République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Blida1

Faculté de Math et Informatique
Département Informatique

Mémoire de fin d'étude en vue de l'obtention
Du diplôme de Master en Informatique
Spécialité : SIR + IL

Implémentation d'un système de contrôle d'accès basé sur le modèle RBAC pour un Cloud Computing privé

Présenté par : M^r^ Sofiane HOUBAN.
M^r^ Elhadi RAID.

Devant le jury :

M^me^ BOUSTIA Narhimene	Université Blida1	Président
M^me^ BACHA Sihem	Université Blida1	Examinateur
M^me^ GHEBGHOUB Yasmina	Université Blida1	Promoteur
M^r^ BOUZOURINE Abd-errezzek	Direction de moudjahidine	Co-Promoteur

Année Universitaire 2019-2020.

Dédicaces

Dédicaces

En tout premier lieu, je remercie le bon Dieu, tout puissant, de m'avoir donné de la force et de l'audace pour dépasser toutes les difficultés. Permis de mener à bien ce travail. Pour avoir bien voulu juger ce travail.

Je tiens à dédier ce travail à :

Mes chers parents pour leurs sacrifices durables.

Mon épouse qui m'a soutenu durant toute cette période.

Mes petites princesses que je les aime très fort

Mes Sœurs et mon frère

A Toute ma famille

A Mon binôme

A touts mes amis et collègues.

Abd-Elhadi

Dédicaces

En tout premier lieu, je remercie le bon Dieu, tout puissant, de m'avoir donné de la force et de l'audace pour dépasser toutes les difficultés. Permis de mener à bien ce travail. Pour avoir bien voulu juger ce travail.

Je tiens à dédier ce travail à :

Mon épouse qui m'a soutenu durant toute cette période.

Mes chers parents et ma belle famille

Mes petites princesses et mon petit prince que je les aime très fort

Mes Sœurs, mes frères et mon beau frère

Toute ma famille

A Mon binôme, touts mes amis et collègues.

Sofiane

Remerciement

Remerciements.

Cette expérience professionnelle et personnelle qui vient d'être finalisée par ce mémoire. Qui n'aurait pas été réalisé sans le savoir et le soutien de nombreuses personnes. Nous tenons ainsi à remercier en quelques lignes tous ceux qui, de près ou de loin ont contribué à ce travail, en espérant n'oublier personne.

Nous remercions en premier lieu ALLAH le tout puissant pour toute la volonté et le courage qu'il m'a donné pour l'achèvement de ce travail, il a été et sera toujours à côté de nous pour réussir à terminer n'importe quel travail.

Ce mémoire n'a pas pu être réalisé sans le support continu de notre promotrice : M^{me} GHEBGHOUB .Y et notre Co-promoteur BOUZOURINE .A. Nous désirons leur adresser un remerciement tout particulier pour leurs précieux commentaires et leurs conseils pertinents qui nous ont grandement aidés tout au long des différentes étapes menant à l'élaboration de ce mémoire.

Nos respects et reconnaissances sont adressés à M^{me} BOUSTIA Narhimene qui nous a fait l'honneur de présider le jury de soutenance.

Nous exprimons nos remerciements à BACHA Sihem qui nous a fait l'honneur d'examiner ce mémoire.

A touts mes enseignants et administrateurs au département des maths et informatique de l'Université de Blida1 pour leurs aides et compréhension.

Résumés

ملخص:

مع تطور المجتمع، أحدث الإنترنت انفجارًا ملحوظًا في البيانات وموارد الأجهزة والبرامج التي انتشرت في جميع أنحاء العالم. تتم إدارة هذه المساحة الهائلة من المعلومات والموارد ومعالجتها بواسطة العديد من أدوات والتقنيات التكنولوجية للمعلومات. وبالتالي ، فإن الحوسبة السحابية هي بنية تحتية أثبتت مكانتها في مجال تكنولوجيات المعلومات. الحوسبة السحابية هي بيئة لتقديم الموارد والخدمات عند الطلب عبر الإنترنت. في هذا الفضاء الهائل من المعلومات، يكون المتدخلون في بعض الأحيان مجهولين وغير مجسدين. يجب أن تتحد مفاهيم وتقنيات أمن تكنولوجيات المعلومات لتمكين التحكم في الدخول الآمن. يعد التحكم في الدخول مكونًا أساسيًا لتأمين أي نظام كمبيوتر، وقد تم اقتراح العديد من نماذج التحكم في الدخول في الأدبيات.

في مشروعنا، درسنا أولاً مفاهيم هذه المشكلة، ثم في مرحلة التصميم، قمنا بتكييف طريقة **UML** لتحليل وتصميم النظام بأكمله. هذا الأخير يعتمد على نموذج **RBAC** في التحكم في الدخول. تتكون سياستنا الأمنية من ثلاث وحدات أساسية: المصادقة على المستخدم، التحقق من التفويضات، وإدارة النموذج RBAC. تم تجسيد المشروع على حوسبة سحابية خاصة تتوافق مع هيئة الدراسة لدينا (وزارة المجاهدين). أخيرًا ننتهي بعرض و توضيح المشروع واختبار للنظام.

الكلمات الدالة :

الأمن، التكنولوجيات، نظم المعلومات، الحوسبة السحابية ، السياسات، النموذج، التحكم في الدخول، **RBAB** **.UML**

ABSTRACT:

With the development of society, the internet has brought about a remarkable explosion of data and resources (hardware and software) that spread throughout the universe. This large space of informations and resources is managed and manipulated by several IT tools and technologies. Thus, Cloud Computing is an infrastructure that has proven its value in the field of information technology. Coud Computing is an environment for the delivery of resources and services on demand over the internet. In this enormous space of information, the interlocutors are sometimes unknown and dematerialized. The concepts and technologies of IT security must combine to enable reliable and secure access control. Access control is an essential component for securing any IT system and several access control models have been proposed in the literature.

In our project, we first studied the concepts of this problem, then in the modeling stage, we adapted the UML method for the analysis and design of the entire system. This latter is based on the RBAC access control model. Our security policy consists of three essential modules: user authentication, authorization verification and administration of the RBAC policy. The implementation was carried out on a private Cloud Computing that we proposed and which is suitable for our study organization (Ministry of Mujahedins). Finally we end with a presentation and a test of the system.

Keywords :

Security, Technology, Computer Systems, Cloud Computing, Policy, Model, Access Control, RBAC, UML.

RESUME :

Avec le développement de la société, l'internet a provoqué une explosion remarquable de données et de ressources matériels et logiciels qui s'étendent vers tout l'univers. Cet espace large d'informations et de ressources est géré et manipulé par plusieurs outils et technologies informatique. Ainsi, Le Cloud Computing est une infrastructure qui a prouvé sa valeur dans le domaine des technologies de l'information. Le Coud Computing est un environnement pour la livraison de ressources et de services à la demande sur internet. Dans cet espace d'informations, les interlocuteurs sont parfois inconnus et dématérialisés. Les concepts et les technologies de la sécurité informatique doivent se combiner pour permettre un contrôle d'accès fiable et sécurisé. Le contrôle d'accès est une composante essentielle pour la sécurisation de tout système informatique et plusieurs modèles de contrôle d'accès ont été proposés dans la littérature.

Dans notre projet, nous avons étudié d'abord les concepts de cette problématique, puis dans l'étape de modélisation, nous avons adapté la méthode UML pour l'analyse et la conception de tout le système. Ce dernier qui est basé sur le modèle de contrôle d'accès RBAC. Notre politique de sécurité est constituée de trois modules essentiels : l'authentification des utilisateurs, vérification des autorisations et administration de la politique RBAC. L'implémentation a été réalisée sur un Cloud Computing privé que nous avons proposé et qui convient à notre organisme d'étude (Ministère des moudjahidines). Enfin nous finissons par une présentation et un test du système.

Mots-clés :

Sécurité, technologie, Système informatique, Cloud Computing, modèle, Politique, Contrôle d'accès, RBAC, UML.

La liste des figures

La liste des Tableaux

La liste des acronymes

Acronymes	Description
ABAC	**A**ttribute **B**ased **A**ccess **C**ontrol
ACL	**A**ccess **C**ontrol **L**ist
AD	**A**ctive **D**irectory
AD FS	**A**ctive **D**irectory **F**ederation **S**ervices
AD LDS	**A**ctive **D**irectory **L**ightweight **D**irectory **S**ervices
AD RMS	**A**ctive **D**irectory **R**ights **M**anagement **S**ervices
API	**A**pplication **P**rogramming **I**nterfaces
ASIC	**A**pplication **S**pecific **I**ntegrated **C**ircuit
BLP	Modèle de confidentialité de **B**ell et **L**a**P**adula
CaaS	**C**ommunication **A**S **A** **S**ervice
CID	**C**onfidentialité **I**ntégrité et **D**isponibilité
CSP	**C**loud **S**ervice **P**rovider
CSU	**C**loud **S**ervice **U**ser
DAC	**D**iscretionary **A**ccess **C**ontrol
DCL	**D**iagramme de **C**lasse
DCU	**D**igramme **C**as **U**tilisation
DDoS	**D**istributed **D**enial **o**f **S**ervice
DHCP	**D**igital **H**igh-Bandwidth **C**ontent **P**rotection
DNS	**D**omain **N**ame **S**ystem
DoS	**D**enial **o**f **S**ervice
DSD	**D**ynamic **S**eparation of **D**uties
DTE	**D**omain and **T**ype **E**nforcement
EDM	**E**ntity **D**ata **M**odel
EBIOS	**E**xpression des **B**esoins et **I**dentification des **O**bjectifs de **S**écurité
FreeBSD	**Free** **B**erkeley **S**oftware **D**istribution
GEORBAC	**G**eography **B**ased **A**ccess **C**ontrol
GPS	**G**lobal **P**ositioning **S**ystem
GTRBAC	**G**eneralized **T**empora **B**ased **A**ccess **C**ontrol
IaaS	**I**nfrastructure **AS** A **S**ervice
IAM	**I**dentity and **A**ccess **M**anagement
IBAC	**I**dentity **B**ased **A**ccess **C**ontrol
IBM	**I**nternational **B**usiness **M**achines
IDE	**I**ntegrated **D**evelopment **E**nvironment
IIS	**I**nternet **I**nformation **S**ervices
iOS	**i**Phone **O**perating **S**ystem
ISO/IEC	**I**nternational **O**rganization for **S**tandardization / **I**nternational

	Electrotechnical **C**ommission
KVM	**K**ernel-based **V**irtual **M**achine
LAN	**L**ocal **A**rea **N**etwork
LrBAC	**L**ocal **R**ole **B**ased **A**ccess **C**ontrol
MAC	**M**andatory **A**ccess **C**ontrol
MPLS	**M**ulti **P**rotocol **L**abel **S**witching
MVC	**M**odel **V**iew **C**ontroller
NET	**Net**work ou Internet
NIST	**N**ational **I**nstitut of **S**tandards and **T**echnology
OR-BAC	**Or**ganization **B**ased **A**ccess **C**ontrol
ORM	**Object-Relational Mapping**
OS	**O**perating **S**ystem
PaaS	**P**latform **A**s **A** **S**ervice
PHP	Hypertext Preprocessor
QoS	**Q**ualité **o**f **S**ervice
RBAC	**R**ole **B**ased **A**ccess **C**ontrol
SaaS	**S**oftware **A**s **A** **S**ervice
SECaaS	**Sec**urity **A**S **A** **S**ervice
SLA	**S**ervice **L**evel **A**greement
SQL	**S**tructured **Q**uery **L**angage
SSD	**S**tatic **S**eparation of **D**uties
TaaS	**T**ransport **A**S **A** **S**ervice
TBAC	**T**ask **B**ased **A**ccess **C**ontrol
TCP/IP	**T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol
TMAC	**T**ask **M**andatory **A**ccess **C**ontrol
TR-BAC	**T**asck **R**ole **B**ased **A**ccess **C**ontrol
UML	**U**nified **M**odeling **L**anguage
USB	**U**niversal **S**erial **B**us
VM	**V**irtual **M**achine
VPC	**V**irtual **P**rivate **C**loud
VPN	**V**irtual **P**rivate **N**etwork
WAN	**W**ide **A**rea **N**etwork
Web	**World Wide Web**
Wi-Fi	**Wi**reless **Fi**delity
XaaS	Anything ou Everything **A**S **A** **S**ervice
XSS	**C**ross **S**ite **S**cripting
WIC	**W**an **I**nterface **C**ard
HWIC	**H**igh **W**an **I**nterface **C**ard

Sommaire

Sommaire

Chapitre 02 : Sécurité dans le Cloud Computing

Chapitre 03 : Contrôles d'accès au système d'information

Partie 02 : Modélisation et implémentation de solution

Chapitre 04 : Modélisation d'un système de contrôles d'accées pour un Cloud Computing Privé

Chapitre 05 : Implémentation de solution

Introduction générale

Actuellement, l'évolution de la société moderne dépend strictement du domaine des technologies de l'information et de la communication, ces technologies révolutionnent tous les secteurs, qu'ils soient industriels, commerciaux, administratifs voire même la vie privée des individus. Le développement du concept de ***Cloud Computing*** (informatique en nuage) a constitué une avancée majeure vers la généralisation des technologies informatique dans le monde. La notion de ***Cloud Computing*** rassemble sous un même nom un ensemble de concepts dans lesquels on retrouve la notion d'environnement ouvert (système en interaction permanant avec l'extérieure). Dans la philosophie moderne des systèmes informatiques, l'interaction et l'échange des informations est inévitable. La contrepartie de cette ouverture vient de l'éventualité des menaces ou les risques de sécurité qu'elle engendre. Les conséquences d'une violation de la sécurité risquent de provoquer des situations graves et d'impacter le fonctionnement des organisations. En terme de sécurité des données, plusieurs critères peuvent être énumérés, tels que la disponibilité, l'intégrité ou encore la confidentialité.

La garantie de la sécurité devient alors l'un des éléments décisifs dans le développement et le fonctionnement des systèmes informatiques modernes, et plus particulièrement les politiques de contrôle d'accès qui constituent l'élément essentiel de la sécurité des données pour la plupart des systèmes industriels actuels.

Notre objectif est de développer un système de contrôle d'accès basé sur le modèle **RBAC** ***(Role Based Access Control)*** pour assurer la sécurité des données dans un Cloud privé. Nous avons déployé le contrôle d'accès au niveau du ministère des moudjahidine et des ayants droit. Ce ministère dispose d'une administration centrale (La direction centrale) qui gère un ensemble des directions opérationnelles (Une direction dans chaque Wilaya) et un ensemble des centres de repos (15 Centres sur le territoire national).

Notre mémoire commence par une première partie dédié à l'état de l'art. Dans le premier chapitre de cette partie nous présentons des notions générales sur la sécurité informatique : Définition, propriétés, menace, vulnérabilité, attaques et processus de

sécurité. Le deuxième chapitre traite la sécurité dans le ***Cloud Computing*** : Caractéristiques, modèles de services, typologies et sécurité. Le troisième chapitre présente une analyse des différentes politiques de contrôle d'accès largement utilisées dans le monde de sécurité informatique : En commençant par les modèles classiques **DAC** ***(Discretionary Access Control)*** et **MAC** ***(Mandatory Access Control)***, après les modèles à base de rôle **RBAC** ***(Role-Based Access Control)*** et à base de taches ***TBAC (Task Based Access Control)***, ensuite nous complétons avec des modèles un peu avancés comme **OR-BAC** à base d'organisation ***(Organization Based Access Control)*** et **CBAC** à base de contexte ***(Context Based Access Control)***, nous concentrons le focus dans ce chapitre sur leurs principes de fonctionnement et leurs domaines d'utilisation.

La deuxième partie de ce document comprend un premier chapitre qui commence par la modélisation du système en s'appuyant sur l'approche **UML**, ensuite la présentation de l'architecture de notre système de sécurité basé sur la politique **RBAC** en exposant ses différents modules. Nous terminons le chapitre par une proposition d'architectures physiques de réseau pour l'implémentation du Cloud privé.

Un deuxième chapitre est consacré à la mise en œuvre de notre projet, nous décrivons à travers ce chapitre les principaux outils de développements, l'implémentation de notre système de contrôle d'accès dans un ***Cloud Computing*** privé qui correspond à notre organisme d'étude et nous finissons par une présentation et un test du système.

Finalement, une conclusion qui comporte les apports de notre travail ainsi que les perspectives envisagées.

Partie 01
Etat de l'art

Chapitre 01
Sécurité Informatique

1. Introduction :

Chaque ordinateur connecté à un réseau informatique ou internet est susceptible d'être victime d'une attaque d'un pirate informatique. Ainsi, pour une entreprise, toute sorte d'attaque intentionnelle ou accidentelle qui touche ses ressources est inacceptable. Pour cette raison la sécurité informatique est très importante pour se protéger de ces attaques réseaux.

Dans ce chapitre, nous faisons un survol des notions de sécurité informatique, et nous Allons montrer la terminologie de base de la sécurité informatique. Ensuite on passe aux différentes menaces informatiques, la vulnérabilité et les attaques. On termine notre le chapitre par la description d'un processus de protection et une conclusion.

2. Définition de la sécurité informatique :

« La sécurité informatique, d'une manière générale, consiste à s'assurer que les ressources matérielles et logicielles d'une organisation sont uniquement utilisées dans le cadre prévu » **[7]**.

« La sécurité des systèmes informatiques se cantonne généralement à garantir les droits d'accès aux données et ressources d'un système en mettant en place des mécanismes d'authentification et de contrôle permettant d'assurer que les utilisateurs des dites ressources possèdent uniquement les droits qui leurs ont été octroyés » **[6].**

Donc, la sécurité informatique est un processus continu de protection des ressources, dont son objectif est d'empêcher tout accès, modification et utilisation non autorisée des ressources et garantir leurs utilisations dans le cadre prévu.

3. Propriétés de sécurité informatique:

La notion de sécurité informatique fait référence à la propriété d'un système, d'un service ou d'une entité. Elle s'exprime le plus souvent par les objectifs de sécurité suivants **(*Figure 1.2*)** :

3.1. Confidentialité [1]:

C'est le maintien du secret des informations peut être vue comme la « protection des données contre une divulgation non autorisée ».

Il existe deux types d'actions complémentaires permettant d'assurer la confidentialité des données :

- limiter et contrôler leurs accès afin que seules les personnes habilitées à les lire ou à les modifier puissent le faire.
- les rendre inintelligibles en les chiffrant de telle sorte que les personnes qui ne sont pas autorisées à les déchiffrer ne puissent les utiliser.

3.2. Intégrité :

Le critère d'**intégrité** des ressources physiques et logiques (équipements, données, traitements, transactions, services) est relatif au fait qu'elles n'ont pas été détruites (altération totale) ou modifiées (altération partielle) à l'insu de leurs propriétaires tant de manière intentionnelle qu'accidentelle. Une fonction de sécurité appliquée à une ressource pour contribuer à préserver son intégrité, permettra de la protéger plus ou moins efficacement contre une menace de corruption ou de destruction **[1].**

3.3. Disponibilité :

La **disponibilité** d'une ressource est relative à la période de temps pendant laquelle le service offert est opérationnel. Le volume potentiel de travail susceptible d'être pris en charge durant la période de disponibilité d'un service, détermine la **capacité** d'une ressource à être utilisée. Il ne suffit pas qu'une ressource soit disponible, elle doit pouvoir être utilisable avec des temps de réponse acceptables. Sa disponibilité est indissociable de sa capacité à être **accessible** par l'ensemble des ayants droit **[1].**

3.4. Non Répudiation [1]:

La **non-répudiation** est le fait de ne pouvoir nier ou rejeter qu'un événement (action, transaction) a eu lieu. À ce critère de sécurité peuvent être associées les notions d'imputabilité, de ***traçabilité*** ou encore parfois ***d'auditabilité*.**

- L'**imputabilité** se définit par l'attribution d'une action (un événement) à une entité déterminée (ressource, personne).
- La **traçabilité** permet de suivre la trace numérique laissée par la réalisation d'un événement (message électronique, transaction commerciale, transfert de données…).
- L'**auditabilité** se définit par la capacité d'un système à garantir la présence d'informations nécessaires à une analyse ultérieure d'un événement (courant ou exceptionnel) effectuée dans le cadre de procédures de contrôle spécifiques et d'audit.

3.5. Identification et Authentification [1]:

L'authentification doit permettre de vérifier l'identité d'une entité afin de s'assurer entre autres, de l'authenticité de celle-ci. Pour cela, l'entité devra prouver son identité, le plus souvent en donnant une information spécifique qu'elle est censée être seule à détenir telle que, par exemple, un mot de passe ou une empreinte biométrique. Ensuite on passe au contrôle d'accès (voir chapitre 03) selon la figure ci-dessous :

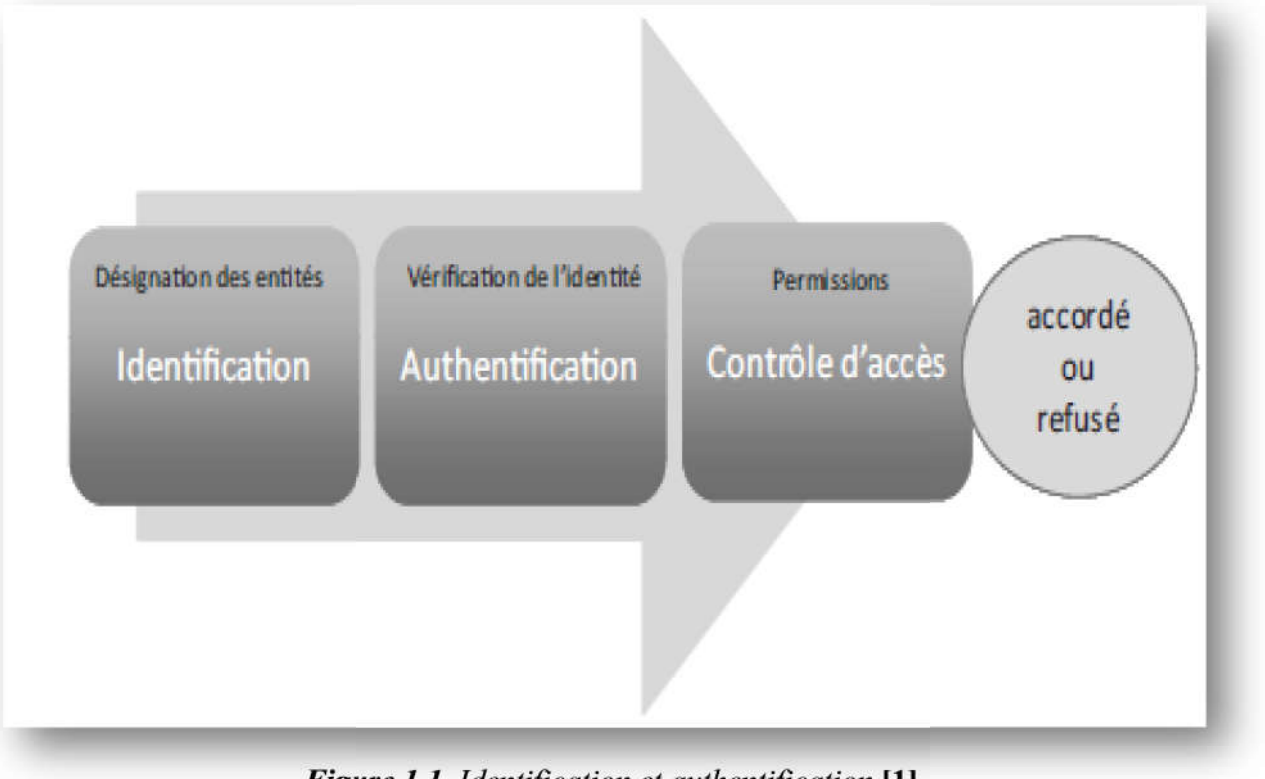

Figure 1.1. *Identification et authentification* **[1].**

En concluant on peut schématiser les critères de sécurité par la figure suivante :

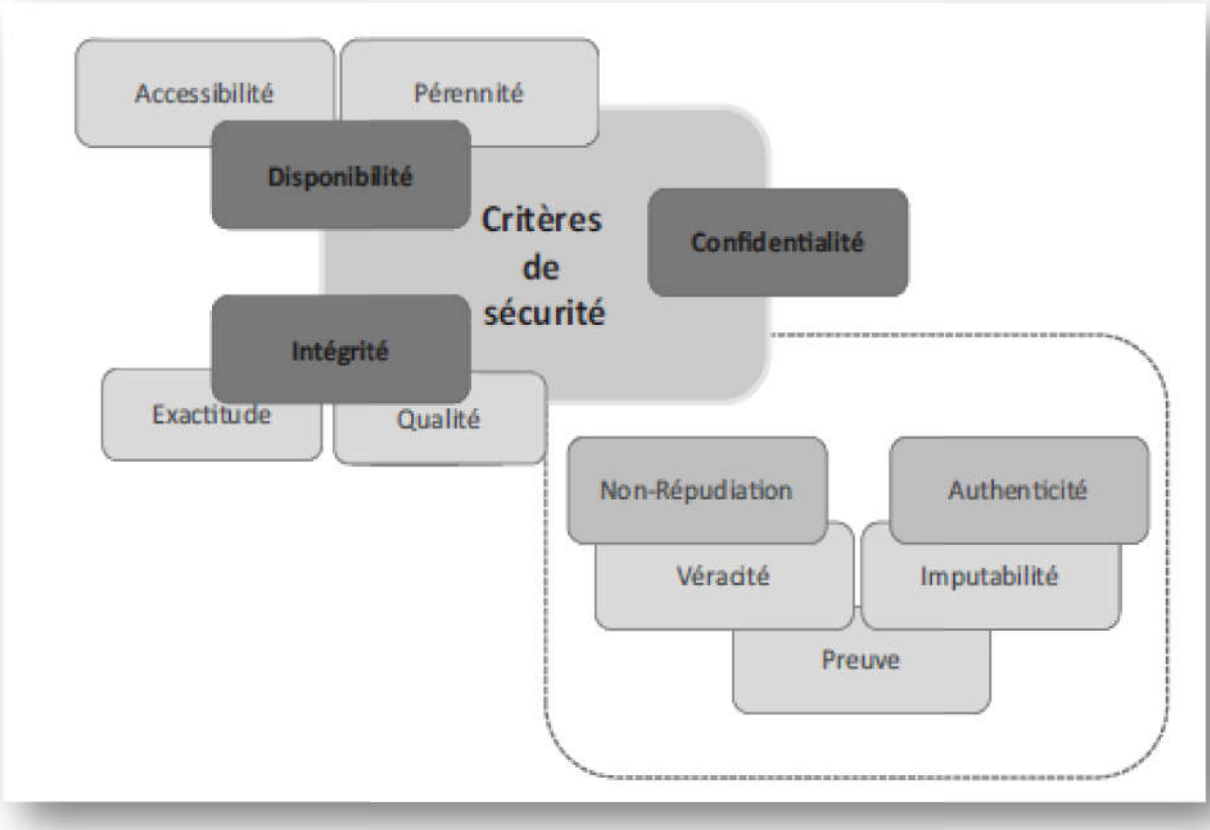

Figure 1.2. *Critères de sécurité* **[1].**

4. Les menaces informatiques :

La menace informatique représente le type d'actions susceptibles de nuire dans l'absolu à un système informatique. En termes de sécurité informatique les menaces peuvent être le résultat de diverses actions en provenance de plusieurs origines **[2]**:

4.1. Origine opérationnel :

Ces menaces sont liées à un état du système à un moment donné. Elles peuvent être le résultat d'un bogue logiciel (Buffer Overflows, format string …etc.), d'une erreur de filtrage des entrées utilisateur d'un dysfonctionnement de la logique de traitement ou d'une erreur de configuration.

4.2. Origine physique :

Elles peuvent être d'origine accidentelle, naturelle ou criminelle. On peut citer notamment les désastres naturels, les pannes ou casses matérielles, le feu ou les coupures électriques.

4.3. Origine Humaine :

Ces menaces sont associées directement aux erreurs humaines, que ce soit au niveau de la conception d'un système d'information ou au niveau de la manière dont on l'utilise. Ainsi elles peuvent être le résultat d'une erreur de conception ou de configuration comme d'un manque de sensibilisation des utilisateurs face au risque lié à l'usage d'un système informatique.

5. Vulnérabilités :

5.1. Définition :

Une vulnérabilité est une faiblesse ou faille d'une ressource informatique qui peut être exploitées par des menaces, dans le but de compromettre cette ressource. Les vulnérabilités sont beaucoup trop nombreuses pour être énumérées exhaustivement, toutefois selon différents standards et écoles, il est possible de les regrouper en trois familles, organisationnelle, physique et technologique (***Figure 1.3***) **[34].**

5.2. Les vulnérabilités au niveau organisationnel (Management) :

L'absence d'une gestion correcte d'un système informatique peut rapidement conduire à sa compromission (ressources jugées critiques internes à l'organisation) **[34].**

5.3. Les vulnérabilités au niveau physique :

Cette famille comprend toutes les vulnérabilités liées aux évènements imprévisibles comme les pannes, les accidents ou encore les atteintes intentionnelles aux matériels **[34].**

5.4. Les vulnérabilités au niveau technologique :

Cette famille de vulnérabilités est de loin la plus mouvante, en effet, elle comprend toutes les vulnérabilités liées à l'utilisation de technologies ou solution (hardware, software) **[34].**

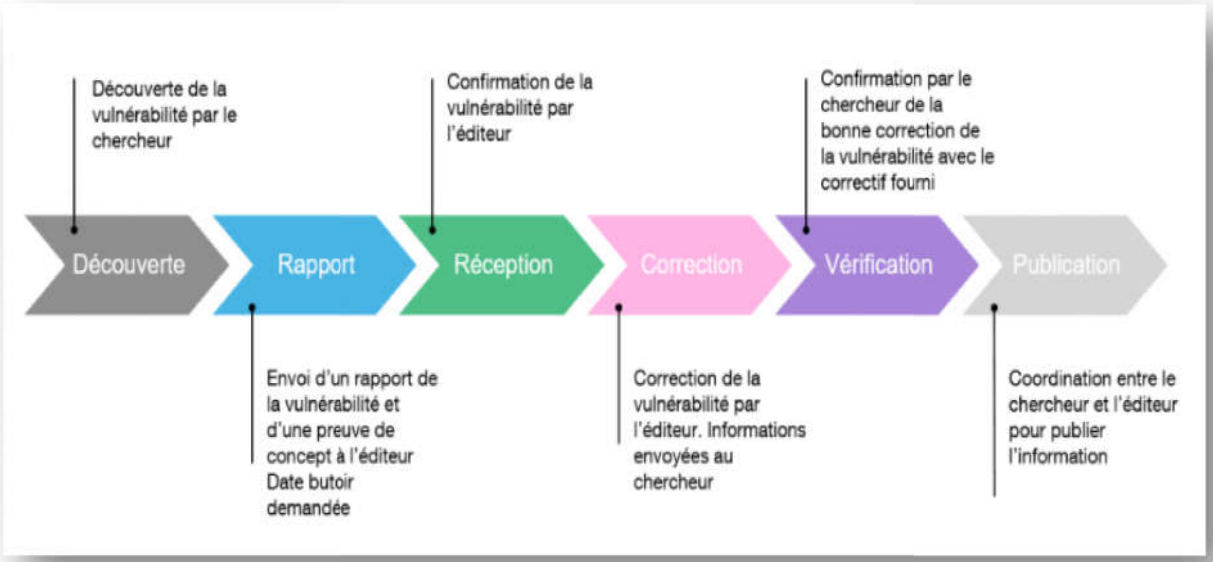

Figure 1.3. *Processus de veille d'une vulnérabilité* **[35].**

6. Attaques :

6.1. Définition :

« Une attaque informatique est tout acte sur un système informatique dont l'intention est de nuire au moins l'une des propriétés de sécurité» **[8].**

« Une attaque est l'exploitation d'une vulnérabilité d'un système informatique à des fins non connus par l'exploitant du système» **[9].**

6.2. Motivation des attaques :

Les motivations des attaques peuvent être de différentes sortes **[32]**:

- Obtenir un accès au système.
- Voler des informations, tels que des secrets industriels ou des propriétés intellectuelles.
- Glaner des informations personnelles sur un utilisateur Les menaces informatiques Motivation des attaques.

