Peer to Peer Matrix: A New Hope!

matthew@matrix.org
@matthew:matrix.org

The world needs secure decentralised communication more than ever.

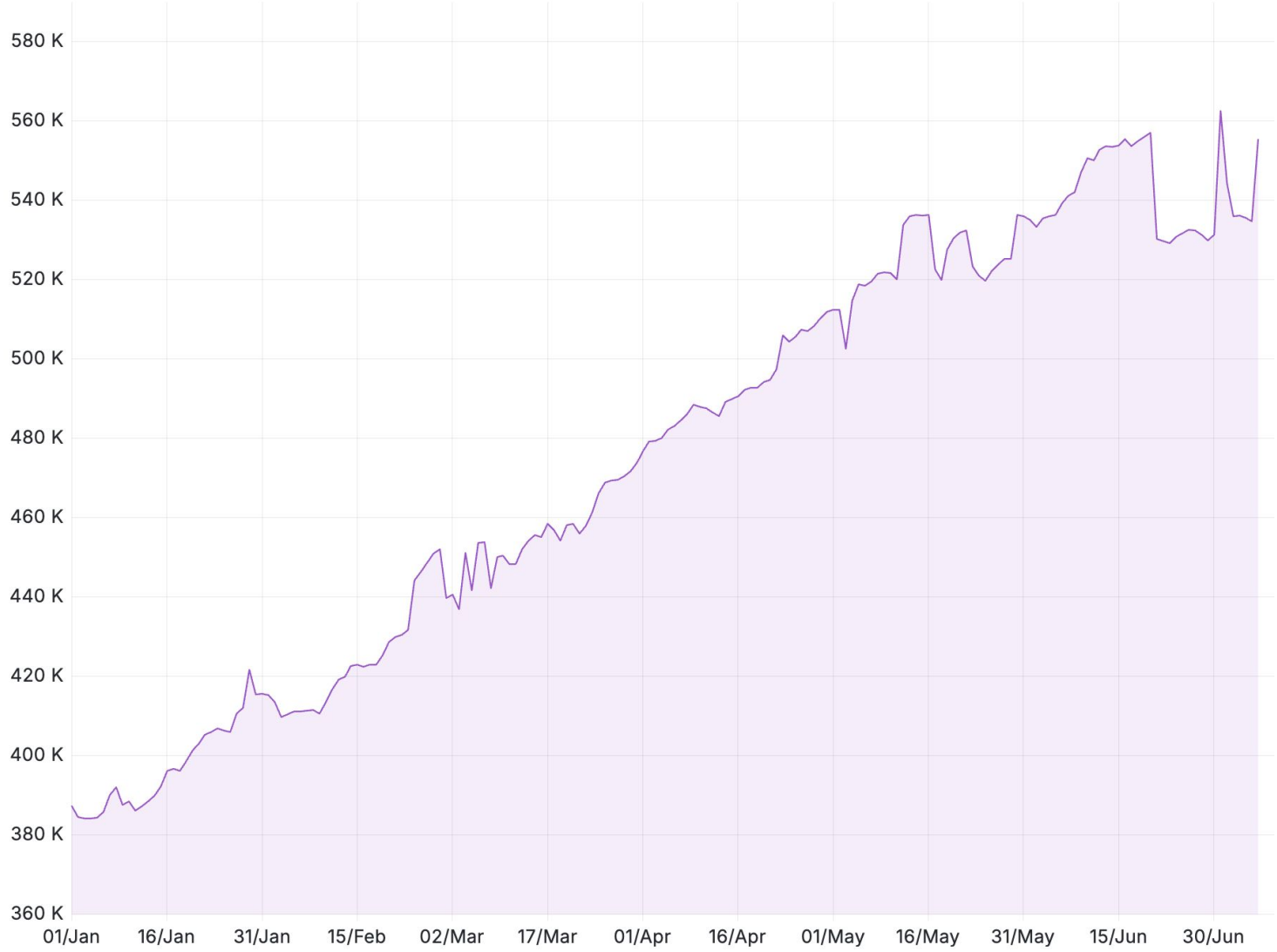
Where is Matrix at?

Current vital stats

Based on Synapse opt-in phone-home stats
(so ignores opt-outs, private deployments, non-Synapse servers)

- 192M unique matrix IDs ever
- 18.3M rooms visible to matrix.org
- 1.2M monthly active users
- 560K 30-day retained users
- 430K daily active users
- ~40% using matrix.org
- ~60Hz of traffic in; ~3kHz of traffic out on matrix.org

