Une vision pervasive d'un système multimédia distribué : extention en vue d'un contrôle d'accès adaptatif

Dana Al Kukhun, Dana Codreanu, Ana-Maria Manzat, Florence Sèdes

Une vision pervasive d'un système multimédia distribué : extention en vue d'un contrôle d'accès adaptatif

Application dans le domaine de la vidéo surveillance

Dana Al Kukhun, Dana Codreanu, Ana-Maria Manzat, Florence Sèdes

IRIT, Université de Toulouse III
118 Route de Narbonne, 31062 Toulouse CEDEX 9, France
prenom.nom@irit.fr

RESUME. *LINDO est un projet visant à proposer un système de gestion de contenus multimédias distribués, en se focalisant sur leur indexation, stockage et recherche avec une contrainte de minimisation de la consommation des ressources. Cet objectif est réalisé par l'implémentation d'une technique d'indexation différée et distribuée des contenus. Dans le contexte dans lequel l'utilisateur est de plus en plus mobile et les données de plus en plus sensibles, le besoin d'ajouter une couche du contrôle d'accès émerge. Ce control d'accès peut engendrer un refus d'accès aux contenus, si l'utilisateur n'a pas les droits d'accès, ce qui peut être inconvénient dans certaines situations. Dans ce papier, nous abordons ces problèmes du refus d'accès et proposons d'appliquer un mécanisme adaptatif qui réécrit les requêtes d'accès en prenant en compte le contexte de l'utilisateur et sa situation. Notre proposition est basée sur l'architecture PSQRS qui assure l'accès aux ressources dans des contextes pervasifs. Le PSQRS utilise le modèle RBAC et le standard XACML.*

ABSTRACT. *The LINDO project proposes an open system dedicated for managing distributed multimedia contents by focusing on their indexing, storage and retrieval. The objective is to provide efficient information retrieval with minimal processing costs. This was achieved through the proposal of an efficient decentralized content indexing mechanism. Considering a pervasive context, where users are mobile and acquire access to sensitive data sources, adding an access control layer on top of the LINDO architecture becomes essential. In this paper, we confront the access denial problems and we propose to apply an adaptive mechanism that rewrites access requests while considering the user's context and situational information. This operation is performed through PSQRS, a query-rewriting system that manages access based on the RBAC model and realizes decision-making using XACML.*

MOTS-CLES : *Systèmes Pervasifs, Systèmes Multimédias Distribués, Contrôle d'Accès, Adaptation, Vidéo Surveillance.*

KEYWORDS: Pervasive Systems, Distributed Multimedia Systems, Access Control, Adaptation, Video Surveillance.

1. Introduction

La nécessité de gérer une quantité toujours plus volumineuse de contenus multimédias complexes créés depuis plusieurs sources, dans un environnement distribué, pose de nouveaux défis concernant l'indexation et l'accès au contenus multimédias, comme: le stockage distribué et la décentralisation du traitement, le choix des algorithmes d'indexation, la recherche d'information en temps réel et l'accès géo-localisé aux contenus. Nous devons considérer également que les utilisateurs sont de plus en plus mobiles et prendre en compte leur besoin d'accéder au système depuis n'importe où, à n'importe quel moment. Dans de tels contextes mobiles et pervasifs, la privacité et la sécurité du système et des données sont des problèmes importants à prendre en compte.

Dans cet article, nous présentons une extension de l'architecture proposée dans le cadre du projet LINDO, afin de relever les défis mentionnés ci-dessus. L'objectif du projet LINDO était de construire un système distribué de gestion de contenus multimédias, et d'assurer une indexation et un stockage efficaces des données multimédias acquises en temps réel. Dans les objectifs du projet, les questions liées à la confidentialité des données et la sécurité n'ont pas été mentionnés.

Sachant que, assurer la protection des contenus multimédias est une question clé dans certains domaines d'application (e.g., la vidéo surveillance, le domaine médical), la gestion du contrôle d'accès doit être prise en compte à différents niveaux dans le traitement des données et elle doit tenir compte de la mobilité de l'utilisateur. Les contraintes de sécurité ne doivent cependant pas affecter l'accessibilité aux données, en particulier dans les situations d'urgence. Notre objectif est d'inclure le contrôle d'accès dans le traitement de la requête dans le cadre du système proposé par le projet LINDO afin de permettre à l'utilisateur d'accéder à des sources multimédias à n'importe quel moment et de n'importe où. Pour atteindre cet objectif, nous avons utilisé PSQRS - Pervasive Situation-aware Query Rewriting System (Al Kukhun et Sèdes, 2008) - qui offre des solutions pour l'adaptation de l'accès aux ressources prenant en compte le contexte de l'utilisateur et la situation. Ce système est basé sur le modèle RBAC (Ferraiolo et Kuhn, 1992) et la norme XACML (OASIS, 2003).

La solution permet de surmonter les refus d'accès dans des situations de mobilité, en temps réel, en modifiant le mécanisme de traitement des requêtes et en fournissant des solutions d'adaptation qui peuvent contourner les contraintes du contrôle d'accès.

L'article est organisé de la façon suivante : dans la section 2, nous présentons un état de l'art couvrant les systèmes de gestion de contenus multimédias distribués

dans la section 2.1, les normes pour la gestion du contrôle d'accès distribué dans la section 2.2 et les travaux sur le contrôle d'accès multimédia dans la section 2.3.

L'approche LINDO pour une gestion efficace des contenus multimédias distribués est décrite dans la section 3 au travers de l'architecture proposée, de l'indexation et des mécanismes d'interrogation adoptés. Dans la section 4, nous appliquons une couche de contrôle d'accès à l'architecture LINDO. Dans la section 5, la solution de contrôle d'accès adaptatif est illustrée par un cas d'utilisation dans le domaine de la vidéosurveillance. Enfin, les conclusions et les perspectives sont fournies dans la section 6.

2. État de l'art

2.1. Systèmes Multimédias Distribués

La forte croissance de la quantité de contenus multimédia qui sont générés chaque jour met en avant des problèmes liés à l'indexation et leur recherche. La solution à ces problèmes passe par la production et la gestion des métadonnées associées aux contenus multimédias.

Une gestion distribuée des contenus multimédias est utilisée par de nombreux projets en raison du contexte d'acquisition mobile de ces contenus. Un avantage de ce type de systèmes est qu'il bénéficie du stockage distribué et du traitement des contenus multimédias et que, par conséquent, les performances du système peuvent être améliorées. Les systèmes distribués qui gèrent les contenus multimédias emploient des architectures pair-à-pair ou orientées services. Le problème majeur que rencontrent ces systèmes est l'hétérogénéité des algorithmes d'indexation, qui ont différentes performances, objectifs et contraintes, (e.g., (Berry et Castellanos, 2008) pour le texte, (Kosch et Maier, 2009) pour les images, (Pinquier et André-Obrecht , 2006) pour l'audio, (Snoek et Worring, 2005) pour les vidéos) et des métadonnées générées. Dans un système d'information multimédia, il n'est pas souhaitable d'exécuter tous les algorithmes d'indexation disponibles sur tous les contenus multimédias, car ceci provoquerais (i) la surcharge du système et (ii) l'extraction des métadonnées non pertinentes ou qui pourraient ne jamais être utilisées. Les projets suivants ont abordé ce problème de différentes manières.

Le projet SAPIR (Search on Audio-visual content using Peer-to-peer Information Retrieval) (Agosti et al., 2007), (Batko et al.,2010) utilise une architecture pair-à-pair (P2P), où tous les serveurs ont les mêmes fonctionnalités. Les contenus multimédias sont indexés au niveau du serveur où ils sont stockés. Les mêmes algorithmes d'indexation sont utilisés sur chaque serveur. Les métadonnées obtenues suite à l'indexation sont stockées sur le même serveur que le contenu.

Le projet DISCO[1] (Distributed Indexing and Search by Content) utilise aussi une architecture P2P (Boisson et al., 2008). Dans ce système les serveurs peuvent avoir leur propre manière de représentation des contenus. Comme dans tout système P2P,

[1] http://www.lamsade.dauphine.fr/disco/index

un index global est créé avec l'information représentative envoyée par chaque serveur. Le système utilise une représentation pivot pour l'index global, et chaque serveur réalise le mapping entre sa représentation et l'index global.

