Gestion des habilitations : modèles et architectures

Abdelmalek Benzekri, François Barrère, Romain Laborde

Gestion des habilitations : modèles et architectures

Abdelmalek Benzekri - François Barrère - Romain Laborde
IRIT/SIERA - Université Toulouse 3 Paul Sabatier

Introduction

La gestion des habilitations intègre la problématique de la gestion des identités et des accès (Identity & Access Management - IAM). Celle-ci est aujourd'hui considérée comme une application à part entière. Entre respect des obligations réglementaires et optimisation de l'administration des droits, les projets IAM renforcent le niveau de sécurité général tant sur les plans fonctionnel (ressources humaines) que technique. La multiplicité des applications métiers nécessitant chacune un contrôle d'accès propre et une administration des droits spécifiques a favorisé les exigences d'une vision globale et l'émergence de processus de gestion des accréditations bien identifiés [1]. La séparation claire des préoccupations et des problèmes de responsabilité a conduit à adopter un modèle organisationnel faisant apparaître différentes entités : fournisseur de service (SP), fournisseur d'identité (IdP), demandeur, et plateforme de gestion des identités.

Le fournisseur de service constitue le cœur de l'application métier. Lorsqu'un utilisateur sollicite un service particulier, il en fait la demande. Il doit pour cela se réclamer d'une identité dont la vérification exige la mise en place d'un service d'authentification. Cette vérification peut être réalisée systématiquement lors de chaque demande par l'application métier. Ainsi un utilisateur peut-il disposer de plusieurs identités au sein du même système d'information, ouvert ou non, et différentes solutions d'authentification peuvent coexister. Des technologies d'authentification unique (Single Sign-On) ont vu le jour permettant de limiter les interactions avec l'utilisateur fondées sur une relation de confiance entre les différents partenaires (IdP et SP). Des protocoles comme OpenID ou OAuth ont été conçus pour permettre à des plates-formes Web de déléguer la gestion de l'authentification en ligne ils permettent de récupérer des éléments d'authentification auprès d'un fournisseur d'identité. La plupart des plates-formes des réseaux sociaux – Facebook ou Twitter en particulier – y font appel.

Le concept de fédération d'identité puise également sa source dans les besoins de rationalisation des informations d'identité, d'interopérabilité et d'ouverture des systèmes d'information aux partenaires. On parle alors d'identité fédérée qui apporte des avantages fonctionnels aussi bien pour l'utilisateur que pour l'entreprise [2]. Le bénéfice pour les IdP chargés d'authentifier l'utilisateur et de gérer son identité, est de proposer un large éventail de services sans coût additionnel. Dans le même temps, un service d'autorisation pertinent est indispensable pour éviter aux SPs une perte de contrôle qui aboutirait à laisser des utilisateurs d'autres domaines entrer dans leurs systèmes d'information [3] [4].

La gestion des habilitations n'en est donc que plus cruciale. « Qui peut faire quoi, où, et quand ? » sont les questions que se posent désormais les administrateurs pour lesquels la protection des données personnelles et partagées est une exigence de plus en plus forte. Chaque ressource, fournie par une organisation, doit être protégée par des règles qui la rendent accessible aux seules entités ayant les accréditations

ABSTRACT

Access control is of major importance in nowadays information systems which are open, multi-domains and multi-suppliers. We address architectural and modelling issues of authorization systems allowing a clear separation of concerns between the requirements of the services to deploy and the access control management including the assurance of the identities of the subjects willing to access a given resource in a given environment.
From AAA solutions to the de facto XACML standard, the policy-based management model has been improved bringing a real and consistent approach to overcome the issues related to the interoperability of open identity and access management systems.

nécessaires. La gestion des accès est alors rendue par un service d'autorisation.

Dans cet article, nous nous intéressons aux infrastructures d'autorisation en termes d'architecture et de modèle. Une architecture d'autorisation doit complémenter le concept de fédération d'identité retenu pour l'échange d'informations de sécurité sur les utilisateurs entre les fournisseurs d'identité et les fournisseurs de services. Dans ce qui suit, nous allons considérer des architectures d'autorisation bâties autour d'un serveur assurant des fonctionnalités d'authentification, d'autorisation et de comptabilité (*AAA : Authentication Authorization Accounting*). Ce concept de serveur AAA permet de bien séparer les services d'authentification et d'autorisation d'un côté et les applications gérées de l'autre. Nous nous attarderons ensuite sur les Infrastructures de Gestion de Privilèges (IGPs) qui offrent elles aussi des moyens pour gérer les accès des utilisateurs aux ressources mises à leur disposition et ceci en considérant leurs privilèges (attributs ou propriétés assignés par une autorité). Ces dernières se révèlent comme une solution possible pour répondre aux besoins relevés en termes de politique de contrôle d'accès (qui a droit de faire quoi, comment et dans quelles circonstances). Enfin, nous présenterons XACML, aujourd'hui standard de fait, avant de conclure.

