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 : RTIC 15/M2/2019

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 attributs(ABAC) Dans le cloud computing

---

Par :

SAIB LINDA

Soutenu le 06 juillet 2019, devant le jury composé de :

Djerou Leila	MCA	Président
Houhou Okba	MAA	Rapporteur
Tibermacine Ahmed	MCB	Examinateur

2018 / 2019

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.

Je dédie ce modeste travail

À 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 SOEURS

***ilhem**, la prunelle de mes yeux, **Djawhara**, la douce, **Sonia** l'aimable, **Nourddine** le généreux et **Ilyess** mon petit frère que j'adore et que j'aime profondément.*
Je vous souhaite une vie pleine de bonheur et de succès et que Dieu, le tout puissant, vous protège et vous garde.

À MES AMIS DE TOUJOURS :

Nadjiya, zineb, manel, zahra et ibtissem

En souvenir de notre sincère et profonde amitié et des moments agréables que nous avons passés ensemble.
Veuillez trouver dans ce travail l'expression de mon respect le plus profond et mon affection la plus sincère.

A tous ceux qui me sont chers et proches.
A tous ceux que j'aime et ceux qui m'aime.

linda

RÉSUMÉ

Le cloud computing a connu des grand progrès au cours des dernières années. De nos jours presque 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 un cadre de contrôle basé sur 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 : [politique, policy, XACML, ALFA, WSO2, cloud computing, ABAC, Axiomatic]

ABSTRACT

Cloud computing has made great progress in recent years. Nowadays almost all companies are changing their infrastructure to the cloud because of the many benefits 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 presented an access control framework and delegation that allows for delegating access, revoking access rights and assessing confidence in deferential cases.

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

ملخص

حققت الحوسبة السحابية تقدمًا كبيرًا في السنوات الأخيرة. في الوقت الحاضر ، تقوم جميع الشركات تقريباً بتغيير بنيتها التحتية إلى السحابة بسبب الفوائد العديدة التي تقدمها ، بالإضافة إلى ذلك ، فإن الانتقال إلى السحابة يقدم أيضًا عددًا من التحديات.

في هذه الأطروحة، ركزنا على واحد 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 : Etat de l'art

Chapitre 3 : la conception et implémentation

Liste des figures

liste des tableaux

Introduction générale

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]

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]

Nous allons présenter un cadre de contrôle basé sur les attributs (ABAC) et de délégation. 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.

Cette mémoire est structurée en trois chapitres :

Le premier chapitre est consacré à l'analyse du contexte et de l'état de l'art. L'objet de cette analyse est de déterminer le choix des modèles à utiliser dans nos travaux. Nous commençons par définir les notions de cloud computing, la sécurité informatique et après on va on passer en revue sur les contrôles d'accès et les modèles existants et on va terminer par la notion de délégation

Le deuxième chapitre est consacré à l'étude de notre système de contrôle d'accès. Ce chapitre traite principalement le modèle de contrôle d'accès basé sur les attributs et les travaux connexe

Le troisième chapitre traite de l'architecture protocolaire et l'implémentation de notre système**.** Nous implémentons une politique qui gère un contrôle d'accès basé sur les attributs, on mettrons notre politique de sécurité a l'épreuve et contrôler son bon fonctionnement a l'aide de l'outil WSO2 Identity Server.

Nous terminerons cette mémoire avec une Conclusion.

Chapitre 1

1.1. Introduction

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 de 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]

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. [1]

Dans ce chapitre on va traiter le cloud computing et après on discute un peu sur la sécurité du système informatique, puis on va définir les notions qui environnent le contrôle d'accès : définition, quelques exemples, les modèles de contrôle d'accès existé et les domaines d'application

1.2. L'informatique en nuage (Cloud computing)

1.2.1. Définition de l'informatique en nuage

L'informatique en nuage est un modèle qui offre aux utilisateurs du réseau un accès à la demande, à un ensemble de ressources informatiques partagées et configurables (serveurs, stockage, applications et services), qui peuvent être rapidement provisionnée et libre par minimum d'efforts de gestion ou d'interaction avec le fournisseur de service. [9]

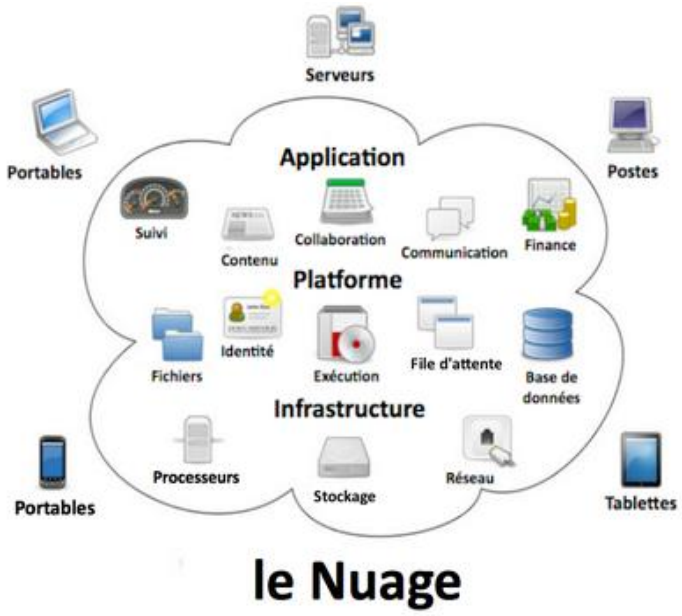


Figure1.1. Schéma donnant un aperçu sur les facteurs principaux du cloud computing [9]

1.2.2. Caractéristiques de l'informatique en nuage

Les Services de l'informatique en nuage ont des caractéristiques qui les distinguent des autres technologies: [14]

- Les utilisateurs de l'informatique en nuage ne sont pas propriétaires des ressources informatiques qu'ils utilisent.
- Les services sont fournis selon le modèle pay-per-use ou le modèle d'abonnement.
- Les ressources et les services fournis au client sont souvent virtuels et partagés par plusieurs utilisateurs.
- Les services sont fournis via l'Internet.

1.2.3. Les services de l'informatique en nuage

L'informatique en nuage propose plusieurs types de services permis ces services :

- L'infrastructure en tant que service (IaaS: Infrastructure as a Service): l'utilisateur loue des moyennes de calcule et de stockage, des capacités (partage de charge pare-feu, cache) l'utilisateur a la possibilité de déployer n'importe quel de logiciel incluant le système de l'exploitation. : VMware, ES2, S3…etc. [9]

- La plate-forme en tant que service (PaaS: Platform as a Service): les utilisateurs sont la possibilité de crée et déployer sur une infrastructure de l'informatique en nuage PaaS ses propres applications en utilisant les langages et les outils de fournisseur : Windows Azure, forse.com …etc. [9]
- L'application en tant que service (SaaS: Software as a Service): ce model de service caractérisé par l'utilisateur d'une application partagée qui fonctionne sur infrastructure de l'informatique en nuage. L'utilisateur accède à l'application par le réseau ou travers de divers types de terminaux (souvent via un navigateur web): Gmail, Yahoo, Google Docs…etc. [25]
- La communication en tant que service (CaaS: Communication as a Service): service de communication audio/vidéo, services collaboratifs, communications unifiées, messagerie [23]
- Le réseau en tant que service (NaaS*: Network as a Service*)*:* Internet managé (garantie du débit, Disponibilité, etc.), réseaux virtualisés VPN couplée aux services de l'informatique en nuage, bande passante flexible et à la demande. [14]

1.2.3. Les modèles de déploiement

Ces modèles de déploiement, au nombre de quatre, sont définis en fonction de leur relation à l'entreprise. [28]

1.2.3.1. Cloud public

Dans un cloud public, l'environnement est entièrement détenu par la société qui met à disposition ses services cloud. Les utilisateurs et clients d'un cloud public n'ont de droit ni sur l'infrastructure, ni sur le matériel, ni sur les logiciels, ni sur quoi que ce soit d'autre. Le fournisseur de cloud met à disposition des utilisateurs des ressources. Il conçoit, gère, maintient et fait évoluer ces ressources en fonction des besoins des clients au fil du temps. 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.

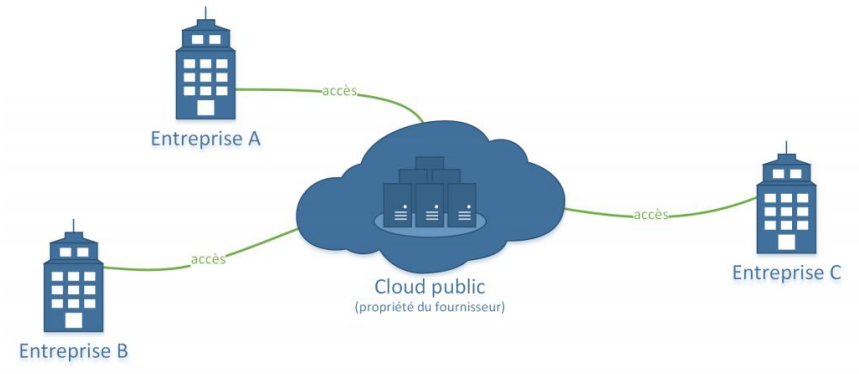


Figure1.2. Modèle de déploiement d'un cloud public [28]

1.2.3.2. Cloud privé

Le terme cloud privé est utilisé pour décrire l'infrastructure qui émule le cloud computing sur un réseau privé. 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.

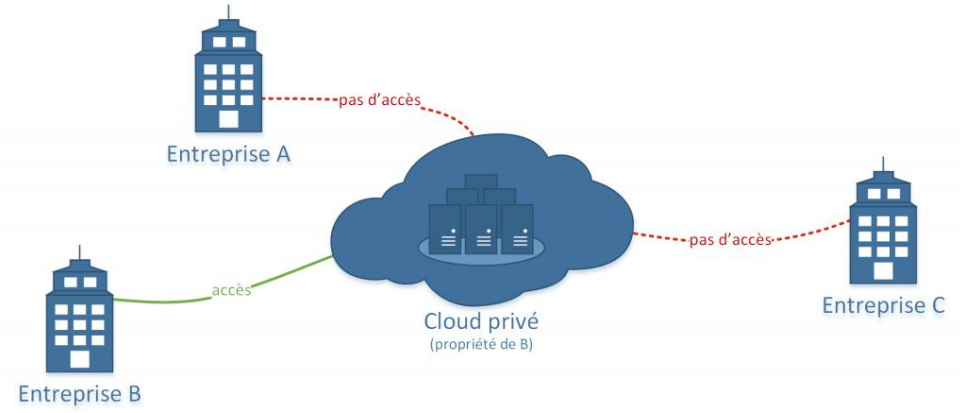


Figure1.3. Modèle de déploiement d'un cloud privé [28]

1.2.3.3. Cloud communautaire

Le cloud de type communautaire est un modèle de déploiement multi tenant partagé entre plusieurs entreprises ou organisations et qui est régi, géré, et sécurisé par l'ensemble des participants ou par un fournisseur de service.

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.

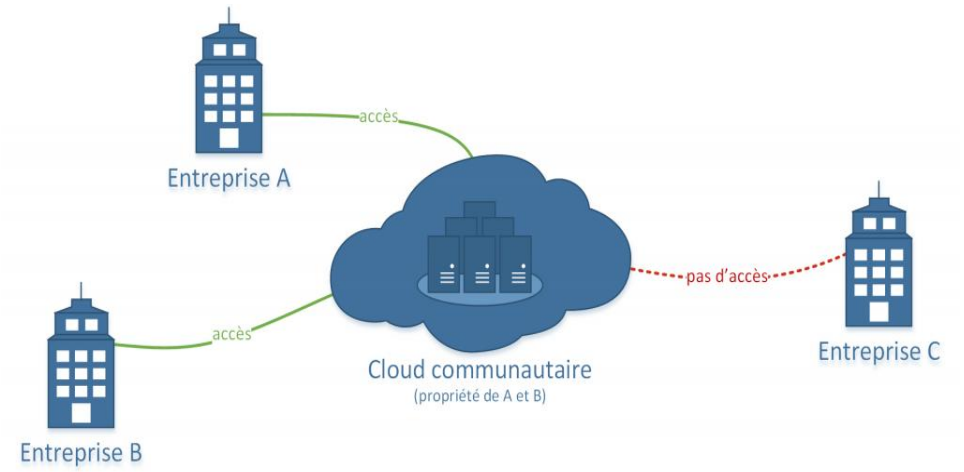


Figure1.4. Modèle de déploiement d'un cloud communautaire [28]

1.2.3.4. Cloud hybride

Comme son nom l'indique, un cloud hybride est la combinaison de plusieurs modèles de déploiement de clouds. Avec un cloud hybride, une entreprise peut tirer parti de la simplicité et du faible coût d'un cloud public — pour héberger des services classiques ne requérants pas de précautions particulières — tout en créant son propre cloud privé — pour des applications étroitement intégrées aux systèmes existants ou pour le stockage de données sensibles. Elle a également la possibilité de privilégier l'utilisation de son cloud privé tout en gardant la possibilité de déborder sur une offre de cloud public en cas de besoin temporaire.

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.

