MIMI Delivery Service & Transport

draft-robert-mimi-delivery-service
draft-kohbrok-mimi-moh

IETF117, San Francisco
Raphael Robert & Konrad Kohbrok

What is MLS?

Things you know:

- Efficient E2EE in groups

Things you might not know:

- Group membership management
- Synchronize arbitrary state (through extensions)
- Server component
- Client authentication

Architecture

MLS has two dependencies:

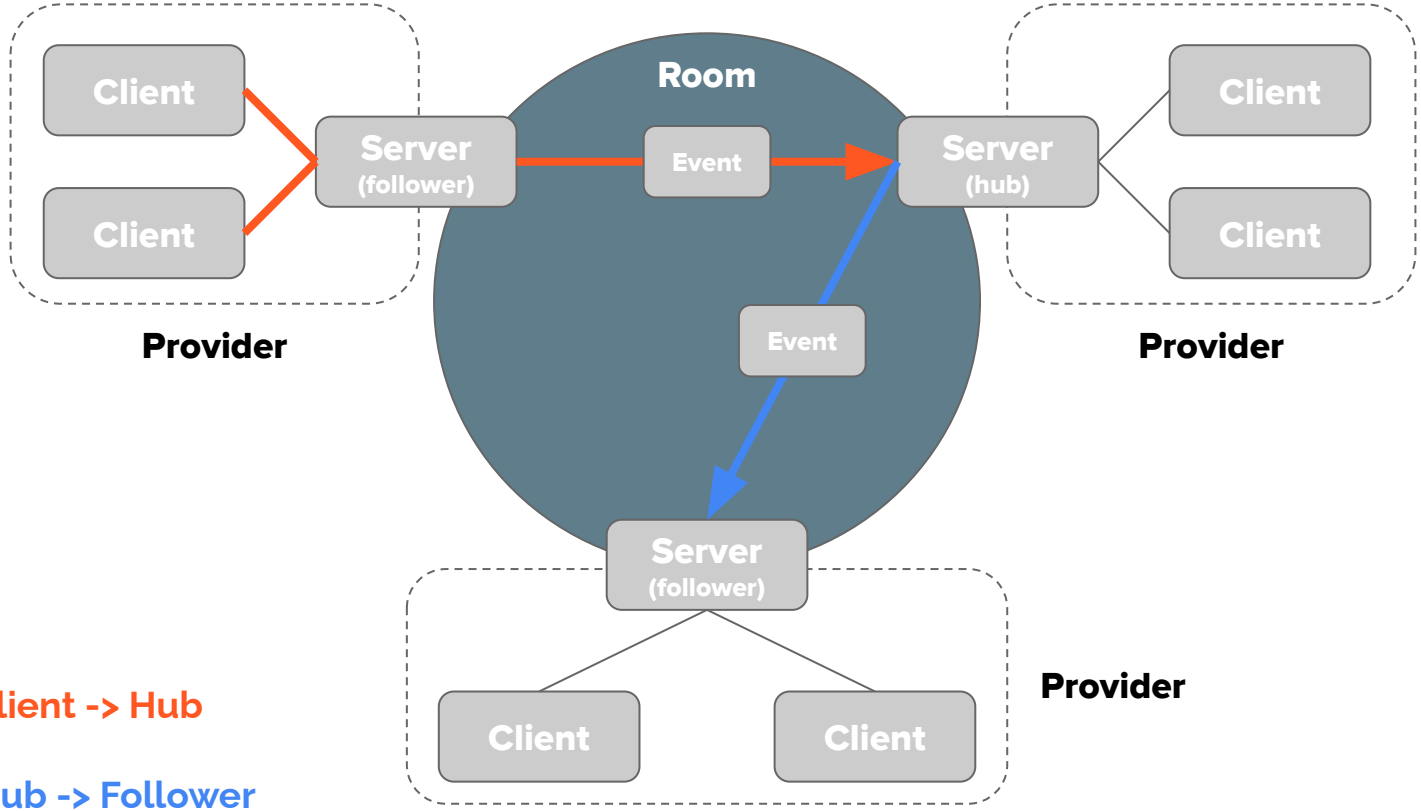
- Authentication Service (AS)
- **Delivery Service (DS)**

Architecture

The Delivery Service comes in two flavors:

- **Consistent and Partition-tolerant, or Strongly Consistent**
- Available and Partition-tolerant, or Eventually Consistent

