```
XCON                                                        H. Khartabil
Internet-Draft                                           P. Koskelainen
Expires: March 10, 2005                                        A. Niemi
                                                                  Nokia
                                                      September 9, 2004
```

The Conference Policy Control Protocol (CPCP)
draft-ietf-xcon-cpcp-00

Status of this Memo

By submitting this Internet-Draft, I certify that any applicable patent or other IPR claims of which I am aware have been disclosed, and any of which I become aware will be disclosed, in accordance with RFC 3668.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on March 10, 2005.

Copyright Notice

Copyright (C) The Internet Society (2004). All Rights Reserved.

Abstract

The Conference Policy is defined as the complete set of rules for a particular conference manipulated by the conference policy server. The Conferece Policy Control Protocol (CPCP) is the protocol used by clients to manipulate the conference policy. This document describes the Conference Policy Control Protocol (CPCP). It specifies an Extensible Markup Language (XML) Schema that enumerates the conference policy data elements that enable a user to define a conference policy.

Table of Contents

```
   1.  Introduction . . . . . . . . . . . . . . . . . . . . . . . . .  3
   2.  Conventions Used in This Document  . . . . . . . . . . . . . .  3
   3.  Terminology  . . . . . . . . . . . . . . . . . . . . . . . . .  3
   4.  Structure of a Conference Policy document  . . . . . . . . . .  4
     4.1   MIME Type for CPCP XML Document  . . . . . . . . . . . . .  4
     4.2   Conference Root  . . . . . . . . . . . . . . . . . . . . .  4
     4.3   XML Document Description . . . . . . . . . . . . . . . . .  5
       4.3.1   Conference Settings  . . . . . . . . . . . . . . . . .  5
       4.3.2   Conference Information . . . . . . . . . . . . . . . .  7
       4.3.3   Conference Time  . . . . . . . . . . . . . . . . . . .  8
       4.3.4   Conference Authorization Rules . . . . . . . . . . . .  9
       4.3.5   Conference Dial-Out List . . . . . . . . . . . . . . . 20
       4.3.6   Conference Refer List  . . . . . . . . . . . . . . . . 20
       4.3.7   Conference Media Streams . . . . . . . . . . . . . . . 21
     4.4   XML Schema Extensibility . . . . . . . . . . . . . . . . . 21
     4.5   XML Schema . . . . . . . . . . . . . . . . . . . . . . . . 22
   5.  Conference Policy Manipulation and Conference Entity
       Behaviour  . . . . . . . . . . . . . . . . . . . . . . . . . . 27
     5.1   Overview of Operation  . . . . . . . . . . . . . . . . . . 27
     5.2   Use of External Lists  . . . . . . . . . . . . . . . . . . 27
     5.3   Communication Between Conference Entities  . . . . . . . . 28
     5.4   Manipulating Participant Lists . . . . . . . . . . . . . . 28
       5.4.1   Expelling a Participant  . . . . . . . . . . . . . . . 28
     5.5   Re-joining a Conference  . . . . . . . . . . . . . . . . . 30
   6.  Examples . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
     6.1   A Simple Conference Policy Document  . . . . . . . . . . . 30
     6.2   A Complex Conference Policy Document . . . . . . . . . . . 31
   7.  Security Considerations  . . . . . . . . . . . . . . . . . . . 33
   8.  IANA Considerations  . . . . . . . . . . . . . . . . . . . . . 34
     8.1   XCAP Application Usage ID  . . . . . . . . . . . . . . . . 34
     8.2   application/conference-policy+xml MIME TYPE  . . . . . . . 34
     8.3   URN Sub-Namespace Registration for
           urn:ietf:params:xml:ns:conference-policy . . . . . . . . . 35
   9.  Contributors . . . . . . . . . . . . . . . . . . . . . . . . . 36
   10.   Acknowledgements . . . . . . . . . . . . . . . . . . . . . . 36
   11.   References . . . . . . . . . . . . . . . . . . . . . . . . . 36
   11.1  Normative References . . . . . . . . . . . . . . . . . . . . 36
   11.2  Informative References . . . . . . . . . . . . . . . . . . . 38
       Authors' Addresses . . . . . . . . . . . . . . . . . . . . . . 38
       Intellectual Property and Copyright Statements . . . . . . . . 40
```

1. Introduction

The SIP conferencing framework [13] defines the mechanisms for multi-party centralized conferencing in a SIP environment.

Existing SIP mechanisms allow users, for example, to join and leave a conference, as described in [9]. A centralised server, called focus, can expel and invite users, and may have proprietary access control lists and user privilege definitions. This document defines an XML Schema in Section 4 that enumerates the conference policy data elements that enable a user to define a conference policy. This policy document may be given to a focus using a number of transports that are outside the scope of this document.

A focus conforming to this specification MUST support the XML object defined in Section 4.

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [2].

3. Terminology

This document uses terminology from [13]. Some additional definitions are introduced here.

Conference authorization policy (CAP): Conference authorization policy consists of an unordered set of rules, which control the permissions and privileges that are given to conference participants.

Conference Policy Server (CPS): Conference Policy Server. See [13]

Conference participant: Conference participant is a user who has an on-going session (e.g. SIP dialog) with the conference focus.

Floor control: Floor control is a mechanism that enables applications or users to gain safe and mutually exclusive or non-exclusive access to the shared object or resource in a conference.

Dial-Out List (DL): The Dial-out list (DL) is a list of users who the focus needs to invite to the conference.

Privileged user: A privileged user is a user that has the right to manipulate parts or all of the conference policy XML document.

Conference Policy URI: The URI of conference policy. It identifies the XML document. The URI construction is specified in [10].

Refer List (RL): The Refer list (RL) is a list of users who the focus needs to refer to the conference.

Sidebar: A sub-conference of a main conference.

4. Structure of a Conference Policy document

The conference policy document is an XML [6] document that MUST be well-formed and MUST be valid. The Conference policy documents MUST be based on XML 1.0 and MUST be encoded using UTF-8. This specification makes use of XML namespaces for identifying conference policy documents and document fragments. The namespace URI for elements defined by this specification is a URN [3], using the namespace identifier 'ietf' defined by [4] and extended by [15]. This URN is:

```
urn:ietf:params:xml:ns:conference-policy
```

4.1 MIME Type for CPCP XML Document

The MIME type for the CPCP XML document is "application/conference-policy+xml".

4.2 Conference Root

A conference policy document begins with the root element tag <conference>. Other elements from different namespaces MAY be present for the purposes of extensibility. Elements or attributes from unknown namespaces MUST be ignored. The conference policy is build up using the following:

- The <settings> element: This element is mandatory and contains various conference settings. It contains the conference URI(s), the maximum number of participants, the conference security level, and sidebar settings. It can occur only once in the document.

- The <info> element: This element is optional and includes information describing the conference, that can be used, for example, search purposes. This information can also be used in the session description when the focus is sending invitations. It

can occur only once in the document.

- The <time> element: This optional element defines conference time information, namely elements defining start and stop times for a media mixing.

- The <authorization> element: This optional element is the conference authorisation rules. It contains rules for users who can dial into the conference, users who are blocked from dialling in, amongst others.

- The <dialout-list> element: This optional element is for the dial-out list. It contains URIs for users that the focus will invite to the conference.

- The <refer-list> element: This optional element is for the refer list. It contains URIs for users that the focus will refer to the conference.

