

Développement d'Applications Web DAW

Architecture Client/Serveur

Présenté par :

Me Derkaoui Orkia

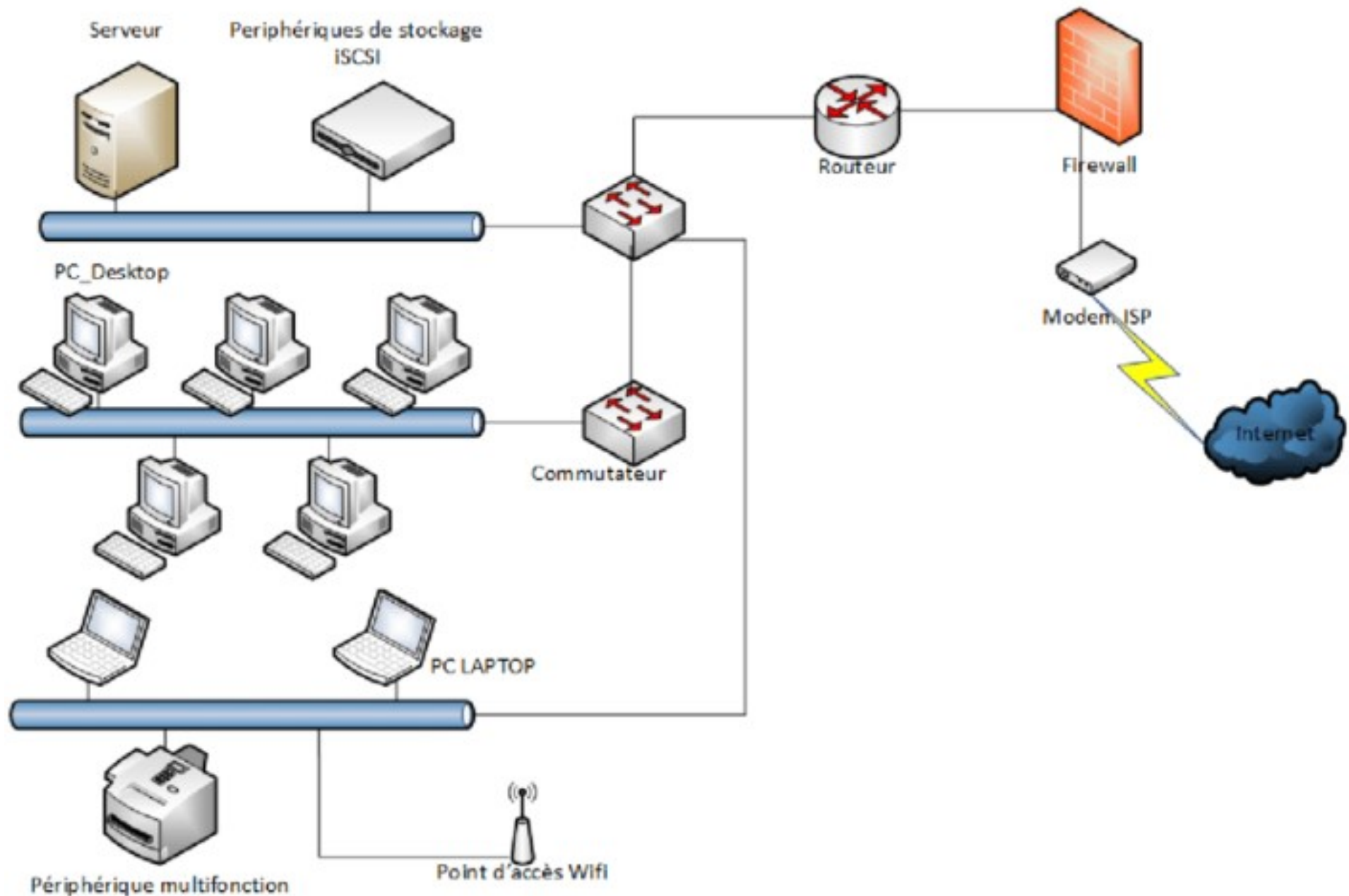
Université Docteur Moulay TAHAR de Saida
derkaouiorkia@gmail.com

Plan du cours

Chapitre 1: Introduction au World Wide Web (WWW)

- Définition et historique
- Architecture Client/Serveur
- Protocole HTTP.

Exemple de Réseau empruntée à [5]



Architecture Client/Serveur

- Une architecture client-serveur représente l'environnement dans lequel des applications de machines clientes communiquent avec des applications de machines de type serveurs.
- L'exemple classique est le navigateur Web d'un client qui demande (on parle de "requête") le contenu d'une page Web à un serveur Web qui lui renvoie le résultat (on parle de "réponse"), [1].

Architecture Client/Serveur

- Un client : Les caractéristiques d'un client sont les suivantes : il est d'abord actif (ou maître), il envoie des requêtes au serveur, il attend et reçoit les réponses du serveur.
- Un serveur : Un serveur est initialement passif, il attend, il est à l'écoute, prêt à répondre aux requêtes envoyées par des clients.
- Dès qu'une requête lui parvient, il la traite et envoie une réponse. [8]

Architecture Client/Serveur

Le dialogue entre le client et le serveur :

- Le client et le serveur doivent bien sûr utiliser le même protocole de communication.
- Un serveur est généralement capable de servir plusieurs clients simultanément.
- Remarques : Une fois le client traité, le serveur peut en traiter un autre.
- Il existe des serveurs multiclents comme les serveurs Web/http qui sont capables de traiter plusieurs clients en même temps.
- Il existe aussi des serveurs « non connectés », dans ce cas il n'y a pas de connexion ou de déconnexion, [8].