Le projet MODEST[2] (Multimedia Object Descriptors Extraction from Surveillance Tapes) propose un système de vidéosurveillance basé sur les multi-agents. Dans cette architecture, les algorithmes d'indexation sont développés comme agents qui peuvent communiquer entre eux pour résoudre leur tache (Abreu et al., 2000). Chaque serveur du système gère les contenus acquis par une seule caméra. Sur chaque serveur, un agent, qui s'occupe de la segmentation du contenu et l'identification des différents objets présents dans le contenu, est installé. D'autres agents installés sur un serveur dédié, mais qui peuvent aussi se déplacer, utilisent les descriptions obtenues par les autres agents pour réaliser des taches plus complexes, comme la classification des objets détectés (e.g, véhicule, camion, personne). Des informations statiques sur le contenu de chaque serveur sont utilisées pour détecter des comportements étranges et pour réaliser des statistiques. L'utilisateur peut seulement visualiser les anomalies qui sont remontées par les agents.

Le projet CANDELA[3] (Content Analysis and Network DELivery Architectures) se focalise sur l'analyse et la recherche du contenu vidéo dans plusieurs domaines (e.g., vidéosurveillance (Jaspers et al., 2005), vidéos personnelles (Pietarila et al., 2005)). L'architecture utilisée par le projet est composée par un serveur central qui contient tous les services mis à disposition de l'utilisateur, et où le contenu multimédia est téléchargé pour être indexé, et des serveurs qui n'ont comme but que de stocker les métadonnées générées par l'indexation (Petkovic and Jonker, 2005). Pour le stockage des métadonnées, une base de données distribuée est employée. Ces descripteurs sont extraits en appliquant un ensemble fixe d'algorithmes d'indexation développés dans le cadre du projet.

Le projet WebLab[4] propose une plateforme d'intégration pour les applications de gestion de contenus multimédias (Giroux et al., 2008). Du point de vue logique l'architecture proposée est composée de 3 couches. Pour assurer l'intégration de tout algorithme d'indexation, la plateforme les considère comme étant des services web qui doivent respecter l'interface proposée. Cette plateforme est utilisée par plusieurs autres projets, parmi lesquels nous citons : WebContent[5], e-Wok Hub[6], AXES[7].

Le projet VITALAS (Video & image Indexing and retrieval in the Large Scale)[8] utilise la plateforme WebLab pour intégrer des algorithmes d'indexation de contenus multimédias et les appliquer sur de grandes collections de documents (Viaud et al., 2008). Dans ce projet, les documents se trouvent sur des serveurs distribués, et sont indexés offline, avec un ensemble fixe d'algorithmes d'indexation.

---

[2] http://www.tele.ucl.ac.be/PROJECTS/MODEST/index.html

[3] http://www.hitech-projects.com/euprojects/candela

[4] http://weblab-project.org/

[5] http://www.webcontent.fr/

[6] http://www-sop.inria.fr/edelweiss/projects/ewok/

[7] http://www.axes-project.eu/

[8] http://vitalas.ercim.org/

Une comparaison de ces projets nous permet d'observer que tous utilisent un ensemble fixe d'algorithmes pour indexer les contenus multimédias sur des serveurs dédiés. Ces algorithmes sont exécutés au moment de l'acquisition du contenu, et donc la consommation des ressources n'est pas optimale. Ce problème est adressé par le projet LINDO, qui propose une architecture distribuée pour la gestion des contenus multimédias en favorisant la réduction de la consommation des ressources, en termes de transfert des données sur le réseau, de stockage et d'utilisation de la CPU. Dans certains de ces systèmes, un accès à tous contenus est accordé à l'utilisateur après une authentification réussie. Dans la plupart des domaines d'application un contrôle d'accès plus complexe est nécessaire. Dans la section suivante, nous détaillons les différentes stratégies de distribution contrôle d'accès.

2.2. La Distribution du Contrôle d'Accès

La gestion des informations confidentielles au sein d'un système décentralisé, nécessite des mécanismes de protection contre toute menace. Afin de garantir la pleine protection de ces informations l'accès doit être contrôlé à tous les niveaux. Plusieurs modèles et standards ont été définis: le modèle Discretionary Access Control (DAC) (Harrison et al., 1976), le modèle Mandatory Access Control (MAC) (NIST, 2006), le modèle Role Based Access Control (RBAC) (Ferraiolo et al., 1992), le standard XACML(OASIS, 2003). Dans cette sous-section, nous présentons le modèle RBAC et le standard XACML, qui sont largement utilisés dans la gestion du contrôle d'accès au sein des environnements distribués.

2.2.1. Le Modèle RBAC

La motivation principale autour de la proposition d'un modèle de contrôle d'accès à base de rôle (RBAC) était de faciliter l'administration des privilèges d'accès pour un grand nombre d'utilisateurs accédant à des ressources distribuées. La solution présentée par (Ferraiolo et al., 1992) était de regrouper les utilisateurs dans des rôles reflétant la structure organisationnelle de l'entreprise puis, de distribuer les permissions à ces rôles au lieu de le répéter par individu.

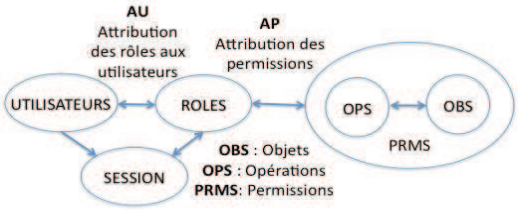


Figure 1. Le Modèle RBAC

Le rôle est le cœur du modèle RBAC et est vu comme une entité intermédiaire entre les utilisateurs et les permissions car il regroupe un ensemble de privilèges et les attribue, ensuite, aux utilisateurs en fonction de leur rôle.

Comme le montre la Figure 1, l'Attribution des rôles dans le modèle RBAC suit une relation mutuelle où un Utilisateur (personne, processus informatique, machine, etc.) peut jouer plusieurs rôles dans une seule session et un rôle peut être attribué à plusieurs utilisateurs.

AU ⊆ Utilisateurs x Rôles

L'Attribution d'un rôle garantira plusieurs permissions (PRMS) à l'utilisateur et une permission peut être attribuée à plusieurs rôles.

AP ⊆ Rôles x PRMS

La nature d'une permission décrit le type d'opérations (OPS) (e.g. lire, écrire, mettre-à-jour, etc.) autorisé sur les objets (OBS) (ressources de données : documents, processus informatique, machines, etc.).

La relation entre ces objets et les opérations attribuées est aussi mutuelle ; une opération peut être autorisée sur plusieurs objets et à un objet peuvent être attribuées différentes permissions.

PRM ⊆ OPS x OBS

Afin de répondre aux besoins évolutifs de la gestion d'accès au sein de l'entreprise, le modèle RBAC a été étendu à différents profils afin de combler les lacunes et atteindre une meilleure performance à travers différents principes tels que la hiérarchie des rôles dans RBAC-1 (Sandhu, 1996) où un utilisateur peut hériter des droits d'accès d'un autre utilisateur, l'inclusion des contraintes lors de l'attribution d'un rôle dans RBAC-2 et la séparation des tâches distinctes entre les différents intervenants d'une mission dans RBAC-3 (Kuhn et al., 1997).

2.2.2. Le Standard XACML

Le modèle RBAC a résolu le problème d'administration des ressources de données distribuées en les gérant d'une manière centralisée. Mais avec l'évolution des architectures orientées-service et les services web, le problème de la gestion d'accès devient plus compliqué car les politiques de contrôle d'accès sont devenues également distribuées et parfois gérées par différents administrateurs. Pour résoudre ce problème, le standard XACML a été introduit par (OASIS, 2003).

XACML (eXtensible Access Control Markup Language) est un standard basé sur XML qui décrit des politiques de contrôle d'accès permettant de définir les privilèges des utilisateurs sur les ressources informatiques d'un système. Ce standard permet d'authentifier et de sécuriser les systèmes en prenant en compte différents éléments reliés au contexte de l'utilisateur.

