Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 27, 2014

P. Saint-Andre

S. Loreto
Ericsson

January 23, 2014

Interworking between the Session Initiation Protocol (SIP) and the Extensible Messaging and Presence Protocol (XMPP): One-to-One Text Chat Sessions

draft-ietf-stox-chat-05

Abstract

This document defines a bidirectional protocol mapping for the exchange of instant messages in the context of a one-to-one chat session between a user of the Session Initiation Protocol (SIP) and a user of the Extensible Messaging and Presence Protocol (XMPP). Specifically for SIP text chat, this document specifies a mapping to the Message Session Relay Protocol (MSRP).

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 July 27, 2014.

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 one-to-one 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 [I-D.ietf-stox-core], including mapping of addresses and error conditions. This document specifies mappings for one-to-one text chat sessions (sometimes called "session-mode" messaging); in particular, this document specifies mappings between XMPP messages of type "chat" and the Message Session Relay Protocol (MSRP) [RFC4975], which is commonly used in SIP-based systems for chat functionality (although note that MSRP is not conjoined to SIP, and can be used by non-SIP technologies). Mappings for single instant messages and groupchat are provided in separate documents.

The approach taken here is to directly map syntax and semantics from one protocol to another. The mapping described herein depends on the protocols defined in the following specifications:

- XMPP chat sessions using message stanzas of type "chat" are specified in [RFC6121].

- MSRP chat sessions using the SIP INVITE and SEND request types are specified in [RFC4975].

In SIP-based systems that use MSRP, a chat session is formally negotiated just as any other session type is using SIP. By contrast, a one-to-one chat "session" in XMPP is an informal construct and is not formally negotiated: a user simply sends a message of type "chat" to a contact, the contact then replies to the message, and the sum total of such messages exchanged during a defined period of time is considered to be a chat session (ideally tied together using an XMPP <thread/> element as described in Section 5.1 of [RFC6121]). To overcome the disparity between these approaches, a gateway that wishes to map between SIP/MSRP and XMPP for one-to-one chat sessions needs to maintain some additional state, as described below.

