{ Architecture du Web }

I

Histoire d’internet et du Web

Le réseau Internet

Internet est une interconnexion de réseaux à l'échelle mondiale.

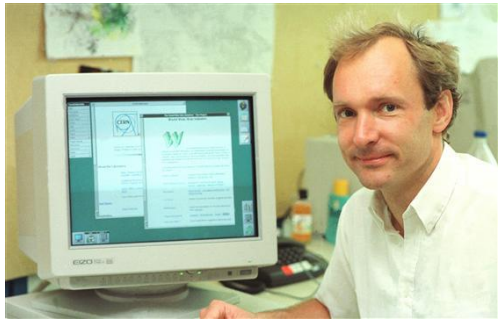
Internet fut inventé dans les années 60 et financé initialement par l'armée américaine.

Le protocole TCP/IP v4 arrive en 1981.

Le World Wide Web voit le jour 10 ans plus tard.

L'invention du Web

Le World Wide Web fut inventé au début des années 90 par **Tim Berners-Lee** au CERN.

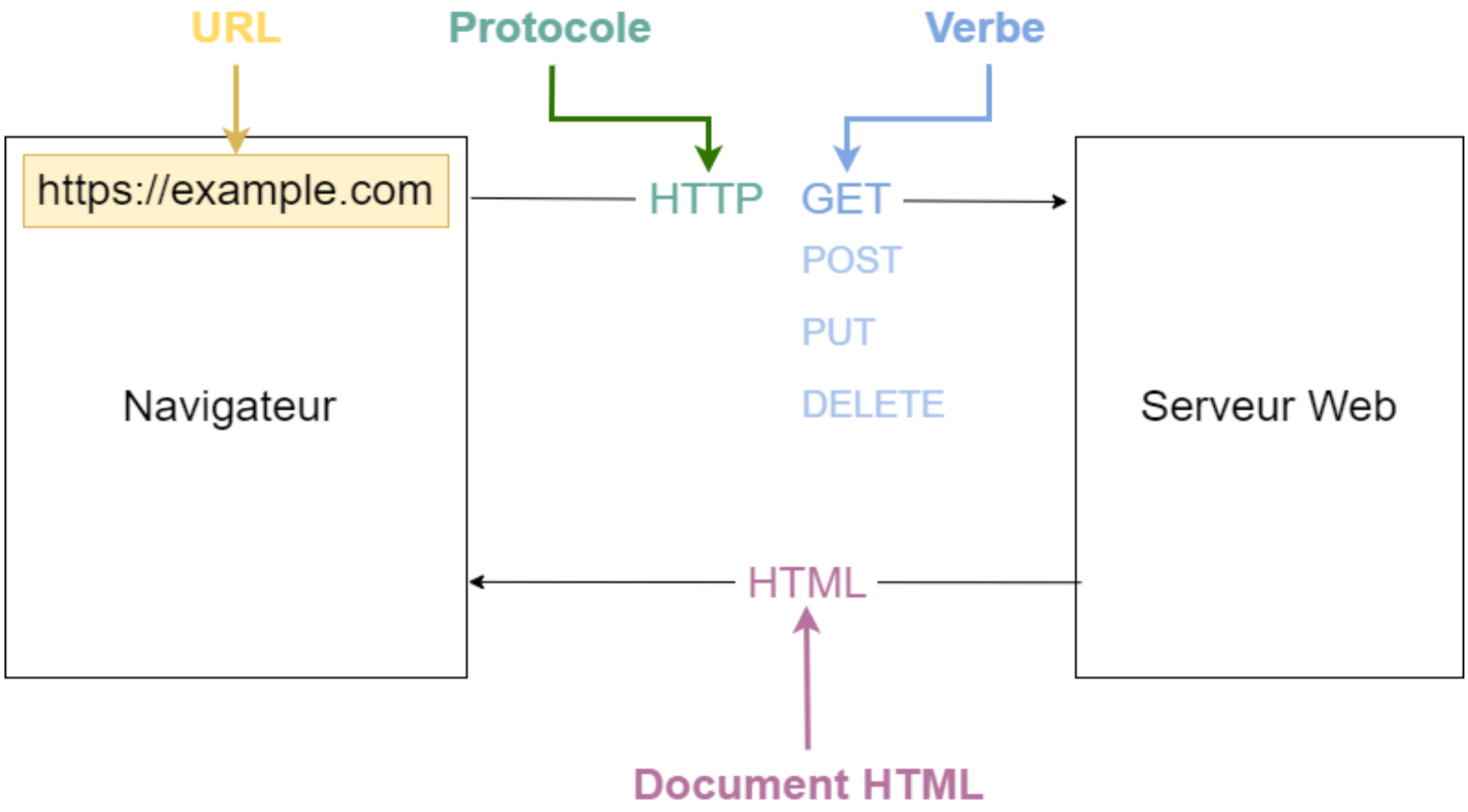


CERN : Conseil Européen pour la Recherche Nucléaire.

Le Web avait pour but de faciliter le partage des connaissances entre les chercheurs du CERN.

Le **30 Avril 1993** le CERN renonce à ses droits d'auteurs : le Web tombe dans le domaine public.

