Groupe de travail Réseau
Request for Comments : 3703
Catégorie : En cours de normalisation
février 2004
Traduction Claude Brière de L'Isle

J. Strassner, Intelliden Corporation
B. Moore, IBM Corporation
R. Moats, Lemur Networks, Inc.
E. Ellesson

Schéma de cœur de politique du protocole léger d'accès à un répertoire (LDAP)

Statut du présent mémoire
Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent document définit une transposition du modèle d'informations de cœur de politique en une forme qui puisse être mise en œuvre dans un répertoire qui utilise le protocole léger d'accès à un répertoire (LDAP, *Lightweight Directory Access Protocol*) comme protocole d'accès. Ce modèle définit deux hiérarchies de classes d'objet : les classes structurelles qui représentent les informations pour les données de politique de représentation et de contrôle comme spécifiées dans la RFC3060, et les classes de relations qui indiquent comment les instances des classes structurelles se rapportent les unes aux autres. Les classes sont aussi ajoutées au schéma LDAP pour améliorer les performances des interactions d'un client avec un serveur LDAP lorsque le client récupère de grandes quantités d'informations en rapport avec la politique. Ces classes n'existent que pour optimiser les restitutions de LDAP : ce ne sont pas des classes dans le modèle d'information qui leur correspond.

Table des Matières

1. Introduction

Le présent document prend pour point de départ le modèle d'informations tourné vers l'objet pour représenter les informations qui figurent les données de politique de représentation et de contrôle comme spécifié dans la [RFC3060]. Les mises en œuvre du protocole léger d'accès à un répertoire (LDAP, *Lightweight Directory Access Protocol*) [RFC3377] sont invitées à noter que dans le présent document, l'utilisation du terme "politique" ne se réfère pas à celle qui est définie dans la Recommandation UIT-T X.501 [X.501]. L'utilisation du terme "politique" tout au long du présent document est plutôt définie comme suit :

> La politique est définie comme un ensemble de règles pour administrer, gérer, et contrôler l'accès aux ressources réseau.

Le présent travail est actuellement en cours de développement conjoint entre le groupe de travail Cadre de politique de l'IETF et le groupe de travail Politique de l'équipe de gestion répartie (DTMF, *Distributed Management Task Force*). Le présent modèle définit deux hiérarchies de classes d'objet : les classes structurelles qui représentent les informations de politique et les commandes de politiques, et les classes de relations qui indiquent comment les instances des classes structurelles sont en rapport les unes avec les autres. En général, ces deux hiérarchies de classes devront être transposées en un magasin de données particulier.

Le présent document définit la transposition de ces classes de modèle d'information en un répertoire qui utilise LDAP comme protocole d'accès. Deux types de transpositions sont impliqués :

- Pour les classes structurelles dans le modèle d'information, la transposition de base est d'un à un : les classes du modèle d'information se transposent en classes LDAP, les propriétés du modèle d'information se transposent en attributs LDAP.
- Pour les classes de relations dans le modèle d'information, différentes transpositions sont possibles. Dans le présent document, les classes de relation du modèle d'information de cœur de politique (PCIM, *Policy Core Information Model*) et leurs propriétés sont transposées de trois façons : en classes auxiliaires LDAP, en attributs représentant les références de nom distinctif (DN, *distinguished name*) et en relations supérieur-subordonné dans l'arborescence d'informations de répertoire (DIT, *Directory Information Tree*).

Les mises en œuvre qui utilisent un répertoire LDAP comme répertoire de politiques et veulent mettre en œuvre des information de politique conformément à la [RFC3060] DEVRONT utiliser le schéma LDAP défini dans le présent document, ou un schéma qui découle de celui défini dans le présent document. L'utilisation du modèle d'information défini dans la [RFC3060] comme point de départ permet d'en hériter et d'étendre les hiérarchies de classes de relations, afin que d'autres types de répertoires de politiques, comme des bases de données relationnelles, puissent aussi utiliser ces informations.

Le présent document entre dans le cadre global pour la représentation, le déploiement, et la gestion des politiques qui sont développées par le groupe de travail Cadre de politique.

Le schéma LDAP décrit dans le présent document utilise le préfixe "pcim" pour identifier ses classes et attributs. Il consiste en dix classes très générales : pcimPolicy (une classe abstraite), trois classes de groupe de politique (pcimGroup, pcimGroupAuxClass, et pcimGroupInstance), trois classes de règle de politique (pcimRule, pcimRuleAuxClass, et pcimRuleInstance), et trois classes auxiliaires spéciales (pcimConditionAuxClass, pcimTPCAuxClass, et pcimActionAuxClass). (Noter que la classe auxiliaire PolicyTimePeriodCondition définie dans la [RFC3060] aurait normalement dû être appelée pcimTimePeriodConditionAuxClass, mais ce nom est trop long pour certains répertoires. Donc, on l'a abrégé en pcimTPCAuxClass).

La transposition pour les classes PCIM pcimGroup et pcimRule est conçue pour être aussi souple que possible. Trois classes sont définies pour ces deux classes PCIM. D'abord, une super classe abstraite est définie comme contenant toutes les propriétés requises de chaque classe PCIM. Puis, une classe auxiliaire ainsi qu'une classe structurelle sont toutes deux déduites

de la super classe abstraite. Cela donne un maximum de souplesse au développeur.

Le schéma contient aussi deux classes moins générales : pcimConditionVendorAuxClass et pcimActionVendorAuxClass. Pour réaliser la transposition des relations du modèle d'information, le schéma contient aussi deux classes auxiliaires : pcimGroupContainmentAuxClass et pcimRuleContainmentAuxClass. Saisir la distinction entre condition de politique spécifique d'une règle et condition de politique réutilisable et les actions de politique introduit sept autres classes : pcimRuleConditionAssociation, pcimRuleValidityAssociation, pcimRuleActionAssociation, pcimPolicyInstance, et trois classes de répertoire de politique (pcimRepository, pcimRepositoryAuxClass, et pcimRepositoryInstance). Finalement, le schéma comporte deux classes (pcimSubtreesPtrAuxClass et pcimElementAuxClass) pour optimiser les restitutions de LDAP. En tout, le schéma contient 23 classes.

Dans le contexte du présent document, le terme "PCLS" (*Policy Core LDAP Schema*, schéma LDAP de cœur de politique) est utilisé pour se référer aux définitions de classe LDAP que contient le présent document. Le terme"PCIM" se réfère aux classes définies dans la [RFC3060].

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGÉ", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC2119].

2. Modèle d'informations de cœur de politique

Le présent document contient un schéma LDAP qui représente les classes définies dans le document d'accompagnement "Modèle d'information de cœur de politique – Spécification de la version 1" [RFC3060]. D'autres documents peuvent être produits ultérieurement avec des transpositions de ce même PCIM sur d'autres technologies de mémorisation. Comme le détail de la sémantique des classes PCIM n'apparaît que dans la [RFC3060], ce document est un prérequis de la lecture et la compréhension du présent document.

3. Hiérarchie d'héritage pour le PCLS

Le diagramme qui suit illustre la hiérarchie des classes pour les classes LDAP définies dans ce document :

```
Sommet
   |
   +--dlm1ManagedElement (abstraite)
   |   |
   |   +--pcimPolicy (abstraite)
   |   |   |
   |   |   +--pcimGroup (abstraite)
   |   |   |  |
   |   |   |  +--pcimGroupAuxClass (auxiliaire)
   |   |   |  |
   |   |   |  +--pcimGroupInstance (structurelle)
   |   |   |
   |   |   +--pcimRule (abstraite)
   |   |   |  |
   |   |   |  +--pcimRuleAuxClass (auxiliaire)
   |   |   |  |
   |   |   |  +--pcimRuleInstance (structurelle)
   |   |   |
   |   |   +--pcimRuleConditionAssociation (structurelle)
   |   |   |
   |   |   +--pcimRuleValidityAssociation (structurelle)
   |   |   |
   |   |   +--pcimRuleActionAssociation (structurelle)
   |   |   |
   |   |   +--pcimPolicyInstance (structurelle)
   |   |   |
   |   |   +--pcimElementAuxClass (auxiliaire)
   |   |
   |   +--dlm1ManagedSystemElement (abstract)
```

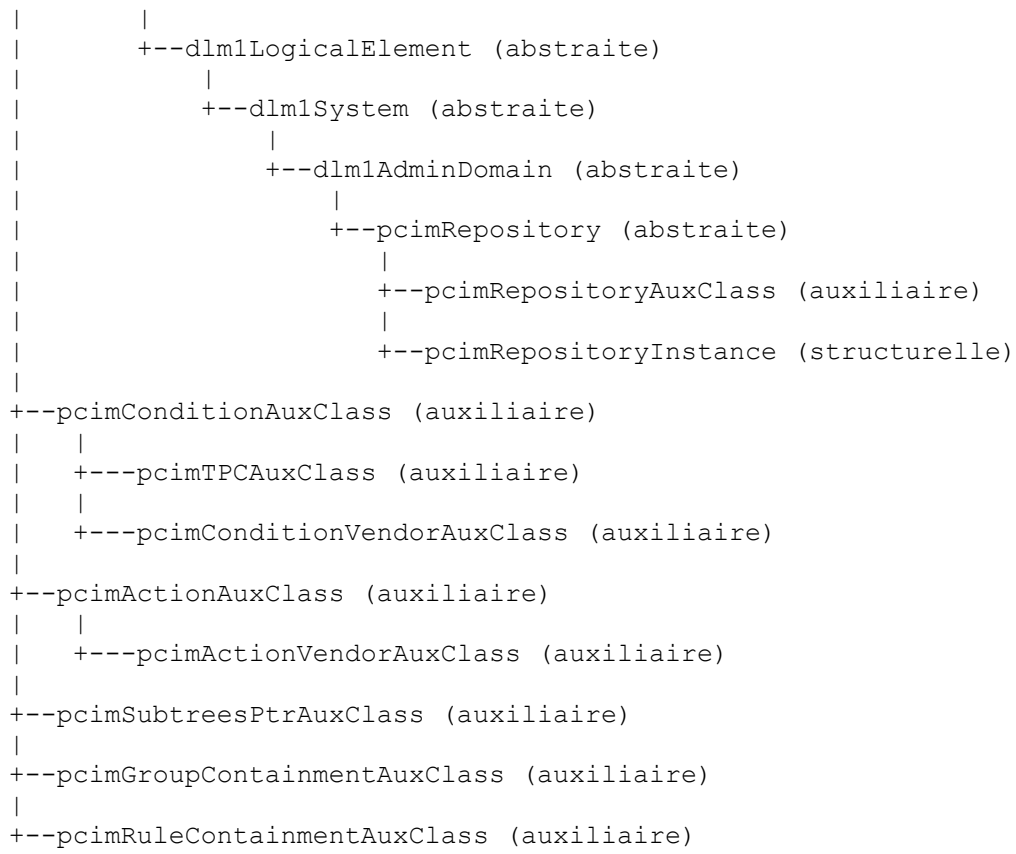


Figure 1 : Hiérarchie d'héritage de classe LDAP pour le PCLS

4. Discussion générale de la transposition du modèle d'information en LDAP

Les classes décrites dans la Section 5 ci-dessous contiennent certaines optimisations pour un répertoire qui utilise LDAP comme protocole d'accès. Un exemple en est l'utilisation des classes auxiliaires pour représenter certaines des associations définies dans le modèle d'information. D'autres magasins de données peuvent avoir besoin de mettre différemment en œuvre ces associations. Un second exemple est l'introduction de classes spécifiquement conçues pour optimiser la restitution de grandes quantités de données en rapport avec la politique à partir d'un répertoire. Cette section expose certains sujets généraux relatifs à la transposition du modèle d'information en LDAP.

Le reste de la section expose les sujets suivants ; le paragraphe 4.1 expose la stratégie utilisée dans la transposition des classes et associations définies dans la [RFC3060] en une forme qui puisse être représentée dans un répertoire qui utilise LDAP comme protocole d'accès. Le paragraphe 4.2 expose le contenu du DIT et les règles de structure, ainsi que les formes de nom. Le paragraphe 4.3 décrit la stratégie utilisée pour définir les attributs de désignation pour le schéma décrit à la Section 5 de ce document. Le paragraphe 4.4 définit la stratégie recommandée pour localiser et restituer les objets déduits de PCIM dans le répertoire.

4.1 Sommaire des transpositions de classe et d'association

Quinze des classes de la PCLS viennent directement des neuf classes correspondantes dans le modèle d'information. Noter que les noms des classes commencent par un caractère majuscule dans le modèle d'information (bien que dans le cas particulier de CIM, la casse ne soit pas significative dans les noms de classe et de propriété) mais par un caractère minuscule dans LDAP. Ceci parce que bien que LDAP n'y attache pas d'importance, X.500 ne permet pas que les noms de classe commencent par une majuscule. Noter aussi que le préfixe "pcim" est utilisé pour identifier ces classes LDAP.

Modèle d'information	**Classes LDAP**
Policy	pcimPolicy
PolicyGroup	pcimGroup, pcimGroupAuxClass, pcimGroupInstance
PolicyRule	pcimRule, pcimRuleAuxClass, pcimRuleInstance
PolicyCondition	pcimConditionAuxClass
PolicyAction	pcimActionAuxClass

VendorPolicyCondition	pcimConditionVendorAuxClass
VendorPolicyAction	pcimActionVendorAuxClass
PolicyTimePeriodCondition	pcimTPCAuxClass
PolicyRepository	pcimRepository, pcimRepositoryAuxClass, pcimRepositoryInstance

Figure 2 : Transposition des classes de modèle d'information en LDAP

Les associations dans le modèle d'information se transposent en attributs qui font référence à des noms distinctifs (DN, *Distinguished Name*) ou à un contenu des arborescences d'information de répertoire (DIT, *Directory Information Tree*) (c'est-à-dire, des relations de supérieur à subordonné) dans LDAP. Deux des attributs qui font référence à des DN apparaissent dans des classes auxiliaires, ce qui permet à chacun d'eux de représenter plusieurs relations du modèle d'information.

Association de modèle d'information	**Attribut/Classe LDAP**
PolicyGroupInPolicyGroup	pcimGroupsAuxContainedSet dans pcimGroupContainmentAuxClass
PolicyRuleInPolicyGroup	pcimRulesAuxContainedSet dans pcimRuleContainmentAuxClass
PolicyConditionInPolicyRule	Contenu du DIT ou pcimRuleConditionList dans pcimRule ou pcimConditionDN dans pcimRuleConditionAssociation
PolicyActionInPolicyRule	Contenu du DIT ou pcimRuleActionList dans pcimRule ou pcimActionDN dans pcimRuleActionAssociation
PolicyRuleValidityPeriod	pcimRuleValidityPeriodList dans pcimRule ou (si réutilisable) référencé par le pcimTimePeriodConditionDN dans pcimRuleValidityAssociation
PolicyConditionInPolicyRepository	Contenu du DIT
PolicyActionInPolicyRepository	Contenu du DIT
PolicyRepositoryInPolicyRepository	Contenu du DIT

Figure 3 : Transposition des associations de modèle d'information dans LDAP

Des classes PCLS restantes, deux (pcimElementAuxClass et pcimSubtreesPtrAuxClass) sont incluses pour rendre plus efficace la navigation à travers le DIT et la restitution des entrées qui s'y trouvent. Ce sujet est exposé au paragraphe 4.5.

Les quatre classes restantes dans le PCLS, pcimRuleConditionAssociation, pcimRuleValidityAssociation, pcimRuleActionAssociation, et pcimPolicyInstance, sont toutes impliquées dans la représentation des conditions de politique et des actions de politique dans un répertoire LDAP. Ce sujet est exposé au paragraphe 4.4.