La spécification XACML fournit une architecture qui décrit le processus de gestion d'accès lors d'une demande d'accès. Quand un utilisateur demande d'accéder une ressource, un Point d'imposition de politique de sécurité PEP (Policy Enforcement Point) interfère pour vérifier si l'accès est autorisé ou non. Afin de vérifier la validité d'une demande d'accès, le système doit vérifier s'il existe une politique de sécurité qui correspond à cette demande. Cette vérification est réalisé

par le PEP qui crée une requête contenant les attributs de l'utilisateur et les envoie au Point de décision de politique de sécurité PDP (Policy Decision Point) qui prend la décision en consultant la liste des politiques d'accès qui sont localisées dans les Magasins de politiques d'accès PAPs (Policy Administration Points). En utilisant la politique de sécurité pertinente (choisi par le PDP), le PEP retourne la réponse appropriée au client et assure que cette décision est respectée et que le client ne peut accéder qu'aux ressources autorisées.

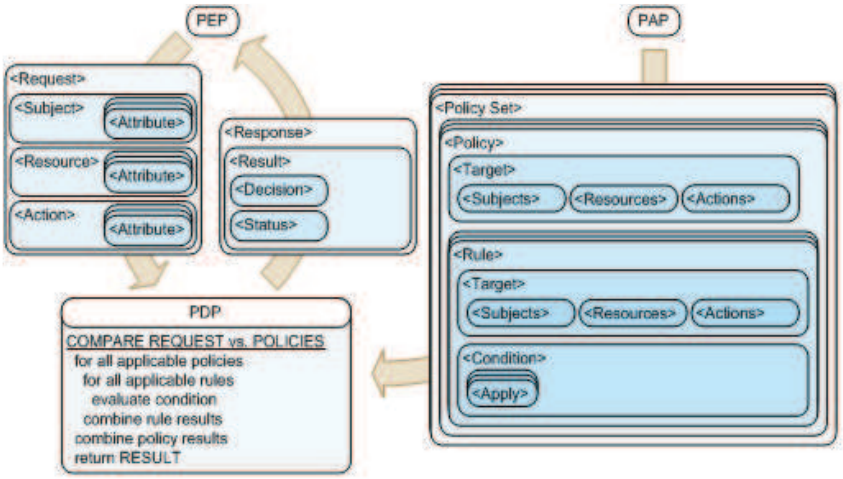


Figure 2. Les Documents XACML et leurs relations

Le standard XACML est composé par quelques types de documents :

- La Requête est composée des informations concernant le sujet qui veut exécuter une certaine action sur une ressource ;
- La Réponse fournit des informations sur la demande d'accès ;
- L'Ensemble des Politiques et composé par un ensemble de règles qui restreignent l'accès aux ressources dans certaines conditions.

Figure 2 montre la structure de ces documents et leur utilisation dans le workflow présenté dans le paragraphe antérieur. XACML est considéré comme un standard efficace de par sa capacité à gérer les droits d'accès d'une manière distribuée en prenant en compte le contexte de l'utilisateur ou le service. Un profil XACML RBAC a été introduit par (OASIS, 2005), en favorisant ainsi la portabilité du standard vers des services à grande échelle.

2.3. La prise en compte du contexte et de la situation dans le control d'accès

Un autre aspect important qui doit être pris en compte dans la gestion de contrôle d'accès est le contexte et la situation de l'utilisateur au moment où il accède au système, afin de fournir un accès adapté aux besoins de l'utilisateur.

Les utilisateurs sont de plus en plus mobiles et ils accèdent au système généralement depuis leurs appareils mobiles. Comme conséquence, leurs contexte (e..g., localisation, temps) est très dynamique. Donc, les droits d'accès a un contenu ne doivent pas dépendre que de l'identité de l'utilisateur, mais aussi de où l'utilisateur se trouve, et de la situation de l'utilisateur et de son environnement.

Beaucoup de travaux ont été proposé des extensions du modèle RBAC afin de prendre en compte notamment le contexte : Temporal RBAC (Bertino et al., 2001), Spatial RBAC (Hansen and Oleshchuk, 2003), Geo-RBAC (Bertino et al., 2005), Dynamic Role Based Access Control model (Zhang and Parashar, 2003), Context-Role Based Access Control model (Park et al., 2006).

En appliquant ces modèles, dans certains cas, l'utilisateur peut ne pas retrouver l'information nécessaire, même si elle est disponible dans le système, parce qu'il n'a pas les droits pour y accéder selon son contexte. Pour éviter ce genre de situation, surtout dans des domaines a risque ou des situations d'urgence peuvent apparaitre, le système doit pouvoir prendre en compte aussi la situation de l'utilisateur.

Le besoin de la prise en compte de la situation a été identifié premièrement par (Povey, 1999). L'auteur a souligné l'importance d'un schéma d'accès qui assouplit les règles d'accès pour les utilisateurs dans des situations exceptionnelles (e.g., urgences médicales). Notamment pour les systèmes médicaux, une solution encore plus flexible a été proposée par (SPC, 2004) : « Bris-de-glas ». Dans une situation d'urgence, les utilisateurs ont accès à tous les documents, même à ceux qu'ils ne sont pas normalement autorisés à consulter.

Le « Bris-de-glas » est une solution extrême, qui viole toutes les règles d'accès au système et qui autorise les utilisateurs à exécuter des actions interdites. Plusieurs travaux se sont concentrés sur l'amélioration de control d'accès pour le « Bris-de-glas » ou sur la proposition d'autres solutions moins dangereuses de prise en compte de la situation : (Ferreira et al., 2009), (Kawagoe and Kasai, 2011).

L'analyse de ces travaux nous même a la conclusion que plus une approche est flexible plus il y a de risques de violation de la sécurité du système. Un bon compromis doit être trouvé entre la flexibilité et la sécurité du système.

2.4. Contrôle d'Accès aux Ressources Multimédias

Les projets mentionnés dans la Section 2.1 se concentrent sur l'indexation et la recherche des contenus multimédias, mais aucun d'entre eux ne considère les problèmes concernant la gestion du contrôle d'accès et la protection de l'anonymat.

Toutefois, beaucoup de solutions pour la sécurisation de l'accès aux bases de données et systèmes multimédias ont été proposées. Certains auteurs se sont intéressés à la sécurité de la connexion aux systèmes et de la distribution des contenus (Sánchez et al., 2006), tandis que d'autres se sont concentrés sur le contrôle d'accès basé sur le contenu multimédia en définissant des restrictions à un niveau de granularité fine (El-Khoury, 2006).

(Chen et al., 2004) propose un framework qui adresse le contrôle d'accès multimédia multi-niveaux en adoptant les technologies suivantes : RBAC, XML et Bases de données relationnelles orientées objet. Les auteurs associent des rôles aux

utilisateurs, adresses IP, objets et intervalles horaires. Tous les contenus multimédias gérés par le système doivent être segmentés. Seuls les objets ayant des rôles associés sont extraits des contenus multimédias. Le système permet le stockage de plusieurs versions des contenus multimédias, l'original et une pour chaque restriction concernant un utilisateur.

(Thuraisingham et al., 2006) adresse les problèmes de confidentialité et anonymat dans le contexte d'un système de vidéo surveillance. Les auteurs on aussi défini des droits d'accès aux différents objets hiérarchiques qui peuvent être extraits des contenus multimédias. Ils se sont concentrés sur la détection des événements anormaux.

Dans le contexte de systèmes pervasifs, où les contenus et les utilisateurs sont mobiles, la gestion de la consommation des ressources, le temps de réponse et la qualité des résultats retournés sont très importants. Le contrôle d'accès ne simplifie pas ces questions. A notre connaissance, il n'y a pas de système qui réponde à tous ces problèmes. Afin d'offrir une meilleure réponse aux besoins de l'utilisateur, le système doit levier entre la consommation des ressources, les résultats retournés et la sécurité. Le projet LINDO offre une solution pour une consommation optimale des ressources, mais il ne s'intéresse pas au contrôle d'accès: il accorde un accès complet à tous les contenus et les ressources pour tout le monde après l'authentification.

3. L'approche LINDO

3.1. L'Architecture Système

Le projet européen LINDO a pour but de développer une architecture générique, dans laquelle non seulement le stockage des données multimédia est distribué, mais également l'indexation qui est répartie sur différentes unités de stockage, éventuellement hétérogènes, éloignées géographiquement et de capacités diverses.

