       Interworking between the Session Initiation Protocol (SIP) and the
          Extensible Messaging and Presence Protocol (XMPP): Groupchat

Abstract

   This document defines a bidirectional protocol mapping for the
   exchange of instant messages in the context of a multi-party chat
   session among users of the Session Initiation Protocol (SIP) and
   users of the Extensible Messaging and Presence Protocol (XMPP).
   Specifically, this document defines a mapping between the SIP-based
   Message Session Relay Protocol (MSRP) and the XMPP Multi-User Chat
   (MUC) extension.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7702.

Copyright Notice

Table of Contents

1. Introduction

   Both the Session Initiation Protocol (SIP) [RFC3261] and the
   Extensible Messaging and Presence Protocol (XMPP) [RFC6120] can be
   used for the purpose of multi-party text chat over the Internet. To
   ensure interworking between these technologies, it is important to
   define bidirectional protocol mappings.

   The architectural assumptions underlying such protocol mappings are
   provided in [RFC7247], including the mapping of addresses and error
   conditions. This document specifies mappings for multi-party text
   chat sessions (often called "groupchat"); specifically, this document
   defines a mapping between the XMPP Multi-User Chat (MUC) extension
   [XEP-0045] and SIP-based multi-party chat using Message Session Relay
   Protocol (MSRP) [RFC4975] as specified in [RFC7701].

   Both MUC and MSRP contain a large set of features, such as the
   ability to administer rooms, kick out and ban users, reserve a
   nickname within a room, change room subject, enable room moderation,
   and destroy the room. This document covers only a basic subset of
   groupchat features: joining a room, establishing or changing (but not
   permanently registering) a room nickname, modifying presence
   information within the room, sending a message to all participants,
   sending a private message to a single participant, inviting another
   user to the room, and leaving the room. Future documents might
   define mappings for additional features beyond this set.

2. Intended Audience

   The documents in this series are intended for use by software
   developers who have an existing system based on one of these
   technologies (e.g., SIP), and who would like to enable communication
   from that existing system to systems based on the other technology
   (e.g., XMPP). We assume that readers are familiar with the core
   specifications for both SIP [RFC3261] and XMPP [RFC6120], with the
   base document for this series [RFC7247], and with the following
   groupchat-related specifications:

   o Multi-party Chat Using MSRP [RFC7701]

   o Multi-User Chat [XEP-0045]

3. Terminology

   A number of technical terms used here are defined in [RFC3261],
   [RFC4975], [RFC6120], and [XEP-0045].

   In flow diagrams, MSRP traffic is shown using arrows such as "%%%>",
   SIP traffic is shown using arrows such as "***>", XMPP traffic is
   shown using arrows such as "...>".

   In protocol flows and examples, provisional SIP responses have been
   elided for the sake of brevity.

   The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
   "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and
   "OPTIONAL" in this document are to be interpreted as described in
   [RFC2119].

4. Architectural Assumptions

   XMPP and MSRP differ in their assumptions regarding groupchat
   traffic. In XMPP, a message of type "groupchat" is just another
   stanza and is handled directly by an XMPP server or routed to an
   associated server component for multi-user chat. By contrast,
   sessions (including groupchat sessions) in MSRP are considered to be
   a type of media (similar to audio/video sessions): signaling to set
   up, manage, and tear down the session is handled by a "conference
   focus" [RFC4353] (here we assume via SIP), but the session data
   itself is handled by a separate entity called an MSRP switch. How
   the conference focus and MSRP switch communicate is a matter of
   implementation and deployment.

   An architectural diagram for a possible gateway deployment is shown
   below, where the entities have the following significance:

   o romeo@example.org -- a SIP user.

   o romeo@example.org;gr=dr4hcr0st3lup4c -- a particular endpoint
      associated with the SIP user.

   o example.org -- a SIP proxy with an associated SIP-to-XMPP gateway
      ("S2X GW") to XMPP.

   o chat.example.org -- a SIP-based conference focus and MSRP switch
      with an associated MSRP-to-SIP gateway ("M2X GW") to XMPP.

   o montague@chat.example.org -- a conference at an MSRP switch; not
      shown in diagram.

   o juliet@example.com -- an XMPP user.

   o juliet@example.com/yn0cl4bnw0yr3vym -- a particular endpoint
      associated with the XMPP user.

   o example.com -- an XMPP server with an associated XMPP-to-SIP
      gateway ("X2S GW") to SIP and an XMPP-to-MSRP gateway ("X2M GW")
      to MSRP.

   o rooms.example.com -- an XMPP MUC service associated with
      example.com.

   o capulet@rooms.example.com -- a chat room at an XMPP MUC service;
      not shown in diagram.

   These are logical entities, and several of them might be co-located
   in the same physical entity. For example, the SIP conference focus
   and MSRP switch and associated gateways, or the XMPP server and MUC
   service and associated gateways, might be part of the same deployed
   code. In addition, it is likely that an XMPP service would not have
   separate gateways for XMPP-to-SIP translation and XMPP-to-MSRP
   translation, but would instead have a single gateway.

```
#####################################################################
#                                                                   #
#                 +------------------+                              #
#  &&&&&&&&&&&&&&&| chat.example.org |<%%%%%%%%%                    #
#  &            &&&| (MSRP switch) +-----+         %                #
#  &          &   +--------------| M2X |           %                #
#  &          &               %   | GW |           %                #
#  &          &               %   +-----+          %                #
#  &          &               %      :             %                #
#  &          &               %   ///////////////////////////////#
#  &          &               %  /  :              %                #
#  &          &               % /   :           +-----+            #
#  &          &               % /   :           | X2M |            #
#  &          &               % /   : +-------| GW  |---+          #
#  &          &               % /   :.>|       +-----+   |          #
#  &          &               % /     |                  |          #
#  & +------------------+      %   /  +-----+             |          #
#  & | chat.example.org |<*******/*| X2S | example.com   |          #
#  & | (conference      |  %  **/*| GW  | (XMPP server) |          #
#  & | focus)    +-----+ %   * / +-----+    |          #
#  & +-----------| S2X | %   * /    |    +------------------+  #
#  &        *    | GW  |......*./....>|    | rooms.example.com |  #
#  &        *    +-----+ %   * /    +-----| (MUC service)     |  #
#  &        *            %   * /     ^ : +------------------+  #
#  & +--------------+    %   * /     : :                        #
#  &&| example.org  |<********* /      : :                        #
#  | (SIP proxy) +-----+ %     /      : :                        #
#  +-------------| S2X | %     /      : :                        #
#        *       | GW  |......./........ :                        #
#        *       +-----+ %    /      :                           #
#        *              %    /      :                           #
#     romeo@example.org      /      juliet@example.com          #
#     ;gr=dr4hcr0st3lup4c    /      /yn0cl4bnw0yr3vym            #
#                           /                                    #
#     --SIP/MSRP DOMAIN--   /       --XMPP DOMAIN--              #
#                          /                                    #
#####################################################################
```
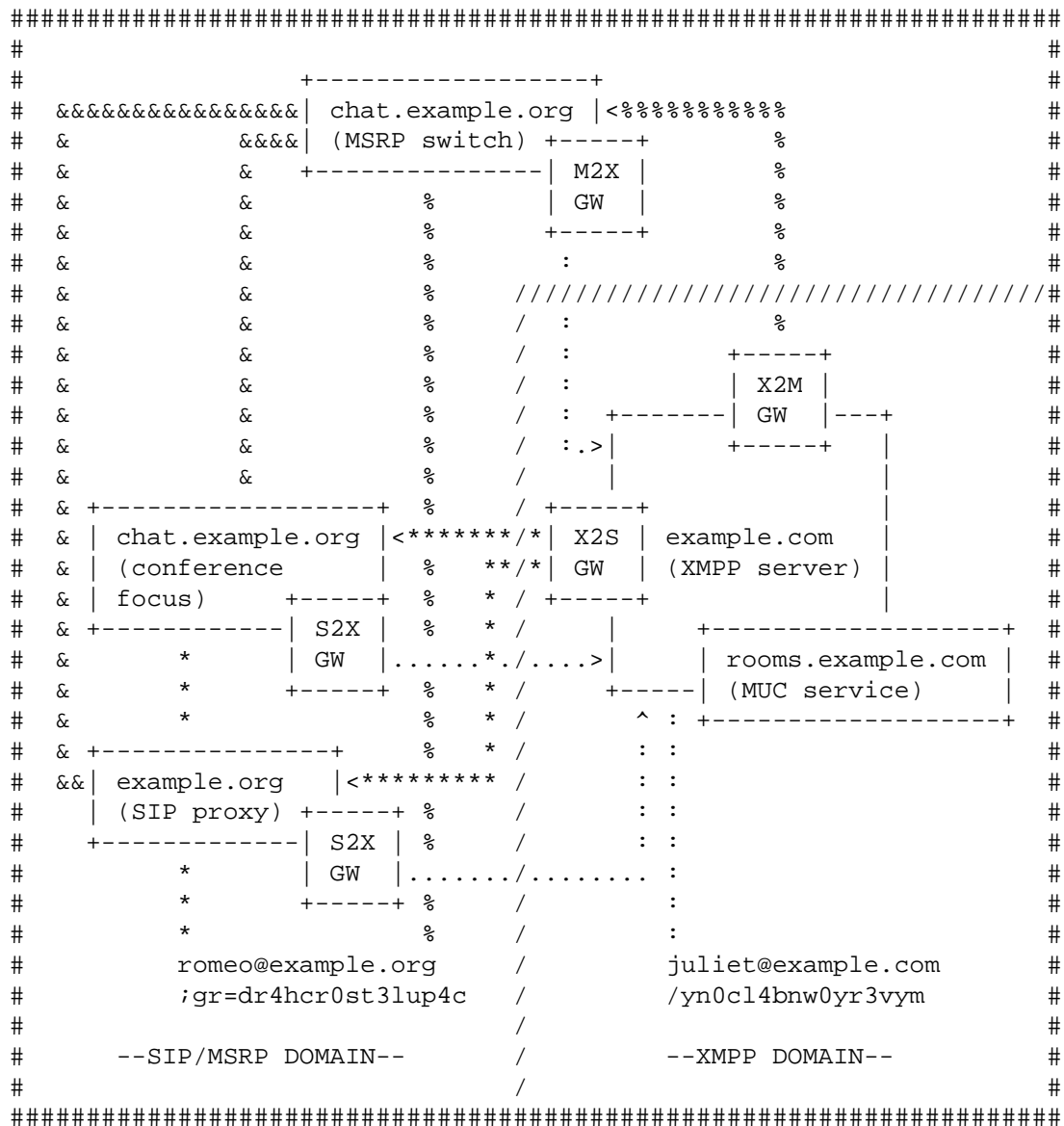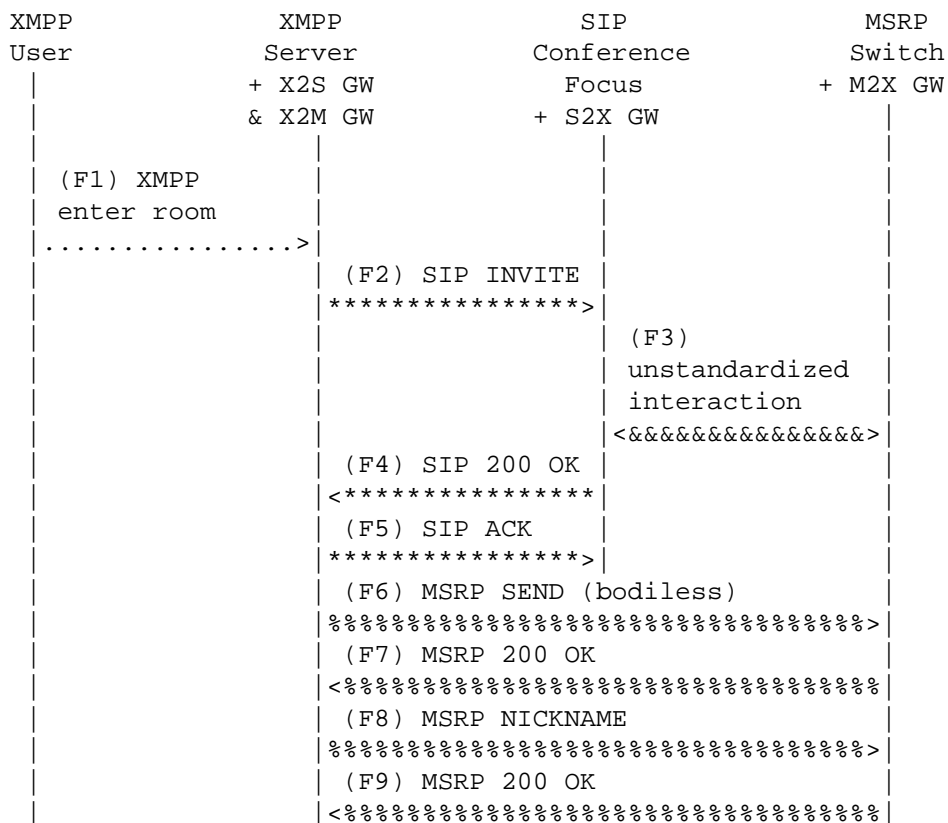
    Legend:
        . = XMPP
        % = MSRP
        * = SIP
        & = unstandardized communication paths
        / = separation of administrative domains

