

Between the protocol and the customer

Federation expectations in production: what Rocket.Chat learned running Matrix for government, defense and critical infrastructure.

Renato Becker

PM, Extensibility, Interoperability and AI



Who we are

One of the world's largest open-source comms platforms

Rocket.Chat Secure CommsOS®

Sovereign, never siloed.

Interoperable by design. Governed by you.

An open source, secure communications platform for messaging, voice, video and governed AI, natively federated over Matrix. Built for defense, the public sector and critical infrastructure that must work across organizations, borders and classification levels.

Open source since 2015

On-prem · air-gapped · sovereign cloud

Six pillars



Unified communications, deployable anywhere

Messaging, voice, video, files, apps and governed AI agents. One platform, one security model.



Zero Trust

ABAC verifies every access, continuously. End-to-end encryption today; post-quantum on the roadmap.



Resilient at the tactical edge

Operates under DDIL conditions and integrates with mission systems such as TAK Server and ATAK.



Continuity of operations

Redundant high-availability architecture and secure fallback channels through outage, cyberattack or crisis.



Sovereign AI agents

Agents act on requests or propose actions. Humans stay in command; nothing runs outside your approvals.



Interoperable by design

Protocol agnostic, with native Matrix federation built in. Agencies, allies and partners work together while each keeps its own servers, identities and rules.

[THE REST OF THIS TALK →](#)

Where we sit

Every organization that adopts Matrix federation **makes the protocol more valuable**. Every gap it hits **makes adoption harder**.

The protocol

Matrix

Open standard, decentralized, built for true federation, not shared channels. Federation traffic travels between servers nobody owns centrally.



The platform

Rocket.Chat Secure CommsOS®

Matrix spoken natively, delivered as a governed, production-ready platform. Where the protocol becomes the outcome.



The customer

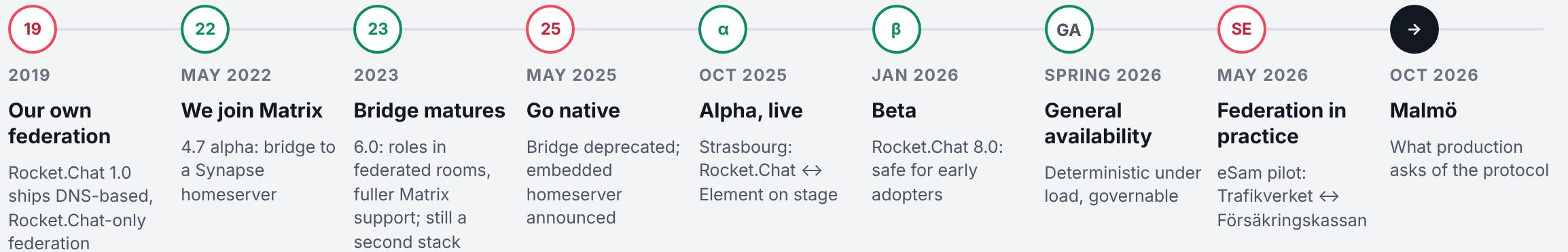
Government, defense, critical infrastructure

They don't buy protocols. They buy "work with the partner agency without giving up control."

1 The journey 2 What production taught us 3 Where adoption gets stuck, and what we propose

Between the protocol and the customer

Seven years from "our own protocol" to a native Matrix homeserver



2019 · Build our own

A protocol only Rocket.Chat spoke.

2022–2023 · Bridge to the standard

Matrix, through someone else's homeserver.

2025 · Embed the standard

Matrix, spoken natively by Rocket.Chat itself.

Interoperability was in the plan before Matrix was

Gabriel Engel, our co-founder and CEO, has always described the same world: every organization runs its own server, owns its own data, sets its own rules, and can still talk to everyone else.

Sovereign

Your infrastructure, your jurisdiction, your keys. No central provider in the path.

Governed

Who can see what is defined, enforced and auditable by the organization that owns it.

Interoperable

Open standards between organizations, so nobody is locked into one vendor to collaborate.

The question was never **whether** to interoperate. It was **how**.

2019 · Attempt one

We built our own federation protocol

Rocket.Chat 1.0 shipped federation between Rocket.Chat servers, discovered over DNS. It worked. It also taught us what we are not.

We don't do protocols.

