Conseil 7

Joël Collinet
Félix Erb
Célia Kocher
Vincent Moitry
Adam Rimelé
Carlo Spiga
Maxime Zimmer

2.B Dossier de conception matérielle et logicielle

Projet Acrobatt 2014

La Clémence d'Auguste

Tuteur : Pierre Kraemer

SOMMAIRE

Introduction

Dans le cadre du projet, il faut choisir l'architecture matérielle et logicielle qui sera utilisée par les développeurs et l'application.

Définir dès maintenant les outils et langages utilisés au niveau du serveur et du client permet une meilleure coordination de l'ensemble des éléments.

L'application repose sur un choix de langages, lui permettant d'exécuter un maximum de fonctionnalités pouvant s'inclure dans le cadre du projet. De bonnes habitudes de sécurité peuvent également être mises en place.

Les développeurs utilisent un ensemble de logiciels et applications communs, permettant de garantir une bonne compatibilité du projet quelque soit la plate-forme. D'autres outils externes peuvent également faciliter le suivi du développement.

I. Architecture logicielle

a) Architecture choisie

Le serveur de calculs est lié à une base de données afin d'en garantir la persistance. Ce même serveur communique avec les navigateurs et clients lourds. Il reçoit les données envoyées par les clients, effectue les traitements découlant de ces données, puis envoie les résultats aux clients (dont le plateau de jeu mis à jour après application des traitements).

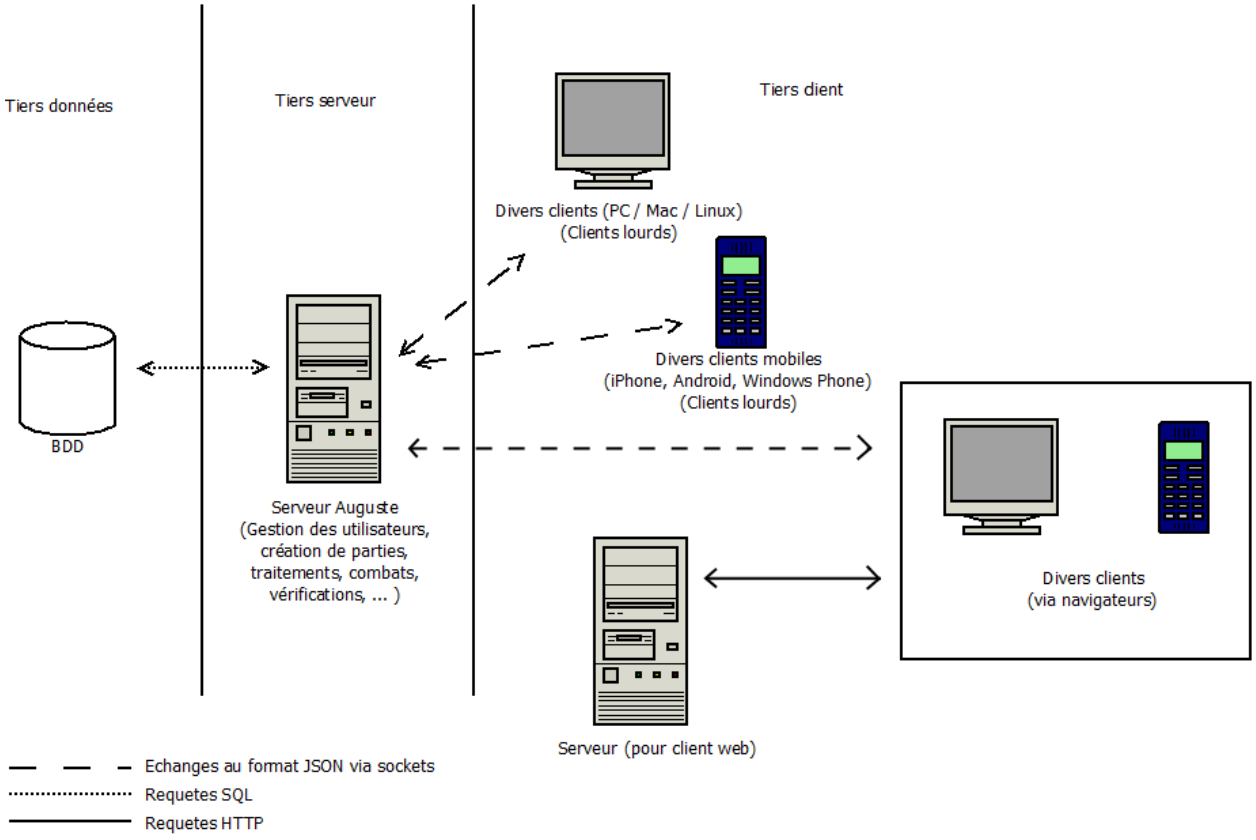
Les clients lourds permettent aux PC et smartphones de disposer d'une interface graphique embarquée. Toutefois, les navigateurs doivent recevoir cette interface depuis un second serveur physique, utilisé comme serveur web (il n'a aucun lien avec le serveur de calculs et a pour seule fonctionnalité d'envoyer l'interface web aux navigateurs).

