

Groupe de travail Réseau
Request for Comments : 4826
 Catégorie : Sur la voie de la normalisation

J. Rosenberg, Cisco
 mai 2007
 Traduction Claude Brière de L'Isle

Formats du langage de balisage extensible (XML) pour représenter les listes de ressources

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.

Notice de Copyright

Copyright (C) The IETF Trust (2007).

Résumé

Dans les communications multimédia, de présence, et de systèmes de messagerie instantanée, il y a un besoin de définir des identifiants de ressource universels (URI, *Uniform Resource Identifier*) qui représentent des services associés à un groupe d'utilisateurs. Un exemple est un service de liste de ressources. Si un usager envoie un message SUBSCRIBE du protocole d'initialisation de session (SIP, *Session Initiation Protocol*) à l'URI qui représente le service de liste de ressources, le serveur va obtenir l'état des usagers dans le groupe associé, et le fournir à l'expéditeur. Pour faciliter la définition de ces services, la présente spécification définit deux documents de langage de balisage extensible (XML, *Extensible Markup Language*). Un document contient les URI de service, avec leur définition de service et une référence au groupe d'utilisateurs associé. Le second document contient la liste des utilisateurs qui sont référencés d'après le premier. Cette liste d'utilisateurs peut être utilisée par d'autres applications et services. Les deux documents peuvent être créés et gérés avec le protocole d'accès à la configuration XML (XCAP, *XML Configuration Access Protocol*).

Table des matières

1. Introduction.....	2
2. Terminologie.....	2
3. Documents de listes de ressources.....	3
3.1 Structure.....	3
3.2 Schéma.....	4
3.4 Usage avec XCAP.....	6
3.4.6 Sémantique des données.....	7
4. Documents de services RLS.....	8
4.1 Structure.....	8
4.2 Schéma.....	9
4.3 Exemple de document.....	9
4.4 Usage avec XCAP.....	10
4.5 Usage d'un document de services RLS par un RLS.....	13
5. Canonisation d'URI SIP.....	13
6. Extensibilité.....	14
7. Considérations sur la sécurité.....	14
8. Considérations relatives à l'IANA.....	14
8.1 Identifiants uniques d'application XCAP.....	15
8.2 Enregistrements de Type MIME.....	15
8.3 Enregistrements de sous espace de noms d'URN.....	16
8.4 Enregistrements de schémas.....	17
9. Remerciements.....	17
10. Références.....	17
10.1 Références normatives.....	17
10.2 Références pour information.....	18
Adresse de l'auteur.....	18
Déclaration complète de droits de reproduction.....	18

1. Introduction

Le protocole d'initialisation de session (SIP, *Session Initiation Protocol*) [RFC3261] définit l'identifiant de ressource universel (URI, *Uniform Resource Identifier*) SIP comme toute ressource à laquelle une demande SIP peut être générée pour établir une forme quelconque d'opération de communication. Ces URI peuvent représenter des utilisateurs (par exemple, sip:joe@exemple.com). L'URI SIP peut aussi représenter un service, comme la messagerie vocale, une conférence, ou une liste de présence. Un schéma courant à travers de tels services SIP est que le service est défini par, et associé à, un URI. Pour fonctionner, ce service doit utiliser une liste d'utilisateurs (ou, plus généralement, une liste de ressources). Quand une demande SIP est envoyée à l'URI de service, le serveur qui fournit le service lit cette liste, et effectue ensuite une opération sur chaque ressource de la liste. C'est ce que montre la Figure 1.

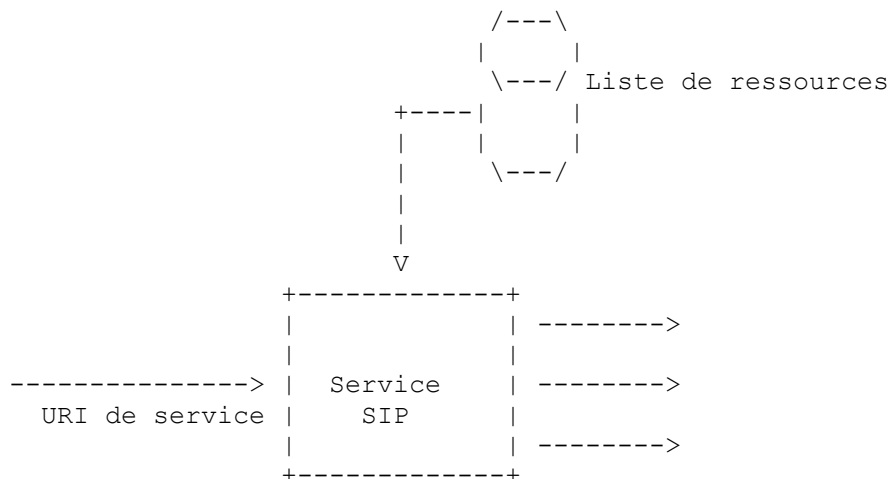


Figure 1

Un important exemple d'un tel service est un service de liste de présence [RFC3856]. Un service de liste de présence permet à un client de générer une demande SIP SUBSCRIBE pour demander des informations de présence pour une liste d'utilisateurs. Le serveur de liste de présence obtient la présence pour les utilisateurs de la liste et les fournit au client. Un serveur de liste de présence est un cas spécifique de serveur de liste de ressources (RLS, *Resource List Server*) [RFC4662], qui permet à un client de générer une demande SIP SUBSCRIBE pour demander les notifications des événements SIP pour une liste de ressources.

Un autre exemple d'un tel service est un service de conférence instantanée. Si un client envoie une demande SIP INVITE à l'URI représentant l'instance de service de conférence, le serveur de conférence va créer un appel de conférence contenant le client et le groupe d'utilisateurs associé.

Il est très utile pour un utilisateur de ces systèmes de définir les groupes d'utilisateurs ou de ressources (généralement appelés une liste de ressources) séparément des services qui accèdent à ces listes de ressources. Bien sûr, il y a des usages des listes de ressources même en l'absence de tout service associé fondé sur le réseau. Par exemple, plutôt que d'utiliser un service de liste de présence, un client pourrait générer des demandes SUBSCRIBE individuelles pour obtenir la présence de chaque utilisateur dans une liste de présence mémorisée en local. Dans ce cas, il y a besoin d'un format pour mémoriser la liste en local sur un disque. De plus, l'utilisateur pourrait souhaiter partager la liste avec des amis, et désireux l'envoyer par messagerie électronique à ces amis. Cela exige aussi un format normalisé pour la liste de ressources.

À ce titre, le présent document définit deux formats de documents en langage de balisage extensible (XML, *Extensible Markup Language*). Le premier est utilisé pour représenter les listes de ressources, indépendamment de tout service particulier. Le second est utilisé pour définir les URI de service pour un RLS, et pour associer une liste de ressources à l'URI de service. Le présent document définit aussi un usage d'application du protocole d'accès à la configuration XML (XCAP, *XML Configuration Access Protocol*) [RFC4825] pour gérer chacun de ces deux documents.

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 la [RFC2119] et indiquent les niveaux d'exigence pour les mises en œuvre conformes.

