

Workgroup:

More Instant Messaging Interoperability

Internet-Draft: draft-barnes-mimi-arch-02

Published: 23 October 2023

Intended Status: Informational

Expires: 25 April 2024

Authors: R. L. Barnes

Cisco

An Architecture for More Instant Messaging Interoperability (MIMI)

Abstract

The More Instant Messaging Interoperability (MIMI) working group is defining a suite of protocols that allow messaging providers to interoperate with one another. This document lays out an overall architecture enumerating the MIMI protocols and how they work together to enable an overall messaging experience.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-barnes-mimi-arch/>.

Discussion of this document takes place on the More Instant Messaging Interoperability Working Group mailing list (<mailto:mimi@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/mimi/>. Subscribe at <https://www.ietf.org/mailman/listinfo/mimi/>.

Source for this draft and an issue tracker can be found at <https://github.com/bifurcation/mimi-arch>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on 25 April 2024.

Copyright Notice

Copyright (c) 2023 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

- [1. Introduction](#)
- [2. Conventions and Definitions](#)
- [3. Overall Scope](#)
- [4. Room State](#)
 - [4.1. End-to-End Security State](#)
 - [4.2. Participants and Members](#)
 - [4.3. Membership Changes](#)
 - [4.4. Policy](#)
- [5. Protocols](#)
 - [5.1. Events and Transport](#)
 - [5.2. Control Protocols](#)
 - [5.3. Messages](#)
- [6. Actors, Identifiers, and Authentication](#)
- [7. Security Considerations](#)
- [8. IANA Considerations](#)
- [9. References](#)
 - [9.1. Normative References](#)
 - [9.2. Informative References](#)
- [Acknowledgments](#)
- [Author's Address](#)

1. Introduction

Today, there are many providers of messaging functionality. A provider typically provides the client software (e.g., a mobile app) and the servers that facilitate communications among clients. The core function of MIMI is enabling users to have messaging interactions across message providers.

This overall goal breaks down into several sub-goals:

- *Message formats that enable the user-level features of a messaging system
- *Tracking of state across multiple providers
- *End-to-end security of user messages
- *Transport of protocol messages among providers

In this document, we describe the high-level functions of these protocols, and how they work together to enable an overall messaging application.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

Terms are generally introduced in context, indicated by *emphasis*.

3. Overall Scope

[Figure 1](#) shows the critical entities in the overall MIMI system and their interactions. Each human *user* is represented in the system by one or more *clients*, where each client is a specific software or hardware system belonging to a single user. Each provider is represented by a *server* (logically a single server, but possibly realized by multiple physical devices).

Messaging interactions are organized around *rooms*. All messaging interactions take place in the context of a room. (Some non-messaging interactions may take place outside of a room, such as operations to fetch information required to set up a room.) Rooms have a notion of *user participation* as well as *client membership*, both tracked as lists. Rooms additionally have policies about things like how the room may be joined and what capabilities each member/participant has.

The protocol interactions that drive a room unfold among the servers whose users are participants in the room. There is exactly one *hub* server for the room, which is in primary control of the room. All other servers are known as *followers*. Follower servers interact directly with the hub server. Interactions between clients occur indirectly, via the servers for the clients' providers.