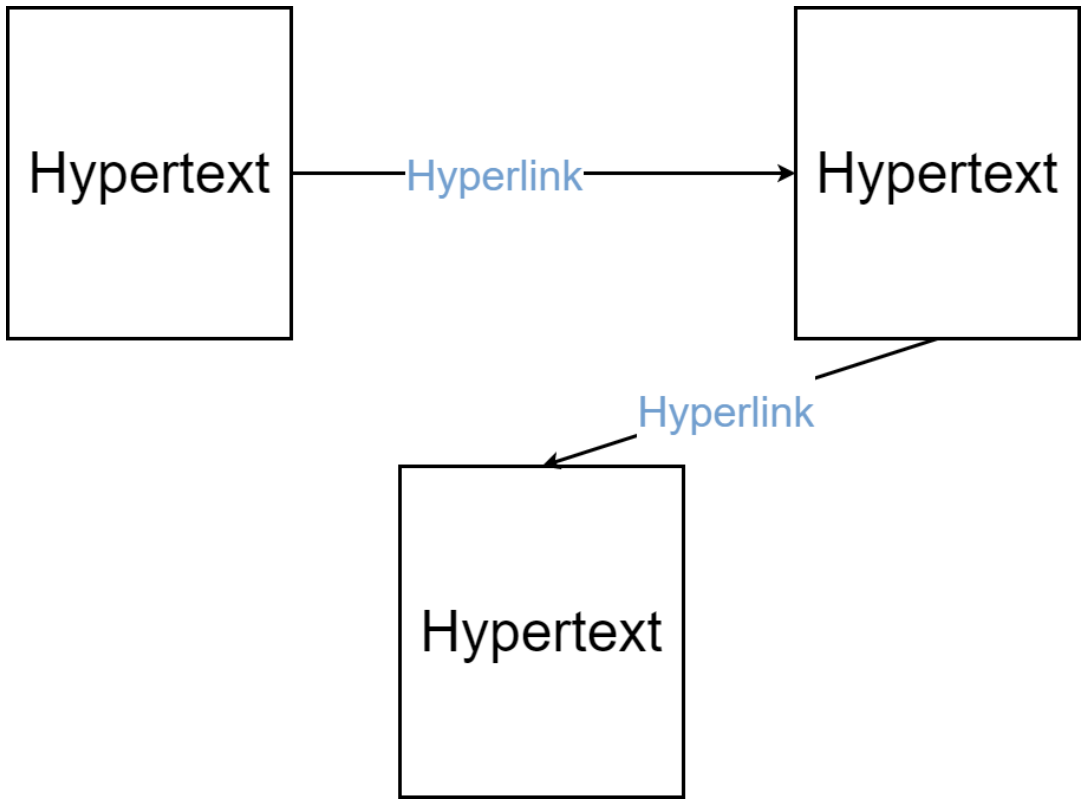
Fonctionnement du Web



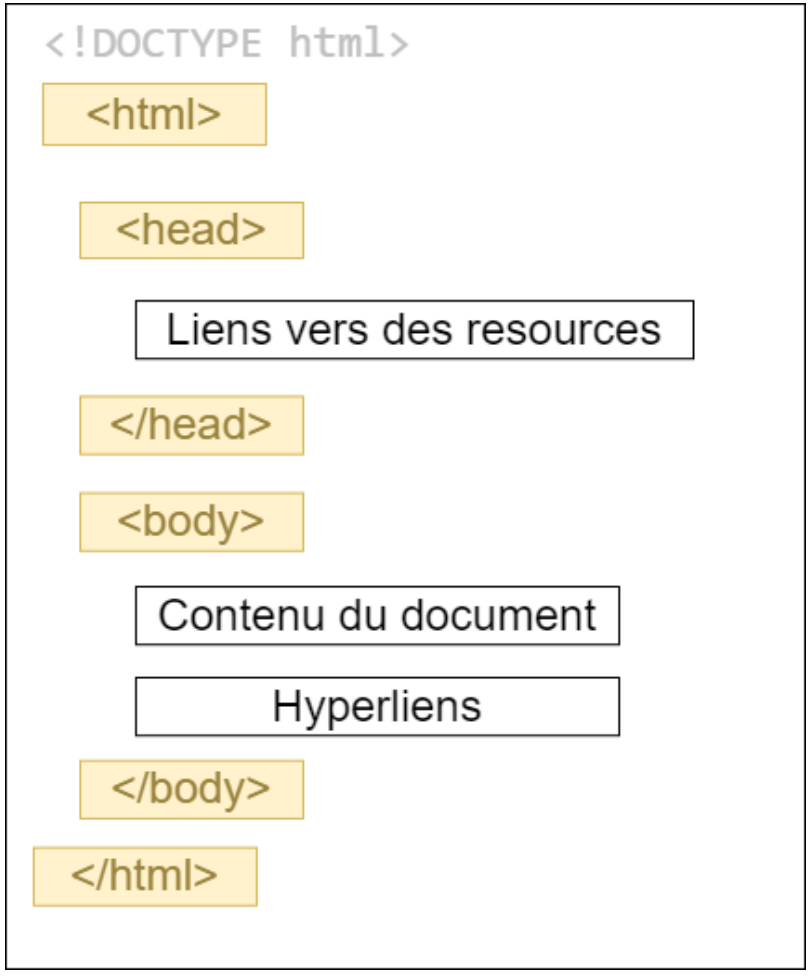
Concept de l'Hypertexte

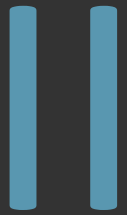
L'**Hypertexte** (*hypertext*) est du texte contenant des liens vers d'autres textes.

Les liens sont appelés **hyperliens** (*hyperlink*).



