

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de L'enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira – Bejaia
Faculté des sciences exactes
Département Informatique



Mémoire De Fin D'étude

En vue de l'obtention d'un diplôme en Master en Informatique professionnel

Option : Génie Logiciel

Le thème

Conception et réalisation d'un mini-cloud pour les étudiants de l'université de Bejaia

Réaliser par :

- REZZOUG Lamine
- ZENATI Massinissa

Membre de juré :

Examineur : Dr.OUZEGGANE Redouane

Examineur : Dr.GHANEM Souhila

Encadré par : Dr.YAICI Malika

REMERCIEMENT

Nous tenons tout d'abord à remercier Dieu, le tout puissant de nous avoir accordé la force, la volonté et le courage d'achever ce modeste travail.

Un grand merci à nos parents, qui nous ont soutenus, encouragés, offert les moyens, assisté durant tout notre cursus et leurs confiances en nous.

Nos remerciements à notre encadrante Mme YAICI Malika pour sa disponibilité, ses idées et ses conseils durant cette année de travail.

Nos remerciements aux membres du jury qui nous font honneur d'évaluer notre modeste travail.

Nous tenons enfin à remercier nos frères et sœurs pour leurs soutiens, nos camarades de tous les niveaux et tous ceux qui ont contribué de près ou de loin dans la réalisation de ce modeste travail.

Dédicace

Nous dédions ce mémoire, A nos très chers parents qui ont toujours été là pour nous, ont pu créer le climat affectueux et adéquat à la poursuite de nos études et à la réalisation de ce travail.

Aucune dédicace, aucun remerciement ne pourrait présenter un grain de notre reconnaissance encore moins du sentiment d'amour envers vous, prunelles de nos yeux.

A nos frères et nos sœurs

A nos familles

A nos amis

Et à Notre encadrante

Table des matières

Table des figures	V
Liste des tableaux	VII
liste des abréviations	1
Introduction générale	2
I Généralité sur le cloud computing	3
I.1 Introduction	3
I.2 Définition	3
I.3 Les caractéristiques du cloud	4
I.4 Types de services du cloud	4
I.5 Types de cloud	6
I.5.1 Cloud public	6
I.5.2 Cloud privé	6
I.5.3 Cloud hybride	7
I.5.4 Cloud communautaire	7
I.6 Les principaux acteurs dans le cloud computing	7
I.7 Challenges du cloud computing	8
I.7.1 Sécurité et particulièrement la sécurité des données	8
I.7.2 Gestion et conformité aux réglementations	8
I.7.3 Accords sur des niveaux de service et qualité de service	9
I.7.4 Intégration et interopérabilité	9
I.8 Composantes technologiques	10

I.8.1	Les centres de données et fermes de serveurs	10
I.8.2	Virtualisation	10
I.8.3	Interfaces de programmation d'applications	10
I.8.4	Le cryptage dans le cloud	11
I.9	Avantages et inconvénients du cloud computing	11
I.9.1	Avantages	11
I.9.2	Inconvénients	11
I.10	Conclusion	12
II	Architecture et fonctionnement d'un cloud	13
II.1	Introduction	13
II.2	Architectures du cloud computing	13
II.2.1	Front-end	13
II.2.2	Back-end	14
II.3	Systèmes de stockage dans le Cloud	14
II.3.1	Formats de stockage	15
II.3.2	Types de données stockées dans le Cloud	15
II.3.3	Architectures de stockage	16
II.4	Cycle de vie de la donnée dans le cloud computing	18
II.4.1	Phase de transit (transfert) des données dans le cloud	18
II.4.2	Stockage des données dans le cloud	19
II.4.3	Traitements des données dans le cloud	19
II.4.4	Sauvegarde des données	20
II.4.5	Réutilisation des données	20
II.4.6	Suppression des données	20
II.5	Mesures de protection dans le cloud	20
II.5.1	Contrôle d'accès	20
II.5.2	Réplication des données dans le cloud	21
II.5.3	Méthodes de réplication	21
II.5.4	Types de réplication dans le cloud	21
II.5.5	Avantages de la réplication dans le Cloud	22

II.5.6	Inconvénients de la réplication dans le cloud	22
II.6	Authentification	23
II.7	Plateformes de Cloud computing	23
II.7.1	Solutions open source	23
II.7.2	Solutions propriétaires	28
II.8	Conclusions	30
III	Conception et réalisation d'un mini cloud étudiants	31
III.1	Introduction	31
III.2	Problématique	31
III.3	Présentation de L'application	32
III.4	Architecture de l'application	32
III.5	Méthodologie de conception	33
III.5.1	Le formalisme	33
III.5.2	Présentation de l'UML	33
III.6	Identification des acteurs	33
III.7	Identification des cas d'utilisations	34
III.7.1	La liste des cas d'utilisation	34
III.7.2	Description des cas d'utilisation	35
III.7.2.1	La description de cas d'utilisation « authentification » . . .	35
III.7.2.2	La description de cas d'utilisation « gérer un compte » . . .	36
III.7.2.3	La description de cas d'utilisation « gérer un fichier » . . .	36
III.8	Diagramme de séquence	37
III.8.1	Diagramme de séquence «Authentification»	37
III.8.2	Diagramme de séquence «option ajouter un fichier »	38
III.8.3	Diagramme de séquence «option consulter un fichier»	39
III.8.4	Diagramme de séquence «option supprimer un fichier»	40
III.8.5	Diagramme de séquence «gestion des fichier»	40
III.8.6	Diagramme de séquence «option ajouter un compte»	41
III.8.7	Diagramme de séquence «option modifier un compte»	41
III.8.8	Diagramme de séquence «option supprimer un compte »	42

III.8.9 Diagramme de séquence «gestion des comptes»	42
III.9 Diagramme de classe	43
III.10 Dictionnaire des données	43
III.11 Conclusion	44
IV Réalisation	45
IV.1 Introduction	45
IV.2 Environnement de développement	45
IV.2.1 Visual studio code	45
IV.2.2 Node js	46
IV.2.3 Mongo Atlas	46
IV.2.4 Visual Paradigm for UML	46
IV.2.5 Postman	47
IV.3 Technologies logicielles	47
IV.3.1 Javascript	47
IV.4 Organisation du code	48
IV.4.1 Backend	48
IV.4.2 Frontend	49
IV.5 diagramme de déploiement	49
IV.6 Présentation des interfaces de l'application	50
IV.6.1 interface d'Authentification	50
IV.6.2 Interface principale de l'administrateur	51
IV.6.3 Interface ajouter un compte	51
IV.6.4 Interface modifier compte	52
IV.6.5 Interface d'ajout de fichier	53
IV.7 Conclusion	53
Conclusion générale	54
Bibliographie	55

Table des figures

I.1	Différentes couches de services offerts par le cloud computing [2]	5
II.1	Architectures du cloud computing.	15
II.2	Architecture NAS[7]	16
II.3	Architectures SAN[7]	17
II.4	Mode transport et mode tunnel[9]	19
II.5	Les différentes couches d'OpenNebula[13].	25
II.6	Architecture d'Eucalyptus (Alrwais, 2011 ; Naing, 2012)[14]	26
II.7	Architecture openstack[15]	28
III.1	Architecture d'application.	32
III.2	diagramme de cas d'utilisation.	34
III.3	Diagramme d séquence Authentification.	38
III.4	Diagramme de séquence «option ajouter un fichier».	39
III.5	Diagramme de séquence «option consulter un fichier».	39
III.6	Diagramme de séquence «option supprimer un fichier».	40
III.7	Diagramme de séquence «gestion des fichier».	40
III.8	Diagramme de séquence «option ajouter un compte».	41
III.9	Diagramme de séquence «option modifier un compte».	41
III.10	Diagramme de séquence «option supprimer un compte ».	42
III.11	Diagramme de séquence «gestion des comptes».	42
III.12	Diagramme de classe.	43
IV.1	Organisation du code.	48
IV.2	Organisation du code Backend	48

IV.3 Organisation du code Frontend	49
IV.4 diagramme de Déploiement	50
IV.5 Interface d'authentification	51
IV.6 Interface principale de l'administrateur	51
IV.7 Interface d'ajout des comptes	52
IV.8 Interface modifier compt	52
IV.9 Interface d'ajout de fichier	53

Liste des tableaux

III.1 La description de cas d'utilisation « authentification »	35
III.2 La description de cas d'utilisation « Gérer un compte»	36
III.3 La description de cas d'utilisation « Gérer un fichier»	37
III.4 Dictionnaire de données	44

Liste des abréviations

AMI	Amazon Machine Images
API	Application Programming Interface
AWS	Amazon Web Services
CC	Cluster Controller
CLC	Cloud Controller
CLI	Command Line Interface
GCP	Google Cloud Platform
GPL	General Public License
GPU	Graphics Processing Units
HTTP	Hypertext Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IAAS	Infrastructure As A Service
IAM	Identity and Access Management
IIS	Internet Information Services
IPSEC	Internet Protocol Security
KVM	Kernel-based Virtual Machine
NAS	Network Attached Storage
NFS	Network File System
NIST	National Institute of Standards and Technology
PAAS	Platform As A Service
REST	Representational State Transfer
SAAS	Software As A Service
SAN	Storage Area Network
SGBD	Système de gestion de base de donnée
SOAP	Simple Object Access Protocol

SSI	Secure Socket Layer
SWT	Simple Web Token
TCP	ransmission Control Protocol
UC	Central Unit
UML	Unified Modeling Language
UP	Unified Process
VMs	Virtual Machines
XML	Extensible Markup Language
XML-RPC	Remote Procedure Call

Introduction générale

L'informatique est un domaine d'activité scientifique, technique et industriel concernant le traitement automatique de l'information par l'exécution de programmes informatiques par des machines.

De nos jours, ce domaine d'activité est devenu indispensable dans notre vie quotidienne, privée ou professionnelle, il est utilisé à tous les niveaux et dans diverses spécialités : économique, social, culturel, médical et même dans les études, etc.

L'apparition du cloud computing et son évolution est dû au problème de stockage de données de plus en plus volumineux et de la nécessité de partage de ressource.

Le cloud est aussi utilisé dans la gestion des données des entreprises, universités, instituts, bureau administratifs, ainsi que des cabinets d'avocats, médicaux, etc.

Nous avons tenu compte de ce dernier fait, pour choisir le thème de notre projet, qui a pour but de créer un mini cloud pour les étudiants de l'université de Bejaia. Notre mission sera donc de créer une application permettant l'organisation et l'automatisation et le partage des documents. Il s'agira de définir les différentes tâches, de mettre à jour et d'organiser les données collectées. Ce rapport est organisé comme suit :

- Le chapitre 1 concerne les généralités sur le cloud computing.
- Dans le chapitre 2, nous allons détailler le fonctionnement et l'architecture du cloud.
- Le chapitre 3, contient la conception et les scénarios envisagés, les diagrammes UML.
- Dans le dernier chapitre, l'environnement matériel et logiciel du développement de l'application et les différentes interfaces avec leurs descriptions y sont présentés.
- Quant à la conclusion, elle établira un bilan du travail et adressera les perspectives du projet.

Chapitre I

Généralité sur le cloud computing

I.1 Introduction

L'informatique a toujours évolué, elle a mis de nouvelles technologies pour répondre à de nouvelles demandes. L'informatique est centralisée avec l'avènement des data center. Et surtout, elle se dématérialise et devient "l'informatique dans les nuages", ou Cloud Computing. La puissance informatique se virtualise et se consomme à l'endroit et au moment où on en a besoin et devient extensible tout ceci grâce à l'internet.

Dans ce chapitre, nous présenterons les principales caractéristiques du Cloud computing, ses types de services (Saas, Paas, Iaas), et les types de système (public, privé, hybride, communautaire), ainsi que ses composantes technologiques.

I.2 Définition

Le Cloud Computing est un concept qui consiste à stocker et accéder à des données sur des serveurs distants via internet. Le cloud se repose sur une architecture distante. Un fournisseur assure la continuité du service, la maintenance, et le support d'exploitation. Ce modèle offre des services de différentes natures, allant des services d'infrastructure (location de capacités de stockage ou de calcul), des services de plateforme (location d'environnements de développement préconfigurés) ou de services d'applications (location d'applications).

I.3 Les caractéristiques du cloud

Le cloud computing possède un ensemble de caractéristiques correspondant aux définitions suivantes [1] :

- **Accessibilité via le réseau** : Les services sont accessibles en ligne et sur tout type de support. Tout se passe dans un navigateur Internet.
- **Mesurabilité** : L'utilisation du service par le client est supervisée et mesurée afin de pouvoir suivre le niveau de performance et facturer le client en fonction de sa consommation réelle.
- **Solution multi client** : Une même instance d'un logiciel est partagée par l'ensemble des clients de façon transparente et indépendante. Tous les clients utilisent la même version du logiciel et bénéficient instantanément des dernières mises à jour. Chaque client dispose d'un paramétrage utilisateur qui lui est propre.
- **Disponibilité à la demande** : Le service peut être souscrit rapidement et rendu opérationnel automatiquement avec un minimum d'interaction avec le fournisseur.
- **Élasticité** : Des ressources supplémentaires peuvent être allouées au service pour assurer la continuité du service en cas de pic de charge, ou bien être réallouées à un autre service dans le cas inverse.
- **Mutualisation des ressources** : Des ressources utilisées pour exécuter le service sont mutualisées pour servir à de multiples clients. Les multiples serveurs sollicités, totalement interconnectés, ne forment plus qu'une seule ressource virtuelle puissante et performante.

I.4 Types de services du cloud

Trois types de services cloud sont offerts, tels que définis dans la figure 1.1 [2] :

- **SaaS (Software as a Service)** : les utilisateurs sont connectés à des applicatifs métiers qui sont hébergés et gérés sur des infrastructures appartenant à un fournisseur de services. La sécurité, le niveau de performance et les changements de version sont de la responsabilité du fournisseur.

- **PaaS (Platform as a Service)** : Un environnement de programmation et d'exécution avec les outils adéquats est fourni pour permettre aux développeurs de logiciels de créer de nouveaux applicatifs ou d'adapter des versions existantes sans que les machines et les outils logiciels de développement soit propriété de l'entreprise. La sécurité relève dans ce cas de la responsabilité partagée entre le fournisseur de service et l'utilisateur : celui-ci est responsable de la sécurité des applications et de leur environnement alors que le fournisseur de services est responsable de la sécurité des autres ressources.
- **IaaS (Infrastructure as a Service)** : C'est une offre proposant des machines virtuelles de traitement et des capacités de stockage virtuel qui sont gérées à distance par les utilisateurs. Certains fournisseurs peuvent proposer des prestations supplémentaires telles qu'une exploitation de premier ordre des équipements physiques.

