SIPREC
Internet-Draft
Intended status: Standards Track
Expires: September 19, 2016

R. Ravindranath
Cisco Systems
Parthasarathi. Ravindran
Nokia Networks
Paul. Kyzivat
Huawei

March 18, 2016

Session Initiation Protocol (SIP) Recording Metadata
draft-ietf-siprec-metadata-21

Abstract

Session recording is a critical requirement in many communications environments such as call centers and financial trading. In some of these environments, all calls must be recorded for regulatory, compliance, and consumer protection reasons. Recording of a session is typically performed by sending a copy of a media stream to a recording device. This document describes the metadata model as viewed by Session Recording Server(SRS) and the Recording metadata format.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on September 19, 2016.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Terminology 3
3. Definitions 3
4. Metadata Model 4
5. Recording metadata format from SRC to SRS 6
5.1. XML data format 6
5.1.1. Namespace 7
5.1.2. recording 7
6. Recording metadata classes 7
6.1. Recording Session 7
6.1.1. Attributes 8
6.1.2. Linkages 8
6.2. Communication Session Group 9
6.2.1. Attributes 9
6.2.2. Linkages 10
6.3. Communication Session 10
6.3.1. Attributes 11
6.3.2. Linkages 12
6.4. CSRSAssociation 13
6.4.1. Attributes 13
6.4.2. Linkages 13
6.5. Participant 14
6.5.1. Attributes 14
6.5.2. Linkages 15
6.6. ParticipantCSAssociation 15
6.6.1. Attributes 16
6.6.2. Linkages 17
6.7. Media Stream 17
6.7.1. Attributes 18
6.7.2. Linkages 18
6.8. ParticipantStreamAssociation 18
6.8.1. Attributes 19
6.8.2. Linkages 20
6.9. Syntax of date/time XML elements 20
6.10. Unique ID format 20
6.11. Metadata version Indicator 21
7. Recording metadata snapshot request format 21
8. SIP Recording Metadata Example 22
8.1. Complete SIP Recording Metadata Example 22

8.2. Partial Update of Recording metadata XML body 24
9. XML Schema definition for Recording metadata 24
10. Security Considerations 28
11. IANA Considerations 29
11.1. SIP recording metadata Schema Registration 29
12. Acknowledgement 29
13. References 29
13.1. Normative References 29
13.2. Informative References 30
Authors' Addresses 31

1. Introduction

Session recording is a critical requirement in many communications environments such as call centers and financial trading. In some of these environments, all calls must be recorded for regulatory, compliance, and consumer protection reasons. Recording of a session is typically performed by sending a copy of a media stream to a recording device. This document focuses on the Recording metadata which describes the communication session. The document describes a metadata model as viewed by Session Recording Server(SRS) and the Recording metadata format, the requirements for which are described in [RFC6341] and the architecture for which is described in [RFC7245].

2. Terminology

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 [RFC2119]. This document only uses these key words when referencing normative statements in existing RFCs."

3. Definitions

Metadata model: An abstract representation of metadata using a Unified Modelling Language (UML) [UML-REF] class diagram.

Metadata classes: Each block in the model represents a class. A class is a construct that is used as a blueprint to create instances(called objects) of itself. The description of each class also has representation of its attributes in a second compartment below the class name.

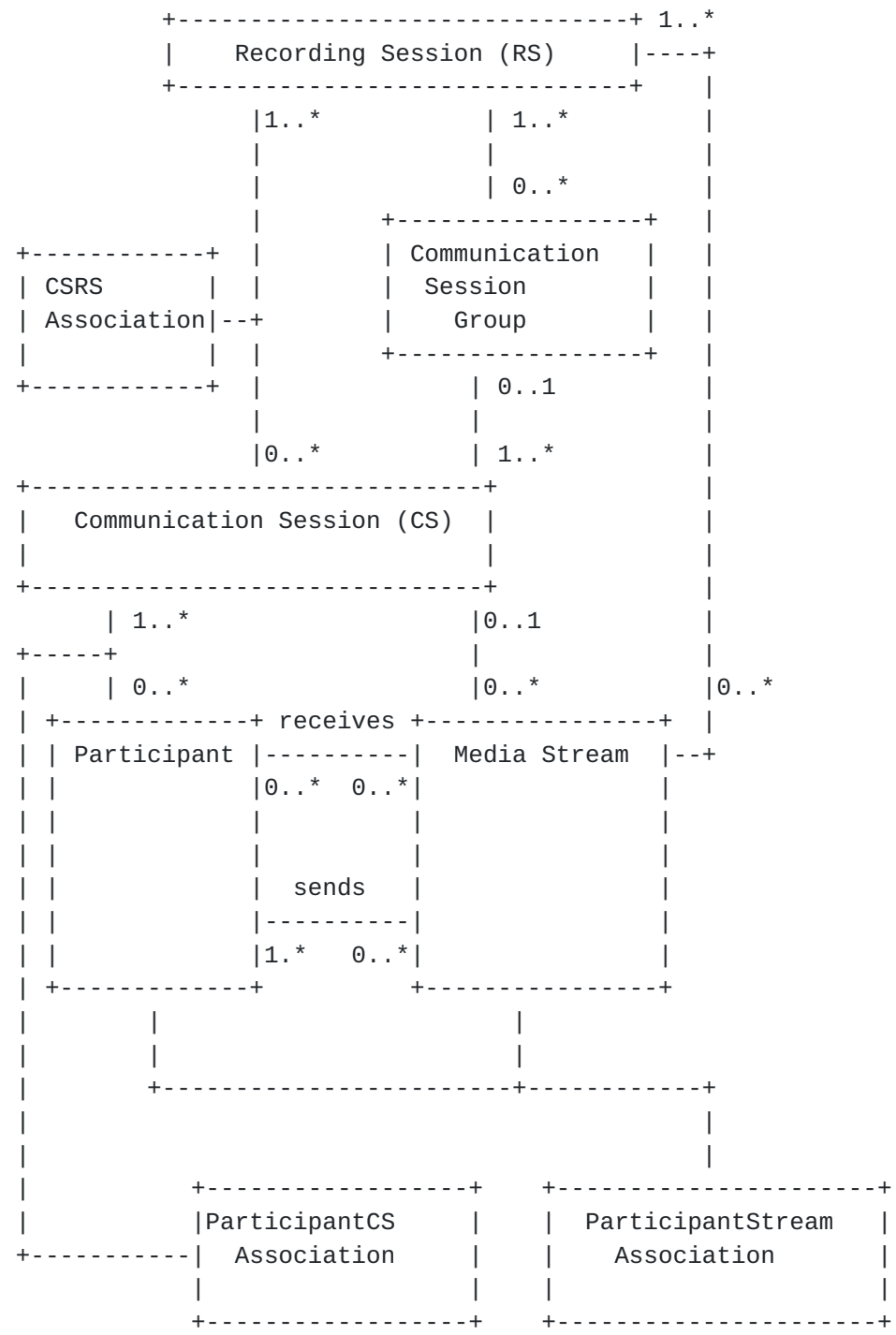
Attributes: Attributes represent the elements listed in each of the classes. The attributes of a class are listed in the second compartment below the class name. Each instance of class conveys values for these attributes which adds to the recording's metadata.

