

INTERNET-DRAFT
Expires: June 2, 2000

Greg Hudson
ghudson@mit.edu
MIT

Instant Messaging / Presence Architecture
[draft-hudson-impp-arch-01.txt](#)

1. 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>.

Please send comments to the IMPP working group at impp@iastate.edu.

2. Abstract

This document proposes an architecture for the protocols to be produced by the IMPP working group.

3. Terminology

The following terms are defined in [[Model](#)] and are used with those definitions in this document:

ACCESS RULES
FETCH
INSTANT INBOX
INSTANT MESSAGE
INSTANT MESSAGE SERVICE
NOTIFICATIONS
PRESENCE INFORMATION
PRESENCE SERVICE
PRESENTITY
SENDER
SUBSCRIPTION
WATCHER
WATCHER INFORMATION

The following terms are defined in [[Reqts](#)] and are used with those definitions in this document:

DOMAIN
IDENTIFIER

The following terms are defined here:

An ENTITY is a PRESENTITY or WATCHER participating in the PRESENCE SERVICE, or a SENDER or INSTANT INBOX participating in the INSTANT MESSAGE SERVICE.

A PRESENCE CLIENT is a network endpoint communicating as a PRESENTITY and WATCHER with the same IDENTIFIER. It may choose to exercise the functions of only one role.

An INSTANT MESSAGE CLIENT is a network endpoint communicating as a SENDER and INSTANT INBOX with the same IDENTIFIER. It may choose to exercise the functions of only one role.

A CLIENT is either a PRESENCE CLIENT or INSTANT MESSAGE CLIENT.

[XXX open issue: these definitions assume that presence and instant messaging will take place over separate wire protocols (separate TCP connections, separate well-known ports, etc.). If we decide they should take place over one wire protocol, it would make more sense to define just "CLIENT" which can communicate as any of the four.]

4. Basic Principles

Each ENTITY has an IDENTIFIER which contains a DOMAIN part and a locally assigned part. The DOMAIN part names a domain within the Domain Name Service [[RFC 1034](#)].

[XXX open issue: does an IDENTIFIER also contain a resource name or some other kind of text which does not affect the identity of the ENTITY? Does an IDENTIFIER contain a service name? These issues don't really affect the rest of the architecture but they certainly live above the actual wire protocol specifications and so belong in this document.]

Each DOMAIN has one or more SERVERS implementing that DOMAIN's part of the INSTANT MESSAGE SERVICE or PRESENCE SERVICE. If a DOMAIN has more than one SERVER for one of the two services, a party communicating with that domain will pick one of the SERVERS and talk to it; the collection of SERVERS must ensure that the other party achieves the same result communicating with any SERVER. The protocol for communication between SERVERS within a DOMAIN is not specified by the IMPP suite.

[XXX open issue: Is server synchronization within a domain our

problem, or do we let the creators of server farms develop their own algorithms? Does the same server-server protocol for inter-domain traffic also apply to servers within a single domain? There are some mildly hard issues not covered by the inter-domain server-server protocol, e.g. if there are a hundred servers for my domain and you don't connect to the one which has my client connection IMs, how does the other server keep track of where to forward your IM request to? I think it's not our problem, personally; it's not important that I be able to put together a server farm consisting of servers from multiple different vendors, so if different vendors' server farms synchronize using proprietary protocols, that's not a big deal. And since synchronization is a tricky issue, perhaps it's better to let people innovate rather than trying to solve it within a standards body.]

A CLIENT speaks to one of the SERVERS for the DOMAIN part of its ENTITIES' IDENTIFIER. The SERVER may establish the authenticity of the CLIENT connection to the local policy of the DOMAIN and will forward requests to and from the SERVERS of other DOMAINS as necessary. Thus, a request such as an instant message transmission which travels from CLIENT 1 in DOMAIN X to CLIENT 2 in a different DOMAIN Y will travel as follows:

```
CLIENT 1 -> DOMAIN X SERVER
           |
           v
CLIENT 2 <- DOMAIN Y SERVER
```

If CLIENT 1 and CLIENT 2 are in the same domain, the communication path is simpler:

```
CLIENT 1 -> DOMAIN X SERVER -> CLIENT 2
```

If a DOMAIN has multiple servers, it may be necessary to forward some requests between servers within a domain. Such forwarding is not shown in the diagrams above because it is not in the scope of this architecture. [XXX affected by the previous open issue about server synchronization within a domain.]

