Groupe de travail Réseau
Request for Comments : 4745
Catégorie : Sur la voie de la normalisation

Traduction Claude Brière de L'Isle

H. Schulzrinne, Columbia University
H. Tschofenig, Siemens Networks
J. Morris, CDT
J. Cuellar, Siemens
J. Polk, Cisco
J. Rosenberg, Cisco
février 2007

Politique commune : format de document pour exprimer les préférences de confidentialité

Résumé
Le présent document définit un cadre pour les politiques d'autorisation qui contrôlent l'accès à des données spécifiques de l'application. Ce cadre combine des aspects d'autorisation communs à la localisation et à la présence. Un schéma XML spécifie le langage pour représenter les règles de politique communes. Le cadre de politique commune peut être étendu à d'autres domaines d'application.

Table des matières

1. Introduction

Le présent document définit un cadre pour créer des politiques d'autorisation pour l'accès aux données spécifiques d'application. Ce cadre est le résultat de la combinaison des aspects communs des systèmes à une seule autorisation qui contrôlent plus spécifiquement l'accès aux informations de présence et de localisation et qui avaient été précédemment développés séparément. L'avantage de la combinaison de ces deux systèmes d'autorisation est double. D'abord, il permet de construire un système qui améliore la valeur de présence avec les informations de localisation d'une façon naturelle et réutilise le même mécanisme d'autorisation sous-jacent. Ensuite, il invite à un cadre d'autorisation plus générique avec des mécanismes d'extensibilité. L'applicabilité du cadre spécifié dans le présent document n'est pas limité aux politiques qui contrôlent l'accès aux données d'information de présence et de localisation, mais peut être étendu aux autres domaines d'application.

Le cadre général défini dans le présent document est destiné à être accompagné et amélioré par des politiques spécifiques d'application spécifiées ailleurs. Le cadre de politique commune décrit ici est amélioré par des doucments de politique spécifiques du domaine, incluant la présence [RFC5025] et la localisation [RFC6772]. Cette relation est montrée par la Figure 1.

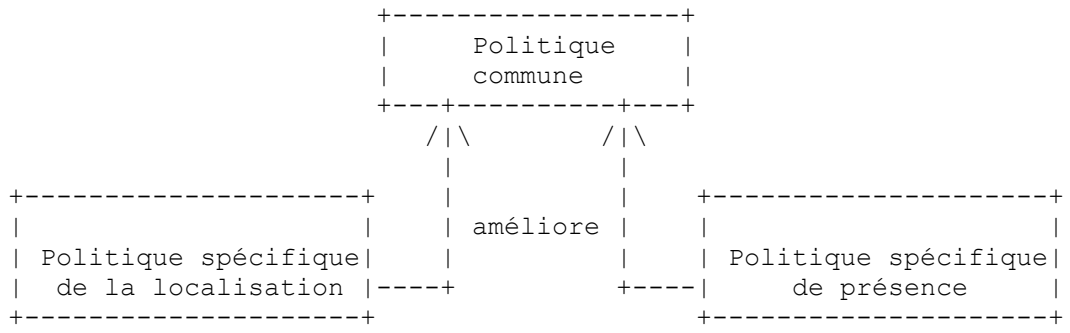

Figure 1 : améliorations de politique commune

Le présent document commence par une introduction à la terminologie à la Section 2, une illustration des modes de base de fonctionnement à la Section 3, une description des buts (Section 4) et des non buts (Section 5) du cadre de politique, suivis par les modèles de données à la Section 6. La structure d'une règle, à savoir les conditions, actions, et transformations, est décrite dans les Sections 7, 8, et 9. La procédure pour combiner les permissions est expliquée à la Section 10 et utilisée quand les conditions pour plus d'une règle sont satisfaites. Une brève description des méta politiques est donnée à la Section 11. Un exemple est fourni à la Section 12. Le schéma XML est exposé à la Section 13. Les considérations relatives à l'IANA de la Section 15 suivent les considérations sur la sécurité de la Section 14.

2. Terminologie

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Le présent document introduit les termes suivants :

PT *(Presentity/Target)* présentité/cible : la PT est l'entité sur laquelle des informations ont été demandées.

RM *(Rule Maker)* faiseur de règles : le RM est une entité qui crée les règles d'autorisation qui resteignent l'accès aux éléments de données.

PS *(Policy Server)* serveur de politique d'autorisation : cette entité a accès aux politiques d'autorisation et aux éléments de données. Dans les applications spécifiques de localisation, l'entité PS est marquée comme serveur de localisation (LS).

WR *(Watcher/Recipient)* observateur/receveur : cette entité demande l'accès aux éléments de données de la PT. Une opération d'accès pourrait être une opération de lecture, d'écriture, ou toute autre opération.

Une politique est donnée par un "ensemble de règles" qui contient une liste non ordonnée de "règles". Une "règle" a une partie "conditions", "actions", et "transformations".

Le terme "permission" indique les composants action et transformation d'une "règle".

Le terme "protocole utilisateur" est défini dans la [RFC3693]. Il se réfère au protocole utilisé pour demander et retourner l'accès aux éléments de données sensibles à la confidentialité.

3. Modes de fonctionnement

La séquence abstraite des opérations peut en gros être décrite comme suit. Le PS reçoit une interrogation sur des éléments de données pour une PT particulière, via le protocole utilisateur. Le protocole utilisateur (ou plus précisément, le protocole d'authentification) fournit l'identité du demandeur, soit au moment de l'interrogation, soit au moment de l'abonnement. L'identité authentifiée du WR, ensemble avec les autres informations fournies par le protocole utilisateur ou généralement disponibles au serveur, est alors utilisée pour chercher dans l'ensemble de règles. Toutes les règles correspondantes sont combinées en accord avec un algorithme de combinaison de permissions décrit à la Section 10. Les règles combinées sont appliquées aux données d'application, résultant en l'application d'une confidentialité fondée sur les politiques de transformation. Les données d'application résultantes sont retournées au WR.