Les architectures d'autorisation AAA

Selon le RFC 2904 de l'IETF, les entités de base qui participent à une autorisation sont (figure 1) :

1) un utilisateur qui demande un service ;
2) l'organisation mère de l'utilisateur partie prenante au contrat établi et qui doit vérifier sous une forme active ou passive si l'utilisateur est habilité ou non à déclencher l'exécution du service ;
3) le serveur AAA du fournisseur de services qui autorise l'accès au service en se basant sur le contrat signé avec l'organisation mère de l'utilisateur ;
4) l'équipement de service dédié à la fourniture de services en réponse aux demandes de service.

Nous nous intéressons aux architectures mettant en œuvre la fonction d'autorisation par le biais du concept de politiques de contrôle d'accès (*PBMS : Policy Based Management Systems*). Les systèmes de gestion à base de politiques sont une solution de remplacement des listes de contrôle d'accès ou ACLs[1] intégrées habituellement aux applications gérées qui rend la gestion plus dynamique et évolutive.

Avec une telle approche, pour une autorisation efficace, deux actions doivent être réalisées :

- une décision d'autorisation doit être prise après consultation des politiques ;

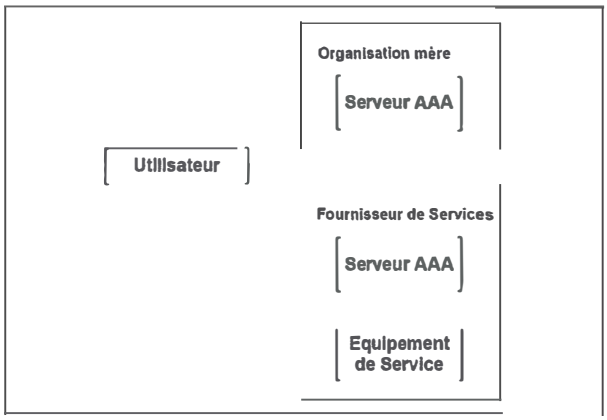

Figure 1 : Architecture d'autorisation AAA.

- la décision doit être appliquée.

Ces deux fonctions sont accomplies par deux entités distinctes nommées respectivement PDP (Policy Decision Point) et PEP (Policy Enforcement Point)

- Un PDP est une entité logique qui prend des décisions d'autorisation en considérant les informations suivantes (RFC 2906)
  - la ressource demandée et l'action requise (consultation, modification, etc.) ;
  - l'entité qui demande la ressource ;
  - la politique qui gère l'accès à la ressource.
- Un PEP est une entité logique qui applique la décision d'autorisation prise par le PDP. C'est le PEP, gardien de la ressource, qui réalise techniquement l'accès. Les interactions entre PDP et PEP peuvent suivre l'un des trois modèles suivants : Agent, Push ou Pull.

Le Modèle Agent

Le modèle Agent permet à l'utilisateur d'adresser sa requête de demande de ressource à une partie tierce (le serveur d'autorisation : PDP/PEP). Ce dernier joue le rôle d'agent entre l'utilisateur et l'équipement fournissant le service. Le serveur d'autorisation applique la politique associée à cette demande. Si l'accès est autorisé, le serveur transfère la requête de l'utilisateur à la ressource ; sinon, il renvoie un message d'interdiction à l'utilisateur. La ressource retourne le résultat de la requête au PEP qui à son tour le transfère à l'utilisateur. Dans ce cas là, le PEP se trouve au niveau du serveur d'autorisation (figure 2).

Le Modèle Push

L'utilisateur adresse sa requête directement au fournisseur de service (i.e. la ressource demandée) et en particulier, au serveur d'autorisation (PDP) qui gère l'accès à la ressource. Ce dernier définit l'information d'autorisation (le droit d'accès ou pas à la ressource) qui est renvoyée à l'utilisateur sous forme d'un ticket. L'utilisateur présente le ticket acquis avec la

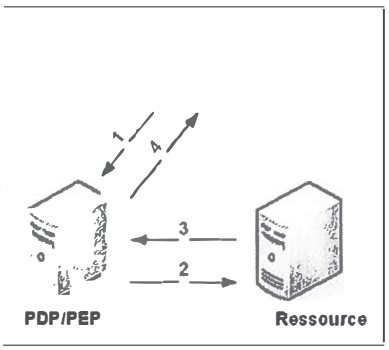

Figure 2 : Modèle Agent.

