

## Rapport de stage

*Réalisé à*



**LOGO DE L'ENTREPRISE**

*Préparé par*

**Nom & prénom**

**Kchaou Anis   Turki Maki   Firas Makni   Bouchaala Oûmar**

---

# SOLUTION VÍSIOCONFÉRENCE

---

*Encadré par :*

M. ....

**Encadrant industriel**

M. ....

**Encadrant académique**

# Remerciement

Que gloire et louange soient rendues au Bon Dieu qui est source de toute vie, de toute sagesse, de toute intelligence. En qui nous trouvons toutes nos inspirations. Il pourvoit à nos besoins et sans cesse veille sur nous.

Nous ne pourrions commencer ce rapport sans remercier notre encadreur Monsieur Tarak CHAARI qui n'a cessé de nous guider et de nous faire bénéficier de leur grand savoir.

Nous remercions aussi chaleureusement les membres de jury d'avoir accepté de juger ce travail. Qu'ils trouvent ici nos reconnaissances et nos respects.

Finalement nous tenons également à remercier tous nos enseignants pour la qualité de l'enseignement qu'ils ont bien voulu nous prodiguer durant nos études afin de nous donner une formation de qualité.

# Listes des figures

[Figure 1.1 Composants de la plateforme BigBlueButton](#)

[Figure 1.2 Architecture de la plateforme BigBlueButton](#)

[Figure 1.3 architecture de 5 apps](#)

[Figure 1.4 Le schéma des différents composants d'Apps Akka](#)

[Figure 1.5 Le schéma des FsEslAkka](#)

[Figure 1.6 architecture h 323](#)

[Figure 1.7 architecture SIP](#)

[Figure 3.1 L'architecture webRTC](#)

[Figure 3.2 Les étapes d'une connexion entre deux clients utilisant WebRTC](#)

[Figure 3.3 les différentes Composants du webRTC](#)

[Figure 3.4 L'architecture de l'API RTCPeerconnection](#)

[Figure3.5 le paquet ip](#)

[Figure 3.6 structure d'un Flux Media](#)

[Figure3.7 le paquet UDP](#)

[Figure3.8 Transiter au travers d'un pare-feu](#)

[Figure3.9 Transiter au travers d'un NAT](#)

[Figure3.10 architecture de STUN](#)

[Figure 3.11 RTP dans le modèle OSI](#)

[Figure3.12 le paquet UDP](#)

[Figure3.13 transport des flux RTSP](#)

[Figure3.14 déroulement de la communication dans le websocket](#)

[Figure3.15 architecture socket io](#)

## Listes des Abreviations

**RTSP**: Real Time Streaming Protocol

**SAP**: Session Announcement Protocol

**RTP**: Real-Time Transport Protocol

**SDP**: Session Description Protocol

**SRTP**: Secure Real-time Transport Protocol

**DTLS**: Datagram Transport Layer Security

**STUN**: Simple Traversal of UDP through NATs

**WebRTC**: Web Real-Time Communication

## Sommaire

Introduction Générale.....	7
1. Introduction : .....	9
2. Etude préalable .....	9
2.1. Les champs de l'étude .....	9
2.2. Objectif à atteindre .....	9
3. Etude de l'existant : .....	10
3.1. la plateforme BigBlueButton.....	10
3.1.1. Présentation .....	10
3.1.2. Fonctionnalités de BigBlueButton .....	10
3.1.3. Composants de la plateforme BigBlueButton .....	10
3.1.4. Les modes dans la plateforme BigBlueButton .....	11
3.1.5. Architecture de la plateforme BigBlueButton .....	11
3.1.6. Différentes versions de BigBlueButton .....	15
3.1.7. Les limites de la plateforme BigBlueButton .....	16
3.2. La technologie VOIP .....	16
3.2.1. Les protocoles de la technologie VOIP .....	16
3.2.2. Le protocole H.323.....	17
3.2.3. Le protocole SIP ( Session Initiation Protocol ) .....	18
3.2.4. Les problèmes de voip.....	18
4. Solution proposées .....	18
5. Conclusion.....	19
1. Introduction : .....	21
2. Spécification des besoins : .....	21
2.1 Les besoins fonctionnels : .....	21
2.2 Les besoins non fonctionnels : .....	22
3. Conclusion.....	23
1. Introduction .....	25
2. Installation et configuration de la plateforme BigBlueButton.....	25
2.1. BigBlueButton version flash .....	25
2.2. On constate que ruby est bien installé. ....	28
2.3. La plateforme BigBlueButton version HTML5 .....	32
3. Les technologies utilisées.....	34
3.1 La technologie webRTC.....	34

3.1.1	Architecture .....	34
3.1.2	Principe du webRTC .....	35
3.1.3	Description générale de la norme .....	37
3.1.4	Les Composants du webRTC .....	38
3.1.5	Problèmes posés par WebRTC et solutions potentielles .....	41
3.1.6	protocoles utilisées par webRTC.....	44
3.1.7	le codec utilisés par webRTC .....	48
3.2	WebSocket.....	48
3.2.1	socket.io.....	49
4	Les modules.....	51
4.1.	Web Conférence .....	51
4.2.	partage des documents.....	51
4.3.	enregistrement .....	51
4.4.	Le tableau blanc.....	51
5	Conclusion.....	51
	Conclusion.....	52

## Introduction Générale

Avec l'accroissement incessant du nombre d'internautes qui demandent de plus en plus des services en ligne, plusieurs institutions et établissements cherchent à faire migrer leurs services du réel au virtuel pour pouvoir rapprocher les deux bouts du monde. De même, les grands développeurs du web se sont mis en compétition pour développer des plateformes performantes de conférence web.

Avec les outils de conférence web, les collaborateurs sont réunis autour d'une plateforme virtuelle pour des réunions de travail, des formations ou encore des séminaires, tous les participants, devant leurs ordinateurs et connectés à Internet, communiquent et échangent les données en temps réel et à distance.

De plus, opter pour une solution web conférence permet un gain de temps et d'argent pour tous les collaborateurs où qu'ils se trouvent puisque les déplacements sont supprimés. Ça permet aussi de faire des économies de papier.

Nous avons opté, ainsi à planifier ce présent rapport sur trois grands chapitres comme suit :

- Le **premier chapitre** s'intitule « Etude préalable » : qui s'intéresse à la description de l'existant, la solution retenue et la spécification des besoins de l'application.

- Le **deuxième chapitre** « Analyse du besoin » : présente une analyse des faiblesses des systèmes existants et les suggestions des utilisateurs de tel plateforme et de proposer éventuellement des solutions à chaque problème.

- Le **troisième chapitre** intitulé « Réalisation » : présente les différentes étapes d'installation et d'exploitation de la plateforme BIGBLUEBUTTON et les différentes technologies mises en service pour assurer le fonctionnement adéquat de tels systèmes.

# **Chapitre1 : Etude Préalable**



## **1. Introduction :**

L'étude préalable constitue une étape préliminaire pour la réalisation d'un projet. En effet, elle nous permet d'analyser et de critiquer l'existant, tout en proposant éventuellement des solutions innovantes. Dans ce chapitre on va présenter une étude de la plateforme de conférences web dénommée «BigBlueButton» dans le but de dégager ses limites d'application et justifier la nécessité de le développer et d'apporter des améliorations éventuelles sur le système.

## **2. Etude préalable**

### **2.1. Les champs de l'étude**

- La visioconférence ou vidéoconférence : est un moyen de communication de plus en plus utilisé par les entreprises aujourd'hui. De par son aspect pratique, la visioconférence ( vidéoconférence ) intéresse les entreprises souhaitant communiquer à distance avec d'autres entreprises ou personnes en évitant de se déplacer.
- L'apprentissage en Vidéo en ligne (ou V-Learning) est un outil des technologies de l'information et de la communication pour l'éducation(TICE) qui marie l'informatique, l'audiovisuel et les sciences de l'éducation et de la formation selon un concept de vidéo numérique. Il privilégie les sociétés à tradition orale à l'inverse du E-Learning qui avantage les sociétés à tradition écrite.
- La télémédecine regroupe les pratiques médicales permises ou facilitées par les télécommunications. C'est un exercice de la médecine par le biais des télécommunications et des technologies qui permettent les prestations de santé à distance et l'échange de l'information médicale s'y rapportant.
- Visioconférence de presse :La visioconférence de presse permet d'organiser des conférences de presse internationales par visioconférence, à travers internet.

### **2.2. Objectif à atteindre**

- recréer de manière très proche un univers de réunion :
- offre un gain de déplacement, donc un gain de temps

- permet de gagner en rapidité dans la prise de décision
- permettent de partager et faire défiler des documents pendant la réunion
- renforcer la réactivité et créez des relations professionnelles extraordinaires

### **3. Etude de l'existant :**

#### **3.1. la plateforme BigBlueButton**

##### **3.1.1. Présentation**

BigBlueButton est une plateforme Open source de web conférence qui permet aux différentes institutions éducatives du monde entier de communiquer leur services d'enseignement (cours, TD, présentations...) en ligne et en multimédia, à un coût réduit. Cette plateforme contient de nombreux outils utiles pour aider à la fois les formateurs et les étudiants à accomplir leur tâche de formation et d'éducation. La plateforme est constitué de plusieurs composants open source et peut fonctionner sous des environnements divers tels que : Linux, Mac, et Windows. La plateforme BigBlueButton, basées sur serveur Asterisk, possède une très bonne qualité audiovisuelle et offre de multiples fonctionnalités tels que : les appels audio et vidéo, le transfert d'appel etc.

##### **3.1.2. Fonctionnalités de BigBlueButton**

Les fonctionnalités de la plateforme BigBlueButton sont :

- Vidéo conférence ;
- Audioconférence ; Partager un bureau à distance ;
- Afficher un document bureautique (PDF, Power Pointe, Images, etc.);
- Tableau blanc ;
- Discussion instantanée (Chat privé et public)
- Voir tous les participants
- Levez la main
- Suivre les présentations
- Partager sa vidéo
- Voir le curseur du présentateur
- Rendre muet ou Ejecter un participant
- Partager des fichiers
- Accorder le droit de présentateur...
- 

##### **3.1.3. Composants de la plateforme BigBlueButton**

Les principales composantes de la plateforme BigBlueButton sont : Asterisk ; Tomcat; MySQL; NGINX ; Red5 ; Flex SDK (Adobe) ; Open office, Libre office, HTML5.....



