

Mapping the Extensible Messaging and Presence Protocol (XMPP) to
Common Presence and Instant Messaging (CPIM)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2004).

Abstract

This memo describes a mapping between the Extensible Messaging and Presence Protocol (XMPP) and the Common Presence and Instant Messaging (CPIM) specifications.

Table of Contents

1. Introduction	2
2. Approach	3
3. Address Mapping	4
4. Syntax Mapping of Instant Messages	5
5. Syntax Mapping of Presence Information	13
6. XMPP-CPIM Gateway as Presence Service	26
7. Security Considerations	31
8. References	32
Author's Address	33
Full Copyright Statement	34

1. Introduction

1.1. Overview

The Instant Messaging and Presence (IMPP) Working Group has defined an abstract framework for interoperability among instant messaging (IM) and presence systems that are compliant with [IMP-REQS]. This framework is commonly called Common Presence and Instant Messaging or "CPIM". The CPIM family of specifications include a Common Profile for Instant Messaging [CPIM] (also called CPIM), a Common Profile for Presence [CPP], a CPIM Message Format [MSGFMT], and a Common Presence Information Data Format [PIDF]. (Note: To prevent confusion, Common Presence and Instant Messaging is referred to herein collectively as "the CPIM specifications", whereas the Common Profile for Instant Messaging is referred to as "CPIM".)

This memo describes how the Extensible Messaging and Presence Protocol ([XMPP-CORE], [XMPP-IM]) maps to the abstract model contained in the CPIM specifications, mainly for the purpose of establishing gateways between XMPP services and non-XMPP services that conform to [IMP-REQS]. Such a gateway, referred to herein as an "XMPP-CPIM gateway", may be established to interpret the protocols of one service and translate them into the protocols of the other service. We can visualize this relationship as follows:



This memo defines a mapping for use by a gateway that translates between XMPP and a non-XMPP protocol via the CPIM specifications. Such a gateway is not an intermediate hop on a network of non-XMPP servers (whose native formats may or may not be defined by the CPIM specifications), but a dedicated translator between XMPP and a non-XMPP protocol, where the CPIM specifications define the common formats into which the protocols are translated for purposes of interworking.

The mapping defined herein applies to instant messages and presence information that are not encrypted or signed for end-to-end security. For information about secure communications to or from an XMPP service through an XMPP-CPIM gateway, refer to [XMPP-E2E].

1.2. Terminology

This memo inherits vocabulary defined in [IMP-MODEL]. Terms such as CLOSED, INSTANT INBOX, INSTANT MESSAGE, OPEN, PRESENCE SERVICE, PRESENTITY, SUBSCRIPTION, and WATCHER are used in the same meaning as defined therein.

This memo also inherits vocabulary defined in [XMPP-CORE]. Terms such as ENTITY, NODE IDENTIFIER, DOMAIN IDENTIFIER, RESOURCE IDENTIFIER, MESSAGE STANZA, and PRESENCE STANZA are used in the same meaning as defined therein.

1.3. Conventions Used in this Document

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

2. Approach

XMPP and CPIM are distinctly foreign technologies. Therefore, care must be taken in mapping between XMPP and the abstract syntax defined by the CPIM specifications.

At root, XMPP is a data transport protocol for streaming XML elements (called "stanzas") between any two endpoints on the network; message and presence stanzas are two of the core data elements defined in XMPP and are often used to exchange instant messages and presence information between IM users (although the inherent extensibility of XML enables applications to use the general semantics of these stanza types for other purposes). XMPP is not based on [MIME]; instead, [XMPP-CORE] defines XML schemas for both message and presence stanzas (for example, the <body/> child of a message stanza contains XML character data that is usually intended to be read by a human user).

The CPIM specifications provide common formats for instant messaging and presence through two [MIME] content-types: "Message/CPIM" for messages ([MSGFMT]) and "application/pidf+xml" for presence ([PIDF]). The syntax of "Message/CPIM" objects is similar to but stricter than that defined in [RFC2822], and provides the ability to include arbitrary MIME media types [MIMETYPES]. By contrast, each "application/pidf+xml" object is a complete XML document whose structure is defined by an XML schema.

The approach taken herein is to specify mappings from XMPP elements and attributes to the headers and MIME formats defined by [MSGFMT] and [PIDF] in order to comply with the semantics defined by [CPIM] and [CPP]. Naturally, mappings in the opposite direction are provided as well.

3. Address Mapping

3.1. Overview

Address mapping may be required since the address formats used to identify XMPP entities (specified in [XMPP-CORE]) are different from those used to identify instant inboxes (the im: URI scheme specified in [CPIM]) and presentities (the pres: URI scheme specified in [CPP]). In particular, different characters are allowed in im: and pres: URIs than are allowed in XMPP addresses:

- o The following [US-ASCII] characters are allowed in im:/pres: URIs but not in XMPP addresses: #26; (&), #27; ('), and #2f; (/).
- o Many non-US-ASCII (specifically, UTF-8) characters are allowed in XMPP addresses but not allowed in im:/pres: URIs, since XMPP allows internationalized local-part addresses.

Note: In this document we discuss characters allowed in local-part addresses only (i.e., we have ruled the mapping of domain names as out of scope for the initial version of this document, since it is a matter for the Domain Name System and the translation of fully internationalized domain names).

3.2. XMPP to CPIM

The following is a high-level algorithm for mapping an XMPP address to an im: or pres: URI:

1. Split XMPP address into node identifier (local-part; mapping described in remaining steps), domain identifier (hostname; mapping is out of scope), and resource identifier (specifier for particular device or connection; discard this for cross-system interoperability)
2. Apply Nodeprep profile of [STRINGPREP] (as specified in [XMPP-CORE]) for canonicalization (OPTIONAL)
3. Translate #26; to &, #27; to ', and #2f; to / respectively
4. For each byte, if the byte is not in the set A-Za-z0-9!\$*.*?_~+= then change to %hexhex as described in Section 2.2.5 of [URL-GUIDE]

5. Combine resulting local-part with mapped hostname to form local@domain address
6. Prepend with 'im:' scheme (for XMPP <message/> stanzas) or 'pres:' scheme (for XMPP <presence/> stanzas)

3.3. CPIM to XMPP

The following is a high-level algorithm for mapping an im: or pres: URI to an XMPP address:

1. Remove URI scheme
2. Split at the first '@' character into local-part and hostname (mapping the latter is out of scope)
3. Translate %hexhex to equivalent octets as described in [Section 2.2.5](#) of [\[URL-GUIDE\]](#)
4. Treat result as a UTF-8 string
5. Translate & to #26;, ' to #27;, and / to #2f respectively
6. Apply Nodeprep profile of [\[STRINGPREP\]](#) (as specified in [\[XMPP-CORE\]](#)) for canonicalization (OPTIONAL)
7. Recombine local-part with mapped hostname to form local@domain address

4. Syntax Mapping of Instant Messages

This section describes how a gateway SHOULD map instant messages between an XMPP service and a non-XMPP service using a "Message/CPIM" object as the bearer of encapsulated text content in order to comply with the instant messaging semantics defined by [\[CPIM\]](#).

