Module BDR
Master d'Informatique (SAR)

Cours 4- SGBD Objet : implémentation

Stéphane Gançarski
Stephane.Gancarski@lip6.fr

SGBD objet : implémentation

Architecture

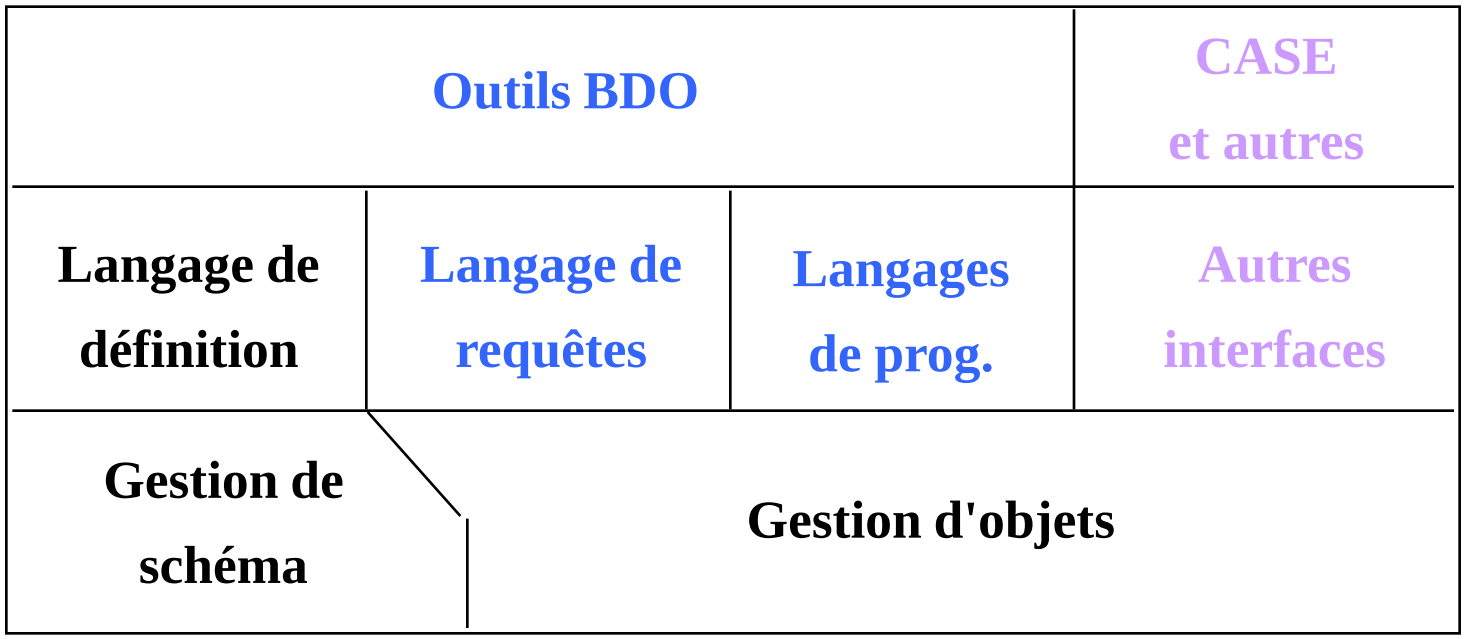
Fonctionnalités

Quelques systèmes existants

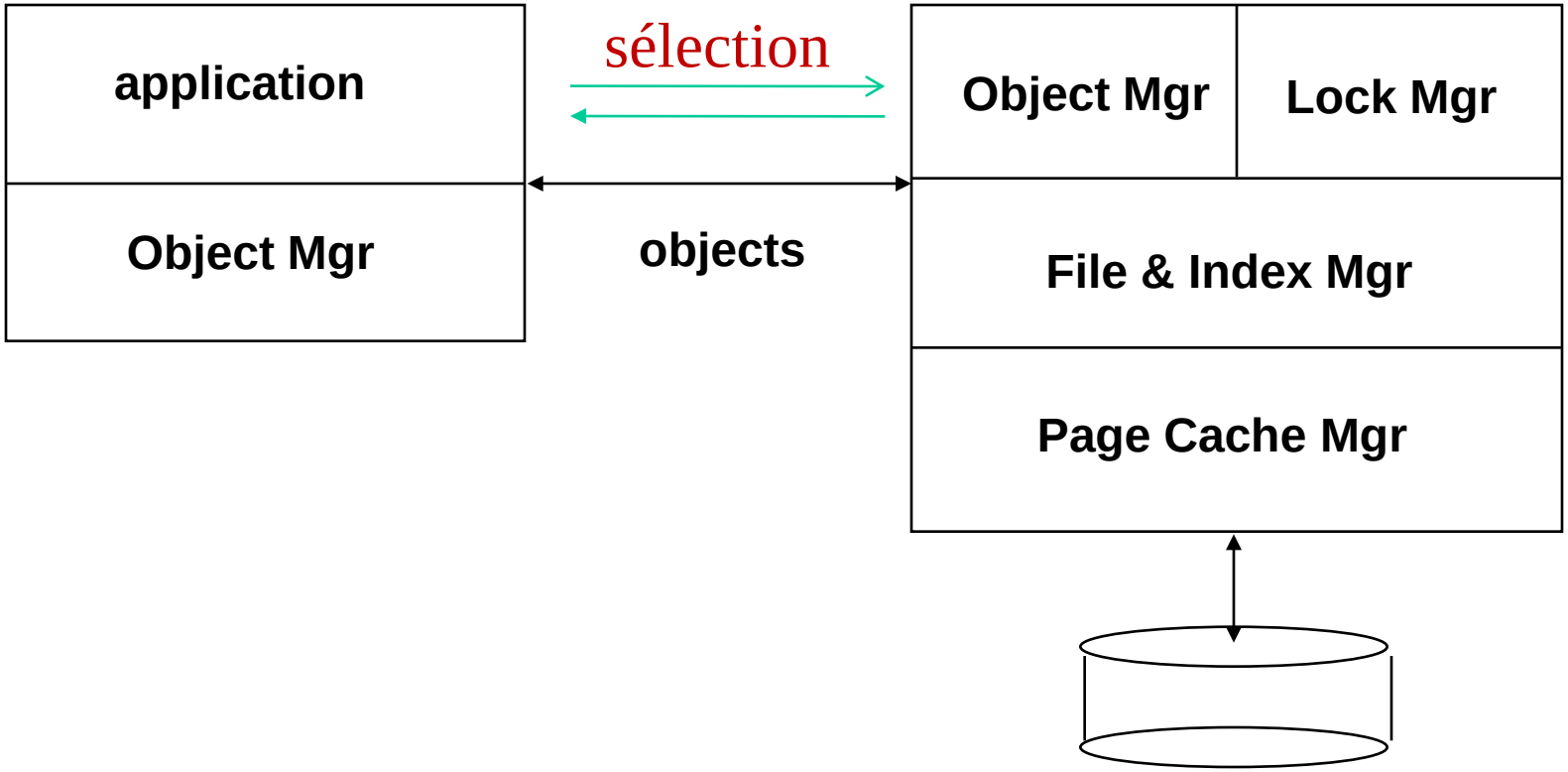
Challenges

- Stockage d’objets structurés
- Stockage de gros objets
- Performances (cache de méthodes, pointer swizzling)
- Index
- Optimisation de requêtes
- …

