



EPITECH  
INNOVATIVE  
PROJECTS

La Vie est un Jeu !

## Bilan Documentation

*Ce document présente l'architecture du projet « La Vie Est Un Jeu ».*



# Le Groupe

Lepage Barbara [lepage.barbara@gmail.com](mailto:lepage.barbara@gmail.com)  
Caradec Guillaume [guillaume.caradec@gmail.com](mailto:guillaume.caradec@gmail.com)  
Corsin Simon [simoncorsin@gmail.com](mailto:simoncorsin@gmail.com)  
Glorieux François [fra.glorieux@gmail.com](mailto:fra.glorieux@gmail.com)  
Klarman Nicolas [nickoas@gmail.com](mailto:nickoas@gmail.com)  
Lassagne David [david.lassagne@gmail.com](mailto:david.lassagne@gmail.com)  
Louvigny Guillaume [guillaume@louvigny.fr](mailto:guillaume@louvigny.fr)  
El-Outmani Youssef [youssef.eloutmani@gmail.com](mailto:youssef.eloutmani@gmail.com)  
Le-Cor Wilfried [wilfried.lecor@gmail.com](mailto:wilfried.lecor@gmail.com)  
Lenormand Frank [lenormf@gmail.com](mailto:lenormf@gmail.com)





# Résumé

« La Vie Est Un Jeu » est un projet sur trois ans dans le cadre des « Epitech Innovative Projects » mené par un groupe de dix étudiants.

Ce projet, sous forme d'un site web et d'applications mobiles, propose à ses utilisateurs de pimenter leur quotidien. Pour cela, on leur propose de manière ludique de se fixer des objectifs, les réaliser, les collectionner et enfin les partager.

C'est donc à la fois un jeu et un réseau social, destiné à tous les âges !

Dans ce document traitant de l'architecture de notre projet, nous nous efforcerons d'avoir une approche aussi globale que possible, abordant un certain nombre d'aspects qui sont les suivants :

- utilisateur, au travers duquel nous étudierons les différentes fonctionnalités qui seront mises à disposition dans notre application finale ;
- hardware, afin d'identifier les différentes composantes aussi bien internes (afin de pouvoir prendre en compte nos limites en termes de matériel) qu'externes (afin d'avoir une certaine visibilité sur les outils à même d'utiliser notre application future) ;
- données, afin d'avoir une première réflexion sur notre façon de structurer les informations que nous allons stocker au niveau de notre base de données ;
- logique, afin d'avoir d'ores et déjà une idée d'architecture logicielle qui structurera notre approche du projet en découpant en couches les différents niveaux applicatifs.

Chacune de ces facettes sera étudiée au travers de diagrammes UML qui seront à chaque fois introduits avant d'être présentés.

En dehors de ces différents points qui seront abordés au travers de diagrammes divers, nous tenterons d'identifier les différentes contraintes induites par notre projet afin d'anticiper les problématiques à venir, qu'elles soient de nature fonctionnelle ou non fonctionnelle, et nous nous efforcerons d'avoir un regard qualitatif sur notre produit final au travers d'un audit et de conjectures concernant diverses données chiffrées.



# Informations du document

Type du document	Bilan Documentation
Titre complet du document	Bilan Architecture du projet « La Vie Est Un Jeu »
Mots clés	« bilan », « architecture », « UML », « diagrammes », « conception », « bases de données », « lavieestunjeu »
Nombre de pages	37
Nom du groupe	La Vie Est Un Jeu
Responsable	Chef de groupe : Barbara Lepage
Membres du groupe	Voir page de garde
Auteurs	Membres du groupe, voir page de garde
Contact	lavieestunjeu@googlegroups.com
Révision actuelle	2.0
Site vitrine	<a href="http://eip.epitech.eu/2014/lavieestunjeu/">http://eip.epitech.eu/2014/lavieestunjeu/</a>
Site officiel	Non disponible



## Table des révisions

Version	Auteur	Commentaire	Date
0.1	Barbara Lepage	Rappel de l'EIP	11/07/2012
0.2	Guillaume Louvigny	Architecture, buts et contraintes (objectifs)	11/07/2012
0.3	Guillaume Louvigny	Architecture, buts et contraintes (contraintes fonctionnelles)	11/07/2012
0.4	Guillaume Louvigny	Architecture, buts et contraintes (contraintes non-fonctionnelles)	11/07/2012
0.5	David Lassagne	Cas d'utilisation général, Fonctions Standard Utilisateurs	11/07/2012
0.6	David Lassagne	Cas d'utilisations détaillés, Inscription, login	11/07/2012
0.7	David Lassagne	Vues processus	11/07/2012
0.8	Barbara Lepage	Vue déploiement	11/07/2012
0.9	David Lassagne	Vue données	11/07/2012
0.10	Nicolas Klarman	Tailles et performances	11/07/2012
0.11	Youssef El-Outmani	Qualité	11/07/2012
0.12	Barbara Lepage	Schéma d'introduction	11/07/2012
0.13	François Glorieux	Partie Composants, schéma et correction	12/07/2012
0.14	Guillaume Louvigny	Mise en page et correction mineures	12/07/2012
1.0	Barbara Lepage	Document final de rendu (AA1)	15/07/2012
1.1	Barbara Lepage	Ajout de légendes sur tous les schémas	20/12/2012
1.2	Barbara Lepage	Détails de la base de donnée	20/12/2012
1.3	Barbara Lepage	Technologies de la base de donnée	20/12/2012
1.4	Barbara Lepage	Technologie du projet : Ocsigen	20/12/2012
1.5	Barbara Lepage	Portabilité client/serveur du projet	20/12/2012
2.0	Barbara Lepage	Document final de rendu (DA1)	21/12/2012



# Table des matières

<b>I</b>	<b>Introduction et contexte</b>	<b>7</b>
I.1	Epitech, l'EIP et « La Vie Est Un Jeu » . . . . .	7
I.1.1	Epitech, une école d'informatique pas comme les autres . . . . .	7
I.1.2	L'EIP, élément clé de la réussite scolaire . . . . .	7
I.1.3	« La vie est un Jeu », bien plus qu'un simple EIP : une révolution ! . . . . .	7
I.2	Vocabulaire du projet . . . . .	8
I.2.1	Qu'est-ce qu'un « achievement » ? . . . . .	8
I.2.2	Définitions, Acronymes et Abréviations . . . . .	8
I.3	Pré-requis et informations sur le projet . . . . .	10
I.3.1	Destinataires du projet . . . . .	10
I.3.2	Utilisation du projet, rapport à l'existant . . . . .	10
I.3.3	Remise en contexte technologique . . . . .	11
I.3.4	But final du projet . . . . .	11
I.4	Références . . . . .	11
<b>II</b>	<b>Représentation de l'architecture globale</b>	<b>12</b>
<b>III</b>	<b>Architecture, buts et contraintes</b>	<b>14</b>
III.1	Objectifs spécifiques ayant un impact sur l'architecture . . . . .	14
III.2	Contraintes fonctionnelles . . . . .	14
III.3	Contraintes non fonctionnelles . . . . .	15
<b>IV</b>	<b>Des technologies nouvelles, originales, performantes</b>	<b>16</b>
IV.1	Node.js, une technologie montante, Ocsigen, puissant framework OCaml . . . . .	16
IV.1.1	Server . . . . .	16
IV.1.2	Client . . . . .	17
IV.2	Portabilité client-serveur . . . . .	18
IV.2.1	Serveur . . . . .	18
IV.2.2	Client . . . . .	18
<b>V</b>	<b>Vue globale du projet</b>	<b>20</b>
V.1	Cas d'utilisation principaux . . . . .	20
V.2	Cas d'utilisation détaillés . . . . .	22
<b>VI</b>	<b>Vue Processus</b>	<b>23</b>
<b>VII</b>	<b>Vue Déploiement</b>	<b>27</b>
<b>VIII</b>	<b>Couches applicatives et Composants principaux</b>	<b>29</b>



<b>IX</b>	<b>Base de données</b>	<b>31</b>
IX.1	Description des tests de premier niveau . . . . .	31
IX.2	Vue conceptuelle . . . . .	32
IX.3	Schéma de la base de données . . . . .	33
IX.4	Technologie et portabilité de la base de données . . . . .	34
<b>X</b>	<b>Taille et Performance</b>	<b>35</b>
<b>XI</b>	<b>Qualité</b>	<b>36</b>
<b>XII</b>	<b>Conclusion</b>	<b>37</b>



# Chapitre I

## Introduction et contexte

### I.1 Epitech, l'EIP et « La Vie Est Un Jeu »

#### I.1.1 Epitech, une école d'informatique pas comme les autres

Epitech est une école formant des experts en informatique. Sa pédagogie par projets implique directement les étudiants dans leur apprentissage et les rend plus à même de réagir et s'adapter aisément, par exemple aux évolutions technologiques qui auront lieu au cours de leur carrière.

