CHAPITRE 1

Approfondir la notion Client/Serveur

1. **Principe Client/serveur**
2. Architectures 2-tiers
3. Types de serveurs web

PARTIE 1

Introduction

À partir du moment où les architectures matérielles peuvent être interconnectées, nous pourrons réaliser une **architecture client/serveur**.

L'architecture client-serveur s'appuie sur **un poste central, le serveur**, qui envoie des données aux machines clientes. Les programmes qui accèdent au serveur sont appelés **programmes clients** (client FTP, client mail, navigateur).

Cette architecture est basée sur l'utilisation de deux types de logiciels : **un logiciel serveur et un logiciel client** s'exécutant normalement sur 2 machines différentes.

L'élément important dans cette architecture est l'utilisation de mécanismes de communication entre deux applications.

Le logiciel client-serveur est indépendant des plateformes matérielles et logicielles dite hétérogénéité.

Le logiciel client-serveur masque aux clients la localisation du serveur.

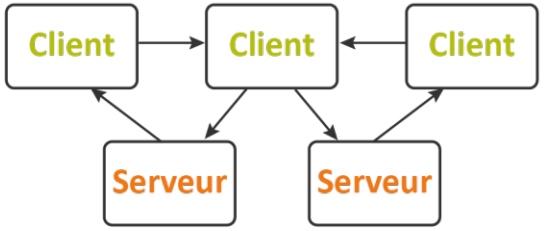
Les services internet sont conçus selon cette architecture.

Chaque application est composée de logiciel serveur et logiciel client.

Présentation de https://perso.univ-lyon1.fr/olivier.gluck :

Requête/Réponse

Fig. : Un client, un serveur



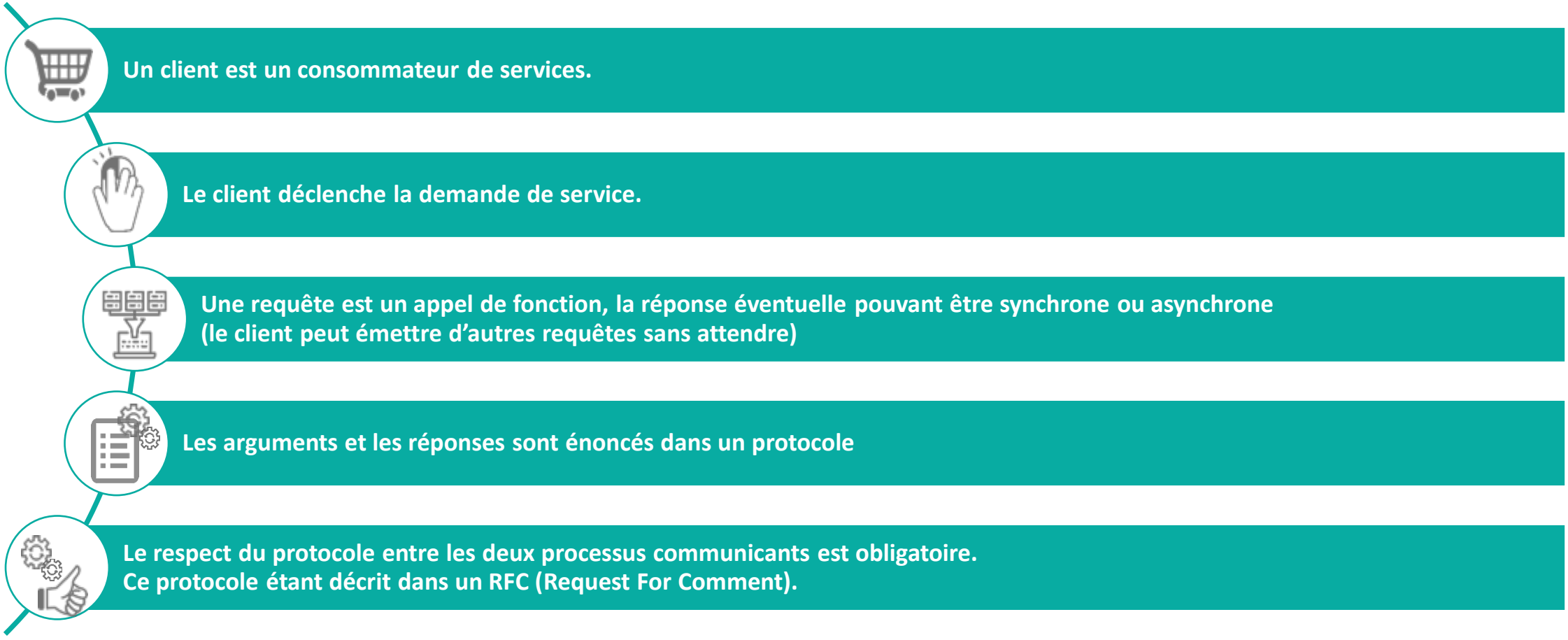
Le serveur traite plusieurs requêtes simultanées

