Document d'architecture logicielle

Version 2.0

Historique des révisions

Date	Version	Description	Auteur
2023-03-19	1.0	Structure cas d'utilisation	Julien
2023-03-21	1.0	Diagramme de paquetage et classes	Majeed
2023-03-21	1.0	Diagrammes de séquence	Maxime
2023-04-18	2.0	Ajustements pour changements du sprint 3	Majeed, Maxime

Table des matières

1. Introduction **4**

2. Vue des cas d'utilisation **5**

a. Diagramme de contexte générale 5

b. Diagramme de contexte : CU-01 Jouer à un jeu seul 6

c. Diagramme de contexte : CU-02 Jouer à un jeu Multijoueur 7

d. Diagramme de contexte : CU-04 Administrer les jeux 8

3. Vue des processus **9**

a. Diagramme de séquence: partie 1v1 9

b. Diagramme de séquence: partie coopérative en contre la montre 9

c. Diagramme de séquence: visualisation de reprise de partie 11

d. Diagramme de séquence: visualisation de l'historique des parties 11

e. Diagramme de séquence: modification des paramètres de jeu contre la montre par le panneau de configuration 13

f. Diagramme de séquence: obtention d'indices 13

g. Diagramme de séquence: Remise a valeurs défault des paramètres de jeu contre la montre par le panneau de configuration 14

4. Vue logique **15**

a. Diagramme de paquetage général du projet: 15

b. Paquetage: Vue 16

Vue 16

c. Paquetage: Logique Client 16

Logique Client 16

d. Paquetage: Communication client 16

Communication client 16

Diagramme de paquetage application client 17

e. Paquetage: Communication serveur 18

Communication serveur 18

f. Paquetage: Logique serveur 18

Logique serveur 18

g. Paquetage: Persistance 18

Persistance 18

5. Vue de déploiement **20**

a. Diagramme de déploiement 20

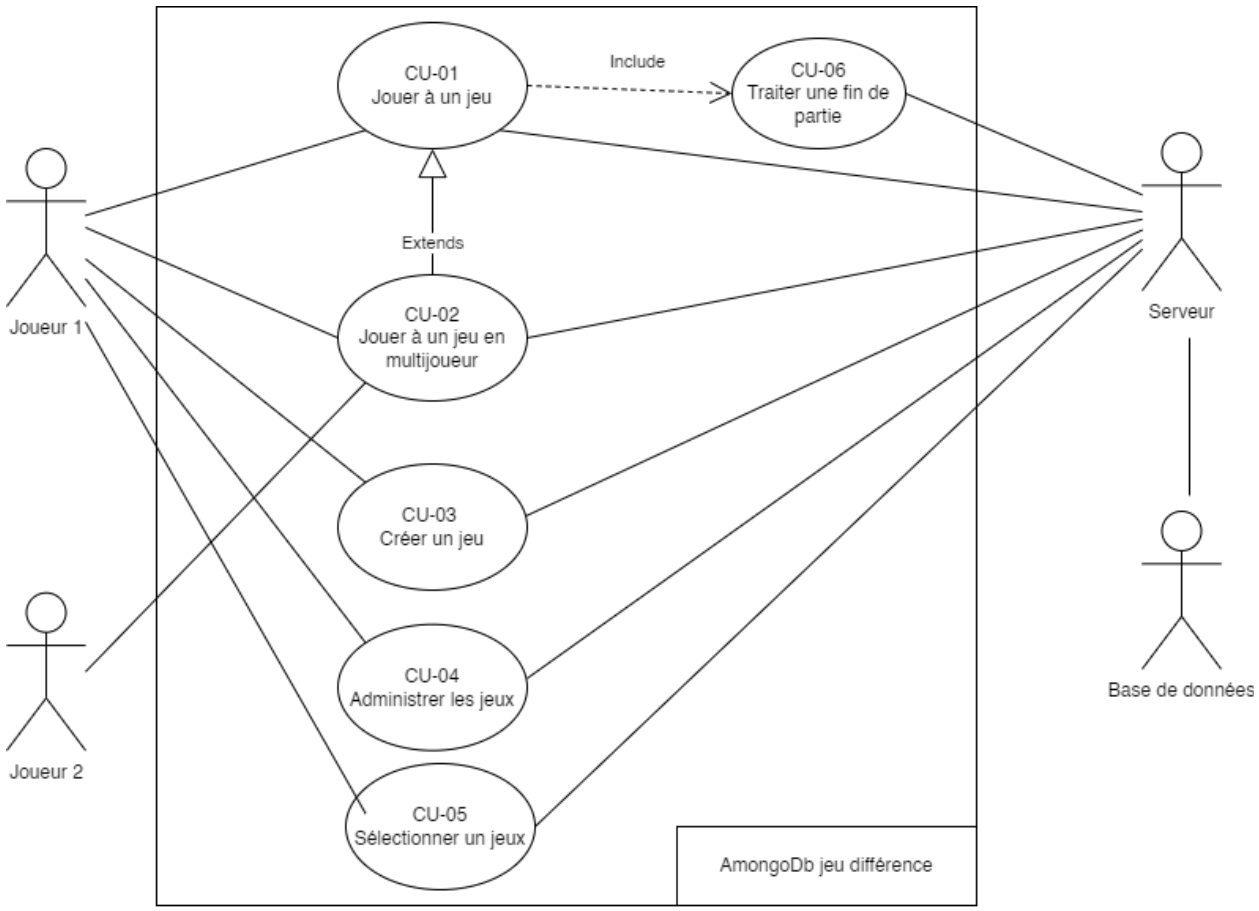
Document d'architecture logicielle

1. Introduction

Cette documentation a pour objectif de fournir une vue d'ensemble du système avec les fonctionnalités du sprint 3 à l'aide de différents diagrammes UML. Cette documentation se compose de quatres parties principales: la vue des cas d'utilisation, la vue des processus, la vue logique et vue de déploiement.
La vue des cas d'utilisation permet de comprendre les différentes interactions possibles entre les utilisateurs et le système. Elle décrit les fonctionnalités du système à partir du point de vue des acteurs impliqués.
La vue des processus décrit comment le système fonctionne en interne. Elle montre comment les différentes fonctions du système interagissent entre elles pour réaliser les fonctionnalités du système.
La vue logique présente la structure interne du système. Elle montre comment les différentes classes et objets du système sont organisés et interagissent entre eux.
La vue de déploiement décrit comment le système sera déployé sur les différentes plates-formes informatiques. Elle montre comment les différents composants du système seront distribués sur les différents ordinateurs et serveurs.

