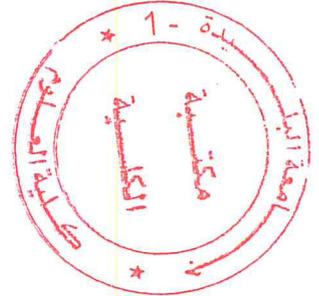


République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Saad Dahlab Blida



Faculté des sciences
Département d'informatique

Mémoire présenté par :

CHEURFA Ihssan BENBITAT Hind

En vue d'obtenir le diplôme de master

**Sujet : Gestion des canaux de communication pour les
Webinaires inter-universités**

Option : système informatique et réseaux

Organisme d'accueil : centre de développement et de technologies avancés

Soutenu le : 04/07/2018

Devant le jury :



M. OULD AISSA Ahmed

Mm. ARKAM Meriem

M. DOUGA Yassine

M. HENTOUT Abdelfetah

M. TIBERKAK Allal

Examineur

Présidente

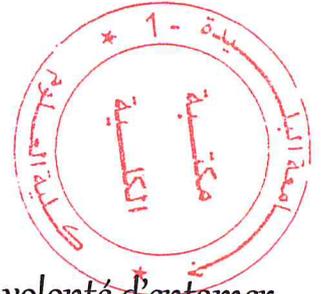
Promoteur

Encadreur

Encadreur

2017/2018

Remerciement



On remercie dieu le tout-puissant de nous avoir donnés la santé et la volonté d'entamer
et de terminer ce mémoire.

Tout d'abord, ce travail ne serait pas aussi riche et n'aurait pas pu avoir le jour sans
l'aide et l'encadrement de

Mr Allal TIBERKAK et Mr Hentout, on les remercie pour la qualité de leur encadrement,
pour leur patience, rigueur et leur disponibilité durant notre préparation de ce
mémoire.

Nos remerciements s'adressent à Mr Yassine DOUGA. Notre promoteur pour son aide
et ses conseils et ses encouragements.

Nous sommes conscientes de l'honneur que nous a fait Mr Mm ARKAM Meriem en
étant président du jury

Et Mr A. OULD AISSA d'avoir accepté d'examiner ce travail.

Nos remerciements s'adressent également à tous nos professeurs pour leurs générosités
et à Mr M. Ould khaoua pour la grande patience dont il a su faire preuve malgré ces
charges académiques et professionnelles.

Nos profonds remerciements vont également à toutes les personnes qui nous aient aidés et soutenus de près ou de loin principalement à tous l'effectif du Centre de développement des technologies avancées CDTA

Dédicace

A l'homme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que dieu te garde dans son vaste paradis, à toi mon père.

A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur ; maman que j'adore.

A la personne la plus cher dans ma vie mon grand-père, que dieu le garde dans son vaste paradis

Aux personnes dont j'ai bien aimé la présence dans ce jour, à mon cher frère et mes sœurs, Insaf, fatma-zohra, Malec, et ma princesse Walaa, je dédie ce travail dont le grand plaisir leurs revient en premier lieu pour leurs conseils, aides, et encouragements.

A mes tantes et mes oncles, et surtout mes chères cousines soulef, imen, amel, hasna, lamia, nessrine, zakia, et mes petits asma, karim Lila et adem

A mon binôme hind et toutes la famille benbitat, pour leurs accueils, hospitalité, et générosité, et surtout à toi maroua.

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures, mes aimables amis, collègues d'étude, et frères de cœur, toi asma souidi, lilia ghemouri, nessrine h-zerouk, chatal asmaa, mohamed hanifi, salim boukhalifa, allali mustapha .

CHEURFA Ihssan

Dédicace

A l'homme de ma vie, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui s'est toujours sacrifié pour me voir réussir, que dieu te garde dans son vaste paradis, à toi mon père.

A la lumière de mes jours, la source de mes efforts, la flamme de mon cœur, ma vie et mon bonheur ; maman que j'adore.

A mes chères grands parents que dieu les pour nous

Aux personnes dont j'ai bien aimé la présence dans ce jour, à ma chère sœur Maroua et mon cher frère Fahed et sa femme Nour, je dédie ce travail dont le grand plaisir leurs revient en premier lieu pour leurs conseils, aides, et encouragements.

A mes tantes et mes oncles, et surtout mes chères cousines Selma, Neila K, Rym, Neila A, Asma, Anissa, Soumia, Souhila, Khalida, Sofia, shahinaz et toutes mes autres cousines et à mes chers cousins Billel et Badro

A mon binôme Ihssan et toutes la famille Cheurfa, pour leurs générosité.

Aux personnes qui m'ont toujours aidé et encouragé, qui étaient toujours à mes côtés, et qui m'ont accompagnaient durant mon chemin d'études supérieures, mes aimables amis, collègues d'étude, toi Djelloul Yahiaoui, Ahlem Oudini, Nesma Ouanougui, Mohamedi Hadjer, Lilia Ghemouri, Asma Aid, Amina Khanafif, Meriem et Amira
Adour.

BENBITAT Hind

Table des matières

| | |
|--|----|
| Remerciement..... | 2 |
| Dédicace..... | 5 |
| Table des figures | 9 |
| Liste des tableaux..... | 10 |
| ملخص..... | 11 |
| Abstract..... | 12 |
| Résumé :..... | 13 |
| Introduction générale | 14 |
| Chapitre 1 : Etat de l'art..... | 16 |
| 1 Introduction | 16 |
| 2 Architectures réseaux | 16 |
| 2.1 Architecture Client/Serveur | 16 |
| 2.2 Architecture peer-to-peer..... | 18 |
| 2.2.1 L'architecture centralisée | 18 |
| 2.2.2 L'architecture distribuée ou décentralisée | 19 |
| 2.2.3 L'architecture hybride..... | 21 |
| 3 Protocoles | 22 |
| 3.1 Protocole RTP..... | 22 |
| 3.2 Protocole RTCP..... | 23 |
| 4 WebRTC | 23 |
| 5 Signalisation | 24 |
| 5.1 Protocoles de signalisation | 25 |
| 5.1.1 Protocole SDP..... | 25 |
| 5.1.2 Protocole SIP | 25 |
| 6 Pile protocolaire..... | 26 |
| 7 Conclusion | 27 |
| Chapitre 2 : Systèmes webinaires | 28 |
| 1 Introduction | 28 |
| 2 Définitions..... | 28 |
| 2.1 Séminaire | 28 |
| 2.2 Voix sur IP..... | 29 |
| 2.3 Web conférence..... | 29 |
| 2.4 Webinaire..... | 29 |
| 3 Webinaires | 30 |

| | | |
|--|--|----|
| 3.1 | Eléments d'un webinaire | 30 |
| 3.2 | Fonctionnalités des webinaires..... | 31 |
| 3.3 | Avantages des webinaires..... | 31 |
| 3.4 | Exemples des systèmes webinaires | 32 |
| 3.4.1 | Skype : | 32 |
| 3.4.2 | Google Hangouts..... | 32 |
| 3.4.3 | iChat | 33 |
| 3.4.4 | WebEx : | 34 |
| 3.4.5 | Ready talk..... | 34 |
| 3.4.6 | Adobe Connect..... | 35 |
| 4 | Etude de l'existant | 36 |
| 4.1 | Analyse de l'existant | 36 |
| 4.1.1 | Architecture de Skype..... | 36 |
| 4.1.2 | Architecture Hangout Google + | 38 |
| 4.1.3 | Architecture iChat | 38 |
| 4.2 | Etude critique..... | 39 |
| 5 | Conclusion..... | 40 |
| Chapitre 3 : Conception et proposition de solution | | 42 |
| 1 | Introduction | 42 |
| 2 | Topologie du système « UnivWebinar » | 42 |
| 3 | Langage UML..... | 44 |
| 3.1 | Intérêt d'UML..... | 45 |
| 4 | Processus de développement | 45 |
| 4.1 | Processus unifié | 45 |
| 5 | Conception de l'application UnivWebinar | 46 |
| 5.1 | Spécification des besoins | 46 |
| 5.1.1 | Besoins fonctionnels | 46 |
| 5.1.2 | Besoins non fonctionnels | 47 |
| 5.2 | Diagramme de cas d'utilisation..... | 47 |
| 5.2.1 | Identification des acteurs..... | 47 |
| 5.2.2 | Cas d'utilisation..... | 48 |
| 5.2.3 | Description textuelle des principaux cas d'utilisation | 49 |
| 5.3 | Diagramme de séquence | 53 |
| 5.4 | Diagramme de classe | 61 |
| 6 | Description de la solution | 62 |
| 6.1 | Serveur de signalisation | 62 |
| 6.2 | Le Nœud..... | 63 |
| 6.3 | Algorithme du chemin optimal | 63 |

| | | |
|-----|--|----|
| 7 | Conclusion..... | 65 |
| | Chapitre 4 : Implémentation..... | 66 |
| 1 | Introduction | 66 |
| 2 | Outils de développement..... | 66 |
| 3 | Présentation des interfaces de « UnivWebinar » | 67 |
| 3.1 | Interface du serveur de signalisation..... | 67 |
| 3.2 | Interface du nœud | 68 |
| 3.3 | Interface d'inscription..... | 69 |
| 3.4 | Interface d'authentification | 71 |
| 3.5 | Interface du conférencier | 72 |
| 3.6 | Interface du participant | 75 |
| 4 | Scénario et tests..... | 78 |
| 5 | Conclusion..... | 80 |
| | Conclusion générale..... | 81 |
| | Glossaire..... | 83 |
| | Bibliographie | 84 |

Table des figures

| | |
|--|----|
| Figure 1: modèle client-Serveur | 17 |
| Figure 2 : l'architecture centralisée | 19 |
| Figure 3 : l'architecture décentralisée | 20 |
| Figure 4: l'architecture hybride | 21 |
| Figure 5 : Etablissement d'une connexion entre deux clients utilisant WebRTC..... | 24 |
| Figure 6 : communication avec le protocole SIP | 26 |
| Figure 7 : pile de protocoles de transport et signalisation | 27 |
| Figure 8 : interface de Skype..... | 32 |
| Figure 9 : Interface de Google Hangouts | 33 |
| Figure 10 : Interface d'iChat..... | 33 |
| Figure 11 : Interface de WebEx | 34 |
| Figure 12 : Interface de Ready Talk..... | 35 |
| Figure 13 : Interface d'Adobe Connect..... | 36 |
| Figure 14 : L'architecture Skype | 37 |
| Figure 15 : L'architecture de Hangouts | 38 |
| Figure 16 : L'architecture d'iChat | 39 |
| Figure 17 : Architecture du système « UnivWebinar »..... | 43 |
| Figure 18 : Les diagrammes UML | 45 |
| Figure 19 : Diagramme de cas d'utilisation du conférencier..... | 48 |
| Figure 22 : Diagramme de séquence système pour l'inscription..... | 53 |
| Figure 23 : Diagramme de séquence système pour l'authentification..... | 54 |
| Figure 24 : Diagramme de séquence système pour créer webinaire | 55 |
| Figure 25 : Diagramme de séquence système pour lancer webinaire | 56 |
| Figure 26 : Diagramme de séquence pour activer nœud..... | 57 |
| Figure 27 : Diagramme de séquence pour consulter webinaire | 58 |
| Figure 28 : Diagramme de séquence pour rejoindre webinaire | 59 |
| Figure 29 : Diagramme de séquence du cas d'utilisation quitter webinaire(conférencier) | 60 |
| Figure 30 : Diagramme de séquence du cas d'utilisation quitter webinaire (participant) | 60 |
| Figure 31 : diagramme de classe | 61 |
| Figure 32 : Matrice des couts | 62 |
| Figure 33 : matrice des conférences | 63 |
| Figure 35 : Interface du serveur de signalisation | 68 |
| Figure 36 : Interface du nœud | 69 |
| Figure 37 : Interface d'inscription | 70 |
| Figure 38 : Interface d'authentification | 72 |
| Figure 39 : Interface du conférencier (1) | 74 |
| Figure 40 : Interface du conférencier (2) | 75 |
| Figure 41 : Interface du participant (1) | 76 |
| Figure 42 : Interface du participant (2) | 77 |
| Figure 43 : Scénario de tests du système UnivWebinar..... | 79 |

Liste des tableaux

| | |
|---|----|
| Tableau 1 : Comparaison des architectures des systèmes webinaires..... | 40 |
| Tableau 2 : Description textuelle du cas d'utilisation inscription..... | 50 |
| Tableau 3 : Description textuelle du cas d'utilisation authentification..... | 50 |
| Tableau 4 : Description textuelle du cas d'utilisation créer webinaire | 51 |
| Tableau 5 : Description textuelle du cas d'utilisation lancer webinaire | 51 |
| Tableau 6 : Description textuelle du cas d'utilisation rejoindre webinaire..... | 52 |
| Tableau 7 : Description textuelle du cas d'utilisation activer nœud | 52 |

ملخص

لتسهيل التعليم عن بعد أو العمل التعاوني، اختارت الجامعات مؤخرًا استخدام الندوات عبر الإنترنت. تشير ندوات الويب إلى الاجتماعات والندوات التفاعلية عبر الإنترنت. المشكلة في مثل هذه الحالة هي أنه يجب على كل مشارك فتح قناة سمعية بصرية ثنائية الاتجاه مع خادم البث. هذا النهج غير مناسب لمشاركة المؤتمرات المتحركة على مستوى الجامعات الجزائرية، لأن هذا الحل يتطلب توفر نطاق ترددي كبير على مستوى خادم البث.

الغرض من هذا المشروع هو تطوير نظام إدارة ديناميكي لقنوات الاتصال بين المشاركين والمتحدث لضمان النشر والتشغيل السلس واستمرارية المؤتمرات. يسمح حلنا بتوجيه التدفقات عبر عقد متعددة لزيادة قابلية التحجيم وخفض الحمل على جهاز المتحدث. بالإضافة إلى ذلك، فإنه يوفر المسار الأمثل لكل تدفق.

من أجل تحقيق مشروعنا، اخترنا حلاً مستوحى من تقنية الويب في الوقت الحقيقي وبصورة أدق جزء الإشارة.

Abstract

In order to facilitate distance education and collaborative work, universities have recently opted for the use of webinars (web seminar). Webinars refer to interactive meetings and seminars via the internet. The problem in this kind of service is that each participant has to open a bidirectional audio-visual channel with the broadcast server. This approach is not suitable for sharing animated conferences in respect to Algerian universities, this solution requires the availability of a large bandwidth at the level of the broadcast server.

The purpose of this project is to develop a dynamic management system of communication channels between participants and speakers to ensure a good diffusion, a good proceeding and the continuity of conferences. Our proposed approach allows routing flows across multiple nodes to increase scalability and decrease the load on the lecturer terminal. In addition, it must offer an optimal path for each flow.

For the realization of our project, we opted for a solution inspired by the real-time web technology (WebRTC) and more precisely the signaling part.

Key words : Webinaire, WebRTC, Socket, Signalisation, SIP, UDP, TCP.

Résumé :

Afin de faciliter l'enseignement à distance et le travail collaboratif, les universités ont récemment optés pour l'utilisation des webinaires (web séminaire). Les webinaires réfèrent aux réunions interactives et aux séminaires via internet. Le problème dans ce genre de service est que chaque participant doit ouvrir un canal audio-visuel bidirectionnel avec le serveur de diffusion. Cette approche ne convient pas au partage de conférences animées au niveau des universités algériennes, car cette solution requiert la disponibilité d'une bande passante importante au niveau du serveur de diffusion.

Le but de ce projet est de développer un système de gestion dynamique des canaux de communication entre les participants et les conférenciers pour assurer une bonne diffusion, le bon déroulement et la continuité des conférences. Notre approche proposée permet d'acheminer les flux à travers plusieurs nœuds pour augmenter la scalabilité et diminuer la charge sur le terminal du conférencier. De plus, il doit offrir un chemin optimal pour chaque flux.

Pour la réalisation de notre projet, nous avons opté pour une solution inspirée de la technologie web temps réel (WebRTC) et plus précisément la partie de signalisation.

Mots clé : Webinaire, WebRTC, Socket, Signalisation, SIP, UDP, TCP.

Introduction générale

Internet s'est fortement développé comme outil de communication et elle est devenue un standard dans la communication au niveau international. Les entreprises, mais aussi les citoyens, ont recours à ce média pour réaliser plusieurs actions concernant les loisirs, mais aussi les activités professionnelles. Ainsi, dans plusieurs domaines, Internet est devenu un outil fondamental, comme c'est le cas de l'enseignement supérieur.

Pour gérer leurs projets, les entreprises ont de plus en plus besoin de mettre en commun des ressources avec différents acteurs. Le développement des technologies de l'information et de la communication (TIC) a permis l'émergence de nouvelles applications via Internet : audio/vidéoconférences, téléréunions, chat électronique, webinaire, etc.

Pour le travail collaboratif et l'enseignement à distance les webinaires sont devenus un outil indispensable car ils assurent un environnement éducatif en ligne pour des liens plus étroits entre les étudiants, les éducateurs, les chercheurs et les autres participants.

Cependant, il existe une panoplie de solutions logicielles sur le marché. Telles solutions proposent un certain nombre d'outils de web conférence. Ces solutions requièrent la disponibilité d'une bande passante importante au niveau du serveur de diffusion et mènent à la saturation et la surcharge au niveau du centre de diffusion. C'est dans cette optique que s'inscrit ce travail de master. Dont l'objectif est de développer un système de gestion des canaux de communication pour les webinaires entre les participants et les centres de diffusions. Un tel système permet de gérer les différents canaux de communication bidirectionnelle entre chaque participant et les centres de diffusions traversés par les canaux. Ce système offre des fonctionnalités comme la possibilité d'interactions et de partage de données entre les acteurs tout en s'inspirant du principe de la technologie WebRTC.

La suite de ce mémoire est organisée comme suit :

- **Chapitre 1** : Ce chapitre présente les architectures réseaux client/serveur et pair à pair ainsi des notions réseaux comme la pile protocolaire (TCP/IP) et les protocoles de transport temps réel (RTP, RTCP) à la fin de ce chapitre nous allons décrire la technologie WebRTC en se basant sur la partie de signalisation et ces principaux protocoles.
- **Chapitre 2** : Ce chapitre présente une vision globale sur les systèmes webinaire et leurs fonctionnalités, nous allons étudier également quelques exemples sur les différentes

architectures utilisées dans certains outils de webinaires disponibles sur le marché et à la fin nous proposons une solution au problème posé.

- **Chapitre 3 :** Dans ce chapitre nous décrivons l'étude conceptuelle de notre application en utilisant le langage UML et nous décrivons notre solution proposée.
- **Chapitre 4 :** Ce chapitre concerne l'implémentation de notre application avec des jeux de tests qui sont présentés à la fin pour illustrer les différentes fonctionnalités de notre application.
- Finalement une conclusion générale pour donner un aperçu et discuter sur ce qui est fait dans ce mémoire. Nous allons définir aussi des perspectives à réaliser dans le futur.

Chapitre 1 : Etat de l'art

1 Introduction

Internet est devenu un nouvel outil d'information et de communication en pleine évolution offrant des perspectives de croissance exceptionnelles. C'est un système mondial d'interconnexion de réseaux informatiques, utilisant un ensemble standardisé de protocoles de transfert, de contrôle et de sécurité de données et un ensemble de technologies de communication en temps réel.

Dans ce chapitre nous allons présenter les architectures réseaux telles que l'architecture client-serveur et l'architecture paire à paire ensuite nous décrirons les protocoles de transport en temps réel, nous abordons également la technologie WebRTC liés à la communication en temps réel, et au partage de données, audio et vidéo avec la notion de signalisation et d'autres notions qui pourront nous être utile pour la réalisation de notre projet.

2 Architectures réseaux

Il existe plusieurs architectures réseaux que les systèmes adoptent pour la transmission et la réception de données, dans ce qui suit nous allons citer quelques architectures réseau les plus connues avec leurs avantages et inconvénients.

2.1 Architecture Client/Serveur¹

L'environnement client-serveur (figure 1) désigne un mode de communication à travers un réseau entre plusieurs programmes : l'un, qualifié de client, envoie des requêtes ; l'autre ou les autres, qualifiés de serveurs, attendent les requêtes des clients et y répondent. Par extension, le client désigne également l'ordinateur ou la machine virtuelle sur lequel est exécuté le logiciel client, et le serveur, l'ordinateur ou la machine virtuelle sur lequel est exécuté le logiciel serveur.

Les serveurs sont des ordinateurs généralement dédiés au logiciel serveur qu'ils abritent, et dotés de capacités supérieures à celles des ordinateurs personnels en ce qui concerne la puissance de calcul, les entrées-sorties et les connexions réseau. Les clients sont souvent des ordinateurs personnels ou des appareils individuels (téléphone, tablette), mais pas systématiquement. Un serveur peut répondre aux requêtes d'un grand nombre de clients.

¹<http://www.wikipédia.org/Client-serveur>

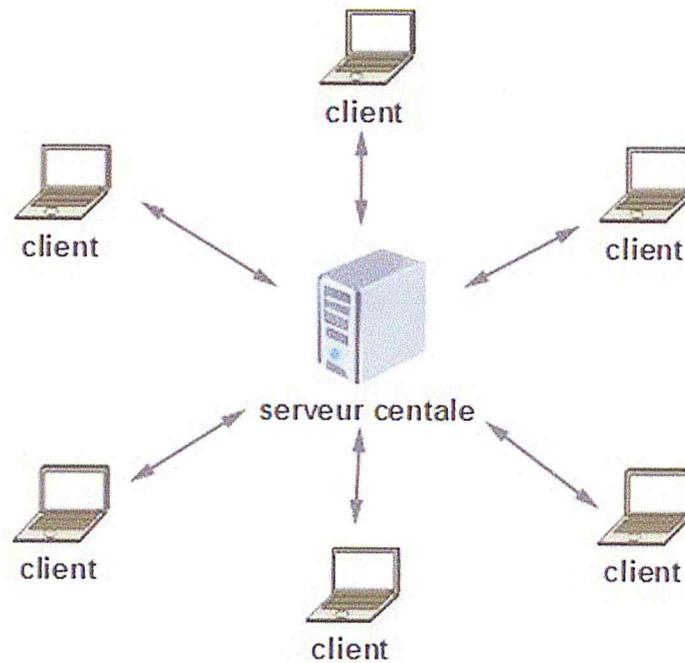


Figure 1: modèle client-Serveur¹

2.1.1 Avantages de l'environnement Client/Serveur²

Le modèle client/serveur est particulièrement recommandé pour des réseaux nécessitant un grand niveau de fiabilité, ses principaux atouts sont :

- **des ressources centralisées** : étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, comme par exemple une base de données centralisée, afin d'éviter les problèmes de redondance et de contradiction
- **une administration au niveau serveur** : les clients ayant peu d'importance dans ce modèle, ils ont moins besoin d'être administrés
- **un réseau évolutif** : grâce à cette architecture il est possible de supprimer ou rajouter des clients sans perturber le fonctionnement du réseau et sans modification majeure

2.1.2 Inconvénients de l'environnement Client/Serveur²

L'architecture client/serveur a tout de même quelques inconvénients parmi lesquelles :

- **un coût élevé** dû à la technicité du serveur (nécessite une bande passante élevée)

² <https://www.commentcamarche/environnement-client-serveur>

- **un maillon faible** : le serveur est le seul maillon faible du réseau client/serveur, étant donné que tout le réseau est architecturé autour de lui donc si le serveur tombe en panne tout le réseau tombe en panne.