A protocol only one vendor speaks is not interoperability. Spec work, governance and an ecosystem of implementations are a different business from building a collaboration platform.

2019

github.com/RocketChat/Rocket.Chat.Federation

"Chat servers. United."

rc-a.example ↔ rc-b.example ✓

rc-a.example ↔ any other platform ✗

May 2022 · Attempt two

We found a protocol that shares our values

Open, decentralized, run by a foundation, built for organizations that keep their own servers. We pivoted: Rocket.Chat 4.7 shipped Matrix federation in alpha.

Application Service

Bridge

Synapse homeserver

PostgreSQL

[matrix]

"The more organisations that join Matrix, the more useful and valuable the network becomes for everyone."

Matrix.org · "Welcoming Rocket.Chat to Matrix" · 30 May 2022

That sentence is the thesis of this talk, four years early.

Customers wanted interoperability, not a second platform to run

- ◆ A second stack beside Rocket.Chat: Synapse, PostgreSQL and the bridge
- ◆ Two databases to keep in sync, patch, back up and accredit
- ◆ Silent failures and little visibility into where data flowed
- ◆ Federation traffic was almost entirely Rocket.Chat to Rocket.Chat

2
stacks to
operate

2
databases

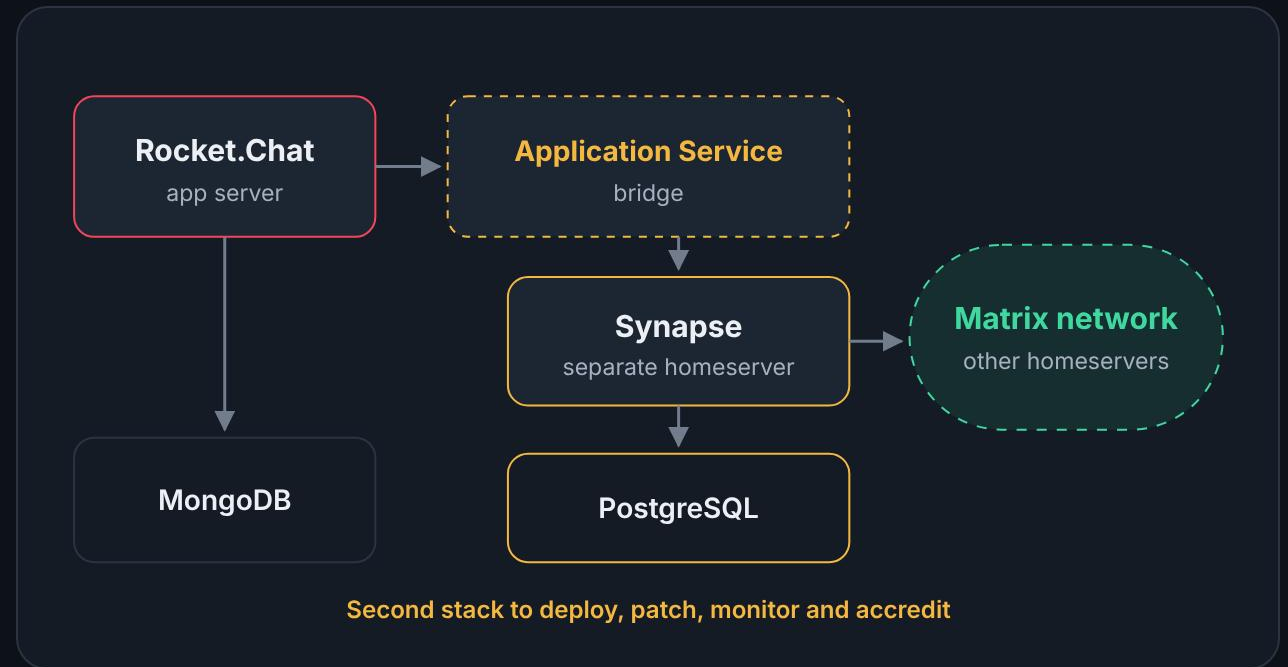
Synapse
homeserver

TAKEAWAY

Interoperability must cost less to run than isolation. **Add a second platform, and customers choose the silo.**

Bridge era · 2022

What if...?



May 2025 · Attempt three

The bold call: build our own Matrix homeserver

Matrix as a native part of the platform, not a bridge to someone else's homeserver. The federation engine shares Rocket.Chat's process and database: no external homeserver, no second system to accredit.

