

Programmation d'applications d'entreprise

Chapitre 1 – Introduction
UFR Sciences et Techniques - Antenne de Blois
Sameh.kchaou@univ-tours.fr
Supports de cours adaptés de Anthony Bocci

Organisation

- ❑ **CLÉ CELENE : PAEL3**
- ❑ **Séances de cours : 8**
- ❑ **Séances de TD : 6**
- ❑ **Séances de TP : 6**

Évaluation

- Contrôle continu
- TP
- Projet

Objectifs

- ❑ Comprendre les architectures d'applications Web :
 - ✓ Client – Serveur
 - ✓ 3-tiers
 - ✓ Multi-tiers
- ❑ Maîtriser les principales technologies Java EE :
 - ✓ Servlet
 - ✓ JSP – JSTL
 - ✓ JSF
 - ✓ EJB

Plan

- ☐ Définition
- ☐ Avantages des applications d'Enterprise
- ☐ Concepts de base
- ☐ Architecture Centralisée
- ☐ Architecture Client/serveur
- ☐ Architectures Distribuées

Définition

- ❑ Les applications d'entreprise sont des solutions logicielles conçues pour simplifier et automatiser divers processus dans les opérations d'une organisation.

- ❑ Objectifs principaux :
 - ✓ Accroître la productivité
 - ✓ Améliorer l'efficacité
 - ✓ Favoriser la collaboration entre les services

Avantages des applications d'entreprise

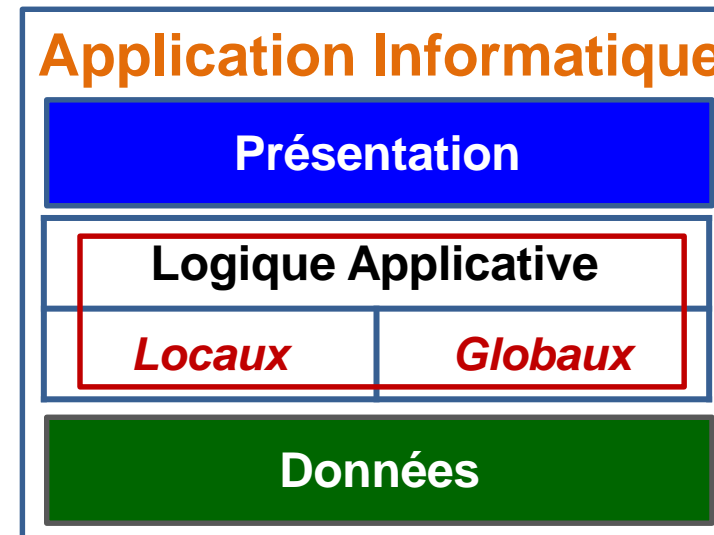
- ☐ Réduction des coûts
- ☐ Portabilité
- ☐ Évolutivité
- ☐ Maintenabilité
- ☐ Montée en charge
- ☐ Sécurité

Concepts de base

❑ Application Informatique : Trois niveaux d'abstraction

- ✓ Une application informatique peut être découpée en trois niveaux d'abstraction distincts:

- Couche Présentation
- Couche Logique Applicative
- Couche Données



Concepts de base

❑ Application Informatique : Trois niveaux d'abstraction

➤ Couche Présentation

- Couche IHM (Interface Homme-Machine)
- Permet l'interaction d'application avec l'utilisateur :
Elle contrôle les saisies au clavier et à la souris
ainsi que la présentation à l'écran,
- Doit être conviviale

Application Informatique

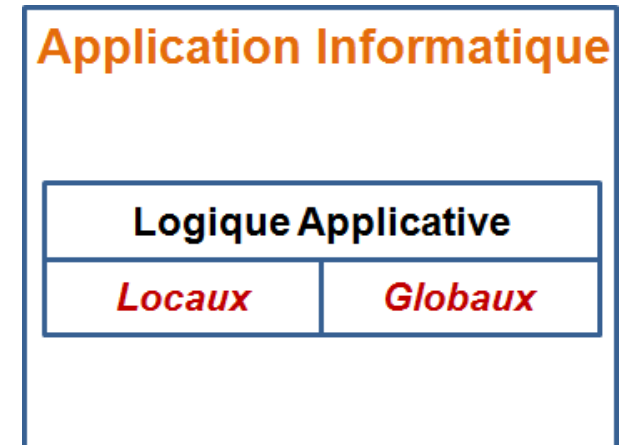
Présentation

Concepts de base

❑ Application Informatique : Trois niveaux d'abstraction

➤ Couche Logique Applicative

- Couche Métier ou couche Traitements
- Décrit les traitements à exécuter par l'application afin de répondre aux besoins des utilisateurs
- Deux types de traitements :
 - *Traitements locaux* : tiennent compte les contrôle effectués au niveau du dialogue avec l'IHM (formulaires, champs, boutons,...)
 - *Traitements globaux* : représentent les règles de l'application appelées aussi **logique métier (Business Logic)**

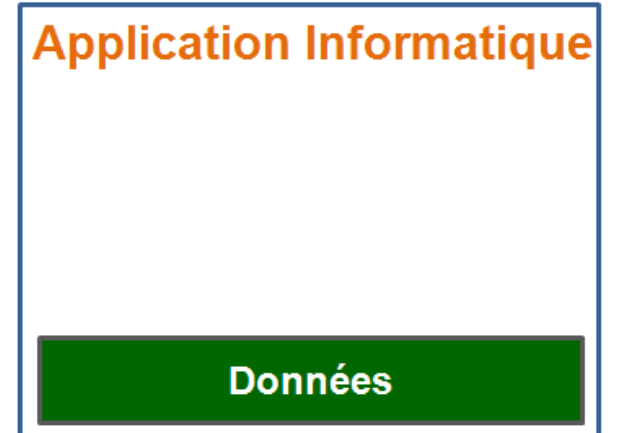


Concepts de base

❑ Application Informatique : Trois niveaux d'abstraction

➤ Couche Données

- Couche Persistance
- Prend en charge les actions liées aux accès aux données
 - Regroupe l'ensemble des mécanismes permettant la gestion des informations stockées par l'application
- Garantie souvent les fonctions classiques d'un SGBD à savoir :
 - Définition de données
 - Manipulation de données
 - Sécurité de données