              Figure 1: Logical Deployment Architecture

   In SIP, there is no necessity for a SIP user such as
   romeo@example.org to make use of his SIP proxy in order to join a
   chat room on the XMPP network; for example, he could try to directly
   find a SIP service at example.com or independently locate a SIP-to-
   XMPP gateway. Although, as a simplifying assumption, this document
   shows the more expected path of using one's "home" SIP proxy and
   shows gateways as associated with the sending domain, nothing in this
   document ought to be construed as discouraging other deployment
   architectures or communication paths (e.g., services hosting their
   own inbound gateways).

5. Multi-party Messaging Session from XMPP MUC to MSRP

   This section describes how to map an XMPP MUC session to an MSRP
   Multi-party Messaging session. The following diagram outlines the
   overall protocol flow of a sample session, which includes some
   optional exchanges (such as sending messages, changing a nickname,
   and inviting another user).

```
   XMPP                 XMPP                  SIP                MSRP
   User                Server             Conference           Switch
    |                  + X2S GW              Focus            + M2X GW
    |                  & X2M GW            + S2X GW              |
    |                    |                    |                  |
    | (F1) XMPP          |                    |                  |
    | enter room         |                    |                  |
    |...............>|                    |                  |
    |                    | (F2) SIP INVITE |                  |
    |                    |***************>|                  |
    |                    |                    | (F3)             |
    |                    |                    | unstandardized   |
    |                    |                    | interaction      |
    |                    |                    |<&&&&&&&&&&&&&&&>|
    |                    | (F4) SIP 200 OK |                  |
    |                    |<***************|                  |
    |                    | (F5) SIP ACK    |                  |
    |                    |***************>|                  |
    |                    | (F6) MSRP SEND (bodiless)            |
    |                    |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%>|
    |                    | (F7) MSRP 200 OK                     |
    |                    |<%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
    |                    | (F8) MSRP NICKNAME                   |
    |                    |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%>|
    |                    | (F9) MSRP 200 OK                     |
    |                    |<%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
```

```
|                       |  (F10) SIP          |                   |
|                       |  SUBSCRIBE          |                   |
|                       |  Event:             |                   |
|                       |  conference         |                   |
|                       |***************>|                        |
|                       |  (F11) SIP 200 OK|                       |
|                       |<***************|                         |
|                       |  (F12) SIP NOTIFY|                       |
|                       |<***************|                         |
|                       |  (F13) SIP 200 OK|                       |
|                       |***************>|                        |
| (F14) XMPP            |                     |                   |
| presence              |                     |                   |
|<...............|                            |                   |
| (F15) XMPP            |                     |                   |
| MUC subject           |                     |                   |
|<...............|                            |                   |
.                       .                     .                   .
.                       .                     .                   .
| (F16) XMPP            |                     |                   |
| groupchat             |                     |                   |
| message               |                     |                   |
|...............>|                            |                   |
|                       |  (F17) MSRP SEND                        |
|                       |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%>|
|                       |  (F18) MSRP 200 OK                      |
|                       |<%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
| (F19) XMPP            |                     |                   |
| groupchat             |                     |                   |
| message               |                     |                   |
|<...............|                            |                   |
.                       .                     .                   .
.                       .                     .                   .
| (F20) XMPP            |                     |                   |
| private               |                     |                   |
| message               |                     |                   |
|...............>|                            |                   |
|                       |  (F21) MSRP SEND                        |
|                       |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%>|
|                       |  (F22) MSRP 200 OK                      |
|                       |<%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
.                       .                     .                   .
.                       .                     .                   .
| (F23) XMPP            |                     |                   |
| presence:             |                     |                   |
| change nick           |                     |                   |
|...............>|                            |                   |
```

```
|                       |  (F24) MSRP NICKNAME           |
|                       |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%>|
|                       |  (F25) MSRP 425 Error          |
|                       |<%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|
|  (F26) XMPP           |                       |        |
|  presence             |                       |        |
|  error                |                       |        |
|<...............|      |                       |        |
.                .      .                       .        .
.                .      .                       .        .
|  (F27) XMPP           |                       |        |
|  message:             |                       |        |
|  invite               |                       |        |
|...............>|      |                       |        |
|                       |  (F28) SIP            |        |
|                       |  REFER                |        |
|                       |***************>|      |        |
|                       |  (F29) SIP            |        |
|                       |  200 OK               |        |
|                       |<***************|      |        |
|                       |  (F30) SIP            |        |
|                       |  NOTIFY               |        |
|                       |<***************|      |        |
.                .      .                       .        .
.                .      .                       .        .
|  (F31) XMPP           |                       |        |
|  presence:            |                       |        |
|  exit room            |                       |        |
|...............>|      |                       |        |
|                       |  (F32) SIP BYE        |        |
|                       |***************>|      |        |
|                       |  (F33) SIP            |        |
|                       |  200 OK               |        |
|                       |<***************|      |        |
|  (F34) XMPP           |                       |        |
|  presence             |                       |        |
|  unavailable          |                       |        |
|<...............|      |                       |        |
|                |      |                       |        |
```

Detailed protocol flows and mappings are provided in the following
sections.

