Network Working Group
Internet-Draft
Expires: December 21, 2003

P. Saint-Andre
Jabber Software Foundation
T. Bamonti
Jabber, Inc.

June 22, 2003

XMPP CPIM Mapping
draft-ietf-xmpp-cpim-00

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on December 21, 2003.

Copyright Notice

Copyright (C) The Internet Society (2003). All Rights Reserved.

Abstract

This document 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 4
1.1 Overview 4
1.2 Terminology 5
1.3 Conventions Used in this Document 5
1.4 Discussion Venue 5
1.5 Intellectual Property Notice 5
2. Approach 6
3. Mapping of Instant Messages 7
3.1 Identification of Instant Inboxes 7
3.2 Syntax Mapping from XMPP to CPIM Specifications 7
3.2.1 From Address 7
3.2.2 To Address 8
3.2.3 CPIM Courtesy Copy 8
3.2.4 XMPP Stanza ID 8
3.2.5 XMPP Message Type 8
3.2.6 XMPP Message Thread 9
3.2.7 CPIM DateTime Header 9
3.2.8 Message Subject 9
3.2.9 CPIM Header Extensions 9
3.2.10 CPIM Required Headers 10
3.2.11 MSGFMT MIME Content-type 10
3.2.12 MSGFMT MIME Content-ID 10
3.2.13 Message Body 10
3.2.14 XMPP Message Extensions 11
3.3 Syntax Mapping from CPIM Specifications to XMPP 11
3.3.1 From Address 11
3.3.2 To Address 12
3.3.3 CPIM Courtesy Copy 12
3.3.4 XMPP Message Type 12
3.3.5 CPIM DateTime Header 12
3.3.6 Message Subject 12
3.3.7 CPIM Header Extensions 13
3.3.8 CPIM Required Headers 13
3.3.9 MSGFMT MIME Content-type 13
3.3.10 MSGFMT MIME Content-ID 13
3.3.11 Message Body 14
4. Mapping of Presence 15
4.1 Identification of Presentities 15
4.2 Syntax Mapping from XMPP to CPIM Specifications 15
4.2.1 From Address 15
4.2.2 To Address 16
4.2.3 CPIM Courtesy Copy 17
4.2.4 XMPP Stanza ID 17
4.2.5 CPIM DateTime Header 17
4.2.6 CPIM Subject Header 17
4.2.7 CPIM Header Extensions 17

4.2.8 CPIM Required Headers 18
4.2.9 PIDF MIME Content-type 18
4.2.10 PIDF MIME Content-ID 18
4.2.11 XMPP Presence Type 18
4.2.12 XMPP Show Element 19
4.2.13 XMPP Status Element 20
4.2.14 PIDF Contact Element 21
4.2.15 Presence Priority 22
4.2.16 PIDF Timestamp Element 22
4.2.17 XMPP Presence Extensions 22
4.3 Syntax Mapping from CPIM Specifications to XMPP 23
4.3.1 From Address 23
4.3.2 To Address 23
4.3.3 CPIM Courtesy Copy 24
4.3.4 CPIM DateTime Header 24
4.3.5 CPIM Subject Header 24
4.3.6 CPIM Header Extensions 24
4.3.7 CPIM Required Headers 24
4.3.8 PIDF MIME Content-type 25
4.3.9 PIDF MIME Content-ID 25
4.3.10 PIDF Basic Presence Status 25
4.3.11 PIDF Extended Status Information 26
4.3.12 PIDF Note Element 27
4.3.13 PIDF Contact Element 28
4.3.14 Presence Priority 29
4.3.15 PIDF Timestamp Element 29
5. XMPP-CPIM Gateway as Presence Service 30
5.1 Requesting a Subscription 30
5.2 Receiving a Subscription Request 31
5.3 Subscription Durations 32
5.4 The Notify Operation 32
5.5 Unsubscribing 33
5.6 Cancelling a Subscription 33
6. Mapping of Character Encodings 35
7. Security Considerations 36
Normative References 37
Informative References 38
Authors' Addresses 38
Intellectual Property and Copyright Statements 39

1. Introduction

1.1 Overview

The Instant Messaging and Presence (IMPP) Working Group has defined an abstract framework for interoperability among instant messaging and presence systems that are compliant with RFC 2779 [1]. This framework is commonly called Common Presence and Instant Messaging or "CPIM". The CPIM specifications include a Common Profile for Instant Messaging [2] (also called CPIM), a Common Profile for Presence [3] (CPP), a CPIM Message Format [4] (MSGFMT), and a Common Presence Information Data Format [5] (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". However, the term "XMPP-CPIM Gateway" is used to refer to a gateway between a XMPP service and a non-XMPP service, where the common format is defined by the CPIM specifications.)