Concepts de base

❑ Application Informatique : Trois niveaux d'abstraction

- Le **découpage** et la **répartition** des trois niveaux d'abstraction d'une application informatique permettent de distinguer plusieurs types d'architecture à savoir :
 - Architecture 1-tiers ➔ Architecture **Centralisée**,
 - Architecture 2-tiers ➔ Architecture **Client/serveur**,
 - Architecture 3-tiers ➔ Architectures **Distribuées**,
 - Architecture n-tiers ➔ Architectures **Distribuées**.

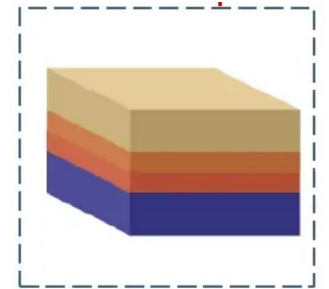
Architectures d'applications Web

❑ Architecture 1-tier (niveau)

- Les trois couches d'abstraction sont **intimement liées** et s'exécutent sur la **même machine**

=> l'informatique centralisée

- Deux types d'architecture mettant en œuvre des applications 1-tiers, notamment dans le contexte multi-utilisateurs
 - ✓ Applications sur site central (Mainframe)
 - ✓ Applications 1-tiers réparties (déployées) sur des machines indépendantes communiquant par partage de fichiers.



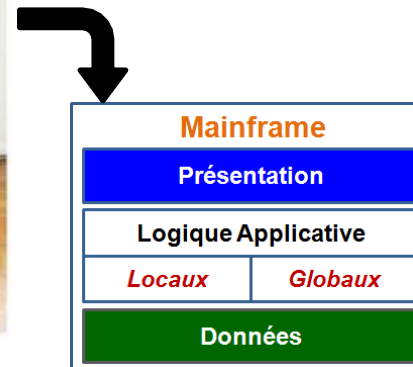
Architecture 1-tier

Architectures d'applications Web

❑ Architecture 1-tier (niveau)

➤ Applications Mainframes

- Le **serveur central** prend en charge l'intégralité des **traitements** et des **données**, y compris la présentation gérant l'affichage qui est simplement déporté sur des terminaux passifs.

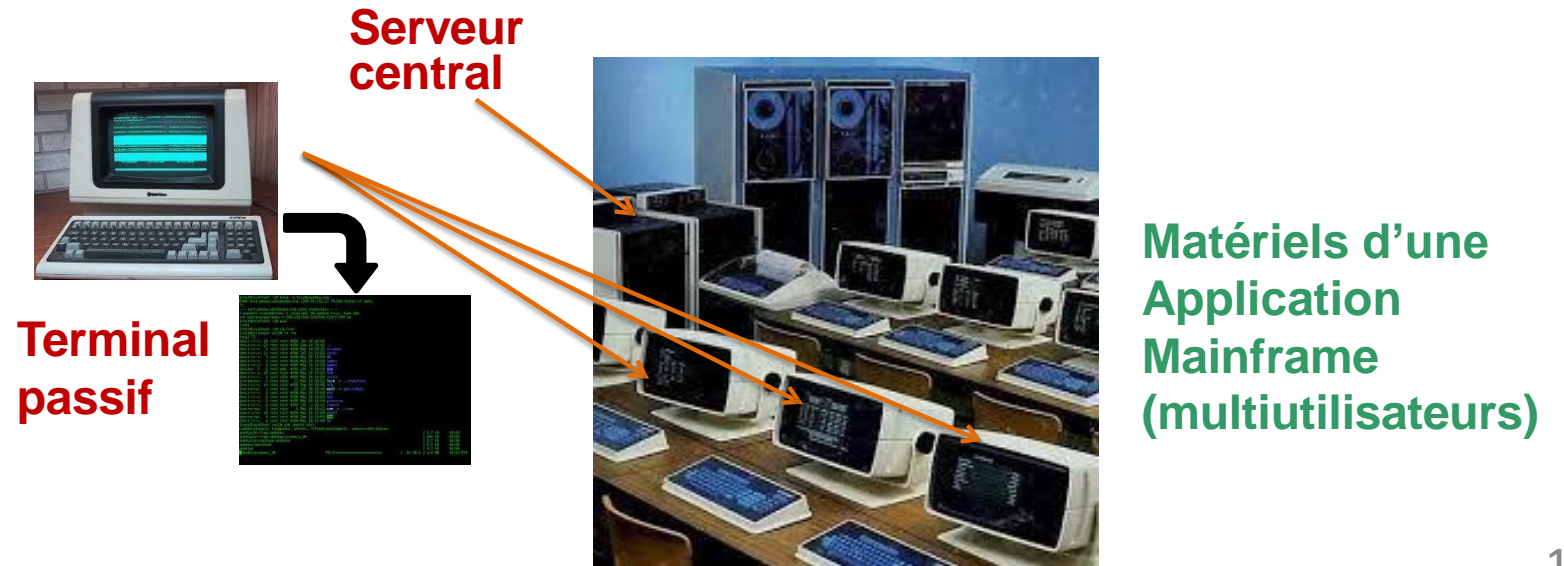


Architectures d'applications Web

❑ Architecture 1-tier (niveau)

➤ Applications Mainframes

- Les utilisateurs se connectent aux applications exécutées par le **serveur central** (le **mainframe**) à l'aide de **terminaux passifs**



Architectures d'applications Web

❑ Architecture 1-tier (niveau)

➤ Exemple : **Systèmes bancaires**

- **Banques et institutions financières** utilisent des mainframes pour gérer des millions de transactions chaque jour. Cela inclut des opérations telles que les virements, la gestion des comptes, les transactions par carte de crédit, etc.