#### I.1.2 L'EIP, élément clé de la réussite scolaire

Un Epitech Innovative Project ou EIP est l'élément clé du cursus Epitech. Il s'agit d'un projet de fin d'études regroupant un minimum de six étudiants autour d'un but commun. Ce projet est conduit sur une durée de trois ans, beaucoup plus importante que celles des projets réalisés lors des trois premières années d'études. De plus, l'EIP amène les étudiants à se confronter au monde de l'entreprise.

#### I.1.3 « La vie est un Jeu », bien plus qu'un simple EIP : une révolution !

Dans le cadre de notre EIP, nous avons décidé de réaliser un réseau social à but ludique basé sur les « listes de choses à faire avant de mourir » : il définit toutes les choses que son auteur désire faire de son existence, une sorte de mémo pour ne pas gâcher sa vie. Notre projet permettra à nos visiteurs de construire leurs propres listes et de faire valider leurs exploits tout en les partageant avec leurs réseaux sociaux. Ainsi, chaque action réalisée par un utilisateur (ajout d'une activité, succès ou échec) sera un fil de discussion dans lequel le visiteur et son réseau pourront discuter et partager différents types de médias (photos, vidéos, etc.). Ce fil de discussion se trouvera sur les flux d'informations propres à chacun. L'activité d'un utilisateur sera validée par son propre réseau et apparaîtra sous forme de « succès », comme dans un jeu vidéo. Le site s'étendra par la suite en proposant d'autres caractéristiques propres aux jeux vidéo.





## I.2 Vocabulaire du projet

### I.2.1 Qu'est-ce qu'un « achievement » ?

Le terme achievement (qui peut être traduit par succès ou réalisation en français) est, dans le cadre vidéoludique, un objectif défini à accomplir par le joueur, en dehors de l'objectif principal (c'est-à-dire, gagner ou finir le jeu).

Les achievements sont donc des récompenses honorifiques ajoutant du challenge pour le joueur.

Nous pourrions utiliser une traduction française du terme achievement mais nous pensons que achievement est le terme le plus populaire dans le monde vidéoludique.

Les achievements permettent au joueur de découvrir plus en profondeur le contenu du jeu et donc d'explorer de nouveaux horizons.

L'obtention est souvent un moment agréable pour le joueur.

Il ressent une certaine satisfaction et se sent récompensé pour un effort.

Il peut ensuite les partager avec ses amis afin de recueillir les honneurs ou défier ses amis de faire autant ou mieux, ce qui améliore grandement l'immersion au sein du jeu.

Nous pensons, qu'il est intéressant d'en faire une analogie avec la vie.

La vie est un jeu comme un autre et mérite d'avoir elle aussi ses achievements.

### I.2.2 Définitions, Acronymes et Abréviations

**Algorithme** Suite finie et non-ambiguë d'instructions permettant de donner la réponse à un problème.

**API** En français « Interface de Programmation », c'est une interface fournie par un programme informatique permettant l'interaction des programmes les uns avec les autres.

**Application mobile** Une application mobile est une application développée pour être installée sur un appareil électronique mobile.

**Architecture web** Une architecture web désigne la structure générale inhérente à un environnement web.

**Base de données** Une base de données est un lot d'informations stockées dans un dispositif informatique.

**Breaking news** En français « Dernières Nouvelles ».

**Bug tracker** En français « Logiciel de suivi de problèmes », c'est un logiciel permettant d'aider les utilisateurs et les développeurs à améliorer la qualité d'un logiciel en trouvant les failles de celui-ci.

**Cahier des charges** Le Cahier des charges vise à définir simplement les spécifications d'un produit ou d'un service à réaliser.

**Dépôt** Un dépôt est un stockage centralisé et organisé de données.

**Diagramme de Gantt** Un diagramme de Gantt est un outil utilisé en ordonnancement et gestion de projet et permettant de visualiser dans le temps les diverses tâches liées composant un projet.

**Diaporama** Un diaporama est une suite d'images ou de documents reliés par des effets et sur lesquels il est possible de mettre du son.



**Doodle** Doodle.com est un site web de planification et de sondage de la société suisse Doodle AG.

**GitHub** Github est un service web d'hébergement et de gestion de développement de logiciels, utilisant le programme Git.

**Google Docs** Google Docs est la suite des évolutions de Google Spreadsheets, logiciel de traitement de texte. Ces programmes fusionnés permettent un travail collaboratif en ligne.

**Google Talk** Google Talk est un logiciel propriétaire et service de messagerie instantanée et de voix sur IP basé sur Jabber et développé par la société Google.

**IRC** IRC est un protocole de communication textuelle sur Internet.

**JavaScript** JavaScript est un langage de programmation de scripts principalement utilisé pour les pages web interactives.

**Login** En français « Identifiant », c'est une information permettant à une personne de s'identifier auprès d'un système.

**Mailing list** En français « Liste de diffusion », c'est une utilisation spécifique du courrier électronique qui permet le publipostage d'informations aux utilisateurs qui y sont inscrits.

**Mise en production** Mise à disposition totale d'un service ou d'un produit.

**Ocaml** Anciennement connu sous le nom d'Objective Caml, c'est l'implémentation la plus avancée du langage de programmation Caml.

**Ocsigen** Framework de développement web, développé par le laboratoire français PPS.

**Réseau** Maillage de liens entre différents équipements informatiques permettant un partage d'informations.

**Réseau social** Ensemble d'identités sociales, telles que des individus ou encore des organisations, reliées entre elles par des liens créés lors d'interactions sociales.

**Service** Apporte une valeur ajoutée à un produit ou assure un travail nécessaire à une entreprise ou à un particulier.

**Site vitrine** Site internet composé de quelques pages présentant une société. Permet à une entreprise de communiquer avec le monde.

**Smartphone** Téléphone mobile disposant aussi des fonctions d'un assistant numérique personnel. Il fournit des fonctionnalités basiques comme l'agenda, le calendrier, la navigation sur le Web, la consultation du courrier électronique, la messagerie instantanée, le GPS, ...

**Android** Système d'exploitation utilisant le noyau Linux pour smartphones, PDA et terminaux mobiles conçu par Android, une startup rachetée par Google.

**iOS** Système d'exploitation mobile développé par Apple pour l'iPhone, l'iPod touch, et l'iPad. Il est dérivé de Mac OS X dont il partage les fondations.

**Windows Phone** Système d'exploitation mobile développé par Microsoft pour succéder à Windows Mobile, sa précédente plate-forme logicielle.

**Version Bêta** Version de test comportant toutes les fonctionnalités d'un programme. C'est grâce à cette version que les testeurs remontent les éventuels problèmes.

**Wiki** Espace collaboratif sur lequel les utilisateurs sont invités à rédiger des documents de travail.



## I.3 Pré-requis et informations sur le projet

### I.3.1 Destinataires du projet

Les utilisateurs ciblés par le projet sont très nombreux. Il suffit qu'une personne ait un centre d'intérêt ou une passion couvert(e) par le site pour qu'elle ait une raison de s'y inscrire.

Le site comme les applications mobiles seront conçus en vue d'une internationalisation étendue.

### I.3.2 Utilisation du projet, rapport à l'existant

Pour les utilisateurs finaux, le projet sera composé d'un site web et de plusieurs applications mobiles incluant Android, iOS et Windows Phone. Pour les développeurs tiers, une API sera mise en service, permettant la création de nouveaux usages.

Le projet a été pensé de façon à pouvoir répondre à un ensemble de fonctionnalités non présentes dans les projets déjà existants. Ces fonctionnalités ne sont pas combinées ou présentes sur les sites concurrents. En effet, nous tenons à mettre en place une plate-forme suffisamment ludique pour être visitée au quotidien, sans pour autant négliger l'aspect social permettant au visiteur de discuter de ses loisirs avec son réseau et de pouvoir partager aisément ses expériences à travers photos et vidéos.

À l'issue de la première année de travail sur le projet en partenariat avec l'INRIA et le Koalab Epitech, nous estimons avoir un site fonctionnel. Pendant la seconde année de travail sur l'EIP nous aimerions pouvoir signer des partenariats commerciaux avec divers acteurs de domaines culturels ou encore événementiels. Cette période sera également l'occasion d'ajouter de nouvelles fonctionnalités annexes répondant à des besoins émanant par exemple des utilisateurs.



### I.3.3 Remise en contexte technologique

Aujourd'hui, une part de plus en plus importante d'utilisateurs d'Internet sont équipés de smartphones, il est même prévu que le nombre d'utilisateurs en mobilité dépasse celui des utilisateurs sédentaires d'ici quelques années.