4.1. Message Syntax Mapping from XMPP to CPIM Specifications

This section defines the mapping of syntax primitives from XMPP message stanzas to "Message/CPIM" objects with encapsulated text content.

Note: As specified in [\[MIME\]](#), the default Content-type of a MIME object is "Content-type: text/plain; charset=us-ascii". Because XMPP uses the [\[UTF-8\]](#) character encoding exclusively, the encapsulated MIME object generated by an XMPP-CPIM gateway MUST set the

"Content-type" MUST be set to "text/plain" and the charset MUST be set to "utf-8".

4.1.1. From Address

The 'from' attribute of an XMPP message stanza maps to the 'From' header of a "Message/CPIM" object. In XMPP, the sender's server stamps or validates the "from" address and sets its value to the full <user@host/resource> negotiated between client and server during authentication and resource binding as defined in [XMPP-CORE]. Thus an XMPP-CPIM gateway will receive from the sender's XMPP server a message stanza containing a "from" address of the form <user@host/resource>. To map the 'from' attribute of an XMPP message stanza to the 'From' header of a "Message/CPIM" object, the gateway MUST remove the resource identifier, MUST append the "im:" Instant Messaging URI scheme to the front of the address, and MAY include a CPIM "Formal-name" for the sender (if known).

Example: From Address Mapping

XMPP 'from' attribute

```
<message from='juliet@example.com/balcony'>
...
</message>
```

CPIM 'From' header

```
From: Juliet Capulet <im:juliet@example.com>
```

4.1.2. To Address

The 'to' attribute of an XMPP message stanza maps to the 'To' header of a "Message/CPIM" object. In XMPP, the sender SHOULD include a 'to' attribute on a message stanza, and MUST include it if the message is intended for delivery to another user. Thus an XMPP-CPIM gateway will receive from the sender's XMPP server a message stanza containing a "to" address of the form <user@host> or <user@host/resource>. To map the 'to' attribute of an XMPP message stanza to the 'To' header of a "Message/CPIM" object, the gateway MUST remove the resource identifier (if included), MUST append the "im:" Instant Messaging URI scheme to the front of the address, and MAY include a CPIM "Formal-name" for the recipient (if known).

Example: To Address Mapping

XMPP 'to' attribute

```
<message to='romeo@example.net/orchard'>
  ...
</message>
```

CPIM 'To' header

```
To: Romeo Montague <im:romeo@example.net>
```

4.1.3. Stanza ID

An XMPP message stanza MAY possess an 'id' attribute, which is used by the sending application for the purpose of tracking stanzas and is not a globally-unique identifier such as is defined by the MIME Content-ID header. Because the XMPP 'id' attribute does not have the same meaning as the MIME Content-ID header, it SHOULD NOT be mapped to that header; however, if the 'id' is known to be unique (e.g., if it is generated to be unique by the XMPP server and that fact is known by the XMPP-CPIM gateway), then it SHOULD be so mapped.

4.1.4. Message Type

An XMPP message stanza MAY possess a 'type' attribute, which is used by the sending application to capture the conversational context of the message. There is no mapping of an XMPP 'type' attribute to a "Message/CPIM" header, common MIME features, or encapsulated text content. Therefore if an XMPP stanza received by an XMPP-CPIM gateway possesses a 'type' attribute, the gateway SHOULD ignore the value provided.

4.1.5. Message Thread

An XMPP message stanza MAY contain a <thread/> child element to specify the conversation thread in which the message is situated. There is no mapping of an XMPP <thread/> element to a "Message/CPIM" header, common MIME features, or encapsulated text content. Therefore if an XMPP message stanza received by an XMPP-CPIM gateway contains a <thread/> child element, the gateway SHOULD ignore the value provided.

4.1.6. Message Subject

An XMPP message stanza MAY include a <subject/> child element. If included, it maps to the 'Subject' header of a "Message/CPIM" object. To map the XMPP <subject/> element to the 'Subject' header of a "Message/CPIM" object, the gateway SHOULD simply map the XML character data of the XMPP <subject/> element to the value of the

'Subject' header. The <subject/> element MAY include an 'xml:lang' attribute specifying the language in which the subject is written. If an 'xml:lang' attribute is provided, it MUST be mapped by including ';lang=tag' after the header name and colon, where 'tag' is the value of the 'xml:lang' attribute.

Example: Subject Mapping

```
XMPP <subject/> element
  <subject>Hi!</subject>
  <subject xml:lang='cz'>Ahoj!</subject>
```

```
CPIM 'Subject' header
  Subject: Hi!
  Subject:;lang=cz Ahoj!
```

4.1.7. Message Body

The <body/> child element of an XMPP message stanza is used to provide the primary meaning of the message. The XML character data of the XMPP <body/> element maps to the encapsulated text message content.

Example: Message Body

```
XMPP message <body/>
  <message>
    <body>Wherefore art thou, Romeo?</body>
  </message>
```

```
Encapsulated MIME text content
  Content-type: text/plain; charset=utf-8
  Content-ID: <123456789@example.net>
```

Wherefore art thou, Romeo?

4.1.8. Message Extensions

As defined in [XMPP-CORE], an XMPP message stanza may contain "extended" content in any namespace in order to supplement or extend the semantics of the core message stanza. With the exception of extended information qualified by the 'urn:ietf:params:xml:ns:xmpp-e2e' namespace as defined in [XMPP-E2E], an XMPP-CPIM gateway SHOULD ignore such information and not pass it through the gateway to the intended recipient. No mapping for such information is defined.

4.1.9. Gateway-Generated CPIM Syntax

CPIM specifies the existence of "Message/CPIM" headers in addition to those described above, but there is no exact analogue for those headers in the core XMPP specifications. These include:

- o cc -- specifies the address of an entity that is to receive a "courtesy copy" of the message (i.e., a non-primary addressee)
- o DateTime -- specifies the datetime at which the message was sent
- o NS -- specifies the namespace of a feature extension
- o Require -- specifies mandatory-to-recognize features

An XMPP-CPIM gateway MAY independently generate such headers based on its own information (e.g., the datetime at which it received a message stanza from an XMPP entity) or based on data encoded in non-core XMPP extensions, but rules for doing so are out of scope for this memo.

4.2. Message Syntax Mapping from CPIM Specifications to XMPP

This section defines the mapping of syntax primitives from "Message/CPIM" objects with encapsualted text content to XMPP message stanzas.

4.2.1. From Address

The 'From' header of a "Message/CPIM" object maps to the 'from' attribute of an XMPP message stanza. To map the CPIM 'From' header to the XMPP 'from' attribute, the gateway MUST remove the "im:" Instant Messaging URI scheme from the front of the address and MUST remove the CPIM "Formal-name" (if provided).

