



633-2.2

Dialogue avec la base de données

1. Architecture client serveur

Cedric.Baudet@he-arc.ch



- Filière
 - Bachelor of Science en informatique de gestion
- Domaine
 - Technologies informatiques
- Module
 - 633 – Architectures distribuées
- Unité d'enseignement
 - 633-2.1 – Applications réseaux et services applicatifs
 - [633-2.2 – Dialogue avec la base de données](#)



- Objectifs du module
 - comprendre et mettre en œuvre les services applicatifs fondamentaux;
 - développer, déployer et configurer des composants métiers sur un serveur d'applications.
- Objectifs de l'unité d'enseignement « Dialogue avec la base de données »
 - comprendre les différences entre l'architecture "serveur de fichiers" et "client/serveur";
 - maîtriser le principe du dialogue à sessions;
 - comprendre la notion de concurrence;
 - savoir manipuler des données à l'aide d'un langage hôte.



1. Architecture client/serveur



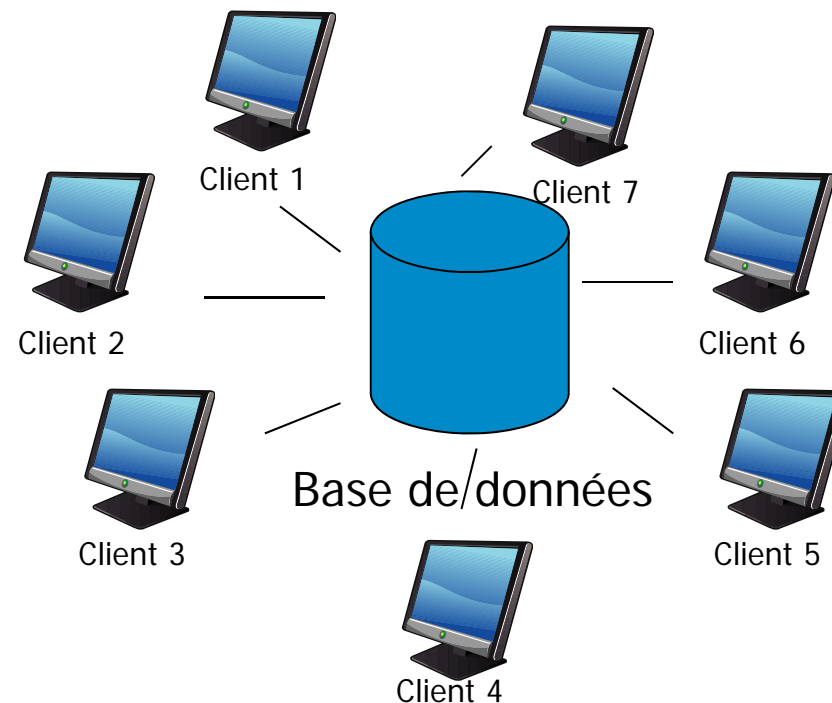
2. Transactions et verrous



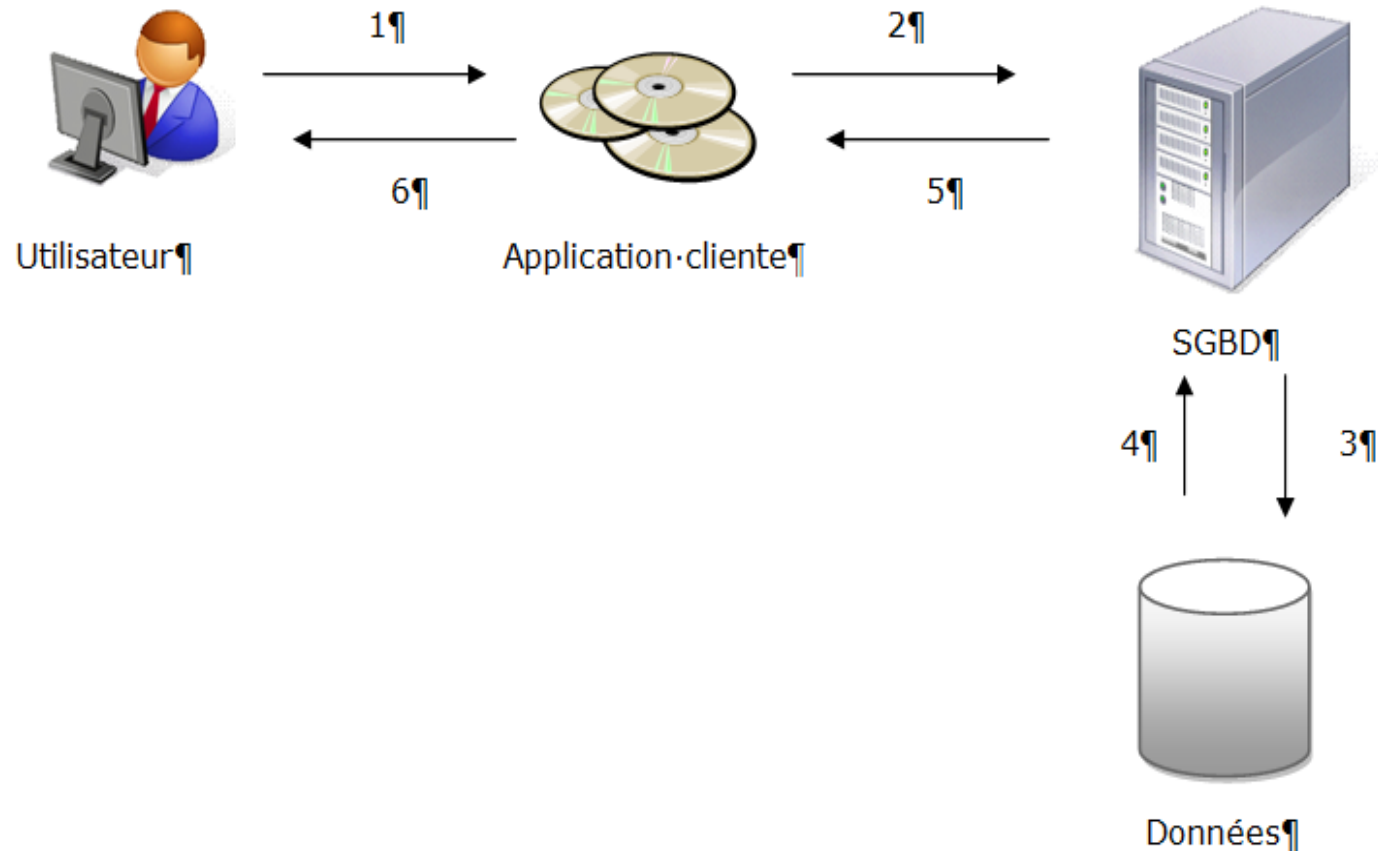
3. Manipulation des données avec un langage hôte



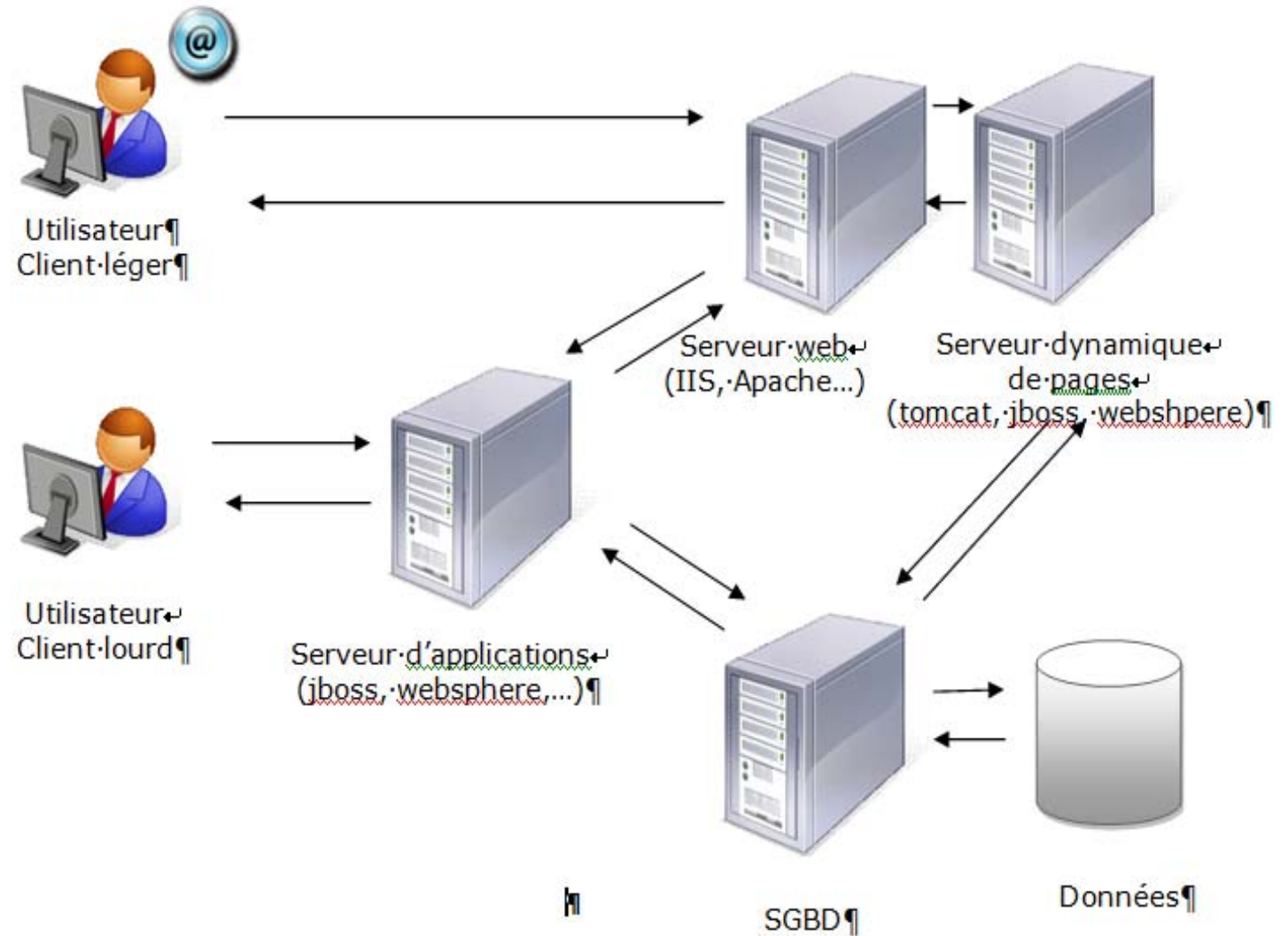
- Par définition, les données sont uniques et centralisées.
- En général les applications clientes sont distribuées sur plusieurs machines.
- Le transfert des données se fait par l'intermédiaire d'un réseau.