Linkages: Linkages represent the relationship between the classes in the model. Each represents a logical connection between classes(or objects) in class diagrams(or object diagrams). The linkages used in the metadata model of this document are associations.

This document also refers to the terminlogy defined in [RFC6341].

4. Metadata Model

Metadata is the information that describes recorded media and the Communication Session(CS) to which they relate. The diagram below shows a model for metadata as viewed by a SRS.



The metadata model is a class diagram in UML. The model describes the structure of metadata in general by showing the classes, their attributes, and the relationships among the classes. Each block in the model above represents a class. The linkages between the classes

represent the relationships which can be associations or composition. The metadata is conveyed from SRC to SRS.

The model allows metadata describing communications sessions to be communicated to the SRS as a series of snapshots, each representing the state as seen by a single SRC at a particular instant in time. Metadata changes from one snapshot to another reflect changes in what is being recorded. For example, if a participant joins a conference, then the SRC sends the SRS a snapshot of metadata having that participant information (with attributes like name/AoR pair and associate-time.)

Some of the metadata is not required to be conveyed explicitly from the SRC to the SRS, if it can be obtained contextually by the SRS(e.g., from SIP or SDP signalling). For example, the label attribute within the 'stream' XML element references an SDP 'a=label' attribute that identifies an m-line within the Recording Session(RS) SDP. The SRS would learn the media properties from media line.

5. Recording metadata format from SRC to SRS

This section gives an overview of the Recording metadata format. Some data from the metadata model is assumed to be made available to the SRS through Session Description Protocol (SDP)[RFC4566], and therefore this data is not represented in the XML document format specified in this document. SDP attributes describe different media formats like audio, video. The other metadata attributes, such as participant details, are represented in a new recording specific XML document of type 'application/rs-metadata+xml'. The SDP label attribute [RFC4574] provides an identifier by which a metadata XML document can refer to a specific media description in the SDP sent from the SRC to the SRS.

The XML document format can be used to represent either the complete metadata or a partial update to the metadata. The latter includes only elements that have changed compared to the previously reported metadata.

5.1. XML data format

Every recording metadata XML document sent from SRC to SRS contain a'recording' element. The 'recording' element acts as a container for all other elements in this XML document. A 'recording' object is an XML document. It has the XML declaration and contain an encoding declaration in the XML declaration, e.g., "<?xml version='1.0' encoding='UTF-8'?>". If the charset parameter of the MIME content type declaration is present and it is different from the encoding declaration, the charset parameter takes precedence.

Every application conforming to this specification MUST accept the UTF-8 character encoding to ensure the minimal interoperability.

Syntax and semantic errors in an XML document should be reported to the originator using application specific mechanisms.

5.1.1. Namespace

This document defines a new namespace URI for elements defined by this specification is the following URN:

urn:ietf:params:xml:ns:recording:1

5.1.2. recording

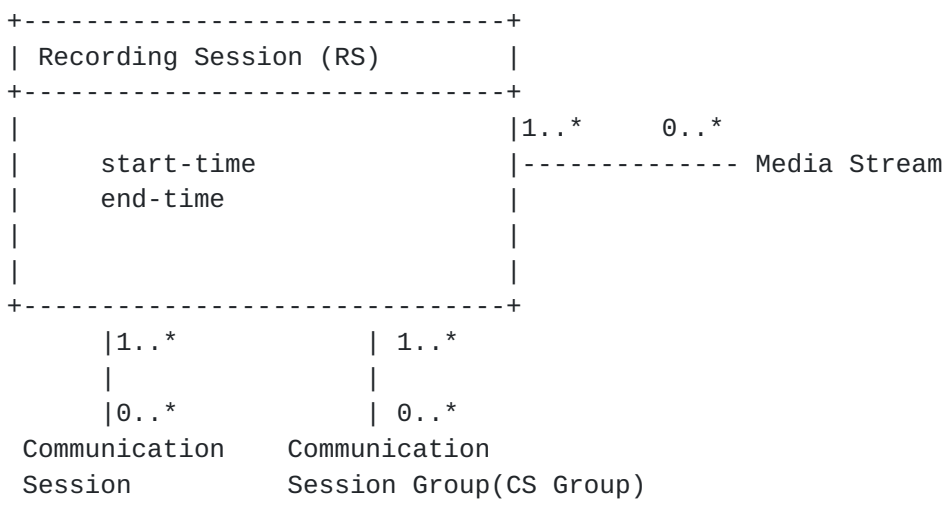
The 'recording' element MUST contain an xmlns namespace attribute with value as urn:ietf:params:xml:ns:recording:1. Exactly one recording element MUST be present in every recording metadata XML document.

A recording element MAY contain a <dataMode> element indicating whether the XML document is a complete document or a partial update. If no <dataMode> element is present then the default value is "complete".

6. Recording metadata classes

This section describes each class of the metadata model, and the attributes of each class. This section also describes how different classes are linked and the XML element for each of them.

6.1. Recording Session



Each instance of a Recording Session(RS) class namely the RS Object represents a SIP session created between an SRC and SRS for the purpose of recording a Communication Session(CS).

RS object is represented in XML schema using 'recording' element. That in turn relies on the SIP/SDP session with which the XML document is associated to provide the attributes of the RS element.

6.1.1. Attributes

An RS class has the following attributes:

- start-time - Represents the start time of an RS object.

- end-time - Represents the end time of an RS object.

'start-time' and 'end-time' attribute values are derivable from Date header(if present in SIP message) in RS. In cases where Date header is not present, 'start-time' is derivable from the time at which SRS receives the notification of SIP message to setup RS and and 'end-time' is derivable from the time at which SRS receives disconnect on the RS SIP dialog.

6.1.2. Linkages

Each instance of RS has:

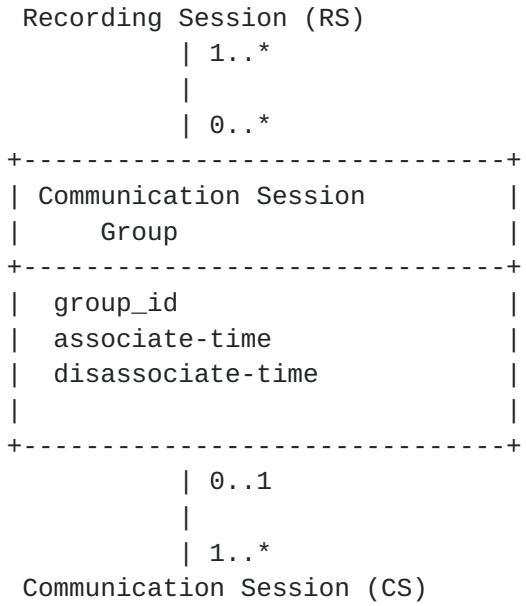
- Zero or more instances of Communication Session Group (CS-Group).

