

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 بليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Master II

Mention Électronique

Instrumentation Biomédicale

présenté par

ZAHRA Abdelouafi

Plate-forme E-santé Télémédecine: des technologies de l'information et de la communication (TIC) appliquées à la santé

Proposé par : Dr. ANOU Abderrahmane

Année Universitaire 2019-2020.

ملخص :

التطبيب عن بعد هو استخدام تكنولوجيا المعلومات والاتصالات في مجال الرعاية الصحية لتسهيل التواصل بين المرضى والمتخصصين في الرعاية الصحية، من خلال السماح بنقل الفيديو والصوت عبر الإنترنت .

نقترح في هذا العمل نظامًا للتطبيب عن بُعد يعتمد على واجهة برمجة التطبيقات WebRTC لإرسال الفيديو والصوت في الوقت الفعلي عبر الإنترنت من خلال متصفح الويب إلى جانب التحكم في الميكروفون وكاميرا الويب.

الكلمات الجوهرية: الطب عن بعد، تكنولوجيا المعلومات والاتصالات، واجهة برمجة التطبيقات، متصفح الويب.

Résumé :

La télémédecine est l'utilisation des technologies de l'information et de la communication (TIC) dans le domaine des soins de santé pour faciliter la communication entre les patients et les professionnels de la santé, en permettant le transfert vidéo et vocal sur Internet.

Dans ce travail, on propose une plate-forme de télémédecine basée sur WebRTC Application Programming Interface (API) pour transmettre la vidéo et la voix en temps réel sur Internet, via un navigateur Web. Outre le contrôle du microphone et de la webcam

Mots-clés: télémédecine, technologies de l'information et de la communication (TIC), interface de programmation d'applications (API), navigateur Web.

Abstract :

Telemedicine is the use of information and communication technology (ICT) in healthcare domaine to smooth the communication between patients and healthcare professionals, by allowing video and voice transfer over the Internet.

In this work we propose a telemedicine platform based on WebRTC Application Programming Interface (API) to transmit video and voice in real time over the Internet, through a web browser.

Besides microphone and webcam control

Keywords: Telemedicine, Information and communication technology (ICT), Application Programming Interface (API), web browser.

Dédicace

A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études.

Remerciement :

En premier lieu, je tiens à remercier DIEU le tout puissant qui ma donner la force, le courage et la patience pour continuer mes études et d'accomplir ce travail.

J'exprime mes respectueux dévouements à mon promoteur Dr ANOU Abderrahmane pour son aide, sa compréhension, son observation et sa disponibilité durant la réalisation de ce mémoire.

Mes vifs remerciements vont également à monsieur le président de jury qui à bien voulu accepter de présider le jury de ce mémoire, ainsi qu'a tous les membres de jury.

Je tiens à exprimer ma profonde gratitude à mon professeur Mr CHERFA Yazid, pour sa disponibilité, ses encouragements et son soutien autant scientifique que moral qui m'ont permis d'aller au bout de mon PFE, ainsi que pour tous ses conseils judicieux quant à mes orientations futures.

Je remercie vivement tous les professeurs et enseignants de l'Université de Blida 1, plus particulièrement ceux du département d'électronique, qui ont contribué à ma formation.

Table des matières

Introduction Générale	1
CHAPITRE 1 : La santé dans l'environnement internet.....	3
1-1 INTRODUCTION.....	3
1-2 Internet Naissance et évolution	3
1.2.1 L'histoire d'Internet.....	3
1.2.2 L'Internet des années 90	4
1.2.3 Les années 2000 et le Web 2.0.....	4
1.3 La E-santé : historique, définitions, et contexte	4
1.3.1 Origine du concept « E-santé »	4
1.3.2 Définitions et contextes	4
1.4 Quelques exemples d'applications santé	7
1.4.1 Cardio	7
1.4.2 Familivac	8
1.5 Quelques exemples dispositifs médicaux connectés	8
1.5.1 Stéthoscope	8
1.5.2 Tensiomètre.....	9
1.6 Conclusion	9
CHAPITRE 2 : La télémédecine ou la télécommunication biomédicale	11
2.1 Introduction.....	11
2.2 La télémédecine	11
2.3 Télémédecine Station de travail et techniques d'interfaçage	13
2.3.1 Plateforme matérielle de télémédecine	13
2.3.2 Configurations de réseautage informatique	16
2.3.3 Logiciel de télémédecine	17
2.3.4 Interface des dispositifs médicaux avec les ordinateurs	18
2.3.5 Types d'interfaces	19
2.4 Normes d'échange de données médicales	22
2.5 Conclusion.....	26
CHAPITRE 3 : La conception de la plate-forme biomédicale	27

3.1 Introduction.....	27
3.2 Adressage dans le réseau IP.....	288
3.2.1 Adresse IP.....	288
3.2.2 Adresse privée et adresse publique.....	30
3.2.3 Notion de port	31
3.2.4 Le protocole TCP	31
3.2.5 Le protocole UDP.....	32
3.3 La technologie WebRTC.....	32
3.3.1 Architecture Web	33
3.3.2 Architecture WebRTC	33
3.3.3 WebRTC dans le navigateur	35
3.3.4 Prise en charge du navigateur.....	35
3.3.5 Le signaling.....	36
3.3.6 Le framework ICE et les protocoles STUN et TURN.....	37
3.3.7 Connexion au serveur de signalisation	39
3.3.8 Les API WebRTC	41
3.3.9 Accès aux périphériques multimédias	41
3.3.10 Sécurité	43
3.4 Définition de HTML 5	44
3.5 Définition de CSS 3 et Bootstrap.....	46
3.6 Définition de JavaScript	47
3.7 Définition de NODE.JS	49
3.7.1 Définition de Node.js : le JavaScript côté serveur.....	50
3.7.2 Définition de NPM	51
3.7.3 Configurer un serveur statique (local)	51
3.7.4 Le Déploiement de GBM TeleMed.....	53
3.8 Conclusion.....	56
Conclusion Général.....	58
Annexe.....	59
Bibliographie.....	61

Liste des figures

FIGURE 1.1 : DIVISIONS DE LA E-SANTÉ [3].....	5
FIGURE 1.2 : ESTIMATION DU NOMBRE D'OBJETS CONNECTÉS DE SANTÉ (OCS) DANS LE MONDE ENTRE 2015 ET 2020 [7].....	7
FIGURE 1.3 : CARDIO.	7
FIGURE 1.4 : FAMILIVAC.	8
FIGURE 1.5 : STÉTHOSCOPE.	8
FIGURE 1.6 : TENSIOMÈTRE.	9
FIGURE 2.1 LES DIVISIONS DE LA TÉLÉMÉDECINE.	12
FIGURE 2.2 TÉLÉCONSULTATION CÔTÉ SPÉCIALISTE.	13
FIGURE 2.3 EXEMPLES D'IMAGES IRM UTILISÉES POUR L'ANALYSE.	14
FIGURE 2.4 STATION DE TRAVAIL MOBILE.....	15
FIGURE 2.5 SYSTÈME CLIENT-SERVEUR TYPIQUE.	16
FIGURE 2.6 CLOUD COMPUTING EN TÉLÉRADIOLOGIE.	17
FIGURE 2.7 : L'INTERFACE D'UN DMP.	18
FIGURE 2.8 : RS-232 PINOUT (25 PIN).	19
FIGURE 2.9 : CONNECTEUR SÉRIE R232.	19
FIGURE 2.10 :EXPORTATION DE DONNÉES À PARTIR D'APPAREILS MÉDICAUX DANS LE PASSÉ, LE PRÉSENT ET L'AVENIR [14].....	20
FIGURE 2.11 : IEEE-488 CONNECTEURS.	21
FIGURE 2.12 : IEEE-488.	22
FIGURE 2.13 : USB TYPE A ET B.....	22
FIGURE 2.14 : LE SERVEUR HL7 GATEWAY.	23
FIGURE 2.15 : DICOM LOGO.	23
FIGURE 2.16 : IMAGE MÉDICALE EN FORMAT DICOM.	24
FIGURE 2.17 : LES NŒUDS DU RÉSEAU MIB.	24
FIGURE 2.18 : CONFIGURATION DU PORT COM AVEC LE PROTOCOLE ASTM (E1381OU E1394).....	25
FIGURE 2.19 : NCPDP LOGO.	25
FIGURE 3.1 : FORMAT ADRESSE IPv4.	28
FIGURE 3.2 : FORMAT IPv4 CLASSE A.....	28
FIGURE 3.3 : FORMAT IPv4 CLASSE B.	28
FIGURE 3.4 : FORMAT IPv4 CLASSE C.	29
FIGURE 3.5 : TRANSLATION D'ADRESSE.	31
FIGURE 3.6 : WEBRTC LOGO.	32
FIGURE 3.7 : HISTORIQUE DES SOLUTIONS DE VOIX ET VIDÉO SUR IP.	33
FIGURE 3.8 : LE TRAPÈZE WEBRTC.....	34
FIGURE 3.9 : LE TRIANGLE WEBRTC.....	34
FIGURE 3.10 : COMMUNICATION EN TEMPS RÉEL DANS LE NAVIGATEUR.	35
FIGURE 3.11 : ARCHITECTURE JSEP.	37
FIGURE 3.12 :UTILISATION DE SERVEURS STUN POUR OBTENIR LES IP PUBLIQUES.....	38

FIGURE 3. 13 : SCALEDRONE LOGO.....	39
FIGURE 3. 14 : EXEMPLE DE MULTIPPOINT CONTROL UNIT.....	40
FIGURE 3. 15 : OUVRIR NOTRE EXEMPLE DANS CHROME.....	43
FIGURE 3. 16 : OUVRIR NOTRE EXEMPLE DANS FIREFOX.....	43
FIGURE 3. 17 : OUVRIR NOTRE EXEMPLE DANS OPERA.....	43
FIGURE 3. 18 : LOGO DU HTML.....	44
FIGURE 3. 19 : EXEMPLE DE STRUCTURE DES ÉLÉMENTS EN HTML 5.....	46
FIGURE 3. 20 : OUVRIR NOTRE HTML DANS UN NAVIGATEUR.....	46
FIGURE 3. 21 EXEMPLE DU CODE CSS 3.....	47
FIGURE 3. 22 : EXEMPLE DU CODE BOOTSTRAP.....	47
FIGURE 3. 23 : JAVASCRIPT EST UN LANGAGE DIT CLIENT-SIDE.....	48
FIGURE 3. 24 : EXEMPLE DU CODE JS (AVANT ET APRÈS CLICKER).....	49
FIGURE 3. 25 : PAGE D' ACCUEIL GBM TELEMED (CRÉÉ PAR HTML5,CSS3,BOOTSTRAP,JS).....	49
FIGURE 3. 26 : LE LOGO NODE.JS.....	50
FIGURE 3. 27 : PHP SUR LE SERVEUR, JAVASCRIPT CHEZ LE CLIENT.....	50
FIGURE 3. 28 : JAVASCRIPT SUR LE SERVEUR AVEC NODE.JS.....	50
FIGURE 3. 29 : LE LOGO NPM.....	51
FIGURE 3. 30 :LA PAGE VIDEO CALL DE GBM TELEMED.....	55
FIGURE 3. 31 : TÉLÉCONSULTATION AVEC GBM TELEMED.....	56

Liste des tableaux

TABLEAU 2. 1 : BROCHAGE RS 232 [13].....	20
TABLEAU 2. 2 : BROCHAGE IEEE488 24 POINTS.....	21
TABLEAU 3. 1 : PLAGE D' ADRESSE PRIVÉE.....	30
TABLEAU 3. 2 : EXEMPLES D' ASSOCIATION DE SERVICES ET DE NUMÉROS DE PORTS.....	31

Listes des acronymes et abréviations

API	: Application Programming Interface
CDA	: Clinical Document Architecture
CSS	: Cascading Style Sheets
DICOM	: Digital imaging and communications in medicine
DMP	: Dossier Médical Personnalisé
DNS	: Domain Name System
DPI	: Deep packet inspection
DTLS	: Datagram Transport Layer Security
FTP	: File Transfer Protocol
GBMTeleMed	: Génie Biomedical télé-médecine
GIPS	: Global IP Solutions
HIPAA	: Healthcare Insurance Portability and Accountability Act
HTML	: HyperText Markup Language
HTTP	: Hypertext Transfer Protocol
ICE	: Interactive Connectivity Establishment
IEEE	: Institute of Electrical and Electronics Engineer.
IETF	: Internet Engineering Task Force
IP	: Internet Protocol
Ipv4	: Internet Protocol version 4
JIT	: Just In Time
JS	: JavaScript
JSEP	: JavaScript Session Establishment Protocol
MCU	: Multipoint Control Unit
MIB	: Medical Interface Bus-
MQTT	: Message Queuing Telemetry Transport
NAT	: Network address translation
NPM	: Node Package Manager
PHP	: Hypertext Preprocessor
SIH	: Système d'information hospitalier
SIP	: Session Initiation Protocol
SIS	: Système d'information de santé
SMTP	: Simple Mail Transfer Protocol
SRTP	: Secure Real-time Transport Protocol

SSL	: Secure Sockets Layer
STUN	: Session Traversal Utilities for NAT
TCP	: Transmission Control Protocol
Telnet	: Terminal network ou telecommunication network
TIC	: Technologie d'information et de la communication
TURN	: Traversal Using Relays around NAT
UDP	: User Datagram Protocol
URI	: Uniform Resource Identifier
URL	: Uniform Resource Locator
USB	: Universal Serial Bus
VSCODE	: Visual Studio Code
W3C	: World Wide Web Consortium
WebRTC	: Web Real-Time Communication
Wi-Fi	: Wireless Fidelity

Notations

Introduction Générale

Introduction Générale

Depuis la nuit des temps, la science-fiction fascine le monde, avec ses inventions, ses moyens de transport dans le temps, et ses outils de communication audio-visuels immatériels. Le développement de la technologie et de l'informatique, a fait qu'aujourd'hui le monde arrive à communiquer en audio et en visio en même temps, avec des petits outils légers, pour le voyage dans le temps, on a rien de crédible pour l'instant, mais personne ne sait ce que les années ou les prochains mois réservent à l'humanité.

Dans le domaine médical, la technologie ne cesse pas de se développer aussi, à un point que l'individu se trouve à poser des questions comme :

Le médecin va-t-il être toujours indispensable dans la vie et la ville de demain ? Quel rôle va jouer le développement des appareils médicaux dans les hôpitaux de demain ? Est-il possible qu'un jour le développement technologique des outils médicaux arrive à connecter les êtres humains entre eux, avec une sorte de matrice, déconnectée du monde matériel, pour surfer dans un monde immatériel, virtuel parallèle ?

En étant dans une ère de croissance numérique continue, depuis plus d'une dizaine d'année, les inventions et le développement des outils médicaux facilitent de plus en plus la pratique de la médecine classique. De nos jours, Internet et les appareils médicaux font partie du quotidien du médecin pour l'aider à exercer son travail, et des patients aussi, car cela leur permet de se renseigner sur leur maladie, les soins possibles, et même vérifier et comparer le diagnostic de plusieurs médecins (sur internet)

Aujourd'hui, plus le médecin a de nouveaux outils pour travailler, plus les patients lui font confiance, comme si le patient trouve son assurance dans les appareils et outils utilisés plus que dans le diagnostic du médecin,

Le patient d'aujourd'hui, de spectateur, il devient de plus en plus acteur, grâce à l'accès à l'information, qui n'est plus détenue par les médecins, le patient d'aujourd'hui n'a pas souvent besoin de son médecin, grâce au développement technologique, plusieurs appareils peuvent déjà remplacer le travail du médecin et des laboratoires, comme l'appareil qui mesure l'attention, le rythme cardiaque, le taux du sucre dans le sang, ...etc. Et en étant en ère de développement et de croissance numérique, ces appareils risquent de devenir immatériels, et d'autres moyens et outils médicaux immatériels peuvent voir le jour.

L'objectif de ce mémoire est d'expliquer la conception et le développement d'une plate-forme E-santé qui permet d'effectuer les différentes applications de télémédecine.

Ce mémoire est constitué de trois chapitres :

Le premier chapitre se compose de deux sections séparées ; La première donne un aperçu sur l'évolution d'internet, et la seconde décrit la notion de l'E-santé.

Dans le second chapitre, on se concentre sur la présentation du domaine de notre travail. Cette présentation porte sur la télémédecine, applications, théorie et l'aspect technique.

Dans le dernier chapitre, on propose notre modèle de la plate-forme télémédecine, et leur ingrédients techniques et théoriques inclus dans la conception

Le mémoire se termine par une conclusion qui résume le travail effectué, suivie par une annexe qui complète le manuscrit.

CHAPITRE 1

CHAPITRE 1 : La santé dans l'environnement internet

1-1 INTRODUCTION

Les années 2000 sont un tournant de la communication grâce au développement des nouvelles technologies d'information et de la communication (TIC). Le web se transforme en un web de partage, simple, accessible et interactif, permettant aux internautes ayant peu de connaissances techniques de s'approprier les nouvelles fonctionnalités d'Internet. L'essor des téléphones intelligents ou Smartphones, disposant d'un assistant numérique personnel, octroie aux utilisateurs la capacité de rester connecté avec le monde indépendamment de leur localisation. Cette évolution numérique est maintenant largement adoptée dans de nombreux secteurs d'activités tels que l'agro-alimentaire, la finance ou encore le tourisme.

Malgré sa prépondérance et l'activité économique qu'il génère, le secteur de la santé (médical ou pharmaceutique) était jusqu'ici à l'écart du développement de l'outil numérique. Depuis quelques années, on peut voir apparaître une certaine adaptation du domaine médical avec notamment l'apparition de la e-santé.

1-2 Internet Naissance et évolution

1.2.1 L'histoire d'Internet

La révolution numérique est née au XXème siècle. Qualifiée de quatrième révolution industrielle, C'est en septembre 1967 que pour la première fois deux calculateurs électroniques échangent des données par paquets par le biais d'un câble de cuivre, à l'Université de Californie – Los Angeles (UCLA). Ces calculateurs peuvent être assimilés à la forme ancestrale du *personal computer* (PC), telle qu'elle est connue aujourd'hui. En novembre 1969 naissait l'Arpanet (*Advanced Research Projects Agency Network*) , premier réseau d'ordinateurs situés en Californie et en Utah aux Etats-Unis, formant ainsi le premier réseau dit « décentralisé » et dont la visée était militaire – l'ancêtre d'Internet [1].

Cinq ans plus tard, en 1972, le projet prend une nouvelle tournure : on souhaite coupler ce premier réseau Arpanet à d'autres appareils, en utilisant un nouveau protocole de transmission, le TCP/IP (Transmission Control Protocol/Internet Protocol). Le projet prend rapidement le nom « *internetting* », et des applications commerciales commencent à voir le jour. Mais la naissance d'Internet tel que nous le connaissons date de 1989, au Centre Européen de Recherche Nucléaire (CERN) à la frontière franco-suisse.

C'est à cette date notamment qu'a été créée l'URL (pour *Uniform Resource Locator*) par le chercheur Tim Berners-Lee, par souci de simplification du protocole TCP/IP, et pouvant être définie comme une suite de caractères menant à une ressource web. Le *World Wide Web*, ou *WWW*, quant à lui est créé en 1991, par Tim Berners-Lee également, à destination du grand public, il regroupe l'ensemble des ressources en ligne, accessibles via une URL, c'est-à-dire des pages internet. Le premier pas vers la révolution numérique est fait [1].

1.2.2 L'Internet des années 90

L'internet que nous connaissons est né au début des années 1990, forgé par la convergence d'une innovation technique, le protocole IP et ses équipements de routage, et d'une innovation d'usage, le lien hypertexte. Un web 1.0 s'est développé jusqu'au tournant du millénaire sur les outils de portail puis de moteur de recherche, donnant naissance à une vive spéculation financière repérée comme bulle internet. C'est l'époque où se forge l'industrie internet, celle de Microsoft, de Yahoo!, d'Amazon, d'IAC, d'eBay, puis de Google. C'est l'époque où internet promet une nouvelle économie et une nouvelle société [2].