Principe d'une requête sur une base de données



Principe d'une requête sur une base de données



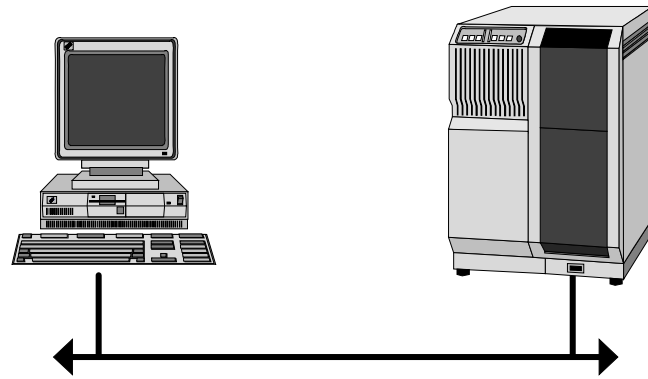


- On parle de client léger selon Pierre-Yves Cloux:
 - si aucune donnée stockée sur le client;
 - si tout (applicatifs + données) est téléchargé depuis le réseau;
 - si le client est standardisé et n'aura donc pas besoin d'être modifié avant un délai raisonnable.

- Sinon c'est un client lourd!

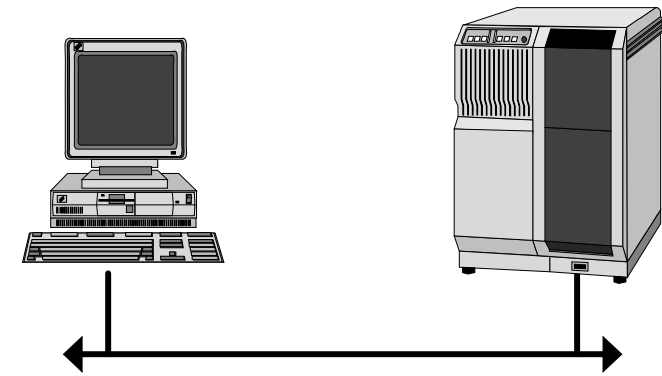


- Le SGBD est accueilli par la machine contenant le programme
 - Cette architecture est couramment nommée **serveur de fichiers**.
- Le SGBD est accueilli par la machine contenant les données.
 - Cette architecture est couramment nommée **serveur de données** ou **client serveur**.
 - Il s'agit de l'architecture la plus utilisée en entreprise.