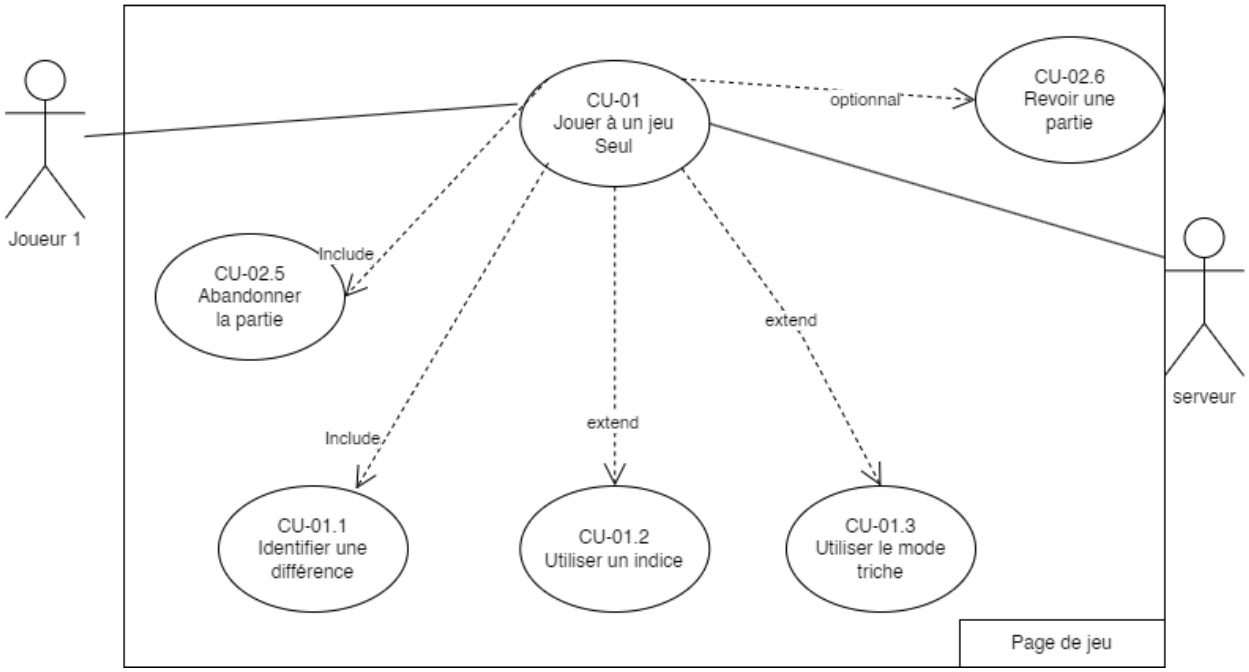
2. Vue des cas d'utilisation

a. Diagramme de contexte générale



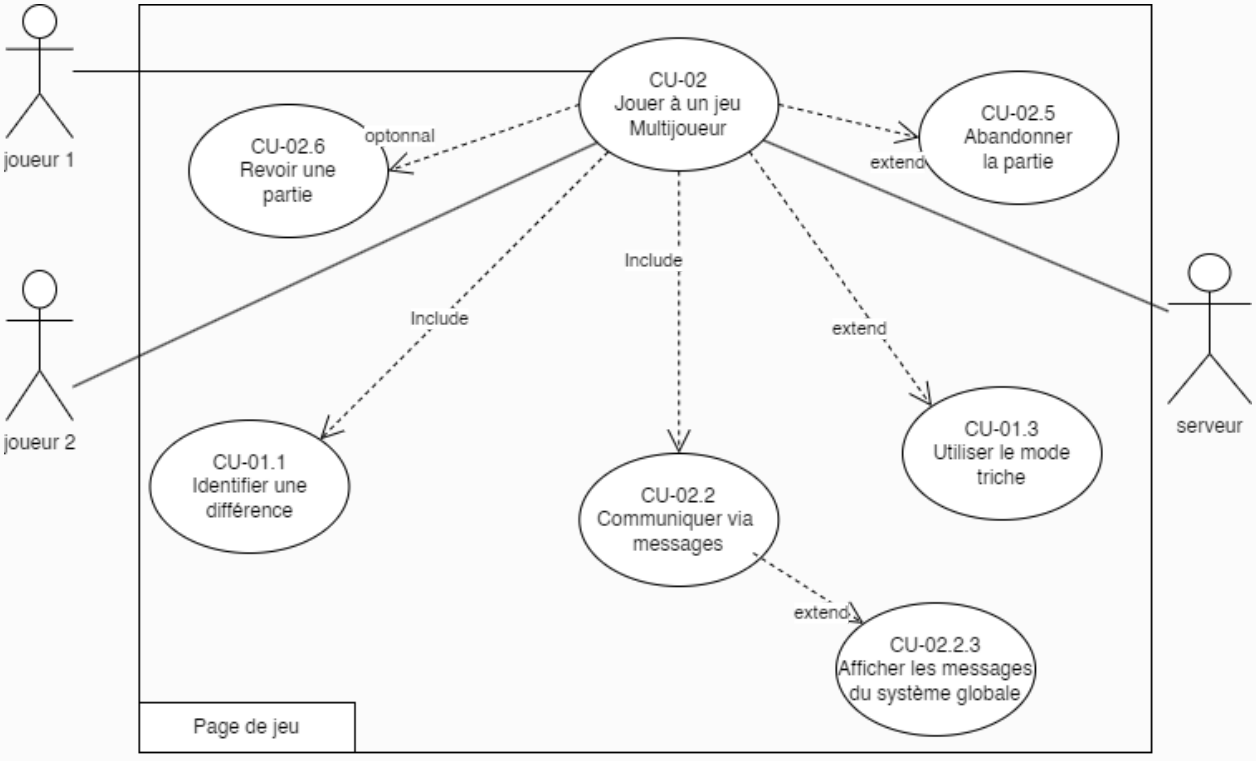
* Le joueur 2 n'a pas été connecté à tous les CU pour clarté.

