
الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب بليلة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master II

Filière Télécommunication

Spécialité Réseaux et télécommunications

La Conception et la réalisation d'une application Web progressive (PWA) de partage de fichiers

Réalisé par : Hiba ZAHRA

encadré par : Dr. N,AMIROUCHE

Année Universitaire 2020-2021

Remerciement :

je voudrais tout d'abord remercier le bon DIEU de m'avoir donné la force et le courage pour mener à bien cet humble travail.

Par la suite, je tiens à exprimer ma reconnaissance à mon encadrante Dr, N.AMIROUCHE d'avoir accepté d'encadrer et d'orienter mon projet.

Enfin, je remercie tous les membres du jury pour l'attention portée à ce travail.

Dédicace

A mes parents, pour leur amour indéfectible.

A mes frères et sœurs, pour leurs précieux conseils, leur soutien permanent et leur encouragement durant mes cinq années d'études universitaires

ملخص :

تطبيق الويب التدريجي (أو PWA) هو تطبيق ويب تم تطويره باستخدام لغات تطوير الويب ويتم تنفيذه من مستعرض ويب. يهدف تشغيله إلى أن يكون قريباً جداً مما يمكن أن يقدمه التطبيق الكلاسيكي (يتم تنزيله من متجر). في هذا العمل ، سنطور تطبيق ويب تقدمياً لمشاركة الملفات باستخدام تقنية WebRTC.

الكلمات المفتاحية :

تطبيق ويب تقدمي ، لغات تطوير الويب ، متصفح ، WebRTC

Résumé :

Une Application Web Progressive (PWA) est une application web développée avec des langages de développement web et exécutée depuis un navigateur web. Son fonctionnement se veut très proche de ce que pourrait proposer une application classique (téléchargée depuis un store). dans ce travail on va développer d'une application Web progressive de partage de fichiers en utilisant la technologie WebRTC .

Les mots clés :

Application Web progressive, langages de développement Web, navigateur, WebRTC

Abstract:

A progressive web application (PWA) is a web application developed using web development languages and executed from a web browser. Its operation is intended to be very close to what the classic application can offer (download from the store). In this work, we will develop a progressive web application for file sharing using WebRTC technology.

Keywords :

Progressive web application, web development languages, browser, WebRTC

Liste des figures

FIGURE 1.1 :L'évolution du web vers la synthèse et l'intelligence artificielle.....	03
FIGURE 1.2 :Etude d'utilisation des systèmes d'exploitation mobiles aux Etats-Unis.....	07
FIGURE 1.3 :Logo de PWA.....	10
FIGURE 2.1 :Fonctionnement du protocole FTP.....	13
FIGURE 2.2 :le client FTP FileZilla.....	14
FIGURE 2.3 :Protocole FTP et SFTP.....	16
FIGURE 2.4 :HTTP et HTTPS sécurité.....	18
FIGURE 2.5 :Echange de communications entre client serveur.....	19
FIGURE 2.6 :Connexion TCP.....	20
FIGURE 2.7 :La persistance.....	21
FIGURE 2.8 :La pipeline.....	22
FIGURE 2.9 :Les certificats de sécurité web.....	24
FIGURE 2.10 :AS2 Messages.....	27
FIGURE 2.11:MDN Async et Sync.....	29
FIGURE 2.12:Protocole du Bank PeSIT.....	30
FIGURE 2.13:Interface de serveur PeSIT.....	32
FIGURE 2.14:Logo de AirDrop.....	32
FIGURE 2.15:L'interface d'AirDrop.....	33
FIGURE 2.16:Logo de Inshare.....	34
FIGURE 2.17:Interface d'InShare.....	34
FIGURE 2.18:Logo de SHATEit.....	35
FIGURE 2.19:Interface de SHAREit.....	35
FIGURE 2.20:Logo Xender.....	36
FIGURE 2.21:Interface de Xender.....	36
FIGURE 2.22:Logo Zapy.....	37
FIGURE 2.23:Interface de Zapy.....	37
FIGURE 3.1 :Logo de WebRTC.....	39
FIGURE 3.2 :Historique des solutions de voix et vidéo sur IP.....	40
FIGURE 3.3 :Communication en temps réel dans le navigateur.....	41
FIGURE 3.4 :Le Triangle WebRTC.....	43
FIGURE 3.5 :Trapèze WebRTC.....	44
FIGURE 3.6 :Serveur STUN.....	46
FIGURE 3.7 :Serveur TURN.....	46
FIGURE 3.8 :Architecture WebRTC.....	47
FIGURE 3.9 :Fichier de l'exemple.....	49
FIGURE 3.10 :la page avant l'envoi.....	53
FIGURE 3.11 :La page après l'envoi.....	54
FIGURE 3.12 :Le Service worker.....	56
FIGURE 3.13 :Logo de HTML5 et CSS3.....	64
FIGURE 3.14 :Logo de Javascript.....	65
FIGURE 3.15 :Logo de Node.js.....	66
FIGURE 3.16 :Logo de Heroku.....	68

Liste des tableaux

TABLEAU 1.1 : Les phases d'évolution du web 1.0 au web 4.0.....	05
TABLEAU 2.1 : Comparaison entre FTP et SFTP.....	17
TABLEAU 3.1 : Les quatre plans de Heroku.....	68-69

Listes des acronymes et abréviations

API	: Application Programming Interface
AS2	: Applicability Statement 2
B2B	: Business to Business
CA	: Certificate Authority
CERN	: Conseil Européen pour la Recherche Nucléaire
CSS	: Cascading Style Sheets
DOM	: Document Object Model
DTLS	: Datagram Transport Layer Security
DV	: Validés par Domaine
EDI	: Echange de Données Informatisé
EV	: Validation Etendue
FTP	: File Transfer Protocol
GSIT	: Groupement d'Intérêt Economique des Systèmes Interbancaires de Téléclearing
HTML	: HyperText Markup Language
HTTP	: Hypertext Transfer Protocol
HTTPS	: Hypertext Transfer Protocol Secure
ICE	: Interactive Connectivity Establishment
IP	: Internet Protocol
JS	: JavaScript
MDN	: Message Disposition Notification
MIC	: Message Integrity Check
NAT	: Network Address Translation
OV	: Validation Organisationnelle
PaaS	: Platform as a Service
PeSIT	: Protocol d'Echanges pour un Systeme Interbancaire de Telecompensation
PDA	: Personal Digital Assistant
PHP	: HypertextPreprocessor
PWA	: Progressive Web Application
S/MIME	: Secure Multipurpose Internet Mail Extensions
SCTP	: Stream Control Transmission Protocol
SFTP	: SSH File Transfer Protocol
SIP	: Session Initiation Protocol
SSL	: Secure Sockets Layer
STUN	: Session Traversal Utilities for NAT
SVG	: Scalable Vector Graphics
TCP	: Transmission Control Protocol
TLS	: Transport Layer Security
TURN	: Traversal Using Relays around NAT
UI	: User Interface
URL	: Uniform Resource Locator
VoIP	: Voice over Internet Protocol
VAN	: Value-Added Network

W3C : World Wide Web Consortium
WebRTC : Web Real-Time Communication
Wi-Fi : Wireless Fidelity
XML : Extensible Markup Language
XUL : XML User Interface Language

Table des matières

Introduction générale:.....	
Chapitre 1 : la révolution de partage de fichiers avec le développement du web	1
1 Introduction	1
2 Le partage de fichier	1
3 Les généralités sur web	1
3.1 La définition du web	1
3.2 La naissance du web	2
3.3 Les étapes de l'évolution du web	2
3.3.1 Web 1.0 début du web traditionnel (ou passif).....	3
3.3.2 Web 2.0 ; web social et participatif	3
3.3.3 Web 3.0 : Web sémantique (smart web)	4
3.3.4 Web 4.0 : Web intelligent.....	4
3.4 Le web mobile	5
3.5 Les applications mobiles.....	6
3.6 Les systèmes d'exploitation mobiles	6
3.7 Les types des applications mobiles	7
3.7.1 Les applications natives.....	7
3.7.2 Application web.....	8
3.7.3 Les applications hybrides	9
4 Définition de PWA	10
5 Conclusion :.....	11
CHAPITRE 2 : Les protocoles et les applications de transfert de fichiers	12
1 Introduction	12
2 Les protocoles de transfert de fichiers	12
2.1 Le protocole FTP	12
2.1.1 Définition et généralités sur FTP	12
2.1.2 Fonctionnement du protocole FTP	12
2.1.3 Les avantages et les inconvénients de FTP	14
2.1.4 Les différents types de FTP	15
2.2 Une comparaison entre FTP et SFTP	16
2.3 HTTP et HTTPS	17
2.3.1 Définition de HTTP et HTTPS.....	17

2.3.2	Client serveur dans le http	19
2.3.3	HTTP : Mécanisme de travail.....	22
2.3.4	Les certificats de sécurité dans le https	23
2.4	Protocole AS2	25
2.4.1	AS2 Définition.....	25
2.4.2	AS2 Messages	26
2.4.3	MDN synchrone et asynchrone	28
2.5	Le Protocole PeSIT	29
2.5.1	PeSIT Définition.....	29
2.5.2	Les Fonctions de service PeSIT	30
3	Les Applications de Transfert de fichiers.....	32
3.1	AirDroid : Description.....	32
3.2	Inshare : Description.....	33
3.3	SHAREit : Description	34
3.4	Xender : Description.....	35
3.5	Zapya : Description.....	36
4	Conclusion.....	37
Chapitre 3 : La Conception du PWA Transfert de fichiers basée Sur le WebRTC.....		38
1	Introduction	38
2	La technologie WebRTC : concept et description.....	38
2.1	WebRTC et Compatibilité sur le navigateur	40
2.2	Architecture WebRTC	42
2.2.1	Le Triangle WebRTC :.....	42
2.2.2	Trapèze WebRTC :.....	43
2.3	La signalisation.....	44
2.3.1	Description	44
2.3.2	Les serveurs ICE (STUN/TURN)	45
2.4	Les API WebRTC	47
2.4.1	MediaStream (alias getUserMedia) :.....	47
2.4.2	RTCPeerConnection (alias PeerConnection) :.....	48
2.4.3	RTCDataChannel (alias DataChannel) :	48
2.5	RTCDataChannel : Description et exemple d'utilisation.....	48
3	PWA : une description technique	54

3.1	Exigences techniques de PWA	54
3.1.1	Le service worker	55
3.1.2	Le fichier manifeste	59
3.1.3	Une connexion sécurisée (HTTPS)	60
3.2	PWA : Outils de programmation	62
3.2.1	HTML : Description	62
3.2.2	CSS : Description	63
3.2.3	Javascript : Description	64
3.2.4	Node.js : Description	65
3.3	PWA : Déploiement.....	66
3.3.1	Les types d'hébergement web	66
3.3.1.1	Les hébergements mutualisés :.....	66
3.3.1.2	Les serveurs dédiés :	67
3.3.2	Heroku : Description et mode d'emploi.....	68
4	Conclusion.....	74
	Conclusion générale	75
	ANNEXE : L'interface de PWA transfert de fichiers.....	76
5	Bibliographie	82

Introduction générale:

L'acte de partage de fichiers n'est certainement pas nouveau. Nous partageons numériquement des fichiers depuis que la disquette a été inventée par Alan Shugart en 1967. Ce qui est nouveau dans le partage de fichiers, c'est la façon dont nous procédons et la vitesse à laquelle les fichiers sont partagés.

De nos jours, Cela peut être fait en quelques clics, et il n'y a certainement pas de limite quant à ce que nous échangeons et partageons avec autrui ou à quelle fréquence. En fait, grâce à la nouvelle technologie, c'est plus facile que jamais.

L'un des biais les plus communs et fiables de transfert de fichier consiste à utiliser un programme de protocole de transfert de fichiers (FTP). Cette méthode transfère des fichiers entre un ordinateur et un serveur sur Internet et est généralement utilisée pour transférer de gros fichiers ou de grandes quantités de données entre plusieurs parties prenantes.

Les utilisateurs de cette technique peuvent accéder aux fichiers partagés à partir d'un site de serveur FTP spécifique. Ces sites offrent soit le partage de fichiers publics, soit la possibilité de visualiser ou de télécharger des fichiers en ayant un mot de passe spécifique. Mais avec l'avènement de la société de l'information et du développement, de nouveaux moyens plus rapides, commodes et faciles ont été mis en place telles que les médias sociaux et les applications numériques abordés dans ce mémoire.

Ce mémoire est composé de trois chapitres :

-Le premier chapitre consiste à donner les étapes de l'évolution du web et les diverses technologies des applications numériques.

-Le deuxième chapitre nous présentons les protocoles employés et leur fonctionnement et les applications mobiles dans le transfert de fichier.

-Le troisième chapitre, c'est la partie la plus cruciale à travers laquelle nous expliquons et décrivons le mode de fonctionnement des applications web progressives (PWA) et Web RTC dans la conception et la réalisation de notre projet.

Pour terminer, la section conclusion synthétise les résultats d'étude et permet de souligner les perspectives d'avenir du projet, suivie par une annexe couvrant le mode d'emploi de notre projet d'étude.

Chapitre 1 : la révolution de partage de fichiers avec le développement du web

1 Introduction

Le web tel que nous le connaissons entre dans sa troisième décennie d'existence. Au cours de cette période, le web a subi de nombreux changements et améliorations surtout le partage de fichiers pour transférer des fichiers volumineux. Mais s'il possède de grands pouvoirs, il a aussi ses limites qui l'empêchent d'offrir une expérience comparable à celle des applications natives. Dans ce chapitre nous verrons les étapes du processus d'innovation numérique de partage de fichiers.

2 Le partage de fichier

Le partage de fichiers impliquait à l'utilisation d'employer des protocoles de transfert de fichiers ou bien la copie de fichiers sur un disque, puis leur envoi par courrier ou leur transmission à un collègue.

Des e-mails peuvent également être envoyés, de nombreuses personnes utilisent encore l'e-mail comme principale option de "partage de fichiers", mais les limites de taille des pièces jointes et les problèmes de sécurité ont découragé cette pratique. Pour remédier à cette situation, un projet pour la refonte et le développement de web a offert un environnement plus concurrentiel et des options presque infinies comme dropox, box, google drive, microdoff et appel ...etc, pour résoudre la question de partage de fichiers

3 Les généralités sur web

3.1 La définition du web

Le projet WWW (world wide web) : « la toile mondiale » en français , connu plus communément le web, est un système hypermédia créé dans le but de distribuer l'information et de faciliter l'accès à n'importe quel ordinateur bureau.[1]

3.2 La naissance du web

C'est une application d'internet, elle a été inventée en 1989 par le britannique Tim Berners-Lee au CERN (centre européen pour la recherche nucléaire) à Genève, vingt ans après l'invention d'internet permettant de relier les ordinateurs formant ainsi une toile entre eux. en se basant sur l'architecture d'internet, offrant aux utilisateurs d'ordinateurs la porte d'accès sur presque n'importe quel sujet. Tim Berners-Lee ne s'est pas arrêté là, il a aussitôt commencé à développer le langage de balisage hypertexte (HTML) ; une technologie qui relie et partage l'information en créant ainsi un réseau de nœuds via lequel l'utilisateur peut surfer sur le web. Il a également écrit des protocoles de communication pour former l'épine dorsale de son nouveau système d'information en particulier le protocole de communication TCP/IP pour la transmission de l'information sur le web. Tim Berners-Lee crée le premier navigateur et éditeur web, appelé WorldWideWeb et le premier serveur HTTP appelé CERN httpd, légué au domaine public en 1992. Avec l'obtention de l'accord du CERN en 1993, le web a connu une forte utilisation avec un programme qui donne accès au monde hypertexte que l'on appelle un navigateur web Mosaic, développé par Eric Bina et Marc Andreessen de l'équipe NCSA (National Center for Supercomputing Applications) dans Illinois; une interface graphique conviviale fournit une interface utilisateur unique à de grandes classes d'informations (image au textes). Puis, ils ont quitté la NCSA pour fonder Netscape communications corporation en 1994 créeront Netscape navigator pour les connexions bas débit, aujourd'hui connu sous le nom du Mozilla Firefox [2]

3.3 Les étapes de l'évolution du web

Le web tel que l'on connaît aujourd'hui et qui investit notre vie quotidienne depuis plus de 30 ans, a déclenché une véritable révolution numérique. Cette révolution est passée par différentes phases.

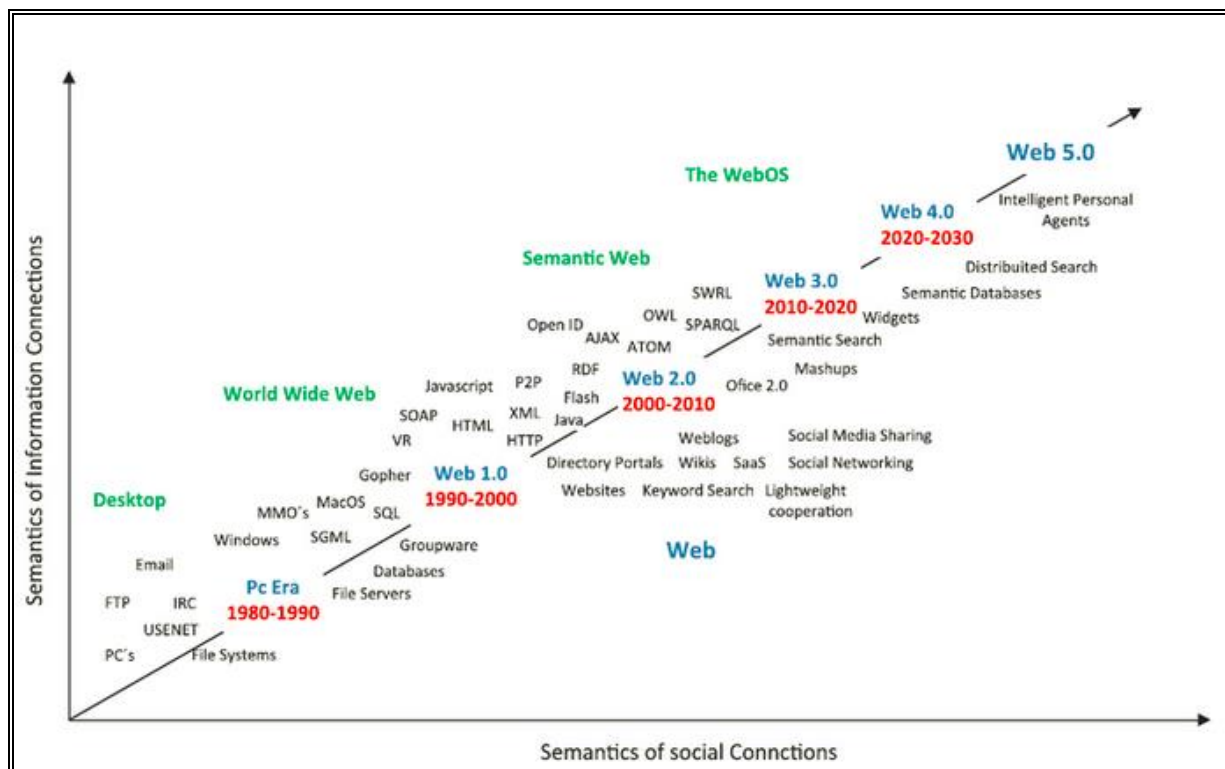


Figure 1-1 L'évolution du web vers la synthèse et l'intelligence artificielle [3]

3.3.1 Web 1.0 début du web traditionnel (ou passif)

La première implémentation du web représente le web 1.0, des années 90, qui, selon Berners-Lee, pourrait être considéré comme le "web en lecture seule". En d'autres termes, le web qui au début nous a permis de rechercher des informations et de les lire, avait un fonctionnement linéaire, très peu d'interaction avec l'utilisateur ou de contribution au contenu. Cependant, c'est exactement ce que la plupart des propriétaires de sites Web voulaient : leur objectif pour un site Web était d'établir une présence en ligne et de rendre leurs informations accessibles, au format statique et sans aucune forme d'interactivité avec les lecteurs. Des technologies telles que les applications commerciales, l'émergence de l'e-commerce et du panier shopping, catalogues produits en ligne et les encyclopédies en ligne. [4]

3.3.2 Web 2.0 ; web social et participatif

Début des années 2000 apparaît le Web 2.0, ou du Web "lecture-écriture" si l'on s'en tient à la méthode de Berners-Lee pour le décrire. La capacité nouvellement introduite de contribuer au contenu et d'interagir avec d'autres utilisateurs du Web a radicalement changé le paysage du Web en peu de temps. En faisant allusion aux numéros de version qui désignent généralement

les mises à niveau logicielles, l'expression « Web 2.0 » fait allusion à une forme améliorée du WWW.

Des technologies telles que les blogs (blogs), les wikis, les podcasts, (et d'autres formes de publication plusieurs-à-plusieurs), les API Web et les services Web en ligne tels que eBay et Gmail offrent des améliorations par rapport à la lecture. Uniquement des sites Web. Stephen Fry (acteur, auteur et diffuseur) décrit le Web 2.0 comme « une idée dans la tête des gens plutôt qu'une réalité. C'est en fait une idée que la réciprocité entre l'utilisateur et le fournisseur est ce qui est mis en avant. En d'autres termes, une véritable interactivité, si vous aimez, simplement parce que les gens peuvent aussi bien télécharger que télécharger »

Selon certaines sources, le terme Web 2.0 existe depuis environ octobre 2004. De Wikipedia, l'encyclopédie Web gratuite, il est défini comme Web 2.0 est un terme souvent appliqué à une transition en cours perçue du WWW d'une collection de sites Web à une plate-forme informatique à part entière servant des applications Web aux utilisateurs finaux. À terme, les services Web 2.0 devraient remplacer les applications informatiques de bureau à de nombreuses fins. [5]

3.3.3 Web 3.0 : Web sémantique (smart web)

Le Web 3.0 est un terme utilisé pour décrire diverses évolutions de l'utilisation et de l'interaction du Web selon plusieurs chemins. Cette version a vu le jour en 2010. En prolongeant les explications de Tim Berners-Lee, le Web 3.0 s'apparenterait à un Web "lecture-écriture-exécution". Le Web 3.0 est défini comme la création de contenu et de services de haute qualité produits par des individus doués en utilisant les technologies Web 2.0 comme plate-forme habilitante. Il s'agit notamment de la transformation du Web en base de données, d'une évolution vers la mise à disposition du contenu par de multiples applications sans navigateur, l'exploitation des technologies d'intelligence artificielle, le Web sémantique, le Web géospatial ou le Web 3D. [4]

3.3.4 Web 4.0 : Web intelligent

Le Web 4.0 émerge comme une nouvelle génération Web. Il rassemblera tous les aspects du Web 2.0 et du Web 3.0 pour devenir vraiment omniprésent, s'apparenterait à un web "lecture-écriture-exécution-concurrence", couvrant un ensemble de dimensions multiples. Chacune de ces dimensions offre une vue distincte, mais simultanément complète, du paradigme Web 4.0. Il ressort que les applications telles que les réseaux sociaux et les technologies telles que

l'Internet des objets, le Big Data, l'intelligence artificielle et l'apprentissage automatique jouent un rôle clé dans l'adoption et la mise en œuvre du Web 4.0 .

Les principales différences entre ces formats WEB résident dans la dynamique et l'interactivité, le tableau ci-dessous résume les phases du web.

WEB 1.0	Période	1991-1999
	Objectif	Donner accès au contenu en ligne – présenter des produits aux consommateurs.
	Langage	Protocoles HTTP, HTML (langage avec liens hypertextes), puis XML Java & JavaScript.
	Outils de communication	Email – Forums
	Utilisation	Consommateur passif – "read only" – "view and link" – le web est un énorme magazine en ligne que l'utilisateur consulte.
WEB 2.0	Période	2000-2009
	Objectif	Partager du contenu.
	Langage	XML (HTML structuré) qui consiste à décrire les choses avec des mots + RSS (flux de contenus).
	Outils de communication	Réseaux sociaux – plateformes collaboratives SMS – MMS – video-streaming.
	Utilisation	Consommateur et acteur. Néanmoins, seule une minorité devient auteur "read & write", la majorité se contente d'un engagement restreint "read & share".
WEB 3.0	Période	2010-????
	Objectif	Connecte les personnes, orienté communautés et "tribu".
	Langage	Outre le XML, de nouveaux langages émergent : RDF + OWL + SWRL pour une meilleure compréhension de ce que les humains cherchent.
	Outils de communication	Tous les outils adaptés à l'internet mobile (tablette, smartphones) + des outils cross media tels que QR codes, RFID (radio frequency identification).
	Utilisation	Consommateur et acteur. Néanmoins, seule une minorité devient auteur "read & write", la majorité se contente d'un engagement restreint "read & share".
WEB 4.0	Période	2020 ? - ????
	Objectif	Connecte l'intelligence, orienté interaction individus/objets.
	Langage	Langage unique Serveurs / Mobiles / Objets connectés.
	Outils de communication	Les outils de collecte d'infos et d'affichage d'information sont multiples et font partie de notre quotidien et de notre environnement, de notre environnement.
	Utilisation	L'humain devient consomm-acteur et cré-acteur malgré lui. La technologie nous facilite la vie sans plus avoir à se poser la question : Est-ce que c'est bien ou mal... C'est tout simplement obligatoire.

Tableau 1-1 Les phases d'évolution du web 1.0 au web 4.0

3.4 Le web mobile

le marché de l'internet mobile est en plein essor surtout avec l'avènement des smartphones et ses fonctionnalités emblématique. Un smartphone se caractérise par un navigateur web et des applications web. En 2011 près d'un tiers est vendu dans le monde, destiné au grand public, devenu ainsi indispensable : discussion instantanée, navigation sur internet ...etc.

3.5 Les applications mobiles

Une application mobile, aussi appelée application web, également application en ligne ou tout simplement application, est un type de logiciel applicatif développé pour s'exécuter sur un appareil mobile (smartphone ou une tablette).

Ses applications ne pourraient être exécutées qu'à partir d'un système d'exploitation mobile, le plus utilisé est Android.

3.6 Les systèmes d'exploitation mobiles

- 1- **Android (open source mobile)** : racheté par Google en 2007, développé avec le langage java, adapté pour les Smartphones, tablettes et PDA (Personal Digital Assistant). Renommé sous le nom de « Google Play » en 2012, permet aux développeurs de distribuer leurs applications en quelques heures et compatible au-delà des appareils mobiles. [6]

- 2- **IOS(iPhone Operating System)** : conçu par Apple, uniquement pour ses appareils et périphériques compatibles (Mac, Appel TV, Appel Watch). Développé par le langage Objective-C et Swift. Swift a été conçu pour être plus rapide que Objective-C en 2014. Il a également été conçu pour améliorer la sécurité des produits iOS. [6]

- 3- **Windows phone** : conçu par microsoft, restreint uniquement aux professionnels, fut élargie par la suite aux particuliers. Développé par C#, similaire à java en terme de syntaxe, permet la création d'applications informatiques de toutes sortes.

