

# **Les architectures I\*Net**

## **Première approche**

## Les architectures I\*Net

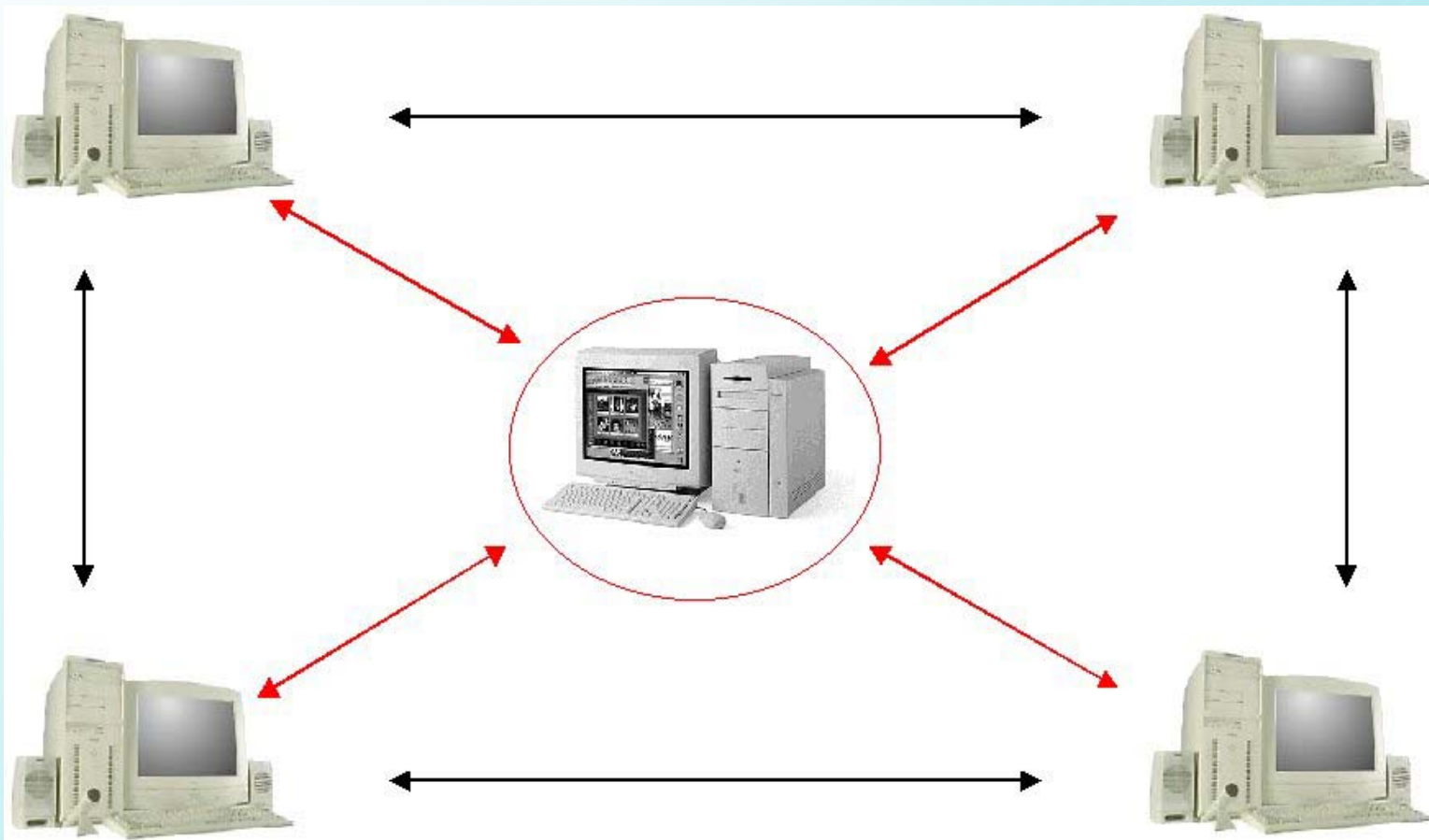
- **L'histoire d'Internet**
- **De l'architecture Mainframe à l'architecture I\*net**
- **J2EE et les architectures I\*net**

# L' histoire d'Internet

- **Définition des réseaux**
- **Définition d'Internet**
- **Historique de l'Internet**
- **Définition d'un Intranet**
- **Définition d'un Extranet**

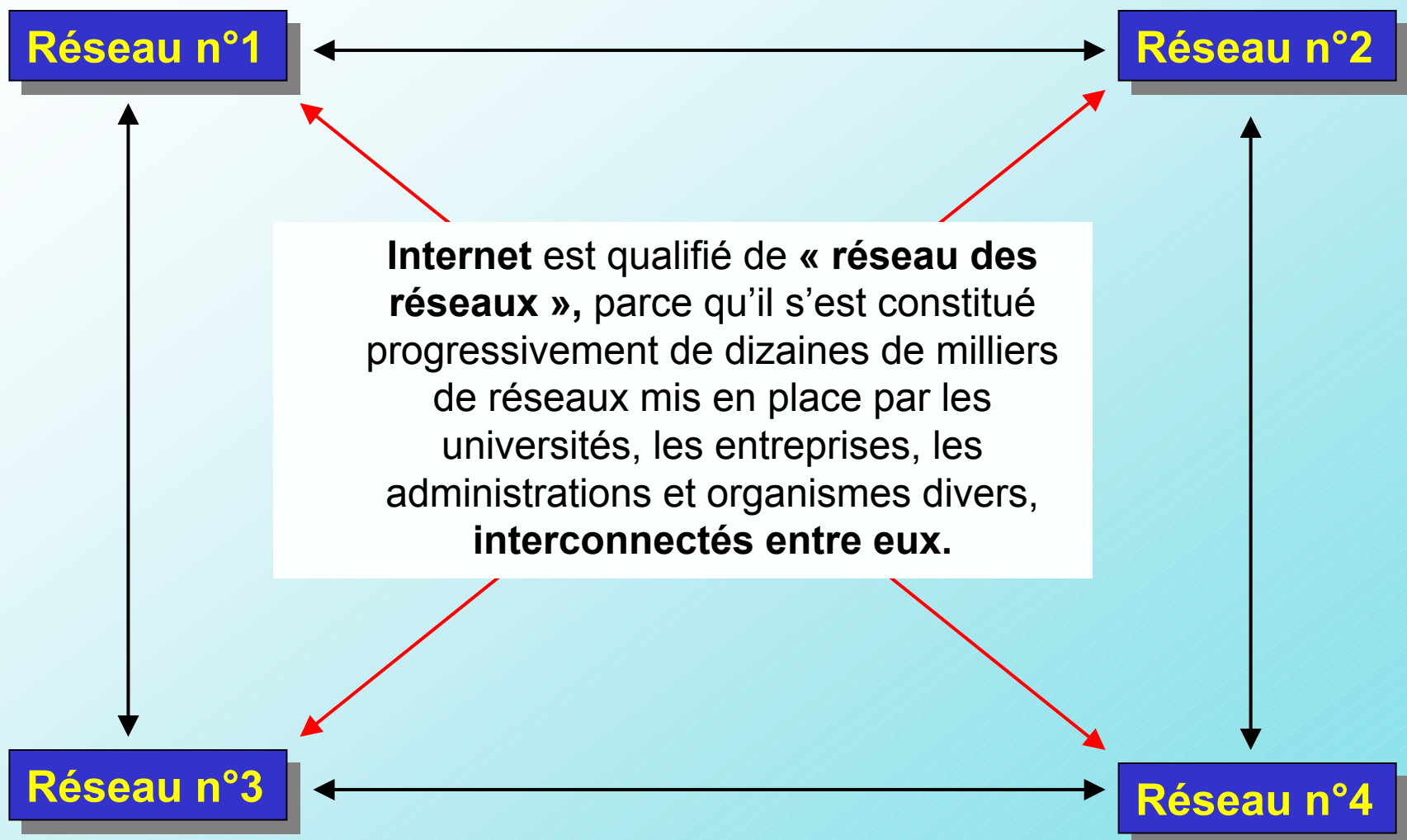
# I. Qu'est ce qu'un réseau ?

- Un réseau est constitué d'ordinateurs reliés entre eux à l'aide des moyens de communications actuels.



- Définition des réseaux
- **Définition d'Internet**
- Historique de l'Internet
- Définition d'un Intranet
- Définition d'un Extranet

# II. Qu'est ce que l'Internet ?



**Internet** est qualifié de « **réseau des réseaux** », parce qu'il s'est constitué progressivement de dizaines de milliers de réseaux mis en place par les universités, les entreprises, les administrations et organismes divers, **interconnectés entre eux.**

## II. Qu'est ce que l'Internet ?

- Internet n'appartient à personne.
- Des organismes spécialisés définissent les normes techniques.
- Le financement du réseau est assuré par les différents acteurs (internauts, fournisseurs d'accès, organismes divers, entreprises, etc.).