Example: From Address Mapping

CPIM 'From' header

From: Romeo Montague <im:romeo@example.net>

XMPP 'from' attribute

```
<message from='romeo@example.net'>
  ...
</message>
```

4.2.2. To Address

The 'To' header of a "Message/CPIM" object maps to the 'to' attribute of an XMPP message stanza. To map the CPIM 'To' header to the XMPP 'to' attribute, the gateway MUST remove the "im:" Instant Messaging URI scheme from the front of the address and MUST remove the CPIM

"Formal-name" (if provided). If the gateway possesses knowledge of the resource identifier in use by the XMPP entity, the gateway MAY append the resource identifier to the address.

Example: To Address Mapping

CPIM 'To' header

To: Juliet Capulet <im:juliet@example.com>

XMPP 'to' attribute

<message to='juliet@example.com/balcony'>

...

</message>

4.2.3. Courtesy Copy

The core XMPP specification does not include syntax for specifying a "courtesy copy" (non-primary addressee) for a message stanza. Therefore, if an XMPP-CPIM gateway receives a "Message/CPIM" object that contains a 'cc' header, it SHOULD NOT pass the information contained in that header on to the XMPP recipient.

4.2.4. DateTime Header

The core XMPP specification does not include syntax for specifying the datetime at which a message stanza was sent. Therefore, if an XMPP-CPIM gateway receives a "Message/CPIM" object that contains a 'DateTime' header, it SHOULD NOT pass the information contained in that header on to the XMPP recipient.

4.2.5. Message Subject

The 'Subject' header of a "Message/CPIM" object maps to the <subject/> child element of an XMPP message stanza. To map the CPIM 'Subject' header to the XMPP <subject/> element, the gateway SHOULD simply map the value of the 'Subject' header to the XML character data of the XMPP <subject/> element. The 'Subject' header MAY specify the "lang" in which the subject is written. If "lang" information is provided, it MUST be mapped to the 'xml:lang' attribute of the <subject/> element, where the value of the 'xml:lang' attribute is the "tag" value supplied in the string ';lang=tag' included after the CPIM 'Subject' header name and colon.

Example: Subject Mapping

CPIM 'Subject' header

Subject: Hi!

Subject:;lang=cz Ahoj!

XMPP <subject/> element

<subject>Hi!</subject>

<subject xml:lang='cz'>Ahoj!</subject>

4.2.6. Header Extensions

"Message/CPIM" objects MAY include an optional 'NS' header to specify the namespace of a feature extension. An XMPP-CPIM gateway MUST NOT pass such headers through to the XMPP recipient, and no mapping for such headers is defined.

4.2.7. Require Header

"Message/CPIM" objects MAY include an optional 'Require' header to specify mandatory-to-recognize features. In general, such a header would be included by the non-XMPP sending application to (1) insist that the receiving application needs to understand functionality specified by a particular header or (2) indicate that some non-header semantics need to be implemented by the receiving application in order to understand the contents of the message (e.g., "Locale.MustRenderKanji"). Because the mandatory-to-recognize features would be required of the XMPP receiving application rather than the XMPP-CPIM gateway itself, the gateway cannot properly handle the 'Require' header without detailed knowledge about the capabilities of the XMPP receiving application. Therefore, it seems appropriate that the XMPP-CPIM gateway SHOULD return a warning or error to the non-XMPP sending application if it includes one or more 'Require' headers in a "Message/CPIM" object; the exact nature of the warning or error will depend on the nature of the non-XMPP technology used by the foreign system, and is not defined herein. Furthermore, any mapping of the 'Require' header into XMPP or an XMPP extension is left up to the implementation or to a future specification.

4.2.8. MIME Content-ID

XMPP does not include an element or attribute that captures a globally unique ID as is defined for the Content-ID MIME header as specified in [MIME]. If an XMPP-CPIM gateway receives a MIME object that includes a Content-ID, it MAY provide the Content-ID as the value of the message stanza's 'id' attribute, but this is OPTIONAL.

Example: Content-ID for Encapsulated Object

MIME header

Content-ID: <123456789@example.net>

XMPP 'id' attribute (OPTIONAL)

<message id='123456789@example.net'>

...

</message>

4.2.9. Message Body

If the Content-type of an encapsulated MIME object is "text/plain", then the encapsulated text message content maps to the XML character data of the <body/> child element of an XMPP message stanza.

Example: Message Body

Encapsulated MIME text content

Content-type: text/plain; charset=utf-8

Content-ID: <123456789@example.net>

Wherefore art thou?

XMPP message <body/>

<message id='123456789@example.net'>

<body>Wherefore art thou?</body>

</message>

If the Content-Type is not "text/plain", the XMPP-CPIM gateway MAY map the content to an XMPP extension but MUST NOT map it to the <body/> child of the XMPP message stanza, which is allowed to contain XML character data only. The only exception to this rule is a multi-part MIME object of the kind specified in [XMPP-E2E], which is to be mapped as described in that memo.

If the charset is "US-ASCII" or "UTF-8", the gateway MUST map the "Message/CPIM" object; otherwise it SHOULD NOT.

4.2.10. Gateway-Generated XMPP Syntax

XMPP specifies the existence of a 'type' attribute for XMPP message stanzas, which enables the sender to define the conversational context of the message. There is no exact analogue for this attribute in CPIM. An XMPP-CPIM gateway MAY independently generate the 'type' attribute based on its own information, but this is OPTIONAL and rules for doing so are out of scope for this memo.

5. Syntax Mapping of Presence Information

This section describes how a gateway SHOULD map presence information between an XMPP service and a non-XMPP service using a "Message/CPIM" object as the bearer of an encapsulated [PIDF] object in order to comply with the presence semantics defined by [CPP].

5.1. Presence Syntax Mapping from XMPP to CPIM Specifications

This section defines the mapping of syntax primitives from XMPP presence stanzas to "Message/CPIM" objects with encapsulated "application/pidf+xml" objects.

Note: As specified in [MIME], the default Content-type of a MIME object is "Content-type: text/plain; charset=us-ascii". Because XMPP uses the [UTF-8] character encoding exclusively and because PIDF specifies the "application/pidf+xml" MIME type, the encapsulated MIME object generated by an XMPP-CPIM gateway for presence information MUST set the 'Content-type' header for that object. The "Content-type" MUST be set to "application/pidf+xml" and the charset MUST be set to "utf-8".

5.1.1. From Address

The 'from' attribute of an XMPP presence stanza maps to the 'From' header of a "Message/CPIM" object. In XMPP, the sender's server stamps or validates the "from" address and sets its value to the <user@host/resource> negotiated between client and server during authenticating and resource binding as defined in [XMPP-CORE]. Thus an XMPP-CPIM gateway will receive from the sender's XMPP server a presence stanza containing a "from" address of the form <user@host/resource>. To map the 'from' attribute of an XMPP presence stanza to the 'From' header of a "Message/CPIM" object, the gateway MUST remove the resource identifier, MUST append the "im:" Instant Messaging URI scheme to the front of the address, and MAY include a CPIM "Formal-name" for the sender (if known).