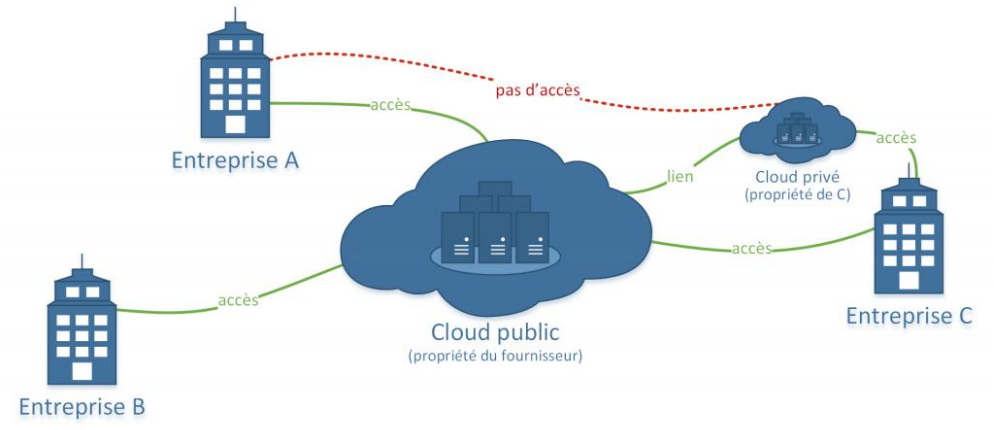


Figure1.5. Modèle de déploiement d'un cloud hybride [28]

1.2.4. Les applications du cloud computing

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, CRM, agilité, disponibilité, productions, RS, relation client. Il existe déjà plusieurs mises en œuvre du cloud computing telles qu'Amazon EC2, Windows Azure, Office 365 ou Google App Engine.

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... [29]

1.3. La sécurité informatique

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

1.3.3. Définition

La sécurité informatique est un ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, garantir la sécurité de l'information [3] et assurer le bon fonctionnement d'un système informatique et consiste à garantir que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu. [4]

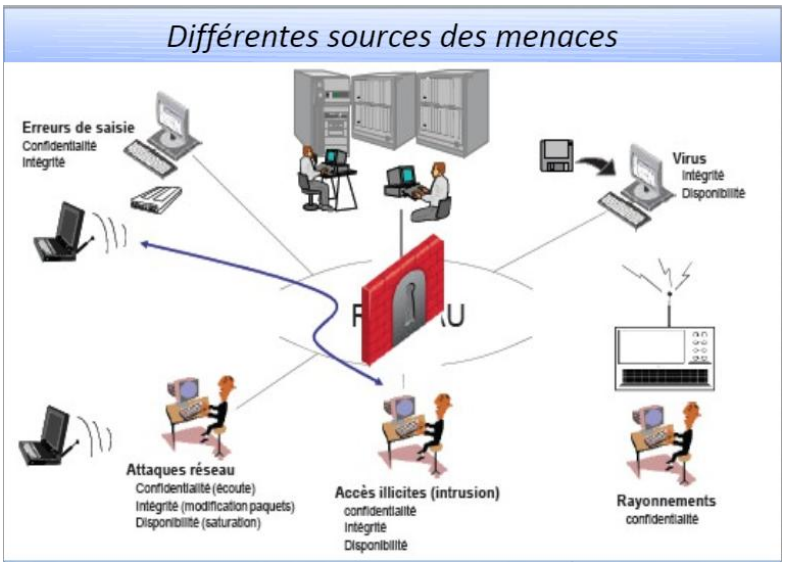


Figure 1.6. Exemple la sécurité des systèmes informatique [7]

1.3.4. Les technologies de sécurisation

Une classification utile les technologies de la sécurité est la suivante :

- **Confidentialité :** est la protection contre les attaques passives des données transmises. Plusieurs niveaux de protection de la confidentialité sont envisageables. La technique le plus général protège toutes les données transmises entre deux utilisateurs pendant une période donnée. Des formes restreintes de cette technologie peuvent également être définies, incluant la protection d'un message élémentaire ou même de champs spécifiques à l'intérieur d'un message. Un autre aspect de la confidentialité est la protection du flot de trafic contre l'analyse. Cela requiert qu'un attaquant ne puisse observer les sources et destinations, les fréquences, longueurs ou autres caractéristiques du trafic existant sur un équipement de communication. [5]

- **Authentification :** permet évidemment d'assurer l'authenticité d'une communication. Dans le cas d'un message élémentaire, tel un signal d'avertissement, d'alarme, ou un ordre de tir, la fonction du service d'authentification est d'assurer le destinataire que le message a bien pour origine la source dont il prétend être issu. Dans le cas d'une interaction suivie, telle une connexion d'un terminal à un serveur, deux aspects sont concernés. En premier lieu, lors de l'initialisation de la connexion,

il assure que les deux entités sont authentiques (c'est-à-dire, que chaque entité est celle qu'elle dit être). Ensuite, la technologie dont assurer que la connexion n'est pas perturbée par une tierce partie qui pourrait se faire passer pour une des deux entités légitimes à des fins de transmissions ou de réceptions non autorisées. [5]

- **Intégrité :** L'intégrité évite la corruption ou la destruction des données traitées par le système. Il faut d'abord empêcher des utilisateurs malveillants d'accéder aux données non autorisées (assurer la confidentialité et l'isolation des informations) mais également assurer que les données sont accédées de façon cohérente (par exemple avec un accès transactionnel). [1]

- **Non-répudiation:** empêche tant l'expéditeur que le receveur de nier avoir transmis ou reçu un message. Ainsi, lorsqu'un message est envoyé, le receveur peut prouver que le message a bien été envoyé par l'expéditeur prétendu. De même, lorsqu'un message est reçu, l'expéditeur peut prouver que le message a bien été reçu par le receveur prétendu.

- **Disponibilité :** est la garantie que les données et les services du système sont disponibles aux utilisateurs autorisés. Il faut pour cela empêcher un utilisateur malveillant d'arrêter ou de bloquer un service ("Denial of Service attacks"). [1]

- **Contrôle d'accès:** dans le contexte de la sécurité des réseaux, le contrôle d'accès est la faculté de limiter et de contrôler l'accès à des systèmes et des applications via des maillons de communications. Pour accomplir ce contrôle, chaque entité essayant d'obtenir un accès doit d'abord être authentifiée, ou s'authentifier, de telle sorte que les droits d'accès puissent être adaptés à son cas. [5]

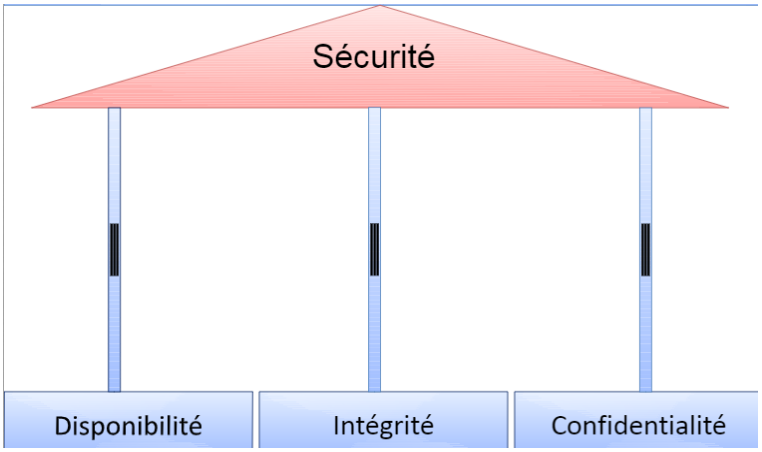


Figure 1.7. Les fondements de la sécurité informatique [7]

1.3.5. 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.

1.4. Les contrôles 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 [1]

1.4.3. 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 .[8]

Il existe une grande variété de méthodes, de modèles, de technologies et de capacités administratives utilisés pour proposer et concevoir des systèmes de contrôle d'accès. Ainsi, chaque système de contrôle d'accès a ses propres attributs, méthodes et fonctions, qui découlent d'une politique ou d'un ensemble de politiques. [39]

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. [39]

Le modèle de base de toutes les politiques de contrôle d'accès est montré sur la figure 1.3 :

Figure 1.8. Le modèle de base du contrôle d'accès [14]

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]

La matrice de contrôle d'accès est une structure qui contient une ligne par sujet et une colonne par objet dans le système. La cellule à l'intersection d'une ligne et d'une colonne décrit les droits d'accès du sujet sur l'objet. La figure 3.2 montre un exemple de matrice de contrôle d'accès. [14]

	$Fichier_1$	$Fichier_2$	$Fichier_3$	programme
Ann	propriétaire	lire écrire		exécuter
Bob	lire		lire écrire	
Carl	écrire			exécuter lire

Figure 1.9. La matrice de contrôle d'accès [15]

Bien que la matrice offre une bonne représentation conceptuelle des autorisations, elle est inappropriée pour l'implémentation. En effet, cette matrice peut être immense et son stockage direct comme un tableau à deux dimensions peut consommer beaucoup d'espace mémoire [15] Pour cette raison, il existe en pratique trois approches pour implémenter la matrice :

1. Table d'autorisation : les entrées vides de la matrice ne sont pas reportées dans la table. La table est composée de trois colonnes qui correspondent aux sujets, aux actions et aux objets. Chaque n-uplet de la table correspond à une autorisation.
2. ACL (Access Control List) : la matrice est stockée par colonne. Chaque objet est associé à une liste indiquant pour chaque utilisateur les actions pouvant être exercées par ce dernier sur cet objet.
3. Capacité (capability) : la matrice est stockée par ligne. Chaque utilisateur a une liste, appelée une liste de capacité, indiquant pour chaque objet les actions que l'utilisateur est autorisé à effectuer sur cet objet.

1.4.4. Les exemples de contrôle d'accès

- dans une organisation, seul le personnel du département finances peut voir la masse salariale des employés. Avec un mécanisme de contrôle d'accès en place, il est possible de s'assurer que seuls les employés qui travaillent dans le département des finances ou possèdent des attributs spécifiques ont accès aux données de paie de tous les employés. [39]
- une personne veut retirer de l'argent au guichet automatique, le contrôle d'accès se définit ici par le faite de vérifier que la personne dispose de sa carte bancaire et du bon NIP [2].

1.4.5. Les types de contrôle d'accès

Il existe deux types principaux de contrôle d'accès : [18]

1. Le contrôle d'accès physique permet de limiter les accès aux campus, aux bâtiments, aux salles et aux matériels informatiques.
2. Le contrôle d'accès logique restreint les connexions aux réseaux informatiques, aux fichiers système et aux données

1.4.6. Les objective de contrôle d'accès

Le contrôle d'accès a pour objectifs : [16]

- 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 réduisant :
  - ✓ les accès non autorisés
  - ✓ les diffusions non autorisées

1.4.7. Les modèles de contrôle d'accès

Avant de présenter ces différents modèles, nous allons définir dans cette section ce que nous appelons une politique de contrôle d'accès.

1.4.7.1. Définition d'une 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: [15]

- 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 …

Forme des règles d'une politique de sécurité:[11]

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

1.4.7.2. Les modèles

Il existe différents types de modèles de sécurité dont le souci majeur est de veiller au respect des exigences de confidentialité, de disponibilité, d'intégrité et de non répudiation de l'information. Ces modèles peuvent être catégorisés de différentes façons et en différents groupes dépendamment de leurs caractéristiques communes. Ces groupes sont aussi présents dans l'évolution des modèles de contrôle d'accès depuis les années 70 à nos jours. En effet, ils sont passés de simples modèles couvrant une seule organisation avec une gestion mono-utilisateur sous une autorité de sécurité unique vers des modèles supportant multiples organisations, une multitude d'utilisateurs et un contrôle par plusieurs autorités. Selon cette évolution on est passé d'un accès direct d'un sujet à un objet se basant sur son identité vers un accès indirect à travers des rôles attribués à ce sujet. Cela s'est manifesté par une chronologie d'évolution des principales familles de modèles de contrôle d'accès. On a eu donc des modèles basés sur l'identité, puis basés sur les règles, ensuite basés sur les rôles et finalement basés sur les attributs. [20] La Figure 1.5 illustre cette évolution.

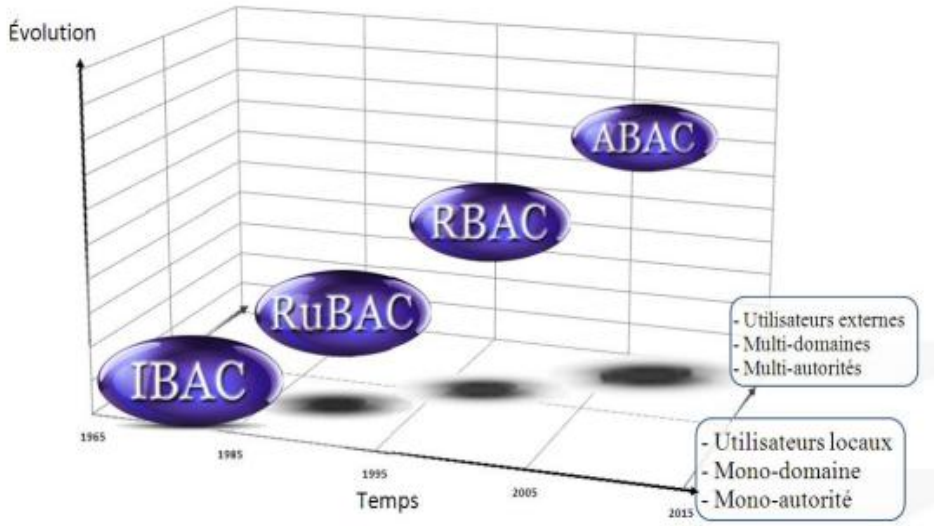


Figure 1.10. Évolution des modèles de sécurité [20]

1.4.7.2.1. Contrôle d'accès basé sur l'identité(IBAC)

Le contrôle d'accès basé sur l'identité (IBAC - Identity Based Access Control) est parmi les premiers modèles de contrôle d'accès. Ce modèle introduit les concepts

fondamentaux de sujet, d'action et d'objet. [10]

L'objectif de ce modèle IBAC est de contrôler tout accès direct des sujets aux objets via l'utilisation des actions. Ce contrôle est basé sur l'identité du sujet et l'identificateur de l'objet, d'où le nom du modèle IBAC. [10]

Dans IBAC, une permission a le format suivant : un sujet a la permission de réaliser une action sur un objet. La politique d'autorisation qui permet de spécifier les permissions est définie grâce à l'utilisation d'une matrice de contrôle d'accès dans laquelle les lignes et colonnes de la matrice correspondent respectivement à l'ensemble des sujets et des objets du système d'information. [10]

	Objet 1	Objet 1	Objet 1
Sujet 1	Rw	R	W
Sujet 2	R	-	Rw
Sujet 3	W	-	R

Tableau 1.1. Matrice IBAC [10]

Cependant la limite de ce modèle est sa mise à l'échelle. En effet, la politique devient complexe à maintenir lorsque le nombre d'entités est important. [10]

Le modèle RBAC introduit une abstraction de l'entité sujet qui devient rôle, permettant ainsi de réduire cette complexité. [10]

Les premières générations de modèles d'accès font partie du IBAC (Exemples : MAC, DAC, …).

