



Université du Québec


École de technologie supérieure

Architecture et performance des bases de données MM

Plan de la présentation

- **Introduction (8.1)**
- Spécificités des serveurs vidéo (8.1.1)
- Architectures BDMM (8.1.2 à 8.1.6)
- Performance (8.2)
- Gestion du contenu (8.3)
- Serveur vidéo d'Oracle
- Exemples d'architectures

Plan de la présentation

- Introduction (8.1)
 - Spécificités des Serveurs Vidéo (8.1.1)
 - Architectures BDMM (8.1.2 à 8.1.6)
 - **Performance (8.2)**
 - Gestion du contenu (8.3)
 - Serveur vidéo d'Oracle
 - Exemples d'architectures
- 
- Optimisation de l'espace de table
 - Optimisation de la requête
 - Politique de répartition des données (stripping policy)
 - Équilibrage des charges (load balancing)
 - Contrôle des accès
 - Céduteur de disques
 - Serveur, proxy et caches

Le test ACIDL

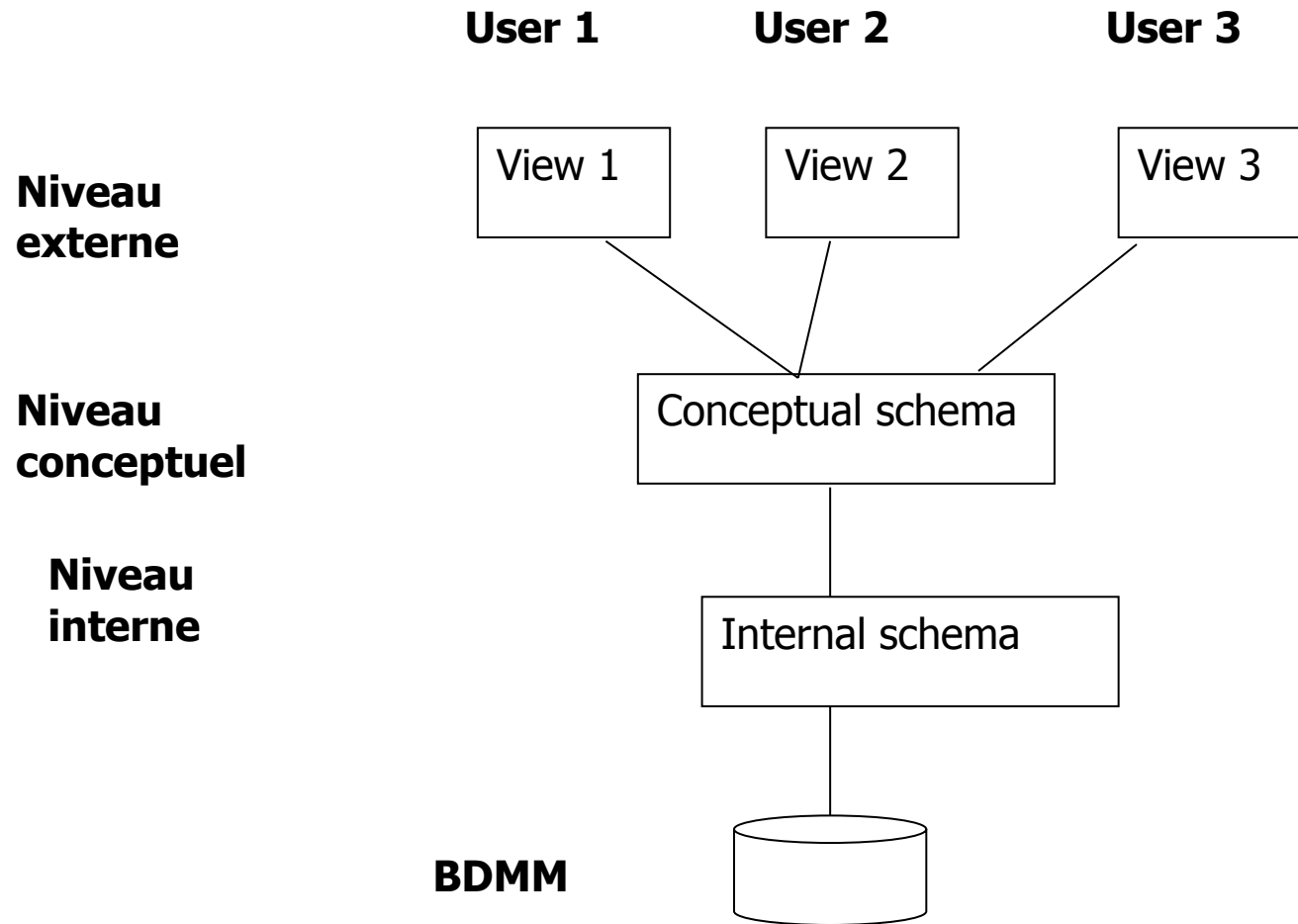
- Atomicité : Une transaction est une unité indivisible traitée dans son ensemble ou pas du tout
- Consistance : Une transaction doit permettre de passer d'un état consistant à un autre
- Indépendance (isolation) : Une transaction ne doit pas être consciente des effets des autres transactions incomplètes
- Durabilité : Une transaction complétée doit être enregistrée d'une manière permanente dans une base de donnée
- Légalité : Une transaction doit être autorisée en conformité avec les droits d'accès accordés au requérant

Architecture à 3 niveaux ANSI-SPARC

- Le **niveau externe** représente la vue utilisateur des besoins d'affaire et consiste en un certain nombre de perspectives différentes sur les mêmes données (*niveau conceptuel*)
- Le **niveau conceptuel** est la vue de l'administrateur des données et permet d'expliquer les contraintes, les relations, la sémantique et la sécurité des données (*schéma logique*)
- Le niveau **interne** se préoccupe de la manière de représenter, accéder et conserver l'intégrité des données. (*niveau physique*)

Fig 8.1

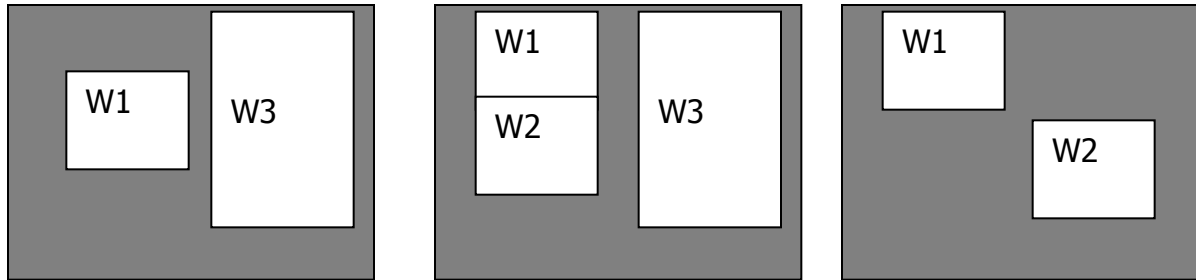
Architecture



Options architecturales BDMM

- **BDMM centralisées** tournent sur un seul ordinateur qui n'a aucune interaction avec d'autres ordinateurs
- **BDMM client/serveur** est un réseau d'ordinateurs (à 2 ou 3 couches) qui permet la division du travail entre le client (présentation) et le serveur (services de données)
- **BDMM réparties** permet la distribution géographique et l'administration de données réparties sur plusieurs ordinateurs

**Vue
usager**



**Vue
distribuée**



**Vue
conceptuelle
des données**



**Vue
physique**

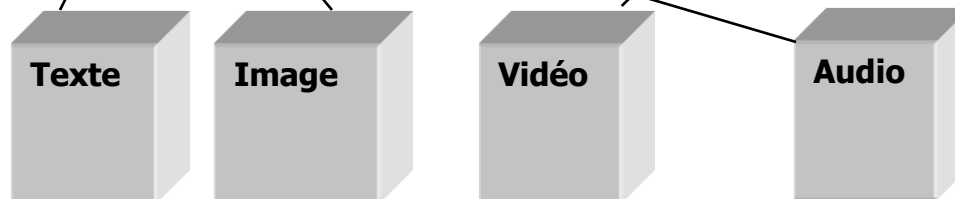


Fig 8.2

Plan de la présentation

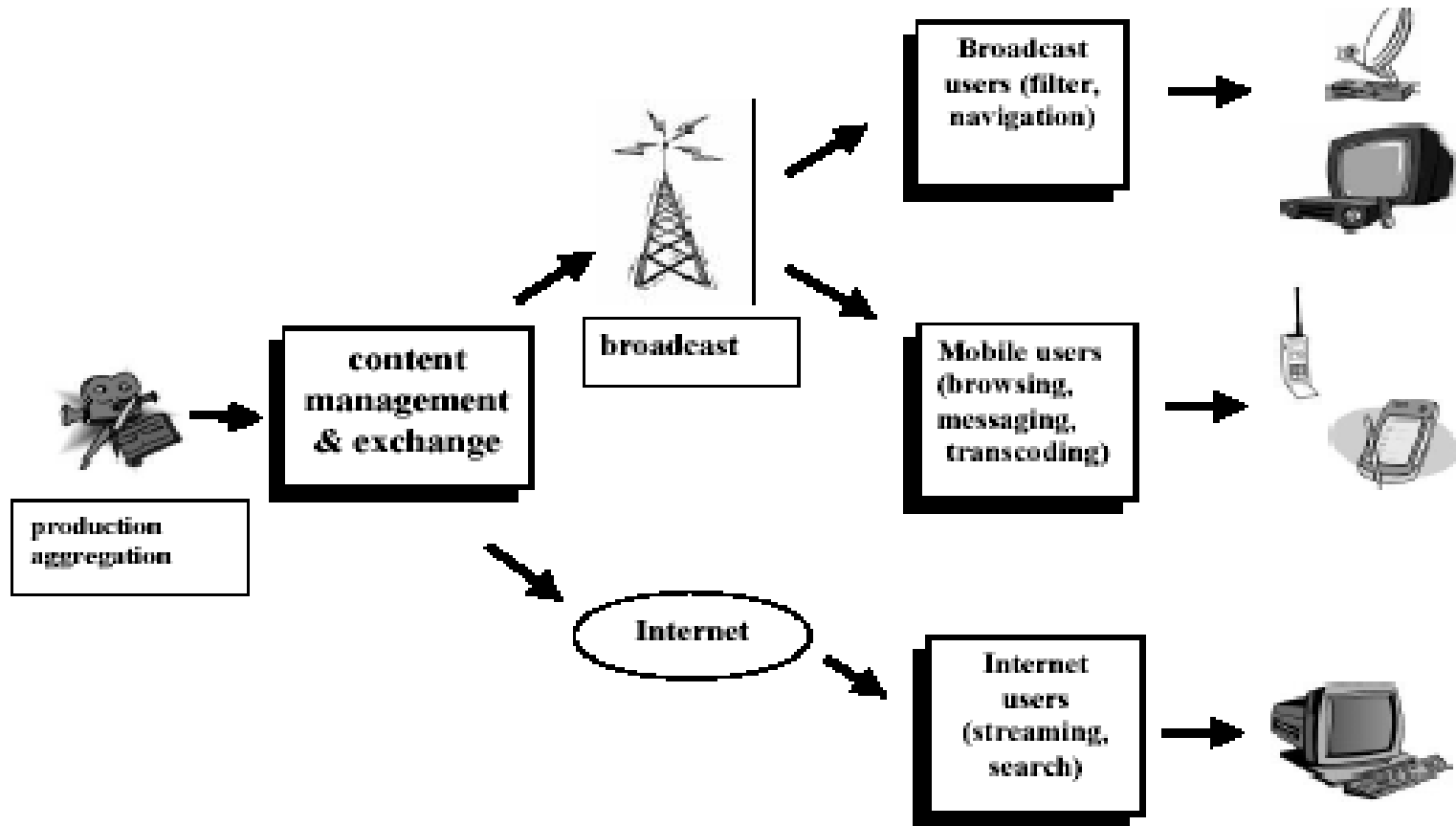
- Introduction (8.1)
- **Spécificités des serveurs vidéo (8.1.1)**
- Architectures BDMM (8.1.2 à 8.1.6)
- Performance (8.2)
- Gestion du contenu (8.3)
- Serveur vidéo d'Oracle
- Exemples d'architectures