b. Diagramme de contexte : CU-01 Jouer à un jeu seul



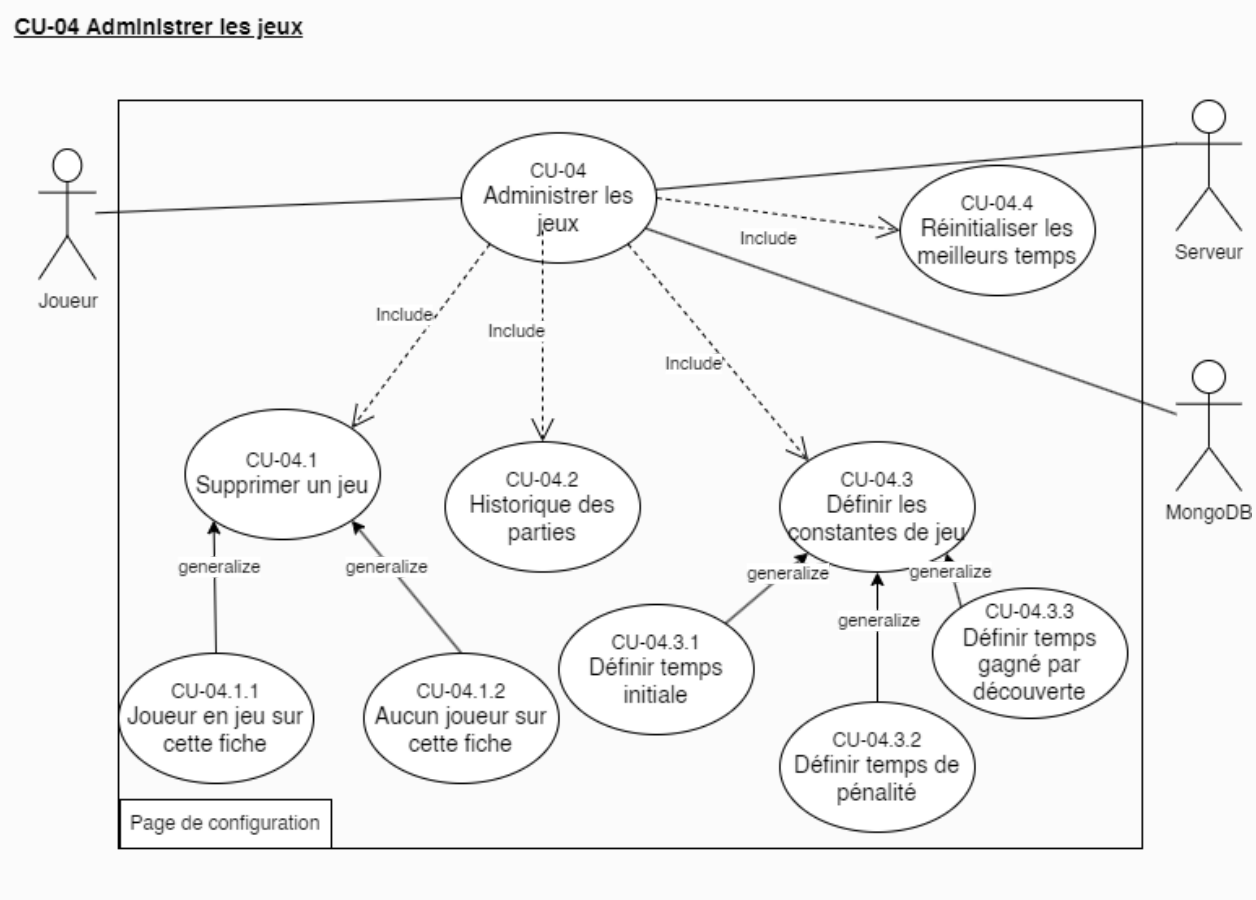
Cas d'utilisation:	**CU-01 Jouer à un jeu seul**
Acteur(s):	**Joueur**
Acteur(s) secondaire:	**Serveur**
Type:	**Primaire**
Description:	**Le joueur cherche une différence entre deux images, si il est en mode classique un compteur de différence sera incrémenté et la partie se terminera une fois toutes les différences trouvées. Si il est en mode temps limité, trouver une différence changera la paire d'images, la partie continue tant que toutes les images n'ont pas été présentées au joueur ou qu'il reste du temps.**

c. Diagramme de contexte : CU-02 Jouer à un jeu Multijoueur



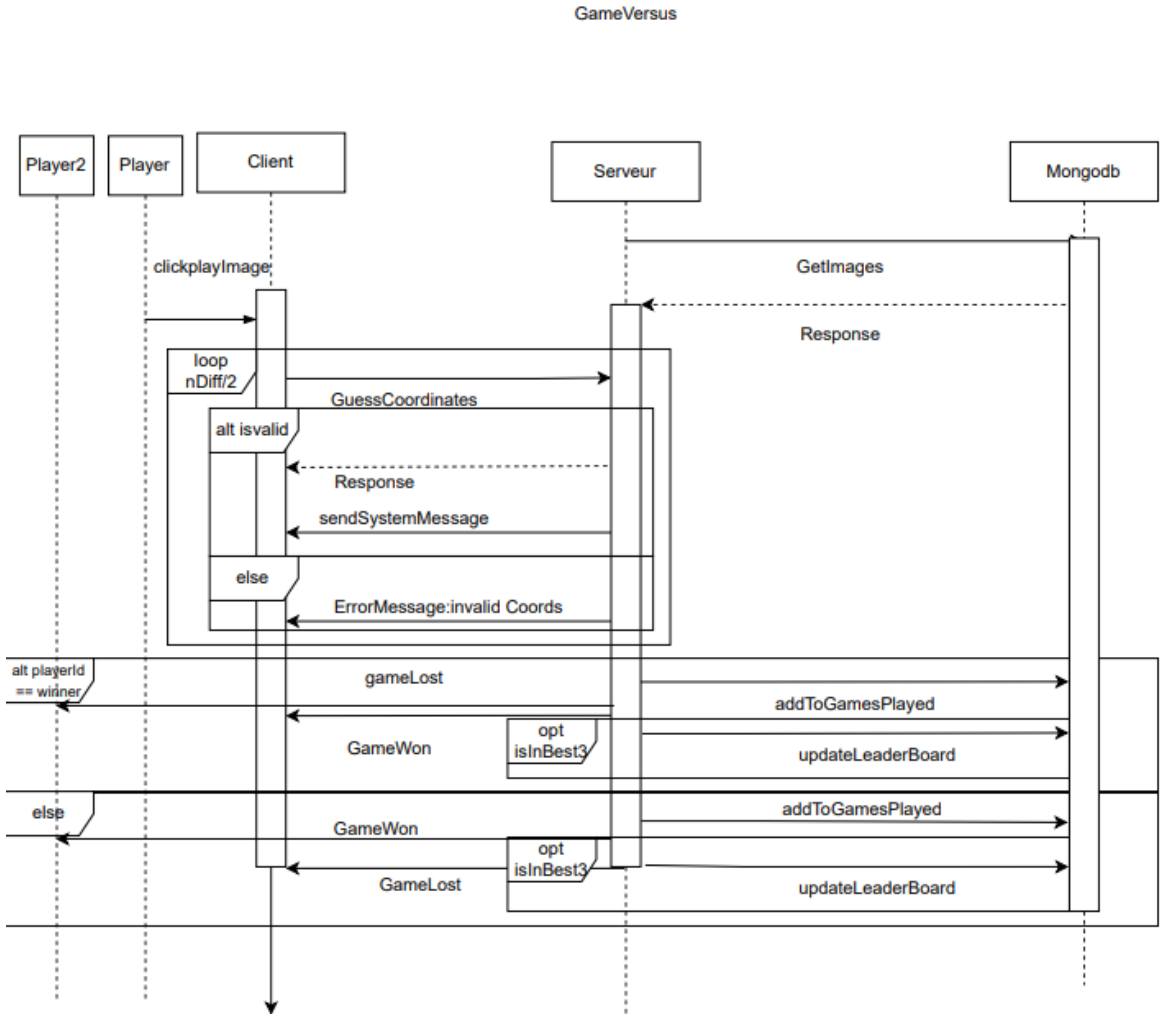
Cas d'utilisation:	**CU-02 Jouer à un jeu en multijoueur**
Acteur(s):	**Deux Joueur**
Acteur(s) secondaire:	**Serveur**
Type:	**Primaire**
Description:	**Les joueurs cherchent une différence entre deux images, si ils sont en mode classique le compteur du joueur qui a trouvé sera incrémenté et la partie se terminera une fois que la moitié des différences auront été trouvé par un joueur ou si l'un des joueurs quitte. S' ils sont en mode temps limité, trouvé une différence changera la paire d'images des deux joueurs, la partie continue tant que toutes les images n'ont pas été présentées au joueur ou qu'il reste du temps. Dans le cas où un joueur quitte la partie se transformera en temps limité solo pour que l'autre joueur puisse continuer.**