Architecture Client/Serveur

Qu'est-ce qu'une architecture client-serveur ?

Definition 1 : Une architecture client-serveur représente l'environnement dans lequel des applications de machines clientes communiquent avec des applications de machines de type serveurs.

L'exemple classique est le navigateur Web d'un client qui demande (on parle de "requête") le contenu d'une page Web à un serveur Web qui lui renvoie le résultat (on parle de "réponse").

[1]

Architecture Client/Serveur

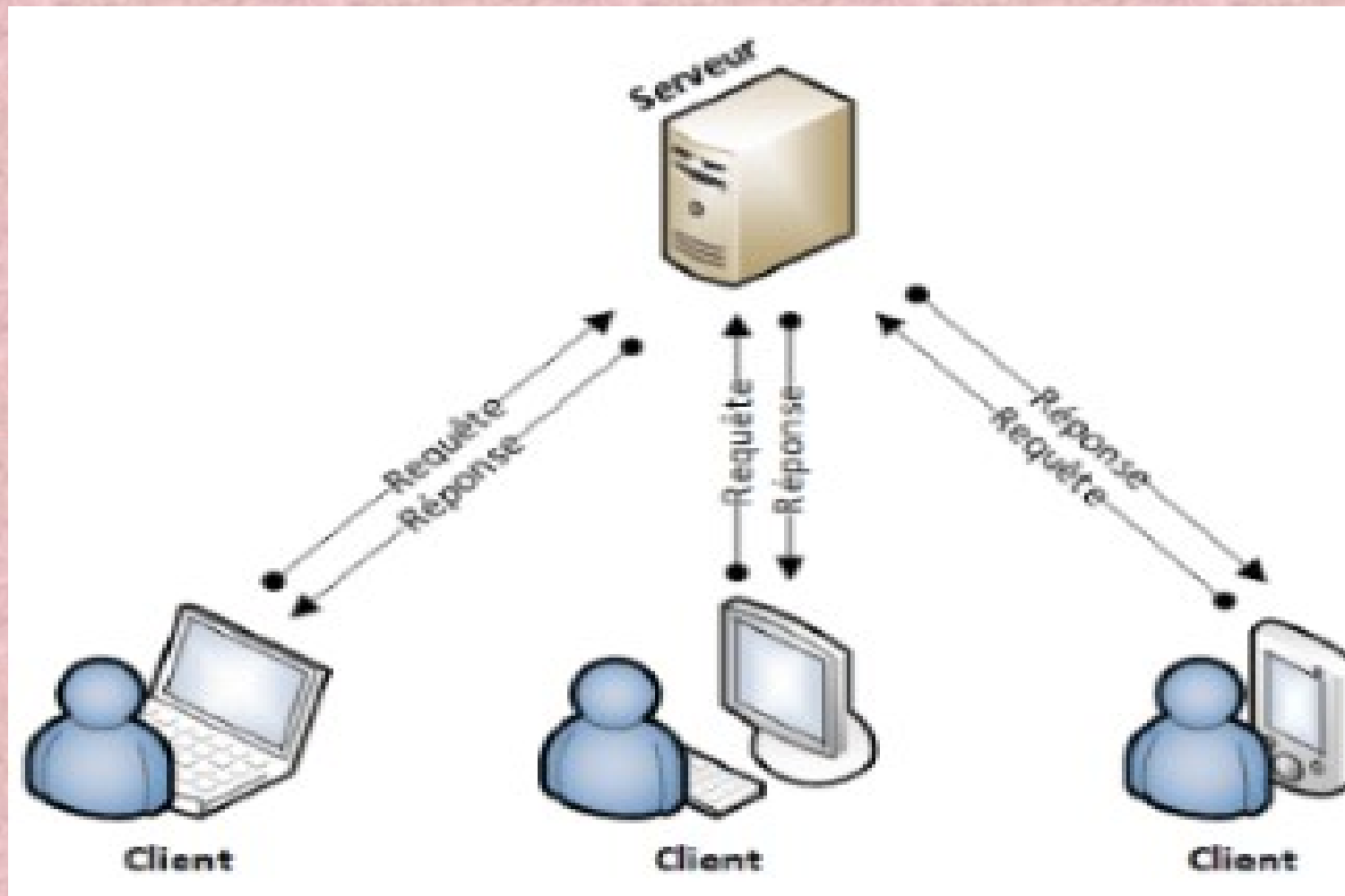


Image empruntée à [1]

Architecture Client/Serveur

Qu'est-ce qu'une architecture client-serveur ?