Pour s'adapter à cette évolution et faire face à l'expansion notable de nouvelles technologies, nous avons choisi de proposer à nos utilisateurs des versions dédiées aux plates-formes mobiles majeures.

### I.3.4 But final du projet

Le projet a pour but de créer une communauté d'utilisateurs autour d'un système d'achieve-ments, directement lié à la vie quotidienne, aux passions ou à la vie professionnelle.

Les utilisateurs ciblés sont très nombreux. En théorie, toute personne ayant un centre d'intérêt ou une passion est une cible.

À plus long terme, des partenariats commerciaux permettront de cibler des marques et des lieux.

Le site sera multilingue, donc ouvert à l'internationalisation.

## I.4 Références

Ce document présente l'architecture du projet « La Vie Est Un Jeu ». Pour mieux le comprendre, il est fortement recommandé de prendre connaissance des documents suivants :

**Étude de l'existant** Ce document présente les projets existants ressemblant au notre. En étudiant ainsi le marché, nous nous assurons d'éviter les erreurs de nos concurrents et d'en faire la solution la meilleure.

**Cahier des charges** Ce document présente dans les détails les fonctionnalités qui seront présentes dans notre projet, les plates-formes sur lesquelles il sera disponible, la portabilité serveur/client, les technologies utilisées, les raisons du choix de celles-ci, une introduction à la base de donnée et enfin l'organisation générale du projet avec planning et outils.

**Diagramme de Gantt** Ce diagramme détaille les phases d'avancement du projet, les tâches à réaliser, la durée pour les réaliser, les dates de début et les dates limites de celle-ci et enfin les responsables de chacune d'entre elles.

Ces documents peuvent être téléchargés sur notre site vitrine :

<http://eip.epitech.eu/2014/lavieestunjeu/>





## Chapitre II

# Représentation de l'architecture globale

Dans ce document traitant de l'architecture de notre projet, nous nous efforcerons d'avoir une approche aussi globale que possible, abordant un certain nombre d'aspects qui sont les suivants :

- utilisateur, au travers duquel nous étudierons les différentes fonctionnalités qui seront mises à disposition dans notre application finale ;
- hardware, afin d'identifier les différentes composantes aussi bien internes (afin de pouvoir prendre en compte nos limites en termes de matériel) qu'externes (afin d'avoir une certaine visibilité sur les outils à même d'utiliser notre application future) ;
- données, afin d'avoir une première réflexion sur notre façon de structurer les informations que nous allons stocker au niveau de notre base de données ;
- logique, afin d'avoir d'ores et déjà une idée d'architecture logicielle qui structurera notre approche du projet en découpant en couches les différents niveaux applicatifs.

Chacune de ces facettes sera étudiée au travers de diagrammes UML qui seront à chaque fois introduits avant d'être présentés.

En dehors de ces différents points qui seront abordés au travers de diagrammes divers, nous tenterons d'identifier les différentes contraintes induites par notre projet afin d'anticiper les problématiques à venir, qu'elles soient de nature fonctionnelle ou non fonctionnelle, et nous nous efforcerons d'avoir un regard qualitatif sur notre produit final au travers d'un audit et de conjectures concernant diverses données chiffrées.

Le schéma de la page suivant présente l'architecture globale du projet.





## Chapitre III

# Architecture, buts et contraintes

### III.1 Objectifs spécifiques ayant un impact sur l'architecture

Nos objectifs, qui incluent la création d'un service web et d'un ensemble d'applications mobiles, ont un certains nombres de contraintes. Celles-ci peuvent impacter l'architecture du projet : comme l'organisation même du développement, de la communication ou encore le code, et ce qu'elles soient inhérentes aux technologies retenues ou aux facteurs géographiques imposés par l'école.

### III.2 Contraintes fonctionnelles

Plusieurs contraintes fonctionnelles, provenant directement de l'organisation même du projet sont à noter. Il s'agit principalement du départ à l'étranger de la moitié de l'équipe, qui sera alors répartie sur un ensemble très large de fuseaux horaires : États-Unis Pacifique à Chine. Des outils de communication adaptés sont d'ors et déjà mis en place tels qu'une liste de diffusion (indépendante de celle fournie par le Lab EIP pour des raisons de goûts personnels), un bug tracker (fourni par Github) ou encore un canal IRC sur les serveurs de l'école (#eip-life). Ces ressources permettent de palier à l'absence potentielle de communication de vive voix entre les différentes composantes de l'équipe. Cependant des réunions ponctuelles planifiées à intervalles réguliers auront toujours lieu via visio conférence pour que chacun puisse faire part de sa progression dans les diverses tâches qui lui auront été assignées.

Une autre contrainte fonctionnelle que nous avons identifié est quant à elle matérielle. Il s'agit de la disponibilité des divers équipements proposés par le Lab EIP. Nous auront en effet besoin de plusieurs smart-phones pour réaliser les versions mobiles de notre projet. Si, l'équipe présente à Paris pourra faire usage de ces appareils les membres étudiant dans les universités partenaires d'EPITECH ne pourront pas en profiter. Il faudra soit qu'ils passent par leurs équipements personnels (ne couvrant généralement qu'une infime partie des modèles de téléphone d'un système d'exploitation donné), ou alors les émulateurs (Android) ou simulateurs (iOS, Windows Phone) ne reproduisant pas de façon optimale les conditions réelles d'utilisation des applications.

Notre projet étant un service web des serveurs seront nécessaire pendant toute la réalisation de celui-ci que ce soit les phases de formation, de développement ou encore de déploiement.



Dans un premier temps seul un serveur de capacité réduite est nécessaire, puisque nous n'auront pas à faire face à un trafic important. Cependant par la suite il sera nécessaire d'acquérir des machines plus performantes dans le cadre du déploiement du projet. Cette phase devrait arriver assez tôt, approximativement vers l'automne 2013 si l'on se fit au calendrier convenu avec le Lab EIP.

La dernière contrainte fonctionnelle que nous avons mis en lumière est notre incapacité actuelle à réaliser des tâches nécessitant des compétences plus larges que le simple développement : comme la conception d'interfaces utilisateurs ergonomiques, la réalisation d'une charte graphique propre au site etc. Pour répondre à ces besoins nous aurons le choix de nous former ou alors de faire appel à des personnes compétentes, extérieures à l'école.

### III.3 Contraintes non fonctionnelles

Une certaine émulation aura lieu dans le domaine de notre projet : quelques acteurs ont des projets relatifs au nôtre et seront potentiellement amenés à empiéter sur notre offre. Il faudra alors être amené à réagir promptement et convenablement à ces concurrents, tout en veillant à ne pas laisser un seul acteur avoir une position hégémonique sur ce secteur.

Ocsigen, le framework web en Ocaml conçu par les équipes de l'INRIA est puissant et performant. Cependant, le fonctionnement du langage, directement issu des principes de la programmation fonctionnelle énoncés par Church dans les années 1930, tranche beaucoup trop de la programmation impérative pour permettre une adaptation rapide et aisée à celui-ci. La formation au langage et, à plus forte mesure au framework, prendra alors une part importante dans la réalisation du projet.

La technologie que nous employons, Ocsigen, permet une telle réactivité sur le client web, cependant les processus de validations d'applications sur iOS ou Windows Phone restent un frein à cette même réactivité.

Toujours sur le plan des applications mobile, il ne faut pas perdre de vue que les mêmes fonctionnalités sont à coder quatre fois pour cibler toutes les plates-formes. En effet, il n'est pas rare que des entreprises comme Facebook ou Twitter proposent à plusieurs jours d'intervalle entre les divers systèmes mobiles des améliorations, ou alors que des fonctionnalités soient manquantes sur certains OS.





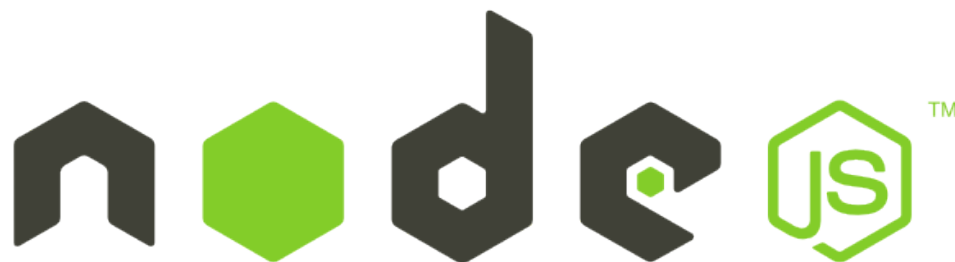
## Chapitre IV

# Des technologies nouvelles, originales, performantes

### IV.1 Node.js, une technologie montante, Ocsigen, puissant framework OCaml

#### IV.1.1 Server

Côté server, nous utiliseront le framework s'intitulant « Node.js ». Technologie en vogue codée en JavaScript et encore en phase de maturation, Node est une solution de plus en plus adoptée par les sites webs à la recherche de performance concernant les accès concurrentiels.



