Network working group
Internet draft

G. Klyne, Content Technologies Ltd.
D. H. Crocker, Brandenburg Consulting
M. T. Rose, Invisible Worlds, Inc.

18 October 2000
Expires: April 2001

Instant Messaging using IMXP
<draft-klyne-apex-impp-00.txt>

Status of this memo

Copyright Notice

Abstract

This document describes how to provision an instant text messaging and presence service using IMXP.

Discussion of this document

Please send comments to: <IMXPwg@lists.invisibleworlds.com>.

To subscribe: send a message with the body 'subscribe' to
<IMXPwg-request@lists.invisibleworlds.com>.

Table of contents

1. Introduction

This document describes how to provision an instant text messaging and presence service using IMXP [4,5,6]. The service described here conforms to CPIM, the common interoperability framework for instant messaging [3].

1.1 Structure of this document

Section 2 provides background material, and sets out the goals of this service specification.

Section 3 describes the IMXP-provisioned instant text messaging service in terms of the basic CPIM Message operation.

Section 4 describes the IMXP-provisioned presence service in terms of basic CPIM Subscribe/Unsubscribe/Notify operations.

1.2 Document terminology and conventions

Many special terms used in this document are described in RFC 2778 [2].

The 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 [7].

> NOTE: Comments like this provide additional nonessential information about the rationale behind this document. Such information is not needed for building a conformant implementation, but may help those who wish to understand the design in greater depth.

[[[Editorial comments and questions about outstanding issues are provided in triple brackets like this. These working comments should be resolved and removed prior to final publication.]]]

2. Background and goals

The requirements for an instant messaging and presence (IM/P) service are set out in RFC 2779 [1]. This sets out facilities for sending instant messages, real-time discovery information about the status of other parties on the network (presence), and discovering who is monitoring information about somebody's status (watchers).

A framework for interoperability between different protocols satisfying these requirements is set out in CPIM [2].

The service described by this document is a provisioning of an instant text messaging and presence service using IMXP.

2.1 Overview of IMXP

The basic philosophy of IMXP is:

- The core mechanism is very simple: a range of services can be built using the basic core mechanisms provided.

- Its design is based on familiar Internet mail architecture, leveraging a wealth of related experience.

The IMXP relay mesh uses lightweight, near-real-time application datagram transfer nodes, analogous to email MTAs:

- Relay processing is kept simple. Essential intelligence is kept at or near the network edge to enhance scalability; relays are not required to maintain state concerning message transfer.

- Addressing and routing follow the classic email model. This uses hierarchical addresses (domain names) that can be understood by the entire relay mesh.

- Hop-by-hop security framework. Authentication, privacy, and authorization rely on domains "keeping their own houses in order", in line with current Internet infrastructure. End-to-end security (OpenPGP or S/MIME) may be added to provide greater security.

- Transport independence. A convergence layer (BEEP [8]) carries IMXP identically over variety of transports.

- Other applications may use the same relay mesh. Asynchronous near-real-time message exchange with accessible, predictable behaviour is applicable to a numnber of loosely-coupled applications, or which instant text messaging is just one.

2.2 INSTANT INBOX addressing

CPIM defines an address format for instant messaging based on a URI containing a domain name and a local part.

IMXP uses a message address with the same form as an e-mail address, also containing a domain name and a local part. Thus, a perfect conversion between IMXP and CPIM addressing can be achieved:

<local-part>@<domain> <==> im:<local-part>@<domain>

2.3 Identifying PRESENTITIES

CPIM defines an identifier format for a presentity that is a URI containing a domain name and a local part.

The IMXP presence service [6] uses a presentity identifier format with the same form as an e-mail address, also containing a domain name and a local part.

Thus, a perfect conversion between IMXP and CPIM presentity identifiers can be achieved:

<local-part>@<domain> <==> pres:<local-part>@<domain>

2.4 WATCHER addressing

CPIM and IMXP both use the same form for watcher addresses that they use for presentity identifiers, so the same mapping applies.