Fig. : Plusieurs clients, un serveur

Le serveur contacté peut faire appel à un service sur un autre serveur (ex.SGBD)

Fig. : Un client, plusieurs serveurs

C’est quoi un Client ?



C’est quoi un Serveur ?

- Un serveur est un ordinateur ou un système qui met des ressources, des données, des services ou des logiciels à la disposition d’autres ordinateurs, qualifiés de « clients », sur un réseau.
- Les machines serveurs sont généralement dotées de capacités supérieures à celles des ordinateurs personnels en ce qui concerne la puissance de calcul, les entrées-sorties et les connexions réseau, afin de pouvoir répondre de manière efficace à un grand nombre de clients.
- Pour pouvoir offrir les services d’un serveur en permanence, le serveur doit être sur un site avec accès permanent sans interruption de connexion qui pourra interrompre la communication ainsi ses services. Dans le schéma par exemple c’est grâce à une connexion Internet.
- Pour un logiciel serveur, plusieurs logiciels clients sont utilisés dans des environnements différents : Unix, Mac, PC... par exemple.
- **Un serveur itératif :** désigne une implémentation qui traite une seule requête à la fois.
- **Un serveur parallèle :** fonctionne en mode concurrent, désigne une implémentation capable de gérer plusieurs tâches en apparence simultanées.
- **Service avec état :** le serveur conserve localement un état pour chacun des clients connectés : informations sur le client, les requêtes précédentes, …
- **Service sans état :** le serveur ne conserve aucune information sur l'enchaînement des requêtes…

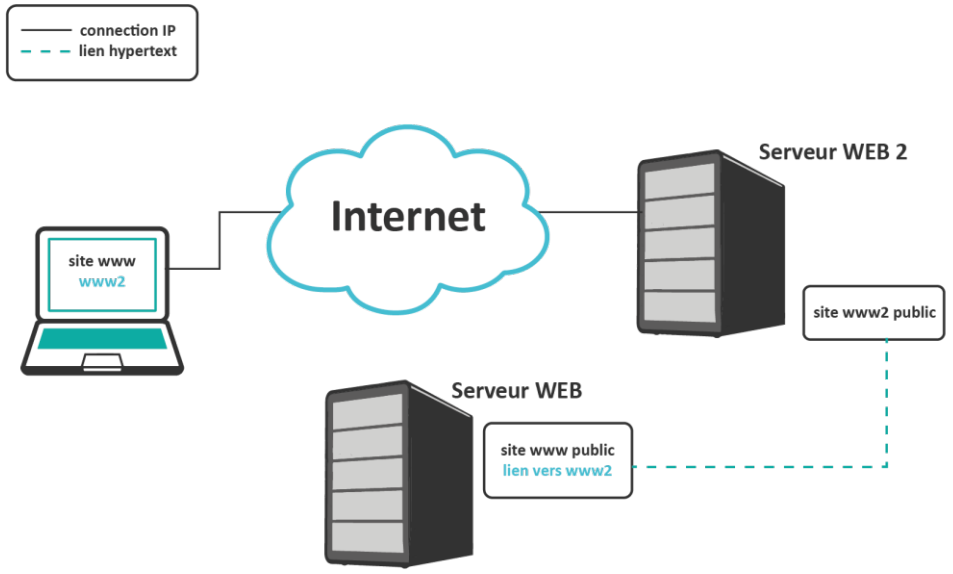


Fig : Internet connexion client serveur. Source: https://www.ipgp.fr/

PARTIE 1

PARTIE 1

C'est quoi un Middleware ?

