OUTILS INTERNET II

Présenter par Mme Bensalem Hana

Université de Badji-Mokhtar Annaba

Promotion: MasterI ,GADM

Département d'informatique

CONTENU DU COURS

- **CHAPITRE1 : notions de bases**
- **CHAPITRE 2 : les documents sur le web**
- **Chapitre 3 :PHP les bases**
- **Chapitre4 :PHP et communication**
- **Chapitre5 : Base de donnée &MySql(à inserrer dans le précedent,à la place mêttre des outils de machine learning et deepllearning tel que orange…avec exemples)**
- **Chapitre 6 : ETUDE DE CAS(à supprimer, donner des tp dés la fin du 3 iéme chap)**

DES NOTIONS DE BA

Les **protocoles** régissent la manière dont les **données** sont **envoyées**, **reçues** et **traitées** sur Internet.

- **Internet :** INTERconnected NETworks
  - réseau mondial de réseaux d'ordinateurs né à la fin des années 1970
  - utilise le protocole de communication TCP/IP
    - **T**ransmission **C**ontrol **P**rotocol /**I**nternet Protocol
  - initiés par des organisations comme ARPA aux États-Unis.[1]

« Internet est une araignée mondiale de réseaux d'ordinateurs, interconnectés de manière à échanger librement des informations. » **– Tim Berners-Lee**

DES NOTIONS DE BASES (2)

- **L'hypertexte:** est une représentation de l'information sous forme d'un réseau de liens.
  - à consulter de manière **non linéaire**.
  - Le concept introduit en 1945 par **Vannevar Bush**
  - utilisé pour la première fois par **théodore Nelson** 20 ans après.
- En 1989,apparait **WWW** le concept d'un système hypertexte distribué au niveau mondial.
- créer par **Tim Berners-Lee** au **CERN** à Genève (l'Organisation européenne pour la recherche nucléaire).
- la navigation réaliser dans ce système est réalisé par un **navigateur**.

DES NOTIONS DE BASES (3)

- Le **Web**, ou **World Wide Web**, est l'un des services accessibles via Internet. Contrairement à Internet.
- le **Web** est un sous-ensemble d'**Internet.**
- Par exemple:
  - l'envoi et la réception d'e-mails,
  - le téléchargement de fichiers via FTP
  - ou l'utilisation d'applications mobiles se font sur **Internet**, mais pas nécessairement sur le **Web**.

DES NOTIONS DE BASES (4)

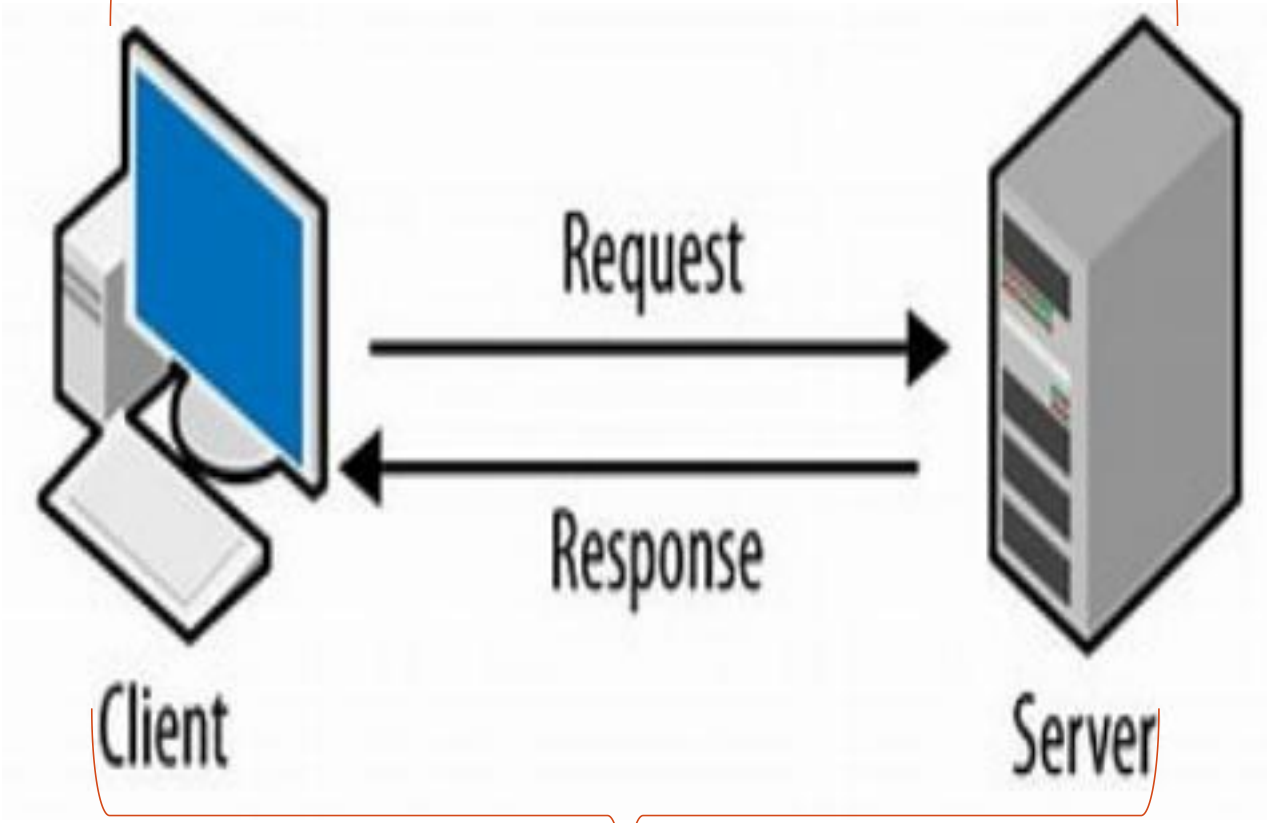
- l'ergonomie du Web n'a cessé d'évoluer [2],
  - Web 1.0 (1990) : mise en ligne de données et transfert d'informations descendantes, du communicant vers son récepteur ;
  - Web 2.0 (2000) : Web interactif avec l'apparition des réseaux sociaux (Twitter, Facebook…), des sites de partage (YouTube…) et de vente en ligne. L'internaute devient acteur;
  - Web 3.0 (2010) : apparition des données de masse et des applications intégrées ; avènement de l'Internet mobile.
  - Web 4.0 (≈ 2020) : l'omniprésence de l'intelligence artificielle, de l'apprentissage automatique (machine learning) et de **l'Internet des objets** (**IoT**), Des internautes sont rémunérés pour leur collaboration, avec de nouveaux services sans intermédiaires (blockchains…)

DES NOTIONS DE BASES (5)

