

# Chapitre 1\_1: Concepts fondamentaux du développement web

---

ING 3 2025/2026

S. NOUFEL



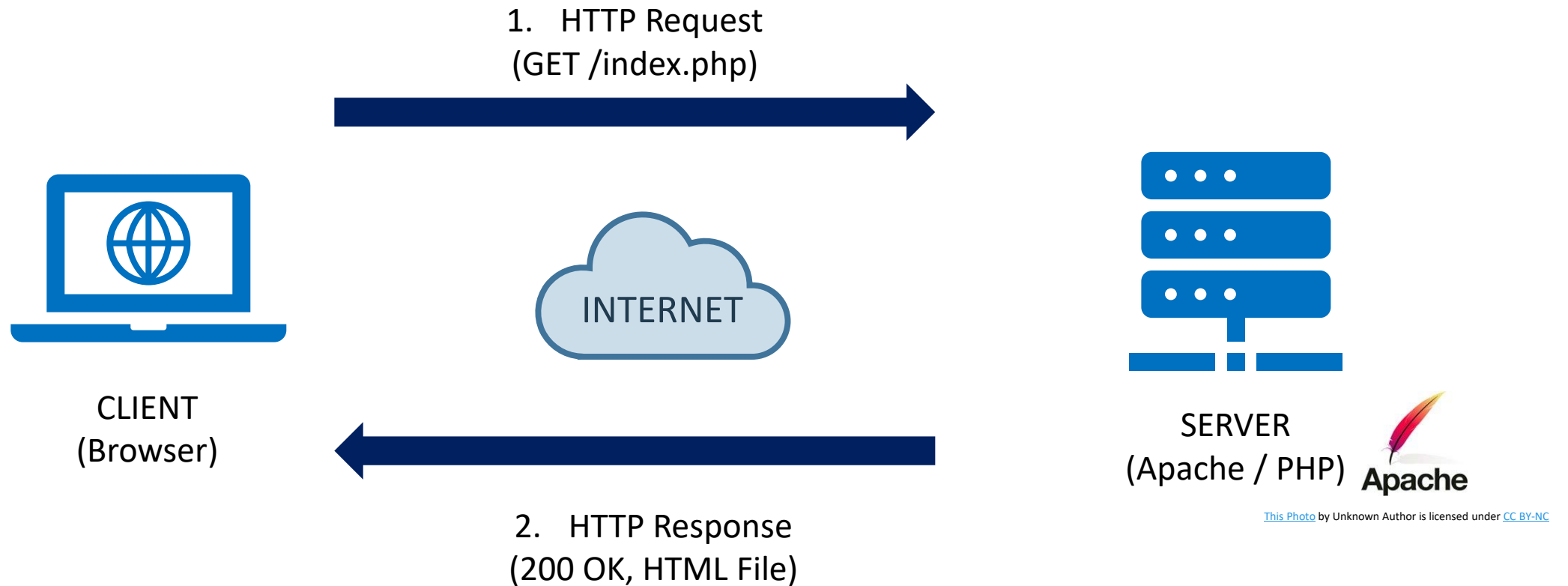
# Plan du cours

---

- I. Protocole HTTP
- II. Architecture Client Serveur
  - 1. 2-tiers
  - 2. 3-tiers
- III. Pattern MVC d'une application Web
  - 1. Modèle
  - 2. Vue
  - 3. Contrôleur

# Comment fonctionne le Web?

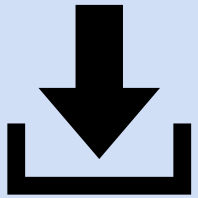
## (Le protocole HTTP)



