

Monographie - Web temps réel

Cartouche

- Code du sujet : 20W
- Code de l'alliance : 2020A20W
- Liste des équipes
 - E03
 - * Boulier, Florent (FloBouboule)
 - * Lecordier, Jean (jlecordier)
 - * Picard, Lisa (lisapicard)

Partie A

A0 : Introduction

La technologie que nous allons étudier dans ce projet est celle du **web temps réel**. En effet, de nombreuses applications utilisent aujourd'hui cette technologie pour mettre à jour un écran sans qu'aucune interaction utilisateur ne l'ait demandé. Grâce à cette technologie, l'internaute n'est plus en attente d'information, il la reçoit lorsqu'elle est disponible. Cela permet notamment le travail collaboratif en temps réel et permet une meilleure interaction avec l'utilisateur.

Les informations reçues par l'utilisateur peuvent être diverses : Des messages courts, des messages de type "alerte", des liens... On peut par exemple citer les statuts Facebook, les recherches Google ou les tweets.

Il existe aujourd'hui plusieurs techniques pour implémenter du web en temps réel, comme les WebSockets de HTML5, XMPP ou encore PubSubHubbub. La plupart de ces techniques utilisent le pattern ajax-reverse, explicité dans la partie A7 références théoriques.

- HTML5 est un standard des technologies web, qui apporte des API et des protocoles permettant d'implémenter le web temps réel. Les WebSockets sont un exemple de ces fonctionnalités : Ils permettent une communication bidirectionnelle entre le serveur et le client.
- XMPP est un ensemble de protocoles standards pour la messagerie instantanée.
- PubSubHubbub est un protocole ouvert qui fournit une notification en temps réel des mises à jour.

L'objectif de ce projet est de déterminer la ou les techniques les plus adaptées permettant la mise en place d'un système de web temps réel. Pour ce faire, l'alliance étudiera les 3 techniques à l'état de l'art proposées puis les comparera en mesurant certains indicateurs. Cette alliance ne comprenant qu'une équipe, une seule technique sera mesurée. Dans ce but, l'équipe devra appliquer la technique sur une application web permettant à des anciens ASI répartis sur plusieurs sites distants de passer en revue une liste de tâches, pour organiser une manifestation comme la journée des métiers.

Le cas d'utilisation est le suivant :

- > Un utilisateur maître M0 se connecte à l'application web A depuis un navigateur 1 et visualise une liste de tâches (vide).
- > Il saisit 3 tâches T1, T2, T3.
- > L'utilisateur M démarre la conférence.
- > Deux utilisateurs observateurs O1 et O2 se connectent à l'application web A depuis les navigateurs 2 et 3.
- > Ils visualisent la liste des tâches.
- > L'utilisateur M surligne la tâche T1.
- > Les deux utilisateurs O1 et O2 voient la ligne surlignée sans qu'ils n'aient effectué aucune intervention.
- > O1 saisit un commentaire.
- > Les deux autres utilisateurs M0 et O2 voient le commentaire.

A1. Glossaire et Mots-clés

- **Application Web** : Une application hébergée sur un serveur et accessible via un navigateur Web. Elle ne nécessite pas d'installation sur la machine cliente.
- **Temps réel** : En direct, au moment où les choses se passent.
- **Interaction utilisateur** : Action réciproque entre l'utilisateur et le système (ici une page web).
- **Navigateur** : Logiciel conçu pour naviguer sur le Web.
- **Rafraîchissement** : Fait d'actualiser/de recharger une page web pour avoir sa version la plus récente.
- **Liste de tâches** : Outil de gestion de temps permettant de donner une priorité aux tâches à faire.
- **ajax-reverse** : Pattern explicité en A7
- **WebSockets** : voir A0
- **XMPP** : voir A0
- **PubSubHubbub** : voir A0

A2. Webographie

- WebSockets :
 - https://developer.mozilla.org/fr/docs/Web/API/WebSockets_API
 - Dernière consultation le 27/11/2020
 - Documentation complète sur la technologie des WebSockets et présentation d'outils permettant d'implémenter des WebSockets.
- XMPP Overview : <https://xmpp.org/about/technology-overview.html>
- What is PubSubHubbub and how does it work? :
 - <https://www.programmableweb.com/news/what-pubsubhubbub-push-styled-api-and-how-does-it-work/analysis/2017/04/03>
 - Dernière consultation le 27/11/2020
 - Présentation du protocole PubSubHubbub et explication de son fonctionnement.

