Une Approche Dynamique pour la Gestion des Politiques de Délégation dans les Systèmes de Contrôle d´Accès

Khaled Gaaloul, François Charoy

Une Approche Dynamique pour la Gestion des Politiques de Délégation dans les Systèmes de Contrôle d´Accès

Khaled Gaaloul[*,**] **— François Charoy**[**]

[*] *SAP CEC Karlsruhe, Security & Trust Group*
Vincenz-Priessnitz-Strasse 1, 76131 Karlsruhe, Germany
{kgaaloul, charoy}@loria.fr

[**] *LORIA - INRIA - CNRS - UMR 7503*
BP 239, F-54506 Vandœuvre-lès-Nancy Cedex, France

RÉSUMÉ. *La délégation de tâche est un mécanisme qui permet d'obtenir une certaine flexibilité organisationnelle dans un système de gestion de workflow. Il permet également d'assurer une forme de délégation des autorisations dans un système de contrôle d'accès. Dans cet article, nous définissons une approche qui permet la délégation dynamique d'autorité dans un environnement de contrôle d'accès. La nouveauté consiste à raisonner sur les autorisation en fonction des évènements de délégation de tâche et de spécifier ces autorisations en terme de politiques de délégation. Lorsqu'un évènement de délégation est produit, la politique d'autorisation peut changer proactivement pour refléter les changements induits par la délégation d'autorité. Les travaux existants sur les systèmes de contrôle d'accès ne considèrent pas cette perspective. Nous montrerons ces limitations et proposons un cadre pour la délégation de tâche qui supporte la mise en place et le contrôle de politiques de délégation.*

ABSTRACT. *Task delegation is a mechanism that supports organisational flexibility in the human-centric workflow systems, and ensures delegation of authority in access control systems. In this paper, we define an approach to support dynamic delegation of authority within an access control framework. The novelty consists of reasoning on authorisation dependently on task delegation events, and specifies them in terms of delegation policies. When one of these events changes, our access policy decision may change proactively implying dynamic delegation of authority. Existing work on access control systems remain stateless and do not consider this perspective. We highlight such limitations, and propose a task delegation framework to support proactive enforcement of delegation policies.*

MOTS-CLÉS : *Workflow, Tâche, Délégation, Contrôle d'accès, Politique.*

KEYWORDS: *Workflow, Task, Delegation, Access Control, Policy.*

[e] soumission à *INFORSID 2009*, le 28/02/09.

1. Introduction

Les systèmes de gestion de workflow font maintenant partie de l'environnement classique des grandes organisations. Ces systèmes permettent la gestion et l'automatisation des procédés métiers et organisationels. Un procédé est classiquement défini comme un ensemble d'activités coordonnées, également appelées tâches. Les systèmes existants sont cependant aujourd'hui considérés comme rigides et de nombreux travaux ont pour but d'introduire de la flexibilité dans la modélisation et l'exécution des procédés et dans une moindre mesure de la flexibilité organisationnelle. C'est à cette dernière que nous allons nous intéresser à travers un mécanisme particulier : la délégation de tâches (Schaad, 2007; Schaad, 2003). En effet, nous considérons la délégation de tâche comme un support à la flexibilité organisationnelle dans des systèmes de workflow. La délégation permet de façon dynamique de transférer l'exécution d'une tâche de la personne à laquelle elle était assignée, en franchissant la barrière des rôles définies par l'organisation. Si nous considèrons le graphe des relations entre les rôles d'une organisation, la délégation va permettre, de façon temporaire, exceptionnelle, de transgresser certaines règles liées à ce graphe. Il faudra donc s'assurer que cette délégation entraine les délégations d'autorités nécessaires dans le système de contrôle d'accès basé sur le graphe organisationnel. En particulier, la délégation d'une tâche à un subordonné, ou la délégation d'une tâche à un membre d'un autre département de l'organisation va nécessiter des adaptations dynamique de la politique de sécurité. Cette délégation doit elle même faire partie de la politique de sécurité de l'organisation.

Normalement, les organisations établissent un ensemble de politiques de sécurité qui règle la façon de gérer les procédés métiers et les ressources (Atluri *et al.*, 2005). Une politique simple peut spécifier comment un utilisateur peut être assigné pour exécuter une tâche. Une politique plus complexe peut spécifier des contraintes d'autorisation supplémentaires pour permettre la délégation. Les contraintes d'autorisations de délégation sont définies par rapport à des évènements sur les couches de contrôle, de données et d'assignement de tâches du workflow (Gaaloul *et al.*, 2008). Une délégation sécurisée de tâches implique la présence d'un ensemble défini d'évènements de délégations d'autorités et de règles définissant les possibles délégations d'autorités ainsi que les moyens de contrôler les politiques associées.

L'essentiel du travail fait dans le domaine des contraintes de sécurité et des droits d'accès ne traite pas de la délégation de façon assez détaillée. Les travaux traitant du contrôle d'accès basé sur le modèle RBAC ("Role Based Access Control") (Sandhu *et al.*, 1996), qui permettent de définir des politiques de délégation manquent de flexibilité. L'essentiel des possibilités ou des règles de délégations sont définies à priori. (Seitz *et al.*, 2005). Les travaux traitant des systèmes d'autorisation ne considèrent pas la possibilité de contrôler des politiques dynamiques, c'est à dire susceptible de changer en fonction d'évènements de contexte (Chadwick *et al.*, 2006; Bertino *et al.*, 2000). Actuellement, les requêtes sur un système de contrôle d'accès sont sans état. La réponse à une requête donnée n'est valide qu'à l'instant ou la requête est faite. Si cette réponse change en raison d'une adaptation de la politique, aucun mé-

canisme n'existe pour transférer cette nouvelle réponse au demandeur initial de façon proactive. Ce mécanisme est cependant vital pour permettre une délégation dynamique d'autorité. Lorsque nous déléguons un tâche, une adaptation de la politique peut être nécessaire en fonction des évènements de la délégation. Le délégué, la personne à qui nous déléguons, peut acquérir de nouveaux droits grâce à cette délégation mais il peut les reperdre en cas de révocation de cette délégation. Ces évènements de délégation sont donc liés à des changements dynamiques des autorisations. Ceci ne peut être négligé par un système de sécurité avancé permettant la délégation de tâche.