Serveur de fichiers

Client	Serveur
Application	
SGBDR	
	DATA

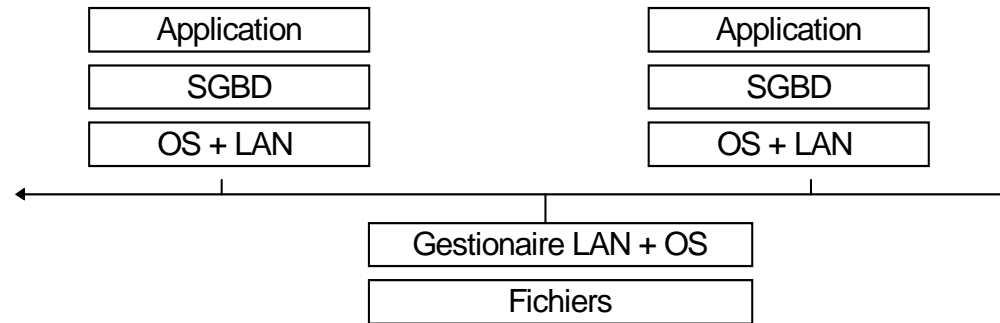


Serveur de données

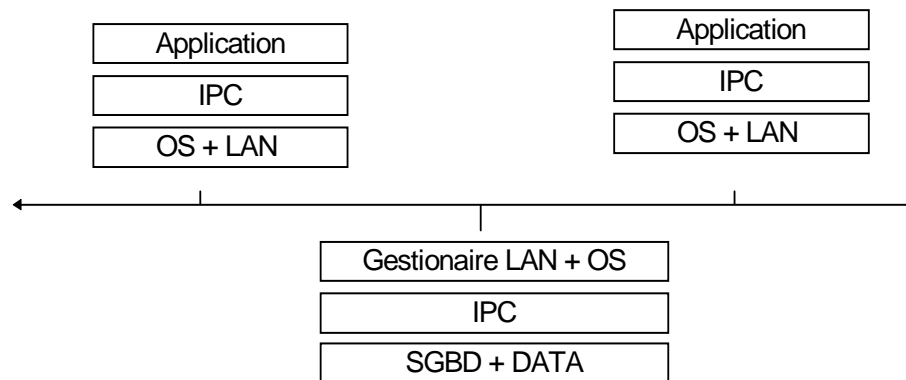
Client	Serveur
Application	
	SGBDR
	DATA



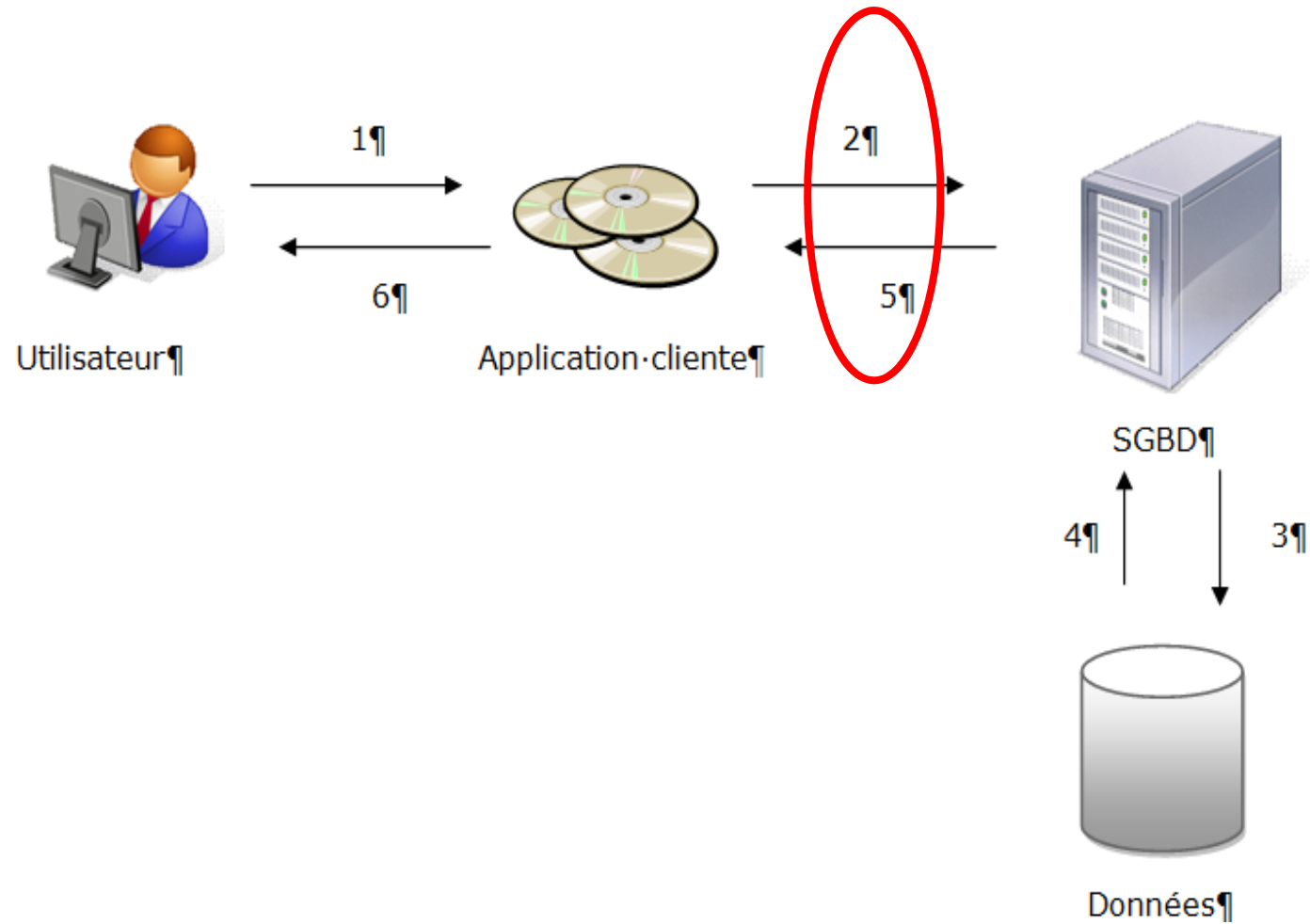
- Pour le modèle **Serveur de fichiers**



