

SIPREC
Internet-Draft
Intended status: Standards Track
Expires: March 4, 2012

Ram Mohan. Ravindranath
Cisco Systems, Inc.
Parthasarathi. Ravindran
Sonus Networks
Paul. Kyzivat
Unaffiliated
September 1, 2011

Session Initiation Protocol (SIP) Recording Metadata
draft-ietf-siprec-metadata-04

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 March 4, 2012.

Copyright Notice

Copyright (c) 2011 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	4
2.	Terminology	4
3.	Definitions	4
4.	Metadata Model	5
5.	Recording Metadata Format	7
5.1.	XML data format	7
5.1.1.	Namespace	8
5.1.2.	recording	8
6.	Recording Metadata classes	8
6.1.	Recording Session	8
6.1.1.	Attributes	9
6.1.2.	Linkages	9
6.1.3.	XML element	9
6.2.	Communication Session Group	10
6.2.1.	Attributes	10
6.2.2.	Linkages	10
6.2.3.	XML element	11
6.3.	Communication Session	11
6.3.1.	Attributes	12
6.3.2.	Linkages	12
6.3.3.	XML element	13
6.4.	Participant	13
6.4.1.	Attributes	14
6.4.2.	Linkages	14
6.4.3.	XML element	15
6.5.	Media Stream	15
6.5.1.	Attributes	16
6.5.2.	Linkages	17
6.5.3.	XML element	17
6.6.	Extension Data	17
6.6.1.	Linkages	17
6.6.2.	XML element	17
6.7.	associate-time/disassociate-time	18
6.8.	Unique ID format	18
7.	SIP Recording Metadata Example	18
7.1.	Complete SIP Recording Metadata Example	18
7.2.	Partial Update of Recording metadata XML body	20
8.	XML Schema definition for Recording metadata	20

9.	Security Considerations	23
9.1.	Connection Security	23
10.	IANA Considerations	24
10.1.	SIP recording metadata Schema Registration	24
11.	Acknowledgement	24
12.	Appendix A: Metadata Model Object Instances	24
12.1.	Use case 1: Basic Call	24
12.2.	Use case 2: Hold/Resume	25
12.3.	Use case 3: Basic call with Transfer	27
12.4.	Conference Use Cases	28
12.4.1.	Case 1:	29
12.4.2.	Case 2:	31
12.4.3.	Case 3:	33
12.4.4.	Case 4:	34
13.	Appendix B: Metadata XML schema Instances	35
13.1.	Use case 1: Basic Call	35
13.2.	Use case 2: Hold/resume	37
13.3.	Use case 3: Basic Call with transfer	39
14.	References	43
14.1.	Normative References	43
14.2.	Informative References	44
	Authors' Addresses	44

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 and the Recording metadata format, the requirements for which are described in [RFC6341] and the architecture for which is described in [I-D.ietf-siprec-architecture].

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: The Metadata model is a class diagram in Unified Modelling Language(UML). The model is a structure diagram that describes the structure of a recording's Metadata 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 represents the relationships between the classes. Object diagrams represents instance diagrams of the class diagram and conveys snapshot of metadata.

Metadata classes: Each block in the model represents a class. In the metadata model each class is represented as a block having the block name. The description of each class also has representation of its attributes in a second compartment below the class name. Each instance of a class(namely the object) contributes some information to the recording's Metadata.

Attributes: Attributes represents the attributes 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 represents the relationship between the classes in

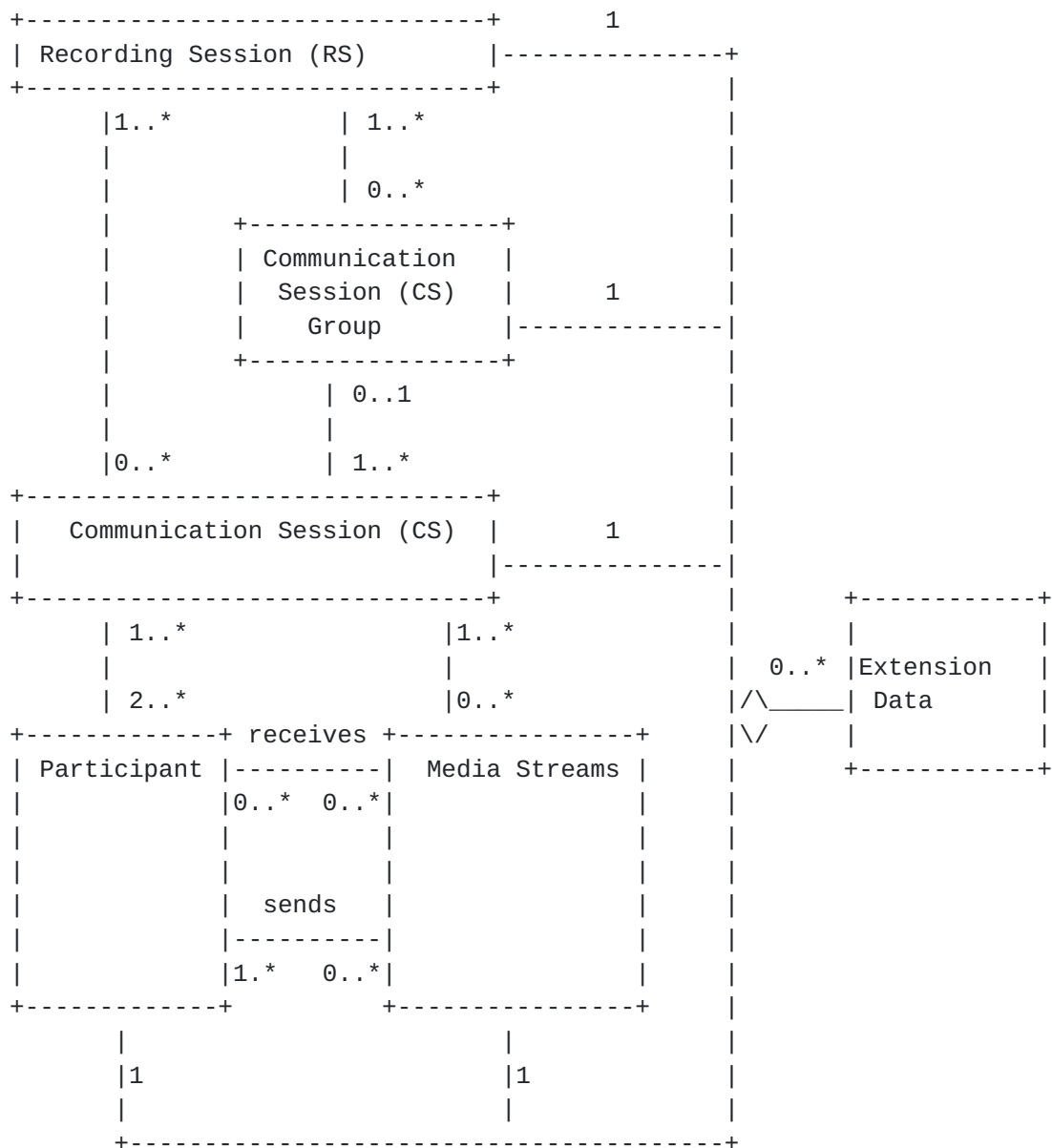
the model. It represents the logical connections between classes (or objects) in class diagrams/ object diagrams. The linkages can be associations or composition in the Metadata model of this document. An association represents a family of links. Binary associations (with two ends) are normally represented as a line, with each end connected to a class/object box. An association can be named, and the ends of an association can be adorned with role names, ownership indicators, multiplicity, visibility, and other properties. For instance, in the metadata model we have "sends" association between participant and media stream classes. The relation composition represents owns/holds relationship to show classes contained in another class. For instance, extension data class is contained in one of the other class (like stream or participant or any of the other in the model). The UML graphical representation of a composition relationship is a filled diamond shape on the containing class end of the tree of lines that connect contained class(es) to the containing class.

XML element: An XML element represent one XML schema complexType element (xs:complexType) of XML schema

XML attributes: An XML attribute represent one XML schema element (xs:element) of XML schema

4. Metadata Model

Metadata is the information that describes recorded media and the CS to which they relate. Below diagram shows a model for Metadata as viewed by Session Recording Server (SRS).



The Metadata model is a class diagram in Unified Modelling Language (UML). The model is a structure diagram that describes the structure of a recording's Metadata 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 represents the relationships which can be associations or Composition. Session Recording Client (SRC) MAY initiate the Recording Session. Here, Recording Session is a completely independent from the Communication Session that is being recorded at both the SIP dialog level and at the session level. The metadata

MUST be conveyed from SRC to SRS.

The model allows the capture of a snapshot of a recording's Metadata at a given instant in time. Metadata changes to reflect changes in what is being recorded. For example, if the call is transferred from one participant to another, then the SRC conveys the changes in the model (In this instance change of participant and the properties of the new media stream) to the SRS.

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. For instance, the timing of RS object changes (like Start / Stop time) may not be explicitly conveyed from the SRC to the SRS (The Date header in RS dialog SIP message provides the timing, but it is optional). In such cases the time a change occurred may be assumed to be the same as the time when notification of the change is received by the SRS. This is not true in cases where SRS requests a snapshot of metadata from SRC.

5. Recording Metadata Format

This section gives an overview of 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 describes about different media formats like audio, video. The other metadata attributes like participant details are represented in a new Recording specific XML document namely 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

Recording Metadata document is an XML document. recording element MUST present in all recording metadata XML document. recording acts as container for all other elements in this XML document.

Recording object is a XML document. It MUST have the XML declaration and it SHOULD 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 semantics error in recording XML document has to be informed to the originator using application specific mechanism.

[5.1.1.](#) Namespace

The namespace URI for elements defined by this specification is a Uniform Resource Namespace (URN) [[RFC2141](#)], using the namespace identifier 'ietf' defined by [[RFC2648](#)] and extended by [[RFC3688](#)].