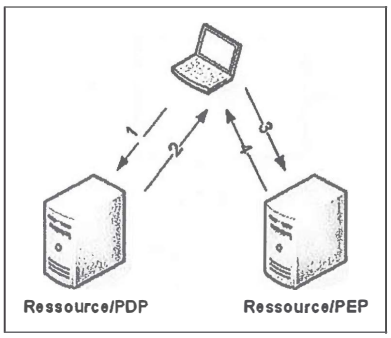

Figure 3 : Modèle Push.

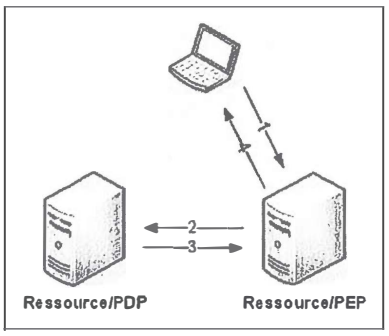

Figure 4 : Modèle Pull.

demande d'accès à la ressource. Ainsi, le PEP qui se trouve au niveau de la ressource cette fois-ci donne le droit d'accès à l'utilisateur si le ticket est valide (figure 3).

Le Modèle Pull

Le modèle pull laisse la responsabilité de la récupération de l'information d'autorisation au PDP seul. Une fois que l'utilisateur a demandé l'accès à une ressource, le PEP qui se trouve au niveau de cette dernière, récupère l'information d'autorisation d'une façon active auprès du PDP et donne le droit d'accès à l'utilisateur si la décision est affirmative (figure 4).

Le modèle Agent convient mieux dans le cas de fédération d'organisations membres s'appuyant sur une entité tierce pour la gestion des accès aux ressources partagées comme dans le cas d'une entreprise étendue avec une autorité centrale.

Les modèles Push et Pull sont également adaptés aux entreprises étendues et aux campus numériques où chacun des membres garde le contrôle sur ses propres ressources et à qui il revient d'autoriser les utilisateurs d'un partenaire d'accéder les ressources partagées. Préférer un modèle à un autre est une question de choix stratégique et n'a rien à voir avec la performance ou la capacité des modèles.

Dans la recommandation X.509, l'UIT définit des Infrastructures de Gestion de Privilèges (IGPs) (*PMI : Privilege Management Infrastructure*) qui mettent en œuvre une approche fondée sur des politiques. Elles permettent de gérer (affectation, vérification, validation, etc.) des attributs des entités et ainsi offrent un moyen d'autoriser l'accès des utilisateurs en fonction de leurs privilèges et non plus de leurs seules identités. Il reste à noter que les Infrastructures de Gestion de Privilèges (IGPs) peuvent adopter n'importe quel modèle de gestion parmi les trois définis ci-dessus.

Les Infrastructures de Gestion de Privilèges

L'ISO définit une IGP comme étant *"the infrastructure able to support the management of privileges in support of a comprehensive authorization service and in relationship with a Public Key Infrastructure"*. L'IETF la définit comme étant *"the set of hardware, software, people, policies and procedures needed to create, manage, store, distribute, and revoke Attribute Certificates (ACs)"*.

Une IGP fournit donc un cadre d'autorisation basé sur les attributs des utilisateurs pour gérer les accès aux ressources et aux services. Pour cela, la norme X.509, qui est surtout connue pour sa spécification des certificats d'identités et de l'infrastructure de gestion de ces certificats [5], a formalisé à partir de sa version 4 la notion de certificats d'attributs afin de fournir un document exportable garantissant qu'un utilisateur possède des accréditations/privilèges. L'utilisation de certificats d'attributs rend le service d'autorisation plus flexible. Ainsi une IGP permet l'allocation, la délégation, la révocation et le retrait des privilèges ou droits des utilisateurs d'une façon électronique.

Les certificats d'attributs établissent un lien fort entre une identité d'un utilisateur et un attribut (un rôle, un groupe, un privilège). Un certificat d'attributs contient donc un ensemble d'attributs qui donnent des informations sur les privilèges du possesseur du certificat [6].