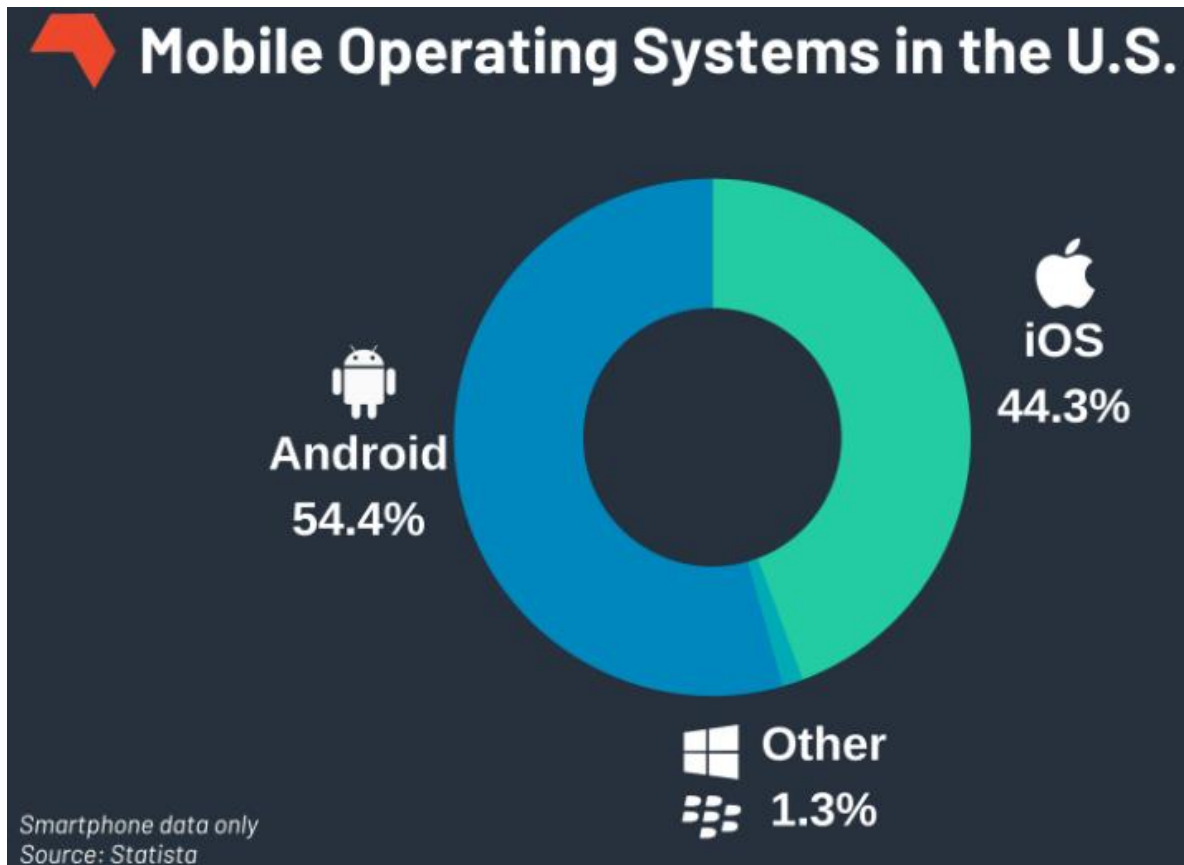


Figure 1-2 Etude d'utilisation des systèmes d'exploitation mobiles aux Etats-Unis (mai 2021)

3.7 Les types des applications mobiles

Il existe trois types d'applications mobiles ; natives, web et hybrides. [7]

3.7.1 Les applications natives

Sont conçues pour une plate-forme et un système d'exploitation uniques. Les utilisateurs peuvent utiliser des applications mobiles Android natives ou des applications iOS natives, qui ne sont pas mentionnées dans la plupart des autres plates-formes et appareils mobiles.

Le développement de ce type d'application requiert le recours à la mémoire du Smartphone. Ils peuvent tirer pleinement parti de toutes les capacités d'un appareil ; le microphone, l'appareil photo, ainsi que le GPS, le système de navigation, la boussole, la liste de contacts et bien d'autres. Les applications natives sont écrites dans différents langages de programmation. Certains des langages de programmation sont Java, React Native, Perl, Python, Swift, JavaScript, Kotlin, Android, Ruby, C ou C++... etc.

Les Avantages et les inconvénients des applications natives

Avantages :

- Elles sont plus rapides et plus précis en termes de qualité.
- Offrir aux clients une application mobile plus personnalisée, plus performante et de profiter d'une meilleure expérience utilisateur.
- Utiliser et intégrer à un large éventail les fonctionnalités d'appareils mobiles telles que Bluetooth, annuaire téléphonique, contacts, GPS...etc., Par exemple, une application avec un accès au GPS permettra de vous géo-localiser et de calculer un itinéraire vers un point précis. [7]

Inconvénients :

- La création d'une application native avec une plate-forme ne peut pas être répétée sur une autre plate-forme.
- Pour une mise à jour, l'utilisateur doit télécharger son dernier fichier, puis réinstaller le précédent et prendre une grande partie de stockage.
- Coût lié au développement d'une application native est très élevé.

3.7.2 Application web

Sont des versions réactives des sites Web, l'utilisateur peut accéder à l'application web par le biais de son navigateur depuis son mobile et donc vous n'avez pas besoin de la télécharger, car elles sont distribuées pour utiliser un navigateur Web mobile et elles ne se servent pas de la mémoire et les fonctionnalités du Smartphone.

Les applications web sont conçues à partir de divers langages de programmation : HTML, CSS, React, PHP, JavaScript, Native JS, Python, SQL, NET, Ruby, Angular Java...etc. Pour le travail sur le site Web.

Les avantages et les inconvénients des Applications Web

Avantages :

- Les utilisateurs n'auront pas besoin de télécharger ni d'installer ce type d'application car ils accèdent via un simple URL.
- Applications très réactifs qui personnalisent la conception de leur interface en fonction du système sur lequel l'utilisateur travaille.
- Elles ne consomment pas du stockage de mémoire dynamique sur votre ordinateur comme une application native.

Inconvénients :

- Accès restreint à certaines fonctions de votre appareil.
- Vitesse légèrement réduite à celle hébergée sur un serveur local
- Elles nécessitent une connexion Internet active pour s'exécuter, si vous perdez la connexion, vous ne pouvez pas accéder à votre application web.

3.7.3 Les applications hybrides

sont le mélange de deux applications, à savoir les applications natives (développées pour une plateforme en particulier, iOS ou Android par exemple) et les applications Web (disponibles sur internet via un navigateur et non installées localement).elle utilise également un mélange des technologies natives et Web : Java, Perl, Python, Swift, Kotlin, Android, Ruby, C ou C++, HTML, CSS, React, PHP, JavaScript, Native JS, Python, SQL, .NET, Ruby, Angular...etc.

Les avantages et les inconvénients d'application hybride [7]

Avantages :

- Rapidité de son développement donc coût réduit.
- Les mises à jour sont plus simples et rapides.

-Evolution d'application avec des fonctionnalités dans les deux plateformes.

Inconvénients :

-La difficulté ou l'impossibilité d'exploiter toutes les capacités des plateformes.

-L'interface utilisateur limitée donc pas fluide et lente.

-Profil de développeur compliqué à trouver.

4 Définition de PWA

Les applications Web progressives (PWA) sont un type d'applications Web fournies via le Web, qui combinent les fonctionnalités avancées offertes par les navigateurs avec les avantages d'une expérience mobile. Les applications Web progressives promettent une expérience de navigation puissante, semblable à celle d'une application. Ils sont utiles aux utilisateurs dès la première visite de votre site Web dans un onglet de navigateur car ils n'ont pas besoin d'être installés comme des applications natives. [8]

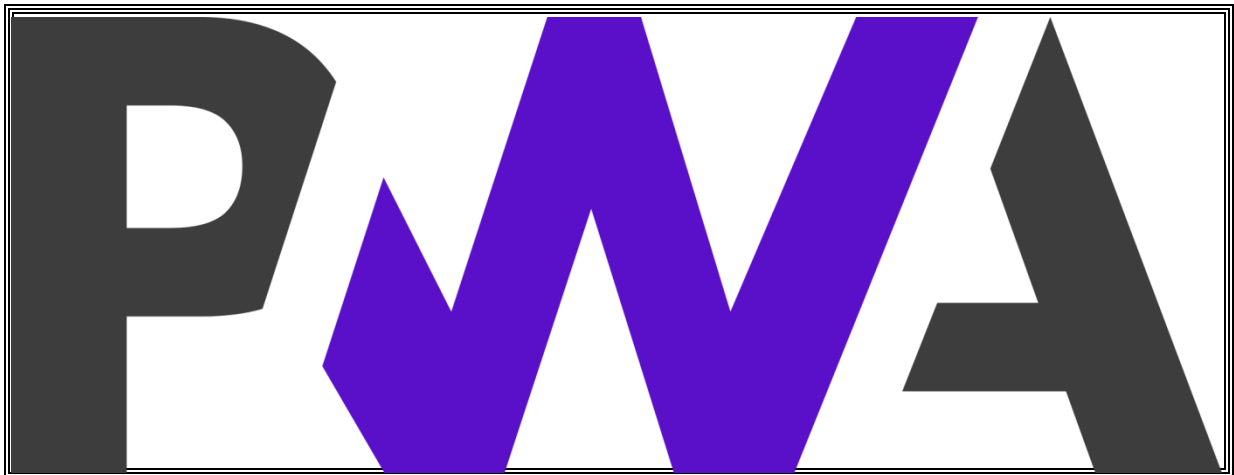


Figure 1-3 Logo de PWA

Une application native, comme celles que vous téléchargez sur l'App Store d'Apple ou le Play Store de Google, est souvent construite dans un langage de programmation spécifique à cette plate-forme. Donc, pour les applications iOS, ce serait Swift et pour les applications Android, Java. Si vous souhaitez créer une application pour ces plates-formes, vous devez connaître la technologie. Certes, il existe des raccourcis, mais ceux-ci ont leurs propres limites. [8]

Si vous souhaitez avoir une application sur toutes les plateformes mobiles, vous devez connaître toutes les différentes technologies. Il n'y a pas de moyen facile d'en créer un et de le publier dans tous les magasins. Bien sûr, il existe des moyens de tirer le meilleur parti des deux mondes. Une application Web progressive, par exemple. Cela s'exécute dans le navigateur et, une fois enregistré sur l'écran d'accueil, fonctionne comme une application native. Il a même accès au matériel et aux logiciels sous-jacents auxquels le navigateur ne peut pas accéder pour des raisons de sécurité. Si le PWA fonctionne très bien, les utilisateurs ne sauront jamais qu'ils utilisent une application Web au lieu d'une application native.[8]

-Le PWA a une base de code unique pour toutes les plateformes.

-Ils ont accès à la plupart des capacités matérielles mobiles natives.

-Ils envoient des notifications push.

-Le PWA fonctionne comme un site Web réactif ou une application mobile en fonction de l'appareil de l'utilisateur.

-Les PWA n'ont pas besoin d'être téléchargés ou installés, il suffit de visiter le site Web

5 Conclusion :

Dans ce chapitre, on a étudié les avancées du web dans l'amélioration de la problématique de partage de fichier.

CHAPITRE 2 : Les protocoles et les applications de transfert de fichiers

1 Introduction

Le transfert de fichiers est l'un des piliers de la télécommunication aujourd'hui, car nos transactions quotidiennes ne sont pas dépourvues de transfert d'image, de son ou de tout type de fichier. Pour qu'un processus de transfert de fichiers ait lieu, nous devons utiliser un ensemble de protocoles de transfert de fichiers. Bien sûr, nous ne traiterons pas directement ces protocoles, mais nous utiliserons un ensemble d'applications mobile pour le transfert de fichiers. Dans ce chapitre, nous essaierons de passer en revue les protocoles les plus importants pour le transfert de fichiers ainsi que ces applications mobile.

2 Les protocoles de transfert de fichiers

2.1 Le protocole FTP

2.1.1 Définition et généralités sur FTP

Le protocole de transfert de fichiers (FTP) est un protocole de communication utilisé pour transférer ou échanger des fichiers entre deux ordinateurs. Ces transferts de fichiers sont généralement authentifiés par des identifiants de nom d'utilisateur et de mot de passe. Le FTP anonyme permet aux utilisateurs d'accéder à des fichiers, programmes et autres données à partir d'Internet sans avoir besoin d'un identifiant ou d'un mot de passe. Les sites Web sont parfois conçus pour permettre aux utilisateurs d'utiliser « anonyme » ou « invité » comme identifiant d'utilisateur et une adresse e-mail pour un mot de passe. File Transfer Protocol est la norme Internet pour le transfert ou l'échange de fichiers d'un ordinateur à un autre à l'aide de réseaux TCP ou IP.[9]

2.1.2 Fonctionnement du protocole FTP

FTP est un protocole client-serveur qui repose sur deux canaux de communication entre le client et le serveur : **un canal de commande** pour contrôler la conversation et **un canal de données** pour transmettre le contenu du fichier.

Voici comment fonctionne un transfert FTP typique :

- 1) Un utilisateur doit généralement se connecter au serveur FTP, bien que certains serveurs rendent tout ou partie de leur contenu disponible sans connexion, un modèle connu sous le nom de FTP anonyme.

- 2) Le client initie une conversation avec le serveur lorsque l'utilisateur demande à télécharger un fichier.
- 3) En utilisant FTP, un client peut charger, télécharger, supprimer, renommer, déplacer et copier des fichiers sur un serveur.

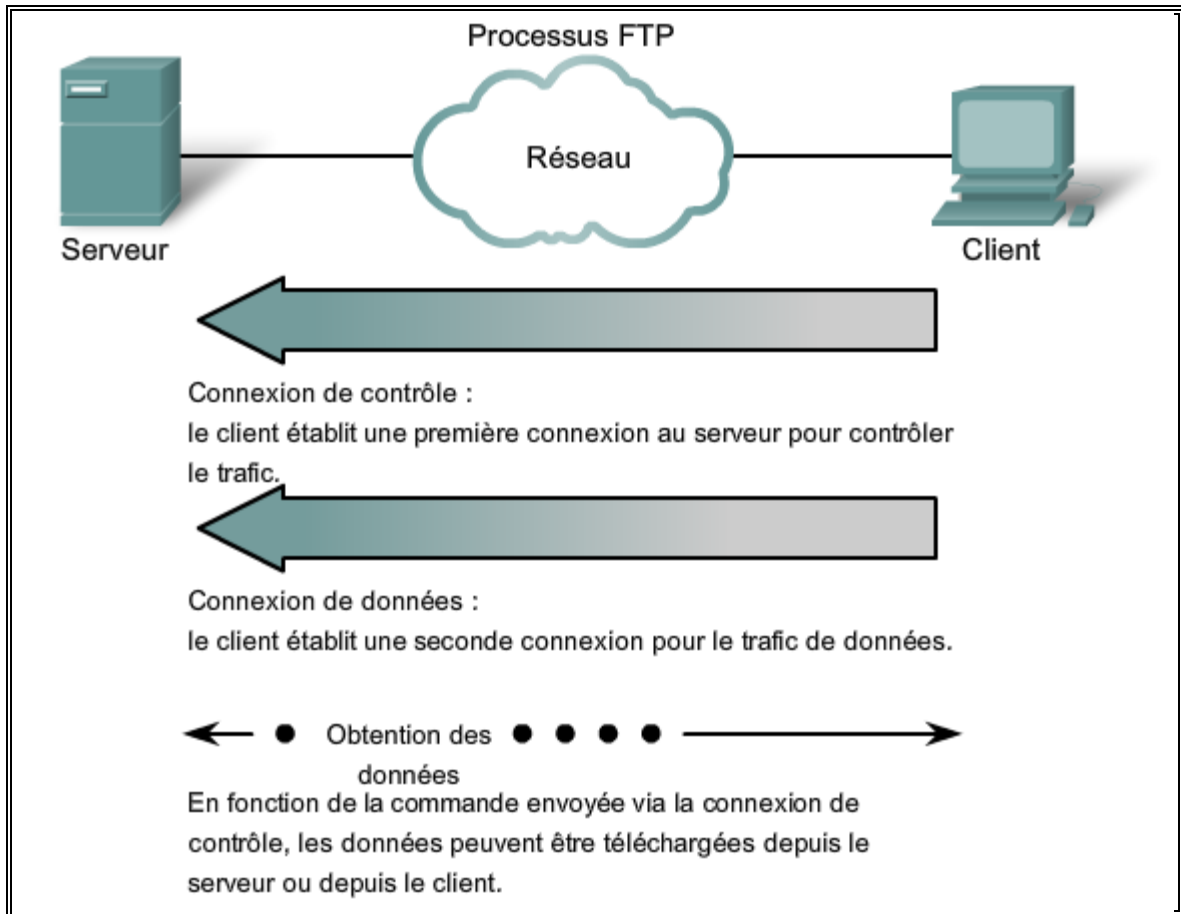


Figure 2-1 Fonctionnement du protocole FTP [10]

Lorsque vous allez utiliser le protocole FTP, il faudra un serveur FTP. Celui-ci va permettre de stocker les différents fichiers que vous souhaitez partager. Dans le cadre d'un site internet, il va héberger les fichiers qui permettront l'affichage de ce dernier.[10]

Mais pour se connecter sur le serveur FTP, il vous faudra un client FTP. Il s'agit d'un logiciel qui vous permettra d'établir la connexion entre vous et le serveur FTP afin d'y échanger des fichiers. Lorsque vous serez connecté, il sera dès lors possible de transférer, supprimer, modifier... vos fichiers afin qu'il soit accessible (ou non) depuis votre site Internet.

La plupart des navigateurs Web sont livrés avec des clients FTP qui permettent aux utilisateurs de transférer des fichiers de leur ordinateur vers un serveur et vice versa. Certains

utilisateurs peuvent souhaiter utiliser un client FTP tiers, car nombre d'entre eux offrent des fonctionnalités supplémentaires. Des exemples de clients FTP téléchargeables gratuitement incluent **FileZilla Client**, **FTP Voyager**, **WinSCP**, **CoffeeCup Free FTP** et **Core FTP**.



Figure 2-2 le client FTP FileZilla

2.1.3 Les avantages et les inconvénients de FTP

A- Les avantages de FTP

- 1) **Vitesse** : L'un des plus grands avantages du FTP est la vitesse. Le FTP est l'un des moyens les plus rapides de transférer les fichiers d'un ordinateur à un autre.
- 2) **Efficace** : C'est plus efficace car nous n'avons pas besoin de terminer toutes les opérations pour obtenir l'intégralité du fichier.
- 3) **Sécurité** : Pour accéder au serveur FTP, nous devons nous connecter avec le nom d'utilisateur et le mot de passe. Par conséquent, nous pouvons dire que FTP est plus sécurisé.
- 4) **Mouvement de va-et-vient** : FTP nous permet de transférer les fichiers dans les deux sens. Supposons que vous soyez un responsable de l'entreprise, que vous envoyez des informations à tous les employés, et qu'ils renvoient tous des informations sur le même serveur.

B- Les inconvénients de FTP

- 1) L'exigence standard de l'industrie est que toutes les transmissions FTP doivent être cryptées. Cependant, tous les fournisseurs FTP ne sont pas égaux et tous les

fournisseurs ne proposent pas de cryptage. Nous devons donc rechercher les fournisseurs FTP qui fournissent le cryptage.

- 2) FTP sert à deux opérations, c'est-à-dire envoyer et recevoir des fichiers volumineux sur un réseau. Cependant, la taille limite du fichier est de 2 Go pouvant être envoyé. Il ne vous permet pas non plus d'effectuer des transferts simultanés vers plusieurs récepteurs.
- 3) Les mots de passe et le contenu des fichiers sont envoyés en texte clair qui permet les écoutes indésirables. Ainsi, il est tout à fait possible que des attaquants puissent effectuer l'attaque par force brute en essayant de deviner le mot de passe FTP.
- 4) Il n'est pas compatible avec tous les systèmes.

2.1.4 Les différents types de FTP

Il existe plusieurs manières différentes pour un serveur FTP et un logiciel client d'effectuer un transfert de fichiers via FTP :[11]

- 1) **FTP anonyme** : C'est la forme la plus basique de FTP. Il prend en charge les transferts de données sans crypter les données ni utiliser de nom d'utilisateur et de mot de passe. Il est le plus souvent utilisé pour le téléchargement de matériel autorisé à une distribution illimitée.
- 2) **FTP protégé par mot de passe** : il s'agit également d'un service FTP de base, mais il nécessite l'utilisation d'un nom d'utilisateur et d'un mot de passe, bien que le service puisse ne pas être crypté ou sécurisé. Il fonctionne également sur le port 21.
- 3) **FTP sécurisé (FTPS)** : Parfois appelé FTP sécurisé Sockets Layer (FTP-SSL), cette approche active la TLS (Transport Layer Security) implicite dès qu'une connexion FTP est établie. FTPS a été initialement utilisé pour permettre une forme plus sécurisée de transfert de données FTP. Il utilise généralement par défaut le port 990.
- 4) **FTP sur SSL/TLS explicite (FTPES)** : cette approche permet la prise en charge explicite de TLS en mettant à niveau une connexion FTP sur le port 21 vers une connexion cryptée. Il s'agit d'une approche couramment utilisée par les services Web et de partage de fichiers pour permettre des transferts de fichiers sécurisés.

5) **Secure FTP (SFTP)** : Ce n'est techniquement pas un protocole FTP, mais il fonctionne de manière similaire. Au contraire, SFTP est un sous-ensemble du protocole Secure Shell (SSH) qui s'exécute sur le port 22. SSH est couramment utilisé par les administrateurs système pour accéder à distance et en toute sécurité aux systèmes et aux applications, et SFTP fournit un mécanisme au sein de SSH pour le transfert de fichiers sécurisé.[11]

2.2 Une comparaison entre FTP et SFTP

Secure File Transfer Protocol ou SSH File Transfer Protocol (SFTP) et File Transfer Protocol (FTP) ne sont PAS la même chose. SFTP, à ne pas confondre avec FTP Secure (FTPS), est un réseau qui permet l'accès, le transfert et la gestion de fichiers sur un flux de données sécurisé. Il s'agit d'une extension de la version 2.0 du protocole Secure Shell (SSH), dont le seul objectif est de fournir des capacités de transfert sécurisées et de fonctionner fonctionnellement avec d'autres protocoles. FTP est un protocole réseau mis en œuvre pour échanger des fichiers sur un réseau TCP (Transmission Control Protocol) et IP (Internet Protocol).

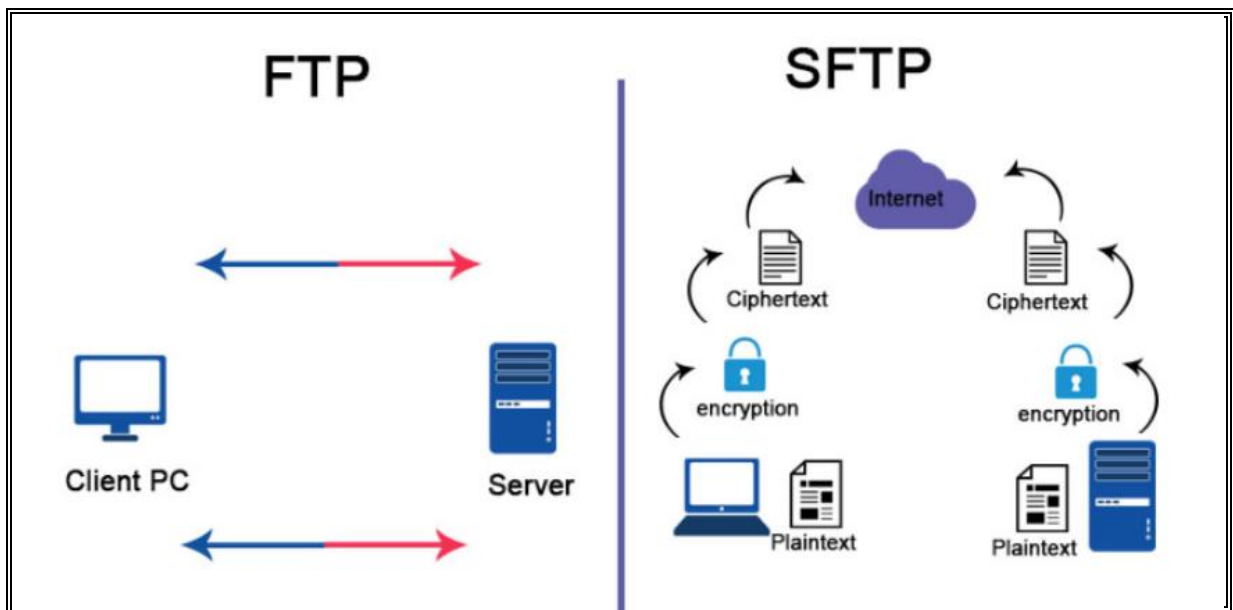


Figure 2-3 Protocole FTP et SFTP

FTP	SFTP
FTP signifie File Transfer Protocol.	SFTP signifie Secure File Transfer Protocol.
FTP télécharge ses données sans aucune sécurité.	SFTP fournit une sécurité totale aux données à l'aide de l'authentification du protocole SSH. Il utilise également des clés SSH pour fournir l'authentification.

Tout le monde peut accéder au FTP.	SFTP n'est accessible que par le propriétaire du serveur car le port 22 n'est pas ouvert en cas d'hébergement partagé.
FTP n'offre pas de canal de transmission sécurisé pour les documents entre les hôtes.	Pour transférer des fichiers entre l'hôte, SFTP fournit un canal sécurisé.
DANS la plupart des cas, FTP n'est pas crypté car FTP est accessible de manière anonyme.	Avant de l'envoyer à un autre hôte, SFTP crypte les informations.
Dans le FTP, l'architecture client-serveur est utilisée.	Dans le protocole de transfert de fichiers sécurisé, l'architecture SSH est utilisée.
Dans le protocole de transfert de fichiers, il y a le protocole TCP/IP.	SFTP est un composant du programme d'application de connexion SSH qui est un protocole de connexion à distance.
Dans le protocole de transfert de fichiers, 2 canaux sont utilisés.	Dans le protocole de transfert de fichiers sécurisé, un seul canal est utilisé.
Le protocole de transfert de fichiers est couramment utilisé.	Mais le protocole de transfert de fichiers sécurisé n'est pas utilisé couramment.
Le protocole de transfert de fichiers utilise une méthode directe pour transférer des fichiers.	Le protocole de transfert de fichiers sécurisé utilise la méthode de tunneling pour le transfert de fichiers.
File Transfer Protocol Autoriser le lien entrant du port 21.	Protocole de transfert de fichiers sécurisé Autoriser les liens entrants sur le port 22
Filezilla est la meilleure solution pour utiliser FTP.	WinSCP est un excellent service pour utiliser SFTP.

Tableau 2-1 Comparaison entre FTP et SFTP

2.3 HTTP et HTTPS

2.3.1 Définition de HTTP et HTTPS

HTTP est la façon dont les navigateurs Web demandent des pages Web. C'était l'une des trois principales technologies définies par Tim Berners-Lee lorsqu'il a inventé le Web, ainsi que des identifiants uniques pour les ressources (d'où proviennent les Uniform Resource Locators, ou URL). et le langage de balisage hypertexte (HTML).[4]

HTTP est l'abréviation de protocole de transfert hypertexte. Il s'agit de la principale méthode par laquelle les données des pages Web sont transférées sur un réseau. Les pages Web sont stockées sur des serveurs, qui sont ensuite servis à l'ordinateur client lorsque l'utilisateur y accède. Les messages HTTP sont envoyés sur Internet non cryptés et sont donc lisibles par

toute partie qui voit le message lorsqu'il est acheminé vers sa destination. Internet, comme son nom l'indique, est un réseau d'ordinateurs, pas un système point à point. Internet n'offre aucun contrôle sur la façon dont les messages sont acheminés, et vous, en tant qu'internaute, n'avez aucune idée du nombre d'autres parties qui verront vos messages lorsqu'ils sont envoyés sur Internet depuis votre fournisseur d'accès Internet (FAI) aux entreprises de télécommunications et autres parties. Parce que HTTP est en texte brut, les messages peuvent être interceptés, lus et même modifiés en cours de route. [12]

HTTPS est la version sécurisée de HTTP qui crypte les messages en transit à l'aide du protocole Transport Layer Security (TLS), bien qu'il soit souvent connu sous son incarnation précédente sous le nom de Secure Sockets Layer (SSL), comme indiqué dans Fig 2-4 ci-dessous.