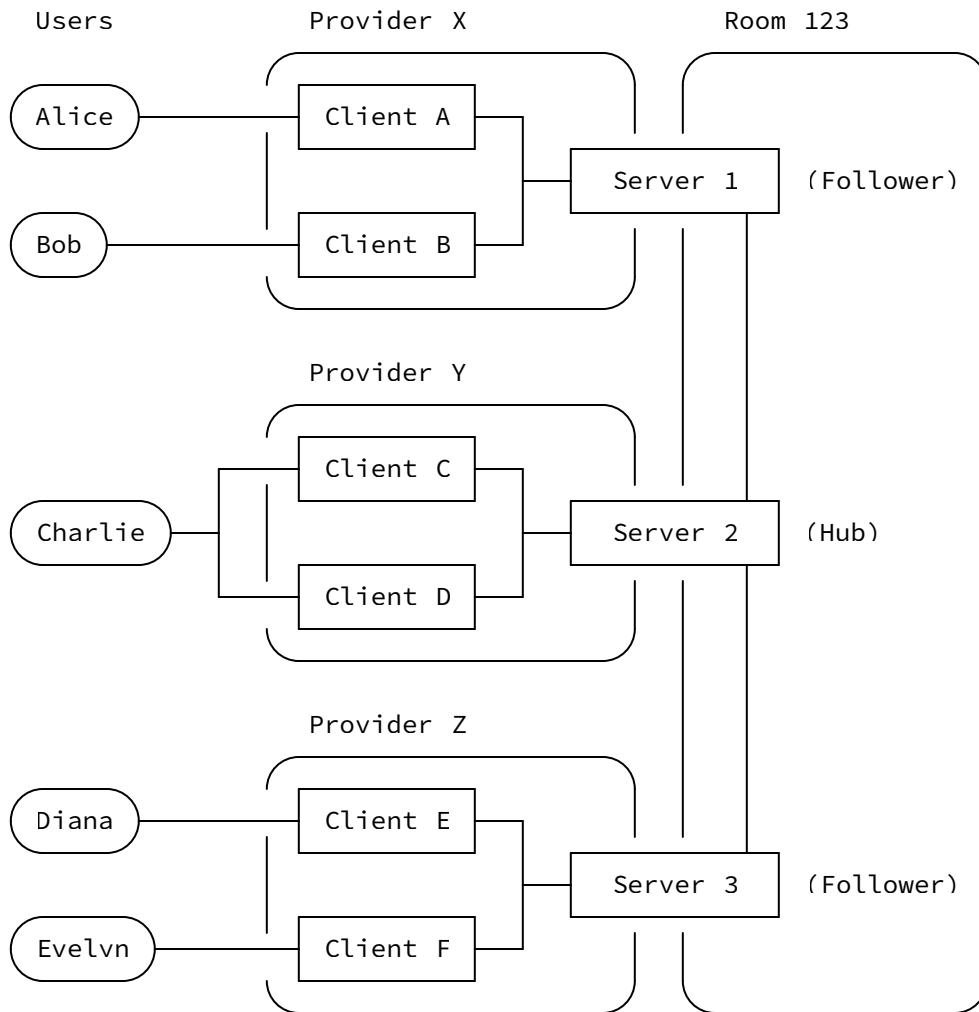


Figure 1: MIMI Entities and Interactions

4. Room State

A room represents a messaging interaction among a specific set of clients, with a single *state*. A major goal of the MIMI protocols is to synchronize the state of a room across all of the servers and clients participating in the room. Changes to the room's state can be proposed by either clients or servers, though as discussed in [Section 4.4](#), one important aspect of the room's state is an authorization policy that determines which actors are allowed to make which changes.

The creation of a room is a local operation on the hub server, and thus outside the scope of MIMI. The hub server establishes the initial state of the room.

The state of the room includes a few types of information, most importantly:

- *The end-to-end security state of the room
- *The user-level participation state of the room
- *The authorization policy for the room