2.2 Architecture peer-to-peer

Le peer-to-peer est un modèle de réseau dans lequel chaque machine fait à la fois office de serveur et de client. Les éléments composant le réseau sont appelés des nœuds. [1]

Les systèmes peer-to-peer permettent à plusieurs ordinateurs de communiquer via un réseau, en y partagent des fichiers le plus souvent, mais également des flux multimédia continus (streaming³), le calcul réparti, un service comme la téléphonie sur IP...

La particularité des architectures pair-à-pair réside dans le fait que les données puissent être transférées directement entre deux postes connectés au réseau, sans transiter par un serveur central. Il permet ainsi à tous les ordinateurs de jouer directement le rôle de client et serveur.

Il existe trois types d'architectures distinctes pour les réseaux « P2P » : le réseau centralisé, le réseau distribué et enfin le réseau hybride qui, comme son nom l'indique, combine les deux premières formes de réseau. [2]

2.2.1 L'architecture centralisée

Le représentant le plus connu de ce mode de P2P est Napster. Cette architecture se compose d'un unique serveur, dont le but est de recenser les fichiers proposés par les différents clients comme le montre la figure 2. Contrairement au mode client/serveur, ce serveur centralisé ne dispose pas des fichiers. Il dispose principalement de deux types d'informations : celles sur le fichier (nom, taille, ...), et celles sur l'utilisateur (nom utilisé, IP, nombre de fichiers, type de connexion, etc.)

Du côté du client, rien de plus simple : une fois connecté grâce au logiciel spécifique, il peut faire des recherches comme avec un moteur classique. On obtient alors une liste d'utilisateurs disposant de la ressource désirée. [3]

³ Streaming : permet la lecture d'un flux audio ou vidéo à mesure qu'il est diffusé.

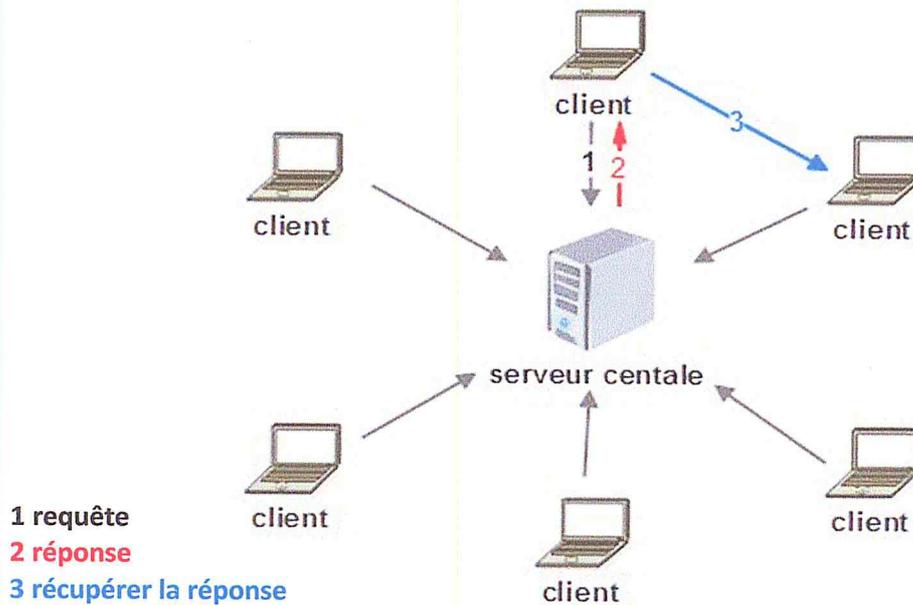


Figure 2 : l'architecture centralisée [3]

Avantages :

- Le confort d'utilisation est grand, puisqu'il n'y a pas de soucis de connexion au bon serveur.
- La recherche des documents est facilitée. En effet, le serveur est omniscient : si un fichier est disponible sur le réseau, on est en mesure de le savoir systématiquement. [3]

Inconvénients :

- Il suffit de bloquer le serveur pour déconnecter tous les utilisateurs et stopper le fonctionnement de l'ensemble du réseau.
- Sensible au partitionnement du réseau (serveur inatteignable) et aux attaques.
- Aucun anonymat n'est possible, puisque chaque utilisateur est identifié sur le serveur. [3]

2.2.2 L'architecture distribuée ou décentralisée

Une autre architecture de P2P est l'architecture décentralisée. Gnutella est un des représentants de cette architecture p2p. Ici il n'y a plus de serveurs centraux, ce sont tous les éléments du réseau qui vont jouer ce rôle. Chaque machine dans ses rôles est identique à une autre, c'est pour cela que l'on appelle ces types de réseaux purs peer to peer.

Lorsqu'un client souhaite se connecter, il va donc être nécessaire d'envoyer un message broadcast afin de savoir quelles autres personnes du réseau sont actives. Seules ces personnes répondant au message broadcast⁴. On est alors connecté au réseau. [3]

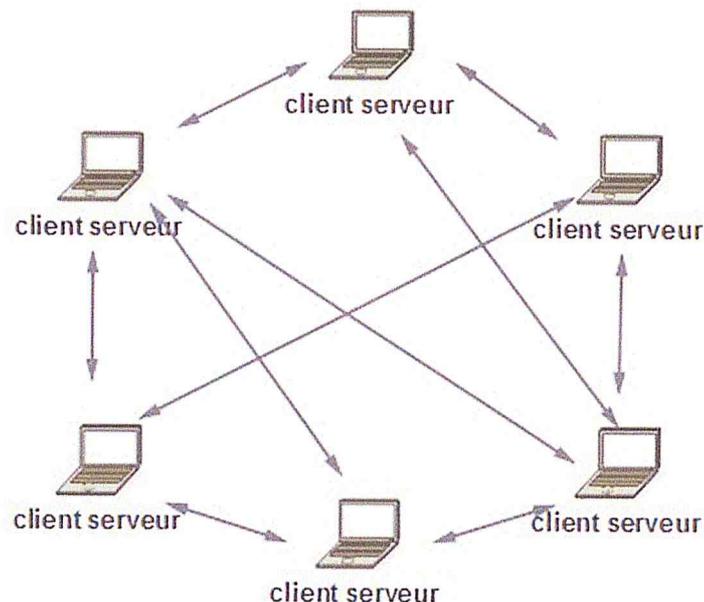


Figure 3 : l'architecture décentralisée [3]

Avantages :

- La taille d'un tel réseau est théoriquement infinie, contrairement aux autres architectures dont le nombre de clients dépend du nombre et de la puissance des serveurs.
- L'utilisation d'un tel réseau est anonyme, c'est-à-dire qu'il est impossible d'y repérer quelqu'un volontairement.
- Réseau très tolérant aux fautes [3]

Inconvénients :

- Gros consommateur de bande passante⁵
- Pas de garantie de succès, ni d'estimation de la durée des requêtes
- Pas de sécurité, ni de réputation (pas de notion de qualité des pairs, ni des données fournies).

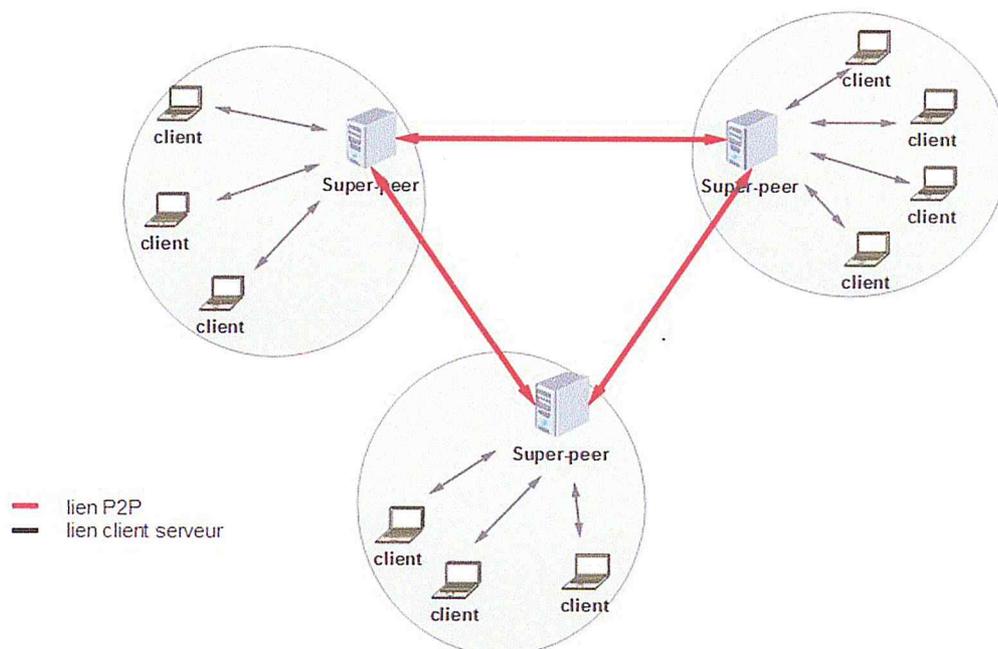
⁴ **Broadcast** : c'est un message envoyé à l'aveugle par un ordinateur vers tous les périphériques reliés avec lui.

⁵ **Bande passante** : c'est le taux de transfert de données pouvant être transporté d'un point à un autre dans un laps de temps données

- La recherche prend plus de temps car chaque requête sera adressée à chaque utilisateur connecté surtout si des milliers d'utilisateurs sont présents [3]

2.2.3 L'architecture hybride

Le réseau de ce type s'appuie sur un ensemble de serveurs (l'ordinateur d'un utilisateur peut devenir un serveur) appelé (super peer) gérant un groupe d'utilisateurs appelés (ordinary peer) suivant l'architecture centralisée. Puis, chaque super peer est ensuite connecté à d'autres super peer suivant l'architecture distribuée. De cette façon, si un fichier recherché par un utilisateur n'est pas indexé par le super peer auquel il est rattaché, celui-ci transmet alors la requête à un



autre super peer. [4]

Figure 4: l'architecture hybride [3]

Avantages :

- utilise les avantages des 2 types de réseaux (centralisé et décentralisé)
- permet de bénéficier d'une meilleure bande passante en réduisant le trafic de requêtes.
- permet de diminuer le nombre de connexions sur chaque serveur, et ainsi d'éviter les problèmes de bandes passantes.
- Le réseau n'est plus pollué par les trames de broadcast

- L'extensibilité c'est-à-dire un nouvel utilisateur a juste besoin de se connecter à un nœud réseau pour devenir à son tour un nœud du réseau [5]

Inconvénients :

- Plus complexe à mettre en œuvre
- L'anonymat n'est pas assuré [5]

3 Protocoles

Le transfert de données sur Internet s'effectue par paquets de données. Cette structure repose sur l'utilisation de protocoles TCP/IP (Transport Control Protocol/ Internet Protocol).

Chaque document, qu'il s'agisse de texte, image ou voix, est numérisé puis réparti en paquets. Chacun de ces paquets est alors envoyé sur Internet indépendamment des autres et essaie de prendre le chemin le plus rapide pour parvenir à sa destination. Ceci est réalisé en fonction de l'encombrement d'une partie ou de l'autre du réseau au moment où le paquet est expédié. Dans le reste de cette section, nous ferons un bref descriptif des protocoles de transport temps réel RTP et RTCP :

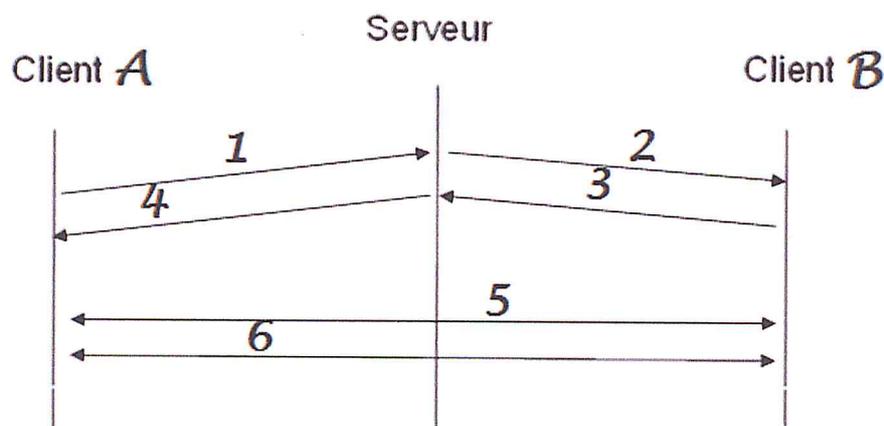
3.1 Protocole RTP

Le groupe de l'IETF a développé en 1993 le protocole de transport en temps réel (RTP, RFC 1889) dont le but est de transmettre sur Internet des données qui ont des propriétés temps réel (audio, vidéo...) [8]. C'est un protocole de la couche application du modèle OSI et utilise les protocoles de transport TCP ou UDP, mais, généralement, il utilise UDP qui est mieux approprié à ce genre de transmission.

Le rôle principal de RTP consiste à mettre en œuvre des numéros de séquence de paquets IP pour reconstituer les informations de voix ou vidéo même si le réseau sous-jacent change l'ordre des paquets. Plus généralement, RTP permet :⁶

- d'identifier le type de l'information transportée,
- d'ajouter des marqueurs temporels et des numéros de séquence l'information transportée
- de contrôler l'arrivée à destination des paquets.

⁶ <https://www.commentcamarche.com/les-protocoles-rtp-rtcp>



Le WebRTC comprend une construction triangulaire qui implique un serveur et deux clients.

Figure 5 : Etablissement d'une connexion entre deux clients utilisant WebRTC

La Figure ci-dessus montre l'établissement d'une connexion entre deux clients utilisant WebRTC. A et B exécutent une application WebRTC à partir d'un navigateur web [9]

- 1 : A demande au serveur une connexion avec B.
- 2 : Le serveur relaie la demande de A à B.
- 3 : Si B accepte, il envoie une demande de connexion à A.
- 4 : Le serveur relaie la demande à A.
- 5 et 6 : Les Peer Connexion bidirectionnelles sont établies.

5 Signalisation

La signalisation est le mécanisme qui gère des ressources nécessaires pour l'établissement d'un canal de communication et permet d'envoyer et recevoir des messages de contrôle. Pour que deux clients puissent communiquer, ils doivent se mettre d'accord sur le type de média à échanger et un canal commun. Si le client A souhaite communiquer avec le client B, alors ce dernier doit fournir une « adresse » à laquelle le client A peut le joindre.

Avant de pouvoir communiquer directement, de pair à pair, les clients WebRTC doivent obligatoirement échanger 3 types d'informations qui aideront à établir la communication qui sont les messages de contrôle de session (initialiser les communications et rapporter les erreurs), les

De plus, RTP peut être véhiculé par des paquets multicast afin d'acheminer des conversations vers des destinataires multiples.

3.2 Protocole RTCP

RTCP (Real Time Control Protocol, RFC 1889) est un protocole de contrôle utilisé conjointement avec RTP pour contrôler les flux de données et la gestion de la bande passante. [8] RTCP permet de contrôler le flux RTP, et de véhiculer périodiquement des informations de bout en bout pour renseigner sur la qualité de service de la session de chaque participant à la session.

Des quantités telles que le délai, les paquets reçus et perdus sont très importants pour évaluer la qualité de service de toute transmission et réception temps réelle.

4 WebRTC ⁷

WebRTC (Web Real-Time Communication, littéralement « communication en temps réel pour le Web ») est une interface de programmation (API) JavaScript développé au sein du W3C et de l'IETF. Cette technologie offre une communication temps réel de manière native à partir d'un navigateur Web. Les navigateurs web s'échangent directement des flux de données multimédias, aucun serveur intermédiaire n'est impliqué.

Bien que la communication de base WebRTC utilise le mode pair à pair, l'étape initiale d'établissement de cette communication requiert de la coordination. Cette coordination est fournie par un serveur de signalisation. Cela permet à deux ou plusieurs navigateurs Web disposant de la capacité WebRTC de se joindre, d'échanger les informations de contact, de négocier une session qui définit la manière dont ils vont communiquer, puis finalement d'établir les canaux média pair à pair pour le transport des flux média échangés directement entre eux.

⁷ <https://www.effort.com/signalisation-dans-larchitecture-WebRTC>

capacités "médias" (résolutions et codecs⁸ supportés par les devises / browsers, etc.), les configurations réseau pour le monde "extérieur", (adresse IP, ports, etc.).

5.1 Protocoles de signalisation :

Plusieurs protocoles de signalisation existent et le plus utilisé dans l'Internet est le protocole SIP. Ce dernier utilise SDP pour décrire les détails de l'appel, sous un modèle d'offre-réponse. Nous présenterons dans ce qui suit le protocole d'initialisation de session (SIP) et le protocole de description de session (SDP).

5.1.1 Protocole SDP

Spécifié par l'IETF, SDP (session description protocol) est un protocole de description, conçu à l'origine pour être utilisé par des applications de diffusion. Son usage s'est généralisé avec le protocole de signalisation SIP [10] et il est utilisé par le WebRTC au niveau du serveur de signalisation.

SDP permet de décrire les propriétés des sessions multimédias, les différents types de médias supportés et les propriétés d'un point de communication : son adresse IP, le port d'écoute, les codecs à utiliser ainsi que les différents candidats-relais (adresse, IP, protocole, et port, etc.) pour établir la communication.

5.1.2 Protocole SIP

SIP (Session Initiation Protocol) est un protocole de signalisation de la couche applicative du modèle OSI défini par l'IETF (Internet Engineering Task Force) permettant l'établissement, la libération et la modification de sessions multimédias.

Il se charge de l'authentification et de la localisation des multiples participants. Il se charge également de la négociation sur les types de médias utilisables par les différents participants en encapsulant des messages SDP (Session Description Protocol).

SIP ne transporte pas les données échangées durant la session comme la voix ou la vidéo. Ce protocole étant indépendant de la transmission de tous type de données car de protocoles conçus spécifiquement pour être utilisé pour cet échange [11].

⁸ Codec : dispositif matériel ou logiciel permettant de mettre en œuvre l'encodage ou décodage d'un flux de données numérique

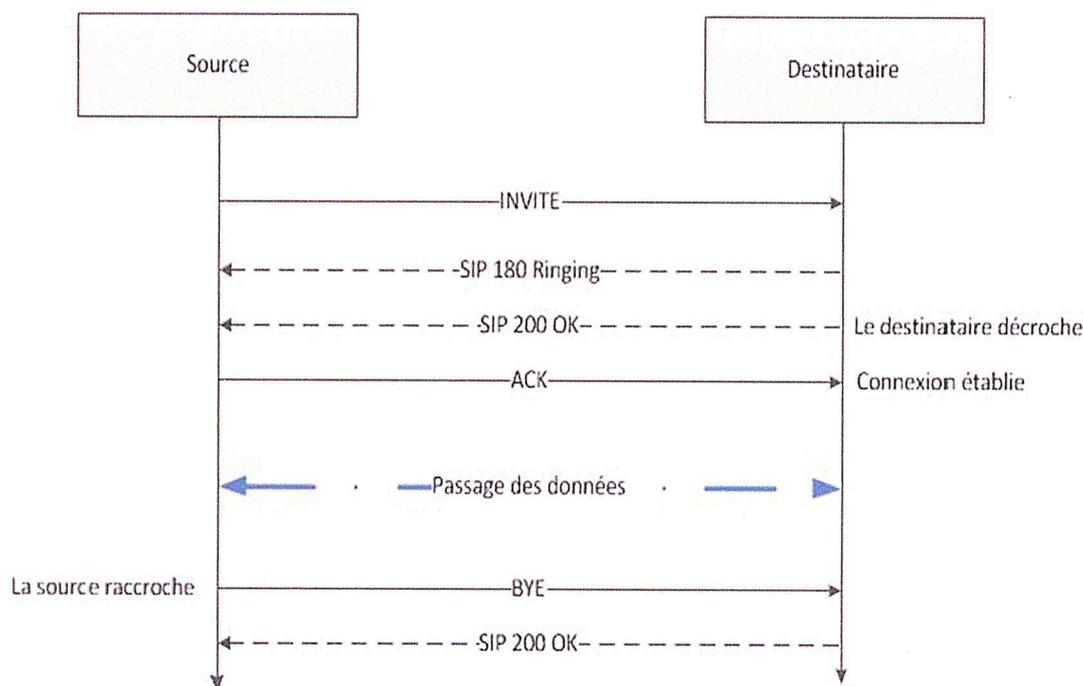


Figure 6 : communication avec le protocole SIP

La figure 6 ci-dessus représente un exemple de communication avec le protocole SIP, une source appelle son destinataire qui répond d'abord par une réponse provisoire. Une fois que les deux agents ont répondu positivement, une session est ouverte et laisse place à un échange de données, entre autres les flux vidéo et audio, qui se fera grâce à l'aide du protocole de transport choisi. Quand l'un des deux raccroche, un signal est envoyé à l'autre de façon à l'avertir que la communication a été arrêtée.

6 Pile protocolaire

Pour récapituler l'ensemble de protocoles que nous avons déjà cités, telle que les protocoles de transports temps réel RTP et RTCP et les protocoles de signalisation SDP, SIP. Nous allons les représenter dans la figure ci-dessous selon leurs localisations dans les couches du modèle TCP/IP

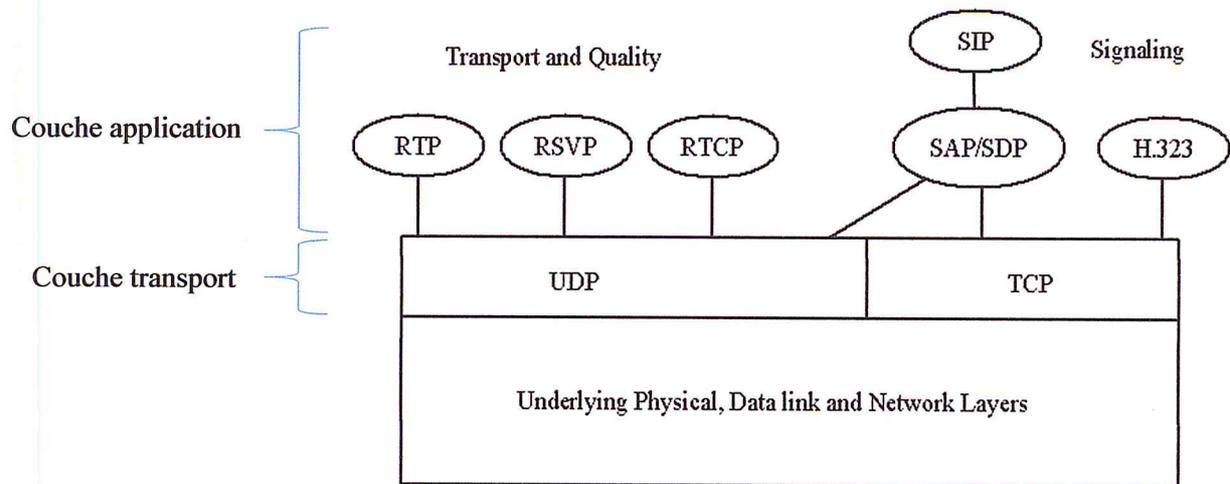


Figure 7 : pile de protocoles de transport et signalisation [12].