Trois modes de fonctionnement différents peuvent être distingués :

3.1 Demande-réponse passive - PS comme serveur (répondant)

Dans un mode de demande-réponse passif, le WR interroge le PS sur les éléments de données de la PT. Des exemples de protocoles suivant ce mode de fonctionnement incluent HTTP, FTP, LDAP, finger, et divers protocole d'appels de procédure à distance (RPC, *remote procedure call*) parmi lesquels Sun RPC, Environnement de calcul réparti (DCE, *Distributed Computing Environment*), Modèle d'objet composant réparti (DCOM, *Distributed Component Object Model*), Architecture commune de courtier de demande d'objet *(Corba, common object request broker architecture*), et Protocole simple d'accès aux objets (SOAP, *Simple Object Access Protocol*). Le PS utilise l'ensemble de règles pour déterminer si le WR est autorisé à accéder aux informations de la PT, refusant la demande si nécessaire. De plus, le PS pourrait filtrer les informations en supprimant des éléments ou en réduisant la résolution des éléments.

3.2 Demande-réponse active - PS comme client (initiateur)

Autrement, le PS peut contacter le WR et convoyer les éléments de données. Les exemples incluent HTTP, l'établissement de session SIP (demande INVITE), l'établissement de session H.323 ou SMTP.

3.3 Notification d'événement

La notification d'événement ajoute une phase d'abonnement au mode de fonctionnement "Demande-réponse active - PS comme client (initiateur)". Un observateur ou abonné demande à être ajouté à la liste de notification pour une présentité ou événement particulier. Quand la présentité change d'état ou quand l'événement se produit, le PS envoie un message au WR contenant l'état mis à jour. (La présence est un cas particulier de notification d'événement ; donc, on utilise souvent les termes de façon interchangeable.)

De plus, l'abonné peut lui-même ajouter un filtre à l'abonnement, limitant le taux ou le contenu des notifications. Si un événement, après filtrage par des règles fournies par le faiseur de règles et par des règles fournies par l'abonné, produit seulement le même contenu de notification qu'envoyé précédemment, aucune notification d'événement n'est envoyée.

Un seul PS peut autoriser l'accès aux éléments de données dans plus d'un mode. Plutôt que d'avoir différents ensembles de règles pour les différents modes, les trois modes sont pris en charge avec un schéma d'ensemble de règles. Les instances spécifiques d'ensemble de règles peuvent omettre des éléments qui sont seulement applicables au modèle d'abonnement.

4. Buts et hypothèses

Ci-dessous, on résume les objectifs et les contraintes de la conception.

Représentation en tableau : chaque règle doit être représentable par une rangée dans une base de données relationnelle. Cet objectif de conception devrait permettre une mise en œuvre de politique efficace en utilisant les techniques standard d'optimisation de base de données.

Permission seulement : les règles fournissent seulement des permissions plutôt que les refuser. Supprimer une règle ne peut jamais augmenter des permissions. Selon l'interprétation des règles 'refuser' et 'permettre', l'ordre des règles pourrait avoir de l'importance, rendant la mise à jour des ensembles de règles plus compliqués car de tels mécanismes de mise à jour auraient à prendre en charge l'insertion à des localisations spécifiques dans l'ensemble de règles. De plus, cela compliquerait les ensembles de règles répartis. Donc, seulement les actions 'permettre' sont permises, résultant en un traitement plus efficace des règles. Cela implique aussi que l'ordre des règles n'est pas important. Par conséquent, une décision de politique exige le traitement de toutes les règles.

Permissions additives : Une interrogation pour l'accès aux éléments de données est confrontée aux règles dans la base de données des règles. Si plusieurs règles correspondent, alors les permissions globales accordées au WR sont l'union de ces permissions. Une discussion plus détaillée est fournie à la Section 10.

Mise à niveau : Il devrait être possible d'ajouter ensuite des règles supplémentaires, sans casser les PS qui n'ont pas été mis à niveau. Aucune de ces mises à niveau ne doit dégrader les contraintes de confidentialité, mais les PS non encore mis à niveau peuvent révéler moins d'informations que ce que le faiseur de règles aurait choisi.

Prise en charge de capacités : en plus du but précédent, un RM devrait être capable de déterminer quelles extensions sont prises en charge par le PS. Le mécanisme utilisé pour déterminer la capacité d'un PS sort du domaine d'application de la présente spécification.

Indépendance au protocole : l'ensemble de règles prend en charge des contraintes sur les notifications ou interrogations ainsi que sur les abonnements pour les système fondés sur l'événement comme les systèmes de présence.

Pas de fausse assurance : il paraît plus dangereux de donner à l'utilisateur l'impression que le système va empêcher automatiquement la divulgation, mais échouer à le faire avec une probabilité significative d'erreur ou d'incompréhension de l'opérateur, que de forcer l'utilisateur à invoquer explicitement des règles plus simples. Par exemple, des règles fondées sur des gammes de jour de la semaine et d'heures du jour semblent particulièrement sujettes à la mauvaise interprétation et aux hypothèses fausses de la part du RM. (Par exemple, un RM non technique va probablement supposer que les règles se fondent sur la zone horaire de sa localisation courante, qui peut ne pas être connue des autres composants du système.)

5. Non buts

On a décidé explicitement qu'un certain nombre de capacités valables possibles sortent du domaine d'application de cette première version. De futures versions pourront inclure ces capacités, en utilisant le mécanisme d'extension décrit dans le présent document. Les non buts incluent :

Pas de références externes : les attributs dans des règles spécifiques ne peuvent pas se référer à des ensembles de règles, bases de données, répertoires, ou autres éléments de réseau externes. Toutes ces références externes rendraient une simple mise en œuvre de base de données difficile à appliquer et elles ne sont pas prises en charge par cette version.

Pas d'expressions régulières : les conditions sont confrontées sur des comparaisons d'égalité ou de style "plus grand que", pas sur des expressions régulières, des correspondances partielles telles que l'opérateur SQL LIKE (par exemple, LIKE "%foo%"), ou des correspondances de style glob ("*@exemple.com"). La plupart d'entre elles sont mieux exprimées par

des éléments explicites.

Pas de temps répétés : les temps répétés (par exemple, chaque jour de 9 h à 15 h) sont difficiles à mettre correctement en œuvre à cause des différentes zones horaires que PT, WR, PS, et RM peuvent occuper. Il apparaît que des suggestions pour inclure des intervalles de temps sont souvent fondées sur des distinctions entre fonctionnant/ne fonctionnant pas, qui sont malheureusement difficiles à capturer par le seul temps. Noter que cette caractéristique ne doit pas être confondue avec l'élément "Validity" qui fournit un mécanisme pour restreindre la durée de vie d'une règle.

6. Modèle et traitement des données de base

Un ensemble de règles (ou aussi dit une politique) consiste en zéro, une ou plusieurs règles. L'ordre de ces règles est sans importance. L'ensemble de règles peut être mémorisé au PS et convoyé du RM au PS comme un seul document, dans des sous ensembles ou dans des règles individuelles. Une règle comporte trois parties : les conditions (Section 7) les actions (Section 8) et les transformations (Section 9).