*Figure 1.1 Composants de la plateforme BigBlueButton*

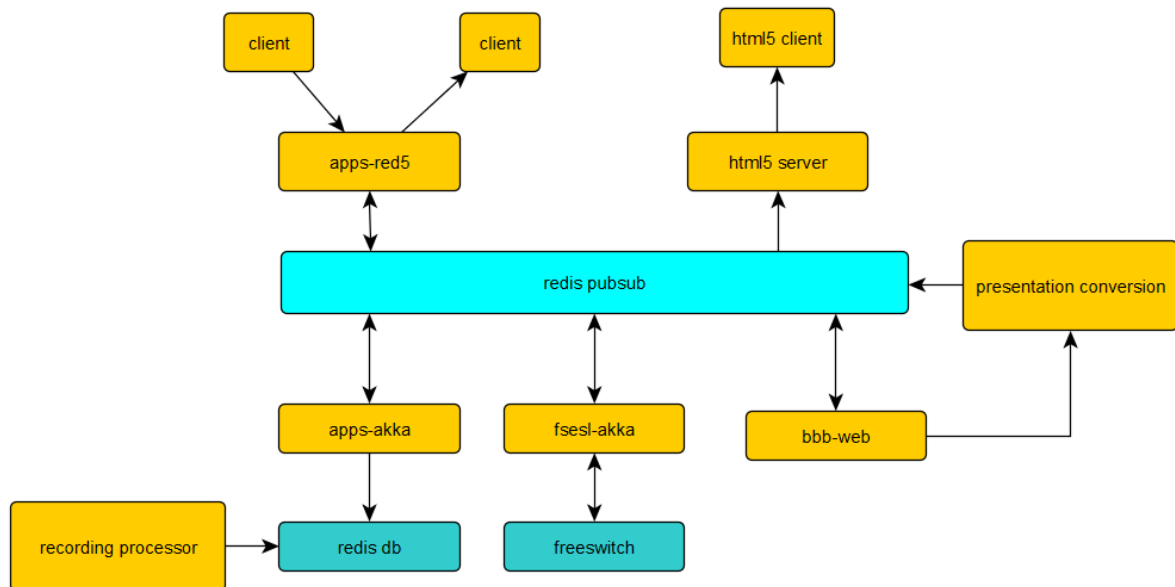
### 3.1.4. Les modes dans la plateforme BigBlueButton

La communication avec la plateforme BigBlueButton est effectuée selon trois modes différents en tant que:

- **Visualiseur (Viewer)** : où ses droits sont limités. Il peut suivre les présentations, visualiser et partager les vidéos des autres, faire du chat et peut aussi poser des questions. C'est le rôle qu'il convient de donner à un étudiant par exemple.
- **Présentateur (Presenter)** : qui dispose de toutes les fonctionnalités qu'offrent BigBlueButton. Il a ainsi le droit de soumettre et de modifier sa présentation, de partager son bureau et autres...
- **Modérateur (Moderator)** : qui dispose de toutes les fonctions qu'offrent le site BigBlueButton et ainsi il peut aider le présentateur dans la gestion des étudiants en accordant le droit de poser des questions à un étudiant qui se manifeste pour un service donné.

### 3.1.5. Architecture de la plateforme BigBlueButton

La structure de la plateforme est présentée sur la figure 1.2



*Figure 1.2 Architecture de la plateforme BigBlueButton*

La plateforme BigBlueButton est composée des éléments suivants :

### **BBB-web :**

L'API Web fournit le point d'extrémité d'intégration pour les applications tierces telles que Moodle, Wordpress, Canvas, Sakai, etc. - pour contrôler le serveur BigBlueButton. Chaque accès à BigBlueButton passe par un portail frontal (nous appelons une application tierce). Pour activer le système de gestion de l'apprentissage (LMS) comme Moodle, les enseignants peuvent installer des salles BigBlueButton dans leur cours et les étudiants peuvent accéder aux salles d'enregistrements. Le BigBlueButton est livré avec quelques démonstrations d'API simple

### **Redis PubSub**

Redis PubSub fournit un canal de communication entre différentes applications côté serveur.

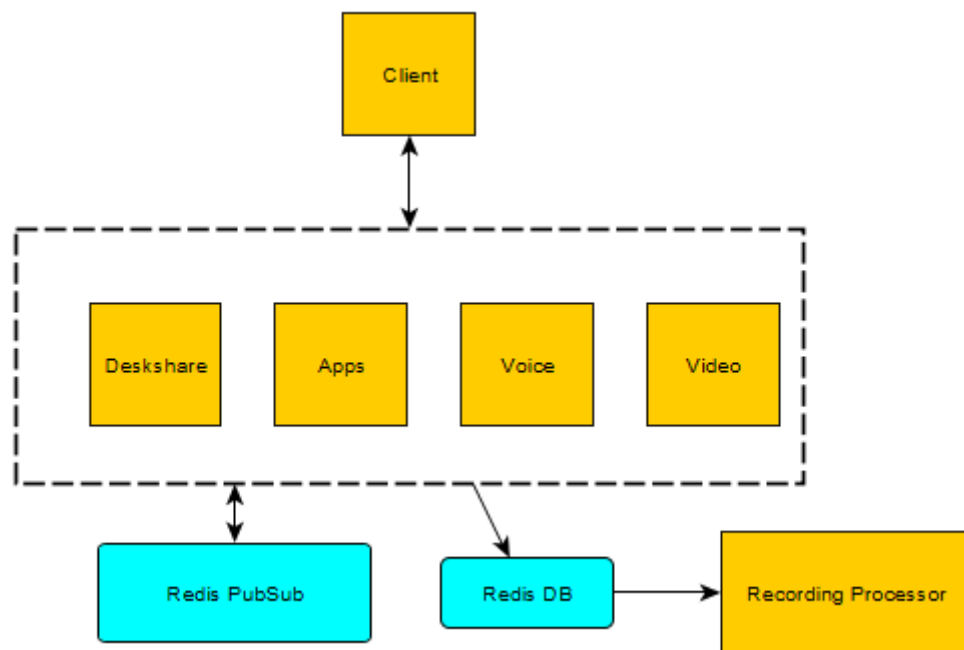
### **Redis Db :**

Lorsqu'une réunion est enregistrée, tous les événements sont stockés dans la base de données redis. Lorsque la réunion est terminée, le Processeur d'enregistrement prendra en charge tous les événements enregistrés ainsi que les différents fichiers bruts (PDF, WAV, FLV).

### **Red5 Apps(Deskshare,Apps,voice,video) :**

Les applications Red5 sont des applications différentes qui assurent le streaming des médias lors de la réunion et transmettent les messages entre les clients et Apps Akka.

L'Apps est la principale application BigBlueButton qui gère les utilisateurs, le chat, le tableau blanc, les informations de présentation partagées par tous les utilisateurs d'une réunion. L'application Deskshare permet au présentateur de partager le bureau. L'application Voice permet à l'utilisateur de intervenir oralement dans la conférence vocale à l'aide d'un casque ou de se joindre en mode écoute seule. L'application Vidéo permet à un utilisateur de partager sa webcam avec les autres utilisateurs de la conférence.



*Figure 1.3 architecture de 5 apps*

### **Apps Akka**

BigBlueButton Apps est l'application principale qui rassemble les différentes applications pour fournir une collaboration en temps réel dans la réunion. Il fournit également la liste des utilisateurs, chat, tableau blanc, présentations lors d'une réunion.

Vous trouverez ci-dessous un schéma des différents composants d'Apps Akka.

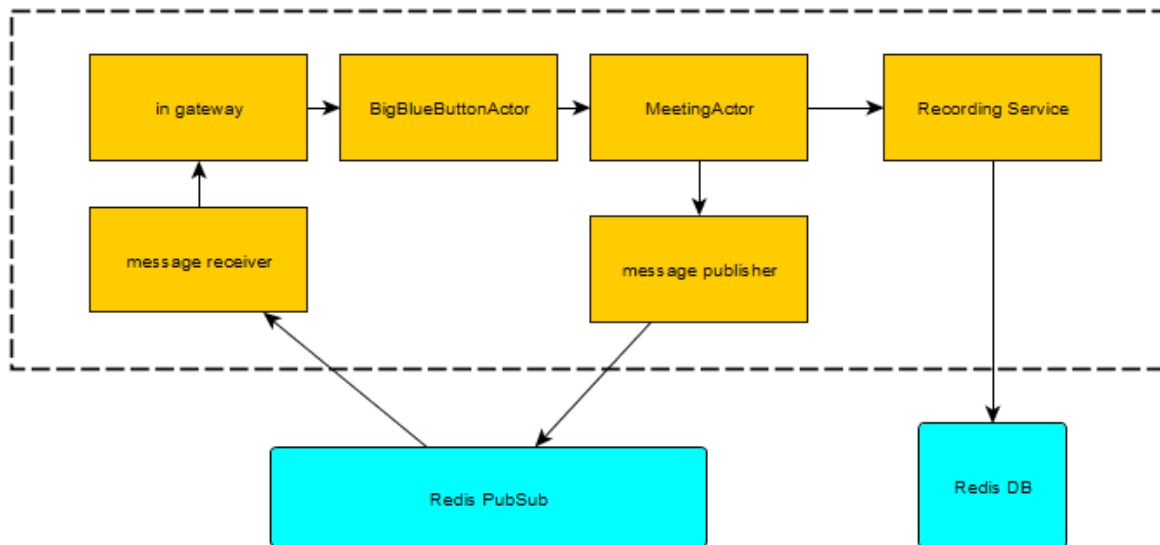


Figure 1.4 Le schéma des différents composants d'Apps Akka

### FsEslAkka :

Nous avons extrait le composant qui s'intègre avec FreeSWITCH dans sa propre application. Cela permet à d'autres utilisateurs du système de conférence vocale autres que FreeSWITCH de créer facilement leur propre intégration. La communication entre apps et fsesl utilise des messages via redis pubsub.

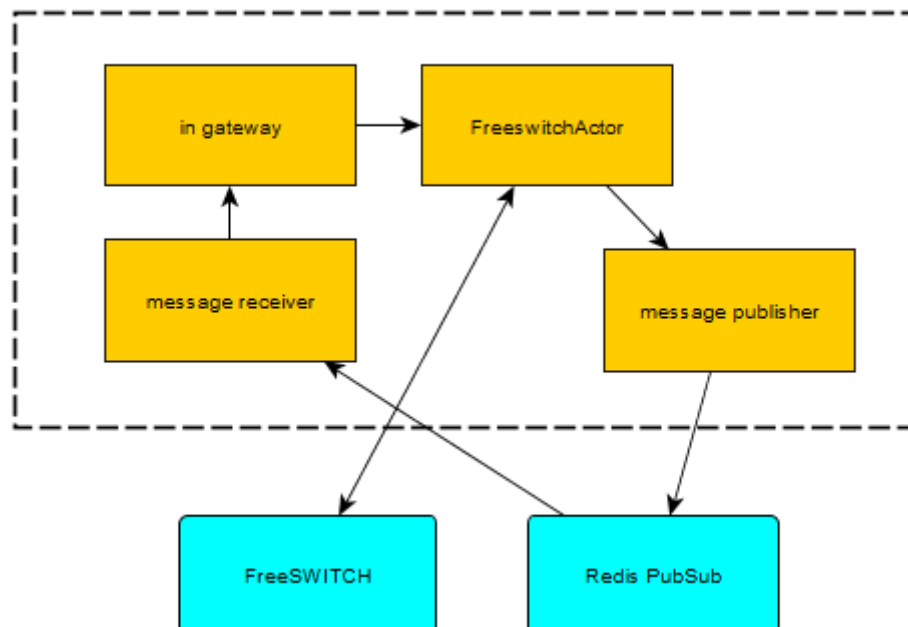


Figure 1.5 Le schéma des **FsEslAkka**

### FreeSWITCH

FreeSWITCH fournit la fonction de conférence vocale dans BigBlueButton. Les utilisateurs peuvent se joindre à la conférence vocale via le casque. Les utilisateurs, utilisant Google

Chrome ou Mozilla Firefox, sont en mesure de profiter d'une qualité audio supérieure en se connectant à l'aide de WebRTC. FreeSWITCH peut également être intégré avec les fournisseurs de VOIP afin que les utilisateurs qui ne sont pas en mesure de se joindre à l'aide du casque seront en mesure d'appeler à l'aide de leur propre téléphone.