Architecture fonctionnelle



Serveurs d'objets

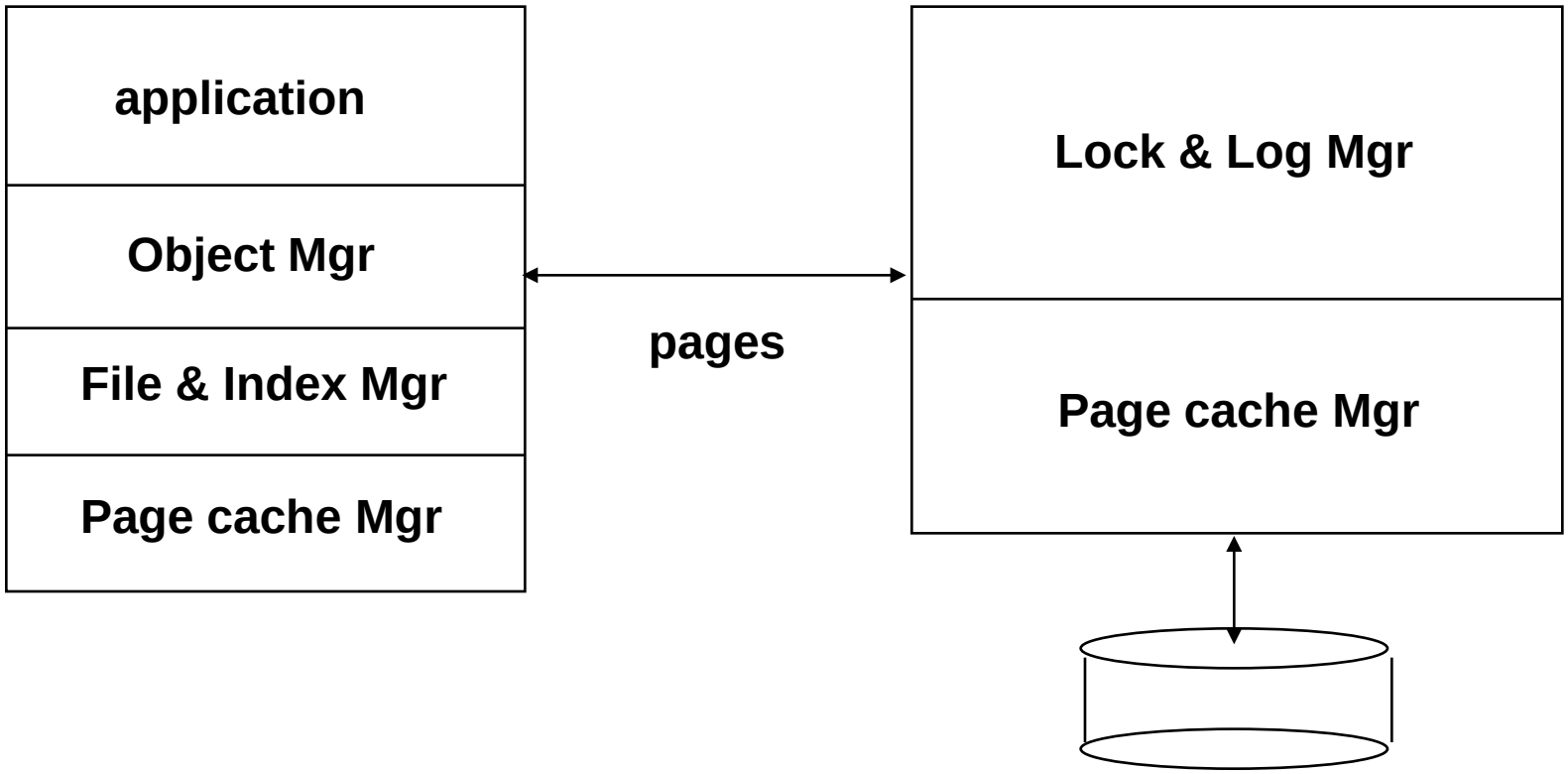


Avantages et inconvénients

+ clients et serveurs jouent un rôle symétrique
    les sélections sur le serveur évitent le transfert d'objets
    décharge la station
+ Verrouillage objet possible sur le serveur

- Surcharge du serveur
- gestion d'objets dupliquée
- au pire, 1 RPC par référence à chaque objet
- complique la communication client/serveur

Serveurs de pages



Avantages et inconvénients

+ minimise la charge du serveur
+ simplicité du protocole de communication
+ coût de transfert d'une page peu supérieur à celui d'un objet

- pas de sélection possible au niveau du serveur
- impossibilité de répartir la charge entre client et serveur
- contrôle de concurrence au niveau page

CORBA

(Common Object Request Broker Architecture)