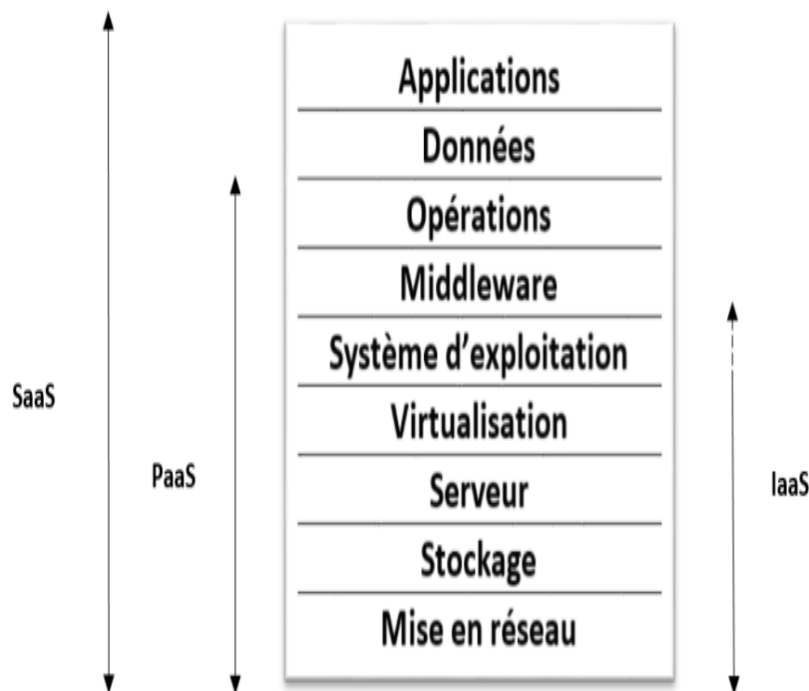


FIGURE I.1 – Différentes couches de services offerts par le cloud computing [2]

I.5 Types de cloud

Nous distinguons quatre types de cloud :

I.5.1 Cloud public

Les utilisateurs n'ont aucun contrôle de l'infrastructure ni aucun pouvoir d'audit. Les fournisseurs mettent à disposition des ressources – par exemple applicatifs ou capacités de stockage – accessibles au moyen de connexions de type internet ou réseau privé virtuel. Les infrastructures et les logiciels associés sont la propriété des fournisseurs. L'utilisateur paie pour les services consommés grâce à des moyens appropriés de facturation. La sécurité est moins élevée que dans le cloud privé. Il existe trois grandes classes de cloud publics :

- **Cloud gratuit** : aucun cout n'est à supporter par l'usager. Le business model repose sur la génération de revenus par la publicité. Les services offerts se limitent par exemple au courrier électronique et à celui d'un moteur de recherche. Aucune mesure de sécurité n'est mise à disposition. Le terminal est la cible potentielle de spams et de cookies, souvent dans le cadre de ce qui est appelé le one-to-one marketing.
- **Cloud payant à services non négociables** : il n'y a pas de publicité et des mécanismes de sécurité sont disponibles. Les termes du contrat de services sont prédéfinis et ne sont pas négociable. De plus, ils peuvent être modifiés unilatéralement par le fournisseur.
- **Cloud payant à services négociables** : dans ce cas, il n'y pas de publicité et un contrat est négocié. Il concerne essentiellement les points suivants : nature des services fournis, périodes de service, conditions de fin de contrat et destination des données en fin de contrat.

Parmi les cloud publics on y retrouve : Microsoft azure, Google cloud, Amazone web service (AWS) [2].

I.5.2 Cloud privé

Le Cloud privé est un réseau informatique propriétaire ou un centre de données qui fournit des services hébergés pour un nombre limité d'utilisateurs. Il peut être géré par

l'entreprise utilisatrice elle-même ou par un prestataire externe qui met à disposition de l'utilisateur un parc de machines s'adaptant à la demande de l'utilisateur (Cloud privé virtuel). Une même infrastructure peut accueillir plusieurs cloud privés virtuels appartenant à différents utilisateurs, chacun peut accéder à son cloud privé via son propre réseau [3].

I.5.3 Cloud hybride

Les deux approches précédentes sont combinées. Les données considérées comme critiques par l'entreprise sont maintenues dans des domaines privés, alors que les données moins critiques sont logées dans un cloud public. Entre les deux peut être mis en place une passerelle de façon à permettre l'échange de données de manière sécurisée. La sécurité des données constitue donc un facteur déterminant de prise de décision dans les choix d'architecture cloud [2].

I.5.4 Cloud communautaire

Il s'agit d'une configuration fermée partagée par plusieurs organisations. Elle peut être constituée d'un périmètre de plusieurs cloud privés connectés entre eux et isolés vis-à-vis de l'extérieur [2].

I.6 Les principaux acteurs dans le cloud computing

Les acteurs définis dans l'architecture de référence de NIST¹. Chaque acteur est une entité qui participe à une transaction ou à un processus et/ou effectue des tâches dans le cloud computing [4].

- **Client cloud** : Une personne ou une organisation qui entretient une relation d'affaires ou utilise le service de fournisseur cloud ou bien les deux
- **Fournisseur** : Une personne, une organisation ou une entité responsable de services à la disposition du client cloud.

1. NIST :Le National Institute of Standards and Technology, est une agence du département du Commerce des États-Unis. Son but est de promouvoir l'économie en développant des technologies, la métrologie et des normes de concert avec l'industrie.

- **Vérificateur cloud** : Une entité qui peut procéder à une évaluation indépendante des services de cloud computing, au fonctionnement du système d'information, à la performance et à la sécurité de la mise en œuvre du cloud.
- **Courtier cloud** : une entité qui gère l'utilisation, la performance et la prestation des services de cloud computing et négocie les relations entre les fournisseurs cloud et les clients cloud.
- **Transporteur cloud** : Un intermédiaire qui fournit la connectivité et le transport des services de cloud computing depuis le fournisseur cloud jusqu'au client cloud.

I.7 Challenges du cloud computing

Le cloud computing est confronté à quatre challenges [2].

I.7.1 Sécurité et particulièrement la sécurité des données

Les failles de sécurité du cloud computing sont devenues une principale problématique, parmi ces problèmes [2] :

- Absence de connaissance sur la localisation des données stockées.
- Les plates-formes partagées sont moins sécurisées que les plates-formes monutilisateur.
- Utilisation de la virtualisation comme technologie sous-jacente, créant un sentiment d'absence de contrôle des applications hébergées.

I.7.2 Gestion et conformité aux réglementations

Beaucoup de grandes entreprises sont encore en train d'établir des modèles pour la gestion des données dans le cadre des services cloud et pour leur protection. Ces points sont d'autant plus importants quand il y a des réglementations à respecter comme la législation de l'union européenne sur la protection des données [2].

I.7.3 Accords sur des niveaux de service et qualité de service

La préoccupation des grandes organisations est la qualité de service (disponibilité, fiabilité, performance) [2] :

- Tous les fournisseurs de services ne proposent pas toujours des accords de niveaux de service bien définis ou ceux proposés ne satisfont pas les exigences des entreprises. Par exemple, le temps de reprise du fonctionnement après arrêt est décrit par les meilleurs délais, au lieu d'une durée déterminée ou d'une plage horaire. Les mesures correctives du fournisseur ne couvrent pas les pertes potentielles du client en cas de panne.
- Les clients sont pratiquement dans l'incapacité d'influencer les termes des contrats de service. Du point de vue du fournisseur de service, il est également difficile de pouvoir personnaliser les contrats pour chaque client.
- La performance globale d'un service cloud dépend de la performance de composants qui sont hors du contrôle non seulement du client mais aussi du fournisseur de services cloud, comme la connexion réseau par les opérateurs.

I.7.4 Intégration et interopérabilité

L'identification et la migration des applications éligibles pour être transférées sur le cloud sont rendues complexes par les interdépendances caractérisant de façon typique les applications de gestion.

L'intégration et l'interopérabilité des applications destinées à être exploitées sur le cloud sont confrontées aux enjeux suivants [2] :

- Un manque d'interfaces standard (API) pour intégrer des applications existantes a des services cloud.
- Des supports logiciels qui doivent résider dans le cloud pour des raisons de performance, mais pour lesquels des licences ne peuvent être accordées pour des environnements cloud.
- Des questions d'interopérabilité entre les services cloud de différents fournisseurs. Ces questions s'expriment par des interrogations sur la façon dont des applications

disparates, déployées sur des sites géographiquement dispersés, peuvent interagir sans obstacles et fournir les niveaux de service attendus.

I.8 Composantes technologiques

I.8.1 Les centres de données et fermes de serveurs

Un centre de données (data center) est un site physique sur lequel se regroupe des équipements constituant un système d'information de l'entreprise (mainframes, serveurs, baies de stockage, équipements réseaux, etc.). Il peut être interne ou externe à l'entreprise [5].

I.8.2 Virtualisation

Le concept de la virtualisation consiste à considérer et à traiter comme une structure indépendante, une entité logique constituée à partir de ressources physiques. Entre les ressources physiques et les propriétés de l'entité logique, propriétés qui sont accessibles à un utilisateur humain ou non, sont mises en œuvre des mécanismes logiciels et matériel ou un des deux cachant à l'utilisateur la façon dont les services sont délivrés opérationnellement.

Ainsi, la virtualisation peut se manifester sous deux formes différentes [2] :

- Partage d'une entité physique ou logicielle en entités virtuelles multiples indépendantes les unes des autres.
- Combinaison d'entités physique multiples en une seule entité disposant d'un large spectre de fonctionnalités et de grandes capacités qu'une seule entité physique ne pourrait fournir.

I.8.3 Interfaces de programmation d'applications

API (Application Programming Interface) offre des fonctionnalités aux clients, comme l'auto-provisionnement et le contrôle des services et des ressources. Ils permettent également la communication entre les applications étrangères et le service sur le cloud. Le type d'API dépend du modèle de déploiement [5].

I.8.4 Le cryptage dans le cloud

Le cryptage est le processus de codage des messages ou des informations d'une manière que seuls les parties autorisées puissent lire cette information, empêchant ainsi des parties indésirables à les intercepter. Cela se fait habituellement à l'aide d'une clé de cryptage qui spécifie la façon dont le message est codé. Ensuite, la partie autorisée utilise une clé de décryptage secrète pour déchiffrer le message et le lire [5].

I.9 Avantages et inconvénients du cloud computing

I.9.1 Avantages

- **Accessibilité garantie** : avec le cloud, les services et les applications sont accessibles à tout moment et depuis n'importe quel ordinateur, téléphone portable ou tablette.
- **Coût optimisé** : le cloud computing est avant tout économique, il représente un gain de cout non négligeable.
- **Flexibilité et partage** : les services sont flexibles et peuvent être ajustés à tout moment en fonction des besoins et de l'activité de l'entreprise. Celle-ci peut diminuer ou augmenter les ressources disponibles, payant ainsi seulement ce qu'elle consomme. Ces ressources peuvent être partagées permettant aux employés de travailler à plusieurs sur un même document, et ce en temps réel.
- **Mises à jour automatiques** : en plus de la maintenance, le fournisseur cloud se charge de toutes les mises à jour du service, ce qui permet à l'entreprise et à ses employés de se concentrer plus efficacement sur leurs missions et par la même occasion, d'optimiser leur productivité.

I.9.2 Inconvénients

- **Connexion Internet obligatoire** : Les services nécessaires au différents processus dépendent de la connexion internet, cette dernière perturbe son fonctionnement en cas de panne.

- **Confidentialité et sécurité des données** : les données sont hébergées en dehors de l'entreprise. Le fournisseur proposant le service héberge les données de l'entreprise utilisatrice. Cela peut donc poser un risque potentiel pour l'entreprise de voir ses données mal utilisées ou volées.
- **Dépendance** : si l'entreprise souhaite des fonctionnalités très spécifiques, il peut être difficile de convaincre le fournisseur de proposer ces fonctionnalités

I.10 Conclusion

Le cloud computing est l'une des grandes révolutions informatiques, dans la manière de gérer et de partager des ressources informatiques, c'est une technologie qui permet d'effectuer plusieurs tâches à moindre coût.

Chapitre II

Architecture et fonctionnement d'un cloud

II.1 Introduction

L'explosion de l'usage des Smartphones et le développement des Cloud a engendrée une masse gigantesque de données de manière plus ou moins temporaire et sécurisé. A chaque avancée technologique, le nombre d'utilisateurs croît et les données qui seront stockée augmenteront en parallèle avec ce dernier.

II.2 Architectures du cloud computing

L'architecture Cloud Computing comprend de nombreux composants cloud, on peut diviser l'architecture cloud en deux parties :

- Front-end.
- Back-end.

Chacune est connectée via un réseau, généralement Internet.

II.2.1 Front-end

Fait référence à la partie client du système de cloud computing. Il se compose d'interfaces et d'applications nécessaires pour accéder aux plateformes de cloud computing

comme le démontre la figure II.1, Exemple : Navigateur Web.

II.2.2 Back-end

Il se compose de toutes les ressources nécessaires pour fournir des services de cloud computing. Il comprend un énorme stockage de données, des machines virtuelles, un mécanisme de sécurité, des services, des modèles de déploiement, des serveurs, etc.

- **Application** : C'est une partie importante de l'architecture principale. Il fait référence à l'interface utilisateur (dashboard service) que le back-end offre à l'utilisateur final pour envoyer des requêtes. Cette couche du back-end prend en charge les demandes et les exigences du client.
- **Service** : c'est l'un des composants les plus importants du cloud. Il fait référence aux trois principaux types de services (SaaS, PaaS, IaaS). Il gère le type de service auquel l'utilisateur accède.
- **Stockage** : il maintient et gère toute quantité de données sur Internet, cependant, la capacité de stockage varie en fonction des fournisseurs de services disponibles sur le marché.
- **La gestion** : il alloue des ressources spécifiques à une tâche spécifique. En outre, il gère les fonctions de l'environnement cloud, il aide à la gestion de composants tels que l'application, la tâche, le service, la sécurité, le stockage de données et l'infrastructure cloud. En simples termes, il établit une coordination entre les ressources.
- **Sécurité** : La sécurité fait partie intégrante de l'infrastructure cloud, elle permet de protéger les ressources, les systèmes, les fichiers et l'infrastructure du cloud. En outre, elle assure la sécurité du serveur cloud avec des pare-feu virtuels, ce qui permet d'éviter la perte de données.