2.5 PRESENCE SERVICE addressing

CPIM uses the presentity identifier or watcher address to locate the corresponding service.

The IMXP presence service [6] uses a presence service address with the same form as an e-mail address, containing a domain name and a local part, but in which the local-part is fixed as "imxp=presence".

When mapping IMXP to CPIM, the service address is not needed (having the same domain address part as the corresponding presenity identifier or watcher address).

When mapping CPIM to IMXP, a service address is constructed using the domain part of the corresponding CPIM presence URI:

pres:<local-part>@<domain> ==> imxp=presence@<domain>

2.6 IMXP access provisioning

RFC 2779 requires that there be mechanisms for authorizing who is allowed to perform various operations, or access private information. This requirement is not reflected in CPIM because it is seen as being handled locally within a domain.

IMXP provides a flexible mechanism based on the IMXP access service [5]. IMXP components operating within a domain are required to check that the requested operation is allowed according to permissions lodged with the domain's access service.

2.7 Service goals

The goals of the service defined here are:

(a) to satisfy the requirements set out in RFC 2779 [1].

(b) to allow interoperability with other IM/P protocols using the common framework set out in CPIM [3].

(c) to allow simple text messages to be exchanged between instant messaging user agents.

3. Instant Message service

IMXP message content is transfered as a MIME object [12] within a multipart/related, or as an XML element in an IMXP Data operation. The IMXP Data operation contains a URI-reference [14] to indicate the message content: a cid: URI is used to indicate another body part within an enclosing multipart/related, or a fragment identifier to indicate an XML <data-content> element within the IMXP Data element. See section 4.1 of the IMXP core specification [4] for further details.

3.1 Format of Instant Messages

An instant message consists of a message header, based on RFC822 [15] encoded in XML [16], and a message content.

The instant message header contains information about the message that is conveyed between message user agents, and not used by the message transfer mechanisms. This may include who the message is from, who it is addressed to, other parties to whom it has been copied, subject of the message, date the message was composed, etc.

The message header also contains a reference to the message content, in the same fashion that an IMXP <Data> element references the entire message. Thus, an instant message can be constructed as a MIME multipart/related:

```
Content-type: Multipart/related
Content-Type: multipart/related; boundary="boundary";
              start="<1@example.com>";
              type="message/rfc822+xml"

--boundary
Content-Type: message/rfc822+xml
Content-ID: <1@example.com>

<rfc822:message
    xmlns:rfc822='URN:IANA:message:rfc822:'
    content='cid:2@example.com>
  <rfc822:from>im:fred@example.com</rfc822:from>
  <rfc822:to>im:barney@anotherdomain.com</rfc822:to>
  <rfc822:to>im:wilma@yetanotherdomain.com</rfc822:to>
  <rfc822:cc>im:dino@example.com</rfc822:cc>
  <rfc822:subject>Hello</rfc822:subject>
  <rfc822:date>Tue, 10 Oct 2000 12:06:16 -0700</rfc822:date>
</rfc822:message>
--boundary
Content-Type: text/plain;charset=UTF-8
Content-ID: <2@example.com>

And this is the <message> text!
--boundary--
```

Alternatively, if the content is expressed in pure text and/or XML it can be contained within the message header:

```
Content-Type: message/rfc822+xml;charset=UTF-8

<message
    xmlns='URN:IANA:message:rfc822:'
    content='#message-text>
  <from>im:fred@example.com</from>
  <to>im:barney@anotherdomain.com</to>
  <to>im:wilma@yetanotherdomain.com</to>
  <cc>im:dino@example.com</cc>
  <subject>Hello</subject>
  <date>Tue, 10 Oct 2000 12:06:16 -0700</date>
  <message-content name='message-text' type='text/plain'>
    And this is the &lt;message&gt; text!
  </message-content>
</message>
```

Notes:

- The RFC822 message in XML format is described more fully in a separate document [18].
- The 'content=' attribute of the <message> element indicates a URI or fragment identifier that names the message content.
- Headers are represented as XML elements, whose content is the corresponding header value.
- Header field names are based on RFC822 header names, using all lower-case characters [15]. (XML element names are case sensitive [16].) The exception is element <message-content>.
- Header field names are associated with an XML namespace [17] identified as 'URN:IANA:message:rfc822:'. (This namespace identifier is based on a work-in-progress [19].)
- Individual message header elements in the message header may appear in any order, except the <message-content> element, which, if used, MUST be the last element in the message header.

