REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mohamed Khider – BISKRA
Faculté des Sciences Exactes, des Sciences de la Nature et de la Vie
Département d'informatique

N° d'ordre : 20/M2/2020

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Réseaux et Technologie d'Informatique et Communication**

Contrôle d'accès basé sur les rôles (RBAC) et les attributs (ABAC) dans le Cloud Computing

---

Par :

TORCHI NESRINE

Soutenu le date juin 2020, devant le jury composé de :

Nom et prénom	Grade	Président
Houhou Okba	MAA	Rapporteur
Nom et prénom	Grade	Examinateur

Remerciement

Tout d'abord, nous remercions le Dieu, notre créateur de nos avoir donné Les forces, la volonté et la patience durant ces longues années d'étude.

La première personne que nous tenons à remercier est notre encadrant Mr : ***Houhou Okba****, pour sa gentillesse et son soutien et aussi pour l'orientation, la patience qui a constitué un apport considérable sans lequel ce travail n'aurait pas pu être menée au bon port.*

Nous tenant à remercier sincèrement aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail.

On n'oublie pas nos parents pour leur contribution, leur soutien et leur patience.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragées au cours de la réalisation de ce mémoire

Dédicace

C'est avec l'immense plaisir et le grand honneur que je dédie ce mémoire :

A MES CHERS PARENTS

Aucune dédicace ne saurait exprimer mon respect, mon amour éternel et ma considération pour les sacrifices que vous avez consenti pour mon instruction et mon bien être. Je vous remercie pour tout le soutien et l'amour que vous me portez depuis mon enfance et j'espère que votre bénédiction m'accompagne toujours.

Puisse Dieu, le Très Haut, vous accorder santé, bonheur et longue vie et faire en sorte que jamais je ne vous déçoive.

A MES CHERS ET ADORABLE FRERES ET SOURS

***Faiza**, la prunelle de mes yeux, **Souad**, la douce, **Amal**, l'aimable, **Yassin** le généreux et **Dassi** mon petit frère qui j'adore, Je vous souhaite une vie pleine de bonheur et de succès.*

A MES AMIS : Nedjma, Abire, Ratiba.

En souvenir de notre sincère et profonde de amitié et des moments agréable qui nous avons passé ensemble

A tous ceux qui me sont chers et proches.

Nesrine.

RÉSUMÉ

La technologie de l'internet se développe de manière exponentielle depuis sa création. Actuellement, Le Cloud Computing a connu des grand progrès, toutes les entreprises change leur infrastructure vers le Cloud en raison des nombreux avantages qu'elle offre.

Hormis, le passage au Cloud présente également un certain nombre de défis. Dans ce mémoire, nous nous focalisions sur l'un des principaux défis, à savoir le contrôle d'accès et la délégation dans un Cloud

Nous avons présenté deux cadres le contrôle basé sur les rôles (RBAC), les attributs (ABAC) et de délégation, qui permet déléguer l'accès, révoquer les droits d'accès et d'évaluer le taux de confiance dans déférent cas

Mot clé : [cloud computing, politique, policy, XACML, ALFA, WSO2, RBAC, ABAC]

ABSTRACT

Internet technology has developed exponentially since its inception. Currently, Cloud Computing has experienced great progress, all companies change their infrastructure to the Cloud because of the many advantages it offers.

Aside from this, the move to the cloud also presents a number of challenges. In this thesis, we focused on one of the biggest challenges, namely access control and cloud delegation.

We have presented two frameworks role-based (RBAC),attribute control (ABAC)and delegation that allows for delegating access, revoking access rights and assessing confidence in deferential cases.

Keyword: [cloud computing, politique, policy, XACML, ALFA, WSO2, RBAC, ABAC]

ملخص

تطورت تكنولوجيا الإنترنت بشكل كبير منذ نشأتها. حاليًا ، شهدت الحوسبة السحابية تقدمًا كبيرًا ، حيث قامت جميع, ا لشركات بتغيير بنيتها التحتية إلى السحابة بسبب المزايا العديدة التي تقدمها ، بالإضافة إلى ذلك ، فإن الانتقال إلى السحابة ,يقدم أيضًا عددًا من التحديات.

في هذه الأطروحة، ركزنا على اثنان من أكبر التحديات ، ألا وهو التحكم في الوصول وتفويض السحب.

قدمنا إطار التحكم في الوصول والتفويض الذي يسمح بتفويض الوصول، وسحب حقوق الوصول والثقة المتميزة في مختلف الحالات.

Table de matière

Chapitre 2 : Contrôle d'accès

Chapitre 3 : Conception et Implémentation

Liste de figures

Liste de tableaux

Le cloud computing est considéré comme l'un des paradigmes les plus dominants dans l'industrie des technologies de l'information (TI) de nos jours. Il offre de nouveaux services rentables à la demande telle que le logiciel en tant que service (SaaS), l'infrastructure en tant que service (IaaS) et la plate-forme en tant que service (PaaS). Cependant, avec tous ces services promettant des installations et des avantages, il existe encore un certain nombre de défis associés à l'utilisation de l'informatique en nuage tels que la sécurité des données, les initiés malveillants et les cyber-attaques. Parmi toutes les exigences de sécurité du cloud computing, le contrôle d'accès est l'une des exigences fondamentales pour éviter l'accès non autorisé aux systèmes et protéger les actifs des organisations. [31]

Un système informatique distribué est une collection de postes ou calculateurs autonomes qui sont connectés à l'aide d'un réseau de communication. Chaque poste exécute des composantes, par exemple des séquences de calculs et utilise un middleware, Une propriété importante des systèmes distribués est que la distribution est généralement cachée pour l'utilisateur et les programmeurs de l'application, Cette system est généralement séparable en plusieurs composantes entièrement autonomes.

L'information dans le cloud computing est susceptible d'être partagée entre différentes entités, qui pourraient avoir différents degrés de sensibilité, et ce partage doit être plus sécurisé et il faut garanti que les donné accessible aux utilisateurs autoriser sont permet avec un accès fluide aux ressources partagées tout en respectant les limites des droits d'accès dans un système.

Le contrôle d'accès est l'une des exigences communes et fondamentales pour tous les types d'utilisateurs du cloud. Cependant, les modèles de contrôle d'accès classiques ne peuvent pas être appliqués dans l'environnement cloud pour les raisons suivantes:

- Différentes autorisations d'accès à un même utilisateur de cloud, et lui donnant la possibilité d'utiliser plusieurs services en ce qui concerne l'authentification et l'heure de connexion.
- Le partage des ressources entre les locataires potentiels non fiables, l'hébergement mutualisé et la virtualisation, les mécanismes permettant de transférer les références des clients entre couches pour accéder aux services et aux ressources sont des aspects cruciaux de tout modèle de contrôle d'accès déployé dans le cloud computing.

Dans un environnement de cloud computing, différents fournisseurs de services Cloud (CSP) doivent établir des relations de confiance les uns avec les autres lors de l'exécution afin de partager les

ressources de l'autre. Grâce à cette relation, les utilisateurs peuvent non seulement utiliser les ressources d'autres fournisseurs CSP approuvés, mais également déléguer des droits d'accès. [30]

Le passage au Cloud estompe les limites du périmètre de sécurité réseau classique, les organisations ont donc des difficultés à fournir, implémenter et gérer des règles d'accès unifiées aux ressources distribuées de l'entreprise. [41]

La problématique du contrôle d'accès ne se limite plus à la volonté d'assurer la sécurité des locaux et bureaux en faisant barrage aux intrusions de malfaiteurs et les pirates : il faut désormais être en mesure de savoir qui est entré où, à quel moment. [42]

Notre objectif dans ce travail, Nous allons présenter deux cadres de contrôle d'accès « basé sur les rôles (RBAC) et les attributs (ABAC) » avec la délégation. Les deux cadres proposé un système basé sur des attributs environnementaux avec des rôles est capable de s'adapter à des changements sans précédent car il peut déléguer des droits d'accès à des utilisateurs non autorisés dans une situation d'urgence et révoquer les droits d'accès des utilisateurs. De plus, notre système ses caractérises par un temps plus court, traçabilité avec un gestion d'administration facile, nous trouvons également des résultats très précis avec garantie la sécurité dans système à travers ces deux cadres (RBAC, ABAC).

Ce mémoire est constitué de Trois chapitres et de brève introduction et conclusion générale :

Chapitre 1 : ici on va étudier les principes généraux du Cloud Computing en traitant la définition, les services, les modèles, les caractéristiques, Après on va la sécurité informatique.

Chapitre 2 : Ce chapitre traite principalement des contrôles d'accès et les modèles existants et détail sur le modèle de contrôle d'accès basé sur les rôles (RBAC) et les attributs (ABAC) et on va terminer par la notion de délégation, avec quelques des travaux connexe.

Chapitre 3 : dans ce dernier chapitre on va présenter l'architecture protocolaire et l'implémentation de notre système**.** Nous implémentons des politiques qui gère un contrôle d'accès basé sur rôles et les attributs, on mettrons notre politique de sécurité à l'épreuve et contrôler son bon fonctionnement à l'aide de l'outil WSO2 Identity Server.

Enfin, une conclusion générale qui comporte les apports de notre travail ainsi que les perspectives envisagées est présentée ce mémoire.

Chapitre 1 :

Cloud Computing

1.1. Introduction

L'Internet actuel fournit un contenu sous les formes des vidéos, e-mails et les informations services dans les pages web. Avec le Cloud Computing, la prochaine génération d'internet va nous permettre d'acheter des services informatiques à partir d'un portail web. Avec le progrès constant des technologies de stockage et de calcul, le passage aux technologies Cloud Computing devient imminent. Et si l'on parle Cloud on parle aussi des ressources partagées que il faux les protéger et à contrôler l'accès à ces ressources pour cela on besoin la sécurisation de nos ressource informatique.

La sécurisation des systèmes informatiques est un domaine ouvert uniquement à un certain nombre de personnes regroupant des compétences et un savoir-faire avéré. En effet, la sécurité informatique fait ressortir plusieurs notions telles que l'intégrité, la disponibilité, la confidentialité, la non-répudiation et l'authentification des données circulant au sein d'un système d'information. Ainsi, l'un des axes principaux de la sécurité informatique est le contrôle d'accès. [2]

Dans ce premier chapitre, nous allons mettre l'accent sur le Cloud Computing, la sécurité du système informatique et nous allons présenter les avantages et les inconvénients du Cloud Computing et ses caractéristiques essentielles. Enfin, nous avons essayé de déterminer le rôle de ce système et leur efficacité d'utilisation des services informatiques.

1.2. L'informatique en nuage (Cloud Computing)

La définition exacte du Cloud Computing est encore en évolution. Le Cloud Computing est un nuage de services et de données. Plus précisément, c'est un paradigme, et à ce titre il est difficile de lui donner une définition exacte et de dire avec certitude s'il s'agit ou non de Cloud. Nous considérons la définition de Cloud Computing proposée par l'Institut national de la norme et de la technologie (NIST). [11]

1.2.1. Définition de l'informatique en nuage

Le Cloud Computing est un modèle informatique qui permet un accès facile et à la demande par le réseau à un ensemble partage de ressources informatique configurable (serveurs, stockage, application et services) qui peuvent être rapidement provisionnées et libérées par un minimum d'efforts de gestion ou d'interaction avec le fournisseur du service. » [1]

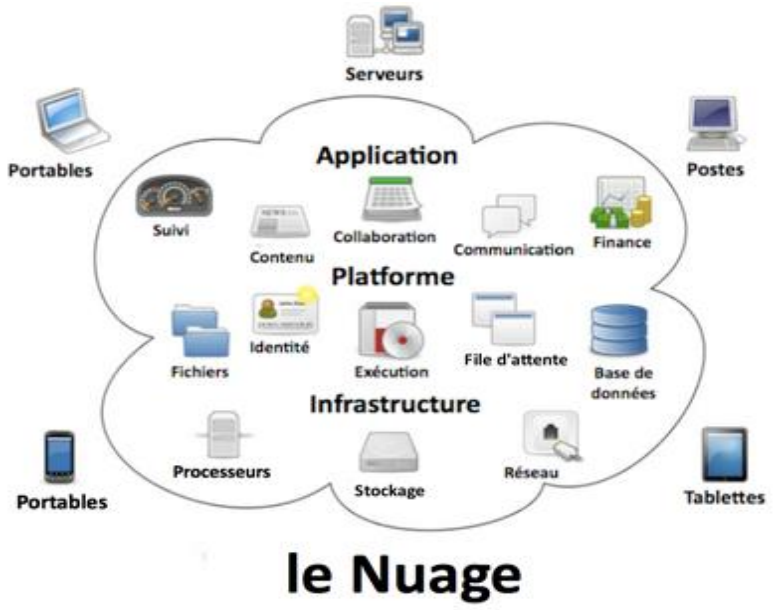

Figure1.1. Schéma donnant un aperçu sur les facteurs principaux du Cloud Computing. [9]

1.2.2. Caractéristiques de l'informatique en nuage

Le modèle Cloud se différencie par les cinq caractéristiques suivantes [2] :

- **Accès aux services par l'utilisateur à la demande :** La mise en œuvre des systèmes est entièrement automatisée et c'est l'utilisateur, au moyen d'une console de commande, qui met en place et gère la configuration à distance.
- **Accès réseau large bande :** Ces centres de traitement sont généralement raccordes directement sur le backbone internet pour bénéficier d'un excellente connectivite. Les grands fournisseurs repartissent les centres de traitement sur la planète pour fournir un accès aux systèmes en moins de 50 ms de n'importe quel endroit.
- **Réservoir de ressources non localisé :** La plupart de ces centres comportent des dizaines de milliers de serveurs et de moyens de stockage pour permettre des montées en charge rapides. Il est souvent possible de choisir une zone géographique pour mettre les données « près » des utilisateurs.
- **Redimensionnement rapide, élasticité :** La mise en ligne d'une nouvelle instance d'un serveur est réalisée en quelques minutes, l'arrêt et le redémarrage en quelques secondes. Toutes ces opérations peuvent s'effectuer automatique des scripts. Ces mécanismes de gestion permettent de bénéficier de calcul au trafic instantané.

- **Facturation à l'usage** : Il n'y a généralement pas de cout de mise en service (c'est l'utilisateur qui réalise les opérations). La facturation est calculée en fonction de la durée et de la quantité de ressources utilisées. Une unité de traitement stoppée n'est pas facturée.

1.2.3. Les services de l'informatique en nuage

Le Cloud Computing fournit un espace dans lequel il est possible de placer, de manière virtuelle, des infrastructures serveur ou réseau, des plateformes de développement ou d'exécution, des catalogues de services, sous forme des services accessibles à la demande. Le groupe de travail NIST a divisé ces services en trois grandes modèles (Figure 1.2).

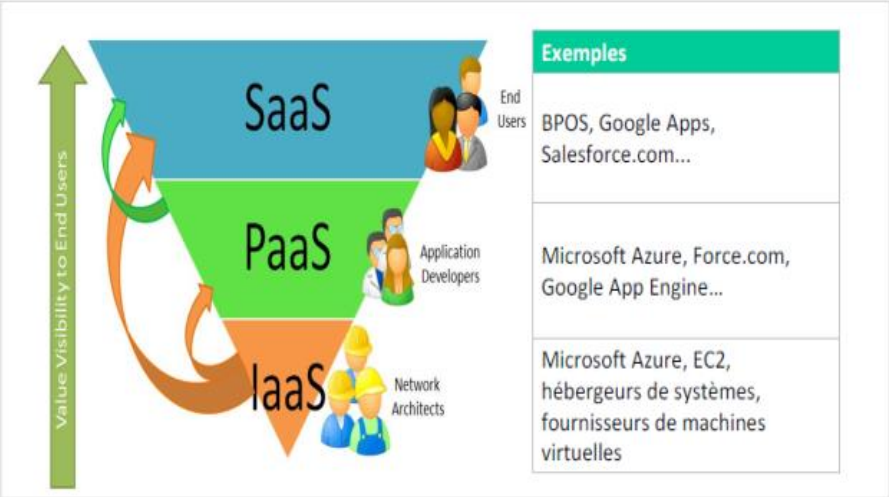

Figure1.2. Modèles des services. [4]

1.2.3.1. SaaS (Software as a Service)

Ce modèle de service est caractérise par l'utilisation d'une application partagée qui fonctionne sur une infrastructure Cloud. L'utilisateur accède à l'application par le réseau à travers de diverse types de terminaux (souvent via un navigateur web).l'administrateur de l'application ne gère pas et ne contrôle pas l'infrastructure sous-jacente (réseau, serveur, applications, stockage). Il ne contrôle pas les fonctions de l'application à l'exception d'un paramétrage de quelques fonctions utilisateurs limitées. [1]

1.2.3.2. PaaS (Platform as a Service)

L'utilisateur a la possibilité de créer et de déployer sur une infrastructure Cloud PaaS ses propres applications en utilisant les langages et les outils du fournisseur. L'utilisateur ne gère pas ou ne contrôle pas l'infrastructure Cloud. Grâce aux PaaS, le

déploiement d'applications dans différents environnements est très facile (test, pré-production et production sans se soucier de l'infrastructure et de la plateforme dans lesquelles vont s'exécuter l'application ou le stockage de données). [1]

1.2.3.3. IaaS (Infrastructure as a Service)

L'utilisateur loue des moyens de calcul et des stockages, des capacités réseau et d'autres ressources indispensables (partage de charge, pare-feu, cache).l'utilisateur a la possibilité de déployer n'importe quel type de logiciel incluant les systèmes d'exploitation. L'utilisateur ne gère pas ou ne contrôle pas l'infrastructure Cloud sous-jacente mais il a le contrôle sur les systèmes d'exploitation, le stockage et les applications. Il peut aussi choisir les caractéristiques principales des équipements réseau comme le partage de charge, les pare-feu. [1]

1.2.4. Les modèles de déploiement

Ces modèles de déploiement, au nombre de quatre, sont définis en fonction de leur relation à l'entreprise. [2]

1.2.4.1. Cloud public

L'infrastructure Cloud est ouverte au public ou à de grands groupes industriels. Cette infrastructure est possédée par une organisation qui vend des services Cloud. C'est celui de la plate-forme Amazon Web Services déjà citée. Dans le domaine du Cloud, et du Cloud public en particulier, la sécurité est la clé de voute. L'infrastructure étant partagée avec de nombreux clients la sécurité est de la responsabilité du fournisseur qui en assure la gestion. Ce point est d'autant plus crucial que le client n'a qu'un degré de contrôle et de surveillance très faible des aspects physiques et logiques de sécurité sur les ressources qui sont mises à sa disposition. Le fournisseur doit donc tout mettre en œuvre pour garder la confiance de ses utilisateurs. [28]

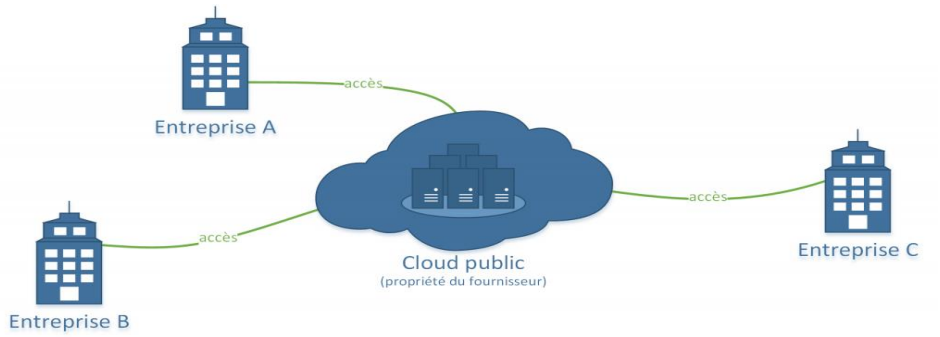

Figure1.3. Modèle de déploiement d'un Cloud public.

1.2.4.2. Cloud privé

L'infrastructure Cloud est utilisée par une seule organisation. Elle peut être gérée par l'organisation ou par une tierce partie. L'infrastructure peut être placée dans les locaux de l'organisation ou à l'extérieur. Le Cloud privé a pour ambition d'offrir certains avantages du Cloud Computing tout en en limitant ses inconvénients. Un Cloud privé est détenu par l'entreprise utilisatrice, cela nécessite d'acheter, de construire et de maintenir l'ensemble de ses constituants, ce qui implique de supporter un investissement initial très important. Les Clouds privés diffèrent des Clouds publics en ce que les réseaux, serveurs, et infrastructures de stockage qui lui sont associés sont dédiés à une seule entreprise et ne sont pas partagés avec d'autres. Puisque le Cloud est entièrement contrôlé par l'entreprise elle-même, les risques de sécurité associés à un Cloud privé sont minimisés. Ce haut degré de contrôle et de transparence permet au propriétaire d'un Cloud privé de se conformer plus facilement à des normes, politiques de sécurités ou conformités réglementaires qui peuvent être requises dans certains domaines. [28]

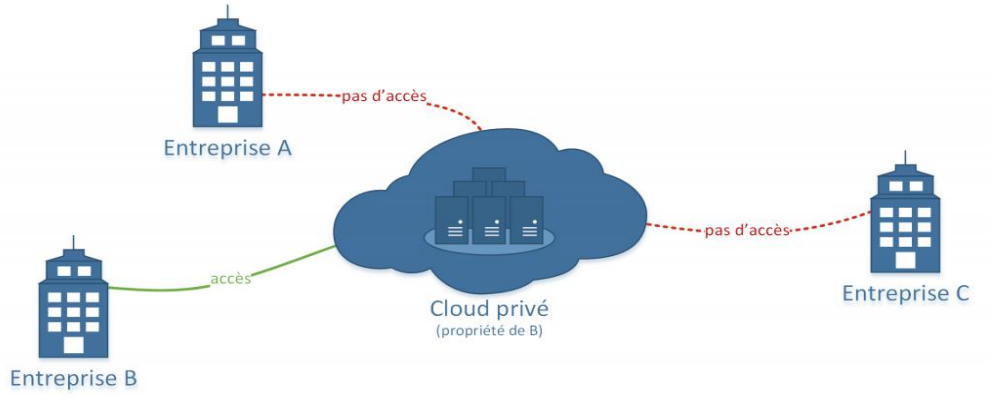

Figure1.4. Modèle de déploiement d'un Cloud privé.

