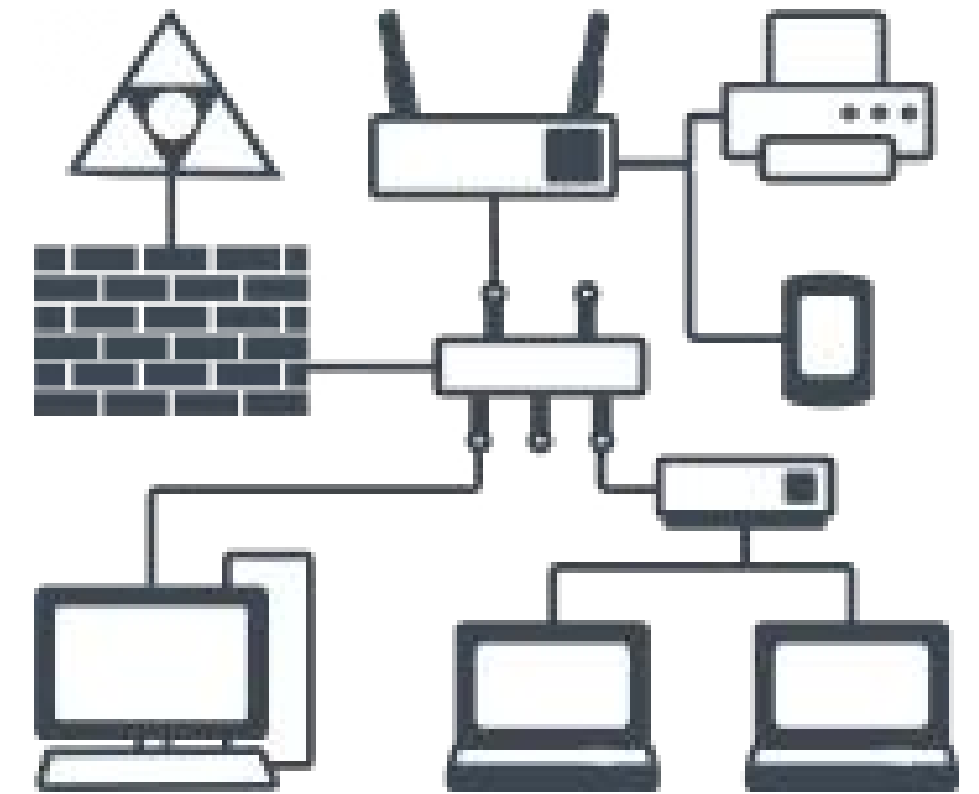
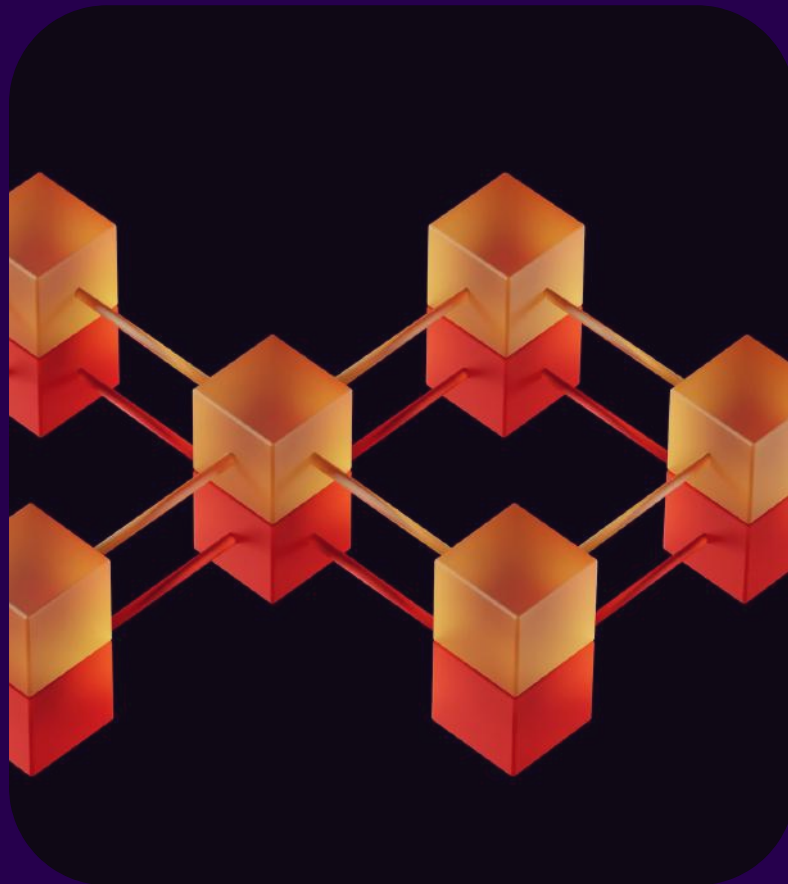