### **3.1.6. Différentes versions de BigBlueButton**

Version	Release date
0.32	1 September 2008
0.4	12 June 2009
0.5	21 July 2009
0.60	12 August 2009
0.61	15 September 2009
0.62	11 November 2009
0.63	25 January 2010
0.64	3 April 2010
0.70	15 July 2010
0.71	9 November 2010
0.71a	13 January 2011
0.8-beta1	12 September 2011
0.8-beta2	12 September 2011
0.8-beta3	1 December 2011

Version	Release date
0.8-beta4	6 March 2012
0.80	18 June 2012
0.81	7 November 2013 <sup>1</sup>
0.90-beta	15 October 2014
0.90-RC	19 April 2015
0.90	30 April 2015
1.0-beta	6 October 2015
1.0	17 May 2016 <sup>6</sup>

### 3.1.7. Les limites de la plateforme BigBlueButton

- Adobe Flash Player est trop lent et il est connu pour les lags et les bugs.
- problème temps de réponse au niveau de fonctionnalité file sharing

## 3.2. La technologie VOIP

La téléphonie sur IP (VoIP) est la transmission de la voix en utilisant le protocole IP. Le support utilisé peut être le réseau public Internet ou un réseau privé.

### 3.2.1. Les protocoles de la technologie VOIP

Les principaux protocoles permettant l'établissement de la connexion

- H.323
- SIP
- IAX (Asterisk)
- MGCP (Media Gateway control Protocol)
- SCCP (propriétaire Cisco Systems)

Les principaux protocoles permettant le transport de la voix

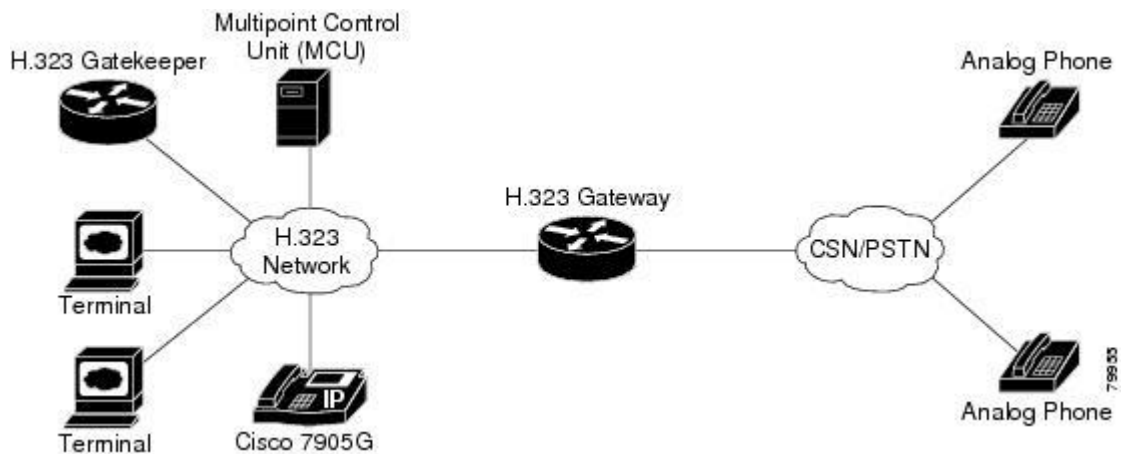


- RTP
- RTCP

Les protocoles Secondaires sont :

- DHCP : attribution des adresses
- IP, DNS, TFTP pour la configuration et la mise à jour
- DNS pour les services d'annuaire et de localisation
- HTTP pour l'administration

### 3.2.2. Le protocole H.323

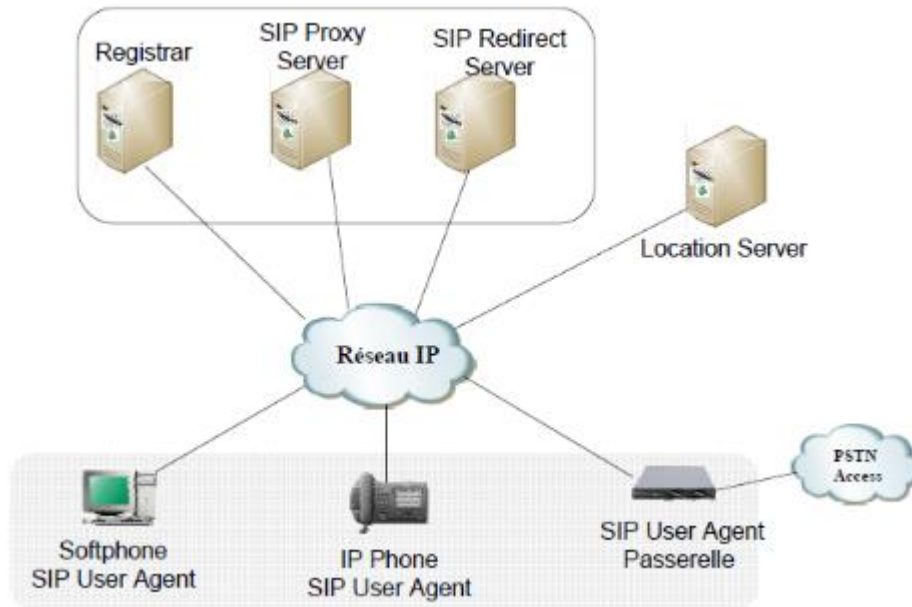


*Figure 1.6 architecture h 323*

Le protocole Standard H.323 fournit les services pour le transfert de l'audio, de la vidéo ou de données à travers des réseaux IP. Les composants principaux sont:

- Le portier (gatekeeper): Il est responsable de la translation entre un numéro de téléphone et une adresse IP et fournit les mécanismes pour l'authentification et l'enregistrement des terminaux.
- La passerelle H.323 (Gateway) : La passerelle permet de faire l'interface entre différents réseaux .Elle permet à un terminal H.323 de pouvoir appeler un terminal qui est sur un réseau différent
- Le Pont de conférence (MCU : MultiPoint Control Unit) : Le MCU est une station sur le réseau qui fournit les possibilités pour trois terminaux ou plus de participer à une conférence multipoints.
- Le Terminal H.323: représente l'entité qui se trouve à chaque extrémité d'une connexion.

### 3.2.3. Le protocole SIP ( Session Initiation Protocol )



*Figure 1.7 architecture SIP*

SIP est un protocole de signalisation défini par l'IETF (Internet Engineering Task Force) en 1999, permettant l'établissement et la modification de sessions multimédias (voix, vidéo, données). Il a été étendu afin de supporter de nombreux services tels que la messagerie instantanée, le transfert d'appel, la conférence et les services complémentaires de téléphonie.

### 3.2.4. Les problèmes de voip

l'interception des appels et le détournement du service (n'importe qui pouvant alors téléphoner via l'infrastructure de l'entreprise). Ou encore l'usurpation de l'identité de l'appelant (se faire passer pour le patron d'un grand groupe auprès de quelques employés importants peu s'avérer particulièrement tentant).

## 4. Solution proposées

suite au faiblesse que nous avons listés on propose une solution Visio conférence qui exploite les nouvelle technologie dans le domaine audio visuelle pour assuré une communication audio vidéo plus flexible et facile a l'utilisation

## **5. Conclusion**

On a présenté dans ce chapitre une étude du préalable suivie d'une analyse de l'existant qui comporte la plateforme Bigbluebutton et la technologie de VOIP. Puis on a discuté les problématiques posées par ces deux technologies et la solution qu'on peut apporter pour résoudre ces problèmes et essayer d'ajouter une idée innovante par rapport à l'existant.

## **Chapitre 2 : Analyse des besoins et spécification**

## **1. Introduction :**

L'analyse et la spécification des besoins représentent la première phase du cycle de développement d'un logiciel. Elle sert à identifier les acteurs réactifs du système et leur associer chacun l'ensemble d'actions avec lesquelles il intervient dans l'objectif de donner un résultat optimal et satisfaisant au client.

## **2. Spécification des besoins :**

### **2.1 Les besoins fonctionnels :**

Les modules proposés par notre application se résument en quatre actions majeures :

- Un module chat :

Le module Chat permet aux participants de tenir une conversation en temps réel qui se déroule dans un « salon ».

Le chat est un moyen très pratique d'échanger les différents points de vue de chacun à propos d'un sujet.

- Un module pour WebRTC Audio et Vidéo :

Les utilisateurs des navigateurs Chrome et FireFox bénéficieront d'un audio WebRTC de haute qualité et à faible latence. (Les utilisateurs d'autres navigateurs utiliseront en toute transparence l'audio Flash)

Aussi Les utilisateurs bénéficieront d'un vidéoWebRTC pour pouvoir partager leur webcam en même temps.

- Un module pour record :

Ce module permet l'enregistrement de tous les événements et données multimédias générés pendant une session de conférence.

- Un module pour partage de bureau

Ce module permet de diffuser votre ordinateur de bureau pour que tous les utilisateurs puissent le voir.

- Un module pour canvas :

Ce module est utilisé pour dessiner des graphiques, il a plusieurs méthodes pour tracer des chemins, des boîtes, des cercles, du texte et ajouter des images.

- Un module pour la présentation des documents :

Ce module Vous permet de télécharger une présentation PDF ou un document MS Office.

Il maintient tout le monde en synchronisation avec votre diapositive actuelle, zoom, panoramique, annotations et pointeur de la souris.

## **2.2 Les besoins non fonctionnels :**

Les besoins non fonctionnels décrivent toutes les contraintes auxquelles est soumis le système pour sa réalisation et son bon fonctionnement.

- Ergonomie et souplesse

L'application doit offrir une interface conviviale et ergonomique exploitable par l'utilisateur en envisageant toutes les interactions possibles à l'écran du support tenu.

- Rapidité

L'application doit optimiser les traitements pour avoir un temps d'exécution raisonnable.

- Efficacité

L'application doit être fonctionnelle indépendamment de toutes circonstances pouvant entourer l'utilisateur.

- Maintenabilité et scalabilité

Le code de l'application doit être lisible et compréhensible afin d'assurer son état évolutif et extensible par rapport aux besoins du marché.

### **3. Conclusion**

Ce chapitre nous a permis de couvrir les différents besoins fonctionnels et non fonctionnels de notre système. Nous essayerons dans le chapitre qui suit de concevoir clairement l'architecture de notre système avec les étapes de réalisation.

## **Chapitre 3 :**

## **Réalisation**



## 1. Introduction

Ce chapitre sera consacré à la phase de réalisation. La présentation de l'environnement de réalisation fera l'objet de la première section. Une deuxième section sera consacrée aux détails de la réalisation et de l'implémentation.

## 2. Installation et configuration de la plateforme BigBlueButton

### 2.1. BigBlueButton version flash

#### ❖ Pré requis

- Ubuntu 10.04 32 bits ou 64bits
- 4 Go de mémoire ram
- Processeur 2.6 Ghz
- Les ports 80, 1935, 9123 accessibles
- Port 80 ne doit pas être utilisé par une autre application
- 500 G d'espace libre sur le disque
- Minimum de 100 Mbits/sec de bande passant
- On doit l'installer sur une machine physique.

Pour l'installation de la plateforme BigBlueButton dans une machine virtuelle (VirtualBox, VMWare, QEMU), il faut que l'ordinateur ait une capacité assez importante (3Go de RAM ou plus, et un CPU puissant et un disque dur de grande capacité).

La plateforme BigBlueButton recommande à consacrer 1 Go de RAM au moins, pour cela le Pc doit avoir un processeur dual-core 2 GHz au moins. Une installation de la plateforme BigBlueButton complète nécessite 2 Go.

**NB** : Il est à noter que notre Ubuntu 10.04 se trouve sur une machine virtuelle (VirtualBox).

Avant l'opération d'installation de la plateforme BigBlueButton, il faut procéder à quelques vérifications.