- The <ms> element: This optional element contains the media streams to be used in the conference.

The elements are described in more detail in the forthcoming sections.

A user may create a new conference at the CPS by placing a new conference policy document. Depending on server policy and user privileges, the CPS may accept the creation, or it may reject it.

A conference can be deleted permanently by removing the conference policy from the CPS, which consequently frees the resources. When the user deletes a conference, the CPS MUST also delete all its sub-conferences ("sidebars") at a server. Conference sidebars have unique URIs at the server. Sidebars are created in [18].

4.3 XML Document Description

4.3.1 Conference Settings

The <settings> element contains 2 sub-elements; the <conference-uri> element and the <max-participant-count> element.

<conference-uri> is a mandatory element. It can occur more than once to accommodate multiple signaling protocols. Once a conference URI is set, it MUST NOT be changed or removed for the duration of the conference. Only one URI per protocol MUST be set. URIs can be added at any time.

<max-participant-count> is an optional element. It carries the maximum number of participants allowed in the conference. When the maximum number of participants threshold is reached, no new users are not allowed to join until the number of participants decreases again. If using SIP, the server can reject a request to join (INVITE) with a "480 Temporarily Unavailable" response. Alternatively, the sever may implement a waiting queue.

<security-level> is an optional element. It describes the security level that the creator of the conference wishes to have for the conference being created, including signalling and media. There are 4 security levels defined: none, low, medium, and high with medium being the default value if this element is absent. Those levels are loosely defined here. The interpretation of those levels and the security protocols applied is left as a local policy of the focus. A focus may interpret those levels as follows:

none: No security is required for the signalling nor the media

low: Signalling and media integrity is required

medium: Signalling and media confidentiality is required

high: Signalling and media integrity and confidentiality are required

<allow-sidebars> is an optional element with a boolean value indicating if sidebars are allowed in this conference or not. The default value, if omitted, is "true" indicating that sidebars are allowed.

<sidebar> is an element identifying a side bar. Multiple <sidebar> elements can occur indicating multiple sidebars. No <sidebar> elements appearing in a conference policy indicates that there are no sidebars currently for this conference. A <sidebar> element contains a mandatory 'id' attribute that uniquely identifies the sidebar. It also contains an <uri> element that hold the sidebar URI. It can occur more than once to accommodate multiple signaling protocols. Once a sidebar URI is set, it MUST NOT be changed or removed for the duration of the conference. Only one URI per protocol MUST be set. URIs can be added at any time.

A sidebar MAY have its own policy. This policy is created exactly in the same manner as any other conference. The <policy> element in the <sidebar> element points to such policy. If the <policy> element is omitted, the sidebar inherits the policy of the conference it is a sidebar of.

A conference is identified by one or more conference URIs, one for each call signaling protocol that is supported. There must be at least one URI for a conference. Conference URIs can be proposed by the creator of the conference policy, as it may be useful to have human-friendly name in some cases, or can be assigned by the CPS. If the creator has proposed a conference URI, the server needs to decide whether to accept the name proposed by the client or not. It does this determination by examining if the conference URI already exists or not. If it exists, the CPS rejects the request to create the conference with that conference URI. Similarly, the CPS rejects the request to create a conference with a conference URI for a signalling protocol it does not support.

A Conference URI can be SIP, SIPS, TEL, or any supported URI scheme. The CPS MAY assign multiple conference URIs to a conference, one for each call signaling protocol that it supports.

Sidebar URIs are subject to the same behaviour.

4.3.2 Conference Information

The <info> element includes informative conference parameters which may be helpful describing the purpose of a conference, e.g. for search purposes or for providing host contact information. The <info> element has a special attribute 'xml:lang' to specify the language used in the contents of this element as defined Section 2.12 of [6].

Each conference has an optional <subject> element, which describes the current topic in a conference. The optional <display-name> element is the display name of the conference, which usually does not change over time.

<free-text> and <keywords> are optional elements. They provide additional textual information about the conference. This information can be made available to potential conference participants by means outside the scope of this document. Examples of usage could be searching for a conference based on some keywords.

The optional <web-page> element points to a URI where additional information about the conference can be found.

The optional <host-info> element contains the <uri>, <e-mail> and <web-page> elements. They give additional information about the user hosting the conference. This information can, for example, be included into the SDP fields of the SIP INVITE requests sent by the focus. The <uri> element is optional and can occur more than once.

4.3.3 Conference Time

The information related to conference time and lifetime is contained in the <time> element. The conference may occur for a limited period of time (i.e. bounded), or the conference may be unbounded (i.e. it does not have a specified end time). Bounded conferences may occur multiple times(e.g. on weekly basis).

The <time> element contains one or more <occurrence> elements each defining the time information of a single conference occurrence. Multiple <occurrence> elements MAY be used if a conference is active at multiple times; each additional <occurrence> element contains time information for a specific occurrence.

For each occurrence, the <mixing-start-time> element specifies when conference media mixing starts. the <mixing-stop-time> element specifies the time a conference media mixing stops. If the <mixing-start-time> element is not present, it indicates that the conference media mixing starts immediately. If the <mixing-stop-time> element is not present, it indicates that the conference occurrence is not bounded, i.e. permanent unitl the conference policy is removed from the server.

<mixing-start-time> and <mixing-stop-time> elements both have the mandatory 'require-participant' attribute. This attribute has one of 3 values: "none", "key-participant", and "participant". For mixing start time, this attribute allows a privileged user to define when media mixing starts based on the latter of the mixing start time, and the time the first participant or key participant arrives. If the value is set to "none", mixing starts according to the mixing start time. For mixing stop time, this attribute allows a privileged user to define when media mixing stops based on the earlier of the mixing stop time, and the time the last participant or key participant leaves. If the value is set to "none", mixing stops according to the mixing stop time.

Users can be allowed to join a conference before the media mixing time starts and after a certain time. A conference privileged user can indicate the time when users can join by populating the <can-join-after> element. Similarly, a conference privileged user can define the time after which new users are not allowed to join the conference anymore. This is done by populating the <must-join-before> element.

It is possible to define the time when users or resources on the dial-out list and on the refer-list are requested to join the conference by using the <request-users> element. It is also possible to define that the users and resources on the dial-out list and the

refer-list are requested to join the conference only after the first a participant or key participant has joined. This is achieved with the 'require-participant' attribute. A value of "none" indicates that the focus sends the requests immediately after the specified time has lapsed.

The absence of this conference time information indicates that a conference starts immediately and terminates when the conference policy is removed.

A running conference instance can be extended or stopped by modifying the conference time information. Note that those conference times do not guarantee resources for the conference to be available.

If a conference is in progress when deleted or stopped, the focus issues signalling requests to terminate all conference related sessions it has with participants. In SIP, the focus issues BYE requests.

4.3.4 Conference Authorization Rules

One of the key components of conference policy is the set of authorization rules that specify who is allowed to join a conference, see floors and request/grant them, subscribe to conference-information notifications and so on. The unordered list of authorization rules together define the conference authorization policy

The conference authorization rules are enclosed in the <authorization-rules> element and are formatted according to the XML schema defined in the common policy framework [1]. In the <authorization-rules> element, there can be multiple rules, each rule is represented by the <rule> element, each of which consist of three parts: conditions, actions and transformations. Conditions determine whether a particular rule applies to a request. Each action or transformation in the applied rule is a positive grant of permission to the conference participant. The details of each specific element and attribute is described in [1].

