

Network Working Group
Request for Comments: 4575
Category: Standards Track

J. Rosenberg
Cisco Systems
H. Schulzrinne
Columbia University
O. Levin, Ed.
Microsoft Corporation
August 2006

A Session Initiation Protocol (SIP)
Event Package for Conference State

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document defines a conference event package for tightly coupled conferences using the Session Initiation Protocol (SIP) events framework, along with a data format used in notifications for this package. The conference package allows users to subscribe to a conference Uniform Resource Identifier (URI). Notifications are sent about changes in the membership of this conference and optionally about changes in the state of additional conference components.

Table of Contents

1. Introduction	4
2. Terminology	4
3. Conference Event Package	4
3.1. Event Package Name	5
3.2. Filtering	5
3.3. Subscription Duration	5
3.4. NOTIFY Bodies	5
3.5. Notifier Processing of SUBSCRIBE Requests	6
3.6. Notifier Generation of NOTIFY Requests	6
3.7. Subscriber Processing of NOTIFY Requests	6
3.8. Handling of Forked Requests	7
3.9. Rate of Notifications	7
3.10. State Agents	7
4. Conference Document	7
4.1. Format	7
4.2. Namespace	7
4.3. Versioning	8
4.4. Partial Notifications Mechanism	8
4.5. Element Keys	9
4.6. Constructing Coherent State Procedure	9
5. Conference Data	11
5.1. Overview	11
5.2. <conference-info>	13
5.3. <conference-description>	14
5.3.1. <conf-uris>	14
5.3.2. <service-uris>	15
5.3.3. <maximum-user-count>	16
5.3.4. <available-media>	16
5.4. <host-info>	17
5.4.1. <display-text>	17
5.4.2. <web-page>	17
5.4.3. <uris>	17
5.5. <conference-state>	17
5.5.1. <user-count>	17
5.5.2. <active>	18
5.5.3. <locked>	18
5.6. <users> and Its <user> Sub-elements	18
5.6.1. <display-text>	19
5.6.2. <associated-aors>	19
5.6.3. <roles>	19
5.6.4. <languages>	19
5.6.5. <cascaded-focus>	19
5.6.6. <endpoint>	20
5.7. <endpoint>	20
5.7.1. <display-text>	20
5.7.2. <referred>	21

5.7.3.	<status>	21
5.7.4.	<joining-method>	22
5.7.5.	<joining-info>	23
5.7.6.	<disconnection-method>	23
5.7.7.	<disconnection-info>	23
5.7.8.	<media>	24
5.7.9.	<call-info>	24
5.8.	<media>	24
5.8.1.	<display-text>	25
5.8.2.	<type>	25
5.8.3.	<label>	25
5.8.4.	<src-id>	25
5.8.5.	<status>	26
5.9.	Sidebars	26
5.9.1.	<sidebars-by-ref>	26
5.9.2.	<sidebar-by-val>	26
6.	XML Schema	26
7.	Examples	35
7.1.	Basic Example	35
7.2.	Rich Example	36
8.	Security Considerations	40
8.1.	Connection Security	40
8.2.	Authorization Considerations	41
9.	IANA Considerations	41
9.1.	conference Event Package Registration	41
9.2.	application/conference-info+xml MIME Registration	42
9.3.	URN Sub-Namespace Registration for	43
9.4.	XML Schema Registration	43
9.5.	URI Purposes Sub-registry Establishment	44
10.	Acknowledgements	45
11.	References	45
11.1.	Normative References	45
11.2.	Informative References	46

1. Introduction

The Session Initiation Protocol (SIP) events framework [10] defines general mechanisms for subscribing to, and receiving notifications of, events within SIP networks. It introduces the notion of a package, which is a specific "instantiation" of the events framework for a well-defined set of events. Here, we define a SIP event package for tightly coupled conferences. This package can be used by the conference notification service as outlined in the SIP conferencing framework [16]. As described there, subscriptions to a conference URI are routed to the focus that is handling the conference. It acts as the notifier and provides clients with updates on conference state.

The information provided by this package is comprised of conference identifier(s), conference participants (optionally with their statuses and media description), conference sidebars, conference service URIs, etc.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [1] and indicate requirement levels for compliant implementations.

This document uses the conferencing terminology defined in Conferencing Framework [16]. In addition, the "roster" term is used to collectively refer to participants in a conference or a sub-conference.

3. Conference Event Package

The conference event package allows a user to subscribe to the information relating to a conference. In SIP, conferences are represented by URIs. These URIs identify a SIP user agent (UA), called a focus, that is responsible for ensuring that all users in the conference can communicate with each other, as described in Conferencing Framework [16]. The focus has sufficient information about the state of the conference to inform subscribers about it.

It is possible that a participant in the conference may in fact be another focus. In order to provide a more complete participant list, the focus MAY subscribe to the conference package of the other focus to discover the participant list in the cascaded conference. This information can then be included in notifications by use of the <cascaded-focus> element as specified by this package.

This section provides the details for defining a SIP-specific event notification package, as specified by [RFC 3265 \[10\]](#).

3.1. Event Package Name

The name of this event package is "conference". This package name is carried in the Event and Allow-Events header, as defined in [RFC 3265 \[10\]](#).

3.2. Filtering

Filters, which can be applied to conference subscriptions, are a desirable feature and can be considered as a subject of future standardization activities. This document does not define the filters for the conference package to be included in the SUBSCRIBE body.

A SUBSCRIBE for a conference package, being sent without a body, implies the default subscription filtering policy. The default policy is as follows:

- o Notifications are generated every time there is any change in the state of the conference.
- o Notifications do not normally contain full state; rather, they only indicate the state that has changed. The exception is a NOTIFY sent in response to a SUBSCRIBE. These NOTIFYS contain the full state of the information requested by the subscriber.

3.3. Subscription Duration

The default expiration time for a subscription to a conference is one hour. Once the conference ends, all subscriptions to that particular conference are terminated, with a reason of "noresource" as defined in [RFC 3265 \[10\]](#).

3.4. NOTIFY Bodies

According to [RFC 3265 \[10\]](#), the NOTIFY message will contain bodies that describe the state of the subscribed resource. This body is in a format listed in the Accept header field of the SUBSCRIBE, or a package-specific default if the Accept header field was omitted from the SUBSCRIBE.

In this event package, the body of the notification contains a conference information document that describes the state of a conference. All subscribers and notifiers MUST support the "application/conference-info+xml" data format described in [Sections 9](#)

and 5. By default, i.e., if no Accept header is specified to a SUBSCRIBE request, the NOTIFY will contain a body in the "application/conference-info+xml" data format. If the Accept header field is present, it MUST include "application/conference-info+xml" and MAY include any other types.

3.5. Notifier Processing of SUBSCRIBE Requests

The conference information contains very sensitive information. Therefore, all subscriptions SHOULD be authenticated and then authorized before approval. Authorization policy is at the discretion of the administrator, as always.

However, it is RECOMMENDED that all users in the conference be allowed to subscribe to the conference event package.

3.6. Notifier Generation of NOTIFY Requests

Notifications SHOULD be generated for the conference state when a new participant joins (i.e., gets "connected" to) or a participant leaves (i.e., gets "disconnected" from) the conference.

Subject to a local focus policy, additional changes in participants' status, changes in their media types, and other optional information MAY be reported by the focus.

Changes in sidebar rosters SHOULD be reported by the focus to their participants and MAY be reported to others, subject to local policy.

Changes in conference identifiers and service URIs SHOULD be reported by the focus to the conference package subscribers.

Changes in other conference state information MAY be reported by the focus to the conference package subscribers.

3.7. Subscriber Processing of NOTIFY Requests

The SIP events framework expects packages to specify how a subscriber processes NOTIFY requests in any package-specific ways, and in particular, how it uses the NOTIFY requests to construct a coherent view of the state of the subscribed resource.

Typically, the NOTIFY for the conference package will only contain information about those users whose state in the conference has changed. To construct a coherent view of the total state of all users, a subscriber to the conference package will need to combine NOTIFYS received over time.

