



XMPP

XEP-0353: Jingle Message Initiation

Philipp Hancke

<mailto:fippo@andyet.com>

<xmpp:fippo@goodadvice.pages.de>

Peter Saint-Andre

<mailto:stpeter@stpeter.im>

<xmpp:stpeter@jabber.org>

<https://stpeter.im/>

Thilo Molitor

<mailto:thilo+xmpp@eightysoft.de>

<xmpp:thilo.molitor@juforum.de>

2022-01-29

Version 0.6.0

Status	Type	Short Name
Experimental	Standards Track	jingle-message

This specification provides a way for the initiator of a Jingle session to propose sending an invitation in an XMPP message stanza, thus taking advantage of message delivery semantics instead of sending IQ stanzas to all of the responder's online resources or choosing a particular online resource.

Legal

Copyright

This XMPP Extension Protocol is copyright © 1999 – 2024 by the [XMPP Standards Foundation](#) (XSF).

Permissions

Permission is hereby granted, free of charge, to any person obtaining a copy of this specification (the "Specification"), to make use of the Specification without restriction, including without limitation the rights to implement the Specification in a software program, deploy the Specification in a network service, and copy, modify, merge, publish, translate, distribute, sublicense, or sell copies of the Specification, and to permit persons to whom the Specification is furnished to do so, subject to the condition that the foregoing copyright notice and this permission notice shall be included in all copies or substantial portions of the Specification. Unless separate permission is granted, modified works that are redistributed shall not contain misleading information regarding the authors, title, number, or publisher of the Specification, and shall not claim endorsement of the modified works by the authors, any organization or project to which the authors belong, or the XMPP Standards Foundation.

Warranty

NOTE WELL: This Specification is provided on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

Liability

In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall the XMPP Standards Foundation or any author of this Specification be liable for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising from, out of, or in connection with the Specification or the implementation, deployment, or other use of the Specification (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if the XMPP Standards Foundation or such author has been advised of the possibility of such damages.

Conformance

This XMPP Extension Protocol has been contributed in full conformance with the XSF's Intellectual Property Rights Policy (a copy of which can be found at <https://xmpp.org/about/xsf/ipr-policy>) or obtained by writing to XMPP Standards Foundation, P.O. Box 787, Parker, CO 80134 USA).

Contents

1	Introduction	1
2	Requirements	1
3	Use Cases	2
3.1	Indicating Intent to Start a Session	2
3.2	Indicating a Ringing Device	3
3.3	Disavowing Intent to Start a Session	3
3.4	Accepting Intent to Start a Session	4
3.5	Rejecting Intent to Start a Session	5
3.6	Initiating the Jingle Session	6
3.7	Finishing a Started Session	7
4	Tie Breaking	8
4.1	No existing session	8
4.2	Existing session	10
5	Business Rules	12
6	Security Considerations	13
7	Acknowledgements	13
8	IANA Considerations	13
9	XMPP Registrar Considerations	13
9.1	Protocol Namespaces	13
9.2	Protocol Versioning	14
10	Design Considerations	14
11	XML Schema	14

1 Introduction

Jingle (XEP-0166) ¹ uses <iq/> stanzas for all interactions between the parties to a jingle session. When sending an invitation the initiator needs to either pick one of the responder's resources (e.g., based on Entity Capabilities (XEP-0115) ² information) or send the invitation to all of the responder's resources that support Jingle. The first method is prone to error (e.g., in cases where more than one resource supports Jingle) and the second method requires sending a separate invitation to each resource. Neither of these is ideal. Although Presence Decloaking (XEP-0276) ³ proposed a way to overcome the problem, it too has issues (e.g., dependency on a presence service and the need to reveal all supported XMPP features) and in any case has not been widely implemented.

This document proposes an alternative solution: exchanging a <message/> stanza before sending the Jingle invitation in an <iq/> stanza. (Indeed, in the early discussions leading up to the Jingle protocol the authors considered using <message/> stanzas instead of <iq/> stanzas, but chose the latter for their deterministic handling semantics.) This method effectively results in a kind of decloaking for Jingle purposes.