La partie conditions est un ensemble d'expressions, dont chacune s'évalue à VRAI ou FAUX. Quand un WR demande des informations sur une PT, le PS passe en revue chaque règle de l'ensemble de règles. Pour chaque règle, il évalue les expressions dans la partie conditions. Si toutes les expressions s'évaluent à VRAI, alors la règle est applicable à cette demande. Généralement, chaque expression spécifie une condition fondée sur une variable qui est associée au contexte de la demande. Ces variables peuvent inclure l'identité du WR, le domaine du WR, l'heure du jour, ou même des variables externes, comme la température ou l'humeur de la PT.

En supposant que la règle est applicable à la demande, les actions et transformations (couramment appelées des permissions) dans la règle spécifient comment le PS est supposé traiter cette demande. Si la demande est pour voir la localisation de la PT, ou pour voir sa présence, l'action normale est "permet", qui permet à la demande de poursuivre.

En supposant que l'action permet à la demande de se poursuivre, la partie transformations de la règle spécifie comment les informations sur la PT – ses informations de localisation, de présence, etc. -- sont modifiées avant d'être présentées au WR. Ces transformations sont sous la forme de permissions positives. C'est-à-dire, elles spécifient toujours un élément d'information qu'il est permis de voir au WR. Quand un PS traite une demande, il prend les transformations spécifiées dans toutes les règles qui correspondent, et crée leur union. Pour calculer cette union, les types de données, comme entier, booléen, ensemble, ou le type de données indéfini, jouent un rôle. Les détails de l'algorithme pour les permissions combinées sont décrits à la Section 10. L'union résultante représente effectivement un "gabarit" -- il définit quelles informations sont exposées au WR. Ce gabarit est appliqué aux données réelles de localisation ou de présence pour la PT, et les données qui sont permises par le gabarit sont montrées au WR. Si le WR demande seulement un sous ensemble des informations (comme seulement des données de localisation au niveau de la ville, au lieu de toutes les informations de localisation) les informations livrées au WR DOIVENT être l'intersection des permissions accordées au WR et des données demandées par le WR.

Les règles sont codées en XML. À cette fin, la Section 13 contient un schéma XML qui définit le langage de balisage de politique commune. Ceci est cependant un pur format d'échange entre le RM et le PS. Le format n'implique pas que le RM ou le PS utilise ce format en interne, par exemple, en confrontant une interrogation aux règles de politique. Les règles sont conçues pour qu'un PS puisse traduire les règles dans un tableau de base de données relationnelles, chaque règle étant représentée par une rangée de la base de données. La représentation en base de données n'est en aucune façon obligatoire ; on va l'utiliser comme un exemple pratique et largement compris d'une représentation interne. Le modèle de base de données a l'avantage que les opérations sur les rangées ont une signification bien définie. De plus, il paraît plausible que les mises en œuvre à plus grande échelle vont employer une base de données d'arrière plan pour mémoriser et interroger les règles, car elles peuvent alors bénéficier des mécanismes d'indexation optimisée existants, de contrôle d'accès, d'adaptabilité, et de contrainte d'intégrité. Les mises en œuvre de plus petite échelle peuvent bien choisir des mises en œuvre différentes, par exemple, une simple traversée de l'ensemble de règles.

6.1 Identification des règles

Chaque règle est équipée d'un paramètre qui identifie la règle. Cet identifiant de règle est un jeton opaque choisi par le RM. Un RM NE DOIT PAS utiliser le même identifiant pour deux règles disponibles au PS au même moment pour une PT donnée. Si plus d'un RM modifie le même ensemble de règles, il a alors besoin de s'assurer qu'un identifiant unique est choisi pour chaque règle. Un RM peut réaliser cela en restituant l'ensemble de règles déjà spécifié et en choisissant un nouvel identifiant pour une règle qui est différente de l'ensemble de règles existant.

6.2 Extensions

Le cadre de politique défini dans ce document est destiné à être extensible à des domaines d'application spécifiques. De telles extensions sont réalisées en définissant les conditions, actions, et transformations qui sont spécifiques du domaine d'application désiré. Chaque extension DOIT définir son propre espace de noms.

Les extensions ne peuvent pas changer le schéma défini dans le présent document, et ce schéma n'est pas supposé changer sauf révision de cette spécification. Donc, aucune procédure de versions n'est fournie pour ce schéma ou espace de noms.

7. Conditions

L'accès aux éléments de données doit être confronté à l'ensemble de règles mémorisées au PS. Chaque instance d'une demande a des attributs différents (par exemple, l'identité du demandeur) qui sont utilisés pour l'autorisation. Une règle dans un ensemble de règles pourrait avoir à satisfaire à un certain nombre de conditions avant d'exécuter les parties restantes d'une règle (c'est-à-dire, des actions et transformations). Les détails sur la confrontation de règle sont décrits à la Section 10. Le présent document spécifie seulement quelques conditions (c'est-à-dire, identité, sphère, et validité). D'autres éléments de condition peuvent être ajoutés via des extensions au présent document. Si un élément fils de l'élément <conditions> est dans un espace de nom inconnu ou non pris en charge, cet élément fils s'évalue à FAUX.

Comme noté à la Section 5, les conditions sont confrontées par des comparaisons pour égalité ou de style "plus grand que", plutôt que des expressions régulières. L'égalité est déterminée conformément aux règles pour les types de données associés à l'élément dans le schéma donné à la Section 13, sauf si des étapes explicites de comparaison sont incluses dans ce document. Pour les types xs:anyURI, les lecteurs peuvent souhaiter consulter la [RFC3987] pour sa discussion de xs:anyURI, ainsi que le texte de la Section 13.

7.1 Condition d'identité

7.1.1 Généralités

La condition identité restreint la confrontation d'une règle à une seule entité ou groupe d'entités. Seules des entités authentifiées peuvent être confrontées ; les moyens acceptables d'authentification sont définis dans des documents spécifiques du protocole. Si l'élément <identity> est absent, les identités ne sont pas considérées, et donc, les autres conditions de la règle s'appliquent à tout utilisateur, authentifié ou non.

La condition <identity> est considérée VRAIE si un quelconque de ses éléments fils (par exemple, les éléments <one/> et <many/> définis dans ce document) s'évaluent à VRAI, c'est-à-dire, les résultats des éléments fils individuels sont combinés en utilisant un OU logique.