Asking the focus to allow certain users to join the conference is achieved by modifying an existing authorization rule or creating a new one. The CPS then informs the focus of such change.

If the conference is long-lasting, it is possible that new rules are added all the time but old rules are almost never removed (some of them are overwritten, though). This leads easily to the situation that the conference policy contains many unnecessary rules which are not really needed anymore. Therefore, there is a need to delete

rules. This can be achieved by removing that portion of the policy.

Conflicting rules may exist (for example, both allowed and blocked action is defined for same target). The common policy directives [1] dictate the behaviour in such situations.

This section outlines the new conditions, actions and transformations for conference authorization policy.

4.3.4.1 Conditions

4.3.4.1.1 Validity

The <validity> element, as defined in the common policy framework [1], expresses the rule validity period by two attributes, a starting and a ending time. Times are expressed in XML dateTime format. Expressing the lifetime of a rule implements a garbage collection mechanism. A rule maker might not have always access to the conference policy server to remove some rules which grant permissions. Hence this mechanisms allows to remove or invalidate granted permissions automatically without further interaction between the rule maker and the conference policy server.

To give a real life example, there are often meetings where management are allowed to join the first half of the conference and engineers are only allowed to join the conference during the second half of that meeting to report technical findings, etc. Two rules can be set in this scenario, the first rules allows the managers to join the conference without specifying a validity contraint. The second rule allows engineers to join an hour into the conference. The following example demostrates this. The meeting starts at 9:30 and ends at 12:30. The manager can join at any time while the engineer cannot only join before 10:30 (Note that the example is simplified for clarity).

```
     <rule id="1">
       <conditions>
         <identity>
           <id>manager@example.com</id>
         </identity>
       </conditions>
       <actions>
         <join-handling>allow</join-handling>
       </actions>
       <transformations/>
     </rule>
     <rule id="2">
       <conditions>
         <validity>
           <from>2004-12-17T10:30:00-05:00</from>
           <to>2004-12-17T12:30:00-05:00</to>
         </validity>
         <identity>
           <id>engineeer@example.com</id>
         </identity>
       </conditions>
       <actions>
         <join-handling>allow</join-handling>
       </actions>
       <transformations/>
     </rule>
     ...
     <time>
       <occurrence>
         <mixing-start-time required-participant="participant">
           2004-12-17T09:30:00-05:00</mixing-start-time>
         <mixing-stop-time required-participant="none">
           2004-12-17T12:30:00-05:00</mixing-stop-time>
       </occurrence>
     </time>
```

4.3.4.1.2 Identity

The <identity> element is already defined in the common policy framework [1]The presence of the <identity> element is a condition requires any identity within it to be authenticated before a rule is applied to it. This includes the <id> element (Section 4.3.4.1.2.1), the <any> element (Section 4.3.4.1.2.2), the <external-list> element (Section 4.3.4.1.2.4), their exceptions, and any future extension that carries an identity. The absence of the <identity> element with in a condition indicated that the rule applies to all unauthenticated

identities. That is participants that have provided no authenticated identity to the conference focus.

4.3.4.1.2.1 Interpreting the <id> Element

As earlier indicated, the <identity> element is already defined in the common policy framework [1]. However, the rules for interpreting the identities in <id> elements are left for each application to define separately. This document, however, does not define the rules for interpreting identities in <id> elements in conferencing applications since those interpretation rules are signalling protocol specific.

> OPEN ISSUE: Do we need to state more than this? How are identities derived from users that join using POTS, H.323, etc.?

4.3.4.1.2.2 Matching Any Identity

The <any> element is used to match any participant. This allows a conference to be open to any authenticated user. Just as for the <domain> element in <identity> element, The <any> element contains a list of <except> elements and allows to implement a simple blacklist mechanism. The <except> element contains the identity. It differs from the <domain> element in that the domain part is needed in the identity since it has not domain to refer to.

4.3.4.1.2.3 Matching Pseudonymous Identities

The <pseudonymous> element is used to match participants that have provided an authenticated identity to the conference focus, but have requested pseudonymity in the conference itself. A user requests pseudonymity by authenticating himself to the conference focus and providing an pseudonym in the signalling protocol (for example, using the From-header of a SIP reqeust). A rule allowing pseudonymous users to join looks like the following:

```
<rule id="4">
        <conditions>
                <pseudonymous>
        </conditions>
        <actions>
                <join-handling>allow</join-handling>
        </actions>
        <transformations/>
</rule>
```

The <pseudonymous> element can be combined with the <identity> element to provide the focus with a rule on what to do when a specific identity is authenticated and that identity is requesting pseudonymity through the signalling protocol. An example of such a rule follows:

```
<rule id="4">
  <conditions>
    <identity>
      <id>alice@example.com</id>
    </identity>
    <pseudonymous>
  </conditions>
  <actions>
    <join-handling>allow</join-handling>
  </actions>
  <transformations/>
</rule>
```

4.3.4.1.2.4 Matching Identities in External Lists

The <external-list> element can be used to match those participants that are part of a resource list that is created externally. Section 5.2 talks about the use of this condition in more detail.

4.3.4.1.2.5 Matching Referred Identities

The <has-been-referred> element can be used to match those participants that the focus has referred to the conference.

4.3.4.1.2.6 Matching Invited Identities

The <has-been-invited> element can be used to match those participants that the focus has invited into the conference.

4.3.4.1.2.7 Matching Identities of Former Conference Participants

The <has-been-in-conference> element can be used to match those participants that have joined the conference in the past.

4.3.4.1.2.8 Matching Identities Currently in the Conference

The <is-in-conference> element can be used to match those participants that are currently participating in the conference.

4.3.4.1.2.9 Matching Key Participant Identities

The <key-participant> element can be used to match those participants that are key participants of a conference.

4.3.4.1.2.10 Matching Identities on the Dial-out List

The <is-on-dialout-list> element can be used to match those participants that are on the dial-out list.

4.3.4.1.2.11 Matching Identities on the Refer List

The <is-on-refer-list> element can be used to match those participants that are on the refer list.

4.3.4.1.2.12 Floor ID

The <floor-id> element can be used to assign users as floor moderators. It MUST be used in conjunction with the <id> element that identifies the floor moderator. The <floor-id> element carries the floor ID of the floor that the user is a moderator of. The transformation <is-floor-moderator> is used to assert that the user identified using the <id> condition is the floor moderator of the floor identified in the <floor-id> condition.

The <floor-id> element is also used with the <floor-request-handling> element (Section 4.3.4.2.6) to set rules on who is allowed to request a floor.

4.3.4.1.2.13 Matching PIN Codes

The <pin> element can be used to match those participants that are have knowledge on a PIN code for the conference. For example:

```
<rule id="1">
        <conditions>
                <pin>12345</pin>
        </conditions>
        <actions>
                <join-handling>allow</join-handling>
        </actions>
        <transformations/>
</rule>
```

So the condition is the PIN. If any user knows the PIN, ignoring

their identity, the user is allowed to join.