– Delivery Service protocol



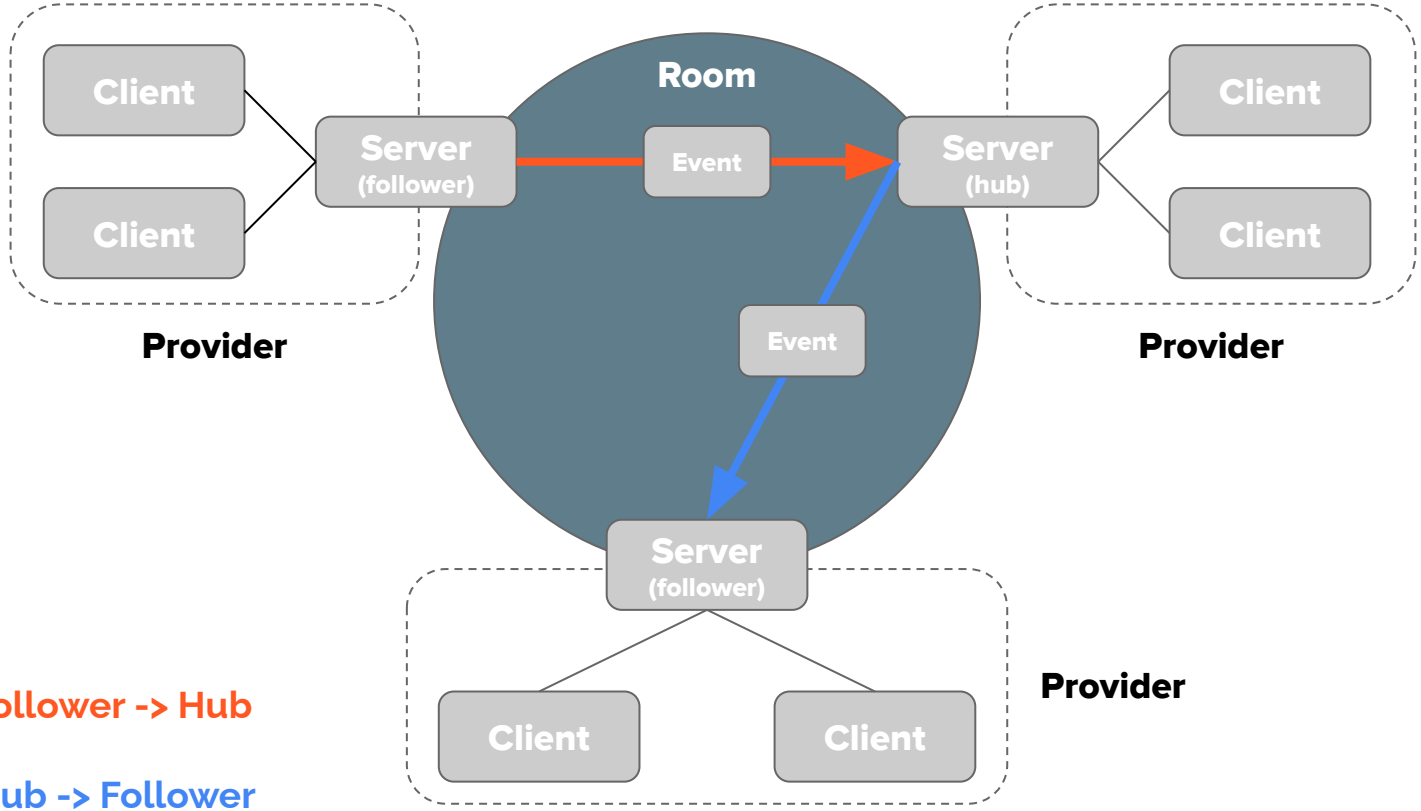
Leg 1: Client -> Hub

Leg 2: Hub -> Follower

Delivery Service scope

- `draft-robert-mimi-delivery-service` covers the leg between clients and the hub server, and between the hub server and the follower server
- It is independent of the underlying transport protocol (e.g. HTTP, XMPP, Matrix, etc.) as long as a request-response scheme is supported for Part A, uses TLS encoding

– Transport protocol



Leg 1: Follower -> Hub

Leg 2: Hub -> Follower

Protocol overview

Supports all operations that can be done with MLS:

- Send messages
- Inspect group membership
- Adding/removing members
- Updating key material
- Joining a group (via Welcome, External Commit, External Member Proposal or New Member Proposal)
- 1:1 connection requests
- Resync (when clients get corrupted)

