

Groupe de travail Réseau  
**Request for Comments : 4660**  
 Catégorie : Sur la voie de la normalisation  
 Traduction Claude Brière de L'Isle

H. Khartabil, Telio  
 E. Leppanen, Nokia  
 M. Lonnfors, Nokia  
 J. Costa-Requena, Nokia  
 septembre 2006

## Description fonctionnelle du filtrage de notification d'événement

### 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 Internet Society (2006).

### Résumé

Le cadre de notification d'événement SIP décrit l'usage du protocole de description de session (SIP, *Session Initiation Protocol*) pour les abonnements et notifications de changements de l'état d'une ressource. Ce document ne décrit pas un mécanisme par lequel le filtrage des informations de notification d'événement peut être réalisé.

Ce document décrit les opérations effectuées par un abonné afin de mettre en place les règles de filtrage associées à un abonnement aux informations de notification d'événement. Le traitement, par l'abonné, des réponses aux abonnements portant des règles de filtrage et le traitement des notifications avec des règles de filtrage qui leur sont appliquées est aussi décrit. De plus, le document décrit comment le notificateur se comporte quand il reçoit de telles règles de filtrage et comment une notification est construite.

## Table des matières

1. Introduction.....	2
2. Conventions.....	2
3. Fonctionnement du client.....	2
3.1 Mécanisme de transport .....	2
3.2 Corps SUBSCRIBE.....	2
3.3 Génération des demandes SUBSCRIBE par l'abonné.....	2
3.4 Traitement par l'abonné des demandes NOTIFY.....	4
4. Comportement du serveur de liste de ressources.....	4
4.1 URI de demande contre URI de filtre.....	4
4.2 Changement des filtres au cours d'un dialogue.....	5
5. Fonctionnement du serveur.....	6
5.1 Corps NOTIFY.....	6
5.2 Traitement par le notificateur des demandes SUBSCRIBE.....	6
5.3. Génération des demandes NOTIFY par le notificateur.....	6
5.4 Traitement des cas anormaux.....	8
6. Validation du document XML.....	8
7. Exemples.....	8
7.1 Exemples spécifiques de Présence.....	8
7.2 Exemples d'informations spécifiques de l'observateur.....	13
8. Considérations sur la sécurité.....	17
9. Considérations relatives à l'IANA.....	18
10. Remerciements.....	18
11. Références.....	18
11.1 Références normatives.....	18
11.2 Références pour information.....	19
Adresse des auteurs.....	19
Déclaration complète de droits de reproduction.....	20

## 1. Introduction

La notification d'événement SIP est décrite dans la [RFC3265]. Elle définit un cadre général pour l'envoi des abonnements et la réception des notifications dans les systèmes fondés sur SIP. Elle introduit le concept de paquetage d'événements, qui sont des applications concrètes du cadre général d'événement pour un usage spécifique des événements.

Le filtrage est un mécanisme pour contrôler le contenu des notifications d'événement. De plus, l'abonné peut spécifier les règles sur quand une notification devrait lui être envoyée. Le mécanisme de filtrage est supposé être particulièrement utile aux utilisateurs d'appareils mobiles sans fil. Les caractéristiques des appareils incluent normalement une forte latence, une faible bande passante, de faibles capacités de traitement de données, un petit affichage, et une puissance limitée de la batterie. De tels appareils peuvent bénéficier de la capacité de filtrer la quantité d'informations générée à la source de la notification d'événement. Cependant, les développeurs doivent être conscients de la charge de calcul sur la source de la notification d'événement. Ceci est discuté plus avant à la Section 8.

Il est déclaré dans la [RFC3265] que le notificateur peut envoyer un NOTIFY à tout moment, mais normalement, il est envoyé quand l'état de la ressource change. Elle déclare aussi que les notifications vont contenir l'état complet et actuel de la ressource qu'un certain abonné est autorisé à voir. Le format de ces informations d'état de ressource est spécifique du paquetage. Dans le présent mémoire, on suppose que le NOTIFY pour tout paquetage contient un document XML.

Le présent document, avec la [RFC4661], présente un mécanisme de filtrage par lequel un abonné décrit ses préférences sur quand les notifications sont à lui envoyer et ce qu'elles doivent contenir. Il décrit aussi comment fonctionne le notificateur et quand générer les notifications en tenant compte des filtres et fonctionnalités par défaut du paquetage/service.

Le format XML pour définir le filtre est décrit dans la [RFC4661].

## 2. Conventions

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] et indique les niveaux d'exigences pour les mises en œuvre conformes.

"Content" se réfère au document XML qui apparaît dans une notification reflétant l'état d'une ressource.

## 3. Fonctionnement du client

### 3.1 Mécanisme de transport

Le transport du filtre au serveur est réalisé en insérant le document XML, comme défini dans la [RFC4661], dans le corps de la demande SUBSCRIBE. Autrement, le document XML peut être chargé sur le serveur en utilisant des moyens qui sortent du domaine d'application du présent document.

### 3.2 Corps SUBSCRIBE

Les entités SIP qui se conforment à la présente spécification DOIVENT prendre en charge le type de contenu 'application/simple-filter+xml'.

### 3.3 Génération des demandes SUBSCRIBE par l'abonné

Ce paragraphe présente les fonctions supplémentaires exigées de l'abonné quand des filtres sont utilisés dans le corps des demandes SUBSCRIBE. Le fonctionnement normal des services (par exemple, comme défini dans les [RFC3856], [RFC3857], et [RFC4662]) est par ailleurs suivi.

Comme défini dans la [RFC3265], le message SUBSCRIBE PEUT contenir un corps. Ce corps va porter des informations de filtrage. Le respect de ces filtres est à la discrétion du notificateur et peut dépendre des politiques locales.

Aucun contenu dans le corps d'un SUBSCRIBE n'indique au notificateur qu'aucun filtre n'est demandé, de sorte que le

notificateur a pour instruction d'envoyer toutes les demandes NOTIFY en utilisant la propre politique du notificateur ou une politique spécifique du service. Noter que, par exemple, dans le cas de liste [RFC4662], le filtre peut avoir été chargé sur le serveur auparavant (par des moyens qui sortent du domaine d'application du présent document).

Si le corps du SUBSCRIBE inclut le filtre, le corps DOIT être du type MIME 'application/simple-filter+xml'.

### 3.3.1 Définition des règles de filtrage

Plusieurs filtres PEUVENT être inclus dans un SUBSCRIBE. Ceci est réalisé en incluant plusieurs éléments <filter> dans le filtre [RFC4661]. Chaque élément <filter> peut inclure un attribut 'uri'.

Une demande SUBSCRIBE destinée à un URI de liste [RFC4662] PEUT inclure plusieurs filtres spécifiques de ressources individuelles. Ceci est réalisé en incluant plusieurs éléments <filter> avec des URI de ressource différents dans chacun de ces éléments. Ce filtre spécifique de ressource est traité avant tout filtre spécifique de liste, si il en est. Le filtre spécifique de liste peut ou non inclure un URI.

De plus, sans considération de si le SUBSCRIBE est destiné à un URI de liste, il ne peut y avoir qu'un filtre applicable à une seule ressource ou domaine au sein d'un seul SUBSCRIBE. C'est-à-dire que chaque filtre dans un abonnement DOIT identifier de façon univoque une ressource ou un domaine.