5.1. Enter Room

   As defined in the XMPP Multi-User Chat (MUC) specification
   [XEP-0045], when an XMPP user (say, "juliet@example.com") wants to
   join a groupchat room (say, "montague@chat.example.org"), she sends a
   directed <presence/> stanza [RFC6121] to that chat room. In her
   request she also specifies the nickname she wants to use within the
   room (say, "JuliC"); in XMPP this room nickname is the resourcepart
   of an occupant JID (thus "montague@chat.example.org/JuliC"). The
   joining client signals its ability to speak the multi-user chat
   protocol by including in the initial presence stanza an empty
   element qualified by the 'http://jabber.org/protocol/muc' namespace.

   Example 1: Juliet Enters Room (F1)

   | <presence from='juliet@example.com/yn0cl4bnw0yr3vym'
   | to='montague@chat.example.org/JuliC'>
   | <x xmlns='http://jabber.org/protocol/muc'/>
   | </presence>

   Upon receiving such a presence stanza, the XMPP server needs to
   determine the identity of the domainpart in the 'to' address, which
   it does by following the procedures discussed in [RFC7247]. Here we
   assume that the XMPP server has determined the domain is serviced by
   a SIP server, that it contains or has available to it an XMPP-to-SIP
   gateway or connection manager (which enables it to speak natively to
   SIP servers), and that it hands off the presence stanza to the
   XMPP-to-SIP gateway.

   Because a multi-user chat service accepts the presence stanza shown
   above as a request to enter a room, the XMPP-to-SIP gateway
   transforms it into a SIP INVITE request.

Example 2: SIP Mapping of Room Join (F2)

```
|   INVITE sip:montague@chat.example.org SIP/2.0
|   To: <sip:montague@chat.example.org>
|   From: "Juliet" <sip:juliet@example.com>;tag=786
|   Contact: <sip:juliet@example.com>;gr=yn0cl4bnw0yr3vym
|   Call-ID: BC466C1C-E01D-4FD1-B766-9AD174BAF2E7
|   CSeq: 1 INVITE
|   Content-Type: application/sdp
|   Content-Length: ...
|
|   c=IN IP4 x2s.example.org
|   m=message 7654 TCP/MSRP *
|   a=accept-types:text/cpim
|   a=accept-wrapped-types:text/plain text/html
|   a=path:msrp://x2m.example.com:7654/jshA7weztas;tcp
|   a=chatroom:nickname private-messages
```

Here the Session Description Protocol (SDP) offer specifies the XMPP-
to-MSRP gateway on the XMPP side (in the SDP 'path' attribute
specified in [RFC4975]) as well as other particulars of the session.

   There is no direct mapping for the MSRP URIs. In fact, an MSRP
   URI identifies a session of instant messages at a particular
   device; it is ephemeral and has no meaning outside the scope of
   that session. The authority component of the MSRP URI here MUST
   contain the XMPP-to-MSRP gateway hostname or numeric IP address
   (as well as, in accordance with [RFC4975], an explicit port
   number).

The mapping of XMPP syntax elements to SIP and [RFC4566] syntax
elements MUST be as shown in the following table.

Table 1: Message Syntax Mapping from XMPP to SIP/SDP

XMPP Element or Attribute	SIP Header or SDP Contents
from	From
to (without the /nick)	To

As shown in the foregoing example and described in [RFC7247], the
XMPP-to-SIP gateway MUST map the bare JID ("localpart@domainpart") of
the XMPP sender to the SIP From header and include the resourcepart
of the full JID as the Globally Routable User Agent URI (GRUU)

portion [RFC5627] of the SIP URI. However, note that a SIP response
uses the same From and To as in the SIP request, whereas an XMPP
response swaps the from and to attributes.

Here we assume that the SIP conference focus accepts the session
establishment. The Contact header field of the SIP 200 OK response
includes the 'isfocus' feature tag specified in [RFC4353] along with
other relevant feature tags. The conference focus also includes an
answer session description that acknowledges the choice of media,
specifies the MSRP URI of the switch (in the 'path' attribute
specified in [RFC4975]), and contains the extensions specified in
[RFC7701].

Example 3: Chat Room Accepts Session Establishment (F4)

```
|   SIP/2.0 200 OK
|   From: "Juliet" <sip:juliet@example.com>;tag=786
|   To: <sip:montague@chat.example.org>;tag=087js
|   Call-ID: BC466C1C-E01D-4FD1-B766-9AD174BAF2E7
|   CSeq: 1 INVITE
|   Contact: <sip:montague@chat.example.org;transport=tcp>;isfocus
|   Content-Type: application/sdp
|   Content-Length: ...
|
|   v=0
|   c=IN IP4 example.org
|   s=-
|   m=message 12763 TCP/MSRP *
|   a=accept-types:message/cpim
|   a=accept-wrapped-types:text/plain text/html
|   a=path:msrp://chat.example.org:12763/kjhd37s2s20w2a;tcp
|   a=chatroom:nickname private-messages
```

Upon receiving such a response, the XMPP-to-SIP gateway sends a SIP
ACK to the conference focus on behalf of the joining user.

Example 4: Gateway Sends ACK to Conference Focus (F5)

```
|   ACK sip:montague@chat.example.org SIP/2.0
|   To: <sip:montague@chat.example.org>;tag=087js
|   From: "Juliet" <sip:juliet@example.com>;tag=786
|   Call-ID: BC466C1C-E01D-4FD1-B766-9AD174BAF2E7
|   CSeq: 2 ACK
```

In accordance with [RFC4975], the gateway sends a bodiless MSRP
message (F6) to the switch immediately upon establishing the
connection, and the switch acknowledges that message (F7).

5.2. Set Nickname

   If the chat room server accepted the session, the XMPP-to-MSRP
   gateway sets up the nickname as received in the presence stanza
   (i.e., the resourcepart of the 'to' address, such as "JuliC" in
   "montague@chat.example.org/JuliC"). This is done using the extension
   specified in [RFC7701].

   Example 5: Gateway Sets Up Nickname (F8)

   | MSRP a786hjs2 NICKNAME
   | To-Path: msrp://montague@chat.example.org:12763/kjhd37s2s20w2a;tcp
   | From-Path: msrp://x2m.example.com:7654/jshA7weztas;tcp
   | Use-Nickname: "JuliC"
   | -------a786hjs2

   The MSRP switch analyzes the existing allocation of nicknames,
   accepts the nickname proposal, and answers with a 200 response.

   Example 6: MSRP Switch Accepts Nickname Proposal (F9)

   | MSRP a786hjs2 200 OK
   | To-Path: msrp://x2m.example.com:7654/jshA7weztas;tcp
   | From-Path: msrp://montague@chat.example.org:12763/kjhd37s2s20w2a
   | ;tcp
   | -------a786hjs2

   This section assumes that the nickname request is successful. The
   error flow resulting from a nickname conflict is described under
   Section 5.6.

5.3. Conference Subscription

   As mentioned in [RFC7701], the joining user will typically also
   subscribe to a conference event package (see [RFC4575] and [RFC6502])
   at the focus. Although such a subscription is not required by
   [RFC7701] in practice the temporary and context-dependent presence
   subscriptions and room rosters involved in joining an XMPP MUC room
   are best mapped to the conference event package.

Example 7: Gateway Subscribes to the Conference (F10)

```
|   SUBSCRIBE sip:montague@chat.example.org SIP/2.0
|   To: <sip:montague@chat.example.org>;tag=087js
|   From: "Juliet" <sip:juliet@example.com>;tag=786
|   Contact: <sip:juliet@example.com>;gr=yn0cl4bnw0yr3vym
|   Call-ID: BC466C1C-E01D-4FD1-B766-9AD174BAF2E7
|   CSeq: 3 SUBSCRIBE
|   Event: conference
|   Expires: 600
|   Accept: application/conference-info+xml
|   Allow-Events: conference
|   Content-Length: 0
```

The focus will accept or reject the request based on local policy.

Example 8: Focus Accepts Subscription Request (F11)

```
|   SIP/2.0 200 OK
|   To: <sip:montague@chat.example.org>;tag=087js
|   From: "Juliet" <sip:juliet@example.com>;tag=786
|   Call-ID: BC466C1C-E01D-4FD1-B766-9AD174BAF2E7
|   CSeq: 3 SUBSCRIBE
|   Contact: <sip:montague@chat.example.org;transport=tcp>;isfocus
|   Expires: 600
|   Content-Length: 0
```

If the conference focus accepts the request to enter a room, the XMPP
user expects to receive back presence information from all the
existing occupants of the room. To make this happen, the XMPP-to-SIP
gateway subscribes to the conference event package [RFC4575] at the
focus.

5.4. Presence Broadcast

When the conference event package subscription is completed, the
focus sends to the XMPP-to-SIP gateway a NOTIFY request containing
the presence information of all the existing occupants, represented
using the format defined in [RFC4575].