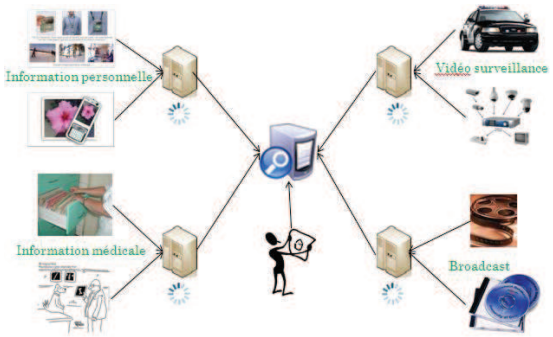


Figure 3. Les défis du traitement d'une requête d'un utilisateur

Plutôt que de déplacer les contenus et les métadonnées vers les serveurs de traitement, une solution alternative a été considérée où les routines d'indexation pertinentes sont exécutées sur les sites distants. En conséquence, seule la stricte information nécessaire répondant à une requête sera transférée à l'utilisateur, voir Figure 3. Pour une présentation plus détaillée de l'architecture LINDO, voir (Brut et al., 2011a).

Cette stratégie distribuée de l'indexation et du stockage des contenus multimédias et de leurs métadonnées est avantageuse car elle vise à éviter de nombreux inconvénients du traitement centralisé tels que :

1. La lenteur de traitement d'une requête : le traitement d'une requête sur la totalité des métadonnées du système a de fortes chances de surcharger le serveur central, surtout lors du traitement de requêtes complexes et lorsque plusieurs requêtes sont traitées simultanément ;

2. La surcharge de la bande passante : tous les contenus multimédias ou toutes les métadonnées doivent être transférés par le réseau au serveur central ;

3. La centralisation du système : si le serveur central ne répond plus, l'ensemble des métadonnées doit de nouveau être recalculé et renvoyé sur un serveur central. De plus, dans le cas d'un système d'information dynamique, la mise-à-jour du serveur central serait très coûteuse ;

4. Le non-respect des droits d'accès sur les données : certaines métadonnées ne doivent pas être stockées sur le serveur central pour des raisons, par exemple, de respect de la vie privée.

3.2. La Fonctionnalité du Système

Les fonctionnalités adoptées dans le système présenté dans la section précédente ont : le contenu est acquis et stocké sur les serveurs distants et la collection d'algorithmes d'indexation est stockée et gérée au niveau du serveur central. Cette collection est variable, à n'importe quel moment nous pouvons intégrer de nouveaux algorithmes avec différentes fonctionnalités, contraintes d'exécution et performances.

3.2.1. Le Mécanisme d'Indexation

Afin de réduire la consommation des ressources, l'indexation des contenus multimédias est réalisée au moment de l'acquisition (i.e., indexation implicite) avec des algorithmes génériques (e.g., détection des personnes, détection de la couleur dominante) et au moment de la requête (i.e., indexation explicite) avec des algorithmes qui vont analyser le contenu multimédia plus en détail (e.g. reconnaissance de personnes, reconnaissance du numéro de plaque d'immatriculation). Cela évite d'exécuter tous les algorithmes disponibles dans le système d'un coup et de générer des métadonnées qui ne seront peut-être jamais utilisées mais pose de problèmes des droits d'accès qui concernent l'indexation

explicite. Les Figure 4 et Figure 5 présentent des exemples qui montrent la différence du niveau de détail atteint par l'indexation implicite et explicite. Ces algorithmes sont classés en fonction de deux contextes d'exécution (à l'intérieur et à l'extérieur).

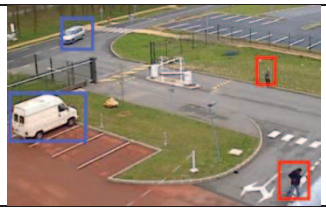
	A l'intérieur	**A l'extérieur**
Intrusion	- Présence de personnes	- Présence de personnes et de véhicules
Comptage	- Nombre de personnes - Couleur dominante de la partie supérieure du corps	- Nombre de personnes, nombre de véhicules - Couleur dominante de la partie supérieure - Couleur dominante du véhicule
		

Figure 4. Exemple d'information extraite par les algorithmes d'indexation implicites

	A l'intérieur	**A l'extérieur**
Intrusion	- Présence de personnes	- Présence de personnes et de véhicules
Comptage	- Nombre de personnes - Couleur dominante de la partie supérieure du corps - Reconnaissance de visage - Reconnaissance de voie et transcription de la parole	- Nombre de personnes, nombre de véhicules - Couleur dominante de la partie supérieure du corps - Couleur dominante du véhicule - Reconnaissance de la plaque minéralogique - Reconnaissance de visage
		

Figure 5. Exemple d'information extraite par les algorithmes d'indexation explicits

3.2.2. Le Mécanisme de traitement de la requête

Le traitement de la requête (illustré dans la Figure 6) commence avec la saisie de la requête au niveau du serveur central. D'abord, la requête est traitée et exécutée sur la collection de métadonnées sur le serveur central (cette collection est un résumé des collections de métadonnées sur les serveurs distants) afin de sélectionner les serveurs distants qui pourraient fournir des réponses à la requête et elle est envoyée aux serveurs sélectionnés.

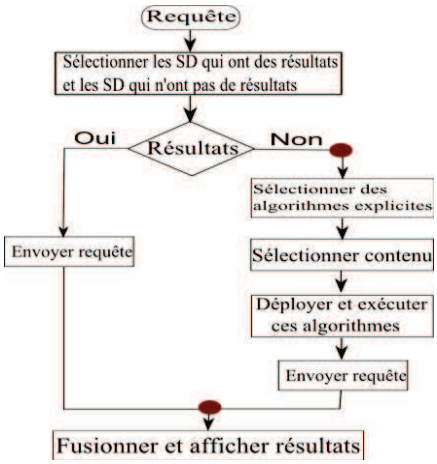


Figure 6. Diagramme du traitement de la requête

Parmi les serveurs qui n'ont pas été sélectionnés, il est possible d'avoir des serveurs qui contiennent de l'information pertinente mais qui n'a pas été indexée avec les bons algorithmes. Pour cette raison, la solution LINDO détecte ces algorithmes supplémentaires (Brut et al., 2011b) et les exécute (i.e., indexation explicite) sur une sous collection de contenus multimédias. Tous les résultats obtenus des serveurs distants sont envoyés au serveur central, où ils sont fusionnés et affichés à l'utilisateur.

4. La mise-en-place d'une Couche du Contrôle d'accès à l'architecture LINDO

La sensibilité du contenu des ressources de vidéo surveillance et la loi d'anonymat définie sur le contenu justifie l'application d'un contrôle d'accès qui gère et personnalise l'accès selon le rôle de l'utilisateur. Cette couche est chargée de gérer :

1. Les droits d'accès des utilisateurs/services aux ressources de données qui varient non seulement en fonction de leur rôle mais aussi en fonction de leur contexte (temps, localisation, etc.) ;

2. Les droits d'accès pour l'utilisation des algorithmes explicites : le risque de dévoiler des informations personnelles ou confidentielles s'élève avec le niveau de granularité de détail recherché et fourni par l'algorithme.

Nous surlignons que dans le contexte de la prise en compte de la couche du contrôle d'accès, l'absence de réponse à une requête peut intervenir non seulement à cause du manque de résultats ou de ressources pertinentes mais aussi à cause des restrictions d'accès imposées par la couche de sécurité.

4.1. Le système LINDO vu comme un SIP

L'objectif est de trouver des solutions qui peuvent assurer plus d'accessibilité aux ressources demandées à n'importe quel moment, depuis n'importe où et n'importe comment.

Dans le système LINDO, la complexité d'obtention des résultats est souvent due à des contraintes liées au contrôle d'accès, ce qui constitue un cas d'application idéal pour valider notre proposition en particulier si l'on prend en compte les caractéristiques pervasives de LINDO telles que :

- la distribution des ressources de données ;
- la variation des autorités gérant ces ressources ;
- la nature évolutive de ces ressources (générées en temps réel) ;
- la sensibilité et la confidentialité du contenu de ces ressources ;
- la richesse des informations contextuelles ;
- la distribution des algorithmes d'indexation ;
- l'exécution des demandes d'accès en temps réel ;
- l'importance d'obtenir des solutions réactives lors de situations critiques.