## II. Qu'est ce que l'Internet ?

- Encore appelé le NET, INTERNET permet :
  - la transmission du texte, du son, des images fixes et des images animées (le Word Wide Web).
  - l'envoi et la réception de messages (courrier électronique),
  - l'échange de fichiers informatiques,
  - la participation interactive à des forums de discussion encore appelés groupes de nouvelles ou encore news,
  - la consultation d'informations commerciales ou non,
  - l'achat ou la vente : c'est l'économie en ligne,
  - la téléconférence.

- Définition des réseaux
- Définition d'Internet
- Historique de l'Internet
- Définition d'un Intranet
- Définition d'un Extranet

### III. Historique de l'Internet

- **1969** : Durant la guerre froide, le DoD (Département of Défense) demande à l'ARPA (Advanced Research Project Agency) de créer un réseau d'interconnexion (réseau ARPANET) capable de :
  - transférer des fichiers sur la totalité du territoire américain ;
  - résister à une attaque nucléaire ;
  - fonctionner même en cas de destruction partielle.

### III. Historique de l'Internet

- **1979** : Le réseau devient utilisable par les chercheurs civils. Il est administré et financé par la NSF (National Science Foundation) équivalent au CNRS français .

### III. Historique de l'Internet

- **1983** : L'agence de communication de la défense décide d'ouvrir le réseau grandissant au domaine civil et le divise en deux sous-réseaux :
  - **Milnet** : réseau militaire sous la tutelle d'une autorité militaire ;
  - **Arpanet** : réseau destiné aux chercheurs et scientifiques placé sous une autorité universitaire.

### III. Historique de l'Internet

- **1986** : La fondation scientifique nationale américaine (National Science Foundation) crée le NSFNet.
- **1990** : Arpanet cesse d'exister. Le nom d'Internet (pour Inter Networking) s'impose peu à peu.
- **1992** : La mise en place du "Web" (avec le multimédia) a optimisé son développement.

### III. Historique de l'Internet

- **1995** : NSFNet disparaît à son tour. Composé de réseaux interconnectés, Internet devient le réseau mondial que nous connaissons aujourd'hui.

## L'histoire d'Internet

- **Définition des réseaux**
- **Définition d'Internet**
- **Historique de l'Internet**
- **Définition d'un Intranet**
- **Définition d'un Extranet**



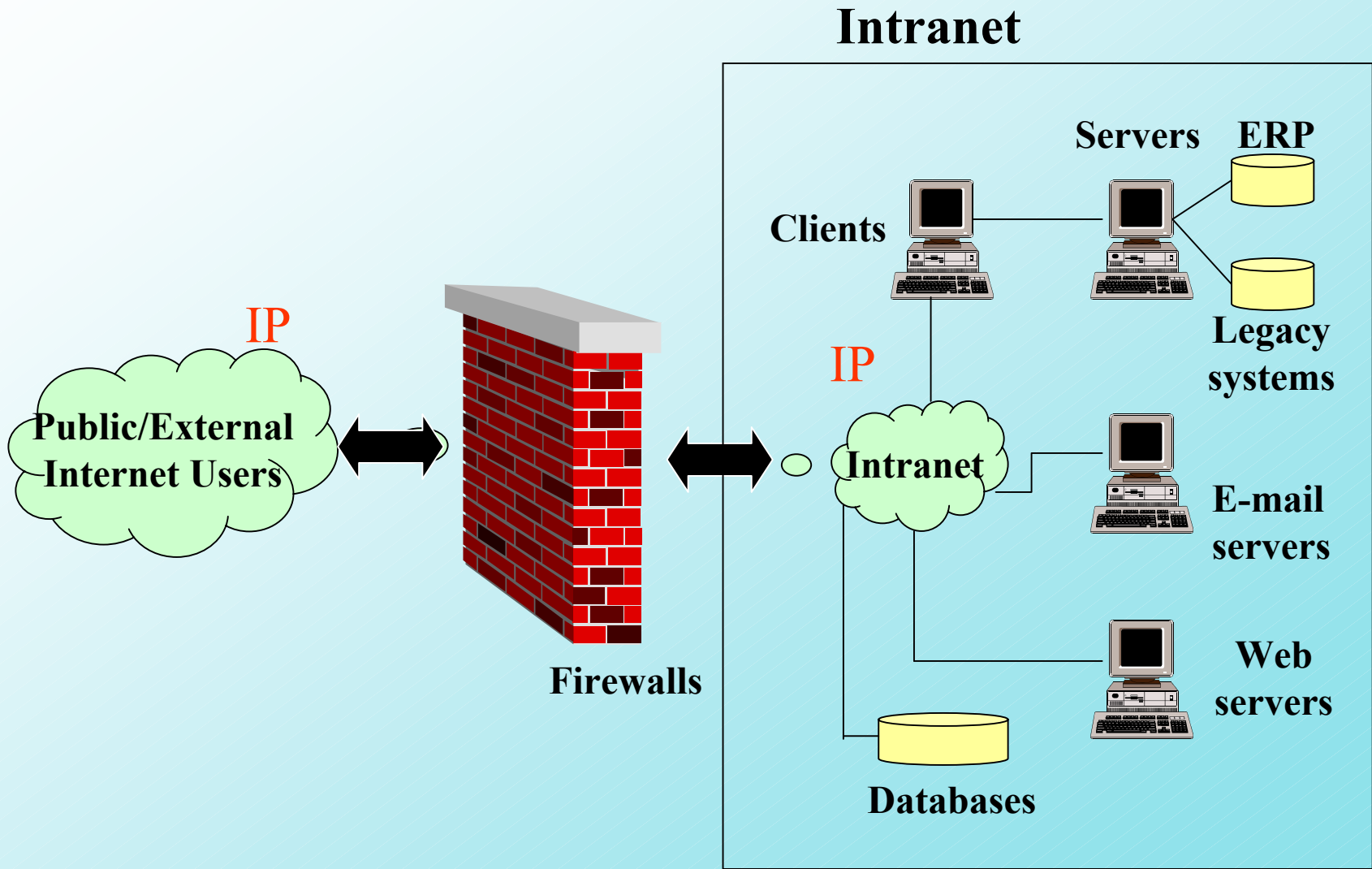
## IV. Qu'est ce qu'un Intranet ?

- **Définition :**
  - Fonctionnant comme Internet, c'est un réseau interne au sein de l'entreprise, pour des locaux d'un même site ou des locaux géographiquement décentralisés.

## IV. Qu'est ce qu'un Intranet ?

- L'Intranet permet d'optimiser le système d'information en vue de faciliter :
  - la communication ;
  - le travail de groupe ;
  - les échanges (envoi et réception de fichiers, agendas, messagerie) ;
  - le partage de l'information (navigation, consultation) ;
  - la gestion des données électroniques (bases de données).

