

INTERNET-DRAFT
Expires February 28, 2000

Mark Day
Lotus

Jonathan Rosenberg
Bell Labs

Hiroyasu Sugano
Fujitsu

A Model for Presence and Instant Messaging

[draft-ietf-impp-model-03.txt](#)

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/lid-abstracts.txt>

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

This document and related documents are discussed on the impp mailing list. To join the list, send mail to impp-request@iastate.edu. To contribute to the discussion, send mail to impp@iastate.edu. The archives are at <http://lists.fsck.com/cgi-bin/wilma/pip>. The IMPP working group charter, including the current list of group documents, can be found at <http://www.ietf.org/html.charters/impp-charter.html>.

1. Abstract

This document defines an abstract model for a presence and instant messaging system. It defines the various entities involved, defines terminology, and outlines the services provided by the system. The goal is to provide a common vocabulary for further work on requirements for protocols and markup for presence and instant messaging.

2. Introduction

A presence and instant messaging system allows users to subscribe to each other and be notified of changes in state, and for users to send each other short instant messages. A number of requirements and protocols have been proposed to support it [WHODP, ENVY, PRESENCE, RVP-ADDR, RPIM, SIP-PIP, PIPR, RVP]. To facilitate development of a suite of protocols to provide this service, we believe that it is

valuable to first develop a model for the system. The model consists of the various entities involved, descriptions of the basic functions they provide, and most importantly, definition of a vocabulary which can be used to facilitate discussion.

We note that the purpose of this model is to be descriptive and universal: we want the model to map reasonably onto all of the systems that are informally described as presence or instant messaging systems. The model is not intended to be prescriptive or achieve interoperability: an element that appears in the model will not necessarily be an element of an interoperable protocol, and may not even be a good idea.

In this document, each element of the model appears in upper case (e.g., PRESENCE SERVICE). No term in lower case or mixed case is intended to be a term of the model.

The first part of this document is intended as an overview of the model. The overview includes diagrams, and terms are presented in an order that is intended to help the reader understand the relationship between elements. The second part of the document is the actual definition of the model, with terms presented in alphabetical order for ease of reference.

The overview is intended to be helpful but is not definitive; it may contain inadvertent differences from the definitions in the model. For any such difference, the definition(s) in the model are taken to be correct, rather than the explanation(s) in the overview.

3. Overview

The model is intended to provide a means for understanding, comparing, and describing systems that support the services typically referred to as presence and instant messaging. It consists of a number of named entities that appear, in some form, in existing systems. No actual implementation is likely to have every entity of the model as a distinct part. Instead, there will almost always be parts of the implementation that embody two or more entities of the model. However, different implementations may combine entities in different ways.

The model defines two services: a PRESENCE SERVICE and an INSTANT

MESSAGE SERVICE. The PRESENCE SERVICE serves to accept information, store it, and distribute it. The information stored is (unsurprisingly) PRESENCE INFORMATION. The INSTANT MESSAGE SERVICE serves to accept and deliver INSTANT MESSAGES to INSTANT INBOXES.

3.1 PRESENCE SERVICE

The PRESENCE SERVICE has two distinct sets of "clients" (remember, these may be combined in an implementation, but treated separately in the model). One set of clients, called PRESENTITIES, provides PRESENCE INFORMATION to be stored and distributed. The other set of clients, called WATCHERS, receives PRESENCE INFORMATION from the service.

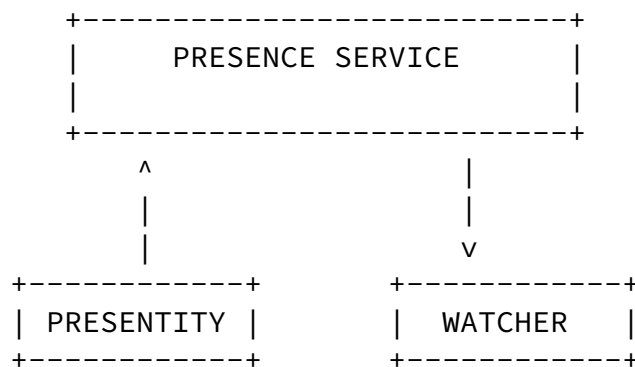


Fig. 1: Overview of Presence Service

There are two kinds of WATCHERS, called FETCHERS and SUBSCRIBERS. A FETCHER simply requests the current value of some PRESENTITY's PRESENCE INFORMATION from the PRESENCE SERVICE. In contrast, a SUBSCRIBER requests notification from the PRESENCE SERVICE of (future) changes in some PRESENTITY's PRESENCE INFORMATION. A special kind of FETCHER is one that fetches information on a regular basis. This is called a POLLER.