Composition d'un page Web





Evolution du Web

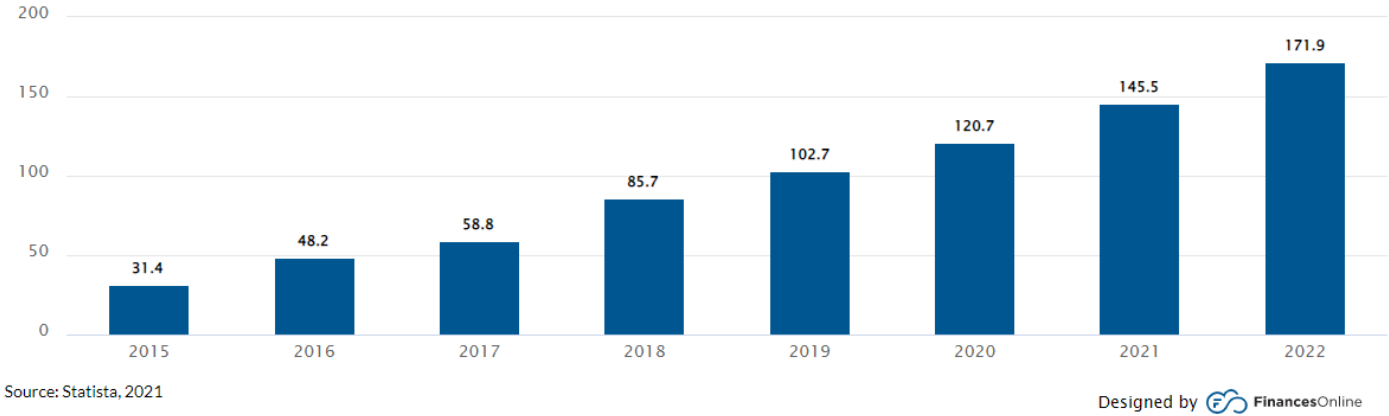
Le Web en statistiques

- **5 Milliards** de personne ont accès à internet
- **130 000 Milliards** de pages indexés
- **175 Zettaoctets** de données stockées

L'évolution du Web

- Les clients lourds ont laissé place aux SaaS au fil des années
- Un **client lourd** est un logiciel nécessitant une installation sur la machine de l'utilisateur
- **SaaS** pour **S**oftware **a**s **a** **S**ervice, désigne tout logiciel accessible simplement depuis un navigateur Web.

Milliard de dollars dépensés dans les logiciels SaaS entre 2015 et 2022



L'évolution du Web

Le passage au SaaS implique de :

- Centraliser le code applicatif sur un serveur distant
- Séparer l'application client du code serveur
- Connecter l'application client au serveur
- Gérer les requêtes concurrentes
- Passer à une architecture **stateless** côté serveur dans la majorité des cas

L'évolution du protocole HTTP

HTTP (Hypertext Transfer Protocol) est protocole réseau applicatif par dessus TCP/IP.

Permet d'échanger des ressources du serveur vers le client.

HTTP 1 (1996) : Une connexion TCP par requête HTTP.

HTTP 1.1 (1997): Une même connexion peut être réutilisée pour demander plusieurs ressources.

HTTP 2 (2015): Une seule connexion TCP pour toute les requêtes.

HTTP 3 (2022) : Utilisation du protocole UDP pour améliorer les performances.

Le protocole WebSocket

Websocket est un protocole introduit en 2011 utilisant une connexion TCP et permettant une communication **bidirectionnelle** entre le client et le serveur.

Une fois établie, la connexion WebSocket permet au serveur d'envoyer de la donnée au client en temps réel et vice versa.

La connexion WebSocket est initiée via une requête HTTP durant la phase de **handshake.**

C'est durant la phase de **handshake** que le client va basculer d'une connexion HTTP à une connexion WebSocket.

Le WebRTC

WebRTC (Web Real-Time Communication) est une suite de protocoles permettant d'établir une communication directe entre les clients (peer-to-peer).

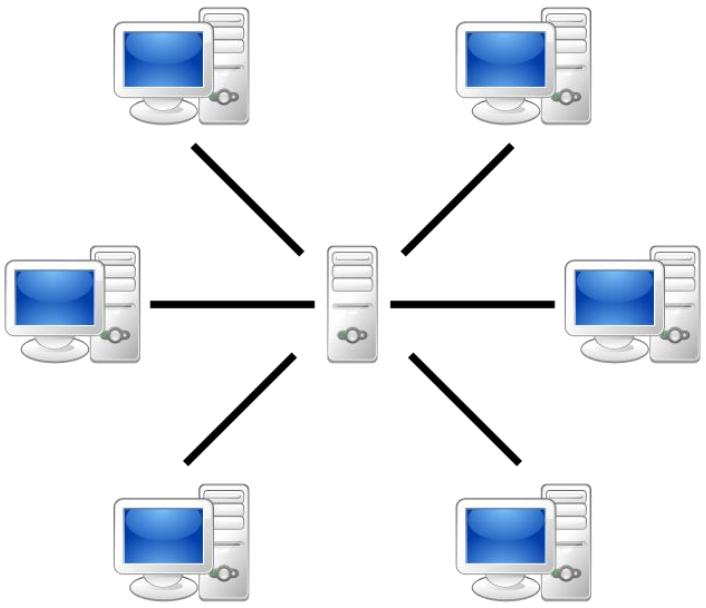
Le but est de simplifier la communication client-client en permettant l'échange de données sans passer par un serveur.

Un serveur est toutefois nécessaire pour mettre en relation les clients cherchant à s'échanger de la donnée.

