# **Introduction**

## **Contexte**

Les temps changent; l’enseignement change; les universités changent. Dans le cadre de la mondialisation, on assiste à l'emploi du numérique, avec des méthodes de plus en plus créatives et collaboratives, pour une meilleure éducation. Cela reste valable dans le milieu de l’enseignement supérieur. Il va sans dire que des facteurs comme la pandémie de COVID-19 ou encore, l'indisponibilité de cadres de cours adéquats, rendent plus urgent le besoin de transitionner vers des salles de classe virtuelle, pour répondre aux besoins. De ce fait, les technologies de l’information et de la communication constituent un atout décisif dans le succès de cette transition.

## **Problématique**

L’expansion de l’emploi des cours en ligne est désormais un fait. Cela requiert une organisation logistique accrue et un investissement financier pour les entités universitaires. Toutefois, l’on note l’emploi de solutions génériques, qui rendent difficile, voire impossible l'émulation d’un environnement de classe. Pour peu qu’elles soient conformes aux exigences, c’est alors le prix qui peut poser problème. C’est la raison d'être de notre projet, qui vise la mise en place d’une application, pour répondre au besoin d'interactivité lors des cours en ligne, et éliminer les barrières d’ordre logistique et financier, imposées par les solutions génériques.

## **Objectifs**

Le principal objectif est la conception de **StudX**, un prototype d’application SaaS, permettant la tenue de cours en ligne. En termes de fonctionnalités et buts, il s’agira notamment de pouvoir:

* organiser les différentes classes, filières ou promotions des entités en sections bien définies
* définir le calendrier des cours à tenir
* organiser des sections d’audio-conférence pour le déroulement des cours
* mettre en place des fonctionnalités telles le partage d'écran et bien d’autre pour émuler un tant soit peu, un environnement de classe présentiel
* minimiser les coûts requis dans le cadre de la mise en oeuvre d’une solution de classe virtuelle

## **Organisation du mémoire**

Le présent document renferme trois chapitres. Dans le premier chapitre, nous ferons une revue de littérature sur le sujet et présenterons les généralités sur quelques notions essentielles. Le second chapitre relate les méthodes employées pour la conception de notre solution ainsi que les outils et matériels utilisés à cette fin. Le dernier chapitre sera consacré à la présentation des résultats obtenus, des interfaces conçues ainsi que des potentielles insuffisances liées à la solution que nous avons développée.

# 

# 

# 

# 

# **Revue de littérature**

## **Introduction**

Le concept de la formation à distance ne date pas d’hier. Dans ce chapitre, nous ferons une revue des origines de cette méthode d’enseignement. S’en suivra une analyse des techniques modernes de communication en temps réel et des solutions existantes qui permettent de dispenser des cours à distance.

**1.1. Formation à distance**

L'encyclopédie Wikipedia définit la formation à distance comme une forme d’enseignement ou l’enseignant et l'étudiant sont séparés dans le temps et par l’espace [1].

**1.1.1 Origines**

Les premiers essais de formation à distance remontent à bien avant l'ère moderne. En effet, déjà en 1728, Caleb Phillips, un professeur, recherchaient des étudiants désirant acquérir des compétences en sténographie, auxquels il dispensait les cours par courrier.

Au sens moderne, le premier cours d’enseignement à distance, est attribué à Isaac Pitman, toujours en rapport avec la sténographie. Un nouvel élément qui apparaît dans le cas actuel, c’est la rétroaction des étudiants, qui devaient envoyer leurs transcriptions par la poste, pour correction. Ce mode de fonctionnement fut rendu possible par l’uniformisation des tarifs postaux en Angleterre. Plusieurs institutions telles que Oxford et l'université de Londre ont également expérimenté l’enseignement à distance.

Aujourd’hui, l’expansion d’Internet et du World Wide Web, permettent la mise en œuvre des moyens toujours plus sophistiqués, pour dispenser les cours à distance.

**1.1.2 Internet et formations en ligne**

L'avènement des nouvelles technologies de l’information et de la communication a donné lieu à la mise en place des formations en ligne, une forme évoluée de formation à distance [2]. En 2020, par exemple, sous l’impulsion de la pandémie alors en cours, plusieurs universités dont celle d’Abomey-Calavi [url uac], effectuent la transition partielle ou totale vers les classes virtuelles [3].