- La liaison entre le client et le serveur qui se charge de toutes les communications entre les processus est appelée Middleware.
- Un middleware est un logiciel médiateur ou intergiciel qui se charge de la liaison.

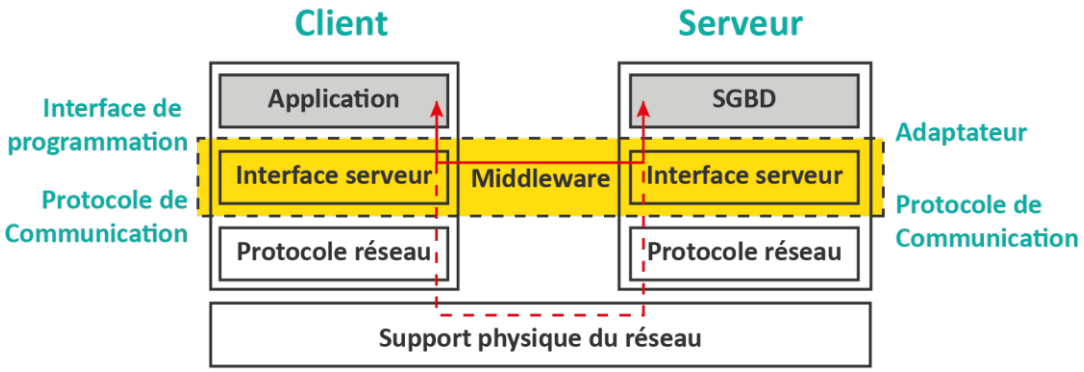


Fig : Représentation Middleware selon Gartner Group

Les fonctions d'un Middleware

- Procédure d'établissement de connexion
- Exécution des requêtes
- Récupération des résultats
- Procédure de fermeture de connexion
- Initiation des processus sur différents sites
- Services de répertoire (nommage)
- Accès aux données à distance
- Gestion des accès concurrents
- Sécurité et intégrité
- Monitoring
- Terminaison des processus
- Mise en cache des résultats et des requêtes

Les services d'un Middleware

- Conversion
- Adressage
- Sécurité
- Communication

Avantages et Inconvénients

AVANTAGES	INCONVÉNIENTS
• Des ressources centralisées, tout en garantissant la cohérence des données. • Une meilleure sécurité, puisque le nombre de points d'entrée permettant l'accès aux données est connus. • Une administration au niveau serveur. • Un réseau évolutif, on peut supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modifications majeures. • Une interface utilisateur riche. • Une appropriation des applications par l'utilisateur. • Une notion d'interopérabilité.	• Déploiement coûteux : un coût élevé dû à la technicité du serveur • Trafic réseau important : tout le réseau est architecturé autour du serveur • Déploiement difficile : le poste client doit constamment être mis à jour pour répondre au besoin

Fonctionnement

Fonctionnement d'un réseau client /serveur :

- Le client, pour recevoir des informations du serveur, lui émet une requête passant par un port du PC (exemple : port 25 pour les mails, port 80 pour le web et port 21 pour le FTP)
- Le serveur envoie ensuite les informations grâce à l'adresse IP de la machine cliente
- Le client traite et affiche les informations en provenance du serveur

La communication peut être établie de deux manières entre deux appareils ou plus, qui sont orientés connexion (mode connecté) et sans connexion (mode non connecté).

Le mode non connecté ne garantit pas :

- l'intégrité des données
- l'ordonnancement des données
- la non-duplication des données

Le mode connecté garantit les propriétés ci-dessus qui ne sont pas garanties par l'application dans le mode non connecté. D'autant plus, le mode connecté :

- Implique une diminution des performances brutes par rapport au mode non-connecté
- Peut constituer une contrainte
- Permet une implémentation asynchrone des échanges

Types d'architectures

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.

L'architecture 1-tiers : Les trois couches sont intimement liées et s'exécutent sur la même machine. On parle donc de l'informatique centralisée.