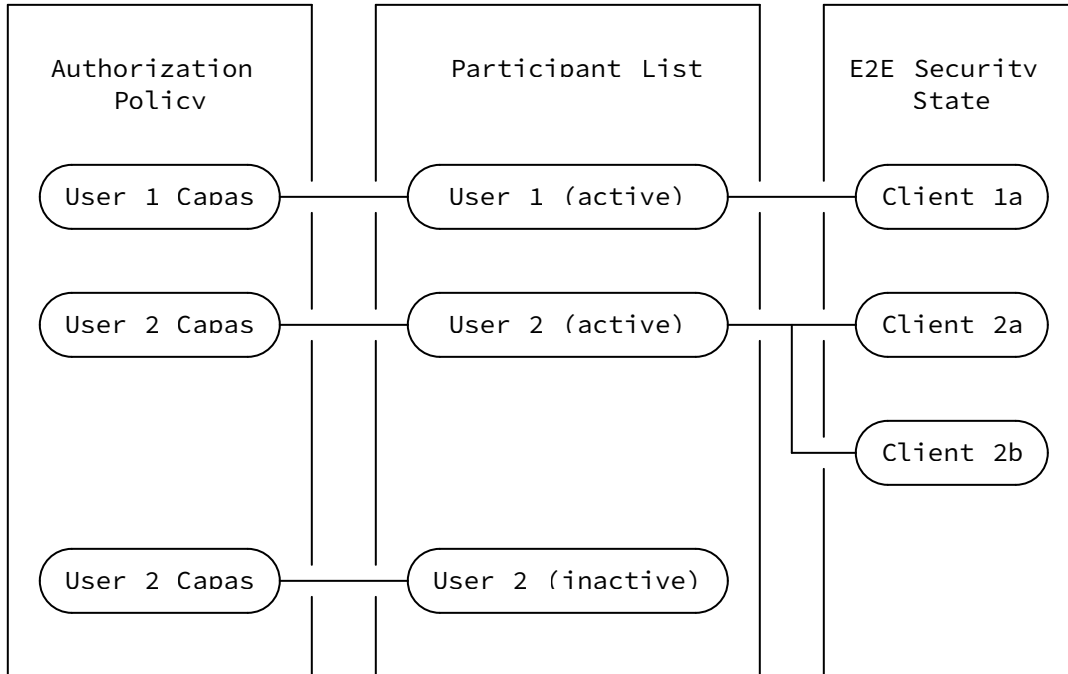


Figure 2: Elements of the Room State

4.1. End-to-End Security State

Messages sent within a room are protected by an end-to-end security protocol to ensure that the servers handling messages cannot inspect or tamper with messages. This means that the required cryptographic keys need to be provisioned to any client from which a user can interact with the room. The state of this end-to-end security protocol thus represents the precise set of clients that can send and receive messages in the room, the most precise notion of membership for a room. A client that has the required keys for end-to-end security is said to be a member of the end-to-end security state of the room.

The end-to-end security state of a room has public and private aspects. Servers may store the public aspects of the end-to-end security state, such as identities and credentials presented by the

clients in the room. The private aspects of the group, such as the symmetric encryption keys, are known only to the clients.

4.2. Participants and Members

The *participant list* for a room is the set of users who are allowed to interact with the room in some way. The specific list of ways in which a user may participate is defined by authorization policy, as discussed in [Section 4.4](#).

Note the parallel terminology with regard to inclusion of clients or users in the room:

*A *client* is a member of the end-to-end security state of the room

*A *user* is a *participant* in the room

The user-level *participant list* and the client-level *membership* of the room are distinct entities managed by separate protocols, but they must be consistent with each other. A client may be a member of the E2EE state of a room only if its user is a participant in the room. However, a user may be a participant in a room without any client belonging to the user being part of the end-to-end security state of the room. (Such a user will not be able to read or send messages, but may be able to take other actions. It is up to client implementations how this state is represented.)

A user with at least one client joined to the end-to-end security state of the room is known as an *active user*, since such a user can fully participate in the room.

4.3. Membership Changes

The participant list and client membership of a group can change over time, via *add* and *remove* operations at both the user level and the client level. These operations are independent at the protocol level: For example, a user may be added to a room before any of its clients are available to join, or a user may begin using a new device (adding the device without changing the user-level participation).

As discussed above, user-level participation and client-level membership must be kept in sync. When a user is added, some set of their clients should be added as well; when a user leaves or is evicted, any clients joined to the room should be removed. The cryptographic constraints of end-to-end security protocols mean that servers cannot perform this synchronization; it is up to clients to keep these two types of state in sync.

4.4. Policy

Each room has an associated *policy* that governs which protocol actions are authorized for the room while the policy is in effect. The policy defines several aspects of the room's behavior, for example:

*Admission policy: Do new members need to be explicitly added by a current member of the room, or can some set of users join unilaterally?

*Capabilities per user: Is a given user allowed to ...

- Send messages in the room?

- Add or remove other users?

- Grant or deny capabilities to other users?

*Capabilities per server: Is a given server participating in the room allowed to...

- Add or remove users?

- Grant or deny capabilities to users?

The hub server for a room defines the *policy envelope* for the room, the set of acceptable policies for the room. The hub also sets the initial policy for the room when it is created. Pursuant to that initial policy, the clients and servers participating in the room may then make further changes to the policy.

At any given time, all of the clients and servers have the same view of the room's policy. A client or server that receives an event that is not compliant with the room's policy may thus safely discard it, since all of the other participating clients/servers should also reject the event.

5. Protocols

As shown in [Figure 3](#), MIMI protocols define server-to-server interactions and client-to-client interactions. Each client interacts with the overall system by means of its provider's server (whether hub or follower). Client-to-client interactions are done by means of these servers.

The messages sent within a room are forwarded among participating clients by servers. However, messages are protected by an end-to-end security protocol so that their content is only accessible to the clients participating in the room. In addition to forwarding

messages, servers participate in control protocols that coordinate the state of the room across the participating providers. Both message forwarding and control protocols leverage a common framework for sharing *events* among servers.