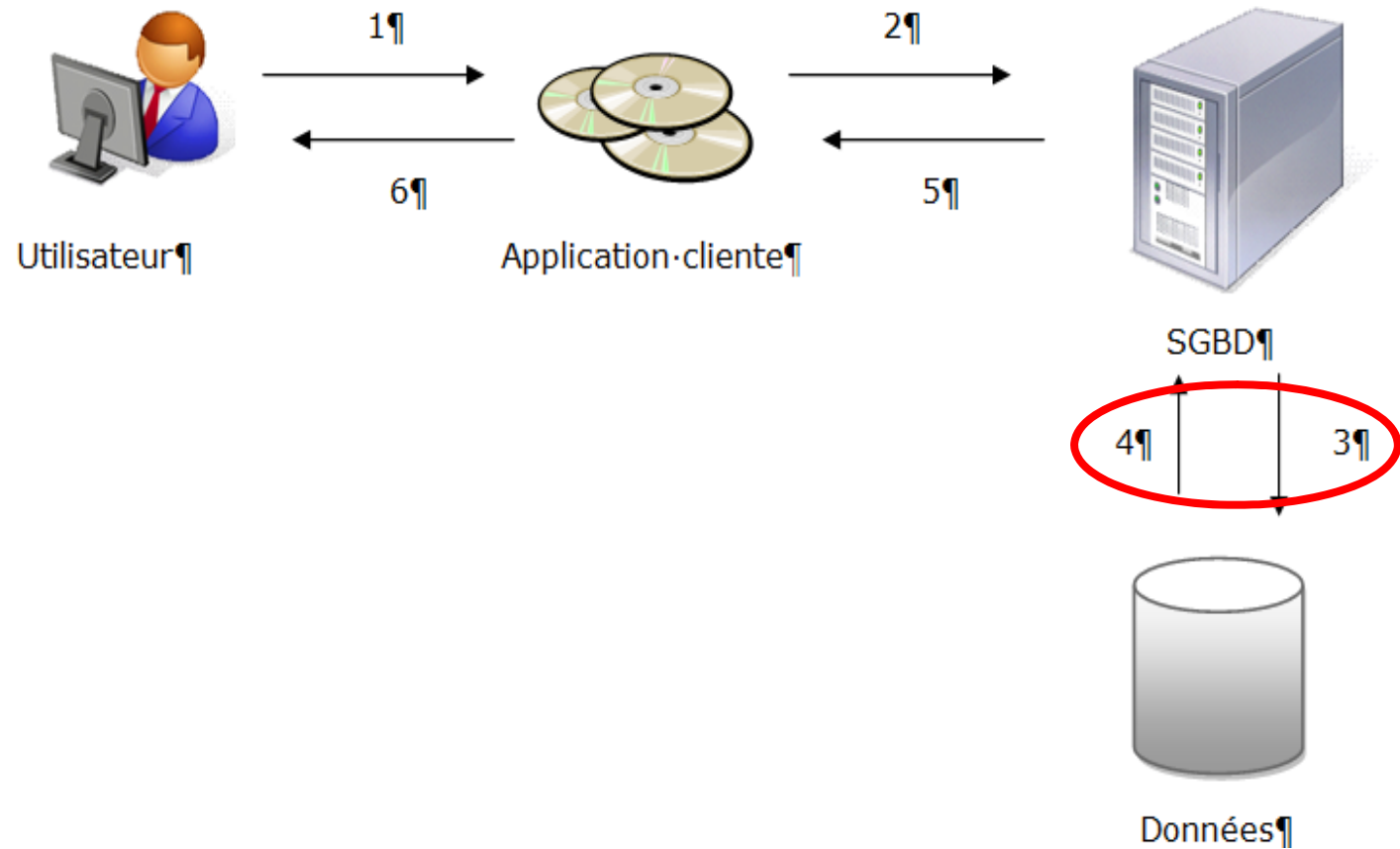
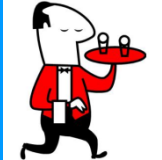
- Pour le modèle **Serveur de données (client serveur)**