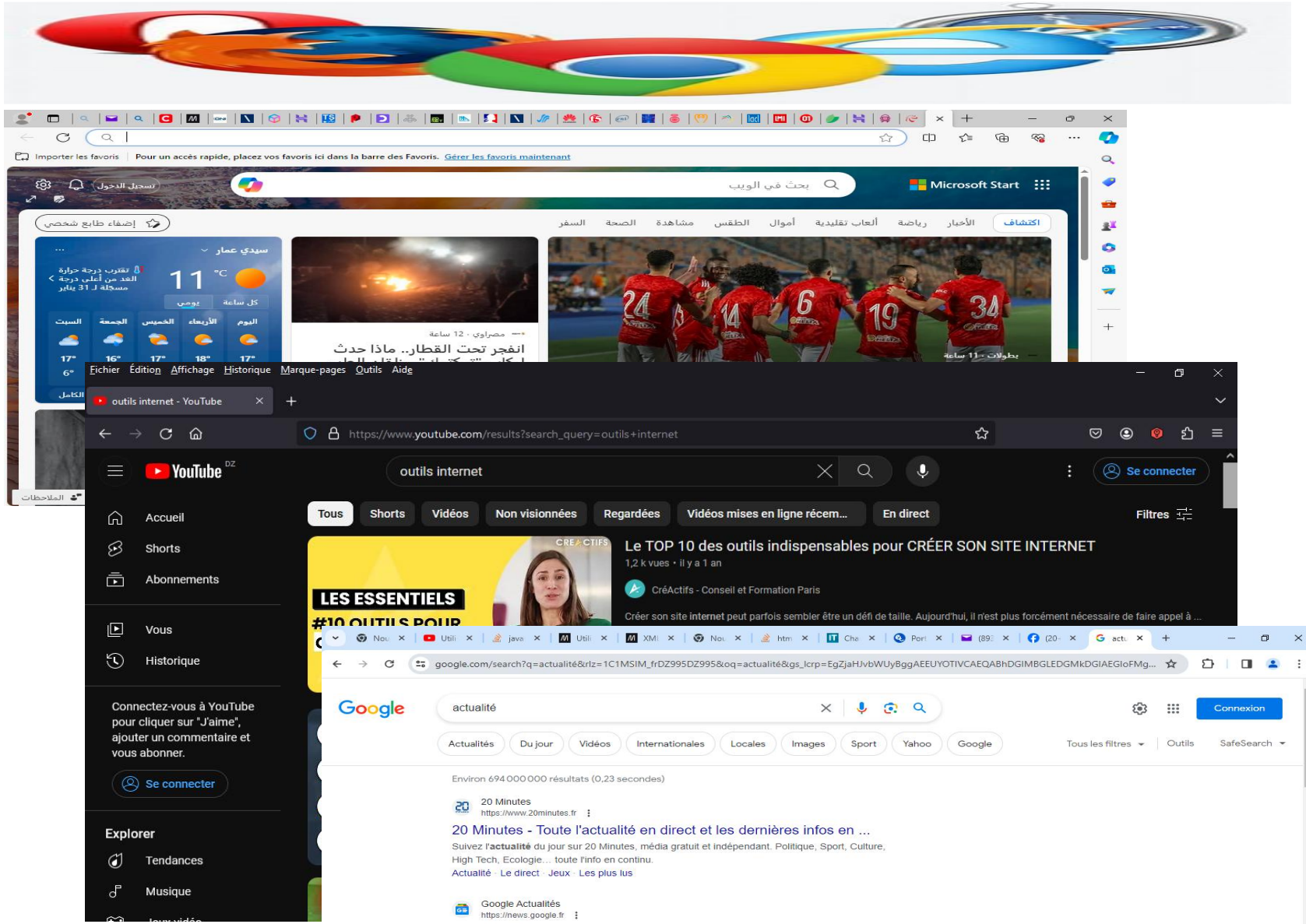
ARCHITECTURE CLIENT/SERVEUR(1)

émettent/ traitent les requêtes (demande de données ou résultat de calcul) sur un port dédié au sein d'un réseau informatique

sur des ordinateurs



sont des applications



DES NOTIONS DE BASES (5) CLIENT/SERVEUR(2)

Le client:

- est un navigateur sur le web,
- qui est un logiciel principalement gratuit
- qui peut s'exécuter sur plusieurs systèmes d'exploitation (interopérable)

- Exemple de navigateur : Internet Explorer, Mozilla Firefox, Google chrome, Microsoft Edge…

DES NOTIONS DE BASES (5)
CLIENT/SERVEUR(3)

- Les navigateurs :
  - permettent d'**accéder** aux sites web.
  - **interprètent** le code des **pages** web pour les afficher correctement sur écran
  - le premier navigateur capable d'afficher une image au format GIF fut : Mosaic sur Xwindows.

  - Les différences entre navigateur et moteur de recherche:

navigateurs	Moteurs de recherche
permettent d'accéder et afficher sites web, via une adresse ou identifiant donné	permettent de trouver des informations spécifiques en ligne, sans adresse spécifique
sont des logiciels installés sur l'ordinateur ou l'appareil mobile personnel d'un utilisateur	sont des sites web accessibles depuis n'importe quel navigateur.

Les serveurs Web regroupés dans des datacenters

DES NOTIONS DE BASES (5) CLIENT/SERVEUR(4)

Les serveurs Web :

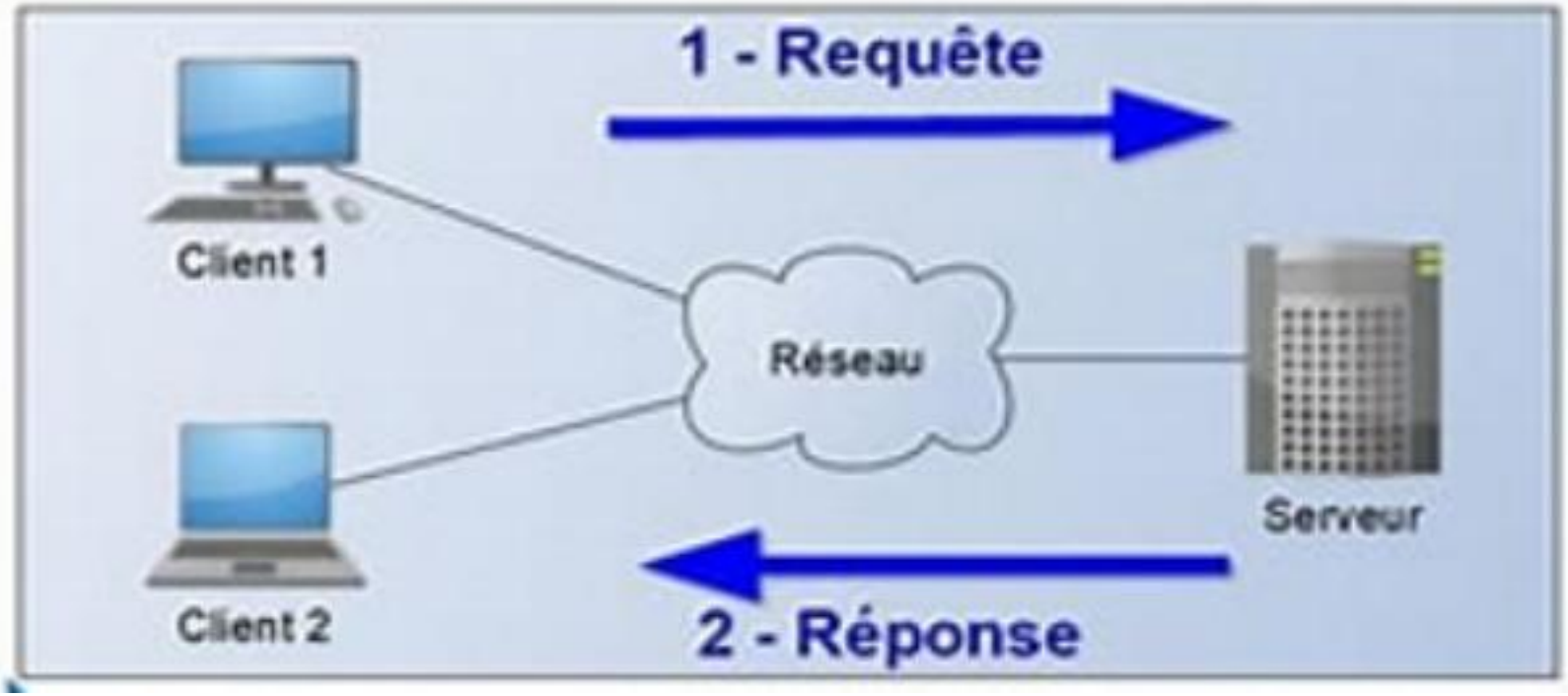
- Matériellement c'est un ordinateur qui héberge les sites web.
- Au niveau logiciel, il contient différents fragments qui contrôlent l'accès aux fichiers hébergés.
- Exemple de serveurs web : Apache, Microsoft IIS, Serveur Web Nginx, Serveur Web LightSpeed. Le serveurs **Apache Http Server**[2] est le serveur le plus répandus.