Le certificat d'attribut est signé par une Autorité d'Attribut (AA) et concerne en particulier l'autorisation (ex. rôle, appartenance à un groupe) et non pas l'identification comme le certificat d'identité.

Le niveau de confiance que nous pouvons avoir dans le certificat d'attribut dépend de la confiance que nous avons dans l'IGP qui a généré le certificat.

Les entités qui composent une IGP sont :

- L'Autorité d'Attribut (AA) responsable de l'émission des certificats d'attributs. Des annuaires de certificats d'attributs X.509 ainsi que des listes de révocation de certificats d'attributs sont utilisés pour la publication et révocation des certificats. La distribution des certificats d'attribut X.509 suit le modèle "push and pull" ;
- L'origine de l'autorité (*SOA : Source of Authority*) responsable des accès à une ressource. Toutes les demandes d'accès à une ressource doivent prouver que leurs privilèges

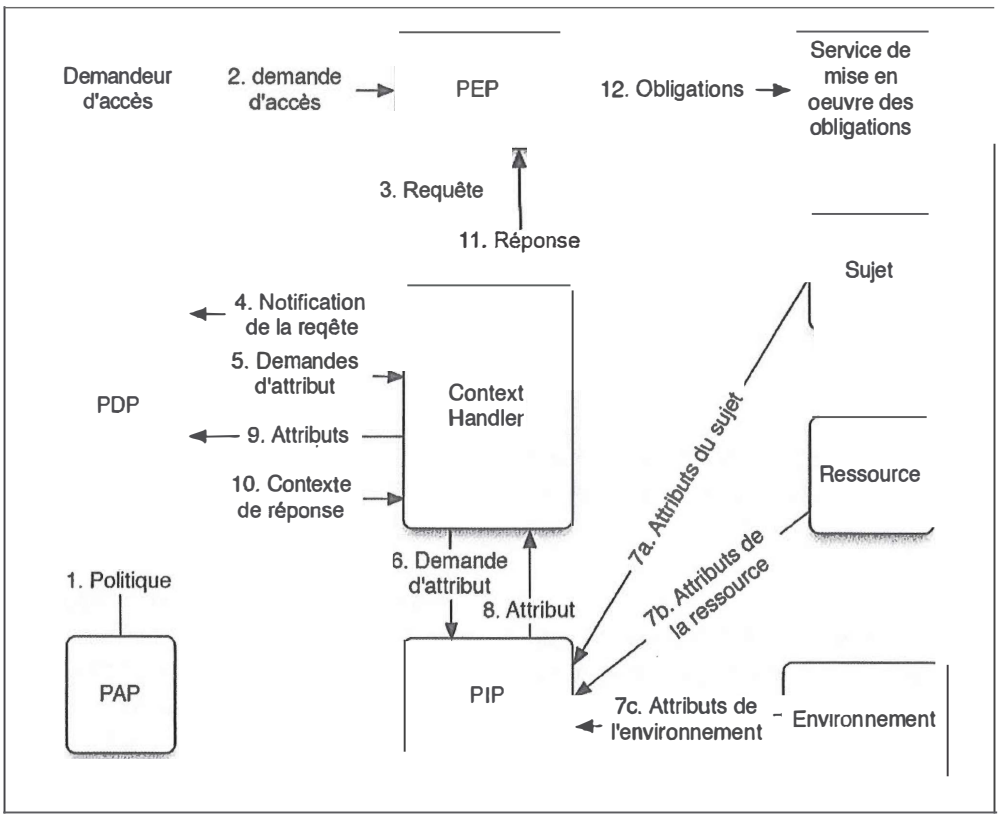

Figure 5 : Architecture XACML.

découlent de la SOA qui contrôle la ressource. Une SOA est l'équivalent d'une autorité de certification racine dans une infrastructure de gestion de confiance (IGC).

Dans la littérature, nous retrouvons un certain nombre d'Infrastructures de Gestion de Privilèges qui peuvent être basées ou non sur le standard X.509 [7], en particulier pouvant être couplées à XACML défini par l'OASIS[2].

Modèles de contrôle d'accès

Les politiques de contrôle d'accès sont des exigences spécifiant comment les accès sont gérés et qui, dans quelles circonstances, peut accéder à quelle information. Afin d'exprimer des politiques de contrôle d'accès qui soient cohérentes et empêcher des brèches de sécurité dans les systèmes protégés, des modèles de contrôle d'accès sont utilisés pour l'expression de ces politiques. Ces modèles permettent une représentation formelle des politiques.

