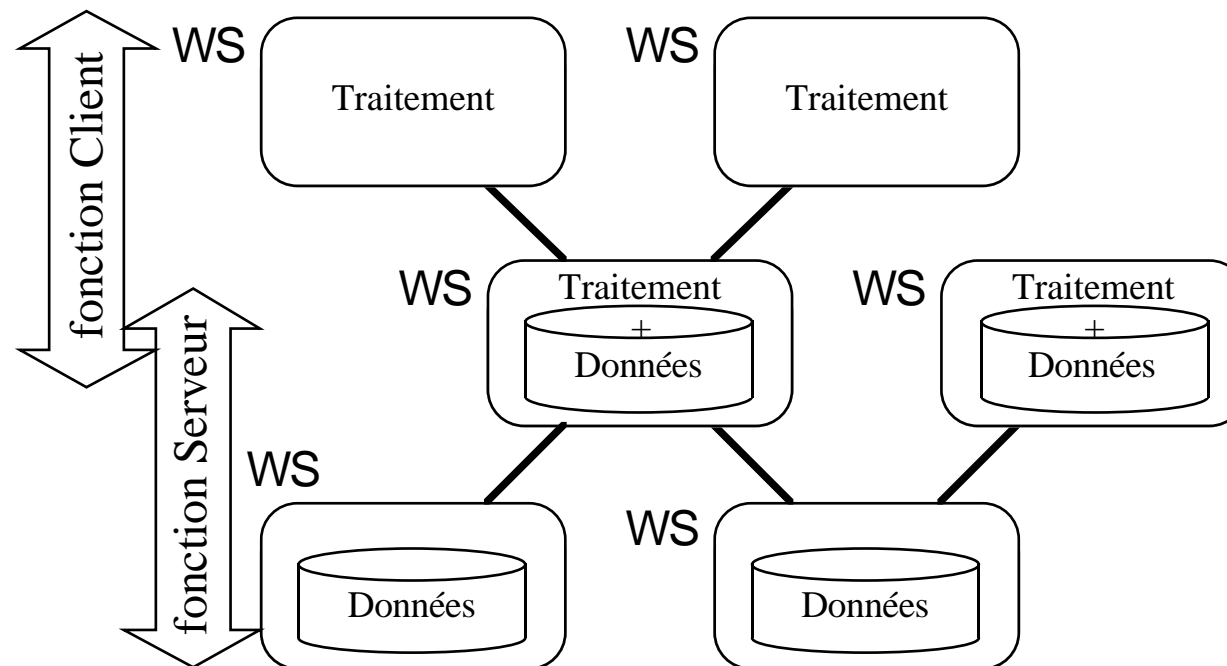
# le Modèle des Workspaces

- un noyau générique de Gérant d'Objet

fonctions du client et du serveur

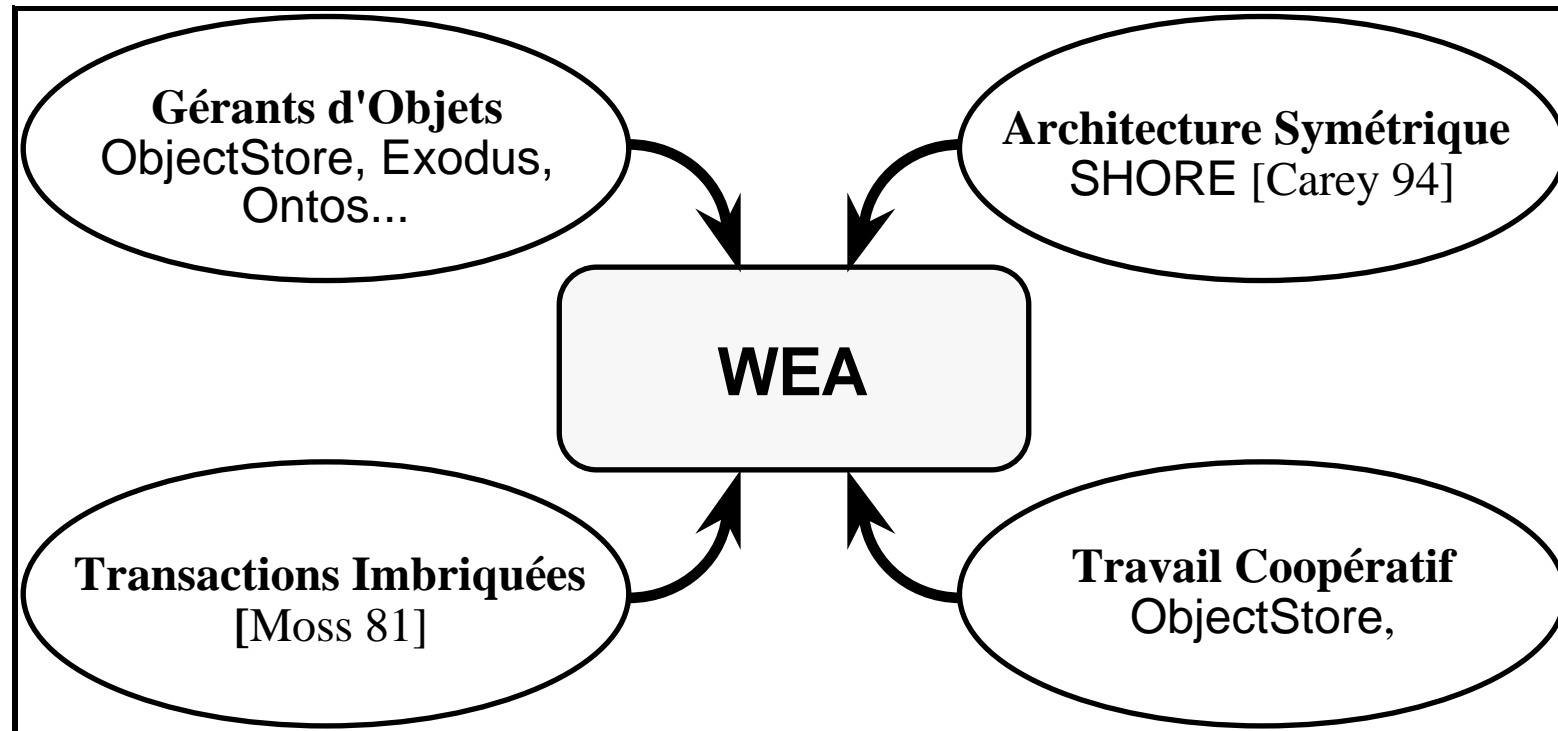
- une mécanisme de communication basé sur les services

imbrication par des services récursifs



---

# Travaux les plus Voisins





---

## **le Workspace : un noyau de GOP générique**

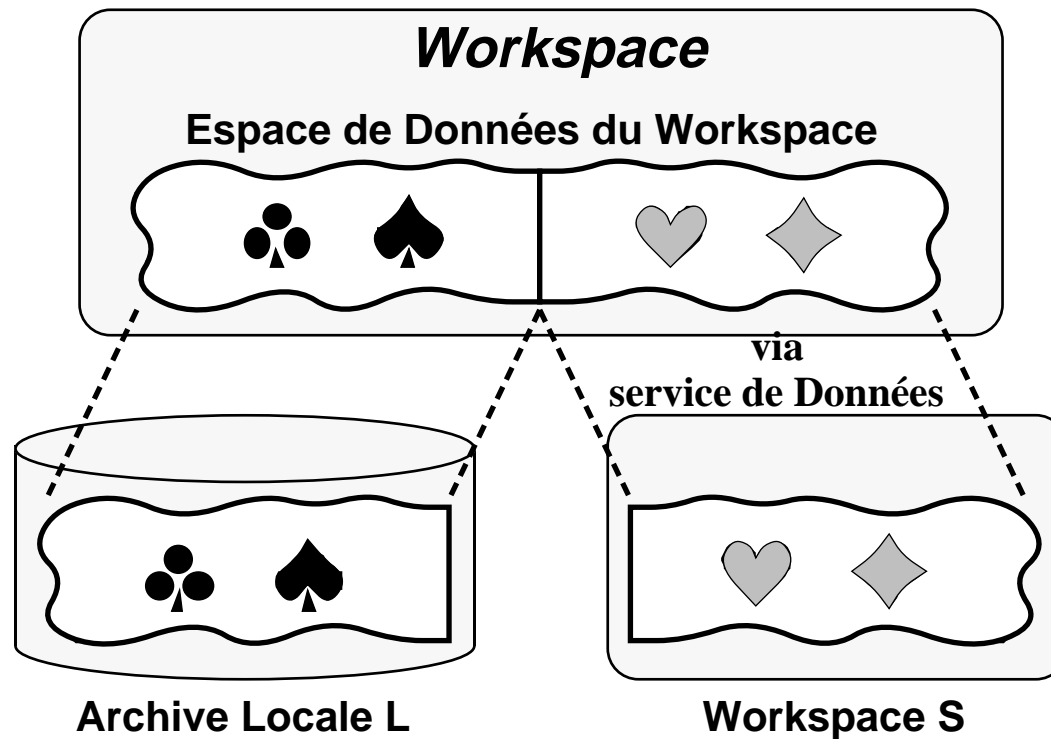
- un espace de données
- un ensemble de transactions
- un ensemble d'activités
- une propriété de validation des transactions

# le Workspace

## un espace de données

fusion { d'une vue sur une base locale  
d'une vue sur un autre espace de données