BDMM: Besoins utilisateurs

- Besoins utilisateurs

- Accès à l'information multimédia
 - en tout temps
 - de partout (maison, bureau, auto, sur la rue)
 - sur tout type de terminal
 - de toutes sources
- Composant physique Network/device transparent
 - Qualité du service (dégradation graduelle)
 - Outils et interfaces intelligents
 - Transactions automatisées et transparentes

Chaîne d'application BDMM



Besoins utilisateurs BDMM

- Exemple: PDR (Tivo) Approche en tout temps

<http://www.ee.columbia.edu/~sfchang/course/vis/SLIDE/lecture1-distribute.pdf>

- Time-shift, local storage: ~20 hours
- Instant record, live pause, simultaneous record/playback
- Search/retrieval, multi-source comparison/summarization, bookmarking
→ Indexing !!!! (MPEG-7 contr.)
- Personal profile, multi-user profile
- Target services, ads, consumer usage data
- Pay per choice, e-Commerce



Spécifications des serveurs MM

- Temps réponse minimal
- Processeurs rapides/ temps rapide d'accès
- Fiabilité et accessibilité
- Habileté à soutenir/garantir un nombre de sessions
- Livraison du service en temps-réel
- Haute capacité
- Qualité de Service (QoS)
- Exploite les patrons d'accès des clients
- Pas trop cher à exploiter (donc à facturer)

Avantages BD réparties

Table 8.1

- Structure correspond à l'organisation
- Partageable et autonome localement
- Disponibilité améliorée – données peuvent être situées où elles sont nécessaires
- Fiabilité améliorée – plus d'un serveur est impliqué
- Économique – des composantes individuelles moins chères sont utilisées
- Croissance Modulaire – le système n'est pas limité

Désavantages BD réparties

Table 8.1

- Complexe (répartition des données, transactions réparties)
- Coûts (développement, maintenance, transfert de données)
- Sécurité – difficulté de sécuriser tous les serveurs répartis
- Intégrité – difficile à maintenir
- Peu d'expérience sur le marché

Plan de la présentation

- Introduction (8.1)
- Spécificités des serveurs vidéo (8.1.1)
- **Architectures BDMM (8.1.2 à 8.1.6)**
- Performance (8.2)
- Gestion du contenu (8.3)
- Serveur vidéo d'Oracle
- Exemples d'architectures

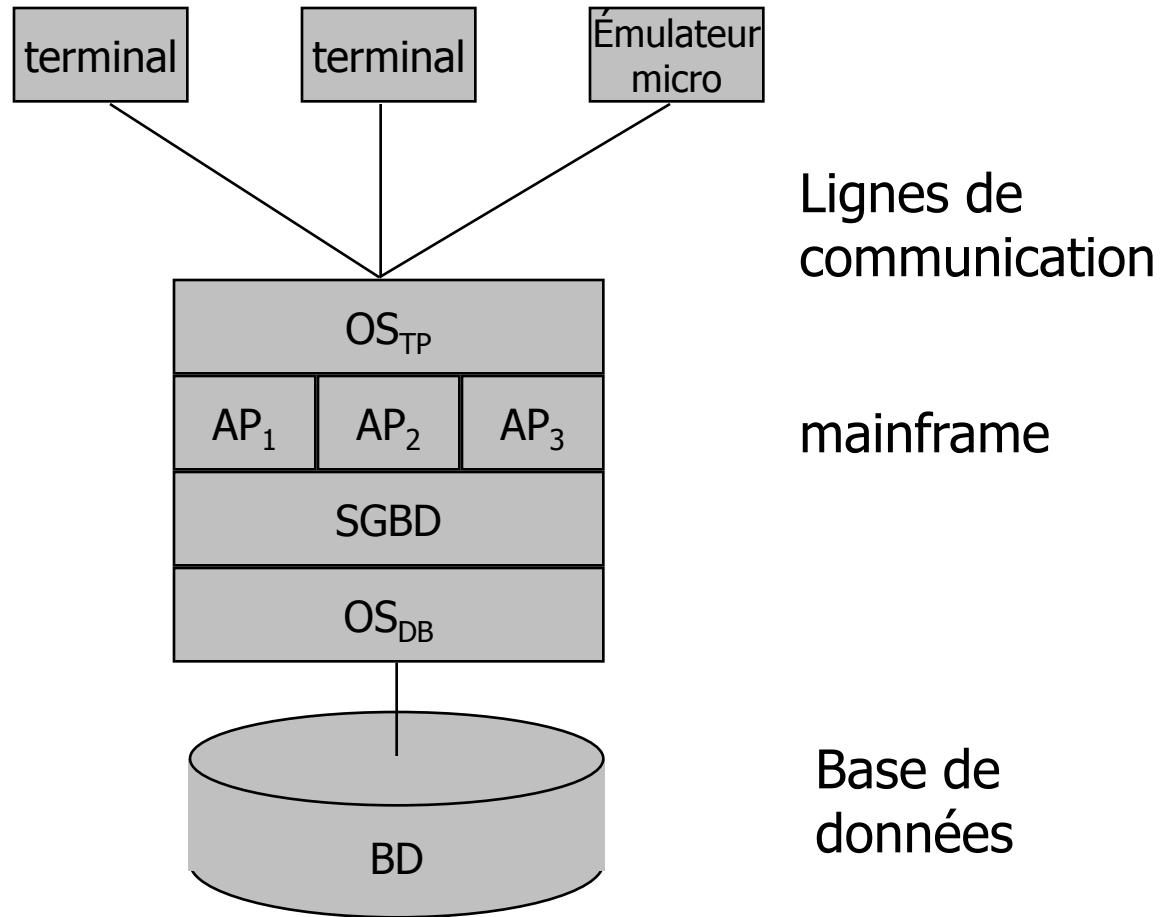
Croissance de BDMM (8.1.2)

- **Croissance du nombre d'utilisateurs**
 - La performance des protocoles multimédia a tendance à se détériorer quand le nombre d'utilisateurs augmente
- **Taille des objets multimédias**
 - La taille des BD audio et vidéo engorge les réseaux et diminue la capacité des E/S, résultant en problèmes de croissance
- **Volume des données disponibles**
 - L'augmentation du volume de données alourdit les accès, les repérages et la gestion des données
- **Distribution des requêtes pas uniforme**
 - Les requêtes des utilisateurs ne parviennent pas d'une manière prédictible et uniforme (dans la journée)

Les architectures de BDMM

- Approches anciennes (centralisée, partage de fichiers, ...)
- Le concept du Super Serveur (8.1.3)
- Le Client/Serveur (8.1.4)
- Le Peer-to-Peer (P2P) (8.1.5)
- Le distribué et l'hétérogène (8.1.6)
- L'approche fédérée
- L'approche parallèle

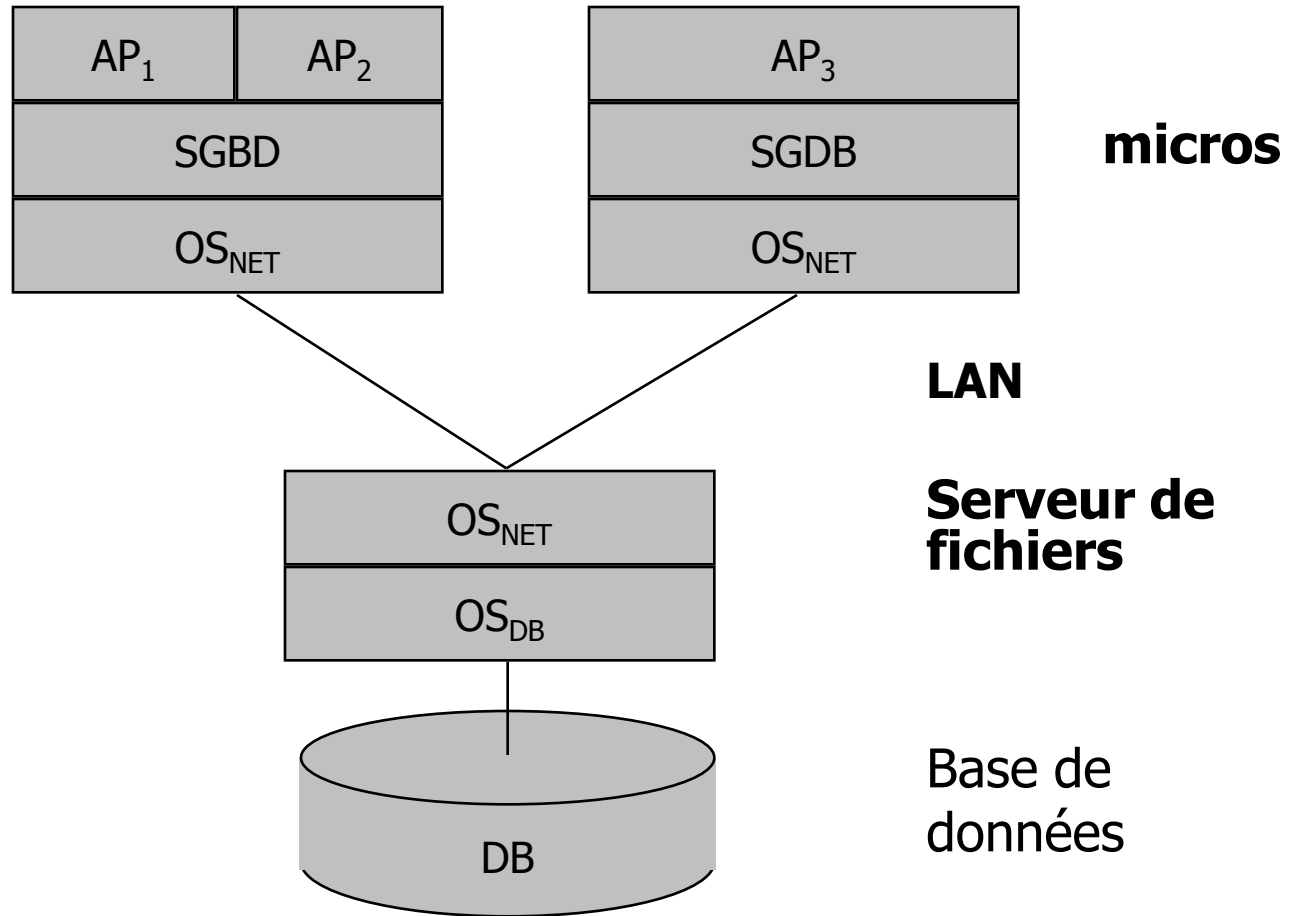
Base de donnée centralisée



Caractéristiques – BD centralisées

- Accès de terminaux non-intelligents et quelques stations clientes en émulation
- Applications et BD résident sur un seul serveur centralisé
- Lignes de communications dédiées (typiquement SNA ou X.25)
- Formatage des écrans est inclu dans le protocole de communication
- Interface utilisateur (caractères seulement)

