République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab Blida1

Faculté des sciences

Département d'informatique

Mémoire Présenté par :

Bouanani Walid

Soufi Mehdi

En vue d'obtenir le diplôme de master

Domaine : Mathématique et informatique
Filière : Informatique
Spécialité : Informatique
Option : Ingénierie de logiciel

Sujet :

Implémentation d'un modèle de contrôle d'accès dynamique pour un réseau social

Soutenu le :

Devant le jury :

M. S. Benotif	Président
M. ZAHRA	Rapporteur
M. ZAIR	Examinateur
Mlle. Boustia Narhimene	Promotrice
Mme. Guesmia Khalida	Encadrante

Promotion
2014 / 2015

Résumé:

Dans notre époque, le phénomène des réseaux sociaux a pris de l'ampleur, vu l'intérêt, beaucoup de questions de leur impact sur notre vie privée sont posées.

La gestion de confidentialité des données est primordiale pour éviter la divulgation de données privées au public. Dans ce cas le contrôle d'accès joue un rôle essentiel.

Le travail présenté dans ce mémoire propose un modèle de contrôle d'accès dynamique, simplifié et plus expressif que la plupart des réseaux sociaux présents actuellement ce qui offrira une meilleure gestion de la vie privée.

La solution mise en œuvre permettra aux utilisateurs de mieux sécuriser leurs comptes ainsi que les données partagées dans les réseaux, cette solution est flexible et est facilement intégrable aux réseaux sociaux actuels.

Mots clés: Réseaux sociaux, Services Web, Contrôle d'accès, Sécurité, Vie privée, Confidentialité.

ملخص:
في عصرنا هذا، رأت ظاهرة الشبكات الاجتماعية تطورا ملحوظا، مما أفضى إلى تساؤلات حول تأثير هذه الشبكات على حياتنا الخاصة.
إن إدارة خصوصيات المعلومات لهو أمر بالغ الأهمية لتجنب الكشف عن البيانات الخاصة للعامة، وفي هذه الحالة، فإن التحكم في الدخول يلعب دورا رئيسيا.
العمل المقدم في هذا البحث يوفر نموذجا ديناميكيا للتحكم في الدخول، وهو نموذج مبسط وأكثر شمولا من معظم الشبكات الاجتماعية الموجودة حاليا، وهذا من شأنه أن يوفر إدارة أفضل للخصوصية.
فإن الحل المنقذ يسمح للمستخدمين بأن يؤمنوا حساباتهم والبيانات المشتركة على نحو أفضل في الشبكة. وهو يتسم بالمرونة كما يمكن إدماجه بسهولة في الشبكات الاجتماعية القائمة.
الكلمات المفتاحية: الشبكات الاجتماعية، خدمات الشبكة العالمية، التحكم في الدخول، أمن، حياة خاصة، خصوصية.

Abstract:

In our time, the phenomenon of social networks has grown, in view of this interest, many questions about their impact in our life are asked.

The management of the confidentiality of the data is paramount to avoid the disclosure of private information to the public. In this case the access control plays a key role.

The present work in this project offers dynamic access control model, simplified and more comprehensive than most social networks currently present which provide a better management of privacy.

The implemented solution will allow the users to better secure their account and the shared data in the network, this solution is flexible and easily integrated into existing social networks.

Key words: Social networks, Web service, Access control, Security, Privacy, Confidentiality.

REMERCIMENTS:

Tout d'abord, on tient à remercier notre DIEU le tout miséricordieux de nous avoir permis de terminer ce projet.

Nous adressons nos grands remerciements à mlle Boustia , notre promotrice , pour ses conseils, son aide et ses encouragements.

Nous remercions notamment mlle Guesmia qui nous a aidés dans notre travail, et tous nos chers enseignants pour leurs efforts durant ces cinq années d'études.

Enfin, on remercie toute personne ayant contribué de prés ou de loin à la progression de ce projet.

DEDICACES:

Je dédie mon travail

À mes très chers parents qui ont toujours été là pour moi, et qui m'ont donné un magnifique modèle de labeur et de persévérance. J'espère qu'ils trouveront dans ce travail toute ma reconnaissance et tout mon amour.

À mon frère cadet Adel et à toute ma famille.

À tous mes ami(e)s et toute personne faisant partie de mon entourage.

WALID

DEDICACES:

À celle qui m'a donné la vie, le symbole de tendresse, qui s'est sacrifiée pour mon bonheur et ma réussite, à ma mère.

À mon père, école de mon enfance, qui a été mon ombre durant toutes les années d'études, et qui a veillé tout au long de ma vie à m'encourager

À mon adorable sœur maha et mon cher frère Imed

À mon binôme qui a su trouver les mots nécessaires pour me donner le courage d'accomplir ce travail ainsi que sa famille pour leur soutien.

À ma belle sœur Nassima pour son aide

À mon amour de nièce Kamelia

À tous mes amis: Ismail, Walid, Abdou et Zaki..... pour leur bonne humeur

Je dédie ce modeste travail

MEHDI

Liste des abréviations:

RS	Réseau social
SI	Système d'information
API	Application Programming Interface
PaaS	Plateform as a Service
SaaS	Software as a Service
IaaS	Infrastructure as a Service
DaaS	Data as a Service
E2EE	End to End Encryption
OrBAC	Organization-Based Access Control
ABAC	Attribut Based Access Control
DoD	Departement of Defense américain
XACML	eXtensible Access Control Markup Language
DAC	Discretionary Access Control
MAC	Mandatory Access Control
RBAC	Role Based Access Control
PEP	Policy Enforcement Point
PDP	Policy Decision Point
PAP	Policy Administration Point
PIP	Policy Information Point
CA	Contrôle d'accès
SOA	Service-Oriented Architecture
XML	Extensible Markup Language
UML	Unified Modeling Language
WSDL	Web Services Description Language
HTTP(S)	HyperText Transfer Protocol Secure
W3C	World Wide Web Consortium
SOAP	Simple Object Access Protocol
EDI	Échange de Données Informatisées
REST	Representational State Transfer
RPC	Remote Procedure Call
SMTP	Simple Mail Transfer Protocol

URI	Uniform Resource Identifier
WS	Web Service
RMI	Remote method invocation
CORBA	Common Object Request Broker Architecture
DCOM	Distributed Component Object Model
PHP	PHP: Hypertext Preprocessor
OMG	Object Management Group
MVC	Model View Controler
SQL	Structured Query Language
DOM	Document Object Model
GPL	General Public License

Table des matières:

Table des figures:

Introduction Générale:

Introduction:

Les sites de réseaux sociaux en ligne prennent de plus en plus de place dans notre vie quotidienne. Les plus connus sont bien évidemment Facebook, Twitter, Google+. L'émergence des réseaux sociaux est liée aux révolutions techniques et technologiques. Le développement des tablettes et des Smartphones ont également accéléré et accentué l'effet grandissant des réseaux sociaux. Ces sites proposent des fonctionnalités aux individus comme se rencontrer, interagir et partager des informations avec des personnes sur toute la planète. La popularité des sites de réseaux sociaux encourage les utilisateurs à dévoiler eux-mêmes leur propre vie privée. La divulgation de la vie privée à travers ces sites représente un danger indéniable pour les gens, d'autant plus que de nombreux utilisateurs sont présents sur ces sites et n'ont pas conscience des conséquences des informations qu'ils partagent.

Problématique:

Lorsqu'un utilisateur introduit une nouvelle donnée (statuts, photos, vidéos, liens) dans son espace personnel, il a le choix entre la rendre complètement publique ou la réserver à ses contacts, donc l'utilisateur est sensé renforcer les paramètres de confidentialité sur son profil afin de mieux contrôler ses données et limiter l'accès aux informations personnelles qu'il partage. Cependant, plusieurs études démontrent que de nombreux profils sont insuffisamment protégés, et ce, pour plusieurs raisons, les utilisateurs ont des difficultés à configurer correctement leurs paramètres de confidentialité ou à les modifier; une multitude d'options de paramètres de confidentialité à gérer; la politique de gestion de confidentialité n'est pas stable.

La plupart des réseaux sociaux n'offrent pas une hiérarchisation des contacts, la moyenne par utilisateur et de 130 amis et 80 groupes et événements. Suivant le degré de confiance, l'utilisateur devrait avoir le choix de donner plus de privilège à un contact que pour un autre et en conséquence moins d'accès à ces données personnelles à celui-ci. En plus de cela, le degré de confiance est instable, il peut augmenter ou diminuer suivant les humeurs de l'utilisateur.

Ajouté à ça, l'utilisateur n'a aucune maîtrise sur le temps ou le lieu, prenons exemple d'un utilisateur qui poste un statut exprimant son avis sur un sujet quelconque. Si cette opinion vient à changer dans un futur proche ou lointain, à défaut d'être supprimée, cette dernière restera toujours dans le mur (espace personnel) de cet utilisateur et peut-être actualisée à tout moment, dès qu'un autre utilisateur la commente ou la partage, cela peut causer défaut à l'utilisateur principal. De même pour le lieu, si un utilisateur crée un évènement, et souhaite que seuls les utilisateurs qui se situent dans sa région y participent.

En 2014, Facebook a fait face à une plainte massive déposée par un collectif qui regroupe plus de 20 000 internautes, l'objet de cette plainte est la violation du respect du droit à la vie privée de ses utilisateurs.

Objectifs:

Notre travail consiste à améliorer l'outillage de gestion de la confidentialité dans les réseaux sociaux. Notre objectif est de concevoir un outil qui aide les utilisateurs à gérer leurs paramètres de confidentialité avec un minimum d'effort et un grand niveau de précision.

Organisation du mémoire :

Ce mémoire est composé d'une introduction générale suivie des cinq chapitres qui représentent tout le travail effectué sur ce projet de fin d'étude, structuré comme suit:

Chapitre I: Présente les réseaux sociaux dans leur globalité, leurs caractéristiques et leurs différences avec les systèmes d'informations.

Chapitre II: On va évoquer les différents travaux qui ont étés réalisés dans le domaine de la sécurité dans les réseaux sociaux, ainsi que quelques modèles de contrôle d'accès.

Chapitre III: On va présenter l'approche utilisée pour réaliser des politiques de contrôle d'accès dynamique qui seront appliquées à notre réseau social.

Chapitre IV: Dans ce chapitre on va décrire la conception du projet et présenter la solution proposée.

Chapitre V: Il est composé d'une série de captures d'écran qui représenteront l'interface graphique, et des tests de l'application qui permettrons de valider notre solution.

Chapitre I: Réseaux Sociaux

1. Introduction:

Les réseaux sociaux (RS) sont ancrés dans la vie quotidienne de millions de personnes à travers le monde, chaque utilisateur peut utiliser ce moyen de communication pour s'exprimer et montrer son talent, ou tout simplement pour partager ces bons et mauvais moments.

Avec l'apparition des réseaux sociaux numériques, le spectre d'une marchandisation émerge, où l'individu constitue la matière première pour faire vivre le réseau. Ceci se produit dans une organisation collective de prise de parole, phénomène participatif encouragé par le développement du Web 2.0.

Le Web 2.0 [1] : Le terme web 2.0 représente la deuxième génération du Web à base de communauté (comme les blogs, Wikis, etc.). Il vise à faciliter la créativité, promouvoir la collaboration et le partage entre les utilisateurs. Avec le web 1.0, les internautes étaient juste des récipiendaires de l'information publiée par l'auteur. Cependant, avec le Web 2.0, les utilisateurs sont également des participants actifs à la création des contenus.

L'intérêt croissant aux RS apporte de nombreuses questions sur l'impact qu'ont ces derniers sur la vie privée de leurs utilisateurs. En effet, dans un RS chaque utilisateur est son propre administrateur, c'est lui qui gère sa confidentialité, contrairement au système d'informations classiques.

De nombreux travaux portent sur la confidentialité et la garantie de protection de la vie privée des utilisateurs des RS, à cause du manque d'une politique de contrôle d'accès. Qui a le droit de voir mes photos ? qui peut commenter mes vidéos ? qui peut me taguer (m'identifier)?.

2. Les réseaux sociaux:

Depuis sa création, l'être humain se base sur la socialisation pour vivre et évoluer. C'est le processus au cours duquel un individu apprend et intériorise les normes et les valeurs tout au long de sa vie dans la société à laquelle il appartient, c'est aussi l'échange d'idées, d'opinions et de savoir faire avec les autres.

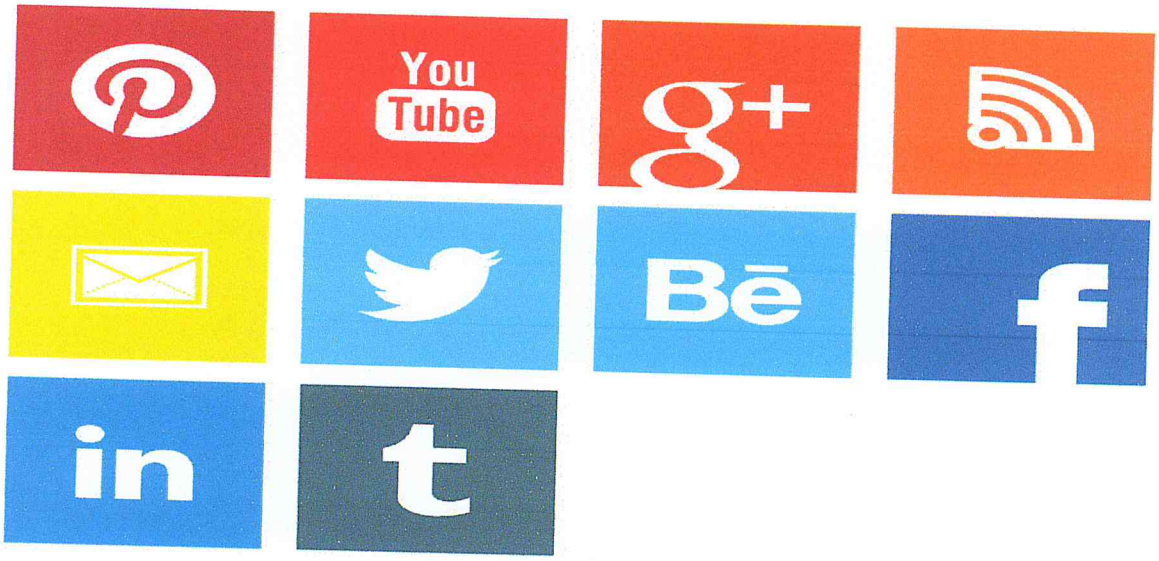


Figure 1.1- *Logos de réseaux sociaux.*

Partant de ce principe, les RS ont vu le jour. Ces derniers sont des services basés web qui permettent aux individus de construire un profil public ou semi-public dans un système fermé, de créer une liste d'utilisateurs avec lesquels ils partagent une connexion (lien entre deux utilisateurs), de visualiser et d'explorer leurs propres listes de contacts (ensemble de connexions) et celles des autres au sein du système. La nature et la nomenclature de ces connexions peuvent varier d'un site à un autre. Même si en parlant de RS, un mot nous vient automatiquement à l'esprit "Facebook", cependant il n'est pas le premier. Car en 1995 Randy Conrads a créé "Classemates" un site mis en ligne qui a pour objectif de remettre en contact des anciens camarades de classe[2].

Beaucoup d'autres ont suivi, comme le réseau social professionnel "Ryze". S'ils ont inspiré ceux qui réussiront à s'imposer plus tard, ils ont aussi servi à éviter de rééditer leurs erreurs. Facebook étendit ainsi le nombre de ses membres par étapes, contrairement à "Friendster" lancé un an avant et victime de son succès, les serveurs de l'entreprise furent incapables d'assumer la charge des nouveaux adhérents [3].

Les géants actuels sont Linkedin, Facebook, Twitter et bientôt Google +; nous allons nous intéresser à Facebook parce que c'est le leader mondial actuellement. Fondé en 2004 par Mark Zuckerberg, un utilisateur peut directement communiquer avec ses amis par messagerie, uploader différents types d'informations (photo, vidéo, etc) et les partager avec d'autres amis, il peut aussi rejoindre des groupes, ainsi que des pages fans, et organiser des événements [4]. Le site est devenu incontournable au fil des années.

Il dépasse aujourd'hui le milliard d'utilisateurs actifs mensuels. Les statistiques d'usage de Facebook sont impressionnantes, quelle que soit la variable étudiée; le nombre de membres, l'activité sur les mobiles, les résultats financiers, et les photos ajoutées chaque jour sur le service.

Comme toute avancée technologique, les RS ont leurs avantages et leurs inconvénients, c'est l'un de ces inconvénients, à savoir l'atteinte à la vie privée, sur lequel nous allons travailler.