Un filtre peut être activé et désactivé en utilisant l'attribut 'enabled' dans l'élément <filter>, comme décrit dans la [RFC4661].

### 3.3.2 URI de demande contre URI de filtre

L'URI dans le filtre définit une ressource cible. Par exemple, dans le cas du service Présence, ce sont les informations de présence de la présentité auxquelles le filtre est appliqué. L'abonné PEUT choisir de laisser indéfini l'URI dans le filtre. Si l'URI n'est pas défini dans le filtre, le filtre s'applique à la ressource identifiée dans l'URI de demande. De même, l'abonné PEUT définir un URI de filtre. Si l'URI de demande est un URI de liste [RFC4662], l'URI de filtre DOIT être l'URI de liste, un URI de sous liste, ou une ressource dont l'URI est un des URI qui résultent d'une recherche, par un serveur de liste de ressources (RLS, *Resource List Server*) sur l'URI de demande. Si il ne l'est pas, le filtre peut être ignoré ou peut être rejeté. La correspondance d'URI est faite en accord avec les règles de correspondance définies pour un schéma particulier (les règles de correspondance d'URI SIP sont définies dans la [RFC3261]).

Un filtre peut aussi être adressé à un domaine en utilisant l'attribut 'domain' au lieu de l'attribut 'uri'. Dans ce cas, le filtre s'applique aux ressources dans ce domaine. Cela peut être utilisé quand un abonnement est pour une ressource qui est une liste d'événements avec de nombreuses ressources provenant de différents domaines. Si un filtre spécifique d'une ressource individuelle est présent avec le filtre de domaine, ce filtre spécifique de ressource outrepassa tout filtre spécifique de domaine, si il en est.

### 3.3.3 Changement des filtres au cours d'un dialogue

L'abonné PEUT rétablir ou changer le filtre en produisant une nouvelle demande SUBSCRIBE dans le dialogue existant. Un SUBSCRIBE dans le dialogue existant qui ne contient pas de filtre est supposé maintenir les filtres existants. Cela signifie que les filtres sont persistents au sein d'un dialogue et ne sont supprimés que de façon explicite.

Un abonné qui demande la suppression d'un filtre peut le faire en utilisant l'attribut 'remove="true"', comme défini dans la [RFC4661].

Dans le cas où l'URI dans le filtre est celui d'une liste, un abonné peut outrepasser le filtre existant avec un filtre pour une ressource individuelle qui fait partie de la liste souscrite antérieurement en produisant un nouveau SUBSCRIBE dans le dialogue existant et en incluant un filtre, spécifique pour cette ressource individuelle, en utilisant un nouvel identifiant de filtre. Le nouveau filtre n'a pas besoin d'inclure le filtre d'origine car un filtre n'est supprimé que de la manière indiquée ci-dessus.

Un filtre est remplacé par l'abonné qui produit à nouveau le filtre en utilisant le même identifiant de filtre et en remplaçant le contenu du filtre. Remplacer un filtre en changeant l'identifiant de filtre et en gardant l'URI de ressource est considéré comme une erreur car cela amène le serveur à supposer que deux filtres sont placés sur la même ressource.

Là encore, un filtre peut être désactivé et réactivé en utilisant l'attribut 'enabled' dans l'élément <filter>, comme décrit dans la [RFC4661].

### 3.3.4 Interprétation des réponses SIP par l'abonné

La demande SUBSCRIBE va être confirmée par une réponse finale. Une réponse de classe 200 indique que l'abonnement a été accepté et qu'un NOTIFY va être envoyé immédiatement. Une réponse "200" indique que l'abonnement a été accepté et que le filtre est accepté. Une réponse "202" indique simplement que l'abonnement a été compris, que le type de contenu a été accepté, et que l'autorisation peut ou non avoir été accordée. Une réponse "202" indique aussi que le filtre n'a pas encore été accepté. L'acceptation du filtre PEUT arriver dans un NOTIFY suivant.

Une réponse finale non de classe 200 indique qu'aucun abonnement ou dialogue n'a été créé, et qu'aucun message NOTIFY suivant ne va être envoyé. Toutes les réponses finales non de classe 200 ont la même signification et traitement que décrit dans les [RFC3261] et [RFC3265].

Spécifiquement, une réponse "415" indique que le type MIME 'application/simple-filter+xml' n'est pas compris par le notificateur. Une réponse "488" indique que le type de contenu (filtre) est compris mais que certains de ses aspects n'ont pas été soit compris, soit acceptés.

### 3.4 Traitement par l'abonné des demandes NOTIFY

Si la réponse 2xx a été retournée pour le SUBSCRIBE, le NOTIFY qui suit PEUT contenir un corps qui décrit l'état présent de la ressource après l'application des filtres.

Si le NOTIFY indique qu'un abonnement a été terminé [RFC3265], l'abonnement est supposé être terminé. Le comportement dans de tels événements est aussi décrit dans la [RFC3265].

Si l'abonnement est indiqué comme actif, les demandes NOTIFY sont traitées comme décrit dans les documents spécifiques de paquetage et dans la [RFC3265].

## 4. Comportement du serveur de liste de ressources

Le serveur de liste de ressources est défini dans la [RFC4662]. Cette section décrit comment une telle entité se comporte en présence d'un filtre dans un abonnement à une liste.

### 4.1 URI de demande contre URI de filtre

Si l'URI n'est pas défini dans le filtre, le filtre s'applique à la liste de ressources identifiée dans l'URI de demande de la demande SUBSCRIBE. Il en résulte que les filtres sont appliqués à toutes les notifications que le RLS produit à cet abonnement. Le même traitement s'applique à un filtre qui définit un URI qui correspond à l'URI de demande de la demande SUBSCRIBE. C'est à-dire que le filtre s'applique à toutes les notifications que le RLS produit à cet abonnement.

Si l'URI indiqué par le filtre est pour une ressource dont l'URI est un de ceux qui résultent d'une recherche par le RLS sur l'URI de demande, le filtre pour cette ressource particulière est extrait et propagé dans la demande SUBSCRIBE envoyée à cette ressource. Il est possible d'avoir plus d'un filtre dans un corps de demande SUBSCRIBE, et donc un filtre spécifique d'une ressource DOIT être extrait et lui seul est propagé. Par exemple, si l'URI de demande dans un SUBSCRIBE a la valeur "sip:mybuddies@exemple.com", où "bob@exemple.com" est une ressource appartenant à cette liste, et si l'URI dans un filtre est "sip:bob@exemple.com", le filtre spécifique pour Bob est extrait et placé dans le corps du SUBSCRIBE envoyé à "bob@exemple.com".

Si l'URI indiqué par le filtre est pour une ressource dont l'URI N'est PAS sous le contrôle administratif du RLS, le RLS propage le filtre à tous les abonnements déployés. C'est pour s'accommoder du scénario où l'abonné sait qu'il y a des sous listes dans la liste des événements qui sont sous un domaine administratif différent de celui d'où l'abonnement d'origine a été envoyé, et l'abonné souhaite établir un filtre pour une ressource dans cette sous liste.