A combination of the <identity> condition and the <pin> condition creates the possibility of assigning users personal PIN codes to enable them to join a conference. For example:

```
<rule id="2">
        <conditions>
                <identity>
                        <id>358401234567</id>
                </identity>
                <pin>67890</pin>
        </conditions>
        <actions>
                <join-handling>allow</join-handling>
        </actions>
        <transformations/>
</rule>
```

4.3.4.1.2.14 Matching Passwords

The <password> element can be used to match those participants that are have knowledge on a password for the conference. For example:

```
<rule id="3">
        <conditions>
                <password>pass1</password>
        </conditions>
        <actions>
                <join-handling>allow</join-handling>
        </actions>
        <transformations/>
</rule>
```

So the condition is the password. If any user knows the password, ignoring their identity, the user is allowed to join.

A combination of the <identity> condition and the <password> condition creates the possibility of assigning users personal passwords to enable them to join a conference. For example:

```
        <rule id="4">
                <conditions>
                        <identity>
                                <id>alice@example.com</id>
                        </identity>
                        <password>pass2</password>
                </conditions>
                <actions>
                        <join-handling>allow</join-handling>
                </actions>
                <transformations/>
        </rule>
```

4.3.4.2 Actions

4.3.4.2.1 Conference State Events

The <allow-conference-state> element represents a boolean action. If set to TRUE, the focus is instructed to allow the subscription to conference state events, such as the SIP Event Package for Conference State [14]. If set to FALSE, the subscription to conference state events would be rejected.

If this element is undefined it has a value of TRUE, causing the subscription to conference state events to be accepted.

4.3.4.2.2 Floor Control Events

The <allow-floor-events> element represents a boolean action. If set to TRUE, the focus is instructed to accept the subscription to floor control events. If set to FALSE, the focus is instructed to reject the subscription.

If this element is undefined, it has a value of FALSE, causing the subscription to floor control events to be rejected.

4.3.4.2.3 Conference Join Handling

The <join-handling> element defines the actions used by the conference focus to control conference participation. This element defines the action that the focus is to take when processing a particular request to join a conference. This element is an enumerated integer type, with defined values of:

block: This action instructs the focus to deny access to the conference. This action has a value of zero and it is the lowest value of the <join-handling> element. This action is the default action taken in the absence of any other actions.

confirm: This action instructs the focus to place the participant on a pending list (e.g., by parking the call on a music-on-hold server), awaiting moderator input for further actions. This action has a value of one.

allow: This action instructs the focus to accept the conference join request and grant access to the conference within the instructions specified in the transformations of this rule. This action has a value of two.

Note that placing a value of block for this element doesn't guarantee that a participant is blocked from joining the conference. Any other rule that might evaluate to true for this participant that carried an action whose value was higher than block would automatically grant confirm/allow permission to that participant.

4.3.4.2.4 Dynamically Referring Users

The <allow-refer-users-dynamically> element represents a boolean action. If set to TRUE, the identity is allowed to instruct the focus to refer a user to the conference without modifying the refer-list (in SIP terms, the identity is allowed to send a REFER request to the focus which results in the focus sending a REFER request to the user the referrer wishes to join the conference). If set to FALSE, the refer request is rejected.

If this element is undefined it has a value of FALSE, causing the refer to be rejected.

4.3.4.2.5 Dynamically Inviting Users

The <allow-invite-users-dynamically> element represents a boolean action. If set to TRUE, the identity is allowed to instruct the focus to invite a user to the conference without modifying the dial-out list (in SIP terms, the identity is allowed to send a REFER request to the focus which results in the focus sending an INVITE requested to the user the referrer wishes to join the conference). If set to FALSE, the refer request is rejected.

If this element is undefined it has a value of FALSE, causing the refer to be rejected.

4.3.4.2.6 Floor Request Handling

The <floor-request-handling> element defines the actions used by the conference focus to control floor requests. This element defines the action that the focus is to take when processing a particular request to a floor within a conference. This element is an enumerated integer type, with defined values of:

block: This action instructs the focus to deny the floor request. This action has a value of zero and it is the lowest value of the <floor-request-handling> element. This action is the default action taken in the absence of any other actions.

confirm: This action instructs the focus to allow the request. The focus then uses the defined floor algorithm to further allow of deny the floor. The algorithms used are outside the scope of this document.

Note that placing a value of block for this element doesn't guarantee that a participant is blocked from joining the conference. Any other rule that might evaluate to true for this participant that carried an action whose value was higher than block would automatically grant confirm/allow permission to that participant.

4.3.4.3 Transformations

4.3.4.3.1 Key Participant

When the <is-key-participant> element is set to TRUE, the joining participant is denoted as a key participant. If set to FALSE, the participant is not denoted as a key participant.

If this element is undefined, it has a value of FALSE, causing no key participant status to be given to the participant.

4.3.4.3.2 Floor Moderator

When the <is-floor-moderator> element is set to TRUE, the joining conference participant is denoted as floor moderator, meaning that they are privileged to control the floor in the conference. If set to FALSE, floor moderator privileges are not given to the conference participant.

If this element is undefined, it has a value of FALSE, causing no floor moderator privileges to being granted.

4.3.4.3.3 Conference Information

The <show-conference-info> element is of type boolean transformation. If set to TRUE, conference information is shown to the conference participant. If set to FALSE, conference information is not shown to the participant.

The <show-conference-info> element controls whether information in the <settings>, <time> and <info> elements may be made available publicly. For example, an application at a conference server might list the ongoing conferences on web page, or it may allow searching for conferences based on the keywords listed in the <Conference-info> element. Not setting this transformation to any users instructs the application not to reveal any such information to any user. However, information in other elements, such as <dialout-list>, should not be seen by anyone else other than a privileged user, even with this transformation enabled for a user.

If this element is undefined, it has a value of FALSE, causing no conference information to being shown.

4.3.4.3.4 Floor Holder

The <show-floor-holder> element is of type boolean transformation. If set to TRUE, the conference participant is able to see who is currently holding the floor. If set to FALSE, the participant is not able to see the floor holder.

If this element is undefined, it has a value of FALSE, causing the floor holder not be shown to the participant.

4.3.4.3.5 Floor Requests

The <show-floor-requests> element is of type boolean transformation. If set to TRUE, the conference participant is able to see the floor requests. If set to FALSE, the conference participant is not able to see floor requests.

If this element is undefined, it has a value of FALSE, causing the floor requests to not being seen by the conference participant.

4.3.4.3.6 Providing anonymity

A rule can be set that provides anonymity to a specific identity. In this case, the focus provides to the rest of the participants an anonymous identity for that user, for example anonymous1. This can be achieved by using the <provide-anonymity> element. It is a boolean transformation. If set to TRUE, the conference participants

will see an anonymous identity for the user whose identity is present in the conditions. An example of such rule follows:

```
<rule id="4">
  <conditions>
    <identity>
      <id>alice@example.com</id>
    </identity>
  </conditions>
  <actions>
    <join-handling>allow</join-handling>
  </actions>
  <transformations>
    <provide-anonymity>
  </transformation>
</rule>
```

If this element is undefined, it has a value of FALSE, causing the identity to be revealed.

4.3.5 Conference Dial-Out List

The dial-out list (DL) is a list of user URIs that the focus uses to learn who to invite to join a conference. This list can be created at conference policy creation time or updated during the conference lifetime so it can be used for mid-conference invites (and mass-invites) as well.

Asking the focus to invite (add) a user into the conference is achieved by adding that user's URI to the Dial-Out List (DL). The CPS then triggers the focus to send the conference invitation, eg: SIP INVITE as needed. Similarly, a user can be removed from the Dial-out list by removing the URI from the dial-out list.