La capacité de ce framework à faire fonctionner des applications hautement scalables et à gérer, en théorie, des dizaines de milliers de connections simultanées sur une unique machine physique, en fait un choix non négligeable dans le cadre du développement d'une application à visée communautaire.



Les avantages de cette technologie assez récente par rapport à d'autres bien connues telles que RoR, Django, J2EE ou encore plus basiquement PHP sont multiples :

- Tout d'abord elle a l'avantage de corriger les faiblesses des technologies plus matures, étant mise au point par des personnes ayant déjà de l'expérience dans l'utilisation des dites technologies ;
- Son déploiement est particulièrement simple et rapide à effectuer ;
- Elle possède une communauté grandissante qui rassure sur le choix de cette technologie plutôt qu'une autre ;
- Elle est parfaitement adaptée à l'utilisation orientée web-service que nous voulons lui faire prendre ;

Toujours relativement jeune, l'évolution de cette technologie nécessitera d'être suivie et impliquera éventuellement certains travaux de maintenance.

Pour en savoir plus :  
<http://nodejs.org/>

#### IV.1.2 Client

Côté client, la technologie utilisée s'intitule « Ocsigen », puissant framework entièrement conçu en OCaml.



La programmation fonctionnelle est complètement adaptée au domaine du web comme le décrivent de nombreux articles sur Internet, tels que celui-ci :

<http://www-lipn.univ-paris13.fr/~loddo/funding/projet-hyper-learning.pdf>

Les avantages et les contraintes de la programmation fonctionnelle s'appliquent aussi dans la programmation avec Ocsigen.

Son typage fort résout de nombreuses problématiques de sécurité, comme celles que l'on peut rencontrer en PHP par exemple.

Ainsi, si l'on modifie une partie du programme et que cela a un impact sur une autre fonctionnalité, alors le programme ne compilera tout simplement pas tant que tout ne sera pas



correct.

OCaml facilitant la manipulation d'arbres, le code généré en HTML est fait à partir d'un AST (Abstract Syntax Tree) forcément valide.

Du fait de sa récente mise à disposition, il a été pensé et conçu pour le HTML5, le JavaScript et les dernières technologies côté client du web. Lorsque l'on programme avec Ocsigen, on réalise un véritable programme complet, compilé et lancé.

## IV.2 Portabilité client-serveur

### IV.2.1 Serveur

Bien que toujours considérée comme jeune, Node.js a su attirer son public et se trouve aujourd'hui disponible aussi bien sur Microsoft Windows, Apple Mac OS X ou sur les distributions Linux.

Notre projet se trouve ainsi parfaitement portable sur les plates-formes aujourd'hui les plus utilisées.

### IV.2.2 Client

Le choix de l'utilisation d'Ocsigen côté client est adaptée aux trois cas d'utilisation que représentent l'accès *via* un poste fixe, un smartphone ou encore une tablette. Dans le cadre de la version web mobile, Ocsigen gère de plus de nombreuses fonctionnalités spécifiques aux terminaux mobiles :

- tactile (toucher simple, toucher glissé, ...);
- géolocalisation ;
- orientation ;
- appareil photo ;

Ainsi, notre service est garanti d'être utilisable sur trois types de plates-formes.

- Navigateurs internet classiques
  - Google Chrome
  - Mozilla Firefox
  - Internet Explorer
  - Apple Safari
- Terminaux mobiles format téléphone



- Android
- iOS
- Windows Mobile
  
- Tablettes
  - iPad
  - Tablette Android

Pour chacune de ces trois plates-formes, nous aurons une interface différente adaptée aux différentes caractéristiques de l'outil.

Pour chacun des types de périphériques, nous aurons un programme différent, codé dans le langage propre à celui-ci, faisant appel à notre service Node.js *via* l'API développeurs.

Au même titre que la partie server, « La Vie Est Un Jeu » sera une application parfaitement portable.