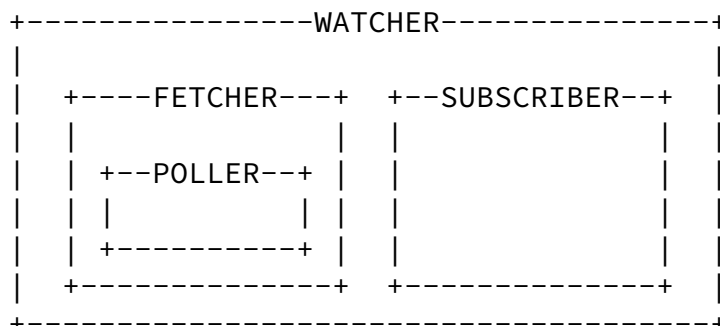


Fig. 2: Varieties of WATCHER

The PRESENCE SERVICE also has WATCHER INFORMATION about WATCHERS and their activities in terms of fetching or subscribing to PRESENCE INFORMATION. The PRESENCE SERVICE may also distribute WATCHER INFORMATION to some WATCHERS, using the same mechanisms that are available for distributing PRESENCE INFORMATION.

Changes to PRESENCE INFORMATION are distributed to SUBSCRIBERS via NOTIFICATIONS. Figures 3a through 3c show the flow of information as a piece of PRESENCE INFORMATION is changed from P1 to P2.

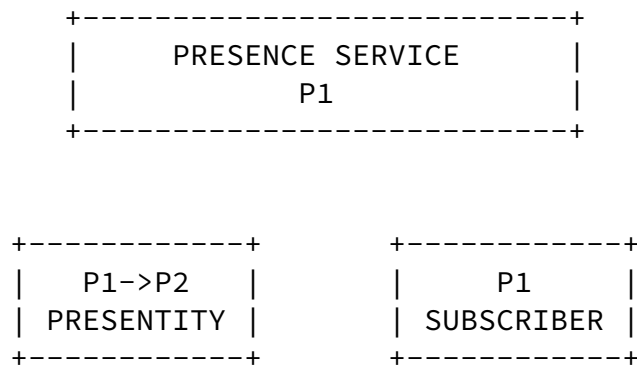


Fig. 3a: NOTIFICATION (Step 1)

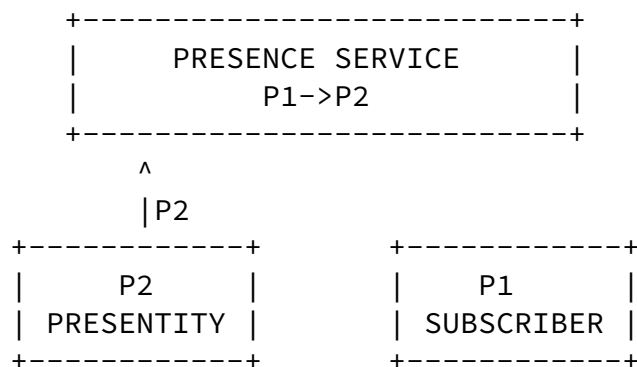
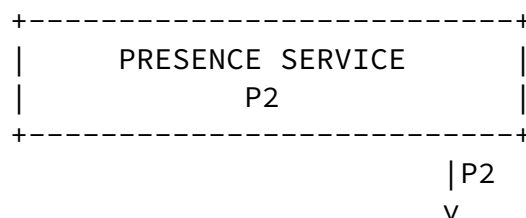


Fig. 3b: NOTIFICATION (Step 2)



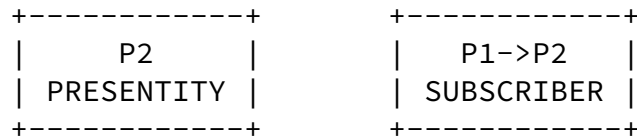


Fig. 3c: NOTIFICATION (Step 3)

3.2 INSTANT MESSAGE SERVICE

The INSTANT MESSAGE SERVICE also has two distinct sets of "clients": SENDERS and INSTANT INBOXES. A SENDER provides INSTANT MESSAGES to the INSTANT MESSAGE SERVICE for delivery. Each INSTANT MESSAGE is addressed to a particular INSTANT INBOX ADDRESS, and the INSTANT MESSAGE SERVICE attempts to deliver the message to a corresponding INSTANT INBOX.

