



# XMPP

## XEP-0276: Presence Decloaking

Simon McVittie

<mailto:simon.mcvittie@collabora.co.uk>

<xmpp:simon.mcvittie@collabora.co.uk>

Peter Saint-Andre

<mailto:xf@stpeter.im>

<xmpp:peter@jabber.org>

<http://stpeter.im/>

Robert McQueen

<mailto:robert.mcqueen@collabora.co.uk>

<xmpp:robert.mcqueen@collabora.co.uk>

2012-07-13

Version 0.3

Status	Type	Short Name
Deferred	Standards Track	NOT YET ASSIGNED

This specification defines an XMPP protocol extension that enables a user to send directed presence with a request for the target to also share presence information for the duration of a communications session.

# Legal

## Copyright

This XMPP Extension Protocol is copyright © 1999 – 2018 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	Approach	2
4	Scenario	2
5	Sharing Presence with a Gateway	3
6	The reason Attribute	4
7	Business Rules	5
8	Security Considerations	5
9	IANA Considerations	5
10	XMPP Registrar Considerations	6
11	XML Schema	6
12	Acknowledgements	6

## 1 Introduction

Various XMPP extensions, such as [Jingle \(XEP-0166\)](https://xmpp.org/extensions/xep-0166.html)<sup>1</sup>, require additional support from clients, advertised in presence via [Entity Capabilities \(XEP-0115\)](https://xmpp.org/extensions/xep-0115.html)<sup>2</sup>, or require that IQ stanzas are sent to a particular resource. For instance, Jingle calls can be made only by sending an IQ to a particular resource. However, two parties who wish to communicate do not always share presence information through subscriptions and therefore cannot use entity capabilities to determine the proper full JID for communication. Indeed, one of the parties might not even use XMPP: e.g., a remote user on the other side of a gateway to a network based on the Session Initiation Protocol (SIP; [RFC 3261](http://tools.ietf.org/html/rfc3261))<sup>3</sup> or to the Public Switched Telephone Network (PSTN). It would be helpful if a user could make a call through such a gateway by typing the SIP URI or telephone number of an arbitrary contact, without first exchanging presence.

[RFC 6121](http://tools.ietf.org/html/rfc6121)<sup>4</sup> already defines a way to send directed presence to another entity. This document supplements RFC 6121 by defining an XML protocol extension enabling two parties to explicitly share presence with each other on a temporary basis through an "intentional presence leak"; we call this "decloaking".

Note: This protocol has already been implemented using an XML namespace of "http://telepathy.freedesktop.org/xmpp/protocol/decloak" but the [XMPP Registrar](https://xmpp.org/registrars/)<sup>5</sup> was requested to issue the XMPP URN "urn:xmpp:decloak:0" upon publication of this proposal in the [XMPP Extension Protocols \(XEP-0001\)](https://xmpp.org/extensions/xep-0001.html)<sup>6</sup> series.

## 2 Requirements

An entity should be able to attempt to initiate a communication session that requires IQs and/or capability negotiation (Jingle, a file transfer, end-to-end encryption, or other similar communication modes) with an arbitrary entity.

If the receiving entity agrees to divulge their presence in this way, minimal presence (with no 'type' attribute, <show/> element, avatar hash, etc.) and capabilities should be communicated to the initiating entity, so that the initiating entity can continue to initiate the communication session.

---

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

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

<sup>3</sup>RFC 3261: Session Initiation Protocol (SIP) <<http://tools.ietf.org/html/rfc3261>>.

<sup>4</sup>RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <<http://tools.ietf.org/html/rfc6121>>.

<sup>5</sup>The 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/registrars/>>.

<sup>6</sup>XEP-0001: XMPP Extension Protocols <<https://xmpp.org/extensions/xep-0001.html>>.

### 3 Approach

The approach taken here is that the user who wishes to initiate presence sharing for the length of a communications session sends directed presence (including entity capabilities) to the bare JID <localpart@domain.tld> of the initiator's intended communication partner, including a special XMPP extension <decloak xmlns='urn:xmpp:decloak:0'/>. Upon receipt of this directed presence stanza, if configured to do so the recipient's sends directed presence (including entity capabilities) to the initiator's full JID <localpart@domain.tld/resource>. Once the parties complete their communications session, they can terminate presence sharing by sending directed <presence type='unavailable'/> to each other; alternatively, at any time they could "upgrade" their session-based presence sharing to a full XMPP presence subscription as described in [XMPP IM](#)<sup>7</sup>.