L'ensemble des informations transmises entre le serveur de calculs et les clients est sérialisée avec l'aide du format JSON. Ce format permet de structurer facilement les données et permet leur identification rapide des deux côtés. Ces JSON sont transmis via des websockets.

Les clients s'identifient avec l'aide du couple d'identifiants nom / mot de passe. Le mot de passe est préalablement haché avant d'être stocké en base de données, afin de ne jamais transférer le mot de passe en clair.

Le protocole TCP est utilisé pour transférer les informations.

Aucune autre sécurité n'est prévue pour l'instant (hormis les vérifications au niveau du jeu en lui-même), l'application reste un simple jeu et ne contient pas d'informations sensibles.



Le serveur web est un serveur Apache fournissant la structure du site aux navigateurs (le langage PHP y est utilisé). La structure de l'application est incluse dans les clients lourds.

Ces clients interagissent avec le serveur de calculs. Les informations y sont échangées (gestion des utilisateurs, chat, création/gestion de partie).

Les données reçues par le serveur sont ensuite stockées dans une base de données, cela permet d'assurer leur persistance. Seul le serveur de calculs peut accéder directement cette base.

Enfin, la base est sauvegardée chaque nuit (l'endroit de stockage de la base est encore à définir, le serveur FTP et le disque dur externe sont envisagés).

b) Logiciels

Au niveau des langages, le Java a été choisi pour sa simple portabilité entre les différents systèmes d'exploitation au niveau logiciel. Concernant le serveur web, le PHP est utilisé (ce serveur n'ayant aucune interaction avec le serveur de calculs, il est possible d'utiliser ce langage pour le gérer sans problème de compatibilité). Côté navigateur, l'interface est structurée avec le HTML (pour le contenu) et le CSS (pour disposer le contenu), le JavaScript est également utilisé pour faciliter les traitements dynamiques et rendre le jeu un peu plus interactif pour le client.

Pour le développement, le projet est géré avec l'aide de Git, chaque membre de l'équipe peut ainsi accéder aux différentes versions du projet. NetBeans 7.4 est l'environnement de développement intégré utilisé par défaut.

Le projet en lui-même est suivi avec l'aide de l'outil en ligne Trello. Il permet un suivi des tâches efficace et intuitif (affecter un membre à une tâche, suivre l'évolution de cette tâche, ...).

Hors IUT, les membres de l'équipe communiquent à l'aide de Facebook (avec les outils de conversation et de groupe), éventuellement Skype.

Enfin, l'espace de stockage en ligne Google Drive permet d'héberger temporairement les différents documents, et permet une modification simple de ces documents par les divers membres de l'équipe (laisser des commentaires est également possible). Une version texte et une version pdf sont ensuite mises à disposition sur Github.