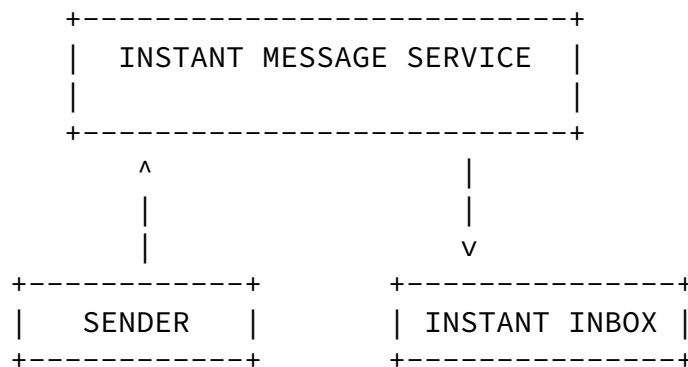


Fig. 4: Overview of Instant Message Service

3.3 Protocols

A PRESENCE PROTOCOL defines the interaction between PRESENCE SERVICE, PRESENTITIES, and WATCHERS. PRESENCE INFORMATION is carried by the PRESENCE PROTOCOL.

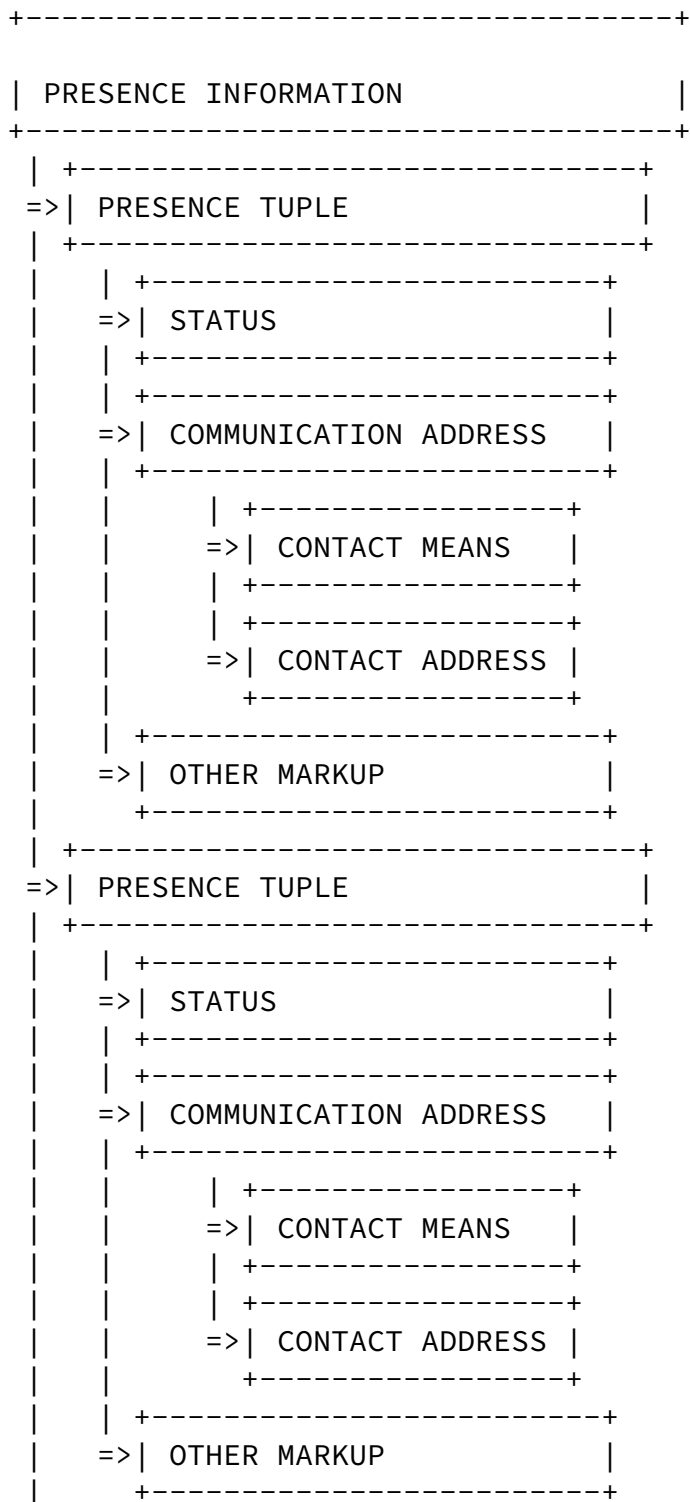
An INSTANT MESSAGE PROTOCOL defines the interaction between INSTANT MESSAGE SERVICE, SENDERS, and INSTANT INBOXES. INSTANT MESSAGES are carried by the INSTANT MESSAGE PROTOCOL.

In terms of this model, we believe that the IMPP working group is planning to develop detailed requirements and specifications for the structure and formats of the PRESENCE PROTOCOL, PRESENCE INFORMATION, INSTANT MESSAGE PROTOCOL, and INSTANT MESSAGES.