Quelques serveurs gratuit : **000WebHost(Hostinger), PlanetHoster gratuit,OVH,LWS,Sites.Google.com…**

DES NOTIONS DE BASES (5)
CLIENT/SERVEUR(4)

- Le serveurs **Apache Http Server**
- **D**éveloppé et maintenu par Apache Software Foundation.
- Le logiciel est produit sous licence Apache, ce qui le rend gratuit et à source ouverte.
- Il est disponible pour divers systèmes d'exploitation: Windows, Mac OS X, Unix, Linux, Solaris, Novell Netware et FreeBSD.

L'architecture client -Serveur



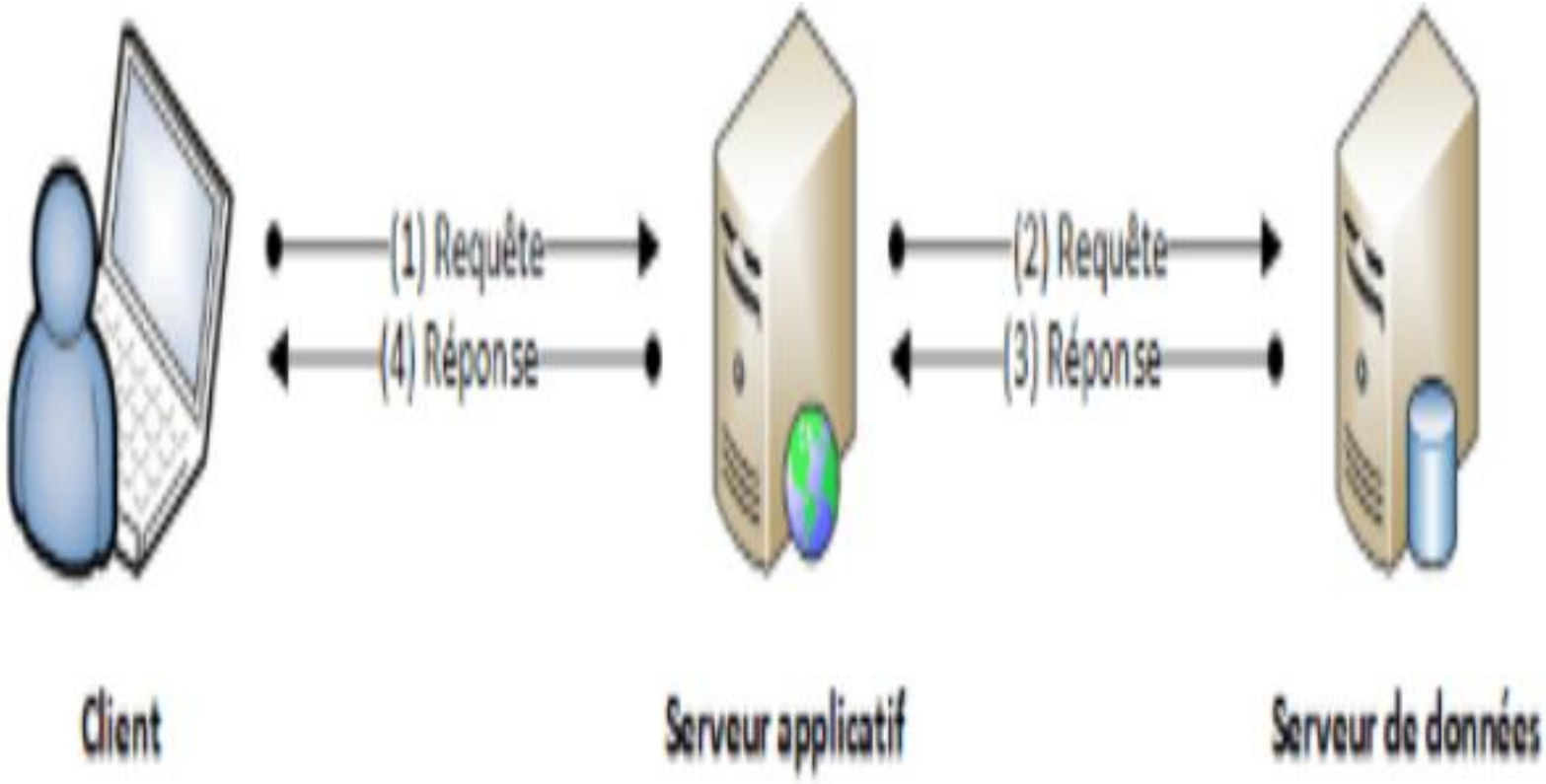
Architecture 2tiers : repose sur deux acteurs seulement :
le client avec sa requête et le serveur qui lui répond

DES NOTIONS DE BASES (5) CLIENT/SERVEUR(4)

Déroulement :

1. le navigateur(client) émet une requête vers un serveur afin d'obtenir la page web désirée.
2. Le serveur envoie les données demandées si le demandeur a le droit d'accès à ce document.
3. Le navigateur interprète les instructions de mise en page contenues dans les données envoyées par le serveur afin d'afficher la page web .
4. Un cas particulier: réseau pair à pair ou (peer-to-peer - P2P). Chaque ordinateur est à la fois client et serveur

- Architecture 3 tiers



- Un serveur web fournit des données statiques telles que des images, des fichiers et du texte et traite exclusivement les requêtes HTTP.
- Un **serveur d'applications** ajoute une **logique métier** pour calculer la réponse du serveur web, via un nombre illimité de protocoles[5].

DES NOTIONS DE BASES (5) CLIENT/SERVEUR(5)

dès lors qu'une Base de Données est utilisée pour générer la réponse du serveur,

- un troisième acteur entre en jeu. C'est le **serveur des bases de données**
- il sera interrogé par le serveur web pour répondre à la requête du client.
- Il accèdera à la base de données pour exécuter la requête
- puis il transmettra la réponse au serveur web.

DES NOTIONS DE BASES (5)

- **Architecture** multi-Tiers ou N niveau **[3,4]:**
- peut contenir un nombre élevé de couches suivant la complexité du problème, et la multiplicité des services.
- peut contenir : des clients légers, des clients lourds, plusieurs serveurs.
- Client « léger »: Le poste client accède à une application située sur un ordinateur dit "serveur" via une interface et un navigateur Web. L'application fonctionne entièrement sur le serveur, le poste client reçoit la réponse.
- Client « lourd »:est un logiciel qui propose des fonctionnalités complexes avec un traitement autonome. Et contrairement au client léger, il ne dépend du serveur que pour l'échange des données dont il prend généralement en charge l'intégralité du traitement.
- Serveur lourd: beaucoup plus de traitement est effectué sur le serveur, le déploiement est plus aisé.

DES NOTIONS DE BASES (5)

- Les avantages de l'architecture client serveur:
  - ◆ Structuration
  - ◆ séparation interface du service - réalisation
  - ◆ client et serveur peuvent être modifiés (remplacés) indépendamment
  - ■ Protection: client et serveur s 'exécutent dans des domaines de protection différents
  - ■ Gestion des ressources: le serveur peut être partagé entre de nombreux clients. Mais, il doit assurer la gestion des ressources partagée.
- Les inconvénients:
  - *Panne de serveur:* une panne serveur entraîne une panne générale du système.
  - *Ressources serveur:* Le serveur exécute des tâches qui demandent beaucoup de ressources. Le client fonctionne avec des exigences en matière de ressources nettement moins élevées. Un serveur qui disposerait de trop peu de ressources aurait une incidence sur le travail de tous les clients.
  - L'intrusion malveillante est le premier danger que recoure un tel modèle, lors des transactions(interception des messages).

