



[matrix]

The Matrix State of the Union

The Matrix Conference 2024

matthew@matrix.org

@matthew:matrix.org

The story so far...

**Sept 2014:
Matrix
launches**

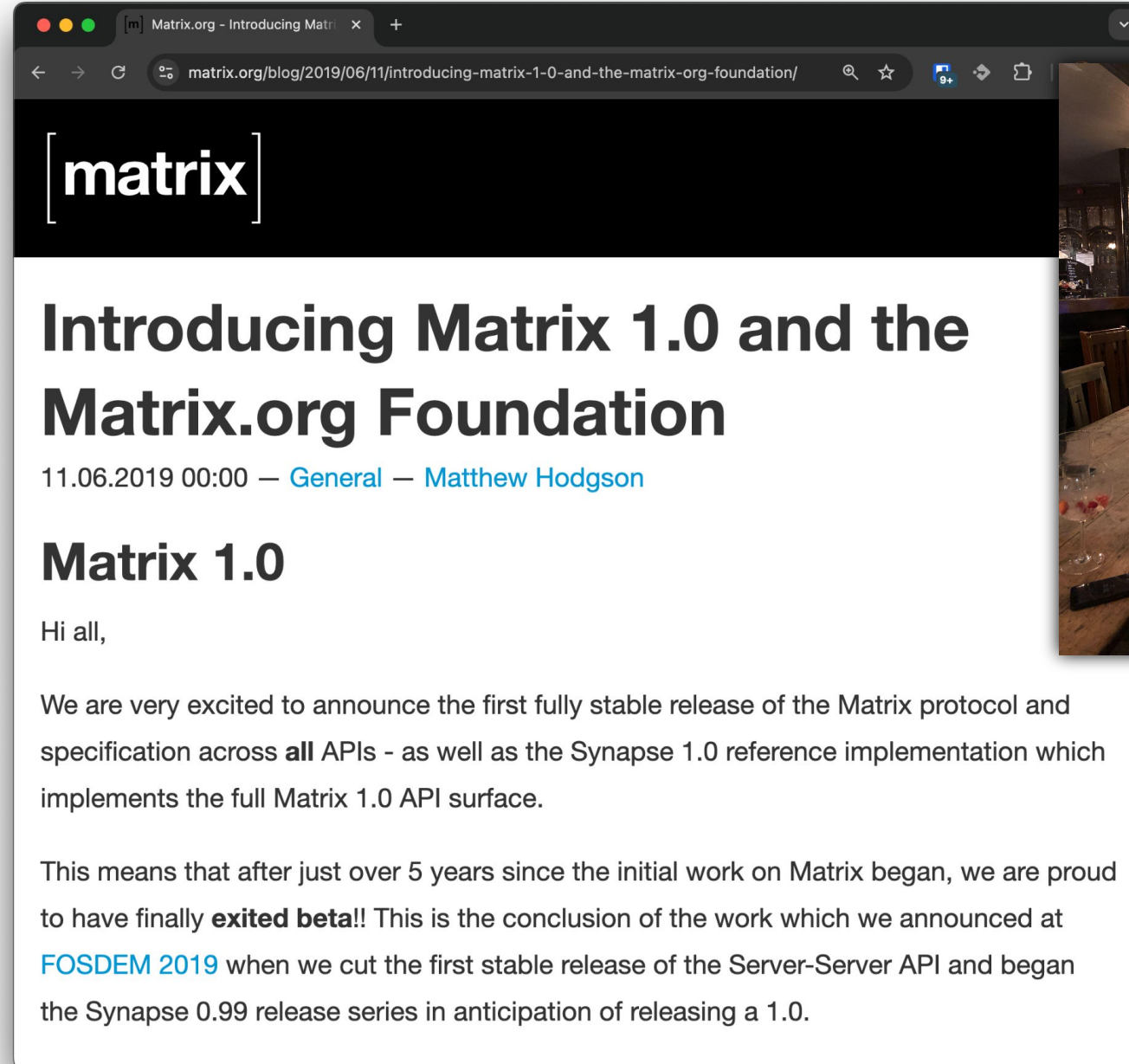
“Make it work!”