1.2.4.3. Cloud communautaire

L'infrastructure Cloud est partagée par plusieurs organisations ou entreprises pour les besoins d'une communauté qui souhaite mettre en commun des moyens (sécurité, conformité, etc.).Elle peut être gérée par les organisations ou par une tierce partie et peut être placée dans les locaux ou à l'extérieur. Un Cloud communautaire est une forme hybride de Cloud privé construit et exploité spécifiquement pour un groupe restreint et ciblé. Ces communautés ont des exigences semblables et réunissent leurs moyens humains et financiers pour atteindre leurs objectifs communs. L'infrastructure

commune est spécifiquement conçue pour répondre aux exigences d'une communauté; à titre d'exemple, des organismes gouvernementaux, des hôpitaux ou des entreprises de télécommunication qui auraient des contraintes de réseau, de sécurité, de stockage, de calcul ou d'automatisation similaires pourraient trouver des intérêts communs à déployer collectivement un Cloud communautaire. [28]

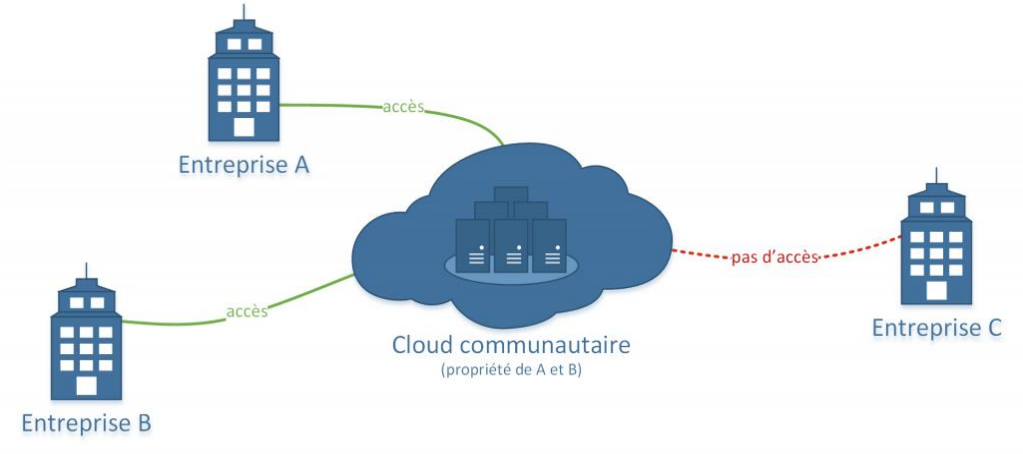

Figure1.5. Modèle de déploiement d'un Cloud communautaire.

1.2.4.4. Cloud hybride

Comme son nom l'indique, L'infrastructure du Cloud hybride est la combinaison de plusieurs modèles de déploiement de Clouds. Ces infrastructures sont liées entre elles par la même technologie qui autorise la portabilité des applications et des données. C'est une excellente solution pour repartir ses moyens en fonction des avantages recherches. Dans un Cloud hybride, les Clouds public, privé ou communautaire restent des entités uniques, mais sont reliés entre eux par une technologie normalisée ou propriétaire qui permet la portabilité des données et des applications. En raison de la complexité que la combinaison de plusieurs types de Clouds engendre, la conception, la gestion et le maintien d'un Cloud hybride peuvent être un véritable défi. [28]

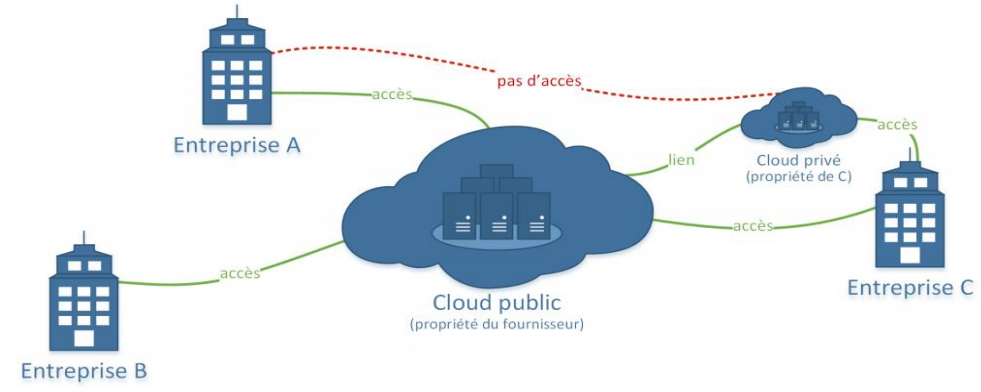

Figure1.5. Modèle de déploiement d'un Cloud hybride.

Le tableau 1 résume les caractéristiques de chaque modèle de déploiement de la manière suivante: [4]

- **La propriété** : Organisation (O), Un tiers, ou les deux.
- **Le contrôle** : Organisation (O), Un tiers, ou les deux.
- **Le coût** : Faible, Moyen ou Elevé.
- **La localisation** : Hors site, Sur place, ou les deux.
- **La sécurité** : Faible, Moyen ou Élevé.

Modèle de déploiement	**Propriété**	**Contrôle**	**Coût**	**Localisation**	**Sécurité**
Le Cloud public	Un tiers	Un tiers	Faible	Hors site	Faible
Le Cloud privé	O ou un tiers	O ou un tiers	Élevé	Sur place	Elevé
Le Cloud communautaire	O ou un tiers	O ou un tiers	Elevé	Sur place	Elevé
Le Cloud hybride	Les deux	Les deux	Moyen	Les deux	Moyen

Tableau 1.1: Principales caractéristiques de chaque modèle de déploiement.

1.2.5. Les applications de l'informatique en nuage

Les grandes entreprises du secteur informatique se sont massivement impliquées dans les activités liées au cloud computing, et proposent un éventail de services attenants, espace de stockage alloué, service de messagerie, outils collaboratifs, agilité, disponibilité, productions, RS, CRM, relation client. [10]

Un exemple grand-public du Cloud Computing est iCloud d'Apple, lancé en septembre 2011, le système de sauvegarde et de synchronisation pour iOS et Macintosh qui offre 5 Go de stockage gratuit, ou encore, en France, Molotov TV, un service de distribution de chaînes de télévision permettant d'enregistrer ses émissions préférées dans le Cloud de la société17. Un exemple entreprise est celui d'Office 365, qui propose en abonnement tout un ensemble de services professionnels de types messagerie, stockage, synchronisation, communication, réseau social d'entreprise. [10]

1.3. Les avantages et les inconvénients de l'informatique en nuage

1.3.1. Les Avantages

Les avantages du Cloud Computing sont [3] :

- **Souplesse d'évolution :** il n'y a pas de logiciel à installer et l'accès se fait avec un simple navigateur web.
- **Simplicité :** l'entreprise cliente n'a plus besoin de développements coûteux et déplace la responsabilité du fonctionnement du service sur le fournisseur.
- **Liberté de changement de service :** le Cloud Computing étant généralement facturé à la demande ou par abonnement mensuel, il est très facile pour une entreprise d'arrêter le service si elle n'en a plus besoin ou si elle souhaite aller chez un concurrent.
- **Coût :** la force du Cloud Computing réside dans la possibilité de proposer le même service à un grand nombre d'utilisateurs, finalement, le coût de Cloud Computing sera donc très raisonnable. Ainsi que
  - Bas coût d'ordinateurs, d'infrastructure et de softwares.
  - Rendements élevés.
  - Capacité de stockage illimitée.
  - Peu d'entretien.
  - Mises à jour instantanées de logiciel.
  - Sureté accrue de données.
  - Une collaboration plus facile de groupe.
  - Accès universel aux documents.

1.3.2. Les Inconvénients

Les inconvénients du Cloud Computing [3] :

- **Confidentialité et sécurité des données :** les données sont hébergées en dehors de l'entreprise. Les fournisseurs proposant le service héberge des données d'entreprise utilisatrice, Cela peut donc poser un risque potentiel pour l'entreprise de voir ses données mal utilisées ou volées. Il s'agit donc d'assurer que le fournisseur dispos d'une sécurité suffisante et qu'il propose une politique de confidentialité concernant les données d'utilisateur.
- **Dépendance:** si l'entreprise souhaite des fonctionnalités très spécifiques, il peut être difficile de convaincre le fournisseur de proposer ces fonctionnalités. Et

en général, s'il y a un problème, l'entreprise est tributaire du service client d'un fournisseur. Il s'agit donc de choisir un fournisseur en qui l'on a confiance.

- ✓ Besoin d'un raccordement constant d'Internet.
- ✓ Exige une grande largeur de bande.
- ✓ Peut-être plus cher pour certain cas d'utilisation.

1.4. La sécurité informatique

Le système d'information représente un patrimoine essentiel de l'organisation, qu'il convient de protéger.

1.4.1. Définition

Ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir la sécurité de l'information, du système d'information et des systèmes et ressources informatiques. [22]

C'est une dixipline qui se veut de protéger l'intégrité et la confidentialité des informations stockées dans un système informatique. Elle a pour lent de veiller à ce que les ressource d'un système d'information puissent être utilisées tel qu'une organisation ou qu'un utilisateur l'ait décidé sans interférences.

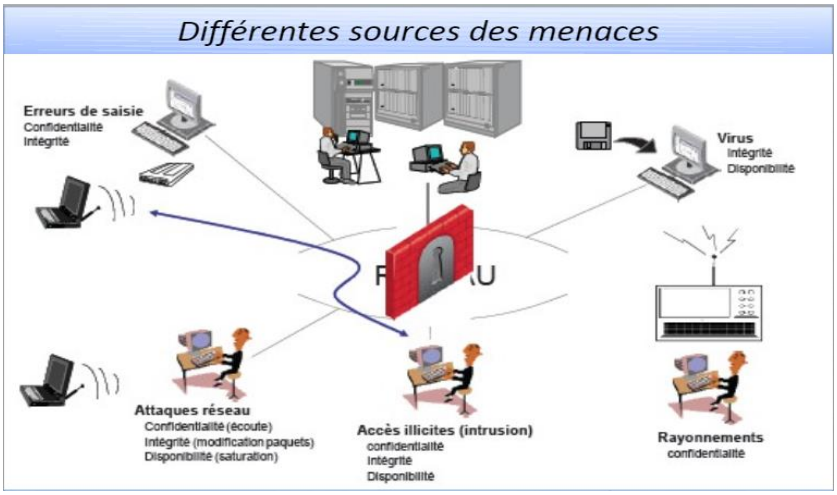

Figure 1.7. Exemple la sécurité des systèmes informatique [22]

1.4.2. Les technologies de sécurisation

Certains utilisateurs utilisent l'isolement physique pour protéger leurs serveurs. Du point de vue de la technologie, la sécurité des données des utilisateurs peut être réfléchie dans les règles suivantes [23]:

➢ L'authentification consiste à demander à un utilisateur de prouver son identité (en fournissant un mot de passe ou des données biométriques, par exemple).

➢ la confidentialité garantit aux utilisateurs qu'aucune donnée n'a pu être lue et exploitée par un tiers malveillant.

➢ l'intégrité assure aux utilisateurs que leurs données n'ont pas été indûment modifiées au cours de la transmission dans le réseau.

➢ La non-répudiation empêche un utilisateur de nier la réalité d'un échange de données.

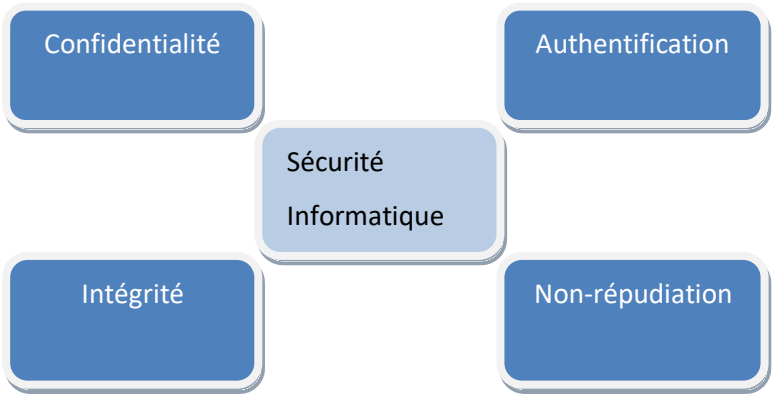

Figure 1.8. Les technologies La sécurité informatique.

1.4.3. La politique de sécurité [6]

- Définit le cadre d'utilisation des ressources du système d'information.
- Identifie les techniques de sécurisation à mettre en œuvre dans les différents services de l'organisation.
- Sensibilise les utilisateurs à la sécurité informatique.

1.5. Conclusion

Le Cloud Computing est une nouvelle technologie d'utilisation des services informatiques, nous pouvons être beaucoup plus flexibles et productif dans l'utilisation des ressources allouées dynamiquement.

Dans ce chapitre, nous avons donné un aperçu détaillé sur l'approche du Cloud Computing. Nous avons présenté des définitions de cette notion et ses caractéristiques essentielles. Ensuite, nous avons montré comment le Cloud offre un large choix de services informatiques à la demande et avec facturation à l'usage aux utilisateurs selon leurs besoins. Ces services se présentent sous forme d'un logiciel, plate-forme ou infrastructure et qui sont déployés sous quater modèles possibles qui sont : le Cloud privé, le Cloud public, le Cloud communautaire et le Cloud hybride, Ensuit nous avons représenté La sécurité informatique du Cloud Computing.

En fin les utilisateurs finaux trouvent que cette technologie est un bon choix pour l'utilisation des services. Malgré, toutes ces solutions produites par le Cloud, il existe toujours des limites. Nous avons présenté les principaux défis dont la technologie Cloud Computing doit faire face pour améliorer la qualité des services fournis aux utilisateurs.

Chapitre 2 :

Contrôle d’accès

2.1. Introduction

L'abonnement d'une entreprise à un opérateur du Cloud Computing peut être prolongé au déport de l'ensemble de son système d'information, mais le problème de la sécurité des données qui se pose déjà au niveau du software interne, risque de devenir une problématique Plus compliquée suite à cette migration. [4]

Si l'on parle Cloud on parle aussi de ressources partagées ce qui nous pousse à protéger ces ressources et à contrôler l'accès. Le contrôle d'accès est indispensable pour la sécurité dans les systèmes informatiques. Paradoxalement son étude n'a pas reçu beaucoup d'attention de la part de la communauté de la recherche. Depuis le début des années 1980, la doctrine des politiques mandataires et discrétionnaires établie dans le contexte des systèmes militaires a lentement été remise en cause comme base appropriée pour le contrôle d'accès dans les systèmes civils. Ensuite nous avons vu l'introduction de différents modèles de contrôle d'accès (par exemple le contrôle d'accès basé sur les rôles) développés pour répondre aux différents besoins de protection dans les systèmes civils. [5]

Dans Ce Chapitre nous allons présenter les concepts du contrôle d'accès, puis on va les typé de contrôle d'accès avec comparaison enter typés de contrôle d'accès, ensuit nous allons présenter le domaine du contrôle d'accès.

2.2. Contrôle d'accès

La politique de sécurité se compose de trois sous politiques de contrôle : d'accès physique, administratif et logique. Nous nous intéressons plus particulièrement à la politique de contrôle d'accès logique et aux modèles de protection et mécanismes nécessaires pour la réaliser [5].

2.2.1. Définition

Un système de contrôle d'accès est une collection de composants et de méthodes qui déterminent l'admission correcte des utilisateurs légitimes aux activités en fonction des autorisations d'accès préconfigurées et des privilèges définis dans la politique de sécurité d'accès .[40]

Cette system qui peut être utilisée pour déterminer les utilisateurs ou les programmes autorisent à voir ou à utiliser les ressources d'un environnement informatique.il existe deux types principaux de contrôle d'accès : physique et logique.

- Le contrôle d'accès physique permet de limiter les accès aux campus, aux bâtiments, salles et aux matériels informatiques.

- Le contrôle d'accès logique restreint les connexions aux réseaux informatiques, aux fichiers système et aux données.

L'objectif fondamental de tout système de contrôle d'accès est de restreindre l'utilisateur à ce qu'il devrait pouvoir faire et de protéger les informations contre tout accès non autorisé. [38]

Le modèle de base de toutes les politiques de contrôle d'accès est montré sur cette figure:

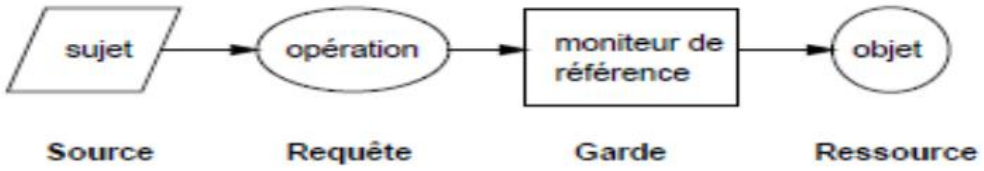

Figure 2.1. Le modèle de base du contrôle d'accès [39]

La figure montre un sujet qui souhaite faire une opération sur un objet. Le système transforme l'opération en une requête qu'il passe au moniteur de référence qui contrôle l'accès aux ressources. Si le sujet est autorisé à accéder à l'objet selon la politique de sécurité en vigueur, l'accès à l'objet va être accordé et l'opération peut se dérouler normalement. [14]

Il consiste à vérifier si une entité (une personne, un ordinateur, etc.) désireuse d'accéder à une ressource possède les droits nécessaires pour le faire. [8]

2.2.2. Les objectifs de contrôle d'accès

Le contrôle d'accès a pour objectifs : [9]

- de gérer et contrôler les accès logiques aux ressources informationnelles par des personnes ou des dispositifs.
- De détecter les accès non autorisés.
- De préciser les règles à observer en matière d'identification, d'authentification et d'autorisation d'accès des personnes ou des dispositifs.
- D'assurer la disponibilité de l'information en réduisant :
  - ✓ les attaques de déni de service
  - ✓ la propagation d'un code malicieux entre systèmes informatiques
  - ✓ les erreurs d'opération ou de configuration des applications
- D'assurer l'intégrité de l'information en réduisant :
  - ✓ Les altérations par des utilisateurs non autorisés
  - ✓ Les erreurs d'utilisation.

- D'assurer la confidentialité de l'information en réduisent :
  - ✓ Les accès non autorisés
  - ✓ Les diffusions non autorisées

2.2.3. La Politique de contrôle d'accès

Les politiques de contrôle d'accès sont définies comme étant des directives (règles) de haut niveau qui spécifient qui a la permission d'exercer quoi sur quelle donnée. A partir de cette définition nous dégageons trois concepts fondamentaux d'une politique de contrôle d'accès qui sont: [6]

- **Sujet :** entité active qui accède aux données du système. Le sujet peut être un utilisateur, une application, une adresse IP …
- **Objet :** entité passive qui représente les données à protéger. L'objet peut être, par exemple, un fichier, une table relationnelle, une classe …
- **Action :** représente l'action à traiter par le sujet sur l'objet. L'action peut être lire, écrire, exécuter….

Permettre aux sujets dans une organisation un accès illimité aux objets peut compromettre la sécurité de cette organisation, où la nécessité d'une politique de contrôle d'accès qui permet de déterminer si des sujets sont autorisés à exécuter des activités particulières. La forme générale des règles d'une politique de contrôle d'accès est la suivante : [7]

Un sujet à { La permission / L'interdiction / L'obligation } de réaliser l'action a sur l'objet o

2.2.4. Les modèles des contrôles d'accès

Il existe une grande variété de modelés en raison des différences dans les exigences pour les politiques de sécurité commerciale, deux types de politiques distinctes ont été élaborés, deux modèles différents de contrôle d'accès: Discretionary Access Control (DAC) et Mandatory Access Control (MAC). Ces modèles ont un certain nombre de failles, ce qui a conduit à la proposition d'autres modèles tels que Role Based Access Control (RBAC) et Attribute Based Access Control (ABAC).