- Transparence  
à la localisation
- Récursivité



# le Workspace

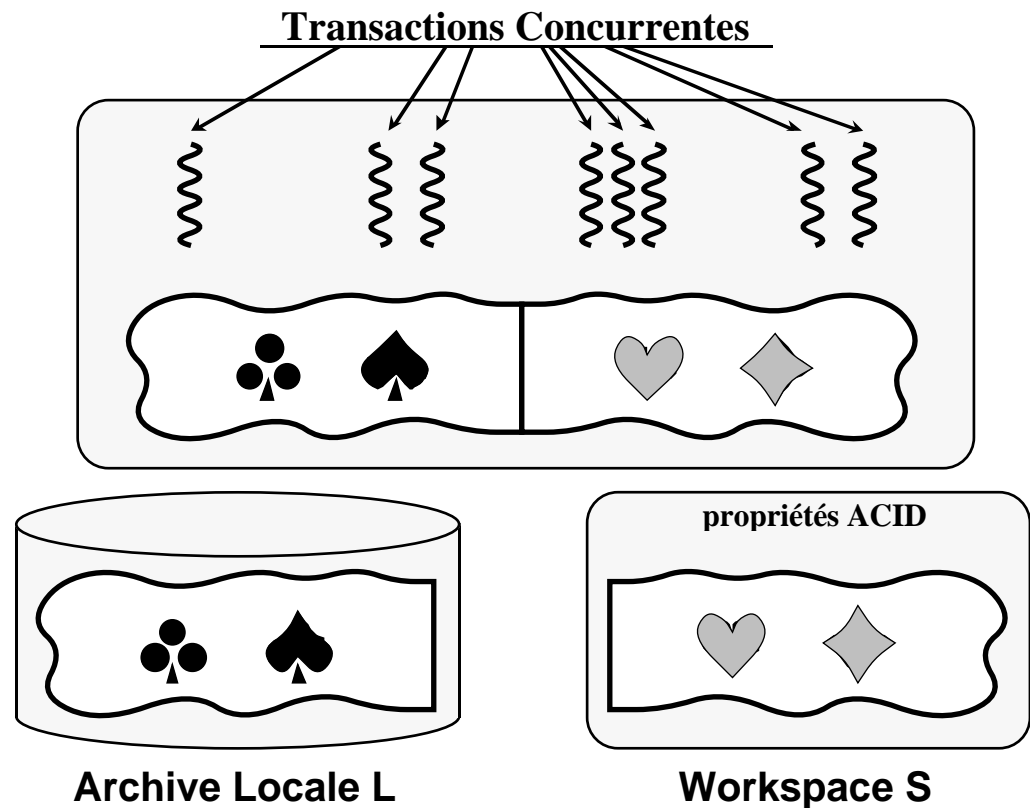
## un ensemble de transactions

Modèle d'exécution  
transactionnelle

Propriétés ACID

Réalisation

- ✓ des applications
- ✓ des services

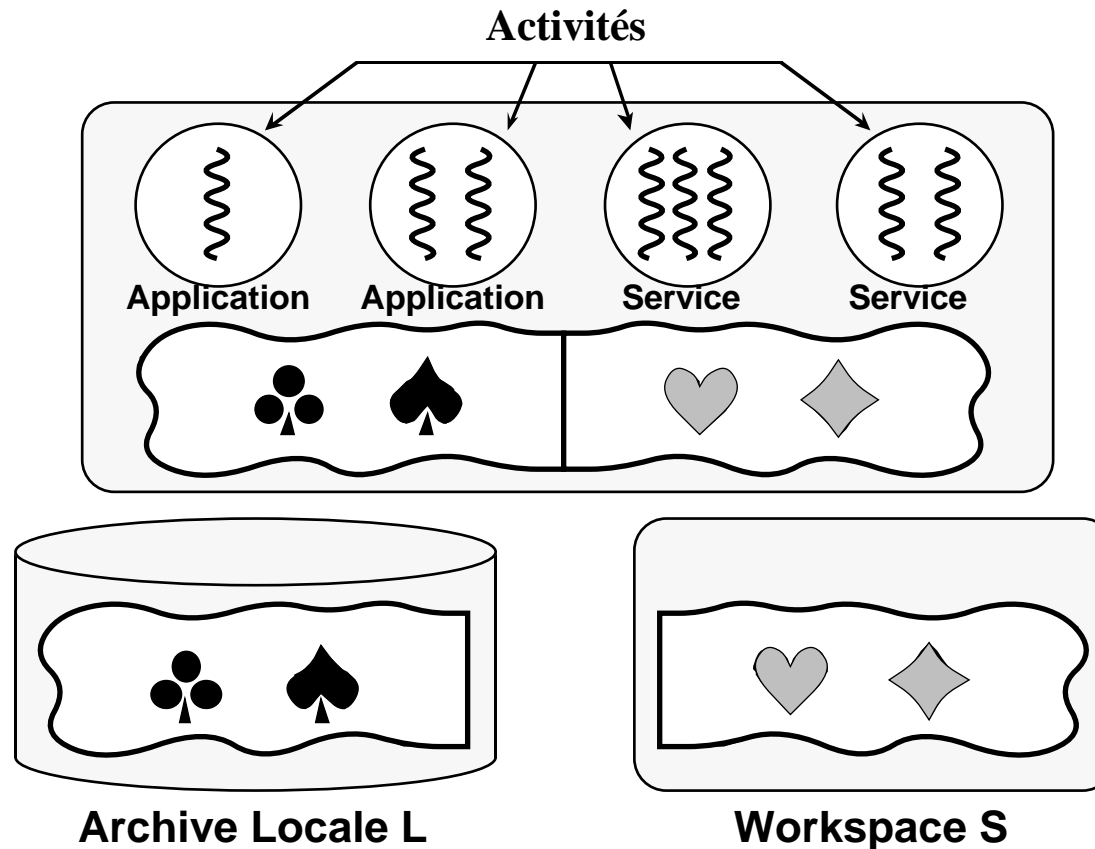


# le Workspace : un ensemble d'activités

activité = groupe de transactions participant à la réalisation

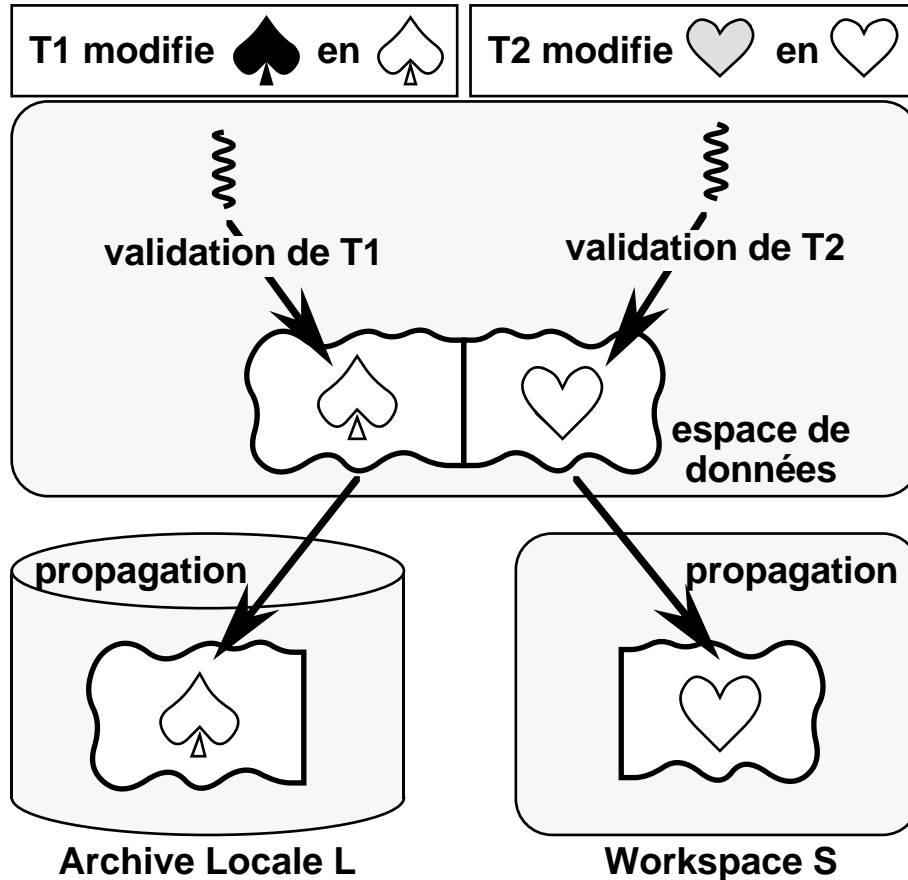
✓ d'une application

✓ à un service au profit d'un client



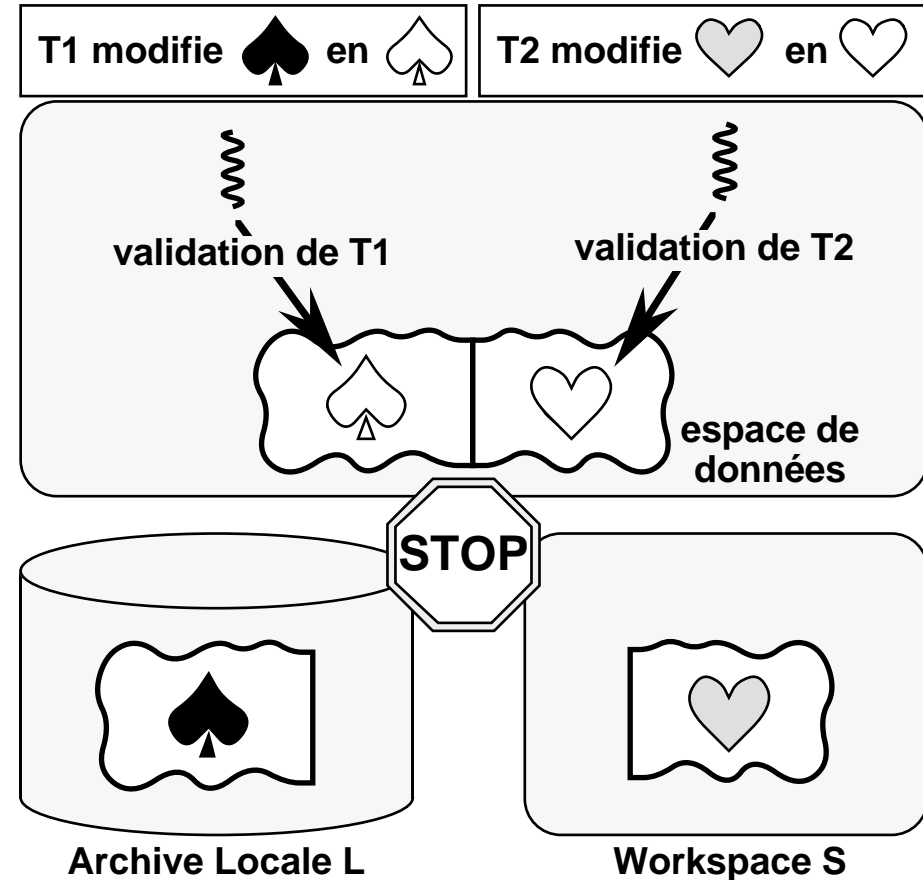
# le Workspace : Validations des Transactions