Emplacement du réseau en client serveur (serveur de données)



Emplacement du réseau avec un serveur de fichiers





- Ce type d'architecture est apparu au milieu des années 80 lorsque les entreprises ont adhéré aux PC.
 - loi de Grosh → A puiſſe égale, n petites machines coûtent moins cher qu'une grosse;
 - augmentation de la productivité des développeurs à l'aide de L4G;
 - augmentation des coûts (parc machine, maintenance...)
- Ce modèle architectural fonctionne comme la relation client serveur dans la restauration...



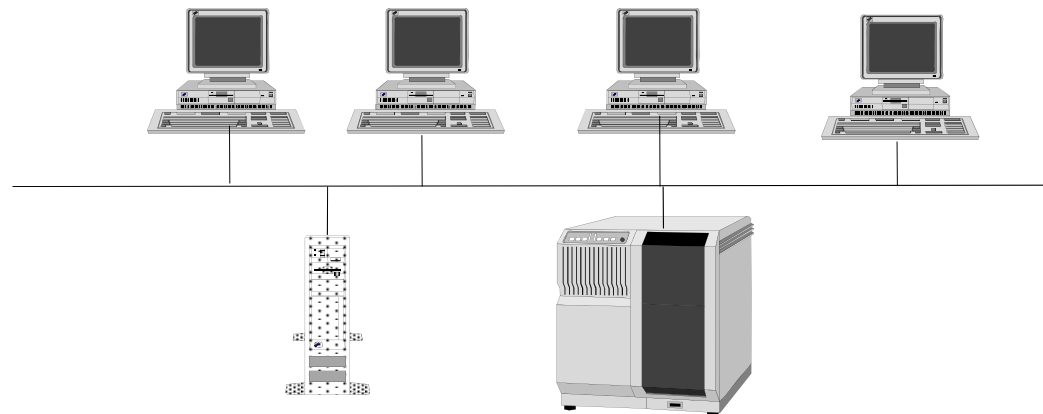
- Démonstration...
 - 2 élèves
 - le prof...
 - Les autres notent les dialogues



- Est conforme au modèle client - serveur, une application qui fait appel à des services **distants** au travers d'un échange de **messages** (les requêtes) plutôt que par un échange de **fichiers**.
- «L'idée» du modèle est de transporter sur le réseau que ce qui est nécessaire en faisant **dialoguer** les machines.

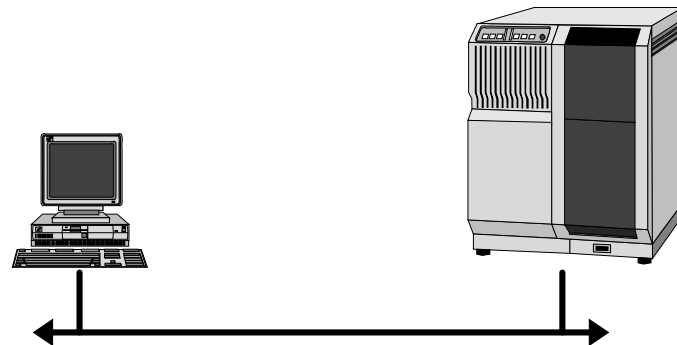


- L'architecture client serveur, conduit à bâtir des architectures « spécialisées » où les membres d'un même réseau sont soit clients, soit serveurs. (au moins à un instant t)





- L'architecture Client serveur repose sur trois composants :
 - client;
 - serveur;
 - réseaux.
- Atrophier l'un de ses composants, pénalise tout le système.