4.2 Usage des règles de contenu et structure DIT et des formes de nom

Trois outils puissants peuvent être utilisés pour aider à définir les schémas. Le premier, règles de contenu de DIT, est une façon de définir le contenu d'une entrée pour une classe d'objet structurelle. Elle peut être utilisée pour spécifier les caractéristiques suivantes de l'entrée :

- des attributs obligatoires supplémentaires que les entrées sont obligées de contenir,
- des attributs facultatifs supplémentaires qu'il est permis aux entrées de contenir,
- l'ensemble des classes d'objet auxiliaires supplémentaires dont il est permis à ces entrées d'être membres,
- tous attributs facultatifs provenant des définitions de classe d'objet structurelle et auxiliaire que les entrées sont obligées d'éviter.

Les règles de contenu de DIT NE SONT obligatoires pour AUCUNE classe d'objet structurelle.

Une règle de structure de DIT, conjointement à une forme de nom, contrôle le placement et la désignation d'une entrée au sein de la portée d'un sous schéma. Les formes de nom définissent l'utilisation de quels types d'attributs sont requis et permis pour former les noms distinctifs relatifs (RDN, *Relative Distinguished Name*) des entrées. Les règles de structure de DIT spécifient à quelles entrées il est permis d'être supérieures aux autres entrées, et donc de contrôler la façon dont les RDN sont ajoutés ensemble pour constituer des noms distinctifs.

Une forme de nom spécifie ce qui suit :

- la classe d'objet structurel des entrées désignées par cette forme de nom,
- les attributs dont l'utilisation est obligatoire dans la formation des RDN de ces entrées,
- les attributs dont l'utilisation est permise dans la formation des RDN de ces entrées,
- un identifiant d'objet pour identifier de façon univoque cette forme de nom.

Noter que les formes de nom ne peuvent être spécifiées que pour les classes d'objet structurelles. Cependant, chaque entrée dans le DIT doit avoir une forme de nom qui la contrôle.

Malheureusement, les serveurs LDAP actuels présentent une grande disparité dans la prise en charge de ces caractéristiques. Il y a aussi trois points cruciaux qui doivent être suivis pour la mise en œuvre. D'abord l'utilisation selon X.500 des règles de structure exige qu'une classe d'objet structurelle sans règle de structure supérieure soit un point administratif de sous schéma. C'est exactement ce qu'on NE VEUT PAS pour les informations de politique. Ensuite, lorsque une classe auxiliaire est sous classée, si il existe une règle de contenu pour la classe structurelle à laquelle se réfère la classe auxiliaire, cette règle de contenu doit alors être augmentée. Enfin, la plupart des serveurs LDAP ne prennent malheureusement pas en charge l'héritage des règles de structure et de contenu.

Étant donnés ces problèmes, les règles de structure et de contenu de DIT ont été retirées du PCLS. Cela parce que, si elles y étaient incluses, elles seraient des références normatives et exigeraient des OID. Cependant, on ne veut pas perdre l'acquis de la construction des règles de structure et de contenu des versions précédentes du schéma. On décrit donc où de telles règles pourraient être utilisées dans ce schéma, ce qu'elles contrôleraient, et quel effet elles auraient.

4.3 Attributs de désignation dans le PCLS

Dans un répertoire, les instances sont identifiées par des noms distinctifs (DN), qui donnent le même type d'organisation hiérarchique que fournit un système de fichiers dans un système informatique. Un nom distinctif est une séquence de RDN. Un RDN donne un identifiant unique d'une instance dans le contexte de son supérieur immédiat, de la même façon qu'un nom de fichier fournit un identifiant unique pour un fichier dans le contexte du dossier dans lequel il réside.

Pour préserver le maximum de souplesse de dénomination pour les administrateurs de politique, trois attributs de désignation facultatifs (c'est-à-dire, "PEUT") ont été définis. Ce sont :

- Chacune des classes structurelles définies dans ce schéma a son propre attribut de dénomination unique ("PEUT"). Comme les attributs de dénomination sont différents, un administrateur de politique peut, en utilisant ces attributs, garantir qu'il n'y aura pas de collision de noms entre des instances de différentes classes, même si la même valeur est allouée aux attributs de dénomination respectifs des instances.
- L'attribut LDAP cn (qui correspond au commonName de X.500) est inclus comme attribut PEUT dans la classe abstraite pcimPolicy, et donc par héritage dans toutes ses sous classes. Dans X.500, un commonName fonctionne normalement comme un attribut de RDN, pour les instances de dénomination de nombreuses classes (par exemple, la classe "person" de X.500).
- Un attribut spécial est fourni pour les mises en œuvre qui s'attendent à une transposition entre les représentations natives CIM et LDAP d'informations de politique. Cet attribut, appelé orderedCimKeys, est défini dans la classe dlm1ManagedElement [DTMF]. La valeur de cet attribut est déduite par un algorithme des valeurs déjà présentes dans une instance de politique CIM. La référence normative de cet algorithme est contenue dans [DTMF]. Voir à l'appendice du présent document la description de l'algorithme.

Comme tous ces attributs de désignation PEUVENT être utilisés pour désigner une instance d'une classe PCLS, les mises en œuvre DOIVENT être capables de s'accommoder des instances désignées de toutes ces façons.

Noter qu'il est recommandé que deux de ces attributs ou plus NE DEVRAIENT PAS être utilisés ensemble pour former un RDN multi parties, car la prise en charge des RDN multi parties est limitée dans les mises en œuvre existantes de répertoire.

4.4 Conditions et actions spécifiques de règle et réutilisables

PCIM [RFC3060] distingue entre deux types de conditions de politique et d'actions de politique : celles associées à une seule règle de politique, et celles qui sont réutilisables, dans ce sens qu'elles peuvent être associées à plus d'une règle de politique. Bien qu'il n'y ait pas de différence fonctionnelle inhérente entre une condition ou action spécifique de règle et celle qui est réutilisable, il y a entre elles une différence à la fois d'usage et de mise en œuvre.

Définir une condition ou action comme réutilisable ou comme spécifique d'une règle reflète une décision consciente de la part de l'administrateur qui définit comment elles sont utilisées. De plus, il y a des variations qui reflètent la mise en œuvre des conditions ou actions de politique spécifiques d'une règle et celles réutilisables et la façon dont elles sont traitées dans un répertoire de politique. Les différences majeures de mise en œuvre entre une condition ou action de politique spécifique d'une règle et la réutilisable sont décrites ci-dessous.

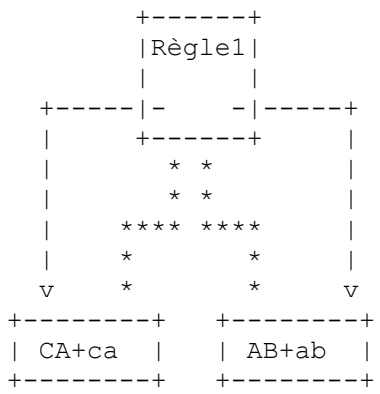
1. Il est naturel qu'une condition ou action spécifique d'une règle soit retirée du répertoire de politiques en même temps que la règle. C'est juste le contraire pour les conditions et actions réutilisables. Cela parce que la condition ou action est conceptuellement attachée à la règle dans le cas spécifique de la règle, tandis qu'elle est référencée (par exemple, pointée) dans le cas réutilisable. La persistance d'une instance de pcimRepository est indépendante de celle d'une instance de pcimRule.
2. Les permissions d'accès pour une condition ou action spécifique d'une règle sont usuellement identiques à celles pour la règle elle-même. D'un autre côté, les permissions d'accès des conditions et actions réutilisables doivent être exprimables

sans référence à une règle de politique.

3. Les conditions et actions spécifiques d'une règle requièrent moins d'accès, parce que les conditions et actions sont "attachées" à la règle. À l'opposé, les conditions et actions réutilisables exigent plus d'accès, parce que chaque condition ou action qui est réutilisable requiert un accès séparé.
4. Les conditions et actions spécifiques d'une règle sont conçues pour être utilisées par une seule règle. Comme le nombre de règles qui utilisent la même condition spécifique d'une règle augmente, des problèmes subtils sont créés (le plus évident étant comment garder à jour les conditions et actions spécifiques d'une règle pour refléter la même valeur). Les conditions et actions réutilisables se prêtent à l'utilisation par de multiples règles indépendantes.
5. Les conditions et actions réutilisables offrent une optimisation lorsque plusieurs règles utilisent la même condition ou action. Cela parce que la condition ou action réutilisable a seulement besoin d'être mise à jour une fois, et par la vertu de la référence au DN, la règle de politique sera automatiquement mise à jour.

Le paragraphe précédent ne contient pas une liste exhaustive des façons dont les conditions réutilisables et spécifiques d'une règle devraient être traitées différemment. Son objet est simplement de justifier qu'on fasse une distinction sémantique entre spécifique d'une règle et réutilisable, puis de refléter cette distinction dans le répertoire de politiques lui-même.

Lorsque le répertoire de politiques est réalisé dans un répertoire accessible par LDAP, la distinction entre conditions et actions spécifiques de règle et réutilisables est réalisée via le placement de classes auxiliaires et via l'enracinement dans une DIT. La Figure 4 illustre une règle de politique Règle1 avec une condition CA spécifique de règle et une action AB spécifique de règle.



Légende :
***** enracinement DIT
+ rattachement auxiliaire
----> référence de DN

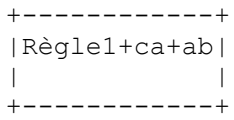
Figure 4 : Conditions et actions de politique spécifiques d'une règle

Parce que la condition et l'action sont spécifiques de la Règle1, les classes auxiliaires ca et ab qui les représentent sont attachées respectivement aux classes structurelles CA et AB. Ces classes structurelles ne représentent pas la condition ca et l'action ab elles-mêmes, mais plutôt les associations entre Règle1 et ca, et entre Règle1 et ab.

Comme l'illustre la Figure 4, la Règle1 contient les références de DN aux classes structurelles CA et AB qui apparaissent en dessous d'elles dans la DIT. À première vue, il peut paraître que ces références de DN sont inutiles, car une recherche de sous arborescence en dessous de la Règle1 trouverait toutes les classes structurelles représentant les associations entre Règle1 et ses conditions et actions. S'appuyer seulement sur une recherche de sous arborescence fait cependant courir le risque de manquer des conditions ou actions qui auraient dû apparaître dans la sous arborescence, mais ne l'ont pas fait pour une raison quelconque, ou de trouver des conditions ou actions qui ont été placées par inadvertance dans la sous arborescence, ou qui auraient dû être retirées de la sous arborescence, mais ne l'ont pas été pour une raison quelconque. L'expérience de la mise en œuvre suggère que beaucoup de ces risques (mais pas tous) ont été éliminés.

Cependant, on doit noter que ceci vient en plus. L'utilisation des références de DN, comme le montre la Figure 4, contrarie l'héritage des informations de contrôle d'accès ainsi que l'existence des informations de dépendance. Elle est aussi sujette à des considérations d'intégrité de références. Donc, elle est incluse comme option du concepteur.

La Figure 5 illustre une seconde façon de représenter les conditions et actions spécifiques d'une règle dans un répertoire accessible par LDAP : le rattachement des classes auxiliaires directement à l'instance qui représente la règle de politique. Lorsque toutes les conditions et actions sont rattachées à une règle de politique de cette façon, la règle est appelée une règle de politique "simple". Lorsque les conditions et actions ne sont pas rattachées directement à une règle de politique, la règle est appelée une règle de politique "complexe".



Légende : + rattachement auxiliaire

Figure 5 : Règle de politique simple

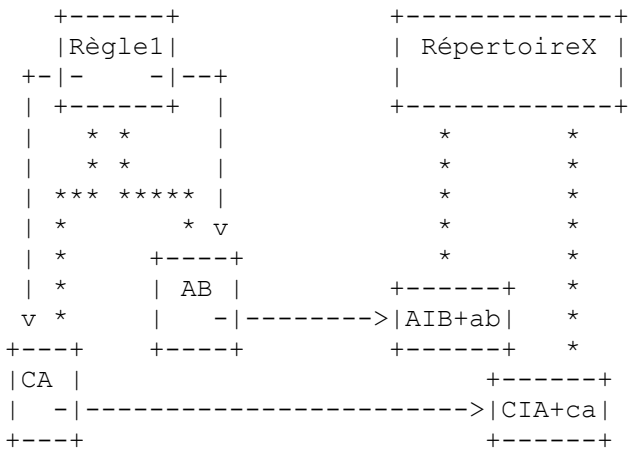
La distinction simple/complexe pour une règle de politique n'est pas de tout ou rien. Une règle de politique peut avoir ses conditions rattachées à elle-même et ses actions rattachées à d'autres entrées, ou elle peut avoir ses actions rattachées à elle-même et ses conditions rattachées à d'autres entrées. Cependant, elle NE DEVRA PAS avoir soit ses conditions soit ses actions rattachées toutes deux à elle-même et aux autres entrées, à une seule exception près : une règle de politique peut référencer ses périodes de validité avec l'attribut pcimRuleValidityPeriodList, mais avoir ses autres conditions rattachées à elle-même.

Les compromis entre règles de politique simples et complexes sont entre l'efficacité des règles simples et la souplesse et le plus grand potentiel de réutilisation des règles complexes. Avec une règle de politique simple, les options sémantiques sont limitées :

- Toutes les conditions sont ajoutées par l'opérateur logique ET. Cette combinaison peut être représentée de deux façons dans les caractéristiques d'expression au format normal disjonctif (DNF, *Disjunctive Normal Form*)/format conjonctif normal (CNF, *Conjunctive Normal Form*) (voir la définition de ces termes dans la [RFC3060]) des conditions de politique : comme une expression DNF avec un seul groupe ET, ou comme expression CNF avec plusieurs groupes OU d'une seule condition. La première est choisie arbitrairement comme représentation des conditions ET dans une règle de politique simple.
- Si plusieurs actions sont incluses, aucun ordre ne peut être spécifié pour elles.

Si un administrateur de politique a besoin de combiner des conditions d'une autre façon, ou si il y a un ensemble d'actions qui doivent être ordonnées, alors la seule option est d'utiliser une règle de politique complexe.

Finalement, la Figure 6 illustre la même règle de politique Règle1, mais cette fois sa condition et son action sont réutilisables. Les classes d'association CA et AB sont toujours présentes, et elles sont encore enracinées dans la DIT sous la Règle1. Mais plutôt que d'avoir les classes auxiliaires ca et ab rattachées directement aux classes d'association CA et AB, chacune contient maintenant des références de DN aux autres entrées auxquelles ces classes auxiliaires sont rattachées. Ces autres entrées, CIA et AIB, sont contenues dans la DIT sous le RépertoireX, qui est une instance de la classe pcimRepository. Parce que elles sont nommées sous une instance de pcimRepository, ca et ab sont clairement identifiées comme réutilisables.



Légende :
***** enracinement DIT
+ rattachement auxiliaire
----> référence de DN

Figure 6 : Conditions et actions de politique réutilisable

Les classes pcimConditionAuxClass et pcimActionAuxClass ne représentent pas elles-mêmes les conditions et actions réelles : elles sont introduites dans leurs sous classes. Ce qu'introduisent pcimConditionAuxClass et pcimActionAuxClass est la sémantique d'une condition de politique ou d'une action de politique. C'est la sémantique dont toutes les sous classes de pcimConditionAuxClass et pcimActionAuxClass héritent. Dans cette sémantique il y a la représentation d'une condition de

politique ou action de politique spécifique d'une règle ou réutilisable .

Pour préserver la capacité à représenter une condition ou action spécifique de règle ou réutilisable, aussi bien qu'une règle de politique simple, toutes les sous classes de pcimConditionAuxClass et pcimActionAuxClass DOIVENT aussi être des classes auxiliaires.