DES NOTIONS DE BASES (6)

- **Standard du web :**Le web repose sur trois standards :
  - URL:**(Uniform Resource Locator)** est un sous-ensemble des **URI** (Uniform Resource Identifier).
  - Tout URI indique : le **mécanisme** d'accès à la ressource, l'ordinateur **hébergeant** (serveur) et le **chemin** d'accès sur cet ordinateur à cette ressource. Si le mécanisme d'accès est un protocole (tel que Http, Ftp, SMTP…) on emploie le terme **URL.**
  - Example: https://elearning-facsci.univ-annaba.dz/course/index.php?categoryid=117

Protocole://authentification@nomserveur:port/chemin d'accès ? argument

DES NOTIONS DE BASES (6)

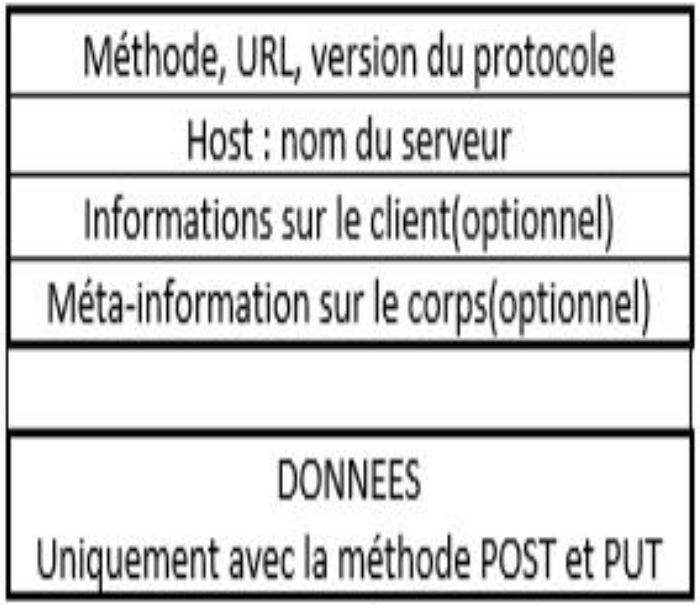
- **Protocole** : il existe plusieurs protocoles tel que: Ftp : Transfert de Fichiers sans visualisation, Smtp : Transfert courrier électronique et HTTP **:**Hypertext Tranfer Protocol est le protocole de transfert d'information sur le WWW…
- **Nom du serveur** : c'est l'adresse internet du serveur. Il peut être une adresse IP ou un nom associé à cette adresse ;
- **Chemin :** est une chaine de caractère décrivant la suite de répertoires qui conduisent à la source séparés par « / ».Il est donné par rapport à la racine du serveur web qui peut être WWW ou précisé par l'administrateur.
- **Document :**si le nom du document est absent c'est le fichier index.html qui est recherché sur le serveur.
- **Les caractères :** le caractère « # » accolé au nom signifie qu'on cherche une partie spécifique de ressource , toute la page sera charger ,mais seulement la partie référencé sui sera visualisés.
- Les caractères autorisés dans l' URL sont :alphabet majuscules et minuscules , chiffres,.,'-',' _ ' ,' !','*' ,'(' ,') ','~ ' , ' ' '.
- **PORT** : est une adresse(entier) unique interne ai serveur pour spécifier l'application avec laquelle le client veut communiquer. le port attribué par défaut à http est 80.

DES NOTIONS DE BASES (7)

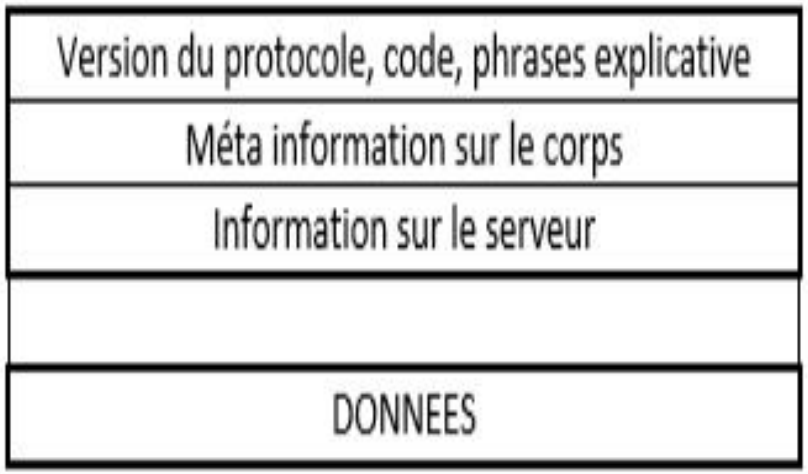
- **Le protocole http :**utilisé depuis 1990.
- Soit un client http se connecte au port 80 du serveur web, il émet une requête http pour :
  - -**obtenir** une page web/ressource multimédias stockée sur le serveur.
  - **Effacer/placer** des fichiers sur le serveur
  - **envoyer** le contenu d'un formulaire au serveur.
- Le serveur traite la requête et envoi une réponse aux clients. La réponse peut être un message d'erreur en cas de non-satisfaction.
- Les requêtes et réponses sont composées de deux partie (entête et le corps) séparée par une ligne. L'entête indique toujours la méthode à appliquer à la ressource.

DES NOTIONS DE BASES (7)

méthode	HTTP	Description
GET	0.9	est utilisée pour récupérer des informations depuis un serveur
POST	1.0	Soumettre le bloc de données contenu dans le corps de la requête à l'url indiquée
HEAD	1.0	Envoyer uniquement l'entête de la réponse
DELETE	1.1	Supprimer du serveur la ressource dont l'URL est précisée
PUT	1.1	Stocker la ressource envoyée dans le corps sur le serveur sous l'URL indiquée



Structure d'une requête



Structure de la réponse

REQUÊTE/RÉPONSE

- sont composées de deux parties **entête** et le **corps** séparée par **une ligne**..
- Si le contenu du formulaire est soumis au serveur avec la méthode POST les données sont envoyées dans le corps.
- une requête avec la méthode GET, le corps est vide. le contenu du formulaire est placé dans la partie argument.
- La deuxième ligne qui commence par « Host » indique le nom du serveur.

REQUÊTE/RÉPONSE

Requête	If-Modified-Since :	GET retourne la ressource uniquement si elle a été modifiée depuis la date indiquée.
	User-agent :	Informations sur le client (nom du navigateur)
	Content-Encoding :	Indique la compression utilisée sur le média contenu dans le corps de la requête. Présent aussi dans la réponse
Réponse	Code	Il informe sur l'état de satisfaction de la requête : 1xx: réponse provisoire ; 100= continue 2xx : requête reçus comprise et acceptée ;200=OK 3xx : redirection traitement incomplet 4xx : erreur client 5xx : erreur serveur
	Content-Language :	Langue utilisée dans le corps
	Date :	Date de la réponse

Quelque champs de l'entête de requête/réponse

VERSIONS HTTP

- La version initiale du HTTP **version 0.9** et s'intitulait « protocole une ligne »,permettait uniquement de consulter un fichier HTML depuis un serveur.

alourdi le téléchargement d'une page Web avec plusieurs fichiers

- En 1996, la **version HTTP/1** fut décrite par l'Internet Engineering Task Force (IETF) dans le mémo RFC1945. cette version du HTTP présentait les propriétés suivantes :
  - **Sans connexion** : le client établit/coupe la connexion avec le serveur à chaque requête.
  - **Sans statut** : les deux côtés, client et serveur, s'« oublient » mutuellement, immédiatement.
  - **Indépendant du média** : le HTTP permet de transmettre n'importe quel type de fichier dans la mesure.

- Le premier standard officiel : **HTTP/1.1(en 1907),**ces nouveautés par rapport au HTTP/1 :
- **Keepalive** : le client peut maintenir la connexion au-delà d'une requête (*persistent connection*) en envoyant un *keepalive* dans l'en-tête.
- Le **HTTP-Pipelining** permettant au client d'envoyer la requête suivante avant d'avoir reçu la réponse à la première requête.
- Dans les **chats**, le navigateur peut actualiser la fenêtre en utilisant le type MIME *multipart/replace*.
- Il est également possible de transmettre des données **du client au serveur**.
- **TRACE :Méthode** introduite permet de suivre le chemin du client au serveur Web.
- **Cache** : de nouveaux mécanismes pour mettre des contenus en mémoire tampon sont disponibles.
- **Host** : les requêtes HTTP fonctionnent également lorsque plusieurs domaines différents sont hébergés sous une même adresse IP (Shared Webhosting).