II.3 Systèmes de stockage dans le Cloud

Le stockage dans le Cloud est l'abstraction, la mise en commun et le partage des ressources de stockage via Internet. Ce dernier est facilité par les technologies du Cloud computing.

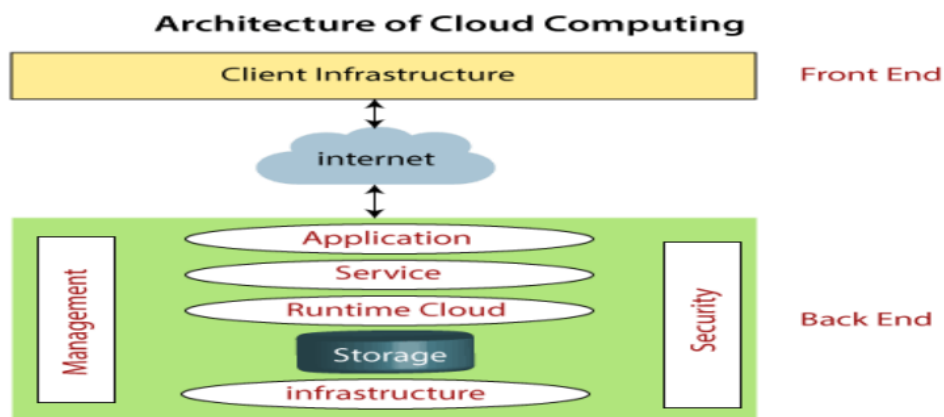


FIGURE II.1 – Architectures du cloud computing.

II.3.1 Formats de stockage

On distingue trois formats de stockage dans le Cloud [6] :

- **Mode fichier** : Le stockage en mode fichier est le plus utilisé sur les systèmes NAS¹. Il permet d'organiser les données et de les présenter aux utilisateurs. Sa structure hiérarchique permet de parcourir les données du haut vers le bas en toute simplicité, mais augmente le temps de traitement.
- **Mode bloc** : Le stockage en mode bloc permet de diviser un seul volume de stockage en plusieurs instances individuelles appelées blocs. Il s'agit d'une solution de stockage rapide à faible latence, idéale pour les charges de travail hautes performances.
- **Mode objet** : Le stockage en mode objet attribue à chaque donnée des identifiants uniques, que l'on appelle les métadonnées. Compte tenu du fait que les objets ne sont ni compressés ni chiffrés, ils sont rapidement accessibles à très grande échelle. Il s'agit donc d'une solution idéale pour les applications Cloud native.

II.3.2 Types de données stockées dans le Cloud

Pour bien comprendre la problématique du stockage de données dans le Cloud computing, il est utile de prendre conscience de leur taxonomie vis-à-vis de leurs usages [2] :

1. NAS : Network Attached Storage, c'est un boîtier de stockage en réseau

- Données relatives aux applications critique dont l'accessibilité rapide est indispensable et qui sont conservées sur des supports : data sensitive applications.
- Données relatives aux applications transactionnelles en ligne pour lesquelles le temps de mise à jour des informations sur disque est critique : I/O sensitive applications.
- Données utilisées pour des calculs qui nécessite peu d'entrées/sorties : computing intensive applications.

II.3.3 Architectures de stockage

Deux types d'architecture ont été développés [7] :

2.3.3.1 Architectures NAS

Architectures NAS (Network attached Storage) pour laquelle les ressources de stockage sont distribuées sur le réseau accessible au moyen de protocoles de haut niveau tels que NFS (network file system) assure un accès transparent a des ressources distantes en donnant l'impression à l'utilisateur que ces ressources sont locales, quels que soient les réseaux et protocoles utilisé, il est indépendant du système d'exploitation même s'il doit maintenir la sémantique UNIX.

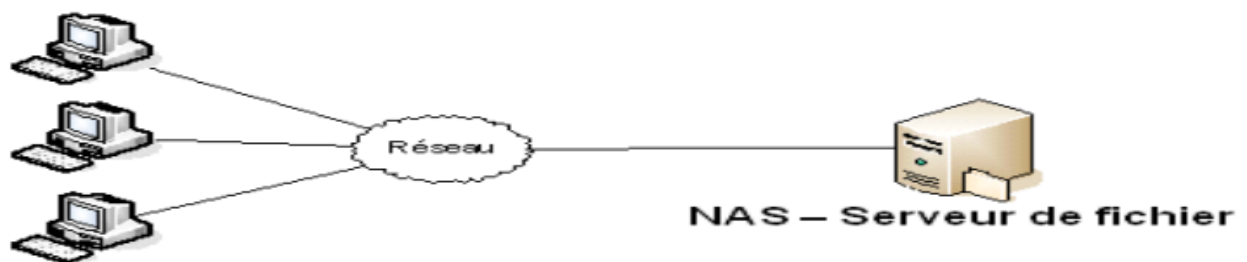


FIGURE II.2 – Architecture NAS[7]

- **Avantage**

- Augmentation de la capacité de stockage, en ajoutant des disques durs. Sans remplacer des serveurs, et encore moins de désactiver le réseau.
- Meilleure performance.
- Configuration simple.
- tous les périphériques en réseau ont accès au NAS.
- Tolérance aux pannes

- **Inconvénients**

- Déconseillé avec des applications demandant de grosses performances disques comme des bases de données.
- Demande beaucoup de ressources CPU².

2.3.3.1 Architectures SAN

Architectures SAN (Storage area network) est configurée autour de réseaux à haut débit auxquels sont connectées des unités de stockage fonctionnant indépendamment.

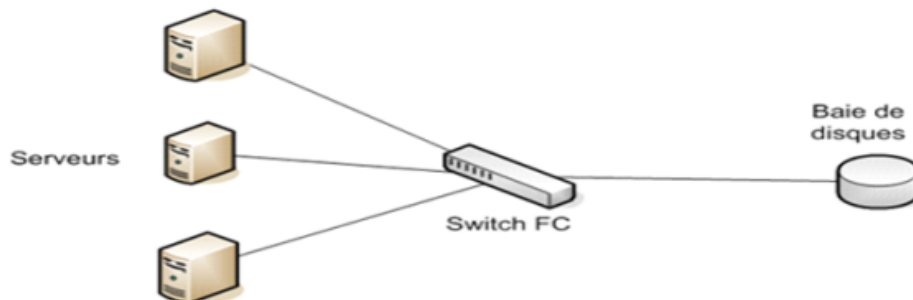


FIGURE II.3 – Architectures SAN[7]

- **Avantage**

- Fournit de grosses performances disques.

2. CPU : Central processing unit, est un composant présent dans de nombreux dispositifs électroniques qui exécute les instructions machine des programmes informatiques. Avec la mémoire, c'est notamment l'une des fonctions qui existent depuis les premiers ordinateurs.

- Fournit une capacité disque illimitée avec l'ajout sans cesse de nouveaux périphériques de blocs pour sauvegarder les données.
- Il permet la consolidation des données en évitant de devoir à chaque fois de rajouter des périphériques de blocs séparés des autres.

- **Inconvénients**

- Il demande des ressources humaines spécifiques ou une formation du personnel afin de pouvoir l'administrer et le maintenir correctement
- Demande beaucoup de ressources CPU.

II.4 Cycle de vie de la donnée dans le cloud computing

Les données dans le cloud computing par plusieurs phase qui sont [8] :

II.4.1 Phase de transit (transfert) des données dans le cloud

La phase liée à l'envoi des données depuis les systèmes internes d'une entreprise vers le Cloud. Les données peuvent être chiffrées en internes par l'entreprise et ensuite envoyées, ou alors on utilise une couche de transport intégrant cette fonction de chiffrement. Dans cette seconde catégorie, les protocoles standards qui sont IPSEC (internet Protocol Security) et SSL (secure socket layer) sont très répandus. En liaison avec une authentification basée sur des clés asymétriques (certificats à clé publique par exemple), ces protocoles permettent de transmettre des données en toute sécurité vers ou depuis le Cloud, ainsi les systèmes deviennent fiables et faciles à utiliser [9].

2.4.1.1 SSL (secure socket layer)

Un protocole SSL (Secure Socket Layer) permet de sécuriser la transmission de données sur Internet. Grâce au chiffrement des données transmises par HTTPS. Lorsqu'un certificat SSL est installé sur une plateforme cloud, celui-ci génère des clés de chiffrement :

- **Une clé publique** : celle-ci est installée sur une plateforme en ligne et permet aux utilisateurs de chiffrer leurs informations avant qu'elles soient envoyées (leurs identifiants de connexion par exemple).
- **Une clé publique** : celle-ci est installée sur une plateforme en ligne et permet aux utilisateurs de chiffrer leurs informations avant qu'elles soient envoyées (leurs identifiants de connexion par exemple).

2.4.1.2 IPSEC (internet Protocol Security)

IPsec (Internet Protocol security) est un cadre normalisé pour sécuriser les communications IP en cryptant et/ou en authentifiant chaque paquet IP dans un flux de données. Ce dernier a deux mode de fonctionnement :

- **Mode transport** : Dans ce mode uniquement les données transférées sont chiffrés et/ou authentifiés. Le reste du paquet IP est inchangé et de ce fait le routage des paquets n'est pas modifié.
- **Mode tunnel** : Dans ce mode c'est la totalité des paquet IP qui sont chiffré et/ou authentifiés.

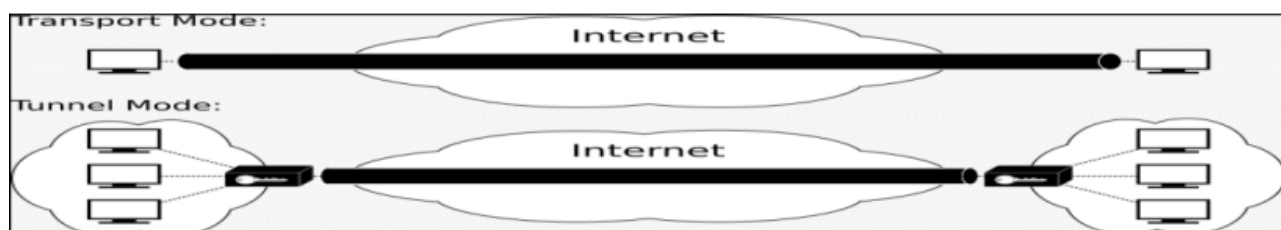


FIGURE II.4 – Mode transport et mode tunnel[9]

II.4.2 Stockage des données dans le cloud

Les données fraîchement créés ou récolté doivent être stockées dans un environnement sécurisé est protégé. De ce fait un process de restauration doit voir le jour afin de s'assurer de la bonne conservation des datas dans le cycle.

II.4.3 Traitements des données dans le cloud

Dans cette étape, les données collectées doivent être traitées, elles peuvent subir soit :

simple compression. Un cryptage des informations. Un data wrangling (nettoyage des données).

II.4.4 Sauvegarde des données

Lors de cette étape, une copie des données est créée puis stockée dans un nouveau support afin d'optimiser sa sécurité, elle peut être réalisée plusieurs fois lors du cycle de vie des données.

II.4.5 Réutilisation des données

Les données peuvent être réutilisées pour de nombreuses raisons par exemple : répondre à une interrogation, ajouter des données nouvelles pour les ajouter à des anciennes données afin d'obtenir de nouveaux résultats, ... ect

II.4.6 Suppression des données

Une fois que des données ont été récupérées depuis un Cloud, il est important de s'assurer leur destruction. Il convient de demander quels sont les engagements, moyens et procédures mise en œuvre par un fournisseur pour effacer toute trace de vos données. Dans ce cas, le chiffrement peut être utilisé. En effet, sans la clé pour les déchiffrer, des données préalablement chiffrées sont totalement inutilisables. Donc pour détruire des données, il suffit de détruire la clé de chiffrement. C'est ce concept qui permet de s'assurer que des données sont bien inaccessibles même dans le cas extrême où un fournisseur de Cloud vient de détruire la clé sans prévenir.

II.5 Mesures de protection dans le cloud

II.5.1 Contrôle d'accès

Contrôler l'accès aux données s'appuie sur des mécanismes d'authentification. Une personne, un système ou un programme doit être fiable (saint) afin de pouvoir accéder aux données. Cloud Identity and Access Management (IAM) permet aux administrateurs

d'accorder les autorisations nécessaires aux utilisateurs intervenant sur des ressources spécifiques. Grâce à cette solution, contrôler complètement l'accès aux ressources Cloud et leur visibilité afin de centraliser leur gestion.

II.5.2 Réplication des données dans le cloud

La réplication est la copie continue des changements de données d'un site principale vers un site secondaire. Les deux sites sont généralement situés dans différents serveurs physiques, résultant dans le cadre de l'équilibrage de charges en répartissant les requêtes, offrant ainsi une capacité de reprise. Le serveur secondaire peut être configuré comme un serveur de secours en cas de défaillance du serveur principal [10].

II.5.3 Méthodes de réplication

- Réplication instantanée : Les données sont copiées du site principal vers les secondaires. Les changements au niveaux des sites secondaires doivent provenir du principal. Ainsi, seuls les secondaires sont interrogés, mais leurs données ne peuvent être modifiées par les clients.
- Réplication par fusion : Les données sont combinées de deux ou plusieurs sites vers un troisième site. Cette méthode est plus difficile à mettre en œuvre qu'une réplication instantanée.
- Réplication transactionnelle : La donnée est copiée complètement, suivit des mises à jours qui sont copiées périodiquement des sites principaux vers les sites secondaires.

II.5.4 Types de réplication dans le cloud

- Réplication synchrone : Garantit que, lorsqu'une transaction met à jour une réplique primaire, toutes ses répliques secondaires sont mises à jour dans la même transaction. L'avantage essentiel de la mise à jour synchrone est de garder toutes les données au dernier niveau de mise à jour. Le système peut alors garantir la fourniture de la dernière version des données quel que soit la réplique accédée. Les inconvénients sont cependant multiples, ce sont d'une part, la nécessité de gérer des transactions multiples coûteuses en ressources et d'autre part, la complexité des algorithmes de

gestion de panne d'un site, etc. C'est pour cela que l'on préfère souvent le mode de mise à jour asynchrone.