Base de donnée – Partage de fichier



Caractéristiques – Partage de fichiers

- Applications et fichiers sur la station cliente
- Serveur de fichier sur un serveur indépendant
- Les clients et le serveur communique via un réseau local
- Traffic élevé sur LAN car de gros fichiers (et leurs index) doivent être envoyés aux clients
- Contention d'accès importante
- Bon pour des requêtes sur des images de données stables (snapshot data)
- Mauvais pour des applications transactionnelles

Architecture Super Serveur (8.1.3)

- Un Super Serveur maintient l'information sur
 - la localisation des données en architecture distribuée
 - la charge courante de chaque serveur
 - le trafic courant sur le réseau
- Un Super Serveur peut décider de
 - rediriger une demande de service
 - dupliquer une donnée sur un serveur local à la demande
 - démarrer un autre serveur
- *Architecture Super Serveur*
 - *Distribuée* : tous les serveurs sont des Super Serveur
 - *Centralisée* : quelques serveurs dédiés au rôle de Super Serveur

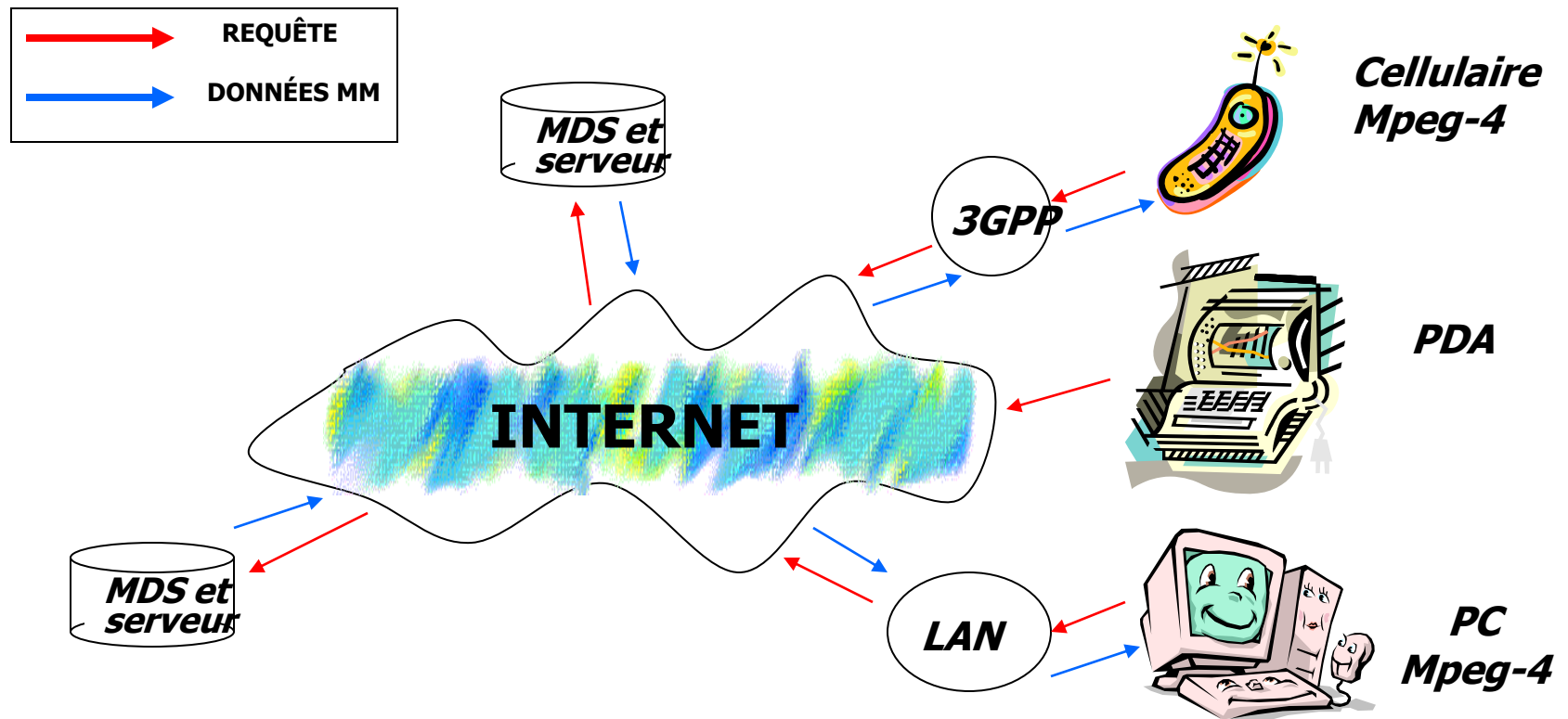
Architecture Client/Serveur (8.1.4)

- *Spécification DataLinks (SQL3) – Gestion des données externes*
 - Permettre la gestion intégrée de données localisées sur un serveur/SGF distant spécialisé en multimédia
 - Ces données multimédia demeurent sous le contrôle du SGBD
 - Intégrité référentielle
 - Accès concurrent et récupération
 - SQL accède à ces données distantes comme si elles faisaient partie de la BD, en utilisant des pointeurs
 - Nouveau type DATALINK pour ces pointeurs
 - Nouvelles fonctionnalités sur le SGF distant

Architecture Client/Serveur

- *Développement d'intergiciels intelligents*
 - Passerelles pour connections entre BD hétérogènes
 - Pour rendre l'hétérogénéité transparente, on développe des logiciels qui transposent requêtes et résultats
 - SQL dynamique
 - SQL/CLI
 - ODBC (Object DataBase Connectivity)
 - Deux standards applicables
 - ISO Remote Data Access (RDA)
 - IBM's Distributed Relational Database Architecture (DRDA)

Un exemple de C/S de Harald Kosh



Faiblesse du C/S pour BDMM

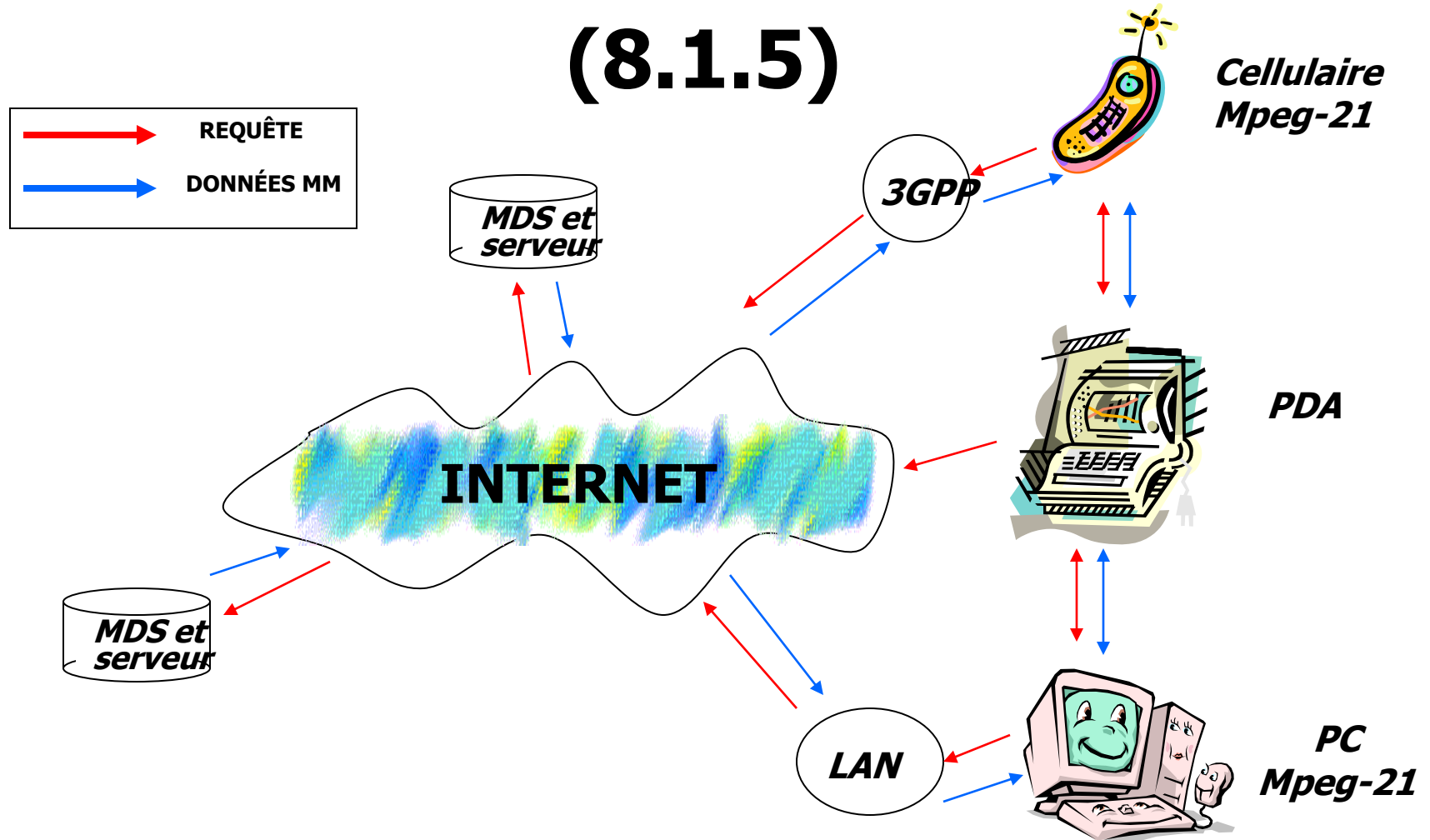
- Pas tous les protocoles (ex: pda ?)
- Parallélisme à son minimum
- Croissance difficile
 - # de clients
 - Terminaux hétérogènes
 - Taille des données
- Typiquement, ces systèmes ont de la difficulté à donner un bon niveau de service

Un vs. plusieurs processeurs symétriques

- Des instances de 'serveurs vidéos' peuvent être assignées à chaque processeur
- Parallélisme élevé versus un seul processeur
- Encore difficile avec l'hétérogénéité

http://www-rohan.sdsu.edu/doc/oracle/video217/A42271_4/adm_conf.htm#421753

Architecture Peer-to-Peer (8.1.5)