2.2.4.1. Contrôle d'accès discrétionnaire (DAC)

Le modèles de politiques discrétionnaires (DAC - Discretionary Access Control) considèrent que chaque sujet peut détenir un droit de possession sur un objet. Le

propriétaire de l'objet peut alors accorder des droits sur son objet à d'autres sujets. Il en résulte cependant un problème de perte de confidentialité de l'information. Plusieurs modèles sont associés au DAC : le modèle Take-Grant, le modèle Lampson, etc. [18]

Les contrôles sont dits discrétionnaires dans le sens où le sujet est capable de transférer les permissions d'accès à d'autres sujets (La transmission des droits est exercée à la discrétion du sujet). Le contrôle d'accès discrétionnaire (DAC) a été principalement implanté au sein des systèmes d'exploitation (Microsoft Windows, Solaris, Linux, FreeBSD). [24]

Dans ces systèmes les règles d'autorisation sont exprimées sous forme positive ou négative. Une règle d'autorisation positive spécifie l'ensemble des sujets qui peuvent accéder aux objets. Une règle d'autorisation négative spécifie l'ensemble des sujets qui ne peuvent pas accéder aux objets. L'implantation de ce modèle a donné lieu à la constitution de matrices d'accès initialement introduite en 1971 par Lampson [Lampson 1971] qui a été généralisée en 1976 par Harison,Ruzzo et Ullman (HRU) [Harrison et al. 1976]. Dans ce dernier, 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 2.1). [24]

Objets / Sujets	Fichier	Table
Alice	Lire Ecrire Exécuter	Exécuter
Bob	Lire	Exécuter

Tableau 2.1. Exemple d'une matrice d'accès. [24]

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. [24]

Il existe en pratique deux approches pour implémenter la matrice d'accès :

- **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 DAC n'a pas la possibilité de contrôler le flux d'informations ou de gérer les chevaux de Troie pouvant hériter des autorisations d'accès .En outre, un utilisateur peut transmettre ses droits à un autre utilisateur, ce qui peut porter atteinte à l'intégrité et à la confidentialité des objets.[26]

2.2.4.2. Contrôle d'accès obligatoire (MAC)

Afin d'apporter une solution aux problèmes de fuites d'information des modèles de contrôle d'accès discrétionnaires, les modèles d'autorisation obligatoire (MAC - Mandatory Access Control) centralisent complètement l'autorité d'administration. Il s'agit d'une restriction des politiques de sécurité où les sujets ne peuvent altérer l'accès aux objets. Ainsi, le problème de perte de confidentialité ne peut exister. Les modèles associés au MAC : le modèle de Biba, modèle de Clark et Wilson, modèle de muraille de Chine. Prenons le modèle multi-niveaux de Bell et La Paula qui vise la confidentialité. [18]

Prenons le modèle multi-niveaux de Bell et La Padula qui vise la confidentialité. Ce modèle est basé sur la classification des sujets et des objets. Le principe consiste à attribuer une classe d'accès à chaque sujet et à chaque objet. Généralement, une classe d'accès est constituée de deux composants : un niveau de sécurité et un ensemble de catégories. Le niveau de sécurité est un élément d'un ensemble hiérarchique ordonné, tel que Top Secret > Secret > Confidential > Unclassified. L'ensemble des catégories est un sous-ensemble d'un ensemble non ordonné, dont les éléments représentent soit une compétence, une région, un département. Pour éviter les fuites d'information, l'accès aux objets doit obligatoirement respecter deux principes fondamentaux:

- **No Read up :** un sujet est autorisé à lire un objet donné uniquement si sa classe d'accès domine la classe d'accès de l'objet.
- **No Write down :** un sujet est autorisé à écrire dans un objet donné uniquement si La classe d'accès de l'objet domine sa classe d'accès.

D'après la politique de contrôle d'accès, un fichier contenant des informations sensibles ne peut être accédé que par un utilisateur exécutant une application d'un niveau de sécurité *secret*. Si un utilisateur invoque l'application avec un niveau de sécurité *unclassified*, l'opération de lecture de ce fichier sera bloquée (Le principe de No-read up). Alors, en respectant les deux principes, le modèle MAC résout le problème de fuite d'information des modèles DAC. Il est quand même un modèle très rigide, car il ne permet pas de gérer les exceptions entre les différents niveaux de sécurité. [24]

Le MAC a besoin d'une autorité centrale pour déterminer quelles informations devraient être rendues accessibles et par qui. Par exemple, un responsable peut vouloir accéder à des informations sur un membre du personnel, mais il ne devrait pas avoir accès au fichier des membres, car il pourrait accéder à des informations sensibles telles que les coordonnées bancaires. [26]

2.2.4.3. Contrôle d'accès basé sur les Rôles (RBAC)

Le modèle RBAC (Role Based Access Control) est un type de contrôle d'accès à un système d'information dans lequel chaque décision d'accès est basée sur le rôle assigne à l'utilisateur. Dans le modèle RBAC, élaboré par le National Institute of Standards and Technology (NIST) à partir de 1992, les habilitations sont affectées à des rôles. [25]

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. [26]

Ce modèle 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 et Sybase Adaptive Server ont mis en œuvre tout ou partie des principes des modèles RBAC. [11]

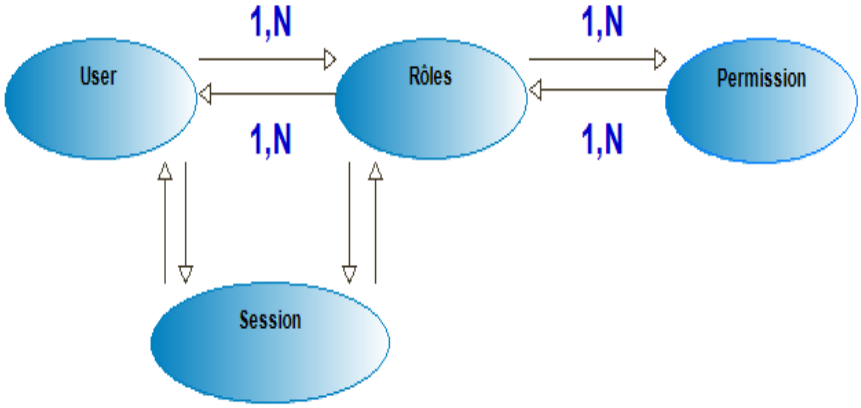

Figure 2.2: Model RBAC.

Des exemples de type RBAC :

Exemple 1 : Les étudiants ont le droit d'accéder uniquement à leurs données personnelles.

Exemple 2 : Seuls les étudiants qui se trouvent dans le bâtiment A ont le droit d'accéder au serveur de leur université.

Exemple 3 : Le temps de réponse est crucial dans de nombreuses applications comme un système de soins de santé.

Pour le premier exemple, une solution envisageable est de créer pour chaque étudiant un rôle privé. Théoriquement c'est une solution mais en pratique cela n'est pas faisable car s'il existe un très grand nombre d'étudiants, cela fait perdre à RBAC sa simplicité d'administration. En outre, pour le deuxième exemple un nouvel élément doit être introduit dans le modèle correspondant à la « location physique de l'utilisateur ». Pour le troisième exemple Un consultant absent d'un hôpital doit accéder au système en temps opportun, sans tenir compte d'un certain nombre de demandes d'accès au RBAC et à distance.

2.2.4.3.1. L'architecture d'autorisation

Dans ce modèle, les permissions sont affectées à des rôles. La gestion des permissions est alors simplifiée. La permission est représentée par un couple (r, a) avec ($r \in R$ et $a \in A$), tel que R représente l'ensemble de ressources et A l'ensemble des actions. Ensuite, les sujets peuvent être attribués aux rôles qui découlent généralement de la structure d'une organisation (Figure 3.1). [15]

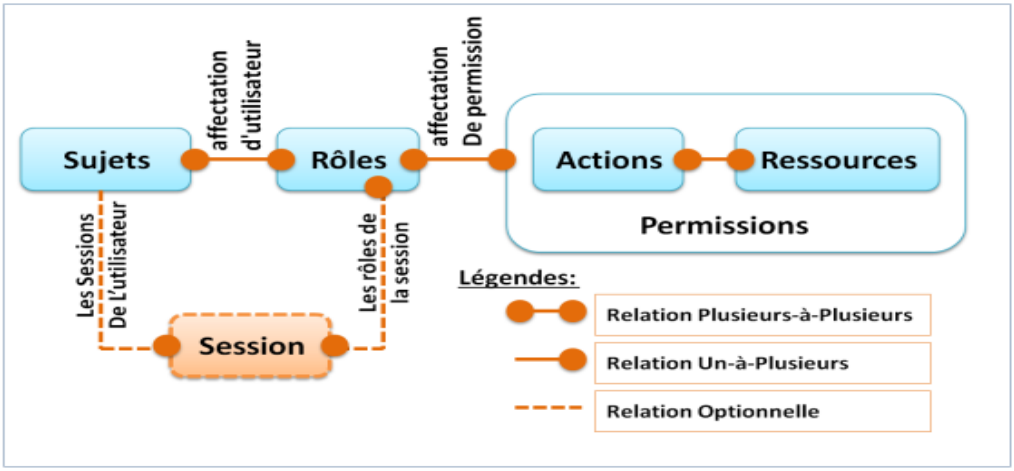

Figure 2.3. Architecture du modèle RBAC. [15]

2.2.4.3.2. Déclinaisons d'un RBAC

Par ailleurs, le NIST propose plusieurs déclinaisons du modèle RBAC:[26]

- Le modèle RBAC0 ou « the flat model », qui présente les concepts et relations de base
- 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.

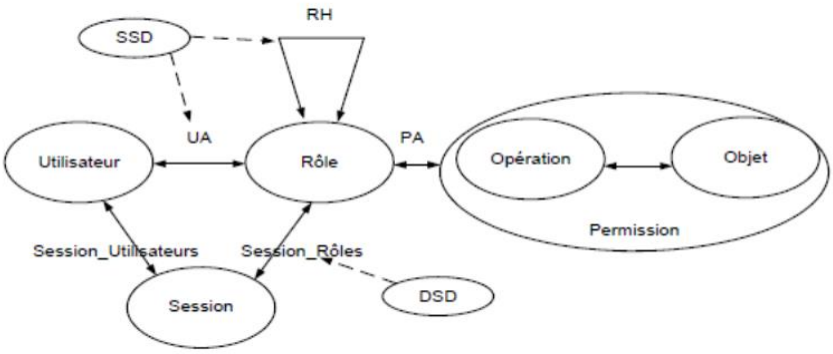

Figure 2.4. L'architecture d'autorisation RBAC. [26]

Le modèle RBAC0 définit les concepts ou entités de base pour spécifier les politiques de contrôle accès RBAC. Le modèle RBAC1 ajoute la possibilité de construire une

hiérarchie de rôles, i.e. une approche pour structurer cette notion de rôle. Les versions 0 et 1 du modèle RBAC introduisent les ensembles suivants :

- **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
- UA ⊆ Utilisateur × Rôle : permet d'affecter des rôles aux utilisateurs
- PA ⊆ Permission × Rôle : permet d'affecter des permissions aux rôles
- 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.

Les deux modèles RBAC2 et RBAC3 apportent 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, RBAC propose deux types de contraintes:

- les séparations statiques (Static Separation of Duties ou SSD)
- les séparations dynamiques (Dynamic Separation of Duties ou DSD).

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.

La DSD évite qu'un utilisateur possède deux rôles en conflit en même temps dans une même session.

2.2.4.3.3. Caractéristiques RBAC

- permettant des délégations à la fois de l'administration des rôles et de l'administration des droits d'accès.
- permettre que les degrés de confiance soient associés aux délégations de rôle et d'accès.
- pouvoir spécifier les autorisations positives et négatives à accorder aux rôles.
- permettre de spécifier et de raisonner sur la confiance dans les politiques RBAC déléguées.

2.2.4.3.4. Avantages et Inconvénients de modèle RBAC

Les avantages	Les inconvénients
✓ Fournit une politique neutre/flexible. ✓ la contrainte des séparations des devoirs. ✓ Capacité d'exprimer DAC, MAC, et les politiques spécifique d'utilisateur en utilisant la hiérarchie des rôles et les contraintes.	✓ Caractéristiques administratives faible. ✓ Satisfait le principe du moindre privilège. ✓ Peut être facilement incorporé dans les technologies courantes.

Tableau 2.2. Les avantages et les inconvénients RBAC.

2.2.4.4. Contrôle d'accès basé sur les attributs (ABAC)

Le modèle ABAC (Attribute Based Access Control) repose sur un ensemble d'attributs associés à un demandeur ou à une ressource à consulter pour prendre des décisions d'accès.Les attributs peuvent ou non être liés les uns aux autres. Après avoir défini les attributs utilisés dans le système, chaque attribut est considéré comme une valeur discrète et les valeurs de tous les attributs sont comparées à un ensemble de valeurs par un point de décision de politique pour accorder ou refuser l'accès.

Ces types de modèles sont également connus sous le nom de contrôle d'accès basé sur la politique (PBAC) ou de contrôle d'accès basé sur les réclamations (CBAC). De plus, un sujet n'a pas besoin d'être préalablement connu du système, il doit simplement s'authentifier auprès du système puis fournir ses attributs. Cependant, parvenir à un accord sur le type d'attributs à utiliser et sur le nombre d'attributs pris en compte pour prendre des décisions d'accès est une tâche complexe dans le Cloud computing. Ce modèle n'a pas encore été mis en œuvre pour des systèmes d'exploitation bien connus.

Enfin, il est essentiel de proposer une politique de sécurité pouvant fonctionner correctement avec ce type de modèle de contrôle d'accès, car la politique de sécurité

est responsable de la sélection des attributs importants qui sont utilisés pour prendre des décisions d'accès. [14]

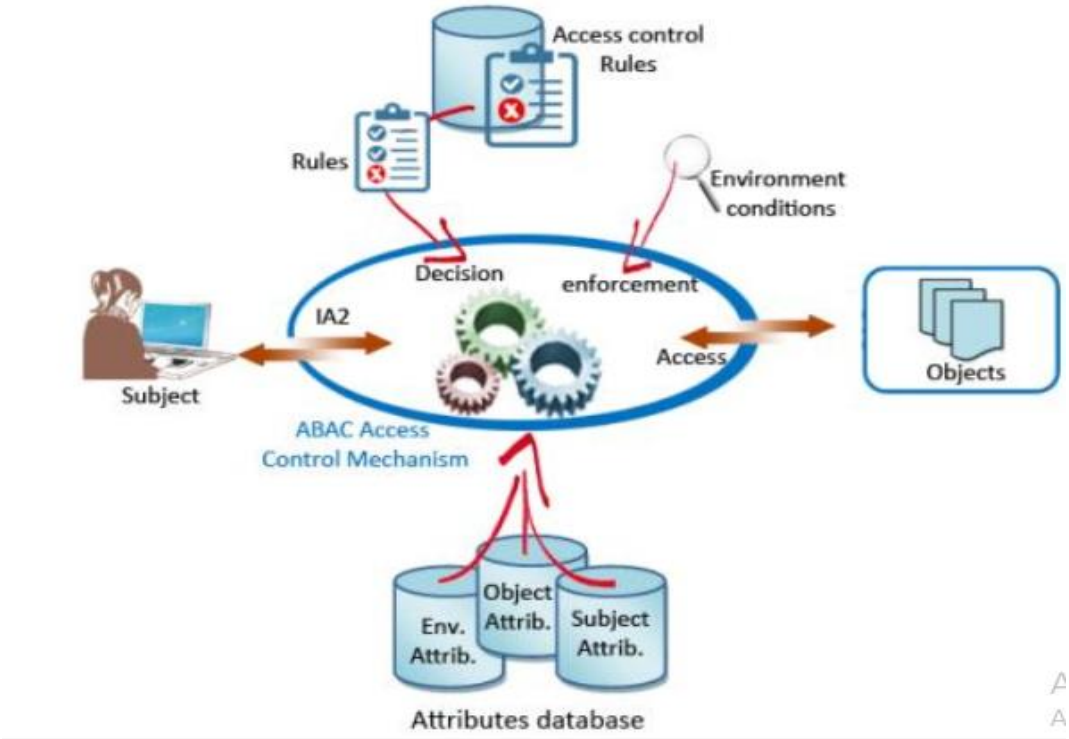

Figure 2.5. Modèle de contrôle d'accès ABAC. [17]

Des exemples de type ABAC :

Exemple 1 : Le Règle: Un utilisateur dont le rôle est « Médecin » ou « Infirmière » de type « Clinique » et ayant réussi la certification « Règles de PRP HIPAA » peut lire et ajouter des notes dans un dossier de type « Patient » du même « GMF » dont l'attribut « Consentement GMF » est « Oui ».

Exemple 2 : Le Règle: Un utilisateur dont le rôle est «Conseiller en architecture» affecté à un projet est collaborateur du site du projet (accès en écriture). Affectation de l'utilisateur: code de projet autorisé dans le système de feuilles de temps Rôle de l'utilisateur: système de GRH.

2.2.4.4.1. L'architecture d'autorisation

Une fois les politiques définies, il est nécessaire d'expliciter la façon dont le système va effectuer les vérifications des règles avant de fournir ou non l'autorisation d'accès. L'architecture d'autorisation définie par ABAC est la suivante : [15]

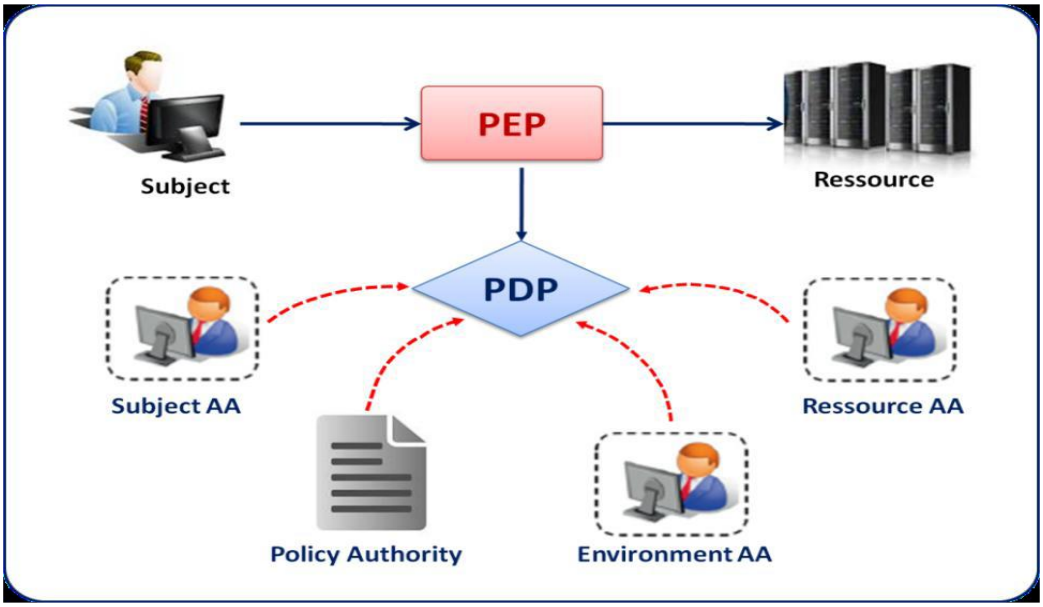

Figure 2.6. Architecture d'autorisation définie par ABAC. [15]

- **Les AA (Atributes Authorities)** : sont les entités responsables de la création et de la gestion des attributs. Ils sont également responsables d'établir les relations entre les attributs, leur valeur et l'entité correspondante.

- **Le PEP (Policy Enfoncement Point):** est l'entité chargée d'effectuer les requêtes d'autorisation et d'appliquer la politique. Il est logiquement situé entre les sujets et les ressources, ce qui lui permet d'intercepter toute tentative d'accès, d'effectuer la requête vers le système de sécurité afin de vérifier si la tentative d'accès est autorisée ou non. Notons que s'il est représenté ici comme un point unique, il peut être physiquement distribué en plusieurs points du système. La seule condition étant que le système soit architecturée de manière à ce qu'il ne soit pas possible d'accéder à une ressource protégée sans passer par le PEP.

- **Le PDP (Policy Decision Point):** est l'entité chargée d'évaluer les politiques applicables et de prendre une décision concernant une requête d'accès à une ressource par un sujet. Il reçoit donc les requêtes du PEP, contacte les différents AA pour récupérer les attributs qui ne sont pas présents dans la requête, et applique les règles de sécurité pour donner sa décision au PEP (accès autorisé ou refusé).

- **La PAP (Policy Authorit) :** crée et gère les politiques de contrôle d'accès (règles de décision, conditions, etc.).

2.2.4.4.2. Les attributs

Comme son nom l'indique, le modèle ABAC définit les autorisations d'accès en se basant sur des caractéristiques de chaque entité, appelés attributs. Trois groupes d'attributs se distinguent selon le type de l'entité à laquelle ils s'appliquent [18] :

1. **Les attributs des sujets** : un sujet est une entité qui peut agir sur une ressource. A chaque sujet on associe des attributs qui définissent son identité et ses caractéristiques. Par exemple le rôle du sujet peut aussi être considéré comme un attribut, tout comme le nom, le prénom, ou le titre, etc.
2. **Les attributs des sujets** : un sujet est une entité qui peut agir sur une ressource. A chaque sujet on associe des attributs qui définissent son identité et ses caractéristiques. Par exemple le rôle du sujet peut aussi être considéré comme un attribut, tout comme le nom, le prénom, ou le titre, etc.
3. **Les attributs d'environnement :** l'environnement peut être décrit par des informations opérationnelles, techniques, liées à la situation ou encore au contexte dans lequel l'accès à l'information se produit.

2.2.4.4.3. Formalisme de politique de sécurité ABAC

La formalisation de politique de sécurité ABAC est indiqué comme suivant:[18]

- S, R et E sont respectivement les sujets, les ressources et les environnements
- SAk, ($1 \leq k \leq K$), RAm, ($1 \leq m \leq M$), AEn ($1 \leq n \leq N$) sont respectivement les(k-ième, mième et n-ième) attributs d'un sujet, d'une ressource, d'un environnement(avec k, m et n compris entre 1 et le nombre d'attribut définit pour chaque entité)
- ATTR(s), ATTR(r) et ATTR(e) sont les relations d'attributions des attributs aux entités (sujet, ressource et environnement) respectivement.

