

**<WALL-E2>**

**Document d'architecture logicielle**

**Version 2.0**

WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11

## Historique des révisions

Date	Version	Description	Auteur
2017-01-28	1.0	Échange entre les membres de l'équipe sur les détails de l'architecture logicielle de notre système.	Équipe 9
2017-02-02	1.1	Réalisation des vues suivantes: cas d'utilisation, logique, processus et déploiement.	Samia Jalil et Winnie Tchouandem
2017-02-07	1.2	Rédaction des parties : objectifs et contraintes, taille et performance.	Morgane Renard
2017-02-08	1.3	Ajout de l'introduction, la vue de déploiement et modification des différents diagrammes (cas d'utilisation, paquetage, classe, séquence)	Samia Jalil et Winnie Tchouandem
2017-02-09	2.0	Dernières modifications et corrections, version finale pour la première livraison.	Samia Jalil et Winnie Tchouandem
2017-04-11	3.0	Dernières modifications et corrections, version finale	Samir El Kaoukabi, Winnie Tchouandem, Morgane Renard, et Samia Jalil

WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11

# Table des matières

1.	Introduction	4
2.	Objectifs et contraintes architecturaux	4
2.1	Objectifs	4
2.1.1	Client lourd	4
2.1.2	Serveur	4
2.1.3	Client léger	4
2.1.4	Autres	5
2.2	Contraintes	5
2.2.1	Sécurité et confidentialité	5
2.2.2	Portabilité	5
2.2.3	Échéanciers	5
2.2.4	Outils et langage de développement	6
2.2.5	Réutilisation et maintenance	6
3.	Vue des cas d'utilisation	6
4.	Vue logique	8
4.1	Client léger	9
4.2	Serveur	12
4.3	Client lourd	12
5.	Vue des processus	14
5.1	Création du profil utilisateur	14
5.2	Connexion au client lourd ou client léger	15
5.3	Connexion à une carte privée ou publique	16
5.4	Tutoriel interactif	17
5.5	Site Web	18
5.6	Clavardage	19
5.7	Edition/Simulation	20
5.8	Gestion des comptes	21
6.	Vue de déploiement	22
7.	Taille et performance	23

WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11

# Document d'architecture logicielle

## 1. Introduction

L'architecture logicielle permet de décrire l'organisation générale d'un système et sa décomposition en sous-systèmes. Elle consiste à déterminer les interfaces et les interactions avec le flot de contrôle entre les sous-systèmes. Aussi, elle décrit les composants utilisés pour implémenter les fonctionnalités des sous-systèmes. Dans ce document, nous présenterons la structure détaillée sur laquelle sera basé notre système. Dans un premier temps nous présenterons les objectifs et les contraintes architecturaux. Ensuite, nous présenterons les différentes vues dont celle des cas d'utilisation, logique, des processus, et de déploiement. On terminera avec une description des caractéristiques de taille et de performance qui ont un impact sur l'architecture et le design du logiciel.

## 2. Objectifs et contraintes architecturaux

Les objectifs et contraintes liées à l'architecture logicielle sont spécifiés par le client et explicités dans les documents informatifs fournis par ce dernier.

### 2.1 Objectifs

#### 2.1.1 Client lourd

Au niveau du client lourd, l'équipe de développement se concentre sur quatre principaux objectifs. Le premier est que l'application liée au client lourd doit être une extension du projet intégrateur INF2990 donné en hiver 2016. Ce qui signifie qu'il faut récupérer toutes les fonctionnalités et commandes de l'ancien projet et les transférer sur le nouveau. L'utilisateur doit pouvoir effectuer les mêmes actions qu'auparavant. Le second est d'ajouter la gestion des profils. Tout utilisateur voulant faire une édition ou une simulation en ligne doit se créer un profil et se connecter à ce dernier. Toutes les informations liées au profil utilisateur doivent être privées et protégées. Le troisième est d'améliorer l'ancienne version du projet d'INF2990 en ajoutant la possibilité d'effectuer toutes les actions simultanément en réseau et en temps réel. L'édition et la simulation en ligne sont possibles pour un maximum de quatre personnes connectées au même serveur. Cela signifie que l'utilisateur doit pouvoir éditer ou simuler une carte en ligne au même moment que trois autres personnes. Et enfin le quatrième objectif est que l'application doit être accessible à partir d'ordinateurs portables ou fixes plus particulièrement à partir de ceux mis à la disposition des équipes.

#### 2.1.2 Serveur

Au niveau du Serveur les objectifs de l'équipe sont surtout liés à la communication client-serveur. Les échanges entre le client lourd ou le client léger et le serveur doivent être possibles en tout temps. Les communications entre le client lourd et le serveur doivent utiliser le réseau filaire de l'école Polytechnique Montréal. Tandis que celles entre le client léger et le serveur doivent utiliser le réseau sans fil sécurisé « eduroam ». Le but est de permettre un clavardage accessible à tout moment (pendant l'édition et la simulation) et permettant l'échange de messages entre utilisateurs.

#### 2.1.3 Client léger

Au niveau du client léger, l'équipe du projet travaille sur trois principaux objectifs. Les fonctionnalités présentes dans l'éditeur du projet intégrateur INF2990 d'hiver 2016 doivent être récupérées et remaniées afin de mieux correspondre au nouvel environnement matériel et logiciel. Le but est de profiter au maximum des possibilités et avantages offerts par la présence d'un écran tactile. De plus l'édition sur le client léger doit, au

WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11

même titre que celle du client lourd, s'effectuer en ligne et de façon simultanée. Et enfin l'application mobile doit être installée sur iPad. Évidemment elle doit être parfaitement fonctionnelle sur ce nouvel équipement fourni par le client.

#### 2.1.4 *Autres*

L'équipe de développement vise également d'autres objectifs en lien avec l'ensemble des technologies du projet. L'interface doit absolument être conviviale et harmonieuse. L'expérience utilisateur doit être agréable. L'application doit être fluide et cohérente. Si plusieurs personnes sont en train soit d'éditer ou de simuler en ligne une même carte, elles doivent avoir exactement la même carte devant les yeux et doivent toutes percevoir la moindre modification en temps réel. De plus les informations des utilisateurs doivent être protégées, quelle que soit la plateforme ou l'application utilisée. L'application doit demeurer sécuritaire et confidentielle.

## 2.2 **Contraintes**

### 2.2.1 *Sécurité et confidentialité*

Chaque utilisateur a la possibilité de créer un profil et de se connecter à son profil dès qu'il le souhaite. Ce profil doit être protégé par un identifiant et un mot de passe tous deux fournis par l'utilisateur lors de la création de son profil. Les informations personnelles contenues et visibles dans un profil sont consultables uniquement par le propriétaire du profil en question lorsqu'il s'est identifié. Pour utiliser l'application en ligne (édition et simulation en ligne) l'utilisateur doit obligatoirement s'authentifier sinon son rôle doit se limiter à celui d'un simple spectateur. L'application doit alors absolument prévoir une gestion des profils utilisateurs.

### 2.2.2 *Portabilité*

Ce projet est la continuité du projet intégrateur INF2990 proposé en hiver 2016. Les fonctionnalités implémentées auparavant doivent rester identiques sauf dans le cas de l'éditeur sur iPad où elles doivent être adaptées au nouvel environnement. La différence avec l'ancienne version c'est que l'édition et la simulation doivent pouvoir s'effectuer en ligne et de façon simultanée sur le client lourd. Il en va de même sur le client léger sauf que, dans ce cas, seul le mode édition est concerné. Il faut que l'application soit fonctionnelle à la fois sur le client lourd par le biais d'un ordinateur quelconque de la salle de laboratoire, à la fois sur le client léger par le biais d'un iPad. Le projet intégrateur INF2990 doit donc être amélioré et doit être facilement transplantable sous de nouvelles technologies.

### 2.2.3 *Échéanciers*

Le projet est divisé en deux livrables : la réponse à l'appel d'offres avec la remise des prototypes de communication et des artefacts et enfin le produit final qui inclut les artefacts complétés et mis à jour. Chaque livrable comprend ses différents lots de travail et chaque lot de travail est évalué en terme de durée et d'effort à fournir par l'équipe de projet. Cette évaluation est laissée à la discrétion du groupe toutefois les dates de tombée de chaque livrable sont imposées par le client. Le premier livrable est à remettre le 10 février 2017 avant 23h59. Le second doit être remis le 11 avril 2017 avant 23h59.

WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11

#### 2.2.4 Outils et langage de développement

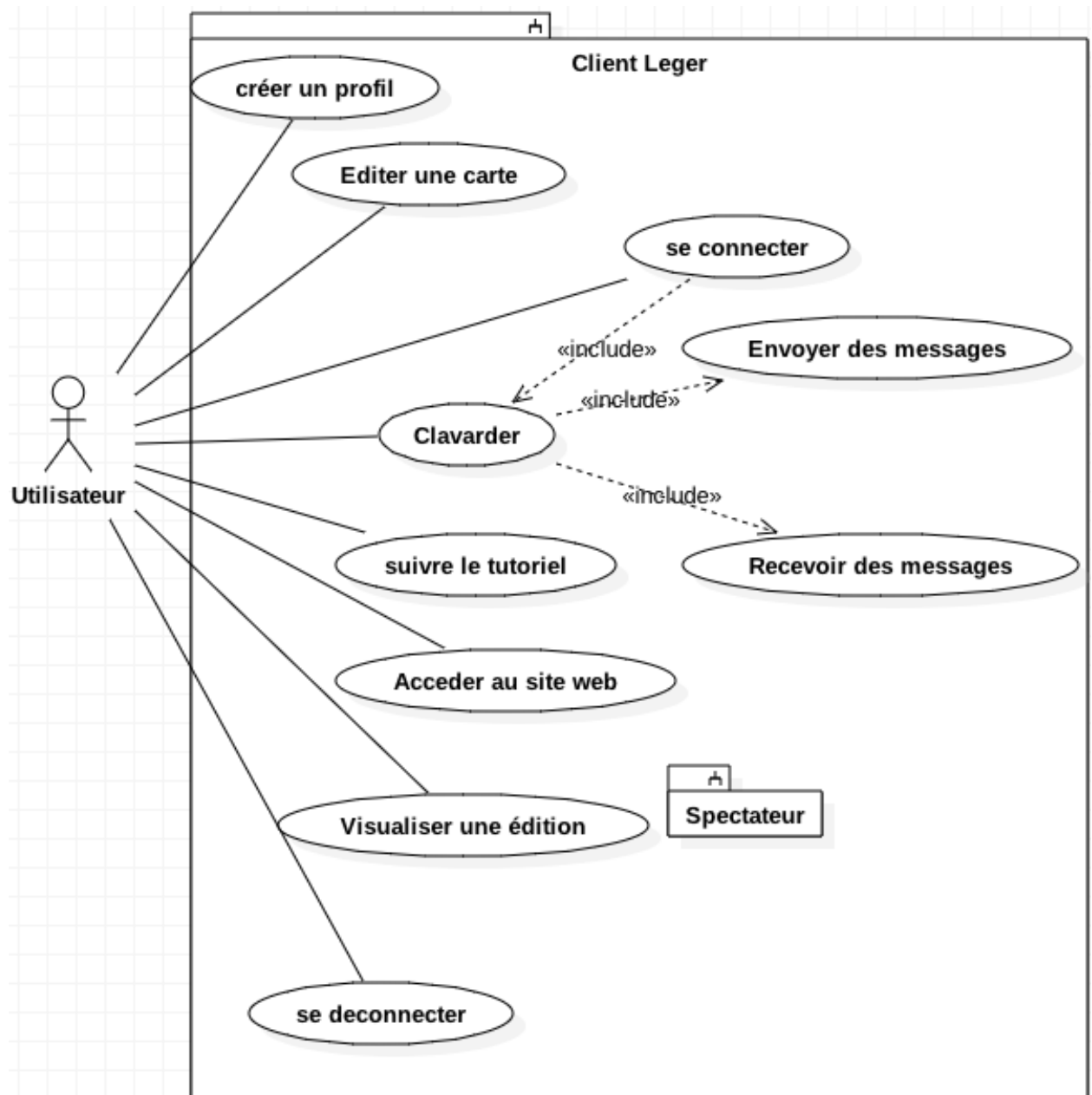
L'environnement de développement utilisé pour développer le serveur et le client lourd est Visual Studio 2015. Le code source est implémenté en C++ et toutes les interfaces sont écrites en C#. L'application développée sur le client lourd doit être fonctionnelle sur n'importe quel ordinateur de la salle de laboratoire. En ce qui concerne le client léger, l'environnement de développement employé est Xcode et l'application est implémentée en Swift. Cette dernière doit être disponible sur iPad.

#### 2.2.5 Réutilisation et maintenance

Comme précisé auparavant (2.2.1 Portabilité) ce projet est un prolongement et une amélioration du projet intégrateur INF2990 d'hiver 2016. Il est donc possible que ce projet soit de nouveau réutilisé, enrichi, voire corrigé par la suite. Par conséquent, l'équipe de développement doit s'attacher à rédiger des documents et une architecture logicielle clairs, explicites et à jour, à rendre du code bien structuré et bien organisé et surtout à générer et fournir une documentation complète et limpide.

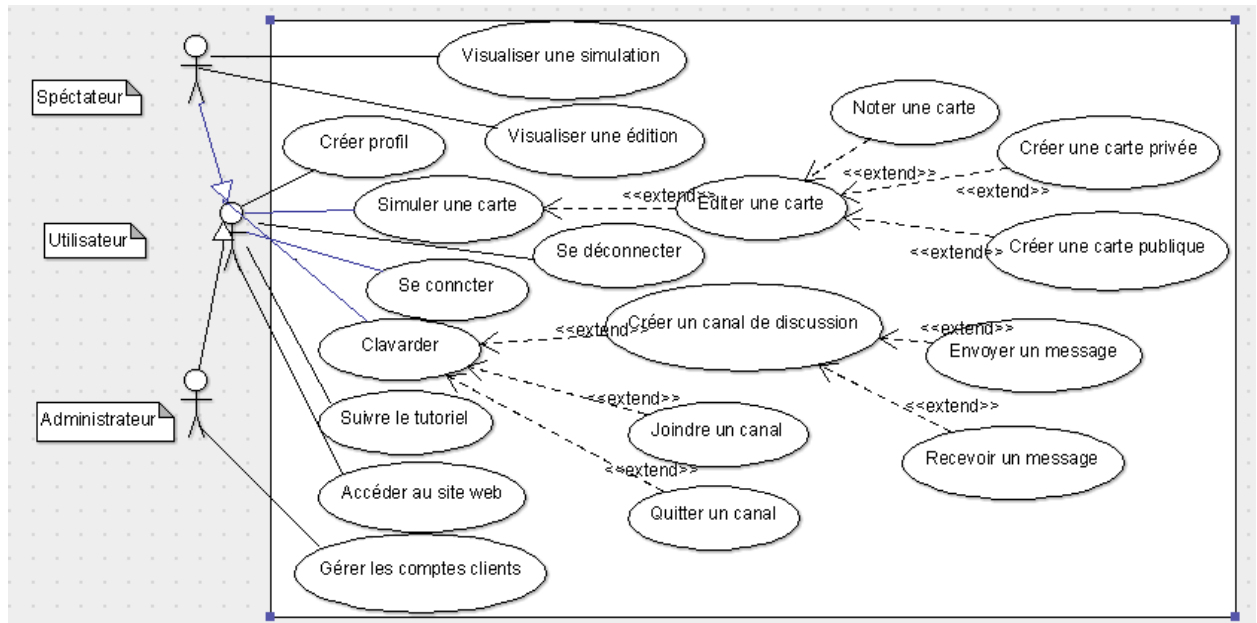
### 3. Vue des cas d'utilisation

Dans cette section, nous présenterons dans les deux figures ci-dessous, les différents cas d'utilisation possibles du simulateur de robot pour le client lourd et le client léger.



*Figure1 : Diagramme de cas d'utilisation client léger*

WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11



*Figure 2 : Diagramme de cas d'utilisation client lourd*

#### 4. Vue logique

Cette section présente une vue globale du design de notre système. Nous avons d'une part le diagramme de paquetage, et d'une autre part les diagrammes de classe des paquets impliqués. Il est à noter que nous ne reviendrons pas en détail sur certains paquets vus dans le projet d'INF2990 de deuxième année comme FacadeInterfaceNative et Noyau.

À noter que le paquetage commun est comme son nom l'indique commun entre le serveur et le client lourd.

Il contient une seule classe 'Card'. Cette classe sera impliquée dans les diagrammes des classes des paquets qui l'utilisent.