3.4 Formats

The model defines the PRESENCE INFORMATION to consist of an arbitrary number of elements, called PRESENCE TUPLES. Each such element consists

of a STATUS marker (which might convey information such as online/offline/busy/away/do not disturb), an optional COMMUNICATION ADDRESS, and optional OTHER PRESENCE MARKUP. A COMMUNICATION ADDRESS includes a COMMUNICATION MEANS and a CONTACT ADDRESS. One type of COMMUNICATION MEANS, and the only one defined by this model, is INSTANT MESSAGE SERVICE. One type of CONTACT ADDRESS, and the only one defined by this model, is INSTANT INBOX ADDRESS. However, other possibilities exist: a COMMUNICATION MEANS might indicate some form of telephony, for example, with the corresponding CONTACT ADDRESS containing a telephone number.



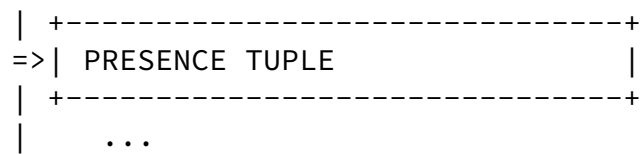


Fig. 5: The structure of PRESENCE INFORMATION

STATUS is further defined by the model to have at least two states that interact with INSTANT MESSAGE delivery -- OPEN, in which INSTANT MESSAGES will be accepted, and CLOSED, in which INSTANT MESSAGES will not be accepted. OPEN and CLOSED may also be applicable to other COMMUNICATION MEANS -- OPEN mapping to some state meaning "available" or "open for business" while CLOSED means "unavailable" or "closed to business." The model allows STATUS to include other values, which may be interpretable by programs or only by persons. The model also allows STATUS to consist of single or multiple values.

[3.5](#) Presence and its effect on Instant Messages

An INSTANT INBOX is a receptacle for INSTANT MESSAGES. Its INSTANT INBOX ADDRESS is the information that can be included in PRESENCE INFORMATION to define how an INSTANT MESSAGE should be delivered to that INSTANT INBOX. As noted above, certain values of the STATUS marker indicate whether INSTANT MESSAGES will be accepted at the INSTANT INBOX. The model does not otherwise constrain the delivery mechanism or format for instant messages. Reasonable people can disagree about whether this omission is a strength or a weakness of this model.

[3.6](#) PRINCIPALS and their agents

This model includes other elements that are useful in characterizing how the protocol and markup work. PRINCIPALS are the people, groups, and/or software in the "real world" outside the system that use the system as a means of coordination and communication. It is entirely outside the model how the real world maps onto PRINCIPALS -- the system of model entities knows only that two distinct PRINCIPALS are distinct, and two identical PRINCIPALS are identical.

A PRINCIPAL interacts with the system via one of several user agents (INBOX USER AGENT; SENDER USER AGENT; PRESENCE USER AGENT; WATCHER USER AGENT). As usual, the different kinds of user agents are split apart in this model even though most implementations will combine at least some of them. A user agent is purely coupling between a PRINCIPAL and some core entity of the system (respectively, INSTANT INBOX; SENDER; PRESENTITY; WATCHER).