The URN is as follows: urn:ietf:params:xml:ns:recording

[5.1.2.](#) recording

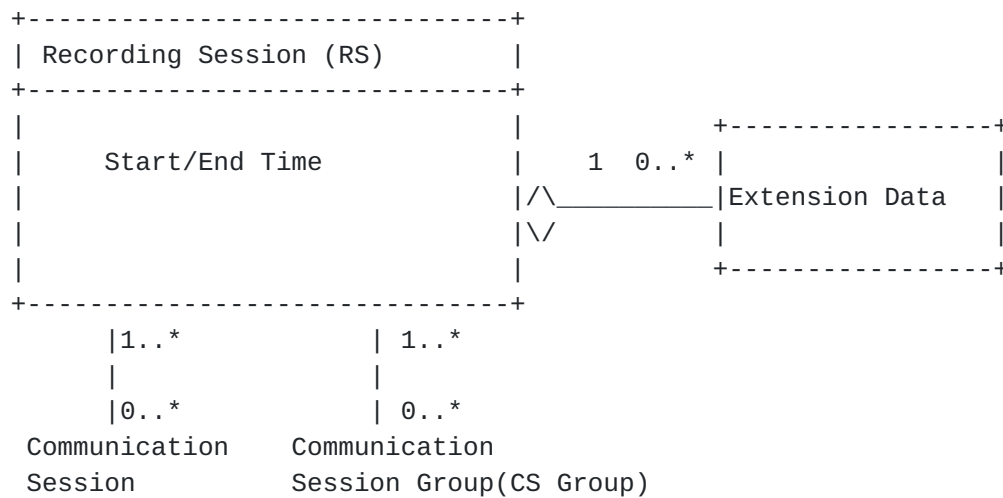
recording element MUST contain an xmlns namespace attribute with value as urn:ietf:params:xml:ns:recording. One recording element MUST present in the all recording metadata XML document.

dataMode element shows whether the XML document is complete document or partial update. 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 class (namely the Recording Session Object) represents a SIP session created between an SRC and SRS for the purpose of recording a Communication Session.

6.1.1. Attributes

A Recording Session class has the following attributes:

- o Start/End Time - Represents the Start/End time of a Recording Session object.

6.1.2. Linkages

Each instance of Recording Session has:

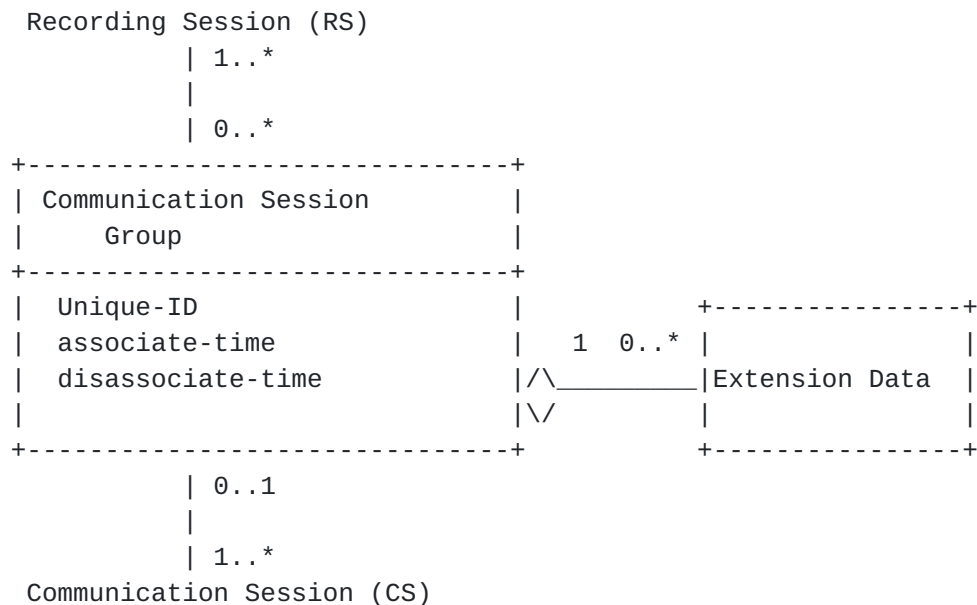
- o Zero or more instances of Communication Session Group. CSG may be zero because it is optional metadata object. Also the allowance of zero instances is to accommodate persistent recording, where there may be none.
- o Zero or more instances of Communication Session objects.

6.1.3. XML element

Recording Session object is represented by recording XML element. That in turn relies on the SIP/SDP session with which the XML document is associated to provide some of the attributes of the Recording Session element.

Start and End time value are derivable from Date header(if present in SIP message) in RS. In cases where Date header is not present, Start/End time are derivable from the time at which SRS receives the notification of SIP message to setup RS / disconnect RS.

6.2. Communication Session Group



One instance of a Communication Session Group class (namely the Communication Session Group object) provides association or linking of Communication Sessions.

6.2.1. Attributes

A CS Group has the following attributes:

- o Unique-ID - This Unique-ID is to group different CSs that are related. SRC (or SRS) is responsible for ensuring the uniqueness of Unique-ID in case multiple SRC interacts with the same SRS. The mechanism by which SRC groups the CS is outside the scope of SIPREC.
- o Associate-time - Associate-time for CS-Group shall be calculated by SRC as the time when a grouping is formed. The rules that determine how a grouping of different Communication Session objects is done by SRC is outside the scope of SIPREC.
- o Disassociate-time - Disassociate-time for CS-Group shall be calculated by SRC as the time when the grouping ends

6.2.2. Linkages

The linkages between Communication Session Group class and other classes is association. A communication Session Group is associated with RS and CS in the following manner:

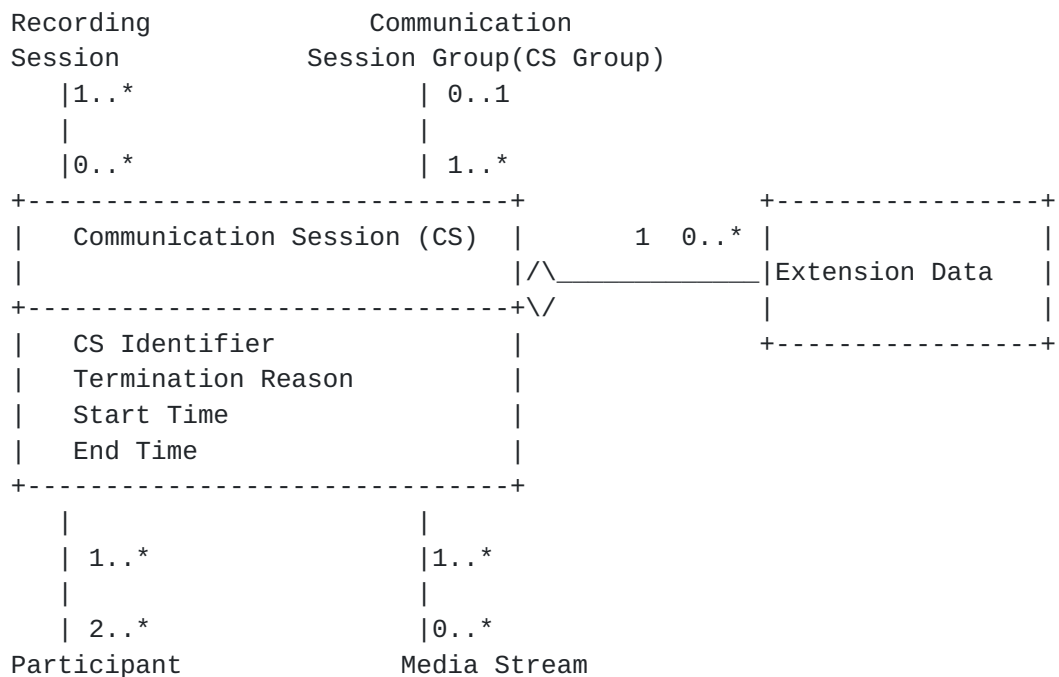
- o There is one or more Recording Session objects per Communication Session Group.
- o Each Communication Session Group object has to be associated with one or more RS [Here each RS can be setup by the potentially different SRCs]
- o There is one or more Communication Sessions per CS Group [e.g. Consult Transfer]

6.2.3. XML element

Group element is an optional element provides the information about the communication session group

Each communication session group (CSG) object is represented using one group element. Each group element has unique URN UUID attribute which helps to uniquely identify CSG.

6.3. Communication Session



A Communication Session class and its object in the metadata model represents Communication Session and its properties needed as seen by SRC.

6.3.1. Attributes

A communication Session class has the following attributes:

- o Termination Reason - This represents the reason why a CS was terminated. The communication session MAY contain a Call Termination Reason. This MAY be derived from SIP Reason header of CS.
- o CS Identifier - This attribute is used to uniquely identify a CS.
- o Start Time - This optional attribute represents CS start time
- o End Time - This optional attribute represents CS end time

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 policy. The ability for the SRC to influence this policy is outside the scope of this document. However if there are implementations where SRC has enough information, this could be sent as Extension Data attached to CS

6.3.2. Linkages

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

- o CS to have atleast two or more participants
- o Participant is associated with one or more CS's. 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 conf focus. To summarize even if 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
- o The model also allows participants in CS that are not participants in the media. An example is the identity of a 3pcc controller 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:

- o A CS to have zero or more Streams
- o A stream can be associated with 1 or more CS. An example is multicast MoH stream which might be associated with many CSs.

Association between CS and RS allows:

- o Each instance of RS has Zero or more instances of Communication Session objects.
- o Each CS has to be associated with one more RS [Here each RS can be potentially setup by different SRCs]

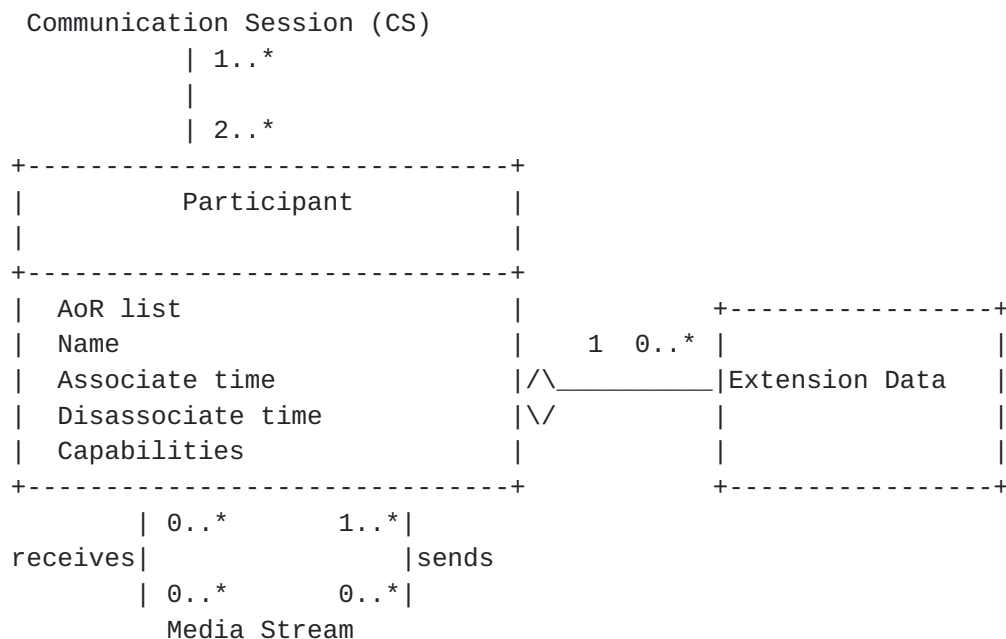
6.3.3. XML element

Session element provides the information about the communication session

Each communication session(CS) object is represented by one session element. Each session element has unique URN UUID attribute which helps to uniquely identify CS.