# Hyper Text Transfert Protocol (HTTP)

---

**GET**



*Récupérer le  
contenu d'un  
document*

**POST**



*Soumissionner  
des formulaires*

**PUT**



*Envoyer des  
fichiers*

**DELETE**



*Supprimer un  
document*

**HEAD**

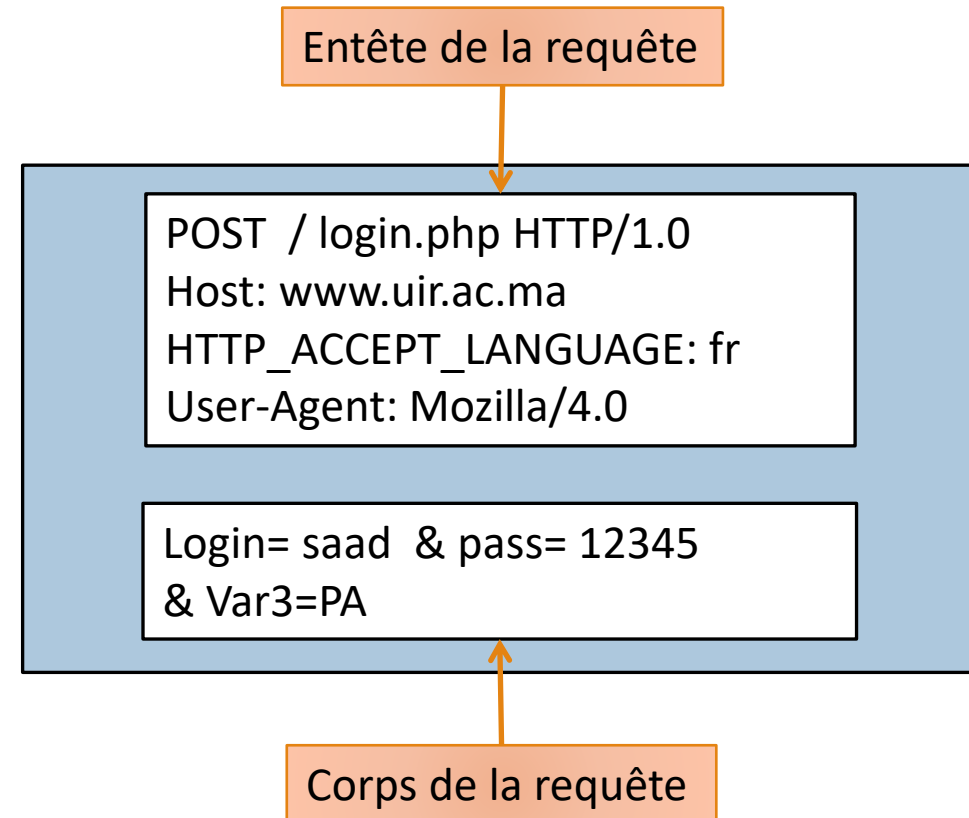


*Récupérer des  
informations*

# Hyper Text Transfert Protocol (HTTP)

---

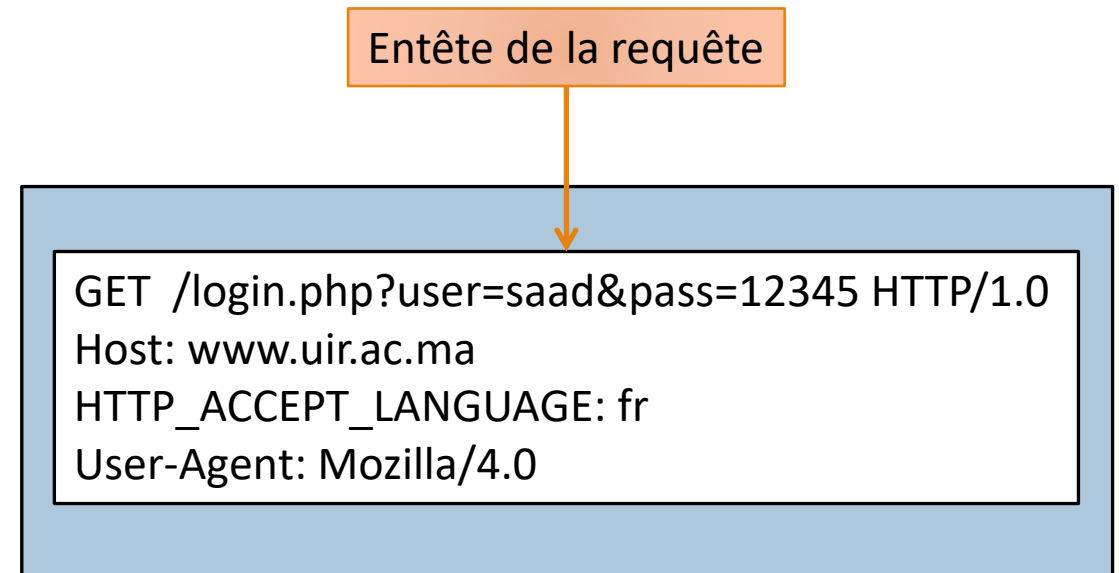
Requête du client en utilisant  
la méthode POST:



# Hyper Text Transfert Protocol (HTTP)

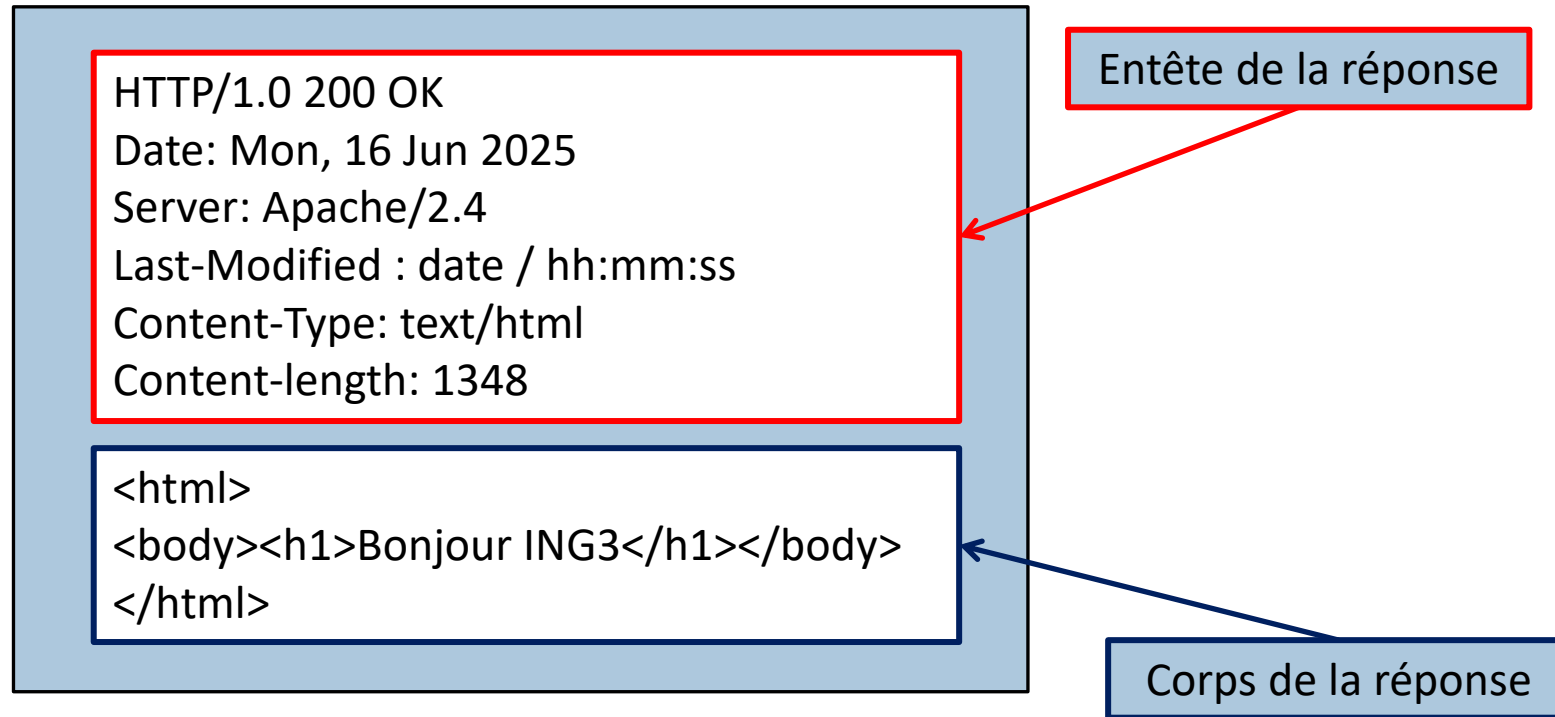
---

Requête du client en utilisant la méthode GET:



# Hyper Text Transfert Protocol (HTTP)

## Réponse du serveur



# Architecture monolithique

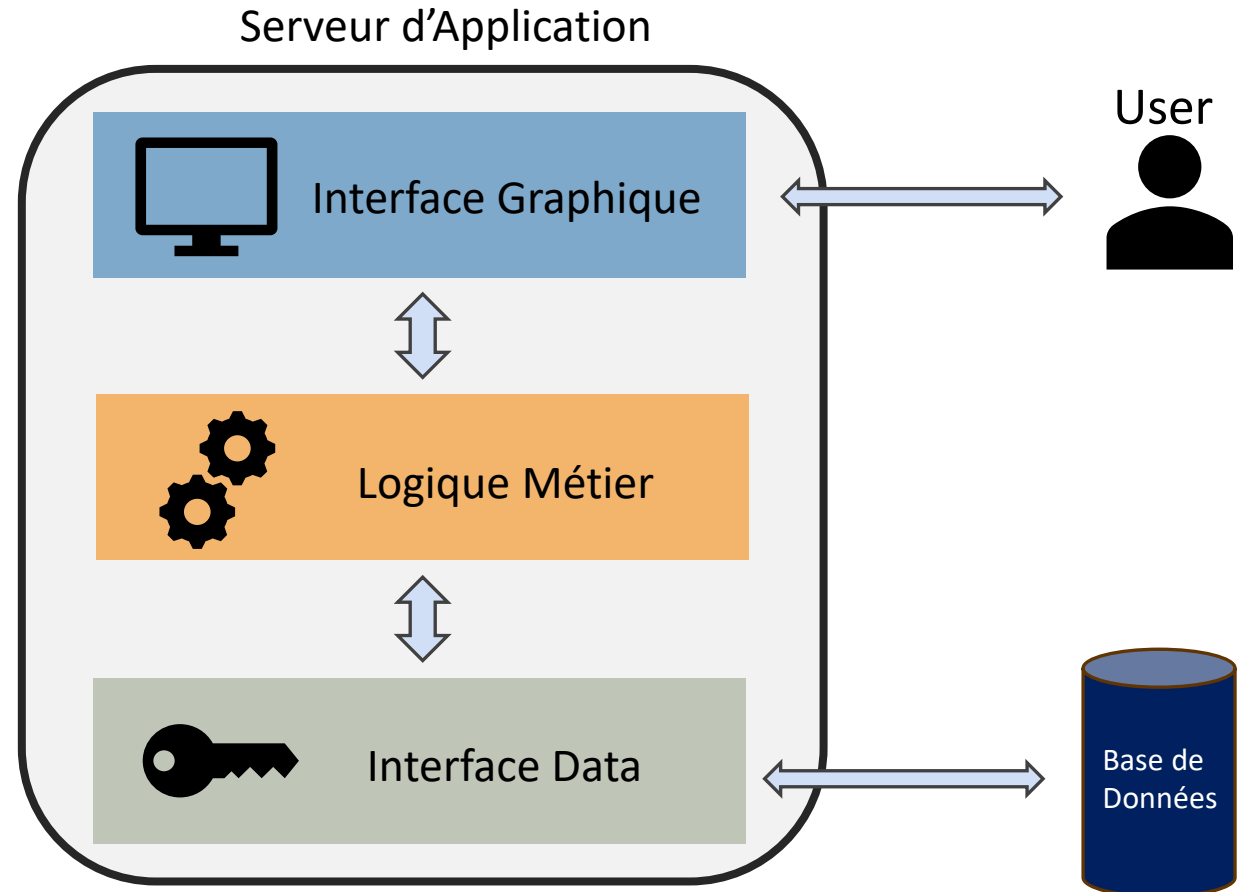
- Logicielle est développée comme une seule unité
- Déploiement sur la même unité
- Interdépendance entre UI, métier et la persistance de données