Note that some parts of the overall system are explicitly out of scope for MIMI. Namely, client-server interactions internal to a provider (indicated by "(Provider)" in [Figure 3](#)) can be arranged however the provider likes.

A MIMI server thus participates in a few classes of protocols:

- *A transport protocol
- *Control protocols
- *A message forwarding protocol

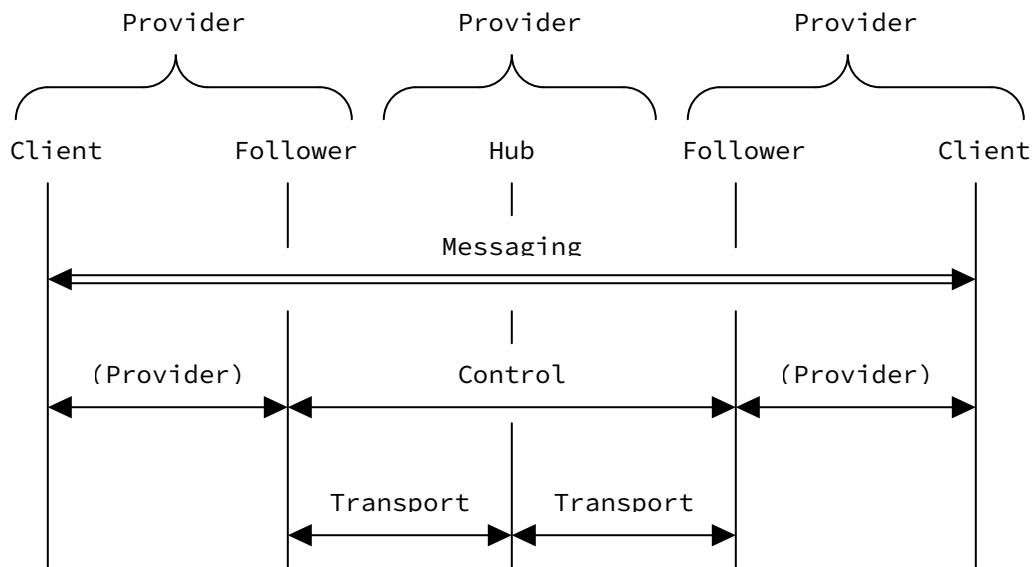


Figure 3: MIMI Protocols

5.1. Events and Transport

A room's activities are realized by servers exchanging *events*. Events come in two types:

- ***State events**, which make changes to the room state
- ***Message events**, which describe actual messaging activity in the room

Each event originates at one of the servers participating in the room (possibly as a result of some interaction with a client). The originating server sends the event to the hub server for the room, who distributes it to the other follower servers.

Each event is authenticated by its originating server so that all other participating servers can verify its origin, even those to whom the event has been distributed by the hub. If an event was ultimately created by a client, it is also authenticated by the client that created it.

The overall MIMI protocol defines this event framework, including its authentication scheme, as well as the mechanics of how events are delivered from one server to another.

5.2. Control Protocols

The servers involved in a room use control protocols to perform actions related to different types of information that comprise a room's state, particularly those listed in [Section 4](#). Because these types of information and the operations they require are largely orthogonal, it makes sense to have a separate control protocol for each type of information.

The **policy control protocol** distributes information about the policy envelope of a room, and allows participants in a room to propose changes to the policy within that envelope.

The **participation control protocol** manages the user-level membership of the room, including the various ways that members might join or leave a room (or be added/removed by other users).

The **end-to-end security control protocol** manages the end-to-end security state of the room. In addition to distributing messages that add or remove clients from the end-to-end security state, this protocol also allows servers to distribute cryptographic information that clients have pre-registered, which allows clients to be asynchronously added to rooms.

5.3. Messages

Message events are end-to-end secure objects that carry application messages in the standard MIMI content format. The end-to-end encapsulation ensures that the message content is only accessible to the clients participating in the room, not the servers that help to distribute it.

The MIMI message format [[I-D.ietf-mimi-content](#)] defines how clients achieve the various features of a messaging application, for example:

- *Text messaging
- *File attachments
- *Replies
- *Reactions
- *Initiation of real-time sessions

Messages transit MIMI servers by means of a **message forwarding protocol**, which carries an opaque, encrypted message payload together with enough metadata to facilitate delivery to the clients participating in a room.