Message Carbons (XEP-0280) ⁴ and Message Archive Management (XEP-0313) ⁵ (including Message Processing Hints (XEP-0334) ⁶) make sure all devices (offline or not) know about the jingle session, its start timestamp and even its end timestamp and the state in between.

2 Requirements

This protocol was designed with the following requirements in mind:

- Allow responder to choose the resource or device on which to take the call.
- Result in "ring-on-all-devices" behavior.
- Not rely on presence stanzas.
- Make use of "forking" semantics for message stanzas.
- Allow indication of session content.
- Work with push notifications.
- Work with offline devices
- Work with devices that use negative resource priorities

¹XEP-0166: Jingle <<https://xmpp.org/extensions/xep-0166.html>>.

²XEP-0115: Entity Capabilities <<https://xmpp.org/extensions/xep-0115.html>>.

³XEP-0276: Presence Decloaking <<https://xmpp.org/extensions/xep-0276.html>>.

⁴XEP-0280: Message Carbons <<https://xmpp.org/extensions/xep-0280.html>>.

⁵XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

⁶XEP-0334: Message Processing Hints <<https://xmpp.org/extensions/xep-0334.html>>.

- Allow all devices of the involved users to display start and end timestamps of the call

3 Use Cases

All <message/> stanzas exchanged by this protocol MUST be of type="chat" and contain [Message Processing Hints \(XEP-0334\)](#)⁷ <store/> hints.

3.1 Indicating Intent to Start a Session

In order to prepare for sending a Jingle invitation, the initiator (e.g., Romeo) sends a <message/> stanza containing a <propose/> element qualified by the 'urn:xmpp:jingle-message:0' namespace. The <propose/> element MUST possess an 'id' attribute being a globally unique identifier. It therefore is RECOMMENDED to use UUIDv4. This id will also be used for the session invitation of [Jingle \(XEP-0166\)](#)⁸ later on. The <propose/> element MUST contain one <description/> element for each media type associated with the intended session.

Listing 1: Initiator Sends Intent Message

```
<message from='romeo@montague.example/orchard'
  to='juliet@capulet.example'
  type='chat'>
  <propose xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
    a412-a6e9f832298d'>
    <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio' />
  </propose>
  <store xmlns="urn:xmpp:hints" />
</message>
```

The server of the responder (e.g., Juliet) distributes this message stanza to all of Juliet's available resources (and to push resources as appropriate) thanks to [Message Carbons \(XEP-0280\)](#)⁹ and [Message Archive Management \(XEP-0313\)](#)¹⁰. Those devices might start ringing as a result.

Consistent with the recommendation for one-to-one chat sessions in Section 5.1 of [RFC 6121](#)¹¹, the initiator SHOULD also send directed presence to the responder if the two entities do not already share presence information; including Entity Capabilities (XEP-0115) information in this directed presence stanza enables the responder to know the availability of the initiator (e.g., in case the message is actually delivered quite a bit later because it is saved to [Message Archive Management \(XEP-0313\)](#)¹² storage) and also to know the XMPP features supported

⁷XEP-0334: Message Processing Hints <<https://xmpp.org/extensions/xep-0334.html>>.

⁸XEP-0166: Jingle <<https://xmpp.org/extensions/xep-0166.html>>.

⁹XEP-0280: Message Carbons <<https://xmpp.org/extensions/xep-0280.html>>.

¹⁰XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

¹¹RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <<http://tools.ietf.org/html/rfc6121>>.

¹²XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

by the initiator.

Listing 2: Initiator Sends Directed Presence

```
<presence to='romeo@montague.example/orchard'
  from='juliet@capulet.example'>
  <c xmlns='http://jabber.org/protocol/caps'
    hash='sha-1'
    node='http://psi-im.org'
    ver='q07IKJEyJvHSyhy//CH0CxmKi8w=' />
</presence>
```

3.2 Indicating a Ringing Device

Upon receiving the <propose/> message, the responder's various devices will start "ringing" and indicate so by sending a message to the bare JID of the initiator containing a <ringing/> element qualified by the 'urn:xmpp:jingle-message:0' namespace and specifying the session ID of the original <propose/> message.