- Reverse-Ajax :
 - <https://www.ibm.com/developerworks/library/wa-reverseajax1/index.html>
 - Dernière consultation le 27/11/2020
 - Présentation d'Ajax, du Reverse Ajax (Comet) et des WebSockets. Explication des différentes techniques permettant de mettre en place le Reverse Ajax.

A3. Bibliographie

- **Comet and Reverse Ajax**, Phil McCarthy, Dave Crane, 2009, Apress
 - *What are Comet and Reverse Ajax?* (p.1-9) : Présentation de Comet et Reverse Ajax.
 - *Simple Ways to achieve Push* (p.11-31) : Possibilité autres que Comet pour mettre en place Push.
 - *Comet the Easy Way* (p.53-70) : Mettre en place simplement Comet avec des outils tels que Direct Web Remoting.
- **Ajax Patterns and Best Practices**, Christian Gross, 2006, Apress
 - Christian Gross est un ancien consultant pour Microsoft avec une très grande expérience sur les architectures client/serveur.
 - *Persistent Communications Pattern* (p.225-263) : Explication du mécanisme permettant une communication bidirectionnelle.
- **Ajax: The Complete Reference**, Thomas A. Powell, 2008, McGraw-Hill Osborne
 - Thomas A. Powell est impliqué dans la communauté internet depuis 1987. En 1994, il crée son entreprise, spécialisée dans le Web design et Web development.
 - *Networking considerations* (p.231-381) : Présentation de difficultés quant à l'utilisation d'Ajax et solutions possibles.

A4. Acteurs

Google :

Google LLC est une entreprise américaine de services technologiques fondée en 1998 en Californie, par Larry Page et Sergey Brin. Les services qu'elle propose sont nombreux : courrier électronique, moteur de recherche, publicité en ligne, cloud, édition et collaboration (Google Docs, Sheets, Slides).

En 2009, Google a annoncé le lancement d'une fonction de recherche en temps réel. Cette fonction sert à chercher les termes entrés par l'utilisateur sur les pages Twitter, Facebook et MySpace. Les résultats sont classés par pertinence (nombre d'amis s'il s'agit d'un compte Facebook, nombre de retweets pour Twitter).

World Wide Web Consortium :

Le World Wide Web Consortium, W3C, est un organisme de standardisation à but non lucratif, fondé en 1994. Il a pour objectif de promouvoir la compatibilité

des technologies Web (HTML, XML, CSS, SVG...).

Le W3C a contribué au développement en 2011 de WebRTC (Web Real-Time Communication), une interface de programmation JavaScript qui permet la communication en temps réel. Elle permet notamment le partage de fichiers en peer-to-peer.

Internet Engineering Task Force :

L'Internet Engineering Task Force (IETF) est un organisme de normalisation à but non lucratif, fondé en 1986. L'objectif de cette organisation est d'élaborer et de promouvoir des standards internet.

XMPP (Extensible Messaging and Presence Protocol) est un ensemble de protocoles standards de l'IETF, très utilisé pour le web temps réel.

L'IETF a également contribué au développement de WebRTC. Leur livre blanc sur les technologies Web en temps réel s'intitule Web Real-Time Communication Use Cases and Requirements, est a été co-écrit par C. Holmberg, S. Hakansson et G. Eriksson.