L'architecture 2-tiers encore appelée client-serveur de première génération ou client-serveur de données, le poste client se contente de déléguer la gestion des données à un service spécialisé.

L'architecture 3-tiers est l'extension du modèle client/serveur. Ce type d'architecture est le plus courant des architectures multi-tiers. Il est également appelé client-serveur de deuxième génération ou client-serveur distribué.

L'architecture n-tiers est une généralisation de l'architecture 3-tiers à partir d'un modèle de composants. Les liens entre les composants sont rendus possibles par l'existence d'un bus logiciel.

D'après la source http://www-igm.univ-mlv.fr/~dr/XPOSE2001/perrot/Intro-Comparatif.htm on dispose d'un tableau comparatif entre l'architecture 2-tiers et 3 et n-tiers, qui nous permet de recenser les avantages et les limitations de chacun des processus (voir tableau).

	2-tiers	3 et n tiers
Administration du système	Complexe (la couche application est physiquement répartie sur plusieurs postes clients)	Moins complexe (les applications peuvent être gérées centralement sur le serveur)
Sécurité	Faible (sécurité au niveau des données)	Elevée (raffinée au niveau des services ou des méthodes)
Encapsulation des données	Faible (les tables de données sont directement accessibles)	Elevée (le client fait appel a des services ou méthodes)
Performance	Faible (plusieurs requêtes SQL sont transmise sur le réseaux, les données sélectionnées doivent êtres acheminées vers le client pour analyse)	Bonne (seulement les appels de service et les réponse sont mise sur le réseau)
Extensibilité	Faible (gestion limitée des liens réseaux avec le clients)	Excellente (possibilité de répartir dynamiquement la charge sur plusieurs serveurs)
Réutilisation	Faible (application monolithique sur le client)	Excellente (réutilisation des services et des objets)
Facilité de développement	Elevée	En progression (des outils intégrés pour développer la partie du client et du serveur)

Tab : Comparatif architecture 2-tiers Vs 3 et n-tiers

	2-tiers	3 et n tiers
Lien Serveur-serveur	Non	Oui (via le middleware Serveur/Serveur)
Intégration des systèmes déjà en place	Non	Oui (via des passerelles encapsulées par les services ou objets)
Soutien Internet	Faible (les limitation de la bande passante pénalisent le téléchargement d'application de type « fat-client »)	Excellente (les applications de type « thin-client »sont facilement téléchargeable et les appels aux services repartissent la charge sur un ou plusieurs serveurs)
Sources de données hétérogènes	Non	Oui (les applications 3-tier peuvent utiliser plusieurs bases de données dans la même transaction)
Choix de communication de type « riche »	Non (synchrone et RPC)	Oui (gestion asynchrone de message, files de livraison, publication et abonnement, « broadcast »
Flexibilité d'architecture matériel	Limitée	Excellente (possibilité de faire résider les couches 2 et 3 sur une ou plusieurs machines)
Relève en cas de pannes	Faible	Excellente (possibilité d'avoir la couche du centre « middle-tier » sur plusieurs serveurs)

Tab : Comparatif architecture 2-tiers Vs 3 et n-tiers

CHAPITRE 1

Approfondir la notion Client/Serveur

1. Principe Client/serveur
2. **Architectures 2-tiers**
3. Types de serveurs web

Introduction

- L'architecture 2-tiers (aussi appelée client-serveur de première génération, ou encore client-serveur de données) **caractérise les systèmes clients/serveurs pour lesquels le client à un besoin et le serveur le lui fournit directement**, en utilisant ses propres ressources.
- Le serveur traite la demande du client **sans recours à des applications tierces**.
- Ce type d'architecture permet de bénéficier pleinement de la puissance des ordinateurs déployés en réseau pour **fournir à l'utilisateur une interface riche**, tout en garantissant la cohérence des données, qui reste gérée de façon centralisée.
- La gestion des données est **prise en charge par un SGBD** (Système de gestion de base de données) centralisé, s'exécutant le plus souvent sur un serveur dédié.
- **Le langage de requête SQL** (Structured Query Language) est la méthode la plus utilisée pour consulter la base de données.
- Le dialogue entre client et serveur se résume donc à l'envoi de requêtes et au retour des données correspondant aux requêtes.

