Mapping the Extensible Messaging and Presence Protocol (XMPP) to Common Presence and Instant Messaging (CPIM)
draft-ietf-xmpp-cpim-03

Status of this Memo

Copyright Notice

Abstract

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

Table of Contents

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". In order to refer to a gateway between an Extensible Messaging and Presence Protocol (XMPP) service and a non-XMPP service where the common format is defined by the CPIM specifications, the term "XMPP-CPIM Gateway" is used herein.)

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 gateways 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:

```
  +-------------+        +-------------+        +------------+
  |             |        |             |        |            |
  |    XMPP     |        |  XMPP-CPIM  |        |  Non-XMPP  |
  |   Service   | <----> |   Gateway   | <----> |  Service   |
  |             |        |             |        |            |
  +-------------+        +-------------+        +------------+
```

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].

1.4 Discussion Venue

The authors welcome discussion and comments related to the topics presented in this document. The preferred forum is the <xmppwg@jabber.org> mailing list, for which archives and subscription information are available at <http://www.jabber.org/cgi-bin/mailman/listinfo/xmppwg/>.

1.5 Intellectual Property Notice

This document is in full compliance with all provisions of Section 10 of RFC 2026. Parts of this specification use the term "jabber" for identifying namespaces and other protocol syntax. Jabber[tm] is a registered trademark of Jabber, Inc. Jabber, Inc. grants permission to the IETF for use of the Jabber trademark in association with this specification and its successors, if any.

1.6 Contributors

Tony Bamonti contributed to the first version of this memo.

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. 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].

3.1 Identification of Instant Inboxes

There is a one-to-one relationship between an XMPP entity and a CPIM instant inbox when the address of the XMPP entity contains only a node identifier and domain identifier, and the node identifier uniquely corresponds to an IM user who possesses an account on an XMPP server. However, the syntax for addressing an instant inbox is specified as including the 'im:' URI scheme as defined in [CPIM], whereas an XMPP address does not include that scheme, so any mapping between an instant inbox address and an XMPP address must add or remove the 'im:' URI scheme as appropriate.

3.2 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.

3.2.1 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>
```

3.2.2 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>
```

3.2.3 CPIM Courtesy Copy

The core XMPP specification does not include syntax for specifying a "courtesy copy" (non-primary addressee) for a message stanza. Therefore, an XMPP-CPIM gateway SHOULD NOT generate the 'cc' header of a "Message/CPIM" object.

3.2.4 XMPP Stanza ID

An XMPP message stanza MAY possess an 'id' attribute, which is used by the sending application for the purpose of tracking stanzas. There is no mapping of an XMPP 'id' 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 an 'id' attribute, the gateway SHOULD ignore the value provided.

3.2.5 XMPP 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.

3.2.6 XMPP 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.

3.2.7 CPIM DateTime Header

The core XMPP specification does not include syntax for specifying the datetime at which a message stanza was sent. However, an XMPP-CPIM gateway MAY include a 'DateTime' header in the "Message/CPIM" object it generates, the value of which SHOULD be the datetime at which the message stanza was received for processing by the gateway.

3.2.8 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!
```

3.2.9 CPIM Header Extensions

A "Message/CPIM" object MAY include an optional 'NS' header to specify the namespace of a feature extension. An XMPP-CPIM gateway SHOULD NOT generate such headers.

3.2.10 CPIM Required Headers

A "Message/CPIM" object MAY include an optional 'Required' header to specify mandatory-to-recognize features. An XMPP-CPIM gateway SHOULD NOT generate such headers.

3.2.11 MSGFMT MIME Content-type

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' header for that object. The "Content-type" MUST be set to "text/plain" and the charset MUST be set to UTF-8.

Example: Content-type for Encapsulated Object

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

3.2.12 MSGFMT MIME Content-ID

As specified in [MIME], the Content-ID is OPTIONAL for MIME objects. While an XMPP-CPIM gateway MAY generate a Content-ID for encapsulated MIME objects, it is NOT REQUIRED to do so. If included, Content-ID values MUST be generated to be world-unique.

Example: Content-ID for Encapsulated Object

```
  Content-ID: <123456789@example.net>
```

3.2.13 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?
```

3.2.14 XMPP 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.

3.3 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.