Embedded library

Runs in the Rocket.Chat process

One port

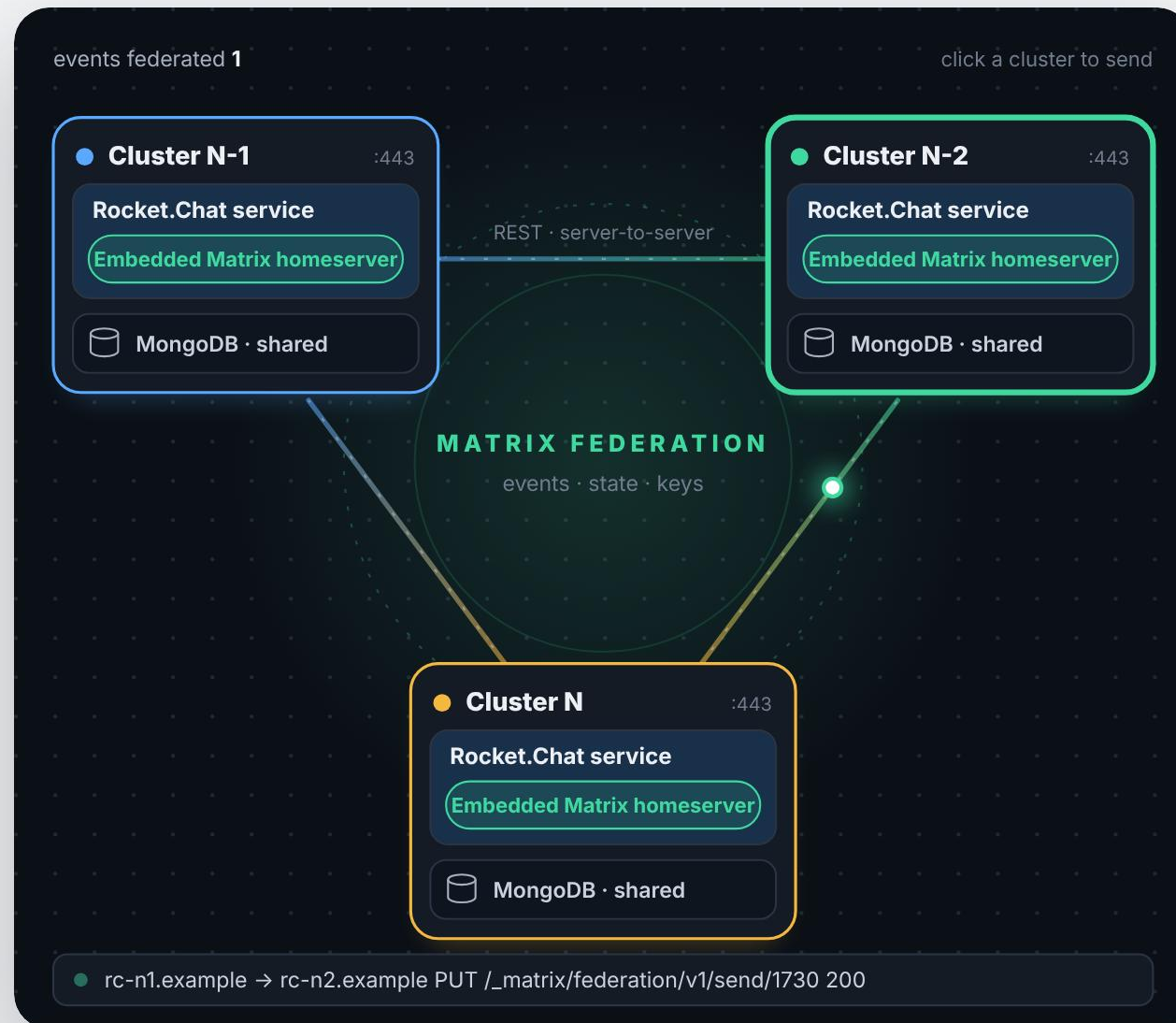
Federation REST APIs beside the app

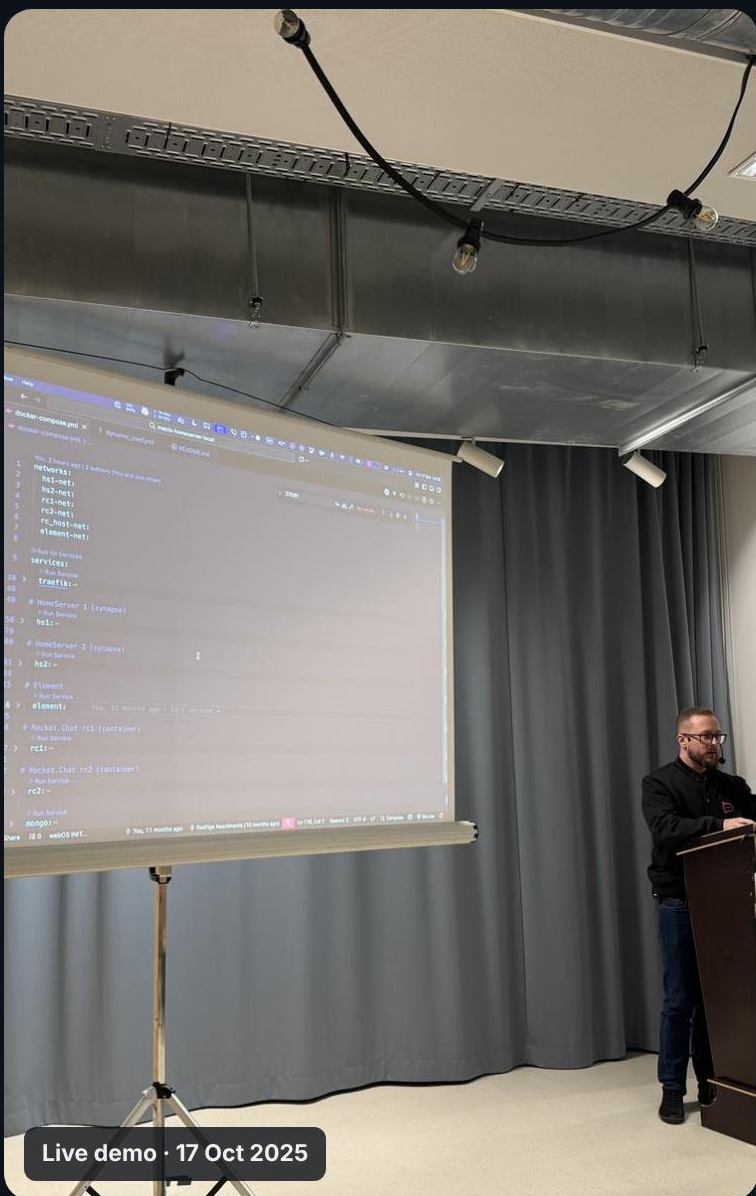
Shared MongoDB

No PostgreSQL, no sync job

Synapse as reference

When unsure, match its behavior





Live demo · 17 Oct 2025



"Entering the Matrix"



Strasbourg

Oct 2025 · Alpha

We demoed a months-old homeserver live, mid-conference

Alpha federated with Element and messages flowed both ways. The UI was clunky. That was expected: alpha existed to prove the Matrix primitives.

Rocket.Chat α



Element

Chapter one, not the finish line.