Notifications within this package can convey partial information; that is, they can indicate information about a subset of the state associated with the subscription. This means that an explicit algorithm needs to be defined in order to construct coherent and consistent state. The details of this mechanism are specific to the particular document type. See [Section 4.6](#) for information on constructing coherent information from an application/conference-info+xml document.

3.8. Handling of Forked Requests

By their nature, the conferences supported by this package are centralized. Therefore, SUBSCRIBE requests for a conference should not generally fork. If forking happens in the network, subscribers to this package MUST NOT establish more than a single SIP dialog as a result of a single SUBSCRIBE request. In the foci cascading case, detailed conference information can be retrieved by establishing an individual SUBSCRIBE dialog with each participating focus.

3.9. Rate of Notifications

For reasons of congestion control, it is important that the rate of notifications not become excessive. As a result, it is RECOMMENDED that the server does not generate notifications for a single subscriber at a rate faster than once every 5 seconds.

3.10. State Agents

Conference state is ideally maintained in the element in which the conference resides. Therefore, a conference focus is the best-suited element to handle subscriptions to it. Cascaded foci MAY implement state agents (as defined in [RFC 3265](#) [10]) for this package.

4. Conference Document

4.1. Format

Conference information is an XML document that MUST be well-formed and valid. It MUST be based on Extensible Markup Language (XML) 1.0 and MUST be encoded using UTF-8 [14].

4.2. Namespace

This specification makes use of XML namespaces for identifying conference information documents and document fragments. The namespace URI for elements defined by this specification is a Uniform Resource Namespace (URN) [2], using the namespace identifier 'ietf'

defined by [6] and extended by RFC 3688 [21]. This URN is as follows:

urn:ietf:params:xml:ns:conference-info

4.3. Versioning

The conference information is described by a hierarchical XML structure with the root element <conference-info>. The root element is the only element in the schema that carries a meaningful version number for all the elements in the document. The whole conference information is associated with this version number.

The 'version' attribute MUST be included in the root <conference-info> element.

4.4. Partial Notifications Mechanism

This specification defines a basic partial notifications mechanism by using a 'state' attribute as described below. This mechanism MUST be supported by all subscribing clients. Additional general partial notifications mechanisms can be defined and applied to this package in the future.

All sub-elements in the <conference-info> hierarchical XML structure can be classified in two groups: permissible for partial notifications or not. Elements that carry a substantial amount of data that is subject to frequent changes are permissible for partial notifications and have a 'state' attribute attached to them. All future extensions to this schema MUST define which extension elements can also use this mechanism. All other elements can be updated as atomic pieces of data only.

Below is the complete list of elements permissible to use the partial notifications mechanism defined in this specification. (Note that future partial notifications mechanisms can potentially be applicable to additional elements.)

- o Element <conference-info>
- o Element <users>
- o Element <user>
- o Element <endpoint>
- o Element <sidebars-by-val>
- o Element <sidebars-by-ref>

The 'state' attribute value indicates whether the reported information about the element is "full" or "partial", or whether the element has been "deleted" from the conference state document. The default value of any 'state' attribute is "full".

A 'state' attribute of a child element in the document MUST be consistent with its parent 'state'. This means that if the parent's 'state' is "full", the state of its children MUST be "full". If the parent's 'state' is "partial", the state of its children MAY be either "partial", "full", or "deleted". A parent element with "deleted" 'state' SHOULD NOT contain child elements. Any information provided for child elements of a "deleted" parent MUST be ignored by the package subscriber.

4.5. Element Keys

The defined XML schema has a property of unique identification among sub-elements of a common parent, which makes it possible to use the partial notifications mechanism defined in this document. This property is achieved by defining a key to each sub-element that can appear multiple times under the common parent.

In the context of this specification, the element key is the set of mandatory attributes or sub-elements of an element. The key value MUST be unique for the element among its siblings of the same type.

In the context of this specification, two keys of type xs:anyURI are considered to be equal if the UTF-8 representations of the keys (including all URI parameters that can be included with the URI) are identical. Consequently, using relative URIs and lexical white space in these keys is NOT RECOMMENDED.

Below is the list of elements (subject to partial notifications of their parent elements) with their keys as defined by this specification:

- o Element <user> uses as the key 'entity'
- o Element <endpoint> uses as the key 'entity'
- o Element <media> uses as the key 'id'
- o Element <entry> (child to <sidebars-by-val>) uses as the key 'entity'
- o Element <sidebars-by-ref> uses as the key <uri>

4.6. Constructing Coherent State Procedure

This section describes the algorithm for constructing a coherent conference state by a subscriber to the conference package. Using software programming abstraction, the subscriber maintains a single

local version number for the whole conference document and a local element instance for each element in the schema. Also, for each element with keys (as defined above), the subscriber maintains a virtual table with a row for each existing key value.

The first time a NOTIFY with a "full" document is received (as indicated by the value of the 'state' attribute in the <conference-info> element), the conference package subscriber MUST set the local 'version' number to the value of the 'version' attribute from the received <conference-info> and populate local data with the received information.

Each time a new NOTIFY is received, the value of the local version number and the value of the 'version' attribute in the new received document are compared. If the value in the document is equal to or less than the local version, the document is discarded without processing.

Otherwise, if the received NOTIFY contains a "full" or "deleted" state, the conference package subscriber MUST set the local 'version' number to the value of the 'version' attribute from the received <conference-info> and replace the local information with the received document. Receiving "deleted" state for the <conference-info> element means that the conference has ceased to exist and the subscriber SHOULD terminate the subscription by sending the SUBSCRIBE with Expires = 0.

Otherwise (i.e., if the received NOTIFY contains "partial" state), if the 'version' number in the received document is more than one number higher than the previous local version number, the subscriber MUST generate a subscription refresh request to trigger a full state notification. If the 'version' number in the document is one higher than the local version number, the local version number is updated accordingly and the document is used to update the local content as described below.

For each sub-element of the <conference-info> element in the received document,

1. If the element contains "full" state, the whole local element content is flushed and repopulated from the document.
2. Otherwise, if the element contains "deleted" state, the whole element MUST be removed from the local content.
3. Otherwise, if the element contains "partial" state:

- 3.1. For elements with keys, the subscriber compares the keys received in the update with the keys in the local tables.
 - 3.1.1. If a key does not exist in the local table, a row is added, and its content is set to the element information from the update.
 - 3.1.2. Otherwise, if a key of the same value does exist, for each sub-element in the row, the algorithm is applied from step 3.2.
- 3.2. For each atomic element received in the schema, the element is replaced with the new information as a whole. For each non-atomic element received in the schema with either no 'state' attribute included or the state attribute is set to "full", the element is replaced with the new information as a whole. Also, for each element with the state attribute set to "deleted", the whole element is removed from the local content.
- 3.3. For each non-atomic element with the state attribute set to "partial", the algorithm is applied recursively starting from step 3.1.

5. Conference Data

5.1. Overview

The <conference-info> document format is designed to convey information about the conference and about participation in the conference. The following non-normative diagram gives an example of the overall hierarchy used in this format. Conferences contain users who can be represented by multiple endpoints, each of which can have multiple media streams. Conferences can also include or reference "sidebar conferences". When sidebar information is incorporated into a conference information document in a <sidebars-by-val> element, each <entry> element represents a sidebar and can include any sub-elements permitted in the <conference-info> top-level element.

```

conference-info
|
|-- conference-description
|
|-- host-info
|
|-- conference-state
|
|-- users
|   |-- user
|   |   |-- endpoint
|   |   |   |-- media
|   |   |   |-- media
|   |   |   |-- call-info
|   |   |-- endpoint
|   |   |   |-- media
|   |-- user
|   |   |-- endpoint
|   |   |   |-- media
|
|-- sidebars-by-ref
|   |-- entry
|   |-- entry
|
|-- sidebars-by-val
|   |-- entry
|   |   |-- users
|   |   |   |-- user
|   |   |   |-- user
|   |-- entry
|   |   |-- users
|   |   |   |-- user
|   |   |   |-- user
|   |   |   |-- user

```

In most cases, this document does not mandate how the information, presented through the conference document to the subscribers, is obtained by the focus. In many cases, the information can be dynamically learned from the call signaling and can also be manually populated by an administrator - all subject to local policies. This document specifies what the XML elements mean in order to allow the subscribers to appropriately interpret it. Some portions of the information are intended for processing by automata; others are for human consumption only. For example, the <display-text> sub-elements of elements <conf-uris>, <service-uris>, <available-media>,

<host-info>, <endpoint>, and <media> are intended for display to human subscribers only.