3.3.1 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>
```

3.3.2 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>
```

3.3.3 CPIM 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 that information on to the XMPP recipient.

3.3.4 XMPP Message Type

MSGFMT does not possess the concept of a message type that can map to the XMPP 'type' attribute for message stanzas. Therefore an XMPP-CPIM gateway SHOULD NOT include the 'type' attribute on the messages it sends to XMPP recipients.

3.3.5 CPIM 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 that information on to the XMPP recipient.

3.3.6 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 the "tag" value supplied in the string ';lang=tag' included 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>
```

3.3.7 CPIM 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.

3.3.8 CPIM Required Headers

"Message/CPIM" objects MAY include an optional 'Required' 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.

3.3.9 MSGFMT MIME Content-type

As specified in [MSGFMT], a "Message/CPIM" object MAY contain any arbitrary MIME content. However, support for arbitrary content types is not a requirement in XMPP; in particular, the <body/> child element of an XMPP message stanza MUST contain XML character data only. Therefore, an XMPP-CPIM gateway MUST NOT map to an XMPP message stanza a "Message/CPIM" object whose encapsulated MIME object has a Content-type other than "text/plain" (with the exception of multi-part MIME objects as specified in [XMPP-E2E]).

3.3.10 MSGFMT 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 is NOT REQUIRED to do so.

Example: Content-ID for Encapsulated Object

```
MIME header
  Content-ID: <123456789@example.net>

XMPP 'id' attribute (OPTIONAL)
  <message id='123456789@example.net'>
    ...
  </message>
```

3.3.11 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>
```

4. Mapping of Presence

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].

4.1 Identification of Presentities

There is a one-to-one relationship between an XMPP entity and a CPP presentity when the address of the XMPP entity contains only a node identifier and domain identifier, and the node identifier uniquely corresponds to an IM user who possesses an account on an XMPP server. However, the syntax of presentities is specified as including the 'pres:' URI scheme as defined in [CPP], whereas XMPP addresses do not include that scheme, so any mapping between presentities and XMPP addresses must add or remove the 'pres:' URI scheme as appropriate.

4.2 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.

4.2.1 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 PIDF <presence/> root element. To map the XMPP 'from' attribute to the PIDF '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 (PIDF)

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

PIDF '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 PIDF <tuple/> child element.

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>
```

4.2.2 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>
```

4.2.3 CPIM Courtesy Copy

The core XMPP specification does not include syntax for specifying a "courtesy copy" (non-primary addressee) for a presence stanza. Therefore, an XMPP-CPIM gateway SHOULD NOT generate the 'cc' header of a "Message/CPIM" object.

4.2.4 XMPP Stanza ID

An XMPP presence stanza MAY possess an 'id' attribute, which is used by the sending application for the purpose of tracking stanzas. There is no mapping of an XMPP 'id' attribute to a "Message/CPIM" header, common MIME features, or PIDF elements and attributes. Therefore if an XMPP stanza received by an XMPP-CPIM gateway possesses an 'id' attribute, the gateway SHOULD ignore the value provided.

4.2.5 CPIM DateTime Header

The core XMPP specification does not include syntax for specifying the datetime at which a presence stanza was sent. However, an XMPP-CPIM gateway MAY include a 'DateTime' header in the "Message/CPIM" object it generates, the value of which SHOULD be the datetime at which the presence stanza was received for processing by the

gateway.

4.2.6 CPIM Subject Header

An XMPP presence stanza contains no information that can be mapped to the 'Subject' header of a "Message/CPIM" object. Therefore an XMPP-CPIM gateway SHOULD NOT generate such headers when mapping XMPP presence stanzas.

4.2.7 CPIM Header Extensions

A "Message/CPIM" object MAY include an optional 'NS' header to specify the namespace of a feature extension. An XMPP-CPIM gateway SHOULD NOT generate such headers.

4.2.8 CPIM Required Headers

A "Message/CPIM" object MAY include an optional 'Required' header to specify mandatory-to-recognize features. An XMPP-CPIM gateway SHOULD NOT generate such headers.

4.2.9 PIDF MIME Content-type

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 time, 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.