Caractéristiques du Peer-to-Peer

- Chaque terminal peut
 - Découvrir dynamiquement les autres terminaux
 - Questionner les autres terminaux pour obtenir du contenu
 - Offrir son contenu aux autres terminaux
- Pas de notion de serveur central
- Se base sur les métadonnées de chaque terminal

Difficultés du Peer-to-Peer

- Peu de clients = peu de nœuds avec qui traiter
- Tous les terminaux ont un même protocole
 - CC-PP (Composite Capability/Preference Profiles) : protocole d'échange du W3C basé sur HTTP et RDF
<http://www.w3.org/TR/NOTE-CCPPexchange>
 - JXTA (JuXTApose, i.e. P2P) : famille de protocoles développés par Sun <http://www.jxta.org/>
- Nouvelle approche d'agents mobiles
- Norme internationale MPEG-21
- Actuellement il y a de la recherche dans la QoS (Quality of Service)

SGBD réparti et hétérogène (8.1.6)

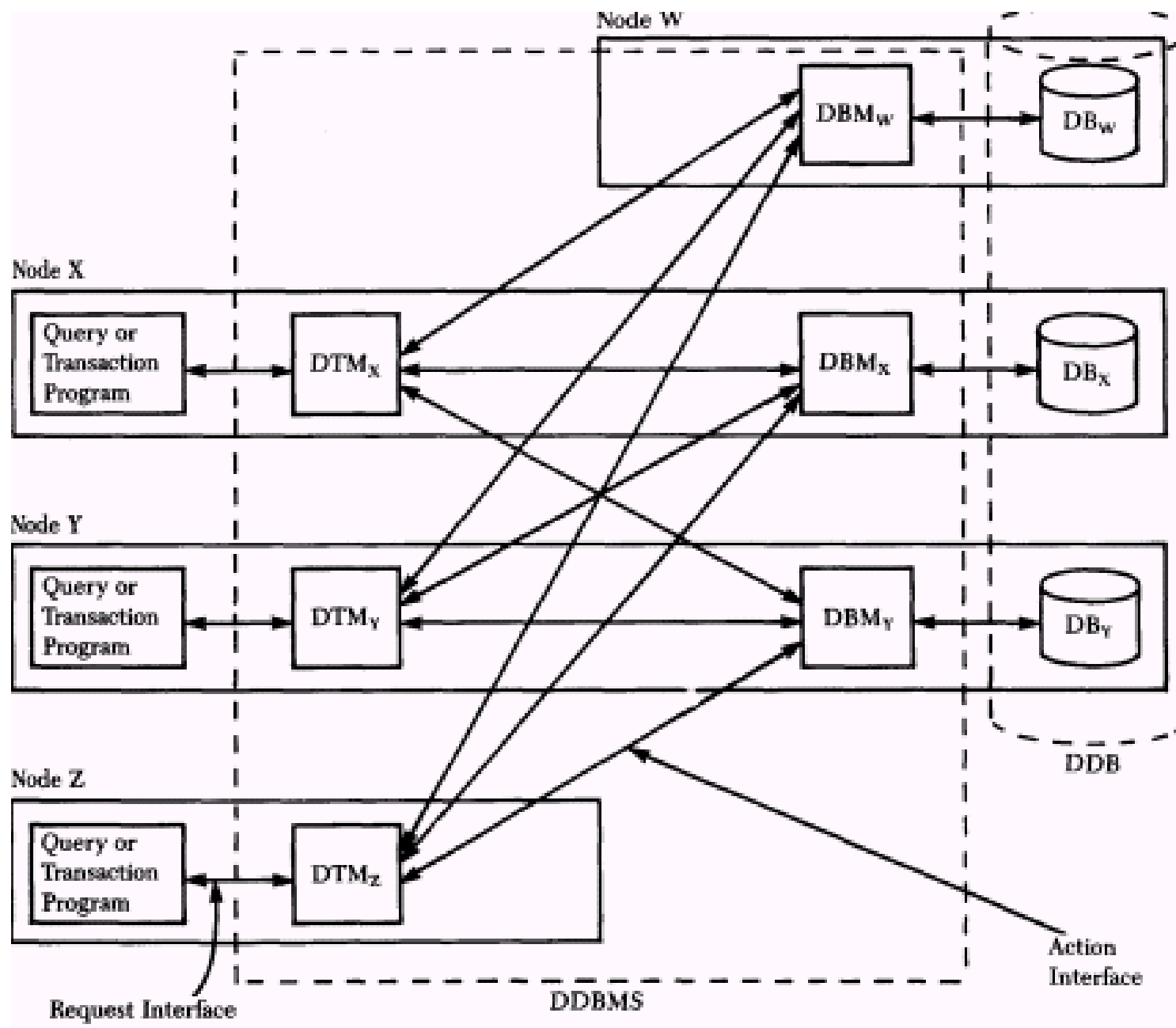
- SGBD répartis
 - Instances distinctes de SGBD sur des serveurs de données
 - BD répartie : fragments horizontaux, verticaux, duplication
- SGBD répartis homogènes: même SGBD et modèle de données sur tous les sites
- SGBD répartis hétérogènes
 - Différents SGBD
 - Différents modèles de données (relationnel, objet)
 - Différentes plateformes / OS

Transparence de la répartition des BD

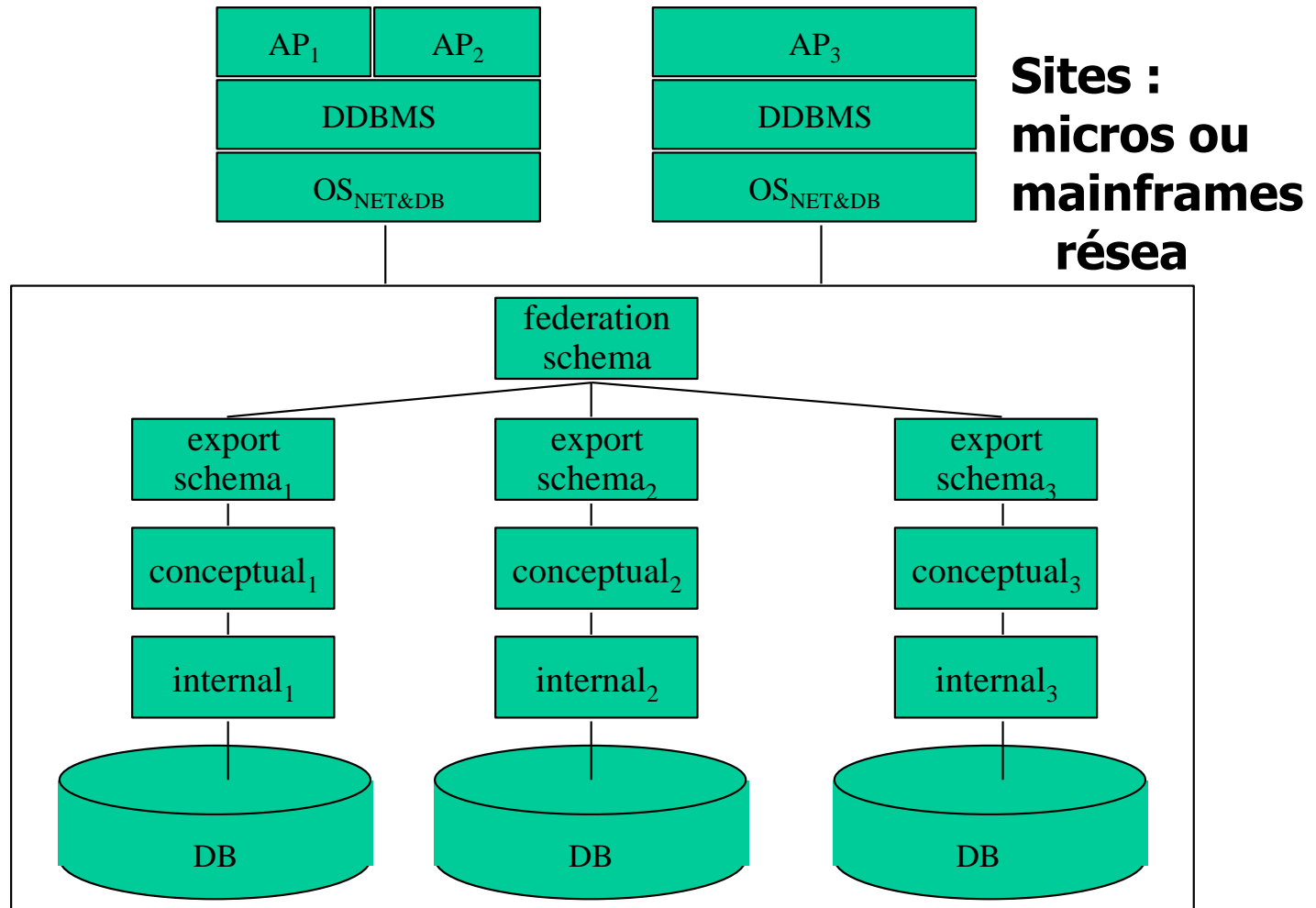
- La fragmentation et/ou duplication d'une BD doivent être transparentes à l'utilisateur
 - Transparence de la localisation des données
 - Gestion transparente des transactions
 - Transparence de la performance
 - Transparence de l'hétérogénéité

3 niveaux de transparence

- Haut: Transparence de la fragmentation
 - Un nom logique pour tous les sites
- Moyen: Transparence de la localisation
 - Un nom logique pour chaque site
- Bas: Transparence d'adresse locale
 - Utilise l'adresse et le nom de la BD



BD fédérée



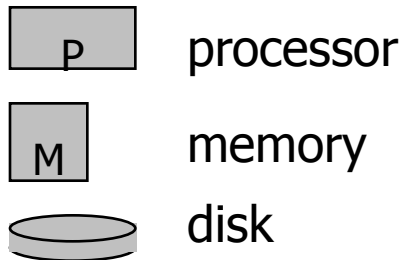
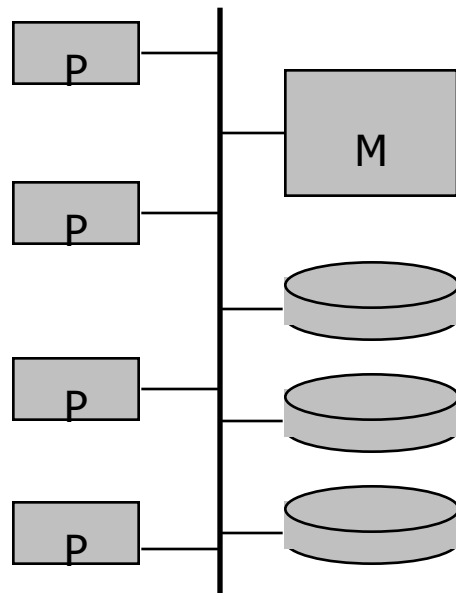
Caractéristiques – BD fédérée