Dans cet article, nous allons donc proposer la définition d'une approche permettant la délégation dynamique d'autorité. Celle ci devra supporter des politiques proactives dans un système de contrôle d'accès. Nous motiverons cette approche sur un exemple nécessitant la délégation pour l'exécution d'un procédé. Nous identifierons les évènements spécifiques du modèles de tâches correspondant à la délégation qui entraine des changements dynamiques de la politique. Nous séparerons les différents aspects de la délégation entre les utilisateurs, les tâches et les évènements, et nous les spécifierons en terme de politiques de délégation. Les politiques définies incluront les comportements nécessaires pour permettre les interactions en temps réel assurant la délégation dynamique d'autorité. Nous présenterons ensuite l´environnement de contrôle d'accès existant et discuterons leur fonctionnalités et leur limitations. Pour finir nous montrerons comment notre approche permet de contrôler dynamiquement les autorisations liées à la délégation et comment elle peut être intégrée dans les systèmes existants.

2. Contexte et problématique

2.1. *Exemple de Motivation*

Pour illustrer la problématique que nous voulons développer, nous allons nous appuyer sur un exemple tiré d'un cas réel impliquant la délégation de tâche. Dans le cadre des procédures criminelles au niveau européen, ont été mises en place des procédures d'assistance judiciaire mutuelles. Ceci permet par exemple à un état membre de faire exécuter une mesure de protection des témoins par un autre état dans une procédure criminelle (Gaaloul *et al.*, 2007). Nous ne décrivons ici que la partie Eurojust[1] du pays A de la procédure d'assistance criminelle ("Mutual Legal Assistance" - MLA). Elle consiste à recevoir une requête d'assistance d'un membre d'Europol[2], pour pouvoir la traiter et la transmettre à l'autorité concernée du pays B (Figure 1). L'utilisateur Alice, qui a le rôle procureur est assignée à une partie du procédé d'Eurojust A. Les activités du procédé sont représentées comme des tâches.

Nous avons appliqué la notation BPMN (Business Process Modeling Notation) à notre exemple de motivation. BPMN définit une notation standard pour la modélisation de procédés métiers, notamment aux niveaux d'analyse et de conception (Object

1. European Judicial Cooperation Unit. http ://www.eurojust.europa.eu/
2. European Police Office. http ://www.europol.eu.int/

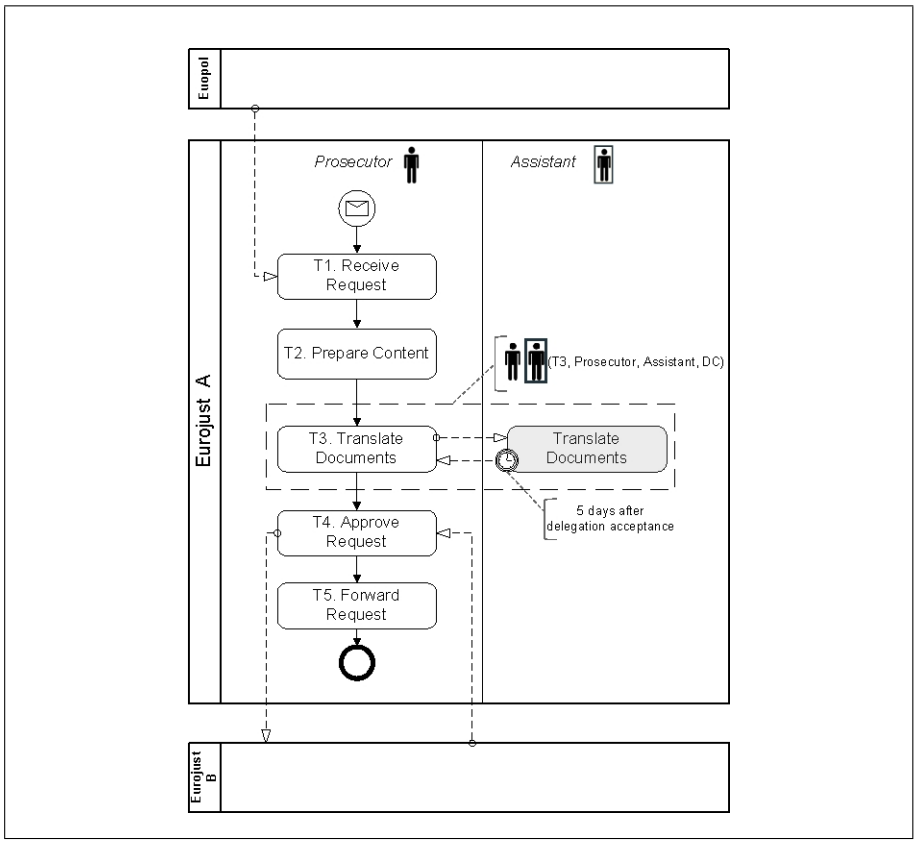

Figure 1. *Scénario de délégation MLA*

Management Group, 2006). Dans ce scénario, la tâche "Translate Documents" T3 n'est accessible au départ que par le procureur Alice. Ceci est défini par la politique du procédé. Cette tâche est de longue durée et il est prévu 5 jours pour la terminer. Alice n'est pas disponible pour la réaliser pour cause de maladie et doit donc la déléguer à Bob. Bob a pour rôle Assistant et est subordonné au procureur dans la hiérarchie de l'organisation. La délégation doit fournir les moyens suffisants pour permettre ce type de flexibilité organisationnelle (Gaaloul *et al.*, 2008).