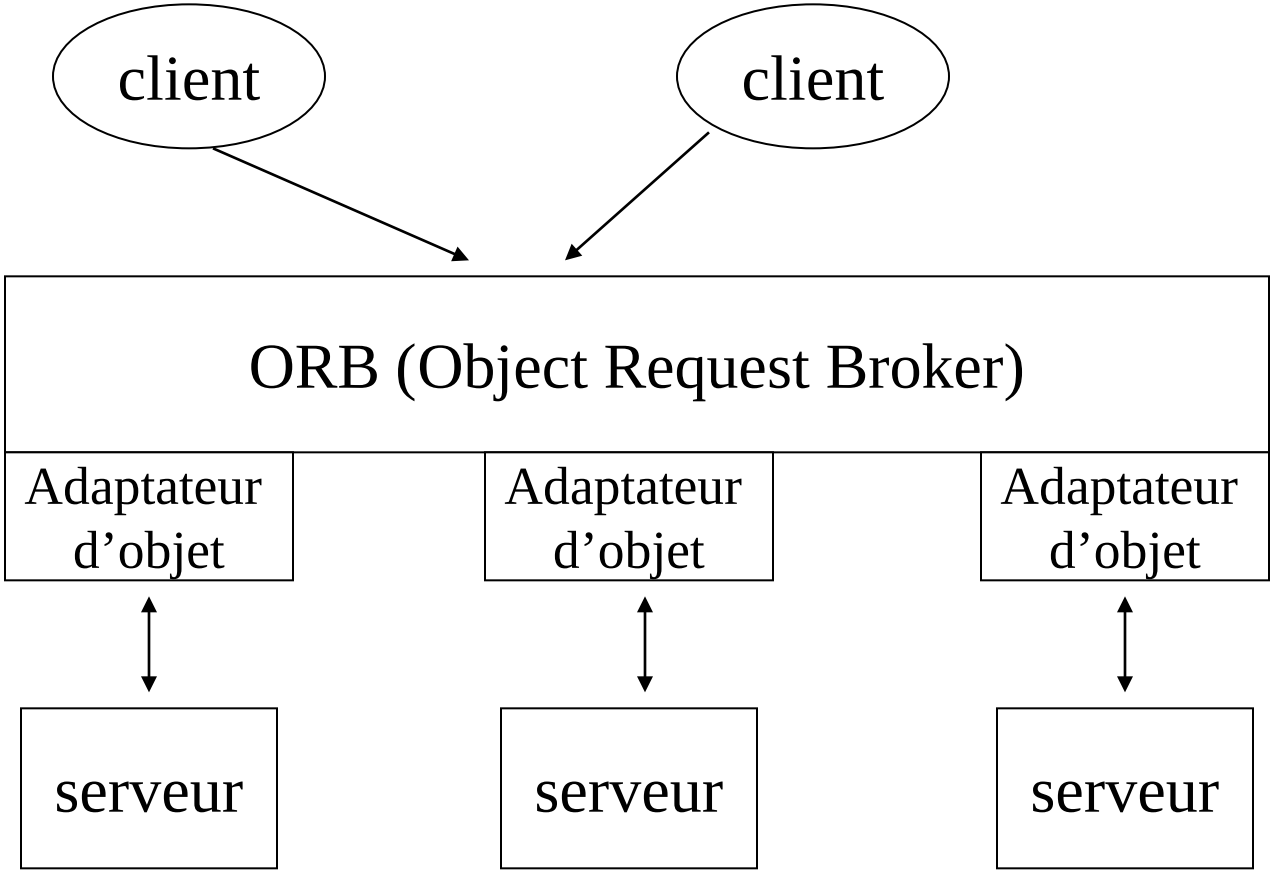
Défini par l'OMG.

Architecture d'échange de requêtes et de réponses entre objets.

L'OMA (Object Management Architecture) comprend la définition d'un modèle d'objets commun, d'un modèle commun d'interaction (CORBA), et d'un ensemble de services objets (COSS).

La communication entre objets s'effectue par requêtes.

SGBDO et ORB


client
client
ORB (Object Request Broker)
Adaptateur d'objet
Adaptateur d'objet
Adaptateur d'objet
serveur
serveur
serveur

CORBA

- ORB : permet d’échanger des requêtes et des réponses entre objets de façon transparente (localisation et implémentation)
- OA (Object Adapter) : permet l’accès à l’implémentation des objets (génération et interprétation de références d’objets, appels de méthodes, contrôles, etc.)
- La communication entre ORB s’effectue par le protocole IIOP (Internet Inter-ORB Protocol).

Fonctionnalités

- Gestion d'objets
  - Représentation des objets
  - Identificateurs d'objet
  - Mécanismes d'adressage
  - Stockage des objets
  - Index et optimisation des requêtes
  - ramasse-miette
- Support du travail coopératif
  - transactions longues
  - versions
- Modélisation et évolution de schéma

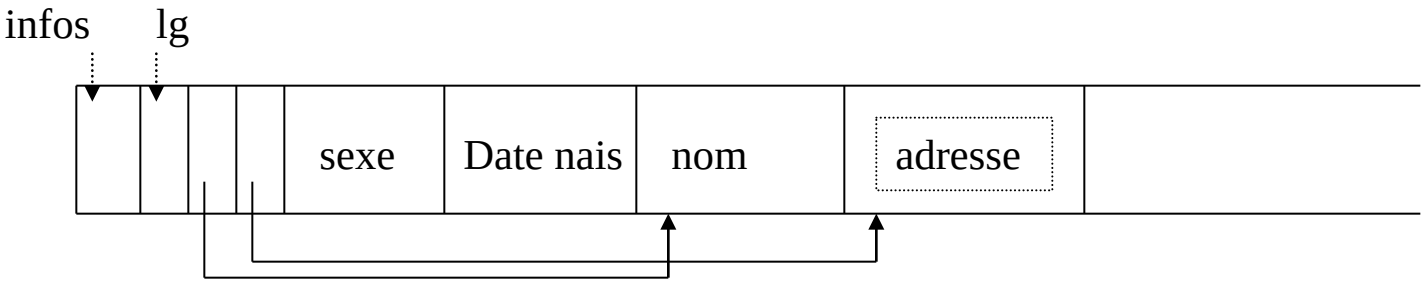
Représentation des objets

- Retrouver efficacement tous les composants d'un objet, un sous-ensemble des composants
- Implanter efficacement les opérations du langage
- Gestion des gros objets
- Rappel: le schéma est stocké par le SGBD, donc disponible

En relationnel, tout est simple car les n-uplets sont de taille fixe. En connaissant le schéma d'une relation, on peut retrouver facilement toute information (valeur d'attribut)…

Représentation des objets

- N-uplet : enregistrement avec des champs (attributs) de longueur fixe ou variable



- Listes : arbres triés
- Ensembles : tableaux de pointeurs, enregistrement avec entête contenant le type des éléments, la longueur totale, la cardinalité, puis liste de déplacements.
- Pb: objets longs, dépassant la longueur d'une page