Definition 2 :

- Dans un réseau internet, les nœuds peuvent être des serveurs ou des clients.
- **Les nœuds serveur** fournissent des ressources telles que la mémoire, la puissance de traitement ou les données aux nœuds clients. Les nœuds serveur peuvent également gérer le comportement des nœuds clients.
- **Les clients** peuvent communiquer entre eux, mais ils ne partagent pas de ressources. Par exemple, certains appareils informatiques dans des réseaux d'entreprise stockent des données et des paramètres de configuration. Ces appareils sont les serveurs du réseau. Les clients peuvent accéder à ces données en adressant une requête à la machine serveur. [2]

Serveur

- **Qu'est-ce qu'un serveur informatique ? [7]**
- C'est un système qui fournit des données, des services ou des programmes informatiques accessibles sur un réseau internet ou intranet.
- Il s'apparente à un super ordinateur qui, à la demande d'un autre ordinateur appelé « client », envoie des réponses adaptées puisées à partir d'une multitude de ressources. Il traite les requêtes effectuées à partir d'un PC personnel, d'un smartphone ou d'une tablette : on parle alors de relation « client-serveur ».

Client

Types de clients applicatifs [1]

Les clients applicatifs, sont des logiciels qui tournent sur les machines ou terminaux des utilisateurs. Il est possible d'en distinguer 3 types majeurs:

- **Client léger**
- **Client lourd**
- **Client riche**

Client

Client léger

- Un *client léger* est une application où le traitement des requêtes est entièrement effectué par le serveur, le client se contente de recevoir et mettre en forme pour afficher les réponses calculées et envoyées par le serveur. Le client le plus souvent est un navigateur Web, avec des pages web n'utilisant pas ou peu de JavaScript côté client, terminaux Terminal Services, Secure Shell, Apple Remote Desktop, Citrix XenApp, TeamViewer, etc.)

Quelques avantages:

- peu de puissance de calcul est nécessaire au niveau du client.
- la mise à jour de l'application s'effectue uniquement sur le serveur, excepté l'éventuelle mise à jour du client Web.
- plus grande indépendance du développement de l'application et du serveur vis à vis de la machine cliente et de son environnement.
- un travail de développement concentré sur le serveur

Client

Client lourd

- Un *client lourd* est une application (applications de bureau, applications mobile) où les traitements sont principalement effectués sur la machine locale dite cliente. Le serveur se contentant principalement de répondre aux demandes de données du client.

Quelques avantages:

- le client peut parfois fonctionner même en cas de déconnexion du serveur
- une partie des traitements est réalisé par le client, ce qui soulage les ressources du serveur.
- plus grande indépendance vis à vis des temps de réponse réseau et serveur

Client

Client riche

- Un *client riche* est une application où le traitement des requêtes du client (applications Web utilisant beaucoup de JavaScript côté client) est effectué majoritairement par le serveur, le client recevant les réponses « semi-finies » et les finalisant.
- C'est un client léger plus évolué permettant de mettre en œuvre des fonctionnalités comparables à celles d'un client lourd. C'est un compromis entre les clients légers et lourds.

Serveur Web

- **Qu'est-ce qu'un serveur informatique ? [7]**

- Le serveur informatique est un outil qui peut être matériel (hardware), logiciel (software) ou virtuel.
- En *hardware*, il s'agit d'un système qui relie un réseau d'ordinateurs à des données stockées en local ou en cloud.
- En software, le serveur informatique effectue des interactions en réseau avec d'autres programmes appelés logiciels clients.
- Le serveur informatique qualifié de virtuel se base sur un dispositif physique qui accueille plusieurs serveurs virtualisés grâce à un logiciel ad hoc : l'hyperviseur.

Quels sont les différents types de serveurs informatiques ?

Compte tenu de la multitude de services qu'il propose, le serveur informatique se décline en plusieurs catégories.

