



[matrix]

The Matrix State of the Union

The Matrix Conference 2025

matthew@matrix.org

@matthew:matrix.org

**The world needs Matrix
more than ever.**

**Matrix feels like it is really
growing up.**

**We have all the pieces
needed to outcompete the
centralised alternatives...**

...and it's happening!

**We have an impressively
diverse ecosystem.**

Just looking at Clients...

A galaxy of mature clients...

[matrix]



Beeper



Native Mobile UI
(SwiftUI or Compose)

mautrix



Go

A galaxy of mature clients...

[matrix]



Beeper



Element
X



Native Mobile UI
(SwiftUI or Compose)

mautrix

matrix-rust-sdk



Go



Rust

A galaxy of mature clients...

[matrix]



Beeper



Element
X



Fractal



Native Mobile UI
(SwiftUI or Compose)



GTK4

mautrix

matrix-rust-sdk



Go



Rust

A galaxy of mature clients...

[matrix]



Beeper



Element
X



Fractal



Robrix



Native Mobile UI
(SwiftUI or Compose)



GTK4



MakePad

mautrix

matrix-rust-sdk



Go



Rust

A galaxy of mature clients...

[matrix]



Beeper



Element
X



Fractal



Robrix



Filament



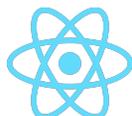
Native Mobile UI
(SwiftUI or Compose)



GTK4



MakePad



React
Native

mautrix

matrix-rust-sdk



Go



Rust

A galaxy of mature clients...

[matrix]



Beeper



Element
X



Fractal



Robrix



Filament



Axiom



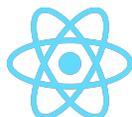
Native Mobile UI
(SwiftUI or Compose)



GTK4



MakePad



React
Native



Flutter

mautrix

matrix-rust-sdk



Go



Rust

A galaxy of mature clients...

[matrix]



Beeper



Element
X



Fractal



Robrix



Filament



Axiom



Fluffy
Chat



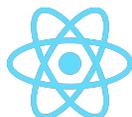
Native Mobile UI
(SwiftUI or Compose)



GTK4



MakePad



React
Native



Flutter

mautrix

matrix-rust-sdk

matrix-
dart-sdk



Go



Rust



Dart

A galaxy of mature clients...

[matrix]



Beeper



Element
X



Fractal



Robrix



Filament



Axiom



Fluffy
Chat



Tammy



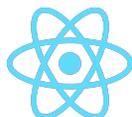
Native Mobile UI
(SwiftUI or Compose)



GTK4



MakePad



React
Native



Flutter



Kotlin
Multi
Platform

mautrix

matrix-rust-sdk

matrix-
dart-sdk

Trixnity
SDK



Go



Rust



Dart



Kotlin

A galaxy of mature clients...

[matrix]



Beeper



Element
X



Fractal



Robrix



Filament



Axiom



Fluffy
Chat



Tammy



Nheko



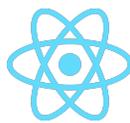
Native Mobile UI
(SwiftUI or Compose)



GTK4



MakePad



React
Native



Flutter



Kotlin
Multi
Platform



Qt

mautrix

matrix-rust-sdk

matrix-
dart-sdk

Trixinity
SDK

mtxclient

lib
Quotient



Go



Rust



Dart



Kotlin



C++

A galaxy of mature clients...

[matrix]



Beeper



Element
X



Fractal



Robrix



Filament



Axiom



Fluffy
Chat



Tammy



Nheko



NeoChat



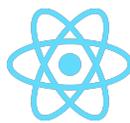
Native Mobile UI
(SwiftUI or Compose)



GTK4



MakePad



React
Native



Flutter



Kotlin
Multi
Platform



Qt

mautrix

matrix-rust-sdk

matrix-
dart-sdk

Trixnity
SDK

mtxclient

lib
Quotient



Go



Rust



Dart



Kotlin



C++

A galaxy of mature clients...