Example: From Address Mapping

XMPP 'from' attribute

```
<presence from='juliet@example.com/balcony'>
  ...
</presence>
```

CPIM 'From' header

```
From: Juliet Capulet <im:juliet@example.com>
```

In addition, the 'from' attribute of an XMPP presence stanza maps to the 'entity' attribute of a PIDsF <presence/> root element. To map the XMPP 'from' attribute to the PIDsF 'entity' attribute, the gateway MUST remove the resource identifier and MUST append the "pres:" Instant Messaging URI scheme to the front of the address.

Example: From Address Mapping (PIDsF)

XMPP 'from' attribute

```
<presence from='juliet@example.com/balcony'>
...
</presence>
```

PIDsF 'entity' attribute

```
<presence entity='pres:juliet@example.com'>
...
</presence>
```

Finally, an XMPP-CPIM gateway SHOULD map the resource identifier of the XMPP address contained in the XMPP 'from' attribute to the 'id' attribute of the PIDsF <tuple/> child element.

Example: Resource Identifier Mapping

XMPP 'from' attribute

```
<presence from='juliet@example.com/balcony'>
...
</presence>
```

PIDsF 'id' for <tuple/>

```
<presence entity='pres:juliet@example.com'>
  <tuple id='balcony'>
    ...
  </tuple>
</presence>
```

5.1.2. To Address

The 'to' attribute of an XMPP presence stanza maps to the 'To' header of a "Message/CPIM" object. In XMPP, the sender MAY include a 'to' attribute on a presence stanza, and MUST include it if the presence stanza is intended for delivery directly to another user (presence stanzas intended for broadcasting are stamped with a 'to' address by the sender's server). Thus an XMPP-CPIM gateway will receive from the sender's XMPP server a presence stanza containing a "to" address of the form <user@host> or <user@host/resource>. To map the 'to' attribute of an XMPP presence stanza to the 'To' header of a "Message/CPIM" object, the gateway MUST remove the resource

identifier (if included), MUST append the "im:" Instant Messaging URI scheme to the front of the address, and MAY include a CPIM "Formal-name" for the recipient (if known).

Example: To Address Mapping

XMPP 'to' attribute

```
<presence to='romeo@example.net/orchard'>
  ...
</presence>
```

CPIM 'To' header

To: Romeo Montague <im:romeo@example.net>

5.1.3. Stanza ID

An XMPP presence stanza MAY possess an 'id' attribute, which is used by the sending application for the purpose of tracking stanzas and is not a globally-unique identifier such as is defined by the MIME Content-ID header. Because the XMPP 'id' attribute does not have the same meaning as the MIME Content-ID header, it SHOULD NOT be mapped to that header; however, if the 'id' is known to be unique (e.g., if it is generated to be unique by the XMPP server and that fact is known by the XMPP-CPIM gateway), then it SHOULD be so mapped.

5.1.4. Presence Type

An XMPP presence stanza MAY possess a 'type' attribute. If no 'type' attribute is included, the presence stanza indicates that the sender is available; this state maps to the PIDF basic presence type of OPEN. If the 'type' attribute has a value of "unavailable", the presence stanza indicates that the sender is no longer available; this state maps to the PIDF basic presence type of CLOSED. Thus both the absence of a 'type' attribute and a 'type' attribute set to a value of "unavailable" correspond to the [CPP] "notify operation". All other presence types are used to manage presence subscriptions or probe for current presence; mappings for these other presence types are defined under XMPP-CPIM Gateway as Presence Service ([Section 6](#)).

Example: Available Presence

XMPP available presence

```
<presence from='juliet@example.com/balcony' />
```

PIDF basic presence (OPEN)

```
<?xml version='1.0' encoding='UTF-8'?>
<presence xmlns='urn:ietf:params:xml:ns:pidf'
  entity='pres:juliet@example.com'>
```

```

    <tuple id='balcony'>
      <status>
        <basic>open</basic>
      </status>
    </tuple>
  </presence>

```

Example: Unavailable Presence

XMPP unavailable presence

```

<presence from='juliet@example.com/balcony' type='unavailable'/>

```

PIDF basic presence (CLOSED)

```

<?xml version='1.0' encoding='UTF-8'?>
<presence xmlns='urn:ietf:params:xml:ns:pidf'
  entity='pres:romeo@example.net'>
  <tuple id='balcony'>
    <status>
      <basic>closed</basic>
    </status>
  </tuple>
</presence>

```

5.1.5. Show Element

The <show/> child element of an XMPP presence stanza provides additional information about the sender's availability. The XML character data of the XMPP <show/> element maps to extended <status/> content in PIDF. The defined values of the <show/> element are 'away', 'chat', 'dnd', and 'xa'; as soon as values are specified for extended status states in the 'urn:ietf:params:xml:ns:pidf:im' namespace, the XMPP values will be mapped to the PIDF values.

Example: Show Element

XMPP <show/> element

```

<presence from='juliet@example.com/balcony'>
  <show>away</show>
</presence>

```

PIDF extended presence information

```

<?xml version='1.0' encoding='UTF-8'?>
<presence xmlns='urn:ietf:params:xml:ns:pidf'
  xmlns:im='urn:ietf:params:xml:ns:pidf:im'
  entity='pres:juliet@example.com'>
  <tuple id='balcony'>
    <status>
      <basic>open</basic>

```



```

    <im:im>away</im:im>
  </status>
</tuple>
</presence>

```

5.1.6. Status Element

The <status/> child element of an XMPP presence stanza provides a user-defined, natural-language description of the sender's detailed availability state. The XMPP <status/> element maps to the PIDF <note/> child of the PIDF <tuple/> element.

Example: Status Element

XMPP <status/> element

```

<presence from='juliet@example.com/balcony'>
  <show>away</show>
  <status>retired to the chamber</status>
</presence>

```

PIDF <note/> element

```

<?xml version='1.0' encoding='UTF-8'?>
<presence xmlns='urn:ietf:params:xml:ns:pidf'
  xmlns:im='urn:ietf:params:xml:ns:pidf:im'
  entity='pres:juliet@example.com'>
  <tuple id='balcony'>
    <status>
      <basic>open</basic>
      <im:im>away</im:im>
    </status>
    <note>retired to the chamber</note>
  </tuple>
</presence>

```

5.1.7. Presence Priority

An XMPP presence stanza MAY contain a <priority/> child element whose value is an integer between -128 and +127. The value of this element MAY be mapped to the 'priority' attribute of the <contact/> child of the PIDF <tuple/> element. If the value of the XMPP <priority/> element is negative, an XMPP-CPIM gateway MUST NOT map the value. The range of allowable values for the PIDF 'priority' attribute is any decimal number from zero to one inclusive, with a maximum of three decimal places. If an XMPP-CPIM gateway maps these values, it SHOULD treat XMPP <priority>0</priority> as PIDF priority='0' and XMPP <priority>127</priority> as PIDF priority='1', mapping intermediate values appropriately so that they are unique (e.g., XMPP priority 1 to PIDF priority 0.007, XMPP priority 2 to PIDF priority 0.015, and

so on up through mapping XMPP priority 126 to PIDF priority 0.992; note that this is an example only, and that the exact mapping shall be determined by the XMPP-CPIM gateway).