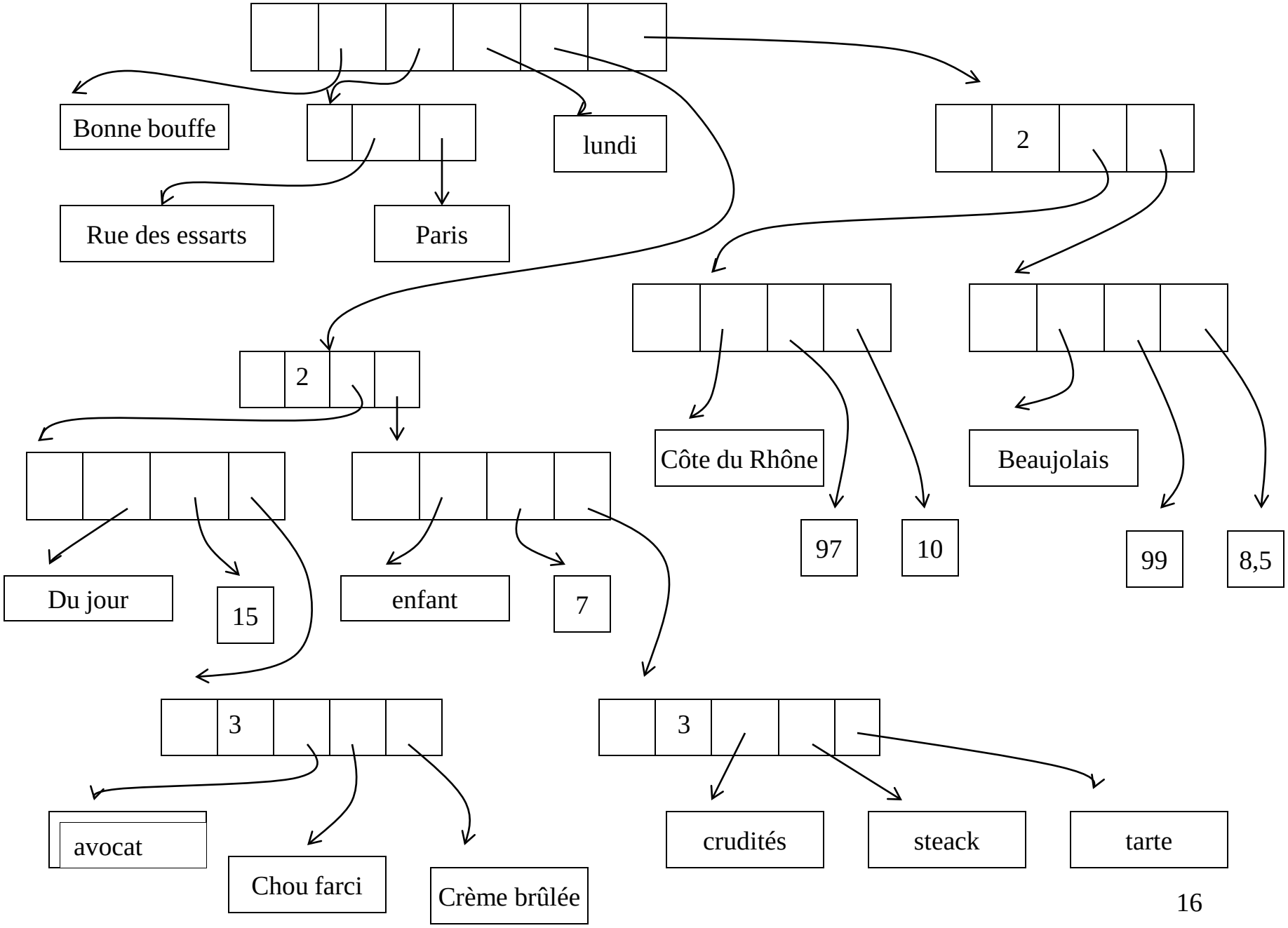
Objets complexes

- N-uplet : tableau de pointeurs (un par champ)
- Ensemble : tableau de pointeurs, avec une entrée par élément
- Toute l'information est dans les feuilles

  \+ les informations peuvent être localisées de façon indépendante

  \+ évolution facile, bonne adaptabilité

  \- trop grande fragmentation, la structure est répartie sur des enregistrements physiques distincts


Bonne bouffe
lundi
2
Rue des essarts
Paris
2
Côte du Rhône
Beaujolais
97
10
99
8,5
Du jour
15
enfant
7
3
3
avocat
Chou farci
Crème brûlée
crudités
steack
tarte

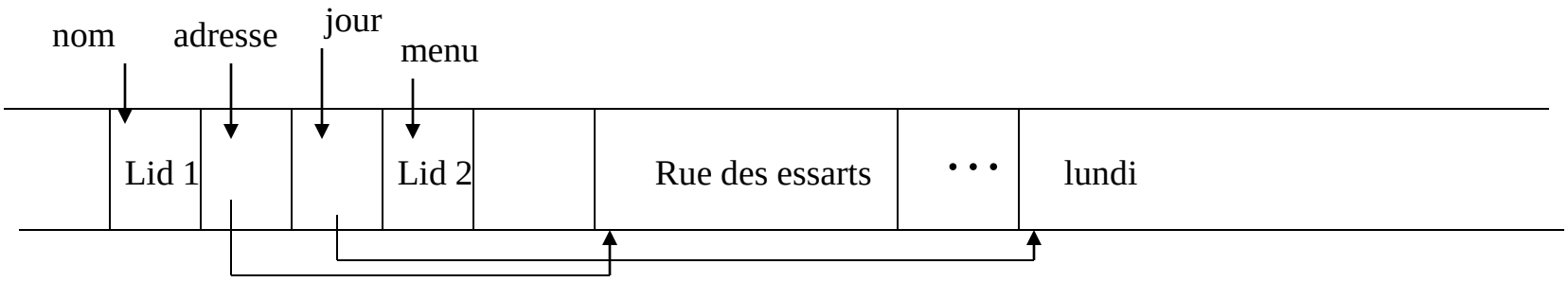
Objets longs

Objets dont la taille est supérieure à une page.

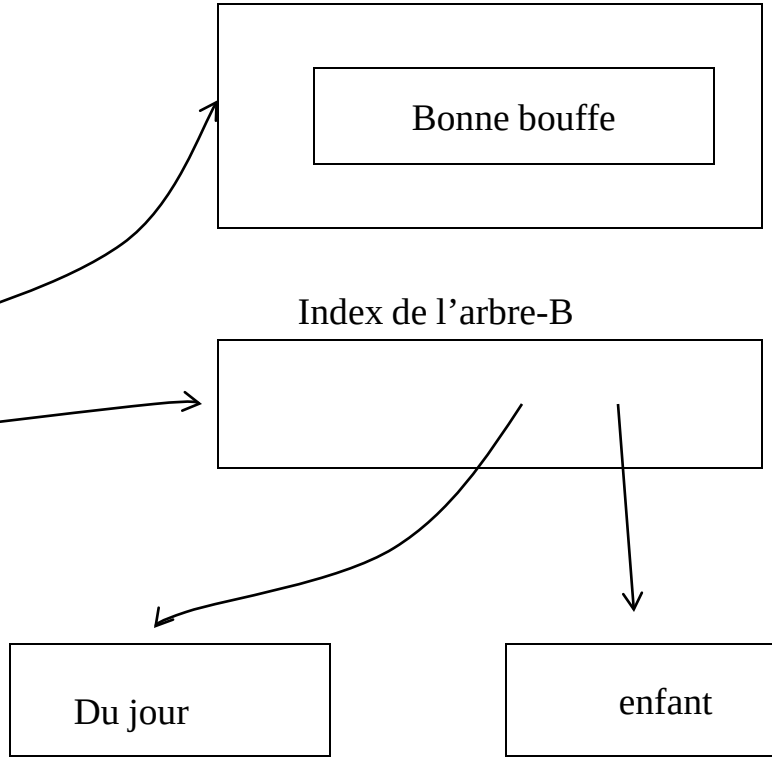
Un identificateur d'objet long permet d'accéder à un catalogue d'objets longs, donnant le type de l'objet et des indications complémentaires sur l'accès aux composants.