## Avantages

- Facilité de développement

## Inconvénients:

- Maintenance
- Un seule code base
- Scalling



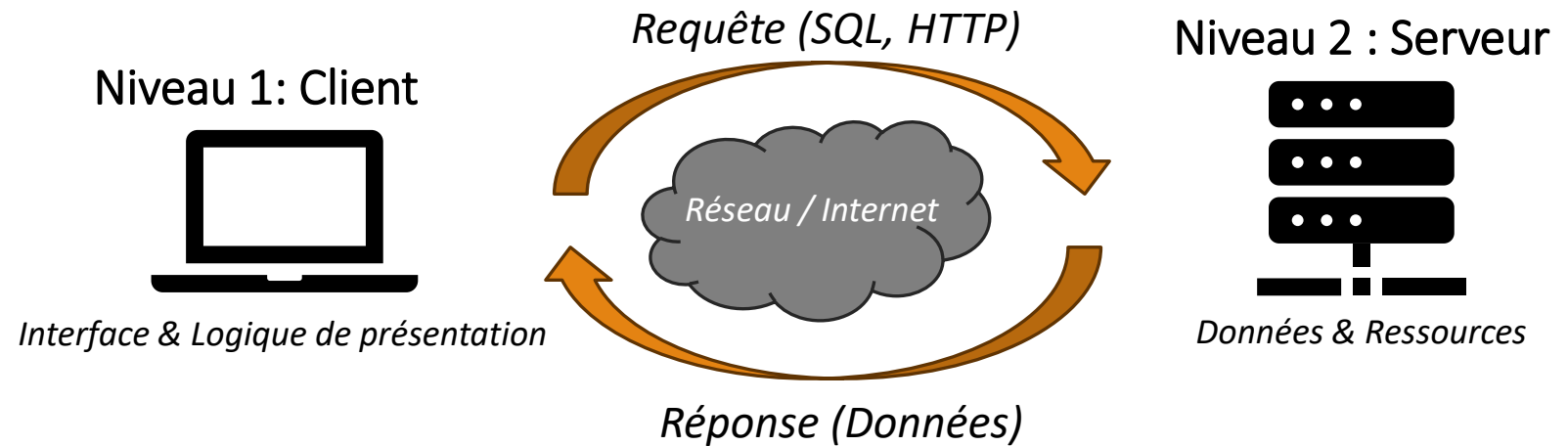


# Architecture client serveur (2-tiers)

---

L'architecture 2-tiers :

- Deux niveaux client et serveur
- Concerne les machines interconnectées par un réseau
- Une machine joue le rôle du client et envoie une demande (request, afficher une page web, pdf)
- Une machine joue le rôle du serveur renvoie la ressource demandée (Response, HTTP)



# Architecture client serveur (2-tiers)

---

## Avantages :

- La centralisation des données
- Facilite de gestion des ressources plus la sécurisation d'accès aux données

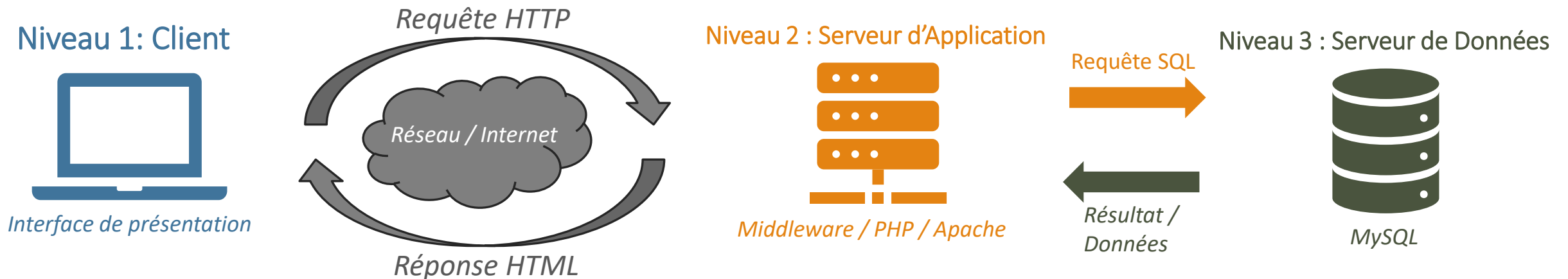
## Inconvénients :

- En cas de panne du serveur, la centralisation de toutes les ressources implique la suspension totale du fonctionnement du système

# Architecture client serveur (3-tiers )

Architecture 3-tiers implique des interactions entre trois niveaux client, serveur d'application, serveur de la BD.