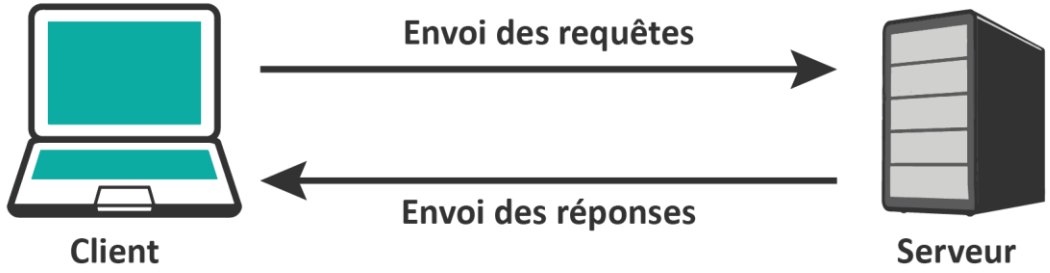


Fig. : Architecture 2 tier

C’est quoi un client ?

Le client demande un service au serveur.

Un client lourd est une application où les traitements sont principalement effectués sur la machine locale dite cliente.

Caractéristiques d'un client :

- Il est actif.
- Il envoie des requêtes au serveur.
- il attend et reçoit les réponses du serveur.

C’est quoi un serveur ?

Le serveur exécute la requête et renvoie le résultat au client.

Les performances du serveur sont atteintes rapidement après la sollicitation d’un nombre important de clients.

Caractéristiques d'un serveur :

- Il est passif. Il est à l'écoute, prêt à répondre aux requêtes envoyées par les clients.
- Il traite la requête du client dès qu‘il la reçoit.
- Il envoie une réponse au client après le traitement de la requête.

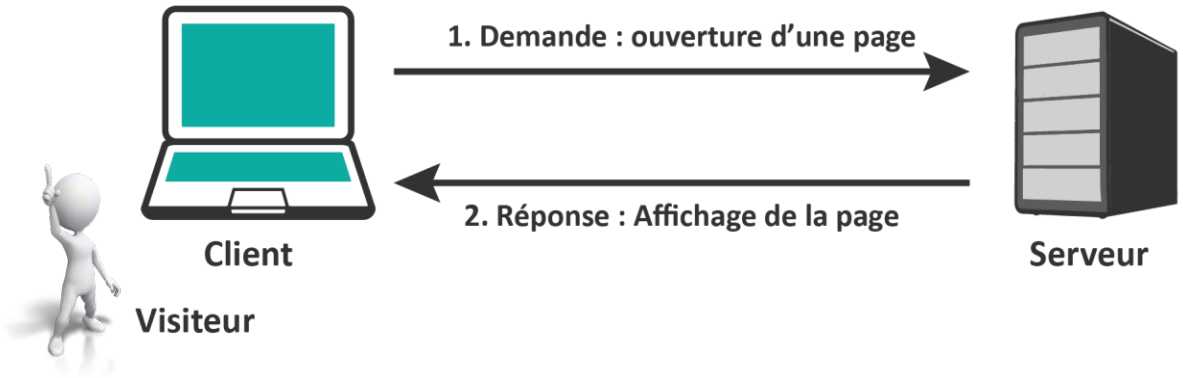


Fig. : Exemple du mécanisme de requête

Niveau d'abstraction

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

Couche Présentation (Couche IHM : Interface Homme-Machine)

- Elle permet l'interaction d'application avec l'utilisateur.
- Elle contrôle en effet les saisies au clavier et à la souris ainsi que la présentation à l'écran.
- Elle doit être conviviale et ergonomique

Couche Traitement (Logique Applicative ou couche métier)

- Elle décrit les traitements à exécuter par l'application afin de répondre aux requêtes clients.
- Les traitements locaux : tiennent compte les contrôle effectués au niveau du dialogue avec l'IHM (formulaires, champs, boutons,...).
- Les traitements globaux : représentent les règles de l'application appelées aussi logique métier (Business Logique).

Couche Données (couche Persistance)

- Elle 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 (Définition de données, Manipulation de données, Sécurité de données, gestion des transactions, …).