1.2.3 Les années 2000 et le Web 2.0

Le 21^e siècle apporte la connexion haut débit permanente, puis l'ère du smartphone :

Le web 2.0 se met en place, fonder sur des relations sociales et commerciales plus intenses. Mais alors que la toile offre de plus en plus de services et de connexions et que l'univers s'y connecte et fait littéralement exploser ses trafics, surgisse un certain nombre d'interrogations et de doutes sur les promesses d'internet., le Web 2.0 a redéfini l'internaute comme acteur-clé de création de contenu, de partage et de collaboration [2].

1.3 La E-santé : historique, définitions, et contexte

1.3.1 Origine du concept « E-santé »

Le concept lié au terme d'E-santé aurait été inventé en 1999 par le consultant en santé australien John Mitchell, lors du 7^{ème} congrès international de télémédecine à Londres. Ce dernier l'a alors défini comme « l'usage combiné de l'internet et des technologies de l'information à des fins cliniques, éducationnelles et administratives, à la fois localement et à distance » [1].

1.3.2 Définitions et contextes

1.3.2.1 La E-santé

L'Organisation Mondiale de la Santé (OMS) définit la E-santé comme « *les services du numérique au service du bien-être de la personne* », c'est-à-dire comme « *l'application des*

technologies de l'information et de la communication (TIC) au domaine de la santé et du bien-être ».

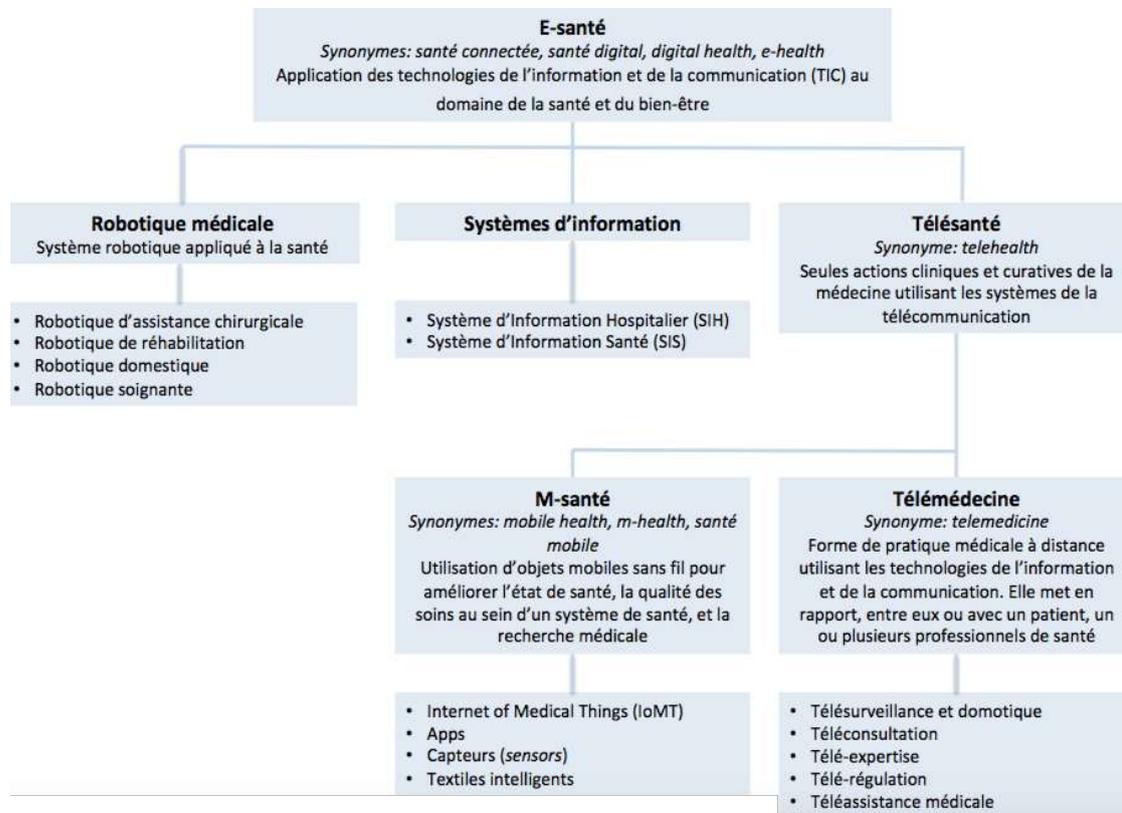


Figure 1.1 : Divisions de la E-santé [3].

La E-santé regroupe un certain nombre de pratiques :

A. Robotique médicale

Les robots médicaux sont un type de robots de service professionnel utilisés dans et hors des hôpitaux pour améliorer le niveau global de soins aux patients. Ils allègent la charge de travail du personnel médical, ce qui leur permet de passer plus de temps à s'occuper directement des patients, tout en créant d'importantes efficacités opérationnelles et des réductions de coûts pour les établissements de santé [4].

Les applications diverses des robots médicaux incluent aujourd'hui les robots de télé-présence pour les soins à distance, les robots désinfectants pour réduire les infections nosocomiales, les robots qui peuvent prélever le sang plus précisément et plus efficacement, et les exosquelettes robotiques qui fournissent un soutien externe et un entraînement musculaire pour la réadaptation. Des robots médicaux mobiles sont également utilisés pour la livraison de médicaments et d'autres matériaux sensibles en milieu hospitalier [4].

B. Le Systèmes d'information

Les systèmes d'information de santé (SIS) ou hospitaliers (SIH):ceux-ci organisent les échanges d'information entre hôpital et médecine de ville ou entre services d'un même hôpital Ainsi que le Dossier Médical Personnalisé (DMP) est un service conçu pour contribuer à l'amélioration de la coordination, de la continuité et la qualité des soins. Il s'agit d'un dossier informatisé et sécurisé, qui accompagne le patient tout au long de sa vie.

C. La télésanté

La télésanté regroupe la télémédecine et la m-santé

C – 1 la télémédecine (Voir Chapitre 2)

Grâce aux nouvelles technologies, la télémédecine permet l'accès à distance d'un patient à un médecin ou à une équipe médicale. Elle représente une autre manière de soigner, avec les mêmes exigences de qualité et de sécurité.

C – 2 La m-santé

La m-santé, ou santé-mobile (mobile Health en anglais), regroupe tous les services touchant de près ou de loin à la santé, disponibles en permanence via les Smartphones. Elle consiste en l'utilisation des technologies modernes mobiles, tels que les Smartphones afin de délivrer, renforcer et améliorer les services de santé [5].

Sur le plan des usages, le périmètre de la m-santé s'étend des fonctions basiques du téléphone (voix ou SMS) aux fonctionnalités les plus sophistiquées faisant appel aux technologies les plus récentes : smartphone et tablette, qui sont devenus les points d'accès Internet quasi exclusifs.

La m-santé est divisée en deux catégories majeures d'outils : les applications mobiles et les objets connectés de santé.

Applications mobiles de santé/bien être (« Apps »)

Ce sont des logiciels spécifiquement conçus pour fonctionner sur un équipement tel que smartphone ou tablette.

Ces applications permettent un suivi de la santé des individus et de leur qualité de vie en temps réel, avec un impact potentiellement significatif sur le dépistage de maladie ou encore sur l'adhésion des patients à leur traitement. Par exemple, l'envoi de SMS peut aider les patients à respecter leurs prises médicamenteuses [6].

Les objets connectés de santé/bien être

Sont des objets générant de l'information physiologique synchronisée via un réseau sans fil (Wi-Fi ou Bluetooth) à un téléphone portable, une tablette, ou au site internet du fabricant. Ils communiquent avec d'autres systèmes informatiques pour obtenir ou fournir d'autres

informations. Ils sont appelés « objets intelligents ». Ce sont des objets destinés à un usage médical ou, souvent à gérer la santé et le bien-être. Ils permettent de collecter des données en vie réelle à domicile de l'individu, de les stocker, les traiter et les diffuser ou peuvent accomplir des actions spécifiques en fonction des informations reçues. Le tensiomètre, la balance, ou encore la montre connectée, sont des exemples très largement utilisés dans la population générale. De plus en plus d'objets intégrés au corps commencent à apparaître, comme les lentilles mesurant la glycémie ou le patch électronique greffé sous la peau [6].

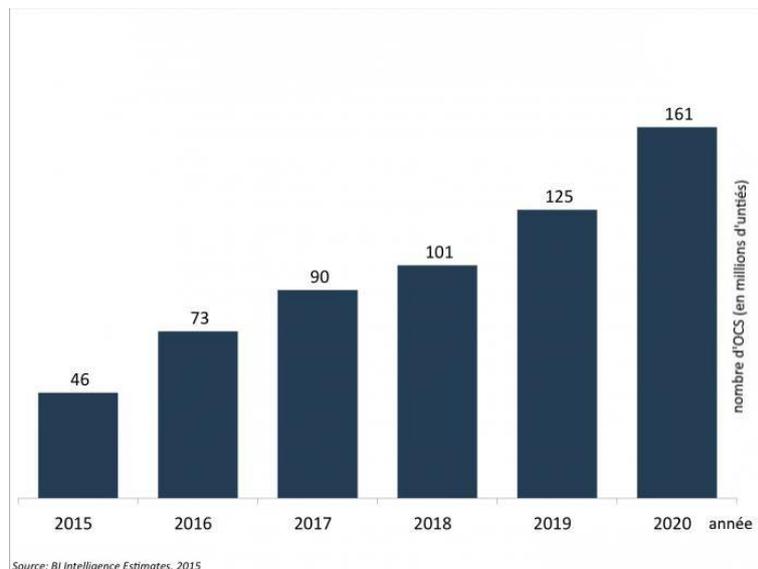


Figure 1. 2 : Estimation du nombre d'objets connectés de santé (OCS) dans le Monde entre 2015 et 2020 [7].

1.4 Quelques exemples d'applications santé

1.4.1 Cardio

Développée par des chercheurs du MIT et de Harvard, Cardio permet de mesurer son rythme cardiaque en disposant simplement son doigt au niveau de la caméra d'un Smartphone [5].



Figure 1. 3 : Cardio.

1.4.2 Familivac

Le laboratoire Sanofi Pasteur MSD a lancé une application dédiée à la mémoire santé de la famille. Familivac permet d'informer l'utilisateur des vaccinations recommandées pour l'ensemble de la famille mais aussi de stocker les informations santé telles que : antécédents, numéros de sécurité sociale, allergies [5].



Figure 1. 4 : Familivac.

1.5 Quelques exemples dispositifs médicaux connectés

1.5.1 Stéthoscope

L'auscultation se passe comme avec n'importe quel stéthoscope classique. Les données sont transférées via une connexion Bluetooth à un périphérique fonctionnant sous IOS, ANDROID ou Windows [7].



Figure 1. 5: Stéthoscope.

1.5.2 Tensiomètre

L'appareil se place autour du bras comme n'importe quel autre tensiomètre non connecté, sa particularité est d'afficher les données sur le smartphone dont le patient doit être équipé. Les données sont transférées par Bluetooth sur une application dédiée (propre à chaque constructeur) [7].



Figure 1. 6: Tensiomètre.

1.6 Conclusion

Dans ce chapitre, on a présenté l'histoire et le développement d'internet ainsi que le Concept du E-santé et les différents types de divisions.

CHAPITRE 2

CHAPITRE 2 : La télémédecine ou la télécommunication biomédicale

2.1 Introduction

La télémédecine est une autre manière de soigner, avec les mêmes exigences de qualité et de sécurité que des actes classiques. Elle fait évoluer la médecine pour répondre à des défis tels que le vieillissement de la population ou encore le suivi approfondi des maladies chroniques. Elle est également un vecteur important d'amélioration de l'accès aux soins, en particulier dans les zones fragiles. En effet, elle permet une meilleure prise en charge au plus près du lieu de vie des patients et contribue aux prises en charge coordonnées par les professionnels de santé et les professionnels du secteur médico-social. Elle constitue aussi un facteur d'amélioration de l'efficacité et de l'organisation des soins [8].

2.2 La télémédecine

Le terme (Télé) est un préfixe d'origine grec signifiant loin ou à distance. Ainsi la télémédecine signifie : ((Médecine dispensée (délivrée) à distance. Cette définition suggère que la télémédecine est limitée au traitement des patients par des médecins. Pourtant ceci devient de moins en moins correct avec le développement de la pratique télé médicale. Au-delà de cette suggestion la définition ne donne aucun indice de la façon d'application de la médecine à distance. C'est pourquoi une autre définition plus précise et plus informative est apparue : « *Le transfert des données électroniques médicales d'un endroit à l'autre* ».

Alors que cette définition implique l'utilisation des TIC pour le transfert et remplace le mot (Médecine) par le terme « transfert des données médicales » elle ne dit rien sur le but de ce transfert.

En 1999 une nouvelle définition apparaît dans le cadre d'un congrès porté sur la télémédecine aux EU (Etats-Unis) : (*La télémédecine utilise les technologies de l'information et des communications (TIC) pour transférer des informations médicales pour le diagnostic, le traitement et l'éducation*).

Les données médicales peuvent être des images, des données (Visio) et (audio) en temps réel et non réel le dossier médical du patient et des données obtenues par les dispositifs médicaux. La télémédecine n'a pas pour objectif de remplacer les actes médicaux en présentiel mais elle est complémentaire. Elle ne se substitue pas aux pratiques médicales plus habituelles mais constitue une réponse aux défis auxquels est confrontée l'offre de soins aujourd'hui [9].

La télémédecine comprend la téléconsultation, la télé-expertise, la télé-surveillance et la téléassistance :

- **La téléconsultation** est une consultation médicale qui met en relation, à distance, le patient et un ou des médecins et, le cas échéant d'autres professionnels de la santé.
- **La télé-expertise** est une forme de téléconsultation suivant laquelle un acte médical est posé à distance par un médecin sans la présence du patient à des fins diagnostiques ou thérapeutiques en réponse à une demande de consultation par un collègue médecin ou un tiers.
- **La télé-surveillance** est le monitoring à distance par un médecin de données cliniques, radiologiques ou biologiques d'un patient transmis par TIC, qu'elles soient recueillies par le patient lui-même, un médecin ou un autre professionnel de la santé à des fins de diagnostic ou de traitement [9].
- **La téléassistance** est un acte médical posé par un médecin lorsqu'il assiste à distance un autre médecin ou un autre professionnel de la santé en train de réaliser un acte médical ou chirurgical.

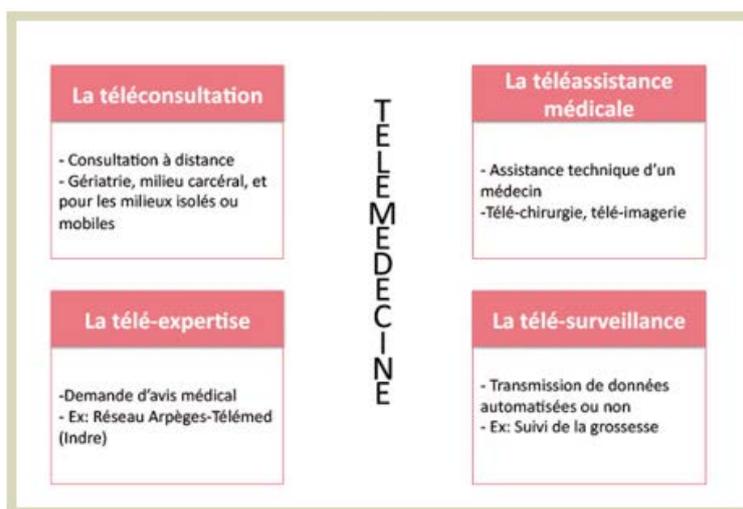


Figure 2. 1 Les divisions de la télémédecine.



Figure 2. 2 Téléconsultation côté spécialiste.

2.3 Télémédecine Station de travail et techniques d'interfaçage

2.3.1 Plateforme matérielle de télémédecine

Dans un système de télémédecine, l'échange d'informations, le traitement et la gestion des données nécessitent des ordinateurs capables de communiquer et de faciliter la sélection des données, l'application des protocoles de transmission et l'assurance de la sécurité. De plus, les ordinateurs assurent également un diagnostic rapide et efficace. services thérapeutiques et administratifs, Par conséquent, les ordinateurs sont des composants actifs et importants dans le réseau de télécommunications et la conception de leur interaction est une considération importante dans tout système de télémédecine [10].

Le rôle principal des ordinateurs en télémédecine est l'utilisation du dossier médical électronique (DME), la planification des patients, le stockage et la transmission des données. Interface avec d'autres dispositifs médicaux et acquisition de signaux physiologiques tels que l'ECG pour leur affichage et analyse. De plus, les ordinateurs trouvent des applications en télémédecine pour le courrier électronique, le numérique Messagerie. Applications multimédias, téléenseignement et vidéoconférence.

Dans l'hôpital où le système de télémédecine est conservé et exploiter à un endroit fixe, le PC de bureau offre la flexibilité nécessaire en termes d'ajout de nouveaux dispositifs cliniques et d'interface de communication, d'extension de mémoire et d'ajout de nouveaux périphériques, des périphériques médicaux compatibles avec les ordinateurs sont constamment développés et évalué pour répondre aux besoins de la télémédecine. Les ordinateurs de bureau modernes

sont également conçus pour être modulaires, de sorte que leurs sous-systèmes peuvent facilement être remplacés ou mis à niveau, chaque fois qu'un tel besoin se fait sentir.

la connectivité réseau est également utilisée pour connecter et faciliter l'échange de données entre divers systèmes de télémédecine au sein d'un hôpital (LAN), entre divers hôpitaux d'une ville (Métropolitain Area Network(MAN)) et sur de grandes surfaces (Réseau étendu (WAN)).

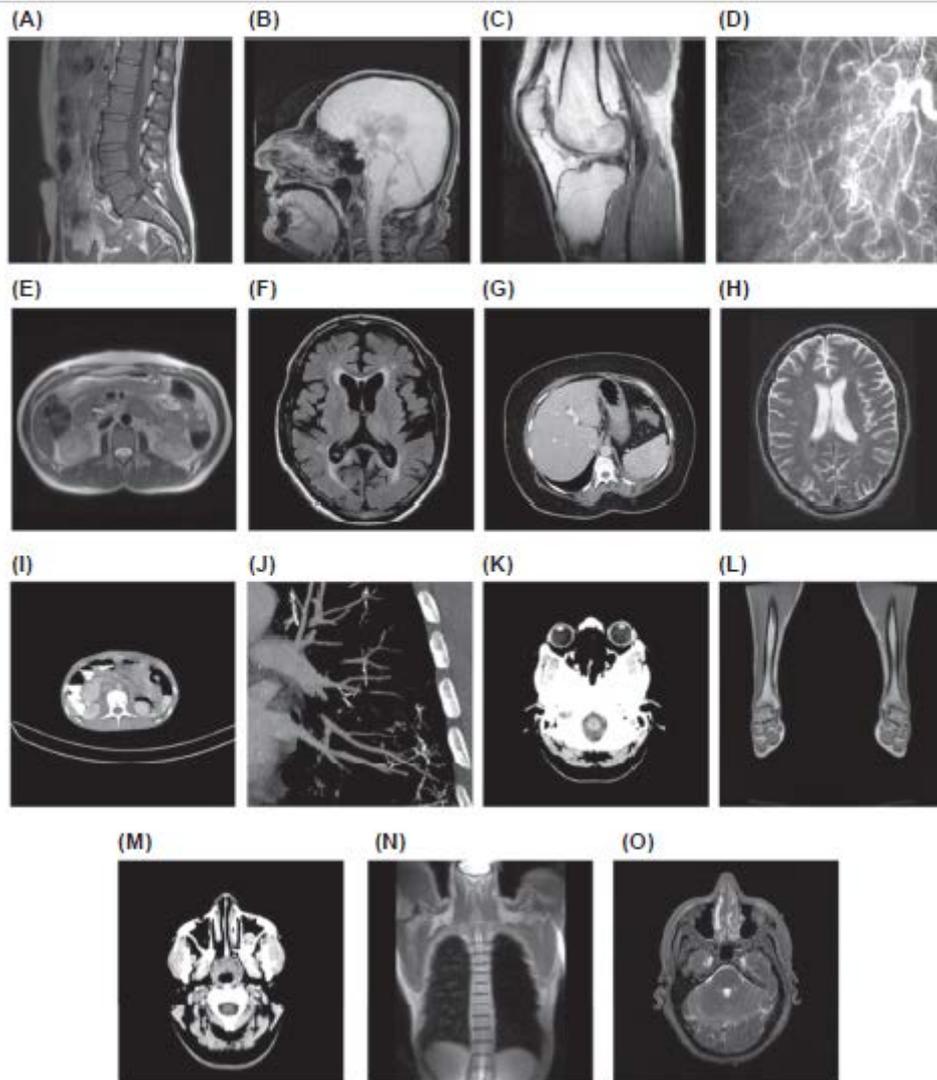


Figure 2. 3 Exemples d'images IRM utilisées pour l'analyse.

