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

N° d'ordre : 20/M2/2021

Mémoire

Présenté pour obtenir le diplôme de master académique en

Informatique

Parcours : **Système d'Information Optimisation et Décision(SIOD)**

Une approche hybride pour le contrôle d'accès

Par :
KHAIZAR YOUCEF

Soutenu le 31/06/2021 devant le jury composé de :

Nom Prénom	grade	Président
Houhou Okba	MAA	Rapporteur
Nom Prénom	grade	Examinateur

Année universitaire 2020-2021

Remerciement :

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

La première personne que nous tenons à remercier est notre directeur de recherche M. 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

Résumé :

Le cloud computing a connu des grands 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 les rôles (RBAC) et la séparation des tâches 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.

Abstract:

Cloud computing has seengreatprogress in recentyears. Nowadaysalmost all companies are changingtheir infrastructure to the cloud due to the manyadvantagesitoffers, apartfrom the move to the cloud alsopresents a number of challenges.

In thisthesis, wefocused on one of the main challenges, namelyaccess control and delegation in a cloud.

We have presented a control frameworkbased on attributes (ABAC) and roles (RBAC) and separation of duties and delegation, whichallows to delegateaccess, revokeaccessrights and assess the level of trust. in deferent case

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

ملخص

شهدت الحوسبة السحابية تقدمًا كبيرًا في السنوات الأخيرة. في الوقت الحاضر ، تقوم جميع الشركات تقريبًا بتغيير بنيتها التحتية إلى السحابة نظرًا للمزايا العديدة التي توفرها ، بصرف النظر عن الانتقال إلى السحابة ، فإنه يمثل أيضًا عددًا من التحديات.

في هذه الأطروحة ، ركزنا على أحد التحديات الرئيسية ، وهي التحكم في الوصول والتفويض في السحابة.

لقد قدمنا إطار عمل للرقابة على أساس السمات (ABAC) والأدوار (RBAC) والفصل بين الواجبات والتفويض ، مما يسمح بتفويض الوصول وإلغاء حقوق الوصول وتقييم مستوى الثقة.

Table desmatières :

Liste desfigures :

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, l'abus de services cloud, 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. [42]

[42]A. Younis Y, et al., An access control model for cloud computing, Journal of Information Security and Applications (2014), http://dx.doi.org/10.1016/j.jisa.2014.04.003

L'information dans le cloud computing est susceptible d'être partagée entre différentes entités, qui pourraient avoir différents degrés de sensibilité. Par conséquent, cela nécessiterait une isolation robuste et un contrôle des mécanismes d'accès.

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.[43]

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

Nous allons présenter un cadre de contrôle basé sur les rôles (RBAC) et les attributs (ABAC) et de délégation et de séparation des taches. 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 :

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

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

Chapitre 3 : Conception et réalisation, dans ce dernier chapitre on va présenter un scénario ou on va mettre notre politique de sécuritéa l'épreuve et contrôler son bon fonctionnement à l'aide de l'outil WSO2 Identity Server. Nous terminerons ce mémoire avec une Conclusion et Conclusion Générale.

Chapitre 1 :

Cloud Computing

1. L'informatique en nuage (Cloud Computing) :

1.1. Introduction :

Le terme Cloud Computing est à la base une rumeur apparue récemment dans la littérature informatique. La plupart des fournisseurs ont immédiatement introduit ce terme à tort et à travers dans leurs offres, ce qui ne simplifie pas la compréhension. Le but de ce dossier est de faire le point sur cette technologie.Pour mieux comprendre le phénomène Cloud Computing, voici une définition complète de ce terme.

Le Cloud Computing est un terme général employé pour désigner la livraison de ressources et de services à la demande par internet. Il désigne le stockage et l'accès aux données par l'intermédiaire d'internet plutôt que via le disque dur d'un ordinateur. Il s'oppose ainsi à la notion de stockage local, consistant à entreposer des données ou à lancer des programmes depuis le disque dur. La notion de Cloud ne doit pas non plus être confondue avec celle du Network Attached Storage (NAS), utilisée par beaucoup d'entreprises via un serveur en résidence. Ces réseaux locaux n'entrent pas dans la définition du Cloud. Cependant, certains NAS permettent d'accéder aux données à distance depuis Internet.

De manière générale, on parle de Cloud Computing lorsqu'il est possible d'accéder à des données ou à des programmes depuis internet, ou tout du moins lorsque ces données sont synchronisées avec d'autres informations sur internet. Il suffit donc pour y accéder de bénéficier d'une connexion internet et d'un support de lecture (ordinateur, smartphone, tablette, TV connectée). On peut donc le définir comme étant un modèle d'accès des ressources informatiques configurables, disponibles à la demande.[1]

1.2. Définition :

La définition opérationnelle retenue par le NIST est la suivante :

« Le Cloud Computing est un modèle informatique qui permet un accès facile et à la demande par le réseau à un ensemble partagé de ressources informatiques configurables (serveurs, stockage, applications et services) qui peuvent être rapidement provisionnées et libérées par un minimum d'efforts de gestion ou d'interaction avec le fournisseur du service ».[2]

1.3. Cloud-Computing-Schema :

L'image du Cloud est utilisée de façon métaphorique pour désigner internet. Cette comparaison date de l'époque à laquelle on représentait les infrastructures gigantesques des fermes de serveurs internet sous la forme d'un grand nuage blanc, acceptant les connexions et distribuant des informations tout en flottant.

Cette technologie permet aux entreprises d'acheter des ressources informatiques sous la forme de service, de la même manière que l'on consomme de l'électricité, au lieu d'avoir à construire et entretenir des infrastructures informatiques en interne.

Selon U.S. National Institute of Standards and Technology, le Cloud Computing est un modèle permettant d'établir un accès à la demande en réseau vers un bassin partagé de ressources informatiques configurable. Ces ressources sont par exemple des réseaux, des serveurs, de l'espace de stockage, des applications et des services. Elles peuvent être approvisionnées rapidement avec un effort de gestion et une interaction avec le fournisseur de services minimes. Le modèle Cloud met en avant la disponibilité, et se compose de cinq caractéristiques essentielles, trois mes de livraisons, et quatre modèles de déploiement.

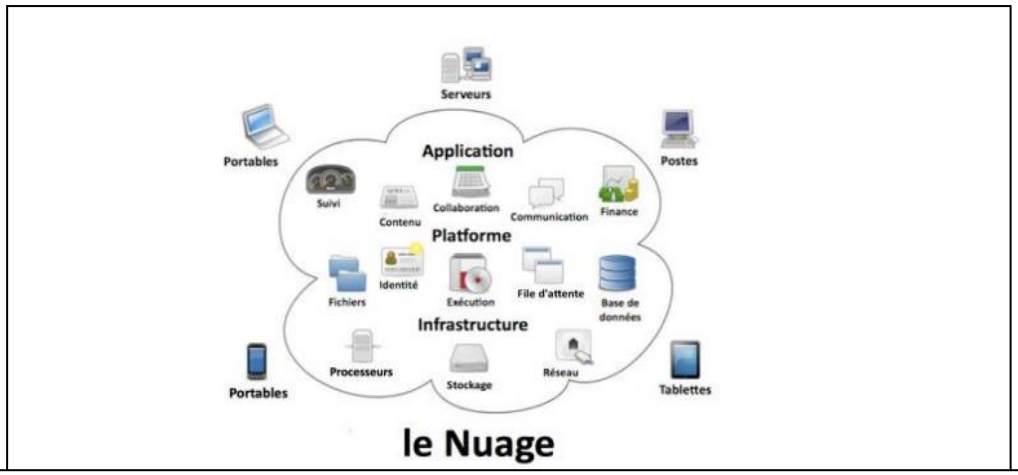


Figure 1 : Schéma donnant un aperçu sur les facteurs principaux du Cloud Computing. [36]

1.4. Types de services cloud : IaaS, PaaS, serverless et SaaS :

La plupart des services de cloud computing peuvent être classés en quatre grandes catégories : IaaS (infrastructure as a service), PaaS (platform as a service), serverlesset SaaS (software as a service). On les appelle parfois « pile » de cloud computing, car elles s'empilent les unes sur les

autres. Si vous savez en quoi elles consistent et en quoi elles sont différentes, vous pourrez plus facilement atteindre vos objectifs.

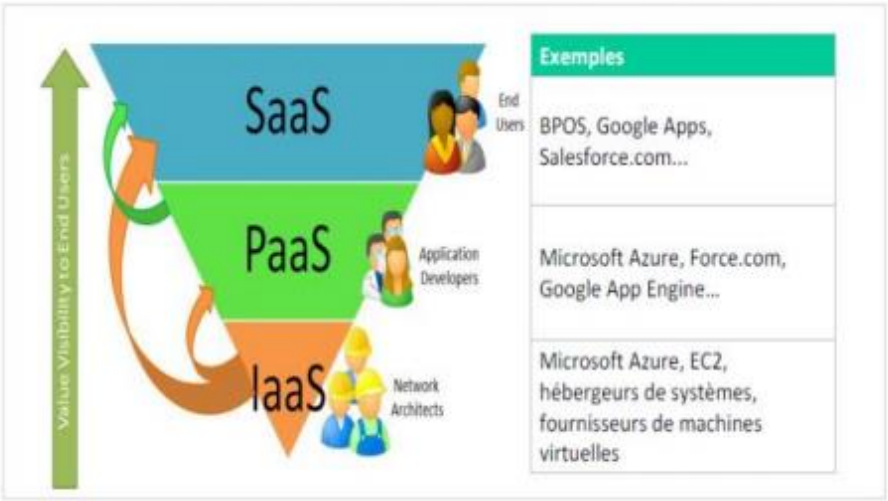


Figure 2 : Modèles des services. [36]

1.4.1. Infrastructure as à service (IaaS) :

La catégorie la plus basique des services duCloud Computing. Avec l'IaaS, vous pouvez louez une infrastructure informatique (serveurs, machines virtuelles, stockage, réseaux, systèmes d'exploitation) auprès d'un fournisseur de services cloud, avec un paiement en fonction de l'utilisation.

IaaS est l'un des quatre types de services cloud, parallèlement à *software as a service* (SaaS), *platform as a service* (PaaS) et serverless.

La migration de l'infrastructure de votre organisation vers une solution IaaS vous permet de réduire la maintenance des centres de données locaux, de réaliser des économies sur les coûts matériels et d'obtenir des insights professionnels en temps réel. Les solutions IaaS vous offrent la flexibilité nécessaire à la mise à l'échelle (vers le haut ou le bas) de vos ressources informatiques par rapport à la demande. Elles vous permettent également d'approvisionner rapidement de nouvelles applications et d'augmenter la fiabilité de votre infrastructure sous-jacente.[3]

1.4.2. Plateforme en tant que service (PaaS) :

Le terme plate-forme en tant que service (PaaS) décrit les services de Cloud Computing qui fournissent un environnement à la demande pour le développement, le test, la livraison et la gestion d'applications logicielles. PaaS est conçu pour permettre aux développeurs de créer rapidement des applications Web ou mobiles sans avoir à se soucier de la configuration ou de la gestion del'infrastructure du serveur, du stockage, du réseau et des bases de données nécessaire au développement.

PaaS peut également être défini comme un environnement de développement et de déploiement complet basé sur le cloud, avec les ressources nécessaires pour vous permettre de fournir n'importe quel service, des simples applications cloud aux applications d'entreprise complexes. Vous pouvez obtenir les ressources dont vous avez besoin auprès d'un fournisseur de services cloud, sur une base de paiement à l'utilisation, et y accéder via une connexion Internet sécurisée.

Tout comme l'IaaS, le PaaS comprend des infrastructures c'est-à-dire des serveurs, des composants de stockage et des réseaux, mais aussi des middlewares, des outils de développement, des services d'aide à la décision (BI pour*Business Intelligence*), des systèmes de gestionnaires de bases de données…etc. PaaS est conçu pour prendre en charge l'intégralité du cycle de vie des applications Web : conception, test, déploiement, gestion et mise à jour.

PaaS permet d'éviter les dépenses et les tracas liés à l'achat et à la gestion de licences pour les logiciels, les applications d'infrastructure de base et les middlewares, les formateurs de conteneurs tels que Kubernetes ou les outils de développement et d'autres ressources. Vous gérez les services et les applications que vous développez, et votre fournisseur de cloud s'occupe généralement du reste.[3]

1.4.3. Software as a service (SaaS) :

Le logiciel en tant que service (SaaS, Software-as-a-Service) est une méthode de diffusion d’applications logicielles via Internet, à la demande et en général sur abonnement. Avec le SaaS, les fournisseurs de services cloud hébergent et gèrent les applications logicielles et l’infrastructure sous-jacente, et gèrent la maintenance, par exemple la mise à niveau des

logiciels et l'application des correctifs de sécurité. Les utilisateurs se connectent à l'application via Internet, en général par l'intermédiaire d'un navigateur web sur leur téléphone, leur tablette ou leur PC.

Le Software as a service (SaaS) permet aux utilisateurs de se connecter à des applications cloud et de les utiliser via Internet. Les exemples les plus courants sont les outils de messagerie, de calendrier et les outils de bureautique, comme Microsoft Office 365.

Le SaaS offre une solution logicielle complète pour laquelle vous payez en fonction de l'utilisation à un fournisseur de services cloud. Vous louez l'utilisation d'une application pour votre organisation, et vos utilisateurs s'y connectent par Internet, en général avec un navigateur web. Toute la structure sous-jacente, les intergiciels (middleware), les logiciels et les données des applications se trouvent dans le centre de données du fournisseur de services. Le fournisseur du service gère le matériel et les logiciels, et avec un contrat de niveau de service approprié, il peut aussi assurer la disponibilité et la sécurité de l'application et de vos données. Le SaaS permet à votre organisation de mettre rapidement en œuvre une application, avec un niveau d'investissement minimal.[3]

1.4.4. L'informatiqueserverless(sans serveur) :

L'informatique serverless permet aux développeurs de créer des applications plus rapidement en éliminant la nécessité de gérer l'infrastructure. Avec les applications serverless, le fournisseur de services cloud fournit, met à l'échelle et gère automatiquement l'infrastructure requise pour exécuter le code.

Pour comprendre ce qu'est par définition l'informatique serverless, il est important de noter que les serveurs exécutent toujours le code. Le terme serverless vient du fait que les tâches associées au provisionnement et à la gestion de l'infrastructure sont invisibles pour le développeur. Cette approche permet aux développeurs de se concentrer davantage sur la logique métier et de mieux valoriser leur cœur de métier. L'informatique serverless aide les équipes à augmenter leur productivité et à commercialiser plus rapidement les produits. Elle permet également aux entreprises d'optimiser leurs ressources et de rester concentrées sur l'innovation.

1.5. Types de déploiement dans le Cloud Computing:

Tous les Cloud ne sont pas identiques et aucun type de Cloud Computing ne convient à tout le monde. Plusieurs modèles, types et services différents ont évolué pour vous aider à trouver la solution adaptée à vos besoins.

Vous devez commencer par déterminer le type de déploiement cloud ou d'architecture de Cloud Computing sur lequel vos services Cloud seront implémentés. Il existe trois modes de déploiement de services cloud : le cloud public, le cloud privé et le cloud hybride.

1.5.1.Cloud public:

Un cloud public est détenu et exploité par un fournisseur de services cloud tiers, qui fournit des ressources informatiques, telles que des serveurs et du stockage, sur Internet. Microsoft Azure est un exemple de cloud public. Dans un cloud public, tout le matériel, les logiciels et l'infrastructure appartiennent au fournisseur de cloud. Vous pouvez accéder à ces services et gérer votre compte via un navigateur Web.

Il peut également être défini comme : Un cloud public est l'ensemble des services informatiques proposés par des fournisseurs tiers sur l'Internet public, accessibles à toute personne souhaitant l'utiliser ou l'acheter. Ces services peuvent être gratuits ou vendus à la demande, de sorte que les clients n'ont à payer que pour les cycles de processeur, le stockage ou la bande passante qu'ils consomment.

Contrairement à un cloud privé, un cloud public permet aux entreprises d'économiser sur les coûts d'achat, de gestion et de maintenance de leur infrastructure matérielle et applicative sur site, car le fournisseur de services cloud est chargé d'assurer la gestion et la maintenance du système. Le cloud public peut également être déployé plus rapidement que l'infrastructure sur site, fournissant une plate-forme qui est presque infiniment évolutive. Chaque employé de l'organisation peut utiliser la même application depuis n'importe quel bureau ou service, en utilisant n'importe quel appareil de son choix tant qu'il dispose d'une connexion Internet. Bien que des problèmes de sécurité aient été soulevés concernant les environnements de cloud public, qui sont correctement mis en œuvre, un cloud public peut être aussi sûr que la mise en œuvre

d'un cloud privé mieux géré tant que le fournisseur utilise des technologies de sécurité appropriées, telles que les systèmes de détection et de prévention des intrusions (IDPS) . [3]

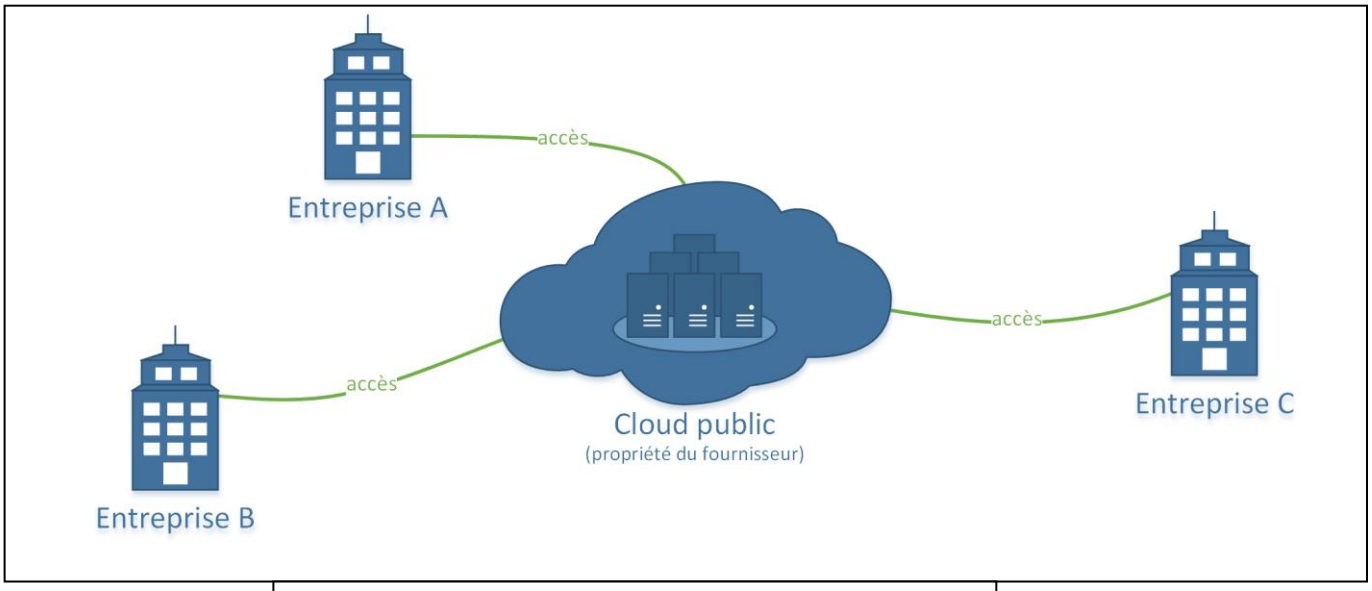


Figure 3 : Modèle de déploiement d'un Cloud public.[37]

1.5.2. Cloud privé :

Le cloud privé est l'ensemble des ressources de cloud computing utilisées de façon exclusive par une entreprise ou une organisation. Le cloud privé peut se trouver physiquement dans le centre de données local de l'entreprise. Certaines entreprises paient également des fournisseurs de services pour qu'ils hébergent leur cloud privé. Le cloud privé est un cloud dans lequel les services et l'infrastructure se trouvent sur un réseau privé.