Example: Content-type for Encapsulated PIDF Object

```
  Content-type: application/pidf+xml; charset=utf-8
```

4.2.10 PIDF MIME Content-ID

As specified in [MIME], the Content-ID is OPTIONAL for MIME objects. While an XMPP-CPIM gateway MAY generate a Content-ID for encapsulated MIME objects, it is NOT REQUIRED to do so. If included, Content-ID values MUST be generated to be world-unique.

Example: Content-ID for Encapsulated Object

```
  Content-ID: <123456789@example.net>
```

4.2.11 XMPP 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 5).

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>
```

4.2.12 XMPP 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>
```

4.2.13 XMPP 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>
```

4.2.14 PIDF Contact Element

The core XMPP specification does not include syntax for specifying the URL of a contact address, since the contact address is implicit in the 'from' attribute of the XMPP presence stanza. However, an XMPP-CPIM gateway MAY include the <contact/> child of the <tuple/> element, the value of which SHOULD be the <user@host> of the XMPP sender, prepended by the "im:" Instant Messaging URI scheme.

Example: PIDF Contact Element

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

   PIDF <contact/> 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>im:juliet@example.com</contact>
       </tuple>
     </presence>
```

4.2.15 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>
```

4.2.16 PIDF Timestamp Element

The core XMPP specification does not include syntax for specifying the datetime at which a presence stanza was sent. However, an XMPP-CPIM gateway MAY include a <timestamp/> element within the PIDF document it generates, the value of which SHOULD be the datetime at which the presence stanza was received for processing by the gateway.

4.2.17 XMPP 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.

4.3 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.

4.3.1 From Address

The 'From' header of a "Message/CPIM" object maps to the 'from' attribute of an XMPP presence 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
  <presence from='romeo@example.net'>
    ...
  </presence>
```

4.3.2 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>
```

4.3.3 CPIM 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 that information on to the XMPP recipient.

4.3.4 CPIM 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 that information on to the XMPP recipient.

4.3.5 CPIM 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 that information on to the XMPP recipient.

4.3.6 CPIM 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.

4.3.7 CPIM Required Headers

"Message/CPIM" objects MAY include an optional 'Required' 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.

4.3.8 PIDF MIME Content-type

As specified in [MSGFMT], a "Message/CPIM" object MAY contain any arbitrary MIME content. However, support for arbitrary content types is not a requirement in XMPP. 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]).

4.3.9 PIDF 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 is NOT REQUIRED to do so.

Example: Content-ID for Encapsulated Object

```
MIME header
  Content-ID: <123456789@example.net>

XMPP 'id' attribute (OPTIONAL)
  <presence id='123456789@example.net'>
    ...
  </presence>
```

4.3.10 PIDF 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'/>
```

4.3.11 PIDF Extended Status Information

PIDF documents may contain extended <status/> content. As of this writing there are no pre-defined extended status states that correspond to the defined values of the XMPP <show/> element ('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 PIDF values will be mapped to the relevant XMPP values.

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>
```

4.3.12 PIDF 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>Wooing Juliet</note>
    </tuple>
  </presence>
```

XMPP <status/> element

```
  <presence from='romeo@example.net/orchard'>
    <show>dnd</show>
    <status>Wooing 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 5.4.2)).

4.3.13 PIDF Contact Element

The core XMPP specification does not include syntax for specifying the URL 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, the gateway MAY map the 'priority' element as specified in the following section.

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'/>
```

4.3.14 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).

4.3.15 PIDF 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 that information on to the XMPP recipient.

5. 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.)

5.1 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:

- 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.

- 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.

- There is no XMPP mapping for the CPP "duration parameter", since XMPP subscriptions are active until they have been explicitly "unsubscribed" (see Subscription Durations (Section 5.3)).

- 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:

- 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.

- 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.

- 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.

5.2 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:

- 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.

- 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.

- There is no XMPP mapping for the CPP "duration parameter", since XMPP subscriptions are active until they have been explicitly "unsubscribed" (see Subscription Durations (Section 5.3)).

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

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:

- The target parameter (XMPP "to" address) does not refer to a valid XMPP entity

- Access control rules do not permit the watcher presentity to subscribe to the target XMPP entity

- 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.

5.3 Subscription Durations

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.

5.4 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 4).