3. Documents de listes de ressources

3.1 Structure

Un document de listes de ressources est un document XML [XML] qui DOIT être bien formé et DOIT être valide en accord avec les schémas, incluant les schémas d'extension, disponibles au valideur et applicables au document XML. Les documents de listes de ressources DOIVENT être fondés sur XML 1.0 et DOIVENT être codés en utilisant UTF-8. La présente spécification utilise les espaces de noms XML pour identifier les documents de listes de ressources et les fragments de document. L'URI d'espace de noms pour les éléments définis par la présente spécification est un URN [RFC2141] qui utilise l'identifiant d'espace de noms 'ietf' défini par la [RFC2648] et étendu par la [RFC3688]. Cet URN est : urn:ietf:params:xml:ns:resource-lists

Un document de listes de ressources a l'élément <resource-lists> comme élément racine du document. Cet élément n'a pas d'attributs. Son contenu est une séquence de zéro, un ou plusieurs éléments <list>, dont chacun définit une seule liste de ressources.

Chaque élément <list> peut contenir un attribut "name" facultatif. Cet attribut est une bride pour la liste. Lorsque il est présent, il DOIT être unique parmi tous les autres éléments <list> au sein du même élément parent. L'élément <list> peut aussi contenir des attributs provenant d'autres espaces de noms, pour les besoins de l'extensibilité.

Chaque élément <list> est composé d'un nom d'affichage facultatif, d'une séquence de zéro, un ou plusieurs éléments, dont chacun peut être un élément <entry>, un élément <list>, un élément <entry-ref>, ou un élément <external>, suivi par tout nombre d'éléments provenant d'autres espaces de noms, pour les besoins d'extensibilité. La capacité d'un élément <list> à contenir d'autres éléments <list> signifie qu'une liste de ressources peut être structurée hiérarchiquement. Le <display-name> (*nom d'affichage*) permet alors qu'un nom lisible par l'homme soit associé à chaque niveau de la hiérarchie. Un élément <entry> décrit une seule ressource, définie par un URI, qui fait partie de la liste. Un élément <entry-ref> permet d'inclure une entrée dans un document au sein de la même racine XCAP par référence, plutôt que par une valeur. Un élément <external> contient une référence à une liste mémorisée dans ce serveur ou un autre.

L'élément <entry> décrit une seule ressource. L'élément <entry> a un seul attribut obligatoire, "uri". Cet attribut est égal à l'URI qui est utilisé pour accéder à la ressource. Le format de liste de ressources lui-même n'impose pas de contrainte sur le type d'URI qui peut être utilisé. Cependant, le service qui utilise la liste de ressources peut exiger des schémas d'URI spécifiques. Par exemple, les services de RLS vont exiger des URI qui représentent des ressources auxquelles on peut s'abonner. Cela inclut les URI SIP et pres [RFC3859]. L'attribut "uri" DOIT être unique parmi tous les autres attributs "uri" dans les éléments <entry> au sein du même parent. L'unicité est déterminée par des comparaisons de chaîne sensibles à la casse. À ce titre, il est possible que deux attributs "uri" aient le même URI lorsque ils sont comparés en utilisant les règles d'égalité fonctionnelle définies pour ce schéma d'URI, mais différents quand ils sont comparés en utilisant une comparaison de chaîne sensible à la casse. L'élément <entry> peut aussi contenir des attributs provenant d'autres espaces de noms pour les besoins d'extensibilité.

L'élément <entry> contient une séquence d'éléments qui donnent des informations sur l'entrée. Un seul élément de cette sorte est défini pour l'instant, qui est <display-name>. Cet élément donne une chaîne codée en UTF-8, destinée à l'usage de l'utilisateur humain, qui décrit la ressource. À la différence de l'attribut "name" de l'élément <entry>, le <display-name> n'a pas d'exigence d'unicité. L'élément <display-name> peut contenir l'attribut "xml:lang", qui donne le langage du nom affiché. L'élément <entry> peut contenir d'autres éléments provenant d'autres espaces de noms. Ceci est destiné à prendre en charge l'inclusion d'autres informations sur l'entrée, comme un numéro de téléphone ou une adresse postale.

L'élément <entry-ref> permet d'inclure une entrée dans la liste par référence, plutôt que par une valeur. Cet élément n'est significatif que quand le document a été obtenu par XCAP. Dans ce cas, l'entrée référencée doit exister dans la même racine XCAP. L'élément <entry-ref> a un seul attribut obligatoire, "ref". L'attribut "ref" DOIT être unique parmi tous les autres attributs "ref" dans les éléments <entry-ref> au sein du même parent. L'unicité est déterminée par une comparaison de chaîne sensible à la casse. L'élément <entry-ref> permet aussi des attributs provenant d'autres espaces de noms, pour les besoins d'extensibilité. Le contenu d'un élément <entry-ref> est un nom d'affichage facultatif, suivi par un nombre

quelconque d'éléments provenant d'autres espaces de noms, pour les besoins d'extensibilité. Le nom d'affichage est utile pour fournir un surnom local comme solution de remplacement au nom défini dans l'entrée à laquelle se réfère <entry-ref>.

Le contenu de l'attribut "ref" est un URI relatif HTTP [RFC3986]. Précisément, il DOIT être une référence relative de chemin, où l'URI de base est égal à l'URI racine XCAP du document dans lequel le <entry-ref> apparaît. Cet URI relatif, si il est résolu en un URI absolu en accord avec les procédures de la RFC 3986, DOIT se résoudre en un élément <entry> au sein d'un document de listes de ressources. Par exemple, supposons qu'un élément <entry> au sein d'une racine XCAP spécifique a été identifié par l'URI HTTP suivant :

```
http://xcap.exemple.com/resource-lists/users/sip:bill@exemple.com/
  index/~~/resource-lists/list%5b@name=%22list1%22%5d/entry%5b@uri=%22sip:petri@exemple.com%22%5d
```

Si http://xcap.exemple.com est l'URI racine XCAP, alors un élément <entry-ref> pointant sur cette entrée va avoir la forme suivante :

```
<entry-ref ref="resource-lists/users/sip:bill@exemple.com/
  index/~~/resource-lists/list%5b@name=%22list1%22%5d/entry%5b@uri=%22sip:petri@exemple.com%22%5d"/>
```

Noter que la coupure de ligne dans l'URI HTTP et l'attribut XML ci-dessus est seulement à des fins de lisibilité. Noter aussi que, comme décrit dans la RFC 3986, l'URI de chemin relatif ne commence pas par le caractère "/". Comme l'URI relatif utilisé dans l'attribut "ref" doit être un URI de chemin relatif, le "/" ne va jamais être présent comme premier caractère dans le contenu d'un attribut "ref". Comme le contenu de l'attribut "ref" est un URI HTTP valide, il doit être codé en pourcentage dans le document XML.

L'élément <external> est similaire à l'élément <entry-ref>. Comme <entry-ref>, il n'est significatif que dans les documents obtenus d'un serveur XCAP. Lui aussi est une référence à un contenu mémorisé ailleurs. Cependant, il se réfère à une liste entière, et de plus, il permet à cette liste d'être présente sur un autre serveur. L'élément <external> a un seul attribut obligatoire, "anchor", qui spécifie la liste externe au moyen d'un URI HTTP absolu. L'attribut "anchor" DOIT être unique parmi tous les autres attributs "anchor" dans les éléments <external> au sein du même parent. L'unicité est déterminée par des comparaisons de chaîne sensibles à la casse. L'élément <external> peut aussi contenir des attributs provenant d'autres espaces de noms, pour les besoins d'extensibilité. Le contenu d'un élément <external> est un <display-name> facultatif suivi par un nombre quelconque d'éléments provenant d'un autre espace de noms, pour les besoins d'extensibilité. La valeur de l'attribut "anchor" DOIT être un URI HTTP absolu. Cet URI DOIT identifier une ressource XCAP, et en particulier, il DOIT représenter un élément <list> au sein d'un document de listes de ressources. L'URI DOIT être codé en pourcentage.