HTTPS ajoute trois concepts importants aux messages http

- 1) Cryptage : les messages ne peuvent pas être lus par des tiers pendant leur transit.
- 2) Intégrité : le message n'a pas été altéré pendant le transit, car l'intégralité du message chiffré est signée numériquement et cette signature est vérifiée cryptographiquement avant le déchiffrement.
- 3) Authentification : Le serveur est celui avec qui vous aviez l'intention de parler



Figure 2-4 HTTP et HTTPS sécurité

2.3.2 Client serveur dans le http

La différence la plus évidente entre les clients http et les serveurs est la responsabilité d'initier la communication. Seul un client peut le faire. Un serveur peut avoir beaucoup d'informations qu'il peut fournir et de nombreuses fonctions qu'il peut exécuter, mais il ne fait quelque chose que lorsqu'un client le lui demande. Un client http agit et un serveur http réagit.[5]

Comme de nombreux protocoles de communication, http fait une distinction clé entre les deux parties communicantes. Dans tout échange http, un système assume le rôle de client tandis que l'autre est un serveur. Cette différence est très importante, car http oblige les clients et les serveurs à suivre des règles et des procédures très différentes. Dans une simple session Web, le pc naviguant sur le Web est un client http, tandis que le système hébergeant le site Web agit comme un serveur http. Même si ces deux systèmes communiquent tous les deux en utilisant http, ils ont évidemment des responsabilités très différentes dans cette communication. le client, qui initie toujours les communications http, contrôle plusieurs caractéristiques importantes de la session, y compris :

- a) la connexion TCP
- b) la persistance
- c) le pipeline

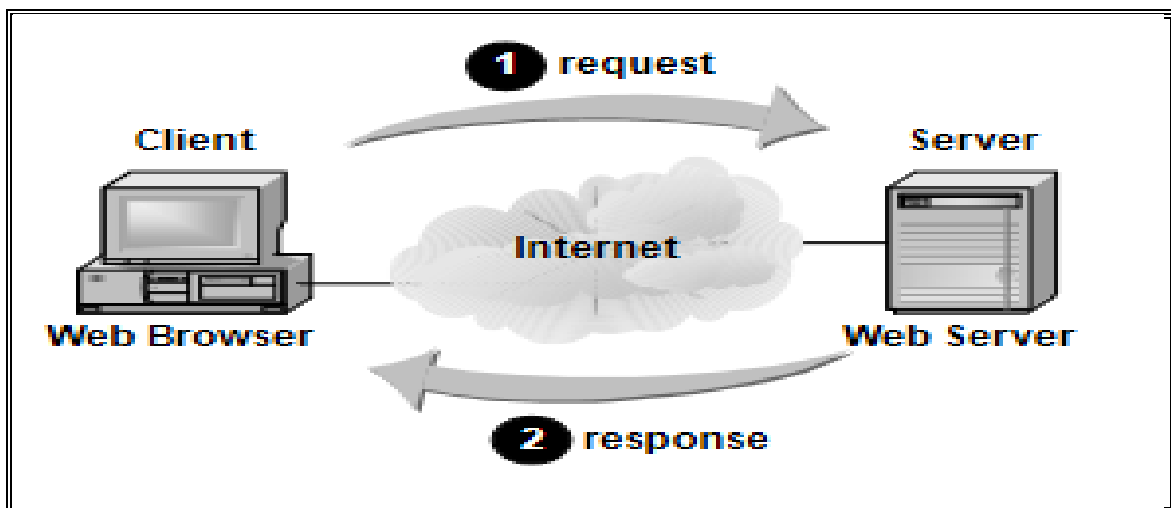


Figure 2-5 Echange de communications entre client serveur

- a) **la connexion TCP** : Comme tout protocole d'application qui utilise tcp, http nécessite une connexion tcp. Étant donné que le client http est responsable du lancement des communications http, le client est également responsable du lancement du processus qui crée la connexion TCP. Comme le montre la figure 2-6, ce processus nécessite l'échange de trois

messages TCP. Les messages tcp sont affichés en texte gris. Après l'échange tcp initial, le client peut envoyer sa requête http. Cette demande et la réponse du serveur sont en texte noir. La figure montre également les messages requis pour fermer une connexion TCP. Le serveur initie cet échange car il sait quand il a répondu à la demande du client. [5]

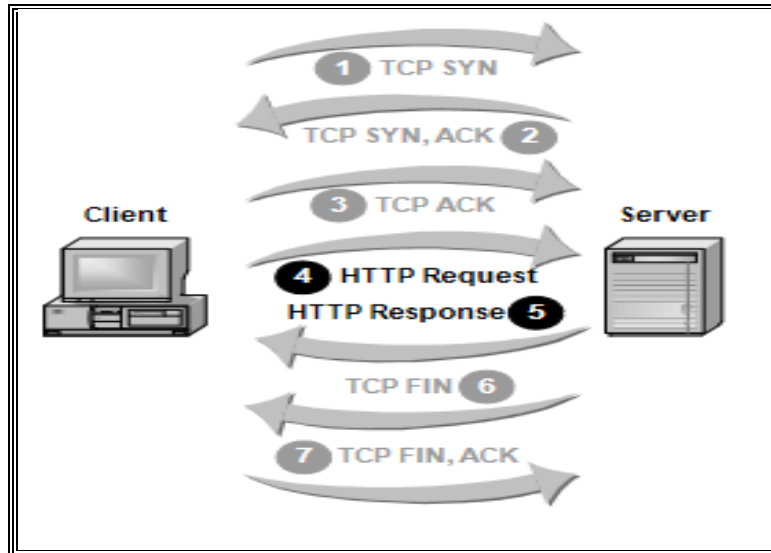


Figure 2-6 Connexion TCP

La figure 2-6 met en évidence les principales caractéristiques des messages TCP. Le premier message envoyé par le client a un indicateur SYN, pour « synchroniser », Le SYN indique que le client souhaite établir une connexion. Le serveur répond en définissant les drapeaux SYN et ACK (pour « accusé de réception »), indiquant sa volonté d'accepter la connexion. Le client termine l'établissement de la connexion en envoyant un message TCP avec uniquement l'indicateur ACK. Ces trois messages sont généralement appelés « prise de contact à trois ». La fermeture de la connexion ne nécessite que deux messages. Le premier a le drapeau FIN (pour « fini »), et le second a les deux drapeaux FIN et ACK définis.

- b) la persistance :** Les premières versions de http obligeaient les clients à établir une connexion TCP distincte à chaque requête. Pour les pages Web simples, cette exigence n'a pas posé beaucoup de problème. À mesure que les sites Web devenaient de plus en plus complexes et graphiques, l'établissement d'une connexion TCP a commencé à avoir un effet notable sur les performances Web. C'est parce que les pages Web complexes se composent de nombreux objets distincts et que le client doit émettre une requête http distincte pour récupérer chacun de ces objets, la version 1.1 du protocole http élimine le problème des connexions TCP multiples avec une fonctionnalité connue sous le nom de persistance. (Bien que la persistance a été introduit dans la version http 1.0, tous les systèmes ne pouvaient pas le prendre en charge ; avec la version 1.1, c'est le comportement par défaut.) La persistance permet à un client de

continuer à utiliser une connexion TCP existante après que sa requête initiale ait été satisfaite. Le client émet simplement une nouvelle requête sur la même connexion. La figure 2-7 montre ce comportement en fonctionnement. [13]

Avec les connexions persistantes, un client peut émettre de nombreuses requêtes HTTP sur une seule connexion TCP. La première requête est à l'étape 4, à laquelle le serveur répond à l'étape 5. À l'étape 6, le client continue en envoyant au serveur une autre requête sur la même connexion TCP. Le serveur répond à cette requête à l'étape 7 puis ferme la connexion TCP. (figure 2-7)

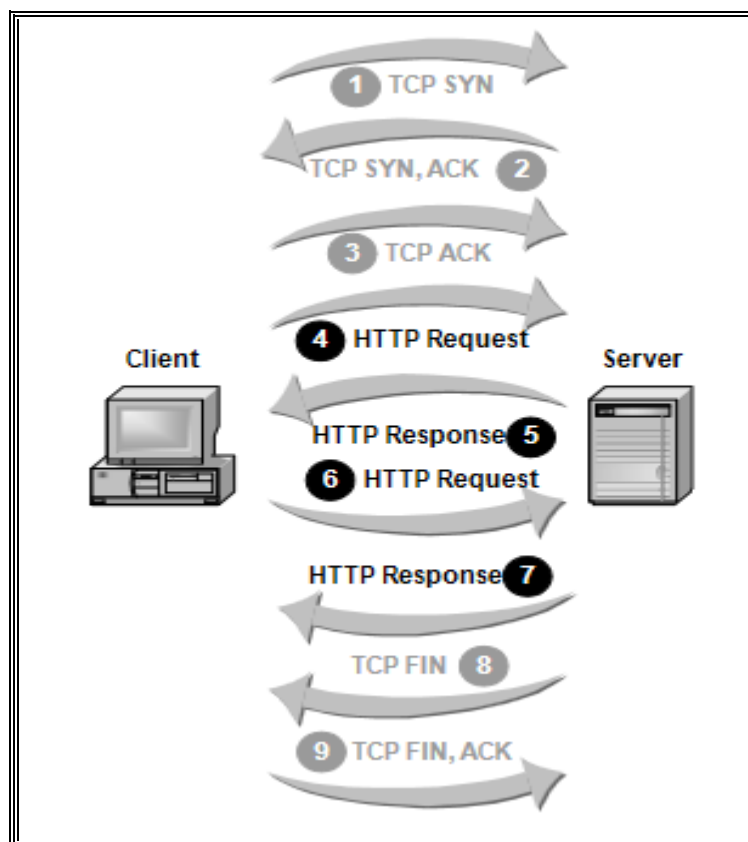


Figure 2-7 La persistance

La persistance nécessite la coopération du client et du serveur. Le client, évidemment, doit prendre la décision d'utiliser une connexion de manière persistante. Il ne peut le faire, cependant, que si le serveur le permet. Le serveur ne doit pas fermer la connexion TCP après avoir répondu à la demande initiale du client.

- c) **le pipeline** : La persistance permet une autre fonctionnalité http qui améliore les performances : le pipeline. Avec le pipeline, un client n'a pas à attendre une réponse à une requête avant d'émettre une nouvelle requête sur la connexion. Il peut suivre

immédiatement la première demande d'une seconde demande. La figure 2-8 montre comment un client peut utiliser le pipeline pour envoyer des requêtes sans attendre les réponses. [5]

Le pipeline permet à un client HTTP d'émettre de nouvelles requêtes sans attendre les réponses de ses messages précédents. Dans la figure 2-8, le client envoie sa première requête à l'étape 4. Il s'ensuit immédiatement une seconde requête à l'étape 5. Le client n'attend pas la réponse du serveur, qui arrive à l'étape 6

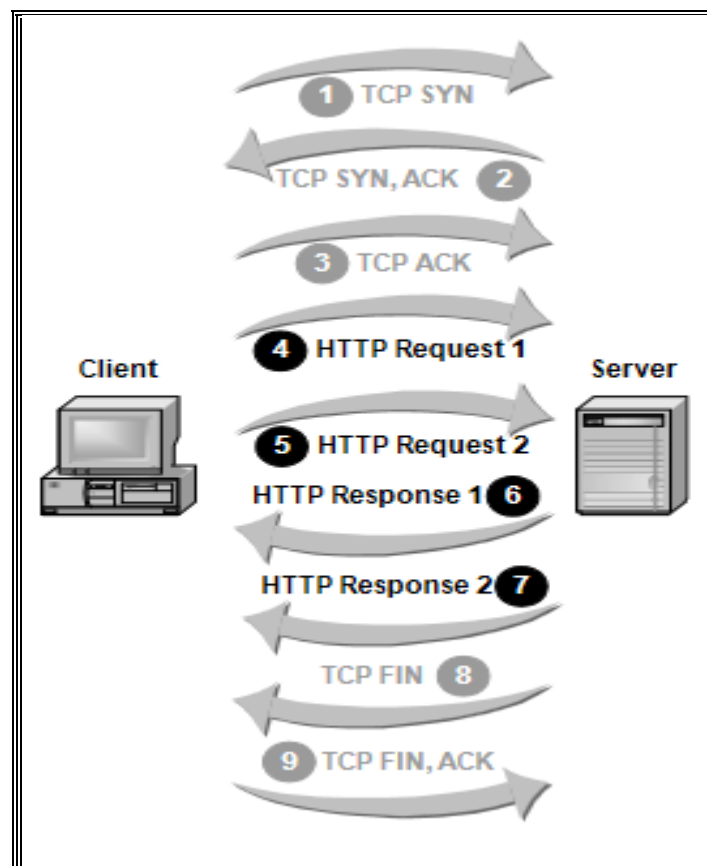


Figure 2-8 La pipeline

2.3.3 HTTP : Mécanisme de travail

Lorsqu'un client souhaite communiquer avec un serveur, que ce soit le serveur final ou un proxy intermédiaire, il effectue les étapes suivantes :

- 1) Ouvrir une connexion TCP : La connexion TCP permet d'envoyer une ou plusieurs requêtes et de recevoir une réponse. Le client peut ouvrir une nouvelle connexion, réutiliser une connexion existante ou ouvrir plusieurs connexions TCP aux serveurs.
- 2) Envoyer un message HTTP : les messages HTTP (avant HTTP/2) sont lisibles par l'homme. Avec HTTP/2, ces messages simples sont encapsulés dans des trames, ce qui les rend impossibles à lire directement, mais le principe reste le même. Par exemple :

```
GET / HTTP/1.1
Host: univ-blida.dz
Accept-Language: fr
```

- 3) Lisez la réponse envoyée par le serveur, par exemple :

```
HTTP/1.1 200 OK
Date: Sun, 01 Aug 2021 16:04:45 GMT
Server: Apache
Last-Modified: Wen, 08 Apr 2020 14:06:13 GMT
ETag: "51142bc1-7449-479b075b2891b"
Accept-Ranges: bytes
Content-Length: 29845
Content-Type: text/html
```

2.3.4 Les certificats de sécurité dans le https

HTTPS fonctionne en utilisant le cryptage à clé publique, qui permet aux serveurs de fournir des clés publiques sous la forme de certificats numériques lorsque les utilisateurs se connectent pour la première fois. Votre navigateur crypte les messages en utilisant cette clé publique, que seul le serveur peut déchiffrer, car lui seul possède la clé privée correspondante. Ce système vous permet de communiquer en toute sécurité avec un site Web sans avoir à connaître à l'avance une clé secrète partagée, ce qui est crucial pour un système comme Internet, où de nouveaux sites Web et utilisateurs vont et viennent chaque seconde de chaque jour. Les certificats numériques sont émis et signés numériquement par diverses autorités de certification (CA) approuvées par le navigateur, c'est pourquoi il est possible d'authentifier que la clé publique est pour le serveur auquel vous vous connectez. Un gros problème avec HTTPS est qu'il indique uniquement que vous vous connectez à ce serveur, pas que le serveur est digne de confiance. Les faux sites de phishing peuvent être configurés facilement avec HTTPS pour un domaine différent mais similaire (exmplebank.com au lieu de

examplebank.com). Les sites HTTPS sont généralement affichés avec un cadenas vert dans les navigateurs Web, ce que de nombreux utilisateurs considèrent comme sûr, mais cela signifie simplement crypté de manière sécurisée.[14]

Certaines autorités de certification effectuent un contrôle supplémentaire sur les sites Web lorsqu'elles émettent des certificats et fournissent un certificat à validation étendue (appelé certificat EV), qui crypte le trafic HTTP de la même manière qu'un certificat normal, mais affiche également le nom de l'entreprise dans la plupart des navigateurs Web, comme illustré à la figure 2-9.

De nombreuses personnes contestent les avantages des certificats EV, principalement parce que la grande majorité des utilisateurs ne remarquent pas le nom de l'entreprise et n'agissent pas différemment sur les sites qui utilisent des certificats EV ou standard validés par domaine (DV). Un juste milieu de certificats à validation organisationnelle (OV) effectue certaines des vérifications mais ne donne pas de notification supplémentaire dans les navigateurs, ce qui les rend largement inutiles au niveau technique (bien que les autorités de certification puissent inclure des engagements de support supplémentaires dans le cadre de leur achat). [14]

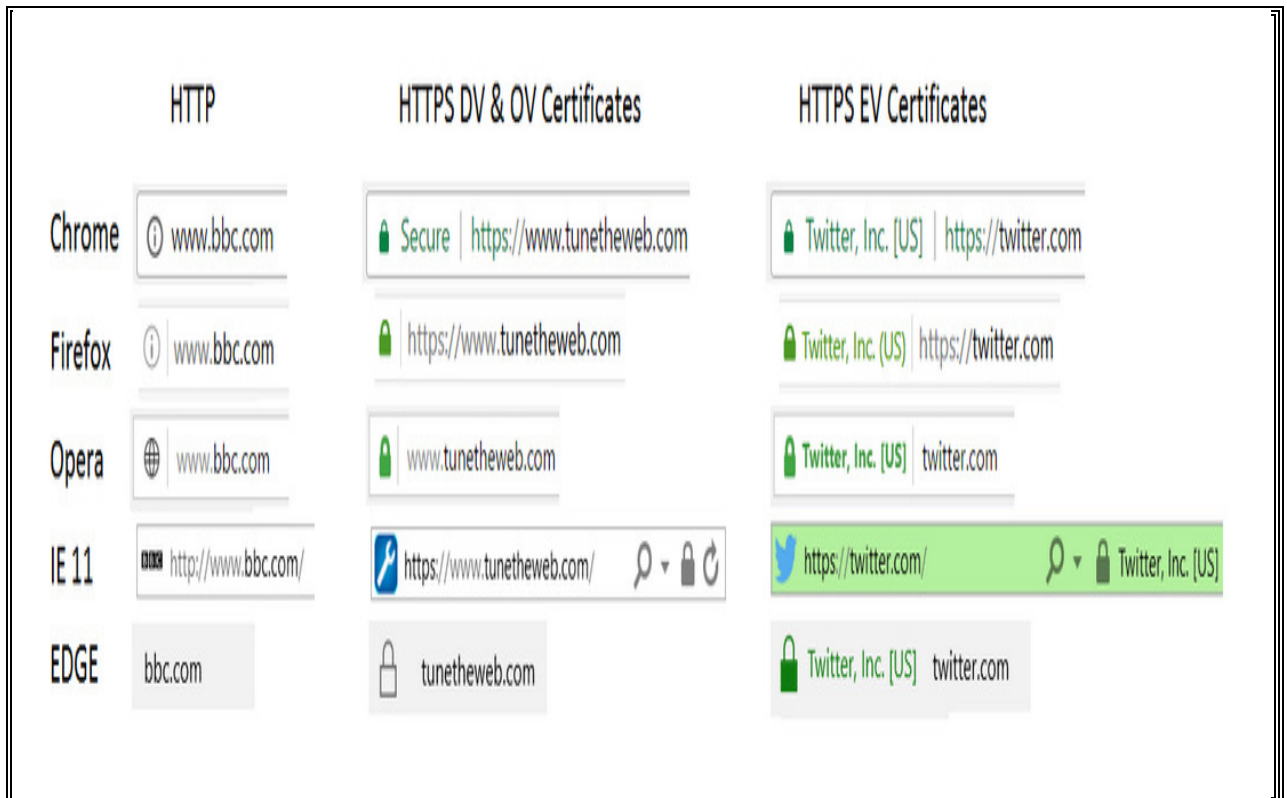


Figure 2-9 Les certificats de sécurité web

HTTPS est construit autour de HTTP et est presque transparent avec le protocole HTTP lui-même. Il est hébergé sur un port différent par défaut (port 443 par opposition au port 80 pour HTTP standard), et il a un schéma d'URL différent (https:// par opposition à http://), mais il ne modifie pas fondamentalement la façon dont HTTP est utilisé en termes de syntaxe ou de format de message, à l'exception du cryptage et du décryptage lui-même.

2.4 Protocole AS2

2.4.1 AS2 Définition

La déclaration d'applicabilité 2 ou spécification AS2 définit un mécanisme pour le transfert sécurisé et fiable de données commerciales structurées sur Internet.[7]

Contrairement aux autres protocoles de trading B2B traditionnels, AS2 offre un environnement de trading sécurisé, efficace et simple à utiliser sans avoir besoin d'appareils propriétaires, de logiciels ou de réseaux privés coûteux ou de réseaux à valeur ajoutée. Certains des principaux avantages de l'utilisation d'AS2 incluent :

- 1) Cryptage des charges utiles réelles échangées : afin que seul le partenaire commercial destinataire puisse déchiffrer un message ou un fichier
- 2) Possibilité de recevoir un reçu signé avec l'assurance que la charge utile a été reçue intacte : fournie par un MDN qui comprend un reçu signé du hachage de charge utile accepté.
- 3) Prévention de l'usurpation d'identité : les messages AS2 signés permettent au destinataire de vérifier que le message provient d'un partenaire commercial de confiance et non d'un imitateur.
- 4) La convivialité Internet et Firewall et les économies de coûts associées : AS2 efface ainsi les VAN (Value Added Networks) qui sont plus complexes, propriétaires et coûteux.[7]

Étant donné que le protocole AS2 fonctionne sur HTTP ordinaire, il peut facilement traverser les pare-feu et utiliser le cryptage SSL de niveau de transport en option et l'authentification HTTP, etc. pour une sécurité supplémentaire. Le protocole AS2 utilise des certificats numériques pour crypter les messages envoyés sur l'Internet public, avec des signatures numériques sur les charges utiles pour garantir l'intégrité et la non-répudiation. Les notifications de disposition de message ou MDN sont des reçus émis par

un récepteur qui sont généralement signés, de sorte que la partie émettrice puisse vérifier que la charge utile a été transmise en toute sécurité sans alternance et acceptée par la partie réceptrice. Un MDN agit ainsi comme un récépissé numérique contraignant pour l'acceptation d'un message, et joue ainsi un rôle clé pour faciliter le commerce B2B sur Internet. En règle générale, une entreprise utilise une application logicielle prenant en charge le protocole AS2 et l'intègre à l'infrastructure informatique et aux systèmes internes existants. Une fois le logiciel AS2 configuré, il est généralement appelé station AS2, le terme « Station locale » implique vos systèmes AS2 et le terme « Station distante » implique une autre station AS2 de l'un de vos partenaires commerciaux. De cette façon, AS2 permet de connecter des systèmes d'application internes à des partenaires externes et à leurs systèmes internes.

N.B

EDI : Echange de Données Informatisé. Il s'agit d'un terme général désignant le transfert de documents entre partenaires commerciaux et les normes auxquelles ils adhèrent. Il est également représenté par EDIINT (EDI sur Internet).

AS2 : ApplicabilityStatement 2. Une norme de transport sécurisé spécifique pour l'EDI, à savoir l'EDI sur HTTPS.

S/MIME : Extensions de messagerie Internet multi-usages sécurisées. C'est l'ensemble des normes utilisées pour le cryptage et la signature d'un message/document. Cela régit non seulement les fonctions de signature et de cryptage, mais fournit également des normes pour le formatage du message final afin qu'un lecteur conforme puisse facilement identifier la structure du message.

MDN : (Message Disposition Notification), est un accusé de réception électronique qu'un partenaire commercial peut éventuellement demander lors d'un échange AS2.

B2B : (business to business), L'abréviation B to B désigne l'ensemble des activités commerciales nouées entre deux entreprises.

MIC : (Message Integrity Check), Un hachage numérique utilisé pour vérifier qu'un message n'a pas été altéré.

2.4.2 AS2 Messages

A) Envoi de messages : Les applications métier du partenaire expéditeur (LOB) qui nécessitent l'échange de documents EDI, transmettront ces documents à l'infrastructure AS2, également connue sous le nom de station AS2. La station AS2

détermine alors le partenaire commercial cible et récupère les informations correspondantes telles que :

- _ Niveau de sécurité requis lors de l'envoi de fichiers à ce partenaire.
- _ Type de requête MDN (sync/async) et identifiants AS2.

La station AS2 émettrice calcule ensuite un hachage numérique sur les fichiers de charge utile source à transmettre, créant un condensé MD5 ou SHA-1, appelé contrôle d'intégrité du message ou MIC.

Les fichiers originaux sont ensuite fusionnés en un message en plusieurs parties, puis signés numériquement à l'aide de la clé privée de la station émettrice, dont la clé publique (et le certificat) seront disponibles auprès du destinataire cible.

Le message en plusieurs parties est ensuite chiffré avec la clé publique du partenaire commercial destinataire, de sorte que seul ce destinataire spécifique qui possède la clé privée correspondante puisse déchiffrer le message.

Le message crypté est désormais transféré via le protocole HTTP/S compatible avec le pare-feu, via l'Internet public ou le lien VPN privé, ainsi que d'autres informations d'en-tête.

Le ou les noms de fichier du document EDI doivent être conservés pendant le transport des messages MIME entre la station AS2 et le partenaire AS2. Ceci sera accompli par l'utilisation des en-têtes Content-Disposition MIME dans le message EDI MIME.

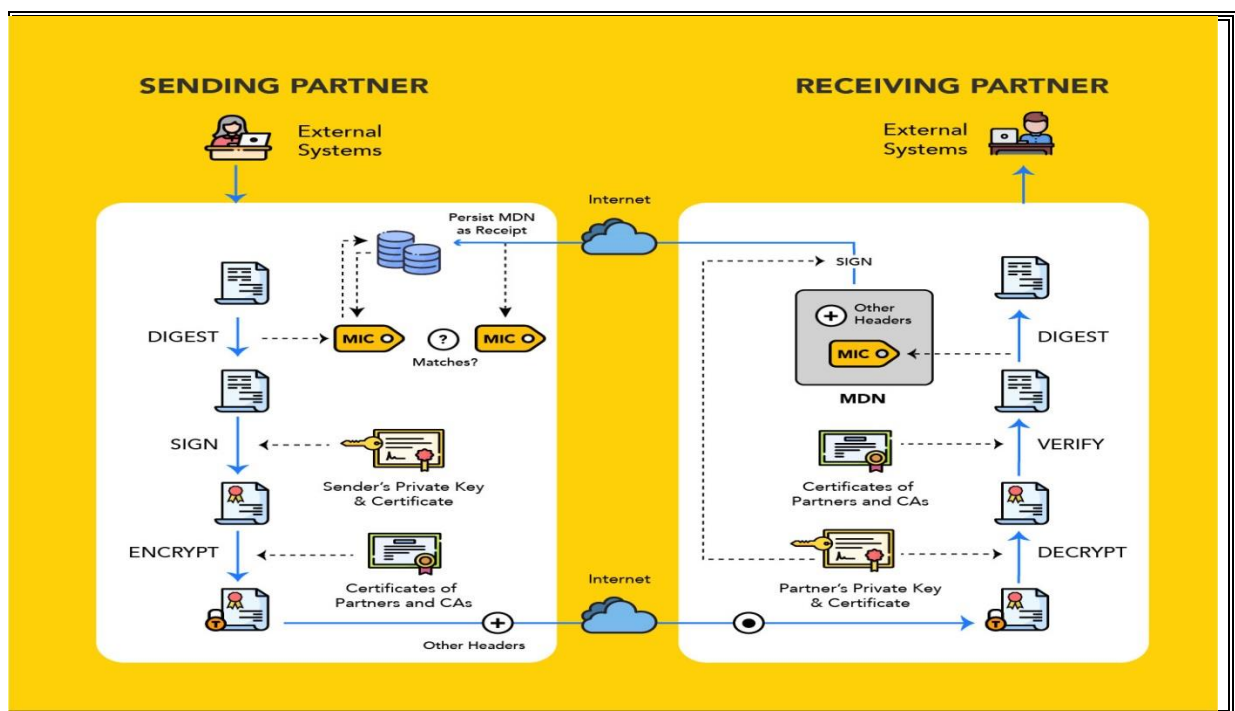


Figure 2-10 AS2 Messages

B) Réception de messages : Une fois que le message atteint la station AS2 du partenaire commercial du destinataire cible, il est déchiffré à l'aide de la clé privée du partenaire. La signature numérique du message déchiffré est ensuite vérifiée par rapport à la liste des certificats de partenaires de confiance, pour s'assurer que :

_ Le message a été signé par le partenaire expéditeur revendiqué

_ Et que le message n'a subi aucune modification lors de la transmission.

Ensuite, le MIC est à nouveau calculé sur la charge utile au niveau de l'application en utilisant le même algorithme que le partenaire expéditeur. Le MIC calculé est généralement signé numériquement par le partenaire commercial destinataire à l'aide de sa clé privée, et cet accusé de réception est renvoyé à l'expéditeur d'origine en tant qu'avis de disposition du message ou MDN.

Le partenaire commercial expéditeur d'origine comparera la valeur MIC dans le MDN reçu avec la valeur MIC déjà stockée du message d'origine. Le MDN reçu sera stocké en tant que garantie numérique d'un envoi réussi du message AS2 au partenaire commercial destinataire.