L'expression « cloud privé » se définit comme un ensemble de services de calcul accessibles via Internet ou un réseau interne privé à un ensemble restreint d'utilisateurs sélectionnés plutôt qu'au grand public. Également appelé cloud interne ou cloud d'entreprise, le cloud computing privé offre aux entreprises bon nombre des avantages d'un cloud public (dont le libre-service, l'extensibilité et l'élasticité), auxquels s'ajoutent les possibilités de contrôle et de personnalisation que permettent des ressources dédiées sur une infrastructure de calcul hébergée localement. De plus, les cloud privés offrent un niveau de sécurité et de confidentialité supérieur résultant des pares-feux de l'entreprise et de l'hébergement interne, qui garantissent que les opérations et les données sensibles ne sont pas accessibles à des fournisseurs tiers. Un

inconvénient est qu'il incombe au service informatique de l'entreprise de maîtriser les coûts et d'assurer la gestion du cloud privé. Ainsi, un cloud privé occasionne les mêmes dépenses en personnel, gestion et maintenance que la propriété d'un centre de données traditionnel.

Deux types de services cloud peuvent être livrés via un cloud privé. Le premier, nommé IaaS (Infrastructure as a Service, infrastructure en tant que service), permet à une entreprise d'utiliser des ressources d'infrastructure, par exemple en matière de calcul, de réseau et de stockage, en tant que service. Le second, nommé PaaS (Platform as a Service, plateforme en tant que service), permet à une entreprise de proposer absolument tout, des applications cloud les plus simples aux applications professionnelles les plus sophistiquées. Un cloud privé peut également être combiné avec un cloud public afin de constituer un cloud hybride permettant à l'organisation de tirer parti de la solution cloud bursting (éclatement de cloud) pour libérer de l'espace et étendre les services de calcul au cloud public lorsque la demande explose.[3]

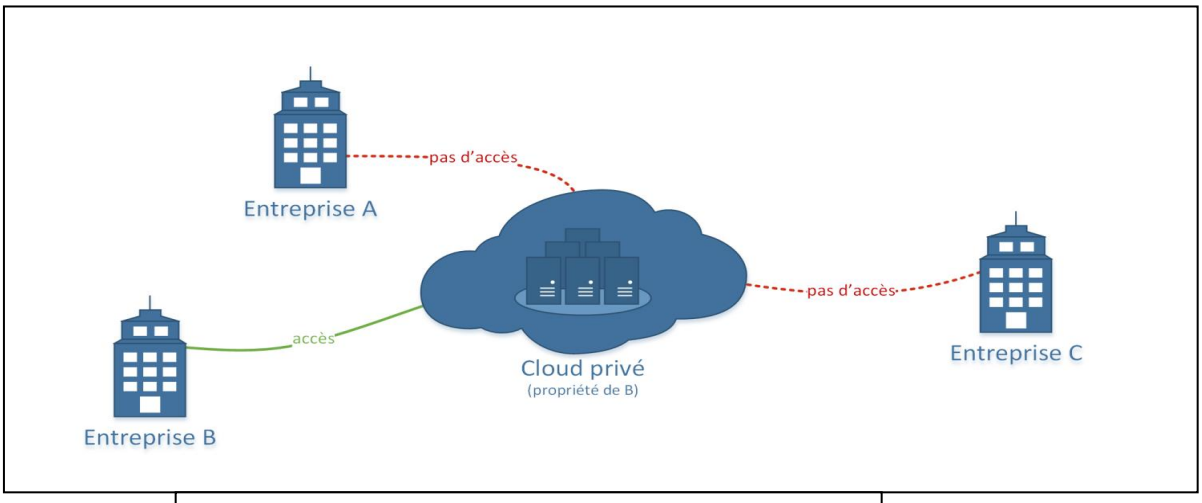


Figure 4 : Modèle de déploiement d'un Cloud privé.[37]

1.5.3. Cloud hybride :

Le cloud hybride combine des clouds publics et privés, liés à une technologie qui leur permet de partager des données et des applications. En permettant aux données et aux applications de se déplacer entre les Cloud privés et publics, un cloud hybride offre à votre entreprise une plus

grande flexibilité, plus d'options de déploiement et une infrastructure existante, une sécurité et une conformité améliorées. Il s'agit d'un environnement informatique qui combine un centre de données sur site (également appelé cloud privé) avec un cloud public, permettant de partager des données et des applications entre elles. Selon certaines personnes, un cloud hybride comprend des configurations « multicloud » où une organisation utilise plusieurs clouds publics en plus d'un centre de données sur site.[3]

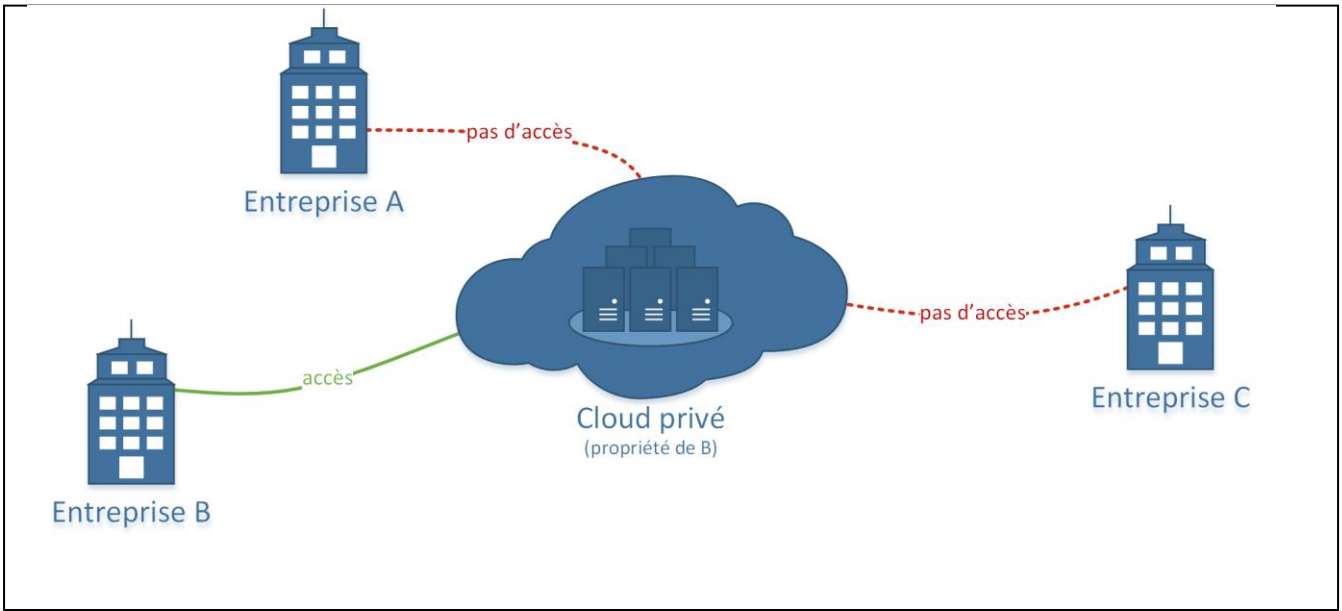


Figure 5 : Modèle de déploiement d'un Cloud hybride [37]

1.5.4. Cloudcommunautaire :

En matière de Cloud, il n'y a pas de solution générique pouvant convenir à tous. Certaines de vos applications doivent pouvoir évoluer rapidement pour répondre aux besoins de calcul ou de stockage dynamiques. D'autres doivent peut-être être hébergées dans certaines zones géographiques afin de respecter les réglementations sur la confidentialité des données ou de réduire la latence pour les utilisateurs finaux. D'autres encore, comme les applications existantes ou les charges de travail stratégiques, doivent être hébergées sur votre infrastructure sur site.

Une architecture multicloud est une architecture qui associe des ressources provenant de plusieurs types de Cloud public ou privé. Par exemple, vous pouvez choisir un fournisseur de services de Cloud public en fonction d'un accord sur les niveaux de service et d'une structure tarifaire spécifiques, et un autre en fonction de sa situation géographique. De même, vous

pouvez disposer de Clouds privés à différents endroits. Une stratégie multicloud vous donne la liberté et le contrôle vous permettant d'utiliser vos ressources disponibles sans compromis.[4]

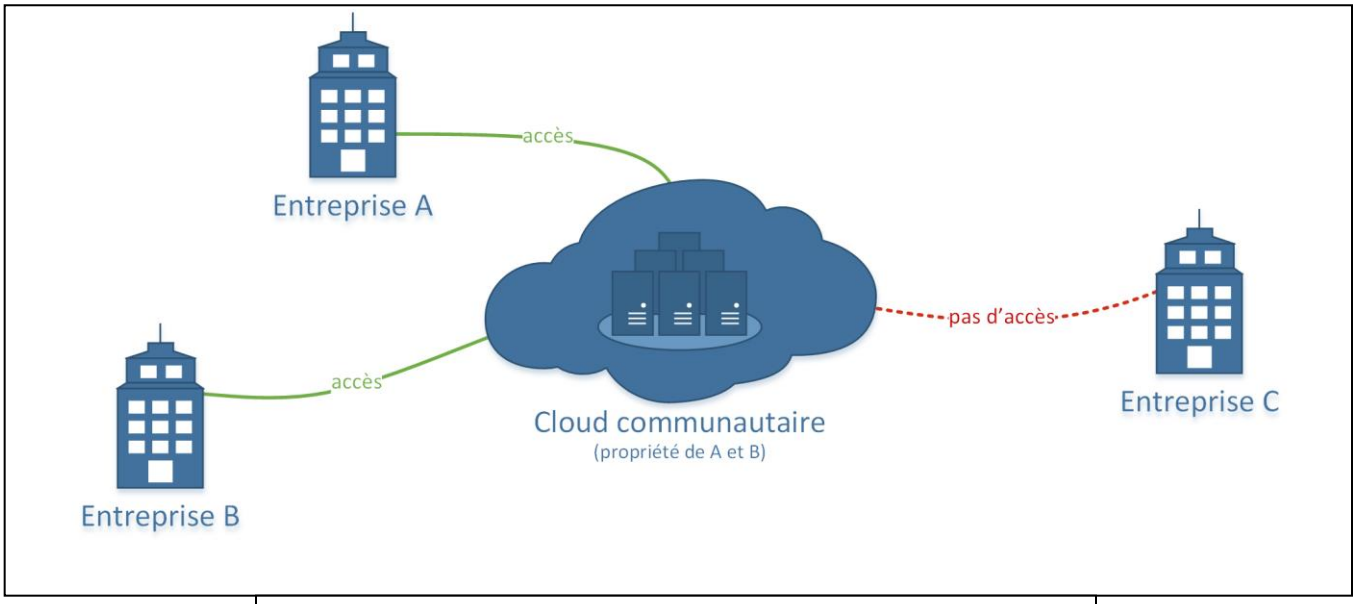


Figure 6 : Modèle de déploiement d'un Cloud communautaire. [37]

1.6. Applications :

Les logiciels et applications cloud computing ont connu une progression fulgurante ces dernières années, générant un chiffre d'affaires de plus de 219 milliards de dollars au niveau mondial en 2016. La part du Saas (Software as a Service) est majoritaire et devrait, grâce à une croissance stable, générer à elle seule près de 100 milliards de dollars à l'horizon de 2020.[5]

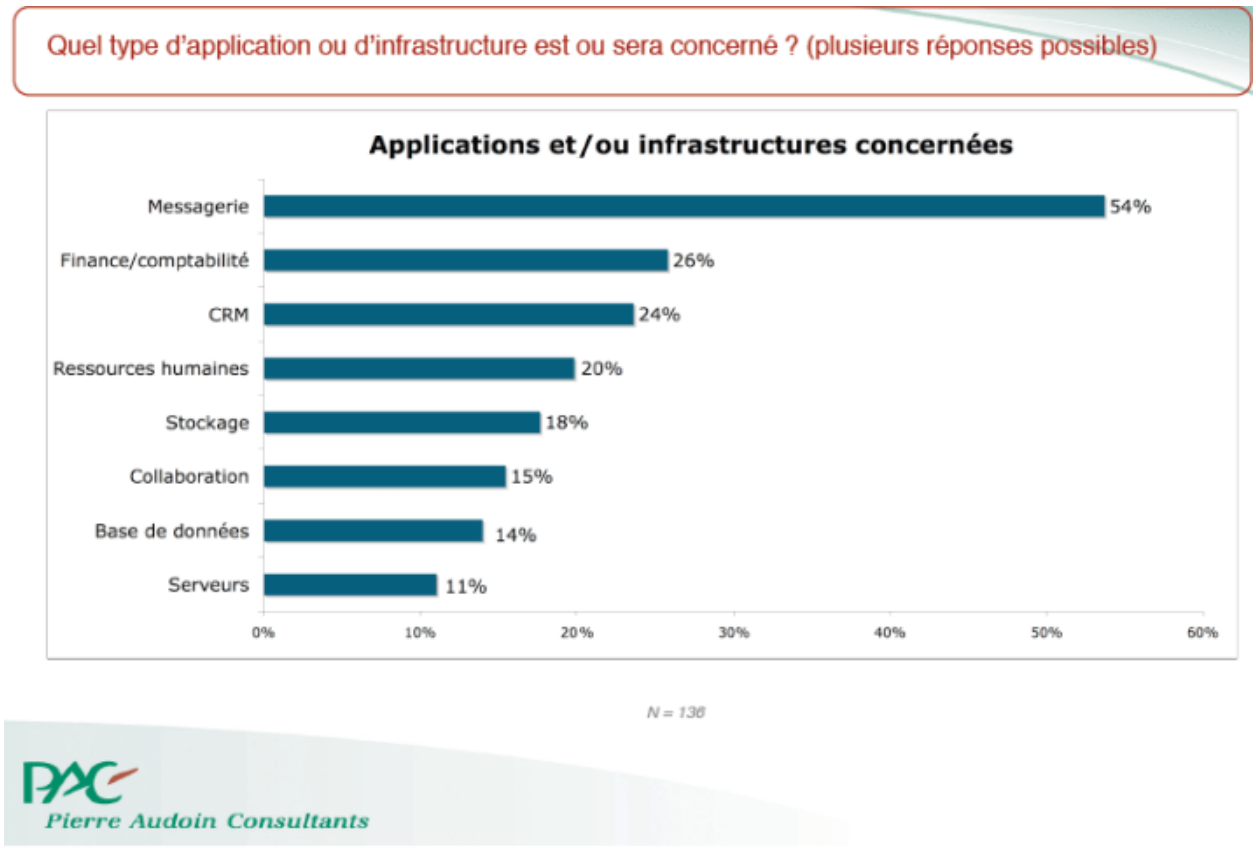


Figure 7 : type d'application ou d'infrastructure est ou sera impliqué dans le cloud computing. [5]

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

- **Les Avantages :**

-Flexibilité

- Évolutivité : l'infrastructure cloud à la demande est proportionnelle à la charge.

- Options de stockage : cela dépend de leurs besoins en matière de sécurité et d'autres considérations.

- Options de contrôle : Une entreprise peut choisir son propre niveau de contrôle via des options « as a service » : SaaS, PaaS et IaaS.

- Sélection d'outils.

- Fonctions de sécurité : Grâce au cloud privé virtuel, au chiffrement et aux clés API**.**

- Efficacité

- Accessibilité.

- Rapidité de mise sur le marché.

- Sécurité des données.

- Économies sur l'équipement : Le cloud computing utilise des ressources distantes, permettant aux entreprises d'économiser sur le coût des serveurs et autres équipements.

- Mode de tarification : Vous ne payez qu'en fonction de ce que vous consommez.

-Valeur stratégique :

- Simplifiez le travail.

- Mises à jour régulières.

- Collaboration : avec un accès mondial, les membres de l'équipe peuvent collaborer à distance.

- Avantage concurrentiel : En licenciant du personnel informatique pour gérer l'infrastructure.[6]

- **Inconvénients :**

-La sécurité informatique : La protection et la confidentialité des données hébergées par des fournisseurs cloud restent les deux principales craintes des entreprises.

Pourtant, Gartner affirme que « les services desClouds publics offerts par les principaux fournisseurs sont sécurisés. La vraie difficulté est de les utiliser de manière sécurisée ».

Cette nuance est essentielle : les entreprises doivent en effet adopter des bonnes pratiques (chiffrement des informations, gestion des accès…) en matière de cyber sécurité.

- Un usage qui peut devenir onéreux : Le mode « pay-as-you-go » est pratique, mais il peut s'avérer onéreux si celui-ci n'est pas maitrisé en interne. Laissé en libre-service sans contrôle des coûts, il peut s'avérer exorbitant.

- La reprise de l'existant : Certaines applications ne supportent pas la virtualisation et nécessitent d'être sur des serveurs physiques dédiés.

- Les problématiques de Licences logicielles : Certaines licences ne sont pas réutilisables. L'entreprise doit alors racheter des licences ou migrer vers une nouvelle version.[7]

1.8. Conclusion :

Dans ce chapitre, nous avons couvert les concepts de base du cloud computing.

Nous avons donné des définitions de ce concept et de ses caractéristiques fondamentales. Nous avons également fourni un aperçu détaillé de l'approche du cloud computing, en plus des quatre modèles possibles : cloud privé, cloud public, cloud communautaire et cloud hybride.

En fin de compte, les utilisateurs finaux trouvent que cette technologie est une bonne option pour utiliser les Services. Malgré toutes ces solutions produites dans le cloud, il y a encore des limites. Nous avons présenté les principaux défis auxquels la technologie du cloud computing doit faire face pour améliorer la qualité des services fournis aux utilisateurs.

Chapitre 2 :

Contrôle d'accès

2. Le contrôle d'accès :

2.1. Introduction :

Le contrôle d'accès est la première ligne de défense de la chaîne de sécurité de votre site, qu'il s'agisse de personnes ou de véhicules. Securitas Technologies applique les systèmes de contrôle d'accès les plus adaptés à vos besoins pour sécuriser tous vos flux.

En complément des protections mécaniques nécessaires pour améliorer la sécurité, nos solutions de contrôle d'accès augmentent considérablement le niveau de sécurité de votre organisation.

Utilisant des technologies avancées pour identifier les personnes, analyser les données ou gérer à distance, une variété des services estfournie pour soutenir la politique de sécurité des utilisateurs.

Le contrôle d'accès désigne les différentes solutions techniques qui permettent de sécuriser et gérer les accès physiques à un bâtiment ou un site, ou les accès logiques à un système d'information. On distingue ainsi le contrôle d'accès physique et le contrôle d'accès logique.

Il existe une grande variété de méthodes, de modèles, de technologies et de capacitésadministratives utilisés pour proposer et concevoir des systèmes de contrôle d'accès. Ainsi,chaque système de contrôle d'accès à ses propres attributs, méthodes et fonctions, qui découlentd'une politique ou d'un ensemble de politiques.

La politique de sécurité se compose de trois sous-politiques de contrôle : AccèsPhysique, administratif et logique. Dans ce chapitre, nous parlerons de politique de contrôle d'accès logique et aux modèles de protection et mécanismesnécessaires pour la réaliser.