Les caractéristiques essentielles d'un système de télémédecine cliniquement utile comprennent la programmabilité, flexibilité et évolutivité. Il devrait être possible d'interfacer divers équipements de modalité d'imagerie médicale tels que les images radiographiques, tomodensitométriques et IRM pour la télé-cardiologie, une machine à ultrasons pour l'obstétrique, un otoscope pour la télé-audiologie, un ophtalmoscope pour la télé-ophtalmologie, un appareil photo numérique haute résolution pour la dermatologie et un

appareil ECG, une échocardiographie et un stéthoscope électronique pour la télé-cardiologie. Le système doit fournir une compression adéquate de la vidéo, du son et des images pour aider dans des applications comme la vidéoconférence et les téléconsultations. Un système haut performances est nécessaire pour gérer les grandes quantités de données entrantes émanant d'images hautes résolution ou de vidéos animées. De plus, le Système doit pouvoir utiliser au maximum tous les types de supports de transmission (lignes fixes, liaisons par satellite) par son adaptation à une large gamme de bandes passantes qui peut varier de 56 Kbps à plus de 45 Mbps. Ce type d'exigences est généralement satisfait par une station de travail [10].

Les stations de travail ont généralement des performances beaucoup plus élevées que celles que l'on trouve normalement sur un ordinateur à usage général. Les paramètres distinctifs concernent en particulier une capacité graphique améliorée, une puissance de traitement plus élevée, une capacité de mémoire plus élevée et capacité multitâche. Les postes de travail sont spécialement conçus pour le traitement d'images dans le domaine de la gestion des images que ce soit en téléradiologie, télépathologie ou télédermatologie. Ils ont un moniteur haute résolution conçu pour afficher et traiter les procédures graphiques basées sur des informations numériques, la principale installation incluse dans les stations de travail graphiques est le formatage d'image, Affichage d'image statique et dynamique, régulation des états de contraste et de saturation, Amplification d'image, filtres d'image, Contour d'image et définition des zones d'intérêt.



Figure 2. 4 Station de travail mobile

2.3.2 Configurations de réseautage informatique

La plupart des utilisateurs d'ordinateurs ont fini par reconnaître les avantages de la mise en réseau à la fois comme solution aux problèmes de croissance et d'expansion du système et comme méthode flexible d'intégration des ressources informatiques. Voici les configurations couramment utilisées pour les réseaux informatiques:

2.3.2.1 Informatique client-serveur

L'informatique client-serveur est un arrangement efficace dans les applications de télémédecine car elle réduit le trafic réseau et par conséquent le temps de réponse. Ceci présente un avantage significatif car les images reçues aident rapidement à une meilleure navigation et manipulation des images en temps réel, ce qui permet d'effectuer des tâches de diagnostic plus efficacement et plus efficacement.

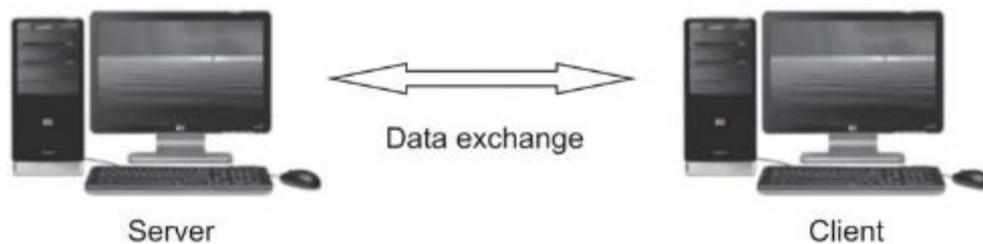


Figure 2. 5 Système client-serveur typique.

La figure 2-5 montre un système client-serveur typique, dont une application typique est l'échange d'images et de données médicales entre deux sites, le système comprend un serveur qui envoie les informations et un ordinateur client qui reçoit les informations. Côté serveur, l'adresse client et le fichier à envoyer sont sélectionnés. Après sélection de tous les paramètres, le serveur est initialisé.

Il attend ensuite qu'un client lui soit connecté. De même, les paramètres du Client sont sélectionnés de la même manière, les fichiers reçus destination et le serveur auquel le client est associé. Une fois le serveur et le client initialisé, les fichiers image peuvent être envoyés via la connexion.

Ils sont également envoyés au poste de l'expert pour interprétation et simultanément au poste du patient d'où la demande a été reçue. De cette façon, nous pouvons éviter d'avoir une archive indépendamment sur le scanner. IRM, échographie et centres de médecine nucléaire, car les enregistrements souhaités sont facilement accessibles via Ethernet

2.3.2.2 Le Cloud computing

est une expression imagée désignant un ensemble de technologies matérielles et logicielles qui offrent à un utilisateur ou à une entreprise le moyen d'accéder en libre-service, n'importe quand et n'importe où, à des fichiers personnels, des applications logicielles opérationnelles ou toute autre ressource numérique au travers d'une infrastructure réseau fiable et sécurisée. Traduit en français par informatique dans les nuages, informatique dématérialisée ou encore informatique virtuelle, ce concept rend possible l'externalisation de la puissance de calcul et de stockage. Il permet de déporter, sur des serveurs distants, des traitements informatiques traditionnellement exécutés sur la machine locale de l'utilisateur. L'intérêt d'une telle approche est double : la machine client de l'utilisateur devient un simple terminal peu gourmand en ressources, et la gérance informatique n'est plus assurée localement sur chaque poste mais de manière globale et unifiée par un prestataire tiers. L'utilisateur a virtuellement les avantages d'un accès garanti à une capacité de calcul et à un espace de stockage modulables (voire infinis), sans les inconvénients d'installation, de maintenance logicielle, de sécurisation et de sauvegarde de données [11].



Figure 2. 6 Cloud computing en téléradiologie.

2.3.3 Logiciel de télémédecine

Le logiciel de télémédecine devrait avoir la capacité d'exécuter les fonctions essentielles suivantes:

- (1) Acquisition d'informations sur le patient à partir de diverses sources.
- (2) Stockage des informations patient sur un support approprié.
- (3) Affichage des informations patient dans un format spécifié.
- (4) Transmission des informations intégrées sur un réseau de télécommunications.
- (5) Prise de rendez-vous avec des médecins spécialistes.

De plus, le logiciel devrait avoir les fonctionnalités importantes suivantes pour prendre en charge plusieurs formats et faciliter la capacité d'interfaçage:

- (1) Capture d'images et de documents à partir d'un scanner
- (2) Capture de données vidéo et autres à partir de la sortie de l'équipement médical et le convertir au format DICOM,
- (3) Conversion des informations de DICOM vers des formats non DICOM, selon les besoins.

Cependant, les spécificités de chacune des fonctions susmentionnées dépendraient de la équipement à interface et les applications de télémédecine prévues [12].

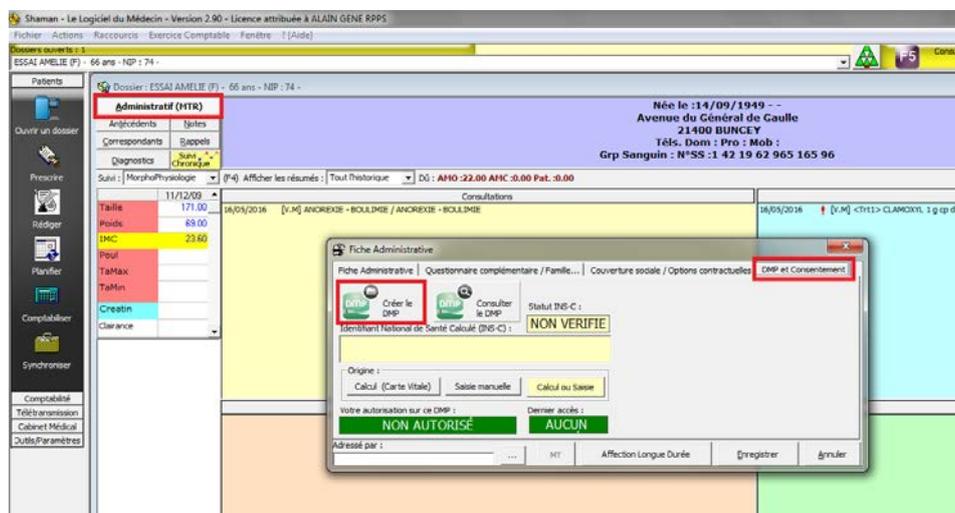


Figure 2. 7 : l'interface d'un DMP.

2.3.4 Interface des dispositifs médicaux avec les ordinateurs

Pour la transmission en temps réel et les données collectées à partir de divers appareils médicaux à des fins de télémédecine, il est nécessaire de connecter l'appareil à un ordinateur pour lequel une variété de possibilités d'interfaçage sont disponibles. Le but d'une interface est d'accepter les données comme présenté par l'appareil et transmettre ces données à l'ordinateur sous une forme qu'il peut traiter.

La configuration de l'interface dépend:

- Caractéristiques des données à collecter, qui à leur tour dépendront des besoins de l'étude.
- Capacités de l'ordinateur, il peut accepter les débits de données et effectuer le traitement en temps réel requis.

L'interface entre le dispositif d'acquisition de données et l'ordinateur se compose de matériel et de logiciels résidant dans l'ordinateur ou sous la forme d'intelligence intégrée dans le

dispositif lui-même. Les proportions de fonctions d'interface fournies par le matériel et les logiciels dépendent du type d'interface.

2.3.5 Types d'interfaces

Il existe quatre types courants d'interfaces pour obtenir des informations d'un périphérique vers un ordinateur:

- 1) **Interfaces analogiques** : Les instruments dépourvus de sortie numérique nécessitent une interface analogique pour convertir le signal de sortie analogique au format numérique avant de le transmettre à un ordinateur numérique.
- 2) **Interface série RS 232C** : RS 232C, une technique d'interfaçage très populaire, a deux importantes limitations en termes de distance et de bande passante, ce qui limite son utilité. Circuits RS 232C sont sensibles à plus de bruit lorsque la vitesse de transmission augmente et que les distances s'allongent [13].

RS232 25 Pin

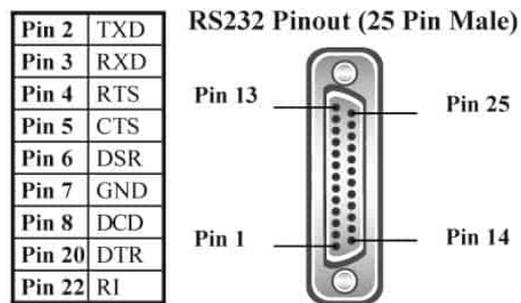


Figure 2. 8 : RS-232 pinout (25 pin).



Figure 2. 9 : Connecteur série R232.

Les lignes d'interface RS 232C sont par conséquent généralement limitées à des distances plus courtes, entre l'appareil et l'ordinateur et cela pourrait être de seulement 20 m pour certains appareils, donc il est généralement considéré comme approprié pour des distances

allant jusqu'à environ 50 m. La bande passante est également limitée à un maximum de 20 Kbps.

Pin	Connecteur DB 25	Connecteur DB 9	Fonction
TXD	2	3	Transmitted Data
RXD	3	2	Received Data
RQS ou RTS	4	7	Request To Send
CTS	5	8	Clear To Send
DSR	6	6	Data Set Ready
SG	7	5	Masse du Signal
DTR	20	4	Data Terminal Ready
CD	8	1	Carrier Detect
RI	22	9	-

Tableau 2. 1 : Brochage RS 232 [13].

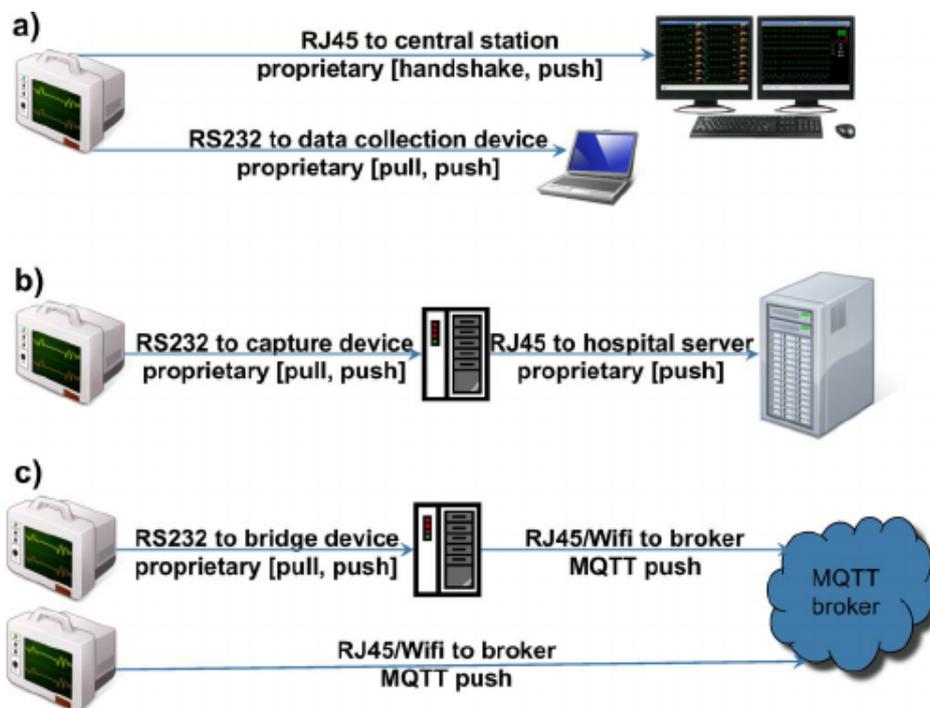


Figure 2. 10 :Exportation de données à partir d'appareils médicaux dans le passé, le présent et l'avenir [14].

La partie a (en haut) montre l'extraction de données en série à des fins de recherche ou de connexion réseau à une station de surveillance centrale. La partie b (milieu) montre l'utilisation d'un dispositif d'intégration de données poussant les données dans l'entrepôt de données électronique de l'hôpital, tandis que la partie c (bas) montre le flux d'informations à l'aide des messages push MQTT pour les appareils hérités. **N.B** (MQTT est un protocole de

messaging publish-subscribe basé sur le protocole TCP/IP. Il est conçu pour les connexions avec des sites distants où la bande passante du réseau est limitée) [14].

3) Interface standard IEEE-488: communément appelé bus d'interface à usage général (GPIB), il définit l'interface entre un périphérique et un ordinateur dans un système d'acquisition de données numériques, il spécifie essentiellement le bus à 16 lignes (câble physique) qui relie le périphérique et l'ordinateur [13].

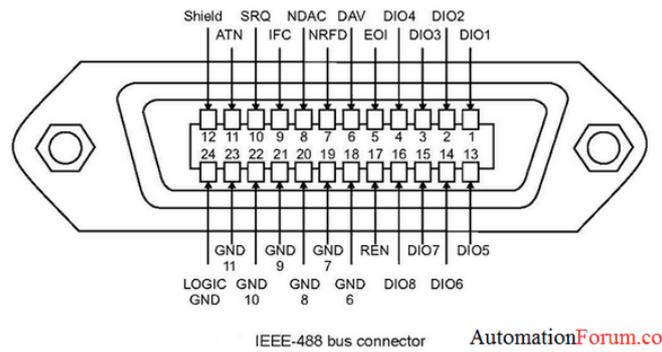


Figure 2. 11 : IEEE-488 Connecteurs.

n°	Lignes	Désignation	Commentaires
1	DIO1	Data In Out 1	Donnée 1
2	DIO2	Data In Out 2	Donnée 2
3	DIO3	Data In Out 3	Donnée 3
4	DIO4	Data In Out 4	Donnée 4
13	DIO5	Data In Out 5	Donnée 5
14	DIO6	Data In Out 6	Donnée 6
15	DIO7	Data In Out 7	Donnée 7
16	DIO8	Data In Out 8	Donnée 8
11	ATN	AtTeNtion	1 : tous les appareils attendent une commande
6	DAV	Data Valid	0 : validation des données par le transmetteur
5	EOI	End Or Identify	1 : fin de transmission
9	IFC	InterFace Clear	1 : initialisation de tous les appareils
8	NDAC	Not Data ACcept	0 : donnée non acceptée
7	NRFD	Not Ready For Data	0 : non prêt à recevoir des données
17	REN	Remote ENable	1 : appareil sous contrôle du bus
10	SQR	Service ReQuest	1 : demande de service

Tableau 2. 2 : Brochage IEEE488 24 points.

Le bus IEEE 488 comporte 16 lignes de type TTL standard:

- 8 lignes de données : DIO1 à DIO8,
- 8 lignes de contrôle : ATN, DAV, EOI, IFC, NDAC, NRFD, REN, SQR.



Figure 2. 12 : IEEE-488.

4) Universal Serial Bus (USB) : L'USB, définit une norme se rapportant à un bus informatique. Il s'agit d'un dispositif de transmission de données qui permet de connecter des périphériques informatiques à un ordinateur ou à tout autre appareil équipé d'un port USB. Depuis son apparition à la fin des années 1990, l'USB a connu différentes évolutions, allant de l'USB 1.0 à l'USB 2.0 puis USB 3.0, en proposant simultanément un débit de 1,5 Mbit/s à 10 Gbit/s.

On distingue aujourd'hui différents types de connecteurs : de type A, de type B, et de type C. Les deux premiers sont constitués de quatre broches. Les deux premières broches sont utilisées pour le transfert des données, la troisième pour jouer le rôle de la masse et la quatrième pour transmettre l'énergie électrique [13].

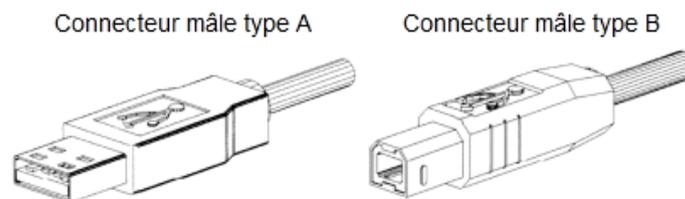


Figure 2. 13 : USB type A et B.

2.4 Normes d'échange de données médicales

Les hôpitaux, médecins et autres centres de santé du monde entier échangent quotidiennement de grandes quantités d'informations sur les soins de santé. Par conséquent, ils doivent avoir la capacité d'envoyer et de recevoir des données de soins de santé de manière efficace. Les données peuvent inclure des informations sur le patient, informations cliniques recherche et données administratives. Les données médicales sont souvent extrêmement compliquées en raison de la terminologie clinique complexe, ainsi que de la complexité structurelle dans la préparation des informations présentées. Ainsi, ces informations doivent être présentées dans

un format standardisé afin de garantir que les données sont universellement organisées, acceptées et comprises. Pour ce faire, il est nécessaire que toutes les informations de santé soient envoyées dans un langage de santé standardisé [15].

Certaines des normes populaires sont:

HL7: Les messages HL7 sont conçus pour permettre l'échange de données entre l'hôpital et les systèmes de dossiers médicaux, tandis que les documents HL7 Clinical Document Architecture (CDA) sont destinés à communiquer des documents qui peuvent être des notes de médecin et d'autres documents connexes [15].

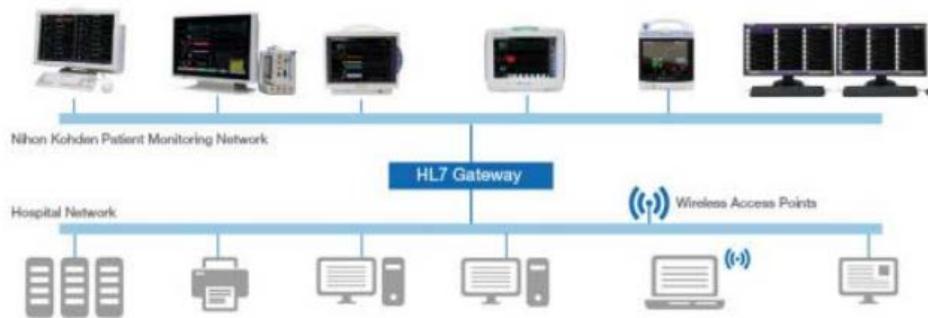


Figure 2. 14 : Le serveur HL7 Gateway.