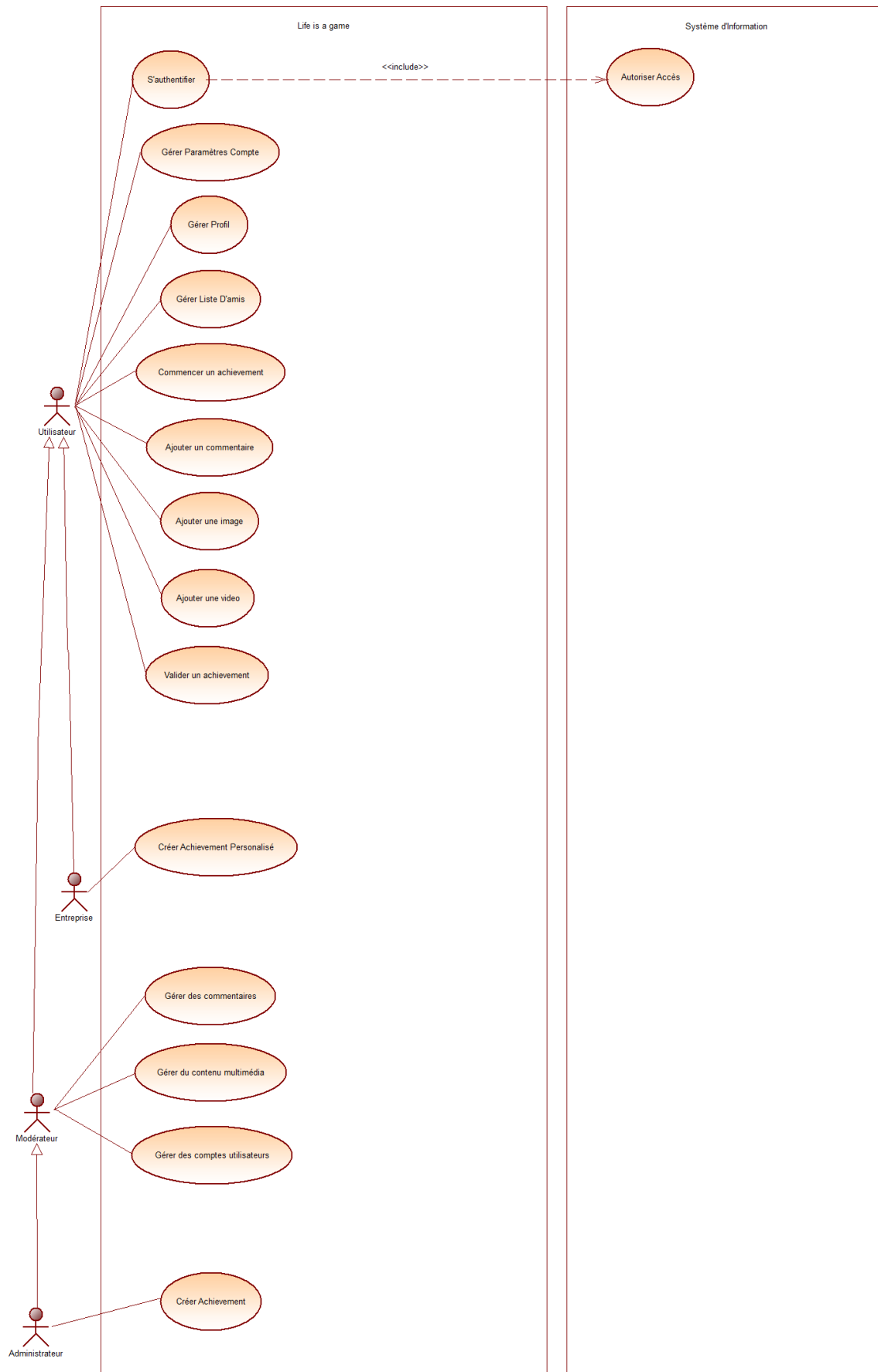


# Chapitre V

## Vue globale du projet

### V.1 Cas d'utilisation principaux

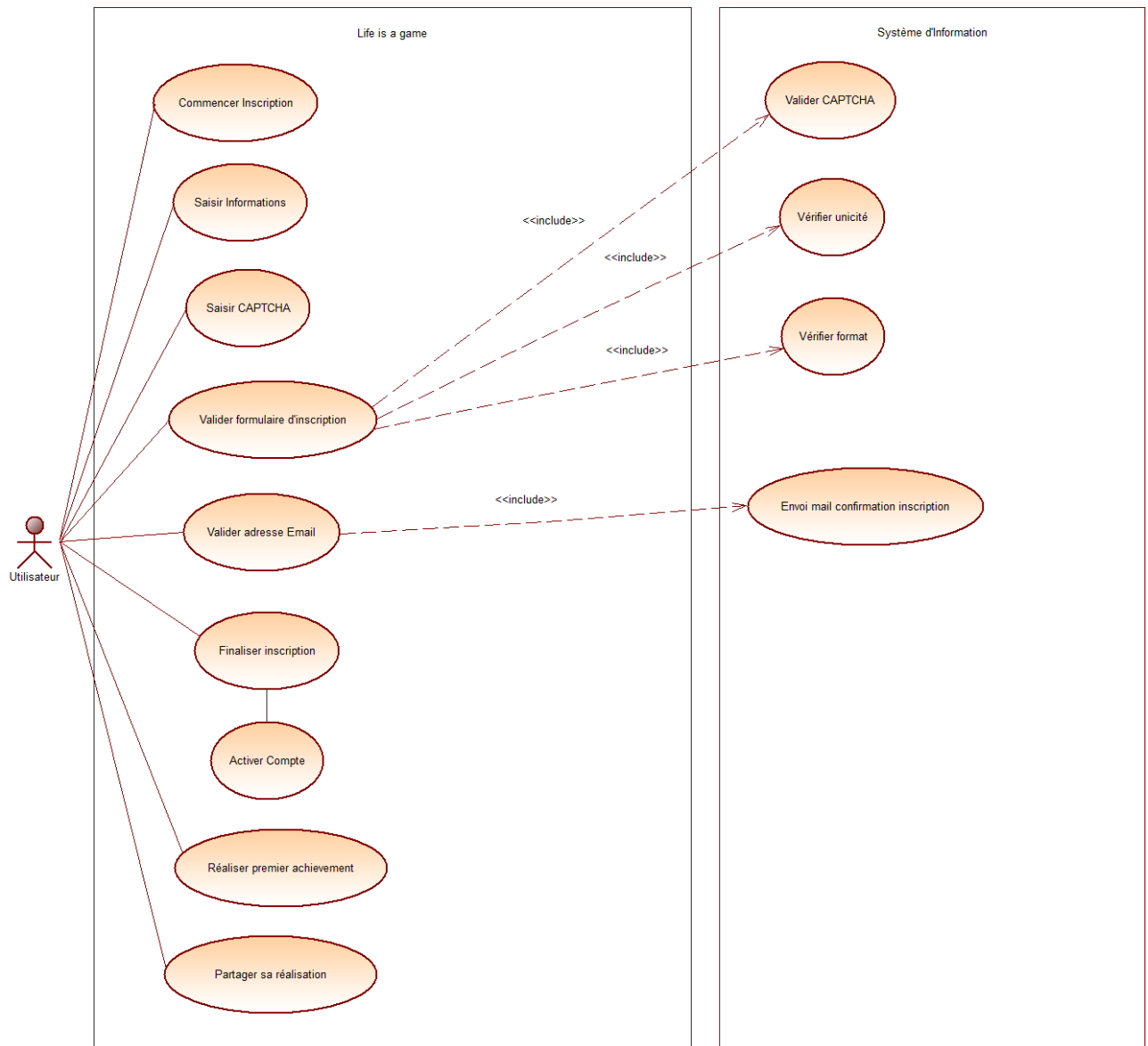
Le schéma suivant présente les principaux cas d'utilisation du projet. Les acteurs retenus sont les administrateurs, les modérateurs, les entreprises ainsi que les utilisateurs classiques.





## V.2 Cas d'utilisation détaillés

Le schéma qui suit expose les cas d'utilisation détaillés du projets, et ce pour un utilisateur dans le cadre de son inscription au service.





## Chapitre VI

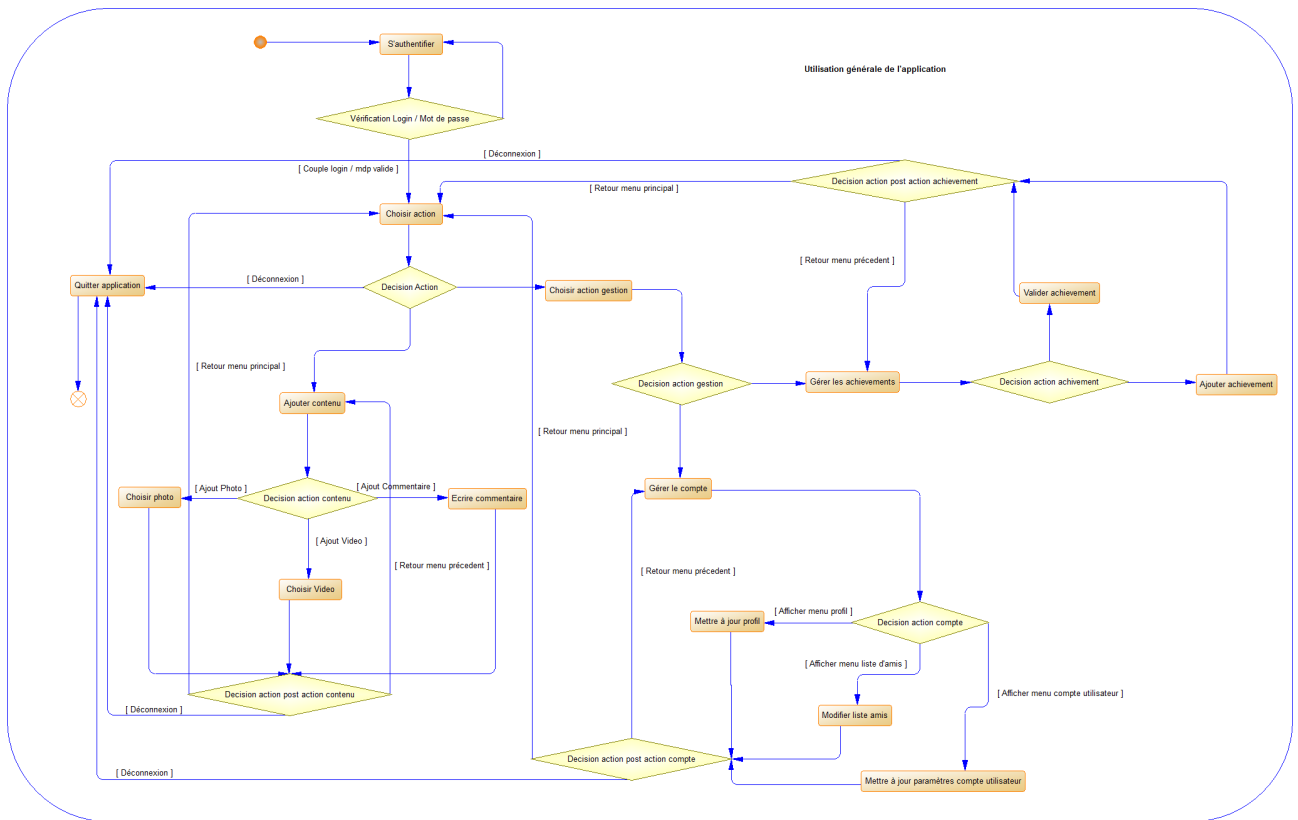
### Vue Processus

La série de schémas suivante expose diverses utilisations du projet, celle d'un utilisateur « classique » ayant un usage général de l'application, les fonctionnalités supplémentaires offertes à un modérateur, à une entreprise et aux modérateurs. Enfin l'inscription au service sera également détaillée.