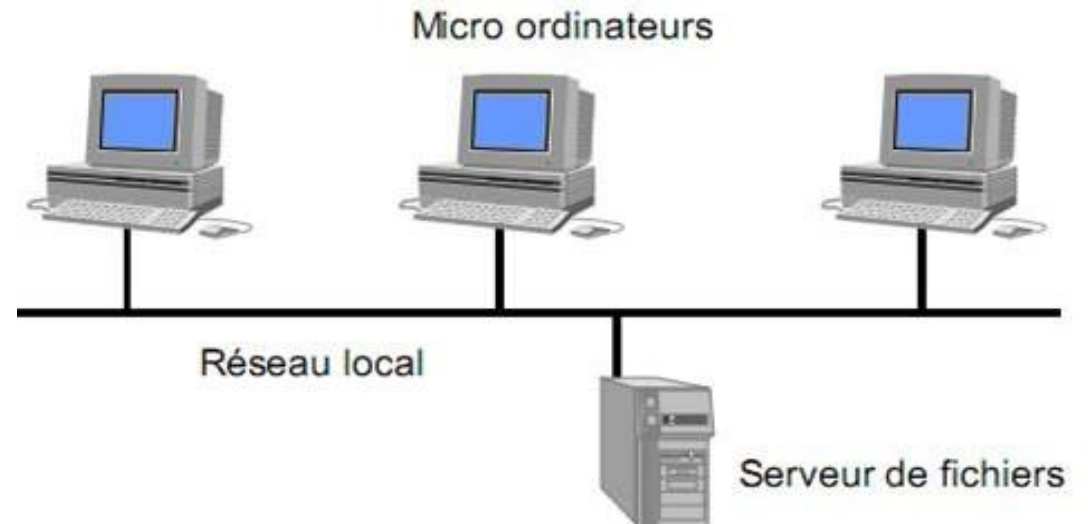
- **Exemple concret** : Les systèmes COBOL sont souvent utilisés pour des opérations telles que la gestion des comptes bancaires, la mise à jour des soldes en temps réel et les traitements par lot pendant la nuit.

Architectures d'applications Web

❑ Architecture 1-tier (niveau)

➤ Applications 1-tiers déployées

- C'est une application 1-tiers déployée sur plusieurs ordinateurs indépendants.
- Ce type d'architecture est souvent réservé à des applications non critiques exploitées par de petits groupes de travail qui partagent des fichiers de données stockés sur un serveur commun central.

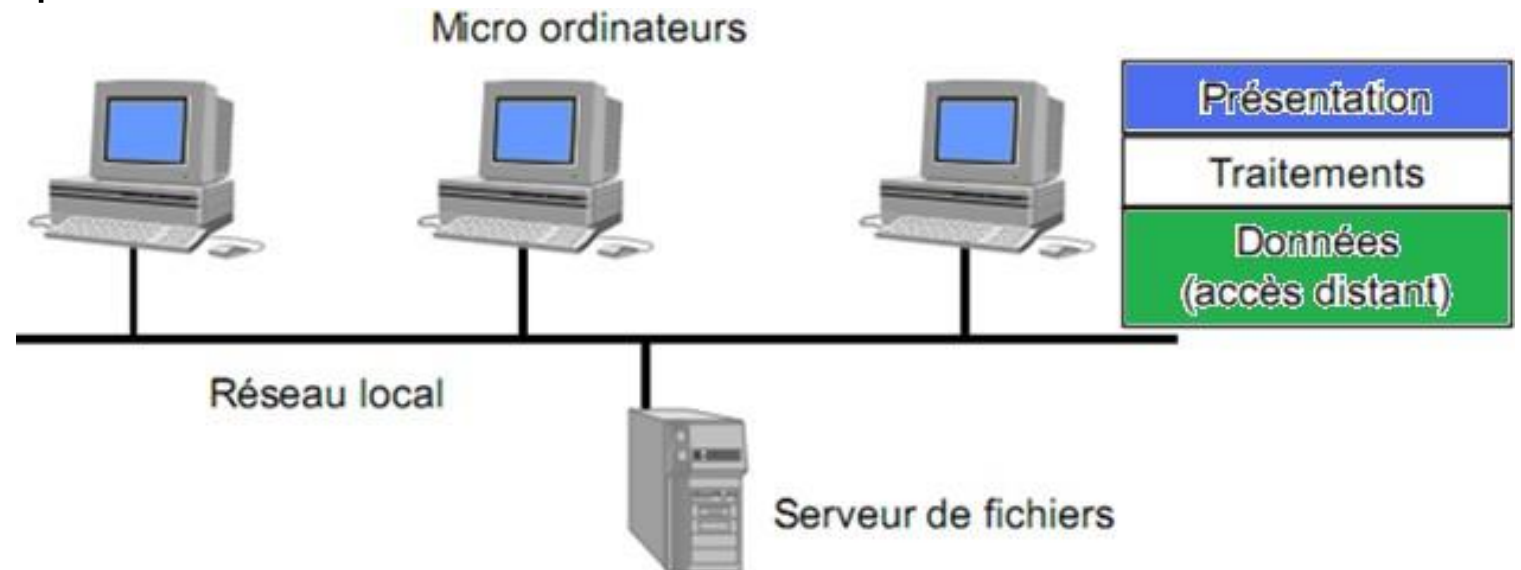


Architectures d'applications Web

❑ Architecture 1-tier (niveau)

➤ Applications 1-tiers Déployées

- Les **trois couches** sont également déployées sur plusieurs sites
- La gestion de données est exécuté indépendamment sur chaque site.
 - **Par conséquent**, la gestion des conflits d'accès doit être prise en charge par chaque site de manière indépendante.