Si l'URI indiqué par le filtre est pour une ressource dont l'URI est sous le contrôle administratif du RLS mais ne fait pas partie de la liste de ressources à laquelle l'abonnement était adressé, le filtre n'est pas propagé. Dans ce cas, il est de la

responsabilité du RLS de s'assurer que ce filtre est appliqué aux notifications produites, si des informations sur cette ressource sont présentes.

Par exemple : Si on a deux listes, chacune localisée sur son propre RLS :

List1 (list1@exemple.com) sur le RLS1 a : bob@exemple.com list2@biloxi.com

List2 sur le RLS2 a : alice@biloxi.com sarah@exemple.com (Note : list2 est une ressource dans list1)

RLS1 reçoit la demande SUBSCRIBE suivante (le SUBSCRIBE est adressé à list1 et contient 2 filtres : un pour sarah@exemple.com et l'autre pour alice@biloxi.com) :

```
SUBSCRIBE sip:List1@exemple.com SIP/2.0
...
<?xml version="1.0" encoding="UTF-8"?>
<filter-set xmlns="urn:ietf:params:xml:ns:simple-filter">
  <ns-bindings>
    <ns-binding prefix="pidf" urn="urn:ietf:params:xml:ns:pidf"/>
  </ns-bindings>
  <filter id="999" uri="sip:sarah@exemple.com">
    <what>
      <include type="namespace">
        urn:ietf:params:xml:ns:pidf</include>
      <exclude>
        //pidf:tuple/pidf:note</exclude>
    </what>
  </filter>
  <filter id="8439" uri="sip:alice@biloxi.com">
    <what>
      <include>
        //pidf:tuple/pidf:status/pidf:basic</include>
    </what>
  </filter>
</filter-set>
```

RLS1 ventile les abonnements aux ressources sur list1. Le texte ci-dessus suggère que si un filtre est destiné à une ressource qui ne fait pas partie de la liste et est en-dehors du domaine administratif d'un RLS, alors ce filtre est propagé. Les restes sont consommés. Dans notre exemple, seul le filtre sur alice@biloxi.com est propagé car biloxi.com n'est pas sous le domaine administratif de RLS1. Le filtre sur sarah@exemple.com est consommé, et RLS1 doit appliquer ce filtre aux notifications qu'il reçoit.

La confrontation d'URI est faite en accord avec les règles de concordance définies pour un schéma particulier (les règles de concordance d'URI SIP sont définies dans la [RFC3261]).

Un filtre peut aussi être adressé à un domaine en utilisant l'attribut 'domain' au lieu de l'attribut 'uri'. Dans ce cas, le filtre s'applique aux ressources dans ce domaine, et le RLS NE DOIT PAS appliquer de filtres à des notifications qu'il envoie. Il DOIT plutôt transmettre le filtre avec tous les abonnements déployés aux notificateurs.

Comme indiqué au paragraphe 3.3.1, plusieurs filtres peuvent être présents dans une demande SUBSCRIBE. Des filtres peuvent aussi être ajoutés ou modifiés comme indiqué au paragraphe 3.3.3. Dans de telles circonstances, un RLS DOIT vérifier qu'il n'y a pas de filtres adressés à la même ressource ou domaine, et si il y en a, il DOIT rejeter la demande SUBSCRIBE avec une réponse d'erreur "488".

## 4.2 Changement des filtres au cours d'un dialogue

Si un RLS reçoit une demande de rafraîchissement d'abonnement sans filtre spécifié (charge utile vide) le RLS suppose que le client ne souhaite pas mettre à jour les filtres. Si un RLS reçoit un rafraîchissement d'abonnement avec un filtre contenant l'attribut 'remove="true"', comme défini dans la [RFC4661], le RLS suppose que le client retire ce filtre identifié par l'identifiant de filtre.

Si un RLS reçoit une demande de rafraîchissement d'abonnement avec un filtre qui existe déjà (c'est-à-dire, ayant le même identifiant de filtre) le RLS l'interprète comme un remplacement du filtre existant. Remplacer un filtre en changeant l'identifiant de filtre et en gardant l'URI de ressource est considéré comme une erreur car cela fait que le RLS suppose que deux filtres sont en place pour la même ressource.

Un filtre peut être désactivé et réactivé en utilisant l'attribut 'enabled' dans l'élément <filter>, comme décrit dans la [RFC4661].

## 5. Fonctionnement du notificateur

### 5.1 Corps SUBSCRIBE

Les entités SIP qui se conforment à la présente spécification DOIVENT prendre en charge le type de contenu 'application/simple-filter+xml'.

### 5.2 Traitement par le notificateur des demandes SUBSCRIBE

Ce paragraphe présente la fonction supplémentaire requise du notificateur quand des filtres sont utilisés dans les corps des demandes SUBSCRIBE. La fonction normale spécifique de paquetage est par ailleurs suivie.

Le notificateur va examiner le champ d'en-tête Content-Type et va retourner une réponse 415 si il ne comprend pas le type de contenu 'application/simple-filter+xml'.

Une réponse de classe 200 indique que l'abonnement a été accepté, et le NOTIFY va être envoyé immédiatement. une réponse "200" indique que l'abonnement a été accepté, que l'utilisateur est autorisé, et que le filtre est accepté. Une réponse "202" indique simplement que l'abonnement a été compris, mais que l'autorisation peut ou non avoir été accordée. Une réponse "202" indique aussi que les filtres n'ont pas encore été acceptés. L'acceptation des filtres PEUT arriver dans un NOTIFY suivant.

Les procédures décrites au paragraphe 5.4 sont suivies si une erreur est rencontrée.

Comme indiqué au paragraphe 3.3.1, plusieurs filtres peuvent être présents dans une demande SUBSCRIBE. Des filtres peuvent aussi être ajoutés ou modifiés comme indiqué au paragraphe 3.3.3. Dans de telles circonstances, un serveur DOIT vérifier qu'il n'y a pas de filtres adressés à la même ressource ou domaine, et si il y en a, il DOIT rejeter la demande SUBSCRIBE avec une réponse d'erreur "488".

#### 5.2.1 URI de demande contre URI de filtre

L'abonné peut avoir choisi de laisser indéfini l'URI dans le filtre. Si l'URI n'est pas défini dans le filtre, le filtre s'applique aux ressources identifiées dans l'URI de demande.

De même, l'abonné peut avoir choisi d'inclure un URI dans le filtre. Dans ce cas, le filtre s'applique à toutes les notifications envoyées avec un contenu associé à la ressource avec cet URI pour cet abonnement. Si l'URI de demande et l'URI dans le filtre ne correspondent pas, le filtre peut être ignoré ou rejeté. La confrontation d'URI est faite selon les règles de correspondance définies pour un schéma particulier (les règles de correspondance d'URI SIP sont définies dans la [RFC3261]).

Un filtre peut aussi être adressé à un domaine en utilisant l'attribut 'domain' au lieu de l'attribut 'uri'. Dans ce cas, le filtre s'applique aux ressources de ce domaine. Un notificateur DOIT ignorer tout filtre qui utilise un attribut 'domain' qui contient un domaine pour lequel ce notificateur n'est pas responsable. Le notificateur NE DOIT PAS appliquer un tel filtre à une notification qu'il envoie. Les notificateurs qui appartiennent au domaine DOIVENT appliquer le filtre à toutes les notifications qu'ils envoient pour cet abonnement, sauf si leur politique en décide autrement.