- Chaque SGBD fédéré a un ensemble d'applications, une BDMM distribuée et une BD locale
- Une partie de la BDMM fédéré est **exportée**, i.e. accessible à la fédération
- L'union des BDMM fédérées constitue la BDMM fédérée
- Chaque BD de la fédération **doit** répondre lors d'une requête ou mise à jour d'une autre BDMM
- Chaque BD de la fédération a **plus d'autonomie** qu'une BD répartie typique

BD parallèle

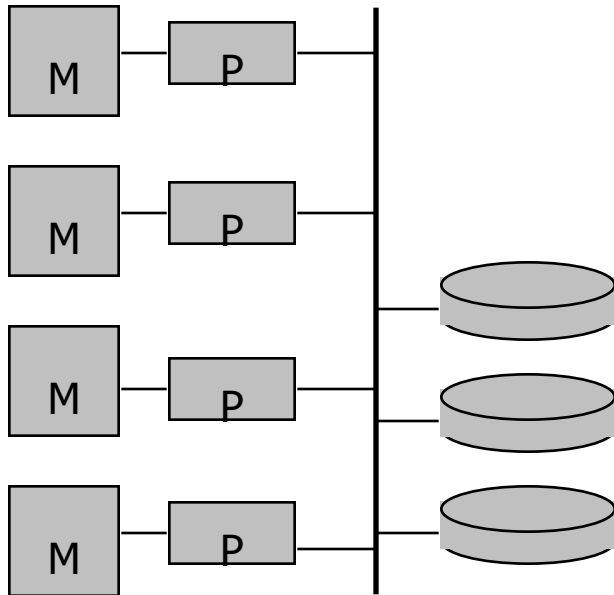
- Définition: BD qui permet à une requête de s'exécuter en parallèle sur plusieurs processeurs
- Trois architectures populaires
 - Mémoire partagée
 - Disques partagés
 - Seulement les processeurs en commun

BD parallèle – Mémoire partagée



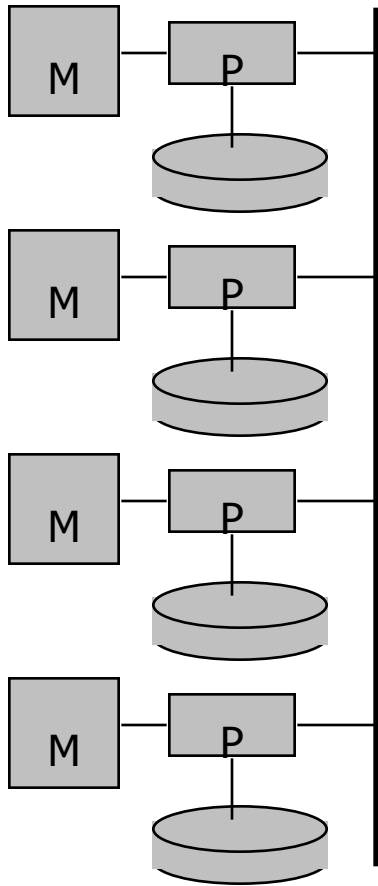
- processeurs partagent la mémoire via un bus
- très performant
- communication des processeurs via écriture mémoire
- Contention au bus

BD parallèle – Disques partagés



- processeurs partagent les disques via réseau
- Bus mémoire sans contention
- Tolérance aux fautes
- Meilleure garantie qu'avec la mémoire partagée
- Contention du réseau d'accès disque
- Utilisé par ORACLE

BD parallèle – Sans partage



- Meilleure garantie que mémoire et disques partagés
- Principaux problèmes:
 - Coûts de communication inter-processeurs
 - Coûts d'accès disque distant
- Architecture de NCR dans la BD Teradata

Un serveur BD répartie doit fournir

- Catalogue système étendu
 - localisation des données
- Traitement des requêtes distribuées
 - Incluant l'optimisation des requêtes distribuées
- Gestion de la concurrence avancée
 - Protocoles de confirmation, d'expiration et de récupération
- Fonctions d'archivage et de recouvrement avancées

Influence des programmes d'application dans la conception

- Le site de l'application qui va utiliser la BDMM
- La fréquence d'utilisation de l'application
- Le nombre, le type et la distribution statistique des accès faits par chaque application à chaque objet multimédia de la BDMM

Objectifs de conception BDMM

- Localisation du traitement: situer les données le plus près possible de l'application de l'utilisateur
- Disponibilité et fiabilité des données réparties: utilisez plus d'une copie
- Distribution de la charge de travail: prenez une approche parallèle
- Coûts d'emmagasinement et de disponibilité des données

Requêtes sur BDMM hétérogène

- Si les SGBD sont similaires, mais le matériel différent: simple conversion (code, longueur de mot mémoire)
- Différent SGBD: mapping compliqués des structures de données d'un modèle à l'autre
- Si les deux sont différents, on doit faire les deux ajustements ci-haut

Problèmes avec passerelles pour BDMM hétérogènes

- Attention: les passerelles et routeurs ne supportent pas toujours la gestion des transactions entre deux SGBD différents
- Provision pour un modèle conceptuel commun
 - intégration des schémas locaux doit être faite manuellement

Plan de la présentation

- Introduction (8.1)
- Spécificités des serveurs vidéo (8.1.1)
- Architectures BDMM (8.1.2 à 8.1.6)
- **Performance (8.2)**
- Gestion du contenu (8.3)
- Serveur vidéo d'Oracle
- Exemples d'architectures

La performance (8.2)

- **Optimisation de l'espace de table**
- Optimisation de la requête
- Répartition cyclique des données
- Équilibrage des charges
- Contrôle des accès
- Céduteur de disques
- Serveur, proxy et cache

Optimisation de BDMM (8.2.1)

- Optimisation du plan d'exécution d'une requête
 - spécifie l'ordre d'exécution des opérations élémentaires (sélections, projections, jointures, tri, etc.)
 - Dépend des opérations, des statistiques de tables et d'accès, des indexes et des verrous
- Premiers optimiseurs à base de règles (heuristiques) à partir de l'algèbre relationnelle (RB – Rule Based)
 - e.g. projections et sélections avant jointures
- Approche récente par coûts (CBO – Cost Based Optimization)
 - prends en comptes les statistiques (paquetage DBMS_STATS)
- Combiner les deux approches

Gestion de l'espace de table (8.2.2)

- Le schéma interne (physique) d'une BD est stocké dans plusieurs tables
- On utilise les termes **partition**, **tablespace**, **database space** ou **dbspace** pour y référer
- Sous Oracle, les objets physiques sont stockés comme **segment**
- Un **segment** contient plusieurs **extents** (espace disque) à l'intérieur d'un même **tablespace**

TABLESPACE *de Robert Godin*

- DATABASE = ensemble de TABLESPACE

- minimum 1 (SYSTEM TABLESPACE)
 - contient le dictionnaire de données
- peut ajouter des TABLESPACE

```
CREATE TABLESPACE des 2 DATAFILE 'c:\orawin\dfs\WDBdes2.ora'  
    SIZE 14M DEFAULT STORAGE (PCTINCREASE 0) ONLINE;
```

- TABLESPACE

- espace pour TABLE (et autres objets)
- = ensemble de DATAFILE

```
ALTER TABLESPACE SYSTEM ADD DATAFILE  
    'c:\orawin\dfs\WDBSYS1.ora' SIZE 60M REUSE
```

Clauses TABLESPACE et STORAGE *de Robert Godin*

```
CREATE TABLE Client...
```

```
  INITRANS      1
```

```
  MAXTRANS      5
```

```
  PCTFREE       10
```

```
  PCTUSED       40
```

```
  TABLESPACE   user_data
```

```
  STORAGE (
```

```
    INITIAL      20480
```

```
    NEXT         20480
```

```
    MINEXTENTS   1
```

```
    MAXEXTENTS   10
```

```
    PCTINCREASE  10)
```

Taille des
EXTENTS



Paramètres de disques Oracle

Table 8.3

- **INITIAL:** Taille de l'allocation primaire
= taille moyenne / maximale estimée de la table
- **MAXEXTENTS:** Nombre maximum d'allocations secondaires permises par segment
- **PCTINCREASE:** Contrôle la croissance et la fragmentation
- **PCTFREE:** Espace réservé par bloc pour la croissance dûe aux mises à jour (UPDATE)
- **PCTUSED:** Contrôle la croissance dûe aux insertions

EXTENT, SEGMENT *de Robert Godin*

- EXTENT

- granule d'allocation d'espace
- ensemble de blocs contigus
- dans un DATAFILE du TABLESPACE

- Clause STORAGE de défaut

- niveau TABLESPACE
- niveau DATABASE

- SEGMENT

- ensemble de EXTENT d'un objet
 - (DATA, INDEX, ROLLBACK, TEMPORARY SEGMENTS)



Exemple (Page 275)

```
CREATE TABLE employee
(employee_number  CHAR(4) ,
 employee_name    VARCHAR2(30) ,
 salary          NUMBER(6,2) ,
 start_date       DATE ,
 CONSTRAINT prim_emp PRIMARY KEY(employee_number)
 USING INDEX
 PCTFREE 2
 STORAGE (INITIAL 1M MAXEXTENTS UNLIMITED
          PCTINCREASE 0)
 TABLESPACE users)
```

Conception des TABLESPACE

de Robert Godin

1. Unité de base de prise de copie
2. Sélectivement désactivés (offline)
3. TABLESPACE sur des disques différents
 - réduire congestion
4. EXTENT de mêmes tailles ensemble
5. Limiter la fragmentation (PCTFREE)
 - TABLESPACE de défaut
 - associé au *authorizationID*

Paramètres pour données LOB (Page 280)

- **Clause LOB**
 - stocker un LOB dans un tablespace séparé
 - permettre l'accès simultanée à plusieurs colonnes
- **Paramètre CHUNK**
 - taille des allocations pour manipuler des LOB
- **Paramètre PCTVERSION**
 - pourcentage maximum de l'espace total d'un LOB utilisé pour la création de nouvelles versions
- **Paramètre NOCACHE**
 - interdit l'utilisation de l'antémémoire pour stocker des données LOB

La performance (8.2)

- Optimisation de l'espace de table
- **Optimisation de la requête**
- Répartition cyclique des données
- Équilibrage des charges
- Contrôle des accès
- Céduteur de disques
- Serveur, proxy et cache

Optimisation de la requête

BDMM *de Robert Godin*

- Cas Oracle

- Outils

- EXPLAIN PLAN
 - SET SQL_TRACE = true;
 - SQL ANALYSE
 - Oracle EXPERT

Indices (hints) *de Robert Godin*

```
SELECT /*+ RULE*/ nom  
FROM Client  
WHERE noClient = 10 ;
```

```
SELECT /*+ INDEX(EMPLOYÉ INDEX_SEXE)*/ nom, adresse  
FROM EMPLOYE WHERE SEXE = 'F'
```

Paramètre OPTIMIZER_MODE

de Robert Godin

- ALTER SESSION SET OPTIMIZER_GOAL =
RULE | ALL_ROWS | FIRST_ROWS | CHOOSE
 - RULE : règles et heuristiques
 - ALL_ROWS : CBO – prends en compte la taille des tables
 - FIRST_ROWS : CBO – ne prends pas en compte la taille des tables
 - CHOOSE (défaut): RULE si aucune statistiques; CBO autrement

Exemple d'utilisation de **EXPLAIN PLAN** *de Robert Godin*

SQL> start utlxplan.sql *Pour créer la table plan_table*
Table created.