This makes it possible to reflect the real state of the call in the UI and therefore comprises for better UX. It also somewhat compensates for the (intentionally) missing discovery of this protocol.

Listing 3: One of Responder's Resources Reports it is Ringing

```
<message from='juliet@capulet.example/phone'
  to='romeo@montague.example'
  type='chat'>
  <ringing xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
    a412-a6e9f832298d' />
  <store xmlns="urn:xmpp:hints" />
</message>
```

3.3 Disavowing Intent to Start a Session

It can happen that the initiator might want to disavow intent to send a session invitation (e.g., because the initiator has accepted another session). The initiator can do so by sending a message stanza containing a <retract/> element qualified by the 'urn:xmpp:jingle-message:0' namespace and specifying the session ID of the original <propose/> message.

The <retract/> element SHOULD contain a <reason/> element as defined in [Jingle \(XEP-0166\)](#)¹³ section 7.4. This SHOULD use a condition of <cancel/>, but implementations MAY use other conditions if deemed more appropriate (see [Security Considerations](#) below for details and rationale).

In conjunction with [Message Archive Management \(XEP-0313\)](#)¹⁴ upon ending the catchup

¹³XEP-0166: Jingle <<https://xmpp.org/extensions/xep-0166.html>>.

¹⁴XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

phase the responder SHOULD consider all sessions for which it received a <propose/> but no <retract/> or <finish/> message to be still active and allow the user to [accept the intent to start a session](#).

Listing 4: Initiator Sends Retract Message

```
<message from='romeo@montague.example/orchard'
  to='juliet@capulet.example'
  type='chat'>
  <retract xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
    a412-a6e9f832298d'>
    <reason xmlns="urn:xmpp:jingle:1">
      <cancel/>
      <text>Retracted</text>
    </reason>
  </retract>
  <store xmlns="urn:xmpp:hints"/>
</message>
```

3.4 Accepting Intent to Start a Session

The responder will answer the call on a particular device. Here we assume that since this is an audio-only call, Juliet chooses to take the call on the device associated with her "phone" resource.

Her "phone" resource informs all of her resources and all of the initiator's resources about accepting the call by sending a message to the bare JID of the initiator containing an <proceed/> element qualified by the 'urn:xmpp:jingle-message:0' namespace and specifying the session ID of the original <propose/> message.

Listing 5: One of Responder's Resources Accepts the Call

```
<message from='juliet@capulet.example/phone'
  to='romeo@montague.example'
  type='chat'>
  <proceed xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
    a412-a6e9f832298d' />
  <store xmlns="urn:xmpp:hints"/>
</message>
```

Juliet's server broadcasts this accept message to all of her resources (as described in [Message Carbons \(XEP-0280\)](#) ¹⁵), which stop ringing, and to all of Romeo's resources (as described in [RFC 6121](#) ¹⁶). Romeo's resources that did not send the <propose/> can use this <message/> stanza to update their UI or choose to ignore this <message/> stanza altogether.

¹⁵XEP-0280: Message Carbons <<https://xmpp.org/extensions/xep-0280.html>>.

¹⁶RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <<http://tools.ietf.org/html/rfc6121>>.

Next, the device from which Juliet accepted the call SHOULD also send directed presence to Romeo if the two entities do not already share presence information, for the reasons described above.

Listing 6: Responder Sends Directed Presence

```
<presence from='juliet@capulet.example/phone'
          to='romeo@montague.example/orchard'>
  <c xmlns='http://jabber.org/protocol/caps'
    hash='sha-1'
    node='http://code.google.com/p/exodus'
    ver='QgayPKawpkPSDYmwT/WM94uAlu0=' />
</presence>
```

3.5 Rejecting Intent to Start a Session

Instead of accepting the call, the responder might want to decline the call and tell all of her devices to stop ringing (e.g., perhaps because Romeo is getting to be a bit of a nuisance). She does this by rejecting the call on one of her devices and having that device tell all of the other devices to stop ringing by sending a <message/> stanza containing a <reject/> element qualified by the 'urn:xmpp:jingle-message:0' namespace and specifying the session ID of the original <propose/> message to the bare JID of Romeo.