   Example 9: Conference Focus Sends Presence Information (F12)

```
| NOTIFY sip:montague@chat.example.org SIP/2.0
| To: "Juliet" <sip:juliet@example.com>;tag=786
| From: <sip:montague@chat.example.org>;tag=087js
| Call-ID: BC466C1C-E01D-4FD1-B766-9AD174BAF2E7
| CSeq: 4 NOTIFY
| Event: conference
| Subscription-State: active;expires=3600
| Content-Type: application/conference-info+xml
| Content-Length: ...
|
| <conference-info version="0" state="full"
|     entity="sip:3402934234@chat.example.org">
|   <conference-description>
|     <subject>Today in Verona</subject>
|     <conf-uris>
|       <entry>
|         <uri>tel:+18882934234</uri>
|       </entry>
|     </conf-uris>
|   </conference-description>
|   <users>
|     <user entity="sip:montague@chat.example.org;gr=Romeo"
|           state="full">
|       <display-text>Romeo</display-text>
|       <roles>
|         <entry>participant</entry>
|       </roles>
|       <associated-aors>
|         <entry>
|           <uri>xmpp:romeo@example.org/dr4hcr0st3lup4c</uri>
|         </entry>
|       </associated-aors>
|       <endpoint entity="sip:montague@chat.example.org;gr=Romeo"
|                 state="full">
|         <status>connected</status>
|         <joining-info>
|           <when>2013-12-12T10:01:03.691128+01:00</when>
|         </joining-info>
|         <media id="211835820">
|           <type>message</type>
|         </media>
|       </endpoint>
|     </user>
|     <user entity="sip:montague@chat.example.org;gr=Ben"
|           state="full">
|       <display-text>Ben</display-text>
```

```
   |          <roles>
   |            <entry>participant</entry>
   |          </roles>
   |          <endpoint entity="sip:montague@chat.example.org;gr=Ben"
   |                    state="full">
   |            <status>connected</status>
   |            <media id="211835821">
   |              <type>message</type>
   |            </media>
   |          </endpoint>
   |        </user>
   |        <user entity="sip:montague@chat.example.org;gr=JuliC"
   |              state="full">
   |          <display-text>JuliC</display-text>
   |          <roles>
   |            <entry>participant</entry>
   |          </roles>
   |           <endpoint entity="sip:montague@chat.example.org;gr=JuliC"
   |                     state="full">
   |             <status>connected</status>
   |             <media id="211835822">
   |               <type>message</type>
   |             </media>
   |           </endpoint>
   |        </user>
   |      </users>
   |   </conference-info>
```

The syntax mapping from the RFC 4575 payload to the XMPP participant
list MUST be as shown in the following table. (Mappings for elements
not mentioned are undefined.)

Table 2: Participant list mapping

RFC 4575 Element or Attribute	XMPP Element or Attribute
<conference-info/> 'entity'	room JID
<subject/>	room subject
<user/> 'entity'	occupant JID
<display-text/>	participant nickname
<endpoint/> 'entity'	occupant JID
<user/> 'associated-aors'	user full JID (if avail.)

   Upon receiving such a response, the XMPP-to-SIP gateway sends a SIP
   200 OK response to the conference focus (example not shown) and
   translates the participant list into a series of XMPP presence
   stanzas.

   Example 10: XMPP Mapping of Chat Room Presence (F14)

   | <presence from='montague@chat.example.org/Romeo'
   | to='juliet@example.com/yn0cl4bnw0yr3vym'>
   | <x xmlns='http://jabber.org/protocol/muc#user'>
   | <item affiliation='none' role='participant'/>
   | </x>
   | </presence>

   | <presence from='montague@chat.example.org/Ben'
   | to='juliet@example.com/yn0cl4bnw0yr3vym'>
   | <x xmlns='http://jabber.org/protocol/muc#user'>
   | <item affiliation='none' role='participant'/>
   | </x>
   | </presence>

   | <presence from='montague@chat.example.org/JuliC'
   | to='juliet@example.com/yn0cl4bnw0yr3vym'>
   | <x xmlns='http://jabber.org/protocol/muc#user'>
   | <item affiliation='none' role='participant'/>
   | <status code='110'/>
   | </x>
   | </presence>

   If the NOTIFY request included a subject, the gateway converts that
   into a separate XMPP message.

   Example 11: XMPP Mapping of Chat Room Subject (F15)

   | <message from='montague@chat.example.org/mayor'
   | to='juliet@example.com/yn0cl4bnw0yr3vym'
   | id='mbh2vd68'>
   | <subject>Today in Verona</subject>
   | </message>

   The mapping of SIP and [RFC4575] payload syntax elements to XMPP
   syntax elements MUST be as shown in the following table. (Mappings
   for elements not mentioned are undefined.)

   Table 3: Message Syntax Mapping from SIP to XMPP

```
+------------------------------+----------------------------+
| SIP Header or RFC 4575 Contents | XMPP Element or Attribute |
+------------------------------+----------------------------+
|   <user/> 'entity'           |  from                      |
|   To with <display-text>     |  occupant JID              |
|   <role>participant</role>   |  role='participant'        |
|   [N/A]                      |  affiliation='none'        |
+------------------------------+----------------------------+
```

5.5. Exchange Messages

   Once the user has joined the chat room, the user can exchange an
   unbounded number of messages, both public and private.

   The mapping of XMPP syntax elements to MSRP syntax elements MUST be
   as shown in the following table. (Mappings for elements not
   mentioned are undefined.)

   Table 4: Message Syntax Mapping from XMPP Message to MSRP

```
+----------------------------+----------------------------+
| XMPP Element or Attribute  | CPIM Header                |
+----------------------------+----------------------------+
| to                         | To                         |
| from                       | From                       |
| <body/>                    | body of the SEND request   |
+----------------------------+----------------------------+
```

5.5.1. Send a Message to All Occupants

   When Juliet wants to sends a message to all other occupants in the
   room, she sends a message of type "groupchat" to the <room@service>
   itself (in our example, <montague@chat.example.org>).

   The following examples show an exchange of a public message.

   Example 12: Juliet Sends Message to All Occupants (F16)

```
|   <message from='juliet@example.com/yn0cl4bnw0yr3vym'
|            to='montague@chat.example.org'
|            type='groupchat'
|            id='lzfed24s'>
|        <body>Who knows where Romeo is?</body>
|   </message>
```

Upon receiving such a message, the XMPP-to-MSRP gateway translates it
into an MSRP SEND message.

Example 13: Gateway Maps XMPP Message to MSRP (F17)

```
|   MSRP a786hjs2 SEND
|   To-Path: msrp://chat.example.org:12763/kjhd37s2s20w2a;tcp
|   From-Path: msrp://x2m.example.com:7654/jshA7weztas;tcp
|   Message-ID: 87652491
|   Byte-Range: 1-*/*
|   Content-Type: message/cpim
|
|   To: <sip:montague@chat.example.org>
|   From: "Juliet" <sip:juliet@example.com>
|   DateTime: 2008-10-15T15:02:31-03:00
|   Content-Type: text/plain
|
|   Who knows where Romeo is?
|   -------a786hjs2$
```

Upon receiving the SEND request, if the request either contains a
Failure-Report header field value of "yes" or does not contain a
Failure-Report header at all, the MSRP switch immediately generates
and sends a response.

Example 14: MSRP Switch Returns 200 OK (F18)

```
|   MSRP d93kswow 200 OK
|   To-Path: msrp://x2m.example.com:7654/jshA7weztas;tcp
|   From-Path: msrp://chat.example.org:12763/kjhd37s2s20w2a;tcp
|   -------d93kswow$
```

Since an XMPP MUC room could be moderated and an XMPP user cannot be
sure whether her message has been accepted without receiving it back
from the server, [XEP-0045] states that the sender needs to receive a
reflected copy of the message it sent. So, in this scenario, the
XMPP-to-MSRP gateway has to reflect the message back to the sender.
This procedure only applies to XMPP endpoints.

Example 15: Gateway Reflects Message to XMPP User (F19)

```
|   <message from='montague@chat.example.org/JuliC'
|            to='juliet@example.com/yn0cl4bnw0yr3vym'
|            type='groupchat'
|            id='ix51z73m'>
|        <body>Who knows where Romeo is?</body>
|   </message>
```

5.5.2. Send a Private Message

   Since each occupant has a unique JID, Juliet can send a "private
   message" to a selected occupant through the service by sending a
   message to the user's occupant JID. The XMPP message type ought to
   be "chat" (and is not allowed to be "groupchat").

   The following examples show an exchange of a private message.

   Example 16: Juliet Sends Private Message (F20)

   | <message from='juliet@example.com/yn0cl4bnw0yr3vym'
   | to='montague@chat.example.org/Romeo'
   | type='chat'
   | id='6sfln45q'>
   | <body>O Romeo, Romeo! wherefore art thou Romeo?</body>
   | </message>

   Upon receiving such a message, the XMPP-to-MSRP gateway translates it
   into an MSRP SEND message.