Pour les deux éléments <entry-ref> et <external>, la responsabilité de la résolution de leurs références revient à l'entité qui utilise le document. Quand le document est utilisé en conjonction avec XCAP, cela signifie que cette charge revient au client XCAP. Si le client XCAP est une application d'un ordinateur personnel qui utilise le document de listes de ressources comme une liste de présence, les références vont probablement être résolues sur une demande explicite de l'utilisateur. Elles peuvent, bien sûr, être résolues à tout moment. Si le client XCAP est lui-même un RLS, les références vont être résolues quand le RLS reçoit une demande SUBSCRIBE pour un service RLS associé à cette liste de ressources qui contient une de ces références (voir ci-dessous). Un serveur XCAP défini par la présente spécification ne va pas tenter de résoudre les références avant de retourner le document au client. De même, si, du fait d'erreurs du réseau ou de quelque autre problème, les références ne peuvent pas être résolues, le traitement est spécifique de l'usage du document. Pour les listes de ressources utilisées par les services de RLS, le traitement est discuté ci-dessous.

3.2 Schéma

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:resource-lists"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:resource-lists"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
<xs:import namespace="http://www.w3.org/XML/1998/namespace"
  schemaLocation="http://www.w3.org/2001/xml.xsd"/>
<xs:complexType name="listType">
  <xs:sequence>
    <xs:element name="display-name" type="display-nameType"
      minOccurs="0"/>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
```

```

<xs:choice>
  <xs:element name="list">
    <xs:complexType>
      <xs:complexContent>
        <xs:extension base="listType"/>
      </xs:complexContent>
    </xs:complexType>
  </xs:element>
  <xs:element name="external" type="externalType"/>
  <xs:element name="entry" type="entryType"/>
  <xs:element name="entry-ref" type="entry-refType"/>
</xs:choice>
</xs:sequence>
<xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="name" type="xs:string" use="optional"/>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="entryType">
  <xs:sequence>
    <xs:element name="display-name" minOccurs="0">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="display-nameType"/>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="uri" type="xs:anyURI" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>

<xs:complexType name="entry-refType">
  <xs:sequence>
    <xs:element name="display-name" type="display-nameType" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="ref" type="xs:anyURI" use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:complexType name="externalType">
  <xs:sequence>
    <xs:element name="display-name" type="display-nameType" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="anchor" type="xs:anyURI"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<xs:element name="ressource-lists">
  <xs:complexType>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="list" type="listType"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:complexType name="display-nameType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute ref="xml:lang"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

```

```

</xs:simpleContent>
</xs:complexType>
</xs:schema>

```

3.3 Exemple de document

Voici un exemple d'un document conforme au schéma. Tous les retours à la ligne dans le contenu des éléments sont seulement à des fins d'affichage.

```

<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<list name="amis">
<entry uri="sip:bill@exemple.com">
<display-name>Bill Doe</display-name>
</entry>