Protocol capabilities

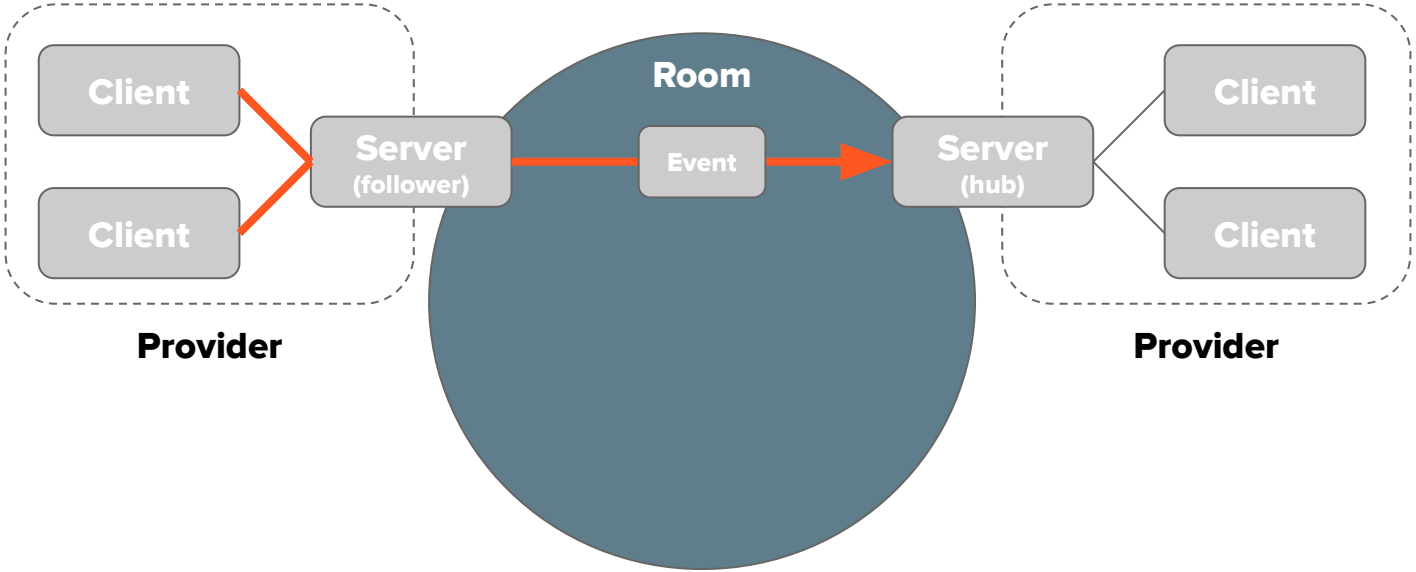
The protocol has the following capabilities:

- Multi-device support (more than one device per user)
- Can enforce policies server-side (e.g. from draft-mahy-mimi-group-chat)
- Additional Associated Data (AAD)
- Can do multi-level rate-limiting for abuse prevention & spam
- Distinguish between 1:1 connections and groups, incl. consent-based connection

Authentication

- Federated messaging means that messages cross domain boundaries
- Cross-domain traffic can be tricky, how do we do authentication?
- We can use the fact that clients have certificates for cross-domain authentication