$ATTR(s) \subseteq SA1 \times SA2 \times \ldots \times SAk$

$ATTR(r) \subseteq SA1 \times SA2 \times \ldots \times Sa2m$

$ATTR(e) \subseteq SA1 \times SA2 \times \ldots \times SA2n$

Les prédicats d'attributs sont définis de la façon suivante :

- « CurrentDate(e) = 6/12/2008 » signifie qu'on affecte la valeur 6/12/2008 à l'attribut d'environnement CurrentDate
- « Role(s) = "Service Consumer" » le sujet s joue le rôle de consommateur de service
- « ServiceOwner(r) = "XYZ, Inc." » la ressource ServiceOwner est libellée par

XYZ,Inc. Les règles sont définies comme étant des fonctions booléennes des attributs de s, r et e.On définit ensuite une politique comme un ensemble de règles regroupant plusieurs sujets et plusieurs ressources au sein d'un même domaine de sécurité. La gestion des autorisations se fait alors via l'évaluation de l'ensemble des règles de la politique.

Rule (X) : can_access (s, r, e) ← f (ATTR(s), ATTR(r), ATTR(e)).

Soit un ensemble de sujets et de ressources. On définit pour chaque sujet un attribut nommé «rôle», et pour chaque ressource un attribut « Name ». La règle « Les managers peuvent accéder aux ressources nommées ApprovePurchase » s'exprime alors de la façon suivante :

Rule 1 : can_access (s, r, e) ← (Rôle(s) = 'Manager') ∧ (Name(r) = 'ApprovePurchase').

2.2.4.4.4. Caractéristiques ABAC

Le modèle ABAC est connu par plusieurs caractéristiques lesquelles: [19]

- Plus de flexibilité dans la prise de décision
- Moins de travail pour établir des liens statiques entité-rôle et rôle-permission
- Plus de travail au moment de la prise de décision (impact de performance)
- Meilleur contrôle d'accès en conditions changeantes
- Plus difficile de retracer la justification d'une décision prise à un moment précis
- Fortement recommandé de documenter les règles en langage d'affaires
- Plus difficile d'appliquer des politiques d'entreprise centralisées (PBAC)
- Il faut associer des attributs aux objets (métadonnées)
- La gestion des attributs peut être déléguée.

2.2.4.4.5. Les applications ABAC :

Le concept d'ABAC peut s'appliquer à n'importe quel niveau de la pile de technologies et d'une infrastructure d'entreprise. Par exemple, ABAC peut être utilisé au niveau du pare-feu, du serveur, de l'application, de la base de données et de la couche de données. L'utilisation d'attributs apporte un contexte supplémentaire pour évaluer la légitimité de toute demande d'accès et informer la décision d'accorder ou de refuser l'accès. [20]

1. Sécurité de l'application : un des principaux avantages d'ABAC est que les stratégies et l'attribut d'autorisation peuvent être définis de manière neutre sur le plan technologique. Cela signifie que les stratégies définies pour les API ou les bases de données peuvent être réutilisées dans l'espace d'application. Les applications courantes pouvant tiré parti d'ABAC sont:

- Systèmes de gestion de contenu
- Applications maison
- Applications web

2. Sécurité de la base de données: la sécurité des bases de données est depuis longtemps spécifique aux fournisseurs de bases de données: les solutions Oracle VPD, IBM FGAC et Microsoft ,RLS sont autant de moyens permettant d'obtenir une sécurité de type ABAC plus fine.ABAC permet de définir des stratégies s'appliquant à plusieurs bases de données. C'est ce qu'on appelle le masquage dynamique des données Un exemple serait :

- Politique: les gestionnaires peuvent voir les transactions dans leur région
- Stratégie retravaillée d'une manière centrée sur les données: les utilisateurs avec le rôle == manager peuvent effectuer l'action == SELECT sur la table ==TRANSACTIONS si user.region == transaction.region

3. Sécurité de big data : le contrôle d'accès basé sur les attributs peut également être appliqué à des systèmes Big Data tels que Hadoop. Des stratégies similaires à celles utilisées précédemment peuvent être appliquées lors de la récupération de données à partir de lacs de données.

4. Sécurité du serveur de fichiers : depuis Windows Server 2012, Microsoft a mis en œuvre une approche ABAC pour contrôler l'accès aux fichiers et aux dossiers. Cela est possible grâce aux listes de contrôle d'accès dynamiques (DACL) et au langage de définition de descripteur de sécurité (SDDL). SDDL peut être considéré comme un langage ABAC car il utilise les métadonnées de l'utilisateur (revendications) et du fichier / dossier pour contrôler l'accès.

5. Sécurité des API et des micros services : ABAC peut être utilisé pour appliquer une autorisation détaillée basée sur des attributs aux méthodes ou fonctions de l'API.

2.2.4.4.6. Avantages et Inconvénients de modele ABAC

Les avantages	Les inconvénients
✓ Plus de flexibilité dans la prise de décision ✓ Moins de travail pour établir des liens statiques entité-rôle et rôle-permission ✓ Meilleur contrôle d'accès en conditions changeantes.	✓ Plus difficile de retracer la justification d'une décision prise à un moment précis ✓ Il faut associer des attributs aux objets (métadonnées) ✓ difficile d'effectuer un audit avant le fait et de déterminer les autorisations disponibles pour un utilisateur spécifique.

Tableau 2.3. Les avantages et les inconvénients ABAC.

2.2.4.5. Combiner RBAC et ABAC

Les entreprises commencent souvent par mettre en place un RBAC plat. Ce modèle est plus facile à configurer et à entretenir. À mesure que les organisations grandissent et gèrent des données plus sensibles, elles réalisent le besoin d'un système de contrôle d'accès plus complexe. RBAC et ABAC peuvent être utilisés ensemble, RBAC effectuant le gros du travail et ABAC le complétant par un filtrage plus fin. [33]

Ce modèle d'accès est également appelé **RBAC-A** .Il existe trois approches RBAC-A qui gèrent les relations entre les rôles et les attributs:[33]

- **Centré sur les attributs :** Un rôle devient le nom de l'un des attributs utilisateur. Cela ressemble à un titre de poste. L'attribut «rôle» dans un tel modèle est utilisé pour marquer un ensemble d'attributs requis pour une certaine position.
- **Centré sur le rôle :** Des attributs sont ajoutés pour contraindre les rôles. Dans un tel modèle, les attributs peuvent réduire les autorisations disponibles pour un utilisateur. Cette approche renforce la sécurité de vos données.
- **Rôles dynamiques :** Des attributs tels que l'heure de la journée sont utilisés pour déterminer le rôle du sujet. Dans certains cas, le rôle d'un utilisateur peut être entièrement déterminé par des attributs dynamiques.

2.2.4.6. Comparatif des modèles de contrôle d'accès

Le Tableau 2 ci-dessous présente un comparatif résumé des propriétés des principales familles de modèles de contrôle d'accès qui sont déjà traités relativement aux

droits d'accès, à l'implémentation du contrôle de flux et au support d'architectures multi domaines.

	MAC	DAC	RBAC	ABAC
Autorité de sécurité	Centrale	Utilisateur	Généralement centrale	Centrale
Audit d'accès	Centrale	Utilisateur	Centrale	Centrale
Propagation des droits d'accès	Centrale	Utilisateur	Généralement centrale	Centrale
Contrôle de flux d'information	OUI	NON	NON	NON
Multi-domaines	NON	NON	NON	OUI

Tableau 2.4. Comparatif des modèles de contrôle d'accès. [17]

2.2.5. Les domaines d'application

Le contrôle d'accès est utilisé dans une plusieurs domaine lesquelles:[16]

- **Les entreprises:** les entreprises traitent des données confidentielles à sein de leurs entités. Ces données sont stockées dans des bases de données informatiques ou physiquement dans des locaux. Cela suppose que tout le monde ne peut pas avoir accès à toutes ces données. Pour cela les entreprises mettent en place des contrôles d'accès logiques. La création de comptes utilisateurs avec des mots de passe, ou par l'attribution de badges électroniques ou encore par un contrôle biométrique sont utilisés dans les entreprises.
- **L'administrateur :** L'administrateur du système d'information configure l'accès ou non aux utilisateurs aux différents logiciels et bases de données du système d'information. C'est donc l'administrateur qui définit les autorisations selon les utilisateurs.
- **Les gouvernements:** tous les gouvernements ont une obligation de protection vis-à-vis de leurs systèmes d'information sensibles. Les États Unis le font à
- travers la NSA. Le gouvernement français par l'agence nationale de la sécurité des systèmes d'information a émis une liste d'opérateurs d'importance vitale où la sécurité des bases de données se doit d'être forte car vitale pour le pays. Ces

opérateurs sont aussi bien des entreprises comme EDF ou la SNCF que des administrations comme la défense nationale.

2.3. Délégations

La délégation est définie comme la délégation des droits d'accès que l'on a sur une ressource à un autre utilisateur. De plus, la délégation est un mécanisme pratique dans un environnement dynamique, un utilisateur peut avoir besoin pour accomplir une tâche. Il fournit également des moyens d'exprimer et d'appliquer des politiques de contrôle d'accès. [21]

On note trois classes de délégation :

- **Chaîne d'informations d'identification:** le délégant (ou le service de jeton de sécurité côté délégant) crée un nouveau jeton de sécurité fourni au délégataire. Exemples pour cela classe sont Sec PAL et SPKI.
- **Manipulation des politiques :** le délégateur modifie la politique de contrôle d'accès de Ressource. Des exemples pour ce type sont la modification de la politique d'autorisation à un Service de jeton de sécurité (STS) ou à un point de décision de politique (PDP), ou en ajoutant un identificateur du délégataire (par exemple, certificat X.509, nom d'utilisateur, alias d'entreprise, empreinte digitale) à une liste de contrôle d'accès (ACL).
- **Hybride:** Combinaison de la chaîne d'informations d'identification et de la manipulation des politiques. Pour Par exemple, le délégant crée une nouvelle paire d'identifiants, l'enregistre au ressource, et le fournit au délégataire. Dans d'autres cas, le délégant pourrait demander le délégataire pour s'inscrire / créer un compte avant la délégation.

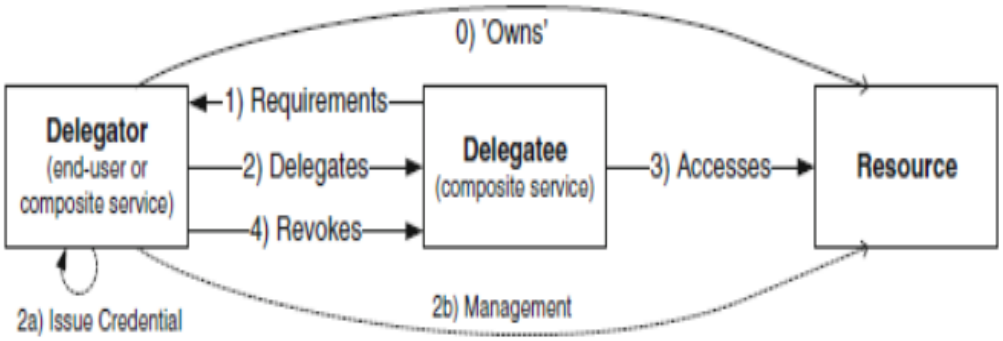

Figure 2.7. Scénario généralisé pour la « le délégant, le délégataire et la ressource ». [21]

2.4. Les travaux connexes

Le travail dans [12] porte sur « Flexible attribute enriched role based access control model », cette travail qui explique le contrôle d'accès, le contrôle d'accès basés sur les rôles et l'attribut, L'idée principale derrière le RBAC est qu'un rôle est un module intermédiaire entre les utilisateurs et les permissions, en particulier le contrôle d'accès basé sur les attributs (ABAC), ABAC a les avantages de DAC, MAC et RBAC et aussi surmonter la limitation de ce modèle, Les deux ont leur limitation qui est complémentaire à L'une et l'autre. De nombreuses recherches sont menées qui intègrent le modèle RBAC et ABAC. Il est nécessaire de développer le modèle qui surmonte la limitation de RBAC et ABAC. Pour cela ce travail a des objectifs sur ce problème de définir un système de contrôle d'accès basé sur les rôles enrichi d'attributs flexibles modèle de contrôle à ces dernières et garantir un niveau de protection adéquat des ressources protégées accessibles au travers ces interfaces. On va le modèle propose (Figure 3.4)

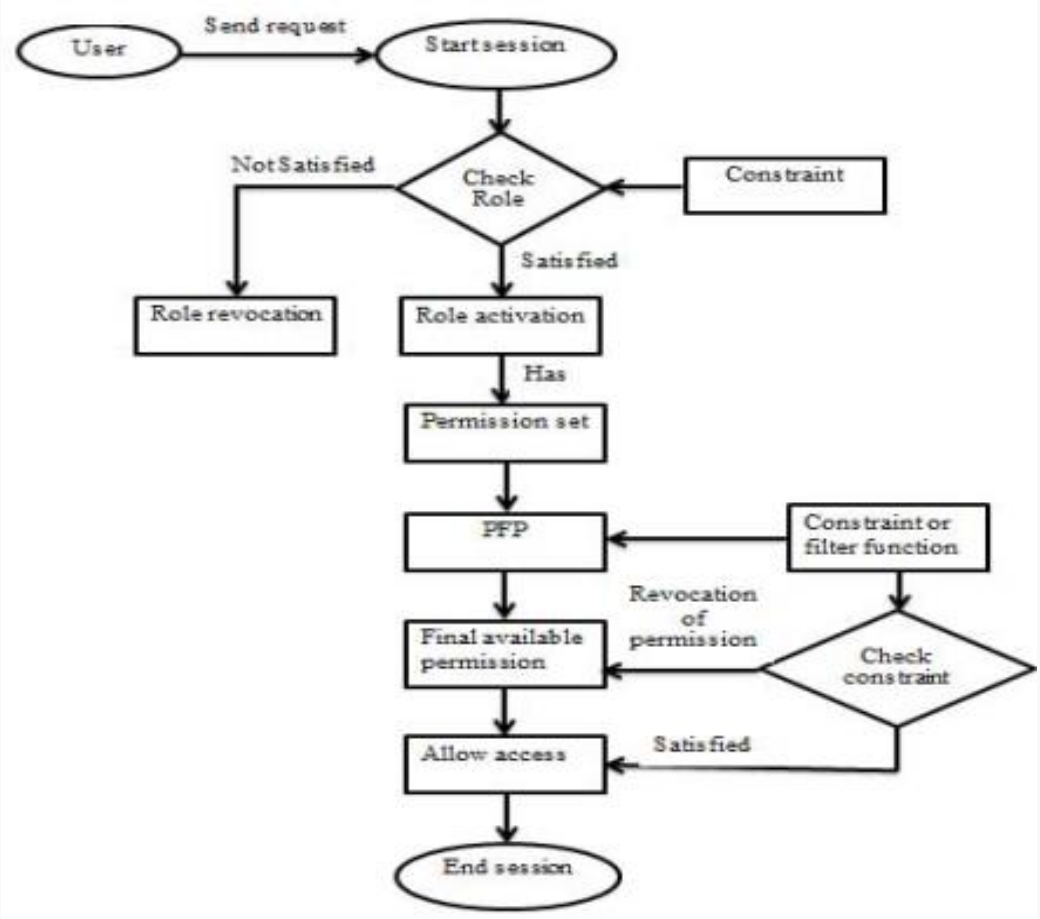

Figure 2.8 : Flux système proposé.

Dans le système proposé lorsque les utilisateurs envoient le demander que sa session démarre mais avant l'activation du rôle la condition contextuelle est vérifiée si elle est satisfaite alors seulement son le rôle est activé sinon le rôle n'est pas activé. Après l'activation du jeu de rôles d'autorisations associé vérifié contre la condition et enfin ne fournir l'accès que sur ces autorisations qui sont filtrées par la condition. Puis en accédant à cette condition si la condition de contexte devient échoue à tenir alors que l'autorisation est révoquée de l'autorisation finale disponible, Le système proposé présente de nombreux avantages et surmonte également la limitation de RBAC et ABA (comme: Visualisation facile des modifications de politique, Révocation des autorisations en fonction du contexte, ect.).

L'objectif principale de ce travail est mettre en place une politique de contrôle d'accès qui fournit l'intégration de RBAC et ABAC de manière à surmonter la limitation des deux RBAC et ABAC Fournissant ainsi le meilleur modèle que le pur RBAC et ABAC.

Dans [13] les auteurs ont proposé une modèle « RBAC-ABAC Model» qui est un nouveau modèle de contrôle d'accès basé sur le rôle et l'attribut, dans lequel RBAC est utilisé pour gérer les attributs statiques, tandis qui ABAC est utilisé pour gérer les attributs dynamiques. Ce modèle peut réduire efficacement le nombre de rôles et de règles de contrôle d'accès, rendant le contrôle d'accès plus flexible et facile à utiliser. De plus, nous appliquons ce modèle à l'environnement de réseau d'intégration de l'espace spatial et concevons un cadre de contrôle d'accès distribué en fonction des caractéristiques de ce réseau. Cela donne des idées pour la recherche et la construction du réseau d'intégration espace-sol.

Le nouveau modèle peut réduire le nombre de rôles et de règles de contrôle d'accès, réduire la complexité de la gestion, mais n'affecte pas la flexibilité du contrôle d'accès. Cet article conçoit un cadre de contrôle d'accès distribué en fonction des caractéristiques des réseaux d'intégration espace-sol, décrit les nœuds clés, les fonctions et les relations entre les nœuds dans le cadre, définit le modèle de flux de travail de contrôle d'accès, aud détermine le déploiement des nœuds dans le système de contrôle d'accès des réseaux d'intégration espace-sol.

Critiques

Dans les travaux précédents que on les vu sont traité quelques avantages pour l'utilisation de contrôle d'accès dans le Cloud Computing spécialement les deux modèles de contrôle d'accès RBAC et ABAC mais on remarque que ils n'ont pas touché le point de délégation au les contrôles d'accès à cause la difficulté de la délégation et la dangereux et la

sécurité dans Cloud Computing, on va présenter un cadre de délégation dans les contrôle d'accès adaptatif (dynamique) et s'adapter aux changements induits par la reconfiguration du système. Le cadre proposé est capable de s'adapter à des changements sans précédent car il peut déléguer des droits d'accès à des utilisateurs non autorisés dans une situation d'urgence et révoquer les droits d'accès des utilisateurs en fonction de facteurs environnementaux.

2.5. Conclusion

Nous avons vu dans ce chapitre la définition de contrôle d'accès, les types de l'application et les objectifs ensuite la politique de contrôle d'accès et les modèles de technologies et de capacités administratives, En effet on a expliqué les détaille sur les types des contrôles d'accès (MAC, DAC, RBAC, ABAC).Nous basons sur les types de contrôle d'accès RBAC et ABAC de réaliser une application, puis nous avons fait le tour sur Les domaines d'application avec une comparaison entre les types de contrôle d'accès.

Enfin nous avons représenté une comparaison entre les types de contrôle d'accès, nous allons voir quelques travaux connexes.

Chapitre 3 :

Conception et Implémentation

3.1. Introduction

Dans ce chapitre nous allons parler sur les étapes suivies pour concevoir et implémenter notre politique de sécurité qui repose sur les modèles RBAC (Role Based Access Controle)et ABAC (Attribute Based Access Control), les outils et les différents environnements de développement que nous avons utilisé, on va aussi présenter et expliquer quelques fragment de code source (XACML, ALFA) qui expliquent notre politique d'accès.

3.2. Conception

3.2.1. Architecture générale

L'architecture générale de notre Système de contrôle d'accès est présentée comme suit :

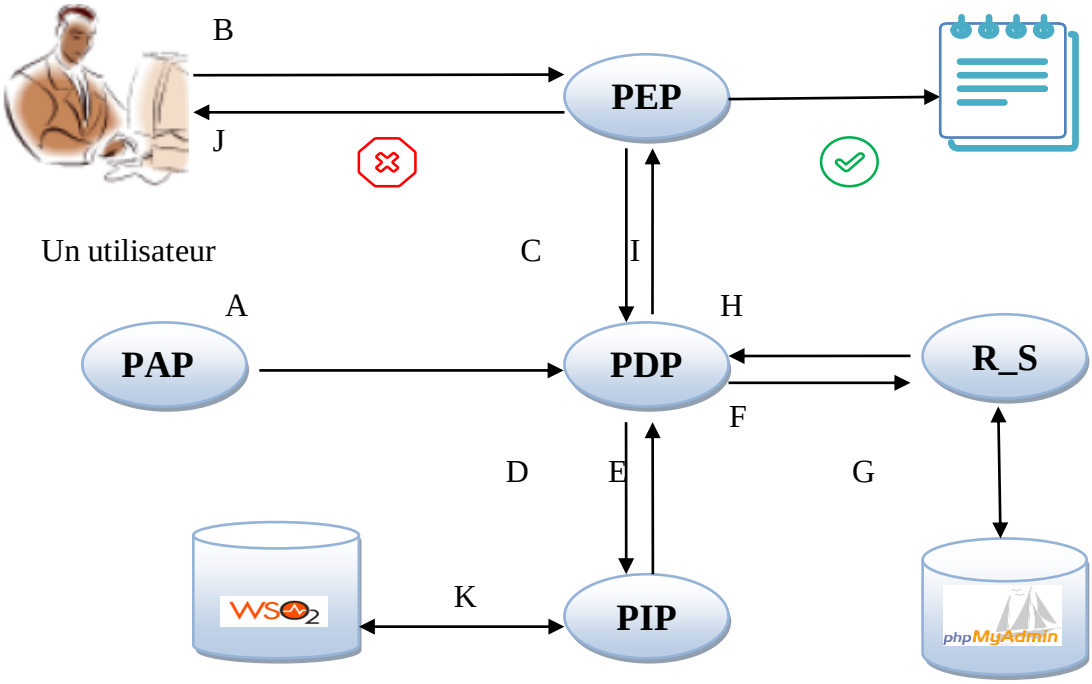

Figure 3.1. Architecture général.