2.1 Comparaison entre les RS et les systèmes d'informations classiques:

L'informatique a complètement changé le comportement des hommes, le monde est interconnecté et des échanges peuvent se produire entre des individus vivants dans des extrémités de la planète. Mais cela a commencé d'abord par des petites connexions au sein d'une même organisation, c'est la naissance des systèmes d'informations. Un système d'information (SI) est un ensemble de moyens humains et matériels ayant pour finalité d'élaborer, traiter, stocker, acheminer, présenter ou détruire l'information[5].

Même si l'objectif diffère complètement des RS, les SI rejoignent les RS sur le fait du partage de ressources entre utilisateurs. En effet, un SI met à disposition des données de l'entreprise pour les employés de celle-ci, et dans les RS c'est les utilisateurs qui mettent à disposition leurs propres données pour d'autres utilisateurs du RS. Il ya donc un contrôle d'accès à faire. Dans les SI, tous les employés n'ont pas le même droit d'accès, par exemple ceux qui travaillent dans le service marketing n'accèdent pas aux mêmes informations que ceux qui travaillent au service comptabilité. Il va de même pour les RS, les données (photos, vidéos, statuts...) mises par l'utilisateur pour ses amis ne seront pas accessibles par n'importe quel utilisateur sauf s'il appartient à sa liste d'amis.

Pour les SI, la sécurité contre les logiciels malveillants ou le courrier indésirable qui polluent nos environnements de travail, est assurée par le contrôle d'accès, car les grandes entreprises qui se basent généralement sur les SI redoutent la mise hors d'usage de leurs systèmes de production et la fuite d'informations confidentielles[6]. Le principe du contrôle d'accès au sein d'un SI repose sur l'authentification et l'autorisation.

Un administrateur établit la politique de sécurité et soit autorise l'accès à un employé, soit l'en prive. Avec l'arrivée de RBAC (Role Based Access Control), la notion de rôle est prise en compte, et l'accès est accordé à un groupe d'employés ayant le même rôle.

Cela est faisable car les SI représentent une organisation bien structurée et l'administrateur a connaissance du nombre et des rôles des employés.

Tandis que pour les RS cette notion disparait, même la notion d'administrateur n'existe plus car chaque utilisateur est son propre administrateur, donc c'est lui qui gère ses règles de sécurité. Et c'est là où se situe le problème, l'utilisateur fait face à un nombre de données interdépendantes massives et à un très grand nombre d'adhérents, en plus, le panel de choix proposé par les RS concernant la confidentialité ne satisfait pas toutes les exigences en termes de sécurité.

Le défi qui s'impose à nous est de prendre exemple sur les politiques mises en place dans les SI et qui ont fait leurs preuves, et de les adapter et les mettre à disposition des utilisateurs des RS avec un maximum de clarté.

2.2 Les caractéristiques des réseaux sociaux:

Nous décrivons ci-dessous les composantes principales des réseaux sociaux:

2.2.1 Profils:

Les profils peuvent être considérés comme les briques de base du RS. Les profils contiennent généralement des informations démographiques basiques sur l'utilisateur tel que son nom, son sexe, sa ville natale et sa profession actuel…etc. Parallèlement à ces informations personnelles considérées essentielles pour chaque profil, la plupart des RS encouragent également les utilisateurs à écrire une courte biographie sur eux-mêmes et de partager leurs goûts et leurs intérêts.

Pourtant ces types d'informations ne sont pas obligatoires pour pouvoir s'inscrire sur les RS, de nombreux utilisateurs mettent beaucoup de détails facultatifs sur leurs profils.

2.2.2 Les amis:

La plupart des RS sont conçus et construits autour du concept d'«amis» ou «Friends». Sur un RS, un «ami» peut être un ami, un membre de la famille, une connaissance, un ami d'un ami, ou même quelqu'un que l'utilisateur n'a jamais rencontré auparavant, sauf en ligne. En 2013, le nombre moyen d'amis pour chaque utilisateur sur Facebook est de 175[2]. Le RS permet à l'utilisateur de garder la trace des activités de ses amis.

Par exemple, quand ils publient une nouvelle photo, mettent à jour leurs profils, changent leurs statuts ou lorsqu'ils achètent quelque chose de nouveau en ligne. Le RS a généralement une fonctionnalité de recherche qui peut aider l'utilisateur à trouver de nouveaux amis.

2.2.3 Fonctionnalités de réseautage:

En plus des relations d'amitié, certains RS proposent également des fonctionnalités de réseautage pour faciliter l'interaction entre les utilisateurs, tels que les groupes et la messagerie instantanée. Chaque RS a aussi des fonctionnalités particulières propres à lui tels que l'envoi des « pokes » sur Facebook.

2.2.4 Les groupes:

La plupart des RS s'appuient sur la notion de groupe pour aider les utilisateurs à trouver des personnes ayant des intérêts similaires ou à s'engager dans des discussions sur certains sujets. Parfois, les groupes sont appelés par d'autres noms, tel que «les réseaux» sur Linkedin.

2.2.5 Les évènements:

C'est une fonctionnalité de réseautage permettant aux «amis» de connaître les événements à venir dans leur communauté ainsi que d'organiser des rassemblements sociaux. Par exemple, sur MySpace, il est possible de publier un questionnaire ou de décorer la page de l'événement.

2.2.6 Les « tags »:

Un tag est un mot-clé ou terme assigné à un élément d'information. Par exemple, un tag peut être un bookmark en ligne, une photo numérique ou un fichier.

Ce type de métadonnées décrit un objet et permet de le trouver par la recherche ou en navigant. Facebook et Friendster permettent aux utilisateurs d'associer un tag à une zone spécifique dans l'image. Par exemple, l'utilisateur peut taguer les personnes figurant dans une image d'une famille dans une place particulière par leurs noms et mettre un tag pour spécifier le nom de la place où la photo a été prise. Si le nom utilisé pour le tag est associé à un membre de Facebook ou à une page (région connue par exemple), le tag se transforme en un lien hypertexte vers le profil ou la page.

2.2.7 Flux d'actualité (News Feeds):

Les flux d'actualité sont des outils utiles pour rester en contact avec les «amis». Par exemple, les mises à jour de profil, les messages sur le blog, les photos et vidéos publiées sont souvent diffusées sous forme de «news feeds» .

2.2.8 Les applications sociales:

Les RS comprennent un grand nombre d'applications sociales que les utilisateurs peuvent ajouter à leur profil. Ces applications peuvent être programmées à travers des interfaces ouvertes aux développeurs tiers pour concevoir et mettre en œuvre des applications ou des jeux sur la plateforme.

Les applications créées à l'aide de ces API[1] posent des problèmes pour la vie privée car elles demandent aux utilisateurs d'accéder à leurs informations personnelles (situation familiale...etc.), à leurs listes d'amis, aux informations personnelles de leurs amis…etc. Voire même de publier sur leur profil et avoir un accès permanent à ces derniers.

3. Conclusion:

Les recherches effectuées pour réaliser ce chapitre nous ont permis de connaître les principaux leaders mondiaux des réseaux sociaux, le fonctionnement et les caractéristiques de chacun d'eux. Aussi des recherches avancées sur leurs architectures de sécurité, ce qui nous a permis de nous orienter et avoir une vue globale tout en détaillant ce qui existe déjà. Nous allons évoquer dans le chapitre suivant les différents modèles de contrôle d'accès existants.

[1] Application programming interface est une interface de programmation qui est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert de façade par laquelle un logiciel offre des services à d'autres logiciels. Elle est offerte par une bibliothèque logicielle ou un service web.

Chapitre II: Modèles de contrôle d'accès

1. Introduction:

À travers ce chapitre, nous allons voir dans un premier temps les mesures de sécurité concernant la vie privée des utilisateurs mises en place par les RS, suivies d'un ensemble de travaux réalisés sur ce sujet. Et tenter de trouver une solution adéquate à la problématique soulevée dans le cadre de notre projet. en présentant quelques modèles de contrôle d'accès jugés intéressants.

2. Contrôle d'accès pour les réseaux sociaux:

De nombreux RS utilisent le Cloud Computing[2] pour la catégorie de services qu'il offre, plate-forme en tant que service (PaaS), logiciel en tant que service (SaaS), infrastructure en tant que service (IaaS), donnée en tant que service (DaaS) [7].

Un de ces services est en charge d'assurer l'élaboration de la politique de contrôle d'accès, il s'agit du DaaS. Pour gagner en temps d'exécution, la plupart des réseaux sociaux adoptent un contrôle d'accès statique.

3. Travaux réalisés sur la vie privée dans les RS:

En vue du nombre impressionnant d'adhérents aux RS, de nombreux chercheurs de différentes disciplines se sont intéressés à ce nouveau phénomène, surtout dans le domaine de l'informatique. Il ya un large panel de travaux effectués sur plusieurs brèches trouvées dans les RS, mais néanmoins l'une des questions primordiales est la sécurité.

En effet, des millions d'utilisateurs partagent, commentent, postent des photos et des vidéos le plus souvent personnelles, d'eux ou de leurs amis, sans avoir la moindre idée des conséquences que cela peut impliquer. Sensibiliser les gens à cette réalité ne suffit pas, car les politiques de confidentialité de ces RS sont jugées insuffisamment claires et rigoureuses pour assurer la sécurité de la vie privée des utilisateurs. Nous allons voir quelques travaux jugés intéressants sur ce sujet.

[2] Sorte de calcul hautement évolutif qui utilise les ressources virtuelles et qui peuvent être partagées par des utilisateurs.

3.1 Un nouveau style de contrôle d'accès pour Facebook:

La popularité des réseaux sociaux a fait de la protection des renseignements personnels des utilisateurs un problème important. Dans la littérature liée à cette question, des systèmes de contrôle d'accès basés sur les relations ont été proposés pour résoudre ce problème.

Cependant, avec l'évolution dynamique des RS, nous identifions de nouvelles exigences en matière de contrôle d'accès qui ne peuvent pas être totalement prises en compte actuellement [8]. Dans ce travail, les chercheurs ont démontré le danger trouvé dans les RS pour la vie privée de ces utilisateurs, ils se sont concentrés sur les informations public dans les RS et proposent une nouvelle dimension que les utilisateurs peuvent utiliser pour réglementer l'accès à leurs ressources. En donnant des catégories aux ressources et aux contacts (Amis, Famille, Collègue par ex), une variante est calculée permettant l'accès à une ressource public. La notion de public serait donc restreinte à des groupes bien définis.

Ce travail n'a pas abouti en pratique pour sa complexité. En effet, pour que la solution soit réalisable, les réseaux sociaux actuels devraient changer une grande partie de leur code, et augmenter les interactions avec la base de données.

3.2 Encryptage des réseaux sociaux en ligne:

Des études menées sur les RS ont montré une faille de sécurité au sein de ces derniers, celle-ci se penche sur la divulgation non désirée et l'utilisation des messages privés des utilisateurs [9,10].

En partant de cette constatation, des chercheurs ont voulu mettre en œuvre des techniques cryptographiques existantes pour protéger la vie privée des utilisateurs des RS, comme le chiffrement de bout en bout (E2EE).

En permettant aux utilisateurs de crypter leurs messages privés, les RS peuvent renforcer leur contrôle d'accès en protégeant les données des utilisateurs grâce à ce cryptage, en particulier pour les postes qu'on veut garder confidentiels du grand public, pour qu'ils ne soient pas sujet à des divulgations involontaires et diminuer les risques de violation de la vie privée.

Le cryptage permettrait également aux utilisateurs d'avoir plus de choix granulaire pour le type de confidentialité de leurs données[11].

Mais cette approche risque de surcharger le système, beaucoup de RS sont réticents à l'adopter. En plus, la mise en place d'un cryptage tel que E2EE serait trop complexe et engendrerait un délai de mise en place long et couteux pour les RS.

3.3 Règles de contrôle d'accès basé sur les réseaux sociaux:

En 2006, Barbara Carminati, Elena Ferrari et Andrea Perego proposent un mécanisme de contrôle d'accès pour les réseaux sociaux sur le web, qui adopte une approche à base de règles pour spécifier les politiques d'accès sur les ressources détenues par les participants du réseau, et, où les utilisateurs autorisés sont indiqués en fonction du type, de la profondeur, et du niveau de relation de confiance entre les nœuds du réseau. Différent des systèmes de contrôle d'accès traditionnels, leur mécanisme fait appel à une architecture semi-décentralisée, où l'application de contrôle d'accès est réalisée côté client. L'accès à une ressource est accordé lorsque le demandeur est en mesure de démontrer qu'il est autorisé à accéder en fournissant une preuve [12].

L'idée de spécifier de nouvelles politiques d'accès à granularité plus fine est intéressante, mais le fait d'obliger les utilisateurs à fournir une preuve à chaque fois qu'ils veulent accéder à une ressource en sachant que ce processus ne peut pas se faire automatiquement, (l'utilisateur doit saisir un code ou s'identifier de nouveau) risque de lasser ce dernier.

3.4 Un modèle de contrôle d'accès contextuel de réseau social en ligne:

Le dernier travail sélectionné, propose de spécifier de nouvelles politiques de contrôle d'accès basées sur la notion de rôle, le modèle choisi a été OrBAC, car son entité centrale est l'Organisation [13].

Suivant cette approche, les chercheurs considèrent les RS comme des organisations (le profil de l'utilisateur, page, fan, groupe, application et événement). Un sujet sera toute entité active dans un système qui accède à des objets [14].

Ce travail met le point aussi sur la notion de temps, de lieu et de contexte, qui ne sont pas pris en compte par les RS.

Nous adhérons complètement à cette approche, à un détail prêt, nous pensons qu'avec ABAC nous aurons plus de résultats espérés. Car la notion d'attribut utilisé dans ABAC offre plus de flexibilité.

Comme les réseaux sociaux sont un domaine totalement nouveau, de nombreuses recherches ont été effectuées, notamment sur la question de la sécurité, notre choix s'est porté sur les travaux sélectionnés parce qu'ils sont très explicites et résument bien l'idée de départ où tous les travaux se rejoignent.

4. Les modèles de contrôle d'accès:

Dans les années 70, Departement of Defense américain (DOD) a lancé les premières recherches sur le contrôle d'accès.

Le contrôle d'accès a pour objectif de contrer les atteintes à la confidentialité (divulgations d'informations non autorisées), contrer les atteintes à l'intégrité (modifications non autorisées) et contrer des atteintes à la disponibilité (déni de service). Afin d'assurer une protection, chaque accès aux données doit être contrôlé, et bien évidemment tous les accès non autorisés doivent être impérativement bloqués. Cela s'exprime en vérifiant si un sujet demandant d'accéder à un objet, possède les droits nécessaires pour le faire. Il est régi avec des règles qui peuvent être exprimées en différents langages tels que Ponder et XACML.

Dans une règle de contrôle d'accès nous trouvons les paramètres suivants:

- Le sujet qui peut être un utilisateur, une machine, un processus, un programme, etc.
- L'objet qui peut être un fichier, une base de données, une machine, un programme, etc.
- Le droit d'accès qui désigne l'effet recherché lorsqu'un sujet accède à un objet (lire, écrire, modifier, etc.).
- Le contexte qui est une contrainte qui lie le sujet, le droit d'accès et l'objet (contexte temporel, contexte spatial, etc.)[15].

Tout système informatique doit élaborer des politiques de sécurité basées sur le contrôle d'accès, les RS n'échappent pas à cette règle, vue la quantité de données partagées par les utilisateurs, cette tâche reste délicate et complexe à élaborer mais néanmoins importante pour garantir la confidentialité de la vie privée des utilisateurs.

4.1 Le contrôle d'accès discrétionnaire:

Les politiques de contrôle d'accès discrétionnaires s'appuient sur les notions de propriété (tout sujet est propriétaire d'un ensemble d'objets) et de droit d'accès (lecture, écriture, etc). Le contrôle d'accès est dit discrétionnaire lorsque la technique de restriction d'accès aux objets est basée sur l'identité des sujets et/ou des groupes auxquelles ils appartiennent. Le contrôle est discrétionnaire dans le sens où un sujet possédant un certain droit d'accès est capable de conférer ce droit à tout autre utilisateur.

Le contrôle d'accès discrétionnaire ou «Discretionary Access Control» (DAC) [16] permet à un sujet d'attribuer des permissions à d'autres sujets. Ce contrôle d'accès est flexible mais peut générer des erreurs. Il pose quelques problèmes comme la vulnérabilité aux chevaux de Troie, et il ne permet pas d'assurer des propriétés de sécurité telles que la confidentialité et l'intégrité car les permissions d'accès se font "à la discrétion" du propriétaire d'un objet. En effet, le propriétaire d'un objet peut attribuer à un sujet malveillant l'accès à des informations importantes parce qu'il n'a pas une vision globale du système.

4.2 Le contrôle d'accès mandataire:

Le concept MAC (Mandatory Access Control) a été introduit par Bell et LaPadula [17], il est utilisé principalement dans les environnements militaires à cause de son contrôle centralisé et il permet à l'administrateur du système de définir des privilèges pour protéger la confidentialité et l'intégrité des ressources dans le système.