N-uplets : enregistrements fragmentés et reliés par pointeurs, séquence d'éléments de longueur variable

Ensembles : arbre-B



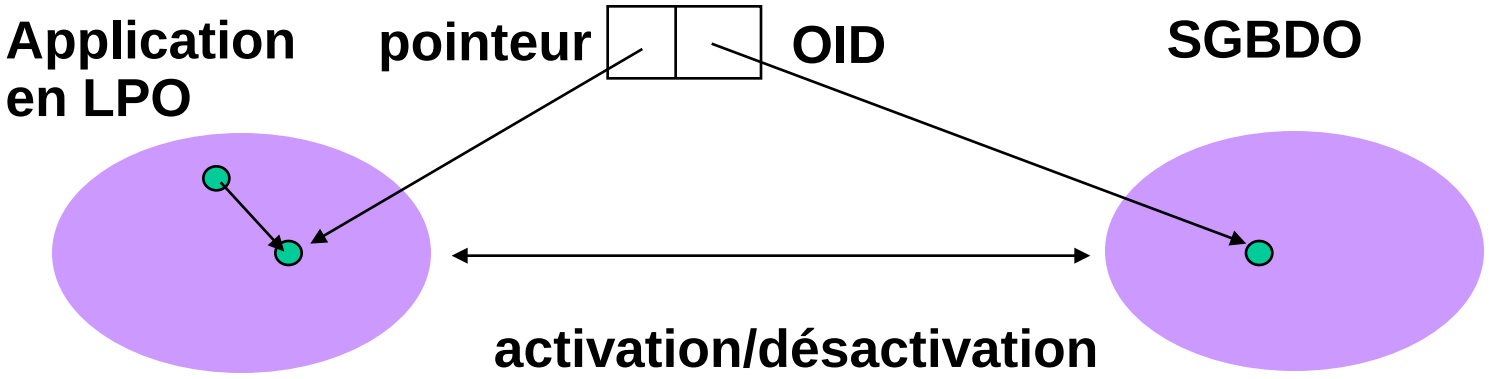
Lid	type	informations	
1	file	1	
2	Arbre-B	chaîne	



Identificateurs d’objet

- Problèmes :
  - Maintenir l’unicité
  - Éviter les références folles
  - Efficacité d’accès en MC et en MS
  - Adaptabilité des mises à jour
- Plusieurs techniques :
  - Oid physique : reflète la position de l’objet sur le disque, accès rapide, mais difficulté pour les màj.
  - Oid logique : plus facile pour changer de place un objet sur le disque, mais nécessite une table (souvent très grande) de correspondance.

Accès aux objets



Pointer swizzling: lors de l'activation, conversion des références inter-objets en pointeurs

+ accès performant aux objets persistants en mémoire

- surcoût amorti si nombreux accès aux objets déjà en MC

Stockage des objets

Placement des objets sur le disque

tenir compte des liens d'héritage et de composition

Plusieurs stratégies :

par classe

multiclasses

par proximité d'un autre objet (parent)

On peut avoir des index de chemins

emp.affectation.chef.nom	emp-oid
"Martin"	e1, e2, e6
"Smith"	e3, e4, e7, e8

Indexation

- Problèmes :
  - Héritage et sous-typage : l'index doit renvoyer aussi les objets des sous-classes
  - Objets complexes (avec éventuellement des cycles) : indexer la chaîne de composition
  - Ensembles
  - Indexation sur méthode (retrouver des terrains selon leur surface, calculée par méthode)

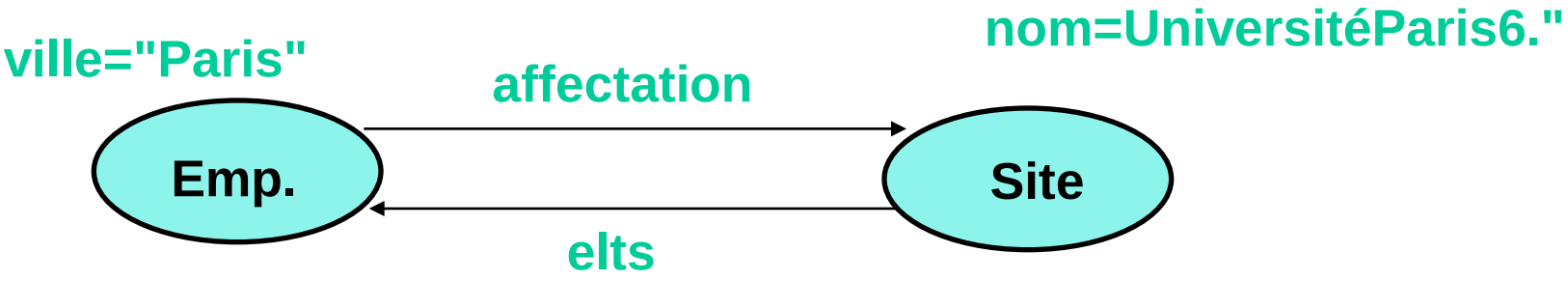
Optimisation de requêtes

Problèmes :

- Support des méthodes
  - évaluation du coût
  - évaluation des résultats
- Jointures par références
  - possibilité de jointures n-aires
  - traversée des chemins en profondeur, en largeur, …
- Structures complexes
  - parcours de collection
  - application de prédicats
- Utilisation d'index et placement
  - prise en compte du placement des objets
  - utilisation d'index de chemin

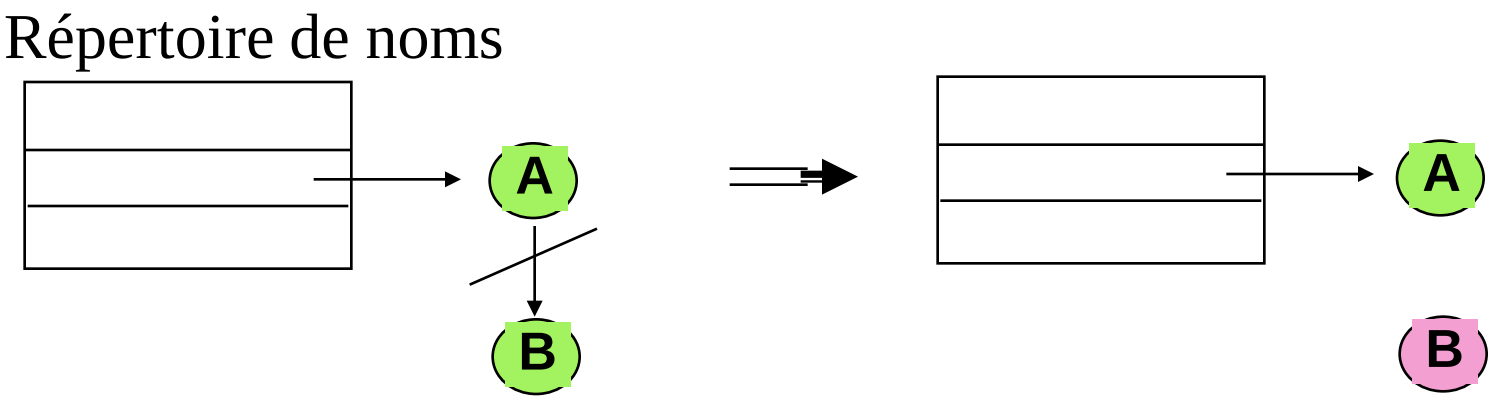
Optimisation de requête

```
select e
from e in Emp
where e.adresse.ville = "Paris" and
        e.affectation.nom = " Université Paris 6."
```



considérer toutes les traversées possibles du graphe et choisir la meilleure en fonction des index et de son coût estimé