   Example 17: Gateway Maps Private Message from XMPP to MSRP (F21)

   | MSRP a786hjs2 SEND
   | To-Path: msrp://chat.example.org:12763/kjhd37s2s20w2a;tcp
   | From-Path: msrp://x2m.example.com:7654/jshA7weztas;tcp
   | Message-ID: 87652491
   | Byte-Range: 1-*/*
   | Content-Type: message/cpim
   |
   | To: <sip:montague@chat.example.org>;gr=Romeo
   | From: <sip:juliet@example.org>;gr=yn0cl4bnw0yr3vym
   | DateTime: 2008-10-15T15:02:31-03:00
   | Content-Type: text/plain
   |
   | O Romeo, Romeo! wherefore art thou Romeo?
   | -------a786hjs2$

   After acknowledging the message by sending an MSRP 200 OK message
   (step F22, not shown), the MSRP switch is responsible for sending the
   message to the intended recipient. When doing so, it modifies the
   From header to the sender's address within the chat room.

   Example 18: Switch Sends Private Message to SIP User

   | MSRP a786hjs2 SEND
   | To-Path: msrp://chat.example.org:12763/kjhd37s2s20w2a;tcp
   | From-Path: msrp://x2m.example.com:7654/jshA7weztas;tcp
   | Message-ID: 87652491
   | Byte-Range: 1-*/*
   | Content-Type: message/cpim
   |
   | To: <sip:romeo@example.org>
   | From: <sip:montague@chat.example.org>;gr=JuliC
   | DateTime: 2008-10-15T15:02:31-03:00
   | Content-Type: text/plain
   |
   | O Romeo, Romeo! wherefore art thou Romeo?
   | -------a786hjs2$

   Note: If an XMPP-to-MSRP gateway has support for private messaging,
   it MUST advertise that fact by adding a "private-messages" value to
   the a=chatroom SDP attribute it sends to the conference focus, as
   specified in [RFC7701].

   | a=chatroom:nickname private-messages

5.6. Change Nickname

   The XMPP user might want to change her nickname. She can do so by
   sending an updated presence stanza to the room, containing a new
   nickname.

   Example 19: Juliet Changes Her Nickname (F23)

   | <presence from='juliet@example.com/yn0cl4bnw0yr3vym'
   | to='montague@chat.example.org/CapuletGirl'/>

   So far we have assumed that the requested nickname did not conflict
   with any existing nicknames. The following text describes the
   handling of a nickname conflict.

   The MSRP switch analyzes the existing allocation of nicknames, and
   detects that the nickname proposal is already provided to another
   participant. In this case, the MSRP switch answers with a 425
   response.

   Example 20: MSRP Switch Does Not Accept Nickname Proposal (F25)

   | MSRP a786hjs2 425 Nickname usage failed
   | To-Path: msrp://x2m.example.com:7654/jshA7weztas;tcp
   | From-Path: msrp://chat.example.org:12763/kjhd37s2s20w2a;tcp
   | -------a786hjs2

   Upon receiving such a response, the XMPP-to-MSRP gateway translates
   it into an XMPP presence stanza of type "error", specifying a
   error condition (which implies that the XMPP client will
   then need to choose another nickname and repeat the process of
   joining).

   Example 21: Conflict Error for Nickname (F26)

   | <presence from='montague@chat.example.org/JuliC'
   | to='juliet@example.com/yn0cl4bnw0yr3vym'
   | type='error'>
   | <x xmlns='http://jabber.org/protocol/muc'/>
   | <error type='cancel' by='montague@chat.example.org'>
   | <conflict xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
   | </error>
   | </presence>

   Alternatively, the gateway might generate a new nickname request on
   behalf of the XMPP user, thus shielding the XMPP client from handling
   the conflict error.

5.7. Invite Another User to a Room

   In XMPP, there are two methods for inviting another user to a room:
   direct invitations [XEP-0249] (sent directly from the user's real JID
   outside the room to the invitee's real JID) and mediated invitations
   (sent through the room from the user's occupant JID to the invitee's
   JID). In this document, we cover mediated invitations only.

   For example, if Juliet decides to invite Benvolio to the room, she
   sends a message stanza with an invite and Benvolio's JID (which could
   be his real JID or an occupant JID in another room).

   Example 22: Juliet Invites Benvolio to the Room (F27)

   | <message from='juliet@example.com/yn0cl4bnw0yr3vym'
   | id='nzd143v8'
   | to='montague@chat.example.org'>
   | <x xmlns='http://jabber.org/protocol/muc#user'>
   | <invite to='benvolio@example.com'/>
   | </x>
   | </message>

   The XMPP-to-SIP gateway then sends a SIP REFER request to the
   conference focus indicating who needs to be invited in the Refer-To
   header, as per Section 5.5 of [RFC4579].

   Example 23: SIP Mapping of Invite (F28)

   | REFER sip:montague@chat.example.org SIP/2.0
   | To: <sip:montague@chat.example.org>
   | From: "Juliet" <sip:juliet@example.com>;tag=786
   | Call-ID: BC466C1C-E01D-4FD1-B766-9AD174BAF2E7
   | CSeq: 5 REFER
   | Contact: <sip:juliet@example.com>;gr=yn0cl4bnw0yr3vym
   | Accept: message/sipfrag
   | Refer-To: <sip:benvolio@example.com>
   | Supported: replaces
   | Content-Length: 0

   The conference focus then acknowledges the SIP REFER request with a
   200 OK response (step F29, not shown).

   The progress of the invitation will be tracked by the received NOTIFY
   requests as per [RFC3515].

   Example 24: Progress Notification for Invitation (F30)

   | NOTIFY sip:juliet@example.com SIP/2.0
   | To: <sip:juliet@example.com>;tag=786
   | From: <sip:montague@chat.example.org>;tag=087js
   | Call-ID: BC466C1C-E01D-4FD1-B766-9AD174BAF2E7
   | CSeq: 6 NOTIFY
   | Max-Forwards: 70
   | Event: refer
   | Subscription-State: active;expires=60
   | Contact: <sip:montague@chat.example.org;transport=tcp>;isfocus
   | Content-Type: message/sipfrag;version=2.0
   | Content-Length: ...

   Note: Implementers might want to be aware that several recently
   published specifications modify the way in which REFER requests
   handle SIP notifications (see [RFC7647] and [RFC7614]).

5.8. Exit Room

   If Juliet decides to exit the chat room, her client sends a directed
   presence stanza of type "unavailable" to the occupant JID she is
   currently using in the room (here <montague@chat.example.org/JuliC>).

   Example 25: Juliet Exits Room (F31)

   | <presence from='juliet@example.com/yn0cl4bnw0yr3vym'
   | to='montague@chat.example.org/JuliC'
   | type='unavailable'/>

   Upon receiving such a stanza, the XMPP-to-SIP gateway terminates the
   SIP session by sending a SIP BYE to the conference focus and the
   conference focus responds with a SIP 200 OK (steps F32 and F33, not
   shown).

   Juliet can include a custom exit message in the presence stanza of
   type "unavailable", in which case it is broadcast to other
   participants using the methods described above.

   Example 26: Juliet Exits the Chat Room (F31)

   | <presence from='juliet@example.com/yn0cl4bnw0yr3vym'
   | to='montague@chat.example.org/JuliC'
   | type='unavailable'>
   | <status>O, look! methinks I see my cousin's ghost</status>
   | </presence>

6. MSRP Multi-party Messaging Session to XMPP MUC

   This section describes how to map a Multi-party Instant Message (IM)
   MSRP session to an XMPP MUC session. As before, the following
   diagram outlines the overall protocol flow of a sample session, which
   includes some optional exchanges (such as sending messages, changing
   nickname, and inviting another user).

```
    SIP                 SIP               MSRP              XMPP
    User               Proxy             Switch            Server
    |                + S2X GW          + M2X GW           +X2S GW
    |                  |                 |                +X2M GW
    |                  |                 |                  |
    | (F35) SIP        |                 |                  |
    | INVITE           |                 |                  |
    |***************>|                 |                  |
    | (F36) SIP        |                 |                  |
    | 200 OK           |                 |                  |
    |<***************|                 |                  |
    | (F37) SIP ACK    |                 |                  |
    |***************>|                 |                  |
    | (F38) SIP        |                 |                  |
    | SUBSCRIBE        |                 |                  |
    | Event:           |                 |                  |
    | conference       |                 |                  |
    |***************>|                 |                  |
    | (F39) SIP        |                 |                  |
    | 200 OK           |                 |                  |
    |<***************|                 |                  |
    |                  | (F40) XMPP presence: enter room   |
    |                  |.................................>|
    |                  | (F41) XMPP presence               |
    |                  |<.................................|
    | (F42) SIP        |                 |                  |
    | NOTIFY           |                 |                  |
    |<***************|                 |                  |
    | (F43) SIP        |                 |                  |
    | 200 OK           |                 |                  |
    |***************>|                 |                  |
    .                  .                 .                  .
    .                  .                 .                  .
    .                  .                 .                  .
    | (F44) MSRP SEND  |                 |                  |
    |%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%>|                  |
    |                  |                 | (F45) XMPP       |
    |                  |                 | groupchat        |
    |                  |                 | message          |
    |                  |                 |................>|
    |                  |                 | (F46) XMPP       |
    |                  |                 | groupchat        |
    |                  |                 | message          |
    |                  |                 |<................|
    | (F47) MSRP 200 OK|                 |                  |
    |<%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|                  |
    .                  .                 .                  .
```

```
.                .             .                  .
|  (F48) MSRP SEND              |                  |
|%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%>|                  |
|  (F49) MSRP 200 OK            |                  |
|<%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|                  |
|                |              | (F50) XMPP       |
|                |              | message          |
|                |              |................>|
.                .             .                  .

.                .             .                  .
|  (F51) MSRP NICKNAME          |                  |
|%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%>|                  |
|                |              | (F52) XMPP       |
|                |              | presence         |
|                |              |................>|
|                |              | (F53) XMPP       |
|                |              | presence         |
|                |              | error            |
|                |              |<................|
|  (F54) MSRP 425 Error         |                  |
|<%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%|                  |
.                .             .                  .

.                .             .                  .
|  (F55) SIP REFER |            |                  |
|***************>|              |                  |
|  (F56) SIP     |              |                  |
|  200 OK        |              |                  |
|<***************|              |                  |
|  (F57) SIP     |              |                  |
|  NOTIFY        |              |                  |
|<***************|              |                  |
|                | (F58) XMPP message invite       |
|                |................................>|
.                .             .                  .

.                .             .                  .
|  (F59) SIP BYE |              |                  |
|***************>|              |                  |
|                | (F60) XMPP presence unavailable |
|                |................................>|
|                | (F61) XMPP presence unavailable |
|                |<................................|
|  (F62) SIP     |              |                  |
|  200 OK        |              |                  |
|<***************|              |                  |
|                |              |                  |
```