[matrix]



Beeper



Element
X



Fractal



Robrix



Filament



Axiom



Fluffy
Chat



Tammy



Nheko



NeoChat



Element
Web



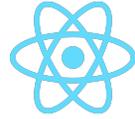
Native Mobile UI
(SwiftUI or Compose)



GTK4



MakePad



React
Native



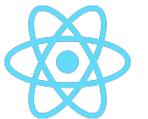
Flutter



Kotlin
Multi
Platform



Qt



React

mautrix

matrix-rust-sdk

matrix-
dart-sdk

Trixnity
SDK

mtxclient

lib
Quotient

matrix-js-
sdk



Go



Rust



Dart



Kotlin



C++



Web

A galaxy of mature clients...

[matrix]



Beeper



Element
X



Fractal



Robrix



Filament



Axiom



Fluffy
Chat



Tammy



Nheko



NeoChat



Element
Web



Cinny



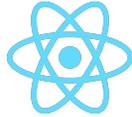
Native Mobile UI
(SwiftUI or Compose)



GTK4



MakePad



React
Native



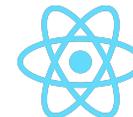
Flutter



Kotlin
Multi
Platform



Qt



React

mautrix

matrix-rust-sdk

matrix-
dart-sdk

Trixnity
SDK

mtxclient

lib
Quotient

matrix-js-
sdk



Go



Rust



Dart



Kotlin



C++



Web

Uptake is accelerating.

Retained 30-day users (Based on Synapse phone-home stats)