7 Conclusion :

Dans ce chapitre nous avons présenté les protocoles de transport ainsi que les architectures réseau les plus répandues : le modèle client/serveur et le modèle pair à pair. Nous avons aussi présenté la technologie WebRTC en se basant sur la partie de signalisation dont on va s'inspirer pour la mise en œuvre de notre projet.

Dans le prochain chapitre nous allons présenter les systèmes webinaires en détaillant ses fonctionnalités et quelques exemples d'applications.

Chapitre 2 : Systemes webinaires

1 Introduction

La popularité des applications de webinaire et vidéoconférence a augmenté de manière significative ces dernières années. Les outils de webinaires sont devenus une forme de communication indispensable et attirent de plus en plus l'attention avec l'avancement des technologies de l'apprentissage en ligne. Ces dernières facilitent la communication en temps réel et enrichissent l'interactivité dans un environnement d'apprentissage en ligne.

La mise en œuvre des applications webinaires est très difficile, car elles doivent soutenir simultanément un grand nombre de participants, avec des changements dynamiques à l'adhésion des participants et faire face à l'exigence de bande passante élevée.

Dans ce chapitre nous abordons la signification du webinaire ses fonctionnalités ses éléments, ainsi que quelques architectures utilisées dans les applications webinaires et leurs étude critique.

2 Définitions

La communication est le fait d'établir une relation ou une liaison avec autrui et de leur transférer, partager et diffuser des informations à travers la linguistique, elle représente le pilier des séminaires soit de façon classique c'est-à-dire séminaires dans des salles physiques ou bien séminaires virtuels via internet.

Dans cette section nous allons définir des notions ayant une relation avec les conférences.

2.1 Séminaire⁹

Le séminaire est une pratique de l'enseignement supérieur, à l'université, dans un organisme de formation, dans un groupement professionnel. Il s'agit de réunir un petit groupe dans des rencontres régulières, centrées chaque fois sur un sujet particulier, où on demande à tous une participation active.

Le séminaire universitaire peut prendre la forme d'un dialogue socratique mené par l'instructeur, ou d'une présentation plus systématique du résultat de recherches préparatoires. En

⁹ [https://fr.wikipedia.org/séminaire_\(enseignement\)](https://fr.wikipedia.org/séminaire_(enseignement))

principe, les participants ne sont pas debutants dans le domaine en discussion, et a l'universite les seminaires sont generalement reserves aux etudiants avances.

2.2 Voix sur IP

La voix sur IP ou « VoIP » pour voice over IP designe tous les types de communication vocale utilisant la technologie IP (Internet Protocol) au lieu de la technologie traditionnelle a commutation de circuits.¹⁰

Longtemps confinee au transport de la voix, la VoIP diffuse aujourd'hui les appels audio et video ainsi que la messagerie instantanee par le biais d'applications dediees telles que Skype, Facebook Messenger, WhatsApp . [13]

En ce qui concerne la telephonie uniquement, cette technologie est complementaire de la telephonie sur IP (« ToIP » pour Telephony over Internet Protocol), qui concerne les fonctions realisees par un autocommutateur telephonique IPBX.¹¹

2.3 Web conference¹²

C'est une conference en ligne et une application internet qui offre la possibilite d'organiser des conferences, des reunions de travail ou des formations virtuelles avec des personnes distantes. C'est un outil de collaboration synchrone (tous les participants voient et entendent la meme chose au meme moment).

Differeents canaux de communications peuvent etre etablis entre les participants : une liaison audio (VoIP), une liaison data (presentation de donnees de toutes sortes ou partage d'applications) et parfois une liaison video.

2.4 Webinaire

Le webinaire ou sa version anglophone « webinar » est la combinaison des mots web et seminaire, ce qui signifie simplement un seminaire sur internet.

C'est un evenement d'apprentissage en ligne visant a transmettre des connaissances et des competences a un public cible, il offre un melange de l'audio, de la video, du texte et d'autres

¹⁰ <https://www.3cx.fr/voip-sip/voip-definition/>

¹¹ https://fr.wikipedia.org/Voix_sur_IP

¹² <https://www.techno-science.net/definition>

formes d'interactions en directe. Il est organise autour de strategie d'apprentissage specifique, identifiee a l'avance, et qui constitue la base de contenu de webinaire pertinent.

Les webinaires ont ete principalement utilises par les entreprises business to business (B2B)¹³, et recemment ont ete introduits dans le domaine de l'education, et sont tous pret a apporter une revolution dans le secteur de l'education.

3 Webinaires

Dans la section precedente nous avons parle des differents types de conferences, vu que notre travail est base sur les webinaires dans la suite de notre memoire nous allons se baser sur ce type de systeme.

3.1 Elements d'un webinaire

Un webinaire est compose de plusieurs entites indispensables parmi eux : [14]

- **L'organisateur** : L'organisateur est la personne qui ordonne et gere la reunion, notamment en attribuant les droits de parole et d'intervention des participants. Il peut aussi creer des salles privees, ou un nombre restreint de participants se retrouve en toute confidentialite. Selon l'importance de la reunion et sa complexite, l'organisateur a un role purement technique de gestion ou d'animation en temps reel.
- **Les participants** : Conviés a la reunion par l'organisateur, ils ont deux manipulations a effectuer pour participer se connecter a un site internet et rejoindre la salle de reunion virtuelle. En fonction des droits accordés par l'organisateur, ils seront simples spectateurs, ou pourront prendre la parole, voire modifier des documents.
- **La salle de conference virtuelle** : La quasi-totalite des systemes de webinaire propose des salles de reunion virtuelles sur leur site. Ne peuvent y acceder que les personnes possédant un droit d'entree, exceptees pour les reunions ouvertes. L'organisateur dispose, dans cette salle, d'un tableau de commande de la reunion. Les participants ont acces a differents outils (demande de prise de parole, bulletin de vote, acces aux documents, etc.).

¹³ **B2B** : l'ensemble des relations commerciales entre les entreprises et les professionnels, on l'appelle aussi « commerce interentreprises »

- **Les documents partagés :** Les plus basiques sont des fichiers de présentation Powerpoint. Certaines web conférences permettent de montrer des textes ou des tableaux, du type Word ou Excel, afin de pouvoir les modifier en ligne. On entre alors dans la réunion collaborative.

3.2 Fonctionnalités des webinaires

La caractéristique clé d'un webinaire est la capacité de discuter et partager des informations. Amener des experts à interagir avec les étudiants à travers cette plate-forme collaborative virtuelle,

Les webinaires offrent des fonctionnalités qui sont citées ci-dessous : [15]

- La capacité de discuter et partager des informations.
- diffusion live (et si possible HD) d'une vidéo.
- Enregistrement de la présentation par le présentateur pour la réutiliser plus tard.
- Les participants peuvent poser leurs questions pendant la session sans perturber le flux de la session.
- L'interaction avec le présentateur en privé, ou avec d'autres participants, ou même avec tous les participants.
- Mener un sondage d'opinion.
- Le présentateur peut répondre aux questions des participants pendant ou après la fin de la conférence.
- envoyer des emails automatiques pour rappeler le webinaire.

3.3 Avantages des webinaires

L'utilisation des webinaires augmente de plus en plus au sein des entreprises et des universités par rapport au mode de conférences classiques grâce à ces avantages offerts dans le cadre professionnel et éducatif. Nous présentons ces avantages dans ce qui suit :

- Un gain de temps et d'argent ;
- L'audience n'est pas limitée par la taille de la salle ;
- Une interactivité accrue avec les participants ;
- Une possibilité de visionner la conférence si le webinaire est enregistré.

3.4 Exemples des systèmes webinaires

3.4.1 Skype :¹⁴

C'est un logiciel de visioconférence qui permet aux utilisateurs de passer des appels téléphoniques via Internet. Les appels d'utilisateur à utilisateur sont gratuits, tandis que ceux vers les lignes téléphoniques fixes et les téléphones mobiles sont payants.

Skype permet aussi la discussion instantanée et l'échange de fichier entre utilisateurs. Mais l'accès restreint aux seuls utilisateurs de Skype et de l'ensemble des fonctionnalités restent actuellement un obstacle face à d'autres logiciels concurrents de conférence Webinaire.

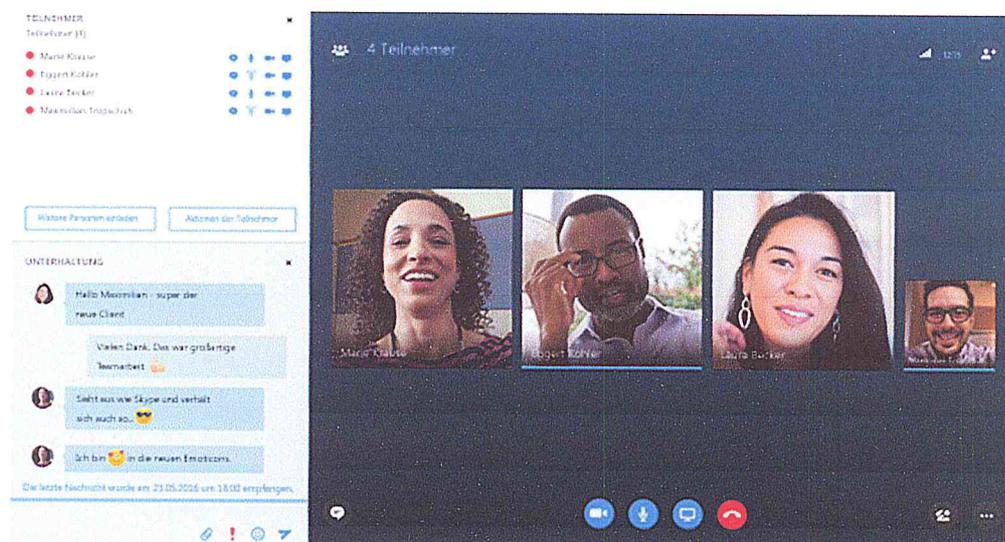


Figure 8 : interface de Skype

3.4.2 Google Hangouts¹⁵

Google Hangouts est un service de visioconférence et webinaire multi-participants intégré au réseau social Google Plus, également connecté à la messagerie Gmail, et disponible sous forme d'application mobile pour Android et iOS.

Les réunions virtuelles peuvent accueillir dix participants connectés via un compte Google. L'outil, permet également l'utilisation d'autres applications, par exemple pour le partage et la lecture de documents (SlideShare, Google Drive. etc.).

¹⁴ <https://www.skype.com>

¹⁵ <https://www.commentcamarche.com/google-hangouts>

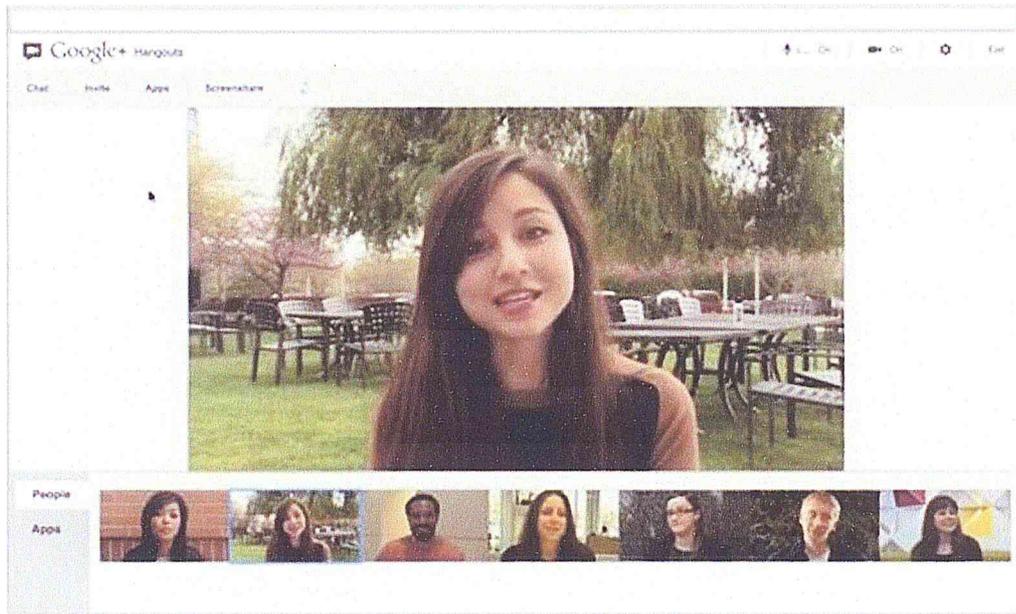


Figure 9 : Interface de Google Hangouts

3.4.3 iChat¹⁶

iChat est un logiciel de messagerie instantanée pour Mac OS X, édité par Apple, les messageries vocales et vidéo et les capacités de partage d'écran sont aussi des fonctionnalités d'iChat.

Cette application peut prendre en charge jusqu'à quatre personnes dans une conférence vidéo et dix personnes dans une conférence audio.



¹⁶ <https://fr.wikipedia.org/iChat>

Figure 10 : Interface d'iChat

3.4.4 WebEx :

Commercialisé par l'entreprise Cisco System la plateforme webEx est un outil qui permet d'organiser des webinaires et d'évènements en ligne à fort impact en reliant de vastes groupes de personnes géographiquement dispersées.

Cet outil est beaucoup plus utilisé par les grandes entreprise afin d'annoncer les dernières avancées de leurs produits quel que soit la région ou le fuseau horaire où ils se trouvent.

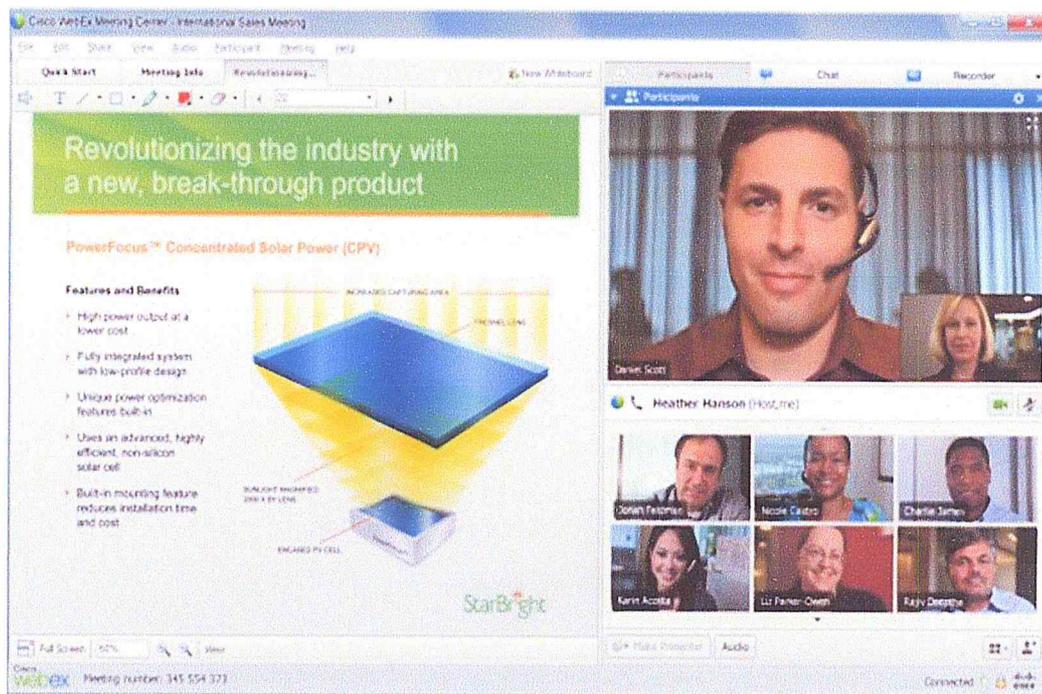


Figure 11 : Interface de WebEx

3.4.5 Ready talk

Ready Talk est l'un des principaux fournisseurs de solutions de réunions audio et Web intuitives à utiliser, simples à joindre et faciles à gérer.

Ce logiciel permet d'organiser des séminaires Web, d'organiser des réunions en ligne, d'organiser des conférences téléphoniques et d'enregistrer et de distribuer facilement des sessions tout en économisant du temps, il autorise entre 150 et 3 000 participants.

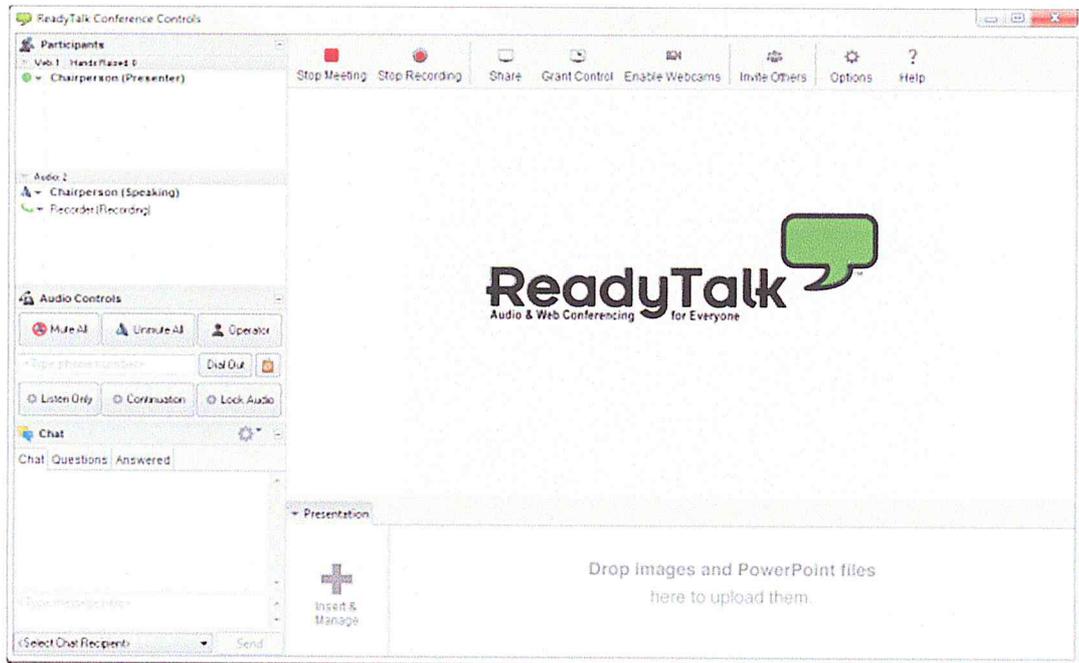


Figure 12 : Interface de Ready Talk

3.4.6 Adobe Connect

Adobe Connect est un outil très puissant de visioconférence qui assure différentes fonctionnalités selon la formule choisie parmi les 3 offres disponibles : *Adobe Connect Meetings* ; *Adobe Connect Learning* ; *Adobe Connect Webinars*.

Ces trois types d'offres sont le reflet des possibilités offertes par la solution d'Adobe :

- Faire des réunions en temps réel à plusieurs où on peut partager des documents et des écrans ;
- Présenter des cours à distance avec une partie de chat et d'interactions en temps réel réservée aux apprenants

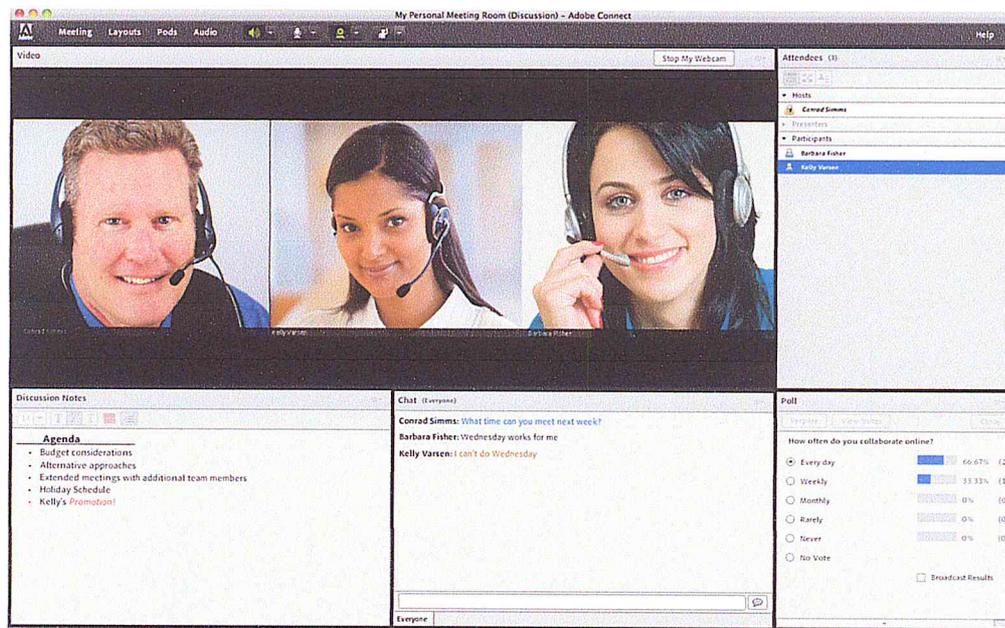


Figure 13 : Interface d'Adobe Connect

4 Etude de l'existant

Cette section a pour objectif d'étudier les architectures des outils de webinaires les plus utilisés. Cette étude permet de dégager les points forts et les points faibles de chacune de ces solutions.

Dans ce qui suit, nous présentons une analyse de l'existant, puis nous détaillons la critique de l'existant.

4.1 Analyse de l'existant

Les systèmes webinaires adoptent généralement les architectures des réseaux Peer to Peer, cependant ces derniers présentent des inconvénients liés à la consommation abusive de la bande passante et la surcharge sur le diffuseur de la conférence.

Dans le but de résoudre ces inconvénients nous allons citer des exemples des systèmes qui utilisent les différents types d'architectures des réseaux Peer to Peer, tels que Skype, Google Hangouts et iChat.

4.1.1 Architecture de Skype

Le système Skype est un mélange de VoIP et pair à pair et s'appuie précisément sur l'architecture hybride comme le montre la figure 14 ci-dessous.

Il existe deux types de nœuds au sein du réseau Skype, les nœuds ordinaires et les super-nœuds (SN). Un nœud ordinaire est une application Skype qui peut être utilisée pour passer des appels vocaux et envoyer des messages texte. Un super nœud est le point d'arrivée d'un nœud ordinaire sur le réseau Skype. Tout nœud avec une adresse IP publique ayant suffisamment de CPU¹⁷, de mémoire et de bande passante réseau est un candidat à devenir un super nœud. Un hôte ordinaire doit se connecter à un super nœud et doit s'enregistrer avec le serveur de connexion Skype pour une connexion réussie.

Une autre entité importante du réseau Skype est le serveur de login. Tous les clients doivent s'authentifier auprès de celui-ci pour s'assurer que les identifiants soient uniques sur le réseau. Ce serveur est la seule entité centralisée du protocole Skype. [16]

Skype utilise généralement UDP pour transmettre l'audio et la vidéo et le change vers TCP si UDP est bloqué. Différent De Google+ et iChat, Skype n'utilise pas RTP et RTCP.