Although in multiple places this document states that specific information "SHOULD" be communicated to the subscribers, note that particular conference package subscribers (e.g., representing a moderator, an administrator, or a cascaded focus) rely on accuracy of this information for their proper operation. Therefore, a conferencing server MUST ensure that all critical changes (stated as "SHOULD") are communicated to these specific subscribers; otherwise, these changes MUST be communicated to all subscribers to the conference information.

Following sections describe the XML schema in more detail.

5.2. <conference-info>

A conference information document begins with the root element tag <conference-info> of conference-type.

The following attributes are defined for <conference-info>:

entity: This attribute contains the conference URI that identifies the conference being described in the document. This is the SIP URI that an interested entity needs to SUBSCRIBE to in order to get the conference package information. Note that this URI can be listed as one of the URIs to be used in order to access the conference by SIP means and in accordance with [Section 5.3.1](#) below.

state: This attribute indicates whether the document contains the whole conference information ("full") or only the information that has changed since the previous document ("partial"), or whether the conference ceased to exist ("deleted"). For more detail, see [Section 4](#).

version: This attribute allows the recipient of conference information documents to properly order the received notifications, and it MUST be used with the root <conference-info> element. Version number is a 32-bit monotonically increasing integer scoped within a subscription. A server MUST increment the version number for each notification (full, partial, and deleted) being sent to a subscriber and reporting a change in the conference document state. For each partial notification, the version number MUST be increased by one. Note that a partial notification and a subsequent full notification over the same dialog MAY contain the same version number if no change in the conference state occurred in between.

The <conference-info> element is comprised of <conference-description>, <host-info>, <conference-state>, <users>, <sidebars-by-ref> and <sidebars-by-val> child elements. A "full" conference document MUST at least include the <conference-description> and <users> child elements.

Following sections describe these elements in detail. The full XML schema is provided in [Section 6](#).

5.3. <conference-description>

The <conference-description> element describes the conference as a whole.

The child elements <display-text>, <subject>, <free-text>, and <keywords> are used to describe the conference content:

<display-text>: Contains descriptive text suitable for human consumption, for example, listing in a directory

<subject>: Contains the subject of the conference

<free-text>: Contains an additional longer description of the conference

<keywords>: Contains a list of space-separated string tokens that can be used by search engines to better classify the conference

Additional child elements <conf-uris> and <service-uris> are used to describe the conference-related URIs; <maximum-user-count> and <available-media> are used to describe the overall characteristics.

This information is typically derived from the system conference policies, is set before the conference activation, and is rarely changed during the conference lifetime.

The following sections describe the remaining elements in more detail. Other sub-elements can extend <conference-description> in the future.

5.3.1. <conf-uris>

This element contains a sequence of <entry> child elements - each containing the URI to be used in order to access the conference by different signaling means. The value of the URI MUST be unique in the conference context and is included in the <uri> sub-element.

Each <entry> MAY contain additional information useful to the participant when accessing the conference.

An <entry> element MAY contain the <display-text> sub-element that provides a textual description meant for human consumption.

Each <entry> element SHOULD contain a <purpose> sub-element that describes what happens when accessing the URI. The currently defined <purpose> values to be used with the <conf-uris> are the following:

participation: Accessing a URI with this <purpose> will bring the party into the conference.

streaming: Accessing a URI with this <purpose> will commence streaming the conference, but not allow active participation.

Examples of suitable URI schemes include sip: and sips: [8], xmpp: [22], h323: [20], and tel: [19] URIs. The rtsp [18] URI is suitable for streaming.

Future extensions to this schema may define new values and register them with IANA under the registry established by this specification.

5.3.2. <service-uris>

This element describes auxiliary services available for the conference. Like <conference-uris>, this element contains a set of <entry> child elements - each containing the URI to be used in order to access different services available for the particular conference. The value of the URI MUST be unique in the conference context and is included in the <uri> sub-element.

An <entry> element MAY contain the <display-text> sub-element that provides a textual description meant for user consumption.

Each <entry> element SHOULD contain a <purpose> sub-element. The currently defined <purpose> values to be used with the <service-uris> are the following:

web-page: Indicates the web page containing the additional information about the conference.

recording: Indicates the link at which the recorded conference context can be retrieved.

event: Indicates the URI at which a subscription to the conference event package may be requested. This would typically be the conference URI of the main conference.

Future extensions to this schema may define new values and register them with IANA under the registry established by this specification.

5.3.3. <maximum-user-count>

The value of this element provides a hint to the recipient of the conference document about the number of users that can be invited to the conference. Typically, this value represents the overall number of users allowed to join the conference by different means as published through the conference document in <conf-uris>. Note that this value is set by an administrator and can reflect any local policies combination such as network consumption, CPU processing power, and licensing rules.

5.3.4. <available-media>

This element contains a sequence of <entry> child elements of conference-medium-type, each being indexed by the attribute 'label'.

The 'label' attribute is the media stream identifier assigned by the conferencing server: its value will be unique in the <conference-info> context. The value of this attribute will typically correspond to the Session Description Protocol (SDP) "label" media attribute defined in [17].

Each <entry> describes a single media stream available to the participants in the conference and contains the following information:

<display-text>: This element contains the display text for the media stream.

<type>: This element contains the media type of the media stream. The value of this element MUST be one of the values registered for "media" of SDP [3] and its later revision(s), for example, "audio", "video", "text", and "message".

<status>: This element indicates the available status of the media stream available to the conference participants. For example, this would be the status of the media stream, which would be offered by the focus, in a 'dial-out' scenario. Using normal SIP offer/answer mechanisms (being defined in RFC 3264 [9]) in both dial-in and dial-out scenarios, a participant can of course establish only a subset of the available stream (i.e., request or accept the stream in one direction only, if both directions are available). The valid values are "sendrecv", "sendonly", "recvonly", or "inactive" as defined in SDP [3] and its later

revision(s). (Note that the value specifies the direction from the participants' point of view.)

5.4. <host-info>

This element contains information about the entity hosting the conference. This information is set before the conference activation, and it is rarely changed during the conference lifetime, unless the whole conference is moved to be hosted by another entity. The host information is comprised of the following elements:

5.4.1. <display-text>

This element contains display text describing the entity hosting the conference.

5.4.2. <web-page>

This element contains HTTP: or HTTPS: URI of a web page describing either the conference service or the user hosting the conference.

5.4.3. <uris>

This element contains a set of <entry> child elements, each containing the URI value and optionally its description.

5.5. <conference-state>

By including this element in the conference document, the server can inform the subscribers about the changes in the overall conference information. The <conference-state> child elements are described below.

5.5.1. <user-count>

The value of this element tells the recipient of the conference document the overall number of users participating in the conference at a certain moment. Typically, this value represents the overall number of users who joined the conference by different means as published through the conference document in <conf-uris>. Note that this number does not necessarily need to match and MAY exceed the number of the entries in the <users> container. For example, in a lecturing scenario, large conference notifications may not include every participant in the <users> element, but instead report only the panelists or the speakers.

5.5.2. <active>

This Boolean element indicates whether the conference is currently active. A conference is active if calling one of the <conf-uris> by an authorized client results in successful establishment of a signaling session between the client and the focus and a successful joining of the conference.

5.5.3. <locked>

This Boolean element says whether the conference is currently locked. In this context, "locked" means that the conference roster cannot be added to (although participants may leave or be removed from the conference).