Example: Presence Priority

XMPP <status/> element

```
<presence from='juliet@example.com/balcony'>
  <priority>13</priority>
</presence>
```

PIDF <note/> element

```
<?xml version='1.0' encoding='UTF-8'?>
<presence xmlns='urn:ietf:params:xml:ns:pidf'
  entity='pres:juliet@example.com'>
  <tuple id='balcony'>
    ...
    <contact priority='0.102'>im:juliet@example.com</contact>
  </tuple>
</presence>
```

5.1.8. Presence Extensions

As defined in [XMPP-CORE], an XMPP presence stanza may contain "extended" content in any namespace in order to supplement or extend the semantics of the core presence stanza. With the exception of extended information qualified by the 'urn:ietf:params:xml:ns:xmpp-e2e' namespace as defined in [XMPP-E2E], an XMPP-CPIM gateway SHOULD ignore such information and not pass it through the gateway to the intended recipient. No mapping for such information is defined.

5.1.9. Gateway-Generated CPIM and PIDF Syntax

5.1.9.1. CPIM Message Headers

CPIM specifies the existence of "Message/CPIM" headers in addition to those described above, but there is no exact analogue for those headers in the core XMPP specifications. These include:

- o cc -- specifies the address of an entity that is to receive a "courtesy copy" of the presence information (i.e., a non-primary addressee)
- o DateTime -- specifies the datetime at which the presence information was sent
- o NS -- specifies the namespace of a feature extension

- o Subject -- specifies the subject or topic of the encapsulated "Message/CPIM" object
- o Require -- specifies mandatory-to-recognize features

An XMPP-CPIM gateway MAY independently generate such headers based on its own information (e.g., the datetime at which it received a presence stanza from an XMPP entity) or based on data encoded in non-core XMPP extensions, but rules for doing so are out of scope for this memo.

5.1.9.2. PIDF Elements

PIDF specifies the existence of XML elements in addition to those described above, but there is no exact analogue for those XML elements in the core XMPP specifications. These include:

- o <contact/> -- specifies an address (e.g., an im:, tel:, or mailto: URI) at which one may communicate with the presentity; an XMPP-CPIM gateway MAY include this element, in which case it SHOULD set its value to the <user@host> of the XMPP sender, prepended by the "im:" Instant Messaging URI scheme.
- o <timestamp/> -- specifies the datetime at which the presence information was sent; an XMPP-CPIM gateway MAY independently generate this element based on its own information (e.g., the datetime at which it received the presence stanza from an XMPP entity) or based on data encoded in non-core XMPP extensions, but rules for doing so are out of scope for this memo.

5.2. Presence Syntax Mapping from CPIM Specifications to XMPP

This section defines the mapping of syntax primitives from "Message/CPIM" objects with encapsulated "application/pidf+xml" objects to XMPP presence stanzas.

Note: An XMPP-CPIM gateway MUST NOT map to an XMPP presence stanza a "Message/CPIM" object whose encapsulated MIME object has a Content-type other than "application/pidf+xml" (with the exception of multi-part MIME objects as specified in [XMPP-E2E]).

5.2.1. From Address

The 'From' header of a "Message/CPIM" object maps to the <user@host> portion of the 'from' attribute of an XMPP presence stanza, and the 'id' attribute of the PIDF <tuple/> child element maps to the resource identifier portion XMPP 'from' attribute. Therefore, to map the CPIM and PIDF information to the XMPP 'from' attribute, the

gateway MUST remove the "im:" Instant Messaging URI scheme from the front of the address and MUST remove the CPIM "Formal-name" (if provided) in order to generate the <user@host> portion of the XMPP 'from' attribute, then add a '/' character followed by the value of the PIDF <tuple/> element's 'id' attribute.

Example: From Address Mapping

CPIM 'From' header

From: Romeo Montague <im:romeo@example.net>

XMPP 'from' attribute

```
<presence from='romeo@example.net'>
...
</presence>
```

Example: Resource Identifier Mapping

XMPP 'from' attribute

```
<presence from='juliet@example.com/balcony'>
...
</presence>
```

PIDF 'id' for <tuple/>

```
<presence entity='pres:juliet@example.com'>
  <tuple id='balcony'>
    ...
  </tuple>
</presence>
```

5.2.2. To Address

The 'To' header of a "Message/CPIM" object maps to the 'to' attribute of an XMPP presence stanza. To map the CPIM 'To' header to the XMPP 'to' attribute, the gateway MUST remove the "im:" Instant Messaging URI scheme from the front of the address and MUST remove the CPIM "Formal-name" (if provided). If the gateway possesses knowledge of the resource identifier in use by the XMPP entity, the gateway MAY append the resource identifier to the address.

Example: To Address Mapping

CPIM 'To' header

To: Juliet Capulet <im:juliet@example.com>

XMPP 'to' attribute

```
<presence to='juliet@example.com/balcony'>
  ...
</presence>
```

5.2.3. Courtesy Copy

The core XMPP specification does not include syntax for specifying a "courtesy copy" (non-primary addressee) for a presence stanza. Therefore, if an XMPP-CPIM gateway receives a "Message/CPIM" object with encapsulated PIDF object that contains a 'cc' header, it SHOULD NOT pass the information contained in that header on to the XMPP recipient.

5.2.4. DateTime Header

The core XMPP specification does not include syntax for specifying the datetime at which a presence stanza was sent. Therefore, if an XMPP-CPIM gateway receives a "Message/CPIM" object with encapsulated PIDF object that contains a 'DateTime' header, it SHOULD NOT pass the information contained in that header on to the XMPP recipient.

5.2.5. Subject Header

An XMPP presence stanza contains no information that can be mapped to the 'Subject' header of a "Message/CPIM" object. Therefore, if an XMPP-CPIM gateway receives a "Message/CPIM" object with encapsulated PIDF object that contains a 'Subject' header, it SHOULD NOT pass the information contained in that header on to the XMPP recipient.

5.2.6. Header Extensions

"Message/CPIM" objects MAY include an optional 'NS' header to specify the namespace of a feature extension. An XMPP-CPIM gateway MUST NOT pass such headers through to the XMPP recipient, and no mapping for such headers is defined.

5.2.7. Require Header

"Message/CPIM" objects MAY include an optional 'Require' header to specify mandatory-to-recognize features. An XMPP-CPIM gateway MUST NOT pass such headers through to the XMPP recipient, and no mapping for such headers is defined.