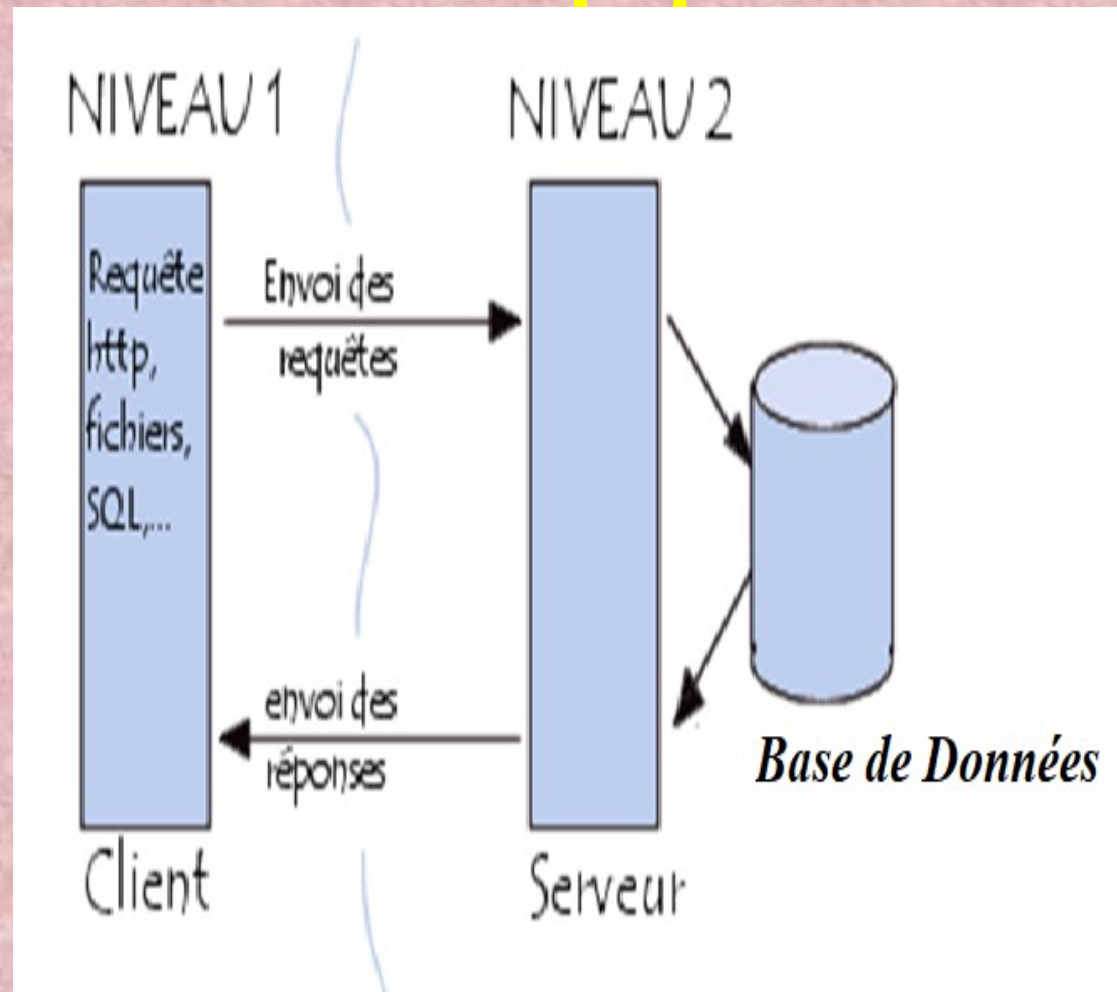
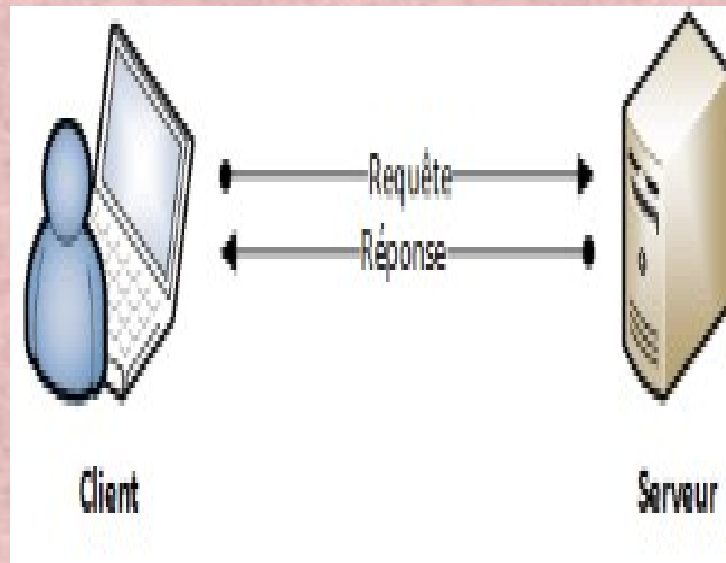
- Les serveurs web : HTTPS
- Les serveurs de fichiers : FTP
- Les serveurs de bases de données:
- Les serveurs de messagerie :
- Les serveurs DNS :
- Les serveurs d'impression :
- Les serveurs d'applicatifs métiers :
- Les serveurs PROXY :
- Les serveurs de jeux proposent de jouer en ligne.

Architecture 2-tier [6]

Architecture 2-tier (2 niveaux)

- L'architecture à deux niveaux (aussi appelée *architecture 2-tier*, *tier* signifiant *rangée* en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource et le serveur la lui fournit directement, en utilisant ses propres ressources.
- Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir une partie du service.

Architecture 2-tier [6]



Images empruntées à [1] [6]

Architecture 2-tier [6]