Ramasse-miettes



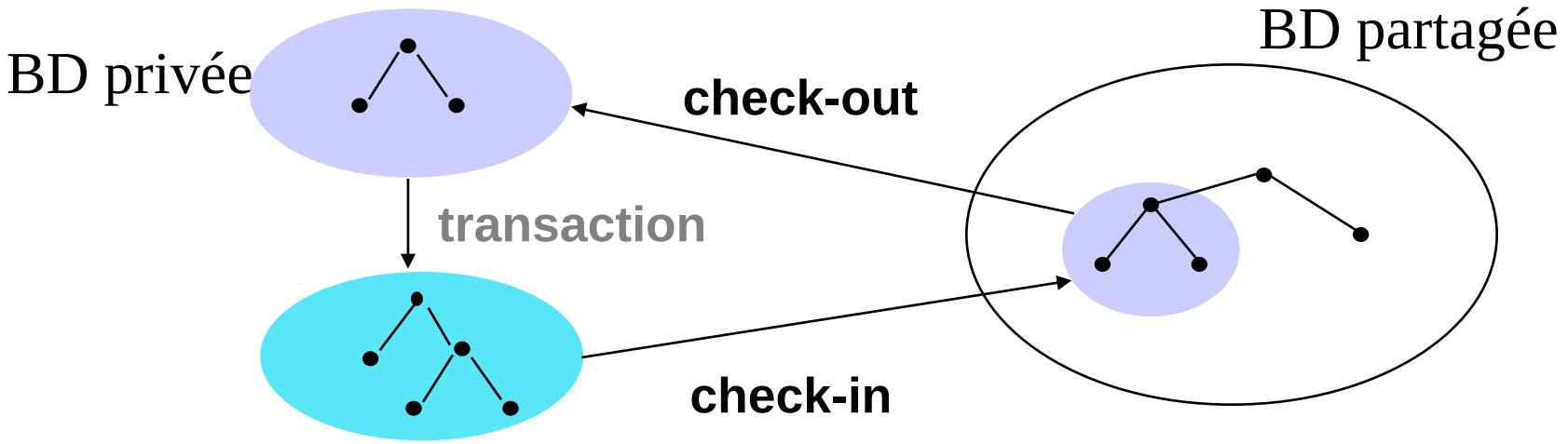
Comment désallouer B ?

programme utilisateur : source d'erreurs

compteur de références : efficace mais lourd à gérer

parcours dans la base : complet mais lourd, et bloquant

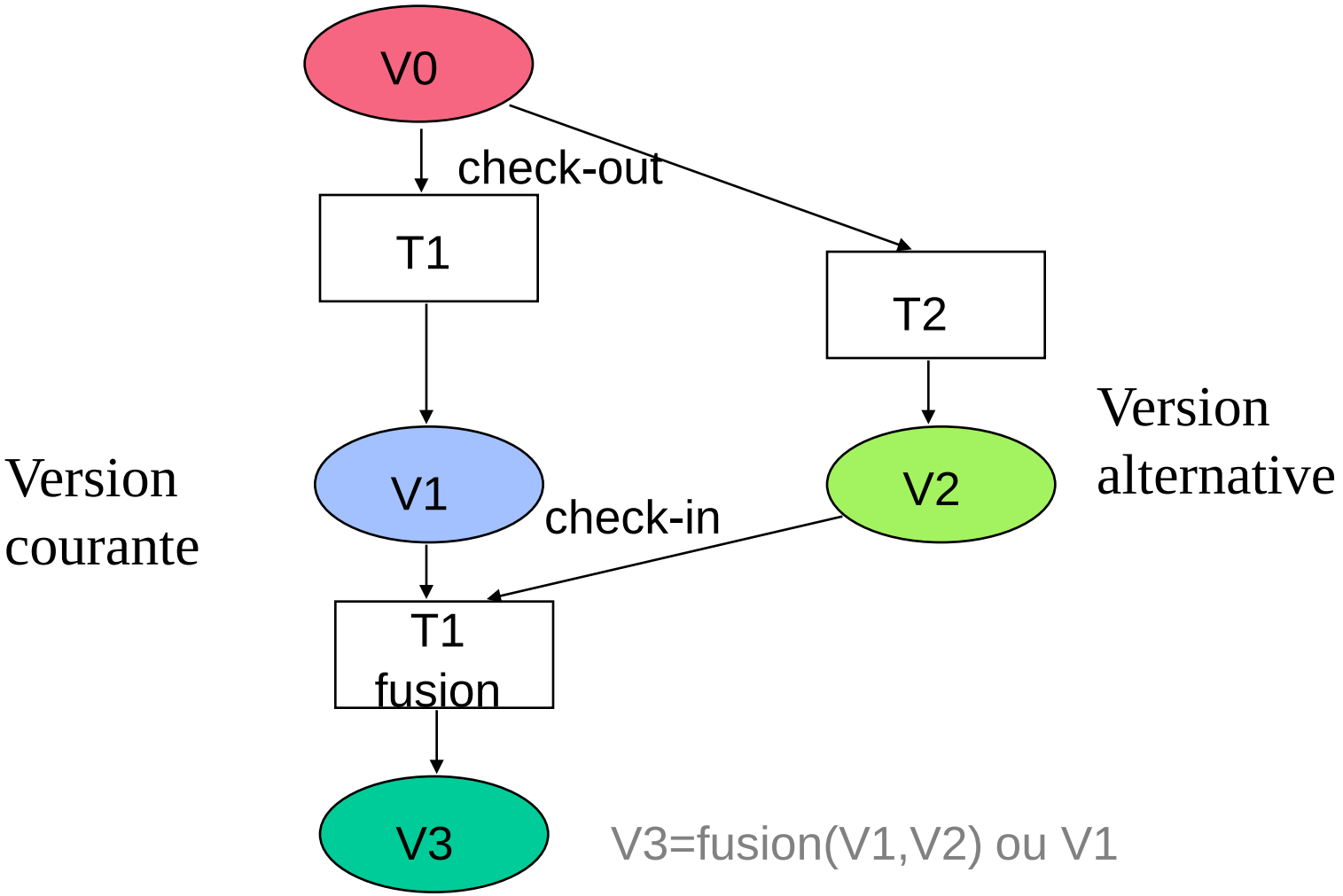
Travail coopératif



Check out : extraire un objet de la base publique et l'intégrer à une base privée en vue d'une mise à jour de longue durée