Architecture Client-Serveur



4 QUESTIONS :



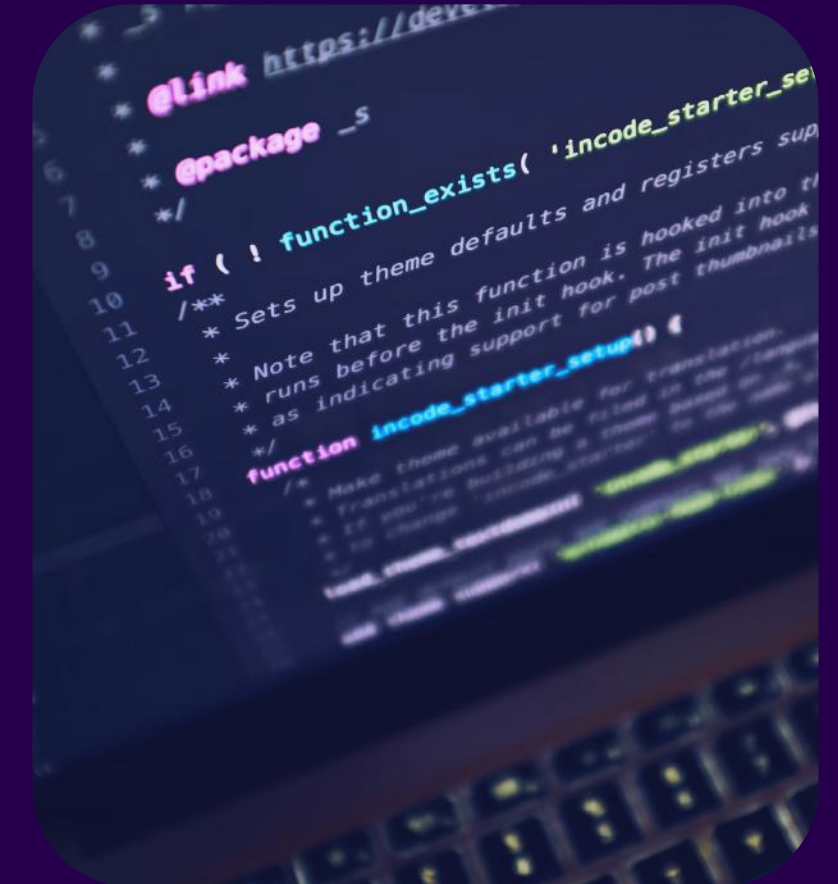
Architecture client-serveur
Importance



Protocole HTTP
Importance



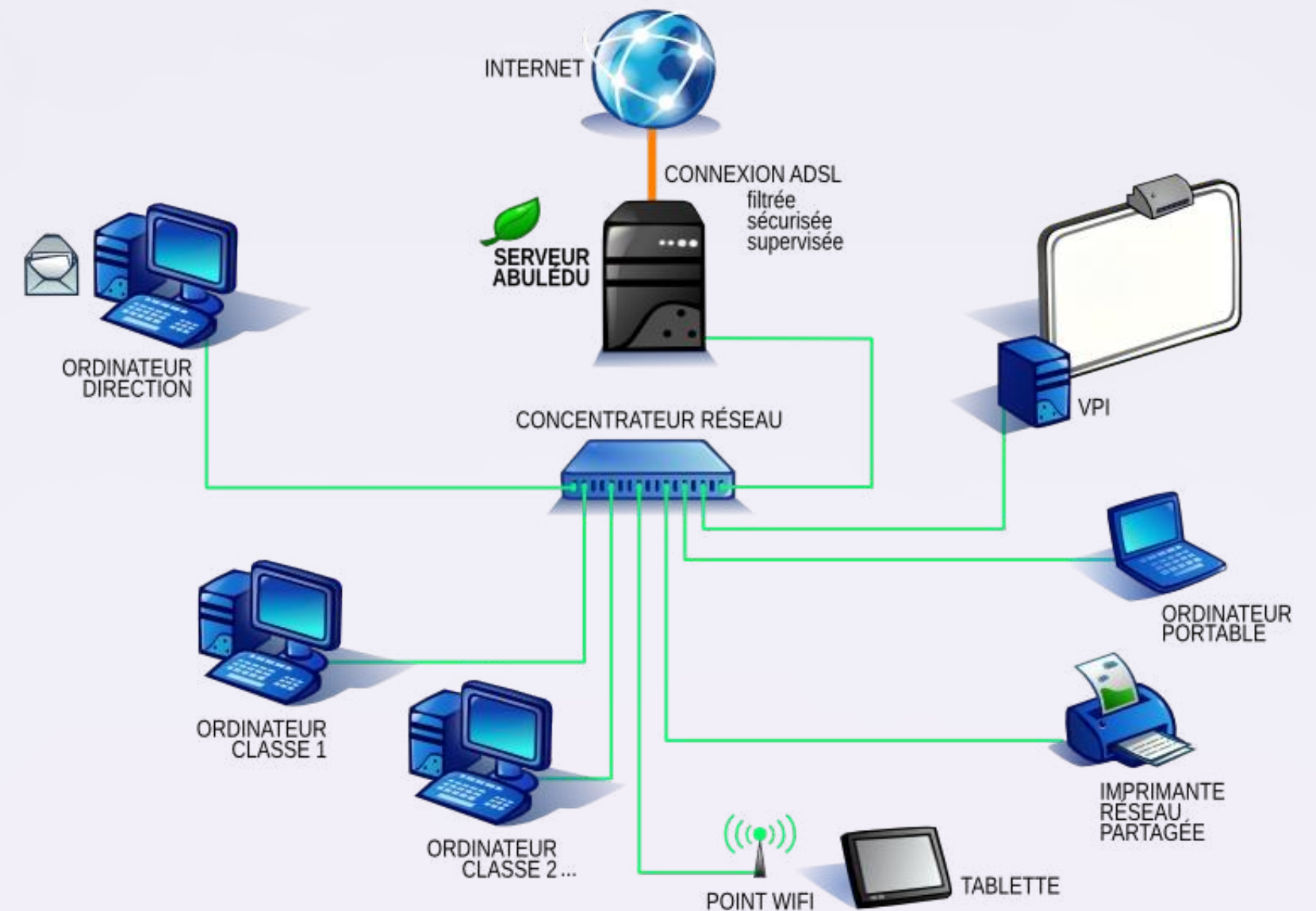
Les méthodes HTTP
Different



Codes Réponse HTTP
Categories

Introduction

L'architecture client-serveur est un modèle qui définit la **relation** et la **répartition** des tâches entre un client et un serveur. Elle permet un **traitement** efficace des données et une **extensibilité** en déchargeant certaines tâches vers le serveur, tandis que le client gère les interactions de l'utilisateur.



