UNIVERSITE DE YAOUNDE I
****
ECOLE NATIONALE SUPERIEURE POLYTECHNIQUE
****
DEPARTEMENT DE GENIE INFORMATIQUE

UNIVERSITY OF YAOUNDE I
****
NATIONAL ADVANCED SCHOOL OF ENGINEERING
****
DEPARTMENT OF COMPUTER ENGINEERING

Élaboration d'une politique de sécurité pour un ERP de type SaaS : Cas de Mobility

Mémoire de fin d'études/Master of Engineering

Présenté et soutenu par :
MASSAGA NGONDA ARISTIDE MARIE

En vue de l'obtention du :
Diplôme d'Ingénieur de Conception de Génie Informatique

Sous la Direction de :

Dr KOUAMOU Georges Édouard
Ing. CHOPGWE ALAH Leonard

Devant le jury composé de :

Président : **Pr TANGHA Claude**
Rapporteur : **Dr KOUAMOU Georges Édouard**
Examinateur : **Dr NANA MBINKEU Rodrigue Carlos**
Invité : **Ing. CHOPGWE ALAH Leonard**

Année académique 2016 - 2017
Mémoire soutenu le 17 juillet 2017

Dédicace

Je dédie ce travail à ma **Famille**.

Remerciements

Ce travail est l'aboutissement de nombreux efforts et sacrifices et n'aurait jamais été accompli sans l'aide et le soutien des personnes suivantes :

- Le Pr TANGHA Claude qui me fait l'honneur de présider ce jury ;
- Le Dr NANA MBINKEU Rodrigue Carlos pour avoir accepté d'examiner ce travail ;
- Le Dr KOUAMOU Georges Edouard, mon encadrant académique et enseignant à l'Ecole Nationale Supérieure Polytechnique, pour son encadrement, sa disponibilité, ses conseils, et toute l'attention qu'il a portée à l'accomplissement de ce travail ;
- Mon encadrant professionnel, Ing. CHOPGWE ALAH Leonard pour son sens poussé de la méthodologie et de la rigueur, qui sont les qualités premières d'un ingénieur ;
- Toute l'équipe d'IniMov : SOFACK Armand le Directeur Général, Ing. TONFEU Perez, Ing. TEGANTCHOUANG Boris, Ing. KOUNGUE Clotaire,Mme NDONG-SONG Estelle, Mme MUNDI Vernice, pour leur accompagnement, leurs conseils, leur haute disponibilité, leur convivialité tout au long du stage.
- Tous les enseignants de l'ENSP, particulièrement ceux du département de Génie Informatique, pour leur dévouement et les enseignements qu'ils m'ont prodigués tout au long de ces trois (03) dernières années.

Je tiens également à adresser mes sincères remerciements :

- A mes parents MASSAGA Emmanuel et AKAMBA Joséline, à mes frères et sœurs : Gwladys, Yannick, Pierrette, Emmanuelle, Marc, Audrey et à toutes les grandes familles MASSAGA et NDENGUE pour tout leur soutien, leurs aides, leurs conseils ;
- A mes co-stagiaires MBOUOMBO Modeste et MEMA Cissé avec qui j'ai passé de bons moments en stage ;
- A mes amis et camarades de la promotion 2017 du Génie Informatique de l'École Nationale Supérieure Polytechnique de Yaoundé pour ces agréables moments passés ensemble ;
- A mes amis, de la paroisse Sainte Thérèse de l'Enfant Jésus de Nkol-Nguié, pour leurs conseils, leur attention et leur sollicitude.

A toutes les personnes qui de près ou de loin ont contribué au bon accomplissement de ce stage académique, et dont je n'ai pas mentionné le nom, vous qui vous souciez de moi dans le meilleur comme dans le pire, un grand **MERCI !**

Glossaire

IniMov	*Innovation In Motion*
SaaS	*Software as a Service (Logiciel en tant que service)*
PaaS	*Platform as a Service (Plate-forme en tant que service)*
IaaS	*Infrastructure as a Service (Infrastructure en tant que service)*
ERP	*Enterprise Resource Planning*
SOA	*Service Oriented Architecture (Architecture Orientée Service)*
MVC	*Modèle Vue Contrôleur*
REST	*REpresentational State Transfer*
URI	*Uniform Resource Identifier*
URL	*Uniform Resource Locator*
HTTP(S)	*HyperText Transfer Protocol(Secure)*
ACL	*Access Control List*
API	*Application Programming Interface (Interface de Programmation Applicative)*
IHM	*Interface Homme Machine*
TLS	*Transport Layer Security*
ISO	*International Organization for Standardization*
IP	*Internet Protocol*
TCP	*Transmission Control Protocol*

Résumé

Le désir d'automatiser, de contrôler et d'optimiser le fonctionnement des entreprises, a conduit à la création d'une nouvelle catégorie de logiciel que sont les ERP. Toutefois, ce désir reste conditionné par les coûts (acquisition, formation et maintenance). Le cloud computing vient donc apporter une solution à ce problème : premièrement par la virtualisation et la mutualisation des ressources (matériel et logiciel) ; deuxièmement par le paiement à l'usage et troisièmement en les rendant accessible via le web (partout et à tout moment). Cependant, cette virtualisation, cette mutualisation et cette accessibilité via internet posent de nombreux problèmes de sécurité qui se résument en la confidentialité, l'intégrité et la disponibilité ; à cela on peut ajouter la protection des données stockées et celles qui circulent sur le réseau ; la protection du code contre les injections de codes malveillants.

Ce travail de mémoire, a pour objectif l'élaboration d'une politique de sécurité, pour la protection de l'ERP Mobility que développe l'entreprise IniMov avant sa mise sur le marché. Pour cela, nous commençons par réaliser un état de l'art de la recherche sur la sécurité dans le cloud et particulièrement des applications qui y sont déployées, puis une analyse du fonctionnement et des fonctionnalités de Mobility (afin de ressortir les éléments à risque), nous continuons par une description de la politique élaborée et terminons par la présentation des résultats obtenus.

Mots clés : ERP, cloud computing, sécurité.

Abstract

The desire to automate, control and optimise the functioning of companies has led to the creation of a new category of software called ERP. However, this desire is conditioned by different costs (acquisition, training and maintenance). Thus, cloud computing brings a solution to this problem in several ways : firstly, through virtualization and mutualisation of ressources (hardware and software) ; moreover through the payment of usage ; and lastly to rendering then accessible via the web (everywhere and at any moment). However this virtualization, mutualisation and accessibility via internet causes a lot of security problems that can be summed up to confidentiality, integrity and availability. To this, we can add saved data protection and web flowing data, and the protection of the code against the insertion of malevolent codes.

This work aims at elaboration a security policy which will secure ERP Mobility which is developed by IniMov company before its commercialization .With regard to this objective, we first of all, set up state of research art on the security in cloud particularly with applications that are displayed. Moreover, we put in place an analysis of the functioning and fonctions of Mibility (so as to sort exit unsafe elements). For the more, we discounted the policy elaborated. Lastly, presented the results obtained.

Keys words : ERP, cloud computing, security.

Table des figures

Liste des tableaux

Table des matières

Introduction générale

Contexte

Le développement d'Internet a permis l'émergence de nouvelles technologies permettant aux entreprises de simplifier leurs fonctionnements tout en améliorant leurs performances et en réduisant les coûts : Le **Cloud Computing** est l'une de ces technologies. Il consiste, pour une entreprise dite fournisseur de Cloud, à externaliser une partie de ses ressources matérielles et logicielles de sorte que des organisations ou des particuliers puissent y accéder, effectuer leurs traitements et payer ce qu'ils ont utilisé. Parmi les caractéristiques principales du cloud se trouvent : paiement à l'usage et la virtualisation dont les corollaires sont la scalabilité, la mutualisation, l'abstraction matérielle. Les ressources proposées comprennent des infrastructures (IaaS), des plateformes de développement et d'exécution (PaaS) ou des applications (SaaS). Pour une organisation ayant choisi de souscrire à un service Cloud ou à toute autre application pour son fonctionnement quotidien, garantir la sécurité c'est-à-dire la confidentialité, l'intégrité et la disponibilité de ses informations reste un défi majeur.

Afin d'offrir aux entreprises camerounaises un moyen simple, efficace, sécurisé et faiblement couteux de gérer leurs business, l'entreprise IniMov a initié le projet de développement d'un ERP adapté au contexte local. Cette application sera offerte aux entreprises sous forme de service de type SaaS offrant différentes fonctionnalités indépendantes pour la gestion informatique de l'entreprise. Pour cela, il faut être capable : d'assurer la haute disponibilité des données et des informations ; de protéger ses données d'attaques visant à les corrompre ou y accéder ; de garantir qu'une organisation n'accédera qu'aux ressources ou fonctionnalités auxquelles elle a, au préalable, souscrit ; de permettre aux organisations d'implémenter, dans le système, l'aspect de sa politique de sécurité visant à garantir la confidentialité et l'intégrité des informations au moyen de la séparation des privilèges et du contrôle d'accès.

Problématique

Le Cloud computing est une révolution économique et une évolution technologique, qui repose sur l'accessibilité des ressources informatiques en tant que service via internet. Il propose une meilleure solution aux organisations pour gérer les données, les infrastructures et les applications. Cependant, le Cloud peut être la cible d'attaques externes (réseau) ou internes (employés) qui ont pour but d'interrompre, d'intercepter, de modifier, de copier et de fabriquer des informations. La sécurisation des données en transit dans le Cloud reste un challenge car elle doit se faire en tenant compte de ses trois principales couches : la couche infrastructure, la couche plateforme et la couche applicative. Au niveau des couches infrastructures et plateformes de développement et d'exécution, le fournisseur de l'infrastructure cloud met déjà sur pied des mécanismes aidant à garantir la confidentialité des données, l'intégrité et la disponibilité. Parmi ces mécanismes nous avons : la virtualisation (isolation), la réplication des données, la segmentation en domaines et sous domaines, etc.

Cette étude se concentre sur la couche applicative : il sera question pour nous de faire une analyse des politiques de sécurité existantes de manière générale dans le cloud, puis d'élaborer une qui convient mieux au système Mobility.

Objectifs

L'objectif global de ce travail de mémoire est de pourvoir Mobility d'une politique de sécurité adaptée. Celui-ci se décline en trois objectifs spécifiques :

— L'évaluation des risques et leur niveau de criticité, il s'agit de répondre aux questions « quels risques et quelles menaces, sur quelles données et quelles activités, avec quelles conséquences ? » ;
— La recherche et la sélection de parades, il s'agit de répondre aux questions « que va-t-on sécuriser, quand et comment ? » ;
— La mise en œuvre des protections dans le système, et la vérification de leur efficacité.

Plan du mémoire

Le début du mémoire est consacré à la réalisation de l'état de l'art. Ainsi, le Chapitre 1 donne l'état de la recherche sur la sécurité dans le cloud et particulièrement des applications qui y sont déployées et se termine par une présentation de Mobility qui est notre système d'implémentation et de test. Le cœur du document présente, en détails, la contribution du mémoire. Ainsi le Chapitre 2 se concentre sur l'élaboration de la politique de sécurité de

Mobility. Nous y menons, en particulier, une étude des risques qui pèsent sur le système et proposons des mesures pour y faire face. La fin du mémoire, à travers le Chapitre 3, présente les outils importants utilisés pour l'implémentation des mesures décrites au chapitre 2 et quelques résultats obtenus. La conclusion générale clôt le mémoire tout en présentant les perspectives d'amélioration futures à partir de notre travail.

Chapitre 1

Etat de l'art

Ce chapitre présente une étude synthétique de l'état de l'art dans les domaines du cloud computing (concepts, types, acteurs), de la sécurité dans le cloud (sûreté, sécurité, risques) et des mécanismes mis en œuvre pour solutionner ces problèmes et maintenir la confiance en le cloud. Nous terminons par une présentation de Mobility.

1.1 Le Cloud Computing

Le Cloud Computing est un concept assez récent, né dans les années 2002 au sein de l'entreprise Amazon. Il est peut-être défini comme étant un ensemble de ressources/applications/services s'exécutant dans un environnement distribué, accessible via les protocoles web standards, et dont l'ensemble fournit un service ayant les caractéristiques suivantes :

- Paiement à l'usage : les systèmes de cloud surveillent et rapportent automatiquement l'utilisation des ressources par des moyens appropriés au type de service ;
- Elasticité rapide ou scalabilité : les ressources peuvent être allouées et libérées de manière élastique et automatique ;
- Abstraction de l'infrastructure matérielle : les services sont accessibles depuis le réseau internet par des équipements traditionnels, fixes ou mobiles ;
- Mutualisation entre plusieurs utilisateurs : les ressources informatiques du fournisseur sont regroupées et utilisées pour servir plusieurs clients selon un modèle de co-résidence, avec des ressources dynamiquement allouées selon les demandes.