Une politique peut être définie comme un moyen pour définir les accès aux ressource d'une tâche. Nous définissions une politique d'autorisations P pour le procédé MLA. Pendant une délégation, la politique P est mise à jour pour que l'utilisateur Bob puisse réaliser la tâche à la place d'Alice. Ainsi, Alice et Bob sont le délégant et le délégué, respectivement. Bob revendique la tâche, une requête de contrôle d'accès est effecuée, l'accès est accordé et il peut exécuter la tâche. Après deux jours, Alice revient au travail. Elle veut reprendre la tâche qui lui était affectée. Elle réclame à nouveau la tâche. En raison de sa qualification, la tâche est réaffectée à Alice et retirée

de la liste de tâches de Bob. La politique de sécurité doit être à nouveau mise à jour pour refléter le fait que seule Alice a maintenant accès à la tâche. La requête d'accès qui avait permis à Bob d'accéder aux ressources de la tâche retourne maintenant une décision de refus (deny).

Dans les environnements de contrôle d'accès classique, un mécanisme qui préviendrait Bob que son accès à une ressource est annulé automatiquement n'existe pas. Il n'est pas possible de revoquer une réponse donnée par une requête d'accès précédente. En outre, un contrôle manuel des droits d'accès couramment accordés pour l'exécution de tâche serait long, couteux et source d'erreurs. Un mécanisme dynamique permettrait d'informer dynamiquement le délégué d'un changement de politique en fonction des évènements de délégation. Ceci nécessite de supporter certaines interactions spécifiques dans l'architecture de contrôle d'accès utilisée. Ces interactions concernent en particulier les évènements de délégation de tâches qui doivent être capturés et retourné à l'émetteur de la requête pour prendre les mesures appropriées.

2.2. *Analyse et Discussion*

Nous allons nous baser sur un modèle de délégation basé sur les rôles pour contrôler des interactions inter-humaines dans le contexte de tâches de longue durée (Barka *et al.*, 2000). Nous faisons l'hypothèse que l'exécution d'une tâche est atomique et que la délégation d'autorité est accordée uniquement au délégué. Ainsi, nous ne considérons pas la possibilité de faire de la délégation en cascade ou de la délégation partielle (Zhang *et al.*, 2003). Lors d'un évènement de délégation, nous définissons comme suit la relation de délégation :

Definition 1. Un relation de délégation est définie par $RD = (T,u_1,u_2,DC)$, où T est la tâche déléguée, u_1 le délégateur, u_2 le délégué, et DC l'ensemble des contraintes de délégation.

Les contraintes de délégation correspondents aux droits de déléguer par rapport à la politique définie pour le procédé. Par exemple, DC est basé sur la hiérarchie de rôles (RH) d'Eurojust, où l'assistant Bob est sous l'autorité de la procureur Alice. De plus, DC implique aussi une délégation temporaire. Une période de délégation doit être définie. Bob n'est pas autorisé à dépasser la date limite de T3 (5 jours de travail). Nous définisons la relation de délégation pour T3 ainsi :
$RD = (T3,Alice,Bob,(RH,5\ days))$.

La disponibilité d'un mécanisme dynamique ou proactif d'exécution des politiques de sécurité est vital pour permettre la délégation de tâches de longue durée. Cela nécessite d'être capable de capturer les évènements de délégation et de les transmettre à l'émetteur de la requête pour faire les mises à jour nécessaires dans le système de contrôle d'accès. Actuellement, nous savons contrôler les droits d'accès d'une délégation à travers une adaptation de la politique, pour permettre au délégué d'effectuer

la tâche déléguée. Ensuite nous devons mettre à jour la relation de délégation dans la politique du procédé chaque fois qu'un tel évènement est produit. Cela consiste en l'introduction de nouvelles règles d'autorisation pour le délégué. Si cette règle change suite à une révocation, une nouvelle réponse devra être transmise au délégué dynamiquement.

3. Politiques de délégation de tâche basées sur un modèle événementiel

Dans cette section, nous nous intéressons à l´identification d´événements qui pourraient être à l´origine de définition de nouvelles politiques d´autorisation. Tout d'abord, nous présentons notre modèle de délégation de tâche, où nous définissons les transitions comme des événements décrivant le cycle de vie d´une tâche. Ensuite, nous analysons les événements capable d´appliquer de nouvelles règles de changement aux politiques de délégation. En effet, notre préoccupation est d´assurer une politique de délégation dynamique supportant l´exécution d´une tâche déléguée. Le but est de pouvoir répondre aux changements dynamiques qui peuvent se produire lorsqu´une tâche est déléguée. Ceci est motivé par le fait que lorsqu´un événement se produit, notre politique de gestion d´accès doit répondre à ce changement conformément à l´évolution de l´exécution de la tâche déléguée.