Le contrôle d'accès est dit mandataire (ou obligatoire) lorsque l'accès aux objets est basé sur le niveau de sensibilité de l'information (Figure 2.1) contenue dans les objets.

L'autorisation d'accéder à un objet 1 est accordée à un sujet 2 si le niveau d'autorisation de ce sujet est en accord avec le niveau de sensibilité de l'information [18].

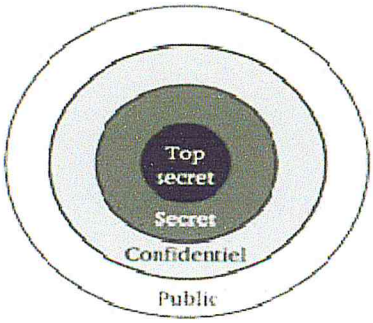


Figure 2.1 - *Les niveaux de sensibilité dans le modèle MAC[18].*

Il attribue un niveau de sécurité à chaque utilisateur et à chaque ressource. On accorde l'accès à un utilisateur seulement si son niveau de sécurité est supérieur ou égal au niveau de la ressource à laquelle il veut accéder.

Les règles diffèrent selon qu'il s'agisse de maintenir des propriétés de confidentialité ou d'intégrité. Les politiques obligatoires les plus fréquemment utilisées sont les politiques multi-niveaux. Ces politiques reposent sur des classes de sécurité affectées aux informations et aux niveaux des habilitations affectées aux utilisateurs [18].

4.3 Le contrôle d'accès basé sur les rôles RBAC:

RBAC (Role Based Access Control) ou le modèle de contrôle d'accès basé sur les rôles est à l'origine du concept de rôle. Il représente les permissions comme des couples *(objet o, action a)* avec ($o \in O$ *et* $a \in A$) affectés à des rôles spécifiques au lieu d'être affectés directement à des sujets. Ce modèle simplifie les opérations tels que l'ajout ou la suppression d'un sujet.

Le rôle est une notion permettant de décrire facilement les fonctionnalités des organisations. Il désigne une entité intermédiaire entre utilisateurs et privilèges.

On associe à chaque rôle un ensemble de permissions. Tous les sujets ayant reçu l'autorisation de jouer un rôle héritent alors des permissions associées à ce rôle, la figure 2.2 illustre ce modèle.

L'utilisation de la notion de rôle apporte un certain nombre d'avantages. La compréhension de la structure de l'organisation est facilitée et la complexité de gestion des droits d'accès est réduite.

Les rôles peuvent être organisés de manière à former une hiérarchie [19] permettant ainsi de raffiner les différentes permissions attribuées à chaque rôle. En effet, les permissions ne sont pas attribuées aux sujets séparément et les sujets ne peuvent acquérir ces permissions qu'à partir de leurs rôles ce qui fait que RBAC est considéré comme un système « idéal » pour les organisations dont la fréquence de changement du personnel est élevée.

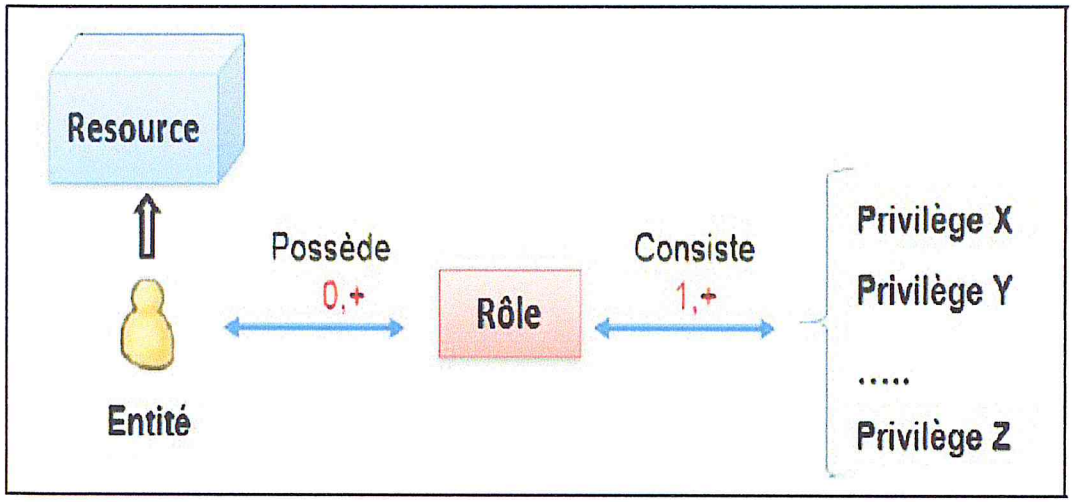


Figure 2.2 - *Modèle RBAC[20].*

4.4 Le contrôle d'accès basé sur l'organisation OrBAC:

Le contrôle d'accès basé sur l'organisation OrBAC (Organization-Based Access Control) a été présenté pour la première fois en 2003 [18]. Il reprend les principes de rôles des modèles du type RBAC, en offrant en plus, la possibilité de modifier la politique de sécurité de façon dynamique en fonction d'un contexte.

Dans OrBAC, la possibilité d'exprimer des permissions, des obligations et des interdictions qui dépendent de contextes, est un élément qui va vers une plus grande expressivité. L'abstraction des entités traditionnelles du contrôle d'accès (sujet, action, objet) en méta entités (rôle, activité, vue) permet d'élaborer une politique de sécurité à deux niveaux, un niveau concret et un niveau abstrait [21].

L'introduction d'un niveau abstrait organisationnel permet aussi la structuration des entités comme on le voit sur la figure 2.3 : Nous obtenons alors une politique de sécurité à deux niveaux, le modèle OrBAC permet ainsi d'établir une politique de sécurité abstraite (rôle, activité, vue) indépendante des choix d'implémentation (sujet, action, objet).

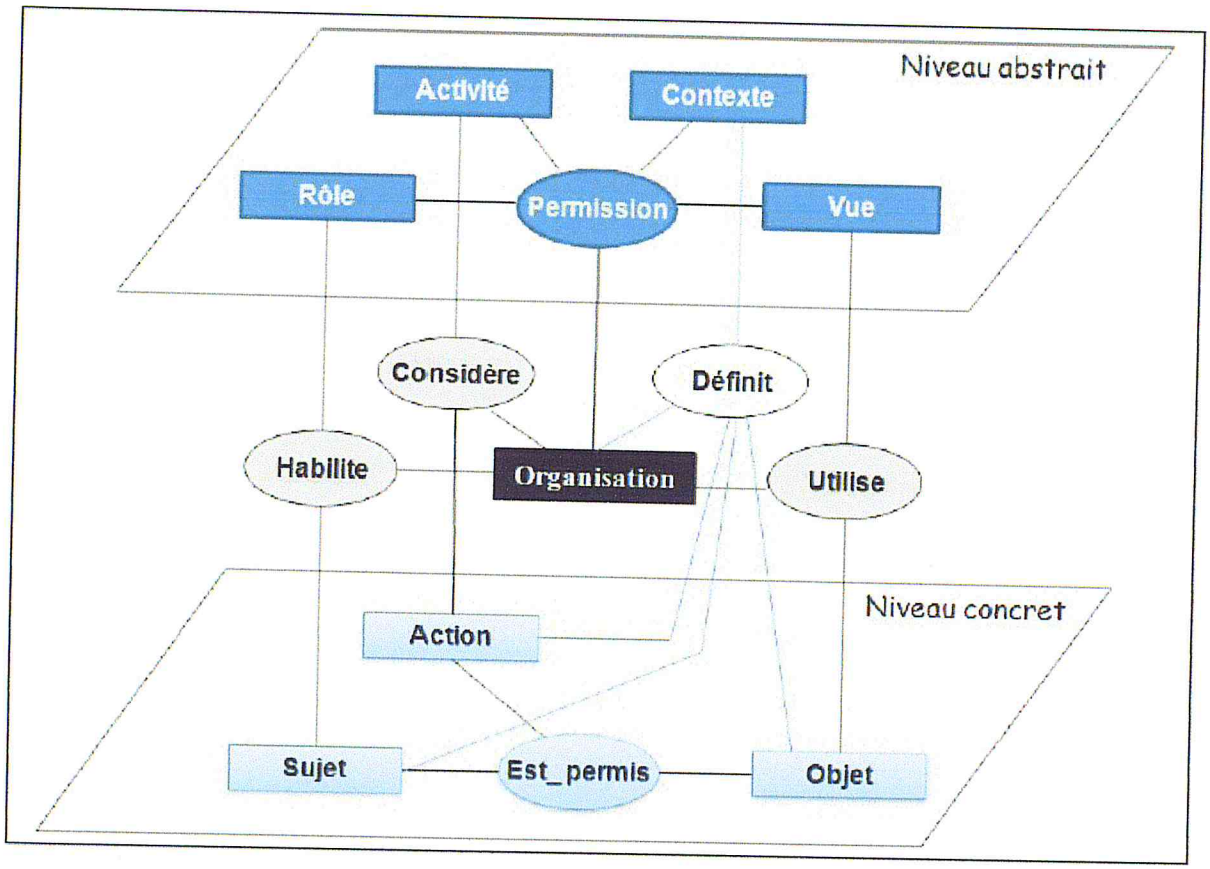


Figure 2.3 - *Le modèle OrBAC[20].*

4.5 Le contrôle d'accès basé sur l'attribut ABAC:

ABAC (Attribute Based Access Control) est un modèle de contrôle d'accès, où la décision est prise sur la base de condition booléenne sur des valeurs d'attributs.

Les entités traditionnelles (sujet, objet, action) sont des catégories qui regroupent des attributs. Ces derniers, sont des jeux d'étiquettes ou des propriétés qui peuvent être utilisées pour décrire toutes les entités qui doivent être considérées à des fins d'autorisation. Chaque attribut se compose d'une paire clé-valeur.

Par exemple les attributs du sujet pourrait être Nom(sujet)=Walid, Genre(sujet)=masculin, Age(sujet)=24 etc.....

Dans ABAC, les requêtes de contrôle d'accès sont exprimées à l'aide d'ensemble de couple (attribut= valeur), elle représente l'état du demandeur.

P.ex. (Nom(sujet)=Mehdi) et (Age(sujet)= 24) et (Identif(action)=lecture) et (Identif(ressource)=livre ABAC).

Les politiques de contrôle d'accès dans ABAC sont un ensemble de règles qui sont reliées par une idée commune, ces règles sont basées sur des cibles représentées par des expressions booléennes (la logique).

Par ex: Permettre si (Type(sujet)= étudiant) et (Type(action)= consultation) et (Type(ressource)=livre en réserve) et (Temps(requête)= heures de travail).

ABAC possède une architecture qui est décomposée en éléments qui peuvent être implémentés différemment.

- PEP: Policy Enforcement Point:

  Donne ou refuse un accès

- PDP: Policy Decision Point:

  Prend la décision si l'accès doit être donné ou refusé, en utilisant 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.

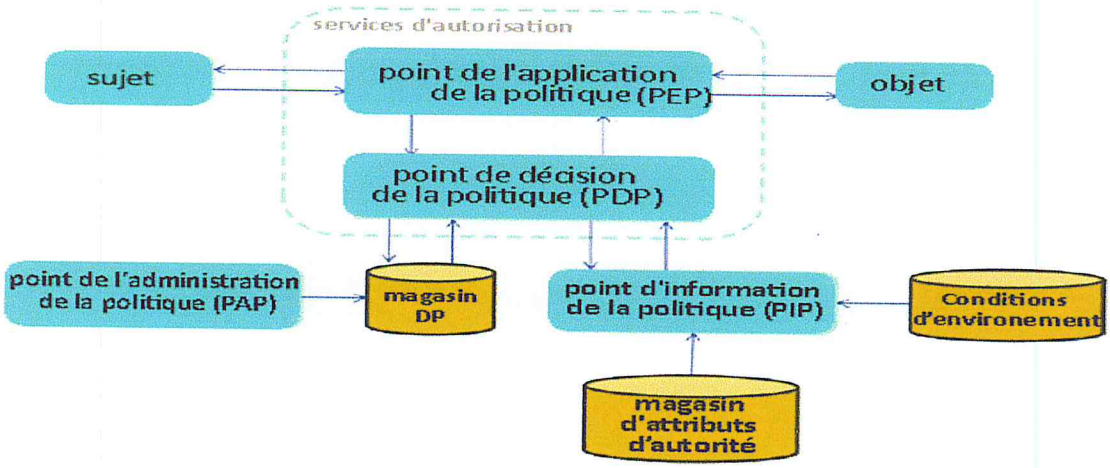


Figure 2.4 - *Architecture du modèle ABAC[22].*

La figure 2.4 nous montre les différentes interactions entre les éléments architecturaux d'ABAC.

ABAC offre aussi une approche orientée attribut. Les attributs sont des entités nécessaires à la description de la politique. ABAC attribue des caractéristiques particulières aux différents éléments nécessaires à l'expression d'une requête de demande d'accès. Les différentes entités retenues dans ABAC sont les sujets, les objets (appelés aussi ressources), les actions et l'environnement. Pour chacune de ces entités, on peut définir les attributs nécessaires à l'expression d'une politique de CA (contrôle d'accès). ABAC est étroitement lié au langage XACML, dont l'utilisation est détaillée dans le chapitre suivant[23].

Le contrôle d'accès ABAC, est apparu pour pallier les limites des contrôles d'accès standards. Les droits d'accès à une ressource ou un service sont définis pour un ou plusieurs attributs que les identités sont susceptibles de posséder. Ce paradigme offre donc plus de flexibilité. De plus, en définissant un attribut se rapprochant de la notion de rôle, ABAC permet de simuler le comportement d'un modèle RBAC, mais le généralise en ne limitant pas les droits d'accès aux seuls utilisateurs présents dans l'organisation. Il permet notamment de déterminer des droits d'accès avec une granularité plus fine. De plus, en définissant un rôle comme un ensemble d'attributs, il est plus facile de gérer les conflits. Par ailleurs, la gestion des droits d'accès est facilitée, car elle ne nécessite pas d'informations supplémentaires. Cependant, la sécurité des accès repose alors sur les valeurs affectées aux attributs et donc sur la qualité et l'intégrité des informations liées aux identités.

5. Comparaison entre les modèles de contrôle d'accès:

Quel que soit le modèle implémenté au sein d'une organisation, l'objectif est de limiter la capacité d'action des entités utilisatrices sur les ressources au strict nécessaire pour réaliser leurs missions.

Le contrôle d'accès discrétionnaire est généralement défini par opposition au contrôle d'accès obligatoire (MAC) [17] qui impose des règles incontournables garantissant l'atteinte des objectifs de sécurité visés. Dans ce type de contrôle d'accès les sujets ne peuvent pas intervenir dans l'attribution des droits d'accès.

Ce contrôle d'accès est plus rigide que le contrôle d'accès discrétionnaire mais plus sûr. MAC est donc un système supportant une politique de contrôle d'accès obligatoire, il peut être utilisé aisément dans une administration centralisée [18].

Par contre, il n'est pas recommandé pour les environnements distribués, car les utilisateurs n'ont pas suffisamment de privilèges pour gérer leurs propres besoins de confidentialité, notamment ceux concernant la vie privée.

Bien que les modèles DAC et MAC ont fait leurs preuves en matière de sécurité ils ont exprimé des limites à ce sujet et un manque de flexibilité, vu qu'ils ont une approche portée sur le sujet, c'est-à-dire qu'ils attribuent directement les permissions aux sujets, or si un sujet venait à disparaître de l'organisation il faut refaire toute la politique pour un nouveau sujet.

Le modèle RBAC permet de diminuer la taille de liste des habilitations. Les contrôles d'accès sont réalisés sur les rôles attribués aux comptes. Les rôles applicatifs sont octroyés en fonction du profil métier. Cependant, le contrôle d'accès basé sur les rôles est insuffisant pour satisfaire tous nos besoins en matière de protection. L'un des problèmes majeurs de ce modèle est le fait que tous les utilisateurs associés au même rôle possèdent forcément les mêmes privilèges. Ceci réduit la flexibilité des politiques de sécurité. En effet, l'expression des aspects contextuels liée aux autorisations d'accès n'est pas présente dans le modèle RBAC, on constate une confusion entre rôle et organisation. On peut seulement exprimer les permissions (pas d'interdiction), ce qui entraîne une gestion complexe des exceptions.

Dans le modèle OrBAC les autorisations ou interdictions reposent sur des expressions contextuelles définies d'après la structure organisationnelle de l'établissement. Même si OrBAC offre un très grand nombre d'avantages mais son implémentation serait complexe, car les RS même si on peut les considérer comme une organisation, vu le nombre d'utilisateurs et de ressources mis à disposition, on ne pourra pas définir l'ensemble des contextes et la hiérarchisation de tous ces objets.