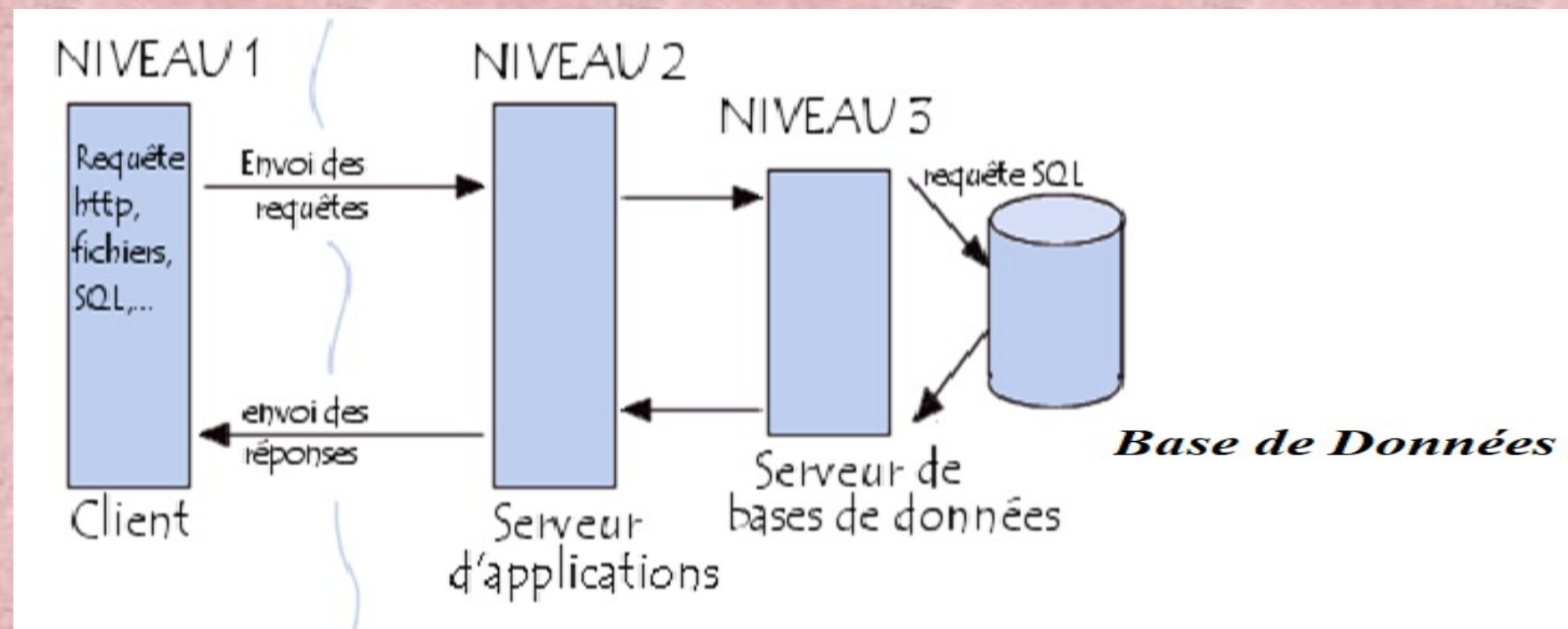
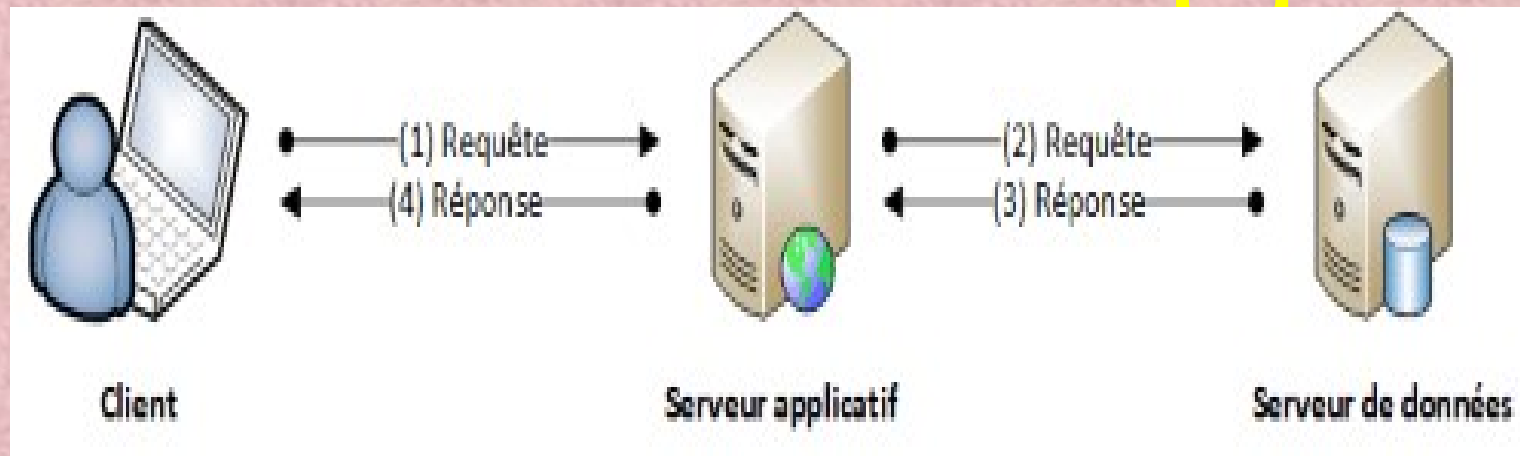
Images empruntées à [6]

Architecture 3-tier [6]

Dans l'architecture à 3 niveaux (appelée *architecture 3-tier*), il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

1. Un client, c'est-à-dire l'ordinateur demandeur de ressources, équipée d'une interface utilisateur (généralement un navigateur web) chargée de la présentation ;
2. Le serveur d'application (appelé également **middleware**), chargé de fournir la ressource mais faisant appel à un autre serveur
3. Le serveur de données, fournissant au serveur d'application les données dont il a besoin.