#### 5.2.2 Changement des filtres au cours d'un dialogue

Si un serveur reçoit une demande de rafraîchissement d'abonnement sans filtre spécifié (charge utile vide) il suppose que le client ne souhaite pas mettre à jour les filtres. Si il reçoit un rafraîchissement d'abonnement avec un filtre contenant l'attribut 'remove="true"', comme défini dans la [RFC4661], le serveur suppose que le client supprime le filtre identifié par

l'identifiant de filtre.

Si le serveur reçoit une demande de rafraîchissement d'abonnement avec un filtre qui existe déjà (c'est-à-dire, ayant le même identifiant de filtre) il l'interprète comme un remplacement du filtre existant. Remplacer un filtre en changeant l'identifiant de filtre et en gardant l'URI de ressource est considéré comme une erreur car cela amène le serveur à supposer que deux filtres sont placés sur la même ressource.

À réception du SUBSCRIBE avec le filtre, le notificateur DEVRAIT conserver le filtre tant que l'abonnement persiste. Le filtre PEUT être incorporé dans un abonnement existant (dans un dialogue actif) en envoyant un nouveau SUBSCRIBE qui inclut le filtre dans le corps.

### 5.3. Génération des demandes NOTIFY par le notificateur

Si la réponse envoyée au SUBSCRIBE était "202" et si le "202" a été choisi parce que le filtre pourrait n'être pas accepté cette fois, un NOTIFY PEUT être utilisé pour terminer l'abonnement si le filtre est trouvé inacceptable.

Comme décrit dans la [RFC3265], un message NOTIFY PEUT contenir un corps qui décrit l'état de la ressource. Ce corps est dans un des formats mentionnés dans le champ d'en-tête Accept du SUBSCRIBE, ou dans le format par défaut spécifique du paquetage si le champ d'en-tête Accept est omis.

Sur la base du contenu d'un filtre, le traitement suivant a lieu :

- o un filtre avec seulement un élément <what> va résulter en l'envoi des informations d'état de la ressource demandée dans cet élément <what> chaque fois qu'il y a un changement dans l'état de la ressource.
- o Un filtre avec seulement un élément <trigger> va résulter en l'envoi de toutes les informations d'état de la ressource chaque fois qu'il y a un changement dans l'état de la ressource qui correspond aux déclencheurs.
- o Un filtre avec des éléments <what> et <trigger> va résulter en l'envoi des informations d'état de la ressource demandée dans cet élément <what> chaque fois qu'il y a un changement dans l'état de la ressource qui correspond aux déclencheurs.

Quand un filtre est désactivé (en réglant l'attribut 'enabled' à "faux") cela signifie la même chose que l'absence de ce filtre. c'est-à-dire, tous les états et changements d'état sont rapportés en produisant une notification à l'abonné (en supposant qu'il n'y a pas d'autres filtres).

Quand un filtre est réactivé (en réglant l'attribut 'enabled' à "vrai" ou en omettant l'attribut 'enabled') le notificateur se comporte comme si le filtre avait juste été placé par la demande SUBSCRIBE qui l'active. les règles de NOTIFY immédiates, déclarées au paragraphe 5.3.1, s'appliquent.

#### 5.3.1 Génération de contenus NOTIFY

Si le NOTIFY envoyé est celui qui est envoyé immédiatement après une réponse 2xx au SUBSCRIBE original, son contenu DOIT être rempli conformément à l'élément <what> du filtre, sauf si le traitement des filtres va prendre trop longtemps ou si la demande NOTIFY suit une réponse "202" à la demande SUBSCRIBE et termine l'abonnement. Dans le cas où le filtre prend trop de temps à traiter, la demande NOTIFY envoyée peut être vide ou peut être remplie avec une valeur pré-configurée autorisée pour cet abonné. Si l'application du filtre résulte en ce qu'aucun contenu n'est livré, le NOTIFY DOIT être envoyé avec un contenu vide. Si le filtre contient des éléments <trigger>, le notificateur ignore les valeurs de déclencheur quand il génère la première demande NOTIFY.

L'entrée au filtre de contenu est un document XML spécifique du paquetage (par exemple, [RFC3863] et [RFC3858]) déduit conformément aux spécifications spécifiques du paquetage, (par exemple, [RFC3856] et [RFC3857]).

Le contenu est filtré conformément aux expressions dans l'élément <what> du filtre. L'expression indique les éléments et/ou attributs XML livrés. Les préfixes des espaces de noms des éléments du document XML à filtrer doivent être développés avant d'appliquer le filtre aux éléments.

L'expression déclare directement les éléments et attributs XML à livrer dans le NOTIFY, ainsi que leurs valeurs. En plus du contenu choisi, les espaces de noms de tous les éléments choisis sont aussi inclus dans le NOTIFY. Les éléments et/ou

attributs XML indiqués par l'expression dans l'élément <what> doivent être des éléments que l'abonné est autorisé à voir. Si ils ne le sont pas, la politique du notificateur dicte le comportement du notificateur (qui peut ignorer le filtre, des parties du filtre, ou le rejeter complètement). Les mises en œuvre doivent examiner avec attention de telles décisions de mise en œuvre ; l'abonné peut ne pas être informé des contenus autorisés et va donc très probablement inclure un filtre demandant des contenus non autorisés. Il est donc RECOMMANDÉ que les notificateurs ignorent juste les parties du filtre qui demandent des informations non autorisées (c'est-à-dire, le filtre dans l'élément <filter> où les contenus non autorisés sont demandés est ignoré). Si un blocage poli est utilisé par le notificateur, celui-ci peut choisir de livrer les notifications qui contiennent des informations boguées dans l'élément ou attribut non autorisé et appliquer le filtre après coup.

Le document XML résultant DOIT être bien formé et valide selon le schéma XML. Cela signifie que tous les éléments et attributs obligatoires, avec leurs valeurs, DOIVENT être inclus dans le document XML sans considération de l'expression. En d'autres termes, si le résultat de l'application d'un filtre sur un document XML est un document XML non valide, le notificateur DOIT ajouter des éléments et attributs, avec leurs valeurs, à partir du document XML original dans le nouveau qui est formulé afin qu'il soit valide.

### 5.3.2 Traitement des règles de déclenchement de notification

Il peut y avoir plusieurs éléments <trigger> dans un élément <filter>. Si les critères pour un des éléments <trigger> sont satisfaits, un NOTIFY DEVRAIT être généré.

Les éléments (éléments et/ou attributs XML) indiqués par l'expression dans l'élément <changed>, <added>, ou <removed> doivent être des éléments auxquels l'abonné est autorisé à accéder. Si ils ne le sont pas, la politique du notificateur dicte le comportement du notificateur (qui peut ignorer le filtre, des parties du filtre, ou le rejeter complètement).