<entry-ref ref="resource-lists/users/sip:bill@exemple.com/index/~~/
resource-lists/list%5b@name=%22list1%22%5d/entry%5b@uri=%22sip:petri@exemple.com%22%5d"/>
<list name="amis-proches">
<display-name>Amis proches</display-name>
<entry uri="sip:joe@exemple.com">
<display-name>Joe Smith</display-name>
</entry>
<entry uri="sip:nancy@exemple.com">
<display-name>Nancy Gross</display-name>
</entry>
<external anchor="http://xcap.exemple.org/resource-lists/users/
sip:a@exemple.org/index/~~/resource-lists/list%5b@name=%22mkti
ng%22%5d">
<display-name>Marketing</display-name>
</external>
</list>
</list>
</resource-lists>

```

3.4 Usage avec XCAP

Les documents de listes de ressources peuvent être manipulés avec XCAP. Ce paragraphe donne les détails nécessaires pour un tel usage.

3.4.1 Identifiant unique d'application

XCAP exige des usages d'application qu'ils définissent un identifiant unique d'application (AUID, *Application Unique ID*) dans l'arborescence de l'IETF ou dans une arborescence de fabricant. La présente spécification définit l'AUID "resource-lists" au sein de l'arborescence de l'IETF, via l'enregistrement de l'IANA à la Section 8.

3.4.2 Type MIME

Le type MIME pour ce document est "application/resource-lists+xml".

3.4.3 SchémaXML

Le schéma XML pour ce document est défini comme seul contenu du paragraphe 3.2.

3.4.4 Espace de noms par défaut

L'espace de noms par défaut utilisé dans l'expansion des URI est urn:ietf:params:xml:ns:resource-lists.

3.4.5 Contraintes supplémentaires

En plus du schéma, il y a des contraintes sur les valeurs présentes dans l'attribut "name" de l'élément <list>, l'attribut "uri" de l'élément <external>, l'attribut "ref" du *n*ième élément <entry-ref>, et l'attribut "anchor" de l'élément <external>. Ces contraintes sont définies au paragraphe 3.1. Certaines de ces contraintes sont appliquées par le serveur XCAP. Ce sont :

- o L'attribut "name" dans un élément <list> DOIT être unique parmi tous les autres attributs "name" des éléments <list> au sein du même élément parent. L'unicité est déterminée par une comparaison de chaîne sensible à la casse.
- o L'attribut "uri" dans un élément <entry> DOIT être unique parmi tous les autres attributs "uri" des éléments <entry> au sein du même élément parent. L'unicité est déterminée par une comparaison de chaîne sensible à la casse.
- o L'URI dans l'attribut "ref" de l'élément <entry-ref> DOIT être unique parmi tous les autres attributs "ref" des éléments <entry-ref> au sein du même élément parent. L'unicité est déterminée par une comparaison de chaîne sensible à la casse. La valeur de l'attribut DOIT être une référence de chemin relative. Noter que le serveur n'est pas chargé de vérifier que la référence se résout en un élément <entry> dans un document au sein de la même racine XCAP.
- o L'URI dans l'attribut "anchor" de l'élément <external> DOIT être unique parmi tous les autres attributs "anchor" des éléments <external> au sein du même élément parent. L'unicité est déterminée par une comparaison de chaîne sensible à la casse. La valeur de l'attribut DOIT être un URI HTTP absolu. Noter que le serveur n'est pas chargé de vérifier que la référence se résout en un élément <list> dans un document. Bien sûr, comme l'URI peut référencer un serveur dans un autre domaine, l'intégrité référentielle ne peut pas être garantie sans ajouter une substantielle complexité au système.

3.4.6 Sémantique des données

La sémantique pour le contenu de document est fournie au paragraphe 3.1.

3.4.7 Conventions de dénomination

Les documents de listes de ressources sont généralement identifiés comme des références provenant d'autres usages d'application. Par exemple, un document de services de RLS contient une référence à la liste de ressource qu'il utilise.

Fréquemment, un client XCAP va souhaiter insérer ou supprimer un élément <entry>, <entry-ref>, ou <external> d'un document sans avoir une copie en antémémoire de ce document. Dans ce cas, l'attribut "uri" de l'élément <entry>, l'attribut "ref" de l'élément <entry-ref>, ou l'attribut "anchor" de l'élément <external> est utilisé comme un indice pour sélectionner l'élément sur lequel opérer. Le serveur XCAP va déterminer l'unicité par une comparaison de chaîne sensible à la casse. Cependant, chacun de ces attributs contient des URI, et les règles d'égalité d'URI pour leurs schémas peuvent permettre que deux URI soient les mêmes, même si ils sont différents par une comparaison de chaîne sensible à la casse. À ce titre, il est possible qu'un client tente un PUT ou DELETE pour essayer de modifier ou supprimer un élément existant. À la place, le PUT finit par insérer un nouvel élément, ou le DELETE finit par retourner une réponse d'erreur.

Si le client XCAP ne peut pas déterminer si l'intention de l'utilisateur est de créer ou remplacer, le client DEVRAIT canoniser l'URI avant d'effectuer l'opération. Pour un URI SIP (souvent présent dans l'attribut "uri" de l'élément <entry>) cette procédure de canonisation est définie à la Section 5. On s'attend à ce que les URI SIP soient placés dans des documents de listes de ressources qui seront généralement de la forme sip:usager@domaine, et incluent éventuellement un paramètre d'utilisateur. Les règles de canonisation fonctionnent parfaitement pour ces URI.

Pour les URI HTTP, un algorithme de base de canonisation est comme suit. Si l'accès dans l'URI est égal à l'accès par défaut (80 pour les URI http) alors d'accès est supprimé. Le nom d'hôte est converti en minuscules. Tout codage en pourcentage dans l'URI pour des caractères qui n'ont pas besoin d'être codés en pourcentage est supprimé. Un caractère a besoin d'être codé en pourcentage quand il n'est pas permis dans cette partie de l'URI sur la base de la grammaire pour cette partie de l'URI.

3.4.8 Interdépendances de ressources

Aucune interdépendance de ressources n'est identifiée par cet usage d'application.

3.4.9 Politiques d'autorisation

Cet usage d'application ne modifie pas la politique d'autorisation XCAP par défaut, qui est que seul un utilisateur peut lire, écrire, ou modifier ses propres documents. Un serveur peut permettre à des utilisateurs privilégiés de modifier des documents qui ne leur appartiennent pas, mais l'établissement et l'indication de telles politiques sort du domaine d'application du présent document. Il est prévu qu'un futur usage d'application définira à quels utilisateurs il est permis de modifier une ressource de liste.

4. Documents de services de RLS

4.1 Structure

Un document de services de RLS est utilisé pour définir les URI qui représentent les services fournis par un serveur de liste de ressources (RLS, *Resource List Server*) comme défini dans la [RFC4662]. Un document de services de RLS est un document XML [XML] qui DOIT être bien formé et DOIT être valide en accord avec les schémas, incluant les schémas d'extension, disponibles au valideur et applicables au document XML. Les documents de services de RLS DOIVENT être fondés sur XML 1.0 et DOIVENT être codés en utilisant UTF-8. La présente spécification utilise les espaces de noms XML pour identifier les documents de services de RLS et les fragments de document. L'URI d'espace de noms pour les éléments définis par cette spécification est un URN [RFC2141] qui utilise l'identifiant d'espace de noms 'ietf' défini par la [RFC2648] et étendu par la [RFC3688]. Cet URN est : urn:ietf:params:xml:ns:rls-services

L'élément racine d'un document de services de RLS est <rls-services>. Il contient une séquence d'éléments <service>, dont chacun définit un service disponible à un RLS.

Chaque élément <service> a un seul attribut obligatoire, "uri". Cet URI définit les ressources associées au service. C'est-à-dire que si un client s'abonne à cet URI, il va obtenir le service défini par l'élément <service> correspondant. L'élément <service> peut aussi contenir des attributs provenant d'autres espaces de noms, pour les besoins d'extensibilité. L'élément <service> contient des éléments fils qui définissent le service. Pour un service de RLS, très peu de définition de service est nécessaire : juste la liste de ressources auxquelles le serveur va effectuer des abonnements virtuels [RFC4662] et l'ensemble de paquetages d'événements que le service prend en charge. La première peut être portée d'une des deux façons suivantes. Il peut y avoir un élément <resource-list> qui pointe sur un élément <list> dans un document de liste de ressources, ou il peut y avoir un élément <list>, qui inclut directement la liste de ressources. Les paquetages pris en charge sont contenus dans l'élément <packages>. L'élément <service> peut aussi contenir des éléments provenant d'autres espaces de noms, pour les besoins d'extensibilité.

En incluant directement le contenu de la liste de ressources, un utilisateur peut créer des listes et y ajouter des membres avec une seule opération XCAP. Cependant, la liste résultante devient "cachée" au sein de la définition de service de RL, et n'est pas utilisable par d'autres usages d'application. Pour cette raison, l'élément <resource-list> existe comme solution de remplacement. Il peut faire référence à un élément <list> dans un document resource-lists. Comme la liste est séparée de la définition de service, elle peut être facilement réutilisée par d'autres usages d'application.

L'élément <list> est le type de liste défini par le schéma pour les listes de ressources. Il est discuté au paragraphe 3.1.

L'élément <resource-list> contient un URI. Cet élément n'a de signification que quand le document a été obtenu par XCAP. L'URI DOIT être un URI HTTP absolu représentant une ressource d'élément XCAP. Sa racine XCAP DOIT être la même que la racine XCAP du document de services de RLS. Quand le document de services de RLS est présent dans le répertoire d'accueil d'un utilisateur, l'URI HTTP DOIT exister sous le répertoire d'accueil de cet utilisateur dans l'usage d'application resource-lists. Quand le document de services de RLS est dans le répertoire global, l'URI HTTP DOIT exister sous le répertoire d'accueil de tout utilisateur dans l'usage d'application resource-lists. Dans l'un et l'autre cas, l'élément référencé par l'URI DOIT être un élément <list> au sein d'un document resource-lists. Toutes ces contraintes sauf la dernière (qui est une contrainte d'intégrité référentielle) vont être appliquées par le serveur XCAP.

L'élément <packages> contient une séquence d'éléments <package>. Le contenu de chaque élément <package> est le nom d'un paquetage d'événement SIP [RFC3265]. L'élément <packages> peut aussi contenir des éléments provenant d'espaces de noms supplémentaires, pour les besoins d'extensibilité. L'élément <packages> est facultatif. Quand il n'est pas présent, cela signifie que le service de RLS va accepter des abonnements pour tout paquetage d'événements.

4.2 Schéma

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:rls-services"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:rls-services"
  xmlns:rl="urn:ietf:params:xml:ns:resource-lists"
  elementFormDefault="qualified" attributeFormDefault="unqualified">
  <xs:element name="rls-services">
    <xs:complexType>
      <xs:sequence minOccurs="0" maxOccurs="unbounded">
        <xs:element name="service" type="serviceType"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:complexType name="serviceType">
    <xs:sequence>
      <xs:choice>
        <xs:element name="resource-list" type="xs:anyURI"/>
        <xs:element name="list" type="rl:listType"/>
      </xs:choice>
      <xs:element name="packages" type="packagesType" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="uri" type="xs:anyURI" use="required"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
  <xs:complexType name="packagesType">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="package" type="packageType"/>
      <xs:any namespace="##other" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:simpleType name="packageType">
    <xs:restriction base="xs:string"/>
  </xs:simpleType>
</xs:schema>
```

4.3 Exemple de document

Ce document montre deux services. L'un est sip:mespotes@exemple.com, et l'autre est sip:marketing@exemple.com. Le premier service fait référence à une liste de ressources dans un document resource-lists, et le second inclut une liste locale. Les deux services sont seulement pour le paquetage d'événements de présence.

```
<?xml version="1.0" encoding="UTF-8"?>
<rls-services xmlns="urn:ietf:params:xml:ns:rls-services"
  xmlns:rl="urn:ietf:params:xml:ns:resource-lists"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <service uri="sip:mespotes@exemple.com">
    <resource-list>http://xcap.exemple.com/resource-lists/user
      /sip:joe@exemple.com/index/~~/resource-lists/list%5b@name=%2211%22%5d</resource-list>
    <packages>
      <package>presence</package>
    </packages>
  </service>
  <service uri="sip:marketing@exemple.com">
    <list name="marketing">
```

```

  <rl:entry uri="sip:joe@exemple.com"/>
  <rl:entry uri="sip:sudhir@exemple.com"/>
</list>
<packages>
  <package>presence</package>
</packages>
</service>
</rls-services>

```

4.4 Usage avec XCAP

Les documents de services de RLS peuvent être manipulés avec XCAP. Cette section donne les détails nécessaires pour un tel usage.