2.2. Définition :

Au niveau le plus élémentaire, le contrôle d'accès est un moyen de contrôler qui pénètre dans un lieu et quand. La personne qui entre dans ce lieu peut être un employé, un sous-traitant ou un visiteur.

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.

Elle peut être à pied, conduire un véhicule ou utiliser un autre moyen de transport. Le lieu dans lequel elle entre peut s'apparenter à un site, un bâtiment, une pièce ou même juste une armoire.

Les entreprises sont de plus en plus amenées à tracer leurs accès informatiques à l'aide d'un Reportant des Droits d'Accès.

Le contrôle d'accès désigne les différentes solutions techniques qui permettent de sécuriser et gérer les accès physiques à un bâtiment ou un site, ou les accès logiques à un système d'information. On distingue ainsi le contrôle d'accès physique et le contrôle d'accès logique.

Lecontrôle d'accès physique est un dispositif permettant un accèscontrôlé une machine ou des équipementsspécifiques.[8]

L'utilisation d'une clé physique est la manière la plus simple de contrôler l'accès d'une porte, et la méthode la plus utilisée par de nombreuses petites organisations. Cependant, même pour une petite entreprise, l'utilisation de clés mécaniques présente plusieurs défauts et limites, surtout en cas de croissance de votre entreprise.

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

Le contrôle d'accès logique est un système de contrôle d'accès à un système d'information. Il est souvent couplé avec le contrôle d'accès physique et permet de restreindre le nombre d'utilisateurs du système d'information. [9]

Afin de protéger votre entreprise, quelles que soient sa taille et son activité, la mise en place d'un système de contrôle d'accès est indispensable. Il assure efficacement la sécurité de vos locaux et de vos employés.

Ainsi, vous pouvez gérer l'accès aux zones sensibles et précieuses (coffre-fort, casier robuste, salle de stockage, salle des serveurs et inventaire) de votre organisation ainsi que vos espaces de bureau. Sur les différents services de votre entreprise.[10]

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

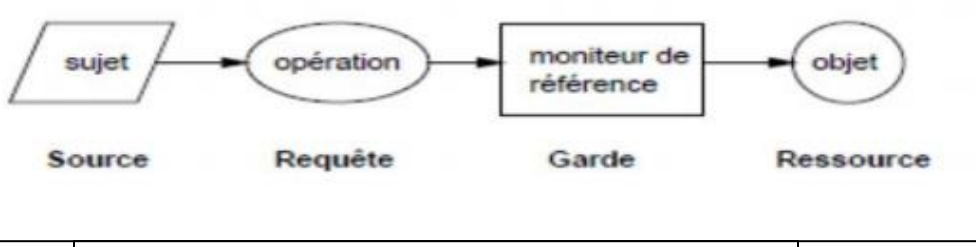


Figure 8 : Le modèle de base du contrôle d'accès [39]

- Complément :

Les droits d'accès peuvent être symboliquement représentés dans une matrice de droits d'accès dont les lignes représentent les sujets et les colonnes représentent les objets. Une cellule de la matrice contient donc les droits d'accès d'un sujet sur un objet. La matrice est gérée conformément aux règles définies dans la politique de sécurité. D'une manière générale, les règles de la politique de sécurité sont spécifiées en termes de: Permissions (par exemple tout médecin a le droit d'accéder aux dossiersmédicaux de ses patients), Interdictions (par exemple, les médecins n'ont pas le droit d'effacer desdiagnostics déjà établis), Obligations (les médecins sont obligés de conserver les dossiers médicaux pendant la durée fixée par la loi)

- Concepts de base :

Le contrôle d'accès consiste à gérer et à vérifier les droits d'accès, en fonction des règles spécifiées dans la politique de sécurité. On dit qu'un sujet possède un droit d'accès sur un objet si et seulement s'il est autorisé à effectuer la fonction d'accès correspondante sur cet objet.

Sujets :

Ce sont les entités actives du système qui : Demandent des droits d'accès correspondant à l'autorisation d'exécuter des actions sur les objets Incluent toujours les utilisateurs du système Incluent souvent les processus en cours d'exécution pour le compte desutilisateurs.

Les objets :

Ce sont les entités passives du système. Elles contiennent les informations et ressources à protéger.

Les actions :

Ce sont les moyens permettant aux sujets de manipuler les objets du système

Principe du moindre privilège :

Ce principe consiste à affecter des préférences aux utilisateurs de façon à ce qu'ils aient pas plus de permissions que nécessaires pour effectuer leurs tâches l'application stricte de ce principe entraîne : 1. Différents niveaux de permissions, 2. Différents niveaux de permissions à des moments différents,3. Permissions limitées dans le temps.

Exemple : - Domaine médical

Sujets : médecins, infirmiers, secrétaire médicale.

Objets : patients, dossier médical ou administratif.

Actions : consulter, modifier le dossier médical d'un patient, créer le dossier médical d'un patient.

2.3. Les modèles des contrôles d'accès :

. Modèles de contrôle d'accès classiques :

En raison des différences dans les exigences pour les politiques de sécurité militaire et commerciale, deux types de politiques distinctes ont été élaborés, deux modèles différents de contrôle d'accès :Discretionary Access Control (DAC) et Mandatory Access Control (MAC). Ces modèles ont un certain nombre de failles, ce qui a conduit à la proposition d'autres modèles tels que le contrôle d'accès basé sur le rôle (RBAC).

2.3.1. Discretionary Access Control (DAC):

Le Contrôle d'accès discrétionnaire (DAC pour Discretionaryaccess control) est un genre de contrôle d'accès, défini par le Trusted Computer System Evaluation Criteria (TCSEC) comme « des moyens de limiter l'accès aux objets basés sur l'identité des sujets ou des groupes auxquels ils appartiennent. Le contrôle est discrétionnaire car un sujet avec une certaine autorisation d'accès est capable de transmettre cette permission (peut-être indirectement) à n'importe quel autre sujet (sauf restriction du contrôle d'accès obligatoire). »

Définition élargie :

Le contrôle d'accès discrétionnaire est généralement défini par opposition au contrôle d'accès obligatoire (qui est un contrôle d'accès non discrétionnaire parfois nommé égal). On peut également dire qu'un système a le contrôle d'accès discrétionnaire ou purement discrétionnaire comme manière de dire que le système manque d'un contrôle d'accès obligatoire. D'une part, on peut dire que des systèmes mettent en application l'IMPER et le DAC simultanément, où le DAC se rapporte à une catégorie de contrôles d'accès que les sujets peuvent transférer parmi l'un l'autre, et où l'IMPER se rapporte à une deuxième catégorie des contrôles d'accès qui impose des contraintes à la première.

La théorie et la pratique :

Cependant, la signification de la limite n'est pas dans la pratique aussi définie que la définition donnée dans la norme TCSEC. Par exemple, la limite est utilisée généralement dans les contextes qui supposent que, sous DAC, chaque objet a probablement un propriétaire qui commande les permissions d'accéder à l'objet, parce que beaucoup de systèmes appliquent le DAC en utilisant le concept d'un propriétaire. Mais la définition de TCSEC n'indique rien au sujet des propriétaires, techniquement un système de contrôle d'accès ne doit pas avoir un concept de propriétaire pour concilier la définition du TCSEC et du DAC.

Autre exemple, des possibilités sont parfois décrites en tant que commandes discrétionnaires parce qu'elles permettent à des sujets de transférer leur accès à d'autres sujets, quoique la sécurité ne soit fondamentalement pas au sujet de l'accès basé sur l'identité des sujets.[11]

2.3.2. Mandatory Access Control (MAC) :

Pour protéger les données ou les paramètres d'un système contre tout accès non autorisé ou contre une modification malveillante, les entreprises attribuent généralement à tout utilisateur un accès restreint aux fichiers dont il a besoin dans l'exercice de ses fonctions. Définir et attribuer ces droits d'accès reste cependant une tâche complexe pour les petites et moyennes entreprises. Une entreprise se décline généralement en différents services, comme par exemple le service financier, le service commercial et les ressources-humaines. Le personnel de chacun de ces services à besoin d'un droit d'accès spécifique, lui permettant de réaliser les tâches qui lui sont confiées. Certains employés ont par ailleurs besoin de droits d'accès élargis pour assumer certaines responsabilités ou fonctions spécifiques. Pour pouvoir mettre en œuvre et contrôler ces différents droits d'accès, différentes stratégies de sécurité ont été conçues, parmi lesquelles le Mandatory Access Control (MAC) ou Contrôle d'Accès obligatoire. Grâce à une telle stratégie, chaque utilisateur bénéficie d'un accès restreint aux seules ressources dont il a véritablement besoin. Le terme « obligatoire » implique cependant que le contrôle d'accès repose sur des règles clairement définies qu'il est impératif de respecter.

Ce modèle est utilisé pour mettre en œuvre et gérer les contrôles d'accès aux données, y compris le contrôle d'accès obligatoire. Ce modèle est également utilisé dans des industries telles que les agences militaires et gouvernementales, des domaines où il est nécessaire de protéger les données contre les abus. Ci-dessous, nous expliquons le fonctionnement de ce contrôle d'accès structuré et vous présentons en même temps ses forces et ses faiblesses.

L'organisation des droits d'accès dans le MAC :

Les droits d'accès sont administrés de façon centralisée. Généralement, la personne qui assume cette fonction est quelqu'un qui connaît bien la répartition des tâches au sein de l'entreprise ou de l'organisation. Cela permet ainsi à chacun d'exercer pleinement ses fonctions sans être restreint dans son travail en raison de droits manquants. Dans une entreprise, c'est généralement l'administrateur système qui assume cette tâche. La mise en œuvre et la réactualisation permanente se fait généralement au moyen du système d'exploitation **ou** d'un noyau de sécurité**.** Si un utilisateur tente d'accéder aux données, le système valide ou non son

accès. L'avantage de cette application automatique est qu'elle exclut efficacement les accès malveillants.

La validation des différents droits d'accès est définie sur la base des facteurs suivants **:**

- Les utilisateurs et les processes
- Les objets (les ressources auxquelles on accède)
- Des règles et des propriétés : catégorisations, labels, mots de passe

Le contrôle d'accès obligatoire repose sur une approche hiérarchique : chaque élément constituant un système de fichiers se voit assigner un niveau de sécurité qui dépend du degré de confidentialité des données. Parmi les niveaux de sécurité, on a typiquement les mentions « confidentiel » ou « top secret ». On attribue ce type de sécurité également aux utilisateurs et aux appareils. Si un utilisateur cherche à accéder à une ressource, le système vérifie automatiquement si l'accès est autorisé ou non. Par ailleurs, on attribue une catégorie à toutes les informations et à tous les utilisateurs. Le système vérifie aussi automatiquement cette adéquation lors d'une tentative d'accès. Pour pouvoir accéder aux données, l'utilisateur doit répondre aux deux critères : le niveau de sécurité et la catégorie.

Les formes du MAC :

On distingue deux formes de Mandatory Access Control :

- *Les systèmes de sécurité à multi-niveaux :*

Ce modèle est la **forme simple et originale** de MAC, composée d'une **suite verticale de niveaux de protection et de sécurité**. Les informations ne circulent qu'à l'intérieur de ces domaines. Un niveau de sécurité est également assigné aux utilisateurs. Ils peuvent ainsi accéder à leur niveau, et aux niveaux inférieurs.

- *Des systèmes de sécurité multilatéraux :*

Ces systèmes sont plus **complexes** et permettent des accès sur la base de segments qui constituent un ensemble. Ces segments ont à leur tour des niveaux de sécurité et des mots de passe. Il en résulte un **système de sécurité horizontale** qui renferme des **niveaux de sécurité verticaux**.

Les avantages et les inconvénients du MAC :

Le Mandatory Access Control ou Contrôle d'Accès obligatoire fait partie des contrôles d'accès les plus sécurisés, car on ne peut quasiment pas l'enfreindre. À la différence du RBAC, le système MAC ne permet aux utilisateurs d'apporter aucune modification. Le contrôle et l'attribution des droits d'accès se font de façon parfaitement automatique par le système. De ce fait, le Mandatory Access Control offre un haut niveau de **confidentialité**. Le système est par ailleurs caractérisé par une excellente **intégrité**. Sans autorisation préalable, il est impossible de modifier les données qui sont donc parfaitement à l'abri d'un usage malveillant.

La mise en œuvre d'un contrôle d'accès obligatoire demande cependant une **planification détaillée** en amont, et son administration après implémentation impose un suivi important. Chaque assignation de droit sur des objets ou des utilisateurs demande une vérification et une réactualisation permanentes. Il en va de même des tâches d'entretien courantes, parmi lesquelles on compte l'ajout de nouvelles données, de nouveaux utilisateurs, la prise en compte de modifications dans la catégorisation ou dans la classification. Généralement, l'attribution de ces droits repose sur une seule personne. Cela garantit certes un niveau de sécurité élevé, mais représente souvent une dose de travail importante pour l'administrateur.

Domaines d'utilisation de MAC :

Le haut niveau de confidentialité et d'intégrité du Mandatory Access Control ou Contrôle d'Accès obligatoire a imposé son usage dans les secteurs utilisant des **données sensibles**, notamment dans les **environnements exposés sur le plan de la sécurité**. C'est le cas par exemple de l'armée, des autorités gouvernementales, du secteur de la politique, du commerce extérieur, du domaine de la santé et du service de renseignements. Il n'est pas rare non plus que des entreprises ordinaires aient recours au MAC. Le système d'exploitation **Security-Enhanced Linux (SELinux)** est par exemple bâti sur l'implémentation d'un contrôle d'accès obligatoire dans le noyau de Linux. [12]

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

Pour fournir des droits d'accès à l'utilisateur, il est important de connaître les responsabilités de l'utilisateur assignées par l'organisation. Mais dans Les droits d'utilisation des données du DAC jouent un rôle important, ne constituent pas un bon et dans MAC, les utilisateurs doivent prendre

des habilitations de sécurité et les objets ont besoin de classifications de sécurité. RBAC essaie de réduire l'écart en combinant les contraintes organisationnelles forcées avec flexibilité des autorisations explicites [14].

RBAC principalement utilisé pour contrôler l'accès à la Ressources d'ordinateur.

RBAC est une méthode très utile pour contrôler ce que type d'informations que les utilisateurs peuvent utiliser sur l'ordinateur, le programmes que les utilisateurs exécutent, et les changements que les utilisateurs peut faire. Dans RBAC, les rôles des utilisateurs sont attribués de manière statique, qui n'est pas utilisé dans un environnement dynamique. Il est plus difficile de modifier les droits d'accès de l'utilisateur sans modifier les rôles spécifiés de l'utilisateur. RBAC est généralement un accès préférable modèle de contrôle pour le domaine local. En raison du rôle statique affectation, il n'a pas de complexité. Il a donc besoin de peu d'attention pour l'entretien [15][16].

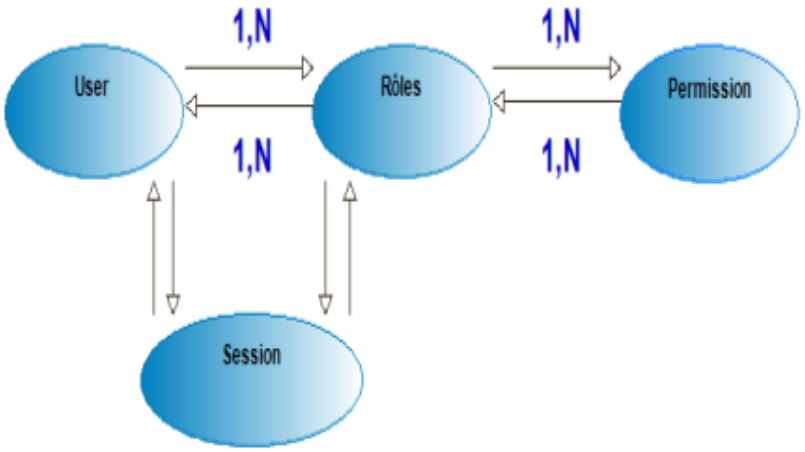


Figure 9 : Model RBAC.

Le rôle n'est rien d'autre que les abstractions du comportement de l'utilisateur et leurs tâches assignées. Ceux-ci sont utilisés pour attribuer le système ressources aux départements et à leurs membres respectifs. À assurer le contrôle d'accès avec la sécurité dans les particuliers systèmes

logiciels, il sera avantageux d'utiliser le concept de rôle. Cela réduit également le coût de la gestion de l'autorité [17].

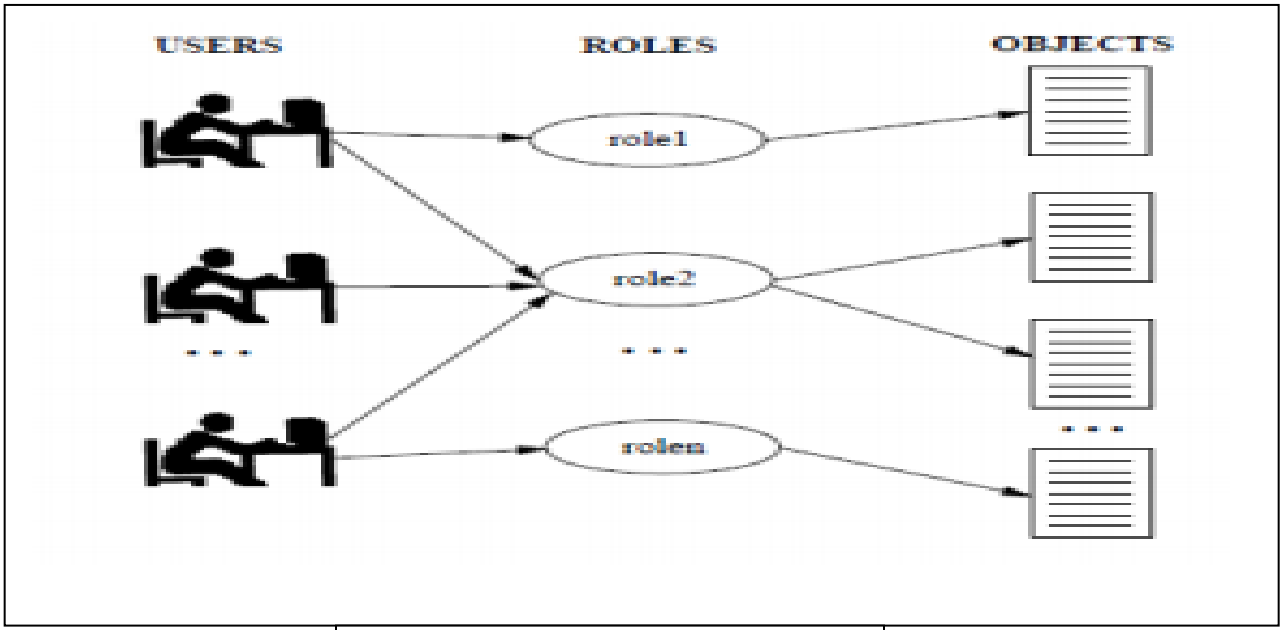


Figure 10: Role-basedaccess control. [38]

Essentiellement, dans les politiques de contrôle d'accès basées sur les rôles, il faut identifier les rôles dans le système, un rôle peut être défini comme un ensemble de responsabilités et d'actions associées à une activité de travail particulière. Dans un modèle de sécurité de contrôle d'accès, un rôle est considéré comme un droit d'accès lié au travail qui peut être accordé aux utilisateurs autorisés au sein d'une organisation. Il permet à l'utilisateur autorisé d'accomplir ses responsabilités associées.

