Oracle Entitlement Server

Présentation

Oracle Entitlement Server (OES) est une solution de gestion des autorisations qui peut être utilisée pour sécuriser l'accès à des applications et à des services au sein d'une organisation. OES fournit une méthode de gestion des autorisations fines pour une large panel de technologies, incluant Java EE, java SE, .NET , SOA, CMS et Databases.

OES permet une claire séparation des cycles de développement et de déploiement, fournissant ainsi aux équipes de développements une approche agnostique du contrôle des accès. D'un point de vue performance et robustesse, OES été conçu pour satisfaire les déploiements les plus complexes : à la différence d'un système d'authentification, l'évaluation fine des autorisations doit remplir des contraintes élevées de temps de latence, de l'ordre de la micro seconde. Une seule page web peut en effet générer plus d'une cinquantaine de requêtes d'autorisation.

OES fournit un modèle de rôle hiérarchique basé sur les deux standards RBAC et ABAC, ainsi qu'une gestion de délégation d'administration multi-niveau qui permet aux multiples organisations et aux parties prenantes dans la gestion des applications de créer, modifier et contrôler les règles d'accès.
OES sécurise l'accès aux ressources d'application et aux composants logiciels (comme les URLs, Enterprise JavaBeans, et Java Server Pages) ainsi que des objets et données métiers (comme des comptes clients ou des enregistrements d'élèves dans une base de données).

Modélisation

Modèle d'autorisations

OES fournit un modèle d'autorisation qui s'applique à des taches métiers, des applications et des processus.

Son objectif est de fournir aux utilisateurs un outil de gestion des autorisations qui permet de modéliser facilement des rôles métiers et des permissions correspondant à des exigences de l'organisation. Il prend en compte la notion d'héritage des permissions.

La structure hiérarchique de OES permet de réduire de façon exponentielle la taille et la complexité des règles d'accès. OES est conforme avec les politique de sécurité ABAC, RBAC, ERBAC et JAAS.

Modélisation des autorisations

Les processus métiers et les entités manipulées sont très souvent de nature hiérarchique. Par exemple, un processus d'approbation de prêt bancaire consiste en plusieurs sous-processus et certains de ceux-ci peuvent entrainer d'autres processus plus fins.

Dans le même ordre d'idée, une application web est souvent structurée en plusieurs pages, chacune d'entre elles étant découpée en sections et chaque section affichant différentes informations. La figure ci-dessous illustre la décomposition d'une page web typique.

Role Base Access Control (RBAC) est une modèle de contrôle d'accès à un système d'information dans lequel chaque décision d'accès est basée sur le rôle auquel l'utilisateur est attaché. En usage depuis plus de trente ans, RBAC est actuellement une des approches les plus largement utilisées dans les différentes industries et la plupart des professionnels de la sécurité en

reconnaissent l'intérêt.

Les Rôles sont la fondation du modèle RBAC.

Les *Enterprise Roles* (ou Groupes) forment la base des autorisations à grande maille. Les *Enterprise Roles* sont assignés de façon statique, dès le moment ou un utilisateur s'authentifie et les droits qui en découlent perdurent jusqu'à la fin de la session.

L'inconvénient de ce type de rôle est d'aboutir à une affectation excessive des permissions. A l'opposé, les *Applications Roles* (*Fine Grained Roles*) sont dynamiques par nature. Ils sont attribués dès le début d'un processus d'autorisation et sont retirés une fois que l'action a été réalisée. Les *Applications Roles* sont donc attribuées sur la base d'un contexte applicatif.

Par exemple, il est utile de distinguer le rôle de *Manager en sein d'une entreprise* par rapport au rôle de *Manager d'une équipe* lorsqu'une action vis-à-vis de ses subordonnées est requise.

OES permet l'assignation dynamique des *Application Roles* basé sur un règlement (ensemble de directives)

Prenons l'exemple d'une université permet de la flexibilité aux professeurs sur la manière dont ils veulent gérer leurs cours et ceux-ci sont autorisés à changer la localisation du cours, le calendrier des examens, l'inscription et les diplômes. Ces privilèges sont restreints seulement au professeur principal de la matière. Cette restriction peut être résumée ainsi :

A/ Un professeur peut administrer un cours, seulement s'il s'agit d'un cours enseigné par lui-même :

Grant or Deny	AppRole	To Subject	On Resource	Only When Condition
Grant	Administrator	professor	Cours	Subject=Cours.Instructor

Les *Role Mapping Policies* peuvent également spécifier une restriction (Deny).