- Gartner Group a proposé un découpage en six vues distinctes montrant les différentes possibilités de répartition entre clients et serveur des trois couches logicielles.

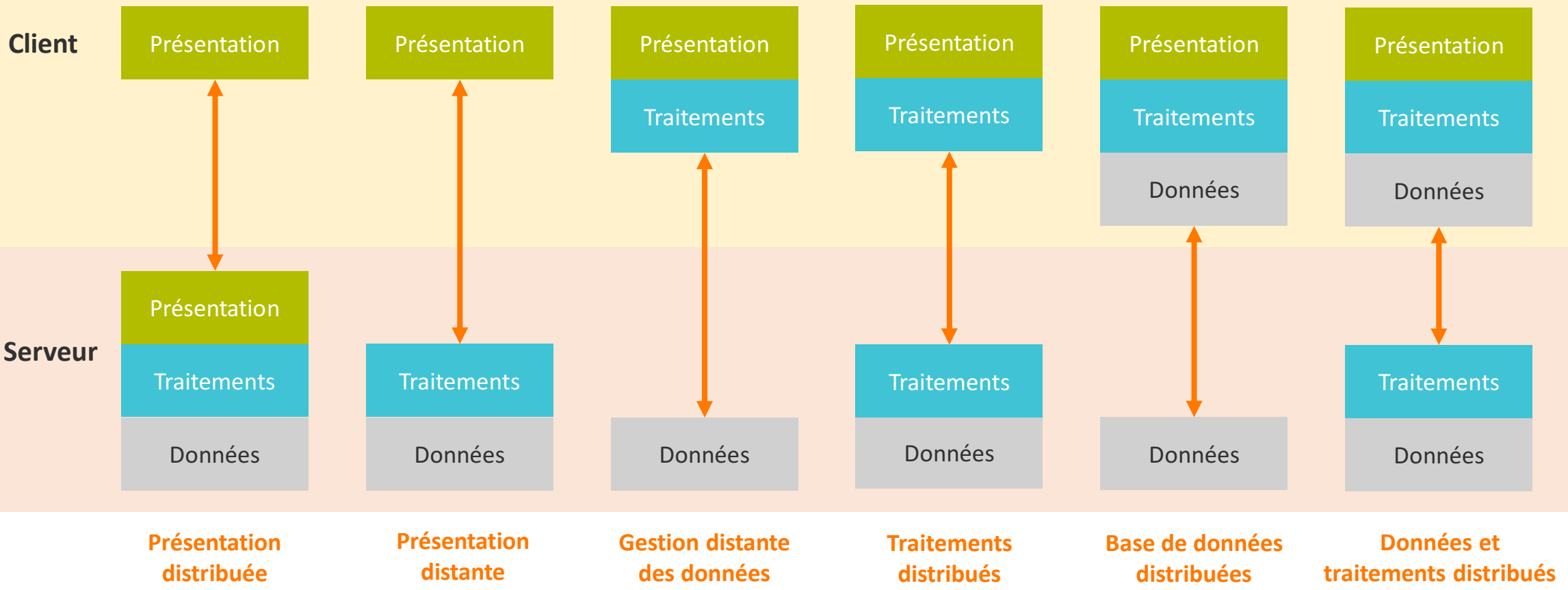


Fig : Modèle de Gartner pour les systèmes à deux niveaux (2-tiers)

CHAPITRE 1

Approfondir la notion Client/Serveur

1. Principe Client/serveur
2. Architectures 2-tiers
3. **Types de serveurs web**

Définition

Un serveur web

- Un serveur Web (aussi appelé serveur http), est tout type de serveur qui permet de diffuser des contenus Web sur Internet ou Intranet. C'est un service logiciel utilisé pour communiquer entre deux appareils sur un réseau.
- Un serveur web sert à rendre accessible des pages web sur internet via le protocole HTTP.
- Un serveur web répond par défaut sur le port 80.
- Pour qu'un site Web soit accessible à tout moment, le serveur Web sur lequel il est hébergé doit être connecté à Internet en permanence
- Un serveur Web en architecture 3 tiers est composé d'un système d'exploitation, un serveur HTTP, un langage serveur et un système de gestion de base de données (SGBD), cela constituant une plate-forme.