PDP (Policy Decision Point): C'est l'entité qui détermine les politiques applicables à une requête et renvoie une décision d'autorisation. Dans notre architecture.

PEP (Policy Enforcement Point): C'est l'entité qui intercepte la requête et l'envoie pour l'évaluer au niveau PDP. Elle contrôle l'accès en appliquant la décision retournée par le PDP. Nous l'avons modélisé par une application java.

PAP (Policy Administration Point) : C'est l'entité du système (administrateur) qui définit les politiques et les rend disponibles au PDP.

PIP (Policy Information Point): C'est l'entité qui extrait des informations supplémentaires pour le PDP avant de prendre la décision.

R_S (Risque De Score) : C'est l'entité qui calcul une opération pour risque de score s'il y a grand.

A. L'administrateur définit la politique ou l'ensemble de politiques et les rend disponibles au PDP
B. L'utilisateur envoie une requête d'accès au PEP
C. Le PEP envoie la requête au PDP
D. Le PDP demande à son tour des attributs au PIP
E. Le PIP envoie les attributs nécessaires pour le PDP
F. Le PDP demande aussi à l'entité R_S
G. Le R_S est calculé le risque de cette utilisateur et le comparer au seuil.
H. Le R_S envoie un résultat de comparaison par apport le rôle d'autorisation.
I. Le PDP retourne la décision en s'appuient sur les résultats de R_S et la politique d'accès et l'envoie au PEP
J. PEP remplit les obligations et, en se fondant sur la décision d'autorisation adressé par PDP, soit permet ou interdit l'accès.
K. Le wso2 Identity Server pour test la politique envoyé par l'utilisateur.

3.2.2. Modélisation UML

Les différents diagrammes utilisés en UML donnent tous une vision particulière du logiciel à développer ou system. Ces diagrammes divisés à plusieurs catégories : statique nous avons à représenter avec le diagramme de classe, dynamique avec le diagramme séquence et traitement avec diagramme de cas d'utilisation.

3.2.2.1. Diagramme de cas d'utilisateur

Vue globale de la fonctionnalité de système, est constitué d'une limite de système, d'acteurs et de cas d'utilisation.

Dans cette diagramme de cas d'utilisation on a un utilisateur qui va accéder au fichier médicale il va premièrement passer par la vérification de la politique,

ensuite les politiques qu'il va l'accès ou pas. En cas de délégation si le docteur est absent alors il va déléguer ses droites a une infirmer.

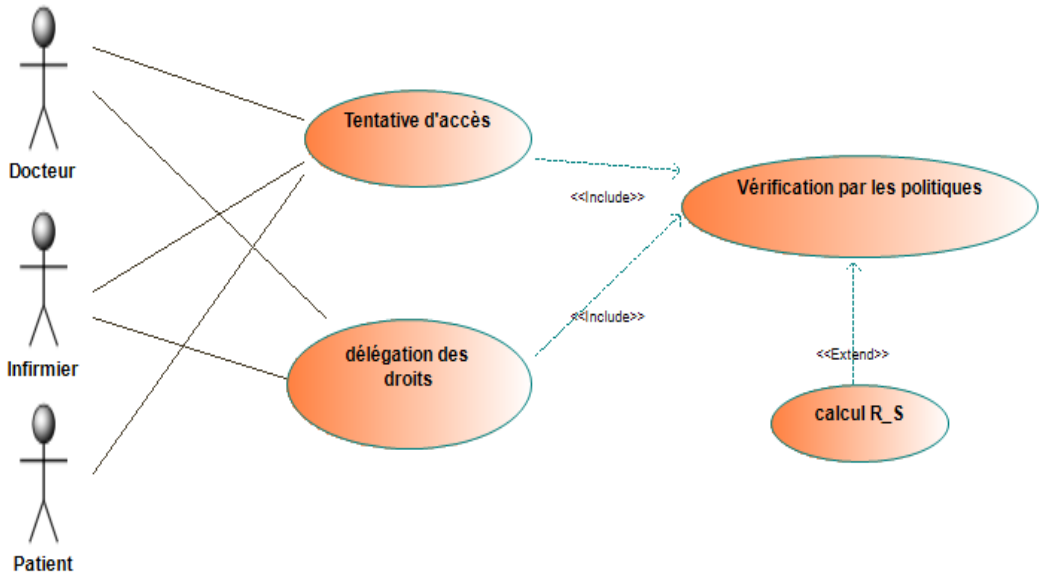

Figure 3.2. Schéma de cas d'utilisation.

3.2.2.2. Diagramme de déploiement

Le Diagramme de déploiement reprisent une vue statique qui sert à représenter l'utilisation de l'infrastructure physique (les serveurs ici) par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux.

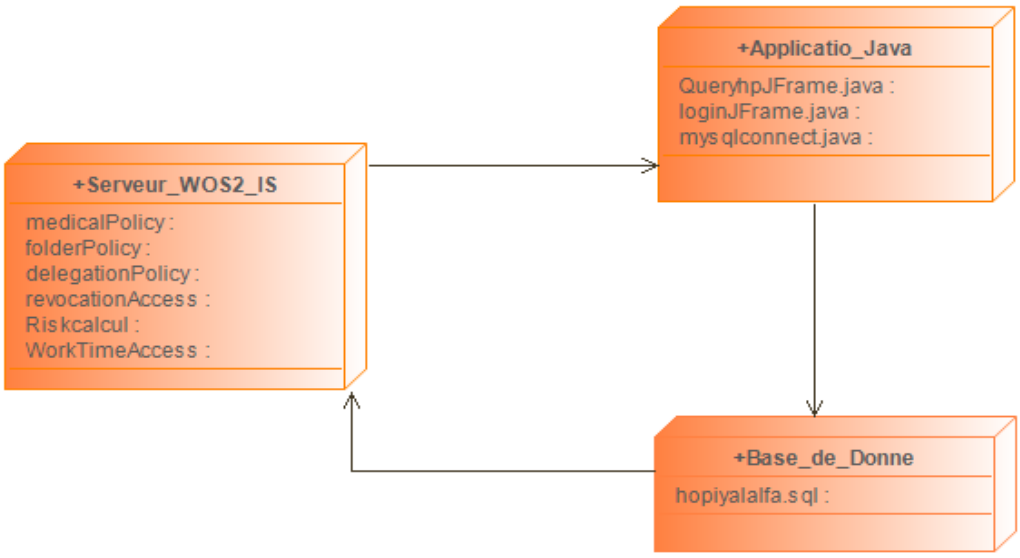

Figure 3.3. Schéma de diagramme de déploiement.

3.2.2.3. Diagramme de séquence

Le diagramme de séquence représente la succession chronologique des opérations réalisées par les acteurs (docteur, infirmier, patient…etc.). Il montre les interactions entre les objets, en montrant les messages qu'ils échangent entre eux ordonnés dans le temps.

- Accès_donné(*) : l'étoile(*) veut dire tous le droits ex ici (lire, écrire, …etc.)
- Car_R_Existe () : teste si y a une relation entre le demandeur d'accès et le dossier médicale.
- Accès_bloqué(*) : y a un blocage et y a pas aucune droits (lire, écrire …etc.) accès.
- Peux_accéder(*) : le demandeur peux accéder au dossier médicale.

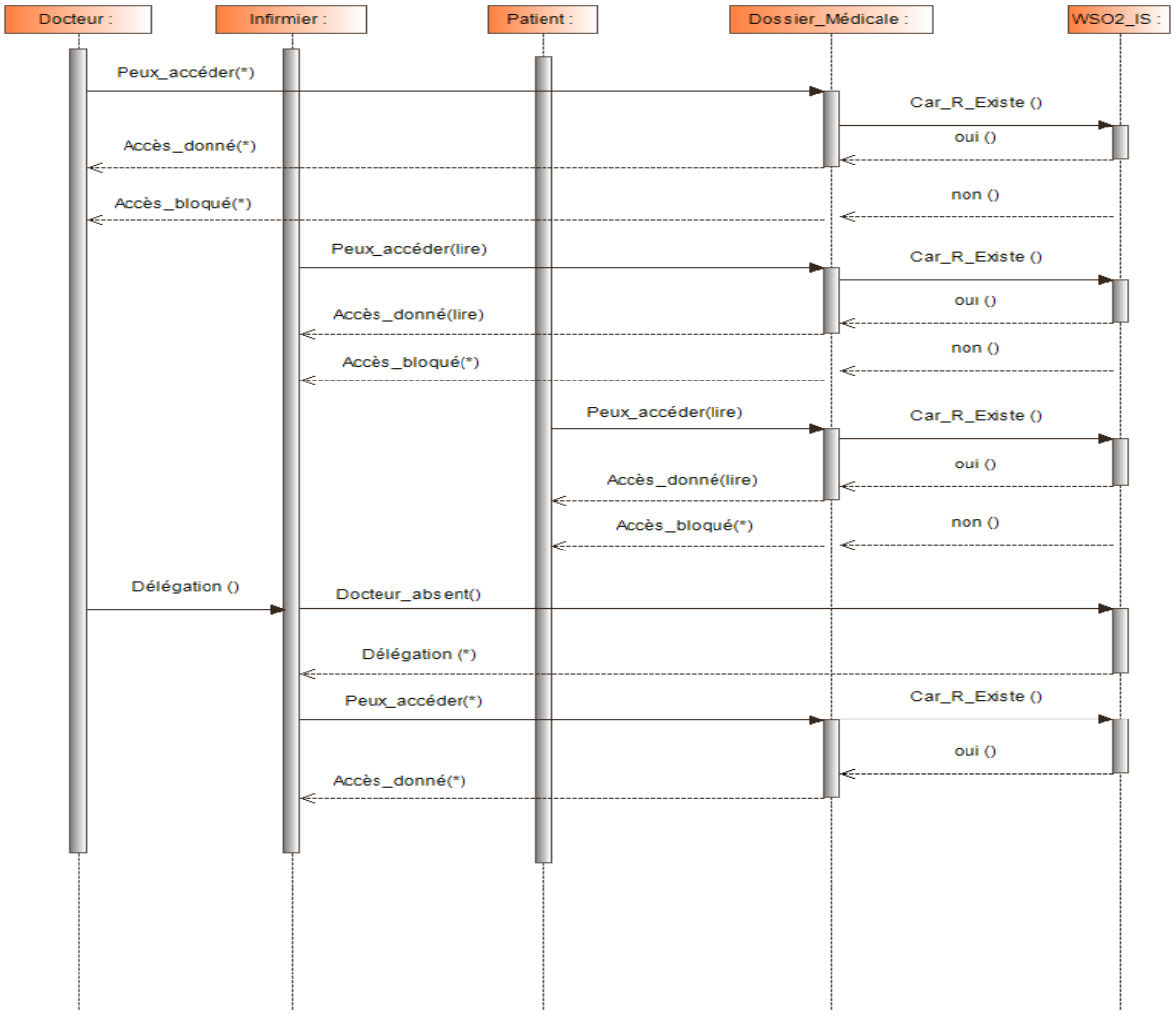

Figure 3.4. Schéma de diagramme de séquence.

On va il y a un diagramme de cas d'utilisation pour travail le serveur (wso2) :

- R_S : Risque Score
- T_S : Trust Score
- (*): tous les droits.

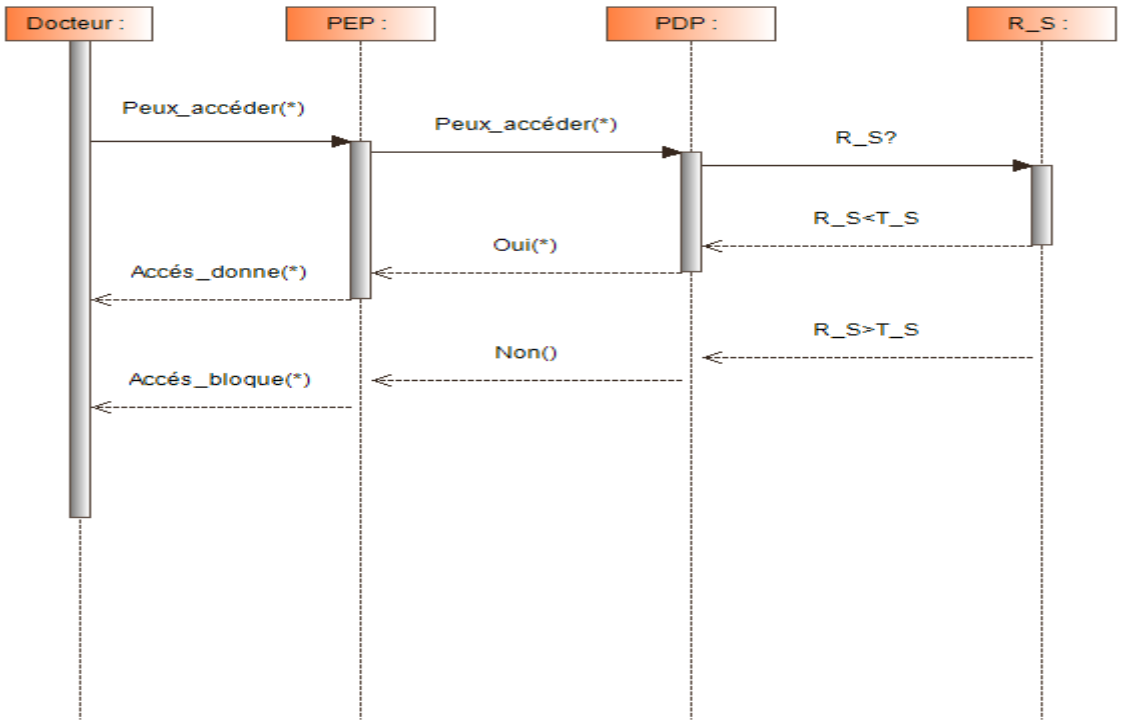

Figure 3.5. Schéma diagramme de séquence de test de la politique avec WSO.

3.2.3. Les tables de la base de données utilisée

Les tables utilisées pour réaliser les besoins de l'application sont deux tables : table d'authentification d'utilisateur avec table de risque de score :

Les attributs	Descriptions	Types
Id	Primary key	Intger
Username	Nom d'utilisateur	String
Password	Mot de passe d'utilisateur	String
Fonctionuser	Le travail d'utilisateur	String

Tableau 3.1: Table d'authentification d'utilisateur.

Les attributs	Descriptions	Types
Id	Primary key	Intger
Username	Nom d'utilisateur	String
Note	Note « qualité de service »	Intger
Notea	Note « Intégrité et disponibilité»	Intger
Notei	Confidentialité des informations	Intger
Notec	Engagement par le temps et les creux	Intger
Reputation	Réputation de hospitale	String
Risquescore	Apré le calcul de risquescore	Intger

Tableau 3.2. Table de risque de score.

3.3. Outils utilisés dans la programmation

3.3.1. Environnement logiciel

3.3.1.1. Modelio :

C'est un outil de modélisation UML disponible sur les plates-formes Windows, Linux et Mac. Il intègre également la modélisation BPMN, et le support de la modélisation des exigences, du dictionnaire, des règles métier et des objectifs. [27]

3.3.2. Outil de développement intégré

3.3.2.1. Eclipse

Eclipse est un IDE, Integrated Development Environment (EDI environnement de développement intégré en français), c'est-à-dire un logiciel qui simplifie la programmation en proposant un certain nombre de raccourcis et d'aide à la programmation. Il est développé par IBM, est gratuit et disponible pour la plupart des systèmes d'exploitation. [29]

Au fur et à mesure que vous programmez, eclipse compile automatiquement le code que vous écrivez, en soulignant en rouge ou jaune les problème qu'il décèle. Il souligne en rouge les parties du programme qui ne compilent pas, et en jaune les parties qui compilent mais peuvent éventuellement poser problème (on dit qu'éclipse lève un avertissement, ou warning en anglais). Pendant l'écriture du code, cela peut

sembler un peu déroutant au début, puisque tant que la ligne de code n'est pas terminée (en gros jusqu'au point-virgule), eclipse indique une erreur dans le code. [29]

3.3.2.2. NetBeans

NetBeans est un environnement de développement intégré (IDE), placé en open source par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License). En plus de Java, NetBeans permet la prise en charge native de divers langages tels le C, le C++, le JavaScript, le XML, le Groovy, le PHP et le HTML, ou d'autres (dont Python et Ruby) par l'ajout de greffons.

Il offre toutes les facilités d'un IDE moderne (éditeur avec coloration syntaxique, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web). Compilé en java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java. [30]

3.3.2.3. WSO2 Identity Server

WSO2 Identity Server est un serveur de gestion des identités et des droits qui facilite la sécurité lors de la connexion et de la gestion de plusieurs identités entre différentes applications.WSO2 Identity Server fournit une gestion sécurisée des identités pour les applications, les services et les API Web d'entreprise en gérant les identités et les droits des utilisateurs de manière sécurisée et efficace.

Identity Server permet aux architectes d'entreprise et aux développeurs de réduire le temps de mise à disposition d'identités, de garantir des interactions sécurisées en ligne et de fournir un environnement de connexion unique réduit. Identity Server nous permet de créer, de gérer et de terminer des comptes d'utilisateurs ainsi que des identités d'utilisateurs sur plusieurs systèmes, y compris les applications Cloud. Lorsque plusieurs applications nécessitent une authentification, les utilisateurs doivent pouvoir se connecter en un seul endroit et bénéficier d'un accès transparent à toutes les autres applications. [31]

3.3.3. Langage de programmation

3.3.3.1. XACML

XACML (eXtensible Access Control Markup Language) est un langage standardisé par OASIS, basé sur XML qui est dédié au contrôle d'accès (Oasis, 2005). Il permet l'expression de politiques selon une approche ABAC.

Dans ce langage, toute entité concernée par le contrôle d'accès (i.e. sujets, ressources, actions et environnement) est spécifiée par un ensemble d'attributs. Le standard inclut également la description d'une architecture qui explique comment un point de décision de politique (PDP) obtient les attributs nécessaires lorsqu'il évalue la politique pour prendre sa décision d'autorisation.

Le langage de politique XACML est utilisé pour décrire les exigences générales de contrôle d'accès en termes de contraintes sur des attributs. Un attribut peut être n'importe qu'elle caractéristiques d'un sujet, d'une action, d'une ressource ou de l'environnement dans lequel la requête d'accès est produite. Le fait de considérer les attributs rend le langage très flexible. De plus, XACML présente des points d'extension standards pour définir de nouveaux types de données, des fonctions additionnelles, des combinaisons de logiques. [32]

3.3.3.2. Java

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Beaucoup d'applications et de sites Web ne fonctionnent pas si Java n'est pas installé et leur nombre ne cesse de croître chaque jour. Java est rapide, sécurisé et fiable. Des ordinateurs portables aux centres de données, des consoles de jeux aux superordinateurs scientifiques, des téléphones portables à Internet, la technologie Java est présente sur tous les fronts. [34]

Le plug-in Java est un composant de l'environnement JRE. Ce dernier permet aux applets écrites en langage de programmation Java d'être exécutées dans différents navigateurs. Le plug-in Java n'est pas un programme autonome et ne peut pas être installé séparément. [33]

3.3.3.3. Xpath

XPath (XML Path Language)est un langage de requête pour localiser une portion d'un document XML. Initialement créé pour fournir une syntaxe et une sémantique aux fonctions communes à XPointer , XPath a rapidement été adopté par les développeurs comme langage d'interrogation simple d'emploi.

XPath joue un rôle d'importation dans XACML lorsque les politiques sont évaluées pour les données XML. Lorsque des données XML sont transmises à travers des nœuds, le PEP peut être un point d'interception qui appelle le PDP avec des données XML d'analyse. Basé sur les données XML, PDP peut prendre des décisions.[35]

3.3.3.4. Axiomatics Language for Authorization (ALFA)

Le langage ALFA (Axiomatics Language for Authorization) est un langage spécifique au domaine pour une description de haut niveau des stratégies XACML. Il est conçu pour la facilité d'utilisation par les développeurs. En outre, il présente des informations spécifiques au domaine tel que les identifiants d'attribut sous forme compacte et il peut être compilé dans XACML 3.0. [37]

3.3.4. Outil de base de données

3.3.4.1. Php MyAdmin

C'est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée principalement en PHP et distribuée sous licence GPL (General Public License). Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, gratuits comme payants, le proposent ce qui évite à l'utilisateur d'avoir à l'installer.