# IV. Qu'est ce qu'un Intranet ?



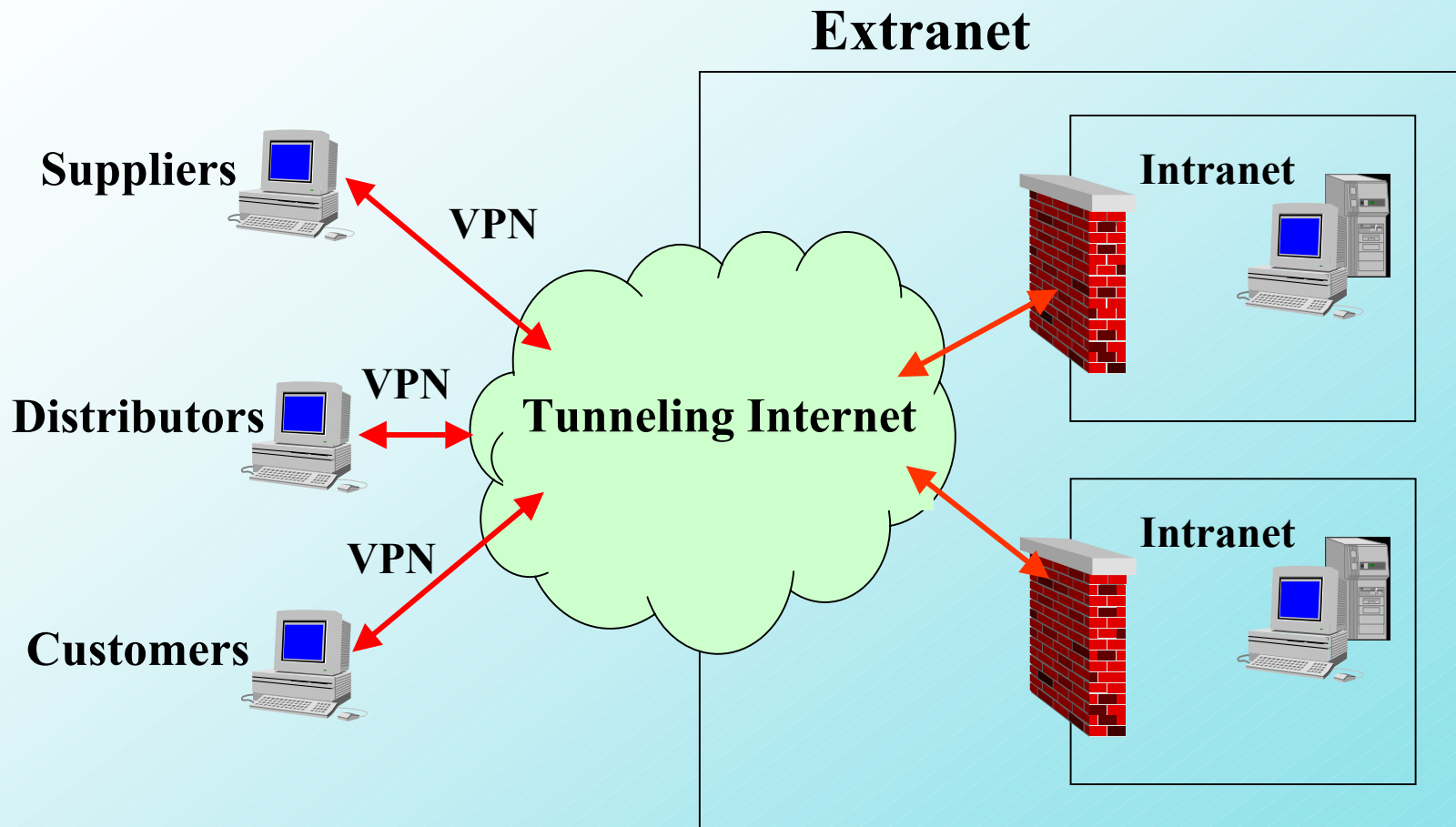
## L'histoire d'Internet

- **Définition des réseaux**
- **Définition d'Internet**
- **Historique de l'Internet**
- **Définition d'un Intranet**
- **Définition d'un Extranet**

## V. Qu'est ce qu'un Extranet ?

- L'Extranet est une extension de l'Intranet qui permet d'étendre le réseau aux fournisseurs ou encore aux clients de l'entreprise.

# V. Qu'est ce qu'un Extranet ?



# De l'architecture Mainframe à l'architecture I\*net

## Du mainframe aux architectures I\*net

- **L'architecture mainframe**
- **Le système client-serveur**
- **Les architectures distribuées**
- **L'architecture 3-tiers**
- **L'architecture n-tiers**



## Description de l'exemple

Pour illustrer notre propos, voici un exemple simple :

Sur un poste client, on affiche une liste de documents datés.

On utilise une zone de test pour créer un filtre sur ces documents.

L'utilisation du filtre permet d'afficher uniquement les documents filtrés.

## Description de l'exemple

L'interface peut ressembler à ceci :

Recherche / filtrer :

22/04/2003	planning organisationnel	▲
22/04/2003	référence matériel	
23/04/2003	disponibilité des ressources	
23/04/2003	les architectures N-tiers	
23/04/2003	Proposition commerciale	▼

Chaque architecture sera décrite par rapport à cet exemple.

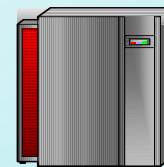
- **L'architecture mainframe**
- **Le système client-serveur**
- **Les architectures distribuées**
- **L'architecture 3-tiers**
- **L'architecture n-tiers**

## L'architecture mainframe

Le client est passif : c'est un terminal qui ne fait que visualiser les données et les saisir.

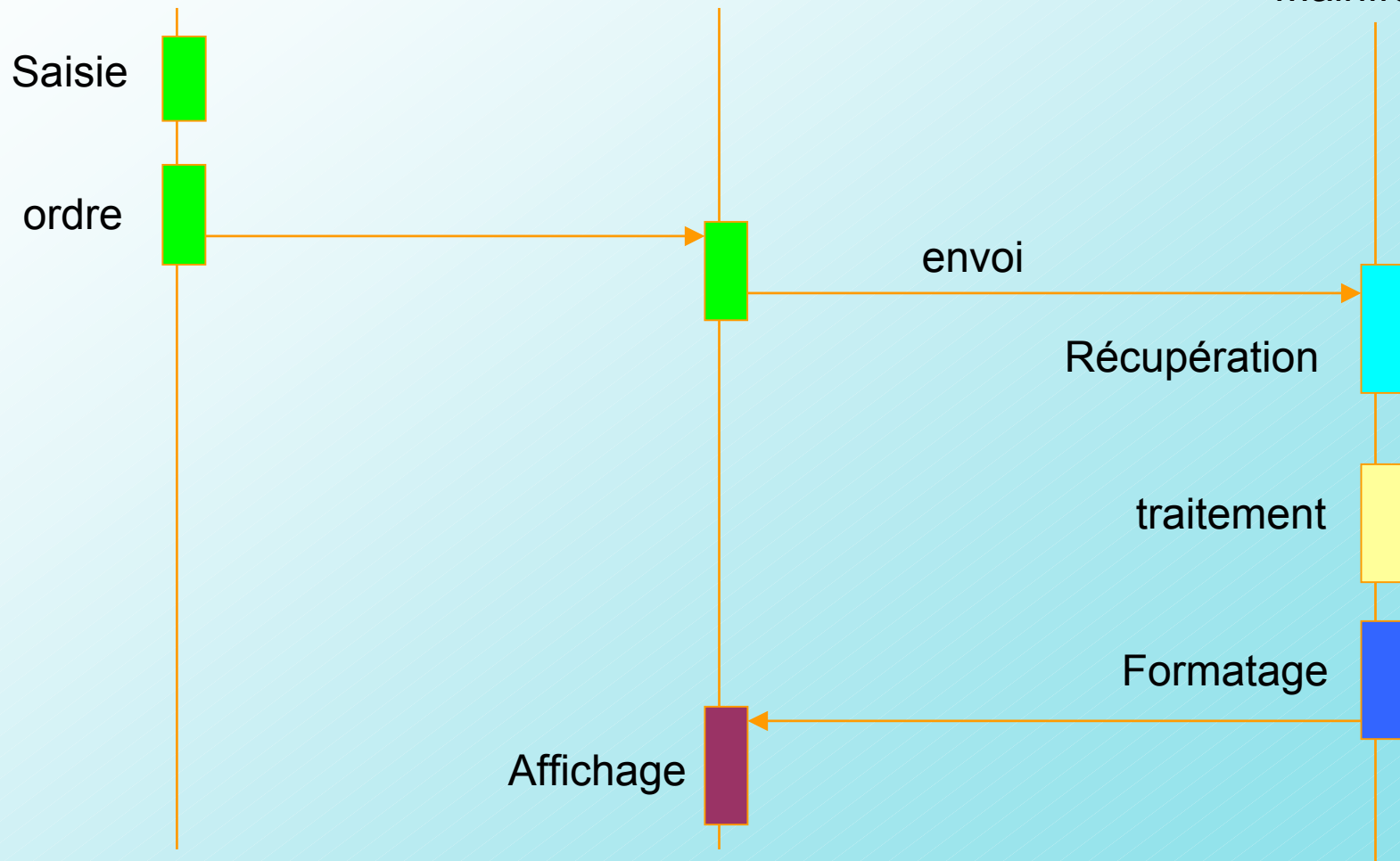
Le site central (mainframe) contient toute la logique applicative ainsi que les données.