4.2. L'adaptation du contrôle d'accès afin de confronter les challenges d'accès

La gestion d'accès devient plus compliquée dans un environnement pervasif à cause de la dynamique des contraintes contextuelles et situationnelles d'un utilisateur mobile. Notre objectif est d'assurer l'efficacité du processus de recherche d'information malgré les challenges de sécurité. Afin de parvenir à cet objectif, nous employons PSQRS (Pervasive Situation-aware Query Rewriting System) (Al Kukhun et Sèdes, 2008)– un système de prise de décision adaptatif qui confronte les refus d'accès générés lors d'une situation du temps réel par la réécriture des requêtes afin de trouver des solutions d'accès à des ressources alternatives.

Comme précisé dans l'introduction de la Section 4 lorsqu'on applique une couche de contrôle d'accès au système LINDO, la manque de résultats retournés à la requête utilisateur peut être due au fait qu'il existe pas de résultats dans le système mais aussi aux restrictions imposés par la couche de sécurité. Ce manque de résultats peut empêcher l'utilisateur à réaliser sa tache. Afin de dépasser cette limite, nous proposons une relaxation du contrôle d'accès respecte les règles d'accès qui composent la politique de sécurité du système et applique la prise de décision adaptative qui concerne deux fonctionnalités :

1) L'exécution des algorithmes d'indexation explicites et la sélection des algorithmes

2) Le filtrage et la visualisation des contenus multimédias

Plus précisément, dans les deux points rouges illustrés dans la Figure 6, une fonction de matching est exécutée afin d'établir en fonction du profil utilisateur et de sa situation s'il a le droit d'accéder le contenu :

F : (Up; Situation; Policy) => Permit/Deny;

ou Up est le profil utilisateur, la Situation est un niveau d'urgence (explicitement fournit par l'utilisateur ou inféré à partir de la localisation et du contexte) et Policy est un ensemble de règles qui définit la politique de sécurité du système.
Le profil utilisateur est définit de la façon suivante:

Up =< Uid; Name; Login; Password; RoleId >

ou Uid est le id de l'utilisateur, Name est le nom de l'utilisateur, Login est son login, Password est le mot de passe de l'utilisateur et RoleId représente le rôle qui est associé à l'utilisateur.
Une règle d'accès est définie de la façon suivante:

Rule =< RuleId;RoleId; Action;Context; Permission >

Une règle définit une certaine permission (Permit ou Deny) pour un certain rôle (i.e, RoleID) et Action (e.g. indexation explicite, visualisation de l'objet) dans un certain contexte (e.g., localisation).

Dans la suite, nous introduisons la fonctionnalité détaillée de PSQRS.

4.3. L'architecture PSQRS

Comme présenté en Figure 7, l'architecture que nous proposons vise à étendre le modèle de prise de décision dans XACML en ajoutant une couche adaptative réalisée par un mécanisme de réécriture de la requête de l'utilisateur dans le cas où elle est refusée par le PDP (Al Kukhun et Sèdes, 2008).

Le système récupère les contraintes contextuelles de l'utilisateur et les reçoit avec une étape d'authentification (1). Puis, quand l'utilisateur lance sa requête, le système la communiquera au générateur de requête (2) qui la traduira vers une requête R de format XACML. Celui-ci prend en compte cette demande et la combine avec les contraintes contextuelles puis l'envoie vers l'Evaluateur de requêtes (3) qui joue le rôle d'un PDP et suit le processus normal de XACML.

Selon les droits d'accès de l'utilisateur (précisés par les politiques d'accès sauvegardées dans les PAPs – voir le schéma XACML en section 3.3), le système répond à cette demande soit en permettant l'utilisateur d'accéder à la ressource demandée (4a), ou en lui répondant avec un refus d'accès (4b). C'est dans ce dernier cas que notre mécanisme adaptatif intervient pour étudier la situation dans laquelle l'utilisateur a consulté le système. Cette situation est définie par le Fournisseur de Sensibilité de la Situation (5 et 6) qui autorise la régénération de la requête R' dans le cas d'une situation d'urgence par exemple.

Cette régénération ou réécriture de requêtes est réalisée grâce au Fournisseur de similarité (7 et 8) qui prend les contraintes contextuelles de l'utilisateur comme un point de départ pour la recherche des documents ou des services autorisés et qui ont des similarités de contenu ou de fonctionnement avec la ressource initialement demandée par l'utilisateur et qui été jugé comme non autorisée.

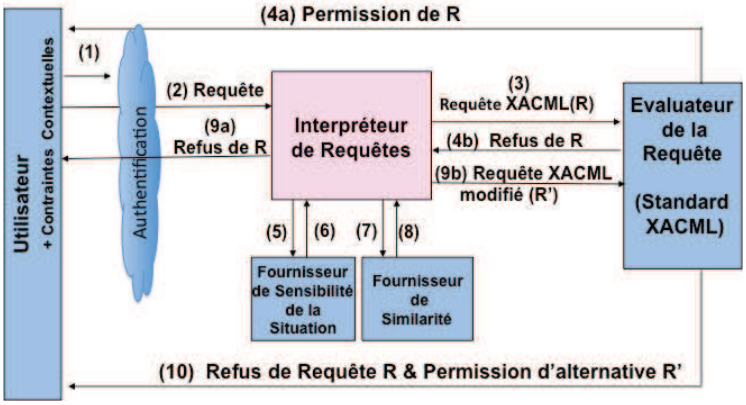


Figure 7. L'architecture de PSQRS

Cette étape est destinée à restituer à l'interpréteur de requêtes des ressources alternatives similaires. Dans le cas où le système n'aura pas des propositions, l'interpréteur des requêtes va envoyer à l'utilisateur un refus d'accès (9a) – cas classique. En revanche, dans le cas où le système trouve des ressources similaires, le Générateur de la Requête réécrit la requête initiale en remplaçant la ressource demandée par la nouvelle ressource jugée similaire (par notre Fournisseur de Similarité) puis l'envoie vers l'Evaluateur de Requêtes qui réévalue la nouvelle requête R' (9b) et répond l'utilisateur avec un refus d'accès pour sa demande initiale et une permission pour accéder aux ressources alternatives (10).

Dans la suite nous présentons plus en détail la façon dont nous avons intégré l'approche PSQRS dans le système LINDO.

4.4. La fusion PSQRS et du système LINDO

Afin d'inclure la couche de contrôle d'accès dans l'architecture LINDO, plusieurs changement doivent être faites à l'intérieur du système. Plus précisément, d'un point de vue fonctionnel, chaque fois qu'une demande d'accès est faite (les points rouges de la Figure 6), le système PSQRS est utilisé avant d'exécuter l'indexation explicite et avant d'afficher les résultats. D'un point de vue architecturel, l'architecture du système présentée en détail dans (Brut et al. 2011) a été étendue avec plusieurs modules et fonctionnalités.

En conséquence, dans la Figure 8, les modules qui ont été ajoutés ou modifiés sont affichés en rouge. La plupart des changements se passent au niveau du serveur central :

- Le module d'authentification (Authentification Module) a été ajouté pour vérifier les utilisateurs qui essayent d'interroger le système. Ce module communique avec le module Metadata Engine afin de déterminer si l'utilisateur est «reconnu » par le système et de retrouver son profil et son contexte ;

- Le module Access Control a été ajouté. Ce module contient les sous modules PSQRS Query Interpreter (Interpréteur de requêtes), Sensitivity Analyser (Fournisseur de Sensibilité de la Situation) et Similarity Provider (Fournisseur de la similarité). Toute information qui est envoyée à l'utilisateur passe par ce module afin d'appliquer une éventuelle adaptation du contrôle d'accès ;

- Le module Terminal Interface a été modifié afin de capturer le contexte et la situation de l'utilisateur ;

- une base de données avec les rôles RBAC, les règles et les profils des utilisateurs a été incluse dans le module Metadata Engine. Cette base de données contient aussi les règles d'adaptation d'accès ;

- Le module Query Analyser a été integré dans le Request Processor ;

- Dans la structure du Remote Server, seulement le module Access Manager a été modifié. En effet, une nouvelle fonctionnalité a été ajouté à ce module : l'exécution des algorithmes d'indexation ;

Pour chaque rôle RBAC identifié, les droits d'accès aux contenus multimédias, et à l'exécution des certains algorithmes d'indexation sont spécifiés en concordance avec le contexte de l'utilisateur au moment ou il interroge le système. La situation de l'utilisateur est capturée de façon implicite (en analysant le contexte) ou de façon explicite (spécifiée par l'utilisateur). Afin de fournir des alternatives à l'utilisateur en fonction de sa situation, le module Similarity Provider peut sélectionner d'autres contenus ou exécuter des algorithmes qui extraient des informations du contenu ou modifient le contenu afin de respecter la politique de sécurité du système.