Pour assurer les quatre caractéristiques fondamentales du cloud, de nombreuses technologies sont utilisées : la principale est la virtualisation, qui est un concept né dans les années 1960, autour des travaux d'IBM sur les *mainframe* ; il s'agit d'une technique permettant d'exécuter plusieurs systèmes d'exploitation sur une même machine physique grâce à des procédés de segmentation des ressources matérielles, physiques, temporelles et spatiales entre plusieurs « **machines virtuelles** ».

Il existe différents modèles de Cloud se distinguant sur deux principaux critères : selon la propriété et selon le service.

- Selon la propriété : ici il s'agit du lien qui existe entre le propriétaire de la plateforme et l'utilisateur
  - Cloud communautaire : il est mis sur pied par une communauté et ne fournit ses services qu'aux membres de celle-ci. Ex : UnivCloud.
  - Cloud privé : il est mis sur pied par une organisation (entreprise ou gouvernement) et ne fournit ses services qu'aux membres de cette organisation.
  - Cloud public : il est mis sur pied à des fins commerciales et donc il est accessible à tous. Ex : Amazon web services, Microsoft Azure, etc.
  - Cloud hybride : combinaison entre un cloud privé et plusieurs clouds publics.
- Selon le service : ici il s'agit du type de service qu'offre la plateforme de cloud.
  - Infrastructure (IaaS) : le cloud fournit un service de stockage et de calcul. On y loue des machines virtuelles. Ex. : Amazon EC2, Windows Azure.
  - Plateforme (PaaS) : le cloud fournit une plate-forme de construction et d'exécution d'applications dans l'infrastructure sous-jacente. Ex. : Google App Engine, Windows Azure web role.
  - Application (SaaS) : le cloud fournit directement l'application dont a besoin

l'utilisateur. Ex. : Google docs, SalesForce.

FIGURE 1.1 – Présentation des services du cloud computing [27]

Autour du Cloud évolue quatre types d'acteur intervenant à différents niveaux pour offrir différents services :

- Fournisseur de cloud : fournit une infrastructure matérielle et un ensemble de services au-dessus. Ex. : Amazon Web Services (AWS), Microsoft Azure, CloudWatt ;
- Utilisateur du cloud : utilise directement les ressources de la plateforme de cloud. Ex. : NASA/JPL, étudiant ;
- Receleur de cloud construit et vend des services en s'appuyant sur des plateformes de cloud dont il n'a pas la propriété. Il est utilisateur du Cloud sur lequel il s'appuie et joue le rôle de fournisseur pour ses propres clients. Ex. : RightScale, Scalr, IniMov ;
- Développeur pour le cloud : produit des outils (déploiement, autoréparation, etc.) pour le cloud. Ex. : VMware, Labo de recherche et entreprise (Roboconf).

1.2 Problématiques de sécurité

Le cloud computing est une technologie basée sur les notions de multi-location (système distribué) et de virtualisation, ce qui introduit de nouveaux risques et des vulnérabilités spécifiques au cloud computing en plus des risques encourus par les environnements traditionnels [39]. En effet, la diversité des utilisateurs et le nombre important de composant

matériel et logiciel entraîne la création de vulnérabilités nouvelles. Ici, nous nous intéressons à la sécurité, et particulièrement à la sécurité des applications dans le cloud.

1.2.1 La sûreté de fonctionnement

Dans sa généralité, la sûreté de fonctionnement désigne l'aptitude d'un système à remplir une ou plusieurs fonctions requises dans des conditions données. Dans le domaine de l'informatique, il désigne la confiance que les utilisateurs placent dans le service délivré. Elle s'articule autour de trois axes : les attributs la caractérisant, les entraves empêchant sa réalisation, et les moyens pour l'atteindre.

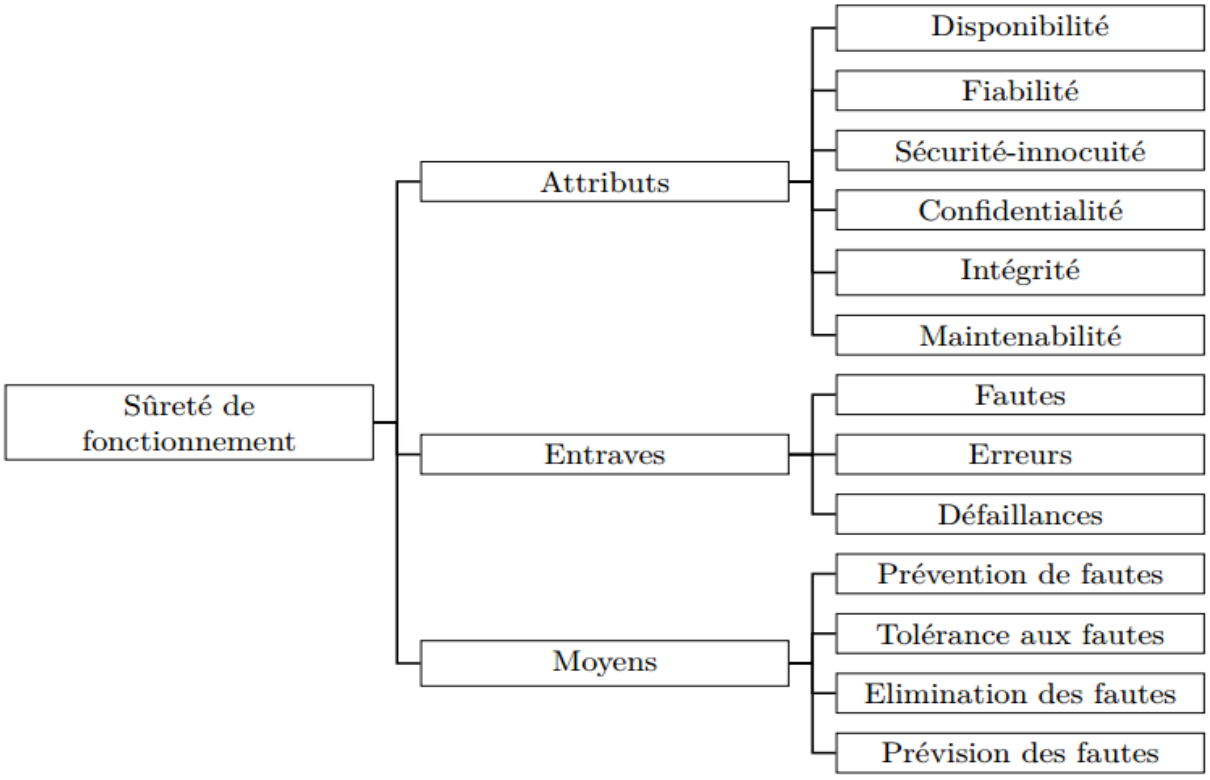

FIGURE 1.2 – Arbre de la sûreté de fonctionnement [15]

1.2.1.1 Principes

Tel que décrit dans la norme ISO 9126, ces principes ont pour objectif, dans un premier temps, de fournir les bases d'un langage commun et des seuils de référence, puis, dans un second temps, de fournir des repères externes pour définir et évaluer la qualité d'un produit. Il s'agit de la disponibilité (aptitude à être en état d'accomplir une fonction), de

la fiabilité (aptitude d'un dispositif à accomplir une fonction requise dans des conditions données pour une période de temps donnée), de la sécurité-innocuité (absence de conséquences catastrophiques), de la confidentialité (absence de divulgations non autorisées de l'information), de l'intégrité (assurer la cohérence, la fiabilité, et la pertinence des données) et de la maintenabilité (aptitude aux réparations et aux évolutions).

1.2.1.2 Entraves à la sûreté de fonctionnement

Les entraves à la sûreté de fonctionnement sont des incidents indésirables, mais non prévisibles, produits ou résultant de la non-sûreté de fonctionnement. Il s'agit de fautes, d'erreurs et de défaillances. Dans la plupart des cas, les défaillances sont les conséquences des fautes et des erreurs. Les fautes peuvent être classées selon leurs causes phénoménologiques (physiques ou dues à l'homme), leur nature (accidentelle ou intentionnelle), leur phase de création ou d'occurrence (au cours du développement ou en opération), leur situation par rapport aux frontières du système (internes ou externes), et leur persistance (permanente ou temporaire) [15].

1.2.1.3 Méthodes pour la sûreté de fonctionnement

Les méthodes pour la sûreté de fonctionnement sont les techniques permettant de fournir au système l'aptitude à délivrer un service conforme à celui supposé, et de donner confiance dans cette aptitude. Les méthodes de sûreté sont essentiellement : la prévention de fautes (consiste à empêcher l'introduction de fautes), l'insensibilité aux fautes (consiste à continuer le service en dépit des fautes), de l'élimination des fautes (consiste à réduire le nombre et la sévérité des fautes) et de la prévision des fautes (consiste à estimer la présence, la création et les conséquences des fautes).

1.2.2 La sécurité

L'ITSEC (Information Technology Security Evaluation Criteria) définit la sécurité comme la combinaison de trois propriétés : la confidentialité, l'intégrité et la disponibilité de l'information. Elle peut aussi être vue comme la capacité du système informatique à continuer son service malgré les agressions externes physiques ou logiques. Ainsi on distingue la sécurité de l'information (élément de connaissance traduit par un ensemble de signaux selon un code déterminé, en vue d'être conservé, traité ou communiqué [40]), des données (représentation d'une information sous une forme conventionnelle adaptée à son exploitation [40]), et des systèmes d'information (ensemble organisé de ressources qui permet de collecter, stocker, traiter et distribuer de l'information).

La sécurité possède sa propre terminologie, dont il est important de rappeler les définitions avant de détailler les trois propriétés qui la composent, puis de donner une vue d'ensemble des moyens pour la sécurité.

1.2.2.1 Terminologie

Parmi les termes les plus utilisés dans le domaine de la sécurité informatique, on retrouve : vulnérabilité, attaque, intrusion, menace, risque et incident.

- **Vulnérabilité ou faille** : est une faiblesse dans un système informatique, permettant à un attaquant de porter atteinte au fonctionnement normal de ce système ;
- **Attaque ou tentative d'intrusion** : acte malveillant, à travers lequel un attaquant cherche à délibérément violer une ou plusieurs propriétés de sécurité ;
- **Intrusion** : acte malveillant externe résultant d'une attaque qui a réussi à exploiter une vulnérabilité ;
- **Menace** : possibilités et probabilités d'attaque contre la sécurité. Une menace est définie par le processus d'attaque, par la cible et par le résultat (conséquences de la réussite d'une attaque) ;
- **Risque** : résultat de la combinaison de menaces et de vulnérabilités ;
- **Incident** : évènement qui ne fait pas partie des opérations standards d'un service et qui provoque ou peut provoquer une interruption de service ou altérer sa qualité.

1.2.2.2 Propriétés

Un système informatique est dit sécurisé, lorsqu'il est en mesure de garantir la confidentialité, l'intégrité et la disponibilité de l'information.

- **Confidentialité** : a été définie par l'Organisation Internationale de Normalisation (ISO) comme « le fait de s'assurer que l'information n'est accessible qu'à ceux dont l'accès est autorisé ». Ex : La désormais célèbre vulnérabilité Heartbleed [41], présente dans la bibliothèque logicielle de cryptographie OpenSSL, permet des attaques violant la confidentialité des données.
- **Intégrité** : La propriété que l'information n'a pas été modifiée de manière non autorisée. Ex : Le ver ILOVEYOU [42], apparu en 2000, se propageait par le biais de la messagerie Microsoft Outlook. Une fois exécuté, il remplaçait de nombreux fichiers présents sur le système ;
- **Disponibilité** : la propriété que les informations sont accessibles et modifiables en temps opportun par les personnes autorisées à le faire. Ex : Les plateformes de jeux-vidéos en ligne de Sony et Microsoft ont été plusieurs fois la cible d'attaques de déni de service, rendant inaccessibles ces plateformes.

D'autres propriétés comme l'**assurance** (réfère à la façon dont la confiance est fournie et gérée dans système), l'**authenticité** (la capacité à déterminer que les autorisations présentées par des personnes ou des systèmes sont authentiques), l'**anonymat** (la propriété que certains enregistrements ou transactions ne sont pas attribuables à un individu). Cependant, elles restent exprimables en termes de confidentialité, intégrité ou disponibilité.

1.2.2.3 Outils pour la sécurité

La figure 1.3, extrait de [15], présente une classification des méthodes disponibles pour garantir la sécurité vis-à-vis des attaques, vulnérabilités et intrusions, en fonction des quatre moyens pour la sûreté de fonctionnement.