VERSIONS HTTP

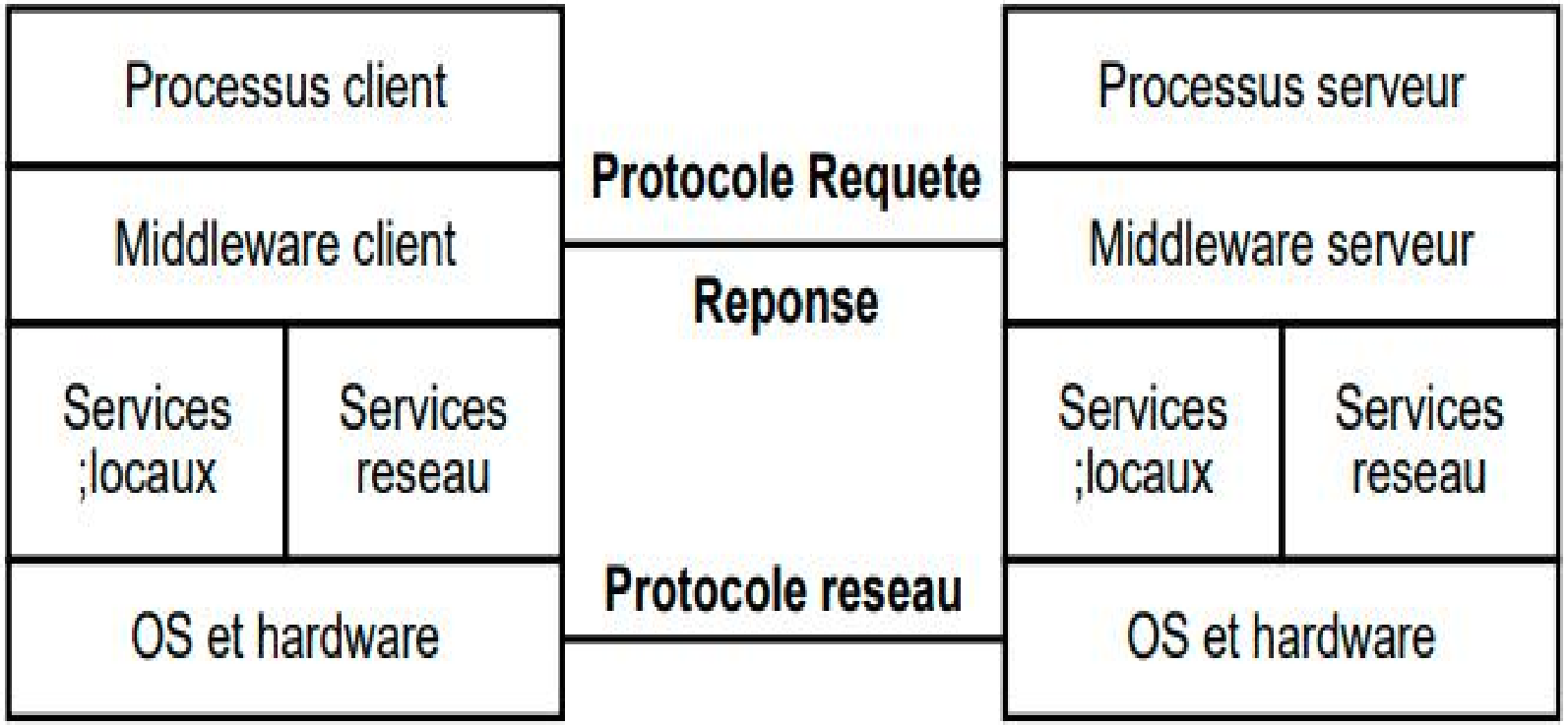
Les *standards* sont définis par des groupes qui n'ont pas de mandats officiels des gouvernements. Un standard est aussi un procédé ou un service qui est largement répandu.

VERSIONS HTTP

- **HTTP/2**
- nouveau **standard** ayant pour objectif d'accélérer le chargement des sites Internet.
- **Binaire** : le protocole est basé sur des données binaires plutôt que sur des fichiers texte.
- **Multiplexe** : le client et le serveur peuvent envoyer et/ou traiter plusieurs requêtes HTTP en parallèle.
- **Compression** : les en-têtes sont compressés pour éviter les redondances inutiles.
- **Serveur Push :** lorsque le serveur est en mesure de prévoir les données dont le client aura encore besoin, il peut les lui envoyer d'emblée dans une mémoire cache automatique.

VERSIONS HTTP

- **HTTPS :**
- HTTPS : n'est autre qu'http mais en plus utilise un protocole de chiffrement pour chiffrer les communications [7].
- Le protocole : Transport Layer Security (TLS)qui était auparavant connu sous le nom de Secure Sockets Layer (SSL) sécurise les communications en utilisant une infrastructure à **clé publique asymétrique**, en utilisant deux clés différentes pour chiffer les communications entre deux parties :
- **La clé privée** : est contrôlée par le propriétaire du site web, réside sur un serveur web et utilisée pour déchiffrer les informations chiffrées par la clé publique.
- **La clé publique** : est disponible pour tous ceux qui souhaitent interagir avec le serveur de manière sécurisée.



Fonctions:

- Procédure établissement/fermeture de connexion
- Mise en cache/Exécution des requêtes
- Récupération des réponse
- Accès aux données à distance
- Gestion des accès concurrents
- Sécurité et intégrité
- …

MIDDLEWARE (INTERGICIEL)

Appelé aussi médiateur, est l'ensemble des couches réseau et services logiciel (l'application)qui assure le dialogue entre les différents composant d'une application répartie.

Il est basé sur

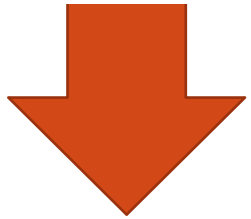
- API : Application Programming Interfaces (interface de programmation au niveau applicatif)
- FAP : Format And Protocols (protocoles de communication et formats de données)

MIDDLEWARE (INTERGICIEL)

- Type et exemples de middlleware[6]:
- Middleware RPC (Remote Proceedure Call)
  - RPC de SUN
- Middlewares orientés objets distribués
  - Java RMI, Corba
- Middlewares orientés composants distribués
  - EJB, Corba, DCOM
- Middlewares orientés messages
  - JMS de Sun, MSMQ de Microsoft, MQSeries de IBM
- Middlewares orientés sevices
  - Web Services (XML-RPC, SOAP)
- Middlewares orientés SGBD
  - ODBC (Open DataBase Connectivity), JDBC de Sun

```html
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="style.css" />
</head>
<body>
<form class="box" action="" method="post" name="login">
<h1 class="box-logo box-title"><a href="https://waytolearnx.com/">WayToLearnX.com</a></h1>
<h1 class="box-title">Connexion</h1>
<input type="text" class="box-input" name="username" placeholder="Nom d'utilisateur">
<input type="password" class="box-input" name="password" placeholder="Mot de passe">
<input type="submit" value="Connexion " name="submit" class="box-button">
<p class="box-register">Vous êtes nouveau ici? <a href="register.php">S'inscrire</a></p>

</form>
</body>
</html>
```



COMPOSANTS D'APPLICATION

la couche présentation (interface Homme/Machine) : gestion de l'affichage, interfaces textuelles ou graphiques, interactions, entrée des données, , …

Langages et outils:

HTML, CSS, Applets, Javascript…

```php
<?php

session_start();

if (isset($_POST['username'])){
    $username = stripslashes($_REQUEST['username']);
    $username = mysqli_real_escape_string($conn, $username);
    $password = stripslashes($_REQUEST['password']);
    $password = mysqli_real_escape_string($conn, $password);
    $query = "SELECT * FROM `users` WHERE username='$username' and password='".hash('sha256', $password)."'";
    $result = mysqli_query($conn,$query) or die(mysql_error());
    $rows = mysqli_num_rows($result);
    if($rows==1){
        $_SESSION['username'] = $username;
        header("Location: index.php");
    }else{
        $message = "Le nom d'utilisateur ou le mot de passe est incorrect.";
    }
}
?>

...

<?php if (! empty($message)) { ?>
    <p class="errorMessage"><?php echo $message; ?></p>
<?php } ?>
```

Ce traitement est local

- Ouverture d'une session
- Lancement d'une requête SQL
- Contrôle du résultat dans la variable $message

COMPOSANTS D'APPLICATION

la couche traitements (ou logique) qui assure la fonctionnalité intrinsèque de l'application (algorithme), traitements associés à l'application .