d. Diagramme de contexte : CU-04 Administrer les jeux



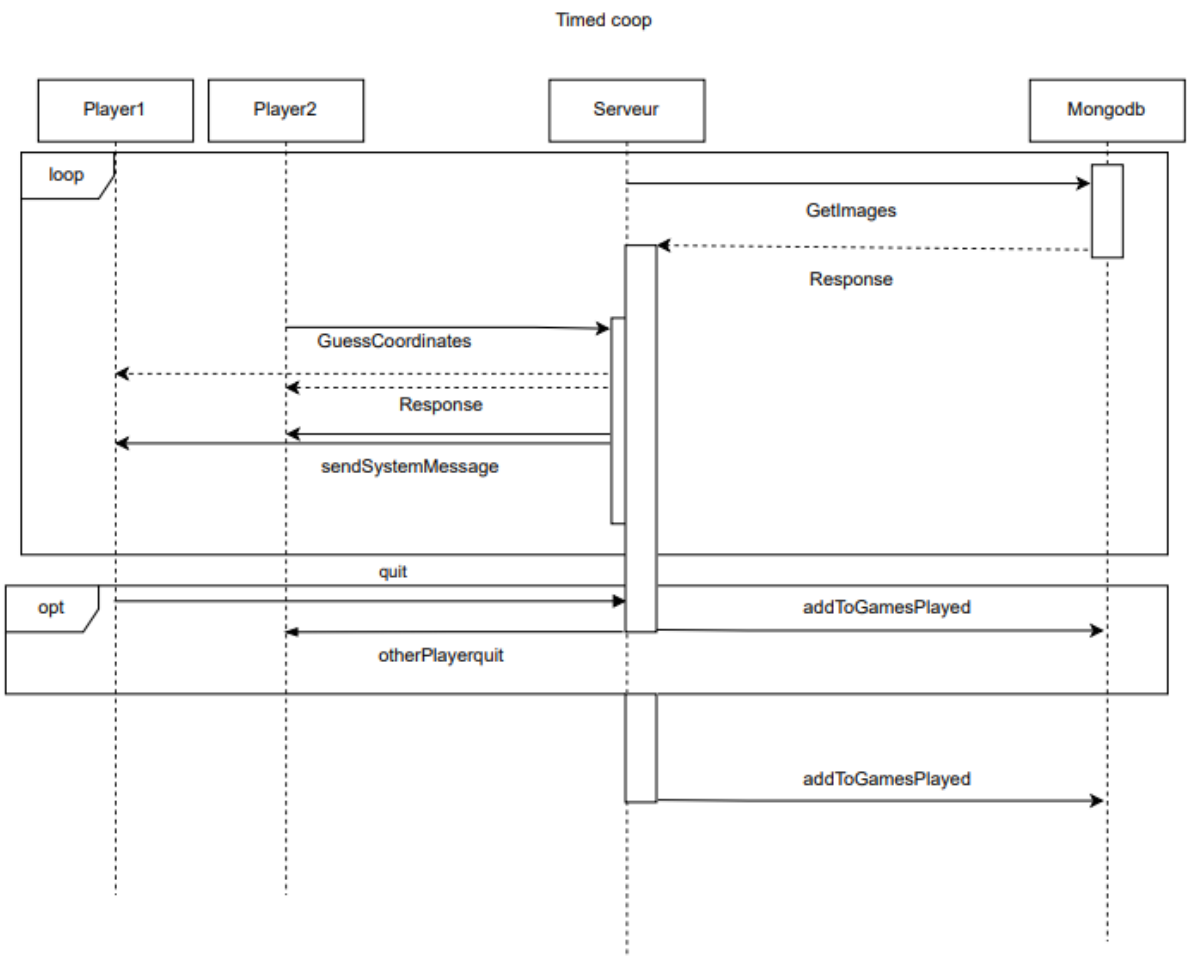
3. Vue des processus

a. Diagramme de séquence: partie 1v1



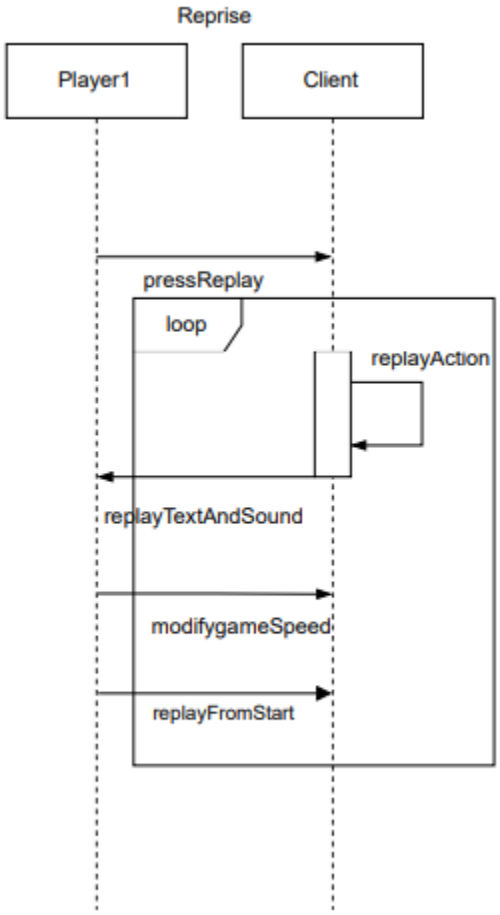
Les messages envoyés au joueur 2 sont en fait envoyés au client 2.