	Attaque humaine	**Attaque technique**	**Vulnérabilité**	**Intrusion**
Prévention	Dissuasion, lois, pression sociale, service secret...	Pare-feu, authentification, autorisation...	Spécifications formelles et semi-formelles, méthodes rigoureuses de développement et gestion	Prévention et élimination des attaques et vulnérabilités
Tolérance	Prévention des vulnérabilités, élimination et tolérance aux intrusions		Prévention et élimination des attaques, tolérance aux intrusions	Détection et recouvrement d'erreurs, masquage des fautes, détection d'intrusion, gestion des fautes
Elimination	Contre-mesures physiques, capture de l'attaquant	Maintenance préventive et corrective visant à supprimer les agents malveillants	Preuve formelle, model-checking, inspection, test, maintenance préventive et corrective, incluant les patchs de sécurité	Elimination des attaques et des vulnérabilités
Prévision	Collecte de renseignements, évaluation des menaces...	Analyse des agents malveillants latents	Evaluation des vulnérabilités, des difficultés de les exploiter, de leurs conséquences potentielles...	Prévision des vulnérabilités et des attaques

FIGURE 1.3 – Classification des méthodes pour la sécurité

1.2.3 Menaces, vulnérabilités et attaques dans le cloud

Le cloud computing est basé sur les technologies de la multi-location et de la virtualisation, ce qui introduit de nouveaux risques et des vulnérabilités de sécurité spécifiques au cloud en plus des risques encourus par les environnements traditionnels [39]. Le tableau

1.1 extrait de [28], donne les principaux risques de sécurité dans le cloud et leur taux de récurrence.

Risques	Récurrence
Fragilité dans la gestion des accès et des identités.	80,00%
Brèche de sécurité au niveau des Datacenters.	66,67%
Perte de données.	66,67%
Piratage de compte.	60,00%
Exploit de vulnérabilités des systèmes d'exploitation et des applications hébergées.	60,00%
Utilisation frauduleuse des technologies cloud en vue de se cacher et perpétuer des attaques.	53,33%
Action malveillante initiée, en interne, dans les effectifs du fournisseur.	53,33%
Utilisation d'API non sécurisés pour l'intégration des applications avec les services cloud.	53,33%
Insuffisances dans les stratégies d'entreprise d'adoption ou de passage au cloud.	40,00%
Attaque par déni de services.	20,00%
Failles liées à l'hétérogénéité des technologies imbriquées.	13,33%

TABLE 1.1 – Risques de sécurité dans le cloud

1.3 Mécanismes de sécurité dans le cloud

Les mécanismes de sécurité ont pour but d'assurer la sécurité des hôtes et des applications hébergées dans le cloud. Nous nous intéressons particulièrement à trois mécanismes de sécurité :

— Le contrôle des accès réseau, réalisé par les pare-feu ou firewall.
— La protection des données par des techniques de chiffrement ;
— Le contrôle d'accès aux applications.

1.3.1 Contrôle d'accès réseau

Un pare-feu est un outil permettant de contrôler le trafic circulant entre l'intérieur et l'extérieur d'un périmètre de sécurité [43]. Le périmètre de sécurité constitue la limite entre

le réseau que l'on considère comme sûr ou que l'on désire protéger et le reste de l'Internet [43]. La fonction principale d'un pare-feu est le filtrage de paquets, qu'il réalise sur la base de règles permettant d'interdire ou autoriser certains types de trafic. Les pare-feu peuvent être répartis en deux principales catégories : pare-feu sans état (stateless firewall) et pare-feu à états (stateful firewall). Les plus utilisés sont les pare-feu avec état qui vérifient en plus l'appartenance des paquets à une connexion en cours (connexion TCP) avant de les autoriser.

La complexité des architectures cloud, le nombre important d'organisations présentent couplés à la complexité des applications actuelles, rendent d'une part les exigences en contrôle d'accès réseau plus importantes et, d'autre part, rendent plus difficile la définition du périmètre de sécurité en comparaison aux environnements traditionnels. Le positionnement des pare-feu et leur comportement sont donc des aspects hautement stratégiques. Pour maintenir une défense en profondeur dans les infrastructures virtuelles, il existe les deux types de pare-feu virtuels suivants :

— **Pare-feu en mode pont** : machine virtuelle déployée comme passerelle des réseaux virtuels, capable de router, filtrer et traduire les adresses du trafic entrant et sortant. Ces pare-feu sont généralement contrôlés par les clients ;
— **Pare-feu en mode hyperviseur** : composant logiciel embarqué dans l'hyperviseur qui filtre le trafic envoyé ou reçu par les machines virtuelles sans prendre en compte la topologie réseau. Il est généralement contrôlé par le fournisseur de services, mais certaines règles peuvent être appliquées par les clients seulement sur certains réseaux virtuels, ce qui implique de donner un contrôle partiel aux clients sur ce type de pare-feu.

1.3.2 Le chiffrement des données

Les données dans le cloud doivent faire face au problème d'intégrité. En effet les données peuvent être corrompues soit au moment de leur transport via le réseau internet soit au moment du stockage chez le fournisseur. Afin de résoudre ces problèmes (transport et stockage), les fournisseurs de services cloud ont concentré leurs efforts sur le développement de nouvelles techniques de chiffrement :

— **Transport Layer Security (TLS)** : est un protocole cryptographique de couche 5 (couche de session) dans le modèle ISO qui permet l'authentification, et le chiffrement des données qui transitent entre les serveurs, les machines et les applications en réseau. Comparé à son prédécesseur le SSL (Secure Socket Layout), beaucoup de bugs et d'attaques connues ont été corrigées, de nouvelles fonctionnalités et de nouveaux algorithmes ont été ajoutés ;
— **Mécanisme OTP** : le mécanisme One Time Password (OTP) ou le mot de passe à usage unique est un mot de passe qui n'est valable que pour une session ou

une transaction. De plus, ce mot de passe n'est plus choisi par l'utilisateur, mais généré automatiquement par une méthode de pré-calculé, ce qui va éliminer certaines lacunes associées aux mots de passe statiques [7] ;

— **Techniques de chiffrements** : pour augmenter le niveau de sécurité des données dans le cloud public, il est important de les chiffrer en utilisant l'anonymisation avec des sauvegardes et des audits. L'anonymisation peut être définie comme l'opération de suppression de l'ensemble des informations permettant d'identifier directement ou indirectement un individu [7].

1.3.3 Contrôle d'accès aux applications

D'après [16], la gestion des accès permet de s'assurer de mettre à la disposition de chacune des personnes impliquées dans le système, à tout instant, tous les moyens nécessaires à la réalisation de sa mission, et que ces moyens se limitent au juste nécessaire. Les droits d'accès permettent donc, dans un système informatique, de garantir la confiendialité et l'intégrité des données. Cela permet aussi d'établir une hiérachisation ou un niveau de confiance que le système accorde à chaque identité.

Pour mettre en œuvre une politique de contrôle d'accès dans une organisation, cinq concepts de base entrent en jeu : l'identité, le modèle de contrôle d'accès, les habilitations (permissions), les comptes utilisateurs et l'approche algorithmique choisie.

- **L'identité** : c'est un ensemble de caractéristiques propres par lesquelles une personne ou une organisation est connue ou reconnue [16]. Une identité appartient à une **entité** qui représente une personne physique ou morale, une ressource ou un groupe d'entités individuelles ; elle est caractérisée par un **identifiant** qui est une information unique qui permet de distinguer, sans ambiguïté, une identité d'une autre pour un contexte donné ;
- **Modèle de contrôle d'accès** : définit la manière dont le contrôle d'accès sera mis en œuvre dans le système. On distingue principalement trois modèles : le modèle discrétionnaire (Discretionary Access Control – DAC), les droits d'accès sont donnés à chaque utilisateur du système ; le modèle mandataire (Mandatory Access Control – MAC), limite les accès des utilisateurs aux ressources en se basant sur des niveaux de sécurité ; le modèle RBAC (Role Based Access Control, figure 5) qui est le plus en vue et dont la création vise à répondre aux problématiques que posaient les systèmes multi-utilisateurs et multi-applications. L'élément central de ce modèle est la notion de rôle qui se définit de manière simple comme étant un ensemble de permissions vis-à-vis d'un système d'information [16].

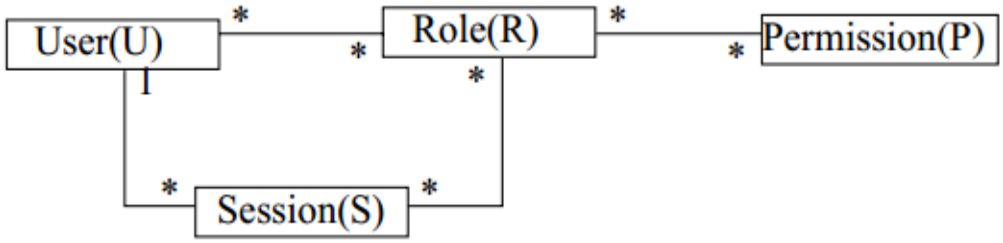

FIGURE 1.4 – Modélisation UML du RBAC [13]

- **Permissions** : représentent le droit d'effectuer une action sur une ressource. Pour un système d'information les principales actions définies sont : lecture, création, mise à jour, suppression ;
- **Le compte utilisateur** : le compte utilisateur est la représentation informatique d'une identité ;
- **L'approche algorithmique** : est la manière dont seront gérées, stockées et calculées les différentes permissions ainsi que leurs mises en relation avec les comptes utilisateurs et les ressources. Il existe plusieurs approches pour mettre en œuvre un contrôle : l'approche matrice de contrôle d'accès (figure 1.5) consiste à une matrice comportant en ligne la liste des utilisateurs et en colonne la liste des ressources (ou inversement) et à chaque intercession se trouve la liste des actions que cet utilisateur peut effectuer sur cette ressource. Cette approche pose un problème d'inefficacité mémoire car de nombreuses cases vides [16] ; l'approche ACL (figure 1.6) consiste à une liste d'utilisateur et à chaque utilisateur est liée la liste des ressources avec, en précision, ses droits sur chacune des ressources.

	Fichier Salaires	Fichier Impôts	Programmes impôts	Imprimante P1
Alice	Lire, Écrire		Exécuter	Écrire
Bob		Lire		
Jean	Lire			Écrire

FIGURE 1.5 – Matrice de contrôle d'accès

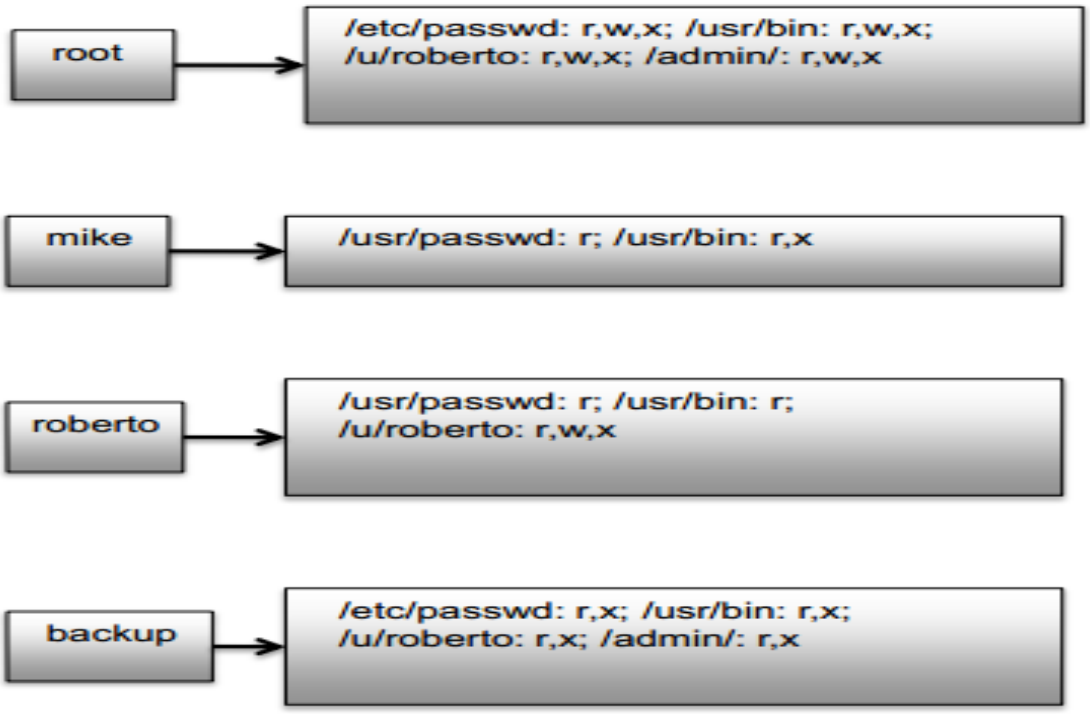

FIGURE 1.6 – Capabilities ACL [9]

1.4 Présentation de Mobility

Mobility est un ERP, c'est-à-dire un progiciel qui permet de gérer l'ensemble des processus d'une entreprise. Il intègre des fonctions comme la gestion des ressources humaines, la gestion financière et comptable, l'aide à la décision, la vente, la distribution, l'approvisionnement, la production ou encore du e-commerce. Le principe fondateur d'un ERP est d'après [26], de construire des applications informatiques correspondant aux diverses fonctions citées précédemment de manière modulaire, tout en partageant une base de données commune au sens logique ; et embarquant un moteur de workflow qui permet, lorsqu'une donnée est enregistrée dans le SI, de la propager dans les modules qui en ont l'utilité, selon une programmation prédéfinie.