1.4.7.2.1.1. Les modèles de politiques discrétionnaires(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 de Lampson, le modèle HRU, le modèle Take-Grant, le modèle TAM, etc. [10]

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). [12]

Le contrôle d'accès discrétionnaire (DAC) a été principalement implanté au sein des systèmes d'exploitation (Microsoft Windows, Solaris, Linux, FreeBSD). 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. [12]

L'implantation de ce modèle a donné lieu à la constitution de matrices d'accès initialement introduite en 1971 par Lampson qui a été généralisée en 1976 par Harison, Ruzzo et Ullman (HRU). 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 1.2).[12]

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

Tableau 1.2. Exemple d'une matrice d'accès [12]

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.[12]

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. [12]

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 [31] .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.[12]

1.4.7.2.1.2. les modèles d'autorisation 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. [10]

Les modèles associés au MAC : modèle de Bell-LaPadula, modèle de Biba, modèle de Clark et Wilson, modèle de muraille de Chine. [10]

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: [12]

- 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. [12]

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. [31]

1.4.7.2.2. Modèle de contrôle d'accès basé sur les règles (RuBAC)

RuBAC est un modèle de contrôle d'accès basé sur les règles et qui contrôle l'accès aux ressources en se basant sur des règles prédéfinies. RuBAC s'applique à un grand nombre de systèmes qui renforcent le contrôle d'accès aux informations par un ensemble de règles définies au sein de l'organisation. Il agit en comparant les demandes d'accès aux règles d'accès préétablies et qui sont généralement sous forme de labels attachés aux différents objets tels que des fichiers ou du matériel. Un des exemples les plus utilisés serait la règle de contrôle d'accès d'ouverture de session dans les systèmes d'exploitation durant des périodes de temps prédéfinies (exemple : entre 8:00 et 17:00) ou encore l'impression exclusivement en noir et 53 blanc pour les utilisateurs appartenant à un département donné dans une entreprise. Les règles peuvent aussi être appliquées à des sujets tel la révocation d'accès à un utilisateur après un certain nombre de tentatives erronées d'accès. Les règles sont définies par l'autorité de sécurité de façon centralisée. RuBAC implémente le contrôle d'accès en instaurant un ensemble de règles prédéfinies qui décrivent les relations entre sujets et objets au sein d'un système (autorisations et restrictions appliquées aux sujets et aux

objets). RuBAC n'est pas forcement un modèle basé sur l'identité puisque certaines règles peuvent, par exemple, être appliquées à tous les utilisateurs sans tenir compte de leur identité. [20]

1.4.7.2.3. Modèles de contrôle d'accès à base de rôle (RBAC)

Contrairement au modèle IBAC où les habilitations sont octroyées directement 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.[13] 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.[31]

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 oeuvre tout ou partie des principes des modèles RBAC.[12]

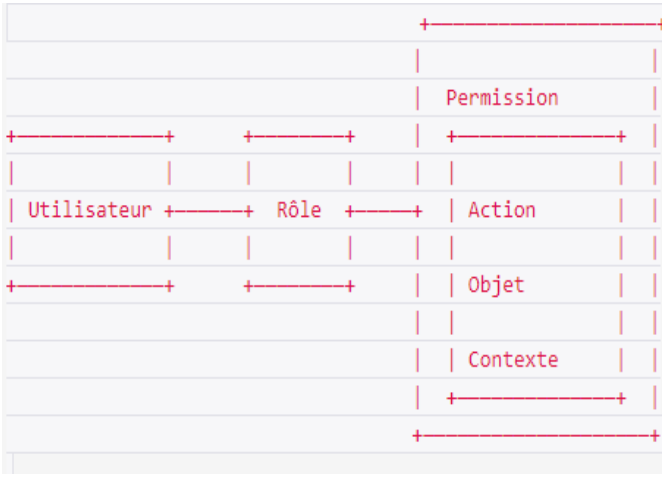


Figure 1.11. Model RBAC [12]

Par ailleurs, le NIST propose plusieurs déclinaisons du modèle RBAC:[13]

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

- Le modèle RBAC3 ou « the symmetric model », qui reprend les modèles RBAC1 et RBAC2 et prend en compte les interactions entre contraintes et hiérarchie.

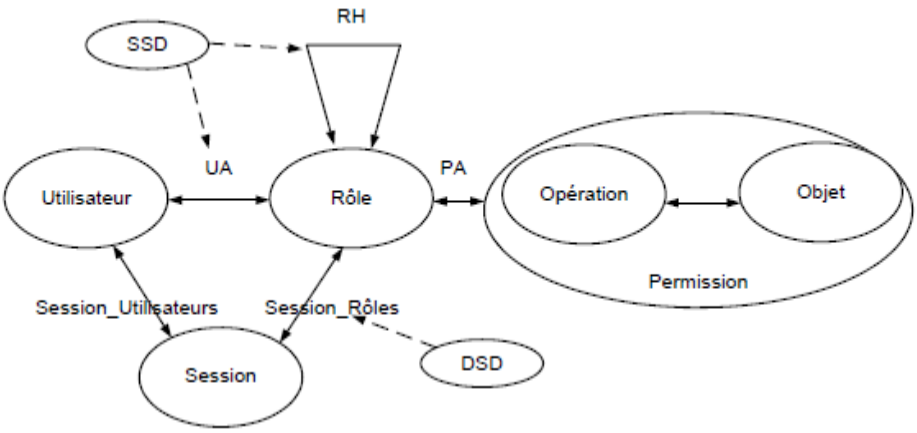


Figure 1.12. Modèle RBAC[12]

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 : [12]

- 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) et 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. [31]

L'inconvénient de RBAC réside dans la difficulté de gérer des règles dépendant du contexte de l'utilisateur, par exemple, de types « les étudiants ont le droit d'accéder uniquement à leurs données personnelles » ou « Seuls les étudiants qui se trouvent dans le bâtiment A ont le droit d'accéder au serveur de leur université ». 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 ». [12]

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

Le modèle ABAC a été développé par Eric Yuan et Jin Tong, dans le but de pallier aux difficultés que rencontrent les architectures web services en termes de sécurité. En effet, les accès à l'information au niveau de ces architectures web services se font non seulement sur les systèmes distribués mais très dynamiquement. Les modèles classiques sont généralement destinés à un fonctionnement statique, ils ne permettent guère une évolution dynamique. De part sa définition, le modèle ABAC peut être le plus adapté pour les architectures fonctionnant dans un environnement ouvert « in the cloud » où différentes organisations peuvent assurer à la fois les accès aux informations et la protection de leurs ressources. [10]

Le Tableau 1.3 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	RuBAC	ABAC
Autorité de sécurité	Centrale	Utilisateur	Généralement Centrale	Centrale	Centrale
Audit d'accès	Centrale	Utilisateur	Central	Centrale	Centrale
Propagation des droits d'accès	Centrale	Utilisateur	Généralement Centrale	Centrale	Centrale
Contrôle de flux	OUI	NON	NON	OUI	NON

d'information					
Multi-domaines	NON	NON	NON	NON	OUI

Tableau 1.3. Comparatif des modèles de sécurit [20]

1.4.8. 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 où l'on ne sait pas à l'avance de quels privilèges 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. [19]

On note trois classes de délégation: [19]

- 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.
- 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, 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.

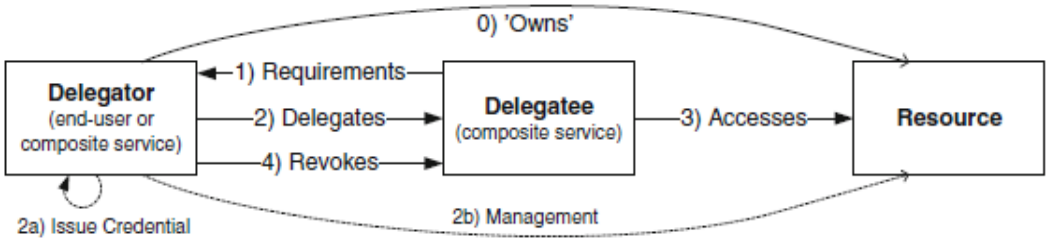


Figure 1.13. Scénario généralisé avec trois participants: le délégant, le délégataire et la ressource [19]

1.4.9. Les domaines d'application

Le contrôle d'accès est utilisé dans une plusieurs domaine lesquelles: [17]

- Les entreprises: les entreprises traitent des données confidentielles au 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 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.

1.4. Conclusion

Dans ce chapitre nous avons discuté brièvement sur le cloud computing puis sur la sécurité informatique, puis nous avons fait le tour sur les concepts de base sur le contrôle d'accès, on a terminé avec quelque domaine d'application

Dans le prochain chapitre nous allons détailler l'un de contrôle d'accès qui est ABAC et nous allons voir quelques travaux connexes

Chapitre 2

2.3. Introduction

La simple mise en place d'une politique de sécurité ne suffit pas pour sécuriser l'interopération. Il faut également tenir compte de son évolution par rapport aux besoins des différents participants à l'interopération .Ce qui amène à penser aux modèles de contrôles d'accès contextuels comme le contrôle d'accès basé sur les attributs (ABAC – Attribute Based Access Control).[10]

Dans ce chapitre, nous allons présenter une étude sur le contrôle d'accès basé sur les attributs. Puis nous allons présenter les travaux les plus marquants déjà fait dans le domaine du contrôle d'accès

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

2.4.1. Définition

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. Il existe plusieurs façons de définir ou d'utiliser des attributs dans ce modèle. Un attribut peut être la date de début du travail d'un utilisateur, l'emplacement d'un utilisateur, le rôle d'un utilisateur ou l'ensemble d'entre eux. 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. [31]

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. [31]

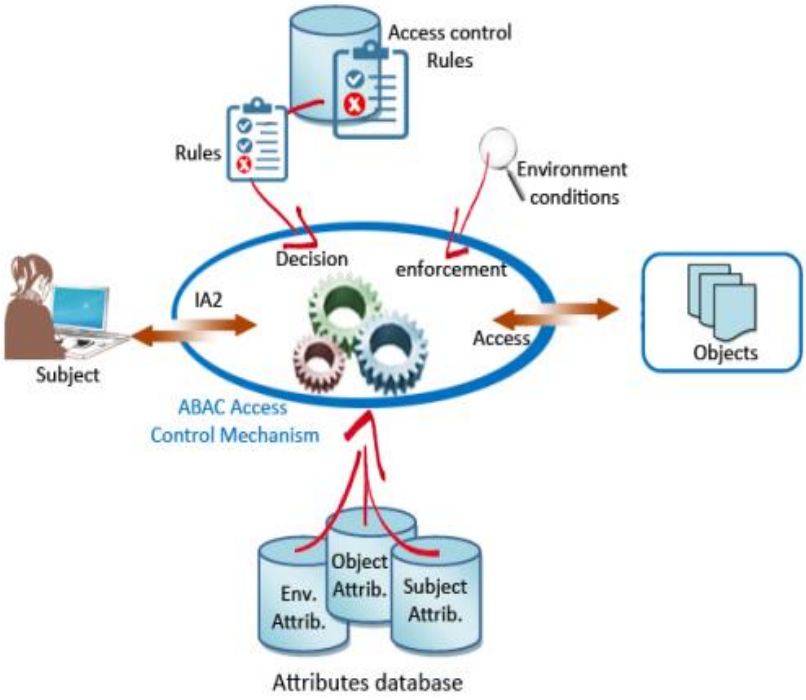


Figure 2.1. Modèle de contrôle d'accès ABAC [20]

2.4.2. Les composants ABAC

2.2.2.2. 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 [24] :

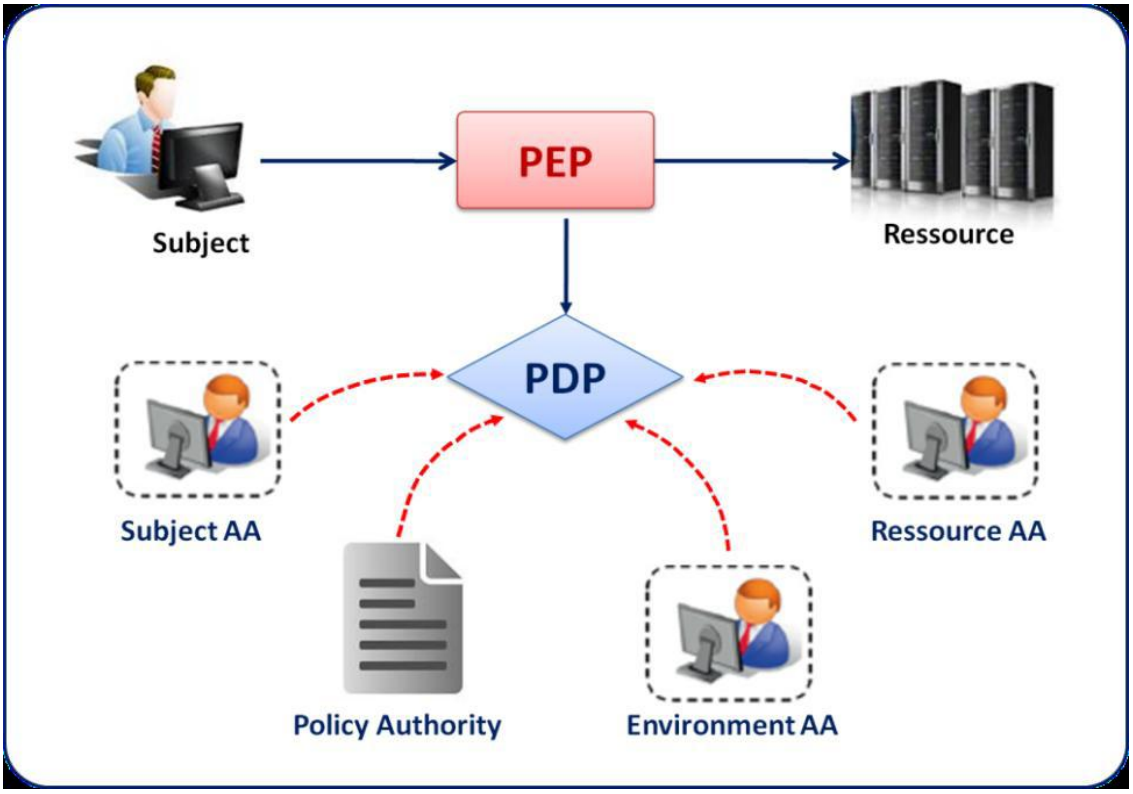


Figure 2.2. Architecture d'autorisation définie par ABAC [24]

1. **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.
2. **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.
3. **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).