Architectures d'applications Web

➤ Applications 1-tiers Déployées

➤ Exemple : Application de bureau (Desktop Application)

- **Microsoft Word** : C'est un traitement de texte installé directement sur un ordinateur. Toute la logique (édition de texte, mise en forme, enregistrement des fichiers) est exécutée localement sur la machine.
- **Caractéristiques** : L'utilisateur interagit avec l'interface directement sur son ordinateur. Il n'y a pas de séparation entre les couches de présentation, de logique métier et de gestion des données, elles sont toutes intégrées dans la même application.
- **Déploiement** : L'application est installée directement sur l'ordinateur de l'utilisateur. Aucun serveur ou autre infrastructure réseau n'est nécessaire pour exécuter l'application de base.

Architectures d'applications Web

❑ Architecture 1-tier : Synthèse

Avantages	Inconvénients
Mainframe : application fiable du fait que la gestion des données est centralisée.	Mainframe : mauvaise IHM du fait qu'elle est en mode caractère.
1-tiers déployée : IHM moderne	1-tiers déployée : application moins fiable au delà d'un certain nombre d'utilisateurs du fait que la gestion des données est déployée
Conclusion: afin de bénéficier des avantages des deux variantes, une solution permettant de déployer les couches de l'application comme suit : <ul style="list-style-type: none">• Centralisée la gestion de données (couche données)• Déployée la gestion des IHM (couche présentation)	
=> Nouveau concept baptisé architecture Client/serveur	

Architectures d'applications Web

□ Architecture Client/serveur

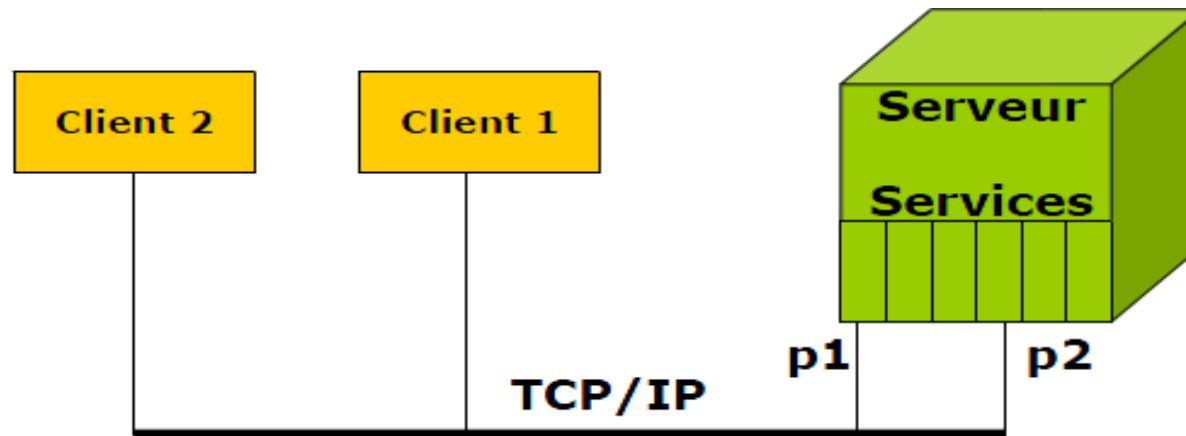
- Il désigne un **mode de communication** à travers un réseau entre plusieurs programmes (processus).
 - l'un, qualifié de **Client**, envoie des requêtes ;
 - l'autre ou les autres, qualifiés de **Serveurs**, attendent les requêtes des clients et y répondent.
- le **Client** désigne également l'**ordinateur** sur lequel est exécuté le **processus Client**.
 - Les clients sont souvent des ordinateurs personnels ou des appareils individuels (téléphone, tablette), mais pas systématiquement.
- le **Serveur**, l'**ordinateur** sur lequel est exécuté le processus **Serveur**.
 - les serveurs sont des ordinateurs dédiés au logiciel serveur qu'ils abritent, et dotés de **capacités supérieures** à celles des ordinateurs personnels en ce qui concerne la **puissance de calcul**, les **entrées-sorties** et les **connexions réseau**.
 - Un **serveur** peut répondre aux requêtes d'un grand nombre de clients.

Architectures d'applications Web

❑ Architecture Client/serveur

➤ Schéma de fonctionnement

- Le **Serveur** fournit des **services** qui seront éventuellement sollicités par les clients via l'envoi des **requêtes**.