4.4.1 Identifiant unique d'application

XCAP exige des usages d'application qu'ils définissent un identifiant unique d'application (AUID) dans l'arborescence de l'IETF ou dans celle d'un fabricant. La présente spécification définit l'AUID "rls-services" au sein de l'arborescence de l'IETF, via l'enregistrement de l'IANA à la Section 8.

4.4.2 Type MIME

Le type MIME pour ce document est "application/rls-services+xml".

4.4.3 Schéma XML

Le schéma XML pour ce document est défini comme le seul contenu du paragraphe 4.2.

4.4.4 Espace de noms par défaut

L'espace de noms par défaut utilisé dans l'expansion des URI est urn:ietf:params:xml:ns:rls-services.

4.4.5 Contraintes supplémentaires

En plus du schéma, il y a des contraintes sur les URI présents dans les éléments <service> et <resource-list>. Ces contraintes sont définies au paragraphe 3.1. Certaines de ces contraintes sont appliquées par le serveur XCAP. Ces contraintes sont :

- o L'URI dans l'attribut "uri" de l'élément <service> DOIT être unique parmi tous les autres URI dans les éléments "uri" dans tout élément <service> dans tout document sur un serveur particulier. Cette contrainte d'unicité s'étend à travers les racines XCAP. De plus, l'URI NE DOIT PAS correspondre à une ressource existante au sein du domaine de l'URI. Si il est demandé à un serveur d'établir l'URI à quelque chose qui existe déjà, le serveur DOIT rejeter la demande avec un code 409, et utiliser les mécanismes définis dans la [RFC4825] pour suggérer d'autres URI de remplacement qui n'ont pas encore été alloués.
- o L'URI dans un élément <resource-list> DOIT être un URI absolu. Le serveur DOIT vérifier que le chemin d'URI contient "resource-lists" dans le segment de chemin correspondant à l'AUID. Si le document de services de RLS est dans l'arborescence d'utilisateur XCAP (par opposition à l'arborescence globale) le serveur DOIT vérifier que le XUI dans le chemin est le même que le XUI dans l'URI pour le document de services de RLS. Ces vérifications sont faites en examinant la valeur de l'URI, par opposition au déréférencement de l'URI. Le serveur n'est pas chargé de vérifier que l'URI pointe réellement sur un élément <list> dans un document de listes de ressources valide.
- o De plus, un document de services de RLS peut contenir un élément <list>, qui à son tour peut contenir des éléments <entry>, <entry-ref>, <list>, et <external>. Les contraintes définies pour ces éléments au paragraphe 3.4.7 DOIVENT être appliquées.
- o Dans certains cas, un client XCAP va souhaiter créer un nouveau service de RLS, et souhaiter lui allouer un "URI de

complaisance", comme sip:amis@exemple.com. Cependant, le client ne sait pas si cet URI satisfait aux contraintes d'unicité définies ci-dessus. Dans ce cas, il peut simplement tenter l'opération de création, et si le résultat est un 409 qui contient un rapport de conflit détaillé avec l'élément <échec d'unicité>, le client saura que l'URI n'a pas pu être alloué. Il peut alors réessayer avec un URI de complaisance différent, ou utiliser une des suggestions du rapport de conflit détaillé.

- o Si le client souhaite créer un nouveau service de RLS, et si il ne se soucie pas de ce qu'est l'URI, il va en créer un au hasard, et tenter l'opération de création. Comme discuté dans la [RFC4825], si cela devait échouer avec un conflit d'unicité, le client peut réessayer avec des URI différents d'un aléa croissant.

4.4.6 Sémantique des données

La sémantique du contenu de document est fournie au paragraphe 4.1.

4.4.7 Conventions de dénomination

Normalement, il y a deux clients XCAP distincts qui accèdent aux documents de services de RLS. Le premier est un client agissant au nom de l'utilisateur d'extrémité dans le système. Ce client édite et écrit les listes de ressources et les documents de services de RLS lorsque ils sont créés ou modifiés par l'utilisateur d'extrémité. L'autre client XCAP est le RLS lui-même, qui lit les documents de services de RLS afin de traiter les demandes SUBSCRIBE.

Pour rendre plus facile à un RLS de trouver l'élément <service> pour un URI particulier, le serveur XCAP tient, dans l'arborescence globale, un seul document de services de RLS représentant l'union de tous les éléments <service> à travers tous les documents créés par tous les utilisateurs au sein de la même racine XCAP. Il y a une seule instance de ce document, et son nom est "index". Donc, si l'URI de services racine est http://xcap.exemple.com, l'URI qu'un RLS va utiliser pour aller chercher cet index est :http://xcap.exemple.com/rls-services/global/index

Comme expliqué ci-dessous, cet index est créé à partir de tous les documents dans l'arborescence de l'utilisateur qui ont aussi le nom "index". Une implication de cela est qu'un client qui opère au nom d'un utilisateur DEVRAIT définir ses services de RLS au sein du document nommé "index". Si l'URI de services racine est http://xcap.exemple.com, pour l'utilisateur "sip:joe@exemple.com" l'URI pour ce document serait :

http://xcap.exemple.com/rls-services/users/sip:joe@exemple.com/index

Si un client choisit de définir des services de RLS dans un document différent, ce document ne sera pas "pris" dans l'index global, et donc, ne sera pas utilisé comme un service de RLS.

4.4.8 Interdépendances de ressources

Comme avec les autres usages d'application, les conventions de dénomination de schéma XML et de ressource XCAP décrivent la plupart des interdépendances de ressources applicables à cet usage d'application.

Cet usage d'application définit une interdépendance de ressources supplémentaire entre un seul document dans l'arborescence globale et tous les documents dans l'arborescence de l'utilisateur avec le nom "index". Ce document global est formé par l'union de tous les documents index pour tous les utilisateurs au sein de la même racine XCAP. Dans ce cas, l'opération d'union implique que chaque élément <service> dans un document d'utilisateur va aussi être présent comme élément <service> dans le document global. L'inverse est aussi vrai. Chaque élément <service> dans le document global existe dans un document d'utilisateur au sein de la même racine XCAP.