5. Illustration dans un système de vidéosurveillance

Dans cette section, nous présentons un exemple où notre proposition est utilisée pour pallier le manque des réponses restituées par le système. Comme nous allons l'illustrer, le système va modifier le traitement de la requête et l'adaptation de prise de décision d'accès selon le niveau d'importance de la situation.

Ce cas d'utilisation concerne une compagnie de transport en commun qui a installé des cameras de vidéo surveillance dans les bus et rames de métro, dans les stations et à coté des machines aux billets. Le système qui gère ces cameras peut être utilisé par les agents de sécurité et le policier. En conséquence on peut identifier deux rôles : agent de sécurité et policier. L'accès au système peut être fait depuis la chambre de contrôle, ou depuis les stations en utilisant un dispositif mobile. Pour chaque rôle un ensemble de restrictions peuvent être spécifiées.

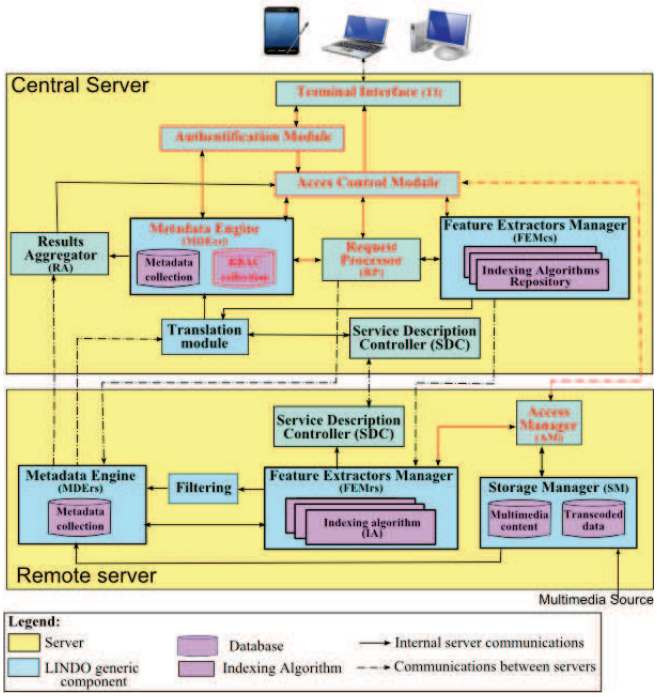


Figure 8: L'architecture LINDO avec l'approche PSQRS incorporée

Tableau 1: Exemples de droits d'accès

Rôle	Contexte de l'utilisateur	Contenus	Action	
			Voir les visages	Indexation explicite
Agent de sécurité	Chambre de contrôle	Tous	Permettre	Permettre
	Station	Cameras métro	Refuser	Permettre, seulement Object Tracking
		Cameras bus	Refuser	Refuser
Policier	Chambre de contrôle	Tous	Permettre	Permettre
	Station	Cameras métro	Permettre	Permettre, Object et Person Tracking
		Cameras bus	Permettre	Refuser

Le Tableau 1 fournit quelques exemples de droits d'accès donnés à chaque rôle, dans un certain contexte pour réaliser une certaine action sous des contenus multimédias.

Scénario : En prenant le métro de la station Trocadéro vers la place d'Italie à 14h15, Hélène a oublié son sac rouge sur un banc d'attente sur un quai. Dès qu'elle s'en est rendue compte, elle est sortie et s'est rendue au guichet de la station pour signaler le problème.

Le traitement typique d'une telle situation passe par l'agent de service clientèle qui ouvre un dossier, prend les descriptifs de l'objet perdu et les transmet à l'agent de sécurité sur place. Ce dernier va suivre différentes étapes pour retrouver l'objet : il va vérifier si l'objet a été déjà retrouvé ou remis au service par quelqu'un. Sinon, il va essayer de consulter le système de vidéo surveillance pour vérifier si l'objet est toujours au même endroit.

5.1. Le traitement typique d'une requête selon LINDO

La Figure 9 montre l'interprétation typique réalisée par le système de recherche d'information fourni par le système LINDO. La requête lancée sera traitée et parcourue afin d'extraire les mots-clés qui ensuite seront reformulés sous forme d'une requête XML.

Après l'extraction des mots-clés de la requête, le traitement de la requête va procéder à la localisation des serveurs gérant les différents flux capturés par les caméras situées dans les quais d'attente de la station Trocadéro. Puis, une étape de filtrage sera effectuée pour restreindre la recherche dans les parties acquises entre 14h00 et 15h00. Le système va déterminer, ensuite, une liste d'algorithmes d'indexation appropriée à l'ensemble des besoins, des propriétés et des contextes exprimés dans la requête. Cette étape va générer les métadonnées liées à la requête.

Query: Trouve toutes les vidéos qui contiennent un sac rouge, oublié à l'arrêt de métro Trocadéro, Paris jeudi 2 février, entre 2:00pm et maintenant (3:00pm).

Dans ce scénario, les informations demandées sont basiques, la requête sera traitée à l'aide des résultats d'indexation réalisés par les algorithmes implicites placés au niveau du serveur central. Le système va poursuivre la recherche pour trouver un objet rouge dans les métadonnées décrivant les segments choisis.

Un processus de filtrage additionnel est appliqué pour la prise en compte des règles de contrôle d'accès. En examinant les droits d'accès de l'agent de sécurité, nous trouvons qu'il n'a pas l'autorisation de consulter des vidéos qui affichent les visages des passagers, ni d'utiliser les algorithmes d'indexation explicites existant au niveau des serveurs distants. Par conséquent, le système filtre les ressources en éliminant les parties qui contiennent des visages de personnes et enfin, renvoie à l'utilisateur des segments qui contiennent un objet rouge (s'il en existe).

5.2. Traitement adaptatif sensible au contexte et à la situation en utilisant PSQRS

L'analyse des résultats restitués à l'agent de sécurité dans ce cas, montre que ces derniers sont insuffisants. Notre proposition intervient à ce niveau afin d'améliorer

la qualité de service et d'offrir à l'utilisateur plus de ressources accessibles sans dépasser les droits d'accès imposés sur la consultation des ressources de données.

```
<UserQuery>
    <QueryInText> Trouve toutes les vidéos qui contiennent un sac rouge, oublié à l'arrêt de métro Trocadéro, Paris jeudi 2 février, entre 2:00pm et maintenant (3:00pm).</QueryInText>
    <MediaLocation> métro Trocadéro, Paris </MediaLocation>
    <MediaFormat>vidéo</MediaFormat>
    <TimeSpan>
        <From>2012-02-02T14:00:00</From>
        <To> 2012-02-02T15:00:00</To>
    </TimeSpan>
</UserQuery>
```

Figure 9. Structure XML d'une requête

L'utilisation de l'architecture PSQRS permettra au système de modifier le niveau d'accessibilité et d'adapter les permissions offertes à l'agent de sécurité selon son contexte et l'importance de la situation de consultation.

L'utilisation de cette solution est liée au déclenchement de la reconnaissance d'une situation ou d'un contexte par le système. Dans ce scénario, la situation sera reconnue depuis l'identifiant du dossier « objet perdu ».

L'implémentation adaptative de notre proposition est réalisée par le système PSQRS qui adapte la prise de décision par la réécriture des requêtes XACML. Cette solution a prouvé son efficacité par sa capacité à fournir une prise de décision d'accès à partir des politiques distribuées à prendre en compte des éléments contextuels liés à la requête.

Par conséquent, cette simple requête lancée par l'agent de sécurité (composée par des mots-clés décrivant le contenu recherché, voir Figure 9) sera incluse dans une demande d'accès sous forme d'une requête XACML plus structurée et enrichie des métadonnées (décrivant les contraintes contextuelles de l'utilisateur, son rôle, l'importance de la situation dans laquelle il consulte le système, voir Figure 10).