7. Conclusion:

A travers ce chapitre nous avons étudié quelques modèles de contrôle d'accès et quelques travaux qui on été fait sur ces derniers, grâce à des comparaisons faites sur ces modèles, cela nous a permis de choisir le modèle ABAC qui satisfait le plus nos besoins pour notre travail.

Dans le chapitre suivant, on présentera l'implémentation d'ABAC ainsi que le serveur de gestion des droits qu'on va utiliser.

Chapitre III: Serveur de gestion des droits

1. Introduction:

Dans ce chapitre on va présenter les différents outils qu'on va utiliser afin de réaliser et de développer notre propre modèle de contrôle d'accès dans un réseau social.

Après de nombreuses recherches et l'aperçu global de l'état de l'art, nous avons constaté que les modèles de contrôle d'accès mis en place par les RS sont incomplets et ils ne répondent pas aux exigences des utilisateurs et sont souvent incompris par ces derniers. On a remarqué aussi qu'il y a un manque dans le contrôle de confidentialité de chaque utilisateur et cela est dû à l'absence de la notion d'attribut. C'est ici que notre regard s'est tourné vers ABAC, un modèle de contrôle d'accès qui est basé sur les attributs des utilisateurs d'un système. Pour réaliser cela, nous allons utiliser la spécification XACML pour rédiger la politique de contrôle d'accès dynamiquement, et pour permettre cela, nous associerons un serveur de gestion des droits.

2. XACML:

XACML (eXtensible Access Control Markup Language), est une spécification qui définit un langage pour le contrôle d'accès, la circulation des règles et l'administration de la politique de sécurité des systèmes d'informations. Elle est souvent utilisée pour assurer la fonction d'autorisation dans les architectures SOA[3] et dans les web services. XACML est une norme d'OASIS[24], c'est une organisation de normalisation internationale, qui travaille sur plusieurs normes de sécurité, et qui a été développée depuis plusieurs années; et la première version est apparue en 2003 et depuis 2013 nous somme à la version 3.0.

Le XACML fournit une architecture du système, une bonne base pour l'implémentation, des principes de communication entre les composants, un langage pour les règles et les politiques, un langage pour les requêtes et les réponses, types de données normalisés (fonctions, algorithmes de combinaison) et de l'extensibilité. Interopérable car basé sur XML, XACML est donc capable de faire communiquer les politiques de contrôle d'accès dans un environnement de services web, un service applicatif distribué sur internet (comme les réseaux sociaux) peut ainsi en invoquer un autre dans la mesure où lui sont alloués les droits correspondants.

[3] Est une forme d'architecture de médiation qui est un modèle d'interaction applicative qui met en œuvre des services

De même, une politique de contrôle d'accès décrite en XACML peut se référer à une autre. Il est ainsi possible de poser des règles *'locales'*, s'appuyant sur celles définies au niveau central.

Cette notion est très importante pour notre travail, car on va s'appuyer sur les politiques de contrôle d'accès mis en place pour développer les nôtres et les combiner à la fin.

```
<Policy PolicyId="ExamplePolicy"
       RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">
  <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">http://server.example.com/code/docs/developer-guide.htr
          <ResourceAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#anyURI"
                                       AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
  <Rule RuleId="ReadRule" Effect="Permit">
    <Target>
      <Subjects>
        <AnySubject/>
      </Subjects>
      <Resources>
        <AnyResource/>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
            <ActionAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#string"
                                       AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
```

Figure 3.1 - Exemple d'une politique XACML.

2.1 Les bases d' XACML:

XACML se base sur la technologie XML pour la syntaxe du langage. Comme il est conçu pour les architectures SOA, il possède une architecture décomposée en PEP, PDP, PIP et PAP comme le modèle d'accès ABAC.

Le langage XACML permet d'écrire trois types d'objets qui sont:

2.1.1 La politique de contrôle d'accès:

En XACML, la politique de contrôle d'accès est décomposée en plusieurs règles. Ces règles correspondent à des autorisations ou des refus pour une personne d'obtenir l'accès à un objet. Pour écrire ces règles, les caractéristiques du sujet et/ou des objets du système sont utilisées pour définir des contraintes.

2.1.2 Les requêtes:

Elles sont envoyées par le système au PDP (Policy Decision Point). Elles contiennent les données nécessaires pour que le PDP puisse prendre une décision en adéquation avec la politique de contrôle d'accès. Les requêtes contiennent les données relatives au sujet (réalisant la requête), l'action qu'il souhaite exécuter, les objets sur lesquels porte la requête et l'environnement (i.e. toutes les données ne rentrant pas dans les catégories précédentes comme le moment).

2.1.3 Les décisions:

Correspondent à la réponse du PDP pour une requête donnée.

2.2 Architecture XACML:

Nous avons motionné plus haut que l'architecture d'XACML est décomposée en PAP, PDP, PIP et PEP. On va tâcher maintenant de mieux définir cette architecture.

Cette architecture est présentée à la figure 3.2 tirée de [24]. Cette architecture s'appuie sur différents éléments présents dans une architecture SOA.

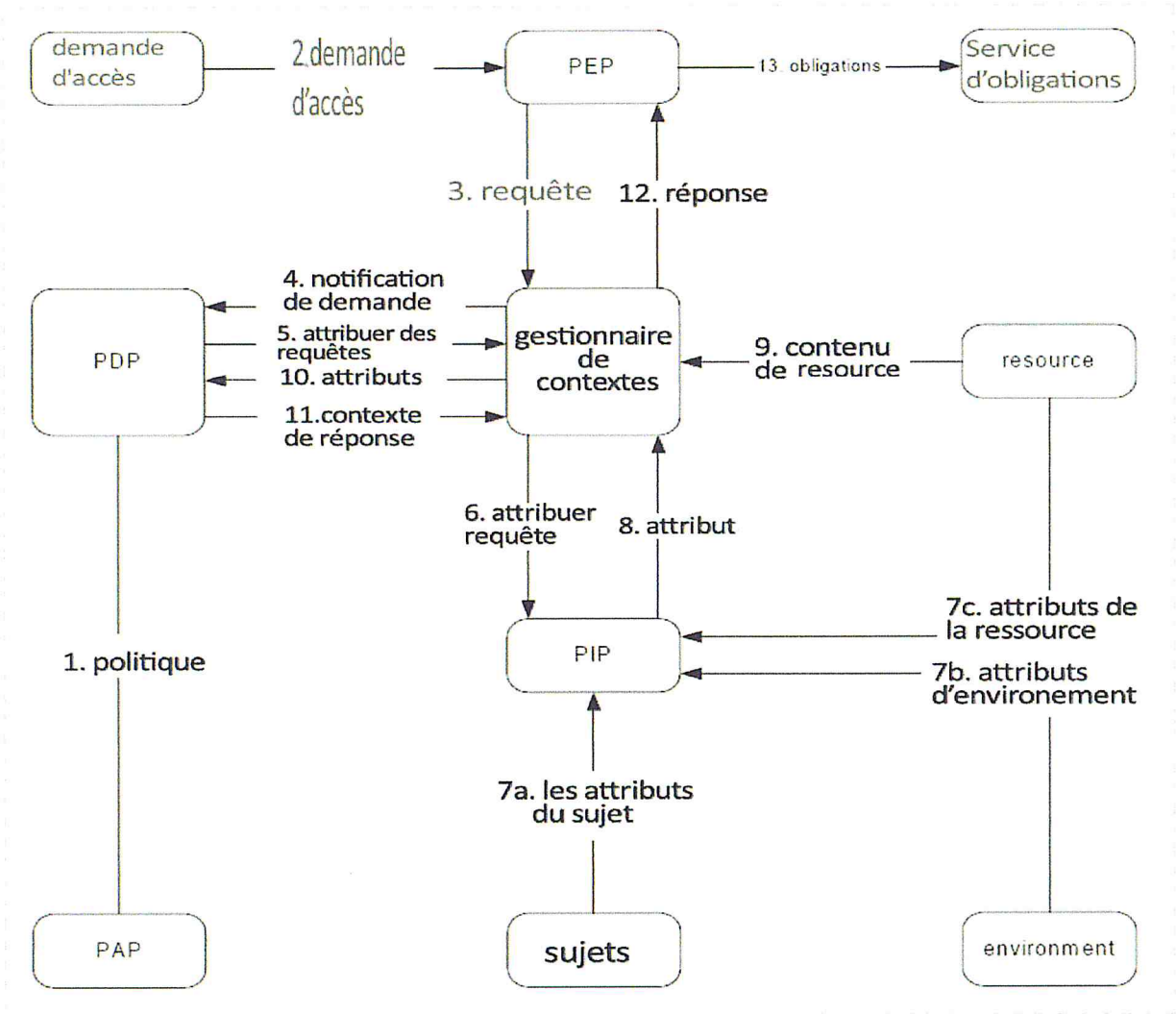


Figure 3.2 - *Architecture standard de l'implémentation d'une solution XACML[24].*

La politique de contrôle d'accès est stockée dans le PAP (Policy Administration Point). Elle est récupérée par le PDP (flèche 1). Les requêtes sont interceptées par le PEP (Policy Enfoncement Point) (flèche 2) et envoyées au context handler (flèche 3).

Ce dernier les convertit de leur forme native vers la forme canonique utilisée en XACML avant de les envoyer vers le PDP (flèche 4). Le PDP les reçoit, les analyse, rapatrie les règles de contrôle d'accès relatives à la requête et évalue la demande en fonction de la politique avant de renvoyer la décision. Le PIP (Policy Information Point) permet de connaître les valeurs des attributs nécessaires à la prise de décision (flèche 5 à 10). La décision est envoyée par le PDP au context handler (flèche 11), qui la convertit en langage natif avant de la renvoyer au PEP (flèche 12).

Le context handler est un convertisseur qui fait la conversion entre le format utilisé à l'extérieur du système et celui utilisé à l'intérieur. C'est un centre de relai pour le système.

2.3 Les implémentations XACML:

Plusieurs implémentations existent qui sont toutes conformes au standard XACML. Elles supportent tous les types de données et les fonctions standard de XACML. Nous citons ici les implémentations suivantes: SUNXACML [25], Java Enterprise XACML [26], HERASAF [27].

2.4 SUNXACML:

Il fournit une implémentation open source du standard XACML version 2.0 écrite en langage de programmation JAVA. Cette implémentation fournit un support pour la majorité des caractéristiques de XACML (analyse de politiques et de requêtes, détermination de l'applicabilité d'une politique, évaluation de requêtes auprès de politiques). En plus, une API (Application Programming Interfaces) Java permet de programmer des PEPs ou PIPs capables d'interagir avec le PDP, et d'ajouter de nouveaux types de données/fonctions au niveau du PDP. Par contre, écrire un PIP, afin de supporter des informations technologiques complémentaires (par exemple des informations dans une base de données), nécessite d'arrêter le système et modifier le code source. La modification est faite en utilisant l'API java pour ajouter des codes supplémentaires au PDP ou changer le code du PDP. SUNXACML s'adapte statiquement pour effectuer le changement du système venant des variations de l'environnement.

2.5 Java Enterprise XACML:

Est une implémentation Java du standard XACML version 2.0 indépendant des autres implémentations de XACML. Elle fournit des extensions avec des API Java qui permettent d'ajouter des fonctions non standards, et des PIPs pour récupérer des attributs sur des systèmes externes sans modifier la structure interne du PDP. Cette implémentation ne supporte pas l'ajout des nouveaux types de données pour répondre à des contextes donnés.

Par exemple, il est impossible d'implémenter les types des données définis dans GeoXACML, qui représentent une extension géo-spécifique à XACML version 2.0, de plus, comme SunXACML, il nécessite d'arrêter le système et modifier le code source du PDP (c.à.d. adaptation statique) pour chaque nouvelle fonctionnalité à ajouter.

2.6 HERASAF:

HERAS^{AF}

Il comprend un module de base ou PDP qui implémente le standard XACML et qui peut supporter des modules supplémentaires. Il permet d'ajouter dynamiquement des modules possédant des fichiers de configuration. Par exemple, un nouveau type de donnée non standard à spécifier dans une politique XACML, nécessite l'écriture d'un nouveau module et l'ajout dynamique au système, afin que cette extension soit prise en compte.

2.7 Structure d'une politique XACML

XACML s'inspire du principe de balisage qu'utilise XML. Ces balises représentent des éléments responsables de la structuration des politiques comme (PolicySet, Policy, Target, Rule....) nous expliquerons cela plus en détail dans le chapitre suivant.

Un élément peut contenir une série d'éléments. On peut représenter cela avec le diagramme de classes UML grâce à la notion d'agrégation.

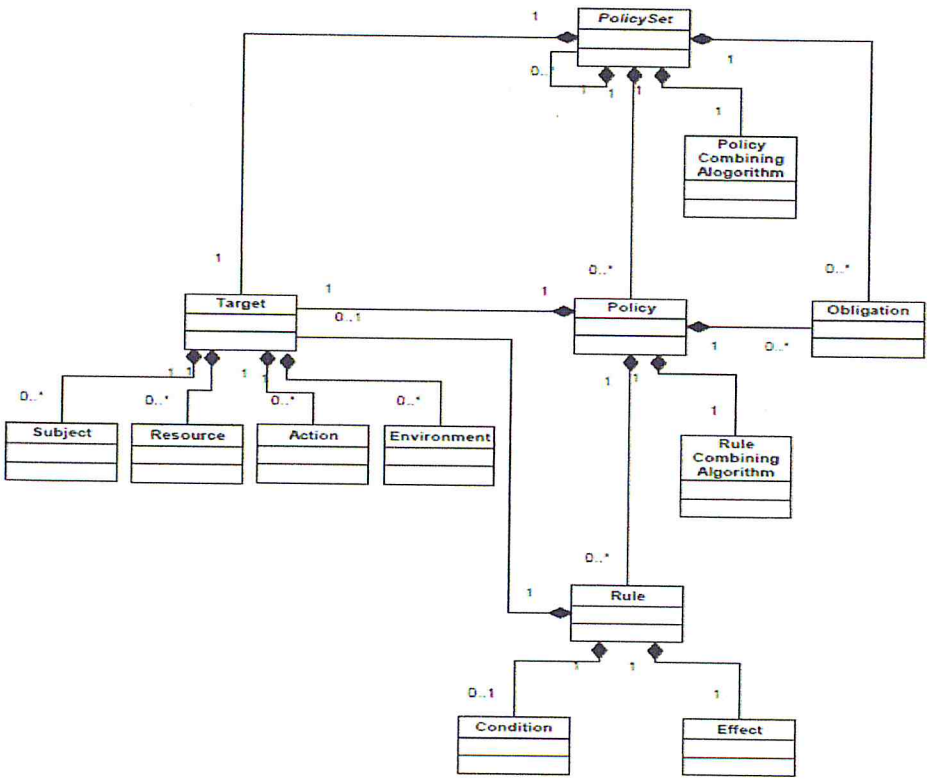


Figure 3.3 - Diagramme général d'une politique d'XACML[24].

3. Serveur d'authentification unifié et de gestion des droits:

Les serveurs d'authentification unifié et de gestion des droits sont des serveurs qui permettent de générer des politiques de contrôle d'accès et de les publier dans leur PDP grâce un une interface graphique spécifique à chaque moteur.

Ils sont développés pour être intégrés au sein de systèmes d'informations par leur administrateur afin de gérer le contrôle d'accès aux ressources.

Ils offrent une facilité et une rapidité d'exécution à leur utilisateur, en effet il suffit de développer un PEP qui se chargera du questionnement du PDP et d'intégrer un PIP qui permet de récupérer les données manquantes à la prise de décision pour le PDP.

Les plus connus sont Axiomatics, XACML space et WSO2 Identity Serveur. Suivant nos critères et nos objectifs on va s'intéresser juste au WSO2 IS.

3.1 WSO2 IS:

Le WSO2 Identity Server est un moteur de droit open source qui permet l'élaboration de politique de contrôle d'accès grâce à une interface graphique ou codifiée (requêtes XML).

Il permet à un utilisateur de s'authentifier auprès de plusieurs sites (devant prendre en charge cette technologie) sans avoir à retenir un identifiant pour chacun d'eux mais en utilisant à chaque fois un unique identifiant OpenID (qui est un système d'authentification décentralisé qui permet l'authentification unique, ainsi que le partage d'attributs) et XACML.

WSO2 Identity Server offre une sécurité sophistiquée et une gestion des identités, applications Web d'entreprise, des services, et des API, et rend la vie plus facile pour les développeurs et les architectes, avec ses exigences en matière de surveillance et de maintenance minimes sans tracas[28].

WSO2 Identity Server prend en charge le contrôle d'accès à base de règles avec XACML 2.0 et 3.0. Le moteur XACML a des actes de serveur d'identité en tant que PAP, PDP et un PIP. Le service de droit du serveur d'identité peut être exécuté par un PEP.