30-day retained users



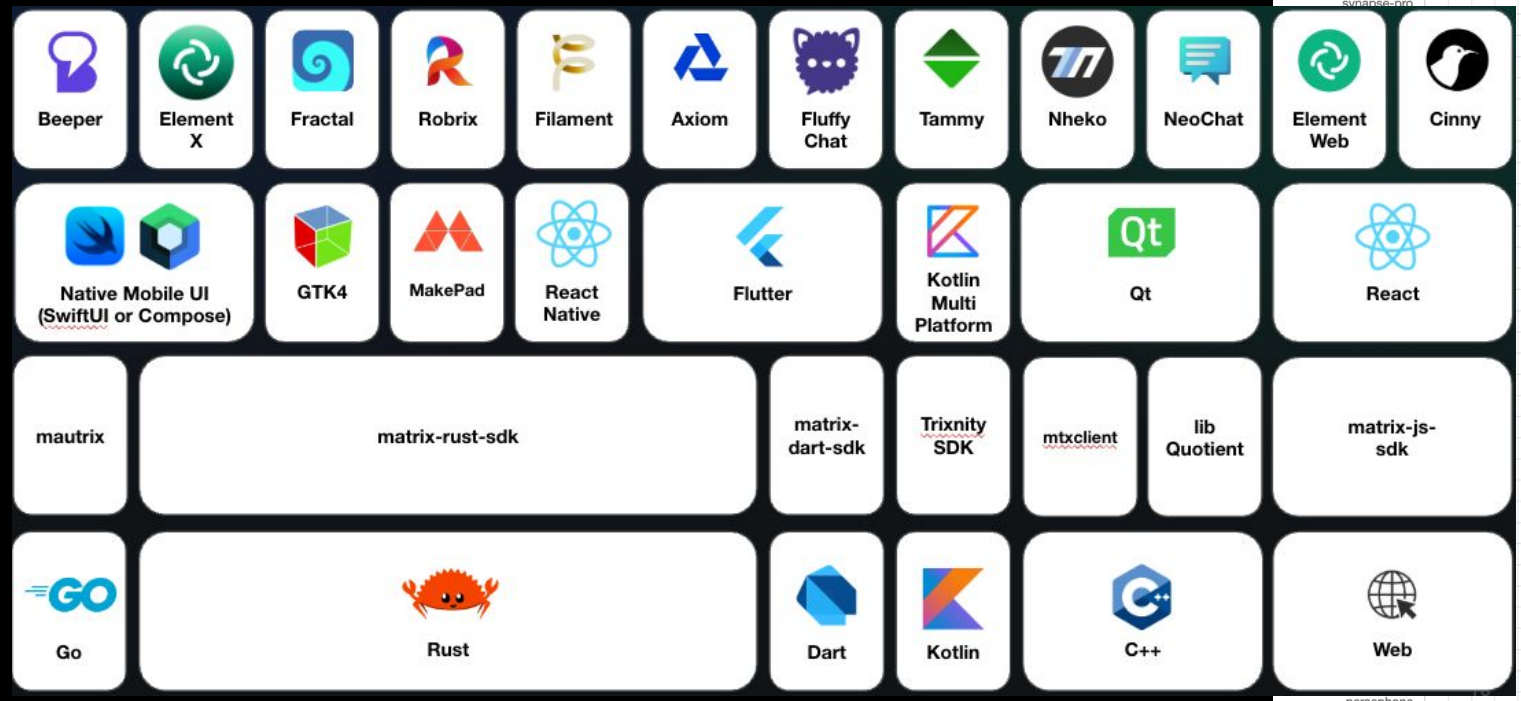
Element Multiple
NI²CE NATO ACT
MijnBureau Netherlands
BwMessenger Germany
BundesMessenger Germany
openDesk Germany
TI-Messenger Germany
Logineo NRW Germany
Tchap France
LaSuite France
SAFOS Sweden
ByCS-Messenger Germany
Schulchat RLP Germany
AOK Germany
TK Germany
Barmer Germany
Famedly Germany
FITKO Germany
grnet Greece
ePost Switzerland
luxchat Luxembourg
luxchat4gov Luxembourg
Rocket.Chat Multiple
Delta Ukraine
Meedio Multiple
BEAM Belgium
KBS Germany
KKH Germany
IKK_Classic Germany
SSH Multiple

Meanwhile the ecosystem continues to grow…

More and more implementations…

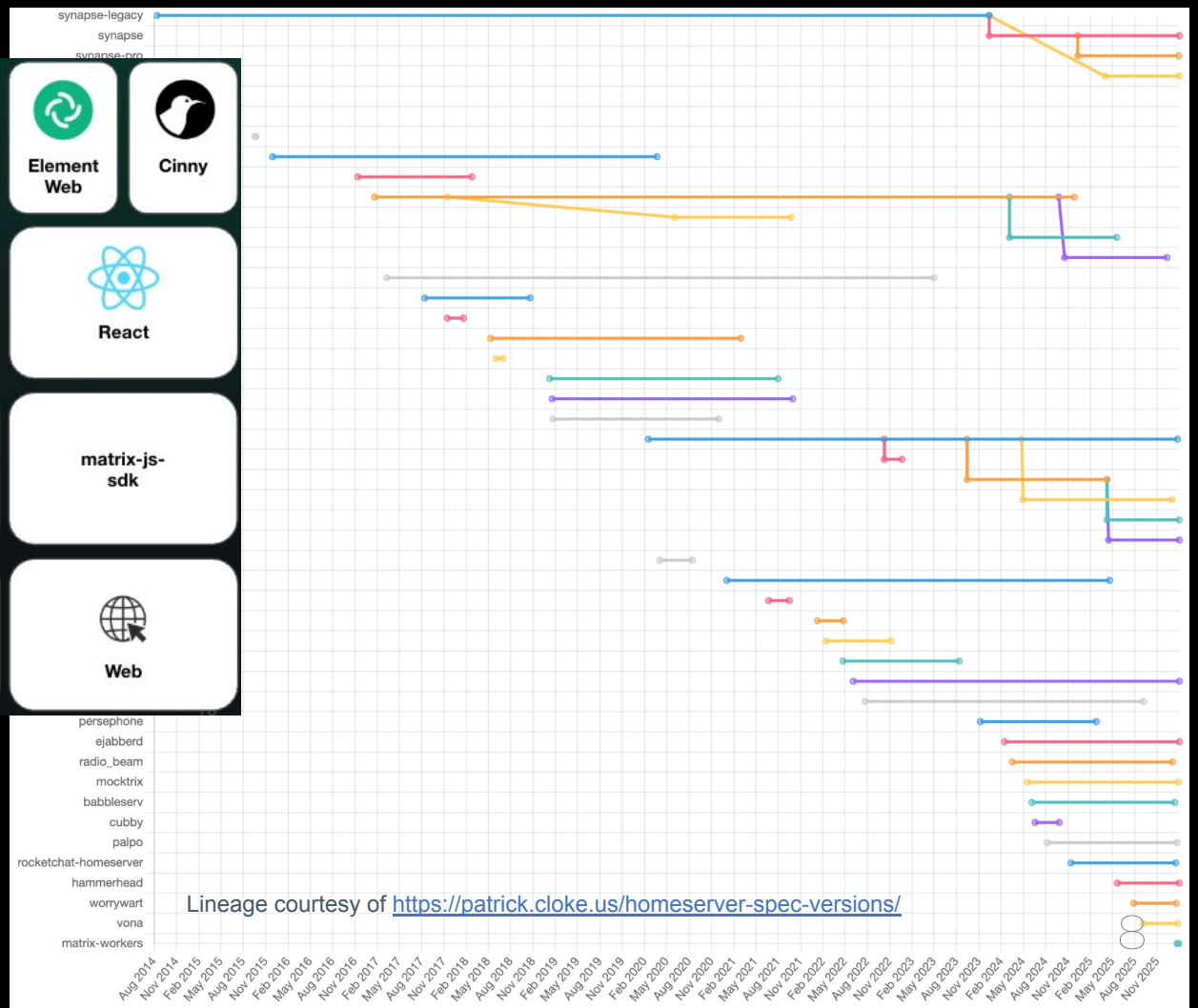
matrix

Tens of clients and sdks built on a myriad of technologies...