Architecture 3-tier [6]



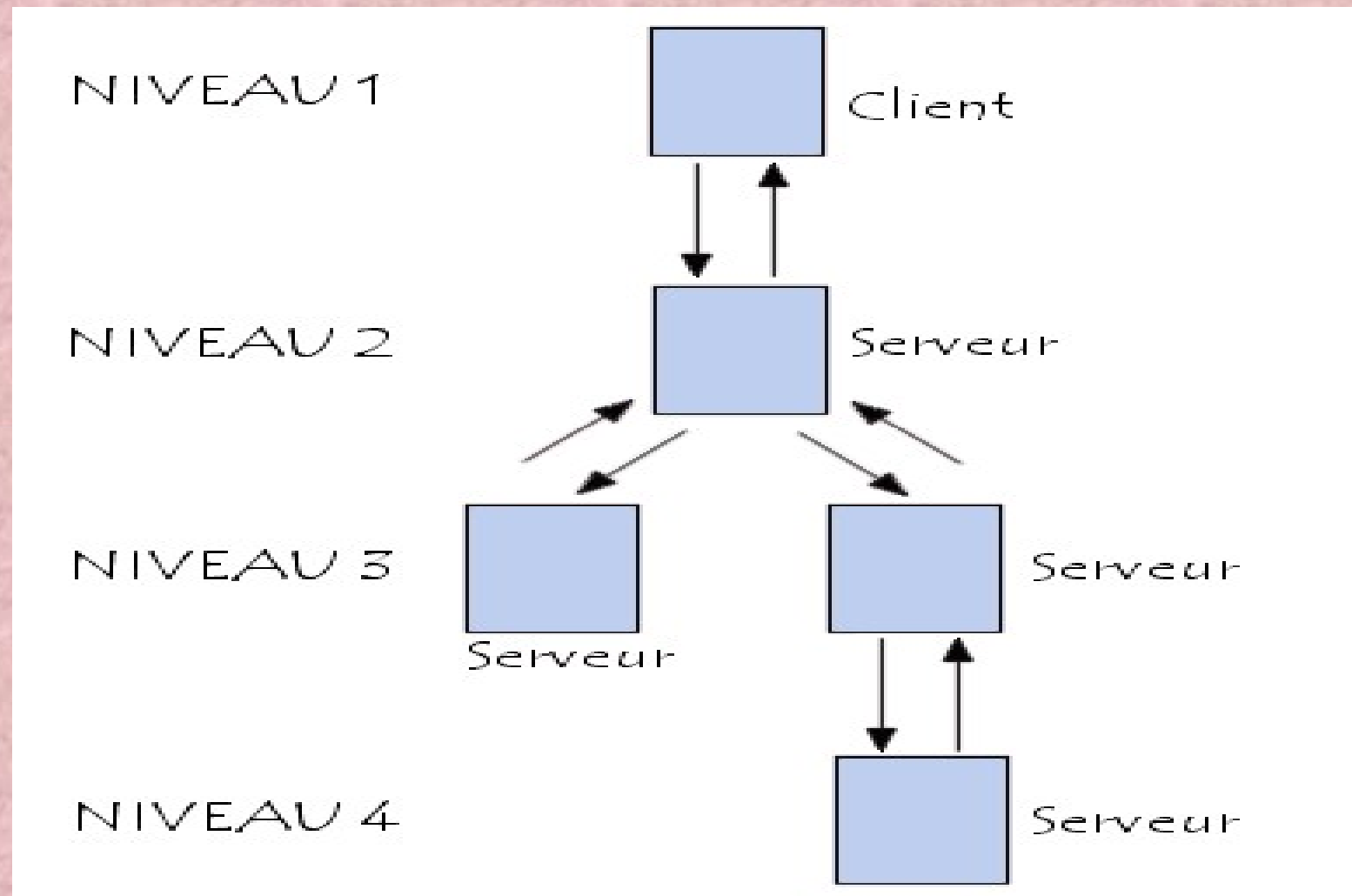
Images empruntées à [1] [6]

Architecture n-tier [6]

Architecture n-tier (multi niveaux)

Dans l'architecture à 3 niveaux, chaque serveur (niveaux 2 et 3) effectue une tâche (un service) spécialisée. Un serveur peut donc utiliser les services d'un ou plusieurs autres serveurs afin de fournir son propre service. Par conséquent, l'architecture à trois niveaux est potentiellement une architecture à N niveaux...

Architecture n-tier [6]



Images empruntées à [6]

Protocole de Communication

- Dans les réseaux informatiques et les télécommunications, un **protocole de communication** est une spécification de plusieurs règles pour un type de communication particulier.
- Initialement, on nommait protocole ce qui est utilisé pour communiquer sur une même couche d'abstraction entre deux machines différentes.
- Par extension de langage, on utilise parfois ce mot aussi aujourd'hui pour désigner les règles de communication entre deux couches sur une même machine.[9]

Protocole HTTP

HTTP (HyperText Transfer Protocol)

- La consultation des pages sur un site web a un fonctionnement basé sur une architecture client/serveur. Un internaute connecté au réseau via son ordinateur et un navigateur web est le client,
- le serveur est constitué par le ou les ordinateurs contenant les applications qui délivrent les pages demandées.
- Dans ce cas, c'est le protocole de communication HTTP (HyperText Transfer Protocol) qui est utilisé.
- Les navigateurs sont les clients (Firefox, Internet Explorer, Google Chrome...).
- Ces clients se connectent à des serveurs http tels qu'Apache ,http Server ou IIS (Internet Information Services).