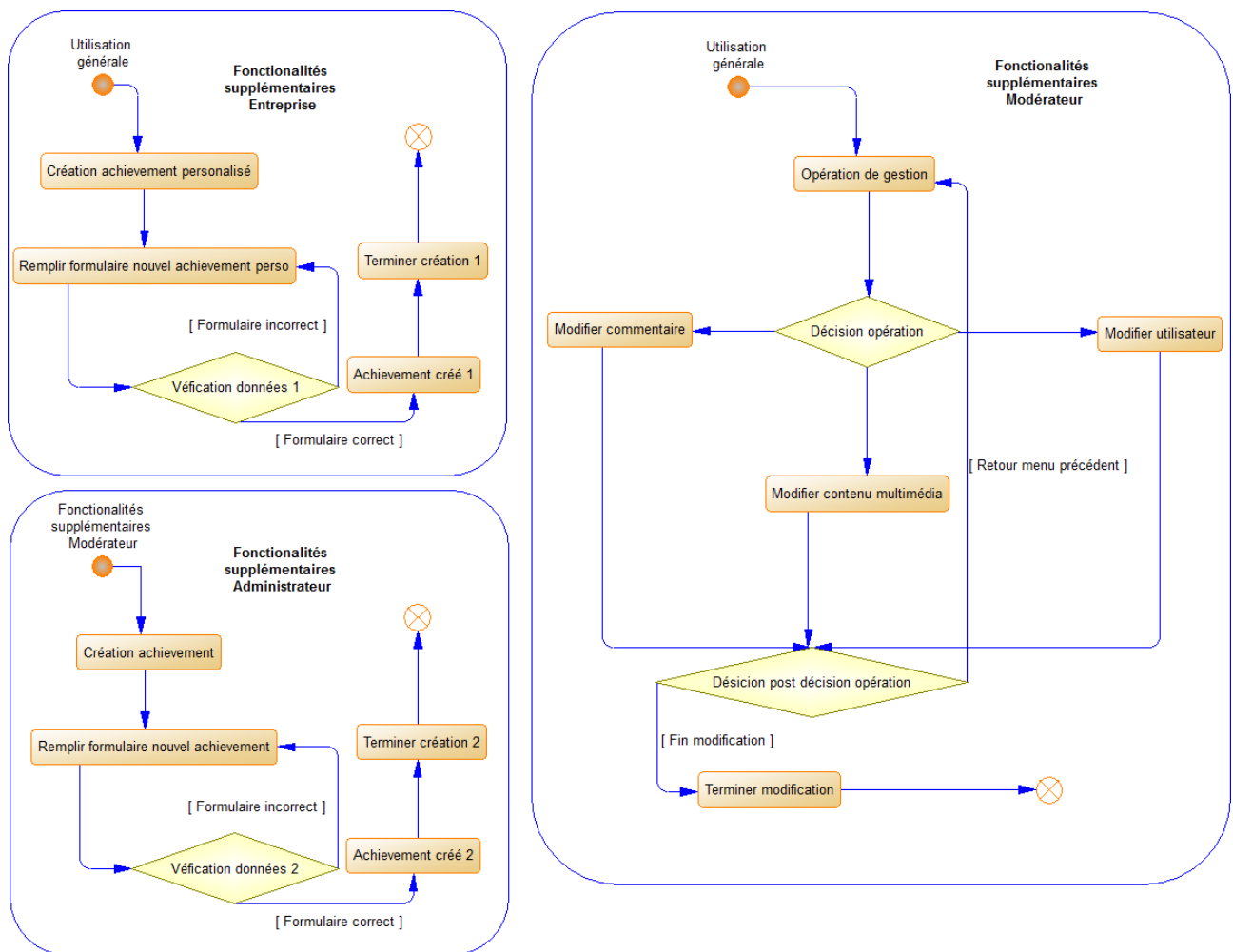


Le schéma suivant le cas d'utilisation du projet pour un utilisateur classique :



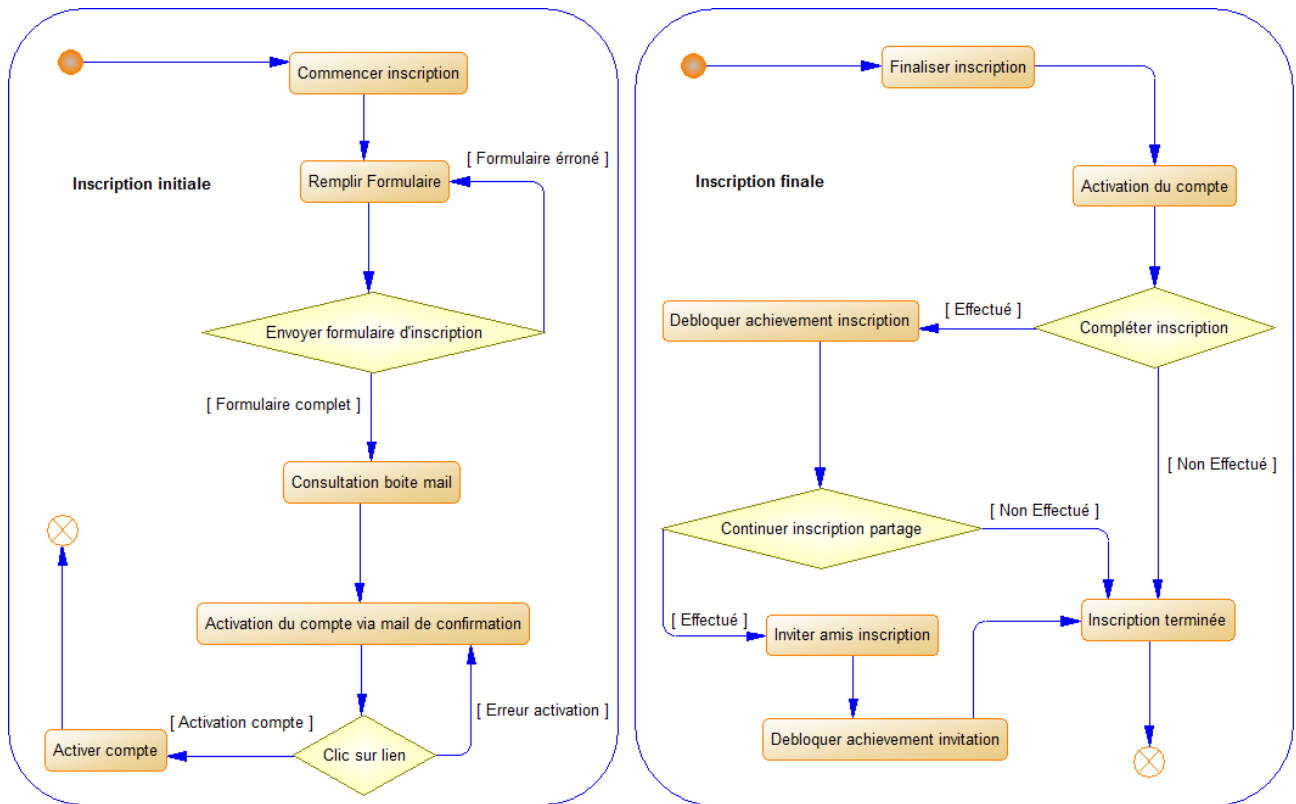


Le schéma suivant le cas d'utilisation du projet pour un modérateur ou une entreprise :





Le schéma suivant le cas d'utilisation du projet pour l'inscription au service :



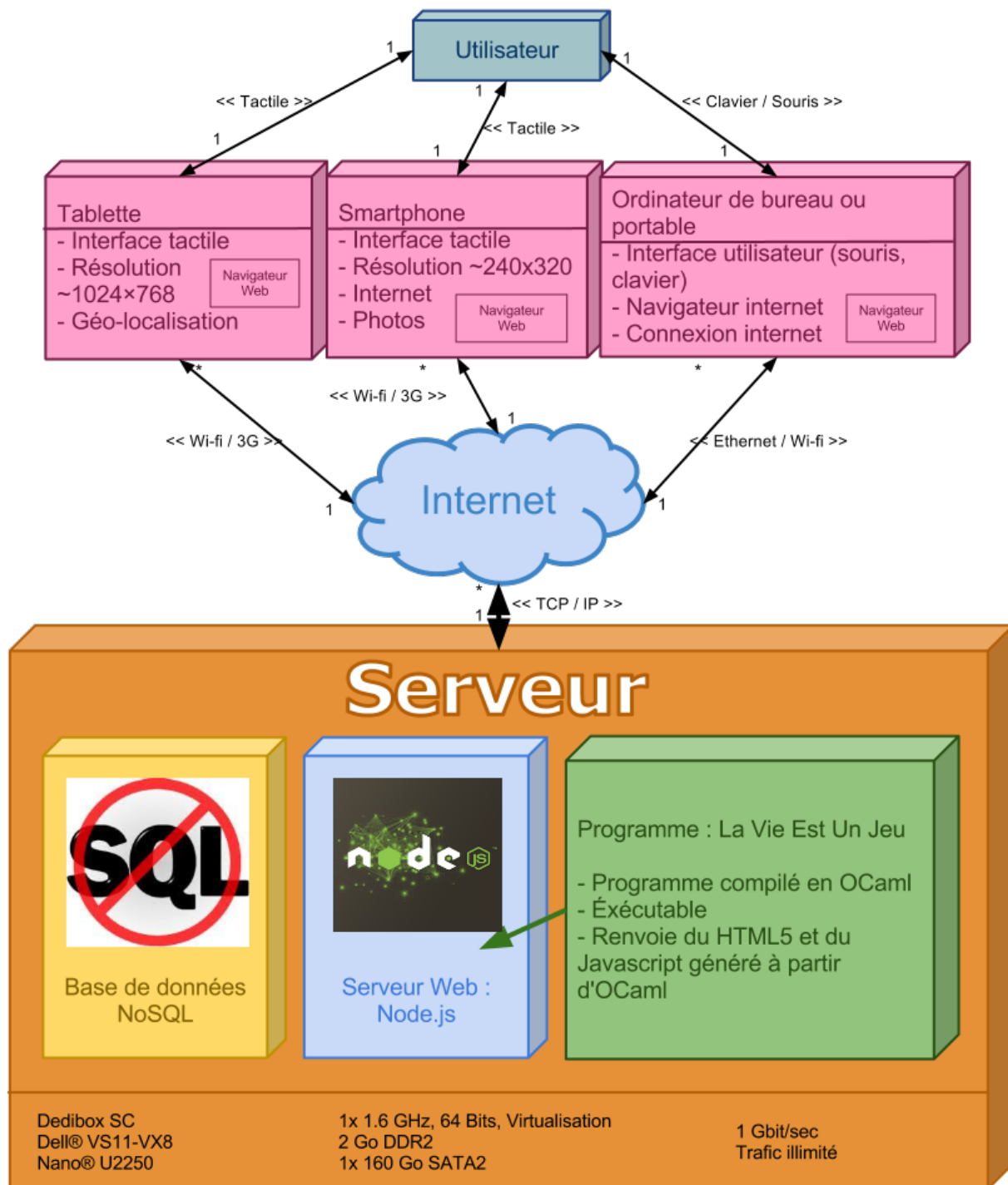


## Chapitre VII

# Vue Déploiement

Le schéma présent sur la page suivante détaille le déploiement de l'application sur le serveur de développement, les différents services utilisés : bases de données, serveurs.