propriété passante



Cache de Données

propriété englobante



WS = Transaction Englobante

# Workspace et Services

## le modèle de communication

service = offrir un accès à une ressource privée

### Publication - Abonnement

#### ① publication du service

↳ le WS devient Serveur

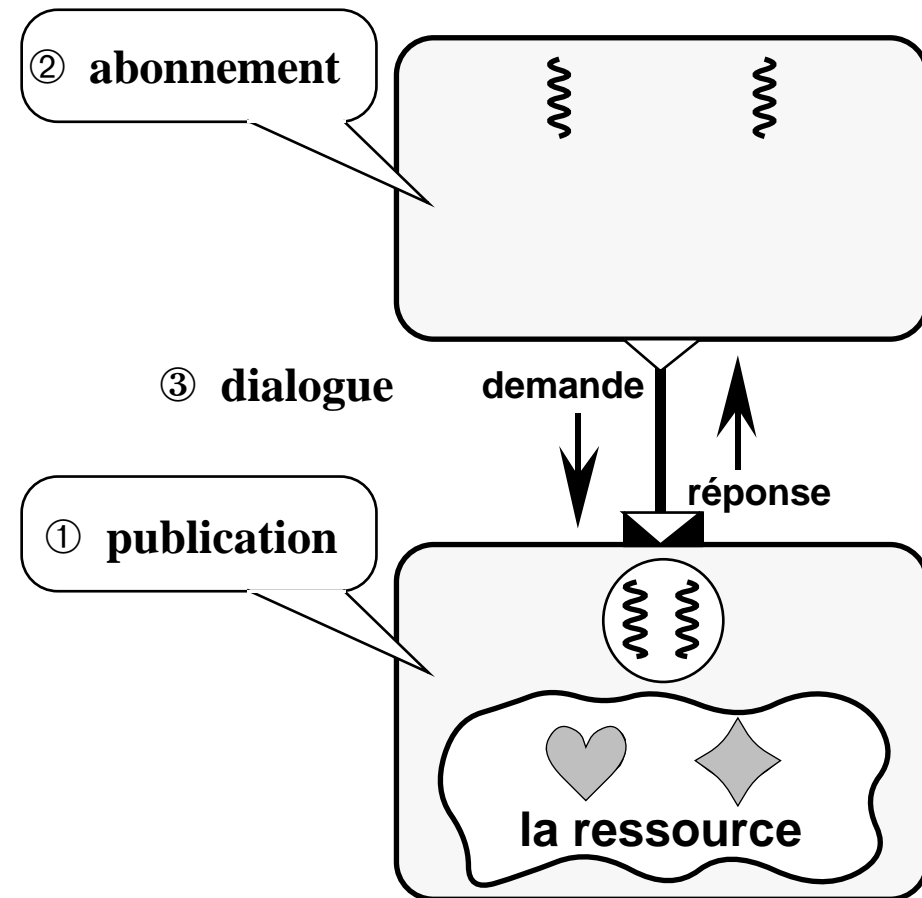
#### ② abonnement au service

↳ le WS devient Client

#### ③ dialogue

↳ le Client demande

↳ le Serveur répond



---

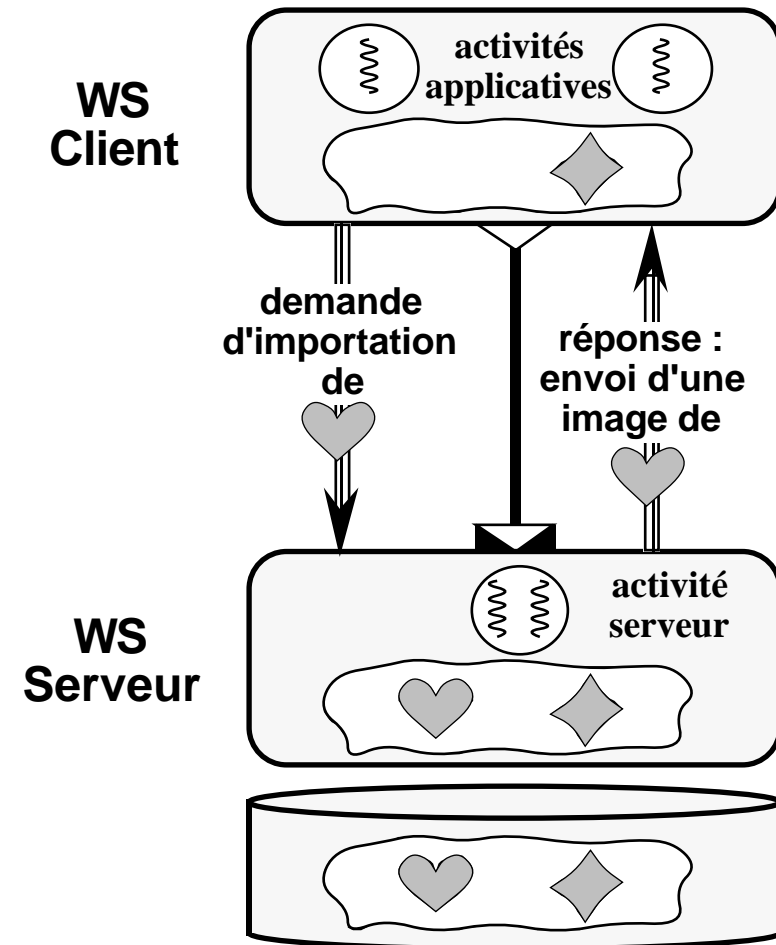
## Workspaces et Services

- service de Données
- service d'Opérations
- service Mixte de Données et d'Opérations
- service Coopératif

# Service de Données

## Echange des Données entre le Client et le Serveur

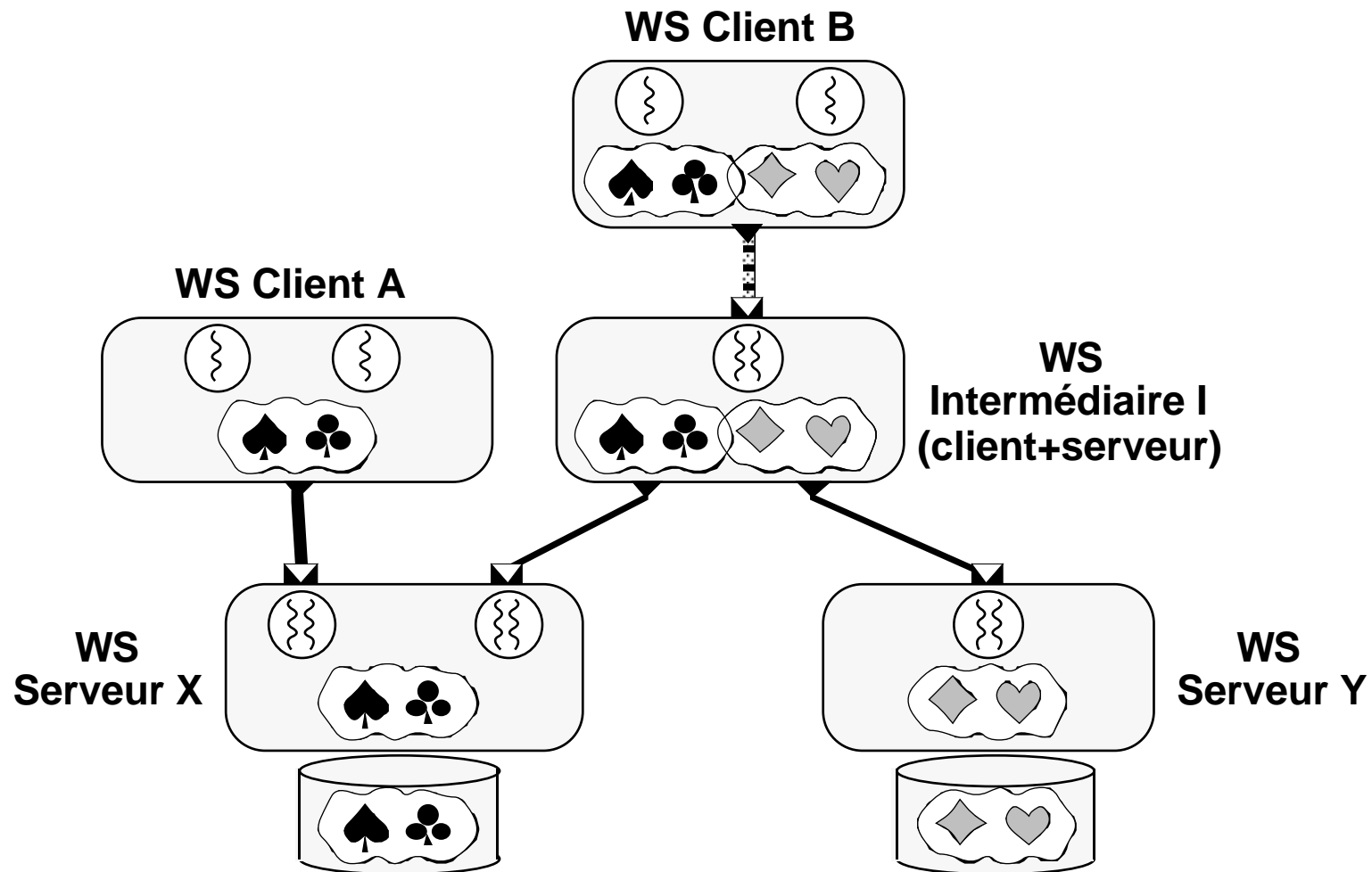
- ↗ chargement des données demandées
- ↘ redescente des données modifiées lors de la validation