5.2.8. MIME Content-ID

XMPP does not include an element or attribute that captures a globally unique ID as is defined for the Content-ID MIME header as specified in [MIME]. If an XMPP-CPIM gateway receives a MIME object that includes a Content-ID, it MAY provide the Content-ID as the value of the presence stanza's 'id' attribute, but this is OPTIONAL.

Example: Content-ID for Encapsulated Object

MIME header

Content-ID: <123456789@example.net>

XMPP 'id' attribute (OPTIONAL)

```
<presence id='123456789@example.net'>
  ...
</presence>
```

5.2.9. Basic Presence Status

The basic presence status types defined in PIDF are OPEN and CLOSED. The PIDF basic presence status of OPEN maps to an XMPP presence stanza that possesses no 'type' attribute (indicating default availability). The PIDF basic presence status of CLOSED maps to an XMPP presence stanza that possesses a 'type' attribute with a value of "unavailable".

Example: OPEN Presence

PIDF basic presence (OPEN)

```
<?xml version='1.0' encoding='UTF-8'?>
<presence xmlns='urn:ietf:params:xml:ns:pidf'
  entity='pres:romeo@example.net'>
  <tuple id='orchard'>
    <status>
      <basic>open</basic>
    </status>
  </tuple>
</presence>
```

XMPP available presence

```
<presence from='romeo@example.net/orchard' />
```

Example: CLOSED Presence

PIDF basic presence (CLOSED)

```
<?xml version='1.0' encoding='UTF-8'?>
<presence xmlns='urn:ietf:params:xml:ns:pidf'
  entity='pres:romeo@example.net'>
  <tuple id='orchard'>
    <status>
      <basic>closed</basic>
    </status>
  </tuple>
</presence>
```

XMPP unavailable presence

```
<presence from='romeo@example.net/orchard'
  type='unavailable' />
```

5.2.10. Extended Status Information

PIDF documents may contain extended <status/> content. As of this writing there are no pre-defined extended status states that can be mapped to the defined values of the XMPP <show/> element ('away', 'chat', 'dnd', and 'xa'). Once PIDF extensions for such extended status states are defined within the Internet Standards Process, a gateway SHOULD map those extensions; however, any such mapping is out of scope for this memo, since the relevant PIDF extensions have not yet been defined.

Example: Extended Status Information (provisional)

PIDF extended presence information

```
<?xml version='1.0' encoding='UTF-8'?>
<presence xmlns='urn:ietf:params:xml:ns:pidf'
  xmlns:im='urn:ietf:params:xml:ns:pidf:im'
  entity='pres:romeo@example.net'>
  <tuple id='orchard'>
    <status>
      <basic>open</basic>
      <im:im>busy</im:im>
    </status>
  </tuple>
</presence>
```

XMPP <show/> element

```
<presence from='romeo@example.net/orchard'>
  <show>dnd</show>
</presence>
```

5.2.11. Note Element

A PIDF <tuple/> element may contain a <note/> child that provides a user-defined, natural-language description of the sender's detailed availability state. The PIDF <note/> element maps to the XMPP <status/> element.

Example: Note Element

PIDF <note/> element

```
<?xml version='1.0' encoding='UTF-8'?>
<presence xmlns='urn:ietf:params:xml:ns:pidf'
          xmlns:im='urn:ietf:params:xml:ns:pidf:im'
          entity='pres:romeo@example.net'>
  <tuple id='orchard'>
    <status>
      <basic>open</basic>
      <im:im>busy</im:im>
    </status>
    <note>Wooin Juliet</note>
  </tuple>
</presence>
```

XMPP <status/> element

```
<presence from='romeo@example.net/orchard'>
  <show>dnd</show>
  <status>Wooin Juliet</status>
</presence>
```

A PIDF document with zero tuples MAY contain one or more <note/> elements as direct children of the PIDF <presence/> element. There is no mapping of such a PIDF document to an XMPP presence stanza; an entity on the non-XMPP side of an XMPP-CPIM gateway SHOULD NOT send such a PIDF document to an XMPP recipient if possible, and an XMPP-CPIM gateway MUST NOT map such a PIDF document to an XMPP presence stanza (see Zero Resources ([Section 6.3.2](#))).

5.2.12. Contact Element

A PIDF document may contain a <contact/> element specifying the URI of an address at which the principal can be contacted (e.g., an im:, tel:, or mailto: URI). The core XMPP specification does not include syntax for specifying the URI of a contact address, since the contact address is implicit in the 'from' attribute of the XMPP presence stanza. Therefore, if an XMPP-CPIM gateway receives a "Message/CPIM" object with encapsulated PIDF object that contains a <contact/>

element, it SHOULD NOT pass the XML character data of the <contact/> element on to the XMPP recipient. (However, see Inclusion of Complete PIDF Document ([Section 5.2.15](#)) below.)

Example: PIDF Contact Element

PIDF <contact/> element

```
<?xml version='1.0' encoding='UTF-8'?>
<presence xmlns='urn:ietf:params:xml:ns:pidf'
  entity='pres:romeo@example.net'>
  <tuple id='orchard'>
    ...
    <contact>im:romeo@example.net</contact>
  </tuple>
</presence>
```

XMPP presence stanza

```
<presence from='romeo@example.net/orchard' />
```

5.2.13. Presence Priority

The <contact/> child of the PIDF <tuple/> element MAY possess a 'priority' attribute whose value is a decimal number between zero and one (with a maximum of three decimal places). The value of this attribute MAY be mapped to the <priority/> child element of an XMPP presence stanza. An XMPP-CPIM gateway MUST NOT map PIDF priority values to negative values of the XMPP <priority/> element. If an XMPP-CPIM gateway maps these values, it SHOULD treat PIDF priority='0' as XMPP <priority>0</priority> and PIDF priority='1' as <priority>127</priority>, mapping intermediate values appropriately so that they are unique (e.g., PIDF priorities between 0.001 and 0.007 to XMPP priority 1, PIDF priorities between 0.008 and 0.015 to XMPP priority 2, and so on up through mapping PIDF priorities between 0.992 and 0.999 to XMPP priority 126; note that this is an example only, and that the exact mapping shall be determined by the XMPP-CPIM gateway).

5.2.14. Timestamp Element

The core XMPP specification does not include syntax for specifying the datetime or timestamp at which a presence stanza was sent. Therefore, if an XMPP-CPIM gateway receives a "Message/CPIM" object with encapsulated PIDF object that contains a <timestamp/> element, it SHOULD NOT pass the XML character data of the <timestamp/> element on to the XMPP recipient.

5.2.15. Inclusion of Complete PIDF Document

Certain PIDF elements do not map to XMPP presence stanza syntax (e.g., the XML character data of the <contact/> element). However, an XMPP client may be able to handle such information by parsing a native PIDF document. To make this possible, an XMPP-CPIM gateway MAY include the complete PIDF document as a child element of the presence stanza, as described in [XMPP-PIDF]. If an XMPP client does not understand this extended data, it naturally MUST ignore it.