3.1.1 Les avantages de WSO2 IS:

- Il est Open source.
- Haute performance d'évaluation XACML avec mise en cache des techniques et d'indexation.
- Rapidité de communication entre PDP et PEP via l'épargne protocole.
- Fonctionnalité PDP exposait comme service Web sécurisé d'une manière standard. Par conséquent l'application Web peut être interprétée comme PEP.
- Connexion facile au PIP (trouve facilement les attributs dans une source de données).
- PDP soutien à une haute disponibilité et basculement à d'autres PDP.

3.1.2 WSO2 IS et notre projet:

Au tout début nous avons tentés d'implémenté WSO2 Identity Server entant que moteur de gestion des droits, d'abord pour sa liberté de modification et aussi la simplicité qu'il offre à ces utilisateurs en mettant en place une interface graphique.

Mais cela n'a pas aboutit a cause du fait que WSO2 IS n'est pas pensé pour un système aussi complexe que les RS, nous nous sommes donc inspiré du fonctionnement et du mécanisme de ce dernier pour développer un serveur de gestion des droits.

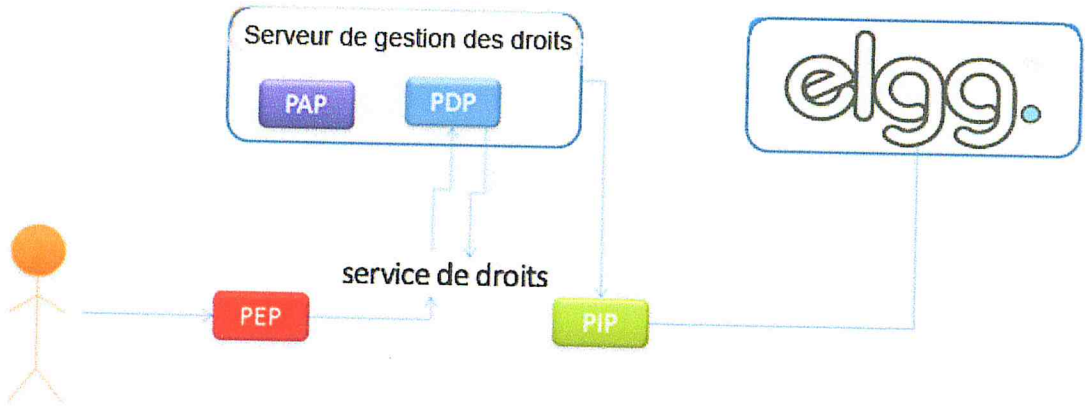


Figure 3.4 - *Architecture de la solution.*

Selon le schéma ci-dessus, la demande de l'utilisateur est envoyé en utilisant PEP (pour exécuter ce scénario, nous intégrerons le PEP dans Elgg).

La politique est écrite à l'aide du PAP de notre serveur de gestion des droits. Lorsque la demande arrive, le moteur de gestion des droits reçoit l'id de l'utilisateur du réseau social. Et regroupe grâce au PIP les informations manquantes au PDP pour la prise de décision.

4. Conclusion:

A l'aide des recherches effectuées pour réaliser ce chapitre nous avons trouvé les outils qu'on va utilisés pour développer notre propre modèle de contrôle d'accès dans un RS. Dans le chapitre suivant on va entamer la conception et présenter la solution proposée pour réaliser notre politique de contrôle d'accès.

Chapitre IV: Solution proposée et conception

1. Introduction:

Dans ce chapitre, nous allons d'abord définir les moyens utilisés pour élaborer un web service responsable de la création et gestion des politiques de contrôle d'accès, ainsi que le réseau social choisi pour montrer le fruit de notre travail.

Nous tacherons aussi d'expliquer comment se fait le passage du modèle ABAC a la spécification XACML

On va aussi utiliser le langage UML pour modéliser les différents diagrammes afin de représenter les interactions entre nos objets et schématiser le fonctionnement global de notre application.

2. Web Service:

Un service web (ou service de la toile[29]) est un programme informatique de la famille des technologies web permettant la communication et l'échange de données entre applications et systèmes hétérogènes dans des environnements distribués.

Il s'agit donc d'un ensemble de fonctionnalités exposées sur internet ou sur un intranet, par et pour des applications ou machines, sans intervention humaine, de manière synchrone ou asynchrone.

Le protocole de communication est défini dans le cadre de la norme SOAP dans la signature du service exposé (WSDL). Actuellement, le protocole de transport est essentiellement HTTP(S).

Le concept a été précisé et mis en œuvre dans le cadre de Web Services Activity[30], au W3C, particulièrement avec le protocole SOAP. Associé avec les Échanges de Données Informatisées (EDI), le consortium ebXML l'a utilisé pour automatiser des échanges entre entreprises. Cependant le concept s'enrichit avec l'approfondissement des notions de ressource et d'état, dans le cadre du modèle REST, et l'approfondissement de la notion de service, avec le modèle SOA.

2.1 SOAP:

Simple Object Access Protocol est un protocole de RPC orienté objet, bâti sur XML.

Il permet la transmission de messages entre objets distants, ce qui veut dire qu'il autorise un objet à invoquer des méthodes d'objets physiquement situés sur un autre serveur. Le transfert se fait le plus souvent à l'aide du protocole HTTP, mais peut également se faire par un autre protocole, comme SMTP.

Le protocole SOAP est composé de deux parties :

- Une enveloppe, contenant des informations sur le message lui-même afin de permettre son acheminement et son traitement.
- Un modèle de données, définissant le format du message, c'est-à-dire les informations à transmettre.

2.2 WSDL:

Web Services Description Language est une grammaire XML permettant de décrire un service web.

Il sert à décrire aussi:

- Le protocole de communication (SOAP RPC ou SOAP orienté message)
- Le format de messages requis pour communiquer avec ce service
- Les méthodes que le client peut invoquer
- La localisation du service.

2.3 Principe d'un web service:

Un service web, avec ses agents, fournisseurs et utilisateurs :

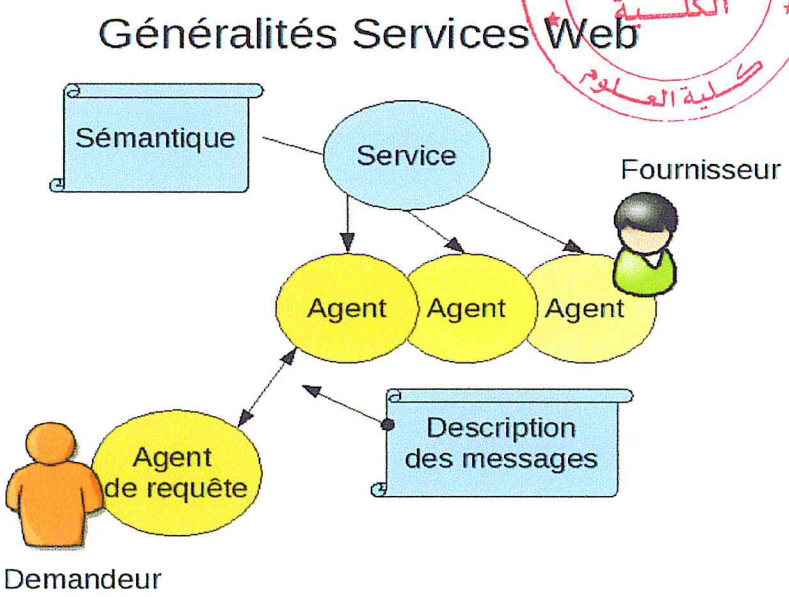


Figure 4.1 - *Très grandes généralités sur un service web[31].*

Dans sa présentation la plus générale, un service web se concrétise par un agent, réalisé selon une technologie informatique précise par un fournisseur du service. Un demandeur, à l'aide d'un agent de requête, utilise ce service.

Fournisseur et demandeur partagent une même sémantique du service web, tandis qu'agent et agent de requête partagent une même description du service pour coordonner les messages qu'ils échangent [37].

2.4 Technologies utilisées pour les web services:

Il existe plusieurs technologies derrière le terme services web:

- Les services web du type Representational state transfer (REST) exposent entièrement ces fonctionnalités comme un ensemble de ressources (URI) identifiables et accessibles par la syntaxe et la sémantique du protocole HTTP. Les Services Web du type REST sont donc basés sur l'architecture du web et ses standards de base : HTTP et URI.

- Les Services Web WS exposent ces mêmes fonctionnalités sous la forme de services exécutables à distance. Leurs spécifications reposent sur les standards SOAP et WSDL pour transformer les problématiques d'intégration héritées du monde Middleware en objectif d'interopérabilité.
- Les standards WS sont souvent décriés, comme risquant de générer une course à la performance technologique[32].

Toutefois leur robustesse dans le milieu des services entre professionnels, est reconnue, et ils restent largement utilisés. Aussi l'on préfère les faire évoluer[33].

3 Développement du Web Service:

Pour notre projet, nos exigences nous ont menés à choisir la solution des web services. On a alors développé cela à l'aide Apache Axis2, qui est un moteur de base de service web. C'est le développement et l'amélioration de son prédécesseur Apache Axis soap, il peut être implémenté en JAVA ou en C.

Grâce à Axis2, nous avons mis en place des services:

3.1 Le service "Contrôle":

Qui intègre un PDP pour la prise de décision et retourne les ressources dont l'accès est accordé au demandeur (le contacte du détenteur de la ressource).

Figure 4.2 - *WSDL Contrôle.*

3.2 Le service "Politique":

Qui permet la création de politique de sécurité dynamique pour les utilisateurs du RS, en générant un fichier XML propre à l'utilisateur et qui représente son profil, et après à chaque fois que l'utilisateur ajoute un poste, une règle de sécurité est ajoutée au fichier XML.

Figure 4.3 - *WSDL politique.*

Cette politique mise en place, il suffit juste de consommer ce service web à l'aide d'un client PHP (vu que le RS utilisé est développé en ce langage) avec la technologie SOAP mentionnée plus haut.

4. Règles de passages du ABAC a XACML:

4.1 Le langage de politiques:

XACML est utilisé pour décrire les exigences générales de contrôle d'accès en matière de contraintes sur des attributs. Un attribut peut-être n'importe quelle caractéristique 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'extensions standards pour définir de nouveaux types de données, des fonctions additionnelles, des combinaisons de logique, etc.

Partant de ce principe, nous avons doté nos sujets et ressources d'un attribut commun pour éviter toute surcharge du système, cet attribut est la catégorie (amis, amis proche, famille, collègue, public, privé) et l'attribut niveau de confiance pour les sujets, par contre nous avons limité les actions à LECTURE et ECRITURE, le fait d'ajouter ces attributs augmente le niveau de sécurité.

Cela s'exprime comme suit:

Un sujet X a un ensemble de contacts par ex {Catégorie(contact1)=Amis, Niveau de confiance(contact1)=3, Catégorie(contact2)=Famille, Niveau de confiance(contact2)=5 etc....}.

Ce même sujet a un ensemble de ressources par ex {Catégorie(ressource)=Amis, Catégorie(contact)=Collègue, etc.... }.

Pour qu'un de ses contacts puisse accéder en lecture par exemple à une ressource il faudra que la catégorie à laquelle il appartient soit la même que celle de la ressource, ou que son niveau de confiance soit suffisamment élevé pour accéder à la catégorie de la ressource; cela s'exprime avec une requête sous la forme:

Permettre si (Catégorie(contact)= Amis et type(ressource)= Photos et type(action)=LECTURE) ou

Permettre si (Catégorie(contact)= Amis et Niveau de confiance(contact) > 3 et type(ressource)=Video et type(action)=LECTURE)

Et tout cela représentera la politique de sécurité qu'adopte chaque utilisateur pour son profil, et qu'il faut exprimer en XACML.

4.2 Structure d'XACML:

À la racine de toute politique XACML, il y a une politique ou un ensemble de politiques. Un ensemble de politiques est une sorte de conteneur qui peut héberger d'autres politiques ou ensemble de politiques mais également des références vers des politiques situées sur des sites distants. Une politique est ici une politique de contrôle d'accès unique exprimée par un ensemble de règles. La figure suivante représente cela dans un schéma.

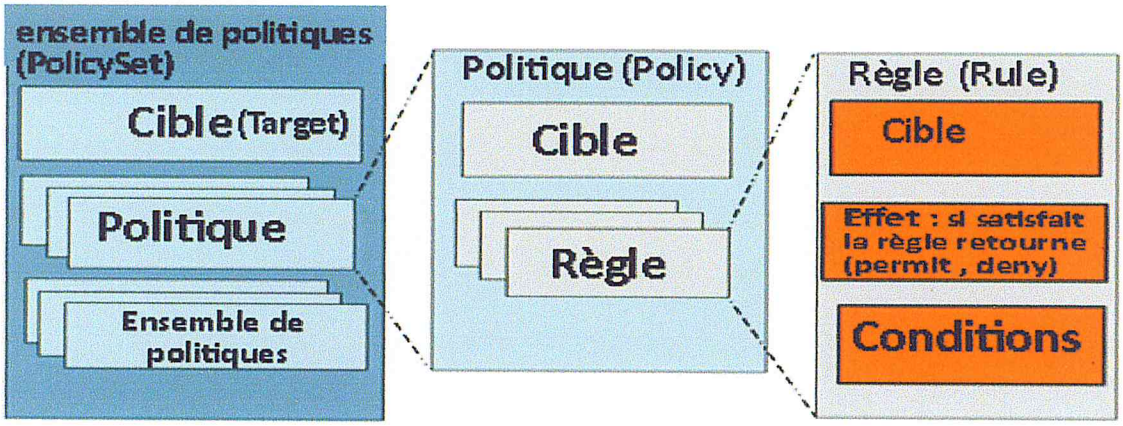


Figure 4.4 - *Structure d'XACML[24].*

4.2.1 Ensembles de politiques:

Soit PolicySet un ensemble d'ensemble de politiques. Chaque élément de PolicySet possède une cible, un algorithme de combinaison et une ou plusieurs politiques (ou alors un ou plusieurs ensembles de politiques).

Exemple:

Le profil d'un utilisateur => <**PolicySet** PolicySetId="Profil-mehdi">

4.2.2 Algorithmes de combinaison:

Un ensemble de politiques ou une politique elle-même peut contenir de multiples politiques ou règles, chacune d'entre elles pouvant amener à différentes décisions de contrôle d'accès, XACML propose des moyens de réconciliation des différentes décisions prises.

Il existe plusieurs algorithmes prédéfinis dans XACML mais nous avons choisi de n'implémenter que les deux les plus utilisés qui sont :

- PermitOverrides qui permet de retourner "permit" dans le cas où il y' aurait un conflit de règles.
- DenyOverrides qui permet de retourner "deny" dans le cas où il y' aurait un conflit de règles.

4.2.3 Politiques:

Soit Policy un ensemble de politiques. Chaque élément de Policy est défini par une cible, un ensemble de règles et un algorithme de combinaison des règles.

Exemple:

Politique de sécurité établie pour la famille => <**Policy** PolicyId="Famille">

4.2.4 Règles:

Soit Rule un ensemble de règles dans XACML. Une règle est définie par un effet, une cible et une condition.

Exemple:

Accorder l'accès à une ressource en lecture =>

```
<Rule RuleId="154" Effect="Permit">
<Resources>
        <Resource>
          <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#anyURI">122</AttributeValue>
            <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI" />
          </ResourceMatch>
        </Resource>
      </Resources>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
            <ActionAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string" />
          </ActionMatch>
        </Action>
<Rule>
```

4.2.5 Cible:

Afin de rendre plus efficace la recherche d'une règle qui doit s'appliquer à une requête donnée, XACML a introduit la notion de « cible » (target).

Une cible est un ensemble d'exigences simples sur le sujet, la ressource et l'action qui sont à considérer dans un ensemble de politiques, une politique ou une règle. Les cibles utilisent des fonctions booléennes simples qui sont rapides à évaluer. Par exemple, une cible peut restreindre la portée d'une politique de sécurité seulement au service FTP. Derrière ce concept, l'idée est de trouver très rapidement la règle de politique à évaluer dans le cas de politiques ou d'ensemble de politiques complexes.

Une fois que la requête correspond aux cibles d'un ensemble de politiques ainsi qu'aux cibles d'une des règles de l'une de ses politiques, la règle est évaluée.

Exemple:

Cibler une catégorie de contact =>

```
<Target>
    <Subjects>
     <Subject>
      <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
       <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Amis</AttributeValue>
       <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:attribute:role"
DataType="http://www.w3.org/2001/XMLSchema#string" />
      </SubjectMatch>
     </Subject>
    </Subjects>
  </Target>
```

4.2.6 Sujets, ressources et actions:

Les sujets, ressources et actions sont des éléments du langage XACML qui ont des propriétés identiques. En effet, chacun de ces éléments est défini par ses attributs ainsi que par leurs valeurs.