Although the <decloak/> element could be sent in presence stanzas of type "subscribe" instead of in directed presence notifications, that behavior is discouraged because the "fall-through" case for subscription requests is a long-lived subscription, not temporary sharing of presence information for the life of a communication session.

### 4 Scenario

Suppose that Juliet wishes to make a media call to Tybalt, but the two parties do not share presence information in accordance with the core definition of XMPP. Suppose also that Juliet and Tybalt have the following presence, although neither can initially see the other's presence:

Listing 1: Initial state

```
<presence from='juliet@shakespeare.lit/balcony'>
  <show>dnd</show>
  <status>on the phone</status>
  <c ver='juliet-caps-hash' .../>
</presence>

<presence from='tybalt@shakespeare.lit/library'>
  <show>dnd</show>
  <status>researching</status>
  <c ver='tybalt-caps-hash' .../>
</presence>

<presence from='tybalt@shakespeare.lit/garden'>
  <show>xa</show>
  <status>gone to the library</status>
  <c ver='tybalt-caps-hash' .../>
</presence>
```

<sup>7</sup>RFC 6121: Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence <<http://tools.ietf.org/html/rfc6121>>.

Juliet requests that Tybalt divulge his availability and capabilities, by sending directed presence to his bare JID <tybalt@shakespeare.lit>, where the presence stanza contains a <decloak/> element.

Listing 2: Requesting that a peer share session presence

```
<presence from='juliet@shakespeare.lit/balcony'
          to='tybalt@shakespeare.lit'>
  <c ver='juliet-caps-hash' .../>
  <decloak xmlns='urn:xmpp:decloak:0' reason='media' />
</presence>
```

Tybalt MAY in response send session presence from one or more resources, by sending directed presence from those resource(s) to Juliet's bare JID.

Listing 3: Sharing presence in response to a request

```
<presence from='tybalt@shakespeare.lit/library'
          to='juliet@shakespeare.lit'>
  <c ver='tybalt-caps-hash' .../>
</presence>

<presence from='tybalt@shakespeare.lit/garden'
          to='juliet@shakespeare.lit'>
  <c ver='tybalt-caps-hash' .../>
</presence>
```

Once Juliet has received the session presence from Tybalt, if necessary she can perform service discovery to find out the meaning of the entity capabilities hashes (if unknown), then proceed to make a Jingle call, initiate a file transfer, or complete some other use case.

Naturally, it's also possible that Tybalt's client will ignore the request (in particular, this will happen for any resource that does not implement this specification). However, in this case the parties are no worse off than they were before Juliet requested decloaking.

## 5 Sharing Presence with a Gateway

Let us now suppose that Juliet wishes to make a media call to Romeo, who does not use XMPP but who has a SIP URI of sip:romeo@shakespeare.lit, which can be called via an XMPP-to-SIP gateway.

Juliet requests that the SIP contact representing Romeo on the gateway shall divulge its availability and capabilities, by sending directed presence to its bare JID at the gateway containing a <decloak/> element.

---

Listing 4: Requesting that a gateway contact shall share session presence

```
<presence from='juliet@shakespeare.lit/balcony'
          to='romeo%shakespeare.lit@sip.shakespeare.lit'>
  <c ver='juliet-caps-hash' .../>
  <decloak xmlns='urn:xmpp:decloak:0' reason='media' />
</presence>
```

In response, the SIP gateway automatically shares session presence on behalf of that JID, in order to tell Juliet what the gateway's capabilities are.

Listing 5: Sharing presence in response to a request

```
<presence from='romeo%shakespeare.lit@sip.shakespeare.lit'
          to='juliet@shakespeare.lit'>
  <c ver='gateway-caps-hash' .../>
</presence>
```

As above, Juliet can now complete service discovery and any protocol-specific use cases.