A5. Facteurs qualité

- **Efficacité d'exécution** : une application conçue pour fonctionner en temps réel doit bien évidemment avoir une efficacité d'exécution irréprochable. Dans le cas contraire, on perdrait l'aspect temps réel.
- **Indépendance machine** : une technologie web doit être indépendante de l'environnement matériel car elle peut être utilisée dans plein d'environnements différents. L'environnement peut être différent à cause du système d'exploitation (ex : Windows/macOS), du navigateur (ex : Firefox/Opera), de l'appareil (ex : PC/Smartphone) et/ou de l'architecture (ex : x86_64/ARM64).
- **Efficacité de stockage** : une efficacité de stockage dans une application web implique une efficacité dans les ressources envoyées et reçues par les serveurs et donc une économie de bande passante et une meilleure vitesse de chargement de l'application. On retrouve des aspects nécessaires à une application temps réel.

A6. Indicateurs qualité

Efficacité d'exécution :

- Indicateurs de base : fiabilité, facilité d'utilisation
- Indicateur dérivé : efficacité

Indépendance machine :

- Indicateurs de base : portabilité, fiabilité
- Indicateur dérivé : interopérabilité

Efficacité de stockage :

- Indicateurs de base : rendement, fiabilité
- Indicateur dérivé : exactitude

A7. Références théoriques

Ajax-Reverse (IBM) : <https://www.ibm.com/developerworks/library/wa-reverseajax1/index.html>

Ce concept a été introduit par Alex Russel et Joe Walker. Il s'agit d'un mode de fonctionnement où le serveur remonte de l'information au client (et non l'inverse), sans que celui-ci ne l'ait demandé explicitement. Il repose sur le mécanisme du long-polling : La requête HTTP peut rester en attente (d'un événement, d'un timeout), dans quel cas elle se clot.

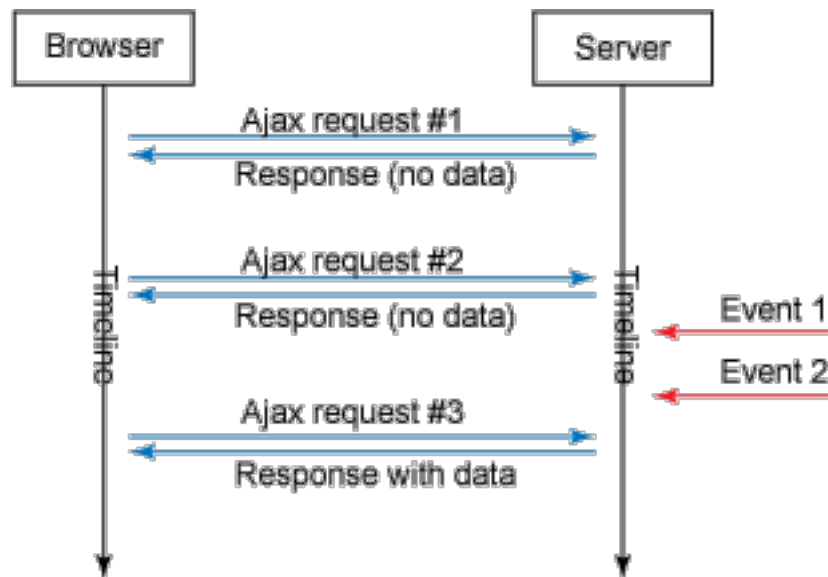


Figure 1: Ajax-Reverse

WebSockets (mozilla)

XMPP (xmpp)

PubSubHubbub (programmableweb)

Partie B

B1. Approche technique

WebSockets

Le WebSocket est une technologie qui permet d'ouvrir un canal de communication bidirectionnelle entre un navigateur (côté client) et un serveur. Avec elle, il est

possible d'envoyer des messages à un serveur et recevoir ses réponses de manière événementielle, sans avoir à consulter le serveur pour obtenir une réponse. Les connexions WebSockets sont similaires aux sockets Unix, mais ce sont bien deux entités distinctes. Il existe des connexions sécurisées **wss://** ou non **ws://**.