DICOM: une norme largement adoptée pour représenter et communiquer la radiologie images et rapports, développés par l'American College of Radiology et la National Electrical Manufacturer's Association.



Figure 2. 15 : DICOM logo.

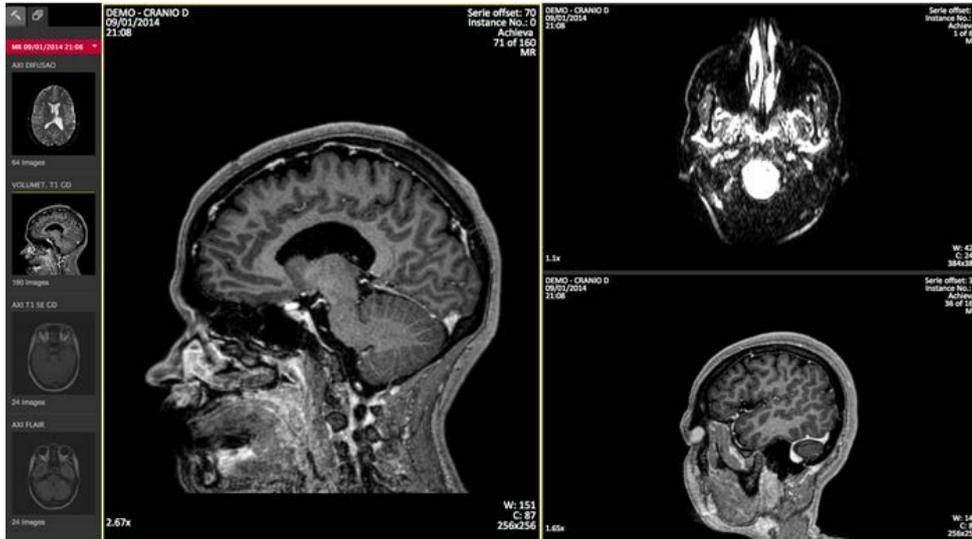


Figure 2. 16 : Image médicale en format DICOM.

ISO: le groupe de travail ISO TC215 a formulé le DMP. Il a également produit un document de spécification technique ISO 18308 qui décrit les exigences pour les architectures de DMP.

P1073: Norme **MIB** (Medical Interface Bus) développée par l'IEEE pour permettre aux hôpitaux et autres prestataires de soins de santé d'interfacer l'équipement médical aux systèmes informatiques.

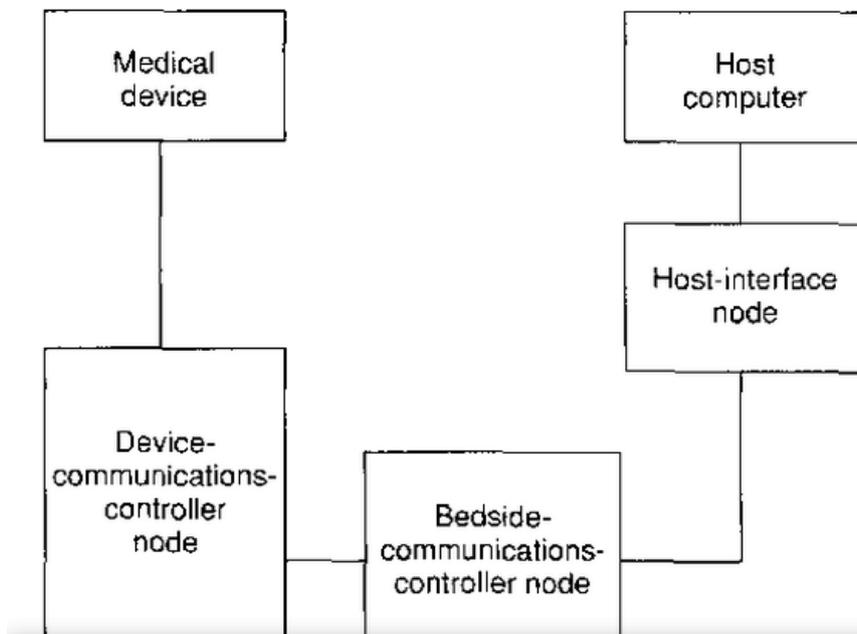


Figure 2. 17 : Les nœuds du réseau MIB.

E 1381/1394: Il s'agit d'une norme d'échange de données de laboratoire entre ordinateurs et instruments, développée par le sous-comité ASTM FM .14 [15].

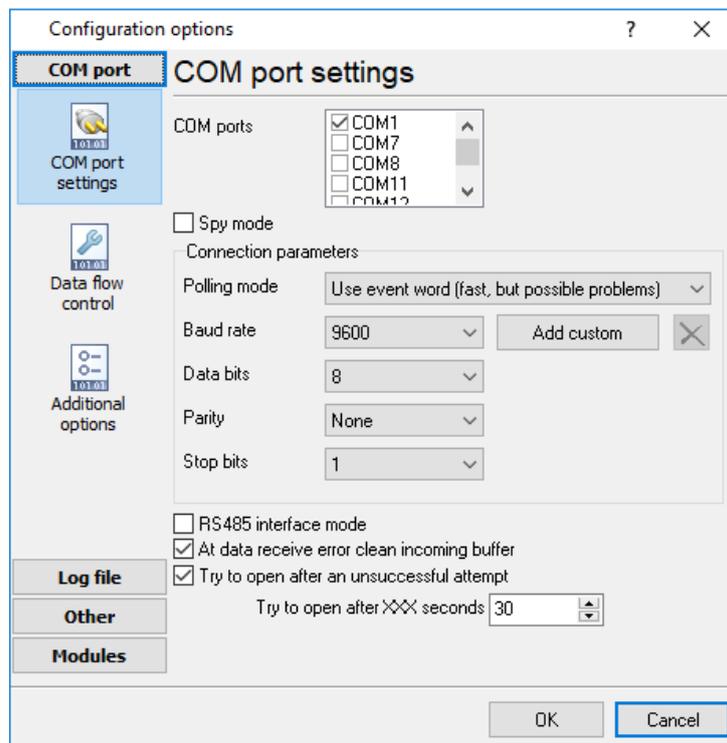


Figure 2. 18 : Configuration du port COM avec le protocole ASTM (E1381ou E1394).

NCPDP: Norme d'échange d'informations pharmaceutiques, élaborée par le Conseil national des pharmacies de médicaments sur ordonnance.

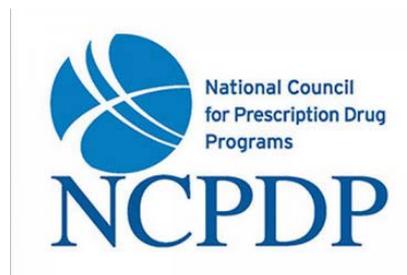


Figure 2. 19 : NCPDP logo.

ANSI X12 : Il s'agit d'un ensemble de protocoles de transaction qui peuvent être utilisés pour transmettre pratiquement tous les aspects des données des patients. Il est particulièrement populaire aux États-Unis pour communiquer des informations liées à la facturation, en raison des exigences de la Healthcare Insurance Portability and Accountability Act (HIPAA) pour la transmission de données.

CEN / EN13606: Il s'agit d'une norme européenne pour la communication d'informations à partir des systèmes de DSE et pour la communication inter-systèmes dans un environnement d'information clinique [15].

2.5 Conclusion

Dans ce chapitre, on a abordé certains aspects de télécommunications en point de vue biomédical. On s'est concentré sur le côté hard, interfaçages, les normes standards trouvées dans les instruments médicaux qui sont très importants dans la conception de la télémédecine.

CHAPITRE 3

CHAPITRE 3 : La conception de la plate-forme biomédicale

GBM TeleMed basée sur WebRTC

3.1 Introduction

Aujourd'hui, la majorité des solutions télémédecine existantes utilisent des couches de transport et des protocoles spécialisés (comme par exemple les outils vendus par CISCO équipant des salles de conférences privées pour grandes entreprises), mais leur faible interopérabilité et la complexité de l'infrastructure matérielle nécessaire sont un gros handicap. La plupart utilisent des solutions centralisées (la vidéo, l'audio et les datas passent par des "relais") qui ne "scalent pas", ou bien à des prix très élevés.

On trouve également des solutions grand public, comme celles intégrées aux outils de type « Messenger » (Skype ou Google hangouts), basées sur http en utilisant le protocole d'encodage H.264. Actuellement les outils les plus populaires de ce type (Skype, Hangouts, Face Time) sont basés sur des solutions paires à pair (P2P) mais nécessitent l'installation de logiciels spécialisés, parfois non disponibles sur tous les systèmes d'exploitation, et non interopérables. Un rapide survol des différentes solutions actuelles de web-conférence, grand public comme professionnelles, permet de les classer selon 2 points de vue:

- D'un point de vue applicatif:
 - soit des **applications natives** (ex. Skype, Vibrer, Facetime, Xmeeting),
 - soit des **applications web** (ex. Adobe Connect, Google Hangouts).

- D'un point de vue technologique:
 - soit base sur des **technologies propriétaires** (Flash, Silverlight...),
 - soit base sur des **technologies open-source** (Web Socket, WebRTC...).

On propose une plate-forme web de télémédecine basée sur WebRTC Application Programming Interface (API) pour transmettre la vidéo et la voix en temps réel sur Internet, via un navigateur Web. Outre le contrôle du microphone et de la webcam, La solution proposée utilise exclusivement des composants logiciels open source et nécessite uniquement un navigateur Web compatible WebRTC, comme Google Chrome ou Firefox. On a réalisé une communication bidirectionnelle patient-médecin.

3.2 Adressage dans le réseau IP

3.2.1 Adresse IP

Une adresse IP (avec IP pour Internet Protocol) est le numéro qui identifie chaque ordinateur connecté à Internet, ou plus généralement et précisément, l'interface avec le réseau de tout matériel informatique (routeur, imprimante) connecté à un réseau informatique utilisant l'Internet Protocol

3.2.1.1 Classes d'adresse

Chaque machine connectée à un réseau IP a une adresse IP représentée par quatre chiffres séparés par des points. Chaque chiffre est équivalent à un nombre binaire de 8 bits. D'où l'adressage IPv4 est de 32 bits. Une adresse IP est constituée de deux parties : un identificateur de réseau et un identificateur de la machine pour ce réseau. Il existe 4 classes d'adresses, chacune permettant de coder un nombre différent de réseaux et de machines [16].

Le format général est défini par la figure 3-3 :

Class ID	adresse Réseau	adresse Machine
-----------------	-----------------------	------------------------

Figure 3. 1 : Format adresse IPv4.

Classe A

Les adresses de classe A s'étendent de 1.0.0.1 à 126.255.255.254. Elles permettent d'adresser 126 réseaux ($2^7 - 2$) et plus de 16 millions de machines ($2^{24} - 2$, soit 16 777 214) par réseau mais les valeurs d'adresse réseau 0 et 127 sont réservées.

0	7 bits	24 bits
----------	---------------	----------------

Figure 3. 2 : Format IPv4 classe A.

Classe B

Les adresses de classe B vont de 128.0.0.1 à 191.255.255.254, ce qui correspond à plus de 16 384 réseaux de 65 533 machines (14 bits pour les réseaux et 16 pour les machines). Cette classe est la plus utilisée et les adresses sont aujourd'hui pratiquement épuisées [16].

10	14 bits	16 bits
-----------	----------------	----------------

Figure 3. 3 : Format IPv4 classe B.

Classe C

La classe C couvre les adresses 192.0.0.1 à 223.255.255.254, elle adresse plus de 2 millions de réseaux (2 097 152) de 254 machines (21 bits pour le réseau et 8 pour les machines).

110	21 bits	8 bits
------------	----------------	---------------

Figure 3. 4 : Format IPv4 classe C.

Classe D

Les adresses de la classe D sont utilisées pour la diffusion (multicast) vers les machines d'un même groupe. Elles vont de 224.0.0.0 à 239.255.255.255 (28 bits pour les machines appartenant à un même groupe). Ce groupe peut être un ensemble de machines mais aussi un ensemble de routeurs (diffusion des tables de routage). Certains systèmes ne supportent pas les adresses de multicast.

Adresse de boucle locale

La machine elle-même ou machine locale peut être auto-adressée avec une adresse de la forme 127. x. x. x, cette adresse dite de boucle locale (*localhost*) est utilisée lors de tests de la machine ou de programmes applicatifs. Tout datagramme émis à destination d'une adresse 127. x. x .x est directement recopié du tampon d'émission vers le tampon de réception, il n'est jamais émis sur le réseau, ce qui protège ce dernier d'éventuels dysfonctionnements du nouvel applicatif [17].

Adresse de diffusion

L'adresse 255.255.255.255 est utilisée pour envoyer un message à toutes les machines du même segment de réseau. La diffusion est limitée aux seules machines de ce segment, le datagramme n'est pas relayé sur d'autres réseaux. L'adresse 255.255.255.255 est dite adresse de diffusion générale ou limitée car la diffusion ne franchit jamais un routeur [17].

3.2.1.2 Masque de réseau

Pour une adresse IP donnée, un masque est utilisé pour dissocier les champs d'adresse réseau de l'adresse machine d'une adresse IP. Il est formé par une succession de 1 à gauche suivi d'une succession de 0. Pour chaque classe, il existe un masque par défaut :

- Classe A : 255.0.0.0
- Classe B : 255.255.0.0
- Classe C : 255.255.255.0

Pour subdiviser un grand réseau en des sous-réseaux comportant des nombres réduits de machines, on peut fabriquer un masque de sous-réseau ou masque personnalisé. Pour ce faire,

on prend le masque par défaut puis on emprunte des bits à la partie adresse machine, et l'on met ces bits à 1. Il faut noter que :

- Deux bits au moins doivent rester pour adresser les machines.
- On emprunte au moins deux bits.
- Par exemple, 255.255.255.224 est un masque de sous-réseau de classe C construit en empruntant trois bits.

3.2.2 Adresse privée et adresse publique

L'organisme IANA offre un plan d'attribution d'adresses pour les réseaux connectés à Internet (réseau public). Or, tous les réseaux n'ont pas nécessairement un besoin d'interconnexion via un réseau public, dans ce cas l'unicité d'adresse au plan mondial est inutile.

Certaines entreprises disposent de leur propre réseau (réseau privé) et n'ont aucun besoin d'interconnexion vers l'extérieur, il est alors possible d'utiliser n'importe quelle adresse IP. Toutefois, afin d'éviter l'anarchie dans l'utilisation des adresses, l'IANA a défini dans la RFC 1918 des plages d'adresses réservées pour ces réseaux privés. Ces adresses sont dites privées et donc non routables sur Internet. Le tableau suivant indique ces plages d'adresses [18]:

Classe d'adresses	Plages d'adresses privées	Masque réseau	Nombre de machines adressables	Nombre de réseaux adressables
A	10.0.0.0 à 10.255.255.255	255.0.0.0	Sur 24 bits, soit 16777216 machines	1
B	172.16.0.0 à 172.31.255.255	255.240.0.0	Sur 20 bits, soit 1048576 machines	16
C	192.168.0.0 à 192.168.255.255	255.255.0.0	Sur 16 bits, soit 65536 machines	256

Tableau 3. 1 : Plage d'adresse privée

Dès que ces réseaux privés ont des besoins de se connecter à un réseau public comme Internet, il faut convertir ces adresses privées en des adresses publiques. Pour ce faire on doit renuméroter tous les terminaux avec des adresses publiques ou bien réaliser une translation d'adresses au moyen d'un translateur d'adresse NAT.

Le processus de NAT fait intervenir une entité entre un terminal, ayant une adresse IP privée, et tout autre ayant une adresse IP publique. Ce mécanisme consiste à insérer un boîtier, appelé passerelle NAT, entre le réseau Internet et le réseau privé (LAN, intranet d'une entreprise,...).

Ce boîtier se charge de la translation des adresses IP privées en des adresses IP publiques, et effectue aussi l'opération inverse [17].

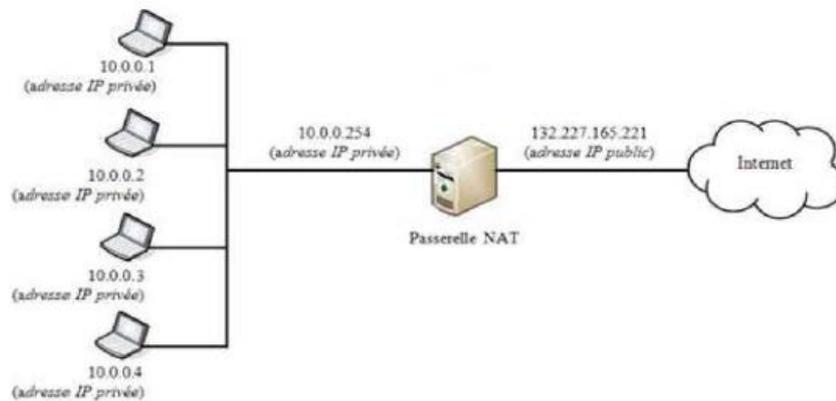


Figure 3. 5 : Translation d'adresse.

3.2.3 Notion de port

La notion de port est utilisée par la couche hôte à hôte pour autoriser plusieurs programmes à établir une connexion simultanée et de multiplexer les données reçues des différentes applications. C'est cette couche qui attribue un numéro de port à une connexion. Un numéro de port identifie une application particulière dans une machine. Les 1024 premiers ports sont réservés, ils sont dits référencés. Voici un tableau qui contient quelques exemples de numéros de ports associés à des services [18].

Nom du service	Numéro de port /Protocole	Commentaire
FTP	21/TCP	Protocole de transfert de fichiers
Telnet	23/TCP	Emulation de terminal
SMTP	25/TCP	Service de messagerie
DNS	53/UDP	Service de noms de domaine
HTTP	80/TCP	Service web
H.323	1720/TCP	Service de visioconférence sur IP
SIP	5060/UDP	Service multimédia sur IP

Tableau 3. 2 : Exemples d'association de services et de numéros de ports.

3.2.4 Le protocole TCP

TCP a été défini en Septembre 1981 par RFC 793 pour un transfert de données fiable sur le réseau IP. En effet, il ouvre une session et effectue lui-même le contrôle d'erreur de bout en bout. On dit qu'il est alors « en mode connecté »[18] :

- ouverture de connexion

- transfert de données
- fermeture de connexion

3.2.5 Le protocole UDP

C'est un protocole défini par le RFC 768 qui est un protocole de transport simple qui ne garantit pas la fiabilité des données car il n'y a aucune ouverture de session et il n'y a pas de contrôle d'erreur, en plus il n'utilise aucun acquittement et ne met en place aucun contrôle de flux. C'est pour cela que l'UDP est dit « mode non connecter » et c'est pourquoi il est rapide par rapport à TCP [18].

3.3 La technologie WebRTC

La communication en temps réel sur le Web (WebRTC) est une nouvelle norme et un effort de l'industrie qui étend le modèle de navigation sur le Web. Pour la première fois, les navigateurs peuvent échanger directement des médias en temps réel avec d'autres navigateurs de manière poste à poste.

Le World Wide Web Consortium (W3C) et l'Internet Engineering Task Force (IETF) définissent conjointement les API JavaScript (Application Programming Interfaces), les balises HTML5 standard et les protocoles de communication sous-jacents pour la configuration et la gestion d'un canal de communication fiable entre n'importe quelle paire de navigateurs Web de nouvelle génération [19].

L'objectif de normalisation est de définir une API WebRTC qui permet à une application Web de s'exécuter sur n'importe quel appareil, grâce à un accès sécurisé aux périphériques d'entrée (tels que les webcams et les microphones).



Figure 3. 6 : WebRTC logo.

En 2011, à l'initiative de Google qui ouvre les sources des technologies de GIPS, des groupes De travail au W3C pour la standardisation d'une solution de RTC sur le Web sont créés. Rapidement, d'autres grands acteurs comme Mozilla, Voxeo, Ericsson, Intel, Cisco, Samsung,

France Telecom, AT&T, Avaya, Huawei et Opera, participent à la naissance du standard WebRTC. Depuis, et par itérations successives, l'écriture du standard avance. Il est publié sur le site du W3C et du code de référence est disponible en open source.