Reason element MAY be included to represent the Termination Reason attribute. group-ref element MAY exist to indicate the group where the mentioned session belongs.

6.4. 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.

6.4.1. Attributes

Participant has attributes like:

- o AoR list - Has list of AoRs. An AoR MAY be SIP/SIPS/TEL URI. There are cases where a participant can have more than one AoR [e.g. P-Asserted-ID which can have both SIP and TEL URIs]
- o Name - This attribute represents Participant name(SIP display name) or DN number (in case it is known)
- o Associate-time - associate-time is calculated by SRC as the time it sees a participant is associated to CS
- o Disassociate-time- Disassociate-time is calculated by SRC as the time it see a participant disassociate from a CS. It is possible that a given participant can have multiple associate/disassociate times within given communication session.
- o Capabilities - A participant capabilities as defined in [RFC 3840](#) [[RFC3840](#)] is an optional attribute that includes the capabilities of a participant in a CS. This attribute is an attribute of association of participant to CS. Each participant shall have Zero or more capabilities. A participant may use different capabilities depending on the role it plays at a particular instance. IOW if a participants moves across different CSs (due to transfer e.t.c) OR is simultaneously present in different CSs its role may be different and hence the capability used.

NOTE: How to represent capabilities attribute in XML is an open item.

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.

6.4.2. Linkages

The participant class is linked to MS and CS class using association relationship. The association between participant and Media Stream allows:

- o Participant to receives zero or more media streams
- o Participant to send zero or more media streams. (Same participant provides multiple streams e.g. audio and video)
- o 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).

- o 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.

6.4.3. XML element

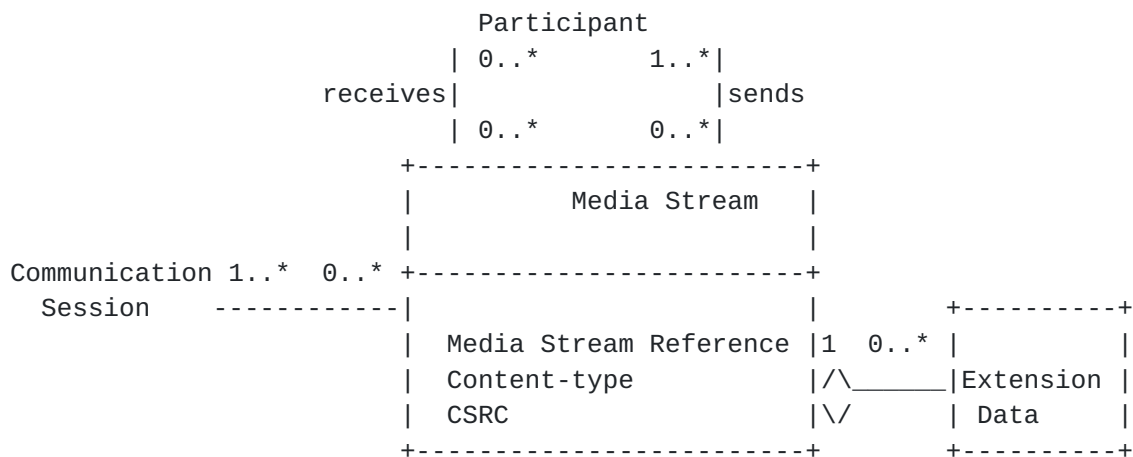
A participant element represents a Participant object.

There MUST be atleast 2 participant for any given session. "send" or "recv" element in each participant is associating SDP m-lines with the participant. send element indicates that participant is sending the stream of media with the mentioned media description. recv element indicates that participant is receiving the stream and by default all participant will receive the stream. recv element has relevance in case whisper call scenario wherein few of the participant in the session receives the stream and not others.

Participant MUST have AOR element which contains SIP/SIPS URI to identify the participant. AOR element is SIP/SIPS URI FQDN or IP address which represents the user. name is an optional element to represent display name.

Each participant element has unique URN UUID attribute which helps to uniquely identify participant and session URN UUID to associate participant with specific session element. URN UUID of participant *MUST* used in the scope of CSG and no new URN UUID has to be created for the same element (participant, stream) between different CS in the same CSG. In case URN UUID has to be used permanent, careful usage of URN UUID to original AoR has to be decided by the implementers and it is implementer's choice.

6.5. Media Stream



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

6.5.1. Attributes

A Media Stream class has the the following attributes:

- o Media Stream Reference - In implementations this can reference to m-line
- o Content - The content of an MS element will be described in terms of value from the [RFC 4796](#) [RFC4796] registry.
- o CSRC - The linkage between the participants to its contributing media stream in a mixed RS stream is provided by CSRC attribute. Not all SRC may have the capability to determine this and hence this is a optional attribute. Having this info can allow the SRS to determine which participants are part(contributors) of particular parts of the mixed stream.

NOTE: How CSRC can be represented in XML is still an open item. CSRC attribute is an attribute of media stream to participant association. Another open items if there is a need for multiple CSRCs. A participant may have multiple CSRCs - potentially a different one for each stream in which it participates.

The metadata model should include media streams that are not being delivered to the SRS. Examples include cases where SRC offered certain media types but SRS chooses to accept only a subset of them OR an SRC may not even offer a certain media type due it its restrictions to record

6.5.2. Linkages

A Media Stream 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.

6.5.3. XML element

stream element represents a Media Stream object. Stream element indicates SDP media lines associated with the session and participants.

This element indicates the SDP m-line properties like label attributes, media mode. Label attribute is used to link m-line SDP body using label attribute in SDP m-line. The media mode helps in understanding whether the media is mixed or not.

Each stream element has unique URN UUID attribute which helps to uniquely identify stream and session URN UUID to associate stream with specific session element.

The content attribute if an SRC wishes to send is conveyed in RS SDP.

6.6. Extension Data

A recording metadata object contains additional data not specified as part of siprec. This is intended to accommodate future standards track extensions, as well as vendor and user specific extensions. The mechanism MUST provide a means of unambiguously distinguishing such extension data.

6.6.1. Linkages

Extension data class is linked to other classes using the composition relationship. An extension data class (and its objects) is contained/owned by another Metadata class. Each instance of Metadata class(except extension data class itself) has

- o Zero or more instances of Extension data class
- o Each Extension data class is contained/owned by a Metadata class other than itself

6.6.2. XML element

Extensiondata element provides the mechanism by which namespace/ element MAY be extended with standard or proprietary information.

extensiondata element MUST include any other XML namespace. Multiple namespace MAY exist under extensiondata. extensiondata element exist in each level like recording, session, participant, stream to provide extensiondata specific to each element. extensiondata element MUST be part of parent element for which the additional information is sent and hence no Unique ID is needed.

6.7. associate-time/disassociate-time

associate-time/disassociate-time contains a string indicating the date and time of the status change of this tuple. The value of this element MUST follow the IMPP datetime format [[RFC3339](#)]. Timestamps that contain 'T' or 'Z' MUST use the capitalized forms. At a time, any of the time tuple associate-time or disassociate-time MAY exist in the element namely group, session, participant and not both timestamp at the same time.

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

6.8. Unique ID format

UUID encoded more densely than the UUID URN (e.g. radix64 of the binary uuid form defined in [RFC 4122](#) [[RFC4122](#)]) SHOULD be used as a format for Unique ID. How to represent this id in XML is still an open item.

7. SIP Recording Metadata Example

7.1. 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'>
    <dataMode>complete</dataMode>
    <group id="urn:uuid:efe3930b-2a31-4e6a-a6ab-203fd7078302">
      <associate-time>2010-12-16T23:41:07Z</associate-time>
      <extensiondata>
        <!-- Standardized extension -->
        <call-center xmlns='urn:ietf:params:xml:ns:callcenter'>
          <supervisor>sip:alice@cisco.com</supervisor>
        </call-center>
      <mydata xmlns='http://example.com/my'>
```



```
<structure>F00!</structure>
  <whatever>bar</whatever>
</mydata>
</extensiondata>
</group>
<session id="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
  <group-ref>urn:uuid:efe3930b-2a31-4e6a-a6ab-203fd7078302
  </group-ref>
  <start-time>2010-12-16T23:41:07Z</start-time>
  <extensiondata>
    <structure>F00!</structure>
    <whatever>bar</whatever>
  </extensiondata>
</session>
<participant
  id="urn:uuid:b2b7c112-5982-469d-9007-6ddbbecca64d3"
  session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
  <aor>sip:partha@blr.cisco.com</aor>
  <send>urn:uuid:8b53f3de-da39-4846-93c7-ee5e5f8f6f0b</send>
  <send>urn:uuid:50000c9b-9191-40a4-8231-5bcbca5e2b17</send>
  <recv>urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a</recv>
  <recv>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</recv>
  <associate-time>2010-12-16T23:41:07Z</associate-time>
  <extensiondata>
    <structure>F00!</structure>
    <whatever>bar</whatever>
  </extensiondata>
</participant>

<participant
  id="urn:uuid:cd27cfa1-2bdd-4830-a653-70374d10f103"
  session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
  <aor>sip:paul@box.cisco.com</aor>
  <send>urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a</send>
  <send>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</send>
  <recv>urn:uuid:50000c9b-9191-40a4-8231-5bcbca5e2b17</recv>
  <recv>urn:uuid:8b53f3de-da39-4846-93c7-ee5e5f8f6f0b</recv>
  <associate-time>2010-12-16T23:41:07Z</associate-time>
  <extensiondata>
    <structure>F00!</structure>
    <whatever>bar</whatever>
  </extensiondata>
</participant>
<stream id="urn:uuid:50000c9b-9191-40a4-8231-5bcbca5e2b17"
  session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
  <csrc>1</csrc>
  <label>96</label>
</stream>
```



```
<stream id="urn:uuid:8b53f3de-da39-4846-93c7-ee5e5f8f6f0b"
  session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
  <label>97</label>
</stream>
<stream id="urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a"
  session="urn:uuid:cd27cfa1-2bdd-4830-a653-70374d10f103">
  <label>98</label>
</stream>
<stream id="urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a"
  session="urn:uuid:cd27cfa1-2bdd-4830-a653-70374d10f103">
  <label>99</label>
</stream>
</recording>
```

SIP Recording Metadata Example XML body

7.2. Partial Update of Recording metadata XML body

The following example provides partial update in Recording Metadata XML body for the above example. The example illustrate the disassociate-time for a participant from a session.