# L'architecture mainframe



Terminal

Mainframe



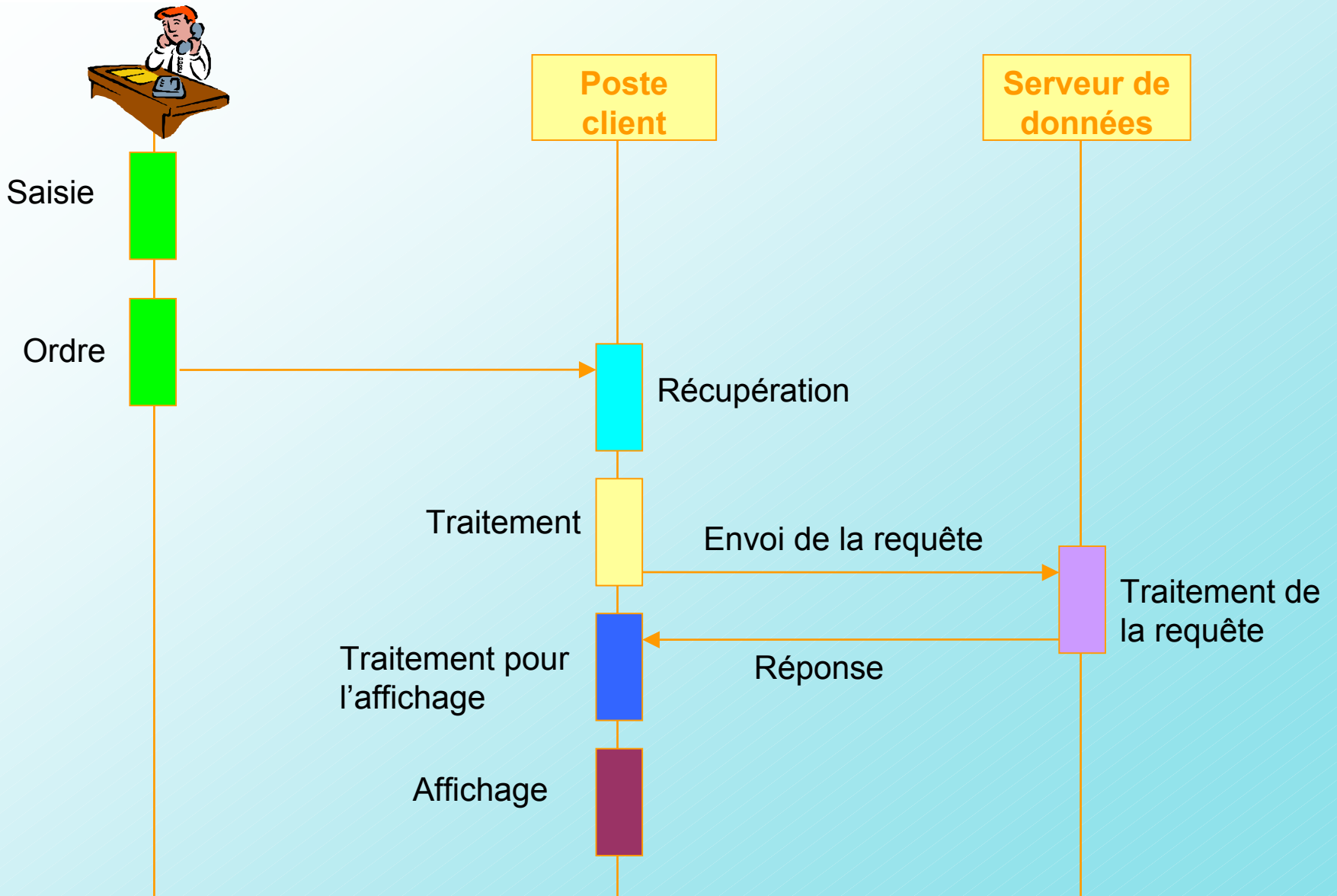
- **L'architecture mainframe**
- **Le système client-serveur**
- **Les architectures distribuées**
- **L'architecture 3-tiers**
- **L'architecture n-tiers**

## L'architecture client-serveur

L'architecture client-serveur de données est apparue avec l'arrivée des PC. Ce dernier étant capable d'exécuter des traitements, le serveur ne prend plus en charge que la gestion des données.

La logique applicative se concentre sur le client qui communique avec le serveur via des requêtes SQL. On parle de client *lourd*, car c'est lui qui gère tous les traitements informatiques.

# L'architecture client-serveur





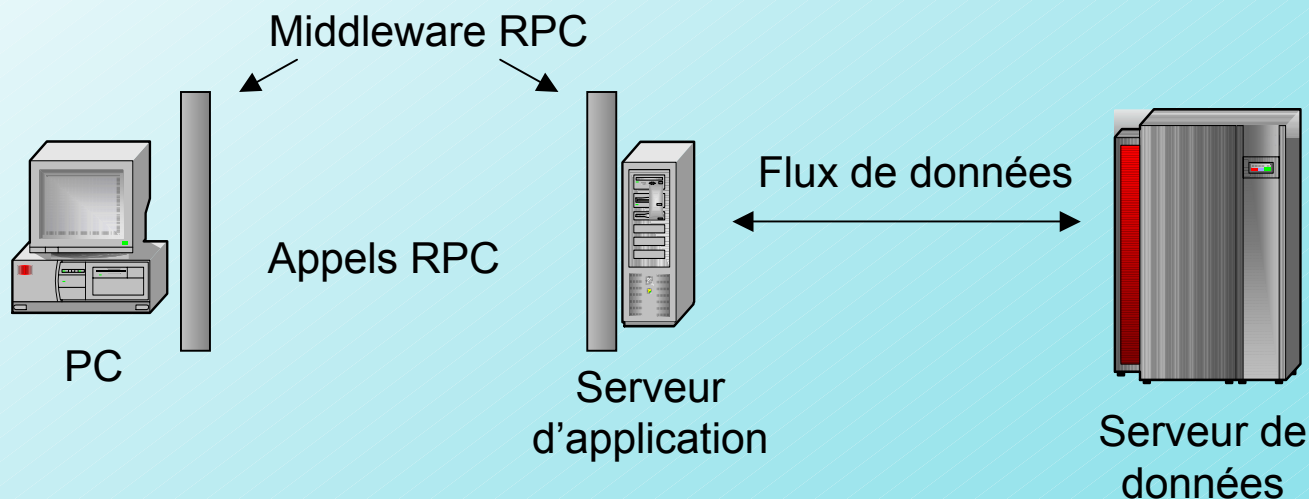
- **L'architecture mainframe**
- **Le système client-serveur**
- **Les architectures distribuées**
- **L'architecture 3-tiers**
- **L'architecture n-tiers**

## Les architectures distribuées

Intérêt des architectures distribuées :

- Alléger la charge du poste client.
- Simplifier les mises à jour du poste client.

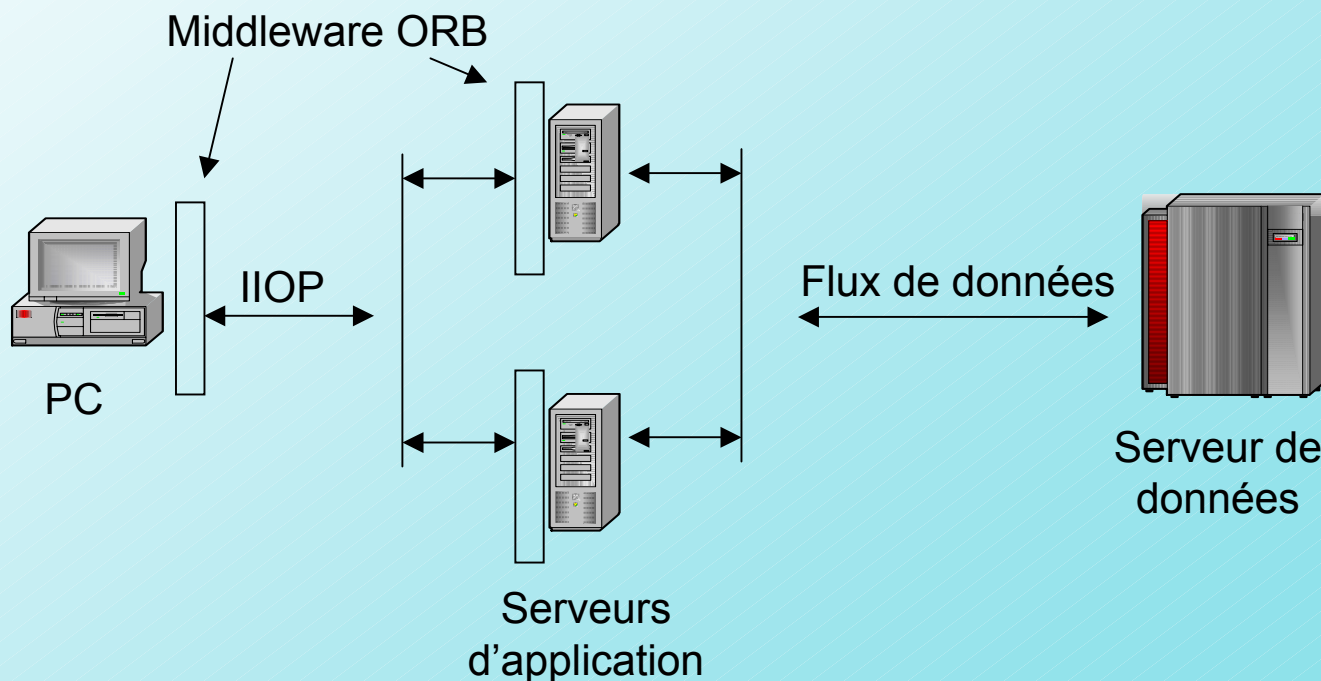
Pour cela, on utilise un serveur dit *serveur d'application* qui va gérer la logique applicative.