... and more!

A growing family of servers...



Many Matrix vendors & builders…

matrix

SSH

ERCOM
Cyber Solutions by Thales

BEEPER

rocket.chat

element

Famedly

meedio
THE SAFE PLACE TO MEET

connect2x

FlatTurtle

fairkom

LIBERBYTE
FREE YOUR DATA

infra.run
SERVICE

gematik

PREMIER MINISTRE
Liberté
Égalité
Fraternité

Direction interministérielle du numérique

LU-CIX

INDIE HOSTERS

And more!

blender®

Pangea Chat

So where is Matrix going?

Matrix's mission is to become the open communication layer of the of the Web.

[matrix]

1.2M registered monthly active users (based on Synapse phonehome stats)

How do we close the gap?

We're seeing significant uptake in five main areas

matrix

1. FOSS/decentralisation/privacy enthusiasts wanting to run their own communication infrastructure for themselves or their organisations.
2. Public sector organisations (central government, local government, healthcare, defence, education) and their suppliers, wanting to provide digitally sovereign Teams/Slack/WhatsApp alternatives on Matrix (e.g. Element)
3. Product companies wanting to add chat to their existing apps (e.g. Reddit)
4. Organisations building consumer-facing communication apps built on Matrix (e.g. Beeper)
5. Folks using Matrix to connect LLMs.

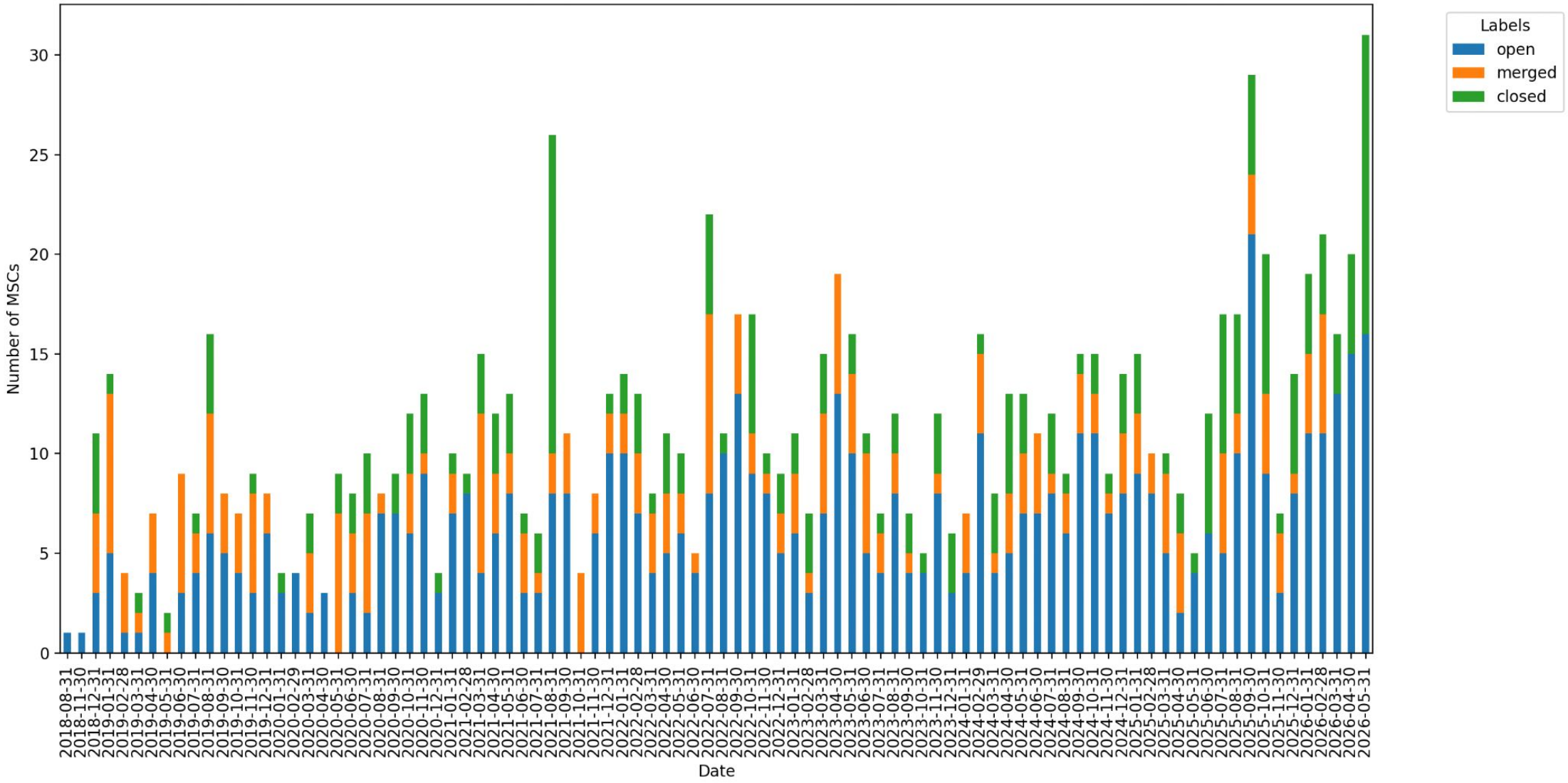
Foundational spec work

- Explain the direction of travel to help manage expectations (👋)
- Aim to shrink rather than grow the spec.
- Evolve the spec incrementally where possible, only doing big changes if there is no choice.
- Ensure that weaknesses in the federation API are fixed. This is Project Hydra.
- Ensure the remaining core features needed to compete with centralised apps (aka Matrix 2.0) are specced.
- Build out Trust & Safety features and tooling.
- Improving encryption; Adding PQ to today's Matrix; client-controlled membership; MLS
- Improving metadata footprint on the server
    - Obfuscating user IDs per room (provided by Hydra phase 2)
    - Encrypting state events
    - Encrypting aggregations
    - Obfuscating membership
- Ensure that other MSCs requested by the broader community also progress (e.g. custom emoji)