4.5 Localisation et récupération des objets de politique dans le répertoire

Lorsque un point de décision de politique (PDP, *Policy Decision Point*) va à un répertoire LDAP pour récupérer les instances d'objet de politique pertinentes pour les points d'application de politique (PEP, *Policy Enforcement Point*) qu'il dessert, il se trouve face à deux problèmes liés :

- Comment localiser et récupérer les entrées de répertoire qui s'appliquent à ses PEP ? Ces entrées peuvent inclure des instances des classes de PCLS, des instances de sous classes spécifiques du domaine de ces classes, et des instances des autres classes modélisant des ressources comme des groupes d'utilisateurs, des interfaces, et des gammes d'adresse.
- Comment récupérer les entrées de répertoire dont il a besoin d'une façon efficace, afin que la restitution des informations de politique du répertoire ne devienne pas une entrave à l'adaptabilité ? Il y a deux facettes à cette efficacité : la restitution des seules entrées de répertoire pertinentes, et la restitution de ces entrées en utilisant aussi peu d'appels LDAP que possible.

Le placement d'objets dans l'arborescence d'informations de répertoire (DIT, *Directory Information Tree*) implique des considérations autres que comment les objets en rapport avec la politique seront récupérés par un PDP. Par conséquent, tout ce que peut faire le PCLS est de fournir une "trousse à outils" de classes pour aider l'administrateur de politique lorsque la DIT est conçue et bâtie. Un PDP DEVRAIT être capable de tirer parti de tous les outils que l'administrateur de politique est capable de construire dans la DIT, mais il DOIT être capable d'utiliser des moyens de restitution moins efficaces si c'est tout ce dont il dispose.

L'idée de base derrière les classes d'optimisation LDAP est simple : rendre possible à un PDP la restitution de tous les objets en rapport avec la politique dont il a besoin, et seulement ces objets, en utilisant aussi peu d'appels LDAP que possible. Une hypothèse importante sous-jacente à cette approche est que l'administrateur de politique a un contrôle suffisant sur la structure de DIT sous-jacente pour définir des sous arborescences pour mémoriser les informations de politique. Si l"administrateur de politique n'a pas ce niveau de contrôle sur la structure de DIT, un PDP peut quand même restituer les objets en rapport avec la politique individuels dont il a besoin. Mais cela va exiger plus d'opérations d'accès LDAP pour faire la restitution de cette façon. La Figure 7 illustre comment l'optimisation LDAP est réalisée.

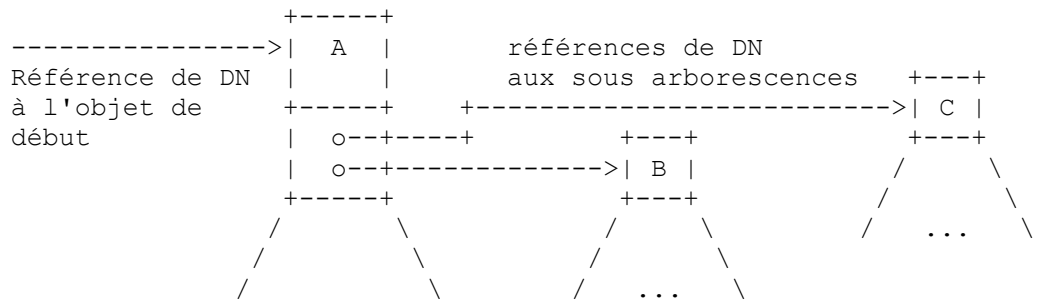


Figure 7 : Utilisation de pcimSubtreesPtrAuxClass pour localiser les politiques

Le PDP est configuré initialement avec une référence de DN à certaines entrées dans la DIT. La classe structurelle de cette entrée n'est pas importante ; le PDP ne s'intéresse qu'à la pcimSubtreesPtrAuxClass qui y est attachée. Cette classe auxiliaire contient un attribut multi valeurs avec des références de DN aux objets qui servent d'ancre aux sous arborescences qui contiennent des objets en rapport avec la politique intéressant le PDP. Comme pcimSubtreesPtrAuxClass est une classe auxiliaire, elle peut être rattachée à une entrée à laquelle le PDP aurait de toute façon besoin d'accéder – peut-être une entrée contenant les réglages de configuration initiale pour le PDP, ou pour un PEP qui utilise le PDP.

Une fois qu'il a restitué les références de DN, le PDP va diriger sur chacun des objets identifiés par elles une demande LDAP que toutes les entrées dans sa sous arborescence soient évaluées par rapport aux critères de choix spécifiés dans la demande. Le répertoire à capacité LDAP retourne alors toutes les entrées de cette sous arborescence qui satisfont les critères spécifiés.

Les critères de choix spécifient toujours que classe d'objet ="pcimPolicy". Comme toutes les classes qui représentent des règles de politique, des conditions de politique, et des actions de politique, aussi bien dans le PCLS que dans tout schéma spécifique d'un domaine qui en sont déduites, sont des sous classes de la classe abstraite Politique, ce critère s'évalue à VRAI pour toutes les instances de ces classes. Pour s'accommoder des cas particuliers où un PDP a besoin de restituer des objets qui ne sont pas

par eux-mêmes en rapport avec la politique (par exemple, un objet de gamme d'adresses IP référencé par une sous classe de pcimActionAuxClass représentant l'action DHCP "allouer à partir de cette gamme d'adresses") la classe auxiliaire pcimElementAuxClass peut être utilisée pour "étiqueter" une entrée, de sorte qu'elle va être trouvée par le critère de sélection "classe d'objet=pcimPolicy".

L'approche décrite dans le précédant paragraphe ne va pas fonctionner pour certaines mises en œuvre de répertoire, parce que ces mises en œuvre ne prennent pas en charge la confrontation des classes auxiliaires dans l'attribut objectClass. Pour des environnements où on s'attend à ce que ces mises en œuvre soient présentes, "l'étiquetage" des entrées comme pertinentes pour la politique peut être réalisé en insérant la valeur spéciale "POLICY" dans la liste des valeurs contenues dans l'attribut pcimKeywords (fourni par la classe pcimPolicy).

Si un PDP a seulement besoin d'un sous ensemble des objets en rapport avec la politique dans les sous arborescences indiquées, il peut alors être configuré avec des critères de sélection supplémentaires fondés sur l'attribut pcimKeywords défini dans la classe pcimPolicy. Cet attribut prend en charge des valeurs aussi bien standard que définies par l'administrateur. Par exemple, un PDP pourrait être configuré à ne demander que les objets en rapport avec la politique qui contiennent les mots-clés "DHCP" et "Est des USA".

Pour optimiser ce qui est attendu comme cas normal, la demande initiale du client inclut non seulement l'objet auquel son "germe" de DN fait référence, mais aussi la sous arborescence contenue sous cet objet. Le filtre pour chercher cette sous arborescence est ce que le client va utiliser plus tard pour chercher les autres sous arborescences : classe d' objet ="pcimPolicy" ou la présence du mot-clé "POLICY", et/ou la présence d'une valeur plus spécifique de pcimKeywords (par exemple, "Politique de QS de bordure").

Revenons à l' exemple de la Figure 7 : on voit que dans le meilleur des cas, un PDP peut obtenir tous les objets en rapport avec la politique dont il a besoin, et seulement ces objets, avec exactement trois demandes LDAP : une à son objet de départ A pour obtenir les références à B et C, ainsi que les objets en rapport avec la politique qu'il a besoin de tirer de la sous arborescence en dessous de A, et puis ensuite une à chacun de B et C pour obtenir tous les objets en rapport avec la politique qui passent les critères de sélection avec lesquels il a été configuré. Une fois qu'il a récupéré tous ces objets, le PDP peut alors traverser leurs diverses références de DN localement pour comprendre la sémantique des relations entre eux. Le PDP devrait aussi être prêt à trouver une référence à une autre sous arborescence rattachée à tout objet qu'il récupère, et à suivre cette référence en premier, avant qu'il suive une des références sémantiquement significatives qu'il a reçu. Cette récurrence permet une approche structurée pour identifier les politiques en rapport. Dans la Figure 7, par exemple, si la sous arborescence sous B inclut des politiques de département et celle sous C inclut des politiques de division, il peut alors y avoir une référence de la sous arborescence sous C à un objet D qui sert de racine à la sous arborescence des politiques de niveau entreprise.

Un PDP DEVRAIT comprendre la classe pcimSubtreesPtrAuxClass, DEVRAIT être capable de restituer et traiter les entrées dans les sous arborescences qu'il référence, et DEVRAIT être capable de faire tout cela de façon récurrente. Les mêmes exigences s'appliquent à toutes les autres entités qui ont besoin de restituer des informations de politique du répertoire. Donc, un outil de gestion de politique qui restitue des entrées de politique à partir du répertoire afin d'effectuer une validation et une détection de conflit DEVRAIT aussi comprendre et être capable d'utiliser la pcimSubtreesPtrAuxClass. Toutes ces exigences sont des "DEVRAIT" plutôt que des "DOIT" parce qu'un client LDAP qui ne les met pas en œuvre peut quand même y accéder et récupérer les entrées de répertoire dont il a besoin. Le processus pour le faire sera simplement moins efficace qu'il ne l'aurait été si le client avait mis en œuvre ces optimisations.

Lorsque il sert d'outil pour créer des entrées de politique dans le répertoire, un outil de gestion de politique DEVRAIT prendre en charge la création d’entrées de pcimSubtreesPtrAuxClass et de leurs références aux instances d’objets.

4.5.1 Alias et autres techniques d’optimisation du DIT

Une souplesse supplémentaire dans la structure de DIT est disponible à l’administrateur de politique via l’alias LDAP et d’autres techniques. Les versions précédentes du présent document ont utilisé des alias. Cependant, comme ils sont expérimentaux, l’utilisation d’alias a été retirée de la présente version du document. Cela parce que l’IETF doit déjà produire la spécification de la façon dont les alias sont représentés dans le répertoire ou comment les mises en œuvre de serveur vont traiter les alias.

5. Définitions de classe

La sémantique des classes d'informations de politique qui sont à transposer directement du modèle d'information en représentation LDAP est précisée dans la [RFC3060]. Par conséquent, tout ce que présente ce document pour ces classes est la spécification de la façon de faire la transposition du modèle d'information (qui est indépendante du type de répertoire et du protocole d'accès) en une forme à laquelle on puisse accéder en utilisant LDAP. On se rappelle que certaines nouvelles classes doivent être créées (qui ne font pas partie de la [RFC3060]) pour mettre en œuvre la transposition LDAP. Ces nouvelles classes qui ne relèvent que de LDAP sont pleinement documentées dans le présent document.

Le langage formel pour spécifier les classes, les attributs, et les règles de structure et de contenu du DIT sont celles définies dans la [RFC2252]. Si une mise en œuvre ne prend pas en charge l'héritage de classe auxiliaire, elle devra faire explicitement la liste des classes auxiliaires dans les règles de contenu ou les définir d'une autre façon (spécifique de la mise en œuvre).

Les notes qui suivent s'appliquent à l'ensemble de cette section.

Note 1 : dans les définitions qui suivent, les définitions de classe et d'attribut suivent la [RFC2252] mais elles utilisent des sauts à la ligne pour améliorer la lisibilité.

Note 2 : lorsque applicable, la possibilité de spécifier des règles de structure et contenu de DIT est notée. Cependant, il faut faire attention quand on spécifie des règles de structure de DIT. Comme [X.501] déclare qu'une entrée ne peut exister dans la DIT que comme subordonnée d'une autre entrée supérieure (le supérieur) si une règle de structure de DIT existe dans le sous schéma gouvernant qui :
1) indique une forme de nom pour la classe d'objet structurelle de l'entrée subordonnée, et
2) soit inclut la règle de structure supérieure de l'entrée comme règle de structure supérieure possible,
3) soit ne spécifie pas de règle de structure supérieure.
Si ce dernier cas (3) s'applique, l'entrée est alors définie comme étant un point administratif de sous schéma. Ce n'est pas ce qui est souhaité. Donc, il faut faire attention en définissant les règles de structure, et en particulier, elles doivent être augmentées en local.

Note 3 : chaque fois que possible, une règle de correspondance d'égalité et une règle de correspondance de sous chaîne sont toutes deux définies pour un attribut particulier (ainsi qu'une règle de correspondance d'ordre pour permettre le tri des résultats de la confrontation). Cela fournit deux choix différents au développeur pour une souplesse maximale.
Par exemple, si on considère l'attribut pcimRoles (paragraphe 5.3). Supposons qu'un PEP ait rapporté qu'il est intéressé par pcimRules pour les trois rôles R1, R2, et R3. Si le but est de minimiser les interrogations, alors le PDP peut fournir trois filtres de sous chaînes contenant les trois noms de rôle.
Ces interrogations vont toutes retourner les pcimRules qui s'appliquent au PEP, mais elles peuvent aussi en attraper qui ne s'appliquent pas (par exemple, celles qui contiennent certains des rôles R1, R2, ou R3 et un ou plusieurs autres rôles présents dans une combinaison de rôles [RFC3060]).
Une autre stratégie serait que le PDP utilise seulement des filtres d'égalité. Cette approche élimine les réponses étrangères, mais exige que le PDP construise explicitement lui-même les combinaisons de rôle désirées. Elle exige aussi des interrogations supplémentaires. Noter que cette approche n'est pratique que parce que les noms de rôle dans une combinaison de rôle doivent apparaître en ordre alphabétique.

Note 4 : dans les définitions qui suivent, noter que toutes les règles de correspondance LDAP sont définies dans la [RFC2252] et la [RFC3698]. Les règles de correspondance X.500 qui leur répondent sont définies dans [X.520].

Note 5 : certaines des définitions d'attribut suivantes spécifient des contraintes supplémentaires sur divers types de données (par exemple, cet entier a des valeurs valides de 1 à 10). Du texte a été ajouté pour donner des instructions aux serveurs et applications sur ce qu'il faut faire si ils rencontrent une valeur en dehors de cette gamme. Dans tous les cas, si une contrainte est violée, la règle de politique DEVRAIT alors être traitée comme ayant été désactivée, ce qui signifie que l'exécution de la règle de politique DEVRAIT être arrêtée.

5.1 Classe abstraite pcimPolicy

La classe abstraite pcimPolicy est une transposition directe de la classe abstraite Policy provenant de PCIM. La valeur de classe "pcimPolicy" est aussi utilisée comme mécanisme pour identifier les instances en rapport avec la politique dans l'arborescence d'information du répertoire (DIT, *Directory Information Tree*). Une instance de toute classe peut être"étiquetée" avec la valeur de cette classe en l'attachant à la classe auxiliaire pcimElementAuxClass. Comme pcimPolicy est dérivée de la classe dlm1ManagedElement définie dans [DTMF], la présente spécification a une dépendance normative à [DTMF] sur cet élément. La définition de classe est la suivante :

(1.3.6.1.1.6.1.1 NOM 'pcimPolicy'

DESCRIPTION : Classe abstraite qui est la classe de base pour toutes les classes qui décrivent des instances qui se rapportent à la politique
SUPÉRIEURE : dlm1ManagedElement
ABSTRAITE
PEUT (cn $ dlmCaption $ dlmDescription $ orderedCimKeys $ pcimKeywords)
)

L'attribut cn est défini dans la [RFC2256]. Les attributs dlmCaption, dlmDescription, et orderedCimKeys sont définis dans [DTMF].

L'attribut pcimKeywords est un attribut multi valeurs qui contient un ensemble de mots-clés pour aider les clients de répertoire à localiser les objets de politique identifiés par ces mots-clés. Il est défini comme suit :

(1.3.6.1.1.6.2.3 NOM 'pcimKeywords'
DESCRIPTION : Ensemble de mots-clés pour aider les clients de répertoire à localiser les objets de politique qui leur sont applicables.
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.15
)

5.2 Les trois classes de groupe de politique

PCIM [RFC3060] définit la classe PolicyGroup pour servir de mécanisme d'agrégation généralisé, permettant que PolicyRules et/ou PolicyGroups soient agrégés ensemble. PCLS transpose cette classe en trois classes LDAP, appelées pcimGroup, pcimGroupAuxClass, et pcimGroupInstance. On fait cela afin de fournir le maximum de souplesse au concepteur de DIT.