5.6. <users> and Its <user> Sub-elements

The <users> element is a container of <user> child elements, each describing a single participant in the conference.

The following attributes are defined for <user> element:

entity: This attribute contains the URI for the user in the conference. This is a logical identifier, which corresponds to the call signaling authenticated identity of the participant. The 'entity' value MUST be unique among all participants in the conference. If, for some participants, the focus decides not to reveal this information (e.g., due to local policies or security reasons), the host portion of the user URI MUST use the .invalid top level domain (TLD) according to definitions of [RFC 2606](#) [5]. The focus also MUST construct the user portion of the URI so that the URI is unique among all participants of the same domain. For example, the convention

"AnonymousX" <sip:anonymousX@anonymous.invalid>

SHOULD be used for a participant requesting privacy in accordance with the guidelines for generating anonymous URIs of [RFC 3323](#) [11]. Note that in a different case, such as when used in conjunction with Enhancements for Authenticated Identity Management in SIP [25], the following convention can be used:

"AnonymousX" <sip:anonymousX@example.com>

state: This attribute indicates whether the document contains the whole user information ("full") or only the information that has changed since the previous document ("partial"), or whether the user was removed from the conference ("deleted").

The following child elements are defined for <user> element:

5.6.1. <display-text>

This element is used to display the user-friendly name in the conference.

5.6.2. <associated-aors>

This element contains additional (to the 'entity') URIs being associated with the <user>. Typically, this information will be manually provided by an administrator showing the logical association between signaling entities otherwise independent. For example, if the 'entity' of a <user> contains a Globally Routable User URI (GRUU) [24] or tel: URI RFC 3966 [19], it would be useful to populate this field with the Address of Record (AOR) of the person who uses these devices, each represented as an independent <user>.

5.6.3. <roles>

This element MAY contain a set of human-readable strings describing the roles of the user in the conference. Note that this information is applicable for human consumption only. This specification does not define the set of possible conferencing roles or the semantics associated with each. It is expected that future conferencing specifications will define these and the corresponding schema extensions, as appropriate.

5.6.4. <languages>

This element contains a list of tokens, separated by spaces, each containing a language understood by the user. This information can be automatically learned via call signaling or be manually set per participant.

5.6.5. <cascaded-focus>

This element contains a conference URI (different from the main conference URI) for users that are connected to the main conference as a result of focus cascading. In accordance with the SIP Conferencing Framework [16], this package allows for representation of peer-to-peer (i.e., "flat") focus cascading only. The actual cascading graph cannot be deduced from the information provided in the package alone. Advanced applications can construct the graph by subscribing to both this package and the Dialog Package [23] of each cascaded focus and correlating the relevant information.

5.6.6. <endpoint>

By including one or more <endpoint> elements under a parent <user> element, the server can provide the desired level of detail (including the state, media streams, and access information) about the user's devices and signaling sessions taking part in the conference.

In a conferencing system where authentication is performed per endpoint (rather than per user), the focus can be unaware of the logical association of multiple endpoints under a common user. In this case, each endpoint will appear as a separate <user> with its own <endpoint> sub-element(s) in the conference document.

In a different case, the focus may choose to shield the information about the participant's multiple endpoints and signaling sessions from other subscribers altogether (e.g., due to privacy policies). To do so, the focus MAY aggregate the multiple signaling sessions' information under a single <endpoint> element. Note that in this case, the detailed call signaling information (represented by <call-info> sub-element) will not be included.

5.7. <endpoint>

This section describes the <endpoint> element in more detail.

The following attributes are defined for the <endpoint> element:

entity: The server MUST generate the 'entity' key for each <endpoint> element included under the parent <user>, such that its value is unique in the user context. In SIP terms, this can be the Contact URI, GRUU, etc.

state: This attribute indicates whether the element contains the whole endpoint information ("full") or only the information that has changed since the previous document ("partial"), or whether the endpoint has been removed from the conference ("deleted").

The following child elements are defined for the <endpoint> element:

5.7.1. <display-text>

This element contains the display text for the endpoint.

5.7.2. <referred>

This element contains information about the user whose action resulted in this endpoint being brought into the conference (e.g., the SIP user identified by this URI sent a REFER to the focus). It MAY contain the following sub-elements:

when: This element of the XML dateTime type contains the date and time that the endpoint was referred to the conference and SHOULD be expressed in Coordinated Universal Time (UTC) format. For example,

```
<when>2005-03-04T20:00:00Z</when>
```

reason: This element contains the reason the endpoint was referred to the conference. Including the information in the format defined by RFC 3326 [12] is RECOMMENDED. For example,

```
<reason>Reason: SIP;text="Ad-hoc Invitation"</reason>
```

by: This element contains the URI of the entity that caused the endpoint to be referred to the conference. In the case of SIP, it will be populated from the Referred-By header defined in RFC 3892 [15].

5.7.3. <status>

This element contains the status of the endpoint and can assume the following values:

connected: The endpoint is a participant in the conference. Depending on the media policies, he/she can send and receive media to and from other participants.

disconnected: The endpoint is not a participant in the conference, and no active dialog exists between the endpoint and the focus.

on-hold: Active signaling dialog exists between an endpoint and a focus, but endpoint is "on-hold" for this conference, i.e., he/she is neither "hearing" the conference mix nor is his/her media being mixed in the conference. As an example, the endpoint has asked to join the conference using SIP, but his/her participation is pending based on moderator approval. In the meantime, he/she is hearing music-on-hold or some other kind of related content.

muted-via-focus: Active signaling dialog exists between an endpoint and a focus and the endpoint can "listen" to the conference, but the endpoint's media is not being mixed into the conference. Note

that sometimes a subset of endpoint media streams can be muted by focus (such as poor-quality video) while others (such as voice or IM) can still be active. In this case, it is RECOMMENDED that the "aggregated" endpoint connectivity <status> reflects the status of the most active media.

pending: Endpoint is not yet in the session, but it is anticipated that he/she will join in the near future.

alerting: A Public Switched Telephone Network (PSTN) ALERTING or SIP 180 Ringing was returned for the outbound call; endpoint is being alerted.

dialing-in: Endpoint is dialing into the conference, not yet in the roster (probably being authenticated).

dialing-out: Focus has dialed out to connect the endpoint to the conference, but the endpoint is not yet in the roster (probably being authenticated).

disconnecting: Focus is in the process of disconnecting the endpoint (e.g., in SIP a DISCONNECT or BYE was sent to the endpoint).

Note that the defined transient statuses (e.g., disconnecting, alerting, etc.) could generate a lot of traffic. Therefore, implementations MAY choose to generate notifications on these statuses to certain participants only or not generate them at all, subject to local policy.

5.7.4. <joining-method>

This element contains the method by which the endpoint joined the conference and can assume the following values:

dialed-in: The endpoint dialed into the conference (e.g., in a SIP sent INVITE to the focus), which resulted in successful dialog establishment.

dialed-out: The focus has brought the endpoint into the conference (e.g., in SIP, the focus sent a successful INVITE to the endpoint).

focus-owner: The endpoint is the focus for this conference. This status is used only when a participant's UA acts as a conference focus.

5.7.5. <joining-info>

This element contains information about how the endpoint joined and MAY contain the following sub-elements:

when: This element of the XML dateTime type contains the date and time that the endpoint joined the conference and SHOULD be expressed in Coordinated Universal Time (UTC).

reason: This element contains the reason the endpoint joined the conference. Including the information in the format defined by RFC 3326 [12] is RECOMMENDED. For example,

```
<reason>Reason: SIP;text="Ad-hoc Invitation"</reason>
```

by: This element contains the URI of the entity that caused the endpoint to join the conference.

5.7.6. <disconnection-method>

This element contains the method by which the endpoint departed the conference and can assume the following values:

departed: In SIP, the endpoint sent a BYE, thus leaving the conference.

booted: In SIP, the endpoint was sent a BYE by the focus, ejecting him/her out of the conference. Alternatively, the endpoint tried to dial into the conference but was rejected by the focus due to local policy.

failed: In SIP, the server tried to bring the endpoint into the conference, but its attempt to contact the specific endpoint resulted in a non-200 class final response. Alternatively, the endpoint tried to dial into the conference without success due to technical reasons.

busy: In SIP, the server tried to bring the endpoint into the conference, but its attempt to contact the specific endpoint resulted in a 486 "Busy Here" final response. Alternatively, the endpoint tried to dial into the conference but the focus responded with 486 response.