Mobility est développé par l'entreprise IniMov, autour d'une Architecture Orientée Service (SOA) qui est un style architectural basé sur un fournisseur, un demandeur et une description de service, et supportant les propriétés de modularité, encapsulation, découplage, réutilisation et composabilité ou encore un modèle de programmation avec ses standards, paradigmes, outils et technologies associées. Dans la pratique, une SOA se manifeste sous forme de ***webservice*** (service web) qui est défini par [11] comme étant un programme

informatique permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués.

Afin de le rendre accessible partout et à faible coût, Mobility a été conçu comme une application à déployer dans le cloud (SaaS), le rendant ainsi utilisable comme un service, offrant aux organisations clientes des outils permettant de gérer à l'heure actuelle les points suivants leur organisation : l'achat, la vente, les stocks, la comptabilité, les points de vente et la gestion des opérations financières (pour les établissements financiers), la génération automatique des rapports.

1.5 Conclusion

Nous avons présenté les principaux travaux associés à la problématique de ce mémoire. Ils concernent deux domaines de recherche : le cloud computing, plus particulièrement, nous avons défini le cloud et donné ses caractéristiques, présenté les différents types de cloud (selon la propriété et les services), et les acteurs y intervenants ; la sécurité dans le cloud d'où nous avons fait ressortir d'une part les problèmes qui pèsent sur les infrastructures et le réseau, sur les applications et sur les informations, et, d'autre part, les mécanismes de sécurité pour y faire face.

Au terme de cette partie nous présentons Mobility qui est une application de la famille des ERPs, déployée dans le cloud comme une SaaS mais qui souffre de nombreux problèmes de sécurité que nous détaillons dans le chapitre suivant et proposons des solutions adaptées.

Chapitre 2

Élaboration d'une politique de sécurité adaptée à Mobility

Dans ce chapitre, nous détaillons la politique de sécurité que nous avons développée au cours de ces travaux de mémoire. Pour ce faire, nous menons préalablement une étude du système, puis présentons les éléments entrant dans la politique.

La politique de sécurité informatique est un des éléments de la politique de sécurité du système d'information. Concrètement, il s'agit d'un plan d'actions définies pour maintenir un certain niveau de sécurité. Elle peut être comparée à une chaîne dont le maillon le plus faible caractérise le niveau de sécurité du système. Ainsi, la politique de sécurité informatique doit être abordée dans un contexte global qui commence par la sensibilisation des utilisateurs, la sécurité de l'information, la sécurité des données, la sécurité des réseaux, la sécurité des systèmes d'exploitation, la sécurité des télécommunications, la sécurité des applications (programmation sécurisée), la sécurité physique (des infrastructures matérielles). La norme l'ISO/CEI 27001 définit une démarche en trois étapes pour la mise en œuvre d'une politique de sécurité :

— **Evaluer les risques et leur criticité** : quels risques et quelles menaces, sur quelles données et quelles activités, avec quelles conséquences ? On parle de « **cartographie des risques** ». C'est d'elle que dépend la qualité de la sécurité qui va être mise en œuvre ;
— **Rechercher et sélectionner les parades** : que va-t-on sécuriser, quand et comment ? Étape difficile des choix de sécurité : dans un contexte de ressources limitées (en temps, en compétences et en argent), seules certaines solutions pourront être mises en œuvre ;
— **Mettre en œuvre les protections, et vérifier leur efficacité.**

La mise en œuvre de cette démarche s'est faite dans ce travail en trois phases réparties entre le chapitre 2 et le chapitre 3. La première phase étant la délimitation du périmètre qui inclut l'identification des risques ; la deuxième est le choix de procédés pour protéger les données et les informations ; la troisième phase étant le chapitre 3 dans son intégralité qui porte sur la mise en œuvre.

2.1 Environnement ou périmètre : Mobility

La construction d'une politique de sécurité, doit tenir compte du système d'implantation de celle-ci pour qu'elle soit efficace. En effet, Mobility est un ERP donc formé de plusieurs composants indépendants échangeant des données selon un workflow prédéfini ; il est bâti sur le style architectural SOA ; il est accessible comme service cloud (SaaS), par conséquent les clients n'accèdent qu'aux services auxquels ils ont souscrit. L'objectif de cette partie est de ressortir les risques et les menaces qui pèsent sur Mobility.

2.1.1 Identification des ressources

Comme dans tout système informatique, les ressources sont de trois catégories : matérielles, logicielles et utilisateurs.

— **Les ressources matérielles** : Mobility étant dans le cloud, alors l'infrastructure matérielle se résume à deux serveurs virtuels, auxquels on accède via une connexion SSH (Secure SHell, qui est à la fois un programme informatique et un protocole de communication sécurisé) ;
— **Les ressources logicielles** : la première étant le système d'exploitation RedHat, puis les applications installées : le serveur Apache2 (serveur web), MySql (serveur de bases de données), PHP (moteur de script) et Plesk (application de gestion des serveurs). A cela s'ajoutent les deux applications qui composent le système Mobility et les sept services indépendants qu'il offre ;
— **Les ressources utilisateurs** : les utilisateurs peuvent être infinis puisque le système est ouvert, toutefois ils se regroupent en deux catégories : les gestionnaires, ce sont les employés d'IniMov et les clients qui peuvent être des individus ou des systèmes informatiques ;

2.1.2 Identification des risques

Différentes méthodes d'analyse des risques sur le système informatique existent (Cramm, ISO 27005, OCTAVE, EBIOS). Dans le cadre de ce travail nous utilisons la méthode EBIOS (Expression des Besoins et Identification des Objectifs de Sécurité). Elle définit le **risque** comme étant un scénario qui combine un événement redouté (sources de menaces, bien essentiel, besoin de sécurité, impacts) et un ou plusieurs scénarios de menaces (sources de menaces, bien support, critère de sécurité, menaces, vulnérabilités) [14] et le structure en cinq éléments : le contexte (il est invariant ; c'est InMov et Mobility), les évènements redoutés classés en niveau de cricité (critique, important, limité ou négligeable), scénarios de menaces (les scénarios possibles) classés en niveau de vraisemblance (minime, significatif, fort ou maximal), les risques, les mesures de sécurité.

Dans cette partie, nous identifions les évènements redoutés, les scénarios de menaces et les risques ; les mesures pour y faire face sont abordées dans la partie 2.2. Pour identifier au mieux les risques, une étude poussée de Mobility est primordiale. Nous la menons sur deux plans : les risques liés au réseau et les risques liés à l'implémentation.

2.1.2.1 Les risques liés au réseau

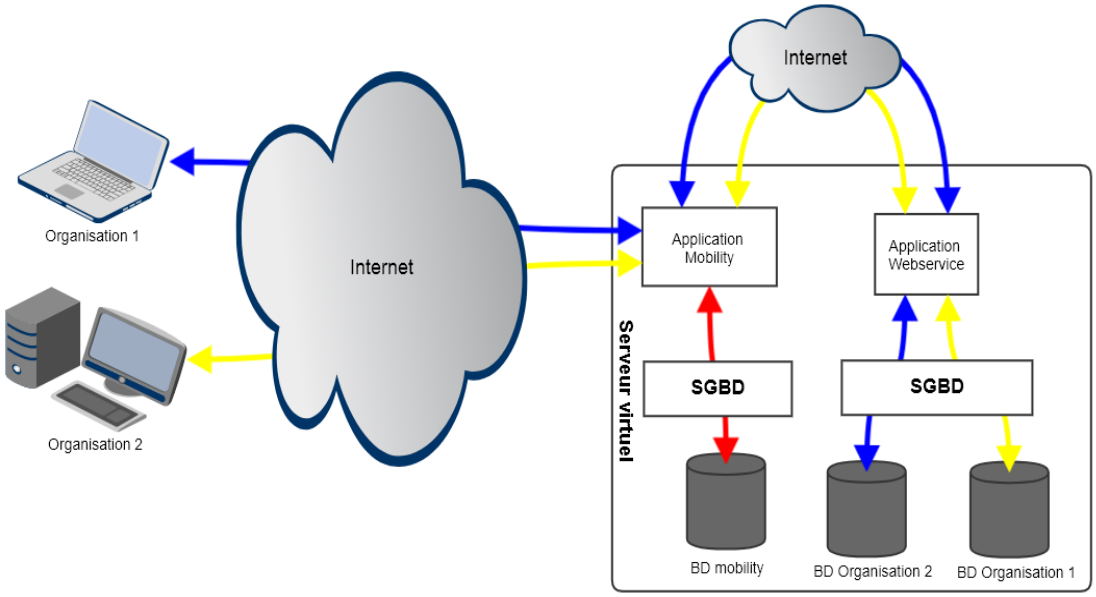

FIGURE 2.1 – Architecture réseau de Mobility

La figure 7 présente l'architecture réseau de Mobility ; elle est constituée d'un serveur virtuel, à l'intérieur duquel sont installées trois applications : l'application Mobility, qui reçoit les requêtes des clients voulant utiliser les services, c'est aussi elle qui effectue les traitements métiers ; l'application Webservice communique uniquement avec l'application Mobility par internet et a pour rôle de stocker et retourner les données des clients ; une application de gestion de bases de données qui gère les différentes base de données (celle de Mobility et celles des clients).

De l'analyse de cette architecture, il ressort deux principaux évènements redoutables, se déclinant en six scénarios de menaces ayant un certain niveau de criticité et de vraisemblance :

Evènements redoutés	Scénarios de menaces	Criticité / Vraisemblance	Risques
Piratage	Un individu écoute les connexions.	Critique / Maximale	**Biens essentiels** : données des clients (besoin de disponibilité, d'intégrité et de confidentialité). **Biens supports** : les applications Mobility et webservice (modifiable), serveur virtuel (détournable). **Conséquences** : indisponibilité, vol ou modification des données, perte de confiance, perte de client, utilisation malveillante du serveur.
	Un individu intercepte et modifie les données échangées.	Critique / Maximale	
	Un individu pirate le serveur virtuel.	Critique / Significative	
Interruption du service	Mauvaise manipulation d'un employé (volontaire ou non).	Critique / Significative	
	Panne matériel.	Critique / Minime	
	Attaque de déni de service (DOS).	Critique / Significative	

TABLE 2.1 – Risques liés au réseau

2.1.2.2 Les risque liés à l'implémentation

L'objectif de cette partie est de faire une analyse de l'implémentation du système et plus précisément de son architecture logicielle afin d'en ressortir les risques de sécurité.

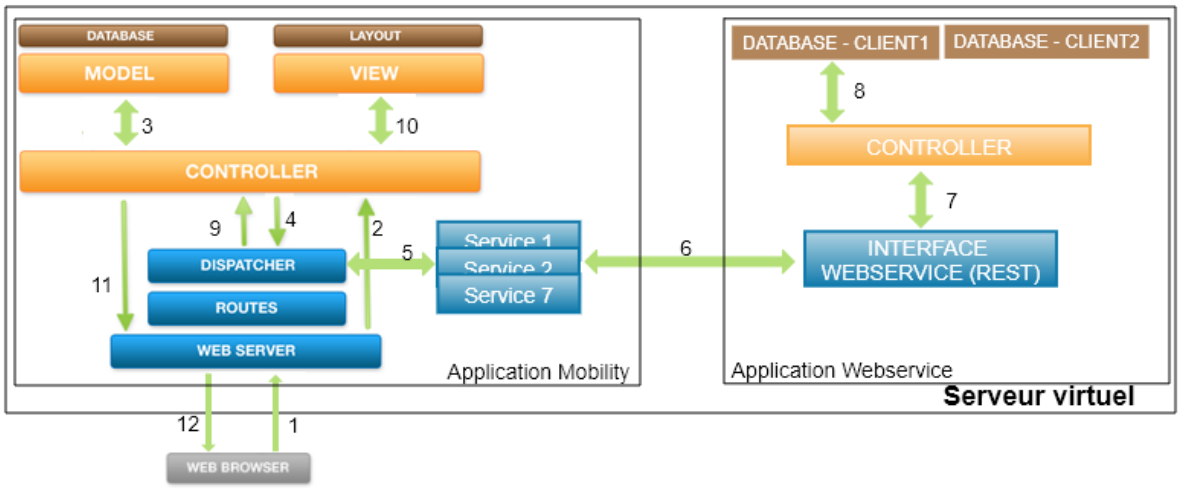

FIGURE 2.2 – Architecture d'implémentation

De l'étude de celle-ci (figure 2.2), il ressort que le système est constitué de deux applications indépendantes bâties suivant le patron MVC (Modèle-Vue-Contrôleur) et communiquant via le web. L'application webservice est constituée des trois composants de base qu'impose le patron MVC et a pour unique fonction le stockage et la restitution des données des clients. L'application Mobility, en plus des trois composants qu'impose le patron MVC, elle comprend deux composants supplémentaires : « Dispatcher » dont le rôle principal est d'appeler les services et « Service » qui sont sept composants représentant chacun un service offert par Mobility ; sa fonction est d'effectuer les traitements métiers et appeler l'application webservice lorsqu'il y'a des opérations à effectuer sur les données.