- Header contents are same syntax and meaning as corresponding RFC822 header contents [15], except that:

  - Characters are not limited to US-ASCII. UTF-8 charset encoding is used.

  - Address values are presented as URIs. Note that characters in URIs are drawn from a limited repertoire; the URI '%' escape sequence may be used to represent other characters that are legal for the URI scheme used [14].

  - Encoded words (=?...?=) are not needed, and SHOULD NOT be used.

- The 1st example above uses RFC822 as an XML namespace prefix [17]; any name may be used here.

- The 2nd example above uses default XML namespace [17], so no namespace prefix needs to be used.

- The XML reserved characters in the 2nd example above are replaced by their corresponding XML entity references ('<' and '>').

- When message content is included in the message header, its MIME content-type is indicated by the 'type=' attribute of the <message-content> element. The charset for <message-content> data is UTF-8 (i.e. inherited from the surrounding XML header).

- Although theoretically possible in some cases, nested multipart/related structures MUST NOT be flattened.

3.2 Content of an instant message

The instant message format described above allows any kind of message content.

Instant message service endpoints MUST be able to process message content that is provided as "text/plain;charset=US-ASCII" or "text/plain;charset=UTF-8" [13]. Endpoints SHOULD support the "Format=flowed" parameter, per RFC 2646 [11].

3.3 Sending an instant message

An instant message is sent using the IMXP Data operation to the desired recipient address [4].

CPIM does not provide for end-to-end confirmation of receipt, so if a 'statusRequest' option is specified with "targetHop='final'" or 'targetHop=all', it SHOULD NOT also indicate "seeNoEvil='false'" or the IMXP/CPIM gateway may fail the message.

4. Presence service

4.1 Format of presence information

Presence information is transferred in the form of a <publish> element [6] in an IMXP Data operation [4].

4.2 Subscribing to receive presence information

Subscription to presence information is achieved by sending a <Subscribe> element [6] in an IMXP Data operation [4].

The response is an IMXP Data operation with a <Publish> element [6] containing the desired information, followed by further such messages each time the indicated presence information changes.

Updates to the presence information continue to be provided until the specified duration elapses, or a <terminate> element is send to the presence service.

CPIM uses a 'subscribe' operation, and returns a confirmation 'response' operation and at least one separate 'notify' operation, until the duration elapses or an 'Unsubscribe' operation is issued. An IMXP/CPIM gateway should handle creation and correlation of the separate IMXP 'response' operation.

4.3 Polling presence information

Polling for presence information is achieved by issuing a zero-duration subscription.

4.4 Publishing presence information

And endpoint publishes presence information by sending a <publish> element in an IMXP Data operation to its presence service.

The presence service propagates presence information by sending <publish> elements to endpoints that have requested them.

CPIM handles propagation of presence information by issuing 'Notify' operations. (Publication or updating of presence information by an endpoint is not covered by CPIM.)

4.5 Watcher operations

IMXP watcher operations are performed by sending <watch>, <reply>, <notify> and <terminate> elements [6] in IMXP Data operations [4].

(CPIM does not describe distribution of watcher information, as this is regarded as local to an administrative domain.)

5. Internationalization considerations

IMXP uses the address format defined by RFC822, which limits characters in address local parts to US-ASCII. This means that many foreign language personal names cannot be represented.

Similarly, the characters that can be used in domain names are currently severely constrained. Work is under way to define internationalized forms for domain names.