B/ Dans le contexte d'un établissement bancaire, apres la fermeture de l'agence, aucun employé ne peut effectuer des taches de guichetier :

Grant or Deny	AppRole	To Subject	On Resource	Only When Condition
Deny	`Teller`	`Employee`	`Bank`	`Bank_Closed() AND Current_Time > Bank.Closing_Time + 1hr`

Les rôles métiers (*Business Roles*) sont souvent structurés de façon hiérarchique. Les employés ayant une position élevée dans l'organisation bénéficient automatiquement des mêmes privilèges que les personnes sous leur responsabilité.

Pour modéliser ces relations, courantes de la vie réelle, OES supporte la notion de rôle hiérarchique. Ci-dessous, les rôles de développeur, Ingénieur développement, Chef de projet, responsable QA et responsable des tests sont implicitement assignés au directeur de la R&D. Ce type de structure permet ainsi à un président d'une organisation d'hériter des rôles des toutes les personnes de l'entreprise.

Le prédicat qui s'applique sur la directive OES donne un contrôle additionnel sur la manière dont l'octroi ou la résection de rôle opère.

Il est possible d'employer des expressions basées sur :

- le profil de l'utilisateur connecté
- Les *Enterprise Roles* (Groupe)
- Les caractéristiques de la ressource concernée

- des attributs environnementaux.

Par exemple, la figure ci-dessous montre comment écrire la condition de directive :

> Les traders juniors peuvent seulement s'engager sur un montant d'opérations inférieur à un million euros/jour.

OES permet la mis en correspondance direct des attributs de directive avec ceux issus d'un annuaire LDAP, une *database* ou bien d'appels de WebServices. Des plugins spécifiques peuvent invoquer également des conditions supplémentaires à l'intérieur de fonctions. Les utilisateurs complètent avec leurs propres fonctions les fonctions d'OES.

Entitlements

Les processus du monde réel mettent enjeu la plupart du temps d'avantage qu'une seule ressource. En utilisant les *entitlements*, OES permet de modéliser fidèlement des flux métiers et UI

Un *Entitlement* (accréditation) est une combinaison de ressources et d'actions nécessaires pour accomplir une tache métier.

Par exemple, l'ouverture d'un compte bancaire consiste en plusieurs étapes impliquant le remplissage de plusieurs formulaires, la création d'entrées pour chaque tenant du compte et au final, la création du compte proprement dit.

Lorsqu'un banquier est autorisé à ouvrir un nouveau compte, il lui faudra disposer d'un accès à plusieurs ressources. C'est ici que le concept *d'entitlement* intervient. Il représente dans notre exemple, l'ensemble des accès nécessaires à l'accomplissement du processus de bout en bout. Tous les privilèges (s'appliquant aux ressources et actions) nécessaires peuvent alors être groupés dans

un seule *entitlement*. Ainsi, *L'entitlement* permet à des administrateurs *sécurité* de se concentrer sur des définitions métiers de processus plutôt que sur des privilèges de plus bas niveau.

L'assignation dynamique des rôles adresse seulement la première moitié du problème d'autorisation. Il reste à préciser quels privilèges du monde bien réel qui rentrent dans le périmètre de chaque rôle

OES autorisation met en correspondance les utilisateurs, les Enterprise Roles et les *Application Roles* sur des privilèges.
Les directives d'autorisation fournissent en retour à un sujet un Permit/Deny qui lui permettra de déclencher ou non des taches sur une ressource lorsque la condition est satisfaite. Par exemple, un centre d'appel peut avoir une directive qui permet aux employés d'avoir accès aux informations relatives à leur carte de paiement seulement lorsqu'ils sont physiquement présents dans les bureaux.

Cette directive peut être exprimée ainsi :

> *Autorise les employés à consulter les numéros de carte de crédit des clients seulement lorsqu'ils sont connectés avec une adresse IP de l'intranet.*

OES support XACML, la notion d'obligation.

Pour certains cas, une simple *Permit/Deny* provenant d'un moteur d'autorisation n'est pas suffisant.