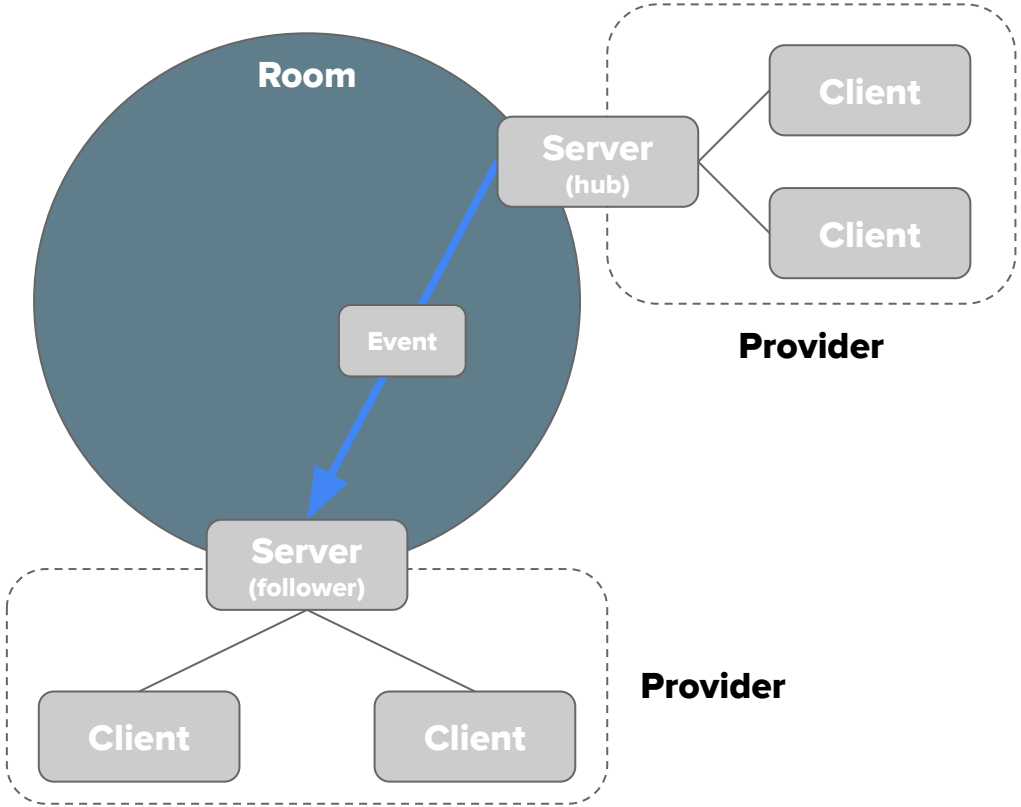
Authentication



1. Clients sign messages with their client certificate.
2. The hub can verify the signatures since it has the client certificates in the MLS ratchet tree.

Authentication

1. Clients can provision a credential for the fanout
2. The follower server only accepts those credentials.



State on the server-side component

The hub holds the following information for the protocol:

- A list of all groups it hosts
- For each group, it holds
  - The GroupInfo
  - The MLS ratchet tree
  - Information needed for per-client/per-user authentication and message fanout
- KeyPackages for
  - 1:1 connections
  - Groups

Server-side validation & assistance

- The server can inspect and validate all messages, since we use MLS public messages
- By inspecting MLS control messages, the server can update its copy of the ratcheting tree, clients can save bandwidth
- The server can help new clients join the group by making the group state available to them

Requirements

- Satisfies most of the requirements from draft-mahy-mimi-transport-design-reqs
- The missing ones can be added, but draft-mahy-mimi-transport-design-reqs needs discussion first

RM: Reduced metadata variant

- Objective: Reduce sensitive metadata (such as identifiers) as much as possible
- Specifically, the DS component on the hub should not have to store sensitive metadata
- Reference target: The Signal service
- Ideally, observable metadata is also reduced

RM: Non-goals

- Hide external identifiers such as IP addresses
- Prevent statistical and behavioral analysis

RM: Scope

- The protocol supports nearly the same functionality
- The protocol supports the same security mechanisms
- The syntax of the messages mostly similar, with slight differences
- Right now both variants are described in the same document

RM: Mechanisms

The state on the DS cannot contain sensitive metadata:

- The credentials/identifiers are replaced by pseudonyms
- Multi-layered at-rest-encryption
- Part of the rate-limiting is enforced by using Privacy Pass

RM: Downsides (for now)

- Conceptually new technology
- Only large-scale deployment is Signal (but with different technologies)
- Might not support all group types/features
- Has requirements for the transport protocol
- Has requirements regarding certificates
- More complexity in the design
- More complexity in the implementations
- Requires more analysis
- Trickier to deploy

Path forward

- RM most likely has a longer timeline
- Untangle the two variants: Extract RM and start a new document: `mimi-ds` vs `mimi-ds-rm`
- Iterate on `mimi-ds` and port changes to `mimi-ds-rm` where possible

– Stack

Message Content	draft-ietf-mimi-content
MLS	ietf-mls-protocol
Group policies	draft-mahy-mimi-group-chat
Delivery Service	draft-robert-mimi-delivery-service
Transport	draft-kohbrok-mimi-moh

– Implementation

Message Content	draft-ietf-mimi-content	**Partial**
MLS	ietf-mls-protocol	**Full**
Group policies	draft-mahy-mimi-group-chat	**-**
Delivery Service	draft-robert-mimi-delivery-service	**Full**
Transport	draft-kohbrok-mimi-moh	**Full**