The <dialout-list> element is optional and includes zero or more <target> elements and zero or more <external> elements. Those two elements includes the mandatory 'uri' attribute. The use of the <external> element is described in more detail in Section 5.2

4.3.6 Conference Refer List

The Refer List (RL) contains a list of resources that the focus needs to refer to the conference. In SIP, this is achieved by the focus sending a REFER request to those potential participants. This list can be updated during the conference lifetime so it can be used for

mid-conference refers as well.

The Refer List differs from the Dial-out list in that the dial-out list contains a list of resources that the focus will initiate a session with. The resources on the refer list, on the other had, are expected to initiate the session establishment towards the focus themselves. It is also envisioned that difference users will have different access rights to those lists and therefore a separation between the two is needed.

The <refer-list> element is optional and identical to the <dialout-list> element in Section 4.3.5.

4.3.7 Conference Media Streams

Media policy is an integral part of the conference policy. It defines e.g. what kind of media topologies exist in the conference. Media policy is documented in [20].This document does not define media policy, but instead enables the user to specify the media streams a conference has. This is used by the focus to know what media streams to invite users with and what media streams it should accept from dialling in users. Media can be added to or removed from a conference by a privileged user before or during a conference occurance. This might result in the focus modifying the session it has with each participant. In SIP, this means re-issuing and INVITE request modifying the session description (SDP).

The definition starts with the optional <media-streams> element. This element lists the media streams allowed for this conference. It can contain at most one of each media type using the <video>, <audio>, <message> and <text> elements.

4.4 XML Schema Extensibility

The schema as be extended at multiple places:

- The <conference> element to enable more conference policy information to be added

- The <settings> element to allow for future conference settings to be defined

- The <info> element to allow further conference and host information to be conveyed

- The <occurrence> element to allow further conference timing information

- The <target> element and the <external> element in <dialout-list> and <refer-list> to allow extensions on the behaviour of the focus. For example, how many times to retry inviting a user