# Service de Données - la Récursivité

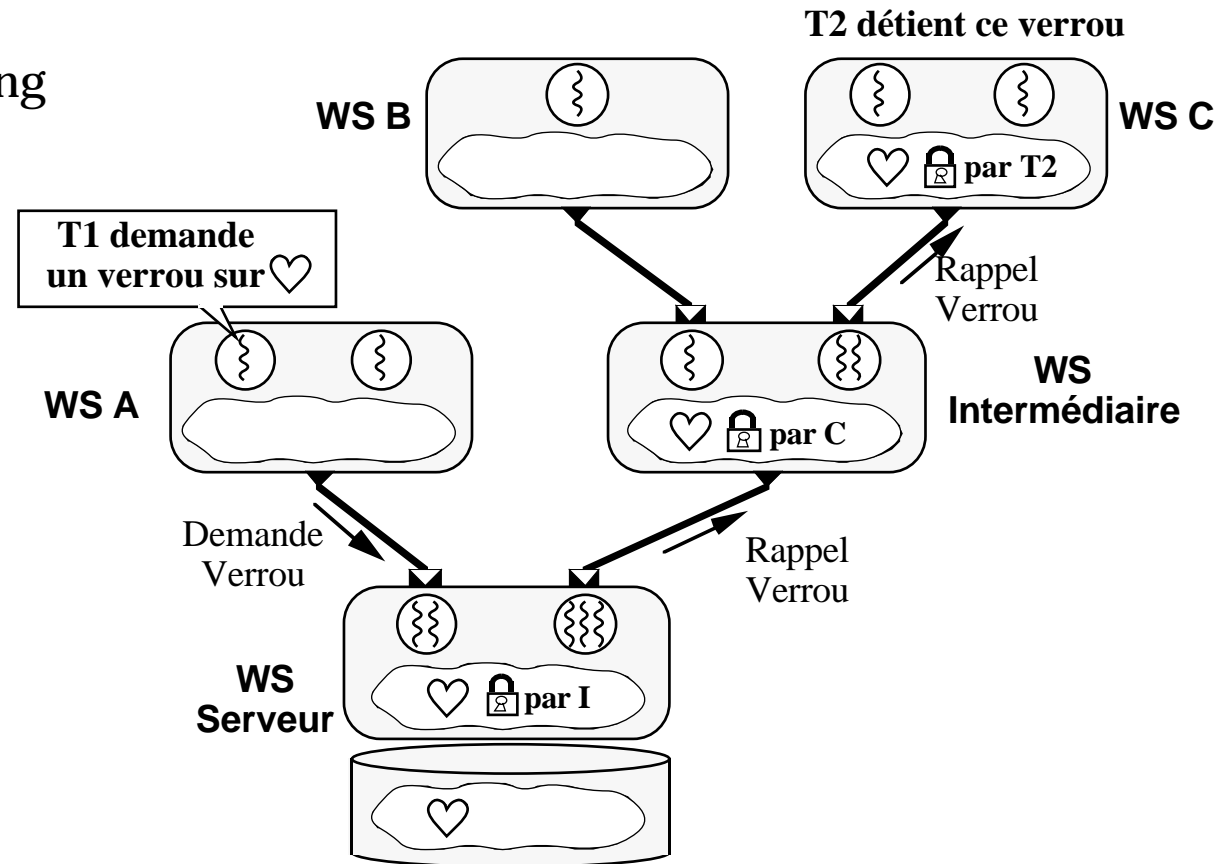
hiérarchie de clients et de serveurs



# Service de Données

## hiérarchisation du Contrôle de Concurrency

Callback Locking  
[Wang 1991]

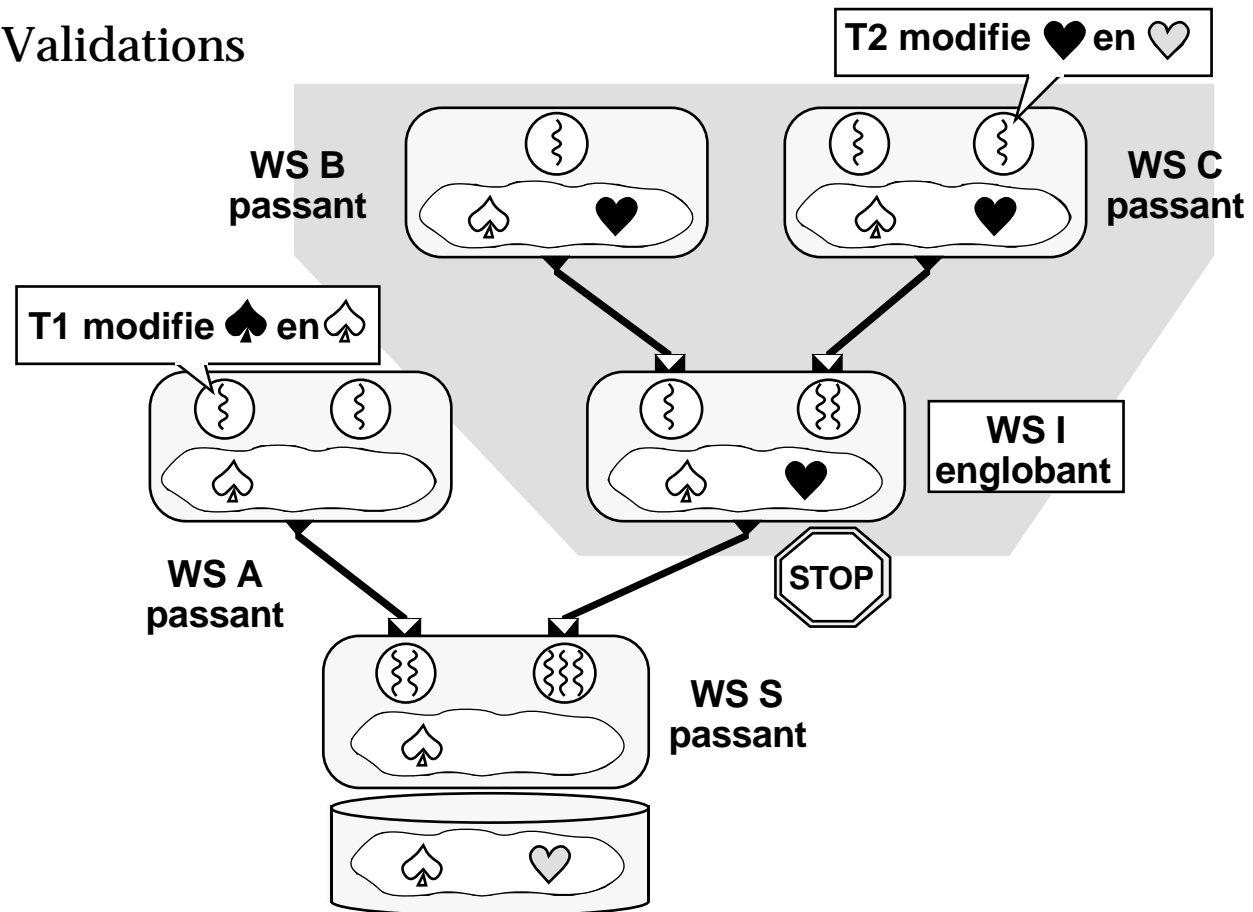


# Service de Données

## hiérarchisation de la Validation des Transactions

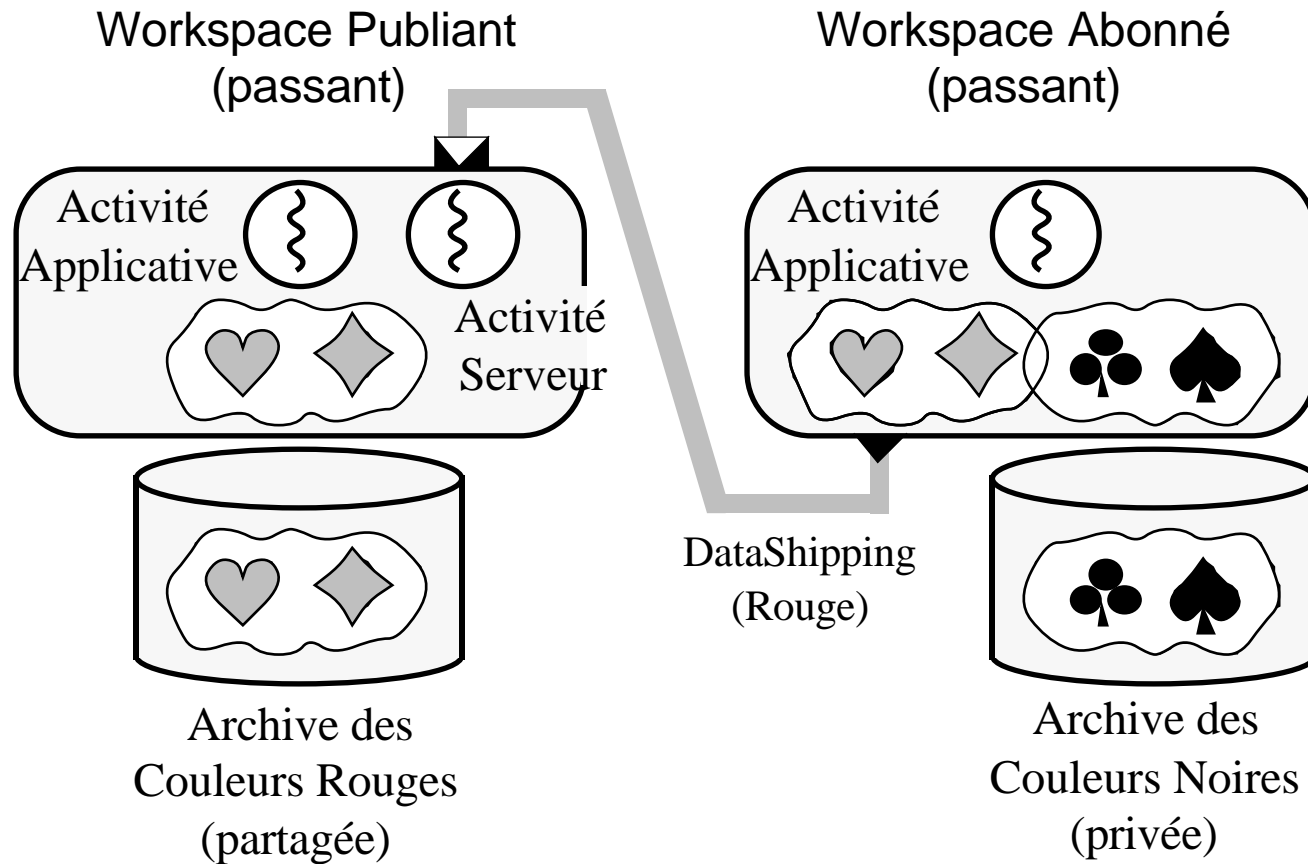
2 propriétés de Validations

- ✓ passante
- ✓ englobante



# Service de Données

## Bases Privées / Bases Partagées



# Bases Client/Serveur

## Structure Directe

connexion directe  
d'un client vers le serveur

Avantages :