June 2019: Matrix 1.0

“Make it work
right!”

[matrix]



[matrix]

Introducing Matrix 1.0 and the Matrix.org Foundation

11.06.2019 00:00 — [General](#) — [Matthew Hodgson](#)

Matrix 1.0

Hi all,

We are very excited to announce the first fully stable release of the Matrix protocol and specification across **all** APIs - as well as the Synapse 1.0 reference implementation which implements the full Matrix 1.0 API surface.

This means that after just over 5 years since the initial work on Matrix began, we are proud to have finally **exited beta!!** This is the conclusion of the work which we announced at [FOSDEM 2019](#) when we cut the first stable release of the Server-Server API and began the Synapse 0.99 release series in anticipation of releasing a 1.0.



Sept 2024: Matrix 2.0...

[matrix]

**“Make it work
FAST!”**

- Project Announced at FOSDEM 2023
 - **Goal: make Matrix able to outcompete mainstream apps on performance and usability.**
- First implementation (Element X “Ignition”) at Matrix Community Summit, Sept 2023
 - Many features missing (although those there worked)
 - Intended only for existing Matrix power-users.
- Status update at FOSDEM 2024, but still not ready.
- And now today...



Matrix 2.0 is here.



Everyone can now use the MSCs that
make up Matrix 2.0.

Matrix 2.0 is here.

1. Synapse now ships with (Simplified) Sliding Sync support by default (MSC4186)
2. Next Gen Auth is now available with Matrix Authentication Service (MSC3861) - coming soon on matrix.org itself.
3. MatrixRTC Native VoIP with E2EE is live in Element Call (MSC4143) (and integrated with Element X & Element Web)
4. Invisible Encryption is under way in Element X (MSC4153)

=> **The MSCs have stabilised and are on track for FCP.**

We will bump the major version of the spec to 2.0 once they (and their dependencies) merge.

1. Simplified Sliding Sync: SSS (MSC4186)

- “Instant login, Instant Launch and Instant Sync.” - but simpler.
- The original Sliding Sync MSC (MSC3575) has been simplified.
 - No more sliding windows; the client manages the roomlist order.
 - The roomlist is paginated in the background, as clients need to know all rooms.
 - Same performance as original SS, but doesn't try to minimise bandwidth.
 - Simplified Sliding Sync is pretty much a subset of Sliding Sync.
- Now natively implemented in Synapse (thanks to DINUM)
- Shipping in Synapse, enabled by default since 1.114
- More refinements are still possible (e.g. paginating incremental sync), but these will be later MSCs - no more creature feep!

Demo!

The Fate Of The Sliding Sync Proxy

- We can't maintain both SS and SSS, so heads up:
 - The Sliding Sync proxy will be deprecated in ~1 month (~week of Oct 14th)
 - The old SS code will be removed from matrix-rust-sdk in ~2 months (Nov 25th).
- This is intended to give folks adequate time to:
 - Implement SSS in more servers
(e.g. turning SS in conduit into SSS should be easy)
 - Upgrade to their server to a version which supports SSS
 - Decommission their proxies
- (Of course, if someone wants to fork the SS proxy and add SSS support, e.g. for Dendrite users, please feel free!)

2. Next Gen Auth (MSC3861)

- Why Next Gen Auth (Native OIDC)?
 - **Matrix is a communication protocol, not an authentication protocol.**
 - Matrix's current auth is *very* similar to OAuth 2.0, but doesn't benefit from all the effort that goes into improving the OpenID ecosystem.
 - We've already had to fix some security issues which OIDC had already solved
 - There's a set of other security deficiencies too (see the MSC)
 - It brings us a tonne of new features, including everything that the broader OIDC ecosystem comes up with over time.
 - **This is not SSO or Social Login** (although you can use it for that if you want)
 - It's the next generation of the existing auth APIs, adopting an existing standard.
- It's not just us: XMPP and ATPProto also have projects on the go to migrate to OIDC for auth.