Strategic areas

- In the end, organisations will only federate with the public Matrix network if there is value to be had from the network. This means:
  - Minimising abuse.
  - Making it easy to finding people to talk to
  - Making it easy to find rooms to talk in
  - Encouraging useful services to expose themselves on the public network - bridges, bots, agents, etc.
- Steering clients towards good UX (e.g. making it easy to onboard, hiding the scary decentralisation bits)
- Improving decentralisation - avoid matrix.org becoming a single point of failure
- P2P Matrix (see more on that later)
- DMA: still provides a way to encourage large players to join Matrix - we haven't given up on doing this with WhatsApp yet; should the Foundation be funded to do so we would continue to work on making DMA interop a success.
- Proliferating Matrix URLs
- Further improving the website.

MSCs by month



Work in flight

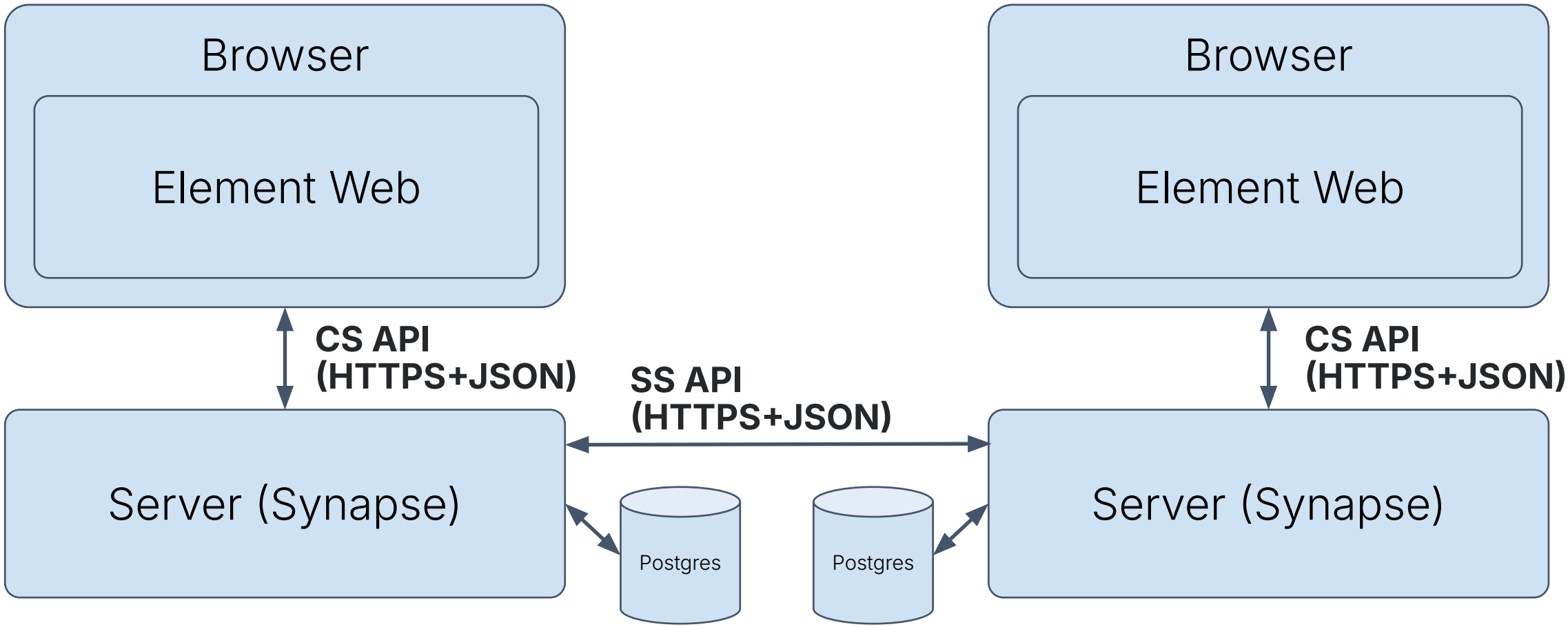
- **Improve today's Matrix**
  - Hydra Phase 2: State DAGs ((MSC4242 - **Synapse PR**: PR19718)
  - Hydra Phase 2: Account as keys
  - Hydra Phase 3: Finality and State Res 3 (ERA)
- **Unblock remaining Matrix 2.0 components**
  - MatrixRTC (MSC4143) + components (MSC4075, 4310, 4196, 4195)
    - Delayed events (MSC4140) + Sticky events (MSC4354)
  - Sliding Sync (MSC4186) + Exts (MSC3884, 3885, 3959, 3960, 3961, 4306, 4308, 4480)
- Why isn't this going faster?
  - Security (e.g. Hydra) comes first.
  - The Spec -> Impl -> Prove -> Rethink cycle is real.
  - The more popular Matrix gets, the more feedback there is to incorporate.
  - All of the SCT are volunteers, and dayjob commitments can get in the way.

So let’s talk P2P.