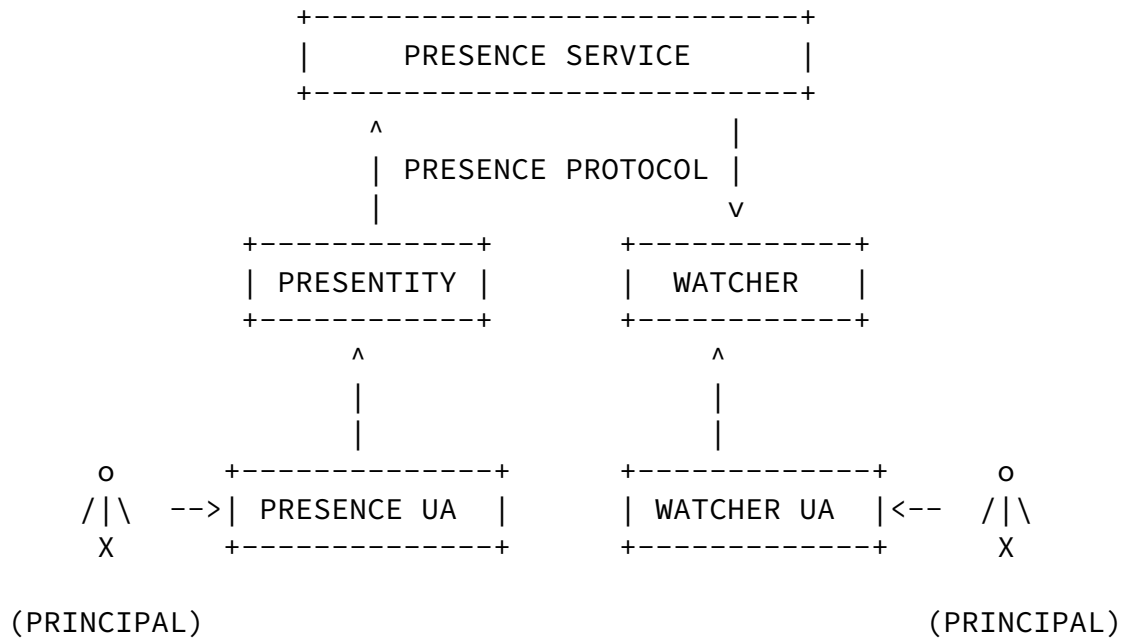


Fig. 6: A presence system

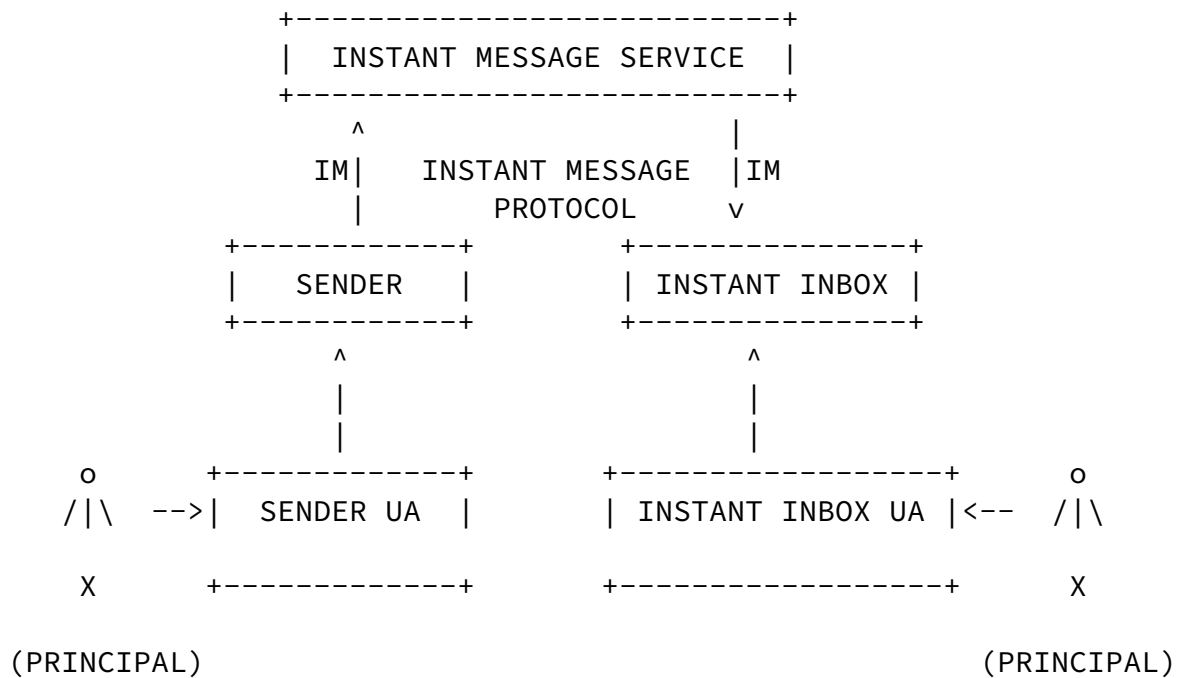


Fig. 7: An instant messaging system

3.7 Examples

A simple example of applying the model is to describe a generic "buddy

list" application. These applications typically expose the user's presence to others, and make it possible to see the presence of others. So we could describe a buddy list as the combination of a PRESENCE USER AGENT and WATCHER USER AGENT for a single PRINCIPAL, using a single PRESENTITY and a single SUBSCRIBER.

We could then extend our example to instant messaging and describe a generic "instant messenger" as essentially a buddy list with additional capabilities for sending and receiving instant messages. So an instant messenger would be the combination of a PRESENCE USER AGENT, WATCHER USER AGENT, INBOX USER AGENT, and SENDER USER AGENT for a single PRINCIPAL, using a single PRESENTITY, single SUBSCRIBER, and single INSTANT INBOX, with the PRESENTITY's PRESENCE INFORMATION including an INSTANT INBOX ADDRESS that leads to the INSTANT INBOX.

4. Model

ACCESS RULES: constraints on how a PRESENCE SERVICE makes PRESENCE INFORMATION available to WATCHERS. For each PRESENTITY's PRESENCE

INFORMATION, the applicable ACCESS RULES are manipulated by the PRESENCE USER AGENT of a PRINCIPAL that controls the PRESENTITY.