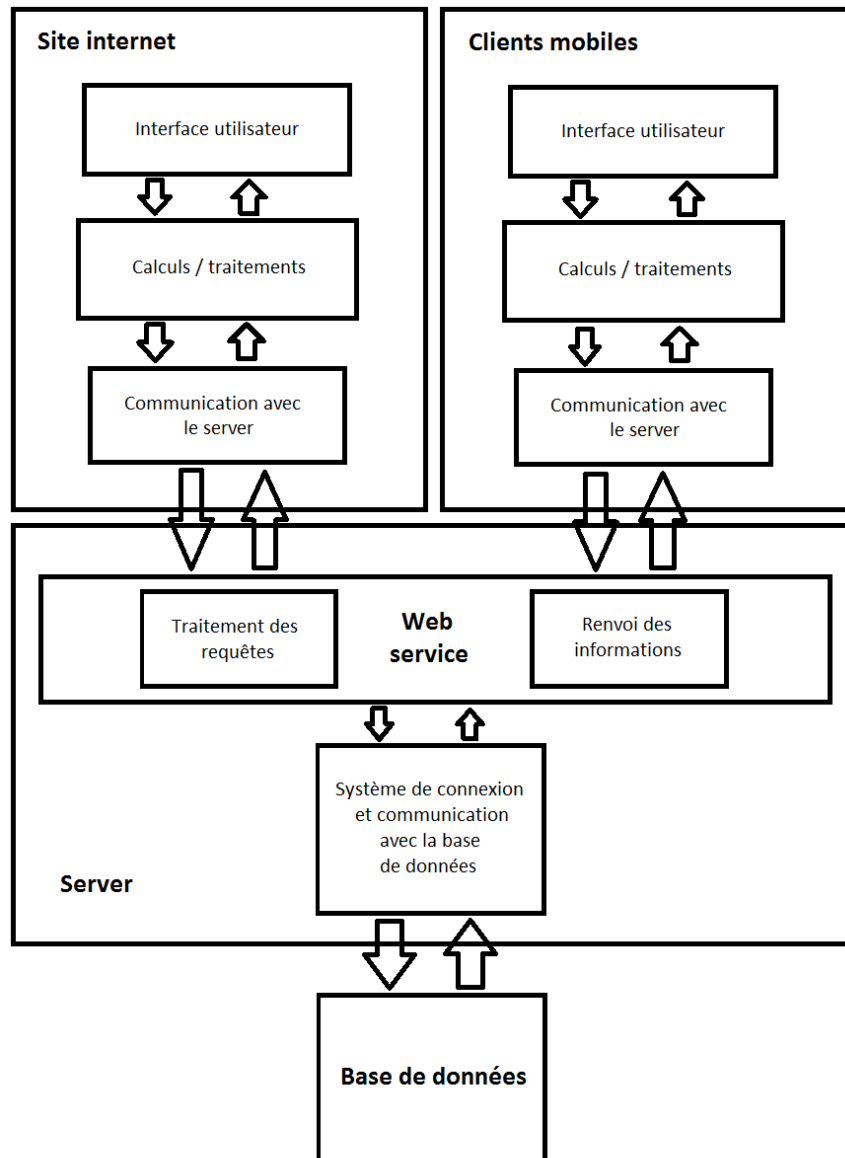




## Chapitre VIII

# Couches applicatives et Composants principaux

Le schéma suivant présente les couches applicatives :



Le projet disposera de plusieurs composants. La plus accessible, permettant la plus grande diversité au niveau du matériel dont dispose le client sera une application web (portable pour les mobiles). Plusieurs applications mobiles en dériveront pour les plates-formes Android, iOS et Windows Phone.

Les données utilisateurs seront stockées dans une base de données.

Un serveur Node.js disposera d'un accès direct à cette base et transmettra les données aux différents clients.



# Chapitre IX

## Base de données

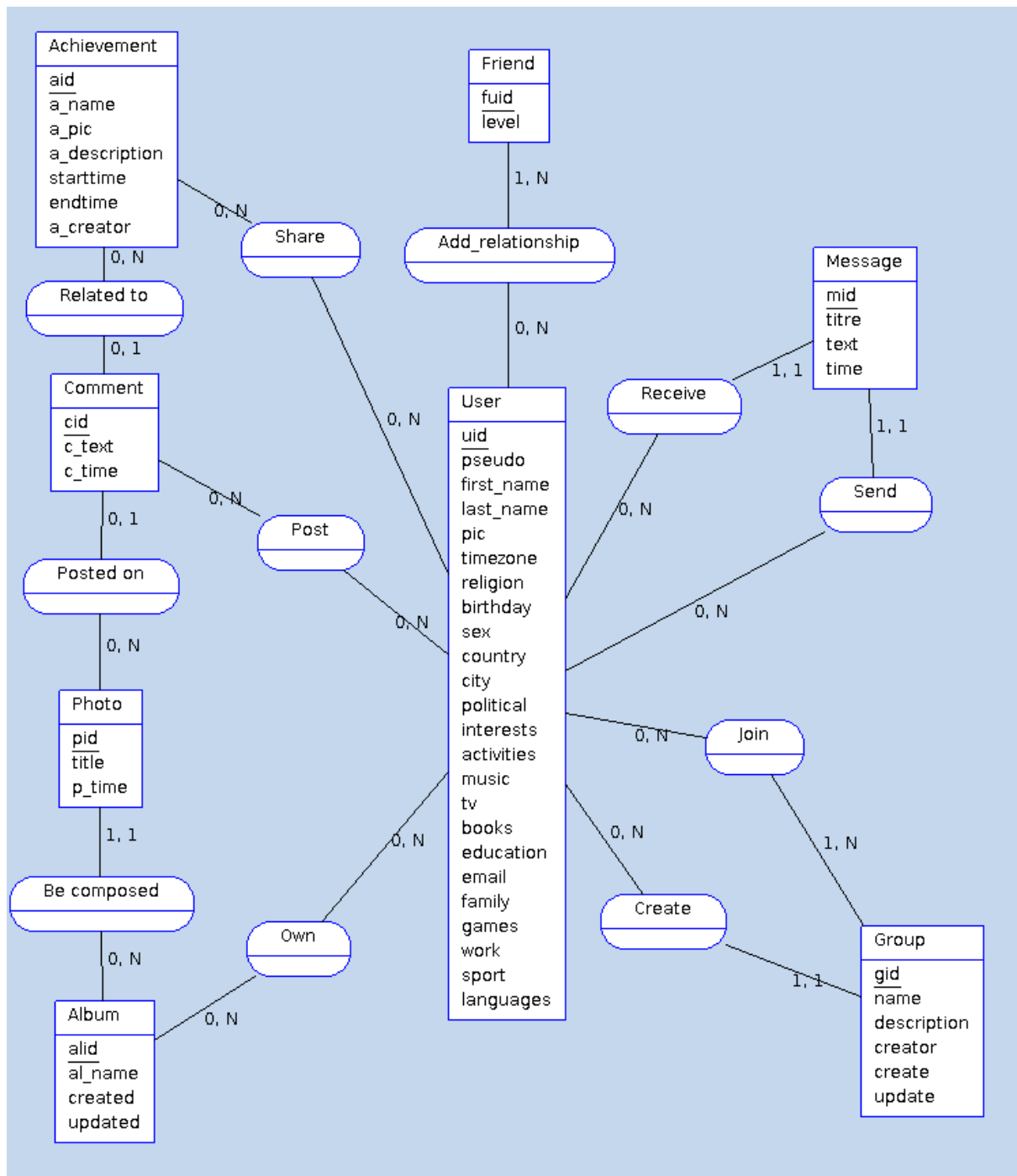
### IX.1 Description des tests de premier niveau

- Création de compte.
- Restriction d'accès par cercles.
- Listing des achievements déjà présents dans la base de données.
- Sélection d'achievements parmi ceux disponibles.
- Pondération des achievements.
- Classement : tests des différents algorithmes de notation.
- Restriction des achievements pour une catégorie d'utilisateurs (ceux de niveau 3 ne seront pas accessibles au moins de 18 ans et ceux de niveau 2 aux moins de 14 ans).