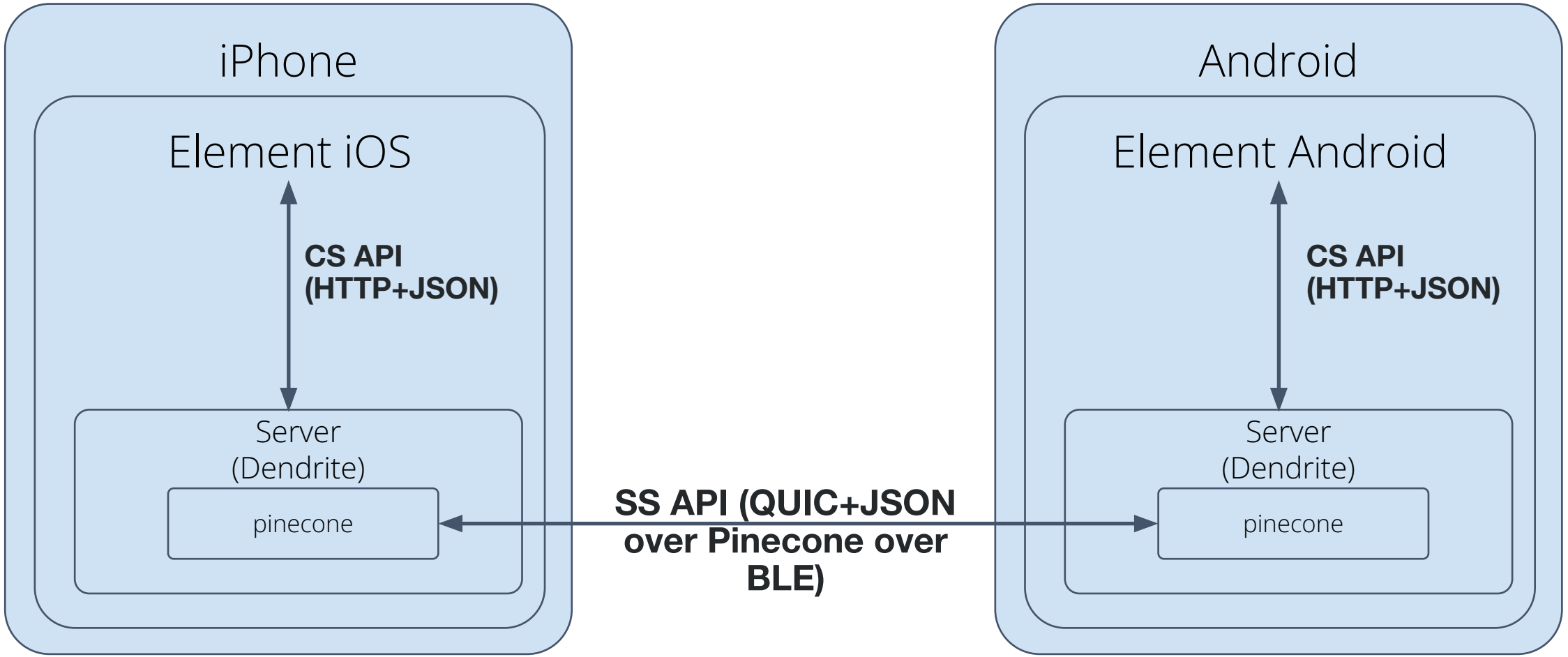
P2P Matrix: 2020-2023

- “What if you ran your Matrix server inside the client?”
- Embedded Dendrite (golang server) inside Element Classic iOS & Android
- Fixes metadata exposure to servers… by removing the server.
- Solves client-controlled cryptographic group membership (as state resolution runs clientside)
- Supports fully disconnected mesh networks, and innoculates against surveillance capitalism
- For connectivity we tried global overlay networks:
  - …first libp2p…
  - …then yggdrasil…
  - …then pinecone (a yggdrasil variant supporting highly dynamic networks)
- Three main things went wrong:
  - We hit the limits of Matrix’s resilience to network partitions (at the time).
  - Building a robust P2P overlay network is an open area of research in its own right.
  - Element ran out of funding for next gen work, focusing instead on keeping the lights on.

Before P2P...


Browser
Element Web
Browser
Element Web
CS API
(HTTPS+JSON)
SS API
(HTTPS+JSON)
CS API
(HTTPS+JSON)
Server (Synapse)
Postgres
Postgres
Server (Synapse)

P2P via Dendrite + Pinecone (~2023)



P2P Matrix: a new hope!

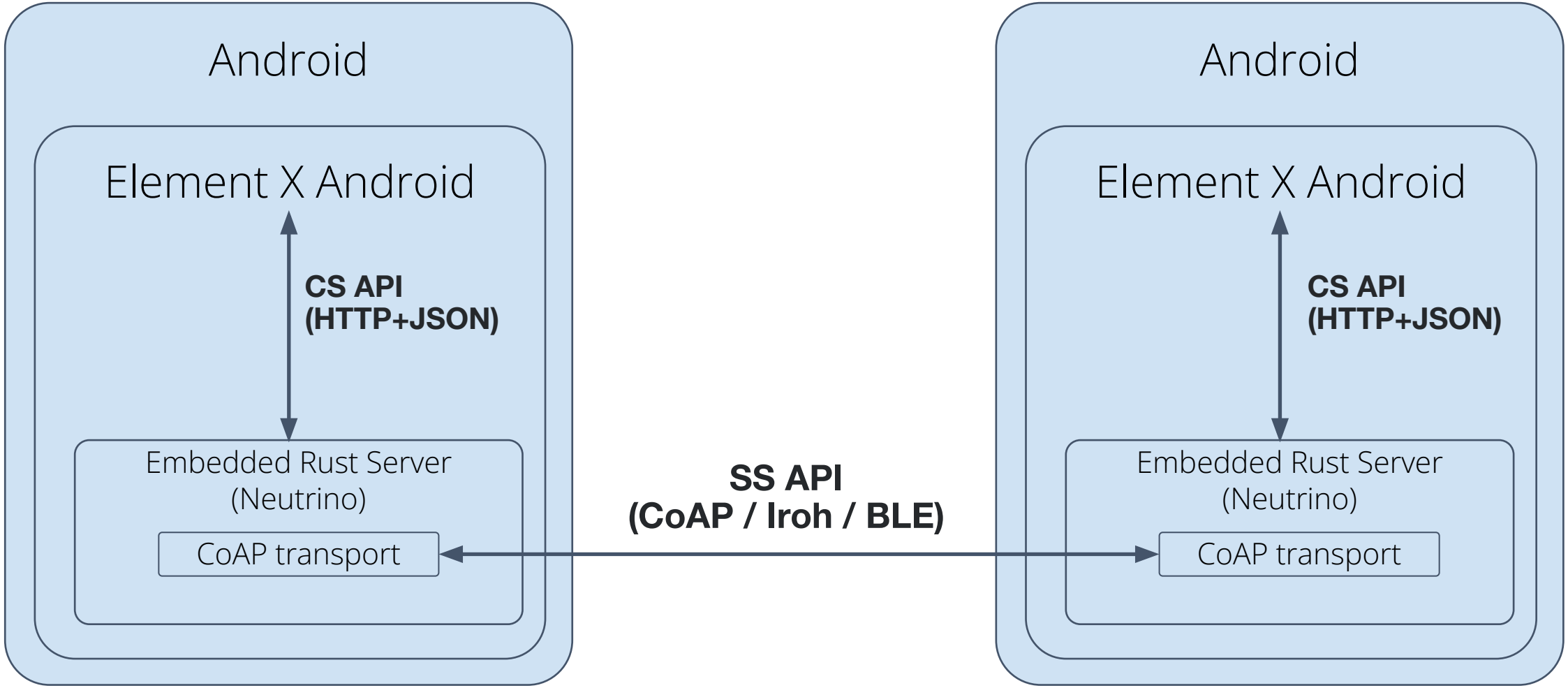
- **Fast forward to 2026:**
  - Element found a customer to fund P2P Matrix again: the Dutch Government!
  - This time it's different.
    - We've fixed Matrix to be fully resilient to network partitions (more on this later)
    - We're not going to build an overlay network.
    - Instead, we'll target smallish P2P meshes, which can default to using normal Matrix servers for store-and-forward.
  - One possible approach:
    - For instance, using multihomed accounts, imagine if @bob:example.com could live on both his normal example.com homeserver but *also* on a server on her phone?
    - Then if @alice:example.com wants to talk to him, she could try looking him up on whatever mesh she's on - e.g. using Multicast DNS (Bonjour / Zeroconf), or BLE broadcast, etc.
    - If she can talk direct, then hooray - we have P2P! If not, she can relay via the example.com homeserver like normal Matrix today.