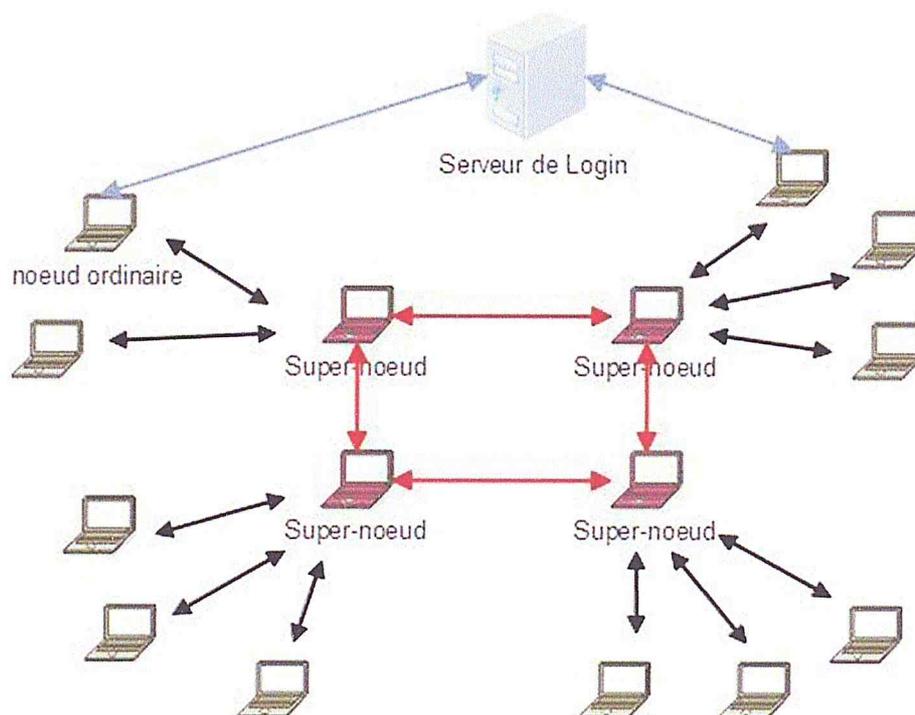


Figure 14 : L'architecture Skype [17]

¹⁷ CPU : Central Processing Unit, c'est le coeur de l'ordinateur chargé de traiter et d'exécuter les instructions ou des programmes

4.1.2 Architecture Hangout Google +

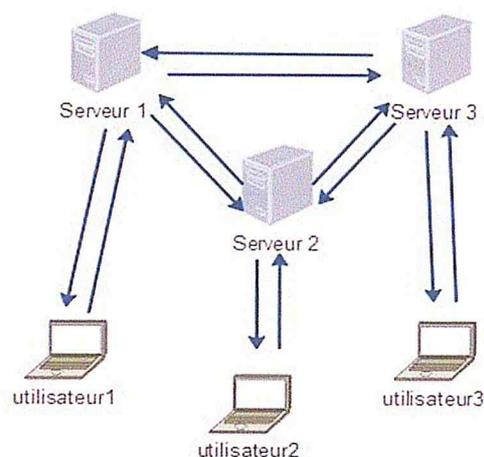
Dans l'architecture de Google+ figure 15, Les appels vidéo entre deux utilisateurs et entre plusieurs utilisateurs, utilisent toujours une topologie centralisée sur le serveur.

Chaque utilisateur envoie sa voix et sa vidéo à un seul serveur proxy dédié et reçoit également la voix et la vidéo d'autres utilisateurs à partir de ce serveur.

Il n'y a pas de transmissions directes entre les utilisateurs. Généralement, différents utilisateurs choisissent différents serveurs proxy. Ainsi, les serveurs proxy doivent communiquer entre eux pour échanger les voix et les vidéos des utilisateurs.

Chaque utilisateur ouvre quatre connexions avec son serveur proxy sur le même port de serveur (Port 19305). La plupart du temps, ces quatre connexions utilisent toutes UDP. TCP est utilisé seulement lorsque nous bloquons délibérément le trafic UDP.

Google+ utilise également le protocole RTP pour transmettre la voix et la vidéo. Les deux autres flux sont conformes au format du protocole RTCP. [17]



La figure ci-dessous représente l'architecture de Hangouts :

Figure 15 : L'architecture de Hangouts [17]

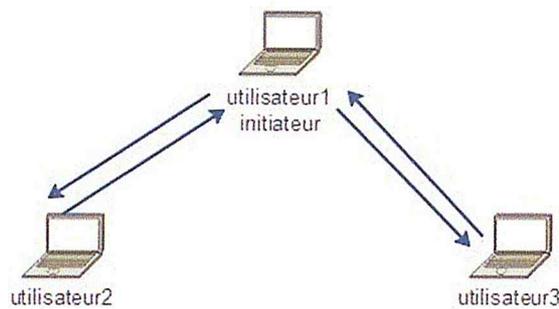
4.1.3 Architecture iChat

iChat utilise une architecture P2P, les utilisateurs d'une conférence forment une topologie en étoile à la couche d'application comme indiqué dans la figure 16.

Le concentrateur central est l'initiateur de la conférence, c'est-à-dire l'utilisateur qui a initié la conférence. Seul l'initiateur de la conférence a le droit d'ajouter des personnes à la conférence ou de fermer la toute la conférence.

Un utilisateur normal envoie ces données voix et vidéo à l'initiateur grâce à un flux UDP et reçoit les données vocales et vidéo des autres participants à partir de l'initiateur grâce à un autre flux UDP, les participants utilisent le port UDP 16402. La voix et la vidéo sont transportées en utilisant le protocole RTP.

Les utilisateurs normaux se connectent uniquement à l'initiateur [17]. Il n'y a pas de connexion directe entre deux utilisateurs normaux. iChat ne peut pas fonctionner si le trafic UDP est bloqué.



La figure ci-dessous représente l'architecture d'iChat :

Figure 16 : L'architecture d'iChat [17]

4.2 Etude critique

Le tableau suivant récapitule les avantages et les inconvénients de tous les outils traités

| Architecture de l'outil | Avantages | Inconvénients |
|-------------------------|--|--|
| Skype | <ul style="list-style-type: none"> - Bénéficie d'une meilleure bande passante - N'utilise pas les trames de broadcast - Le trafic de requête est réduit | <ul style="list-style-type: none"> - Risque de défaillance d'un super-nœud. - Le serveur login risque de s'arrêter donc perte de données |
| Hangout Google + | <ul style="list-style-type: none"> - Offre une vue d'ensemble très complète du réseau. | <ul style="list-style-type: none"> - Le risque de surcharge sur le serveur centrale qui peut devenir |

| | | |
|-------|---|---|
| | | <p>un point de defaillance du systeme.</p> <ul style="list-style-type: none"> - Il suffit de supprimer le serveur pour que l'integralite du reseau soit inactive. |
| iChat | <ul style="list-style-type: none"> - Si une machine tombe en panne, cela ne remet pas en cause l'ensemble du systeme. - Le reseau est faiblement couplé | <ul style="list-style-type: none"> - Ne peut pas fonctionner si le trafic UDP est bloqué - La securite est tres peu presente - Consomme beaucoup de bande passante |

Tableau 1 : Comparaison des architectures des systemes webinaires

On remarque du tableau que chaque architecture a ces limites et ces points forts par rapport aux autres architectures. Dont on ne peut pas dire qu'une architecture est bonne par rapport a une autre, ce qui nous mène a la création de notre propre architecture qui répond à notre problématique et à nos besoins.

5 Conclusion

Dans ce chapitre, nous avons défini le champ de notre étude suivi d'une étude de l'existant afin de préciser nos objectifs à atteindre pour résoudre le problème liée à la gestion des canaux de communications qui requiert la disponibilité d'une bande passante importante au niveau du centre de diffusion. De plus les connexions reliant les universités au centre de diffusion se saturent rapidement en présence d'un nombre important de participants.

Cette étude nous a aidés à choisir l'architecture qui nous convient pour la création de notre système de gestion des canaux de communications pour les webinaires. Dans notre solution nous envisageons que notre système se basera sur l'architecture hybride vue ses avantages liée à la bande passante et scalabilité. Il doit aussi gérer les connexions on se basant sur le fonctionnement du WebRTC dans la partie de signalisation. Enfin assurera l'acheminement des flux à travers plusieurs universités tout en passant par le chemin optimal.

Dans le chapitre qui suit nous présenterons les démarches de développement et de conception de notre système de gestion.

Chapitre 3 : Conception et proposition de solution

1 Introduction

La phase de conception représente une étape importante dans la réalisation d'un projet. Elle doit d'écrire de manière non ambiguë le fonctionnement futur du système, afin d'en faciliter la réalisation. Pour cela, différentes méthodes existent permettant de formaliser les étapes préliminaires du développement.

Dans ce chapitre nous décrivons la topologie du système après nous déterminerons le langage utilisé pour spécifier cette conception ainsi que les fonctionnalités de notre application ensuite nous détaillerons les différents diagrammes qui modélisent notre application (diagrammes de cas d'utilisation, diagrammes de séquence et diagramme de classe), enfin nous ferons une description de notre solution proposée.

2 Topologie du système « UnivWebinar »

Notre système de gestion des canaux de communication pour les webinaires se déploie sur une architecture réseau qui associe le modèle centralisé et distribué peer-to-peer. Il s'agit en effet d'une architecture hybride, qui a été choisi vue ses avantages cités dans le premier chapitre.

Le réseau est constitué d'un serveur principal centralisé représenté par le serveur de signalisation où tous les éléments du système doivent passer par ce dernier avant d'effectuer une communication entre eux, le réseau est également constitué d'un ensemble de clients par rapport au serveur de signalisation (des participants ou conférenciers) et un autre type de client sous forme de nœud (MCU)¹⁸. Un MCU peut jouer au même temps le rôle d'un serveur par rapport aux conférenciers, aux participants, et d'autres MCUs pour véhiculer les flux multimédias.

Les échanges entre les différents clients et le serveur de signalisation s'effectuent par socket avec le protocole TCP pour garantir le transfert de données de façons fiable et permettre une communication bidirectionnelle.

Cependant, les échanges entre les participants ou les conférenciers avec le nœud et entre les nœuds s'effectuent par socket avec le billet du protocole UDP, ce choix a été pris pour la

¹⁸ **MCU** : une machine servant à établir simultanément plusieurs communications de visioconférence ou de VoIP.

diffusion des conférences car UDP offre une performance en termes de délais, sachant qu'on peut tolérer des pertes de paquets.

La figure 17 montre l'architecture de notre système, et les interactions entre ses différents composants :

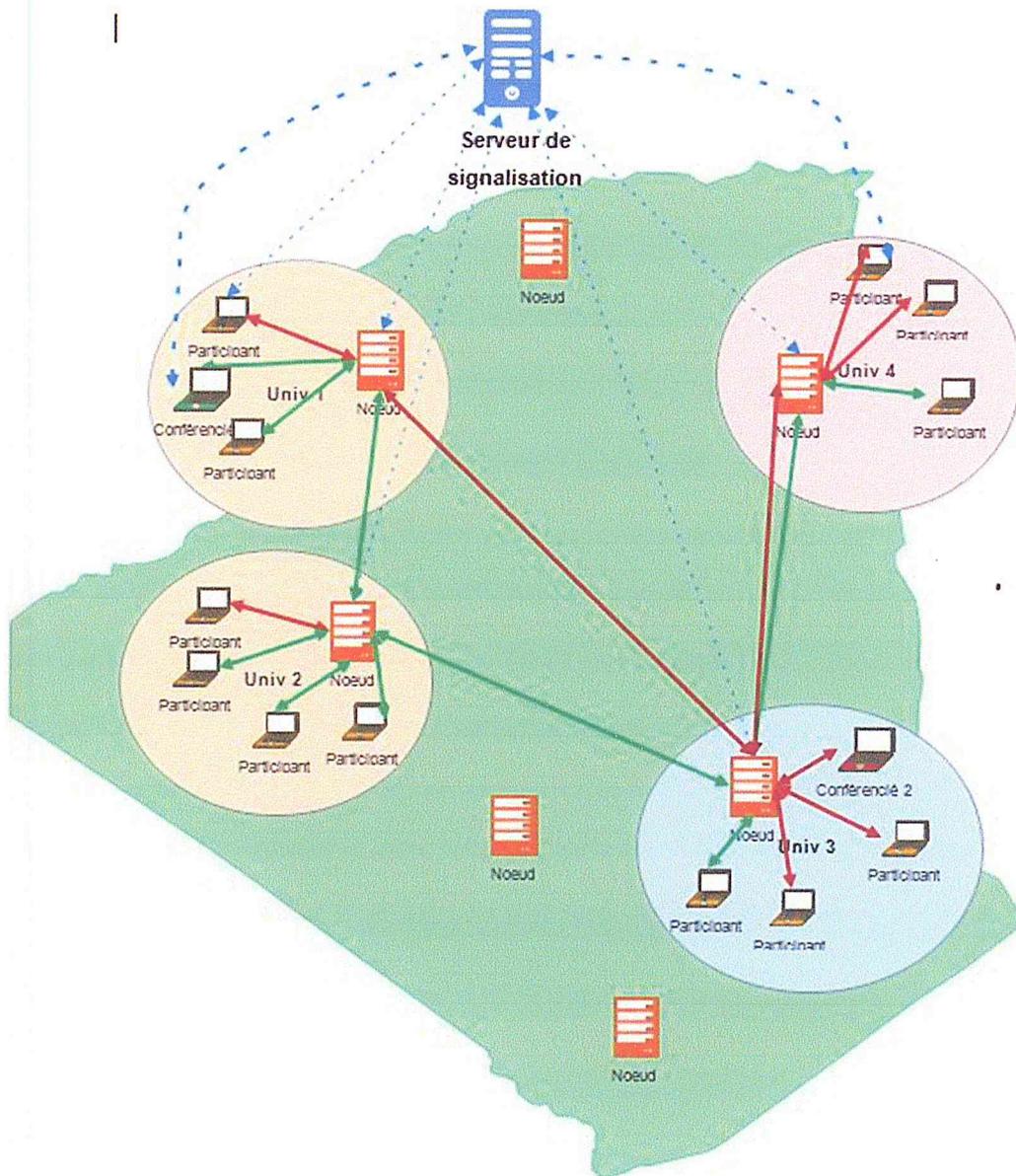


Figure 17 : Topologie du système « UnivWebinar »

Notre système est composé de trois applications, l'application du serveur de signalisation, l'application du nœud, et celle du conférencier(ou participant), ainsi des liens de communications qui relient ces applications :

- L'application serveur de signalisation représente le cœur du système, où toutes les autres applications communiquent avec lui via des Socket TCP.
- L'application nœud représente le point d'arrivée des participants ou des conférenciers et considéré comme le centre de diffusion d'une université particulière.
- L'application webinaire est composée de deux types d'utilisateurs qui sont les conférenciers ou les participants
- Un lien bidirectionnel entre l'application du serveur de signalisation et les deux autres applications via Socket TCP
- Un lien bidirectionnel entre le nœud et les conférenciers (ou les participants) via Socket UDP.
- Un chemin optimal pour la diffusion des conférences, démarrant du conférencier et passant entre les différents nœuds (université) vers tous les participants de la conférence.

3 Langage UML

UML (Unified Modeling Language) se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions, et communiquer des points de vue.

UML est fondé sur des concepts orientés objets, il propose de décrire un système à l'aide de quatorze types de diagrammes regroupés en deux grandes classes : des diagrammes structurels et des diagrammes comportementaux figure 18. [18]

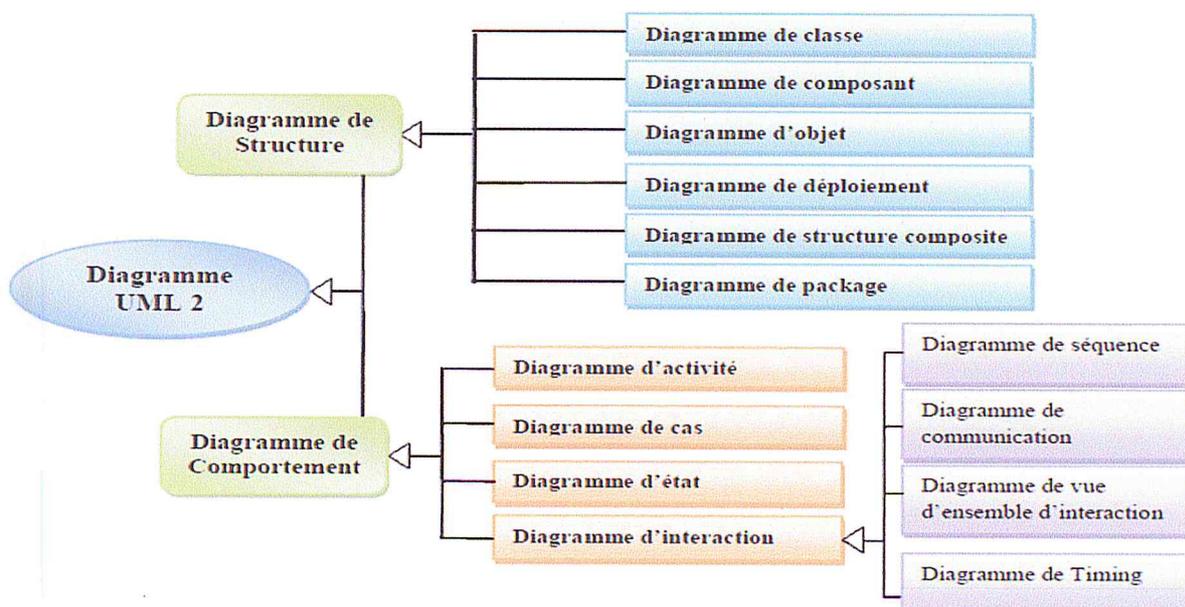


Figure 18 : Les diagrammes UML

La figure illustre les quatorze types de diagrammes UML qui sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.

3.1 Intérêt d'UML

UML est un langage semi-formel et normalisé [19]:

- Gain de précision.
- Gagne de stabilité.
- Encourage l'utilisation d'outils.

UML est un support de communication performant :

- Il cadre l'analyse.
- Il facilite la compréhension de représentations abstraites complexes.
- Son caractère polyvalent et sa souplesse en font un langage universel.

4 Processus de développement

Un processus définit une séquence d'étapes, partiellement ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. L'objet d'un processus de développement est de produire des logiciels de qualité qui répondent aux besoins de leurs utilisateurs dans des temps et des coûts prévisibles [18].

Plus simplement, un processus doit permettre de répondre à la question fondamentale : « Qui fait quoi et quand ? ».

4.1 Processus unifié

Le Processus Unifié (UP, pour Unified Process) est un processus de développement logiciel « itératif et incrémental, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques »

- Itératif et incrémental : le projet est découpé en itérations de courte durée (environ 1 mois) qui aident à mieux suivre l'avancement global. À la fin de chaque itération, une partie exécutable du système final est produite, de façon incrémentale.

- Centré sur l'architecture : tout système complexe doit être décomposé en parties modulaires afin de garantir une maintenance et une évolution facilitées. Cette architecture (fonctionnelle, logique, matérielle, etc.) doit être modélisée en UML et pas seulement documentée en texte.
- Piloté par les risques : les risques majeurs du projet doivent être identifiés au plus tôt, mais surtout levés le plus rapidement possible. Les mesures à prendre dans ce cadre déterminent l'ordre des itérations.
- Conduit par les cas d'utilisation : le projet est mené en tenant compte des besoins et des exigences des utilisateurs. Les cas d'utilisation du futur système sont identifiés, décrits avec précision et priorisés [18].

5 Conception de l'application UnivWebinar

Dans cette partie nous aborderons la partie conceptuelle de notre système «UnivWebinar» qui utilise l'architecture hybride, il se compose de 3 applications réparties qui sont : l'application du serveur de signalisation, l'application du nœud et l'application des participants (ou conférenciers).

Nous commencerons par les cas d'utilisation qui traduisent les exigences du système, suivis par des diagrammes de séquence système.

5.1 Spécification des besoins

L'application envisagée doit satisfaire les besoins fonctionnels qui seront exécutés par le système et les besoins non fonctionnels qui perfectionnent la qualité logicielle du système.

5.1.1 Besoins fonctionnels

Les besoins fonctionnels ou besoins métiers représentent les actions que le système doit exécuter, il ne devient opérationnel que s'il les satisfait. Cette application doit couvrir principalement les besoins fonctionnels suivants :

- Trouver le chemin optimal entre le conférencier et chaque participant.
- Ouvrir des canaux de communication UDP entre le conférencier et les participants passant par des nœuds.

- La diffusion de la vidéo du conférencier vers tous ses participants.
- La diffusion de plusieurs webinaire à la fois.
- Enregistrement des utilisateurs (participant, conférencier) dans une base de données.
- Authentification des utilisateurs.
- Envoi et réception des messages textes.
- Affichage de la liste des utilisateurs présents dans le webinaire.
- Affichage des webinaires disponibles.
- Limiter le nombre de participants supportés par chaque nœud (la charge).

5.1.2 Besoins non fonctionnels

Ce sont des exigences qui ne concernent pas spécifiquement le comportement du système mais plutôt identifient des contraintes internes et externes du système. Les principaux besoins non fonctionnels de notre application se résument comme suit :

- Le code doit être clair et bien commenté pour permettre des futures évolutions ou améliorations.
- Le code doit être modulaire pour faciliter la maintenance et la réutilisation.
- L'ergonomie : l'application offre une interface conviviale et facile à utiliser.
- La sécurité : l'application doit respecter la confidentialité des données.
- Élaborer un guide d'installation et d'utilisation.

5.2 Diagramme de cas d'utilisation

Le diagramme de cas d'utilisation représente la structure des grandes fonctionnalités nécessaires aux utilisateurs du système. C'est le premier diagramme du modèle UML, celui où s'assure la relation entre l'utilisateur et les objets que le système met en œuvre [20].

5.2.1 Identification des acteurs

Un acteur représente un rôle joué par un utilisateur humain ou un autre système qui interagit directement avec le système étudié. Un acteur participe à au moins un cas d'utilisation. Un acteur

peut consulter et/ou modifier directement l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données [20].

Dans notre cas le système est composé de 3 acteurs :

- **Administrateur du nœud** : activer le nœud, limiter le nombre des participants à supporter et diffuser les flux reçus à des participants spécifiques.
- **Conférencier** : publie et lance et anime un webinaire particulier.
- **Participant** : assiste à un webinaire particulier.

5.2.2 Cas d'utilisation

Chaque cas d'utilisation décrit un ensemble d'interactions successives d'une entité en dehors du système (utilisateurs) avec le système lui-même pour réaliser une fonctionnalité.

