MIMI Design Team Report

IETF 118

Travis Ralston & Konrad Kohbrok

Agenda

Agreed-upon previously:

- Signaling should be crypto-agnostic to allow on-ramping
- As few documents as possible to ease readability
- Alice - Bob flow as presented previously

Today:

- Document structure
- Changes since last DT report

Design Team Proposal

- four documents
  - MIMI Architecture (non-normative, gives overview over protocol architecture, terminology and other documents)
  - MIMI Message Content (already adopted, details format of content messages)
  - MIMI Delivery service (details how MIMI uses MLS, i.e. the group level operations, as well as the guarantees it provides)
  - MIMI Protocol (depends on MIMI Delivery Service and MIMI Message Content, details the signaling, i.e. the room level operations)

MIMI Architecture

- Terminology
- Protocol overview
- Protocol architecture
- Documents overview

MIMI Delivery Service

- Goal: specify a relatively generic MLS delivery service
- New “Interface” section with overview over capabilities
  - Ordering of handshake messages (MLS requirement)
  - Membership management via Proposals, also by non-members
  - Proposal-commit logic, everyone can propose, only clients can commit
  - MLS-specific verification of messages (including authentication)
  - Tracking of public group state (including membership list)
  - Assistance for joiners (download GroupInfo for external joiners)
  - Download of KeyPackages

MIMI Delivery Service cont'd

- Removed specialized Add/Remove/Update operations
- Now: Propose and Commit operations
- Simpler interface for use by MIMI Protocol

MIMI Protocol

- Room level operations
  - Signaling based on events
  - Room state changes (currently only participant list changes) based on MLS proposals
  - Signaling proposals take immediate effect on room state (not upon commit)
- Makes use of MIMI DS
  - Commits anchor room state with MLS group, as they include signaling proposals
  - Signaling and MLS based proposals can be sent as part of one commit to allow atomic operations

MIMI Protocol cont'd

Events

- m.room.user: change participant list (via MLS proposal)
- m.room.info: get room info
- ds.proposal: send MLS proposal(s)
- ds.commit: send MLS commit(s)
- ds.send_message: send MLS application message(s)
- ds.fetch_key_package: fetch MLS KeyPackage
- ds.fetch_group_info: fetch MLS GroupInfo

MIMI Protocol document structure

- Abstract + Intro
- Example flow: Alice adds Bob
- Framing
- Rooms and events
  - Room state
  - Event schema
  - Cryptographic state anchoring
- User participation
  - Participation states
  - Invite/add/leave/join/etc flows
- Transport

Alice adds Bob example flow

Assumptions

- Any discovery of Bob's service-specific identifier has already happened
- Alice has any necessary consent to add Bob to a room
- There may be other join flows; this is just one example

Step 1: Alice creates group/room

Alice creates the MLS group and associated room within their messaging provider, independent of MIMI.

1. Alice's server becomes the Hub server for the room
2. Initial room state covers participant list (just Alice), policy, etc.

Not a MIMI protocol operation, but initializes room state that drives the protocol later.

Alice fetches Bob's KeyPackages

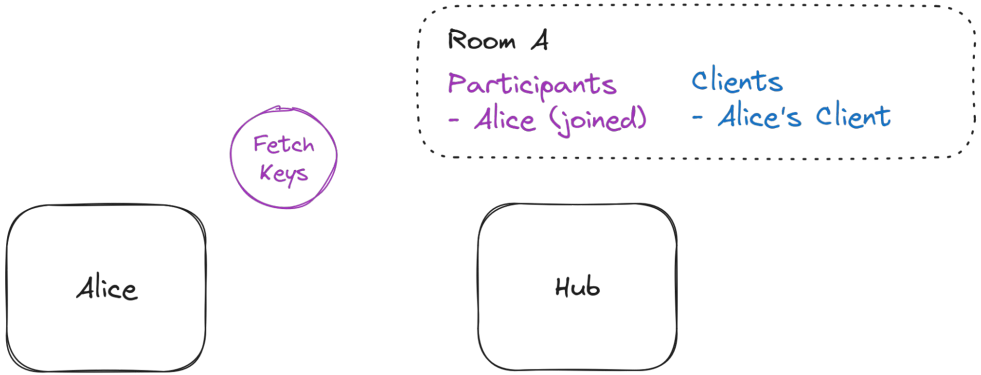
Legend:

Signalling

DS

Transport

Policy



Fetch Keys = ds.fetch_key_packages

Alice fetches Bob's KeyPackages

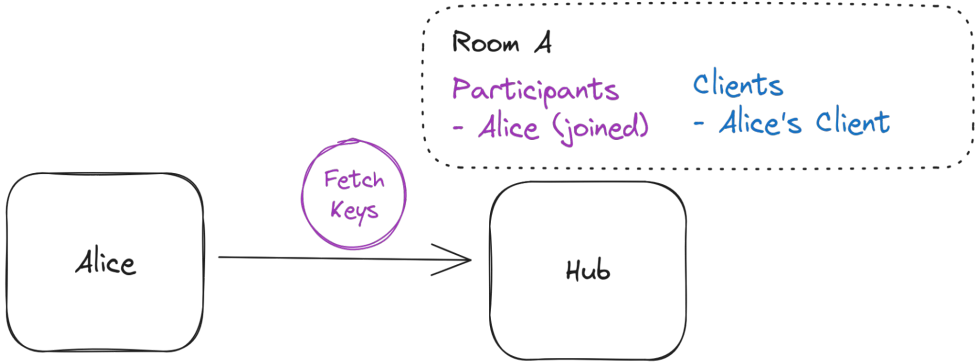
Legend:

Signalling

DS

Transport

Policy



Fetch Keys = ds.fetch_key_packages

Alice fetches Bob's KeyPackages

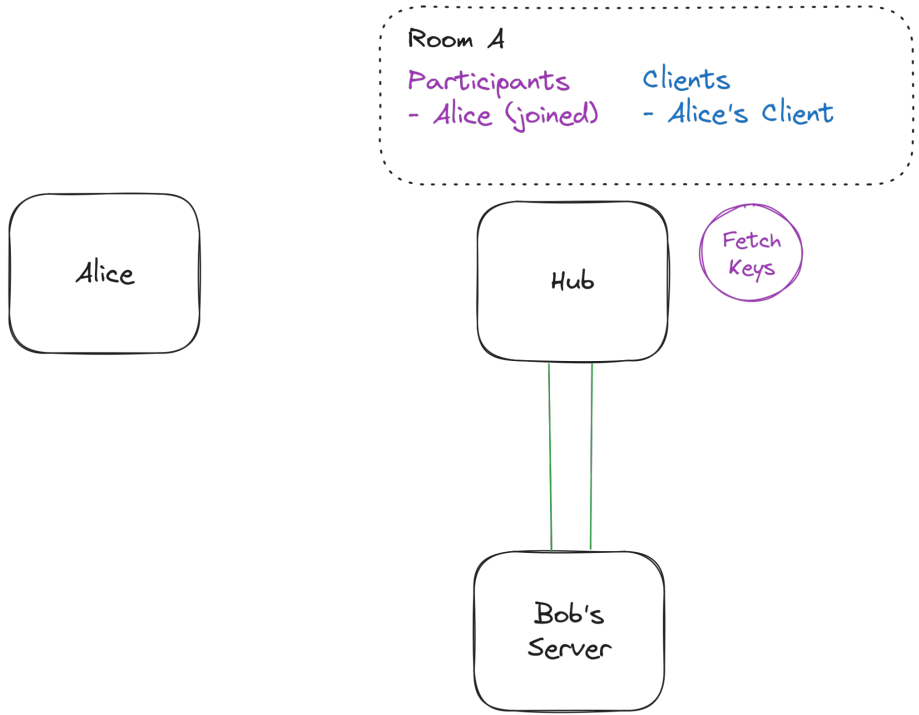
Legend:

Signalling

DS

Transport

Policy



Fetch Keys = ds.fetch_key_packages

Alice fetches Bob's KeyPackages

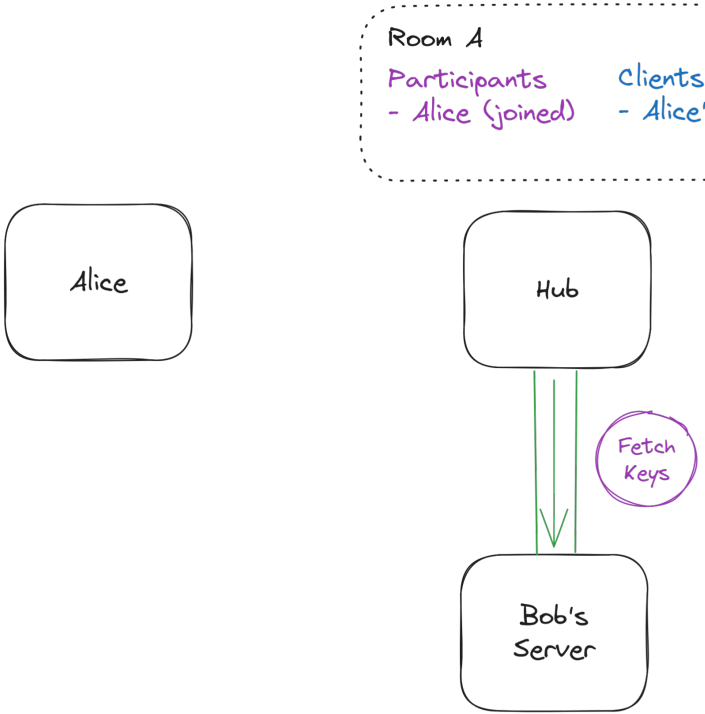
Legend:

Signalling

DS

Transport

Policy



Fetch Keys = ds.fetch_key_packages

Alice fetches Bob's KeyPackages

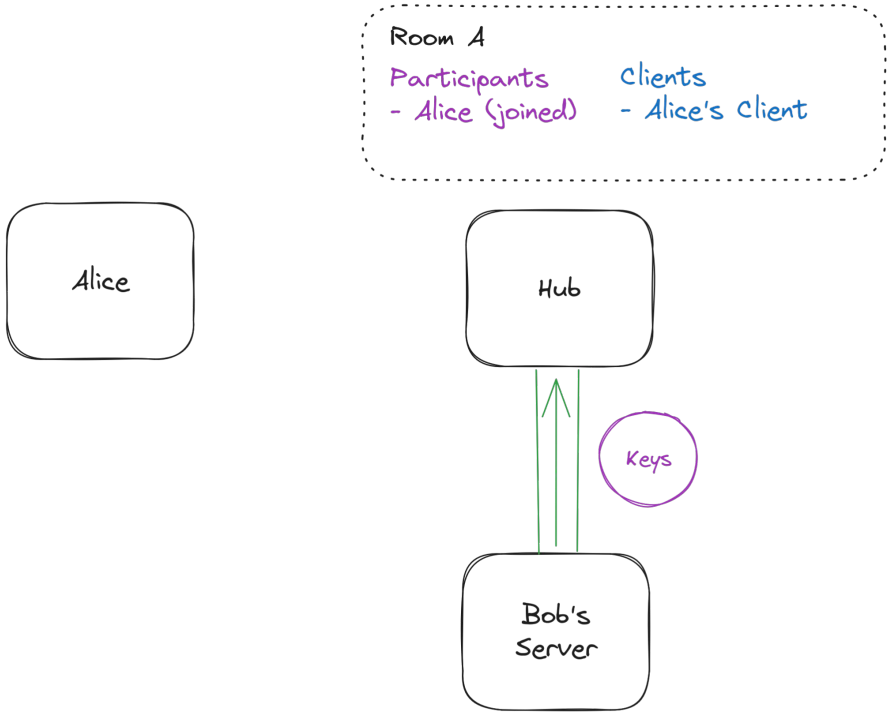
Legend:

Signalling

DS

Transport

Policy



Keys =
MIMIResponse (KeyPackages)

Alice fetches Bob's KeyPackages

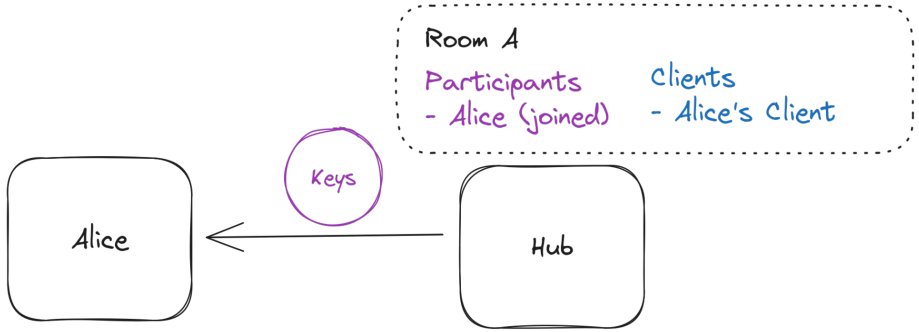
Legend:

Signalling

DS

Transport

Policy



Keys =
MIMIResponse (KeyPackages)

Alice creates an add event for Bob

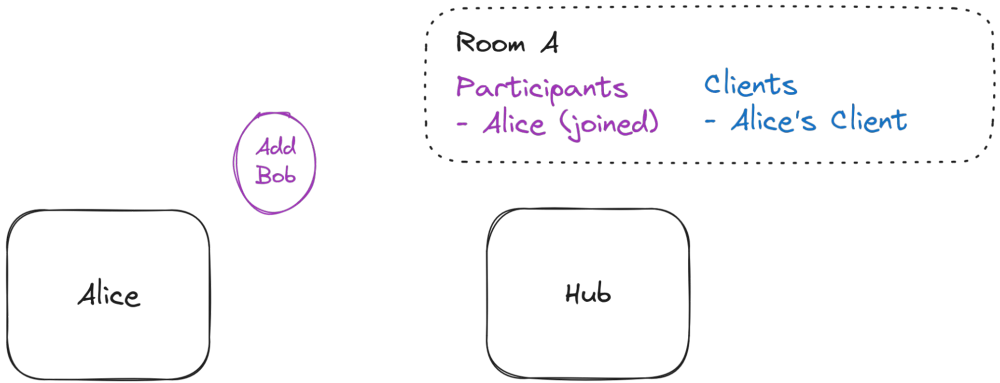
Legend:

Signalling

DS

Transport

Policy



Add Bob = ds.commit
- m.room.user
- Add

Alice sends the invite to the hub

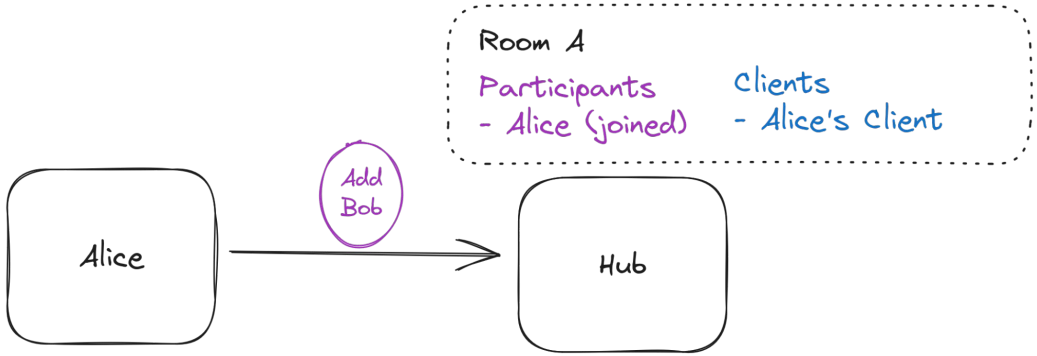
Legend:

Signalling

DS

Transport

Policy



Add Bob = ds.commit
- m.room.user
- Add

Hub establishes secure transport

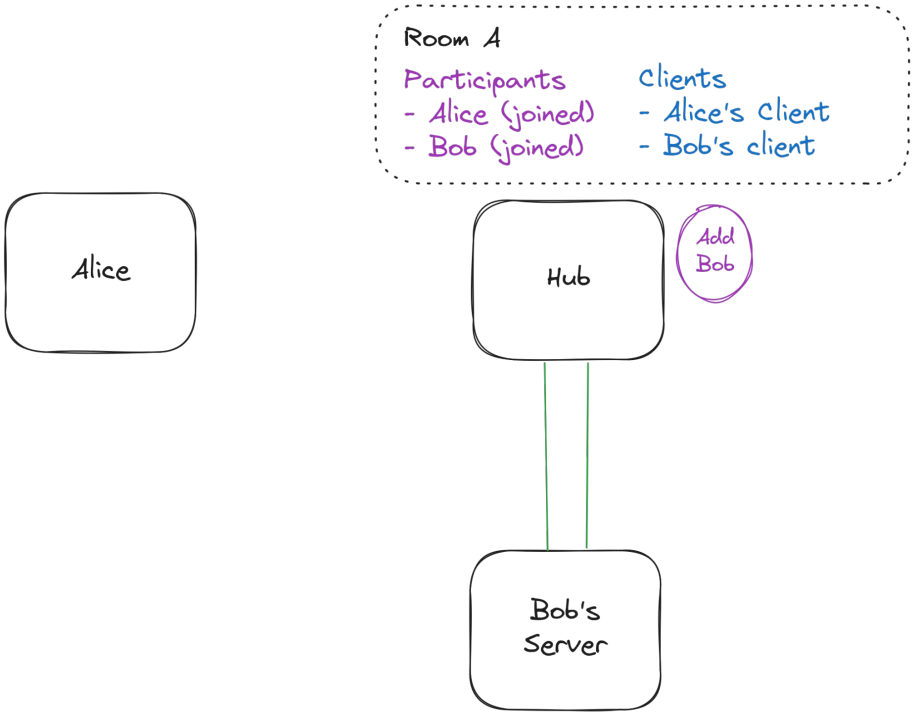
Legend:

Signalling

DS

Transport

Policy



Add Bob = ds.commit
- m.room.user
- Add

Hub fans out the signalling event

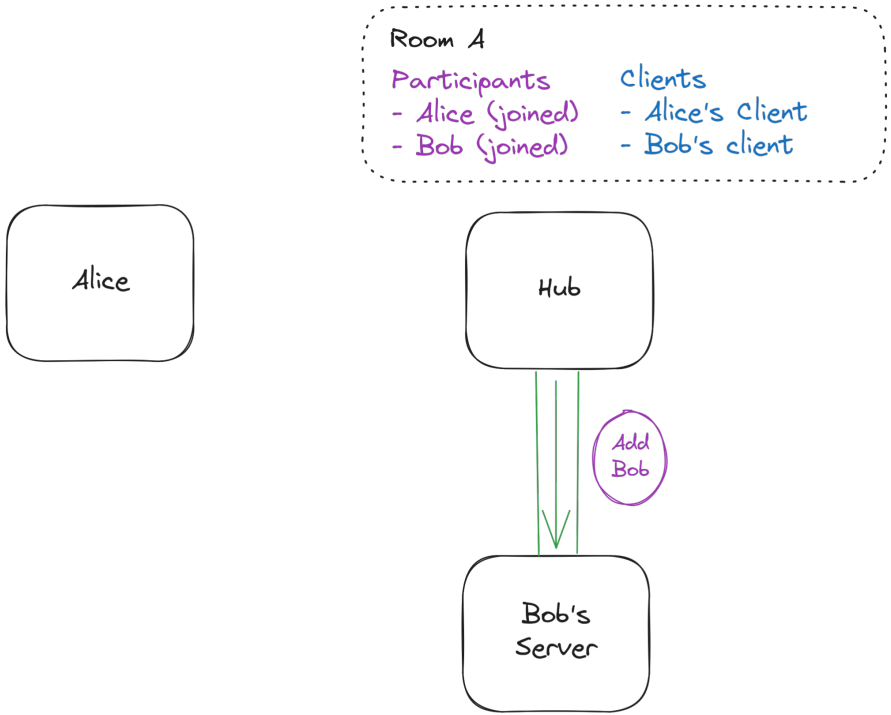
Legend:

Signalling

DS

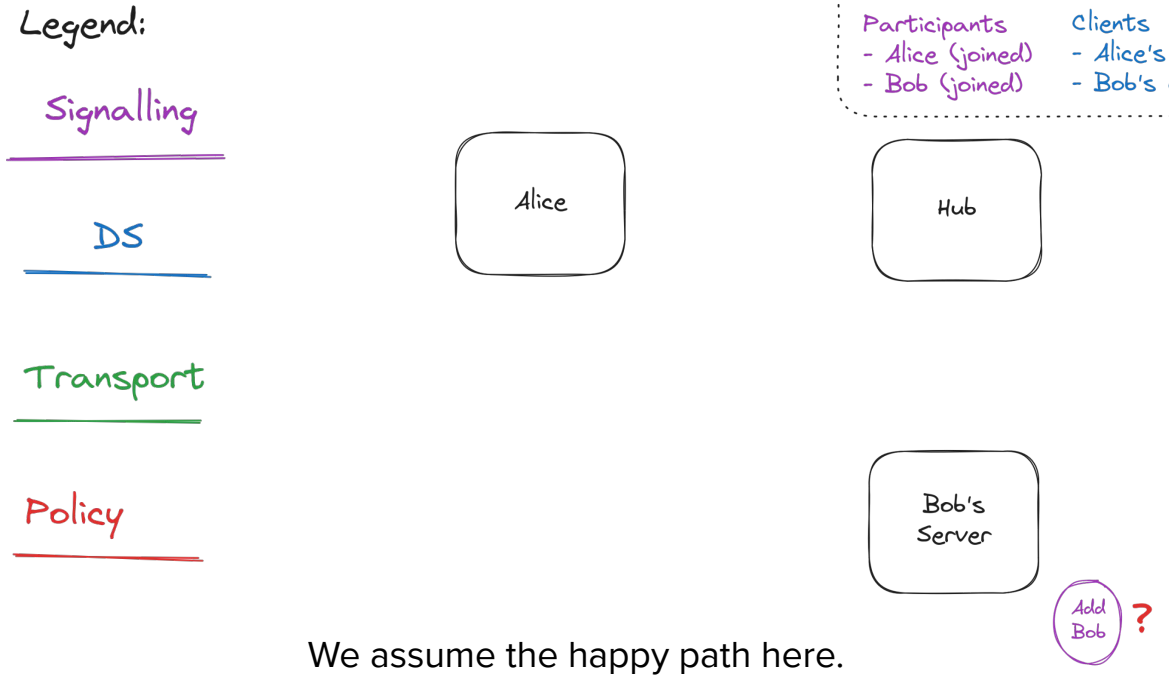
Transport

Policy



Add Bob = ds.commit
- m.room.user
- Add

Bob's server checks policy support



Add Bob = ds.commit
- m.room.user
- Add

We assume the happy path here.

Bob's server stores-and-forwards

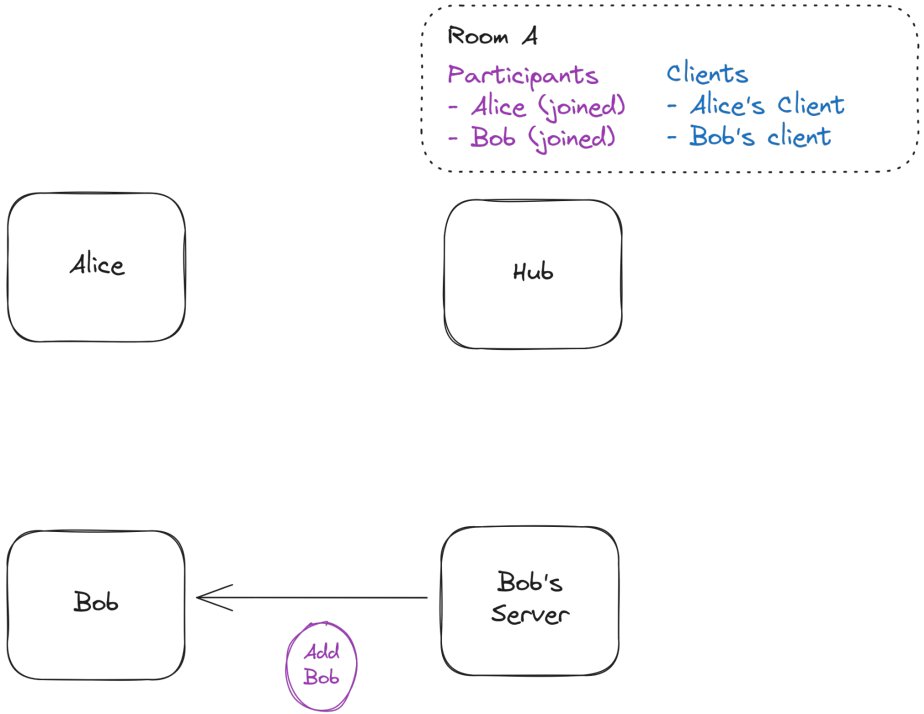
Legend:

Signalling

DS

Transport

Policy



Add Bob = ds.commit
- m.room.user
- Add

Hello world

Alice and Bob can converse! 🎉