#### a) Vérification de la distribution :

```
root@bigbluebutton:~# cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=10.04
DISTRIB_CODENAME=lucid
DISTRIB_DESCRIPTION="Ubuntu 10.04 LTS"
root@bigbluebutton:~#
```

Cela signifie que, nous avons ubuntu 10.04

#### b) Vérification de la langue

```
root@bigbluebutton:~# cat /etc/default/locale
LANG="fr_FR.UTF-8"
root@bigbluebutton:~#
```

Cela signifie que la localisation du serveur est fr\_US.UTF-8. Cependant il va falloir la modifier en anglais:

- **Installation du paquet**

```
root@bigbluebutton:~# apt-get install language-pack-en
```

- **Modification de la langue**

```
root@bigbluebutton:~# update-locale LANG=en_US.UTF-8
root@bigbluebutton:~#
```

- **Ré vérification**

```
root@bigbluebutton:~# cat /etc/default/locale
LANG=en_US.UTF-8
root@bigbluebutton:~#
```

Cela veut dire que la langue à été bien modifiée.

- c) **Vérifions si le port 80 n'est pas utilisé par d'autres applications**

```
root@bigbluebutton:~# lsof -i:80
root@bigbluebutton:~#
```

Cela veut dire que le port 80 n'est pas encore utilisé par une autre application.

- d) **Mise à jour du serveur**

```
root@bigbluebutton:~# apt-get update
root@bigbluebutton:~# apt-get dist-upgrade
```

- e) **Ajout des dépôts BigBlueButton**

- Récupérer et ajouter la clé de référentiel BigBlueButton

```
root@bigbluebutton:~# wget http://ubuntu.bigbluebutton.org/bigbluebutton.asc -O- | sudo apt-key a
dd -
```

- Ajouter les dépôts de BigBlueButton à votre liste de dépôt dans le fichier /etc/apt/sources.list

```
root@bigbluebutton:~# echo "deb http://ubuntu.bigbluebutton.org/lucid_dev_08/ bigbluebutton-lucid
main" | sudo tee /etc/apt/sources.list.d/bigbluebutton.list
deb http://ubuntu.bigbluebutton.org/lucid_dev_08/ bigbluebutton-lucid main
root@bigbluebutton:~#
```

```
root@bigbluebutton:~# echo "deb http://us.archive.ubuntu.com/ubuntu/ lucid multiverse" | sudo tee
-a /etc/apt/sources.list
deb http://us.archive.ubuntu.com/ubuntu/ lucid multiverse
root@bigbluebutton:~#
```

Après avoir apporté les modifications ci-dessus, faites un dist-upgrade pour assurer que les nouveaux paquets sont pris en comptes et le serveur est à jour avant d'installer BigBlueButton

#### f) Mise à jour de nouveau paquet

```
root@bigbluebutton:~# apt-get update
```

```
root@bigbluebutton:~# apt-get dist-upgrade
```

#### g) Installation de Ruby

On vérifie s'il est installé où pas, la version doit correspondre à 1.9.2p290.

```
root@bigbluebutton:~# dpkg -l | grep ruby
root@bigbluebutton:~#
```

Il n'est pas encore installé, alors nous allons l'installer de façon suivante :

- Installation des dépendances

```
root@bigbluebutton:~# apt-get install zlib1g-dev libssl-dev libreadline5-dev libyaml-dev build-es
sential bison checkinstall libffi5 gcc checkinstall libreadline5 libyaml-0-2
```

- Création de script d'installation de ruby
- Création du script

```
root@bigbluebutton:~# touch inst-ruby.sh
root@bigbluebutton:~# vim.tiny inst-ruby.sh
```

```
#!/bin/bash
cd /tmp
wget http://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.2-p290.tar.gz
tar xvfz ruby-1.9.2-p290.tar.gz
cd ruby-1.9.2-p290
./configure --prefix=/usr\
            --program-suffix=1.9.2\
            --with-ruby-version=1.9.2\
            --disable-install-doc
make
sudo checkinstall -D -y\
                --fstrans=no\
                --nodoc\
                --pkgname='ruby1.9.2'\
                --pkgversion='1.9.2-p290'\
                --provides='ruby'\
                --requires='libc6,libffi5,libgdbm3,libncurses5,libreadline5,openssl,libyaml-0-2, zlib1g'\
                --maintainer=brendan.ribera@gmail.com
sudo update-alternatives --install /usr/bin/ruby ruby /usr/bin/ruby1.9.2 500 \
                        --slave /usr/bin/ri ri /usr/bin/ri1.9.2 \
                        --slave /usr/bin/irb irb /usr/bin/irb1.9.2 \
                        --slave /usr/bin/erb erb /usr/bin/erb1.9.2 \
                        --slave /usr/bin/rdoc rdoc /usr/bin/rdoc1.9.2
sudo update-alternatives --install /usr/bin/gem gem /usr/bin/gem1.9.2 500
```

- Attribution du droit d'exécution et lancement du script

```
root@bigbluebutton:~# chmod +x inst-ruby.sh
root@bigbluebutton:~# ./inst-ruby.sh
```

- Vérification de ruby

```
root@bigbluebutton:~# ruby -v
ruby 1.9.2p290 (2011-07-09 revision 32553) [i686-linux]
root@bigbluebutton:~#
```

## 2.2. On constate que ruby est bien installé.

- Vérification de la version de ruby

```
root@bigbluebutton:~# gem -v
1.3.7
root@bigbluebutton:~#
```

- Test si gem fonctionne normalement

```
root@bigbluebutton:~# gem install hello
Successfully installed hello-0.0.1
1 gem installed
Installing ri documentation for hello-0.0.1...
Installing RDoc documentation for hello-0.0.1...
root@bigbluebutton:~#
```

## h) Installation de FreeSWITCH pour l'enregistrement de sessions

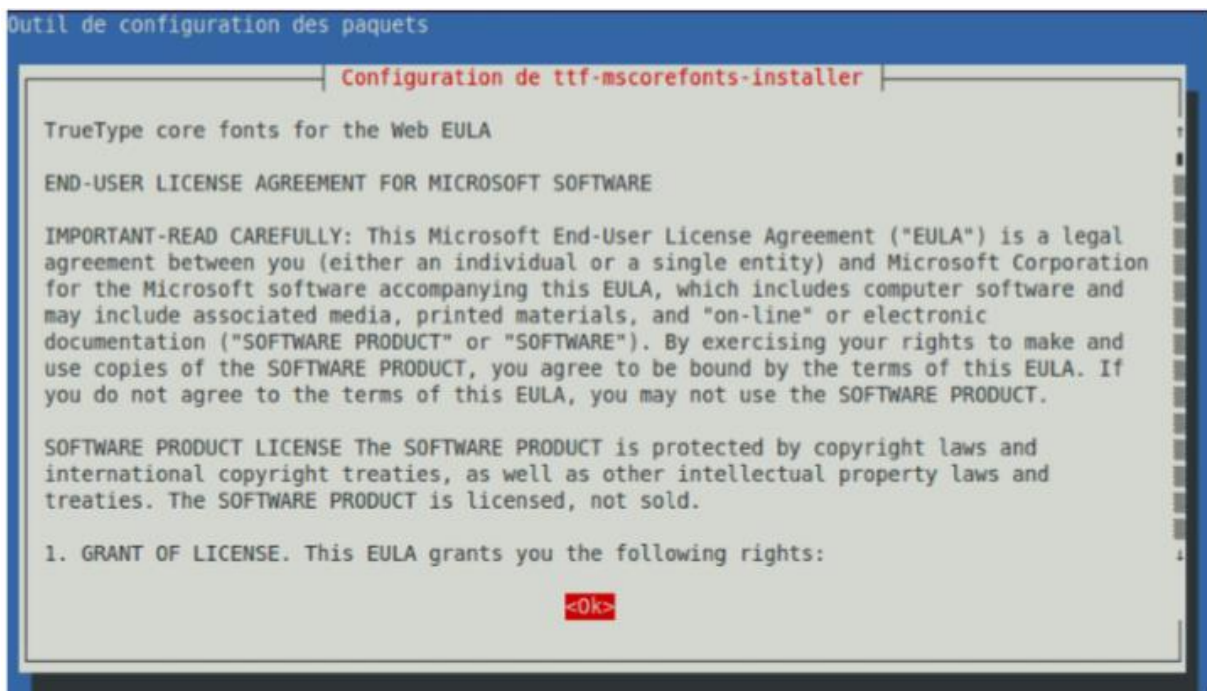
**NB :** Cette installation est facultative car la plateforme BigBlueButton 0.80 installe et configure FreeSWITCH 1.0.7

```
root@bigbluebutton:~# apt-get install bbb-freeswitch-config
```

## i) Installation de la plateforme BigBlueButton

```
root@bigbluebutton:~# apt-get install bigbluebutton
```

## j) Acceptation de la licence



- Installation de la démo de l'API pour essayer l'interactive BigBlueButton



```
root@bigbluebutton:~# apt-get install bbb-demo
```

k) Redémarrage du serveur

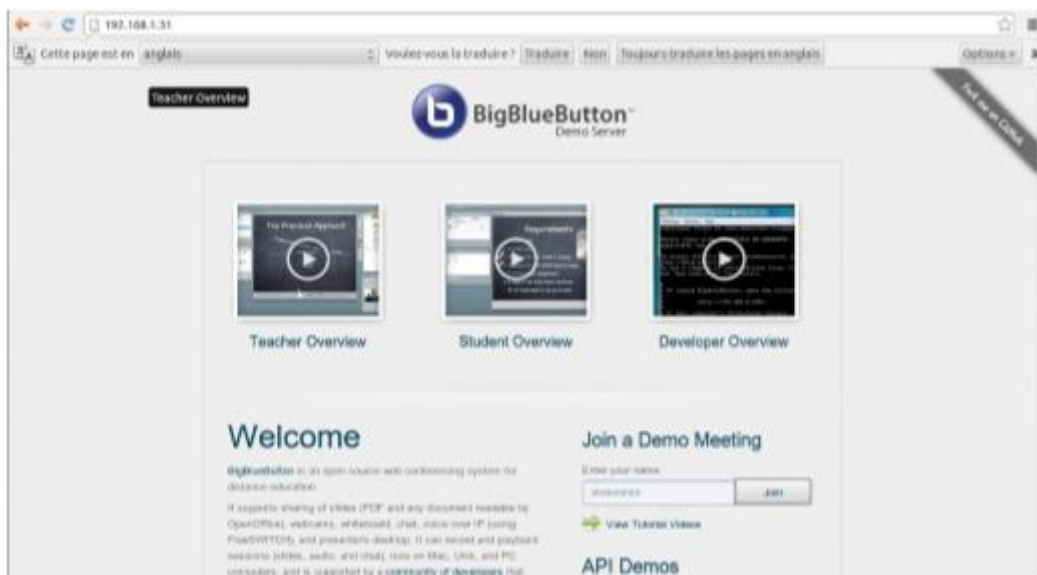
```
root@bigbluebutton:~# bbb-conf --clean
```

```
root@bigbluebutton:~# bbb-conf --check
```

- Modification de l'adresse IP du serveur

```
root@bigbluebutton:~# bbb-conf --setip 192.168.1.31
Assigning 192.168.1.31 for testing for firewall in /var/www/bigbluebutton/client/conf/config.xml
Assigning 192.168.1.31 for rtmp:// in /var/www/bigbluebutton/client/conf/config.xml
Assigning 192.168.1.31 for servername in /etc/nginx/sites-available/bigbluebutton
Assigning 192.168.1.31 for http:// in /var/www/bigbluebutton/client/conf/config.xml
```

- Interface graphique





**NB** : Il faut installer sur les navigateurs de participants des plugins tels que : adobe flash players pour l'accès à l'interface de BigBlueButton et java pour le partage de bureau.