Le principal évènement redoutable ressorti de cette analyse, se décline en trois scénarios de menaces ayant un certain niveau de criticité et de vraisemblance :

Evènements redoutés	Scénarios de menaces	Criticité / Vraisemblance	Risques
Utilisation frauduleuse des services	La page d'accueil affiche tous les services existant au lieu de ceux auxquels le client a souscrit.	Critique / Maximale	**Biens essentiels** : données des clients (besoin de confidentialité), logique métier. **Biens supports** : les applications Mobility et webservice (modifiable). **Conséquences** : vol ou modification des données, utilisation de service non facturée.
	Un individu interroge directement l'application webservice.	Critique / Significative	
	Un individu essaie d'interroger directement un service.	Critique / Significative	

TABLE 2.2 – Risques liés à l'implémentation

2.2 Mesures de sécurité

Notre objectif est de donner une réponse à la problématique énoncée à la fin de l'introduction générale. Nous voulons ainsi bâtir une politique de sécurité fiable qui répond aux besoins de Mobility. Pour cela, dans la partie 2.1 nous avons cartographié les risques qui pèsent sur lui. Dans cette partie nous proposons des mesures pour répondre à ces risques.

2.2.1 Organisation : Responsabilités

Cette mesure répond aux besoins de confidentialité et d'intégrité ; et a pour objectif d'apporter une réponse au scénario « Mauvaise manipulation d'un employé (volontaire ou non) » de l'évènement redouté « Interruption du service ». Cette mesure consiste à regrouper les acteurs (un acteur est un utilisateur qui a toujours le même comportement vis-à-vis du système [34]) en plusieurs catégories, avec des niveaux de responsabilité différents vis-à-vis du système. D'après les dix principes de sécurité cité dans [9], cette mesure correspond à la fois aux principes de séparation des privilèges (ce principe exige que des conditions multiples soient requises pour accéder à des ressources restreintes ou qu'un programme

puisse effectuer des actions) et du moindre privilège (chaque programme et utilisateur d'un système informatique devrait fonctionner avec les privilèges minimaux nécessaires pour fonctionner correctement). Dans ce système, nous avons recensé trois catégories d'acteurs (Tableau 2.3) :

Acteurs	Description	Rôle
Administrateur	C'est un employé d'IniMov.	— Ajouter/Retirer des services à un client ; — Désactiver un client lorsque celui-ci le souhaite ou n'a pas payé ;
Utilisateur	C'est un employé d'une organisation cliente.	Utiliser les services lorsqu'il a le droit.
Gestionnaire	C'est un employé d'une organisation cliente, il a tous les droits sur les informations.	— Utiliser les services ; — Créer des groupes d'utilisateur ; — Créer des utilisateurs et leur affecter un groupe ; — Affecter des droits d'accès aux groupes.

TABLE 2.3 – Acteurs du système

2.2.2 Protection et disponibilité des données

Cette mesure répond aux besoins de disponibilité, de confidentialité et d'intégrité ; et a pour objectif d'apporter trois réponses aux scénarios « Un individu écoute les connexions », « Un individu intercepte et modifie les données échangées », « Un individu pirate le serveur virtuel », « Panne matériel », «Un individu interroge directement l'application webservice » des évènements redoutés « Piratage », « Interruption du service », « Utilisation frauduleuse des services ». Cette mesure se décompose en deux sous mesures :

2.2.2.1 Connexion sécurisée

La sécurisation de la connexion répond à trois besoins principaux : authentifier le serveur, garantir la confidentialité et l'intégrité des données échangées. D'après la figure 2.1,

deux connexions sont susceptibles d'être piratées : la connexion entre client et l'application Mobility et la connexion entre l'application Mobility et l'application webservice. Dans les deux cas, il s'agit de connexion utilisant les protocoles web. Pour les sécuriser, nous utilisons le protocole HTTPS basé sur le protocole TLS. Celui-ci protège les données en utilisant un mécanisme de vérification d'intégrité et un algorithme de chiffrement symétrique dont la clé a préalablement été générée et distribuée, en utilisant un algorithme de chiffrement asymétrique principalement RSA. En effet, lors de l'initialisation de la connexion, le client envoie une requête au serveur qui répond en envoyant son certificat qui contient sa clé publique (pour le chiffrement asymétrique), ses informations ainsi qu'une signature numérique sous forme de texte chiffré dont le déchiffrement par la clé publique d'une autorité de certification permet de confirmer l'authenticité du certificat transmis.

2.2.2.2 Authentification des requêtes

L'application webservice est développée comme une API REST (Application Programming Interface, REpresentational State Transfer) et a pour fonction le stockage et la restitution des données clients, uniquement à l'application Mobility lorsque demande est faite par celle-ci. Pour répondre au scénario «Un individu interroge directement l'application webservice », nous devons restreindre l'accès à celle-ci. Pour cela nous mettons en place deux mécanismes :

- La restriction d'accès en fonction de l'origine grâce au paramètre « Access-Control-Allow-Origin », qui est une fonction native dans les serveurs web. Il s'agit de donner comme valeur à ce paramètre l'URL de l'application Mobility ;
- L'**OAuth** est un protocole libre qui permet d'autoriser une application client d'utiliser l'API sécurisée d'une autre application pour le compte d'un utilisateur. Il est basé sur l'utilisation d'un jeton (token) attribué par serveur d'autorisation et qui donne l'accès à des ressources protégées sur le serveur de ressource.

2.2.2.3 Cluster actif-passif

Le cluster actif-passif est une architecture de haute disponibilité de type miroir. L'application est exécutée sur un serveur primaire et redémarrée automatiquement sur un serveur secondaire si le serveur primaire est défaillant. La réplication des données est configurée pour s'exécuter en temps réel synchrone vu le caractère sensible des données et ne s'applique qu'aux fichiers de bases de données et de session du serveur web.

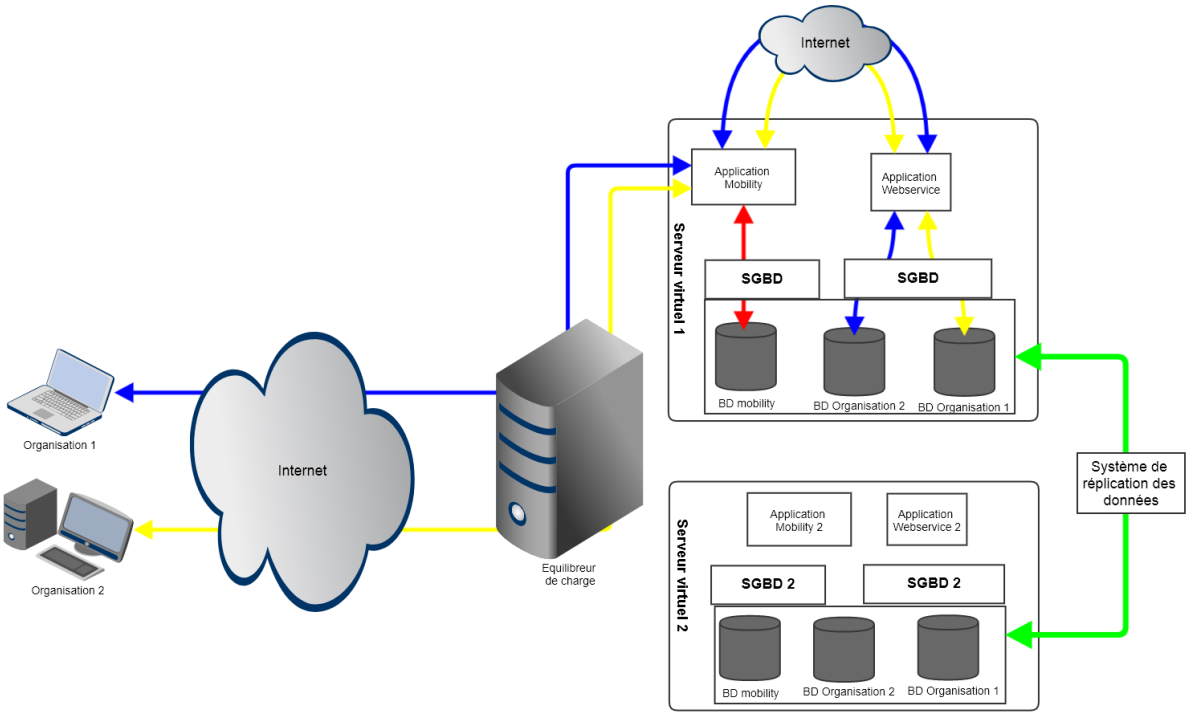

FIGURE 2.3 – Architecture de haute disponibilité

Le fonctionnement du cluster actif-passif temps réel miroir est le suivant :

— Les utilisateurs sont connectés à l'adresse IP de l'équilibreur de charge, qui redirige les requêtes sur le serveur 1 ou 2 selon sa disponibilité ;

— Le serveur 1 exécute l'application et le système de réplication réplique les fichiers ouverts par l'application. Seules les modifications faites par l'application à l'intérieur des fichiers sont répliquées en continu à travers le réseau, minimisant ainsi le trafic ;

— Lorsque le serveur 1 est défaillant, l'équilibreur de charge redirige les requêtes sur le serveur 2. L'application poursuit son exécution sur le serveur 2 en modifiant localement ses fichiers qui ne sont plus répliqués vers le serveur 1 ;

— A la reprise, après panne du serveur 1, le système de réplication resynchronise automatiquement les fichiers de ce serveur à partir de l'autre serveur. L'équilibreur de charge réoriente les requêtes de nouveau vers le serveur 1 et le processus recommence.

2.2.3 Sécurisation de l'information : le contrôle d'accès

Cette mesure répond aux besoins de confidentialité et d'intégrité ; et a pour objectif d'apporter une réponse aux scénarios « La page d'accueil affiche tous les services existant

au lieu de ceux auxquels le client a souscrit », « Un individu essaye d'interroger directement un service.», de l'évènement redouté « Utilisation frauduleuse des services ». Cette mesure se concentre autour du contrôle d'accès.

D'après la partie 1.3.3, le contrôle d'accès s'assure de mettre à disposition des personnes impliquées dans le système, tous les moyens nécessaires à la réalisation de sa mission. La mise en œuvre d'une politique de contrôle d'accès dans un système informatique nécessite de définir au préalable : un modèle de contrôle d'accès, une approche algorithmique pour le stockage et la représentation des droits et une méthode de calcul des droits.

2.2.3.1 Modèle de contrôle d'accès

Dans la partie 1.3.3, nous avons présenté trois modèles de contrôle d'accès : RBAC, DAC, MAC. De ces trois modèles, nous avons choisi le modèle **RBAC** parce qu'il est le plus récent et le plus utilisé, il est très flexible, il est conçu pour résoudre les problèmes des systèmes multi-utilisateurs et multi-services ce qui est le nôtre. Ce modèle met en jeu : des utilisateurs ayant plusieurs rôles, des rôles auxquels ont été attribuées des permissions. Voir figure 1.4.

NB : Par la suite nous confondons la notion de rôle avec celle de groupe.

2.2.3.2 Approche algorithmique

Dans la partie 1.3.3 nous avons présenté deux approches pour stockage et le traitement des droits d'accès : l'approche matrice d'accès et l'approche ACL. De ces deux approches, nous avons retenu l'approche **ACL** qui est la plus utilisée actuellement et qui offre les meilleures performances mémoires. Elle est constituée de deux principales parties : **L'identifiant**, constitué de deux parties la première étant le nom du service et la seconde le nom de la fonctionnalité, selon le format suivant « service.fonctionnalité » ; **Les permissions**, ici elles sont au nombre de cinq : créer (perm_create), lire et rechercher (perm_read), mise à jour des données (perm_write), suppression des enregistrements existants (perm_delete), imprimer ou exporter les données (perm_export). Elles sont stockées sous forme d'arborescence dont la racine est l'identifiant, les branches sont les permissions et les feuilles les groupes.

NB : Tout droit qui n'est pas explicitement attribué est implicitement refusé.