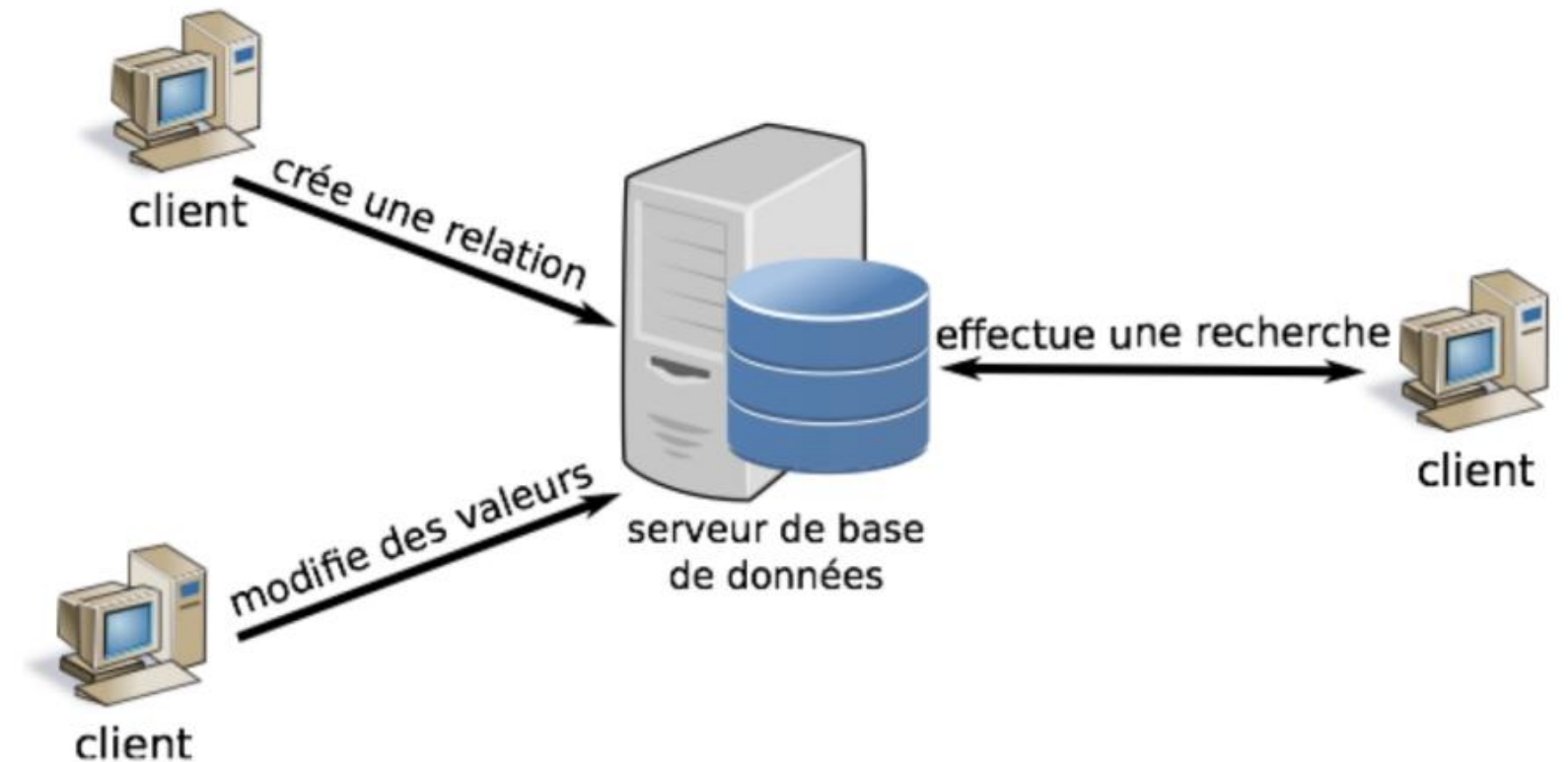
Architecture client serveur

Principe :

- Le client envoie des **requêtes**.
- Le serveur attend les requêtes et y **répond** par l'envoi de fichiers html css js ou Json.
- Les requêtes passent par le réseau ou en local

Différents types de serveurs :

1. **Serveur Web** publie pages Web demandée par navigateur web,
2. **Serveur de messagerie** électronique transmet des courriels à des clients de messagerie
3. **Serveur de fichiers**
4. **Serveur de bases de données**



Le client et le serveur doivent utiliser le même **protocole de transmission** des request / response (règles qui régissent les échanges de données).

→ **Protocole HTTP pour le web**

Protocole HTTP

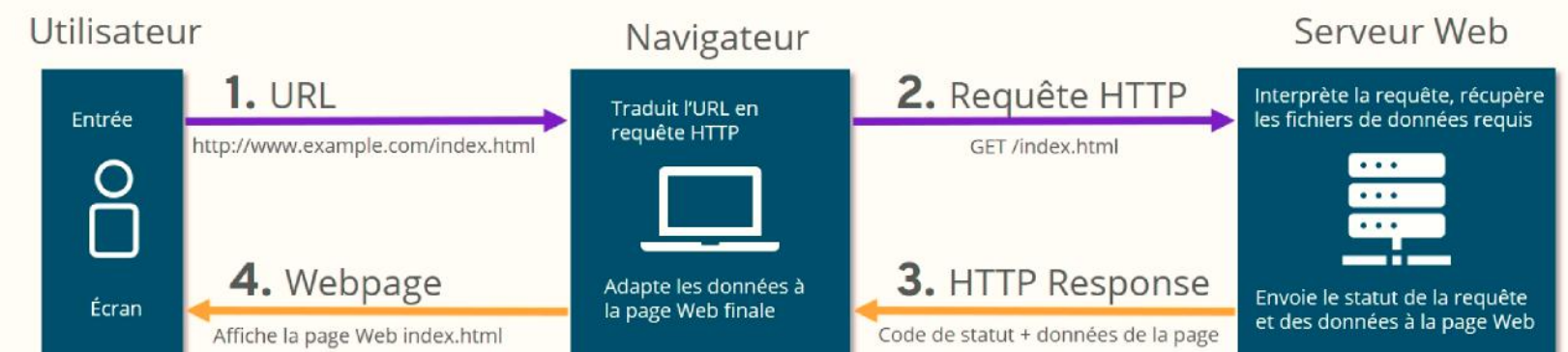
Principe :

Le protocole HTTP (Hypertext Transfer Protocol) est une règle de communication qui permet aux ordinateurs de se parler sur le web.

Voici quelques points importants :

1. **Requête-Réponse:** Quand vous ouvrez une page web, votre ordinateur demande au serveur de la montrer, et le serveur répond avec la page.
2. **Méthodes :** La demande peut demander différentes choses, comme montrer une page (GET) ou envoyer des informations (POST).
3. **URL:** L'adresse web que vous tapez dit au serveur quelle page vous voulez voir.
4. **Codes de Statut:** Après chaque demande, il y a un code qui dit si tout va bien.

Processus de communication HTTP



5. **Sécurité:** HTTPS garde les informations en sécurité.
6. **Mises à Niveau :** De temps en temps, ils améliorent la façon dont les ordinateurs parlent pour que tout soit plus rapide et plus sûr.

Différentes méthodes utilisées pour faire une requête HTTP

Méthodes HTTP (actions qui peuvent être réalisées sur une ressources Web):

- **GET** : demande des données au serveur sans les modifier
- **POST** : Soumet des données au serveur pour traitement (Ex: envoi d'un formulaire)
- **PUT** : Met à jour des données existantes sur le serveur.
- **DELETE** : Supprime des données.

Ces méthodes correspondent aux opérations sur la base de données **CRUD: Create / Read / Update / Delete**

Dans FLASK:

```
from flask import Flask, request
app = Flask(__name__)
@app.route('/example', methods=['GET', 'POST'])
def example():
    if request.method == 'GET':
        # Handle GET request
        return 'This is a GET request to pythongeeks'
    elif request.method == 'POST':
        # Handle POST request
        return 'This is a POST request to pythongeeks'
```

Dans HTML

```
<form method="POST" action="url">
```

Lien entre requête HTTP et code réponse

Les codes de réponse indiquent le résultat de la requête et permettent au client de comprendre si l'opération a réussi, échoué ou nécessite une action supplémentaire.
Contribue à la communication efficace entre le client et le serveur.