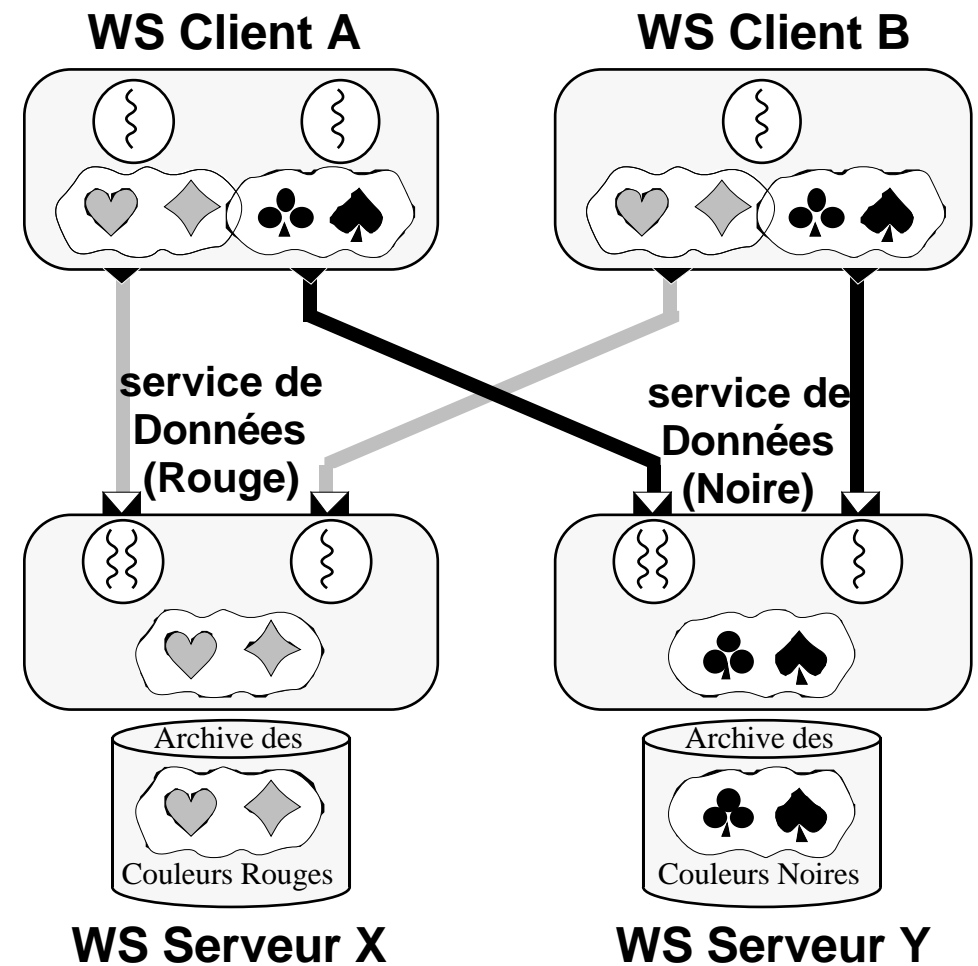
souplesse

pas d'intermédiaire

Inconvénients :

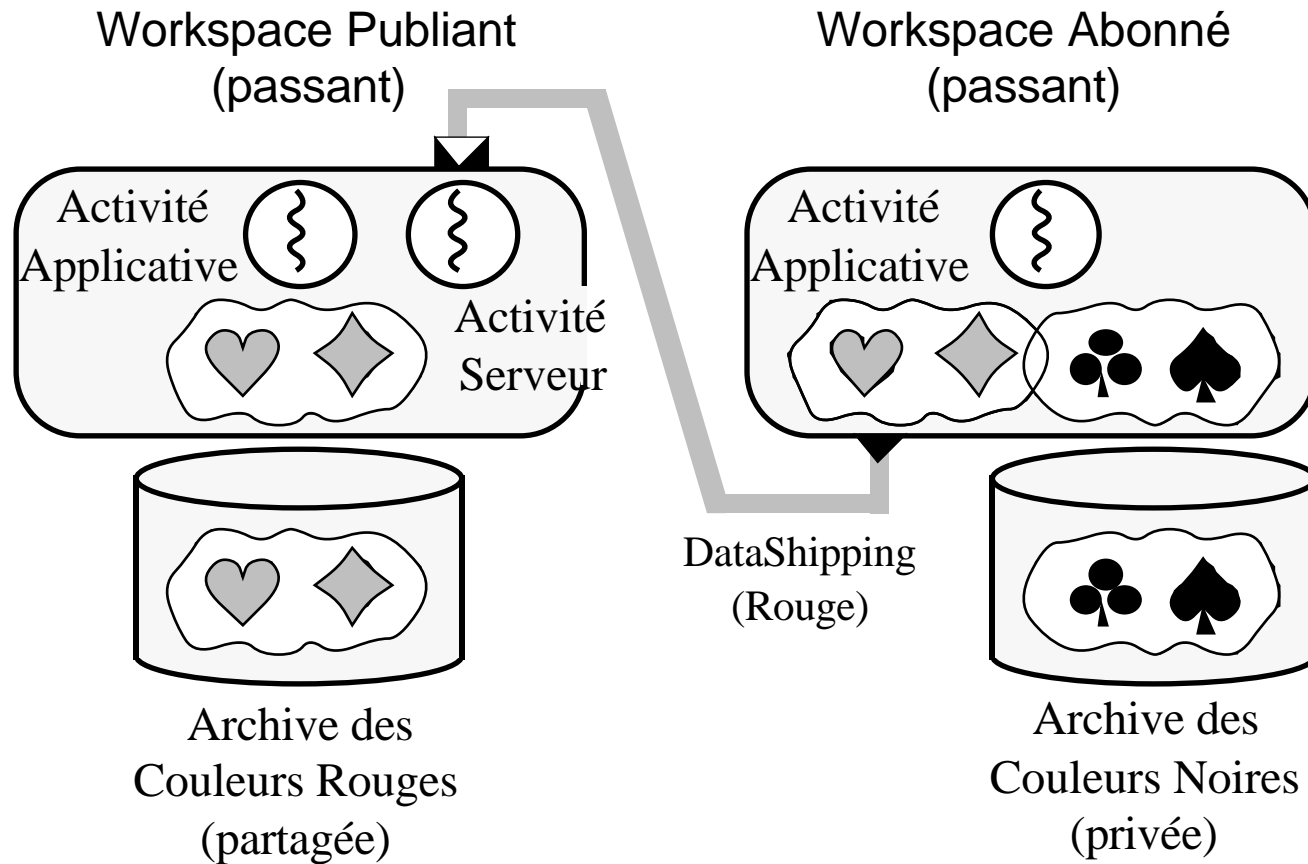
contrôle fiable des validations

cas de plusieurs applications  
sur un même client



# Service de Données

## Bases Privées / Bases Partagées



# Bases Client/Serveur

## Structure Symétrique

connexion directe d'un serveur vers les autres serveurs

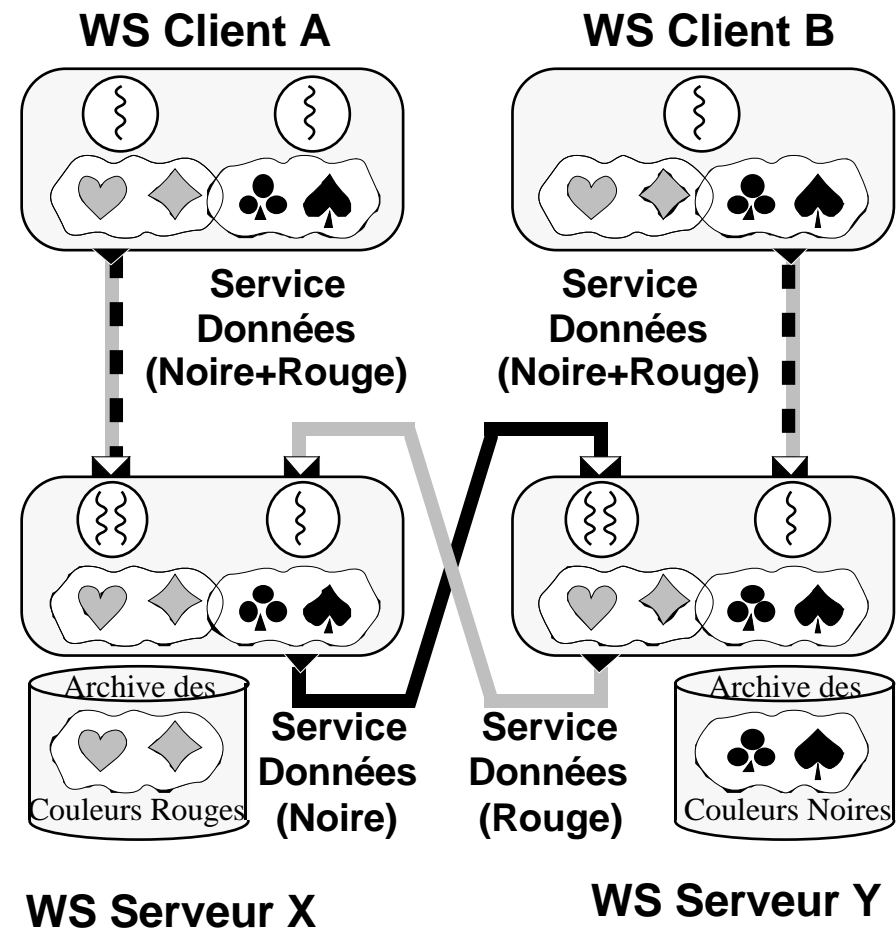
Avantages :