5.4.1 Multiple Resources

Semantically, PIDF contains the notion of multiple presence "tuples". Normally, a PIDF 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 PIDF document that contains multiple tuples into multiple XMPP presence stanzas, and SHOULD generate only one PIDF 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 PIDF 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 PIDF recommendation to provide information about multiple "resources" in multiple tuples rather than in multiple PIDF 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.

5.4.2 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. However, the notion of presence about a user rather than a user's resources is meaningless in the XMPP context. Therefore, an XMPP-CPIM gateway MUST NOT map a PIDF document with zero tuples into an XMPP presence stanza, and MUST NOT generate such a PIDF document when receiving a presence stanza from an XMPP entity (i.e., all PIDF documents generated by the gateway MUST contain at least one <tuple/> element).

5.5 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:

- 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.

- 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.

- The CPP "duration parameter" MUST be set to zero.

- 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.

5.6 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:

- 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.

- 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.

- The CPP "duration parameter" MUST be set to zero.

- 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.

6. Internationalization Considerations

6.1 Addresses

Although XMPP enables full internationalization of XMPP addresses

(see [XMPP-CORE]), there is no guarantee that non-XMPP addresses can include characters other than [US-ASCII]. If an XMPP user attempts to communicate with a non-XMPP user through an XMPP-CPIM gateway, the XMPP user SHOULD NOT assume that the non-XMPP service supports fully internationalized addresses.

6.2 Character Encodings

The following rules apply to the mapping of character encodings (charsets):

1. A gateway SHOULD map a "Message/CPIM" object whose charset is "US-ASCII" or "UTF-8".

2. A gateway SHOULD NOT map a "Message/CPIM" object whose charset is other than "US-ASCII" or "UTF-8".

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].

Normative References

[CPIM] Peterson, J., "Common Profile for Instant Messaging (CPIM)", draft-ietf-impp-im-04 (work in progress), August 2003.

[CPP] Peterson, J., "Common Profile for Presence (CPP)", draft-ietf-impp-pres-04 (work in progress), August 2003.

[IMP-MODEL]
Day, M., Rosenberg, J. and H. Sugano, "A Model for Presence and Instant Messaging", RFC 2778, February 2000,

<http://www.ietf.org/rfc/rfc2778.txt>.

[IMP-REQS]
Day, M., Aggarwal, S. 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] Atkins, D. and G. Klyne, "Common Presence and Instant Messaging Message Format", draft-ietf-impp-cpim-msgfmt-08 (work in progress), January 2003.

[PIDF] Fujimoto, S., Sugano, H., Klyne, G., Bateman, A., Carr, W. and J. Peterson, "CPIM Presence Information Data Format", draft-ietf-impp-cpim-pidf-08 (work in progress), May 2003.

[TERMS] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[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", RFC 2279, January 1998.

[XMPP-CORE]
Saint-Andre (ed.), P., "Extensible Messaging and Presence Protocol (XMPP): Core", draft-ietf-xmpp-core-20 (work in progress), November 2003.

[XMPP-E2E]
Saint-Andre (ed.), P., "End-to-End Object Encryption in the Extensible Messaging and Presence Protocol (XMPP)", draft-ietf-xmpp-e2e-06 (work in progress), November 2003.

[XMPP-IM] Saint-Andre (ed.), P., "Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence", draft-ietf-xmpp-im-19 (work in progress), November 2003.

Informative References

[RFC2822] Resnick, P., "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.

Author's Address

Peter Saint-Andre
Jabber Software Foundation

EMail: stpeter@jabber.org

Appendix A. Revision History

Note to RFC Editor: please remove this entire appendix, and the corresponding entries in the table of contents, prior to publication.

A.1 Changes from draft-ietf-xmpp-cpim-02

- Removed references to UTF-16.

- Changed MUST NOT to SHOULD NOT for mapping of courtesy copy data.

- Added information about internationalization of addresses.

- Completed formatting changes to meet RFC Editor requirements.

A.2 Changes from draft-ietf-xmpp-cpim-01

- Added subsection about handling presence notifications for multiple XMPP resources and multiple PIDF tuples.

- Added subsection about PIDF documents that contain zero tuples.

- Further specified mapping between XMPP addresses and CPIM instant inboxes and presentities.

A.3 Changes from draft-ietf-xmpp-cpim-00

- Updated references.

- Made several small editorial changes.

Intellectual Property Statement

Full Copyright Statement

Acknowledgment

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