5.7.7. <disconnection-info>

This element contains information about the endpoint's departure from the conference and MAY contain the following sub-elements:

when: This element of the XML dateTime type contains the date and time that the endpoint departed the conference and SHOULD be expressed in Coordinated Universal Time (UTC).

reason: This element contains the reason the endpoint departed the conference. When known and meaningful, including the information as conveyed/reported by the call signaling in the format defined by RFC 3326 [12] is RECOMMENDED. For example,

```
<reason>Reason: SIP;cause=415;text="Unsupported Media Type"</reason>
```

by: This element contains the URI of the entity that caused the endpoint to depart the conference.

5.7.8. <media>

This element contains information about a single media stream and is included for each media stream being established between the focus and the <endpoint>. The media stream definition can be found in SDP [3].

Note that if the <call-info> sub-element of the endpoint is not included in the document by the server, it is possible that the media streams listed under the common <endpoint> were established by separate signaling sessions.

5.7.9. <call-info>

The <call-info> element provides detailed call signaling information for a call being maintained between the participant and the focus. Privacy policies MUST be consulted before revealing this information to other participants.

The <sip> sub-element contains the SIP dialog identifier of the endpoint's dialog with the focus. The element includes sub-elements <display-text>, <call-id>, <to-tag>, <from-tag>.

In future, the <call-info> element can be expanded to include call signaling protocol information for other protocols besides SIP.

5.8. <media>

This section describes the <media> element in more detail.

A single 'id' attribute is defined for this element. This is the media stream identifier being generated by the server such that its

value is unique in the endpoint context. This attribute is the key to refer to a particular media stream in the conference document.

The following child elements are defined for <media>:

5.8.1. <display-text>

This element contains the display text for the media stream. The value of this element corresponds to the SDP description media attribute ("i") defined in SDP [3].

5.8.2. <type>

This element contains the media type for the media stream. The value of this element MUST be one of the values registered for "media" of SDP [3] and its later revision(s).

5.8.3. <label>

The <label> element carries a unique identifier for this stream among all streams in the conference and is assigned by the focus. The value of this element will typically correspond to the SDP "label" media attribute defined in [17] and is exchanged between a participant and a focus over the signaling connection between them.

If the <available-media> information (described in [Section 5.3.4](#)) is included in the conference document, the value of this element MUST be equal to the 'label' value of the corresponding media stream <entry> in the <available-media> container.

5.8.4. <src-id>

The <src-id> element, if applicable, carries the information about the actual source of the media. For example, for Real-time Transport Protocol (RTP) / RTP Control Protocol (RTCP) [13] media streams, the value MUST contain the synchronization source (SSRC) identifier value generated by the endpoint for the stream it sends.

When an RTP mixer generates a contributing source (CSRC) identifiers' list according to RTP/RTCP [13], it inserts a list of the SSRC identifiers of the sources that contributed to the generation of a particular packet into the RTP header of that packet. A quote from [RFC 3550](#) [13] explains as follows: "An example application is audio conferencing where a mixer indicates all the talkers whose speech was combined to produce the outgoing packet, allowing the receiver to indicate the current talker, even though all the audio packets contain the same SSRC identifier (that of the mixer)."

If an RTP mixer compliant to the above is used, participants can perform an SSRC to user mapping and identify "a current speaker".

5.8.5. <status>

The element <status> indicates the status in both directions of the media stream and has the values "sendrecv", "sendonly", "recvonly", or "inactive" as defined in SDP [3] and its later revision(s). Note that value specifies the direction from the participant's point of view. For example, a muted participant's stream will have the value of "recvonly".

5.9. Sidebars

If a participant in the main conference joins a sidebar, a new <user> element representing the user is created either as a part of a separate sub-conference referenced from the <sidebars-by-ref> element or under one of the <sidebars-by-val> elements as described below.

Note that the <user> in the main roster is not being deleted, but its media statuses can be updated to reflect the effect being caused by his/her participation in the sidebar. The display of this information can vary among subscribers to the same conference information, subject to local policies and to the subscriber role both in the sidebar and in the main conference.

5.9.1. <sidebars-by-ref>

This element contains a set of <entry> child elements, each containing a sidebar conference URI. The recipient of the information can then subscribe to sidebar information independently from the main conference package subscription.

5.9.2. <sidebars-by-val>

This element contains a set of <entry> child elements, each containing information about a single sidebar. By using this element of conference-type, the server can include a full or partial description of each sidebar (as a sub-conference) in the body of the main conference document.

6. XML Schema

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema
targetNamespace="urn:ietf:params:xml:ns:conference-info"
xmlns:tns="urn:ietf:params:xml:ns:conference-info"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
```