Introducing Neutrino

- **So, we've built a new P2P Matrix stack.**
  - Element X (rather than Element Classic)
    - Powered by matrix-rust-sdk
  - Talking to a new, tiny, embedded Matrix server implementation in Rust: Neutrino
    - https://github.com/element-hq/neutrino
  - Uses CoAP as a low bandwidth transport.
  - Uses whatever overlay is available: currently using
    - Iroh for "connect to this key please" and
    - BLEW for BLE mesh connectivity
- This comes with **very important caveats**
  - This is very very early and experimental: we are opening the repository for the first time today.
  - **Deliberately not yet secure for untrusted networks.** (Does not set/check event signatures).

Neutrino P2P Demo

P2P via Neutrino



Why Neutrino?

- Written specifically to experiment with P2P Matrix dialects.
- Written specifically as an embedded homeserver rather than forcing a serverside homeserver to run clientside.
- It is **not** intended to run as a standalone homeserver.
- It only implements the very latest room versions (Hydra Phase 2 and 3) - e.g. State DAGs: MSC4242)
- Gives us somewhere to evaluate Hydra and state resets in the harshest conditions (i.e. P2P)
- Neutrino is not the future of Synapse or Synapse Pro, although we expect Neutrino and Synapse to share Rust code in future
- This not a re-run of Dendrite (whose learnings got subsumed into Synapse & Synapse Pro)
- We also want somewhere to experiment freely with new ideas (e.g. temporal state storage) which can get backported and/or shared with Synapse & Synapse Pro.
- Disclaimer: Neutrino development has been heavily LLM-accelerated. We are very mindful of the ethical and environmental concerns of LLM use, but concluded that using big tech to accelerate decentralisation of communication is net positive in this instance.

What's next for Neutrino?

- Figuring out store & forward semantics for interop with normal Matrix
- Figuring out the maze of accounts-as-keys?

MSC	User ID Format	Who owns the signing key?	Signing key scope	Localpart location	Domain location
Now	@localpart:domain	Server	Per-Server	User ID	User ID
1228	^user-room-key	Server	Per-User, Per-Room	m.room.member content	m.room.member content
2787	^0x01user-delegated-key	Server (attested by Client)	Per-User, Per-Room	Undefined	m.room.member content
4014	sender_key	Server	Per-User, Per-Room	m.room.member content	m.room.member content
4243	@account-key:example.com	Server	Per-User	Server	User ID
4345	@localpart:server-key	Server	Per-Server	User ID	m.server.participation content
4348	@account-key	Client?	Per-User (attested by Server)	Server	m.server.participation via Server signature
4430	@account-id:server-key	Server	Per-Server	m.room.member content	m.room.member content

- Adding support for untrusted networks.
- Adding more features
- Potentially: using it as a testbed for Hydra Phase 3 (ERA Finality)
- Potentially: using it as a testbed for the State Res v3 changes that go with Finality.

Another path…

Let's talk about MLS

matrix

- MLS is Messaging Layer Security: IETF RFC 9420
- Provides continuous key agreement across a set of endpoints
- MLS benefits:
  - Many eyeballs thanks to IETF process.
  - Provides a solution to cryptographic group membership (across a set of devices)
  - Better Post Compromise Security (if a sender starts a new session, all senders update)
  - Better performance O(log N) in best-case conditions, but we're not seeing it in practice.
- MLS drawbacks:
  - Vanilla MLS needs group changes to be centralised for a given room, creating a SPOF.
  - Particularly problematic for larger or unreliable (e.g. P2P) networks.
  - MLS expects to drive the group membership from that centralised server, missing the whole point of Matrix's decentralised access control & state resolution.
  - Large key packages (somewhat addressed by Partial MLS)
  - No cryptographic deniability (unlike Signal or Olm/Megolm)