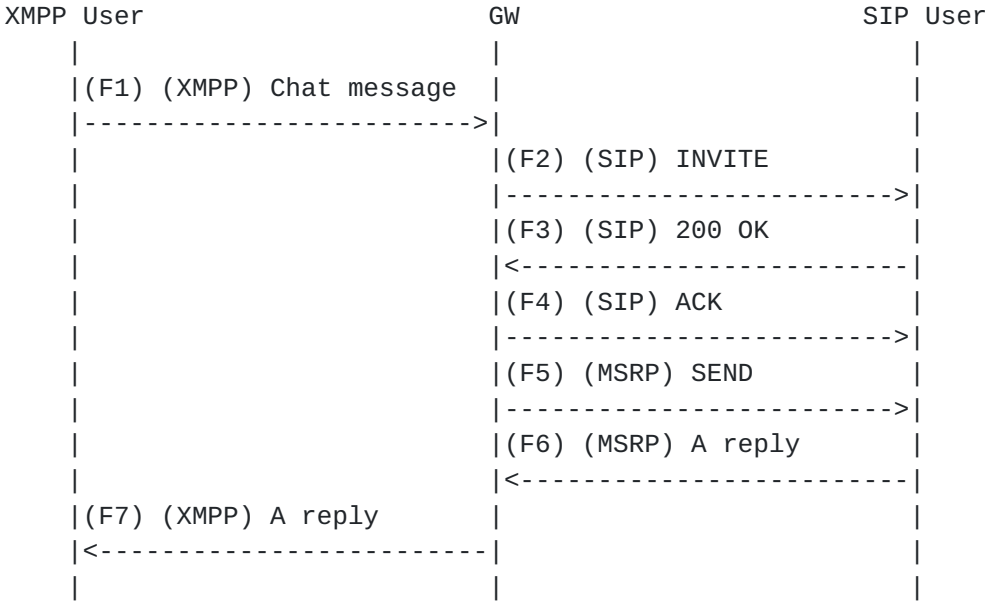
2. Terminology

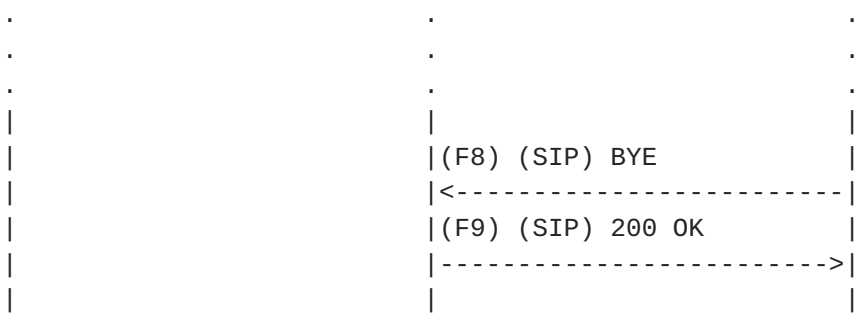
The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. XMPP to MSRP

In XMPP, the "informal session" approach is to simply send someone a <message/> of type "chat" without starting any session negotiation ahead of time (as described in [RFC6121]). The XMPP "informal session" approach maps very well into a SIP MESSAGE request, as described in [I-D.ietf-stox-core]. However, the XMPP informal session approach can also be mapped to MSRP if the XMPP-to-SIP gateway maintains additional state.

The order of events is as follows.





The mapping of XMPP syntax to SIP syntax SHOULD be as shown in the following table. (Mappings for several aspects not mentioned here are specified in [I-D.ietf-stox-im].)

Table 1: Message syntax mapping from XMPP to SIP

XMPP Element or Attribute	SIP Header or Contents
<thread/>	Call-ID
id	transaction identifier

First the XMPP user would generate an XMPP chat message.

Example 1: Juliet sends XMPP message (F1)

```
| <message from='juliet@example.com/balcony'
|          to='romeo@example.net'
|          id='a786hjs2'
|          type='chat'>
|   <thread>29377446-0CBB-4296-8958-590D79094C50</thread>
|   <body>Art thou not Romeo, and a Montague?</body>
| </message>
```

Upon receiving such a message stanza, the XMPP server needs to determine the identity of the domainpart in the 'to' address, which it does by following the procedures explained in Section 5 of [I-D.ietf-stox-core]. Here we assume that the XMPP server has determined the domain is serviced by an MSRP server, that it contains or has available to it an XMPP-to-SIP gateway or connection manager (which enables it to speak natively to MSRP servers), and that it hands off the message stanza to the XMPP-SIP gateway.

The XMPP-to-SIP gateway at the XMPP server would then initiate an MSRP session with Romeo on Juliet's behalf (since there is no reliable way for the gateway to determine if Romeo's client supports MSRP, it simply needs to guess).

Example 2: Gateway starts SIP session on behalf of Juliet (F2)

```
| INVITE sip:romeo@example.net SIP/2.0
| To: <sip:romeo@example.net>
| From: <sip:juliet@example.com>
| Contact: <sip:juliet@example.com>;gr=balcony
| Subject: Open chat with Juliet?
| Call-ID: 29377446-0CBB-4296-8958-590D79094C50
| Content-Type: application/sdp
|
| c=IN IP4 x2s.example.com
| m=message 7654 TCP/MSRP *
| a=accept-types:text/plain
| a=path:msrp://x2s.example.com:7654/jshA7weztas;tcp
```

Here we assume that Romeo accepts the MSRP session request.

Example 3: Romeo accepts session request (F3)

```
| SIP/2.0 200 OK
| To: <sip:juliet@example.com>
| From: <sip:romeo@example.net>
| Contact: <sip:romeo@example.net>;gr=orchard
| Call-ID: 29377446-0CBB-4296-8958-590D79094C50
| Content-Type: application/sdp
|
| c=IN IP4 s2x.example.net
| m=message 12763 TCP/MSRP *
| a=accept-types:text/plain
| a=path:msrp://s2x.example.net:12763/kjhd37s2s20w2a;tcp
```

The XMPP-to-SIP gateway then acknowledges the session acceptance on behalf of Juliet.

Example 4: Gateway sends ACK to Romeo (F4)

```
| ACK sip:juliet@example.com SIP/2.0
| To: <sip:romeo@example.net>;gr=orchard
| From: <sip:juliet@example.com>
| Contact: <sip:juliet@example.com>;gr=balcony
| Call-ID: 29377446-0CBB-4296-8958-590D79094C50
```

The XMPP-to-SIP gateway then transforms the original XMPP chat message into MSRP.