Les définitions de classe pour les trois classes de groupe de politique sont données ci-dessous. Ces définitions de classe n'incluent pas d'attribut pour réaliser les associations PolicyRuleInPolicyGroup et PolicyGroupInPolicyGroup de PCIM. C'est parce que un objet pcimGroup se réfère aux instances de pcimGroup et pcimRule via, respectivement, l'attribut pcimGroupsAuxContainedSet dans la classe d'objet pcimGroupContainmentAuxClass et l'attribut pcimRulesAuxContainedSet dans la classe d'objet pcimRuleContainmentAuxClass.

Pour maximiser la souplesse, la classe pcimGroup est définie comme abstraite. La sous classe pcimGroupAuxClass fournit un rattachement auxiliaire à une autre entrée, tandis que la sous classe structurelle pcimGroupInstance est disponible pour représenter un groupe de politique comme une entrée autonome. Les définitions de classe sont comme suit. D'abord, la définition de la classe abstraite pcimGroup :

(1.3.6.1.1.6.1.2 NOM 'pcimGroup'
DESCRIPTION : conteneur pour un ensemble de pcimRules en rapports et/ou un ensemble de pcimGroups en rapports
SUPÉRIEURE : pcimPolicy
ABSTRAITE
PEUT (pcimGroupName)
)

L'attribut de pcimGroup est pcimGroupName. Cet attribut est utilisé pour définir un nom facile à prononcer de ce groupe de politique, et peut être utilisé comme attribut de dénomination si désiré. Il est défini comme suit :

(1.3.6.1.1.6.2.4 NOM 'pcimGroupName'
DESCRIPTION : nom facile à prononcer de ce groupe de politique
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.15
MONO-VALEUR
)

Les deux sous classes de pcimGroup sont définies comme suit. La classe pcimGroupAuxClass est une classe auxiliaire qui peut être utilisée pour collecter un ensemble de classes pcimRule et/ou pcimGroup en rapports. Elle est définie comme suit :

(1.3.6.1.1.6.1.3 NOM 'pcimGroupAuxClass'

DESCRIPTION : classe auxiliaire qui collecte un ensemble d'entrées de pcimRule et/ou pcimGroup en rapports
SUPÉRIEURE : pcimGroup
AUXILIAIRE
)

La classe pcimGroupInstance est une classe structurelle qui peut être utilisée pour collecter un ensemble de classes pcimRule et/ou pcimGroup en rapports. Elle est définie comme suit :

(1.3.6.1.1.6.1.4 NOM 'pcimGroupInstance'
DESCRIPTION : classe structurelle qui collecte un ensemble d'entrées pcimRule et/ou pcimGroup en rapports
SUPÉRIEURE : pcimGroup
STRUCTURELLE
)

On pourrait écrire une règle de contenu de DIT pour permettre d'y rattacher une instance de pcimGroupInstance par des références à un ou plusieurs groupes de politique (en utilisant pcimGroupContainmentAuxClass) ou des références à une ou plusieurs règles de politique (en utilisant pcimRuleContainmentAuxClass). Ceci serait utilisé pour formaliser la sémantique de la classe PolicyGroup [RFC3060]. Comme cette sémantique n'inclut pas de spécifier de propriétés de la classe PolicyGroup, la règle de contenu n'aurait pas besoin de spécifier d'attributs.

De même, trois règles de structure de DIT séparées pourraient être écrites dont chacune se réfèrerait à une forme de nom spécifique qui identifierait un des trois attributs de désignation spécifiques (c'est-à-dire, pcimGroupName, cn, et orderedCIMKeys) pour la classe d'objet pcimGroup. Cette règle de structure DEVRAIT inclure une superiorStructureRule (voir la note 2 au début de la section 5). Les trois formes de nom référencées par les trois règles de structure définiraient chacune un des trois attributs de désignation.

5.3 Les trois classes de règle de politique

Le modèle d'information définit une classe PolicyRule pour représenter la sémantique "Si condition, alors action" associée aux traitement des informations de politique. Pour une souplesse maximum, le PCLS transpose cette classe en trois classes LDAP.

Pour maximiser la souplesse, la classe pcimRule est définie comme abstraite. La sous classe pcimRuleAuxClass assure un rattachement auxiliaire à une autre entrée, tandis que la sous classe structurelle pcimRuleInstance est disponible pour représenter une règle de politique comme entrée autonome.

Les conditions et actions associées à une règle de politique sont modélisées, respectivement, avec des sous classes auxiliaires des classes auxiliaires pcimConditionAuxClass et pcimActionAuxClass. Chacune de ces sous classes auxiliaires est rattachée à une instance d'une des trois classes structurelles. Une sous classe de pcimConditionAuxClass est rattachée à une instance de pcimRuleInstance, à une instance de pcimRuleConditionAssociation, ou à une instance de pcimPolicyInstance. De même, une sous classe de pcimActionAuxClass est rattachée à une instance de pcimRuleInstance, à une instance de pcimRuleActionAssociation, ou à une instance de pcimPolicyInstance.

L'attribut pcimRuleValidityPeriodList (défini ci-dessous) réalise l'association PolicyRuleValidityPeriod définie dans le PCIM. Comme cette association n'a pas de propriétés supplémentaires à côté de celles qui lient l'association à ses objets associés, cette association peut être réalisée en utilisant simplement un attribut. Donc, l'attribut pcimRuleValidityPeriodList est simplement un attribut multi valeurs qui fournit un ensemble non ordonné de références de DN à une ou plusieurs instances de la pcimTPCAuxClass, indiquant quand la règle de politique est programmée pour être active et quand elle est programmée pour être inactive. Une règle de politique est programmée pour être active si elle est active conformément à AU MOINS UNE des instances de pcimTPCAuxClass référencées par cet attribut.

Les associations PolicyConditionInPolicyRule et PolicyActionInPolicyRule ont cependant des attributs supplémentaires. L'association PolicyActionInPolicyRule définit un attribut entier pour mettre à la suite les actions, et l'association PolicyConditionInPolicyRule a à la fois un attribut entier pour grouper les termes de condition et une propriété booléenne pour spécifier si une condition doit être niée.

Dans le PCLS, ces attributs supplementaires d'association sont représentés comme des attributs de deux classes introduites spécifiquement pour modéliser ces associations. Ces classes sont la classe pcimRuleConditionAssociation et la classe pcimRuleActionAssociation, qui sont définies respectivement aux paragraphes 5.4 et 5.5. Donc, ils n'apparaissent pas comme attributs de la classe pcimRule. Les attributs pcimRuleConditionList et pcimRuleActionList peuvent être utilisés à la place pour référencer ces classes.

Les définitions de classe pour les trois classes pcimRule sont les suivantes.

La classe abstraite pcimRule est une classe de base pour représenter la sémantique "Si condition, alors action" associée à une règle de politique. Elle est définie comme suit :

(1.3.6.1.1.6.1.5 NOM 'pcimRule'
DESCRIPTION : classe de base pour représenter la sémantique "Si condition, alors action" associée à une règle de politique
SUPÉRIEURE : pcimPolicy
ABSTRAITE
PEUT : (pcimRuleName $ pcimRuleEnabled $ pcimRuleConditionListType $ pcimRuleConditionList $ pcimRuleActionList
$ pcimRuleValidityPeriodList $ pcimRuleUsage $ pcimRulePriority $ pcimRuleMandatory $
pcimRuleSequencedActions $ pcimRoles)
)

PCIM [RFC3060] définit sept propriétés pour la classe PolicyRule. Le PCLS définit onze attributs pour la classe pcimRule, qui est l'équivalent LDAP de la classe PolicyRule. De ces onze attributs, sept sont transposés directement des propriétés correspondantes dans la classe PolicyRule de PCIM. Les quatre attributs restants sont un attribut facultatif de désignation spécifique de la classe, et trois attributs utilisés pour réaliser les trois associations auxquelles participe la classe pcimRule.

L'attribut pcimRuleName est utilisé comme nom facile à prononcer de cette règle de politique, et peut aussi servir d'attribut facultatif de désignation spécifique de la classe. Il est défini comme suit :

(1.3.6.1.1.6.2.5 NOM 'pcimRuleName'
DESCRIPTION : nom facile à retenir de cette règle de politique
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.15
MONO-VALEUR
)

L'attribut pcimRuleEnabled est une énumération d'entiers qui indique si une règle de politique est activée administrativement (valeur=1), désactivée administrativement (valeur=2), ou activée pour débogage (valeur=3). Il est défini comme suit :

(1.3.6.1.1.6.2.6 NOM 'pcimRuleEnabled'
DESCRIPTION : entier qui indique si une règle de politique est activée administrativement (valeur=1), désactivée (valeur=2),
ou activée pour débogage (valeur=3)
ÉGALITÉ : integerMatch
ORDRE : integerOrderingMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.27
MONO-VALEUR
)

Note : Toutes les autres valeurs de l'attribut pcimRuleEnabled sont considérées comme des erreurs, et l'administrateur DEVRAIT traiter cette règle comme désactivée si une valeur invalide est trouvée.

L'attribut pcimRuleConditionListType est utilisé pour indiquer si la liste de conditions de politique associées à cette règle de politique est en forme disjonctive normale (DNF, valeur=1) ou conjonctive normale (CNF, valeur=2). Il est défini comme suit :

(1.3.6.1.1.6.2.7 NOM 'pcimRuleConditionListType'
DESCRIPTION : une valeur de 1 signifie que cette règle de politique est en forme disjonctive normale ; une valeur de 2
signifie que cette règle de politique est en forme conjonctive normale
ÉGALITÉ : integerMatch
ORDRE : integerOrderingMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.27
MONO-VALEUR
)

Note : Toute valeur autre que 1 ou 2 pour l'attribut pcimRuleConditionListType est considérée comme erreur. Les administrateurs DEVRAIENT traiter cette règle comme désactivée si une valeur invalide est trouvée, car la structure à donner à cette liste de conditions n'est pas claire.

L'attribut pcimRuleConditionList est un attribut multi valeurs qui est utilisé pour réaliser l'association policyRuleInPolicyCondition définie dans la [RFC3060]. Il contient un ensemble de DN d'entrées de

pcimRuleConditionAssociation qui représentent les associations entre cette règle de politique et ses conditions. Aucun ordre n'est impliqué. Il est défini comme suit :

(1.3.6.1.1.6.2.8 NOM 'pcimRuleConditionList'
DESCRIPTION : ensemble non ordonné de DN d'entrées de pcimRuleConditionAssociation qui représentent les associations entre cette règle de politique et ses conditions
ÉGALITÉ : distinguishedNameMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.12
)

L'attribut pcimRuleActionList est un attribut multi valeurs qui est utilisé pour réaliser l'association policyRuleInPolicyAction définie dans la [RFC3060]. Il contient un ensemble de DN d'entrées de pcimRuleActionAssociation qui représentent les associations entre cette règle de politique et ses actions. Aucun ordre n'est impliqué. Il est défini comme suit :

(1.3.6.1.1.6.2.9 NOM 'pcimRuleActionList'
DESCRIPTION : ensemble non ordonné de DN d'entrées de pcimRuleActionAssociation qui représentent les associations entre cette règle de politique et ses actions
ÉGALITÉ : distinguishedNameMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.12
)

L'attribut pcimRuleValidityPeriodList est un attribut multi valeurs qui est utilisé pour réaliser l'association pcimRuleValidityPeriod qui est définie dans la [RFC3060]. Il contient un ensemble de DN d'entrées de pcimRuleValidityAssociation qui déterminent quand l'activité ou l'inactivité de pcimRule est programmée. Aucun ordre n'est impliqué. Il est défini comme suit :

(1.3.6.1.1.6.2.10 NOM 'pcimRuleValidityPeriodList'
DESCRIPTION : ensemble non ordonné de DN d'entrées de pcimRuleValidityAssociation qui déterminent quand la pcimRule est programmée pour être active ou inactive
ÉGALITÉ : distinguishedNameMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.12
)

L'attribut pcimRuleUsage est une chaîne de forme libre qui fournit des lignes directrices sur la façon dont cette politique devrait être utilisée. Il est défini comme suit :

(1.3.6.1.1.6.2.11 NOM 'pcimRuleUsage'
DESCRIPTION : cet attribut est une chaîne de forme libre qui donne des lignes directrices sur la façon dont cette politique devrait être utilisée
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.15
MONO-VALEUR
)

L'attribut pcimRulePriority est un entier non négatif qui est utilisé pour donner la priorité à cette pcimRule par rapport aux autres pcimRules. Une plus grande valeur indique une priorité supérieure. Il est défini comme suit :

(1.3.6.1.1.6.2.12 NOM 'pcimRulePriority'
DESCRIPTION : entier non négatif pour donner la priorité à cette pcimRule par rapport aux autres pcimRules. Une plus grande valeur indique une priorité supérieure
ÉGALITÉ : integerMatch
ORDRE : integerOrderingMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.27
MONO-VALEUR
)

Note : si la valeur du champ pcimRulePriority est 0, elle DEVRAIT alors être traitée comme "ne pas tenir compte". D'un autre côté, si la valeur est négative, elle DEVRAIT alors être traitée comme une erreur et les administrateurs DEVRAIENT traiter cette règle comme étant désactivée.

L'attribut pcimRuleMandatory est un attribut booléen qui, si il est VRAI, indique que pour cette règle de politique, l'évaluation

de ses conditions et l'exécution de ses actions (si la condition est satisfaite) sont exigées. Si il est FAUX, l'évaluation de ses conditions et l'exécution de ses actions (si la condition est satisfaite) n'est alors pas exigée. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.13 NOM 'pcimRuleMandatory'
DESCRIPTION : si VRAI, indique que pour cette règle de politique, l'évaluation de ses conditions et l'exécution de ses actions (si la condition est satisfaite) est requise
ÉGALITÉ : booleanMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.7
MONO-VALEUR
)

L'attribut pcimRuleSequencedActions est une énumération d'entiers qui est utilisée pour indiquer que l'ordre des actions défini par l'attribut pcimActionOrder est soit obligatoire (valeur=1), soit recommandée (valeur=2), soit ne pas tenir compte (valeur=3). Il est défini comme suit :

(1.3.6.1.1.6.2.14 NOM 'pcimRuleSequencedActions'
DESCRIPTION : énumération d'entiers qui indique que l'ordre des actions défini par l'attribut pcimActionOrder est obligatoire (1), recommandé (2), ou ne pas tenir compte (3)
ÉGALITÉ : integerMatch
ORDRE : integerOrderingMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.27
MONO-VALEUR
)

Note : si la valeur du champ pcimRulesSequencedActions n'est pas une de ces trois valeurs, les administrateurs DEVRAIENT alors traiter cette règle comme étant désactivée.

L'attribut pcimRoles représente la propriété policyRoles de la [RFC3060]. Chaque valeur de cet attribut représente une combinaison de rôles, qui est une chaîne de la forme : <NomDeRôle>[&&<NomDeRôle>]* où les noms des rôles individuels apparaissent en ordre alphabétique conformément à la séquence de collationnement de UCS-2. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.15 NAME 'pcimRoles'
DESCRIPTION : chaque valeur de cet attribut représente une combinaison de rôle
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.15
)

Note : si la valeur de l'attribut pcimRoles ne se conforme pas au format "<NomDeRôle>[&&<NomDeRôle>]*" (voir au paragraphe 6.3.7 de la [RFC3060]) cet attribut est alors malformé et sa règle de politique DEVRAIT être traitée comme désactivée.