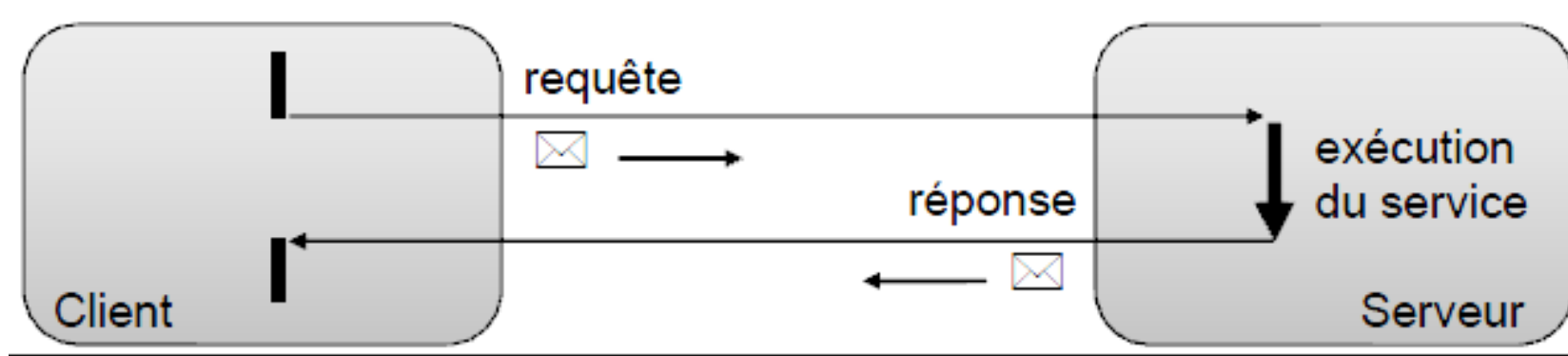


Architectures d'applications Web

❑ Architecture Client/serveur

➤ Schéma de fonctionnement

- Le dialogue entre le client et le serveur est initié par le **Client**.
- Le **serveur** décode le message de la **requête** afin d'extraire les paramètres nécessaires pour **l'exécution du service**.
- Enfin, un deuxième message de **réponse** sera envoyé au **Client**



Architectures d'applications Web

❑ Architecture Client/serveur

➤ Exemple : Site Web (Navigateur Web - Serveur Web)

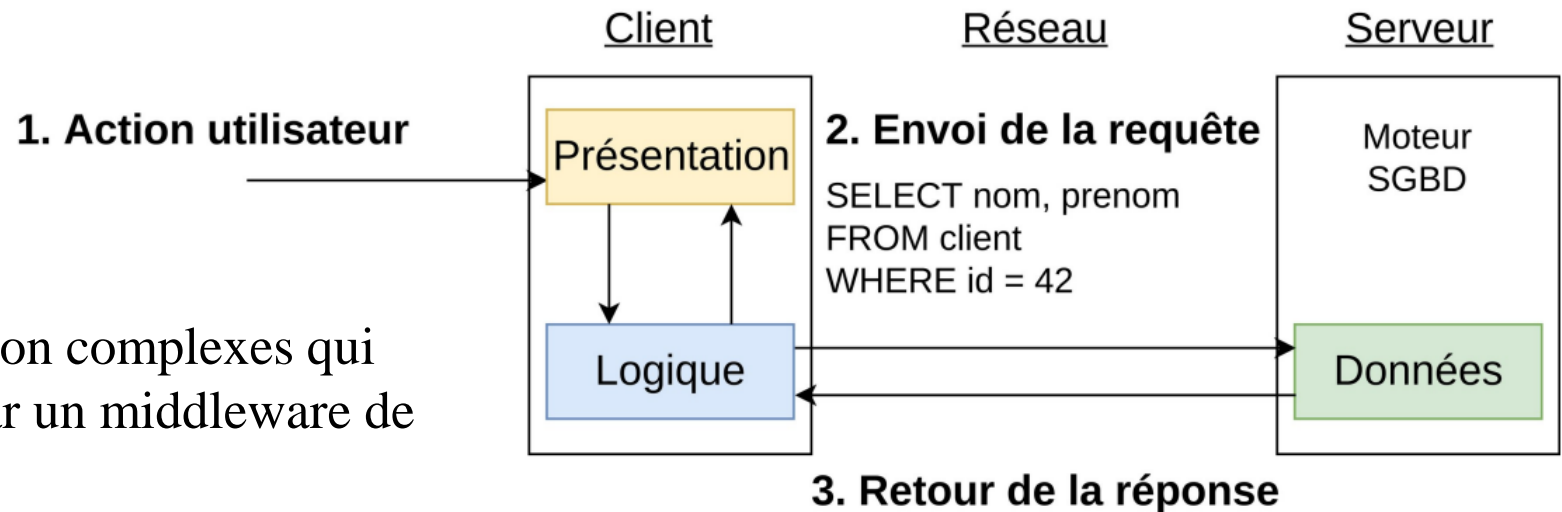
- **Client** : Le navigateur web (comme Google Chrome, Firefox ou Safari) sur l'ordinateur ou le téléphone de l'utilisateur.
- **Serveur** : Un serveur web (comme Apache ou Jboss) qui héberge le site web et répond aux requêtes HTTP envoyées par le navigateur.
- **Exemple** : Lorsqu'un utilisateur accède à un site comme **Amazon**, son navigateur envoie une requête au serveur d'Amazon. Le serveur renvoie ensuite la page web qui est affichée sur l'écran de l'utilisateur.

Architectures d'applications Web

❑ Architecture 2-tiers

- Le client s'occupe de la logique et de la présentation
- Le serveur s'occupe de la gestion des données
- Les données sont gérées par un SGBD, souvent sur un serveur dédié (serveur de données).
- Le serveur de données comprend un langage de requêtes, souvent le SQL.