La station commerciale du partenaire destinataire doit être en mesure d'appliquer la capacité de préservation du nom de fichier en utilisant l'en-tête MIME Content-Disposition.

2.4.3 MDN synchrone et asynchrone

Synchronous ou Sync MDN est une option dans laquelle le MDN est envoyé à l'expéditeur du message via la même connexion HTTP/S qui a été utilisée pour livrer le message EDI d'origine. D'autre part, Asynchrone ou ASync MDN est une option dans laquelle le MDN est envoyé ultérieurement via une connexion HTTP/S différente. [8]

Pour illustrer la différence entre ces deux options :

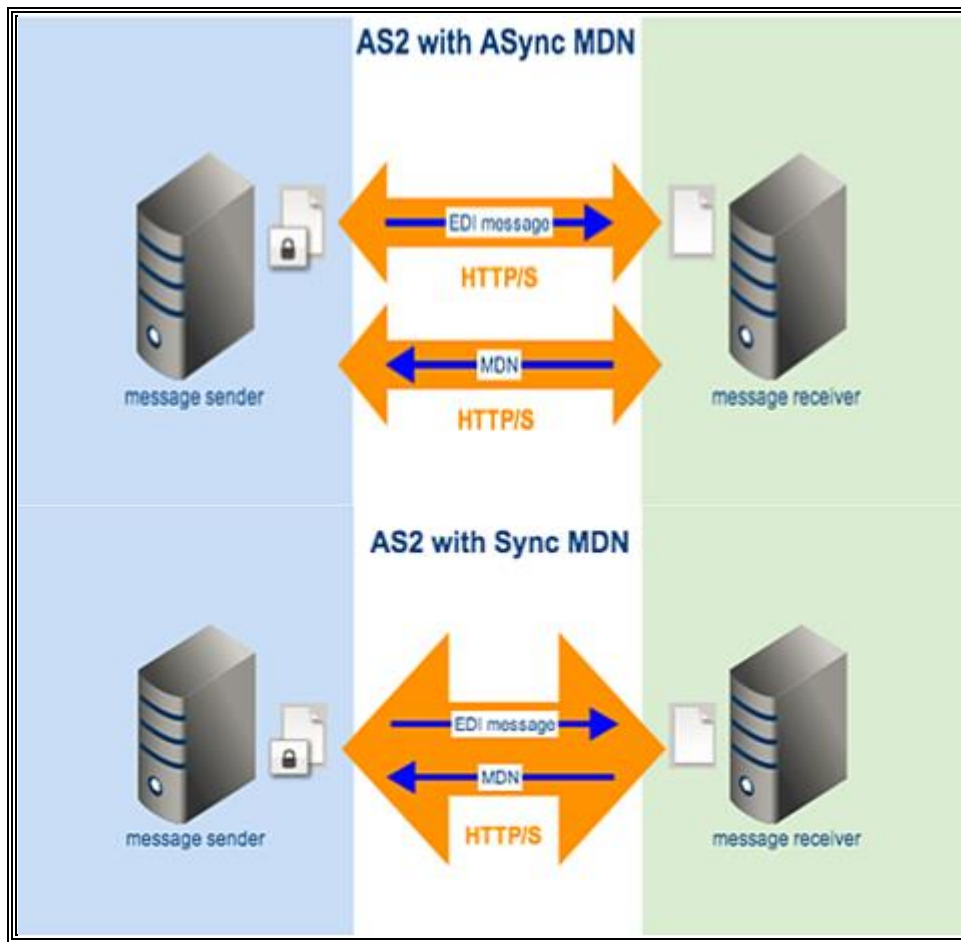


Figure 2-11 MDN Async et Sync

Parce qu'il utilise la même connexion HTTP que la transmission d'origine, Sync MDN est l'option la plus rapide. Cependant, il n'est pas adapté aux transferts de fichiers volumineux car il peut être brusquement annulé pendant le délai d'attente, en particulier dans des conditions de faible bande passante. Donc, si vous devez transférer des fichiers volumineux mais que vous ou votre partenaire commercial êtes soumis à de mauvaises conditions de réseau, alors ASync MDN serait un meilleur choix. [16]

2.5 Le Protocole PeSIT

2.5.1 PeSIT Définition

PeSIT, (Protocol d'Echanges pour un Système Interbancaire de Télécompensation) est un protocole de transfert de fichiers open source de bout en bout développé en 1986 par le Groupement d'Intérêt Economique des Systèmes Interbancaires de Téléclearing (GSIT) .

PeSIT est un protocole de transfert de fichier qui permet l'écriture et la lecture de fichier d'une machine à une autre, reliées par une liaison de télécommunication (liaison spécialisée, réseau public, réseau local).[9]

Afin de s'affranchir des différences entre les systèmes de gestion de fichier propres à chaque machine, PeSIT utilise le concept de fichier virtuel qui est une modélisation commune à toute machine de l'organisation des fichiers. Chaque utilisateur de PeSIT gère la traduction entre le fichier réel et le fichier virtuel transféré par le protocole. Conformément à la terminologie adoptée dans les normes internationales, on distingue : [17]

- le service PeSIT qui est l'interface entre le logiciel de transfert de fichier et son utilisateur
- et le protocole PeSIT qui définit l'ensemble des règles de dialogue et le format des messages échangés entre deux entités PeSIT homologues.

Les Spécifications Techniques de PeSIT décrivent à la fois le service et le protocole PeSIT.

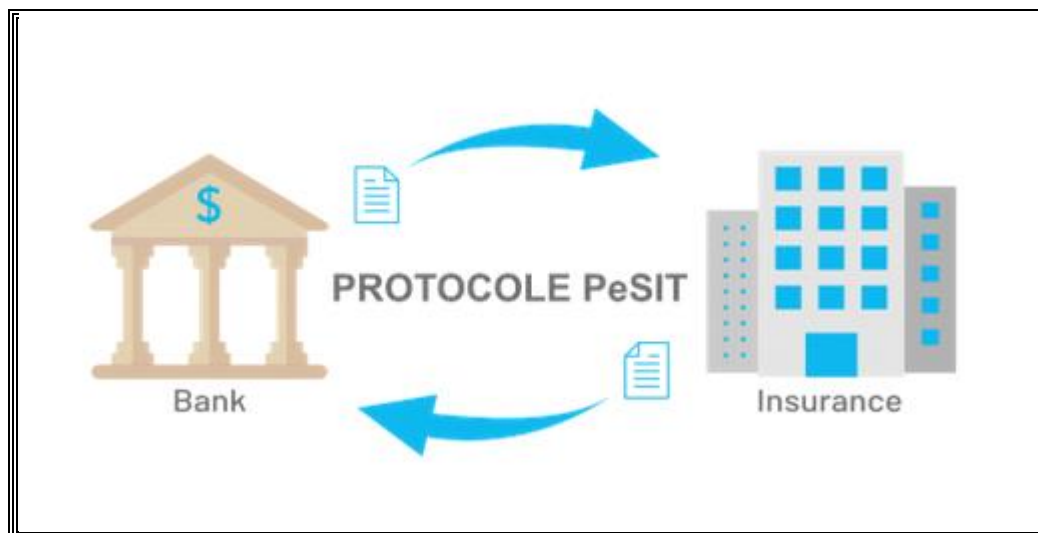


Figure 2-12 Protocole du Bank PeSIT

2.5.2 Les Fonctions de service PeSIT

1- Ecriture (d'un fichier distant): Cette fonction permet à un utilisateur de PeSIT de transférer le contenu d'un fichier à un autre utilisateur de PeSIT.

2- Lecture (d'un fichier distant): Cette fonction permet à un utilisateur de PeSIT de demander qu'un autre utilisateur de PeSIT lui transfère le contenu d'un fichier.

3- Point de contrôle pendant un transfert: cette fonction permet à l'expéditeur de définir des jalons, également appelés points de contrôle, qui sont numérotés séquentiellement pendant le transfert. Le destinataire peut alors accuser réception des points de contrôle, ce qui signifie que les données ont été reçues et enregistrées correctement.[17]

4- Sécurité de transfert: cette fonction permet aux utilisateurs de PeSIT de mettre en œuvre les mécanismes de sécurité suivants:

- Authentification réciproque
- Confidentialité des données transmises
- Intégrité des données transmises
- Non-répudiation réciproque

5- Compression de données: Cette fonction permet aux utilisateurs de mettre en œuvre les mécanismes de compression de données de PeSIT, réduisant ainsi la quantité de données transmises.

6- Contrôle des erreurs: en utilisant un algorithme de détection d'erreur polynomiale appliqué à chaque message du protocole PeSIT, cette fonction permet la détection des messages qui ont été détériorés par un support de transmission non fiable.

7- La suspension de transfert : un utilisateur peut interrompre un transfert (donc de fermer et désélectionner le fichier concerné) pour réutiliser la connexion courante afin de procéder, sur cette connexion, à un transfert de priorité supérieure. Le transfert ainsi suspendu fera par la suite l'objet d'une procédure de reprise.

8- La reprise d'un transfert : l'utilisateur demandeur peut reprendre un transfert interrompu avant son achèvement.

9- La resynchronisation en cours de transfert : un utilisateur peut demander à son partenaire de reprendre le transfert à partir d'un point de synchronisation antérieur, en cas d'incident pendant le transfert des données.

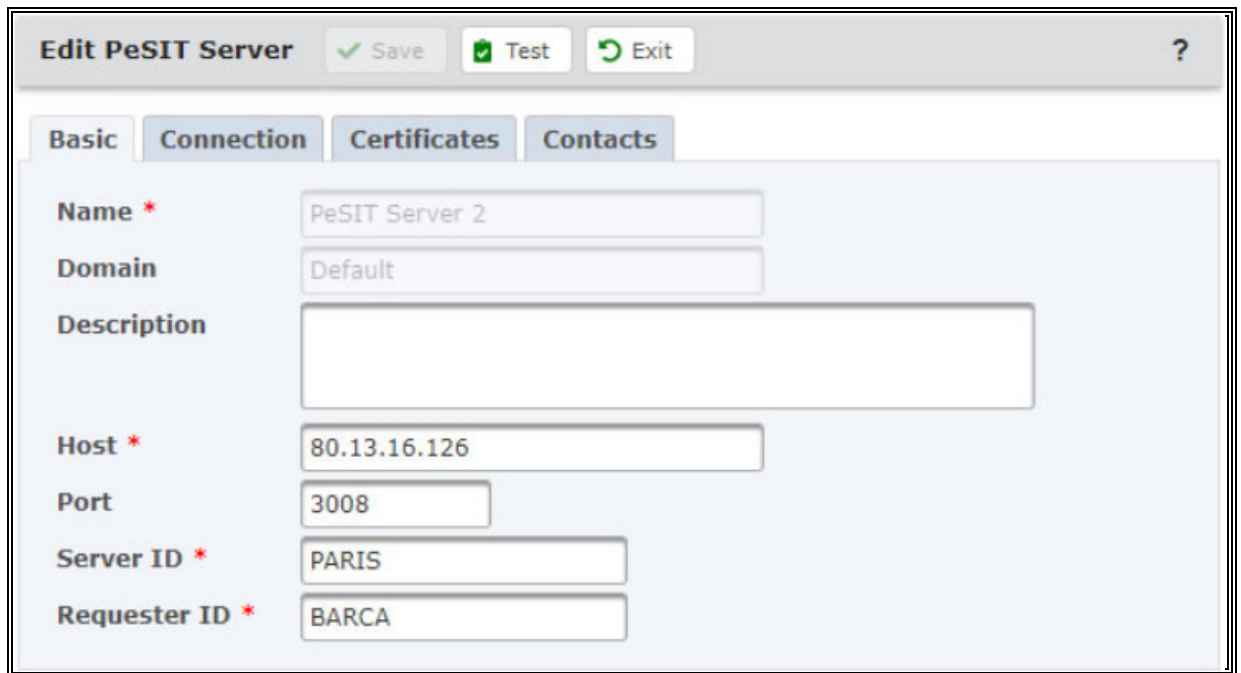


Figure 2-13 Interface de serveur PeSIT

3 Les Applications de Transfert de fichiers

Dans cette partie, nous expliquerons les applications de transfert de fichiers les plus utilisées, car (UI) User interface de ces applications nous aidera beaucoup dans la programmation de notre propre application.

3.1 AirDroid : Description

Apple AirDrop est un moyen rapide et facile de partager des fichiers, des photos, des vidéos, des contacts et plus encore entre vos appareils Apple.



Figure 2-14 Logo de AirDrop

AirDrop utilise la technologie Bluetooth pour créer un réseau Wi-Fi peer-to-peer entre les appareils. Cela signifie que vous n'avez pas besoin d'être connecté à votre routeur ou même à Internet pour avoir une connexion AirDrop. Cependant, vous devez activer le Wi-Fi et le Bluetooth.

Chaque appareil crée un pare-feu autour de la connexion et les fichiers sont envoyés cryptés, ce qui le rend en fait plus sécurisé que la transmission par e-mail. AirDrop détectera automatiquement les appareils pris en charge à proximité, et les appareils doivent être suffisamment proches pour établir une bonne connexion Wi-Fi, permettant le partage de fichiers dans plusieurs pièces

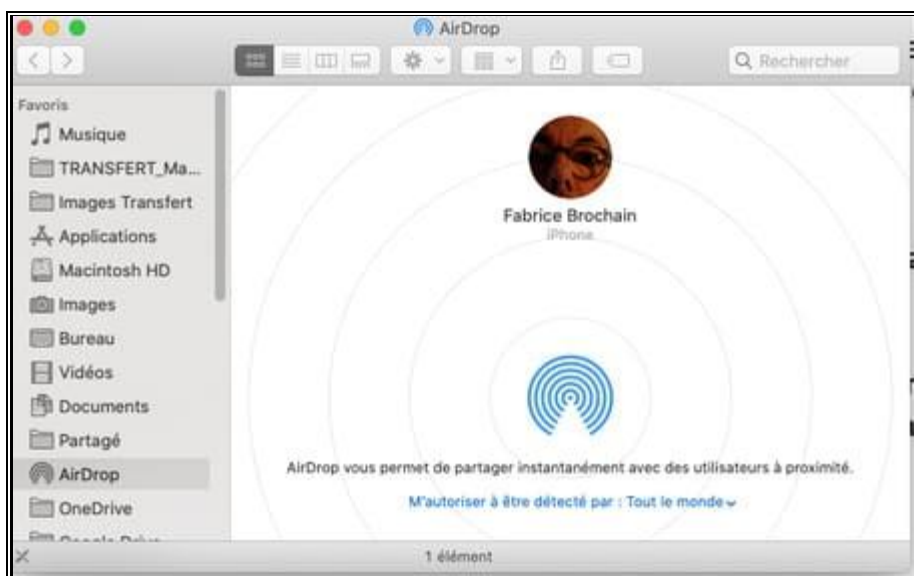


Figure 2-15 L'interface d'AirDrop

3.2 Inshare : Description

InShare est une application incroyablement utile pour partager et recevoir facilement toutes sortes de transferts de fichiers. Si vous souhaitez envoyer des fichiers volumineux à un ami sans perdre de temps ni payer pour un service spécial, cette application est un excellent moyen d'envoyer n'importe quel fichier sur votre smartphone Android en quelques secondes.



Figure 2-16 Logo de Inshare

inshare a une interface propre et belle qui le rend convivial avec plus de 30 langues prises en charge.inShare est gratuit et vous permet de transférer plusieurs fichiers et dossiers e n une seule fois.

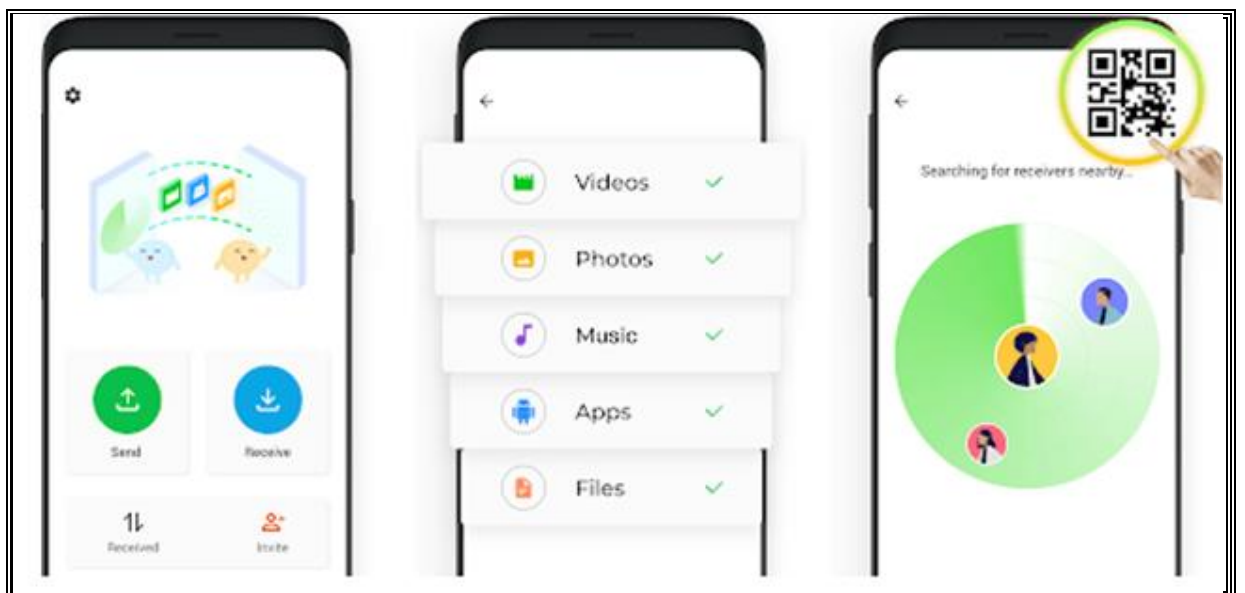


Figure 2-17 Interface d'InShare

3.3 SHAREit : Description

SHAREit est un logiciel gratuit proposé par Lenovo permettant de partager des fichiers entre plusieurs périphériques, qu'il s'agisse de PC, tablette ou smartphone. Pour fonctionner, SHAREit ne nécessite pas de connexion à Internet ou d'être connecté à un réseau puisque le partage s'effectue directement d'un périphérique à l'autre.

Vous pouvez transférer des fichiers pouvant aller jusqu'à 100 Go. Principal avantage de

SHAREit, les fichiers échangés ne transitent pas sur le Cloud vous assurant ainsi la confidentialité de vos données



Figure 2-18 Logo de SHATEit

Une interface élégante, sobre, en français et fonctionnelle permet alors de transférer avec simplicité n'importe quel type de fichier d'un matériel à l'autre.

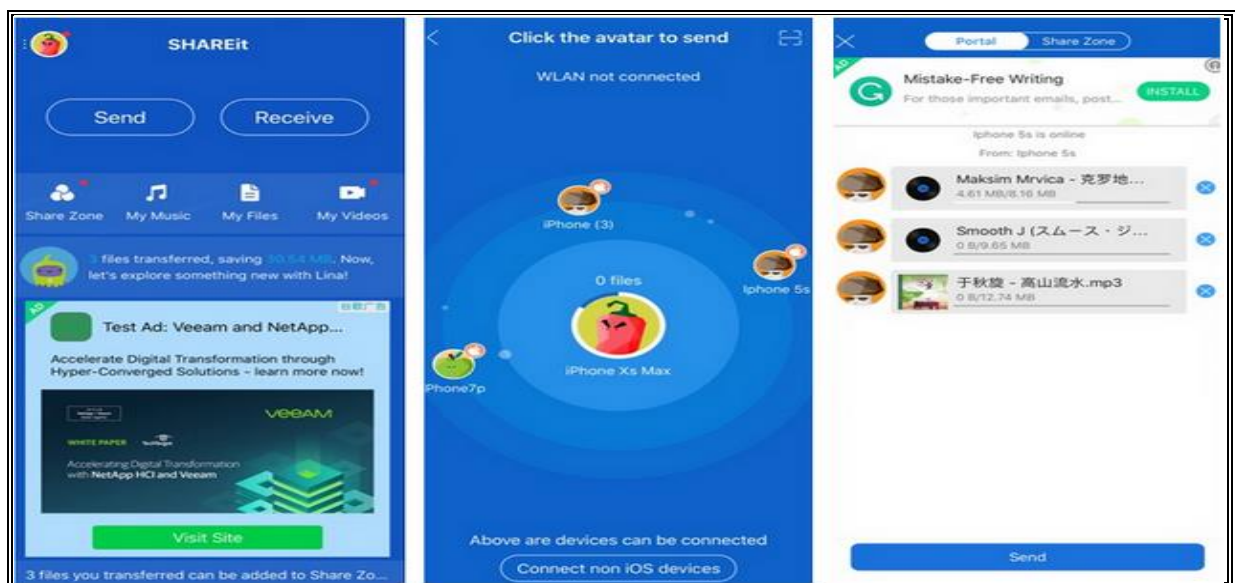


Figure 2-19 Interface de SHATEit

3.4 Xender : Description

Xender est une application de transfert et de partage de fichiers pour Windows, Android et iOS. Documents, musiques, vidéos et applications peuvent être transférés d'un appareil à un autre de façon illimitée et sans connexion.

En effet, il n'est pas nécessaire d'avoir une connexion internet ni de Bluetooth, ni de câble USB ou autre ou encore d'avoir les données mobiles activées. Et pourtant, le partage est rapide, jusqu'à 10 Mo par seconde.



Figure 2-20 Logo Xender

L'application dispose d'une interface claire et fluide et elle est disponible dans plus d'une vingtaine de langues différentes.



Figure 2-21 Interface de Xender

3.5 Zarya : Description

Zarya est un outil de transfert de fichiers peer-to-peer spécialement conçu pour les appareils mobiles. Il offre jusqu'à cinq mégaoctets de transfert de fichiers par seconde et peut gérer des documents volumineux, des présentations, des images, des vidéos, des PDF, des feuilles de calcul et bien d'autres.

Cette application n'a pas besoin d'informations cellulaires ou d'un accès Internet rapide. Il n'a besoin que d'un réseau sans fil pour connecter différents appareils mobiles.



Figure 2-22 Logo Zapya

Zapya possède une interface simple et conviviale qui combine le type de gestion de groupe d'amis simple que vous pouvez trouver dans une application de média social avec la fonction de partage de fichiers.



Figure 2-23 Interface de Zapya

4 Conclusion

L'objectif de ce chapitre était de couvrir le processus de transfert de fichiers de deux côtés, le côté protocole et le côté applications, où nous avons mentionné les protocoles et applications les plus importants actuellement utilisés, et cela nous donnera un aperçu complet de notre sujet de mémoire, ainsi qu'à nous aider à comprendre et à développer notre application de transfert de fichiers.

Chapitre 3 : La Conception du PWA Transfert de fichiers basée Sur le WebRTC

1 Introduction

Comme la plupart des choses dans la vie, l'industrie du Web est soumise à des changements constants, dans un état de recherche permanente de meilleures solutions. Aujourd'hui, l'utilisation des appareils mobiles a connu une augmentation considérable et la tendance est sur le point de se poursuivre dans le futur.

Et tout comme nous pensions que l'industrie était déjà poussée à ses limites, les Progressive Web Apps (PWA), une nouvelle technologie incroyablement prometteuse, ont émergé pour changer de manière irréversible la façon dont les gens interagissent avec les sites Web. Même si la technologie existe depuis un certain temps. maintenant, nous voyons que la popularité des PWA a grimpé en flèche au cours des deux dernières années.

Les PWA combinent le meilleur du site Web avec le meilleur de ce que les applications natives ont à offrir, créant une expérience utilisateur mobile phénoménale. Ce sont des applications Web spéciales accessibles comme des sites Web normaux, mais offrant des avantages tels que la convivialité hors ligne, les notifications push et l'accès au matériel de l'appareil - des éléments qui n'étaient auparavant disponibles que pour les applications natives. L'un des plus grands défis de WebRTC a été de fournir un support cohérent et fiable sur toutes les plateformes. Pour la plupart des applications, en particulier celles qui ont démarré sur le Web, cela signifie généralement développer une application mobile native ou hybride en plus de prendre en charge l'application Web. Les applications Web progressives (PWA) sont un nouveau concept qui promet d'unifier le Web pour de nombreuses applications en permettant aux applications Web de ressembler et d'agir comme des applications mobiles natives sans introduire un cadre hybride intermédiaire. Comme nous le verrons dans ce chapitre à travers lequel nous utiliserons les avantages de PWA et du WebRTC ensemble pour concevoir notre propre projet de transfert de fichiers .

2 La technologie WebRTC : concept et description

WebRTC (Web Real Time Communications) est un cadre ouvert (open framework) pour le Web qui permet les communications en temps réel dans le navigateur. Il comprend les blocs

de construction fondamentaux pour des communications de haute qualité sur le Web, tels que les composants réseau, audio et vidéo utilisés dans les applications de chat vocal et vidéo et de partage de fichiers. [18]

Ces composants, lorsqu'ils sont implémentés dans un navigateur, sont accessibles via une API JavaScript, permettant aux développeurs d'implémenter facilement leur propre application Web RTC.[18]

L'histoire du WebRTC a débuté en Mai 2010 avec le rachat de la société Global IP Solutions (GIPS) par Google. L'objectif de ce rachat est plus technologique que business. En effet, un an après le rachat de GIPS, Google publie sous licence open source les technologies issues de ce rachat, servant de fondation au projet WebRTC.

Dès 2011, Google s'associe à différents éditeurs comme Mozilla et Opéra ainsi qu'aux organismes de standardisations de W3C et de l'IETF pour développer le WebRTC et en faire un standard technologique ouvert.



Figure 3-1 Logo de WebRTC

Le WebRTC est ainsi devenu le nouveau standard de communication peer-to-peer entre navigateurs web. Il permet notamment à tout internaute disposant d'une application web VoIP compatible, de passer des appels ou de réaliser des visioconférences directement au sein de son navigateur. Plus besoin d'extension, de widgets ou de plugins, tout est 100% natif au navigateur. [18]

Les principes fondateurs du projet WebRTC sont que ses APIs doivent être open source, gratuites, standardisées, compatibles avec les principaux navigateurs, tout en offrant une couverture fonctionnelle supérieure aux solutions existantes.