Codes de réponse HTTP :

- 1xx : Informations.
- 2xx : Succès.
- 3xx : Redirection.
- 4xx : Erreur client.
- 5xx : Erreur serveur.

Par exemple :

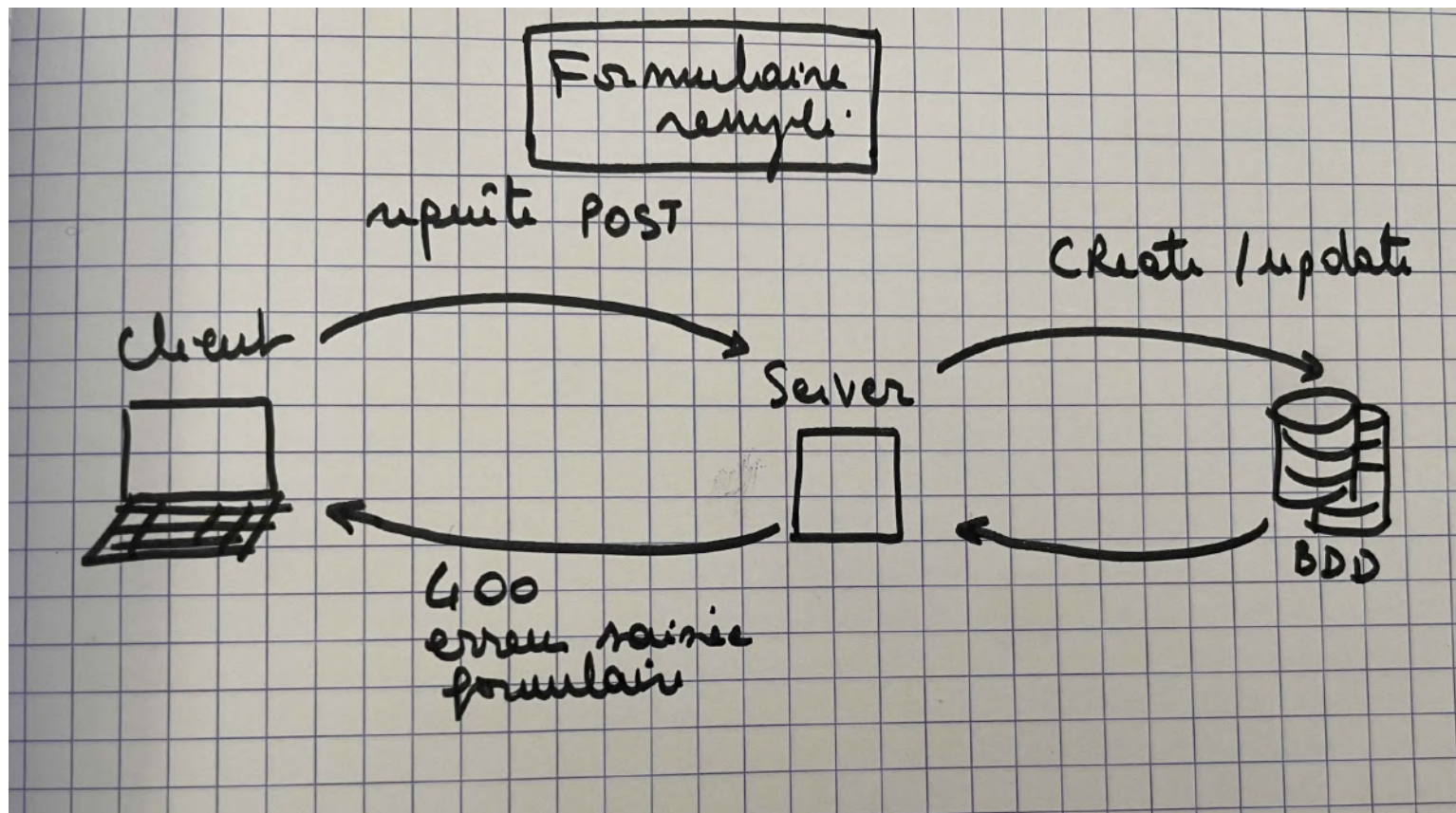
GET → 200 OK : La requête a réussi, et la réponse contient les données demandées.

GET → 404 Not Found : La ressource demandée n'a pas été trouvée.

POST → 201 Created : La requête a été traitée avec succès et une nouvelle ressource a été créée.

POST → 400 Bad Request : La requête est mal formée ou manque des informations nécessaires.