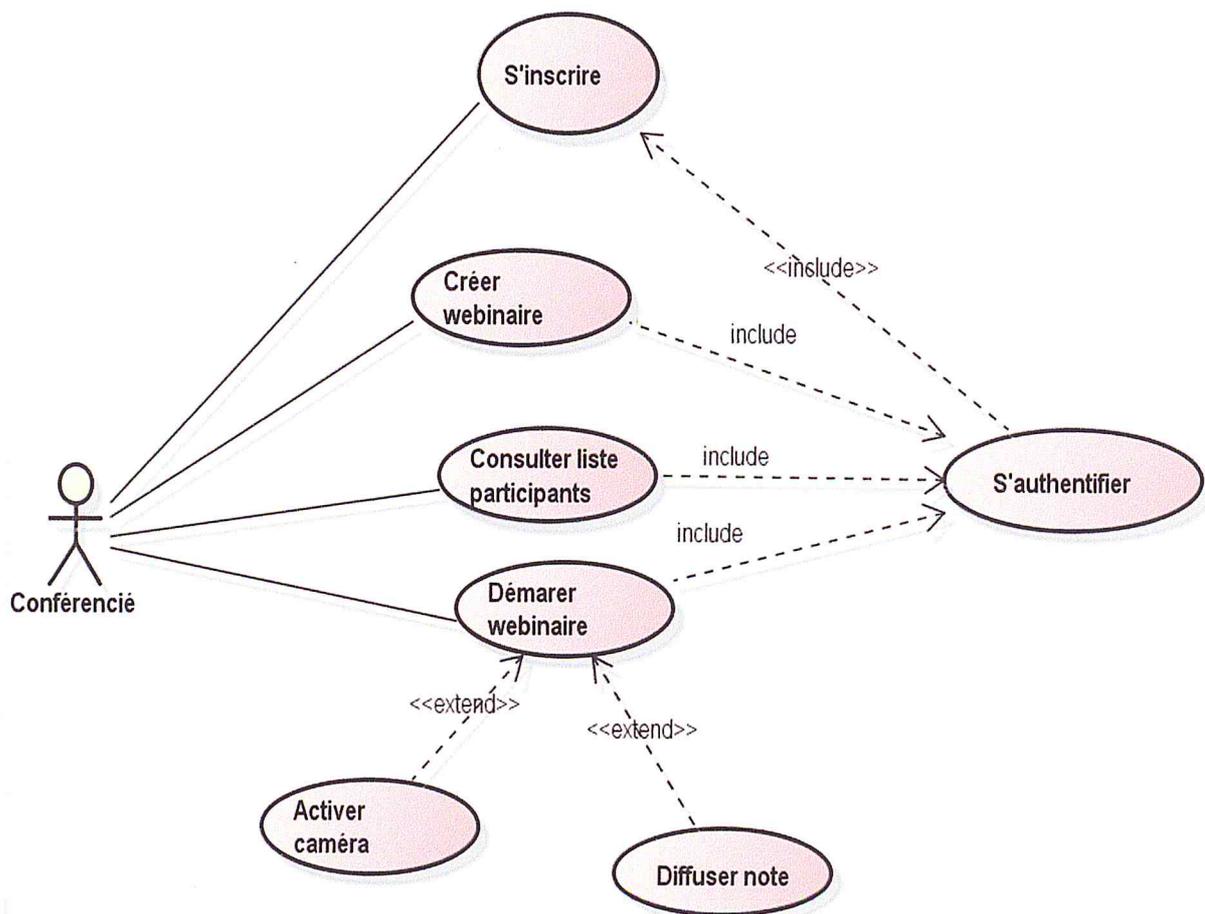
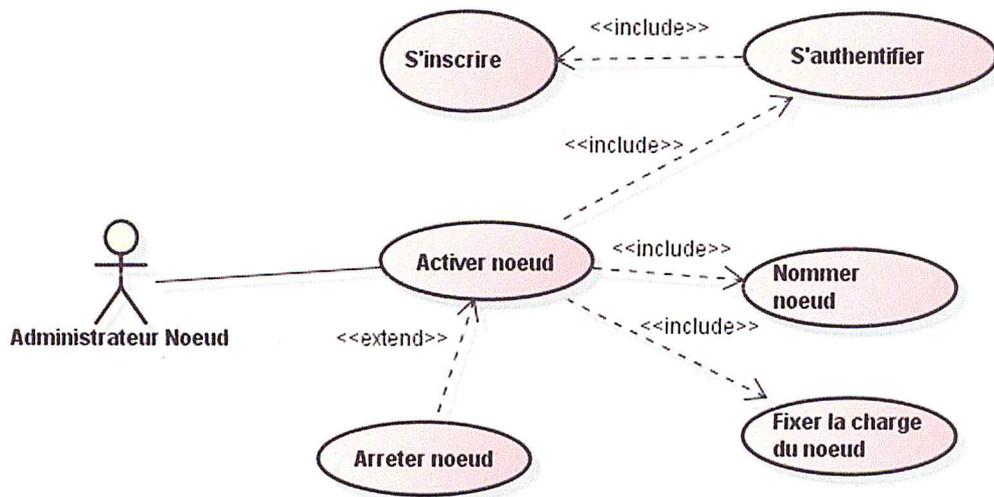
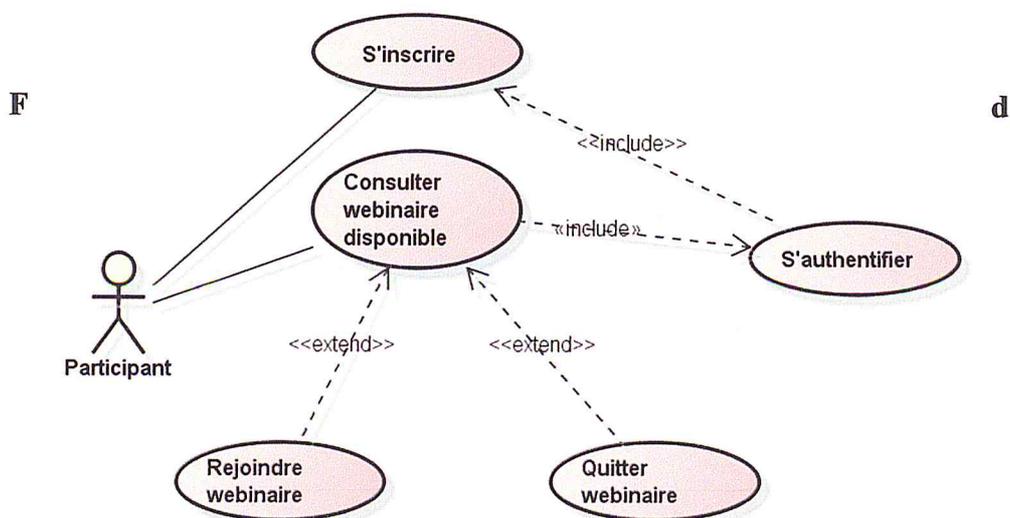


Figure 19 : Diagramme de cas d'utilisation du conférencier



5.2.3 Description textuelle des principaux cas d'utilisation

Dans le but de mieux comprendre notre système et les interactions avec les utilisateurs, dans cette partie nous allons détailler les scenarios de principaux cas d'utilisation



| | |
|----------------------------------|---|
| Cas d'utilisation N° 1 | Inscription |
| Résumé | Permet à l'acteur de s'inscrire. |
| Acteurs | Conférencier, Participant. |
| Précondition | Néant. |
| Scénario nominal | <p>« Début »</p> <ol style="list-style-type: none"> 1. L'acteur lance une demande d'inscription. 2. Un formulaire d'inscription est affiché. 3. L'acteur saisie les informations. 4. Le système vérifie la validité des informations saisies. 5. Le système enregistre ces informations dans la base de données. 6. Le système notifie l'acteur du bon déroulement de l'inscription. <p>« Fin »</p> |
| Scénario alternative | Dans le cas où les informations fournies sont incomplètes ou qui existe déjà un message d'erreur sera affiché. |

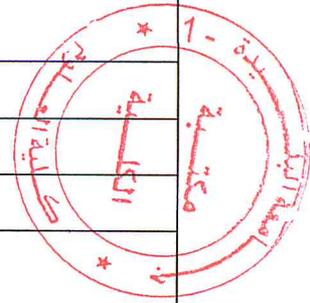


Tableau 2 : Description textuelle du cas d'utilisation inscription

| | |
|----------------------------------|--|
| Cas d'utilisation N° 2 | Authentification |
| Résumé | Permet à l'acteur d'avoir accès au système. |
| Acteurs | Conférencier, Participant |
| Précondition | Avoir déjà un compte |
| Scénario nominal | <p>« Début »</p> <ol style="list-style-type: none"> 1. L'acteur saisie le nom d'utilisateur et le mot de passe. 2. Le système vérifie le nom d'utilisateur et le mot de passe. 3. Le système donne accès à l'interface correspondante. <p>« Fin »</p> |
| Scénario alternative | Dans le cas où les informations sont incomplètes ou incorrectes le système réaffiche le formulaire d'authentification et attend que l'acter ressaisie les informations. |

Tableau 3 : Description textuelle du cas d'utilisation authentification

| | |
|----------------------------------|--|
| Cas d'utilisation N° 3 | Créer un webinaire |
| Résumé | Permet au conférencier de créer son propre webinaire. |
| Acteurs | Conférencier. |
| Précondition | Le conférencier doit être connecté au système (authentification). |
| Scénario nominal | <p>« Début »</p> <ol style="list-style-type: none"> 1. Le système affiche un formulaire qui permet de créer un nouveau webinaire. 2. Le conférencier saisie les données relatives au webinaire. 3. Le système vérifie les informations saisies. 4. Le système valide la création et stocke les informations dans une base de données. 5. Le système notifie le conférencier. <p>« Fin »</p> |
| Scénario alternative | Dans le cas où les informations saisies sont incomplètes un message d'erreur est affiché. |

Tableau 4 : Description textuelle du cas d'utilisation créer webinaire

| | |
|----------------------------------|--|
| Cas d'utilisation N° 4 | Lancer un webinaire. |
| Résumé | Permet au conférencier de lancer un webinaire. |
| Acteurs | Conférencier. |
| Précondition | <ul style="list-style-type: none"> • Le conférencier doit être connecté au système (authentification) • Le webinaire doit être créé. |
| Scénario nominal | <p>« Début »</p> <ol style="list-style-type: none"> 1. Le conférencier choisit le webinaire qu'il souhaite lancer. 2. Le système affiche l'interface spécifique au conférencier. 3. Le conférencier autorise l'accès à son caméra et envoi des notes aux participants. <p>« Fin »</p> |

Tableau 5 : Description textuelle du cas d'utilisation lancer webinaire

| | |
|----------------------------------|---|
| Cas d'utilisation N° 5 | Rejoindre un webinaire |
| Résumé | Permet au participant de rejoindre un webinaire particulier |
| Acteurs | Participant |
| Précondition | <ul style="list-style-type: none"> • Le participant doit être connecté au système (authentification). • Le webinaire doit être lancé. |
| Scénario nominal | <p>« Début »</p> <ol style="list-style-type: none"> 1. Le système affiche les webinaire disponible. 2. Le participant choisit un webinaire. 3. Le système affiche l'interface spécifique au participant. 4. Le participant peut envoyer des questions au conférencier. <p>« Fin »</p> |

Tableau 6 : Description textuelle du cas d'utilisation rejoindre webinaire

| | |
|----------------------------------|--|
| Cas d'utilisation N° 6 | Activer nœud. |
| Résumé | Permet à l'administrateur d'activer le nœud (MCU). |
| Acteurs | Administrateur du nœud. |
| Précondition | L'administrateur du nœud doit être connecté au système du nœud (authentification). |
| Scénario nominal | <p>« Début »</p> <ol style="list-style-type: none"> 1. L'administrateur remplit les informations liées à l'activation du nœud. 2. Le système vérifie les informations saisies. 3. Le système notifie l'administrateur de l'activation du nœud. <p>« Fin »</p> |
| Scénario alternative | Dans le cas où les informations saisies sont incomplètes ou incorrectes un message d'erreur est affiché. |

Tableau 7 : Description textuelle du cas d'utilisation activer nœud

5.3 Diagramme de séquence

Un diagramme de séquences est un diagramme d'interaction qui expose en détail la façon dont les opérations sont effectuées : quels messages sont envoyés et quand ils le sont.

Les diagrammes de séquences sont organisés en fonction du temps qui s'écoule au fur et à mesure que nous parcourons la page.