XACML met à disposition beaucoup plus d'éléments, mais dans le cadre de notre travail les éléments présentés suffisent largement à satisfaire nos objectifs.

Voici une portion du code XACML pour un profil d'un utilisateur:

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet PolicySetId="Politique-de-profile"
PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-
combining-algorithm:permit-overrides">
 </Target>
 <Policy PolicyId="Amis" RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combi-
ning-algorithm:permit-overrides">
  <Target>
   <Subjects>
    <Subject>
     <SubjectMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">Amis</AttributeValue>
      <SubjectAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:attribute:role"
DataType="http://www.w3.org/2001/XMLSchema#string" />
     </SubjectMatch>
    </Subject>
   </Subjects>
  </Target>
  <Rule Effect="Deny" RuleId="Deny-Rule" />
  <Rule RuleId="100" Effect="Permit">
   <Target>
    <Resources>
     <Resource>
      <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
       <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#anyURI">100</AttributeValue>
       <ResourceAttributeDesignator
AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
DataType="http://www.w3.org/2001/XMLSchema#anyURI" />
      </ResourceMatch>
     </Resource>
    </Resources>
    <Actions>
     <Action>
      <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
       <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
       <ActionAttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-
id" DataType="http://www.w3.org/2001/XMLSchema#string" />
      </ActionMatch>
     </Action>
    </Actions>
   </Target>
  </Rule>
 </Policy>
```

Figure 4.5 - *Politique d'un profil exprimer en XACML.*

4.2.7 Requête:

Soit "Request" un ensemble de requêtes dans XACML. Chaque requête est définie par un ou plusieurs sujets, une seule ressource et une seule action.

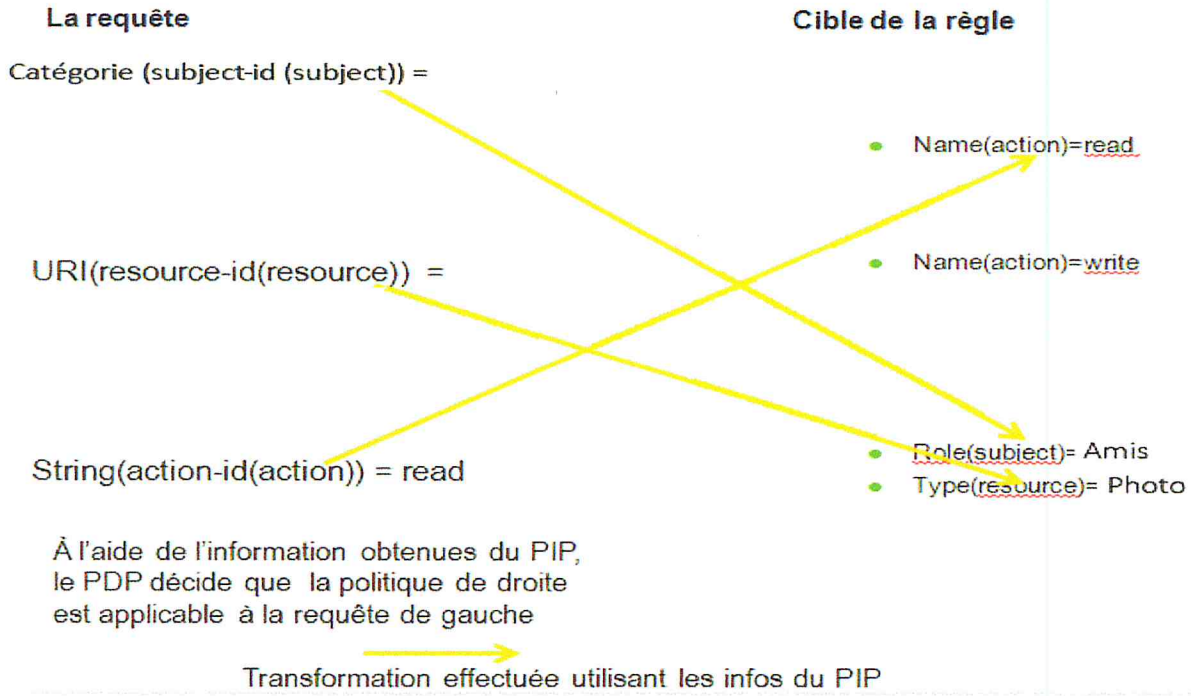


Figure 4.6 - *Exemple de requête.*

5. UML:

Le langage de modélisation unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçus pour fournir une méthode normalisée et visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet. L'UML est le résultat de la fusion de précédents langages de modélisation objet : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG).[34]

5.1 Les diagrammes:

La hiérarchie des diagrammes UML 2.0 sous forme d'un diagramme de classes:

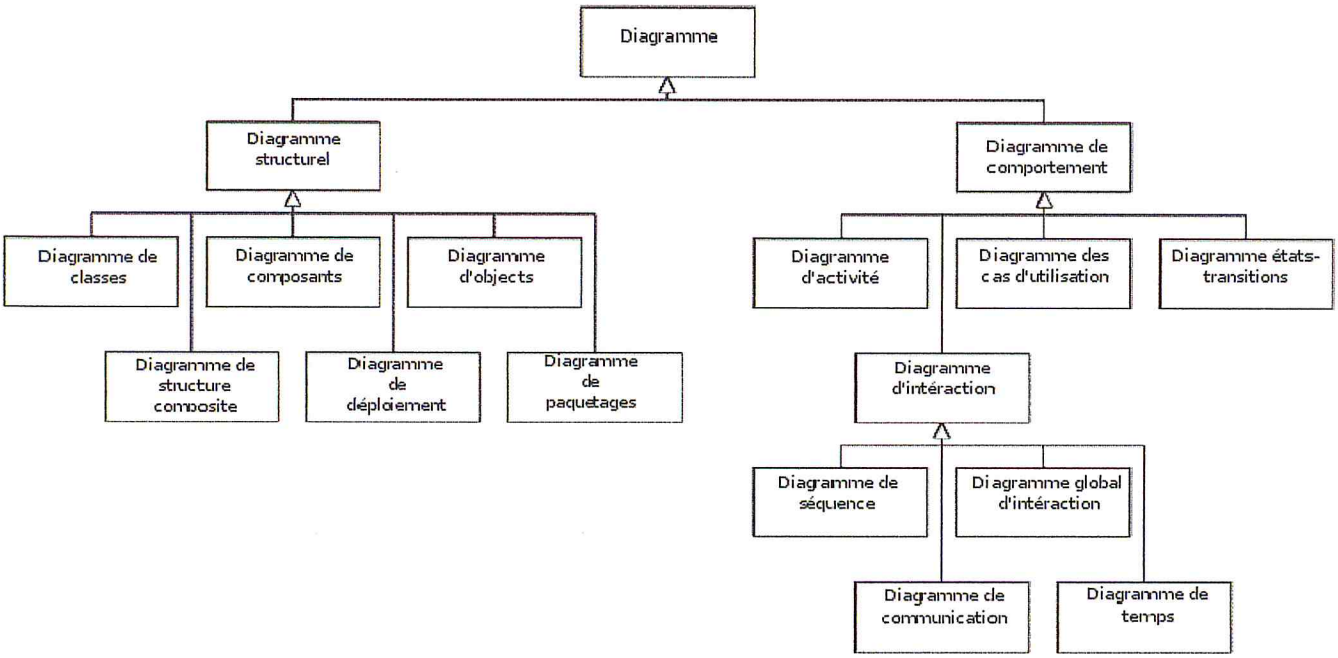


Figure 4.7 - *Hiérarchie des diagrammes UML 2.0.*

Les diagrammes UML sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie.

5.2 Diagrammes utilisés dans cette conception:

Diagramme des cas d'utilisation (use-cases ou Use Case Diagram): il permet d'identifier les possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire toutes les fonctionnalités que doit fournir le système. Le diagramme des cas d'utilisation est un diagramme comportemental.

Diagramme de séquence (Sequence Diagram): représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs. Le diagramme de séquence est un diagramme d'interaction ou dynamique.

5.3 Diagramme de cas d'utilisation:

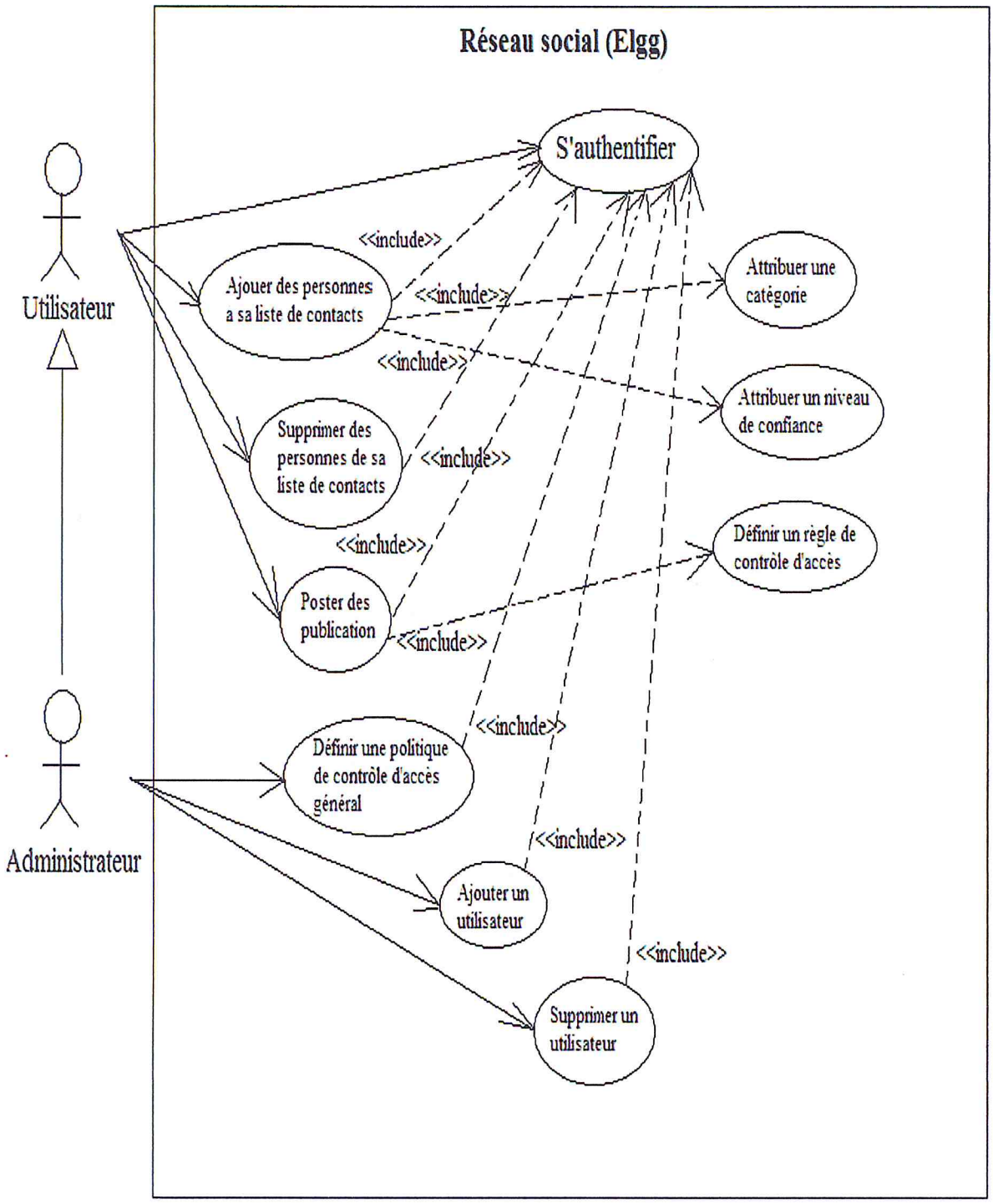


Figure 4.8 - *Diagramme de cas d'utilisation général.*

5.4 Diagrammes de séquence:

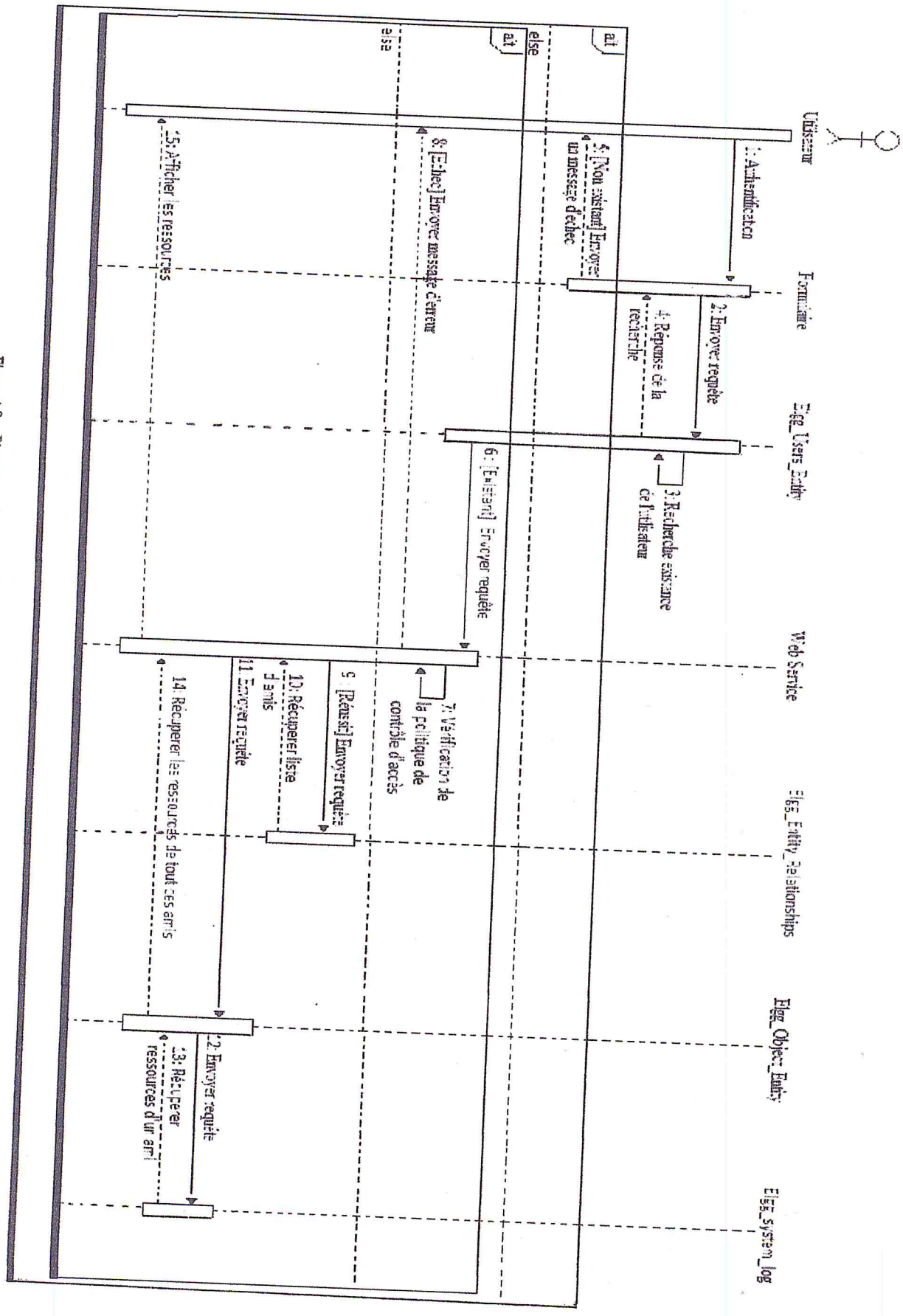


Figure 4.9 : Diagramme de séquence "Fil d'actualité"

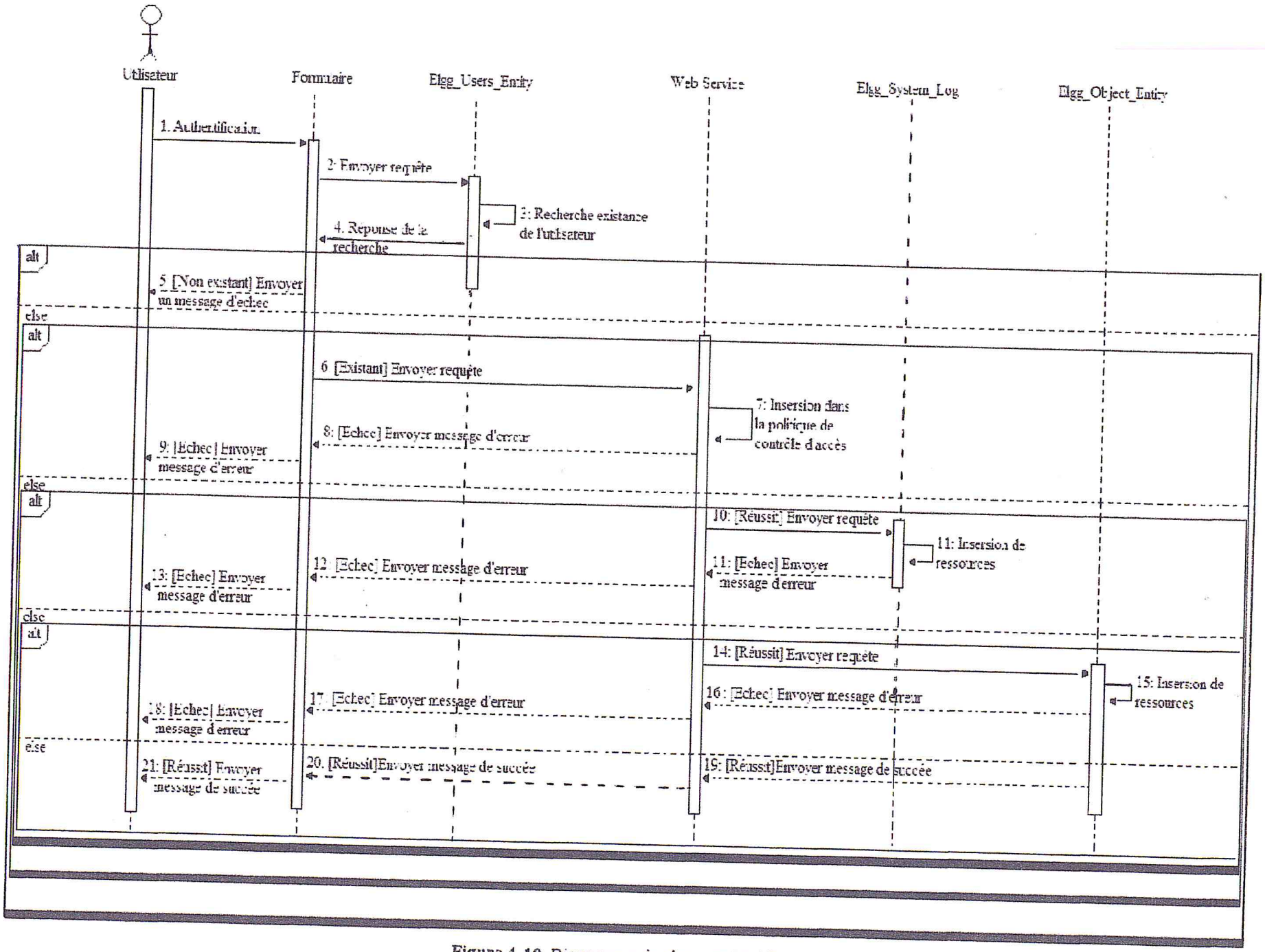


Figure 4.10 *Diagramme de séquence "Publication".*

6. Conclusion:

Ce chapitre représente le cœur de notre projet, il nous a permis de concevoir et schématiser notre travail et représenter les différents éléments du langage avec les quels nous avons développé notre politiques de contrôle d'accès. Il nous a permis aussi de concevoir le fonctionnement global de notre projet afin que tout le monde puisse le comprendre. Dans le chapitre suivant on présentera l'implémentation et les tests de notre application.

Chapitre V:

Implémentation

1. Introduction:

Dans ce chapitre on va présenter l'implémentation et les tests réalisés de notre web service sur un réseau social open source, on va mettre en pratique nos politiques de contrôle d'accès et présenter notre interface simplifiée pour que chaque utilisateur puisse facilement gérer ses paramètres de confidentialité et les régler à sa guise.

Tout cela va être représenté grâce à des captures d'écran et des observations sur les résultats obtenus.

2. Elgg:

Elgg est un framework Open source de développement rapide avec des fonctionnalités sociales intégrées. Il est utilisé pour la construction de n'importe quelle application où les utilisateurs se connectent et partagent l'information. Il permet de mettre en place un réseau social en ligne. Il fournit notamment des outils de blog, de microblogging, de partage de fichiers, de mise en réseau des profils d'utilisateurs, de gestion de groupes d'utilisateurs, d'agrégation de données et de nombreuses autres fonctionnalités. [35]

Il est utilisé pour construire toutes sortes d'applications sociales:

- Réseaux ouverts (similaires à Facebook)
- Topique (comme la Communauté Elgg)
- Intranets privés
- Datation
- Pédagogique
- Blog d'entreprise[36]

2.1 Composants d'Elgg:

Tout d'abord Elgg est developpé suivant le modèle MVC (Model View Controler) qui est un modèle destiné à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants au sein de leurs architectures respectives.

2.1.1 Les interfaces:

Elgg possède deux interfaces, une côté utilisateur qui est simple, clair et bien distribué, et une autre côté administrateur où on y accède à travers le petit lien "Administration", elle nous présente tout se qui ce passe dans le site, nous permet d'activer ou désactiver des Plug In, de créer des utilisateurs etc. C'est là où l'administrateur peut contrôler l'intégralité du site.

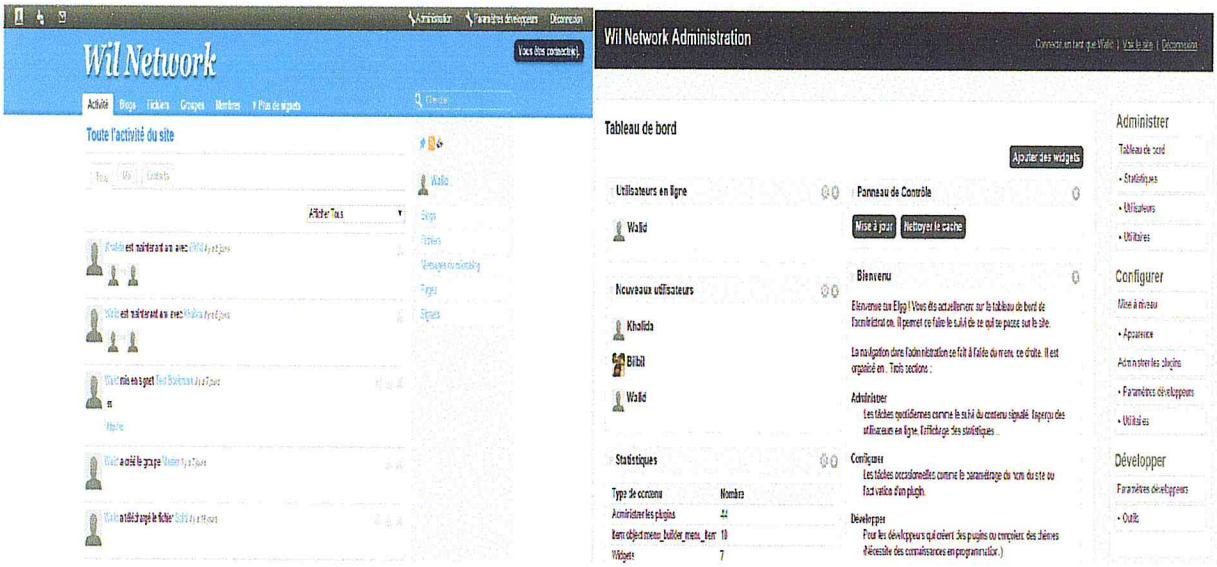


Figure 5.1 - *Interface d'Elgg.*

2.1.2 Base de donnée:

Elgg met à disposition une base de données, composée de 22 tables, tournant autour d'une table centrale qui fait la relation entre toutes les entités d'elgg (utilisateur, objet, groupe, etc...), cette table est le cœur du fonctionnement d'elgg; c'est la table elgg_system_log.

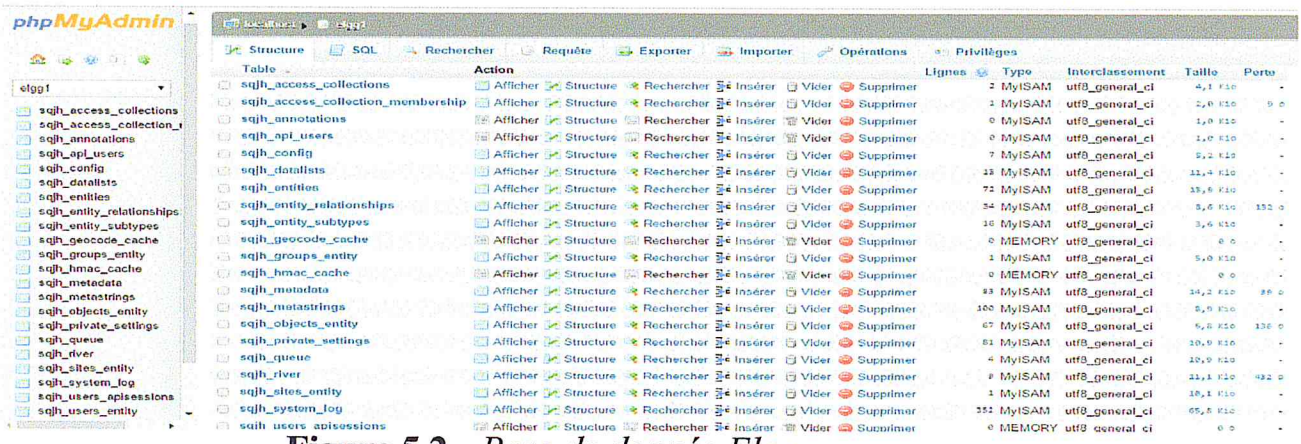


Figure 5.2 - *Base de donnée Elgg.*

2.1.3 Les Plugins:

En informatique, un plugin est un paquet qui complète un logiciel hôte pour lui apporter de nouvelles fonctionnalités.

Le terme plugin provient de la métaphore de la prise électrique standardisée et désigne une extension prévue des fonctionnalités, par opposition aux ajouts non prévus initialement apportés à l'aide de correctifs (patchs).[37]

La plupart du temps, ces programmes sont caractérisés de la façon suivante :

- ✓ Ils ne peuvent fonctionner seuls car ils sont uniquement destinés à apporter une fonctionnalité à un ou plusieurs logiciels.
- ✓ Ils sont mis au point par des personnes n'ayant pas nécessairement de relation avec les auteurs du logiciel principal.

Les plugin dans Elgg améliorent au site en ajoutant des fonctionnalités supplémentaires, des langues et des thèmes. Ils sont apportés par des membres de la communauté Elgg. Les plugins peuvent modifier le comportement du site ou ajouter de nouvelles fonctionnalités à ce dernier. [38]

2.1.4 Développement de notre plugin:

L'axe central de notre travail est d'améliorer la sécurité de la vie privée des utilisateurs, en créant un outil simple, et en offrant plus de choix à ces derniers, de mettre en place leur propre politique de contrôle d'accès. Le défi majeur qui se présente à nous est d'avoir un temps d'exécution optimal, car c'est le point de concurrence entre les différents réseaux sociaux.

Pour permettre cela nous avons donc choisi de créer un web service qui propose de prendre les décisions d'accès pour chaque utilisateur d'un RS, et qui soit flexible pour chaque RS. Cette approche nous a offert le résultat espéré, nous expliquerons cela plus en détail par la suite.

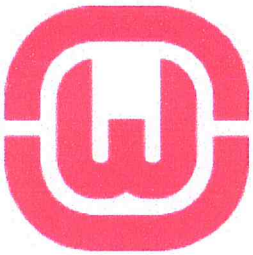
Maintenant que nous avons créé cet outil, il faut l'intégrer dans un RS pour tester son efficacité, c'est à ce moment que Elgg entre en scène. Mais pour éviter de modifier le noyau central du site, on a utilisé le principe de plugin que nous offre Elgg.

Ce dernier permet l'ajout de contacts et leur classement dans des catégories prédéfinies (**voir règle de passage page 52**), mais également l'ajout de ressources suivant toujours la logique des catégories.

3. Outil utilisé:

Pour réaliser ce projet nous avons utilisé un ensemble d'outils, de logiciels et de langages que nous allons présenter:

3.1 WAMP:



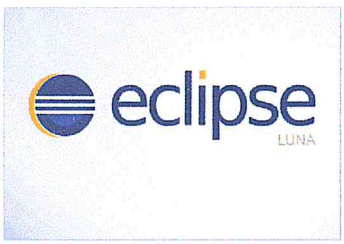
WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement nos bases de données.[39]

3.2 PHPMYADMIN:

phpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL.

Il s'agit de l'une des plus célèbres interfaces pour gérer une base de données MySQL sur un serveur PHP. De nombreux hébergeurs, qu'ils soient gratuits ou payants le proposent, ce qui permet à l'utilisateur de ne pas avoir à l'installer[40].

3.3 Eclipse:



Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de la Fondation Eclipse visant à développer un environnement de production de logiciels libres qui soient extensibles, universels et polyvalents, en s'appuyant principalement sur Java.

3.4 JAVA:

Le langage Java est un langage de programmation informatique 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), il a été présenté officiellement le 23 mai 1995 au SunWorld.

3.5 PHP:

PHP: Hypertext Preprocessor[41], plus connu sous son sigle PHP (acronyme récursif), est un langage de programmation libre principalement , utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale. PHP est un langage impératif orienté objet comme C++.

4. Implémentation de notre méthode:

Nous avons évoqué auparavant le développement d'un web service responsable du contrôle d'accès, nous allons maintenant donner plus de détail autour de cela.

Notre web service est développé en langage java, il se compose de trois classes principales:

4.1 La classe contrôle:

A l'aide de SUNXACML évoqué plus haut, cette classe représente le PDP, qui est l'axe central de notre travail. Grâce à elle les décisions d'accorder à un utilisateur l'accès (ou de l'en privé) sont prises.