Si un élément fils de l'élément <identity> est dans un espace de nom inconnu ou non pris en charge, alors cet élément fils s'évalue à FAUX.

7.1.2 Correspondance à une entité

L'élément <one> correspond à l'identité authentifiée (comme contenue dans l'attribut 'id') d'exactement une entité ou utilisateur. Pour les considérations qui concernent l'attribut 'id', voir au paragraphe 7.2.

On donne un exemple ci-dessous :

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy">
   <rule id="f3g44r1">
      <conditions>
         <identity>
            <one id="sip:alice@exemple.com"/>
            <one id="tel:+1-212-555-1234" />
            <one id="mailto:bob@exemple.net" />
```

```
        </identity>
      </conditions>
      <actions/>
      <transformations/>
    </rule>
</ruleset>
```

Cet exemple correspond si l'identité authentifiée du WR est sip:alice@exemple.com, tel:+1-212-555-1234, ou mailto:bob@exemple.net.

7.1.3 Correspondace à plusieurs entités

L'élément <many> est un mécanisme pour prendre des décisions d'autorisation sur la base de la partie domaine de l'identité authentifiée. À ce titre, il permet de correspondre à un grand nombre éventuellement inconnu d'utilisateurs au sein d'un domaine.

De plus, il est possible d'inclure un ou plusieurs éléments <except> pour exclure des utilisateurs individuels ou des utilisateurs appartenant à un domaine spécifique. Exclure des entités individuelles est mis en œuvre en utilisant une déclaration <except id="..."/>. La sémantique de l'attribut 'id' de l'élément <except> a la même signification que l'attribut 'id' de l'élément <one> (paragraphe 7.2). Exclure des utilisateurs qui appartiennent à un domaine spécifique est mis en œuvre en utilisant l'élément <except domain="..."/> qui exclut tout utilisateur appartenant au domaine indiqué.

Si plusieurs éléments <except> sont mentionnés comme éléments fils de l'élément <many>, alors le résultat de chaque élément <except> est combiné en utilisant un OU logique.

La politique commune DOIT utiliser UTF-8 ou UTF-16 pour mémoriser les noms de domaines dans l'attribut 'domain'. Pour les noms de domaine qui ne sont pas internationalisés (IDN, *Internationalized Domain Name*) l'ASCII en minuscules DEVRAIT être utilisé. Pour l'opération de comparaison entre la valeur mémorisée dans l'attribut 'domain' et la valeur de domaine fournie via le protocole utilisateur (appelée un "identifiant de domaine de protocole") les règles suivantes sont applicables :

1. Traduire le codage en pourcentage pour toute chaîne.
2. Convertir les deux chaînes de domaine en utilisant l'opération ToASCII décrite dans la [RFC3490].
3. Comparer les deux chaînes de domaine en égalité ASCII, pour chaque étiquette. Si la comparaison de chaîne pour chaque étiquette indique l'égalité, la comparaison réussit. Autrement, les domaines ne sont pas égaux.

Si la conversion échoue à l'étape (2), les domaines ne sont pas égaux.

7.1.3.1 Correspondance à toute entité authentifiée

L'élément <many/>sans aucun élément fils ou attribut correspond à tout utilisateur authentifié.

L'exemple suivant montre une telle règle qui correspond à tout utilisateur authentifié :

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy">

  <rule id="f3g44r5">
    <conditions>
      <identity>
        <many/>
      </identity>
    </conditions>
    <actions/>
    <transformations/>
  </rule>
</ruleset>
```

7.1.3.2 Correspondance à toute identité authentifiée sauf les domaines/identités énumérés

L'élément <many> contenant un ou plusieurs éléments <except domain="..."/> correspond à tout utilisateur de tout domaine sauf ceux mentionnés. L'élément <except id="..."/> exclut des utilisateurs particuliers. La sémantique de l'attribut 'id' de l'élément <except> est décrite au paragraphe 7.2. Les résultats des éléments fils de l'élément <many> sont combinés en utilisant un OU logique.

Un exemple est montré ci-dessous :

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy">
<rule id="f3g44r1">
    <conditions>
      <sphere value="work"/>
      <identity>
        <many>
          <except domain="exemple.com"/>
          <except domain="exemple.org"/>
          <except id="sip:alice@bad.exemple.net"/>
          <except id="sip:bob@good.exemple.net"/>
          <except id="tel:+1-212-555-1234" />
          <except id="sip:alice@exemple.com"/>
        </many>
      </identity>
      <validity>
        <from>2003-12-24T17:00:00+01:00</from>
        <until>2003-12-24T19:00:00+01:00</until>
      </validity>
    </conditions>
    <actions/>
    <transformations/>
  </rule>
</ruleset>
```

Cet exemple correspond à tous les utilisateurs excepté tout utilisateur dans exemple.com, ou dans exemple.org ou les utilisateurs particuliers alice@bad.exemple.net, bob@good.exemple.net, et celui qui a le numéro de téléphone 'tel:+1-212-555-1234'. Le dernier élément 'except' est redondant car alice@exemple.com est déjà exclu par la première ligne.

7.1.3.3 Correspondance à toute identité authentifiée dans un domaine sauf les identités énumérées

L'élément <many> avec un attribut 'domain' et zéro, un ou plusieurs éléments <except id="..."/> correspond à tout utilisateur authentifié provenant du domaine indiqué sauf ceux explicitement énumérés. La sémantique de l'attribut 'id' de l'élément <except> est décrite au paragraphe 7.2.

Il n'y a pas de sens à avoir des domaines dans l'attribut 'id' qui ne correspondent pas à la valeur de l'attribut 'domain' dans l'élément <many> qui l'enclôt.

Un exemple est montré ci-dessous :

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy">

  <rule id="f3g44r1">
    <conditions>
      <identity>
        <many domain="exemple.com">
          <except id="sip:alice@exemple.com"/>
          <except id="sip:bob@exemple.com"/>
        </many>
      </identity>
    </conditions>
```

```
    <actions/>
    <transformations/>
  </rule>
</ruleset>
```

Cet exemple correspond à tout utilisateur au sein de exemple.com (comme carol@exemple.com) sauf alice@exemple.com et bob@exemple.com.

7.2 Une seule entité

L'attribut 'id' utilisé dans les éléments <one> et <except> se réfère à une seule entité. Dans le texte qui suit, on utilise le terme "entité d'un seul utilisateur" comme synonyme des éléments <one> et <except>. L'élément <except> répond à l'objet d'exclure des éléments de l'ensemble solution.