### 1) Quelques Commandes BigBlueButton

- Modifier l'emplacement d'hôte du serveur BigBlueButton

Pour changer le nom\_serveur des fichiers de configuration Nginx (et d'autres BigBlueButton). Le nom\_serveur peut être une adresse IP (par exemple, 192.168.1.13) ou une URL (comme bigbluebutton.ec2lt.sn)

```
bbb-conf --setip bigbluebutton.ec2lt.sn
```

Si elle ne fonctionne pas, essayer un redémarrage propre du système BigBlueButton :

```
bbb-conf --propre
```

- Changer le port d'écoute du Bureau

Pour changer le port d'écoute de du serveur web Nginx

```
bbb-conf --setip bigbluebutton.ec2lt.sn : 81
```

Puis Redémarrer le serveur

- Installation le partage de bureau

```
apt-get install bbb-apps-Deskshare
```

```
root@bigbluebutton:~# apt-get install bbb-apps-Deskshare
```

### 2.3. La plateforme BigBlueButton version HTML5

- Installation PPA pour LibreOffice 4.3

```
$ sudo apt-get install software-properties-common
```

```
$ sudo add-apt-repository ppa:libreoffice/libreoffice-4-3
```

- Installation de key pour BigBlueButton

```
$ wget http://ubuntu.bigbluebutton.org/bigbluebutton.asc -O- | sudo apt-key add -
```

```
$ echo "deb http://ubuntu.bigbluebutton.org/trusty-090/ bigbluebutton-trusty main" |  
sudo tee /etc/apt/sources.list.d/bigbluebutton.list
```

```
$ sudo apt-get update
```



- Install ffmpeg

Pour installer ffmpegil faut créer un fichier nommer install-ffmpeg.sh et copier ce

```
sudo apt-get install build-essential git-core checkinstall yasm texi2html
libvorbis-dev libx11-dev libvpx-dev libxfixes-dev zlib1g-dev pkg-config netcat
libncurses5-dev

FFMPEG_VERSION=2.3.3

cd /usr/local/src
if [ ! -d "/usr/local/src/ffmpeg-${FFMPEG_VERSION}" ]; then
    sudo wget "http://ffmpeg.org/releases/ffmpeg-
${FFMPEG_VERSION}.tar.bz2"
    sudo tar -xjf "ffmpeg-${FFMPEG_VERSION}.tar.bz2"
fi

cd "ffmpeg-${FFMPEG_VERSION}"
sudo ./configure --enable-version3 --enable-postproc --enable-libvorbis --
enable-libvpx
sudo make
sudo checkinstall --pkgname=ffmpeg --pkgversion="5:${FFMPEG_VERSION}"
--backup=no --deldoc=yes --default
```

Ensuite, rendez le script exécutable et exécutez-le.

```
$ chmod +x install-ffmpeg.sh
$ ./install-ffmpeg.sh
```

- Installer BigBlueButton

```
$ sudo apt-get install bigbluebutton
```

```
$ sudo apt-get install bbb-demo
```

- Installer Client Self-Check

```
$ sudo apt-get install bbb-check
```

### 3. Les technologies utilisées

Aujourd'hui sur 7 milliards d'habitants, 3,2 milliards sont constamment connectés à internet. L'usage des Smartphones et des tablettes est en croissance considérable, actuellement, on peut faire quasiment tout ce que l'on veut en utilisant des appareils smartphones. De façon inévitable, le besoin d'une meilleure communication émerge, et c'est là que le WebRTC intervient. Des analyses suggèrent que d'ici à la fin 2018, les utilisateurs individuels du WebRTC atteindront 1 milliard; et les PC, Smartphones, et tablettes pouvant être connectés au WebRTC atteindront presque les 4,7 milliards.

#### 3.1 La technologie webRTC



La technologie WebRTC (Web Real-Time Communication, littéralement « communication en temps réel pour le Web ») est une interface de programmation (API) JavaScript développée au sein du W3C et de l'IETF. Ce système facilite les applications de navigateur à navigateur pour les appels vocaux, le chat vidéo et le partage de fichiers

VP8 est le codec actuellement pris en charge pour WebRTC.

##### 3.1.1 Architecture

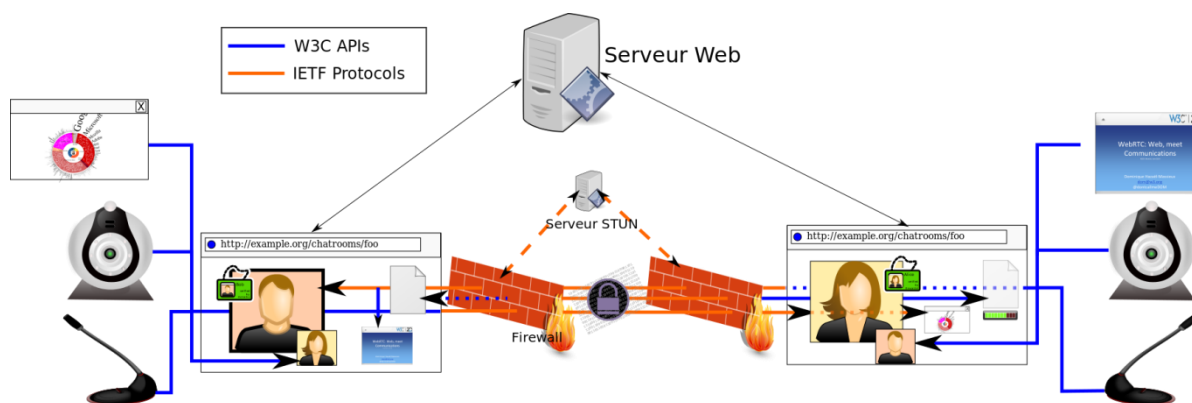


Figure 3.1 L'architecture webRTC

L'API repose sur une architecture triangulaire puis pair à pair dans laquelle un serveur central est utilisé pour mettre en relation deux pairs désirant échanger des flux de médias ou de données qui s'échangent.

### 3.1.2 Principe du webRTC

- Le modèle Peer to Peer



Le modèle pair à pair (en anglais peer-to-peer, souvent abrégé « P2P ») est un modèle de réseau informatique proche du modèle client-serveur où chaque client est aussi un serveur. Les termes « pair », « nœud », et « utilisateur » sont généralement utilisés pour désigner les entités composant un réseau P2P

- utilité

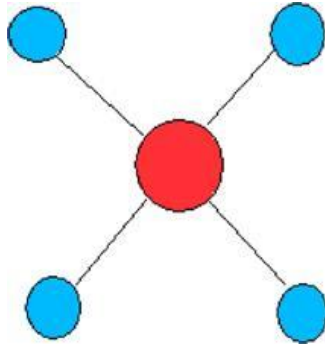
Les systèmes pair-à-pair permettent à plusieurs ordinateurs de communiquer via un réseau, en y partageant simplement des objets ou particulièrement des fichiers le plus souvent, mais également des flux multimédia continus (streaming), le calcul réparti, ou même les services comme la téléphonie sur IP

- Architectures P2P

Les serveurs pair-à-pair fonctionnent dans la quasi-totalité des cas en **mode synchrone** : le transfert d'information est limité aux éléments connectés en même temps que le réseau. Ils peuvent utiliser le protocole **TCP** comme couche de transport des données. En revanche,

certaines utilisations comme le continu (streaming) nécessitent l'emploi d'un protocole plus léger et plus rapide, comme **UDP**, moins fiable.

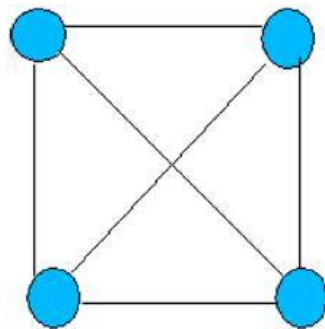
- Architectures Centralisé



Dans une architecture centralisée comme le montre le schéma, le contenu, services, ressources et les fonctions de recherche, localisation, indexation et ainsi que la publication sont centralisés sur un seul serveur.

Exemples : Napster, Audiogalaxy, eDonkey2000

- Architectures Décentralisé



Dans ce type d'architecture, tout est décentralisé. Chaque nœud est à la fois client et serveur.

Exemple : CAN, Chord, Freenet, GNUnet, Tapestry, Pastry et Symphony

- Avantages et Inconvénients du P2P
  - Les avantages du P2P
    1. Les communications sont directes
    2. Décentralisation
    3. Le réseau est faiblement couplé
    4. Possibilité de créer des groupes

- Les inconvénients du P2P
  1. Pas de QoS
  2. problèmes de sécurité

### 3.1.3 Description générale de la norme

L'architecture de l'API WebRTC repose sur une construction triangulaire impliquant un serveur et deux pairs. Les deux navigateurs téléchargent depuis un serveur une application JavaScript vers leur contexte local

- Les étapes d'une connexion entre deux clients utilisant WebRTC

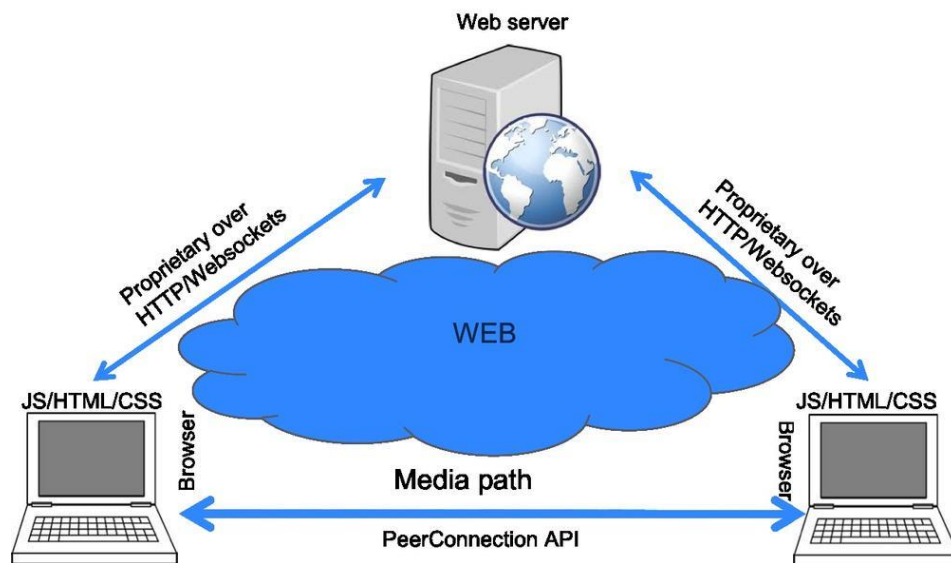


Figure 3.2 Les étapes d'une connexion entre deux clients utilisant WebRTC

1-Afin d'établir une connexion utilisant le standard WebRTC, les navigateurs A et B doivent être connectés simultanément à la page du service et télécharger la page HTML ainsi que le code JavaScript permettant de maintenir la connexion ouverte par HTTPS ou socket.

2-Lorsque le navigateur A souhaite établir la connexion avec B, l'API instancie un objet **PeerConnection** qui, une fois créé, permet d'établir des flux de médias ou de données.

3-Une fois cet objet PeerConnection créé par A, le navigateur envoie au serveur un paquet contenant les informations sur les médias partagés ainsi qu'une empreinte liant la connexion à A.

4-Le serveur va décoder ce paquet et identifier qu'il s'agit d'une communication à destination de B et enverra donc un signal à B. B est notifié du souhait de A d'établir une connexion et accepte ou non sa requête.

5-Si elle est acceptée, le même processus a lieu entre B et A cette fois afin d'établir la connexion bidirectionnelle. Une fois celle-ci établie, les flux de médias ou de données peuvent être ajoutés à la connexion librement.