This document describes how the Extensible Messaging and Presence Protocol (XMPP Core [6], XMPP IM [7]) maps to the abstract model contained in the CPIM specifications, mainly for the purpose of establishing gateways between XMPP services and non-XMPP instant messaging (IM) services that conform to RFC 2779 [1]. Such gateways may be established to interpret the protocols of one service and translate them into the protocols of the other service. In the case of communications between an XMPP service and a non-XMPP service, we can visualize this relationship as follows:



This document 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 End-to-End Object Encryption in XMPP [8].

1.2 Terminology

This memo inherits vocabulary defined in RFC 2778 [9]. 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 [6]. Terms such as ENTITY, JID, 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 RFC 2119 [10].

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.

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 fragments (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 [11]; instead, XMPP Core [6] 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 [11] content-types: "Message/CPIM" for messages (MSGFMT [4]) and "application/pidf+xml" for presence (PIDF [5]). The syntax of "Message/CPIM" objects is similar to but stricter than that of RFC 2822 [12], and includes the ability to include arbitrary MIME media types [13]. 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 [4] and PIDF [5] in order to comply with the semantics defined by CPIM [2] and CPP [3]. 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 [2].

3.1 Identification of Instant Inboxes

There is a one-to-one relationship between an XMPP entity and a CPIM instant inbox when the JID of the 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.

3.2 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 MUST NOT include a 'from' attribute; instead, the sender's server stamps the "from" address and sets its value to the full authzid (including resource identifier) provided by the client when authenticating. Thus an XMPP-CPIM gateway will receive from the sender's XMPP server a message stanza containing a "from" address of the form <node@domain/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@capulet.com/balcony'>
    ...
  </message>

CPIM 'From' header
  From: <im:juliet@capulet.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 <node@domain> or <node@domain/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@montague.net/orchard'>
    ...
  </message>

CPIM 'To' header
  To: <im:romeo@montague.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 MUST 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. The <subject/> element MAY include an 'xml:lang' attribute specifying the language in which the subject is written. To map the XMPP <subject/> element to the 'Subject' header of a "Message/CPIM" object, the gateway SHOULD simply map the XMPP CDATA to the value of the 'Subject' header. 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

RFC 2045 [11] specifies that the default Content-type of a MIME object is "Content-type: text/plain; charset=us-ascii". Because XMPP uses a different character encoding (either UTF-8 or UTF-16 depending on stream negotiation), 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 the character encoding negotiated for the XML stream used by the sender, i.e., either UTF-8 or UTF-16.

Example: Content-type for Encapsulated Object

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

3.2.12 MSGFMT MIME Content-ID

RFC 2045 [11] specifies that 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@montague.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 CDATA 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@montague.net>

  Wherefore art thou, Romeo?
```

3.2.14 XMPP Message Extensions

As defined in XMPP Core [6], 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 End-to-End Object Encryption in XMPP [8], 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 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@montague.net>

XMPP 'from' attribute
  <message from='romeo@montague.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@capulet.com>

XMPP 'to' attribute
  <message to='juliet@capulet.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 MUST 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 a XMPP message stanza. The 'Subject' header MAY

specify the "lang" in which the subject is written. To map the CPIM 'Subject' header to the XMPP <subject/> element, the gateway SHOULD simply map the value of the 'Subject' header to the XMPP CDATA. 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

CPIM [4] specifies that 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 used for End-to-End Object Encryption in XMPP [8]).

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 RFC 2045 [11]. 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@montague.net>

XMPP 'id' attribute (OPTIONAL)
  <message id='123456789@montague.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 CDATA 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@montague.net>

  Wherefore art thou?

XMPP message <body/>
  <message id='123456789@montague.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 [5] object in order to comply with the presence semantics defined by CPP [3].

4.1 Identification of Presentities

There is a one-to-one relationship between an XMPP entity and a CPP presentity when the JID of the 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.

4.2 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 MUST NOT include a 'from' attribute; instead, the sender's server stamps the "from" address and sets its value to the full authzid (including resource identifier) provided by the client when authenticating. Thus an XMPP-CPIM gateway will receive from the sender's XMPP server a presence stanza containing a "from" address of the form <node@domain/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@capulet.com/balcony'>
    ...
  </presence>

CPIM 'From' header
  From: <im:juliet@capulet.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@capulet.com/balcony'>
    ...
  </presence>

PIDF 'entity' attribute
  <presence entity='pres:juliet@capulet.com'>
    ...
  </presence>
```