...

SQL> run

1 explain plan

2 set statement_id = 'com'

3 for select * from commandes, lignes_de_commande

4* where no_commande = commande_no_commande

Explained.

Suite *de Robert Godin*

SQL> run

```
1 SELECT LPAD(' ',2*(LEVEL-1))||operation||' '||options
2 ||' '||object_name
3 ||' '||DECODE(id, 0, 'Cost = '||position) "Query Plan"
4 FROM plan_table
5   START WITH id = 0 AND statement_id = 'com'
6*  CONNECT BY PRIOR id = parent_id AND statement_id
   ='com'
```

Query Plan

```
-----
SELECT STATEMENT   Cost =
  NESTED LOOPS
    TABLE ACCESS FULL LIGNES_DE_COMMANDE
    TABLE ACCESS BY ROWID COMMANDES
    INDEX UNIQUE SCAN COMMANDE_PK
```

SQL Analyse - Affichage du plan d'exécution

The screenshot shows the Oracle SQL Analyze interface. The title bar is "Oracle SQL Analyze - SYSMAN@uqam-mtks51cp9t". The menu bar includes File, Edit, View, SQL, TopSQL, History, Tools, and Help. The toolbar contains various icons for file operations and analysis. The left pane shows a tree view of databases, with "ora817i" expanded to show "SQL002" and "Rule". The main pane displays the SQL statement: "SELECT DISTINCT noarticle FROM commande, lignecommande WHERE commande.nocommande = lignecommande.nocommande AND noclient = 10". The schema is "GODIN". Below the SQL statement, the "Explain" tab is selected, showing the execution plan. The plan is a tree structure with the following steps and order:

| Execution Step | Order |
|--|-------|
| SELECT STATEMENT | 6 |
| SORT (UNIQUE) | 5 |
| NESTED LOOPS | 4 |
| TABLE ACCESS (BY INDEX ROWID) OF 'GODIN.COMMANDE' | 2 |
| INDEX (RANGE SCAN) OF 'GODIN.INDEXCOMMANDENOCLIENT...' | 1 |
| INDEX (RANGE SCAN) OF 'GODIN.SYS_C001705' (UNIQUE) | 3 |

The bottom of the window has a status bar with "For Help, press F1" and "OVR" buttons.

SQL Analyse - Consommation de ressources

The screenshot shows the Oracle SQL Analyze tool interface. The title bar reads "Oracle SQL Analyze - SYSMAN@uqam-mtks51cp9t". The menu bar includes File, Edit, View, SQL, TopSQL, History, Tools, and Help. The toolbar contains various icons for file operations and analysis. The left pane shows a tree view of databases, with "ora817i" expanded to show "SQL002" and its "Rule". The main pane displays the SQL statement: "SELECT DISTINCT noarticle FROM commande, lignecommande WHERE commande.nocommande = lignecommande.nocommande AND noclient = 10". The "Statistics" tab is selected, showing a table of resource consumption metrics.

SQL Statement: SQL002 - Rule Schema: GODIN

```
SELECT DISTINCT noarticle
FROM commande, lignecommande
WHERE commande.nocommande = lignecommande.nocommande
AND noclient = 10
```

Explain Compact View Statistics

Total Fetches: 1

| Statistics | Latest Fetch | Average Over Fetches |
|---------------------------|--------------|----------------------|
| Elapsed Time (in seconds) | 0.08 | 0.08 |
| CPU Time (in seconds) | 0.00 | 0.00 |
| First Row (in seconds) | 0.08 | 0.08 |
| Logical Blocks Read | 5 | 5.00 |
| Physical Blocks Read | 3 | 3.00 |
| Recursive Calls | 0 | 0.00 |
| Database Calls | 4 | 4.00 |
| Chained Rows | 0 | 0.00 |
| Number of Rows Returned | 6 | 6.00 |

For Help, press F1

La performance (8.2)

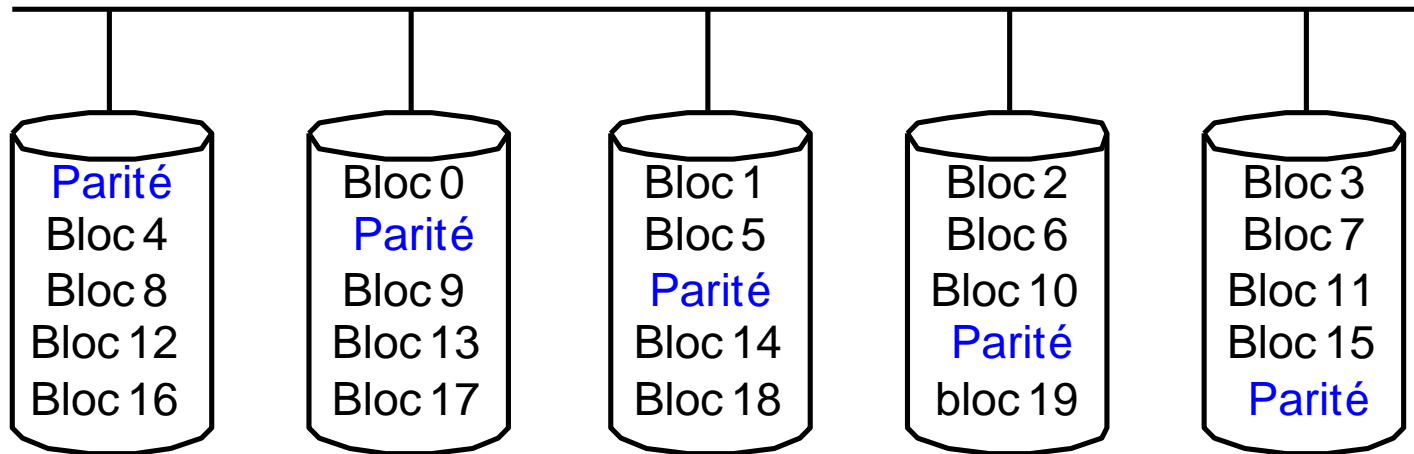
- Optimisation de l'espace de table
- Optimisation de la requête
- **Répartition cyclique des données**
- Équilibrage des charges
- Contrôle des accès
- Céduleur de disques
- Serveur, proxy et cache

Stripping Policy

- Utilisation de disques parallèles : essentiel en BDMM pour la performance
- Architectures RAID (*Redundant Array of Independent Disks*) – pour la performance et/ou la fiabilité
- Stripping : répartition cyclique des données
 - Par bloc : ex. (3 disques) blocs 1,4 sur disque 1; blocs 2,5 sur disque 2; blocs 3,6 sur disque 3
 - Par bit : ex. (8 disques) 1 bit de chaque mot mémoire sur chaque disque
- Pour la fiabilité : avec ou sans code de parité ou CCE (Code Correcteur d'Erreur)
- Le code de parité ou CCE peut aussi être réparti

RAID & STRIPPING *de Robert Godin*

Raid 5 : Répartition par bloc + parité répartie



- Niveaux 0 à 7 : différentes combinaisons
- Certaines implémentations combinent deux architectures RAID : 10, 50, 31, 51, etc.
- Les plus utilisés : 5, 1, 0, 10 et 50
- <http://www.raid-advisory.com/>

La performance (8.2)

- Optimisation de l'espace de table
- Optimisation de la requête
- Répartition cyclique des données
- **Équilibrage des charges**
- Contrôle des accès
- Céduteur de disques
- Serveur, proxy et cache

Équilibrage de la charge BDMM

- À cause des patrons d'accès initialement imprévisibles il est difficile de prévoir les effets du 'stripping'
- Chou *et al* : approche de répartition sur un seul serveur + objets populaires dupliqués sur plusieurs serveurs = approche de BD parallèle

La performance (8.2)

- Optimisation de l'espace de table
- Optimisation de la requête
- Répartition cyclique des données
- Équilibrage des charges
- **Contrôle des accès**
- Céduleur de disques
- Serveur, proxy et cache

Contrôle des accès BDMM

- Pour donner accès aux données
 - Vérifier la disponibilité des ressources
 - Vérifier le nombre de sessions concurrentes
- Hard et Soft existent
 - Hard : durée fixe déterminée par QoS
 - Soft : accepte des tâches qui ne rencontreront pas la QoS

La performance (8.2)

- Optimisation de l'espace de table
- Optimisation de la requête
- Répartition cyclique des données
- Équilibrage les charges
- Contrôle des accès
- **Céduleur de disques**
- Serveur, proxy et cache

Céduleur de disques BDMM

- Algorithmes MM développés pour l'extraction continue des données (streaming/temps réel)
- Objectifs
 - Rencontrer la QoS
 - Minimiser l'espace de tampons
 - Traiter des demandes concurrentes
- Nouvelles techniques
 - EDF
 - GSS
 - Double Buffering

Céduleur de disques BDMM

- EDF (Earliest Deadline First)
 - Algorithme d'ordonnancement préemptif à priorité dynamique (idéal pour les STR)
 - Il attribue une priorité à chaque requête en fonction de l'échéance de cette dernière. Plus l'échéance d'une tâche est proche, plus sa priorité est grande
- GSS (Group-Sweeping Scheduling)
 - Les demandes sont servis en cycles (round-robin)
 - Pour réduire les mouvements des bras des disques (« demandes »), l'ensemble des flux est divisé en groupes
 - Servis en ordre fixe un après les autres réduisant ainsi l'espace tampon

Céduleur de disques BDMM

- Double Buffering
 - Alloue des tampons pour le producteur et le consommateur (disque et réseau)

La performance (8.2)

- Optimisation de l'espace de table
- Optimisation de la requête
- Répartition cyclique des données
- Équilibrage des charges
- Contrôle d'accès
- Céduleur de disques
- **Serveur, proxy et cache**

Serveur, proxy et cache BDMM

- L'antémémoire (cache) est un aspect central de la performance en BDMM
 - Minimiser les accès disque
- Approches classiques ne fonctionnent pas pour les objets multimédias accédés en continu (ex. vidéo sur demande)
 - Antémémoire utilisé pour régulariser le débit
- Solution: antémémoire de proxy
 - Augmenter la taille des antémémoires
 - Contrôler la taille des antémémoires en fonction des variations du débit de consommation par le client

Traitement en parallèle (8.2.4)

- Recommandé dans les cas suivants:
 - Requêtes sur grand volume ($> 1,000,0000$)
 - Jointures sur deux grandes tables ou plus
 - Création d'index volumineux
 - Chargements massif en lot
 - Copie de grandes quantités de données entre types d'objet
 - Données nécessitant de l'espace sur plusieurs disques ou partitions de table

Quelques règles pour BDMM réparties

- Favoriser l'autonomie locale (fédérée)
- Éviter les goulots d'étranglement (site central)
- Assurer la transparence
 - localisation des données
 - fragmentation / duplication
 - matériel / système d'exploitation / réseau
- Traitement des requêtes distribuées
 - Optimisation globale
 - Gestion distribuée des transactions