- The <media-streams> element to allow introduction of new media streams

- The <sidebar> element to allow introduction of new sidebar information

4.5 XML Schema

```
   <?xml version="1.0" encoding="UTF-8"?>
   <xs:schema targetNamespace="urn:ietf:params:xml:ns:conference-policy"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns="urn:ietf:params:xml:ns:conference-policy"
elementFormDefault="qualified">
        <!-- This import brings in the XML language attribute xml:lang-->
        <xs:import namespace="http://www.w3.org/XML/1998/namespace"
schemaLocation="http://www.w3.org/2001/xml.xsd"/>
        <!-- The root Conference Element -->
        <xs:element name="conference">
                <xs:complexType>
                        <xs:sequence>
                                <xs:element name="settings"
type="ConferenceSettings"/>
                                <xs:element name="info" type="ConferenceInfo"
minOccurs="0"/>
                                <xs:element name="time" type="ConferenceTime"
minOccurs="0"/>
                                <xs:element name="authorization-rules"
type="ConferenceAuthorizationRules" minOccurs="0"/>
                                <xs:element name="dailout-list" type="UserList"
minOccurs="0"/>
                                <xs:element name="refer-list" type="UserList"
minOccurs="0"/>
                                <xs:element name="media-streams"
type="ConferenceMediaStreams" minOccurs="0"/>
                                <xs:any namespace="##other"
processContents="lax" minOccurs="0"/>
                        </xs:sequence>
                </xs:complexType>
        </xs:element>
        <!-- Conference Settings -->
        <xs:complexType name="ConferenceSettings">
                <xs:sequence>
```

```
                        <xs:element name="conference-uri" type="xs:anyURI"
maxOccurs="unbounded"/>
                        <xs:element name="max-participant-count"
type="xs:nonNegativeInteger" minOccurs="0"/>
                        <xs:element name="security-level" type="SecurityLevel"
default="medium" minOccurs="0"/>
                        <xs:element name="allow-sidebars" type="xs:boolean"
default="true" minOccurs="0"/>
                        <xs:element name="sidebar" type="Sidebar" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:any namespace="##other" processContents="lax"
minOccurs="0"/>
                </xs:sequence>
        </xs:complexType>
        <!-- Conference Info -->
        <xs:complexType name="ConferenceInfo">
                <xs:sequence>
                        <xs:element name="subject" type="xs:string"
minOccurs="0"/>
```

```
                        <xs:element name="display-name" type="xs:string"
minOccurs="0"/>
                        <xs:element name="free-text" type="xs:string"
minOccurs="0"/>
                        <xs:element name="keywords" minOccurs="0">
                                <xs:simpleType>
                                        <xs:list itemType="xs:string"/>
                                </xs:simpleType>
                        </xs:element>
                        <xs:element name="web-page" type="xs:anyURI"
minOccurs="0"/>
                        <xs:element name="host-info" minOccurs="0">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:element name="uri"
type="xs:anyURI" minOccurs="0" maxOccurs="unbounded"/>
                                                <xs:element name="e-mail"
type="xs:anyURI" minOccurs="0"/>
                                                <xs:element name="web-page"
type="xs:anyURI" minOccurs="0"/>
                                        </xs:sequence>
                                </xs:complexType>
                        </xs:element>
                        <xs:any namespace="##other" processContents="lax"
minOccurs="0"/>
                </xs:sequence>
                <xs:attribute ref="xml:lang"/>
        </xs:complexType>
        <!-- Conference time -->
        <xs:complexType name="ConferenceTime">
                <xs:sequence>
                        <xs:element name="occurrence" minOccurs="0"
maxOccurs="unbounded">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:element name="mixing-start-
time" type="StartStopTime" minOccurs="0"/>
                                                <xs:element name="mixing-stop-
time" type="StartStopTime" minOccurs="0"/>
                                                <xs:element name="can-join-
after" type="xs:dateTime" minOccurs="0"/>
                                                <xs:element name="must-join-
before" type="xs:dateTime" minOccurs="0"/>
                                                <xs:element name="request-
users" type="StartStopTime" minOccurs="0"/>
                                                <xs:any namespace="##other"
processContents="lax" minOccurs="0"/>
                                        </xs:sequence>
                                </xs:complexType>
                        </xs:element>
                </xs:sequence>
        </xs:complexType>
        <!-- Conferenece Authorisation -->
        <xs:complexType name="ConferenceAuthorizationRules">
                <xs:sequence>
                        <xs:element name="rule" type="ruleType" minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
        </xs:complexType>
        <xs:complexType name="ruleType">
                <xs:sequence>
                        <xs:element name="conditions" minOccurs="0">
                                <xs:complexType>
```

```
                                        <xs:sequence>
                                                <xs:element ref="condition"
minOccurs="0" maxOccurs="unbounded"/>
                                        </xs:sequence>
                                </xs:complexType>
                        </xs:element>
                        <xs:element name="actions" minOccurs="0">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:element ref="action"
minOccurs="0" maxOccurs="unbounded"/>
                                        </xs:sequence>
                                </xs:complexType>
                        </xs:element>
                        <xs:element name="transformations" minOccurs="0">
                                <xs:complexType>
                                        <xs:sequence>
                                                <xs:element
ref="transformation" minOccurs="0" maxOccurs="unbounded"/>
                                        </xs:sequence>
                                </xs:complexType>
                        </xs:element>
                </xs:sequence>
                <xs:attribute name="id" type="xs:string" use="required"/>
        </xs:complexType>
        <xs:element name="condition" abstract="true"/>
        <xs:element name="action" abstract="true"/>
        <xs:element name="transformation" abstract="true"/>
        <xs:element name="validity" substitutionGroup="condition">
                <xs:complexType>
                        <xs:all>
                                <xs:element name="from" type="xs:dateTime"/>
                                <xs:element name="to" type="xs:dateTime"/>
                        </xs:all>
                </xs:complexType>
        </xs:element>
        <xs:element name="identity" substitutionGroup="condition">
                <xs:complexType>
                        <xs:choice>
                                <xs:element name="id" type="xs:string"
maxOccurs="unbounded"/>
                                <xs:sequence>
                                        <xs:element name="domain"
type="xs:string"/>
                                        <xs:sequence minOccurs="0">
                                                <xs:element name="except"
type="xs:string" maxOccurs="unbounded"/>
                                        </xs:sequence>
                                </xs:sequence>
                                <xs:sequence>
                                        <xs:element name="any"
type="xs:string"/>
                                        <xs:sequence minOccurs="0">
                                                <xs:element name="except"
type="xs:string" maxOccurs="unbounded"/>
                                        </xs:sequence>
                                </xs:sequence>
                                <xs:sequence>
                                        <xs:element name="external-list"
type="xs:string"/>
                                        <xs:sequence minOccurs="0">
                                                <xs:element name="except"
type="xs:string" maxOccurs="unbounded"/>
                                        </xs:sequence>
                                </xs:sequence>
                        </xs:choice>
                </xs:complexType>
        </xs:element>
        <xs:element name="pseudonymous" type="xs:string"
substitutionGroup="condition"/>
        <xs:element name="has-been-referred" type="xs:string"
substitutionGroup="condition"/>
        <xs:element name="has-been-invited" type="xs:string"
substitutionGroup="condition"/>
        <xs:element name="has-been-in-conference" type="xs:string"
substitutionGroup="condition"/>
        <xs:element name="is-in-conference" type="xs:string"
substitutionGroup="condition"/>
        <xs:element name="key-participant" type="xs:string"
substitutionGroup="condition"/>
        <xs:element name="is-on-dialout-list" type="xs:string"
substitutionGroup="condition"/>
        <xs:element name="is-on-refer-list" type="xs:string"
substitutionGroup="condition"/>
        <xs:element name="floor-id" type="xs:anyURI"
substitutionGroup="condition"/>
        <xs:element name="pin" type="xs:anyURI" substitutionGroup="condition"/>
        <xs:element name="password" type="xs:anyURI"
substitutionGroup="condition"/>
        <xs:element name="allow-conference-state" type="xs:boolean"
substitutionGroup="action"/>
        <xs:element name="allow-floor-events" type="xs:boolean"
substitutionGroup="action"/>
        <xs:element name="join-handling" type="JoinHandling"
substitutionGroup="action"/>
        <xs:element name="allow-refer-users-dynamically" type="xs:boolean"
substitutionGroup="action"/>
        <xs:element name="allow-invite-users-dynamically" type="xs:boolean"
substitutionGroup="action"/>
        <xs:element name="floor-request-handling" type="xs:boolean"
substitutionGroup="action"/>
        <xs:element name="is-key-participant" type="xs:boolean"
substitutionGroup="transformation"/>
        <xs:element name="is-floor-moderator" type="xs:boolean"
```

```
substitutionGroup="transformation"/>
        <xs:element name="show-conference-info" type="xs:boolean"
substitutionGroup="transformation"/>
        <xs:element name="show-floor-holder" type="xs:boolean"
substitutionGroup="transformation"/>
        <xs:element name="show-floor-requests" type="xs:boolean"
substitutionGroup="transformation"/>
        <xs:element name="provide-anonymity" type="xs:boolean"
substitutionGroup="transformation"/>
        <!-- User List -->
        <xs:complexType name="UserList">
                <xs:sequence>
                        <xs:element name="target" type="Target" minOccurs="0"
maxOccurs="unbounded"/>
                        <xs:element name="external" type="Target" minOccurs="0"
maxOccurs="unbounded"/>
                </xs:sequence>
        </xs:complexType>
        <!-- Conference Media Streams -->
        <xs:complexType name="ConferenceMediaStreams">
                <xs:sequence>
                        <xs:element name="video" type="xs:string"
minOccurs="0"/>
                        <xs:element name="audio" type="xs:string"
minOccurs="0"/>
                        <xs:element name="message" type="xs:string"
minOccurs="0"/>
                        <xs:element name="text" type="xs:string" minOccurs="0"/
>
                        <xs:any namespace="##other" processContents="lax"
minOccurs="0"/>
```

```
                </xs:sequence>
        </xs:complexType>
        <!-- Target  -->
        <xs:complexType name="Target">
                <xs:sequence>
                        <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
                </xs:sequence>
                <xs:attribute name="uri" type="xs:anyURI" use="required"/>
        </xs:complexType>
        <!-- Start/Stop time -->
        <xs:complexType name="StartStopTime">
                <xs:simpleContent>
                        <xs:extension base="xs:dateTime">
                                <xs:attribute name="required-participant"
use="required">
                                        <xs:simpleType>
                                                <xs:restriction
base="xs:string">
                                                        <xs:enumeration
value="key-participant"/>
                                                        <xs:enumeration
value="participant"/>
                                                        <xs:enumeration
value="none"/>
                                                </xs:restriction>
                                        </xs:simpleType>
                                </xs:attribute>
                        </xs:extension>
                </xs:simpleContent>
        </xs:complexType>
        <!-- Security Level -->
        <xs:simpleType name="SecurityLevel">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="none"/>
                        <xs:enumeration value="low"/>
                        <xs:enumeration value="medium"/>
                        <xs:enumeration value="high"/>
                </xs:restriction>
        </xs:simpleType>
        <!-- Join Handling -->
        <xs:simpleType name="JoinHandling">
                <xs:restriction base="xs:string">
                        <xs:enumeration value="block"/>
                        <xs:enumeration value="allow"/>
                        <xs:enumeration value="confirm"/>
                </xs:restriction>
        </xs:simpleType>
        <!-- Sidebar -->
        <xs:complexType name="Sidebar">
                <xs:sequence>
                        <xs:element name="uri" type="xs:anyURI"
maxOccurs="unbounded"/>
                        <xs:element name="policy" type="xs:anyURI"
minOccurs="0"/>
                        <xs:any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
```

```
                </xs:sequence>
                <xs:attribute name="id" type="xs:string" use="required"/>
        </xs:complexType>
   </xs:schema>
```

5. Conference Policy Manipulation and Conference Entity Behaviour

5.1 Overview of Operation

This document assumes that the user knows the location of conference policy serve, the details of that discovery are beyond the scope of this document.

CPCP allows clients to manipulate the conference policy at the conference policy server (CPS). CPS is able to inform the focus about changes in conference policy, if necessary. For example, if new users are added to the dial-out list, then conference policy server informs the focus which makes the invitations as requested.

Some assumptions about the conferencing architecture are made. Clients always connect to the conference policy server (CPS) when they perform manipulation operations. It is assumed that the CPS informs other conferencing entities, such as focus, the floor control server and the mixer directly or via the focus. For example, if user A wants to expel user B from an ongoing conference, user A must first manipulate the conference policy data. The CPS then communicates that change to the focus to perform the operation.