```
xmlns="urn:ietf:params:xml:ns:conference-info"
elementFormDefault="qualified"
attributeFormDefault="unqualified">
<!--
  This imports the xml:language definition
-->
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
    schemaLocation="http://www.w3.org/2001/03/xml.xsd"/>
<!--
  CONFERENCE ELEMENT
-->
  <xs:element name="conference-info" type="conference-type"/>
<!--
  CONFERENCE TYPE
-->
  <xs:complexType name="conference-type">
    <xs:sequence>
      <xs:element name="conference-description"
        type="conference-description-type" minOccurs="0"/>
      <xs:element name="host-info"
        type="host-type" minOccurs="0"/>
      <xs:element name="conference-state"
        type="conference-state-type" minOccurs="0"/>
      <xs:element name="users"
        type="users-type" minOccurs="0"/>
      <xs:element name="sidebars-by-ref"
        type="uris-type" minOccurs="0"/>
      <xs:element name="sidebars-by-val"
        type="sidebars-by-val-type" minOccurs="0"/>
      <xs:any namespace="##other" processContents="lax"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="entity"
      type="xs:anyURI" use="required"/>
    <xs:attribute name="state"
      type="state-type" use="optional" default="full"/>
    <xs:attribute name="version"
      type="xs:unsignedInt" use="optional"/>
    <xs:anyAttribute namespace="##other" processContents="lax"/>
  </xs:complexType>
<!--
  STATE TYPE
-->
  <xs:simpleType name="state-type">
    <xs:restriction base="xs:string">
      <xs:enumeration value="full"/>
      <xs:enumeration value="partial"/>
      <xs:enumeration value="deleted"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

```
</xs:restriction>
</xs:simpleType>
<!--
  CONFERENCE DESCRIPTION TYPE
-->
<xs:complexType name="conference-description-type">
  <xs:sequence>
    <xs:element name="display-text"
      type="xs:string" minOccurs="0"/>
    <xs:element name="subject"
      type="xs:string" minOccurs="0"/>
    <xs:element name="free-text"
      type="xs:string" minOccurs="0"/>
    <xs:element name="keywords"
      type="keywords-type" minOccurs="0"/>
    <xs:element name="conf-uris"
      type="uris-type" minOccurs="0"/>
    <xs:element name="service-uris"
      type="uris-type" minOccurs="0"/>
    <xs:element name="maximum-user-count"
      type="xs:unsignedInt" minOccurs="0"/>
    <xs:element name="available-media"
      type="conference-media-type" minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<!--
  HOST TYPE
-->
<xs:complexType name="host-type">
  <xs:sequence>
    <xs:element name="display-text" type="xs:string"
      minOccurs="0"/>
    <xs:element name="web-page" type="xs:anyURI"
      minOccurs="0"/>
    <xs:element name="uris" type="uris-type"
      minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<!--
  CONFERENCE STATE TYPE
-->
<xs:complexType name="conference-state-type">
```

```
<xs:sequence>
  <xs:element name="user-count" type="xs:unsignedInt"
    minOccurs="0"/>
  <xs:element name="active" type="xs:boolean"
    minOccurs="0"/>
  <xs:element name="locked" type="xs:boolean"
    minOccurs="0"/>
  <xs:any namespace="##other" processContents="lax"
    minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<!--
  CONFERENCE MEDIA TYPE
-->
<xs:complexType name="conference-media-type">
  <xs:sequence>
    <xs:element name="entry" type="conference-medium-type"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<!--
  CONFERENCE MEDIUM TYPE
-->
<xs:complexType name="conference-medium-type">
  <xs:sequence>
    <xs:element name="display-text" type="xs:string"
      minOccurs="0"/>
    <xs:element name="type" type="xs:string"/>
    <xs:element name="status" type="media-status-type"
      minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="label" type="xs:string"
    use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<!--
  URIs TYPE
-->
<xs:complexType name="uris-type">
  <xs:sequence>
    <xs:element name="entry" type="uri-type"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="state" type="state-type"
```

```
        use="optional" default="full"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
    <!--
      URI TYPE
    -->
    <xs:complexType name="uri-type">
      <xs:sequence>
        <xs:element name="uri" type="xs:anyURI"/>
        <xs:element name="display-text" type="xs:string"
          minOccurs="0"/>
        <xs:element name="purpose" type="xs:string"
          minOccurs="0"/>
        <xs:element name="modified" type="execution-type"
          minOccurs="0"/>
        <xs:any namespace="##other" processContents="lax"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
    <!--
      KEYWORDS TYPE
    -->
    <xs:simpleType name="keywords-type">
      <xs:list itemType="xs:string"/>
    </xs:simpleType>
    <!--
      USERS TYPE
    -->
    <xs:complexType name="users-type">
      <xs:sequence>
        <xs:element name="user" type="user-type"
          minOccurs="0" maxOccurs="unbounded"/>
        <xs:any namespace="##other" processContents="lax"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="state" type="state-type"
        use="optional" default="full"/>
      <xs:anyAttribute namespace="##other" processContents="lax"/>
    </xs:complexType>
    <!--
      USER TYPE
    -->
    <xs:complexType name="user-type">
      <xs:sequence>
        <xs:element name="display-text" type="xs:string"
          minOccurs="0"/>
        <xs:element name="associated-aors" type="uris-type"
```

```
    minOccurs="0"/>
<xs:element name="roles" type="user-roles-type"
  minOccurs="0"/>
<xs:element name="languages" type="user-languages-type"
  minOccurs="0"/>
<xs:element name="cascaded-focus" type="xs:anyURI"
  minOccurs="0"/>
<xs:element name="endpoint" type="endpoint-type"
  minOccurs="0" maxOccurs="unbounded"/>
<xs:any namespace="##other" processContents="lax"
  minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="entity" type="xs:anyURI"/>
<xs:attribute name="state" type="state-type"
  use="optional" default="full"/>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<!--
  USER ROLES TYPE
-->
<xs:complexType name="user-roles-type">
  <xs:sequence>
    <xs:element name="entry" type="xs:string"
      maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<!--
  USER LANGUAGES TYPE
-->
<xs:simpleType name="user-languages-type">
  <xs:list itemType="xs:language"/>
</xs:simpleType>
<!--
  ENDPOINT TYPE
-->
<xs:complexType name="endpoint-type">
  <xs:sequence>
    <xs:element name="display-text" type="xs:string"
      minOccurs="0"/>
    <xs:element name="referred" type="execution-type"
      minOccurs="0"/>
    <xs:element name="status" type="endpoint-status-type"
      minOccurs="0"/>
    <xs:element name="joining-method" type="joining-type"
      minOccurs="0"/>
    <xs:element name="joining-info"
      type="execution-type"/>
```

```
    minOccurs="0"/>
<xs:element name="disconnection-method"
  type="disconnection-type"
  minOccurs="0"/>
<xs:element name="disconnection-info"
  type="execution-type"
  minOccurs="0"/>
<xs:element name="media" type="media-type"
  minOccurs="0" maxOccurs="unbounded"/>
<xs:element name="call-info" type="call-type"
  minOccurs="0"/>
<xs:any namespace="##other" processContents="lax"
  minOccurs="0" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute name="entity" type="xs:string"/>
<xs:attribute name="state" type="state-type"
  use="optional" default="full"/>
<xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<!--
  ENDPOINT STATUS TYPE
-->
<xs:simpleType name="endpoint-status-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="pending"/>
    <xs:enumeration value="dialing-out"/>
    <xs:enumeration value="dialing-in"/>
    <xs:enumeration value="alerting"/>
    <xs:enumeration value="on-hold"/>
    <xs:enumeration value="connected"/>
    <xs:enumeration value="muted-via-focus"/>
    <xs:enumeration value="disconnecting"/>
    <xs:enumeration value="disconnected"/>
  </xs:restriction>
</xs:simpleType>
<!--
  JOINING TYPE
-->
<xs:simpleType name="joining-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="dialed-in"/>
    <xs:enumeration value="dialed-out"/>
    <xs:enumeration value="focus-owner"/>
  </xs:restriction>
</xs:simpleType>
<!--
  DISCONNECTION TYPE
-->
```



```
<xs:simpleType name="disconnection-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="departed"/>
    <xs:enumeration value="booted"/>
    <xs:enumeration value="failed"/>
    <xs:enumeration value="busy"/>
  </xs:restriction>
</xs:simpleType>
<!--
  EXECUTION TYPE
-->
<xs:complexType name="execution-type">
  <xs:sequence>
    <xs:element name="when" type="xs:dateTime"
      minOccurs="0"/>
    <xs:element name="reason" type="xs:string"
      minOccurs="0"/>
    <xs:element name="by" type="xs:anyURI"
      minOccurs="0"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<!--
  CALL TYPE
-->
<xs:complexType name="call-type">
  <xs:choice>
    <xs:element name="sip" type="sip-dialog-id-type"/>
    <xs:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:choice>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<!--
  SIP DIALOG ID TYPE
-->
<xs:complexType name="sip-dialog-id-type">
  <xs:sequence>
    <xs:element name="display-text" type="xs:string"
      minOccurs="0"/>
    <xs:element name="call-id" type="xs:string"/>
    <xs:element name="from-tag" type="xs:string"/>
    <xs:element name="to-tag" type="xs:string"/>
    <xs:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
```

```
<!--
  MEDIA TYPE
-->
<xs:complexType name="media-type">
  <xs:sequence>
    <xs:element name="display-text" type="xs:string"
      minOccurs="0"/>
    <xs:element name="type" type="xs:string"
      minOccurs="0"/>
    <xs:element name="label" type="xs:string"
      minOccurs="0"/>
    <xs:element name="src-id" type="xs:string"
      minOccurs="0"/>
    <xs:element name="status" type="media-status-type"
      minOccurs="0"/>
    <xs:any namespace="##other" processContents="lax"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="id" type="xs:string"
    use="required"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
<!--
  MEDIA STATUS TYPE
-->
<xs:simpleType name="media-status-type">
  <xs:restriction base="xs:string">
    <xs:enumeration value="recvonly"/>
    <xs:enumeration value="sendonly"/>
    <xs:enumeration value="sendrecv"/>
    <xs:enumeration value="inactive"/>
  </xs:restriction>
</xs:simpleType>
<!--
  SIDEBARS BY VAL TYPE
-->
<xs:complexType name="sidebars-by-val-type">
  <xs:sequence>
    <xs:element name="entry" type="conference-type"
      minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="state" type="state-type"
    use="optional" default="full"/>
  <xs:anyAttribute namespace="##other" processContents="lax"/>
</xs:complexType>
</xs:schema>
```