Plan de la présentation

- Introduction (8.1)
- Spécificités des serveurs vidéo (8.1.1)
- Architectures BDMM (8.1.2 à 8.1.6)
- Performance (8.2)
- **Gestion du contenu (8.3)**
- Serveur vidéo d'Oracle
- Exemples d'architectures

Gestion du contenu

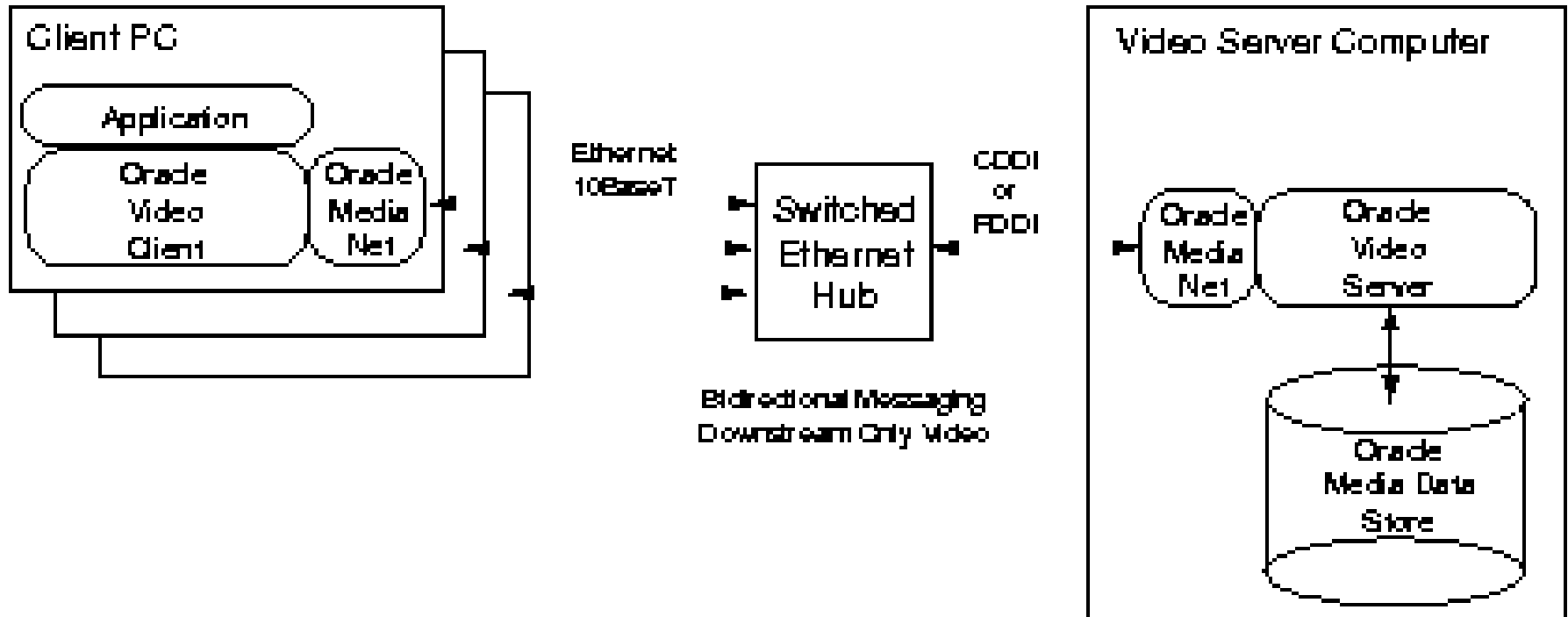
- Serveurs vidéo doivent traiter les documents comme une collection (sons, images et vidéo)
 - Algorithmes d'estimation de bande passante qui cumulent les objets d'une présentation synchronisée
 - Algorithmes présents sur le client et le serveur
 - Regroupements logiques pour la synchronisation des trames
 - Réservation d'accès automatiques (RSVP)
- Optimisation des requêtes doit tenir compte des différentes versions (formats, résolutions) des objets média (semantic rewrite)

Plan de la présentation

- Introduction (8.1)
- Spécificités des serveurs vidéo (8.1.1)
- Architectures BDMM (8.1.2 à 8.1.6)
- Performance (8.2)
- Gestion du contenu (8.3)
- **Serveur vidéo d'Oracle**
- Exemples d'architectures

Oracle Video Server

Figure 1: The Oracle Video Server system



Oracle Video Server

- Oracle Video Client (logiciel sur station client)
 - Décompresse les fichiers MPEG
 - Décode et ré-organise les paquets transmis
 - Contient les librairies de développement
- Oracle Video Server (ordinateur dédié)
 - Emmagazine les vidéos
 - Établit les connections avec les applications clients
 - Livre le vidéo sur demande des applications clients

Oracle Video Server (suite)

- Oracle Media Net (logiciel d'accès)
 - Accès centralisé au MDS (Media Data Store)
 - Système d'adressage indépendant
 - Traite tous les protocoles en couche inférieure
- Oracle MDS
 - Fonctionnalités d'accès disques conçues pour supporter les particularités du multimédia
 - Architecture RAID
 - Répartition par bloc (Stripping)

Administration du Serveur Vidéo d'Oracle

Étapes

- Planification du système
 - Nombre de sessions concurrentes
 - Qualité des vidéos
 - Quantité de données
 - Disponibilité nécessaire
- Codification du contenu des vidéos
- Préparer et charger les vidéos
- Maintenance du MDS
- Surveillance du Serveur Vidéo

Administration du Serveur Vidéo d'Oracle (suite)

- ***Dilemme Niveau de Service***

- Besoin d'un Bit Rate élevé pour une vidéo de qualité (ex: 2Mbps)
- Bit Rate élevé \Rightarrow + bande passante
- + sessions concurrentes \Rightarrow + bande passante

Administration du Serveur Vidéo d'Oracle (suite)

- ***Planifier le nombre de sessions concurrentes en fonction de la bande passante***
 - FDDI (Fiber Distributed Data Interface)
100Mbps : BPConstante 80%
 - = $BPConstante / BRVidéo = \# \text{ Sessions Concurrentes possibles}$
 - Ex: Pour un réseau à BPC de 40Mbps avec des vidéos encodés à 1.5Mbps
 - $\#SC = (80\% * 40) / 1.5 = 48 \text{ sessions}$

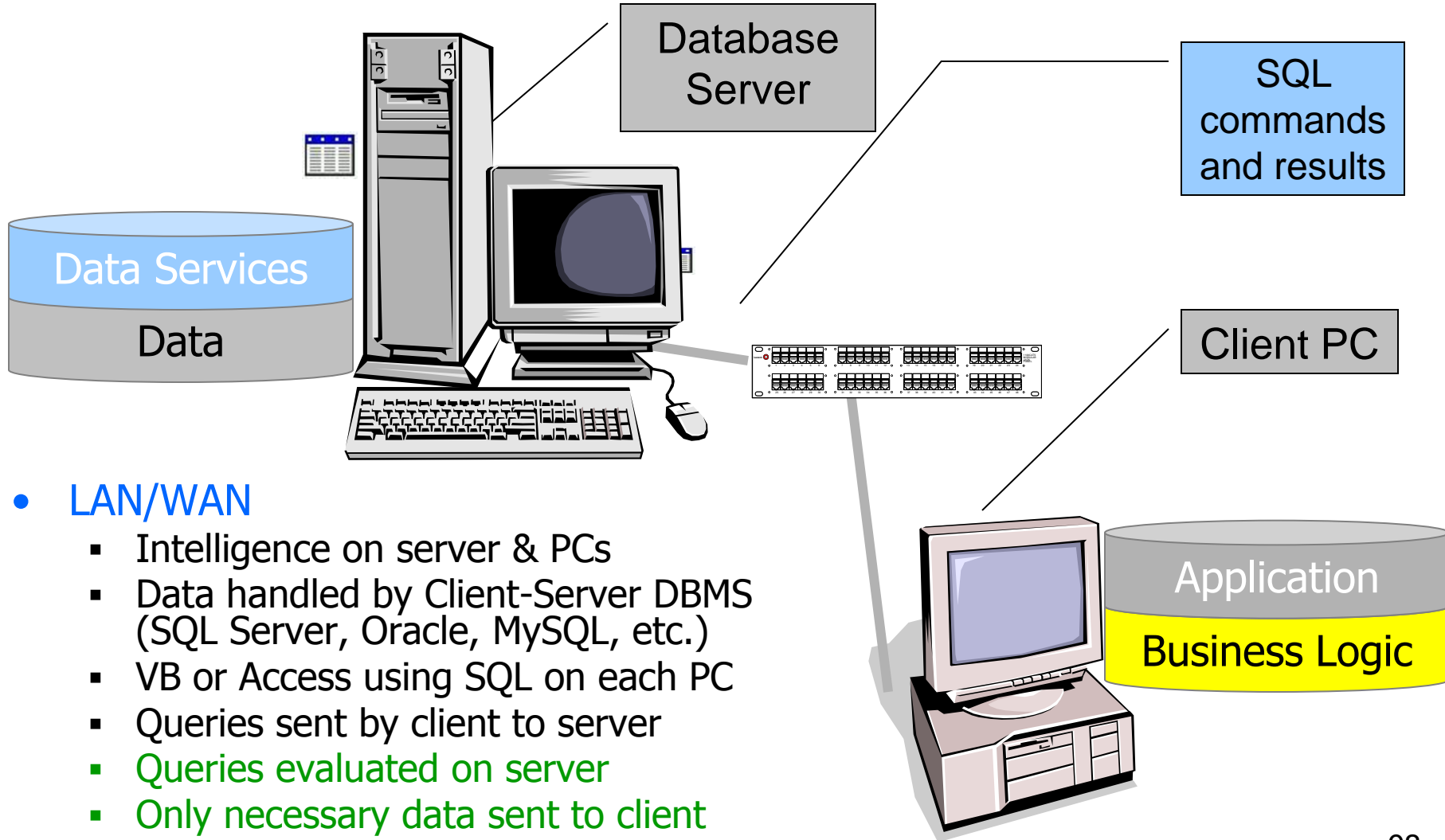
Plan de la présentation

- Introduction (8.1)
- Spécificités des serveurs vidéo (8.1.1)
- Architectures BDMM (8.1.2 à 8.1.6)
- Performance (8.2)
- Gestion du contenu (8.3)
- Serveur vidéo d'Oracle
- **Exemples d'architectures**