3.1. *Modèle de délégation de tâche*

Dans nos récents travaux, nous avons défini un modèle de délégation de tâche (MDT) (Gaaloul *et al.*, 2008; Russell *et al.*, 2005). Ce modèle est basé sur les spécifications du cycle de vie d´une tâche définies par la coalition de gestion de systèmes de workflow (Workflow Management Coalition, n.d.a). MDT définit un diagramme UML (Unified Modeling Notation) d´états/transitions d'une tâche au sein d'un procédé de workflow. Une tâche une fois créé, est généralement attribué à un utilisateur. L'utilisateur pourrait par la suite l´exécuter ou la déléguer à quelqu´un d´autres. La délégation dépend du droit dont dispose cet utilisateur pour requérir une demande de délégation.

Les événements intermédiaires définissent le contrôle de délégation de tâche (e.g, *"delegate"*, *"cancel"*, *"revoke"*). Par exemple, le délégant a besoin d´annuler son action. Dans ce cas, le TDM doit assurer une alternative pour annuler la délégation (l´événement "cancel") et revenir à l´état précédent : "Assigned". Le contrôle de délégation reste interne au modèle de tâche et son comportement suit le flux de contrôle habituel, où "Completed", "Cancelled" et "Failed" représentent les états finaux (voir Figure 2).

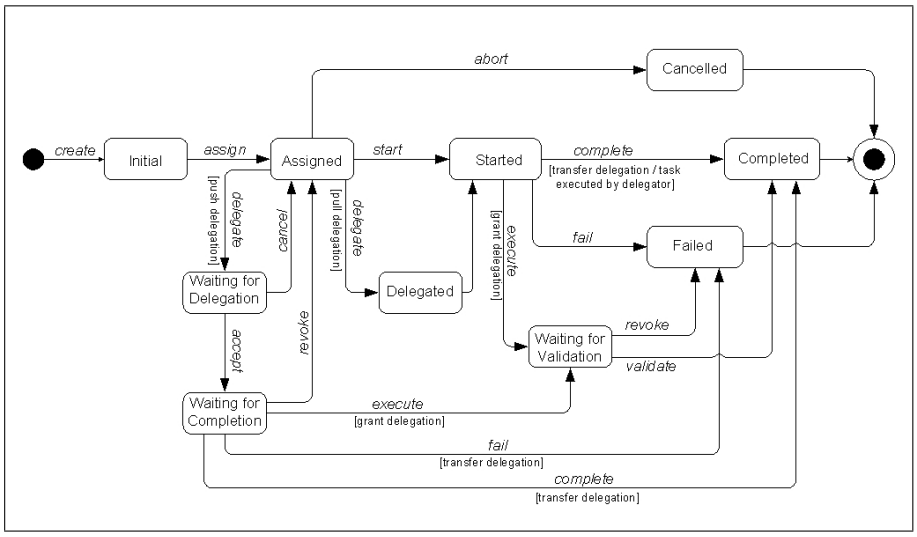

Figure 2. *Modèle de délégation de tâche*

3.2. *Analyse de besoins en sécurité*

Dans notre modèle MDT, nous définissons les transitions comme des événements qui contrôlent le comportement de délégation. Nous avons enrichi ce modèle avec des propriétés décrivant le mode et le type de délégation. En effet, une tâche déléguée peut s´exécuter de différentes manières selon le contexte de son exécution. Un délégant peut avoir à sa disposition une liste de délégués potentiels qui peuvent exécuter la tâche pour son compte. Il s´agit dans ce cas d´un mode "*Pull"*. Une autre propriété concerne le type d´attribution de privilèges afin d´accéder aux ressources de la tâche pour l´exécuter. Dans la littérature, nous distinguons deux types : *"grant"* et *"transfer"* (Crampton *et al.*, 2006). Dans ce qui suit, nous analysons les exigences en sécurité qui doivent être pris en compte pour définir les politiques de délégation basées sur les événements de délégation (voir Figure 2).

– **Mode de délégation** : Il défini la manière dont une requête de délégation est issue. Le *"Pull"* mode suppose que le délégant dispose d´une liste de délégués potentiels qui peuvent exécuter la tâche pour son compte. Le *"Push"* mode suppose qu'un délégant est en attente d'acceptation d'un délégué correspondant au profil de la requête (e.g. rôle). Nous définissons les évènements suivants *"accept"*, *"cancel"* et *"revoke"* comme les évènements relatifs au mode *"Push"*.

– **Type de délégation** : Il existe deux types : *"grant"* et *"transfer"*. Une délégation type *"grant"* permet que le droit d´accès (privilèges) soit disponible, à la fois, pour le délégant et le délégué. À ce titre, le délégant garde toujours le contrôle sur sa tâche et a le droit de valider (*"validate"*) ou de révoquer (*"revoke"*) le travail du délégué. Toutefois, en cas de transfert de la délégation, tous les droits sont transférés au délégué.

Il n'y a pas de validation requise et la tâche est terminée (*"complete"/"fail"*) par le délégué. Le type *"transfer"* est pratique dans la délégation de tâches administratives.

– **La delegation d´autorité** : Il permet à un délégant de céder une partie de ses privilèges à un délégué qui, à priori, ne possèdent pas les autorisations nécessaires pour exécuter la tâche. Par exemple, *"delegate"* définit un événement qui va déclencher la délégation de tâche. Ainsi, *"delegate"* va appliquer une nouvelle politique de contrôle d'accès pour le délégué. Nous allons assurer une nouvelle règle dans la politique afin d´autoriser le délégué à exécuter la tâche déléguée.

– **Application de contrôle d´accès** : Le but est d´assurer une politique de délégation dynamique. Par exemple, l´évènement *"cancel"* implique la révocation de la délégation où le délégant va reprendre le contrôle sur sa tâche et, par conséquent, annuler la décision précédente. Cette annulation définit une nouvelle règle dans la politique en appliquant un refus d´accès instantané au délégué.