Le WebRTC est aujourd'hui pleinement intégré aux navigateurs Google Chrome, Mozilla Firefox et Opera. Il est aujourd'hui utilisé dans des usages aussi variés que les jeux vidéos, le transfert de fichiers, le streaming, la messagerie instantanée ou tout autre application nécessitant de la communication en temps réel.[18]

Le WebRTC vise à grandement simplifier l'expérience de voix sur IP dans un contexte Web. Quoi de plus ennuyeux que de devoir télécharger une application spécifique avant une visioconférence. Le WebRTC permet de s'affranchir de tout plugin et de directement se connecter à une chambre de conférence, de passer un appel, de partager des documents, depuis son navigateur. (voir la Figure 3-2)

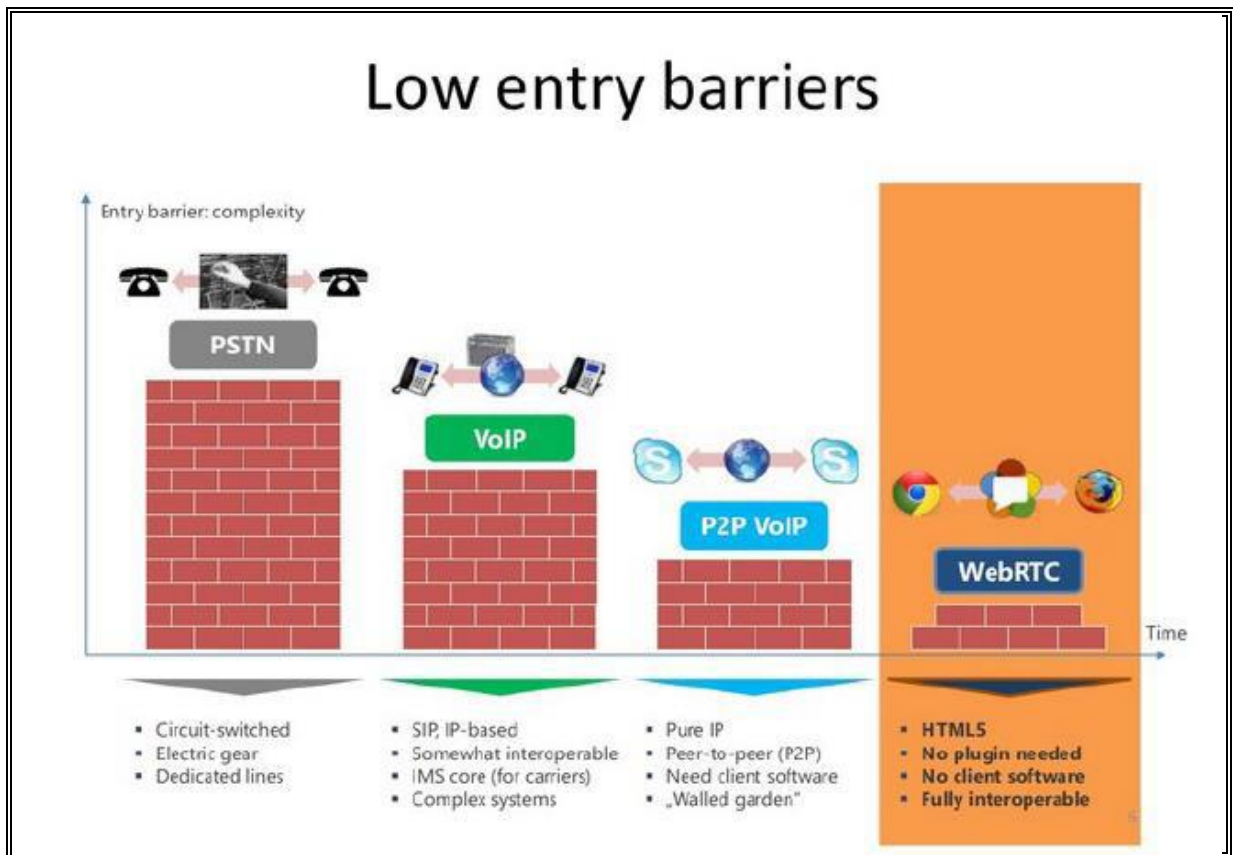


Figure 3-2 Historique des solutions de voix et vidéo sur IP

2.1 WebRTC et Compatibilité sur le navigateur

Une application WebRTC (généralement écrite comme un mélange de HTML et JavaScript) interagit avec les navigateurs Web via l'API WebRTC normalisée, lui permettant d'exploiter et de contrôler correctement la fonction de navigateur en temps réel (voir la figure 3-3). L'application WebRTC interagit également avec le navigateur, en utilisant à la fois WebRTC et d'autres API standardisées, à la fois de manière proactive (par exemple, pour interroger les capacités du navigateur) et de manière réactive (par exemple, pour recevoir des notifications générées par le navigateur). [19]

L'API WebRTC doit donc fournir un large éventail de fonctions, comme la gestion des connexions (de façon poste à poste), la négociation, la sélection et le contrôle des capacités de codage / décodage, le contrôle des médias, le pare-feu et la traversée des éléments NAT, etc...

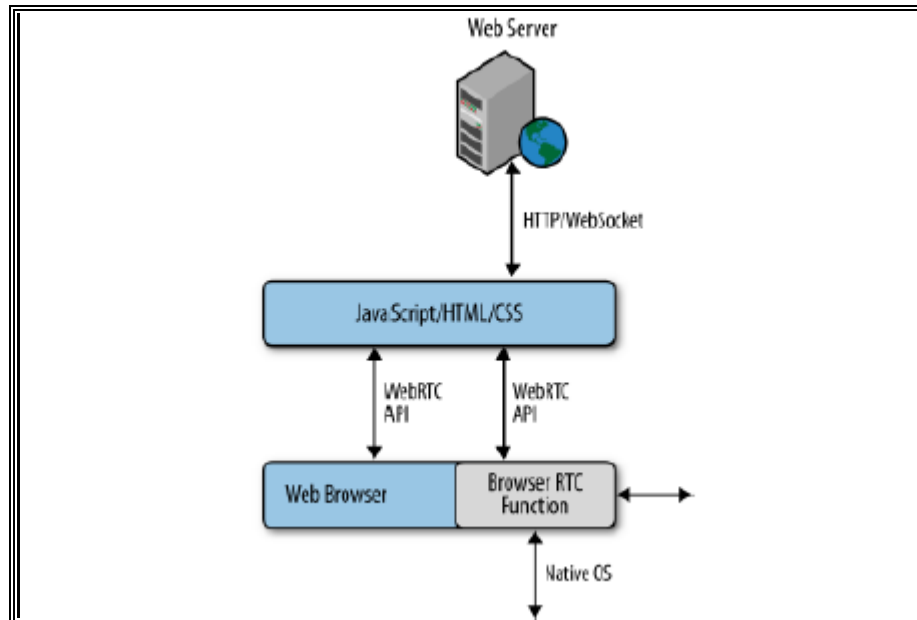


Figure 3-3 Communication en temps réel dans le navigateur

La prise en charge actuelle de WebRTC dans l'espace du navigateur est illustrée dans la section suivante

1) Compatibilité avec Chrome, Firefox et Opera

Les dernières versions de Chrome, Firefox et Opera sur les principaux systèmes d'exploitation PC tels que Mac OS X, Windows et Linux, prennent toutes en charge WebRTC prêt à l'emploi. Et plus important encore, les ingénieurs des équipes de développeurs Chrome et Firefox ont travaillé ensemble pour résoudre les problèmes afin que ces deux navigateurs puissent communiquer facilement entre eux. [19]

2) Compatibilité avec Android OS

C'est également le cas pour les systèmes d'exploitation Chrome, Firefox et Android. Les applications basées sur WebRTC doivent fonctionner immédiatement et être capables d'interagir avec d'autres navigateurs après Android version 4.0 (IceCream Sandwich). Cela est

dû à la notion de partage de code entre les versions de bureau et mobile de Chrome et Firefox[19]

3) Compatibilité avec Apple

Apple a fait peu d'efforts pour activer WebRTC dans Safari ou iOS. Il y a des rumeurs de soutien, mais aucune date officielle ne précise la date à laquelle le soutien interviendra. La solution de contournement que d'autres ont utilisée pour les applications iOS hybrides natives / Web consiste à incorporer le code WebRTC directement dans leurs applications et à charger une application WebRTC dans une WebView[19]

4) Compatibilité avec Internet Explorer

Microsoft ne prend pas en charge WebRTC sur les ordinateurs de bureau. Mais ils ont officiellement confirmé qu'ils allaient implémenter ORTC (Object Realtime Communications) dans les futures versions d'IE (Edge). Ils ne prévoient pas de prendre en charge WebRTC 1.0. Ils ont étiqueté leur ORTC comme WebRTC 1.1, bien qu'il ne s'agisse que d'une amélioration de la communauté et non de la norme officielle. Récemment, ils ont ajouté le support ORTC à la dernière version de Microsoft Edge.

2.2 Architecture WebRTC

L'architecture WebRTC implique des serveurs Web et des navigateurs clients. Le serveur Web "alimente" les applications Web avec Javascript intégré, et les clients navigateurs (PC, tablettes, smartphones) exécutent l'application Javascript. Traditionnellement, les navigateurs Web communiquent uniquement avec les serveurs Web. Ce qui est unique à propos de WebRTC, c'est que l'application Web peut désormais activer les communications peer-to-peer (P2P) entre deux navigateurs clients .[20]

2.2.1 Le Triangle WebRTC :

Initialement, le scénario le plus courant est probablement celui où les deux navigateurs exécutent la même application Web WebRTC, téléchargée à partir de la même page Web. Cela produit le « Triangle » WebRTC illustré à la figure 3-4. Cet arrangement est appelé un triangle en raison de la forme de la signalisation (côtés du triangle) et des flux de médias ou de données (base du triangle) entre les trois

éléments. Une connexion homologue établit le transport pour les médias vocaux et vidéo et les flux de canaux de données directement entre les navigateurs.[20]

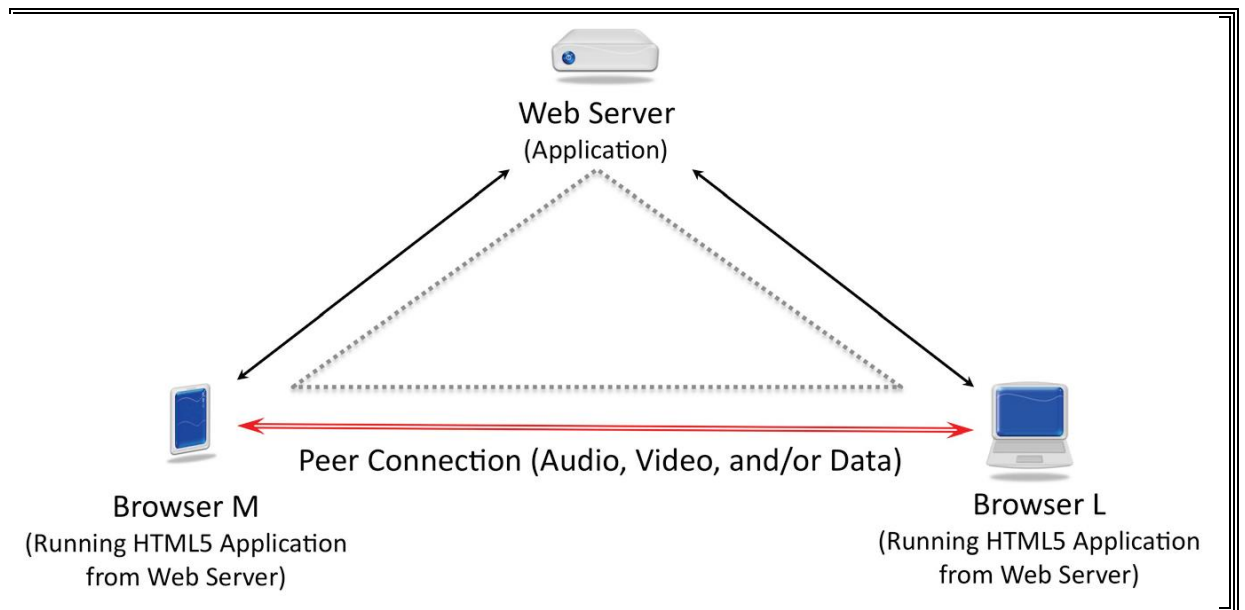


Figure 3-4 Le Triangle WebRTC

Notez que même si nous appelons parfois la connexion entre le navigateur et le serveur la signalisation, il ne s'agit pas vraiment de signalisation telle qu'elle est utilisée dans les systèmes de téléphonie. La signalisation n'est pas standardisée dans WebRTC car elle est simplement considérée comme faisant partie de l'application. Cette signalisation peut s'exécuter via HTTP ou WebSockets vers le même serveur Web qui sert les pages HTML au navigateur, ou vers un serveur Web complètement différent qui gère simplement la signalisation. [20]

2.2.2 Trapèze WebRTC :

La figure 3-5 montre le trapèze WebRTC, basé sur le trapèze SIP [RFC3261]. Les deux serveurs Web sont montrés en train de communiquer à l'aide d'un protocole de signalisation standard tel que le protocole d'initiation de session (SIP), utilisé par de nombreux systèmes VoIP et de vidéoconférence, ou Jingle [XEP-0166], utilisé pour ajouter des capacités vocales et vidéo à Jabber [RFC6120] messagerie instantanée et systèmes de présence. En variante, un protocole de signalisation propriétaire pourrait être utilisé. Notez que dans ces cas plus compliqués, le média peut ne pas circuler

directement entre les deux navigateurs, mais peut passer par des relais média et d'autres éléments.[20]

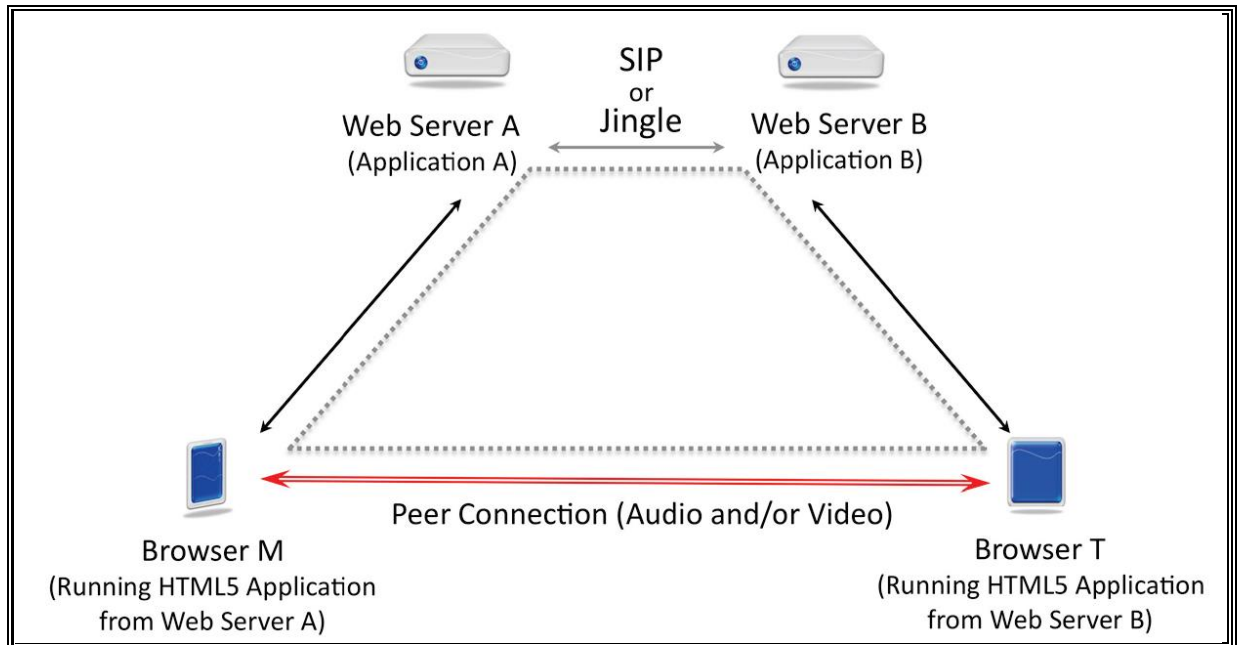


Figure 3-5 Trapèze WebRTC

2.3 La signalisation

2.3.1 Description

Pour vous connecter à un autre utilisateur, vous devez savoir où il se trouve sur le Web. L'adresse IP de votre appareil permet aux appareils connectés à Internet d'envoyer des données directement entre eux. L'objet `RTCPeerConnection` en est responsable. Dès que les appareils savent comment se trouver sur Internet, ils commencent à échanger des données sur les protocoles et les codecs pris en charge par chaque appareil. [21]

Pour communiquer avec un autre utilisateur, vous avez simplement besoin d'échanger des informations de contact et le reste sera fait par WebRTC. Le processus de connexion à l'autre utilisateur est également appelé signalisation et négociation. [21]

Il se compose de quelques étapes :

- Créer une liste de candidats potentiels pour une connexion entre pairs.
- L'utilisateur ou une application sélectionne un utilisateur avec lequel établir une connexion.

- La couche de signalisation notifie à un autre utilisateur que quelqu'un souhaite se connecter à lui. Il peut accepter ou refuser.
- Le premier utilisateur est averti de l'acceptation de l'offre.
- Le premier utilisateur initie RTCPeerConnection avec un autre utilisateur.
- Les deux utilisateurs échangent des informations logicielles et matérielles via le serveur de signalisation.
- Les deux utilisateurs échangent des informations de localisation.
- La connexion réussit ou échoue.

La spécification WebRTC ne contient aucune norme sur l'échange d'informations. Gardez donc à l'esprit que ce qui précède n'est qu'un exemple de la façon dont la signalisation peut se produire. Vous pouvez utiliser n'importe quel protocole ou technologie que vous aimez. [21]

2.3.2 Les serveurs ICE (STUN/TURN)

Le réseau Internet est en fait constitué de plusieurs millions de réseaux publics et privés reliés par de nombreux câbles, routeurs et une panoplie d'autres dispositifs comme des pare-feu ou encore des routeurs NAT (Network Address Translation) ainsi que différentes restrictions déterminées par votre fournisseur de service Internet (ISP).[21]

La majorité des appareils disposent d'une adresse IP privée (adresse non unique d'un point de vue global) et nécessite un dispositif NAT qui s'occupe de transformer l'adresse privée en adresse publique (ainsi que les ports) pour pouvoir accéder à l'Internet. En réalité, dans la grande majorité des cas, deux firewalls ne pourront tout simplement pas communiquer directement ensemble puisqu'aucun des deux ne connaît l'adresse IP et le port qu'il peut utiliser pour atteindre l'autre.

C'est ici que la technique ICE (Interactive Connectivity Establishment) intervient pour permettre aux agents WebRTC de communiquer en faisant abstraction de la complexité des réseaux formant la réalité d'Internet. ICE va utiliser des serveurs Session Traversal Utilities for NAT (STUN) et TraversalUsingRelaysaround NAT (TURN) pour palier aux problèmes causés par les pare-feu, les routeurs NAT et autres restrictions potentielles.[21]

Un serveur STUN permet à l'agent de connaître les informations réseaux qu'il expose à l'externe, soit son adresse IP publique, le port, mais aussi le type de routeur NAT devant lui.

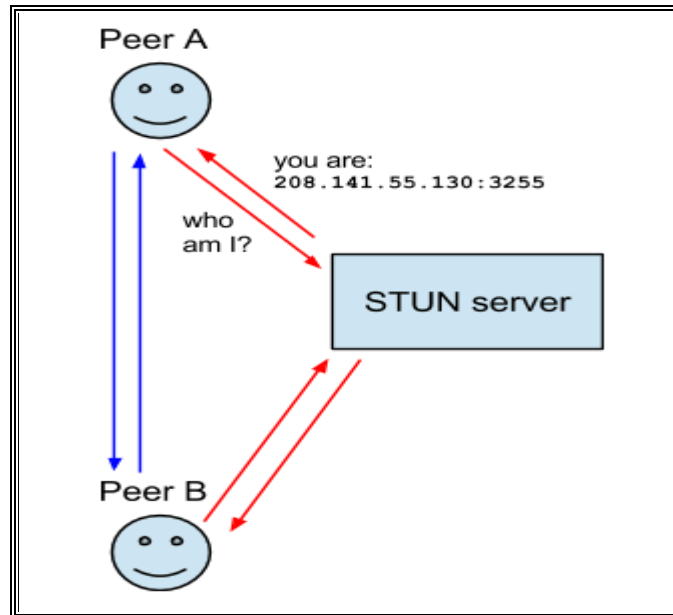


Figure 3-6 Serveur STUN

Un serveur TURN est une extension à STUN lorsque celui-ci ne parvient pas à établir la connexion. Il va permettre de relayer le média d'un agent à l'autre et devra être présent durant toute la communication contrairement à STUN.

TURN est une opération définitivement plus exigeante et la communication directe devrait en tout temps être privilégiée.[21]

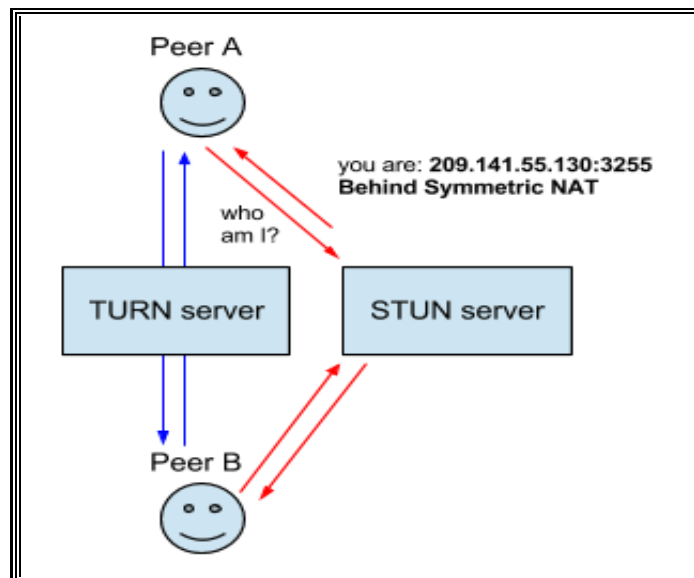


Figure 3-7 Serveur TURN

En réalité, ICE fait partie intégrante de l'étape d'initialisation de la communication et voici à quoi ressemble notre nouvelle architecture après avoir ajouté nos serveurs STUN et TURN:

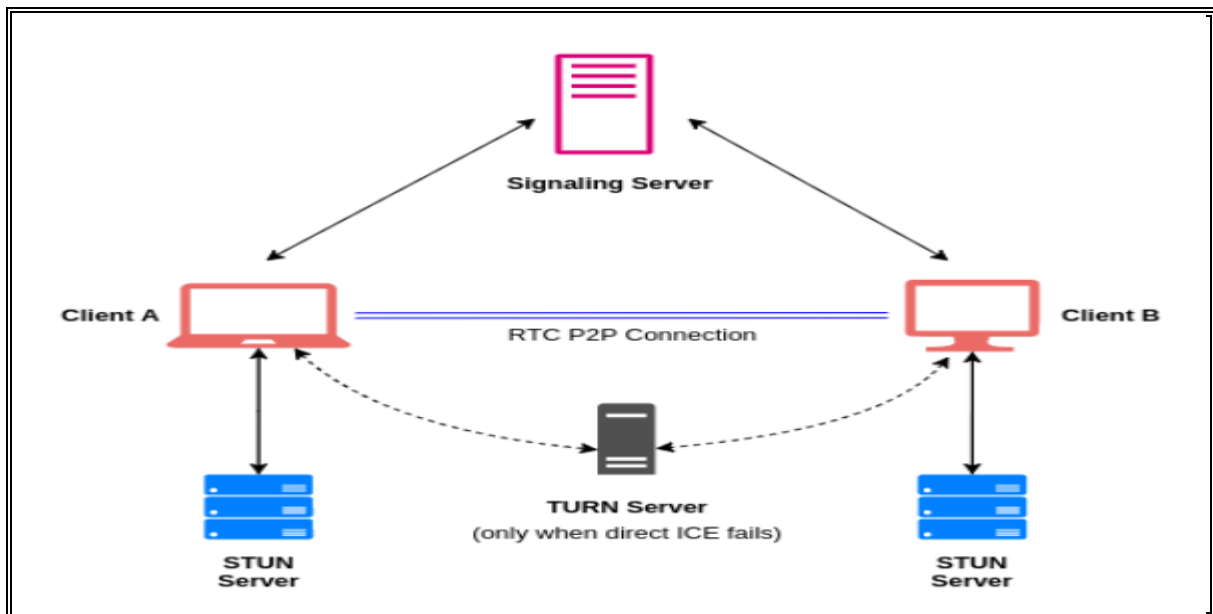


Figure 3-8 Architecture WebRTC

2.4 Les API WebRTC

Le terme API est un acronyme qui signifie « Application Programming Interface ».

Pensez à une Application Programming Interface comme un menu de restaurant. Le menu fournit une liste de plats que vous pouvez commander, ainsi qu'une description de chaque plat. Lorsque vous spécifiez les éléments du menu que vous souhaitez, la cuisine du restaurant fait le travail et vous fournit le plats en question. Vous ne savez pas exactement comment le restaurant prépare ces plats, et vous n'avez pas vraiment besoin de le savoir.

De même, une Application Programming Interface énumère un ensemble d'opérations que les développeurs peuvent utiliser, ainsi qu'une description de ce qu'ils font. Le développeur n'a pas nécessairement besoin de savoir comment, par exemple, un système d'exploitation se construit, Il doit juste savoir que cette fonction est disponible pour être utilisée dans son application.[22]

Le WebRTC repose sur trois API JavaScript intégrées directement dans les navigateurs Web ne nécessitant aucun client ou plug-in de navigateur afin de communiquer directement avec un autre navigateur compatible WebRTC. Ces API sont :

2.4.1 **MediaStream (alias getUserMedia) :**

vous permet d'accéder à la caméra, au microphone ou à l'écran de l'appareil utilisé par l'utilisateur. En tant que couche de sécurité supplémentaire, l'utilisateur aura un accès autorisé avant que vous ne soyez autorisé à diffuser ses médias. Si l'utilisateur se connecte à partir d'une connexion sécurisée (HTTPS), l'utilisateur n'aura besoin d'accorder l'accès qu'une seule

fois pour l'application, mais si vous vous connectez à partir d'une connexion non sécurisée (HTTP), l'utilisateur sera invité à chaque fois que l'application aura besoin d'accéder [22]

2.4.2 RTCPeerConnection (alias PeerConnection) :

Permet à deux utilisateurs de communiquer directement, d'égal à égal. Il encode et décode les médias envoyés vers et depuis votre machine locale vers un homologue distant recevant vos médias.

2.4.3 RTCDataChannel (alias DataChannel) :

Représente un canal de données bidirectionnel entre deux homologues. Il se superpose au RTCPeerConnection, ce qui vous permet d'envoyer des données directement entre les deux pairs connectés en toute sécurité. [22]

2.5 RTCDataChannel : Description et exemple d'utilisation

RTCDataChannel est responsable de l'échange de toutes les données en temps réel qui ne sont pas audiovisuelles. Cela signifie des chats textuels, le partage de fichiers peer-to-peer et d'autres formes traditionnelles de données qui ne contiennent pas d'audiovisuel.

RTCDataChannel peut être exploité par le développeur astucieux pour créer des solutions nouvelles et rentables à certains des problèmes les plus anciens d'Internet. [18]

Il existe certainement des technologies existantes sur Internet qui peuvent offrir des piliers tels que le chat textuel et le partage de fichiers avec une relative facilité, mais il y a quelques considérations sous le capot qui font de RTCDataChannel une option attrayante pour les développeurs qui souhaitent incorporer des données traditionnelles en temps réel dans leur applications. La nature peer-to-peer de RTCDataChannel signifie que les développeurs peuvent éviter la perspective d'avoir à maintenir un équipement côté serveur coûteux pour faciliter les transferts de données.

Avec des technologies telles que WebSocket, AJAX et les événements côté serveur, certains peuvent considérer l'option d'un autre canal de données comme redondante. Mais RTCDataChannel offre quelques distinctions clés qui le séparent des autres choix. D'une part, il peut être utilisé avec l'API RTCPeerConnection de WebRTC pour activer automatiquement la communication peer-to-peer. Cela évite le besoin d'un serveur d'initiation, ce qui conduit à

une latence plus faible. La sémantique de livraison du protocole SCTP (Stream Control Transmission Protocol) de RTCDataChannel permet une configuration de livraison et de retransmission asynchrone. Cela signifie que les données peuvent arriver de la manière la plus rapide possible, plutôt que dans un ordre séquentiel rigide.