Cette interface pratique permet d'exécuter, très facilement et sans grandes connaissances en bases de données, des requêtes comme les créations de table de données, insertions, mises à jour, suppressions et modifications de structure de la base de données, ainsi que l'attribution et la révocation de droits et l'import/export. Ce système permet de sauvegarder commodément une base de données sous forme de fichier .SQL et d'y transférer ses données, même sans connaître SQL.Les requêtes SQL restent possibles, ce qui permet de les tester interactivement lors de la

création d'un site pour les utiliser ensuite en batch (c'est-à-dire en différé) une fois au point. [36]

3.4. Implémentation

Dans le présent projet, nous avons utilisé de nombreux programmes et langages de programmation, j'ai utilisé ordinateur portable version « dell i5 » avec contient « Windows 10 », En plus de l'application java sur «' eclipse OXYGEN.3 march 2018' ' NetBeans IDE 8.2 RC '», Ensuite on va des langages de programmation « java, XCAML, ALFA », et j'ai utilisé deux serveur « wampserver 64 avec php myadmine pour la base de donnes d'utilisateurs avec le serveur WSO2 IS version 5.9.0 pour teste les politiques de contrôle d'accès ».nous avons également utilisé Le plugin ALFA pour Eclipse IDE afin d'extraire un code XACML 3.0.

3.4.1. Axiomatics Language for Authorization(ALFA)

Dans cette partie nous allons expliquer comment procéder avec ALFA pour écrire le pseudo code qui va nous donner le code XACML 3.0.

Le plugin ALFA pour Eclipse IDE, Axiomatics fournit des moyens plus simples de créer des politiques XACML 3.0 pour aider les développeurs à s'attaquer aux autorisations plus rapidement que jamais auparavant. Le plugin génère des politique XACML 3.0 à partir d'un nouveau langage, ALFA, le langage d'autorisation d'Axiomatics, qui emprunte une grande partie de sa syntaxe et de son apparence aux langages de programmation courants tels que Java et C++.

D'abord, il faut télécharger java JDK ainsi que l'environnement de développement java « Eclipse IDE ». Ensuite on télécharge le plugin ALFA pour Eclipse et on l'ajoute à l'environnement de manière classique, on crée un nouveau projet « Hopitalfin», dans cette projet créer un fichier « Hopital.alfa» le faite que l'on donne l'extension «.alfa» à notre fichier. Eclipse va automatiquement savoir que c'est une politique et va demander d'ajouter la Nature « Xtext » avec la boite de dialogue (Figure 4.5) après nous allons appuyer sur « Yes ».

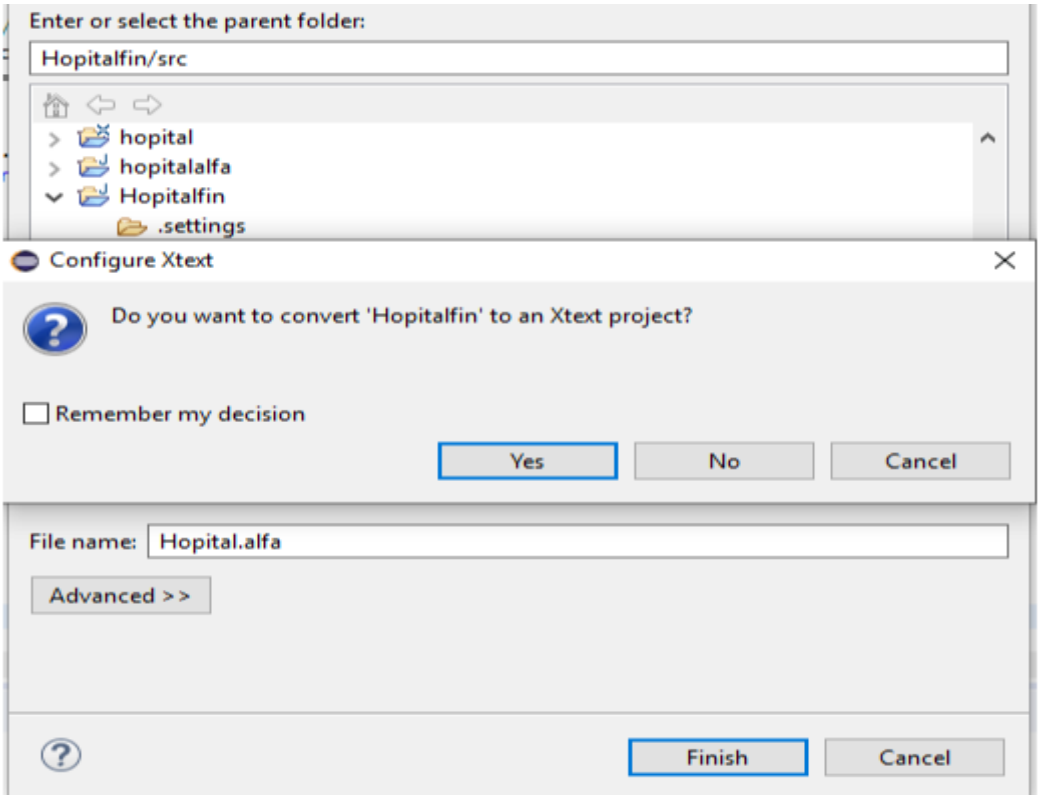

Figure 3.6. Ajoute nature Xtext.

Ensuite on va copier les deux fichiers fondamentaux « system.alfa » et « standard-attributes.alfa » dans le projet « hopital ». le premier fichier pour la distribution ALFA qui contient des définitions pour les fonctions XACML standard. Et le deuxieme fichier qui contient des définitions des identificateurs d'attribut pour les attributs standard de la spécification XACML.

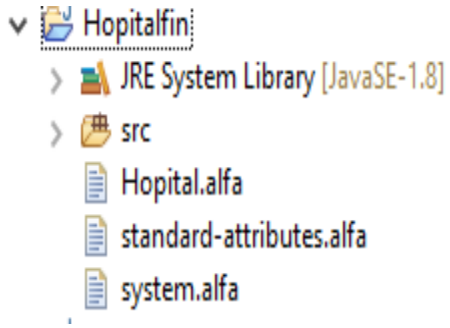

Figure 3.7. Copier les deux fichiers fondamentaux pour alfa.

Donc on commence à taper le début de code ALFA, et on crée les politique, nous avons créé 5 politiques.

3.4.2. Politique utilise

1. **La politique « médicalPolicy » :** cette politique qui permet l'accès seulement si le docteur est assigné à ce malade, ceci est fait en testant l'état dans la condition de la règle avec l'attribut « Car_R_Existe () » cette condition va retourner un booléen (true, false) , si c'est « true » alors le docteur a le droit d'accéder au dossier médical sinon il ne pourra pas et sera dirigé vers la règle « notdoctor » qui affichera un message « there is no care relation » . Le message « there is blocked » sera affiché si le dossier est bloqué pour n'importe quelle raison

```
policy medicalPolicy
 {
  target clause resource.resourceType == "medical-record"
  apply firstApplicable
  rule doctoract{
  permit
  target clause user.role == "Doctor"
  condition (booleanOneAndOnly(resource.careRelationExists))
 }

  rule blockedacces{
  deny

  condition booleanOneAndOnly(resource.recordIsBlocked)
  on deny {
  advice ObligationAdvice.reasonForDeny {
  action.message = "There is blocked!!!!"

     }
          }
 }

//en cas ou le docteur et absent et la nurse est deleger

rule nurseact{
  permit
  target clause user.role == "Nurse"

  condition (booleanOneAndOnly(resource.careRelationExists))
  && (booleanOneAndOnly(resource.delegationRelationExists))
 }
 }
```

Figure 3.8. Code Alfa pour la politique medicalPolicy.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--This file was generated by the ALFA Plugin for Eclipse from Axiomatics AB (http://www.axiomatics.com).-->
<!--Any modification to this file will be lost upon recompilation of the source ALFA file-->
<xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" PolicyId="http://axiomatics.com/alfa/identifier/Hospital.me
      <xacml3:Description>medicalPolicy</xacml3:Description>
        <xacml3:PolicyDefaults>
          <xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</xacml3:XPathVersion>
        </xacml3:PolicyDefaults>
        <xacml3:Target>
           <xacml3:AnyOf>
              <xacml3:AllOf>
                 <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                    <xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">medical-record</xacml3:AttributeValue>
                    <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" Category="urn:oasis:na
                 </xacml3:Match>
              </xacml3:AllOf>
           </xacml3:AnyOf>
         </xacml3:Target>
        <xacml3:Rule Effect="Permit" RuleId="Hospital.medicalPolicy.doctoract">
        <xacml3:Description/>
           <xacml3:Target>
              <xacml3:AnyOf>
                 <xacml3:AllOf>
                    <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                       <xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Doctor</xacml3:AttributeValue>
                       <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" Category="urn:oasi
                    </xacml3:Match>
                 </xacml3:AllOf>
              </xacml3:AnyOf>
           </xacml3:Target>
```

Figure 3.9. Code xacml génère pour la politique medicalPolicy.

L'un des caractéristiques les plus importantes à ALFA, un fichier XACML concernant cette politique est généré automatique, et mis par défaut dans le sous-dossier src-gen comme est indiqué dans la figure dissous

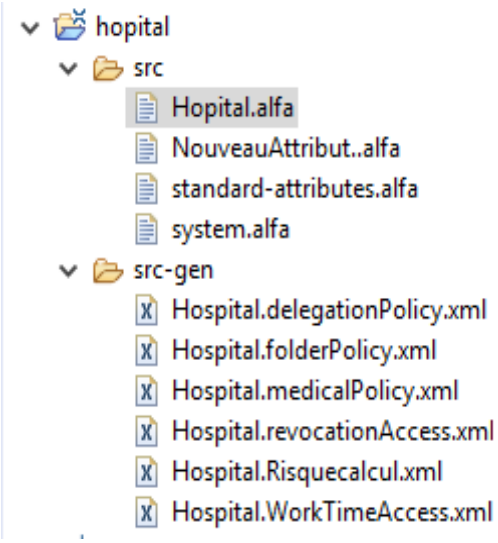

Figure 3.10. Le dossier src-gen.

2. La politique « folderPolicy» : pour traiter les droits sur un fichier médical par exemple un docteur a le droit d'écrire et lire un dossier médical tandis que l'infirmière et le patient n'ont que le droit de lire seulement.

```
policy folderPolicy {
    target clause resource.resourceType == "medical-record"
    apply firstApplicable
rule p{
    permit
target
    clause user.role == "Doctor"

    or user.role == "Nurse"
    or user.role == "Patient"
    clause action.actiontodo == "read"

    }

    rule p1{
        permit
        target clause user.role == "Doctor"
        clause action.actiontodo == "write"
    }
    rule d{
        deny
        }
}
```

Figure 3.11. Code alfa pour La politique folderPolicy.

```
<xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17"
               PolicyId="http://axiomatics.com/alfa/identifier/Hospital.folderPolicy"
               RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:first-applicable"
               Version="1.0"><xacml3:Description>folderPolicy</xacml3:Description>
               <xacml3:PolicyDefaults>
               <xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</xacml3:XPathVersion>

               </xacml3:PolicyDefaults><xacml3:Target><xacml3:AnyOf><xacml3:AllOf>
               <xacml3:Match
               MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <xacml3:AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">medical-record</xacml3:AttributeValue>
               <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                                           Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                                           DataType="http://www.w3.org/2001/XMLSchema#string"
                                           MustBePresent="false"/>
                                           </xacml3:Match></xacml3:AllOf></xacml3:AnyOf>
                                           </xacml3:Target><xacml3:Rule
                                                  Effect="Permit"
                                                  RuleId="Hospital.folderPolicy.p">
  <xacml3:Description/>
  <xacml3:Target>
  <xacml3:AnyOf><xacml3:AllOf>
  <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
  <xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Doctor</xacml3:AttributeValue>
  <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                              Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                              DataType="http://www.w3.org/2001/XMLSchema#string"
                              MustBePresent="false"/>
```

Figure 3.12. Code xacml génère pour la politique folderPolicy.

3. **La politique « délégationPolicy » :** cette politique est très importante et nécessaire, car grace à cette politique, il y a toujours une personne responsable qui remplace en l'absence d'un des officiels. On va s'appliquer sur l'infirmière dans notre exemple, si seulement si le docteur chargé du patient est absent ce dernier devrait être remplacé par l'infermière adéquate, et va lui déléguer ses droits comme ici lire et écrire dans le dossier médical de son patient. Cette politique est essentielle et cruciale car grâce à la politique de délégation le travail.

```
policy delegationPolicy {
   target clause resource.resourceType == "medical-record"
   apply permitOverrides
rule delegation {
   permit
   condition booleanOneAndOnly(resource.doctorIsAbsent)

target
   clause  user.role == "Nurse"
   clause  action.actiontodo == "read" and action.actiontodo == "write"
   } }
```

Figure 3.13. Code alfa La politique délégationPolicy.

```
<xacml3:Description>delegationPolicy</xacml3:Description>
<xacml3:PolicyDefaults>
<xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</xacml3:XPathVersion>
 </xacml3:PolicyDefaults><xacml3:Target>
 <xacml3:AnyOf><xacml3:AllOf>
 <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
 <xacml3:AttributeValue
  DataType="http://www.w3.org/2001/XMLSchema#string">medical-record</xacml3:AttributeValue>
  <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                              Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource"
                              DataType="http://www.w3.org/2001/XMLSchema#string"
                              MustBePresent="false"/></xacml3:Match></xacml3:AllOf>
   </xacml3:AnyOf></xacml3:Target>
   <xacml3:Rule Effect="Permit"
                RuleId="Hospital.delegationPolicy.delegation">
                <xacml3:Description/><xacml3:Target>
                <xacml3:AnyOf><xacml3:AllOf>
                <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                <xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Nurse</xacml3:AttributeValue>
                <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
                                            Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                                             DataType="http://www.w3.org/2001/XMLSchema#string"
                                             MustBePresent="false"/></xacml3:Match></xacml3:AllOf>
              </xacml3:AnyOf><xacml3:AnyOf>
              <xacml3:AllOf>
              <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal"><xacml3:AttributeValue
             DataType="http://www.w3.org/2001/XMLSchema#string">read</xacml3:AttributeValue>
              <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                                          Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
                                          DataType="http://www.w3.org/2001/XMLSchema#string"
                                          MustBePresent="false"/>
              </xacml3:Match><xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">write</xacml3:AttributeValue><
```

Figure 3.14. Code xacml génère pour la politique délégationPolicy.

4. **La politique «WorkTimeAccess » :** c'est une politique qui permet l'autorisation d'accès ou non sur le dossier médical selon le calendrier des travaux.

```
/************************ WorkTimeAcces *********************/


policyset WorkTimeAccess{
   target clause resource.resourceType == "medical-record"
       apply firstApplicable
   /**
    * working hours hospital
    */
      policy denyOutHospital{

         apply firstApplicable
         rule p1{
         permit
         target clause user.role == "nurse"
         and user.role =="Doctor"

        condition (booleanOneAndOnly(resource.careRelationExists))
    }
         rule denyBefore8am{
         target clause currentTime<"08:00:00":time and currentTime >"16:00:00":time
         deny
      }
    }
 }
```

Figure 3.15. Code Alfa pour la politique WorkTimeAccess.

```
<xacml3:Description>WorkTimeAcces</xacml3:Description>
   <xacml3:PolicySetDefaults>
      <xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</xacml3:XPathVersion>
   </xacml3:PolicySetDefaults>
   <xacml3:Target>
      <xacml3:AnyOf>
         <xacml3:AllOf>
            <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">medical-record</xacml3:AttributeValue>
               <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" Category="urn:oasis:names.
            </xacml3:Match>
         </xacml3:AllOf>
      </xacml3:AnyOf>
   </xacml3:Target>
<xacml3:Policy PolicyId="http://axiomatics.com/alfa/identifier/Hospital.WorkTimeAccess.denyOutHospital" RuleCombiningAlgId="urn:oasi
   <xacml3:Description>working hours hospital</xacml3:Description>
      <xacml3:PolicyDefaults>
         <xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</xacml3:XPathVersion>
      </xacml3:PolicyDefaults>
      <xacml3:Target/>
         <xacml3:Rule Effect="Deny" RuleId="Hospital.WorkTimeAccess.denyOutHospital.denyBefore8am">
            <xacml3:Description/>
       <xacml3:Target>
          <xacml3:AnyOf>
             <xacml3:AllOf>
                <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:time-greater-than">
                   <xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">08:00:00</xacml3:AttributeValue>
                   <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
```

Figure 3.16. Code xacml génère pour la politique WorkTimeAccess.

5. La politique « revocationPolicy » : c'est une politique qui révoque le droit d'accès sur une ressource donnée en se basent sur des critères préalablement définis ou calculés en temps réel.

```
policy revocationAccess {
    apply denyOverrides
    rule revocation {
    deny
    condition resource.risquescores > resource.trustscore

target  clause
    user.role == "Nurse"
    or user.role =="Doctor"
    clause action.actiontodo == "read"
     and action.actiontodo == "write"

    on deny {
    advice ObligationAdvice.reasonForDeny {
    action.message = "You are not trustworthy"
}}
    }
}
```

Figure 3.17. Code Alfa pour la politique revocationPolicy.

```
<xacml3:Policy xmlns:xacml3="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" PolicyId="http://axiomatics.com/alfa/identifier/Hospital.re
   <xacml3:Description>RevocationPolicy</xacml3:Description>
      <xacml3:PolicyDefaults><xacml3:XPathVersion>http://www.w3.org/TR/1999/REC-xpath-19991116</xacml3:XPathVersion>
      </xacml3:PolicyDefaults>
      <xacml3:Target>
         <xacml3:AnyOf>
            <xacml3:AllOf>
              <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                  <xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">medical-record</xacml3:AttributeValue>
                  <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" Category="urn:oasis:names
              </xacml3:Match>
            </xacml3:AllOf>
         </xacml3:AnyOf>
       </xacml3:Target>
    <xacml3:Rule Effect="Deny" RuleId="Hospital.revocationAccess.revocation">
    <xacml3:Description/>
    <xacml3:Target>
       <xacml3:AnyOf>
          <xacml3:AllOf>
             <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                <xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Nurse</xacml3:AttributeValue>
                <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" Category="urn:oasis:names:tc
             </xacml3:Match>
             <xacml3:Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                <xacml3:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">Doctor</xacml3:AttributeValue>
                <xacml3:AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" Category="urn:oasis:names:tc
             </xacml3:Match>
```

Figure 3.18. Code xacml génère pour la politique revocationPolicy.

Nous avons toujours la ressource le dossier médicale du patient, le docteur et l'infirmière ont un attribut ressource « risquescores » qui est calculé à partir des points négatifs et positifs attribués selon le retard, le comportement avec les patient, l'assiduité, le travail rigoureux et la disponibilité … tandis que le «trustscore » est un seuil fixer au préalable par l'administrateur, PAP ici.

Si le « risquescores » dépasse le « trustscore » tous les droit d'accès sont révoque et un message est afficher pour avertir la personne qui tente d'accéder de la cause de son blocage.

Cree un fichier « NouveauAttribut.alfa » et ajoutes Tous les attributs utiles nécessaires sont déclarés dans cette fichier.

Hopital.alfa | NouveauAttribut..alfa

```
40
41    namespace resource {
42
43         attribute resourceType {
44        id = "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
45        type = string
46        category = resourceCat
47    }
48
49     attribute careRelationExists {
50        id = "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
51        type = boolean
52        category = resourceCat
53    }
54
55
56     attribute delegationRelationExists {
57        id = "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
58        type = boolean
59        category = resourceCat
60    }
61
62     attribute recordIsBlocked {
63        id = "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
64        type = boolean
65        category = resourceCat
66    }
67
68     attribute doctorIsAbsent {
69        id = "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
70        type = boolean
71        category = resourceCat
72
73    }
```

Figure 3.19. Le fichier "NouveauAttribut.alfa".

3.4.3. Le PEP

Le PEP (policy enforcement point) point d'application de la politique sera modélisé par une application java.

Il y a deux interfaces la première interface pour Authentification. L'utilisateur doit entrer son nom d'utilisateur et son mot de passe pour continuer ses fonctions.

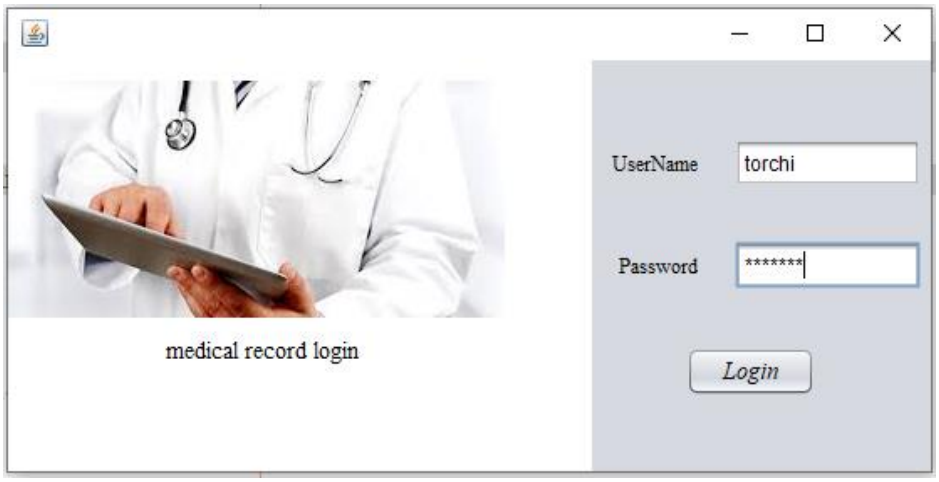

Figure 3.20. Authentification d'utilisateur.

Après avoir vérifié le nom d'utilisateur et le mot de passe, l'utilisateur est autorisé à effectuer la deuxième interface qui permettant aux utilisateurs de noter les infirmiesr/infermières et le service d'hôpital (Figure 4.15), En se basant sur des métriques, l'application va calculer une moyenne (Figure 4.16) et la stocker en une base de données pour être utilisée ultérieurement par la politique (par ex, la politique de révocation) afin de déterminer une décision d'accès.

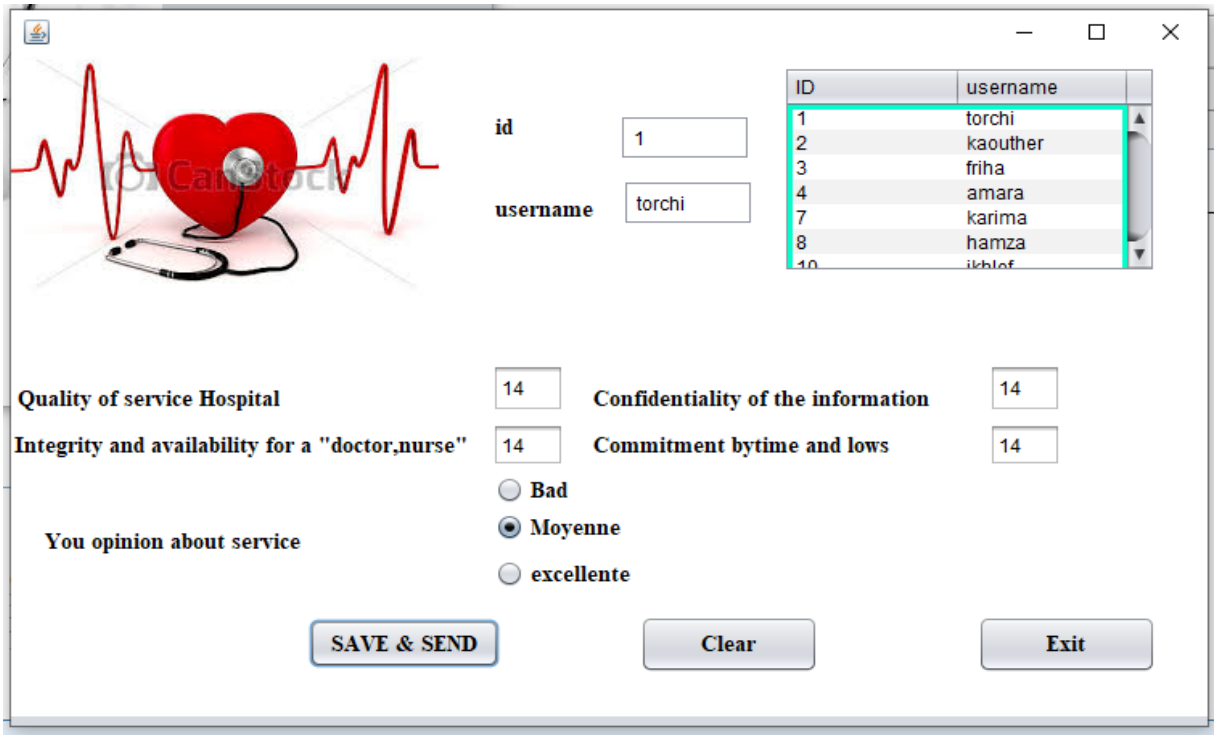

Figure 3.21. Interface PEP.