User privileges are defined in [19]

5.2 Use of External Lists

External lists MAY be used in a conference policy. They can be used in the dial-out list, the refer-list and the authorization policy. An external list is a list of resources created by means outside the scope of this document.

A privileged user of the conference policy uses an external list by placing its URI in an conference policy element that is dedicated to carrying external list URIs. The external list URI is the URI used to manipulate the list and not the URI used to signal to the list. There are three such elements documented in this memo: the <external-list> element in the authorization rules (Section 4.3.4.1.2.4) and the <external> element in both, the dialout list (Section 4.3.5) and the refer list (Section 4.3.6). At the time the focus needs to activate the policy surrounding the URI, the focus fetches the URIs for the members of the external list using the list

URI. For example, a conference creator creates a conference and places the URI of an external list in the dial-out list. At some point, the focus needs to invite using on the dial-out list to join the conference. It is at that moment that the focus retrieves the members of the external list. It then sends INVITE (in SIP terms) to the members of that external list. This results in all participants connected to one focus.

It can happen that the external list is not accessible at the time the focus requires it. In this case, the external list is ignored, and in the case of an authorization rule, that rule fails.

There are also cases where the external list has been manipulated. It is outside the scope of this document how the focus can learn of such manipulation. But if is does, it reacts in a similar manner as it would have if the list was local and has been modified.

If an external list contains a reference to yet another list, that referenced list is also fetched if the focus has not already done so. This is to avoid list loops.

5.3 Communication Between Conference Entities

The communication between different (logical) conferencing elements is beyond the scope of this document. It can be expected that in most cases CPS includes also those logical functions.

5.4 Manipulating Participant Lists

A user with sufficient privileges is allowed to perform user management operations, such as adding a new user to the conference or expelling a user from the conference. These operations are performed by modifying the conference policy at the conference policy server. After authorising the user to do such manipulations, the conference policy server communicates the change to the focus. The focus reacts by performing singlling operations such as sending SIP INVITE, BYE or REFER.

5.4.1 Expelling a Participant

Expelling a user is performed by a privileged user creating or manipulating an existing authorization rule and setting that user's <join-handling> action to "block>. The focus reacts by terminating the session with that participant, such as a sending SIP BYE request.

Care must be taken since if one rules allows a user to join and one blocks a user from joining, the result in that the user is allowed to join. For example, Bob can join a conference since an authorization

rule has been defined to allow everyone at example.com:

```
     <rule id="1">
             <conditions>
                     <identity>
                             <domain>example.com</domain>
                     </identity>
             </conditions>
             <actions>
                     <join-handling>allow</join-handling>
             </actions>
             <transformations/>
     </rule>
```

Setting the following rule will not block Bob from joining nor will it expel him since the above rule overrides it:

```
     <rule id="2">
             <conditions>
                     <identity>
                             <uri>bob@example.com</uri>
                     </identity>
             </conditions>
             <actions>
                     <join-handling>block</join-handling>
             </actions>
             <transformations/>
     </rule>
```

So, in order to expel Bob, the original rule has to be modified using the <except> element:

```
        <rule id="1">
                <conditions>
                        <identity>
                                <domain>example.com</domain>
                                <except>bob@domain.com</except>
                        </identity>
                </conditions>
                <actions>
                        <join-handling>allow</join-handling>
                </actions>
                <transformations/>
        </rule>
```

5.5 Re-joining a Conference

Participants can drop out of a conference for many reasons including: client crash, out of coverage, had to leave for a while. It might be of interest to enable that user to re-join the conference. To allow that, participants that have departed the conference gracefully can only re-join if a privileged user has added an authorization rule allowing them to join. Participants that have departed the conference ungracefully (eg: crash) require a special behaviour from the focus . The focus is aware when a user has not gracefully departed a conference (for example; it did not receive a SIP BYE request and media is no longer being received). If this is the case, the focus is required to re-issue the invitation or referral to that user after a pre-configured unit of time.

6. Examples

6.1 A Simple Conference Policy Document

The simplist of a conference policy document contains the conference URI, a dial-out list, and the media. An example looks like this:

```
   <?xml version="1.0" encoding="UTF-8"?>
   <conference xmlns="urn:ietf:params:xml:ns:conference-policy"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <settings>
                <conference-uri>sip:myconference@example.com</conference-uri>
        </settings>
        <dailout-list>
                <target uri="sip:bob@example.com"/>
                <target uri="sip:alice@example.com"/>
                <target uri="sip:john@example.com"/>
                <target uri="sip:robert@example.com"/>
        </dailout-list>
        <media-streams>
                <audio/>
        </media-streams>
   </conference>
```

6.2 A Complex Conference Policy Document

Alice creates a conference with the follows policy:

- Conference URIs are suggested to be sip:myconference@example.com and tel:+3581234567.

- Maximum number of participants in the conference is 10.

- The security level for the conference is medium.

- The conference allows sidebars

- Media mixing starts at the latter of 9:30 am and the first participant arrives

- Media mixing sends at 12:30 pm. The conference does not need a key participant to continue.

- Users can join 5 minutes before media mixing starts and cannot join half an hour before media mixing ends.

- Users are requested to join a conference (invited and referred) 5 minutes before the conference starts and no participant nor key-participant is needed for this action to take place.

- Everyone at the domain example.com is allowed to join and can subscribe to the conference state event package.

- Alice is a key participant

- Alice will be invited to join the conference while Sarah will be referred to the conference.

- Two media are made available in the conference:audio and video.

The resulting CPCP document looks like

```
   <?xml version="1.0" encoding="UTF-8"?>
   <conference xmlns="urn:ietf:params:xml:ns:conference-policy"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <settings>
                <conference-uri>sip:myconference@example.com</conference-uri>
                <max-participant-count>10</max-participant-count>
                <allow-sidebars>true</allow-sidebars>
        </settings>
        <info xml:lang="en-us">
                <subject>What's happening tonight</subject>
                <display-name>Party Goer's</display-name>
                <free-text>John and Peter will join the conference soon</free-
text>
                <keywords>party nightclub beer</keywords>
                <host-info>
                        <uri>sip:Alice@example.com</uri>
                        <uri>tel:+3581234567</uri>
                        <e-mail>mailto:Alice@example.com</e-mail>
                        <web-page>http://www.example.com/users/Alice</web-page>
                </host-info>
        </info>
        <time>
                <occurrence>
                        <mixing-start-time required-
participant="participant">2004-12-17T09:30:00-05:00</mixing-start-time>
                        <mixing-stop-time required-
participant="none">2004-12-17T12:30:00-05:00</mixing-stop-time>
                        <can-join-after>2001-12-17T09:25:00-05:00</can-join-
after>
                        <must-join-before>2004-12-17T12:00:00-05:00</must-join-
before>
                        <request-users required-
participant="none">2001-12-17T09:30:00-05:00</request-users>
                </occurrence>
        </time>
        <authorization-rules>
                <rule id="1">
```

```
                        <conditions>
                                <identity>
                                        <domain>example.com</domain>
                                </identity>
                        </conditions>
                        <actions>
                                <allow-conference-state>true</allow-conference-
state>
```

```
                                <join-handling>allow</join-handling>
                        </actions>
                        <transformations/>
                </rule>
                <rule id="2">
                        <conditions>
                                <identity>
                                        <id>alice@example.com</id>
                                </identity>
                        </conditions>
                        <actions/>
                        <transformations>
                                <is-key-participant>true</is-key-participant>
                        </transformations>
                </rule>
        </authorization-rules>
        <dailout-list>
                <target uri="sip:bob@example.com"/>
        </dailout-list>
        <refer-list>
                <target uri="sip:sarah@example.com"/>
        </refer-list>
        <media-streams>
                <video/>
                <audio/>
        </media-streams>
   </conference>
```

7. Security Considerations

A conference document may contain information that is highly sensitive. Its delivery to the conference server needs to happen strictly, paying special attention to integrity and confidentiality. Reading the document is also a security concern since the conference policy contains sensitive information like the topic of the conference, who is allowed to join and the URIs of the users that can participate.