- Réplication asynchrone : Lorsqu'une transaction met à jour une copie primaire, chaque copie secondaire est mise à jour dans une transaction séparée. L'avantage est la possibilité de mettre à jour en temps choisi des données, tout en autorisant l'accès aux versions anciennes avant la mise à niveau. L'inconvénient est bien sûr que l'accès à la dernière version n'est pas garanti [11].

II.5.5 Avantages de la réplication dans le Cloud

- Une Disponibilité des données : permettre à un serveur secondaire de prendre rapidement la place du serveur principal lorsque celui-ci échoue, ou alors lorsque les répliques répondent aux requêtes qui sollicitent une même donnée. S'il y a une perte de toute connectivité réseau, avoir une réplique complète signifie qu'il y a toujours accès à toutes les données.
- Favorisation du partage et du parallélisme ou équilibrage de charges : permettre à plusieurs serveurs de servir les mêmes données.
- Amélioration des performances d'accès aux données répartis (meilleur temps de réponse, diminution du temps de transfert des données, etc) [12].

II.5.6 Inconvénients de la réplication dans le cloud

- La conservation de doublons des mêmes données dans divers emplacements entraîne une augmentation du stockage et des frais généraux du processeur.
- L'exécution et la gestion du processus de réplication nécessitent du temps engagé de la part d'une équipe interne pour garantir la cohérence des données copiées avec les données source d'origine.
- La préservation de la cohérence entre les répliques de données peut augmenter le trafic réseau. diminution du temps de transfert des données, etc).
- La latence ou les interruptions de service pendant le transfert de données peuvent entraîner des difficultés de réplication des données. Le processus

- La synchronisation des mises à jour entre les environnements distribués est compliquée car la copie de données à partir de diverses sources à des intervalles de temps différents peut entraîner une désynchronisation de certains ensembles de données avec le reste.

II.6 Authentification

L'authentification est le processus qui consiste à déterminer l'identité d'un principal de sécurité. L'autorisation consiste à accorder à un principal authentifié l'autorisation d'effectuer une action ou d'accéder à une ressource. Parfois, l'authentification est raccourcie AuthN et l'autorisation est raccourcie vers AuthZ . Les applications Cloud natives doivent s'appuyer sur des protocoles HTTP ouverts pour authentifier les principaux de sécurité, car les clients et les applications peuvent s'exécuter n'importe où dans le monde sur n'importe quelle plateforme ou n'importe quel appareil. Le seul facteur commun est HTTP.

II.7 Plateformes de Cloud computing

Les solutions du Cloud sont ainsi classées en deux grandes catégories : les solutions de développements propriétaires et les solutions open source.

II.7.1 Solutions open source

Les solutions open source du Cloud computing sont destinées au déploiement de l'architecture en privé pour un usage en interne.

II.7.1.1 OpenNebula

Il s'agit d'une plateforme purement open source permettant de déployer des Cloud privés, publics et hybrides. Elle est centrée autour d'un nœud appelé front-end qui permet de superviser toute l'architecture, et elle est basée sur la personnalisation en offrant plusieurs modules configurables et adaptés aux besoins, ce qui est un avantage mais aussi



un inconvénient car ça engendre des difficultés de configuration entraînant des erreurs de mise en œuvre et des échecs de déploiement des machines virtuelles dans le réseau[13].

L'architecture interne d'OpenNebula est constituée de trois couches d'éléments appelées respectivement : Tools, core et drivers.

Tools : C'est l'ensemble des outils de gestion de l'architecture. Il est constitué des interfaces de lignes de commandes CLI (Command Line Interface) pour l'interaction avec le système.

Core : C'est un ensemble de composants impliqués dans la gestion et le contrôle des nœuds, des utilisateurs et des machines virtuelles de l'architecture. Elles communiquent entre eux à travers le protocole XML-RPC³.

Driver : C'est à ce niveau que se déroulent les processus liés aux transferts de machines virtuelles d'un nœud à un autre.

3. Un protocole RPC, une spécification simple et un ensemble de codes qui permettent à des processus s'exécutant dans des environnements différents de faire des appels de méthodes à travers un réseau

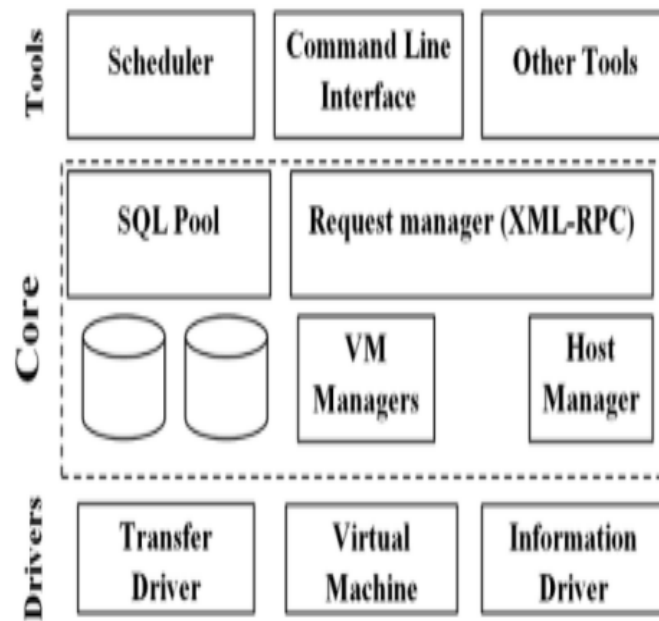


FIGURE II.5 – Les différentes couches d’OpenNebula[13].

II.7.1.2 Eucalyptus



Eucalyptus est un outil open source développé en C, Java, Python. eucalyptus est disponible en deux licence, une licence GPL (General Public License) gratuite supportant les hyperviseurs Xen (un hyperviseur de machine virtuelle) et KVM (Kernel-based Virtual Machine) et une licence commerciale offrant des fonctionnalités avancées telles que le support de VMware. Eucalyptus permet de construire des solutions privées du Cloud computing, son principal avantage c’est qu’il est intégré dans les distributions Ubuntu et Debian. L’architecture d’Eucalyptus est constituée de quatre composants principaux[14].

Le contrôleur de nœud (Node controller NC) : Permet de contrôler l’exécution, et

l'arrêt des machines virtuelles présentes sur le nœud où il est exécuté.

Le contrôleur de cluster (cluster controller CC) : Permet de collecter les informations qui se trouvent sur les différents nœuds d'un cluster et de planifier l'exécution des machines virtuelles sur chaque nœud.

Le contrôleur de stockage (Warlus) : Permet de gérer l'accès au service de stockage.

Le contrôleur de Cloud (CLC) : C'est la partie front-end et le point d'accès des utilisateurs et administrateurs du système. Permet de collecter des informations sur les nœuds et de planifier leur exécution au travers des contrôleurs de clusters (Ccs).

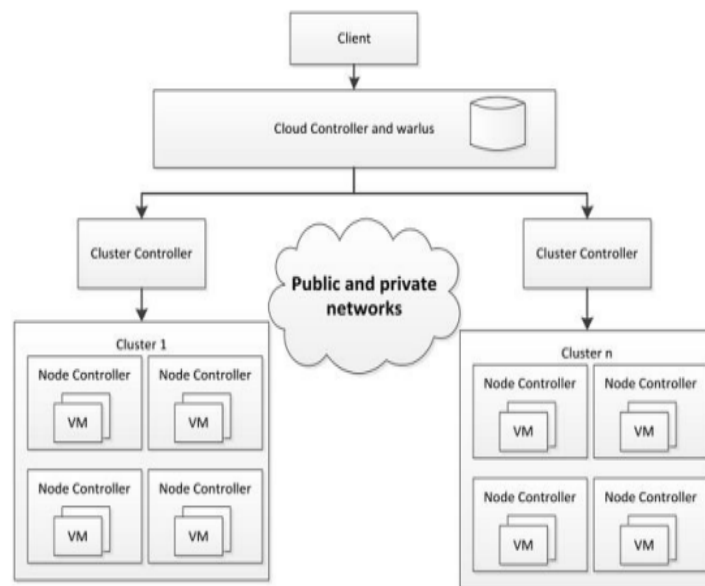


FIGURE II.6 – Architecture d'Eucalyptus (Alrwais, 2011 ; Naing, 2012)[14]

II.7.1.3 Openstack



OpenStack est une plateforme qui permet de créer et gérer des clouds privés ou publics à partir de pools de ressources virtuelles. Il est constitué d'un ensemble d'outils qui permettent d'assurer les principaux services du Cloud computing tel que le calcul, la mise en réseau, le stockage, etc[15].

Composants

Six services essentiels assurent la puissance de calcul, la mise en réseau, le stockage, la gestion des identités et la gestion des images.

Nova : C'est un outil qui permet de gérer des ressources de calcul d'OpenStack et des accès, et de gérer la planification, la création et la suppression des ressources.

Neutron : Permet de connecter les réseaux pour tous les autres services OpenStack.

Swift : C'est un service de stockage en mode objet à forte tolérance aux pannes, qui permet de stocker et récupérer les objets de données non structurées au moyen d'une API RESTful.

Cinder : Cinder est un service de stockage persistant en mode bloc, accessible via une API en libre-service.

Keystone : Permet de prendre en charge l'authentification et l'autorisation de l'ensemble des services OpenStack.

Glance : Permet de stocker et récupérer les images disque des machines virtuelles, depuis

divers emplacements.

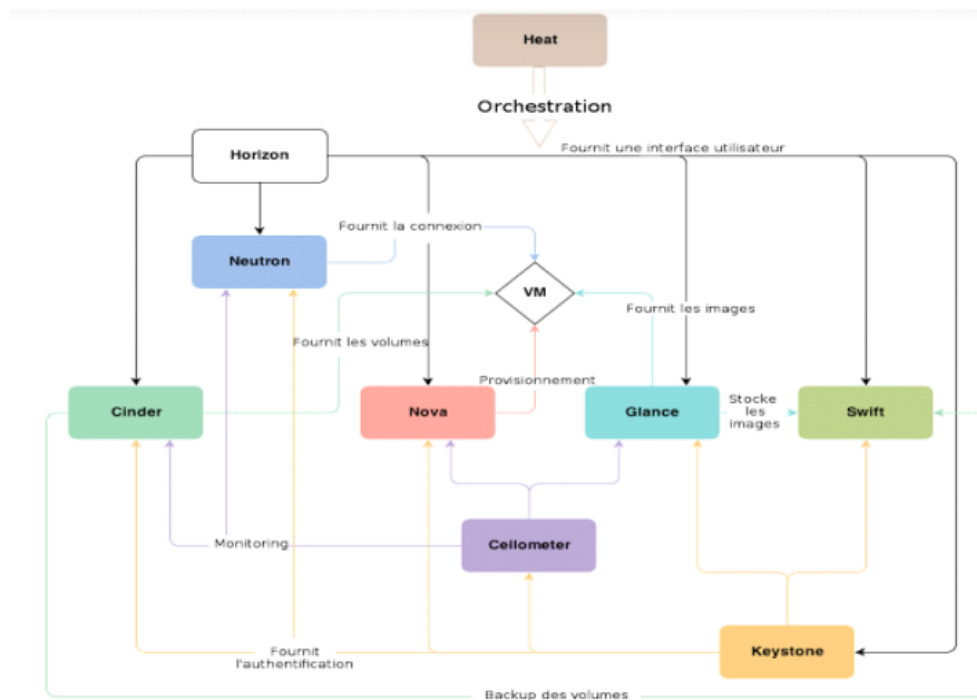


Figure 8

FIGURE II.7 – Architecture openstack[15]

II.7.2 Solutions propriétaires

II.7.2.1 Microsoft azure



C'est une plateforme de Microsoft pour les services PaaS du cloud computing. Il s'agit d'une plateforme de développement d'applications fournissant les services d'exécution et d'administration d'applications en offrant les outils nécessaires.

Azure est une plateforme flexible qui supporte plusieurs langages de Programmations tels que .Net, C, Java, PHP, Python, etc. De plus, elle supporte les Standards et protocoles tels que SOAP, XML[16].

Microsoft azure est constitué de trois composants :

- **Windows azure** : Windows azure est la plateforme en elle-même peut à son tour être divisé en 3 composants :
 - **Fabric** : Il est constitué d'un ensemble de serveurs, chaque serveur faisant tourner plusieurs VMs (machine virtuelle) qui vont être utilisés selon les besoins afin de représenter une instance compute ou Storage. Il est constitué d'un contrôleur (fabric controller), qui permet de s'en charger des opérations sur les machines virtuelles.
 - **Compute** : Ce composant se charge de l'exécution du code. Pour cela le développeur doit créer des rôles, qui vont se comporter tel un serveur IIS (Internet Information Services), ou comme un serveur Windows (Windows rôle) selon les besoins.
 - **Storage** : Permet aux développeurs de stocker dans une instance azure des données non relationnelles
- **SQL azure** : SQL azure est un système de gestion de base de données dans le cloud. Son utilisation revient à utiliser une base de données SQL server.
- **AppFabric** : Ses composants permettent de gérer la communication entre les applications, qu'elles soient hébergées sur azure ou sur site.
 - **Service bus** : Permet d'exposer des services simplement, puis le bus se charge du routage des requêtes vers le service concerné. Il supporte des connexions en full duplex, et propose des accès au choix par REST, tcp ou http.
 - **Access control** : Permet de gérer l'accès au service bus suivant des mécanismes tel que ' OAUTH' et les simples 'web tokens' (SWT) pour les services REST.
 - **Composite App service / composition model** : Fournissent un environnement de développement pour faciliter la création, la gestion et le déploiement d'application composite.

II.7.2.2 Amazon Elastic Compute Cloud (EC2)

Une instance EC2 est un serveur virtuel hébergé dans Elastic Compute Cloud (EC2)



pour exécuter des applications sur l'infrastructure Amazon Web Services (AWS).

AWS est une plateforme de Cloud computing, alors qu'EC2 est un service qui permet d'exécuter des programmes applicatifs.