En ce qui concerne le modèle RBAC, nous décrivons deux types de sujets : les utilisateurs liés au système et les transactions qui sontexécuté au nom de ces utilisateurs. Les utilisateurs peuvent accéder à des objets particuliers en exécutant des transactions sur cet objet. Une transaction peut être considérée comme un ensemble d'opérations exécutables qui provoqueconsommation d'une ressource système [16]. Par exemple, dans une banque, les caissiers sont autorisés à exécuter une transaction de dépôt et de retrait, pour cela, il nécessite un accès en lecture et en écriture aux champs spécifiques dans

Compte. Un superviseur de compte a les mêmes droits ou plus pour effectuer des opérations de correction.

La protection du système est basée sur l'autorisation qui décrit un droit d'accès donné à un objet particulier ou à un ensemble d'objets. Dans le modèle RBAC, nous sommes confrontés à un accès non autorisé aux ressources et aux données du système informatique [29].

Puisque nous n'avons considéré que les droits d'accès dont les utilisateurs avaient besoin pour exécuter une transaction particulière sur un objet particulier à partir de l'ensemble défini d'objets.

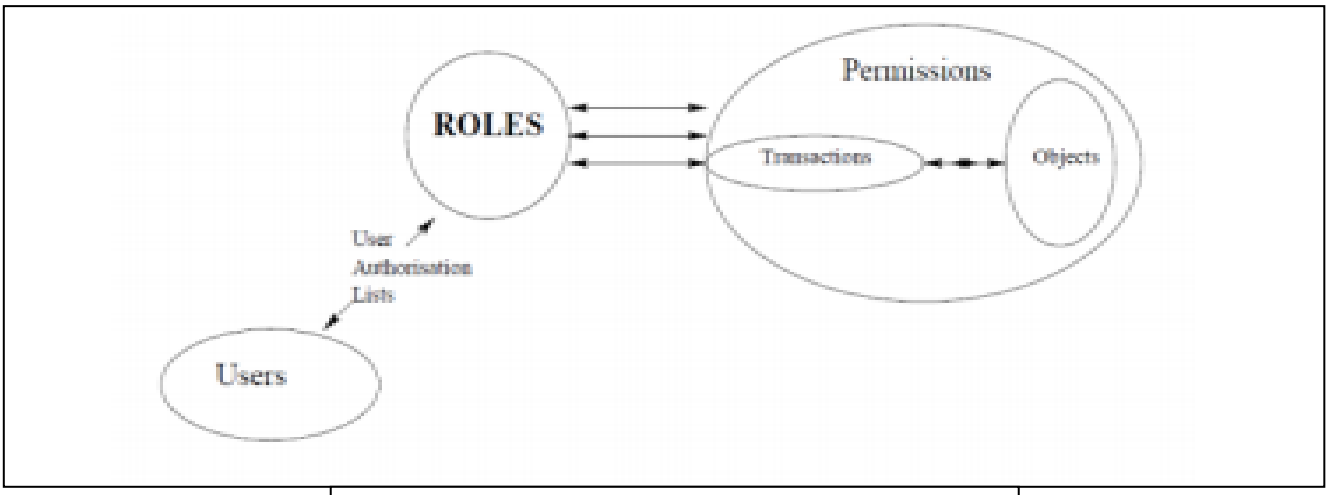


Figure 11 : Mappage utilisateur-rôle-autorisation. [40]

Une autorisation p est une paire < trans, objset>, où trans représente la transaction qui s'exécute sur l'ensemble d'objets qui est objset. Considérez P indique l'ensemble universel d'autorisations, Trans indique l'ensemble universel de transactions et Obj indique l'ensemble d'objets. Nous pouvons définir l'association entre permission/transaction et permission/objet avec les fonctions suivantes.

TransP(p) : P → Tr, Il donne la transaction associée à l'autorisation spécifiée p.

ObP (p) : P → 2^{obj}, Il donne l'ensemble d'objets associé à l'autorisation spécifiée p.

Un rôle est créé en collectant des autorisations selon le les exigences fonctionnelles et logiques de ce rôle devraient représenter. Chaque rôle a un nom qui lui est associé et il identifie de manière unique ce rôle dans le système.

Un rôle r est une paire de < rn, pset>, où rn indique le rôle Name et pset indiquent l'ensemble des autorisations de rôle.

Le mappage entre les rôles et les autorisations peut être défini avec la fonction suivante :

PR(r) : R → 2^p, Il donne l'ensemble d'autorisations associé au rôle spécifié r. Ici, R indique l'ensemble universel des rôles.

Lors de l'attribution des autorisations aux rôles, il est nécessaire de s'assurer que le principe du moindre privilège, c'est-à-dire que chaque rôle ne devrait avoir quedroits requis pour ses exigences fonctionnelles.

Avantages et Inconvénients de RBAC :

Avantage :

-Le contrôle d'accès basé sur les rôles présente de nombreux avantages.

- Certains d'entre eux sont décrits ci-dessous.

- Fournit une politique neutre/flexible.

- la contrainte des séparations des devoirs.

- Capacité d'exprimer DAC, MAC, et les politiques spécifique d'utilisateur en utilisant la hiérarchie des rôles et les contraintes.

Inconvénient :

Dans le modèle RBAC, il reste encore du travail à faire pour couvrir toutes les exigences qui peuvent représenter le monde réel scénario.

Définir les rôles dans un contexte différent est difficile et peut aboutir à une grande définition de rôle. Parfois, il produit plus rôles que les utilisateurs.

De nos jours, nécessitent des résultats à grain fin mais RBAC ne donne pas des résultats fins

[23].

Il maintient la relation entre les utilisateurs et ses rôles. Ça aussi maintient la relation entre les autorisations et les rôles.

Par conséquent, pour mettre en œuvre le modèle RBAC, les rôles doivent être attribués à l'avance et il n'est pas possible de modifier l'accès droits sans modifier les rôles.

. **Mécanismes de sécurité associés à RBAC :**

La séparation des tâches :

En particulier, la séparation des tâches est une exigence importante dans de nombreux systèmes commerciaux et l'une des caractéristiques les plus souhaitables du système RBAC. L'exclusion mutuelle des rôles est un moyen de mettre en œuvre des politiques de séparation des tâches. Cette partie Il explore certaines caractéristiques de l'exclusion mutuelle des rôles dans systèmes RBAC. (D'autres moyens de mettre en œuvre la séparation des tâches, tels que les séquences de transactions, n'ont pas été pris en compte ici.) Les résultats présentés dans le document sont utiles pour comparer l'exclusion mutuelle des rôles avec d'autres mécanismes potentiels pour mettre en œuvre la séparation des contrôles de tâches et pour comprendre Implications pour la mise en œuvre de différents types d'exclusion mutuelle.

***Séparation des tâches par le rôle Exclusion :**

Lorsqu'elle est mise en œuvre à l'aide de règles d'exclusion de rôles, la séparation des affectations peut être analysée selon au moins deux dimensions : lorsqu'elle est réciproque L'exclusion est appliquée et les opérations auxquelles elle est appliquée. Deux types d'exclusion mutuelle sont considérés, l'exclusion au moment de l'autorisation et l'exclusion au moment de l'exécution, selon que la règle d'exclusion mutuelle est appliquée au moment de la délégation de rôle, ou au moment de l'exécution, au cours d'une session utilisateur. Deux attributs supplémentaires {exclusion complète et exclusion partielle} indiquent si les rôles mutuellement exclusifs ne partagent aucun privilège ou partagent certains mais pas tous les privilèges.

***état de sécurité :**

Le but des règles de séparation des tâches est d'empêcher une personne d'effectuer toutes les parties d'un travail qui devrait nécessiter deux personnes ou plus, afin d'éviter la collusion ou la fraude. Par exemple, de nombreuses organisations exigent que les dépenses importantes soient demandées et approuvées par deux personnes distinctes. S'il n'y a que deux privilèges pour une telle tâche, alors chaque privilège peut être affecté à des rôles distincts et les rôles sont rendus mutuellement exclusifs. Si plus de deux franchises sont incluses, elles peuvent être divisées en deux rôles ou plus. Nous définissons une condition de sécurité qui doit être remplie pour garantir que les exigences de séparation des tâches ne soient pas violées. Allons C[t] : mission ! 2- Le privilège est une affectation de tâches qui nécessitent de séparer le devoir en ensembles de privilèges requis pour ces tâches. Ensuite, pour s'assurer que personne ne peut effectuer toutes les parties de la tâche, aucun utilisateur ne peut accéder à tous les privilèges en C[t].

Le terme « exclusion mutuelle » a une signification intuitive, mais certaines complications peuvent survenir lors de l'exploration des implications de l'exclusion des rôles dans le système RBAC.

Ces complications surviennent si des privilèges sont accordés à d'autres rôles qui peuvent ne pas être identifiés comme s'excluant mutuellement. Il faut veiller à ce qu'un certain ensemble de rôles ne permette pas à l'utilisateur d'avoir des privilèges qui doivent s'exclure mutuellement. Par conséquent, toute discussion sur l'exclusion de rôle doit tenir compte du fait que tous ou seulement certains des avantages du rôle sont privés d'un rôle mutuellement exclusif. . En théorie, les privilèges d'un rôle déterminé comme exclusif avec un autre rôle peuvent être mis à la disposition d'autres rôles qui ne font pas partie de la paire mutuellement exclusive. Une autre considération, par conséquent, est de savoir si les avantages peuvent être partagés à travers des rôles en dehors de la paire de rôles mutuellement exclusifs. Pour simplifier l'analyse, les privilèges qui nécessitent une séparation des tâches peuvent être attribués à des rôles uniques, puis ces rôles peuvent être hérités par d'autres. Ainsi, il existe de nombreuses alternatives au partage des privilèges, les principales étant :

Exclusion de durée d'autorisation/d'exécution

Exclusion complète/partielle

***propriétés de base :**

Théorème1: Si l'exclusion au moment de l'autorisation est vérifiée, alors l'exclusion au moment de l'exécution est maintenue.

Théorème 2 : Si l'exclusion complète est vérifiée, alors l'exclusion partielle est maintenue.

Théorème 3 : Les rôles mutuellement exclusifs ne peuvent pas être introduits dans l'ensemble actif. [13]

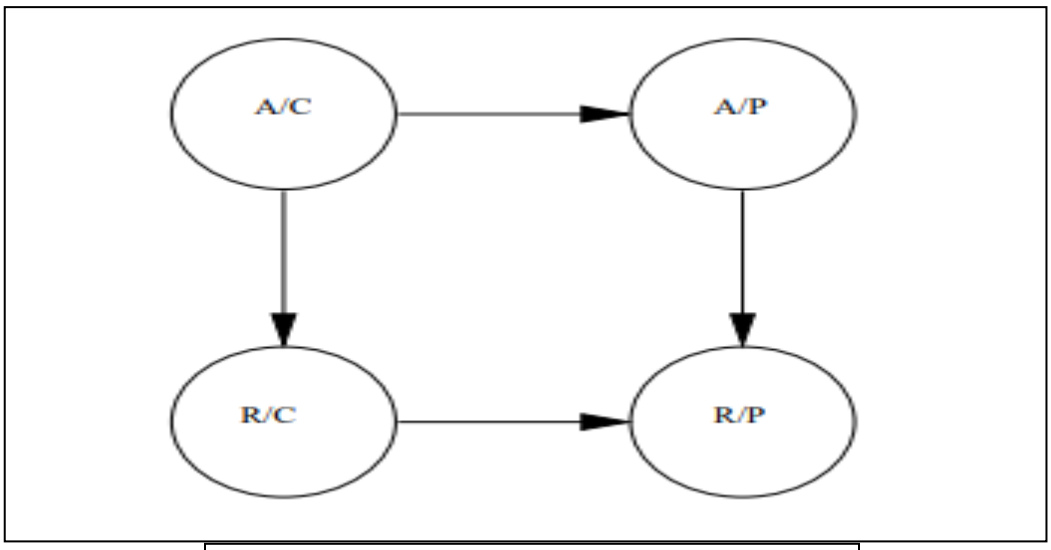


Figure 12 : Relations de séparation des taches. [41]

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

Dans ABAC, la décision est prise sur la base de conditions booléennes sur des valeurs d'attributs

Sujet Ressource et Action sont des catégories qui regroupent des attributs P.ex. Attributs du sujet pourraient être : âge, rôle, nom, type etc.…

Les attributs ont des valeurs P.ex. Nom(sujet)=Gervais, Genre(sujet)=masculin, Age(sujet)=25, Identif(action)=écriture, Type(action)=modification

La requête de contrôle d'accès est un ensemble de couples (attribut=valeur) – elle représente

l'état du demandeur

P. ex. (Nom (sujet)=Gervais) et (Age (sujet)= 25) et (Identif (action)=lecture) et (Identif (ressource)=livre RBAC)

Les règles de contrôle d'accès sont basées sur des cibles :

Représentées par des expressions booléennes – la logiquePermettre si (Type (sujet)= étudiant) et (Type (action)= consultation) et (Type (ressource)=livre en réserve) et (Temps (requête)= heures de travail).

Autre Exemple de Condition :

Le sujet qui fait requête est un étudiant de notre université il est localisé dans le bâtiment Taché ou Brault, pas St-Jérôme L'heure est entre 8h et 20h Il n'est pas dimanche.

Règles et politiques Les règles nous les avons vues :

Les politiques sont des ensembles de règles Normalement les règles dans une politique sont reliées par une idée commune

P.ex. on pourrait avoir des politiques :Accès à la bibliothèque virtuelle qui réunit toutes les règles qui déterminent l'accès à cette bibliothèque accès aux dossiers de comptabilité.

Éléments architecturels d'ABAC :

PEP: Policy EnforcementPoint :Donne ou refuse un accès.

PDP: Policy DecisionPoint :Prend la décision si l'accès doit être donné ou refuséUtilisant les politiques et règles qui sont enregistrées dans une base de données appelée Policy Store.

PAP: Policy Administration Point ; Gère le Policy Store : ajout, enlèvement de politique.

PIP: Policy Information Point - fournit les informations dont le PDP a besoin pour prendre ses

décisions

Les modèle d'attributs et les ontologies, v. aprèsL'état de l'environnement : P.ex. l'heure Location de l'usager ou de la ressource.

Modèle d'attributs :

Un système ABAC a un modèle qui connaît tous les attributs qui peuvent être utilisés, dans des requêtes ou dans des règles Dans le modèle, les valeurs peuvent être organisés en ontologies, p.ex. on peut savoir qu'un 'fichier d'inventaire' est un 'fichier de comptabilité'

Qu'ilsoit au niveau 'secret' qu'un 'acroread' est un type de 'read'

Qu'un 'professeur auxiliaire' est un 'professeur'

Qu'il y a différents types d'étudiants : à temps plein, à temps partiel, de 1er cycle, de 2ème cycle etc.

Les hiérarchies de rôles, s'il y en a, sont donc stockées dans cette base de données Ainsi que les niveaux d'autorisation s'il est désiré d'implémenter MAC.[14]

Avantages d'ABAC :

ABAC résout le problème d'attribution de rôle d'utilisateur qui se présente dans RBAC et au lieu de se concentrer sur les rôles, il se concentre sur les attributs d'un utilisateur pour attribuer les droits d'accès.

ABAC offre une plus grande flexibilité dans un environnement distribué, ouvert, partageable et dynamique où le nombre d'utilisateurs est très élevé. Par conséquent, ABAC est un modèle très flexible pour l'administration et il fonctionne très bien que RBAC.

ABAC prend également en charge l'accord global pour les attributs afin que les attributs fournis dans un domaine puissent être transmis à l'autre domaine au point d'interaction de domaine à domaine.

ABAC est également utilisé pour l'administrateur Web afin d'améliorer la structure du site Web.

Il fournit un stockage central pour les attributs des utilisateurs, il augmente l'interopérabilité et le partage entre plusieurs fournisseurs de services pour décider des droits des utilisateurs.[19]

Inconvénients de l'ABAC :

En raison de l'hétérogénéité des informations utilisateur, la complexité est accrue, par conséquent, pour résoudre ce problème, il faut une base de données centrale ayant tous les attributs dans le même format, représentant les sujets et les objets, perdant ainsi les avantages de la fonctionnalité ABAC flexible et dynamique. [20]

2.5. Conclusion :

Nous avons présenté un bilan des divers modèles formels de contrôle d'accès existants. Ces modèles ont été conçus pour traiter du contrôle d'accès dans des situations bien différentes. Ils introduisent de multiples concepts pour spécifier les politiques de contrôle d'accès. Le but de ces divers modèles est d'offrir la palette la plus vaste possible d'expression de politiques de contrôle d'accès. [21]

Chapitre 3 :

Conception et implémentation

1. Introduction :

Dans ce chapitre, au début, nous allons parler des étapes suivies pour concevoir et implémenter notre politique de sécurité qui repose sur le model ABAC (AttributeBased Access Control), les outils et les différents environnements de développement que nous avons utilisé, ensuite on va présenter et expliquer quelques fragments de code source (XACML, ALFA) qui sont situés dans l'annexe et expliquer notre politique d'accès.

2. Conception :

2.2. Architecture générale :

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

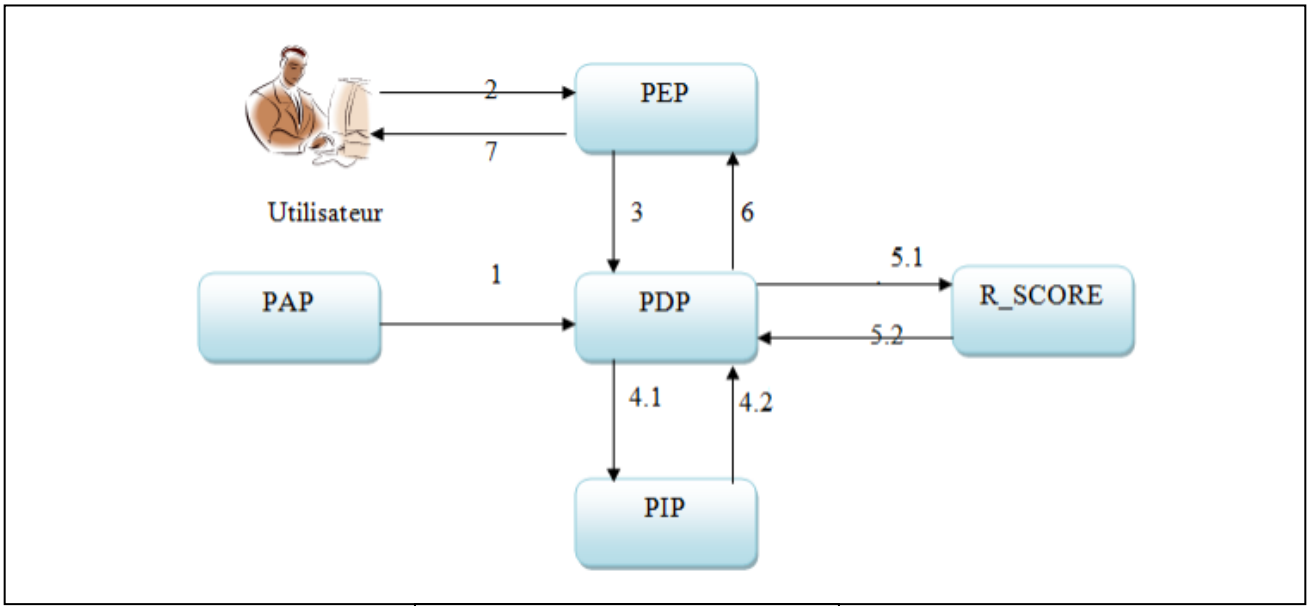


Figure 13 : Architecture général

- *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

1. L'administrateur définit les politiques ou l'ensemble de politiques et les rend disponibles au PDP

2. L'utilisateur envoie une requête d'accès au PEP.

3. Le PEP envoie la requête au PDP

4.1 Le PDP demande à son tour des attributs au PIP

4.2 Le PIP envoie les attributs nécessaires pour le PDP

5.1 Le PDP demande aussi a l'entité R_SCORE de calcule le risque de cette utilisateur et le comparer au seuil.

5.2 Le R_SCORE envoie un résultat de comparaison

6. Le PDP retourne la décision en s'appuient sur les résultats de R_SCORE et la politique d'accès et l'envoie au PEP

7. PEP remplit les obligations et, en se fondant sur la décision d'autorisation adressé par PDP, soit permet ou interdit l'accès

2.2. Diagramme de cas d'utilisations.

2.3. Modélisation UML :

Le langage UML (Unified Modeling Language, ou langage de modélisation unifié) a été pensé pour être un langage de modélisation visuelle commun, et riche sémantiquement et syntaxiquement. Il est destiné à l'architecture, la conception et la mise en œuvre de systèmes logiciels complexes par leur structure aussi bien que leur comportement.