Decentralised MLS

matrix

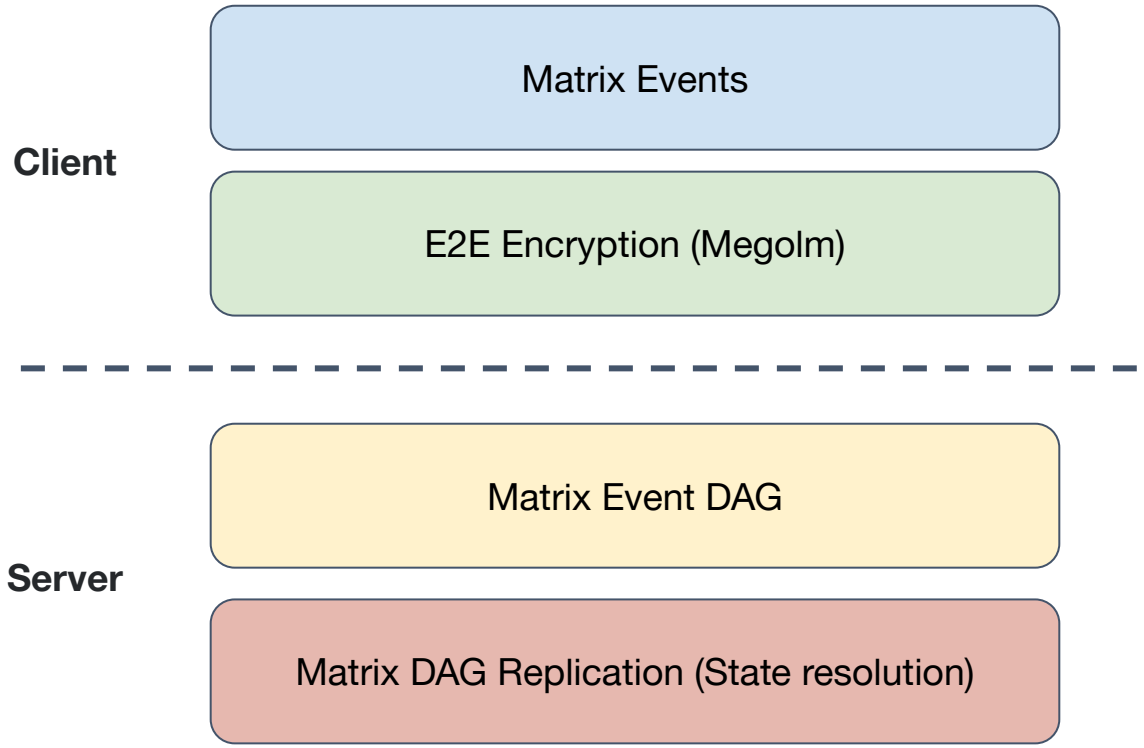
- Back in 2020 we defined Decentrali**s**ed MLS, defining a way to merge together MLS trees which evolve separately during a network partition.
- This came at the expense of forward secrecy (as you have to keep historical key material around to merge trees).
- In 2025, Phoenix R&D proposed Decentrali**z**ed MLS, an IETF draft which uses Puncturable Pseudo-Random Functions (PPRFs) as a way to improve forward secrecy.
- Germ have proposed Distributed MLS, another IETF draft which is similar to Megolm, but uses per-sender MLS trees to encrypt - O(N) performance.
- For Matrix: we don't want to be locked to any specific encryption mechanism.
  - We do want to improve our metadata footprint however…
  - …and we do want to make our group membership client controlled.
  - We also want the option of using vanilla centralised MLS (which Hydra Phase 3 gives us)

Client-controlled group membership.

matrix

- Probably the biggest defect of Matrix today is the fact that the server unilaterally controls who participates in its conversation.
- Instead, the only people who should be able to add new users should be users within those rooms who have permission to do so.
- Neutrino provides one way of addressing this, by running Matrix state resolution in the server in the client: letting the client prove via state res that they are allowed to do certain operations.
- Any servers which do end up participating in a room end up seeing the room's metadata, however.
- **What if we could make Matrix entirely metadata-protecting while also solving the client-controlled group membership problem?**