- Récupérer des données bancaires.
- S'informer sur l'organisation (entreprise de l'utilisateur, etc.).
- Troubler le bon fonctionnement d'un service.
- Utiliser le système de l'utilisateur comme « rebond » pour une attaque.
- Utiliser les ressources du système de l'utilisateur.

6.3. Classification des attaques :

Il existe plusieurs techniques de classification des attaques qui permettent de les regroupés dans des différentes catégories selon un critère particulier. Parmi ces critères, les plus récurrents sont **(*Figure 1.4*) [8]** :

- La cause de l'attaque (utilisateur interne ou externe, intrus, etc.).
- Le mode ou le type de l'attaque (virus, ver, écoute passive, déguisement, etc.).
- Le résultat de l'attaque (divulgation, perturbation, etc.).
- La vulnérabilité exploitée par l'attaque.

Il est ainsi possible de catégoriser les attaques de la manière suivante **[7]**:

6.3.1. Accès physique : il s'agit d'un cas où l'attaquant a accès aux locaux, éventuellement même aux machines :

- Coupure de l'électricité.
- Extinction manuelle de l'ordinateur.
- Vandalisme.
- Ouverture du boîtier de l'ordinateur et vol de disque dur.
- Ecoute du trafic sur le réseau.
- Ajout d'éléments (clé USB, point d'accès Wifi…etc.).

6.3.2. Interception de communications :

- Vol de session,
- Usurpation d'identité,
- Détournement ou altération de messages.

6.3.3. Dénis de service : il s'agit d'attaques visant à perturber le bon fonctionnement d'un service. On distingue habituellement les types de déni de service suivant :

- Exploitation de faiblesses des protocoles TCP/IP.
- Exploitation de vulnérabilité des logiciels serveurs.

6.3.4. Intrusions :

- Balayage de ports.
- Elévation de privilèges : ce type d'attaque consiste à exploiter une vulnérabilité d'une application.
- Malveillant : (virus, vers, et chevaux de Troie).

6.3.5. Ingénierie sociale : dans la majeure partie des cas le maillon faible est l'utilisateur lui-même. En effet, c'est souvent lui qui, par méconnaissance ou par duperie, va ouvrir une brèche dans le système, en donnant des informations (mot de passe par exemple) au pirate informatique.

6.3.6. Trappes : il s'agit d'une porte dérobée dissimulée dans un logiciel, permettant un accès ultérieur à son concepteur.

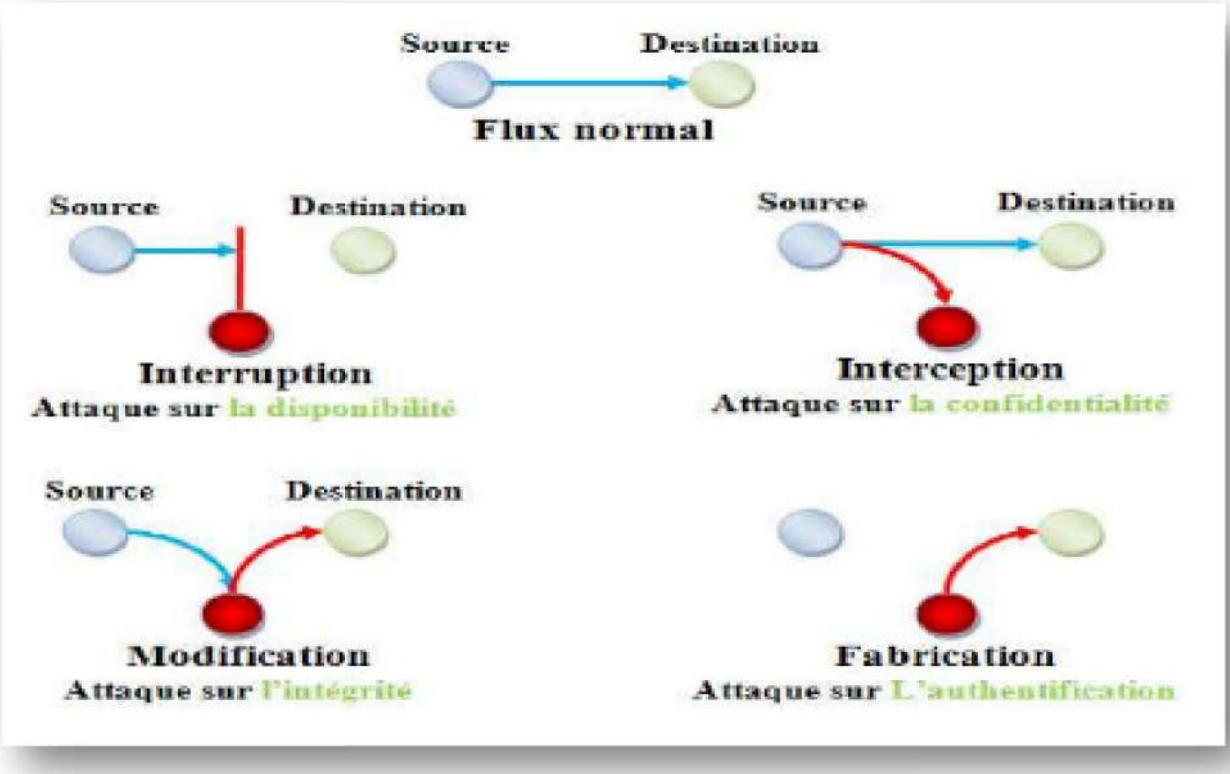

Figure 1.4. *Les différents principes de base d'une attaque informatique* **[36].**

7. Processus de sécurité :

La sécurité est un processus garanti au moyen de plusieurs outils, Mr. DONALD Pepkin a défini cinq mécanismes importants dans son livre « **sécurité des systèmes d'information** » pour élaborer un bon plan de sécurité **[10].**

7.1. Inspection :

C'est un mécanisme d'identification des fonctionnalités de base pour l'évaluation des besoins sécuritaires de l'entreprise, l'inspection est applicable en six étapes **[10]**:

- Inventaire des ressources.
- Estimation de la menace et l'analyse des pertes potentielles.
- Identification des vulnérabilités et l'organisation de la protection.
- Évaluation de l'état actuel.

7.2. Protection :

C'est un mécanisme assuré via un ensemble de moyens au but de réduire dynamiquement les risques, parmi les moyens utilisé **[7]**:

- Le logiciel antivirus.
- Le contrôle d'accès (Authentification).
- Les dispositifs de protections (Par-Feu, Zone démilitarisée, Serveurs mandataires, Réseaux privés virtuels).
- L'établissement de procédure de sécurité.
- Le chiffrement des données.
- Les mécanismes de sécurité physique.
- Les sauvegardes.
- Les mécanismes de redondance de l'information.

7.3. Détection :

Est un mécanisme de réaction contre les risques, la détection est basée généralement sur la prévention des attaques avant quelles seraient exploitées, ce mécanisme se compose de trois processus de base **[10]**:

- **Analyse de signature :** Ce processus permet la récupération des informations concernant l'ouverture d'une session afin de la comparer avec une base de traces des attaques pré connues pour le système ainsi permet le sauvegarde de nouvelles attaques dans la base (les attaques sont identifiés par leur signatures)
- **Analyse statique :** Ce processus permet la détection des failles sur les produits afin de définir le niveau de risque et prévenir les dégâts pouvant être causés par ces vulnérabilités.
- **Analyse dynamique :** Ce processus permet la détection des attaques au moment d'interception du comportement normale du produit en basant sur le résultat de l'analyse de signature.

7.4. Réaction :

Ce mécanisme consiste à définir un plan de secours en cas de détection d'une attaque, le plan de secours se roule généralement sur trois composants importants **[10]**:

- **Surveiller et avertir :** Ce composant consiste à envoyer des messages d'alertes à l'administrateur une fois un fait suspect est détecté alors que l'administrateur.
- **Réparer et signaler :** Certaines attaques s'agissent d'être remédier le plus vite possible, dans ce cas l'intervention du système de sécurité ne doit pas par la perte du temps d'avertir l'administrateur alors qu'il essaye de régler le problème tous seul puis envoyer un rapport à l'administrateur pour l'informer sur l'état de comportement avant, lors et après la correction.
- **Poursuivre en justice :** Dans le cas des dégâts importants et catastrophiques sur le produit, l'attaquant doit être poursuit en justice, c'est pour cela que le composant de poursuivre doit informer le service juridique de l'entreprise.

7.5. Réflexion :

Ce mécanisme est exécuté après la remise du système à son état normale, ce mécanisme permet l'étude de l'évènement afin de réaliser un plan d'intervention et de protection en basant sur le résultat des dégâts causés par l'attaque, le mécanisme de réflexion se roule en quatre étapes **[10]**:

- Documentation de l'incident.
- Évaluation de l'incident.
- Relations publiques.
- Suites judiciaires.

Remarque *: Il existe d'autres normes pour le processus de sécurité, par exemple la méthode* ***EBIOS*** *(Expression des Besoins et Identification des Objectifs de Sécurité****).***

8. Conclusion :

Ce chapitre constitue une petite synthèse de l'état de l'art on se qui concerne la sécurité informatique. Cette dernière est la préoccupation de tous les organisations modernes.

Dans ce chapitre, nous avons fait un survol sur des notions générales de la sécurité informatique : La terminologie de base, différentes menaces informatiques, la vulnérabilité et les attaques. Enfin le chapitre est terminé par la description d'un processus de protection. Toutes ces notions et principes de sécurité servent comme référence pour le développent des systèmes de sécurité.

Chapitre 02
Sécurité Dans le Cloud Computing

1. Introduction :

Le ***Cloud Computing*** est un domaine qui regroupe les technologies de distribution de service informatique logiciel et matériel via internet et à la demande des utilisateurs. Cette technologie permet aux entreprises d'acheter des ressources informatiques telles que les réseaux, les serveurs, l'espace de stockage, les applications et même des services sous la forme de service au lieu d'avoir à construire et entretenir des infrastructures informatiques en interne.

Dans ce chapitre, nous allons présenter la définition du ***Cloud Computing***, suivie par ses caractéristiques. Ensuite nous décrivons ses modèles de services et de déploiement sans l'oubli de ses chalenges. Nous terminons le chapitre par la présentation de la sécurité dans le ***Cloud Computing*** et une conclusion.

2. Définition de Cloud Computing :

Il existe de nombreuses définitions du terme ***Cloud Computing*** (informatique en nuage) qui reflètent sa diversité et sa richesse technologique. Dans ce qui suit, nous citons quelques une des plus pertinentes :

« *Le Cloud Computing est un modèle qui offre aux utilisateurs du réseau un accès à la demande, à un ensemble de ressources informatiques partagées et configurables, et qui peuvent être rapidement mises à la disposition du client sans l'interaction direct avec le prestataire de service »* **[13]**.

« *Le Cloud Computing, souvent appelé simplement « nuage », est une fourniture de ressources informatiques à la demande, des applications aux centres de données, en passant par l'internet et le paiement à l'utilisation*» **[15].**

« *Le Cloud Computing est un modèle pour permettre l'accès réseau ubiquitaire, facile et à la demande à un ensemble partagé de ressources informatiques (réseaux, serveurs, stockage, applications et services) configurables, qui peuvent être rapidement provisionnées et libérées avec un minimum d'efforts de gestion ou d'interaction avec le fournisseur du service*» **[17].**

La figure suivante donne une vue globale sur le concept du Cloud Computing

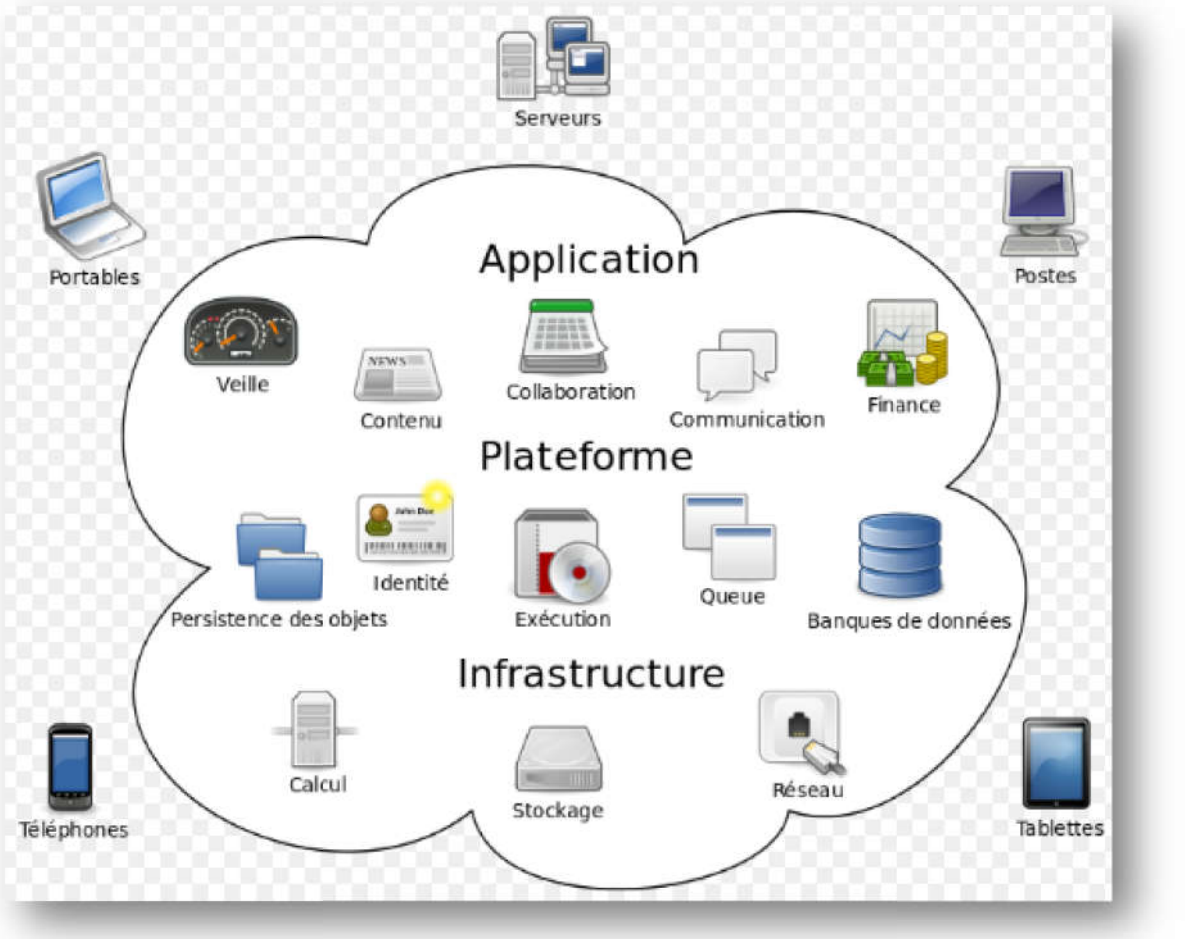

Figure 2.1. *Cloud Computing (Informatique en nuage)* **[14].**

En analysant ces définitions, on constate que le Cloud Computing tourne au tour de trois principaux acteurs qui sont :

- Les fournisseurs des services Cloud (**CSP** : Cloud Service Provider).
- Les utilisateurs des services Cloud (**CSU** : Cloud Service User).
- Les offres des services Cloud.

L'organisme **NIST**[1] précise que le Cloud Computing est composé de cinq caractéristiques essentielles, trois modèles de services de base et quatre modèles de déploiement (***Figure 2.2***)**.**

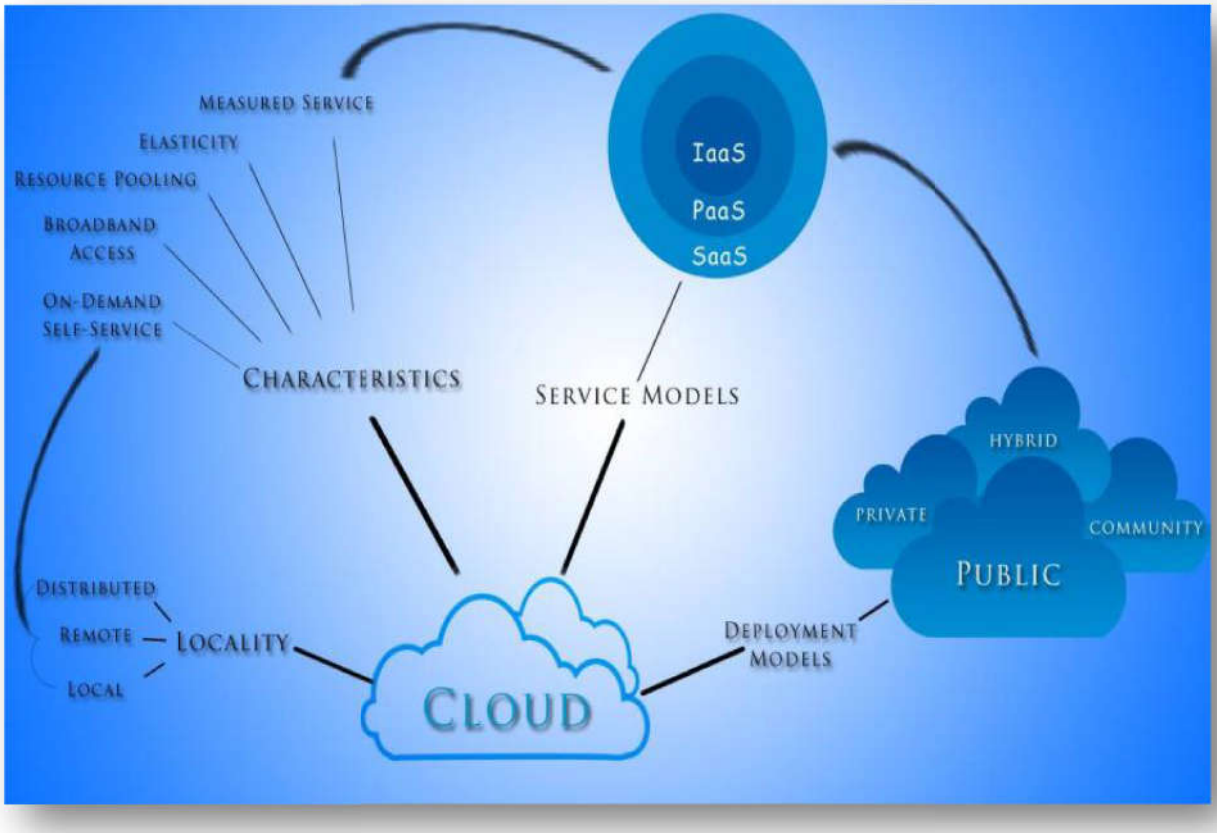

Figure 2.2. *Vue générale de l'environnement Cloud Computing* **[14]**.

3. Caractéristiques du Cloud Computing :

Les spécificités de la technologie *Cloud Computing* offrent à ses utilisateurs la possibilité d'accès à des logiciels et à des ressources informatiques avec la flexibilité et la modularité souhaitées et à des coûts très compétitifs. Selon l'organisme NIST, les caractéristiques essentielles sont **[13]**:

[1] **NIST** : Organisme fédéral américain gère de nombreux programmes conçus pour développer et favoriser les activités liées au mesurage, aux normes et aux technologies.

- **Libre-service à la demande** : Le client peut consommer les services Cloud automatiquement selon son besoin sans aucune nécessitée d'une interaction humaine avec le fournisseur.
- **Large accès réseau (Ubiquité)** : Les capacités sont disponibles sur le réseau et accessibles via des mécanismes standards qui favorisent l'utilisation de plates-formes.
- **Mise en commun des ressources (pooling)** : Les ressources et les services fournis au client sont souvent virtuels et partagés par plusieurs utilisateurs.
- **Elasticité rapide** : Les utilisateurs peuvent rapidement augmenter et diminuer leurs ressources en fonction des besoins, ainsi que de libérer les ressources pour d'autres utilisations quand ils ne sont plus nécessaires.
- **Les services sont fournis selon le modèle pay-per-use** : le Cloud utilise un modèle de paiement de type «payez ce que vous utilisez» ou en mode d'abonnement.

En plus des cinq caractéristiques définies par NIST, il y a d'autres caractéristiques dont nous citons les plus pertinentes **[14]**:

- **Autonome :** Les ressources du Cloud peuvent être automatiquement reconfigurés, orchestrés et consolidés en une seule image qu'elle sera fournie à l'utilisateur.
- **Service mesuré :** Toutes les ressources allouées peuvent être surveillées et contrôlées afin de mesurer leurs consommations avec un niveau d'abstraction approprié selon le type du service (ex stockage, temps de calcul, bande passante).
- **Fiabilité et tolérance aux pannes :** Les environnements Cloud Computing tirent parti de la redondance intégrée du grand nombre de serveurs qui les composent en permettant des niveaux élevés de disponibilité et de fiabilité (Par exemple, la probabilité de perte de donnée doit être quasi nulle).

- **Évolutivité :** Les services de type Cloud Computing devraient être évolutifs d'une façon automatique et en cours d'exécution pour satisfaire toute demande de croissance de la part des utilisateurs selon le besoin des services et des ressources allouées.
- **Simplicité d'utilisation :** L'allocation, la gestion et l'utilisation des ressources Cloud Computing doivent être simples. Idéalement, elles doivent se faire à travers des interfaces et des *Application Programming Interfaces*(APIs) efficaces et génériques.
- **Garantie de la qualité de service (QoS) :** Elle se caractérise essentiellement par la performance des ressources (Bande passante, Taille de mémoire, …etc.) et le délai de réponse.
- **Basé sur un contrat SLA (SLA : S**ervice **L**evel **A**greement**) : SLA** est un contrat par lequel un prestataire informatique (**CSP**) s'engage à fournir un ensemble de services à un ou plusieurs clients (**CSU)** en assurant des garanties de QoS.

4. Modèles de services :

Un environnement informatique standard peut être composé de plusieurs couches qui partent du bas niveau (le matériel physique) vers le haut niveau (les applications à utiliser) **[14]**. Selon cette structure, on découpe les services du Cloud Computing en trois principaux modèles, qui sont :

4.1. Infrastructure AS A Service (IaaS) [14]:

Le modèle « ***Infrastructure en tant que service*** » correspond à des ressources infrastructures offertes à la demande. Ces ressources sont des ressources de calculs, de stockage ou de réseau et peuvent être soit virtuelles ou physiques. Le fournisseur a la gestion des couches Calcul, Stockage, Réseau et Virtualisation. L'utilisateur des ressources **IaaS** est responsable de la gestion de toutes les couches à partir et au-dessus du système d'exploitation. L'utilisateur n'a ni le contrôle, ni la gestion, ni la visibilité de l'infrastructure sous-jacente.

Exemples des fournisseurs IaaS :

Amazon EC2, VPC, IBM Blue Cloud, Eucalyptus, FlexiScale, Joyent, Rackspace Cloud, ...etc.

4.2. Platform As A Service (PaaS):

Le modèle « ***Plate forme en tant que service*** » est un type de Cloud Computing où le fournisseur de services met à la disposition de ses utilisateurs des plate-formes d'exécution, de déploiement et de développement d'applications qui sont prêtes à l'emploi et fonctionnelles. Pour cela, ce modèle fournit un niveau d'abstraction supplémentaire par rapport à **IaaS** : en plus de l'infrastructure matérielle, l'hébergement et le Framework d'application sont dématérialisés **[16]**.

Avec **PaaS**, les développeurs peuvent souvent créer des applications sans avoir installer des outils sur leurs ordinateurs, et ils peuvent ensuite déployer ces applications sans aucunes connaissances spécialisées du système d'administration **[13]**.

Exemples des fournisseurs PaaS :

Windows Azure de Microsoft, App Engine de Google, Force.com de Salesforce.
Chaque fournisseur de PaaS propose des environnements de développement différents, Google App Engine se limite à Java et Python, tandis que Windows Azure permet de travailler avec les langages .NET, PHP, Python, Ruby et Java.

4.3. Software As A Service (SaaS):

Le modèle de services **« logiciel en tant que service »** permet de fournir des applications à la demande sur Internet. Ces services du Cloud sont fournis à des millions d'utilisateurs et sont accessibles à partir de différents dispositifs clients. Ceci permet de réduire un certain coût relatif aux logiciels et aux serveurs **[12]**.

Exemples des fournisseurs SaaS :

Serveurs web, serveurs de messagerie (Yahoo Email, Gmail, …), Google Apps, Facebook, Twitter, MobileMe, etc.

La figure suivante représente les couches manipulées par les utilisateurs et les couches manipulées par les fournisseurs du Cloud dans chaque modèle :

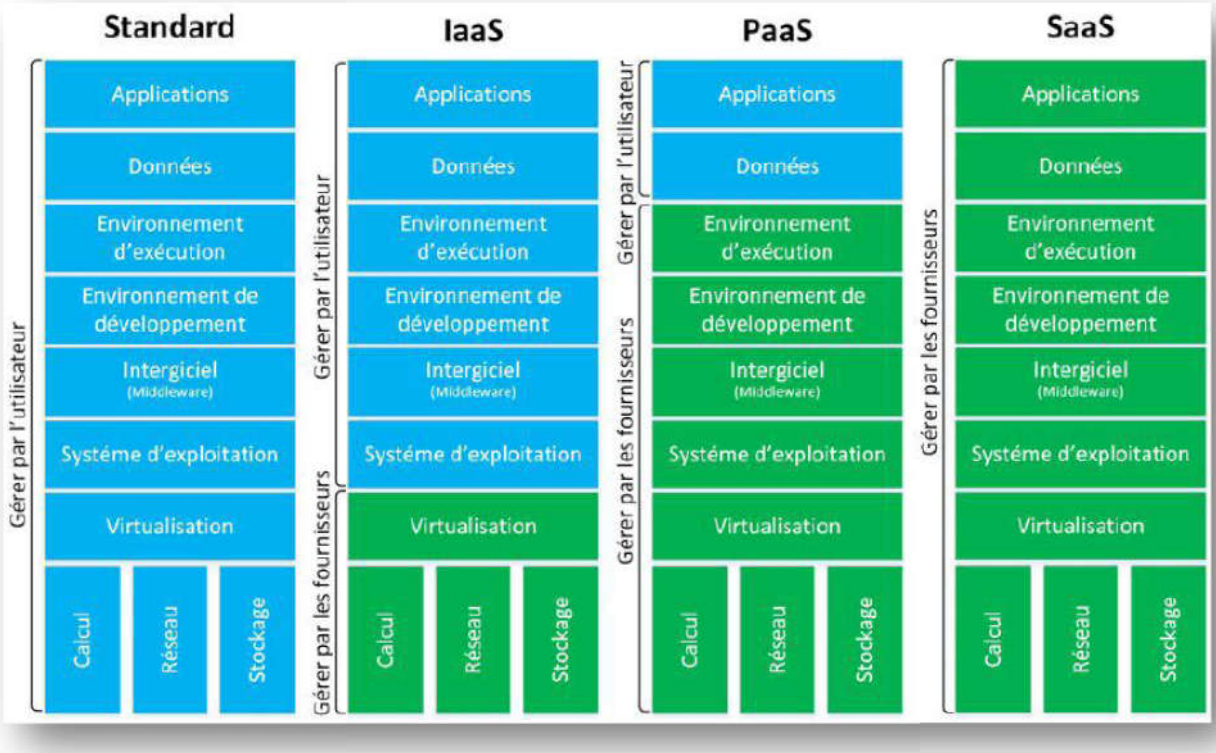

Figure 2.3. *Service du Cloud Computing* **[14].**

4.4. Anything ou Everything AS A Service (XaaS):

Le modèle de services « **Tout en tant que service** » désigne les services et les applications auxquels les utilisateurs peuvent accéder sur Internet sur demande. D'autre part, la désignation en tant que service est associée à divers composants numériques, par exemple données, sécurité, communication, …etc. Donc dans ce modèle de Cloud Computing, chaque infrastructure, fonction ou systèmes seront fournis comme un service **[37]**.

Bien que les trois types de services Cloud Computing (**IaaS**, **PaaS** et **SaaS**) soient la base de distinction du type de service, chacune des nouvelles abréviations (nouveaux types de service) peut-être vues comme un sous-ensemble d'un ou plusieurs des trois types de base. Parmi ces abréviations, nous citons les exemples suivants **[37]**:

- **Sécurité en tant que service (SECaaS) :** Il s'agit du modèle de gestion de la sécurité externalisée. Un fournisseur intègre ses services de sécurité dans l'infrastructure de votre entreprise et, en règle générale, les fournit sur Internet.
  Ces services peuvent inclure des logiciels antivirus, le chiffrement, l'authentification, des solutions de détection d'intrusions et plus encore.
- **Transport en tant que service (TaaS)** : Les tendances importantes de la société moderne sont la mobilité et la liberté de transport à différentes distances. Il existe de nombreuses applications connectées au transport, donc une partie de cette industrie se transforme en un modèle du Cloud.
- **Communication en tant que service (CaaS) :** Ce modèle comprend différentes solutions de communication telles que la VoIP (voix sur IP ou téléphonie Internet), la messagerie instantanée (messagerie instantanée), les applications de vidéoconférence hébergées dans le Cloud du fournisseur. Une entreprise peut déployer de manière sélective des applications de communication qui répondent le mieux à ses besoins actuels pendant une certaine période et payer uniquement pour cette période d'utilisation.

5. Modèle de déploiement :

Le Cloud Computing repose sur des ressources physiques qui peuvent être situées chez le client ou chez le fournisseur, être partagées ou non. Ainsi, les utilisateurs du Cloud peuvent choisir entre se construire leurs propres infrastructures ou en louer une chez un fournisseur de service spécialisé, bénéficier de services plus ou moins étendus proposés par ces fournisseurs ou encore combiner ces deux options. Selon les approches des entreprises, on distingue les typologies suivantes de Cloud Computing :

5.1. Le Cloud publique [13]:

Un Cloud public est un service IaaS, PaaS ou SaaS proposé et hébergé par un tiers d'un ou plusieurs centres de données (***Figure 2.4***) : Amazon, Google et Microsoft propose un Cloud public dans lequel n'importe quel particulier ou n'importe quelle entreprise peut y héberger ses applications, ses services ou ses données. Pour les

consommateurs, il n'y a donc aucun investissement initial fixe et aucune limite de capacité.

Les fournisseurs de Cloud public sont les responsables pour la gestion de la sécurité et facturent à l'utilisation et garantissent une disponibilité de services au travers des contrats SLA.

5.2. Le Cloud privé :

L'ensemble des ressources d'un Cloud privé est exclusivement mis à disposition d'une entreprise ou organisation unique. Le Cloud privé peut être géré par l'entreprise elle-même (Cloud privé interne) ou par une tierce partie (Cloud privé externe). Les ressources d'un Cloud privé se trouvent généralement dans les locaux de l'entreprise ou bien chez un fournisseur de services. Dans ce dernier cas, l'infrastructure est entièrement dédiée à l'entreprise et y est accessible via un réseau sécurisé (de type VPN). L'utilisation d'un Cloud privé permet de garantir, par exemple, que les ressources matérielles allouées ne seront jamais partagées par deux clients différents **[14]**.

5.3. Le Cloud communauté :

Le modèle Cloud communautaire ou Cloud collectif est pour partager l'infrastructure par plusieurs organisations indépendantes ayant des intérêts communs. Cette communauté d'organisation peut partager aussi les tâches de gestion de ces infrastructures, comme la sécurisation des données, le déploiement d'applications, l'authentification, etc. L'avantage des Clouds communautaires est qu'ils permettent à plusieurs entités indépendantes d'obtenir les avantages financiers d'un Cloud non public partagé tout en évitant les problèmes de sécurité et de réglementation qui peuvent être associés à l'utilisation d'un Cloud public générique qui ne répondait pas à ces préoccupations dans son contrat SLA. Pour cela, différents types de Cloud communauté sont envisagés spécialement aux États-Unis et aux l'Union européenne sur les gouvernements aux niveaux national ou local **[15]**.