Les deux sous classes de la classe pcimRule sont définies comme suit. D'abord, pcimRuleAuxClass est une classe auxiliaire pour représenter la sémantique "Si condition, alors action" associée à une règle de politique. Sa définition de classe est la suivante :

(1.3.6.1.1.6.1.6 NAME 'pcimRuleAuxClass'
DESCRIPTION : classe auxiliaire pour représenter la sémantique "Si condition, alors action" associée à une règle de politique
SUPÉRIEURE : pcimRule
AUXILIAIRE
)

pcimRuleInstance est une classe structurelle pour représenter la sémantique "Si condition, alors action" associée à une règle de politique. Sa définition de classe est la suivante :

(1.3.6.1.1.6.1.7 NOM 'pcimRuleInstance'
DESCRIPTION : classe structurelle pour représenter la sémantique "Si condition, alors action" associée à une règle de politique
SUPÉRIEURE : pcimRule
STRUCTURELLE
)

Une règle de contenu de DIT pourrait être écrite pour permettre à une instance de pcimRuleInstance d'avoir rattachée à elle soit des références à une ou plusieurs conditions de politique (en utilisant pcimConditionAuxClass) soit des références à une ou plusieurs actions de politique (en utilisant pcimActionAuxClass). Cela serait utilisé pour formaliser la sémantique de la classe PolicyRule [RFC3060]. Comme cette sémantique n'inclut pas de spécifier de propriétés de la classe PolicyRule, la règle de contenu n'aurait pas besoin de spécifier d'attributs.

De même, trois règles de structure de DIT séparées pourraient être écrites, dont chacune se réfèrerait à une forme de nom spécifique qui identifierait un de ses trois attributs de dénomination possibles (c'est-à-dire, pcimRuleName, cn, et orderedCIMKeys). Cette règle de structure DEVRAIT inclure une superiorStructureRule (voir la note 2 au début de la section 5). Les trois formes de nom référencées par les trois règles de structure définiraient chacune un des trois attributs de dénomination.

5.4 Classe pcimRuleConditionAssociation

Cette classe contient des attributs pour représenter les propriétés de l'association PolicyConditionInPolicyRule de PCIM. Les instances de cette classe sont en relation avec une instance de pcimRule via l'incorporation dans la DIT. Les conditions de politique elles-mêmes sont représentées par des sous classes auxiliaires de la classe auxiliaire pcimConditionAuxClass. Ces classes auxiliaires sont rattachées directement aux instances de pcimRuleConditionAssociation pour les conditions de politique spécifiques de règle. Pour une condition de politique réutilisable, la sous classe auxiliaire policyCondition est rattachée à une instance de la classe pcimPolicyInstance (qui est présumée associée à une pcimRepository par l'incorporation dans la DIT) et l'attribut policyConditionDN (de cette classe) est utilisé pour référencer l'instance réutilisable policyCondition. La définition de classe est la suivante :

(1.3.6.1.1.6.1.8 NOM 'pcimRuleConditionAssociation'
DESCRIPTION : cette classe contient des attributs caractérisant la relation entre une règle de politique et une de ses conditions de politique
SUPÉRIEURE : pcimPolicy
DOIT (pcimConditionGroupNumber $ pcimConditionNegated)
PEUT (pcimConditionName $ pcimConditionDN)
)

Les attributs de cette classe sont définis comme suit.

L'attribut pcimConditionGroupNumber est un entier non négatif. Il est utilisé pour identifier le groupe auquel la condition référencée par cette association est allouée. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.16 NOM 'pcimConditionGroupNumber'
DESCRIPTION : numéro du groupe auquel une condition de politique appartient. C'est utilisé pour former l'expression de DNF ou de CNF associée à une règle de politique
ÉGALITÉ : integerMatch
ORDRE : integerOrderingMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.27
MONO-VALEUR
)

Noter que ce numéro est non négatif. Une valeur négative pour cet attribut est invalide, et toute règle de politique qui se réfère à une entrée invalide DEVRAIT être traitée comme désactivée.

L'attribut pcimConditionNegated est un attribut booléen qui indique si cette condition de politique est à nier ou non. Si elle est VRAI (FAUX), cela indique qu'une condition de politique EST (N'EST PAS) niée dans l'expression de DNF ou de CNF associée à une règle de politique. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.17 NOM 'pcimConditionNegated'
DESCRIPTION : si VRAI (FAUX) cela indique qu'une condition de politique EST (N'EST PAS) niée dans l'expression de DNF ou de CNF associée à une règle de politique
ÉGALITÉ : booleanMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.7
MONO-VALEUR
)

Le nom pcimConditionName est un nom facile à retenir pour identifier cette condition de politique, et peut être utilisé comme

attribut de dénomination si désiré. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.18 NOM 'pcimConditionName'
DESCRIPTION : nom facile à retenir pour une condition de politique
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.15
MONO-VALEUR
)

L'attribut pcimConditionDN est un DN qui fait référence à une instance d'une condition de politique réutilisable. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.19 NOM 'pcimConditionDN'
DESCRIPTION : DN qui fait référence à une instance d'une condition de politique réutilisable
ÉGALITÉ : distinguishedNameMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.12
MONO-VALEUR
)

Une règle de contenu de DIT pourrait être écrite pour permettre qu'une instance de pcimRuleConditionAssociation ait une instance rattachée de la classe auxiliaire pcimConditionAuxClass, ou une de ses sous classes. Cela serait utilisé pour formaliser la sémantique de l'association PolicyConditionInPolicyRule. Précisément, cela serait utilisé pour représenter une condition de politique spécifique d'une règle [RFC3060]. De même, trois règles de structure de DIT séparées pourraient être écrites. Chacune de ces règles de structure de DIT se réfèrerait à une forme de nom spécifique qui définit deux sémantiques importantes. D'abord, chaque forme de nom identifierait un des trois attributs de dénomination spécifiques (c'est-à-dire, pcimConditionName, cn, et orderedCIMKeys) pour la classe d'objet pcimRuleConditionAssociation. Ensuite, chaque forme de nom exigerait qu'une instance de la classe pcimRuleConditionAssociation ait comme supérieur une instance de la classe pcimRule. Cette règle de structure DEVRAIT aussi inclure une superiorStructureRule (voir la note 2 au début de la section 5).

5.5 Classe pcimRuleValidityAssociation

L'agrégationValidityPeriod est transposée en la classe PCLS pcimRuleValidityAssociation. Cette classe représente l'activation et la désactivation programmées d'une règle de politique en liant la définition des fois où la politique est active à la règle de politique elle-même. Les fois "programmées" sont soit identifiées à travers une classe auxiliaire pcimTPCAuxClass rattachée, soit référencées à travers leur attribut pcimTimePeriodConditionDN. Cette classe est définie comme suit :

(1.3.6.1.1.6.1.9 NOM 'pcimRuleValidityAssociation'
DESCRIPTION : cela définit l'activation ou la désactivation programmée d'une règle de politique
SUPÉRIEURE : pcimPolicy
STRUCTURELLE
PEUT (pcimValidityConditionName $ pcimTimePeriodConditionDN)
)

Les attributs de cette classe sont définis comme suit :

L'attribut pcimValidityConditionName est utilisé pour définir un nom facile à retenir de cette condition, et peut être utilisé comme attribut de dénomination si désiré. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.20 NOM 'pcimValidityConditionName'
DESCRIPTION : nom facile à retenir pour identifier une instance d'une entrée de pcimRuleValidityAssociation
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.15
MONO-VALEUR
)

L'attribut pcimTimePeriodConditionDN est un DN qui fait référence à une condition de période de temps réutilisable. Il est défini comme suit :

(1.3.6.1.1.6.2.21 NOM 'pcimTimePeriodConditionDN'
DESCRIPTION : référence à une condition de période de temps de politique réutilisable
ÉGALITÉ : distinguishedNameMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.12
MONO-VALEUR
)

Une règle de contenu de DIT pourrait être écrite pour permettre à une instance de pcimRuleValidityAssociation d'avoir rattachée à elle une instance de la classe auxiliaire pcimTPCAuxClass, ou une de ses sous classes. Cela serait utilisé pour formaliser la sémantique de l'agrégation PolicyRuleValidityPeriod [RFC3060].

De même, trois règles de structure de DIT séparées pourraient être écrites. Chacune de ces règles de structure de DIT se réfèrerait à une forme de nom spécifique qui définit deux sémantiques importantes. D'abord, chaque forme de nom identifierait un des trois attributs de dénomination possibles (c'est-à-dire, pcimValidityConditionName, cn, et orderedCIMKeys) pour la classe d'objet pcimRuleValidityAssociation. Ensuite, chaque forme de nom exigerait qu'une instance de la classe pcimRuleValidityAssociation ait comme supérieur une instance de la classe pcimRule. Cette règle de structure DEVRAIT aussi inclure une superiorStructureRule (voir la note 2 au début de la section 5).

5.6 Classe pcimRuleActionAssociation

Cette classe contient un attribut pour représenter la seule propriété de l'association PCIM PolicyActionInPolicyRule, ActionOrder. Cette propriété est utilisée pour spécifier un ordre d'exécution des actions associées à une règle de politique. Les instances de cette classe se rapportent à une instance de pcimRule via l'incorporation au DIT. Les actions elles-mêmes sont représentées par des sous classes auxiliaires de la classe auxiliaire pcimActionAuxClass.

Ces classes auxiliaires sont rattachées directement aux instances de pcimRuleActionAssociation pour les actions de politique spécifiques d'une règle. Pour une action de politique réutilisable, la sous classe auxiliaire pcimAction est rattachée à une instance de la classe pcimPolicyInstance (qui est présumée associée à un pcimRepository par l'incorporation à la DIT) et l'attribut pcimActionDN (de cette classe) est utilisé pour référencer l'instance réutilisable pcimCondition. La définition de classe est la suivante :

(1.3.6.1.1.6.1.10 NOM 'pcimRuleActionAssociation'
DESCRIPTION : cette classe contient des attributs qui caractérisent la relation entre une règle de politique et une de ses actions de politique
SUPÉRIEURE : pcimPolicy
DOIT (pcimActionOrder)
PEUT (pcimActionName $ pcimActionDN)
)

L'attribut pcimActionName est utilisé pour définir un nom facile à retenir de cette action, et peut être utilisé comme attribut de dénomination si désiré. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.22 NOM 'pcimActionName'
DESCRIPTION : nom facile à retenir pour une action de politique
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.15
MONO-VALEUR
)

L'attribut pcimActionOrder est un entier non signé qui est utilisé pour indiquer la position relative d'une action dans une séquence d'actions qui sont associées à une certaine règle de politique. Lorsque ce nombre est positif, il indique une place dans la séquence d'actions à effectuer, les plus petites valeurs indiquant une position antérieure dans la séquence. Si la valeur est zéro, cela indique alors que l'ordre n'est pas pertinent. Noter que si deux actions ou plus ont la même valeur non zéro, elles peuvent être effectuées dans n'importe que ordre pour autant qu'elles soient chacune effectuées à l'endroit correct dans la séquence d'actions globale. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.23 NOM 'pcimActionOrder'
DESCRIPTION : entier qui indique l'ordre relatif d'une action dans le contexte d'une règle de politique
ÉGALITÉ : integerMatch
ORDRE : integerOrderingMatch

SYNTAXE: 1.3.6.1.4.1.1466.115.121.1.27
MONO-VALEUR
)

Note : si la valeur du champ pcimActionOrder est négatif, elle DEVRAIT alors être traitée comme une erreur et toute règle de politique qui se réfère à une telle entrée DEVRAIT être traitée comme désactivée.

L'attribut pcimActionDN est un DN qui fait référence à une action de politique réutilisable. Il est défini comme suit :

(1.3.6.1.1.6.2.24 NOM 'pcimActionDN'
DESCRIPTION : DN qui fait référence à une action de politique réutilisable
ÉGALITÉ : distinguishedNameMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.12
MONO-VALEUR
)

Une règle de contenu de DIT pourrait être écrite pour permettre à une instance de pcimRuleActionAssociation d'avoir rattachée à elle une instance de la classe auxiliaire pcimActionAuxClass, ou une de ses sous classes. Cela serait utilisé pour formaliser la sémantique de l'association PolicyActionInPolicyRule. Précisément, cela serait utilisé pour représenter une action de politique spécifique d'une règle [RFC3060].

De même, trois règle de structure de DIT séparées pourraient être écrites. Chacune de ces règles de structure de DIT se réfèrerait à une forme de nom spécifique qui définit deux sémantiques importantes. D'abord, chaque forme de nom identifierait un des trois attributs de dénomination possibles (c'est-à-dire, pcimActionName, cn, et orderedCIMKeys) pour la classe d'objet pcimRuleActionAssociation. Ensuite, chaque forme de nom exigerait qu'une instance de la classe pcimRuleActionAssociation ait comme supérieur une instance de la classe pcimRule. Cette règle de structure devrait aussi inclure une superiorStructureRule (voir la note 2 au début de la section 5).

5.7 Classe auxiliaire pcimConditionAuxClass

L'objet d'une condition de politique est de déterminer si l'ensemble des actions (contenues dans la pcimRule à laquelle s'applique la condition) devrait ou non être exécuté. Cette classe définit la sémantique organisationnelle de base d'une condition de politique, comme spécifié dans la [RFC3060]. Les sous classes de cette classe auxiliaire peuvent être rattachées aux instances de trois autres classes dans le PCLS. Lorsque une sous classe de cette classe est rattachée à une instance de pcimRuleConditionAssociation, ou à une instance de pcimRule, elle représente une condition de politique spécifique d'une règle. Lorsque une sous classe de cette classe est rattachée à une instance de pcimPolicyInstance, elle représente une condition de politique réutilisable.

Comme toutes les classes auxquelles les sous classes de cette classe auxiliaire peuvent être rattachées sont dérivées de la classe pcimPolicy, les attributs de pcimPolicy vont déjà être définis pour les entrées auxquelles ces sous classes se rattachent. Donc, cette classe est dérivée directement du "sommet". La définition de classe est la suivante :

(1.3.6.1.1.6.1.11 NOM 'pcimConditionAuxClass'
DESCRIPTION : classe qui représente une condition à évaluer en conjonction avec une règle de politique
SUPÉRIEURE : sommet
AUXILIAIRE
)

5.8 Classe auxiliaire pcimTPCAuxClass

Le PCIM définit une classe de période de temps, PolicyTimePeriodCondition, pour fournir un moyen de représenter les périodes de temps durant lesquelles une règle de politique est valide, c'est-à-dire, active. Il définit aussi une agrégation, PolicyRuleValidityPeriod, afin que ces périodes de temps puissent être associées à une PolicyRule. La transposition LDAP fournit aussi deux classes, une pour la condition de temps elle-même, et une pour l'agrégation.

Dans le PCIM, la classe période de temps est appelée PolicyTimePeriodCondition. Cependant, le nom résultant de la classe auxiliaire dans cette transposition (pcimTimePeriodConditionAuxClass) excède la longueur des noms que certains répertoires peuvent mémoriser. Donc, le nom a été abrégé en pcimTPCAuxClass. La définition de classe est la suivante :

(1.3.6.1.1.6.1.12 NOM 'pcimTPCAuxClass'
DESCRIPTION : cela donne la capacité d'activer ou désactiver une règle de politique selon un programme prédéterminé
SUPÉRIEURE : pcimConditionAuxClass

AUXILIAIRE
PEUT (pcimTPCTime $ pcimTPCMonthOfYearMask $ pcimTPCDayOfMonthMask $ pcimTPCDayOfWeekMask $ pcimTPCTimeOfDayMask $ pcimTPCLocalOrUtcTime)
)

Les attributs de pcimTPCAuxClass sont définis comme suit.

L'attribut pcimTPCTime représente la période de temps pendant laquelle une règle de politique est activée. Cet attribut est défini comme une chaîne dans la [RFC3060] avec un format spécial qui définit une période de temps avec une date de début et une date de fin séparées par une barre oblique ("/"), comme suit : aaaammjjThhmmss/aaaammjjThhmmss, où la première date et heure peut être remplacée par la chaîne "CECIETAVANT" ou la seconde date et heure peut être remplacée par la chaîne "CECI ETFUTUR". Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.25 NOM 'pcimTPCTime'
DESCRIPTION : heure de début et de fin de la période sur laquelle une règle de politique est valide
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.44
MONO-VALEUR
)

La valeur de cet attribut DEVRAIT être vérifiée par rapport au format défini ("aaaammjjThhmmss/aaaymmjjThhmmss", où la première et la seconde chaînes de date peuvent être remplacés par la chaîne "CECIETAVANT" et "CECI ETFUTUR"). Si la valeur de cet attribut ne se conforme pas à cette syntaxe, ce DEVRAIT alors être considéré comme une erreur et la règle de politique DEVRAIT être traitée comme désactivée.