Beta rule: federation must never put internal comms at risk

Switched off in federated rooms

- End-to-end encryption
- Voice and video calls
- Read receipts

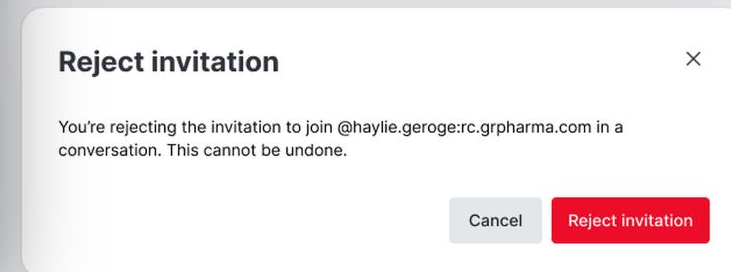
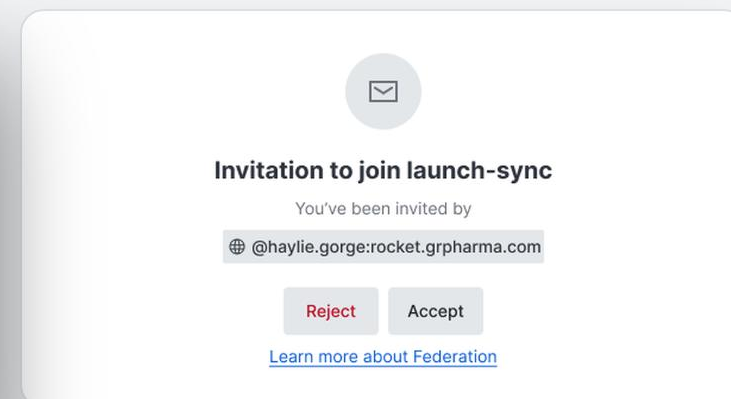
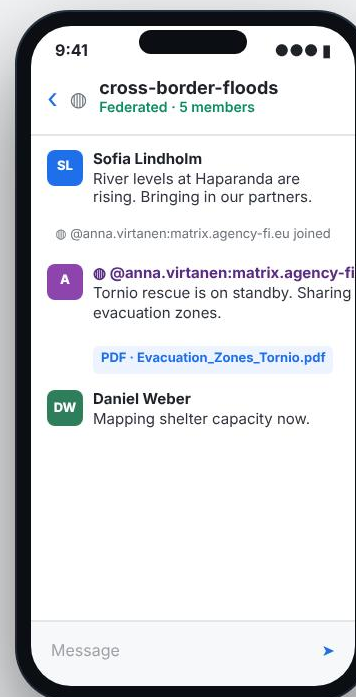
Scope decisions, not gaps in vision. Half-working features erode trust faster than missing ones.

Primitives added

- Invitation handling
- A federation permission and role
- Human-readable errors instead of stack traces
- Federated rooms on desktop and mobile

Proof before release

- 100+ protocol test cases
- First load baseline on 3 homeservers
- Room state and message order checked across servers



LEGACY PATH Bridge-based federation deprecated. We started migrating existing bridge deployments to Native Federation.

Spring 2026 · General availability

GA meant moving from best effort to deterministic

The question stopped being “does it work?” It became “what happens to my operation when a partner server misbehaves?” Partners started to doubt around 500–600 federated users.

Load-tested limits

Known capacity, published to customers

Circuit breakers

Stop calling a partner that stops answering

Federated ACKs (read receipts)

Proof that the message arrived

Remote profile rendering

Partners show up with their real display names and avatars

Workload isolation

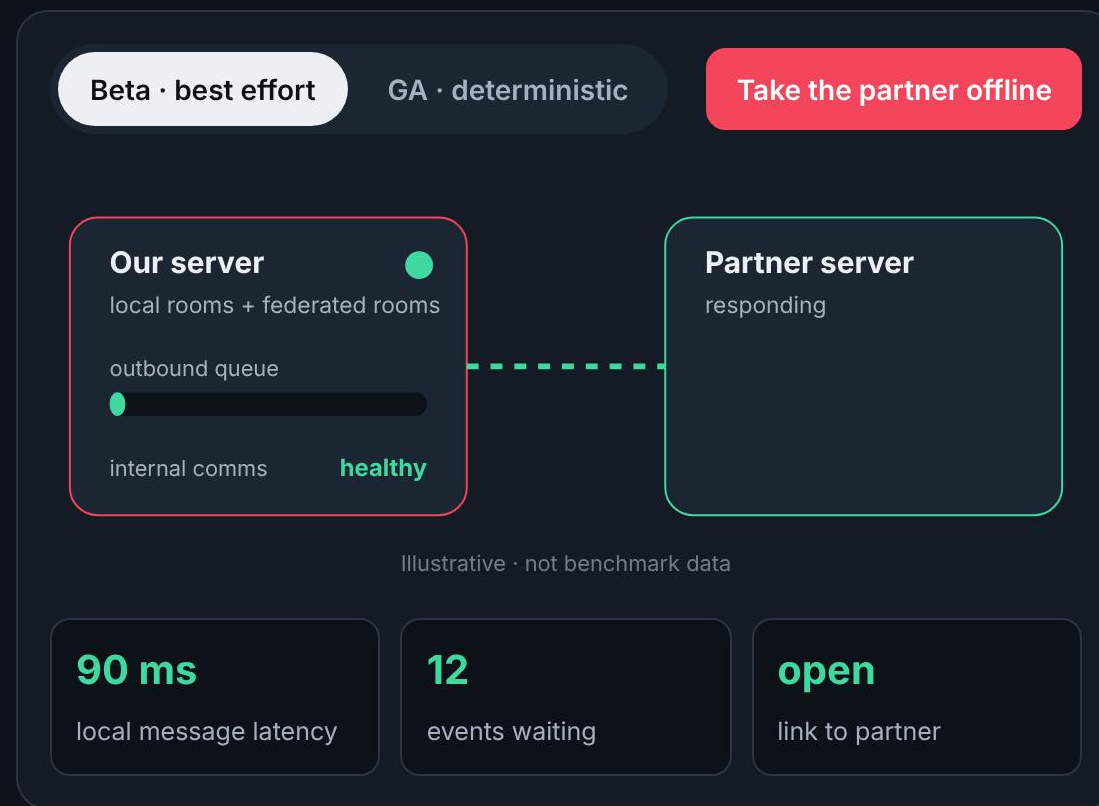
Federated traffic can't starve local rooms