5.4. Le Cloud hybride :

Un Cloud hybride est une combinaison de déploiement des modèles de Cloud (public, privé et communauté) qui tente de remédier aux limitations de chaque approche. Dans un Cloud hybride, une partie du service de l'infrastructure s'exécute dans des Clouds privés tandis que la partie restante est dans des Clouds publics. Un Cloud hybride offre plus de flexibilité qu'un Cloud public ou privé, puisqu'il fournit un meilleur contrôle et une meilleure sécurité pour les données d'application des utilisateurs par rapport aux Clouds publics et une tarification avantageuse par rapport aux Clouds privés. Cependant, la conception d'un Cloud hybride nécessite une étude détaillée afin de déterminer la meilleure répartition entre les composantes de Cloud public et privé **[12]**.

Pour la plupart des organisations, le choix du modèle de Cloud dépend du cas d'utilisation et des exigences du CSU (sécurité, QoS, …etc.).

6. Les défis du Cloud Computing :

Les utilisateurs du Cloud Computing reconnaissent bien volontiers ses avantages. Mais, l'adoption du Cloud représente un véritable bouleversement et une remise en question pour les entreprises car elles recherchent de vraies garanties autour d'un certain nombre de points, en particulier, la contractualisation rigoureuse et la qualité et la sécurité des services. Dans cette section, nous présentons quelques enjeux de recherche dans le Cloud Computing qui sont **[25]**:

- Qualité de service notamment la performance et la disponibilité.
- Sécurité de données et confidentialité.
- Migration des données entre les environnements non standards.
- Migration de machines virtuelles.
- Consolidation de serveurs.
- Gestion de l'énergie.
- Précision et uniformalisation des contrats.

7. La sécurité dans le Cloud Computing :

La sécurité du Cloud Computing est classée comme son plus grand défis ou problème. Elle est définie comme étant l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires pour protéger les données, les applications et l'infrastructure associée au Cloud contre une faiblesse d'ordre logicielle ou matérielle qui peut être exploitée par une ou plusieurs menaces internes ou externes **[12]**.

La sécurité du Cloud implique un ensemble des concepts tels que :

7.1. Sécurité selon des normes :

Il est important que les exigences de sécurité soient cohérentes avec les standards appropriés, comme l'ISO 27001, l'ISO 27002 et récemment ISO/IEC 27701 (Aout 2019) pour bénéficier de nombreuses et meilleures expériences pratiques **[14]**.

7.2. Sécurité approuvé par SLA :

La sécurité dans le Cloud peut être définie dans un accord de niveau de service SLA afin de garantir un niveau de sécurité pour chaque service. Cet accord de niveau de service peut être négocié et établi entre le CSP et le CSU ou entre les CSP. Il définit la sécurité mise en place contre les attaques malveillantes et les pannes éventuelles entrainant des problèmes de sécurité. De plus, la spécification d'un niveau de sécurité mesurable dans le SLA est utile pour améliorer la transparence et la confiance dans la relation entre le CSU et le CSP **[12]**.

7.3. Sécurité de base :

Le Web 2.0, une technologie clé permettant l'utilisation de logiciel en tant que service (ex : SaaS). Cette technologie soulage les utilisateurs dans les tâches de la maintenance et l'installation de logiciels. Alors sa sécurité est devenue plus importante que jamais pour un tel environnement. Par conséquent, la sécurité à différents niveaux est nécessaire pour garantir la bonne mise en œuvre du Cloud Computing, telles que: la sécurité d'accès de serveur, la sécurité d'accès à Internet, la sécurité d'accès aux bases de données, la sécurité des données et la sécurité des programmes. En outre, il

doit assurer la sécurité des données au niveau de la couche réseau et la sécurité des données au niveau de la couche physique et de la couche application pour maintenir un Cloud sécurisé **[15]**.

7.4. Sécurité physique [15]:

Les locaux ou le bâtiment dans lequel les Datacenters du Cloud sont hébergés, font aussi l'objet de différentes menaces, y compris les actions humaines, l'accès privilégiés à tous les utilisateurs, les zones désastres et les dangers naturels. Pour cela la sécurité physique d'un local doit être vue comme un système de protection pour mettre en place une défense multi-niveaux ou multicouches dont chaque couche est associée à un contrôle général automatisé.

L'organisation d'une sécurité physique efficace nécessite une conception environnementale qui prend en compte les activités normales et les situations imprévues. Les éléments de la sécurité physique doivent être soutenus par des procédures appropriées, par exemple chaque centre de données ou zone sensibles (salles de serveurs, armoires réseau, armoires utilitaires) requiert au moins une authentification basée sur un lecteur de badge. Les tentatives d'accès infructueuses répétées déclenchent une alerte pour les gardes de sécurité. Les essais d'accès sont enregistrés, et les journaux de vidéosurveillance et les journaux de l'historique d'accès ont aussi conservés pendant des périodes déterminés. L'étendue des difficultés de sécurité physique est vaste et implique de nombreuses mesures pour empêcher, découvrir et répondre aux accès non autorisés aux locaux, aux ressources ou aux informations présentes dans les locaux.

7.5. Sécurité au niveau du réseau [15]:

Pour assurer la sécurité du réseau, il convient de prendre en compte les points suivants: la confidentialité et l'intégrité dans le réseau, le contrôle d'accès et le maintien de la sécurité appropriés contre les menaces de tiers. Les attaques au niveau du réseau comprennent : l'attaque de DNS, attaques par Sniffer, problème ou la réutilisation d'adresse IP, l'attaque par déni de service (DoS) et l'attaque par déni de service distribué (DDoS), …etc.

7.6. Sécurité au niveau de l'application [15]:

La sécurité au niveau de l'application fait référence à l'utilisation des ressources logicielles et matérielles pour assurer la sécurité des applications de telle sorte que les attaquants ne puissent avoir aucun contrôle ou modification sur les applications.

Avec les récents progrès technologiques, différentes méthodes et techniques a été réalisés pour faire face aux problèmes de sécurité d'application. L'utilisation du dispositif **ASIC**[2] orienté tâche, il est capable de gérer une tâche spécifique offrant des niveaux de sécurité plus élevés avec des performances plus élevées.

Les menaces sur la sécurité des applications incluent les attaques XSS, l'empoisonnement des cookies, la manipulation de champs cachés, les attaques par injection SQL, les attaques par déni de service, les options de porte dérobée et de débogage, etc., résultants à l'utilisation non autorisée des applications.

7.7. Sécurité au niveau des données:

La sécurité des données se réfère principalement à la confidentialité, l'intégrité et la disponibilité qui sont le principal problème pour les fournisseurs. Dans un modèle de déploiement d'applications sur site traditionnel, les données sensibles de chaque entreprise continuent de résider dans les limites de l'entreprise et sont soumises à leur sécurité physique, logique et personnelle et ses politiques de contrôle d'accès **[5]**.

Cependant, dans le Cloud Computing, les données d'entreprise sont stockées en dehors des limites de l'entreprise. Par conséquent, le fournisseur de Cloud doit adopter des contrôles de sécurité supplémentaires pour garantir la sécurité des données et éviter les violations **[5]**. Les éléments clés suivants doivent être soigneusement pris en compte afin d'assurer la sécurité des données pour l'entreprise:

7.7.1. Cycle de vie des données [15]:

- **Création :** La création est la génération d'un nouvel élément ou la modification d'un élément de données numériques existe. On peut donc l'appeler également en

[2] l'**ASIC : A**pplication **S**pecific **I**ntegrated **C**ircuit en anglais, est un circuit intégré exclusivement dédié à une application et à un utilisateur

tant que phase de création / mise à jour. Il peut s'agir de tout type de contenu, pas seulement d'un document ou d'une base de données c'est-à-dire peut être structuré ou non structuré. Dans cette phase, les données sont classées et les droits appropriés sont déterminés.

- **Stockage :** Le stockage est l'action pour former les données numériques selon un type référentiel de stockage structuré ou non structuré (base de données ou fichier). Cette opération se produit généralement en même temps que la création. Ici, la classification et les droits des contrôles de sécurité doivent être mappés, y compris les contrôles d'accès, le chiffrement et la gestion des droits.
- **Utilisation :** Les données sont visualisées, traitées ou utilisées dans une manière où ces données originales ne sont pas modifiées. Ces activités s'appliquent généralement aux données stockées au moment de l'utilisation d'après d'un PC ou d'une application de l'utilisateur. Pour assurer ce type d'activité, il existe des contrôles de détection tels que la surveillance d'activité, des contrôles préventifs tels que la gestion des droits et des contrôles logiques qui sont généralement appliqués dans les bases de données et les applications.
- **Partage :** Les données sont rendues accessibles aux autres, et elles sont échangées entre les utilisateurs, les clients et les partenaires. Les contrôles de cette phase incluent une combinaison des opérations de détection et de prévention, de cryptage pour un échange sécurisé des données, des contrôles logiques ainsi que la sécurité des applications.
- **Archivage :** Les données restent sans utilisation et entrent dans la mémoire à long terme doivent être archivées, ici, la protection des données et leur disponibilité sont assurées par une combinaison de gestion de cryptage et de gestion des bénéfices.
- **Suppression :** Les données sont détruites de manière permanente à l'aide des moyens physiques ou logiques. Les données doivent être supprimées de manière sécurisée et doivent être utilisées des outils pour retrouver les copies permanentes.

7.7.2. Localisation et accès :

Une sécurité élevée de données peut être obtenue en identifiant ces mouvements et en appliquant les contrôles appropriés aux limites de sécurité appropriées. Ces environnements peuvent être des Clouds internes, externes, publics ou privés, des fournisseurs de Cloud ou des sous-traitants traditionnels, etc. Pour cela, il est très important de comprendre les emplacements logiques et physiques des données **[15]**.

7.7.3. Mesures de protection :

A chaque étape de cycle de vie des données, différentes mesures peuvent être mises en œuvre pour assurer la sécurité des données. Les mesures de protection sont de deux types **[11]**:

- **Le 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. L'ensemble des techniques, systèmes et moyens de contrôle d'accès sont regroupés sous l'acronyme IAM (Identity and Access Management).
- **Le chiffrement :** Grâce à un logiciel spécifique, un individu peut " crypter " ses propres données. Ainsi, leur accès est limité dans la mesure où il faut avoir la clé de déchiffrement pour pouvoir les lire. Par conséquent, la personne qui détient la clé est la seule à pouvoir avoir accès aux données.

8. Conclusion :

Nous avons donné dans ce chapitre une vue détaillée sur l'approche du ***Cloud Computing*** dont ses offres ne cessent pas de s'augmenter dans le but de satisfaire les besoins des utilisateurs et en particulier les entreprises. Grâce aux avantages offerts par le Cloud, les utilisateurs finaux trouvent que cette technologie est un bon choix pour l'utilisation des services. Malgré tous les bénéfices offerts par le Cloud, il existe toujours des limites et en particulier la sécurité des données. Cette insuffisance inspire l'intérêt de notre recherche. La partie suivante de notre travail, consiste à proposer une solution sur la sécurité des données dans un Cloud privé en se basant sur une technique du contrôle d'accès.

Chapitre 03
Contrôle d'accès aux systèmes d'information

1. Introduction :

Généralement, dans un système d'information on considère souvent la confidentialité, l'intégrité et la disponibilité des données comme les trois propriétés fondamentales à respecter pour assurer la sécurité (***Figure3.1***). De nombreuses pratiques et dispositifs participent à garantir ces propriétés de sécurité: la protection des réseaux, le cryptage de l'information, la sauvegarde des données, les architectures d'authentification. Parmi les moyens mis en œuvre pour renforcer la sécurité, nous allons nous intéresser aux politiques de contrôle d'accès.

Ce chapitre commence par définir la politique de contrôle d'accès et ces types (statique et dynamique) et aussi par montrer son importance dans la sécurité des systèmes d'information. Ensuite, nous montrons les étapes du mécanisme de moniteur de référence mis en œuvre pour réaliser le contrôle d'accès. Puis nous expliquons les différents modèles de contrôle d'accès largement utilisés dans le monde industriel (les modèles classiques et les modèles à base de rôle) en présentant ces principes de fonctionnement et ces points faibles.

2. Le contrôle d'accès :

Le contrôle d'accès, dans un système d'information, est l'ensemble des mesures en place pour restreindre l'accès aux ressources du système suivant des contraintes préétablies **[23]**. Le contrôle d'accès est aussi un mécanisme grâce auquel un système autorise ou interdit les actions demandées par des sujets (entités actives) sur des objets (entités passives). Il renforce particulièrement la confidentialité et l'intégrité de l'information, a fortiori sa disponibilité **[22]**.

Il existe de nombreux modèles de contrôle d'accès. Une instance d'un tel modèle représente **une politique de contrôle d'accès**. Cette dernière définit donc les accès aux ressources d'un système **[23].**

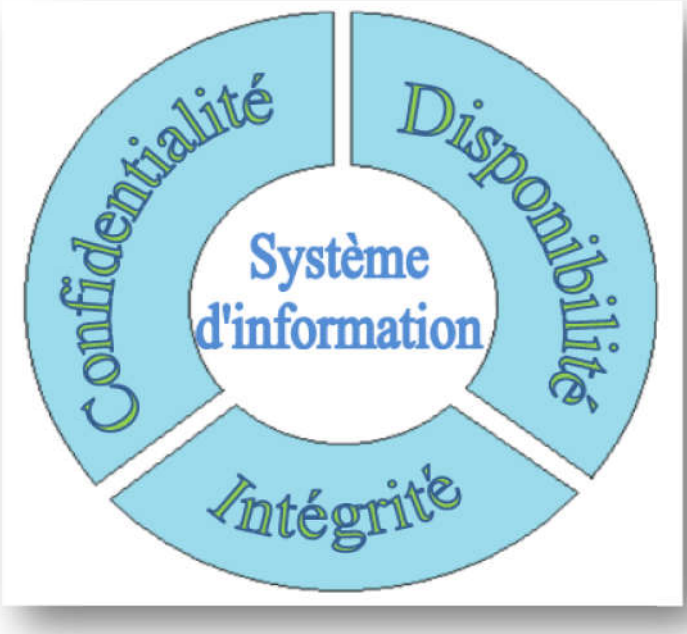

Figure3.1. *Les trois propriétés fondamentales de la sécurité*
« Image illustrée de **[26]** *».*

Un modèle de contrôle d'accès comprend **[27]**:

- **Une politique de contrôle d'accès** qui spécifie les accès autorisés aux données.
- **Une politique d'administration** qui indique comment la politique de contrôle d'accès peut être mise à jour.

2.1. Politiques de contrôle d'accès :

Les contraintes qui régissent les accès aux ressources d'un système peuvent être de nature statique ou dynamique. On distingue ainsi deux types de politiques **[23]**:

- Les politiques de contrôle d'accès statique.
- Les politiques de contrôle d'accès dynamique.

2.1.1. Politique de contrôle d'accès statique [23]:

Pour un système d'information donné, une politique de contrôle d'accès statique est caractérisée par le fait que son état ne change pas par rapport à l'évolution dynamique du système, car elle comporte seulement des contraintes statiques. Celle-ci peut être mise à jour pour refléter divers changements dans l'organisation (par exemple, un changement d'affectation dans une organisation qui entraîne une

augmentation des privilèges d'un utilisateur). L'initiation d'une action par un utilisateur déclenche l'évaluation de l'état de la politique. En fonction des autorisations accordées par la politique dans son état courant, l'exécution de l'action est permise ou pas. Dans la plupart des implémentations, une politique de contrôle d'accès statique associe les utilisateurs du système à leurs privilèges.

2.1.2. Politique de contrôle d'accès dynamique [23]:

Pour un système d'information donné, une politique de contrôle d'accès dynamique possède plusieurs états, car elle est associée à l'évolution du système. L'autorisation de l'exécution d'une action est basée sur l'évaluation de l'état courant du système et sur la définition même de la politique. L'état courant est mis à jour lors de chaque exécution d'une action contrôlée. La version élémentaire de ce type de politiques de contrôle d'accès utilise un historique des actions exécutées par le système d'information qui est mise à jour par le gestionnaire de mise en œuvre de la politique. Les langages formels qui supportent les traces d'événements, comme les langages basés sur une algèbre de processus, se prêtent bien à l'expression de contraintes dynamiques.

2.2. Moniteur de référence :

L'application du contrôle d'accès également appelée **enforcement** est assurée par un module appelé **moniteur de référence (**un moniteur, intermédiaire entre les utilisateurs et les ressources auxquelles ces derniers essaient d'accéder) qui intercepte tous les accès aux données et détermine s'ils sont légitimes ou non. L'ensemble des droits et conditions appliqués par le moniteur de référence constitue la politique de contrôle d'accès qui peut être représentée et exprimée de multiples façons **[28]**.

La définition d'un tel moniteur est délicate, car ce dernier doit être *incontournable*, *inviolable* et *vérifié*. Lorsqu'un utilisateur demande un accès, le moniteur va décider si cet accès est autorisé ou non d'après la politique de contrôle d'accès : une instance spécialisée de la politique de sécurité logique, qui s'attache à définir les droits des utilisateurs des systèmes **[22]**.

Le principe de fonctionnement d'un moniteur est décomposable en 6 étapes, comme l'illustre **la figure 3.2 [22]** :

1. Envoi de la requête de l'utilisateur au moniteur.
2. Interrogation de la politique de contrôle d'accès.
3. Réponse de la politique.
4. Le moteur accède à la ressource si l'accès est autorisé pour exécuter la requête.
5. Retour de l'exécution de la requête.
6. Retour de la requête ou exception en cas d'accès non autorisé.

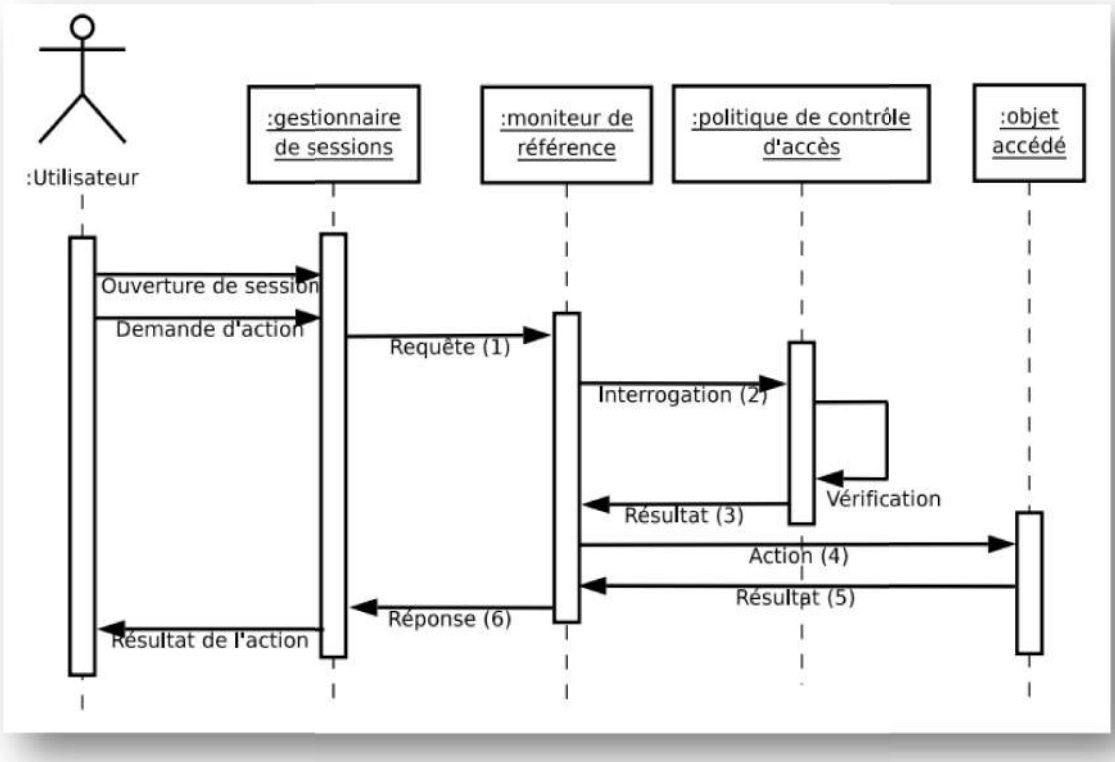

Figure 3.2. *Mécanisme de moniteur mis en œuvre pour réaliser le contrôle d'accès* **[22].**

2.3. Formalisation du contrôle d'accès [23]:

Le contrôle d'accès dans un système d'information permet de contraindre l'accès à ses ressources. Une façon d'interpréter une politique est de considérer un système d'information composé d'un ensemble d'états Q et d'un ensemble de transitions entre ces états, où chaque transition représente un accès à ses ressources. Ainsi une politique

de contrôle d'accès partitionne l'ensemble des états *Q* en un ensemble d'états autorisés *Qaut* ⊆ *Q* et un ensemble d'états interdits. La mise en œuvre des politiques de contrôle d'accès est réalisé par un *mécanisme de sécurité* qui empêche le système de se retrouver dans un état interdit. Désignons *Qacc* l'ensemble des états accessibles par le mécanisme de sécurité. Le mécanisme est dit *sûr* lorsque *Qacc* ⊆ *Qaut*. Dans ce cas, les états accessibles par le mécanisme de sécurité sont tous des états autorisés par la politique de contrôle d'accès. Lorsque *Qacc* = *Qaut*, le mécanisme de sécurité est dit *précis* et tous les états autorisés par la politique sont accessibles par le mécanisme de sécurité et vice-versa. Dans le cas général où *Qacc* ∩ *Qaut* ≠ ∅, le mécanisme de sécurité est dit *large*, car le système peut se retrouver dans un état interdit. Les modèles présentés dans la suite sont des moyens qui précisent comment les di!érents accès au système sont réalisés de telle sorte que son état courant soit toujours dans l'ensemble *Qaut*.

3. Les modèles de contrôle d'accès :

Il existe différents modèles de contrôle d'accès, chacun étant est adapté à des besoins différents. Nous détaillons ici quelques modèles de contrôle d'accès largement étudiés dans la sécurité informatique et beaucoup utilisés dans le monde industriel. Aussi nous expliquant les différents modèles de contrôle d'accès existants en indiquant leur domaine d'application. Ces modèles sont répartis dans différentes catégories **[27]**:

1. Les modèles de contrôle d'accès classiques :
   - Le modèle discrétionnaire (**DAC :** Discretionary Access Control).
   - Le modèle obligatoire (**MAC :** Mandatory Access Control).
2. Les modèles de contrôle à base de taches (**TBAC :** Task Based Access Control**).**
3. Les modèles de contrôle à base de rôles (**RBAC :** Role Based Access Control**)**.
4. Les modèles de contrôle à base d'organisation (**OR-BAC :**Oganization Role Based Access Control).
5. Les modèles de contrôle d'accès contextuels (**CBAC :** Contexte Based Access Control**)**.

Pour qu'on se met d'accord on considère les notions suivantes :

- **Un sujet (S)** : Un sujet peut être un processus, un utilisateur ou une application.
- **Un objet (O)** : Un objet est un conteneur d'informations sur lequel un sujet peut effectuer des actions (exemples : fichiers, sockets de communication, périphériques matériels.. etc.).
- **Une action (A)** : représente l'action à traiter par le sujet sur l'objet. (exemples : lecture, écriture, exécution d'un fichier, envoi de signaux ou de messages inter-processus ...etc.).

3.1. Contrôle d'accès discrétionnaire DAC:

Désigné par l'acronyme **DAC**, la politique de contrôle d'accès discrétionnaire est représentée sous la forme d'une série de triplets (sujet, action, objet). Chaque triplet (**S, A, O**) signifie « le sujet **S** a la permission d'effectuer l'action **A** sur l'objet **O** », ce modèle permet d'associer l'identité d'un sujet à un ensemble d'autorisations sur des objets. Lorsqu'un sujet fait une requête sur un objet, l'action est accordée si et seulement si une autorisation le permet **[27]**.

3.1.1 Principe de la politique DAC :

La spécification de la politique **DAC** se concrétise par l'aspect **discrétionnaire** qui s'interprète par l'attribution des droits d'accès de la façon suivante :

- Un utilisateur du système peut lui-même attribuer des droits à d'autres sujets sur les objets dont il possède des autorisations. En revanche il ne peut pas attribuer de droits qu'il ne possède pas (***Figure 3.3***) **[28]**.
- Le sujet propriétaire d'un objet a toute latitude pour décider quels autres sujets peuvent exercer des actions sur l'objet (lecture, écriture, exécution) **[18]**.

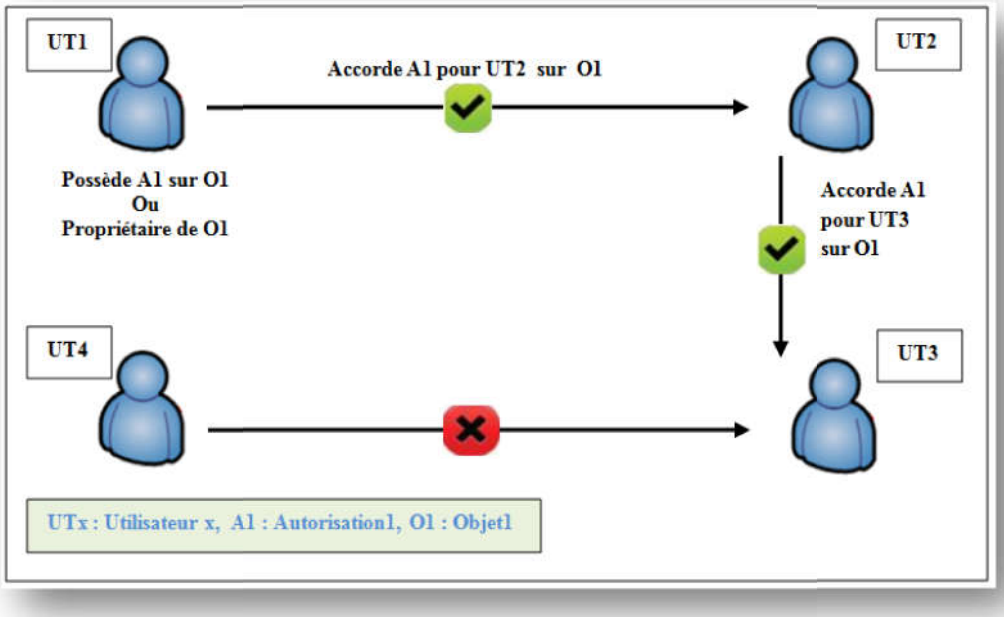

Figure 3.3. *Un exemple de modèle DAC* **[5].**

L'implantation de ce modèle a donné lieu à la constitution de matrices d'accès, l'état du système est défini par un triplé (*S*, *O*, *M*) où *S* représente l'ensemble des sujets (e.g. utilisateur, processus etc.) pouvant exercer un ensemble d'actions. *O* représente l'ensemble des objets (e.g. fichier, table, classe, programme etc.). Enfin, *M* représente la matrice d'accès, où les lignes correspondent aux sujets et les colonnes correspondent aux objets (***Tableau 3.1***) **[18]**.

Objets / Sujets	Fichier	Tables
Ahmed	Lire Ecrire	Ecrire Lire
Sara	Lire Ecrire Exécuter	Lire

Tableau 3.1. *Exemple d'une matrice d'accès* **[18].**

Les droits correspondent généralement à des actions élémentaires comme *lire, écrire, exécuter ou posséder* (mais ne sont pas limités à ces derniers). En effet, si de nouveaux objets, de nouveaux sujets ou de nouvelles actions sont ajoutés dans le système, il devient nécessaire d'enregistrer toutes les permissions accordées pour ces nouvelles entités **[18]**.

Il existe en pratique deux approches pour implémenter la matrice d'accès **[18]**:

- Par une liste de contrôle d'accès (ou **ACL** pour **Access Control List**) : la matrice est stockée par colonne. A chaque objet est associée une liste de règles indiquant pour chaque utilisateur les actions pouvant être exercées par ce dernier sur cet objet.
- Par une liste de capacité (ou capability) : la matrice est stockée par ligne. A chaque utilisateur correspond une liste, appelée liste de capacité, indiquant pour chaque objet les actions que l'utilisateur est en droit d'effectuer sur cet objet.

Le contrôle d'accès **DAC** a été principalement implanté au sein des systèmes d'exploitation (Microsoft Windows, Solaris, Linux, FreeBSD). Les utilisateurs peuvent ainsi transférer leurs droits à d'autres utilisateurs sur les données ou services qu'ils contrôlent sans avoir besoin d'une autorité centrale qui pilote le tout. Cet aspect discrétionnaire le rend flexible et adapté à de nombreux systèmes multi-utilisateurs **[28]**.

3.1.2. Points faibles de la politique DAC :

Diverses études ont montré la faiblesse des modèles **DAC**. En effet, le contrôle d'accès discrétionnaire repose sur la capacité des utilisateurs à définir correctement les permissions sur les fichiers dont ils sont propriétaires. Toute erreur peut mener à une défaillance de sécurité, Cependant une fois qu'un propriétaire accorde un accès pour un utilisateur sur un objet, il n'a plus aucun contrôle sur les futurs accès qui pourraient être créés. Les attaques possibles contre les systèmes d'exploitation visent à obtenir un accès de niveau **super-utilisateur** (par exemple, root). Lorsqu'une telle attaque est réussie, elle obtient des pouvoirs qui outrepassent le **DAC** et donnent un accès complet à l'ensemble des ressources du système d'information. De fait, la faiblesse de ce contrôle est que la politique de sécurité peut être à tout moment modifiée par le super-

utilisateur du système d'exploitation **[29]**. C'est exactement ce que les modèles de contrôle d'accès obligatoire **MAC** cherchent à éviter.

3.2. Modèles de contrôle d'accès obligatoires MAC :

Les modèles obligatoires de contrôle **MAC** ont été mise en œuvre afin d'apporter des solutions aux problèmes de sécurité d'information et pour rependre à la faiblesse des modèles **DAC.** La politique **MAC** repose sur la délégation du contrôle d'accès à une entité indépendante. L'existence de cette entité indépendante garantit que la politique de sécurité ne soit pas modifiable directement par les utilisateurs du système d'information **[29]**.

On distingue deux orientations dans les modèles de type **MAC** :

- Le premier modèle, appelé modèle de **Bell et La Pudula**, a été développé pour le département de la défense américain et vise, plus particulièrement, à assurer la confidentialité des données dans le contexte de l'utilisation partagée de mainframes **[18]**. Ces modèles répondent à des problématiques très précises, par exemple la nécessité de disposer d'une habilitation adéquate pour la lecture de documents classifiés dans **BLP (**Modèle de confidentialité de Bell et LaPadula**)**. Cependant, l'usage courant des systèmes d'exploitation moderne ne peut souvent pas être décrit par une seule de ces problématiques, mais relèvent plutôt de problématiques plus complexes comme le principe de moindre privilège, la confidentialité des données entre utilisateurs, la nécessité de prendre en compte les activités des utilisateurs sur le système **[29]**.
- Le deuxième modèle, appelé modèle de **Biba** qui prend en compte les exigences commerciales s'intéresse plutôt à l'intégrité des informations **[30]**. Ce modèle est appelé aussi le modèle **DTE** (Domain and Type Enforcement : Modèle de protection associant des domaines aux sujets et des types aux objets). Plutôt que de considérer une problématique particulière, celui-ci fournit un mécanisme générique, et autorise la spécification de politiques adaptées à tout environnement **[29]**.

Les modèles **MAC** ont ensuite été implantés dans des systèmes variés dont voici une liste non exhaustive **[27]** :

- Dans la carte à puce Java multi-applications, les contrôles d'accès obligatoires sont utilisés pour réguler les flux d'informations entre les différents applets Java.
- Le module noyau Security **Enhanced Linux** permet d'activer les contrôles d'accès obligatoires dans les systèmes d'exploitation Linux et Android (depuis la version 4.3).
- Windows Vista et Windows 7 implantent des contrôles d'accès basés sur le modèle de **Biba** afin, en particulier, d'éviter qu'un processus utilisateur ne corrompe un objet système.
- Il existe des versions multi-niveaux de certains SGBD (par exemple Oracle Label Security).

