



# FORMATION

IT - Digital - Management



[m2iinformation.fr](https://m2iinformation.fr)



# Architectures Applicatives

Première partie

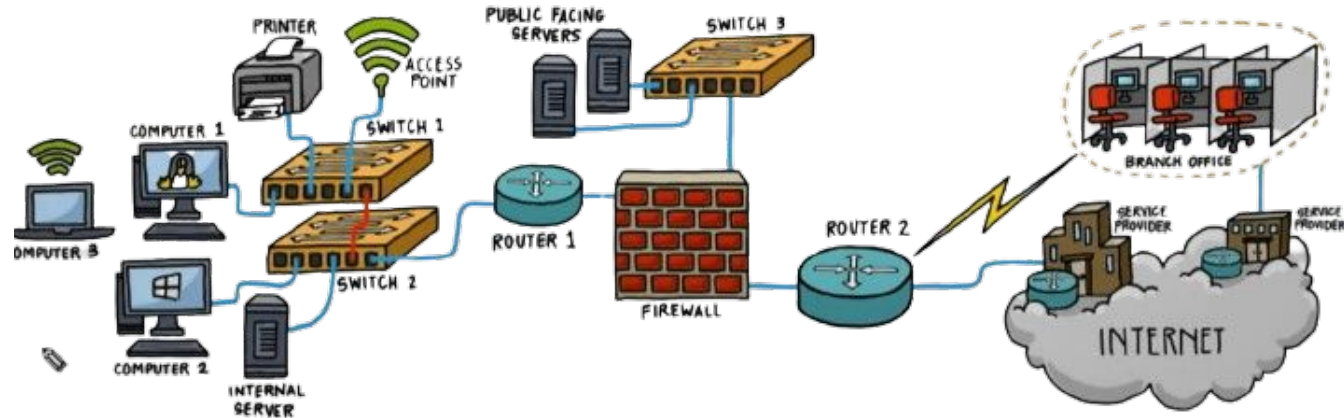
Christian LISANGOLA  
Software Engineer



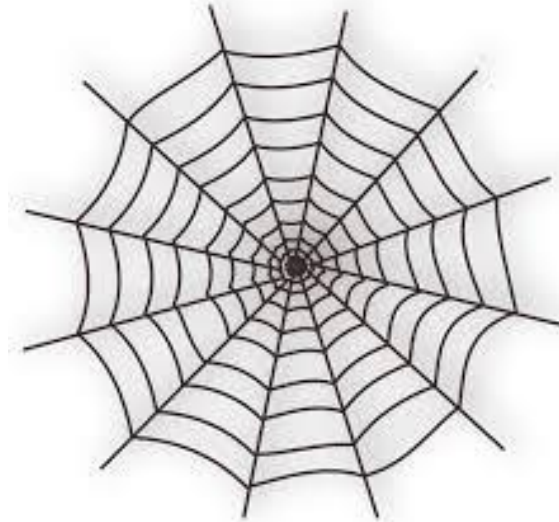
# 1.

## C'est quoi le WEB?

# Accéder à Internet



# Le WEB



Il peut être vu comme une toile d'araignée géante reliant des page web entre elles au travers des liens hypertextes(Liens qui mènent vers d'autres pages).

Surfer sur internet est le fait de naviguer de page en page comme un surfeur passerait d'une vague à une autre.



# Le WEB = Internet??

Le web étant le service le plus utilisé du réseau internet amène souvent à croire que le WEB=Internet.

L'internet est simplement un réseau permettant de communiquer entre les ordinateurs à l'échelle mondiale et cela grâce à plusieurs types de services :

- **Le Web** : Auquel on accède grâce aux navigateurs internet
- La **messagerie électronique** pour échanger des e-mails
- **Le Service FTP** pour l'échange des fichiers
- Etc.

Une autre raison qui amène à croire que le web et l'internet c'est la même chose est le fait qu'il soit aujourd'hui utilisé comme l'interface pour la plupart d'autres services comme par exemple Gmail la messagerie.

Auparavant il fallait utiliser des logiciels dédiés pour chacun de ces services (Outlook ou Thunderbird pour la messagerie électronique).



# 2.

## Architecture

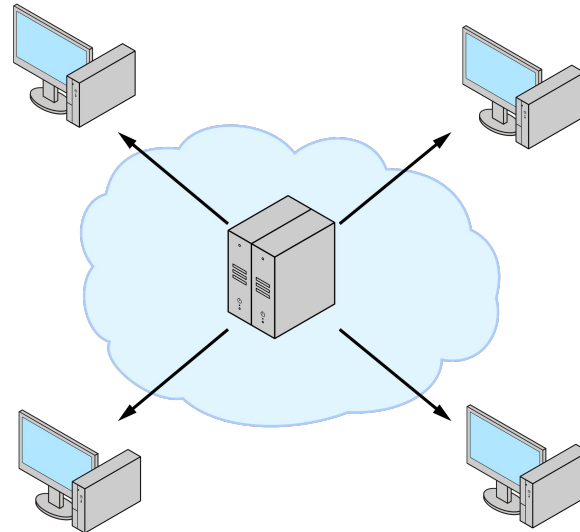
# Architecture client-serveur

Le Web est basé sur une architecture réseau appelé le client-server, une architecture qui consiste à faire dialoguer un programme demandeur de service (appelé client) avec un programme fournisseur de service appelé serveur. La communication entre les 2 entités (client et serveur) se fait au travers du réseau.