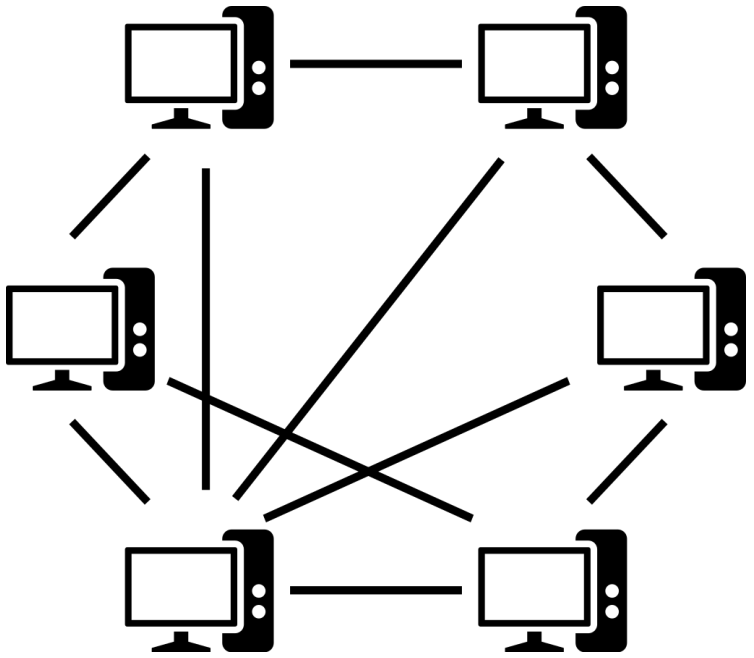
WebRTC permet par exemple de streamer de l'audio ou de la vidéo ou d'envoyer des fichiers.

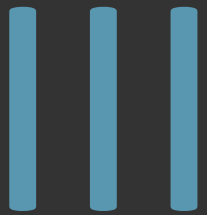
Le WebRTC

Client - Serveur



Pair à pair (peer-to-peer)





Navigateur Web

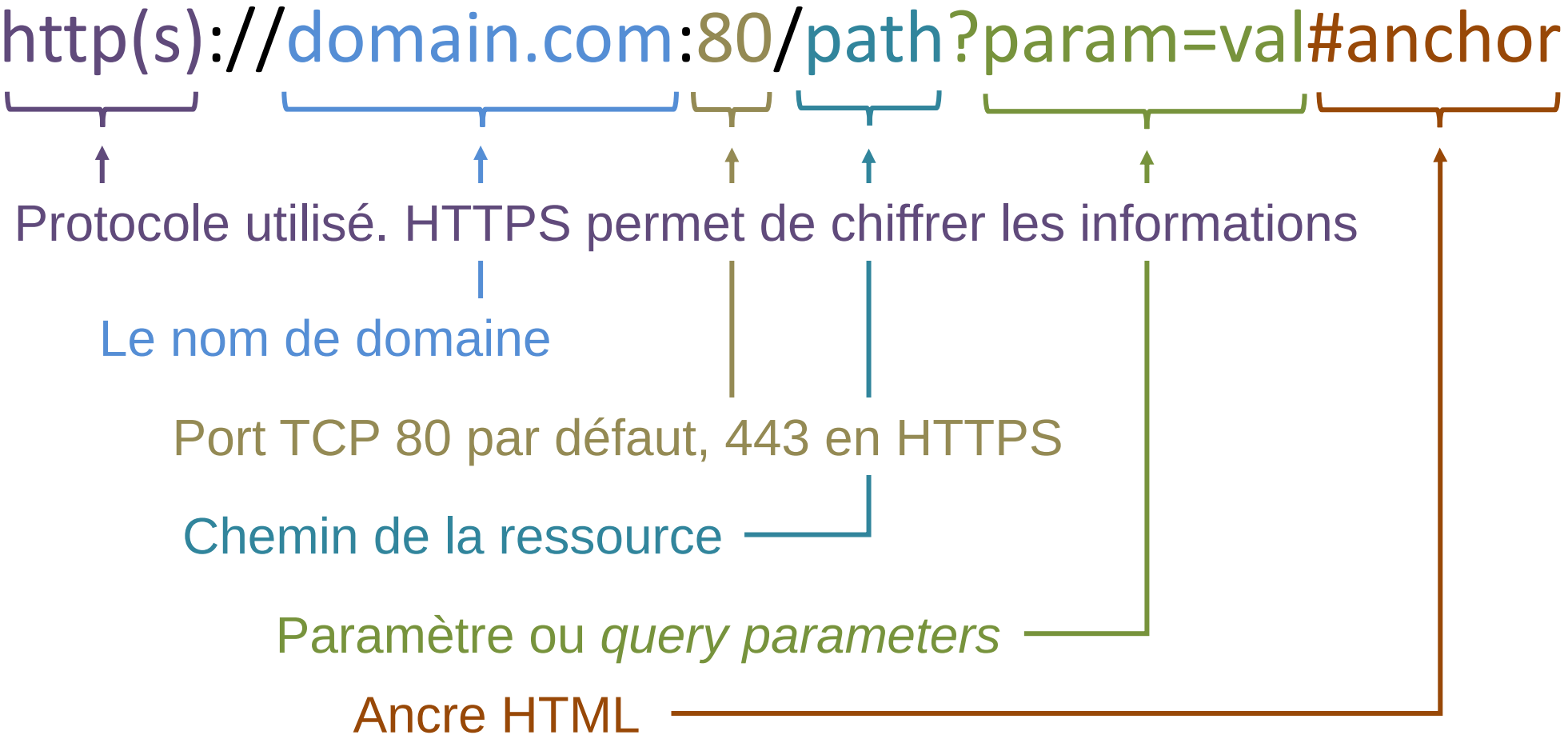
Le Navigateur

Le navigateur ou *browser* est le logiciel qui permet de naviguer sur internet.

Le navigateur va donc :

- gérer la connexion au serveur
- télécharger la page HTML avec les ressources nécessaires à son affichage
- exécuter les scripts
- dessiner le contenu de la page

Anatomie d'une URL



Résolution de l'URL

Internet utilise le protocole TCP/IP pour connecter votre machine au serveur.