Nous renvoyons le lecteur à l'article d'Alban Gabillon sur le contrôle d'accès aux données numériques publié dans ce même dossier.

XACML

Le standard XACML[3] est une spécification XML édictée par OASIS pour la définition de politiques de contrôle d'accès. XACML v3 fournit un langage universel de description des politiques de contrôle d'accès de la forme : qui peut faire quoi et à quel moment ? De plus, ce langage s'appuie sur une architecture pour la mise en œuvre du contrôle d'accès : un protocole de type requête/réponse similaire à celui défini par SAML[4] donne les moyens d'exprimer des requêtes d'accès et les réponses appropriées. L'un de ses avantages est de favoriser l'interopérabilité entre les produits d'administration et d'autorisation hétérogènes présents sur le marché.

La politique de contrôle d'accès permet de définir les droits des utilisateurs (personne ou application) sur les ressources informatiques (données, services, etc.). XACML est un langage d'expression puissant qui utilise la logique pour combiner les règles où toute information de sécurité est considérée comme un attribut du sujet, de la ressource, de l'action ou encore de l'environnement. Il permet ainsi d'exprimer des politiques contextuelles nécessitant des autorisations dyna-

[2] OASIS (Organization for the Advancement of Structured Information Standards) est un consortium à but non lucratif qui produit des standards dans les domaines de la sécurité, de l'informatique des nuages, des architectures orientées services, des services web, etc.

[3] eXtensible Access control Markup Language : www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

[4] Security Assertion Markup Language : OASIS Security Services (SAML) TC www.oasis-open.org/committees/tc_home.php?wg_abbrev=security.

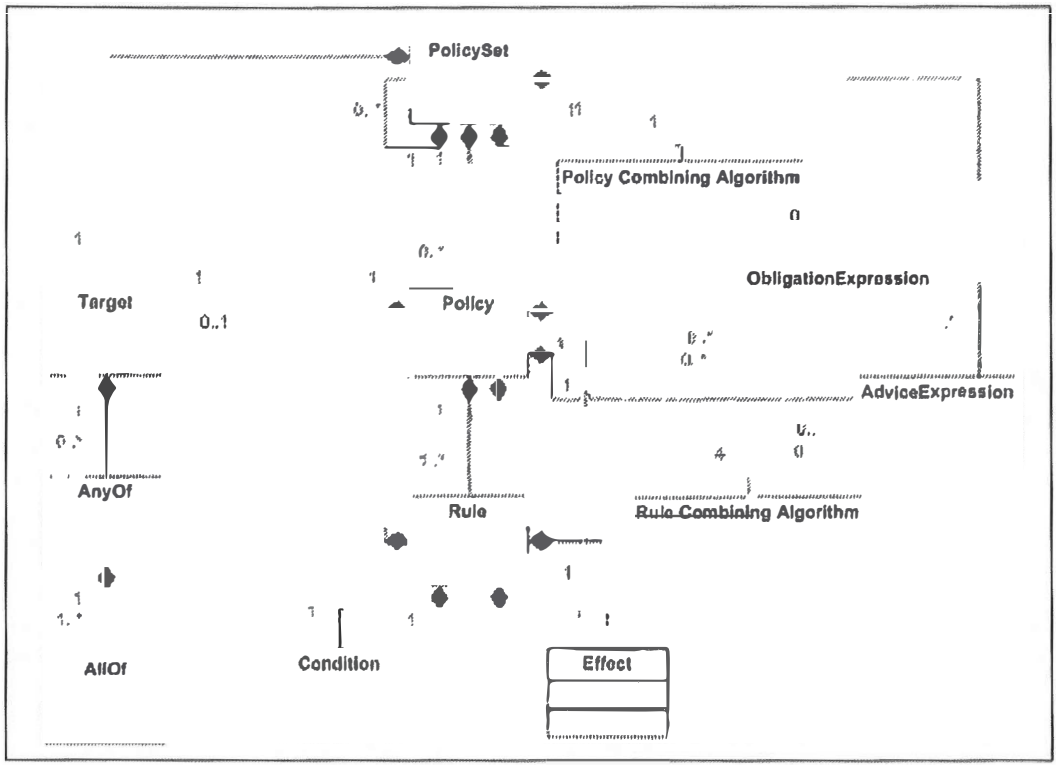

Figure 6 : Modèle du langage de politique XACML v3.

miques. Un exemple d'autorisations dynamiques contrôlées par un moteur de workflow a été présenté dans [8].