- **Client** : envoie les requêtes au serveur d'application
- **Serveur d'application (middleware)** : assure le traitement et la collecte des données pour envoyer une réponse au client
- **Serveur de base de données** : assure la persistance des données et répond aux requêtes du serveur d'application



# Architecture client serveur (3-tiers )

---

## **Avantages**

- Architecture robuste en absence d'un nœud central
- Séparation des spécialités des développeurs front et back end
- Facilité d'évolution du système

## **Inconvénients**

- Coût
- Sécurité

# Architecture client serveur

---

Il existe plusieurs types de clients

- **Client léger:** navigateur qui demande une page web (html – sans javascript)
- **Client lourd:** application desktop et application mobile
- **Client riche:** client qui reçoit la réponse du serveur et la finalise avec de nouvelles requêtes au serveur en utilisant javascript /AJAX

# Architecture Micro-service (nouveau)

Application structurée en micro-service

Déploiement des micro-services d'une manière indépendante

Hétérogénéité des technologies utilisée pour l'implémentation des MS

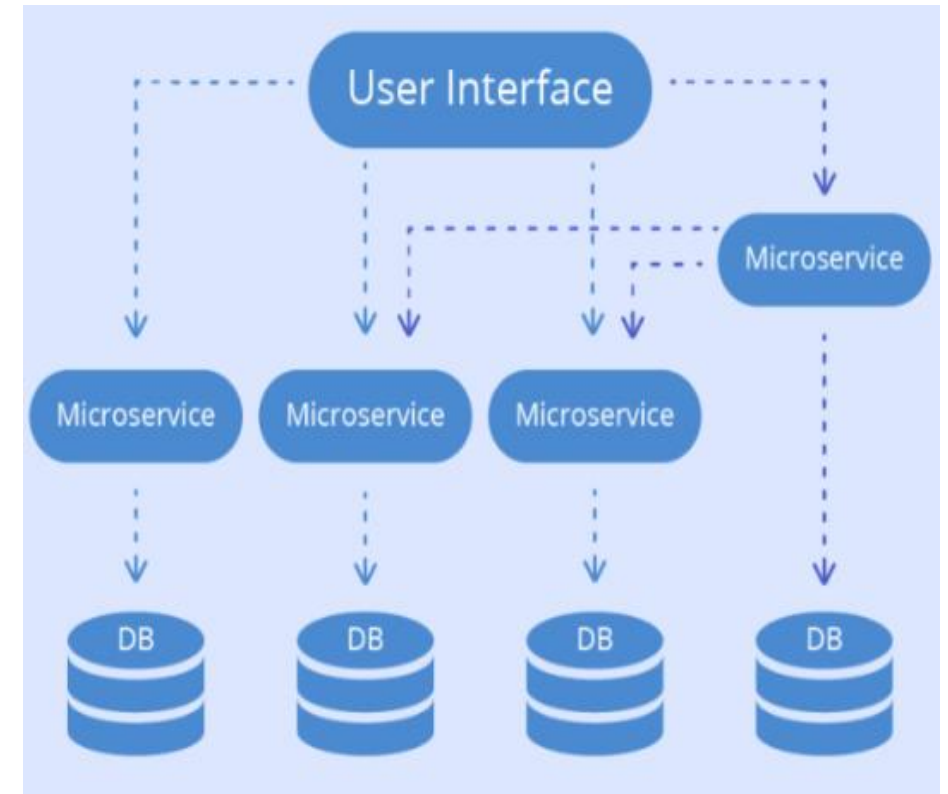
## Avantages :