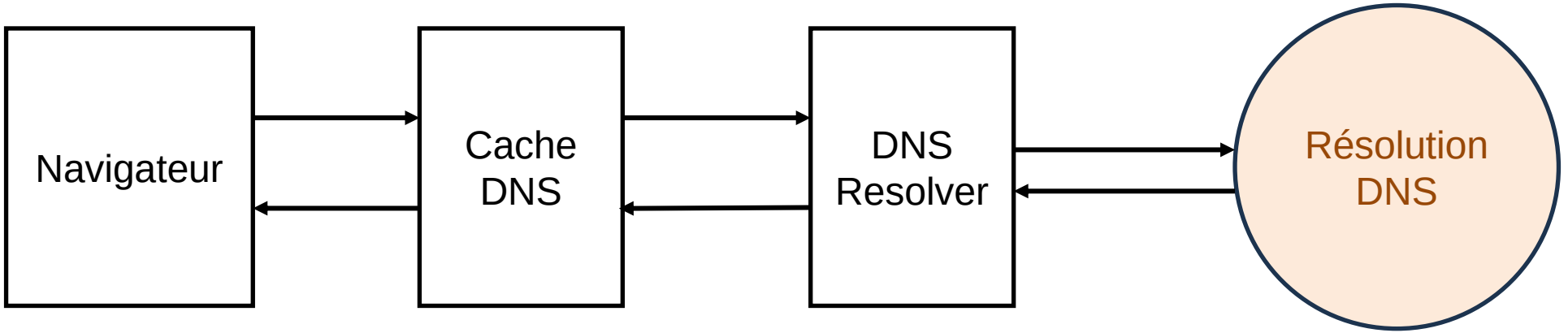
Votre navigateur va donc devoir trouver l'adresse IP du serveur avant de pouvoir ouvrir une requête HTTP.

L'URL (**U**niform **R**esource **L**ocator) est un alias vers l'adresse IP d'une machine.

Le service DNS (**D**omain **N**ame **L**ocator) est un registre qui fournit une adresse IP à partir d'une URL.

Le DNS Lookup

Le processus permettant de trouver une adresse IP à partir d'une URL s'appelle **DNS Lookup**



Le moteur de rendu

De façon à restituer un document HTML à l'écran, les navigateurs s'appuient sur un **moteur de rendu** ou *browser engine.*

La plus part des navigateurs utilisent Webkit développé par Apple.

Chrome utilise Blink qui est un *fork* de Webkit.

Firefox utilise Gecko et est un des rares navigateurs non basé sur Webkit.

Le moteur de rendu est également responsable de la construction du **DOM**.

Le moteur JavaScript

Enfin, le moteur JavaScript est un interpréteur JavaScript généralement développé à part et intégré aux navigateurs.

Ils contiennent un certain nombre d'optimisations pour permettre au JavaScript de conserver des performances optimales.

V8 est le moteur développé par Google très largement répandu car utilisé dans Chromium (Chrome, Opéra, Edge) et par Nodejs.

JavaScriptCore est utilisé dans les produits Apple et par le projet Bun.

Firefox utilise SpiderMonkey.

IV

Architecture du Web

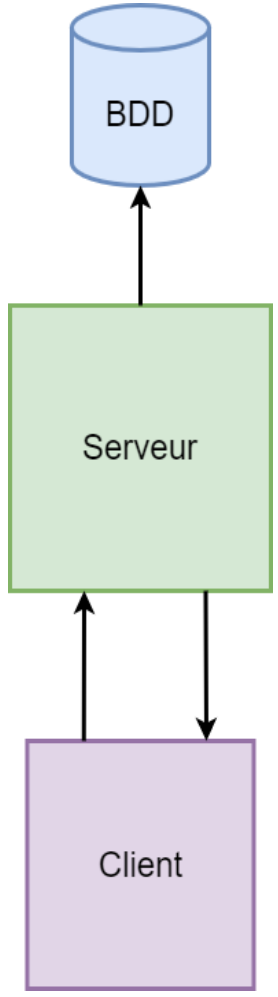
L'architecture 3 tiers

L'architecture 3 tiers est composée d'un client, d'un serveur et d'une base de donnée.

Le **client** est responsable de l'affichage du contenu et des interactions avec l'utilisateur.