## Les architectures distribuées

Une évolution de ce type d'architecture consiste à utiliser plusieurs serveurs d'application.

Pour cela, on utilise des systèmes à objets distribués : l'application doit être spécialement conçue pour pouvoir être répartie sur plusieurs serveurs d'application.



- **L'architecture mainframe**
- **Le système client-serveur**
- **Les architectures distribuées**
- **L'architecture 3-tiers**
- **L'architecture n-tiers**

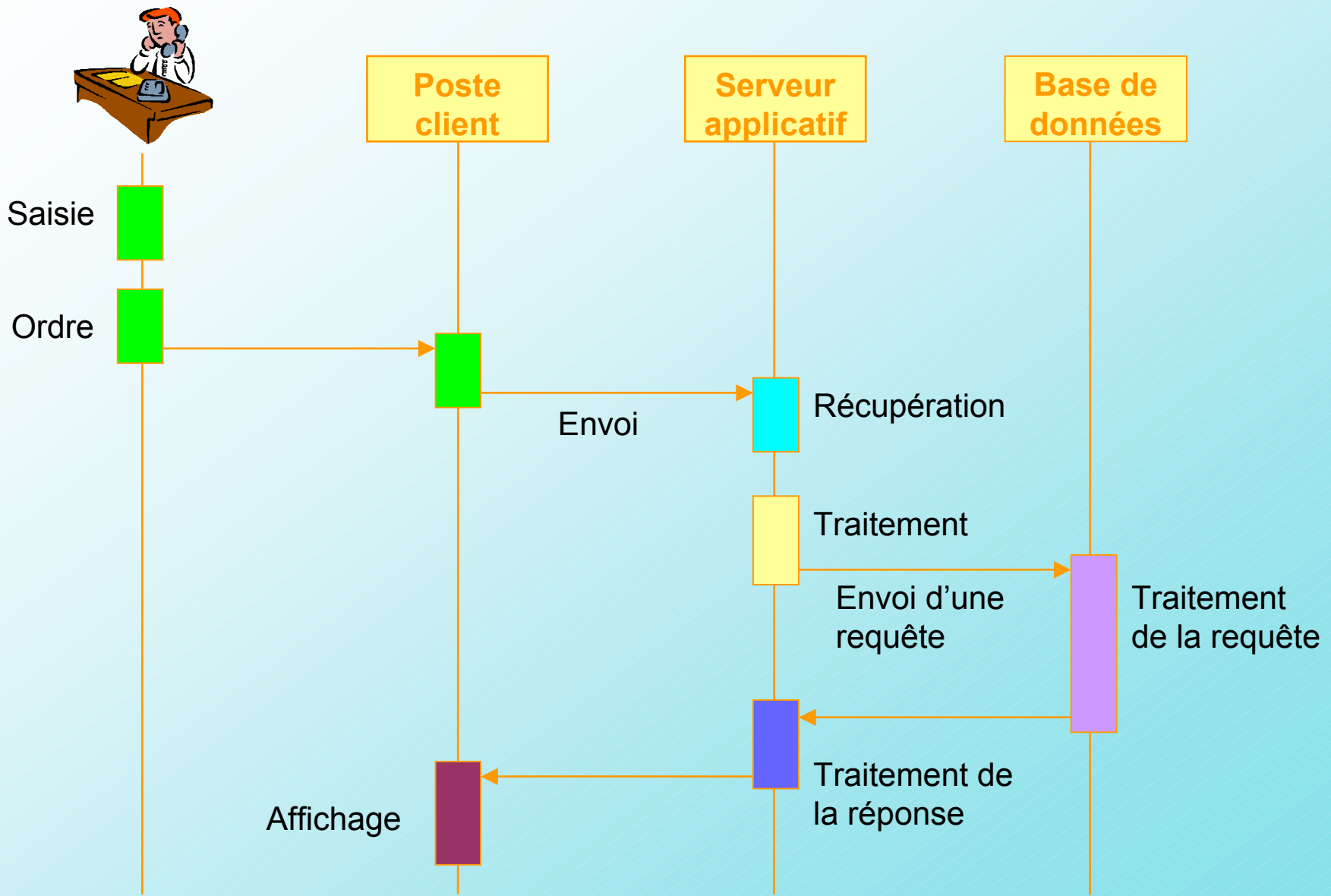
## L'architecture 3-tiers

L'architecture 3-tiers est une architecture distribuée. On parle de 3 tiers applicatifs car les éléments sont séparés :

- La présentation se fait sur un client dit *léger*,
- Le moteur applicatif tourne sur un serveur applicatif,
- La persistance des données est gérée par un serveur de bases de données.

Bien souvent, le poste client est un navigateur Internet. Ce sont en effet les technologies liées à Internet qui ont permis l'élaboration de ces architectures.

# L'architecture 3-tiers



## Du mainframe aux architectures I\*net

- **L'architecture mainframe**
- **Le système client-serveur**
- **Les architectures distribuées**
- **L'architecture 3-tiers**
- **L'architecture n-tiers**

## L'architecture n-tiers

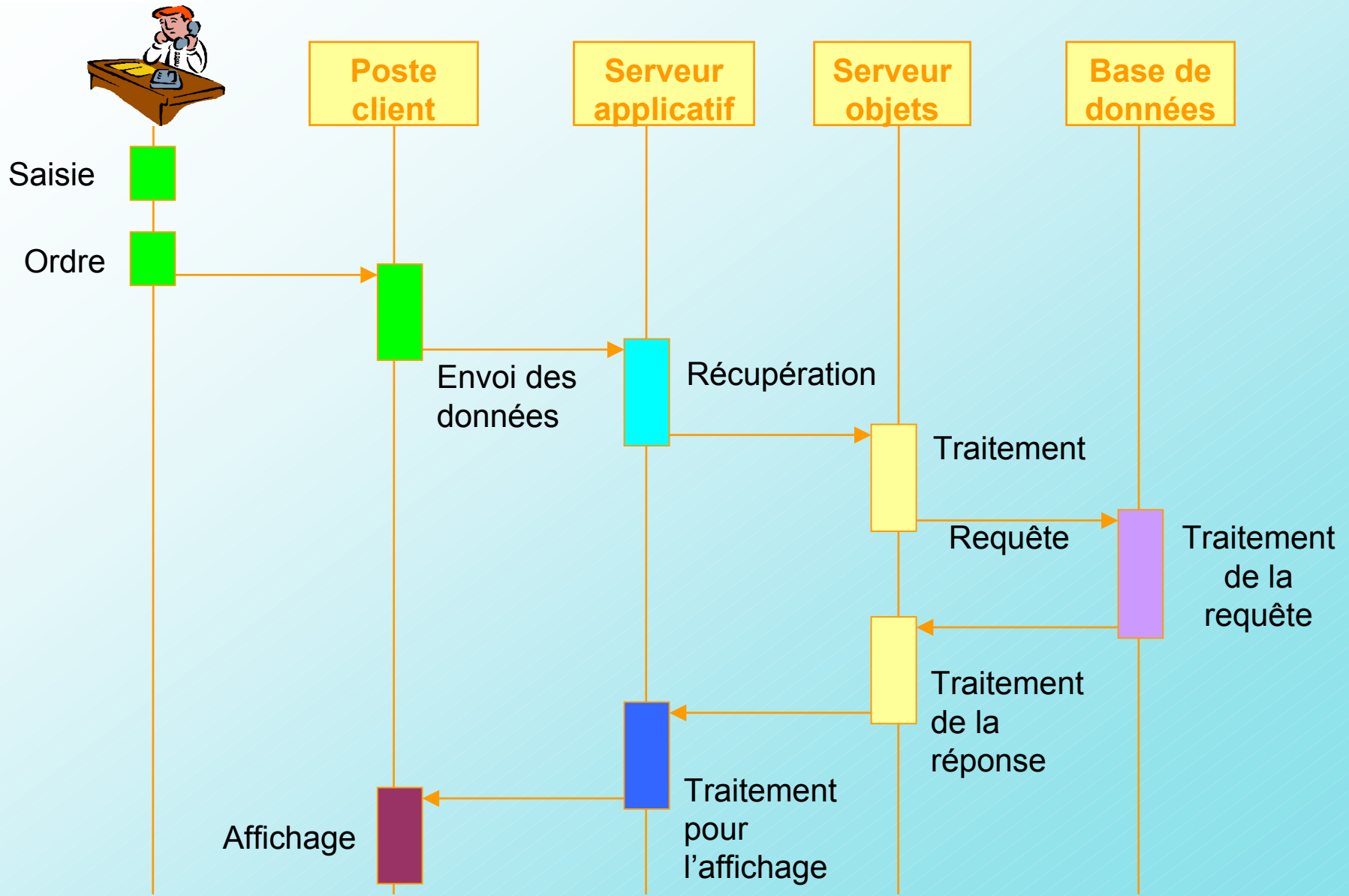
On qualifie d'architecture *n-tiers*, ou *multi-tiers*, les architectures à objets distribués.