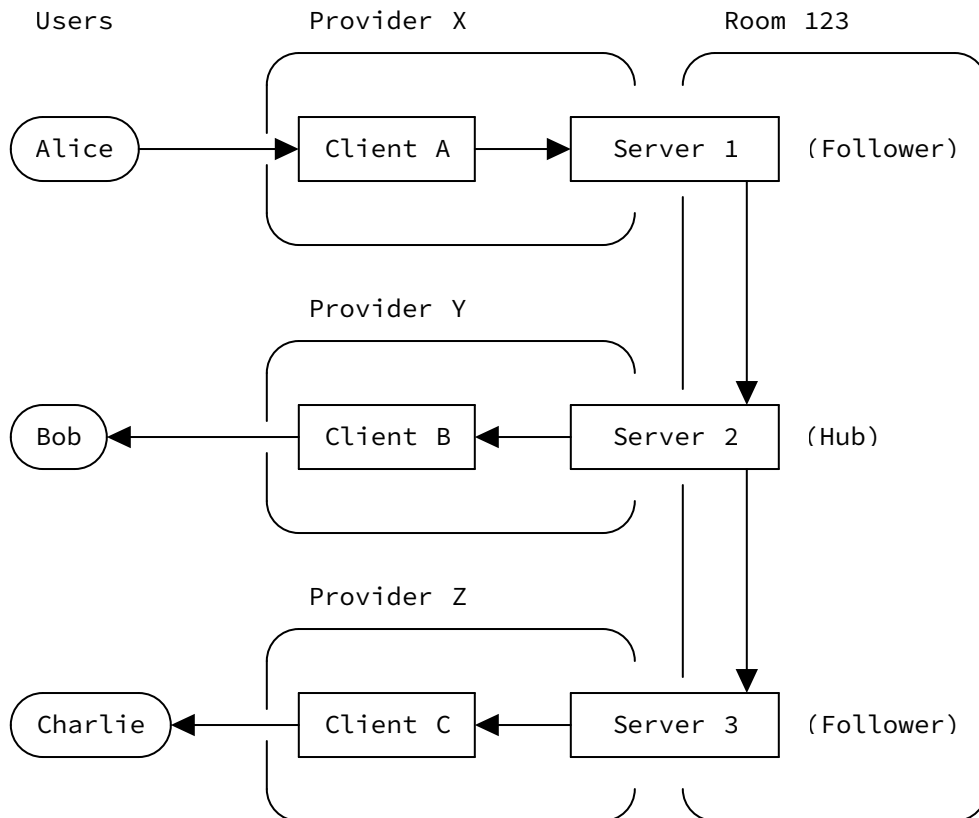


Figure 4: The hub fans out messages to participating servers; servers deliver messages to users' clients.

When a client sends a message, the message is delivered to its provider's server using some provider-internal mechanism. If the provider is not the hub, then the server forwards the message to the hub for delivery. In either case, the hub distributes the message to all of the servers participating in the room. Each provider's server then forwards the message to clients of users who are participating in the room.

6. Actors, Identifiers, and Authentication

There are several types of entity to be identified in the MIMI system, including:

- *Rooms,
- *Servers,
- *Users, and
- *Clients.

A server's identity is effectively the identity of the provider it represents. A room is hosted by a single hub server at a given time, so its identity is within the scope of the hub server's identity.

To facilitate the application of policies based on these identifiers to protocol actions, each actor presents one or more credentials that associate a signature key pair to their identifiers. Protocol messages are then signed by their senders to authenticate the origin of the message.

For a deeper discussion of identity, see [[I-D.mahy-mimi-identity](#)].

7. Security Considerations

TODO

- *Authorization policy attached to a room
- *E2E security for messages provided by message delivery protocol
- *E2E/E2M/M2E/M2M security for events provided by transport protocol
- *HbH security provided by TLS

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.

9.2. Informative References

- [I-D.ietf-mimi-content] Mahy, R., "More Instant Messaging Interoperability (MIMI) message content", Work in Progress, Internet-Draft, draft-ietf-mimi-content-00, 20 June 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-mimi-content-00>>.
- [I-D.mahy-mimi-identity] Mahy, R., "More Instant Messaging Interoperability (MIMI) Identity Concepts", Work in Progress, Internet-Draft, draft-mahy-mimi-identity-02, 10 July 2023, <<https://datatracker.ietf.org/doc/html/draft-mahy-mimi-identity-02>>.

Acknowledgments

TODO acknowledge.

Author's Address

Richard L. Barnes
Cisco

Email: rlb@ipv.sx