L'UML a des applications qui vont au-delà du développement logiciel, notamment pour les flux de processus dans l'industrie. Il ressemble aux plans utilisés dans d'autres domaines et se compose de différents types de diagrammes. Dans l'ensemble, les diagrammes UML décrivent la limite, la structure et le comportement du système et des objets qui s'y trouvent.

L'UML n'est pas un langage de programmation, mais il existe des outils qui peuvent être utilisés pour générer du code en plusieurs langages à partir de diagrammes UML. L'UML a une relation directe avec l'analyse et la conception orientées objet. [22]

2.3.1. Diagramme de cas d'utilisations :

Les diagrammes de cas d'utilisation (DCU) sont des diagrammes UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour développement, les cas d'utilisation sont plus appropriés. En effet, un cas d'utilisation (use cases) représente une unité discrète d'interaction entre un utilisateur (humain ou machine) et un système. Ainsi, dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs (actors), et ils apparaissent dans les cas d'utilisation [24]

Dans le diagramme de cas d'utilisation on si n'importe quel utilisateur qui tente d'accéder va passer par la vérification de la politique, ensuite la politique va demander son Score de risque à l'application qui va lui remettre pour décider l'accès ou pas.

En cas de délégation si le docteur est absent alors il va déléguer ses droites à une infirmer ou même un autre docteur qui va passer par le même procéder que la tentative d'accès classique.

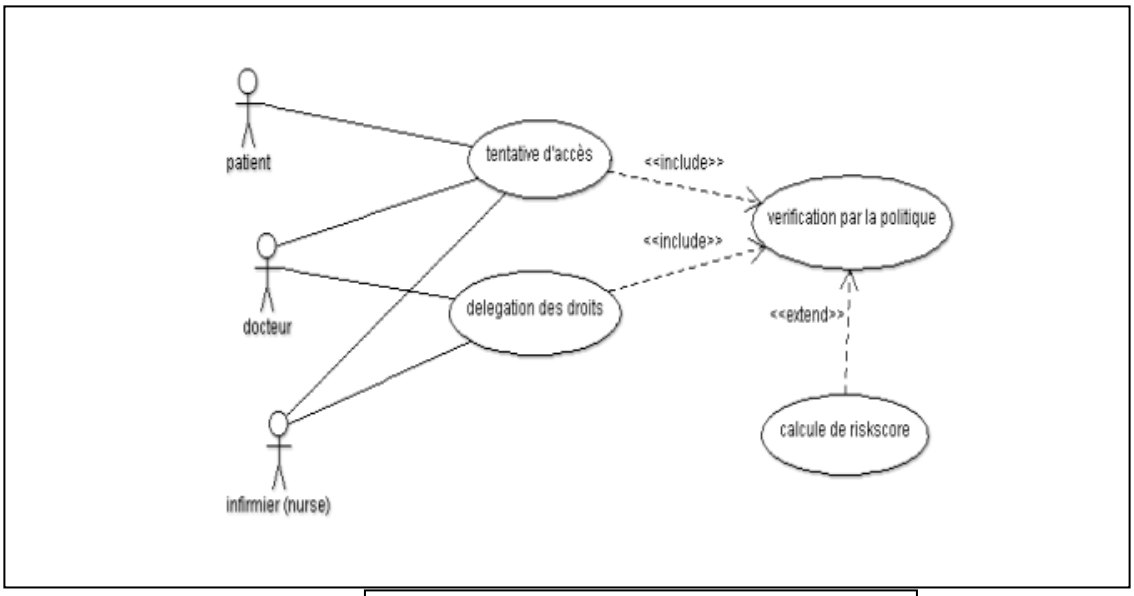


Figure 14 : Diagramme de cas d'utilisation

2.3.2 Diagramme de déploiement :

En UML, un diagramme de déploiement est une vue statique qui sert à représenter l'utilisation de l'infrastructure physique par le système et la manière dont les composants du système sont répartis ainsi que leurs relations entre eux. Les éléments utilisés par un diagramme de déploiement sont principalement les nœuds, les composants, les associations et les artefacts. Les caractéristiques des ressources matérielles physiques et des supports de communication peuvent être précisées par stéréotype.

Un diagramme de déploiement modèle l'architecture de temps d'exécution d'un système. Il affiche la configuration des éléments matériels (nœuds) et affiche comment des éléments logiciels et des artefacts sont mappés sur ces nœuds. [26]

Un diagramme de déploiement modèle l'architecture de temps d'exécution d'un système. Il affiche la configuration des éléments matériels (nœuds) et affiche comment des éléments logiciels et des artefacts sont mappés sur ces nœuds.

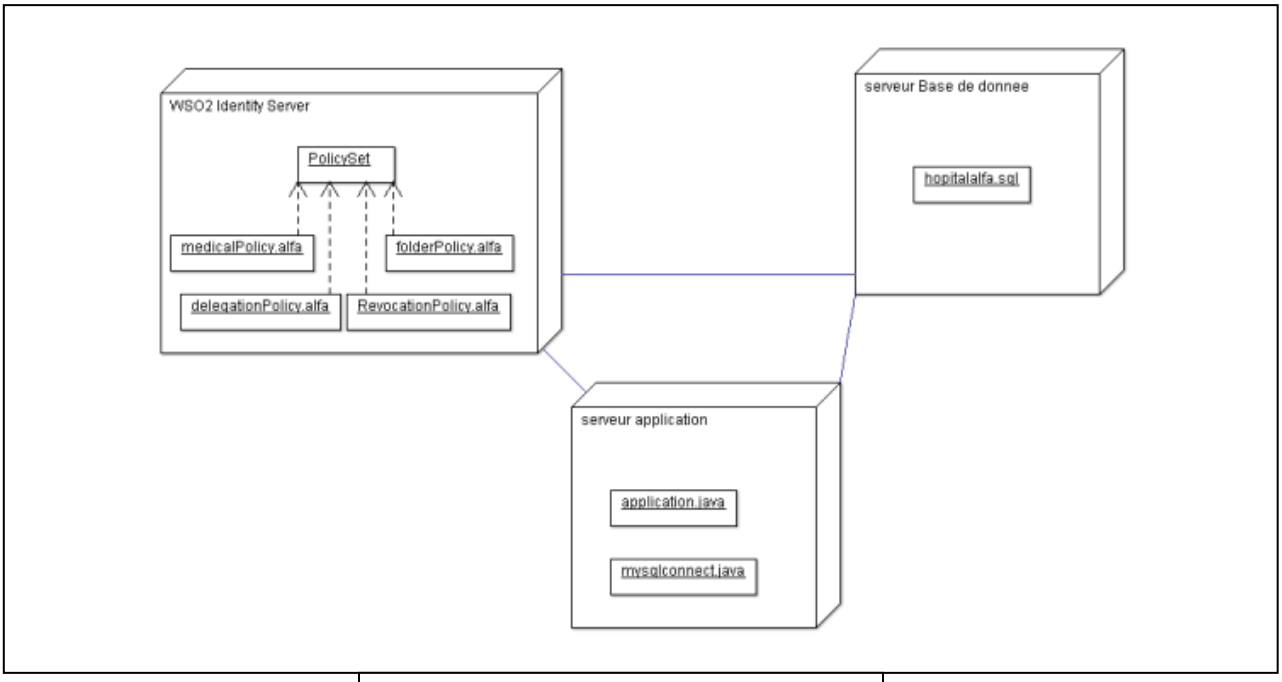


Figure 15 : Diagramme de déploiement

2.3.3. Diagramme de séquence :

Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario d'un Diagramme des cas d'utilisation. Dans un souci de simplification, on représente l'acteur principal à gauche du diagramme, et les acteurs secondaires éventuels à droite du système. Le but étant de décrire comment se déroulent les actions entre les acteurs ou objets.

La dimension verticale du diagramme représente le temps, permettant de visualiser l'enchaînement des actions dans le temps, et de spécifier la naissance et la mort d'objets. Les périodes d'activité des objets sont symbolisées par des rectangles, et ces objets dialoguent à l'aide de messages. [25]

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 tous le droits ex ici (lire, écrire…etc.)

CareR_Excite () : teste si une relation entre le demandeur d'accès et le dossier médicale existe.

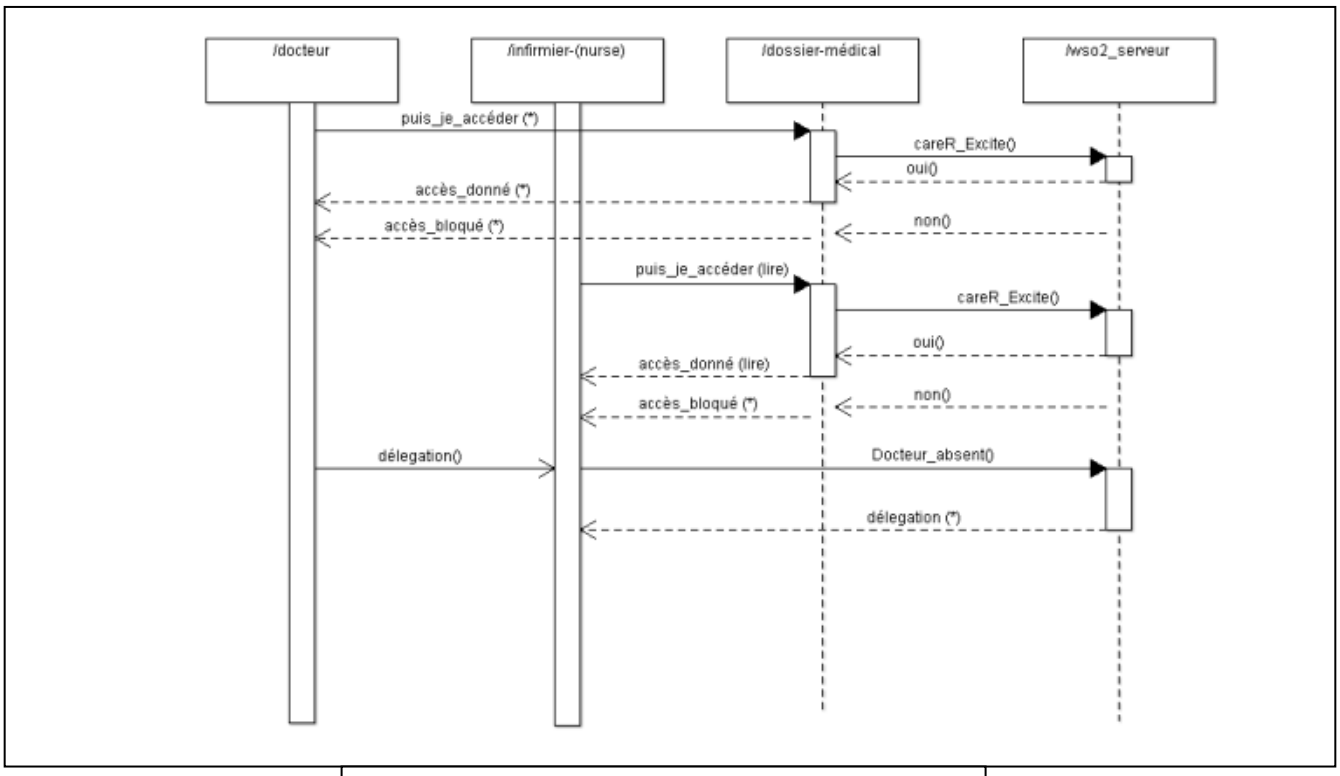


Figure 16 : Diagramme de séquence du processus

Diagramme de cas d'utilisation (wso2)

RS: Risk Score

TS: Trust Score

(*) : Tous les droits

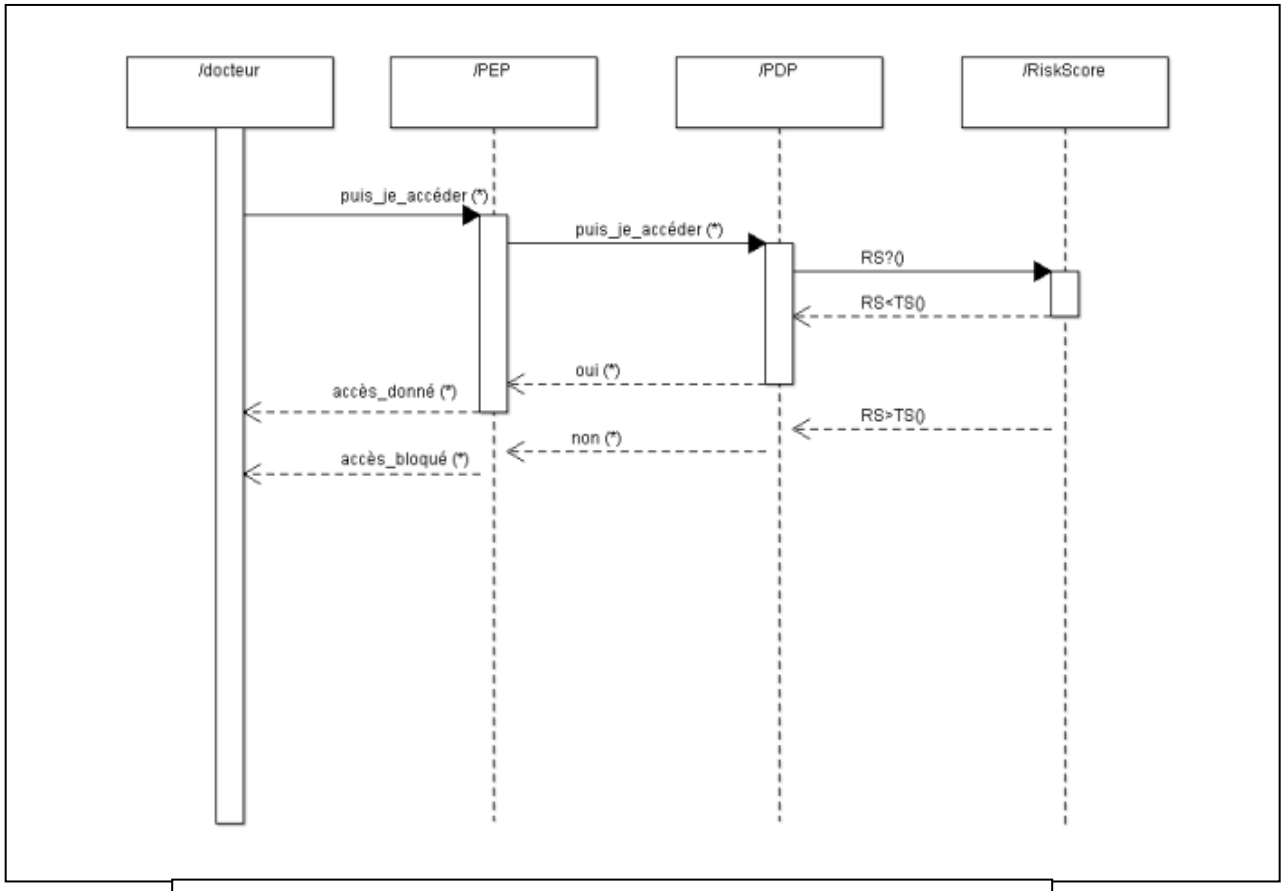


Figure 17 : Diagramme de séquence de test de la politique avec WSO

2.4. Outils de développement :

2.4.1. 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.

La norme définit un grand nombre d'éléments XML, est très détaillée et a un pouvoir expressif élevé, ce qui crée un seuil d'utilisation élevé pour les utilisateurs qui ne sont pas familiers avec elle ou d'autres langages basés sur XML.

La complexité de l'écriture et de la correction des politiques XACML peut être certaines des raisons pour lesquelles des normes d'autorisation plus simples et moins expressives (par exemple, OAuth, RBAC ou des listes de contrôle d'accès simples) peuvent être préférées dans les implémentations pratiques, ou même que les utilisateurs décident de déployer leur propre solution d'autorisation, avec la sécurité possible les risques que cela peut engendrer.

Lors de la création d'une politique basée sur XACML, le créateur doit avoir des connaissances à la fois de la normalisation de XACML et du comportement XML général.

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.[27]

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. [28]

2.4.2. JAVA :

Java est un langage de programmation orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 au Sun World.

La société Sun a été ensuite rachetée en 2009 par la société Oracle qui détient et maintient désormais Java.

Une particularité de Java est que les logiciels écrits dans ce langage sont compilés vers une représentation binaire intermédiaire qui peut être exécutée dans une machine virtuelle Java (JVM) en faisant abstraction du système d'exploitation. [29]

Beaucoup d'applications et de sites Web ne fonctionne 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 ![30]

Le langage Java reprend en grande partie la syntaxe du langage C++. Néanmoins, Java a été épurée des concepts les plus subtils du C++ et à la fois les plus déroutants, tels que les pointeurs et références, ou l'héritage multiple contourné par l'implémentation des interfaces. De même, depuis la version 8, l'arrivée des interfaces fonctionnelles introduit l'héritage multiple (sans la gestion des attributs) avec ses avantages et inconvénients tels que l'héritage en diamant. Les concepteurs ont privilégié l'approche orientée objet de sorte qu'en Java, tout est objet à l'exception des types primitifs (nombres entiers, nombres à virgule flottante, etc.) qui ont cependant leurs variantes qui héritent de l'objet Object (Integer, Float...).

Java permet de développer des applications client-serveur. Côté client, les applets sont à l'origine de la notoriété du langage. C'est surtout côté serveur que Java s'est imposé dans le milieu de l'entreprise grâce aux servlets, le pendant serveur des applets, et plus récemment les JSP (JavaServer Pages) qui peuvent se substituer à PHP, ASP et ASP.NET.

Java a donné naissance à un système d'exploitation (JavaOS), à des environnements de développement (eclipse/JDK), des machines virtuelles (MSJVM (en), JRE) applicatives multiplate forme (JVM), une déclinaison pour les périphériques mobiles/embarqués (J2ME), une bibliothèque de conception d'interface graphique (AWT/Swing), des applications lourdes (Jude, Oracle SQL Worksheet, etc.), des technologies web (servlets, applets) et une déclinaison pour l'entreprise (J2EE). La portabilité du bytecode Java est assurée par la machine virtuelle Java, et éventuellement par des bibliothèques standard incluses dans un JRE. Cette machine virtuelle peut interpréter le bytecode ou le compiler à la volée en langage machine. La portabilité est dépendante de la qualité de portage des JVM sur chaque OS.[32]

2.4.3. XPath :

XPath est un langage permettant de sélectionner des parties d'un document XML. Il est utilisé dans de nombreux dialectes XML. Il est déjà apparu dans les contraintes de cohérence des schémas XML. Le langage XSLT fait également un usage intensif de XPath pour désigner les parties à traiter.