Applets, Javascript, PHP, CGI, Servelets…

```sql
-- phpMyAdmin SQL Dump
-- version 4.7.4
-- https://www.phpmyadmin.net/
--
-- Hôte : 127.0.0.1:3306
-- Généré le :  jeu. 09 jan. 2020 à 12:07
-- Version du serveur :  5.7.19
-- Version de PHP :  7.1.9

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET AUTOCOMMIT = 0;
START TRANSACTION;
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de données :  `registration`
--

-- --------------------------------------------------------

--
-- Structure de la table `users`
--

DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `password` varchar(100) NOT NULL
```

COMPOSANTS D'APPLICATION

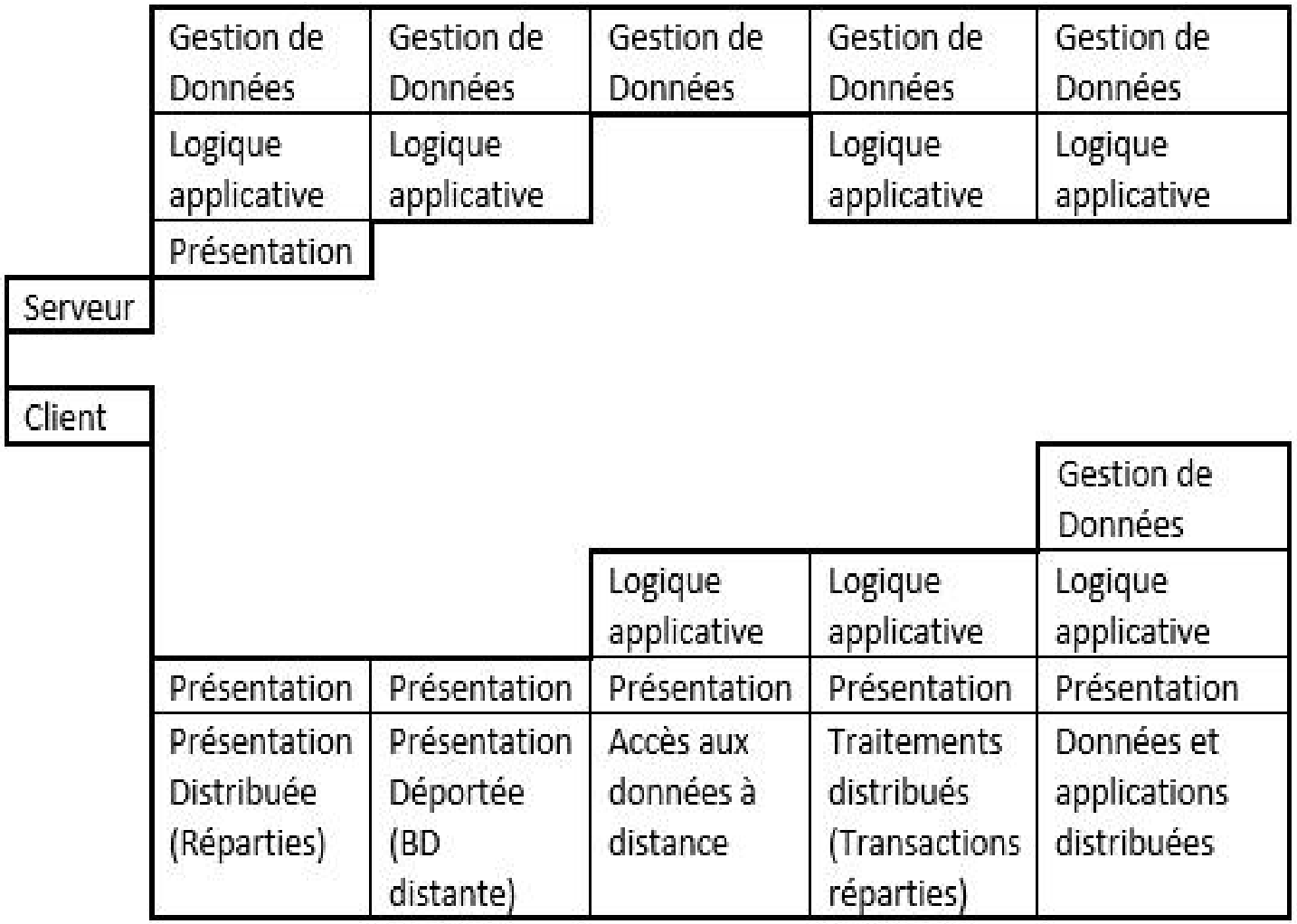
la couche données qui assure la gestion des données de l'application (stockage et accès) : stockage et accès aux données (base de données, serveur web, etc).implémenter via PHP, CGI, Servelets,SQL,…

Lancement d'une requête sql
après récupération des donnéees
saisie dans un formulaire

PAGE WEB DYNAMIQUE

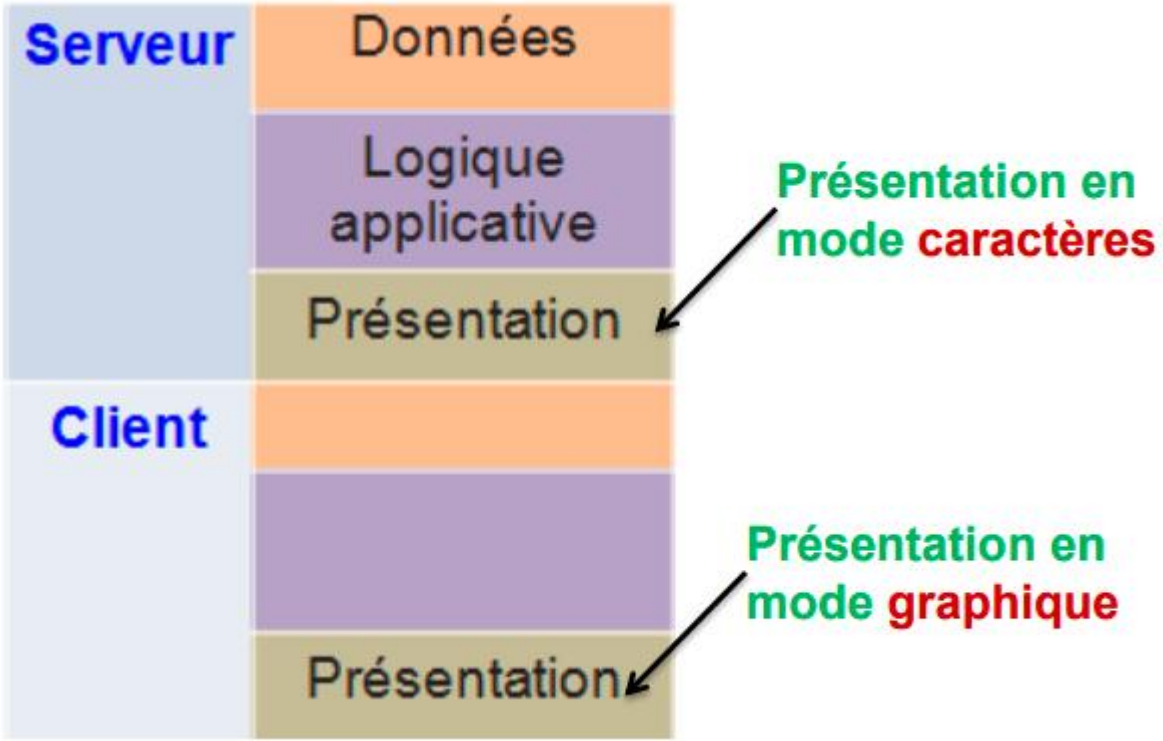
```
</head>
<body>
<?php
require('config.php');
session_start();

if (isset($_POST['username'])){
    $username = stripslashes($_REQUEST['username']);
    $username = mysqli_real_escape_string($conn, $username);
    $password = stripslashes($_REQUEST['password']);
    $password = mysqli_real_escape_string($conn, $password);
    $query = "SELECT * FROM `users` WHERE username='$username' and password='".hash('sha256', $password)."'";
    $result = mysqli_query($conn,$query) or die(mysql_error());
    $rows = mysqli_num_rows($result);
    if($rows==1){
        $_SESSION['username'] = $username;
        header("Location: index.php");
    }else{
        $message = "Le nom d'utilisateur ou le mot de passe est incorrect.";
    }
}
?>
<form class="box" action="" method="post" name="login">
<h1 class="box-logo box-title"><a href="https://waytolearnx.com/">WayToLearnX.com</a></h1>
<h1 class="box-title">Connexion</h1>
<input type="text" class="box-input" name="username" placeholder="Nom d'utilisateur">
<input type="password" class="box-input" name="password" placeholder="Mot de passe">
<input type="submit" value="Connexion " name="submit" class="box-button">
<p class="box-register">Vous êtes nouveau ici? <a href="register.php">S'inscrire</a></p>
<?php if (! empty($message)) { ?>
    <p class="errorMessage"><?php echo $message; ?></p>
<?php } ?>
</form>
</body>
</html>
```