Check-in : réintégration, après mise à jour, de la nouvelle version d'objet dans la base publique. Problème si d'autres versions des mêmes objets ont été extrait par d'autres utilisateurs
+ performances (diminution du coût réseaux, pas de concurrence)

Versions


V0
check-out
T1
T2
Version courante
V1
check-in
V2
Version alternative
T1 fusion
V3
V3=fusion(V1,V2) ou V1

Contrôle de concurrence optimiste

Contrôle repoussé en fin de transaction

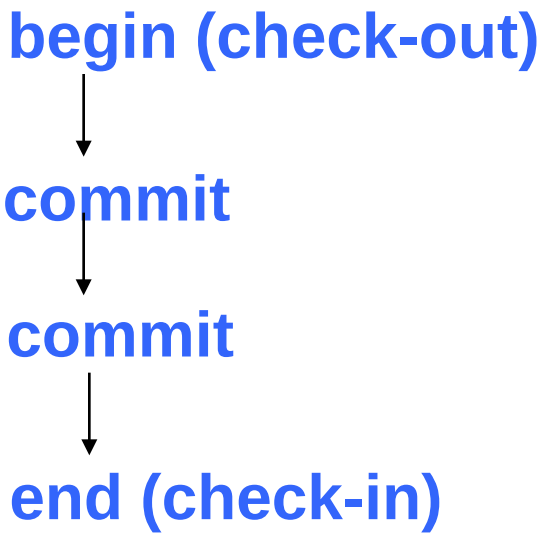
- les objets sont accédés en lecture/écriture sans contrôle
- l'ensemble des OID est gardé
- validation : chaque transaction qui accède à un objet dans un mode incompatible est abandonnée (ou du moins la dernière)

+ pas d'attente des transactions et procédure de validation simple (intersection des ensembles d'OID)

- beaucoup d'abandons si concurrence forte

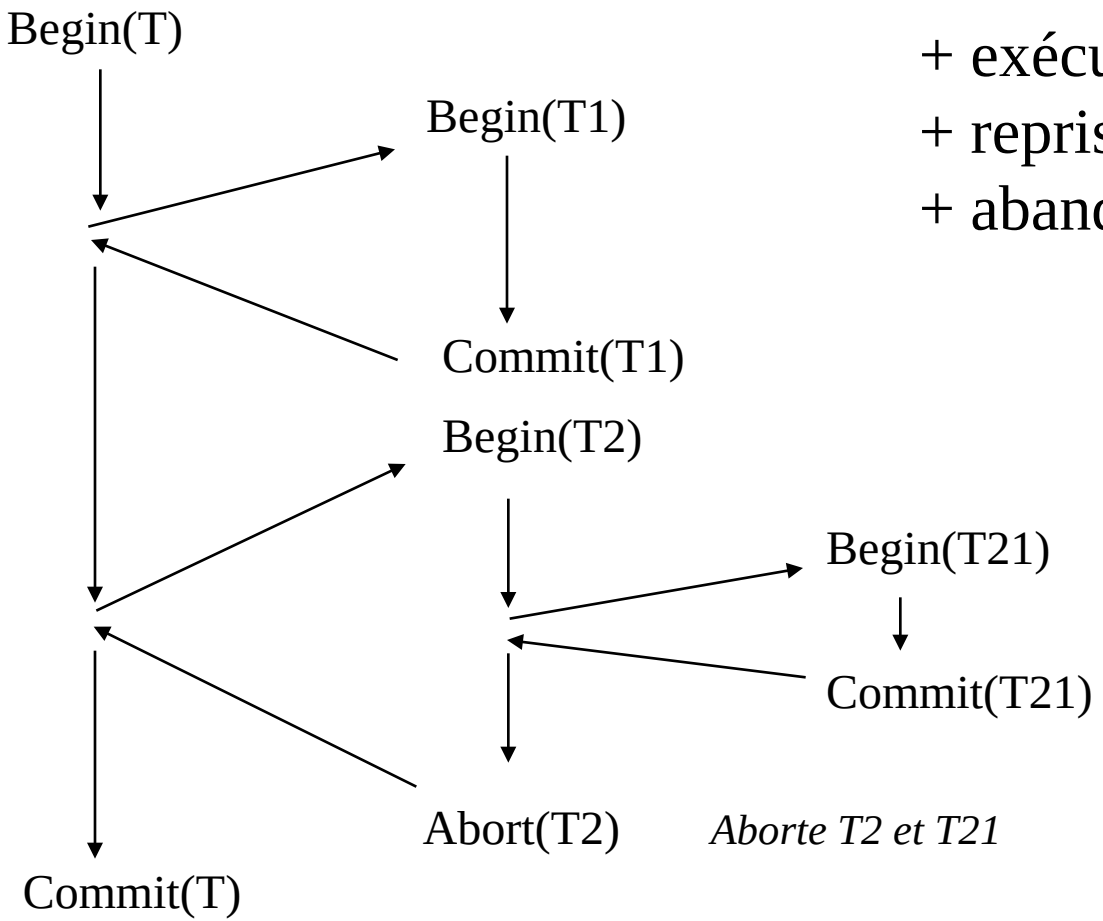
Transactions longues

Introduction de points de sauvegarde



La transaction peut être multi-session, les verrous résistent aux pannes.

Transactions imbriquées



+ exécution parallèle
+ reprise partielle
+ abandon partiel

Interfaces graphiques

Avantage des outils graphiques OO interactifs

- mécanismes par défaut pour affichage (classe Objet)
- personnalisation de ces mécanismes (héritage)
- interface ergonomique, multimedia
- maquettage et prototypage rapide

Outils BDO

- conception de schéma : designer, browser
- conception d’interface utilisateur (présentation)
- conception de programme

Evolution de schéma

- Modification des classes du schéma en présence d'objets
  - modification de type
  - modification d'attribut
  - modification d'opération
  - modification de la relation d'héritage
- Impact
  - recompilation des applications
  - rechargement de la base
- Solutions (partielles)
  - liaison dynamique des opérations
  - modification paresseuse des objets

Oracle

- NCA = architecture à trois niveaux (3-tier)
  - basée sur les standards: Corba, DCOM
  - support des composants Web: ActiveX, Java Beans
  - composants branchables : cartridges
    - cartridge = types, opérations, interfaces
    - implémentés en PL/SQL, C/C++, Java (avec mapping IDL)
    - sûrs: exécutés à l'extérieur du serveur
    - Sedona: outil de développement
  - support de SQL3 depuis la V8