Message content is tagged using standard MIME capabilities (charset parameter for text data [13], and Content-language header for language tagging [22]). IMXP instant messaging user agents are required to be able to process UTF-8 coded character data, but that does not necessarily mean that all characters received can be displayed.

When message content is included in the XML message header, language tagging can be achieved by including an 'xml:lang=' attribute [16] in the <message-content> element.

Presence information is represented using XML. Support for UTF-8 character encoding is requiured for human readable parts of the presence information.

6. Security considerations

The primary form of security provided by IMXP is hop-by-hop, requiring that each IMXP relay must be trusted to handle the messages it receives. The IMXP authorization framework is described in section 4.5 of the IMXP core specification [4], and its use for IMXP instant messaging is elaborated below.

Endpoints may choose to use additional end-to-end security, such as S/MIME [23] or OpenPGP [24] by bilateral arrangement. Such usage is not defined by this specification.

6.1 Security provisioning

IMXP security is based on BEEP session security profiles, coupled with IMXP authorization policies that allow BEEP peers to act as designated IMXP endpoints or relays for designated domains.

IMXP instant messaging endpoints MUST support the following:

- BEEP SASL security profile using the DIGEST-MD5 mechanism [25] [26]. This allows the endpoint to authenticate itself to a relay.

- If confidentiality is required: BEEP TLS security profile, using the TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher suite[9].

- authenticate with BEEP peer identities that are the same as their endpoint identifier. An endpoint thus authenticated should be trusted to originate and receive messages and requests for the indicated endpoint.

IMXP instant messaging relays MUST support the following:

- For communication with IMXP endpoints, support the BEEP security profile noted above.

- For communication with IMXP endpoints or other IMXP relays: BEEP TLS security profile, using the TLS_RSA_WITH_3DES_EDE_CBC_SHA cipher suite [9].

- IMXP relays MUST authenticate themselves to IMXP endpoints and other IMXP relays as "imxp=mesh@<domain>". A relay thus authenticated should be trusted to originate and receive messages and requests for the indicated domain.

NOTE: IMXP instant messaging endpoint support for confidentiality is optional, but if confidentiality is supported then the indicated TLS security profile MUST be supported.

IMXP instant messaging relays are required to support confidentiality when communicating with other relays, using the TLS mechanism indicated.

Security mechanisms other than those noted above may be used by bilateral agreement. Support for additional security profiles can be discovered through BEEP Greeting messages [8].

7. Acknowledgements

The authors thank the following for their contributions: Ned Freed provided valuable advice on the choice of TLS cipher suite.

8. References

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

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

[3] Crocker, D.H., Diacakis, A., Mazzoldi, F., Huitema, C., Klyne, G., Rose, M.T., Rosenberg, J., Sparks, R. and H. Sugano,
"A Common Profile for Instant Messaging (CPIM)",
draft-thenine-im-common-00 (work in progress),
August 2000.

[4] Rose, M.T., Klyne, G. and D.H. Crocker,
"The IMXP",
draft-mrose-imxp-core-01 (work in progress),
September 2000.

[5] Rose, M.T., Klyne, G. and D.H. Crocker,
"The IMXP Access Service"
draft-mrose-imxp-access-01 (work in progress),
September 2000.

[6] Rose, M.T., Klyne, G. and D.H. Crocker,
"The IMXP Presence Service"
draft-mrose-imxp-presence-01 (work in progress),
September 2000.

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

[8] Rose, M.T.,
"The Blocks Extensible Exchange Protocol Framework",
draft-ietf-beep-framework-02 (work in progress),
September 2000.

[9] Dierks, T. and C. Allen,
"The TLS Protocol Version 1.0",
RFC 2246,
January 1999.

[10] Newman, C.,
"Using TLS with IMAP, POP3 and ACAP",
RFC 2595,
June 1999.

[11] Gellens, R.,
"The Text/Plain Format Parameter",
RFC 2646,
August 1999.