Côté client, la technique est composée de 3 éléments :

- Le WebSocket en lui-même, qui est l'interface principale pour se connecter à un serveur WebSocket. Il permet d'envoyer et de recevoir des données sur la connexion ;
- Le CloseEvent qui est l'événement envoyé par l'objet WebSocket lors de la fermeture de la connexion ;
- Le MessageEvent qui est l'événement envoyé par l'objet WebSocket lorsqu'un message en provenance du serveur est reçu.

WebSocket : possède 3 méthodes :

- une méthode permettant de créer la connexion ;
- une méthode permettant d'envoyer des données sur la connexion ;
- une méthode permettant de mettre fin à la connexion.

Généralement, c'est une chaîne de caractères sous le format JSON qui est envoyée, encodée au format UTF-8, mais il est possible de définir un sous-protocole partagé par les deux partis qui définit le format des données envoyées, qu'elles soient des chaînes de caractères ou non.

Les bonnes pratiques veulent que l'utilisation des WebSockets soit faite dans des Web Workers, c'est à dire des threads en tâches de fond, car le thread principal est celui de l'interface utilisateur. Les threads s'échangent des informations grâce à un système de messages.

Un serveur WebSocket doit être écrit dans un langage de programmation capable de gérer les Berkeley sockets.

XMPP

XMPP est un ensemble de technologies pour la messagerie instantanée, entres autres, et de manière générale pour la diffusion de données XML. Les technologies clés sont "Core", "Jingle", "Multi-User Chat", "PubSub" et "BOSH".

- **Core** est, comme son nom l'indique, la technologie au cœur de XMPP, c'est elle qui permet de transmettre des données sous format XML.
- **Jingle** est la technologie permettant de diffuser de la voix, des vidéos, des fichiers et d'autres applications.
- **Multi-User Chat (MUC)** est une extension de Core pour faciliter la communication entre plusieurs clients.
- **PubSub** est une technologie basée sur le design pattern Observer ou Publish-Subscribe, d'où son nom.
- **BOSH** signifie "Bidirectional-streams Over Synchronous HTTP". Elle permet une connexion bidirectionnelle en utilisant le protocole HTTP.

Core définit les méthodes au cœur du protocole. XMPP fonctionne grâce à un réseau distribué de clients et de serveurs communiquant avec des streams XML à travers des connexions TCP. Chaque client possède une adresse de type `identifiant@adresse.serveur`, et chaque serveur possède une adresse de type `adresse.serveur` (ex : `alice@insa-rouen.fr` et `insa-rouen.fr`). Lorsqu'un client A veut envoyer un message à un client B, le client A se connecte au serveur de son domaine et lui transmet le message. Ce serveur va ensuite le faire transiter jusqu'au serveur du domaine du client B qui le stockera. Lorsque le client B se connectera au serveur de son domaine, il recevra le message.

Jingle permet à deux clients XMPP de gérer des sessions pair-à-pair, afin de se transférer des fichiers ou d'établir des flux temps-réel d'audio et/ou de vidéo, pouvant utiliser des canaux TCP ou UDP.

Multi-User Chat (MUC) étend le cœur du protocole pour permettre la création et la modération de salons composés de plusieurs utilisateurs. Ainsi, un client A peut envoyer un message au salon composé des clients A, B et C, et lorsque les clients B et C se connecteront au salon, ils recevront le message.

PubSub étend le cœur du protocole pour permettre aux clients de s'inscrire à des nœuds, et de publier des données sur ces nœuds. Ainsi, les clients A et B peuvent s'inscrire à un nœud, et lorsque le client A publiera un message sur celui-ci, le client B le recevra quand il se connectera au serveur gérant ce nœud.