Comme pour toutes les fonctionnalités WebRTC, le cryptage de RTCDataChannel est requis. Afin de transmettre des données, RTCDataChannel doit sécuriser les données transmises avec Datagram Transport Layer Security (DTLS). SSL est le modèle parent de DTLS, ce qui signifie que les données WebRTC seront sécurisées via n'importe quelle connexion SSL standard. DTLS est un protocole standardisé intégré à tous les navigateurs prenant en charge WebRTC. Cela signifie que les données peuvent être partagées de manière privée, d'égal à égal, à l'abri de l'espionnage par des tiers. [18]

Et maintenant, nous allons concevoir une page Web à travers laquelle nous pouvons transférer des fichiers, le but de la création de cette page est de comprendre comment utiliser les APIs de WebRTC dans cet exemple

Nous devons d'abord créer un fichier dans lequel nous ouvrirons trois fichiers de type HTML et CSS et JAVASCRIPT comme le montre la figure 3-9

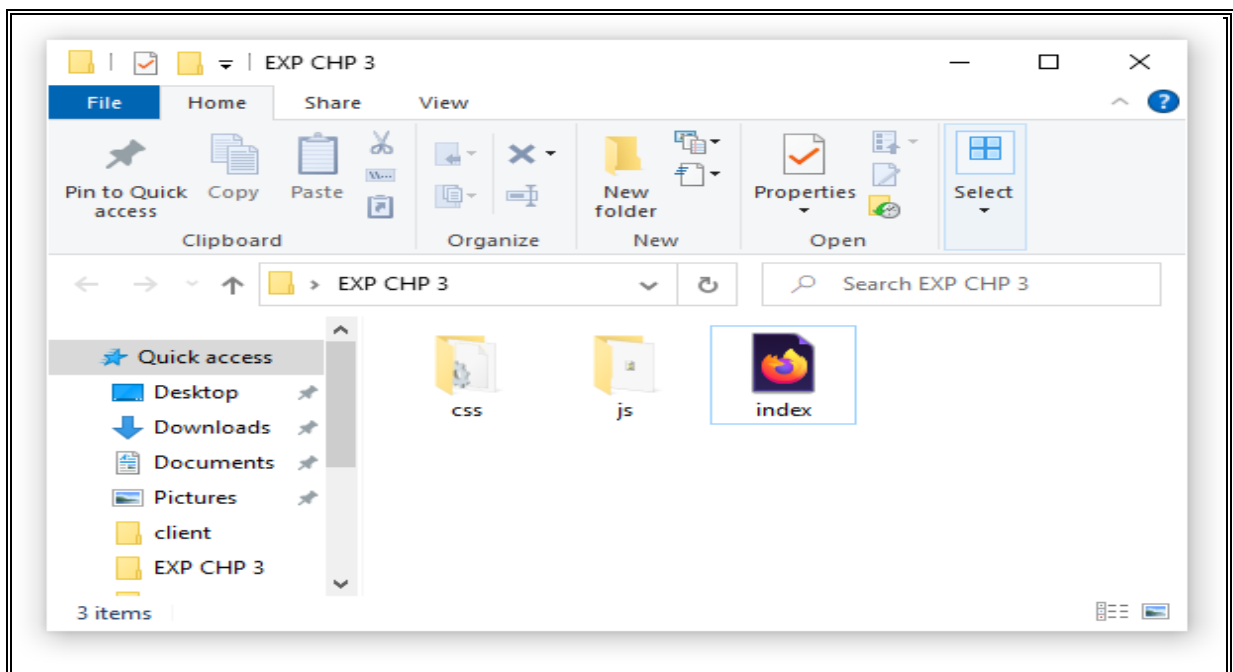


Figure 3-9 Fichier de l'exemple

Cette page web sera simple, nous ne nous attarderons donc pas sur la beauté de l'interface, nous écrivons donc dans le fichier HTML les codes suivants

```

<body>

  <div id="container">

    <section>

      <p>Cette page est un exemple de transfert de fichiers utilisant RTCDataC
hannel API.</p>
      <div style="color:red;font-size: 25px;"

        <p>HIBA ZAHRA  CHAPITRE 3.</p>
      </div>
    </section>

    <section>
      <div >
        <form id="fileInfo">
          <input type="file" id="fileInput" name="files"/>
        </form>
        <button disabled id="sendFile">Send</button>
        <button disabled id="abortButton">Abort</button>
      </div>

      <div class="progress">
        <div class="label">Send progress: </div>
        <progress id="sendProgress" max="0" value="0"></progress>
      </div>

      <div class="progress">
        <div class="label">Receive progress: </div>
        <progress id="receiveProgress" max="0" value="0"></progress>
      </div>

      <div id="bitrate"></div>
      <a id="download"></a>
      <span id="status"></span>

    </section>

  </div>
  <script src="js/main.js"></script>

</body>
</html>

```

L'interface de la page contient un bouton d'envoi et un bouton pour annuler la transmission et nous montre également le pourcentage de téléchargement et le débit binaire moyen .
pour le fichier CSS nous écrivons le code suivant :

```
/*
 * Hiba ZAHRA PROJECT CHAPTER 3.
 */

div.progress, div#bitrate {
  margin: 0 0 1em 0;
  background-color: khaki;
}

div.progress div.label {
  display: inline-block;
  font-weight: 400;
  width: 8.2em;
  background-color: darksalmon;
}

form {
  margin: 0 0 1em 0;
  white-space: nowrap;
}

progress {
  width: calc(100% - 8.5em);
}
```

Étant donné que le code javascript est un peu long, nous afficherons un extrait du code qui montre le API RTCPeerConnection affiché en vert.[18]

```
async function createConnection() {
  abortButton.disabled = false;
  sendFileButton.disabled = true;
  localConnection = new RTCPeerConnection();
  console.log('Created local peer connection object localConnection');

  sendChannel = localConnection.createDataChannel('sendDataChannel');
  sendChannel.binaryType = 'arraybuffer';
  console.log('Created send data channel');

  sendChannel.addEventListener('open', onSendChannelStateChange);
  sendChannel.addEventListener('close', onSendChannelStateChange);
  sendChannel.addEventListener('error', onError);

  localConnection.addEventListener('icecandidate', async event => {
```

```

    console.log('Local ICE candidate: ', event.candidate);
    await remoteConnection.addIceCandidate(event.candidate);
  });

  remoteConnection = new RTCPeerConnection();
  console.log('Created remote peer connection object remoteConnection');

  remoteConnection.addEventListener('icecandidate', async event => {
    console.log('Remote ICE candidate: ', event.candidate);
    await localConnection.addIceCandidate(event.candidate);
  });
  remoteConnection.addEventListener('datachannel', receiveChannelCallback);

  try {
    const offer = await localConnection.createOffer();
    await gotLocalDescription(offer);
  } catch (e) {
    console.log('Failed to create session description: ', e);
  }

  fileInput.disabled = true;
}

function sendData() {
  const file = fileInput.files[0];
  console.log(`File is ${[file.name, file.size, file.type, file.lastModified].
  join(' ')}`);
}

```

Voilà notre page de transfert de fichier avant l'envoi (figure 3-10)

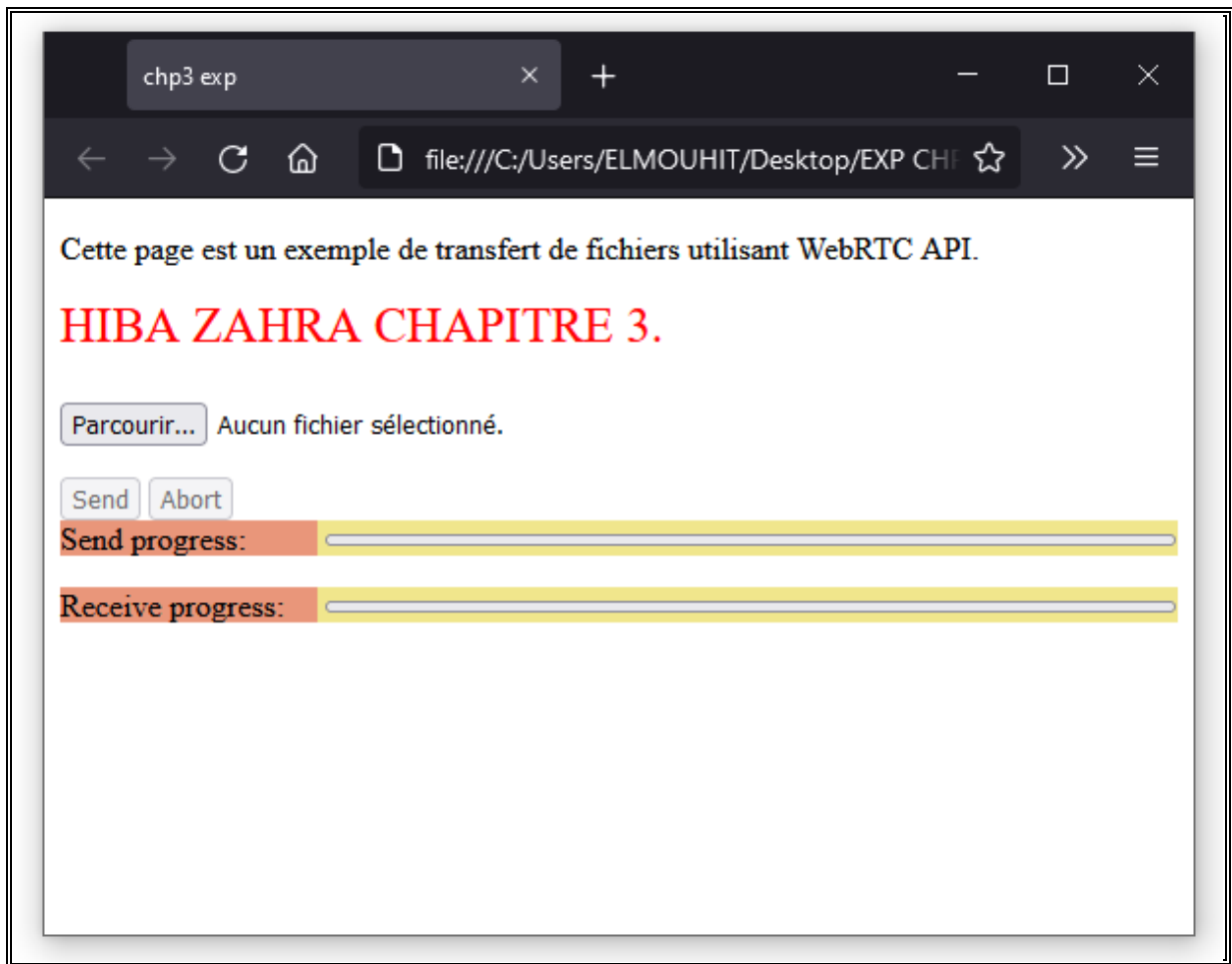


Figure 3-10 la page avant l'envoi

Après avoir envoyé un fichier spécifique et nous choisissons l'image de l'université dans cet exemple, la forme de la page deviendra comme la figure 3-11



Figure 3-11 La page après l'envoi

3 PWA : une description technique

3.1 Exigences techniques de PWA

Les applications mobiles natives ont des fonctionnalités que nos sites internet ne peuvent fournir : installation sur l'écran d'accueil, présence dans les stores, processus externe au navigateur, fonctionnement hors ligne, lecture/écriture dans le système de fichiers, accès au hardware branché ou via bluetooth, manipulation des contacts et événements... Beaucoup de fonctionnalités indispensables semblent faire partie du téléphone. [23]

Pour faire des applications mobiles natives, il faut du code spécifique pour chacune des plateformes. Ce code est donc dupliqué et demande des compétences différentes. Grâce aux nouveautés du web, les Progressive Web Applications (PWA) nous permettent de développer

des expériences de plus en plus proches du natif. Avec le gros avantage de pouvoir partager une bonne partie du code entre les versions desktop, mobiles Android et iOS.[24]

Les PWA ne sont pas créés avec une seule technologie. Ils représentent une nouvelle philosophie pour la création d'applications Web, impliquant des modèles, des API et d'autres fonctionnalités spécifiques. Il ne faut pas grand-chose pour configurer une PWA. Il y a trois choses que vous devez fournir avant que votre site ne se transforme en une PWA valide :

- 1) Un service worker
- 2) Le fichier manifeste
- 3) Une connexion sécurisée (HTTPS)

3.1.1 Le service worker

Un service worker est un script écrit en javascript qui permet d'intercepter et de contrôler les requêtes réseau et la mise en cache des actifs à partir du navigateur Web. Avec les serviceworkers, les développeurs Web peuvent créer des pages Web fiables et rapides et des expériences hors ligne. C'est une fonctionnalité du navigateur mais il n'a pas accès au DOM, son rôle est différent d'un script JS normal. [23]

Comme les serviceWorkers ont accès pour gérer les requêtes réseau, ils peuvent être très puissants, surtout s'ils sont utilisés de manière malveillante par un attaquant, permettant des choses comme l'injection de script ou le remplacement de l'ensemble du site. C'est pourquoi HTTPS est si important ici, afin d'empêcher des tiers de les injecter dans votre site Web.[24]

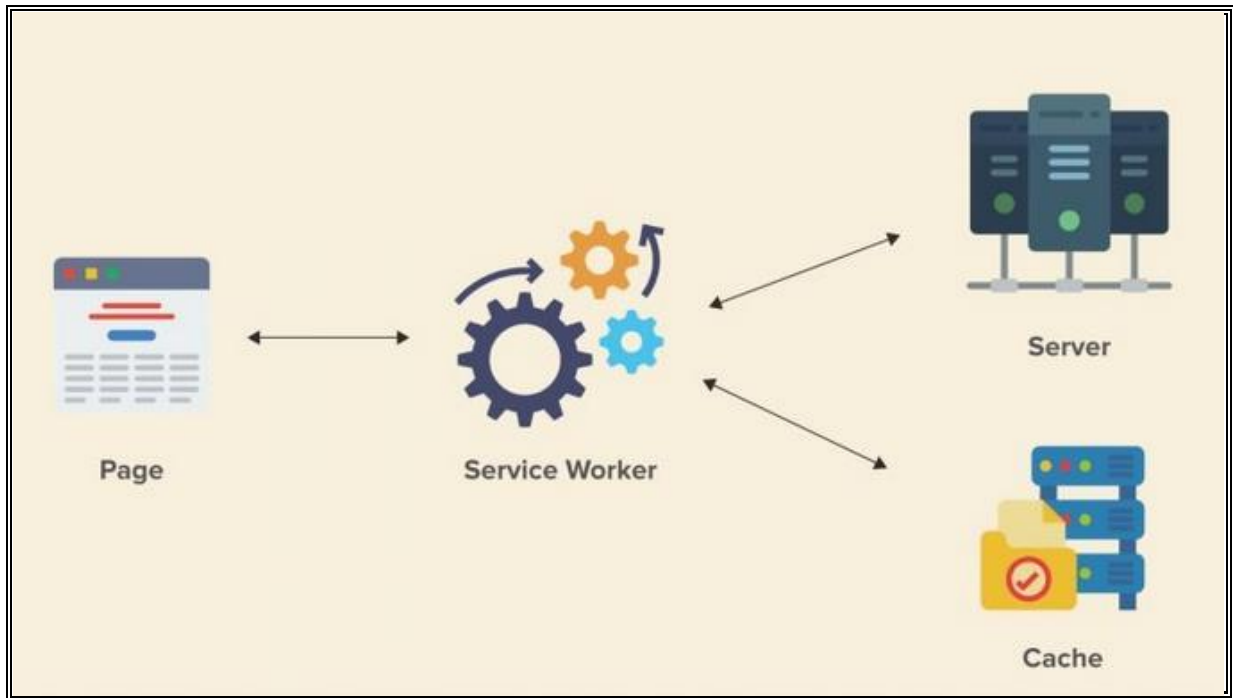


Figure 3-12 Le Service worker

Comment fonctionne un Service Worker ? Eh bien, afin de le rendre aussi simple à comprendre que possible, j'aime la façon dont Jeff Posnick de Google les décrit : « Considérez vos requêtes Web comme des avions qui décollent. Le Service Worker est le contrôleur aérien qui achemine les demandes. Il peut se charger depuis le réseau ou même depuis le cache. En tant que « contrôleurs du trafic aérien ».

Les Service Workers vous donnent un contrôle total sur chaque requête Web effectuée à partir de votre site, ce qui ouvre la possibilité à de nombreux cas d'utilisation différents. De la même manière qu'un contrôleur aérien peut rediriger un avion vers un autre aéroport, voire retarder un atterrissage, un Service Worker vous permet de rediriger vos demandes voire de les arrêter complètement.[23]

Bien que les Service Workers soient écrits en JavaScript, ils sont légèrement différents de votre fichier JavaScript standard. Un technicien de service effectue les tâches suivantes :

- Fonctionne dans son propre contexte de script global
- N'est pas lié à une page Web particulière
- N'est pas en mesure de modifier les éléments de la page Web
- il n'a pas d'accès DOM
- avec HTTPS uniquement

Les Service Workers constituent le point de départ des fonctionnalités qui font fonctionner les applications Web comme des applications natives. Certaines de ces fonctionnalités sont :

API de notifications : un moyen d'afficher et d'interagir avec les notifications à l'aide du système de notification natif du système d'exploitation.

Accès hors ligne : le service worker met en cache le contenu nécessaire pour votre site PWA et le diffusent par la suite. Cela rend efficacement les sites Web alimentés par PWA hors ligne, car les utilisateurs peuvent toujours interagir avec le site et voir tout le contenu mis en cache. [24]

API Push : une API qui permet à votre application de s'abonner à un service push et de recevoir des messages push. Les messages push sont remis à un service worker, qui peut utiliser les informations contenues dans le message pour mettre à jour l'état local ou afficher une notification à l'utilisateur. Étant donné que le service worker s'exécute indépendamment de l'application principale, ils peuvent recevoir et afficher des notifications même lorsque le navigateur n'est pas en cours d'exécution.

API de synchronisation en arrière-plan : vous permet de différer les actions jusqu'à ce que l'utilisateur dispose d'une connectivité stable. Ceci est utile pour s'assurer que tout ce que l'utilisateur veut envoyer est effectivement envoyé. Cette API permet également aux serveurs d'envoyer des mises à jour périodiques à l'application afin que l'application puisse se mettre à jour lors de sa prochaine mise en ligne. [23]

API de messagerie de canal : permet aux web workers et service workers de communiquer entre eux et avec l'application hôte. Les exemples de cette API incluent la notification de nouveau contenu et les mises à jour qui nécessitent une interaction de l'utilisateur.

Les codes javascript suivants représentent le Service worker de notre projet

```
var CACHE_NAME = 'HIBA ZAHRA-cache-v2';
var urlsToCache = [
  'index.html',
  './',
  'styles.css',
  'scripts/network.js',
  'scripts/ui.js',
  'scripts/clipboard.js',
  'scripts/theme.js',
  'sounds/blorp.mp3',
  'images/favicon-96x96.png'
];
```

```

self.addEventListener('install', function(event) {
  // Perform install steps
  event.waitUntil(
    caches.open(CACHE_NAME)
      .then(function(cache) {
        console.log('Opened cache');
        return cache.addAll(urlsToCache);
      })
  );
});

self.addEventListener('fetch', function(event) {
  event.respondWith(
    caches.match(event.request)
      .then(function(response) {
        // Cache hit - return response
        if (response) {
          return response;
        }
        return fetch(event.request);
      })
  );
});

self.addEventListener('activate', function(event) {
  console.log('Updating Service Worker...')
  event.waitUntil(
    caches.keys().then(function(cacheNames) {
      return Promise.all(
        cacheNames.filter(function(cacheName) {
          // Return true if you want to remove this cache,
          // but remember that caches are shared across
          // the whole origin
          return true
        }).map(function(cacheName) {
          return caches.delete(cacheName);
        })
      );
    });
  );
});

```

3.1.2 Le fichier manifeste

Le manifeste de l'application Web est un fichier JSON qui informe le navigateur de votre application Web progressive et de la façon dont elle doit se comporter lorsqu'elle est installée sur le bureau ou l'appareil mobile de l'utilisateur. Un fichier manifeste typique comprend le nom de l'application, les icônes que l'application doit utiliser et l'URL qui doit être ouverte lorsque l'application est lancée.

Les fichiers manifestes sont pris en charge dans Chrome, Edge, Firefox, le navigateur UC, Opera et le navigateur Samsung. Safari a un support partiel. [23]

Le fichier manifeste peut avoir n'importe quel nom, mais est généralement nommé `manifest.json`. La spécification suggère que l'extension devrait être `.webmanifest`, mais les navigateurs prennent également en charge les extensions `.json`, ce qui peut être plus facile à comprendre pour les développeurs. [24]

Liste pour certaines propriétés et fonctionnalités

name Le nom de la PWA retrouvable au sein de l'environnement ayant installé la PWA

short_name Le nom utilisé par l'environnement juste en dessous de l'icône de la PWA

start_url Il faut renseigner l'URL de démarrage de la PWA.

display Permet de spécifier comment la PWA va être présentée. `fullscreen` permet d'ouvrir l'application en plein écran, `stand-alone` de garder la statusbar et le bouton back, `minimal-ui` avec aucun bouton et browser pour une vue par défaut (navigateur classique). Cette fonctionnalité permet aussi de filtrer certains périphériques qui ne sauraient pas gérer le mode d'affichage renseigné. `fullscreen` et `minimal-ui` ne sont pas encore supportés par Safari.

orientation Permet de préciser le type d'affichage pour la PWA à savoir portrait, paysage ou les deux.

background_color Force la couleur du background avant le chargement des CSS. On voit le `background_color` généralement au lancement de la PWA sur une très courte durée.

scope Par défaut c'est la `start_url`, il est possible de renseigner par exemple un scope spécifique pour chaque type de Service Workers.

icons Peut contenir une collection d'icônes avec différentes tailles .

Et voilà le code de fichier manifeste de notre projet

```

"name": "HIBA ZAHRA",
  "short_name": "HIBA ZAHRA",
  "icons": [{
    "src": "images/blida.png",
    "sizes": "192x192",
    "type": "image/png"
  },{
    "src": "images/blida.png",
    "sizes": "512x512",
    "type": "image/png"
  },{
    "src": "images/blida.png",
    "sizes": "192x192",
    "type": "image/png",
    "purpose": "maskable"
  },{
    "src": "images/blida.png",
    "sizes": "512x512",
    "type": "image/png",
    "purpose": "maskable"
  },{
    "src": "images/blida.png",
    "sizes": "96x96",
    "type": "image/png"
  }],
  "background_color": "#d2ffcf",
  "start_url": "/",
  "display": "minimal-ui",
  "theme_color": "#fc170f",
  "share_target": {
    "method": "GET",
    "action": "/?share_target",
    "params": {
      "title": "title",
      "text": "text",
      "url": "url"
    }
  }
}
}

```

3.1.3 Une connexion sécurisée (HTTPS)

Les PWA ne fonctionnent que sur des connexions de confiance, vous devez les servir via une connexion sécurisée. Ce n'est pas seulement pour des raisons de sécurité, mais c'est aussi un facteur de confiance très important pour les utilisateurs.

Les nouvelles fonctionnalités puissantes de la plate-forme Web, telles que la prise de photos ou l'enregistrement audio avec `getUserMedia()`, l'activation d'expériences d'application hors ligne avec le service worker ou la création d'applications Web progressives, nécessitent une autorisation explicite de l'utilisateur avant de s'exécuter. De nombreuses API plus anciennes sont également mises à jour pour nécessiter une autorisation d'exécution, comme l'API de géolocalisation. [23]

Tous les sites Web doivent être protégés par HTTPS, même ceux qui ne traitent pas de données sensibles. HTTPS empêche les intrus de falsifier ou d'écouter passivement les communications entre votre site et vos utilisateurs.

Une page ne peut pas être considérée comme une application Web progressive (PWA) si elle ne s'exécute pas sur HTTPS ; de nombreuses technologies PWA de base, telles que le service worker, nécessitent HTTPS. [23]

Les certificats Secure Socket Layer (SSL) représentent les fondements de la confiance dans la plupart des transactions Web et Internet. La confiance est le maître mot en matière de SSL et HTTPS. Lorsqu'un site Web utilise SSL, la communication entre le navigateur et le serveur est cryptée, mais pour obtenir un certificat SSL, vous devez établir un niveau de confiance avec une autorité émettrice. [23]

Pour activer SSL, vous devez installer un certificat sur votre serveur. Les certificats sont émis par une autorité de certification (CA). Aujourd'hui, il existe de nombreuses autorités de certification, vous devez rechercher le meilleur fournisseur pour vos besoins. Dans un passé pas si lointain, Network Solutions était la seule autorité disponible pour acheter un certificat. , vous avez dû faire face à beaucoup de paperasserie. S'ils n'aimaient pas vos papiers, ils vous rejetteraient. Il était presque impossible pour les particuliers d'acheter un certificat, car la propriété du domaine devait être liée à une entreprise enregistrée. Cette disponibilité limitée a conduit à des prix élevés pour les certificats annuels. Le blog, l'entreprise ou l'organisation moyenne n'a jamais envisagé d'utiliser SSL en raison de son coût. Cela limitait SSL aux sites qui transféraient des informations sensibles, telles que des numéros de carte de crédit et de compte bancaire, en raison des barrières d'origine.

L'histoire du HTTPS a changé depuis lors. Il existe de nombreuses autorités de certification gratuites et à faible coût, éliminant ainsi la barrière des coûts annuels. Le protocole HTTP et

la technologie des serveurs Web ont également progressé. Aujourd'hui, vous pouvez héberger plusieurs sites sur la même adresse IP en utilisant différents certificats et en-têtes d'hôte (domaines). [23]

La première fonctionnalité, et souvent la seule, à laquelle vous penserez lorsque quelqu'un posera des questions sur SSL ou HTTPS est le cryptage. C'est une bonne raison d'utiliser HTTPS, mais ce n'est pas la seule raison, ni même la raison la plus importante. HTTPS nous donne trois propriétés de sécurité :

Identité : le certificat prouve que le serveur est le vrai serveur

Confidentialité : Seuls le navigateur et le serveur peuvent lire les données échangées entre eux.

Intégrité : les données envoyées sont ce que l'autre partie reçoit.

3.2 PWA : Outils de programmation

3.2.1 HTML : Description

HTML est un acronyme qui signifie Hyper TextMarkupLanguage qui est utilisé pour créer des pages Web et des applications Web. Voyons ce que l'on entend par HypertextMarkupLanguage et page Web.[25]

Hypertexte :HyperTexte signifie simplement « Texte dans le texte ». Un texte a un lien en lui, c'est un hypertexte. Chaque fois que vous cliquez sur un lien qui vous amène à une nouvelle page Web, vous avez cliqué sur un hypertexte. L'hypertexte est un moyen de lier deux ou plusieurs pages Web (documents HTML) entre elles. [25]

Langage de balisage : Un langage de balisage est un langage informatique utilisé pour appliquer des conventions de mise en page et de formatage à un document texte. Le langage de balisage rend le texte plus interactif et dynamique. Il peut transformer du texte en images, tableaux, liens, etc.

Page Web : Une page Web est un document généralement écrit en HTML et traduit par un navigateur Web. Une page Web peut être identifiée en entrant une URL. Une page Web peut être de type statique ou dynamique. Avec l'aide de HTML uniquement, nous pouvons créer des pages Web statiques.

Par conséquent, HTML est un langage de balisage qui est utilisé pour créer des pages Web attrayantes à l'aide de styles et qui se présente dans un format agréable sur un navigateur Web.

Un document HTML est composé de nombreuses balises HTML et chaque balise HTML contient un contenu différent. [25]

Voyons un exemple simple de HTML.

```
<!DOCTYPE html>
<html>
<head>
<title>Titre de la page Web</title>
</head>
<body>
<h1>Écrivez votre premier titre</h1>
<p>Écrivez votre premier paragraphe .</p>
</body>
</html>
```

3.2.2 CSS : Description

CSS signifie feuilles de style en cascade. C'est un langage de feuille de style qui est utilisé pour décrire l'apparence et la mise en forme d'un document écrit en langage de balisage. Il fournit une fonctionnalité supplémentaire au HTML. Il est généralement utilisé avec HTML pour modifier le style des pages Web et des interfaces utilisateur. Il peut également être utilisé avec n'importe quel type de documents XML, y compris XML brut, SVG et XUL.

CSS est utilisé avec HTML et JavaScript dans la plupart des sites Web pour créer des interfaces utilisateur pour les applications Web et des interfaces utilisateur pour de nombreuses applications mobiles. [25]

Avant CSS, les balises telles que la police, la couleur, le style d'arrière-plan, l'alignement des éléments, la bordure et la taille devaient être répétées sur chaque page Web. Ce fut un très long processus. Par exemple : si vous développez un grand site Web où des polices et des informations sur les couleurs sont ajoutées sur chaque page, cela deviendra un processus long et coûteux. CSS a été créé pour résoudre ce problème.