Cela arrive parce que certaines parties de la condition ne peuvent pas être évaluée par le moteur. A la place, un filtre de recherche (analogue à une requête SQL) a besoin d'être appliqué. Ainsi, le moteur peut retourner un requête SQL à employer comme filtre. Cela est appelé une *obligation*.

Construction d'un modèle RBAC avec OES

Comme cela a été souligné précédemment, Les modèles RBAC utilisent le concept de *rôle*. Pour RBAC, une permission (ou privilège) est une simple extension de la notion de rôle

Pour rappel, les rôles sont divisés en trois grandes catégories :

- *Enterprise static Roles* : les utilisateurs bénéficient de ces rôles, indépendamment des taches qu'ils réalisent. Ces rôles sont très souvent enregistrés dans un annuaire LDAP.
- *Application Specific Static Roles* : Ces rôles sont attribués dans le cadre d'une application.
- *Application Specific Dynamic Roles* : Il s'agit de rôles que l'on qualifie de dynamiques ou bien décrits comme des rôles qui sont attribués sous condition. Ils sont attribués selon le type de tache à réaliser et sont retirés une fois la tache effectuée. C'est ici que le produit OES fournit un moyen sophistiqué pour la gestion de cette catégorie de rôles.

Construction de modèle ABAC avec OES

Les attributs constituent la base d'utilisation du model ABAC. OES supporte une grande variété d'attributs (utilisateur, *Enterprise Role*, ressource, application rôle et action requise)

Architecture de OES

OES comprend deux composants principaux :

- Administration Server

- Security Modules.

L'administration Server agit comme le point d'administration des directives ou *policies* (PAP). Il est utilisé pour gérer les configurations, les organisations, les applications, les directives de sécurité et les rôles.

OES applique les règles d'autorisation à l'exécution par le biais d'un ou plusieurs Modules de sécurité.

L'architecture d'OES permet à ces modules de sécurité d'être considérés comme un point de décision unique pour l'ensemble des applications et d'appliquer des politiques de droits d'accès définies à travers la console d'administration.

Les modules de sécurité représentent aussi le point d'intégration pour les identités des utilisateurs et l'accès aux attributs externes qui peuvent être inclus dans les politiques de contrôle.

Ces sources d'information peuvent être des bases de données relationnelles ou des annuaires.

Les politiques OES spécifient quels utilisateurs, groupes et/ou rôles peuvent accéder aux ressources de l'application, les permissions dont ils disposent et à quel moment ces ressource sont accessibles.

Une politique typique OES pourrait être une action: 'Accorder Visualisation' pour 'Liste des Notes', pour tout utilisateur se trouvant dans le groupe d'utilisateurs 'Gestionnaire Matière'.

Par son architecture unique, flexible, OES peut évaluer également des attributs spécifiques afin de prendre d'autres décisions de contrôle d'accès.

Utilisation de OES dans le cadre de WebCenter Portal

Oracle Entitlement Server (OES) permet de définir de façon centralisée des règles qui déterminent si un utilisateur bénéficie ou non d'un rôle donné, et ce, de façon dynamique au cours du processus de connexion. Après l'authentification d'un utilisateur, OVD réunira les appartenances de groupes issues des annuaires et les rôles dynamiques issus d'OES.
Une définition de groupe dynamique peut inclure des attributs de profil, le moment de la journée, un seuil numérique et fournit ainsi un moyen flexible pour déterminer finement l'accès à une application ou bien à un composant.

Par exemple, Un apprenant peut accéder à un composant, seulement durant une certains plage horaire dans la journée.

Les rôles dynamiques peuvent être définis dan OES en tant que « Rôles avec Contraintes ». Les rôles définis dans OES sont ajoutés aux *Enterprise Roles* par l'intermédiaire du Plugin OVD. Quand l'utilisateur se connecte, les règles sont évaluées pour déterminer les rôles dynamiques de celui-ci

La figure ci-dessous résume le processus de login:

Login Process

Récupération des Enterprise Group et des rôles dynamiques

Par défaut, WebCenter Portal récupère uniquement les *Enteprise Roles* définis dans le référentiel d'identités. Afin d'utiliser les rôles dynamiques définis dans OES, il est nécessaire d'ajouter le Plugin OVD en tant qu'*authenticator*. Ce plugin peut alors consolider les rôles statiques issus du référentiel d'identités avec ceux, dynamiques, calcules par OES.

Author



- Patrick

  GPM Factory