Le langage XPath n'est pas un langage autonome. C'est un langage d'expressions utilisé au sein d'un autre langage hôte. Il ressemble, dans cet aspect, aux expressions rationnelles, appelées aussi expressions régulières qui est abrégé en regex telles qu'elles sont utilisées dans les langages de script tels que Perl ou Python.

La syntaxe de XPath n'est pas une syntaxe XML car les expressions XPath apparaissent en général comme valeurs d'attributs de documents XML. C'est en particulier le cas pour les schémas, les schmecterons et XSLT.

XPath était au départ un langage permettant essentiellement de décrire des ensembles de nœuds dans un document XML. La version 1.0 de XPath comprenait quelques fonctions pour la manipulation de nombres et de chaînes de caractères. L'objectif était alors de pouvoir comparer les contenus de nœuds. La version 2.0 de XPath a considérablement enrichi le langage. Il est devenu un langage beaucoup plus complet capable, par exemple, de manipuler des listes de nœuds et de valeurs atomiques.

XPath est uniquement un langage d'expressions dont l'évaluation donne des valeurs sans effet de bord. Il n'est pas possible dans XPath de mémoriser un résultat. Il n'existe pas de variables propres à XPath mais une expression XPath peut référencer des variables du langage hôte. Les valeurs de ces variables sont alors utilisées pour évaluer l'expression. L'affectation de valeurs à ces variables se fait uniquement au niveau du langage hôte.

Le cœur de XPath est formé des expressions de chemins permettant de décrire des ensembles de nœuds d'un document XML. Ces expressions ressemblent aux chemins Unix pour nommer des fichiers dans une arborescence.[33]

2.4.4. AxiomaticsLanguage for Authorization (ALFA) :

AxiomaticsAuthorizationLanguage (ALFA) est un langage spécifique à un domaine pour la description de haut niveau des stratégies XACML. Il est conçu pour être facile à utiliser par les développeurs. De plus, il présente des informations spécifiques au domaine telles que les identifiants d'attribut sous une forme compressée et peut être compilé en XACML 3.0. [34]

Il y a un an, le 16 juillet 2012, Axiomatics (situé à Stockholm, en Suède) a publié le plug-in ALFA pour Eclipse lors du Cloud Identity Summit 2012, Vail, CO.

Axiomatics, est le principal fournisseur de solutions d'autorisation basées sur les attributs et basées sur la norme XACML. L'entreprise a une clientèle mondiale Couvrant les secteurs de la santé, du gouvernement et de la finance.

L'outil permet aux développeurs d'écrire des politiques XACML 3.0 sous une forme abstraite appelée ALFA (AxiomaticsLanguage For Authorization). Les développeurs peuvent écrire des politiques directement dans l'IDE Eclipse et le plugin traduit automatiquement d'ALFA en XACML. [35]

Dans ce projet, nous avons utilisé ALFA comme plugin dans Eclipse.

3. Implementation:

3.1. AxiomaticLanguage for Authorization (ALFA) :

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

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

Pour commencer il faut télécharger java JDK ainsi que l'environnement de développement java « Eclipse IDE ». Ensuite on télécharge le plugin ALFA pour Eclipse et on l'ajoute à l'environnement de manière classique, on crée un nouveau projet et dans ce dernier on va créer un fichier qu'on va nommer par exemple « rule.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 (figure 17) nous allons appuyer sur « Yes » .

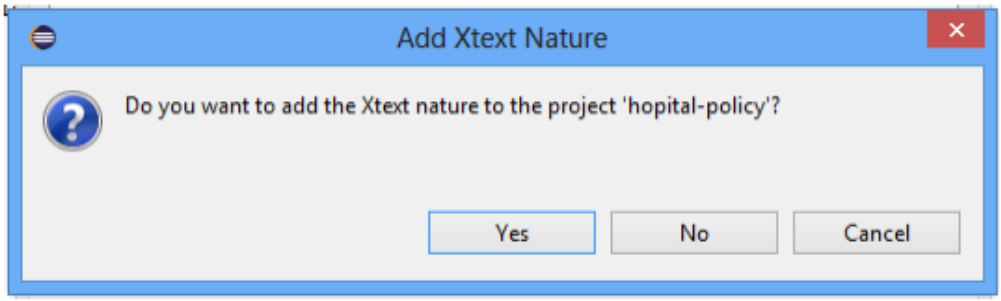


Figure 18 : demande d'ajout de la nature Xtext

Il faut ensuite copier le fichier "system. alfa" de la distribution ALFA dans le projet. Ce fichiercontient 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 4

politiques (voir dans l'Annexe leur code XACML généré):

3.2. Politique utilise :

1. La politique « médicalPolicy » : Une pour le dossier médical du patient « medicalPolicy » qui 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 « thereis no care relation » ; de même pour l'infirmière en cas de délégation* .

Si le dossier est bloqué pour n'importe quelle raison alors un message « the record isblocked » sera affiché.

Grâce à ALFA, un fichier XACML concernant cette politique est généré automatique, et mis par défaut dans le sous-dossier src-gen(voir l'annexe).

```
policy medicalPolicy {
    target clause resource.resourceType == "medical-record"
    apply firstApplicable
    rule doctoract{
    permit
    target clause user.role == "doctor"
    //and user.role == "nurse"
    condition (booleanOneAndOnly(resource.careRelationExists))
}
    rule notdoctor{
    deny
    condition not(booleanOneAndOnly(resource.careRelationExists))
    on deny {
    advice ObligationAdvice.reasonForDeny {
    action.message = "There is no care relation"
}
}
}
    rule blockedacces{
    deny
    condition booleanOneAndOnly(resource.recordIsBlocked)
    on deny {
    advice ObligationAdvice.reasonForDeny {
    action.message = "The record is blocked"
}
}
}

rule nurseact{//en cas ou le docteur et absent et la nurse est deleger
    permit
    target clause user.role == "nurse"

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

}
```

Figure 19 : code Alfa pour la politique medicalPolicy

Figure 20 : Code xacml génère pour la politique medicalPolicy.

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

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

    or user.role == "nurse"
    or user.role == "patient"
    clause action.actiontodo == "read"
    //clause user.role == "doctor" and action.actiontodo == "write"
    }

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

Figure 21. Code alfa pour La politique folderPolicy.

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

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

Figure 22 : Code xacml génère pour la politique folderPolicy.

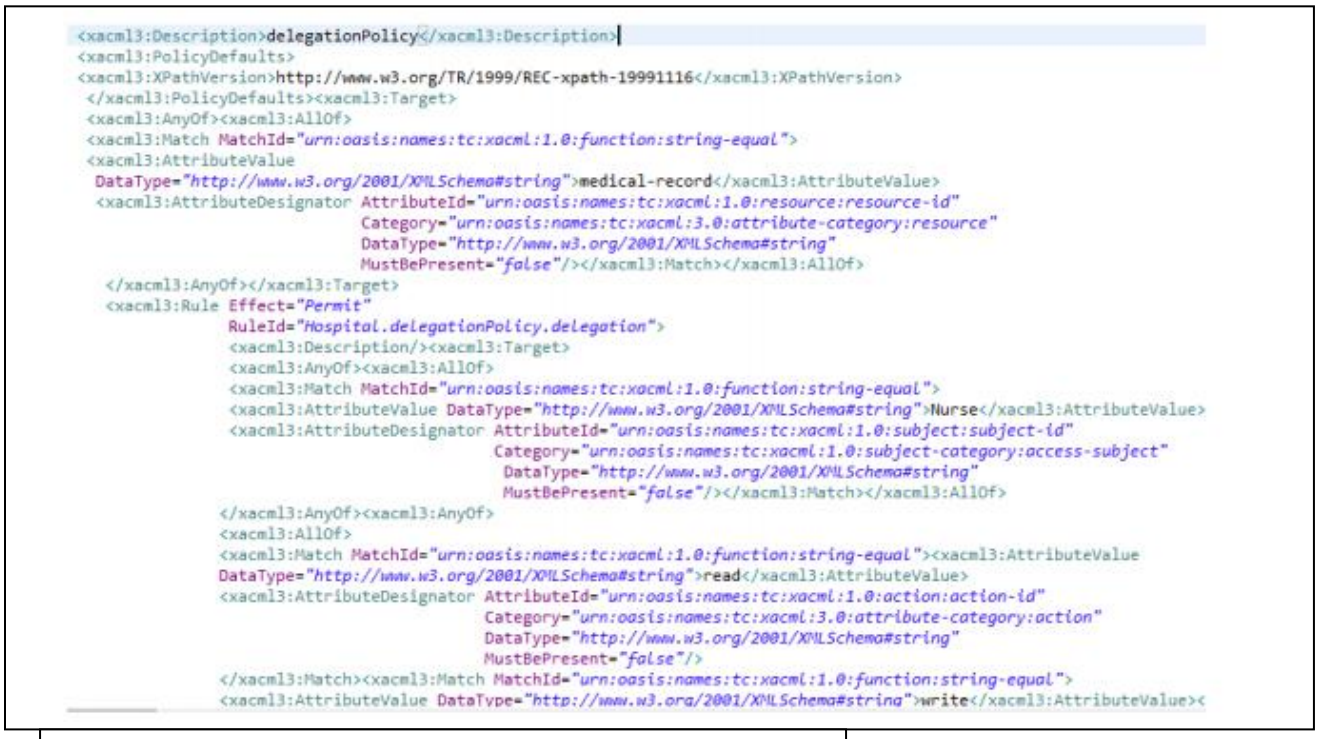
3. La politique « délégationPolicy » : La politique de délégation dans la « Politique de délégation » est une politique petite mais nécessaire et décisive car grâce à cette politique le travail ne s'arrêtera pas en l'absence de la personne responsable, car il y a toujours quelqu'un pour prendre sa place.

Dans notre exemple, cette politique s'appliquera à l'infirmière en l'absence du médecin en charge du patient, ce dernier devra être remplacé par l'infirmière appropriée, et il déléguera ici ses droits de lecture et d'écriture dans son dossier patient.

```
policy delegationPolicy {
    target clause user.role == "nurse"
    apply permitOverrides
rule delegation {
    permit
    condition booleanOneAndOnly(resource.doctorIsAbsent)
    //and condition (booleanOneAndOnly(resource.careRelationExists))
target
    clause  user.role == "nurse"
    clause  action.actiontodo == "read" and action.actiontodo == "write"
    }

}
```

Figure 23. Code alfa La politique délégationPolicy.



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

Figure 24. Code xacml génère pour la politique délégationPolicy.

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

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

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

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

Figure 25 : Code Alfa pour la politique WorkTimeAccess.

Figure 26. Code xacml génère pour la politique

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

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

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

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

target  clause
    user.role == "nurse"
    or user.role =="doctor"//justone
    clause action.actiontodo == "read"
     and action.actiontodo == "write"

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

Figure 27. Code Alfa pour la politique revocationPolicy.

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

Figure 28. Code xacml génère pour la politique revocationPolicy.

6. séparation des tâches :

La politique de séparation des tâches est une politique modeste mais nécessaire car elle repose sur le principe que certaines tâches ne peuvent pas être exécutées simultanément par la même personne.

Dans notre projet ici, les tâches ici sont la lecture et l'écriture, car il n'est pas possible que celui qui rédige et évalue le dossier médical soit la même personne.

```
/*************************************Separation des tache************************/
policy taches_separation {
    target clause resource.resourceType == "medical-record"
     apply permitOverrides
        rule permit_acces{
        permit
        condition booleanOneAndOnly(resource.Modify_file)
         target clause user.role == "Doctor"
         clause action.actiontodo == "write"
       }
      rule denny_acces{
        deny
        condition booleanOneAndOnly(resource.Modify_file)
         target clause user.role == "Nurse"
         or user.role == "Patient"
         clause action.actiontodo == "read"
         clause action.actiontodo == "write"
         on deny {
advice ObligationAdvice { action.message = "you can't acces now, file is update now" }
}
}
}
/*****************************************************************************/
```

Figure 29. Code alfa pour la politique de séparation des taches

3.3. Le PEP

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

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

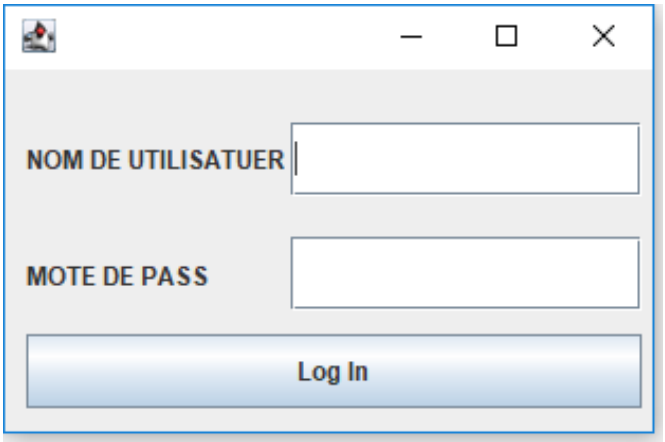


Figure 30. Authentification d'utilisateur

Après avoir vérifié le nom d'utilisateur et le mot de passe, l'utilisateur est autorisé à mettre en œuvre la deuxième interface qui permet aux utilisateurs d'enregistrer les infirmières, sur la base des métriques, l'application calculera la moyenne (Fig. 28) et la stockera dans un base de données qui est ensuite utilisée par la politique (par exemple, politique d'annulation) pour déterminer la décision d'accès.

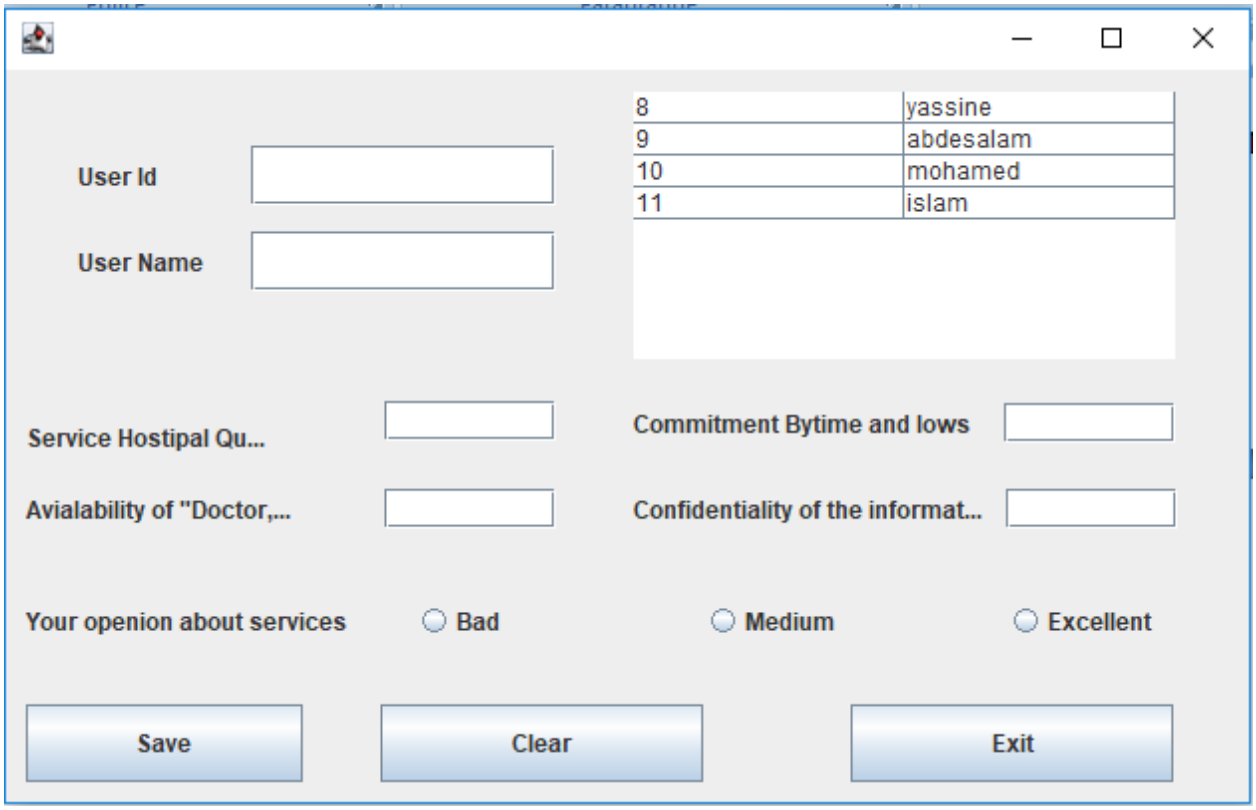


Figure 31 : interface PEP

```
JButton btnNewButton = new JButton("Save");
btnNewButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
    float rs ;
    java.sql.PreparedStatement  stmt;
    rs=((Float.parseFloat(tnote.getText())+
            Float.parseFloat(tnotei.getText())+
            Float.parseFloat(tnotec.getText())+
            Float.parseFloat(tnotea.getText()))/4);
    String sqlstatement ="insert into risquescore values (? , ? , ? , ? , ? , ? , ? , ? , ? )";
    try {
        Class.forName("com.mysql.jdbc.Driver");
        Connection con=DriverManager.getConnection(
        "jdbc:mysql://localhost/hopitalfin","root","");
        stmt=con.prepareStatement(sqlstatement);
        stmt.setString(8, textField_4.getText());
        stmt.setString(1, textField_5.getText());
        stmt.setString(2, tnote.getText());
        stmt.setString(3, tnotei.getText());
        stmt.setString(4, tnotec.getText());
        stmt.setString(5, tnotea.getText());
        stmt.setString(6, expression);
        stmt.setFloat(7, rs);
        stmt.execute();
        JOptionPane.showMessageDialog(null, "Saved Succsesfully");
    }
    catch(Exception e1)
    { System.out.println(e1);}
    }
});
```


Figure 32 : code source montrant la formule de calcul

Dans cette partie nous présenterons le WSO 2 IS Identity Server, ce serveur nous permet de tester les politiques créées avec ALFA.

Nous aurions pu utiliser un autre serveur plus pratique et rationalisé que WSO2IS, mais tous les serveurs, à l'exception de WSO 2 IS Identity Server, sont des serveurs payants et ne sont pas accessibles.

Pour commencer, nous démarrons WSO2 Identity Server "wso2server.bat" dans le répertoire /bin, vous devez vous connecter à la console d'administration avec le nom d'utilisateur Admin et le mot de passe admin.