### 5.4 Traitement des cas anormaux

En cas d'une définition invalide de filtre où le document XML du filtre n'est pas aligné avec le schéma XML du format de filtre [RFC4661], le notificateur rejette la demande SUBSCRIBE avec une réponse "488". Un champ d'en-tête Avertissement dans la réponse peut donner une meilleure indication sur la raison pour laquelle les filtres n'ont pas été acceptés. Si l'abonnement a été accepté avec une réponse "202" mais si le filtre invalide a été découvert après cela, un NOTIFY avec un état d'abonnement de valeur 'terminé' est envoyé. Une valeur de raison d'événement de "mauvais filtre", introduite ici, de subexp-params [RFC3265] PEUT être incluse.

En cas d'une expression erronée dans la définition de filtre, le notificateur ignore la définition de filtre ou termine l'abonnement.

Si un élément <what> ou <trigger> est vide, le notificateur poursuit comme si l'élément n'existait pas.

## 6. Validation du document XML

L'abonné du filtre DOIT s'assurer que le document XML inséré comme corps de demande SUBSCRIBE est bien formé et valide. L'abonné NE DOIT PAS insérer d'élément ou attributs d'extension dans le document XML sauf si il a accès au schéma d'extension et peut valider le document XML. Le notificateur du document XML PEUT valider le document XML conformément aux schémas, incluant les schémas d'extension, auxquels il a accès qui sont applicables à ce document XML.

## 7. Exemples

Les paragraphes qui suivent incluent des exemples de filtrage pour des informations de présence et d'observateur. Le format du filtre est conforme à la [RFC4661].

### 7.1 Exemples spécifiques de Présence

Ce paragraphe décrit trois cas d'utilisation où la solution de filtrage d'informations de présence est utilisée [RFC3856]. Dans le premier cas d'utilisation, l'observateur est intéressé à avoir des informations spécifiques de messagerie d'une certaine présentité. Dans le second cas d'utilisation, l'observateur est intéressé à obtenir des informations sur les moyens de



communication et les adresses de contact qui sont actuellement disponibles chez la présentité pour la communication. Le troisième cas d'utilisation montre comment une présentité peut demander des déclencheurs pour recevoir des notifications.

Ci-dessous sont les informations de présence de la présentité en PIDF [RFC3863]. Elles incluent deux tuplets : un pour la messagerie instantanée et un autre pour les informations relatives à la voix.

```
<?xml version="1.0" encoding="UTF-8"?>
  <presence xmlns="urn:ietf:params:xml:ns:pidf"
    xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
    entity="sip:presentity@exemple.com">
    <tuple id="432sd">
      <status>
        <basic>closed</basic>
      </status>
      <rpid:class>IM</rpid:class>
      <contact>im:presentity@exemple.com</contact>
    </tuple>
    <tuple id="thr76jk">
      <status>
        <basic>open</basic>
      </status>
      <rpid:class>voice</rpid:class>
      <contact>tel:2224055555@exemple.com</contact>
    </tuple>
  </presence>
```

### 7.1.1 L'abonné demande des informations relatives à la messagerie

L'abonné initie un abonnement aux informations de présence relatives à la messagerie de la présentité (MMS, IM, et SMS). L'abonnement inclut le filtre de limitation de contenu.

Le contenu filtré est indiqué avec une expression. Cette expression choisit l'élément <basic> et tous les éléments parents (c'est-à-dire, <status>, <tuple>, et son élément racine) l'élément <class>, et l'élément <contact>. Le filtre correspond si l'élément <class> contient "MMS", "SMS", ou "IM".

Dans ce cas, la notification inclut le contenu du tuple qui a la valeur "IM" dans son élément <class>.

La demande SUBSCRIBE provenant de l'abonné comporte un filtre :

```
SUBSCRIBE sip:presentity@exemple.com
Via: SIP/2.0/TCP client.exemple.com:5060;branch=z9hG4bKxjfdsjfk
To: <sip:presentity@exemple.com>
From: <sip:observateur@exemple.com>;tag:12341111
Call-ID: 32432udfidfjmk342
Cseq: 1 SUBSCRIBE
Expires: 3600
Event: Presence
Contact: <sip:observateur@client.exemple.com>
Content-Type: application/simple-filter+xml
Content-Length: ...
```

```
<?xml version="1.0" encoding="UTF-8"?>
<filter-set xmlns="urn:ietf:params:xml:ns:simple-filter">
  <ns-bindings>
    <ns-binding prefix="pidf" urn="urn:ietf:params:xml:ns:pidf"/>
    <ns-binding prefix="rpid" urn="urn:ietf:params:xml:ns:pidf:rpid"/>
  </ns-bindings>
  <filtre id="123" uri="sip:presentity@exemple.com">
    <what>
      <include type="xpath">
```

```

    //pidf:tuple[rpid:class="IM" ou rpid:class="SMS"
    ou rpid:class="MMS"]/pidf:status/pidf:basic
</include>
<include type="xpath">
    //pidf:tuple[rpid:class="IM" ou rpid:class="SMS"
    ou rpid:class="MMS"]/rpid:class
</include>
<include type="xpath">
    //pidf:tuple[rpid:class="IM" ou rpid:class="SMS"
    ou rpid:class="MMS"]/pidf:contact
</include>
</what>
</filter>
</filter-set>

```

Notification à l'abonné :

```

NOTIFY sip:observateur@client.exemple.com SIP/2.0
Via: SIP/2.0/TCP presence.exemple.com:5060;branch=z9hG4bKxjfdfer
To: <sip:observateur@exemple.com>;tag:12341111
From: <sip:presentity@exemple.com>;tag:232321
Call-ID: 32432udfidfjmk342
Cseq: 1 NOTIFY
Event: Presence
Subscription-State: active; expires=3599
Contact: sip:presentity@serveur.exemple.com
Content-Type: application/pidf+xml
Content-Length: ...

```

```

<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
entity="sip:presentity@exemple.com">
  <tuple id="432sd">
    <status>
      <basic>closed</basic>
    </status>
    <rpid:class>IM</rpid:class>
    <contact>im:presentity@exemple.com</contact>
  </tuple>
</presence>

```

### 7.1.2 L'abonné va chercher les informations sur les moyens de communication "Open"

L'abonné fait un abonnement aux moyens de communication disponibles de la présentité. L'abonnement inclut le filtre de limitation de contenu.

Le contenu filtré est indiqué par une expression. Cette expression choisit l'élément <basic> et tous les éléments parents (c'est-à-dire, <status>, <tuple>, et son élément racine) l'élément <class>, et l'élément <contact>. Le filtre correspond si la valeur de l'élément <basic> est "open".

Dans ce cas, la notification retourne le contenu du tuple qui a la valeur "open" dans l'élément <status>.