The <reject/> element SHOULD contain a <reason/> element as defined in [Jingle \(XEP-0166\)](#)¹⁷ section 7.4. If given, the <reason/> element SHOULD use a condition of <busy/>, but implementations MAY use other conditions if deemed more appropriate (see [Security Considerations](#) below for details and rationale).

In Tie-Breaking scenarios it MUST also contain a <tie-break/> element as defined in [Tie Breaking](#).

Listing 7: One of Responder's Resources Rejects the Call

```
<message from='juliet@capulet.example/phone'
          to='romeo@montague.example'
          type='chat'>
  <reject xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
    a412-a6e9f832298d'>
    <reason xmlns='urn:xmpp:jingle:1'>
      <busy/>
      <text>Busy</text>
    </reason>
  </reject>
  <store xmlns='urn:xmpp:hints' />
</message>
```

¹⁷XEP-0166: Jingle <<https://xmpp.org/extensions/xep-0166.html>>.

3.6 Initiating the Jingle Session

Now Romeo actually initiates the session (segue to Jingle itself).

Listing 8: Initiation

```
<iq from='romeo@montague.example/orchard'
  id='ih28sx61'
  to='juliet@capulet.example/phone'
  type='set'>
  <jingle xmlns='urn:xmpp:jingle:1'
    action='session-initiate'
    initiator='romeo@montague.example/orchard'
    sid='ca3cf894-5325-482f-a412-a6e9f832298d'>
    <content creator='initiator' name='voice'>
      <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio'>
        <payload-type id='96' name='speex' clockrate='16000' />
        <payload-type id='97' name='speex' clockrate='8000' />
        <payload-type id='18' name='G729' />
        <payload-type id='0' name='PCMU' />
        <payload-type id='103' name='L16' clockrate='16000' channels='
          2' />
        <payload-type id='98' name='x-ISAC' clockrate='8000' />
      </description>
      <transport xmlns='urn:xmpp:jingle:transports:ice-udp:1'
        pwd='asd88fgpdd777uzjYhagZg'
        ufrag='8hhy'>
        <candidate component='1'
          foundation='1'
          generation='0'
          id='el0747fg11'
          ip='10.0.1.1'
          network='1'
          port='8998'
          priority='2130706431'
          protocol='udp'
          type='host' />
        <candidate component='1'
          foundation='2'
          generation='0'
          id='y3s2b30v3r'
          ip='192.0.2.3'
          network='1'
          port='45664'
          priority='1694498815'
          protocol='udp'
          rel-addr='10.0.1.1'
          rel-port='8998'
          type='srflx' />
      </transport>
    </content>
  </jingle>
</iq>
```

```
        </transport>
      </content>
    </jingle>
  </iq>
```

3.7 Finishing a Started Session

This protocol in conjunction with [Message Carbons \(XEP-0280\)](#)¹⁸ and [Message Archive Management \(XEP-0313\)](#)¹⁹ allows all devices of both involved parties to get synchronized about session start, rejection etc. To synchronize the ending of the session, both parties SHOULD send a message stanza containing a <finish/> element qualified by the 'urn:xmpp:jingle-message:0' namespace and specifying the same session ID as in [proceed](#) to the bare jid of the other party.

Letting both involved parties send the <finish/> element makes sure we have the correct state in MAM archives etc. even if one client suddenly loses connectivity/power. It even makes possible for a client to determine if the call is still deemed "running" by the other party if it manages to recover from connectivity loss — before the other party runs into a timeout and sends a <finish/> — to recover the session or formally terminate the call (by ending the jingle session and sending a <finish/> message itself). See [Tie Breaking](#) for more infos on this and similar scenarios.

The <finish/> element SHOULD contain a <reason/> element as defined in [Jingle \(XEP-0166\)](#)²⁰ section 7.4. This SHOULD use a condition of <success/> or <expired/>, but implementations MAY use other conditions like <connectivity-error/> if deemed more appropriate (see [Security Considerations](#) below for details and rationale).