Dans le but de satisfaire les besoins de chacun, Amazon propose une gamme de types d'instances aux configurations différentes en termes d'UC (Central Unit), de mémoire, et de ressources de stockage et réseau.

Ces dernières sont regroupées en catégorie selon des profils applicatifs cibles : général, optimisé pour le calcul, instances GPU (Graphics Processing Unit), optimisé pour la mémoire, optimisé pour le stockage et micro-instances.

Les instances sont créées à partir d'images machines AMI (Amazon Machine Images). Ces images machines sont comparables à des modèles configurés avec un système d'exploitation et d'autres logiciels, le tout formant l'environnement d'exploitation de l'utilisateur.

Les utilisateurs peuvent sélectionner une instance AMI auprès d'AWS, de la communauté des utilisateurs ou d'AWS Marketplace. Ils peuvent également créer leurs propres AMIS et les partager. L'utilisation des services d'Amazon est facturée selon le temps d'utilisation des machines louées[17].

II.8 Conclusions

Dans ce chapitre, nous avons mené une étude sur le système de stockage des données dans un cloud ainsi que les alternatives open source est propriétaire des plateformes du cloud computing. Ceci nous a permis d'avoir une idée sur les techniques disponibles pour la création d'un environnement du cloud.

Chapitre III

Conception et réalisation d'un mini cloud étudiants

III.1 Introduction

Dans ce chapitre on s'intéresse à l'étude et la conception d'un mini Cloud destiné aux étudiants de l'université de Bejaia. Le langage de modélisation engagé pour ce fait est l'UML qui est consacré à faire une étude préliminaire de l'application à concevoir. Ce présent chapitre offre également une conception UML bien détaillé de l'application que nous allons réaliser après cette étape de conception.

III.2 Problématique

L'université de Bejaia dispose d'un nombre de ressources et de matérielles comme les serveurs, les Datacenter. Notre travail de master est de mettre en place une solution afin de permettre aux étudiants appartenant à l'université de partager entre eux des documents comme des cours, exercices, etc.

Quel type de solution choisir ? Quel est le matérielle nécessaire ? Est-ce que cette solution est adaptée pour notre université ?

III.3 Présentation de L'application

Le mini Cloud vas permettre aux étudiants d'accéder et de partager facilement des documents (Cours, Emds, ... etc.) stocker par ces derniers dans l'application, et assister l'administrateur dans la gestion des utilisateurs et dans la gestion des fichiers stocker.

III.4 Architecture de l'application

Nous allons présenter l'architecture de notre application dans le schéma que nous établirons ci-dessous

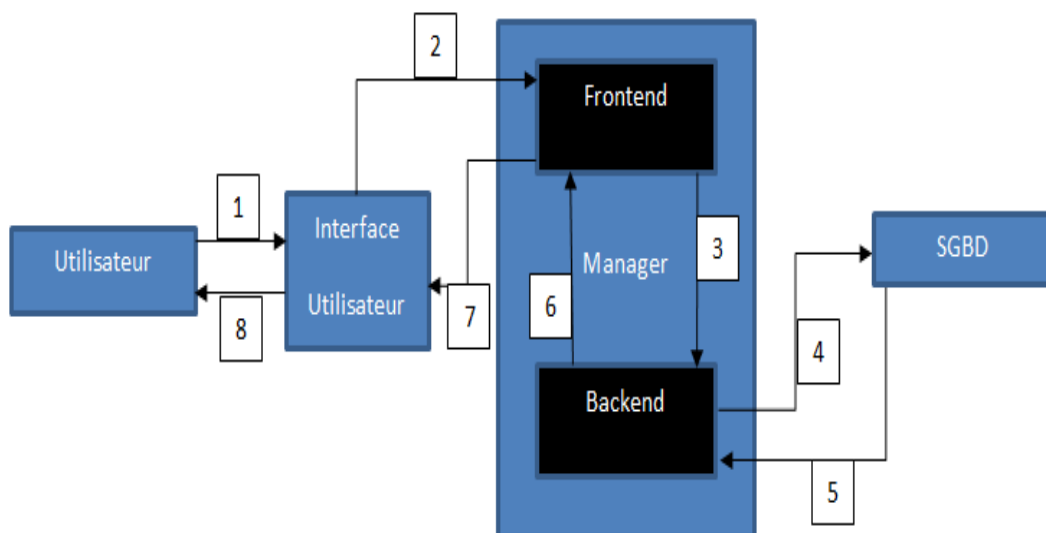


FIGURE III.1 – Architecture d'application.

1. L'utilisateur choisi une action.
2. L'interface utilisateur envoi une requête aux manager(Frontend).
3. Frontend envoi une requête aux Backend pour utiliser l'API adéquat.
4. Backend envoi une requête au SGBD.
5. SGBD renvoi la réponse au backend.
6. Backend envoi la réponse aux frontend.

7. Frontend charge l'interface adéquate et l'envoi à l'interface utilisateur.
8. L'utilisateur peut voir le résultat de son action.

III.5 Méthodologie de conception

Le Processus Unifié (UP pour UnifiedProcess) est un processus générique de développement logiciel construit sur UML. Générique décrit qu'il est nécessaire d'ajuster UP au contexte du projet à réaliser. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels et à différents domaines d'applications [18].

III.5.1 Le formalisme

UML, est l'acronyme anglais pour « Unified Modeling Language ». On le traduit par « Langage de modélisation unifié ». La notion UML est un langage visuel constitué d'un ensemble de schéma, appelés des diagrammes qui donnent chacun une vision différente du projet à traiter. UML nous fournit donc des diagrammes pour représenter le logiciel à développer : son fonctionnement, sa mise en route et les actions susceptibles d'être effectuées par le logiciel [19].

III.5.2 Présentation de l'UML

UML dans sa 2ème version (UML 2.0) propose treize (13) diagrammes pouvant être utilisés dans la description d'un système. Ces diagrammes sont regroupés dans deux grands ensembles [20] :

- Les diagrammes structurels.
- Les diagrammes de comportement.

III.6 Identification des acteurs

Les différents acteurs de notre système sont :

- **Administrateur** : c'est la personne chargé de gérer les utilisateurs qui peuvent accéder au system.

- **Étudiants** : C'est les étudiants de l'université de Bejaia, qui peuvent soit ajouter des fichiers, consulter ou supprimer des fichiers dans la plateforme et modifier leurs compte.

III.7 Identification des cas d'utilisations

III.7.1 La liste des cas d'utilisation

Pour chaque acteur identifié précédemment, il convient de rechercher les différentes interactions possibles selon le système qui sera mis en place.

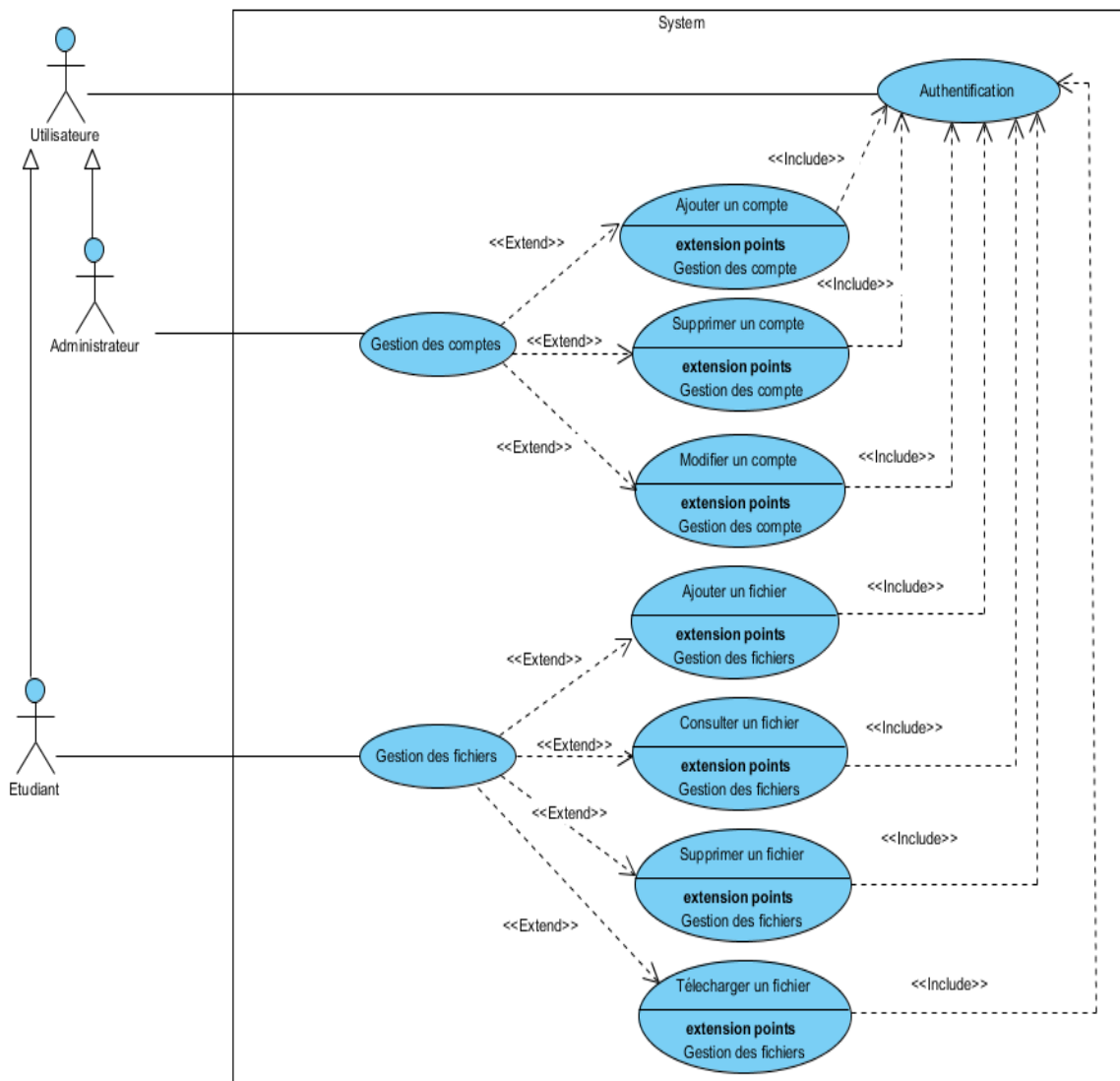


FIGURE III.2 – diagramme de cas d'utilisation.

III.7.2 Description des cas d'utilisation

La description des cas d'utilisations prend souvent une forme rédigée qui convient mieux à la Communication avec les utilisateurs. Des règles de structuration doivent être appliquées pour en faciliter l'expression, la compréhension et la cohérence.

III.7.2.1 La description de cas d'utilisation « authentification »

Le cas d'utilisation authentification permet à chaque acteur d'accéder à son espace personnel.

Cas d'utilisation	Authentification
Acteurs	Administrateur, étudiant
Objectif	Ce cas d'utilisation permet à un utilisateur de se connecter au système
Scénario nominal	<ul style="list-style-type: none"> ◇ Se connecter à l'application. ◇ Le système affiche l'interface d'authentification. ◇ L'utilisateur saisi son nom de l'utilisateur et son mot de passe. ◇ Le système vérifié la conformité des informations saisies puis valider. ◇ Le système récupère l'animation du mot traduite dans l'ontologie. ◇ Le système affiche le résultat à l'utilisateur
Le cas d'erreur	<ul style="list-style-type: none"> ◇ Dans le cas où les informations saisies sont fausses le système affiche l'interface d'authentification et attend que l'utilisateur resaisisse ses informations. ◇ Le système donne l'accès à l'interface correspondante.

TABLE III.1 – La description de cas d'utilisation « authentification »

III.7.2.2 La description de cas d'utilisation « gérer un compte »

Le cas d'utilisation gérer un utilisateur permet à l'administrateur de mettre à jour (ajouter, modifier, supprimer) les utilisateurs.

Cas d'utilisation	Administrateur
Objectif	Mettre à jour la liste des utilisateurs (ajouter, modifier, supprimer).
Précondition	S'authentifier
Scénario nominal	<ul style="list-style-type: none"> ◇ Le système affiche l'interface de gestion des utilisateurs. ◇ L'administrateur établit les modifications voulues et valide. ◇ Le système enregistre la modification.
Le cas d'erreur	<ul style="list-style-type: none"> ◇ Si l'email ou User Name est déjà utilisé, une erreur se déclenche et elle va signaler que les champs sont réservés.

TABLE III.2 – La description de cas d'utilisation « Gérer un compte »

III.7.2.3 La description de cas d'utilisation « gérer un fichier »

Le cas d'utilisation gérer un fichier permet à l'étudiant de mettre à jour (ajouter, consulter, supprimer) les fichiers ajoutés par les étudiants.

Cas d'utilisation	Etudiant
Objectif	Mettre à jour la liste des fichiers (Ajouter, modifier, supprimer).
Précondition	S'authentifier
Scénario nominal	<ul style="list-style-type: none">◇ L'étudiant demande l'interface de gestion des fichiers.◇ Le système affiche l'interface de gestion de fichier.◇ L'étudiant établit les modifications voulues et valide.◇ Le système enregistre les modifications.
Le cas d'erreur	<ul style="list-style-type: none">◇ Si l'étudiant laisse un champ vide lors de l'ajout d'un fichier.

TABLE III.3 – La description de cas d'utilisation « Gérer un fichier »

III.8 Diagramme de séquence

Pour chaque action dans le système il y'a beaucoup d'interaction entre les utilisateurs et le système.