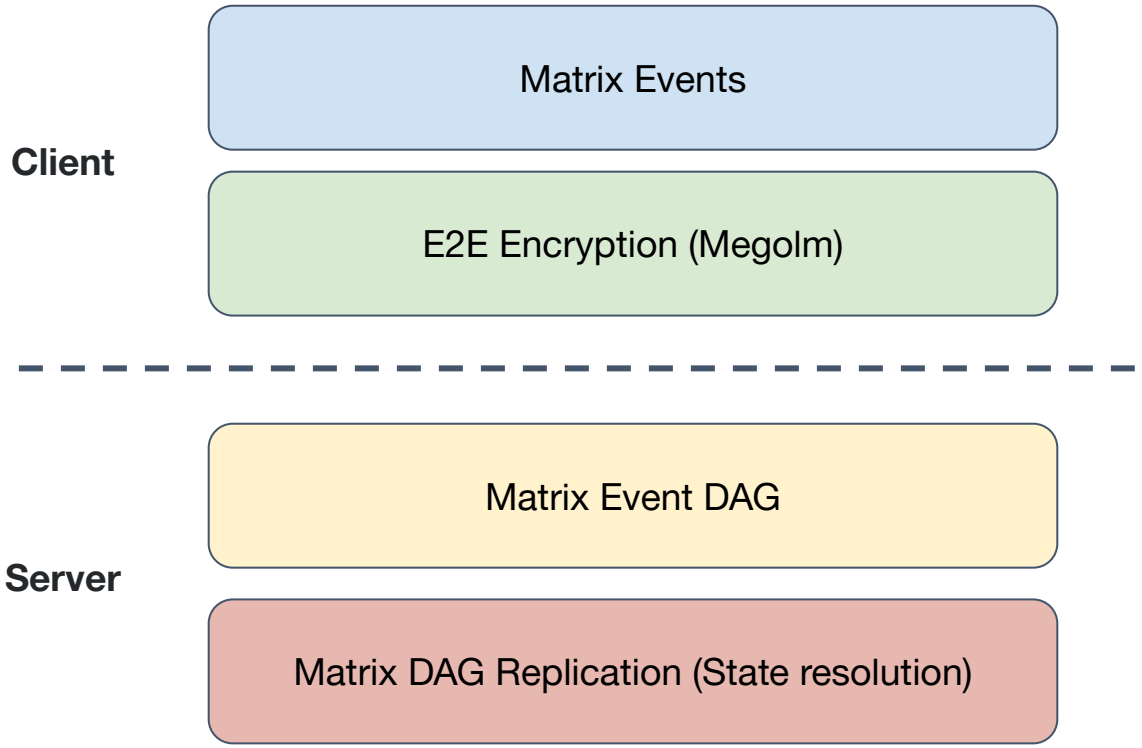
Today's Matrix



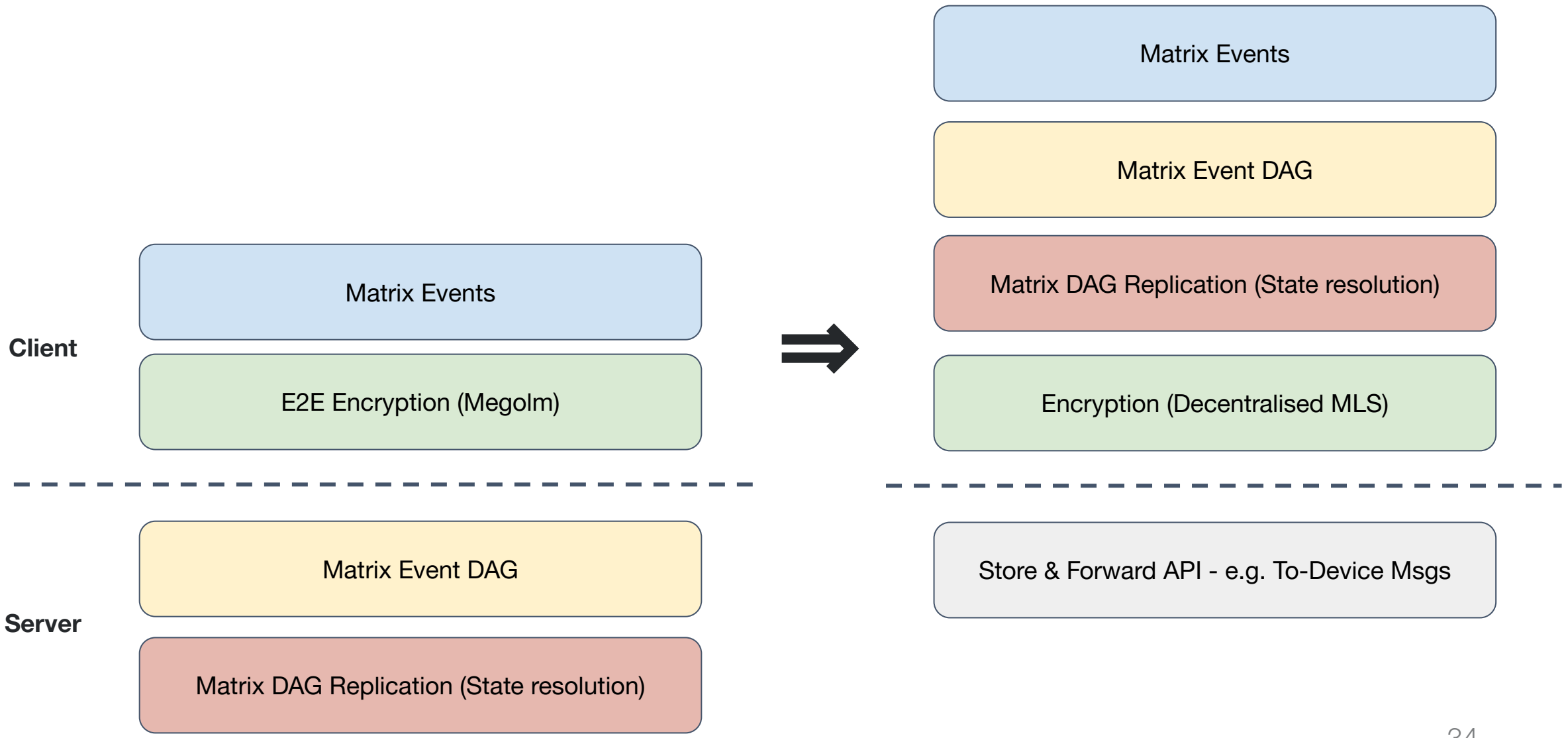
Today's Matrix

matrix

What if…



Client-side State Res with DMLS



matrix

Not quite P2P, but alarmingly close.

Introducing Tachyon

- Tachyon is an experiment to demonstrate Matrix tunnelled over Decentralised MLS.
- Decentralised MLS synchronises keys for E2EE across a set of devices without a central control point
- Tachyon extends matrix-rust-sdk to speak DMLS.
- **Not** a clientside homeserver.
- Sends and receives DMLS packets over normal Matrix to-device messages
- Inside the packets is the normal Matrix DAG, controlled by the client - not the server.
- The Matrix DAG defines which users are in the room via normal decentralised Matrix access control.
- DMLS then restricts which devices participate in the room, driven by the Matrix DAG within the DMLS payloads.
- **As a result, you get fully metadata-protected Matrix conversations which look like normal Matrix rooms, but the server doesn't even know they exist. They're just a cloud of DMLS.**
- This approach could be used on top of **any** store and forward mechanism: SMTP, XMPP, ActivityPub, AT Proto private stores, etc.

Tachyon Demo

https://aurora-tachyon.netlify.app

What's next with Tachyon?

matrix

- The current code is not secure or audited, and **should not be used** for anything serious yet.
- Try to tame the code into something publishable (currently a 120KLOC PR against matrix-rust-sdk)
- Figure out how to encrypt (or not) invite handshakes.
- Get it secure.
- Get it resilient to DoS.
- Consider structuring it as a Pantalaimon-style daemon so any Matrix client can use it as a way to get metadata-protected traffic (rather than requiring clients to use rust-sdk)?
- Consider doing a demo of layering it over another protocol to illustrate how it could be used to unleash metadata-protected E2EE messaging for other protocols?
- Maybe combine with Neutrino somehow?

For Matrix to prevail, we need your support!

- **Use it!** Friends don't let friends use closed chat services!
- **Continue to build cool stuff on Matrix!**
  Clients! Servers! Bots! Bridges! Integrations!...
- **Become a Matrix vendor and grow the ecosystem!**
  Host! Integrate! Develop! Resell!…
- **Join the Working Groups!**
- Getting better but the Foundation still isn't sustainable!
  - Fundraising **right now** to support core spec work, trust & safety work, running the matrix.org infrastructure and governance work
- **Join the Foundation:** https://matrix.org/membership/**!**
- **Buy Matrix solutions from vendors who fund upstream maintenance** and development. Otherwise there will be no upstream projects.

[matrix]

Thank You

https://matrix.org

@matrix@mastodon.matrix.org

@matrix.org

@matthew:matrix.org

matthew@matrix.org