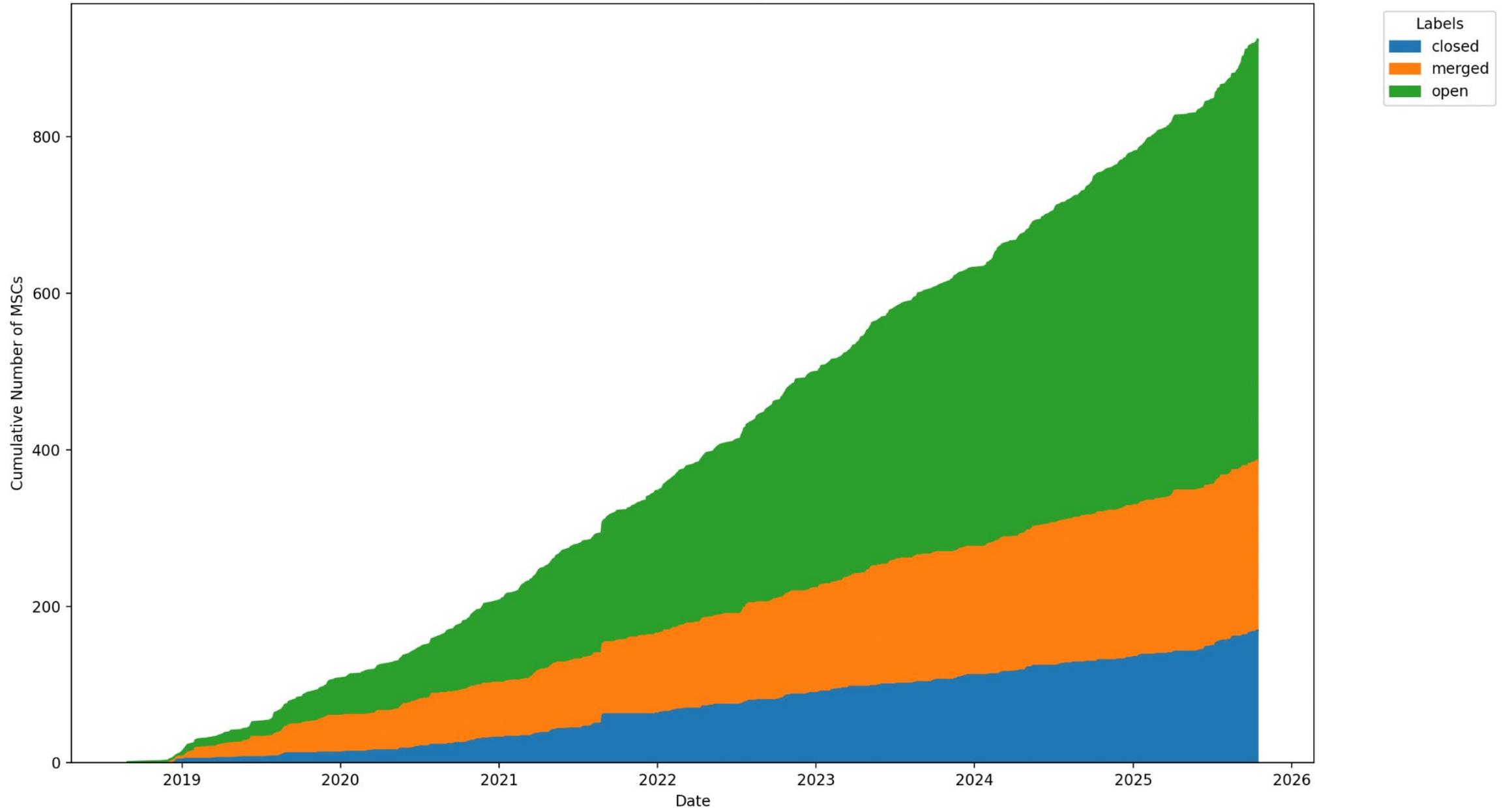
Matrix.org R30 by client ⚠️



- R30 shows committed users, who have been active for over 30 days.
- Doubled from Sep 2024 -> Sep 2025.

**The spec is steadily
improving.**

Cumulative Number of MSCs by Kind Over Time



Matrix 2.0 Update

- Last year we showed off the first impls of the MSCs which the Spec Core Team has earmarked for Matrix 2.0.
- Lots and lots of refinement of the MSCs over the course of the year as they've been tested in the wild.
- We were aiming to get all Matrix 2.0 MSCs in Final Comment Period to merge for this year's Matrix Conference... and we're close.
- But where there are still things to be ironed out, we're not going to rush it and mess up the final details just so we can celebrate it on stage!

Next Gen Auth (MSC3861)

- areweoidcyet.com? **Yes we are!!** 🎉
- We now live in the promised land of industry standard authentication!
 - Password mgr support! Passkeys! 2FA! MFA! QR login! Refresh tokens!
- MSC3861 was merged on Apr 5th and released in Matrix 1.15
- Deployed on matrix.org on Apr 7th, migrating all 45M access tokens in under 30 minutes (come see Quentin's talk tomorrow at 16:30!)
- The QR login MSC however needs some further iteration:
 - Switching from ECIES to RFC 9180 HPKE
 - Ironically HPKE only came into existence after we began the QR login MSC, but as don't want to spec obsolete crypto, we're reworking to use HPKE.

Simplified Sliding Sync: SSS (MSC4186)

[matrix]

- Lots of finessing as the matrix-rust-sdk implementation has matured.
- Threads support being added as an extension
- Spaces support likewise
- **Now gathering votes to enter FCP** as of Sept 29th!
- Ongoing discussion if all the SSS extensions also need to FCP for Matrix 2.0.

MatrixRTC (MSC4143)

[matrix]

- Loads of work to refine the MSCs over the year, as we learnt from the implementation getting fully exercised in Element Call.
- Challenges have included:
 - Handling races on call state due to decentralisation (“call glare”) transparently
 - Permission problems - differentiating between permission to start a call versus join a call.
 - State bloat - using state events to track participation can result in impressive explosions of state events; great for stress-testing state resolution; not great when any user in the call can set them.
 - Refining participation timeouts - ensuring that if users lose connectivity, they rapidly and accurately get removed from the call.

MatrixRTC (MSC4143)

- **We believe these are now fixed!**
- All new edition of MSC4143 was published on Tuesday.
- Rather than tracking participation with state events, we have introduced a whole new class of event in Matrix called **sticky events** (MSC4354).
 - Sticky events are simply timeline events which get reliably delivered to clients, regardless of sync gaps or other timeline activity - sticking around for a timeframe (typically an hour) until they drift back into history.
 - This is super useful for tracking call participation (and call ringing) without users needing permission to set state in the room, and without bloating state.
- We've also introduced a new concept of MatrixRTC "slots" to solve race conditions: each room can have one or more concurrent slots around which RTC sessions converge.
- We should be *much* better set up for FCP now.