Par la suite, nous classons les événements de délégation et identifions les relations entre ces événements, les propriétés de délégation et leurs impacts sur les politiques d´autorisation (voir Tableau 1). Nous pensons que des événements tels que *"accept"* ou *"validate"* font parti des politiques de délégation et ont un impact direct sur le changement dynamiques de politiques.

Les évènements de délégation	Le mode Push		Le mode Pull		Changement de politiques
	Grant	Transfer	Grant	Transfer	
delegate	✓	✓	✓	✓	✓
accept	✓	✓	*x*	*x*	✓
cancel	✓	✓	*x*	*x*	*x*
execute	✓	*x*	✓	*x*	*x*
validate	✓	*x*	✓	*x*	✓
revoke	✓	*x*	✓	*x*	✓
fail	*x*	✓	*x*	✓	*x*
complete	*x*	✓	*x*	✓	*x*

Tableau 1. Politiques d´autorisation basées sur les évènements de délégation

Pour mieux expliquer le tableau, nous considérons les deux exemples suivants :

Exemple 1 : L´événement *"validate"* est défini dans les deux modes de délégation (Push et Pull). Il intervient lors d´une délégation de type *"grant"*, où le travail du délégué doit être validé par le délégant. Cet événement mettra en oeuvre un changement de politique. En effet, la validation conduit à l'achèvement de la tâche et la révocation des privilèges délégués. Ainsi, le travail de Bob est validé par Alice et ses droits d´accès ne seront plus valables dans la politique par la suite.

Exemple 2 : L´événement *"fail"* est défini dans les deux modes de délégation (Push et Pull). Il intervient lors d´une délégation de type *"transfer"*, où un délégué termine la tâche sans besoin de validation. Les politiques de délégation prendront fin suite à la terminaison de la tâche.

Dans le scénario e-gouvernemental que nous avons présenté, nous pouvons observer une politique de délégation dynamique régie par les évènement qui peuvent avoir lieu au cours de l´exécution de la tâche T3. L'utilisateur Alice est de retour avant la terminaison de délégation. Alice devrait révoquer l´effet de délégation en annulant ce qui a été effectué par Bob. Alice sera en mesure d´exécuter la tâche T3 et ainsi annuler la politique d´autorisation initialement issue à Bob. L'événement *"revoke"* sera mis à jour dans la politique globale et une notification sera transmise à Bob pour procéder aux actions nécessaires d´annulation.

4. Une approche sécurisée pour la délégation de tâche

Dans cette section, nous développons une approche sécurisée pour la délégation de tâche. Nous présentons une architecture modulaire assurant une gestion dynamique pour la délégation de privilèges. Notre approche a pour but de supporter des politiques de contrôle d´accès proactives régies par les évènements de délégation définis dans la section précédente. Note approche sera implémentée au sein de systèmes existants de contrôle d´accès. Dans le cadre de la délégation de tâche, lorsqu'une requête est émise, elle est ensuite stockée de manière à informer le délégué si la décision politique à cette requête vient de changer. En effet, la réponse à de telles requêtes évolue en fonction des changements de politiques qui peuvent se produire au cours d´une délégation. Des évènements tels que l´annulation, la révocation ou la non validation d´une tâche déléguée doivent forcément appliquer une nouvelle décision à la réponse précédente (voir les changements de politiques identifiés dans le tableau 1). De ce fait, les requêtes antérieures seront réévaluées, et le délégué sera informé que ses droits d'accès ont changé. L´implémentation de cette approche se base sur des systèmes existants de contrôle d´accès. L´idée est de développer un module dynamique comme extension à ces systèmes afin de supporter des politiques de délégation proactives.

4.1. *Vue d´ensemble de l´architecture*

Nous présentons les principaux composants de l´architecture qui va assurer la gestion de politiques proactives lors de la délégation de tâche. Nous procédons par la description de l´architecture existante et des différents modules nécessaires à notre approche (voir Figure 3).

– **Le Gestionnaire de politiques** : Il permet à un administrateur de définir des politiques de contrôle d ´accès. Grâce à une interface graphique, l´administrateur peut naviguer dans la base de données des politiques, sélectionner un document, modifier une politique (la cible, le sujet, les règles d'autorisation), et préciser les décisions pour