Le serveur applicatif va permettre d'utiliser les composants distribués sur des *serveurs objets* et d'alléger la charge du serveur applicatif.

Les serveurs objets peuvent être nombreux, et vont se partager les traitements applicatifs. Ils sont liés aux différentes bases de données.

Le serveur applicatif fait le lien entre les postes clients et les serveurs objets.





# J2EE et les architectures I\*net

- **Le langage Java**
- **J2EE : un ensemble de solutions**
- **Les modules du serveur Internet**
- **La logique applicative**
- **J2EE et les bases de données**
- **Le modèle MVC**
- **Le modèle MVC 2**

## Le langage Java

- Un langage orienté objet créé par Sun Microsystems en 1995.
- Langage fortement orienté réseau :
  - Mécanismes pour des applications serveur,
  - Mécanismes côté client,
  - Mécanismes de connexion à des bases de données.
- Java propose de nombreux API pour les développeurs.
- J2EE ( *Java 2 Enterprise Edition* ) : la solution Java pour des applications I\*Net.

- Le langage Java
- **J2EE : un ensemble de solutions**
- Les modules du serveur Internet
- La logique applicative
- J2EE et les bases de données
- Le modèle MVC
- Le modèle MVC 2

## J2EE : un ensemble de solutions

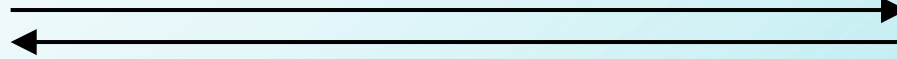
- L'architecture J2EE se compose de 3 éléments :
- Des modules pour le serveur Internet,
  - Un système permettant de gérer l'application,
  - Un système de communication avec des bases de données.

Ces éléments peuvent être utilisés indépendamment les uns des autres.

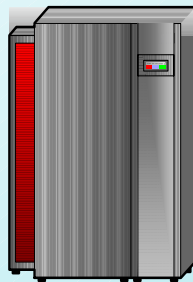
# J2EE : un ensemble de solutions



Navigateur Internet



Serveur Web



Bases de données



Serveur applicatif

## J2EE : un ensemble de solutions

Le module pour le serveur Web va permettre la communication avec le poste client ( affichage du poste et récupération des données).

Le module applicatif permet la gestion de l'application : c'est le moteur logiciel.

Le module de communication avec les bases de données se connecte et dialogue avec un gestionnaire de base de données.



## J2EE et les architectures I\*net

- **Le langage Java**
- **J2EE : un ensemble de solutions**
- **Les modules du serveur Internet**
- **La logique applicative**
- **J2EE et les bases de données**
- **Le modèle MVC**
- **Le modèle MVC 2**

## Les modules du serveur Internet

Le serveur Web a pour rôle :

- De formater l'affichage du poste client,
- De récupérer les informations envoyées par l'utilisateur à partir de son poste client.

Le serveur Web a pour avantage :

- D'utiliser un client léger,
- De permettre l'accès à l'application depuis n'importe quel poste,
- De faciliter la mise à jour des postes clients : seul le serveur doit être mis à jour.

## Les modules du serveur Internet

J2EE possède deux technologies pour la gestion des serveurs Web :

- Les servlets,
- Les JSP.

Une servlet est un programme s'exécutant sur un serveur et traitant des informations provenant ( ou à destination ) d'un client.

Une JSP est une sorte de servlet facile à mettre en place et utilisée pour créer des pages Web dynamiques.

- **Avant d'aller plus loin : les balises HTML**

Les balises HTML permettent de représenter un document qui sera édité sur Internet.

Ces balises sont interprétées par le poste client ( navigateur Web ).

```
<nomBalise param="valeur"> texte </nomBalise>
```

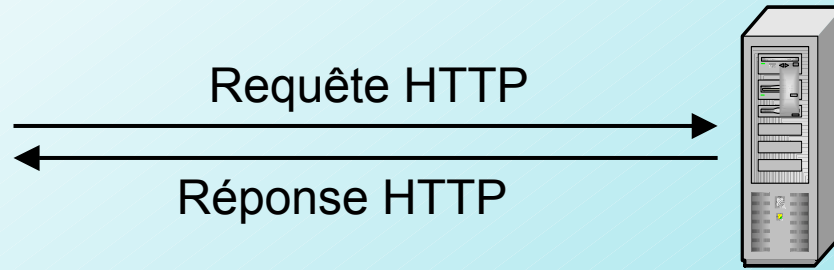
Les balises HTML sont prédéfinies, et peuvent avoir des attributs.

Exemples de balise : <html></html>, <body></body>, <i></i>, <br>, etc.

# Les modules du serveur Internet



Navigateur Internet



Serveur Web

Traitement de la requête par Servlets et / ou JSP

- **Les Servlets**

**Définition :** petit '*serveur*' ou '*services*' écrit en Langage Java qui utilisent une API spécifique.

**Avantages :** Initialisée à la création. Il n'en existe qu'une seule instance.

→ Meilleure performance lors de montée en charge par rapport aux scripts CGI.

**Inconvénients :** Le code HTML est toujours mixé avec le code de programmation.

→ Difficultés de maintenance du code.

- **Technique de Scripting : les JSP**

Avec la technique de scripting, c'est le code de programmation que l'on incorpore dans la page HTML. Le serveur compile automatiquement la page, et l'exécute ensuite.

En Java, les JSP permettent de faire du scripting.

Les langages de scripting sont particulièrement adaptés aux équipes infographistes.

Cette approche nécessite l'incorporation importante de code applicatif dans le HTML

→ Limite la réutilisation du code

## J2EE et les architectures I\*net

- **Le langage Java**
- **J2EE : un ensemble de solutions**
- **Les modules du serveur Internet**
- **La logique applicative**
- **J2EE et les bases de données**
- **Le modèle MVC**
- **Le modèle MVC 2**



## La logique applicative

La logique applicative correspond à tout le travail qui sera effectué par le programme et qui ne correspond ni à la persistance des données, ni à leur affichage.

La logique applicative s'exécute sur le serveur applicatif.

L'application se compose d'éléments que l'on qualifie d'*objets métiers*. Ces objets gèrent tout le moteur logiciel.

## La logique applicative

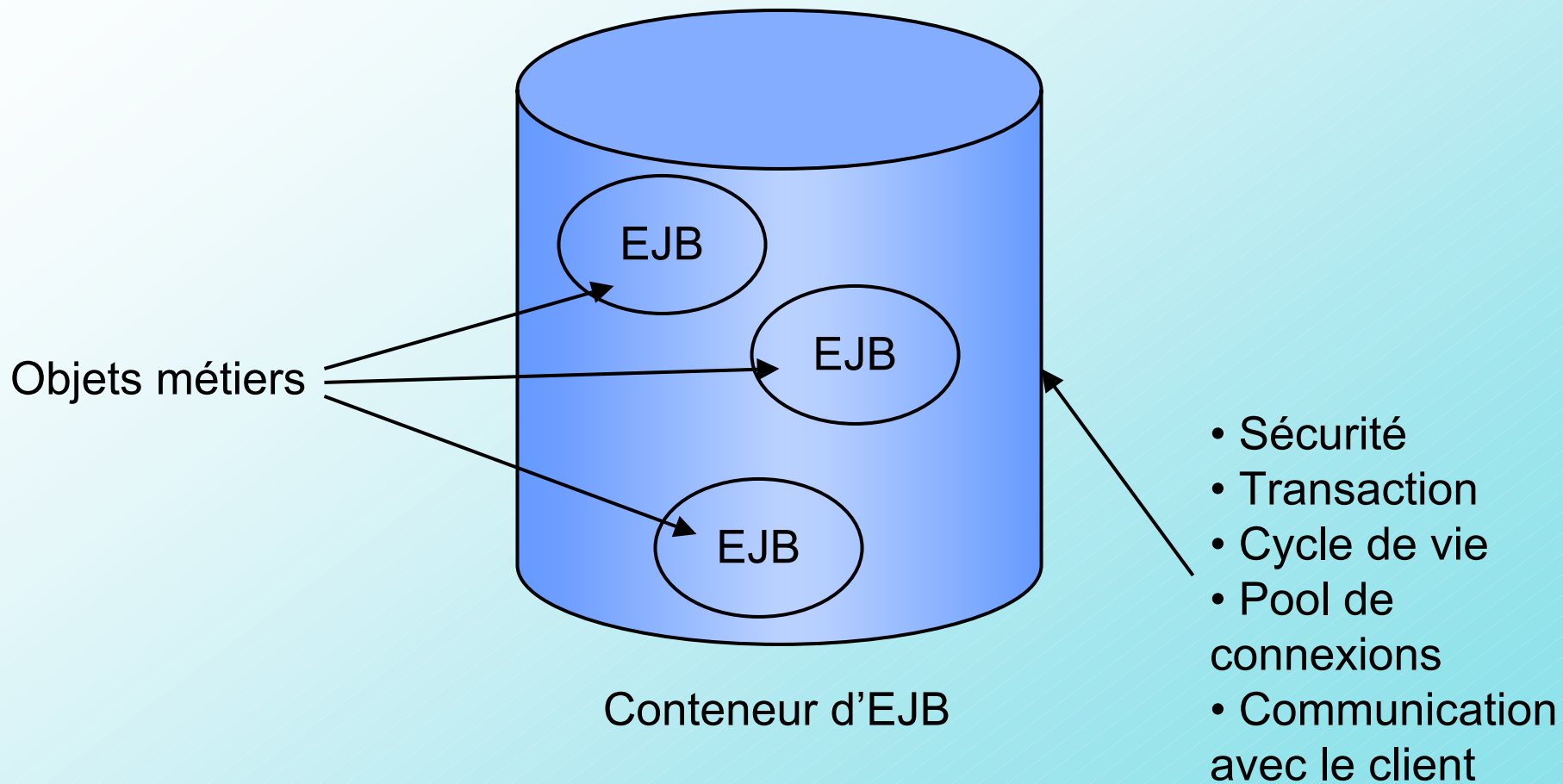
Les objets métiers de J2EE sont appelés les *EJB* ou *Enterprise JavaBean*