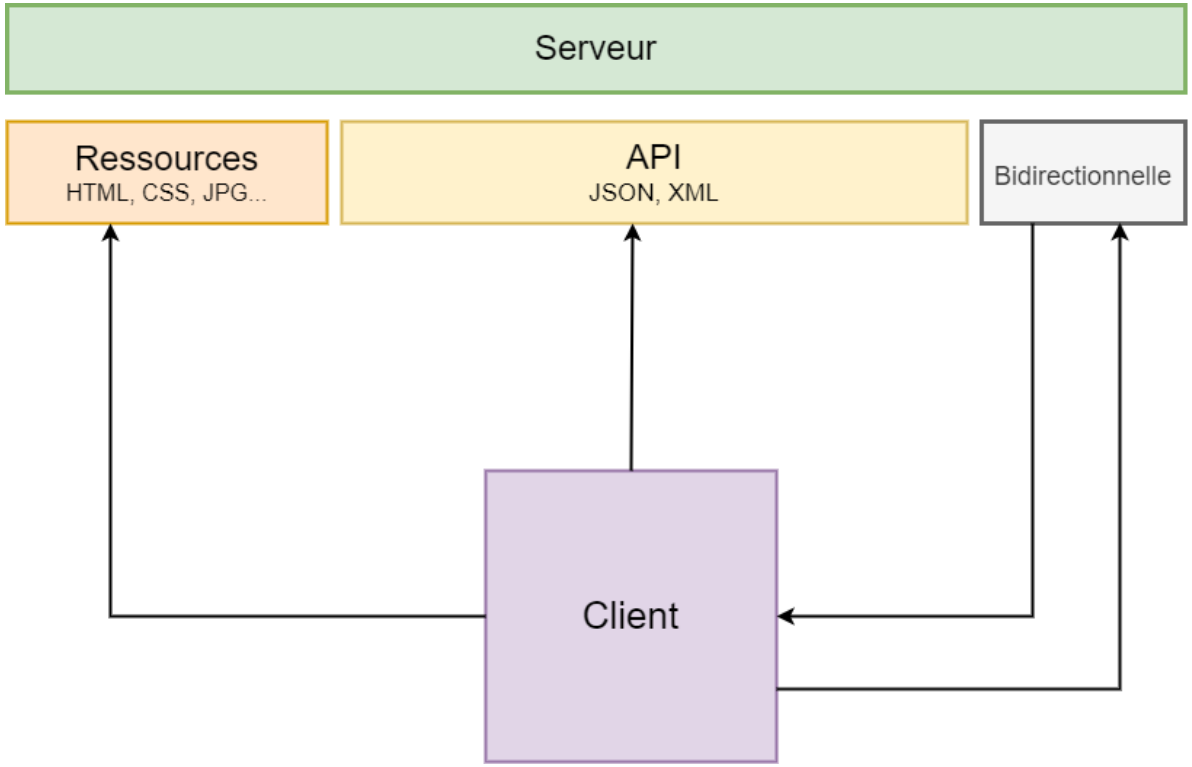
Le **serveur** fournit des ressources et de la donnée au client et lui permet d'interagir avec la **couche métier**.

La **base de donnée** stocke et index la donnée accessible au serveur.



API

Le client communique avec le serveur en passant généralement par une API (Application Programming Interface) exposée par le serveur.



API REST

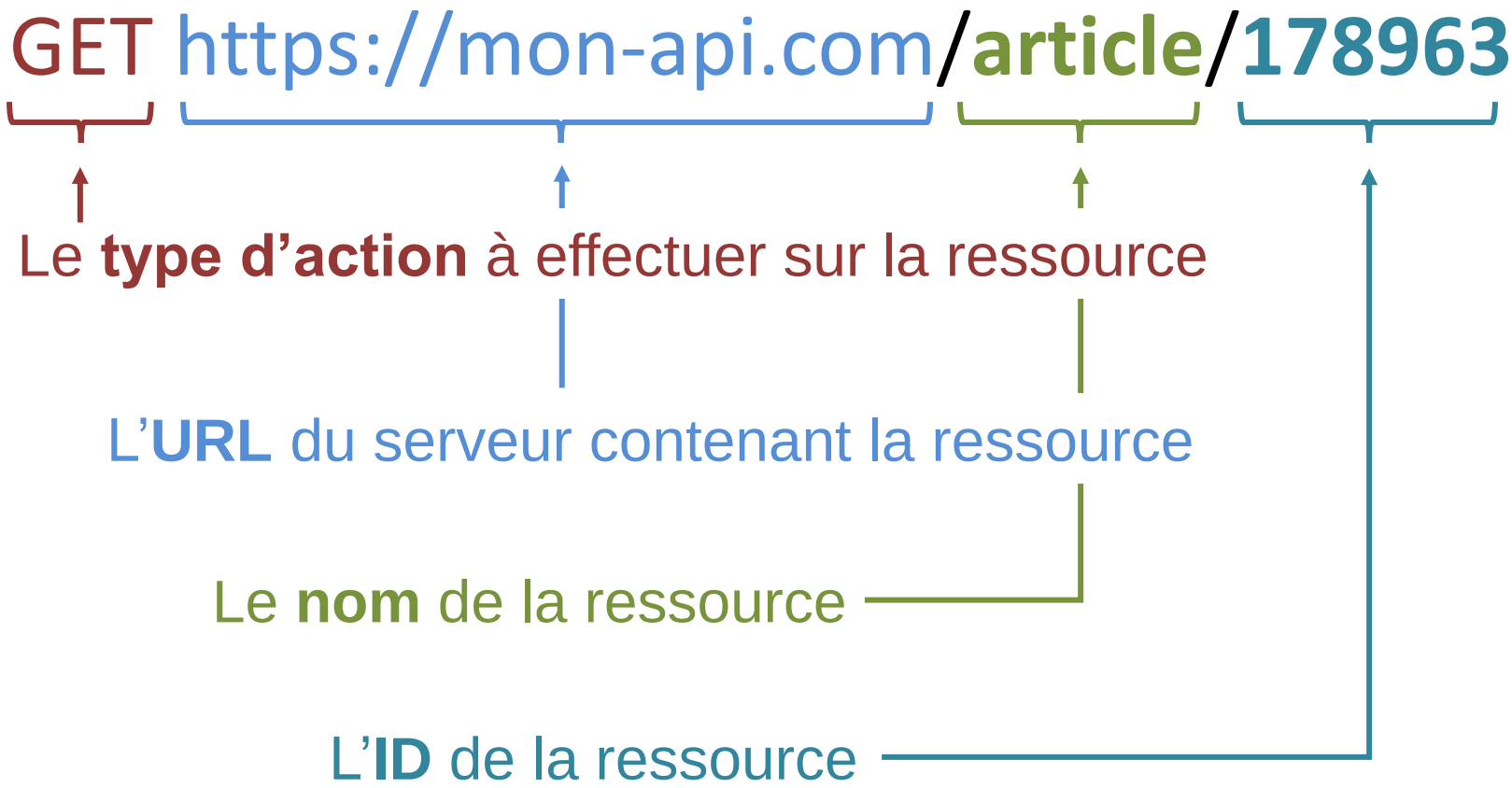
Par convention la plus part de communications se font avec une API REST.