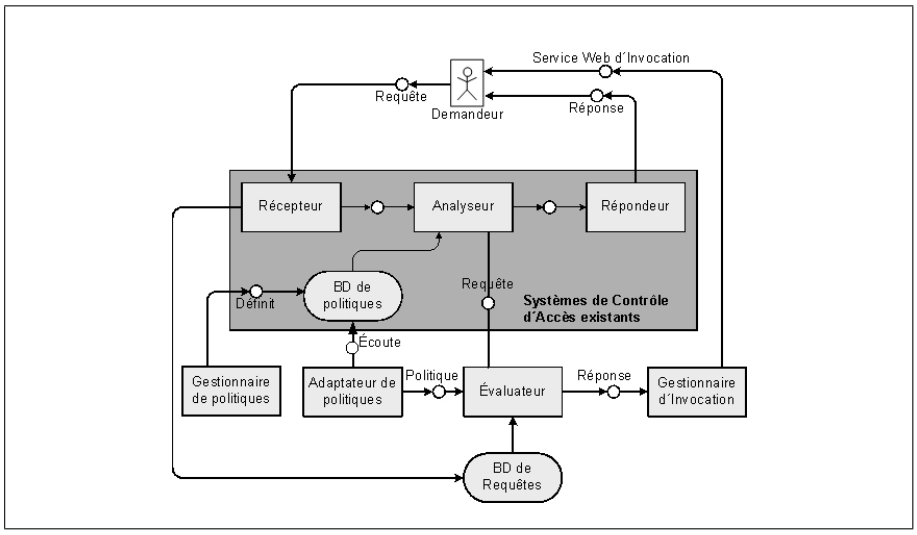

Figure 3. *Un cadre sécurisé pour des politiques de délégation proactives*

les éléments sélectionnés. Par exemple, un administrateur définit une politique d'autorisation *P* qui retourne comme réponse *"permis"* (autoriser) pour la *cible* tâche T3 ayant comme *sujet* l´utilisateur Alice dont le *rôle* est procureur. Dans notre contexte de délégation, nous assumons qu´un délégant est autorisé à administrer des politiques, et ainsi définir des règles de délégation. Les politiques de délégation encapsulent en elles même les attributs d´identification du délégué pour les besoins d'authentification et d'autorisation.

– **Un cadre de contrôle d'accès (CCA)** : Il est défini comme un ensemble de composants logiciels qui gèrent les requêtes d'accès à une ressource donnée en analysant les informations de cette requête dans la Base de données (BD) des politiques. Les informations récupérées de cette requête rassemblent les attributs du demandeur (l´émétteur de la requête) ainsi que la ressource requise (cible). Ces informations seront traitées dans le CCA et une réponse sera émise par la suite. Pour illustrer l'architecture originale d´un CCA, nous décrivons les principaux composants impliqués dans la gestion de contrôle d´accès. Une requête est émise par le demandeur, qui est reçue par le composant *Récepteur*. La requête est transmise ensuite au composant *Analyseur* qui va vérifier les informations dans la base de données. Une réponse est générée enfin par le composant *Répondeur* qui envoie le résultat de la décision (e.g., permis, refus, indéterminé ou non applicable) au demandeur.

– **Mise en oeuvre des politiques dynamiques** : Elle définit notre approche pour supporter des décisions de politiques proactives. Notre approche consiste à étendre l´architecture du CCA avec des composants supplémentaires. Lorsque le *Récepteur* reçoit une requête, il enregistre cette requête dans une *BD de Requêtes*. L´*Adaptateur* de politiques interroge la BD de politiques pour voir si un nouvel évènement est pro-

duit et qui pourrait changer la décision précédente. Il procède en envoyant cette information au composant *Évaluateur* qui va comparer avec l´ancienne politique stockée dans la BD de requêtes. En cas de changement, la nouvelle politique sera renvoyée à l´*Analyseur* et une nouvelle réponse sera émise. Un nouvel évènement tel que *revoke* va appliquer une modification à l´ancienne politique de délégation et ainsi une mise à jour dans la BD des politiques d´autorisation. Au niveau utilisateurs, un service d´invocation via un "point de contact" va notifier le délégué de la nouvelle décision (voir Figure 3).

4.2. *Analyse des besoins*

Sur le plan architectural, les requêtes doivent être réévaluées à chaque changement de politiques. Pour cela, nous aurons besoin d´un mécanisme de stockage des requêtes précédentes ainsi que de leurs résultats. En effet, si la réévaluation d´une requête génère un résultat différent du premier résultat enregistré, le CCA doit d´informer le demandeur du nouveau résultat. Ainsi, il doit exister un mécanisme qui déclenche une nouvelle évaluation lorsqu'il détecte un changement de politique. Ces effets de changement de politiques seraient capturés automatiquement, et puis transmise au demandeur, dans ce cas le délégué, pour procéder aux mesures appropriées à ce changement. En outre, nous définissons un élément d´*Invocation* qui va acheminer cette information au délégué.

Sur le plan du langage, il faudrait définir de nouveaux constructeurs pour la description de la méthode d'invocation que le CCA utiliserait pour contacter le demandeur. Il s´agit de développer un *point de contact* pour le demandeur. Dans une architecture orientée services (SOA), ce point de contact peut être un point final d´un service ("service endpoint") qui pourrait être invoqué par le système (voir le service web d´invocation dans Figure 3). De ce fait, toutes les politiques d'accès doivent être centralisées et référencées par l'architecture SOA, qui sera protégée. Nous définissons un point d'accès unique et nous enregistrons les services web dans notre CCA. Depuis que les services sont essentiellement des boîtes noires, nous définissons la manière de les contacter et de les orchestrer lors d´un changement de politiques.

Sur le plan technique, le *Gestionnaire de politiques* génère des politiques dans lesquelles il intègre des attributs d'authentification et d'autorisation (Gaaloul *et al.*, 2008). Des fournisseurs d´authentification tels que les autorités de certification numérique délivrent des certificats au demandeur afin de traiter sa requête. À ce stade, le *Récepteur* agit comme un élément d´application de la politique ("Policy Enforcement Point") pour supporter la demande d´accès et contrôler la décision de la politique. Par exemple, un Certificat d´Attributs (CA) X.509 est délivré au délégué à des fins d'authentification et d'autorisation. Un CA assure l'intégrité, la protection et la non-répudiation des informations échangées par l'intermédiaire d'une signature numérique. Le *Récepteur* obtient le certificat d'attributs du délégué et calcule ses permissions par la suite. Ces attributs seront validés par rapport à la politique (par exemple, le rôle du demandeur, la période de validité). Une fois que le délégué a été authentifié

avec succès, il tentera d'effectuer des actions sur les ressources de la tâche spécifiée. À chaque tentative, le *Récepteur* transmet la demande d'accès à l´*Analyseur* pour décider. Les résultats de décisions (permis, refus, ou non applicable) seront alors envoyés via le *Répondeur*.