7. Examples

7.1. Basic Example

The following is an example of a full conference information document:

```
<?xml version="1.0" encoding="UTF-8"?>
<conference-info
  xmlns="urn:ietf:params:xml:ns:conference-info"
  entity="sips:conf233@example.com"
  state="full" version="1">
  <!--
    CONFERENCE INFO
  -->
  <conference-description>
    <subject>Agenda: This month's goals</subject>
    <service-uris>
      <entry>
        <uri>http://sharepoint/salesgroup</uri>
        <purpose>web-page</purpose>
      </entry>
    </service-uris>
  </conference-description>
  <!--
    CONFERENCE STATE
  -->
  <conference-state>
    <user-count>33</user-count>
  </conference-state>
  <!--
    USERS
  -->
  <users>
    <user entity="sip:bob@example.com" state="full">
      <display-text>Bob Hoskins</display-text>
    </user>
  </users>
  <!--
    ENDPOINTS
  -->
  <endpoint entity="sip:bob@pc33.example.com">
    <display-text>Bob's Laptop</display-text>
    <status>disconnected</status>
    <disconnection-method>departed</disconnection-method>
    <disconnection-info>
      <when>2005-03-04T20:00:00Z</when>
      <reason>bad voice quality</reason>
      <by>sip:mike@example.com</by>
    </disconnection-info>
  </endpoint>
</conference-info>
```

```

<!--
  MEDIA
-->
  <media id="1">
    <display-text>main audio</display-text>
    <type>audio</type>
    <label>34567</label>
    <src-id>432424</src-id>
    <status>sendrecv</status>
  </media>
</endpoint>
</user>
<!--
  USER
-->
  <user entity="sip:alice@example.com" state="full">
    <display-text>Alice</display-text>
<!--
  ENDPOINTS
-->
  <endpoint entity="sip:4kfk4j392jsu@example.com;grid=433kj4j3u">
    <status>connected</status>
    <joining-method>dialed-out</joining-method>
    <joining-info>
      <when>2005-03-04T20:00:00Z</when>
      <by>sip:mike@example.com</by>
    </joining-info>
<!--
  MEDIA
-->
  <media id="1">
    <display-text>main audio</display-text>
    <type>audio</type>
    <label>34567</label>
    <src-id>534232</src-id>
    <status>sendrecv</status>
  </media>
</endpoint>
</user>
</users>
</conference-info>

```

7.2. Rich Example

The following is an example of a partial conference information document. In this example, there are 32 participants in a voice conference. The user Bob has been ejected from the conference by Mike due to bad voice quality. Note that there are three sidebars in

the conference; two are referenced just by their sidebar URIs, and information about the third sidebar is included in this notification. Also note that while this conference offers both audio and video capabilities, only audio is currently in use.

```
<?xml version="1.0" encoding="UTF-8"?>
<conference-info
  xmlns="urn:ietf:params:xml:ns:conference-info"
  entity="sips:conf233@example.com"
  state="partial" version="5">
  <!--
    CONFERENCE INFO
  -->
  <conference-description>
    <display-text>Weekly Sales Meeting</display-text>
    <subject>Agenda: This month's goals</subject>
    <free-text>We will start strict on time</free-text>
    <keywords>sales meeting weekly</keywords>
    <conf-uris>
      <entry>
        <uri>tel:+18005671234</uri>
        <display-text>TTI Bridge</display-text>
        <purpose>participation</purpose>
      </entry>
      <entry>
        <uri>h323:conf545@h323.example.com</uri>
        <purpose>participation</purpose>
      </entry>
      <entry>
        <uri>http://real.streaming.com/54634/live.ram</uri>
        <purpose>streaming</purpose>
      </entry>
    </conf-uris>
    <service-uris>
      <entry>
        <uri>http://sharepoint/salesgroup/</uri>
        <purpose>web-page</purpose>
      </entry>
      <entry>
        <uri>http://quicktime.com/54634/recording.mov</uri>
        <display-text>Quicktime</display-text>
        <purpose>recording</purpose>
      </entry>
    </service-uris>
    <maximum-user-count>100</maximum-user-count>
    <available-media>
      <entry label="34567">
        <display-text>main audio</display-text>
```

```
<type>audio</type>
<status>sendrecv</status>
</entry>
<entry label="34569">
  <display-text>main video</display-text>
  <type>video</type>
  <status>inactive</status>
</entry>
</available-media>
</conference-description>
<!--
  HOST INFO
-->
<host-info>
  <display-text>Sales Host</display-text>
  <web-page>http://sharepoint/salesgroup/hosts/</web-page>
  <uris>
    <entry>
      <uri>sip:sales@example.com</uri>
    </entry>
  </uris>
</host-info>
<!--
  CONFERENCE STATE
-->
<conference-state>
  <user-count>32</user-count>
  <active>true</active>
  <locked>false</locked>
</conference-state>
<!--
  USERS
-->
<users>
  <user entity="sip:bob@example.com">
    <display-text>Bob Hoskins</display-text>
    <associated-aors>
      <entry>
        <uri>mailto:bob@example.com</uri>
        <display-text>email</display-text>
      </entry>
    </associated-aors>
    <roles>
      <entry>participant</entry>
    </roles>
    <languages>en</languages>
  </user>
</users>
<!--
  ENDPOINTS
```

```

-->
<endpoint entity="sip:bob@pc33.example.com">
  <display-text>Bob's Laptop</display-text>
  <referred>
    <when>2005-03-04T20:00:00Z</when>
    <reason>expert required</reason>
    <by>sip:mike@example.com</by>
  </referred>
  <status>disconnecting</status>
  <joining-method>dialled-out</joining-method>
  <joining-info>
    <when>2005-03-04T20:00:00Z</when>
    <reason>invitation</reason>
    <by>sip:mike@example.com</by>
  </joining-info>
  <disconnection-method>booted</disconnection-method>
  <disconnection-info>

    <when>2005-03-04T20:00:00Z</when>
    <reason>bad voice quality</reason>
    <by>sip:mike@example.com</by>
  </disconnection-info>
<!--
  MEDIA
-->
  <media id="1">
    <display-text>main audio</display-text>
    <type>audio</type>
    <label>34567</label>
    <src-id>432424</src-id>
    <status>sendrecv</status>
  </media>
<!--
  CALL INFO
-->
  <call-info>
    <sip>
      <display-text>full info</display-text>
      <call-id>hsjh8980vhsb78</call-id>
      <from-tag>vav738dvbs</from-tag>
      <to-tag>8954jggjg8432</to-tag>
    </sip>
  </call-info>
</endpoint>
</user>
</users>
<!--
  SIDEBARS BY REFERENCE

```

```
-->
<sidebars-by-ref state="partial">
  <entry>
    <uri>sips:conf233@example.com;grid=45</uri>
    <display-text>sidebar with Carol</display-text>
  </entry>
  <entry>
    <uri>sips:conf233@example.com;grid=21</uri>
    <display-text>private with Peter</display-text>
  </entry>
</sidebars-by-ref>
<!--
  SIDEBARS BY VALUE
-->
<sidebars-by-val state="partial">
  <entry entity="sips:conf233@example.com;grid=77"

    state="partial">
      <users>
        <user entity="sip:bob@example.com"/>
        <user entity="sip:mark@example.com"/>
        <user entity="sip:dan@example.com"/>
      </users>
    </entry>
  </sidebars-by-val>
</conference-info>
```

8. Security Considerations

Subscriptions to conference state information can reveal very sensitive information. For this reason, it is RECOMMENDED that a focus use a strong means for authentication and conference information protection and that it apply comprehensive authorization rules when using the conference notification mechanism defined in this document. The following sections will discuss each of these aspects in more detail.

8.1. Connection Security

It is RECOMMENDED that a focus authenticate a conference package subscriber using the normal SIP authentication mechanisms, such as Digest as defined in [Section 22 of RFC 3261](#) [8].

The mechanism used for conveying the conference information MUST ensure integrity and SHOULD ensure confidentiality of the information. In order to achieve these, an end-to-end SIP encryption mechanism,

such as S/MIME described in [Section 26.2.4 of RFC 3261 \[8\]](#), SHOULD be used.

If a strong end-to-end security means (such as above) is not available, it is RECOMMENDED that a focus use mutual hop-by-hop Transport Layer Security (TLS) authentication and encryption mechanisms described in [Section 26.2.2 "SIPS URI Scheme"](#) and [Section 26.3.2.2 "Interdomain Requests"](#) of [RFC 3261 \[8\]](#).