3.2.1. Principe de la politique MAC :

Dans les modèles **MAC**, un niveau de sécurité est affecté à chaque sujet et à chaque objet. Le niveau de sécurité associé à un objet s'appelle le niveau de classification alors que le niveau de sécurité associé à un sujet s'appelle le niveau d'habilitation. La politique de sécurité est obligatoire c'est-à-dire qu'elle s'impose à tous les utilisateurs et ne peut être modifiée. Si l'objectif est de garantir la confidentialité des données alors la politique de sécurité obligatoire (que l'on appelle aussi la politique de sécurité **multi-niveaux**) est la suivante : « les utilisateurs ont l'interdiction de prendre connaissance des données ayant un niveau de classification supérieur à leur niveau d'habilitation mais ont la permission de prendre connaissance des données classifiées à un niveau égal ou inférieur à leur niveau d'habilitation » (***Figure 3.4***) **[27]**.

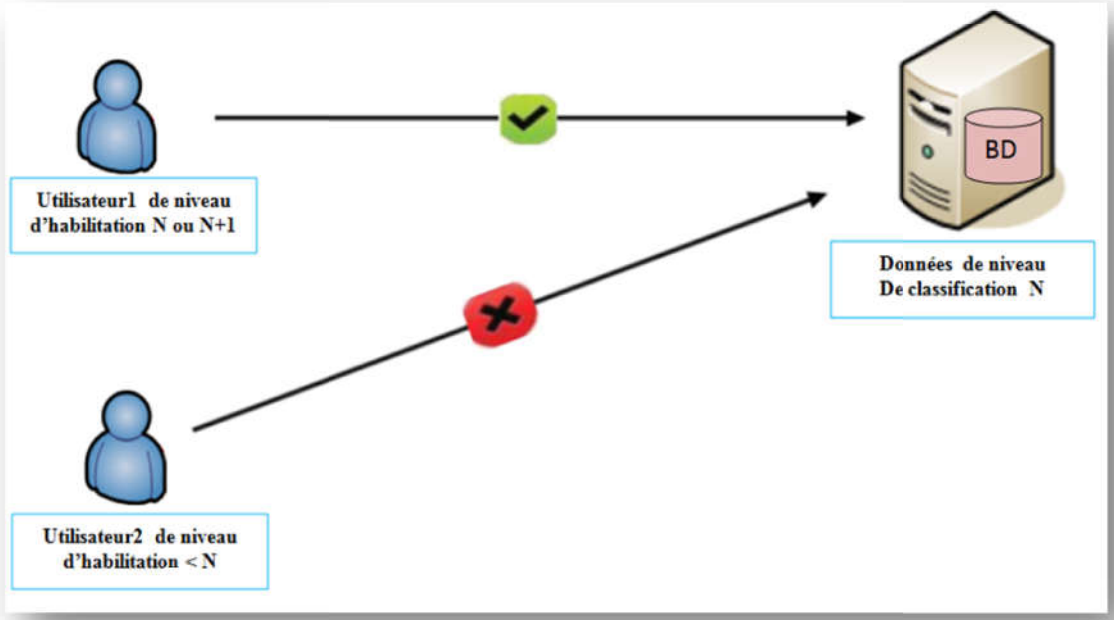

Figure 3.4. *Un exemple de modèle MAC* **[5].**

Bell & LaPadula ont montré qu'il était nécessaire d'appliquer les deux propriétés de contrôle d'accès suivantes pour garantir la politique de sécurité multi-niveaux **[27]**:

- ***No read up*** : cette propriété stipule qu'un sujet habilité à un certain niveau de confidentialité ne peut pas **lire** un objet classifié à un niveau supérieur.

- ***No write down*** : cette propriété stipule qu'un sujet habilité à un certain niveau de confidentialité ne peut pas **écrire** dans un objet classifié à un niveau inférieur.

Ces deux propriétés de contrôle d'accès ne sont toutefois pas suffisantes pour garantir la politique de sécurité multi-niveaux. Les modèles **MAC** s'inscrivent en fait dans la catégorie des modèles de contrôle de flux puisque le seul moyen de garantir totalement la sécurité multi-niveaux est de contrôler tous les flux d'informations possibles. L'information peut en effet transiter illégitimement par des canaux différents des simples opérations de lecture/écriture **[27]**.

3.2.2. Points faibles de MAC :

En revanche, l'utilisation des modèles **MAC** est complexe. Soit ils fournissent une politique trop restreinte, trop peu générale (BLP, BIBA, Clark-Wilson), et sont

alors difficile à déployer en pratique. Soit ils fournissent des mécanismes génériques **DTE**, mais alors le travail à fournir pour définir la politique de sécurité est bien plus exigeant, et n'inclut pas de garantie contre les erreurs de l'administrateur qui la définit [32]. La politique **MAC** est quelques fois difficile à administrer, trop rigide, ce que limiter la diffusion de l'information (à l'inverse de **DAC**). Elle est très appropriée pour des systèmes de haute sécurité **[30]**.

3.3. Contrôle d'accès à base de rôles RBAC :

Le coût de maintenance du contrôle d'accès peut rapidement devenir prohibitif dans les modèles précédemment cités. En particulier dans les systèmes industriels où l'accès aux objets peut se faire par des centaines ou milliers de sujets. La gestion de ces autorisations entraîne une explosion combinatoire qui ralentit l'évaluation du contrôle d'accès et rend complexe l'attribution ou la révocation de droits. De plus, ni la rigidité des modèles **MAC**, ni le manque de contrôle des modèles **DAC** ne sont réellement satisfaisants pour des systèmes industriels où les autorisations doivent pouvoir être déléguées, mais aussi contrôlées. C'est en réponse à ces problématiques qu'est apparu le contrôle d'accès à base de rôles, appelé **Role-Based Access Control (RBAC) [28]**.

3.3.1. Principe de la politique RBAC :

Le modèle de contrôle d'accès à base de rôle **RBAC** a été proposé pour présenter une nouvelle organisation des droits centrée sur le concept de rôle. Un rôle représente une fonction dans le cadre d'une organisation. Utiliser le rôle comme intermédiaire entre les sujets et les permissions facilite et simplifie les tâches d'administration en diminuant le nombre d'affectations à manipuler **[18]**.

La Figure 3.5 illustre la mécanique **RBAC** : les sujets obtiennent des autorisations sur des ressources grâce à des rôles qui leur sont attribués, eux-mêmes associés à un ensemble de permissions. Comme il y a normalement nettement moins de rôles que d'utilisateurs et de ressources, cela simplifie grandement la gestion des autorisations **[28]**.

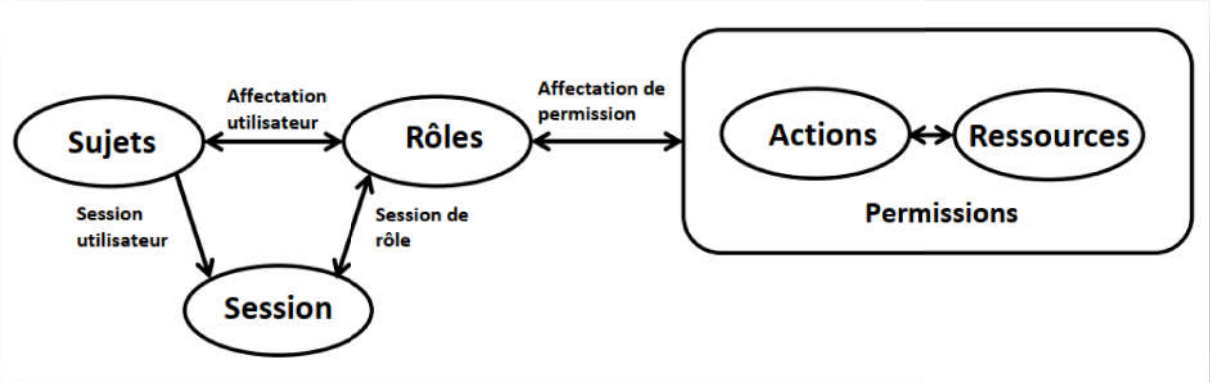

Figure 3.5. *Modèle RBAC* **[28].**

Les principes de base du modèle **RBAC** sont les suivants **[27]** :

- Alors que dans les modèles **DAC**, les permissions ont trait à des opérations de bas niveau telles que les opérations de lecture/écriture, dans les modèles **RBAC** elles concernent des tâches de nature organisationnelle telles que « transférer de l'argent », « acheter un billet d'avion » etc.
- Dans les modèles **RBAC**, le concept de rôle correspond à une fonction professionnelle. Les permissions sont accordées à des rôles et non pas à des utilisateurs. Les rôles sont ensuite distribués aux utilisateurs en fonction de leurs responsabilités au sein de l'organisation. Une même permission peut être affectée à différents rôles et différents rôles peuvent être attribués à un même utilisateur.
- Les modèles **RBAC** offrent une solution pour implanter des mesures de type **séparation des tâches**. Le principe de la séparation des tâches prévoit qu'un même utilisateur ne peut effectuer des tâches qui pourraient être orchestrées pour mettre œuvre des opérations frauduleuses, comme par exemple « autoriser un paiement » et « effectuer un paiement ». Ce principe peut aisément être garanti avec les modèles **RBAC** dans la mesure où deux rôles peuvent être déclarés comme étant mutuellement exclusifs. Deux rôles mutuellement exclusifs ne peuvent alors être affectés à un même utilisateur.

Le **RBAC** est largement adopté par les entreprises et les industriels et a été appliqué dans de grandes structures. Les logiciels commerciaux Trusted Solaris, Windows Authorization Manager, Oracle 9, Sybase, Adaptive Server Microsoft Active

Directory, la plupart des SGBD commerciaux, FreeBSD et Wikipedia ont mis en œuvre tout ou partie des principes des modèles à base de rôle **[22]**.

3.3.1. Les sous-modèles (famille) de RBAC :

La spécification de modèle RBAC comprend les sous-modèles suivants (***Figure 3.6***) **[18]**:

- Le modèle **RBAC0** ou « the flat model », qui présente les concepts et relations de base c.à.d. le noyau du modèle.
- Le modèle **RBAC1** ou « the hierarchical model », qui reprend le modèle RBAC0 et introduit la notion de hiérarchie entre rôles.
- Le modèle **RBAC2** ou « the constrained model », qui reprend le modèle RBAC0 et introduit la notion de contrainte.
- Le modèle **RBAC3** ou « the symmetric model », qui reprend les modèles RBAC1 et RBAC2 et prend en compte les interactions entre contraintes et hiérarchie.

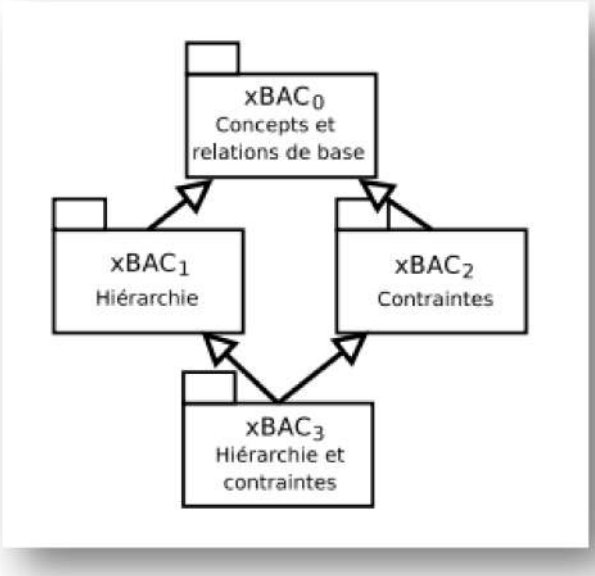

Figure 3.6. *Famille x-BAC (UML)* **[22].**

Ces raffinements successifs illustrent une orientation générale des recherches sur les modèles de contrôle d'accès : à partir d'un noyau, introduisant les concepts et relations principales du modèle, des enrichissements supplémentaires sont proposés. Cette structuration de la famille des modèles **RBAC** a été reprise par exemple dans les modèles **TBAC** (Task Based Access Control) et **GEO-RBAC (**Geospatial aware role

based access control), c'est la raison pour laquelle le terme ***x*-BAC** est utilisé dans la figure précédente **[22]**.

3.3.2. Modèles de contrôle d'accès dérivés de RBAC [18] :

Le modèle **RBAC** a été largement adopté par l'industrie et par la communauté de recherche. Il a déclenché un renouveau des modèles de contrôle d'accès et plusieurs propositions ont été faites pour ajouter de nouveaux concepts ou notions au modèle **RBAC** de base : par exemple le temps, la localisation, le contexte spatial, la position géographique de l'utilisateur etc. Nous classons dans cette section les modèles qui couvrent la plupart de ces nouvelles notions ou concepts dérivés de **RBAC** :

- **La notion d'équipes introduite par le modèle TMAC :**
  La notion d'équipe a été proposée dans le modèle **TMAC** (TeaMbased Access Control). Les permissions sont associées aux rôles ainsi qu'aux équipes. La notion d'équipe a été introduite pour représenter des aspects transversaux des rôles qui ne sont pas directement exprimables dans les modèles **RBAC**. Dans **TMAC**, l'objectif est d'accorder à chaque utilisateur membre d'une équipe des permissions accordées aux autres membres de l'équipe qui sont actifs.
- **La notion de localisation et d'information spatiale dans des applications mobiles introduites par le modèle LRBAC (**Location-aware role-based access control)**:**
  LRBAC étend le modèle **RBAC** pour que le contrôle d'accès puisse être établi en prenant en compte les informations de localisation. Un tel modèle a été proposé pour autoriser ou interdire l'accès lorsque les systèmes sont dans ou hors d'une zone d'opération définie. Ce modèle, proposé dans le cadre de la prolifération d'équipements mobiles, utilise la localisation logique des utilisateurs et/ou des systèmes comme paramètre contextuel.
- **La notion d'information spatiale et plus particulièrement la position physique de l'utilisateur dans des dispositifs comme GPS introduit par le modèle Geo-RBAC (**Geospatial aware role based access control)**:**
  Le modèle **Geo-RBAC**, étend le modèle **RBAC** en définissant de nouveaux concepts spatiaux pour représenter la position des sujets et celles des

objets. Ces nouveaux concepts sont utilisés pour limiter géographiquement l'utilisation des rôles. Le principe proposé dans **Geo-RBAC** est de comparer une position physique, supposée obtenue de façon fiable (par exemple la localisation GPS), à des positions logiques (exemples : route, ville, région) auxquelles sont associées des rôles géographiques

- **La notion de temps et des contraintes de temps dans les systèmes de «Workflow» introduite par le modèle GTRBAC (**Generalized temporal role based access control) **:**

  GTRBAC étend le modèle **RBAC** afin d'exprimer un large éventail de contraintes temporelles. En particulier, le modèle permet d'exprimer le temps et des contraintes temporelles sur les rôles, l'affectation des rôles aux utilisateurs, et l'affectation des permissions aux rôles. Ce modèle répond aux besoins précis d'applications avec une contrainte temporelle forte, comme les systèmes intégrant des workflows où la notion de temps est importante. Ces systèmes sont utilisés par des organisations désirant spécifier des règles d'autorisation qui permettent ou interdisent l'accès à des ressources pendant un intervalle de temps donné.

3.4. Modèles de contrôle d'accès à base des tâches:

En parallèle des travaux originaux sur **RBAC**, le modèle **TBAC** (Task Based Access Control) a été conçu afin d'activer une permission par rapport aux tâches effectuées par l'utilisateur. L'idée essentielle de ce modèle consiste à ajouter la notion de tâche dans des règles d'autorisation. Cela permet de définir les permissions qu'un sujet peut activer selon la tâche qui est encours. Chaque étape d'autorisation correspond à certaines activités ou tâches dans le contexte plus large d'un workflow de l'organisation. Le modèle **TBAC** fut le premier modèle à introduire le concept de tâche. **TBAC** va au-delà des modèles **IBAC** (Identity Based Access Control) où les actions correspondent généralement à des commandes élémentaires (comme la lecture du contenu d'un objet ou l'écriture dans un objet) pour structurer et contrôler la réalisation d'actions composites, appelées tâches ou activités. Il ajoute une étape d'autorisation (authorization step) qui permet de définir les permissions (enabled

permissions) qu'un sujet (executor-trustee) peut activer selon la tâche qui est en cours. Ainsi, **TBAC** offre une approche pour différencier l'affectation et l'activation des permissions par rapport à des tâches données aux utilisateurs au sein de l'organisation. La notion de tâche permet de contrôler les activités exercées par les utilisateurs d'un système d'information au sein de l'organisation **[18]**.

La figure suivante montre le cycle de vie d'une autorisation **TBAC** :

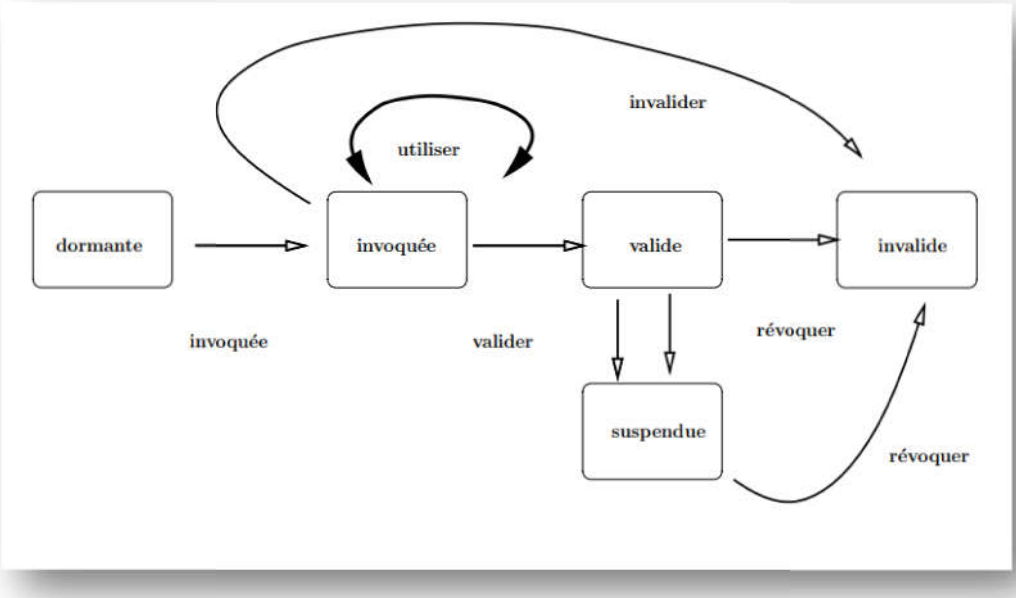

***Figure 3.7**. Cycle de vie d'une autorisation dans le modèle TBAC* **[31].**

TBAC peut parfaitement être adapté et intégrer la notion de rôle. C'est dans cet esprit que le modèle **TR-BAC** (Task and Rôle Based Access Control) a été défini. Dans ce cas, les droits sont activés en fonction d'un rôle et portent sur la réalisation des tâches. **TBAC** ou **TR-BAC** présente l'inconvénient de ne pas prendre en compte des contraintes sur les horaires ou périodes d'accès pendant lesquels les utilisateurs sont en charge de la réalisation de leurs activités. Le manque constaté est couvert par d'autres modèles formels de contrôle d'accès **[18]**.

3.5. Modèle de contrôle d'accès à base d'organisation OR-BAC:

Le modèle de contrôle d'accès **Or-BAC** (Organization Based Access Control) vise à résoudre certains problèmes rencontrés par les premiers modèles de contrôle d'accès des années 90 et à établir une politique de contrôle d'accès plus abstraite. Il s'intéresse, non seulement aux permissions, mais aussi aux interdictions, obligations et

recommandations dans une politique de sécurité. **Or-BAC** prend le concept de rôle dans **RBAC**. En plus de ce concept, il ajoute des nouveaux concepts pour structurer les sujets, les objets et les actions **[18]**.

3.5.1. L'organisation [18]:

Le concept central de ce modèle est la notion d'organisation comme son nom l'indique. Une organisation peut être un groupe structuré de sujets jouant des rôles déterminés. Ce peut être un hôpital, une clinique médicale, un service d'urgence…etc. L'organisation représente l'ensemble des rôles, des activités, et des vues qui représentent les abstractions respectives des utilisateurs, des opérations et des objets par rapport à une organisation donnée. Par exemple, un utilisateur est lié à un ensemble de rôles pour une organisation donnée. Il peut être affecté à d'autres rôles pour une autre organisation.

Le fait d'introduire ce concept « organisation » comme un élément de base dans le modèle de contrôle d'accès permet de structurer les droits en rassemblant plusieurs notions comme le rôle de **RBAC** et l'équipe de **TMAC**. Ces derniers définissent des relations binaires entre l'utilisateur et le rôle dans **RBAC**, ou entre l'utilisateur et l'équipe dans **TMAC**. **Or-BAC** définit des relations ternaires entre les organisations, les sujets et les rôles.

3.5.2. Les sujets et les rôles :

L'entité **Sujet** est utilisée différemment selon les modèles de sécurité. Dans le modèle **OR-BAC**, un sujet peut être soit une entité active, c'est-à-dire un utilisateur, soit une organisation. Un sujet joue un rôle dans une organisation. Ce qui veut dire que l'utilisateur ayant plusieurs rôles peut activer soit tous les rôles soit un sous-ensemble de ses rôles, dans n'importe quelle équipe à laquelle il participe. Dans la pratique, même si un utilisateur possède plusieurs rôles, il n'a pas forcément le droit de les jouer dans toutes les équipes auxquelles il appartient **[21]**.

3.5.3. Les objets et les vues:

Dans notre modèle, l'entité **Objet** représente principalement les entités non actives comme les fichiers, les courriers électroniques, les formulaires imprimés, etc.

Nous l'appelons : entité **Vue**. De manière intuitive, une vue correspond, comme dans les bases de données relationnelles, à un ensemble d'objets qui satisfait une propriété commune. Par exemple dans un système de fichier administratif, la vue « dossiers administratifs » correspond à l'ensemble des dossiers administratifs des patients, alors que la vue « dossiers médicaux » correspond aux dossiers médicaux des patients **[21]**.

3.5.4. Les actions et les activités [21]:

Les politiques de sécurité spécifient les accès autorisés aux entités passives par des entités actives et régulent les actions opérées sur le système. Dans notre modèle, l'entité *Action* englobe principalement les actions informatiques comme lire, écrire, envoyer, etc. Le schéma dans la Figure 3.8 fait apparaitre les deux niveaux de politique **Or-BAC** (abstrait et concret) ainsi que les différentes relations existant entre les entités de ces deux niveaux.

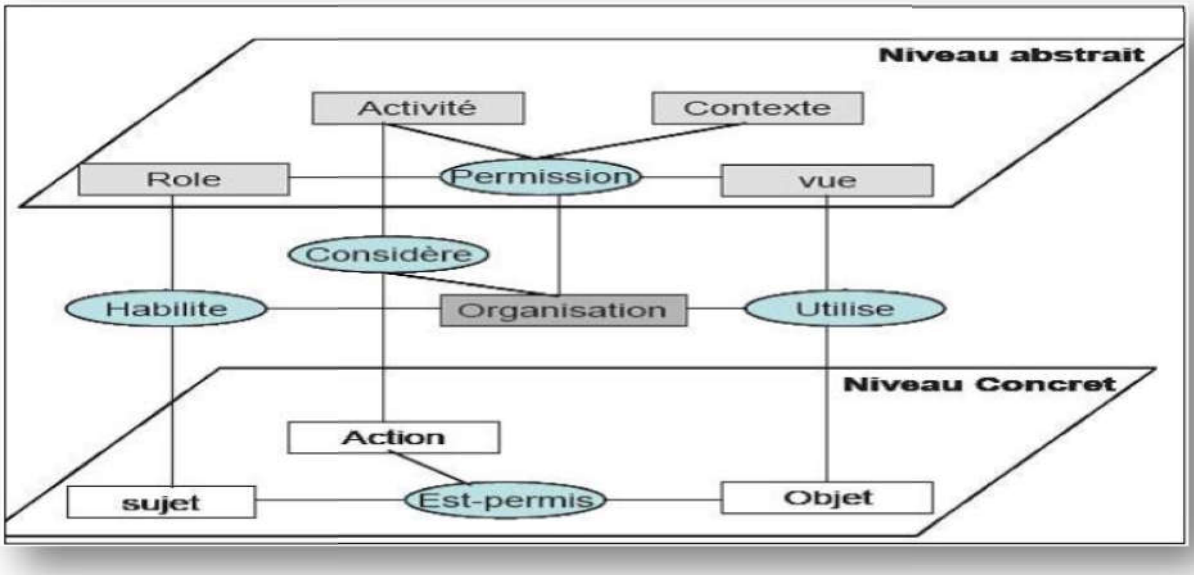

Figure 3.8. *Le modèle Or-BAC* **[18].**

3.6. Modèles de contrôle d'accès à base de contexte (CBAC):

Dans un nombre d'applications de plus en plus grand, la politique de sécurité ne peut plus être définie au moyen de règles d'autorisation statiques. Dans de telles applications, les privilèges accordés aux utilisateurs dépendent de conditions contextuelles. Les modèles de contrôle d'accès qui permettent l'expression de règles

dynamiques où la distribution des autorisations dépend de conditions contextuelles appartiennent à la famille des modèles **CBAC (**Context Based Access Control**)** **[31]**.

La définition du contexte en informatique est délicate, une définition communément admise est proposée par Dey: « *le contexte est l'ensemble de toutes les informations qui peuvent être utilisées pour caractériser la situation d'une entité. Une entité pouvant être un acteur, un lieu, ou un objet de l'environnement considéré comme utile à l'interaction entre un utilisateur et une application, y compris l'utilisateur et l'application eux mêmes* » **[24]**.

Les modèles suivants peuvent être considérés comme étant des modèles **CBAC [27]** :

- Dans le modèle **ABAC** (Attribute-Based Access Control), les autorisations dépendent de conditions booléennes s'appliquant aux attributs du sujet, de l'objet et de l'environnement.
- Certaines propositions étendent le modèle **RBAC** pour prendre en compte des conditions contextuelles telles que la position de l'utilisateur ou le temps. Dans la plupart de ces approches les rôles peuvent être activés en fonction de conditions spatiales ou temporelles.
- Le modèle formel de contrôle d'accès **CRBAC (**Context role based access control) introduit en plus du « rôle » la notion de « contexte ». les rôles sont composés : rôles de sujets comme dans **RBAC**, et rôles contextuels pour capturer des informations de sécurité liées au contexte **[40]**.
- Le modèle **OrBAC**, définit une taxonomie complète de contextes (spatial, temporel, provisionnel etc.) et fournit un cadre formel fondé sur la logique du premier ordre pour exprimer des règles contextuelles (***Figure 3.9***). Il intègre le modèle d'administration **AdOrBAC (**Administration model for Or-BAC).

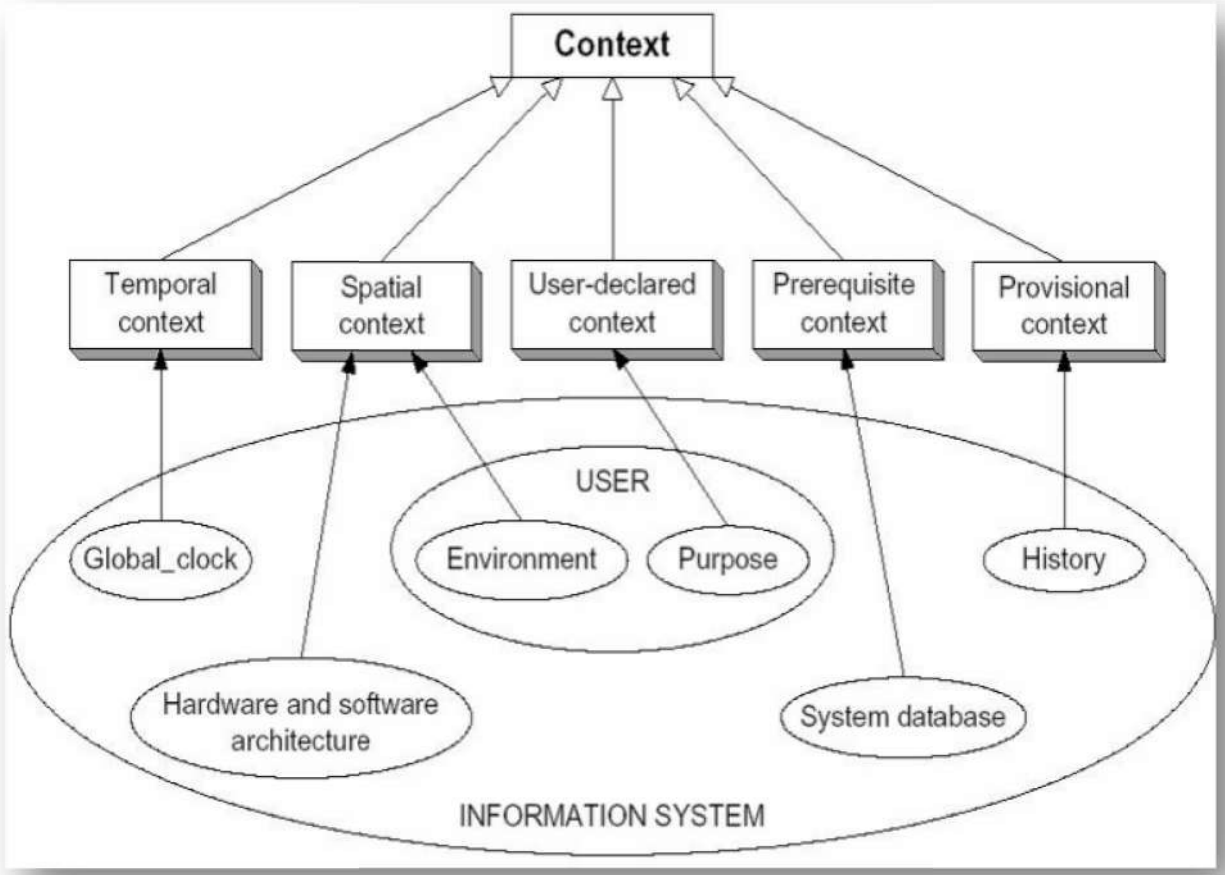

Figure 3.9. *Contexte dans OR-BAC* **[18].**

Une caractéristique importante des modèle **CBAC** réside dans le fait qu'ils permettent d'exprimer des règles d'autorisation qui ne requièrent pas d'authentifier les utilisateurs. Un utilisateur peut en effet obtenir un accès à une information simplement parce que certaines conditions contextuelles sont remplies. Cette capacité à accorder ou refuser un accès sans avoir à authentifier l'utilisateur est très utile dans le cadre d'applications Web interconnectées **[27]**.

4. Conclusion :

Dans ce chapitre nous avons présenté plusieurs modèles de contrôle d'accès existants qui ont été conçus pour assurer la sécurité des données dans des contraintes différentes. En commençant par les modèles de contrôle d'accès classiques (**DAC**, **MAC**) qui ont été proposés pour répondre aux exigences des systèmes d'exploitation. D'autres modèles sont bien adaptés aux structures très organisées, où la confidentialité est une priorité (les modèles à niveau dits aussi « orienté flux » comme **TBAC** et **WRBAC** : Worflow Role Based Access Controle). D'un autre côté, certaines structures ont des organisations beaucoup plus souples et complexes. Elles souhaitent alors définir leurs propres modèles de contrôle d'accès adaptés à leur besoins. Ils réutilisent les concepts présentés précédemment (rôle, organisation, hiérarchies) pour créer leurs propres politiques de contrôle d'accès. D'autres modèles ont préféré compléter le concept de rôle dans des modèles **RBAC** par une ou plusieurs notions pour s'adapter à d'autres situations, comme le contexte dans **Or-BAC** ou **CRBAC.**

Après l'étude que nous avons fait sur les différents politique de control d'accès existantes nous avant remarqué que :

- Les propositions basées sur les rôles forment la plus grande famille des modèles de contrôle d'accès et sont les plus étudiées et utilisés dans le monde de sécurité informatique.
- Il est difficile de considérer qu'un modèle de contrôle d'accès est meilleur qu'un autre : cela dépend essentiellement du domaine d'application et du type de l'organisation qui le met en œuvre.