L'API **REST** (Representational State Transfer) est une architecture par-dessus HTTP qui décrit comment le client et le serveur doivent s'échanger de l'information.

La donnée retournée par le serveur est qualifiée de **ressource.**

La ressource est présentée sous forme de JSON, de XML voir de HTML.

API REST Exemple



API REST Verbes HTTP

Type d’actions possibles sur une ressource :

GET : récupérer la ressource

POST : Créer une nouvelle ressource

PUT : Modifier toute la ressource

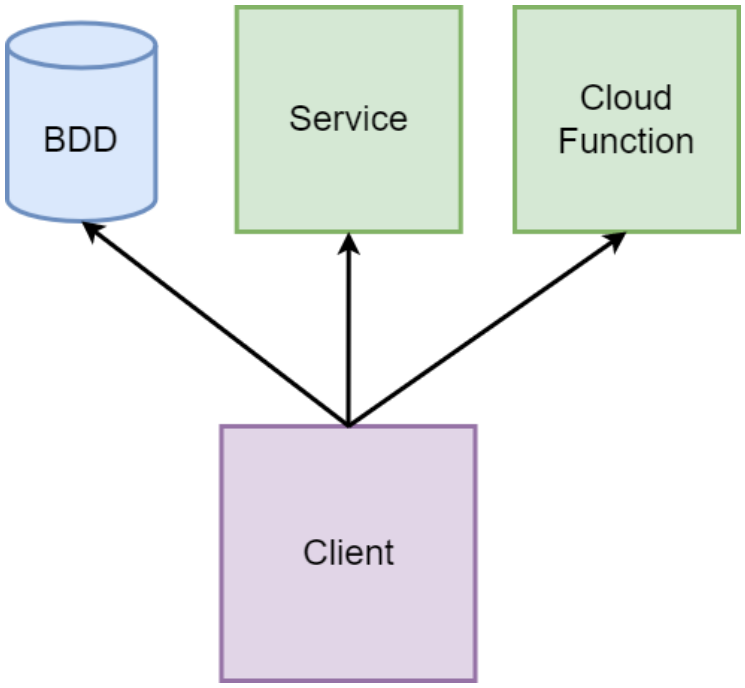
PATCH : Modifier une partie de la ressource

DELETE : Supprimer la ressource

L'architecture Serverless

Dans une architecture severless, le client ne communique pas avec un serveur dédié.

Le client va solliciter divers API généralement hébergée dans une infrastructure Cloud



V

Application Web

Web App vs Website

Un site web est une collection de pages HTML accessibles via un navigateur, qui sont hébergées ou générées par un serveur.

L'application Web ou *Web App* est un programme hébergé sur un serveur Web et accessible via un navigateur web.
Le HTML, CSS et JavaScript sont utilisés pour créer l'interface.

Le site web à pour but de diffuser du contenu là ou la *Web App* donne accès à un programme.

Pourquoi l'application web

Les *Web Apps* ont progressivement remplacées les clients lourds.
Avantages :

- Pas d'installation nécessaire
- Facile à déployer et à mettre à jour
- Multiplateforme : PC, Mac, Smartphones, TV…

Inconvénients :

- Moins performante qu'une application native
- Ne peut pas directement interagir avec l'OS
- Généralement dépendant d'internet

SPA

En général, les applications web n'ont besoin que d'une seule page HTML pour fonctionner. On les appelle des **SPA** (**S**ingle **P**age **A**pplication).

La navigation entre les différentes vues au sein d'une SPA est déléguée au JavaScript.

Le *routes* ne sont donc pas gérées par le serveur mais par le client.

C'est pourquoi une SPA peut être hébergée sur un **CDN** (**C**ontent **D**elivery **N**etwork).

CDN

CDN signifie **C**ontent **D**elivery **N**etwork ou réseau de diffusion de contenu en français.