Support objet d’Oracle

- Type objet SQL3
  - OID, objets complexes
  - appel d’opérations externes en C, C++ ou Java
- Cache objet sur le client pour la navigation rapide
- Support des données non structurées
  - image, vidéo, texte, etc.
- Extensibilité avec les data cartridges (composants)
  - PL/SQL, C/C++, Java (avec JVM intégrée)
  - sûr: dans un processus séparé (excepté Java)

Informix

- Informix Universal Server intègre le SGBDO Illustra
  - support SQL3
    - objets volumineux, type abstrait, opération, héritage multiple, collections, fonctions built-in, cast, etc.
  - extensibilité grâce aux Datablades
    - Datablade = types + fonctions + tables + code utilisateur
    - DataBlades 2D, 3D, image, text, time series, Web, etc.
    - kit de développement DBDK
  - performance
    - exécution des Datablades dans le processus SGBD (exige une procédure de validation)
    - fonctions de coût des Datablades pour l'optimiseur

Sybase

- Adaptive Server : extension objet de System 11
  - support des types de données complexes
    - texte, video, audio, etc.
  - extensibilité des types : snap-in’s
    - intégration de certaines fonctions : séries temporelles
    - fonctions spécialisées : géospatial
  - support de SQL3 pour les types abstraits

GemStone (GemStone Corp.)

- Premier SGBDOO commercialisé (1986)
  - applications techniques et gestion
  - modèle tout objet (extension de Smalltalk)
  - version Java GemStone/J
  - interface C++
  - support du travail coopératif
  - architecture répartie
  - interfaces avec SGBDR
  - support CORBA et Web
- Points forts
  - ramasse-miette, architecture répartie hétérogène

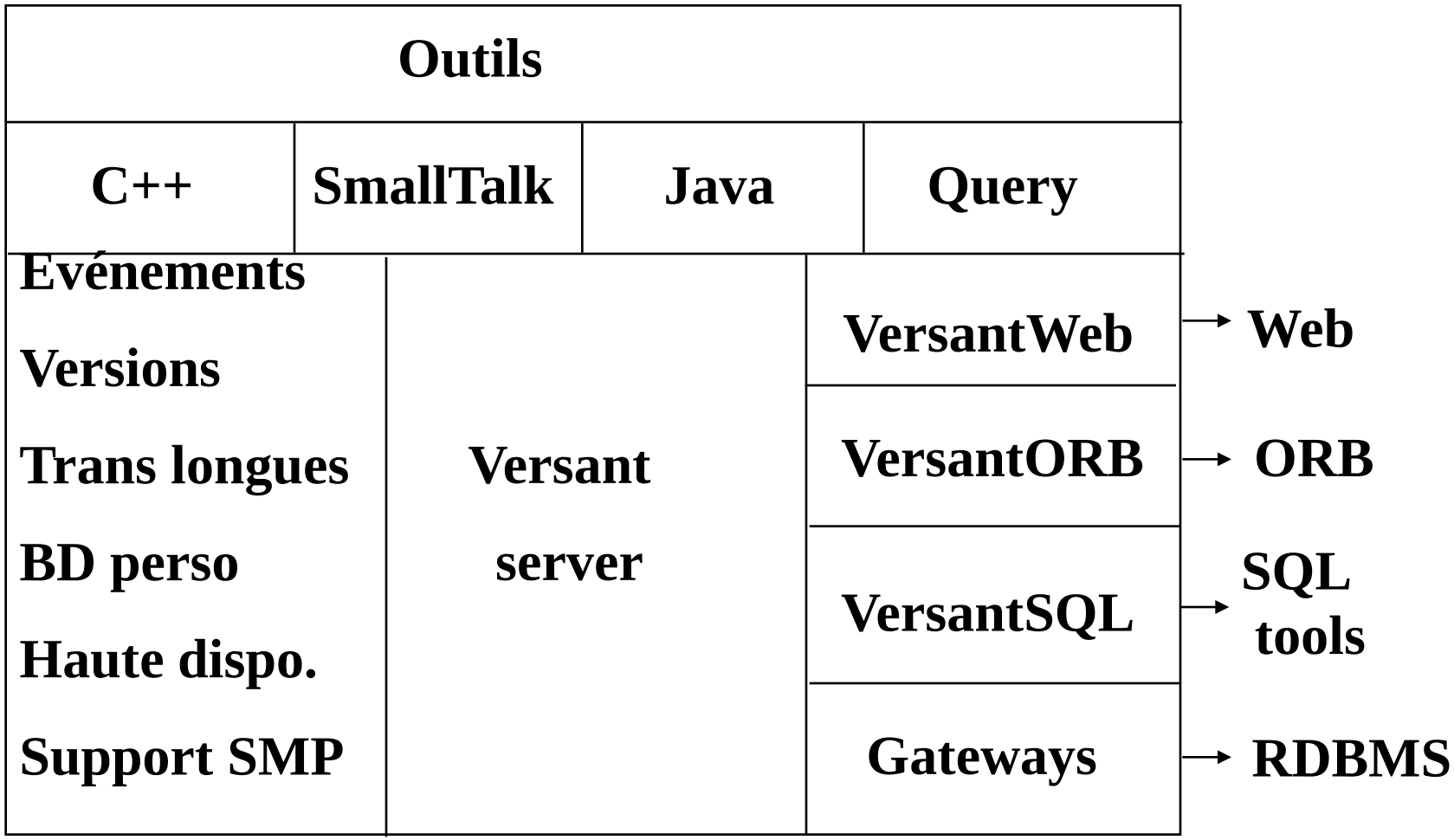
ObjectStore (Object Design Inc.)

- Commercialisé en 1990 (Nasdaq 1996)
  - imposé dans les applications techniques
  - extension de C++ avec objets persistants
  - versions Smalltalk et Java
  - support du travail coopératif
  - distribution et accès aux BDR
  - support CORBA et Web
- Points forts
  - intégration langage, liens, versions, cache d'objet sur le client

Versant (Versant Object Technology)

- Commercialisé en 1990 (Nasdaq 1996)
  - applications techniques, gestion et Internet/Intranet
  - multilangage (SQL/X, C++, Smalltalk, Java)
  - architecture répartie
  - intégration CORBA : Orbix, Powerbroker, NEO
  - VersantWeb : gestion de sessions BD
  - support SMP
- Points forts
  - grandes BD, haute disponibilité, outils de développement

Architecture de Versant



Techniques

- Architecture serveur d'objets
  - cache objet sur le client avec OID logiques
  - accès multithread
    - exclusion mutuelle ou verrous BD
- Evolution de schéma dynamique
- Gestion d'objets
  - placement sur disque par proximité
  - index de chemins
  - ramasse-miette

Techniques (suite)

- Transactions
  - contrôle de concurrence optimiste
  - longues transactions
  - validation en deux phases
- BD personnelle
  - check-in/check-out avec déconnection
- Versions
  - API pour définir des versions d'objets
    - dérivation et fusion de versions
- Evénements
  - gestion asynchrone d'événements BD par un modèle "publish-and-subscribe"