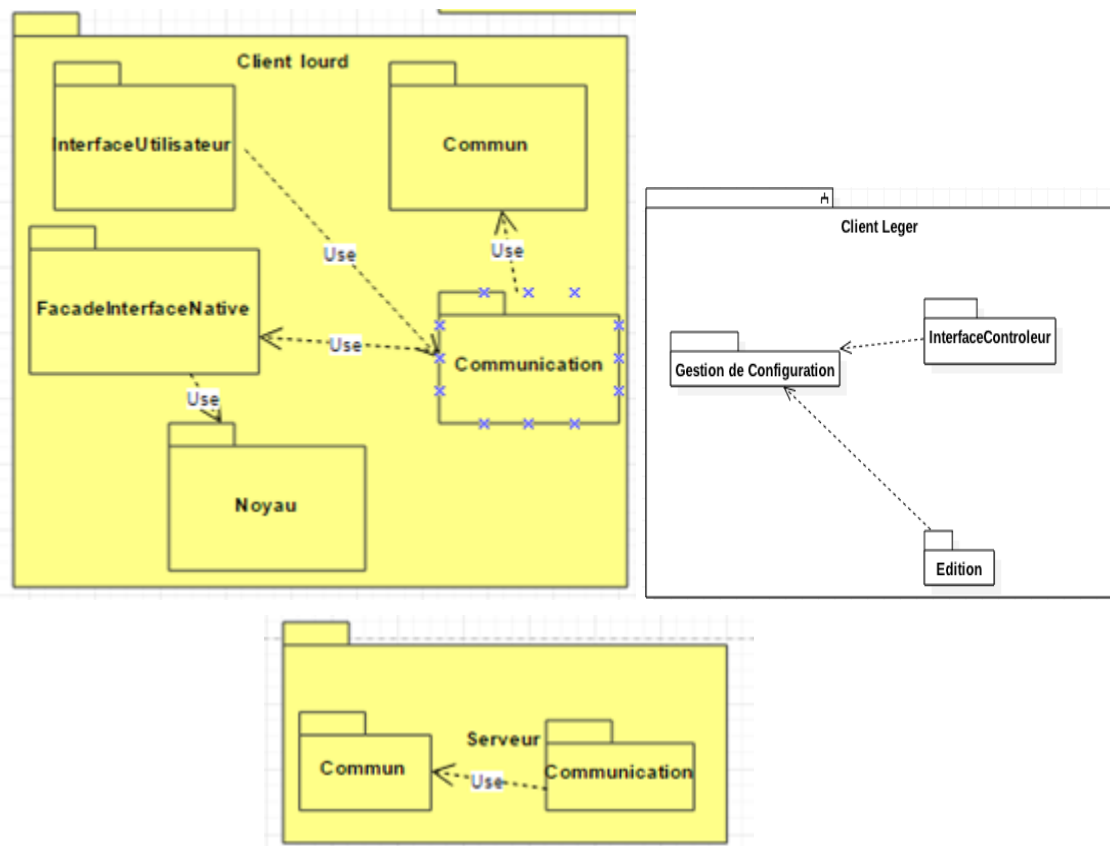
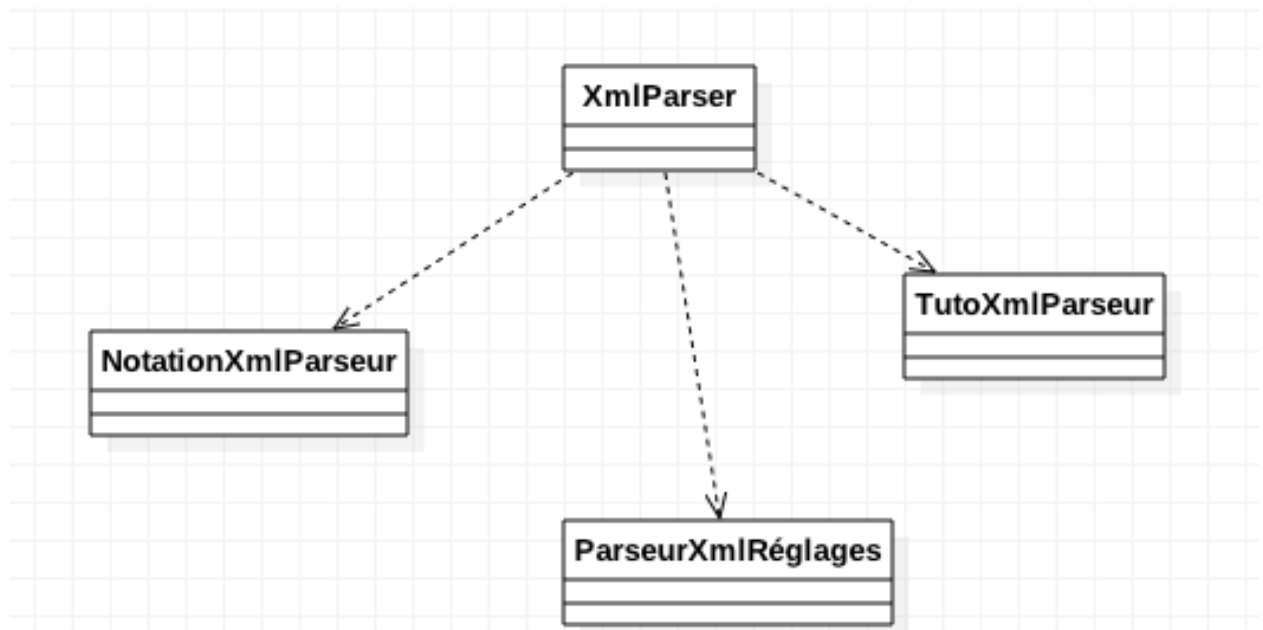


Figure 3 : Diagrammes de paquetages

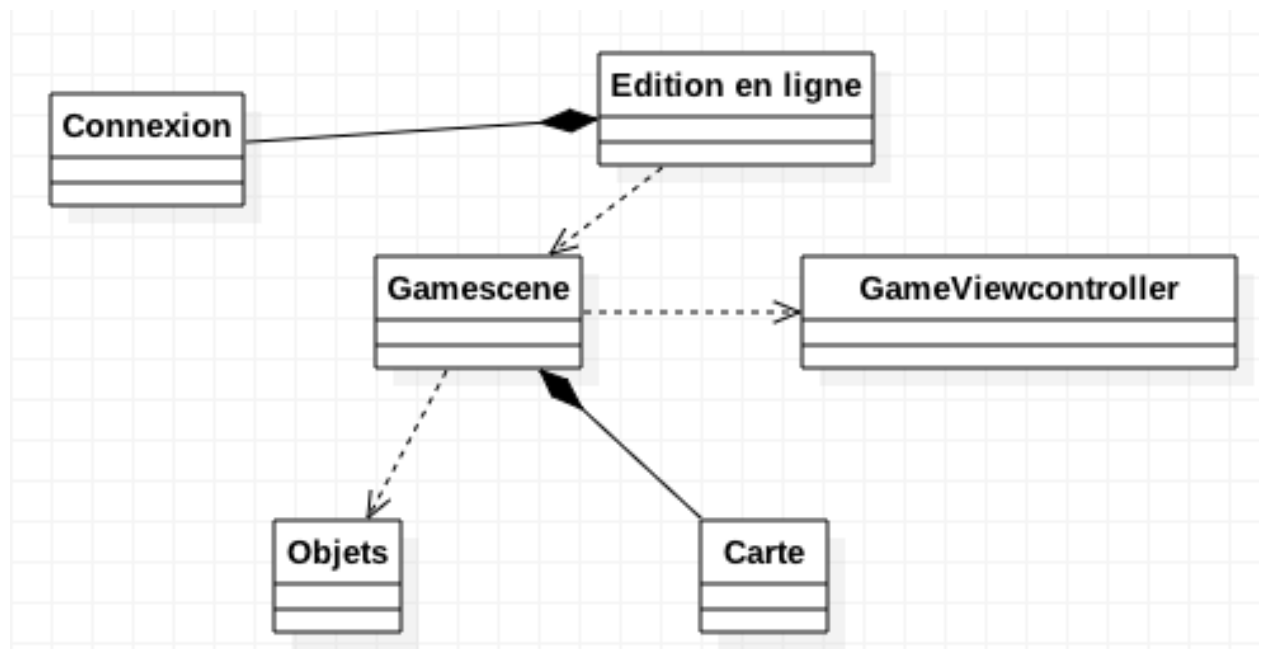
#### 4.1 Client léger

Client Léger	
Description:	Il s'agit de tous les sous-paquetages et des classes interagissant dans ces derniers
Classes incluses:	Toutes les classes des sous-paquetages
Relations:	Aucun
Sous-paquetages:	InterfaceContrôleur, Édition, Gestion de configuration



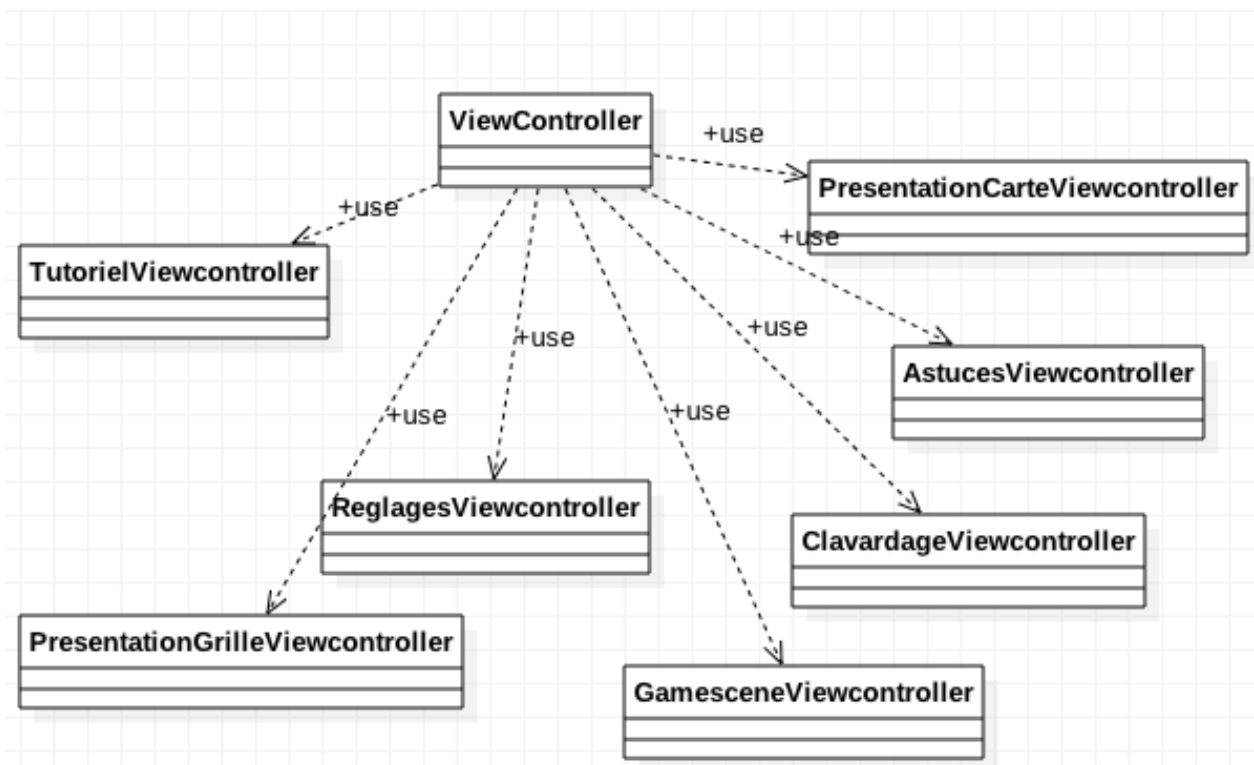
*Figure 4 : Diagramme de classe du paquetage Gestion de Configuration Client léger*