Par exemple, considérons le document de services de RLS pour l'utilisateur sip:joe@exemple.com :

```
<?xml version="1.0" encoding="UTF-8"?>
<rls-services>
  <service uri="sip:mespotes@exemple.com">
    <resource-list>http://xcap.exemple.com/resource-lists/users/sip:joe@exemple.com/index/~~/resource-lists/list
    %5b@name=%221%22%5d</ressource-list>
    <packages>
      <package>presence</package>
    </packages>
```

```
</service>
</rls-services>
```

Et considérons le document de services de RLS pour l'utilisateur bob :

```
<?xml version="1.0" encoding="UTF-8"?>
<rls-services>
  <service uri="sip:marketing@exemple.com">
    <list name="marketing">
      <rl:entry uri="sip:joe@exemple.com"/>
      <rl:entry uri="sip:sudhir@exemple.com"/>
    </list>
    <packages>
      <package>presence</package>
    </packages>
  </service>
</rls-services>
```

Le document global à <http://xcap.exemple.com/rls-services/global/index> ressemblerait à ceci :

```
<?xml version="1.0" encoding="UTF-8"?>
<rls-services xmlns="urn:ietf:params:xml:ns:rls-services"
  xmlns:rl="urn:ietf:params:xml:ns:resource-lists"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <service uri="sip:mespotes@exemple.com">
    <resource-list>http://xcap.exemple.com/resource-lists/users/sip:joe@exemple.com/index/~~/resource-lists/list%5b@name=%221%22%5d</resource-list>
    <packages>
      <package>presence</package>
    </packages>
  </service>
  <service uri="sip:marketing@exemple.com">
    <list name="marketing">
      <rl:entry uri="sip:joe@exemple.com"/>
      <rl:entry uri="sip:sudhir@exemple.com"/>
    </list>
    <packages>
      <package>presence</package>
    </packages>
  </service>
</rls-services>
```

Les demandes faites sur le document global DOIVENT générer des réponses qui reflètent l'état le plus récent de tous les documents d'utilisateur pertinents. Cette exigence n'implique pas que le serveur doive réellement mémoriser ce document global. Il est prévu que la plupart des systèmes vont construire dynamiquement les réponses à toute demande particulière sur la ressource de document.

La contrainte d'unicité sur l'attribut "uri" de <service> va assurer que deux éléments <service> dans le document global n'ont pas la même valeur de cet attribut.

4.4.9 Politiques d'autorisation

Cet usage d'application ne modifie pas la politique d'autorisation par défaut de XCAP, qui est que seul un utilisateur peut lire, écrire, ou modifier ses propres documents. Un serveur peut permettre à des utilisateurs privilégiés de modifier des documents dont ils ne sont pas propriétaires, mais l'établissement et l'indication de telles politiques sort du domaine d'application du présent document. Il est prévu qu'un futur usage d'application définira quels utilisateurs ont la permission de modifier un document de services de RLS.

Le document d'index conservé dans l'arborescence globale représente des informations sensibles, car il contient l'union de toutes les informations pour tous les utilisateurs sur le serveur. À ce titre, son accès DOIT être restreint aux éléments de

confiance au sein du domaine du serveur. Normalement, cela va être limité aux RLS qui ont besoin d'accéder à ce document.

4.5 Usage d'un document de services de RLS par un RLS

Ce paragraphe explique comment un RLS, à réception d'une demande SUBSCRIBE, utilise XCAP et le document de services de RLS pour guider son fonctionnement.

Quand un RLS reçoit une demande SUBSCRIBE pour un URI (présent dans l'URI de demande) il obtient l'élément <service> dont l'attribut uri correspond (sur la base d'une égalité d'URI) à l'URI dans la demande SUBSCRIBE. Ce document ne fait pas de déclaration normative sur la façon dont cela peut être accompli. On décrit ci-après une approche possible.

Le RLS canonise l'URI de demande comme décrit à la Section 5. Il effectue ensuite une opération GET XCAP à l'égard de l'URI formé par la combinaison de la racine XCAP avec le sélecteur de document de l'index global avec un sélecteur de nœud de la forme "rls-services/service[@uri=<canonical-uri>]", où <canonical-uri> est la version canonisée de l'URI de demande. Si la réponse est un 200 OK, elle va contenir la définition de service pour cet URI.

Une fois que l'élément <service> a été obtenu, il est examiné. Si l'élément <packages> est présent, et si le paquetage d'événement dans la demande SUBSCRIBE n'est pas parmi ceux de la liste des éléments <package> au sein de <packages>, la demande DOIT être rejetée avec un code de réponse 489 (Mauvais événement) comme décrit dans la [RFC3265]. Autrement, elle DEVRAIT être traitée. L'étape suivante est d'autoriser que le client souscrive à la ressource. Cela peut être fait, par exemple, en utilisant les données définies dans la [RFC5025]. En supposant que le souscripteur est autorisé à s'abonner à cette ressource, la souscription est traitée conformément aux procédures définies dans la [RFC4662]. Ce traitement exige que le RLS calcule une liste simple d'URI auxquels s'abonner. Si l'élément <service> avait un élément <list>, il est extrait. Si l'élément <service> avait un élément <resource-list>, son contenu d'URI est déréférencé. Le résultat devrait être un élément <list>. Si il ne l'est pas, la demande DEVRAIT être rejetée avec un code 502 (Mauvaise passerelle). Autrement, cet élément <list> est extrait.

À ce point, le RLS a un élément <list> en sa possession. L'étape suivante est d'obtenir une liste d'URI à partir de cet élément. Pour ce faire, il traverse l'arborescence des éléments dont la racine est l'élément <list>. Avant de commencer la traversée, le RLS initialise deux listes : la "liste simple", qui va contenir la liste d'URI après la traversée, et la "liste de traversée", qui contient la liste des URI HTTP dans les éléments <external> qui ont déjà été visités. Les deux listes sont initialement vides. Ensuite, la traversée de l'arborescence commence. Un serveur peut utiliser tout ordre de traversée d'arborescence qu'il veut, comme une recherche en profondeur d'abord, ou une recherche en largeur d'abord. Le traitement à chaque élément de l'arborescence dépend du nom de l'élément :

- o Si l'élément est <entry>, l'URI dans l'attribut "uri" de l'élément est ajouté à la liste simple si il n'y est déjà présent (sur la base d'une égalité de chaîne sensible à la casse) et si le schéma d'URI représenté peut être utilisé pour servir les souscriptions, comme SIP [RFC3261] et présence [RFC3859].
- o Si l'élément est <entry-ref>, la référence de chemin relative constituant la valeur de l'attribut "ref" est résolue dans un URI absolu. Ceci est fait en utilisant les procédures définies au paragraphe 5.2 de la [RFC3986], en utilisant la racine XCAP du document de services de RLS comme URI de base. Cet URI absolu est résolu. Si le résultat n'est pas un 200 OK contenant un élément <entry>, la demande SUBSCRIBE DEVRAIT être rejetée avec un code 502 (Mauvaise passerelle). Autrement, l'élément <entry> retourné est traité comme décrit dans les étapes précédentes.
- o Si l'élément est un élément <external>, l'URI absolu qui constitue la valeur de l'attribut "anchor" de l'élément est examiné. Si l'URI est sur la liste traversée, le serveur DOIT cesser de traverser l'arborescence, et DEVRAIT rejeter la demande SUBSCRIBE avec un code 502 (Mauvaise passerelle). Si l'URI n'est pas sur la liste traversée, le serveur ajoute l'URI à la liste traversée, et déréférence l'URI. Si le résultat n'est pas un 200 OK contenant un élément <list>, la demande SUBSCRIBE DEVRAIT être rejetée avec un code 502 (Mauvaise passerelle). Autrement, le RLS remplace l'élément <external> dans sa copie locale de l'arborescence par l'élément <list> qui a été retourné, et la traversée de l'arborescence continue.