Example 5: Gateway maps XMPP message to MSRP (F5)

```
| MSRP a786hjs2 SEND
| From-Path: msrp://x2s.example.com:7654/jshA7weztas;tcp
| To-Path: msrp://s2x.example.net:12763/kjhd37s2s20w2a;tcp
| Message-ID: 54C6F4F1-A39C-47D6-8718-FA65B3D0414A
| Byte-Range: 1-25/25
| Content-Type: text/plain
|
| Art thou not Romeo, and a Montague?
| -------a786hjs2$
```

Romeo can then send a reply using his MSRP client.

Example 6: Romeo sends reply (F6)

```
| MSRP di2fs53v SEND
| To-Path: msrp://x2s.example.com:7654/jshA7weztas;tcp
| From-Path: msrp://s2x.example.net:12763/kjhd37s2s20w2a;tcp
| Message-ID: 6480C096-937A-46E7-BF9D-1353706B60AA
| Byte-Range: 1-25/25
| Failure-Report: no
| Content-Type: text/plain
|
| Neither, fair saint, if either thee dislike.
| -------di2fs53v$
```

The SIP-to-XMPP gateway would then transform that message into appropriate XMPP syntax for routing to the intended recipient.

Example 7: Gateway maps MSRP message to XMPP (F7)

```
| <message from='romeo@example.net/orchard'
|          to='juliet@example.com/balcony'
|          id='di2fs53v'
|          type='chat'>
|   <thread>29377446-0CBB-4296-8958-590D79094C50</thread>
|   <body>Neither, fair saint, if either thee dislike.</body>
| </message>
```

When the MSRP user wishes to end the chat session, the user's MSRP client sends a SIP BYE.

Example 8: Romeo terminates chat session (F8)

```
| BYE juliet@example.com sip: SIP/2.0
| From: <sip:romeo@example.net>;tag=087js
| To: <sip:juliet@example.com>;tag=786
| Call-ID: 29377446-0CBB-4296-8958-590D79094C50
| Cseq: 1 BYE
| Content-Length: 0
```

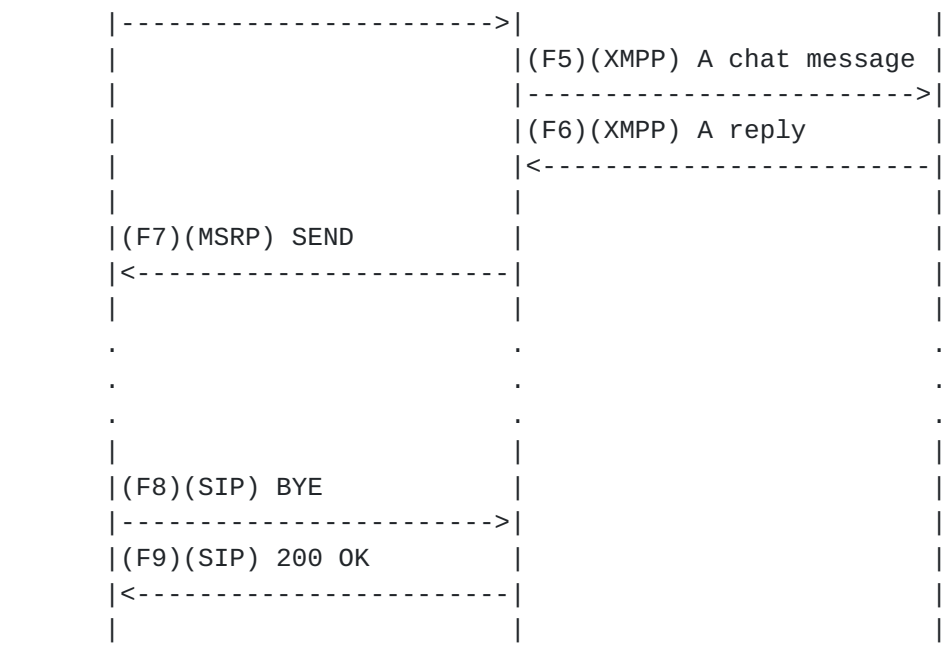
The BYE is then acknowledged by the XMPP-to-SIP gateway.