Les quatre prochains attributs (pcimTPCMonthOfYearMask, pcimTPCDayOfMonthMask, pcimTPCDayOfWeekMask, et pcimTPCTimeOfDayMask) sont tous définis comme des chaînes d'octets dans la [RFC3060]. Cependant, la sémantique de chacun de ces attributs est contenue dans des chaînes binaires de longueur fixe diverses. Donc, le PCLS utilise une syntaxe de Chaîne binaire pour représenter chacun d'eux. La définition de ces quatre attributs est la suivante.

L'attribut pcimTPCMonthOfYearMask définit un gabarit de 12 bits qui identifie les mois de l'année dans lesquels une règle de politique est valide. Le format est une chaîne binaire de longueur 12, représentant les mois de l'année de janvier à décembre. La définition de cet attribut est la suivante :

(1.3.6.1.1.6.2.26 NOM 'pcimTPCMonthOfYearMask'
DESCRIPTION : ceci identifie les mois valides dans l'année pour une règle de politique en utilisant une chaîne de 12 bits qui représente les mois de l'année de janvier à décembre
ÉGALITÉ : bitStringMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.6
MONO-VALEUR
)

La valeur de cet attribut DEVRAIT être vérifiée par rapport au format défini. Si la valeur de cet attribut ne se conforme pas à cette syntaxe, ceci DEVRAIT alors être considéré comme une erreur et la règle de politique DEVRAIT être traitée comme désactivée.

L'attribut pcimTPCMonthOfDayMask définit un gabarit qui identifie les jours du mois pendant lesquels une règle de politique est valide. Le format est une chaîne binaire de longueur 62. Les 31 premières positions représentent les jours du mois en ordre ascendant du premier jour au trente et unième. Les 31 positions suivantes représentent les jours du mois en ordre descendant,du dernier jour au trente et unième jour en partant de la fin. La définition de cet attribut est la suivante :

(1.3.6.1.1.6.2.27 NOM 'pcimTPCDayOfMonthMask'
DESCRIPTION : cela identifie les jours valides du mois pour une règle de politique en utilisant une chaîne de 62 bits. Les 31 premières positions représentent les jours du mois en ordre ascendant, et les 31 positions suivantes représentent les jours du mois dans l'ordre descendant
ÉGALITÉ : bitStringMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.6
MONO-VALEUR
)

La valeur de cet attribut DEVRAIT être vérifiée par rapport à son format défini. Si la valeur de cet attribut n'est pas conforme à

cette syntaxe, cela DEVRAIT alors être considéré comme une erreur et la règle de politique DEVRAIT être traitée comme désactivée.

L'attribut pcimTPCDayOfWeekMask définit un gabarit qui identifie les jours de la semaine pendant lesquels une règle de politique est valide. Le format est une chaîne binaire de longueur 7, représentant les jours de la semaine du dimanche au samedi. La définition de cet attribut est la suivante :

(1.3.6.1.1.6.2.28 NOM 'pcimTPCDayOfWeekMask'
DESCRIPTION : cela identifie les jours valides de la semaine pour une règle de politique en utilisant une chaîne de 7 bits. Elle représente les jours de la semaine du dimanche au samedi
ÉGALITÉ : bitStringMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.6
MONO-VALEUR
)

La valeur de cet attribut DEVRAIT être vérifiée par rapport à son format défini. Si la valeur de cet attribut n'est pas conforme à cette syntaxe, cela DEVRAIT alors être considéré comme une erreur et la règle de politique DEVRAIT être traitée comme désactivée.

L'attribut pcimTPCTimeOfDayMask définit la gamme de temps dans laquelle une règle de politique est valide. Si la seconde fois est plus tôt que la première, l'intervalle s'étend alors après minuit. Le format de la chaîne est Thhmmss/Thhmmss. La définition de cet attribut est la suivante :

(1.3.6.1.1.6.2.29 NOM 'pcimTPCTimeOfDayMask'
DESCRIPTION : cela identifie la gamme valide des temps d'une politique en utilisant le format Thhmmss/Thhmmss
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.44
MONO-VALEUR
)

La valeur de cet attribut DEVRAIT être vérifiée par rapport à son format défini. Si la valeur de cet attribut n'est pas conforme à cette syntaxe, cela DEVRAIT alors être considéré comme une erreur et la règle de politique DEVRAIT être traitée comme désactivée.

Finalement, l'attribut pcimTPCLocalOrUtcTime est utilisé pour choisir entre une représentation en temps local ou UTC. Cela est transposé en une simple syntaxe d'entier, la valeur de 1 représentant l'heure locale et la valeur 2 représentant l'UTC. La définition de cet attribut est la suivante :

(1.3.6.1.1.6.2.30 NOM 'pcimTPCLocalOrUtcTime'
DESCRIPTION : ceci définit si les heures dans cette instance représente l'heure locale (valeur=1) ou l'UTC (valeur=2)
ÉGALITÉ : integerMatch
ORDRE : integerOrderingMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.27
MONO-VALEUR
)

Note : si la valeur de pcimTPCLocalOrUtcTime n'est pas 1 ou 2, cela DEVRAIT alors être considéré comme une erreur et la règle de politique DEVRAIT être désactivée. Si l'attribut n'est pas présent du tout, toutes les heures sont alors interprétées comme si elles avaient la valeur 2, c'est-à-dire l'UTC.

5.9 Classe auxiliaire pcimConditionVendorAuxClass

Cette classe fournit un mécanisme d'extension général pour représenter les conditions de politique qui n'ont pas été modélisées avec des propriétés spécifiques. À la place, ses deux propriétés sont utilisées pour définir le contenu et le format de la condition, comme on l'explique plus loin. Cette classe est destinée à des extensions spécifiques d'un fabricant qui ne sont pas éligibles pour être utilisées dans pcimCondition ; les extensions normalisées NE DEVRAIENT PAS utiliser cette classe. La définition de classe est la suivante :

(1.3.6.1.1.6.1.13 NOM 'pcimConditionVendorAuxClass'
DESCRIPTION : classe qui définit un moyen enregistré de décrire une condition de politique

SUPÉRIEURE : pcimConditionAuxClass
AUXILIAIRE
PEUT (pcimVendorConstraintData $ pcimVendorConstraintEncoding)
)

L'attribut pcimVendorConstraintData est un attribut multi valeurs. Il fournit un mécanisme général pour représenter des conditions de politique qui n'ont pas été modélisées comme attributs spécifique. Ces informations sont codées dans un ensemble de chaînes d'octets. Le format des chaînes d'octets est identifié par l'OID mémorisé dans l'attribut pcimVendorConstraintEncoding. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.31 NOM 'pcimVendorConstraintData'
DESCRIPTION : mécanisme pour représenter les contraintes qui n'ont pas été modélisées comme attributs spécifiques. Leur format est identifié par l'OID mémorisé dans l'attribut pcimVendorConstraintEncoding
ÉGALITÉ : octetStringMatch
ORDRE : octetStringOrderingMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.40
)

L'attribut pcimVendorConstraintEncoding est utilisé pour identifier le format et la sémantique pour l'attribut pcimVendorConstraintData. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.32 NOM 'pcimVendorConstraintEncoding'
DESCRIPTION : OID qui identifie le format et la sémantique de pcimVendorConstraintData pour cette instance
ÉGALITÉ : objectIdentifierMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.38
MONO-VALEUR
)

5.10 Classe auxiliaire pcimActionAuxClass

L'objet d'une action de politique est d'exécuter une ou plusieurs opérations qui vont affecter le trafic et/ou les systèmes, les appareils, etc., du réseau afin de réaliser un état de politique désiré. Cette classe est utilisée pour représenter une action à effectuer par suite d'une règle de politique dont la clause de condition a été satisfaite.

Les sous classes de cette classe auxiliaire peuvent être rattachées aux instances de trois autres classes dans le PCLS. Lorsque une sous classe de cette classe est rattachée à une instance de pcimRuleActionAssociation, ou à une instance de pcimRule, elle représente une action de politique spécifique d'une règle. Lorsque une sous classe de cette classe est rattachée à une instance de pcimPolicyInstance, elle représente une action de politique réutilisable.

Comme toutes les classes auxquelles des sous classes de cette classe auxiliaire peuvent être rattachées sont dérivées de la classe pcimPolicy, les attributs de la classe pcimPolicy vont déjà être définis pour les entrées auxquelles ces sous classes se rattachent. Donc, cette classe est dérivée directement du "sommet". La définition de classe est la suivante :

(1.3.6.1.1.6.1.14 NOM 'pcimActionAuxClass'
DESCRIPTION : classe qui représente une action à effectuer par suite d'une règle de politique
SUPÉRIEURE : sommet
AUXILIAIRE
)

5.11 Classe auxiliaire pcimActionVendorAuxClass

L'objet de cette classe est de fournir un mécanisme d'extension général pour représenter les actions de politique qui n'ont pas été modélisée avec des propriétés spécifique. À la place, ses deux propriétés sont utilisées pour définir le contenu et le format de l'action, comme on l'explique ci-dessous.

Comme son nom le suggères, cette classe est destinée aux extensions spécifiques d'un fabricant qui ne sont pas éligibles à l'utilisation de la classe standard pcimAction. Les extensions normalisées NE DEVRAIENT PAS utiliser cette classe. La définition de classe est la suivante :

(1.3.6.1.1.6.1.15 NOM 'pcimActionVendorAuxClass'
DESCRIPTION : classe qui définit un moyen enregistré de décrire une action de politique
SUPÉRIEURE : pcimActionAuxClass

AUXILIAIRE
PEUT (pcimVendorActionData $ pcimVendorActionEncoding)
)

L'attribut pcimVendorActionData est un attribut multi valeurs. Il fournit un mécanisme général pour représenter les actions de politique qui n'ont pas été modélisée comme attributs spécifiques. Ces informations sont codée dans un ensemble de chaînes d'octets. Le format des chaînes d'octets est identifié par l'OID mémorisé dans l'attribut pcimVendorActionEncoding. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.33 NOM 'pcimVendorActionData'
DESCRIPTION : mécanisme pour représenter des actions de politique qui n'ont pas été modélisées comme attributs spécifiques. Leur format est identifié par l'OID mémorisé dans l'attribut pcimVendorActionEncoding
ÉGALITÉ : octetStringMatch
ORDRE : octetStringOrderingMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.40
)

L'attribut pcimVendorActionEncoding est utilisé pour identifier le format et la sémantique de l'attribut pcimVendorActionData. Cet attribut est défini comme suit :

(1.3.6.1.1.6.2.34 NOM 'pcimVendorActionEncoding'
DESCRIPTION : OID qui identifie le format et la sémantique de l'attribut pcimVendorActionData pour cette instance
ÉGALITÉ : objectIdentifierMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.38
MONO-VALEUR
)

5.12 Classe pcimPolicyInstance

Cette classe n'est pas définie dans PCIM. Son rôle est de servir de classe structurelle à laquelle les classes auxiliaires représentant des informations de politique sont rattachées lorsque les informations sont réutilisables. Pour les classes auxiliaires qui représentent des conditions de politique et des actions de politique, il y a des classes structurelles de remplacement qui peuvent être utilisées. Voir au paragraphe 4.4 l'exposé complet sur les conditions et actions de politique réutilisables, et le rôle que cette classe joue dans la façon de les représenter. La définition de classe est la suivante :

(1.3.6.1.1.6.1.16 NOM 'pcimPolicyInstance'
DESCRIPTION : classe structurelle à laquelle les classes contenant des informations de politique réutilisables peuvent être rattachées
SUPÉRIEURE : pcimPolicy
PEUT (pcimPolicyInstanceName)
)

L'attribut pcimPolicyInstanceName est utilisé pour définir un nom facile à retenir de cette classe, et peut être utilisé comme attribut de dénomination si désiré. Il est défini comme suit :

(1.3.6.1.1.6.2.35 NOM 'pcimPolicyInstanceName'
DESCRIPTION : nom facile à retenir de cette instance de politique
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.15
MONO-VALEUR
)

Une règle de contenu de DIT pourrait être écrite pour permettre à une instance de pcimPolicyInstance d'avoir rattachée à elle des instances d'une ou plusieurs des classes d'objet auxiliaires pcimConditionAuxClass et pcimActionAuxClass. Comme ces sémantiques n'incluent pas de spécifier de propriété, la règle de contenu n'aurait pas besoin de spécifier d'attribut. Noter que d'autres règles de contenu pourraient être définies pour permettre que d'autres classes auxiliaires en rapport avec la politique soient rattachées à pcimPolicyInstance.

De même, trois règles de structure de DIT séparées pourraient être écrites. Chacune de ces règles de structure de DIT se réfèrerait à une forme de nom spécifique qui définit deux sémantiques importantes. D'abord, chaque forme de nom identifierait

un des trois attributs de dénomination possibles (c'est-à-dire, pcimPolicyInstanceName, cn, et orderedCIMKeys) pour cette classe d'objet. Ensuite, chaque forme de nom exigerait qu'une instance de la classe pcimPolicyInstance ait comme supérieur une instance de la classe pcimRepository. Cette règle de structure DEVRAIT aussi inclure une superiorStructureRule (voir la note 2 au début de la section 5).

5.13 Classe auxiliaire pcimElementAuxClass

Cette classe n'introduit pas d'attribut supplémentaire, au delà de ceux définis dans la classe pcimPolicy de laquelle elle est dérivée. Son rôle est "d'étiqueter" une instance d'une classe définie en dehors du domaine des informations de politique comme représenté par PCIM comme étant néanmoins pertinente pour une spécification de politique. Cet étiquetage peut éventuellement prendre place à deux niveaux :

- Chaque instance à laquelle pcimElementAuxClass est rattachée devient une instance de la classe pcimPolicy, car pcimElementAuxClass est une sous classe de pcimPolicy. Une recherche sur classe d'objet ="pcimPolicy" va retourner l'instance. (Comme noté plus haut, cette approche NE fonctionne PAS pour certaines mises en œuvre de répertoire. Pour s'accommoder de ces mises en œuvre, les entrées en rapport avec la politique DEVRAIT être étiquetées avec le mot clé pcim "POLICY".)
- avec l'attribut pcimKeywords qu'elle hérite de pcimPolicy, une instance à laquelle pcimElementAuxClass est rattachée peut être étiquetée comme pertinente pour un certain type ou catégorie d'informations de politique, en utilisant des mots clés standard, des mots clés définis par l'administrateur, ou les deux.