Le niveau d'importance de la situation va servir à déterminer le niveau d'adaptation qui sera réalisé ensuite. L'activation du mode de recherche adaptatif sera communiquée à partir de la réponse XACML sous la forme d'une « obligation » qui accompagne la réponse, voir Figure 11.

```
<Requestxmlns="urn:oasis:names:tc:xacml:2.0:context:schema:os"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:context:schema:os
http://docs.oasisopen.org/xacml/access_control-xacml-2.0-context-schema-os.xsd">
    <Subject>
        <AttributeAttributeId="urn:oasis:names:tc:xacml:2.0:subject:subject-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
            <AttributeValue>John Smith</AttributeValue>
        </Attribute>
        <AttributeAttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
DataType="http://www.w3.org/2001/XMLSchema#anyURI">
            <AttributeValue>Agent de sécurité</AttributeValue>
        </Attribute>
        <Attribute
AttributeId="urn:oasis:names:tc:xacml:2.0:example:attribute:securityAgent-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
            <AttributeValue>sa2023</AttributeValue>
        </Attribute>
    </Subject>
    <Resource>
        <ResourceContent>
            <UserQuery>
                <QueryInText> Trouve toutes les vidéos qui contiennent un sac rouge, oublié
à l'arrêt de métro Trocadéro, Paris jeudi 2 février, entre 2:00pm et maintenant
(3:00pm).</QueryInText>
                <MediaLocation> métro Trocadéro, Paris </MediaLocation>
                <MediaFormat>vidéo</MediaFormat>
                <TimeSpan>
                    <From>2012-02-02T14:00:00</From>
                    <To> 2012-02-02T15:00:00</To>
                </TimeSpan>
            </UserQuery>
        </ResourceContent>
    </Resource>
    <Action>
        <AttributeAttributeId="urn:oasis:names:tc:xacml:2.0:action:action-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
            <AttributeValue>Lire</AttributeValue>
        </Attribute>
    </Action>
    <Environment>
        <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:environment:environment-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
            <AttributeValue>Situation</AttributeValue>
        </Attribute>
        <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:environment:situation-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
            <AttributeValue>Objet oublié</AttributeValue>
        </Attribute>
        <Attribute AttributeId="urn:oasis:names:tc:xacml:2.0:environment:sitLevel-id"
DataType="http://www.w3.org/2001/XMLSchema#string">
            <AttributeValue>1</AttributeValue>
        </Attribute>
    </Environment>
</Request>
```

Figure 10. Requête XACML englobant la requête de l'utilisateur, son rôle, son contexte et sa situation

```
<Response>
  <Result>
    <Decision>Deny</Decision>
    <Status>
      <StatusCode Value="urn:oasis:names:tc:xacml:2.0:status:ok"/>
    </Status>
    <Obligation FulfillOn="Deny" ObligationId="ApplyAdaptiveQueryingMode">
      <AttributeAssignment AttributeId="AQM"
            DataType="http://www.w3.org/2001/XMLSchema#string"> On
      </AttributeAssignment>
    </Obligation>
  </Result>
</Response>
```

Figure 11. Réponse XACML avec les obligations à réaliser

Le déclenchement du mode adaptatif (Adaptive Querying Mode) va changer le processus du traitement de la requête afin d'assurer la réussite de la recherche en proposant des solutions adaptatives.

Cette solution est réalisée dans le système PSQRS au niveau du Fournisseur de Sensibilité de la Situation qui détecte la situation puis, s'oriente vers le Fournisseur de Similarité pour réaliser la réécriture de la requête (c.f. Figure 7).

Dans le cas où la situation de consultation est normale (sitLevel-id = 0 dans la requête XACML), le système réalisera une reformulation sémantique des mots-clés de la requête en utilisant des mots similaires ou des concepts plus génériques au niveau de Fournisseur de similarité. Un travail similaire a été introduit dans (Al Kukhun et Sèdes, 2008), l'objectif étant d'augmenter les chances de restitution des résultats aux utilisateurs malgré les challenges de sécurité.

La reformulation sémantique peut être réalisée avec l'aide d'un dictionnaire lexical standard tel que WordNet . Par exemple, le mot « sac » peut être remplacé par différents synonymes {bagage, cabas, sacoche, etc.}. L'emploi de la reformulation a été également proposé par d'autres travaux de l'équipe (Brut et al., 2011a)

Au niveau du traitement du scénario courant, le niveau d'importance de la situation est plus élevé (sitLevel-id = 1 dans la requête XACML présentée dans la Figure 11). De ce fait, le Fournisseur de Similarité sera remplacé par un Fournisseur de Solutions Adaptatives. Ce composant va réaliser une adaptation automatique ou assister l'utilisateur pour adapter sa requête en lui fournissant des propositions de solutions adaptatives sauvegardées dans une base de données prédéfinie. Le tableau 1 montre des exemples de solutions proposées par le système.

L'alimentation de la base de données peut aussi être effectuée par une méthode d'apprentissage automatique à partir des solutions proposées par les utilisateurs en fonction des situations rencontrées en temps réel. La réussite de telles solutions adaptatives ou alternatives (proposées par les utilisateurs) sera plus probable si on connaît la cause d'un refus d'accès. Les messages d'erreur qui accompagnent souvent les réponses négatives retournées peuvent servir d'indicateurs pour trouver des solutions alternatives.

Tableau 2. Exemples de solutions adaptatives proposées par notre système

Problème	Solution Adaptative
Loi d'anonymat imposée au contenu des ressources vidéo capturées	
Visage non-autorisé	Afficher le contenu après l'emploi d'un algorithme qui floute les visages.
Voix non-autorisée	Utiliser un algorithme de transcription textuelle « speech-to-text ».
Volume de vidéo	
Manque de capacité de stockage sur la machine de l'utilisateur.	Utiliser un algorithme de compression ou de conversion vers un format plus léger.
Format non supporté par la machine.	Utiliser un algorithme de conversion vers un format compatible.
Difficulté de téléchargement due à la faiblesse de la bande passante du réseau.	Utiliser un algorithme de synthèse du contenu de vidéo ou héberger les ressources et les consulter à partir d'un espace externe « Cloud computing ».

Par conséquent, la solution adaptative pour cet exemple va modifier le processus du traitement et va : (i) négliger l'étape de filtrage chargée d'imposer les contraintes du contrôle d'accès et (ii) la remplacer par une étape adaptative liée à la présentation de ressources ayant du contenu non-autorisé.

En appliquant ce processus au scénario décrit précédemment, le système va restituer - s'ils existent- les segments vidéo capturés dans la station Trocadéro entre 14h00 et 15h00 et qui contiennent un objet rouge.

Ces résultats seront classifiés de façon à détecter les parties non autorisées (contenant des visages de personnes) et c'est là que le système appliquera un processus de filtrage qui adapte l'affichage pour qu'il soit conforme aux restrictions d'accès imposées par le système.

L'adaptation de présentation consistera à la détection des visages puis à l'utilisation d'un algorithme qui floute les visages apparaissant dans ces segments afin de les présenter à l'utilisateur en respectant les règles d'accès.

6. Conclusion

Le projet LINDO a introduit une architecture de gestion des données multimédias distribuées qui facilite la consommation réduite des ressources en implémentant une technique d'indexation différée et distribuée. Dans cet article, nous avons montrée comment il a été étendu par une couche de contrôle d'accès adaptatif afin d'assurer l'accès aux ressources dans des contextes pervasifs (dans lesquels l'utilisateur pourra accéder aux ressources de données à n'importe quel moment, depuis n'importe où et n'importe comment).

Afin d'atteindre notre objectif, nous avons employé une solution adaptative du contrôle d'accès sensible au contexte et à la situation de consultation. La solution surmonte les refus d'accès survenus suite à des requêtes utilisateurs en modifiant le processus de traitement des requêtes et en proposant des solutions adaptatives pour contourner l'effet des politiques de contrôle d'accès. La réécriture de la requête est réalisée en utilisant l'architecture PSQRS qui effectue la prise de décision en se basant sur le modèle RBAC et la norme XACML. Notre solution de contrôle d'accès adaptatif est appliquée sur un cas d'utilisation de vidéo surveillance, tenant compte de la sensibilité du contenu consulté et de la mobilité des utilisateurs dans des contextes pervasifs. La solution proposée se situe dans une zone intermédiaire entre le respect de la rigidité des décisions d'accès et la flexibilité extrême de l'option «bris-de-glace » qui est souvent employée dans les situations critiques.