BOSH est une technologie permettant une communication bidirectionnelle en utilisant le protocole HTTP. Elle utilise beaucoup moins de bande-passante et est beaucoup plus réactive que la plupart des autres protocoles de transport bidirectionnel basé sur le HTTP et que les techniques AJAX. Cet avantage est obtenu en évitant le polling HTTP qui consiste à envoyer des requêtes en permanence au serveur et qui entraîne de nombreuses réponses vides. Au contraire, BOSH simule une connexion TCP de longue durée en utilisant une séquence de requête/réponse HTTP, où la requête est mise en attente par le serveur jusqu'à ce que les données arrivent et soient renvoyées dans la réponse. Une fois la réponse reçue par le client, ce dernier renvoie immédiatement une nouvelle requête afin de continuer la boucle. XMPP utilise en grande majorité cette technologie, mais BOSH peut être utilisé à d'autres fins.

PubSubHubbub

PubSubHubbub est une technologie permettant d'envoyer de l'information au client, sans que celui-ci envoie une requête la demandant. Pour ce faire, le système est composé de 3 éléments : un publicateur, un abonné, et un hub.

- Le **publicateur** est l'entité qui possède l'information, et qui la transmet.
- L'**abonné** est l'entité intéressée par l'information du publicateur.
- Le **hub** est l'entité qui récupère les informations proposées par les publicateurs, et qui se charge de les redistribuer aux abonnés intéressés.

Afin de gérer les différentes informations, PubSubHubbub fonctionne grâce à un

système de topics. Chaque publicateur propose une liste de topics concernant les différentes informations qu'il met à disposition. Chaque client peut demander aux publicateurs les topics qu'ils proposent, puis demander au hub de s'abonner aux topics qui les intéressent.

Un topic est représenté par une URL du hub. Afin de recevoir les informations, les clients doivent disposer d'un serveur accessible par le web et indiquer au hub l'endpoint qui doit recevoir l'information d'un topic.

PubSubHubbub propose également un système de challenge, qui permet d'éviter les abonnements non désirés, en demandant une confirmation de la part du client.

B2. Solutions technologiques concurrentes

WebSockets

- HumbleNet: A cross-platform networking library that works in the browser. It consists of a C wrapper around WebSockets and WebRTC that abstracts away cross-browser differences, facilitating the creation of multi-user networking functionality for games and other apps.
- `uWebSockets`: Highly scalable WebSocket server and client implementation for C++11 and Node.js.
- ClusterWS: Lightweight, fast and powerful framework for building scalable WebSocket applications in Node.js.
- CWS: Fast C++ WebSocket implementation for Node.js (`uWebSockets` v0.14 fork)
- Socket.IO: A long polling/WebSocket based third party transfer protocol for Node.js.
- SocketCluster: A pub/sub WebSocket framework for Node.js with a focus on scalability.
- WebSocket-Node: A WebSocket server API implementation for Node.js.
- Total.js: Web application framework for Node.js (Example: WebSocket chat)
- Faye: A WebSocket (two-ways connections) and EventSource (one-way connections) for Node.js Server and Client.
- SignalR: SignalR will use WebSockets under the covers when it's available, and gracefully fallback to other techniques and technologies when it isn't, while your application code stays the same.
- Caddy: A web server capable of proxying arbitrary commands (stdin/stdout) as a websocket.
- ws: a popular WebSocket client & server library for Node.js.
- jsonrpc-bidirectional: Asynchronous RPC which, on a single connection, may have functions exported on the server and, at the same time, on the client (client may call server, server may also call client).
- cowboy: Cowboy is a small, fast and modern HTTP server for Erlang/OTP with WebSocket support.

XMPP

Core :

- aioxmpp : Python
- Erlang/Elixir XMPP : Elixir / Erlang
- libstrophe : C
- MatriX : .net / C# / Mono
- MatriX vNext : .NET Core / .NET Standard / C#
- Mellium : Go
- python-nbxmpp : Python
- QXmpp : C++
- Slixmpp : Python
- StanzaJS : JavaScript
- Stroke : Java
- Strophe.js : JavaScript
- Swiften: C++
- Tigase JaXMPP : Android / Google Web Toolkit / Java
- Tigase Martin : Swift
- Verse : Lua
- Waher Networking : .NET Core / .NET Standard / C#