L'architecture XACML est représentée figure 5.

Le modèle opère selon les étapes suivantes

1. Le *Policy Administration Point* (PAP) écrit les politiques et les rend disponibles au PDP. Ces politiques représentent la politique complète qui contrôle les décisions prises par le PDP ;
2. Le demandeur d'accès envoie une requête d'accès au PEP ;
3. Le PEP envoie la requête d'accès au *context handler* au format natif (langage supporté par le PEP), optionnellement incluant des attributs pour le sujet, la ressource et l'environnement ;
4. Le *context handler* construit un contexte de requête XACML et l'envoie au PDP.
5. Le PDP peut demander des attributs additionnels pour le sujet, la ressource et l'environnement du *context handler* ;
6. Le *context handler* demande ces attributs du *Policy Information Point* (PIP) ;
7) Le PIP obtient les attributs demandés qui peuvent être stockés dans différentes bases de données, annuaires, etc. ;
8) Le PIP retourne les attributs demandés au *context handler* ;
9) Le *context handler* envoie les attributs demandés au PDP qui évalue la politique.
10) Le PDP retourne le contexte de réponse XACML (incluant la décision d'autorisation) au *context handler* ;
11) Le *context handler* traduit le contexte de réponse XACML au format de réponse natif du PEP. Le *context handler* retourne la réponse au PEP qui applique la décision d'autorisation ;
12) Lorsque la décision inclut des obligations (par exemple, « si la décision est « permit » alors envoyer un courriel à l'administrateur »), le PEP met alors en œuvre ces obligations via un service dédié.

La figure 5 montre un élément particulièrement intéressant dans le cas de réseaux distribués collaboratifs où les entités PEPs ne supportent pas nécessairement le langage XACML. Le *context handler* est l'entité du système qui convertit les demandes de décision d'autorisation exprimées dans le format natif du PEP (langage compris par l'application gérée par le PEP) en une forme canonique XACML (on parle de contexte XACML), et convertit les décisions d'autorisation de la forme canonique XACML en une réponse dans le format natif du PEP.

XACML est adapté aux environnements distribués où les PEPs et PDPs sont répartis au niveau d'infrastructures hétérogènes. En séparant les politiques d'accès des applications protégées et en proposant un standard pour l'expression des autorisations, XACML permet aux systèmes de sécurité hétérogènes de partager les politiques. Ainsi, les administrateurs de systèmes ne sont plus obligés d'écrire leurs politiques en utilisant différents langages. Ajouté à cela, des travaux tentent de rendre les implémentations de PDP et PEP modulaires

afin d'améliorer leur réutilisabilité et pouvoir ajouter de nouvelles fonctionnalités à la volée [9] [10] [11].

Une politique XACML est composée d'une cible *Target*, d'un ensemble de règles *Rules* et d'un ensemble facultatif d'éléments d'*Obligations* qui s'appliquent à la requête (figure 6). L'élément *Target* permet d'identifier la politique ou les règles applicables à une requête d'accès ; il spécifie les conditions que le sujet, la ressource et l'action doivent vérifier afin qu'une politique ou une règle soit applicable à la ressource requise. Ainsi, l'élément *Target* fournit un moyen d'indexation et de recherche de politiques. Une fois que la politique applicable est identifiée, la prochaine étape est d'évaluer ses règles.

XACML est un standard incontournable. Les éditeurs IAM l'ont adopté soit en natif soit dans les solutions de gestion des politiques de sécurité [12].

En utilisant des modèles de contrôle d'accès existants, la gestion de contrôle d'accès se simplifie. En effet, OASIS a amélioré le langage de politiques XACML afin de supporter le modèle RBAC[5] et de gérer les conflits dans la définition de ces politiques. OASIS a défini un profil RBAC pour XACML illustrant comment implémenter des politiques RBAC tout en se servant du langage de politiques XACML.

XACML fournit le moyen d'exprimer des ***obligations*** ainsi que des ***avis*** au niveau des règles à considérer par un PEP. On peut noter aussi que la puissance de XACML provient de son caractère extensible par l'ajout 1) de nouveaux attributs (sujet, action, ressource ou environnement), 2) de nouvelles fonctions pour manipuler les attributs, 3) ou encore de nouveaux algorithmes de combinaison permettant d'arrêter une décision tant au niveau des règles que des politiques. Une signature numérique assure l'intégrité des déclarations XACML. Cela étant, un PDP ne doit pas demander qui a signé une politique XACML ou si celle-ci est signée. Par contre, « ***le PDP doit s'assurer que la clé ayant signé une politique est sous le contrôle de l'entité émettrice de la politique*** »[6].