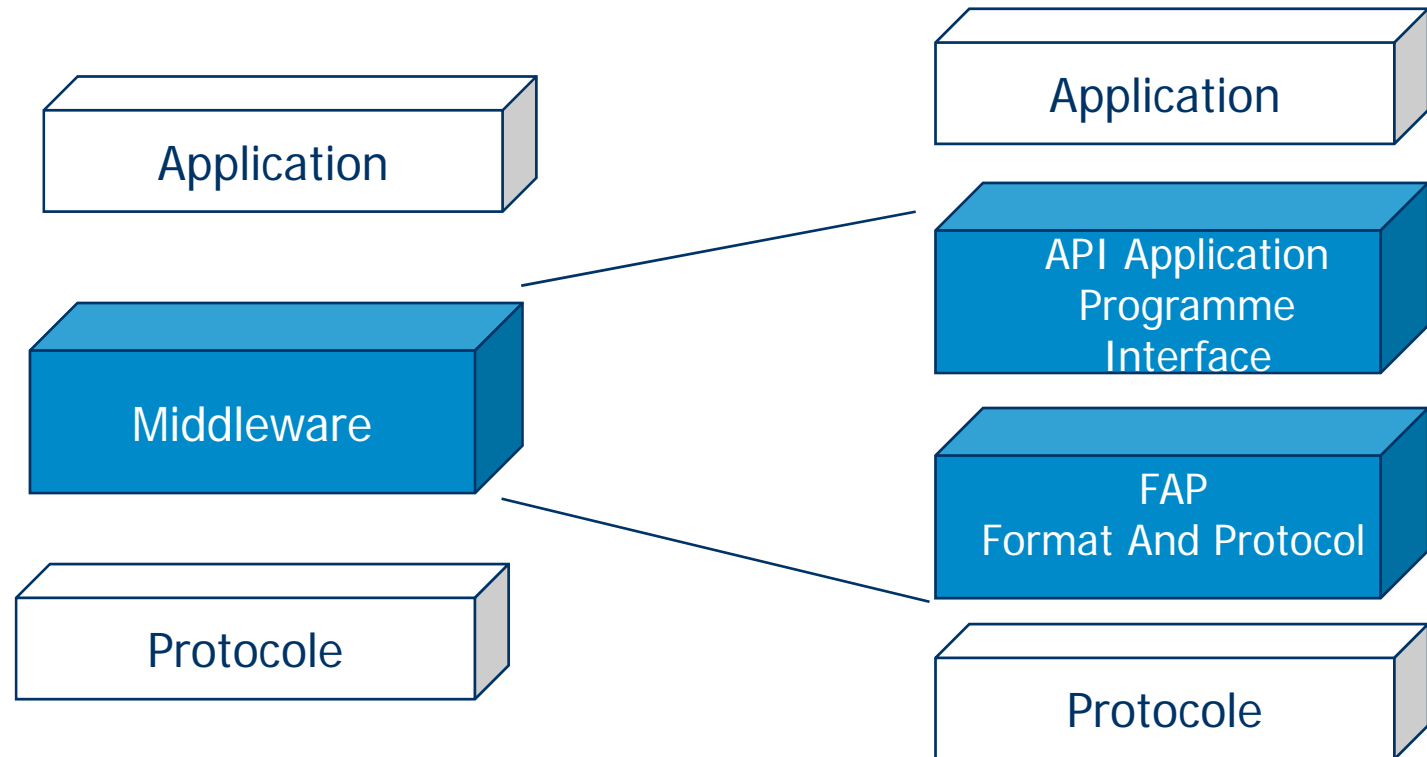


- Sur chaque machine, on retrouve les 3 composants logiciels:
 - application;
 - Middleware (médiateur en français);
 - protocole.





- Selon Pierre-Yves Cloux (2002, p. 37), « *Pour lier efficacement les différentes composantes d'une application client-serveur, il est parfois nécessaire d'avoir recours à des logiciels spécifiques: les middlewares.* »
- Il existe différents types de middlewares:
 - middleware d'accès aux données;
 - ORB - Object Request Broker;
 - moniteurs transactionnels;
 - MOM – Message Oriented Middleware
- Dans le cadre de ce cours, nous nous intéressons aux middlewares d'accès aux données.
- Un médiateur d'accès aux données permet de faire communiquer les applications avec les bases de données (ou différentes sources de données)



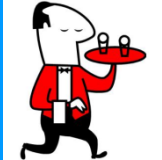
- La couche API et la couche FAP forment le middleware.
- Chaque couche "connaît" ses deux voisines.



- Paramétrage général
 - sqlnet.ora
 - domaine, méthode de nommage, cryptage, traçage.
- Paramétrage du client
 - tnsnames.ora
 - liste des serveurs accessibles, instances accessibles.
- Paramétrage du serveur
 - listener.ora
 - Liste des bases de données du serveur, type de dispatcher.
- Selon vous, où se situent ces fichiers?!?
 - **5 minutes pour trouver... puis restitution en classe.**



- Il n'y a pas une bonne et une mauvaise architecture.
- Certaines applications peuvent nécessiter une architecture client serveur, d'autres n'en ont pas forcément besoin.
- Les concepteurs/développeurs doivent connaître les différences de ces deux modèles et adapter les programmes en conséquence !!!



- Cette architecture est principalement dédiée aux applications monoposte mais sont en règle général capable de travailler en mode multi-utilisateurs.
- Une caractéristique intéressante est une maintenance très simple du SGBD.
- Cette architecture est très efficace avec un petit volume de données et une forte bande passante.
- Exemple MS-Access et SQLite.