- Zero or more instances of CS objects.

- Zero or more instances of MediaStream objects.

CSs and CS-Groups are optional to accommodate persistent recording, where there may sometimes be none. A persistent RS is a SIP dialog that is setup between SRC to SRS even before any CS is setup. The metadata sent from SRC to SRS when the persistent RS SIP dialog is setup may not have any CS(and the related CS-Group) elements in the XML as there may be no session that is associated to the RS yet. For e.g; a phone acting as SRC can setup a RS with SRS possibly even before phone is part of a CS. Once the phone joins a CS, the same RS would be used to convey the CS metadata.

6.2. Communication Session Group



One instance of a Communication Session Group(CS-Group) class namely the CS-Group object provides association or grouping all related CS. For e.g, in a contact center flow a call can get transferred to multiple agents. Each of these can trigger setup of new CS. In cases where the SRC knows the related CSs it can group them using the CS-Group element. CS-Group object is represented in XML schema using 'group' element.

6.2.1. Attributes

A CS-Group has the following attributes:

- group_id - This is to group different CSs that are related. SRC (or SRS) is responsible for ensuring the uniqueness of 'group_id' in case multiple SRC interacts with the same SRS. The mechanism by which SRC groups the CS is outside the scope of this document.

- associate-time - This is the time when a grouping is formed. The rules that determine how a grouping of different CS objects is done by SRC is outside the scope of this document.

- disassociate-time - 'disassociate-time' for CS-Group is calculated by SRC as the time when the grouping ends.

6.2.2. Linkages

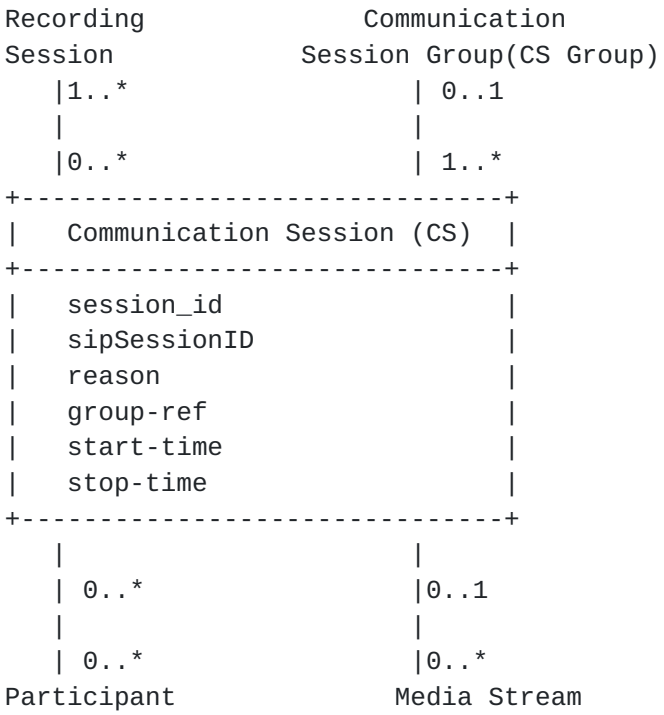
The linkages between CS-Group class and other classes are associations. A CS-Group is associated with RS and CS in the following manner:

- There are one or more RS objects per CS-Group.

- Each CS-Group object has to be associated with one or more RS. Here each RS can be setup by the potentially different SRCs.

- There are one or more CSs per CS-Group (for example, in case where the call is transferred). A CS cannot be associated with more than one CS-Group.

6.3. Communication Session



A Communication Session(CS) class and its object in the metadata model represents the CS and its properties as seen by SRC. CS object is represented in XML schema using 'session' element.

6.3.1. Attributes

A CS class has the following attributes:

- session_id - This attribute is used to uniquely identify an instance of CS object namely the session XML element with in the metadata XML document. 'session_id' is generated using the rules mentioned in Section 6.10.

- reason - This represents the reason why a CS was terminated. The value for this attribute is derived from SIP Reason header [RFC3326] of CS. There MAY be multiple instances of the 'reason' XML element inside a 'session' element. The 'reason' XML element has 'protocol' as an attribute, which indicates the protocol from which the reason string is derived. The default value for protocol attribute is "SIP". The 'reason' element can be derived from a SIP Reason header in the CS.

- sipSessionID - This attribute carries sip Session-ID defined in [I-D.ietf-insipid-session-id]. Each CS object can have zero or more sipSessionID elements. More than one 'sipSessionID' attribute may be present in a CS. For example, if three participants A, B and C are in a conference that has a focus acting as SRC, the metadata sent from the SRC to the SRS will likely have three 'sipSessionID' elements that correspond to the SIP dialogs the focus has with each of the three participants.

- group-ref - A 'group-ref' attribute MAY be present to indicate the group(identified by 'group_id') to which the enclosing session belongs.

- start-time - This optional attribute represents start time of CS as seen by SRC.

- stop-time - This optional attribute represents stop time of CS as seen by SRC.

This document does not specify attributes relating to what should happen to a recording of a CS after it has been delivered to the SRS (E.g., how long to retain the recording, what access controls to apply.) The SRS is assumed to behave in accordance with its local policy. The ability for the SRC to influence this policy is outside the scope of this document. However if there are implementations

where SRC desires to specify its own policy preferences, this could be sent as extension data attached to the CS.

6.3.2. Linkages

A CS is linked to CS-Group, Participant, Media Stream and RS classes using the association relationship. Association between CS and participant allows:

- CS to have zero or more participants
- Participant is associated with zero or more CSs. This includes participants who are not directly part of any CS. An example of such a case is participants in a premixed media stream. The SRC may have knowledge of such participants, yet not have any signaling relationship with them. This might arise if one participant in CS is a conference focus. To summarize, even if the SRC does not have direct signalling relationships with all participants in a CS, it should nevertheless create a participant object for each participant that it knows about.
- The model also allows participants in CS that are not participants in the media. An example is the identity of a Third Party Call Control(3pcc) that has initiated a CS to two or more participants of the CS. Another example is the identity of a conference focus. Of course a focus is probably in the media, but since it may only be there as a mixer, it may not report itself as a participant in any of the media streams.