CLIENT, SERVEUR, ET BASE DE DONNEES

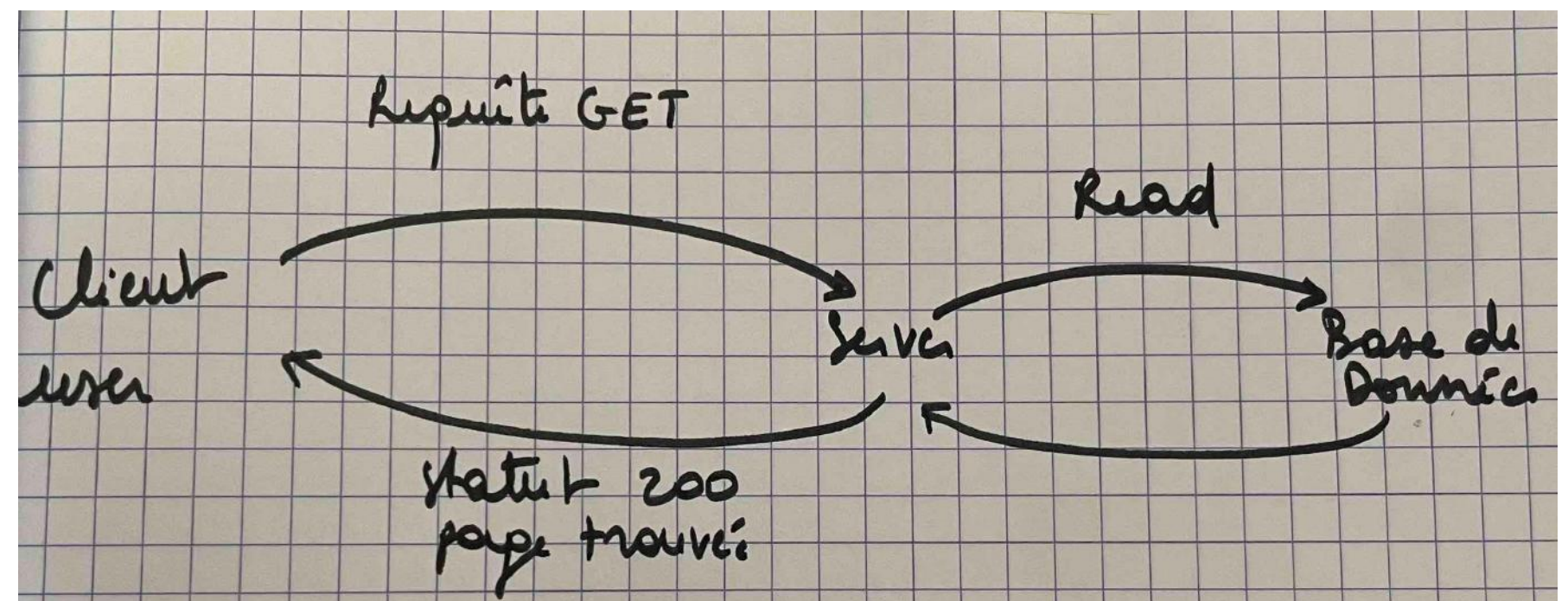


Envoi d'un formulaire / MAJ panier

Le client envoie une requete au serveur pour créer ou modifier une entrée dans la data base en passant par le serveur. Erreur de saisie: STATUT 400.

Recherche d'une page

Le client envoie une requête au serveur pour lire une donnée présente dans la data base en passant par le serveur. La donnée lui sera ensuite transmise par le serveur avec un code : STATUT 200 (page trouvée)



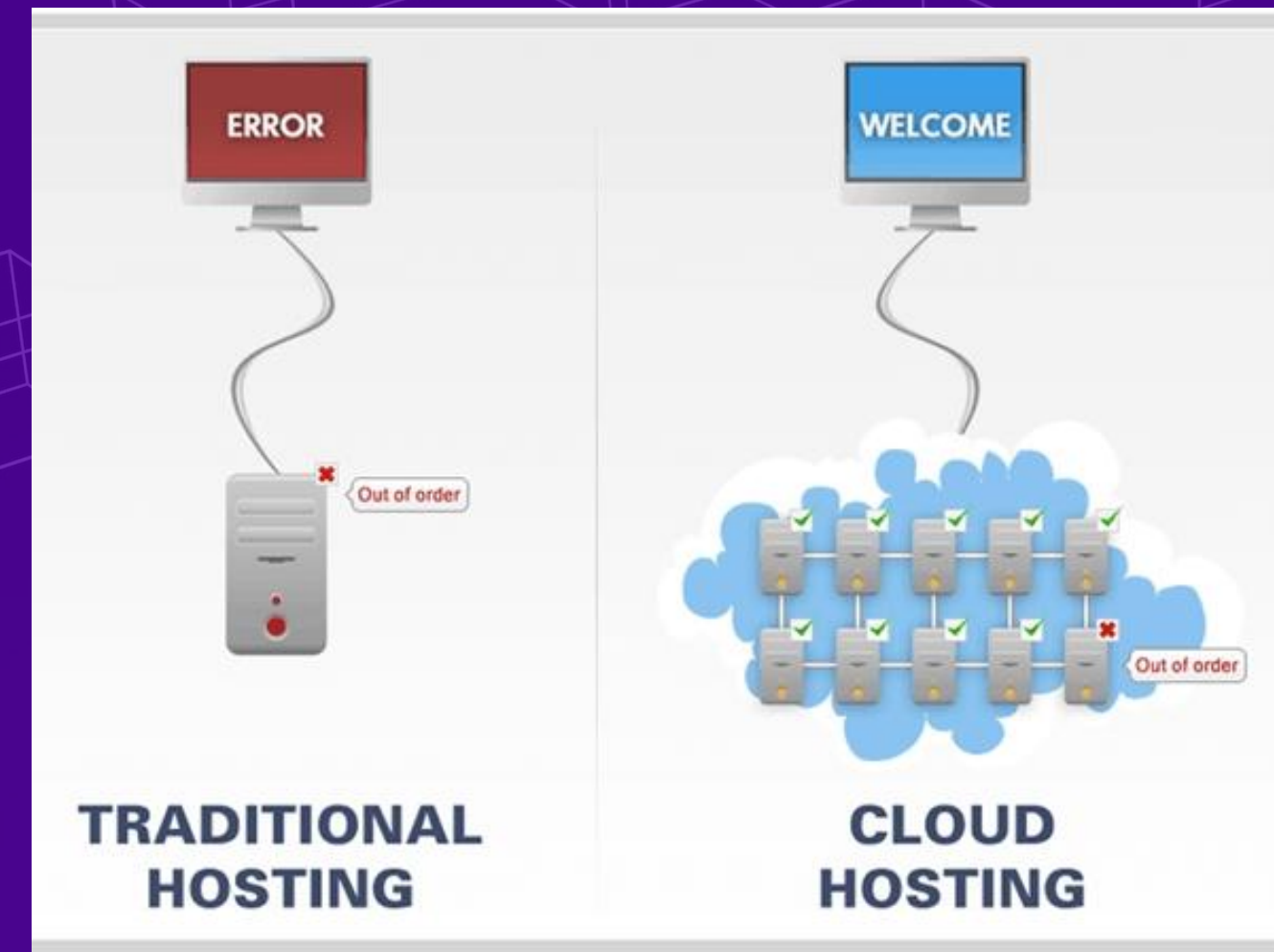
©By Professor Zumbiehl

Cloud Computing (Azure)

Virtualisation de ressources informatiques (hard disk, memory, processor) sur le cloud par le cloud provider Microsoft.