contrôle fort des validations

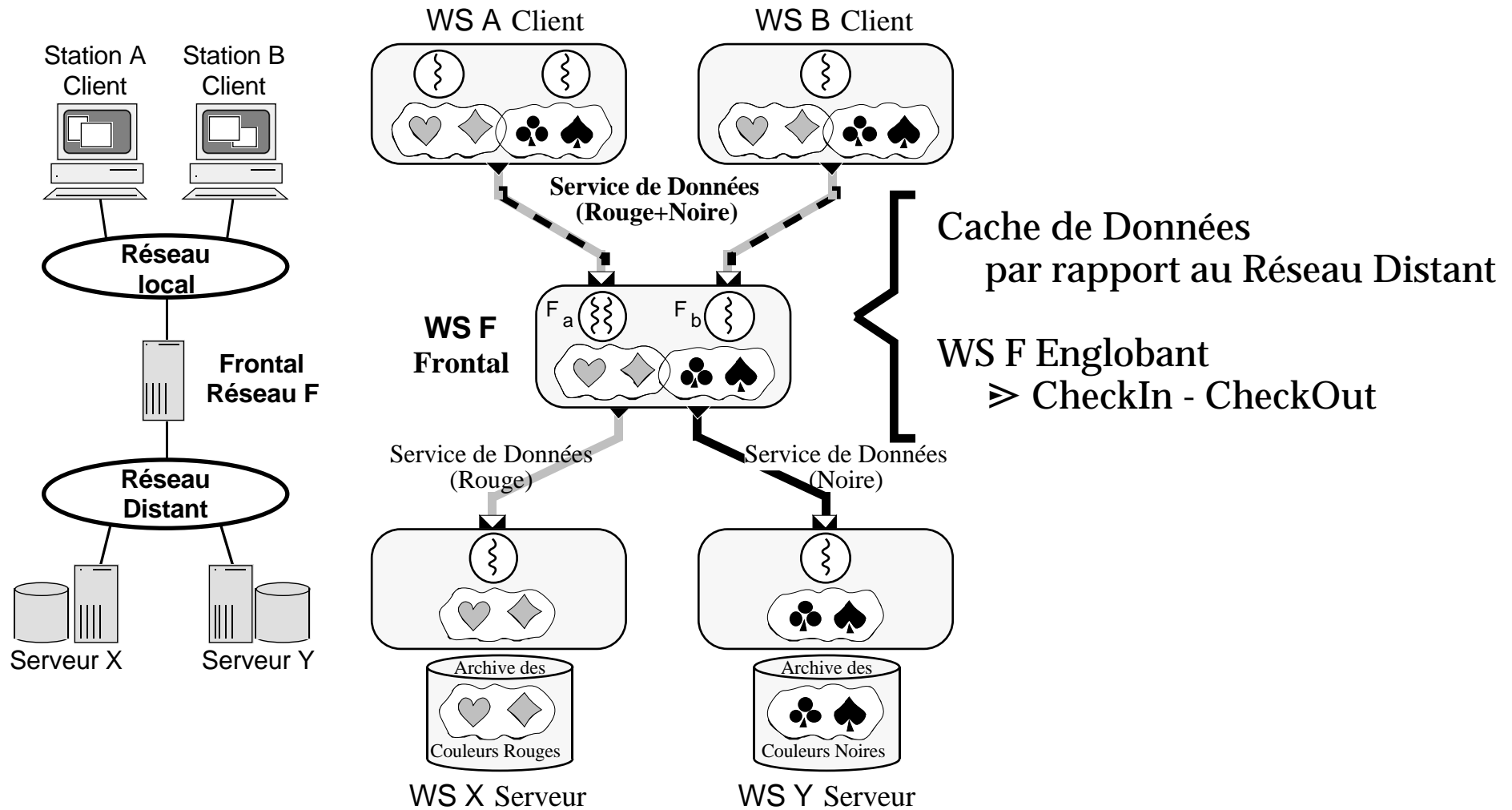
mise en commun

des données importées

(SHORE → serveur local )



# Bases Client/Serveur - Frontal Réseau Distant

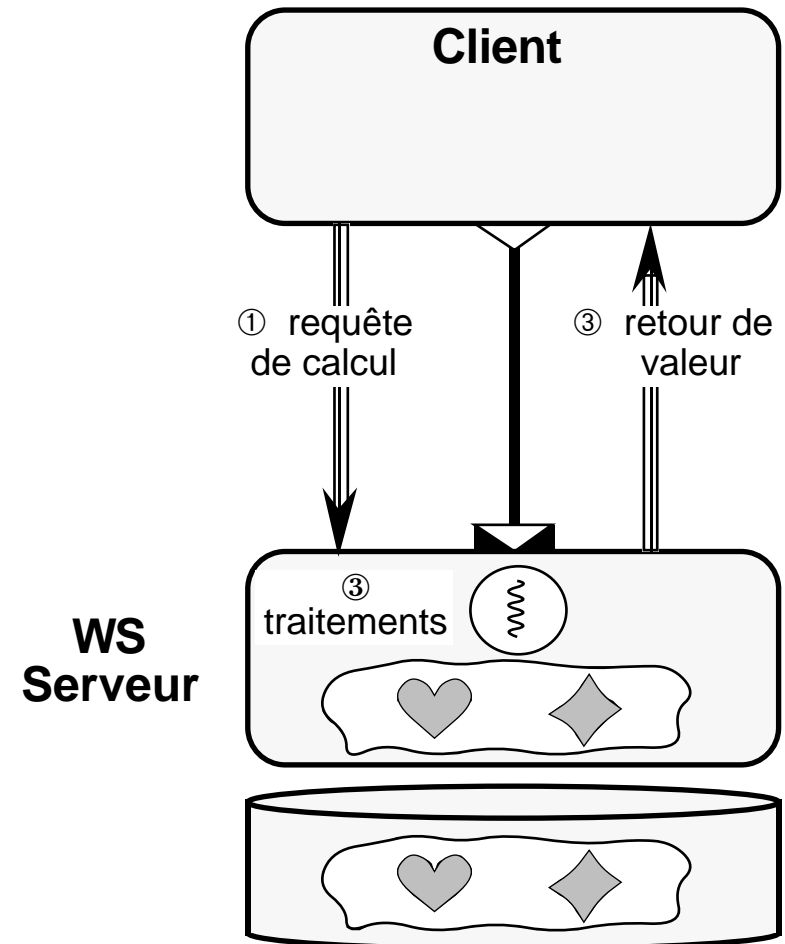




# Service d'Opérations

## Traitement des données par le serveur

- ✓ sécurisation des traitements
- ✓ confidentialité des données



---

# Service Mixte de Données et d'Opérations

bénéficie

## ✗ Data-Shipping

utilisation des ressources du client

## ✗ Query-Shipping

sécurité et confidentialité des traitements et des données

## ✗ Mise en cohérence entre le client et le serveur

---

# Utilisation du Workspace : le Travail Coopératif

## Définition

### ① Initialisation

- le superviseur démarre un groupe de travail
- les coopérants (autorises) s'attachent au groupe

### ② Travail

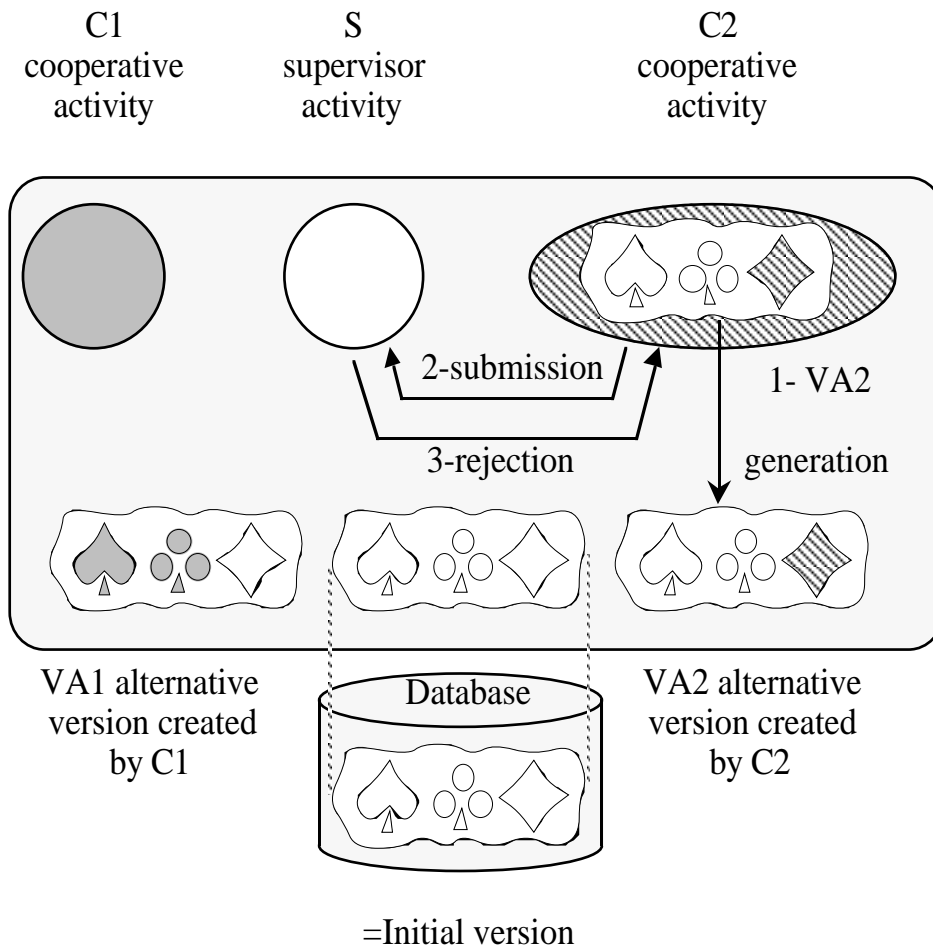
- les coopérants consultent, produisent et modifient des versions alternatives de la base
- le superviseur modère les changements