Demande SUBSCRIBE provenant de l'abonné incluant un filtre :

```

SUBSCRIBE sip:presentity@exemple.com SIP/2.0
Via: SIP/2.0/TCP client.exemple.com:5060;branch=z9hG4bKxjfdsjfk
To: <sip:presentity@exemple.com>
From: <sip:observateur@exemple.com>;tag:12341111
Call-ID: 32432udfidfjmk342

```

Cseq: 1 SUBSCRIBE  
 Expires: 3600  
 Event: Presence  
 Contact: <sip:observateur@client.exemple.com>  
 Content-Type: application/simple-filter+xml  
 Content-Length: ...

```
<?xml version="1.0" encoding="UTF-8"?>
<filtre-set xmlns="urn:ietf:params:xml:ns:simple-filter">
  <ns-bindings>
    <ns-binding prefix="pidf" urn="urn:ietf:params:xml:ns:pidf"/>
    <ns-binding prefix="rpid" urn="urn:ietf:params:xml:ns:pidf:rpid"/>
  </ns-bindings>
  <filter id="123" uri="sip:presentity@exemple.com">
    <what>
      <include type="xpath">
        //pidf:tuple/pidf:status[pidf:basic="open"]/pidf:basic
      </include>
      <include type="xpath">
        //pidf:tuple[pidf:status/pidf:basic="open"]/rpid:class
      </include>
      <include type="xpath">
        //pidf:tuple[pidf:status/pidf:basic="open"]/pidf:contact
      </include>
    </what>
  </filter>
</filtre-set>
```

Notification à l'abonné :

NOTIFY sip:observateur@client.exemple.com SIP/2.0  
 Via: SIP/2.0/TCP presence.exemple.com:5060;branch=z9hG4bKxjfd  
 To: <sip:observateur@exemple.com>;tag:12341111  
 From: <sip:presentity@exemple.com>;tag:232321  
 Call-ID: 32432udfidfjmk342  
 Cseq: 1 NOTIFY  
 Event: Presence  
 Subscription-State: active; expires=3599  
 Contact: sip:presentity@serveur.exemple.com  
 Content-Type: application/pidf+xml  
 Content-Length: ...

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
  entity="sip:presentity@exemple.com">
  <tuple id="thr76jk">
    <status>
      <basic>open</basic>
    </status>
    <rpid:class>voice</rpid:class>
    <contact>tel:2224055555@exemple.com</contact>
  </tuple>
</presence>
```

### 7.1.3 L'abonné demande des notifications quand l'état de la présence change

L'abonné souscrit à la présence, en spécifiant dans le filtre qu'il ne veut de notifications que quand l'élément <basic> a changé à la valeur "open".

Demande SUBSCRIBE provenant de l'abonné incluant un filtre :

```
SUBSCRIBE sip:presentity@exemple.com
Via: SIP/2.0/TCP client.exemple.com:5060;branch=z9hG4bKxjfdsjfk
To: <sip:presentity@exemple.com>
From: <sip:observateur@exemple.com>;tag:12341111
Call-ID: 32432udfidfjmk342
Cseq: 1 SUBSCRIBE
Expires: 3600
Event: Presence
Contact: <sip:observateur@client.exemple.com>
Content-Type: application/simple-filtre+xml
Content-Length: ...
```

```
<?xml version="1.0" encoding="UTF-8"?>
<filter-set xmlns="urn:ietf:params:xml:ns:simple-filter">
  <ns-bindings>
    <ns-binding prefix="pidf" urn="urn:ietf:params:xml:ns:pidf"/>
  </ns-bindings>
  <filter id="123" uri="sip:presentity@exemple.com">
    <trigger>
      <changed from="closed" to="open">
        /pidf:presence/pidf:tuple/pidf:status/pidf:basic
      </changed>
    </trigger>
  </filter>
</filter-set>
```

À un moment durant l'abonnement, un second document PIDF est créé avec les deux tuplets qui ont l'état de "closed" :

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
  entity="sip:presentity@exemple.com">
  <tuple id="432sd">
    <status>
      <basic>closed</basic>
    </status>
    <rpid:class>IM</rpid:class>
    <contact>im:presentity@exemple.com</contact>
  </tuple>
  <tuple id="thr76jk">
    <status>
      <basic>closed</basic>
    </status>
    <rpid:class>voice</rpid:class>
    <contact>tel:2224055555@exemple.com</contact>
  </tuple>
</presence>
```

Aucun NOTIFY n'est envoyé à l'abonné dans ce cas.

Maintenant, un troisième document PIDF est créé quand l'état IM change en "open" :

```
<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
  entity="sip:presentity@exemple.com">
  <tuple id="432sd">
    <status>
      <basic>open</basic>
```

```

</status>
<rpid:class>IM</rpid:class>
<contact>im:presentity@exemple.com</contact>
</tuple>
<tuple id="thr76jk">
  <status>
    <basic>closed</basic>
  </status>
  <rpid:class>voice</rpid:class>
  <contact>tel:2224055555@exemple.com</contact>
</tuple>
</presence>

```

Une notification contenant les deux tuplets est envoyée à l'abonné dans ce cas :

```

NOTIFY sip:observateur@client.exemple.com SIP/2.0
Via: SIP/2.0/TCP presence.exemple.com:5060;branch=z9hG4bKxjfd
To: <sip:observateur@exemple.com>;tag:12341111
From: <sip:presentity@exemple.com>;tag:232321
Call-ID: 32432udfidfjmk342
Cseq: 1 NOTIFY
Event: Presence
Subscription-State: active; expires=3599
Contact: sip:presentity@serveur.exemple.com
Content-Type: application/pdf+xml
Content-Length: ...

```

```

<?xml version="1.0" encoding="UTF-8"?>
<presence xmlns="urn:ietf:params:xml:ns:pidf"
  xmlns:rpid="urn:ietf:params:xml:ns:pidf:rpid"
  entity="sip:presentity@exemple.com">
  <tuple id="432sd">
    <status>
      <basic>closed</basic>
    </status>
    <rpid:class>IM</rpid:class>
    <contact>im:presentity@exemple.com</contact>
  </tuple>
  <tuple id="thr76jk">
    <status>
      <basic>open</basic>
    </status>
    <rpid:class>voice</rpid:class>
    <contact>tel:2224055555@exemple.com</contact>
  </tuple>
</presence>

```

## 7.2 Exemples d'informations spécifiques de l'observateur

Les exemples de ce paragraphe utilisent le gabarit de paquetage indéfini winfo avec le paquetage d'événement de présence [RFC3857].

Informations d'observateur à une présentité :

```

<?xml version="1.0"?>
<watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
  version="0" state="full">
  <watcher-list ressource="sip:presentity@exemple.com"
    package="presence">
    <watcher status="active"

```

```

    id="sr8fdsj"
    duration-subscribed="509"
    expiration="20"
    event="approved">sip:watcherA@exemple.com"</watcher>
<watcher status="pending"
  id="sr8fdsj"
  duration-subscribed="501"
  expiration="100"
  event="subscribe">sip:watcherB@exemple.com"</watcher>
<watcher status="terminated"
  id="sr8fdsj"
  duration-subscribed="500"
  expiration="0"
  event="rejected">sip:watcherC@exemple.com"</watcher>
<watcher status="active"
  id="sr8fdsj"
  duration-subscribed="20"
  expiration="30"
  event="approved">sip:watcherD@exemple.com"</watcher>
</watcher-list>
</watcherinfo>

```

### 7.2.1 L'abonné observateur s'abonne à obtenir toutes les informations sur les observateurs actifs

Demande SUBSCRIBE provenant de la présentité incluant le filtre :

```

SUBSCRIBE sip:presentity@exemple.com
Via: SIP/2.0/TCP client.exemple.com:5060;branch=z9hG4bKxjfdsjfk
To: <sip:presentity@exemple.com>
From: <sip:presentity@exemple.com>;tag:12341111
Call-ID: 32432udfidfjmk342
Cseq: 1 SUBSCRIBE
Expires: 3600
Event: Presence.winfo
Contact: sip:presentity@client.exemple.com
Content-Type: application/simple-filtre+xml
Content-Length: ...