8.2. Authorization Considerations

Generally speaking, conference applications are very concerned about authorization decisions. Mechanisms for establishing and enforcing such authorization rules are a central concept throughout the SIP Conferencing Framework [\[16\]](#). Because most of the information about a conference can be presented using the conference package, many of the authorization rules directly apply to this specification. As a result, a notification server MUST be capable of generating distinct conference information views to different subscribers, subject to a subscriber's role in a conference, personal access rights, etc. - all subject to local authorization policies and rules.

Since a focus provides participant identity information using this event package, participant privacy needs to be taken into account. A focus MUST support requests by participants for privacy. Privacy can be indicated by the conference policy - for every participant or select participants. It can also be indicated in the session signaling. In SIP, this can be done using the Privacy header field described in [RFC 3323 \[11\]](#). For a participant requesting privacy, no identity information SHOULD be revealed by the focus in any included URI (e.g., the Address of Record, Contact, or GRUU). For these cases, the anonymous URI generation method outlined in [Section 5.6](#) of this document MUST be followed.

9. IANA Considerations

This document registers a SIP event package, a new MIME type, application/conference-info+xml, a new XML namespace, and a new XML schema, and creates a sub-registry "URI purposes" under the existing registry: <http://www.iana.org/assignments/sip-parameters>.

9.1. conference Event Package Registration

This specification registers an event package, based on the registration procedures defined in [RFC 3265 \[10\]](#). The following is the information required for such a registration:

Package Name: conference

Package or Template-Package: This is a package.

Published Document: [RFC 4575](#)

Person to Contact: IETF SIPPING Working Group <sipping@ietf.org>, as designated by the IESG <iesg@ietf.org>

9.2. application/conference-info+xml MIME Registration

MIME media type name: application

MIME subtype name: conference-info+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 8](#) of this specification

Interoperability considerations: none

Published specification: This document

Applications which use this media type: This document type has been used to support SIP conferencing applications

Additional Information:

Magic Number: None

File Extension: .xml

Macintosh file type code: "TEXT"

Personal and email address for further information: IETF SIPPING Working Group <sipping@ietf.org>, as designated by the IESG <iesg@ietf.org>

Intended usage: COMMON

Author/Change controller: IETF SIPING Working Group
<sipping@ietf.org>, as designated by the IESG <iesg@ietf.org>

9.3. URN Sub-Namespace Registration for urn:ietf:params:xml:ns:conference-info

This section registers a new XML namespace, as per the guidelines in [RFC 3688](#) [21].

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

Registrant Contact: IETF SIPING Working Group <sipping@ietf.org>,
as designated by the IESG <iesg@ietf.org>

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 Information Namespace</title>

</head>
<body>
  <h1>Namespace for Conference Information</h1>
  <h2>urn:ietf:params:xml:ns:conference-info</h2>
  <p>See <a href="http://www.rfc-editor.org/rfc/rfc4575.txt">
    RFC4575</a>.</p>
</body>
</html>
END
```

9.4. XML Schema Registration

This specification registers a schema, as per the guidelines in [RFC 3688](#) [21].

URI: please assign

Registrant Contact: IETF SIPING Working Group <sipping@ietf.org>,
as designated by the IESG <iesg@ietf.org>

XML: The XML can be found as the sole content of [Section 6](#).

9.5. URI Purposes Sub-registry Establishment

The IANA has created a new sub-registry, "URI purposes", under the already existing registry:

<http://www.iana.org/assignments/sip-parameters>.

The purpose of a URI is an XML element, encoded in the conference event package [RFC 4575](#). The value of the <purpose> element indicates the intended usage of the URI in the context of the conference event package and is defined in Sections [5.3.1](#) and [5.3.2](#) of this specification.

This sub-registry is defined as a table that contains the following three columns:

Value: The token under registration

Description: A descriptive text defining the intended usage of the URI

Document: A reference to the document defining the registration

The IANA has created the table with the initial content as defined below:

Value	Description	Document
-----	-----	-----
participation	The URI can be used to join the conference	[RFC 4575]
streaming	The URI can be used to access the streamed conference data	[RFC 4575]
event	The URI can be used to subscribe to the conference event package	[RFC 4575]
recording	The URI can be used to access the recorded conference data	[RFC 4575]
web-page	The URI can be used to access a web page that contains additional information of the conference	[RFC 4575]

New values of the "URI purposes" are registered by the IANA and are specification required according to the definition of [RFC 2434](#) [4]. The IANA Considerations section of the specification MUST include the following information:

Value: The value of the <purpose> element to be registered

Description: A short description of the intended usage of the URI

10. Acknowledgements

The authors would like to thank Dan Petrie, Sean Olson, Alan Johnston, Rohan Mahy, Cullen Jennings, Brian Rosen, Roni Even, and Miguel Garcia for their comments and inputs.

11. References

11.1. Normative References

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [2] Moats, R., "URN Syntax", [RFC 2141](#), May 1997.
- [3] Handley, M., Jacobson, V. and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [4] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", [BCP 26](#), [RFC 2434](#), October 1998.
- [5] Eastlake 3rd, D. and A. Panitz, "Reserved Top Level DNS Names", [BCP 32](#), [RFC 2606](#), June 1999.
- [6] Moats, R., "A URN Namespace for IETF Documents", [RFC 2648](#), August 1999.
- [7] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", [RFC 3023](#), January 2001.
- [8] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.
- [9] Rosenberg, J. and H. Schulzrinne, "An Offer/Answer Model with Session Description Protocol (SDP)", [RFC 3264](#), June 2002.
- [10] Roach, A.B., "Session Initiation Protocol (SIP)-Specific Event Notification", [RFC 3265](#), June 2002.
- [11] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", [RFC 3323](#), November 2002.

- [12] Schulzrinne, H., Oran, D., and G. Camarillo, "The Reason Header Field for the Session Initiation Protocol (SIP)", [RFC 3326](#), December 2002.
- [13] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, [RFC 3550](#), July 2003.
- [14] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, [RFC 3629](#), November 2003.
- [15] Sparks, R., "The Session Initiation Protocol (SIP) Referred-By Mechanism", [RFC 3892](#), September 2004.
- [16] Rosenberg, J., "A Framework for Conferencing with the Session Initiation Protocol (SIP)", [RFC 4353](#), February 2006.
- [17] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", [RFC 4574](#), August 2006.

11.2. Informative References

- [18] Schulzrinne, H., Rao, A., and R. Lanphier, "Real Time Streaming Protocol (RTSP)", [RFC 2326](#), April 1998.
- [19] Schulzrinne, H., "The tel URI for Telephone Numbers", [RFC 3966](#), December 2004.
- [20] Levin, O., "H.323 Uniform Resource Locator (URL) Scheme Registration", [RFC 3508](#), April 2003.
- [21] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), January 2004.
- [22] Saint-Andre, P., "A Uniform Resource Identifier (URI) Scheme for the Extensible Messaging and Presence Protocol (XMPP)", Work in Progress, December 2004.
- [23] Rosenberg, J., Schulzrinne, H., and R. Mahy, "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", [RFC 4235](#), November 2005.
- [24] Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", Work in Progress, May 2006.

- [25] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", [RFC 4474](#), August 2006.

Authors' Addresses

Jonathan Rosenberg
Cisco Systems
600 Lanidex Plaza
Parsippany, NJ 07054
US

Phone: +1 973 952-5000
EMail: jdrosen@cisco.com
URI: <http://www.jdrosen.net>

Henning Schulzrinne
Columbia University
M/S 0401
1214 Amsterdam Ave.
New York, NY 10027
US

EMail: schulzrinne@cs.columbia.edu
URI: <http://www.cs.columbia.edu/~hgs>

Orit Levin (editor)
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052
US

EMail: oritl@microsoft.com

Full Copyright Statement

Copyright (C) The Internet Society (2006).

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.

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.

Intellectual Property

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.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).