Listing 9: Initiator Sends Finish Message

```
<message from='romeo@montague.example/orchard'
  to='juliet@capulet.example'
  type='chat'>
  <finish xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
    a412-a6e9f832298d'>
    <reason xmlns="urn:xmpp:jingle:1">
      <success/>
      <text>Success</text>
    </reason>
  </finish>
  <store xmlns="urn:xmpp:hints"/>
</message>
```

Listing 10: Responder Sends Finish Message

¹⁸XEP-0280: Message Carbons <<https://xmpp.org/extensions/xep-0280.html>>.

¹⁹XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

²⁰XEP-0166: Jingle <<https://xmpp.org/extensions/xep-0166.html>>.

```

<message from='juliet@capulet.example/phone'
  to='romeo@montague.example'
  type='chat'>
  <finish xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
    a412-a6e9f832298d'>
    <reason xmlns="urn:xmpp:jingle:1">
      <success/>
      <text>Success</text>
    </reason>
  </finish>
  <store xmlns="urn:xmpp:hints"/>
</message>

```

4 Tie Breaking

It is possible that a <propose/> message can be sent at the same time by both parties or a new session started while one is already running. Implementations of this specification SHOULD implement the following solutions to solve this. (This is loosely based upon section 7.2.16 of [Jingle \(XEP-0166\)](#) ²¹.)

4.1 No existing session

In this case (e.g. no party answered the <propose/> message yet) the lower of the two session IDs MUST overrule the other action, where by "lower" is meant the session ID that is sorted first using "i;octet" collation as specified in Section 9.3 of [RFC 4790](#) ²² (in the unlikely event that the random session IDs are the same, the action sent by the lower of the JabberIDs MUST overrule the other action).

The party that receives the <propose/> action with the lower of the two session IDs MUST send a <retract/> message for the higher session ID to the other party with a <tie-break/> child element alongside of a <reason/> element carrying the condition <expired/> and then eventually respond with an <proceed/> or <reject/> message like it would normally do for a received <propose/> message.

The party that receives the <propose/> action with the higher of the two session IDs MUST return a <reject/> message to the other party with a <tie-break/> child element alongside of a <reason/> element carrying the condition <expired/>.

Listing 11: Tie break in propose state

```

<!--{}- lower session ID -{}-->
<message from='romeo@montague.example/orchard'
  to='juliet@capulet.example'

```

²¹XEP-0166: Jingle <<https://xmpp.org/extensions/xep-0166.html>>.

²²RFC 4790: Internet Application Protocol Collation Registry <<http://tools.ietf.org/html/rfc4790>>.

```

        type='chat'>
        <propose xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
          a412-a6e9f832298d'>
          <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio' />
        </propose>
        <store xmlns="urn:xmpp:hints"/>
      </message>

<!--{}- higher session ID -{}-->
<message from='juliet@capulet.example/phone'
  to='romeo@montague.example'
  type='chat'>
  <propose xmlns='urn:xmpp:jingle-message:0' id='fecbea35-08d3-404f-9
    ec7-2b57c566fa74'>
    <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio' />
  </propose>
  <store xmlns="urn:xmpp:hints"/>
</message>

<!--{}- Romeo received the higher ID and rejects the call -{}-->
<message from='romeo@montague.example/orchard'
  to='juliet@capulet.example'
  type='chat'>
  <reject xmlns='urn:xmpp:jingle-message:0' id='fecbea35-08d3-404f-9
    ec7-2b57c566fa74'>
    <reason xmlns="urn:xmpp:jingle:1">
      <expired/>
      <text>Tie-Break</text>
    </reason>
    <tie-break/>
  </reject>
  <store xmlns="urn:xmpp:hints"/>
</message>

<!--{}- Juliet sent the higher ID and retracts the call -{}-->
<message from='juliet@capulet.example/phone'
  to='romeo@montague.example'
  type='chat'>
  <retract xmlns='urn:xmpp:jingle-message:0' id='fecbea35-08d3-404f-9
    ec7-2b57c566fa74'>
    <reason xmlns="urn:xmpp:jingle:1">
      <expired/>
      <text>Tie-Break</text>
    </reason>
    <tie-break/>
  </retract>
  <store xmlns="urn:xmpp:hints"/>
</message>