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

2.2.2.2. 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 [10] :

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 ressources : C'est un objet du système sur lequel un sujet peut agir. Autrement dit, c'est une entité qui peut être accessible à un sujet. Une ressource peut être un fichier, un service, etc. A chaque ressource est associée des attributs, variables selon sa nature, mais qui peuvent être : son type, le nom de son auteur, son propriétaire, la date de modification, 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.

2.2.2.3. Formalisme de politique de sécurité ABAC

La formalisation de politique de sécurité ABAC est indiqué comme suivant:[10]

• 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.

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

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

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

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 resource ServiceOwner est libélée 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) ← (Role(s) = 'Manager') ∧ (Name(r) = 'ApprovePurchase')

2.2.3. Caractéristiques ABAC

Le modèle ABAC est connu par plusieurs caractéristiques lesquelles: [21]

- 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. Les avantages de la modèle ABAC

Les avantages du modèle ABAC sont comme suivant:[10]

- Introduction de la notion de gestion de contexte via les entités d'environnement. Ceci permet d'obtenir des modèles plus souples, qui peuvent

s'adapter à différentes situations et de dynamiser ainsi la prise de décision pour le contrôle d'accès. Cela est très appréciable pour l'application sur des SOA (Services Orienté Architectures), ou de nombreux paramètres peuvent influer sur la disponibilité de certaines ressources.

- Un modèle beaucoup plus adapté à la principale problématique liée au contrôle d'accès au sein de SOA, qui était le dynamisme d'accès aux informations.
- L'utilisation des attributs offre une granularité très fine pour définir les règles d'autorisation : Il suffit de définir un attribut pour prendre en compte un nouveau paramètre entrant en jeu dans la définition des autorisations, qu'il s'applique aux sujets, aux ressources ou aux environnements. exploiter les opportunités offertes par les autres modèles, par exemple avec l'attribut du sujet « rôle » on peut appliquer le modèle RBAC.

2.2.5. Les application 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. [27]

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 tirer parti d'ABAC sont:
    - Systèmes de gestion de contenu
    - ERPs
    - 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.

2.3. Les travaux connexes

Le travaille dans [24] porte sur la définition d'une politique du contrôle d'accès au niveau des APIs du Cloud. les APIs non sécurisées exposent les services Cloud et les données des clients à de potentielles failles de sécurité, tel que par exemple le vol de données qui induit la perte de confidentialité, l'altération de données (perte d'intégrité), destruction de données qui remet sérieusement en cause la continuité d'activité et la modification du niveau de privilèges d'un utilisateur qui est plus insidieux que les risques précédents (sécurité, espionnage).Donc il faut garantir une sécurité totale des API afin d'éviter les attaques ciblant ces interfaces, elles devraient être sujettes aux processus d'authentification et d'autorisation pour protéger les services Cloud des abus, des menaces envers la disponibilité ou des failles de confidentialité des données. Le fournisseur Cloud doit être en mesure de déterminer qui est autorisé à accéder à ses APIs et qui ne l'est pas. Idéalement, le fournisseur Cloud doit être capable de définir la politique du contrôle d'accès exprimant « qui » est autorisé à faire « quoi ».pour cela ce travaille a des objectifs sur cette problèmes de définir un système de contrôle d'accès au niveau des APIs du Cloud Computing afin de sécuriser l'accès à ces dernières et garantir un niveau de protection adéquat des ressources protégées accessibles au travers ces interfaces.

L’objectif principale de ce travaille est mettre en place une politique de contrôle d’accès au niveau des APIs du Cloud Computing pour sécuriser l’accès aux données d’entreprises via ces interfaces.

Première objectif ils ont choisis d’utiliser le contrôle d’accès à base d’attributs (ABAC) comme modèle de base à leurs politiques. ABAC permet de spécifier des permissions par rapport à toute caractéristique liée à la sécurité des utilisateurs, des actions, des ressources et de l’environnement. Il donne les moyens d’écrire des politiques, intégrant une diversité d’informations de sécurité qui sont associées à des exigences propres aux organisations. Cela est très appréciable pour l'application sur le Cloud, ou de nombreux paramètres peuvent influer sur la disponibilité de certaines ressources.

Le deuxième objectif est de renforcer le mécanisme d’authentification au niveau de l’API par l’adjonction d’un jeton d’accès unique et signé avec une durée de vie courte pour prouver l’identité de l’utilisateur et se prémunir contre les risques de fraudes liés par exemple à l’utilisation de fausses identités et au vol de mots de passe à distance.

Dans [21] les auteurs ont proposé une modèle PHR (Personal Health Records) qui est un nouveau modèle pour l'échange d'information sur la santé. Il permet aux patients de créer, de gérer, de contrôler et de partager leurs renseignements sur la santé avec d'autres utilisateurs ainsi que les fournisseurs de soins de santé Cependant, il y des problèmes de confidentialité sérieux sur l'externalisation des données de PHR des patients pour les serveurs Cloud.les auteurs ont proposé une solution a permet aux médecins de crypter et de soumettre leur prescription et les notes de diagnostic à des serveurs utilisant KP-ABE. Dans le système KP-ABE, les textes chiffrés sont marqués avec un ensemble d’attributs, et les clés privées sont associées à des structures d’accès (Access Tree). La clé ne peut déchiffrer un texte chiffré que si l’ensemble d’attributs du texte chiffré soit agrée par la structure d’accès de la clé privée. Le système KP-ABE est désigné la meilleure solution offrant un contrôle d’accès à grain fin

Dans [26] Wang et al ont présenté un modèle de contrôle d'accès dans lequel chaque utilisateur est assigné à un certain domaine, et chaque domaine est associé au rôle et au travail réel de l'utilisateur. Chaque rôle de l'utilisateur est assigné une tâche afin de pratiquer son rôle de travail. L'accès et le flux de données sont sécurisés en étiquetant les données selon leur niveau de sensibilité, qui ne pouvait être accessible uniquement aux utilisateurs à qui une tâche ayant une classification de sécurité qui domine les étiquettes de sécurité des données ciblées.

Critiques

Dans les travaux précédents que on les vu sont traité quelques avantages pour l’utilisation de contrôle d’accès dans le cloud cumputing spécialement le modèle ABAC mais on remarque que ils n’ont pas touché le point de délégation au les contrôles d’accès a cause la difficulté de la délégation et la dangereux, 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.

2.4. Conclusion

A travers ce chapitre, nous avons présenté dans la première partie le modèle de contrôle d’accès basé sur les attributs. Après dans la deuxième parité on a vu quelques travaux concernant le contrôle d’accès, dans le chapitre suivant, nous allons entamer la conception et l’implémentation de notre système proposé.

Chapitre 3

3.7. Introduction

Dans ce chapitre nous allons parler sur les étapes suivies pour concevoir et implémenter notre politique de sécurité qui repose sur le model 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.

3.8. Conception

3.8.1. Architecture générale

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

B

PEP

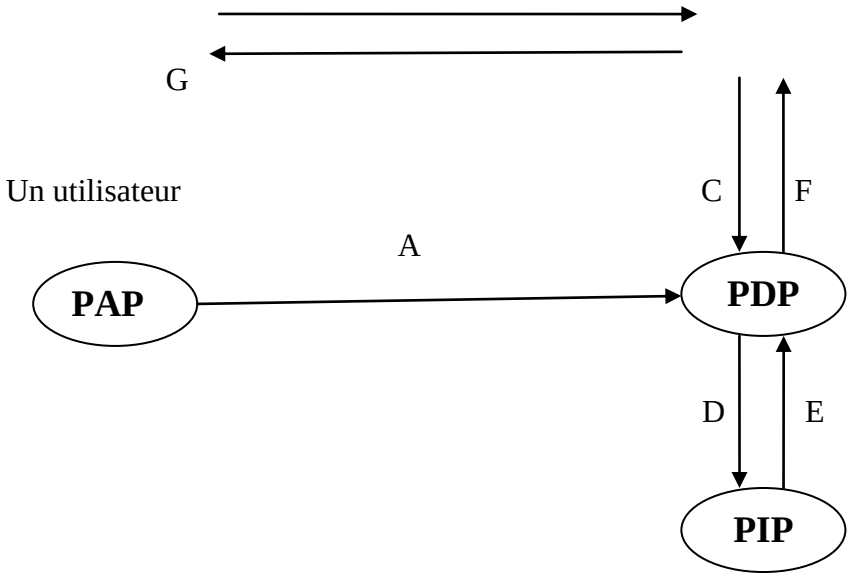


Figure 3.1. Schéma global de l'architecture

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

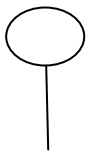
PEP (Policy Enforcement Point) :C'est l'entité qui intercepte la requête et l'envoie pour l'évaluer au niveau PDP

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

A. L'administrateur définit 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 retourne la décision et l'envoie au PEP
G. PEP remplit les obligations et, en se fondant sur la décision d'autorisation adressé par PDP, soit permet ou interdit l'accès

3.8.2. Diagramme de cas d'utilisation



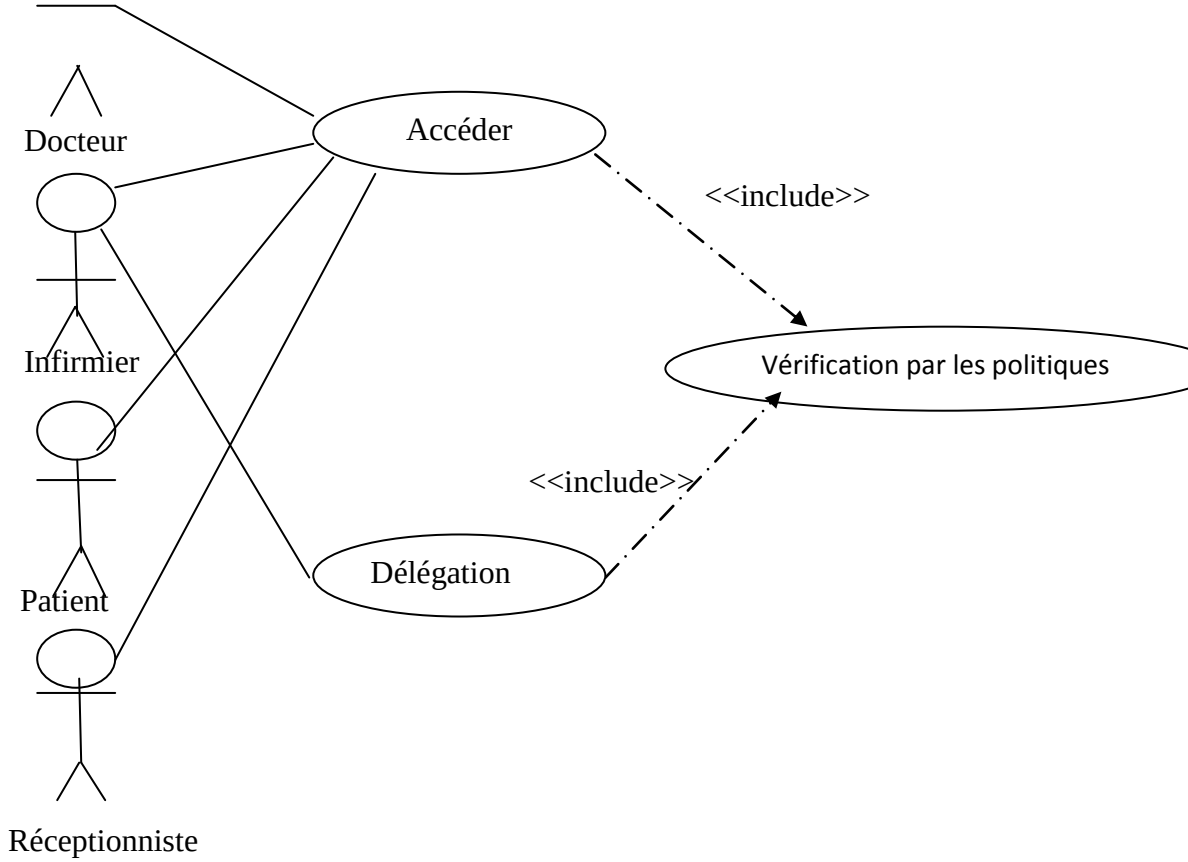
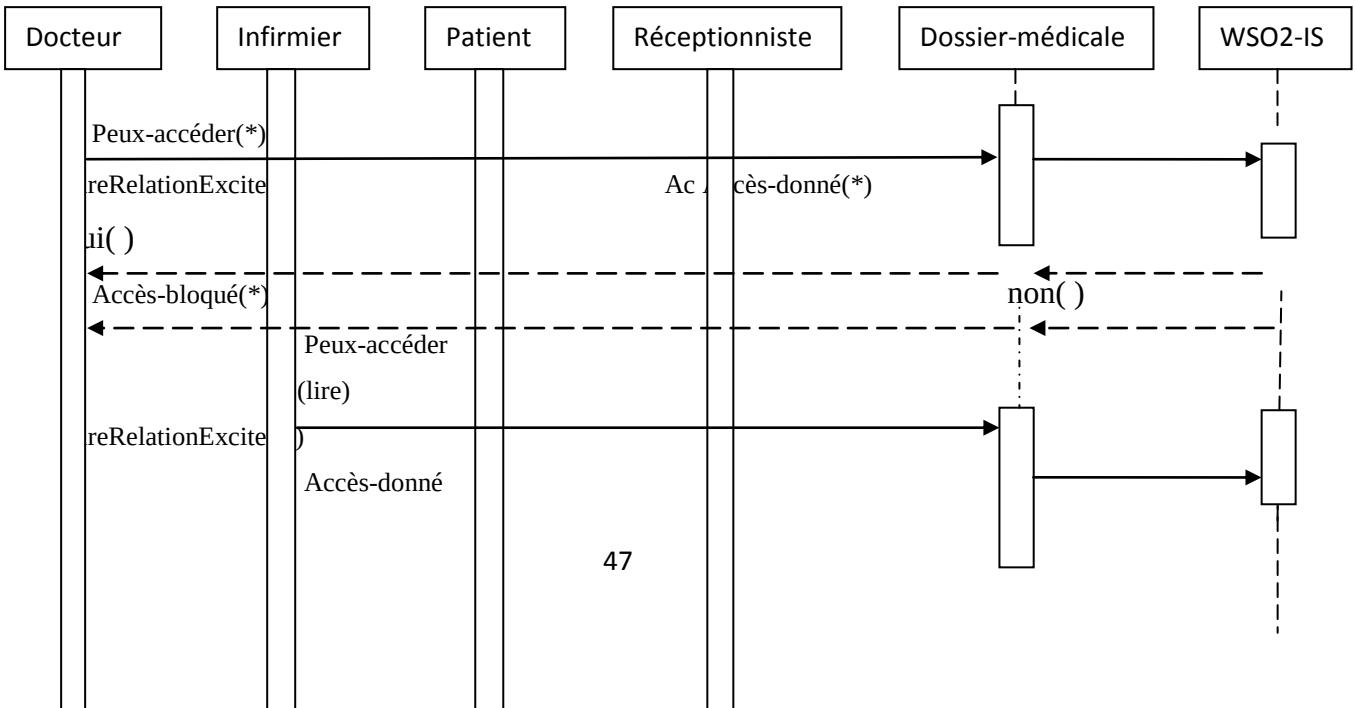