Caractéristiques principales :

- Resource pooling
- Elasticity
- Pay per use
- Automation



Cloud Computing (Azure)

Client-Serveur :

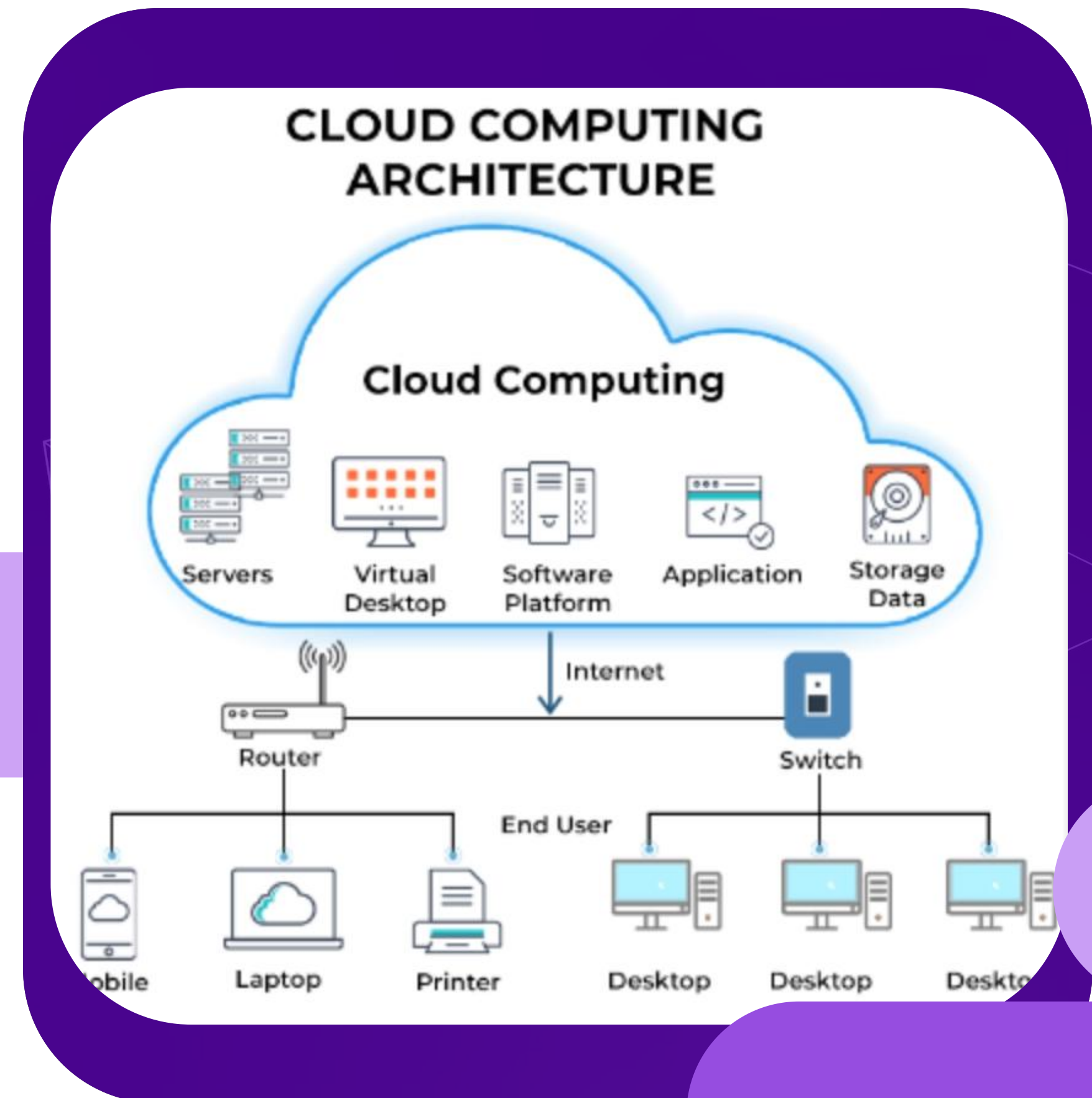
- **Répartition des Tâches** : Le client demande des services, et le serveur fournit ces services. Chacun a un rôle spécifique.
- **Interaction** : Le client envoie des demandes au serveur, qui répond en fournissant les services demandés.

Cloud Computing :

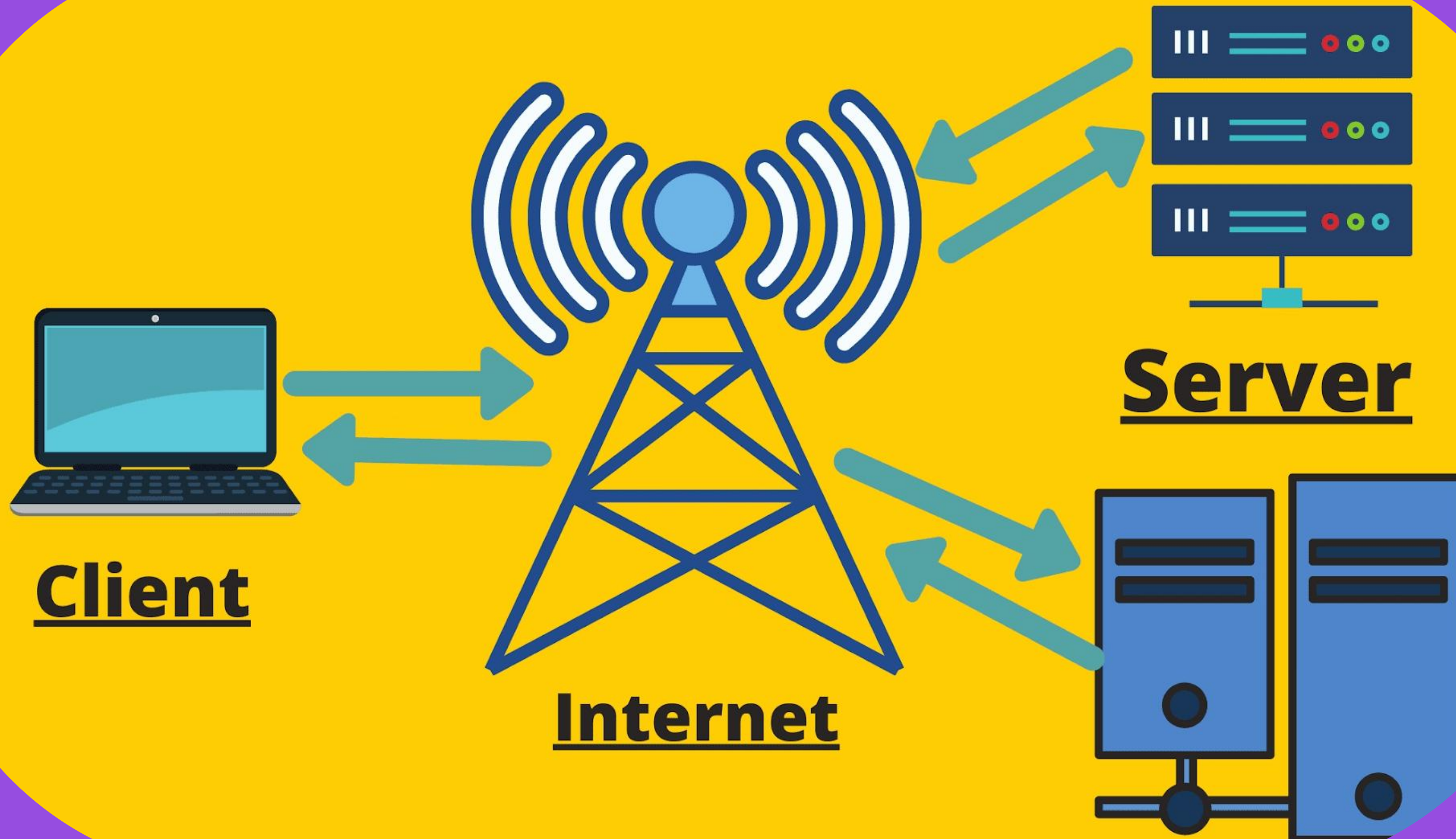
- **Accès à Distance** : Accès à des ressources informatiques (comme des serveurs, des bases de données) via Internet.
- **Fournisseurs de Services** : Des entreprises tiers gèrent ces ressources. Vous payez uniquement pour ce que vous utilisez.
- **Flexibilité** : Vous pouvez ajuster rapidement vos ressources selon vos besoins, sans avoir à posséder une infrastructure physique.