```
<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns='urn:ietf:params:xml:ns:recording'>
  <dataMode>partial</dataMode>
  <participant
    id="urn:uuid:b2b7c112-5982-469d-9007-6ddbbecca64d3"
    session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
    <aor>sip:partha@blr.cisco.com</aor>
    <disassociate-time>2010-12-16T23:41:07Z</disassociate-time>
    <extensiondata>
      <structure>F00!</structure>
      <whatever>bar</whatever>
    </extensiondata>
  </participant>
</recording>
```

Partial update of SIP Recording Example XML body

8. 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"
  xmlns:tns="urn:ietf:params:xml:ns:recording"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified">
  <!-- This import brings in the XML language attribute xml:lang-->
  <xs:import namespace="http://www.w3.org/XML/1998/namespace"/>
  <xs:element name="recording" type="recording"/>
  <xs:complexType name="recording">
    <xs:sequence>
      <xs:element name="datamode" type="dataMode"
        minOccurs="0"/>
      <xs:element name="group" type="group"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="session" type="session"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="participant" type="participant"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="stream" type="stream"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:element name="extensiondata" type="extensiondata"
        minOccurs="0" maxOccurs="unbounded"/>
    </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:element name="extensiondata" type="extensiondata"
        minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="id" type="urnuuid"
      use="required"/>
  </xs:complexType>
  <xs:complexType name="session">
    <xs:sequence>
      <xs:element name="associate-time" type="dateTime"
        minOccurs="0"/>
      <xs:element name="disassociate-time" type="dateTime"
        minOccurs="0"/>
      <xs:element name="reason" type="xs:string"
        minOccurs="0"/>
      <xs:element name="group-ref" type="urnuuid"
        minOccurs="0" maxOccurs="1"/>
      <xs:element name="extensiondata" type="extensiondata"
        minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```



```
</xs:sequence>
<xs:attribute name="id" type="urnuuid"
  use="required"/>
</xs:complexType>
<xs:complexType name="participant">
  <xs:sequence>
    <xs:element name="aor" type="xs:anyURI"
      maxOccurs="unbounded"/>
    <xs:element name="name" type="xs:string"
      minOccurs="0" maxOccurs="1"/>
    <xs:element name="send" type="urnuuid"
      minOccurs="0" maxOccurs="unbounded"/>
    <xs:element name="recv" type="urnuuid"
      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:element name="extensiondata" type="extensiondata"
      minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="id" type="urnuuid"
    use="required"/>
  <xs:attribute name="session" type="urnuuid"
    use="required"/>
</xs:complexType>
<xs:complexType name="stream">
  <xs:sequence>
    <xs:element name="label" type="xs:string"
      minOccurs="0" maxOccurs="1"/>
    <xs:element name="mode" type="streamMode"
      minOccurs="0" maxOccurs="1"/>
    <xs:element name="extensiondata" type="extensiondata"
      minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="id" type="urnuuid"
    use="required"/>
  <xs:attribute name="session" type="urnuuid"
    use="required"/>
</xs:complexType>
<xs:element name='extensiondata'>
  <xs:complexType>
    <xs:any namespace='##other'
      minOccurs='0'
      maxOccurs='unbounded'
      processContents='lax' />
  </xs:complexType>
  <xs:attribute name="id" type="urnuuid"
```



```
        use="required"/>
    <xs:attribute name="parent" type="urnuuid"
        use="required"/>
</xs:element>
<xs:simpleType name="streamMode">
    <xs:restriction base="xs:string">
        <xs:pattern
            value="mixed|separate"/>
    </xs:restriction>
</xs:simpleType>
<xs:simpleType name="urnuuid">
    <xs:restriction base="xs:string">
        <xs:pattern
            value="urn:uuid:[0-9a-zA-Z]{8}-[0-9a-zA-Z]{4}
                -[0-9a-zA-Z]{4}-[0-9a-zA-Z]{4}-[0-9a-zA-Z]{12}"/>
    </xs:restriction>
</xs:simpleType>
</xs:simpleType>
<xs:simpleType name="dataMode">
    <xs:restriction base="xs:string">
        <xs:enumeration value="complete"/>
        <xs:enumeration value="partial"/>
    </xs:restriction>
</xs:simpleType>
```

9. Security Considerations

The metadata information sent from SRC to SRS MAY reveal sensitive information about different participants in a session. For this reason, it is RECOMMENDED that a SRC use a strong means for authentication and metadata information protection and that it apply comprehensive authorization rules when using the metadata format defined in this document. The following sections will discuss each of these aspects in more detail.

9.1. Connection Security

It is RECOMMENDED that a SRC authenticate SRS using the normal SIP authentication mechanisms, such as Digest as defined in [Section 22 of \[RFC3261\]](#). The mechanism used for conveying the metadata 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 [\[RFC3261\]](#), SHOULD be used.

If a strong end-to-end security means (such as above) is not available, it is RECOMMENDED that a SRC use mutual hop-by-hop Transport Layer Security (TLS) authentication and encryption mechanisms described in "SIPS URI Scheme" and "Interdomain Requests"

of [[RFC3261](#)].

10. IANA Considerations

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

10.1. SIP recording metadata Schema Registration

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

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

XML: the XML schema to be registered is contained in [Section 6](#).

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

11. Acknowledgement

We wish to thank John Elwell(Siemens-Enterprise), Henry Lum(Alcatel-Lucent), Leon Portman(Nice), De Villers, Andrew Hutton(Siemens-Enterprise), Deepanshu Gautam(Huawei), Charles Eckel(Cisco), Muthu Arul(Cisco), Michael Benenson(Cisco), Hadriel Kaplan (ACME), Brian Rosen(Neustar), Scott Orton(Broadsoft) for their valuable comments and inputs.

We wish to thank Joe Hildebrand(Cisco), Peter Saint-Andre(Cisco) for the valuable XML related guidance and Martin Thompson for validating the XML schema and providing comments on the same.

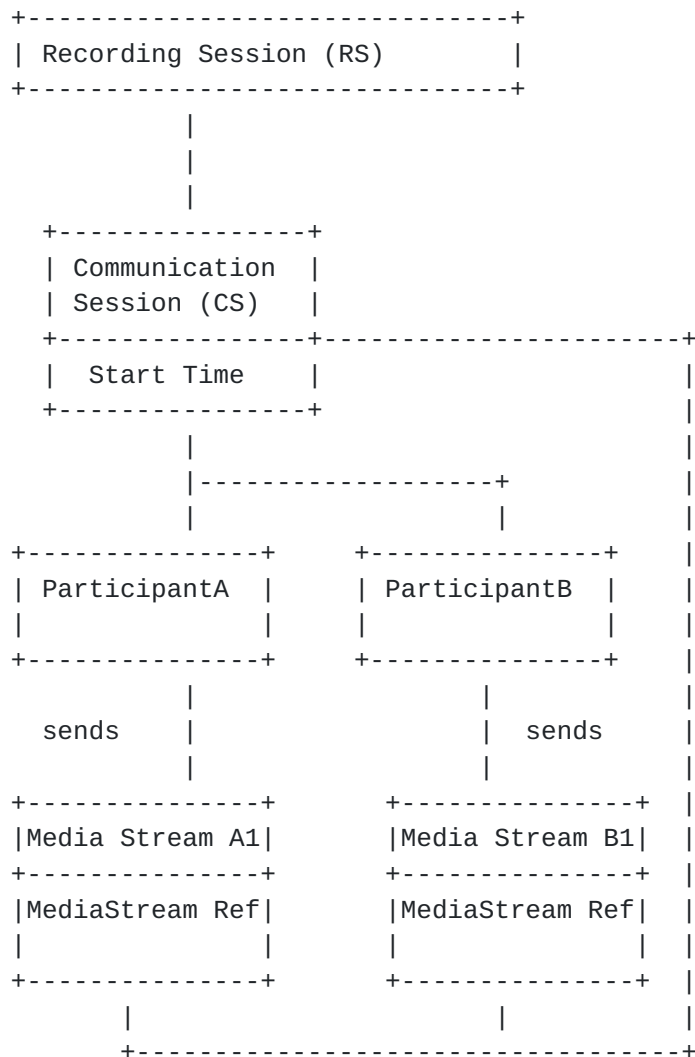
12. [Appendix A](#): Metadata Model Object Instances

This section describes the metadata model object instances for different use cases of SIPREC. For the sake of simplicity as the media streams sent by each of the participants is received by every other participant in these use cases, it is NOT shown in the object instance diagrams below. Also for the sake of ease not all attributes of each object are shown in these instance diagrams.

12.1. Use case 1: Basic Call

Basic call between two Participants A and B. In this use case each participant sends one Media Stream. For the sake of simplicity

"receives" lines are not shown in this instance diagram. Media Streams sent by each participant is received all other participants of that CS.



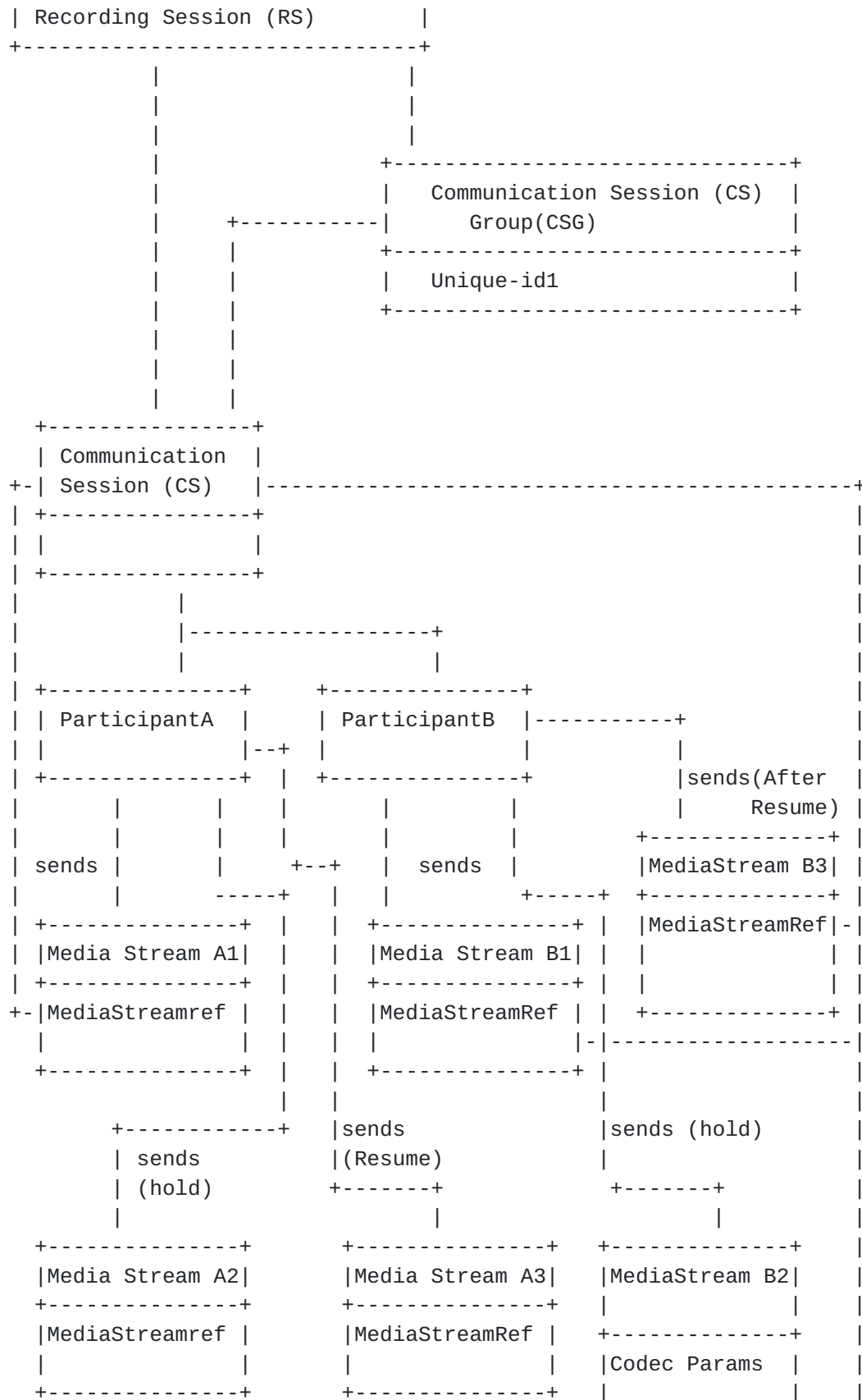
[12.2.](#) Use case 2: Hold/Resume

Basic call between two Participants A and B and with Participant A or B doing a Hold/Resume. In this use case each participant sends one Media Stream. After Hold/Resume the properties of Media can change. For the sake of simplicity "receives" lines are not shown in this instance diagram. Media Streams sent by each participant is received all other participants of that CS.