III.8.1 Diagramme de séquence «Authentification»

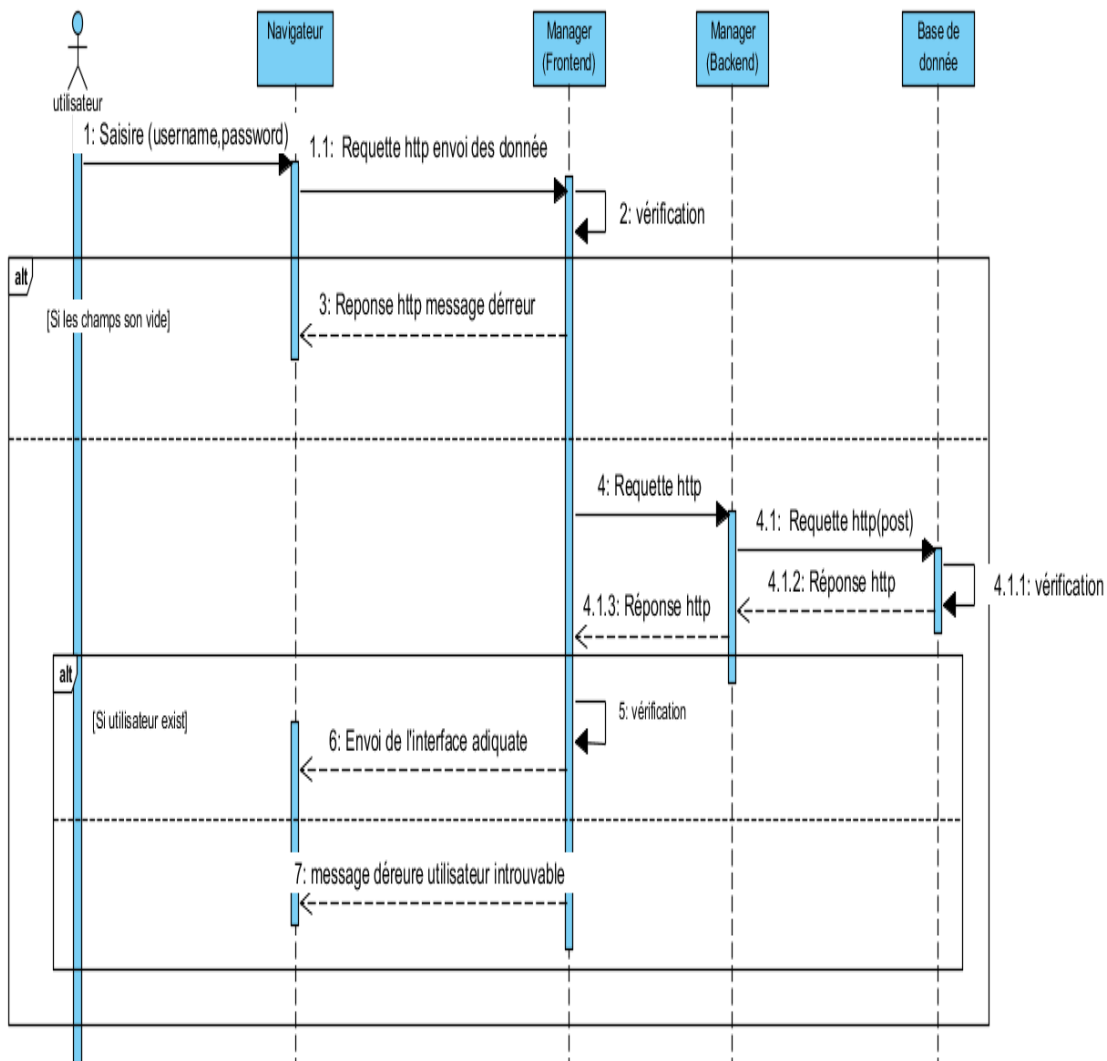


FIGURE III.3 – Diagramme d séquence Authentification.

III.8.2 Diagramme de séquence «option ajouter un fichier »

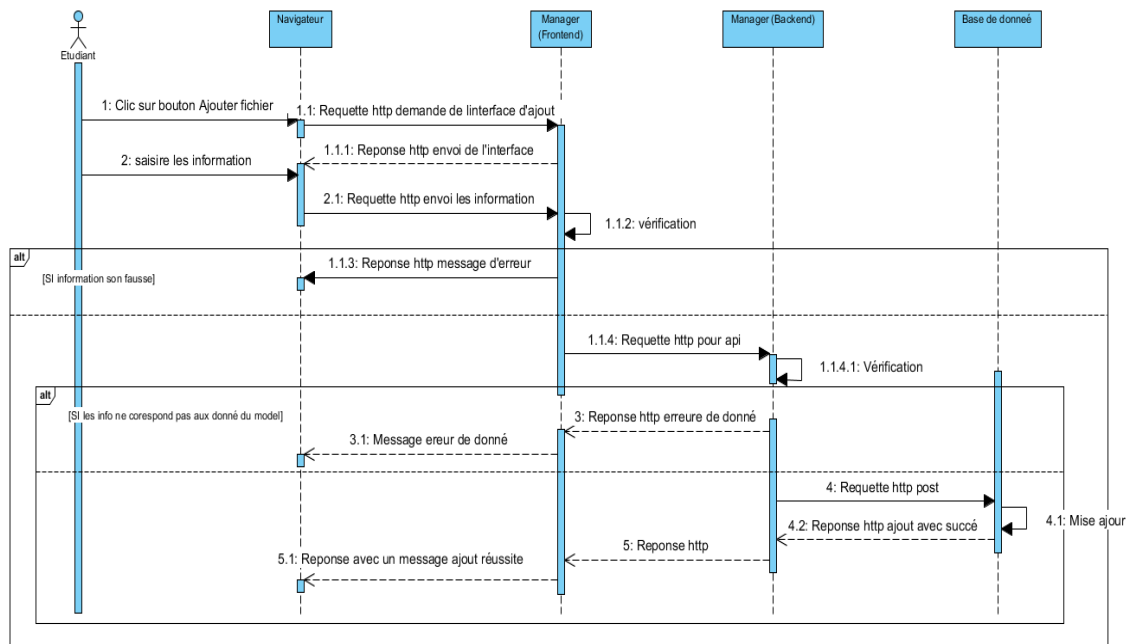


FIGURE III.4 – Diagramme de séquence «option ajouter un fichier».

III.8.3 Diagramme de séquence «option consulter un fichier»

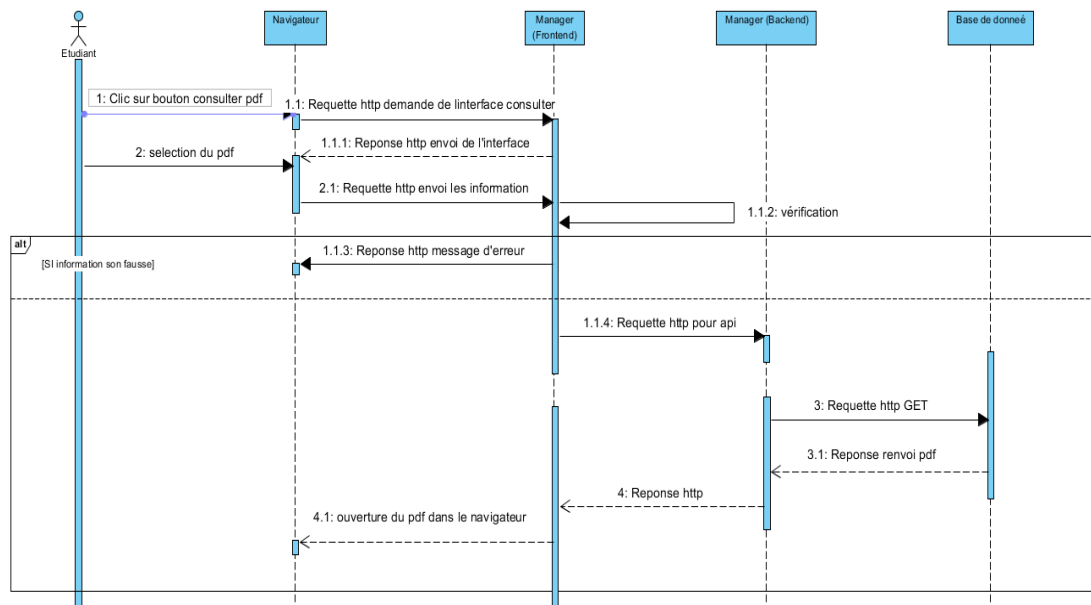


FIGURE III.5 – Diagramme de séquence «option consulter un fichier».

III.8.4 Diagramme de séquence «option supprimer un fichier»

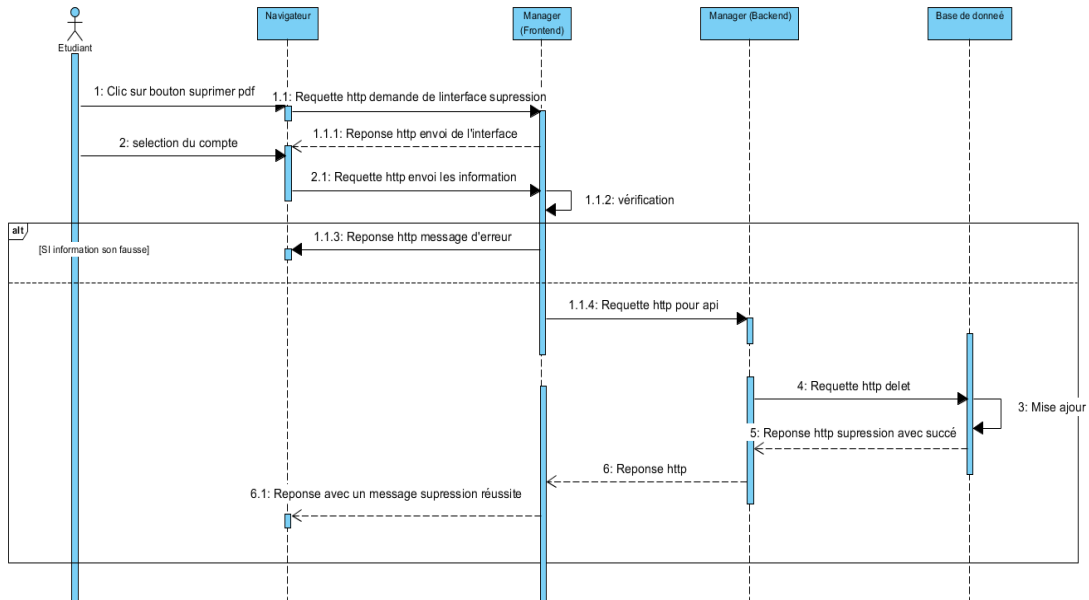


FIGURE III.6 – Diagramme de séquence «option supprimer un fichier».

III.8.5 Diagramme de séquence «gestion des fichier»

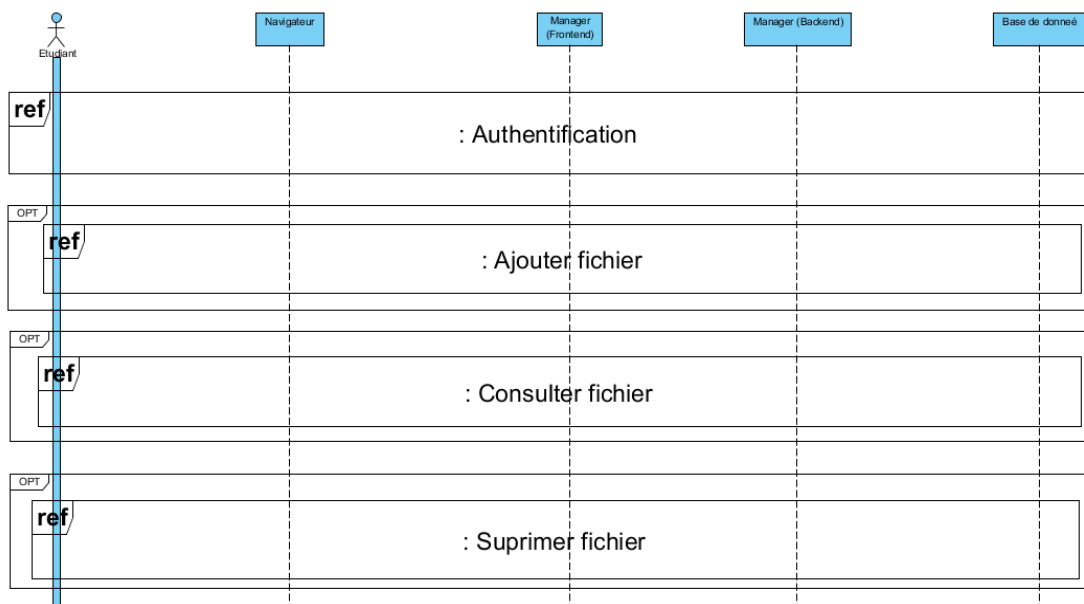


FIGURE III.7 – Diagramme de séquence «gestion des fichier».

III.8.6 Diagramme de séquence «option ajouter un compte»

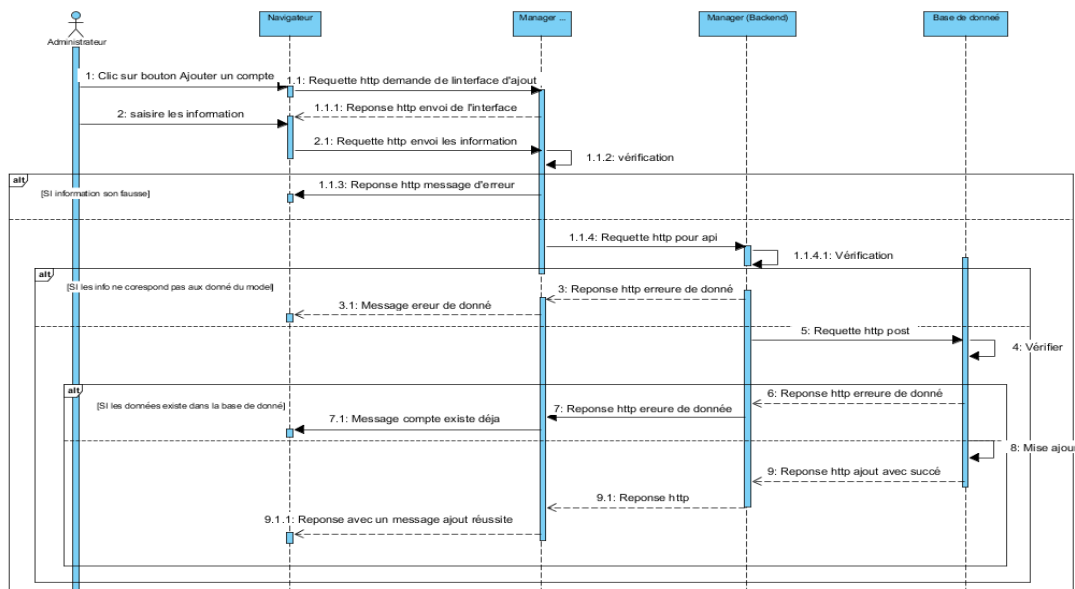


FIGURE III.8 – Diagramme de séquence «option ajouter un compte».