Les *EJB* peuvent être déployés sur plusieurs serveurs applicatifs et sont accessibles à distance.

Les *EJB* sont intégrés dans des conteneurs qui vont gérer :

- La gestion des transactions ( i.e. la communication entre les différents *EJB*),
- La sécurité,
- La communication entre le client et l'*EJB*,
- Le cycle de vie de l'*EJB*,
- La gestion d'un ensemble de connexions aux bases de données.

# La logique applicative



## La logique applicative

Les *EJB* assument deux rôles :

- La gestion du moteur applicatif, c'est à dire l'ensemble des traitements nécessaires au fonctionnement de l'application,
- La persistance de l'ensemble des données inhérentes à chaque *EJB*.

Les conteneurs d'*EJB* permettent de se concentrer sur la logique métier.

## J2EE et les architectures I\*net

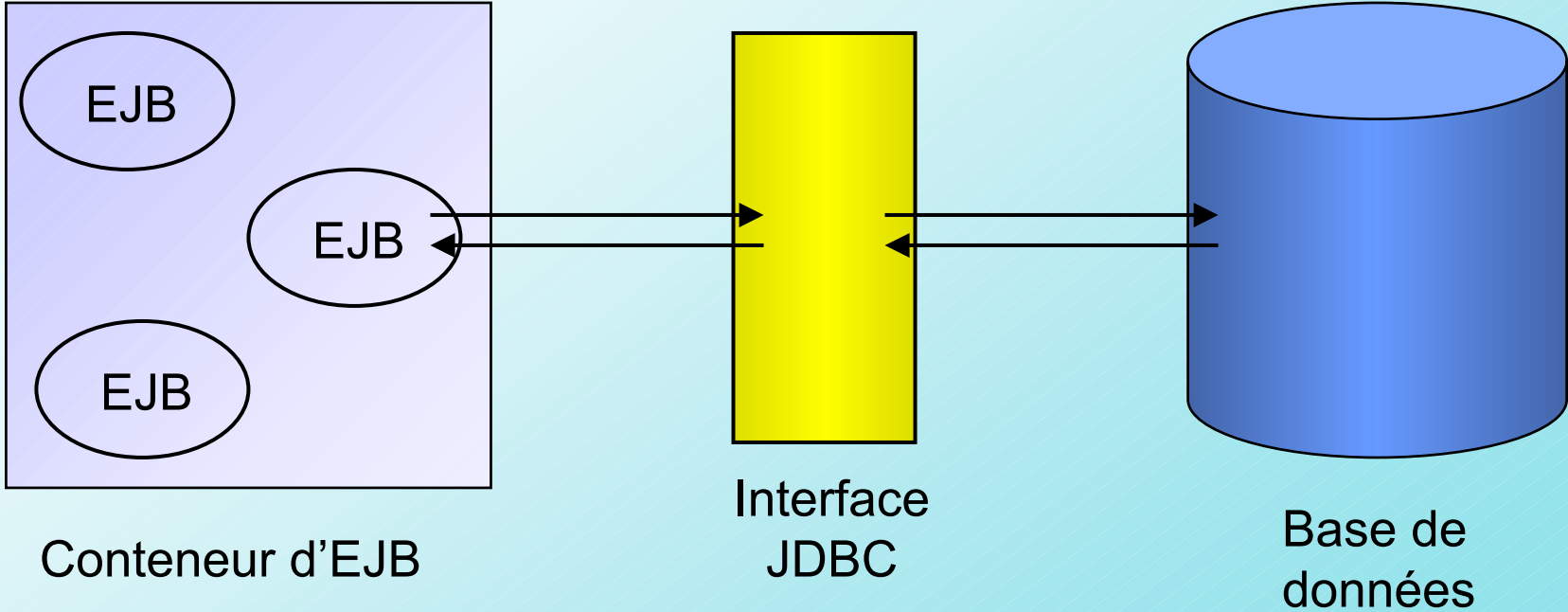
- **Le langage Java**
- **J2EE : un ensemble de solutions**
- **Les modules du serveur Internet**
- **La logique applicative**
- **J2EE et les bases de données**
- **Le modèle MVC**
- **Le modèle MVC 2**

Les EJB gèrent leur propre persistance en base de données.

La technologie *JDBC* ( *Java DataBase Connectivity* ) est une API Java efficace permettant de communiquer avec de nombreux types de bases de données.

JDBC permet l'échange d'informations et de requêtes entre l'application et la base de données via le gestionnaire associé à la base.

# J2EE et les bases de données

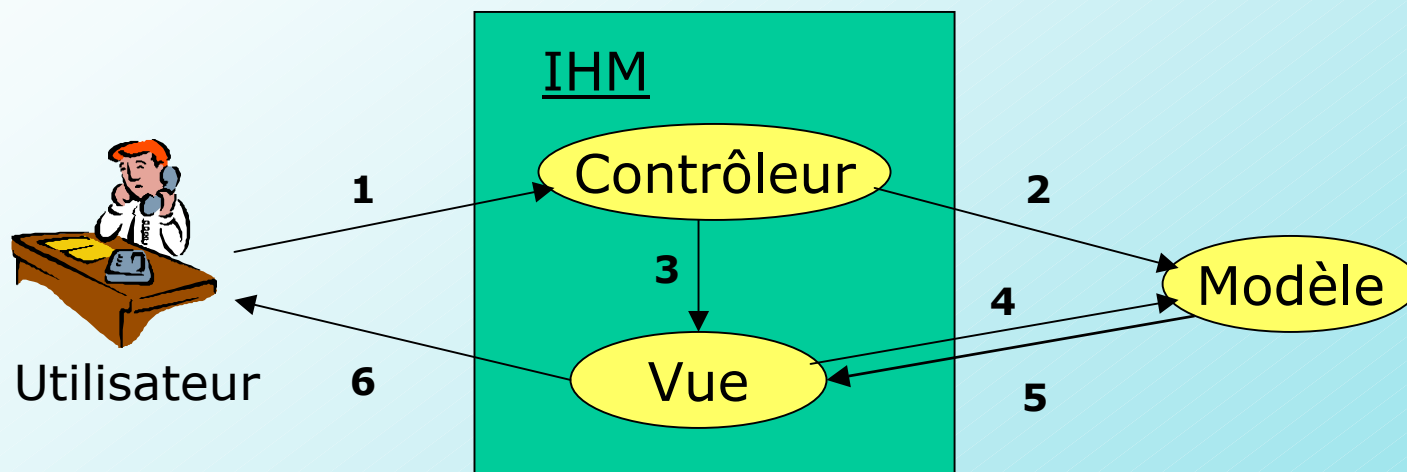


## J2EE et les architectures I\*net

- **Le langage Java**
- **J2EE : un ensemble de solutions**
- **Les modules du serveur Internet**
- **La logique applicative**
- **J2EE et les bases de données**
- **Le modèle MVC**
- **Le modèle MVC 2**



- **Design Pattern MVC (Modèle-Vue-Contrôleur)**



Cheminement d'une requête lorsque l'utilisateur agit sur l'interface

## Le modèle MVC

Le modèle contient la logique et l'état de l'application.