   If the XMPP presence stanza is received before the SIP SUBSCRIBE
   dialog is established for the conference event, then the server
   SHOULD cache the participant list until the subscription is
   established and delivered in a SIP NOTIFY request.

6.1. Enter Room

   When the SIP user ("Romeo") wants to join a groupchat room
   ("capulet"), he first has to start the SIP session by sending out a
   SIP INVITE request containing an offered session description that
   includes an MSRP media line accompanied by mandatory 'path' and
   'chatroom' attributes. Here we assume that Romeo's user agent has
   been configured to be aware of an MSRP switch (chat.example.org) it
   can use. The MSRP media line is also accompanied by an 'accept-
   types' attribute specifying support for a Message/CPIM [RFC3862] top-
   level wrapper for the MSRP message.

   Example 27: SIP User Starts Session (F35)

   | INVITE sip:capulet@rooms.example.com SIP/2.0
   | From: "Romeo" <sip:romeo@example.org>;tag=43524545
   | To: <sip:capulet@rooms.example.com>
   | Contact: <sip:romeo@example.org>;gr=dr4hcr0st3lup4c
   | Call-ID: 08CFDAA4-FAED-4E83-9317-253691908CD2
   | CSeq: 1 INVITE
   | Content-Type: application/sdp
   | Content-Length: ...
   |
   | c=IN IP4 s2x.example.org
   | m=message 7313 TCP/MSRP *
   | a=accept-types:message/cpim text/plain text/html
   | a=accept-wrapped-types:text/plain text/html
   | a=path:msrp://chat.example.org:7313/ansp71weztas;tcp
   | a=chatroom:nickname private-messages

   Upon receiving the INVITE, the SIP proxy needs to determine the
   identity of the domain portion of the Request-URI or To header, which
   it does by following the procedures discussed in [RFC7247]. Here we
   assume that the SIP proxy has determined that the domain is serviced
   by an XMPP server, that it contains or has available to it a SIP-to-
   XMPP gateway or connection manager (which enables it to speak
   natively to XMPP servers), and that it hands off the message to the
   gateway.

   Implementations MAY wait until the nickname is set with an MSRP
   NICKNAME chunk before joining the XMPP MUC or MAY choose a temporary
   nickname (such as the SIP From header display name) and use it to
   join the room. Here we assume the latter.

   Example 28: SIP-to-XMPP Gateway ACKs Session (F36)

   | SIP/2.0 200 OK
   | From: "Romeo" <sip:romeo@example.org>;tag=43524545
   | To: <sip:capulet@rooms.example.com>;tag=a3343df32
   | Contact: <sip:rooms.example.com;transport=tcp>;isfocus
   | Call-ID: 08CFDAA4-FAED-4E83-9317-253691908CD2
   | CSeq: 1 INVITE
   | Content-Type: application/sdp
   |
   | m=message 8763 TCP/MSRP *
   | a=accept-types:message/cpim text/plain text/html
   | a=accept-wrapped-types:text/plain text/html
   | a=path:msrp://chat.example.org:8763/lkjh37s2s20w2a;tcp
   | a=chatroom:nickname private-messages

   The SIP/MSRP user agent subscribes to a conference event package at
   the destination groupchat service.

   Example 29: Gateway Subscribes to the Conference (F38)

   | SUBSCRIBE sip:capulet@rooms.example.com SIP/2.0
   | To: <sip:capulet@rooms.example.com>;tag=a3343df32
   | From: "Romeo" <sip:romeo@example.org>;tag=43524545
   | Contact: <sip:romeo@example.org>;gr=dr4hcr0st3lup4c
   | Call-ID: 08CFDAA4-FAED-4E83-9317-253691908CD2
   | CSeq: 2 SUBSCRIBE
   | Event: conference
   | Expires: 600
   | Accept: application/conference-info+xml
   | Allow-Events: conference
   | Content-Length: 0

   After the conference subscription request is acknowledged, the SIP-
   to-XMPP gateway sends presence from Romeo to the MUC chat room.

   Example 30: Romeo Enters XMPP Chat Room (F40)

   | <presence from='romeo@example.org/dr4hcr0st3lup4c'
   | to='montague@chat.example.org/Romeo'>
   | <x xmlns='http://jabber.org/protocol/muc'/>
   | </presence>

6.2. Presence Broadcast

   If the MUC service is able to add the SIP/MSRP user to the room, it
   sends presence from all the existing occupants' room JIDs to the new
   occupant's full JID, including extended presence information about
   roles in an <x/> element.

   Example 31: XMPP Service Sends Presence from Existing Occupants to
   New Occupant (F41)

   | <presence from='capulet@rooms.example.com/Romeo'
   | to='romeo@example.org/dr4hcr0st3lup4c'>
   | <x xmlns='http://jabber.org/protocol/muc#user'>
   | <item affiliation='none' role='participant'/>
   | <status code='110'/>
   | </x>
   | </presence>
   |
   | <presence from='capulet@rooms.example.com/Ben'
   | to='romeo@example.org/dr4hcr0st3lup4c'>
   | <x xmlns='http://jabber.org/protocol/muc#user'>
   | <item affiliation='none' role='participant'/>
   | </x>
   | </presence>
   |
   | <presence from='capulet@rooms.example.com/JuliC'
   | to='romeo@example.org/dr4hcr0st3lup4c'>
   | <x xmlns='http://jabber.org/protocol/muc#user'>
   | <item affiliation='none' role='participant'/>
   | </x>
   | </presence>

   Upon receiving these presence stanzas, if the conference focus has
   already completed the subscription to the conference event package
   [RFC4575], the XMPP-to-SIP gateway translates them into a SIP NOTIFY
   request containing the participant list (represented in the
   conference-info format specified in [RFC4575]).

   Example 32: SIP Mapping of XMPP Participant Presence Stanzas (F42)

   | NOTIFY sip:romeo@example.org SIP/2.0
   | To: <sip:romeo@example.org>;tag=43524545
   | From: <sip:capulet@rooms.example.com>;tag=a3343df32
   | Call-ID: 08CFDAA4-FAED-4E83-9317-253691908CD2
   | CSeq: 3 NOTIFY
   | Event: conference
   | Subscription-State: active;expires=3600
   | Content-Type: application/conference-info+xml
   | Content-Length: ...
   |
   | <conference-info version="0" state="full"
   | entity="sip:capulet@rooms.example.com">
   | <conference-description>
   | <subject>Today in Verona</subject>
   | <conf-uris>
   | <entry>
   | <uri>tel:+18882934234</uri>
   | <uri>sip:capulet@rooms.example.com</uri>
   | </entry>
   | </conf-uris>
   | </conference-description>
   | <users>
   | <user entity="sip:capulet@rooms.example.com;gr=JuliC"
   | state="full">
   | <display-text>JuliC</display-text>
   | <roles>
   | <entry>participant</entry>
   | </roles>
   | <endpoint entity="sip:capulet@rooms.example.com;gr=JuliC"
   | state="full">
   | <status>connected</status>
   | <media id="211835821">
   | <type>message</type>
   | </media>
   | </endpoint>
   | </user>
   | <user entity="sip:capulet@rooms.example.com;gr=Ben"
   | state="full">
   | <display-text>Ben</display-text>
   | <roles>
   | <entry>participant</entry>
   | </roles>
   | <endpoint entity="sip:capulet@rooms.example.com;gr=Ben"
   | state="full">
   | <status>connected</status>
   | <media id="211835822">

```
|           <type>message</type>
|         </media>
|       </endpoint>
|     </user>
|   </users>
| </conference-info>
```

Because the "room roster" is communicated in XMPP by means of
multiple presence stanzas (one for each participant) whereas the
participant list is communicated in SIP by means of a single
conference information document, the SIP-to-XMPP gateway will need to
keep track of the user's SIP URI and the mapping of that URI into an
XMPP address; then, based on that mapping, it will need to determine
when it has received a complete room roster from the MUC room, i.e.,
when it has received the in-room presence of the SIP user (which
according to [XEP-0045] is the last presence stanza received in the
initial batch sent after joining). Once that happens, the SIP-to-
XMPP gateway can construct the conference information document
containing the complete participant list and send that to the SIP
user.