## 6 The reason Attribute

To signal the type of communication that is desired, the entity that first shares session presence MAY include a 'reason' attribute on the <decloak/> element. The following values for the 'reason' attribute are defined:

**media** Presence is requested for a voice and/or video call, e.g. via Jingle RTP Sessions (XEP-0167) XEP-0167: Jingle RTP Sessions <<https://xmpp.org/extensions/xep-0167.html>>..

**text** Presence is requested for a textual conversation using an extension that requires capabilities to be disclosed, such as XHTML-IM (XEP-0071) XEP-0071: XHTML-IM <<https://xmpp.org/extensions/xep-0071.html>>., Chat State Notifications (XEP-0085) XEP-0085: Chat State Notifications <<https://xmpp.org/extensions/xep-0085.html>>., In-Band Real Time Text (XEP-0301) XEP-0301: In-Band Real Time Text <<https://xmpp.org/extensions/xep-0301.html>>., or end-to-end encryption.

**file** Presence is requested for one or more file transfers, e.g. via Jingle File Transfer (XEP-0234) XEP-0234: Jingle File Transfer <<https://xmpp.org/extensions/xep-0234.html>>. or Stream Initiation (XEP-0095) XEP-0095: Stream Initiation <<https://xmpp.org/extensions/xep-0095.html>>..

Inclusion of the 'reason' attribute can be interpreted by the receiving client as a signal that communication is about to start; for instance, a call accept/reject dialog could double as a UI for accepting or rejecting a session presence request.

## 7 Business Rules

To limit the extent of the presence leak, the receiving entity SHOULD send only bare presence without the XMPP <priority/>, <show/>, or <status/> element. Unfortunately, this has two implications:

1. The initiating entity cannot know which of the receiving entity's resources is more likely to engage in communication. This might imply that the initiating entity will need to send a session initiation request or other communication to more than one of the receiving entity's resources (and then retract the session initiation requests that are not answered by the receiving entity). Solutions to that problem are out of scope for this specification.
2. Establishment of a session might be delayed (e.g., because in Jingle it is desirable to start negotiating candidates as soon as possible but a user interface that prompts the receiving entity to explicitly approve of divulging presence will tend to a delay in call setup). As a result, it may be advantageous to have a way to configure unconditional sharing of session presence in certain deployments, at least within the same trust domain.

## 8 Security Considerations

Because decloaking is a presence leak (albeit intentional), an XMPP client that implements the receiving side of this specification MUST disable sharing of session presence by default and MUST enable the feature only as a result of explicit user confirmation. Such confirmation can be provided per request, at the first request per requestor, by setting some "always decloak" configuration option (e.g., globally or per domain), or through some other suitable means as long as decloaking does not occur by default. (Gateways and other non-user entities MAY divulge their own presence and capabilities unconditionally, if that is appropriate for the service policy at the gateway.)

## 9 IANA Considerations

This document requires no interaction with the [Internet Assigned Numbers Authority \(IANA\)](#)<sup>8</sup>.

---

<sup>8</sup>The 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/>.



## 10 XMPP Registrar Considerations

The XMPP Registrar is requested to issue an initial namespace of "urn:xmpp:decloak:0".

## 11 XML Schema

```
<xs:schema
  xmlns:xs='http://www.w3.org/2001/XMLSchema'
  targetNamespace='urn:xmpp:decloak:0'
  xmlns='urn:xmpp:decloak:0'
  elementFormDefault='qualified'>

  <xs:element name='decloak'>
    <xs:complexType>
      <xs:simpleContent>
        <xs:extension base='empty'>
          <xs:attribute name='reason' use='optional' type='xs:string' />
        </xs:extension>
      </xs:simpleContent>
    </xs:complexType>
  </xs:element>

  <xs:simpleType name='empty'>
    <xs:restriction base='xs:string'>
      <xs:enumeration value='' />
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```

## 12 Acknowledgements

The need for this protocol extension, and a rough proposal for solving the problem, were originally determined at XMPP Summit 5 in the summer of 2007. Thanks to Diana Cionoiu, Justin Karneges, and Justin Uberti for their input to those discussions. Thanks also to Kurt Zeilenga for his feedback on the resulting specification.