```

```

<!--{}- Juliet received the lower ID and accepts the call -{}->
<message from='juliet@capulet.example/phone'
        to='romeo@montague.example'
        type='chat'>
  <proceed xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
    a412-a6e9f832298d' />
  <store xmlns="urn:xmpp:hints" />
</message>

```

4.2 Existing session

If (from the perspective of the responder of the new session) there is already a session to the bare-jid of the initiator active (e.g. call already accepted but no <finish/> element received by the responder so far), the old session MUST be deemed an orphan and terminated by the responder of the new session in favor of the new one. The responder SHOULD transparently accept the new session and finish the old one, because it can be assumed that this new session is a transparent continuation of the old one.

The responder does so by sending a <finish/> message including a <reason/> condition of <expired/> and having a <migrated> child element whose to-attribute refers to the new Jingle session id, and accepting the new session by sending an <proceed/> message like they would do normally.

That makes it possible for the initiator of the new session to transparently switch devices (e.g. migrate the call to a new device) or resume a still running session after a sudden connectivity/power loss.

Listing 12: Tie break in accept state

```

<!--{}- old session gets proposed... -{}->
<message from='romeo@montague.example/orchard'
        to='juliet@capulet.example'
        type='chat'>
  <propose xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
    a412-a6e9f832298d'>
    <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio' />
  </propose>
  <store xmlns="urn:xmpp:hints" />
</message>

<!--{}- ...and accepted -{}->
<message from='juliet@capulet.example/phone'
        to='romeo@montague.example'
        type='chat'>
  <proceed xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
    a412-a6e9f832298d' />
  <store xmlns="urn:xmpp:hints" />
</message>

```

```

<!--{}- ...some time passes... -{}->

<!--{}- new session (device switch) -{}->
<message from='juliet@capulet.example/tablet'
        to='romeo@montague.example'
        type='chat'>
  <propose xmlns='urn:xmpp:jingle-message:0' id='989a46a6-f202-4910-
    a7c3-83c6ba3f3947'>
    <description xmlns='urn:xmpp:jingle:apps:rtp:1' media='audio'/>
  </propose>
  <store xmlns="urn:xmpp:hints"/>
</message>

<!--{}- Romeo finishes the old session with id 'ca3cf894-5325-482f-a412-
  a6e9f832298d' using reason <expired/> -{}->
<!--{}- because the new one is assumed to be the continuation of the
  old one. -{}->
<!--{}- This is including a <migrated/> element pointing to the new
  session id '989a46a6-f202-4910-a7c3-83c6ba3f3947'... -{}->
<message from='romeo@montague.example/orchard'
        to='juliet@capulet.example'
        type='chat'>
  <finish xmlns='urn:xmpp:jingle-message:0' id='ca3cf894-5325-482f-
    a412-a6e9f832298d'>
    <reason xmlns="urn:xmpp:jingle:1">
      <expired/>
      <text>Session migrated</text>
    </reason>
    <migrated to='989a46a6-f202-4910-a7c3-83c6ba3f3947' />
  </finish>
  <store xmlns="urn:xmpp:hints"/>
</message>

<!--{}- ...and directly afterwards accepts the new call with id '989
  a46a6-f202-4910-a7c3-83c6ba3f3947'. -{}->
<message from='romeo@montague.example/orchard'
        to='juliet@capulet.example'
        type='chat'>
  <proceed xmlns='urn:xmpp:jingle-message:0' id='989a46a6-f202-4910-
    a7c3-83c6ba3f3947' />
  <store xmlns="urn:xmpp:hints"/>
</message>

<!--{}- ...some more time passes... -{}->

<!--{}- the new session is terminated normally by Romeo... -{}->
<message from='romeo@montague.example/orchard'
        to='juliet@capulet.example'
        type='chat'>