```
private void cmdsaveActionPerformed(java.awt.event.ActionEvent evt) {

    rs1 = Float.parseFloat(tnote.getText());
    rs2 = Float.parseFloat(tnotea.getText());
    rs3 = Float.parseFloat(tnotei.getText());
    rs4 = Float.parseFloat(tnotec.getText());
   float rs= ((rs1+rs2+rs3+rs4)/4);
   String sql = "insert into risquescore (ID,username,note,notea,notei,"
            + "notec,réputation,risqueScore) "
            + "values ( ?, ?, ?, ?, ?, ?, ?, ?)";
   try{
        pst=conn.prepareStatement(sql);
        pst.setString(1, tid.getText());
        pst.setString(2, tuser.getText());
        pst.setString(3, tnote.getText());
        pst.setString(4, tnotea.getText());
        pst.setString(5, tnotei.getText());
        pst.setString(6, tnotec.getText());
        pst.setString(7, answergl);
        pst.setFloat(8, rs);
        pst.execute();
         JOptionPane.showMessageDialog(null, "saved succesfuly "
                 + "***Thank you for your Evaluationl***");
```

Figure 3.22. Code source pour calcul du score de risque.

3.4.4. WSO2 Identity Server

Dans cette section nous allons présenter le Serveur d'identité WSO 2 IS, ce serveur nous permet de tester les politiques générés avec ALFA. Il existe également de nombreux serveurs et le plus important d'entre eux est XRay (un serveur de test pour les politiques générées avec ALFA) est un serveur plus adéquat et fluide que WSO2IS, donc XRay a développé par Axiomatics INC mais il est payant et inaccessible comparé à WSO2IS.

Pour commencer on démarre WSO2 Identity Server « wso2server.bat » dans le répertoire /bin, il faut se connecter à la console de gestion avec le user name Admin et le mot de passe admin.

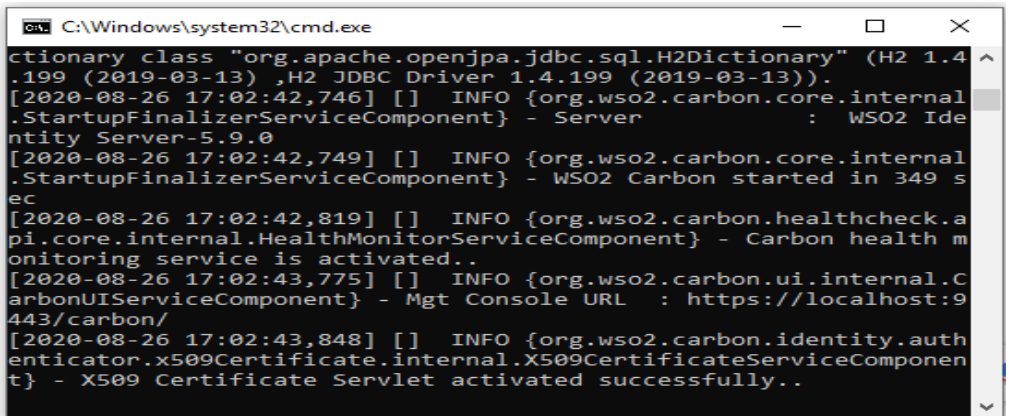

Figure 3.23. Lancement de serveur WSO2-IS.

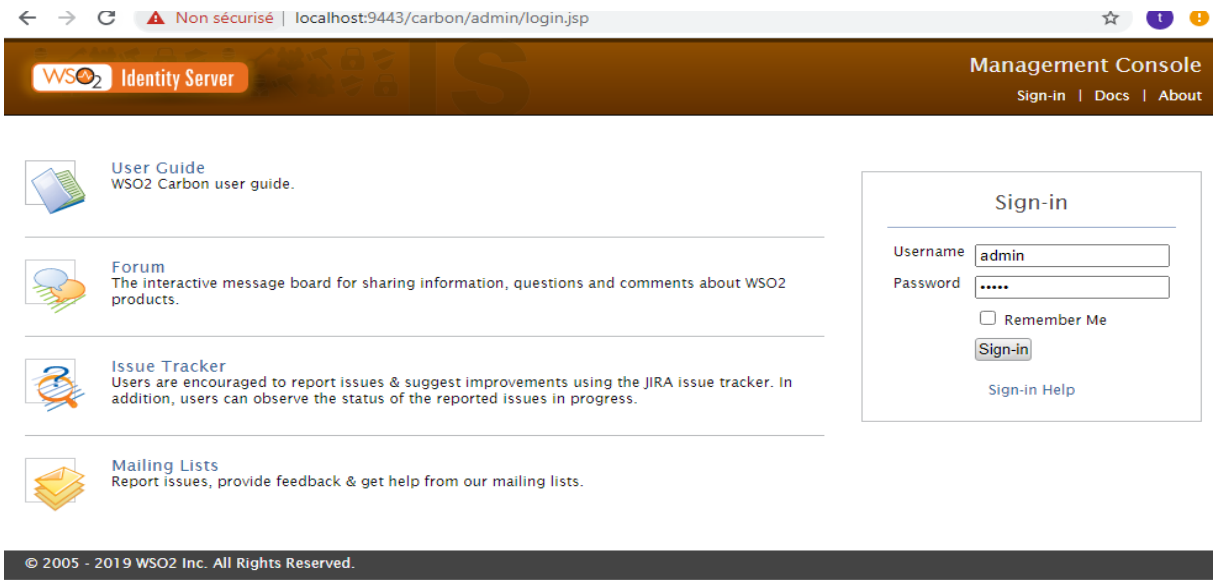

Figure 3.24. Page d'accueil WSO2-IS.

Après vous être connecté à ce serveur, nous allons a accède à « Policy Administration » dans le menu Principal et on clique sur Ajouter une nouvelle politique « Add New Entitilement Policy ».

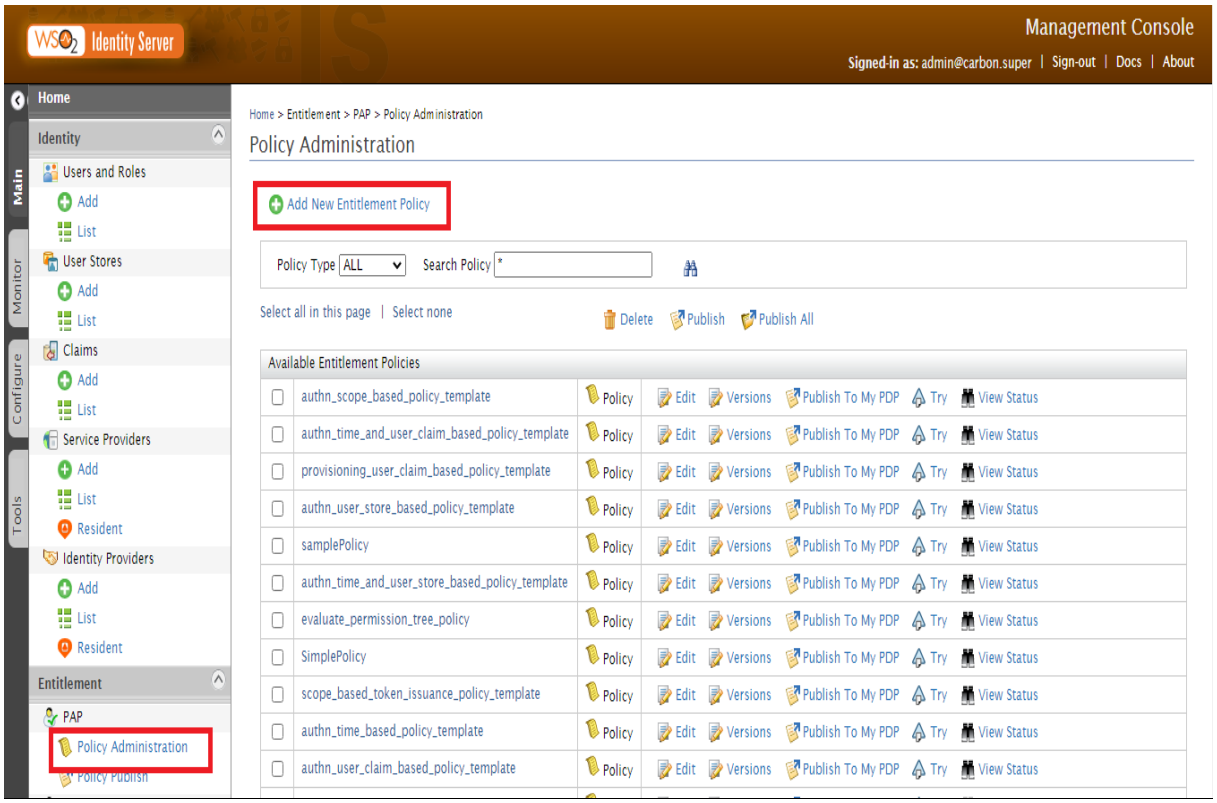

Figure 3.25. La page principale de PAP.

On Clique alors sur « add New Entitement »Policy et après on clique sur Importer une politique existante ou écrire une politique en XML « write Policy in XML » pour ajouter la politique.

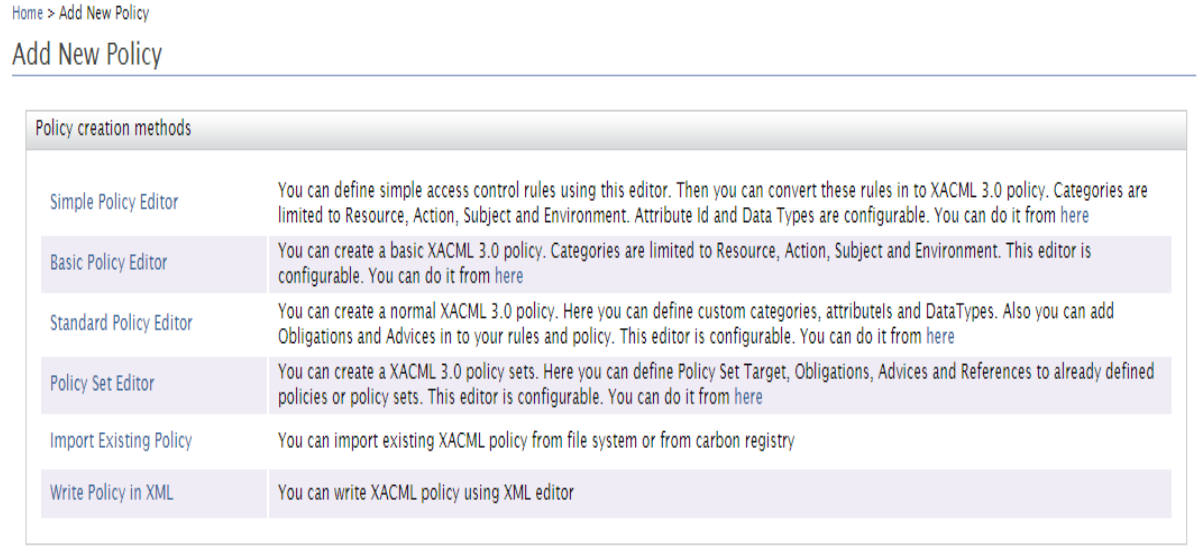

Figure 3.26. Interface pour choisir le choix d'ajoute d'une nouvelle politique.

Si on choisit l'option écrire, un éditeur s'ouvrira comme suit :

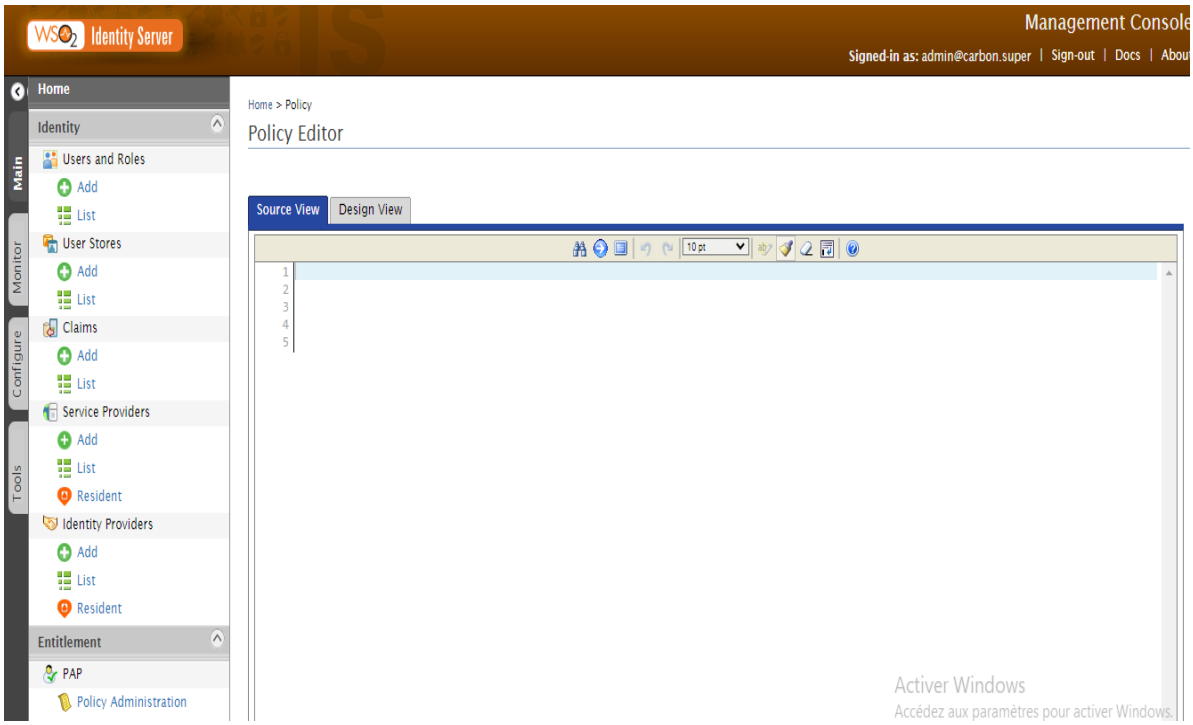

Figure 3.27. Éditeur de politique XML.

Sinon, on clique sur importer, on pourra importer la politique ; dans notre cas, nous allons importer la politique que nous avons générée par ALFA.

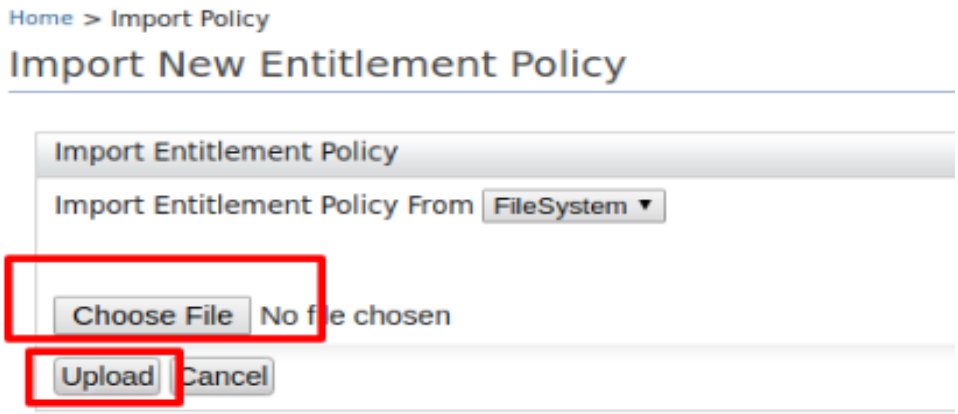

Figure 3.28. Interface d'importation de la politique.

Et après, le message sera affiché qui indique qu'on réussit dans l'ajout de notre politique

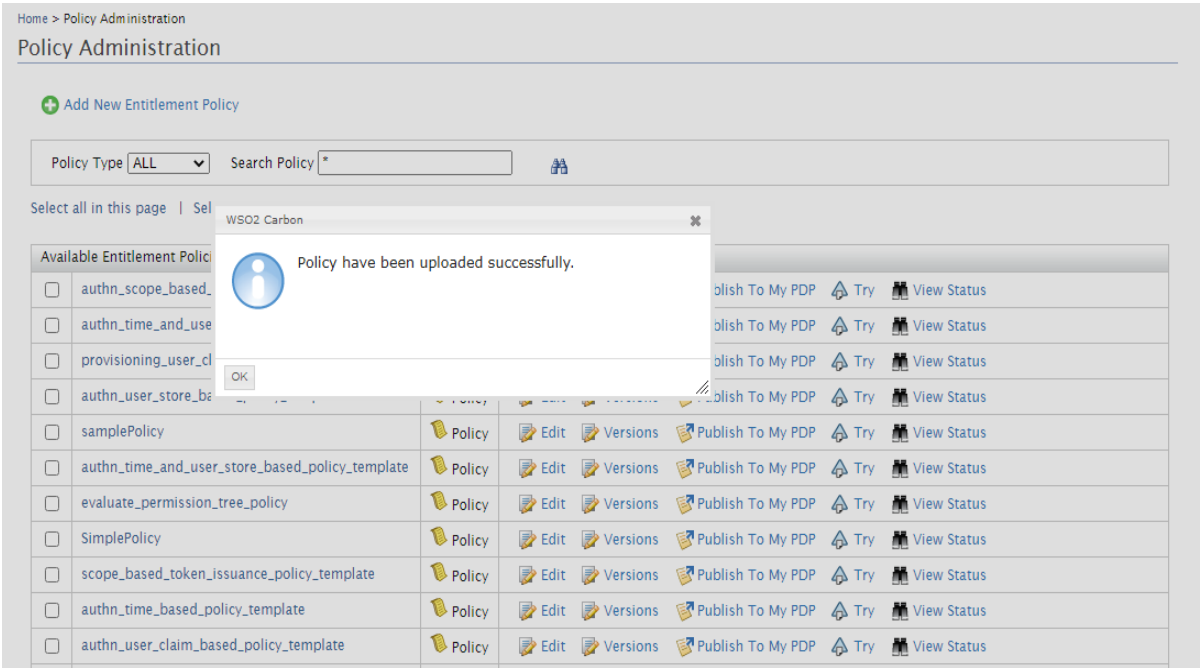

Figure 3.29. Message de réussir dans l'ajout.

Apres nous allons tester notre politique, en cliquant sur « try », Ensuite on remplit les attributs de la requête dans le formulaire « TryIt ».

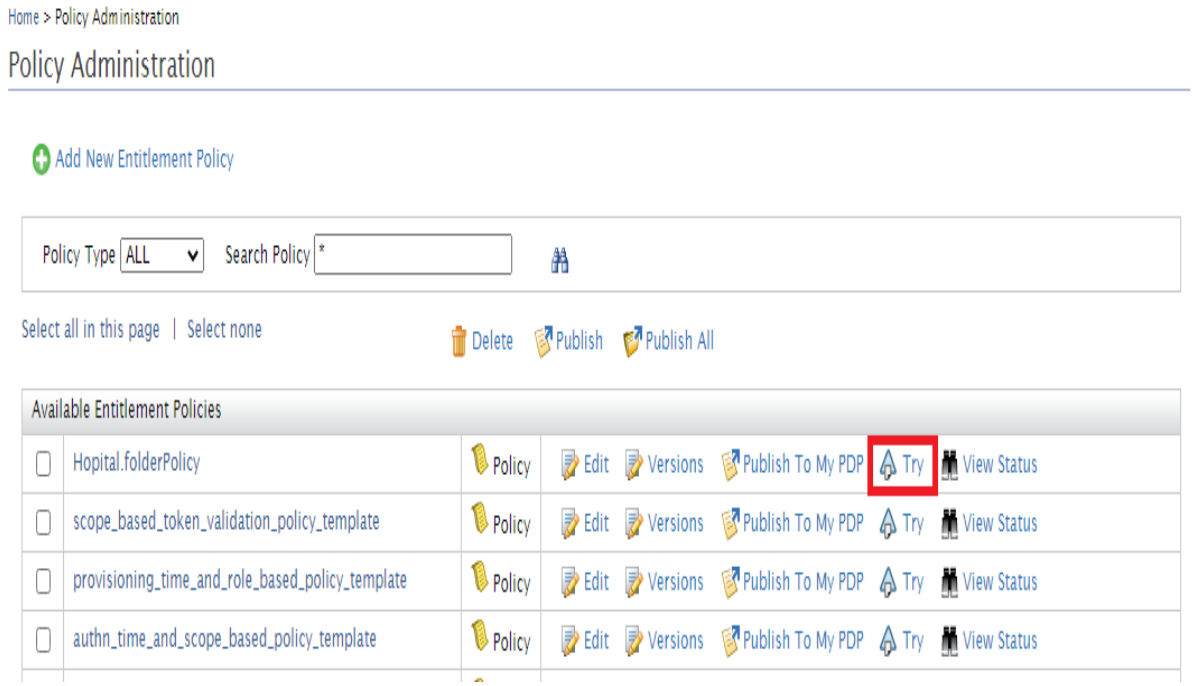

Figure 3.30. Interface montrant notre politique importe avec succès.

Nous allons générer une requête qui teste si un infirmièr au droit de lire un dossier médical du patient assigné, la ressource est dossier médicale «Hopital.folderPolicy».

Le sujet est l'infirmièr et l'action est écrire ; enfin en appuie sur « Test Evaluate » pour voir le résultat.

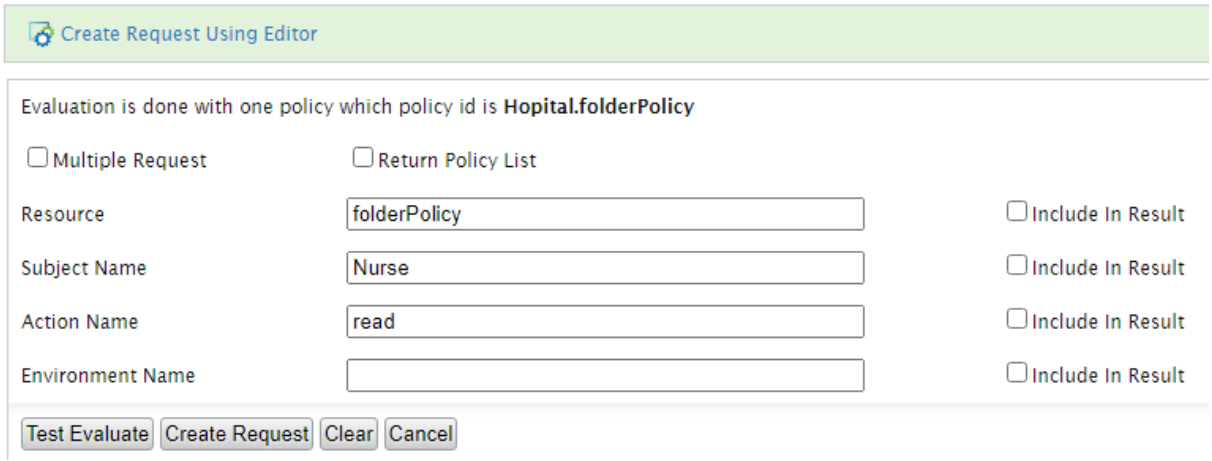

Figure 3.31. Fenêtre de l'éditeur de requête "TryIT".

Si le résultat de test evaluate est correct la réponse attendue sera affiche « permit »

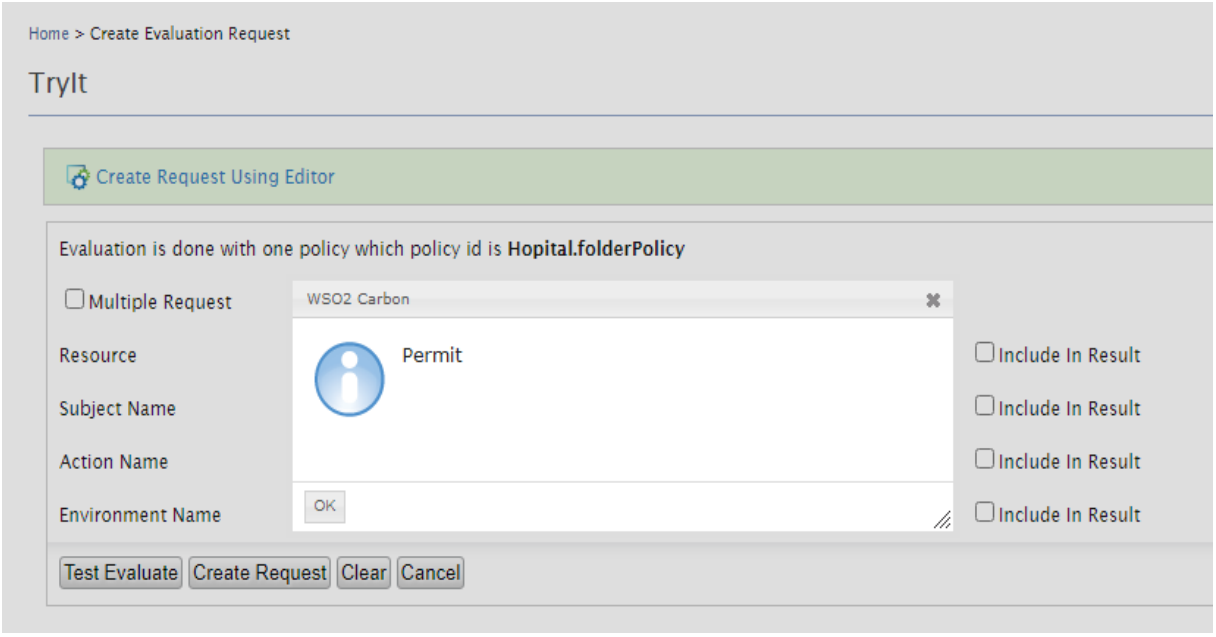

Figure 3.32. Résultat de Test Evaluate correct.

Sinon le résultat de test evaluate est incorrect la réponse attendue sera affiche « deny » comme l'infirmièr n'est pas écrire dans dossier médical

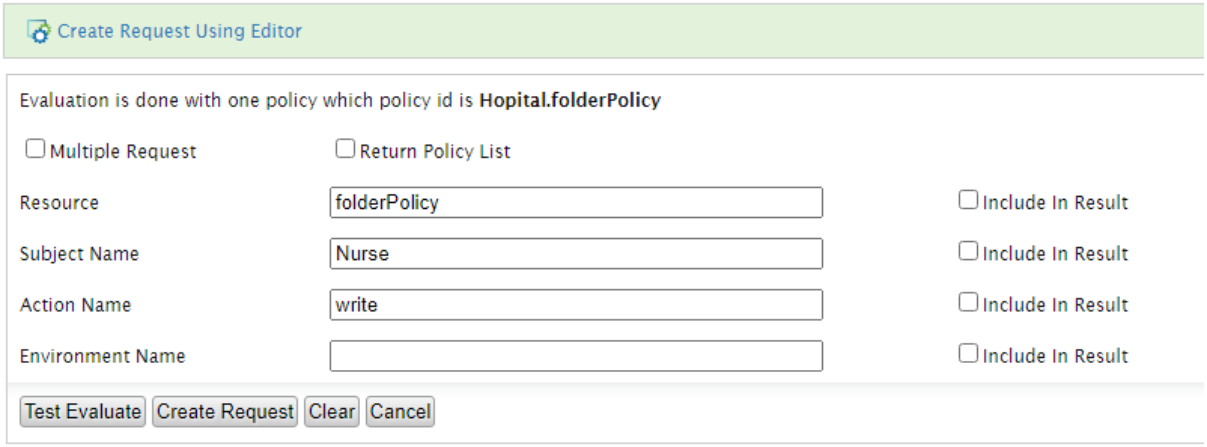

Figure 3.33. Test Evaluate incorrect.

Figure 3.34. Le message affiche dans la requête incorrecte.

Aprés le Test Evaluate on va génère un code XML pour cette requête.

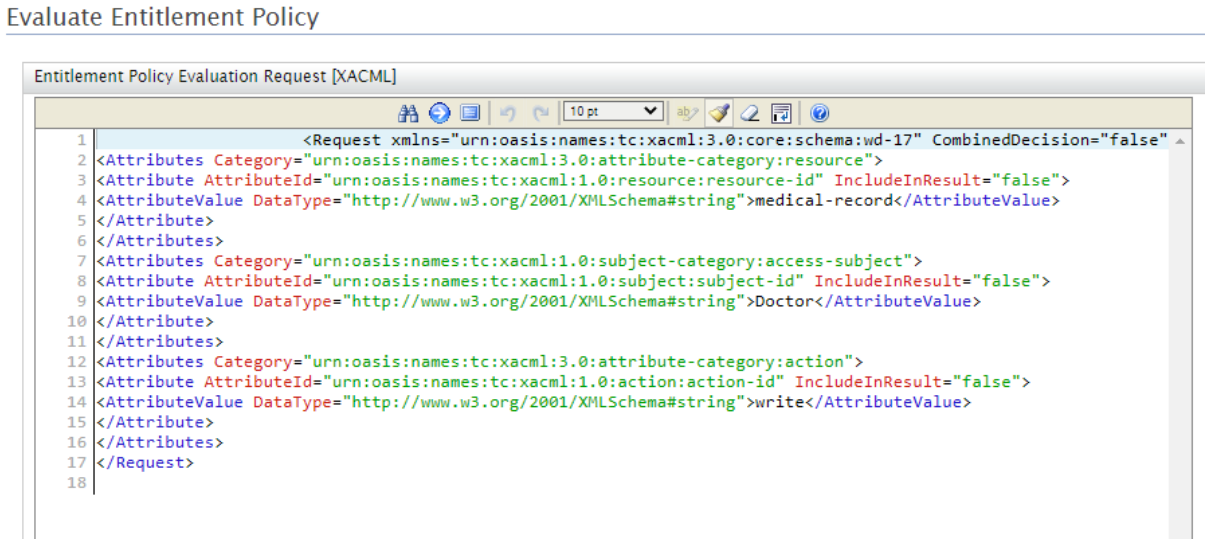

Figure 3.35. Code XML généré pour la requête.

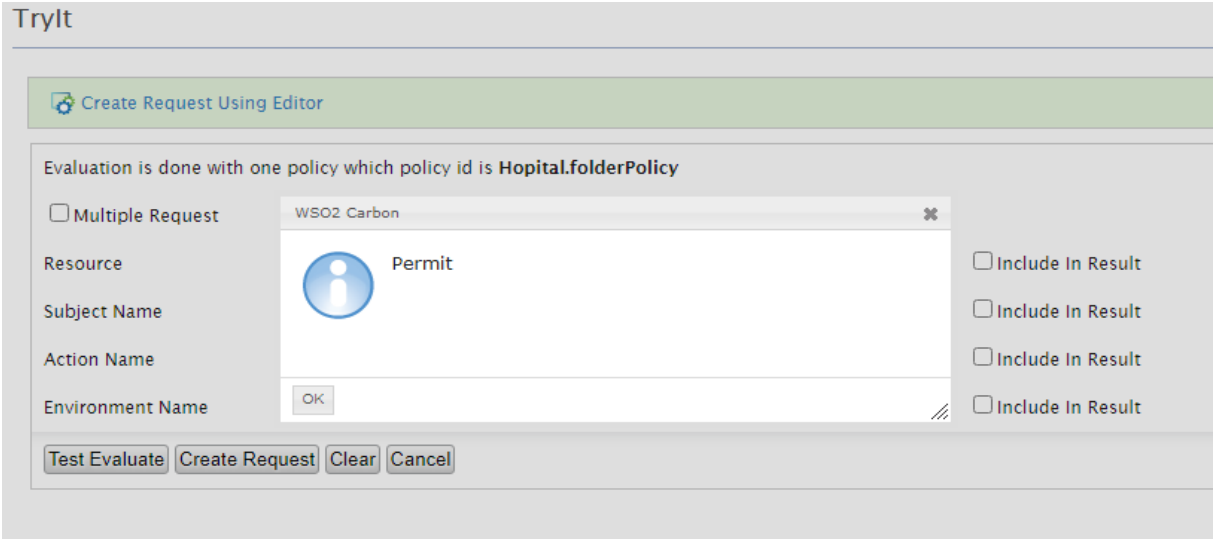

Figure 3.36. Résultat de la requête.

3.5. Conclusion

Dans ce chapitre nous avons présenté le fonctionnement de notre architecture des politiques de sécurité qui repose sur les modèles RBAC et ABAC une politique ainsi que des outils et des différents environnements de développement que nous avons utilisé pour réaliser notre projet, aussi nous avons présenté et expliquer quelques captures d'écran et des fragments du code source.

Conclusion Générale

Bien que des problèmes tels que le manque de responsabilité de délégation et l'accès illimité à une ressource (pas de révocation des droits d'accès) ont été traités en partie dans différents cadres, mais tous ces problèmes reste un point de débat et de recherche et restera pour au moins la décennie qui vient.

Dans ce projet nous avons proposé l'un des solutions convenable pour gérer certain problème majeurs que se pose dans le domaine des politique de contrôle d'accès Nous avons également mis en évidence les lacunes des cadres et des modèles existants.

Dans cette mémoire on a avons discuté brièvement dans le chapitre 1 sur le cloud computing puis sur la sécurité informatique. Ensuite, Dans le chapitre 2 nous avons présenté les modèles de contrôle d'accès. Puis, nous avons proposé deux politiques de contrôle d'accès basé sur les rôles et attributs avec de délégation qui permet déléguer l'accès. , on a terminé avec quelque domaine d'application. Et finalement dans le chapitre 3 on a terminé par une présentation de notre architecture de politique de sécurité qui repose sur les deux modelés RBAC et ABAC une politique ainsi que des outils et des différents environnements de développement que nous avons utilisé pour réaliser notre projet, aussi nous avons présenté et expliquer quelques captures d'écran et des fragments du code source.