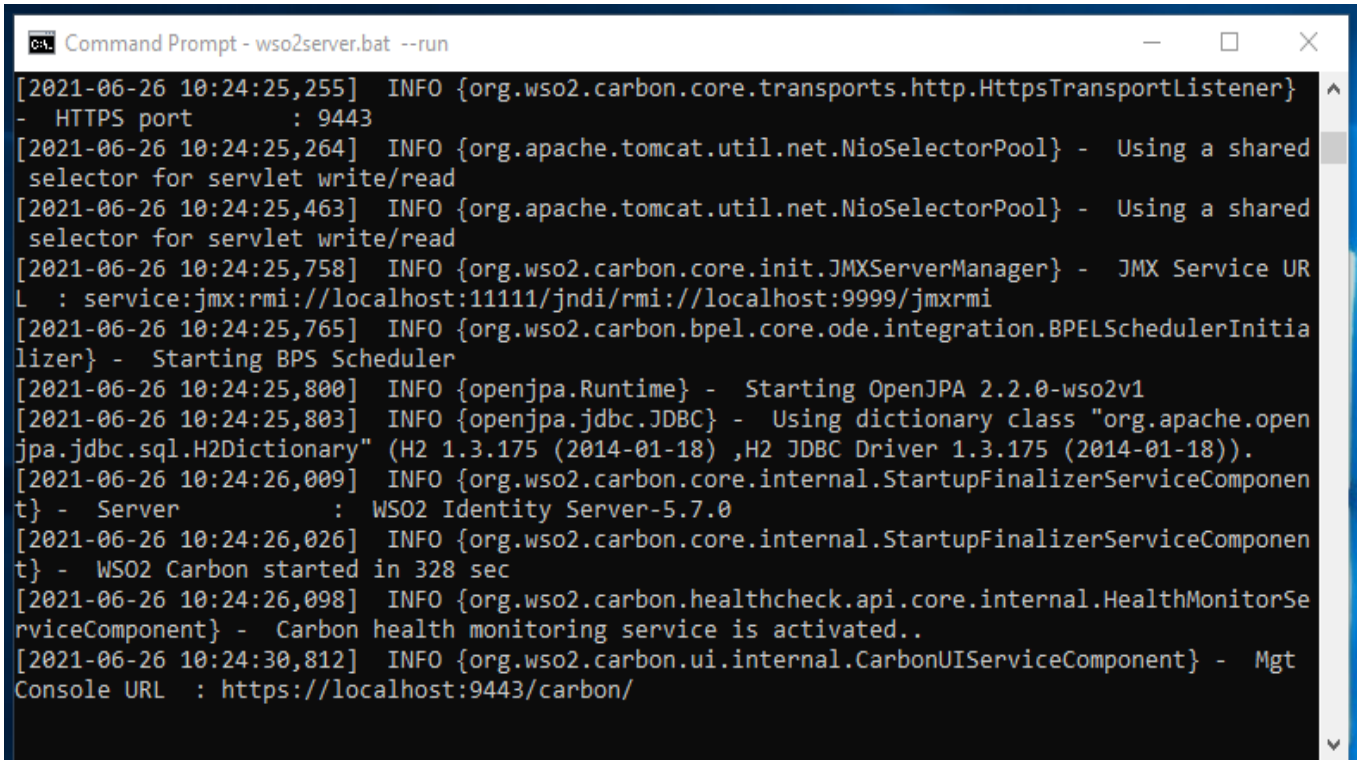


Figure 33 : invite de commande pour lancer WSO2

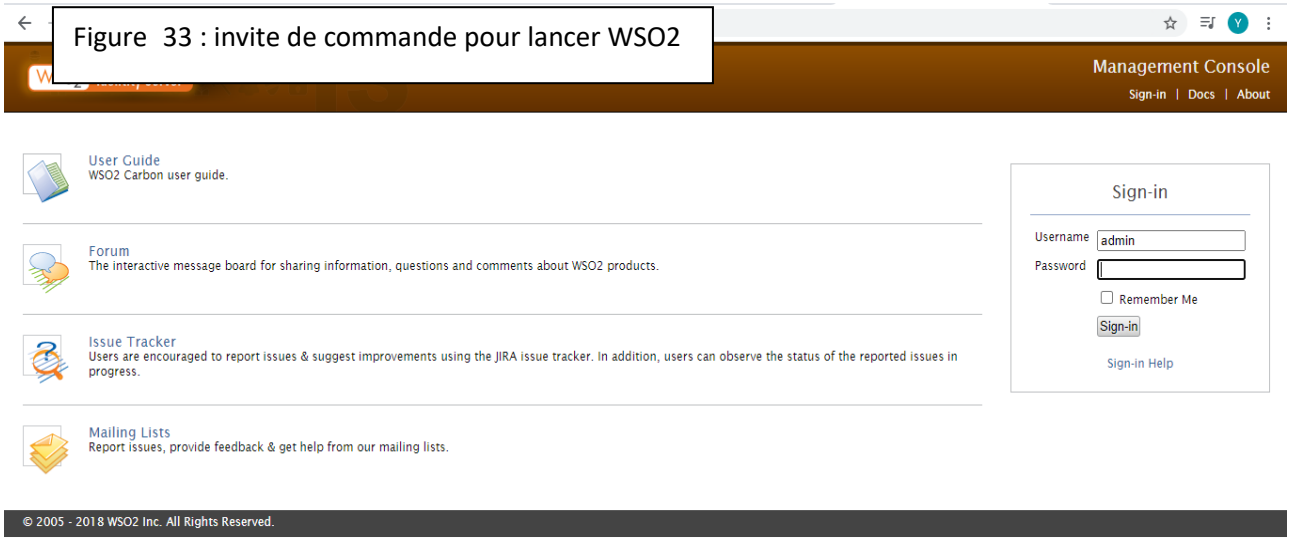


Figure 34 : page d'accueil WSO2

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

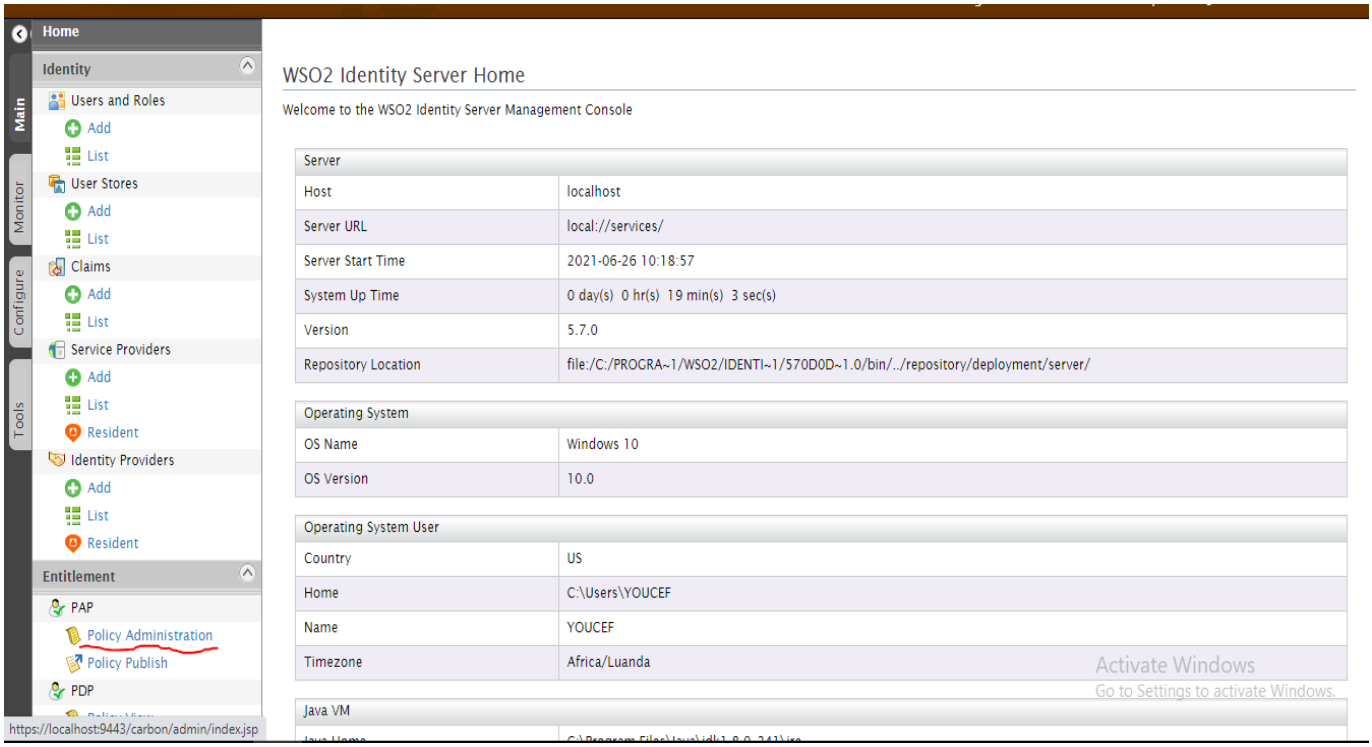


Figure 35 : La page principale de PAP.

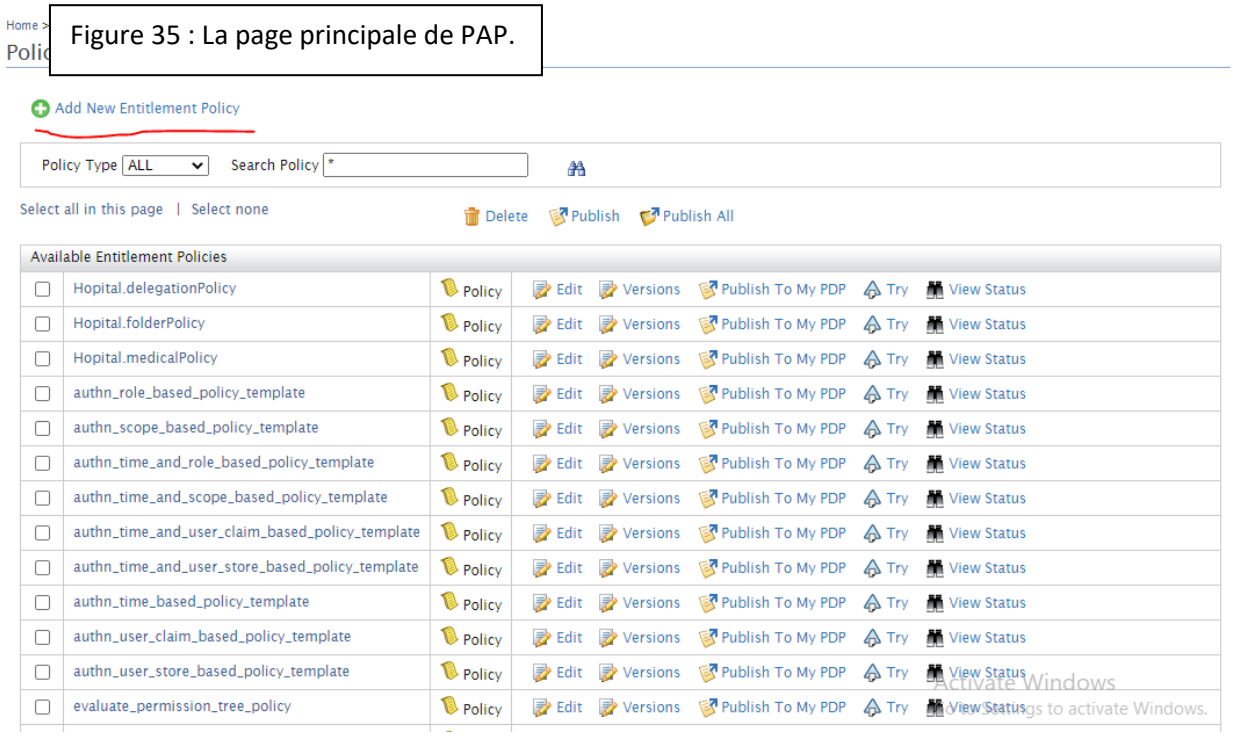


Figure 36 : page de PAP

On Clique alors sur Importer une politique existante ou écrire une politique en XML « write Policy in XML » pour ajouter la politique.

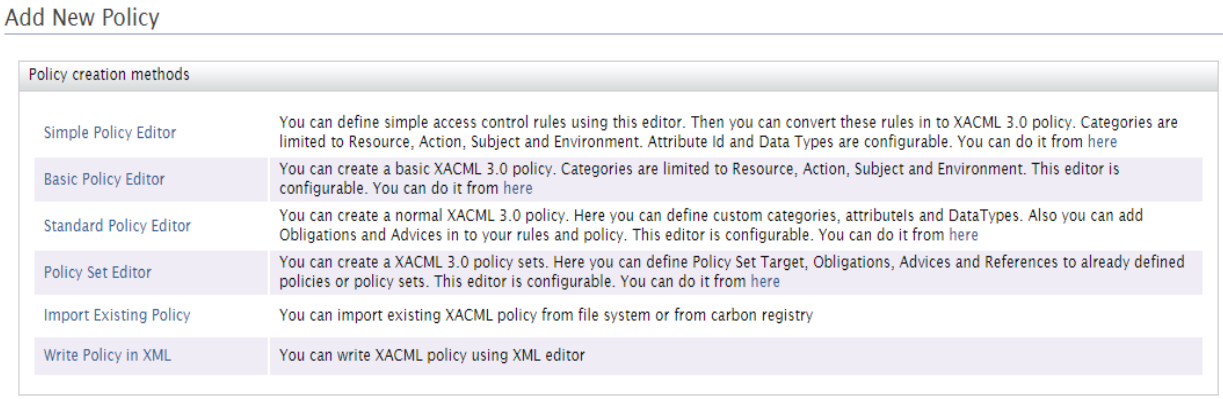


Figure 37 : interface de pour l'ajoute d'une nouvelle politique

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

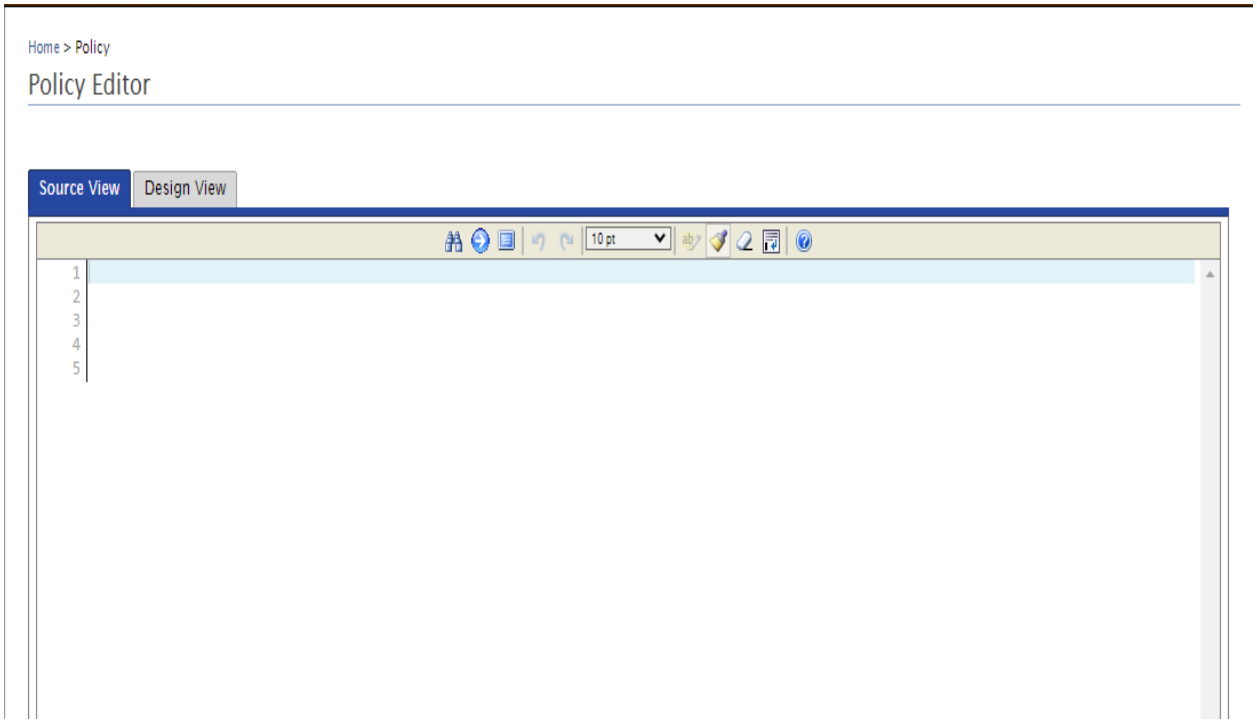


Figure 38 : Éditeur de politique XML.

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

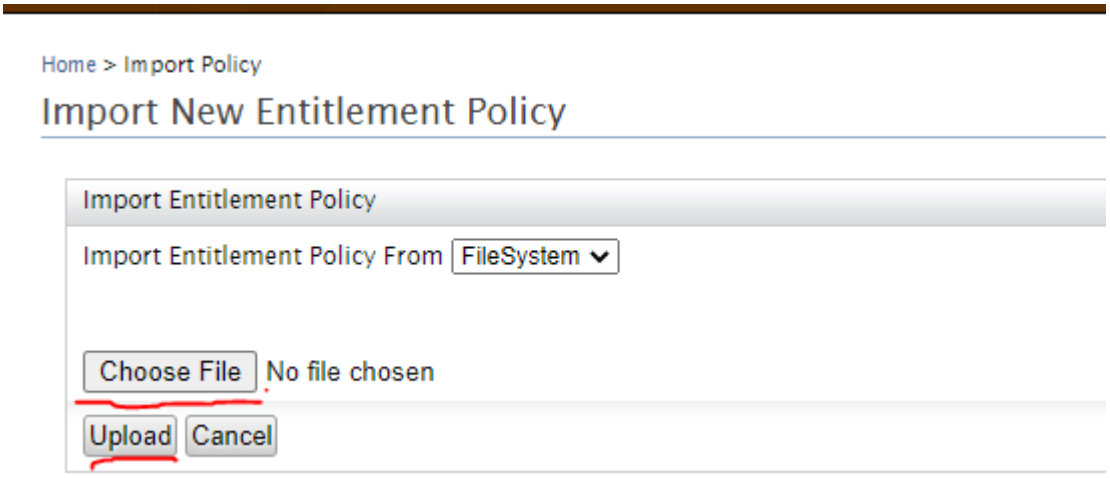


Figure 39 : interface d'importation de la politique

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

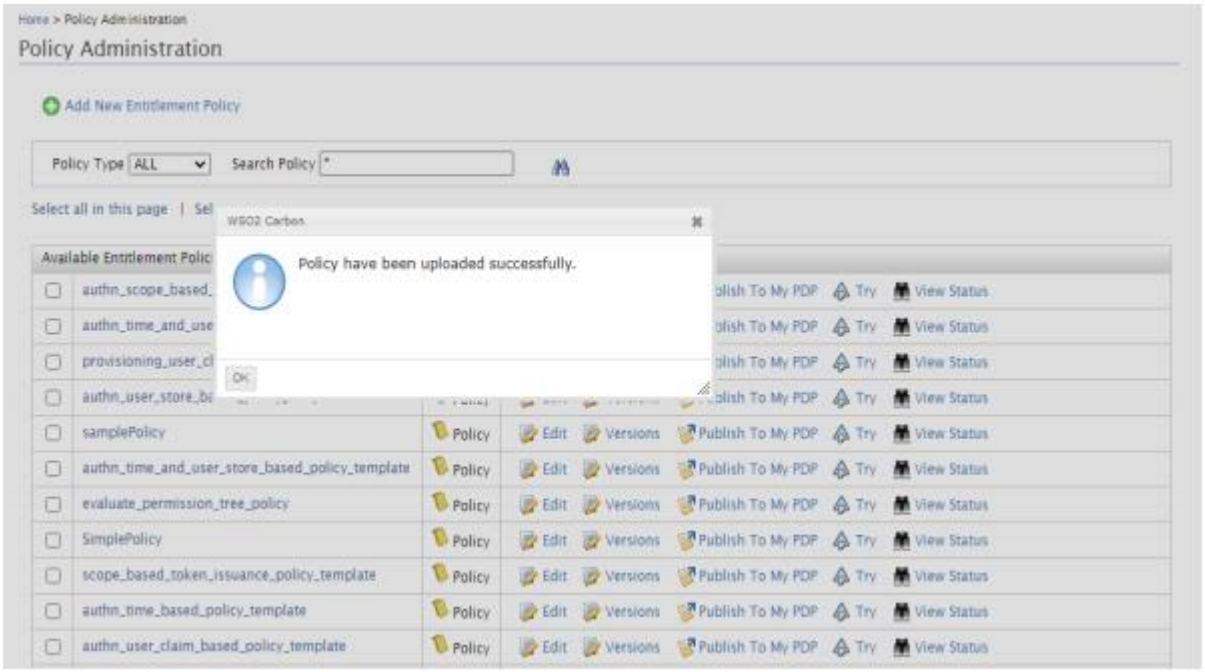


Figure 40 : Message de réussir dans l'ajout.