Invisible Crypto (MSC4153 et al)

[matrix]

- The idea of Invisible Crypto is to **make E2EE invisible to the user**, much like iMessage, WhatsApp or Signal.
- The main step is that all devices should be verified (cross-signed) by their owner - and once this happens, you will not be able to encrypt to or decrypt from devices which have not been cross-signed by their owner.
- This has taken a while, stuck behind other encryption work, plus all clients will need to be updated, including encryption-aware bridges.
- **However, we are finally in the end game.**
- MSC4153 has now passed FCP and is ready for merge into the spec.
- Therefore the clock is ticking - 6 months from now (Apr 2026, or earlier, depending on uptake) we expect another MSC to mandate that all devices **MUST** be verified.

Matrix 2.0 Summary

[matrix]

	Final MSC	MSC proposed for FCP	MSC passed FCP	MSC merged in the spec
Next Gen Auth MSC3861	✓	✓	✓	✓
Simplified Sliding Sync MSC4186	✓	✓		
MatrixRTC MSC4143	✓			
Invisible Crypto MSC4153	✓	✓	✓	

Based on this, we are aiming that the next spec release will be Matrix 2.0!

Other stuff happening...

Project Hydra

- Huge amounts of work this year funded by Element to improve state resolution: how conflicts between rooms are handled. This has turned into a roughly 3 phase plan:
 - **Phase 1: Analysis & baseline (shipped in Room v12, Aug 2025)**
 - Make it explicit that room creators are privileged (MSC4289)
 - Make room IDs hashes of the create event, to stop attacks where different servers claim false m.room.create events (MSC4291)
 - State Resolution v2.1, which stops many classes of state resets (MSC4297)
 - **Phase 2: Additional robustness (early 2026)**
 - Switch to synchronising full state DAGs to improve robustness
 - Switch to user IDs as account keys (MSC4243)
 - **Phase 3: Finality (late 2026)**
 - Add finality, to stop conflicts caused by backdating events
 - Updating State Resolution as needed (v3)

**Don't miss Kegan's
talk, Friday 12:10!**

Trust & Safety

- There has been significant (much needed) progress on Trust & Safety.
- Matrix has a growing ecosystem of developers working on safety tooling
- Modern moderation tools for communities, including Draupnir and Meowlnir
 - Volunteer hosting for moderation tooling: <https://asgard.chat>
- Grassroots organizations tackling abuse on the network
 - Community Moderation Effort (CME) publishing their shared blocklist of spammers
 - homeserver admin communities like <https://muninn-hall.com/>
 - Matrix Safety Guild
- Policy Servers (MSC4284) - gives moderators the ability to filter traffic in rooms *before* it hits users.
 - Closed beta for communities running now
 - Open beta coming imminently, then FOSS launch of v1 soon

Trust & Safety

- Tooling to filter media and invites
 - MSC4278 Media Preview Controls
 - MSC4155 Invite Filtering
- Foundation T&S working groups and Safety Guild progress.
 - Open governance of the matrix.org room directory
- Founding member of ROOST, working on open source T&S tooling.
- Currently investigating Osprey (ROOST's distribution of Discord's open-sourced moderation tooling)
- Still lots to be done: Join gates, Reputation models, Discovery, server admin tools, Improving the things we already have
- Core Trust & Safety work is funded by the Foundation itself alongside handling abuse@matrix.org.

Please become a Fdn member to ensure this work can progress.

Encryption Improvements

- Lots of work on sharing historical keys when you join a room.
- Encrypted state events progress! MSC3414 prototype in Element Web.
- E2EE Search progress in matrix-rust-sdk!
- We expect to resume work on Post-Quantum Olm (PQXDH) in November! Many thanks to those funding Element to make this happen.
- On MLS on Matrix:
 - BWI and Famedly are progressing on MSC4256 (“RFC 9420 MLS mode Matrix”), which effectively replaces Matrix’s state resolution with centralised MLS groups.
 - The Spec Core Team are also looking at more incremental approaches too - Phoenix IM’s “Decentralized MLS” IETF draft, as well as Germ Network’s “Distributed MLS” IETF draft.