Ici les services peuvent être :

- Des fichiers
- La messagerie
- Les pages web
- Etc.

Cette architecture n'est pas spécifique au web





# Architecture client-serveur(suite)

Quand on parle du client et du serveur il peut s'agir des 2 choses :

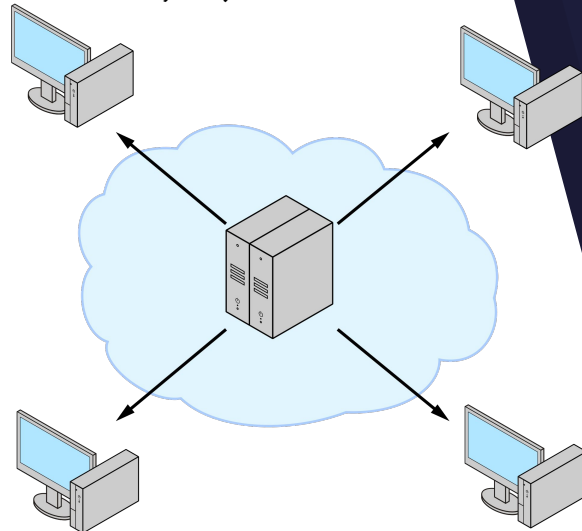
Pour le client il s'agit des :

- ordinateurs(smartphones, montre intelligents,etc..)
- applications(Navigateurs, applications mobiles, Antivirus,etc)

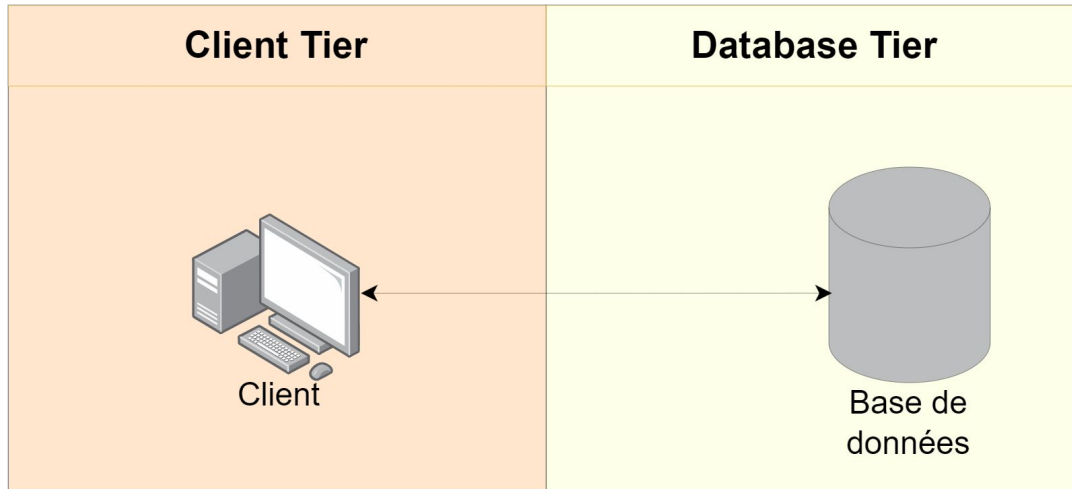
Le client n'est donc pas l'humain.

Pour le serveur il s'agit des :