6. XMPP-CPIM Gateway as Presence Service

[CPP] defines semantics for an abstract presence service. An XMPP-CPIM gateway MAY function as such a presence service, and if so an XMPP entity can use defined XMPP syntax to interact with the gateway's presence service. Because [PIDF] does not specify syntax for semantic operations such as subscribe, this section defines only the XMPP interactions with the presence service offered by an XMPP-CPIM gateway, not the translation of such XMPP syntax into PIDF. (Note: Detailed information about XMPP presence services can be found in [XMPP-IM]; as much as possible, an XMPP-CPIM gateway SHOULD implement the syntax, semantics, and server business rules defined therein.)

6.1. Requesting a Subscription

If an XMPP entity wants to subscribe to the presence information of a non-XMPP presentity through an XMPP-CPIM gateway, it MUST send a presence stanza of type "subscribe" to the target presentity. The syntax mapping is as follows:

- o The XMPP 'from' attribute (user@host) MUST be mapped to the CPP "watcher parameter" field (pres:user@host). The XMPP-CPIM gateway MUST append the "pres:" Presence URI scheme to the front of the address.
- o The XMPP 'to' attribute (user@host) MUST be mapped to the CPP "target parameter" field (pres:user@host). The XMPP-CPIM gateway MUST append the "pres:" Presence URI scheme to the front of the address.
- o There is no XMPP mapping for the CPP "duration parameter", since XMPP subscriptions are active until they have been explicitly "unsubscribed".
- o The XMPP 'id' attribute SHOULD be mapped to the CPP "TransID" field.

If the target presentity approves the subscription request (through whatever protocol it uses to interact with the gateway), the XMPP-CPIM gateway MUST return a presence stanza of type "subscribed" to the XMPP entity and notify the XMPP entity of the target's current available presence. Thereafter, until the subscription is cancelled, the gateway MUST notify the subscribing XMPP entity every time the target's presence information changes.

If the target presentity denies the subscription request, the XMPP-CPIM gateway MUST return a presence stanza of type "unsubscribed" to the XMPP entity and MUST NOT invoke the notify operation.

In addition to the approval and denial cases, one of the following exceptions may occur:

- o The target parameter (XMPP "to" address) does not refer to a valid presentity; if this exception occurs, the XMPP-CPIM gateway MUST return an <item-not-found/> stanza error to the XMPP entity.
- o Access control rules do not permit the entity to subscribe to the target; if this exception occurs, the XMPP-CPIM gateway MUST return a <forbidden/> stanza error to the XMPP entity.
- o There exists a pre-existing subscription or in-progress subscribe operation between the XMPP entity and the target presentity; if this exception occurs, the XMPP-CPIM gateway SHOULD return a <conflict/> stanza error to the XMPP entity.

XMPP services assume that a subscription is active until it is explicitly terminated. However, non-XMPP services may implement subscriptions of limited duration, which must be periodically refreshed in order to mimic the permanence of XMPP subscriptions. Therefore, an XMPP-to-CPIM gateway may need to send such refreshes to the non-XMPP entity on behalf of the XMPP entity to that the subscription does not expire. Whether such refreshes are necessary depends on the native protocol implemented by the CPIM-aware non-XMPP service to which the gateway is translating.

6.2. Receiving a Subscription Request

If a non-XMPP presentity wants to subscribe to the presence information of an XMPP entity through an XMPP-CPIM gateway, it MUST use whatever protocol it uses to interact with the gateway in order to request the subscription; subject to local access rules, the gateway MUST then send a presence stanza of type "subscribe" to the XMPP entity from the non-XMPP watcher. The syntax mapping is as follows:

- o The CPP "watcher parameter" field (pres:user@host) MUST be mapped to the XMPP 'from' attribute (user@host). The XMPP-CPIM gateway MUST remove the "pres:" Presence URI scheme from the front of the address.
- o The CPP "target parameter" field (pres:user@host) MUST be mapped to the XMPP 'to' attribute (user@host). The XMPP-CPIM gateway MUST remove the "pres:" Presence URI scheme from the front of the address.
- o There is no XMPP mapping for the CPP "duration parameter", since XMPP subscriptions are active until they have been explicitly "unsubscribed".
- o The CPP "TransID" field SHOULD be mapped to the XMPP 'id' attribute.

If the target XMPP entity approves the subscription request, it MUST send a presence stanza of type "subscribed" to the watcher presentity. The XMPP-CPIM gateway MUST then notify the watcher presentity of the target XMPP entity's current available presence. Thereafter, until the subscription is cancelled, the gateway MUST notify the watcher presentity every time the target's presence information changes.

If the target XMPP entity denies the subscription request, it MUST send a presence stanza of type "unsubscribed" to the watcher presentity. The XMPP-CPIM gateway MUST NOT invoke the notify operation.

In addition to the approval and denial cases, one of the following exceptions MAY occur:

- o The target parameter (XMPP "to" address) does not refer to a valid XMPP entity
- o Access control rules do not permit the watcher presentity to subscribe to the target XMPP entity
- o There exists a pre-existing subscription or in-progress subscribe operation between the watcher presentity and the target XMPP entity

If any of these exceptions occurs, the XMPP-CPIM gateway MUST inform the watcher presentity of failure.

XMPP services assume that a subscription is active until it is explicitly terminated. With the exception of handling duration parameters whose value is zero, handling duration parameters will be highly dependent on the implementation and requirements of the XMPP-CPIM gateway. Since there are no explicit requirements for supporting a "duration parameter" specified in either [IMP-MODEL] or [IMP-REQS], duration parameter mapping is a local issue that falls outside the scope of this memo. However, an XMPP-CPIM gateway MAY keep track of the duration parameter if received from an entity on the non-XMPP service and delete the subscription after that duration parameter expires.

6.3. The Notify Operation

An XMPP-CPIM gateway invokes the CPP "notify operation" whenever the presence information associated with an XMPP entity or CPP presentity changes and there are subscribers to that information on the other side of the gateway. The syntax mapping for presence information related to a notify operation is defined under Mapping for Presence (Section 5).

6.3.1. Multiple Resources

Semantically, PIDs contains the notion of multiple presence "tuples". Normally, a PIDs document will contain at least one tuple but MAY contain more than one tuple (or zero tuples, for which see next section). In the terminology of XMPP, each tuple would map to presence information for a separate resource. However, XMPP does not include the ability to send presence information about more than one resource at a time, since the resource that generates the presence information is contained in the 'from' address of a presence stanza. Therefore, an XMPP-CPIM gateway that acts as a presence service SHOULD split a PIDs document that contains multiple tuples into multiple XMPP presence stanzas, and SHOULD generate only one PIDs document (with multiple tuples) if an XMPP user currently has multiple connected resources.

In the interest of not multiplying XMPP stanzas beyond necessity, an XMPP-CPIM gateway SHOULD generate an XMPP presence stanza only if the presence information contained in a PIDs tuple communicates a change in the availability status of the device or application associated with that tuple ID.