Low entry barriers

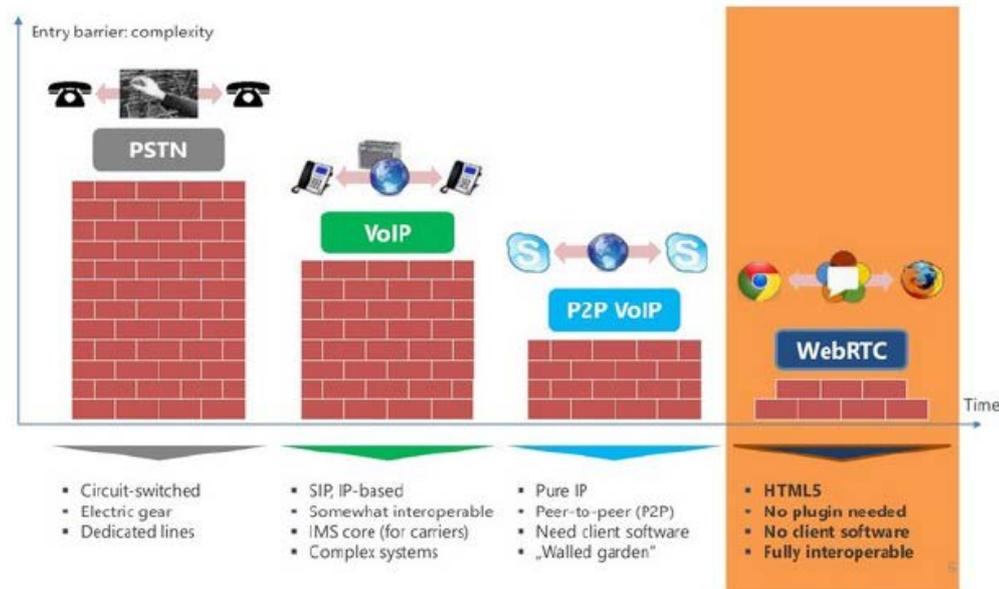


Figure 3. 7 : Historique des solutions de voix et vidéo sur IP.

3.3.1 Architecture Web

La sémantique classique de l'architecture Web est basée sur un paradigme client-serveur, dans lequel les navigateurs envoient une demande de contenu HTTP (Hypertext Transfer Protocol) au serveur Web, qui répond par une réponse contenant les informations demandées. Les ressources fournies par un serveur sont étroitement associées à une entité connue par un URI (Uniform Resource Identifier) ou une URL (Uniform Resource Locator). Dans le scénario d'application Web, le serveur peut incorporer du code JavaScript dans le code HTML page qu'il renvoie au client. Un tel code peut interagir avec les navigateurs via des API JavaScript standard et avec les utilisateurs via l'interface utilisateur [20].

3.3.2 Architecture WebRTC

WebRTC étend la sémantique client-serveur en introduisant une communication peer-to-peer paradigme entre les navigateurs. Le modèle architectural WebRTC le plus général (voir La figure 3-8) tire son inspiration du soi-disant SIP (Session Initiation Protocol) Trapèze (RFC3261).

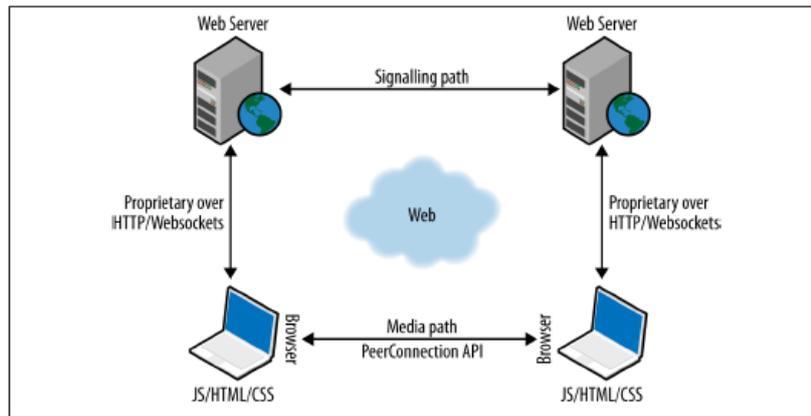


Figure 3. 8 : Le trapèze WebRTC.

Dans le modèle WebRTC Trapèze, les deux navigateurs exécutent une application Web, qui est téléchargée à partir d'un autre serveur Web. Les messages de signalisation sont utilisés pour établir et mettre fin aux communications. Ils sont transportés par le protocole HTTP ou WebSocket via des serveurs Web qui peuvent les modifier, les traduire ou les gérer selon les besoins.

Il convient de noter que la signalisation entre le navigateur et le serveur n'est pas normalisée dans WebRTC, car elle est considérée comme faisant partie de l'application. En ce qui concerne le chemin de données, une PeerConnection permet aux médias de circuler directement entre les navigateurs sans aucun serveur intermédiaire. Les deux serveurs Web peuvent communiquer à l'aide d'un protocole de signalisation standard telle que SIP ou Jingle. Sinon ils peuvent utiliser un protocole de signalisation propriétaire. Le scénario WebRTC le plus courant est probablement celui où les deux navigateurs exécutent la même application Web, téléchargée à partir de la même page Web. Dans ce cas, le trapèze devient un triangle (voir figure 3-9) [20].

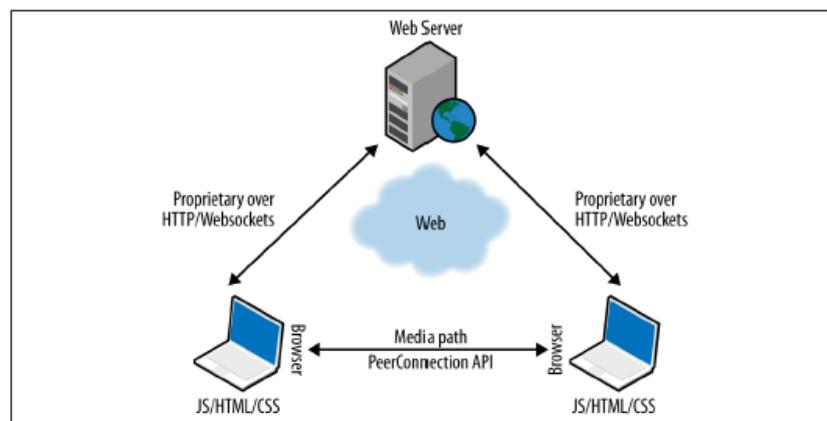


Figure 3. 9 : Le triangle WebRTC.

3.3.3 WebRTC dans 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 3-10). 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). 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...

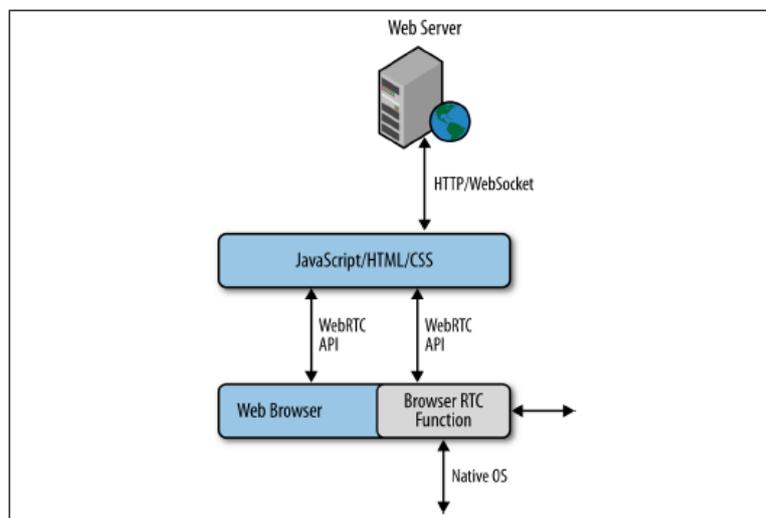


Figure 3. 10 : Communication en temps réel dans le navigateur.

3.3.4 Prise en charge du navigateur

Bien que l'objectif de WebRTC soit d'être omniprésent pour chaque utilisateur, cela ne signifie pas que chaque navigateur possède les mêmes fonctionnalités au même temps.

Différents navigateurs peuvent choisir d'être en avance sur la courbe dans certaines zones, ce qui fait que certaines choses fonctionnent dans un navigateur et pas dans un autre. La prise en charge actuelle de WebRTC dans l'espace du navigateur est illustrée dans la section suivante [20].

a) Compatibilité avec Chrome, Firefox et Opera

Il est possible que le navigateur que vous utilisez actuellement prenne en charge WebRTC. Chrome, Firefox et Opera prennent tous en charge WebRTC prêt à l'emploi. Cela devrait également fonctionner sur tous les systèmes d'exploitation courants, tels que Windows, Mac

et Linux. Les fournisseurs de navigateurs, tels que Chrome et Firefox, ont également collaboré pour résoudre les problèmes d'interopérabilité afin qu'ils puissent tous communiquer facilement.

b) 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 (Ice Cream Sandwich). Cela est dû à la notion de partage de code entre les versions de bureau et mobile de Chrome et Firefox [19].

c) 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].

3.3.5 Le signaling

Le *signaling* est le mécanisme qui coordonne la communication et permet d'envoyer et de recevoir des messages de contrôle. Si WebRTC ne définit pas les méthodes et protocoles de signalisation et impose de passer par un serveur tiers, c'est pour de très bonnes raisons [21]:

1. Pour éviter la redondance et maximiser la compatibilité avec les technologies établies. En effet, différentes applications peuvent nécessiter différents protocoles et le groupe de travail WebRTC ne voulait pas se verrouiller sur un protocole qui pourrait se révéler insuffisant à couvrir toutes les utilisations possibles.
2. WebRTC s'exécute dans un navigateur et le signaling implique que celui-ci soit "*stateful*", c'est à dire qu'il enregistre les états (l'inverse de *stateless*). Or cela devient problématique si la signalisation est perdue à chaque fois que la page est rechargée...

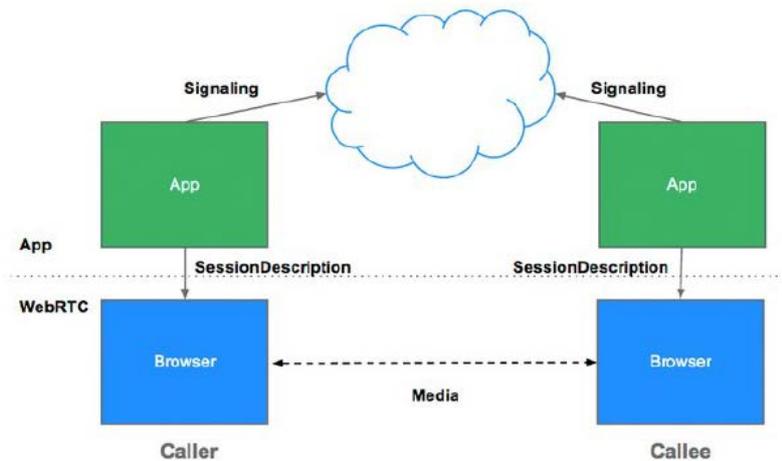


Figure 3. 11 : Architecture JSEP.

L'architecture de JSEP évite au navigateur d'avoir à enregistrer l'état, c'est à dire de fonctionner comme une machine à état de signalisation. Ainsi, plus de perte de données de signalisation à chaque rechargement de la page. Au lieu de cela, l'état de signalisation est enregistré sur un serveur tiers [21].

Le signaling sert à échanger trois types d'information:

- ✓ Les messages de contrôle de session (initialiser les communications et rapporter les erreurs)
- ✓ les configurations réseau pour le monde "extérieur", (adresse IP, ports...)
- ✓ les capacités "médias" (résolutions et codecs supportés par les devises / browsers, etc...).

3.3.6 Le Framework ICE et les protocoles STUN et TURN

Le protocole Session Traversal Utilities for NAT (STUN) permet une application hôte pour découvrir la présence d'un traducteur d'adresses réseau sur le réseau, et dans un tel cas pour obtenir l'IP public alloué et le tuple de port pour la connexion actuelle. Pour ce faire, le protocole nécessite l'assistance d'un serveur STUN tiers configuré qui doit résider sur le réseau public.

Le protocole Traversal Using Relays around NAT (TURN) permet à un hôte derrière un NAT d'obtenir une adresse IP publique et un port à partir d'un serveur relais résidant sur l'Internet public. Grâce à l'adresse de transport relayée, l'hôte peut alors recevoir des médias de n'importe quel pair qui peut envoyer des paquets à l'Internet public [21].

L'Interactive Connectivity Establishment (ICE) est une technique utilisée dans les réseaux informatiques pour trouver des moyens pour que deux ordinateurs se parlent aussi directement que possible dans les réseaux peer-to-peer.

Les applications WebRTC utilisent le Framework ICE pour surmonter les difficultés de la mise en réseau du monde réel: la plupart des dispositifs en mesure d'utiliser WebRTC vivent derrière une ou plusieurs couches de NAT et peuvent avoir des couches de sécurité qui bloquent certains ports et protocoles (parfois avec de l'inspection profonde de paquets IPs ou "Deep Paquet Inspection DPI"). De plus, beaucoup d'entre eux sont derrière des proxies et des firewalls d'entreprise, sans compter les firewalls et NAT des routeurs Wifi domestiques.

Le protocole **STUN** et son extension **TURN** sont utilisés par ICE pour traverser les NATs et se conformer aux aléas du réseau (DPI, etc.). ICE tente de connecter directement les pairs.

Il essaie toutes les possibilités en parallèle et choisit l'option la plus efficace qui fonctionne.

ICE tente d'abord d'établir une connexion directe entre les pairs, avec la plus faible latence possible, en utilisant l'adresse hôte obtenue à partir du système d'exploitation du pair et de sa carte réseau.

En cas d'échec, (NAT ou Firewall...) ICE passera par un serveur STUN pour obtenir une adresse réseau externe. Dans ce processus, les serveurs STUN ont une seule tâche: permettre à un pair derrière un NAT de découvrir son adresse publique et son port. (Google fournit des serveurs STUN publics). Si UDP échoue, ICE tentera TCP: avec HTTP, puis HTTPS. Selon **webrtcstats.com**, 85% des appels WebRTC établissent avec succès leurs connexions via des serveurs STUN [21].

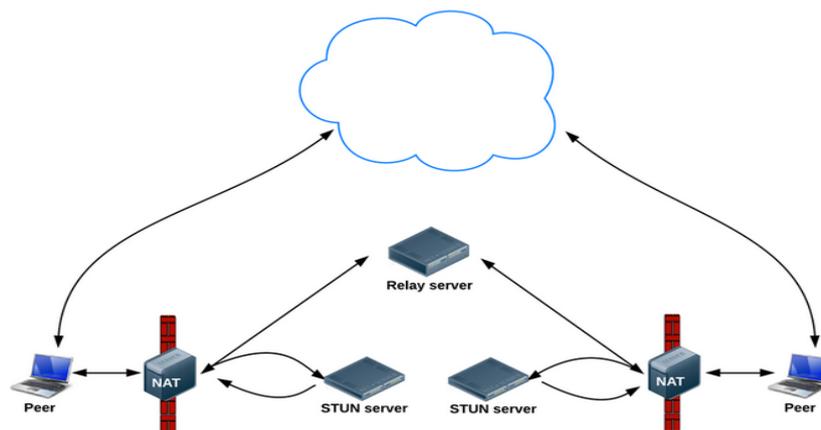


Figure 3. 12 : Utilisation de serveurs STUN pour obtenir les IP publiques.

Si la connexion directe échoue (firewall, DPI, etc...) le trafic est alors route via un serveur relais TURN. Il existe des serveurs publics (Google...). Les serveurs TURN ont un rôle simple: Relayer un flux, mais contrairement aux serveurs STUN, ils consomment de la bande passante.

3.3.7 Connexion au serveur de signalisation

Comme expliqué précédemment, la signalisation est utilisée pour le processus de découverte et de négociation des homologues WebRTC. Dans notre projet **GBM TeleMed** On utilise Librairie open source Scaledrone comme serveur de signalisation car il nous permet d'utiliser WebRTC sans faire de programmation de serveur. Cependant, si on veut écrire notre propre serveur de signalisation, ce programme fonctionnerait toujours correctement.

Scaledrone fonctionne en nous permettant de nous abonner à une salle, puis il diffuse les messages envoyés dans cette salle à tous les utilisateurs abonnés. Cela rend Scaledrone idéal pour la signalisation WebRTC [22].

Pour importer Scaledrone, on ajoute cette balise de script avant la balise de fermeture `</head>`.

```
<script type='text/javascript' src='https://cdn.scaledrone.com/scaledrone.min.js'></script>
```

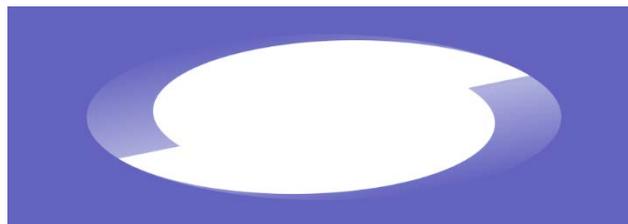


Figure 3. 13 : Scaledrone logo.

Après la connexion à Scaledrone, il faut s'abonner à une salle (le nom de la salle provient du hachage de l'URL). L'événement des membres indique qui est connecté à la salle, [25].

- Si on est le seul utilisateur dans la salle, on lance le code WebRTC et on attend une offre d'un autre utilisateur.
- Si on est le deuxième utilisateur connecté, on lance le code WebRTC et on signale une offre au premier utilisateur.
- Avec trois utilisateurs connectés ou plus, la salle est pleine.

il existe plusieurs Librairies open source comme Scaledrone qui on peut les utilises pour des projets de WebRTC.

PeerJS : propose aux développeurs un outil capable de créer des connexions pair à pair de manière simple. Présenté sous la forme d'une bibliothèque JavaScript, il fonctionne tel un wrapper pour WebRTC et autorise ainsi la création de connexions à l'aide de seulement

quelques lignes de code. PeerJS négocie des connexions avec WebRTC et autorise les connexions par identifiant de pairs, mais nécessite cependant un serveur de signaling avec Node.js.

EasyRTC : Cette bibliothèque propose d'utiliser des serveurs de signaling (STUN) et TURN publics, configurables. Elle inclut également le transfert de fichiers en pair à pair et l'établissement des communications de manière sécurisée via un système de Tokens.

SimpleWebRTC : est certainement la plus ancienne des librairies WebRTC semble avoir perdu sa popularité aujourd'hui.

rtc.io : est une autre librairie populaire, assez simple à utiliser. Utilise des serveurs STUN et TURN publics. Supporte les communications N x N mais pas le transfert de fichier pair à pair.

WebRTC tel qu'il est actuellement implémenté ne supporte que les communications 1 à 1 mais peut être utilisé dans des scénarii plus complexes, par exemple 3 utilisateurs pourraient communiquer les uns avec les autres directement de pair à pair ou en passant par une **MCU** (Multipoint Control Unit), un serveur qui peut gérer un grand nombre de participants et fait du transfert de flux, les mixe ou encore les enregistre [21].

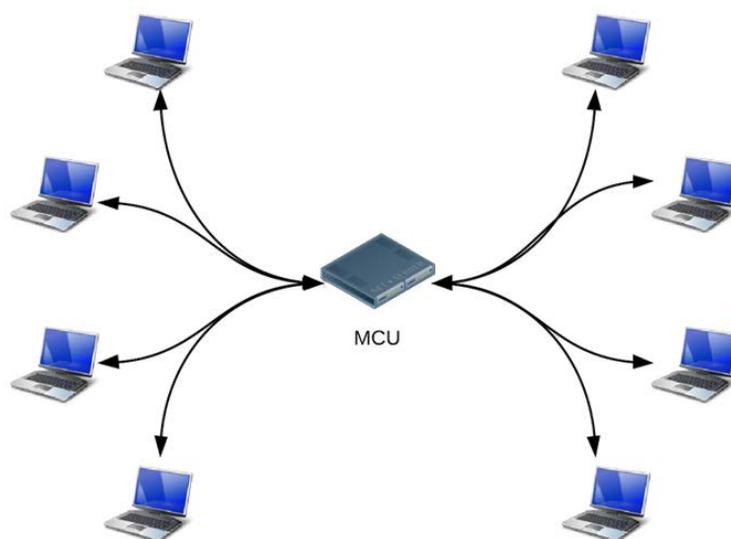


Figure 3. 14 : Exemple de Multipoint Control Unit.