Exemples de systèmes de gestion d'autorisations réseaux

La combinaison des architectures à base de politiques et d'une gestion des habilitations offrant un niveau de flexibilité et d'intégration important, a été mise en œuvre dans de nombreux exemples de gestion des autorisations réseaux. Lopez et al. [13] ont montré comment intégrer un système d'autorisation de type XACML à un point d'accès réseau de type 802.1X ou utilisant PANA[7] afin de contrôler l'accès des utilisateurs au réseau. Les habilitations/attributs des utilisateurs sont échangés dans des déclarations SAML. Les déclarations SAML sont encapsulées dans les protocoles AAA DIAMETER ou EAP qui ont été étendus pour l'occasion. Enfin ils présentent l'intégration du système d'autorisation XACML selon les modèles push et pull et dans des situations intra et inter domaines. Laborde et al. [8] ont présenté une solution basée sur XACML et les fédérations d'identités pour gérer les habilitations et les autorisations des utilisateurs à un serveur web dans le contexte des organisations virtuelles. La gestion des autorisations et des habilitations y est distribuée entre différentes organisations et les politiques XACML expriment des autorisations dynamiques où les utilisateurs n'ont pas les mêmes droits selon le travail en cours défini par le workflow. Demchenko et al. [14] ont défini un profil XACML, appelé XACML-NRP, pour des besoins de Qualité de Service et en particulier l'allocation de ressource à la demande dans les réseaux. Ce profil permet de gérer en même temps les autorisations des utilisateurs ainsi que la qualité de service qui leur sera fournie. L'expression des besoins de qualité de service se fait via des obligations.

Enfin, de nombreux chercheurs proposent aujourd'hui de gérer les autorisations d'accès aux grilles informatiques ou aux nuages informatiques via des systèmes XACML. Par exemple, l'Open Grid Forum (OGF) a défini un protocole permettant à un PEP sur une grille de données d'obtenir des décisions de contrôle d'accès incluant des obligations [15]. L'OGF a aussi proposé un profil XACML pour la gestion des autorisations dans les grilles [16]. Wiebelitz et al. [17] ont implémenté un pare-feu pour grilles basé sur XACML. Enfin, Canh et al. [18] décrivent une architecture de type XACML pour gérer l'allocation de services à la demande dans un nuage informatique.

Conclusion

Le contrôle d'accès est une préoccupation constante surtout dans un contexte de systèmes d'information multi-domaines et multi-fournisseurs. En témoignent les solutions d'authentification et d'autorisation en ligne de plus en plus déployées autour des plates-formes de réseaux sociaux ou de cloud computing. Dans cet article, nous nous sommes intéressés aux architectures et modèles de gestion des autorisations. Des solutions AAA à XACML, les architectures favorisent la séparation claire des exigences des processus de gestion des identités de celles des services à fournir. Les entités induites que sont les fournisseurs d'identités, les fournisseurs de services et les plates-formes de gestion des identités connaissent des spécialisations métiers nouvelles.

[5] Rule Based Access Control.

[6] www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

[7] Protocol for Carrying Authentication for Network Access (PANA) Framework – IETF RFC 5193.

Références

[1] Gestion des identités - rapport technique - http://www.clusif.asso.fr/

[2] R. L. Morgan, S. Cantor, S. Carmody, W. Hoehn, K. Klingenstein, *"Federated Security: The Shibboleth Approach"*, EDUCAUSE Quarterly, vol. 27, no. 4, pp. 12-17, 2004.

[3] M. Kamel, « *Patrons organisationnels et techniques pour la sécurisation des organisations virtuelles* », IRIT/SIERA, UMR 5505, 29 septembre 2008, UPS Toulouse.

[4] S. Landau, T. Moore, *"Economic Tussles in Federated Identity Management"*, in Proceedings of the Tenth Workshop on the Economics of Information Security - (WEIS), (Jun 2011), http://uncommonculture.org/ojs/index.php/fm/article/view/4254.

[5] A. S. Wazan, R. Laborde, F. Barrère, A. Benzekri, D.W. Chadwick, *"PKI Interoperability: Still an Issue? A Solution in the X.509 Realm (regular paper)"*, World Conference on Information Security Education, Auckland, New Zealand, 08/07/2013-10/07/2013, Vol. 406, Springer Berlin/Heidelberg, IFIP Advances in Information and Communication Technology, p. 68-82, juillet 2013.