Motivation: We need some way of talking about hiding presence information from people.

CLOSED: a distinguished value of the STATUS marker. In the context of INSTANT MESSAGES, this value means that the associated INSTANT INBOX ADDRESS, if any, corresponds to an INSTANT INBOX that is unable to accept an INSTANT MESSAGE. This value may have an analogous meaning for other COMMUNICATION MEANS, but any such meaning is not defined by this model. Contrast with OPEN.

COMMUNICATION ADDRESS: consists of COMMUNICATION MEANS and CONTACT ADDRESS.

COMMUNICATION MEANS: indicates a method whereby communication can take place. INSTANT MESSAGE SERVICE is one example of a COMMUNICATION MEANS.

CONTACT ADDRESS: a specific point of contact via some COMMUNICATION MEANS. When using an INSTANT MESSAGE SERVICE, the CONTACT ADDRESS is an INSTANT INBOX ADDRESS.

DELIVERY RULES: constraints on how an INSTANT MESSAGE SERVICE delivers received INSTANT MESSAGES to INSTANT INBOXES. For each INSTANT INBOX, the applicable DELIVERY RULES are manipulated by the INBOX USER AGENT of a PRINCIPAL that controls the INSTANT INBOX.

Motivation: We need a way of talking about filtering instant messages.

FETCHER: a form of WATCHER that has asked the PRESENCE SERVICE to for the PRESENCE INFORMATION of one or more PRESENTITIES, but has not asked for a SUBSCRIPTION to be created.

INBOX USER AGENT: means for a PRINCIPAL to manipulate zero or more INSTANT INBOXES controlled by that PRINCIPAL.

Motivation: This is intended to isolate the core functionality of an INSTANT INBOX from how it might appear to be manipulated by a product. This manipulation includes fetching messages, deleting messages, and setting DELIVERY RULES. We deliberately take no position on whether the INBOX USER AGENT, INSTANT INBOX, and INSTANT MESSAGE SERVICE are colocated or distributed across machines.

INSTANT INBOX: receptacle for INSTANT MESSAGES intended to be read by the INSTANT INBOX's PRINCIPAL.

INSTANT INBOX ADDRESS: indicates whether and how the PRESENTITY's PRINCIPAL can receive an INSTANT MESSAGE in an INSTANT INBOX. The STATUS and INSTANT INBOX ADDRESS information are sufficient to determine whether the PRINCIPAL appears ready to accept the INSTANT MESSAGE.

Motivation: The definition is pretty loose about exactly how any of this works, even leaving open the possibility of reusing parts of the email infrastructure for instant messaging.

INSTANT MESSAGE: an identifiable unit of data, of small size, to be sent to an INSTANT INBOX.

Motivation: We do not define "small" but we seek in this definition to avoid the possibility of transporting an arbitrary-length stream labelled as an "instant message."

INSTANT MESSAGE PROTOCOL: The messages that can be exchanged between a SENDER USER AGENT and an INSTANT MESSAGE SERVICE, or between an INSTANT MESSAGE SERVICE and an INSTANT INBOX.

INSTANT MESSAGE SERVICE: accepts and delivers INSTANT MESSAGES.

-- May require authentication of SENDER USER AGENTS and/or INSTANT INBOXES.

-- May have different authentication requirements for different INSTANT INBOXES, and may also have different authentication requirements for different INSTANT INBOXES controlled by a

single PRINCIPAL.

-- May have an internal structure involving multiple SERVERS and/or PROXIES. There may be complex patterns of redirection and/or proxying while retaining logical connectivity to a single INSTANT MESSAGE SERVICE. Note that an INSTANT MESSAGE SERVICE does not require having a distinct SERVER -- the service may be implemented as direct communication between SENDER and INSTANT INBOX.

-- May have an internal structure involving other INSTANT MESSAGE SERVICES, which may be independently accessible in their own right as well as being reachable through the initial INSTANT MESSAGE SERVICE.

SENDER USER AGENT: means for a PRINCIPAL to manipulate zero or more SENDERS.

NOTIFICATION: a message sent from the PRESENCE SERVICE to a SUBSCRIBER when there is a change in the PRESENCE INFORMATION of some PRESENTITY of interest, as recorded in one or more SUBSCRIPTIONS.

Motivation: We deliberately take no position on what part of the changed information is included in a NOTIFICATION.

OPEN: a distinguished value of the STATUS marker. In the context of INSTANT MESSAGES, this value means that the associated INSTANT INBOX ADDRESS, if any, corresponds to an INSTANT INBOX that is ready to accept an INSTANT MESSAGE. This value may have an analogous meaning for other COMMUNICATION MEANS, but any such meaning is not defined by this model. Contrast with CLOSED.