- cas particulier streaming vidéo

Dans le cadre par exemple d'un streaming vidéo en peer-to-peer entre navigateurs,

1-l'utilisateur télécharge depuis un serveur les métadonnées de la vidéo qu'il souhaite regarder ainsi qu'une liste de paires disponibles et ayant tout ou partie de la vidéo.

2-L'établissement d'une connexion avec les paires permet, par le flux de données, le téléchargement des morceaux de la vidéo qui sont réassemblés après vérification de leur intégrité puis lancement de la vidéo dans un lecteur HTML5

### 3.1.4 Les Composants du webRTC

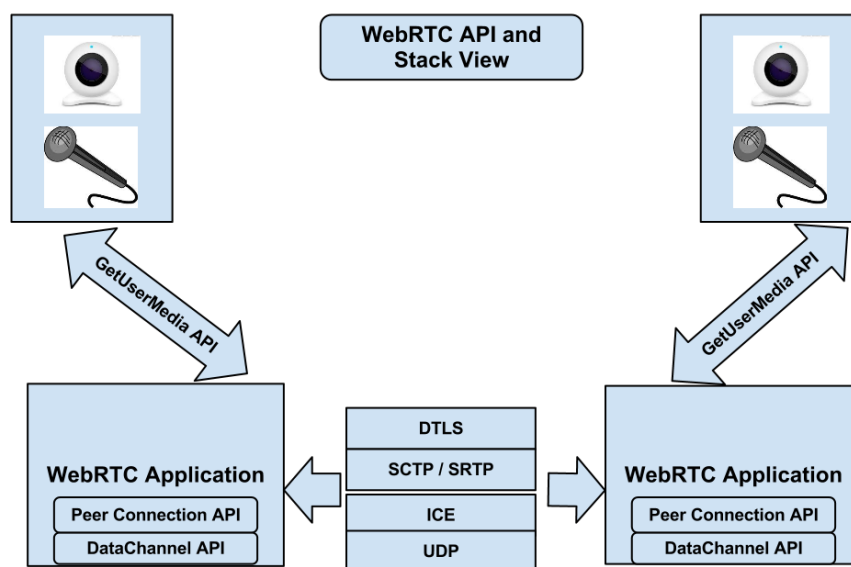


Figure 3.3 les différentes Composants du webRTC

- PeerConnection

L'API `RTCPeerConnection` représente le lien établi avec un navigateur distant, reposant sur le protocole UDP (*habituellement une autre instance de la même application JavaScript s'exécutant sur le client distant*).

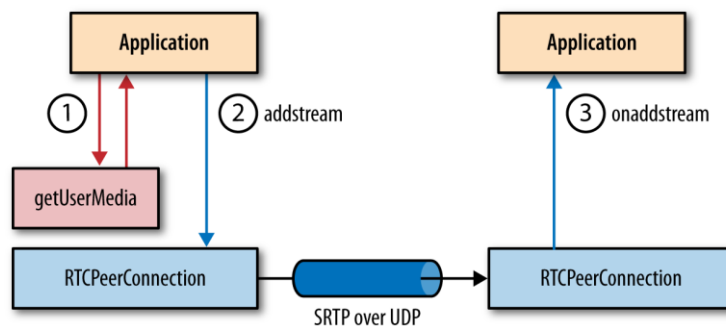


Figure 3.4 L'architecture de l'API RTCPeerconnection

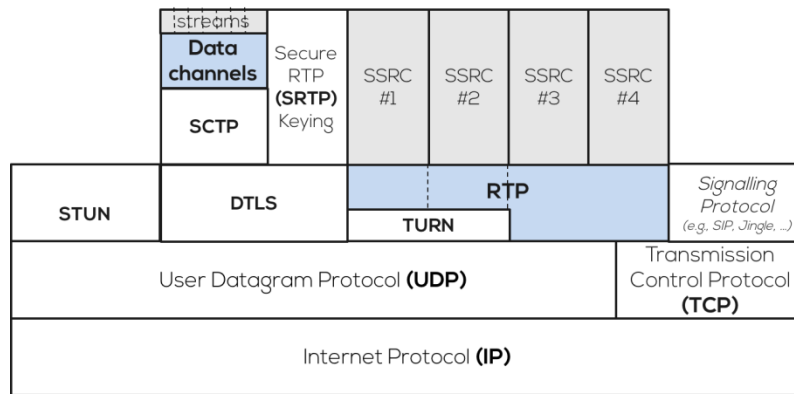
Afin d'établir cette connexion en pair à pair, il est nécessaire de s'appuyer sur un canal de communication établi par un serveur web et utilisant par exemple un objet **XMLHttpRequest** ou une **WebSocket**.

Pour obtenir une connexion, l'une des paires doit obtenir les informations locales (telles que les protocoles supportés pour l'audio ou la vidéo). Cette étape est permise par l'API via **Session Description Protocol**. Ce processus utilisant SDP permet la négociation à la fois pour **RTP** (transport de médias) et pour **SCTP** (permettant le transport de données).

Afin d'assurer la continuité de la connexion en cas de conversion d'adresse par NAT et éviter le blocage par les pare-feux (notamment en entreprise), l'objet PeerConnection utilise les protocoles UDP, STUN et ICE32. Une fois cette connexion en pair à pair a été établie, chaque partie peut installer des MediaStreams ou DataStreams l'utilisant.

- **DataStream**

L'API DATA channels offre un moyen d'échange de données génériques bidirectionnel et pair à pair. Cette composante de webRTC permet l'échange de données telles que des images ou un texte.



Note: \*RTP can be sent over UDP or TCP. Similarly, signalling protocols can be designed to transmit over UDP or TCP.

Figure 3.5 le paquet ip

Ces canaux de données sont créés entre paires en utilisant l'objet `PeerConnection`. Ces données autres que les flux médias sont échangées via le protocole **SCTP**, lui-même encapsulé dans **DTLS**. Cette solution permet au flux de données d'être intégré dans le même paquet que les flux de médias et donc de partager le même numéro de port pour les échanges.

Afin d'assurer la confidentialité et l'authenticité des paquets **SCTP** échangés, chaque flux repose sur le protocole **DTLS**.

- Flux Media

La Structure de la pile de protocoles est utilisée par WebRTC dans un échange de médias. Un `MediaStream` est une représentation d'un flux de données spécifique audio ou vidéo

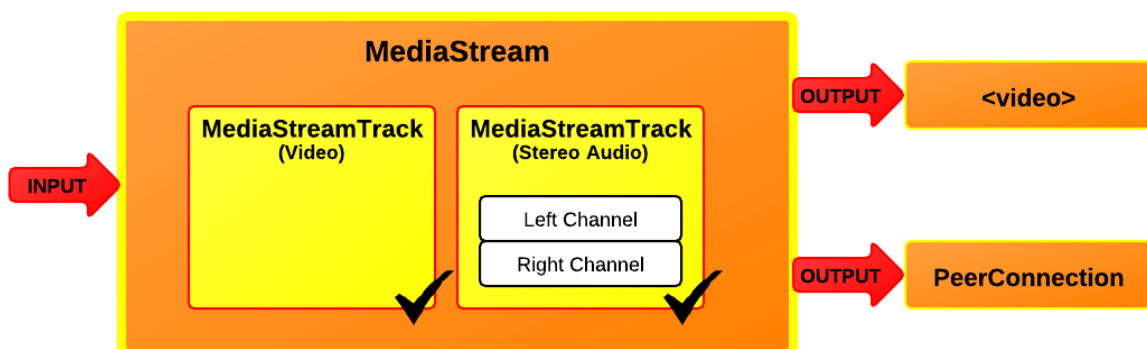


Figure 3.6 structure d'un Flux Media

Un `MediaStream` peut être local ou distant. L'API `MediaStream` gère les flux audio et vidéo et indique à l'application qu'elle doit donner accès à la caméra, aux haut-parleurs et au microphone. Les flux médias sont transportés par le biais du protocole **RTP**, utilisable sur tout



protocole de transport implémentant une abstraction de datagram (UDP par exemple). La confidentialité, l'authentification des messages et la protection contre les répétitions sont apportées par l'utilisation sécurisée de RTP, SRTP. La gestion des clés pour SRTP est assurée par DTLS et donc le flux de données. Il est donc impossible d'avoir un flux média indépendant d'un flux de données là où l'inverse est envisageable. Il est possible également d'associer plusieurs flux médias sur une même connexion SRTP qui utiliseront des ressources médias différentes ou non. Dans ce cas, les sources de chaque flux sont clairement identifiées comme des SSRC.

### 3.1.5 Problèmes posés par WebRTC et solutions potentielles

- Sécurité des applications

Plusieurs problèmes de sécurité se posent lors de l'utilisation de WebRTC :

- L'application JavaScript (le code utilisant WebRTC) peut être téléchargée depuis n'importe quel site sans consentement de l'utilisateur.
- L'exploitation de ressources média locales (par exemple les caméras et microphones) doit requérir l'approbation de l'utilisateur.
- La confidentialité et l'authentification doivent être garanties dans tout échange pour éviter les attaques telles que l'attaque de l'homme du milieu.

Si certains de ces problèmes sont inhérents à toute communication sur l'Internet, d'autres problèmes ont été résolus par l'implémentation de WebRTC. Ainsi les échanges de médias sont sécurisés par le protocole SRTP.

#### Gestion de la perte de paquets

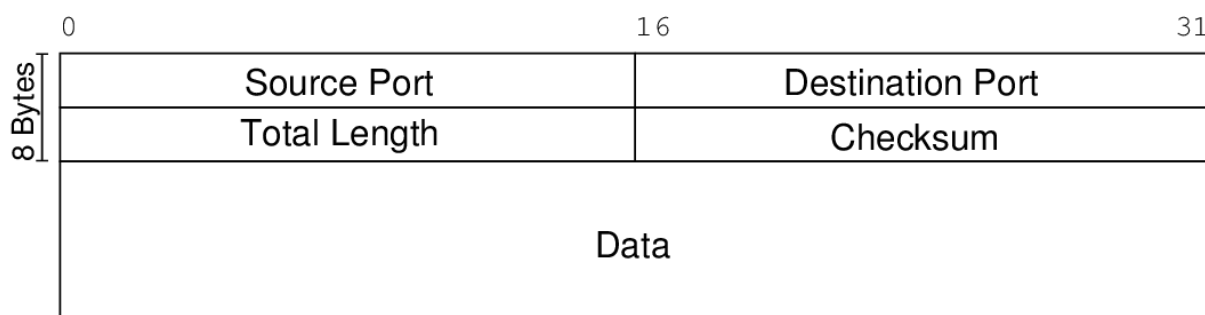


Figure 3.7 le paquet UDP

Le protocole UDP étant déconnecté et n'utilisant pas de système de vérification de réception des paquets (contrairement au protocole TCP par exemple), la perte de paquets est un problème pouvant se poser lors des transmissions en pair à pair de flux médias.