Finally, an XMPP-CPIM gateway SHOULD map the resource identifier of the JID 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@capulet.com/balcony'>
    ...
  </presence>

PIDF 'id' for <tuple/>
  <presence entity='pres:juliet@capulet.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 SHOULD include a 'to' attribute on a presence stanza, and MUST include it if the presence is intended for delivery to another user. Thus an XMPP-CPIM gateway will receive from the sender's XMPP server a presence stanza containing a "to" address of the form <node@domain> or <node@domain/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@montague.net/orchard'>
    ...
  </presence>

CPIM 'To' header
  To: <im:romeo@montague.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 MUST 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 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.

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

RFC 2045 [11] specifies that the default Content-type of a MIME object is "Content-type: text/plain; charset=us-ascii". Because XMPP uses a different character encoding (either UTF-8 or UTF-16 depending on stream negotiation) 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 the character encoding negotiated for the XML stream used by the sender, i.e., either UTF-8 or UTF-16.

Example: Content-type for Encapsulated PIDF Object

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

4.2.10 PIDF MIME Content-ID

RFC 2045 [11] specifies that 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@montague.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 [3] "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@capulet.com/balcony'/>

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

Example: Unavailable Presence

```
XMPP unavailable presence
  <presence from='juliet@capulet.com/balcony'/>

PIDF basic presence (CLOSED)
  <?xml version='1.0' encoding='UTF-8'?>
  <presence xmlns='urn:ietf:params:xml:ns:pidf'
            entity='pres:romeo@montague.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 CDATA 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@capulet.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@capulet.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@capulet.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@capulet.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 bare JID (<user@domain>) of the XMPP sender, prepended by the "im:" Instant Messaging URI scheme.

Example: PIDF Contact Element

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

PIDF <contact/> element
  <?xml version='1.0' encoding='UTF-8'?>
  <presence xmlns='urn:ietf:params:xml:ns:pidf'
            entity='pres:juliet@capulet.com'>
    <tuple id='balcony'>
      ...
      <contact>im:juliet@capulet.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 maximimum of three decimal places. If an XMPP-CPIM gateway maps these values, it SHOULD treat XMPP <priority>0</priority> as PIDF priority='0' and <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@capulet.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@capulet.com'>
    <tuple id='balcony'>
      ...
      <contact priority='0.102'>im:juliet@capulet.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 [6], 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 End-to-End Object Encryption in XMPP [8], 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 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@montague.net>
```

XMPP 'from' attribute

```
<presence from='romeo@montague.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@capulet.com>

XMPP 'to' attribute
  <presence to='juliet@capulet.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 MUST 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

CPIM [4] specifies that 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 used for End-to-End Object Encryption in XMPP [8]).

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 RFC 2045 [11]. 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@montague.net>

XMPP 'id' attribute (OPTIONAL)
  <presence id='123456789@montague.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@montague.net'>
    <tuple id='orchard'>
      <status>
        <basic>open</basic>
      </status>
    </tuple>
  </presence>

XMPP available presence
  <presence from='romeo@montague.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@montague.net'>
    <tuple id='orchard'>
      <status>
        <basic>closed</basic>
      </status>
    </tuple>
  </presence>

XMPP unavailable presence
  <presence from='romeo@montague.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@montague.net'>
  <tuple id='orchard'>
    <status>
      <basic>open</basic>
      <im:im>busy</im:im>
    </status>
  </tuple>
</presence>
```

XMPP <show/> element

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

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 CDATA 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@montague.net'>
    <tuple id='orchard'>
      ...
      <contact>im:romeo@montague.net</contact>
    </tuple>
  </presence>

XMPP presence stanza
  <presence from='romeo@montague.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 [3] 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 [5] 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 [7]; 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 (node@domain) MUST be mapped to the CPP "watcher parameter" field (pres:node@domain). The XMPP-CPIM gateway MUST append the "pres:" Presence URI scheme to the front of the address.