3.3.8 Les API WebRTC

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 [21]:

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
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.
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.

3.3.9 Accès aux périphériques multimédias

Il y a eu une longue histoire derrière la tentative d'afficher des périphériques multimédias sur l'écran du navigateur. Beaucoup ont eu du mal avec diverses solutions basées sur Flash ou d'autres plug-ins qui vous obligeaient à télécharger et installer quelque chose dans votre navigateur pour pouvoir capturer la caméra de l'utilisateur. C'est pourquoi le W3C a décidé de créer enfin un groupe pour intégrer cette fonctionnalité dans le navigateur. Les derniers navigateurs donnent désormais à JavaScript l'accès à l'API `getUserMedia`, également appelée API `MediaStream` [19].

Commençons donc par l'appel d'API `getUserMedia ()` et son objet `MediaStream` renvoyé.

On prépare une simple page HTML avec du code JavaScript qui nous permet d'accéder aux ressources vidéo locales et de les afficher dans une balise HTML5 (vidéo).

L'exemple 2-1 montre la page très simple qui on a créée pour notre exemple.

Exemple 2-1. Notre page HTML compatible WebRTC

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />

    <title>Chapter 3:Accès aux périphériques multimédias</title>
  </head>
  <body>
    <video autoplay></video>
    <script src="getUserMedia.js"></script>
  </body>
</html>
```

L'exemple 2-1 contient une référence à un fichier JavaScript (getUserMedia.js), dont le contenu est illustré dans l'exemple 2-2.

Exemple 2-2. Le fichier getUserMedia.js

```
// Recherchez les différentes manières d'appeler getUserMedia par les
// différents fournisseurs de navigateurs()
// API method:
// Opera --> getUserMedia
// Chrome --> webkitGetUserMedia
// Firefox --> mozGetUserMedia
navigator.getUserMedia = navigator.getUserMedia || navigator.webkitGetUserMedia || navigator.mozGetUserMedia;
// Utilisez des contraintes pour demander un MediaStream uniquement vidéo:
var constraints = { audio: false, video: true };
var video = document.querySelector('video');
// Rappel à appeler en cas de succès ...
function successCallback(stream) {
  // Remarque: mettez le flux renvoyé à la disposition de la console pour inspection
  window.stream = stream;
  if (window.URL) {
    // Cas Chrome: URL.createObjectURL () convertit un MediaStream en URL blob
    video.src = window.URL.createObjectURL(stream);
  } else {
    // Firefox et Opera: le src de la vidéo peut être défini directement à partir du flux
    video.src = stream;
  }
  // Nous sommes tous prêts. Disons simplement lire la vidéo!
  video.play();
}
// Rappel à appeler en cas de panne ...
```

```
function errorCallback(error) {
    console.log('navigator.getUserMedia error: ', error);
}
//Action principale: il suffit d'appeler getUserMedia () sur l'objet
//navigateur
navigator.getUserMedia(constraints, successCallback, errorCallback);
```

Les captures d'écran suivantes montrent à quoi ressemble la page lorsque on la charge dans Chrome (Figure 3-15) ou Firefox (Figure 3-16) ou Opera (Figure 3-17)

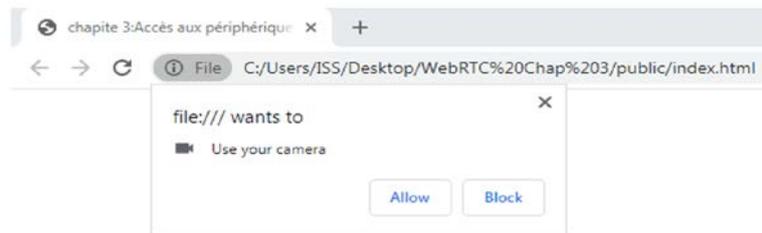


Figure 3. 15 : Ouvrir notre exemple dans Chrome.

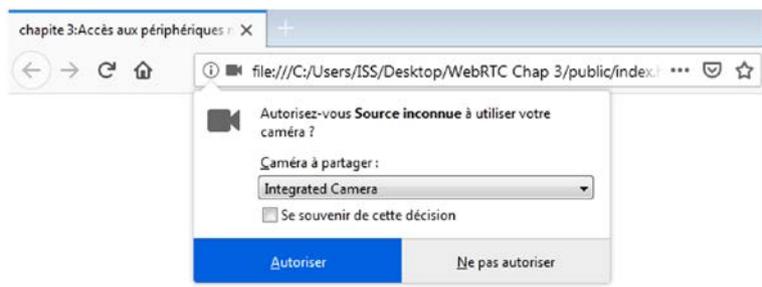


Figure 3. 16 : Ouvrir notre exemple dans Firefox.

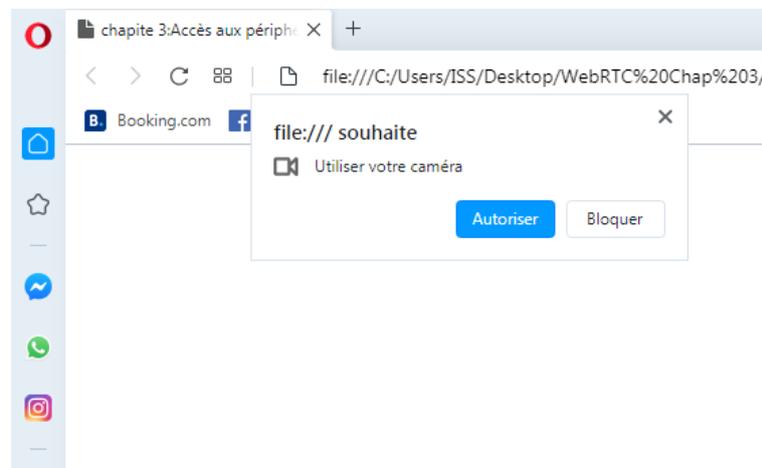


Figure 3. 17 : Ouvrir notre exemple dans Opera.

3.3.10 Sécurité

Les solutions de communication en temps réel présentent plusieurs vulnérabilités potentielles comme par exemple [21]:

- Les médias non cryptés pourraient être interceptés en route entre les navigateurs ou entre un navigateur et un serveur.

- Une application pourrait enregistrer le son et/ou la vidéo à l'insu de l'utilisateur.
- Des malwares ou des virus pourraient être installés cachés dans les plugins ou les applications.

WebRTC apporte plusieurs solutions pour se prémunir contre ces problèmes :

- l'implémentation de WebRTC utilise des couches de sécurisation comme DTLS et SRTP.
- Le cryptage est obligatoire pour tous les composants de WebRTC y compris la signalisation.
- WebRTC n'est pas un plugin : il tourne dans une sandbox du navigateur dans un processus séparé et il ne requiert aucune installation de logiciel tiers. En plus il se met à jour en même temps que le navigateur.
- Les accès à la caméra et au micro doivent être autorisés explicitement et quand ils sont actifs, c'est indiqué de manière claire dans l'interface de l'utilisateur.

3.4 Définition de HTML 5

Le HTML5 n'est pas juste le successeur du HTML 4, il est bien plus que ça ! Alors que les langages HTML 4 et autres XHTML se focalisaient juste sur le contenu des pages Web, le HTML5 se focalise sur les applications Web et l'interactivité, sans toutefois délaissier l'accessibilité et la sémantique. Le HTML5 se positionne également comme concurrent des technologies Flash et Silverlight [21].



Figure 3. 18 : Logo du HTML.

Au-delà de la rénovation syntaxique, HTML 5 introduit de nouveaux éléments inédits dotés de sens et apportant une alternative aux blocs génériques `<div>` et ``. Ces derniers se déclinent dorénavant en un large panel d'éléments sémantiques tels que `<article>`, `<section>`, `<aside>`, `<hgroup>`, `<header>`, `<footer>`, `<nav>`, `<time>`, `<mark>`, `<figure>`, et `<figcaption>`.

Tous ces éléments peuvent être mis en forme nativement via CSS sur l'ensemble des navigateurs modernes, même si leur fonction propre ne sera véritablement reconnue que sur les toutes dernières générations [21].

<header>

L'élément `<header>` représente le bloc d'en-tête d'une section ou d'une page. Il remplace avantageusement son homologue classique `<div id="header">`, mais ne doit pas forcément être considéré comme un élément unique dans le document : toute section est susceptible de disposer de son `<header>` .

<footer>

L'élément `<footer>` regroupe les contenus du pied d'une section ou d'un document (pied de page) et est destiné à recueillir les informations concernant l'auteur, les mentions légales, etc. Tel `<header>`, l'élément `<footer>` peut apparaître à divers endroits du document.

<nav>

`<nav>` a pour fonction de regrouper les liens de navigation considérés comme majeurs ou jugés suffisamment pertinents. Ceux-ci peuvent être internes ou externes à la page.

<aside>

L'élément `<aside>` représente une portion de contenu contextuelle, directement ou indirectement liée aux éléments qui l'entourent, tel un bloc d'archives relatives au contenu précédent. Par extrapolation, cet élément désigne fréquemment les barres latérales classiques du document et peut remplacer l'ancien `<div id="sidebar">` .

<section>

Une `<section>` représente un bloc générique de contenu ayant la même thématique. Cela concerne les chapitres, les en-têtes et pieds de page, ou toute autre partie dans un document.

<article>

L'élément `<article>` désigne une portion du document potentiellement autonome dans le sens où elle pourrait être reprise ou réutilisée, comme un article de journal, de blog ou de forum.

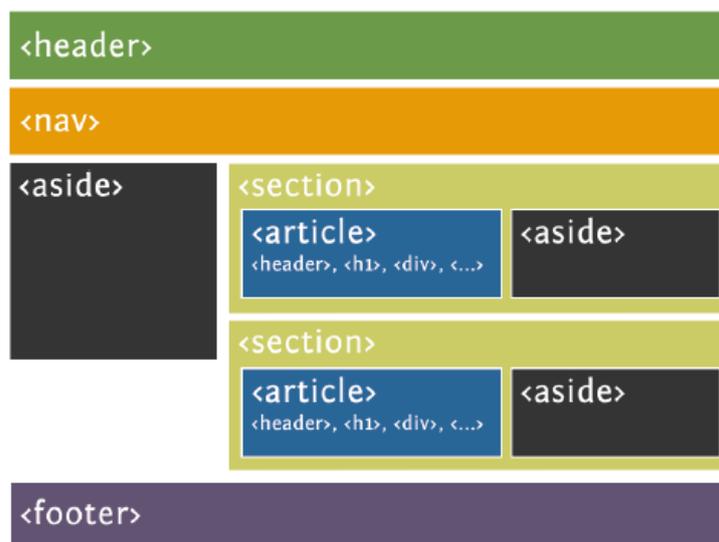


Figure 3. 19 : Exemple de structure des éléments en HTML 5.

↳ [GBM TeleMed](#)

- [Home](#)
- [About](#)
- [Our Doctors](#)
- [Reviews](#)
- [Contact](#)

- [Video call](#)

We take the guesswork out of healthcare decisions

We make all our purchasing decisions based on data. Why should healthcare be any different? .

[Discover more](#)

Telehealth: Technology meets health care

Telehealth is the use of digital information and communication technologies, such as computers and mobile devices, to access health care services remotely and manage your health care.

[Take a Look](#)

Doctors talking to doctors

Figure 3. 20 : Ouvrir notre HTML dans un navigateur.

3.5 Définition de CSS 3 et Bootstrap

Cascading Style Sheets (CSS) est un langage de feuille de style utilisé pour décrire la présentation d'un document écrit en [HTML](#) ou en [XML](#) . CSS décrit la façon dont les éléments doivent être affichés à l'écran, sur du papier, en vocalisation, ou sur d'autres supports.

Les technologies CSS 3 actuellement en cours de développement prévoient de faciliter notre quotidien de concepteur web plus que vous ne pourriez l'imaginer. Les techniques exploitables en production sont déjà très séduisantes : polices exotiques, positionnements intuitifs, effets décoratifs ou d'animation, propriétés avancées ou encore gestion des tailles d'écran [22].



Figure 3. 21 Exemple du code CSS 3.

Bootstrap est un Framework frontal gratuit pour un développement Web plus rapide et plus facile, Bootstrap comprend des modèles de conception basés sur HTML et CSS pour la typographie, les formulaires, les boutons, les tableaux, la navigation, les modaux, les carrousels d'images et bien d'autres, ainsi que des plugins JavaScript en option, Bootstrap vous donne également la possibilité de créer facilement des conceptions réactives [22].

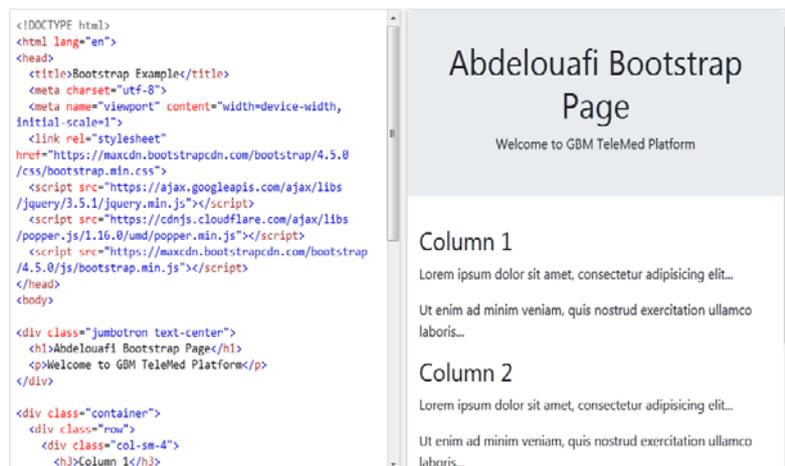


Figure 3. 22 : Exemple du code Bootstrap.

3.6 Définition de JavaScript

JavaScript (qui est souvent abrégé en « JS ») est un langage de script léger, orienté objet, principalement connu comme le langage de script des pages web. Mais il est aussi utilisé dans de nombreux environnements extérieurs aux navigateurs web tels que Node.js, Apache

CouchDB . Le code JavaScript est interprété ou compilé à la volée (JIT). C'est un langage à objets utilisant le concept de prototype, disposant d'un typage faible et dynamique qui permet de programmer suivant plusieurs paradigmes de programmation : fonctionnelle, impérative et orientée objet [24].

3.6.1 Définition de JavaScript, langage de scripts

Le JavaScript est majoritairement utilisé sur Internet, conjointement avec les pages Web HTML. Le JavaScript s'inclut directement dans la page Web (ou dans un fichier externe) et permet de *dynamiser* une page HTML, en ajoutant des interactions avec l'utilisateur, des animations, de l'aide à la navigation, comme par exemple :

- Afficher/masquer du texte ;
- Faire défiler des images ;
- Créer un diaporama avec un aperçu « en grand » des images ;
- Créer des infobulles.

Le JavaScript est un langage dit *client-side*, c'est-à-dire que les scripts sont exécutés par le navigateur chez l'internaute (le **client**). Cela diffère des langages de scripts dits *server-side* qui est exécuté par le serveur Web. C'est le cas des langages comme le PHP [24].

C'est important, car la finalité des scripts *client-side* et *server-side* n'est pas la même. Un script **server-side** va s'occuper de « créer » la page Web qui sera envoyée au navigateur. Ce dernier va alors afficher la page puis exécuter les scripts **client-side** tel que le JavaScript. Voici un schéma reprenant ce fonctionnement :

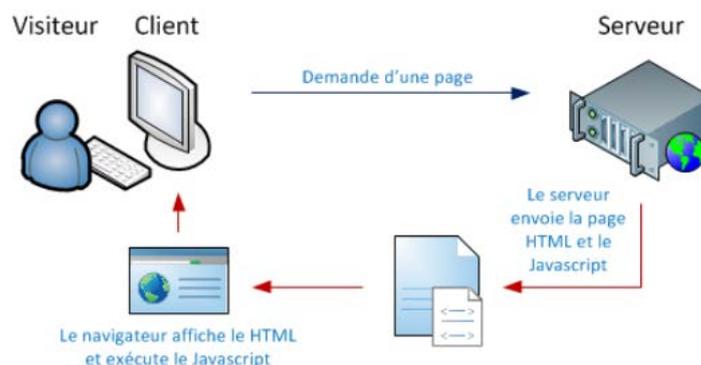


Figure 3. 23 : JavaScript est un langage dit client-side.

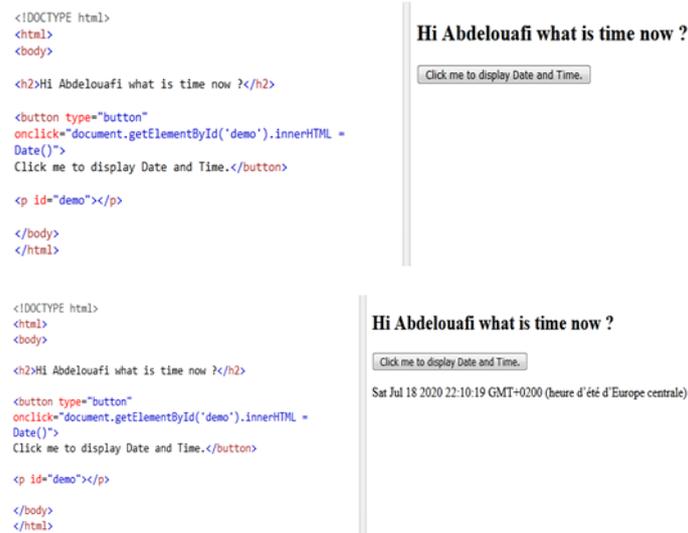


Figure 3. 24 : Exemple du code JS (Avant et après clicker).

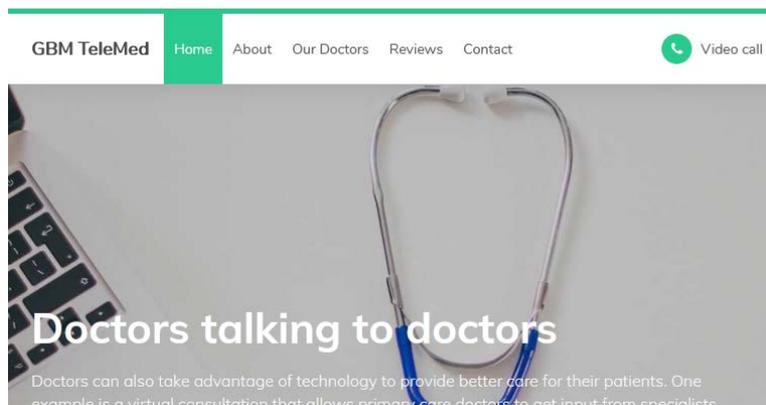


Figure 3. 25 : Page d'accueil GBM TeleMed (créé par HTML5,CSS3,Bootstrap,JS).

3.7 Définition de NODE.JS

NodeJS c'est une plateforme de développement qui met à disposition plusieurs bibliothèques JavaScript directement sur votre machine. Ce n'est pas un serveur, ni un framework, mais un applicatif permettant d'interagir par le biais du langage JavaScript et ainsi effectuer des actions en fonction des entrées / sorties détectées [24].

NodeJS repose entièrement sur le moteur V8 pour fonctionner et permet d'avoir des performances de très haut niveau. D'ailleurs une des utilisations possibles de NodeJS consiste à s'en servir en tant que Serveur Web. Il permet de réaliser les mêmes actions que d'autres langages comme PHP ou Python et s'impose de plus en plus dans ce domaine.

L'un des atouts majeurs de NodeJS est justement ce qui le caractérise, c'est-à-dire l'utilisation du JavaScript. JavaScript est un langage très simple d'apprentissage et très répandu du fait de son utilisation dans les navigateurs Web. Le fait d'utiliser plusieurs langages différents dans

un projet peut rapidement mener à des problèmes de raisonnement ou d'écriture. En utilisant du JavaScript cotés navigateur, mais aussi côté serveur cela permet de simplifier les choses et de construire une solution rapidement.



Figure 3. 26 : Le logo Node.JS

3.7.1 Définition de Node.js : le JavaScript côté serveur

Jusqu'ici, JavaScript avait toujours été utilisé du côté du client, c'est-à-dire du côté du visiteur qui navigue sur notre site. Le navigateur web du visiteur (Firefox, Chrome, IE...) exécute le code JavaScript et effectue des actions sur la page web.

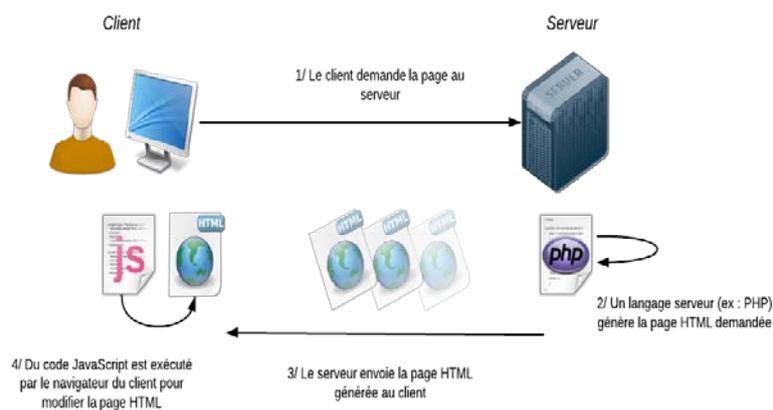


Figure 3. 27 : PHP sur le serveur, JavaScript chez le client.

On peut toujours utiliser du JavaScript côté client pour manipuler la page HTML. Ça ne change pas. Par contre, **Node.js offre un environnement côté serveur** qui nous permet aussi d'utiliser le langage JavaScript pour générer des pages web. En gros, il vient en remplacement de langages serveur comme PHP, Java EE, etc.

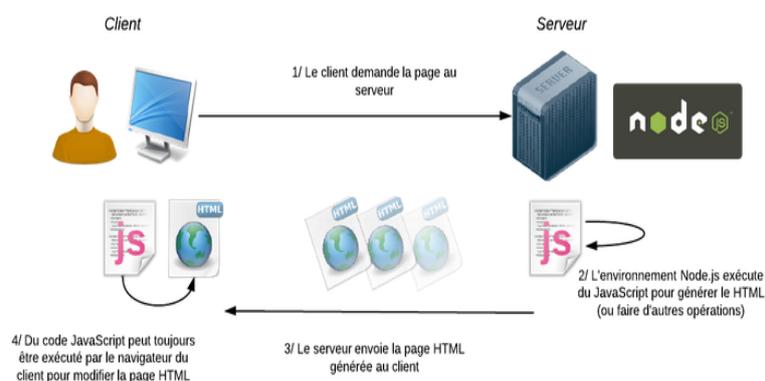


Figure 3. 28 : JavaScript sur le serveur avec Node.JS

3.7.2 Définition de NPM

NPM (Node Package Manager) est un package manager spécialement conçu pour *Node.js*. Il vous permet de partager des **Packages** (aussi nommés *modules*) pour qu'ils soient accessible publiquement, permettant alors à d'autres utilisateurs de les installer simplement dans leur projet. Il en existe plus de 200.000 au jour d'aujourd'hui, avec plus de 2 milliards de téléchargements par mois.

Un package est un **ensemble de fonctions** réalisé pour effectuer diverses tâches pour vous simplifier la vie. On y trouve par exemple *Express* qui est le plus célèbre Framework de *Node.js*, ou encore *Cordova*, *Grunt*, *Angular* et j'en passe.

Ces packages sont très célèbres et sont essentiellement réalisés par de grosses entreprises (comme Google), mais la plupart sont créés par des utilisateurs seuls. Vous trouverez certainement ce que vous cherchez car un développeur est certainement tombé sur la même problématique que vous, l'a résolu et a décidé de partager sa solution sous forme de package sur NPM. Vous n'aurez plus qu'à l'installer en une simple ligne de commande pour utiliser toutes ses fonctions.



Figure 3. 29 : Le logo NPM.

3.7.3 Configurer un serveur statique (local)

La configuration d'un serveur Web local est la première étape de la ceinture d'outils de tout développeur Web. En conjonction avec les éditeurs de texte, les serveurs Web statiques sont également nombreux et varient d'un langage à l'autre. On préfère est d'utiliser Node.js avec node-static, car c'est un excellent serveur Web facile à utiliser, en suivant les étapes suivantes:

1. Visitez le site Web Node.js à <http://nodejs.org/>. Il devrait y avoir un gros bouton INSTALL sur la page d'accueil qui vous aidera à installer Node.js sur votre système d'exploitation.
2. Une fois Node.js installé sur votre système, vous aurez également le package gestionnaire pour Node.js installé appelé gestionnaire de packages de nœuds (nm).
3. Ouvrez un terminal ou une interface de ligne de commande et tapez `npm install -g node-static` (vous aurez probablement besoin de privilèges d'administrateur).

4. Vous pouvez désormais accéder à n'importe quel répertoire contenant les fichiers HTML que vous souhaitez héberger sur le serveur.
5. Exécutez la commande statique pour démarrer un serveur Web statique dans ce répertoire. Tu peux accéder à `http://localhost:5000` pour voir votre fichier dans le navigateur!

Allez à votre cmd ou terminal et écrivez **NPM init** pour créer votre package.json

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: cmd
C:\Users\ISS\Desktop\Travail PFE\GBM TeleMed>npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (gbm-telemed)
version: (1.0.0)
description: gbm-platforme
entry point: (server.js)
test command:
git repository:
keywords:
author: Abdelouafi ZAHRA
license: (ISC)
About to write to C:\Users\ISS\Desktop\Travail PFE\GBM TeleMed\package.json:
```