Partie 02
Modélisation et Implémentation de la Solution

Chapitre 04
Modélisation d'un système de Contrôle d'accès pour un Cloud Computing privé

1. Introduction :

Un Cloud Computing privé fournit des services de Cloud Computing aux utilisateurs autorisés via Internet ou un réseau interne. Puisqu'il ne peut être utilisé que par un groupe d'utilisateurs clairement défini, le Cloud privé est aussi appelé Cloud d'entreprise ou encore Cloud interne. Dans notre modélisation, on s'intéresse à la conception d'un Cloud privé dont l'entreprise exploite elle-même l'infrastructure informatique de ses services de Cloud.

Notre travail se déroule à la direction de moudjahidine de la wilaya de Blida pour une gestion sécurisée des demandes des bénéficiaires (moudjahidine ou ayant droit) pour profiter d'un séjour au nivaux des centres de repos.

Dans ce chapitre, au début, nous allons présenter un aperçu global sur les activités du ministère de moudjahidine suivi par la définition de la méthode UML. Ensuite nous proposons une modélisation du système par la méthode UML. Nous continuons le chapitre par une conception sécurisé du Cloud Computing privé en se basant sur la politique de contrôle d'accès RBAC et nous terminons par une conclusion.

2. Présentation de Ministère de moudjahidine :

Le ministère des moudjahidine et des ayants droit est l'administration algérienne chargée du domaine des moudjahidine depuis 1962 qui comprend l'ensemble des activités destinées à assurer la protection et la promotion des moudjahidine et des ayants droit (Bénéficiaires). Ce Ministère dispose d'une administration centrale (La direction centrale) qui gère :

- Un ensemble des directions opérationnelles dans les wilayas (Une direction dans chaque wilaya qui donne 48 directions).
- Un ensemble des centres de repos (15 Centres sur le territoire national).
- Le musée national de moudjahid.
- Un ensemble des musés régionales.
- Le centre national de traitement des victimes de la guerre de libération nationale.

3. Modélisation du système :

Pour la modélisation de notre système d'information nous avons opté pour la méthode UML.

3.1. Méthode UML [31]:

UML (**U**nified **M**odeling **L**anguage) est un langage de modélisation qui permet de spécifier, visualiser, construire, et documenter les artefacts des systèmes logiciels, ainsi que pour la modélisation d'entreprise et des systèmes non logiciels. Au niveau de Unified Modeling Language, deux éléments importants sont à noter. Le terme "unified" et le terme "langage". Le premier terme signifie que les auteurs ont essayé de regrouper les éléments importants des concepts objets, alors que le deuxième montre qu'il s'agit d'un langage de modélisation et non pas d'une méthode. UML est un langage qui permet de modéliser non seulement des applications informatiques ou des structures de données, mais également les activités d'un domaine : mécanique, biologie, processus métier. Ce langage de modélisation unifié repose sur deux concepts essentiels :

- La modélisation du monde réel au moyen de l'approche orientée objet.
- L'élaboration d'une série de diagrammes facilitant l'analyse et la conception des systèmes, et permettant de représenter les aspects statiques et dynamiques du domaine à modéliser ou à informatiser.

La naissance d'UML est due à la fusion des trois méthodes de référence dans le domaine de la modélisation objet durant les années 1990 et en 2006 la version UML2.0 devient un langage de modélisation de logiciels standard.

L'analyse de la thématique et les différentes problématiques posées par les outils existants ainsi que la compréhension des besoins utilisateurs de notre système a permis de dégager les fonctionnalités suivante :

3.1.1. Identification des acteurs :

3.1.1.1. Définition d'un acteur [19] :

Un acteur est l'utilisateur direct du système. Il doit avoir une bonne connaissance des fonctionnalités du système. Un acteur est une entité externe au système qui interagit ou dialogue avec lui. Il est identifié par des rôles joués par des personnes ou d'autres systèmes logiciels. On distingue :

- **Des acteurs primaires (ou principaux**) : utilisateurs du système pour l'accomplissement de leurs buts.
- **Des acteurs secondaires :** autres participants qui peuvent fournir ou recevoir de l'information, ou s'occuper de la supervision ou de l'entretien du système. Pour trouver les acteurs d'un système, il faut identifier quels sont les différents rôles qui vont devoir jouer ses utilisateurs. Il faut vérifier que les acteurs communiquent bien directement avec le système par émission et réception de messages.

3.1.1.2. Les acteurs du système :

En analysant les intervenant du notre système, nous recensons l'ensemble des acteurs suivants :

- **Administrateur Principal (AP):** C'est le responsable informatique au niveau de la direction centrale, il est le premier responsable sur l'ensemble du système (Infrastructure et logiciel).
- **Administrateur de direction (AD):** C'est le responsable informatique au niveau de la direction opérationnelle pour gérer le réseau local de la direction.
- **Directeur du centre (DC):** C'est le responsable informatique au niveau des centres des repos.
- **Bénéficiaire (BF)**: C'est une personne physique qui représente moudjahid ou l'une des personnes ayants droits.

***Remarque** : Le rôle et les taches de chaque acteur seront détaillées dans la section suivante.*

3.1.2. Diagramme de cas d'utilisation (DCU):

3.1.2.1. Définition d'un DCU :

Un digramme de cas d'utilisation (Use Case Diagram) permet de représenter graphiquement les cas d'utilisation **(*Figure 4.1*)**. Le fait qu'un acteur déclenche un cas d'utilisation est représenté par une flèche entre ces deux derniers **[3]**.

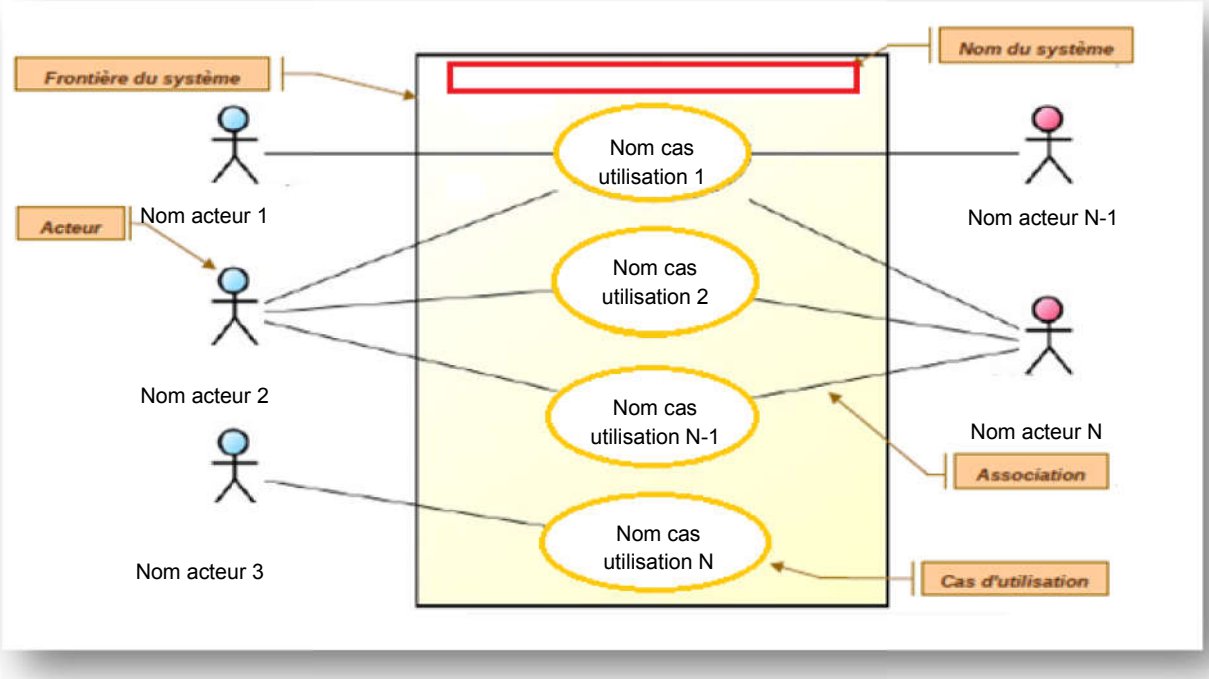

Figure 4.1. *Formalisme générale d'un DCU « Image illustrée de* **[19] ».**

3.1.2. 2. Les concepts d'un cas d'utilisation [20]:

La représentation d'un cas d'utilisation met en jeu trois concepts : l'acteur, le cas d'utilisation et l'interaction ou l'association entre l'acteur et le cas d'utilisation **(*Figure 4.2*)**.

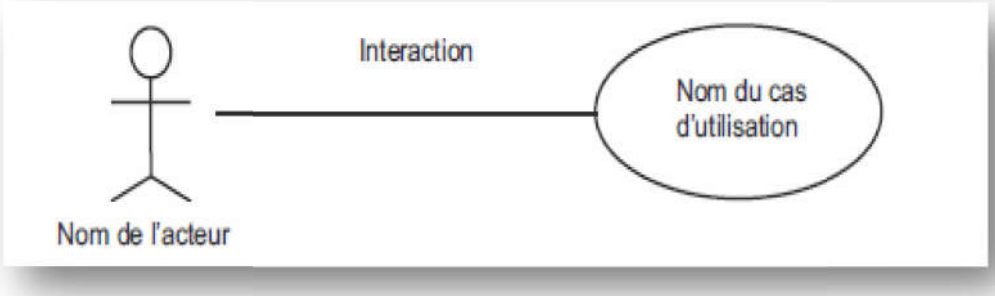

Figure 4.2. *Formalisme de base de représentation d'un cas d'utilisation* **[20].**

- **Acteur :** Un **acteur** est un utilisateur type qui a toujours le même comportement vis-à-vis d'un cas d'utilisation.
- **Cas d'utilisation :** Un **cas d'utilisation** correspond à un certain nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur. Un cas d'utilisation doit produire un résultat observable pour un ou plusieurs acteurs ou parties prenantes du système.
- **Interaction :** Une **interaction** permet de décrire les échanges entre un acteur et un cas d'utilisation.

Le **DCU** constitue la première étape de l'analyse UML en :

- Modélisant les besoins des utilisateurs.
- Identifiant les grandes fonctionnalités et les limites du système.
- Représentant les interactions entre le système et ses utilisateurs.

La figure suivante montre le diagramme de cas d'utilisation de l'acteur « Bénéficiaire ».

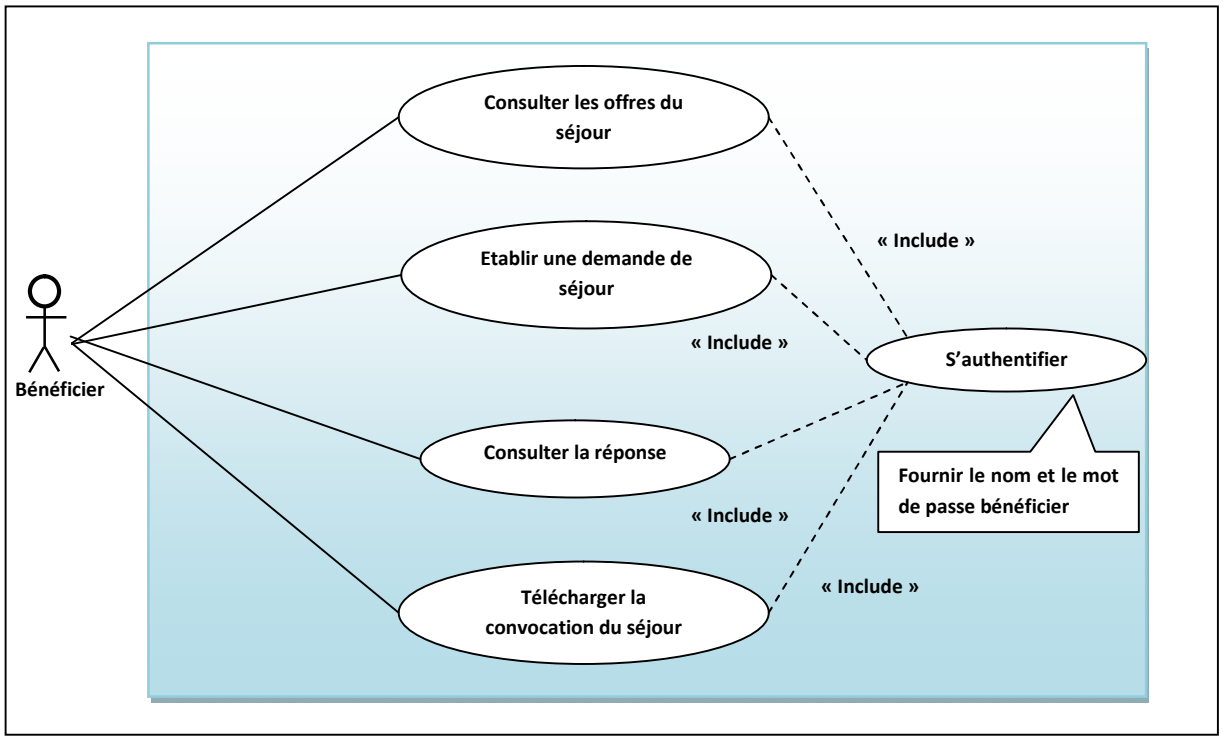

Figure 4.3. *Exemple d'un diagramme de cas d'utilisation (Acteur Bénéficiaire).*

3.1.3. Diagramme de classe (DCL) :

3.1.3.1. Définition de DCL [20] :

Le diagramme de classe constitue l'un des pivots essentiels de la modélisation avec UML (c'est le seul qui soit obligatoire lors de la modélisation objet d'un système). En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations.

La description du diagramme de classe est fondée sur :

- Le concept d'objet,
- Le concept de classe comprenant les attributs et les opérations,
- Les différents types d'association entre classes.

La figure ci-dessous montre le diagramme de classe de notre système d'information :

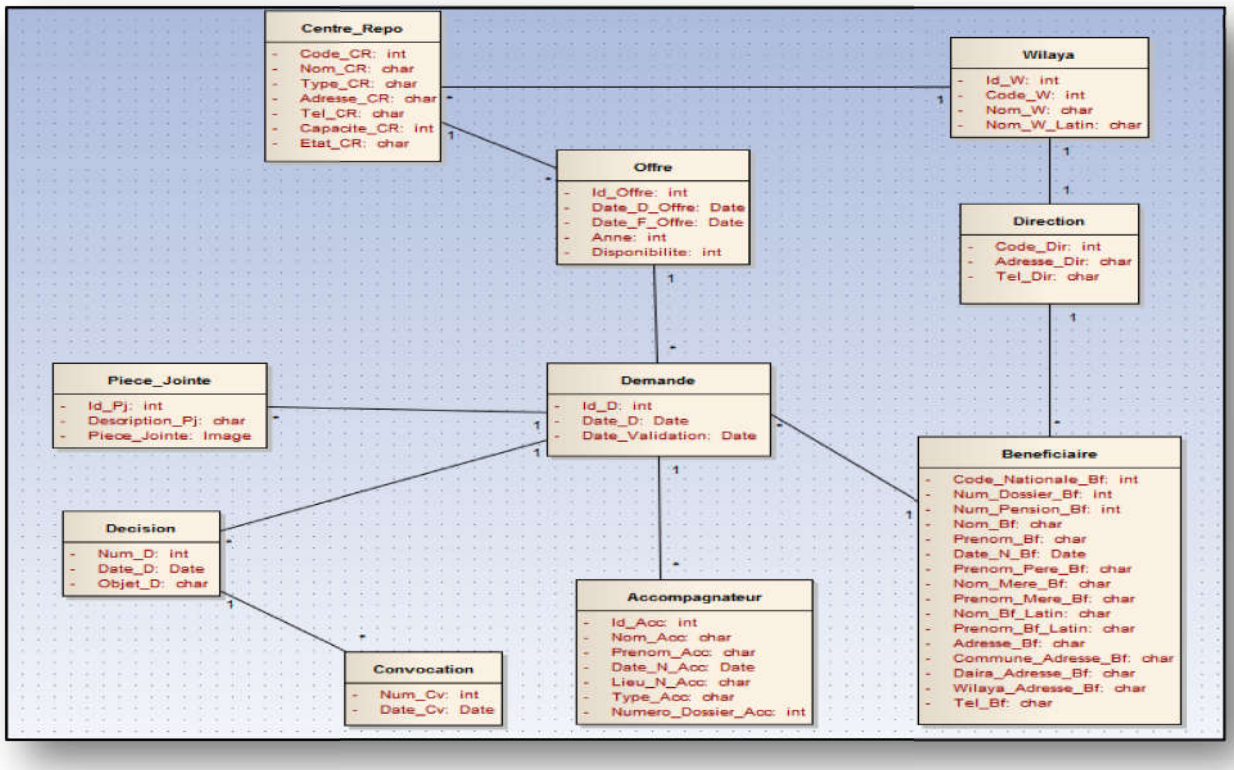

Figure.4.4. *Diagramme de classe de système d'information.*

4. Politique de contrôle d'accès utilisée :

Pour assurer la sécurité de notre système d'information nous avons adopté la politique de contrôle d'accès **RBAC** basée sur les rôles. Le modèle **RBAC** introduit les ensembles suivants (***Figure 4.5***) **[18]**:

- **Utilisateur** : l'ensemble des utilisateurs, où un utilisateur est une entité active, humaine ou logicielle.
- **Rôle** : l'ensemble des rôles, où un rôle est une fonction de travail dans le cadre d'une organisation liée à une autorité et des responsabilités.
- **Permission** : l'ensemble des autorisations afin d'effectuer des opérations sur un ou plusieurs objets protégés.
- **Opération** : l'ensemble des opérations.
- **Objet** : l'ensemble des objets ou des ressources.
- **Session** : une correspondance entre un utilisateur et un ensemble de rôles autorisés.
- **Utilisateur_Assignation : UA ⊆ Utilisateur × Rôle** : permet d'affecter des rôles aux utilisateurs.
- **Permission_Assignation : PA ⊆ Permission × Rôle** : permet d'affecter des permissions aux rôles.
- **Rôle_Hiérarchie : RH ⊆ Rôle × Rôle** : définit un ordre partiel sur l'ensemble Rôle, appelée héritage. Elle est aussi écrite par ≥ tel que rôle1 ≥ rôle2 implique que les permissions de rôle2 sont aussi des permissions de rôle1.
- **Session_Utilisateurs** : Session → Utilisateur, permet d'établir l'utilisateur d'une session.
- **Session_Rôles**: permet d'établir l'ensemble des rôles associés à une session.

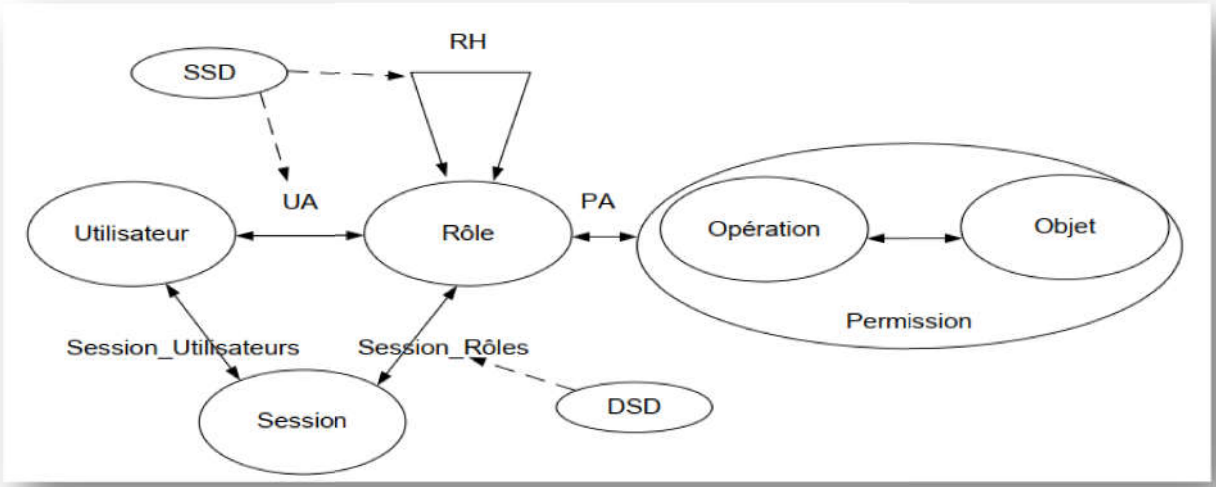

Figure 4.5. *Le modèle RBAC* **[18].**

4.1. Architecture générale du système :

Notre système de contrôle d'accès s'interpose entre l'interface utilisateurs et les services application et permit de contrôler les requêtes des utilisateurs (validation ou refus), comme il est représenté dans la figure 4.6.

Chaque utilisateur dispose d'une interface qui lui permet après avoir s'identifier par un nom utilisateur et un mot de passe d'exécuter des taches en fonction du rôle et d'autorisations accordées.

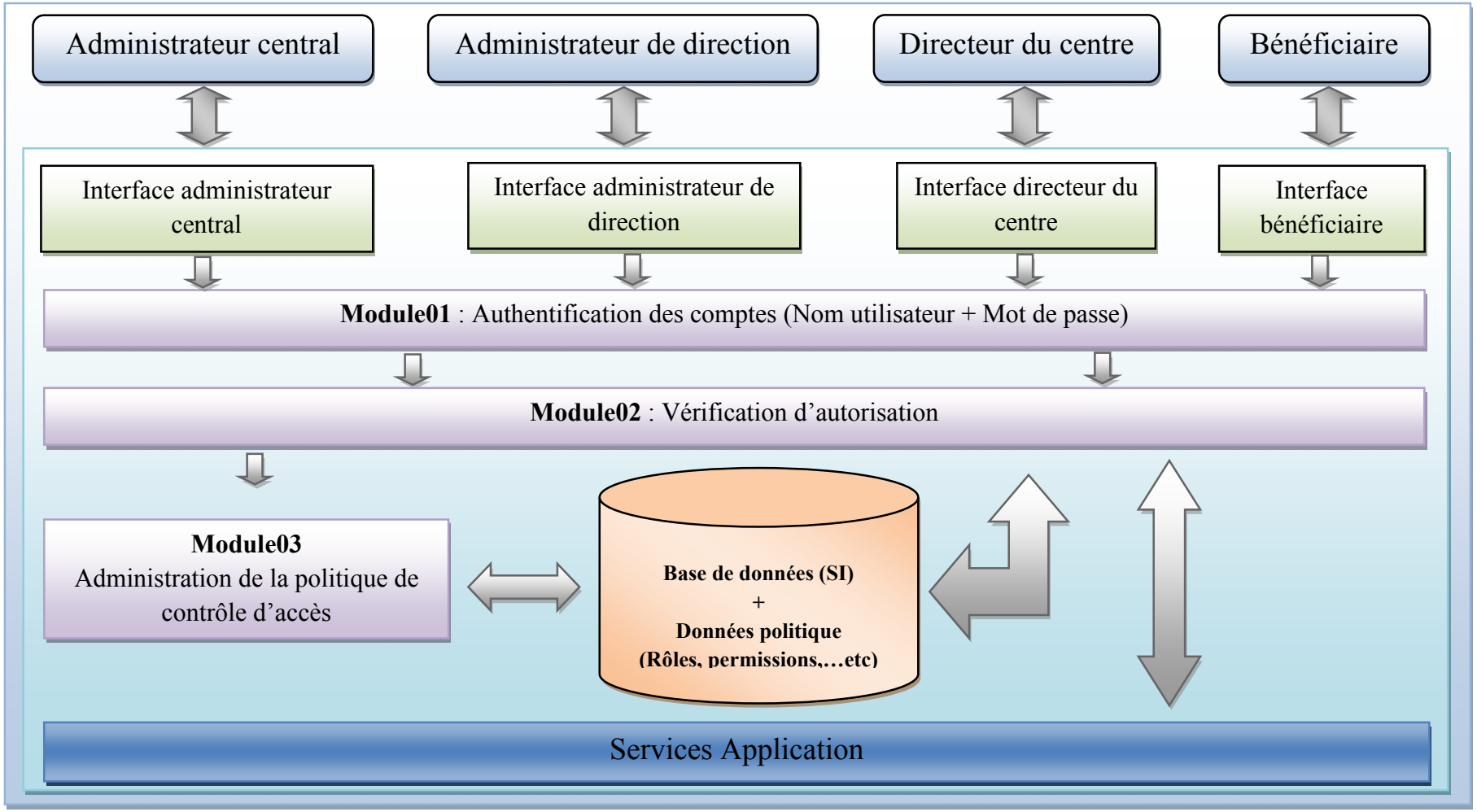

Figure.4.6. *Architecture du système.*

Quatre utilisateurs sont requis pour le système, où chaque utilisateur représente un rôle dans notre politique de contrôle d' accès. Les taches associes sont détaillées dans le tableau ci-dessous :

Acteurs	Taches
Administrateur Principal	- Administration du politique de contrôle : 1. Gérer les rôles, les permissions, et les utilisateurs. 2. Assignation des utilisateurs aux rôles et les rôles aux permissions. - Administration des comptes utilisateurs. - Consulter, chercher et transférer n'importe quel fichier de système.
Administrateur de direction	- Gérer les comptes bénéficiaires. - Consulter, rechercher et transférer n' importe quel fichier de système au niveau de sa direction seulement (exemple : Liste bénéficiaires, convocation, ..etc). - Etablir les convocations des bénéficiaires.
Directeur du centre de repo	- Etablir les offres du séjour. - Consulter, rechercher et valider les demandes de séjour des bénéficiaires (confirmation ou refus). - Etablir et transférer les listes des bénéficiaires acceptés.
Bénéficiaire	- La recherche des offres de séjours ouvertes au niveau de n'importe quel centre de repos dans le pays. - La saisie d'une demande de séjours. - Consulter la réponse de sa demande (confirmée ou refusée). - Télécharger la convocation du séjour.

***Tableau 4.1 :** Ensemble des taches d'utilisateurs du système.*

Comme il est montré dans la figure.4.6 le système se compose de trois modules principaux :

- Le module authentification des comptes.
- Le module vérification des autorisations.
- Le module administration de la politique de contrôle d'accès.

4.3. Module Authentification des comptes:

Le module Authentification se situe dans une couche d'administration des comptes utilisateurs. Cette tache est associée aux deux administrateurs (Centrale et Direction). Un utilisateur doit s'identifier avant d'accéder aux services d'application avec son identifiant et son mot de passe. Une session est crée automatiquement quand il est authentifié par le module authentification et elle sera détruite lors de sa déconnexion.

Avant de créer une session à un utilisateur, le module Authentification doit vérifier si l'utilisateur appartient à l'ensemble des utilisateurs USERS et que son mot de passe est correct.

4.3. Module Vérification d'autorisation:

Après la vérification de l'identité de l'utilisateur par le module Authentification, une session est initialisée avec comme propriétaire l'utilisateur authentifié. Le module Vérification contrôle toutes les opérations de l'utilisateur, gère les sessions et valide les requêtes des utilisateurs.

Les requêtes de l'utilisateur sont vérifiées par une fonction qui détermine si le sujet d'une session donnée est autorisé ou non à exécuter une opération donnée sur un objet donné.

4.4. Module administration de la politique :

L'administration de la politique de contrôle d'accès se fera à travers une interface qui permettra au administrateur centrale après authentification de :

- Définir et d'entretenir les ensembles de base (Utilisateurs, Rôles, Permission, …etc).

- Assigner les utilisateurs et les permissions aux rôles.
- Définir la hiérarchie entre les rôles et les contraintes de séparation des taches.

4.4.1. Gestion des permissions :

La gestion des permissions consiste à définir les ensembles :

- Action.
- Objet (vues).
- Permission (PRMS).

Dans notre système les vues de base de données sont considérer comme des objets (ils sont sauvegardés dans la classe Objet), les actions utilisateurs sont considère comme des opérations (elles sont sauvegardées dans la classe action). Les paires (Objet, Action) sont des permissions qui sont assignées aux rôles.

Voici un tableau d'exemple des permissions des opérations accordées sur des objets.

Identification de la permission	Identification de l'objet (Vue)	Action
PRM1	Liste bénéficiaires acceptée	Créer
PRM2	Liste bénéficiaires acceptée	Consulter
PRM3	Liste bénéficiaires acceptée	Transférer
PRM4	Demande bénéficiaires	Créer
PRM5	Demande bénéficiaires	Consulter
PRM6	Convocation	Créer
PRM7	Convocation	Transférer

Tableau 4.2. *Exemple des actions octroyées sur des objets (vues).*

4.4.2. Gestion des rôles :

La gestion des rôles comporte l'ajout et la suppression des rôles, la hiérarchie des rôles et la séparation des taches.

4.4.2.1. Ajout et suppression des rôles :

Dans la politique de contrôle l'ajout d'un rôle se fait en ajoutant le nom du rôle dans la classe ROLES, le nom du nouveau rôle ne doit pas exister dans cet ensemble. Pour supprimer un rôle ($\mathbf{R_x}$) il faut :

- Vérifier si le rôle appartient d'abord à la classe **ROLES**.
- Fermer toutes les sessions où le rôle ($\mathbf{R_x}$) et activé.
- Supprimer les assignations **Users-Roles** relatives au rôle ($\mathbf{R_x}$).

Dans notre système, la classe **ROLES** contient les rôles suivants : Administrateur central, Administrateur de direction, Directeur du centre, Bénéficiaires.

4.4.2.2. Hiérarchie des rôles :

La hiérarchie entre les rôles se fait en identifiant les inclusions entre les ensembles de permissions assignées à chaque rôle, si un rôle **role1** à tout les permissions qui sont assignées à un rôle **role2**, alors **role1** sera désigné comme un ascendant hiérarchique de **role2**. La figure ci-dessous montre la hiérarchie de rôles dans notre système :

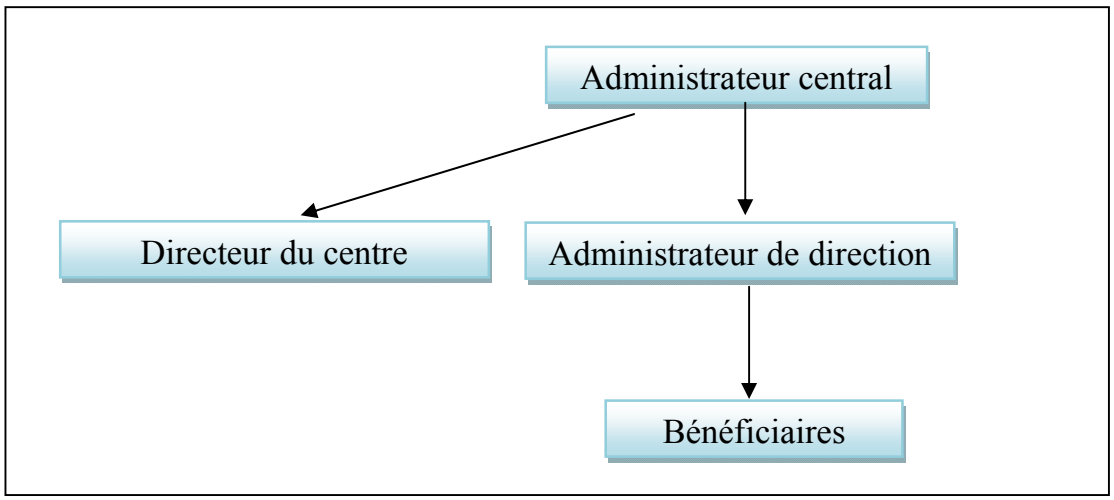

Figure 4.7. *Hiérarchie de rôles.*

Les hiérarchies des rôles sont sauvegardées dans la Classe **RôleHiérarchie** (RH) qui contient le rôle ascendant (R_A) et le rôle descendant (R_D).

4.4.2.3. Séparation des tâches :

Le module de la politique de contrôle apporte la possibilité de gérer les éventuels conflits entre rôles en ajoutant des contraintes pour exprimer la séparation de tâches et l'exclusion mutuelle entre rôles. Ainsi, pour interdire à un utilisateur d'être affecté à deux rôles qui sont en conflit,

Il existe deux types de résonnement **[18]**:

- Les séparations statiques **SSD** (Static Separation of Duties) : La SSD interdit l'affectation d'un utilisateur à deux rôles en conflit, et empêche qu'une hiérarchie de rôles amène un utilisateur à posséder les permissions de deux rôles en conflit.
- Les séparations dynamiques **DSD** (Dynamic Separation of Duties) : La DSD évite qu'un utilisateur possède deux rôles en conflit en même temps dans une même session.