Parce que l'élément <external> est utilisé pour construire dynamiquement l'arborescence, il y a une possibilité d'évaluation récurrente des références. La liste de traversée est utilisée pour empêcher que cela arrive.

Une fois l'arborescence traversée, le RLS peut créer des abonnements virtuels à chaque URI dans la liste simple, comme défini dans la [RFC4662]. Dans les étapes de traitement mentionnées ci-dessus, quand un élément <entry-ref> ou

<external> contient une référence qui ne peut pas être résolue, faire échouer la demande est au niveau DEVRAIT. Dans certains cas, un RLS peut fournir un meilleur service en créant des abonnements virtuels aux URI dans la liste simple qui pourrait être obtenue, et en omettant ceux qui ne le peuvent pas. C'est seulement dans ces cas que la recommandation DEVRAIT devrait être ignorée.

5. Canonisation d'URI SIP

Cette Section donne une technique pour la canonisation d'URI. Cette canonisation produit un URI qui, dans la plupart des cas, est égal à l'URI original (où l'égalité est fondée sur les règles de comparaison d'URI de la RFC 3261). De plus, l'URI canonisé va généralement être lexicalement équivalent à la version canonisée de tout autre URI égal à l'original.

Pour canoniser l'URI, on suit les étapes suivantes :

1. D'abord, la partie domaine de l'URI est convertie en minuscules, et tous les jetons (comme "user" ou "transport" ou "udp") sont convertis en minuscules.
2. Deuxièmement, tout codage en pourcentage dans l'URI pour les caractères qui n'ont pas besoin d'être codés en pourcentage est supprimé. Un caractère a besoin d'être codé en pourcentage quand il n'est pas permis dans cette partie de l'URI sur la base de la grammaire pour cette partie de l'URI. Par exemple, si un URI SIP est sip:%6a0e%20smith@exemple.com, il est changé en sip:joe%20smith@exemple.com. Dans l'URI original, le caractère "j" était codé en pourcentage. Cela est permis, mais pas exigé, car la grammaire permet qu'un "j" apparaisse dans la partie utilisateur. Par suite, il apparaît comme 'j' après cette étape de canonisation.
3. Troisièmement, tous les paramètres d'URI sont réordonnés afin qu'ils apparaissent en ordre lexical sur la base du nom de paramètre. La remise en ordre d'un caractère est déterminée par la valeur numérique en US-ASCII de ce caractère, les plus petits nombres venant en premier. Les paramètres sont ordonnés avec le caractère le plus à gauche comme celui de plus fort poids. Pour les paramètres qui contiennent seulement des lettres, ceci est équivalent à un ordre alphabétique.
4. Finalement, tous les paramètres d'en-tête sont éliminés. Cet URI canonisé est utilisé à la place de l'URI original.

Si deux URI, A et B, sont fonctionnellement égaux (ce qui signifie qu'ils sont égaux selon les règles de comparaison d'URI de la RFC 3261) leurs URI canonisés sont égaux dans une comparaison de chaîne sensible à la casse si ce qui suit est vrai :

- o aucun des deux URI ne contient de paramètre d'en-tête,
- o si un des URI contient un paramètre d'URI non défini dans la RFC 3261, l'autre le fait aussi.

6. Extensibilité

Les documents resource-lists et de services de RLS sont destinés à être étendus. Une extension a lieu en définissant un nouvel ensemble d'éléments dans un nouvel espace de noms, gouverné par un nouveau schéma. Chaque extension DOIT avoir un espace de noms XML approprié qui lui est alloué. L'espace de noms XML de l'extension DOIT être différent de l'espace défini dans cette spécification. L'extension NE DOIT PAS changer la syntaxe ou la sémantique des schémas définis dans le présent document. Toutes les étiquettes et les attributs XML qui font partie de l'extension DOIVENT être qualifiés de façon appropriée afin de les placer dans cet espace de noms.

La présente spécification définit des endroits explicites où les nouveaux éléments ou attributs d'une extension peuvent être placés. Ils sont explicitement indiqués dans les schémas par les éléments <any> et <anyAttribute>. Les extensions à la présente spécification DOIVENT spécifier où leurs éléments peuvent être placés dans le document.

Par suite, un document qui contient des extensions va exiger plusieurs schémas pour déterminer sa validité : un schéma défini dans le présent document, ainsi que ceux définis par les extensions présentes dans le document. Parce que les extensions se produisent par l'ajout de nouveaux éléments et attributs gouvernés par les nouveaux schémas, les schémas définis dans ce document sont fixés et ne seront changés que par une révision de cette spécification. Une telle révision, si elle devait avoir lieu, entreprendrait de permettre aux documents conformes au précédent schéma de rester conformes au nouveau. Par suite, les schémas définis ici ne fournissent pas de versions de schéma explicites, car il n'est pas prévu que ce sera nécessaire.

7. Considérations sur la sécurité

Les informations contenues dans les documents rls-services et resource-lists sont particulièrement sensibles. Elles représentent le principal ensemble de personnes avec qui un utilisateur veut communiquer. Par suite, les clients DEVRAIENT utiliser TLS quand ils contactent les serveurs afin d'aller chercher ces informations. Noter que cela ne représente pas un changement de la force des exigences par rapport à XCAP.

8. Considérations relatives à l'IANA

Plusieurs considérations relatives à l'IANA sont associées à la présente spécification.

8.1 Identifiants unique d'application XCAP

Cette section enregistre deux nouveaux identifiants uniques d'application XCAP (AUID) conformément aux procédures de l'IANA définies dans la [RFC4825].

8.1.1 resource-lists

Nom de l'AUID : resource-lists

Description : une application de listes de ressources est toute application qui a besoin d'accéder à une liste de ressources, identifiées par un URI, auxquelles des opérations, comme des abonnements, peuvent être appliquées.

8.1.2 rls-services

Nom de l'AUID : rls-services

Description : une application de services de serveur de liste de ressources (RLS, *Resource List Server*) est une application du protocole d'initiation de session (SIP, *Session Initiation Protocol*) par laquelle un serveur reçoit des demandes SIP SUBSCRIBE pour une ressource, et génère des abonnements à une liste de ressources.

8.2 Enregistrements de Type MIME

Cette spécification demande l'enregistrement de deux nouveaux types MIME conformément aux procédures de la [RFC4288] et aux lignes directrices de la [RFC3023].

8.2.1 application/resource-lists+xml

Nom de type de support MIME : application

Nom de sous type MIME : resource-lists+xml

Paramètres obligatoires : aucun

Paramètres facultatifs : les mêmes que le paramètre de jeu de caractère application/xml comme spécifié dans la [RFC3023].

Considérations de codage : les mêmes considérations de codage application/xml comme spécifié dans la [RFC3023].

Considérations sur la sécurité : voir la Section 10 de la [RFC3023] et la Section 7 de la RFC 4826.

Considérations d'interopérabilité : aucune

Spécification publiée : RFC 4826

Applications qui utilisent ce type de support : ce type de document a été utilisé pour prendre en charge l'abonnement à des listes d'utilisateurs [RFC4662] pour la présence fondée sur SIP [RFC3856].