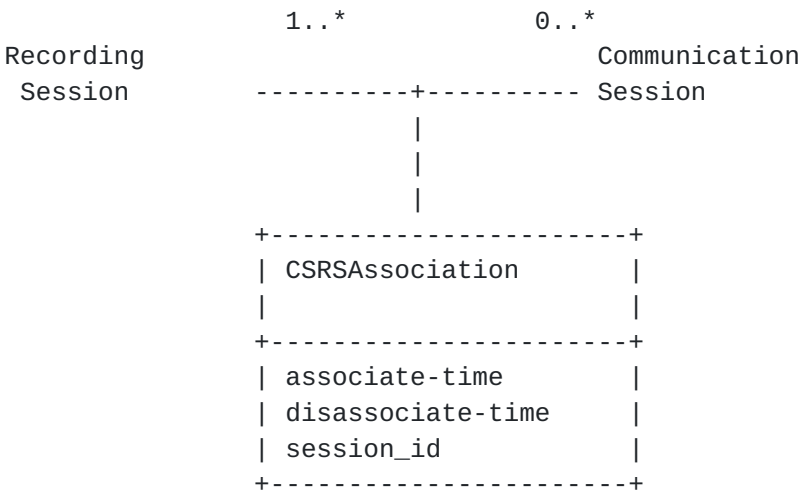
Association between CS and Media Stream allows:

- A CS to have zero or more streams
- A stream can be associated with at most one CS. A stream in a persistent RS is not required to be associated with any CS before the CS is created and hence the zero association is allowed.

Association between CS and RS allows:

- Each instance of RS has zero or more instances of CS objects.
- Each CS has to be associated with one more RS. Each RS can be potentially setup by different SRCs.

6.4. CSRSAssociation



The CSRSAssociation class describes the association of a CS to an RS for a period of time. A single CS may be associated with different RSs (perhaps by different SRCs) and may be associated and dissociated several times.

The CSRSAssociation is represented in XML using 'sessionrecordingassoc' XML element.

6.4.1. Attributes

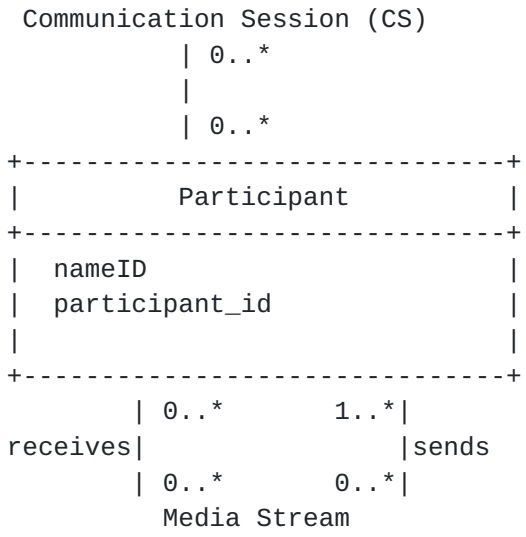
CSRSAssociation class has the following attributes:

- associate-time - associate-time is calculated by SRC as the time it sees a CS associated to a RS
- disassociate-time- disassociate-time is calculated by SRC as the time it see a CS disassociate from a RS.
- session_id - Each instance of this class MUST have 'session_id' attribute that identifies the the CS to which this association belongs to.

6.4.2. Linkages

CSRSAssociation class is linked to CS and RS classes.

6.5. Participant



A participant class and its objects has information about a device that is part of a CS and/or contributes/consumes media stream(s) belonging to a CS.

Participant object is represented in XML schema using 'participant' element.

6.5.1. Attributes

A participant class has two attributes:

- nameID - This attribute is a list of Name, Address-of-Record (AoR) defined in Section 6 of [RFC3261] tuples. An AoR can be one of SIP/SIPS/TEL URI, FQDN or IP address. For example, the AoR may be drawn from the From header field or P-Asserted-Identity header field. SRC's local policy is used to decide on where to draw the AoR from. Name represents participant name(SIP display name) or dialed number (DN) (when known). Multiple tuples are allowed for cases where a participant has more than one AoR. For example, a P-Asserted-identity header [RFC3325] can have both SIP and TEL URIs.

- participant_id - This attribute is used to identify the 'participant' XML element with in the XML document. It is generated using the rules mentioned in Section 6.10. This attribute MUST be used for all references to a participant within

a CS-Group, and MAY be used to reference the same participant more globally.

This document does not specify other attributes relating to participant e.g. participant role, participant type. An SRC which has information of these attributes can indicate the same as part of extension data to participant from SRC to SRS.

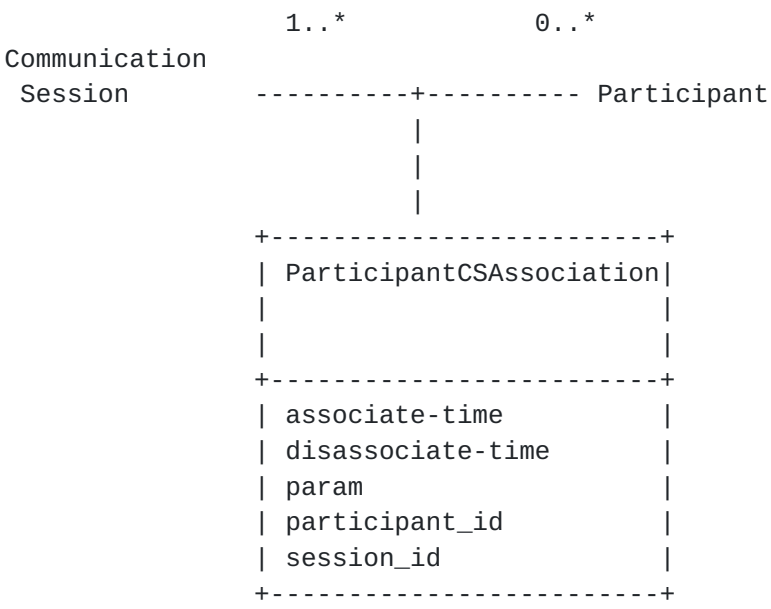
6.5.2. Linkages

The participant class is linked to MediaStream (MS) and CS class using association relationship. The association between participant and MS allows:

- participant to receive zero or more media streams.
- participant to send zero or more media streams. (Same participant provides multiple streams e.g. audio and video)
- media stream to be received by zero or more participants. Its possible, though perhaps unlikely, that a stream is generated but sent only to the SRC and SRS, not to any participant. E.g. In conferencing where all participants are on hold and the SRC is collocated with the focus. Also a media stream may be received by multiple participants (e.g. Whisper calls, side conversations).
- media stream to be sent by one or more participants (pre-mixed streams).

Example of a case where a participant receives zero or more streams - a supervisor may have side conversation with agent, while agent converses with customer.

6.6. ParticipantCSAssociation



The ParticipantCSAssociation class describes the association of a participant to an CS for a period of time. A participant may be associated and dissociated from a CS several times. (For example, connecting to a conference, then disconnecting, then connecting again.)

ParticipantCSAssociation object is represented in XML schema using 'participantsessionassoc' element.

6.6.1. Attributes

ParticipantCS association class has the following attributes:

- associate-time - associate-time is calculated by SRC as the time it sees a participant associated to a CS.

- disassociate-time- disassociate-time is calculated by the SRC as the time it sees a participant disassociate from a CS. It is possible that a given participant can have multiple associate/ disassociate times within given communication session.