```

```

<?xml version="1.0" encoding="UTF-8"?>
<filter-set xmlns="urn:ietf:params:xml:ns:simple-filter">
  <ns-bindings>
    <ns-binding prefix="wi" urn="urn:ietf:params:xml:ns:watcherinfo"/>
  </ns-bindings>
  <filter id="123" uri="sip:presentity@exemple.com">
    <what>
      <include>
        /wi:watcherinfo/wi:watcher-list[@package="presence"/]
        wi:watcher[@status="active"]
      </include>
    </what>
  </filter>
</filter-set>

```

Notification à l'abonné :

```

NOTIFY sip:presentity@client.exemple.com SIP/2.0
Via: SIP/2.0/TCP presence.exemple.com:5060;branch=z9hG4bKxjfdfer
To: sip:presentity@exemple.com;tag:12341111
From: sip:presentity@exemple.com;tag:232321
Call-ID: 32432udfidfjmk342

```

Cseq: 1 NOTIFY  
 Contact: sip:presentity@serveur.exemple.com  
 Event: Presence.winfo  
 Content-Type: application/watcherinfo+xml  
 Content-Length: ...

```
<?xml version="1.0"?>
<watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
version="0" state="full">
  <watcher-list ressource="sip:presentity@exemple.com"
    package="presence">
    <watcher status="active"
      id="sr8fdsj"
      duration-subscribed="509"
      expiration="20"
      event="approved">sip:watcherA@exemple.com"</watcher>
    <watcher status="active"
      id="sr8fdsj"
      duration-subscribed="20"
      expiration="30"
      event="approved">sip:watcherD@exemple.com"</watcher>
  </watcher-list>
</watcherinfo>
```

### 7.2.2 L'abonné observateur demande des informations sur les observateurs avec des conditions de durée d'abonnement spécifiques

Demande SUBSCRIBE provenant de la présentité incluant le filtre :

SUBSCRIBE sip:presentity@exemple.com  
 Via: SIP/2.0/TCP client.exemple.com:5060;branch=z9hG4bKxjfdsjfk  
 To: <sip:presentity@exemple.com>;tag:12341111  
 From: <sip:presentity@exemple.com>  
 Call-ID: 32432udfidfjmk342  
 Cseq: 1 SUBSCRIBE  
 Expires: 0  
 Event: Presence.winfo  
 Contact: <sip:presentity@client.exemple.com>  
 Content-Type: application/simple-filtre+xml  
 Content-Length: ...

```
<?xml version="1.0" encoding="UTF-8"?>
<filter-set xmlns="urn:ietf:params:xml:ns:simple-filter">
  <ns-bindings>
    <ns-binding prefix="wi" urn="urn:ietf:params:xml:ns:watcherinfo"/>
  </ns-bindings>
  <filter id="123" uri="sip:presentity@exemple.com">
    <what>
      <include>
        /wi:watcherinfo/wi:observateur-list[@package="presence"]/
        wi:watcher[@duration-subscribed>500]
      </include>
    </what>
  </filter>
</filter-set>
```

Notification à l'abonné :

NOTIFY sip:presentity@client.exemple.com SIP/2.0  
 Via: SIP/2.0/TCP presence.exemple.com:5060;branch=z9hG4bKxjfd

To: sip:presentity@exemple.com;tag:12341111  
 From: sip:presentity@exemple.com;tag:232321  
 Call-ID: 32432udfidfjmk342  
 Cseq: 1 NOTIFY  
 Contact: sip:presentity@serveur.exemple.com  
 Event: Presence.winfo  
 Content-Type: application/watcherinfo+xml  
 Content-Length: ...

```
<?xml version="1.0"?>
<watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
version="0" state="full">
<watcher-list ressource="sip:presentity@exemple.com" package="presence">
  <watcher status="active"
    id="sr8fdsj"
    duration-subscribed="509"
    expiration="20"
    event="approved">sip:watcherA@exemple.com</watcher>
  <watcher status="pending"
    id="sr8fdsj"
    duration-subscribed="501"
    expiration="100"
    event="subscribe">sip:watcherB@exemple.com</watcher>
</watcher-list>
</watcherinfo>
```

### 7.2.3 L'abonné observateur demande des informations spécifiques d'observateur sur des déclencheurs spécifiques

Ce filtre choisit les notifications d'informations d'observateur [RFC3858] à envoyer quand l'état d'abonnement en cours a changé de "pending" à "terminated". Dans la notification, seuls les observateurs qui ont un état de "terminated" et un événement de "rejected" sont inclus.

Demande SUBSCRIBE provenant de l'abonné observateur incluant le filtre :

SUBSCRIBE sip:presentity@exemple.com  
 Via: SIP/2.0/TCP client.exemple.com:5060;branch=z9hG4bKxjfdsjfk  
 To: <sip:presentity@exemple.com>;tag:12341111  
 From: <sip:presentity@exemple.com>  
 Call-ID: 32432udfidfjmk342  
 Cseq: 1 SUBSCRIBE  
 Expires: 0  
 Event: Presence.winfo  
 Contact: <sip:presentity@client.exemple.com>  
 Content-Type: application/simple-filter+xml  
 Content-Length: ...

```
<?xml version="1.0" encoding="UTF-8"?>
<filter-set xmlns="urn:ietf:params:xml:ns:simple-winfo-filter">
<ns-bindings>
  <ns-binding prefix="wi" urn="urn:ietf:params:xml:ns:watcherinfo"/>
</ns-bindings>
<filter id="123" uri="sip:presentity@exemple.com">
  <what>
    <include>
      /wi:watcherinfo/wi:watcher-list[@package="presence"/]
      wi:watcher[@status="terminated" and @event="rejected"]
    </include>
  </what>
</filter>
</filter-set>
```



```

    <changed from="pending" to="terminated">
      // @status
    </changed>
  </trigger>
</filter>
</filter-set>

```

À un moment durant l'abonnement, un second document Winfo est créé suite à un changement :

```

<?xml version="1.0"?>
  <watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
    version="0" state="full">
    <watcher-list ressource="sip:presentity@exemple.com" package="presence">
      <watcher status="active"
        id="sr8fdsj"
        duration-subscribed="509"
        expiration="20"
        event="approved">sip:watcherA@exemple.com</watcher>
      <watcher status="terminated"
        id="sr8fdsj"
        duration-subscribed="501"
        expiration="100"
        event="rejected">sip:watcherB@exemple.com</watcher>
      <watcher status="terminated"
        id="sr8fdsj"
        duration-subscribed="500"
        expiration="0"
        event="rejected">sip:watcherC@exemple.com</watcher>
      <watcher status="active"
        id="sr8fdsj"
        duration-subscribed="20"
        expiration="30"
        event="approved">sip:watcherD@exemple.com</watcher>
    </watcher-list>
  </watcherinfo>

```

La notification à l'abonné est créée, prenant en compte les éléments <trigger> et <what> :

```

NOTIFY sip:presentity@client.exemple.com SIP/2.0
Via: SIP/2.0/TCP presence.exemple.com:5060;branch=z9hG4bKxjfdfer
To: sip:presentity@exemple.com;tag:12341111
From: sip:presentity@exemple.com;tag:232321
Call-ID: 32432udfidfjmk342
Cseq: 1 NOTIFY
Contact: sip:presentity@serveur.exemple.com
Event: Presence.winfo
Content-Type: application/watcherinfo+xml
Content-Length: ...