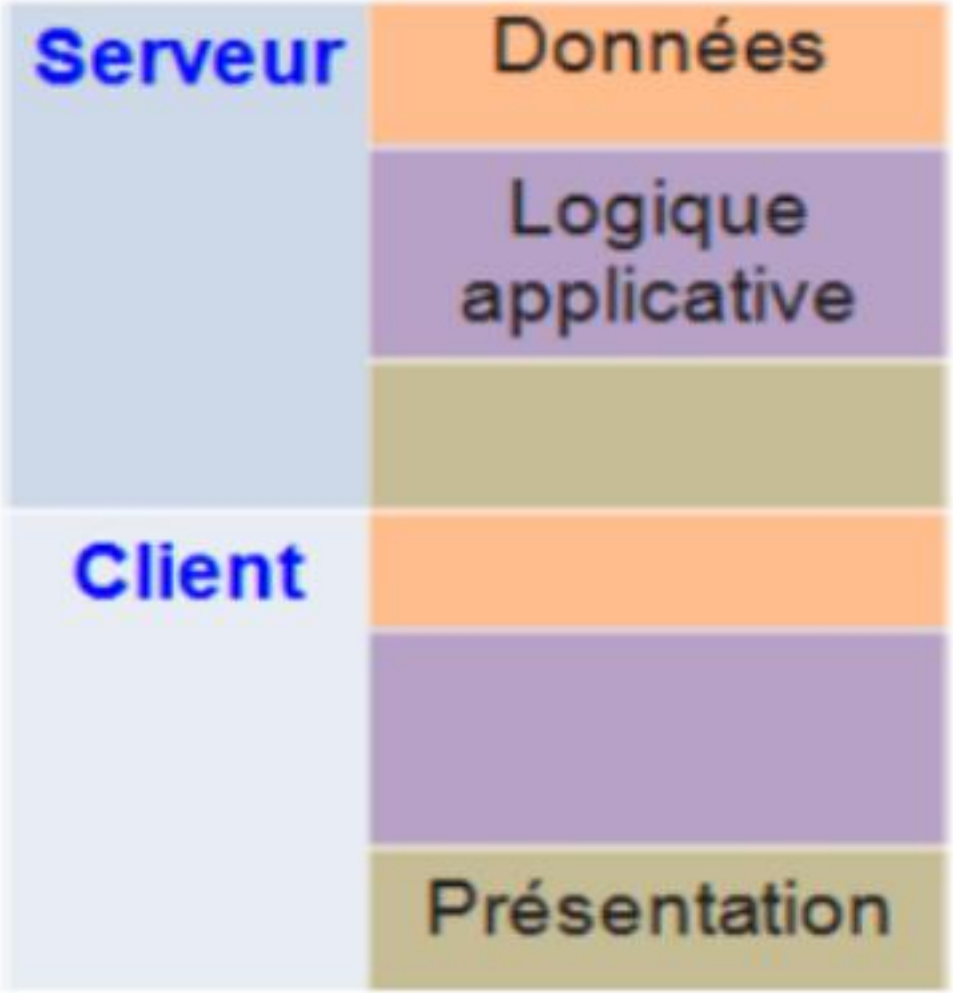
CLASSIFICATION DE GARTNER

- Cette classification ne constituent pas obligatoirement les étapes possibles d'un projet client-serveur,.
- Le choix d'une classe dépend de plusieurs facteurs selon :
  - la nature et l'objectif de l'application cible ainsi que son organisation
  - les moyens financiers et humains dont il dispose.
- Cette classification n'est intéressante que si l'on considère que les éléments mis en jeu dans les différentes couches sont hétérogènes.



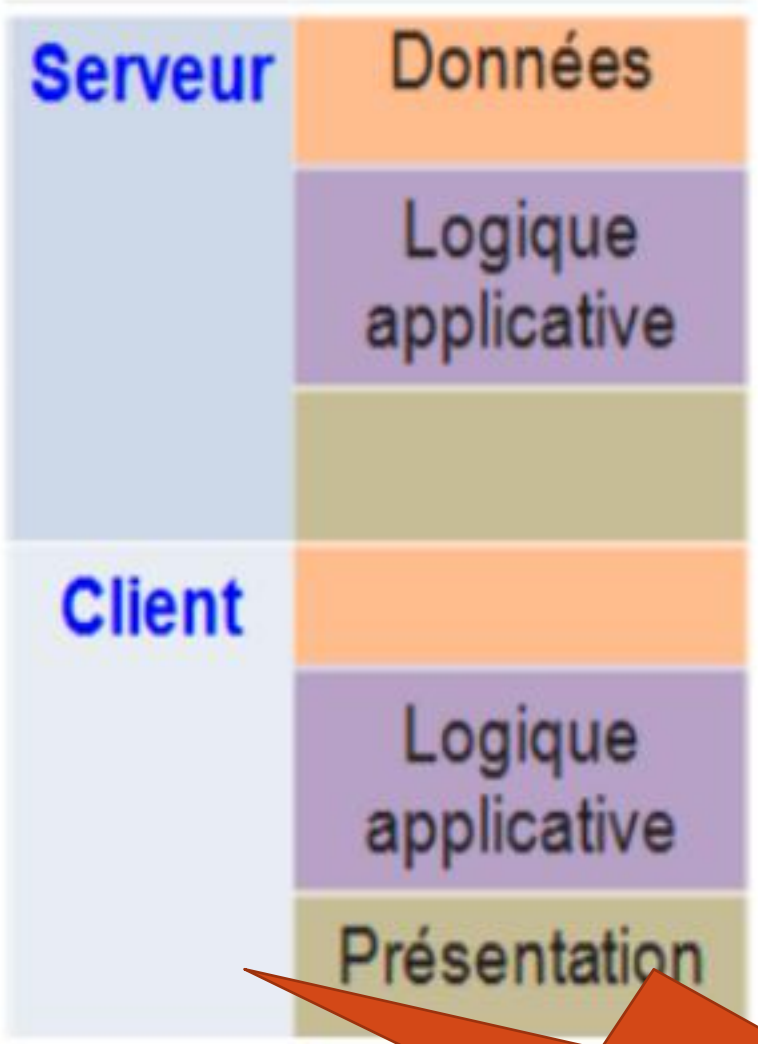
Présentations réparties

- Présentations réparties : Ajustement graphique de l'affichage en mode caractères d'applications fonctionnant sur site central
- le client ne fournit qu'un habillage neuf des données transmises, conçue au départ en mode caractère, en faisant appel à des outils spécifiques d'habillage dits de rewamping (ex. Rumba/400).
- utilisé lorsqu'on ne peut se séparer de ses ordinateurs centraux et des logiciels qui fonctionnent dessus.