Client Leger: Gestion de Configuration	
Description:	Contient des classes permettant l'enregistrement et la sauvegarde des fichiers XML
Classes incluses:	XmlParser, NotationXmlParseur, TutoXmlParseur
Relations:	est utilisé par InterfaceUtilisateur et Édition
Sous-paquetages:	Aucun



*Figure 5 : Diagramme de classe du paquetage Édition Client léger*

Client Leger : Édition	
Description:	Contient les classes pour l'édition des cartes
Classes incluses:	Objects, Gamescene, GameViewController, Carte, Edition En Ligne, Connexion
Relations:	est utilisé par le paquetage Interface
Sous-paquetages:	Aucun



*Figure 6 : Diagramme de classe Interface utilisateur client léger*

Client Leger: Interface utilisateur	
Description:	Il s'agit des classes de l'interface du client Léger
Classes incluses:	Viewcontroller, TutorialViewcontroller, AstucesViewcontroller, ClavardageViewcontroller, GamesceneViewcontroller, PresentationGrilleViewcontroller, PrésentationCarteViewcontroller, RéglagesViewcontroller
Relations:	Utilise les paquetages Gestion de configuration
Sous-paquetages:	Aucun

## 4.2 Serveur

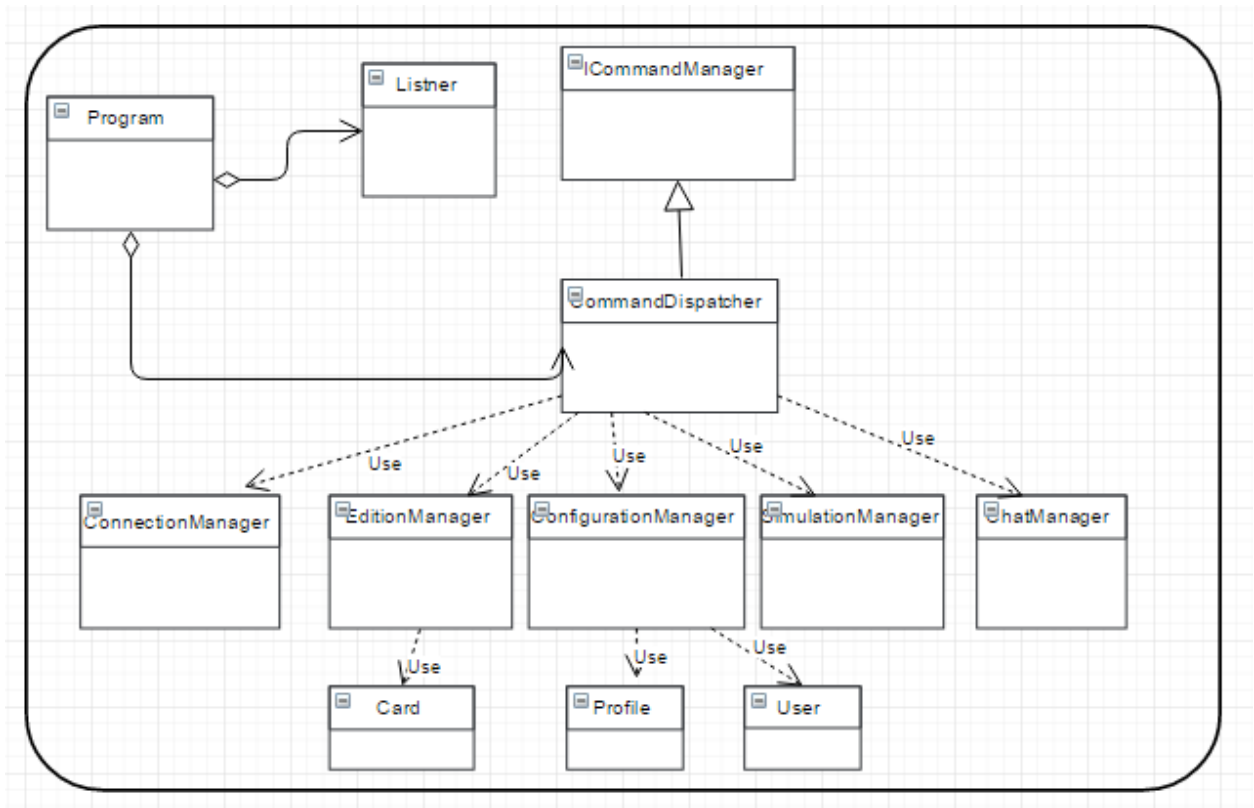


Figure 7 : Diagramme de classe du paquetage Serveur

Serveur	
Description:	Paquetage regroupant toutes classes en lien avec la Connexion et le relais d'information entre les clients.
Classes incluses:	Program, Listener, Card, ICommandManager, CommandDispatcher, ConnectionManager, ConfigurationManager, EditionManager, SimulationManager, ChatManager, User, Profile
Relations:	Aucune
Sous-paquetages:	Aucun

## 4.3 Client lourd

Client lourd	
Description:	Il s'agit de tous les sous-paquetages et des classes interagissant dans ces derniers
Classes incluses:	Toutes les classes des sous-paquetages
Relations:	Aucune

WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11

Sous-paquetages:	Communication, InterfaceUtilisateur, Noyau, FacadeInterfaceNative, Commun
------------------	---

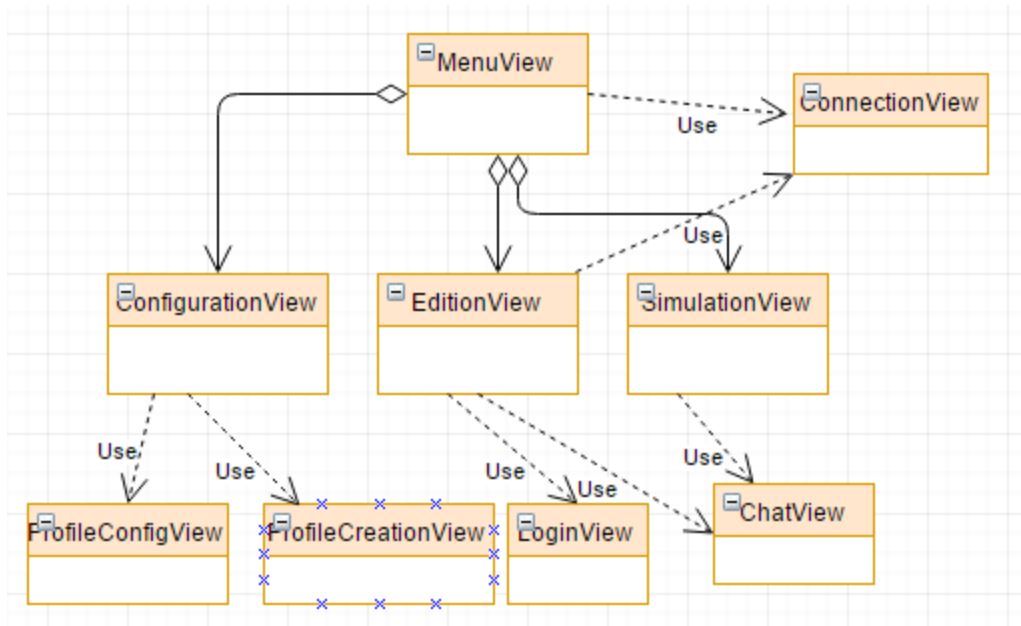
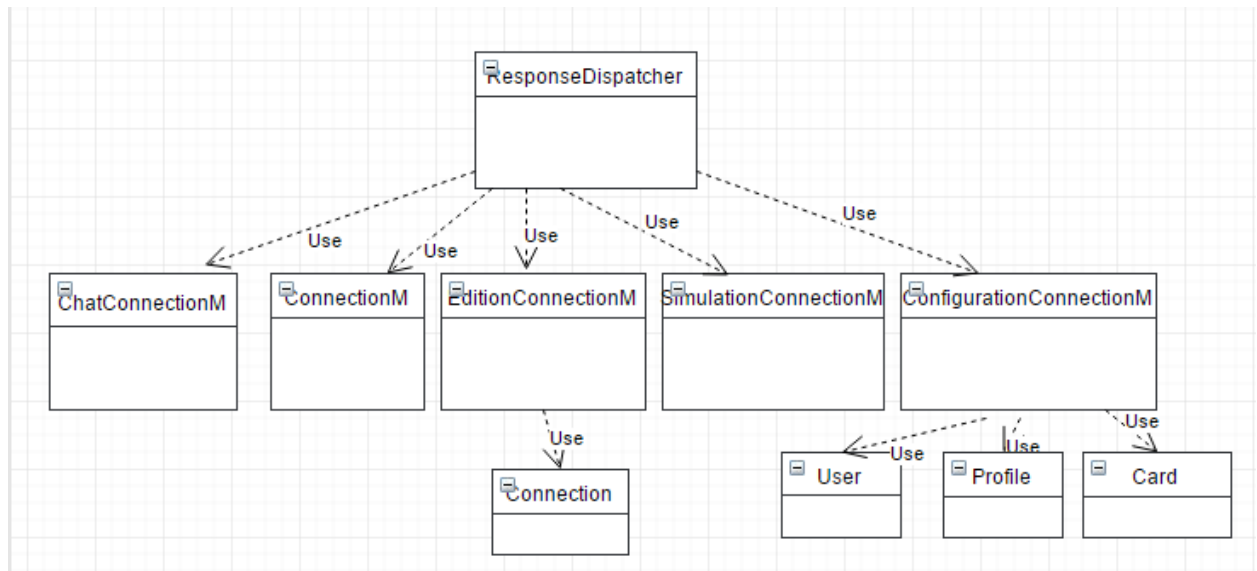