Protocole HTTPS

HTTPS (avec S pour Secured ou Sécurisé)

- **HTTPS** est similaire à **HTTP**, mais le «S» signifie Secure.
- En **HTTP**, les données sont transmises telles qu'elles sont appelées en
- texte brut. Tout le monde peut lire **entre** le serveur et le client.
- Mais dans **HTTPS**, personne ne peut lire les informations **entre** le
- serveur et le client.
- **HTTPS** est la variante de http sécurisée par l'usage des
- Protocoles :
 - **SSL** (Secure Sockets Layer) ou nouvellement nommé
 - **TLS** (Transport Layer Security) qui est un protocole de sécurisation des échanges sur internet.

URL

- Une **URL** (en Anglais Uniform Resource Locator) est l'adresse d'un site web. L'adresse d'un site web, ou plus exactement **l'adresse d'une page d'un site internet**.
- L'URL est indispensable pour localiser cette page dans l'océan des milliards de pages internet existantes. Voilà pourquoi la structure des URL est normalisée dans le monde entier.
- L'URL de votre site est donc l'identité sur le web de votre marque. Que vous décidiez de passer par un logiciel de création de site professionnel ou bien par un autre prestataire, votre URL doit vous « appartenir ». [10]

La structure et les éléments d'une URL

- Une URL est en général composée de plusieurs éléments. Pour comprendre la structure et les composants, nous allons découper l'URL d'exemple suivante :

The diagram illustrates the structure of the URL `http://www.youtube.com/watch?v=QhcwLyyEjOA` by breaking it down into seven numbered components, each enclosed in a colored bracket:

- 1: `http://` (Protocol)
- 2: `www.youtube.com` (Domain)
- 3: `www` (Subdomain)
- 4: `youtube` (Domain)
- 5: `com` (TLD)
- 6: `/watch` (Path)
- 7: `?v=QhcwLyyEjOA` (Query string)

Structure et éléments de l'URL

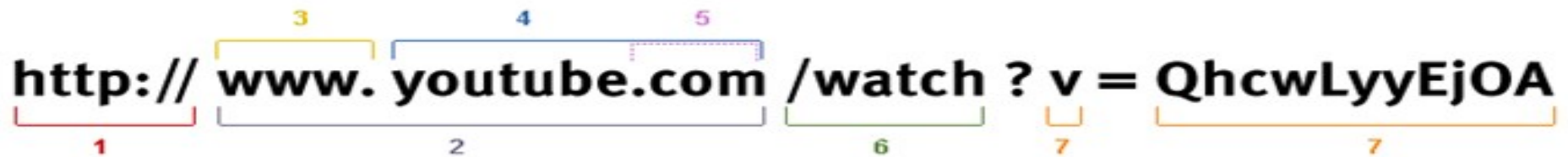
`http://www.youtube.com/watch?v=QhcwLyyEjOA`

La structure et les éléments d'une URL



1. Le *protocole* utilisé dans cet exemple :
HTTP (Hypertext Transfer Protocol)
(littéralement Protocole de transfert hypertexte)
 - Il existe d'autres protocoles comme HTTPS, FTP, etc.
2. Le *serveur* ou *nom de serveur* : *www.youtube.com*

La structure et les éléments d'une URL

The diagram shows the URL 'http://www.youtube.com/watch?v=QhcwLyyEjOA' with seven numbered components: 1. 'http://' (protocol), 2. 'www.' (sub-domain), 3. 'youtube.' (domain), 4. 'com' (top-level domain), 5. '/watch' (path), 6. '?' (query separator), 7. 'v=QhcwLyyEjOA' (query string).

http:// **www.** **youtube.com** **/watch** **?** **v = QhcwLyyEjOA**

3. Le *sous-domaine* : *www.*

4. Le *nom de domaine* : *youtube.com*

5. Le *domaine de premier niveau*

(un suffixe d'adresse web) : *.com*

(aussi référencé par l'abréviation TLD)

6. Le *chemin* : */watch*

- Un chemin fera généralement référence à un fichier ou dossier (répertoire) sur le serveur web (par exemple « /dossier/fichier.html »)

La structure et les éléments d'une URL

The diagram shows the URL `http://www.youtube.com/watch?v=QhcwLyyEjOA` with its components numbered 1 through 7. The components are: 1. `http://` (protocol), 2. `www.youtube.com` (domain), 3. `www` (subdomain), 4. `youtube` (main domain), 5. `.com` (top-level domain), 6. `/watch` (path), 7. `?v=QhcwLyyEjOA` (query string). The query string is further broken down into the parameter name `v` and the parameter value `QhcwLyyEjOA`.

7. Paramètre et valeur

: `v` (paramètre), `QhcwLyyEjOA` (valeur du paramètre)

- Les paramètres sont initialisés par le caractère « ? » dans l'URL.