Figure 3-13 Logo de HTML5 et CSS3

3.2.3 Javascript : Description

JavaScript (js) est un langage de programmation orienté objet léger qui est utilisé par plusieurs sites Web pour scripter les pages Web. Il s'agit d'un langage de programmation interprété à part entière qui permet une interactivité dynamique sur les sites Web lorsqu'il est appliqué à un document HTML. Il a été introduit en 1995 pour ajouter des programmes aux pages Web dans le navigateur Netscape Navigator. Depuis lors, il a été adopté par tous les autres navigateurs Web graphiques. Avec JavaScript, les utilisateurs peuvent créer des applications Web modernes pour interagir directement sans recharger la page à chaque fois. Le site Web traditionnel utilise js pour fournir plusieurs formes d'interactivité et de simplicité. Il existe les fonctionnalités suivantes de JavaScript : [25]

- 1- Tous les navigateurs Web populaires prennent en charge JavaScript car ils fournissent des environnements d'exécution intégrés.
- 2- JavaScript suit la syntaxe et la structure du langage de programmation C. C'est donc un langage de programmation structuré.
- 3- JavaScript est un langage faiblement typé, où certains types sont implicitement transtypés (selon l'opération).
- 4- JavaScript est un langage de programmation orienté objet qui utilise des prototypes plutôt que d'utiliser des classes pour l'héritage.
- 5- C'est une langue légère et interprétée.
- 6- C'est un langage sensible à la casse.
- 7- JavaScript est compatible avec plusieurs systèmes d'exploitation, notamment Windows, macOS, etc.
- 8- Il offre un bon contrôle aux utilisateurs sur les navigateurs Web.



Figure 3-14 Logo de Javascript

3.2.4 Node.js : Description

Node.js est un environnement d'exécution et une bibliothèque multiplateformes permettant d'exécuter des applications JavaScript en dehors du navigateur. Il est utilisé pour créer des applications Web côté serveur et en réseau. Il est open source et gratuit à utiliser. Il peut être téléchargé à partir de ce lien <https://nodejs.org/en/>

De nombreux modules de base de Node.js sont écrits en JavaScript. Node.js est principalement utilisé pour exécuter des applications serveur en temps réel.[26]

Voici une liste de certaines fonctionnalités importantes de Node.js qui en font le premier choix des architectes logiciels.[26]

- 1- Extrêmement rapide : Node.js est construit sur le moteur JavaScript V8 de Google Chrome, sa bibliothèque est donc très rapide dans l'exécution du code.
- 2- Les E/S sont asynchrones et pilotées par les événements : toutes les API de la bibliothèque Node.js sont asynchrones, c'est-à-dire non bloquantes. Ainsi, un serveur basé sur Node.js n'attend jamais qu'une API renvoie des données. Le serveur passe à l'API suivante après l'avoir appelée et un mécanisme de notification des événements de Node.js aide le serveur à obtenir une réponse de l'appel API précédent. C'est aussi une raison pour laquelle il est très rapide.[26]
- 3- À thread unique : Node.js suit un modèle à thread unique avec une boucle d'événement.
- 4- Hautement évolutif : Node.js est hautement évolutif car le mécanisme d'événement aide le serveur à répondre de manière non bloquante.

- 5- Pas de mise en mémoire tampon : Node.js réduit le temps de traitement global lors du téléchargement de fichiers audio et vidéo. Les applications Node.js ne mettent jamais en mémoire tampon aucune donnée. Ces applications génèrent simplement des données en morceaux.[26]
- 6- Open source : Node.js possède une communauté open source qui a produit de nombreux excellents modules pour ajouter des fonctionnalités supplémentaires aux applications Node.js.
- 7- Licence : Node.js est publié sous la licence MIT.

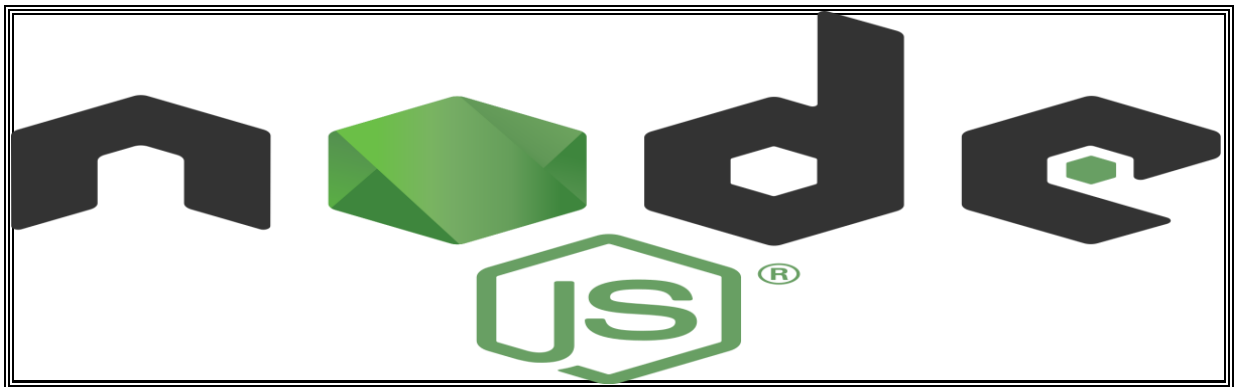


Figure 3-15 Logo de Node.js

3.3 PWA : Déploiement

3.3.1 Les types d'hébergement web

Il existe plusieurs façons d'héberger notre travail, je vais vous en expliquer les 2 plus répandus.

3.3.1.1 Les hébergements mutualisés :

Le plus répandu est bien entendu les hébergements dits mutualisés. C'est-à-dire que vous partagez avec d'autres applications les mêmes ressources sur le serveur. Ces hébergements sont en général du PHP/MySQL car même si ce dernier est un peu en perte de vitesse, les sites Internet sont en PHP et pour 20% d'entre eux sous WordPress.

C'est d'ailleurs le type d'hébergement le plus économique et le plus simple à utiliser quand on a une application web ou un site Internet en PHP/MySQL. Malheureusement, le déploiement est assez archaïque et limité étant donné que vous n'avez pas la main sur le serveur

En général, vous avez un accès à votre base de données via un PHPMyAdmin + un accès FTP à votre hébergement. Votre déploiement est donc manuel : vous utilisez un client FTP pour uploader vos fichiers qui composent votre site Internet ou votre application web et vous utilisez le PHPMyAdmin pour monter et manager votre base de données.[27]

3.3.1.2 Les serveurs dédiés :

L'autre possibilité est d'avoir un serveur dédié et c'est déjà nettement mieux pour une application web.

Il y a 2 possibilités dans les serveurs dédiés : managé ou non managé. Dans les 2 cas, vous aurez des ressources serveur complètement dédié et vous aurez le contrôle total de votre serveur en mode non managé. Cela implique de savoir ce que l'on fait et de configurer totalement votre serveur.[27]

Vous pouvez par exemple très bien effectuer un déploiement de votre application avec installation depuis votre dépôt GitHub si vous créez un nouveau tag de version sur votre branche master par exemple.

Même si le déploiement peut être automatisé, on peut y trouver 3 inconvénients quand on choisit un serveur :

- ça prend du temps à configurer la première fois
- ça prend du temps à maintenir l'architecture du serveur que vous avez mis en place
- vous devez avoir des connaissances techniques pour la configuration d'un serveur.

Mais est-ce que vous vous sentez le courage d'assumer l'installation, le paramétrage et la maintenance de vos serveurs ? Avez-vous les connaissances techniques ?

Car après tout vous êtes développeur, pas un Administrateur Réseau et Système. Les PaaS sont peut-être la solution qui vont balayer toutes les questions précédentes. Mais avant tout, il faut savoir ce qu'est un PaaS.[27]

C'est l'abréviation de « Platform as a Service » ou encore en français « Plate-forme en tant que service ».

On ne parle plus d'hébergement ou de serveur, on parle de payer un service pour utiliser des logiciels et des outils en ligne bien souvent à destination des développeurs, Les PaaS vous permettent de faire abstraction de toute la partie serveur et logiciel qui le compose.

C'est le fournisseur du PaaS qui prend tout cela à sa charge. Vous avez juste besoin de lui communiquer ce dont vous avez besoin pour faire tourner votre application web et il s'occupe de faire le job pour vous.

Parmi beaucoup de fournisseurs qui font du PaaS aujourd'hui comme OVH ou Infomaniak , pour notre PWA on va utiliser Heroku

3.3.2 Heroku : Description et mode d'emploi

Heroku plateforme en tant que Service (Platform as a Service - PaaS) est une entreprise de Salesforce depuis 2010; l'une des entreprises pionnières du cloudcomputing.



Figure 3-16 Logo de Heroku

Heroku est une plate-forme largement utilisée en tant qu'offre de services qui permet aux développeurs d'effectuer le déploiement, la mise à l'échelle et la gestion d'applications sans tracas. Cette plate-forme prend en charge un large éventail de langages de programmation tels que Java, Ruby, PHP, Node.js, Python, Scala et Clojure. Heroku exécute les applications via des conteneurs virtuels appelés Dynos. [28]

Heroku facture ses utilisateurs en fonction du nombre de machines virtuelles nécessaires à leurs applications. La plate-forme Heroku et les applications créées par les utilisateurs utilisent Amazon Web Services comme infrastructure sous-jacente. Les développeurs peuvent réaliser un développement d'applications rapide en l'utilisant car c'est très pratique. [28]

Heroku a quatre plans de facturation: [28]

Free & Hobby: pour les projets personnels.

Production: pour les applications en production

Advanced: pour des applications un peu plus complexes

Entreprise: pour les entreprises. Nécessite de contacter le service commercial de Heroku

Type de plan	Idéal pour	La description
--------------	------------	----------------

Free and Hobby	Produits minimum viables Applications non commerciales Projets en solo	Heroku propose un forfait gratuit et un forfait Hobby, tous deux à partir de 0 \$. Ce sont les plans de démarrage, les plus adaptés aux déploiements d'applications non commerciales.
Production	Applications pour petites entreprises	Le plan de production Heroku commence à partir d'un taux de 25 \$ par mois, mais le taux peut varier en fonction des besoins des clients.
Advanced	Applications critiques	Le plan avancé Heroku commence à partir de 250 \$ par mois. Il convient particulièrement aux entreprises ayant des exigences de développement plus sophistiquées.
Enterprise	Applications de qualité entreprise	Le forfait Heroku Enterprise est un forfait flexible qui oblige les clients à payer en fonction de leurs besoins. Il convient particulièrement aux entreprises ayant des exigences de sécurité et de conformité plus sophistiquées.

Tableau 3-1 Les quatre plans de Heroku

Pour déployer notre PWA il faut disposer des éléments suivants:

- un compte Heroku: Rendez-vous sur le site de Heroku pour la création du compte
- un ordinateur avec git d'installé et un accès à internet
- un projet Node.js

A la racine de notre application, on a créé un fichier *Procfile* (attention de bien respecter l'orthographe), contenant la ligne suivante :

web: node index.js

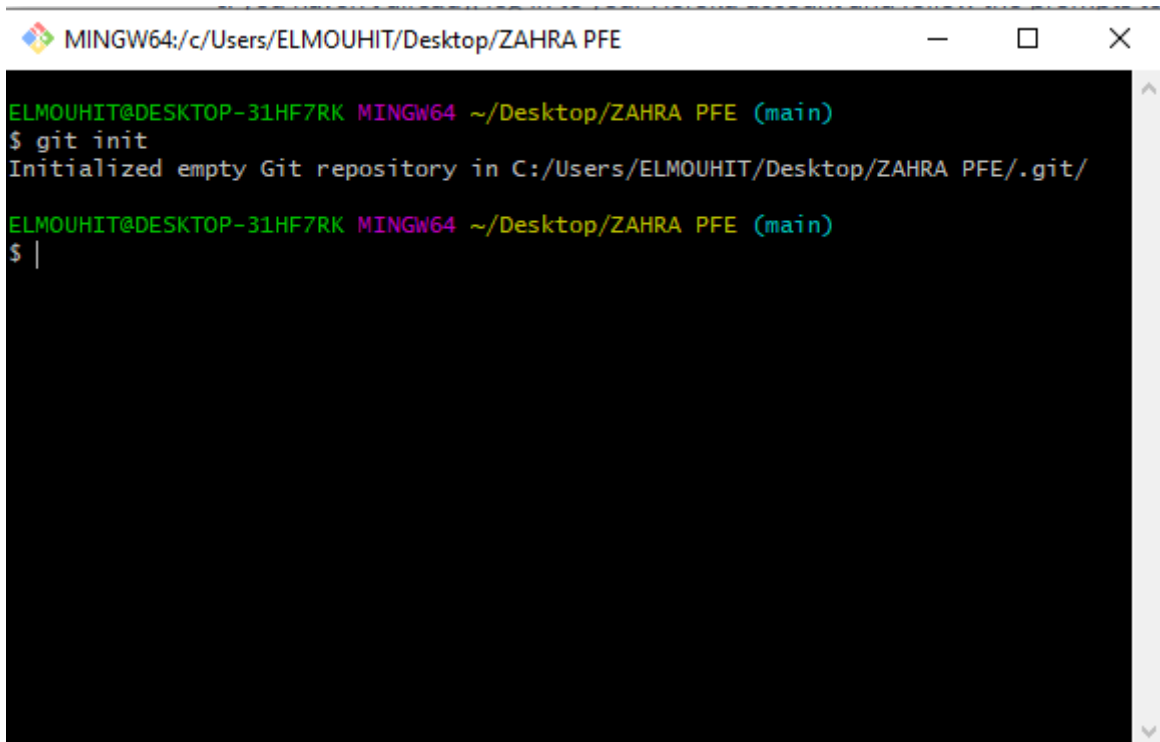
Cette ligne permet au serveur *Heroku* de déterminer la commande de lancement de notre application.

Après avoir choisi le nom de Notre PWA "zahrapfe" au plateforme Heroku, on ouvre le dossier avec git



```
MINGW64:/c/Users/ELMOUHIT/Desktop/ZAHRA PFE
ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$ |
```

Après On initialise le dossier de projet dans git.



```
MINGW64:/c/Users/ELMOUHIT/Desktop/ZAHRA PFE
ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$ git init
Initialized empty Git repository in C:/Users/ELMOUHIT/Desktop/ZAHRA PFE/.git/
ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$ |
```

```
MINGW64:/c/Users/ELMOUHIT/Desktop/ZAHRA PFE
ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$ git init
Initialized empty Git repository in C:/Users/ELMOUHIT/Desktop/ZAHRA PFE/.git/

ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$ heroku git:remote -a zahrapfe
set git remote heroku to https://git.heroku.com/zahrapfe.git

ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$ |
```

On envoie maintenant notre code sur le serveur *Heroku*

```
MINGW64:/c/Users/ELMOUHIT/Desktop/ZAHRA PFE
set git remote heroku to https://git.heroku.com/zahrapfe.git

ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$ git add .
starting fsmonitor-daemon in 'C:/Users/ELMOUHIT/Desktop/ZAHRA PFE'
warning: LF will be replaced by CRLF in .gitattributes.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in index.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in package.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in public/index.html.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in public/manifest.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in public/scripts/clipboard.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in public/scripts/network.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in public/scripts/theme.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in public/scripts/ui.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in public/service-worker.js.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in public/styles.css.
The file will have its original line endings in your working directory

ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$ |
```

```
ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$ git commit -am "make it better"
[main (root-commit) 4c9b951] make it better
17 files changed, 2705 insertions(+)
 create mode 100644 .gitattributes
 create mode 100644 .gitignore
 create mode 100644 README.md
 create mode 100644 index.js
 create mode 100644 package.json
 create mode 100644 procfile
 create mode 100644 public/images/favicon-96x96.png
 create mode 100644 public/index.html
 create mode 100644 public/manifest.json
 create mode 100644 public/scripts/clipboard.js
 create mode 100644 public/scripts/network.js
 create mode 100644 public/scripts/theme.js
 create mode 100644 public/scripts/ui.js
 create mode 100644 public/service-worker.js
 create mode 100644 public/sounds/blip.mp3
 create mode 100644 public/sounds/blip.ogg
 create mode 100644 public/styles.css

ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$ |
```

```
ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$ git push heroku main
Enumerating objects: 23, done.
Counting objects: 100% (23/23), done.
Delta compression using up to 4 threads
Compressing objects: 100% (20/20), done.
Writing objects: 100% (23/23), 126.45 KiB | 7.90 MiB/s, done.
Total 23 (delta 0), reused 0 (delta 0), pack-reused 0
remote: Compressing source files... done.
remote: Building source:
remote:
remote: -----> Building on the Heroku-20 stack
remote: -----> Determining which buildpack to use for this app
remote: -----> Node.js app detected
remote:
remote: -----> Creating runtime environment
remote:
remote:       NPM_CONFIG_LOGLEVEL=error
remote:       NODE_VERBOSE=false
remote:       NODE_ENV=production
remote:       NODE_MODULES_CACHE=true
remote:
remote: -----> Installing binaries
remote:   engines.node (package.json):  unspecified
remote:   engines.npm (package.json):   unspecified (use default)
remote:
remote:   Resolving node version 14.x...
remote:   Downloading and installing node 14.17.6...
remote:   Using default npm version: 6.14.15
remote:
remote: -----> Installing dependencies
remote:   Installing node modules (package.json)
remote:   added 53 packages from 162 contributors and audited 53 packages i
n 2.367s
remote:
remote:   1 package is looking for funding
remote:     run 'npm fund' for details
remote:
remote:   found 0 vulnerabilities
remote:
remote: -----> Build
remote:
remote: -----> Caching build
```

```
MINGW64:/c/Users/ELMOUHIT/Desktop/ZAHRA PFE
remote:   Downloading and installing node 14.17.6...
remote:   Using default npm version: 6.14.15
remote:
remote: -----> Installing dependencies
remote:   Installing node modules (package.json)
remote:   added 53 packages from 162 contributors and audited 53 packages in 2.367s
remote:
remote:   1 package is looking for funding
remote:     run 'npm fund' for details
remote:
remote:   found 0 vulnerabilities
remote:
remote: -----> Build
remote:
remote: -----> Caching build
remote:   - node_modules
remote:
remote: -----> Pruning devDependencies
remote:   audited 53 packages in 0.759s
remote:
remote:   1 package is looking for funding
remote:     run 'npm fund' for details
remote:
remote:   found 0 vulnerabilities
remote:
remote: -----> Build succeeded!
remote: -----> Discovering process types
remote:
remote: ~   Mis-cased procfile detected; ignoring.
remote: ~   Rename it to Procfile to have it honored.
remote:
remote:   Procfile declares types    -> (none)
remote:   Default types for buildpack -> web
remote:
remote: -----> Compressing...
remote:   Done: 33.5M
remote: -----> Launching...
remote:   Released v3
remote:   https://zahrapfe.herokuapp.com/ deployed to Heroku
remote:
remote: Verifying deploy... done.
remote: To https://git.heroku.com/zahrapfe.git
remote: * [new branch]    main -> main
ELMOUHIT@DESKTOP-31HF7RK MINGW64 ~/Desktop/ZAHRA PFE (main)
$
```

URL

Voilà notre application est déployée avec succès.

4 Conclusion

Dans ce chapitre, on a couvert notre projet à partir de quatre aspects de base, dont le premier et le plus important est la technologie WebRTC, dont nous pouvons dire qu'elle est le cœur et le moteur de notre application, et pour cela on lui a donné une grande partie pour couvrir ses aspects les plus importants.

Deuxièmement, on a parlé d'exigences techniques de PWA avec lesquelles nous avons travaillé dans notre projet.

Troisièmement, on a mentionné les outils de programmation utilisés.

Quatrièmement et enfin, on a expliqué la méthode de déploiement de l'application sur Internet en utilisant la plate-forme Heroku.

Ce chapitre était le plan de travail de notre projet dont nous avons utilisé des outils gratuits pour développer notre projet dans les quatre parties mentionnées ci-dessus.

Conclusion générale

Le transfert de fichiers est le processus de copie ou de déplacement d'un fichier d'un ordinateur à un autre via un réseau ou une connexion Internet. Il permet le partage, le transfert ou la transmission d'un fichier ou d'un objet de données logique entre différents utilisateurs et/ou ordinateurs à la fois localement et à distance.

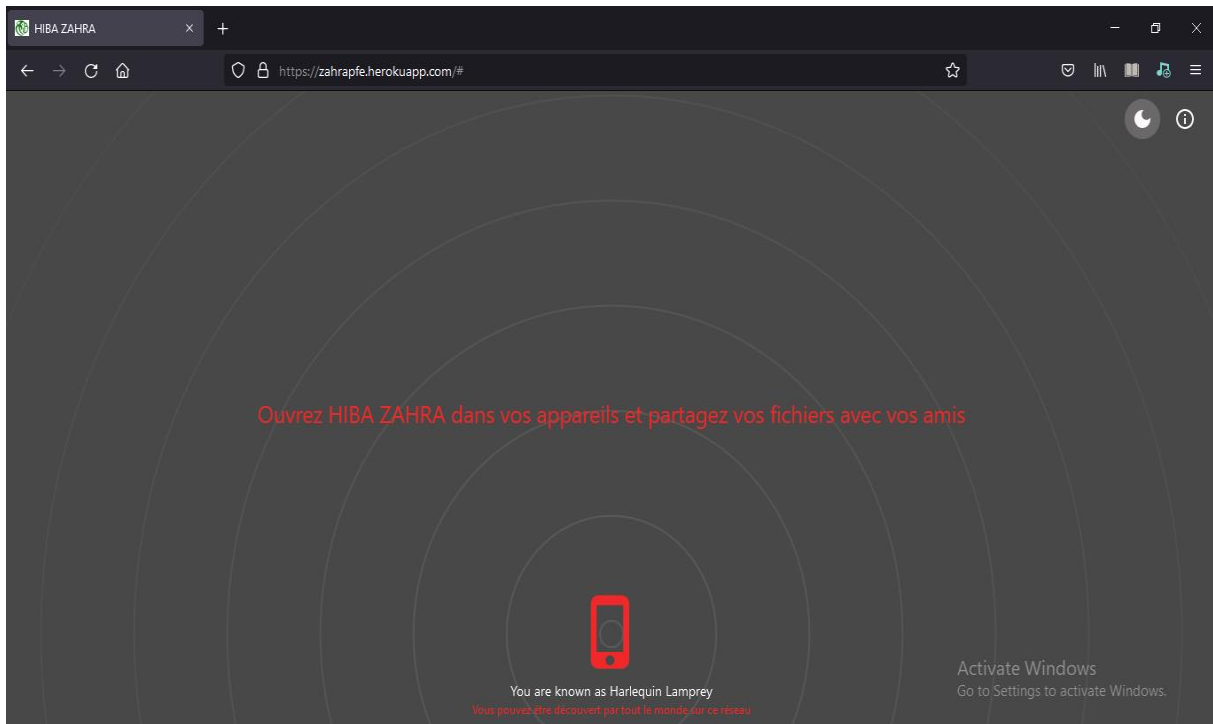
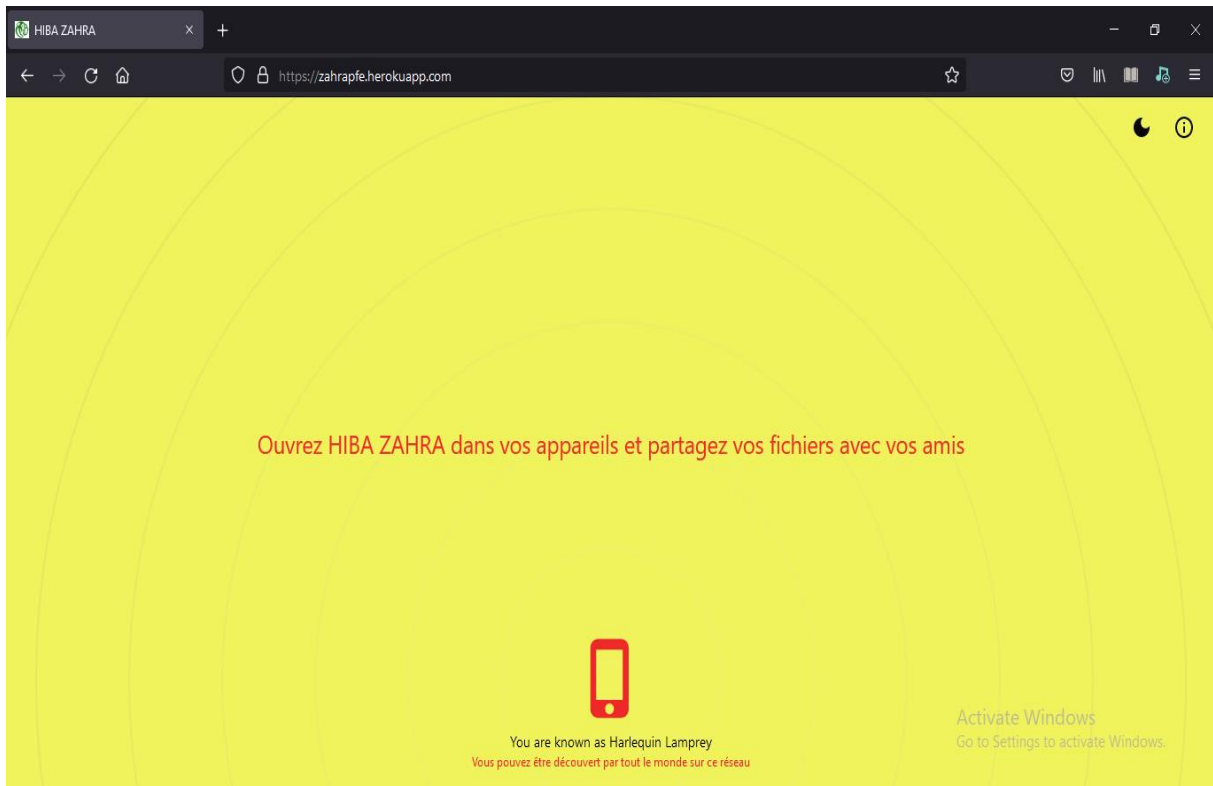
Le mobile occupe de plus en plus l'espace technologique et les applications Web progressives remplissent une fonction particulière. En rationalisant la façon dont nous utilisons les applications mobiles, les applications Web progressives deviendront certainement un élément constitutif de l'interface mobile tant qu'elles représenteront une part importante du commerce électronique et de l'activité des utilisateurs, C'est pourquoi nous avons développé notre PWA transfert des fichiers à l'aide de la technologie WebRTC.

De nombreux problèmes peuvent se présenter dans le secteur de la télécommunication.

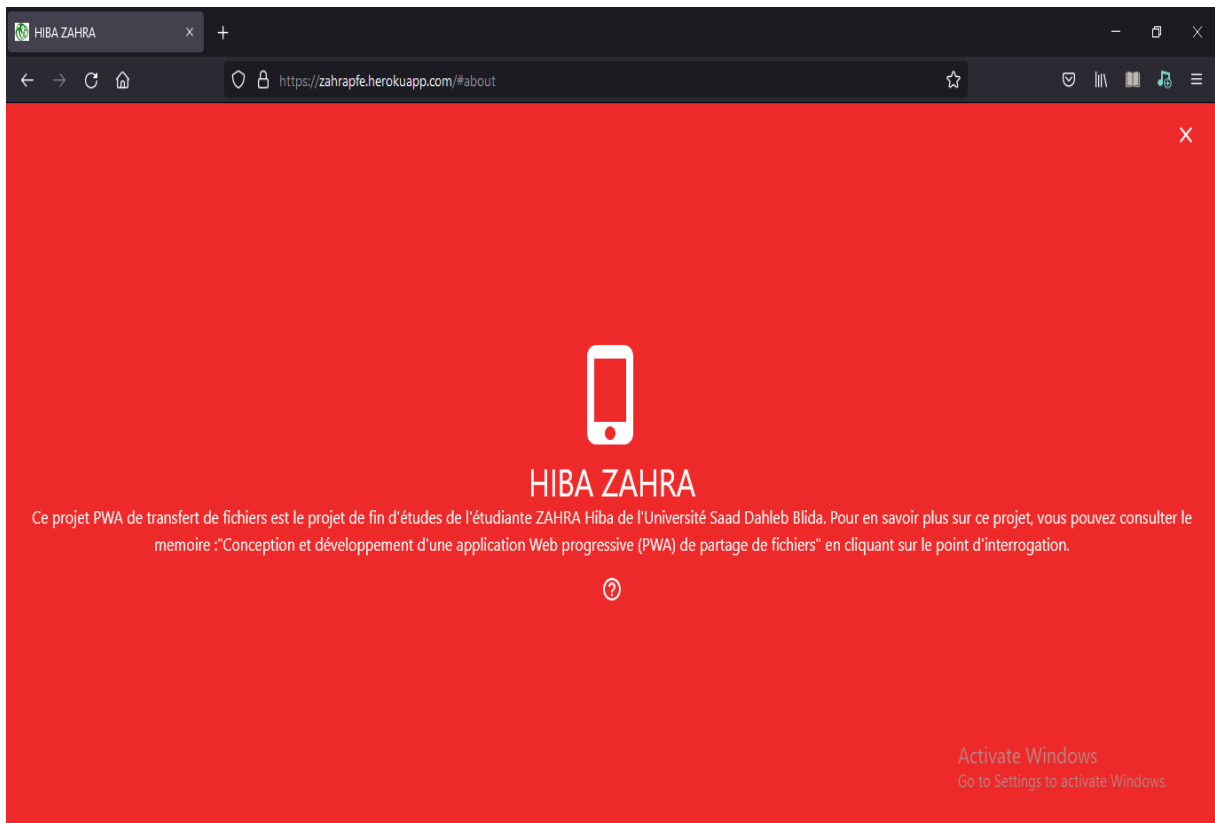
S'agissant entre autres des coûts d'appel très élevés, des problèmes techniques ou des lignes occupées, Le WebRTC offre une solution pour remédier à ces problèmes de communication. dans notre cas c'est La simplicité du partage des fichiers, Notez bien que la plupart des sites de renom choisissent le WebRTC pour échanger des données avec les internautes.