## IX.2 Vue conceptuelle

Le schéma suivant présente une vue conceptuelle de notre modèle de données.

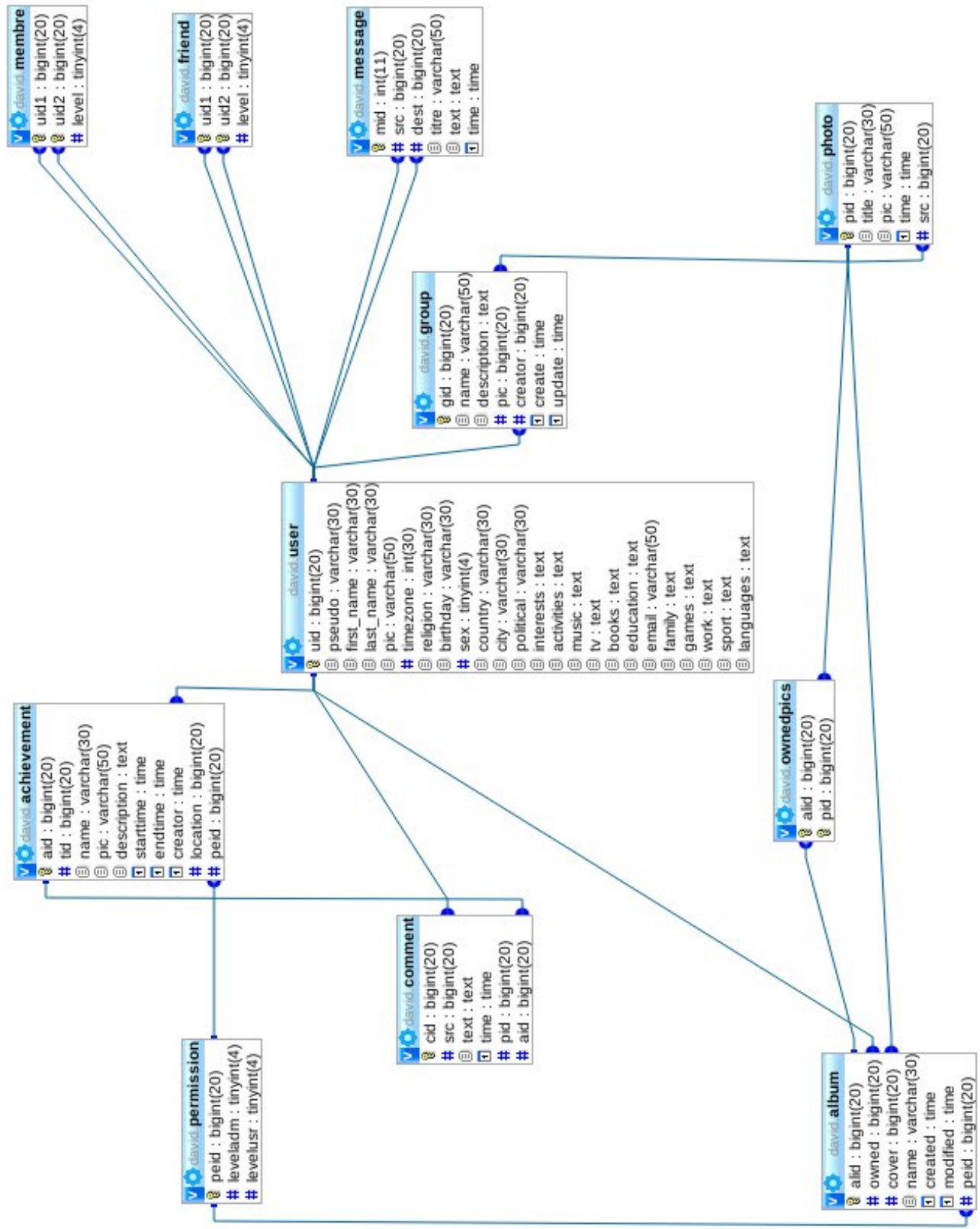






## IX.3 Schéma de la base de données

Le schéma suivant présente une vue schématique de notre modèle de données :





## IX.4 Technologie et portabilité de la base de données

Pour la base de données, nous utilisons un des systèmes basés sur NoSQL les plus populaires du moment à savoir MongoDB..

Celle-ci propose, à l'instar des bases de données relationnelles "classiques", de stocker les données dans un document au format JSON avec un schéma dynamique.

Le choix de cette technologie plutôt qu'une autre a été parfaitement naturelle et découle du choix de Node pour notre côté server, qui implémente parfaitement ce système de bases de données.

Afin de pouvoir utiliser d'autres types de bases de données en cas de besoin pour la suite, nous comptons proposer des convertisseurs de base de données.

Nous proposerons un convertisseur de NoSQL vers MySQL dans un autre langage et d'autres convertisseurs si le besoin s'en fait sentir.





# Chapitre X

## Taille et Performance

Nous présenterons ici les différents chiffres relatifs à la dimension dans laquelle se positionnera notre application.

Pour la plupart des chiffres qui seront ici donnés, tout reste prévisionnel donc approximatif.

**Nombre d'utilisateurs :** Illimité, ou tout du moins la limite du nombre de personne utilisant internet, notre application s'adressant à tous et toutes. Nous serons également probablement limités par les capacités en accueil de notre hébergeur.

**Taille du stockage de données :** Nous aurons besoin d'un espace assez conséquent de stockage, chaque utilisateur ayant la possibilité de publier des photos et des vidéos, mais aussi de laisser des commentaires. Ainsi, plusieurs To ne seraient pas de trop.

**Taille des requêtes :** Les requêtes sont variables.

**Temps de réponse :** Du fait de la nature de notre application, les temps de réponse ne seront pas notre premier axe d'effort comme ce pourrait être le cas dans un jeu par exemple.

**Localisation :** Nous avons pour ambition de proposer notre produit pour un maximum de plates-formes, ainsi, depuis n'importe quel terminal disposant d'un accès à internet, il sera possible d'accéder à notre application.



# Chapitre XI

## Qualité

La plus grande force de la structure du service reste sa modularité. Ainsi, le projet se découpe en plusieurs binaires plus ou moins indépendants les uns des autres, ce qui offre différents avantages sous différents points de vue.

D'abord, cette indépendance relative permet une disposition des ressources plus libre, ce qui est intéressant d'un point de vue pratique ou dans une optique d'optimisation. Ainsi, la base de donnée peut être installée sur une machine autre que celle du serveur web, ou encore du serveur réservé aux clients sur smartphones.

Ensuite, cette organisation permet un usage de langages et de patterns plus ciblé et plus adapté aux fonctions des différents binaires. Cette dernière force également un certain niveau d'abstraction, qui en plus d'être un gage de stabilité du produit, permettra éventuellement d'étendre la portabilité du produit à l'avenir.

Enfin, cette non centralisation des ressources est une mesure de sécurité efficace. En effet, il est intéressant de limiter les dangers éventuels d'une attaque ou d'une perte de données à un seul élément du projet, afin de limiter la casse et de protéger le reste.



## Chapitre XII

### Conclusion

En conclusion de cette étude qui se place au niveau de la conception et de l'architecture de notre EIP, nous pouvons affirmer être aujourd'hui pleinement en mesure d'aborder notre projet l'esprit libre, ayant désormais l'appui d'une ligne directrice sur de nombreux aspects de notre application finale.

Les réflexions et conclusions amenées par nos différents schémas nous ont permis de faire face à de nouvelles problématiques de conception et à initier des dialogues sur des points qui n'avaient pas encore été abordés et qui, dans une phase avancée du projet, auraient pu amener à de plus amples difficultés si elles n'avaient pas été détectées plus tôt.

Nous comprenons aujourd'hui toute l'ampleur et l'utilité de ce travail d'analyse en amont d'un projet de grande envergure, et nous ne manquerons pas de faire évoluer cette étude tout au long de notre EIP afin de respecter la méthode agile de gestion de projet que nous nous sommes fixée et qui nous oblige à revenir de façon itérative sur la réflexion liée à notre conception.