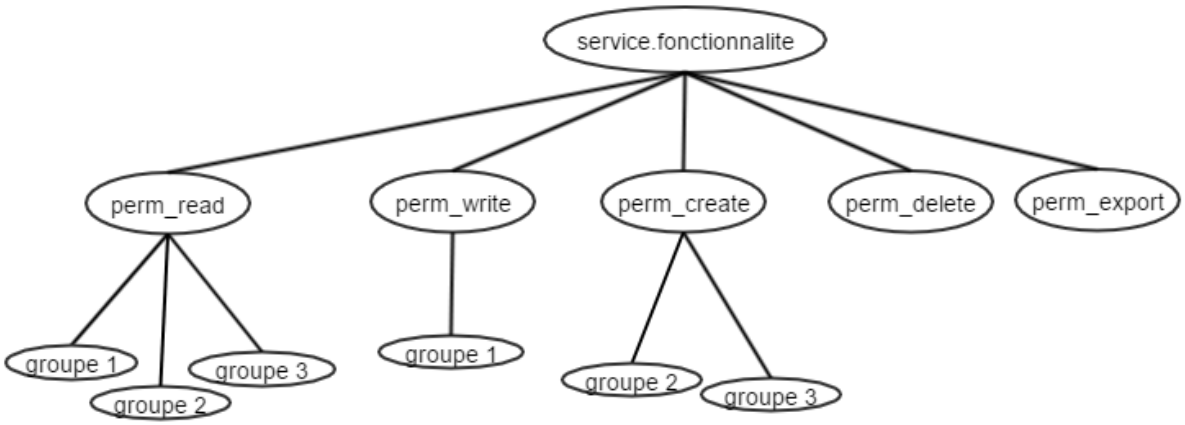

FIGURE 2.4 – Approche algorithmique pour le stockage des accès

2.2.3.3 Méthode de calcul des droits

Le calcul des droits d'un groupe d'utilisateurs sur une ressource, est facilités par le faite que de manière native, les langages de programmation offrent des outils pour gérer les structures de données en arbre. Dans notre cas nous avons choisi d'utiliser les tableaux parce qu'ils sont assez simples à manipuler et offrent de bonnes performances d'exécution sur des arbres assez petits comme le nôtre. Ainsi pour vérifier si un groupe peut accéder à une ressource, on teste l'existence de la case mémoire :

droits[service][fonctionnalités][permission][groupe].

2.3 Conclusion

Dans ce chapitre nous avons réalisé les deux premières phases de mise en œuvre d'une politique de sécurité (cartographie des risques et mesures pour y faire face) telle que définie par la norme ISO/CEI 27001. Pour cartographier les risques nous avons étudié Mobility, son architecture réseau, son architecture logicielle, son fonctionnement global. Au terme de cela, il en ressort que trois principaux risques pèsent sur Mobility : Piratage, Interruption de service, Utilisation frauduleuse des services ; qui peuvent se dérouler suivant neuf scénarios de criticité et de vraisemblance maximale. Les mesures élaborées pour faire face à ces risques reposent essentiellement sur la séparation des privilèges, le chiffrement des données en transit, la réplication des données et la mise en œuvre du contrôle d'accès à l'application.

Menaces	Réponses
Un individu écoute les connexions.	Utilisation du protocole HTTPS basé sur le protocole TLS.
Un individu intercepte et modifie les données échangées.	
Un individu pirate le serveur virtuel.	Protection du serveur, mis en œuvre par le fournisseur de cloud.
Mauvaise manipulation d'un employé (volontaire ou non).	Contrôle d'accès et limitation des privilèges sur le serveur.
Panne matériel.	Réplication du serveur virtuel par le fournisseur du cloud.
Attaque de déni de service (DOS).	Élasticité des serveurs virtuels pour répondre à la montée en charge.
La page d'accueil affiche tous les services existant au lieu de ceux auxquels le client a souscrit.	Contrôle d'accès, séparation des privilèges et moindre privilège.
Un individu essaie d'interroger directement un service.	
Un individu interroge directement l'application webservice.	Restriction d'accès par l'utilisation du protocole OAuth et le paramètre « Access-Control-Allow-Origin ».

TABLE 2.4 – Menaces sécuritaires recensées et les solutions apportées

Dans le chapitre suivant nous mettons en œuvres les mesures que nous avons élaborées et présentons les résultats obtenus.

Chapitre 3

Validation de l'approche

Dans ce chapitre qui correspond à la troisième étape pour la mise en œuvre d'une politique de sécurité telle que décrite par la norme l'ISO/CEI 27001. Pour cela, nous présentons d'abord les technologies utilisées pour l'implémentation des mesures élaborées précédemment puis nous présentons les résultats obtenus et terminons par des recommandations pour l'amélioration de la sécurité.

3.1 Implémentation

Dans cette partie, nous présentons les technologies utilisées pour l'implémentation de la politique de sécurité. Les différents outils utilisés sont mis en évidence ci-dessous en fonction des différentes couches des applications (Mobility et webservice). Le choix des langages de programmation (PHP 5, CSS3, Java Script, Ajax, MySQL 5.5 et HTML 5) a été réalisé par l'entreprise.

3.1.1 Couche métier

Joomla ! est un système de gestion de contenu (CMS) gratuit et open source plus précisément : c'est une application Web qui permet de créer des sites Web dynamiques en toute simplicité et dont le code source est éditable. Il construit selon le modèle architectural MVC toutefois, il peut être utilisé indépendamment de la fonction CMS offrant ainsi le cadre nécessaire pour construire des applications web. Ici nous utilisons la version 3 de Joomla !

3.1.2 Couche service

Slim PHP est un micro-framework PHP ouvert qui aide à écrire rapidement des applications et des APIs web. Il offre les fonctionnalités suivantes :

— Un routeur qui permet d'associer une fonction à une méthode HTTP et une URL ;
— Un système de middlewares qui permet de mettre en place une logique spécifique avant et/ou après chaque requête ;
— Un système de conteneur, pour pouvoir injecter des dépendances externes dans votre application.

Slim a servi pour la mise en œuvre de nos web-services selon l'architecture orientée au service (SOA) ainsi que le style architectural (RESTful). Ici nous utilisons la version 3 de Slim.

3.1.3 Serveurs

- **Serveur de base de données : MySql version 5.7**
  MySQL est un Système de Gestion de Bases de Données Relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisées au monde, autant par le grand public (applications web principalement) que par des professionnels. Il offre les avantages

suivants :
— Offre des performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données ;
— Il est multi-thread (plus grande performance) et multi-utilisateur ;
— C'est un logiciel libre, open source.
Ici nous utilisons la version 5.7 de Mysql.

- **Serveur web : Apache**
Le logiciel libre Apache HTTP Server (Apache) est un serveur web créé et maintenu au sein de la fondation Apache. C'est le serveur web le plus populaire du World Wide Web. Il est distribué selon les termes de la licence Apache. Il offre comme fonctionnalités :
— Apache est conçu pour prendre en charge de nombreux modules lui donnant des fonctionnalités supplémentaires : interprétation du langage Perl, PHP, Python et Ruby, serveur proxy, etc ;
— Permet de servir plusieurs sites à l'aide d'un seul serveur HTTP. Pour les clients, cette fonctionnalité est rendue visible par le fichier .htaccess ;
— La journalisation des traitements effectués par du serveur.
Ici nous utilisons la version 2 d'Apache.

3.2 Présentation des résultats

Des cinq mesures de sécurité présentées à la partie 2.2, les résultats ci-dessous sont ceux des mesures 2.2.1 et 2.2.3 relatives à la séparation des privilèges et à la sécurisation de l'information au moyen du contrôle d'accès. Ces mesures visent à garantir la confidentialité et l'intégrité des données. Par la suite, les résultats sont présentés autour de six points représentant d'une part les rôles joués par les utilisateurs vis-à-vis de Mobility (voir Tableau 4) et d'autre part les requis fonctionnels nécessaires pour l'opérationnalisation de ce système dans l'application.

- **Permettre l'ajout et la suppression de services à une organisation.**

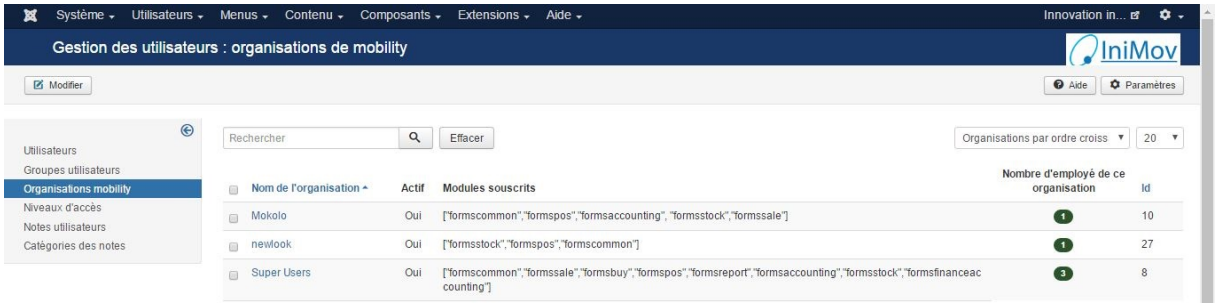

FIGURE 3.1 – Interface présentant les organisations

La figure 3.1 présente la liste des organisations présentes dans le système, les services auxquels ces organisations ont souscrit, le nombre d'utilisateurs enregistrés et si cette organisation est active. En cliquant sur une organisation, on arrive sur l'interface d'édition d'une organisation (figure 3.2). Là, on peut lui ajouter ou retirer l'accès à des services, ou lui retirer l'accès au système tout entier.

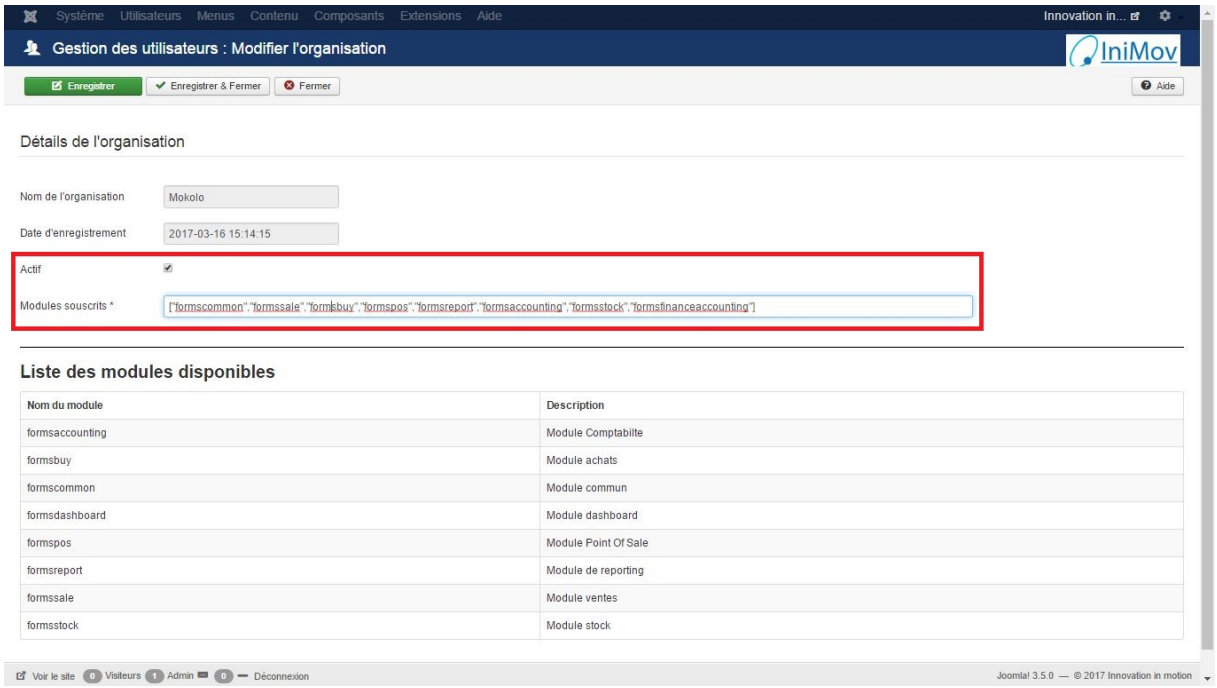

FIGURE 3.2 – Interface d'éditer une organisation

Ces interfaces (figures 3.1 et 3.2) ne sont accessibles qu'aux utilisateurs ayant le rôle de gestionnaire dans le système.

- **Permettre l'authentification d'un utilisateur pour accéder à l'espace de son organisation.**
  Cette première interface (figure 3.3) permet de vérifier que les paramètres d'indenti-

fication saisis, correspondent à un utilisateur de cette organisation (ici mokolo) qui est spécifiée par URL (**mokolo**.mobility.local).

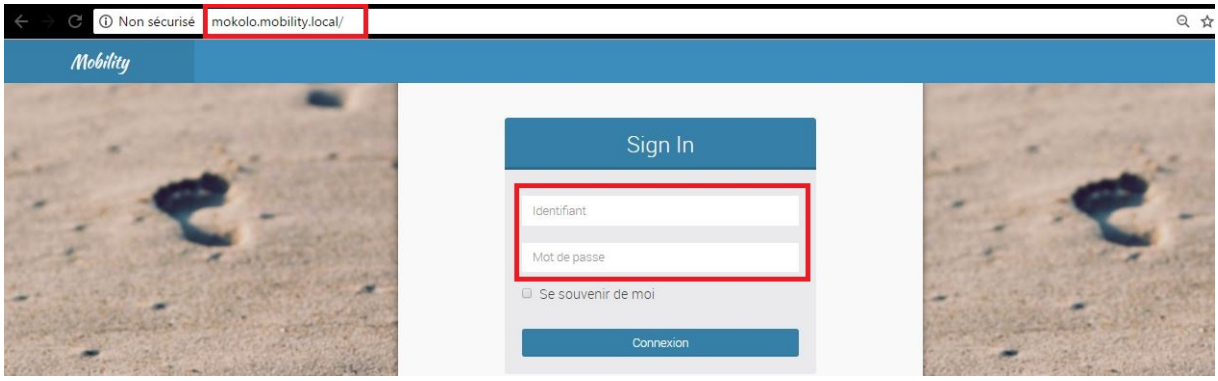

FIGURE 3.3 – Interface de connexion

Cette interface (figure 3.3) est accessible à tout le monde connaissant l'URL.