```

+-----+

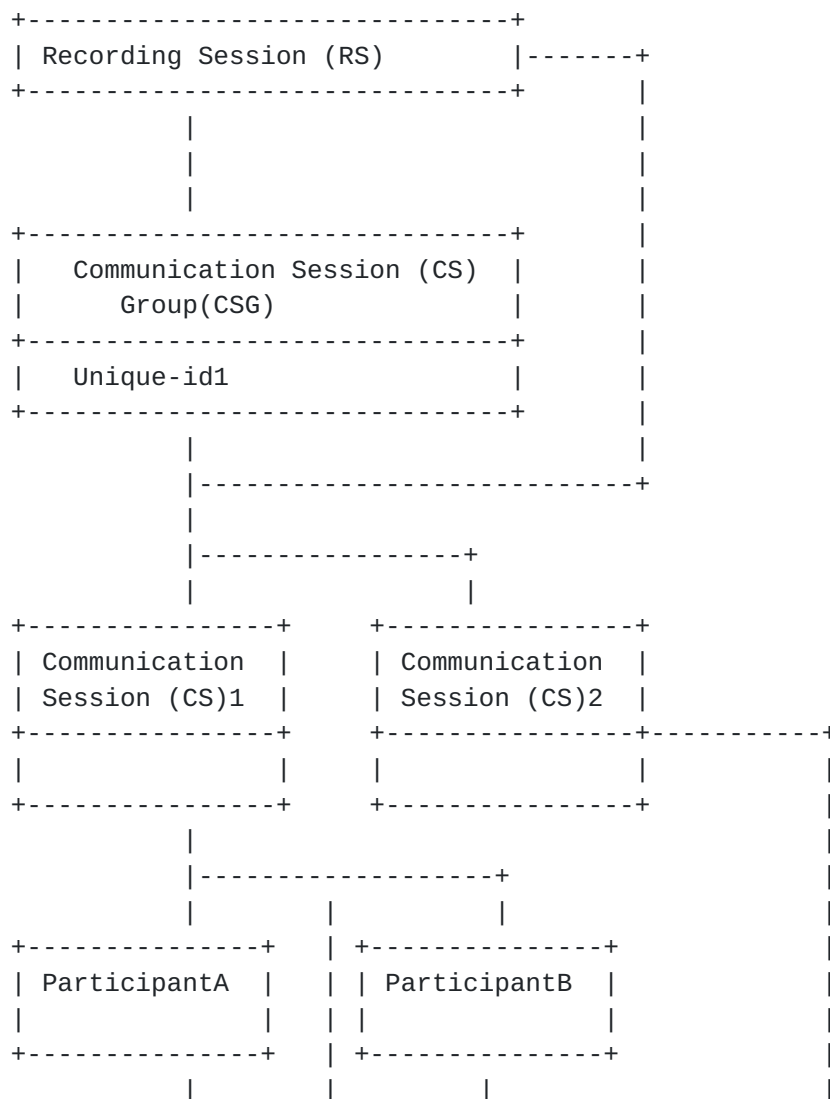
```

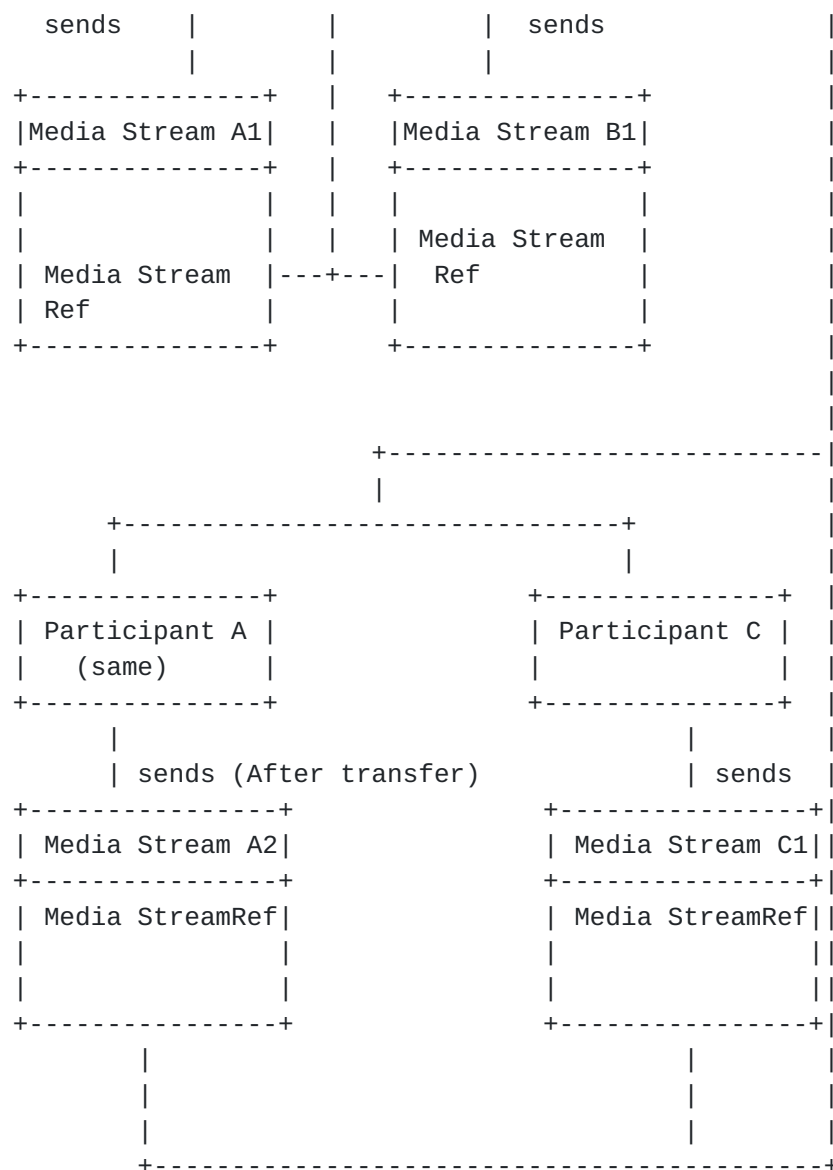





12.3. Use case 3: Basic call with Transfer

Basic call between two Participants A and B and with Participant A transfer(consult transfer) to Participant C. In this use case each participant sends one Media Stream. After transfer the properties of Participant A Media can change. For the sake of simplicity "receives" lines are not shown in this instance diagram. Media Streams sent by each participant is received all other participants of that CS.





12.4. Conference Use Cases

Depending on who act as SRC and the information that an SRC has there can be several ways to model conference use cases. This section has instance diagrams for the following cases:

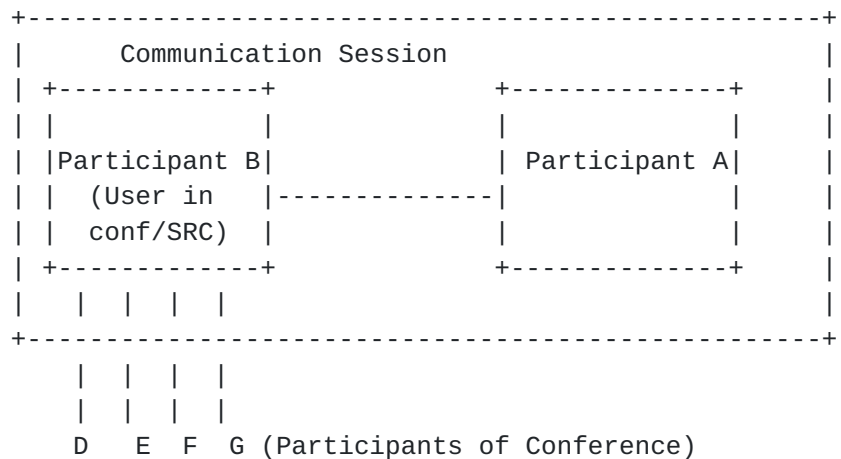
- o A CS where one of the participant (which is also SRC) is a user in a conference
- o A CS where one of the participant is focus (which is also SRC)
- o A CS where one of the participant is user and the SRC is a different entity like B2BUA

- o A CS where one of the participant is focus and the SRC is a different entity like B2BUA

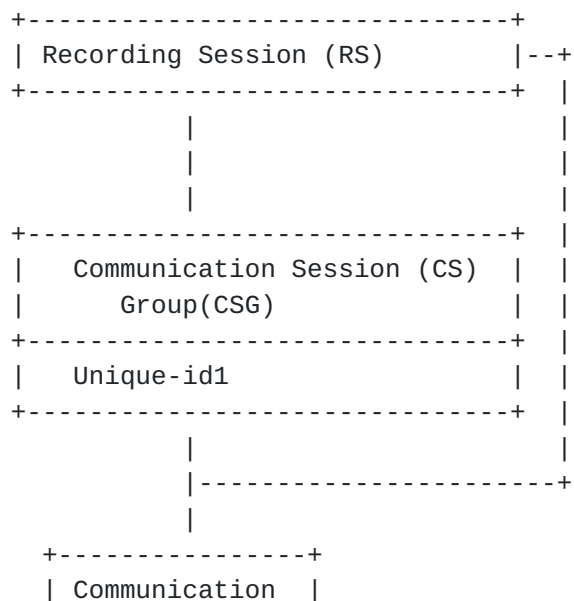
NOTE: There MAY be other ways to model the same use cases depending on what information the SRC has.

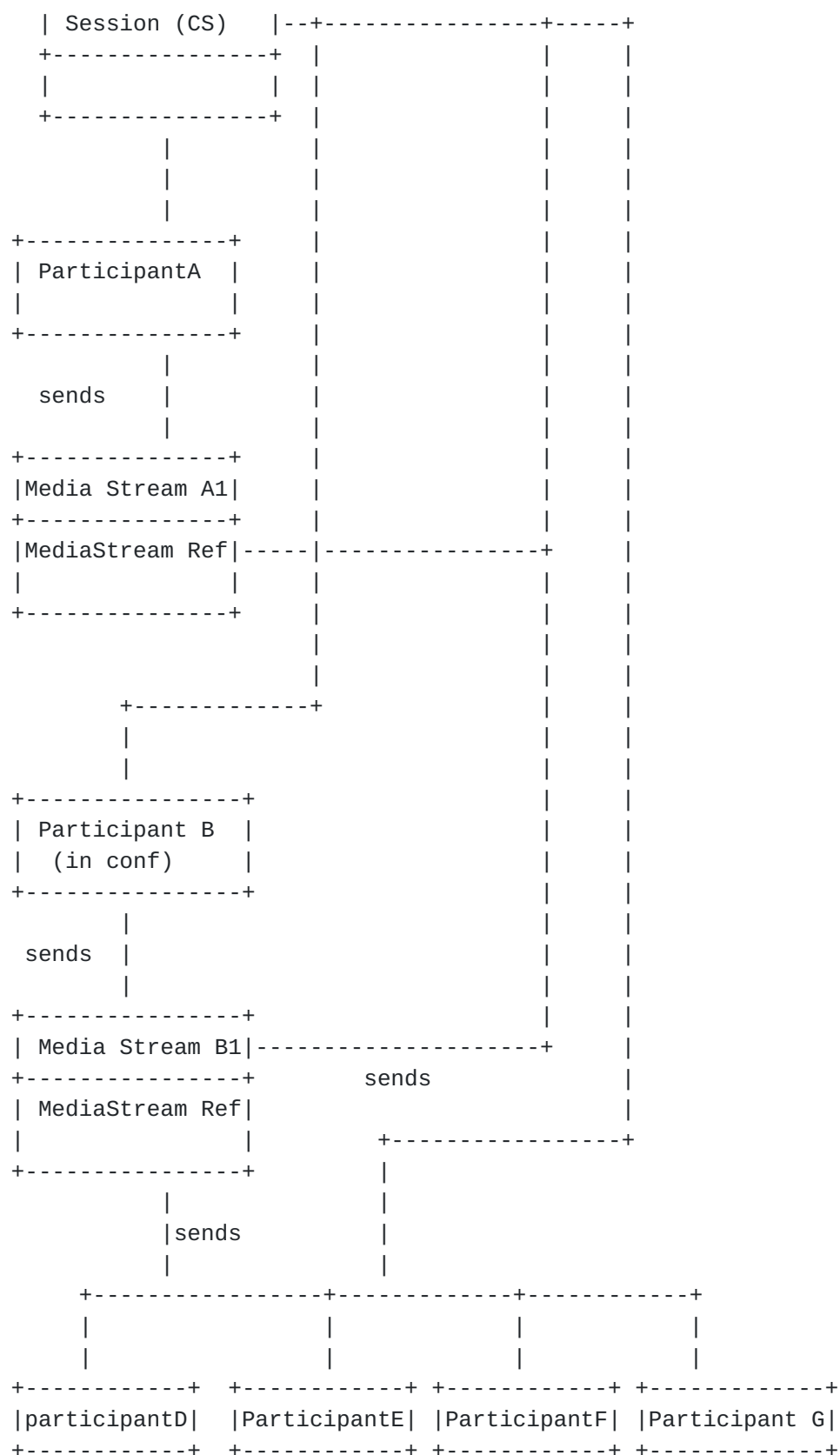
12.4.1. Case 1:

This is the usecase where there is a CS with one of the participant (who is also SRC) as a user in a conference. For the sake of simplicity the receive lines for each of the participant is not shown.



Instance Diagram:

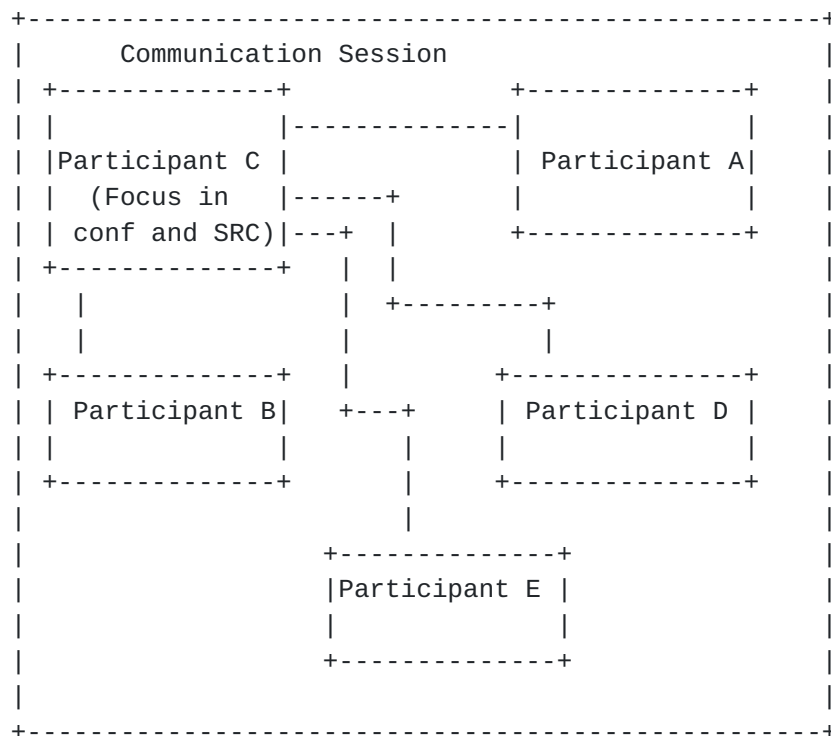




In this example we have two participants A and B who are part of a Communication Session(CS). One of the participants B is part of a conference and also acts as SRC. There can be two cases here. B can be a participant of the conference or B can be a focus. In this instance diagram Participant B is a user in a conference. The SRC (Participant B) subscribes to conference event package to get the details of other participants. Participant B(SRC) sends the same through the metadata to SRS. In this instance diagram the Media Stream(mixed stream) sent from Participant B has media streams contributed by conference participants (D,E,F and G). For the sake of simplicity the "receives" line is not shown here. In this example the media stream sent by each participant(A or B) of CS is received by all other participant(A or B).

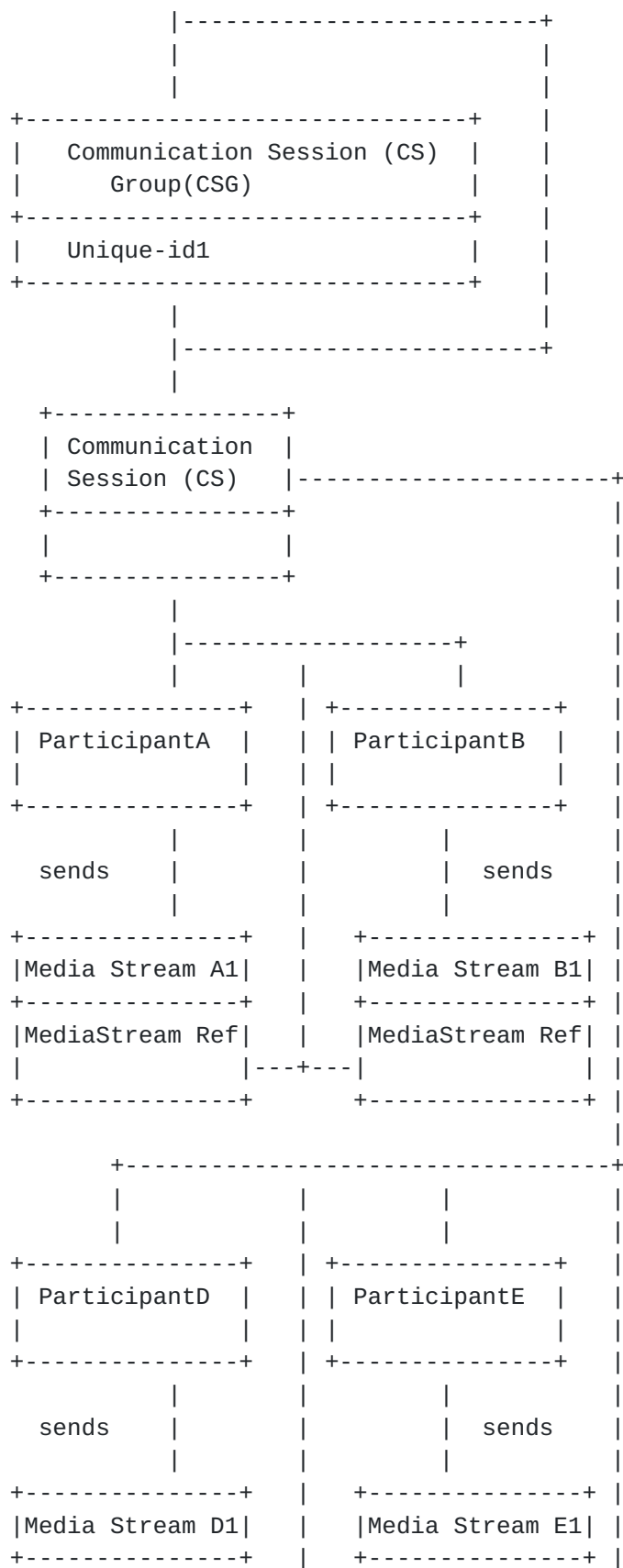
12.4.2. Case 2:

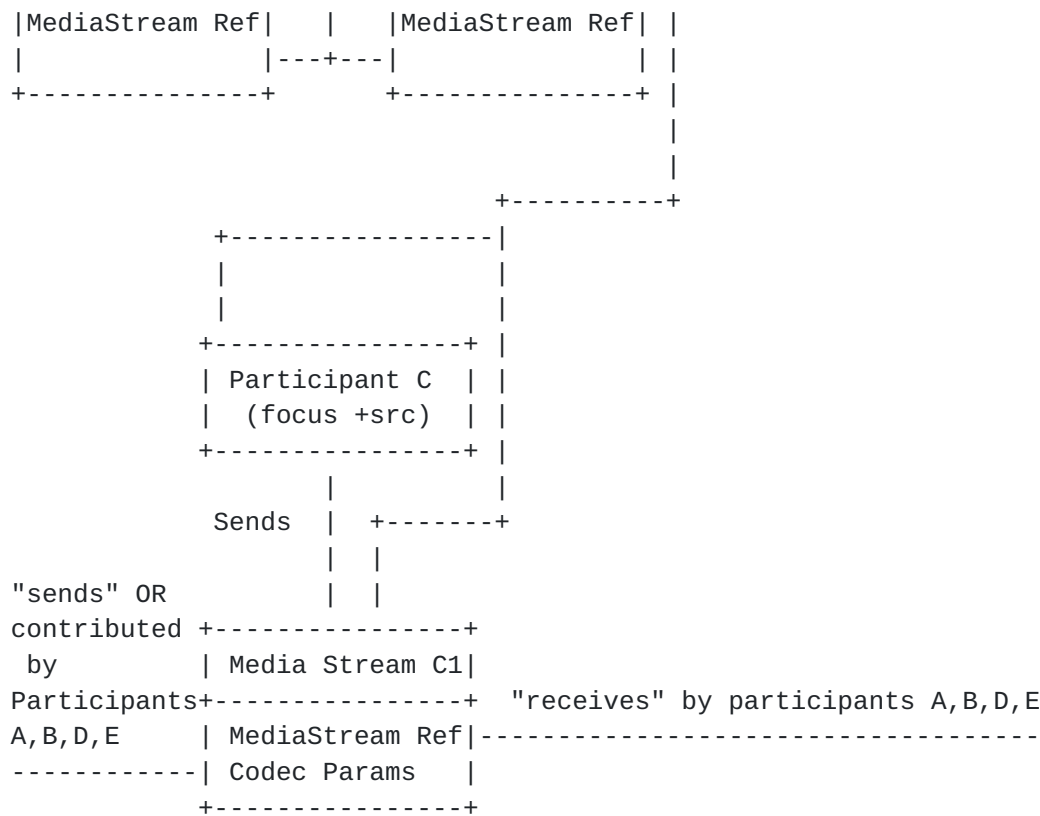
This is the usecase where there is a CS where one of the participant is focus (which is also SRC).



Instance Diagram:





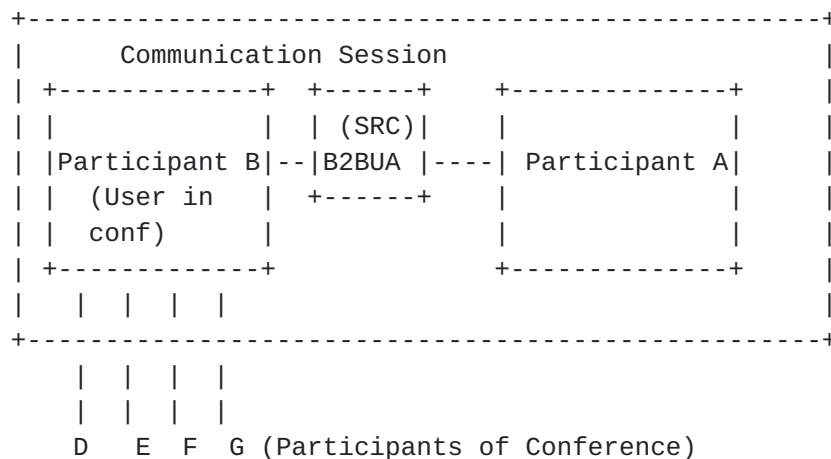


In this example we have two participants A and B who are part of a Communication Session(CS). One of the participants (C) is focus of a conference and also acts as SRC. The SRC (Participant C) being the Focus of the conference has access to the details of other participants. SRC (Participant C) sends the same through the metadata to SRS. In this instance diagram the Media Stream(mixed stream) sent by C has media streams contributed by conference participants (A, B, D and E). Participants A, B,D and E sends Media Streams A1, B1, D1 and E1 respectively. The media stream sent by Participant C(Focus) is received by all other participants of CS. For the sake of simplicity the "receives" line is not shown linked to all other participants.

NOTE: SRC (Participant C) can send mixed stream or seperate streams to SRS

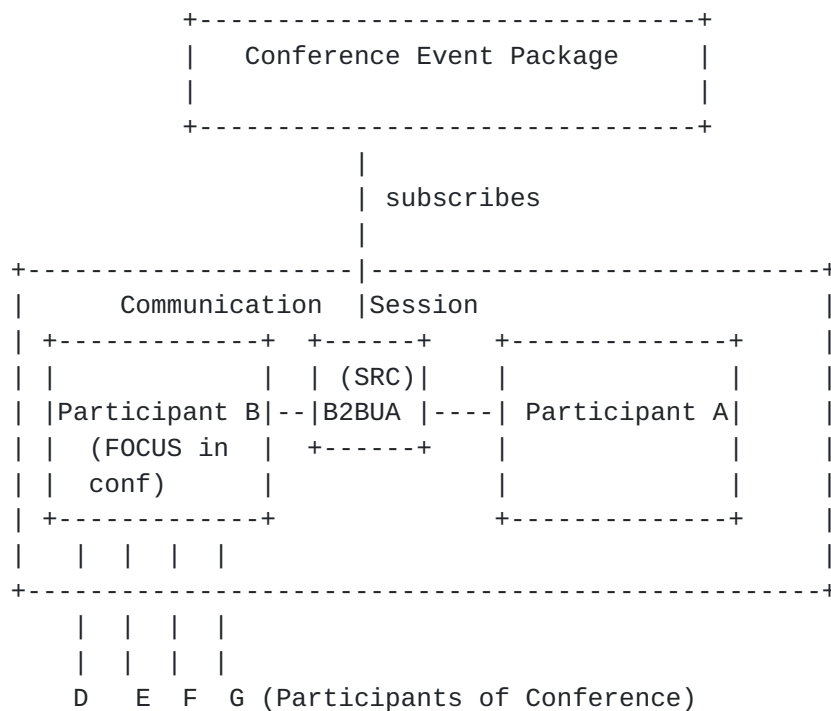
12.4.3. Case 3:

A CS where one of the participant is user and the SRC is a different entity like B2BUA. In this case the SRC may not know that one of the user is part of conference. Hence the instance diagram will not have information about the conference participants.



12.4.4. Case 4:

A CS where one of the participant is focus and the SRC is a different entity like B2BUA. In this case the participant which is focus sends "isfocus" in SIP message to SRC. The SRC subscribe to conference event package on seeing this "isfocus". SRC learns the details of other participants of conference from the conference package and send the same in metadata to SRS. The instance diagram for this use case is same as Case 1.



13. Appendix B: Metadata XML schema Instances

This section describes the metadata model XML instances for different use cases of SIPREC. For the sake of simplicity the complete SIP messages are NOT shown here.

13.1. Use case 1: Basic Call

Basic call between two Participants A(Ram) and B(Partha) who are part of one session. In this use case each participant sends two Media Streams. Media Streams sent by each participant is received all other participants of that CS in this use-case. Below is the initial snapshot sent by SRC that has complete metadata. For the sake of completeness even snippets of SDP is shown. For the sake of simplicity these use-cases assume the RS stream is unmixed.