II. Architecture matérielle et réseau

a) Système

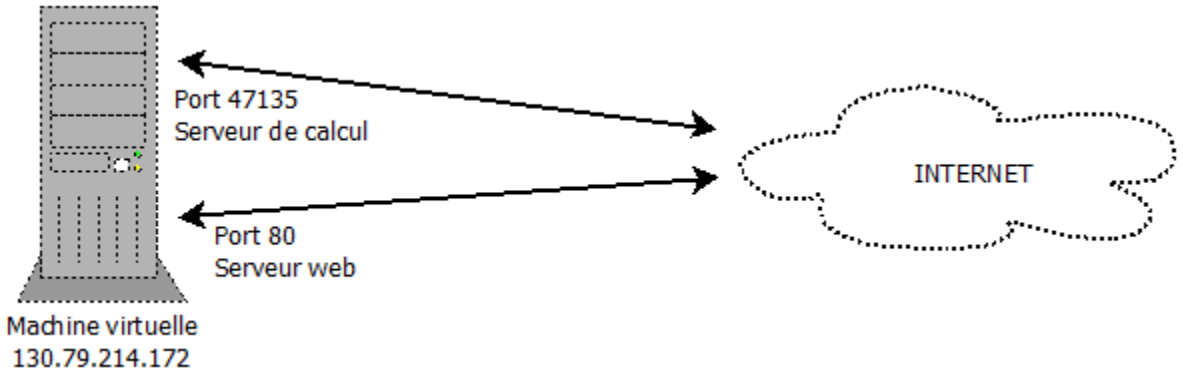
L'application est hébergée sur un serveur Ubuntu 12.04 contenant un serveur web Apache et un serveur de calculs Java accessible via web-sockets. 1 GB de RAM et 100 GB de stockage sont alloués.

Les développeurs travaillent sous divers systèmes d'exploitation (Windows, Mac OS, Linux). L'application étant majoritairement codée en Java, la compatibilité entre systèmes d'exploitation ne devrait pas poser de problèmes (ce qui permet de répondre au critère d'interopérabilité imposé par le cahier des charges).

b) Réseau

Le serveur de tests est accessible via l'IP 130.79.214.171, celui de production via l'IP 130.79.214.172, il est relié au réseau grâce à une connexion filaire. La partie web est joignable via le port 80, et le port de SocketServer est le 47135.

Ces serveurs sont utilisés pour déployer le prototype, puis l'application en production. Aucune autre configuration spécifique n'est mise en place.



c) Sécurité

Plusieurs mesures de sécurité sont mises en place afin de garantir la stabilité de l'ensemble de l'application. Ainsi, la base de données liée à l'application est périodiquement sauvegardée (pour pouvoir être restaurée à un état antérieur en cas de problème).

Cette même base de données n'est accessible que via le serveur de calculs (l'ensemble des applications lourdes et navigateurs doivent passer par ce serveur de calculs qui exécutera les traitements).

Chaque utilisateur doit s'inscrire sur le site et renseigner un mot de passe. Ce mot de passe est haché avant d'être stocké en base de données.

Le serveur de calculs effectue diverses séries de vérifications des coups joués lors d'un partie (le coup joué peut ne pas être autorisé, un deuxième coup peut vouloir être joué sur le même tour, … on peut ainsi garantir la bonne cohérence de la partie et éviter la triche). Ces vérifications sont également opérées côté navigateur (pour avertir le joueur en cas de coup non légal sans avoir à attendre la fin du tour, ce qui peut également être une aide de jeu).

Conclusion

La définition des normes logicielles et matérielles dès le début du projet permet à l'équipe d'analyser les besoins requis par l'application et de délimiter un cadre de travail efficace et durable. Il permet aussi de répartir plus facilement et plus efficacement les tâches entre les différents membres du groupe.

Devoir changer d'organisation matérielle ou logicielle en cours de développement est une tâche relativement pénible. Par exemple, choisir initialement un serveur d'applications qui ne permet finalement pas l'implémentation d'une fonctionnalité en particulier implique son changement pour un serveur différent. Celui-ci risque de ne pas traiter l'application de la même manière, impliquant parfois de drastiques changements dans la forme comme le fond. Le temps de mise en place de ces changements peut s'avérer assez long. De plus, constater ce genre de problème tardivement dans le projet se révèle bloquant dans la majorité des cas (il est trop tard pour changer de mode de fonctionnement sans prendre le risque de rendre inutile une grande partie de travail déjà effectuée et fonctionnelle).

Prévoir l'architecture logicielle et matérielle avant de débuter le projet permet de limiter au maximum le temps pouvant être perdu par un problème susceptible de survenir en plein développement pour cause d'architecture. Cela n'élimine pas totalement un tel risque, mais le limite considérablement.

Les bases du projet sont donc posées, et chaque membre de l'équipe en applique les principes pour s'assurer au mieux de la compatibilité de son travail avec celui des autres membres.

La communication entre client et serveur est également organisée. Il serait dangereux de commencer à développer une application multi-tiers sans planifier les moyens à déployer pour faire communiquer ces tiers.