b. Diagramme de séquence: partie coopérative en contre la montre



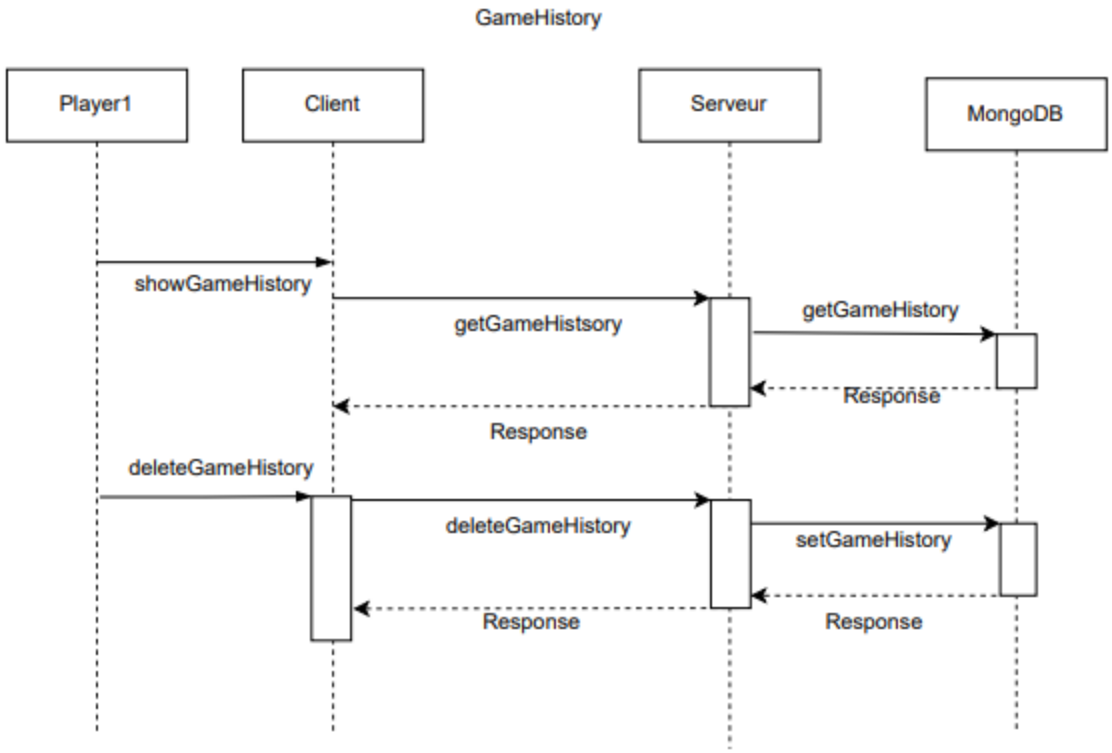
Note: timed solo utilise les mêmes séquences, sauf que les messages du serveur arrivent à un seul client. Toutes les différences se font au niveau du matchmaking

c. Diagramme de séquence: visualisation de reprise de partie

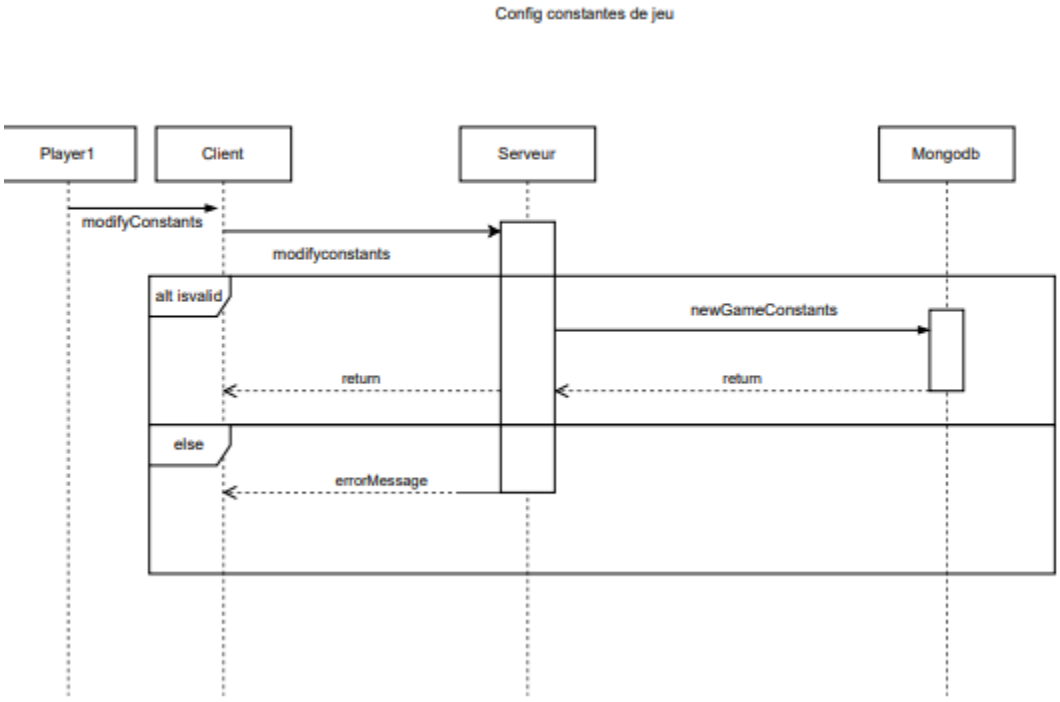


La reprise est entièrement locale. Tous les boutons pause et modérateurs de vitesses ne font que modifier la vitesse de replay.

d. Diagramme de séquence: visualisation de l'historique des parties

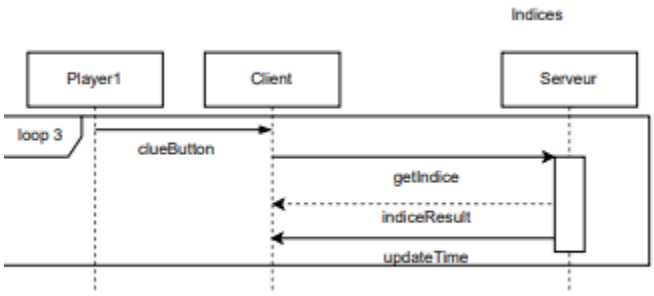

GameHistory
Player1
Client
Serveur
MongoDB
showGameHistory
getGameHistsory
getGameHistory
Response
Response
deleteGameHistory
deleteGameHistory
setGameHistory
Response
Response

e. Diagramme de séquence: modification des paramètres de jeu contre la montre par le panneau de configuration