Example 9: Gateway acknowledges termination (F9)

```
| SIP/2.0 200 OK
| From: <sip:juliet@example.com>;tag=786
| To: <sip:romeo@example.net>;tag=087js
| Call-ID: 29377446-0CBB-4296-8958-590D79094C50
| CSeq: 1 BYE
| Content-Length: 0
```

4. MSRP to XMPP

When an MSRP client sends messages through a gateway to an XMPP client, the order of events is as follows.



The mapping of SIP syntax to XMPP syntax SHOULD be as shown in the following table. (Mappings for several aspects not mentioned here are specified in [I-D.ietf-stox-im].)

Table 2: Message syntax mapping from SIP to XMPP

SIP Header or Contents	XMPP Element or Attribute
Call-ID transaction identifier	<thread/> id

The protocol flow begins when Romeo starts a chat session with Juliet.

Example 10: Romeo starts chat session (F1)

```
| INVITE sip:juliet@example.com SIP/2.0
| To: <sip:juliet@example.com>
| From: <sip:romeo@example.net>
| Contact: <sip:romeo@example.net>;gr=orchard
| Subject: Open chat with Romeo?
| Call-ID: F6989A8C-DE8A-4E21-8E07-F0898304796F
| Content-Type: application/sdp
|
| c=IN IP4 s2x.example.net
| m=message 7313 TCP/MSRP *
| a=accept-types:text/plain
| a=path:msrp://s2x.example.net:7313/ansp71weztas;tcp
```

Upon receiving the INVITE, the SIP (MSRP) server needs to determine the identity of the domain portion of the Request-URI or To header, which it does by following the procedures explained in Section 5 of [I-D.ietf-stox-core]. Here we assume that the SIP server 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.

Example 11: Gateway accepts session on Juliet's behalf (F2)

```
| SIP/2.0 200 OK
| To: <sip:romeo@example.net>;gr=orchard
| From: <sip:juliet@example.com>
| Contact: <sip:juliet@example.com>;gr=balcony
| Call-ID: F6989A8C-DE8A-4E21-8E07-F0898304796F
| Content-Type: application/sdp
|
| c=IN IP4 x2s.example.com
| m=message 8763 TCP/MSRP *
| a=accept-types:text/plain
| a=path:msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
```

Example 12: Romeo sends ACK (F3)

```
| ACK sip:juliet@example.com SIP/2.0
| To: <sip:juliet@example.com>;gr=balcony
| From: <sip:romeo@example.net>
| Contact: <sip:romeo@example.net>;gr=orchard
| Call-ID: F6989A8C-DE8A-4E21-8E07-F0898304796F
```

Example 13: Romeo sends message (F4)

```
| MSRP ad49kswow SEND
| To-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
| From-Path: msrp://s2x.example.net:7313/ansp71weztas;tcp
| Message-ID: 676FDB92-7852-443A-8005-2A1B9FE44F4E
| Byte-Range: 1-32/32
| Failure-Report: no
| Content-Type: text/plain
|
| I take thee at thy word ...
| -------ad49kswow$
```

Example 14: SIP-XMPP gateway maps MSRP message to XMPP (F5)

```
| <message from='romeo@example.net'
|          to='juliet@example.com'
|          id='ad49kswow'
|          type='chat'>
|   <thread>F6989A8C-DE8A-4E21-8E07-F0898304796F</thread>
|   <body>I take thee at thy word ...</body>
| </message>
```

Example 15: Juliet sends reply (F6)

```
| <message from='juliet@example.com'
|          to='romeo@example.net'
|          id='ms53b7z9'
|          type='chat'>
|   <thread>29377446-0CBB-4296-8958-590D79094C50</thread>
|   <body>What man art thou ...?</body>
| </message>
```

Example 16: Gateway maps XMPP message to MSRP (F7)

```
| MSRP ms53b7z9 SEND
| To-Path: msrp://s2x.example.net:7313/jshA7weztas;tcp
| From-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
| Message-ID: 17EBA17B-94C0-463B-AD84-DE405C4C9D41
| Byte-Range: 1-25/25
| Failure-Report: no
| Content-Type: text/plain
|
| What man art thou ...?
| -------ms53b7z9$
```