Une entité d'un seul utilisateur correspond à l'identité authentifiée (telle que contenue dans l'attribut 'id') de exactement une entité ou un utilisateur. Si il y a correspondance, l'entité d'un seul utilisateur est considérée comme VRAIE. L'entité d'un seul utilisateur NE DOIT PAS contenir d'attribut 'domaine'.

L'attribut 'id' contient une identité qui DOIT d'abord être exprimée comme un URI. Les applications qui utilisent le présent cadre doivent décrire comment les identités qu'elles utilisent peuvent être exprimées comme des URI.

7.3 Sphère

L'élément <sphere> appartient au groupe des éléments de condition. Il peut être utilisé pour indiquer un état (par exemple, 'travail', 'domicile', réunion', 'voyage') dans lequel la PT est actuellement. Une condition de sphère correspond seulement si la PT est actuellement dans l'état indiqué. L'état peut être convoyé par configuration manuelle ou par un protocole. Par exemple, RPID [RFC4480] donne la capacité d'informer le PS de sa sphère actuelle. Le domaine d'application doit décrire plus en détails comment l'état de sphère est déterminé. Passer d'une sphère à une autre cause une commutation entre les différents modes de visibilité. Par suite, différents sous ensembles de règles peuvent être applicables.

Le contenu de l'attribut 'valeur' de l'élément <sphere> PEUT contenir plus d'un jeton. Les jetons individuels DOIVENT être séparés par un caractère blanc. Un OU logique est utilisé pour confronter les jetons aux réglages de sphère de la PT. Par exemple, si le contenu de l'attribut 'valeur' dans l'attribut sphere contient deux jetons 'travail' et 'domicile', alors cette partie de la règle correspond si la sphère pour une PT particulière est 'travail' OU 'domicile'. Pour comparer le contenu de l'attribut 'valeur' dans l'élément <sphere> aux informations d'état mémorisées sur les réglage de sphère de la PT, une comparaison de chaîne insensible à la casse DOIT être utilisée pour chaque jeton individuel. Il n'y a ni un registre pour ces valeurs ni une indication spécifique du langage du contenu de la sphère. À ce titre, les jetons sont traités comme des chaînes opaques.

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy">

<rule id="f3g44r2">
  <conditions>
    <sphere value="work"/>
    <identity>
      <one id="sip:andrew@exemple.com"/>
    </identity>
  </conditions>
  <actions/>
  <transformations/>
</rule>

<rule id="y6y55r2">
  <conditions>
    <sphere value="home"/>
    <identity>
      <one id="sip:allison@exemple.com"/>
    </identity>
  </conditions>
```

```
  <actions/>
  <transformations/>
</rule>

<rule id="z6y55r2">
  <conditions>
    <identity>
        <one id="sip:john@doe.exemple.com"/>
    </identity>
    <sphere value="home work"/>
  </conditions>
  <actions/>
  <transformations/>
 </rule>
</ruleset>
```

L'exemple de règle ci-dessus illustre que la règle avec l'entité andrew@exemple.com correspond si la sphère est réglée à 'work'. Dans la seconde règle, l'entité allison@exemple.com correspond si la sphère est réglée à 'home'. La troisème règle correspond aussi car la valeur dans l'élément sphère contient aussi le jeton 'home'.

7.4 Validité

L'élément <validity> est le tpoisième élément de condition spécifié dans ce document. Il exprime la période de validité de la règle par deux attributs, une heure de début et une heure de fin. La condition de validité est VRAIE si l'heure actuelle est supérieure ou égale à au moins un <from> fils, mais moins que le fils <until> après lui. Cela représente un opération OU logique sur chaque paire <from> et <until>. Les temps sont exprimés en format XML dateTime. Un faiseur de règle peut ne pas avoir toujours accès au PS pour invalider certaines règles qui accordent les permissions. Donc, ce mécanisme permet d'invalider automatiquement les permissions accordées sans autre interaction entre le faiseur de règle et le PS. Le PS ne supprime pas les règles ; le faiseur de règles doit plutôt les nettoyer.

Un exemple de fragment de règle est montré ci-dessous:

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy">

  <rule id="f3g44r3">
    <conditions>
      <validity>
        <from>2003-08-15T10:20:00.000-05:00</from>
        <until>2003-09-15T10:20:00.000-05:00</until>
      </validity>
    </conditions>
    <actions/>
    <transformations/>
  </rule>
</ruleset>
```

L'élément <validity> DOIT avoir les sous éléments <from> et <until> en paires. Plusieurs éléments <from> et <until> peuvent apparaître en paires (c'est-à-dire, sans incorporer d'éléments <from> et <until>). Utiliser plusieurs éléments <validity> comme sous éléments de l'élément <conditions> n'est pas utile car tous les sous éléments de l'élément <conditions> sont combinés par un ET logique.

8. Actions

Alors que les conditions sont la partie 'si' des règles, les actions et transformations forment leur partie 'alors'. Les parties actions et transformations d'une règle déterminent quelles opérations le PS DOIT exécuter après avoir reçu d'un WR une demande d'accès à des données qui correspond à toutes les conditions de cette règle. Les actions et transformations permettent seulement certaines opérations ; il n'y a pas de fonction de 'refus'. Les transformations spécifient exclusivement

des opérations du côté PS qui conduisent à une modification des éléments de données demandés par le WR. Concernant les éléments de données de localisation, par exemple, une transformation pourrait forcer le PS à diminuer la précision des informations de localisation qui sont reoturnées au WR.

Les actions spécifient par ailleurs tous les types d'opérations restants que le PS est obligé d'exécuter, c'est-à-dire, toutes les opérations qui ne sont pas du type transformation. Les actions sont définies par les usages spécifiques de l'application de ce cadre. Le lecteur est renvoyé aux extensions correspondantes pour voir des exemples de ces éléments.

9. Transformations

Deux sous parties suivent la partie conditions d'une règle : transformations et actions. Comme défini à la Section 8, les transformations spécifient des opérations que le PS DOIT exécuter et qui modifient le résultat qui est retourné au WR. Cette fonction est particulièrement utile pour réduire la granularité des informations fournies au WR, comme, par exemple, ce qui est nécessaire pour la confidentialité de la localisation. Les transformations sont définies par des utilisations spécifiques de l'application de ce cadre.

Un exemple de simple transformation est fourni à la Section 10.