Figure 3.2. Schéma de cas d'utilisation

Dans le diagramme de cas d'utilisation on a un utilisateur qui va accéder a la ficher 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

3.8.3. Diagramme de séquence



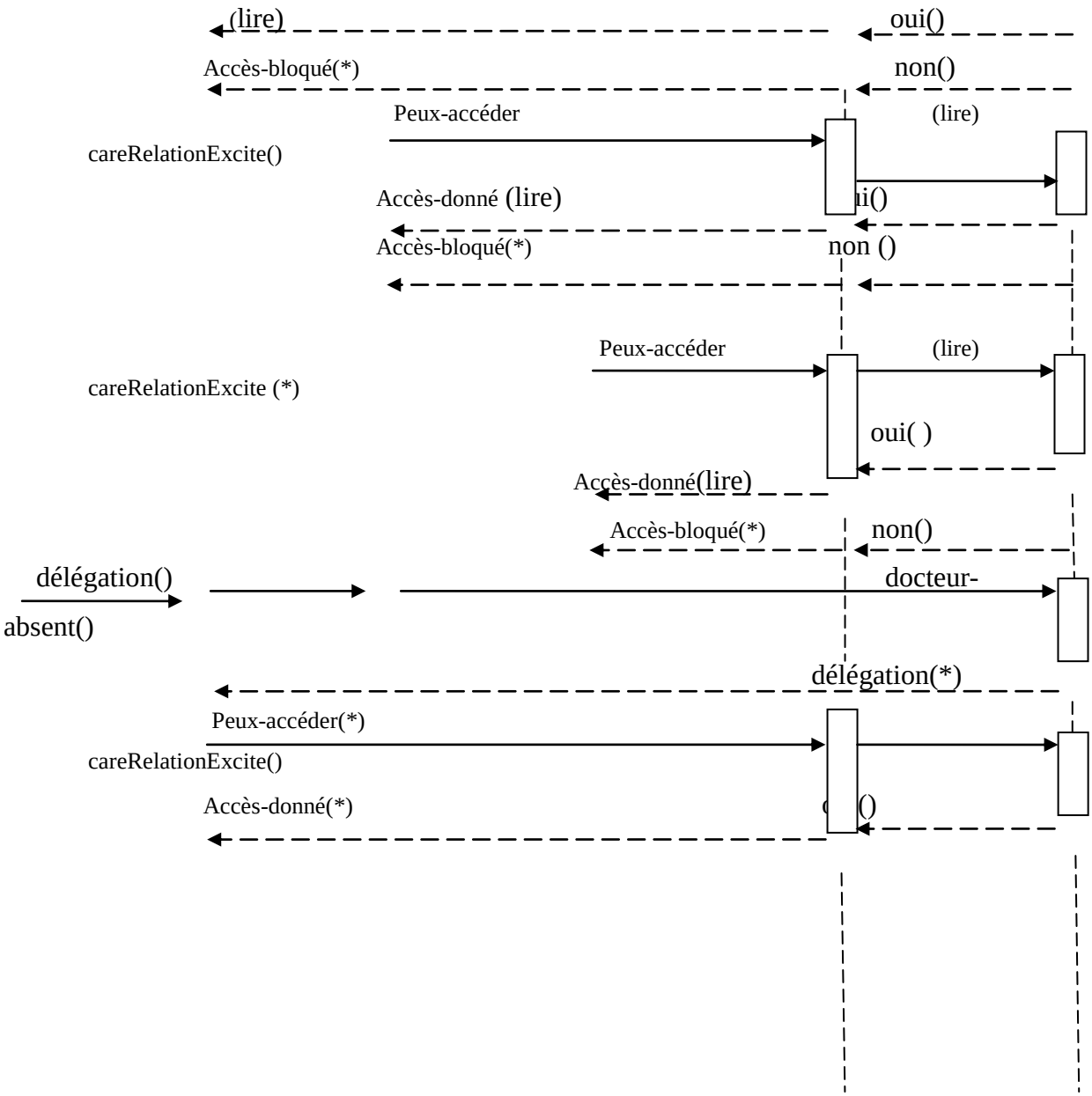


Figure 3.3. Schéma de diagramme de séquence

Le diagramme de séquence représente la succession chronologique des opérations réalisées par les acteurs (docteur, infirmier …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 touts le droits ex ici (lire, écrire…etc.)

CareRelationExcite () : teste si ya une relation entre le demandeur d'accès et le dossier médicale

Accès-bloqué(*) : y a une blocage et y a pas aucune droits (lire, écrire …etc.) accès

Peux-accéder(*) : le demandeur peux accéder au dossier médicale

3.9. Outils utilisés dans la programmation

3.9.1. 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. [32]

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. [32]

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, etc. [32]

3.9.2. 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. [33]

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]

Dans le présent projet, nous avons utilisé ALFA comme Plugin dans Eclipse.

3.9.3. 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.

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'eclipse 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. [34]

3.9.4. My Sql

Le terme MySQL, pour My Structured Query Language, désigne un serveur de base de données distribué sous licence libre GNU (General Public License). Il est, la plupart du temps, intégré dans la suite de logiciels LAMP qui comprend un système d'exploitation (Linux), un serveur web (Apache) et un langage de script (PHP). [35]

Créé en 1995, le serveur MySQL peut être utilisé sur de nombreux systèmes d'exploitation (Windows, Mac OS, etc.). Il supporte les langages informatiques SQL et SQL/PSM. [35]

Dans la pratique, le serveur MySQL peut se résumer à un lieu de stockage et d'enregistrement des données, que celles-ci soient ou non cryptées. Il est alors ensuite possible, via une requête SQL, d'aller récupérer des informations sur ce serveur très rapidement. C'est le cas, par exemple, avec les mots de passe enregistrés sur des sites web. Si le serveur détecte la présence du mot de passe entré dans un formulaire dans ses données, il autorise la connexion. S'il ne trouve pas le mot de passe, la connexion sera refusée. [35]

3.9.5. 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]

3.9.6. 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. [38]

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. [38]

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. [38]

3.10. Implémentation

3.10.1. 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.

Pour commencer il faut télécharger java JDK ainsi que l'environnement de développement java « Eclipse IDE ». En suite on télécharge le plugin ALFA pour Eclipse(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++.) et on l'ajoute à l'environnement de manière classique, on crée un nouveau projet et dans ce dernier on va créer un ficher qu'on va nommer par exemple

« medical.alfa» le faite que l'on donne l'extension « .alfa » à notre ficher, eclipse va automatiquement savoir que c'est une politique et va demander d'ajouter la Nature « Xtext » avec la boite de dialogue nous allons appuyer sur « Yes ».

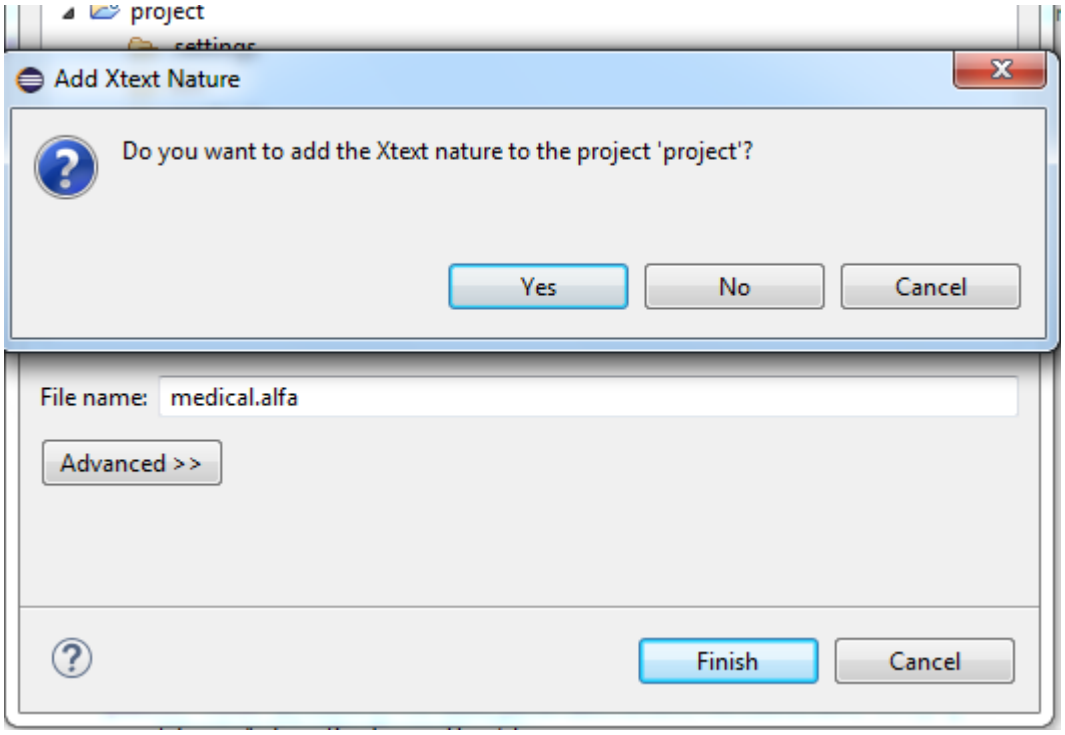


Figure 3.4. Ajoute nature Xtext

Il faut ensuite copier le fichier "system.alfa" de la distribution ALFA dans le projet. Ce fichier contient des définitions pour les fonctions XACML standard. Ensuite, placer le fichier appelé "standard-attributes.alfa" dans le projet. Il contient des définitions des identificateurs d'attribut pour les attributs standard de la spécification XACML.

Ensuite on commence à taper notre code ALFA, et on crée les politique, nous avons créé 3 politiques

1. La politique « hopital-médical » permet l'accès seulement si le docteur est assigné à ce malade, ceci est assuré grâce à un test dans la condition de la règle avec l'attribut « careRelationExists » 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 » ; de même pour l'infirmière en cas de délégation*.