```

```

<finish xmlns='urn:xmpp:jingle-message:0' id='989a46a6-f202-4910-
a7c3-83c6ba3f3947'>
  <reason xmlns="urn:xmpp:jingle:1">
    <success/>
    <text>Success</text>
  </reason>
</finish>
<store xmlns="urn:xmpp:hints"/>
</message>

<!--{}- and Juliet... -{}-->
<message from='juliet@capulet.example/tablet'
to='romeo@montague.example'
type='chat'>
  <finish xmlns='urn:xmpp:jingle-message:0' id='989a46a6-f202-4910-
a7c3-83c6ba3f3947'>
    <reason xmlns="urn:xmpp:jingle:1">
      <success/>
      <text>Success</text>
    </reason>
  </finish>
  <store xmlns="urn:xmpp:hints"/>
</message>

```

5 Business Rules

Participants MUST use [Message Carbons \(XEP-0280\)](#)²³ and [Message Archive Management \(XEP-0313\)](#)²⁴ to make sure all devices of initiator and responder receive all messages exchanged by this protocol. Without [Message Carbons \(XEP-0280\)](#)²⁵ implementations would need to send copies of outgoing messages to their own bare jid, to inform their own devices about an event (like it was done with the <accept/> message in the old urn:xmpp:jingle:jingle-message:0 specification).

In a [Message Archive Management \(XEP-0313\)](#)²⁶ (or [Stream Management \(XEP-0198\)](#)²⁷) catchup scenario client developers MAY choose to not show an "incoming call" UI upon receiving a <propose/> message because they could receive another message for the same Jingle session id later in the catchup process invalidating the <propose/> received before. Showing the "incoming call" UI as soon as receiving a <propose/> might comprise bad UX.

In the rare case of missing <finish/> elements from both initiator and responder, sessions SHOULD be considered terminated after an appropriate timeframe (for example 24 hours) and indicated so in the UI.

²³XEP-0280: Message Carbons <<https://xmpp.org/extensions/xep-0280.html>>.

²⁴XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

²⁵XEP-0280: Message Carbons <<https://xmpp.org/extensions/xep-0280.html>>.

²⁶XEP-0313: Message Archive Management <<https://xmpp.org/extensions/xep-0313.html>>.

²⁷XEP-0198: Stream Management <<https://xmpp.org/extensions/xep-0198.html>>.

All 'id' attributes MUST be globally unique to make sure they do not collide, and therefore it is RECOMMENDED to use UUIDv4.

6 Security Considerations

Because exchanging messages with other entities is effectively a presence leak, an XMPP client that implements the receiving side of this specification MUST disable sending of accept messages by default and MUST enable the feature only as a result of explicit user confirmation. Such confirmation can be provided per request, by automatically allowing requests received from Jingle initiators in the responder's contact list, or through some other suitable means as long as sending accept messages does not occur by default.

Because sending of reasons other than the default ones (e.g. <cancel/> for <retract/>, <busy/> or <expired/> for <reject/> and <success/> or <expired/> (or <connectivity-error/>) for <finish/>) may leak privacy related information the user does not want to leak, sending of those non-default reasons should be carefully considered by client developers.

7 Acknowledgements

Thanks to Lance Stout, Holger Weiß and Daniel Gultsch for their feedback.

8 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)²⁸.

9 XMPP Registrar Considerations

9.1 Protocol Namespaces

This specification defines the following XML namespace:

- urn:xmpp:jingle:jingle-message:0

²⁸The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of unique parameter values for Internet protocols, such as port numbers and URI schemes. For further information, see <http://www.iana.org/>.

The [XMPP Registrar](#)²⁹ includes the foregoing namespace to the registry located at [<https://xmpp.org/registrar/namespaces.html>](https://xmpp.org/registrar/namespaces.html), as described in Section 4 of [XMPP Registrar Function \(XEP-0053\)](#)³⁰.

9.2 Protocol Versioning

If the protocol defined in this specification undergoes a revision that is not fully backwards-compatible with an older version, the XMPP Registrar shall increment the protocol version number found at the end of the XML namespaces defined herein, as described in Section 4 of XEP-0053.