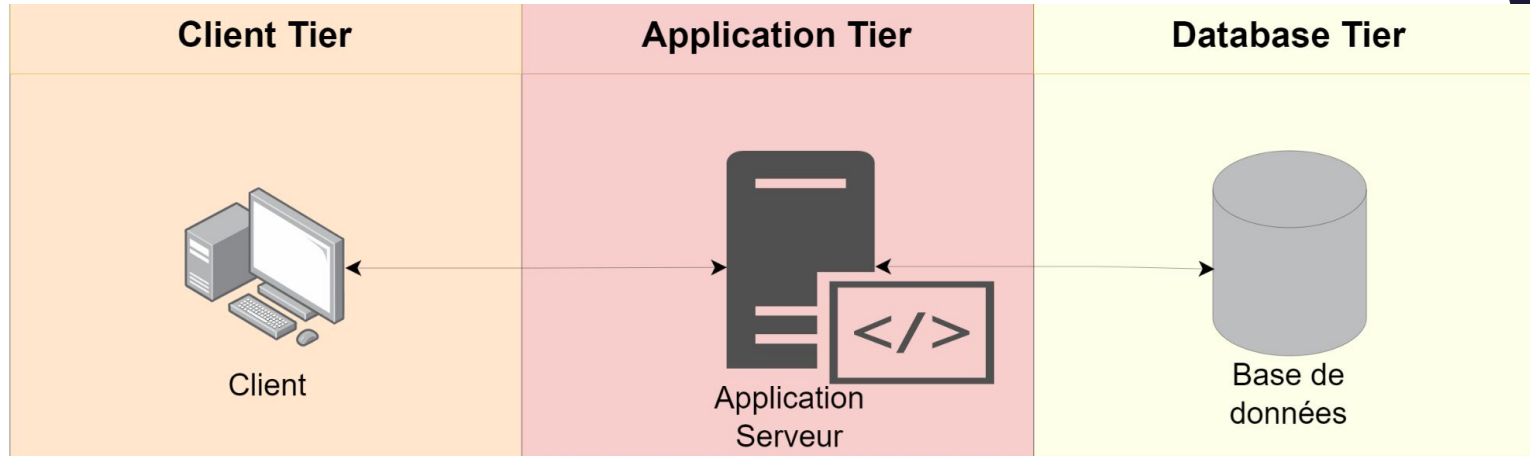
- Ordinateur avec une config particulière
- Application(Apache, Nginx, SMTP,etc.)



# Architecture client-serveur 2-tier



# Architecture client-serveur 3-tier





# Requêtes/ Réponses

La demande que le client effectue auprès du serveur via le réseau est appelée **Requête**, le mécanisme utilisé par le serveur pour satisfaire à la demande du client est appelé **Réponse**.



# 3.

## IP/DNS



# Adresse IP & DNS

Example: 142.251.47.174

Pour que 2 ordinateurs puissent communiquer en réseau ils doivent être en mesure de s'identifier.

En réseau informatique les ordinateurs sont identifiés par leurs adresse IP.

Comment est-ce qu'un client arrive à identifier l'adresse IP d'un ordinateur sur internet en sachant que dans la barre d'url ce qu'on écrit est l'url du site, donc son nom de domaine?

**Le DNS.**

Example: google.com



# 4.

## Protocoles



# C'est quoi un protocole

Avant de définir un protocole, il est important de connaître les 3 règles pour qu'une communication soit possible:

- L'existence d'un émetteur et d'un récepteur
- Un canal de communication(media)
- Un langage commun

En ce qui concerne le dernier point(langage), plusieurs choses doivent ensuite être prise en compte s'il s'agit d'une communication entre des humains(même langue, une voix audible, une bonne gestion du débit avec lequel on prononce les mots,etc...).

L'ensemble de ces regles constituent un protocole, car sans elles la communication n'est pas possible.





# C'est quoi un protocole

Dans le réseau, 2 ordinateurs ne peuvent communiquer que s'il s'accorde sur un protocole commun pour les échanges.

Il existe donc plusieurs protocoles selon le type de services utilisé sur internet:

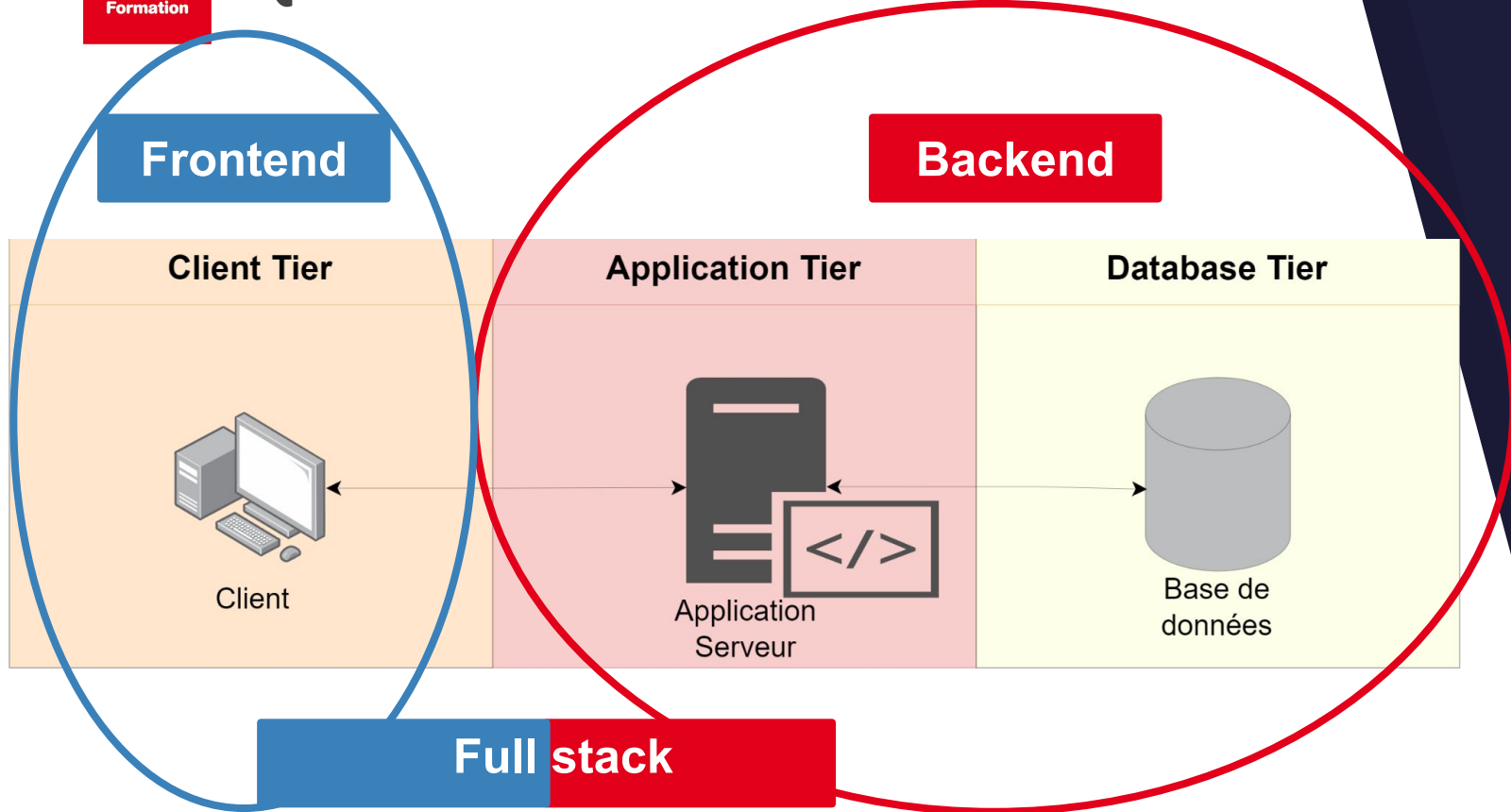
- **HTTP** : Décrit l'ensemble des règles qu'un client HTTP(navigateur web, application mobile,etc..) doit respecter pour communiquer avec un serveur WEB.
- **SMTP** : C'est le protocole utilisé pour le transfert de messages électroniques
- **IMAP/POP** : En informatique, le POP, est un protocole qui permet de récupérer les courriers électroniques situés sur un serveur de messagerie électronique
- **FTP** : Il permet, depuis un ordinateur, de copier des fichiers vers un autre ordinateur, ou encore de supprimer ou de modifier des fichiers sur cet ordinateur.
- **SSH** : Secure Shell (SSH) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés



# 5.

## Métiers du web

# Quels sont les métiers du WEB





# Métiers du WEB

Il existe plusieurs métiers dans le web, mais les compétences principales sont :

- Front-end
- Le Backend



# Développeur Frontend

Le développeur Frontend est responsable du développement des interfaces utilisateur d'une application web ou mobile:

- Les langages du Frontend pour le développement web sont : HTML, CSS et Javascript(+Frameworks comme Angular, React.js ou Vue.js). Ces langages sont appelés des langages clients, parce qu'il sont compris du navigateur Web.
- Les technologies du Frontend pour le développement Mobile sont Java(Android SDK, Flutter, React Native, Swift, Xamarin,etc..).



# Développeur Backend

Le développeur Backend est responsable du développement des applications serveur ou API:

Les langages et technologies du Backend pour le développement web sont : PHP, Java, Python(Django, Flask),Ruby(Ruby on Rails),C#(Asp),Node.js(basé sur Javascript).

A part les langages de programmation du Backend, il y a aussi une couche responsable du stockage des données que sont les SGBD(Système de gestion des bases de données).



# Frameworks

En ce qui concerne le front-end et le backend, il existe des outils appelés “Framework” que les développeurs peuvent utiliser pour être plus productif et développer plus rapidement.

Ces frameworks donnent plusieurs avantages aux développeurs:

- Architecture du code
- Brique de base
- Bonnes pratiques



# API

Le backend sera principalement utilisé pour développer l'API afin que celui-ci puisse servir n'importe quel type de client.

Les clients de leurs côtés n'ont pas besoin de connaître le langage utilisé pour créer l'API parce qu'il suffira de fournir au développeur frontend une documentation sur comment communiquer avec l'API sans qu'il ne se soucie des détails d'implémentation ou langages utilisés.