- The XMPP 'to' attribute (node@domain) MUST be mapped to the CPP "target parameter" field (pres:node@domain). 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:node@domain) MUST be mapped to the XMPP 'from' attribute (node@domain). The XMPP-CPIM gateway MUST remove the "pres:" Presence URI scheme from the front of the address.

- The CPP "target parameter" field (pres:node@domain) MUST be mapped to the XMPP 'to' attribute (node@domain). 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 RFC 2778 [9] or RFC 2779 [1], duration parameter mapping is a local issue that falls outside the scope of this document.

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).

5.5 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 (node@domain) MUST be mapped to the CPP "watcher parameter" field (pres:node@domain). The XMPP-CPIM gateway MUST append the "pres:" Presence URI scheme to the front of the address.

- The XMPP 'to' attribute (node@domain) MUST be mapped to the CPP "target parameter" field (pres:node@domain). 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:node@domain) MUST be mapped to the XMPP 'from' attribute (node@domain). The XMPP-CPIM gateway MUST remove the "pres:" Presence URI scheme from the front of the address.

- The CPP "target parameter" field (pres:node@domain) MUST be mapped to the XMPP 'to' attribute (node@domain). 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. Mapping of Character Encodings

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

1. A gateway SHOULD NOT map a "Message/CPIM" object whose charset is other than "us-ascii", "utf-8", or "utf-16".

2. A gateway SHOULD map a "Message/CPIM" object whose charset is "us-ascii" no matter whether the character encoding negotiated for the XMPP recipient's XML stream is UTF-8 or UTF-16.

3. A gateway SHOULD map a "Message/CPIM" object whose charset is "utf-8" if the character encoding negotiated for the XMPP recipient's XML stream is UTF-8.

4. A gateway SHOULD map a "Message/CPIM" object whose charset is "utf-16" if the character encoding negotiated for the XMPP recipient's XML stream is UTF-16.

5. A gateway SHOULD NOT map a "Message/CPIM" object whose charset is "utf-8" if the character encoding negotiated for the XMPP recipient's XML stream is UTF-16.

6. A gateway SHOULD NOT map a "Message/CPIM" object whose charset is "utf-16" if the character encoding negotiated for the XMPP recipient's XML stream is UTF-8.

7. Security Considerations

Detailed security considerations for instant messaging and presence protocols are given in RFC 2779 [1], 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 [2] and CPP [3]. 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 End-to-End Object Encryption in XMPP [8].

Normative References

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

[2] Crocker, D. and J. Peterson, "Common Profile for Instant Messaging (CPIM)", draft-ietf-impp-im-02 (work in progress), March 2003.

[3] Crocker, D. and J. Peterson, "Common Profile for Presence (CPP)", draft-ietf-impp-pres-02 (work in progress), March 2003.

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

[5] 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.

[6] Saint-Andre, P. and J. Miller, "XMPP Core", draft-ietf-xmpp-core-13 (work in progress), June 2003.

[7] Saint-Andre, P. and J. Miller, "XMPP Instant Messaging", draft-ietf-xmpp-im-12 (work in progress), June 2003.

[8] Saint-Andre, P., "End-to-End Object Encryption in XMPP", draft-ietf-xmpp-e2e-03 (work in progress), May 2003.

[9] 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>.

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

[11] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

Informative References

[12] Resnick, P., "Internet Message Format", RFC 2822, April 2001.

[13] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

Authors' Addresses

Peter Saint-Andre
Jabber Software Foundation

EMail: stpeter@jabber.org

Tony Bamonti
Jabber, Inc.

EMail: tbamonti@jabber.com

Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property 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; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication 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 implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

Full Copyright Statement

Copyright (C) The Internet Society (2003). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assignees.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

Acknowledgement

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