III.8.7 Diagramme de séquence «option modifier un compte»

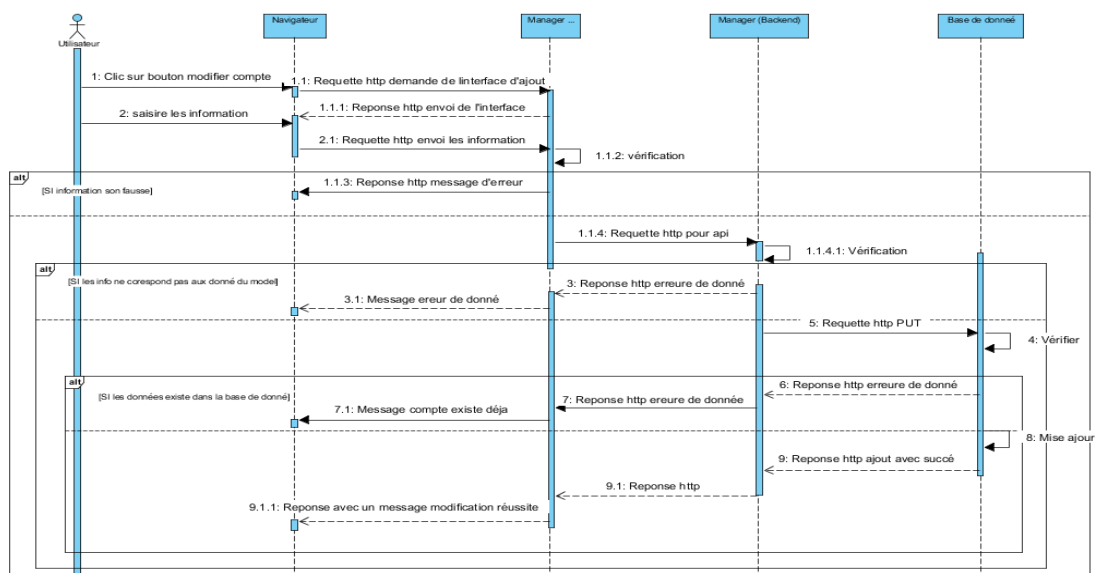


FIGURE III.9 – Diagramme de séquence «option modifier un compte».

III.8.8 Diagramme de séquence «option supprimer un compte

»

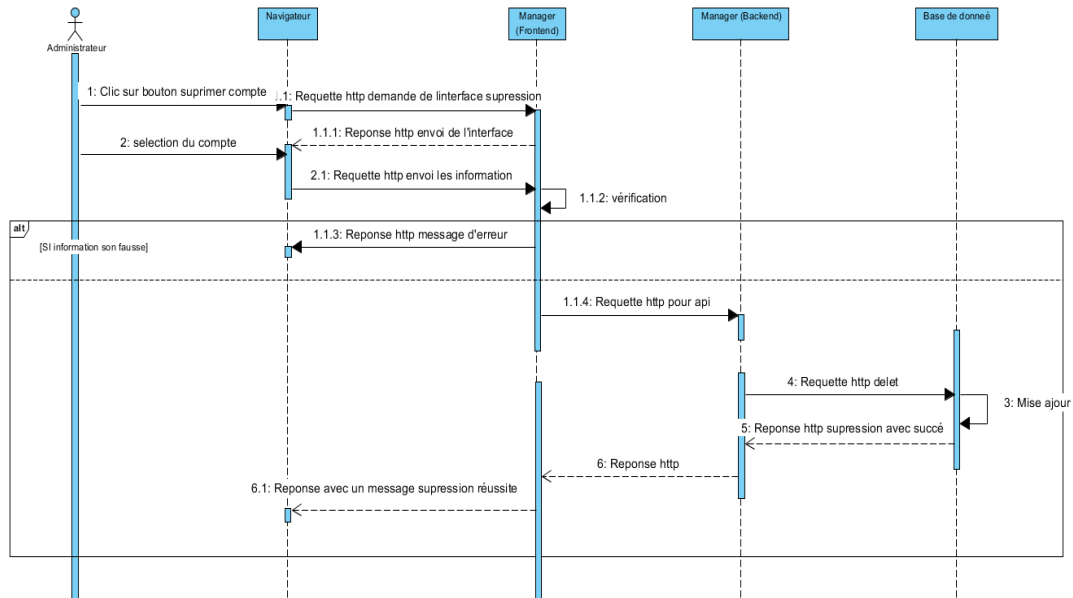


FIGURE III.10 – Diagramme de séquence «option supprimer un compte ».

III.8.9 Diagramme de séquence «gestion des comptes»

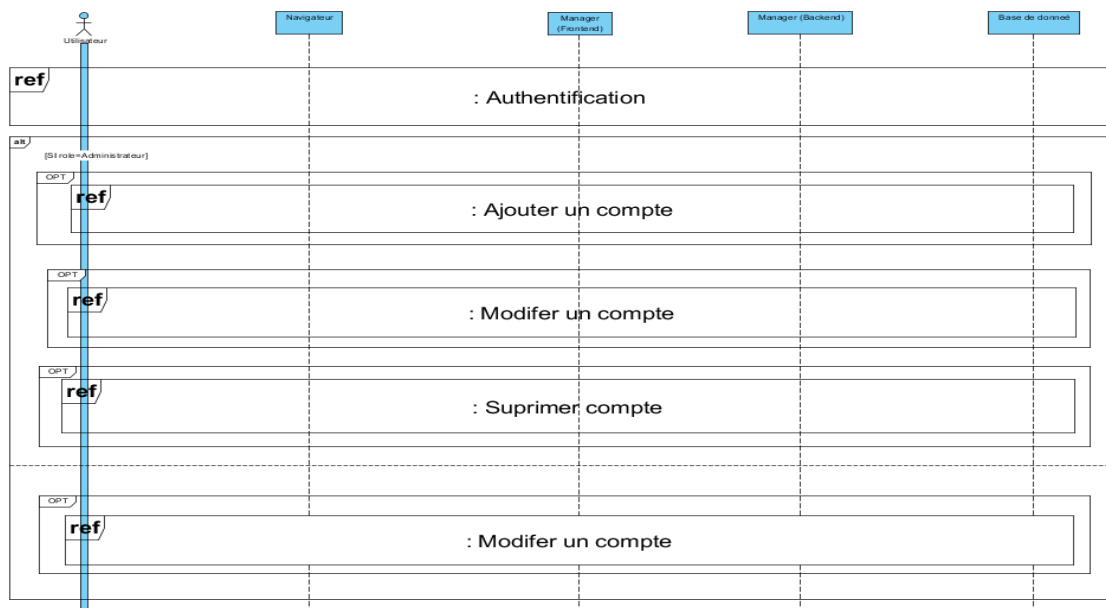


FIGURE III.11 – Diagramme de séquence «gestion des comptes».

III.9 Diagramme de classe

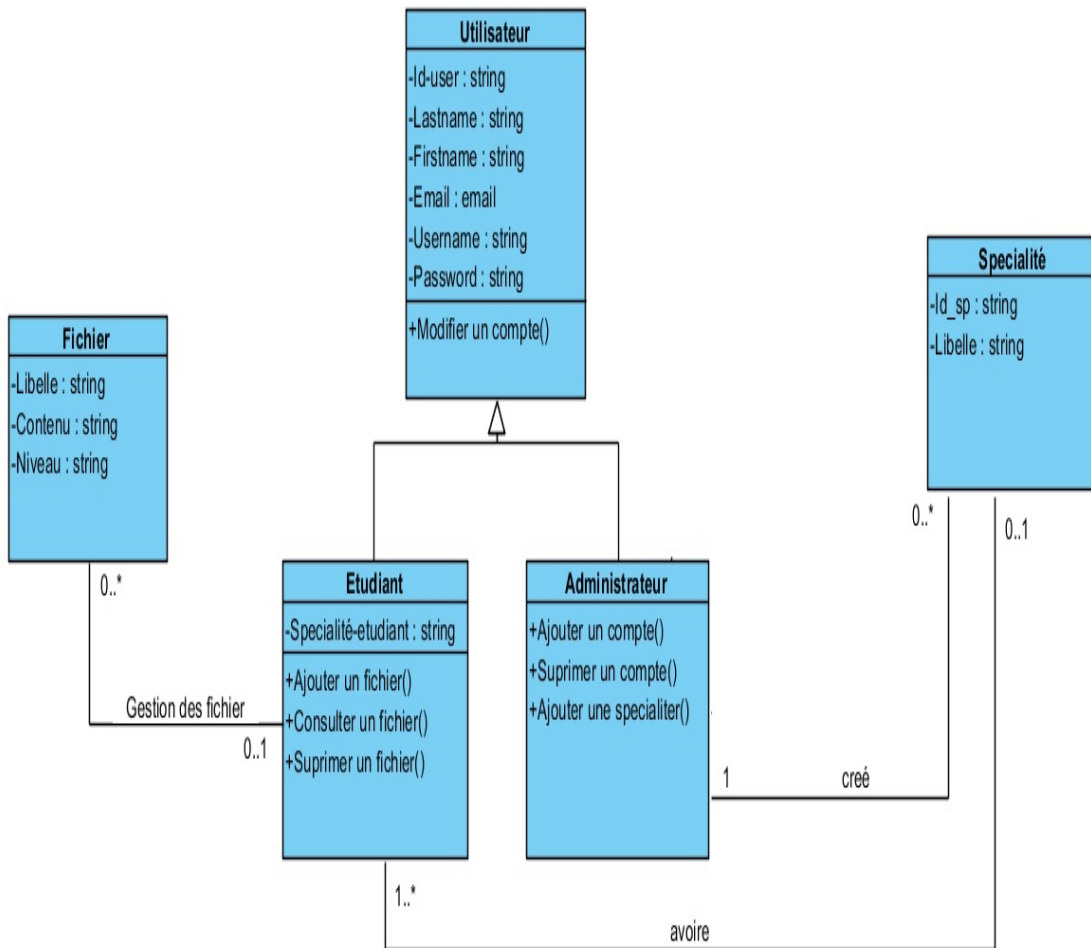


FIGURE III.12 – Diagramme de classe.

III.10 Dictionnaire des données

Classe	Code de la donnée	Désignation	Type	Taille	Méthode
Utilisateur	Id-user	Identifiant de l'utilisateur	String	50	Modifier un compte()
	Lastname	Prénom de l'utilisateur	String	50	
	Firstname	Nom de l'utilisateur	String	50	
	Email	Email de l'utilisateur	email	50	
	Username	Pseudo de l'utilisateur	String	50	
	Password	Mot de passe de l'utilisateur	String	50	
Etudiant	Spécialité-étudiant	Spécialité à laquelle appartient l'étudiant	String	50	Ajouter un fichier() Consulter un fichier() Supprimer un fichier()
Administrateur					Ajouter un compte() Supprimer un compte() Ajouter une spécialité()
Spécialité	Id_sp	Identifiant de la spécialité	String	50	
	Libelle	Libelle de la spécialité	String	50	
	Libelle	Libelle du	String	150	
Fichier		fichier			
	Contenu	Contenu du fichier	String	1000	
	Niveau	Niveau auquel le fichier est destiné	String	50	

TABLE III.4 – Dictionnaire de données

III.11 Conclusion

Nous avons présenté dans ce chapitre la phase de conception de notre projet via les diagrammes de séquence, qui nous ont permis de décrire de manière globale et détaillée le fonctionnement désiré du système que nous avons implémenté. Nous avons recensé par la suite le diagramme de classe qui nous permet d'avoir le schéma de la base de données de l'application à réaliser.

Chapitre IV

Réalisation

IV.1 Introduction

La réalisation de notre application web a nécessité l'utilisation de plusieurs technologies spécifiques. Nous allons donc présenter notre environnement de développement, les différents langages de programmation, en précisant les différents outils auxquels nous avons eu recours, ainsi que les principales interfaces graphiques.

IV.2 Environnement de développement

Nous allons définir l'environnement de développement que nous avons utilisé pour la réalisation de notre application.

IV.2.1 Visual studio code



Visual Studio Code est un éditeur de code multiplateforme, open source et gratuit développé par Microsoft. Il est livré avec une prise en charge intégrée de JavaScript, TypeScript et Node.js et dispose d'un riche écosystème d'extensions pour d'autres langages (C++, C, Java, Python, PHP, Go) et des environnements d'exécution (tels que .NET et Unity)[21].

IV.2.2 Node js



Nodejs est un environnement d'exécution JavaScript asynchrone piloté par des événements, il est conçu pour créer des applications réseau évolutives.

Parmi les modules natifs de Node.js, on retrouve http qui permet le développement de serveur HTTP. Ce qui autorise, lors du déploiement de sites internet et d'applications web développés avec Node.js, de ne pas installer et utiliser des serveurs webs tels que Nginx ou Apache [22].

IV.2.3 Mongo Atlas



MongoDB Atlas est le service de base de données cloud mondial pour les applications modernes. Déployez MongoDB entièrement géré sur AWS, Azure ou GCP. Une automatisation de pointe et des pratiques éprouvées garantissent la disponibilité, l'évolutivité et la conformité aux normes de sécurité et de confidentialité des données les plus exigeantes[23].

IV.2.4 Visual Paradigm for UML

Visual Paradigm for UML est un logiciel de création de diagrammes, dans le cadre d'une programmation tout-en-un, il possède plusieurs options, permettant une large possibilité de modélisation en UML.

IV.2.5 Postman



Postman est une plate-forme d'API pour la création et l'utilisation d'API. Il simplifie chaque étape du cycle de vie des API et rationalise la collaboration afin de pouvoir créer de meilleures API, plus rapidement.

IV.3 Technologies logicielles

Nous allons définir le langage et technologies utilisé pour la réalisation de notre application[24].

IV.3.1 Javascript



JavaScript est un langage de programmation de scripts utilisé principalement dans les pages web interactives et c'est une partie essentielle des applications web. Avec les technologies HTML et CSS, JavaScript est considéré comme l'une des technologies cœur du World Wide Web.

IV.4 Organisation du code

L'architecture utilisée afin que le code de notre application soit bien détaillé nous avons séparé la logique du code en deux parties que l'on retrouve dans des fichiers distincts : Backend et frontend.