Figure 8 : Diagramme de classe du paquetage InterfaceUtilisateur (client lourd)

InterfaceUtilisateur: Client lourd	
Description:	Paquetage regroupant toutes classes liées avec l'interface utilisateur du client lourd.
Classes incluses:	MenuView, ConfigurationView, EditionView, LoginView, SimulationView, ProfileCreationView, ProfileConfig, ConnectionView et ChatView.
Relations:	Utilise le paquetage Communication.
Sous-paquetages:	Aucun



*Figure 9 : Diagramme de classe du paquetage Communication (client lourd)*

Communication: Client lourd	
Description:	Paquetage regroupant toutes classes liées à la connexion réseau et à la transmission de données entre le réseau et l'application.
Classes incluses:	ResponseDispatcher, ChatConnectionM, EditionConnectionM, EditionConnectionM, SimulationConnectionM, ConfigurationConnectionM, Connection, user, profile, Card
Relations:	Est utilisé par InterfaceUtilisateur et utilise FacadeInterfaceNative.
Sous-paquetages:	Aucun

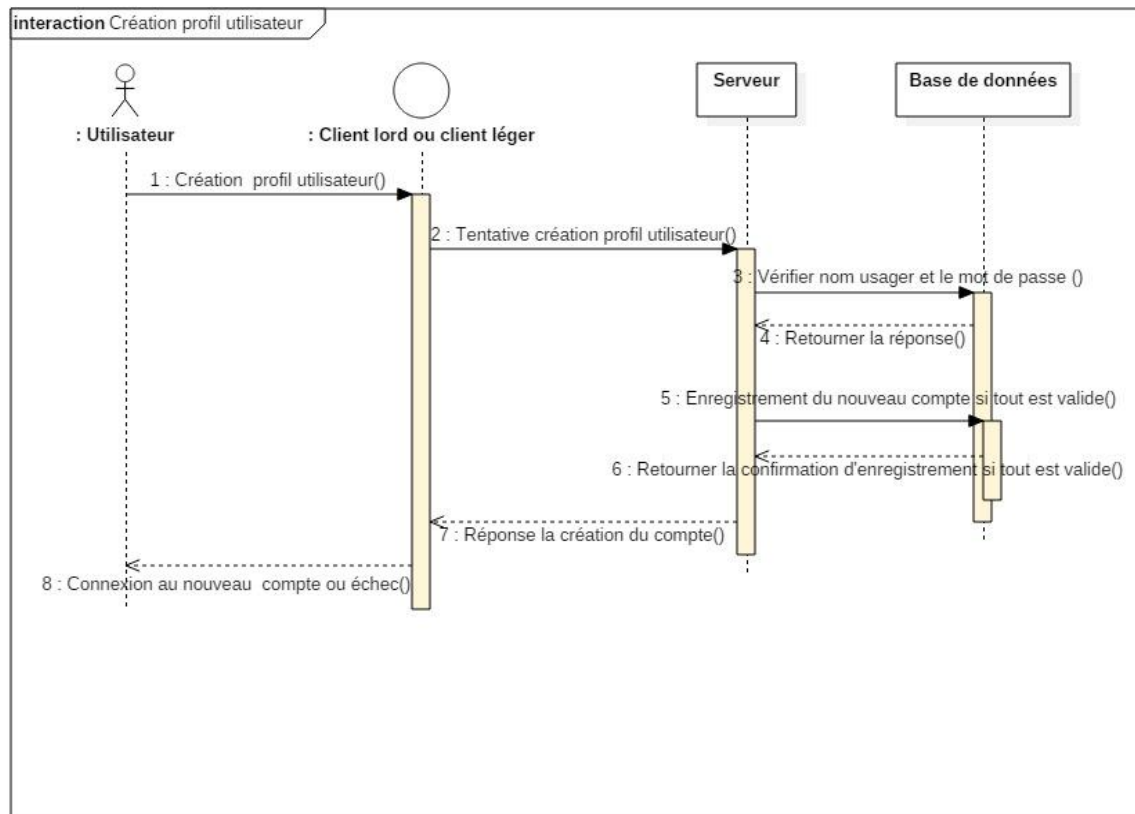
## 5. Vue des processus

Dans cette section, nous illustrons les interactions entre les processus à l'aide de diagrammes de séquences.

### 5.1 Création du profil utilisateur

Nous considérons dans le premier diagramme qu'un utilisateur peut se créer un profil utilisateur afin de l'utiliser pour se connecter au jeu. Pour cela l'utilisateur doit envoyer un nom d'utilisateur et un mot de passe valide (nom utilisateur qui n'existe pas auparavant dans la base de données) et attendre la confirmation du serveur après que ce dernier vérifie la validité des informations des utilisateurs dans la base de données.

WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11

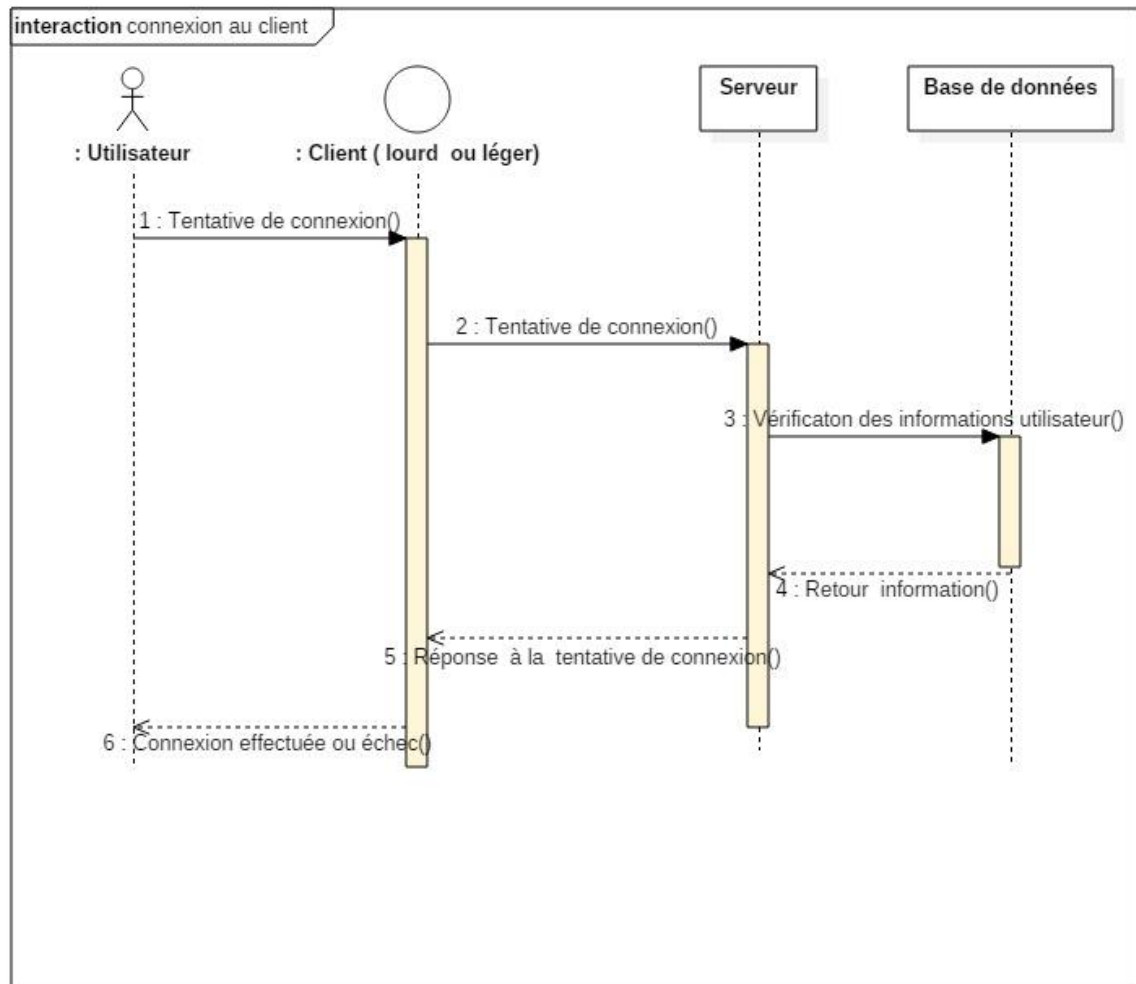


*Figure 10 : Vue des processus : Création du profil utilisateur*

## 5.2 Connexion au client lourd ou client léger

Dans le deuxième diagramme, nous considérons que la connexion au client léger et lourd est similaire. Un utilisateur peut se connecter au jeu en entrant le nom d'utilisateur et le mot de passe. Si ces informations sont erronées, le jeu devra retourner l'erreur de connexion.

WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11



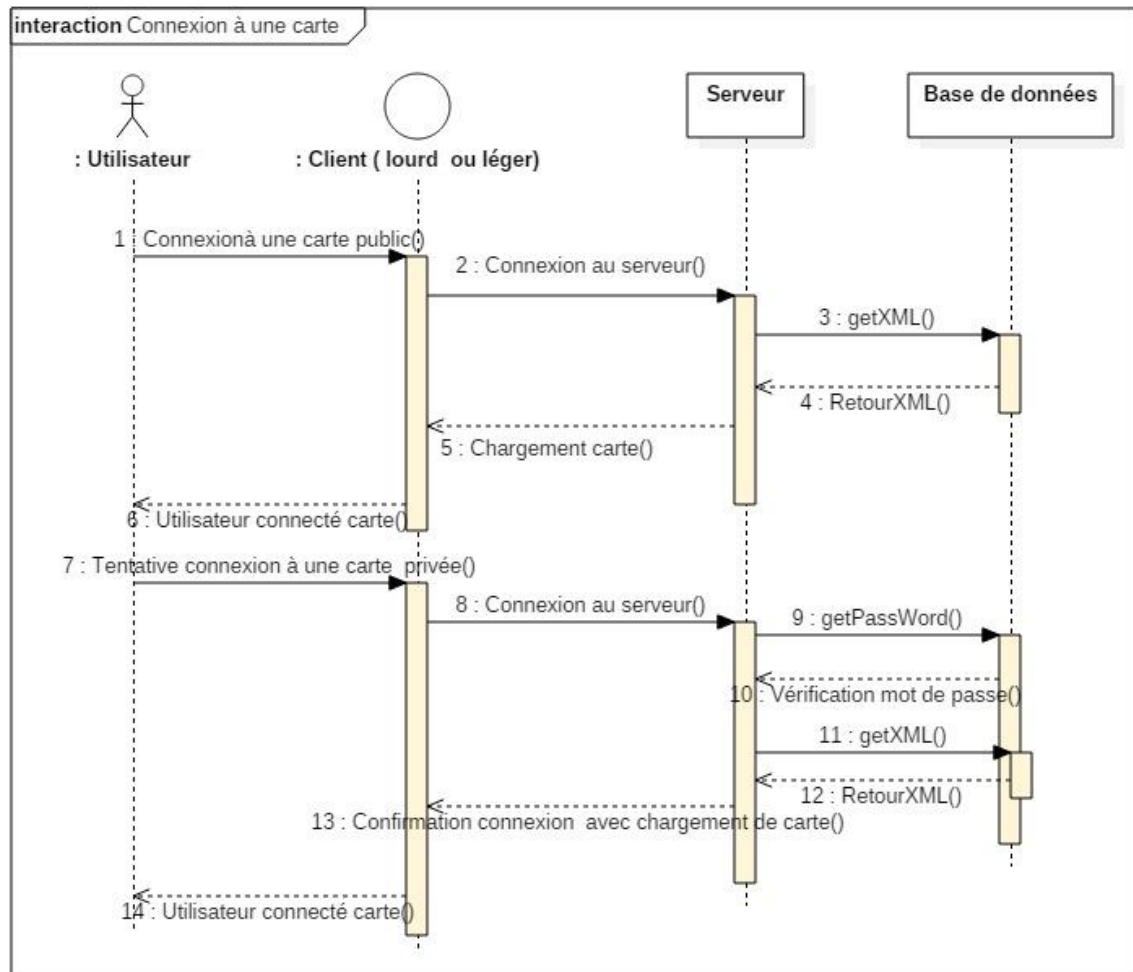
*Figure 11 : Vue des processus : Connexion au client*

### 5.3 Connexion à une carte privée ou publique

Un utilisateur en ligne peut se connecter à une carte. Cette carte peut être privée ou publique. N'importe qui peut accéder à une carte publique. Si la carte est privée seulement les utilisateurs qui entrent le bon mot de passe peuvent y accéder.



WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11

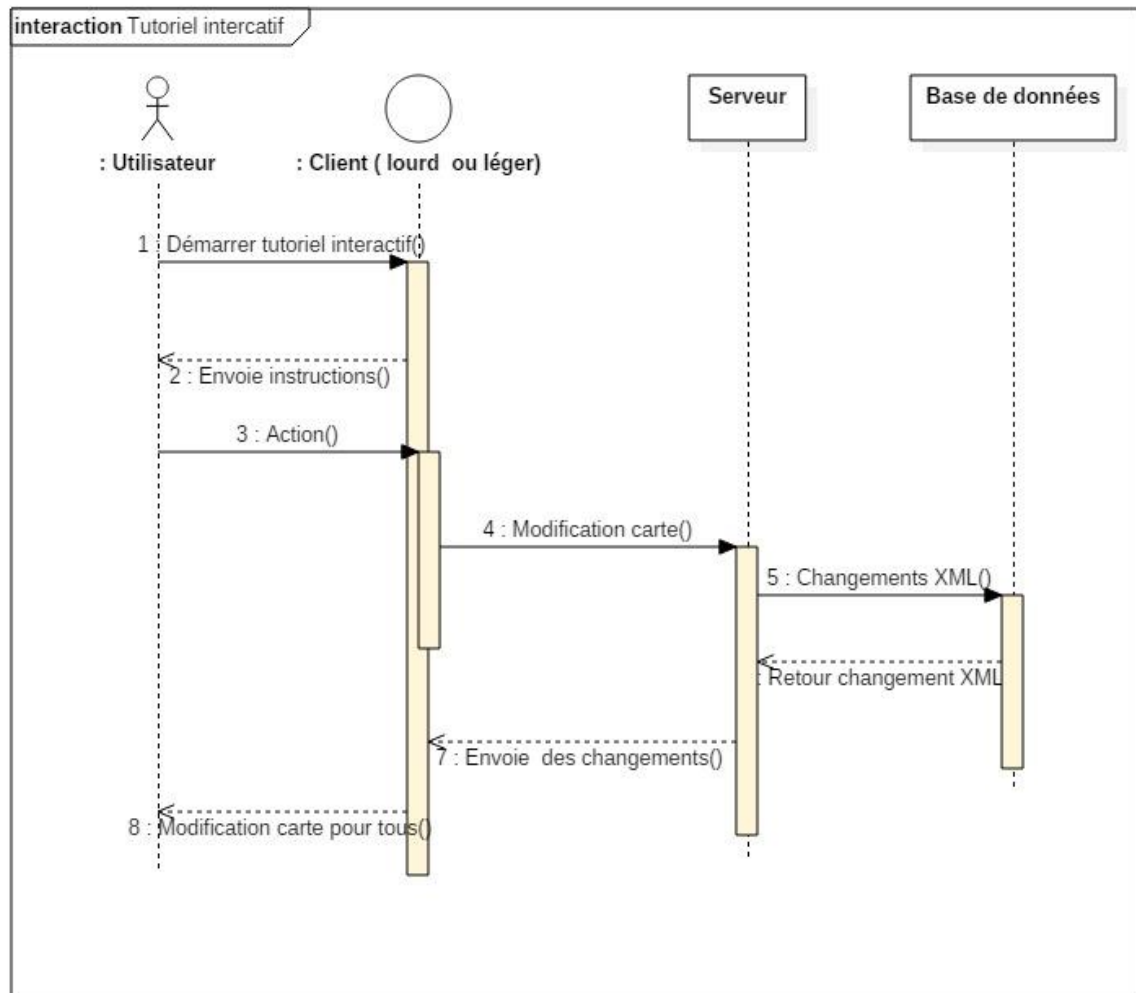


*Figure 12 : Vue des processus : Connexion à une carte*

## 5.4 Tutoriel interactif

Ci-dessous le diagramme de séquence concernant la fonctionnalité des tutoriels interactifs de notre système.

WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11

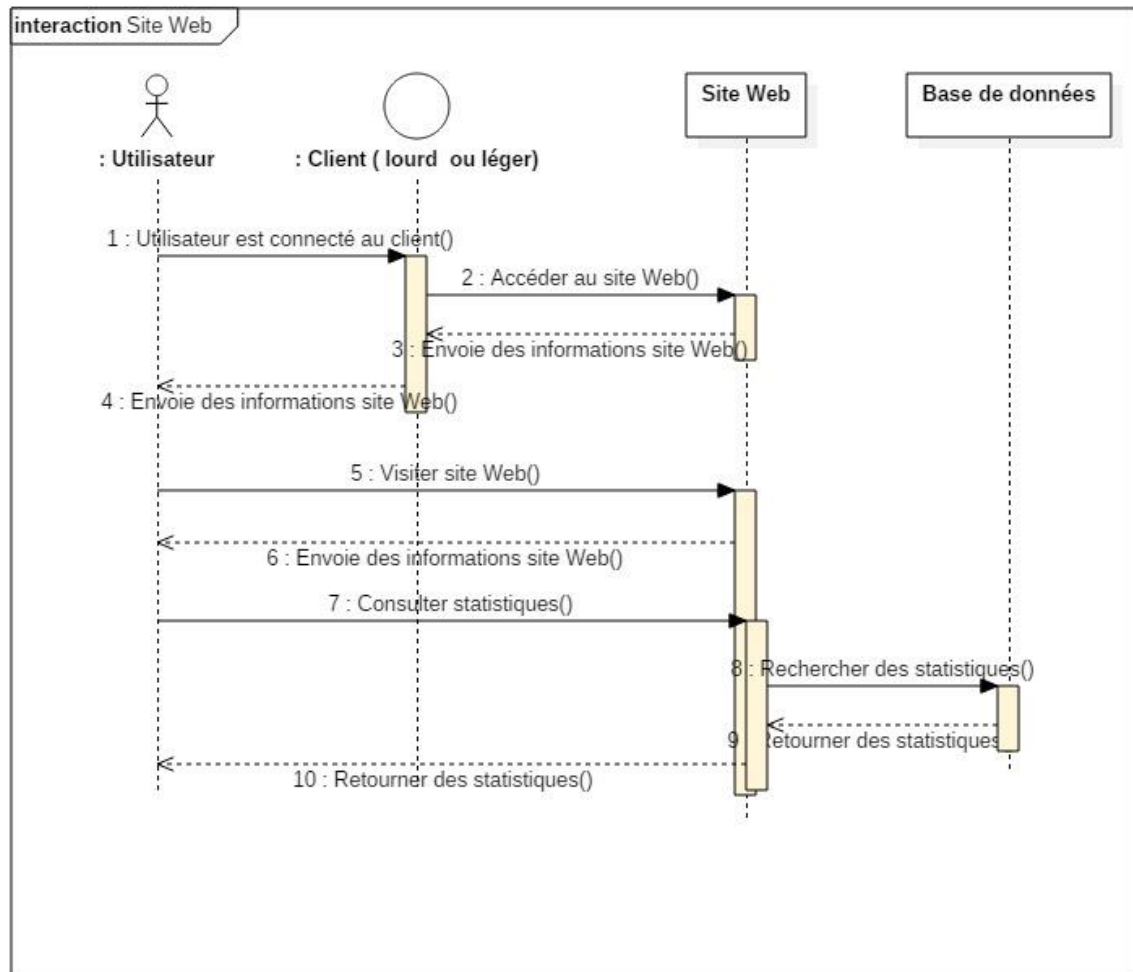


*Figure 13 : Vue des processus : Tutoriel interactif*

## 5.5 Site Web

L'utilisateur peut accéder au site web via notre système. Ce site web affiche des informations sur les utilisateurs, les cartes, les statistiques, les groupes des utilisateurs, etc.

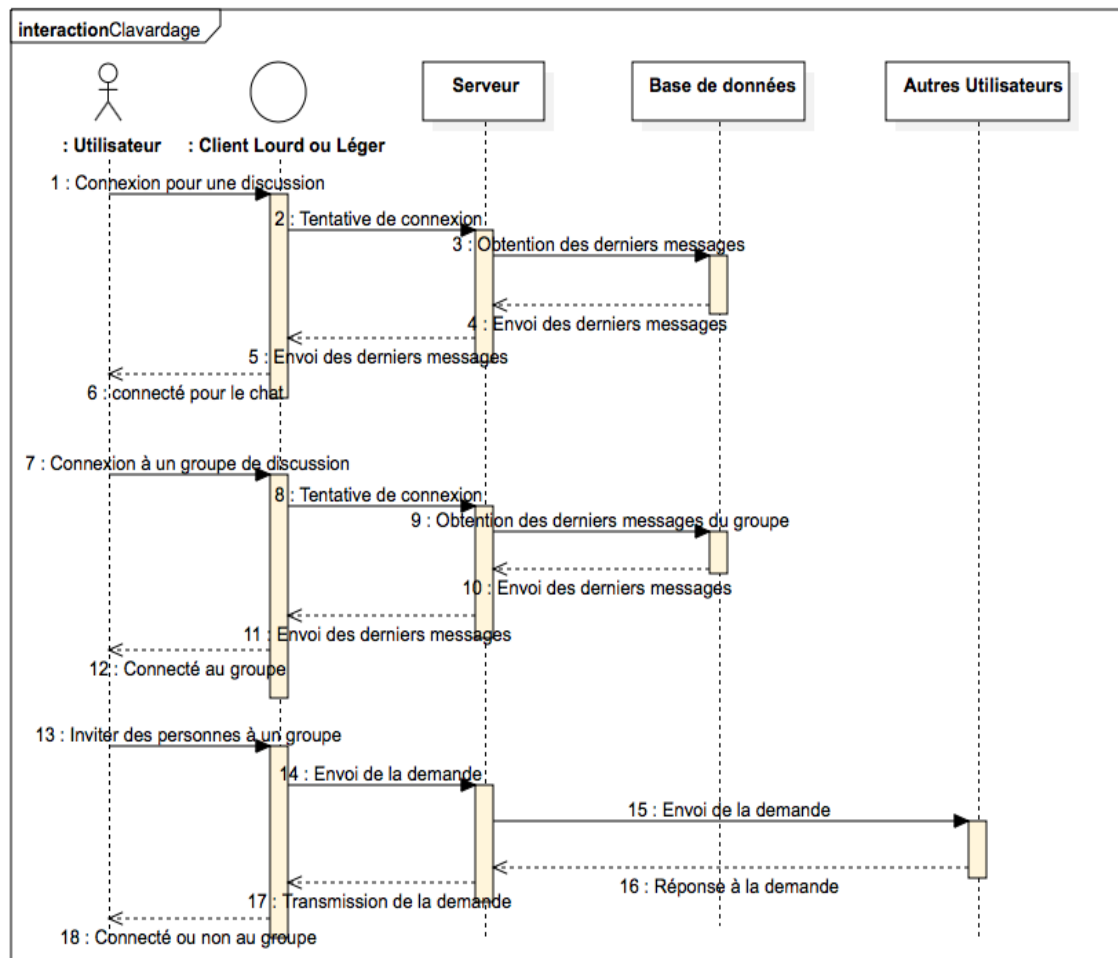
WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11



*Figure 14 : Vue des processus : Site web*

## 5.6 Clavardage

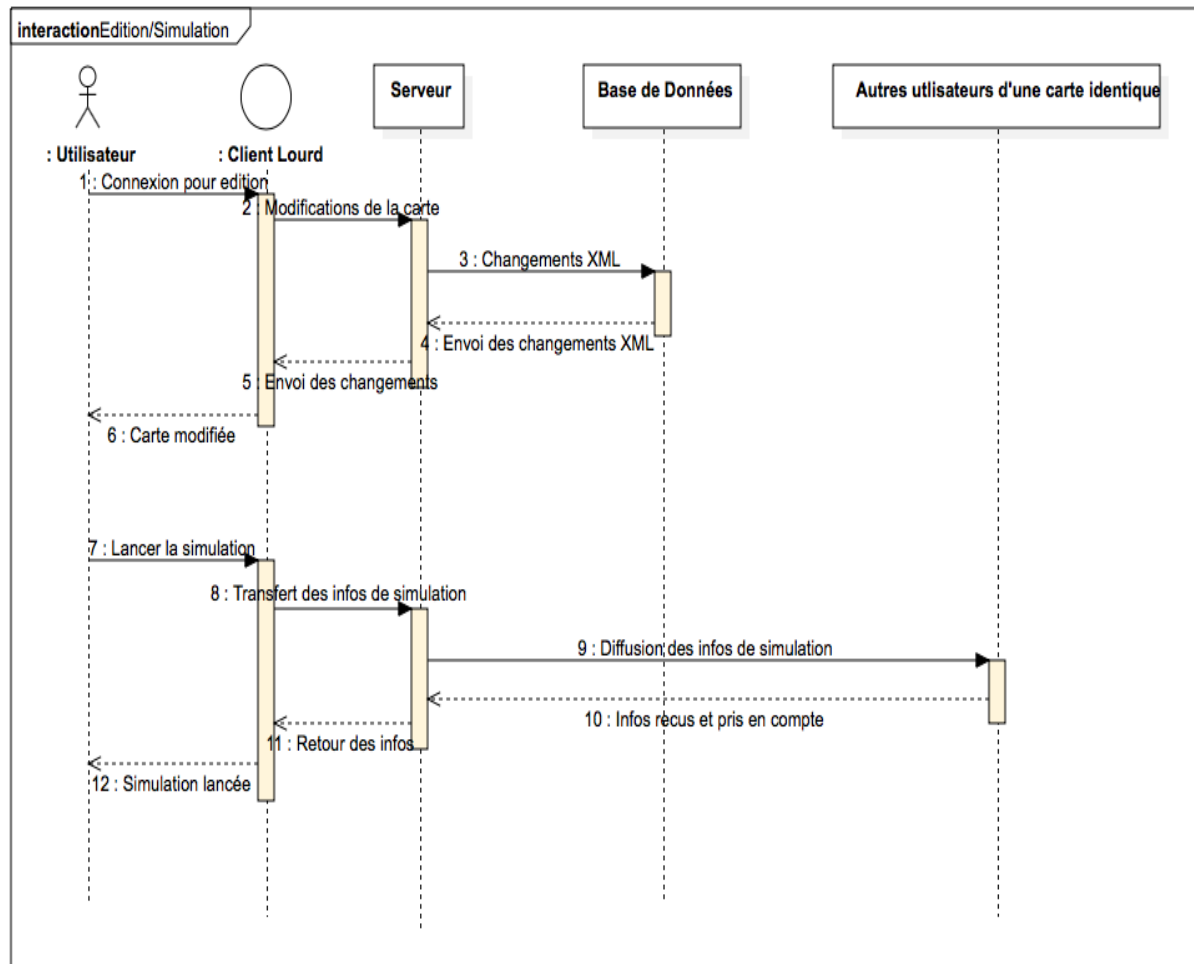
Ce Diagramme de séquence met en évidence 3 cas de figure dans le clavardage : premièrement l'utilisateur se connecte pour échanger avec une personne. Deuxièmement l'utilisateur se connecte et veut joindre un groupe de discussion et enfin l'utilisateur invite d'autres utilisateurs à joindre un groupe de discussion qu'il a instauré. Le clavardage se déroule en général de la même façon tant dans le client lourd que dans le client léger.