Définition

Un serveur web statique

- (aussi appelé une pile) est composé d'un ordinateur (matériel) et d'un serveur HTTP (logiciel).
- Il est appelé « statique » car le serveur envoie les fichiers hébergés « tels quels » vers le navigateur.

Un serveur web dynamique

- Possède d'autres composants logiciels, certains qu'on retrouve fréquemment dont un serveur d'applications et une base de données.
- Il est appelé « dynamique » car le serveur d'applications met à jour les fichiers hébergés avant de les envoyer au navigateur via HTTP.

PARTIE 1

Qui sont les clients ?

Les navigateurs web jouent le rôle de clients :

Brave

- Respect et protection de la vie privée
- Interface et ergonomie soignée
- Le plus rapide du marché

Chrome

- Très bonnes performances
- Simple et agréable à utiliser
- Un navigateur bien sécurisé

Firefox

- Un des plus innovants du marché
- Fonctionnalités pour optimiser l'UX
- Gère les derniers standards vidéo AV1

Edge

- Performances correctes
- Plus léger, rapide et moderne
- L'intégration à l'écosystème Windows/Microsoft...

Opéra

- Stable et performant
- Débits rapides
- Sécurité accrue avec outils de chiffrement efficaces

Vivaldi

- Bonne alternative aux navigateurs du marché
- Niveau de personnalisation
- Fonctionnalités intuitives et originales

UC Browser

Next

Safari

Maxthon

UR Browser

Qui est le Serveur ?

- **Le serveur :** machine qui exécute les requêtes et envoie les réponses. Liste des serveurs web :

- **AOLserver**
- **Apache HTTP Server**
- **Boa Web Server**
- **Caudium**
- **Cherokee Web Server**
- **HTTP File Server (HFS)**
- **Hiawatha Web Server**
- **IBM HTTP Server**
- **Internet Information Services (IIS)**
- **Jetty**
- **Lighttpd**
- **LiteSpeed Web Server**
- **Mongrel**
- **NaviServer**
- **NCSA HTTPd**
- **Nginx**
- **Node js**
- **OpenLink Virtuoso**
- **Oracle HTTP Server**
- **Oracle iPlanet Web Server Perlbal**
- **Thttpd**
- **Apache Tomcat**
- **TUX web server**
- **Wakanda Server**
- **Xitami**
- **Yaws**
- **Zeus Web Server**
- **…**

Le protocole spécifie comment le client et le serveur communiquent.

- **Protocole utilisé :** HyperText Transfer Protocol (HTTP)

Les demandes contiennent l'Uniform Ressource Locator (URL) de la page à afficher. L'URL est l'extension de la notion de nom de fichier sur un réseau

- **Syntaxe d'une URL :** protocole://adresse/fichier (exemple : https://www.ofppt.ma/fr/nos-formations)
  - **Protocole :** HTTP, FTP, news,…
  - **Adresse :** Spécifique au protocole et en général @ IP du serveur HTTP (ou le nom, qui sera résolu par appel au serveur DNS)
  - **Fichier :** nom du fichier à récupérer, facultatif (si omis, page par défaut : index.htm / index.html)

PARTIE 1

Qui est le Serveur ?

Une fois que le serveur a reçu la requête, et si celle-ci est valide, il effectue le traitement.

Les réponses contiennent des documents au format HyperText Markup Langage

Les types MIME (Multipurpose Internet Mail Extensions) permettent de préciser le type de documents transmis lors d'une communication. Repris dans HTTP/1.0 pour que les clients aient connaissance du type de document qui leur est renvoyé

Le serveur envoie la réponse :

- une ligne de statut (version du protocole)
- Un code de succès (ou d'échec)
- Informations (relatives au serveur et au message)
- Corps du message

Les Codes de retour permettent d'informer sur le succès ou l'échec de la requête :

- Information : 100 et 101 (demande de la suite de la requête, changement de protocole)
- Succès : 200 à 206
- Redirection : 300 à 305 + 307
- Erreurs du client : 400 à 417
- Erreurs du serveur : 500 à 505