Nous préconisons pour un travail futur de contrôle d'accès et délégation auto-adaptatif c'est-à-dire un système autonome.

Bibliography

[1] Labib Terrissa, «Informatique Mobil et Nuagique»,Université Mohamed Khider-Biskra, 2018.

[2] « https://blog.3li.com/cloud-les-modeles-de-deploiement/», consulté le 28 novembre 2019.

[3] M.KHEDIM ALLAH AMINE, M.KHEDIM ALLAH AMINE « CLOUD COMPUTING: Application aux systèmes Mobiles et Pair à Pair », Université Abderrahmane Mira de Béjaïa, 2012.

[4] Mlle. BENDIAB GUELTOUM, « Sécurité des applications métiers au niveau du Cloud Computing: Contrôle d'accès au niveau des APIs du Cloud Computing », Université Abdelhamid Mehri – Constantine 2, 14 mai 2015.

[5] Christian Damsgaard JENSEN, «Un modèle de contrôle d'accès générique et sa réalisation dans la mémoire virtuelle répartie unique Arias », Université Joseph-Fourier - Grenoble I, 1999.

[6] Saïda Medjdoub, « Modèle de contrôle d'accès pour XML :"Application à la protection des données personnelles"», Université de Versailles Saint-Quentin-en-Yvelines, 8 décembre 2005.

[7] Odile PAPINI, « Contrôle d'accès », Université de la méditerranée.

[8] KHELIFA Nor Eddine ,«Intégration du modèle de contrôle d'accés RBAC (Role Based Access control) dans les diagrammes UML(Cas d'utilisation et Séquence) », Unversité d'oran.

[9] « Guide de gestion des accès logiques», le Sous-secrétariat du dirigeant principal de l'information et produite en collaboration avec la Direction des communications, Novembre 2016.

[10] « https://fr.wikipedia.org/wiki/Cloud_computing#Services », consulte 30 avril

[11] Timothy Grance, Peter Mell. «The NIST definition of cloud computing». National Institute of Standards and Technology, Septembre 2011.

[12]« https://www.researchgate.net/publication/322945138_Flexible_attribute_enriched_ role_based_access_control_model »,consulte 14 juin 2020.

[13] Hui Qi, Hongxin Ma, Jinqing Li and Xiaoqiang Di, « Access control model based on role and attribute and its applications on space-ground integration networks » 2015 4th

International Conference on Computer Science and Network Technology (ICCSNT), Harbin, 2015, pp. 1118-1122, doi: 10.1109/ICCSNT.2015.7490931.

[14] A. Younis Y, et al, «An access control model for cloud computing», Journal of Information Security and Applications ,2014.

[15] Mlle. BENDIAB GUELTOUM, « Sécurité des applications métiers au niveau du Cloud Computing : Contrôle d'accès au niveau des APIs du Cloud Computing », Université Abdelhamid Mehri – Constantine 2, 14 mai 2015.

[16]« https://fr.wikipedia.org/wiki/Contr%C3%B4le_d%27acc%C3%A8s_logique », consulté le 2 mars 2020.

[17] OMAR ABAHMANE, « Contrôle de flux d'informations basé sur la granularité », Université du Québec en Outaouais, Octobre 2015.

[18] ABAKAR Mahamat Ahmat, « Étude et mise en œuvre d'une architecture pour L'authentification et la gestion de documents numériques certifiés », université Jean Monnet de Saint-Étienne, 22 novembre 2012.

[19] Bruno Guay, « GIA et sécurité des applications», Symposium GIA ,1 mars 2017.

[20]« https://en.wikipedia.org/wiki/Attribute-based_access_control», consulté le 21 mars 2020.

[21] Pham, Quan, et al. « On a taxonomy of delegation», computers & security,2010.

[22] Boukhlouf Djemaa, « Cours Sécurité des systèmes d'Information et Web »,Université Mohamed Kheider Biskra, 2019/2020.

[23] « https://www.universalis.fr/encyclopedie/reseaux-informatiques/6-securite-dans-les-reseaux/ », consulte 3 mai 2020

[24] Marwan Cheaito, « Un cadre de spécification et de déploiement de politiques d'autorisation », L'université Toulouse III- Paul Sabatier, 9 mars 2012.

[25] Guillaume HARRY, « Gestion des identités et des accès»,12 septembre 2013.

[26] A. Younis Y, et al, «An access control model for cloud computing», Journal of Information Security and Applications ,2014.

[27] « https://fr.wikipedia.org/wiki/Modelio », consulté le 25 juin 2020.

[28] «https://blog.3li.com/cloud-les-modeles-de-deploiement/»,consulté le 15 Avril 2020.

[29] « http://dept-info.labri.fr/ENSEIGNEMENT/programmation2/intro-eclipse/ », consulté le 10 avril 2020.

[30] « https://fr.wikipedia.org/wiki/NetBeans », consulté le 10 avril 2020.

[31] « https://docs.wso2.com/display/IS570/Architecture», consulté le 10 avril 2020.

[32] Romain Laborde, Thierry Desprats, « Gestion de conditions stables dans XACML intérêt d'une approche par notification », Université Paul Sabatier, 2007.

[33] « https://www.ekransystem.com/en/blog/rbac-vs-abac », consulté le 7 avril 2020.

[34] « https://www.java.com/fr/download/faq/whatis_java.xml », consulté le 9 avril 2020.

[35] « https://fr.wikipedia.org/wiki/XPath », consulté le 28 mai 2020.

[36] «https://fr.wikipedia.org/wiki/PhpMyAdmin», consulté le 18 juin 2020.

[37] ALFA Plugin for Eclipse User's Guide 1.0.2-01 2013 by Axiomatics AB.

[38] Ferraiolo, David F., Vincent C. Hu, et D. Rick Kuhn. « Assessment of Access Control Systems», (2007).

[39] « Cloud Computing en Afrique Situation et Perspectives », 11 Avril 2012

[40] Younis, Younis A., Kashif Kifayat, ET Madjid Merabti, « An access control model for cloud computing Journal of Information Security and Applications », 2014.

[41] « https://safenet.gemalto.fr/cloud-data-security/saas-security-cloud-access-control/ », consulté 16 septembre 2020.

[42] « https://www.companeo.com/securite-electronique/guide/le-controle-d-acces-1er-levier-de-la-securite », consulté 16 septembre 2020.