- Cette architecture est principalement conçue pour travailler en mode multi utilisateurs.
- Elle est particulièrement efficace sur des bases de données de grande taille ou l'on va extraire des petits jeux de données.
- Le transfert de données se fait par échange de messages basés sur SQL.
- Exemple Oracle, SQL Server et PostgreSQL.



- Dans une application informatique, il existe deux versions des objets métiers:
 - une représentation dans la mémoire de l'application;
 - une représentation, persistante dans la base de données.
- Le programme client doit donc charger une copie de l'objet métier dans la mémoire à partir de l'objet persistant de la base de données.
 - Dans une version multi-utilisateurs, celle-ci doit alors verrouiller l'objet persistant afin d'éviter qu'il soit utilisé par un autre client.

Accès aux données par une méthode navigationnelle



- On part du principe que les objets persistants sont stockés dans des conteneurs ordonnés dont on connaît l'emplacement.
 - Pour charger en mémoire une collection d'objets, on écrit un algorithme basé sur une itération contrôlée de chargement d'objets uniques.
 - Cette méthode est héritée de l'accès aux données par fichiers (ISAM, BFILE,...).
 - Elle n'utilise que très peu les capacités du SGBD et a pour principale conséquence l'exécution d'autant de requêtes SQL qu'il y a d'objets à charger.

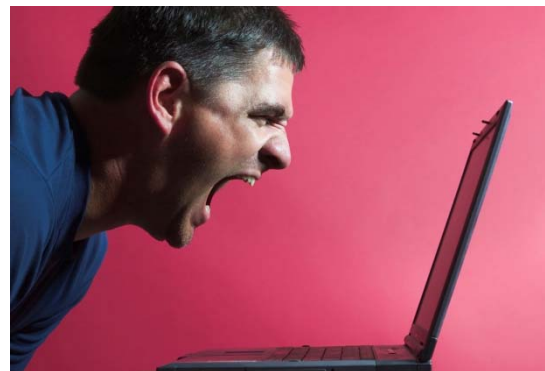
Accès aux données par une méthode ensembliste



- On part cette fois du principe qu'on va charger une collection minimale d'objets qui ont des propriétés communes et identifiées.
- Une fois les propriétés exprimées en langage SQL, elle sera soumise au SGBD pour alimenter la collection d'objets.
- Cette méthode utilise toutes les capacités du SGBD.



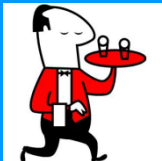
- La méthode navigationnelle n'est pas adaptée avec l'utilisation d'un serveur SQL puisqu'elle n'utilise pas les capacités du SGBD mais elle peut rester efficace si on utilise une architecture serveur de fichiers.
- A l'inverse si on utilise cette même méthode avec une architecture client serveur, les temps de réponses seront catastrophiques, malgré l'utilisation d'un serveur très puissant.
- Ce n'est pas un défaut du modèle client serveur! C'est le programmeur qui a un défaut !!!



Les règles d'or !!



- Ecrire correctement la requête pour déléguer le maximum de tâches au SGBD.
- Laisser dans la base de données tout ce que vous n'avez pas besoin.
- Verrouiller si c'est nécessaire et déverrouiller dès que possible.
- Penser à ce qu'il va se passer sur le réseau ?



Query Form

Form with three text input fields labeled 'texte' and two buttons below them.

List Form

Table with 3 columns labeled 'texte' and 5 rows of input fields.

View Form

Form with five text input fields labeled 'texte' stacked vertically.

```
SELECT att1,  
att2,att3  
FROM table1  
WHERE condition;
```

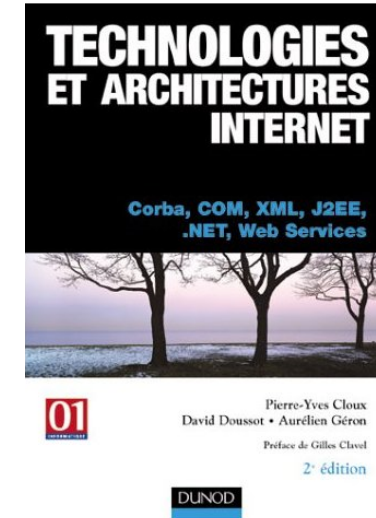
```
SELECT  
att1,att2,att3,att4,att5,att6  
FROM table1  
WHERE identifiant = ...;
```







- CLOUX, Pierre-Yves et al.
*Technologies et architectures
Internet*. Paris : Dunod, 2002







- Il faut déléguer au maximum au SGBD !
- En client/serveur, on ne charge que ce que l'on a besoin.
- Toujours imaginer ce qui se passe sur le réseau.
- Masteriser les transactions entre BD et logiciel qui les utilisent.



- Série 1 – Sur Cyberlearn
 - Besoin de votre compte de BD SOUK.