On distingue deux environnements d’apprentissage. D’une part, les environnements asynchrones [4] offrent une totale liberté à l'apprenant quant à la gestion de son temps. L’enseignant et lui sont séparés littéralement par le temps et la distance. Ainsi, il peut consulter les ressources au moment qui lui convient le mieux. Cela permet une assimilation plus facile, étant donné que chaque apprenant peut s’adapter en fonction de ses besoins spécifiques. Toutefois, il est possible que l’apprenant se retrouve isolé et ne fasse finalement aucun progrès, faute de support.

D’autre part, un environnement synchrone essaie d'émuler une classe présentiel, à la seule différence que les participants sont physiquement distants. Avec des outils de messagerie instantanée et/ou de visioconférence, les apprenants peuvent interagir avec leurs pairs ainsi que le ou les enseignants.

En termes de classification des diverses formes de cours en ligne, Andreas Kaplan[], professeur de marketing, propose une approche simplifiée basée sur les facteurs comme le temps, la distance et le nombre d’apprenants. Le tableau ci-dessous, en fait un récapitulatif.

| Type | Description | Nombre d’apprenants | Type d’environnement |
| --- | --- | --- | --- |
| MOOCs | Massive Online Open Courses | illimité (en théorie) | asynchrone |
| SMOCs |  | illimité (en théorie) | synchrone |
| SPOCs |  | limité | asynchrone |
| SSOCs |  | limité | synchrone |

Fig. Tableau récapitulatif des types de cours en ligne

A ce stade, il est important de préciser que notre projet s'intéresse aux environnement d’apprentissage synchrones, avec le support d’un grand nombre d’apprenants.

**1.2. Communication en temps réel**

Les outils de communications en temps réel désignent une catégorie de logiciels qui garantissent le traitement et la transmission instantanée, ou avec un délai fortement négligeable, de l’information. Parmi les protocoles permettant ce type de communication, le plus en vogue reste WebRTC.

WebRTC est un protocole open source de transmission P2P, qui assure la transmission de média (audio, vidéo) et de données brutes, presque sans latence (moins d’une seconde), le tout dans un contexte hautement sécurisé. Il s’agit en réalité, d’une collection de protocoles datant des années 2000 [ref webrtcforthecurious]. Pour établir une connexion, il faut quatre étapes à savoir la signalisation, la connexion proprement dite, la sécurisation puis la communication.

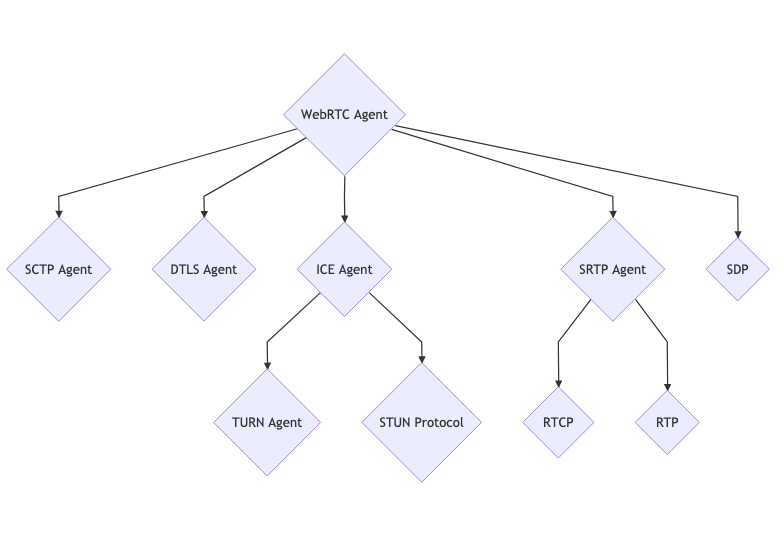


Fig. Protocoles employés par WebRTC

La signalisation désigne le processus initial de mise en relation des pairs. Sans ce processus, une machine quelconque n’a aucune idée de qui voudrait bien la contacter. Pour ce faire, le protocole SDP est utilisé et permet la transmission d’informations capitales comme:

* l’IP et le port de chaque agent WebRTC (plusieurs variantes, en réalité)
* les codecs multimédia supporters
* d’autres valeurs comme des certificats de sécurité nécessaires à la mise en place de la connexion et la sécurisation.