Le message « the record is blocked » sera affiché si le dossier est bloqué pour n'importe quelle raison

```
namespace ObligationAdvice {
advice reasonForDeny = "http://example.com/advice/reasonForDeny"
    }
namespace medical{
    import Attributes.*


    //this policy is about medical record
    policyset topLevel {
      apply permitOverrides
      hopital_medicalPolicy
      folderPolicy
      delegationpolicy
   }

policy hopital_medicalPolicy {
target clause resource.resourceType =="medical record"
apply firstApplicable
rule doctor{

    permit
    target clause resource.userType=="doctor"
    condition (booleanOneAndOnly(resource.careRelationExists))

}

rule blockedacces{

    deny
    condition booleanOneAndOnly(resource.recordIsBloked)
        on deny {
            advice ObligationAdvice.reasonForDeny {
                resource.message="the record is blocked"
            }
        }}
```

Figure 3.5. Politique hôpital-médical

Grâce à ALFA, un ficher XACML concernant cette politique est généré automatique, et mis par défaut dans le sous-dossier src-gen comme est indiqué dans la figure dissous

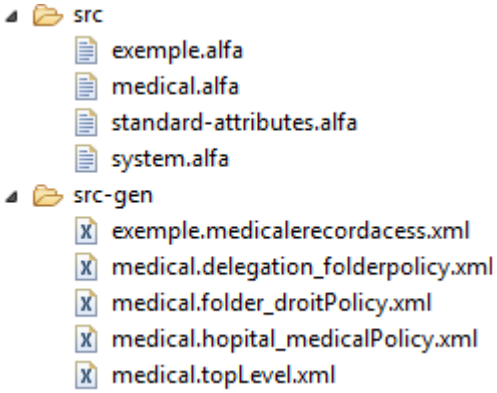


Figure 3.6. Le dossier src-gen

```
            <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:subject:subject-type"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                        MustBePresent="false"
                    />
                </xacml3:Match>
            </xacml3:AllOf>
        </xacml3:AnyOf>
    </xacml3:Target>
    <xacml3:Rule
            Effect="Permit"
            RuleId="http://axiomatics.com/alfa/identifier/medical.hopital_medicalPolicy.doctor">
        <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="user-type"
                            DataType="http://www.w3.org/2001/XMLSchema#string"
                            Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                            MustBePresent="false"
                        />
                    </xacml3:Match>
                </xacml3:AllOf>
            </xacml3:AnyOf>
        </xacml3:Target>
        <xacml3:Condition>
            <xacml3:Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:boolean-one-and-only" >
                <xacml3:AttributeDesignator
```

Figure 3.7. Code xacml génère pour la politique hôpital-médical

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

```
policy folder_droitPolicy{
    target clause resource.resourceType == "medical-record"
    apply permitOverrides
    rule {
        permit
         target clause resource.userType =="doctor" and resource.actionId =="write"
         or resource.userType =="patient"
         or resource.userType =="reception"
         clause resource.actionId=="read"

    }}
```

Figure 3.8. La politique folder_droit

```
        Effect="Permit"
        RuleId="http://axiomatics.com/alfa/identifier/medical.folder_droitPolicy.Id_10">
    <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="user-type"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                        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>
                    <xacml3:AttributeDesignator
                        AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
                        MustBePresent="false"
                    />
                </xacml3:Match>
            </xacml3:AllOf>
            <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">patient</xacml3:AttributeValue
                    <xacml3:AttributeDesignator
                        AttributeId="user-type"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                        MustBePresent="false"
```

Figure 3.9. Code xacml génère pour la politique folder_droit

3. La politique « délégation_folder » 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 ne va pas s'interrompre si la personne responsable ne se manifeste pas, car il y a toujours quelqu'un qui va le remplacer.

```
policy delegation_folderpolicy{
    target clause resource.userType =="nurse"
    apply permitOverrides
    rule delegation{
        permit
        condition booleanOneAndOnly(resource.doctorIsAbsent)
        target clause resource.userType == "nurse"
        clause resource.actionId == "read" and  resource.actionId =="write" }
}


}
```

Figure 3.10. La politique délégation_folder

```
                    <xacml3:AttributeValue
                        DataType="http://www.w3.org/2001/XMLSchema#string">nurse</xacml3:AttributeValue>
                    <xacml3:AttributeDesignator
                        AttributeId="user-type"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
                        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"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
                        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>
                    <xacml3:AttributeDesignator
                        AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                        DataType="http://www.w3.org/2001/XMLSchema#string"
                        Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action"
                        MustBePresent="false"
                    />
                </xacml3:Match>
            </xacml3:AllOf>
        </xacml3:AnyOf>
    </xacml3:Target>
```

Figure 3.11. Code xacml génère pour la politique délégation_folder

Tous les attributs utiles sont déclarés dans le ficher "standard-attributes.alfa"

```
attribute subjectId {
    id = "urn:oasis:names:tc:xacml:1.0:subject:subject-id"
    type = string
    category = subjectCat
}

attribute resourceType {
    id = "urn:oasis:names:tc:xacml:1.0:subject:subject-type"
    type = string
    category = subjectCat
}

 attribute careRelationExists {
    id = "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
    type = boolean
    category = resourceCat
}
  attribute delegationRelationExists{
    id = "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
    type = boolean
    category = actionCat
}

attribute message{
    id = "urn:oasis:names:tc:xacml:1.0:action:action-id"
    type = string
    category = subjectCat
}
  attribute recordIsBloked{
    id = "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
    type = boolean
    category = resourceCat
}
  attribute doctorIsAbsent{
    id = "urn:oasis:names:tc:xacml:1.0:resource:resource-id"
    type = boolean
    category = resourceCat
}
```

Figure 3.12.le fichier "standard-attributes.alfa"

3.10.2. 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.

On aurait pu utiliser un autre serveur plus adéquat et fluide que WSO2IS tel que XRay (un serveur de test pour les politiques générées avec ALFA) développé par Axiomatics INC mais il est payant et inaccessible.

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.

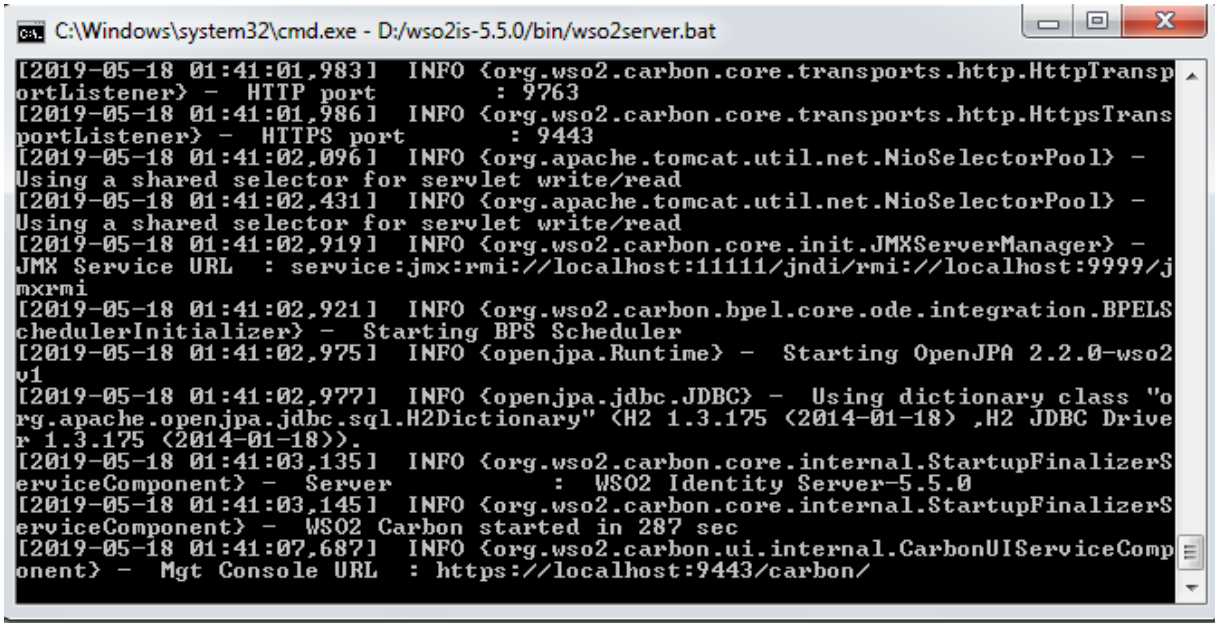


Figure 3.13. Lancement de WSO2-IS

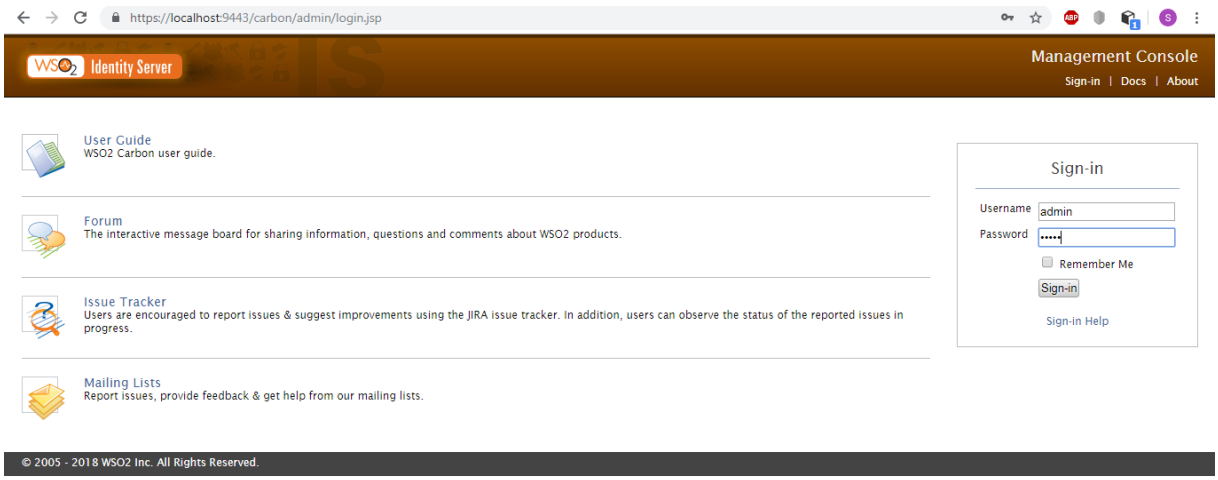


Figure 3.14. Page d'accueil WSO2

On accède à « Policy Administration » dans le menu Principal et on clique sur Ajouter une nouvelle politique « Add New Entitilement Policy ».

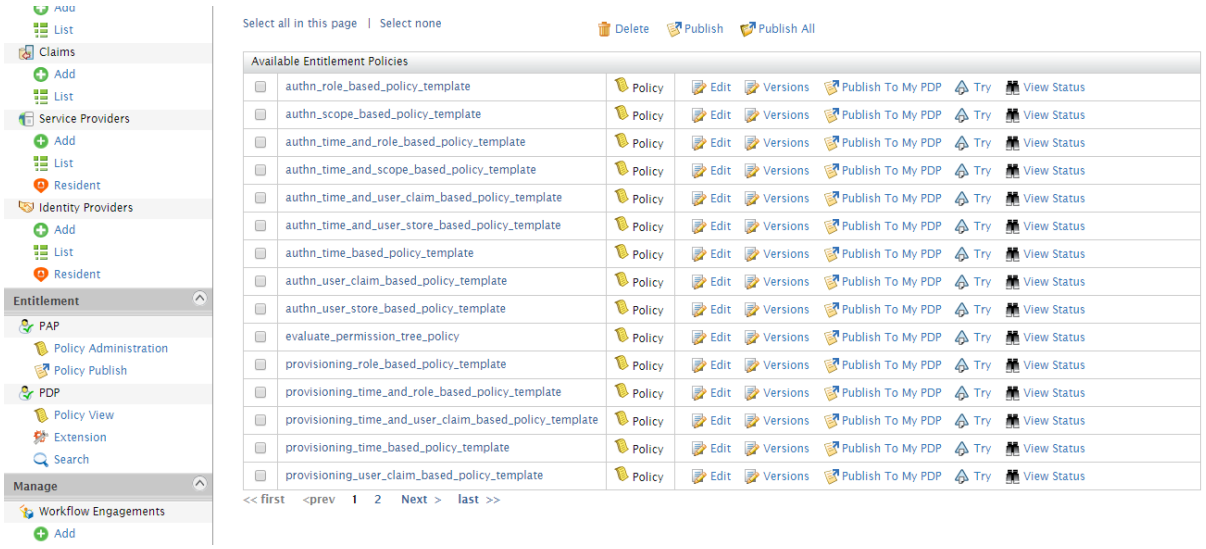


Figure 3.15. La page principale de PAP

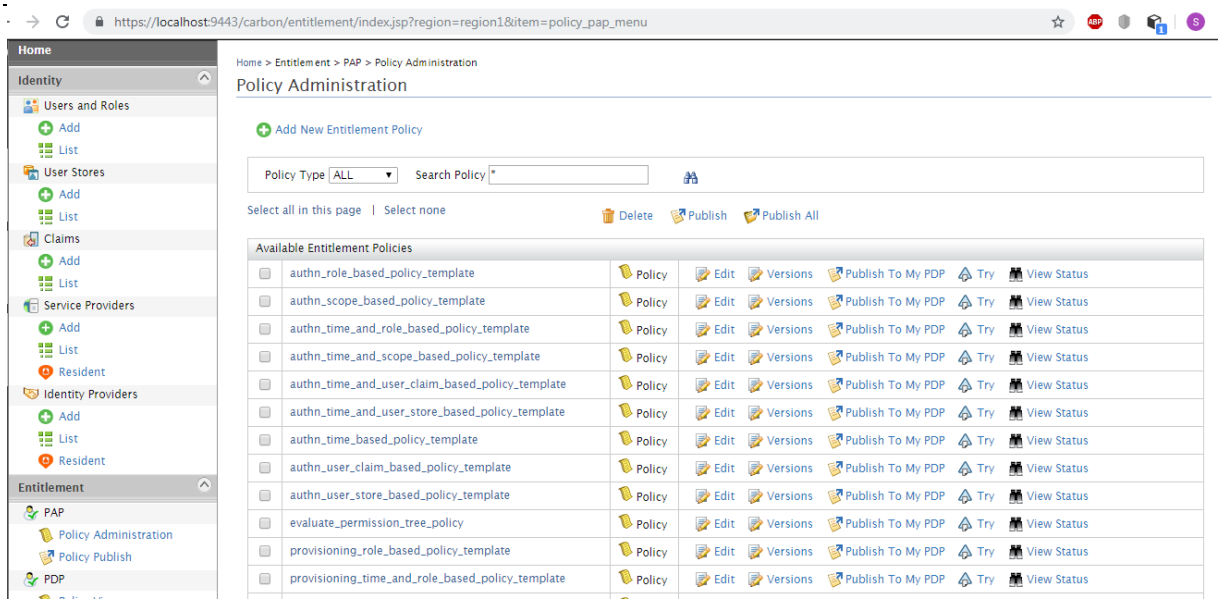


Figure 3.16.la page PAP pour ajoute nouvelle politique

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.