Deux méthodes se présentaient afin de limiter la perte de paquets dus aux problèmes de réseaux :

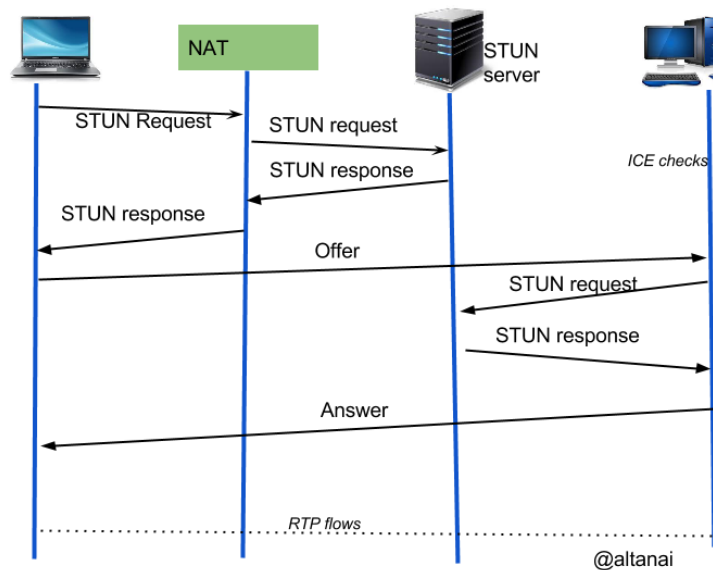
- NACK qui permet de signaler à l'émetteur une réception échouée ou l'absence de transmission ;
- FEC, un code de contrôle qui permet au récepteur de vérifier que la totalité des paquets est arrivée correctement ;

Lors de l'envoi d'un flux média, l'émetteur découpe le flux et calcule une somme de contrôle (FEC) qui est envoyée avec ces paquets. À la réception, les FEC sont recalculés pour vérifier l'absence d'erreurs et les données stockées dans une mémoire tampon. Si des paquets manquent, ils sont redemandés.

Dans le cadre de l'API WebRTC, une solution hybride entre NACK et FEC a été implémentée, accompagnée de contrôles temporels afin d'équilibrer la qualité de la vidéo, sa fluidité et le temps de réponse d'une extrémité de la connexion à l'autre.

- Transiter au travers d'un pare-feu

WebRTC peut être difficile à utiliser en entreprises dans la mesure où celles-ci ont souvent des politiques de sécurité en informatique incompatibles avec les besoins de l'API. En effet, WebRTC est basé sur des flux peer-to-peer entre navigateurs et ces flux sont très sensibles à la latence lors qu'il s'agit de flux médias. En outre, les serveurs utilisés pour faire transiter les paquets peuvent être éloignés géographiquement des paires qui communiquent ou avoir une bande passante trop faible pour permettre un transit correct des paquets.

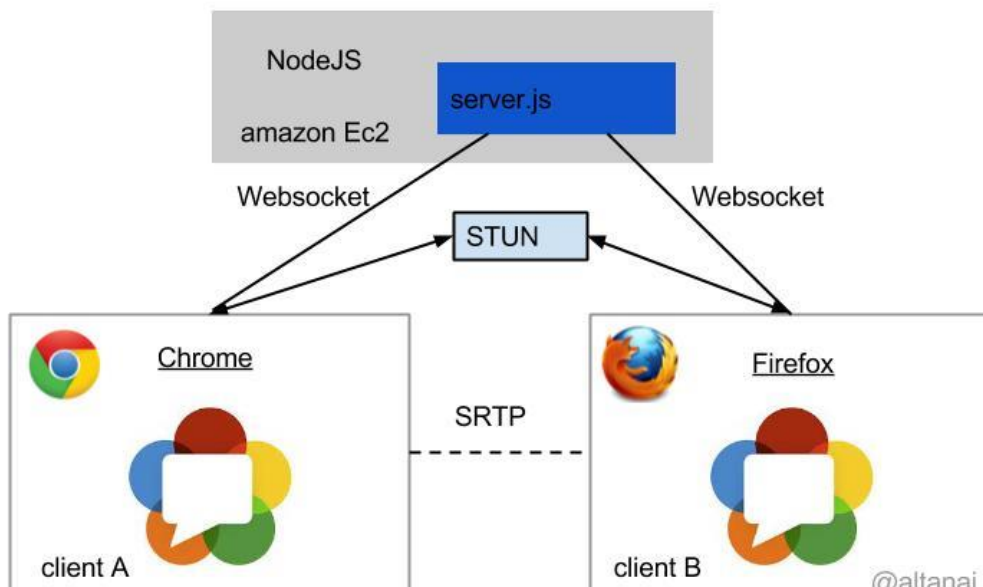


▪ *Figure3.8 Transiter au travers d'un pare-feu*

▪ Transiter au travers d'un NAT

Afin d'être utilisable si l'une des paires se situe derrière un NAT, WebRTC utilise le protocole ICE. Deux techniques principales sont utilisées pour traverser ce genre de difficultés.

Step 2.1 : Put Server on cloud and WebRTC clients on different machine + STUN for address discovery ( NAT traversal )



*Figure3.9 Transiter au travers d'un NAT*

La première technique est souvent appelée Hole Punching (en): l'appareil à l'intérieur du NAT envoie un paquet STUN à un serveur en dehors du NAT. Le serveur répond en informant l'émetteur de l'adresse IP et du port apparent avec lequel le paquet a été envoyé, sur lequel les deux communiqueront.

La seconde technique utilise un relais intermédiaire. Le protocole utilisé pour cela est Traversal Using Relays around NAT: l'entreprise déploie un serveur TURN au sein de la zone démilitarisée (DMZ) avec lequel la paire interne communique et se charge de vérifier qu'il a les droits requis et de surveiller les flux médias qui passent par lui. Le serveur sert ainsi de point de relais aux paquets WebRTC qui transitent entre une paire et l'autre

- Alternatives du WebRTC

En août 2012, Microsoft a présenté une proposition alternative appelée CU-RTC-WEB (Customizable, Ubiquitous Real Time Communication over the Web) au groupe WebRTC du W3C<sup>81,82</sup>, une technologie qui aurait débuté en 2010 conjointement avec Skype (que Microsoft a racheté en 2011)

### 3.1.6 protocoles utilisées par webRTC

- Simple Traversal of UDP through NATs(STUN)

Le STUN (« Simple Traversal of UDP through NATs » ou « traversée simple de UDP à travers les NAT ») est un protocole client-serveur de l'IETF (RFC 34891) permettant à un client UDP situé derrière un routeur NAT (ou de multiples NAT) de découvrir son adresse IP publique ainsi que le type de routeur NAT derrière lequel il est.

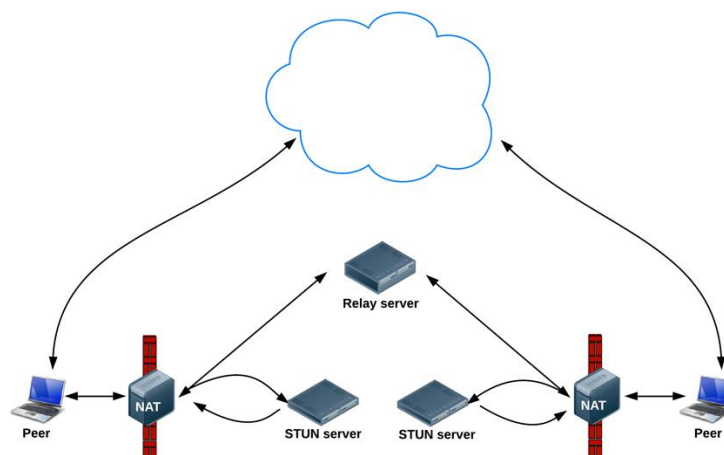


Figure 3.10 architecture de STUN

Ces informations sont utilisées pour échanger correctement des données UDP avec l'extérieur d'un réseau NAT. La RFC 3489 de STUN date de 2003, les sociétés CISCO et Microsoft y ont participé. Un serveur STUN écoute en principe le port 3478 et le 3479.

- Datagram Transport Layer Security(DTLS)

En informatique, le protocole Datagram Transport Layer Security (DTLS) fournit une sécurisation des échanges basés sur des protocoles en mode datagramme. Le protocole DTLS est basé sur le protocole TLS et fournit des garanties de sécurité similaires.

- Secure Real-time Transport Protocol(SRTP)

Secure Real-time Transport Protocol (ou SRTP) définit un profil de RTP (Real-time Transport Protocol) qui a pour but d'apporter le chiffrement, l'authentification et l'intégrité des messages et la protection contre le rejeu (replay) de données RTP. SRTP fonctionne à la fois en envoi ciblé (unicast) et en multidiffusion (multicast). Le SRTP a été conçu par Cisco et Ericsson. Il est ratifié par l'IETF en tant que RFC 3711. Il existe aussi un SRTCP (Secure Real-time Control Protocol), une version sécurisée de RTCP (Real-time Control Protocol).

- Session Description Protocol

Le Session Description Protocol (SDP) est un protocole de communication de description de paramètres d'initialisation d'une session de diffusion en flux (streaming). Le SDP a été créé pour décrire des sessions de communication multimédia, par exemple pour l'annonce de la session, l'invitation à une session et la négociation de paramètres.

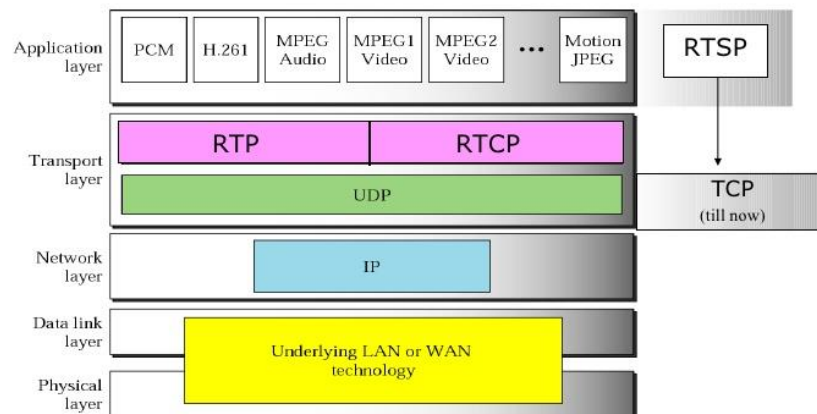
Le SDP ne livre pas le média lui-même, mais est utilisé par l'émetteur et le destinataire pour la négociation du type et du format du média, et les propriétés associées. L'ensemble des paramètres d'une session est souvent appelé un profil de session. Le SDP a été conçu pour être extensible et soutenir les nouveaux types et formats de médias.

Le SDP a commencé comme une composante du Session Announcement Protocol (SAP), mais a trouvé d'autres utilisations en conjonction avec le Real-time Transport Protocol (RTP), le Real-Time Streaming Protocol (RTSP), le Session Initiation Protocol (SIP) et même comme un format autonome pour décrire des sessions de multidiffusion.

- Real-time Transport Protocol

Real-Time Transport Protocol (RTP) est un protocole de communication informatique permettant le transport de données soumises à des contraintes de temps réel, tels que des flux média audio ou vidéo

## Protocol stack for multimedia services



*Figure 3.11 RTP dans le modèle OSI*

RTP est à l'heure actuelle principalement utilisé comme transport de média pour les services de la voix sur IP ou de vidéo conférence, voire de streaming. En mode unidirectionnel, il est toujours associé avec un autre protocole de signalisation qui gère l'établissement de session et permet l'échange du numéro de port utilisé par les deux extrémités. On peut citer :

- le protocole SIP pour les services de VoIP et de visioconférences ;
- le protocole H.323 pour les mêmes services (ancienne génération) ;
- le protocole RTSP pour le streaming bien que ce dernier possède un mode d'encapsulation TCP.
- Le protocole ajoute un en-tête spécifique aux paquets UDP pour spécifier le type et le format (codec) du média transporté ;

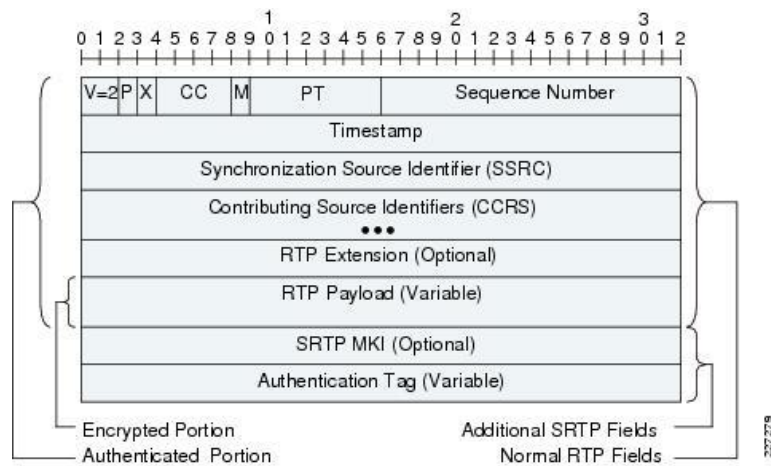


Figure3.12 le paquet UDP

RTP sera utilisé avantageusement sur un réseau temps réel (par exemple un réseau ATM à bande passante garantie, un canal optique, une radiodiffusion ou un canal satellite).