La vue représente l'interface utilisateur.

Le contrôleur gère la synchronisation entre la vue et le modèle.

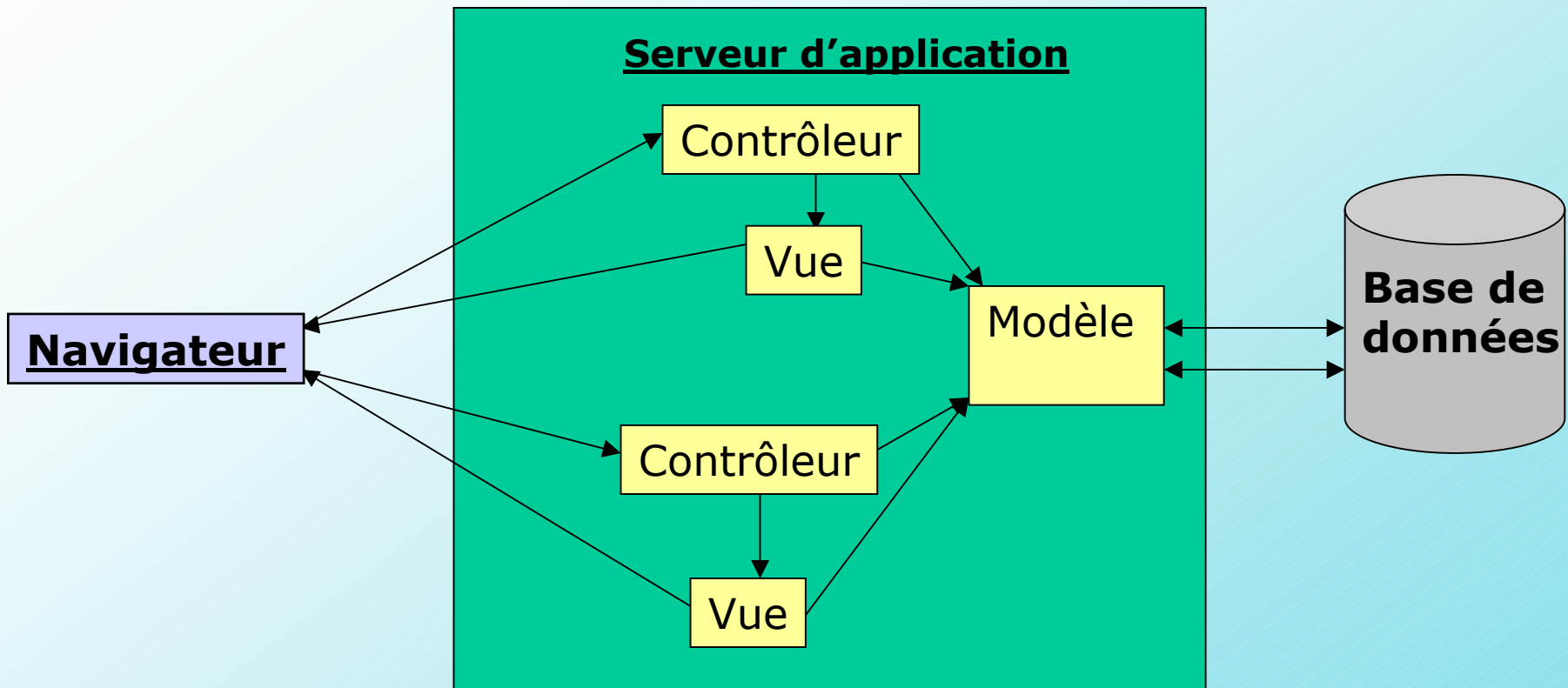
Le contrôleur réagit aux actions de l'utilisateur en effectuant les actions nécessaires sur le modèle.

Le contrôleur surveille les modifications du modèle et informe la vue des mises à jour nécessaires.

## Le modèle MVC

1. L'utilisateur manipule l'interface homme/machine. Un événement est envoyé. Cet événement est récupéré par le contrôleur.
2. Le contrôleur effectue l'action demandée par l'utilisateur en appelant les méthodes nécessaires sur le modèle.
3. Le contrôleur informe la vue d'un changement d'état du modèle.
4. La vue interroge le modèle afin de connaître son état.
5. Le modèle donne les renseignements nécessaires à la vue.
6. L'utilisateur voit le résultat de son action.

- MVC appliqué au WEB**



Les Contrôleurs sont des servlets,  
 Les vues sont des pages JSP,  
 Le modèle est composé d'EJB, ou de Java bean

- **MVC**

Le paradigme MVC est une avancée importante en terme d'architecture d'applications WEB.

Elle n'est cependant pas encore idéale : elle oblige à écrire une multitude de servlets, qui sont autant de points d'entrée dans l'application.

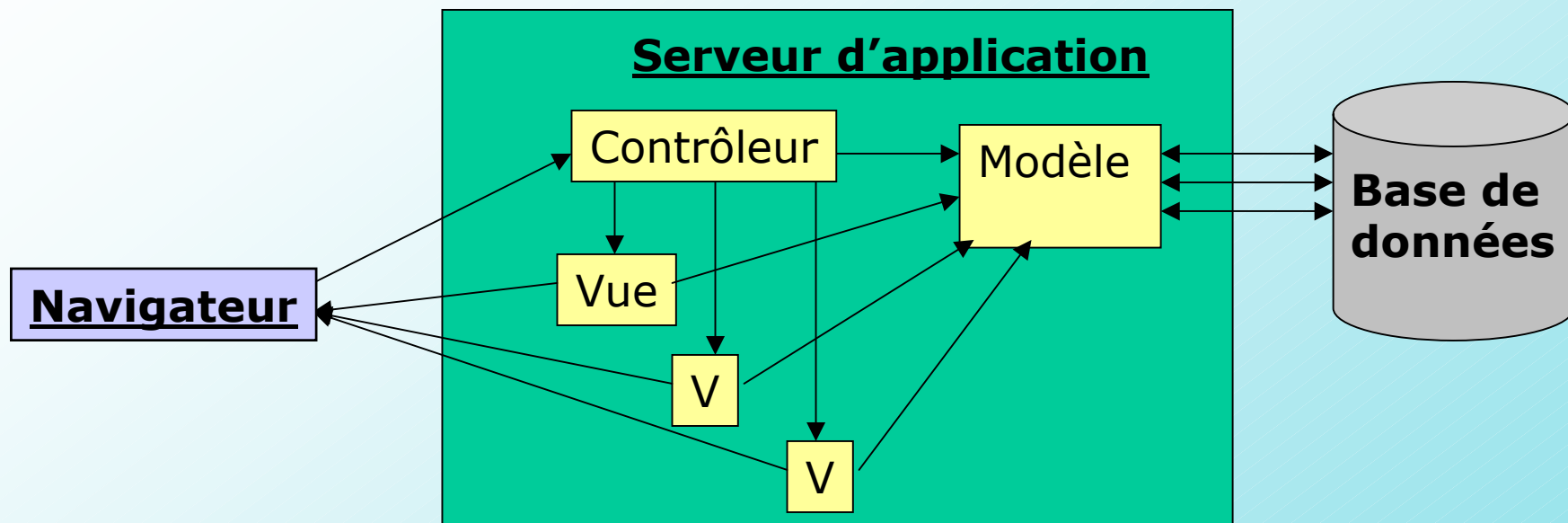
Pour palier cet inconvénient des frameworks ont été développés.

Ces frameworks qui sont composés d'une seule servlet (un seul contrôleur) sont regroupés sous l'étiquette "Model 2" encore appelé "MVC2".

## J2EE et les architectures I\*net

- **Le langage Java**
- **J2EE : un ensemble de solutions**
- **Les modules du serveur Internet**
- **La logique applicative**
- **J2EE et les bases de données**
- **Le modèle MVC**
- **Le modèle MVC 2**

## • Design Pattern MVC II



- Ici un seul contrôleur paramétrable,
- Maintenance Facilité
  - Factorisation de traitements relatifs comme
    - . l'authentification,
    - . la sécurité,
    - . la gestion des erreurs
    - . etc.

## J2EE en quelques mots

### **J2EE est une spécification pour les serveurs d'application.**

Un serveur J2EE doit fournir :

→ Un environnement d'exécution pour les applications WEB :

- Conteneur d'EJB,
- Conteneur de servlets.

→ Un ensemble de services techniques pour simplifier la tâche du développeur : RMI, EJB, JNDI, JDBC, JTA, JMS, JSP, etc.



# L'architecture J2EE