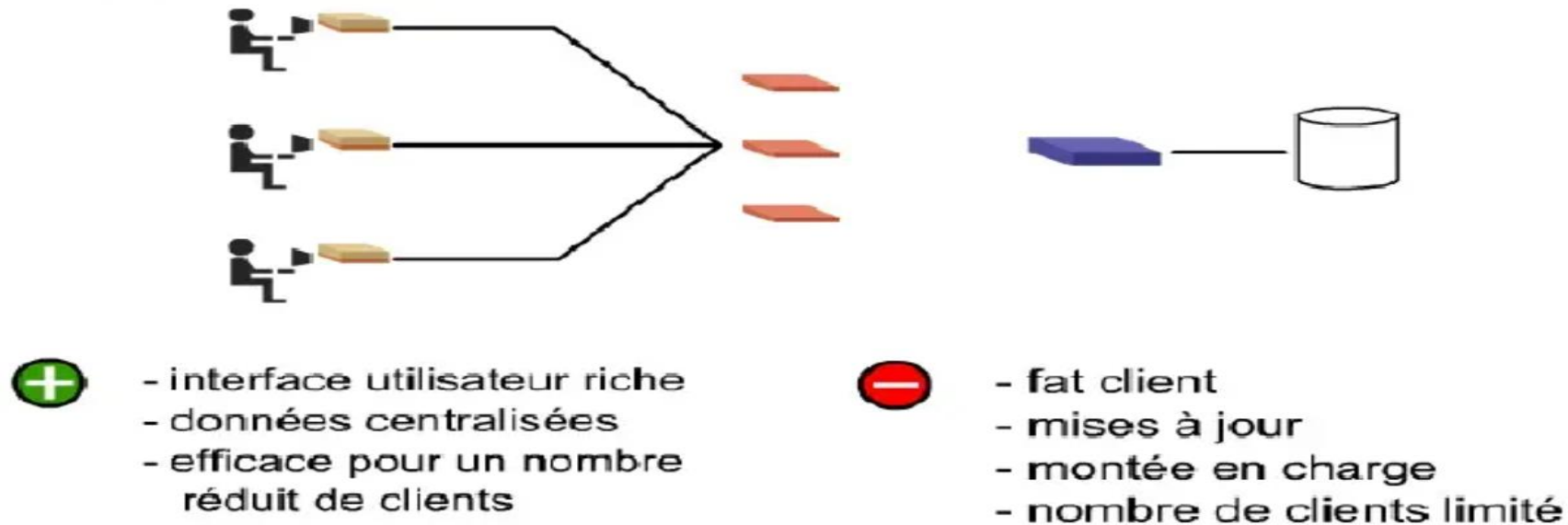
=> Mécanismes de communication complexes qui sont en général pris en charge par un middleware de base de données



Walker, Cordell

Architectures d'applications Web

❑ Architecture Client/serveur (2-tiers)



❑ Ce type d'architecture est grandement rigidifié par les coûts et la complexité des maintenances

→ **Solution:** architecture plus évoluée facilitant les forts déploiements à moindre coût → **architectures distribuées**

Architectures d'applications Web

❑ Architecture distribuée (3-tiers)

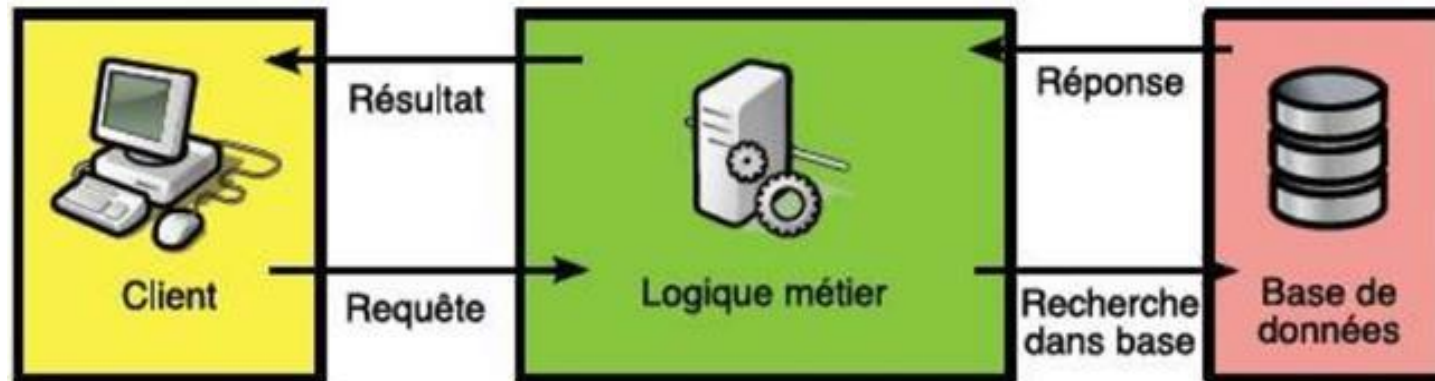
- Son principe de base consiste à séparer **trois couches logicielles**, contenues dans une application, sur **trois niveaux ou machines**.
- Cette séparation a pour but de rendre indépendante chacune des couches afin de faciliter la maintenance et les évolutions futures de l'application.
- Elle assure une sécurité plus importante car l'accès à la base de données n'est autorisé que par la couche de traitements.
- Elle a également l'avantage d'optimiser le travail en équipe et le développement multi-cibles.
- Les données sont toujours gérées de façon centralisée;
- La présentation est toujours prise en charge par le poste client;
- La logique applicative est prise en charge par un serveur intermédiaire;

Architectures d'applications Web

❑ Architecture distribuée (3-tiers)

➤ Fonctionnement

- (1) **Client** : envoie une requête au serveur applicatif
- (2) **Serveur applicatif** : exécute le service sollicité tout en demandant le serveur de BDD les données nécessaires pour le service
- (3) **Serveur de données** : renvoi les données sollicitées en exécutant des requêtes sur les BD.
- (4) **Serveur applicatif** : prépare le service puis le renvoi au client.