Différences :

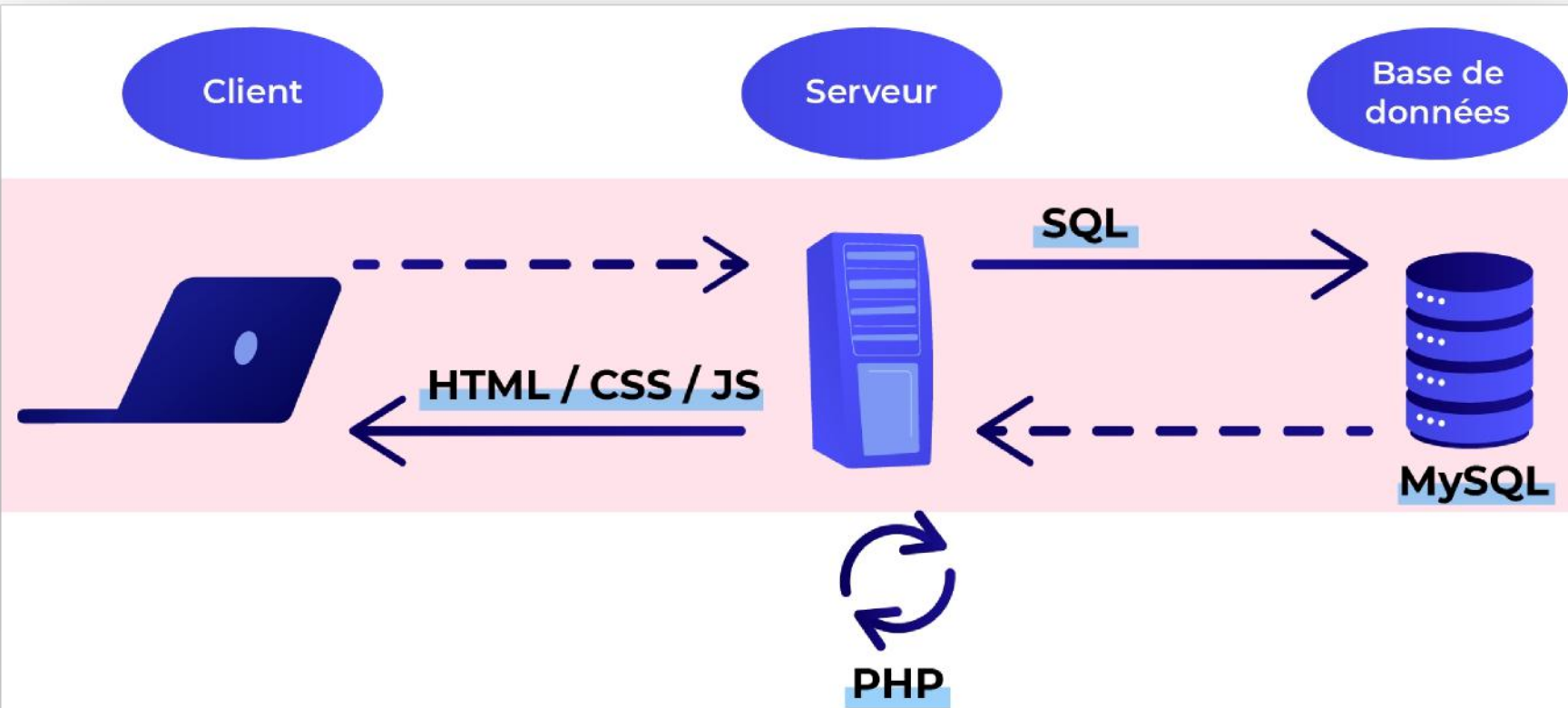
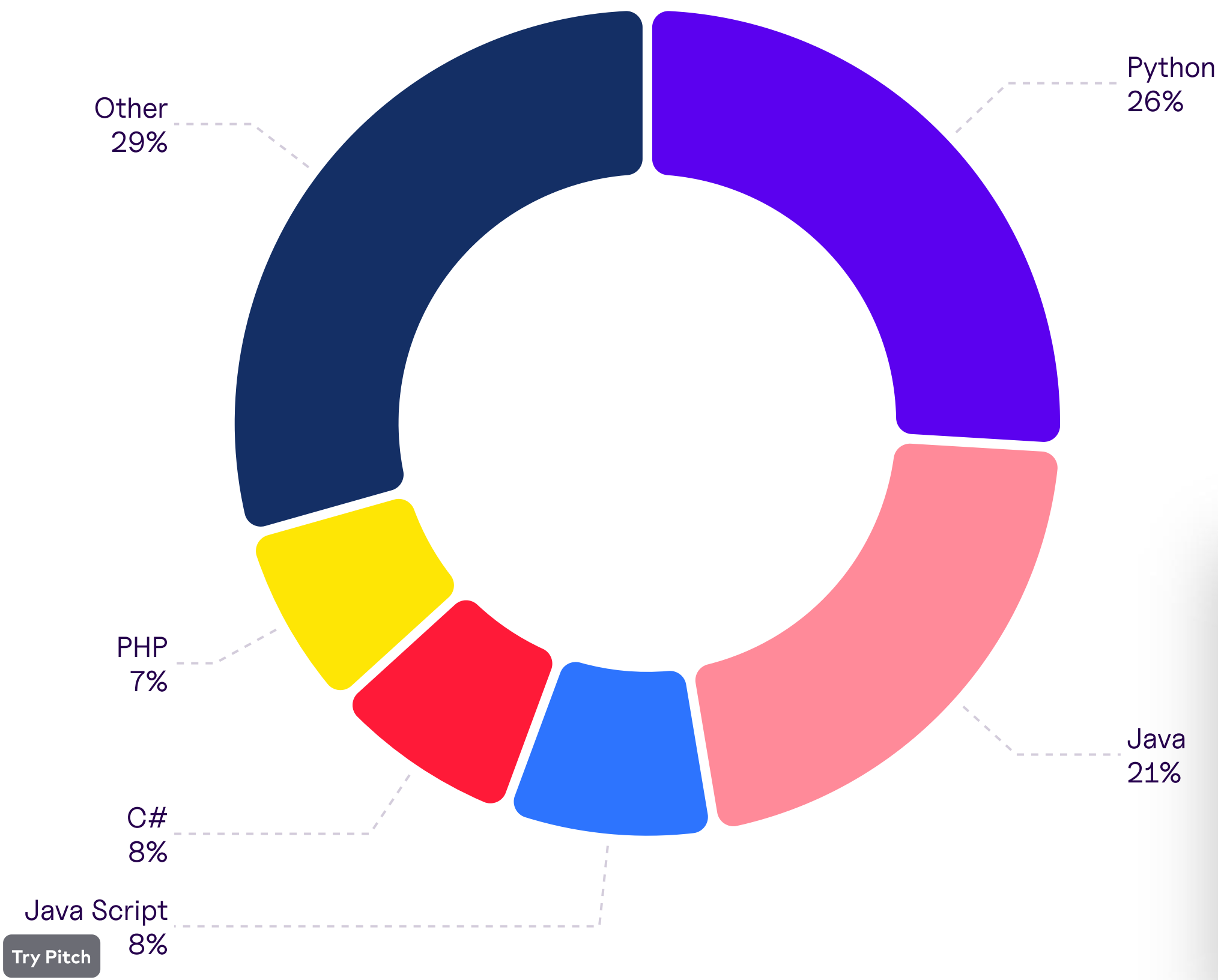
- **Nature** : Le client-serveur se concentre sur qui fait quoi, alors que le Cloud Computing est plus sur l'accès à distance à des ressources.
- **Infrastructure** : Le client-serveur peut utiliser une infrastructure locale, tandis que le Cloud utilise celle gérée par d'autres.
- **Gestion** : Dans le client-serveur, vous gérez généralement vos ressources, alors que dans le Cloud, c'est géré par quelqu'un d'autre.