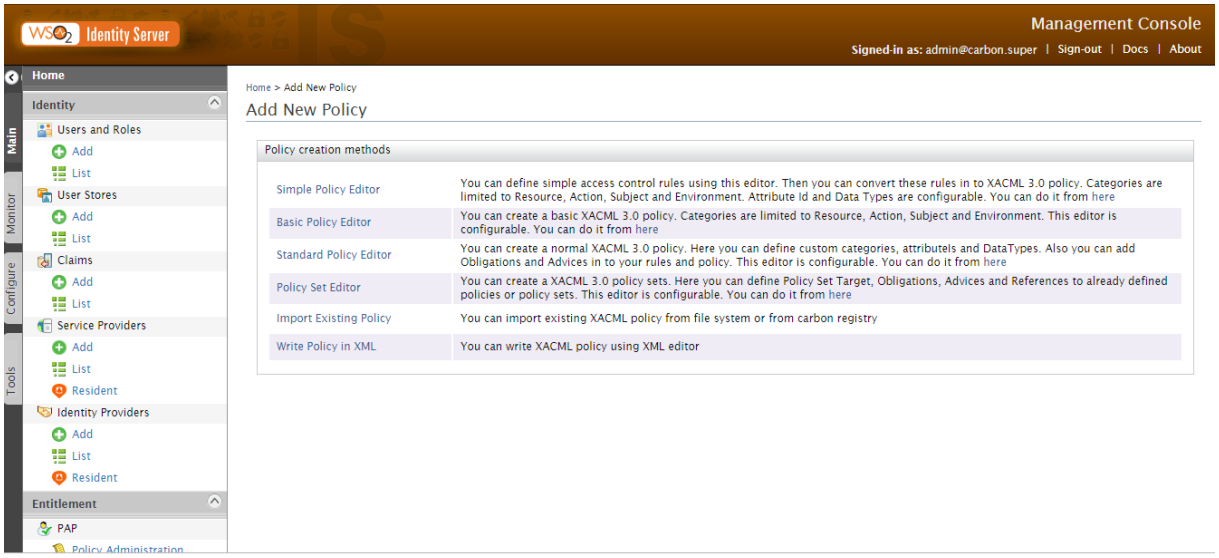


Figure 3.17. Interface pour choisir le choix d'ajoute d'une nouvelle politique

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

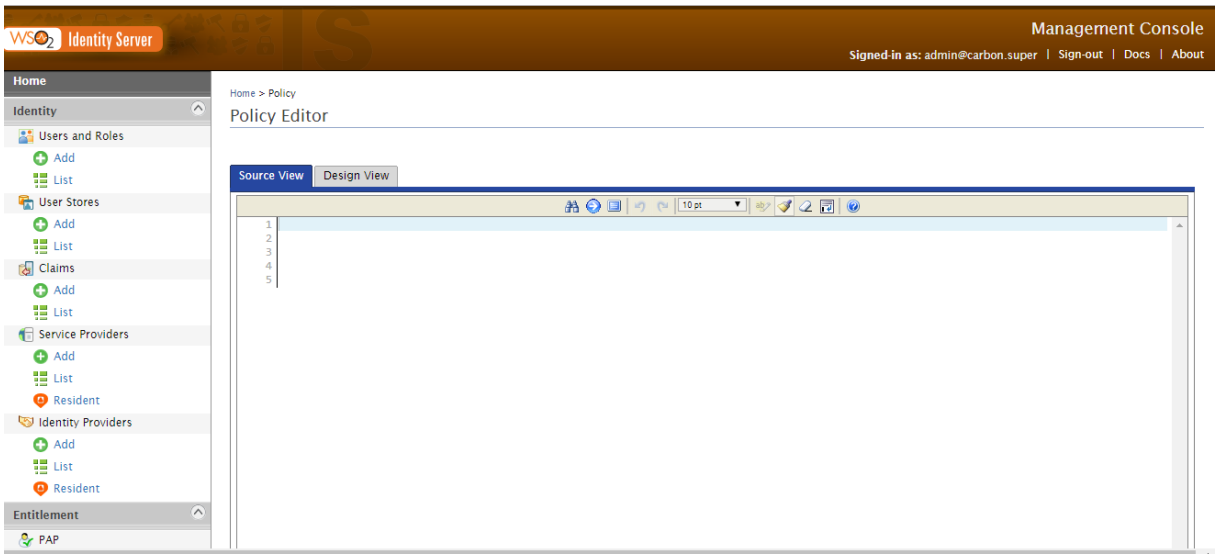


Figure 3.18. Éditeur de politique XML

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

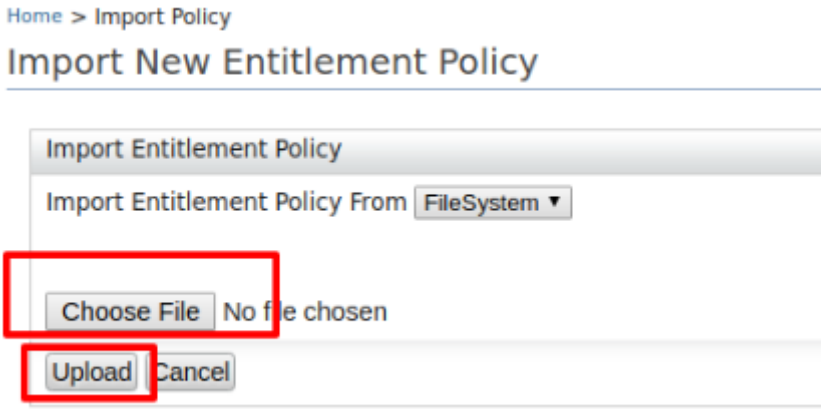


Figure 3.19. Interface d'importation de la politique

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

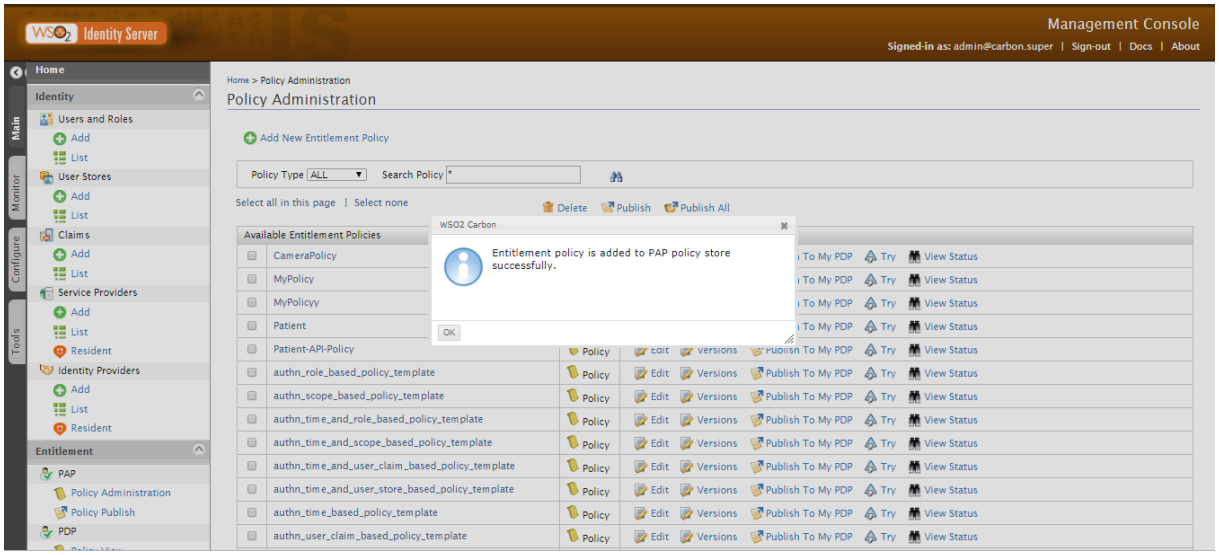


Figure 3.20. Message de réussir dans l'ajout

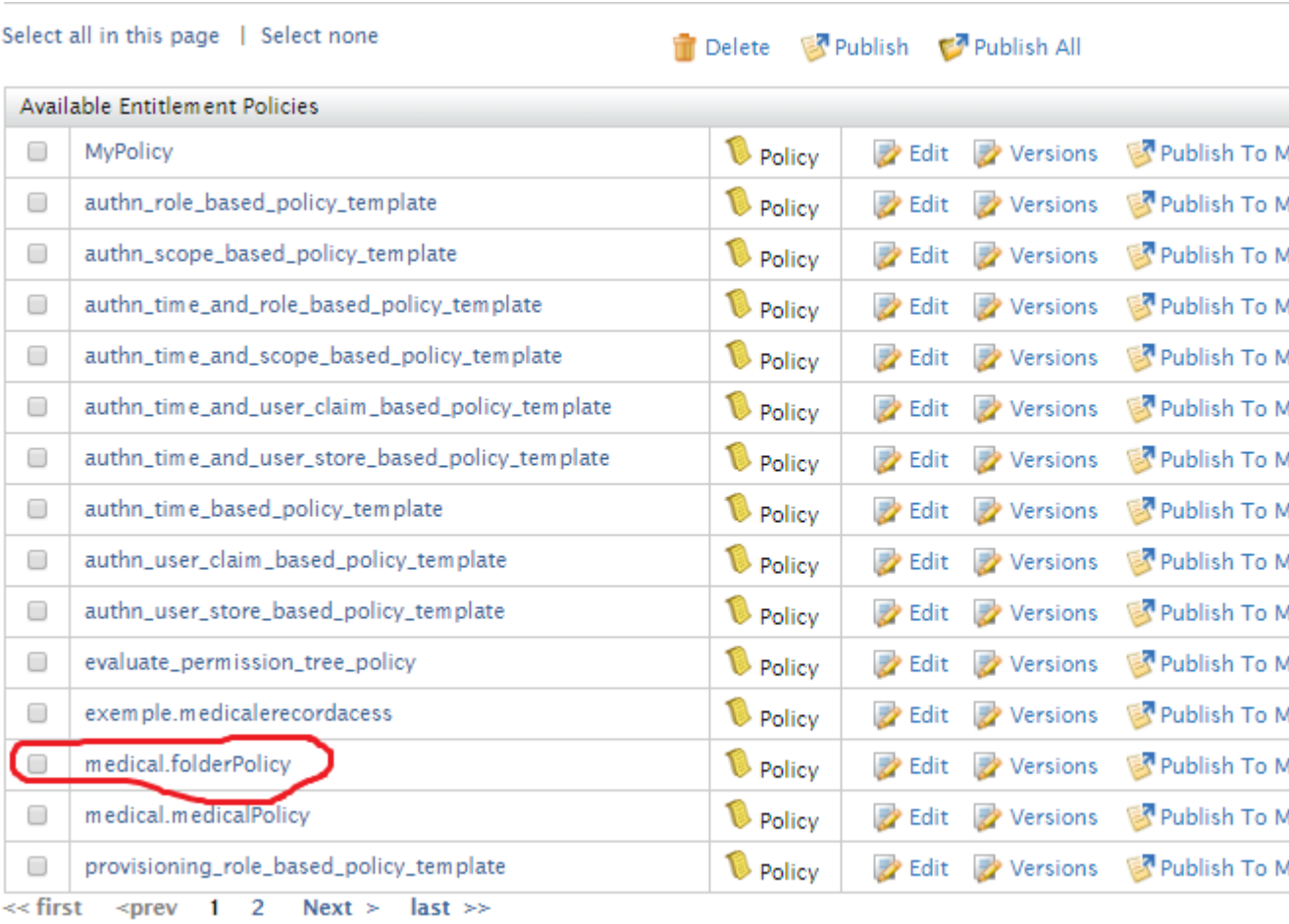


Figure 3.21. Interface montrant notre politique importe avec succès

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

Nous allons généré une requête qui teste si un docteur a le droit de lire un dossier médical du patient assigné, la ressource est dossier médicale « medical-record », le sujet est le docteur et l'action est lire ; enfin en appuie sur « Test Evaluate » pour voir le résultat.

Et la réponse attendue sera « permit »

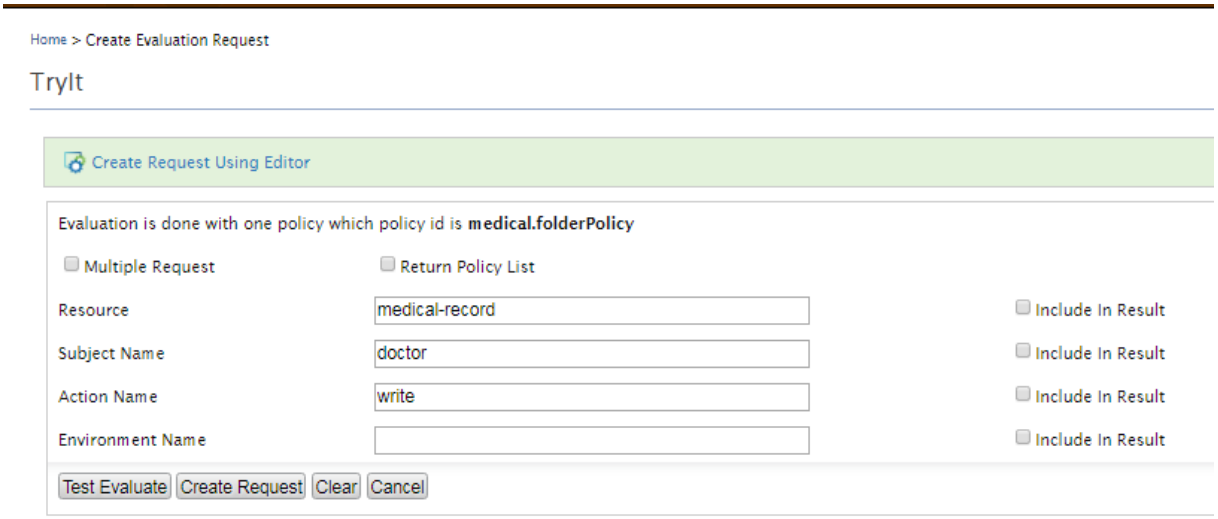


Figure 3.22. Fenêtre de l'éditeur de requête "TryIT"