- param - An optional attribute describing the capabilities of a participant in a CS, as defined in [RFC3840]. For example, in a CS(which can be a conference), you can have participants who are playing the role of "focus". These participants does not contribute to media in the CS, however they switch the media received from one participant to every other participant in the CS. Indicating the capability of participant (here "focus") would be useful for recorder to learn about these kind of participants.

The capablities are represented using 'param' XML element in the metadata. The 'param' XML element encoding defined in [RFC4235] is used to represent the capabilties attributes in metadata. Each participant may have zero or more capabilities. A participant may use different capabilities depending on the role it plays at a particular instance. For example, if a participant moves across different CSs (e.g., due to transfer) or is simultaneously present in different CSs with different roles.

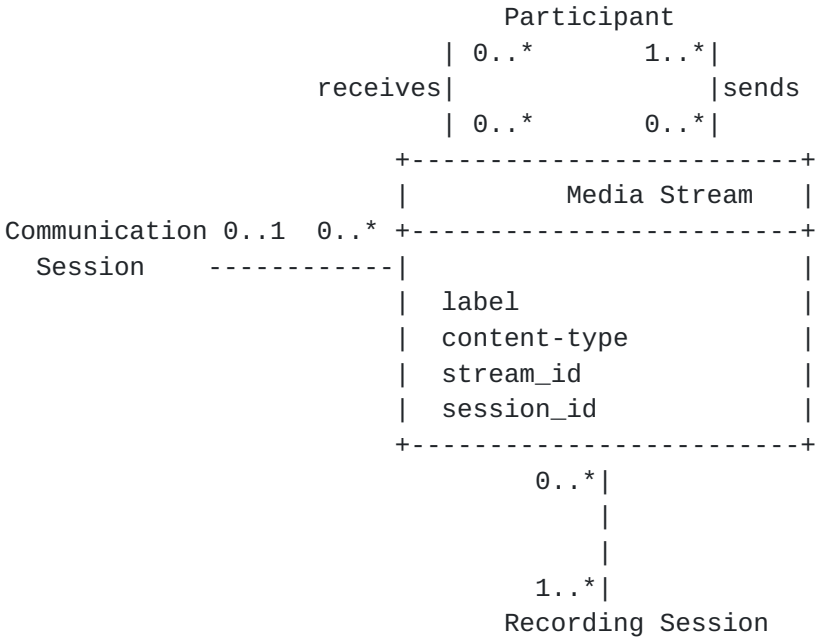
- participant_id - This attribute identifies the participant to which this association belongs to.

- session_id - This attribute identifies the session to which this association belongs to.

6.6.2. Linkages

The participantCSAssociation class is linked to participant and CS classes.

6.7. Media Stream



A MS class (and its objects) has the properties of media as seen by SRC and sent to SRS. Different snapshots of a MS objects may be sent whenever there is a change in media (e.g. direction change like pause/resume and/or codec change and/or participant change.).

MS object is represented in XML schema using 'stream' element.

6.7.1. Attributes

A MS class has the the following attributes:

- label - The label attribute within the 'stream' XML element references an SDP 'a=label' attribute that identifies an m-line within the RS SDP. That m-line carries the media stream from the SRC to the SRS.

- content-type - The content of an MS element will be described in terms of value from the [RFC4796] registry. If the SRC wishes to convey the Content-type to the SRS, it does so by including an 'a=content' attribute with the m-line in the RS SDP.

- stream_id - Each stream element has unique 'stream_id' attribute which helps to uniquely identify stream. This identifier is generated using the rules mentioned in Section 6.10.

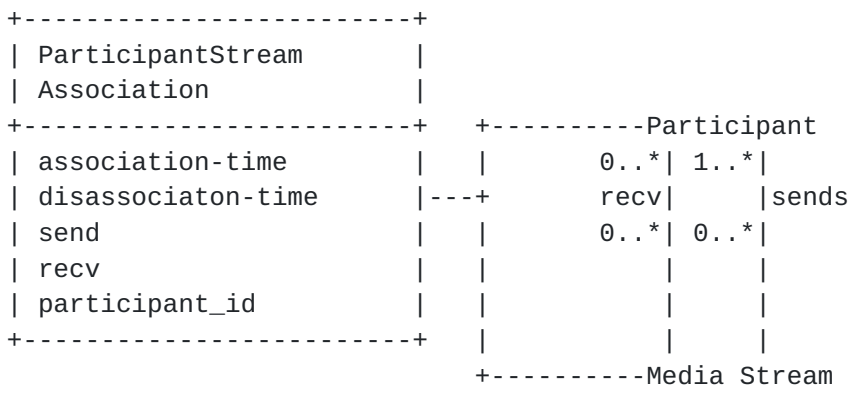
- session_id - This attribute associates the stream with a specific session element.

The metadata model can include media streams that are not being delivered to the SRS. For example, an SRC offers audio, video towards SRS which in response accepts only audio. The metadata snapshots sent from SRC to SRS can continue to indicate the changes to video stream as well.

6.7.2. Linkages

A MS class is linked to participant and CS classes using the association relationship. The details of association with the participant are described in the participant class section. The details of association with CS is mentioned in the CS section.

6.8. ParticipantStreamAssociation



A ParticipantStreamAssociation class describes the association of a Participant to a MS for a period of time, as a sender or as a receiver, or both.

This class is represented in XML using <participantstreamassoc> element.

6.8.1. Attributes

A ParticipantStreamAssociation class has the following attributes:

- associate-time: This attribute indicates the time a participant started contributing to a MS.

- disassociate-time: This attribute indicates the time a participant stopped contributing to a MS.

- send: This attribute indicates whether a participant is contributing to a stream or not. This attribute has a value which points to stream represented by its unique_id. The presence of this attribute indicates that a participant is contributing to a stream. If due to changes in CS if a participant stops contributing to a stream, a snapshot MUST be sent from SRC to SRS with no send element for that stream.

- recv: This attribute indicates whether a participant is receiving a media stream or not. This attribute has a value which points to a stream represented by its unique_id. The presence of this attribute indicates that a participant is receiving a stream. If due to changes in CS(like hold) the participants stops receiving a stream, a snapshot MUST be sent from SRC to SRS with no recv element for that stream.

- participant_id - This attributes points to the participant to which a stream element is associated with.

<participantstreamassoc> XML element is used to represent a participant association with a stream. The 'send' and 'recv' XML elements MUST be used to indicate whether a participant is contributing to a stream or receiving a stream. There MAY be multiple instances of the 'send' and 'recv' XML elements inside a 'particpantstreamassoc' element. If a metadata snapshot is sent with a 'participantstreamassoc' element that does not have any 'send' and 'recv' elements, it means that participant is neither contributing to any streams nor receiving any streams.

6.8.2. Linkages

The ParticipantStreamAssociation class is linked to participant and MS classes.