puis installez le package **node-static**

```
C:\Users\ISS\Desktop\Travail PFE\GBM TeleMed>npm install node-static --save
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN gbm-telemed@1.0.0 No repository field.

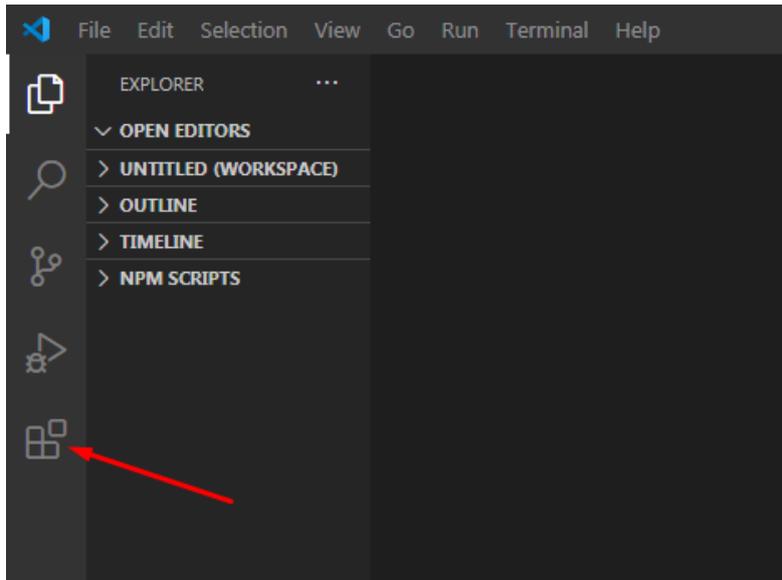
+ node-static@0.7.11
added 6 packages from 8 contributors and audited 6 packages in 3.788s
found 4 vulnerabilities (3 low, 1 high)
  run `npm audit fix` to fix them, or `npm audit` for details

C:\Users\ISS\Desktop\Travail PFE\GBM TeleMed>
```

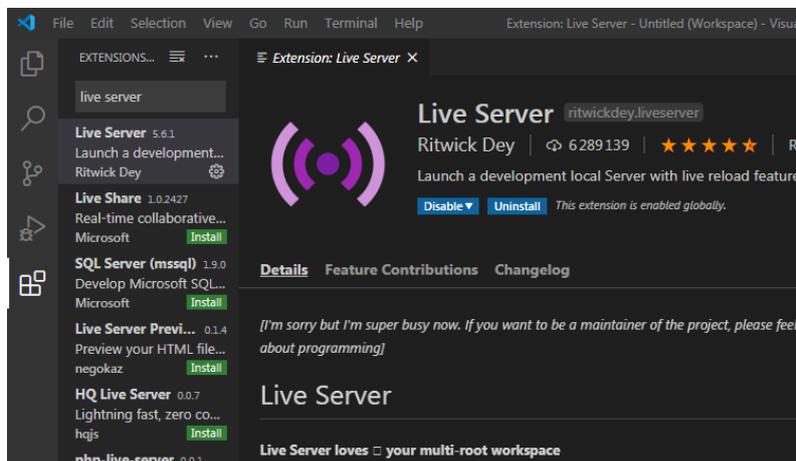
Ensuite écrivez **node server** pour accéder à votre fichier server javascript ce fichier est très important vous pouvez le écrire vous même ou allez à la bibliothèque NPM (www.npmjs.com) écrivez node-static et copiez le script. moi je change le numéro du port 5000 au lieu de 8080.

```
C:\Users\ISS\Desktop\Travail PFE\GBM TeleMed>node server
server started on port 5000
█
```

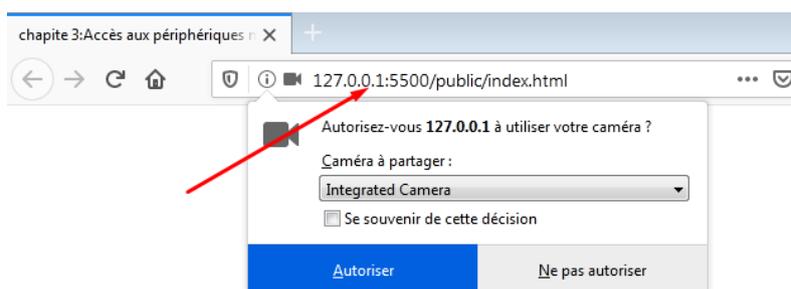
Pour ne pas trop compliquer les choses il existe une méthode raccourcie si on travaille sur Visual Studio code on va à Extensions



Puis écrit Live Server et click sur install



Enfin ouvrir votre Projet avec Live server



3.7.4 Le Déploiement de GBM TeleMed

Le protocole SSL (Secure Sockets Layer) était le protocole cryptographique le plus largement utilisé pour assurer la sécurité des communications sur Internet.

Le SSL crée un canal sécurisé entre deux machines ou appareils communiquant sur Internet ou un réseau interne. Son usage le plus courant est la sécurisation de la communication entre

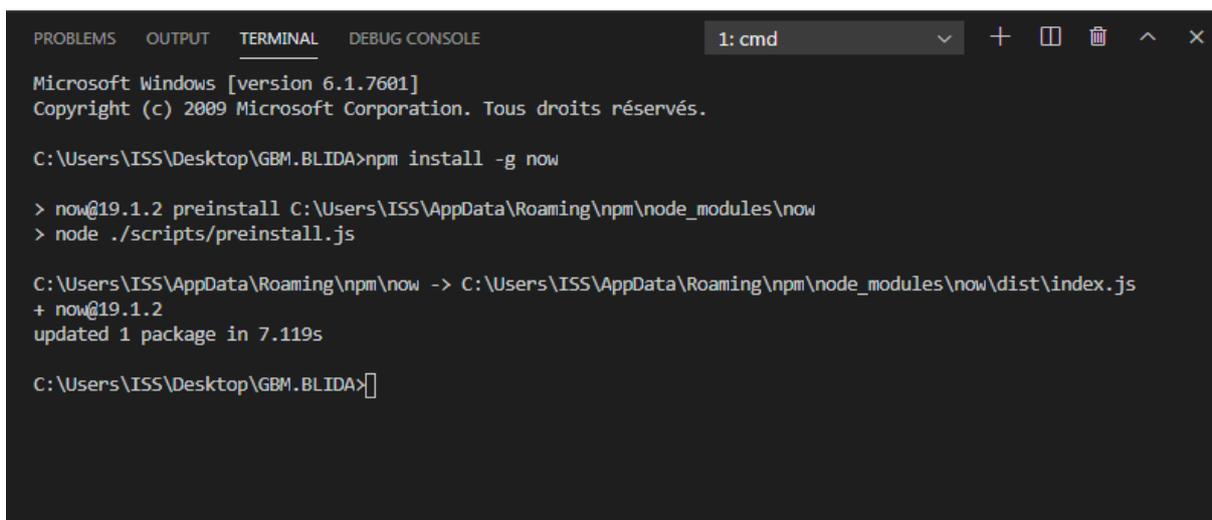
un navigateur web et un serveur web. L'adresse URL passe alors de HTTP a HTTPS, le 'S' signifiant 'sécurisé'.

Malheureusement, tester le code de notre projet *GBM TeleMed* n'est pas possible sans l'exécuter sur un serveur HTTPS Théoriquement, vous pouvez générer un certificat auto-signé pour votre serveur Express afin d'exécuter l'application au sein de votre réseau interne, Mais la mauvaise nouvelle est que les navigateurs ne vous permettront pas d'accéder à la webcam si le certificat ne provient pas d'une autorité de confiance.

La solution la plus simple pour tester notre code consiste à le déployer sur un serveur public prenant en charge le protocole HTTPS.

Cette méthode que on est sur le point d'utiliser est l'un des moyens les plus simples de déployer une application NodeJS. Tout ce que on a à faire est de commencer par créer un compte sur vercel.com

Choisissez simplement le forfait gratuit. Vous devrez fournir votre adresse e-mail. Vous devrez également vérifier votre adresse e-mail pour que votre compte soit activé. Ensuite, installez maintenant l'outil CLI sur votre système: **npm install -g now**



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: cmd
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Users\ISS\Desktop\GBM.BLIDA>npm install -g now

> now@19.1.2 preinstall C:\Users\ISS\AppData\Roaming\npm\node_modules\now
> node ./scripts/preinstall.js

C:\Users\ISS\AppData\Roaming\npm\now -> C:\Users\ISS\AppData\Roaming\npm\node_modules\now\dist\index.js
+ now@19.1.2
updated 1 package in 7.119s

C:\Users\ISS\Desktop\GBM.BLIDA>
```

Une fois l'installation terminée, vous pouvez déployer l'application. Exécutez simplement la commande "**now**" au niveau de la racine du dossier de projet:

Si c'est la première fois que vous exécutez la commande, vous serez invité à saisir votre adresse e-mail. Vous recevrez ensuite un e-mail dont vous aurez besoin pour vérifier votre connexion. Une fois la vérification terminée, vous devrez à nouveau exécuter la commande **now**. Après quelques secondes, votre application sera opérationnelle à une URL spécifiée qui sera imprimée sur le terminal.

Si vous utilisez le terminal intégré VSCode, appuyez simplement sur ALT et cliquez pour ouvrir l'URL dans votre navigateur.

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE 1: cmd
Microsoft Windows [version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tous droits réservés.

C:\Users\ISS\Desktop\GBM.BLIDA>npm install -g now

> now@19.1.2 preinstall C:\Users\ISS\AppData\Roaming\npm\node_modules\now
> node ./scripts/preinstall.js

C:\Users\ISS\AppData\Roaming\npm\now -> C:\Users\ISS\AppData\Roaming\npm\node_modules\now\dist\index.js
+ now@19.1.2
updated 1 package in 7.119s

C:\Users\ISS\Desktop\GBM.BLIDA>now
Now CLI 19.1.2
? Set up and deploy "~\Desktop\GBM.BLIDA"? [Y/n] y
? Which scope do you want to deploy to? Abdelouafi zahra
? Found project "abdelouafi-zahra/gbm-blida". Link to it? [Y/n] y
? Linked to abdelouafi-zahra/gbm-blida (created .vercel and added it to .gitignore)
? Inspect: https://vercel.com/abdelouafi-zahra/gbm-blida/8ai690p2w [4s]
? Preview: https://gbm-blida.abdelouafi-zahra.vercel.app [copied to clipboard] [6s]
? To deploy to production (gbm-blida.vercel.app), run `vercel --prod`

C:\Users\ISS\Desktop\GBM.BLIDA>[]

NOTRE URL
```

Vous devez autoriser la page à accéder à votre caméra et à votre microphone, vous devez accéder à un autre appareil, tel qu'un autre ordinateur portable ou smartphone avec une caméra frontale. Vous pouvez également demander à un ami disposant d'une connexion Internet de vous aider. Accédez simplement à la même URL, en quelques secondes, les deux appareils doivent être connectés au salon de discussion GBM TeleMed.

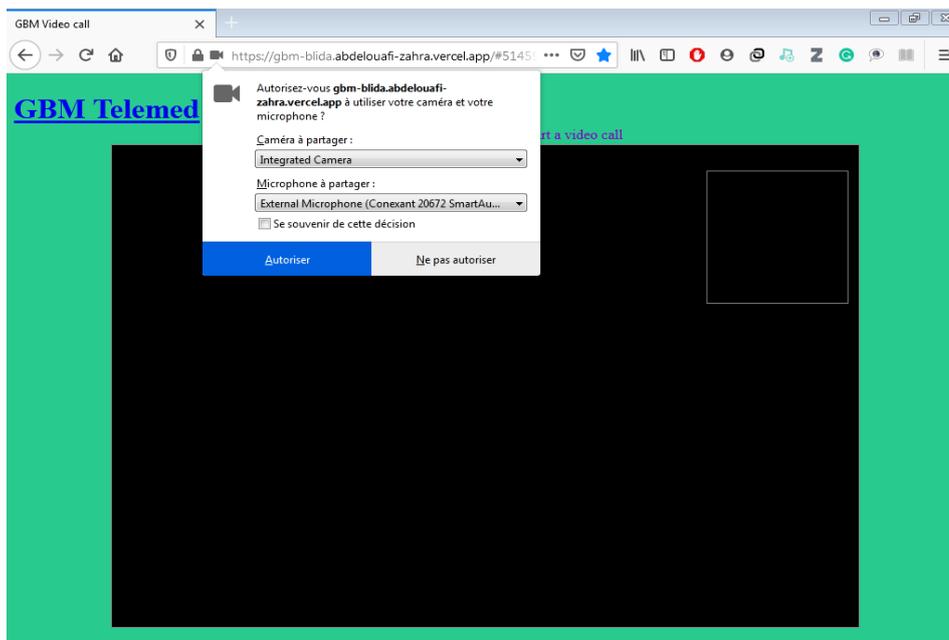


Figure 3. 30 :La page video call de GBM TeleMed.

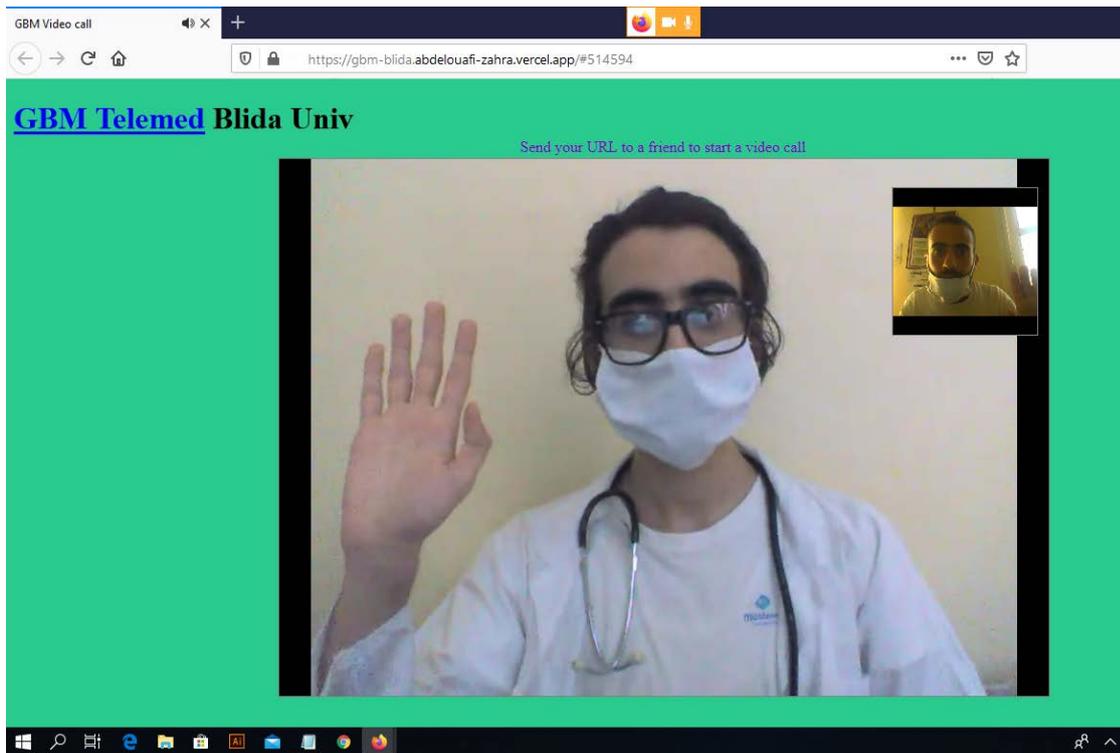


Figure 3. 31 : Téléconsultation avec GBM TeleMed.