Apres avoir ajouté notre politique nous allons la tester en cliquant sur « try »

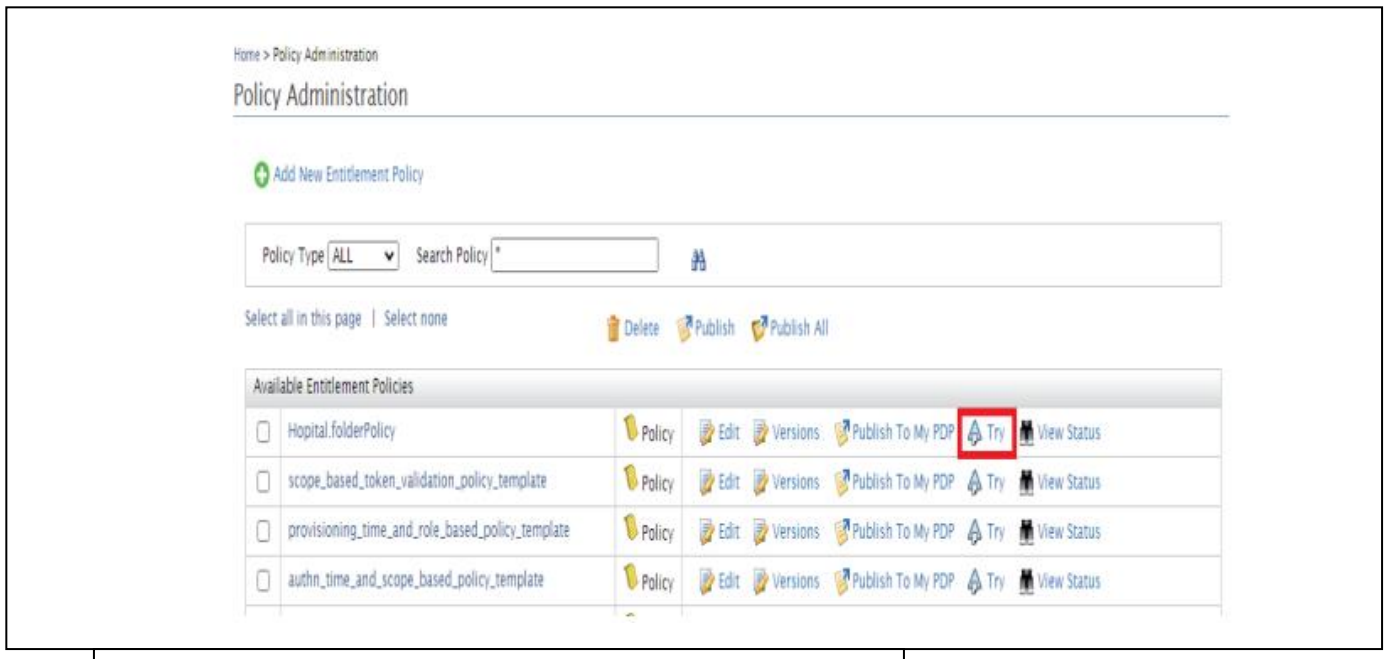


Figure 41 : figure montrant notre politique importe avec succès

Ensuite on remplit les attributs de la requête dans le formulaire « TryIt ».

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

Et la réponse attendue sera « permit »

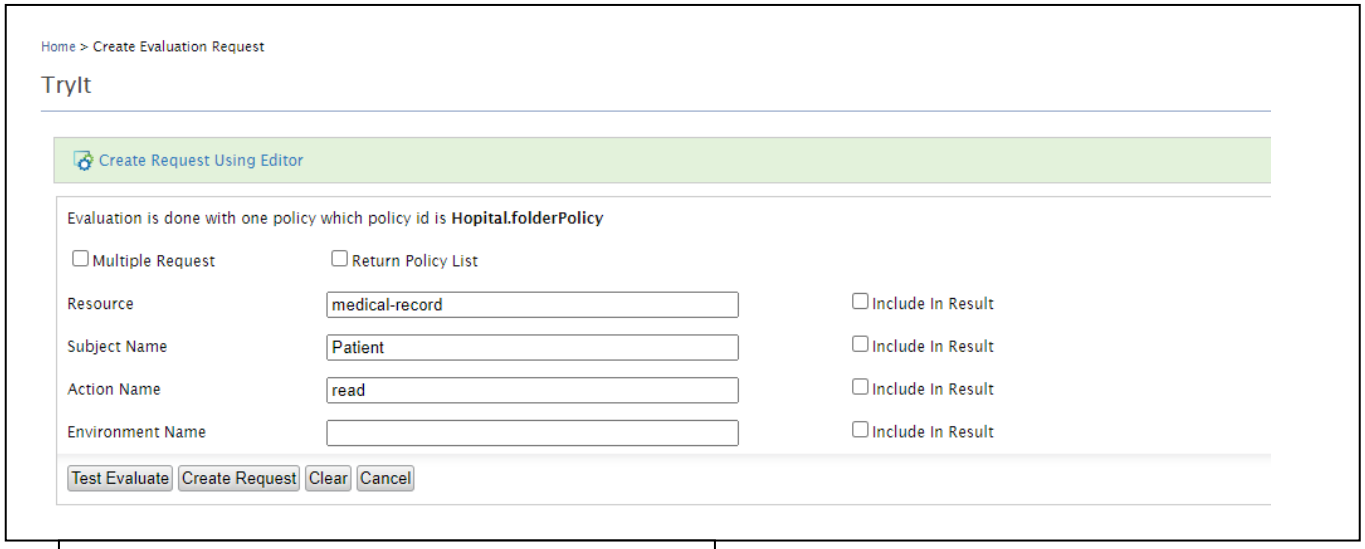


Figure 42 : fenêtre de l'éditeur de requête "TryIT"

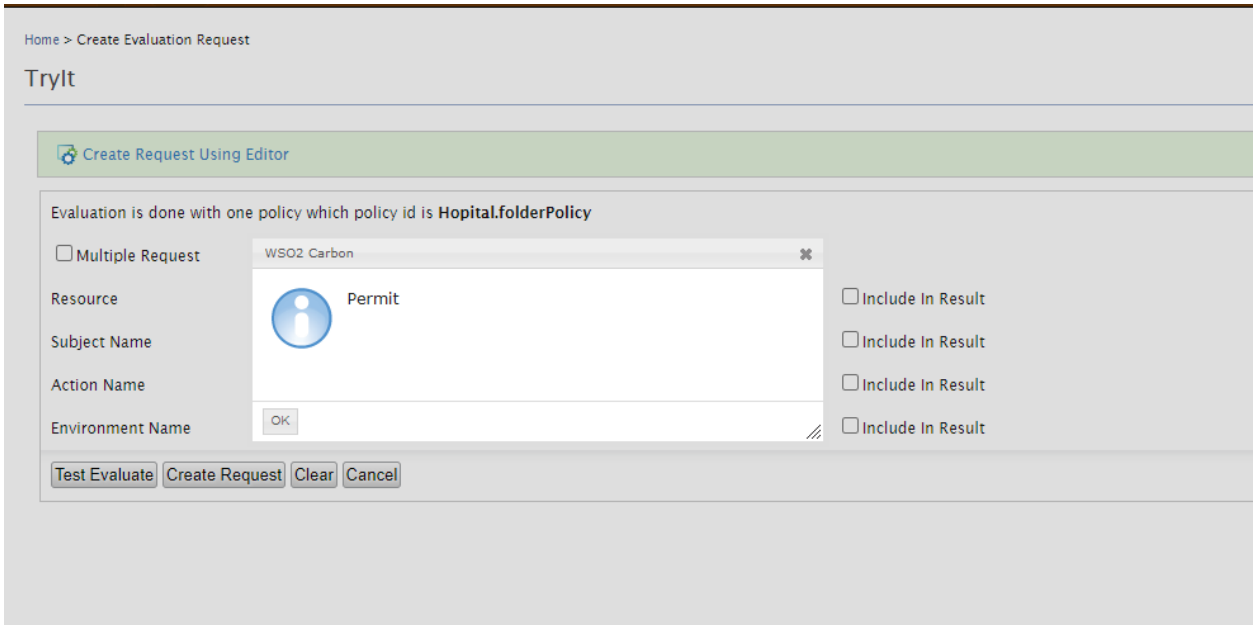


Figure 43 : résultat de la requête

Dans le cas écrire (l'action est write) ; la réponse attendue sera « deny »

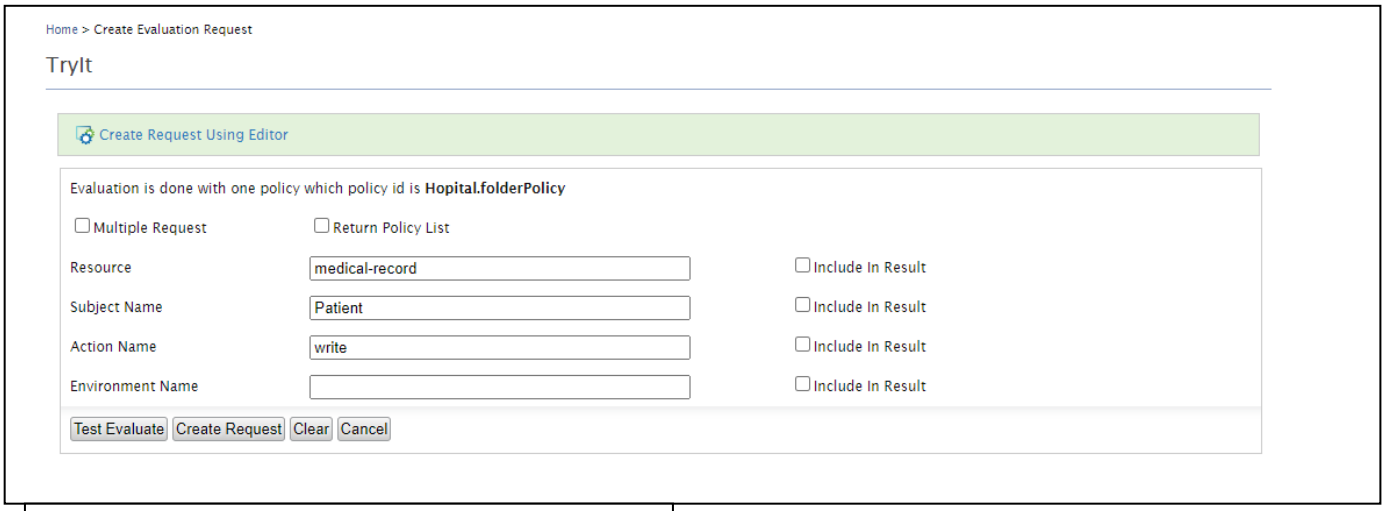


Figure 44 : fenêtre de l'éditeur de requête "TryIT"

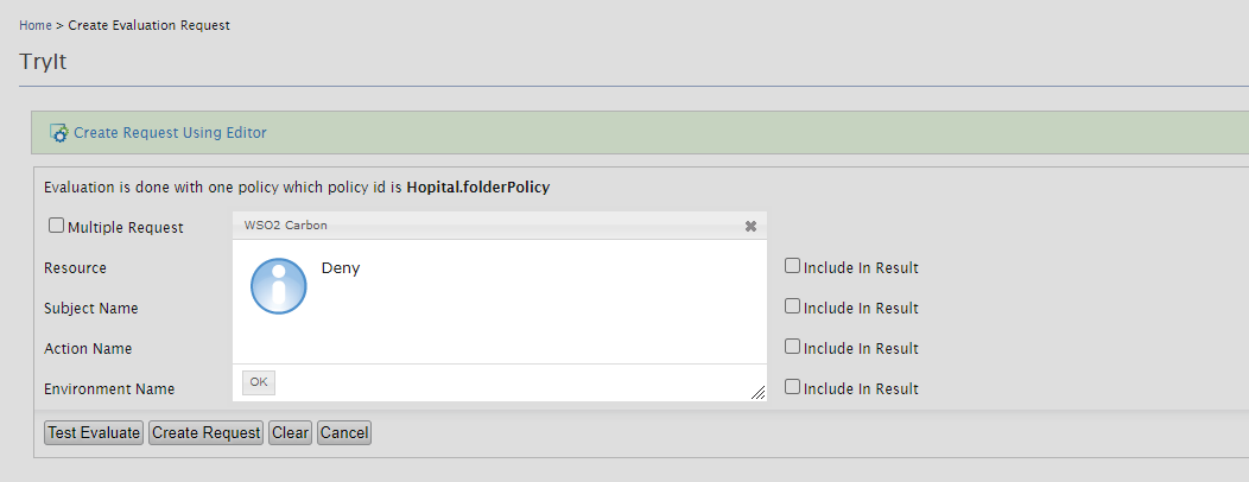


Figure 45 : résultat de la requête

3.5. Conclusion :

Dans ce chapitre nous avons présenté le fonctionnement de notre architecture, 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 restent un point de débat et de recherche et restera pour au moins la décennie qui vient.

Dans ce projet nous avons proposé l'un des solutions convenable pour gérer certain problème majeurs que se pose dans le domaine des politique de contrôle d'accès, Nous avons également mis en évidence les lacunes des cadres et des modèles existants, nous avons proposé une architecture de déroulement standard mais avec un plus, on a ajouté un mécanisme de calcule de score de risque incorporer qui put calculer le risque à partir de métriques donnes, et en se basent sur ce résultat la politique peut déduire l'accès ,délégation ,révocation ou pas d'un utilisateur.

Nous préconisons pour un travail futur de concevoir un calcule automatique de risque score ainsi qu'un contrôle d'accès et délégation auto-adaptatif c'est-à-dire un système autonome.

References:

[1] :http://www.nist.gov/itl/cloud/index.cfm) est habituellement utilisée par la plupart des acteurs.

[2]:https://www.techniques-ingenieur.fr/base-documentaire/technologies-de-l-information-th9/management-des-systemes-d-information-42302210/cloud-computing-h6020/

[3] :https://azure.microsoft.com/fr-fr/overview/what-is-cloud-computing/

[4] :https://www.intel.fr/content/www/fr/fr/cloud-computing/what-is-hybrid-cloud.html

[5] :https://blog.advancia-itsystem.com/applications-cloud-computing/#.YLqaxfn0nIU

[6]:https://www.ibm.com/fr-fr/cloud/learn/benefits-of-cloud-computing

[7] :https://www.scalair.fr/blog/cloud-public

[8] :https://fr.wikipedia.org/wiki/Contr%C3%B4le_d%27acc%C3%A8s

[9] :https://www.nedapsecurity.com/fr/insight/quest-ce-que-le-controle-dacces-et-pourquoi-est-ce-vital/

[10] :http://www.gel-securite.com/produits-controle-acces.php

[11]:https://fr.wikipedia.org/wiki/Contr%C3%B4le_d%27acc%C3%A8s_discr%C3%A9tionnaire

[12] :https://www.ionos.fr/digitalguide/serveur/securite/quest-ce-que-le-mandatory-access-control-mac/

[13] :https://www.researchgate.net/publication/2745230_Mutual_Exclusion_of_Roles_as_a_Means_of_Implementing_Separation_of_Duty_in_Role-Based_Access_Control_Systems

[14] :https://slideplayer.fr/slide/1851210/

[15] :R. Sandhu. The nextgeneration of access control models: Do weneedthem and whatshouldtheybe? In SACMAT'01, page 53. SACMAT, May 2001.

[16] :D. Ferraiolo and R. Kuhn. Role-basedaccesscontrols. In Proc. of the 15th NIST-NCSC Naional Computer Security Conference, pages 554–563, Baltimore, MD, October 1992.

[17]: YAO Zhilin, LI Bing and LIU Shufen, "RoleBased Collaboration Authorizing by UsingOntology", Chinese Journal of Electronics Vol.20, No.3, July 2011.

[18]:Prof. S.A.Ubale and Dr. S.S. Apte, "Study and Implementation of Code Access Security with .Net Framework for Windows Operating System", International Journal of Computer Engineering &Technology (IJCET), Volume 3, Issue 3, 2012, pp. 426 - 434, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.

[19]:Torsten Priebe, Wolfgang Dobmeier, Christian Schläger, Nora Kamprath, "SupportingAttribute-based Access Control in Authorization and Authentication Infrastructures with Ontologies", First International Conference on Availability, Reliability and Security (ARES 2006), Vienna, Austria, April 2006.

[20]:Bokefode J.D, Ubale S. A, Modani D. G, Bhandare P.S. "Enhancing the web site structure to provideeasytraversal on a websitewith minimum changes to its structure ", International Journal of Computer

[21] :Marwan CHEAITO, l'Université Toulouse III - Paul Sabatier, Un cadre de spécification et de déploiment de politiques d'autorisation, 2012

[22] https://www.lucidchart.com/pages/fr/langage-uml

[23] Bernard Stepien, Stan Matwin, Amy Felty, "Advantages of a Non-Technical XACML Notation in Role-BasedModels", 2011 NinthAnnual International Conference on Privacy, Security and Trust.

[24] https://fr.wikipedia.org/wiki/Diagramme_de_cas_d%27utilisation

[25] https://fr.wikipedia.org/wiki/Diagramme_de_s%C3%A9quence

[26]https://www.sparxsystems.fr/resources/uml2_tutorial/uml2_deploymentdiagram.html

[27] https://www.researchgate.net/publication/273333571_A_scratch-based_graphical_policy_editor_for_XACML

[28] Romain Laborde, Thierry Desprats, « Gestion de conditions stables dans XACML

[29] https://fr.wikipedia.org/wiki/Java_(langage)

[31] https://www.java.com/fr/download/help/whatis_java.html

[32]https://fr.wikipedia.org/wiki/Java_(langage)#Aper%C3%A7u

[33] https://www.irif.fr/~carton/Enseignement/XML/Cours/XPath/index.html

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

[35]https://www.webfarmr.eu/axiomatics-language-for-authorization/

[36] 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.

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

[38] YAO Zhilin, LI Bing and LIU Shufen, "RoleBased Collaboration Authorizing by UsingOntology", Chinese Journal of Electronics Vol.20, No.3, July 2011.

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

[40] Prof. S.A.Ubale and Dr. S.S. Apte, "Study and Implementation of Code Access Security with .Net Framework for Windows Operating System", International Journal of Computer Engineering &Technology (IJCET), Volume 3, Issue 3, 2012, pp. 426 - 434, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.

[41]https://www.researchgate.net/publication/2745230_Mutual_Exclusion_of_Roles_as_a_Means_of_Implementing_Separation_of_Duty_in_Role-Based_Access_Control_Systems.