10 Design Considerations

Versions 0.4 and 0.5 of this specification define more or less the same protocol in the namespace `urn:xmpp:jingle:jingle-message:1` (but in many places using `MUST` rather than `SHOULD` and removing `<proceed/>` in favor of `<accept/>`). To provide for greater backwards compatibility, version 0.6 of this document switched back to the old `urn:xmpp:jingle:jingle-message:0` namespace. Future updates requiring a namespace bump should therefore directly bump the namespace version to `:2` and skip `:1`.

11 XML Schema

```
<?xml version='1.0' encoding='UTF-8'?>

<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  xmlns:xm='http://www.w3.org/XML/1998/namespace'
  targetNamespace='urn:xmpp:jingle-message:0'
  xmlns='urn:xmpp:jingle-message:0'
  elementFormDefault='qualified'>

  <xs:element name='propose'>
    <xs:complexType>
      <xs:sequence>
        <xs:any namespace='##other' minOccurs='1' maxOccurs='unbounded' />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

²⁹The XMPP Registrar maintains a list of reserved protocol namespaces as well as registries of parameters used in the context of XMPP extension protocols approved by the XMPP Standards Foundation. For further information, see <https://xmpp.org/registrar/>.

³⁰XEP-0053: XMPP Registrar Function [<https://xmpp.org/extensions/xep-0053.html>](https://xmpp.org/extensions/xep-0053.html).

```

    <xs:attribute name='id' type='xs:string' use='required' />
  </xs:complexType>
</xs:element>

<xs:element name='accept'>
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base='empty'>
        <xs:attribute name='id' type='xs:string' use='required' />
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>

<xs:element name='finish'>
  <xs:complexType>
    <xs:sequence>
      <xs:element name='reason' type='reasonElementType' minOccurs='1' maxOccurs='1' />
      <xs:element name='migrated' type='migratedElementType' minOccurs='0' maxOccurs='1' />
    </xs:sequence>
    <xs:attribute name='id' type='xs:string' use='required' />
  </xs:complexType>
</xs:element>

<xs:element name='reject'>
  <xs:complexType>
    <xs:sequence>
      <xs:element name='reason' type='reasonElementType' minOccurs='1' maxOccurs='1' />
      <xs:element name='tie-break' type='empty' minOccurs='0' maxOccurs='1' />
    </xs:sequence>
    <xs:attribute name='id' type='xs:string' use='required' />
  </xs:complexType>
</xs:element>

<xs:element name='retract'>
  <xs:complexType>
    <xs:sequence>
      <xs:element name='reason' type='reasonElementType' minOccurs='1' maxOccurs='1' />
    </xs:sequence>
    <xs:attribute name='id' type='xs:string' use='required' />
  </xs:complexType>
</xs:element>

<xs:complexType name='migratedElementType'>

```

```
<xs:simpleContent>
  <xs:extension base='empty'>
    <xs:attribute name='to' type='xs:string' use='required' />
  </xs:extension>
</xs:simpleContent>
</xs:complexType>

<xs:complexType name='reasonElementType'>
  <xs:sequence>
    <xs:choice>
      <xs:element name='alternative-session'
        type='alternativeSessionElementType' />
      <xs:element name='busy' type='empty' />
      <xs:element name='cancel' type='empty' />
      <xs:element name='connectivity-error' type='empty' />
      <xs:element name='decline' type='empty' />
      <xs:element name='expired' type='empty' />
      <xs:element name='failed-application' type='empty' />
      <xs:element name='failed-transport' type='empty' />
      <xs:element name='general-error' type='empty' />
      <xs:element name='gone' type='empty' />
      <xs:element name='incompatible-parameters' type='empty' />
      <xs:element name='media-error' type='empty' />
      <xs:element name='security-error' type='empty' />
      <xs:element name='success' type='empty' />
      <xs:element name='timeout' type='empty' />
      <xs:element name='unsupported-applications' type='empty' />
      <xs:element name='unsupported-transports' type='empty' />
    </xs:choice>
    <xs:element name='text' type='xs:string' minOccurs='0' maxOccurs='1' />
    <xs:any namespace='##other' minOccurs='0' maxOccurs='1' />
  </xs:sequence>
</xs:complexType>
</xs:schema>
```