- **Permettre la création, la mise à jour et la suppression d'utilisateur dans une organisation.**

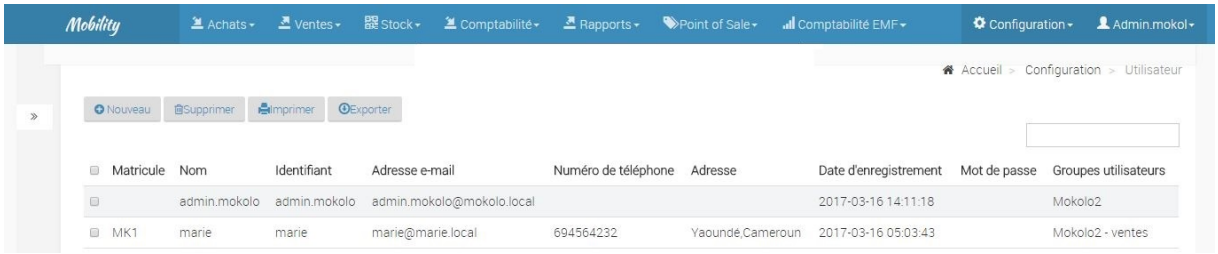

FIGURE 3.4 – Interface présentant les utilisateurs de l'organisation

La figure 3.4 présente la liste des utilisateurs appartenant à l'organisation (ici mokolo) ainsi que le groupe auquel chaque utilisateur appartient. En cliquant sur un utilisateur, on arrive sur l'interface d'édition de l'utilisateur (figure 3.5). Là, on peut modifier ses paramètres (nom d'utilisateur, mot de passe, etc.) mais surtout on peut l'affecter à un groupe.

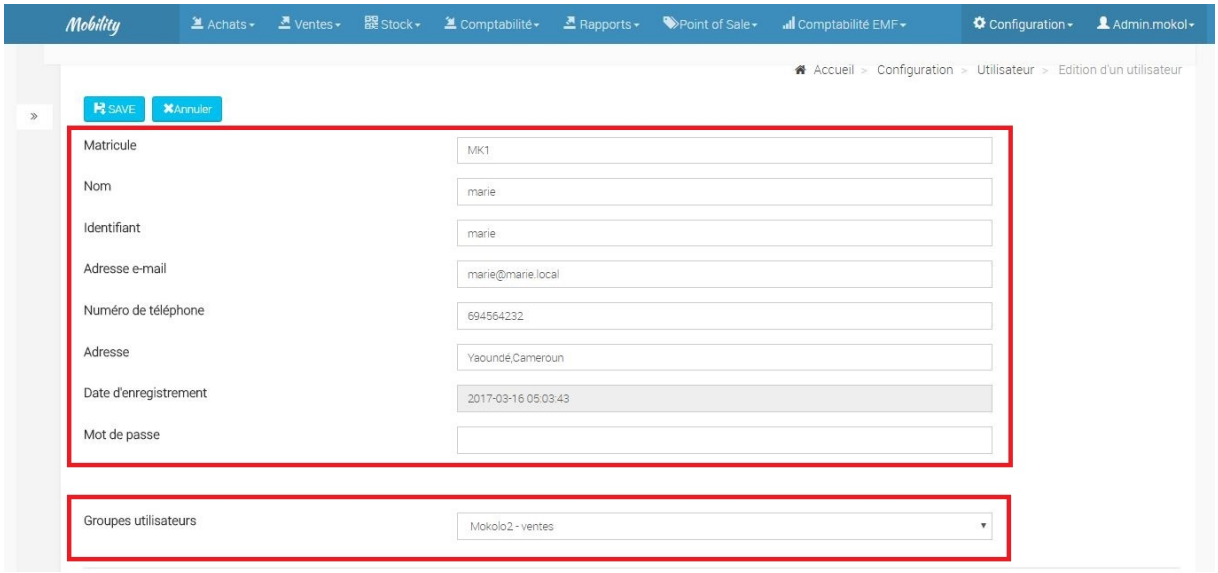

FIGURE 3.5 – Interface d'édition d'un utilisateur

- **Mettre les utilisateurs en groupe.**

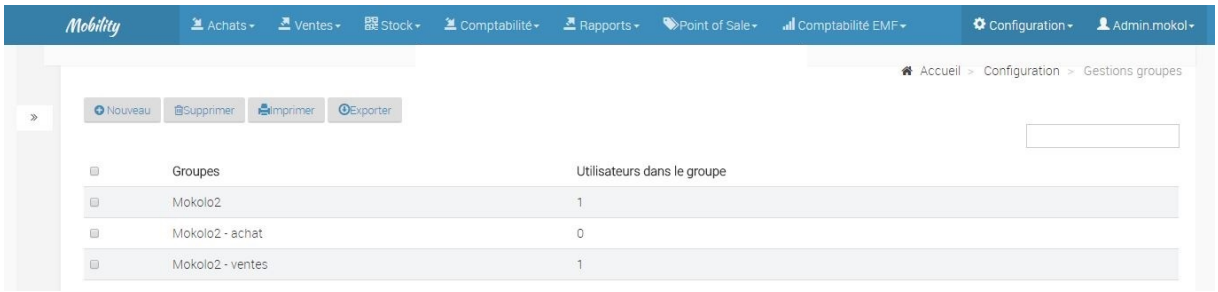

FIGURE 3.6 – Interface groupe d'utilisateur

La figure 3.6 présente la liste des groupes d'utilisateurs présents dans l'organisation et le nombre d'utilisateurs par groupe. En cliquant sur une organisation, on arrive sur l'interface d'édition du groupe (figure 3.7). Là, on peut juste changer son nom.

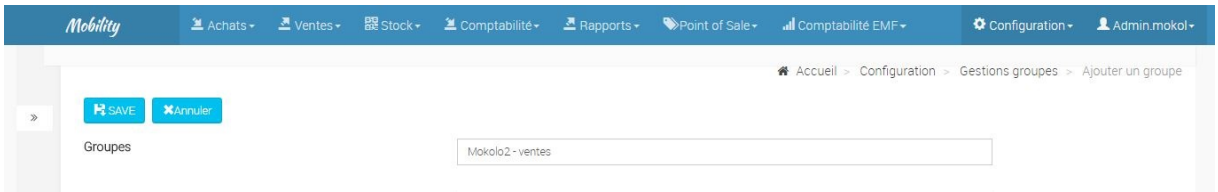

FIGURE 3.7 – Interface d'édition d'un groupe

Ces interfaces (figures 3.6 et 3.7) ne sont accessibles qu'aux utilisateurs ayant le rôle de super-utilisateur dans l'organisation.

- **Permettre l'affectation des droits d'accès aux groupes d'utilisateur.**
  Cette interface (figure 3.8), constituée de 5 principales parties permet d'affecter à un groupe d'utilisateur les permissions sur une ressource. Elle se présente comme suit :

  1. **Modules** : Représente la liste des services auxquels l'organisation a souscrit.
  2. **Fonctionnalités** : Présente les fonctionnalités qui sont comprises dans le service sélectionné.
  3. **Groupes** : Ce sont les groupes d'utilisateurs créés.
  4. **Fil d'ariane** : Constituée de trois parties (un module, une fonctionnalité, un groupe). Pour qu'une permission soit affectée ou calculée, il faut que ces trois paramètres soient définis.
  5. **Permissions** : Elles peuvent être dans deux états (Autorisée ou Refusée). Elles sont cinq (voir, modifier, créer/ajouter, supprimer, exporter/imprimer).

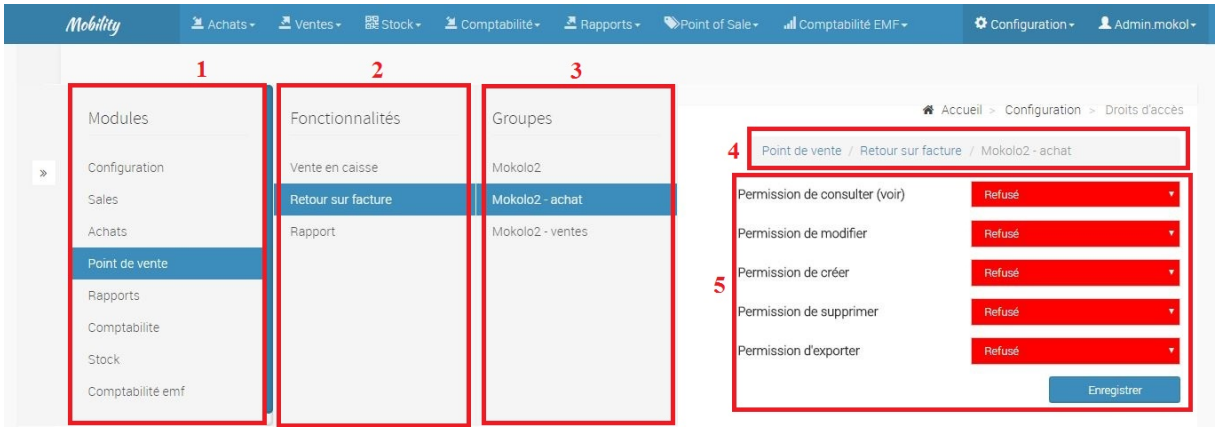

FIGURE 3.8 – Interface d'affectation des droits d'accès

Cette interface (figure 3.8) n'est accessible qu'aux utilisateurs ayant le rôle de super-utilisateur dans l'organisation.

- **Mettre en place le contrôle d'accès aux services souscrits une organisation.**
  La figure 3.9 présente l'interface d'accueil d'un utilisateur créé dans le système et à qui a été attribué les droits d'accès à des fonctionnalités des services Stock et Comptabilité.

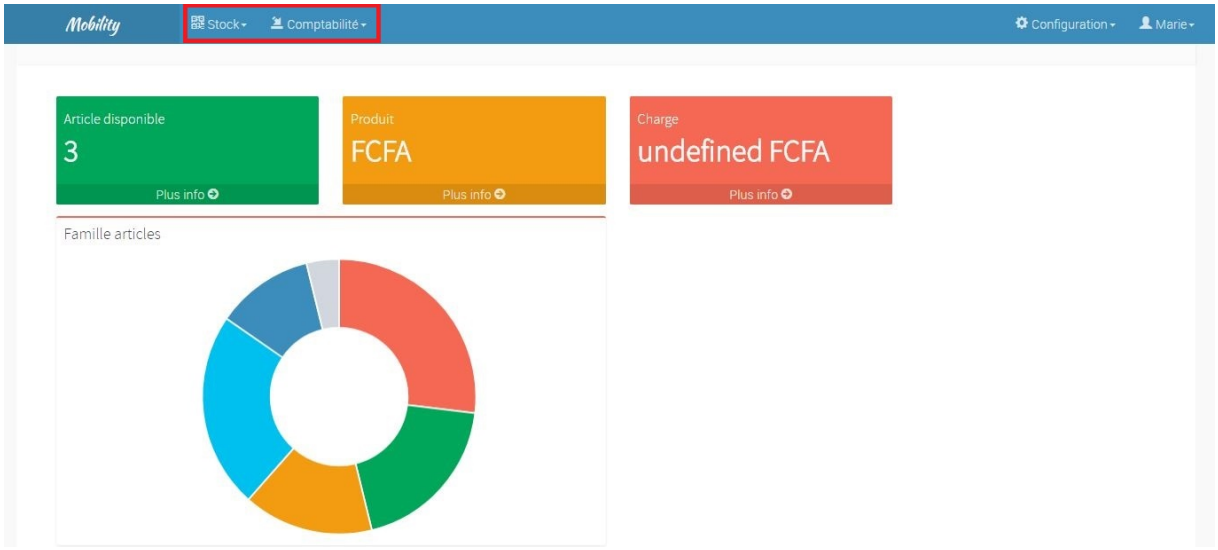

FIGURE 3.9 – Interface d'accueil d'un utilisateur

La figure 3.10 présente l'interface de profil. Cette interface permet aux utilisateurs de modifier certaines de leurs informations.