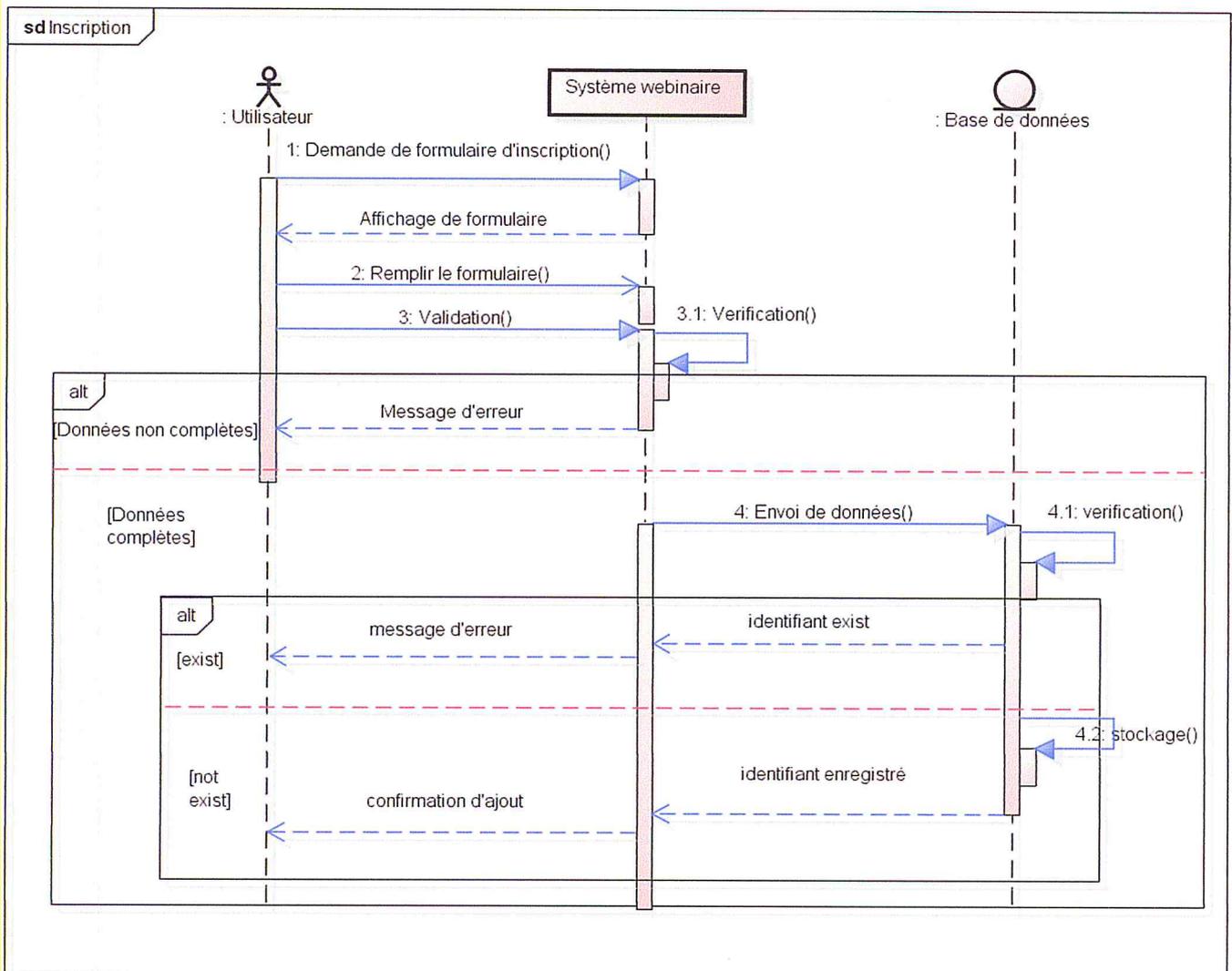


Figure 20 : Diagramme de séquence système pour l'inscription

- Cas d'inscription
- Cas d'authentification

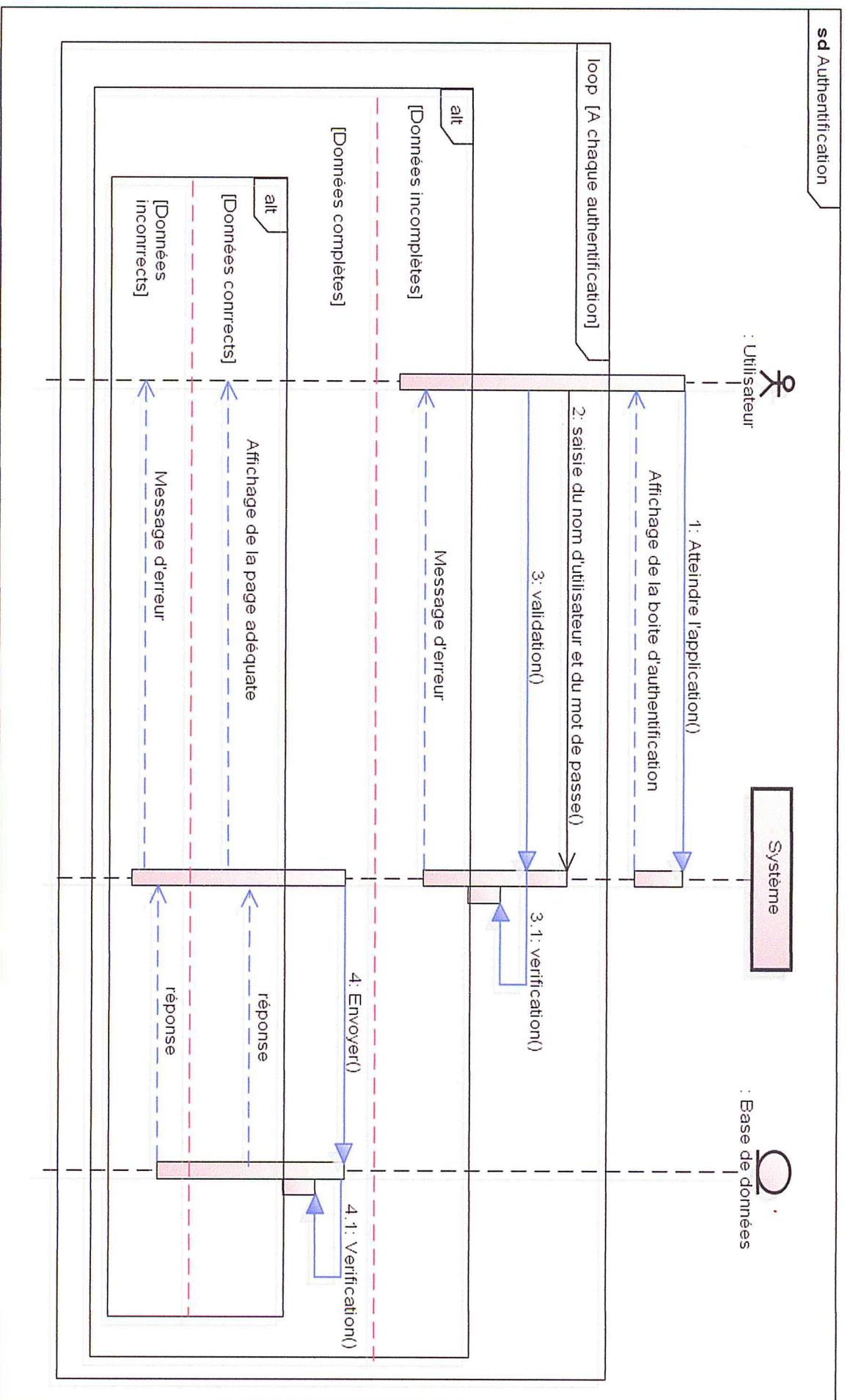


Figure 21 : Diagramme de séquence système pour l'authentification

- Cas créer webinaire

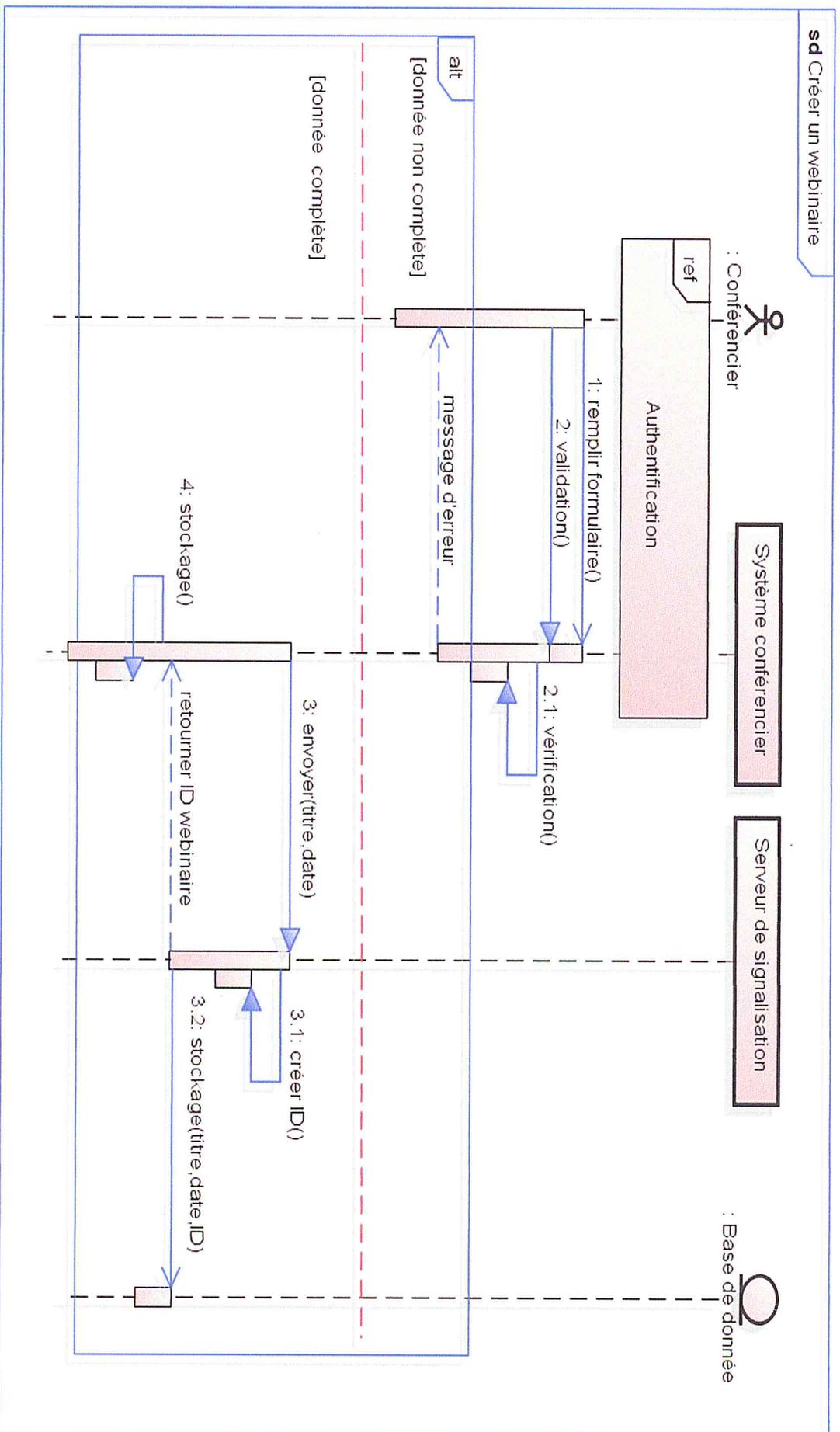


Figure 22 : Diagramme de séquence système pour créer webinaire

• Cas lancer webinaire

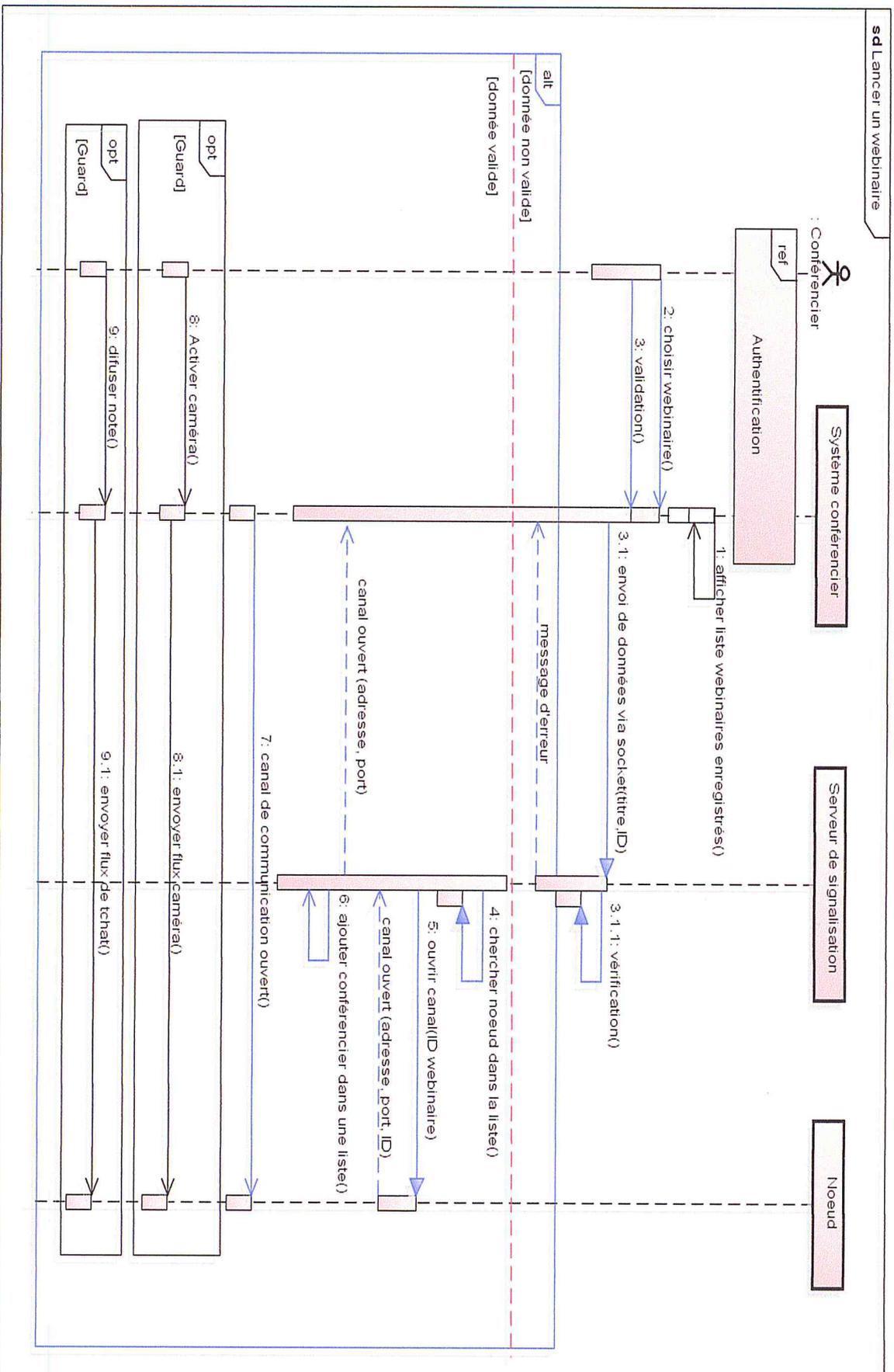


Figure 23 : Diagramme de séquence système pour lancer webinaire

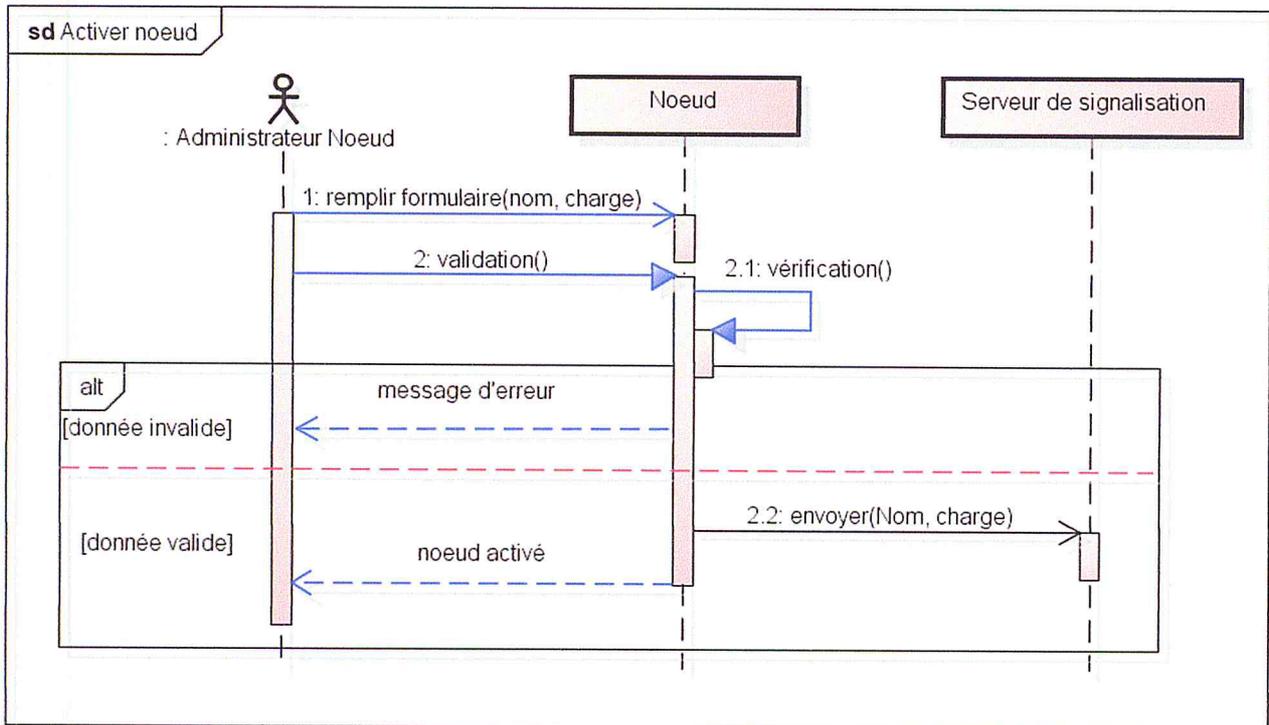


Figure 24 : Diagramme de séquence pour activer noeud

- Cas Activer noeud
- Cas consulter webinaire

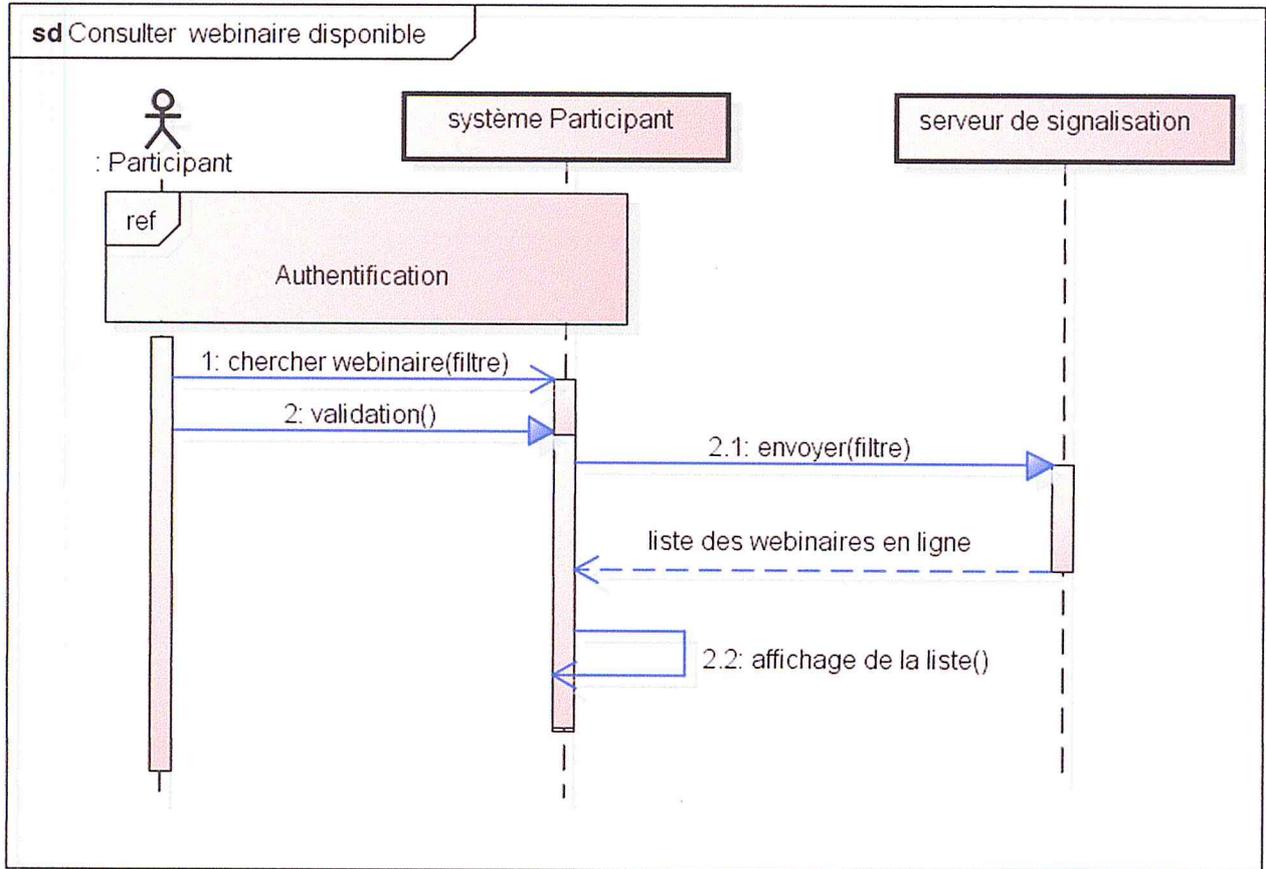


Figure 25 : Diagramme de séquence pour consulter webinaire

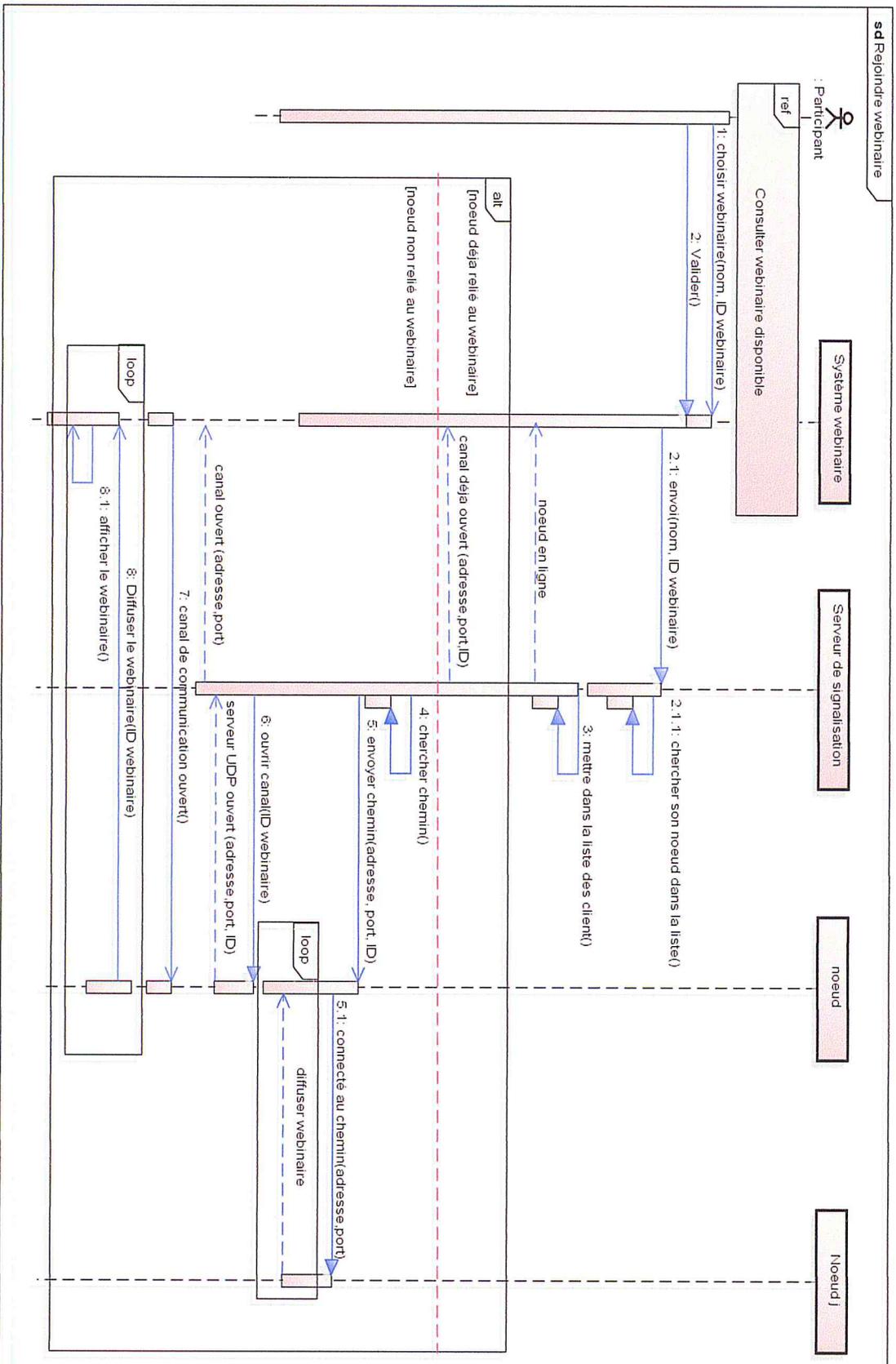
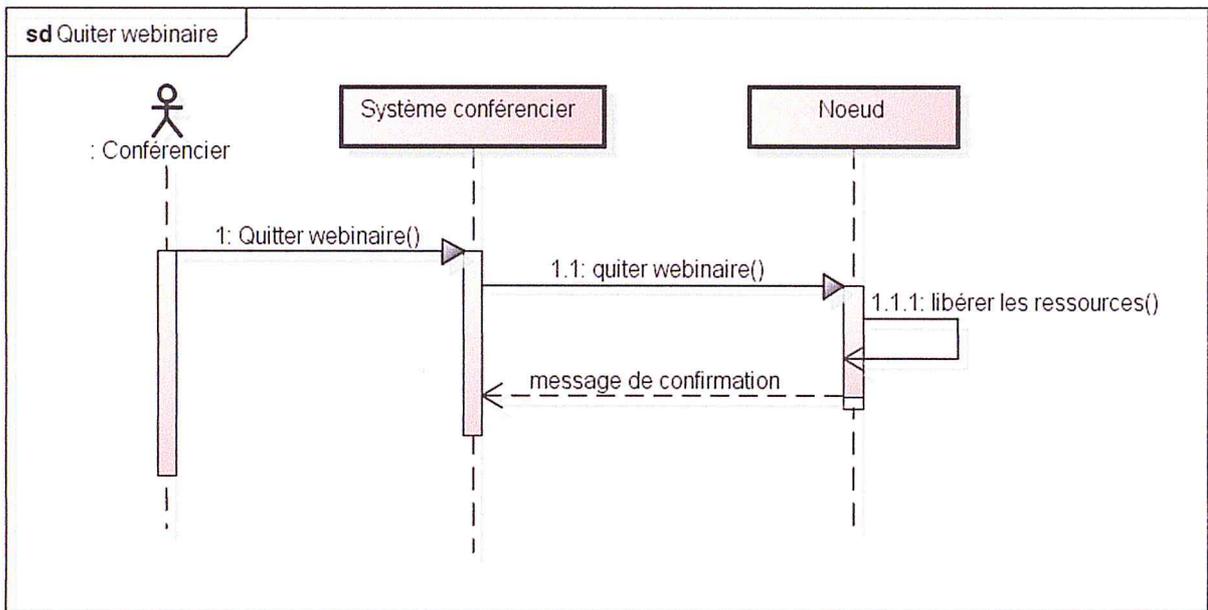


Figure 26 : Diagramme de séquence pour rejoindre webinar

• Cas de rejoindre webinar

- Cas quitter webinaire pour le conférencier



- Cas quitter webinaire pour le participant

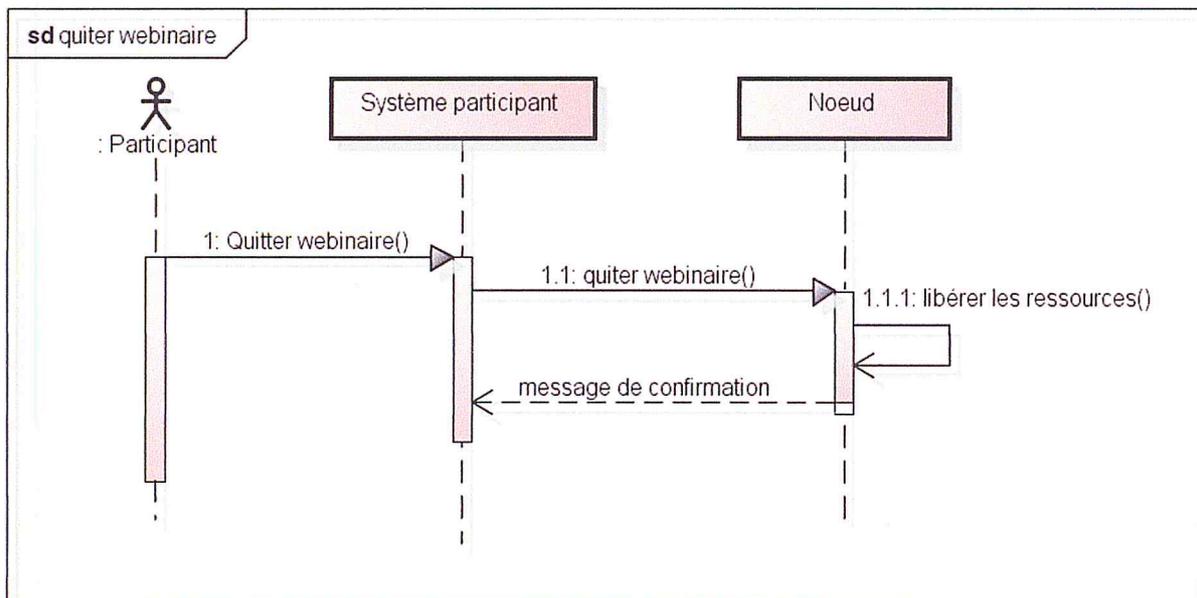


Figure 28 : Diagramme de séquence du cas d'utilisation quitter webinaire (participant)

r)

5.4 Diagramme de classe

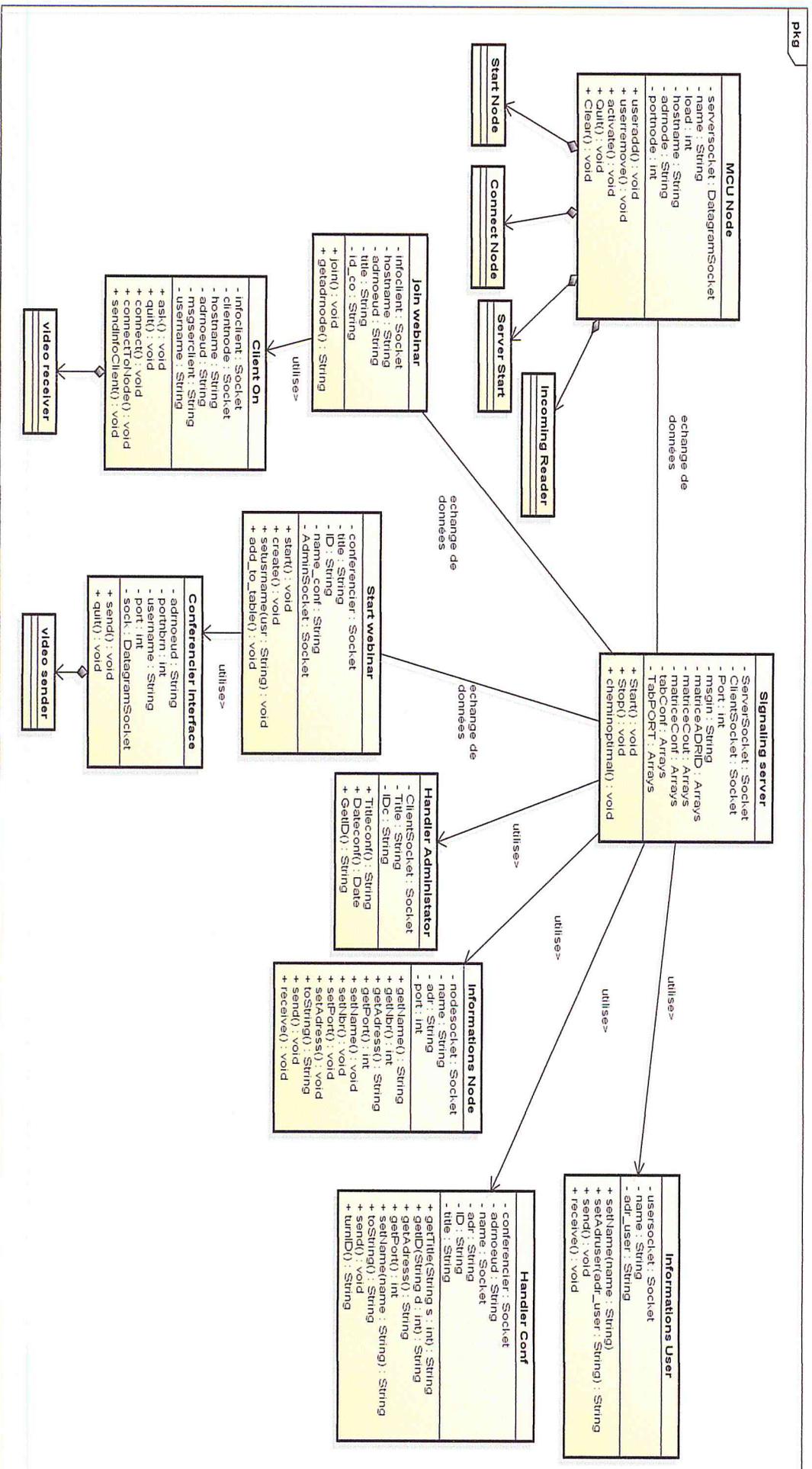


Figure 29 : diagramme de classe

6 Description de la solution

Dans cette section nous décrivons la solution qu'on a adoptée pour l'implémentation de notre système. Dans notre solution nous allons créer les entités suivantes :

6.1 Serveur de signalisation

Le serveur de signalisation représente l'entité principale pour le système de gestion des canaux de communication, ce dernier contient toutes les informations nécessaires pour pouvoir gérer ses clients lors du déroulement des webinaires. Parmi ces informations nous citons :

- Une base de données qui contient les informations d'un webinaire créer par un conférencier titre du webinaire, date de sa diffusion, et l'ID du webinaire générer aléatoirement par le serveur de signalisation.
- Une matrice qui représente le cout qui signifie le temps de transmission entre deux nœud Comme l'illustre la figure 24 ci-dessous.

| | Noeud1 | Noeud2 | Noeud3 | Noeud4 |
|--------|--------|--------|--------|--------|
| Noeud1 | ∞ | | | |
| Noeud2 | 15 | ∞ | | |
| Noeud3 | 76 | 91 | ∞ | |
| Noeud4 | 99 | 37 | 45 | ∞ |

Figure 30 : Matrice des couts

Cette matrice est utilisée pour choisir le chemin optimal pour chaque conférence en choisissant le cout minimal entre deux nœuds, par exemple le nœud 4 a un cout égale à 99 avec le nœud 1, et 37 avec le nœud 2 et 45 avec le nœud 3, donc selon la matrice nous choisissons de le relié avec le nœud 2

- Une matrice de conférence utilisée pour connaître les nœuds qui véhiculent le flux de la conférence, comme le montre la figure 25 ci-dessous :

| | | | | |
|---------------------|--------|--------|--------|--------|
| Conférence 1 | Noeud1 | Noeud2 | Noeud3 | |
| Conférence 2 | Noeud2 | Noeud4 | | |
| Conférence 3 | Noeud3 | | | |
| Conférence n | Noeud2 | Noeud1 | Noeud3 | Noeud4 |

Figure 31 : matrice des conférences

- Une liste des nœuds, liste des conférenciers et liste des participants qui contiennent leurs adresses IP, numéro de port d'entrée et d'autres informations supplémentaires.