Architectures d'applications Web

❑ Architecture distribuée (3-tiers)

➤ **Avantages**

- Déploiement immédiat
- Évolutions transparentes pour l'utilisateur
- Moins de contraintes de caractéristiques pour le poste client

➤ **Inconvénients**

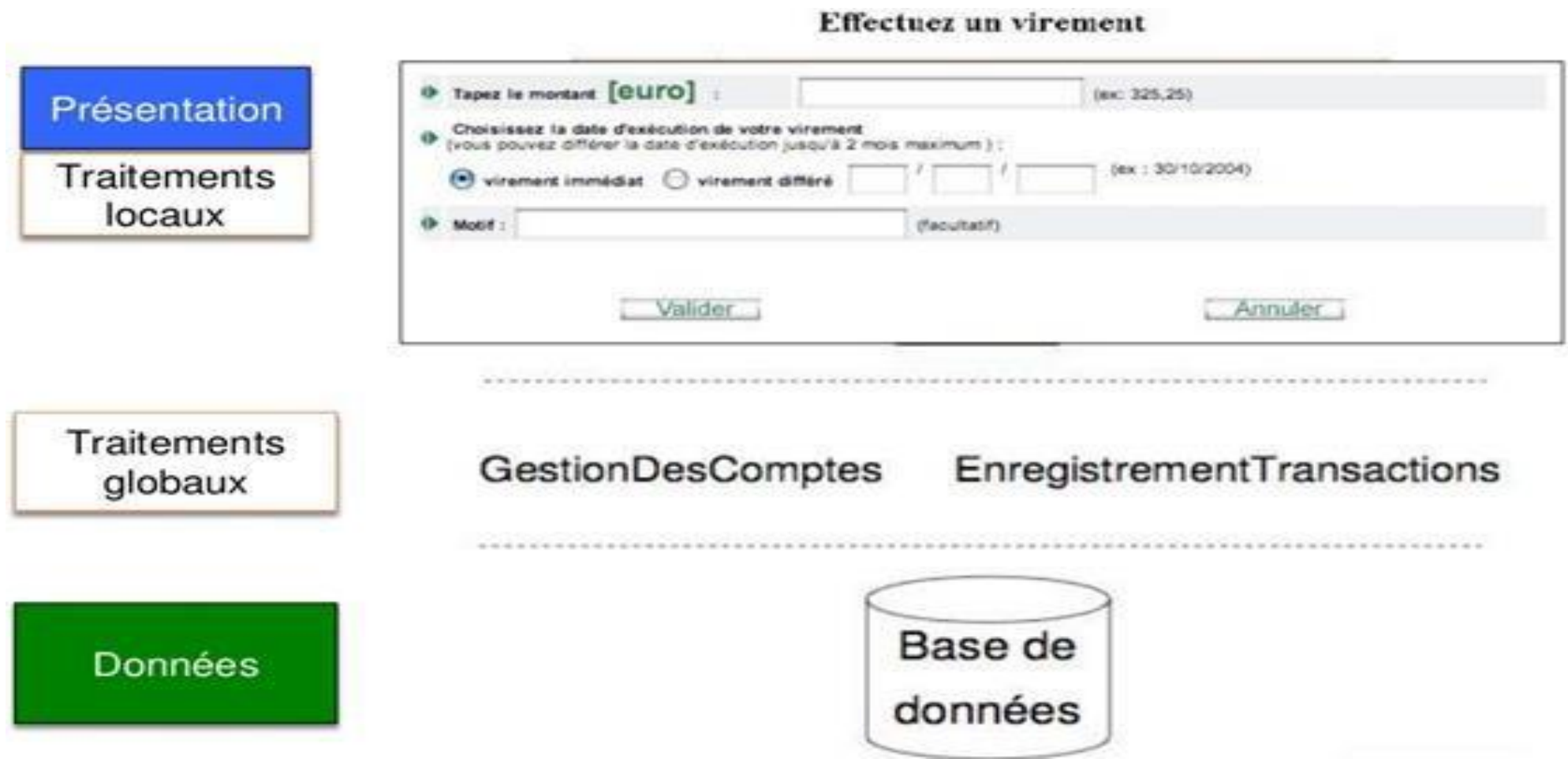
- Le serveur d'application gère la majorité des traitements
 - Problème de gestion de montée en charge rappelant les mainframes : Le serveur d'application gère la majorité des traitements, ceci peut causer des problèmes de montée en charge, lorsque de nombreux utilisateurs accèdent simultanément aux services.

=> L'équilibrage de la charge client/serveur semble atteint avec l'architecture suivante : n-tiers.

Architectures d'applications Web

❑ Architecture distribuée (3-tiers)

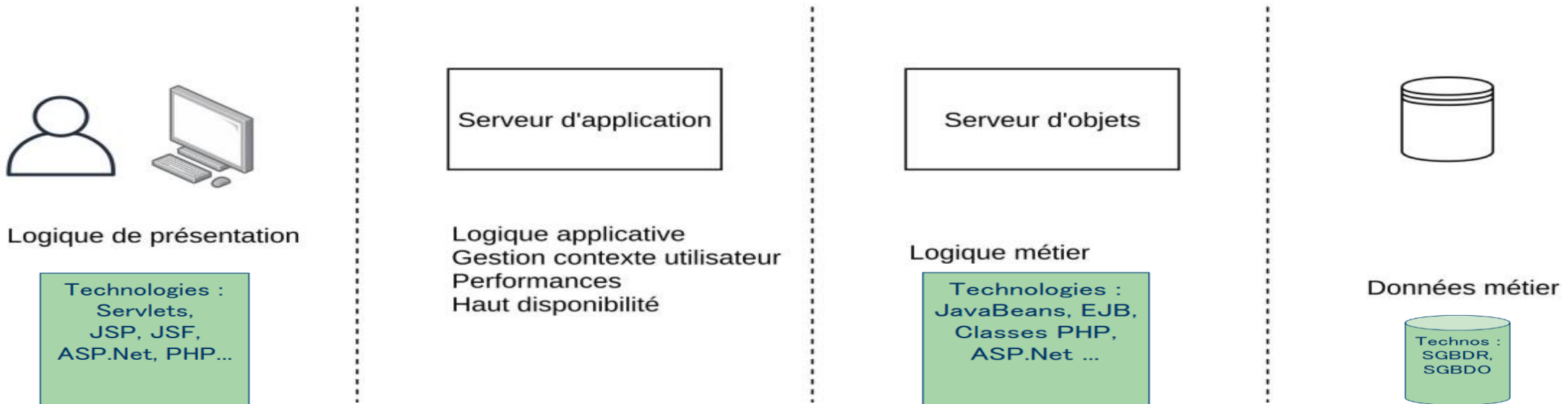
➤ Exemple



Architectures d'applications Web

❑ Architecture distribuée (n-tiers)