Présentations déportée

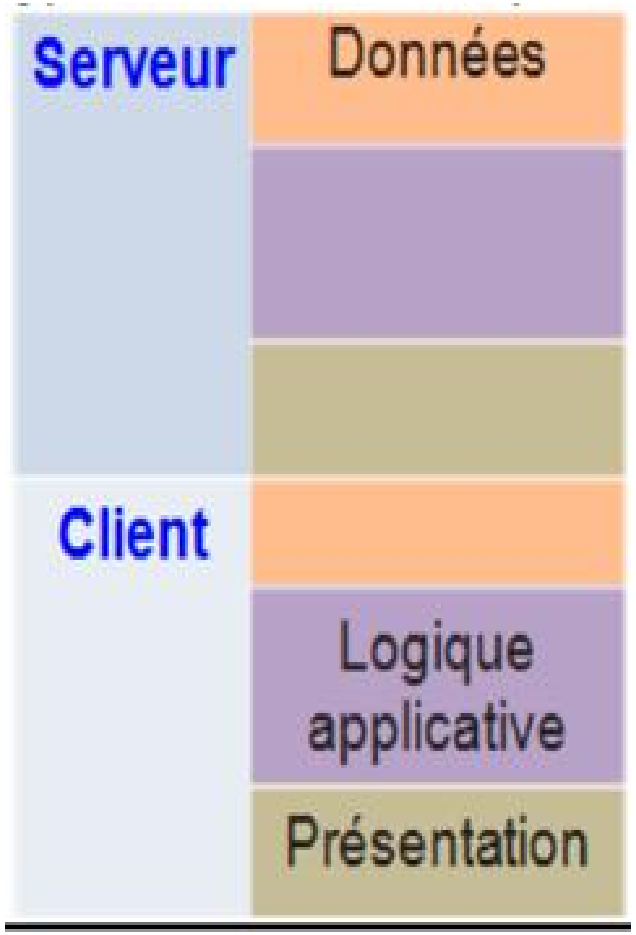
- **la présentation déportée(distante)**:appelé client/ serveur de présentation
- Aucune répartition des charges entre client et serveur
- dédiée à des architectures avec des serveurs puissants et des postes clients très simples (comme des terminaux X).



Cependant il existe plusieurs clients sur le réseau???comment gérer

TRAITEMENTS RÉPARTIS

- **les applications avec des traitements répartis** qui laissent la gestion des données seulement sur serveur (pour des raisons de sécurité et d'accessibilité).
- Les traitements sont distribués entre client et serveurs.
- La présentation existe seulement du coté client.
- Il permet l'optimisation du trafic et de charge.
- Liaison étroite entre client et serveur
- C'est le modèle classique des applications SQL émettant des requêtes complexes vers des bases de données.



Base de données distante

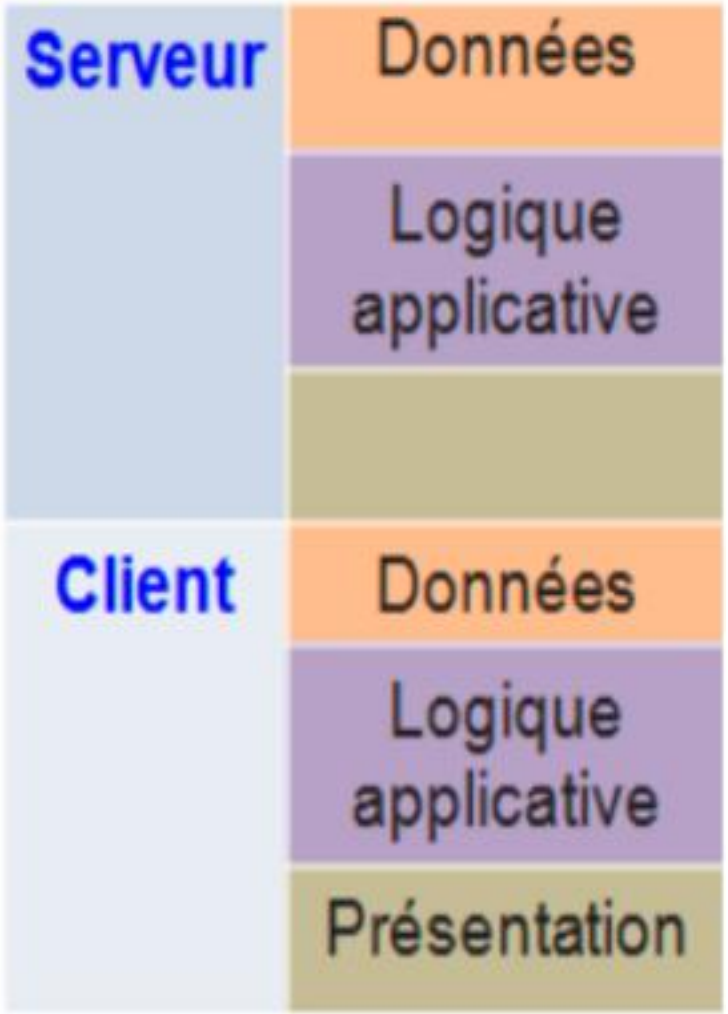
l'accès aux données à distance laissent le gros du travail au client ,appelée client/serveur de données est la plus répandu

accèdent à des serveurs disposant d'une logique applicative de gestion des transaction et des traitements locaux (telles que les procédures stockées).

Pour des application infocentre/banques de données

Trafic réseau assez important

Ne soulage pas le client



TRAITEMENT ET BASE DE DONNÉES RÉPARTIES

Le modèle décentralisé intègre des données et des applications réparties.

Tire partie de la notion de composants réutilisables et distribuable pour répartir la charge

Ce modèle est intéressant si on gère de gros volumes de données, si l'on souhaite disposer de temps d'accès très rapides sur certaines données ou pour répondre à de fortes contraintes de confidentialité.

l'architecture la plus complexe à mettre en œuvre.

	présentation	**présentation**	**présentation**	**présentation**
client			Logique applicative	Logique applicative
				Données
Serveur du milieu	Logique applicative	Logique applicative	Logique applicative	Logique applicative
				Données
Serveur		Logique applicative		Logique applicative
	Données	Données	Données	Données

CLASSIFICATION DE GARTNER AVEC ARCHITECTURE 3TIERS

- met la logique et le traitement de l'application sur un serveur spécialisé.
- Le client conserve la présentation incluant le dialogue avec l'utilisateur et plus généralement l'interface homme-machine.
- enlève l'application du poste client, là où elle est le plus difficilement maintenable et sécurisable.
- l'ajout de serveurs intermédiaires complique l'architecture finale déployée et les règles de gestion associées.

RÉFÉRENCES

- Chapitre1:
- [1]: Quelle est la différence entre Internet et Web ? - Cyber Preventys
- [2}Internet et le web (cea.fr)
- [3] Serveur d'application (univ-batna2.dz)
- [4]notes de cours Outils Internet DR.YKOUBI,UNIV ANNABA
- [5]Différence entre Serveur d'application et Serveur web – WayToLearnX
- [6]Introduction aux Architectures Distribuées & Intergiciel Département d'Informatique, université de Batna2