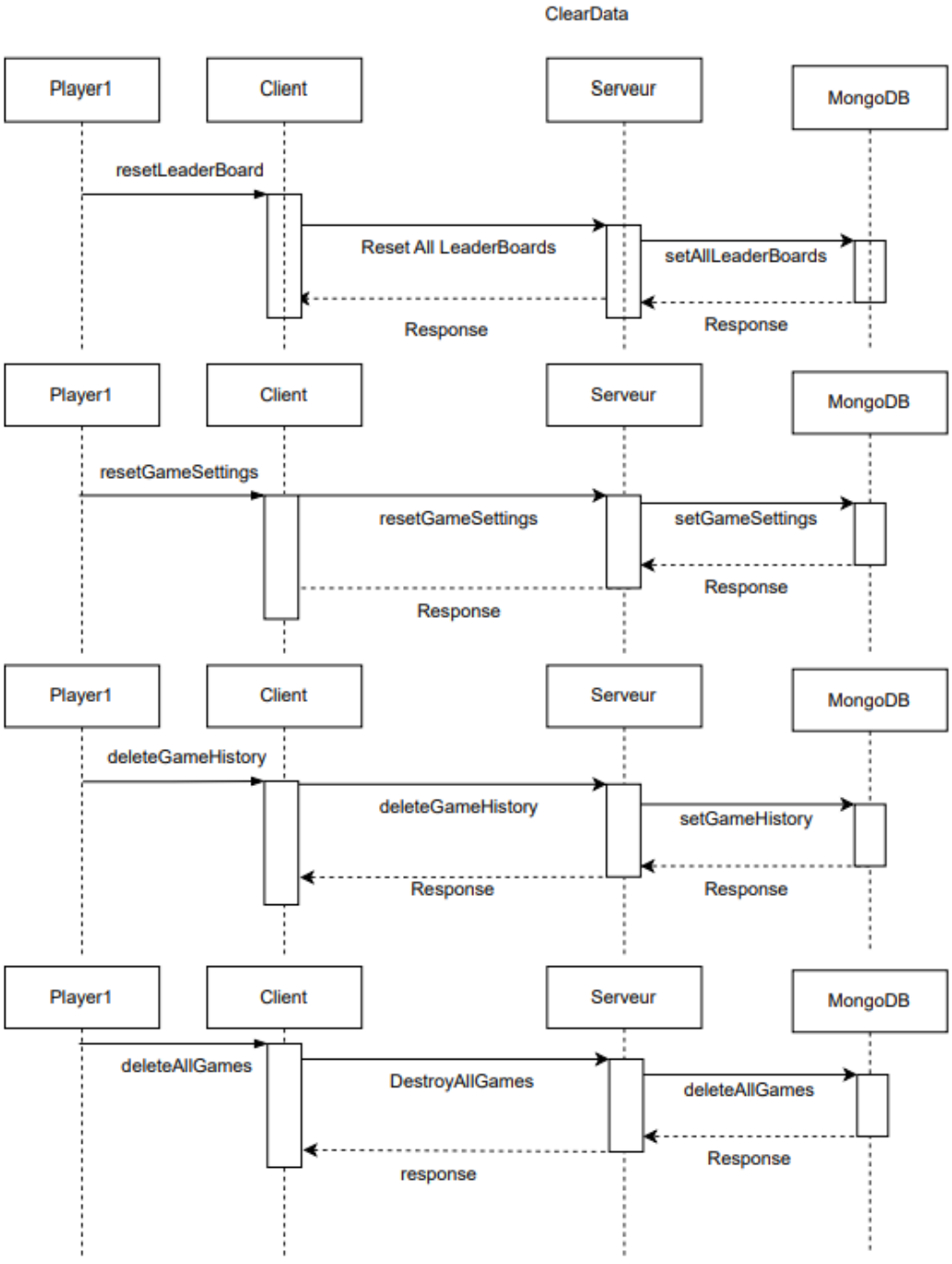
Note: Les gamesettings sont statiques et sont les mems pour tous les jeux contre la montre.

f. Diagramme de séquence: obtention d'indices



Note: Les deux premiers indices n'ont pas besoin d'accéder à la db car les images sont déjà sur le serveur. Pour le troisième indice, le serveur fetch quel type d'indice est nécessaire (flèche ou cercle). Ensuite, le serveur fetch de mongoDB l'image appropriée pour mettre en évidence la différence.

g. Diagramme de séquence: Remise a valeurs défault des paramètres de jeu contre la montre par le panneau de configuration