Informations supplémentaires :

Numéro magique : aucun

Extension de fichier : .rl

Code de type de fichier Macintosh : "TEXT"

Adresse personnelle et de messagerie pour plus d'informations : Jonathan Rosenberg, jdrosen@jdrosen.net

Usage prévu : COURANT

Auteur/contrôleur des changements : l'IETF.

8.2.2 application/rls-services+xml

Nom de type de support MIME : application

Nom de sous type MIME : rls-services+xml

Paramètres obligatoires : aucun

Paramètres facultatifs : les mêmes que le paramètre de jeu de caractère application/xml comme spécifié dans la [RFC3023].

Considérations de codage : les mêmes considérations de codage application/xml comme spécifié dans la [RFC3023].

Considérations sur la sécurité : voir la Section 10 de la [RFC3023] et la Section 7 de la RFC 4826.

Considérations d'interopérabilité : aucune

Spécification publiée : RFC 4826

Applications qui utilisent ce type de support : ce type de document a été utilisé pour prendre en charge l'abonnement à des listes d'utilisateurs [RFC4662] pour la présence fondée sur SIP [RFC3856].

Informations supplémentaires :

Numéro magique : aucun

Extension de fichier : .rs

Code de type de fichier Macintosh : "TEXT"

Adresse personnelle et de messagerie pour plus d'informations : Jonathan Rosenberg, jdrosen@jdrosen.net

Usage prévu : COURANT

Auteur/contrôleur des changements : l'IETF.

8.3 Enregistrements de sous espace de noms d'URN

Ce paragraphe enregistre deux nouveaux espaces de noms XML, conformément aux lignes directrices de la [RFC3688].

8.3.1 urn:ietf:params:xml:ns:resource-lists

URI : l'URI pour cet espace de noms est urn:ietf:params:xml:ns:resource-lists.

Contact d'enregistrement : IETF, groupe de travail SIMPLE, (simple@ietf.org), Jonathan Rosenberg (jdrosen@jdrosen.net).

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>Resource Lists Namespace</title>
```

```
</head>
```

```
<body>
```

```
<h1>Namespace for Resource Lists</h1>
```

```
<h2>urn:ietf:params:xml:ns:resource-lists</h2>
```

```
<p>See <a href="http://www.rfc-editor.org/rfc/rfc4826.txt">
```

```
RFC4826</a>.</p>
```

```
</body>
```

```
</html>
```

FIN

8.3.2 urn:ietf:params:xml:ns:rls-services

URI : l'URI pour cet espace de noms est urn:ietf:params:xml:ns:rls-services.

Contact d'enregistrement : IETF, groupe de travail SIMPLE, (simple@ietf.org), Jonathan Rosenberg (jdrosen@jdrosen.net).

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>Resource List Server (RLS) Services Namespace</title>
</head>
<body>
  <h1>Namespace for Resource List Server (RLS) Services</h1>
  <h2>urn:ietf:params:xml:ns:rls-services</h2>
  <p>Voir <a href="http://www.rfc-editor.org/rfc/rfc4826.txt">
    RFC4826</a>.</p>
</body>
</html>
FIN

```

8.4 Enregistrements de schémas

Ce paragraphe enregistre deux schémas XML selon les procédures de la [RFC3688].

8.4.1 urn:ietf:params:xml:schema:resource-lists

URI : urn:ietf:params:xml:schema:resource-lists

Contact d'enregistrement : IETF, groupe de travail SIMPLE, (simple@ietf.org), Jonathan Rosenberg (jdrosen@jdrosen.net).

Le XML pour ce schéma est le seul contenu du paragraphe 3.2.

8.4.2 urn:ietf:params:xml:schema:rls-services

URI : urn:ietf:params:xml:schema:rls-services

Contact d'enregistrement : IETF, groupe de travail SIMPLE, (simple@ietf.org), Jonathan Rosenberg (jdrosen@jdrosen.net).

Le XML pour ce schéma est le seul contenu du paragraphe 4.2.

9. Remerciements

Les auteurs tiennent à remercier Hisham Khartabil, Jari Urpalainen, et Spencer Dawkins de leurs commentaires et apports. Merci à Ted Hardie pour ses encouragements et son soutien à ce travail.

10. Références

10.1 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](#))
- [RFC2141] R. Moats, "[Syntaxe des URN](#)", mai 1997. (Obsolète, voir [RFC8141](#))
- [RFC2648] R. Moats, "Espace de nom d'URN pour les documents de l'IETF", août 1999. (Information)
- [RFC3023] M. Murata, S. St.Laurent et D. Kohn, "[Types de support XML](#)", janvier 2001. (Obsolète, voir [RFC7303](#))

- [RFC3261] J. Rosenberg et autres, "SIP : [Protocole d'initialisation de session](#)", juin 2002. (*Mise à jour par [3265](#), [3853](#), [4320](#), [4916](#), [5393](#), [6665](#), [8217](#), [8760](#)*)
- [RFC3688] M. Mealling, "[Registre XML de l'IETF](#)", BCP 81, janvier 2004.
- [RFC3986] T. Berners-Lee, R. Fielding et L. Masinter, "[Identifiant de ressource uniforme](#) (URI) : Syntaxe générique", STD 66, janvier 2005. (*P.S. ; MàJ par RFC8820*)
- [RFC4288] N. Freed et J. Klensin, "Spécifications du [type de support et procédures d'enregistrement](#)", [BCP 13](#), décembre 2005.
- [RFC4825] J. Rosenberg, "[Protocole d'accès de configuration \(XCAP\)](#) du langage de balisage extensible (XML)", mai 2007. (*P.S.*)
- [XML] Paoli, J., Maler, E., Bray, T., and C. Sperberg-McQueen, "Extensible Markup Language (XML) 1.0 (Second Edition)", World Wide Web Consortium FirstEdition REC-xml-20001006, octobre 2000, <<http://www.w3.org/TR/2000/REC-xml-20001006>>.

10.2 Références pour information

- [RFC3265] A.B. Roach, "[Notification d'événement spécifique](#) du protocole d'initialisation de session (SIP)", juin 2002. (*MàJ par [RFC6446](#) (Remplacée par la [RFC6665](#))*)
- [RFC3856] J. Rosenberg, "[Paquetage d'événement Presence](#) pour le protocole d'initialisation de session (SIP)", août 2004.
- [RFC3859] J. Peterson, "[Profil commun pour les services de présence](#) (CPP)", août 2004. (*P.S.*)
- [RFC4662] A. B. Roach et autres, "[Extension de notification d'événement](#) du protocole d'initialisation de session (SIP) pour les listes de ressources", août 2006. (*P.S.*)
- [RFC5025] J. Rosenberg, "[Règles d'autorisation de présence](#)", décembre 2007. (*P.S.*)

Adresse de l'auteur

Jonathan Rosenberg
Cisco
Edison, NJ
US
mél : jdrosen@cisco.com
URI : <http://www.jdrosen.net>

Déclaration complète de droits de reproduction

Copyright (C) The IETF Trust (2007)

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations contenues sont fournies sur une base "EN L'ÉTAT" et le contributeur, l'organisation qu'il ou elle représente ou qui le/la finance (s'il en est), la INTERNET SOCIETY, le IETF TRUST et la INTERNET ENGINEERING TASK FORCE déclinent toutes garanties, exprimées ou implicites, y compris mais non limitées à toute garantie que l'utilisation des informations encloses ne viole aucun droit ou aucune garantie implicite de commercialisation ou d'aptitude à un objet particulier.

Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne

prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur le répertoire en ligne des IPR de l'IETF à <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 fourni par l'Internet Society.