- Session Announcement Protocol

Session Announcement Protocol (SAP) est un protocole pour diffuser des informations de session multicast.

- Real Time Streaming Protocol(RTSP)

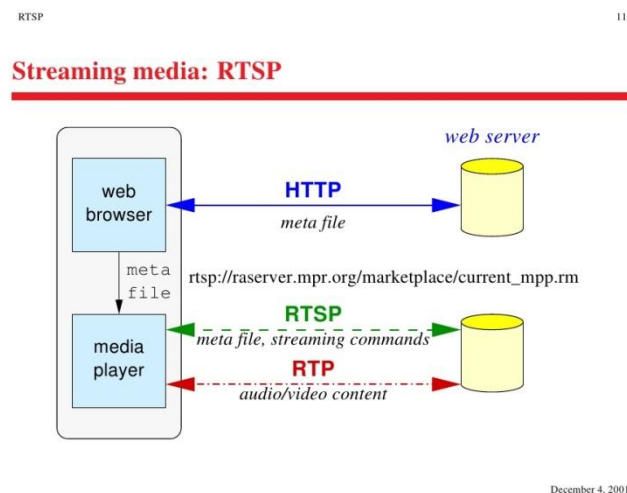


Figure3.13 transport des flux RTSP

RTSP ou Real Time Streaming Protocol (protocole de streaming temps-réel) est un protocole de communication de niveau applicatif (niveau 7 du modèle OSI) destiné aux systèmes de streaming média. Il permet de contrôler un serveur de média à distance, offrant des fonctionnalités typiques d'un lecteur vidéo telles que « lecture » et « pause », et permettant un

accès en fonction de la position temporelle. La RTSP ne transporte pas les données elles-mêmes et doit être associé à un protocole de transport comme RTP ou RDT de RealNetworks pour cette tâche.

### **3.1.7 le codec utilise par webRTC**

- Le codec VP8

VP8 était le dernier codec vidéo de On2 Technologies qui a remplacé VP7, son prédécesseur. Il a été annoncé le 13 septembre 2008. Réalisé à l'origine dans un format propriétaire, il a été racheté par Google qui en a fait un format ouvert le 19 mai 2010 dans le cadre du projet WebM. Il est également utilisé dans le format d'image WebP.

- Le codec VP8 vs H.264

Le remplacement éventuel de H.264 par VP8 (lié à celui de MP4 par WebM) reste un sujet de discussion.

- Utilisation dans des logiciels

La version Windows de Skype 5.5 utilise le VP8 pour la communication entre 2 personnes ou en conférences multiple<sup>3</sup>. Après le rachat par Microsoft le codec a été abandonné.

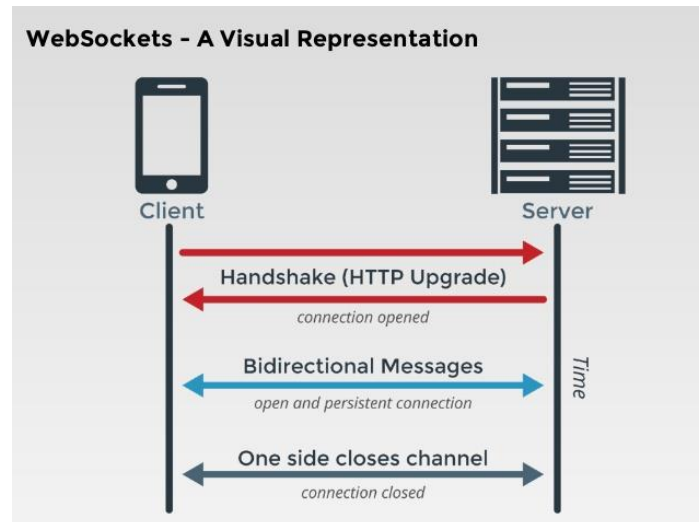
Proposé au W3C comme codec vidéo (au sein de WebM), pour la balise <video>, il peut donc être lu par les navigateurs prenant en charge ce format (Firefox, Google Chrome, Opera, Internet Explorer 9 ou plus avec un composant additionnel).

## **3.2 WebSocket**

La procédure WebSocket est un standard du Web désignant un protocole réseau de la couche application et une interface de programmation du World Wide Web visant à créer des canaux de communication full-duplex par dessus une connexion TCP pour les navigateurs et les serveurs web.

Le protocole a été normalisé par l'IETF dans la RFC 6455 en 2011 et l'interface de programmation est en cours de standardisation par le W3C.





*Figure3.14 déroulement de la communication dans le websocket*

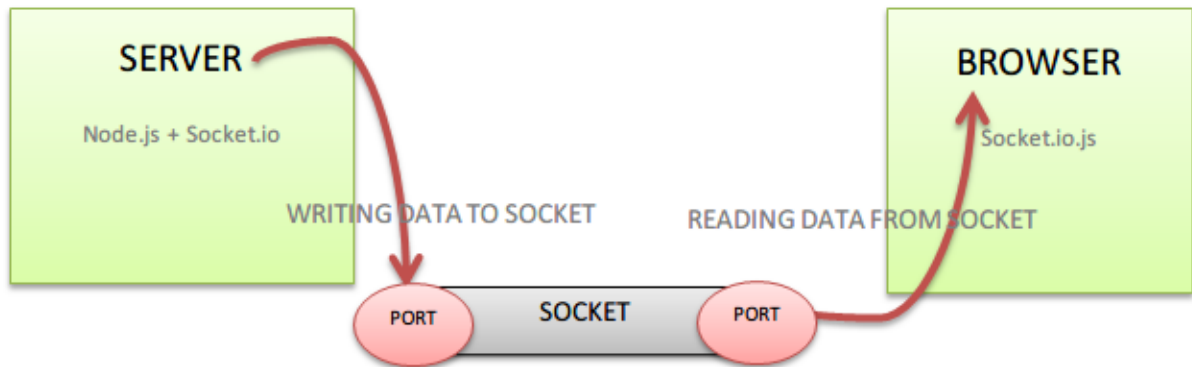
WebSocket utilise Des techniques basées sur l'appel par le client de l'objet XMLHttpRequest et utilisant des requêtes HTTP avec un long TTL stockées par le serveur pour une réponse ultérieure au client ont permis de pallier ce manque et ont été popularisées par le succès des architectures Ajax.

- exemple d'API WebSocket
- Apache WebSocket module3, une implémentation en langage C sous la forme d'une extension pour le serveur httpd Apache(mod\_websocket) ;
- pywebsocket4, une implémentation en Python sous la forme d'une extension pour le serveur httpd Apache (mod\_pywebsocket) ;
- PubNub, implémentation proposant une API, compatible avec tous les langages utilisés par les technologies mobiles, du web, et IoT, service gratuit ou payant ;
- Pusher, implémentation sous forme d'API compatible avec la plupart des langages ;
- SignalR, implementation pour ASP.NET en C# ;
- Socket.io, implémentation javascript du protocole pour Node.js ;

### 3.2.1 socket.io



Socket.io est une librairie Javascript qui permet d'effectuer non seulement des communications asynchrones bidirectionnelles entre client et serveur (comme prévu par le protocol WebSocket) mais également bien plus. Socket.io n'est pas une librairie uniquement basée sur WebSocket, ce qui a des avantages et des inconvénients.



*Figure3.15 architecture socket io*

socket.io ajoute également plusieurs fonctionnalités non prévues par le, ce qui a comme inconvénient de perdre la conformité avec le protocole de base et donc l'interopérabilité avec les autres librairies qui peuvent exister.

Ainsi, si on développe une application avec socket.io le client et le serveur doivent utiliser socket.io. Côté client, la librairie consiste à un simple fichier Javascript à inclure, côté serveur c'est une librairie pour Node.js

- Node.js



Node.js est une plateforme logicielle libre et événementielle en JavaScript orientée vers les applications réseau qui doivent pouvoir monter en charge. Elle utilise la machine virtuelle V8 et implémente sous licence MIT les spécifications CommonJS.

## **4 Les modules**

### **4.1. Web Conférence**

l'utilisateur a 2 choix soit créer une room ou rejoindre une room

lorsque utilisateur créer une room le navigateur le demande d'ouvrir le webcam. si l'utilisateur accepte son webcam sera partagé sinon le webcam est ferme

lorsque l'utilisateur veut rejoindre une room déjà initie il clique sur le bouton rejoindre et il joint ensuite la réunion. il peut aussi partager soit écran ou micros webcam documents faire chat dessiner

### **4.2. partage des documents**

l'utilisateur peut a travers la conférence web de partage des documents comme docx ou PDF ensuite le document sera partage pour tout les participants de la réunion

### **4.3. enregistrement**

Notre plateforme permet pour tout les participants d'enregistrer tout ce qui est vidéo audio et événement qui se déroule au cours de la réunion ,les tableaux dessin é en forme d'ensembles d'image dans un même dossier et permettre l'utilisateur enfin de les lire a travers un mini lecteur enHTML5

Lorsque l'utilisateur finit l'enregistrement qui sur finit enregistrement et un fichier zip contentent tout les fichier sera téléchargé

### **4.4. Le tableau blanc**

cet fonctionnalité permet au utilisateur de dessiner a travers des outils avance et et partage sa propre tableau avec les autres participants

## **5 Conclusion**

Dans ce chapitre nous avons présenté notre environnement de travail matériel et logiciel ainsi que les principales interfaces de notre application avec leurs descriptions.

## **Conclusion**

Dans notre projet nous avons réussi à construire un système de visioconférence pour rendre service aux différentes institutions éducatives du monde entier. Ce système comme il est conçu, contribue à la création d'un environnement propice au partage d'informations et permet aux entreprises de renforcer leur productivité, tout en diminuant la durée et les coûts des déplacements.

Malgré les avantages cités, la vidéoconférence s'avère peu déployée et peu utilisée en dehors des salles de conférence et des bureaux de la direction à cause, de son coût, de la difficulté d'utilisation et des performances qui sont peu convaincantes. Ainsi, nous avons cherché dans notre travail à montrer la nécessité d'une étude complète et plus approfondie pour atteindre l'objectif principal de ce travail qui consiste à développer un programme innovant avec des compétences nationales.

Notre étude a touché plusieurs phases de réalisation de ce projet. Ainsi, nous avons commencé par l'étude préalable puis l'étude et la description de notre système, cette étude a pour objectif de préciser tous les étapes abordées dans notre projet.

Comme perspective, nous espérons ajouter une extension permettant l'enregistrement des différentes sessions de visioconférences. Les enregistrements seront indexés et répertoriés pour être utilisés ultérieurement.