10. Procédure pour combiner les permissions

10.1 Introduction

Cette Section décrit comment les règles sont choisies et comment les actions et permissions sont déterminées. Quand un PS reçoit une demande d'accès à des données sensibles à la confidentialité, la demande est confrontée à l'ensemble de règles. Une règle correspond si toutes les conditions contenues comme éléments fils dans l'élément <conditions> d'une règle s'avaluent à VRAI. Chaque type de condition définit quand il est VRAI. Toutes les règles où les conditions satisfont à la demande forment l'ensemble de règles correspondant. Les permissions dans l'ensemble de règles correspondant sont combinées en utilisant un ensemble de règles de combinaison (CR) décrit au paragraphe 10.2.

10.2 Règles de combinaison (CR)

Chaque type de permission est combiné sur toutes les règles correspondantes. Chaque type d'action ou de transformation est combiné séparément et de façon indépendante. Les règles de combinaison génèrent une permission combinée. Les règles de combinaison dépendent seulement du type des données de permission. Si un type particulier de permission n'a pas de valeur dans une règle, on suppose la plus basse valeur possible pour cette permission pour les besoins du calcul de la permission combinée. Cette valeur est donnée par les types de données pour les booléens (FAUX) et ensembles (ensemble vide) et DOIT être défini par toute extension de la politique commune pour les autres types de données.

Pour les permissions booléennes, la permission résultante est VRAIE si et seulement si au moins une permission dans l'ensemble de règles correspondantes a une valeur de VRAI, et FAUX autrement. Pour entier, permissions de valeur réelle, et date-heure, la permission résultante est la valeur maximum entre les valeurs de permission dans l'ensemble de règles correspondantes. Pour les ensembles, c'est l'union des valeurs parmi les permissions dans l'ensemble de règles correspondant.

10.3 Exemple

Dans l'exemple suivant, on illustre le processus de combinaison de permissions. On va considérer trois conditions à cette fin, à savoir celles d'identité de nom (WR-ID), de sphère, et de validité (from,until). La colonne Identifiant est utilisée comme un identifiant de règle. Pour des raisons de typographie, on omet la partie domaine de l'identité du WR.

On utilise deux actions dans notre exemple, X et Y. Les valeurs de X et Y sont des types de données respectivement booléennes et entières.

La transformation, appelée Z, utilise des valeurs qui peuvent être réglées à '+' (ou 3), 'o' (ou 2) ou '-' (ou 1). La permission Z nous permet de montrer la réduction de granularité par laquelle une valeur de '+' montre sans restriction les informations correspondantes, et '-' ne montre rien. Cette permission pourrait se rapporter à des informations de localisation ou à d'autres attributs de présence comme mood. En interne, on utilise le type de données Entier pour calculer la permission de cet

attribut.

L'étiquette 'NUL' dans le tableau indique qu'aucune valeur n'estdisponible pour une cellule particulière.

Conditions					**Actions/Transformations**		
Id	**WR-ID**	**sphère**	**depuis**	**jusqu'à**	**X**	**Y**	**Z**
1	bob	domicile	A1	A2	VRAI	10	o
2	alice	travail	A1	A2	FAUX	5	+
3	bob	travail	A1	A2	VRAI	3	-
4	tom	travail	A1	A2	VRAI	5	+
5	bob	travail	A1	A3	NUL	12	o
6	bob	travail	B1	B2	FAUX	10	-

Encore pour des raisons typographiques on utilise les abréviations suivantes pour les deux attributs de <validité> 'depuis' et 'jusqu'à' :

A1=2003-12-24T17:00:00+01:00
A2=2003-12-24T21:00:00+01:00
A3=2003-12-24T23:30:00+01:00
B1=2003-12-22T17:00:00+01:00
B2=2003-12-23T17:00:00+01:00

Noter que B1 < B2 < A1 < A2 < A3.

L'entité 'bob' agit comme WR et demande des éléments de données. L'ensemble de règles consiste en les six règles montrées dans le tableau et identifiées par les valeurs 1 à 6 dans la colonne 'Id'. Le PS reçoit l'interrogation à 2003-12-24T17:15:00+01:00, qui tombe entre A1 et A2. Dans notre exemple, on suppose que la valeur de sphère de la PT est actuellement réglé à 'travail'.

Dans une premère étape, il est nécessaire de déterminer quelles règles s'appliquent en évaluant la partie conditions de chacune d'elles.

La règle 1 ne correspond pas car la condition sphère n'est pas satisfaite. La règle 2 ne correspond pas car l'identité du WR (ici 'alice') n'est pas égale à 'bob'. La règle 3 correspond car toutes les conditions s'évaluent à VRAI. La règle 4 ne correspond pas car l'identité du WR (ici 'tom') n'est pas égale à 'bob'. La règle 5 correspond. La règle 6 ne correspond pas car la règle n'est plus valide.

Seules les règles 3 et 5 s'appliquent. On utilise la partie actions et transformations de ces deux règles pour déterminer la permission combinée, comme on le montre ci-dessous.

	Actions/Transformations		
Id	**X**	**Y**	**Z**
3	VRAI	3	-
5	NUL	12	o

Le traitement de chaque colonne est indépendant. La valeur combinée de X est réglée à VRAI car la valeur NUL est égale à FAUX en accord avec la description du paragraphe 10.2. Pour la colonue Y, on applique le maximum de 3 et 12, de sorte que la valeur combinée de Y est 12. Pour la colonne Z, on calcule là encore le maximum de 'o' et '-' (c'est-à-dire, 2 et 1) qui est 'o' (2).

La permission combinée des trois colonnes est donc :

Actions/Transformations		
X	Y	Z
VRAI	12	o

11 Méta politiques

Les méta politiques autorisent un faiseur de règles à insérer, mettre à jour, ou supprimer une règle particulière ou un

ensemble entier de règles. Certaines politiques d'autorisation sont nécessaires pour empêcher la modification non autorisée d'ensembles de règles. Les méta politiques sortent du domaine d'application du présent document. Une mise en œuvre simple pourrait restrendre l'accès à l'ensemble de règles à la seule PT mais des mécanimses plus sophistiqués pourraient être utiles. Comme exemple de telles politiques, on pourrait penser à des parents qui configurent les politiques pour leurs enfants.

12. Exemple

Cette Section donne un exemple d'un document XML valide par rapport au schéma XML défini à la Section 13. On trouvera des exemples sémantiquement plus riches dans les documents qui étendent ce schéma avec des données spépcifiques du domaine d'application (par exemple, des informations de localisation ou de présence).