```

```

<?xml version="1.0"?>
  <watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
    version="0" state="full">
    <watcher-list ressource="sip:presentity@exemple.com" package="presence">
      <watcher status="terminated"
        id="sr8fdsj"
        duration-subscribed="501"
        expiration="100"
        event="rejected">sip:watcherB@exemple.com</watcher>
      <watcher status="terminated"
        id="sr8fdsj"
        duration-subscribed="500"

```

```
    expiration="0"  
    event="rejected">sip:watcherC@exemple.com"</watcher>  
</watcher-list>  
</watcherinfo>
```

## 8. Considérations sur la sécurité

La présence de filtres dans le corps d'un message SIP a un effet significatif sur la façon dont la demande est traitée au serveur. Par suite, il est particulièrement important que les messages qui contiennent cette extension soient authentifiés et autorisés. L'authentification peut être réalisée en utilisant le mécanisme d'authentification par résumé décrit dans la [RFC3261]. La décision d'autorisation se fonde sur les permissions que la ressource (notificateur) a données à l'observateur. Un exemple d'une telle politique d'autorisation se trouve dans la [RFC5025].

Le traitement des demandes et recherche de filtres exige des ensembles d'opérations et de recherches, qui peuvent exiger une certaine quantité de calcul. Cela permet une attaque de déni de service par laquelle un utilisateur peut envoyer des demandes avec des nombres substantiels de messages de gros contenus, dans l'espoir de submerger le serveur. Pour contrer cela, le serveur peut établir une limite au nombre d'occurrences d'éléments <what>, <changed>, <added>, et <removed> qui sont permis dans les filtres. Une limite par défaut de 40 est RECOMMANDÉE ; cependant, les serveurs peuvent relever ou diminuer la limite selon leur capacité spécifique de traitement.

Les demandes peuvent révéler des informations sensibles sur les capacités d'un agent d'utilisateur. Si ces informations sont sensibles, elles DEVRAIENT être chiffrées en utilisant les capacités S/MIME de SIP [RFC3851]. Toutes les mesures de sécurité spécifiques d'un paquetage DOIVENT être suivies.

Propager des filtres dans les demandes SUBSCRIBE à des domaines étrangers révèle des informations sensibles sur les listes de ressources d'un utilisateur. Il est donc exigé qu'un RLS ne transmette pas un filtre si ce filtre est adressé à une ressource qui est sous le contrôle administratif du RLS, mais qui n'est pas sur la liste de ressources. Le paragraphe 4.1 donne un exemple où un tel scénario peut se produire.

Noter qu'un document filtré situé chez un abonné peut projeter une fausse réalité. Par exemple, si un abonné a demandé à être notifié quand une ressource a changé son état de présence de "closed" à "open" mais pas de "open" à "closed", alors l'abonné peut après coup être sous la fausse impression que l'état de présence de la ressource est "open", même longtemps après que la ressource l'a changé en "closed". Donc, les abonnés doivent être sûrs de ce qu'il mettent dans un filtre, comprendre ce qu'ils demandent, et être prêts à être désynchronisés avec l'état réel d'une ressource.

## 9. Considérations relatives à l'IANA

Une nouvelle valeur de raison d'événement "badfiltre" est définie pour représenter l'événement où le filtre n'est pas bien formé et/ou non accepté. Aucun enregistrement de l'IANA n'est requis pour cette valeur.

## 10. Remerciements

Les auteurs tiennent à remercier George Foti, Tim Moran, Sreenivas Addagatla, Juha Kalliokulju, Jari Urpalainen, et Mary Barnes de leurs précieux apports.

## 11. Références

### 11.1 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](#))

[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](#))

- [RFC3265] A.B. Roach, "[Notification d'événement spécifique](#) du protocole d'initialisation de session (SIP)", juin 2002. (MàJ par [RFC6446](#)) (Remplacée par la RFC6665)
- [RFC3851] B. Ramsdell, "Spécification du message d'extensions de messagerie Internet multi-objets/sécurisé (S/MIME) version 3.1", juillet 2004. (Obsolète, voir RFC5751)
- [RFC4661] H. Khartabil et autres, "[Format fondé sur le langage de balisage extensible](#) (XML) pour le filtrage de notification d'événement", septembre 2006. (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.)

## 11.2 Références pour information

- [RFC3856] J. Rosenberg, "[Paquetage d'événement Presence](#) pour le protocole d'initialisation de session (SIP)", août 2004.
- [RFC3857] J. Rosenberg, "[Paquetage-gabarit d'événement d'information](#) d'observateur pour le protocole d'initialisation de session (SIP)", août 2004. (P.S.)
- [RFC3858] J. Rosenberg, "[Format fondé sur le langage de balisage extensible](#) (XML) pour les informations d'observateur", août 2004. (P.S.)
- [RFC3863] H. Sugano et autres, "[Format des données d'information de présence](#) (PIDF)", août 2004.
- [RFC5025] J. Rosenberg, "[Règles d'autorisation de présence](#)", décembre 2007. (P.S.)

## Adresse des auteurs

Hisham Khartabil  
Telio  
P.O. Box 1203 Vika  
Oslo  
Norway  
téléphone : +47 2167 3544  
mél : [hisham.khartabil@telio.no](mailto:hisham.khartabil@telio.no)

Eva Leppanen  
Nokia  
P.O BOX 785  
Tampere  
Finland  
téléphone : +358 7180 77066  
mél : [eva-maria.leppanen@nokia.com](mailto:eva-maria.leppanen@nokia.com)

Mikko Lonnfors  
Nokia  
P.O BOX 321  
Helsinki  
Finland  
téléphone : + 358 71800 8000  
mél : [mikko.lonnfors@nokia.com](mailto:mikko.lonnfors@nokia.com)

Jose Costa-Requena  
Nokia  
P.O. Box 321  
FIN-00045 NOKIA GROUP  
Finland  
téléphone : +358 71800 8000  
mél : [jose.costa-requena@nokia.com](mailto:jose.costa-requena@nokia.com)

## Déclaration complète de droits de reproduction

Copyright (C) The IETF Trust (2006).

Le présent document est soumis aux droits, licences et restrictions contenus dans le BCP 78, et à [www.rfc-editor.org](http://www.rfc-editor.org), et sauf pour ce qui est mentionné ci-après, les auteurs conservent tous leurs droits.

Le présent document et les informations contenues sont fournis 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 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 pourrait 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 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](mailto:ietf-ipr@ietf.org).

**Remerciement**

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