Comme perspective, nous envisageons d'étendre notre proposition en tenant compte d'autres éléments contextuels qui pourraient aussi influencer l'accessibilité aux contenus multimédias (e.g., matériels, réseau, bande passante) et d'appliquer le processus d'adaptation non seulement au niveau de la visualisation mais aussi au niveau du choix des algorithmes d'indexation explicites qui sont aussi protégés par les contraintes RBAC.

Bibliographie et références

Abreu B., Botelho L., Cavallaro A., Douxchamps D., Ebrahimi T., Figueiredo P., Macq B., Mory B., Nunes L., Orri J., Trigueiros M. J., Violante A., (2000) Video-Based Multi-Agent Traffic Surveillance System, *Actes du IEEE Intelligent Vehicles Symposium. IEEE*, p. 457-462

Agosti M., Buccio E. D., Nunzio G. M. D., Ferro N., Melucci M., Miotto R., Orio N., (2007) Distributed information retrieval and automatic identification of music works in SAPIR, *Actes du 15th Italian Symposium on Advanced Database Systems (SEBD)*, p. 479-482.

Al Kukhun D., Sedes, F. (2008): Adaptive Solutions for Access Control within Pervasive Healthcare Systems; *Actes du International Conference On Smart homes and health Telematics (ICOST 2008),* p.42-53.

Batko M., Falchi F., Lucchese C., Novak D., Perego R., Rabitti F., Sedmidubsky J., Zezula P. (2010). Building a web-scale image similarity search system. *Multimedia Tools Appl.* vol. 47, n. 3.

Berry M. W., Castellanos M., (2008). *Survey of Text Mining II: Clustering, Classification, and Retrieval*, Springer.

Bertino E., Bonatti P. A., Ferrari E., (2001) "TRBAC: A temporal role-based access control model," ACM Trans. Inf. Syst. Secur., vol. 4, no. 3, pp. 191–233.

Bertino E., Catania B., Damiani M. L., Perlasca P., (2005) "GEO-RBAC: a spatially aware RBAC," in 10th ACM Symposium on Access Control Models and Technologies SACMAT. ACM, pp. 29–37.

Boisson F., Crucianu M., Vodislav D., (2008). *Publication Framework for Content-Based Search in Heterogeneous Distributed Multimedia Databases*. Rapport de recherche CEDRIC n° 1585, 18 pages.

Brut M., Codreanu D., Dumitrescu S., Manzat A.-M., Sedes F. (2011): A distributed architecture for flexible multimedia management and retrieval. *Acte du Database and Expert Systems Applications (DEXA, 2011), p.*249-263

Brut M., Codreanu D., Manzat A.-M., Sèdes, F. (2011): Adapting Indexation to the Content, Context and Queries Characteristics in Distributed Multimedia Systems. *Acte du International Conference on Signal-Image Technology & Internet-Based Systems (SITIS 2011),* p.118-125.

Chen S.-C., Shyu M.-L., Zhao N. (2004): SMARXO: towards secured multimedia applications by adopting RBAC, XML and object-relational database. *Acte du 12th annual ACM international conf. on Multimedia,* p. 432-435.

El-Khoury, V. (2009): A Multi-level Access Control Scheme for Multimedia Database. *Acte du Workshop on Multimedia Metadata (WMM'09)*.

Ferraiolo D. F., Kuhn R. D., (1992) Role-Based Access Controls. *Acte du 15th National Computer Security Conference*, p.554-563.

Ferreira A., Chadwick D., Farinha P., Correia R., Zao G., Chilro R., Antunes L.,(2009). "How to securely break into RBAC: The BTG-RBAC Model," in Computer Security Applications Conference, ACSAC '09, pp. 23 –31.

Giroux P., Brunessaux S., Brunessaux S., Doucy J., Dupont G., Grilheres B., Mombrun Y., Saval A., (2008). Weblab : An integration infrastructure to ease the development of multimedia processing applications, *Actes du 21st Conference on Software and Systems Engineering and their Applications*.

Hansen F., Oleshchuk V. (2003), "SRBAC: A spatial role-based access control model for mobile systems," in Proceedings of the 7th Nordic Workshop on Secure IT Systems.

Harrison M. A., Ruzzo W. L., Ullman J. D.,(1976) "Protection in operating systems," *Commun. ACM*, vol. 19, no. 8, pp. 461–471.

Jaspers E.G.T., Wijnhoven R.G.J., Albers A.H.R., Desurmont X., Barais M., Hamaide J., Lienard B., (2005). CANDELA - Storage, Analysis and Retrieval of Video Content in Distributed Systems: Real-time Video Surveillance and Retrieval, *Actes du International Conference on Multimedia and Expo*, p. 1553 – 1556.

Joint NEMA/COCIR/JIRA Security and Privacy Committee (SPC), (2004), "Break-glass - an approach to granting emergency access to healthcare systems," White paper.

Kawagoe K., Kasai K., (2011). "Situation, team and role based access control," Journal of Computer Science, vol. 7, no. 5, pp. 629–637.

Kosch H., Maier P., (2009). Content based image retrieval systems – reviewing and benchmarking, *Actes du 9th Workshop on Multimedia Metadata*, p. 1-21.

Kuhn D.R. (1997). "Mutual Exclusion of Roles as a Means of Implementing Separation of Duty in Role-Based Access Control Systems". 2nd ACM Workshop Role-Based Access Control. P. 23–30.

National Institute of Standards and Technology (2006), "Assessment of access control systems," Interagency Report 7316.

OASIS (2003), A brief Introduction to XACML, http://www.oasis-open.org/committees/download.php/ 2713/Brief_Introduction_to_XACML.html

OASIS (2005), "Core and hierarchical role based access control (RBAC) profile of XACML v2.0". February 2005, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-rbac-profile1-spec-os.pdf

Park S.-H., Han Y.-J., Chung T.-M., (2006), "Context-role based access control for context-aware application," in High Performance Computing and Communications, ser. Lecture Notes in Computer Science, Springer Berlin Heidelberg, vol. 4208, pp.572–580.

Petkovic M. , Jonker W., (2005) Content-Based Video Retrieval: A Database Perspective, *Multimedia Systems and Applications*, Berlin: Springer Verlag, vol. 25.

Pietarila P., Westermann U., Jarvinen S., Korva J., Lahti J., Lothman H., (2005). Candela-storage, analysis, and retrieval of video content in distributed systems: Personal mobile multimedia management, *Actes du IEEE International Conference on Multimedia and Expo (ICME)*, p. 1557-1560.

Pinquier J., André-Obrecht R., (2006). Audio Indexing: Primary Components Retrieval - Robust Classification in Audio Documents, *Multimedia Tools and Applications*, vol. 30, n. 3, p. 313-330.

Povey D., (1999), "Optimistic security: a new access control paradigm," in Proceedings of the workshop on New security paradigms, ser. NSPW '99. ACM, pp. 40–45.

Sánchez M., López G., Cánovas O., Sánchez J.-A., Gómez-Skarmeta A. F. (2006): An access control system for multimedia content distribution. *Acte du Third European conference on Public Key Infrastructure: theory and Practice* (EuroPKI 2006),p. 169-183.

Sandhu R., "Role Hierarchies and Constraints for Lattice-Based Access Controls", ESORICS 1996, p. 65-79.

Snoek C. G., Worring M., (2005). Multimodal video indexing: A review of the state of the art, *Multimedia Tools and Applications*, vol. 25, n. 1(January 2005), p. 5- 35.

Thuraisingham B., Lavee G., Bertino E., Fan J., Khan. L. (2006): Access control, confidentiality and privacy for video surveillance databases. *Acte du eleventh ACM symposium on Access control models and technologies* (SACMAT '06), p.1-10.

Viaud M.-L., Thièvre J., Goëau H., Saulnier A., Buisson O., (2008). Interactive components for visual exploration of multimedia archives, *Actes du 7th ACM International Conference on Image and Video Retrieval (CIVR)*, p. 609-616.

Zhang G., Parashar M., (2003). "Dynamic context-aware access control for grid applications," in Proceedings of the 4th International Workshop on Grid Computing, ser. GRID '03. IEEE Computer Society, pp. 101–108.