Jingle :

- libjingle (C/C++)
- Smack (Java)
- Telepathy Gabble (C)
- yjingle (C++)

PubSub :

- strophe (C or JavaScript)
- XMPP4R (Ruby)

BOSH :

- emite (gwt)
- gloox (C++)
- JSJaC (JavaScript)
- strophe (C or JavaScript)
- Swiften (C++)
- XIFF (Flash)
- XMPP4GWT (gwt)
- xmpp4js (JavaScript)
- XMPP4R (Ruby)

PubSubHubbub

Voir les dépôts de <https://github.com/pubsubhubbub/>

B3. Solutions retenues

Comme nous sommes la seule équipe à travailler sur ce projet, nous avons retenu une seule solution qui est la suivante : utiliser des web sockets avec le **framework Socket.IO** de **Node.js**. Node.js est une plateforme de développement JavaScript. Elle repose sur le langage JavaScript avec des bibliothèques permettant de réaliser des actions comme écrire sur la sortie standard, ouvrir/fermer des connexions réseau ou encore créer un fichier. Un WebSocket est une spécification d'un protocole permettant une communication bidirectionnelle et full duplex sur une seule socket TCP entre un client et un serveur. Le module Socket.IO de Node.js implémente des WebSockets par une approche événementielle (avec les événements suivants : emit, on et broadcast).

B4. Liste de métriques

Les métriques utilisées seront les suivantes :

- le temps moyen entre l'envoi d'une requête et la réception de son accusé de réception. Elle pourra être mesurée de la manière suivante : lorsque le serveur envoie une requête, il lance un timer. Lorsque le client reçoit cette requête, il envoie un accusé de réception. Le serveur arrête le timer à la réception de l'accusé. Cette métrique permettra de mesurer le facteur d'**efficacité d'exécution**.
- le nombre d'environnements différents sur lesquels il sera possible d'utiliser la solution. Pour la mesurer, on pourra essayer notre solution à l'aide des différents appareils que nous avons à notre disposition : smartphone, PC sous Linux, PC sous Windows, différents navigateurs web, Raspberry Pi, etc. Cette métrique permettra de mesurer le facteur d'**indépendance machine**.
- la quantité de données échangées sur une période de temps fixe. Il faudra garder en mémoire la taille de chaque requête envoyée et reçue afin de calculer cette métrique. Cette métrique permettra de mesurer le facteur d'**efficacité de stockage**.

B5. Découpage (WBS) pour réalisation des prototypes

Partie C

C1. Architecture solution W

Comme l'alliance était composée seulement de notre équipe, l'équipe E03, nous travaillerons sur la solution W : le framework Socket.IO de Node.js.

Sources :