6.2 Le Nœud

Le nœud représente le serveur de diffusion d'une université il permet à ses participants (étudiants) d'assister aux conférences qu'ils choisissent parmi ses autres fonctionnalités nous citons :

- il se connecte en premier lieu avec le serveur de signalisation pour indiquer sa présence, et attend la demande du serveur de signalisation pour véhiculer des conférences.
- Il se charge d'envoyer les flux de données qu'il reçoit du conférencier à ses participants.
- Il a la possibilité de limiter le nombre de participants qu'il peut supporter.

6.3 Algorithme du chemin optimal

Cet algorithme est exécuté par le serveur de signalisation

Arriver du Participant (P_i)

P_i Choisi un webinaire avec l'identifiant (IDconf)

Chercher le nœud (N_p) du participant (P_i)

Début

Si charge_Np > 0 et Np véhicule **IDconf**

Alors

charge_Np = charge_Np -1 ;

ouvrir canal (Np, Pi) ;

Sinon Si charge_Np <= 0 alors

Message au Pi « Nœud saturé »

Fin Si ;

Si charge_Np > 0 et Np ne véhicule pas **IDconf**

Alors

Récupérer les Nœud qui véhiculent la conférence **IDconf**

Parcourir la matrice des couts et récupérer l'adresse du nœud avec cout minimal (**Nmin**)

ouvrir canal (Np , Nmin) ;

ouvrir canal (Np , Pi) ;

charge_Np = charge_Np -1 ;

charge_Nmin = charge_Nmin -1 ;

Ajouter Np a la matrice des conférences ;

Sinon

Si charge_Np <= 0

Alors

Message au Pi « Nœud saturé »

Fin Si ;

Fin ;

Mot clé :

Pi : participant (i)

IDconf : l'id de la conférence

Np : Nœud du Participant

Charge_Np : Charge du Nœud de Participant

Nmin : Nœud qui a le cout minimal

Charge_Nmin : Charge du Nœud qui a le cout minimal

7 Conclusion

Ce chapitre était consacré à la conception détaillée de notre application, en utilisant le langage UML ainsi que le processus de développement unifié (Unified Processus) que nous avons bien défini, ces deux derniers nous ont bien aidés à décrire le futur fonctionnement de notre application afin d'en faciliter sa réalisation.

Nous avons représenté l'ensemble de séquence d'actions, en identifiant les cas d'utilisation de tous les acteurs du système. Ensuite, nous avons recensé la description graphique des cas d'utilisation, en réalisant des diagrammes de séquence décrivant les détails des opérations des cas d'utilisation essentiels, et nous avons fait une description de notre solution qui nous permettra d'aboutir au résultat souhaité, que nous allons découvrir dans le chapitre suivant.

Chapitre 4 : Implémentation

1 Introduction

Notre but étant fixé et sa conception élaborée, nous allons passer à la phase de réalisation de notre projet, qui est considérée comme étant la concrétisation finale de toute la méthode de conception.

Nous présentons en premier lieu l'architecture que nous avons choisie pour notre système, ensuite nous décrivons les outils de développements adoptés, soit l'environnement utilisé qui est NetBeans, ainsi que le langage de programmation JAVA, enfin nous présentons quelques interfaces de l'application ainsi un scénario pour illustrer le fonctionnement de quelques activités du système.

2 Outils de développement

2.1 NetBeans¹⁹

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le PHP et le HTML, Il offre toutes les facilités d'un IDE moderne (éditeur en couleurs, projets multi-langage, éditeur graphique d'interfaces et de pages Web).

NetBeans constitue par ailleurs une plateforme qui permet le développement d'applications spécifiques (bibliothèque Swing (Java)).

2.2 Java²⁰

Java est un langage de programmation informatique orienté objet. Java permet de développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, et aux JSP (JavaServer Pages)

¹⁹ <https://fr.wikipedia.org/wiki/NetBeans>

²⁰ <https://www.futura-sciences.com/tech/definitions/internet-java-485/>

2.3 SQL²¹

Le langage SQL (Structured Query Language) peut être considéré comme le langage d'accès normalisé aux bases de données. Il est aujourd'hui supporté par la plupart des produits commerciaux que ce soit par les systèmes de gestion de bases de données micro tel qu'Access ou par les produits plus professionnels tels qu'Oracle. Il a fait l'objet de plusieurs normes ANSI/ISO dont la plus répandue aujourd'hui est la norme SQL2 qui a été définie en 1992.

Le succès du langage SQL est dû essentiellement à sa simplicité et au fait qu'il s'appuie sur le schéma conceptuel pour énoncer des requêtes en laissant le SGBD responsable de la stratégie d'exécution. Le langage SQL propose un langage de requêtes ensembliste. Néanmoins, le langage SQL ne possède pas la puissance d'un langage de programmation : entrées/sorties, instructions conditionnelles, boucles et affectations. Pour certains traitements il est donc nécessaire de coupler le langage SQL avec un langage de programmation plus complet.

2.4 Apache Derby²²

Apache Derby est un moteur de base de données relationnelle écrit en langage Java qui peut être embarqué dans des programmes écrits en Java. Étant multiplateforme et de très petite taille, il s'intègre particulièrement bien dans toute application Java.

Apache Derby est un projet open source sous licence Apache 2.0, derby est aussi connu sous les noms IBM Cloudscape et Sun Java DB.

3 Présentation des interfaces de « UnivWebinar »

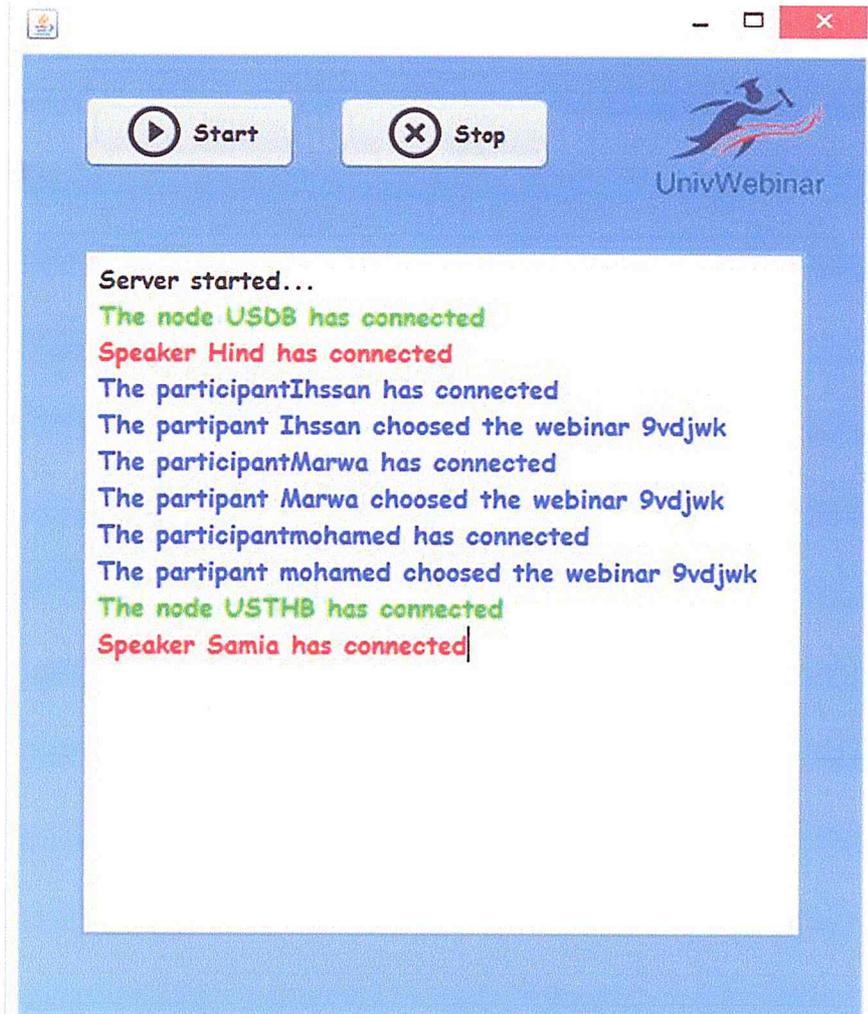
Dans ce qui suit, nous allons présenter les interfaces de notre système.

3.1 Interface du serveur de signalisation

²¹ <https://laurent-audibert.developpez.com/Cours-BD/?page=langage-sql>

²² https://fr.wikipedia.org/wiki/Apache_Derby

Cette interface (figure 35) offre un aperçu sur les différents clients (participants, conférenciers, nœuds) qui se connectent au serveur de signalisation, elle contient un bouton « Start » qui permet de lancer le serveur de signalisation et un bouton « Stop » qui permet de



l'arrêter.

3.2 Interface du nœud

Cette interface (figure 36) montre un aperçu sur les conférences qui passe sur ce nœud et les participants et les conférenciers qui sont reliés à ce nœud.

L'interface du nœud affiche une liste de tous les clients connectés à ce nœud (conférenciers, participants et d'autres nœuds), le bouton « Activate » permet d'activer le nœud, le bouton

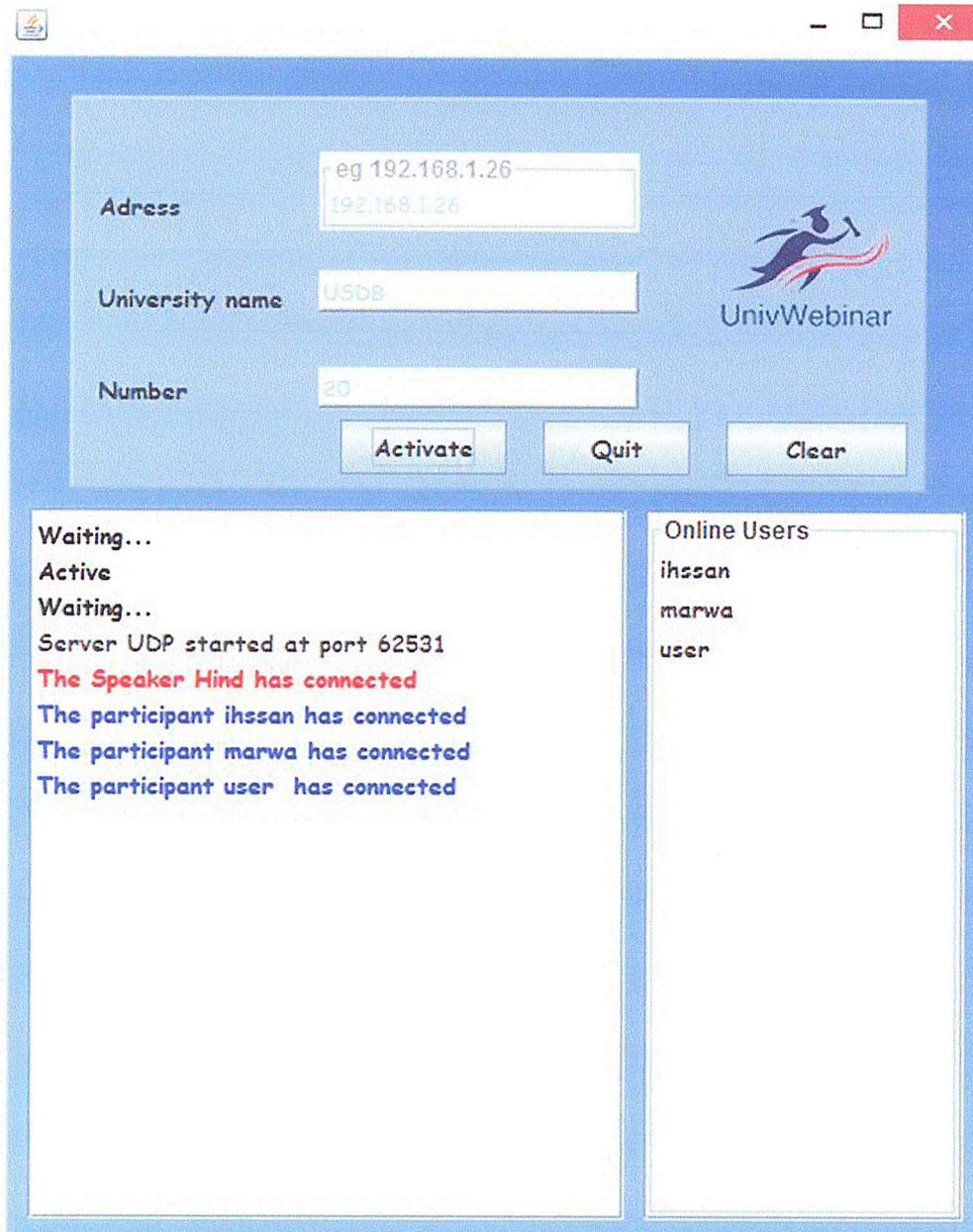
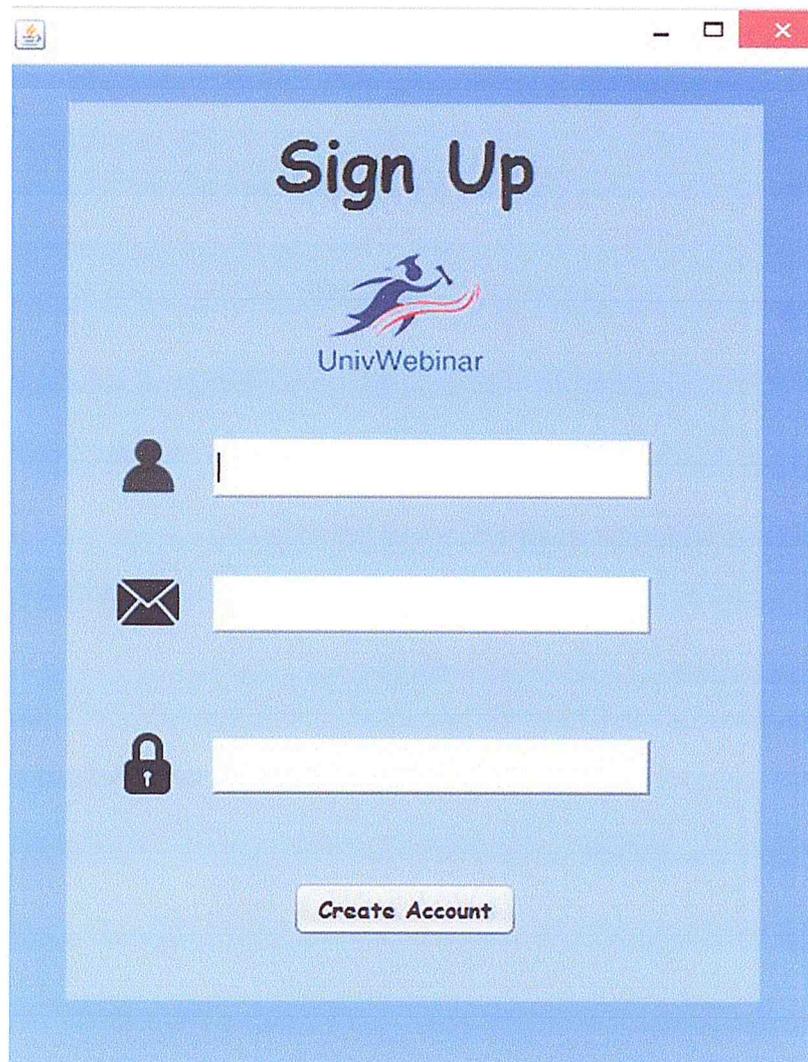


Figure 33 : Interface du nœud

« Quit » permet de le désactiver.

3.3 Interface d'inscription

Cette interface (figure 37) permet la création d'un nouveau compte, l'utilisateur doit saisir un pseudo (nom d'utilisateur), un e-mail et un mot de passe, puis cliquer sur le bouton « Create Account ».



The image shows a web browser window displaying a 'Sign Up' form for 'UnivWebinar'. The form has a light blue background and is centered. At the top, it says 'Sign Up' in a large, bold font, followed by the UnivWebinar logo. Below the logo, there are three input fields, each with an icon to its left: a person icon for the username field, an envelope icon for the email field, and a lock icon for the password field. At the bottom of the form, there is a button labeled 'Create Account'.

Figure 34 : Interface d'inscription

3.4 Interface d'authentification

Cette interface (figure 38) permet l'authentification d'un participant ou d'un conférencier déjà inscrit en insérant son nom d'utilisateur et son mot de passe puis cliquer sur le bouton « login »

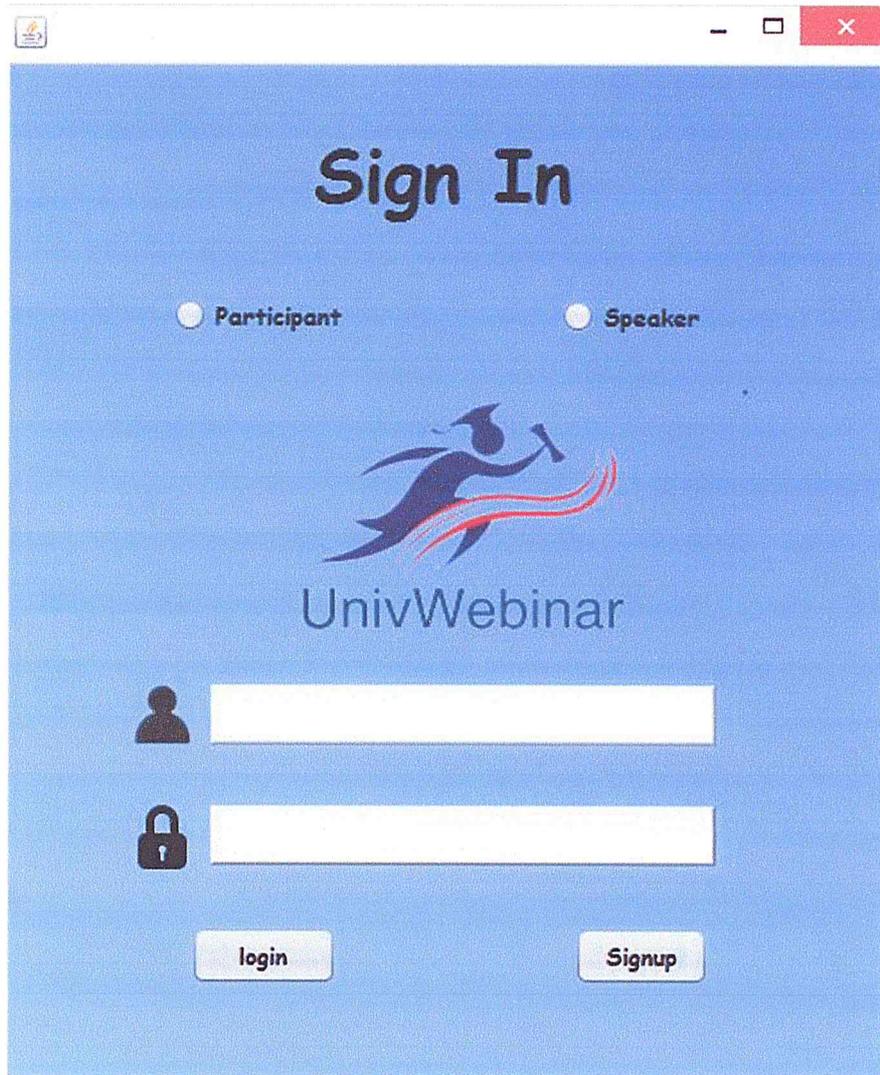


Figure 35 : Interface d'authentification

3.5 Interface du conférencier

L'interface du conférencier est divisée en deux parties une partie pour la création des webinaires et une partie qui permet de lancer un webinaire existant comme le montre la Figure 39 ci-dessous :