Dans notre exemple, le nom du paramètre est « `v` » et sa valeur est « `QhcwLyyEjOA` » (les noms et valeurs des paramètres respectent toujours la même structure: `NomDuParametre=ValeurDuParametre`)

La structure et les éléments d'une URL

The diagram illustrates the structure of the URL `http://www.youtube.com/watch?v=QhcwLyyEjOA` with numbered components:

- 1: `http://` (Protocol)
- 2: `www.youtube.com` (Domain)
- 3: `www` (Subdomain)
- 4: `youtube` (Domain)
- 5: `com` (TLD)
- 6: `/watch` (Path)
- 7: `?v=` (Query string separator and parameter name)
- 7: `QhcwLyyEjOA` (Query string value)

- Les paramètres modifiant le contenu transmettent des informations à la page de destination. Ils doivent obligatoirement et exclusivement être définis dans l'URL finale.
- Par exemple, l'URL **`http://example.com?productid=1234`** permet de rediriger l'utilisateur vers la page de votre site Web concernant le produit 1234.

La structure et les éléments d'une URL

Qu'est ce qu'un nom de domaine ?

- Un **nom de domaine** est une composante de l'adresse d'un site Web (URL). Pour que quelqu'un puisse retrouver votre site Web,
- il lui suffit de connaître votre nom de domaine et de le saisir dans son navigateur Internet (Internet Explorer par exemple).
- Voyons de quoi est composée une adresse Internet (URL) complète:

La structure et les éléments d'une URL

URL d'exemple :	http://	www	.	kogik	.	com	/docs/	nom-domaine	.	php
Composantes :	protocole	machine	.	nom	.	extension	sous répertoire(s)	fichier	.	extension
				Nom de domaine				Nom de la page Web		

La structure et les éléments d'une URL

Qu'est ce qu'un nom de domaine ?

- Le nom de domaine est aussi utilisé pour fabriquer des adresses de courriel :

Courriel d'exemple :	<u>fducharme</u>	@	<u>io-2</u>	.	<u>com</u>
Composantes :	préfix variable	@	nom	.	extension
			Nom de domaine		

La structure et les éléments d'une URL

Éléments à connaître avant de choisir un nom de domaine :

- Choisissez un nom **facile à retenir pour vos visiteurs** :
 - Il devrait être **court** (moins de 15 caractères). Il n'y a pas de différence entre les majuscules et les minuscules.
 - De préférence, ne pas utiliser de **chiffres** et de symboles même si cela est techniquement possible.
- Si possible, optez pour un nom **qui décrit le contenu de votre site**.
- Séparez les mots qui composent votre nom de domaine par un trait-d'union « - ». Ceci améliorera la lisibilité et vos chances d'être bien référencé par les moteurs de recherche comme Google (ex. : hockey-bottine.com au lieu de hockeybottine.com).
- Choisissez une **extension adéquate**. Voici une liste partielle des extensions possibles et de leurs significations :

La structure et les éléments d'une URL

<u>.com</u>	Pour les sites à caractère commercial (commerce)
.net	Activité liée à Internet
<u>.org</u>	Site sans objet commercial (organisation)
<u>.edu</u>	Site en relation avec l'éducation
.info	Sites à vocation informationnelle
<u>.biz</u>	Commerce électronique (business)
.ca	Site canadien
.qc.ca	Site québécois
.us	Site américain

Site de notre université Dr Moulay TAHAR de Saida :

<https://www.univ-saida.dz/>

La structure et les éléments d'une URL

- *Techniquement, il n'y a pas de différence entre les extensions. Ceci dit, les internautes essaient souvent de deviner l'adresse d'un site Web en prenant l'extension «.com» parce qu'elle est la plus populaire.*
- Bas du formulaire
- Il est souvent utile de réserver **plusieurs noms de domaine** pour le même site Web. Par exemple, si vous choisissez l'adresse `www.votre-site.com`, il vaudrait mieux posséder également les adresses `www.votre-site.ca` et `www.votre-site.net`, etc. Ainsi vous éviterez qu'un concurrent mal intentionné achète un nom de domaine pouvant porter à confusion.

Plateforme pour créer des site web

- **Wix, WordPress** sont tous les deux d'excellents logiciels de construction de site.
- Des dizaines de millions de personnes utilisent ces plateformes. Pour réer des sites web.
- **Wix** : <https://fr.wordpress.com/>
- **WordPress** : <https://fr.wix.com/>
- **Blogger** : <https://www.blogger.com/>

Références bibliographiques

- [1] <https://www.geonov.fr/architecture-client-serveur/>
- [2] <https://aws.amazon.com/fr/what-is/computer-networking/>
- [3] <https://www.universalis.fr/encyclopedie/internet-histoire/>
- [4] <https://www.youtube.com/watch?v=JqweQE6TDTc>
- [5] [https://www.researchgate.net/figure/Architecture-du-reseau-existant-13-Explication-technique-du-reseau-Larchitecture fig1 351123585](https://www.researchgate.net/figure/Architecture-du-reseau-existant-13-Explication-technique-du-reseau-Larchitecture_fig1_351123585)
- [6] <https://web.maths.unsw.edu.au/~lafaye/CCM/initiation/3tier.htm#:~:text=L'architecture%20%C3%A0%20deux%20niveaux,en%20utilisant%20ses%20propres%20ressources.>
- [7] <https://blog.hubspot.fr/website/serveur-informatique>
- [8] http://projet.eu.org/pedago/sin/ISN/8-client_serveur.pdf
- [9] https://fr.wikipedia.org/wiki/Protocole_de_communication
- [10] <https://blog-fr.orson.io/creation-de-site-internet/definition-url>