```
import java.io.File;

public class Control  {
    static int tab[];
    public static PDP newPdp(String fichierxml) {
        FilePolicyModule policyModule = new FilePolicyModule();
        policyModule.addPolicy(fichierxml);
        CurrentEnvModule envModule = new CurrentEnvModule();
        PolicyFinder policyFinder = new PolicyFinder();
        Set policyModules = new HashSet();
        policyModules.add(policyModule);
        policyFinder.setModules(policyModules);

        AttributeFinder attrFinder = new AttributeFinder();
        List attrModules = new ArrayList();
        attrModules.add(envModule);
        //attrModules.add(mehdi);
        attrFinder.setModules(attrModules);
       // System.out.println(envModule);
        ResourceFinder resourceFinder = new ResourceFinder();
        PDP pdp = new PDP(new PDPConfig(attrFinder, policyFinder, resourceFinder));
          return pdp;
        }

    private static RequestType newRequest(String subjectId, String resourceId ,
            String actionId) {
          RequestType result = new RequestType();

          result.getSubject().add(newSubject(subjectId));
              result.getResource().add(newResource(resourceId));
              result.setAction(newAction(actionId));
              result.setEnvironment(newEnvironment());

              return result;
          }

```

Figure 5.3 - *Classe Contrôle.*

4.2 La Classe PIP:

Cette classe est en relation directe avec la base de données du RS (Elgg), pour permettre cela nous avons utilisé la bibliothèque java (mysql-connector-java-5.1.23-bin). Elle met à disposition un ensemble de classes et de fonctions pour gérer les requêtes SQL.

```
import java.sql.Connection;

public class Pip {
    public static HashSet<String> amis(int guid){
        HashSet set = new HashSet();
        HashSet set2 = new HashSet();//set.add(ami1);

    //

        try {


            String url = "jdbc:mysql://localhost/elgg";
            String user = "root";
            String passwd = "";
            Class.forName("com.mysql.jdbc.Driver");
            Connection conn = DriverManager.getConnection(url, user, passwd);
            Statement st = (Statement) conn.createStatement();
            ResultSet result = st.executeQuery("SELECT guid_two  FROM `elgg_entity_relationships` WHERE guid_one ="+guid );
            //On récupère les MetaData
            ResultSetMetaData resultMeta = (ResultSetMetaData) result.getMetaData();



            while(result.next()){
                set.add(result.getObject(1).toString());


            }

            result.close();
            st.close();
```

Figure 5.4 - *Classe PIP.*

4.3 La classe Politique:

Cette classe permet l'ajout et l'actualisation des politiques de contrôle d'accès des utilisateurs du RS.

Nous avons utilisé l'API DOM, qui permet de modéliser, de parcourir et de manipuler un document XML.