On montre ci-dessous une règle avec une condition qui correspond pour une certaine identité authentifiée (bob@exemple.com) et dans une période de temps donnée. De plus, la règle correspond seulement si la cible a établi sa sphère à 'travail'.

```
<?xml version="1.0" encoding="UTF-8"?>
<ruleset xmlns="urn:ietf:params:xml:ns:common-policy">

  <rule id="f3g44r1">
    <conditions>
      <identity>
        <one id="sip:bob@exemple.com"/>
      </identity>
      <valeur de sphère ="travail"/>
      <validity>
        <depuis>2003-12-24T17:00:00+01:00</depuis>
        <jusqu'à>2003-12-24T19:00:00+01:00</jusqu'à>
      </validity>
    </conditions>
    <actions/>
    <transformations/>
  </rule>
</ruleset>
```

13. Définition du schéma XML

Cette Section donne la définition du schéma XML pour le langage de balisage de politique commune décrite dans ce document.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:common-policy"
  xmlns:cp="urn:ietf:params:xml:ns:common-policy"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <!-- /ruleset -->
  <xs:element name="ruleset">
    <xs:complexType>
      <xs:complexContent>
        <xs:restriction base="xs:anyType">
          <xs:sequence>
            <xs:element name="rule" type="cp:ruleType"
            minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
```

```
</xs:element>
<!-- /ruleset/rule -->
<xs:complexType name="ruleType">
  <xs:complexContent>
    <xs:restriction base="xs:anyType">
      <xs:sequence>
        <xs:element name="conditions"
        type="cp:conditionsType" minOccurs="0"/>
        <xs:element name="actions"
        type="cp:extensibleType" minOccurs="0"/>
        <xs:element name="transformations"
        type="cp:extensibleType" minOccurs="0"/>
      </xs:sequence>
      <xs:nom d'attribut="id" type="xs:ID" use="required"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<!-- //rule/conditions -->
<xs:complexType name="conditionsType">
  <xs:complexContent>
    <xs:restriction base="xs:anyType">
      <xs:choice maxOccurs="unbounded">
        <xs:element name="identity"
        type="cp:identityType" minOccurs="0"/>
        <xs:element name="sphere"
        type="cp:sphereType" minOccurs="0"/>
        <xs:element name="validity"
        type="cp:validityType" minOccurs="0"/>
        <xs:any namespace="##autre" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
      </xs:choice>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<!-- //conditions/identity -->
<xs:complexType name="identityType">
  <xs:complexContent>
    <xs:restriction base="xs:anyType">
      <xs:choice minOccurs="1" maxOccurs="unbounded">
        <xs:element name="one" type="cp:oneType"/>
        <xs:element name="many" type="cp:manyType"/>
        <xs:any namespace="##autre" processContents="lax"/>
      </xs:choice>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<!-- //identity/one -->
<xs:complexType name="oneType">
  <xs:complexContent>
    <xs:restriction base="xs:anyType">
      <xs:sequence>
        <xs:any namespace="##autre"
        minOccurs="0" processContents="lax"/>
      </xs:sequence>
      <xs:nom d'attribut="id"
      type="xs:anyURI" use="required"/>
    </xs:restriction>
  </xs:complexContent>
</xs:complexType>
<!-- //identity/many -->
<xs:complexType name="manyType">
```