6.3. Exchange Messages

Once the user has joined the chat room, the user can exchange an
unbounded number of messages, both public and private.

The mapping of MSRP syntax elements to XMPP syntax elements MUST be
as shown in the following table. (Mappings for elements not
mentioned are undefined.)

Table 5: Message Syntax Mapping from MSRP Message to XMPP

CPIM Header	XMPP Element or Attribute
To	to
From	from
body of the SEND request	<body/>

6.3.1. Send a Message to All Occupants

When Romeo wants to send a message to all other occupants in the
room, he sends an MSRP SEND request to <room@service>
(<capulet@rooms.example.com> in our example).

The following examples show an exchange of a public message.

Example 33: Romeo Sends a Message to the Chat Room (F44)

```
|   MSRP a786hjs2 SEND
|   To-Path: msrp://room.example.com:7313/ansp71weztas;tcp
|   From-Path: msrp://chat.example.org:8763/lkjh37s2s20w2a;tcp
|   Message-ID: 87652492
|   Byte-Range: 1-*/*
|   Content-Type: message/cpim
|
|   To: <sip:capulet@rooms.example.com>
|   From: "Romeo" <sip:romeo@example.org>;gr=dr4hcr0st3lup4c
|   DateTime: 2008-10-15T15:02:31-03:00
|   Content-Type: text/plain
|
|   Romeo is here!
|   -------a786hjs2$
```

Upon receiving the SEND request, if the request either contains a
Failure-Report header field value of "yes" or does not contain a
Failure-Report header at all, the SIP-to-XMPP gateway immediately
translates it into an XMPP message stanza and then generates and
sends an MSRP response.

Example 34: XMPP Mapping of Message (F45)

```
|   <message from='romeo@example.org/dr4hcr0st3lup4c'
|            to='capulet@rooms.example.com'
|            type='groupchat'
|            id='8gbx1g4p'>
|     <body>Romeo is here!</body>
|   </message>
```

Example 35: MSRP Response to Public Message (F47)

```
|   MSRP d93kswow 200 OK
|   To-Path: msrp://rooms.example.com:8763/lkjh37s2s20w2a;tcp
|   From-Path: msrp://chat.example.org:7313/ansp71weztas;tcp
|   -------d93kswow$
```

Note well that the XMPP MUC room will reflect the sender's message
back to all users, including the sender. The MSRP-to-XMPP gateway
SHOULD wait until receiving this reflected message before sending an
MSRP 200 OK reply to the original sender.

6.3.2. Send a Private Message

   Romeo can send a "private message" to a selected occupant via the
   chat room service by sending a message to the occupant's room
   nickname.

   The following examples show an exchange of a private message.

   Example 36: Romeo Sends a Private Message (F48)

   | MSRP a786hjs2 SEND
   | To-Path: msrp://rooms.example.com:7313/ansp71weztas;tcp
   | From-Path: msrp://chat.example.org:8763/lkjh37s2s20w2a;tcp
   | Message-ID: 87652492
   | Byte-Range: 1-*/*
   | Content-Type: message/cpim
   |
   | To: <sip:capulet@rooms.example.com>;gr=JuliC
   | From: "Romeo" <sip:romeo@example.org>;gr=dr4hcr0st3lup4c
   | DateTime: 2008-10-15T15:02:31-03:00
   | Content-Type: text/plain
   |
   | I am here!!!
   | -------a786hjs2$

   The MSRP switch is responsible for transforming the 'From' address
   into an in-room address (not shown).

   Once the MSRP switch sends that message to the gateway, the gateway
   is responsible for translating it into XMPP syntax.

   Example 37: XMPP Mapping of Private Message (F50)

   | <message from='capulet@rooms.example.com/Romeo'
   | to='capulet@rooms.example.com/JuliC'
   | type='chat'
   | id='rg2ca9k7'>
   | <body>I am here!!!</body>
   | </message>

6.4. Change Nickname

   If Romeo decides to change his nickname within the room, he sends a
   new MSRP NICKNAME request. In fact, modification of the nickname in
   MSRP is not different from the initial reservation and usage of a
   nickname.

Example 38: MSRP User Changes Nickname (F51)

```
| MSRP a786hjs2 NICKNAME
| To-Path: msrp://chat.example.org:7313/ansp71weztas;tcp
| From-Path: msrp://rooms.example.com:8763/lkjh37s2s20w2a;tcp
| Use-Nickname: "montecchi"
| -------a786hjs2
```

Upon receiving such a message, the MSRP-to-XMPP gateway translates it
into an XMPP presence stanza.

Example 39: XMPP Mapping of Nickname Change (F52)

```
| <presence from='romeo@example.org/dr4hcr0st3lup4c'
|           to='capulet@rooms.example.com/montecchi'/>
```

The XMPP server will analyze the nickname allocation and determine if
the requested nickname is available. In case the nickname is not
available or not usable, the server will generate a presence stanza
of type "error" specifying a <conflict/> error condition.

Example 40: XMPP Conflict Error for Nickname (F53)

```
| <presence from='capulet@rooms.example.com/Romeo'
|           to='romeo@example.org/dr4hcr0st3lup4c'
|           type='error'>
|   <x xmlns='http://jabber.org/protocol/muc'/>
|   <error type='cancel' by='capulet@rooms.example.com'>
|     <conflict xmlns='urn:ietf:params:xml:ns:xmpp-stanzas'/>
|   </error>
| </presence>
```

Upon receiving this stanza, the XMPP-to-MSRP gateway will reply to
the NICKNAME request with code 425.

Example 41: Gateway Translates XMPP Nickname Conflict to MSRP Error
Code (F54)

```
| MSRP a786hjs2 425 Nickname usage failed
| To-Path: msrp://chat.example.org:7313/ansp71weztas;tcp
| From-Path: msrp://rooms.example.com:8763/lkjh37s2s20w2a;tcp
| -------a786hjs2
```

6.5. Invite Another User to a Room

If a SIP user wants to invite another user to join the conference he
will send a REFER request indicating who needs to be invited in the
Refer-To header, as per Section 5.5 of [RFC4579].

Example 42: SIP User Invites Another User (F55)

```
|  REFER sip:capulet@rooms.example.com SIP/2.0
|  To: <sip:capulet@rooms.example.com>;tag=a3343df32
|  From: "Romeo" <sip:romeo@example.org>;tag=5534562
|  Call-ID: 08CFDAA4-FAED-4E83-9317-253691908CD2
|  CSeq: 4 REFER
|  Contact: <sip:romeo@example.org>;gr=dr4hcr0st3lup4c
|  Accept: message/sipfrag
|  Refer-To: <sip:benvolio@example.com>
|  Supported: replaces
|  Content-Length: 0
```

The SIP-to-XMPP gateway then acknowledges the SIP REFER request with
a 200 OK response (step F56).

The gateway will then send a NOTIFY request as per [RFC3515]
indicating that the invitation is in progress. Since there is no way
to know the progress of the invitation until the user has joined,
implementations are advised to terminate the REFER dialog
subscription upon receiving the first NOTIFY request, with a status
code of 100 Trying.

Example 43: Progress Notification for Invitation (F56)

```
|  NOTIFY sip:romeo@example.org SIP/2.0
|  To: <sip:romeo@example.org>;tag=5534562
|  From: <sip:capulet@rooms.example.com>;tag=a3343df32
|  Call-ID: 08CFDAA4-FAED-4E83-9317-253691908CD2
|  CSeq: 5 NOTIFY
|  Event: refer
|  Subscription-State: terminated;reason=noresource
|  Contact: <sip:capulet@rooms.example.com;transport=tcp>;isfocus
|  Content-Type: message/sipfrag;version=2.0
|  Content-Length: ...
|
|  SIP/2.0 100 Trying
```

6.6. Exit Room

If Romeo decides to exit the chat room, his client sends a SIP BYE to
the <montague@chat.example.org> chat room.

Example 44: Romeo Terminates Session (F59)

```
|   BYE sip:capulet@rooms.example.com SIP/2.0
|   Max-Forwards: 70
|   From: "Romeo" <sip:romeo@example.org>;tag=5534562
|   To: <sip:capulet@rooms.example.com>;tag=a3343df32
|   Call-ID: 08CFDAA4-FAED-4E83-9317-253691908CD2
|   CSeq: 6 BYE
|   Content-Length: 0
```

Upon receiving the SIP BYE, the SIP-to-XMPP gateway translates it
into a presence stanza of type "unavailable" (F60) and sends it to
the XMPP MUC room service. Then, the SIP-to-XMPP gateway responds
with a 200 OK to the MSRP user (F62).