Une nouvelle réévaluation d'une nouvelle politique requiert de nouveaux CA pour des requêtes ultérieures par rapport aux changements de politiques. Par exemple, une révocation implique l'annulation du CA délivré précédemment pour le délégué. Actuellement, des techniques comme des certificats temporaires ou des listes de révocation de certificats sont basées sur la notion de temps, et par conséquent, ne répondent pas aux exigences des événements. Pour y remédier, nous proposons un environnement orienté services, où un service est appelé à prendre contact avec le délégué. En définissant un accord mutuel entre les deux instigateurs (le délégant et le délégué), des mesures appropriées seront prises suite au déploiement de ce service. Dans notre étude de cas, Bob sera amené à annuler son travail sur la tâche T3 en libérant les ressources et en fermant sa session d´accès.

5. État de l'art

Le modèle de contrôle d´accès exentisble XACML (eXtensible Access Control Markup Language) a été développé afin de définir une manière uniforme de spécification des politqiues de contrôle d'accès dans le langage XML (XACML, February 2005). Les politiques sont décrites par des règles qui peuvent être restreintes par des conditions. Ces règles s´appliquent à des cibles définies par un ensemble de sujets, de ressources et d´actions. Ces spécifications seront reprises lors d´une requête, où elles seront confondues avec les attributs de la requête (e.g. le demandeur comme sujet, la tâche comme ressource, la permission comme action, etc.). Les résultats ou les effets d'une politique peuvent être : Permis, Refus, Non applicable ou Indéterminée. Le standard XACML actuel ne fournit pas de support explicite de délégation. Cependant, il ya eu quelques essaies d´extension du modèle pour des règles de délégation mais qui restent loin d´être concrèts pour des politiques réelles de délégation (SAML 2.0 profile of XACML v2.0, February 2005).

Chadwick et al. (Chadwick *et al.*, 2006) ont proposé une solution XACML supportant une délégation dynamique des autorités. L'approche décrit une entité appelée le service de validation des attributs. Elle a pour but de synchroniser avec le module XACML de décisions ("Policy Decision Point") pour la prise de décisions d´autorisation. L'architecture proposée offre un moyen souple et dynamique pour gérer les informations d'identification, mais cela ne couvre pas tous les aspects de délégation dynamique. En effet, nous avons montré que le fait de déléguer une tâche exige plus d'efforts et de spécifications supplémentaires liées à ses événements. Notre modèle de délégation de tâche événementiel permet de déterminer avec fiabilité les politiques dynamiques de délégation, et ainsi assurer des décisions proactives lorsque des événements tels que la révocation ou la validation sont déclenchés au cours de la délégation des tâches.

Seitz et al. (Seitz *et al.*, 2005) ont étudié la manière dont un système de gestion d´autorisation, utilisant le langage XACML, peut être étendu à des mécanismes de délégation administratives. Ils ont mis au point un module qu´ils ont appelé Delegent pour la gestion des modifications autorisées sur des politiques. Ce travail a été implémenté par la suite dans une solution logicielle appelée Axiomatics pour la gestion de contrôle d´accès (AXIOMATICS, 7-11 April 2008). L´idée est de fournir un outil d´administration pour la politique de contrôle d'accès supportant la mise à jour de ces politiques. Cependant, cette administration reste passive et sans états. Elle manque de réactivité pour supporter des changements dynamiques de politiques. Nous avons besoin d'une approche réactive pour tenir compte des événements de délégation lors de l´administration des politiques d´autorisation.

PERMIS (Chadwick *et al.*, 2003) est une solution intergicielle pour la gestion d´autorisation. PERMIS est basé sur le modèle RBAC ("Role Based Access Control"). La politique est exprimée en XML et offre des spécifications relatives à la validité des rôles, des ressources et de la profondeur de la délégation intra- organisations. Cependant, le cadre PERMIS ne fournit pas de support direct à des échanges bilatéraux de politiques inter-organisations. D´ou les lacunes de sécurité et de confiance dans un environnement collaboratif et dynamique de plusieurs organisations indépendantes partageant leurs ressources. Outre le besoin de collaboration, il y aussi le besoin en interactions humaines inter-organisationnels qui sont définis par des mécanismes de délégation ad-hoc. Ce qui est le cas dans notre exemple de motivation e-gouvernemental.

6. Conclusion et perspectives

Définir une politique de contrôle d´accès dynamique supportant les exigences de délégation est loin d´être une tâche triviale. Dans ce papier, nous avons présenté les problèmes et les exigences que demande un modèle de délégation de tâche. Nous avons également présenté à un niveau conceptuel, les différents éléments qui sont nécessaires pour la mise en oeuvre de telles politiques. La motivation de cette direction est inspirée de scenarios réels relatives aux procédés e-gouvernementaux, où les besoins d´interactions humaines se font de plus en plus sentir. Nous considérons la délégation de tâche comme un support à la flexibilité organisationnelle dans des systèmes de workflow. Il permet également d'assurer une forme de délégation des autorisations dans un système de contrôle d'accès. Pour ce faire, nous avons montrer que les politiques de délégation peuvent changer selon des évènements spécifiques. Nous avons défini la nature de ces événements basés sur notre modèle de tâche MDT, et avons décrit leurs interactions avec les décisions des politiques respectives. Lorsque des événements appropriés se produisent, nous définissons la façon dont ils seront détectés et traités. Dans ce contexte, nous avons proposé une extension aux systèmes de contrôle d'accès afin de permettre la spécification et le traitement des politiques de délégation dynamiques.