```
Content-Type: application/SDP
...
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:96
a=sendonly
...
m=video 49174 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:97
a=sendonly
...
m=audio 51372 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:98
a=sendonly
...
m=video 49176 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:99
a=sendonly
....
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording'>
    <dataMode>complete</dataMode>
    <group id="urn:uuid:efe3930b-2a31-4e6a-a6ab-203fd7078302">
      <associate-time>2010-12-16T23:41:07Z</associate-time>
      <extensiondata>
        <!-- Standardized extension -->
        <call-center xmlns='urn:ietf:params:xml:ns:callcenter'>
          <supervisor>sip:alice@cisco.com</supervisor>
```



```
</call-center>
<mydata xmlns='http://example.com/my'>
  <structure>F00!</structure>
  <whatever>bar</whatever>
</mydata>
</extensiondata>
</group>
<session id="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
  <group-ref>urn:uuid:efe3930b-2a31-4e6a-a6ab-203fd7078302
  </group-ref>
  <start-time>2010-12-16T23:41:07Z</start-time>
  <extensiondata>
    <structure>F00!</structure>
    <whatever>bar</whatever>
  </extensiondata>
</session>
<participant
  id="urn:uuid:b2b7c112-5982-469d-9007-6ddbbecca64d3"
  session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
  <aor>sip:ram@blr.cisco.com</aor>
  <send>urn:uuid:8b53f3de-da39-4846-93c7-ee5e5f8f6f0b</send>
  <send>urn:uuid:50000c9b-9191-40a4-8231-5bcbca5e2b17</send>
  <recv>urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a</recv>
  <recv>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</recv>
  <associate-time>2010-12-16T23:41:07Z</associate-time>
  <extensiondata>
    <structure>F00!</structure>
    <whatever>bar</whatever>
  </extensiondata>
</participant>

<participant
  id="urn:uuid:cd27cfa1-2bdd-4830-a653-70374d10f103"
  session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
  <aor>sip:partha@blr.sonus.com</aor>
  <send>urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a</send>
  <send>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</send>
  <recv>urn:uuid:50000c9b-9191-40a4-8231-5bcbca5e2b17</recv>
  <recv>urn:uuid:8b53f3de-da39-4846-93c7-ee5e5f8f6f0b</recv>
  <associate-time>2010-12-16T23:41:07Z</associate-time>
  <extensiondata>
    <structure>F00!</structure>
    <whatever>bar</whatever>
  </extensiondata>
</participant>
<stream id="urn:uuid:50000c9b-9191-40a4-8231-5bcbca5e2b17"
  session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
  <label>96</label>
```



```
</stream>
<stream id="urn:uuid:8b53f3de-da39-4846-93c7-ee5e5f8f6f0b"
  session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeaeb425">
  <label>97</label>
</stream>
<stream id="urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a"
  session="urn:uuid:cd27cfa1-2bdd-4830-a653-70374d10f103">
  <label>98</label>
</stream>
<stream id="urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a"
  session="urn:uuid:cd27cfa1-2bdd-4830-a653-70374d10f103">
  <label>99</label>
</stream>
</recording>
```

13.2. Use case 2: Hold/resume

Basic call between two Participants A and B. This is the continuation of above use-case. One of the participants(say A) goes on hold and then resumes as part of the same session. The metadata snapshot looks as below

During hold

Content-Type: application/SDP