A la connexion, les agents WebRTC établissent un lien direct entre eux, sans intermédiaire. Face à la multitude de possibilités de connexion (couples constitués de l’IP et du port), le protocole ICE permet de choisir le meilleur candidat, en faisant recours au serveur STUN et parfois, au serveur TURN. Le serveur TURN permet la retransmission des données lorsqu’il est impossible pour un agent WebRTC d'établir un lien direct avec un autre agent en raison de la configuration réseau (NAT et les types de liaisons possibles) [webrtc for the curious types of link].

Pour assurer la sécurité de la connexion, les protocoles DTLS et SRTP offrent une couche de chiffrement pour les contenus multimédia et les paquets brutes.

Enfin, les agents peuvent s'échanger de la donnée, du contenu multimédia, presque sans latence, grâce au protocoles RTP et SCTP.

WebRTC est une technologie complexe qui requiert une certaine expertise quant à la connaissance des protocoles, leur utilisation et la mise en œuvre d'applications en temps réel. Elle sert de base aujourd’hui, la plupart des applications de communication en temps réel.

**1.3. Software as a Service**

Parmi les modèles de distribution de logiciels, le SaaS représente une méthode ou le concepteur ou l'entité tenant l’application, l'héberge en ligne et la rend accessible à ses utilisateurs. En terme de commercialisation, il est possible d’offrir un accès à la plateforme moyennant un abonnement ou l’achat d’une version privée pour les besoins des corporations.

**1.3. Présentation de solutions existantes**

Plusieurs solutions s’inscrivent déjà dans le cadre du déroulement de cours en ligne en temps réel.

Avant de relever les solutions existantes, précisons les critères qui en ont motivé le choix. Premièrement, nous nous intéressons aux solutions permettant à des classes d’avoir lieu via les conférences en ligne. Les plateformes choisies doivent donc couvrir ledit cas d’utilisation. Ensuite, elles doivent offrir un certain niveau d’interaction entre les étudiants et leurs enseignants. Pour finir, ces plateformes doivent être accessibles sans recourir à une mise en place de système informatique par l'entité qui s’en sert. Voyons à présent quelques-unes de ces plateformes.

**1.3.1. Présentation de Moodle**

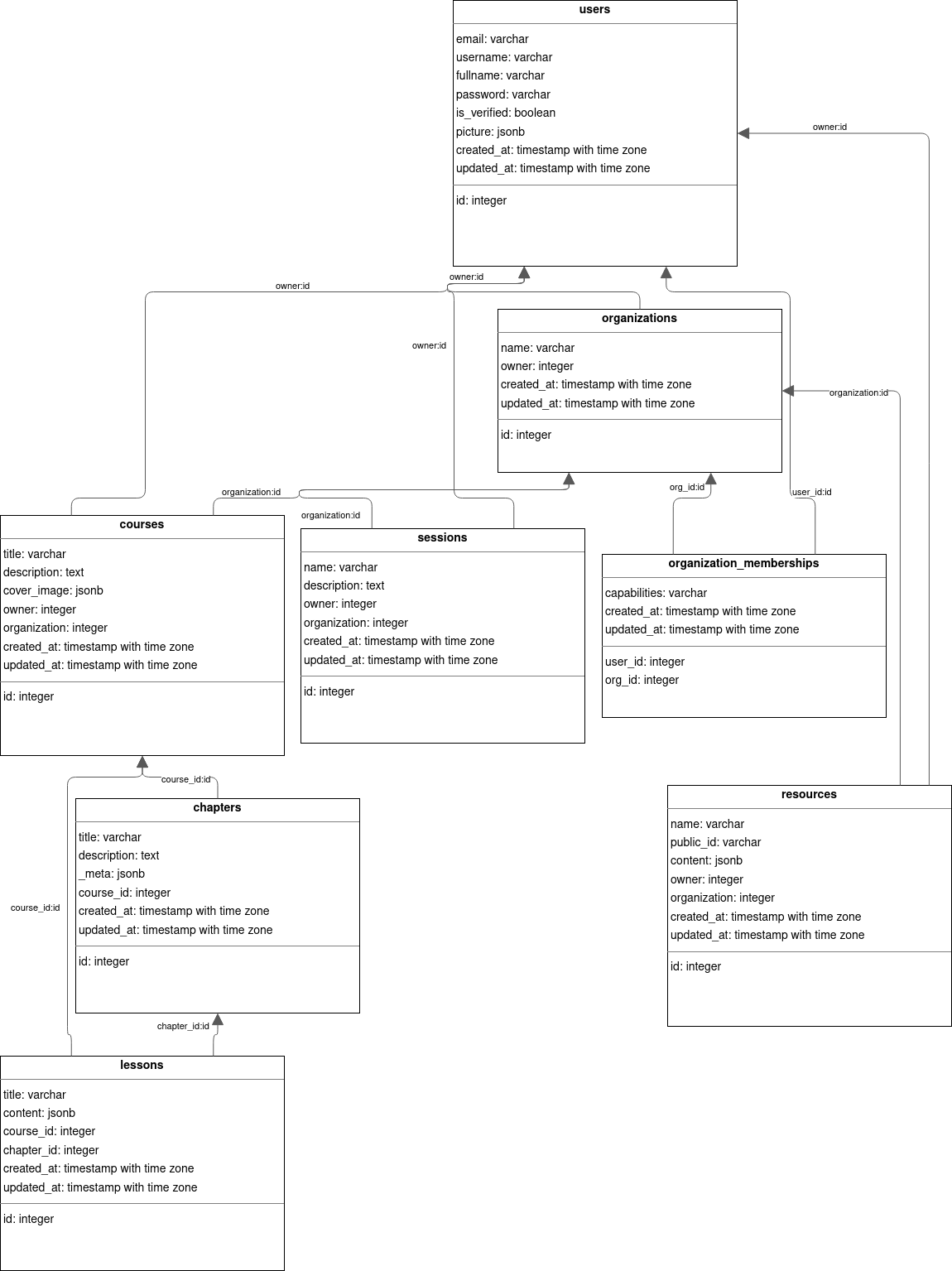
**1.3.2. Présentation de Google Classroom**