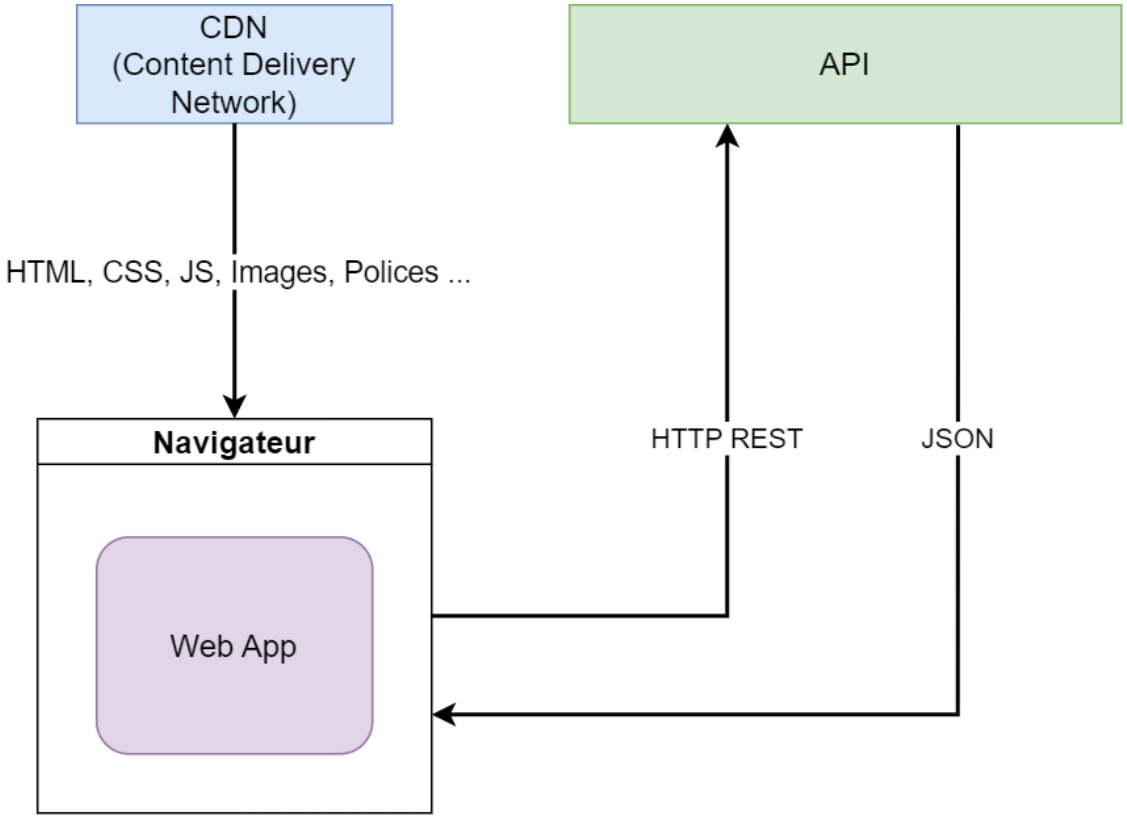
C’est un ensemble de serveurs optimisés pour servir des ressources statiques comme du HTML, JavaScript, CSS, Polices etc.

Pour garantir de meilleures performances, les serveurs vont conserver les fichier en mémoire plutôt que de les lire depuis le *file system* à chaque requête.

Chaque serveur est responsable de servir une région du globe, de façon a garantir les meilleurs temps de réponses.

Architecture de la Web App

Architecture web classique d'une SPA



Le SSR

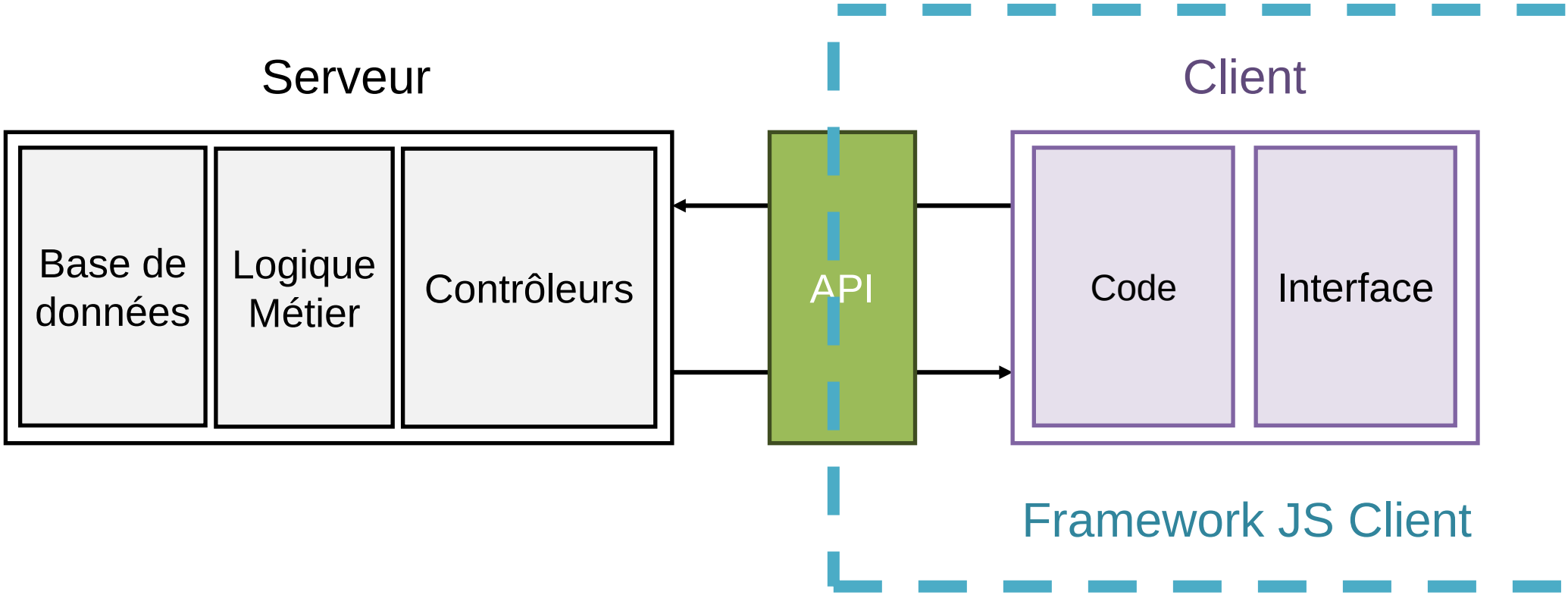
L’usage d’une SPA peut poser quelques problèmes :

- Le client doit potentiellement exécuter beaucoup de JavaScript
- La SPA est incompatible avec le référencement (SEO) : les pages ne peuvent pas être référencées.

Le **S**erver **S**ide **R**endering est une technique qui permet au serveur d’exécuter une partie du code JavaScript de l’application Web.

Le serveur va donc servir des pages HTML générées via du JavaScript.

Le SSR


Serveur
Client
Base de données
Logique Métier
Contrôleurs
API
Code
Interface
Framework JS Client

SSR