```
import java.io.*;

public class Politica
{

   static Element racine = new Element("PolicySet");


   //On crée un nouveau Document JDOM basé sur la racine que l'on vient de créer
   static org.jdom2.Document document = new Document(racine);
public static void initPolicyset(String NomF){
     Attribute PolicySetId = new Attribute("PolicySetId","ExamplePolicyset");
       Attribute PolicyCombiningAlgId = new Attribute("PolicyCombiningAlgId","urn:oasis:names:tc:xacml:1.0:policy-combining-algori
       racine.setAttribute(PolicySetId);
       racine.setAttribute(PolicyCombiningAlgId);
    //creation du policyset
     Element Target = new Element("Target");
    Element Actions = new Element("Actions");
    Element AnyAction = new Element("AnyAction");
    racine.addContent(Target);
   Target.addContent(Actions);
    Actions.addContent(AnyAction);
```

Figure 5.5 - *Classe politique.*

Elle stocke aussi tous les fichiers XML générer dans un fichier, permettant la réutilisation plus tard.

.settings	18/05/2015 20:53	Dossier de fichiers	
build	18/05/2015 20:44	Dossier de fichiers	
src	19/05/2015 11:54	Dossier de fichiers	
WebContent	19/05/2015 11:56	Dossier de fichiers	
.classpath	18/05/2015 20:54	Fichier CLASSPATH	2 Ko
.project	18/05/2015 20:44	Fichier PROJECT	2 Ko
29.xml	19/05/2015 15:31	Document XML	7 Ko
33.xml	20/05/2015 13:12	Document XML	15 Ko
34.xml	18/05/2015 21:09	Document XML	5 Ko
35.xml	18/05/2015 21:10	Document XML	4 Ko
36.xml	18/05/2015 22:21	Document XML	4 Ko
38.xml	19/05/2015 11:40	Document XML	4 Ko

Figure 5.6 - *Dossier contenant des fichiers XML générés.*

5. Test du web service:

Avant de pouvoir intégrer notre web service à Elgg, il faut d'abord vérifier s'il est opérationnel et voir s'il n'est pas gourmand en temps d'exécution.

Il existe une multitude d'outils pour tester les web services, mais nous avons jugé que SoapUI est le plus intéressant, notamment parce qu'il offre le temps d'exécution.

SoapUI est une application open source permettant le test de web service dans une architecture orientée services (SOA).

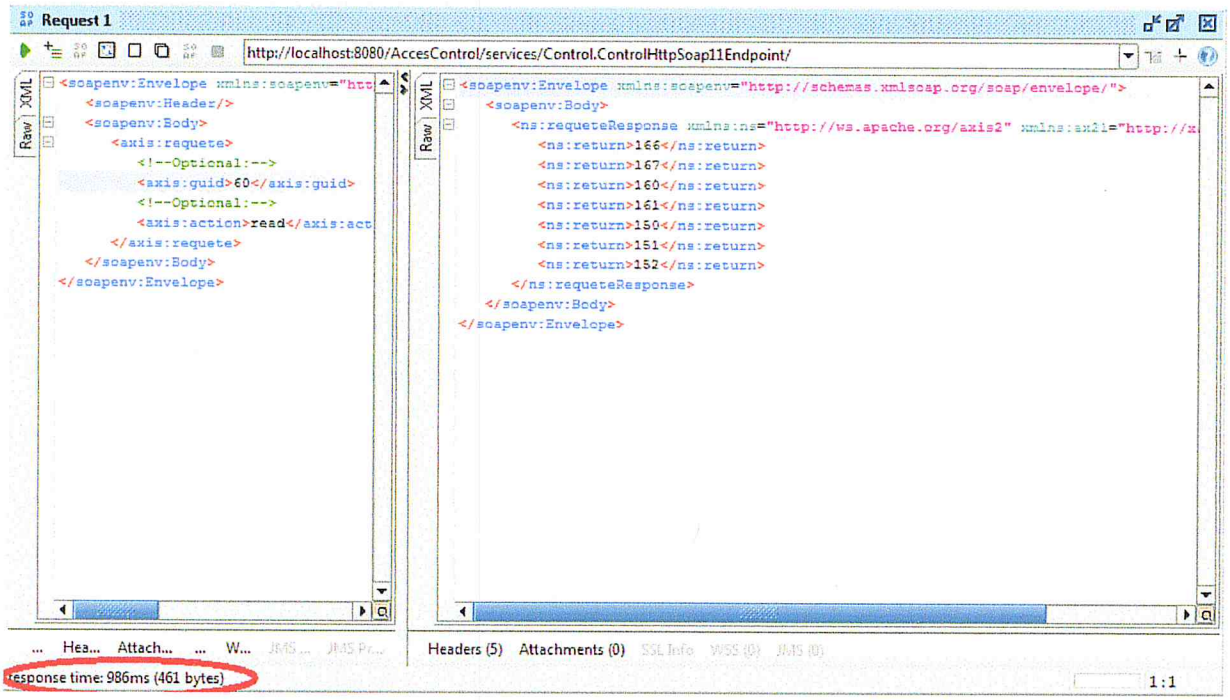


Figure 5.7 - *Test du Web Service grâce SoapUI.*

5.1 Utilisation du web service par Elgg:

Maintenant que le web service est fonctionnel, son utilisation via Elgg est plutôt simple. Car PHP permet la consommation de web service grâce à une bibliothèque (NuSoap) où il suffit d'intégrer dans son environnement de développement, en l'occurrence ici le Plug-in.

Dans un premier temps, il a fallu intégrer les attributs tels que "catégorie" et "niveau de confiance" dans le RS, la figure ci-dessous montre les quelques modification apportées à la l'ajout d'une personne .

Figure 5.8 - *Modification de "l'ajout d'une personne".*

Nous avons intégré les même attributs aux publications.

Figure 5.9 - *Modification de "Publication".*

On a aussi ajouté l'option "Check" qui permet de voir les amis qui ont la possibilité de voir notre publication.

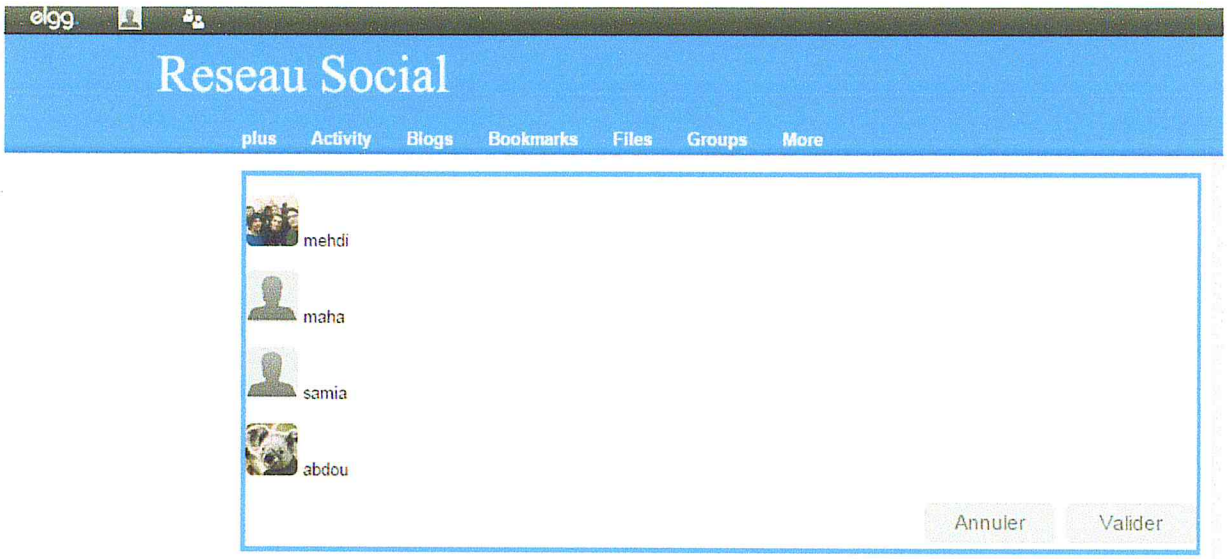


Figure 5.10 - *L'Option Check.*

Maintenant nous allons envoyer ces attributs au web service pour les ajouter à la politique de securité via la méthode "Politique".

```
$namespace="http://ws.apache.org/axis2";
$Client =  new  SoapClient ( "http://localhost:8080/AccesControl/services/Politica?wsdl" );
$param= array("guid"=>$guid,"res"=>$id,"act"=>$action,"cat"=>$_SESSION['cont']);
$price=$Client->call('Politique',$param,$namespace);
```

Figure 5.11 - *Code du client PHP pour Politica.*

Ceci nous a permis de créé un client PHP qui consomme le service politique ici.

Et maintenant nous allons passer à l'affichage des publications dans la file d'actualité.

Pour récupérer les publications nous allons utiliser le service "Contrôle", juste avec l'identifiant de l'utilisateur, nous aurons toutes les ressource de ces contacts où l'accès lui sera permis.

```
    $namespace="http://ws.apache.org/axis2";
$Client =  new  SoapClient ( "http://localhost:8080/AccesControl/services/Control?wsdl" );
$guid=elgg_get_logged_in_user_guid ();
$param= array("guid"=>"$guid","action"=>"read");

$post=$Client->call('requete',$param,$namespace);
```

Figure 5.12 - *Code du client PHP pour Control.*

Voici le résultat final:

Figure 5.13 - *Affichage final.*

6. Conclusion:

Ce chapitre nous a permis de présenter les différents tests et captures d'écran de nos politiques de contrôle d'accès et montrer les améliorations apportées par apport au réseau social open source de base, On a pu présenter les différentes interfaces et bases de données qui nous ont permis de réaliser ce projet ainsi que les différents outils et langages utilisés.

Conclusion générale:

Ce projet a été réalisé pour mieux expliquer l'effet des réseaux sociaux et leurs impact sur la vie privée des utilisateurs, la gestion de la confidentialité des données a été étudiée pour améliorer ces paramètres avec un minimum d'effort et une grande précision, dans ce cadre un web service a été développé pour effectuer le contrôle d'accès a l'aide d'apache Axis2, ce dernier a mené à la mise en place de deux services "contrôle" et "politique".

Les résultats obtenus après essai ont montré que le service politique génère correctement des politiques de contrôle d'accès dynamique comme un PAP, en revanche le service contrôle utilise ces politiques pour accorder l'accès, comme un PDP, le service web développé dans ce travail pourra être intégré dans les RS existants ou bien être l'outil pour une gestion de confidentialité pour un RS, ce dernier point reste donc un objectif à atteindre pour un projet à venir.

Beaucoup de chercheurs s'intéressent à la sécurité concernant la vie privée des utilisateurs des réseaux sociaux, on espère que notre modeste travail les aidera à trouver des réponses ou à leur éviter des erreurs.

On espère aussi que notre travail apportera une amélioration considérable aux utilisateurs des réseaux sociaux et qu'il sera bénéfique à ces derniers, et même si notre travail augmente le niveau de sécurité concernant la vie privée, l'utilisateur reste la première barrière contre la violation de sa vie privée, et maître de ces actes.

Bibliographie:

[1] Hage, "Web2.0, knowledge sharing and privacy in e-learning." Thèse présentée à la Faculté des arts et des sciences en vue de l'obtention du grade de Doctorat en Informatique. Soutenu le 03/03/2011.

[2] Frédéric Santos," L'origine des réseaux sociaux", http://www.memoclic.com/1593-reseau-social/15754-premier-reseau-social.html,18/02/2015.

[3] David Kirkpatrick," La révolution Facebook ", JC Lattès,2011

[4] Jun Pang and Yang Zhang, "A New Access Control Scheme for Facebook-style Social Networks", 2013.

[5] F.Makowski, "système d'information", http://www.marche-Public.fr/Terminologie/Entrees/systeme-information.htm,19/02/2015

[6] Romuald THION, "STRUCTURATION RELATIONNELLE DES POLITIQUES DE CONTRÔLE D'ACCÈS REPRÉSENTATION, RAISONNEMENT ET VÉRIFICATION LOGIQUES", page3 ,2008.

[7] Hayes b, "Cloud computing Communication of the ACM ", 51 (7), 9-11, 2008.

[8] Jun Pang and Yang Zhang, "A New Access Control Scheme for Facebook-style Social Networks", 2013.

[9] Xeni Jardin. US orders Twitter to hand over account data on Wikileaks and multiple Wikileaks supporters. January 7, 2011 http://bit.ly/1357EyV Accessed Oct. 27, 2014.

[10] Will Oremus. Facebook sued for \reading" your private messages. Jan. 3, 2014, http://slate.me/1evQXN1, Accessed Oct. 27, 2014

[11] Ero Balsa, Filipe Beato_,Seda Gürses ," Why Can't Online Social Networks Encrypt?", Belgium yNYU, USA , 2014.

[12] Barbara Carminati , Elena Ferrari , Andrea Perego, "Rule- Based Access Control for Social Networks", Proceedings of the 2006 international conference on On the Move to Meaningful Internet Systems: AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, Montpellier, France, October 29-November 03, 2006.

[13] A. Abou El Kalam, R. El Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte,A. Miège, C. Saurel, and G. Trouessin."Organization Based Access Control". In 8th IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2003), Lake Como, Italy, June 2003.

[14] Boustia N, Guesmia K, " A Contextual Access Control Model for Online Social Network", 2014.

[15] C. Landwehr. "Formal Models for Computer Security". ACM Computing Surveys (CSUR), 247 - 278. 1981.

[16] B. Lampson. Protection. in 5th Princeton Symposium on Information Sciences and,Systems, 1971.

[17] D. E. Bell et L. J. LaPadula, Secure computer systems : Uniffed exposition and multics interpretation. Technical Report ESD -TR - 73 -306.

[18] A.Abou El Kalam, R.El Baida, P.Balbiani, S.Benferhat, F.Cuppens, Y.Deswart, A.Miège, C.Saurel and G.Trouessin ;Organisation Based Access Control. IEEE 4th International Workshop on Policies for Distributed Systems and Networks (Policy 2003) ,Lake Come, Italy, Juin 4 - 6, 2003.

[19] D.F Ferraiolo, R. Sandhu, S. Gavrila, D.R. Kuhn, R. Chandramouli, Proposed NIST Standard for Role-Based Access Control, ACM Transactions on Information and System Security, vol. 4, août2001, p. 224-274.

[20] T.BELLAL, " Expression d'une politique de sécurité dans un réseau social", pages 12, France, Juin 2010.

[21] Frédéric Cuppens, Alexandre Miège ; Or-BAC, Organization Based Access Control, Journées Druide, Le Croisic, Mai 2004.

[22] LNIST Special Publication 800-162. Guide to Attribute Based Access Control, Definitions and Considerations, Final –Draft, Sept. 2013.

[23] Eric YUAN et Jin TONG. Attributed Based Access Control (ABAC) for Web Services . Dans Proceedings of the IEEE International Conference on Web Services, ICWS '05, pages 561–569, Washington, DC, USA, 2005. IEEE Computer Society.

[24] A ANDERSON," XACML Profile for Role Based Access Control (RBAC) ", OASIS Standard, 2004.

[25] SUN XACML PDP implementation, Site web : http://sunxacml.sourceforge.net/, 2011.

[26] Enterprise Java XACML: http://code.google.com/p/enterprise-java-xacml/, R. Kuhn, R. Sandhu, « RBAC Standard Rationale: comments on a Critique of the ANSI Standard on Role Based Access Control, » Dans IEEE Security & Privacy 2007, vol. 5, no. 6, pp. 51-53, 2007.

[27] HERASAF, site web: http://www.herasaf.org/, 2011.

[28] Dana Gardner (2010-06-02). "WSO2 tailors open-source middleware platform for cloud-based applications, deployment models". ZDNet. Retrieved 2011-08-29.

[29] JM.DOUDOUX, "Web Service en JAVA", http://www.jmdoudoux.fr/java/dej/chap-service-web.htm, 2014, 22/04/2015.

[30] [http://www.w3.org/2002/ws/](22/04/2015)

[31] [http://www.w3.org/TR/ws-arch/#whatis](24/04/2015)

[32] [http://www.w3.org/2007/01/wos-papers/lacey](25/04/2015)

[33] [http://www.w3.org/2005/Talks/1115-hh-k-ecows/#](25/04/2015)

[34] Grady Booch, James Rumbaugh, Ivar Jacobson (2000). Le guide de l'utilisateur UML (ISBN 2-212-09103-6)

[35] Cash Costello, Elgg 1.8 Social Networking, PACKT Publishing, coll. « PHP, Open source »

[36] [http://learn.elgg.org/en/1.11/](09/05/2015)

[37] [http://learn.elgg.org/en/latest/admin/plugins.html](11/05/2015)

[38] [https://community.elgg.org/plugins](09/05/2015)

[39] [http://www.wampserver.com/](11/05/2015)

[40] Marc Delisle, Gestion de bases de données avec phpMyAdmin, Campus Press, 2005 (ISBN 9782744019555)

[41] Luke Welling, Laura Thomson, PHP and MySQL Web development, Sams Publishing,(854795897OCLC ,6-32916-672-0ISBN) .e éd4 ,2008