La séparation des tâches est enregistrée dans la classe **SépareTache** (ST) qui contient les deux rôles à séparer (R1, R2), ainsi que le type séparation de taches (type) à utiliser, soit SSD pour statique, ou DSD pour dynamique.

4.4.3. Gestion des utilisateurs :

L'ajout d'un utilisateur se fait en présence des données nécessaire (identifiant, nom, prénom, mot de passe…). L'identifiant de l'utilisateur ne doit pas existé déjà dans la base de données.

Pour supprimer un utilisateur $User_x$ il faut :

- Vérifier si l'utilisateur $User_x$ appartient à la classe USERS.
- Fermer toutes les sessions de l'utilisateur $User_x$.
- Supprimer toutes les assignations Users-Roles relatives a l'utilisateur $User_x$.
- Supprimer l'utilisateur $User_x$ de la base de données.

4.4.4. Les assignations :

Les assignations (utilisateurs-rôles) et (rôles-permissions) permettent l'accès d'un utilisateur aux permissions assignées aux rôles correspondants.

La figure suivante montre le diagramme de classe de politique de contrôle d'accès RBAC.

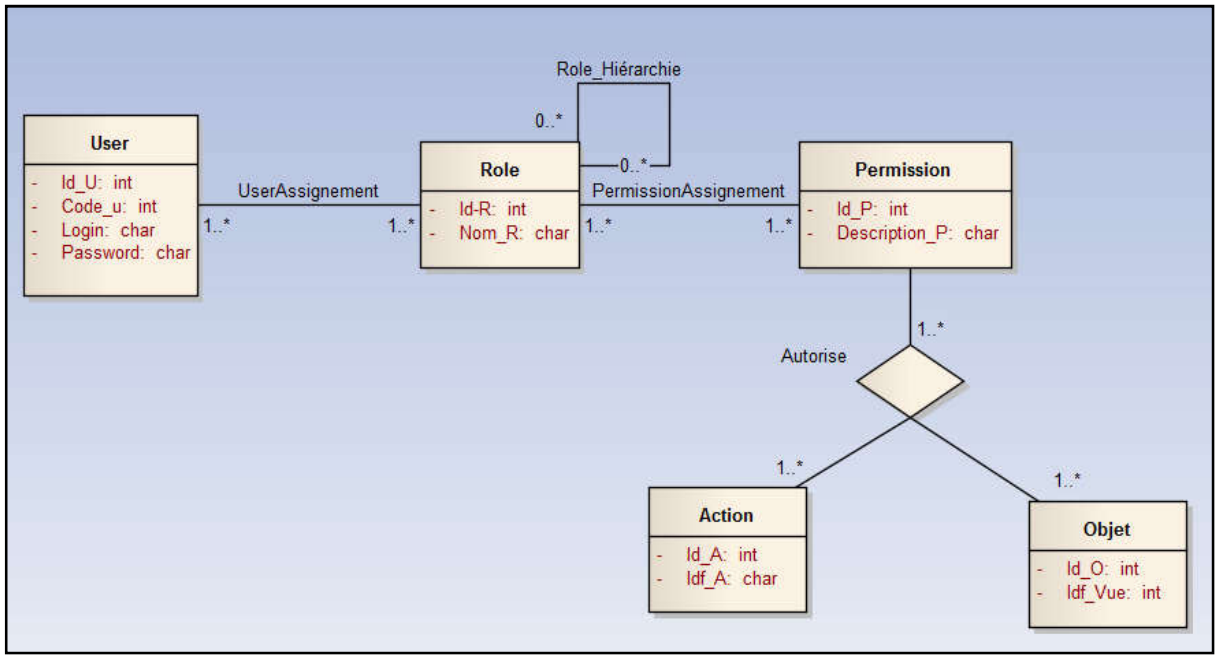

***Figure 4.8**. Diagramme de classe de la politique de contrôle d'accès RBAC.*

5. Modélisation du Cloud sécurisé :

5.1. Etude de l'existant :

5.1.1. Infrastructure réseaux :

En menant un audit sur l'architecture réseau de la direction du moudjahidine de la wilaya de Blida on a pu avoir l'architecture schématisée dans la figure ci-dessus :

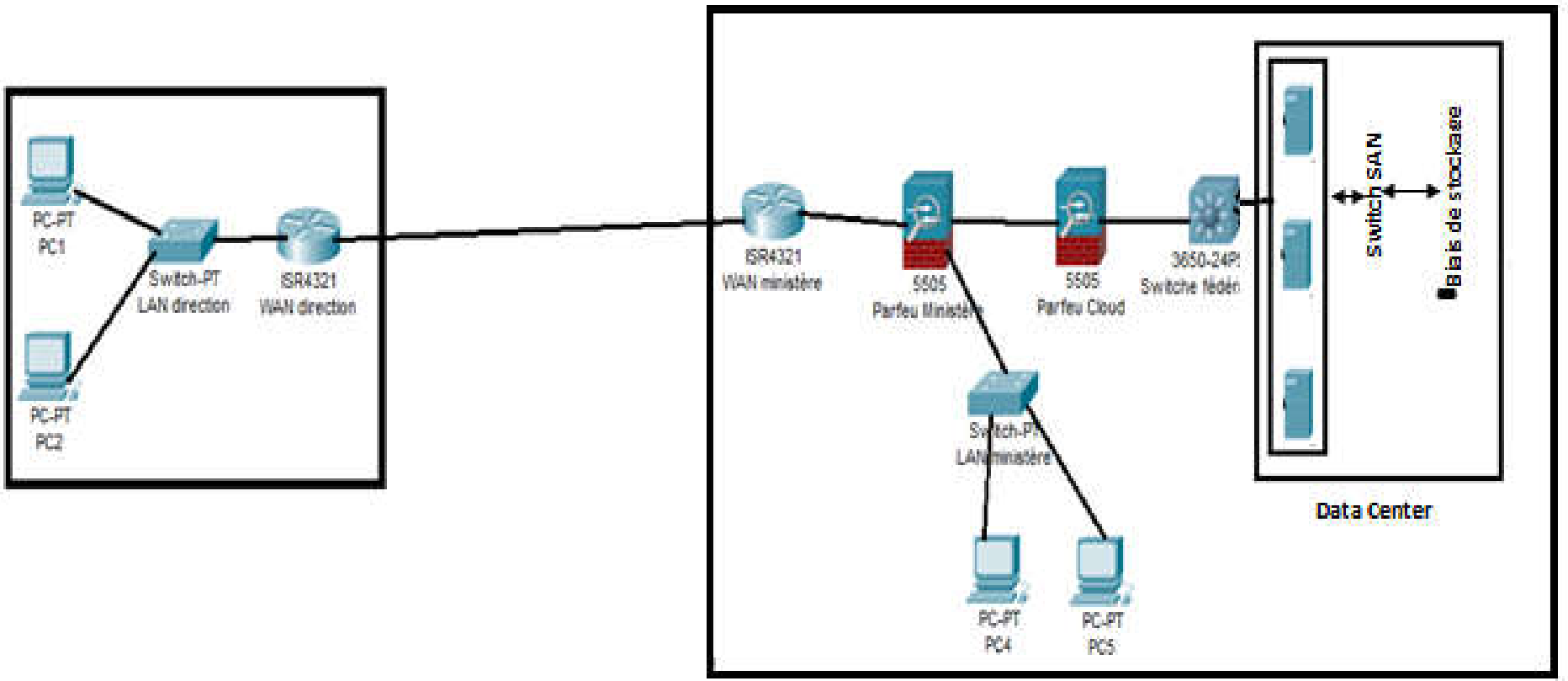

Figure 4.9. *Topologie physique du réseau existant.*

Sachant que :

- La direction centrale est équipée par :
  - Un Datacenter pour la protection des différents serveurs.
  - Un réseau Local pour les services de ministère.
  - Un ensemble des serveurs (Web, Messagerie, Base de données, …) et un biais de stockage.
  - Un Switch fédérateur[1] pour relier l'ensemble des serveurs.
  - Un pare-feu pour la protection des serveurs et un pare-feu pour la protection du ministère vers l'extérieur.
  - Un routeur pour relier le ministère avec le réseau intranet du système.
- Chaque structure (direction de wilaya ou centre de repo) est équipée par :
  - Un réseau Local.
  - Un routeur pour relier la structure avec le réseau intranet du système.

L'ensemble des réseaux sont reliés par la technologie MPLS[2] assurée par Algérie Télécom via des routeurs dotés d'une carte WIC (**W**AN **I**nterface **C**ard).

L'adressage des réseaux locaux est à l'aide de l'adresse **10.2O.X.X / 24** telque :

- Le troisième octet pour faire différencier les différentes directions selon le numéro de wilaya pour créer des sous réseaux différents, un autre numéro pour la direction centrale (supérieur à 48) et autre numéro pour le réseau des serveurs.
- Le quatrième octet pour faire différencier les différents hôtes pour le même réseau LAN et le numéro 1 pour la sortie du routeur.

5.1.2. Services :

Les services offerts pour le moment sont des services de bases proposés par Windows Server 2012 version datacenter (Messagerie, fichier, Web, Activ Directory, …etc) et un serveur base de données pour gérer les dossiers des bénéficiaires.

[1] **Switch Fédérateur :** le cœur du réseau. Généralement le switch fédérateur comporte des ports en fibre optique et assure des liaisons en Gigabit vers les switchs d'étage.

[2] MPLS : Acronyme issu de l'anglais pour « MultiProtocol Label Switching » est une technologie de transport de data basée sur la commutation d'étiquettes (labels) insérées à l'entrée d'un réseau MPLS et retirées à sa sortie. Ce protocole a pour but d'accélérer les flux de trafic réseau.

5.1.3. Sécurité :

Puisque le réseau utilisé est un réseau intranet sans aucun accès vers l'extérieur (Internet), il optimise les risques d'attaque externe, an niveau de l'organisme la sécurité est assuré par trois niveaux (Création des VLAN pour chaque direction, authentification Windows assuré par Active Directory et une authentification assuré par le SGBD pour l'accès au données des bénéficiaires dans le serveur base de données.

5.1.4. Critique de l'existant :

On constate l'existence d'infrastructure du Cloud avec d'autres services qu'on les juge très importants selon les missions de ministères, par exemple la gestion des offres des centres des repos pour les bénéficiaires

5.2. Spécification des besoins :

Pendant cette phase nous allons proposer les besoins en sécurité de l'infrastructure réseau ainsi de Cloud et ses services. Nous étions amenés à répondre aux questions suivantes :

- Quels services doit fournir notre Cloud ?
- Comment avoir un niveau de sécurité élevé pour notre Cloud ?

5.2.1. Infrastructure réseaux

En discutant avec le responsable informatique de la direction de la wilaya de Blida (Membre de projet informatique au niveau du ministère), on a opté à l'architecture réseau suivante :

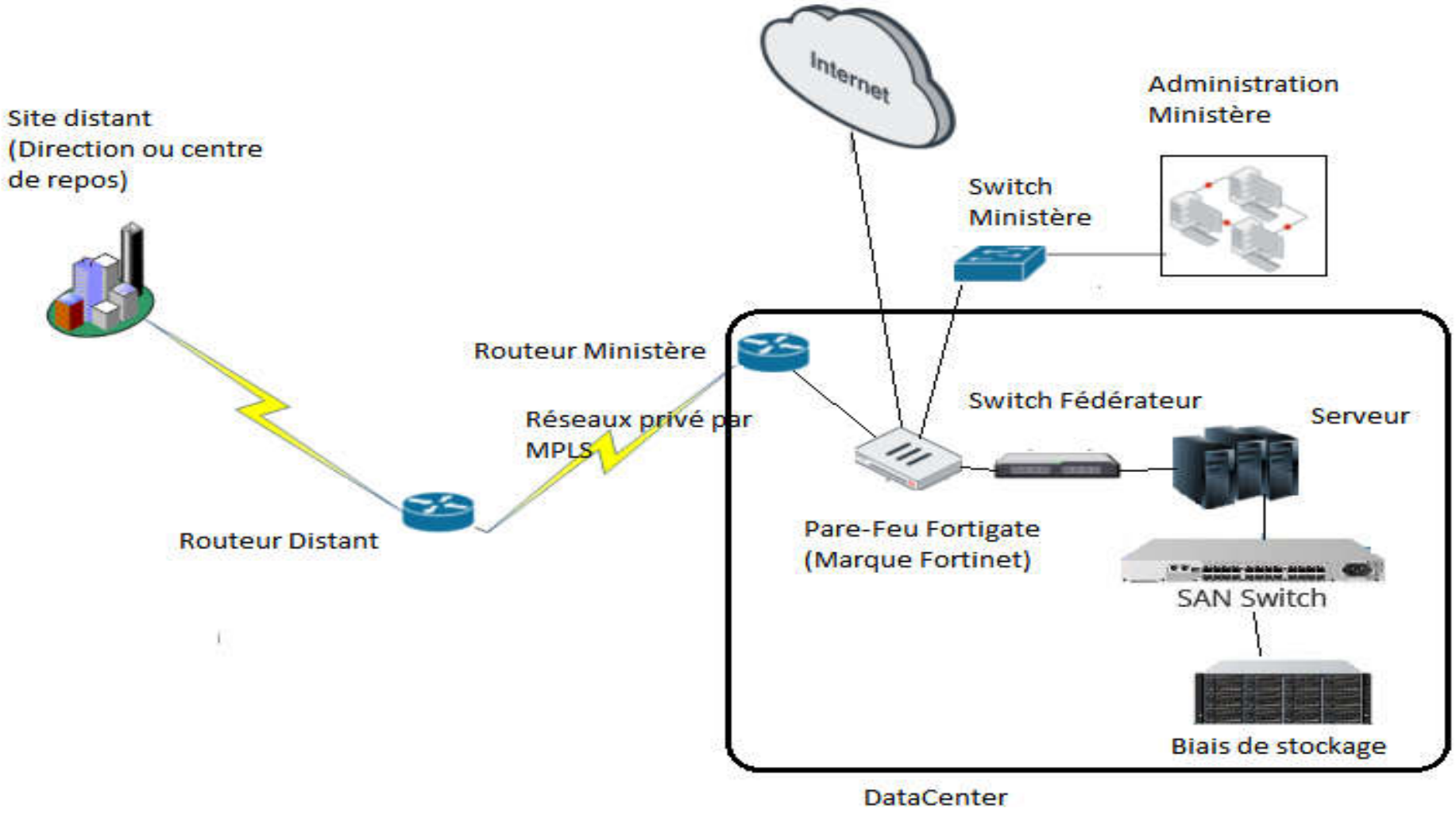

Figure 4.10. *Topologie physique du réseau proposé.*

Sachant qu'on garde l'ancienne architecture en ajoutant les changements suivants:

- Un réseau LAN doit être mis en place dans chaque centre de repos avec la même architecture an niveau des directions de wilaya.
- On garde le réseau intranet de l'entreprise.
- On ajoute un serveur dédié à notre système en créant une autre machine virtuelle dans les serveurs.
- Notre application doit être installée dans cette nouvelle machine virtuelle.
- On change les pare-feu C**isco** par un pare-feu **Fortigate** a cause ses puissantes caractéristiques.
- On ajoute un modem pour accès internet qui doit être relié avec le pare-feu

5.2.2. Sécurité :

Pour les mesures de sécurité, on propose :

- Achat d'une adresse IP fixe et un nom de domaine pour l'hébergement sécurisé de notre site.
- Création d'un autre VLan pour l'accès internet à notre service.
- Authentification mixte (système d'exploitation et SGBD) pour l'accès à notre service.
- Traitement des données se fait par un contrôle d'accès RBAC (Accès basé sur les rôles) pour les différentes opérations possibles.

6. Conclusion :

Nous avons donné dans ce chapitre une vue détaillée sur les démarches à suivre pour la modélisation de notre système. Notre solution se base sur trois volets : Un premier volet qui est la modélisation du système d'information étudié par la méthode UML à l'aide d'un ensemble d'étapes, de concepts et de graphes. Un deuxième volet où nous avons développé des modules qui composent les éléments de notre stratégie de contrôle d'accès. Cette dernière qui se base sur la méthode RBAC et s'interpose entre l'interface utilisateur et les services application. Un troisième volet qui représente la topologie physique du réseau de système.

Enfin, nous constatons que l'implémentation de cette modélisation est l'élément clé qui concrétise notre projet. Dans le chapitre suivant, nous expliquons les étapes de la réalisation de notre système, nous allons d'abord présenter les différents outils et technologies que nous avons utilisé, puis nous abordons l'implémentation de la solution proposée, et sa mise en œuvre.

Chapitre 05
Implémentation de la Solution

1. Introduction :

Dans cette partie, nous allons implémenter notre solution en commençant par la topologie physique de réseau, sa configuration et sa sécurité, en suite on passe à toutes les étapes à suivre pour rendre notre service accessible et sécurisé de l'intérieur (les différents utilisateurs internes) et de l'extérieur (Via internet).

2. La configuration de réseaux :

Comme vous avez constaté dans ***la figure 4.10,*** notre réseaux est très importants ***(48 direction, 15 centre de repos, l'administration du ministère et le Datacenter).*** L'adressage IP de réseaux sera réparti de la manière suivante :

A. Les directions et les centres de repos possèdent la même topologie de réseau avec un nombre différent des hôtes, l'adressage sera fait de la manière suivante :

Interface Ethernet	Adresse IP	Masque de réseau	Passerelle par default
Routeur lié avec le réseau interne de la structure	10.10.X.1	255.255.255.0	/
Hôte 1	10.10.X.2	255.255.255.0	10.10.X.1
Hôte 2	10.10.X.3	255.255.255.0	10.10.X.1
Selon le nombre des postes de la structure	…….	255.255.255.0	10.10.X.1

***Tableau 5.1.** Distribution des adresses IP sur des différents hôtes du système.*

- La valeur de X varie de 1 jusqu'à 48 pour les directions de wilaya, de 49 jusqu'à 63 pour les centres de repos et 64 pour le réseau interne de la direction.
- La carte WIC (Wan Interface Carte) de chaque routeur est reliée avec le matériel d'algérie télécom pour assurer la connexion WAN par la technologie MPLS. Sachant que le routeur de ministère est doté d'une carte HWIC (High

Wan Interface Carte) pour assurer la connexion avec tous les réseaux des structures externes.

- La direction dispose de quatre serveurs adressés 10.10.65.1, 10.10.65.2, 10.10.65.3, 10.10.65.4.
- On ajoute un cinquième serveur pour notre serveur d'application en lui attribuant l'adresse IP 10.10.66.1.

B. Pour l'hébergement interne de notre site on doit acheter un nom de domaine (Centre-Repos) et une adresse Ip fixe pour une communication sécurisé https avec notre site via internet.

C. N'autoriser l'accès externe via le port de pare-feu Fortigate que vert le serveur de l'adresse IP 10.10.66.1 par la redirection du flux.

3. La configuration des serveurs :

Pour la création des différents serveurs de notre Cloud et l'exploitation des différents services, on a suivi la stratégie suivant :

3.1. Système d'exploitation utilisé :

On a utilisé le système d'exploitation **Windows 2012 Serveur version DataCenter** (R2) à cause de ses puissantes caractéristiques qu'on les résume dans **[46] :**

- Plateforme idéale pour les Datacenters et Clouds privés.
- Une plateforme de virtualisation complète.
- La connexion aux services dans le Cloud.
- Simplification de l'administration.
- Un accès de n'importe où, sur n'importe quel terminal, aux environnements de travail virtualisés.

Cette version de Windows offre à l'utilisateur un ensemble des rôles, des fonctionnalités et des services qu'on les résume dans **[40]** :

- Accès à distance.
- Attestation d'intégrité de l'appareil.
- Expérience Windows Server Essentials.
- Hyper-V.
- MultiPoint Services, Serveur de télécopie.
- Serveur DHCP, Serveur DNS, Serveur Web (IIS).
- Service Guardian hôte.
- Services AD DS, Services AD LDS, Services AD RMS.
- Services Bureau à distance.
- Services d'activation en volume.
- Services d'impression et de numérisation de documents.
- Services de certificats Active Directory.
- Services de déploiement Windows.
- Services de fédération Active Directory (AD FS).
- Service de fichiers et de stockage.
- Service de stratégie et d'accès réseau.
- Service WSUS (Windows Server Update Services).

Le gestionnaire de serveur apparaitre automatiquement après le démarrage d'un serveur non configuré avec l'interface graphique suivant pour la configuration des rôles :

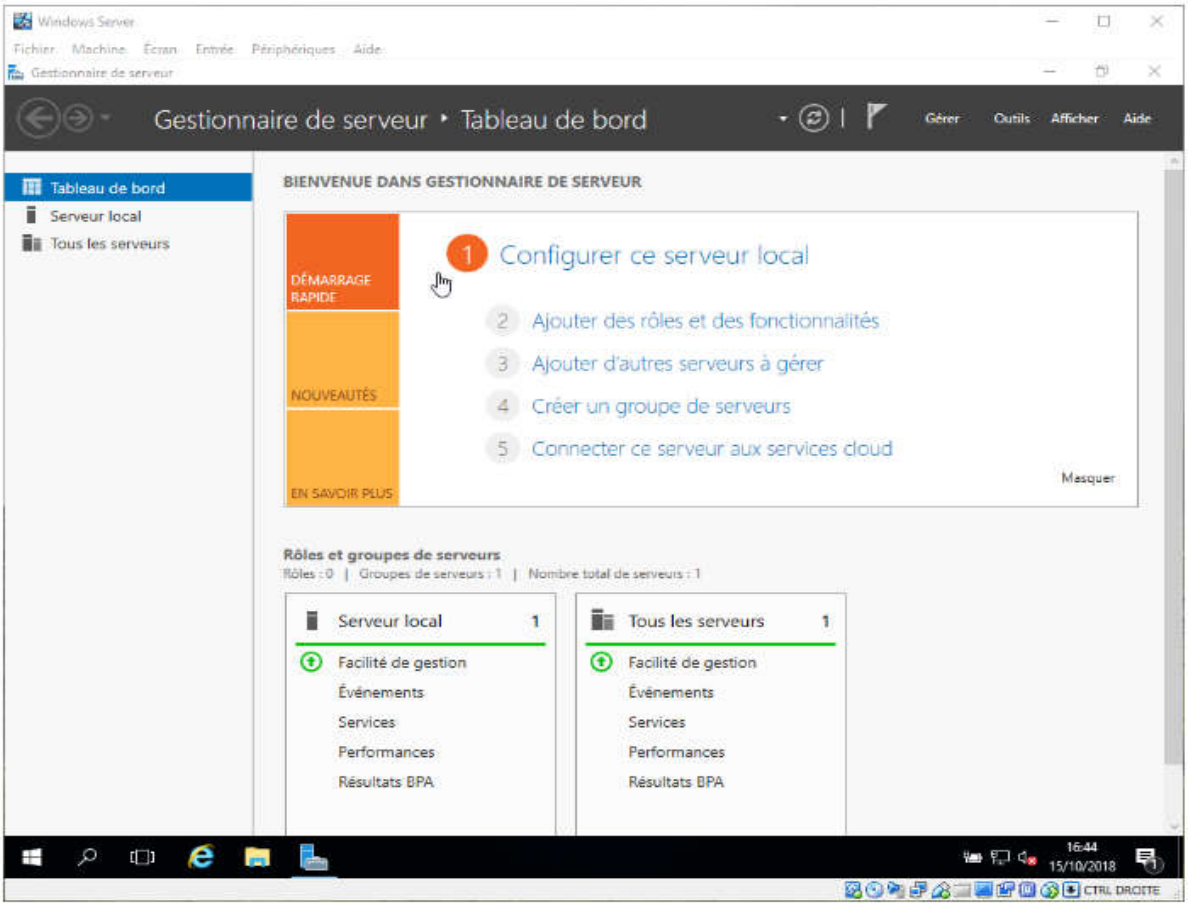

Figure 5.1. *Gestion de serveur Windows serveur 2012 R2.*

3.2. Création des machines virtuelles sur les serveurs :

3.2.1. Virtualisation :

3.2.1.1. Définition [48] :

La virtualisation est un mécanisme informatique qui consiste à faire fonctionner plusieurs systèmes, serveurs ou applications, sur un même serveur physique.

3.2.1.2. Terminologie [49] :

- ***Système hôte :*** *C'est la machine physique*
- ***Hyper-viseur*** *: C'est une plate forme qui permet l'abstraction de la couche matérielle de système hôte.*
- ***Système Invité ou machine virtuelle*** *: C'est le système d'exploitation qui s'exécute dans la machine virtuelle.*

3.2.1.3. Avantages [42] :

- Réduction de nombre des serveurs, de consommation énergétique et d'espace occupé dans les Datacenter.
- Amélioration de la flexibilité, rapidité et qualité des services.
- Utilisation des logiciels prévus pour divers environnements.

3.2.1.4. Type d'hyper-viseur [41] :

Il existe deux types d'hyper-viseurs :

- **l'hyper-viseur de type 1 ou *bare metal*** : il opère directement sur le *hardware*, et devient de ce fait l'outil de contrôle du système d'exploitation. Les OS invités s'exécutent alors par dessus ce hyperviseur.
  Exemples : **VSphere** de l'éditeur **VMware**, **KVM** pour **Linux,** Citrix XenServer et Microsoft Hyper-V (intégré à Windows Server).
- **l'hyper-viseur de type 2 ou *host metal*** : il fonctionne à l'intérieur d'un autre système d'exploitation.
  Exemples : **VirtualBox**, logiciel *Open Source* édité par Oracle, Microsoft Virtual Desktop.

La figure suivante schématise l'architecture des deux types :

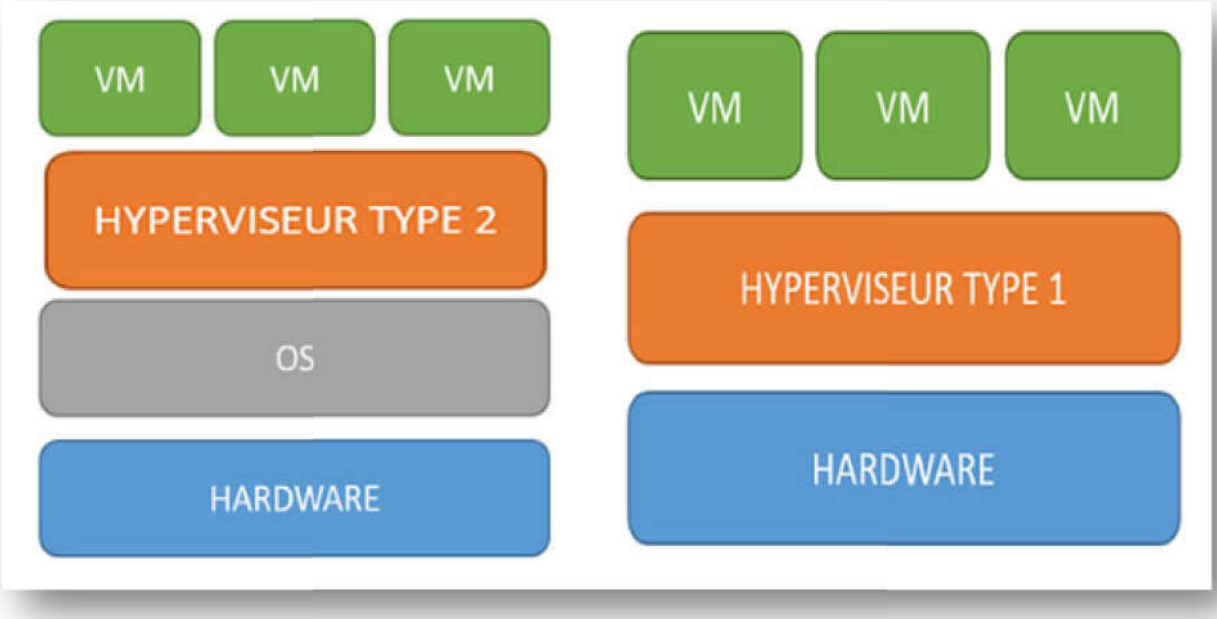

***Figure 5.2.** Type d'hyper-Viseur.*

3.2.2. Création des machines virtuelle en utilisant Hyper-V :

- Activer le rôle Hyper-V en utilisant le gestionnaire de serveur qui va lancer l'installation du hyper-V.
- L'installation de hyper-V entraine la modification suivante à notre serveur :

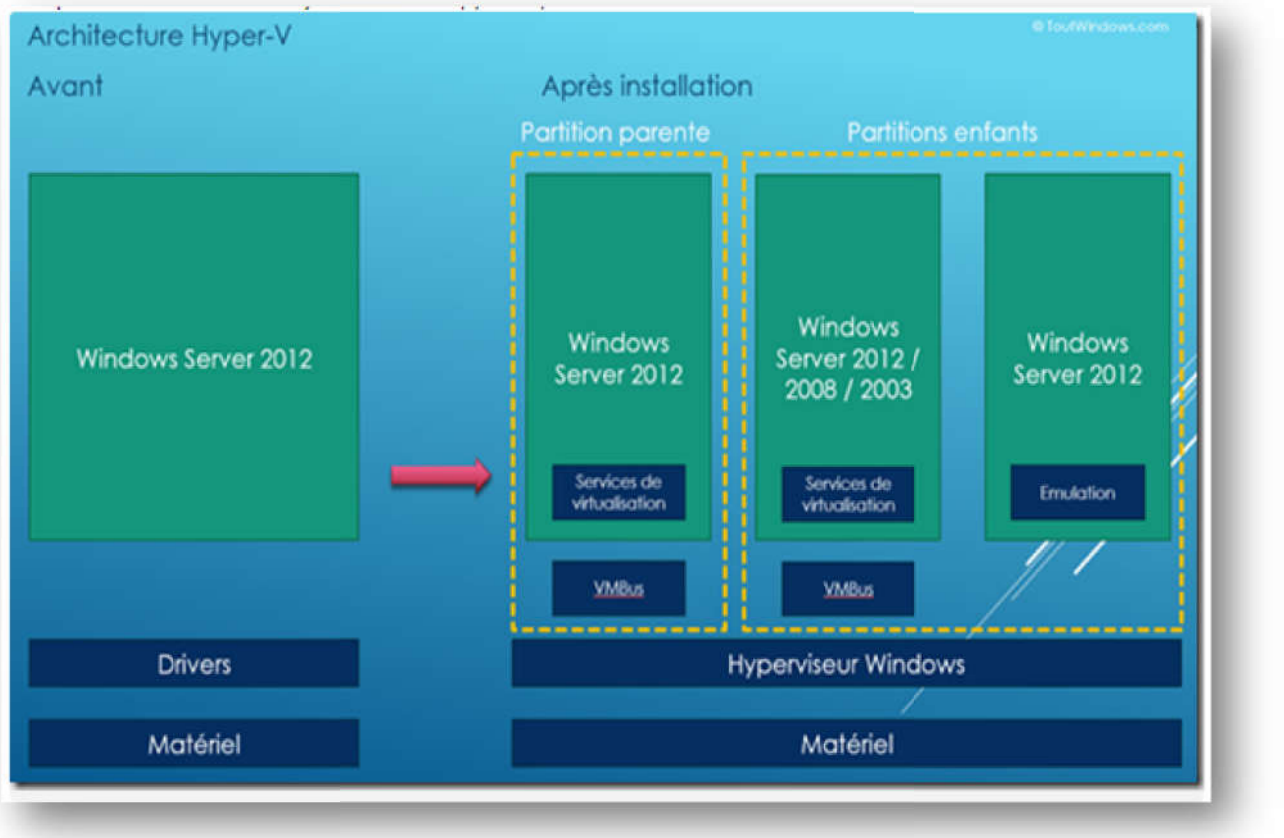

Figure 5.3. *Architecture d'un serveur virtualisé par hyper-V.*

4. Le serveur SGBD utilisé :

Notre service de Cloud s'applique sur un système d'information qui sera implémenté par SQL Server 2014 serveur (SGBD).