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

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

[14] Berners-Lee, T., Fielding, R.T. and L. Masinter,
"Uniform Resource Identifiers (URI): Generic Syntax",
RFC 2396,
August 1998.

[15] Crocker, D.,
"Standard for the format of ARPA Internet text messages",
RFC 822, STD 11,
August 1982.

[16] Tim Bray, Jean Paoli, and C. M. Sperberg-McQueen,
"Extensible Markup Language (XML) 1.0",
W3C recommendation: <http://www.w3.org/TR/REC-xml>,
10 February 1998.

[17] Tim Bray, Dave Hollander, and Andrew Layman
"Namespaces in XML",
W3C recommendation: <http://www.w3.org/TR/REC-xml-names>,
14 January 1999.

[18] Klyne, G., Crocker, D.H. and M.T. Rose,
"XML coding of RFC822 messages"
draft-klyne-message-rfc822-xml-00 (work in progress),
October 2000.

[19] Mealling, M.,
[[[URN namespace for IANA registries]]],
(work in progress),
2000

[20] Levinson, E.,
"The MIME Multipart/Related Content-type",
RFC 2387,
August 1998.

[21] Levinson, E.,
"Content-ID and Message-ID Uniform Resource Locators",
RFC 2392,
August 1998.

[22] Alvestrand, H.,
"Tags for the Identification of Languages",
RFC 1766,
March 1995.
(Defines Content-language header.)

[23] Ramsdell, B.,
"S/MIME Version 3 Message Specification",
RFC 2633,
June 1999.

[24] Callas, J., Donnerhacke, L., Finney, H. and R. Thayer,
"OpenPGP Message Format",
RFC 2440,
November 1998.

[25] Myers, J.,
"Simple Authentication and Security Layer (SASL)",
RFC 2222,
October 1997.

[26] Leach, P. and C. Newman,
"Using Digest Authentication as a SASL Mechanism",
RFC 2831,
May 2000.

9. Authors' addresses

Graham Klyne (editor)
Content Technologies Ltd.
1220 Parkview,
Arlington Business Park
Theale
Reading, RG7 4SA
United Kingdom

Telephone: +44 118 930 1300
Facsimile: +44 118 930 1301
E-mail: GK@ACM.ORG

Marshall T. Rose
Invisible Worlds, Inc.
1179 North McDowell Boulevard
Petaluma, CA 94954-6559
US

Telephone: +1 707 789 3700
E-mail: mrose@invisible.net
URI: http://invisible.net/

David H. Crocker
Brandenburg Consulting
675 Spruce Drive
Sunnyvale, CA 94086
US

Telephone: +1 408 246 8253
E-mail: dcrocker@brandenburg.com
URI: http://www.brandenburg.com/

Appendix A: Amendment history

00a	05-Oct-2000	Memo initially created.
00b	10-Oct-2000	Filled in details of IMXP usage in relation to CPIM. Clarified some presence addressing details. Picked some security profiles.
00c	11-Oct-2000	Introduced message format based on RFC822 encoded with XML. Cosmetic fixes.
00d	11-Oct-2000	Drafted internationalization and security considerations sections. Completed CPIM/IMXP mapping appendices. Revisit security provisioning: moved to "Security considerations" section as it applies to both messaging and presence; specify only authentication is mandatory to implement for endpoint-relay mode, using SASL DIGEST-MD5.
00e	12-Oct-2000	More cosmetic fixes. Delete CPIM/IMXP mapping appendices -- these are now in the IMXP core documents.
01a	18-Oct-2000	Change mandatory-to-implement TLS cipher suite doe relays to TLS_RSA_WITH_3DES_EDE_CBC_SHA. Punt issue of MIME headers in XML message header to separate document. Change content type Message/RFC822|XML to Message/RFC822+XML (per more recent XML-MIME types specification). Clarify that nested Multipart/related structures may not be flattened.

Full copyright statement