OTHER PRESENCE MARKUP: any additional information included in the PRESENCE INFORMATION of a PRESENTITY. The model does not define this further.

POLLER: a FETCHER that requests PRESENCE INFORMATION on a regular basis.

PRESENCE INFORMATION: consists of one or more PRESENCE TUPLES.

PRESENCE PROTOCOL: The messages that can be exchanged between a PRESENTITY and a PRESENCE SERVICE, or a WATCHER and a PRESENCE SERVICE.

PRESENCE SERVICE: accepts, stores, and distributes PRESENCE INFORMATION.

-- May require authentication of PRESENTITIES, and/or WATCHERS, and/or INSTANT INBOXES.

-- May have different authentication requirements for different

PRESENTITIES.

-- May have different authentication requirements for different WATCHERS, and may also have different authentication requirements for different PRESENTITIES being watched by a single WATCHER.

-- May have an internal structure involving multiple SERVERS and/or PROXIES. There may be complex patterns of redirection and/or proxying while retaining logical connectivity to a single PRESENCE SERVICE. Note that a PRESENCE SERVICE does not require having a distinct SERVER -- the service may be implemented as direct communication among PRESENTITY and WATCHERS.

-- May have an internal structure involving other PRESENCE SERVICES, which may be independently accessible in their own right as well as being reachable through the initial PRESENCE SERVICE.

PRESENCE TUPLE: consists of a STATUS, an optional COMMUNICATION ADDRESS, and optional OTHER PRESENCE MARKUP.

PRESENCE USER AGENT: means for a PRINCIPAL to manipulate zero or more PRESENTITIES.

Motivation: This is essentially a "model/view" distinction: the

PRESENTITY is the model of the presence being exposed, and is independent of its manifestation in any user interface. In addition, we deliberately take no position on how the PRESENCE USER AGENT, PRESENTITY, and PRESENCE SERVICE are colocated or distributed across machines.

PRESENTITY (presence entity): provides PRESENCE INFORMATION to a PRESENCE SERVICE.

Motivation: We don't like to coin new words, but "presentity" seemed worthwhile so as to have an unambiguous term for the entity of interest to a presence service. Note that the presentity is not (usually) located in the presence service: the presence service only has a recent version of the presentity's presence information. The presentity initiates changes in the presence information to be distributed by the presence service.

PRINCIPAL: human, program, or collection of humans and/or programs that chooses to appear to the PRESENCE SERVICE as a single actor, distinct from all other PRINCIPALS.

Motivation: We need a clear notion of the actors outside the system. "Principal" seems as good a term as any.

PROXY: a SERVER that communicates PRESENCE INFORMATION, INSTANT

MESSAGES, SUBSCRIPTIONS and/or NOTIFICATIONS to another SERVER. Sometimes a PROXY acts on behalf of a PRESENTITY, WATCHER, or INSTANT INBOX.

SENDER: source of INSTANT MESSAGES to be delivered by the INSTANT MESSAGE SERVICE.

SERVER: an indivisible unit of a PRESENCE SERVICE or INSTANT MESSAGE SERVICE.

SPAM: unwanted INSTANT MESSAGES.

SPOOFING: a PRINCIPAL improperly imitating another PRINCIPAL.

STALKING: using PRESENCE INFORMATION to infer the whereabouts of a PRINCIPAL, especially for malicious or illegal purposes.

STATUS: a distinguished part of the PRESENCE INFORMATION of a PRESENTITY. STATUS has at least the mutually-exclusive values OPEN and CLOSED, which have meaning for the acceptance of INSTANT MESSAGES, and may have meaning for other COMMUNICATION MEANS. There may be other values of STATUS that do not imply anything about INSTANT MESSAGE acceptance. These other values of STATUS may be combined with OPEN and CLOSED or they may be mutually-exclusive with those values.

Some implementations may combine STATUS with other entities. For example, an implementation might make an INSTANT INBOX ADDRESS visible only when the INSTANT INBOX can accept an INSTANT MESSAGE. Then, the existence of an INSTANT INBOX ADDRESS implies OPEN, while its absence implies CLOSED.

SUBSCRIBER: a form of WATCHER that has asked the PRESENCE SERVICE to notify it immediately of changes in the PRESENCE INFORMATION of one or more PRESENTITIES.

SUBSCRIPTION: the information kept by the PRESENCE SERVICE about a SUBSCRIBER's request to be notified of changes in the PRESENCE INFORMATION of one or more PRESENTITIES.