- Tutorial Deployment Diagram
- Wikipedia
- Cours de WOS - ASI5

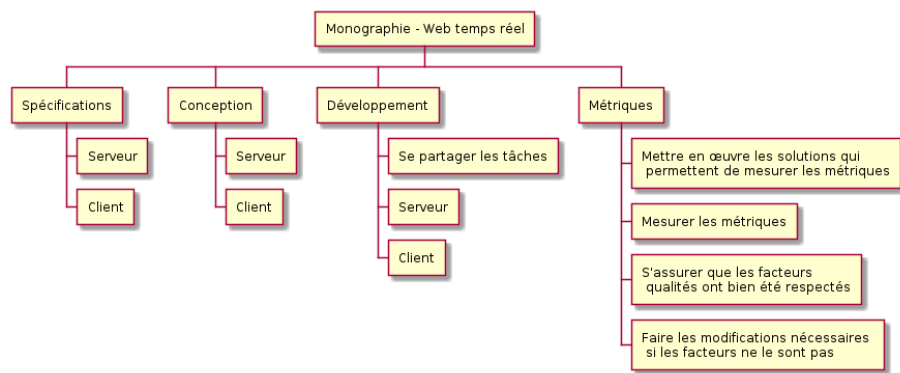


Figure 2: WBS

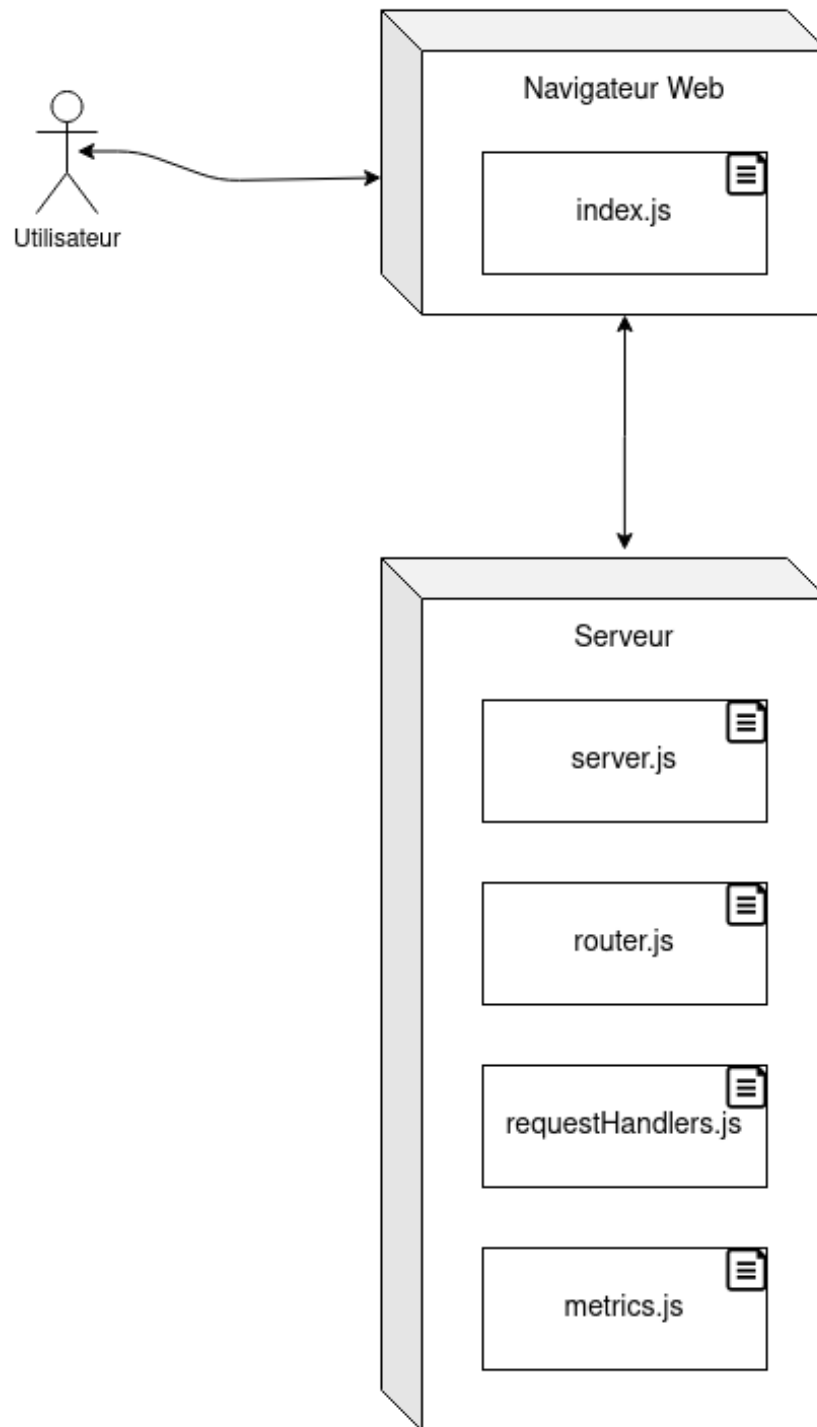


Figure 3: Diagramme de déploiement