### ③ Conclusion

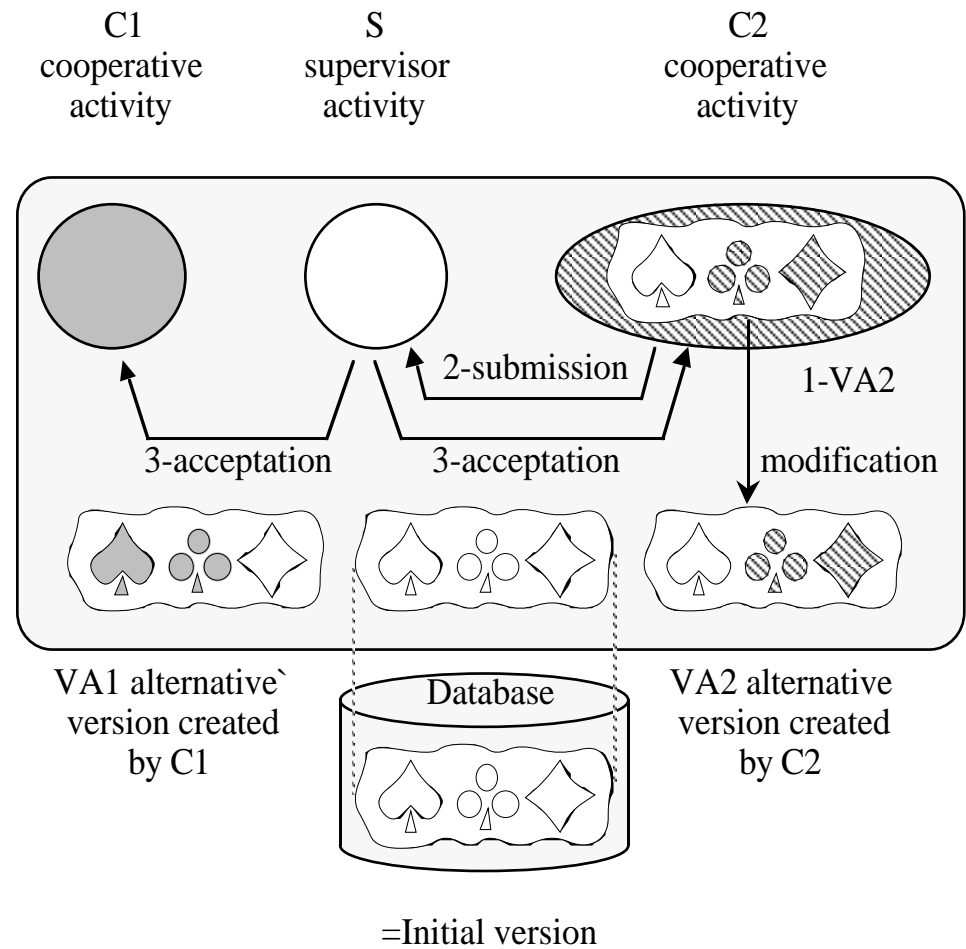
- les coopérants choisissent les propositions à valider
- le superviseur coordonne le consensus et applique celui-ci

# Travail Coopératif - Exemple I

## 1- Soumission et Rejet

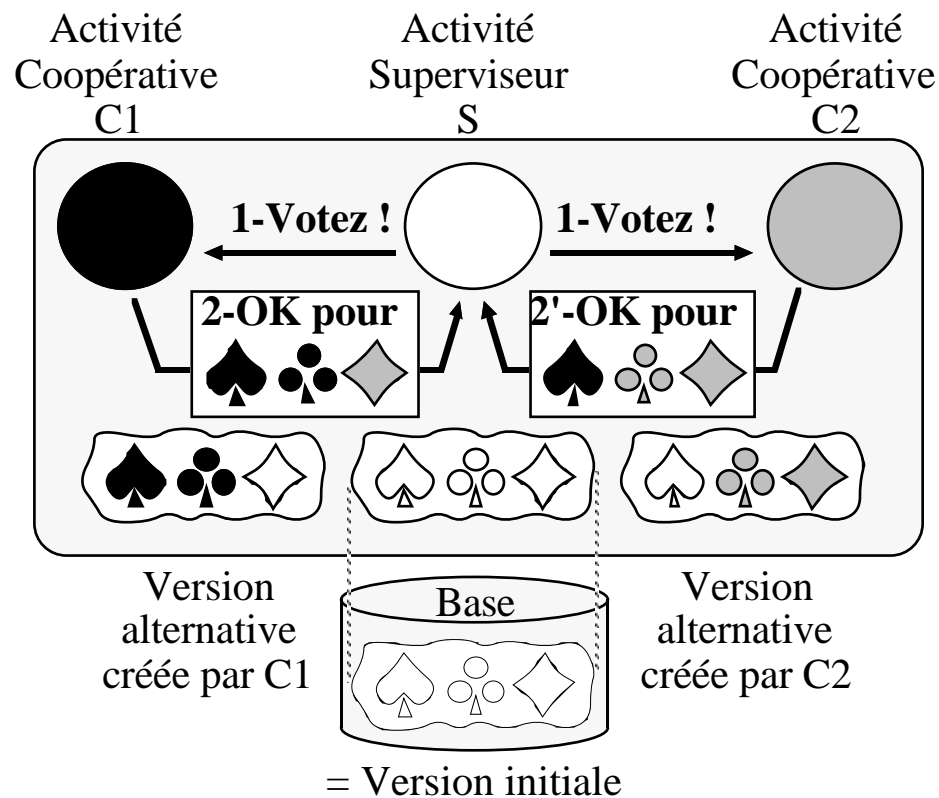


## 2- Soumission et Acceptation

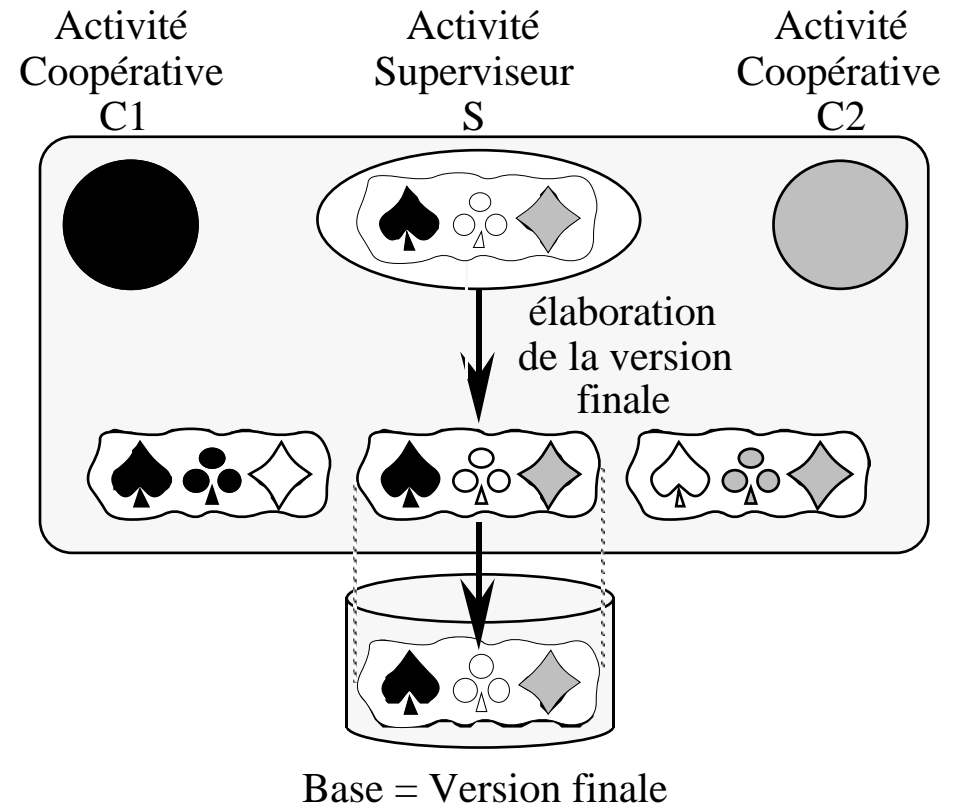


# Travail Coopératif - Exemple

## 1- Consensus d'élaboration



## 2 - Validation du travail de groupe



---

# Travail Coopératif - Besoins

✓ Versions Alternatives

✓ Notifications

Communication Coopérants <-> Superviseur

✓ Service Coopératif

Accès distant par des Coopérants Distants

---

# Implémentation

## ✓ Workspace

MultiThreading

*Transactions, Communications Asynchrones*

Memory-Mapping

*Accès aux données persistantes par matériel*

## ✓ Publication-Abonnement

Service générique

## ✓ Interface C++

Persistance et Verrouillage transparents

## ✎ Distribution par le service de Données en cours

---

# Perspectives

## ▣▣▣▶ MultiMédia

Flots Multiples ( synchronisation, temps réel )

Filtres

## ▣▣▣▶ Travail Coopératif

Comportements (Consensus, Modération, ...)

Persistence du travail (verrou persistant)

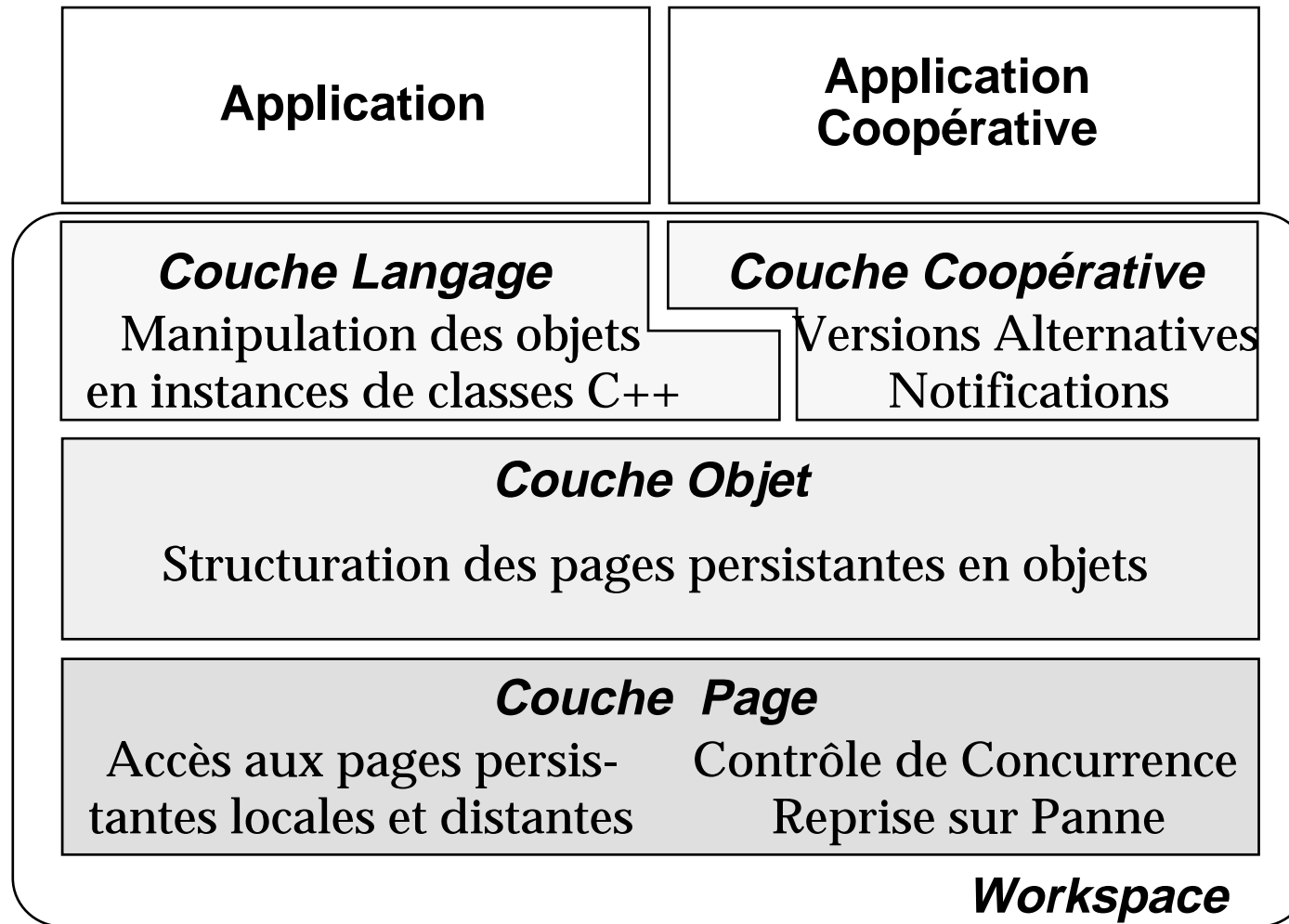
## ▣▣▣▶ Bases de Données Nomades

Migration du contexte d'une transaction

fiabilité des connections (checkin-checkout, CC optimiste)