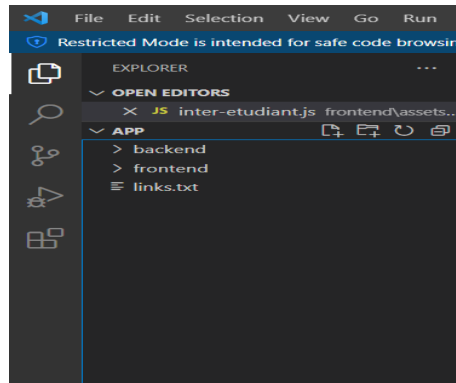


FIGURE IV.1 – Organisation du code.

IV.4.1 Backend

Le rôle de cette partie c'est gérer la communication entre la base de données mongo-atlas et notre application comme aussi dans cette partie on trouve des fichiers distincts : Controllers, middlewares, models, routes.

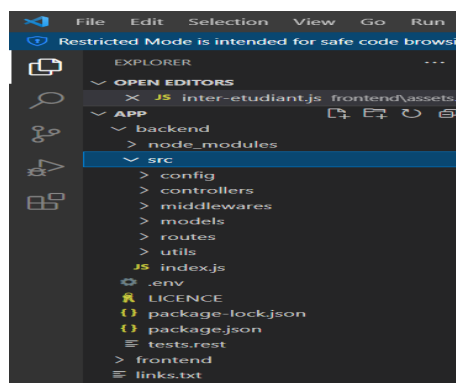


FIGURE IV.2 – Organisation du code **Backend**.

IV.4.2 Frontend

Le rôle de cette partie c'est gérer la communication entre le navigateur et le Backend de notre application comme aussi dans cette partie on trouve des fichiers distinct : assets, parials.

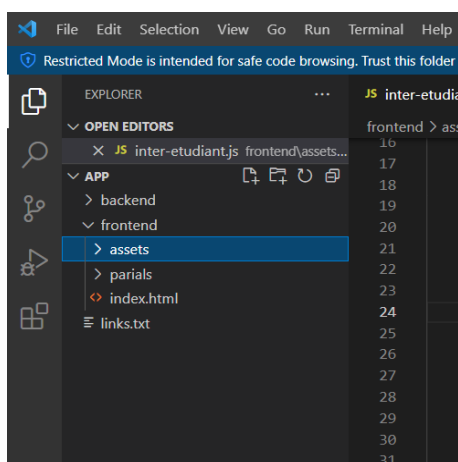


FIGURE IV.3 – Organisation du code **Frontend**.

IV.5 diagramme de déploiement

Le diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux

La figure IV.5 suivante illustre les Répartition des différentes machines physiques :

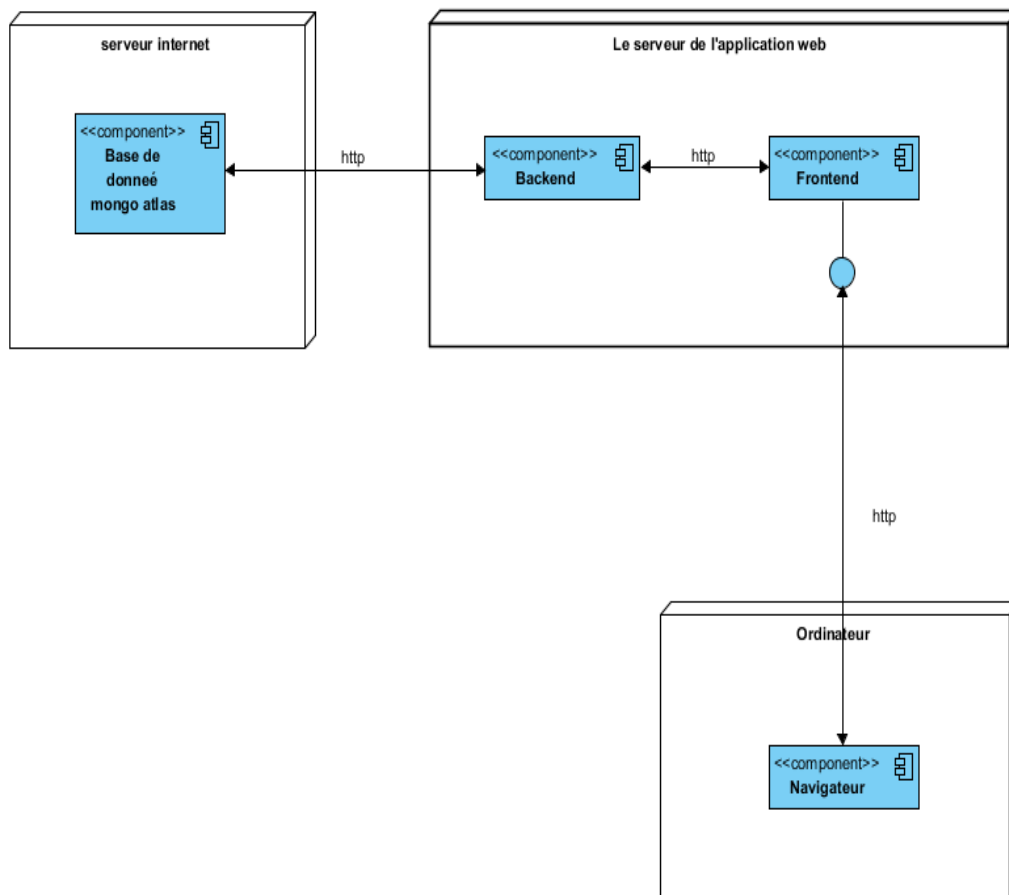


FIGURE IV.4 – diagramme de Déploiement

IV.6 Présentation des interfaces de l'application

Nous allons présenter dans cette partie les interfaces de notre application.

IV.6.1 interface d'Authentification

La figure IV.6.1. Représente l'interface d'authentification, où l'utilisateur devra saisir, username mot de passe, pour pouvoir accéder à son interface apropi.



FIGURE IV.5 – Interface d’authentification

IV.6.2 Interface principale de l’administrateur

La figure IV.6.2. Représente L’interface principale de l’administrateur ou en trouve les fonctionnalités Ajouter un compte, modifier un compte et supprimé un compte.



FIGURE IV.6 – Interface principale de l’administrateur

IV.6.3 Interface ajouter un compte

La figure IV.6.3 l’interface d’ajout d’un compte nous permettra d’ajouter des compte étudiants ou administrateurs en saisissons leur donnés et choisir le type du compte qui

seront stocké dans la base de donnée.

FIGURE IV.7 – Interface d’ajout des comptes

IV.6.4 Interface modifier compte

La figure IV.6.4 représente l’interface de modification d’un compte utilisateur après la sélection de l’utilisateur.

FIGURE IV.8 – Interface modifier compt

IV.6.5 Interface d'ajout de fichier

La figure IV.6.5 représente l'interface qui permet à l'étudiant d'ajouter un fichier pdf.

The screenshot shows a web interface titled "Interface étudiant". On the left, there is a vertical sidebar with four green buttons: "Ajouter un fichier", "Modifier un fichier", "Supprimer un fichier", and "Modifier les informations". The main content area is light blue and contains a form with the following elements: two input fields at the top labeled "Document Libellé" and "Description"; two dropdown menus labeled "Sélectionner un niveau" and "Sélectionner une spécialité"; a button labeled "Sélectionner un fichier"; and two buttons at the bottom labeled "Enregistré" and "Annuler".

FIGURE IV.9 – Interface d'ajout de fichier

IV.7 Conclusion

La phase de réalisation est importante dans le cycle de vie d'une application. Dans ce dernier chapitre nous avons présenté les étapes de réalisation de notre application mini cloud pour les étudiants de l'université de Bejaia et les différents outils de développements que nous avons utilisés.

Conclusion générale

Ce travail se résume en la conception et la réalisation d'un mini cloud pour les étudiants de l'université de Bejaïa.

Pour cela, nous avons créé un mini cloud vu la difficulté du thème nous avons réussi à implémenté une application web qui permet à des étudiants de déposer un type de document et de le retirer à la demande.

Ce travail nous a permis d'apprendre avec précision, les différentes étapes de la conception et réalisation d'une application web, qui peuvent être pratique et théorique. Le développement de cette application nous a permis aussi d'enrichir nos connaissances dans les différents langages et outils auxquels nous avons eu recours, à savoir visual studio code, UML, Mongo-db, ainsi que JavaScript, Html, Css. En ce qui concerne l'aspect humain, ce travail d'équipe nous a donné un aperçu sur la vie professionnelle, à mieux nous organiser dans notre travail, afin d'accomplir les tâches qui nous sont confiées dans les meilleures conditions et dans les plus brefs délais. Grâce à cette application, nous espérons faciliter la gestion des fichiers pour les étudiants pour un meilleur parcours universitaire.

Ceci n'est qu'un commencement pour la conception et la réalisation d'un véritable cloud pour toute la communauté estudiantine de Bejaia. Les perspectives sont donc les suivantes :

- Elargir le mini-cloud pour qu'il puisse gérer plusieurs types de documents, et un grand nombre d'utilisateurs.
- Utilisé des protocoles plus performant pour la sécurité des données.
- Mettre en place une méthode pour la réplication de données afin de garantir la disponibilité des données en cas de panne dans un serveur de données.

Bibliographie

- [1] D.Martinez. Privacy and confidentiality issues in cloud computing architectures. Mémoire de master en informatique, Université polytechnique, Catalogne, 2013.
- [2] Jean-Pierre Briffaut, François Stéphan, Cloud computing évolution technologique révolution des usages, ISBN 978-2-7462-4511-2, Lavoisier, Paris, 2013.
- [3] Le cloud computing une nouvelle filiere fortement structurante. Direction regionale des entreprises, de la concurrence, de la consommation du travail et de l'emploi (DIRECCTE) ILE-DE-FRANCE, septembre 2012.
- [4] R.Bohn, J.Messina, F.Liu, J.Tong, and J.Mao. NIST :cloud computing reference architecture. Rapport de recherche, National Institute of Standards and Technology, Etats-Unis, 2011.
- [5] Z.Aouameur and H.Tahrine. Comparaison et mise en place des plateformes de cloud computing : Openstack et eucalyptus. Mémoire master académique en informatique industrielle, Université Kasdi Merbah, Ouargla, 2013.
- [6] Redhat,data-storage [en ligne] consulté le [mois de Mars 2021] Disponible sur : <https://www.redhat.com/fr/topics/data-storage/what-is-cloud-storage>
- [7] [https://www.unedose.fr/article/network-storage-solutions-explained-cloud-vs--nas-vs--san-vs--das](https://www.unedose.fr/article/network-storage-solutions-explained-cloud-vs-nas-vs-san-vs-das) consulté le [mois de Juin 2021]
- [8] J. F Audenard.Comprendre la protection des données dans le cloud. Publication Orange Business Services, 17 Mars 2011.
- [9] <https://fr.sawakinome.com/articles/security/difference-between-ipsec-and-ssl.html> consulté le [mois de Avril 2021]
- [10] <https://www.spoonylife.org/algorithms-and-computation/methodes-de-clustering> consulté le [mois de Juin 2021].

- [11] L. M. Vaquero, L. Rodero-Merino, Finding your Way in the Fog : Towards a Comprehensive Definition of Fog Computing, ACM SIGCOMM Computer Communication Review Volume 44 Issue 5 October 2014 pp 27–32, 2014.
- [12] C. Pierkot. Gestion de la Mise Jour de Données Géographiques Répliquées. PhDthesis, Université Toulouse III - Paul Sabatier, Juillet 2008.
- [13] Opennebula, Architecture opennebula [en ligne], consulté le [mois de Juin 2021] <http://opennebula.org/>.
- [14] Wikipedia, Architecture Eucalyptus [en ligne], consulté le [mois de Juin 2021] [https://en.wikipedia.org/wiki/Eucalyptus_\(software\)](https://en.wikipedia.org/wiki/Eucalyptus_(software))
- [15] Redhat, Architecture openstack [en ligne], consulté le [mois de Juin 2021] <https://www.redhat.com/fr/topics/openstack>
- [16] consulté le [mois de Juin 2021] <https://philippe.developpez.com/articles/azure-introduction/> consulté le [mois de Juin 2021]
- [17] <https://aws.amazon.com/fr/ec2/?ec2-whats-new.sort-by=item.additionalFields.postDateTime&ec2-whats-new.sort-order=desc> consulté le [mois de Juin 2021]
- [18] <https://www.fichier-pdf.fr/2012/01/13/cours-cours-uml/> consulté le [mois de juillet 2021]
- [19] <https://www.dunod.com/sciences-techniques/uml-2-analyse-et-conception-mise-en-oeuvre> consulté le [mois de juillet 2021].
- [20] https://www.academia.edu/9648899/Modélisation_UML_Diagrammes_Structurels_Diagramme_des_Classes consulté le [mois de juillet 2021].
- [21] <https://code.visualstudio.com/docs> consulté le [mois de septembre 2021].
- [22] <https://nodejs.org/en/about/> consulté le [mois de septembre 2021].
- [23] <https://www.mongodb.com/fr-fr/cloud/atlas> consulté le [mois de septembre 2021].
- [24] <https://www.postman.com/> consulté le [mois de septembre 2021].

Résumé

Ce travail réalisé dans le cadre d'un projet de fin de cycle, contient l'étude, la conception et la réalisation d'une application web, qui permet la gestion le stockage et le partage des fichiers pour les étudiants de l'université de Béjaïa. Ce mémoire contient une étude sur le cloud computing en général, son fonctionnement et son architecture en particulier. La réalisation de ce projet a nécessité une bonne analyse conceptuelle, nous avons donc approfondi nos connaissances en UML, afin d'aboutir à un meilleur résultat lors de la réalisation de notre application.

Mots clés : application web, fichier, UML, JavaScript, node.js, mongo-db, cloud computing, plateforme cloud.

Abstract

This work, carried out as part of an end-of-cycle project, contains the study, design and production of a web application, which allows the management, storage and sharing of files for students of the University of Béjaïa. This thesis contains a study on cloud computing in general, its operation and its architecture in particular. The realization of this project required a good conceptual analysis, so we deepened our knowledge in UML, in order to achieve a better result When making our application.

Keywords : web application, file, UML, JavaScript, node.js, mongo-db, cloud computing, cloud platform..