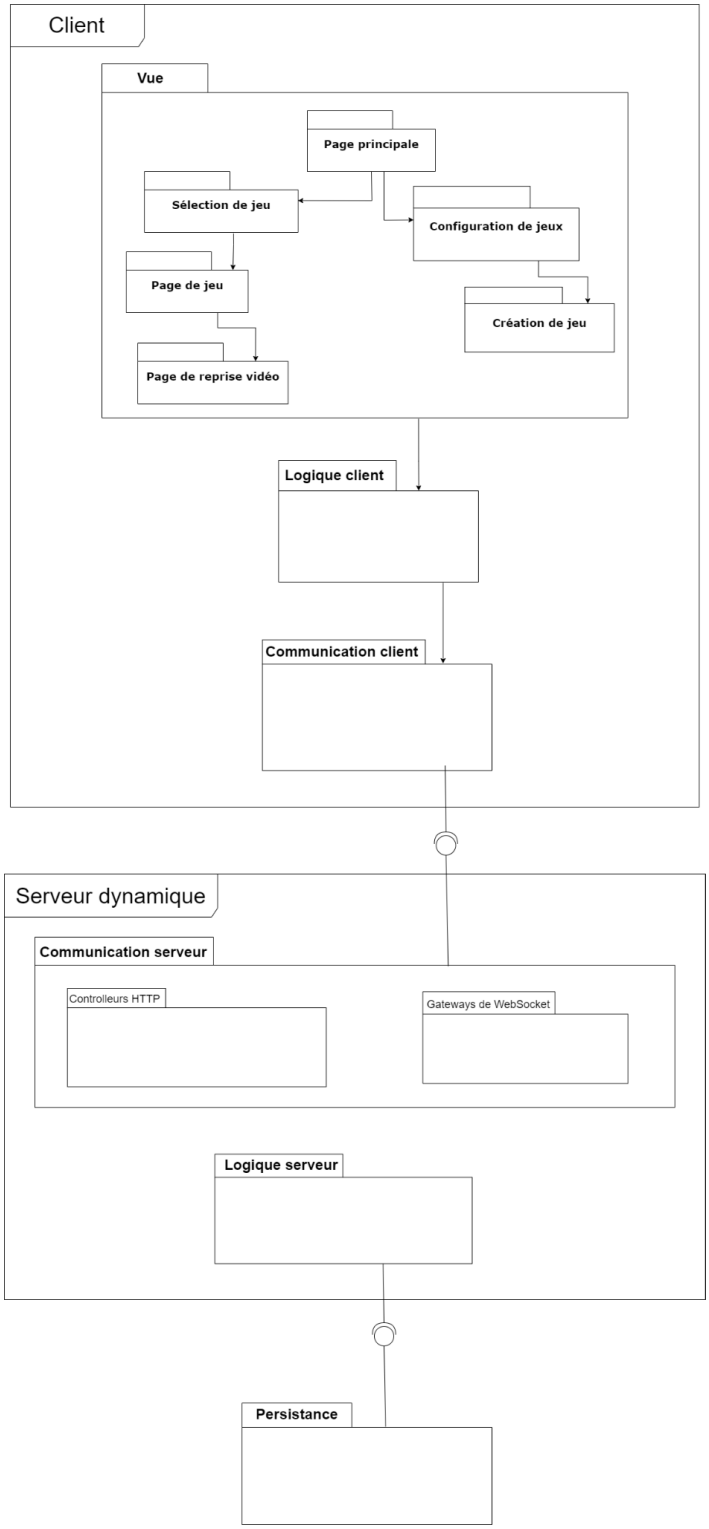


Note: Ces 4 requêtes ne sont pas forcément séquentielles, elles sont arrangées ici dans le même graphique par économie d'espace.

4. Vue logique

*Note: Nous utilisons le français pour les noms de paquetages, mais l'anglais quand il s'agit d'une classe spécifique de notre code puisqu'on programme en anglais (comme GameService, SessionGateway…).

a. Diagramme de paquetage général du projet:



b. Paquetage: Vue

Vue
Ce paquetage comprend l'ensemble du visuel de l'application web client. Il est composé de 6 types de pages, qui ont chacune plusieurs composantes Angular. Les pages de sélection de jeu et de la page de jeu offrent (ou offriront au sprint 3) des variations visuelles en fonction du mode de jeu, c'est-à-dire solo ou multijoueur, et mode classique ou temps limité. La page game-page et limited-time page utilise les services moue-service, in-game service, image opération service et socket client service. La page replay-page utilise les services in-game service, image-operation service et game action Login service. La page configuration utilise validate-image service et history-service. Game Creation utilizes mouse-position service, validate-image et draw service. Game selection et Main n'utilisent pas de services pour leur logique car elle est très simple.

c. Paquetage: Logique Client

Logique Client
Ce paquetage contient les services de logique du côté client, pour la gestion de jeux, le matchmaking. Cela inclut également la gestion d'un jeu en cours (donc une session), et la logique de création de jeu et de modification d'avant-plan d'images.

d. Paquetage: Communication client

Communication client
Le paquetage de communication client se charge de communiquer avec le serveur dynamique en utilisant 2 protocoles différents. Le premier est le protocole HTTP, à travers CommunicationService, et le deuxième et les WebScocket à travers SocketClientService. Ces 2 méthodes opèrent indépendamment.