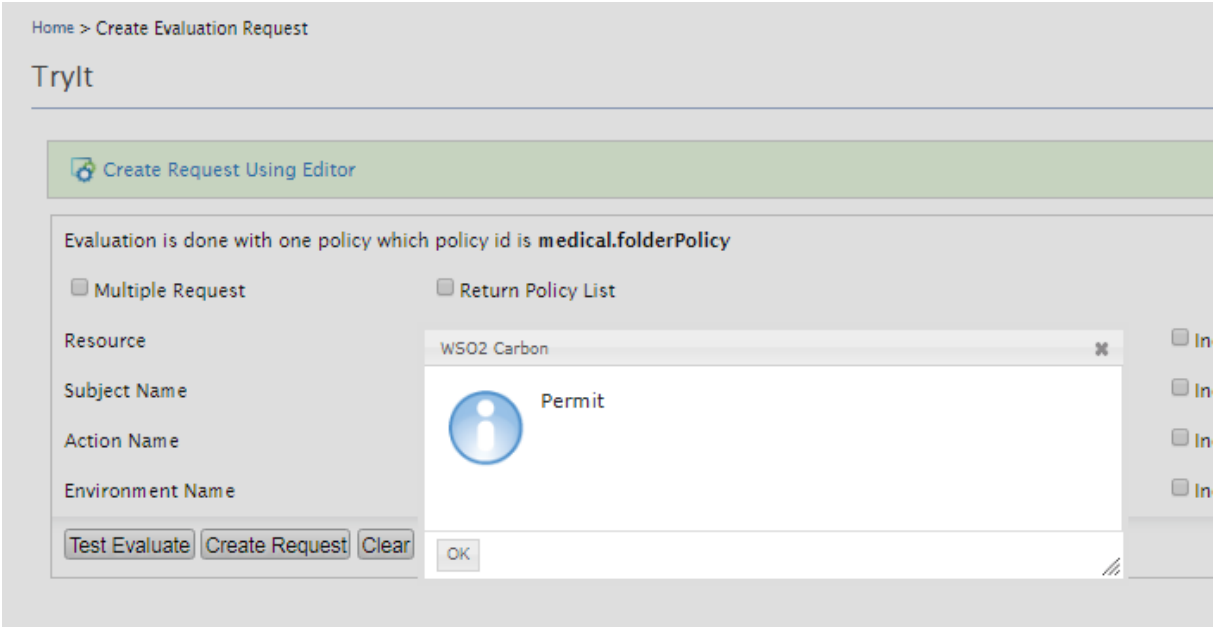


Figure 3.23. Résultat de la requête

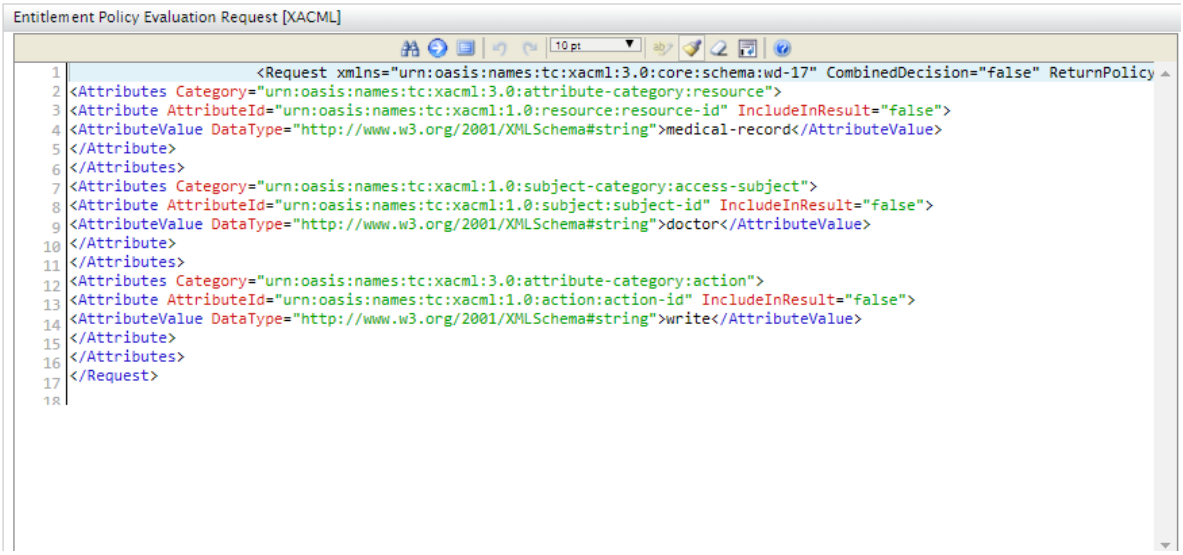


Figure 3.24. Code XML généré pour la requête

Autre test si le patient a le droit d'écrire dans le dossier médical, la ressource est dossier médicale « medical-record », le sujet est le patient et l'action est écrire ; enfin en appuie sur « Test Evaluate » pour voir le résultat.

Et la réponse attendue sera « deny»

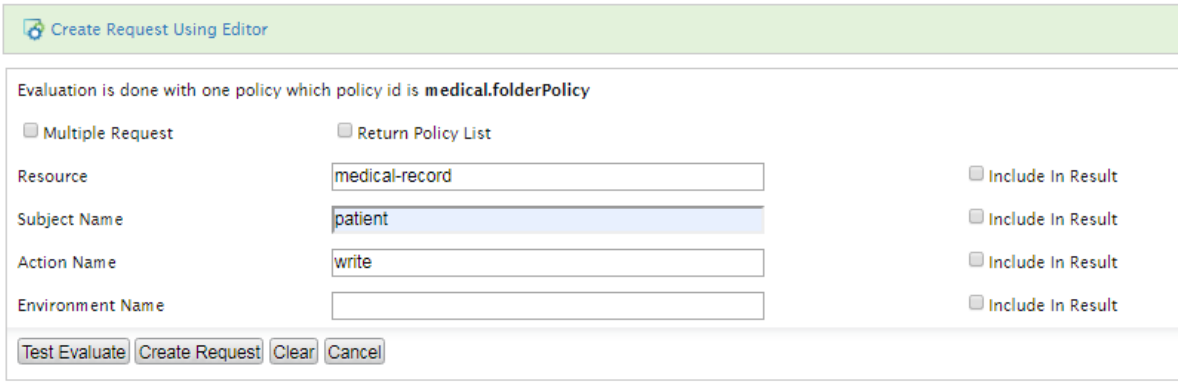


Figure 3.25. Fenêtre de l'éditeur de requête "TryIT" pour la 2 éme test

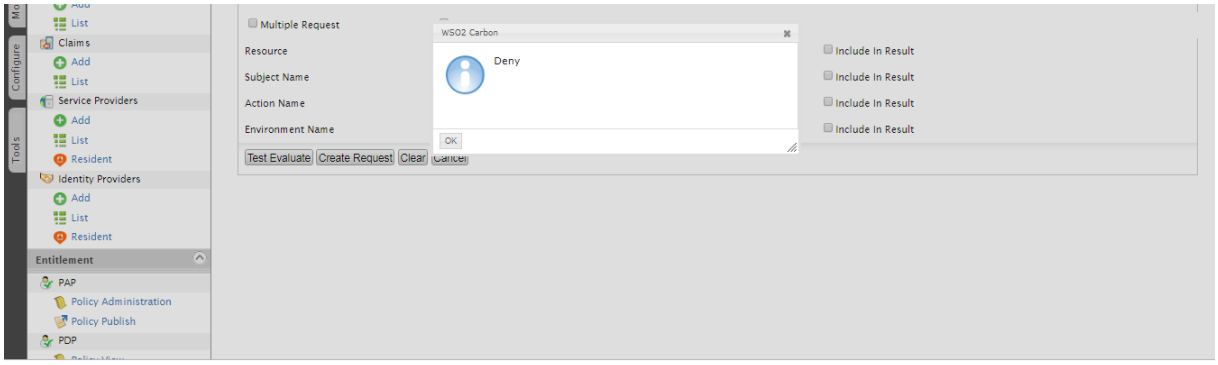


Figure 3.26. Résultat de la requête de 2 éme test

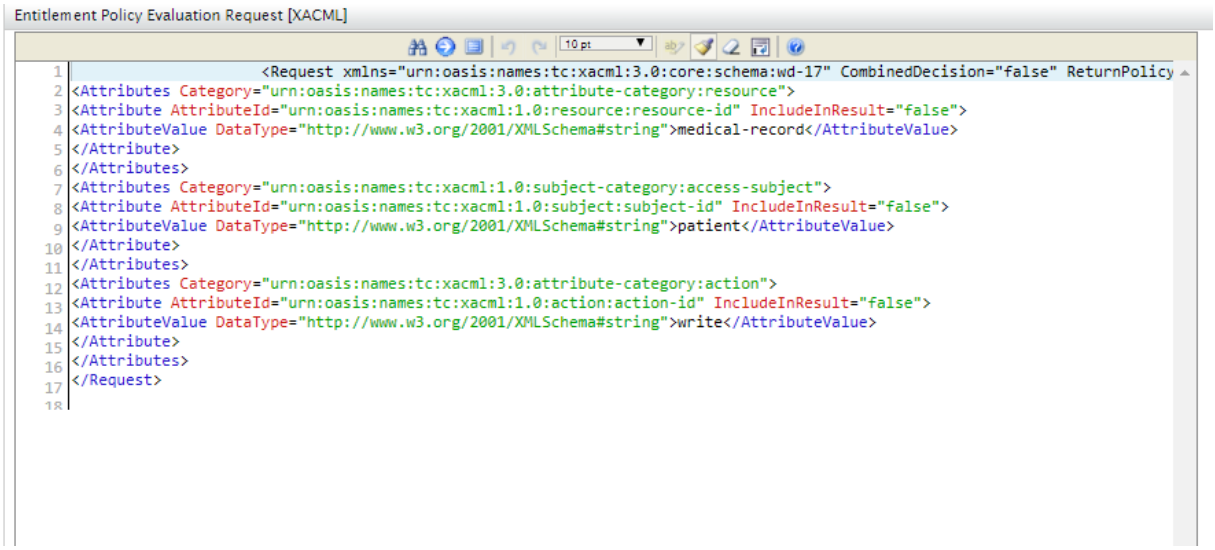


Figure 3.27. Code XML généré pour la requête de la 2 éme test

3.11. Conclusion

Dans ce chapitre nous avons présenté notre architecture de politique de sécurité qui repose sur le model 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 (responsabilité du délégataire ou délégataire) 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 propose 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 propose une politique de contrôle d'accès basé sur 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

Dans cette mémoire on a avons discuté brièvement dans le chapitre 1 sur le cloud computing puis sur la sécurité informatique, puis nous avons fait le tour sur les concepts de base sur le contrôle d'accès, on a terminé avec quelque domaine d'application

Nous avons présenté dans la première partie de chapitre 2 le modèle de contrôle d'accès basé sur les attributs. Après dans la deuxième parité on a vu quelques travaux concernant le contrôle d'accès

Et finalement dans le chapitre 3 on a terminé par une présentation de notre architecture de politique de sécurité qui repose sur le model 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] 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.

[2] MEMEL EMMANUEL LATHE, « Gestion de droits d'accès dans des réseaux Informatiques», Québec, Canada, 2016.

[3] Boukhlouf Djemaa, « Cours Sécurité des systèmes d'Information et Web », Université Mohamed Kheider Biskra, 2017/2018.

[4] Rudi Réz, « La sécurité informatique », CPE Namur-Hainaut, 2013.

[5] « Introduction à la sécurité informatique », Université Paris 13, Institut Galilée

[6] « https://fr.wikipedia.org/wiki/S%C3%A9curit%C3%A9_des_syst%C3%A8mes_d%27information », consulté le 17 fevrier 2019.

[7] « généralités sur la sécurité informatique et motivations », Université Kasdi Merbah
Ouargla, octobre 2014.

[8] Younis A. Younis*, Kashif Kifayat, Madjid Merabti , « An Access control model for cloud computing », School of Computing and Mathematical Sciences, Liverpool John Moores
University, Liverpool L3 3AF, UK.

[9] Labib Terrissa, « Informatique Mobil et Nuagique », Université Mohamed Khider-Biskra, 2017.

[10] 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.

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

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

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

[14] « Cloud Computing en Afrique Situation et Perspectives », Avril 2012.

[15] 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.

[16] « 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.

[17] « https://fr.wikipedia.org/wiki/Contr%C3%B4le_d%27acc%C3%A8s_logique », consulté le 2 mars 2019.

[18] IGNES, GPMSE, SVDI, « Guide méthodologique pour les système de contrôle».

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

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

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

[22] N.MEDDAH, A.TOUMANARI, « Contrôle d'accès au Cloud Computing à base de chiffrement KP;ABE et de proxy de rechiffrement anonyme», l'ENSA-Agadir-Maroc, 21 Mars 2015.

[23] « http://www.efort.com, consulté le 20 mars 2019.

[24] 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.

[25] Tara Salman « Networking Protocols and Standards for Internet of Things», 2015.

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

[27] « https://en.wikipedia.org/wiki/Attribute-based_access_control », consulté le 20 mars 2019.

[28] « https://blog.3li.com/cloud-les-modeles-de-deploiement/ », consulté le 15 Avril 2019.

[29] « https://fr.wikipedia.org/wiki/Cloud_computing#Services », consulté le 23 avril 2019.

[30] « Self-Adaptive Access Control & Delegation in Cloud Computing SNPD 2016», Shanghai, China, May 30-June 2016.

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

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

[33] « https://www.java.com/fr/download/faq/whatis_java.xml », consulté le 24 mai 2019.

[34] « http://dept-info.labri.fr/ENSEIGNEMENT/programmation2/intro-eclipse/ », consulté le24 mai 2019.

[35] « https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203595-mysql-my-structured-query-language-definition/ », consulté le 24 mai 2019.

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

[38] « https://docs.wso2.com/display/IS570/Architecture», consulté le 24 mai 2019.

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