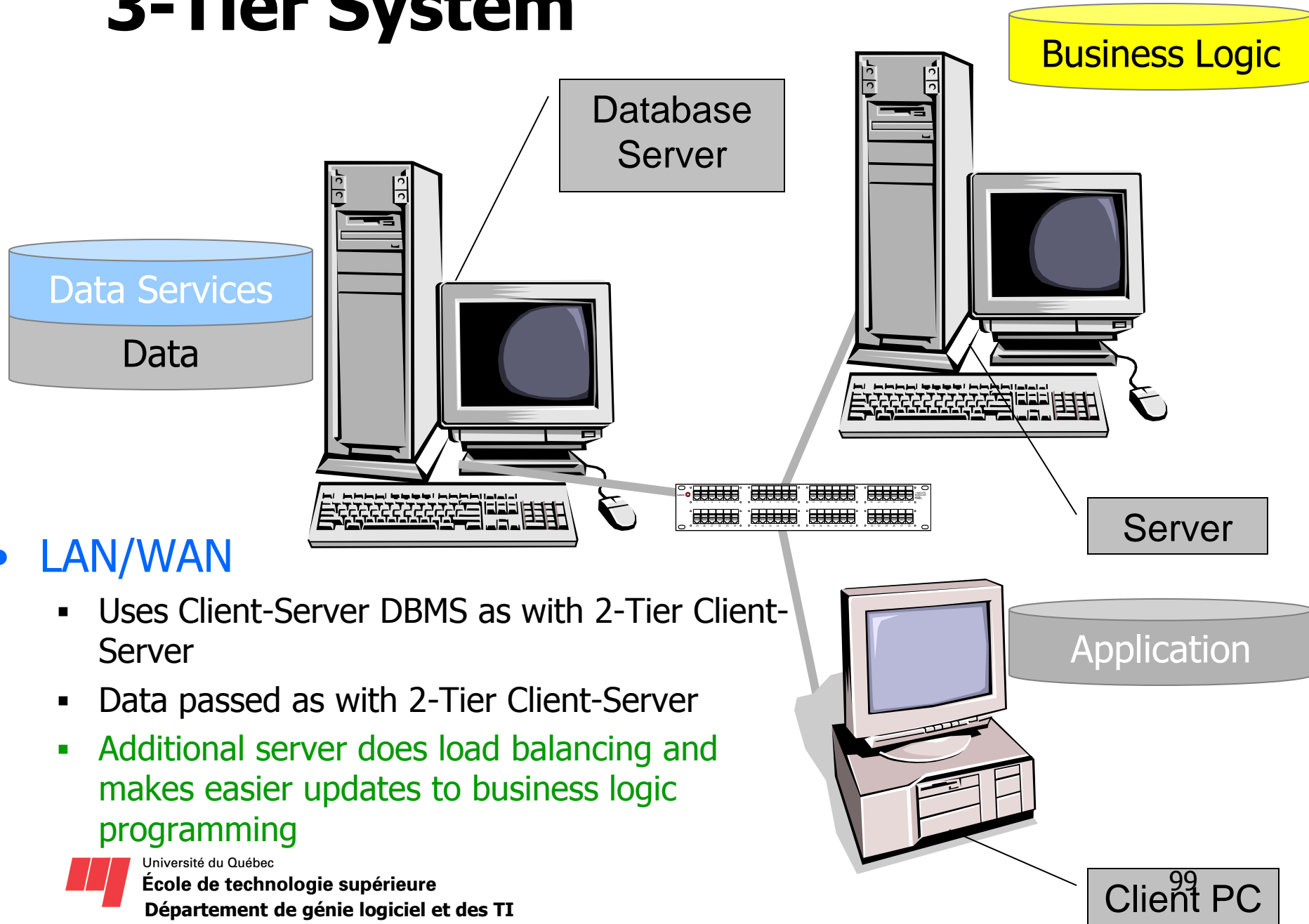
Fat and thin clients

| Type of client | Fat | Thin |
|---------------------|----------------------|----------------------|
| Technology | LAN | Web |
| Application logic | Mostly on the client | Mostly on the server |
| Network load | Medium | Low |
| Data storage | Server | Server |
| Server intelligence | Medium | High |

2-Tier Client-Server System



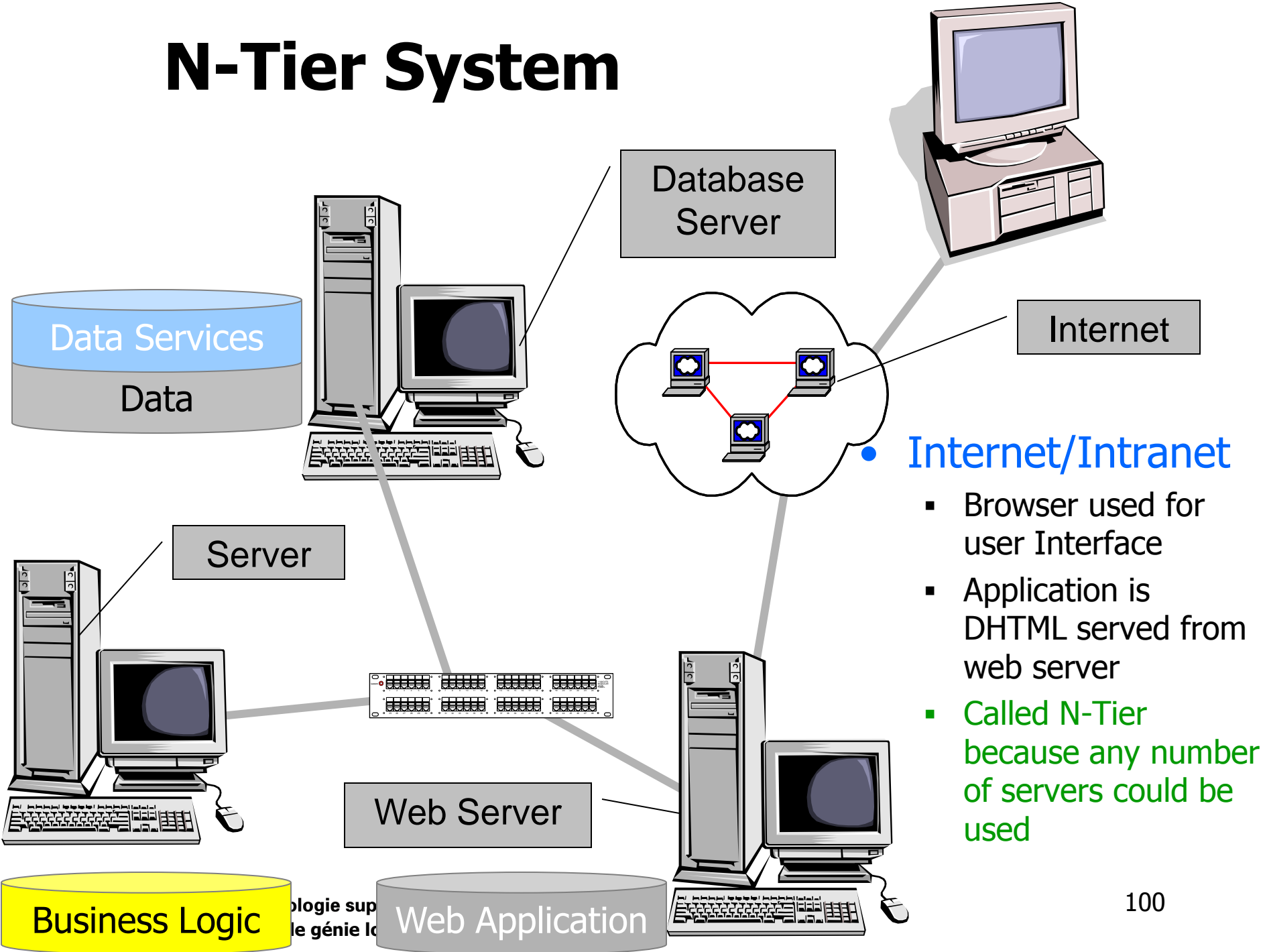
3-Tier System



- LAN/WAN

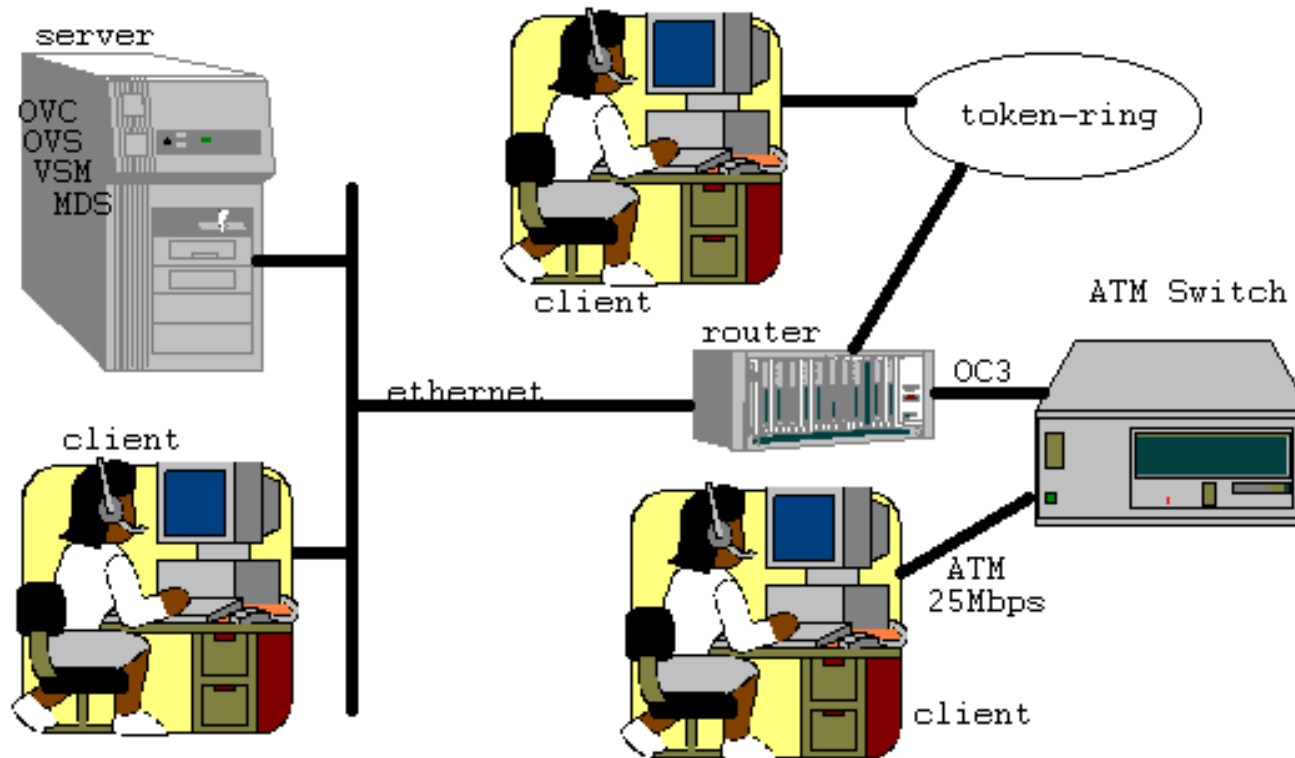
- Uses Client-Server DBMS as with 2-Tier Client-Server
- Data passed as with 2-Tier Client-Server
- Additional server does load balancing and makes easier updates to business logic programming

N-Tier System



Oracle Vidéo Streaming

de: rboyd@vccs.edu



MPEG-7 Vidéo sur demande