Parallel outbound queues

Clear the outbox fast after a spike

Federated ban

Remove a threat from every server



Federation in practice: two agencies, two vendors, one conversation

ROCKET.CHAT

Trafikverket

Swedish Transport Administration

Matrix
open protocol

ELEMENT

Försäkringskassan

Swedish Social Insurance Agency

Proven

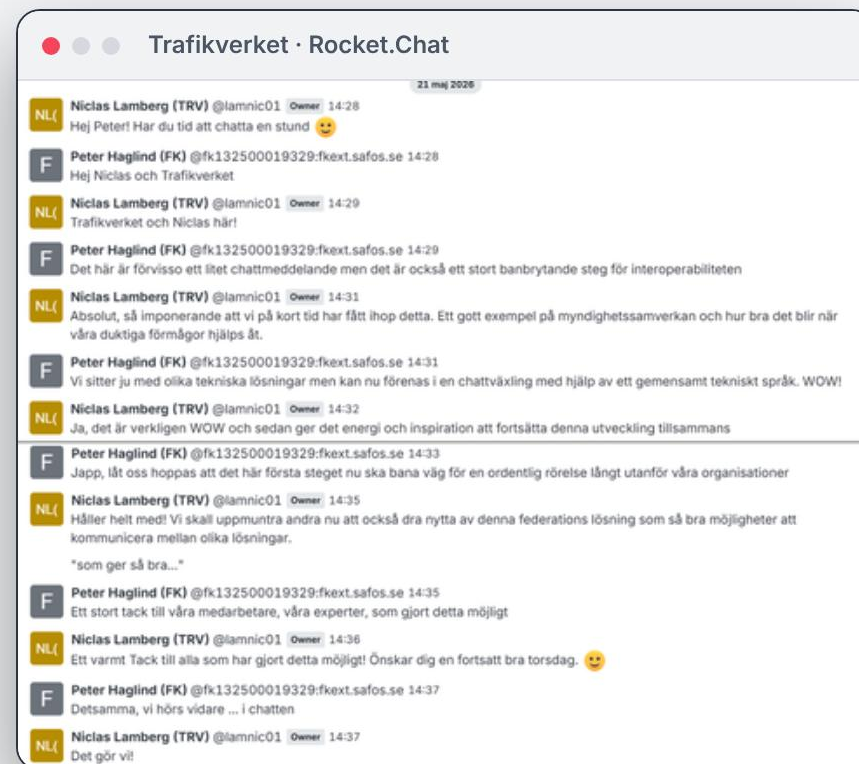
Different systems talk, and each agency **keeps its own solution.**

After summer 2026

The pilot grows to **more agencies and users**; any Matrix-capable platform can join.