La définition de classe est la suivante :

(1.3.6.1.1.6.1.17 NOM 'pcimElementAuxClass'
DESCRIPTION : classe auxiliaire utilisée pour étiqueter des instances de classes définies en dehors du domaine de la politique
comme pertinentes pour une certaine spécification de politique
SUPÉRIEURE : pcimPolicy
AUXILIAIRE
)

5.14 Les trois classes de répertoire de politique

Ces classes fournissent un conteneur pour des informations de politique réutilisables, comme des conditions de politique réutilisables et/ou des actions de politique réutilisables. Le présent document est concernée juste par la transposition des propriétés qui apparaissent dans ces classes. Conceptuellement, cela peut être vu comme une localisation spéciale dans la DIT où des informations de politique peuvent résider. Comme pcimRepository est dérivé de la classe dlm1AdminDomain définie dans la référence [DTMF], la présente spécification a une dépendance normative à cet élément de référence [DTMF] (ainsi qu'à la hiérarchie de dérivation entière,qui apparaît aussi dans la référence [DTMF]). Pour maximiser la souplesse, la classe pcimRepository est définie comme abstraite. Une sous classe pcimRepositoryAuxClass fournit un rattachement auxiliaire à une autre entrée, alors qu'une sous classe structurelle pcimRepositoryInstance est disponible pour représenter un répertoire de politiques comme une entrée autonome. La définition de la classe pcimRepository est la suivante :

(1.3.6.1.1.6.1.18 NOM 'pcimRepository'
DESCRIPTION : conteneur pour des informations de politique réutilisables
SUPÉRIEURE : dlm1AdminDomain
ABSTRAITE
PEUT (pcimRepositoryName)
)

L'attribut pcimRepositoryName est utilisé pour définir un nom facile à retenir de cette classe, et peut être utilisé comme attribut de dénomination si désiré. Il est défini comme suit :

(1.3.6.1.1.6.2.36 NOM 'pcimRepositoryName'
DESCRIPTION : nom facile à retenir de ce répertoire de politiques
ÉGALITÉ : caseIgnoreMatch
ORDRE : caseIgnoreOrderingMatch
SOUS CHAINE : caseIgnoreSubstringsMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.15
MONO-VALEUR
)

Les deux sous classes de pcimRepository sont définies comme suit. D'abord, pcimRepositoryAuxClass est une classe auxiliaire qui peut être utilisée pour agréger des informations de politique réutilisables. Elle est définie comme suit :

(1.3.6.1.1.6.1.19 NOM 'pcimRepositoryAuxClass'
DESCRIPTION : classe auxiliaire qui peut être utilisé pour agréger des informations de politique réutilisables
SUPÉRIEURE : pcimRepository
AUXILIAIRE
)

Dans les cas où des classes structurelles sont nécessaires plutôt qu'une classe auxiliaire, la classe pcimRepositoryInstance est une classe structurelle qui peut être utilisée pour agréger des informations de politique réutilisables. Elle est définie comme suit :

(1.3.6.1.1.6.1.20 NOM 'pcimRepositoryInstance'
DESCRIPTION : classe structurelle qui peut être utilisée pour agréger des informations de politique réutilisables
SUPÉRIEURE : pcimRepository
STRUCTURELLE
)

Trois règles de structure de DIT séparées pourraient être écrites pour cette classe. Chacune de ces règles de structure de DIT se réfèrerait à une forme de nom spécifique qui permettrait à une instance de la classe pcimRepository d'être nommée sous tout supérieur en utilisant un des trois attributs de dénomination possibles (c'est-à-dire, pcimRepositoryName, cn, et orderedCIMKeys). Cette règle de structure DEVRAIT aussi inclure une superiorStructureRule (voir la note 2 au début de la section 5).

5.15 Classe auxiliaire pcimSubtreesPtrAuxClass

Cette classe auxiliaire fournit un seul attribut multi valeurs qui fait référence à un ensemble d'objets qui sont à la racine des sous arborescences de la DIT qui contiennent des informations en rapport avec la politique. En rattachant cet attribut aux instances de diverses autres classes, un administrateur de politique a une façon souple de fournir un point d'entrée dans le répertoire qui permet à un client de localiser et restituer les informations de politique pertinentes pour lui.

Il est prévu que ces entrées soient placées dans la DIT de telle sorte que des DN bien connus puissent être utilisés pour référencer une entrée structurelle bien connue qui a la pcimSubtreesPtrAuxClass qui lui est rattachée. En effet, cela définit un ensemble de points d'entrée. Chacun de ces points d'entrée peut contenir et/ou faire référence, à toutes les entrées en relation avec la politique pour tous les domaines de politique bien connus. La classe pcimSubtreesPtrAuxClass fonctionne comme une étiquette pour identifier des portions de la DIT qui contiennent des informations de politique.

Cet objet ne fournit pas les liens sémantiques entre les objets de politique individuels, comme ceux entre un groupe de politique et les règles de politique qui lui appartiennent. Son seul rôle est de permettre une restitution brute efficace des objets en rapport avec la politique, comme décrit au paragraphe 4.5.

Une fois les objets restitués, un client de répertoire peut déterminer les liens sémantiques en suivant les références contenues dans les attributs multi valeurs, comme pcimRulesAuxContainedSet.

Comme les objets en rapport avec la politique vont souvent être inclus dans la sous arborescence de la DIT derrière un objet auquel la classe auxiliaire est rattachée, un client DEVRAIT demander les objets en rapport avec la politique provenant de la sous arborescence sous l'objet avec ces références en même temps qu'il demande les références elles-mêmes.

Comme les clients sont supposés se conduire de cette façon, l'administrateur de politique DEVRAIT s'assurer que cette sous arborescence ne contient pas trop d'objets sans relation avec la politique pour qu'une recherche initiale faite de cette façon ne provoque pas des problèmes de performances. La classe pcimSubtreesPtrAuxClass NE DEVRAIT PAS être rattachée à la partition racine pour une grande partition du répertoire contenant un nombre relativement faible d'objets en rapport avec la politique avec un grand nombre d'objets sans relation avec la politique (là encore, "politique" se réfère ici à PCIM, et non à la définition et l'utilisation de "politique" de X.501). Une meilleure approche serait d'introduire un objet conteneur immédiatement en dessous de la partition racine, de rattacher pcimSubtreesPtrAuxClass à cet objet conteneur, et de placer ensuite tous les objets en rapport avec la politique dans cette sous arborescence. La définition de classe est la suivante :

(1.3.6.1.1.6.1.21 NOM 'pcimSubtreesPtrAuxClass'
DESCRIPTION : classe auxiliaire fournissant des références de DN aux racines des sous arborescences de la DIT qui contiennent des objets en rapport avec la politique
SUPÉRIEURE : sommet
AUXILIAIRE
PEUT (pcimSubtreesAuxContainedSet)

)

L'attribut pcimSubtreesAuxContainedSet fournit un ensemble non ordonné de références de DN aux instances d'un ou plusieurs objets sous lesquels des informations en relation avec la politique sont présentes. Les objets référencés peuvent eux-mêmes contenir ou non des informations en relation avec la politique. La définition de l'attribut est la suivante :

(1.3.6.1.1.6.2.37 NOM 'pcimSubtreesAuxContainedSet'
DESCRIPTION : DN des objets qui servent de racines aux sous arborescences de DIT qui contiennent des objets en rapport avec la politique
ÉGALITÉ : distinguishedNameMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.12
)

Note : l'attribut cn N'A PAS besoin d'être défini pour cette classe, parce qu'une classe auxiliaire est utilisée comme moyen pour collecter les attributs communs et les traiter comme des propriétés d'un objet. Une bonne analogie est un fichier #include, sauf que comme une classe auxiliaire est une classe, tous les avantages d'une classe (par exemple, l'héritage) peuvent être appliqués à une classe auxiliaire.

5.16 Classe auxiliaire pcimGroupContainmentAuxClass

Cette classe auxiliaire fournit un seul attribut multi valeurs qui fait référence à un ensemble de pcimGroups. En rattachant cet attribut aux instances de diverses autres classes, un administrateur de politique a un moyen souple pour fournir un point d'entrée dans le répertoire qui permet à un client de localiser et restituer les pcimGroups pertinents pour lui.

Comme c'est le cas avec pcimRules, un administrateur de politique peut avoir plusieurs références différentes à un pcimGroup dans la structure globale du répertoire. La classe pcimGroupContainmentAuxClass est le mécanisme qui rend possible à l'administrateur de politique de définir toutes ces différentes références. La définition de classe est la suivante :

(1.3.6.1.1.6.1.22 NOM 'pcimGroupContainmentAuxClass'
DESCRIPTION : classe auxiliaire utilisée pour lier des pcimGroups à un objet conteneur approprié
SUPÉRIEURE : sommet
AUXILIAIRE
PEUT (pcimGroupsAuxContainedSet)
)

L'attribut pcimGroupsAuxContainedSet fournit un ensemble non ordonné de références aux instances d'un ou plusieurs pcimGroups associés à l'instance d'une classe structurelle à laquelle cet attribut a été ajouté. La définition de l'attribut est la suivante :

(1.3.6.1.1.6.2.38 NOM 'pcimGroupsAuxContainedSet'
DESCRIPTION : DN des pcimGroups associés d'une certaine façon à l'instance à laquelle cet attribut a été ajouté
ÉGALITÉ : distinguishedNameMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.12
)

Note : l'attribut cn N'A PAS à être défini pour cette classe pour les mêmes raisons que celles données pour la classe pcimSubtreesPtrAuxClass au paragraphe 5.15.

5.17 Classe auxiliaire pcimRuleContainmentAuxClass

Cette classe auxiliaire fournit un seul attribut multi valeurs qui fait référence à un ensemble de pcimRules. En rattachant cet attribut aux instances de diverses autres classes, un administrateur de politique a un moyen souple de fournir un point d'entrée dans le répertoire ce qui permet à un client de localiser et restituer les pcimRules pertinentes pour lui.

Un administrateur de politique peut avoir plusieurs références différentes pour une pcimRule dans la structure globale du répertoire. Par exemple, il peut y avoir des références à toutes les pcimRules pour le trafic originaire d'un sous réseau particulier à partir d'une entrée de répertoire qui représente ce sous réseau. En même temps, il peut y avoir des références à toutes les pcimRules relatives à un réglage DiffServ particulier à partir d'une instance d'un pcimGroup explicitement introduit comme conteneur pour des pcimRules en rapport avec DiffServ. La classe pcimRuleContainmentAuxClass est le mécanisme qui rend possible à l'administrateur de politique de définir toutes ces références séparées. La définition de classe est la suivante :

(1.3.6.1.1.6.1.23 NOM 'pcimRuleContainmentAuxClass'
DESCRIPTION : classe auxiliaire utilisée pour lier des pcimRules à un objet conteneur approprié
SUPÉRIEURE : sommet
AUXILIAIRE
PEUT (pcimRulesAuxContainedSet)
)

L'attribut pcimRulesAuxContainedSet fournit un ensemble non ordonné de références à une ou plusieurs instances des pcimRules associées à l'instance d'une classe structurelle à laquelle cet attribut a été ajouté. La définition de l'attribut est la suivante :

(1.3.6.1.1.6.2.39 NOM 'pcimRulesAuxContainedSet'
DESCRIPTION : les DN de pcimRules associés d'une certaine façon à l'instance à laquelle cet attribut a été ajouté
ÉGALITÉ : distinguishedNameMatch
SYNTAXE : 1.3.6.1.4.1.1466.115.121.1.12
)

L'attribut cn N'A PAS à être défini pour cette classe pour les mêmes raisons que celles données pour la classe pcimSubtreesPtrAuxClass au paragraphe 5.15.

6. Extension des classes définies

Les paragraphes qui suivent donnent des lignes directrices sur la façon de créer un schéma spécifique d'un domaine dérivé du présent document, ; ils exposent comment devraient être utilisées les classes de fabricant dans le PCLS, et expliquent comment les policyTimePeriodConditions se rapportent aux autres conditions de politique.

6.1 Création des sous classes pcimConditionAuxClass et pcimActionAuxClass

Au paragraphe 4.4, on expose comment, en représentant les conditions de politique et les actions de politique comme des classes auxiliaires dans un schéma, on conserve de la souplesse pour instancier une condition ou action particulière soit comme spécifique d'une règle, soit comme réutilisable. Cette souplesse est perdue si une classe de condition ou d'action est définie comme structurelle plutôt que comme auxiliaire. Pour les schémas normalisés, le présent document spécifie que les informations spécifiques d'un domaine DOIVENT être exprimées dans des sous classes auxiliaires de pcimConditionAuxClass et pcimActionAuxClass. Il est RECOMMANDÉ que les schémas non normalisés suivent aussi cette pratique.

6.2 Utilisation des attributs de politique de fabricant

Comme exposé au paragraphe 5.9, les attributs pcimVendorConstraintData et pcimVendorConstraintEncoding sont inclus dans pcimConditionVendorAuxClass pour fournir un mécanisme de représentation des conditions de politique spécifique de fabricant qui ne sont pas éligibles à être représentées par la classe pcimCondition (ou ses sous classes). Les attributs pcimVendorActionData et pcimVendorActionEncoding dans la classe pcimActionVendorAuxClass jouent le même rôle par rapport aux actions. Cela permet l'interopérabilité entre différents fabricants qui ne pourraient pas interopérer autrement.

Par exemple, imaginons un réseau composé d'appareils d'accès provenant du fabricant A, des appareils de bordure et du cœur provenant du fabricant B, et un serveur de politique du fabricant C. Il est souhaitable que ce serveur de politique soit capable de configurer et gérer tous les appareils des fabricants A et B. Malheureusement ces appareils vont en général avoir peu en commun (par exemple, des mécanismes différents, des façons différentes de contrôler ces mécanismes, des systèmes d'exploitation différents, des commandes différentes, et ainsi de suite). Les conditions d'extension donnent un moyen aux commandes spécifiques d'un fabricant d'être codées comme des chaînes d'octets, de sorte qu'un seul serveur de politique peut couramment gérer des appareils de différents fabricants.

6.3 Utilisation des périodes de validité

Les périodes de validité sont définies comme une sous classe auxiliaire de pcimConditionAuxClass, appelée pcimTPCAuxClass. C'est pour permettre leur inclusion dans les définitions de condition ET/OU pour une pcimRule. Il faut veiller à ne pas diviser en sous classes pcimTPCAuxClass pour ajouter des propriétés de condition spécifique du domaine.

Par exemple, il serait incorrect d'ajouter des propriétés de condition spécifiques de IPsec ou de QS à la classe pcimTPCAuxClass, juste parce que IPsec ou la QS incluent du temps dans leur définition de condition. Le sous classement correct serait de créer des sous classes de pcimConditionAuxClass spécifiques de IPsec ou de QS et ensuite de combiner les

instances de ces classes de condition spécifiques de domaine avec le critère approprié de période de validité. On fait cela en utilisant les capacités d'association ET/OU pour les conditions de politique dans pcimRules.

7. Considérations pour la sécurité

Le PCLS, présenté dans le présent document, fournit une transposition du modèle en mode objet pour décrire les informations de politique (PCIM) dans un modèle de données qui forme le cadre de base de la description des structures des données de politique, au cas où le répertoire de politiques prend la forme d'un répertoire accessible par LDAP.

PCLS n'est pas destiné à représenter une conception ou une mise en œuvre de système particulière. PCLS n'est pas directement utilisable dans un système réel, sans les transpositions spécifiques de discipline qui sont en cours de préparation dans le groupe de travail Cadre de politique de l'IETF.

Ces autres documents dérivés, qui utilisent PCIM et ses extensions spécifiques de discipline comme base, auront besoin de porter des considérations plus spécifiques sur la sécurité (se reporter à la RFC 3060 pour plus d'informations).

La raison pour laquelle PCLS, tel que défini ici, n'est représentatif d'aucun système réel, est que ses classes d'objet ont été conçues comme indépendantes de toute discipline, ou domaine de politique, spécifique. Par exemple, DiffServ et IPsec représentent deux domaines de politique différents. Chaque document qui étend PCIM à un ce ces domaines va déduire des sous classes des classes et relations définies dans la PCIM, afin de représenter les extensions au modèle générique pour couvrir les domaines techniques spécifiques.

Les documents dérivés de PCIM vont donc diviser en sous classes les classes PCIM selon des classes spécifiques de chaque domaine technique de politique (QS, IPsec, etc.) qui vont à leur tout être transposées en schémas spécifiques de répertoires cohérents avec le PCLS documenté ici.

Bien que les exigences de sécurité spécifiques d'une discipline ne soient pas appropriées pour PCLS, des exigences de sécurité spécifiques DOIVENT être définie pour chaque application fonctionnant dans la réalité de PCIM. Tout comme il y aura une large gamme de systèmes de fonctionnement qui utilisent PCIM dans la réalité, il y aura aussi une large gamme d'exigences de sécurité pour ces systèmes. Certains systèmes de fonctionnement du monde réel qui sont déployés en utilisant PCLS peuvent avoir des exigences de sécurité extensives qui impactent presque toutes les classes d'objet utilisée par de tels systèmes, alors que les exigences de sécurité d'autres systèmes peuvent n'avoir que très peu d'impact.