Comme perspectives, nous allons nous intéresser à la spécification des évènements de délégation au sein d'un standard de contrôle d'accès dans le langage XML. Nous

sommes en train de travailler sur une extension du méta modèle en XACML. Cela nous servira de base pour communiquer avec les modules de réévaluation, d´adaptation et d´invocation orientés services. Nos travaux futurs vont aussi dans la direction de gestion des performances basée sur l´extraction de l´historique de délégation. En effet, la gestion d´un tel historique peut être une approche intéressante pour la vérification d'audit.

7. Bibliographie

Atluri V., Warner J., « Supporting conditional delegation in secure workflow management systems », *SACMAT '05 : Proceedings of the tenth ACM symposium on Access control models and technologies*, ACM Press, New York, NY, USA, p. 49–58, 2005.

AXIOMATICS, « Axiomatics : Delegent Authorisation System », 7-11 April 2008. OASIS XACML InterOp DemoRSA. Conference 2008 San Francisco, California, USA.

Barka E., Sandhu R., « Framework for role-based delegation models », *Proceedings of the 16th Annual Computer Security Applications Conference*, IEEE Computer Society, Washington, DC, USA, p. 168–176, 2000.

Bertino E., Castano S., Ferrari E., Mesiti M., « Specifying and enforcing access control policies for XML document sources », Kluwer Academic Publishers, Hingham, MA, USA, p. 139–151, 2000.

Chadwick D. W., Otenko A., « The PERMIS X.509 Role based privilege management infrastructure », *Future Generation Comp. Syst.*, vol. 19, n° 2, p. 277-289, 2003.

Chadwick D. W., Otenko S., Nguyen T.-A., « Adding Support to XACML for Dynamic Delegation of Authority in Multiple Domains », *Communications and Multimedia Security, 10th IFIP TC-6 TC-11 International Conference, CMS 2006, Heraklion, Crete, Greece, October 19-21, 2006, Proceedings*, p. 67-86, 2006.

Crampton J., Khambhammettu H., « Delegation in Role-Based Access Control », *Proceedings of the Computer Security - ESORICS 2006, 11th European Symposium on Research in Computer Security, Hamburg, Germany, September 18-20, 2006*, Lecture Notes in Computer Science, Springer, p. 174-191, 2006.

Gaaloul K., Charoy F., Schaad A., Lee H., « Collaboration for Human-Centric eGovernment Workflows », *Web Information Systems Engineering, Proceedings of the WISE 2007 International Workshops, Nancy, France*, Lecture Notes in Computer Science, Springer, p. 201-212, 2007.

Gaaloul K., Schaad A., Flegel U., Charoy F., « A Secure Task Delegation Model for Workflows », *SECURWARE '08 : Proceedings of the 2008 Second International Conference on Emerging Security Information, Systems and Technologies*, IEEE Computer Society, Washington, DC, USA, p. 10–15, 2008.

Nguyen T.-A., Su L., Inman G., Chadwick D. W., « Flexible and Manageable Delegation of Authority in RBAC », *21st International Conference on Advanced Information Networking and Applications (AINA 2007), Workshops Proceedings, Volume 2, May 21-23, 2007, Niagara Falls, Canada*, p. 453-458, 2007.

Object Management Group, « Business Process Modeling Notation Specification », 2006. http ://www.bpmn.org.

Russell N., van der Aalst W. M. P., ter Hofstede A. H. M., Edmond D., « Workflow Resource Patterns : Identification, Representation and Tool Support », *In Proceedings of the Advanced Information Systems Engineering, 17th International Conference, CAiSE 2005, Porto, Portugal*, p. 216-232, 2005.

SAML 2.0 profile of XACML v2.0, « eXtensible Access Control Markup Language (XACML). OASIS Standard », February 2005. http ://docs.oasis-open.org/xacml/2.0/access-control-xacml-2.0-saml-profile-spec-os.pdf.

Sandhu R. S., Coyne E. J., Feinstein H. L., Youman C. E., « Role-Based Access Control Models », *IEEE Computer*, vol. 29, n° 2, p. 38–47, 1996.

Schaad A., « A Framework for Organisational Control Principles. PhD thesis, The University of York », 2003.

Schaad A., « A Framework for Evidence Lifecycle Management », *Web Information Systems Engineering, Proceedings of the WISE 2007 International Workshops, Nancy, France*, Lecture Notes in Computer Science, Springer, p. 191-200, 2007.

Seitz L., Rissanen E., Sandholm T., Firozabadi B. S., Mulmo O., « Policy administration control and delegation using XACML and Delegent », *6th IEEE/ACM International Conference on Grid Computing (GRID 2005), November 13-14, 2005, Seattle, Washington, USA, Proceedings*, p. 49-54, 2005.

Workflow Management Coalition, « Workflow Management Coalition Terminology and Glossary », n.d.a. Document Number WFMC-TC-1011, February 1999.

Workflow Management Coalition, « Workflow Security Considerations », n.d.b. White Paper, Document Number WFMC-TC-1019, March 2001.

XACML, « eXtensible Access Control Markup Language (XACML v2.0). Standard, Organization for the Advancement of Structured Information Standards (OASIS) », February 2005. http ://docs.oasis-open.org/xacml/2.0/access-control-xacml-2.0-core-spec-os.pdf.

Zhang L., Ahn G.-J., Chu B.-T., « A rule-based framework for role-based delegation and revocation », *ACM Transactions on Information and System Security*, vol. 6, n° 3, p. 404–441, 2003.