Example 17: Romeo terminates chat session (F8)

```
| BYE juliet@example.com sip: SIP/2.0
| To: <sip:juliet@example.com>;gr=balcony
| From: <sip:romeo@example.net>
| Contact: <sip:romeo@example.net>;gr=orchard
| Call-ID: F6989A8C-DE8A-4E21-8E07-F0898304796F
| Cseq: 1 BYE
| Content-Length: 0
```

Example 18: Gateway acknowledges termination of session on behalf of Juliet (F9)

```
| SIP/2.0 200 OK
| To: <sip:juliet@example.com>;gr=balcony
| From: <sip:romeo@example.net>
| Contact: <sip:romeo@example.net>;gr=orchard
| Call-ID: F6989A8C-DE8A-4E21-8E07-F0898304796F
| CSeq: 1 BYE
```

5. Composing Events

Both XMPP and MSRP enable a client to receive notifications when a person's conversation partner is composing an instant message within the context of a chat session.

For XMPP, the Chat State Notifications specification [XEP-0085] defines five states: active, inactive, gone, composing, and paused. Some of these states are related to the act of message composition (composing, paused), whereas others are related to the sender's involvement with the chat session (active, inactive, gone).

For MSRP (and SIP/SIMPLE in general), the Indication of Message Composition for Instant Messaging specification [RFC3994] defines two states: idle and active. Here the idle state indicates that the sender is not actively composing a message, and the active state indicates that the sender is indeed actively composing a message (the sending client simply toggles between the two states, changing to active if the user is actively composing a message and changing to idle if the user is no longer actively composing a message).

Because the XEP-0085 states can represent information that is not captured in RFC 3994, gateways can either (a) map only the composing-related states or (b) map all the XEP-0085 states.

The following mappings are suggested.

Table 3: Mapping of SIP/SIMPLE isComposing events to XMPP chat states

isComposing Event	Chat State
active	composing
idle	active

Table 4: Mapping of XMPP chat states to SIP/SIMPLE isComposing events

Chat State	isComposing Event
active	idle
inactive	idle
gone	[none, see note]
composing	active
paused	idle

Although there is no direct mapping for the "gone" chat state (which is not to be confused with the <gone/> stanza error condition defined in [RFC6120]) to an isComposing event, receipt of the "gone" state can be used as a trigger for terminating the formal chat session within MSRP, i.e., for sending a SIP BYE for the session from the XMPP-SIP gateway to the SIP user. The following examples illustrate this indirect mapping.

Example 19: Juliet sends gone chat state

```
| <message from='juliet@example.com'
|          id='nx62f197'
|          to='romeo@example.net'
|          type='chat'>
|   <thread>29377446-0CBB-4296-8958-590D79094C50</thread>
|   <gone xmlns='http://jabber.org/protocol/chatstates'/>
| </message>
```

Example 20: XMPP-SIP gateway maps gone chat state to SIP BYE

```
| BYE romeo@example.net sip: SIP/2.0
| From: <sip:juliet@example.com>;tag=786
| To: <sip:romeo@example.net>;tag=087js
| Call-ID: 29377446-0CBB-4296-8958-590D79094C50
| Cseq: 1 BYE
| Content-Length: 0
```

6. Delivery Reports

Both XMPP and MSRP enable a client to receive notifications when a message has been received by the intended recipient.

For XMPP, the Message Receipts specification [XEP-0184] defines a method and XML namespace for requesting and returning indications that a message has been received by a client controlled by the intended recipient.

For MSRP, a native reporting feature is included, in the form of REPORT chunks (see Sections 7.1.2 and 7.1.3 of [RFC4975]).

Examples follow.

First, the XMPP user sends a message containing a request for delivery notification.

Example 21: Juliet sends XMPP message with receipt request

```
| <message from='juliet@example.com'
|          id='bf9m36d5'
|          to='romeo@example.net'
|          type='chat'>
|   <thread>29377446-0CBB-4296-8958-590D79094C50</thread>
|   <body>What man art thou ...?</body>
|   <request xmlns='urn:xmpp:receipts'/>
| </message>
```

Example 22: Gateway maps XMPP message to MSRP

```
| MSRP bf9m36d5 SEND
| To-Path: msrp://s2x.example.net:7313/jshA7weztas;tcp
| From-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
| Message-ID: 6187CF9B-317A-41DA-BB6A-5E48A9C794EF
| Byte-Range: 1-25/25
| Success-Report: yes
| Failure-Report: no
| Content-Type: text/plain
|
| What man art thou ...?
| -------bf9m36d5$
```

Next, the recipient returns a report.