4.1. Pourquoi le SQL serveur :

Microsoft SQL Server 2014 est un SGBD relationnel destiné aux plateformes Windows. Il sert à élaborer, déployer et gérer des applications hébergées en local ou dans le Cloud **[43]**.

Les principales particularités de ce SGBD sont **[44]** :

- Multi-Base et Multi-Schéma.
- Gestion des schémas SQL.
- Déclencheurs ré-entrants (table "mutante")
- Vues indexées automatisées
- Table, index et procédure "In Memory"
- Intra jointure (APPLY).
- Stockage de fichiers électroniques : FileStream et FileTable.
- Réplication de données.
- Service d'envoi d'email intégré….etc.

4.2. Implémentation de la base de données :

Le développement de notre système se repose en premier lieu sur l'implémentation de la base de données à l'aide de SQL server 2014 en suivant les démarches suivantes :

- Lancement de Microsoft SQL Server Management Studio  connexion au serveur et le choix d'authentification.
- Création de la base de donnés **DataCenterRepos** constituée des classes pour la gestion de système d'information et d'autre classes pour gérer la politique d'accès. Dont les deux digrammes sont schématisés de la manière suivante :

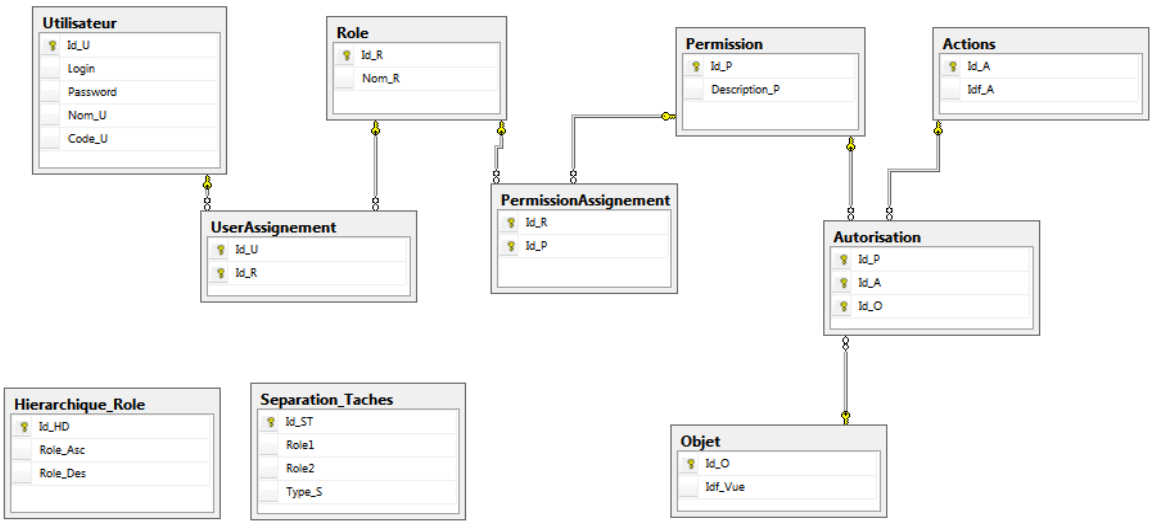

Figure 5.4. *Diagramme de base de données pour gérer la politique de contrôle d'accès.*

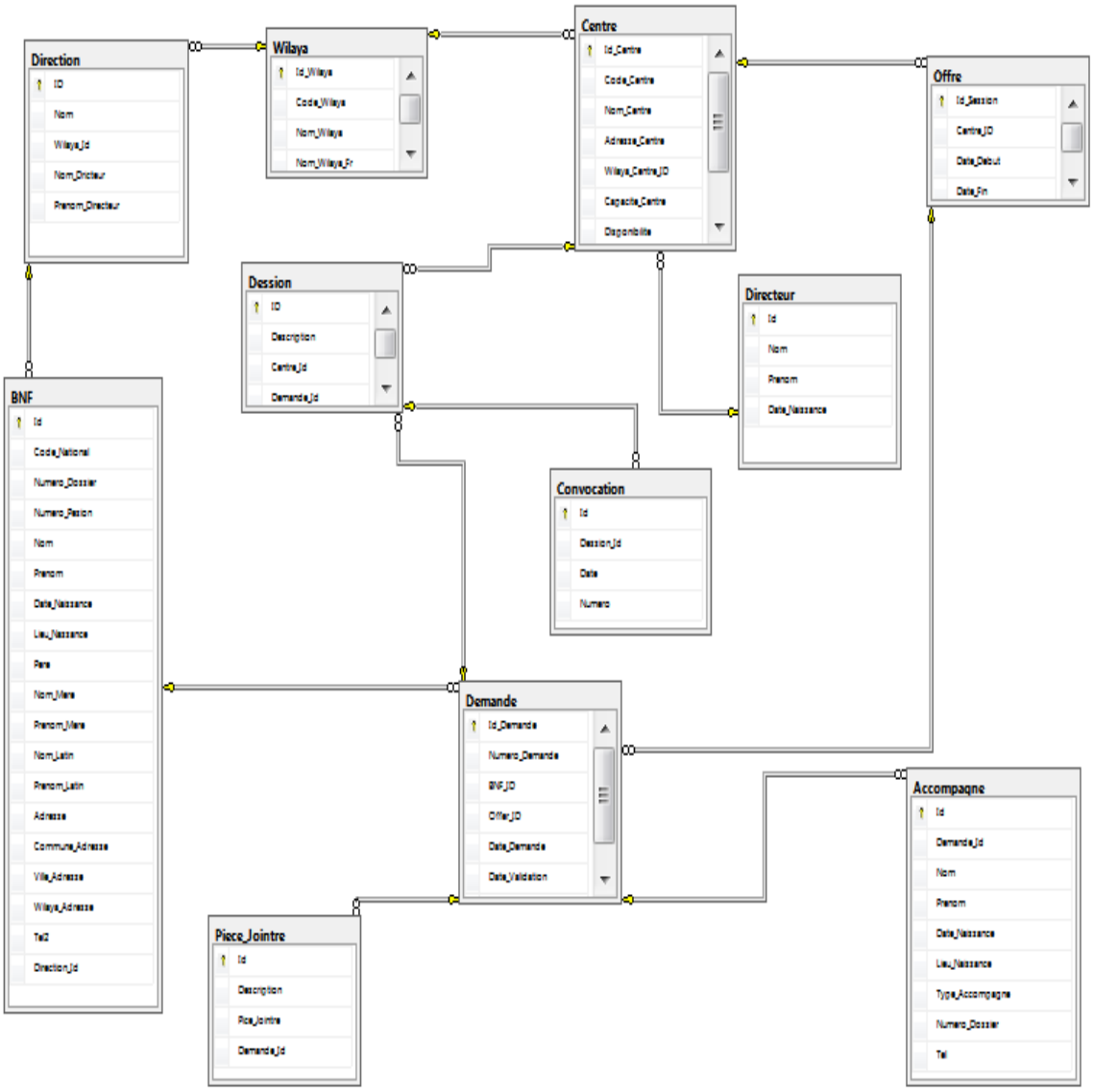

Figure 5.5. *Diagramme de base de données pour gérer le système d'information étudié.*

5. L'outil de développement utilisé :

Pour le développement de notre service, on a opté à utilisé ***Microsoft Visual Studio 2015 Professional.***

5.1. Pourquoi Microsoft Visual Studio 2015 Professional :

Microsoft Visual Studio 2015 est une suite d'outils permettant de créer des logiciels, qui couvre la phase de planification, la conception de l'interface utilisateur, par le codage, le test, le débogage, l'analyse de la qualité et de la performance du code, le déploiement sur les clients et la collecte de la télémétrie sur l'utilisation. Ces outils

sont conçus pour fonctionner ensemble avec la meilleure intégration possible via l'environnement de développement intégré (IDE) de Visual Studio **[45]**.

La figure suivante représente l'IDE de visual Studio :

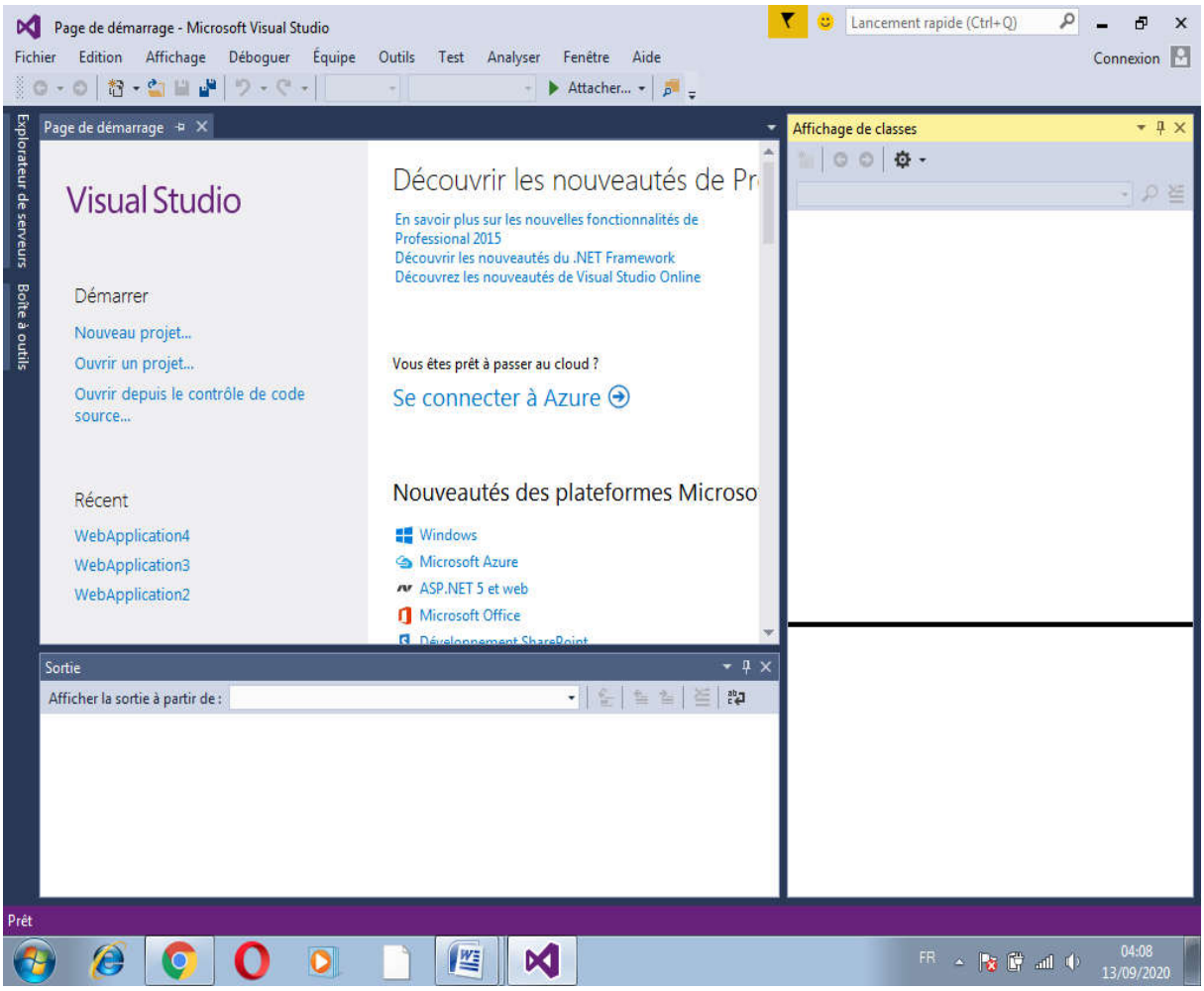

Figure 5.6*. L'environnement IDE de Visual Studio.*

Visual Studio permet de créer de nombreux types d'applications, que ce soit des applications commerciales simples et des jeux pour clients mobiles ou des grands systèmes complexes destinés aux entreprises et aux centres de données **[45]**.

Plus précisément, visual Studio permet de **[46]** :

- Créer Des applications et des jeux qui fonctionnent sur Windows, Android et IOS.
- Développez des applications et des jeux touchant tous les appareils fonctionnant sous Windows notamment la Xbox.

- Écrivez nos propres extensions pour Visual Studio.
- Créez, gérez et déployez facilement des applications à l'échelle du cloud sur Azure.
- Développez des applications et des services web modernes à l'aide d'outils libres performants basés sur ASP.NET, JQuery, AngularJS et d'autres Frameworks répandus.
- Développez et déployez facilement des bases de données SQL Server et Azure SQL.
- Des applications pour des plateformes et des appareils variés (Azure, Office365, SharePoint, Hololens, Kinect et Internet des objets), pour n'en citer que quelques-uns.

Visual Studio prend en charge par défaut C#, C et C++, JavaScript, F# et Visual Basic. Visual Studio fonctionne avec et s'intègre parfaitement aux applications tierces. Vous pouvez étendre Visual Studio vous-même en créant des outils personnalisés qui effectuent des tâches spécialisées.

5.2. Outils de développement utilisés pour la création de notre service :

Comme on a présenté précédemment, Microsoft Visual studio est un environnement très vaste de développement. Dans notre développement on a utilisé les outils suivants :

- **Web ASP.Net** : C'est un Framework de développement web créé par Microsoft et permettant de réaliser des sites web complexes grâce au modèle **MVC** (*Model View Controller*) **[52]**.
- **Visual C # (C-Sharp)** : C # est un langage de programmation orienté objet développé par Microsoft qui s'exécute sur le .NET Framework. Il est utilisé pour développer des applications Web, des applications de bureau, des applications mobiles, des jeux et bien plus encore **[49]**.
- **MVC [54]** : Est d'abord un sigle qui signifie **Modèle Vue Contrôleur**. Avec le MVC, nous allons donc séparer notre programme en trois parties comme ceci :

- **Le modèle** : c'est ce que fait l'application. C'est la logique, le cerveau de l'application.
- **Le contrôleur** : il récupère les informations du modèle et les affiche dans la vue.
- **La vue** : c'est ce que l'utilisateur voit, c'est l'interface de l'application.

- **Entity Framework :** c'est la solution de mapping objet-relationnel (ORM[5]) proposée par Microsoft. Son but est de fournir la couche d'abstraction nécessaire aux développeurs pour qu'ils n'accèdent plus directement à la base de données, mais par l'intermédiaire d'entités définies par un modèle appelé EDM (Entity Data Model) **[50]**.
- **LINQ** : c'est l'acronyme de **L**anguage-**I**ntegrated **Q**uery est le nom d'un ensemble de technologies basé sur l'intégration des capacités de requête directement dans le langage C # **[48]**.
- **LINQ To SQL** : est une framework dans Visual Studio à partir de 2010, elle est est utilisé pour créer automatiquement les modèles fortement typés avec les classes .net basées sur les tables de données SQL des bases de données référencées **[53]**.
- **Transact -SQL** : Microsoft Transact SQL ou T-SQL est un langage de requêtes amélioré par rapport au SQL dont il reprend les bases. Le SQL (Structured Query Language) est le langage standard, créé par IBM dans les années 70, pour la gestion des SGBDR (Systèmes de Gestion de Bases de Données Relationnelles).
  De plus, le Transact SQL prend en compte des fonctionnalités procédurales telles que la gestion des variables, les structures de contrôle de flux, les curseurs, et les lots d'instructions. C'est donc un langage complet qui comporte des instructions, qui manipule des objets SQL, qui admet la programmabilité et qui utilise des expressions.

[5] **ORM : c'est l'acronyme de Object-Relational Mapping**. Il s'agit d'une technique de programmation informatique qui permet de simplifier l'accès à une base de données en proposant à l'informaticien des « objets » plutôt que d'accéder directement à des données relationnelles. **[51]**

5.3. Démarche de développement de notre service :

Etape 01 : Création d'un nouveau projet web Visual Studio

- Lancement de Visual Studio 2015.

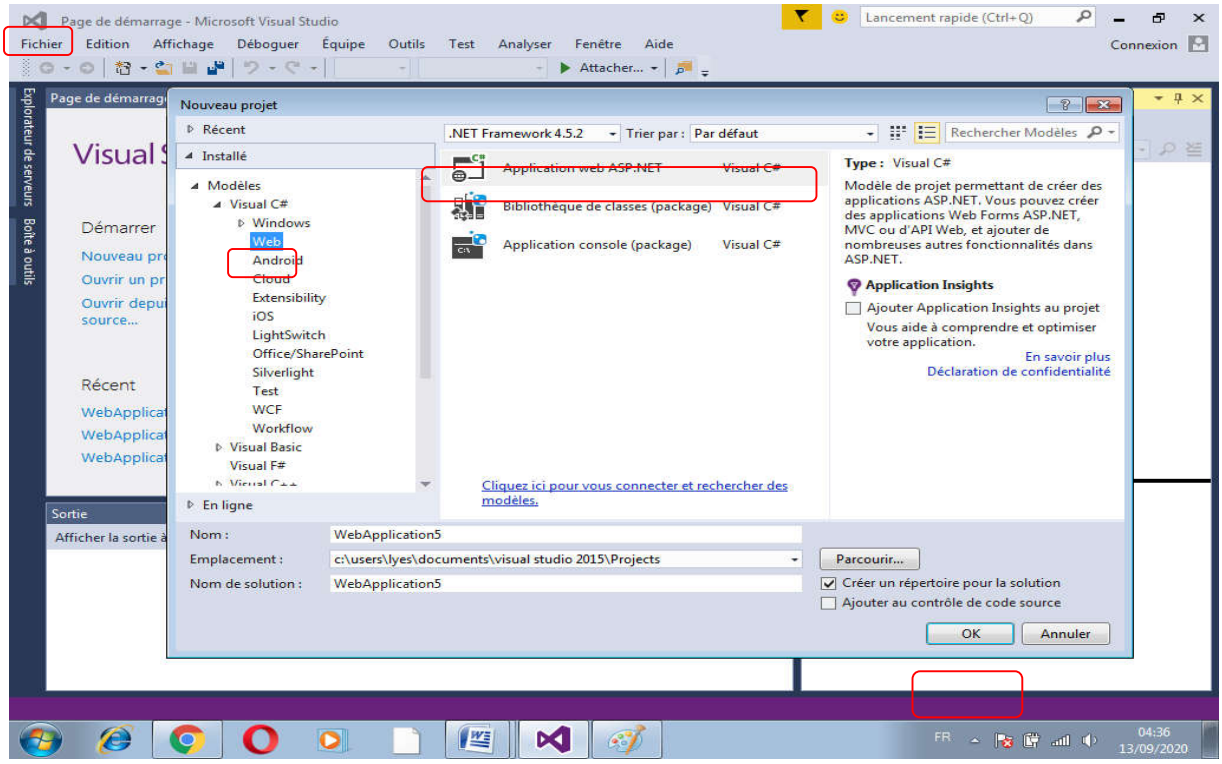

Figure 5.7. *Création de projet et choix de type d'application*

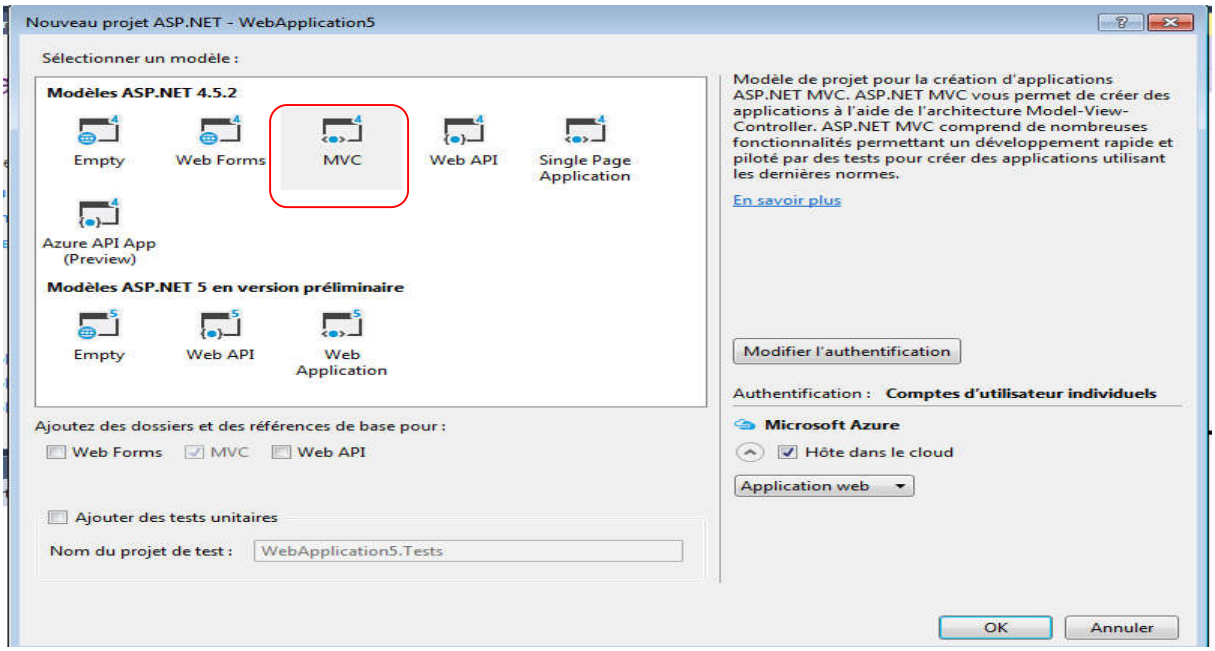

Figure 5.8. *Choix de modèle de conception*

Etape 02 : Connexion à la base de données

Après la création de projet, on doit le connecter à la base de données en respectant les étapes suivantes :

- Afficher **l'explorateur de serveur** ou cliquer sur le menu **Outils**

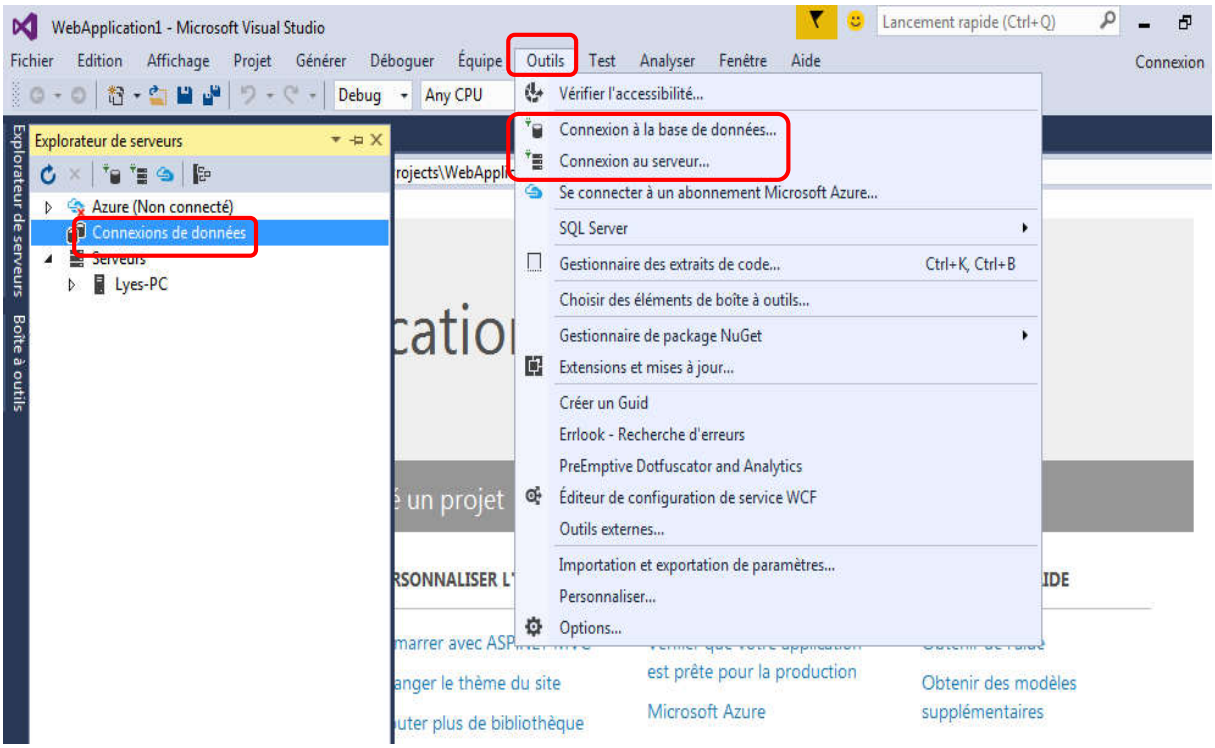

Figure 5.9. *Choix de serveur.*

- Choisir **Connexion au serveur** dans le cas d'un serveur distant en précisant l'adresse IP de serveur ou le nom de domaine.
- Choisir **Connexion à la base de données** pour choisir une BDD locale :
- Une fois la base de données est connectée avec notre application, on passe la conception de l'interface graphique et les contrôles nécessaires.

6. Interface graphique et le test du fonctionnement de service proposé :

Dans cette partie, on propose les pages web de quelques traitements de notre système. C'est un test fait en local en attendant l'hébergement officiel du site :

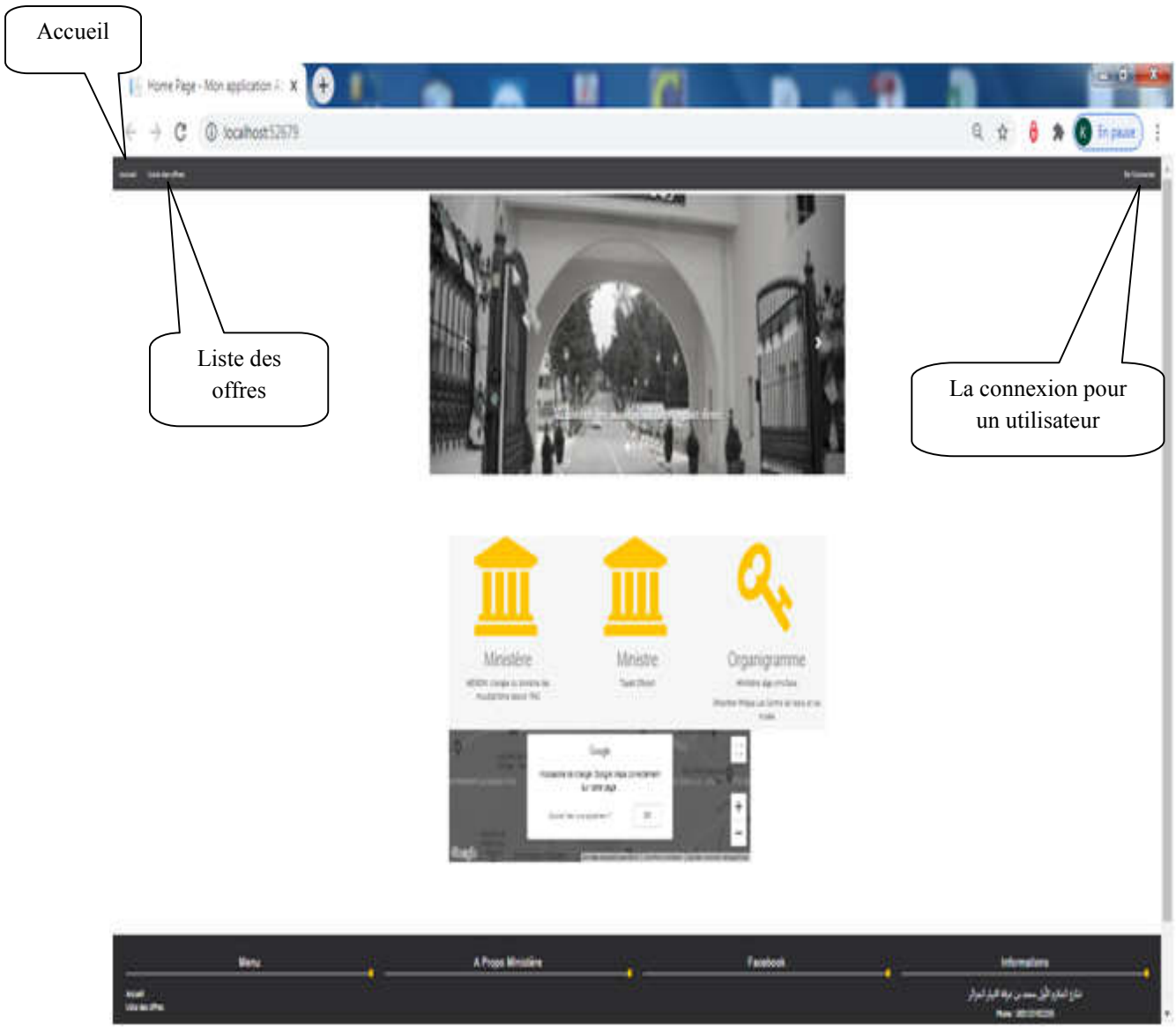

Figure 5.10. *Page d'accueil.*

Cette page, est la page d'accueil de notre système, elle sera affichée en tapant le nom de domaine ou l'adresse ip fixe après une recherche dans le web. Elle nous offre l'accées à notre session ou afficher uniquement la liste des offres.