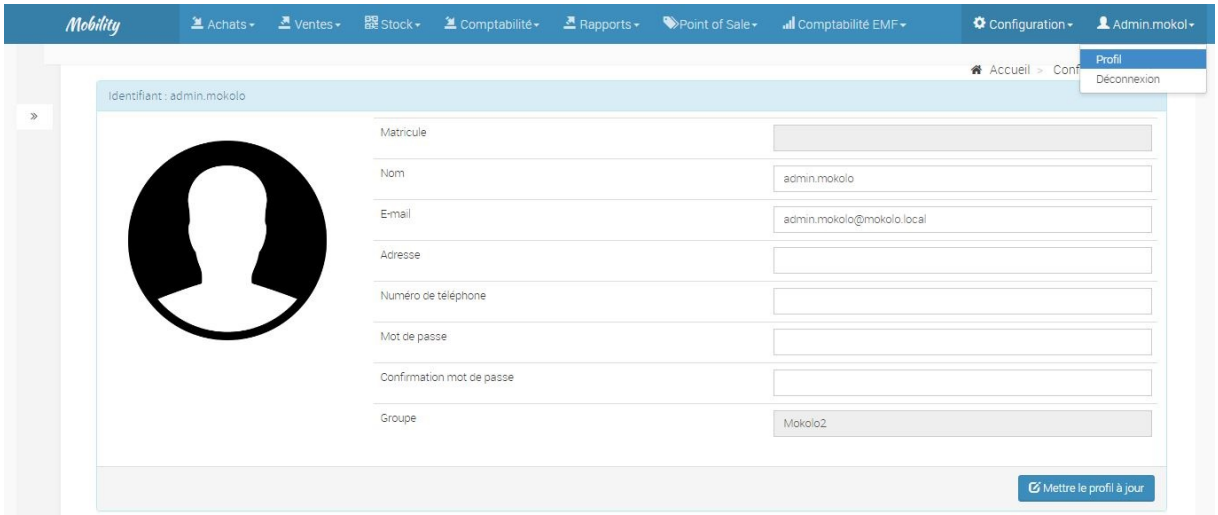

FIGURE 3.10 – Interface profil utilisateur

Ces interfaces (figures 3.9 et 3.10) sont accessibles à tous les utilisateurs authentifiés par le système.

3.3 Conclusion et recommandations

Dans le chapitre précédent, nous avons présenté cinq mesures de sécurité : la séparation des privilèges, la sécurisation des connexions réseaux, l'authentification de l'origine des requêtes émises, la réplication des données et le contrôle d'accès. L'implémentation du contrôle d'accès et de la séparation des privilèges sont les premiers pas pour la sécurisation de toute application manipulant des données sensibles, toutefois cette mesure doit être accompagnée par les autres pour renforcer la sécurité des données, de l'application et du serveur lui-même. Ainsi nous recommandons à l'entreprise IniMov de :

— Finaliser la mise en œuvre des mesures proposées au chapitre 2 ;
— Mener des tests de sécurité poussés pour vérifier la solidité la politique élaborée.
— Établir une charte sécurité utilisateur, pour sensibiliser les utilisateurs sur la manière de se comporter vis-à-vis du système.

Conclusion

Bilan

La croissance rapide du cloud computing ces dernières années a entraîné le développement d'une multitude de services associés. Ces services concernent l'utilisation de ressources informatiques à distance : applications, plateformes de développement et d'exécution, infrastructures. La diversité des parties prenantes dans les environnements cloud, peut impliquer des conséquences négatives sur la sécurité réseau du cloud. Pour se prémunir des menaces, les fournisseurs de services cloud ont adopté des mécanismes chiffrement des données stockées ; des mécanismes de sécurité réseau comme les pare-feu virtuels, les connexions sécurisés ; des mécanismes de contrôle d'accès aux applications hébergées.

Notre objectif était de proposer une politique de sécurité adaptée au système Mobility. Cet objectif se ramifiait en trois sous-objectifs, qui ont tous été atteint : le premier, était la réalisation de la cartographie des risques, effectuée sur la base d'une étude poussée du système ; le deuxième, la recherche et la sélection de mesures, effectuée sur la base de ce qui se fait actuellement dans le domaine de la science ; le troisième, la mise en œuvre de ces parades, grâces aux différents outils de développement.

La politique mise en œuvre est basée sur la séparation des privilèges, la sécurisation du protocole de communication, la réplication des données et le contrôle d'accès aux services de l'application. La politique élaborée a été implémentée puis testée dans le système Mobility, et les résultats obtenus sont encourageants. Ils témoignent de l'intérêt et de la faisabilité de notre approche, et ouvrent la voie à plusieurs perspectives d'amélioration de ce travail.

Perspectives

Pour le temps imparti, nous avons pu mettre en place un système de contrôle d'accès et de séparation des privilèges permettant de protéger les services de Mobility et l'accès

aux données des utilisateurs. Cependant, la conception de Mobility et le cadre dans lequel il a été implémenté (le CMS Joomla !), induisent certaines limites au travail réalisé. Les perspectives de ce travail seraient de :

- Rendre la solution portable et réutilisable, c'est-à-dire la détacher du cadre fixé par le CMS pour qu'elle soit applicable sur d'autres systèmes ;
- Intégrer un mécanisme de chiffrement pour les données critiques avant leur stockage.

Bibiographie

[1] “Access Control List,” Wikipédia. 17-Jan-2017.

[2] J. Michel Embe, “Approches formelles de mise en oeuvre de politiques de contrôle d’accès pour des applications basées sur une architecture orientée services,” UNIVERSITÉ DE SHERBROOKE, École Doctorale MSTIC, 2012.

[3] “Architecture logicielle,” Wikipédia. 30-Dec-2016.

[4] F. CHAMBON, “ARCHITECTURE REST & WEB SERVICES.” 14-Jan-2014.

[5] M. Armbrust et al., “A view of cloud computing,” Communications of the ACM, vol. 53, no. 4, p. 50, Apr. 2010.

[6] “Cloud Computing and SOA.” [Online]. Available : https ://www.mitre.org/sites/default/files/p [Consulté le : 17-Apr-2017].

[7] Z. Al Haddad, M. Hanoune, and A. Mamouni, Cloud Computing et Sécurité : Approches et Solutions, vol. 30. 2015.

[8] K. Won, “Cloud Computing : Today and Tomorrow,” Journal of Object Technology, vol. 8, no. 1, pp. 65–72, Feb. 2009.

[9] “A capability-based security approach to manage access control in the Internet of Things - ScienceDirect.” [Online]. Available : http ://www.sciencedirect.com/science/article/pii/S0 [Accessed : 03-Apr-2017].

[10] “Cloud computing : state-of-the-art and research challenges | SpringerLink.” [Online]. Available : https ://link.springer.com/article/10.1007

[11] “Service web,” Wikipédia. [Online]. Available : https ://fr.wikipedia.org/wiki/Service_web [Consulté le : 03-Apr-2017].

[12] G. Yann-Gaël, “Cours de LOG4430 (Département de génie informatique et de génie logiciel École Polytechnique de Montréal) : Architecture logicielle et conception avancée.” Département de génie informatique et de génie logiciel École Polytechnique de Montréal, 2012.

[13] G. Gilles and H. Fred, “Des cas d’utilisation en UML à la gestion de rôles dans un système d’information. (PDF Download Available),” in ResearchGate, 2000, p. 14.

[14] "EBIOS : la méthode de gestion des risques SSI Un outil simple et puissant." Agence Nationale de Sécurité des Systèmes d'Information (ANSSI).

[15] P. Thibaut, "ÉVALUATION ET ANALYSE DES MÉCANISMES DE SÉCURITÉ DES RÉSEAUX DANS LES INFRASTRUCTURES VIRTUELLES DE CLOUD COMPUTING," FÉDÉRALE TOULOUSE MIDI-PYRÉNÉES, Institut National Polytechnique de Toulouse, 2015.

[16] H. Guillaume, "Gestion des identités et des accès : concepts et états de l'art." 09-Oct-2013.

[17] "Guide pour l'élaboration d'une politique de sécurité de système d'information." bureau conseil de la DCSSI, 03-Mar-2004.

[18] "Joomla CMS : Page principale." [Online]. Available : https ://api.joomla.fr/joomla3/. [Consulté le : 01-Mar-2017].

[19] "Le Role Based Access Control," synbioz.com. [Online]. Available : https ://www.synbioz.com/l [Consulté le : 28-Feb-2017].

[20] "Les base absolues de la manière dont un composant fonctionne — Joomla ! Documentation." [Online]. Available : https ://docs.joomla.org/Absolute_Basics_of_How_a_Compon [Consulté le : 15-Feb-2017].

[21] Alcyonix, "L'insuffisance du modèle RBAC," Alcyonix, 04-Sep-2011. [Online]. Available : https ://www.alcyonix.com/articles/linsuffisance-du-modele-rbac. [Consulté le : 28-Feb-2017].

[22] "Memoire Online - Le cloud computing quel impact organisationnel pour les équipes informatiques des systèmes d'information ? - Eric BERTHELOT," Memoire Online. [Online]. Available : http ://www.memoireonline.com/02/12/5216/m_Le-cloud-computing-quel-impact-organisationnel-pour-les-equipes-informatiques-des-systemes-d-infor3.html. [Consulté le : 26-Apr-2017].

[23] "Modèles Principaux de contrôle d'accès - ppt télécharger." [Online]. Available : http ://slideplayer.fr/slide/1666809/. [Consulté le : 30-May-2017].

[24] "OAuth : Comment ça marche ? | Le blog Sodifrance Netapsys." .

[25] "Politique de Sécurité des Systèmes d'Information." Direction des Technologies de l'Information.

[26] F.-A. Blain, "Présentation générale des ERP et leur architecture modulaire," Developpez.com. [Online]. Available : http ://fablain.developpez.com/tutoriel/presenterp/. [Consulté le : 13-Feb-2017].

[27] "Prudents, précurseurs et visionnaires." [Online]. Available : http ://www.clear-cloud-services.fr/archives/definition-etudes/le-livre-blanc-afdel/prudents-precurseurs-et-visionnaires.

[28] "Quels sont les risques de sécurité majeurs du cloud computing ? Une étude du CSA en révèle douze," Developpez.com. [Online]. Available : http ://cloud-computing.developpez.c sont-les-risques-de-securite-majeurs-du-cloud-computing-Une-etude-du-CSA-en-revele-douze/. [Consulté le : 31-May-2017].

[29] "Réplication de données temps réel et continue dans un cluster miroir actif-passif," Evidian, 30-Mar-2014. [Online]. Available : https ://www.evidian.com/fr/produits/haute-disponibilite-logiciel-clustering-application/replication-de-donnees-temps-reel-continue-cluster-actif-passif/. [Consulté le : 22-Jun-2017].

[30] J.-P. Figer, "REST, un style d'architecture universel.," www.figer.com, août-2005. [Online]. Available : http ://www.figer.com/publications/REST.htm#.WQGgxfk1_IU. [Consulté le : 27-Apr-2017].

[31] B. Mathieu, "Sécurité des systèmes d'exploitation répartis : architecture décentralisée de méta-politique pour l'administration du contrôle d'accès obligatoire," Informatique, UNIVERSITÉ D'ORLÉANS, UNIVERSITÉ D'ORLÉANS, 2006.

[32] "Service Technology Magazine," Service Technology Magazine. [Online]. Available : http ://servicetechmag.com/. [Consulté le : 17-Apr-2017].

[33] P. Aneta, "Spécification UML du contrôle d'accès dans les systèmes d'information : Une approche coopérative de la conception des rôles dans un modèle RBAC," Ecole Polytechnique Silesienne, UNIVERSITÉ D'Artois, 2013.

[34] G. Joseph and G. David, "UML 2 ANALYSE ET CONCEPTION." Dunod, 2008.

[35] M. E. Shin and G.-J. Ahn, "UML-Based Representation of Role-Based Access Control," George Mason University, 2000.

[36] P. Olivier, L. Maryline, and G. Sylvain, "Une architecture de gestion efficace du contrôle d'accès." [Online]. Available : https ://pdfs.semanticscholar.org/4bd0/8329a3d6c42f9393c1 [Consulté le : 21-Feb-2017].

[37] "Une étude de DEFCON révèle l'ampleur du piratage sur le Cloud > Mag-Securs > Communiqués." [Online]. Available : http ://www.mag-securs.com/communiques/id/27666/une-etude-de-defcon-revele-l-ampleur-du-piratage-sur-le-cloud.aspx. [Consulté le : 16-Jun-2017].

[38] M. Jérémy, "Un processus formel d'intégration de politiques de contrôle d'accès dans les systèmes d'information," HAL. [Online]. Available : https ://tel.archives-ouvertes.fr/tel-00674865/document. [Consulté le : 21-Feb-2017].

[39] D. A. B. Fernandes, L. F. B. Soares, J. V. Gomes, M. M. Freire, and P. R. M. Inácio, Security issues in cloud environments : a survey, Int. J. Inf. Secur., vol. 13, no. 2, pp. 113–170, Apr. 2014.

[40] Dictionnaire de l'Académie française, neuvième édition. http ://atilf.atilf.fr/academie9.htm.30

[41] Heartbleed Bug. http ://heartbleed.com. 30/06/15.

[42] Matt Bishop : Analysis of the ILOVEYOU Worm. 2000.

[43] Ludovic Mé : Sécurité des systèmes d'information. Hermès, 2006.