6.9. Syntax of date/time XML elements

XML elements <associate-time>, <disassociate-time>, <start-time> and <stop-time> contain strings representing the date and time. The value of these elements MUST follow the IMPP datetime format [RFC3339]. Timestamps that contain 'T' or 'Z' MUST use the capitalized forms.

As a security measure, the timestamp element MUST be included in all tuples unless the exact time of the status change cannot be determined.

6.10. Unique ID format

A Unique id is generated in two steps:

- The UUID is created using procedures mentioned in Sections 4.3, 4.4 or 4.5 of [RFC4122]. The algorithm MUST ensure that it does not use anything potentially personally identifying to generate the UUIDs. If implementations are using a Name-Based UUID as defined in Section 4.3 of [RFC4122], a name space ID generated using Section 4.2 or 4.5 of [RFC4122] might be a good choice.

- The UUID is encoded using base64 as defined in [RFC4648]

The above mentioned unique-id mechanism SHOULD be used for each metadata element. Multiple SRC's can refer to the same element/UUID (how each SRC learns the UUID here is out of scope of this document). If two SRCs use the same UUID, it MUST retain the UUID/element mapping. If the SRS detects that a UUID is mapped to more than one

element at any point of time it MUST treat this as a error. For example, the SRS may choose to reject or ignore the portions of metadata where it detects same UUID is mapped to a different elements other than the expected element (the SRS learns the mapped UUID when it sees a element for the first time in a metadata instance).

6.11. Metadata version Indicator

Metadata version is defined to help SRC and SRS to know the version of metadata XML schema used. SRCs and SRSs that support this specification MUST use version 1 in the namespace(urn:ietf:params:xml:ns:recording:1) in all the XML documents. Implementations may not interoperate if the version implemented by the sender is not known by the receiver. No negotiation of versions is provided. There is no significance to the version number although documents which update or obsolete this document (possibly including drafts of such documents) should include a higher version number if the metadata XML schema changes.

7. Recording metadata snapshot request format

SRS can explicitly request metadata snapshot from SRC. To request a metadata snapshot the SRS MUST send a SIP request message with a XML document having the namespace urn:ietf:params:xml:ns:recording:1. The XML document has the following elements.

- A <requestsnapshot> XML element MUST be present as the top level element in the XML document.

- A <requestreason> XML element that indicates the reason for requesting snapshot as a string MAY be present as a child XML element of <requestsnapshot>.

The example below shows a metadata snapshot request from SRS.

```
<?xml version="1.0" encoding="UTF-8"?>
  <requestsnapshot xmlns='urn:ietf:params:xml:ns:recording:1'>
  <requestreason xml:lang="it">SRS internal error</requestreason>
</requestsnapshot>
```

Example metadata snapshot request from SRS to SRC

8. SIP Recording Metadata Example

8.1. Complete SIP Recording Metadata Example

The following example provides all the tuples involved in Recording Metadata XML body.

```
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording:1'>
  <datamode>complete</datamode>
  <group group_id="7+OTCyoxTmqmqyA/1weDAg==">
    <associate-time>2010-12-16T23:41:07Z</associate-time>
    <!-- Standardized extension -->
    <call-center xmlns='urn:ietf:params:xml:ns:callcenter'>
          <supervisor>sip:alice@atlanta.com</supervisor>
    </call-center>
    <mydata xmlns='http://example.com/my'>
          <structure>FOO!</structure>
          <whatever>bar</whatever>
    </mydata>
  </group>
  <session session_id="hVpd7YQgRW2nD22h7q60JQ==">
        <sipSessionID>ab30317f1a784dc48ff824d0d3715d86;
        remote=47755a9de7794ba387653f2099600ef2</sipSessionID>
        <group-ref>7+OTCyoxTmqmqyA/1weDAg==</group-ref>
        <!-- Standardized extension -->
    <mydata xmlns='http://example.com/my'>
          <structure>FOO!</structure>
           <whatever>bar</whatever>
        </mydata>
  </session>
  <participant participant_id="srfBElmCRp2QB23b7Mpk0w==">
        <nameID aor="sip:bob@biloxi.com">
           <name xml:lang="it">Bob B</name>
        </nameID>
        <!-- Standardized extension -->
        <mydata xmlns='http://example.com/my'>
                <structure>FOO!</structure>
                <whatever>bar</whatever>
        </mydata>
  </participant>
  <participant participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
        <nameID aor="sip:Paul@biloxy.com">
          <name xml:lang="it">Paul</name>
        </nameID>
        <!-- Standardized extension -->
        <mydata xmlns='http://example.com/my'>
```

```
            <structure>FOO!</structure>
            <whatever>bar</whatever>
          </mydata>
    </participant>
    <stream stream_id="UAAMm5GRQKSCMVvLyl4rFw=="
            session_id="hVpd7YQgRW2nD22h7q60JQ==">
          <label>96</label>
    </stream>
    <stream stream_id="i1Pz3to5hGk8fuXl+PbwCw=="
            session_id="hVpd7YQgRW2nD22h7q60JQ==">
           <label>97</label>
    </stream>
    <stream stream_id="8zc6e0lYTlWIINA6GR+3ag=="
            session_id="hVpd7YQgRW2nD22h7q60JQ==">
          <label>98</label>
    </stream>
    <stream stream_id="EiXGlc+4TruqqoDaNE76ag=="
            session_id="hVpd7YQgRW2nD22h7q60JQ==">
          <label>99</label>
    </stream>
    <sessionrecordingassoc session_id="hVpd7YQgRW2nD22h7q60JQ==">
                  <associate-time>2010-12-16T23:41:07Z</associate-time>
    </sessionrecordingassoc>
    <participantsessionassoc
         participant_id="srfBElmCRp2QB23b7Mpk0w=="
         session_id="hVpd7YQgRW2nD22h7q60JQ==">
          <associate-time>2010-12-16T23:41:07Z</associate-time>
    </participantsessionassoc>
    <participantsessionassoc
         participant_id="zSfPoSvdSDCmU3A3TRDxAw=="
         session_id="hVpd7YQgRW2nD22h7q60JQ==">
             <associate-time>2010-12-16T23:41:07Z</associate-time>
    </participantsessionassoc>
    <participantstreamassoc
         participant_id="srfBElmCRp2QB23b7Mpk0w==">
             <send>i1Pz3to5hGk8fuXl+PbwCw==</send>
             <send>UAAMm5GRQKSCMVvLyl4rFw==</send>
             <recv>8zc6e0lYTlWIINA6GR+3ag==</recv>
             <recv>EiXGlc+4TruqqoDaNE76ag==</recv>
    </participantstreamassoc>
    <participantstreamassoc
        participant_id="zSfPoSvdSDCmU3A3TRDxAw==">
            <send>8zc6e0lYTlWIINA6GR+3ag==</send>
            <send>EiXGlc+4TruqqoDaNE76ag==</send>
            <recv>UAAMm5GRQKSCMVvLyl4rFw==</recv>
            <recv>i1Pz3to5hGk8fuXl+PbwCw==</recv>
    </participantstreamassoc>
  </recording>
```