Example 23: Romeo returns MSRP receipt

```
| MSRP hx74g336 REPORT
| To-Path: msrp://x2s.example.com:8763/lkjh37s2s20w2a;tcp
| From-Path: msrp://s2x.example.net:7313/jshA7weztas;tcp
| Message-ID: 6187CF9B-317A-41DA-BB6A-5E48A9C794EF
| Byte-Range: 1-106/106
| Status: 000 200 OK
| -------hx74g336$
```

Example 24: SIP-XMPP gateway maps receipt to XMPP

```
| <message from='romeo@example.net'
|          id='hx74g336'
|          to='juliet@example.com'>
|   <received xmlns='urn:xmpp:receipts' id='87652491'/>
| </message>
```

7. Internationalization Considerations

Relevant discussion of internationalized text in messages can be found in [I-D.ietf-stox-im].

8. IANA Considerations

This document requests no actions of IANA.

9. Security Considerations

Detailed security considerations for instant messaging protocols are given in [RFC2779], for MSRP chat in [RFC4975] (see also [RFC3261] when SIP is used to negotiate MSRP sessions), and for XMPP-based instant messaging in [RFC6121] (see also [RFC6120]). The security considerations provided in [I-D.ietf-stox-core] also apply.

This document specifies methods for exchanging instant messages through a gateway that translates between SIP/MSRP and XMPP. Such a gateway MUST be compliant with the minimum security requirements of the textual chat protocols for which it translates (i.e., MSRP and XMPP). The addition of gateways to the security model of instant messaging specified in [RFC2779] introduces some new risks. In particular, end-to-end security properties (especially confidentiality and integrity) between instant messaging clients that interface through an MSRP-XMPP gateway can be provided only if common formats are supported. Specification of those common formats is out of scope for this document, although it is suggested to use [RFC3862] for instant messages.

10. References

10.1. Normative References

[I-D.ietf-stox-core]
Saint-Andre, P., Houri, A., and J. Hildebrand, "Interworking between the Session Initiation Protocol (SIP) and the Extensible Messaging and Presence Protocol (XMPP): Core", draft-ietf-stox-core-09 (work in progress), December 2013.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 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.

[RFC3862] Klyne, G. and D. Atkins, "Common Presence and Instant Messaging (CPIM): Message Format", RFC 3862, August 2004.

[RFC3994] Schulzrinne, H., "Indication of Message Composition for Instant Messaging", RFC 3994, January 2005.

[RFC4975] Campbell, B., Mahy, R., and C. Jennings, "The Message Session Relay Protocol (MSRP)", RFC 4975, September 2007.

[RFC6120] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Core", RFC 6120, March 2011.

[RFC6121] Saint-Andre, P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", RFC 6121, March 2011.

[XEP-0085] Saint-Andre, P. and D. Smith, "Chat State Notifications", XSF XEP 0085, September 2009.

[XEP-0184] Saint-Andre, P. and J. Hildebrand, "Message Delivery Receipts", XSF XEP 0184, March 2011.

10.2. Informative References

[I-D.ietf-stox-im] 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", draft-ietf-stox-im-06 (work in progress), December 2013.

[RFC2779] Day, M., Aggarwal, S., and J. Vincent, "Instant Messaging / Presence Protocol Requirements", RFC 2779, February 2000.

Appendix A. Acknowledgements

Special thanks to Eddy Gavita and Nazin Hossain for their co-authorship of an early version of this document.

Thanks to Mary Barnes, Adrian Georgescu, Philipp Hancke, Saul Ibarra Corretge, and Tory Patnoe for their feedback.

The authors gratefully acknowledge the assistance of Markus Isomaki and Yana Stamcheva as the working group chairs and Gonzalo Camarillo as the sponsoring Area Director.

Authors' Addresses

Peter Saint-Andre

Email: ietf@stpeter.im

Salvatore Loreto
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Salvatore.Loreto@ericsson.com