# Implémentation du Workspace

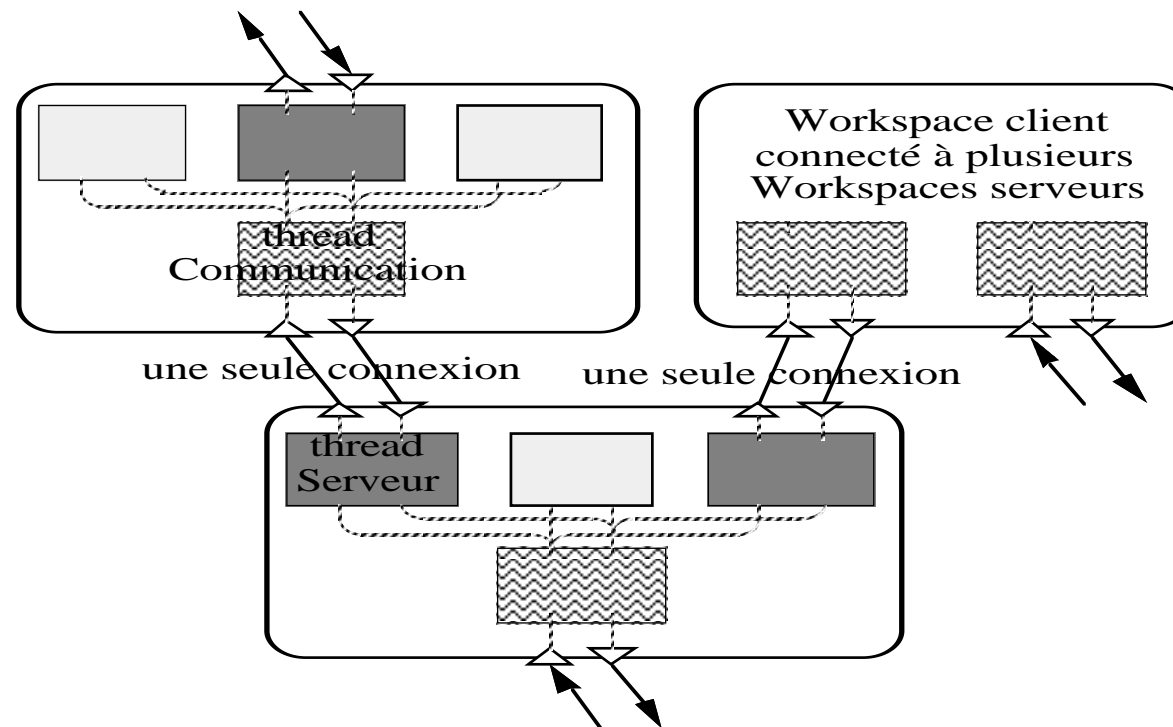


# Exécution des Applications et des Services

utiliser les fonctionnalités des OS

## Threads

- ↳ Transactions Applicatives
- ↳ Transactions Serveurs
- ↳ Threads systèmes (communication asynchrone, journalisation)

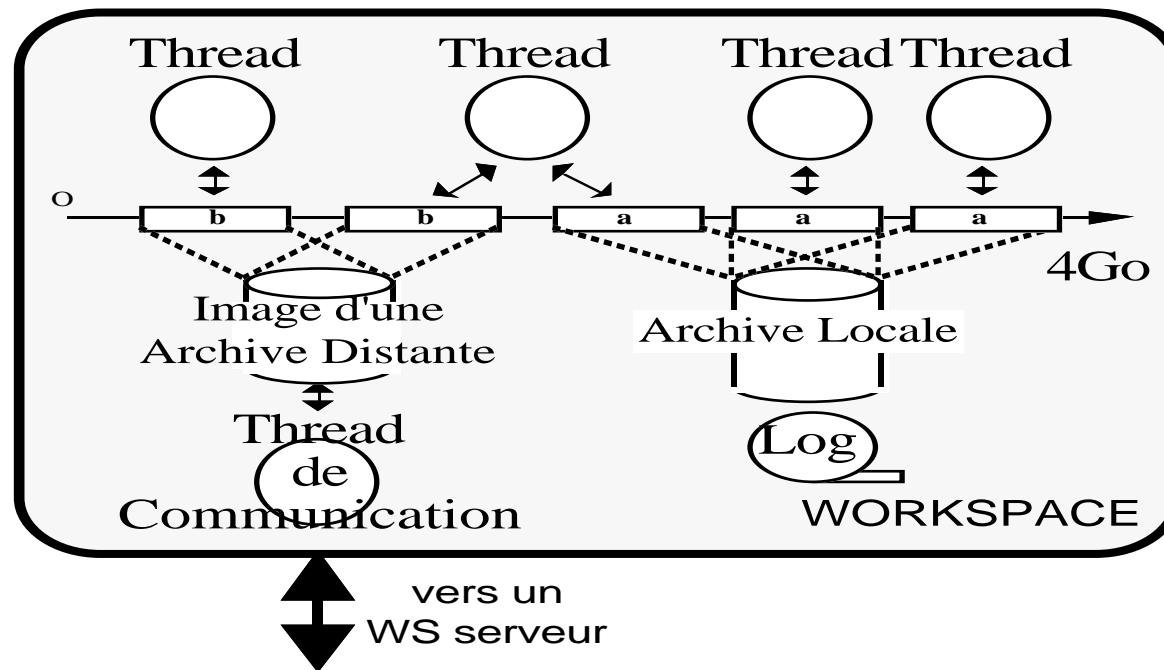


# Accès aux Pages Persistantes

utiliser les fonctionnalités des OS

## Memory-Mapping

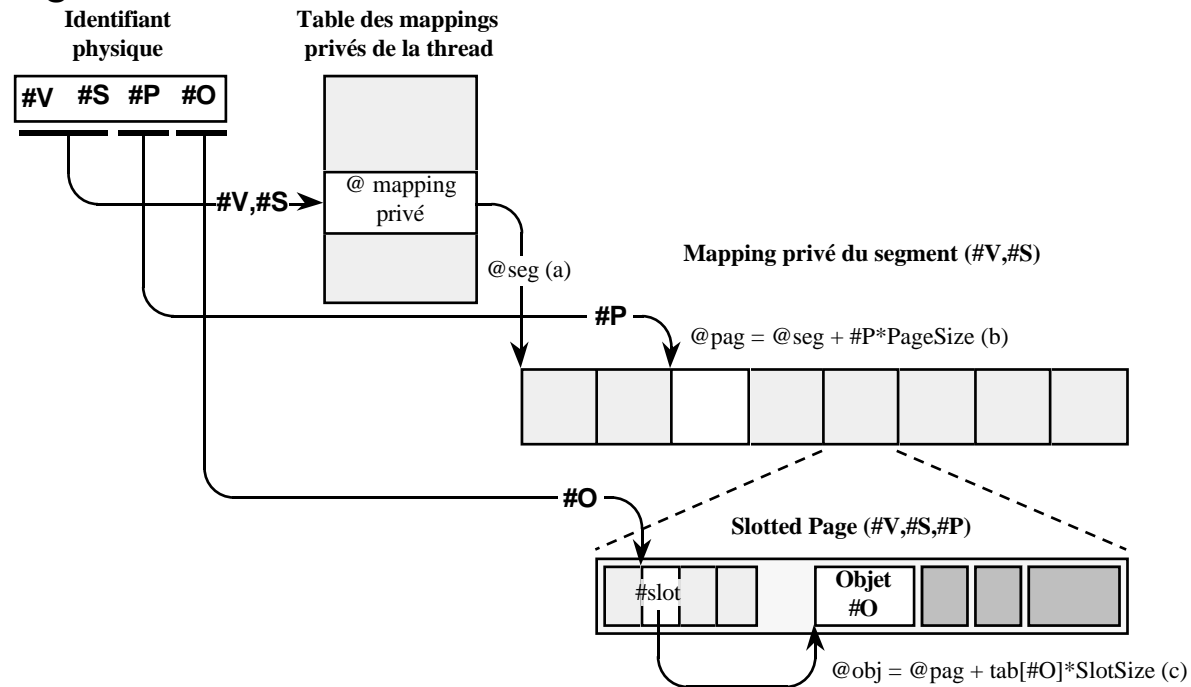
- ➔ Accès direct aux objets
- ➔ Chargement des pages implicite
- ➔ Demande de verrou implicite



# Accès aux Objets Persistants

## Objets Courts

### Slotted Pages



## Objets Longs (BLOBs)

Contiguïté en Mémoire Virtuelle  
(bibliothèque externe, objets)

---

# Interface Langage C++

## ✓ Programmation OO

avec persistance transparente au développeur

Déclaration du schéma de la base

Définition de l'application et des méthodes

## Interface C++=

PréCompilateur élémentaire

+ Compilateur du Marché (GNU ou ATT)

+ Outils associées (Debugger, Profiler ...)

Ajout des méthodes d'accès, ...

Swizzling des Références Persistantes

Acces rapide aux objets