```
...
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:96
a=inactive
...
m=video 49174 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:97
a=inactive
...
m=audio 51372 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:98
a=sendonly
...
m=video 49176 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:99
a=sendonly
....
```

```
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording'>
    <dataMode>partial</dataMode>
    <participant
      id="urn:uuid:b2b7c112-5982-469d-9007-6ddbbecca64d3"
      session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
        <aor>sip:ram@blr.cisco.com</aor>
        <recv>urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a</recv>
        <recv>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</recv>
      </participant>
    <participant
      id="urn:uuid:b2b7c112-5982-469d-9007-6ddbbecca64d3"
      session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeae425">
        <aor>sip:partha@blr.cisco.com</aor>
        <send>urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a</send>
        <send>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</send>
      </participant>
    </recording>
```

During resume

The snapshot will look pretty much same as Use-case 1.

13.3. Use case 3: Basic Call with transfer

Basic call between two Participants A and B is connected as in Use-case 1. Transfer is initiated by one of the participants or by other entity(3PCC case). SRC sends a snapshot of the participant changes to SRS. In this instance participant A(Ram) drops out during the transfer and Participant C(Paul) joins the session. There can be two cases here, same session continues after transfer or a new session (e.g. REFER based transfer) is created

Transfer with same session retained - (.e.g. RE-INVITE based transfer). Participant A drops out and C is added to the same session. No change to session/group element. C will be new stream element which maps to RS SDP using the same labels in this instance.

```

Content-Type: application/SDP
...
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:96
a=sendonly
...
m=video 49174 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:97
a=sendonly
...
m=audio 51372 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:98
a=sendonly
...
m=video 49176 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:99
a=sendonly
....
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording'>
    <dataMode>partial</dataMode>
    <participant
      id="urn:uuid:cd27cfa1-2bdd-4830-a653-70374d10f103"
      session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeaeb425">
        <aor>sip:partha@blr.sonus.com</aor>
        <send>urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a</send>
        <send>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</send>
        <recv>urn:uuid:eb424026-6f54-4ef8-a4d0-bb658a1fc6cf</recv>
      </participant>
    </recording>
  </xml>

```



```
<recv>urn:uuid:01c47915-4777-11d8-bc70-0090272ff725</recv>
</participant>

<participant
  id="urn:uuid:02d9e6d5-9467-382e-8f9b-9300a64ac3cd"
  session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeaeb425">
  <aor>sip:paul@box.mit.com</aor>
  <send>urn:uuid:eb424026-6f54-4ef8-a4d0-bb658a1fc6cf</send>
  <send>urn:uuid:01c47915-4777-11d8-bc70-0090272ff725</send>
  <recv>urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a</recv>
  <recv>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</recv>
  <associate-time>2010-12-16T23:41:07Z</associate-time>
  <extensiondata>
    <structure>F00!</structure>
    <whatever>bar</whatever>
  </extensiondata>
</participant>

<stream id="urn:uuid:eb424026-6f54-4ef8-a4d0-bb658a1fc6cf"
  session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeaeb425">
  <label>96</label>
</stream>
<stream id="urn:uuid:01c47915-4777-11d8-bc70-0090272ff725"
  session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeaeb425">
  <label>97</label>
</stream>
<stream id="urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a"
  session="urn:uuid:cd27cfa1-2bdd-4830-a653-70374d10f103">
  <label>98</label>
</stream>
<stream id="urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a"
  session="urn:uuid:cd27cfa1-2bdd-4830-a653-70374d10f103">
  <label>99</label>
</stream>
</recording>
```

Transfer with new session - (.e.g. REFER based transfer). In this case new session is part of same grouping (done by SRC).

SRC may send an optional snapshot indicating stop for the old session.


```
<?xml version="1.0" encoding="UTF-8"?>
<recording xmlns='urn:ietf:params:xml:ns:recording'>
  <dataMode>Partial</dataMode>
  <session id="urn:uuid:855a5ded-8420-456d-a70f-6da1eeaeb425">
    <group-ref>urn:uuid:efe3930b-2a31-4e6a-a6ab-203fd7078302
    </group-ref>
    <stop-time>2010-12-16T23:41:07Z</stop-time>
    <extensiondata>
      <structure>F00!</structure>
      <whatever>bar</whatever>
    </extensiondata>
  </session>
  <participant
    id="urn:uuid:b2b7c112-5982-469d-9007-6ddbbecca64d3"
    session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeaeb425">
    <disassociate-time>2010-12-16T23:41:07Z</disassociate-time>
  </participant>

  <participant
    id="urn:uuid:cd27cfa1-2bdd-4830-a653-70374d10f103"
    session="urn:uuid:855a5ded-8420-456d-a70f-6da1eeaeb425">
      <aor>sip:partha@blr.sonus.com</aor>
      <disasociate-time>2010-12-16T23:41:07Z</disassociate-time>
    </participant>

</recording>
```

SRC sends a snapshot to indicate the participant change and new session information after transfer. In this example the same RS is used.

Content-Type: application/SDP

```
...
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:96
a=sendonly
...
m=video 49174 RTP/AVPF 96
a=rtpmap:96 H.264/90000
a=label:97
a=sendonly
...
m=audio 51372 RTP/AVP 0
a=rtpmap:0 PCMU/8000
a=label:98
```


a=sendonly

...

m=video 49176 RTP/AVPF 96

a=rtpmap:96 H.264/90000

a=label:99

a=sendonly

....

```
<?xml version="1.0" encoding="UTF-8"?>
  <recording xmlns='urn:ietf:params:xml:ns:recording'>
    <dataMode>partial</dataMode>
    <session id="urn:uuid:6DF2D9F8-D4C5-11E0-8DC5-04EE4724019B">
      <group-ref>urn:uuid:efe3930b-2a31-4e6a-a6ab-203fd7078302
      </group-ref>
      <start-time>2010-12-16T23:41:07Z</start-time>
      <extensiondata>
        <structure>F00!</structure>
        <whatever>bar</whatever>
      </extensiondata>
    </session>
    <participant
      id="urn:uuid:cd27cfa1-2bdd-4830-a653-70374d10f103"
      session="urn:uuid:6DF2D9F8-D4C5-11E0-8DC5-04EE4724019B">
      <aor>sip:partha@blr.sonus.com</aor>
      <send>urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a</send>
      <send>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</send>
      <recv>urn:uuid:eb424026-6f54-4ef8-a4d0-bb658a1fc6cf</recv>
      <recv>urn:uuid:01c47915-4777-11d8-bc70-0090272ff725</recv>
      <associate-time>2010-12-16T23:32:03Z</associate-time>
      <extensiondata>
        <structure>F00!</structure>
        <whatever>bar</whatever>
      </extensiondata>
    </participant>

    <participant
      id="urn:uuid:02d9e6d5-9467-382e-8f9b-9300a64ac3cd"
      session="urn:uuid:6DF2D9F8-D4C5-11E0-8DC5-04EE4724019B">
      <aor>sip:paul@box.mit.com</aor>
      <send>urn:uuid:eb424026-6f54-4ef8-a4d0-bb658a1fc6cf</send>
      <send>urn:uuid:01c47915-4777-11d8-bc70-0090272ff725</send>
      <recv>urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a</recv>
      <recv>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</recv>
      <associate-time>2010-12-16T23:41:07Z</associate-time>
      <extensiondata>
        <structure>F00!</structure>
        <whatever>bar</whatever>
      </extensiondata>
    </participant>
```



```
<stream id="urn:uuid:eb424026-6f54-4ef8-a4d0-bb658a1fc6cf"
  session="urn:uuid:6DF2D9F8-D4C5-11E0-8DC5-04EE4724019B">
  <label>96</label>
</stream>
<stream id="urn:uuid:01c47915-4777-11d8-bc70-0090272ff725"
  session="urn:uuid:6DF2D9F8-D4C5-11E0-8DC5-04EE4724019B">
  <label>97</label>
</stream>
<stream id="urn:uuid:f3373a7b-4958-4e55-8820-d03a191fb76a"
  session="urn:uuid:6DF2D9F8-D4C5-11E0-8DC5-04EE4724019B">
  <label>98</label>
</stream>
<stream id="urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a"
  session="urn:uuid:6DF2D9F8-D4C5-11E0-8DC5-04EE4724019B">
  <label>99</label>
</stream>
</recording>
```

14. References

14.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2141] Moats, R., "URN Syntax", [RFC 2141](#), May 1997.
- [RFC3261] 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.
- [RFC3688] Mealling, M., "The IETF XML Registry", [BCP 81](#), [RFC 3688](#), January 2004.
- [RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", [RFC 3339](#), July 2002.
- [RFC4566] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [RFC4574] Levin, O. and G. Camarillo, "The Session Description Protocol (SDP) Label Attribute", [RFC 4574](#), August 2006.
- [RFC4796] Hautakorpi, J. and G. Camarillo, "The Session Description Protocol (SDP) Content Attribute", [RFC 4796](#),

February 2007.

- [RFC3840] Rosenberg, J., Schulzrinne, H., and P. Kyzivat, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", [RFC 3840](#), August 2004.
- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique Identifier (UUID) URN Namespace", [RFC 4122](#), July 2005.

14.2. Informative References

- [RFC6341] Rehor, K., Portman, L., Hutton, A., and R. Jain, "Use Cases and Requirements for SIP-Based Media Recording (SIPREC)", [RFC 6341](#), August 2011.
- [I-D.ietf-siprec-architecture]
Hutton, A., Portman, L., Jain, R., and K. Rehor, "An Architecture for Media Recording using the Session Initiation Protocol", [draft-ietf-siprec-architecture-02](#) (work in progress), April 2011.
- [RFC2648] Moats, R., "A URN Namespace for IETF Documents", [RFC 2648](#), August 1999.

Authors' Addresses

Ram Mohan Ravindranath
Cisco Systems, Inc.
Cessna Business Park,
Kadabeesanahalli Village, Varthur Hobli,
Sarjapur-Marathahalli Outer Ring Road
Bangalore, Karnataka 560103
India

Email: rmohanr@cisco.com

Parthasarathi Ravindran
Sonus Networks
Prestige Shantiniketan - Business Precinct
Whitefield Road
Bangalore, Karnataka 560066
India

Email: pravindran@sonusnet.com

Paul Kyzivat
Unaffiliated
Boxborough, MA
USA

Email: pkyzivat@alum.mit.edu