[6] P. A. Frausto Bernal, C. Antoine & A. Serhrouchni, « *Utilisations des certificats d'attribut pour accélérer l'usage de la signature électronique* », 5èmes journées Réseaux, JRES, Lille, novembre 2003.

[7] D. Chadwick, G. Zhao, S. Otenko, R. Laborde, L. Su & T. A. Nguyen, *"PERMIS: A Modular Authorization Infrastructure. Concurrency Computation: Practice & Expererience (2008)"*, 20: 1341-1357. doi: 10.1002/cpe.1313.

[8] R. Laborde, M. Kamel, A. S. Wazan, F. Barrère & A. Benzekri, *"A Secure Collaborative Web-Based Environment for Virtual Organisations"*, International Journal of Web Based Communities, Inderscience Publishers, Numéro spécial Dynamic Virtual Communities in the Information Society, Vol. 5 N. 2, p. 273-292, 2009.

[9] R. Laborde, M. Cheaito, F. Barrèrre, A. Benzekri, *"An Extensible XACML Authorization Web Service: Application to Dynamic Web Sites Access Control (regular paper)"*, dans Workshop on Security and Privacy in Telecommunications and Information Systems (SePTIS 2009), Marrakech, Morocco, 29/11/2009-04/12/2009, IEEE Computer Society, p. 499-505, 2009.

[10] M. Cheaito, R. Laborde, F. Barrèrre & A. Benzekri, *"A Deployment Framework for Self-Contained Policies (regular paper)"*, IEEE/IFIP International Conference on Network and Service Management (CNSM 2010), Niagara Falls - CANADA, 25/10/2010-29/10/2010, IEEE Communications Society, p. 88-95, janvier 2011.

[11] R. Laborde, M. Kamel, F. Barrère & A. Benzekri," *PEP = Point to Enhance Particularly"*; IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY 2008), IBM Palisades, 334 Route 9W, Palisades, NY 10964, USA, 02/06/2008-04/06/2008, IEEE Computer Society, p. 93-96, 2008.

[12] ITEA2 PREDYKOT project (Policies REfined DYnamically and Kept On Track) - http://www.itea2.org/project/index/view/?project=10104.

[13] G. López, O. Cánovas, A. F. Gómez, J. D. Jiménez & R. Marín, *"A Network Access Control Approach Based on the AAA Architecture and Authorization Attributes"*, Journal of Network and Computer Applications, Volume 30, Issue 3, August 2007, Pages 900-919, ISSN 1084-8045, http://dx.doi.org/10.1016/j.jnca.2005.07.010.

[14] Y. Demchenko, M. Cristea & C. de Laat, *"XACML Policy Profile for Multidomain Network Resource Provisioning and Supporting Authorisation Infrastructure"*, Policies for Distributed Systems and Networks, 2009. POLICY 2009. IEEE International Symposium on, vol., no., pp. 98, 101, 20-22 July 2009

[15] D.W. Chadwick, L. Su & R. Laborde, *"Use of XACML Request Context to Obtain an Authorisation Decision"*, Diffusion scientifique, novembre 2009. Open Grid Forum, OGSA Authorization WG, GFD Proposed Recommendation, P-REC 159.

[16] R. Ananthakrishnan, G. Garzoglio & O. Koeroo, *"An XACML Attribute and Obligation Profile for Authorization Interoperability in Grids"*, OpenGridForum GFD.205, 2013.

[17] J. Wiebelitz, M. Brenner, C. Kunz & M. Smith, 2010, *"Early Defense: Enabling Attribute-Based Authorization in Grid Firewalls"*, Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (HPDC '10). ACM, New York, NY, USA, 336-339.

[18] C. Ngo, P. Membrey, Y. Demchenko, C. de Laat, *"Security Framework for Virtualised Infrastructure Services Provisioned On-demand"*, Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference, vol., no., pp.698,704, Nov. 29 2011-Dec. 1 2011.

Abdelmalek Benzekri, François Barrère, Romain Laborde. Les auteurs sont membres de l'équipe SIERA de l'Institut de Recherche en Informatique de Toulouse. Ils conduisent leurs recherches dans le domaine de la gestion des technologies de sécurisation des systèmes informatiques. Leurs travaux visent à garantir l'objectif de satisfaction des exigences de sécurité des applications métiers.