*Figure 15 : Vue des processus Clavardage*

## 5.7 Édition/Simulation

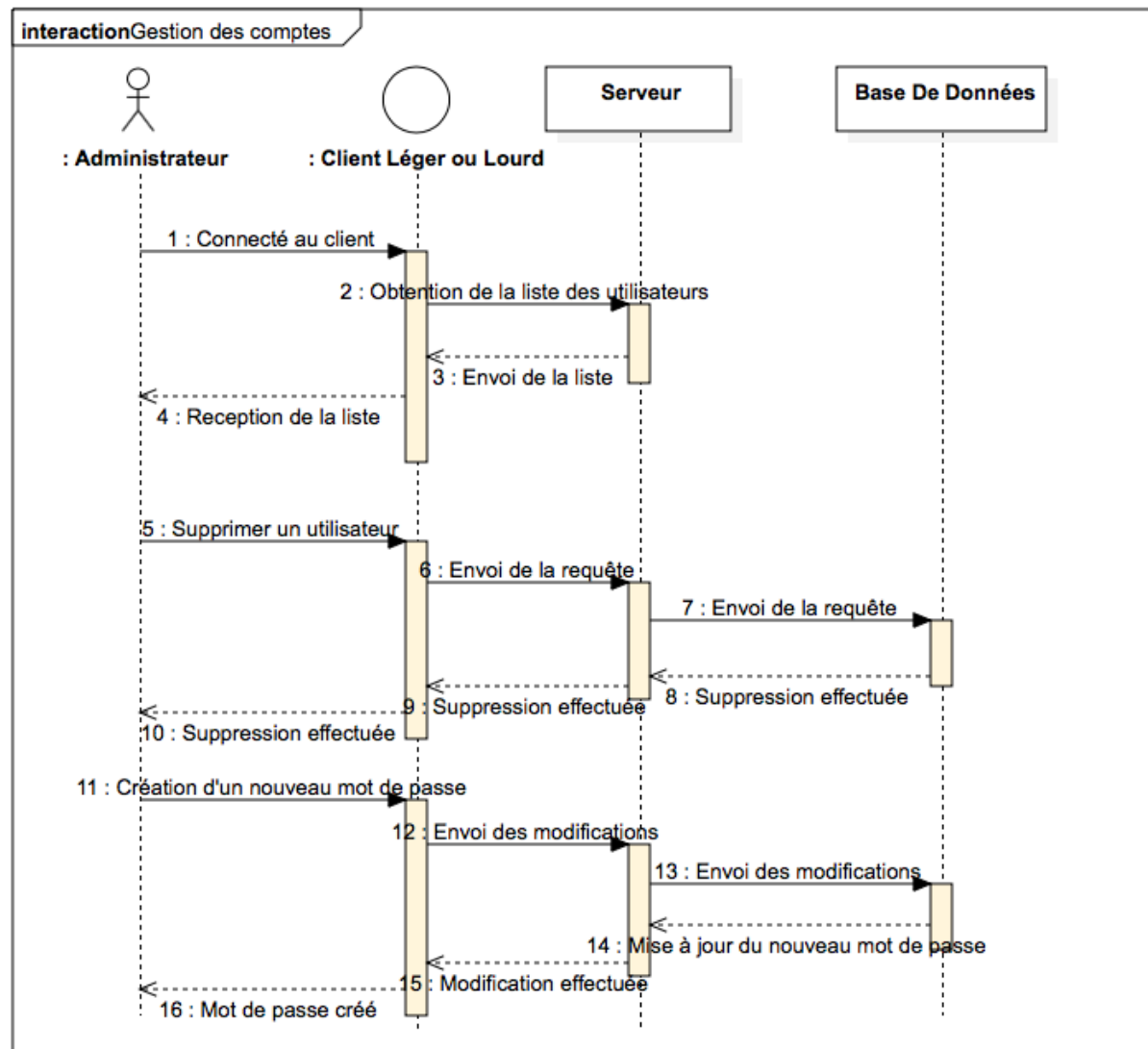
Le processus d'édition d'une carte consiste à créer une zone de simulation, ajouter des murs des poteaux ou des lignes etc. Ce processus est le même dans les deux clients (lourd et léger). La simulation quant à elle s'effectue essentiellement dans le client lourd. Elle nécessite entre autres l'utilisation d'une carte. En plus de l'édition, nous mettons en exergue la simulation simultanée effectuée par plusieurs utilisateurs sur une même carte dans ce cas de figure.



*Figure 16 : Vue des processus Édition/Simulation*

## 5.8 Gestion des comptes

En plus d'effectuer toutes les tâches d'un simple utilisateur, l'administrateur s'occupe aussi de la gestion des comptes. Deux tâches sont mises de l'avant dans ce diagramme : la possibilité de supprimer un utilisateur pour différentes raisons et la création d'un nouveau mot de passe pour un utilisateur qui a oublié le sien.

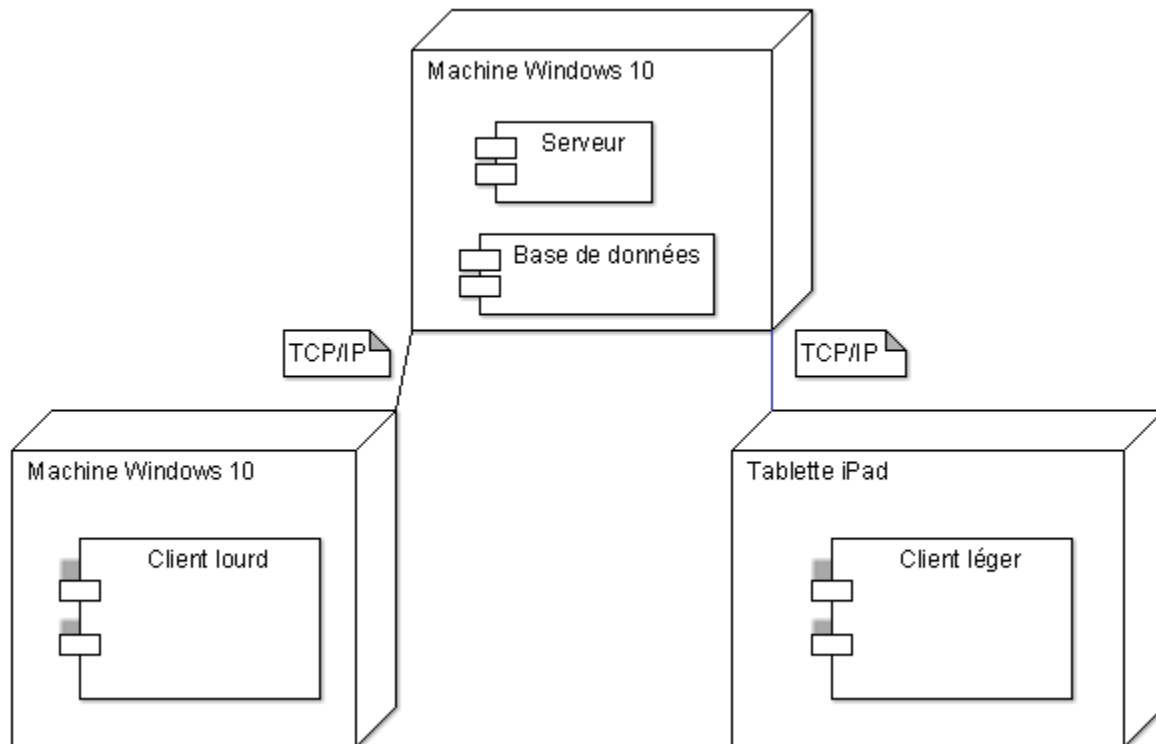


*Figure 17 : Vue de processus : Gestion des comptes utilisateurs*

## 6. Vue de déploiement

Dans cette partie, nous présentons le diagramme de déploiement de notre application. Nous montrons comment les composantes physiques, notamment l'ordinateur et l'iPad sont configurés pour supporter le simulateur de robot.

WALL-E2	Version: 2.0
Document d'architecture logicielle	Date: 2017-04-11



*Figure 18 : Vue de déploiement*

## 7. Taille et performance

L'application est de taille moyenne. De ce fait, elle ne nécessitera pas un grand déploiement au niveau du design. La performance quant à elle est importante, car l'édition et la simulation sont possibles simultanément en réseau. L'application doit être toujours fluide et agréable, quel que soit le client concerné. En se basant sur les mesures préliminaires que nous avons effectuées, si le nombre de joueurs présents reste inférieur à 10, il n'y aura pas de baisse considérable en performance en prenant comme acquis que la performance du réseau reste acceptable. Le rafraîchissement est d'au moins 30 images par secondes. Le temps de navigation à travers les différents clients est court. Les changements effectués par un utilisateur lors d'une simulation ou d'une édition doivent être visibles presque instantanément par tous les autres clients, et doivent être sauvegardés sur le serveur de façon quasi immédiate dépendamment de la taille de l'échange.