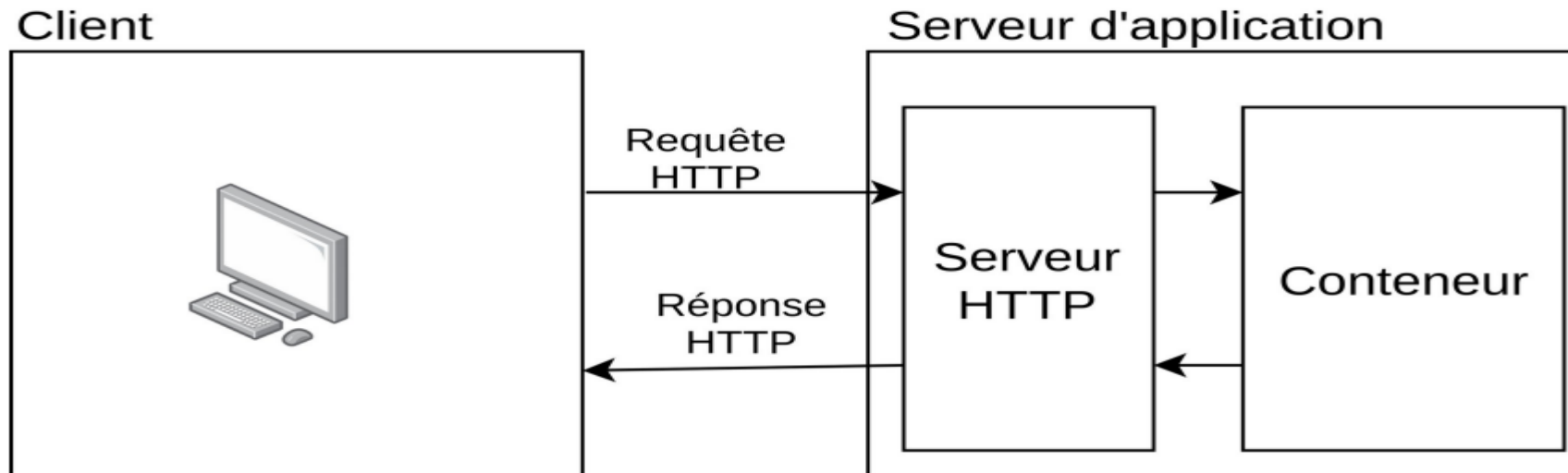
- Distribution de la logique applicative entre plusieurs serveurs
- comprend généralement une couche de présentation, une couche applicative, une couche objets métier et une couche d'accès aux données



Architectures d'applications Web

❑ Architecture distribuée (n-tiers)

- Serveur d'application : Environnement d'exécution des traitements côté serveur
- Le code applicatif, code métier, est stocké sur le serveur d'application, et est déployé et géré de manière centralisée.



QUIZ !

Qu'est-ce qu'une application d'entreprise ?

- a) Un logiciel conçu pour un usage personnel.
- b) Un système distribué destiné à répondre aux besoins d'une entreprise.
- c) Une application destinée à gérer des données personnelles.
- d) Un logiciel destiné uniquement à la gestion financière.

QUIZ !

Quel est le principal objectif des applications d'entreprise ?

- a) Fournir des jeux interactifs.
- b) Gérer les processus métiers de manière efficace.
- c) Permettre la communication entre employés.
- d) Gérer les réseaux sociaux.

QUIZ !

Qu'est-ce qui distingue une application d'entreprise d'un logiciel standard ?

- a) La complexité et la capacité de traiter des volumes élevés de données.
- b) Le fait qu'elle soit gratuite.
- c) La possibilité de personnalisation selon l'utilisateur.
- d) L'absence de maintenance nécessaire.

QUIZ !

Quelles sont les couches généralement comprises dans la logique applicative d'une architecture n-tiers ?

- a) Présentation, applicative, objet métier, accès aux données
- b) Présentation, gestion, stockage, sécurité
- c) Interface, communication, traitement, stockage
- d) Client, serveur, base de données, réseau

QUIZ !

Quel est le rôle principal du serveur d'application ?

- a) Stocker les données
- b) Exécuter le code métier
- c) Gérer l'interface utilisateur
- d) Fournir des mises à jour logicielles

QUIZ !

Dans une architecture 2-tiers, où se trouve généralement la logique métier ?

- a) Sur le client
- b) Sur le serveur
- c) Sur un serveur tiers
- d) Dans le réseau

QUIZ !

Quel type d'application est souvent associé à une architecture 2-tiers ?

- a) Applications web
- b) Applications de bureau
- c) Applications mobiles
- d) Applications cloud

QUIZ !

Quel est le rôle principal de la couche de présentation dans une architecture 3-tiers ?

- a) Gérer les données
- b) Exécuter la logique métier
- c) Afficher les informations à l'utilisateur
- d) Stocker les fichiers

QUIZ !

Dans une architecture 3-tiers, quel rôle joue la couche de données ?

- a) Gérer l'interface utilisateur
- b) Fournir des services de base de données
- c) Traiter les requêtes de l'utilisateur
- d) Afficher des rapports

QUIZ !

Dans une architecture 3-tiers, où se trouve la logique métier ?

- a) Dans la couche de présentation
- b) Dans la couche de données
- c) Dans la couche d'application
- d) Dans le client

Fin



Merci pour votre attention