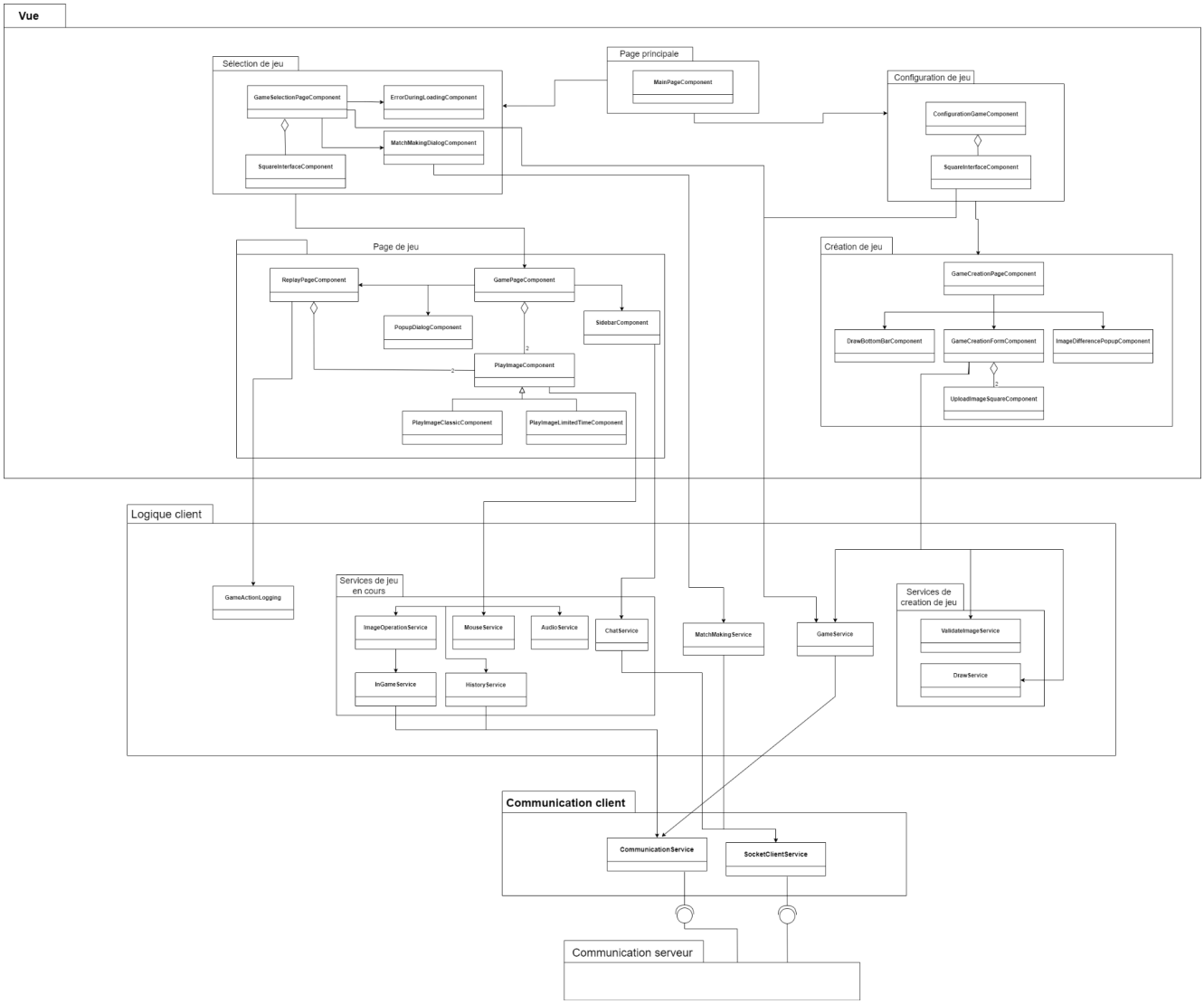


Diagramme de paquetage application client

e. Paquetage: Communication serveur

Communication serveur
Du côté du serveur, le paquetage de communication permet la réception de messages HTTP des clients à travers les controlleurs HTTP et les gateway WebSockets. Le sous-paquetage controlleurs HTTP permet la création et la recherche de jeux et d'images, donc les ressources du serveur dynamique. Le sous-paquetage de gateway WebSockets regroupe quant à lui la gestion d'énènements WebSocket, qui permettent le matchmaking des jeux multijoueurs, la gestion des sessions de jeu et les fonctionnalités de clavardage.

f. Paquetage: Logique serveur

Logique serveur
Le paquetage de logique serveur se charge de la gestion de toute la "business logic" du serveur dynamique. Ce paquetage reçoit et fait la validation de tout nouveau jeu, et analyse les différences des images envoyées avec le DifferenceDetectionService. Il gère la création et la suppression de jeux, et gèrera aussi la modification des paramètres et la réinitialisation des scores d'un jeu. Ces fonctionnalités sont gérées à partir du GameService en communiquant avec la persistance. Il s'occupe ensuite de traiter le progrès de toute session de jeu en cours dans SessionService, en validant les différences trouvées et en notifiant les joueurs des évènements comme la fin du jeu lorsque les différences sont trouvées ou le temps est écoulé dans le mode temps limité. Chaque session est traitée en parallèle dans un objet Session indépendant. La logique du serveur contient aussi une gestion simple des constantes de jeu en temps limité, avec la capacité de vérifier que les valeurs sont valides et de modifier les valeurs sur la database.

g. Paquetage: Persistance

Persistance
Enfin, la persistance correspond à la partie de l'architecture qui s'occupe de sauvegarder toutes les informations nécessaires pour jouer un jeu. Les informations des jeux tels leurs nom, les meilleurs scores, et les temps de pénalités par indice sont stockés dans une base de données MongoDB, gérée à travers un objet GameDocuments. Les images et les listes de différences par jeu sont sauvegardées en tant qu'images BMP et fichiers JSON respectivement, dans le répertoire de fichiers local.

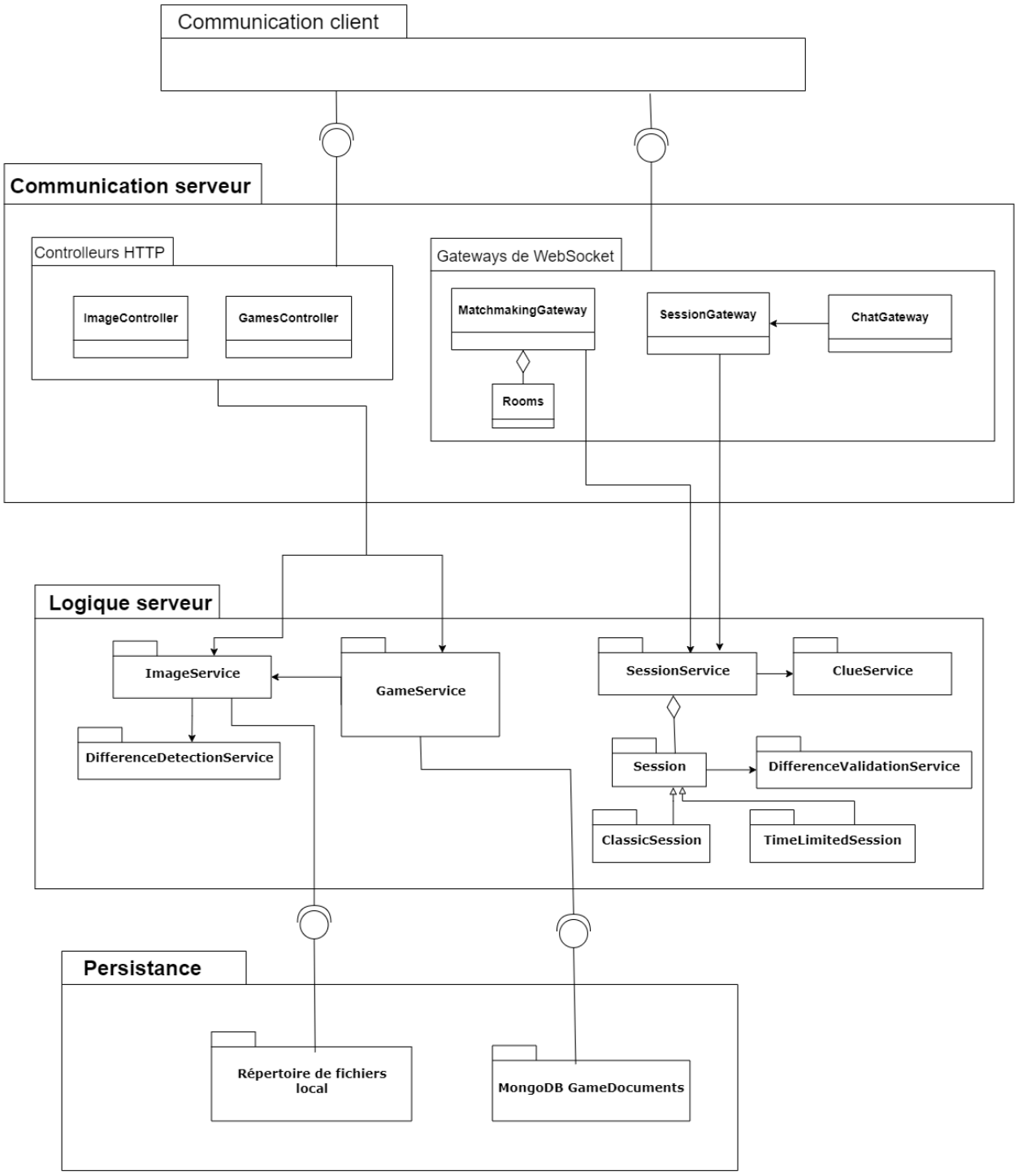


Diagramme de paquetage serveur dynamique

5. Vue de déploiement

a. Diagramme de déploiement