Les documents dérivés, discutés ci-dessus, vont créer le contexte d'application des exigences de sécurité opérationnelles au niveau du système réel par rapport aux divers modèles qui dérivent de PCIM, en cohérence avec PCLS.

Dans certains scénarios de la réalité, les valeurs associées à certaines propriétés, au sein de certaines instances de classes d'objet, peuvent représenter des informations associées à des ressources rares, et/ou coûteuses (et donc précieuses). Il peut se trouver que ces valeurs ne doivent pas être divulguées, ou manipulées, par des parties non autorisées.

Comme le présent document forme la base de la représentation d'un modèle de données de politique dans un format spécifique (un répertoire accessible par LDAP) il est ici approprié de faire référence aux outils et mécanismes spécifiques du modèle de données qui sont disponibles pour réaliser l'authentification et l'autorisation implicites dans une exigence qui interdit l'accès en lecture et/ou lecture-écriture à ces valeurs mémorisées dans un répertoire.

Les considérations générales sur la sécurité pour LDAP s'appliquent, comme décrit dans la [RFC3377]. On trouve les outils et mécanismes d'authentification et d'autorisation spécifiques de LDAP dans les documents en cours de normalisation suivants, qui sont appropriés à l'application à la gestion de la sécurité appliquée aux modèles de données de politique mémorisées dans un répertoire accessible par LDAP :

- RFC 2829 (Méthodes d'authentification pour LDAP)
- RFC 2830 (Protocole léger d'accès à un répertoire (v3) : extension pour la sécurité de la couche transport)

Toute exigence de sécurité identifiée qui n'est pas traitée dans les documents appropriés de modèle d'information spécifique de la discipline, ou dans le présent document, DOIT être traitée dans les documents dérivés de modèle de données qui sont spécifiques de chaque discipline.

8. Considérations relatives à l'IANA

Se référer à la [RFC3383], "Autorité d'allocation des numéros de l'Internet (IANA) : Considérations sur le protocole léger d'accès à un répertoire (LDAP)".

8.1 Identifiants d'objet

L'IANA a enregistré un identifiant d'objet LDAP à utiliser dans la présente spécification technique conformément au gabarit suivant :

Sujet : Demande d'enregistrement d'OID pour LDAP
Adresse personnelle & de messagerie à contacter pour plus d'informations : Bob Moore (remoore@us.ibm.com)
Spécification : RFC 3703
Auteur/Contrôleur des changements : IESG
Commentaires : Les OID alloués seront utilisés comme base pour identifier un certain nombre d'éléments de schéma définis dans le présent document.

L'IANA a alloué un OID de 1.3.6.1.1.6 avec le nom de pcimSchema à cet enregistrement comme enregistré dans le registre : http://www.iana.org/assignments/smi-numbers

8.2 Descripteurs d'identifiant d'objet

L'IANA a enregistré les descripteurs LDAP utilisés dans la présente spécification technique comme détaillé dans le gabarit suivant :

Sujet : Demande de mise à jour d'enregistrement de descripteur LDAP
Descripteur (nom abrégé) : voir le commentaire
Identifiant d'objet : voir le commentaire
Adresse personnelle & de messagerie à contacter pour plus d'informations : Bob Moore (remoore@us.ibm.com)
Usage : voir le commentaire
Spécification : RFC 3703
Auteur/Contrôleur des changements : IESG
Commentaires : Les descripteurs suivants ont été ajoutés :

Nom	Type	ID
pcimPolicy	O	1.3.6.1.1.6.1.1
pcimGroup	O	1.3.6.1.1.6.1.2
pcimGroupAuxClass	O	1.3.6.1.1.6.1.3
pcimGroupInstance	O	1.3.6.1.1.6.1.4
pcimRule	O	1.3.6.1.1.6.1.5
pcimRuleAuxClass	O	1.3.6.1.1.6.1.6
pcimRuleInstance	O	1.3.6.1.1.6.1.7
pcimRuleConditionAssociation	O	1.3.6.1.1.6.1.8
pcimRuleValidityAssociation	O	1.3.6.1.1.6.1.9
pcimRuleActionAssociation	O	1.3.6.1.1.6.1.10
pcimConditionAuxClass	O	1.3.6.1.1.6.1.11
pcimTPCAuxClass	O	1.3.6.1.1.6.1.12
pcimConditionVendorAuxClass	O	1.3.6.1.1.6.1.13
pcimActionAuxClass	O	1.3.6.1.1.6.1.14
pcimActionVendorAuxClass	O	1.3.6.1.1.6.1.15
pcimPolicyInstance	O	1.3.6.1.1.6.1.16
pcimElementAuxClass	O	1.3.6.1.1.6.1.17
pcimRepository	O	1.3.6.1.1.6.1.18
pcimRepositoryAuxClass	O	1.3.6.1.1.6.1.19
pcimRepositoryInstance	O	1.3.6.1.1.6.1.20
pcimSubtreesPtrAuxClass	O	1.3.6.1.1.6.1.21
pcimGroupContainmentAuxClass	O	1.3.6.1.1.6.1.22
pcimRuleContainmentAuxClass	O	1.3.6.1.1.6.1.23
pcimKeywords	A	1.3.6.1.1.6.2.3
pcimGroupName	A	1.3.6.1.1.6.2.4
pcimRuleName	A	1.3.6.1.1.6.2.5
pcimRuleEnabled	A	1.3.6.1.1.6.2.6
pcimRuleConditionListType	A	1.3.6.1.1.6.2.7
pcimRuleConditionList	A	1.3.6.1.1.6.2.8
pcimRuleActionList	A	1.3.6.1.1.6.2.9
pcimRuleValidityPeriodList	A	1.3.6.1.1.6.2.10

pcimRuleUsage	A	1.3.6.1.1.6.2.11
pcimRulePriority	A	1.3.6.1.1.6.2.12
pcimRuleMandatory	A	1.3.6.1.1.6.2.13
pcimRuleSequencedActions	A	1.3.6.1.1.6.2.14
pcimRoles	A	1.3.6.1.1.6.2.15
pcimConditionGroupNumber	A	1.3.6.1.1.6.2.16
pcimConditionNegated	A	1.3.6.1.1.6.2.17
pcimConditionName	A	1.3.6.1.1.6.2.18
pcimConditionDN	A	1.3.6.1.1.6.2.19
pcimValidityConditionName	A	1.3.6.1.1.6.2.20
pcimTimePeriodConditionDN	A	1.3.6.1.1.6.2.21
pcimActionName	A	1.3.6.1.1.6.2.22
pcimActionOrder	A	1.3.6.1.1.6.2.23
pcimActionDN	A	1.3.6.1.1.6.2.24
pcimTPCTime	A	1.3.6.1.1.6.2.25
pcimTPCMonthOfYearMask	A	1.3.6.1.1.6.2.26
pcimTPCDayOfMonthMask	A	1.3.6.1.1.6.2.27
pcimTPCDayOfWeekMask	A	1.3.6.1.1.6.2.28
pcimTPCTimeOfDayMask	A	1.3.6.1.1.6.2.29
pcimTPCLocalOrUtcTime	A	1.3.6.1.1.6.2.30
pcimVendorConstraintData	A	1.3.6.1.1.6.2.31
pcimVendorConstraintEncoding	A	1.3.6.1.1.6.2.32
pcimVendorActionData	A	1.3.6.1.1.6.2.33
pcimVendorActionEncoding	A	1.3.6.1.1.6.2.34
pcimPolicyInstanceName	A	1.3.6.1.1.6.2.35
pcimRepositoryName	A	1.3.6.1.1.6.2.36
pcimSubtreesAuxContainedSet	A	1.3.6.1.1.6.2.37
pcimGroupsAuxContainedSet	A	1.3.6.1.1.6.2.38
pcimRulesAuxContainedSet	A	1.3.6.1.1.6.2.39

où Type A est un Attribut, Type O est une ObjectClass

Ces allocations sont enregistrées dans le registre suivant : http://www.iana.org/assignments/ldap-parameters

9. Remerciements

Nous tenons à remercier Kurt Zeilenga, Roland Hedburg, et Steven Legg de leur relecture de ce document et de leurs nombreuses et utiles suggestions et corrections.

Plusieurs des classes de politique de ce modèle sont d'abord apparues dans les premiers projets de l'IETF sur la politique IPsec et la politique de qualité de service. Les auteurs de ces projets étaient Partha Bhattacharya, Rob Adams, William Dixon, Roy Pereira, Raju Rajan, Jean-Christophe Martin, Sanjay Kamat, Michael See, Rajiv Chaudhury, Dinesh Verma, George Powers, et Raj Yavatkar.

Le présent document est étroitement aligné sur les travaux effectués dans les groupes de travail Politique et réseaux du Distributed Management Task Force (DMTF). Nous tenons particulièrement à remercier Lee Rafalow, Glenn Waters, David Black, Michael Richardson, Mark Stevens, David Jones, Hugh Mahon, Yoram Snir, et Yoram Ramberg de leurs utiles commentaires.

10. Appendice : Construction de la valeur de orderedCIMKeys

Cet appendice est non normatif, et est inclus dans le document comme un guide pour les mises en œuvre qui souhaitent échanger des informations entre les schémas CIM et les schémas LDAP.

Au sein d'un espace de noms CIM, la dénomination est essentiellement plate ; toutes les instances sont identifiées par les valeurs de leurs propriétés de clés et chaque combinaison des valeurs de clé doit être unique. Une forme limitée de dénomination hiérarchique est cependant disponible dans CIM, en utilisant les associations faibles : comme une association faible implique la propagation des propriétés de clé et leurs valeurs de l'objet supérieur à l'objet subordonné, le subordonné peut être vu comme étant désigné "sous" l'objet supérieur. Cependant, une fois qu'elles ont été propagées, les propriétés de clé propagées et leurs valeurs fonctionnent exactement de la même façon que le font les propriété de clé natives et leurs valeurs

pour identifier une instance de CIM.

Le document de transposition de CIM [DTMF] introduit un attribut spécial, orderedCIMKeys, pour aider à transposer de la classe CIM_ManagedElement en la classe LDAP dlm1ManagedElement. Cet attribut DEVRAIT n'être utilisé que dans un environnement où il est nécessaire de transposer entre un répertoire accessible par LDAP et un dépôt CIM. Pour un environnement LDAP, d'autres attributs de désignation LDAP sont définis (c'est-à-dire, cn et un attribut de dénomination spécifique de la classe) qui DEVRAIENT être utilisés à la place.

Le rôle de orderedCIMKeys est de représenter les informations nécessaires pour corréler une entrée dans un répertoire accessible par LDAP avec une instance dans un espace de noms CIM. Selon la façon dont la désignation des entrées relatives à CIM est traitée dans un répertoire LDAP, la valeur de orderedCIMKeys représente une des deux choses suivantes :
- Si la hiérarchie de DIT ne reflète pas la "hiérarchie de faiblesse" de l'espace de noms de CIM, alors orderedCIMKeys représente toutes les clés de l'instance CIM, natives et propagées.
- Si la hiérarchie de DIT reflète bien la "hiérarchie de faiblesse" de l'espace de noms de CIM, orderedCIMKeys peut alors représenter soit toutes les clés de l'instance, soit seulement les clés natives.

Sans considération du terme de l'alternative retenu, la syntaxe de orderedCIMKeys est la même - une DirectoryString de la forme :

<className>.<clé>=<valeur>[,<clé>=<valeur>]*

où les éléments <clé>=<valeur> sont ordonnés par les noms des propriétés des clés, conformément à la séquence de collationnement de l'US ASCII. Les seules espaces permises dans la DirectoryString sont celles qui entrent dans un élément <valeur>. Comme avec l'alphabétisation des propriétés de clés, le but de la suppression des espaces est là encore de rendre prévisible le résultat des opérations de chaînes.

Les valeurs des éléments <valeur> sont dérivées des diverses syntaxes CIM conformément à une grammaire spécifiée dans [CIM].

11. Références

11.1 Références normatives

[CIM] Distributed Management Task Force, Inc., "Common Information Model (CIM) Specification", Version 2.2, 14 juin 1999. Disponible à : http://www.dmtf.org/standards/documents/CIM/DSP0004.pdf

[DTMF] Distributed Management Task Force, Inc., "DMTF LDAP Schema for the CIM v2.5 Core Information Model", 15 avril 2002. Disponible à : http://www.dmtf.org/standards/documents/DEN/DSP0123.pdf

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC2252] M. Wahl, A. Coulbeck, T. Howes, S. Kille, "Protocole léger d'accès à un répertoire (v3) : Définitions de syntaxe d'attribut", décembre 1997. *(P.S.)* (*Obsolète, voir* RFC4510, RFC4517, RFC4523, RFC4512)

[RFC2256] M. Wahl, "Résumé du schéma d'utilisateur X.500(96) à utiliser avec LDAPv3", décembre 1997. *(P.S.)* (*Obsolète, voir* RFC4510, RFC4512, RFC4517, RFC4519, RFC4523)

[RFC3060] B. Moore et autres, "Spécification du modèle d'information de cœur de politique -- version 1", février 2001. *(P.S.)* (*MàJ par* RFC3460)

[RFC3377] J. Hodges, R. Morgan, "Protocole léger d'accès à un répertoire (v3) : Spécification technique", septembre 2002. *(P.S.)* (*Obsolète, voir* RFC4510)

[RFC3698] K. Zeilenga, éd., "Protocole léger d'accès à un répertoire (LDAP) : règles de correspondance supplémentaires", février 2004. *(P.S.)* (*MàJ par* RFC4517)

[X.501] Recommandation UIT-T X.501 "L'Annuaire : Modèles", 2001.

[X.520] Recommandation UIT-T X.520 "L'Annuaire : Types d'attributs choisis", 2001.

11.2 Références pour information

[ARCHI] Strassner, J., "policy architecture BOF presentation", 42ème réunion de l'IETF, Chicago, Illinois, octobre 1998. Le compte rendu de ce BOF est disponible à : http://www.ietf.org/proceedings/98aug/index.html .

[RFC2028] R. Hovey, S. Bradner, "Les organisations impliquées dans le processus de normalisation de l'IETF", octobre 1996. (BCP0011) (*MàJ par* RFC3668, RFC3979)

[RFC2753] R. Yavatkar, D. Pendarakis, R. Guerin, "Cadre pour le contrôle d'admission fondé sur la politique", janvier 2000. *(Info.)*

[RFC2829] M. Wahl et autres, "Méthodes d'authentification pour LDAP", mai 2000. *(P.S.)* (*Obsolète, voir* RFC4513, RFC4510)

[RFC2830] J. Hodges, R. Morgan, M. Wahl, "Protocole léger d'accès à un répertoire (v3) : extension pour la sécurité de la couche transport", mai 2000. *(P.S.)* (*Obsolète, voir* RFC4511, RFC4513, RFC4510)

[RFC3383] K. Zeilenga, "Autorité d'allocation des numéros de l'Internet (IANA) : Considérations sur le protocole léger d'accès à un répertoire (LDAP)", septembre 2002. (*Obsolète, voir* RFC4520)

12. Adresse des auteurs

John Strassner
Intelliden Corporation
90 South Cascade Avenue
Colorado Springs, CO 80903
téléphone : +1.719.785.0648
mél : john.strassner@intelliden.com

Bob Moore
IBM Corporation
P. O. Box 12195, BRQA/B501/G206
3039 Cornwallis Rd.
Research Triangle Park, NC 27709-2195
téléphone : +1 919-254-4436
mél : remoore@us.ibm.com

Ryan Moats
Lemur Networks, Inc.
15621 Drexel Circle
Omaha, NE 68135
téléphone : +1-402-894-9456
mél : rmoats@lemurnetworks.net

Ed Ellesson
3026 Carriage Trail
Hillsborough, NC 27278
téléphone : +1 919-644-3977
mél : ellesson@mindspring.com

13. Déclaration complète de droits de reproduction

http://www.ietf.org/ipr.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf-ipr@ietf.org.

Remerciement
Le financement de la fonction d'édition des RFC est actuellement assuré par la Internet Society.