```
      <xs:complexContent>
        <xs:restriction base="xs:anyType">
          <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element name="except" type="cp:exceptType"/>
            <xs:any namespace="##autre"
            minOccurs="0" processContents="lax"/>
          </xs:choice>
          <xs:nom d'attribut="domain"
          use="optional" type="xs:string"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
    <!-- //many/except -->
    <xs:complexType name="exceptType">
      <xs:nom d'attribut="domain" type="xs:string" use="optional"/>
      <xs:nom d'attribut="id" type="xs:anyURI" use="optional"/>
    </xs:complexType>
    <!-- //conditions/sphere -->
    <xs:complexType name="sphereType">
      <xs:complexContent>
        <xs:restriction base="xs:anyType">
          <xs:nom d'attribut="valeur"
          type="xs:string" use="required"/>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
    <!-- //conditions/validity -->
    <xs:complexType name="validityType">
      <xs:complexContent>
        <xs:restriction base="xs:anyType">
          <xs:sequence minOccurs="1" maxOccurs="unbounded">
            <xs:element name="depuis" type="xs:dateTime"/>
            <xs:element name="jusqu'à" type="xs:dateTime"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
    <!-- //règle/actions ou //règle/transformations -->
    <xs:complexType name="extensibleType">
      <xs:complexContent>
        <xs:restriction base="xs:anyType">
          <xs:sequence>
            <xs:any namespace="##autre" processContents="lax"
            minOccurs="0" maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:restriction>
      </xs:complexContent>
    </xs:complexType>
</xs:schema>
```

14. Considérations sur la sécurité

Le présent document décrit un cadre pour des politiques. Ce cadre est destiné à être amélioré par ailleurs par des données spécifiques du domaine d'application. Les considérations sur la sécurité sont dans une grande mesure dépendantes des données d'application, et doivent donc être couvertes par les documents qui étendent le cadre défini dans la présente spécification. Cependant, de nouvelles permissions d'action et de transformation ainsi que leurs valeurs permises doivent être définies d'une façon telle que l'usage des permissions combinant les règles de la Section 10 ne diminue pas le niveau de protection de la confidentialité. Voir plus de détails à la Section 10 sur cette question de confidentialité.

15. Considérations relatives à l'IANA

Cette Section enregistre un nouvel espace de noms XML, un nouveau schéma XML, et un nouveau type MIME. Cette Section enregistre un nouvel espace de noms XML selon les procédures de la [RFC3688].

15.1 Enregistrement d'espace de noms de politique commune

URI: urn:ietf:params:xml:ns:common-policy

Contact d'enregistrement : groupe de tavail IETF GEOPRIV, Henning Schulzrinne (hgs+geopriv@cs.columbia.edu).

XML :

DÉBUT
```
<?xml version="1.0"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="content-type"
       content="text/html;charset=iso-8859-1"/>
  <title>Espace de noms de politique commune</title>
</head>
<body>
  <h1>Espace de noms pour les politiques communes d'autorisation</h1>
  <h2>urn:ietf:params:xml:ns:common-policy</h2>
<p>Voir <a href="ftp://ftp.rfc-editor.org/in-notes/rfc4745.txt">
   RFC 4745</a>.</p>
</body>
</html>
```
FIN

15.2 Enregistrement de type de contenu pour 'application/auth-policy+xml'

Cette spécification demande l'enregistrement d'un nouveau type MIME en accord avec les procédures de la [RFC4288] et les lignes directrices de la [RFC3023].

Nom du tupe de support MIME : application
Nom du sous-type MIME : auth-policy+xml
Paramètres obligatoires : aucun
Paramètres facultatifs : charset. Indique le codage du jeu de caractères du XML enclos.
Considérations de codage : utilise XML, qui peut employer des caractères de 8 bits, selon le codage de caractères utilisé. Voir au paragraphe 3.2 de la [RFC3023].
Considérations de sécurité : ce type de contenu est conçu pour porter des politiques d'autorisation. Les précautions appropriées devraient être adoptées pour limiter la divulgation de ces informations. Voir la Section 14 de la RFC 4745 et les considérations sur la sécurité de la [RFC3023].
Considérations d'interopérabilité : aucune
Spécification publiée : RFC 4745
Applications qui utilisent ce type de support : systèmes fondés sur présence et de localisation.
Informations supplémentaires :
Numéro magique : aucun
Extension de fichier : .apxml
Code de type de fichier Macintosh : 'TEXT'
Adresse personnelle et de messagerie pour plus d'informations : Hannes.Tschofenig@siemens.com
Usage prévu : USAGE LIMITÉ
Auteur : cette spécification est un produit du groupe de travail GEOPRIV de l'IETF, à <geopriv@ietf.org>.
Contrôleur des changements : IESG <iesg@ietf.org>

15.3 Enregistrement de schéma de politique commune

URI : urn:ietf:params:xml:schema:common-policy
Contact d'enregistrement : groupe de travail IETF GEOPRIV, Henning Schulzrinne (hgs+geopriv@cs.columbia.edu).
XML : le schéma XML à enregistrer est contenu à la Section 13. Sa première ligne est <?xml version="1.0" encoding ="UTF-8"?> et sa dernière ligne est </xs:schema>

16. Références

16.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC3023] M. Murata, S. St.Laurent et D. Kohn, "Types de support XML", janvier 2001. (*Obsolète, voir* RFC7303)

[RFC3490] P. Faltstrom et autres, "Internationalisation des noms de domaine dans les applications (IDNA)", mars 2003. (*Remplacée par les* RFC5890 et 5891, *P.S.*)

[RFC3688] M. Mealling, "Registre XML de l'IETF", BCP 81, janvier 2004.

[RFC3987] M. Duerst et M. Suignard, "Identifiant de ressource internationalisé (IRI)", janvier 2005.

[RFC4288] N. Freed et J. Klensin, "Spécifications du type de support et procédures d'enregistrement", BCP 13, décembre 2005.

16.2 Références pour information

[RFC3693] J. Cuellar et autres, "Exigences pour Geopriv", février 2004. *(Information)*

[RFC4480] H. Schulzrinne et autres, "RPID : Extensions Rich Presence au format de données d'information Presence (PIDF)", juillet 2006. *(P.S.)*

[RFC5025] J. Rosenberg, "Règles d'autorisation de présence", décembre 2007. *(P.S.)*

[RFC6772] H. Schulzrinne, éd., H. Tschofenig, éd., J. Cuellar, J. Polk, J. Morris et M. Thomson, "Politique de géolocalisation : format de document pour exprimer les préférences de confidentialité pour les informations de localisation", janvier 2013.

Appendice A. Contributeurs

Nous tenons à remercier Christian Guenther de son aide pour les versions initiales de ce document.

Appendice B. Remerciements

Le présent document est partiellement fondé sur des discussions au sein du groupe de travail GEOPRIV de l'IETF. Les discussions à la réunion intermédiaire Geopriv de 2003 à Washington, D.C., ont aidé le groupe de travail à progresser sur les politiques d'autorisation sur la base des discussions entre les participants.

Nous tenons à remercier particulièrement Allison Mankin <mankin@psg.com>, Randall Gellens <rg+ietf@qualcomm.com>, Andrew Newton <anewton@ecotroph.net>, Ted Hardie <hardie@qualcomm.com>, et Jon Peterson <jon.peterson@neustar.biz> qui ont discuté un certain nombre de détails avec nous. Ils nous ont aidé à améliorer la qualité de ce document. Allison, Ted, et Andrew nous ont aussi aidé à bien progresser sur la prise en charge de l'internationalisation des attributs d'identifiant/domaine.

De plus, nous tenons à remercier le groupe de travail SIMPLE de l'IETF pour les discussions sur le projet de J. Rosenberg des politiques d'autorisation de présence. Merci aussi à Stefan Berg, Murugaraj Shanmugam, Christian Schmidt, Martin

Thomson, Markus Isomaki, Aki Niemi, Eva Maria Leppanen, Josip Matanovic, et Mark Baker de leurs commentaires. Martin Thomson nous a aidé sur le schéma XML. Mark Baker a fait la relecture du type de support. Scott Brim a fait une relecture au nom de l'équipe de révision des domaines généraux.

Adresse des auteurs

Henning Schulzrinne
Columbia University
Department of Computer Science
450 Computer Science Building
New York, NY 10027
USA
téléphone : +1 212 939 7042
mél : schulzrinne@cs.columbia.edu
URI : http://www.cs.columbia.edu/~hgs

Hannes Tschofenig
Siemens Networks GmbH & Co KG
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany
mél : Hannes.Tschofenig@siemens.com
URI : http://www.tschofenig.com

John B. Morris, Jr.
Center for Democracy et Technology
1634 I Street NW, Suite 1100
Washington, DC 20006
USA
mél : jmorris@cdt.org
URI : http://www.cdt.org

Jorge R. Cuellar
Siemens
Otto-Hahn-Ring 6
Munich, Bavaria 81739
Germany
mél : Jorge.Cuellar@siemens.com

James Polk
Cisco
2200 East President George Bush Turnpike
Richardson, Texas 75082
USA
mél : jmpolk@cisco.com

Jonathan Rosenberg
Cisco Systems
600 Lanidex Plaza
Parsippany, New York 07054
mél : jdrosen@cisco.com
URI : http://www.jdrosen.net

Déclaration complète de droits de reproduction

Remerciement
Le financement de la fonction d'édition des RFC est fourni par l'activité de soutien administratif (IASA) de l'IETF.