VISIBILITY RULES: constraints on how a PRESENCE SERVICE makes WATCHER INFORMATION available to WATCHERS. For each WATCHER's WATCHER INFORMATION, the applicable VISIBILITY RULES are manipulated by the WATCHER USER AGENT of a PRINCIPAL that controls the WATCHER.

Motivation: We need a way of talking about hiding watcher information from people.

WATCHER: requests PRESENCE INFORMATION about a PRESENTITY, or WATCHER INFORMATION about a WATCHER, from the PRESENCE SERVICE. Special types of WATCHER are FETCHER, POLLER, and SUBSCRIBER.

WATCHER INFORMATION: information about WATCHERS that have received PRESENCE INFORMATION about a particular PRESENTITY within a particular recent span of time. WATCHER INFORMATION is maintained by the PRESENCE SERVICE, which may choose to present it in the same form as PRESENCE INFORMATION; that is, the service may choose to make WATCHERS look like a special form of PRESENTITY.

Motivation: If a PRESENTITY wants to know who knows about it, it is not enough to examine only information about SUBSCRIPTIONS. A WATCHER might repeatedly fetch information without ever subscribing. Alternately, a WATCHER might repeatedly subscribe, then cancel the SUBSCRIPTION. Such WATCHERS should be visible to the PRESENTITY if the PRESENCE SERVICE offers WATCHER INFORMATION, but will not be appropriately visible if the WATCHER INFORMATION includes only SUBSCRIPTIONS.

WATCHER USER AGENT: means for a PRINCIPAL to manipulate zero or more WATCHERS controlled by that PRINCIPAL.

Motivation: As with PRESENCE USER AGENT and PRESENTITY, the distinction here is intended to isolate the core functionality of a

WATCHER from how it might appear to be manipulated by a product. As previously, we deliberately take no position on whether the WATCHER USER AGENT, WATCHER, and PRESENCE SERVICE are colocated or distributed across machines.

[5. Security Considerations](#)

This document provides a model and vocabulary for systems with certain intrinsic security issues. In particular, presence and instant messaging systems must deal with "the three S's": STALKING, SPOOFING, and SPAM. ACCESS RULES, VISIBILITY RULES, and WATCHER INFORMATION are intended to deal with STALKING. The several kinds of authentication mentioned for INSTANT MESSAGE SERVICE and PRESENCE SERVICE are intended to deal with SPOOFING. DELIVERY RULES are intended to deal with SPAM.

[6. Conclusion](#)

This document has provided a model for a presence and instant messaging system. The purpose of the model is to provide a common vocabulary for the further work of defining and implementing interoperable presence and instant messaging protocols.

[7. Acknowledgements](#)

This document has been improved by comments from Jesse Vincent and Colin Benson, by the participants in the Cambridge, MA meeting on June 11, 1999, and by Roy Salisbury, who contributed the original version

of Figure 5. The authors gratefully acknowledge their assistance.

8. References

[ENVY] M. Day. 'HTTP Envy' and Presence Information Protocols.
Internet-Draft [draft-day-envy-00.txt](#)

[PIPR] M. Calsyn, L. Dusseault. Presence Information Protocol
Requirements. Internet-Draft [draft-dusseault-pipr-00.txt](#)

[PRESENCE] V. Saraswat, J. Malcolm, C. Apple. The Presence
Protocol. Internet-Draft [draft-saraswat-presenceprotocol-00.txt](#)

[RPIM] M. Day. Requirements for Presence and Instant Messaging.
Internet-Draft [draft-day-rpim-00.txt](#)

[RVP-ADDR] L. Dusseault, G. Mohr. Addressing and Location for
RVP. Internet-Draft [draft-dusseault-rvp-addr-00.txt](#)

[RVP] M. Calsyn, L. Dusseault, G. Mohr. RVP: A Presence Notification
Protocol. Internet-Draft [draft-calsyn-rvp-01.txt](#)

[SIP-PIP] J. Rosenberg, H. Schulzrinne. SIP For Presence.
Internet-Draft [draft-rosenberg-sip-pip-00.txt](#)

[WHODP] G. Mohr. WhoDP: Widely Hosted Object Data Protocol.
Internet-Draft [draft-mohr-whodp-00.txt](#)

9. Authors' Addresses

Mark Day
Lotus Development Corporation
[55](#) Cambridge Parkway
Cambridge, MA 02142
USA
email: Mark_Day@lotus.com

Jonathan Rosenberg
Lucent Technologies, Bell Laboratories
Rm. 4C-526
[101](#) Crawfords Corner Rd.
Holmdel, NJ 07733
USA
email: jdrosen@bell-labs.com

Hiroyasu Sugano
Fujitsu Laboratories Ltd.
[64](#) Nishiwaki, Ohkubo-cho
Akashi 674-8555

Japan

email: suga@flab.fujitsu.co.jp