Manipulations of the conference policy have similar security issues. Users with relevant privileges can manipulate parts of the conference policy giving themselves and others privileges to manipulate the conference policy, including the dial-out list and the security level settings for a conference. This can happen because the conference policy itself carries the identities and the authorization rules that apply to those identities. Those authorization rules carry the privileges that certain identities have. If an unauthorized user gets access to this document (pretending to be someone else), s/he

can manipulate those rules giving himself and other unauthorized users access to the conference policy. S/he can also manipulate other parts of the conference policy under a false identity. Some of the things that a malicious user can do include: denying users certain privileges, giving himself floor moderation, removing users from lists, removing rules for certain identities, giving privileges to other malicious users, changing the media streams and changing conference time. Therefore, it is very important that only authorized clients are able to manipulate the conference policy. Any conference policy transport protocol MUST provide authentication, confidentiality and integrity.

In the case that XCAP is used to create and manipulate a conference policy, the XCAP base specification mandates that all XCAP servers MUST implement HTTP Authentication: Basic and Digest Access Authentication [16]. Furthermore, XCAP servers MUST implement HTTP over TLS [17]. It is recommended that administrators of XCAP servers use an HTTPS URI as the XCAP root services URI, so that the digest client authentication occurs over TLS. By using these means, XCAP client and server can ensure the confidentiality and integrity of the XCAP created conference policy document and its manipulation operations, and that only authorized clients are allowed to perform them.

8. IANA Considerations

8.1 XCAP Application Usage ID

This section registers a new XCAP Application Usage ID (AUID) according to the IANA procedures defined in..

Name of the AUID: conference-policy
Description: Conference policy application manipulates conference policy at a server.

8.2 application/conference-policy+xml MIME TYPE

MIME media type: application

MIME subtype name: conference-policy+xml

Mandatory parameters: none

Optional parameters: Same as charset parameter application/xml as specified in RFC 3023 [7].

Encoding considerations: Same as encoding considerations of application/xml as specified in RFC 3023 [7].

Security considerations: See section 10 of RFC 3023 [7] and section Section 8 of this document.

Interoperability considerations: none.

Published specification: This document.

Applications which use this media type: This document type has been used to support conference policy manipulation for SIP based conferencing.

Additional information:

Magic number: None

File extension: .cl or .xml

Macintosh file type code: "TEXT"

Personal and email address for further information: Petri Koskelainen (petri.koskelainen@nokia.com)

Intended Usage: COMMON

Author/change controller: The IETF

8.3 URN Sub-Namespace Registration for urn:ietf:params:xml:ns:conference-policy

This section registers a new XML namespace, as per guidelines in URN document [15].

URI: The URI for this namespace is urn:ietf:params:xml:ns:conference-policy.

Registrant Contact: IETF, XCON working group, Petri Koskelainen (petri.koskelainen@nokia.com)

XML:

```
BEGIN
<?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>Conference Policy Namespace</title>
</head>
<body>
  <h1>Namespace for Conference Policy</h1>
  <h2>application/conference-policy+xml</h2>
  <p>See <a href="[[[URL of published RFC]]]">RFCXXXX</a>.</p>
</body>
</html>
END
```

9. Contributors

Jose Costa-Requena

Simo Veikkolainen

Teemu Jalava

10. Acknowledgements

The authors would like to thank Markus Isomaki, Adam Roach, Eunsook Kim, Roni Evan, Alan Johnston, Hannes Tschofenig and the IETF XCON working group for their feedback and suggestions.

11. References

11.1 Normative References

[1] Schulzrinne, H., Tschofenig, H., Cuellar, J., Polk, J. and J. Rosenberg, "Common Policy", Internet-Draft I-D.ietf-geopriv-common-policy, February 2004.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, BCD 14, March 1997.

[3] Moats, R., "URN Syntax", RFC 2141, May 1997.

[4] Moats, R., "A URN Namespace for IETF Documents", RFC 2648,

August 1999.

[5] Rosenberg, J., Shulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M. and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[6] Bray, T., Paoli, J., Sperberg-McQueen, C. and E. Maler, "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C REC REC-xml-20001006, October 2000.

[7] Murata, M., Laurent, S. and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[8] Koskelainen, P. and H. Khartabil, "Requirements for conference policy control protocol", draft-ietf-xcon-cpcp-req-01 (work in progress), January 2004.

[9] Johnston, A. and O. Levin, "Session Initiation Protocol Call Control - Conferencing for User Agents", draft-ietf-sipping-cc-conferencing-03 (work in progress), February 2004.

[10] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", draft-ietf-simple-xcap-02 (work in progress), February 2004.

[11] Rosenberg, J., "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usage for Presence Lists", draft-ietf-simple-xcap-list-usage-02 (work in progress), February 2004.

[12] Rosenberg, J., "A Session Initiation Protocol (SIP) Event Package for Modification Events for the Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Managed Documents", draft-ietf-simple-xcap-package-01 (work in progress), February 2004.

[13] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol", draft-ietf-sipping-conferencing-framework-01 (work in progress), October 2003.

[14] Rosenberg, J., Shulzrinne, H. and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State", draft-ietf-sipping-conference-package-03, February 2004.

[15] Mealling, M., "The IETF XML Registry", RFC 3688, January 2004.

[16] Franks, J., "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[17] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

11.2 Informative References

[18] Rosen, B., "SIP Conferencing: Sub-conferences and Sidebars", draft-rosen-xcon-conf-sidebars-00 (work in progress), July 2004.

[19] Khartabil, H. and A. Niemi, "Privileges for Manipulating a Conference Policy", draft-ietf-xcon-conference-policy-privileges-00 (work in progress), September 2004.

[20] Jennings, C. and B. Rosen, "Media Mixer Control for XCON", draft-jennings-xcon-media-control-00 (work in progress), February 2004.

[21] Handly, M., Eriksson, G., Jacobson, V. and C. Perkins, "Grouping of Media Lines in SDP", draft-ietf-mmusic-sdp-new-18 (work in progress), June 2004.

Authors' Addresses

Hisham Khartabil
Nokia
P.O. Box 321
Helsinki FIN-00045
Finland

EMail: hisham.khartabil@nokia.com

Petri Koskelainen
Nokia
P.O. Box 100 (Visiokatu 1)
Tampere FIN-33721
Finland

EMail: petri.koskelainen@nokia.com

Aki Niemi
Nokia
P.O. Box 100
NOKIA GROUP, FIN 00045
Finland

Phone: +358 50 389 1644
EMail: aki.niemi@nokia.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Copyright Statement

Copyright (C) The Internet Society (2004). This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.