➤ Scalabilité et résilience

## Inconvénients

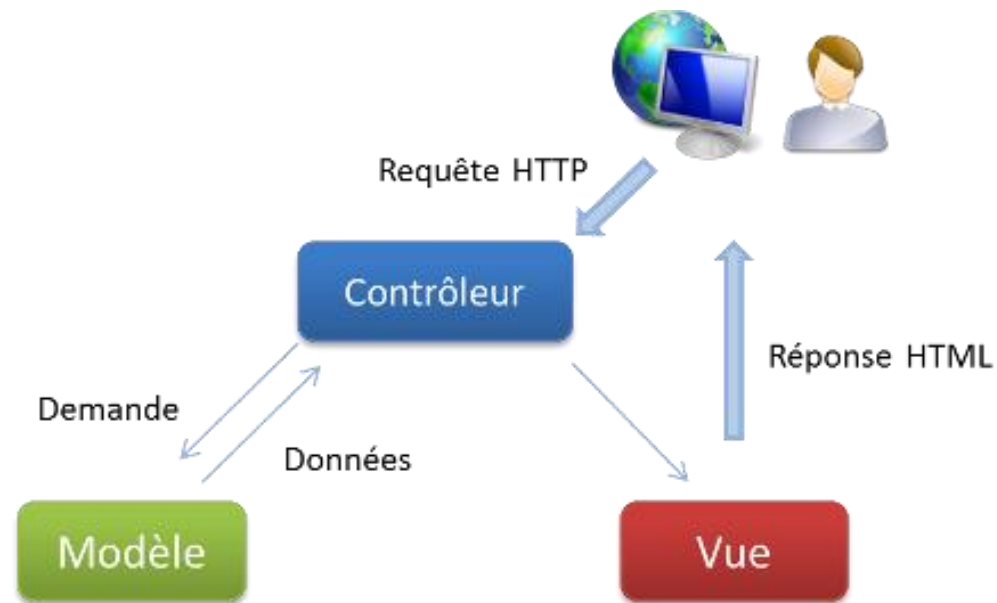
➤ Complexité de développement

➤ Complexité de gestion



# Pattern MVC

- Pattern MVC (Modèle, vue, Contrôleur): L'objectif de MVC est de structurer le code en séparant les responsabilités.



# Pattern MVC

---

## **Modèle :**

- La récupération des informations à partir d'une base de données
- Organisation des informations trouvées
- Utilisation des requêtes SQL

## **Contrôleur :**

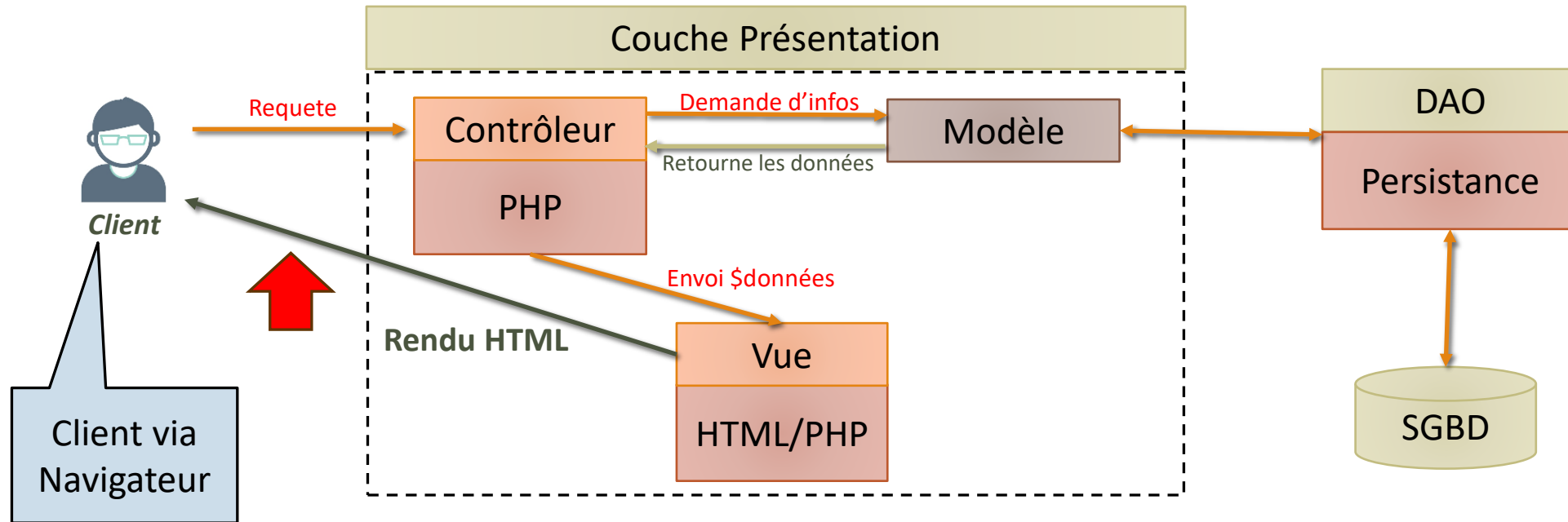
- Joue le rôle d'intermédiaire entre la vue et le model
- Il contient les traitements et prend la décision
- Il récupère les informations à partir du model
- Il traite les informations et décide de la redirection vers une vue appropriée

## **Vue :**

- Se limite à l'affichage des informations
- Utilisation du code HTML
- Récupération des données et de les envoyés au contrôleur



# Application web (rendu serveur)



# Application web (rendu client)