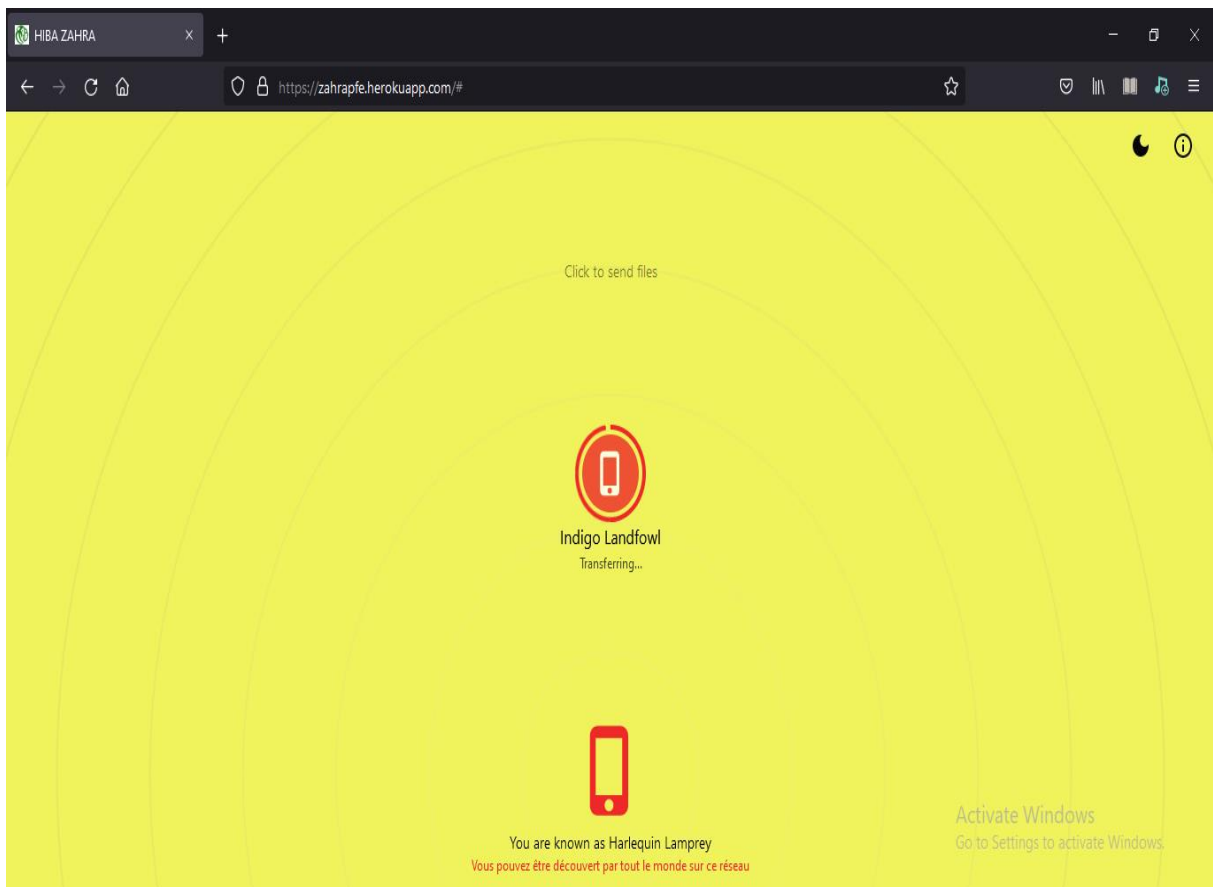
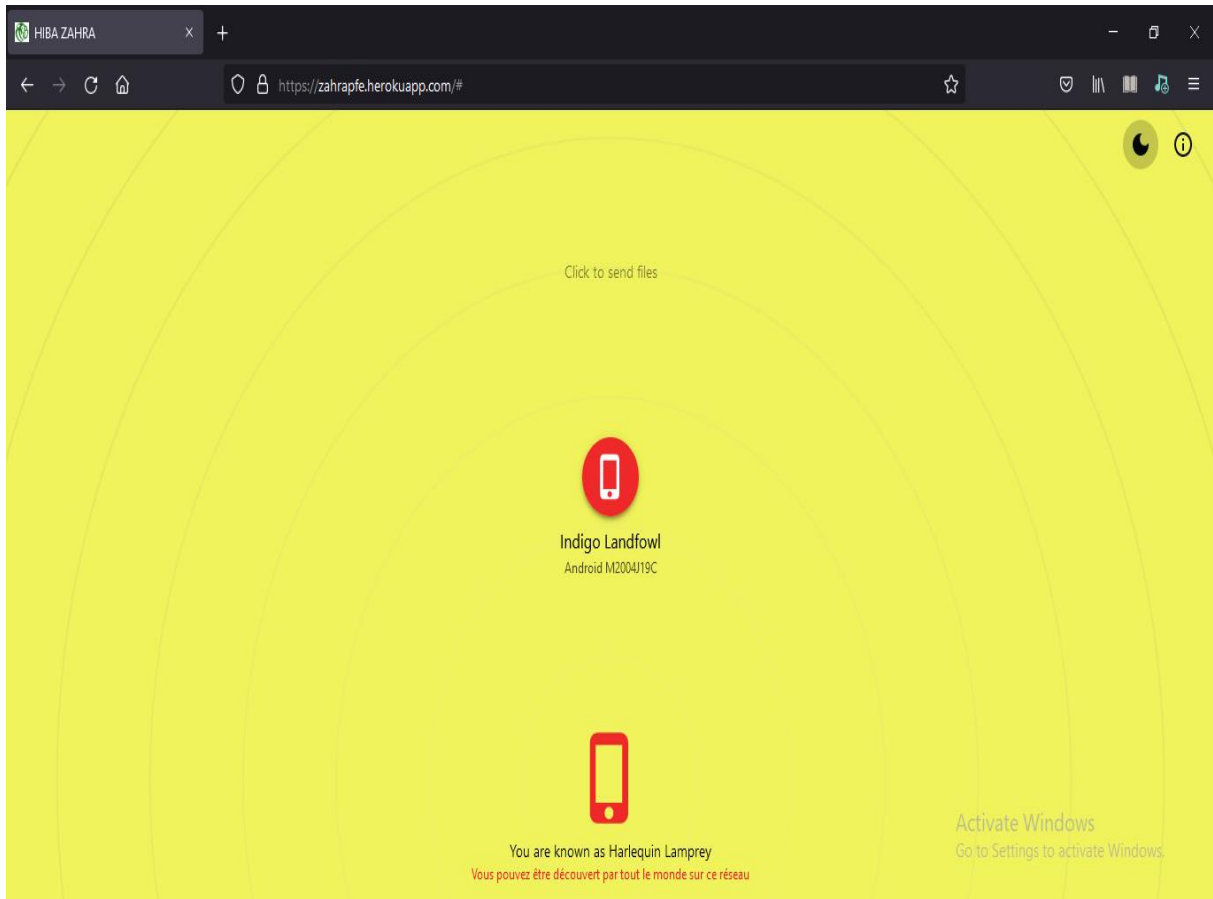
Lors de développement de ce projet de fin d'étude, nous avons profité d'applications transfert de fichiers pour simuler leur fonctionnement et concevoir une interface utilisateur adaptée Nous avons également utilisé des outils de programmation (Html, Css, Javascript, Nodejs, Heroku) qui sont tous disponibles gratuitement car notre projet est pédagogique et non commerciale, afin que d'autres étudiants puissent en bénéficier dans le développement de leurs projets.

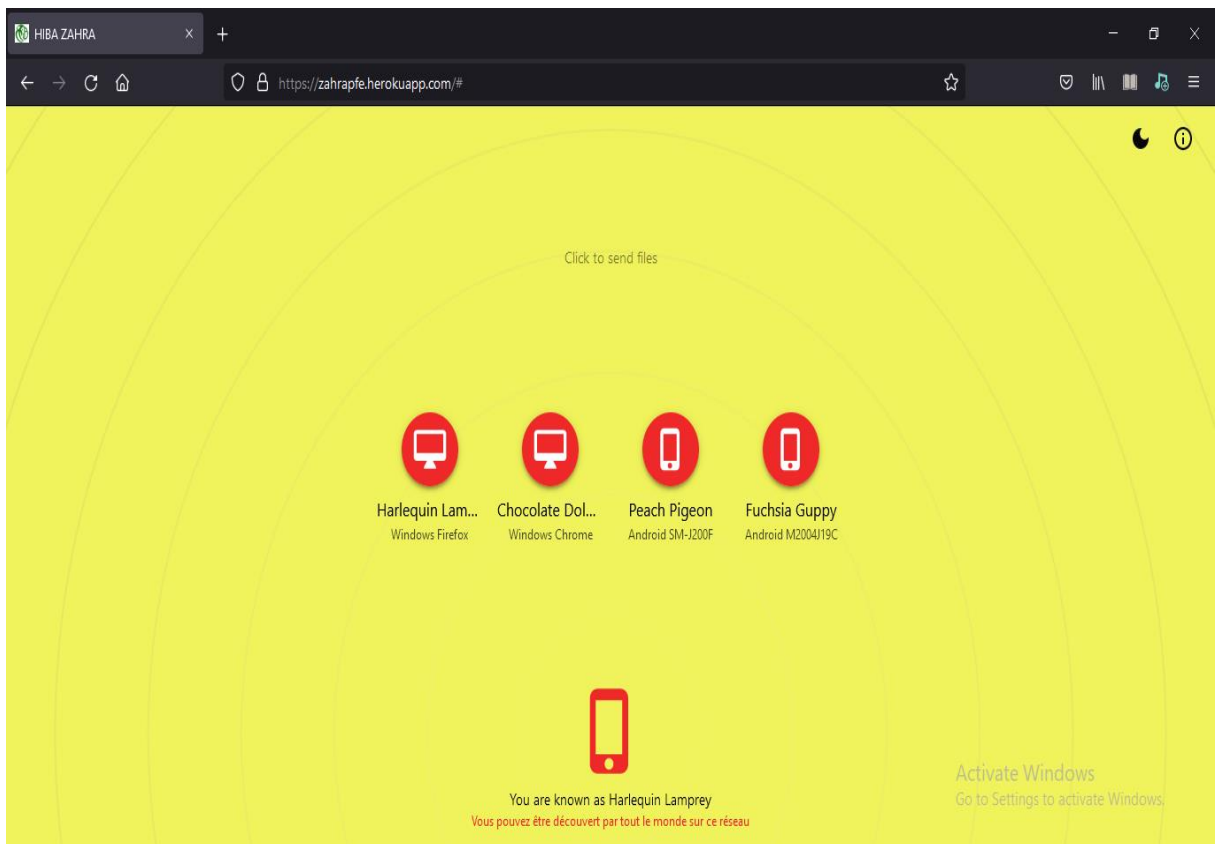
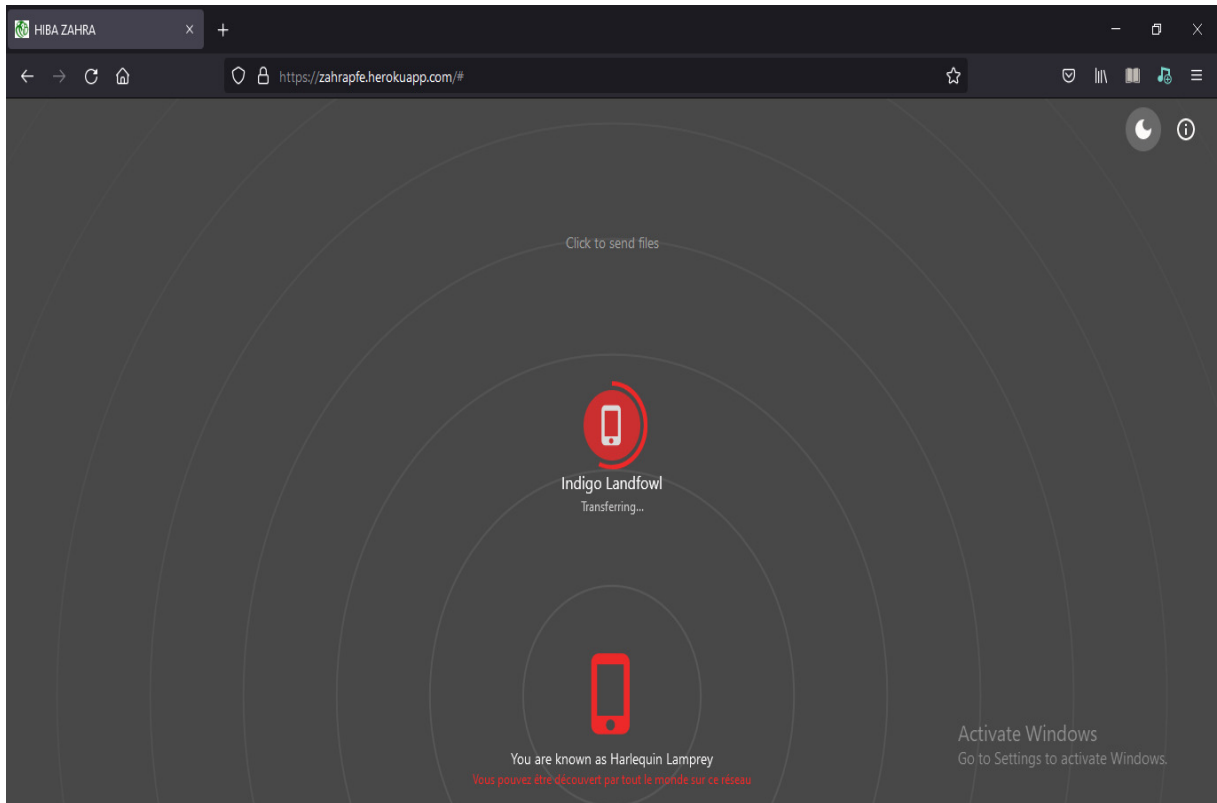
ANNEXE : L'interface de PWA transfert de fichiers

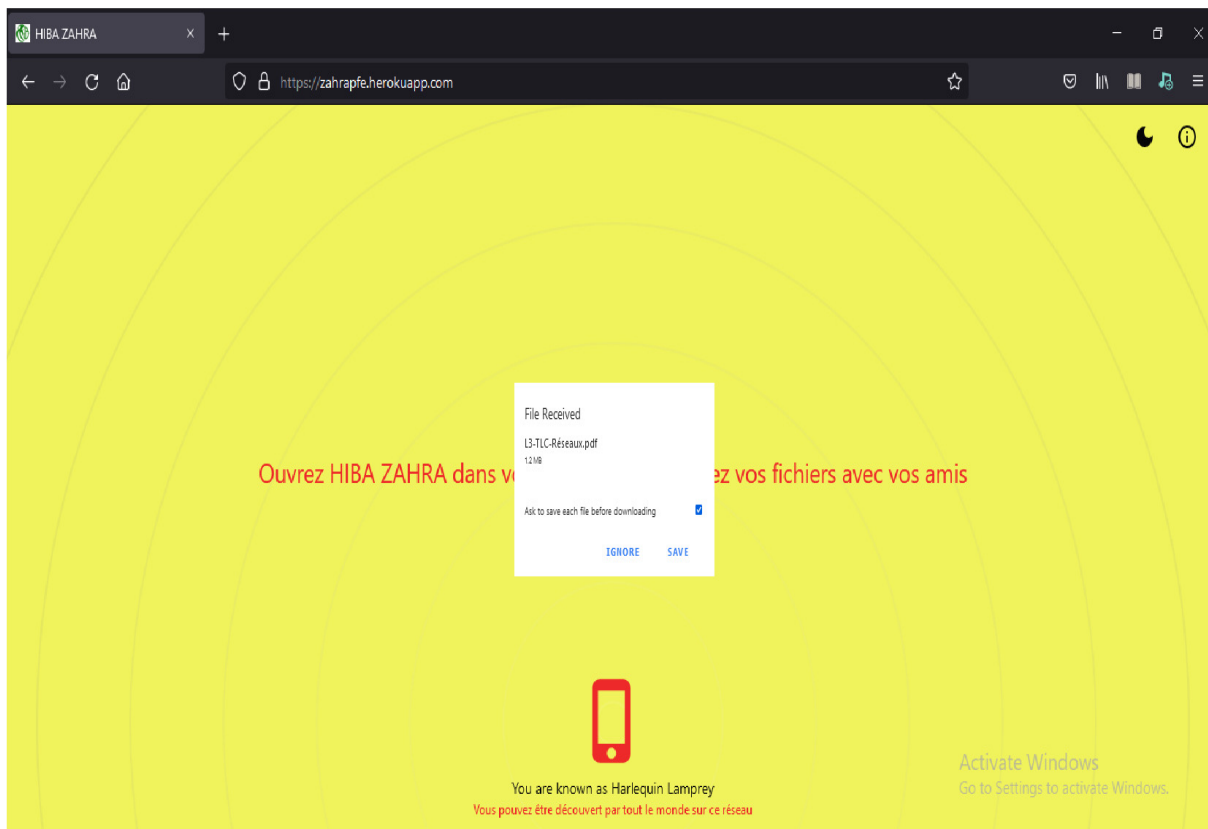
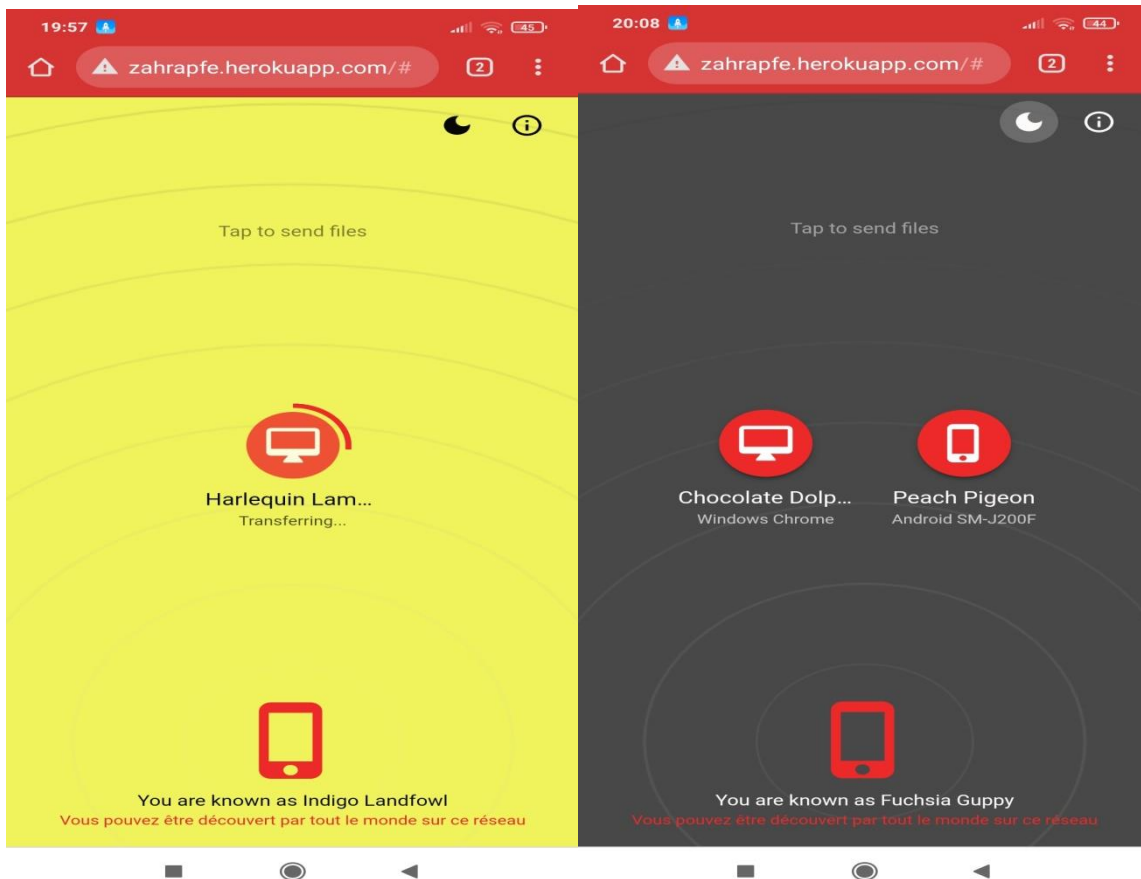


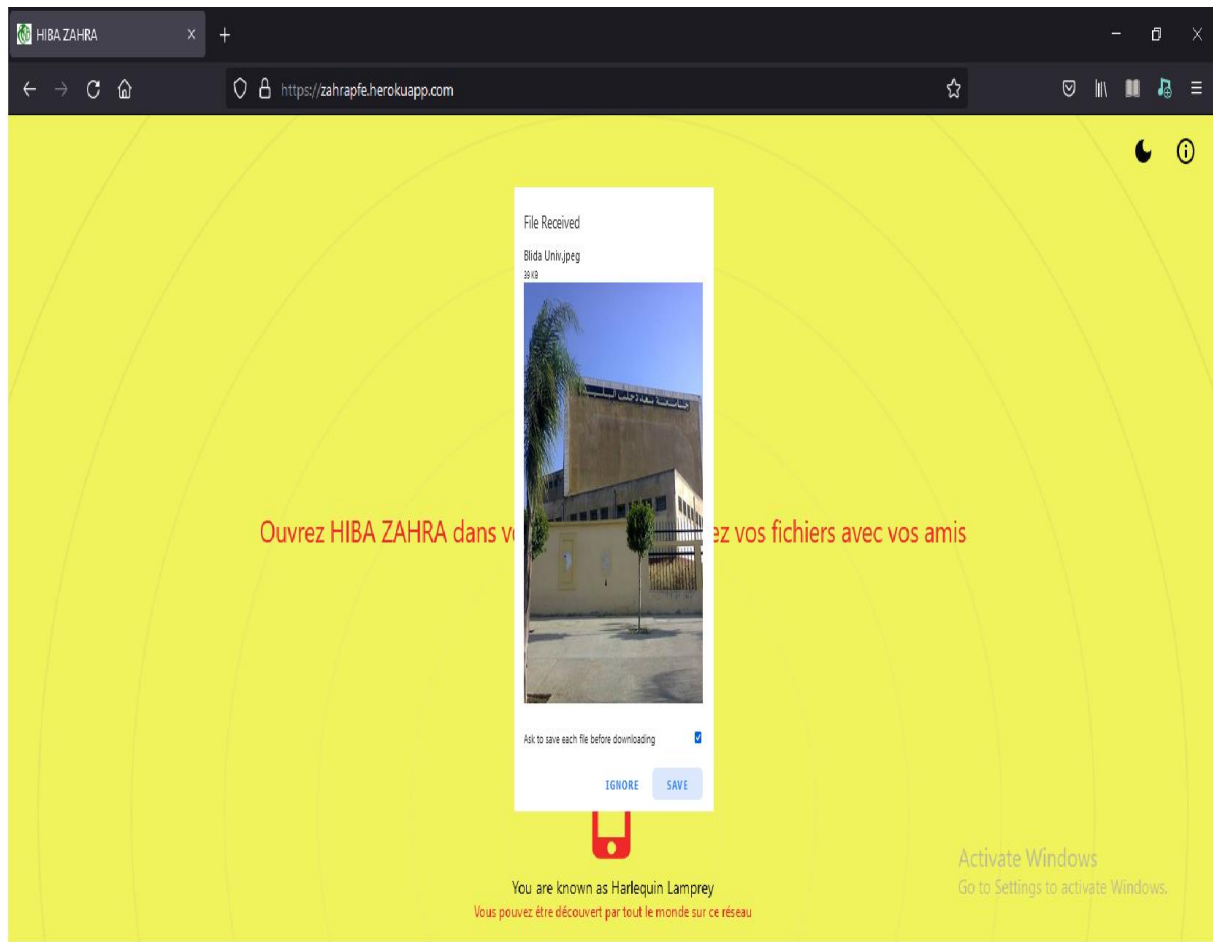












5 Bibliographie

- [1] « World Wide Web » .<https://fr.wikipedia.org>(consulté le août 05, 2021).
- [2] « l'origine du premier site Internet».<https://www.cnrs.fr>(consulté le août 05, 2021).
- [3] «Evolution of World Wide Web» . <https://www.researchgate.net>(consulté le août 05, 2021).
- [4] « Web Evolution from 1.0 to 3.0» .<https://medium.com>(consulté le août 05, 2021).
- [5] N.Sfetcu ,«Web 2.0 / Social Media / Social Networks» : MultiMedia Publishing,2017.
- [6]«Mobile Operating System» .<https://www.sciencedirect.com>(consulté le août 05, 2021).
- [7] K.K, Hiran, R.Doshi ,«Mobile Applications Development»:De Gruyter,2020.
- [8] B.Goswami ,R.Chatterjee . «Progressive Web Apps for Social Development» ,2019.
- [9] «About File Transfer Protocol (FTP) » ,<https://www.ibm.com>(consulté le août 05, 2021).
- [10] «Fonctionnement du protocole FTP» .<https://www.numelion.com>(consulté le août 05, 2021).
- [11] «FTP (File Transfer Protocol) » . <https://searchnetworking.techtarget.com>(consulté le août 05, 2021).
- [12] D. Gourley, B. Totty , «HTTP: The Definitive Guide» : O'Reilly Media , 2002.
- [13] S. A. Thomas , « HTTP Essentials» : Wiley Edition , 2001.
- [14] B. Pollard , «HTTP/2 in Action » : Manning Publications , 2019 .
- [15] GS1 France , « Guide français d'utilisation du protocole AS2 - EDI Internet » , 2010

- [16] « What is an AS2 MDN? », <https://www.jscape.com>(consulté le août 05, 2021).
- [17] « Protocol d'Echanges pour un Système Interbancaire de Télécompensation » .<https://pesit.org/spec/PeSIT-e-en.pdf>(consulté le août 05, 2021).
- [18] « WebRTC », *WebRTC*. <https://webrtc.org/?hl=fr> (consulté le août 05, 2021).
- [19] D. Ristic,«Learning WebRTC » :Packt Publishing Ltd , 2015.
- [20] A. B. Johnston and D. C. Burnett.«WebRTC: APIs and RTCWEB Protocols of the HTML5 Real-Time Web» :Digital Codex LLC , 2012.
- [21] « WebRTC » ,<https://developer.mozilla.org>(consulté le août 05, 2021).
- [22] S. Loreto et S. P. Romano, «Real-Time Communication with WebRTC: Peer-to-Peer in the Browser», 1 edition. Beijing: O'Reilly Media, 2014.
- [23]C. Love . « Progressive Web Application Development by Example » :Packt Publishing Ltd , 2018.
- [24] C. Rojas . « Building Progressive Web Applications with Vue.js » :APRESS,2020.
- [25]J.Stark . « Building Android Apps with HTML, CSS, and JavaScript » :o'reilly edition ,2010.
- [26] « Nodejs » .<https://nodejs.org/en/>(consulté le août 05, 2021).
- [27] <https://apprendre-la-programmation.net/>(consulté le août 05, 2021).
- [28] <https://heroku.com>(consulté le août 05, 2021).