dSam

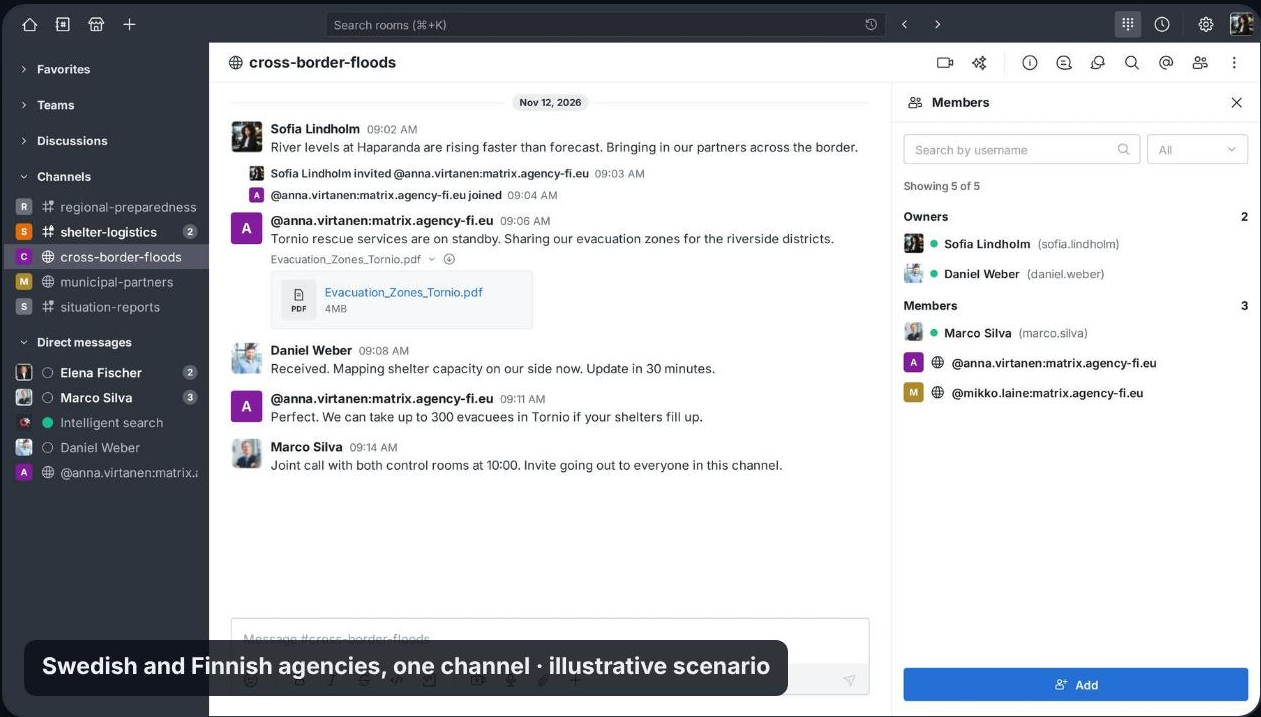
Intends to **recommend Matrix as the common standard** for inter-agency communication.



Live, in Rocket.Chat: Trafikverket's Niclas Lamberg and Försäkringskassan's Peter Haglind, chatting across two vendors.

We're not only Matrix enablers. We're Matrix consumers.

A federated room should feel like just another day in Rocket.Chat: whatever works internally should work across the boundary.



7 of 13 work across federation today, 1 more partly

CAPABILITY	INTERNAL	FEDERATED
Messages, edits, deletes	✓	✓
Threads, reactions, mentions	✓	✓
File sharing	✓	✓
Invitations	✓	✓
Read receipts	✓	✓
Moderation: kick and ban	✓	✓
Desktop and mobile clients	✓	✓
Room roles	✓	🔒
End-to-end encryption	✓	ROADMAP
User discovery	✓	—
Attribute-based access	✓	—
Security labels	✓	—
Voice and video	✓	—

What we hit, what's ahead, and what the field keeps asking for

● Heard from the field ● Hit while embedding Matrix ● Challenge ahead

01

HEARD FROM THE FIELD

Find people across organizations

"Why do I need their address first?"

MSC4258 · federated directory

02

HEARD FROM THE FIELD

Policy-aware rooms (ABAC)

"Is everyone in this room cleared for it?"

MSC4284 · policy servers

03

HIT WHILE EMBEDDING

Roles that survive federation (RBAC)

"Why is our moderator a regular member there?"

MSC4056 · roles in Matrix

04

CHALLENGE AHEAD

Voice and video across vendors

"Why can't I call them?"

MSC4143 · MatrixRTC

Questions

Tack.

VISIT US AT THE BOOTH

See Rocket.Chat's native Matrix federation running live. Bring your questions, your use case, or your gaps. We'd love to compare notes.

[docs.rocket.chat · Federation](https://docs.rocket.chat/Federation)

conference.matrix.org

rocket.chat