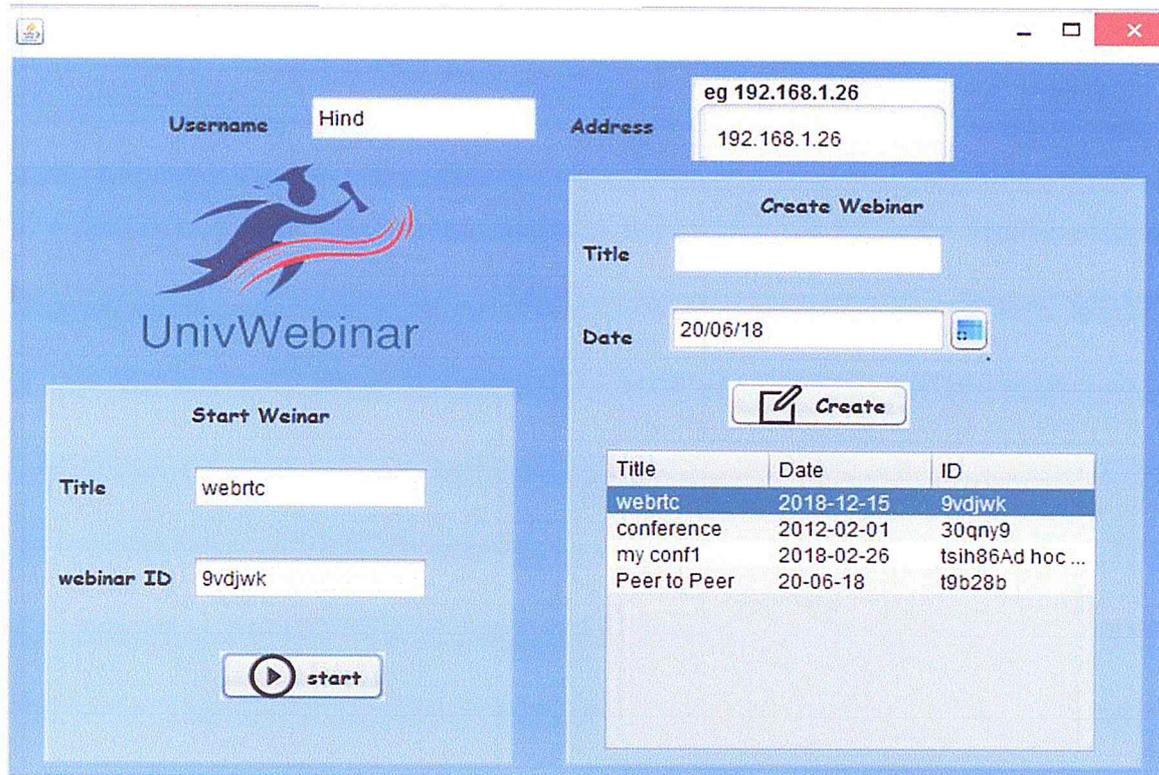


Figure 36 : Interface du conférencier (1)

Quand le conférencier choisit un webinaire et clique sur le bouton « Start » une autre interface s'affiche, il s'agit de l'interface qui lui permet de diffuser la vidéo et envoyer des notes et visionner les participants qui assistent à sa conférence, comme illustré sur la figure 40 ci-

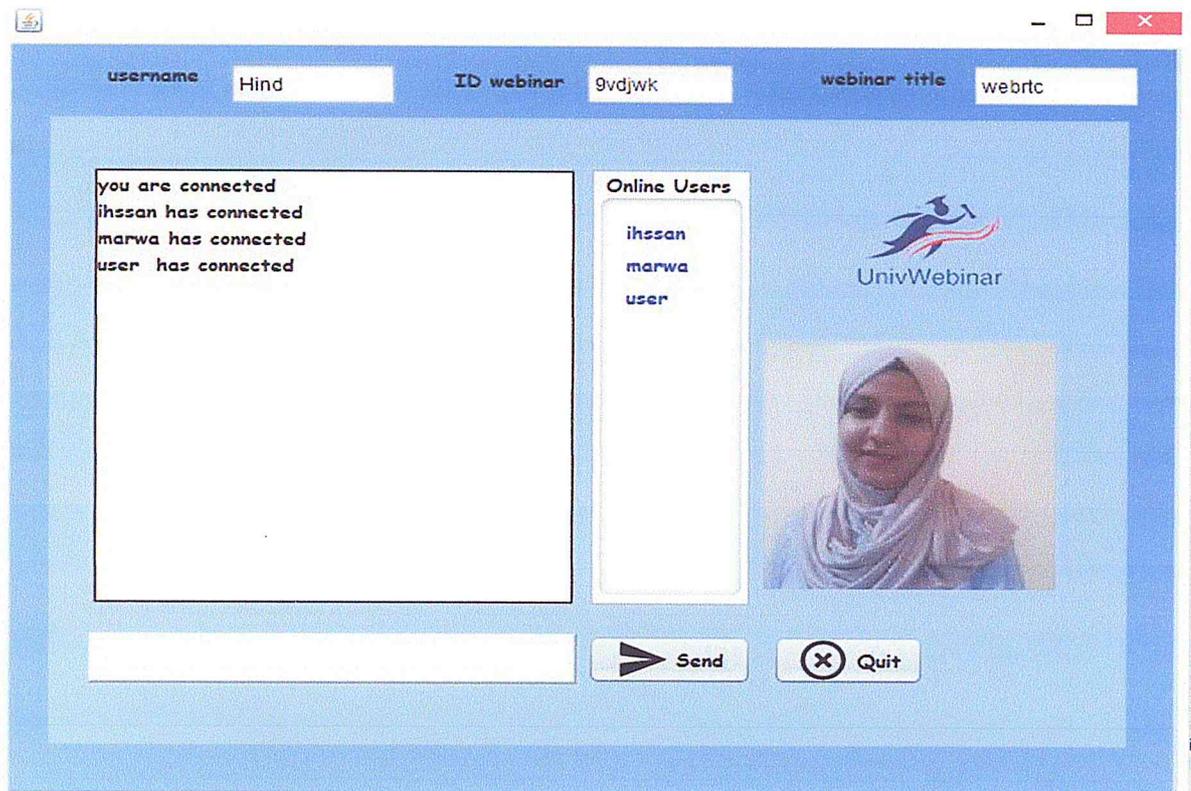
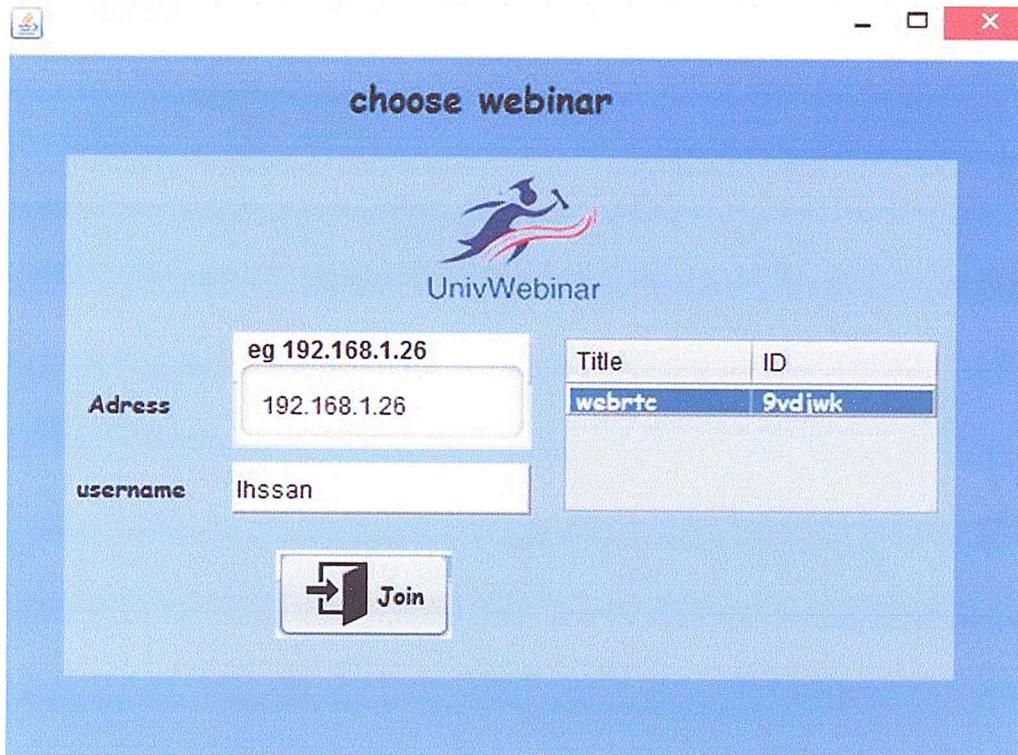


Figure 37 : Interface du conférencier (2)

dessous :

3.6 Interface du participant

La première interface du participant lui permet de consulter la liste des webinaires et assister



un webinaire qui est facultatif comme l'indique la figure 41 ci-dessous :

Si le participant choisit de rejoindre un webinaire particulier une fenêtre sera alors afficher qui permet au participant de visionner la conférence et envoyer des questions au conférencier comme l'indique la figure 42 :

Figure 38 : Interface du participant (1)

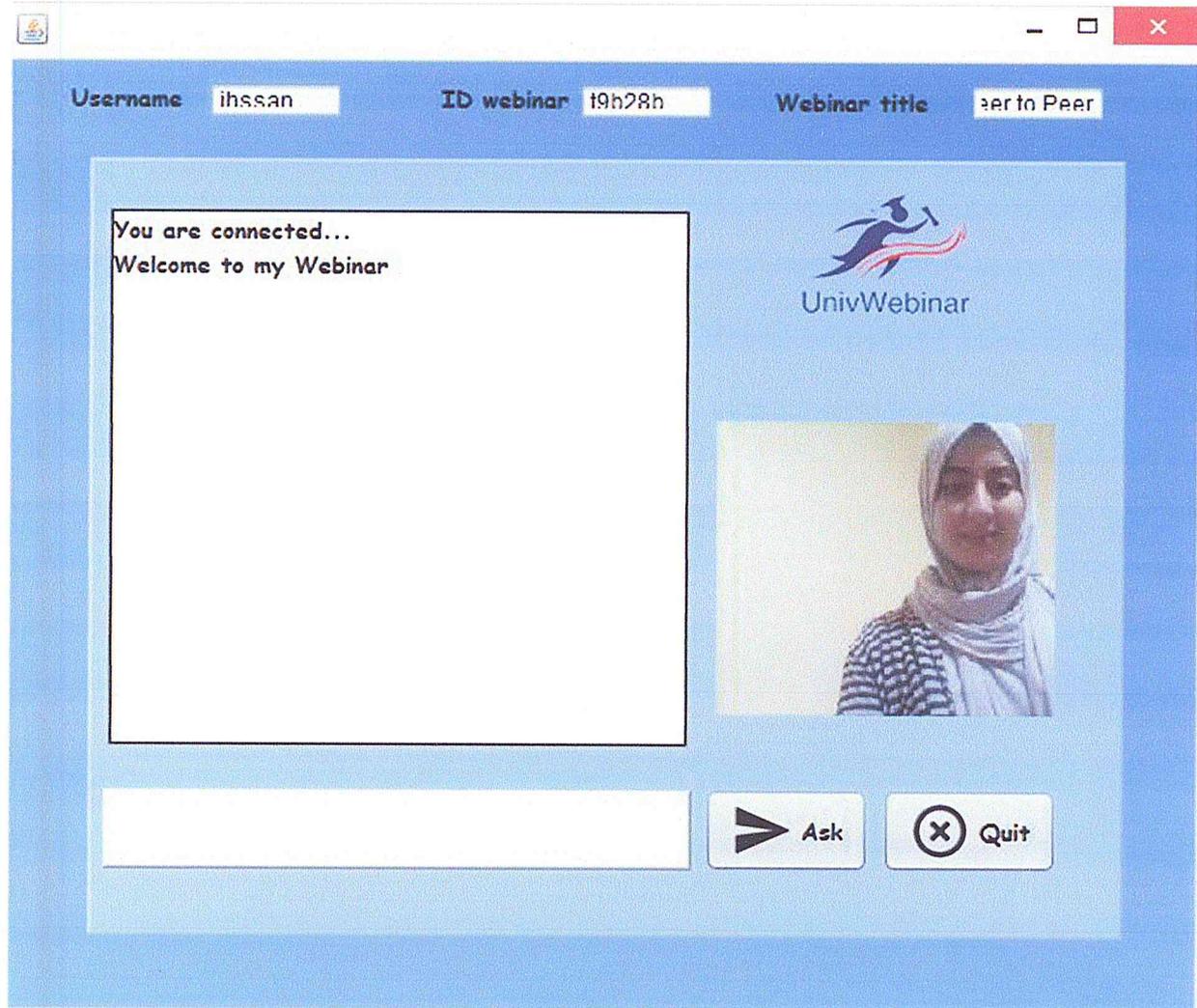


Figure 39 : Interface du participant (2)

4 Scénario et tests

Dans cette partie, nous illustrons les étapes d'un scénario de test qui explique le fonctionnement de notre système « UnivWebinar »

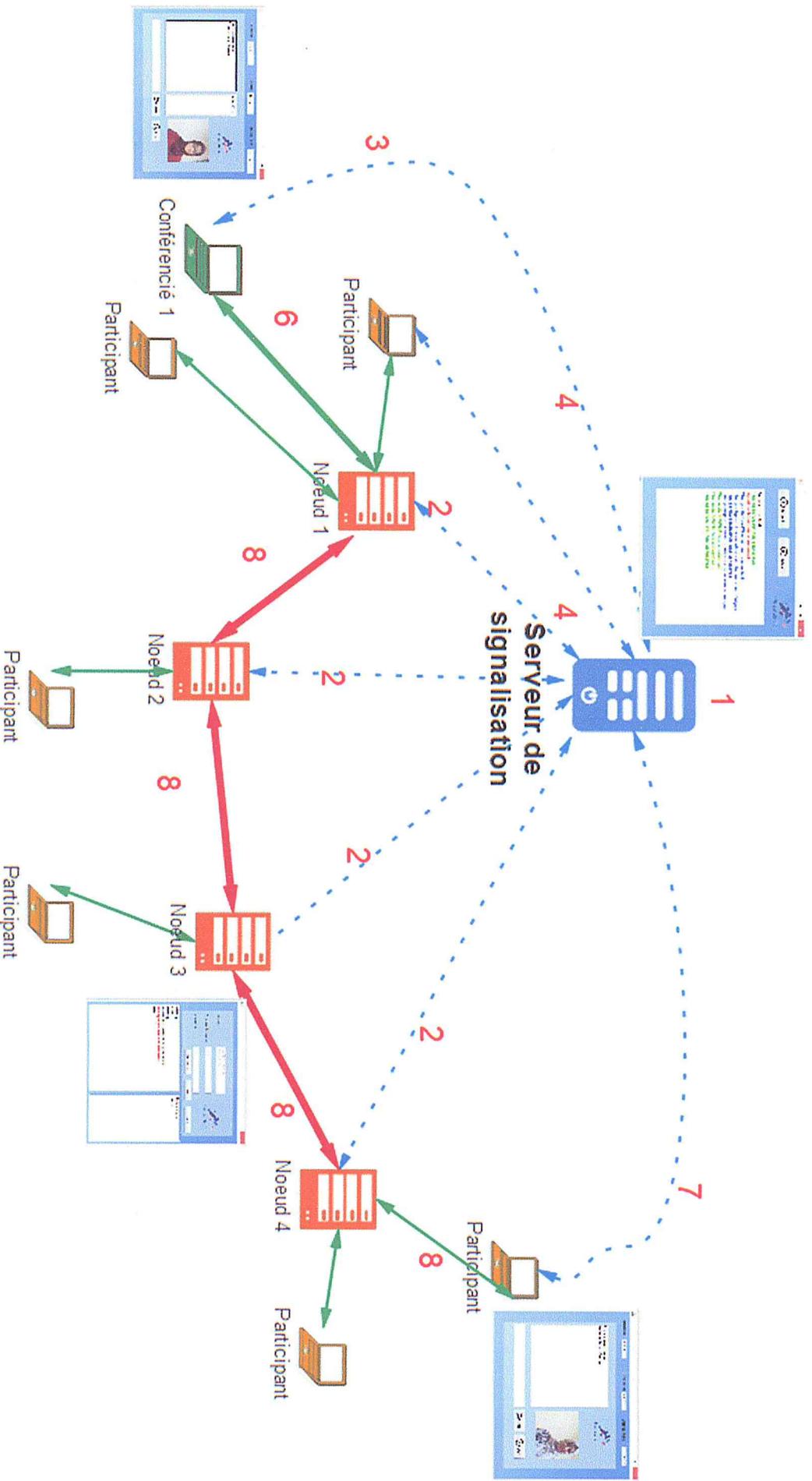


Figure 40 : Scénario de tests du système UnivWebinar

La figure 43 ci-dessus illustre un exemple de scénario de tests du système UnivWebinar dans ce qui suit nous détaillons les étapes de ce scénario :

1. Le serveur de signalisation est activé et attend l'arrivée des clients (nœuds, conférencier et participants).
2. quatre nœuds arrivent après avoir fixé le nom (nom de l'université) et la charge (nombre de participants supporté) de chacun d'entre eux.
3. Un conférencier arrive et choisi un webinaire déjà créer.
4. Le serveur de signalisation vérifie la validité des informations du conférencier, si ces dernière sont valide il demande à son nœud d'ouvrir un serveur UDP par l'ID du webinaire, et récupère le port de ce serveur UDP, et le transmet au conférencier.
5. Le conférencier récupère l'adresse et le port de son nœud (le serveur UDP).
6. Un canal de communication est ouvert entre le conférencier et son nœud, et la diffusion du webinaire commence.
7. Les participants se connectent au serveur de signalisation et récupèrent la liste des webinaires en cours.
8. Après avoir choisi un webinaire, le serveur de signalisation trace un chemin optimal du conférencier vers ces participants. Pour leur permettre de visualiser le webinaire choisit.

5 Conclusion

Nous avons vu dans ce chapitre l'architecture réseau de notre système, l'environnement logiciel sur lequel s'est basé notre travail, ainsi que les langages d'implémentation utilisés.

Nous avons également présenté quelques captures d'écran qui montrent les différentes interfaces graphique du système, et enfin un scénario illustrant le déroulement des différentes fonctionnalités qu'il contient.

Conclusion générale

Dans le cadre de ce travail, l'objectif principal été de développer un système de gestion dynamique des canaux de communications entre les participants et le conférencier dans les webinaires pour assurer la diffusion, le bon déroulement et la continuité des webinaires. La solution que nous avons proposée permet d'acheminer les flux à travers plusieurs nœuds qui représentent les universités d'une manière optimale, pour augmenter la scalabilité et diminuer la charge sur le conférencier.

L'application UnivWebinar que nous avons réalisée adopte une solution inspirée de la partie de signalisation de la technologie web temps réel (WebRTC), cette solution est représenté sous la forme d'un serveur de signalisation central, qui est l'entité principale de notre système son but est de gérer les différentes entités celles du nœud, le conférencier et les participants.

En ce qui concerne la démarche, nous avons en premier lieu effectué une phase d'étude des différentes architectures des outils webinaires existants. En deuxième lieu, nous avons spécifié notre application pour discerner les fonctionnalités. En troisième lieu, nous avons procédé à sa conception ainsi qu'aux choix technologiques pour sa réalisation. Enfin, nous l'avons mise en œuvre.

Bien que notre application UnivWebinar ne soit qu'à sa première version, elle est néanmoins totalement fonctionnelle et offre un certain nombre de fonctionnalités intéressantes telles que :

- Ouvrir des canaux de communication UDP entre le conférencier et les participants en passant par des nœuds.
- La diffusion de la vidéo du conférencier vers tous ses participants
- Trouver le chemin optimal entre le conférencier et chaque participant.

Dans la prochaine version de notre application, nous envisageons de rajouter les fonctionnalités suivantes :

- Système d'information des webinaires pour une bonne gestion des webinaires enregistrées.
- Gestion des interactions entre le conférencier et les participants, c'est-à-dire si le participant demande de poser une question le serveur contrôle sa demande, il l'accepte ou non selon des critères.

- Notifier les participants des nouveaux webinaires ajoutés et les webinaires démarrés.
- Adapter l'application au web.
- Partage de fichier par le conférencier à ses participants.
- Enregistrement des webinaires déjà présentés pour qu'ils soient accessibles plus tard.

Enfin, ce travail nous a permis non seulement de nous initier sur les concepts de base de réseau et d'avoir une idée sur le fonctionnement des systèmes webinaires et leurs intérêts ainsi que les différentes architectures réseaux adoptés par ces systèmes, mais aussi à approfondir nos connaissances dans le domaine de la programmation réseau.

Glossaire

API: Application Programming Interface

CPU: Central Processing Unit

DB: Data Base

IBM: International Business Machines Corporation

IETF: Internet Engineering Task Force

IP: Internet Protocol

IPBX: Internet Protocol Private Branch eXchange

RTCP: Real-time Transport Control Protocol

RTP: Real-time Transport Protocol

SDP: Session Description Protocol

SIP: Session Initiation Protocol

TCP: Transmission Control Protocol

ToIP: Telephony Over IP

UDP: User Datagram Protocol

UML: Unified Modeling Language

VoIP: Voice Over IP

W3C: World Wide Web Consortium

Bibliographie

- [1] J.-N. Anderruthy, Skype et la téléphonie IP: téléphonez gratuitement avec Internet, ENI Edition, 2006.
- [2] S. Androutsellis-Theotokis et D. Spin, «A survey of peer-to-peer content distribution technologies,» *ACM Computing Surveys*, vol. 36, n° 14, p. 336, decembre 2004.
- [3] F. Schuler, «developpez.com,» 7 04 2005. [En ligne]. Available: <https://schuler.developpez.com/articles/p2p/>. [Accès le 06 03 2018].
- [4] D. Caron et D. Poure, «L'empire du Peer to peer,» Université paris, Paris, 2007.
- [5] N. BUDAN, B. TEDESCHI et S. VAUBOURG, «Nouvelles Technologies Réseau, Les réseaux peer-to-pee,» 12 02 2003. [En ligne]. Available: <http://igm.univ-mlv.fr/~duris/NTREZO/20022003/Peer-to-peer.pdf>. [Accès le 06 03 2018].
- [6] G. Pujolle, Les réseaux, 2ème édition éd., Eyrolles, 1998.
- [7] A. S. Tanenbaum, Computer Networks, 3 ème Edition éd., New York: Prentice Hall, 1996.
- [8] B. Douskalis, IP telephony: the integration of robust VoIP services, USA: Prentice Hall , 2000.
- [9] A. B. Johnston et D. C. Burnett, WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web, USA: Digital Codex LLC, 2012.
- [10] V. Jacobson et M. Handley, «SDP: Session Description Protocol,» *RFC 2327, IETF*, p. 06, April 1998.
- [11] J. Rosenberg, H. Schulzrinne et et al, «SIP : Protocole d'initialisation de session,» *RFC 3261*, p. 4, 2002.
- [12] A. Rakesh, «Voice over IP : Protocols and Standards,» *Washington University*, p. 10, 1999.
- [13] L. Greenemeier, «New Video VoIP Software, Webcams End That Pixelated Feeling,» *Scientific American*, 31 10 2007.
- [14] C. Montculier, «www.01net.com,» 03 05 2001. [En ligne]. Available: <https://www.01net.com/actualites/lorganisation-dune-web-conference-150339.html>. [Accès le 17 01 2018].
- [15] A. Verma et A. Singh, «Leveraging Webinar for Student Learning,» chez *International Workshop on Technology for Education*, Bangalore, India, 2009.
- [16] H. Wang, «Skype VoIP service- architecture and comparison,» chez *INFOTECH Seminar Advanced Communication Services*, Allemagne, 2005.
- [17] Y. Xu, C. Yu, J. Li et Y. Liu, Video Telephony for End-Consumers: Measurement Study of Google+, iChat, and Skype, 4 éd., vol. 22, New York: IEEE, 2014.
- [18] P. Roques, UML 2: modéliser une application web, Les Cahiers du programmeur, paris: Eyrolles, 2008.
- [19] J. Steffe, Cours UML, Bordeaux: ENITA de Bordeaux, 2005.

[20] P. Roques, UML2 par la pratique, 5ème édition éd., Paris: EYROLLES, 2006.

[21] P. Roques, les cahiers du programmeur UML 2, 4ème édition éd., Paris: Groupe Eyrolles, 2008.