Example 45: Romeo Exits Chat Room (F60)

```
|   <presence from='romeo@example.org/dr4hcr0st3lup4c'
|             to='capulet@rooms.example.com/Romeo'
|             type='unavailable'/>
```

7. Handling of Nicknames and Display Names

Fundamental rules for mapping addresses between XMPP and SIP are
provided in [RFC7247]. However, chat rooms include a more
specialized, unique identifier for each participant in a room, called
a "nickname". Implementations SHOULD apply the rules for preparation
and comparison of nicknames specified in [RFC7700].

In addition to nicknames, some groupchat implementations also include
display names (which might or might not be different from users'
nicknames). A display name need not be unique within the context of
a room but instead simply provides a user-friendly name for a
participant.

In the SIP conference event package, the nickname is the value of the
Centralized Conferencing (XCON) 'nickname' attribute of the
element [RFC6501] and the display name is the XML character data of
the conference-info element [RFC4575]. In XMPP, the
nickname is the value of the resourcepart of the occupant JID
[XEP-0045] and the display name is the XML character data of the
element [XEP-0172].

In practice, the element is treated as canonical in
SIP implementations, and the element is rarely used in XMPP
implementations. Therefore, for display purposes, SIP
implementations ought to use the element if the XCON

'nickname' attribute is not present, and XMPP implementations ought
to use the resourcepart of the occupant JID if the <nick/> element is
not present.

If there is a conflict between the SIP nickname and the XMPP
nickname, the SIP-to-XMPP or XMPP-to-SIP gateway is responsible for
adjusting the nickname to avoid the conflict and for informing the
SIP or XMPP client of the unique nickname used to join the chat room.

8. Message Size

It is possible for MSRP messages to exceed the size allowed by an
XMPP service on the far end of an MSRP-to-XMPP gateway; see [RFC7573]
for a discussion of this issue.

9. Security Considerations

The security considerations of [RFC3261], [RFC4975], [RFC6120],
[RFC7247], [RFC7701], and [XEP-0045] apply.

This document specifies methods for exchanging groupchat messages
through a gateway that translates between SIP and XMPP. Such a
gateway MUST be compliant with the minimum security requirements of
the protocols for which it translates (i.e., MSRP/SIP and XMPP). The
addition of gateways to the security models of MSRP, SIP, and XMPP
introduces some new risks. In particular, end-to-end security
properties (especially confidentiality and integrity) between user
agents that interface through an MSRP-to-XMPP gateway can be provided
only if common formats are supported; unfortunately, although
[RFC3862] specifies such a format for one-to-one instant messages,
the problem of end-to-end security for multi-party messaging has not
been solved in a standardized way.

Some of the features that are not addressed by the minimal
interoperability baseline defined in this document are relevant to
security, such as the ability to administer rooms, kick out and ban
users, and enable room moderation. Users needing to take advantage
of such features cannot do so through a gateway in a standardized
manner and therefore will need to use native clients for the relevant
protocol (MSRP or XMPP).

As mentioned in [RFC7572], there are several possible methods for
end-to-end encryption of one-to-one instant messages. Unfortunately,
because there is no widely deployed method for end-to-end encryption
of multi-party instant messages, this document cannot provide a
recommendation in this regard.

10. References

10.1. 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>.

   [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>.

   [RFC4579] Johnston, A. and O. Levin, "Session Initiation Protocol
              (SIP) Call Control - Conferencing for User Agents",
              BCP 119, RFC 4579, DOI 10.17487/RFC4579, August 2006,
              <http://www.rfc-editor.org/info/rfc4579>.

   [RFC4975] Campbell, B., Ed., Mahy, R., Ed., and C. Jennings, Ed.,
              "The Message Session Relay Protocol (MSRP)", RFC 4975,
              DOI 10.17487/RFC4975, September 2007,
              <http://www.rfc-editor.org/info/rfc4975>.

   [RFC5627] Rosenberg, J., "Obtaining and Using Globally Routable User
              Agent URIs (GRUUs) in the Session Initiation Protocol
              (SIP)", RFC 5627, DOI 10.17487/RFC5627, October 2009,
              <http://www.rfc-editor.org/info/rfc5627>.

   [RFC6120] Saint-Andre, P., "Extensible Messaging and Presence
              Protocol (XMPP): Core", RFC 6120, DOI 10.17487/RFC6120,
              March 2011, <http://www.rfc-editor.org/info/rfc6120>.

   [RFC6121] Saint-Andre, P., "Extensible Messaging and Presence
              Protocol (XMPP): Instant Messaging and Presence",
              RFC 6121, DOI 10.17487/RFC6121, March 2011,
              <http://www.rfc-editor.org/info/rfc6121>.

   [RFC7247] Saint-Andre, P., Houri, A., and J. Hildebrand,
              "Interworking between the Session Initiation Protocol
              (SIP) and the Extensible Messaging and Presence Protocol
              (XMPP): Architecture, Addresses, and Error Handling",
              RFC 7247, DOI 10.17487/RFC7247, May 2014,
              <http://www.rfc-editor.org/info/rfc7247>.

   [RFC7573] Saint-Andre, P. and S. Loreto, "Interworking between the
              Session Initiation Protocol (SIP) and the Extensible
              Messaging and Presence Protocol (XMPP): One-to-One Text
              Chat Sessions", RFC 7573, DOI 10.17487/RFC7573, June 2015,
              <http://www.rfc-editor.org/info/rfc7573>.

   [RFC7700] Saint-Andre, P., "Preparation, Enforcement, and Comparison
              of Internationalized Strings Representing Nicknames",
              RFC 7700, DOI 10.17487/RFC7700, December 2015,
              <http://www.rfc-editor.org/info/rfc7700>.

   [RFC7701] Niemi, A., Garcia-Martin, M., and G. Sandbakken, "Multi-
              party Chat Using the Message Session Relay Protocol
              (MSRP)", RFC 7701, DOI 10.17487/RFC7701, December 2015,
              <http://www.rfc-editor.org/info/rfc7701>.

   [XEP-0045] Saint-Andre, P., "Multi-User Chat", XSF XEP 0045, February
              2012, <http://www.xmpp.org/extensions/xep-0045.html>.

10.2. Informative References

   [RFC3515] Sparks, R., "The Session Initiation Protocol (SIP) Refer
              Method", RFC 3515, DOI 10.17487/RFC3515, April 2003,
              <http://www.rfc-editor.org/info/rfc3515>.

   [RFC3862] Klyne, G. and D. Atkins, "Common Presence and Instant
              Messaging (CPIM): Message Format", RFC 3862,
              DOI 10.17487/RFC3862, August 2004,
              <http://www.rfc-editor.org/info/rfc3862>.

   [RFC4353] Rosenberg, J., "A Framework for Conferencing with the
              Session Initiation Protocol (SIP)", RFC 4353,
              DOI 10.17487/RFC4353, February 2006,
              <http://www.rfc-editor.org/info/rfc4353>.

   [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>.

   [RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, Ed., "A
              Session Initiation Protocol (SIP) Event Package for
              Conference State", RFC 4575, DOI 10.17487/RFC4575, August
              2006, <http://www.rfc-editor.org/info/rfc4575>.

   [RFC6501] Novo, O., Camarillo, G., Morgan, D., and J. Urpalainen,
              "Conference Information Data Model for Centralized
              Conferencing (XCON)", RFC 6501, DOI 10.17487/RFC6501,
              March 2012, <http://www.rfc-editor.org/info/rfc6501>.

   [RFC6502] Camarillo, G., Srinivasan, S., Even, R., and J.
              Urpalainen, "Conference Event Package Data Format
              Extension for Centralized Conferencing (XCON)", RFC 6502,
              DOI 10.17487/RFC6502, March 2012,
              <http://www.rfc-editor.org/info/rfc6502>.

   [RFC7572] Saint-Andre, P., Houri, A., and J. Hildebrand,
              "Interworking between the Session Initiation Protocol
              (SIP) and the Extensible Messaging and Presence Protocol
              (XMPP): Instant Messaging", RFC 7572,
              DOI 10.17487/RFC7572, June 2015,
              <http://www.rfc-editor.org/info/rfc7572>.

   [RFC7614] Sparks, R., "Explicit Subscriptions for the REFER Method",
              RFC 7614, DOI 10.17487/RFC7614, August 2015,
              <http://www.rfc-editor.org/info/rfc7614>.

   [RFC7647] Sparks, R. and A. Roach, "Clarifications for the Use of
              REFER with RFC 6665", RFC 7647, DOI 10.17487/RFC7647,
              September 2015, <http://www.rfc-editor.org/info/rfc7647>.

   [XEP-0172] Saint-Andre, P. and V. Mercier, "User Nickname", XSF
              XEP 0172, March 2012,
              <http://xmpp.org/extensions/xep-0172.html>.

   [XEP-0249] Saint-Andre, P., "Direct MUC Invitations", XSF XEP 0249,
              September 2011,
              <http://xmpp.org/extensions/xep-0249.html>.

Authors' Addresses

    Peter Saint-Andre
    &yet

    Email: peter@andyet.com
    URI: https://andyet.com/

    Saul Ibarra Corretge
    AG Projects
    Dr. Leijdsstraat 92
    Haarlem 2021RK
    The Netherlands

    Email: saul@ag-projects.com

    Salvatore Loreto
    Ericsson
    Hirsalantie 11
    Jorvas 02420
    Finland

    Email: Salvatore.Loreto@ericsson.com