Example metadata snapshot from SRC to SRS

8.2. Partial Update of Recording metadata XML body

The following example provides partial update in Recording metadata XML body for the above example. The example has a snapshot that carries the disassociate-time for a participant from a session.

```
  <?xml version="1.0" encoding="UTF-8"?>
<recording xmlns='urn:ietf:params:xml:ns:recording:1'>
  <datamode>partial</datamode>
    <participant
      participant_id="srfBElmCRp2QB23b7Mpk0w==">
      <nameID aor="sip:bob@biloxi.com">
                <name xml:lang="it">Bob R</name>
          </nameID>
    </participant>
    <participantsessionassoc
       participant_id="srfBElmCRp2QB23b7Mpk0w=="
       session_id="hVpd7YQgRW2nD22h7q60JQ==">
      <disassociate-time>2010-12-16T23:41:07Z</disassociate-time>
    </participantsessionassoc>
</recording>
```

Partial update of SIP Recording Example XML body

9. XML Schema definition for Recording metadata

This section defines XML schema for Recording metadata document

```
<?xml version="1.0" encoding="UTF-8"?>
   <xs:schema targetNamespace="urn:ietf:params:xml:ns:recording:1"
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      xmlns:tns="urn:ietf:params:xml:ns:recording:1"
      elementFormDefault="qualified"
      attributeFormDefault="unqualified">
    <!-- This import brings in the XML language attribute xml:lang-->
    <xs:import namespace="http://www.w3.org/XML/1998/namespace"
                schemaLocation="https://www.w3.org/2001/xml.xsd"/>
    <xs:element name="recording" type="tns:recording"/>
    <xs:complexType name="recording">
        <xs:sequence>
            <xs:element name="datamode" type="tns:dataMode"
                  minOccurs="0"/>
            <xs:element name="group" type="tns:group"
```

```
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="session" type="tns:session"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="participant" type="tns:participant"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="stream" type="tns:stream"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="sessionrecordingassoc"
                type="tns:sessionrecordingassoc"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="participantsessionassoc"
                type="tns:participantsessionassoc"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="participantstreamassoc"
                type="tns:participantstreamassoc"
                minOccurs="0" maxOccurs="unbounded"/>
            <xs:any namespace='##other'
                    minOccurs='0'
                    maxOccurs='unbounded'
                    processContents='lax'/>
         </xs:sequence>
    </xs:complexType>
    <xs:complexType name="group">
        <xs:sequence>
           <xs:element name="associate-time" type="xs:dateTime"
              minOccurs="0"/>
           <xs:element name="disassociate-time" type="xs:dateTime"
              minOccurs="0"/>
          <xs:any namespace='##other'
                    minOccurs='0'
                    maxOccurs='unbounded'
                    processContents='lax'/>
        </xs:sequence>
        <xs:attribute name="group_id" type="xs:base64Binary"
              use="required"/>
    </xs:complexType>
    <xs:complexType name="session">
        <xs:sequence>
           <xs:element name="sipSessionID" type="xs:string" minOccurs="0"
                   maxOccurs="unbounded"/>
           <xs:element name="reason" type="tns:reason" minOccurs="0"
                   maxOccurs="unbounded"/>
           <xs:element name="group-ref" type="xs:base64Binary"
                  minOccurs="0" maxOccurs="1"/>
           <xs:element name="start-time" type="xs:dateTime"
                  minOccurs="0" maxOccurs="1"/>
           <xs:element name="stop-time" type="xs:dateTime"
                  minOccurs="0" maxOccurs="1"/>
```

```
            <xs:any namespace='##other'
                   minOccurs='0'
                   maxOccurs='unbounded'
                   processContents='lax'/>
        </xs:sequence>
        <xs:attribute name="session_id" type="xs:base64Binary"
             use="required"/>
    </xs:complexType>
    <xs:complexType name="sessionrecordingassoc">
        <xs:sequence>
           <xs:element name="associate-time" type="xs:dateTime"
                  minOccurs="0"/>
           <xs:element name="disassociate-time" type="xs:dateTime"
                  minOccurs="0"/>
           <xs:any namespace='##other'
                   minOccurs='0'
                   maxOccurs='unbounded'
                   processContents='lax'/>
        </xs:sequence>
        <xs:attribute name="session_id" type="xs:base64Binary"
             use="required"/>
    </xs:complexType>
    <xs:complexType name="participant">
        <xs:sequence>
            <xs:element name="nameID" type="tns:nameID"
                   maxOccurs='unbounded'/>
        <xs:any namespace='##other'
                   minOccurs='0'
                   maxOccurs='unbounded'
                   processContents='lax'/>
        </xs:sequence>
        <xs:attribute name="participant_id" type="xs:base64Binary"
             use="required"/>
    </xs:complexType>
    <xs:complexType name="participantsessionassoc">
        <xs:sequence>
        <xs:element name="associate-time" type="xs:dateTime"
                  minOccurs="0"/>
        <xs:element name="disassociate-time" type="xs:dateTime"
                  minOccurs="0"/>
        <xs:element name="param" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
                <xs:attribute name="pname" type="xs:string" use="required"/>
                <xs:attribute name="pval" type="xs:string" use="required"/>
            </xs:complexType>
        </xs:element>
        <xs:any namespace='##other'
                minOccurs='0'
```

```
                maxOccurs='unbounded'
                processContents='lax'/>
        </xs:sequence>
        <xs:attribute name="participant_id" type="xs:base64Binary"
              use="required"/>
        <xs:attribute name="session_id" type="xs:base64Binary"
              use="required"/>
    </xs:complexType>
    <xs:complexType name="participantstreamassoc">
        <xs:sequence>
            <xs:element name="send" type="xs:base64Binary"
               minOccurs="0" maxOccurs="unbounded"/>
            <xs:element name="recv" type="xs:base64Binary"
               minOccurs="0" maxOccurs="unbounded"/>
               <xs:element name="associate-time" type="xs:dateTime"
                  minOccurs="0"/>
            <xs:element name="disassociate-time" type="xs:dateTime"
                  minOccurs="0"/>
               <xs:any namespace='##other'
                     minOccurs='0'
                     maxOccurs='unbounded'
                     processContents='lax'/>
        </xs:sequence>
        <xs:attribute name="participant_id" type="xs:base64Binary"
              use="required"/>
    </xs:complexType>
    <xs:complexType name="stream">
        <xs:sequence>
           <xs:element name="label" type="xs:string"
               minOccurs="0" maxOccurs="1"/>
            <xs:any namespace='##other'
                     minOccurs='0'
                     maxOccurs='unbounded'
                     processContents='lax'/>
        </xs:sequence>
        <xs:attribute name="stream_id" type="xs:base64Binary"
              use="required"/>
        <xs:attribute name="session_id" type="xs:base64Binary"/>
    </xs:complexType>
    <xs:simpleType name="dataMode">
        <xs:restriction base="xs:string">
            <xs:enumeration value="complete"/>
                <xs:enumeration value="partial"/>
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="nameID">
      <xs:sequence>
        <xs:element name="name" type ="tns:name" minOccurs="0"
```

```
                          maxOccurs="1"/>
      </xs:sequence>
      <xs:attribute name="aor" type="xs:anyURI" use="required"/>
    </xs:complexType>
   <xs:complexType name="name">
    <xs:simpleContent>
        <xs:extension base="xs:string">
           <xs:attribute ref="xml:lang" use="optional"/>
        </xs:extension>
    </xs:simpleContent>
   </xs:complexType>
   <xs:complexType name="reason">
      <xs:simpleContent>
       <xs:extension base="xs:string">
          <xs:attribute type="xs:short" name="cause" use="required"/>
          <xs:attribute type="xs:string" name="protocol" default="SIP"/>
       </xs:extension>
      </xs:simpleContent>
   </xs:complexType>
   <xs:element name="requestsnapshot" type="tns:requestsnapshot"/>
    <xs:complexType name="requestsnapshot">
       <xs:sequence>
          <xs:element name="requestreason" type="tns:name"  minOccurs="0"/>
          <xs:any namespace='##other'
            minOccurs='0'
            maxOccurs='unbounded'
            processContents='lax'/>
       </xs:sequence>
     </xs:complexType>
</xs:schema>
```