Digital Markets Act and MIMI

- In Sept 2024 we published “[Update on Native Matrix interoperability with WhatsApp](#)” to gauge interest in WhatsApp interop in face of the limitations present:
 - WhatsApp users would have to be geolocated to the EU
 - Matrix users would have to be geolocated to the EU
 - No multi-device on either side of the bridge
 - No message buffering or history stored in Matrix
- We got no positive feedback.
- Given constrained funding on the Matrix side, we’ve instead focused on improving Matrix - be that with Trust & Safety or MLS work.
- If anyone wants to fund interop work, please let get in touch.

What's next?

The Future

- We're going to continue to laser-focus on being the best decentralised secure comms platform in the world.
- That means we're **not** going to pursue decentralised social - good luck to ATproto, ActivityPub, Nostr etc.
- It means we're **not** going to pursue free-style decentralised data replication - good luck to Automerge, Beehive, Local-First etc.
- It means we're **not** going to build other Matrix showcases (e.g. Thirdroom)
- Instead, we are going to continue focusing **everything** on building a safe, global, resilient, secure communication network which can be used to build decentralised alternatives to WhatsApp, Signal, Telegram, Teams, Slack, Discord and friends.

In practice, this means:

- Finish Hydra.
- Fix client-controlled cryptographic group membership.
 - Either by MLS, DMLS, or client-side state resolution.
- Improve metadata posture
 - Pseudo IDs, encrypted state events, maybe leveraging MLS
- Improving Trust & Safety
- Ensuring that ecosystem-driven features like custom profiles, custom emoji, etc actually land in the spec.
- Figuring out how Matrix can be used by other ecosystems more effectively, to try to avoid ATproto/ActivityPub/Nostr/Beehive etc reinventing Matrix.

Longer term...

- **P2P Matrix is not dead.**
- In fact, Element is funded to resume work on it as of... yesterday.
- P2P solves *many* problems:
 - Keeps metadata entirely on the clients.
 - Provides resilience against blocking/surveillance on servers.
 - Provides connectivity even if the wider network is unavailable.
 - Mandates very robust and resilient federation (hello Hydra!)
- However, it's still an open area of research:
 - Resilient global P2P overlay networks are still a Hard Problem.
 - Naively, you have to sync all possible data to the client, chewing bandwidth and battery. Unless you have smart relays, which then start to sound like servers.

Matrix 3.0 ideas?

- Options are still up in the air:
 - Focus on Trust & Safety?
 - Focus on P2P and metadata protection?
 - Focus on MLS and metadata protection?
 - Focus on “Portable Matrix” targeting other platforms?
 - Just keep on 2.x and only nudge the major version if we get really bored?
- Suggestions welcome!

FRIENDS DON'T LET FRIENDS USE PROPRIETARY CHAT SERVICES

[matrix]



<https://matrix.org/membership/>

- If you benefit commercially from Matrix - **PLEASE** financially support the Foundation
- Run a server
(or get an enterprise one from a supporting vendor)
- Build bridges and bots to your services!
- Build your cool new project on Matrix!
- Follow [@matrix@mastodon.matrix.org](https://mstdn.social/@matrix) & spread the word

[matrix]

Thank you!

[m] @matthew:matrix.org

✉ matthew@matrix.org

🌐 <https://matrix.org>



[@matrix@mastodon.matrix.org](https://matrix.org/@matrix@mastodon.matrix.org)

🦋 [@matrix.org](https://matrix.org)

Trust & Safety

- There has been significant (much needed) progress on Trust & Safety.
 - Policy Servers (MSC4284) - gives moderators the ability to filter traffic in rooms *before* it hits users.
 - Tooling to filter media & invites
 - Foundation T&S working groups and Safety Guild progress.
 - Founding member of ROOST, working on open source T&S tooling.
 - Currently investigating with Osprey (ROOST's distribution of Discord's open-sourced moderation tooling)
- Still lots to be done: Join gates, Reputation models, etc.
- Core Trust & Safety work is funded by the Foundation itself alongside handling abuse@matrix.org.

Please become a Fdn member to ensure this work can progress.