3.8 Conclusion

Dans ce chapitre, on a présenté les ingrédients techniques du notre projet Plate-forme GBM TeleMed basée sur le WebRTC qui utilise des composants logiciels open source, ainsi un petit tutoriel de déploiement avec Node.js .

Conclusion Générale

Conclusion Général

Les acteurs de la santé ont le rôle d'échanger des informations, d'effectuer des consultations, d'interpréter des images médicales, etc., et cela grâce aux moyens de télécommunications. Ces nouvelles technologies (TICs) permettent d'abolir les distances, d'éviter des déplacements inutiles, afin de réaliser l'acte médical le plus approprié dans un environnement favorable. C'est la raison pour laquelle la télémédecine a vu le jour.

La télémédecine est destinée à pallier partiellement les déserts médicaux, à améliorer l'efficacité du parcours de soin, à réduire le temps de prise en charge des patients et enfin à favoriser le transfert des informations médicales.

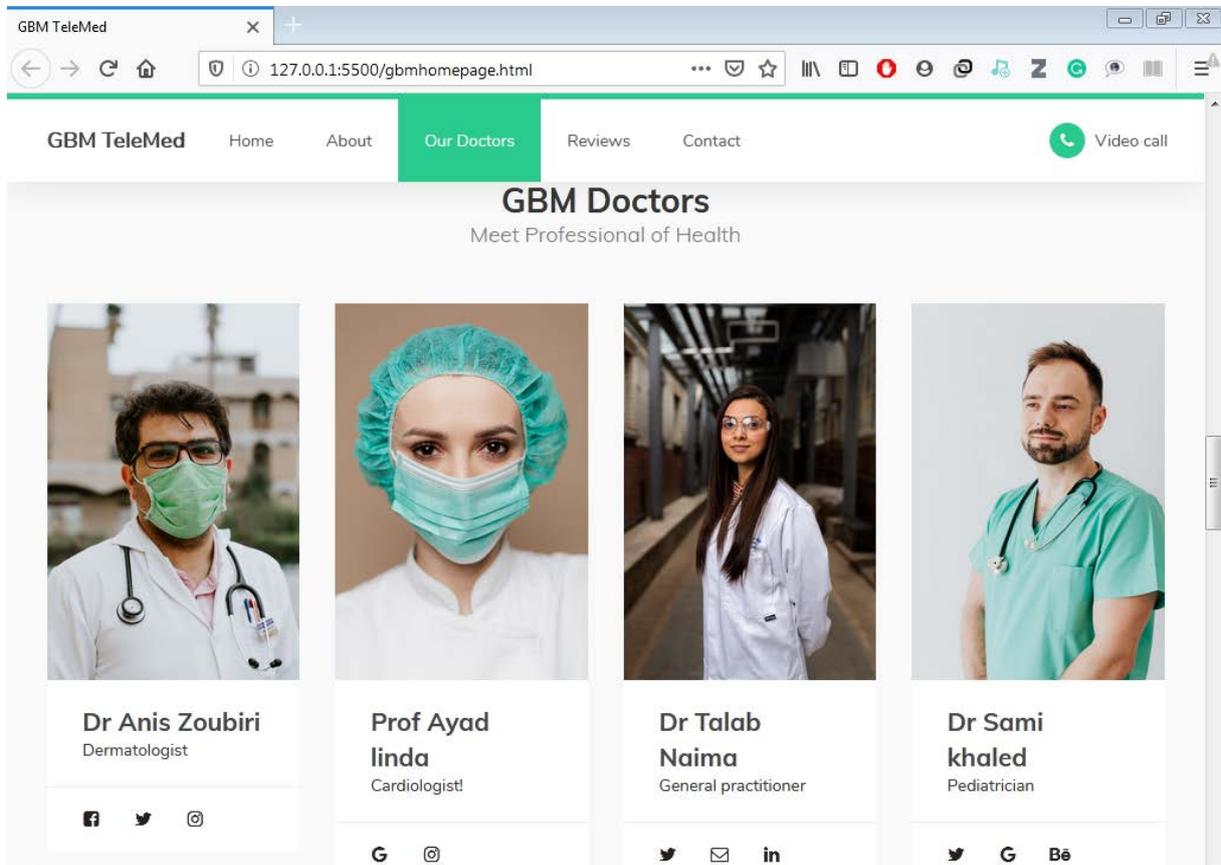
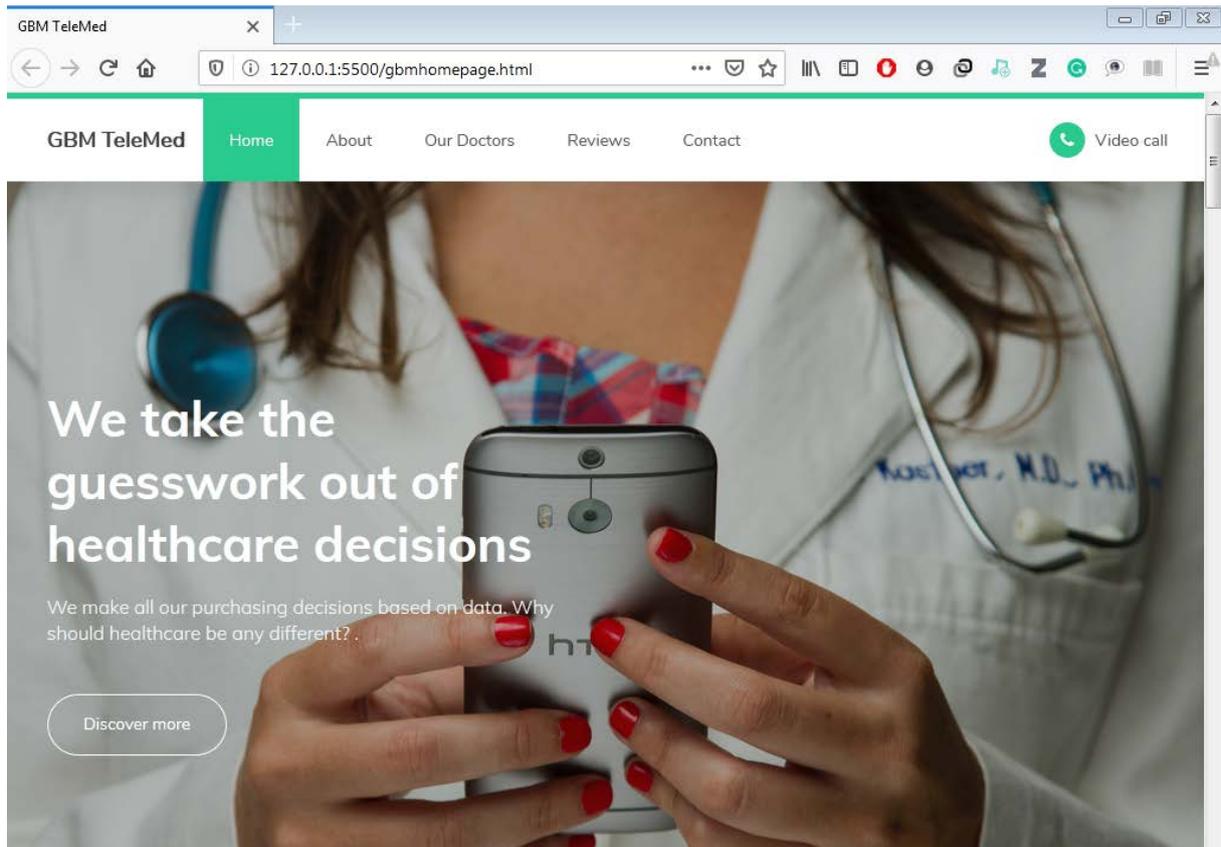
Les travaux menés dans ce projet de fin d'étude, ont pour objectif de répondre à ces problématiques, ils visent la conception, le développement et le déploiement de services en télémédecine, adaptés aux environnements web sous le nom GBM TeleMed (Génie Biomédical télémédecine).

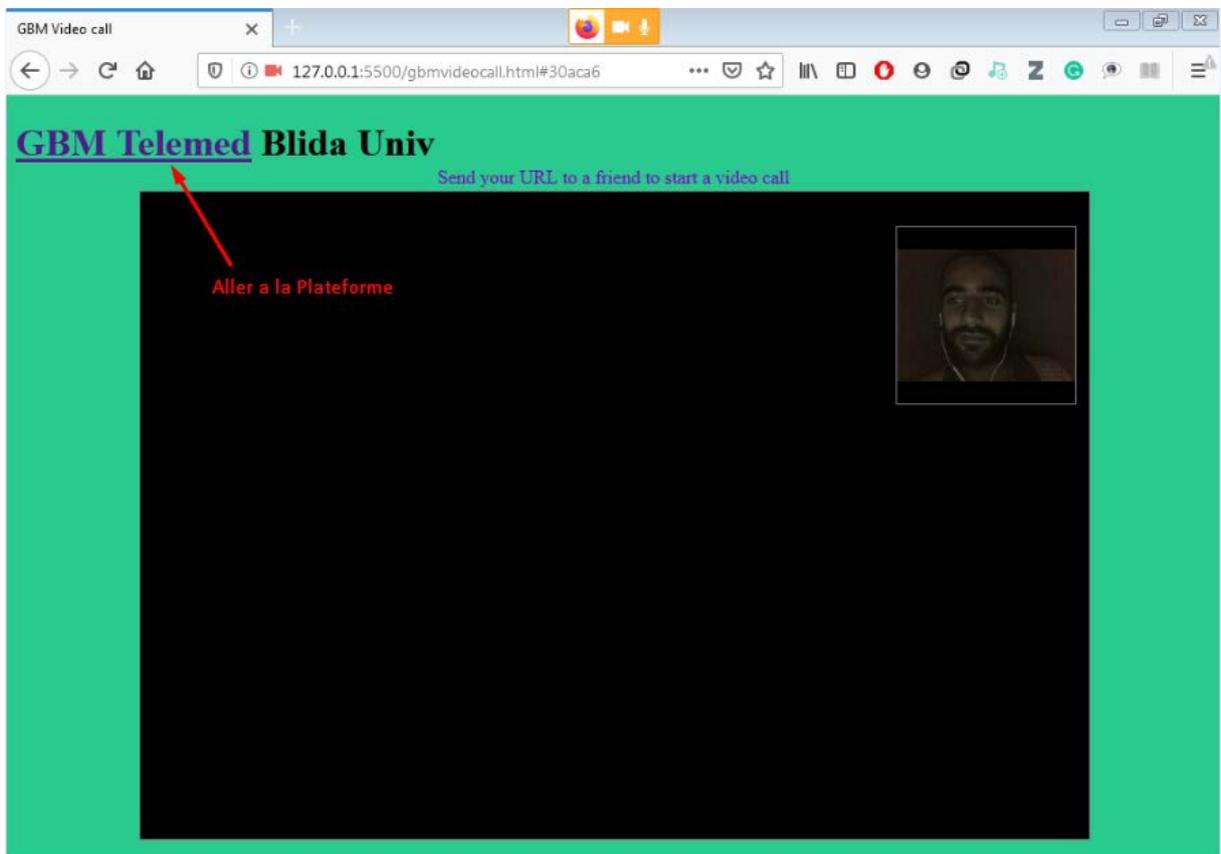
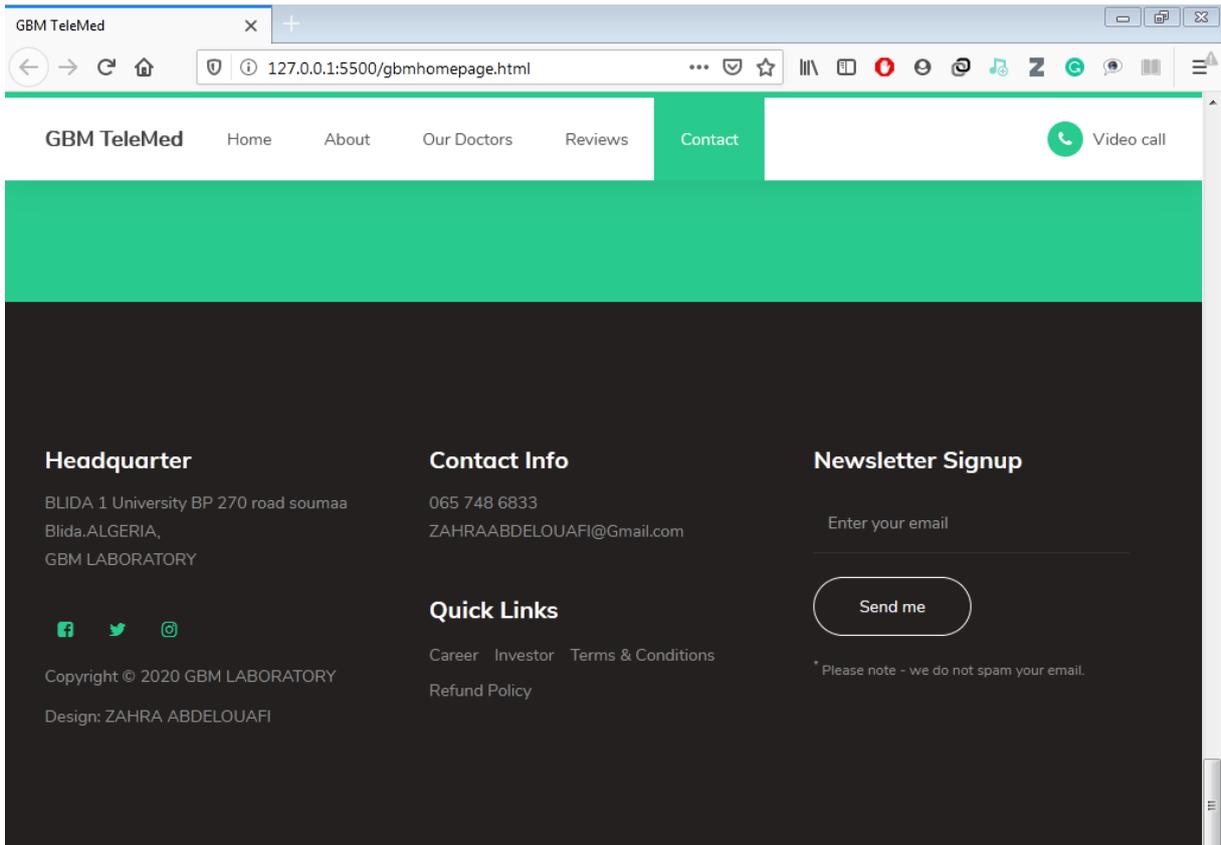
L'expérience de l'être humain avec les différents outils de codage (HTML, Javascript et autre), ont démontré comment l'ordinateur traite et réintroduit chaque information/ code enregistré par l'utilisateur, pour créer quelque chose de plus développée,

Certes, même l'ordinateur est créé par l'humain, qu'il lui a programmé à traiter les informations de cette façon, pour faciliter la tâche aux utilisateurs, et leur permettre d'exprimer leur créativité, et le plus important c'est de savoir comment utiliser cette technologie pour créer des services, des plateformes dématérialisées, des services qui répondent aux besoins actuels de la société d'aujourd'hui,

WebRTC est l'un des ajouts les plus importants à la plate-forme Web d'aujourd'hui, il apporte toute une suite de nouvelles technologies, telles que les caméras, les données en continues et même une toute nouvelle pile de protocoles réseau. Il est étonnant de voir non seulement la quantité de travail entrant dans l'API WebRTC, mais aussi de savoir que tout cela est gratuit pour tous les développeurs d'applications, ce qui leur permet de créer plus de services et de partager plus de savoir, et participer à la création d'un meilleur avenir à l'humanité,

ANNEXE : L'interface de la Plateforme





Bibliographie

- [1] J. Marine, « LE BIG DATA EN SANTE ENJEUX ET USAGES DANS LA PRISE EN CHARGE NUMERIQUE DES PATIENTS REVUE DE LA LITTERATURE », p. 101.
- [2] « Étude - La dynamique d'internet. Prospective 2030 ». <http://archives.strategie.gouv.fr/cas/content/etude-dynamique-internet-2030.html> (consulté le août 05, 2020).
- [3] « Organigramme | ESEA - e-santé en action ». <https://www.esea-na.fr/organigramme> (consulté le août 05, 2020).
- [4] « RIA - Robotics Online - Industrial Robotics ». <https://www.robotics.org/> (consulté le août 05, 2020).
- [5] E. Puydebois, « Le développement de la M-Santé en France et l'analyse du rôle de l'équipe officinale », Limoges, 2012.
- [6] Colnet Henriette de, « Apports de l'e-santé dans les recherches impliquant la personne humaine / Henriette de Colnet ; sous la direction de Jean-Louis Cazin », Thèse d'exercice Pharmacie Lille 2018, 2018.
- [7] Surville Arthur, « Objets connectés et dispositifs médicaux connectés: principaux outils disponibles à la pratique de la médecine générale en France en 2018 / Arthur Surville ; directeur de thèse, Pr Pierre Boyer », Thèse d'exercice MédecineMédecine générale Toulouse 3 2018, 2018.
- [8] « Ministère des Solidarités et de la Santé ». <https://solidarites-sante.gouv.fr/> (consulté le août 05, 2020).
- [9] Collège des médecins du Québec, *Le médecin, la télémédecine et les technologies de l'information et de la communication: guide d'exercice*. Montréal: Collège des médecins du Québec, 2015.
- [10] R. S. Khandpur, *Telemedicine: Technology and Applications*. PHI Learning, 2017.
- [11] E. Universalis, « Encyclopédie Universalis », *Encyclopædia Universalis*. <https://www.universalis.fr/> (consulté le août 05, 2020).
- [12] B. Fong, A. C. M. Fong, et C. K. Li, *Telemedicine Technologies: Information Technologies in Medicine and Telehealth*, 1 édition. Wiley, 2011.
- [13] « Electrotechnique ». <https://sitelec.org/> (consulté le août 05, 2020).
- [14] « ResearchGate | Find and share research », *ResearchGate*. <https://www.researchgate.net/> (consulté le août 05, 2020).

- [15] Bernard Thibault, « Normes et Standards en Informatique de Santé Echange de données Communication », p. 36, 2003.
- [16] A. G. Blank, *TCP / IP JumpStart: Internet Protocol Basics*, 2nd Edition. San Francisco: Sybex, 2002.
- [17] M. W. Murhammer, O. Atakan, S. Bretz, L. R. Pugh, K. Suzuki, et D. H. Wood, « TCP/IP Tutorial and Technical Overview », p. 738.
- [18] J. Bentham et M. Barr, *Tcp/Ip Lean: Web Servers for Embedded Systems*, 2nd New edition. Lawrence, Kan: R & D, 2002.
- [19] « WebRTC », *WebRTC*. <https://webrtc.org/?hl=fr> (consulté le août 05, 2020).
- [20] S. Loreto et S. P. Romano, *Real-Time Communication with WebRTC: Peer-to-Peer in the Browser*, 1 edition. Beijing: O'Reilly Media, 2014.
- [21] R. Goetter et D. Giazman, *CSS avancées: vers HTML5 et CSS3*. Paris: Eyrolles, 2012.
- [22] Scaledrone, « Scaledrone realtime messaging service ». <https://www.scaledrone.com/> (consulté le août 05, 2020).
- [23] « W3Schools Online Web Tutorials ». <https://www.w3schools.com/> (consulté le août 05, 2020).
- [24] Node.js, « Téléchargements », *Node.js*. <https://nodejs.org/fr/download/> (consulté le août 05, 2020).