In the interest of complying with the PIDs recommendation to provide information about multiple "resources" in multiple tuples rather than in multiple PIDs documents, an XMPP-CPIM gateway SHOULD include

information about all of an XMPP user's resources in one PIDF document (with one tuple for each resource), even if the availability status of only one resource has changed.

6.3.2. Zero Resources

A PIDF document may contain zero tuples. For example:

PIDF Document with Zero Tuples

```
<presence entity='pres:juliet@example.com'
  xmlns='urn:ietf:params:xml:ns:pidf' />
```

Because (1) the 'entity' attribute of a PIDF <presence/> element maps to the <user@host> portion of an XMPP address and (2) the 'id' attribute of a PIDF <tuple/> element maps to the resource identifier portion of an XMPP address, a PIDF document that contains zero tuples would provide presence information about a <user@host> rather than a <user@host/resource> when mapped to XMPP. Although the notion of presence notifications about a mere user rather than one of the user's resources is nearly meaningless in the XMPP context, an XMPP-CPIM gateway SHOULD map a PIDF document with zero tuples to an XMPP presence stanza whose 'from' address is the user@host of the non-XMPP entity. However, an XMPP-CPIM gateway MUST NOT generate a PIDF document with zero <tuple/> children when receiving a presence stanza from an XMPP entity (i.e., all PIDF documents communicated by the gateway to a non-XMPP service MUST contain at least one <tuple/> element).

6.4. Unsubscribing

If an XMPP entity wants to unsubscribe from the presence of a non-XMPP presentity through an XMPP-CPIM gateway, it MUST send a presence stanza of type "unsubscribe" to the target presentity. The syntax mapping is as follows:

- o The XMPP 'from' attribute (user@host) MUST be mapped to the CPP "watcher parameter" field (pres:user@host). The XMPP-CPIM gateway MUST append the "pres:" Presence URI scheme to the front of the address.
- o The XMPP 'to' attribute (user@host) MUST be mapped to the CPP "target parameter" field (pres:user@host). The XMPP-CPIM gateway MUST append the "pres:" Presence URI scheme to the front of the address.
- o The CPP "duration parameter" MUST be set to zero.

- o The XMPP 'id' attribute SHOULD be mapped to the CPP "TransID" field.

If the target parameter (XMPP "to" address) does not refer to a valid presentity, the XMPP-CPIM gateway MUST return an <item-not-found/> stanza error to the XMPP entity.

Upon receiving the presence stanza of type "unsubscribe" from the XMPP entity, the XMPP-CPIM gateway MUST NOT send further presence notifications to the XMPP entity.

6.5. Cancelling a Subscription

If an XMPP entity wants to cancel a non-XMPP presentity's subscription to the entity's presence through an XMPP-CPIM gateway, it MUST send a presence stanza of type "unsubscribed" to the target presentity. The syntax mapping is as follows:

- o The XMPP 'from' attribute (user@host) MUST be mapped to the CPP "watcher parameter" field (pres:user@host). The XMPP-CPIM gateway MUST add the "pres:" Presence URI scheme to the front of the address.
- o The XMPP 'to' attribute (user@host) MUST be mapped to the CPP "target parameter" field (pres:user@host). The XMPP-CPIM gateway MUST add the "pres:" Presence URI scheme to the front of the address.
- o The CPP "duration parameter" MUST be set to zero.
- o The XMPP 'id' attribute SHOULD be mapped to the CPP "TransID" field.

Upon receiving the presence stanza of type "unsubscribed" from the XMPP entity, the XMPP-CPIM gateway MUST NOT send further presence notifications to the watcher presentity.

7. Security Considerations

Detailed security considerations for instant messaging and presence protocols are given in [IMP-REQS], specifically in Sections 5.1 through 5.4.

This document specifies methods for exchanging instant messages and presence information through a gateway that implements [CPIM] and [CPP]. Such a gateway MUST be compliant with the minimum security requirements of the instant messaging and presence protocols with which it interfaces. The introduction of gateways to the security model of instant messaging and presence in RFC 2779 also introduces

some new risks. In particular, end-to-end security properties (especially confidentiality and integrity) between instant messaging and presence user agents that interface through an XMPP-CPIM gateway can be provided only if common formats are supported; these formats are specified fully in [XMPP-E2E].

8. References

8.1. Normative References

- [CPIM] Peterson, J., "Common Profile for Instant Messaging (CPIM)", [RFC 3860](#), August 2004.
- [CPP] Peterson, J., "Common Profile for Presence (CPP)", [RFC 3859](#), August 2004.
- [IMP-MODEL] Day, M., Rosenberg, J., and H. Sugano, "A Model for Presence and Instant Messaging", [RFC 2778](#), February 2000.
- [IMP-REQS] Day, M., Aggarwal, S., Mohr, G., and J. Vincent, "Instant Messaging / Presence Protocol Requirements", [RFC 2779](#), February 2000.
- [MIME] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", [RFC 2045](#), November 1996.
- [MSGFMT] Klyne, G. and D. Atkins, "Common Presence and Instant Messaging (CPIM): Message Format", [RFC 3862](#), August 2004.
- [PIDF] Sugano, H., Fujimoto, S., Klyne, G., Bateman, A., Carr, W., and J. Peterson, "Presence Information Data Format (PIDF)", [RFC 3863](#), August 2004.
- [STRINGPREP] Hoffman, P. and M. Blanchet, "Preparation of Internationalized Strings (stringprep)", [RFC 3454](#), December 2002.
- [TERMS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [URL-GUIDE] Masinter, L., Alvestrand, H., Zigmond, D., and R. Petke, "Guidelines for new URL Schemes", [RFC 2718](#), November 1999.

- [US-ASCII] Cerf, V., "ASCII format for network interchange", [RFC 20](#), October 1969.
- [UTF-8] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, [RFC 3629](#), November 2003.
- [XMPP-CORE] Saint-Andre, P., Ed., "Extensible Messaging and Presence Protocol (XMPP): Core", [RFC 3920](#), October 2004.
- [XMPP-E2E] Saint-Andre, P., Ed., "End-to-End Signing and Object Encryption in the Extensible Messaging and Presence Protocol (XMPP)", [RFC 3923](#), October 2004.
- [XMPP-IM] Saint-Andre (ed.), P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", [RFC 3921](#), October 2004.

8.2. Informative References

- [RFC2822] Resnick, P., Ed., "Internet Message Format", [RFC 2822](#), April 2001.
- [MIMETYPES] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", [RFC 2046](#), November 1996.
- [XMPP-PIDF] Saint-Andre, P., "Transporting Presence Information Data/Format (PIDF) over the Extensible Messaging and Presence Protocol (XMPP)", Work in Progress, February 2004.

Author's Address

Peter Saint-Andre
Jabber Software Foundation

EMail: stpeter@jabber.org

Full Copyright Statement

Copyright (C) The Internet Society (2004).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/S HE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the IETF's procedures with respect to rights in IETF Documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.