10. Security Considerations

This document describes an extensive set of metadata that may be recorded by the SRS. Most of the metadata could be considered private data. The procedures mentioned in security consideration section of [I-D.ietf-siprec-protocol] MUST be implemented by SRC and SRS for mutual authentication and to protect the content of the metadata in the RS.

An SRC MAY, by policy, choose to limit the parts of the metadata sent to the SRS for recording. And the policy of the SRS might not require all the metadata it receives. For the sake of data minimization, the SRS MUST NOT record additional metadata that is not explicitly required by local policy. Metadata in storage needs to be provided with a level of security that is comparable to that of the recording session.

11. IANA Considerations

This specification registers a new XML namespace, and a new XML schema.

11.1. SIP recording metadata Schema Registration

URI: urn:ietf:params:xml:ns:recording:1

Registrant Contact: IETF SIPREC working group, Ram Mohan R(rmohanr@cisco.com)

XML: the XML schema to be registered is contained in Section 8.

Its first line is <?xml version="1.0" encoding="UTF-8"?> and its last line is </xs:schema>

12. Acknowledgement

Thanks to John Elwell, Henry Lum, Leon Portman, De Villers, Andrew Hutton, Deepanshu Gautam,Charles Eckel, Muthu Arul Mozhi, Michael Benenson, Hadriel Kaplan, Brian Rosen, Scott Orton, Ofir Roth, Mary Barnes, Ken Rehor, Gonzalo Salgueiro, Yaron Pdut, Alissa Cooper, Stephen Farrell and Ben Campbell for their valuable comments and inputs.

Thanks to Joe Hildebrand, Peter Saint-Andre, Matt Miller for helping in writing the XML schema and Martin Thomson for validating the XML schema and providing comments on the same.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

[RFC2141] Moats, R., "URN Syntax", RFC 2141, DOI 10.17487/RFC2141, May 1997, <http://www.rfc-editor.org/info/rfc2141>.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, DOI 10.17487/RFC3261, June 2002, <http://www.rfc-editor.org/info/rfc3261>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <http://www.rfc-editor.org/info/rfc3339>.

[RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", RFC 4566, DOI 10.17487/RFC4566, July 2006, <http://www.rfc-editor.org/info/rfc4566>.

[RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", RFC 4574, DOI 10.17487/RFC4574, August 2006, <http://www.rfc-editor.org/info/rfc4574>.

[RFC4796] Hautakorpi, J. and G. Camarillo, "The Session Description Protocol (SDP) Content Attribute", RFC 4796, DOI 10.17487/RFC4796, February 2007, <http://www.rfc-editor.org/info/rfc4796>.

[RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", RFC 3840, DOI 10.17487/RFC3840, August 2004, <http://www.rfc-editor.org/info/rfc3840>.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <http://www.rfc-editor.org/info/rfc4122>.

[RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <http://www.rfc-editor.org/info/rfc4648>.

[I-D.ietf-siprec-protocol] Portman, L., Lum, H., Eckel, C., Johnston, A., and A. Hutton, "Session Recording Protocol", draft-ietf-siprec-protocol-18 (work in progress), September 2015.

13.2. Informative References

[RFC6341] Rehor, K., Ed., Portman, L., Ed., Hutton, A., and R. Jain, "Use Cases and Requirements for SIP-Based Media Recording (SIPREC)", RFC 6341, DOI 10.17487/RFC6341, August 2011, <http://www.rfc-editor.org/info/rfc6341>.

[I-D.ietf-insipid-session-id]
Jones, P., Salgueiro, G., Pearce, C., and P. Giralt, "End-to-End Session Identification in IP-Based Multimedia Communication Networks", draft-ietf-insipid-session-id-18 (work in progress), February 2016.

[RFC7245] Hutton, A., Ed., Portman, L., Ed., Jain, R., and K. Rehor, "An Architecture for Media Recording Using the Session Initiation Protocol", RFC 7245, DOI 10.17487/RFC7245, May 2014, <http://www.rfc-editor.org/info/rfc7245>.

[RFC2648] Moats, R., "A URN Namespace for IETF Documents", RFC 2648, DOI 10.17487/RFC2648, August 1999, <http://www.rfc-editor.org/info/rfc2648>.

[RFC3326] Schulzrinne, H., Oran, D., and G. Camarillo, "The Reason Header Field for the Session Initiation Protocol (SIP)", RFC 3326, DOI 10.17487/RFC3326, December 2002, <http://www.rfc-editor.org/info/rfc3326>.

[RFC3325] Jennings, C., Peterson, J., and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, DOI 10.17487/RFC3325, November 2002, <http://www.rfc-editor.org/info/rfc3325>.

[RFC4235] Rosenberg, J., Schulzrinne, H., and R. Mahy, Ed., "An INVITE-Initiated Dialog Event Package for the Session Initiation Protocol (SIP)", RFC 4235, DOI 10.17487/RFC4235, November 2005, <http://www.rfc-editor.org/info/rfc4235>.

[UML-REF] OMG, "Unified Modeling Language (UML)", 2011, <http://www.omg.org/spec/UML/2.4/>.

Authors' Addresses

Ram Mohan Ravindranath
Cisco Systems
Cessna Business Park
Bangalore, Karnataka
India

Email: rmohanr@cisco.com

Parthasarathi Ravindran
Nokia Networks
Bangalore, Karnataka
India

Email: partha@parthasarathi.co.in

Paul Kyzivat
Huawei
Hudson, MA
USA

Email: pkyzivat@alum.mit.edu