NOTE: The protocol suite might be more flexible if it specified ways for CLIENT A to communicate directly with SERVER B or with CLIENT B. But the rigid arrangement has three key advantages:

- * A SERVER never has to worry about authenticating ENTITIES from other DOMAINS. So each DOMAIN can use its own authentication infrastructure without concerning itself with the authentication infrastructure of other DOMAINS.
- * If many users from DOMAIN A are communicating with resources in DOMAIN B, DOMAIN B only needs to

maintain one security context for all of DOMAIN A and the setup of that security context may not need to occur as often as it would if DOMAIN A's users communicated directly with DOMAIN B.

- * A SERVER may act as a gateway to a different messaging service by using a nonstandard protocol to communicate with CLIENTS, and this will be transparent to CLIENTS from other DOMAINS.

Most scenarios which call for direct communication (e.g. several laptops communicating over an ad-hoc private network) can be accommodated by running a SERVER on the same machine as one or more of the CLIENTS. It may be necessary to use different IDENTIFIERS than one ordinarily would in this kind of setup.

5. Protocols

The IMPP suite consists of two applications: instant messaging and presence. The basic architecture divides communications into two categories: communication between a CLIENT and a SERVER for its DOMAIN and communication between SERVERS for two different DOMAINS. There are therefore four different wire protocols in the suite.

[XXX open issue: are presence and instant messaging two different protocols or the same protocol using different methods? Are client-server and server-server communication different two different protocols or the same protocol operating in two different modes? How do we slice this pie? For now I'm assuming that we slice it finely, because it's easier to combine than to separate.]

[XXX open issue: there is also communication between SERVERS within the same DOMAIN, if we are going to tackle this issue, and potentially communication involving proxies, something we haven't discussed in much detail.]

5.1. Client-Server Presence Transfer Protocol

This protocol communicates presence and related data between a PRESENCE CLIENT and a SERVER for its DOMAIN. The following kinds of data may be transferred through this protocol:

- * The PRESENTITY can publish PRESENCE INFORMATION and ACCESS RULES to the SERVER.
- * The WATCHER can FETCH or SUBSCRIBE to PRESENCE INFORMATION for another PRESENTITY, and receive NOTIFICATIONS for PRESENTITIES it has a SUBSCRIPTION to.
- * The WATCHER can request WATCHER INFORMATION about the PRESENTITY.

5.2. Server-Server Presence Transfer Protocol

This protocol allows a SERVER to forward presence and related data from ENTITIES within its DOMAIN to a SERVER for another DOMAIN. Since publishing PRESENCE INFORMATION and retrieving WATCHER INFORMATION occur inside a domain, only one kind of data may be transferred through this protocol:

* A FETCH or SUBSCRIPTION request or a NOTIFICATION may be forwarded from one SERVER to another.

[XXX open issue: several people have suggested that the server-server protocol should also be used for synchronization within a server farm or for proxy servers. In this case, the full range of presence operations would need to be supported.]

5.3. Client-Server Instant Message Transfer Protocol

This protocol allows an INSTANT MESSAGE CLIENT to transfer INSTANT MESSAGES from and to a SERVER for its DOMAIN.

5.4. Server-Server Instant Message Transfer Protocol

This protocol allows a SERVER for one DOMAIN to transfer INSTANT MESSAGES on behalf of one of its SENDERS to a SERVER for another DOMAIN, for delivery to an INSTANT INBOX in the other DOMAIN.

6. References

[RFC 1034]

P. V. Mockapetris. "Domain names - concepts and facilities." RFC 1034, November 1987.

[Model]

M. Day, J. Rosenberg, H. Sugano. "A Model for Presence." Work in progress, [draft-ietf-imp-model-03.txt](#).

[Reqs]

M. Day, S. Aggarwal, G. Mohr, J. Vincent. "Instant Message / Presence Protocol Requirements." Work in progress, [draft-ietf-imp-reqs-03.txt](#).