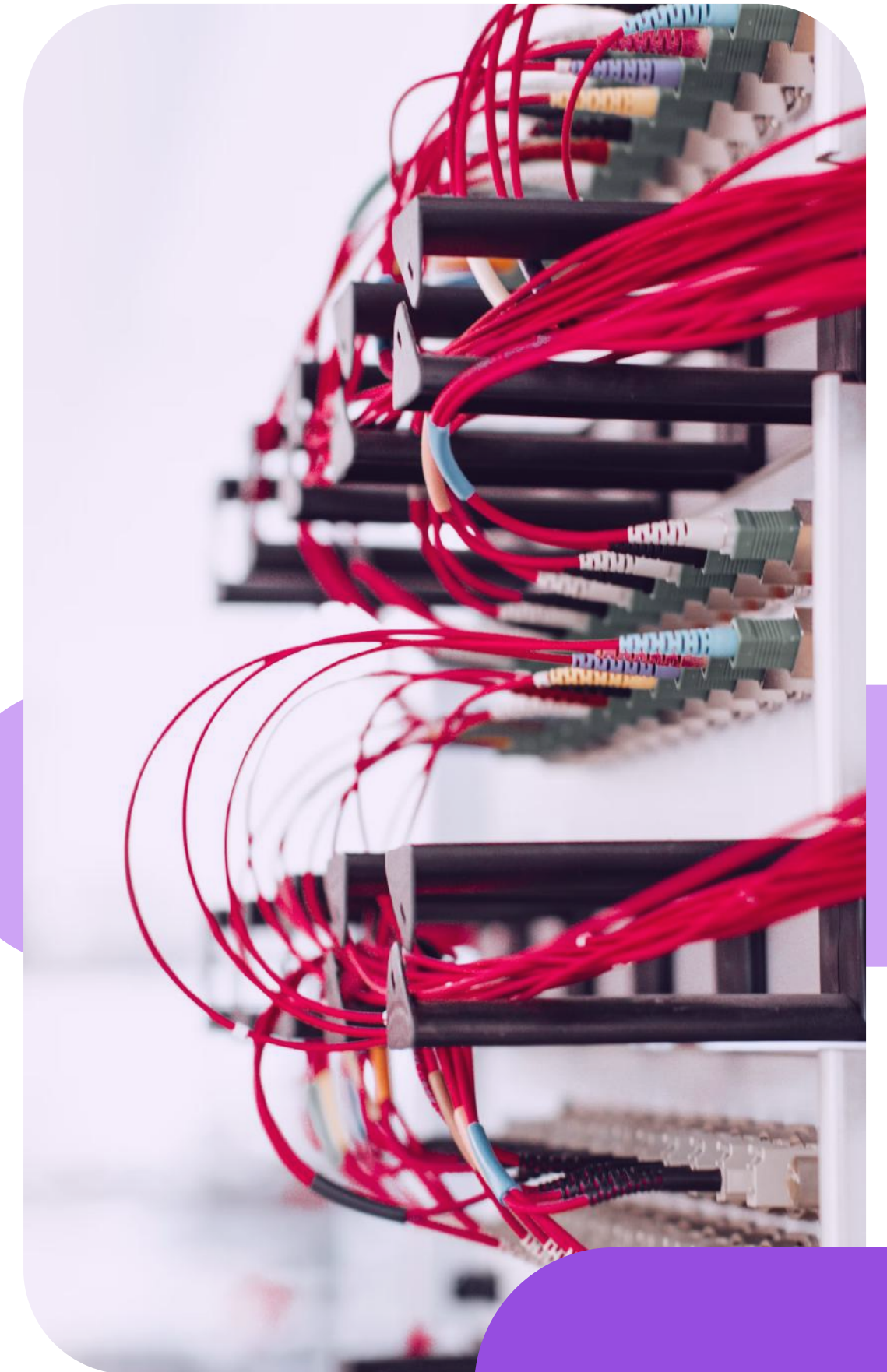
Internet : client et serveur



Langages WEB Les plus utilisés



Summary of client-server architecture and HTTP





Pitch

Want to make a presentation like this one?

Start with a fully customizable template, create a beautiful deck in minutes, then easily share it with anyone.

Create a presentation (It's free)