Après une demande de connexion, une page d'authenfication est t'affichée dont sa structure est la suivante :

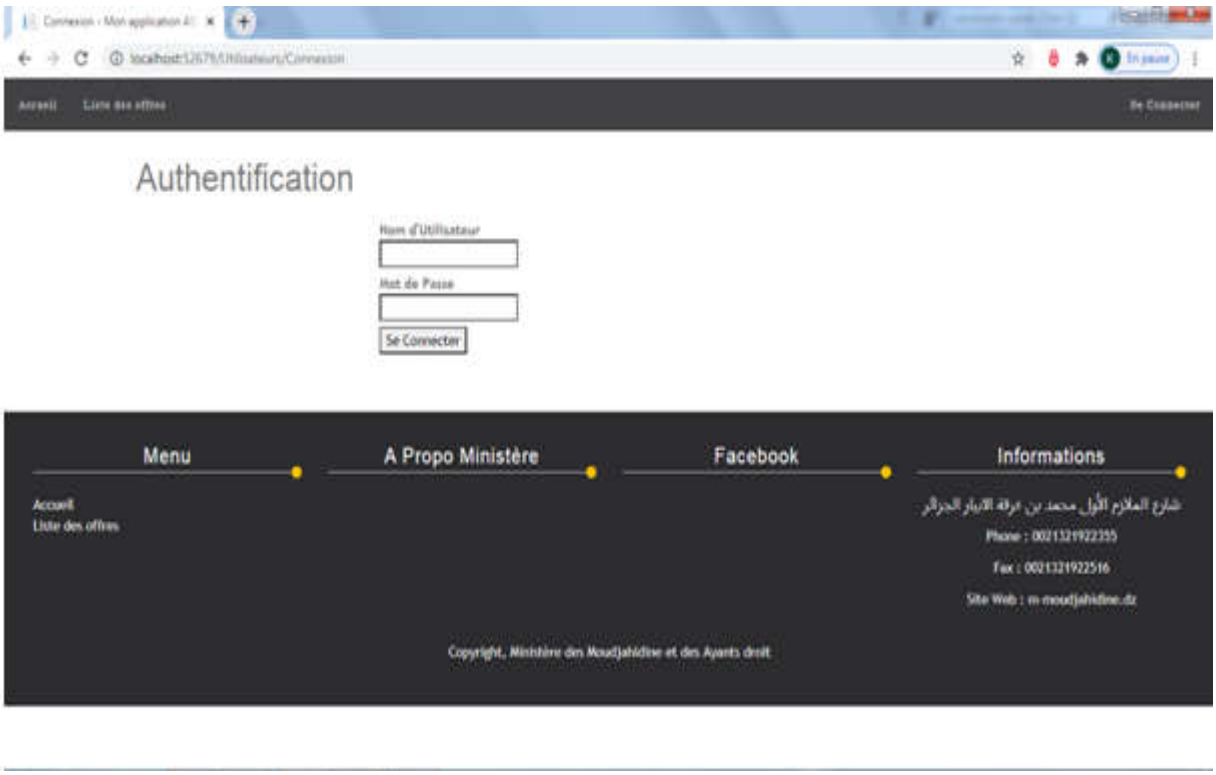

Figure 5.11. *Page d'authentification.*

Après la saisie d'un nom utilisateur et d'un mot de passe, on permet à l'utilisateur d'accéder à son compte selon ses droits d'accès. S'il y a une erreur dans l'authentification, on affiche :

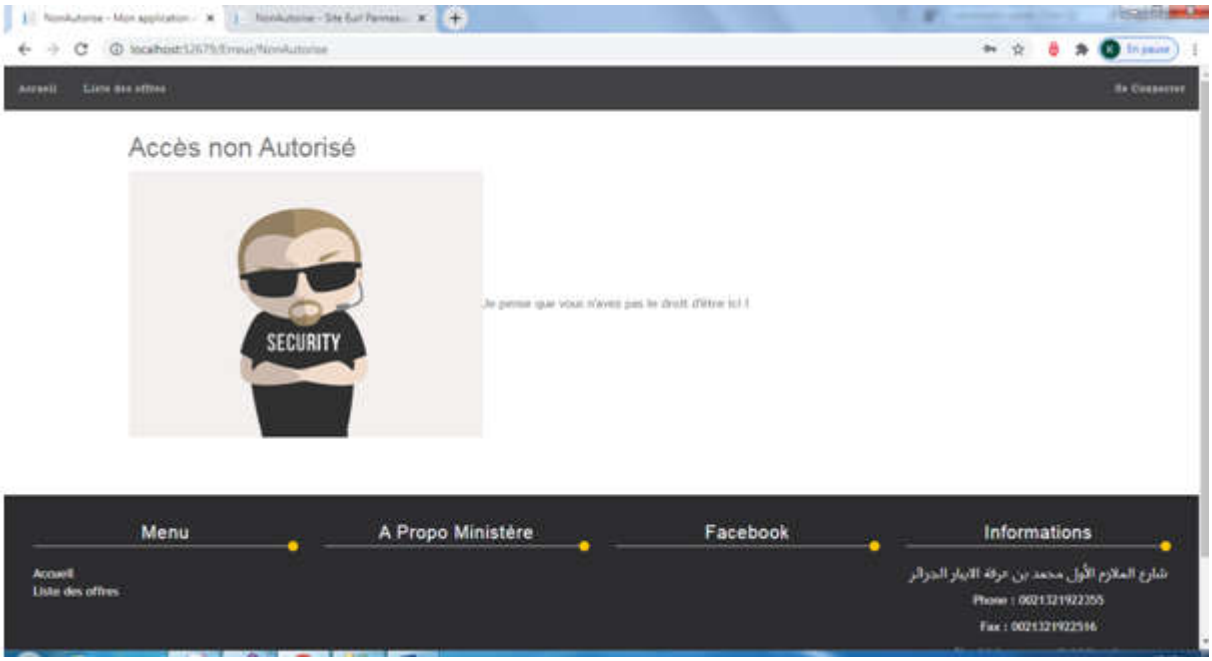

Figure 5.12. *Page d'accès non autorisé.*

Pour la consultation de la liste des offres, on affiche la page suivante :

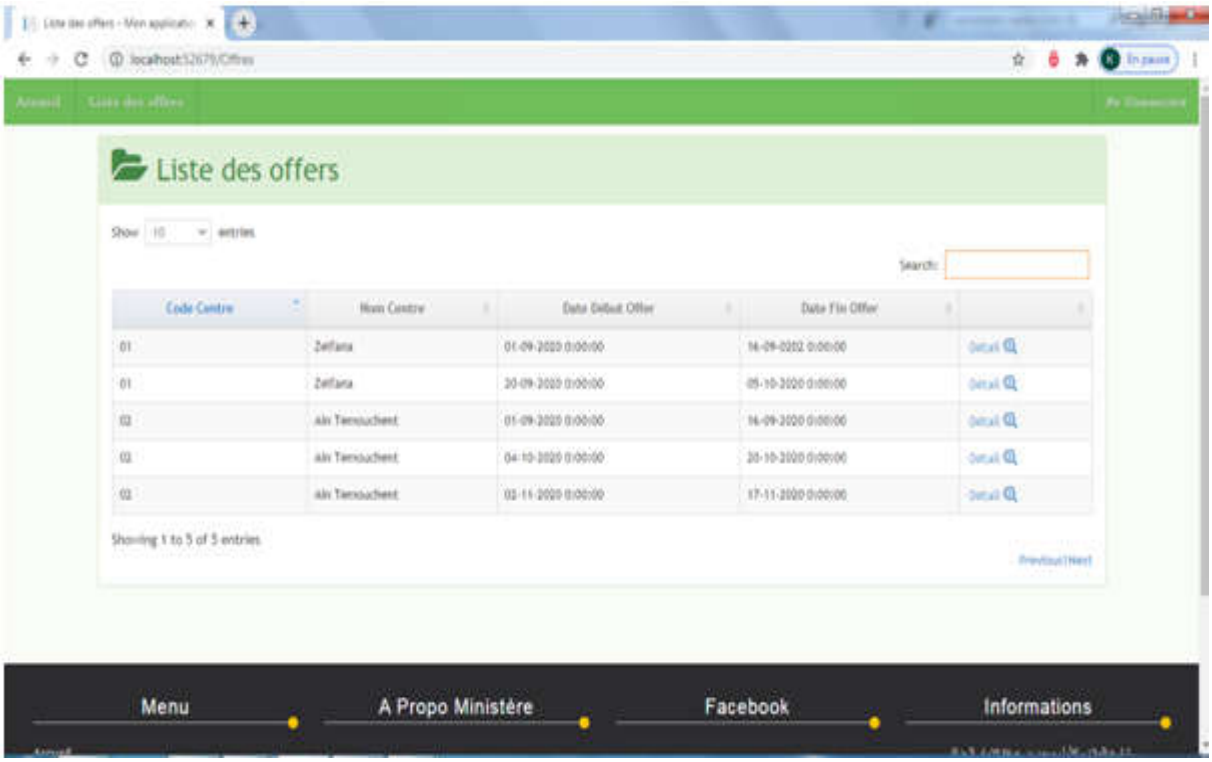

Figure 5.13. *La liste des offres.*

Par la suite, un bénéficiaire puisse saisir sa demande pour l'inscription dans une offre à partir de formulaire suivante (sa phase d'authentification sa passe à ce niveau :

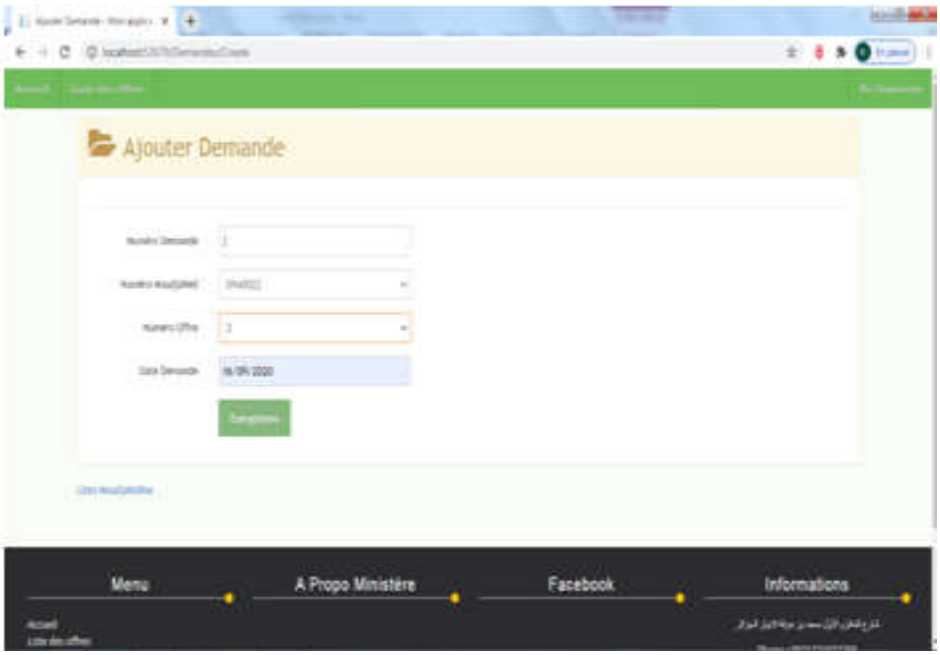

Figure 5.14. *La saisie d'une demande.*

La connexion des utilisateurs est assurée par le script suivant :

```
[HttpPost]
publicActionResult Connexion(string ReturnUrl, LoginModel model)
        {
Data_Centre_ReposEntities db = newData_Centre_ReposEntities();
try
            {
var dataItem = db.Utilisateurs.Where(x => x.Login == model.NomUtilisateur &&
x.Password== model.MotDePasse).First();
if (dataItem != null)
                {

FormsAuthentication.SetAuthCookie(dataItem.Login, false);
if (Url.IsLocalUrl(ReturnUrl) && ReturnUrl.Length > 1 && ReturnUrl.StartsWith("/") &&
                        !ReturnUrl.StartsWith("//") && !ReturnUrl.StartsWith("/\\"))
                    {
return Redirect(ReturnUrl);
                    }
else
                    {

return RedirectToAction("Index", "Home");

                    }
                }
else
                {

return RedirectToAction("NonAutorise", "Erreur");

                }
            }
catch (Exception e)
            {
                ModelState.AddModelError("", "invalide user mots passe ");
return View();
}


              }
```

Pour afficher le nom d'utilisateur dans la page, on a créer une vue partielle à l'aide de script suivant :

```
@if (User.Identity.IsAuthenticated)
{
<span>@User.Identity.Name            | </span>@Html.ActionLink("Déconnexion",
"DeConnexion", "Utilisateurs")
}
else
{
@Html.ActionLink("Se Connecter", "Connexion", "Utilisateurs")

}
```

Pour l'utilisateur Administrateur Principale, sa vue est la suivante :

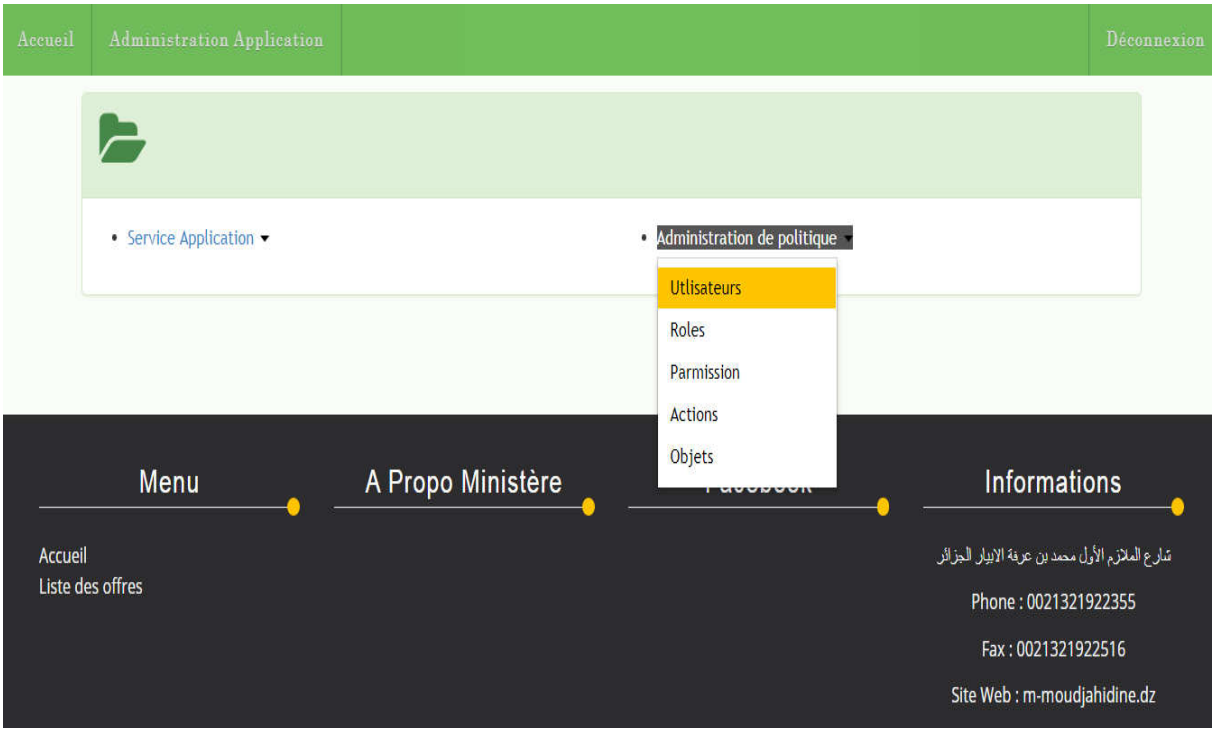

Figure 5.15. *Espace Administrateur Principal.*

Pour le traitement des utilisateurs, on affiche la page suivante :

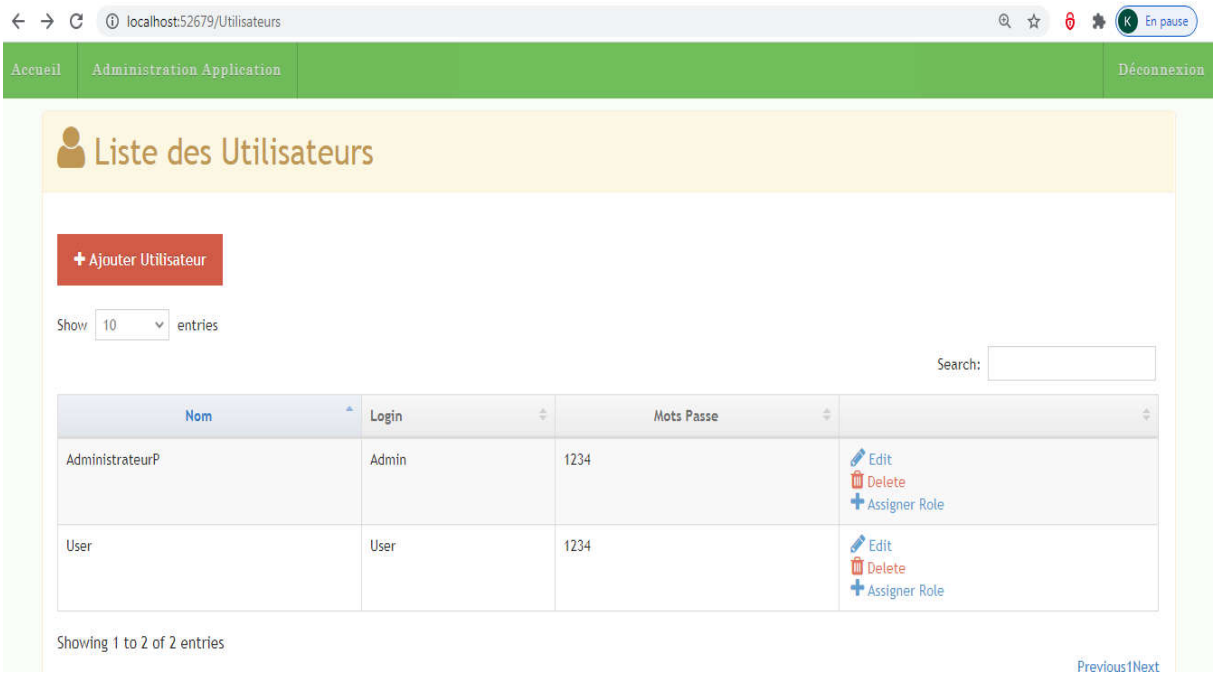

Figure 5.16. *Traitement des utilisateurs.*

Les procédures stockées les plus importantes dans le contrôles d'accès crées par le Trans-SQL sont :

Insertion d'un rôle :

```
Create procedure Inser_Role (@Nom_R nchar(30)) as
declare @i int;
begin
if exists (select * from [role] where Nom_R=@Nom_R)
begin
print('Ce rôle existe Déjà !!');
return;
end;
exec sp_addrole @Nom_R
select @i=count(*)from role
insert into [role](Id_R,Nom_R)values (@i+1,@Nom_R);
end;
```

Assignation d'un utilisateur à un rôle :

```
Create procedure Assig_User_Role(@Log_U nvarchar(30),@Nom_R
nchar(30)) as
declare @Id_R int;
declare @Id_U int;
declare @Idi_u int;
begin
if not exists (select * from utilisateur where [Login]=@Log_U)
begin
print('Cet utilisateur n''existe pas!!');
return;
end;
if not exists (select * from [Role] where Nom_R=@Nom_R)
begin
print('Cet role n''existe pas!!');
return;
end;
set @Idi_u=(select DP.principal_id From
sys.database_principals  DP where DP.name=@Log_U)
if exists( select DP.name
                From sys.database_principals DP
                INNER JOIN sys.database_role_members DRM
                ON DRM.role_principal_id = DP.principal_id
                WHERE (DP.type = 'R') and
(DRM.member_principal_id=@Idi_u)and(DP.name=@nom_R))
          begin
          print('utilisateur déja assig au role');
          return;
          end;
exec sp_addrolemember  @Nom_R , @log_U;
select @Id_U=utilisateur.Id_U from utilisateur where
[login]=@Log_U
```

```
select @Id_R=[Role].Id_R from [Role] where Nom_R=@nom_R
insert into UserAssignement(Id_U,Id_R)values(@Id_U,@Id_R)
end;
```

Suppression d'une assignation d'une permission à un rôle :

```
Create procedure Suppr_Assig_Perm_Role(@Desc_P
nvarchar(30),@Nom_R nchar(30) ) as
declare @Id_R int;
declare @Id_P int;
begin
if not exists (select * from Permission where Desc_P=@Desc_P)
begin
print('Cette permission est invalide !!');
return;
end;
if not exists (select * from [Role] where Nom_R=@Nom_R)
begin
print('Cet role n''existe pas!!');
return;
end;
exec sp_droprolemember  @Nom_R , @Desc_P;
select @Id_P = Permission.Id_P from Permission where
Desc_P=@Desc_P
select @Id_R=[Role].Id_R from [Role] where Nom_R=@nom_R
delete from PermissionAssignement where
(Id_R=@Id_R)and(Id_P=@Id_P)
end;
end;
```

7. Conclusion :

Ce chapitre fait l'objet d'une démarche structurée pour aboutir à un Cloud privé fiable pour notre organisme. Ainsi que le choix des outils choisis se base sur les critères de l'extensibilité, l'externalisation et de la sureté.

En premier lieu, nous avons mis en place l'architecture réseaux et sa configuration. En second lieu nous avons déployé les outils utilisés pour augmenter le niveau de sécurité de notre architecture, ensuite, nous avons présenté les outils de développement pour un accès sécurisé à notre système en implémentant une méthode de contrôles d'accès basée sur la méthode RBAC. Nous terminons le chapitre par la présentation de quelque interface pour la validation dans nos tests et conclusion.

Conclusion Générale et Perspectives

Conclusion générale

Dans ce mémoire, nous avons posé la problématique de la sécurité dans le Cloud Computing privé. En particulier, nous nous sommes focalisés sur les politiques de contrôle d'accès pour assurer la sécurité des données. Afin d'atteindre cet objectif, nous avons au cours de l'étude bibliographique montré les notions théoriques principales entourant le sujet en commençant par des notions de base sur la sécurité informatique. Ensuite nous avons entamé l'étude des différents modèles de contrôle d'accès existants et plus spécifiquement le modèle RBAC (Role Based Access Control) basés sur les rôles. Ce dernier largement adopté par les entreprises et les industriels a été appliqué dans de grandes structures.

En s'appuyant sur le modèle de contrôle d'accès RBAC, le système proposé est doté d'un module d'authentification pour gérer les utilisateurs, un module de vérification d'autorisation et un module d'administration de la politique d'accès.

Dans l'étape de la modélisation du système d'information nous avons choisi le langage UML. Passant par plusieurs étapes en incluant des graphes, des scénarios et des concepts jusqu' à l'obtention du diagramme de classe du système.

Pour l'implémentation du système nous avons proposé une architecture physique de réseau qui va soutenir notre solution, ainsi le Cloud Computing privé.

La conception de notre système de contrôle d'accès a été validée par les implémentations présentées et les tests effectués. Espérant que notre contribution va ajouter un plus dans le domaine de la sécurité du Cloud Computing.

Tout travail de développement et de recherche n'est en réalité qu'une ouverture sur de futurs travaux susceptibles à être améliorés et enrichis, afin de réaliser un système plus performant. Enfin nous terminons le mémoire par l' illustration de quelques perspectives, espérant q' elles enrichissaient notre système:

- Nous proposons d' ajouter l' aspect contextuel au modèle de contrôle d' accès (la notion de localisation et d'information spatiale introduites par le modèle LRBAC ou la notion de temps et des contraintes de temps introduite par le modèle GTRBAC).

- Proposer un système d'aide à la résolution en cas de vulnérabilité éventuelle du système. Cette solution se base sur des contraintes et des situations définies préalablement. Le résultat sera des propositions de résolution avec un pourcentage de réussite pour chaque proposition.

Références Bibliographiques

[1]. GHERNAOUTI Solange, *« **Sécurité informatique et réseaux** »,* Dunod 4éme édition, Livre, Année 2013.

[2]. BLOCH Laurent ***et*** WOLFHUGEL Christophe, *« **Sécurité informatique principes et méthodes** »,* Groupe Eyrolles 3 éme édition, Livre, Année 2011.

[3]. GABAY Joseph, GABAY David *« **UML 2 Analyse Guidée Mise en oeuvre guidée avec études de cas**»,* Edition DUNOD, Paris, 2008, ISBN 978-2-10-053567-5.

[4]. MAHDAOUI Latifa, ABDAT Nadia *« **Pratique des systèmes d'information avec UML**»,* Edition Pages Bleues, Alger, 2007, ISBN 978-9947-850-01-5.

[5]. Yasmina, GHEBGHOUB. « ***La modélisation des aspects de sécurité dans le Cloud*** ». : Faculté des Sciences Département d'Informatique. Informatique décisionnel, Universite Saad Dahleb – Blida 1, Thèse de doctorat 2017.

[6]. REKIA Sidahmed, *« **Elaboration et mise en place d'une stratégie de sécurité et de conformité pour contrôler l'accès aux réseaux** »,* Université SAAD DAHLEB BLIDA1, Mémoire de master, Année 2019.

[7]. BOUCHERBA Khadidja ***et*** ZIANE Saloua, *« **Mise en place d'un pare-feu d'entreprise open source PfSense**»,* Université Abderrahmane MIRA de Béjaïa, Mémoire de master professionnel, Année 2015.

[8]. LONE SANG Fernand, *« **Protection des systèmes informatiques contre les attaques par entrèes-sorties**»,* Université Toulouse, Thèse de Doctorat, Année 2012.

[9]. MESSOUAF Sonia, *« **Génération automatique des scénarios d'attaques dans les systèmes informatiques**»,* Université Abderrahmane MIRA de Béjaïa, Mémoire de master professionnel, Année 2013.

[10]. LOKBANI Ahmed Chaouki, *« **Le problème de sécurité par le DataMining**»,* Université DJILLALI ELYABES SIDI BEL ABBES, Thèse de Doctorat, Année 2017.

[11]. BERKANI Nassima ***et*** MOUSSAOUI Salima Saloua, *« **La sécurité des données dans le Cloud Computing**»,* Université Abderrahmane MIRA de Béjaïa, Mémoire de master professionnel, Année 2016

[12]. HAMZ Mohamad, *« **Autonomie, sécurité et QoS de bout en bout dans un environnement de Cloud Computing**»,* Université Bourgogne, Thèse de Doctorat, Année 2015

[13]. KABOU Salaheddine, *« **La gestion de la confidentialité dans le Cloud Computing**»,* Université Djillali Liabes - Sidi Bel Abbes, Thèse de Doctorat, Année 2017

[14]. KARTIT Zaid, *« **Contribution à la sécurité du Cloud Computing : Application des algorithmes de chiffrement pour sécuriser les données dans le Cloud Storage**»,* Université Mohamed V -Rabat-, Thèse de Doctorat, Année 2016.

[15]. YAGOUB Mohamed Amine, *« **Une approche basée agent pour la sécurité dans le Cloud Computing**»,* Université Mohamed Khider - Biskra, Thèse de Doctorat, Année 2019.

[16]. RELAZA Théodore Jean Richard, *« **Sécurité et disponibilité des données stockées dans les nuages**»,* Université Paul Sabatier - Toulouse, Thèse de Doctorat, Année 2016.

[17]. PROBST Thibaut, *« **évaluation et analyse des mécanismes de sécurité des réseaux dans les infrastructures virtuelles de Cloud Computing**»,* Institut National Polytechnique de Toulouse (INP Toulouse), Thèse de Doctorat, Année 2015.

[18]. Marwan, CHEAITO. ***«Un cadre de spécification et de déploiement de politiques d'autorisation***», Toulouse III : Ecole Doctorale EDMITT : Mathématiques Informatique Télécommunications Toulouse, Thèse de doctorat : Année 2012.

[19]. AOUAG Mouna, *« **Des diagrammes UML 2.0 vers les diagrammes orientés aspect à l'aide de transformation de graphes**»,* Université Constantine 2, Thèse de Doctorat, Année 2014.

[20]. HAOUES Messaouda, BACHA Saada *« **Application de la Méthode DVFS Dans l'Exploitation du Base de Données**»,* Université Akli Moand Oulhadje-Bouira-, Mémoire de master, Année 2018.

[21]. Abou El Kalam et al. « ***Organization Based Access Control*** », IEEE 4th International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Come, Italy, 2003.

[22]. R. Thion, « ***Structuration Relationnelle des politiques de contrôle d'accès Représentation, Raisonnement et Vérification***, ». Thèse de doctorat. Institut National des Sciences Appliquées de Lyon, Année 2008**.**

[23]. Michel, Embe Jiague . *« **Mise en œuvre de politiques de contrôle d'accès formelles pour des applications basées sur une architecture orientée services*** ». Département d'informatique. Faculté des sciences, université de sherbrooke. these de doctorat, année *2012.*

[24]. A. K. Dey, « ***Understanding and Using Context. Personal Ubiquitous Computing »,*** *vol. 5, no. 1, pages 4–7, Springer-Verlag, London, 2001. 41.*

[25] . Article par *:* Insight acquiert : l'entreprise de conseil numérique française*, Aout* 2019-source : https://fr.insight.com/fr/content-and-resources/articles/2013-04-les-5-defis-du-cloud-a-relever,

[26]. J. McCumber**, *«Information Systems Security: A Comprehensive Model*»** in: Proceeding of the 14th National Computer Security Conference, NIST, Baltimore, MD, 1991.P 330.

[27]. Alban Gabillon. « ***Contrôler les accès aux données numériques***. » *La Revue de* l'Electricité et de l'Electronique, Société de l'Électricité, de l'Électronique et des Technologies de l'Information et de la Communication, 2013, 12 p. ffhal-02108021ff.

[28]. Paul Tran Van. « ***Partage de documents sécurisés dans le Cloud Personnel***. » Réseaux et télécommunications [cs.NI]. Université Paris-Saclay, 2018. Français. ffNNT : 2018SACLV015ff. fftel-01779315ff.

[29]. Mathieu Blanc. « ***Sécurité des systèmes d'exploitation répartis : architecture décentralisée de métapolitique pour l'administration du contrôle d'accès obligatoire*** ». Réseaux et télécommunications [cs.NI]. Université d'Orléans, 2006. Français. fftel-00460610.

[30]. Cédric, Brancourt, « ***Le contrôle de droit d'accès et la sécurité de vos systèmes », 2015. Source*** **:** https://www.synbioz.com/blog/tech/autorisation-et-droits-d-acces (Consulté le 15/06/2020).

[31]. Odile, PAPINI. « ***Contrôle d'accès : Cours 04*** ». ESIL. Université de la méditerranée. ***Source :*** http://odile.papini.perso.esil.univmed.fr/ sources/ SSI.htm (consulté le 14/03/2020).

[32]. Les menaces informatiques, ***Source : https://www.cours-gratuit.com/cours-informatique/ cours-sur-les-menaces-informatiques*** **(consulté le 10/03/2020).**

[33].« *Sécurité informatique* »,

Source:https://www.nbs-system.com/blog/introduction-a-la-securite-informatique/ (consulté le 15/03/2020).

[34]. « *Vulnérabilités* »,

Source *:* https://repo.zenk-security.com/Techniques% 20d.attaque s%20%20. %20%20Failles /Vulnerabilites.pdf (consulté le 16/03/2020).

[35]. « *Vulnérabilités : de quoi parle-t-on ?* »,

Source :https://cyberdefense.orange.com /fr/blog /vulnérabilités-de-quoi-parle-t-on/ (consulté le 16/03/2020).

[36]. «*Approche Décentralisée pour la sécurité d'un Réseau de Capteurs Sans Fit* » (CRCSF), **Source *:*** https://www.memoireonline.com/08/10/3831/Approche-distribuee-pour-la-securite-dun-reseau-de-capteurs-sans-fils-RCSF.html (consulté le 16/03/2020).

[37]. «***Everything-as-a-Service (XaaS): Definition and Examples*** ».

Source: https://www.sam-solutions.com/blog/everything-as-a-service-xaas-definition-and-examples/ , Article par : SAKOVICH Natallia, Janvier 2019. (Consulté le 19/03/2020).

[38]. « ***Nationale Institute of standards and Technology (NIST)*** »,

Source : http://crs.nist.gov/groups/SNS/rbac/ (Consulté le 12/05/2020).

[39]. « ***Logiciel MICROSOFT Windows Server 2012 R2 Datacenter*** **»,**

Source:https://www.grosbill.com/4_Microsoft_windows_server_2012_r2_datacenter_-614364-jeux_video-logiciel_systeme (consulté le 28/08/2020).

[40]. ***« Prenez en main windows server*** »,

Source ***:*** *https://openclassrooms.com/ fr/courses/2356306-prenez-en-main-windows-server/ 5835091-prenez-en-main-les-roles-et-fonctionnalites* (consulté le 28/08/2020).

[41]. **« *Virtualisation*** »,

Source :https://www.appvizer.fr/magazine/services- informatiques/ virtualisation (consulté le 30/08/2020).

[42]. **« *Cours_cloud & virtualisation*** »,

Source :https://www.researchgate.net/ publication/338925290_Cours_cloud_virtualisation (consulté le 30/08/2020).

[43]. **« *L'essentiel sur Microsoft SQL Server 2014*** », **source : *https://www.lemagit.fr/ conseil/Lessentiel-sur-Microsoft-SQL-Server-2014 (*consulté le 02/09/2020).**

[44]. Microsoft SQL Server**,**

Source : *https://fr.wikipedia.org/wiki /Microsoft_SQL_Server#Web_Services* (**Consulté le 02/09/2020**).

[45]. **« *Bienvenue dans l'IDE Visual Studio*** »,

Source : https://docs.microsoft.com/fr-fr /visualstudio/ide/visual-studio-ide (consulté le 05/09/2020).

[46]. ***« Que pouvez-vous faire avec Visual Studio ?*** » ,

Source : https://visualstudio.microsoft.com /fr/vs/features/ (consulté le 07/09/2020).

[47]. **« *Développez une iPhone avec le modèle MVC*** »,

Source : https://openclassrooms.com/fr/courses/4504796-developpez-une-application-iphone-avec-le-modele-mvc/4571084-decouvrez-le-modele-mvc(consulté le 07/09/2020).

[48]. « ***Language Integrated Query (LINQ)*** »,

Source: https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/linq/ (consulté le 09/09/2020).

[49]. **« *C# Tutorial*** »,

Source : https://www.w3schools.com/cs/ (consulté le 09/09/2020).

[50]. « ***Introduction à Entity FrameWork*** »**,**

Source :https://pmusso.developpez.com/tutoriels/dotnet/entity-framework/introduction/ (consulté le 09/09/2020).

[51]. « ***Base de données*** »,

Source : https://www.base-de-donnees.com/orm/ (consulté le 10/09/2020).

[52]. « ***Apprenez ASP.Net MVC*** » , **source** : https://openclassrooms.com/fr/ courses/ 1730206-apprendre-asp-net-mvc (consulté le 07/09/2020).

[53]. «***LINQ to SQL Using Visual Studio***»,

Source: https://www.c-sharpcorner.com/ article/linq-to-sql-using-visual-studio/ (Consulté le 09/09/2020).