# 

# **Analyse et Modélisation (draft)**

Cas d’utilisation

Diagrammes (BD,classe)



# **Matériels et méthodes**

# **Matériels**

* **Rust**

Rust est un langage de programmation annoncé par Mozilla en 2010, compilé, multi-paradigme qui se veut fiable, concurrent et pratique >>. Il offre des performances proches de celles du C/C++. Il permet d’éviter les erreurs de segmentation ou de concurrence déjà à l'étape de compilation; tout ceci sans **Garbage Collector**, ce qui lui confère une excellente vitesse d’exécution. Il supporte les styles de programmation fonctionnelle, procédurale, modèle d’acteur et sous certains angles, orienté objet. C’est un langage fiable pour la conception d’applications web, les outils en ligne de commande (CLI/TUI), la programmation système, réseau ou embarquée. C’est la raison pour laquelle nous l’avons choisi pour le développement du backend de notre application web.

* **Actix**

Actix est un framework reposant sur le modèle d’acteurs et écrit en Rust. Il offre un framework web (**actix-web**) qui permet d’implémenter des services web reposant sur le modèle d’acteurs. C’est un framework toutes batteries incluses qui supporte nativement bien de fonctionnalités comme les *Websockets,* le *TLS* ou encore *HTTP/2.* Il offre d’excellentes performances, raison pour laquelle nous l’avons choisi.

* **Typescript**

Typescript est un langage libre et open-source développé par Microsoft. Il est trans compilé en JavaScript et peut donc être exécuté dans n’importe quel navigateur web ou moteur JavaScript. Il offre un typage statique optionnel des variables et des fonctions; la création et l’import de module. Le but principal de ce langage est de combler les lacunes de JavaScript et de permettre une expérience de développement plus agréable avec une détection des erreurs à l'étape de développement plutôt qu’en production.

* **Nuxt**

Nuxt est un méta framework basé sur le framework frontend Vue.js qui facilite la conception d’applications web universelles (rendues côté serveur SSR ou SPA). Il fournit une abstraction facilitant le développement d’applications Vue.js permettant aux développeurs de se consacrer à la logique métier.

* **WebRTC**

WebRTC (Web Real-Time Communication) est un projet open-source offrant une API de communication audio et vidéo aux navigateurs web et aux applications mobiles. Il offre une connexion P2P éliminant le besoin de plugins ou d’applications spécifiques pour la communication entre utilisateurs. Il est supporté par tous les navigateurs majeurs et ses spécifications sont maintenues par le W3C consortium et l’IETF (Internet Engineering Task Force).

Compilation des ressources pour la webographie et la bibliographie: <https://tobihans.notion.site/Refs-51187c42f1bb4e8b9b0b33cac975794d>