2. Next Gen Auth (MSC3861)

- Matrix Authentication Service is usable for new deployments!
 - Login via QR code, complete with E2EE identity!
 - Support 2FA, MFA, Passkeys etc via an OIDC IdP!
 - No more implementing Matrix auth flows on every client (and homeserver)!
 - Users only ever hand their password to their server, not random clients
 - Consistent auth and account management experience across apps
 - Integrates seamlessly with password managers
 - Lets users share authentication between apps (SSO)
 - Finally gives access-token refresh by default
 - OIDC scopes let users control what features an app can access.
 - It will be integrated natively into Synapse in future. 🦀

Demo!

2. Native OIDC (MSC3861)

- We recommend MAS for new deployments.
- The adventurous synapse admin can try migrating via **syn2mas!**
- matrix.org has to:
 - migrate 9,390,992 accounts...
 - including all the social login users...
 - and all the bridged users...
 - and sort out guest access
- ...so isn't running MAS yet, but will be by the end of 2024.
- (As a convenience to Element X users, there's a temporary off-spec registration helper available on matrix.org to let them register pre-MAS).

3. MatrixRTC and Native VoIP (MSC4143)

- We are finally in position to migrate from the old legacy mix of 1:1 VoIP and Jitsi to a single native-Matrix VoIP standard.
- This is MatrixRTC (MSC4143): Signalling for setting up VoIP calls/conferences over Matrix with E2EE.
- Deliberately supports alternative backend implementations for the media layer (similar to E2EE agility, so we can migrate to future implementations as needed)
 - LiveKit SFUs
 - Full mesh
- Implemented in Element Call, enabled now by default in Element X, Web/Desktop (Nightly/Develop)!

3. MatrixRTC and Native VoIP (MSC4143)

- The MatrixRTC MSC layer cake:
 - MSC4143 - MatrixRTC
 - MSC4075 - MatrixRTC Call Ringing
 - MSC4195 - MatrixRTC using LiveKit backend
 - MSC4196 - MatrixRTC voice and video conferencing application m.call
- With dependencies on:
 - MSC3779 - “Owned” state events
 - MSC4140 - Cancellable delayed events
- The MSCs have been evolving a lot, but this stack has now stabilised and (with a bit more polishing) should be on track to be FCP’d.

Demo!

4. Invisible Crypto (MSC4153 et al)

- Thesis: E2EE should be invisible in Matrix clients, just as it is on WhatsApp, iMessage, Signal, etc.
- The only reason it isn't is because of the incremental transition to E2EE back at Matrix 1.0.
- First off: fix Unable To Decrypt errors.
- Then, the idea is:
 - All devices should be verified (cross-signed) by their owner.
 - MSC4153 explicitly proposes excluding untrusted devices.
 - Therefore, no more warnings about untrusted devices.
 - Once MSC4153 lands, **you will not be able to encrypt to or decrypt from devices which have not been cross-signed by their owner.**

4. Invisible Crypto (MSC4153 et al)

- **Next steps:** no more “grey shields”:
(“Authenticity could not be determined on this device”):
 - Authenticated backup: MSC4048 (so keys from backup can be trusted)
 - Including device keys with Olm-encrypted events: MSC4147 (so deleted devices can be trusted)
- Dehydrated devices: MSC3814 (no more UTDs if you log out everywhere)
- Sharing keys when joining a room/space
- Then, at last, Trust On First Use (TOFU)
 - So warn when user identity changes, even if you haven’t verified them.
- In future: Key Transparency
 - Similar to WhatsApp and iMessage who already run KT servers: TOFU backed up with an assertion from KT servers... but decentralised.

Bonus: Authenticated Media

- Not strictly Matrix 2.0, given it already shipped in 1.11
- Requires users and servers to be authenticated in order to view media.
- Critical trust & safety improvement to avoid Matrix being abused as a free CDN (and so expedited as such).
- Enabled on Sept 4th on Matrix.org.
- Has been a bit of a mixed bag:
 - Adding an HTTP Authorization header is more painful than anticipated on Web.
 - The header does not play well with CDNs.
- It's rolled out now and is generally working (thanks to everyone who updated for it), but there's scope to iterate on this going forwards.


So, welcome to Matrix 2.0!

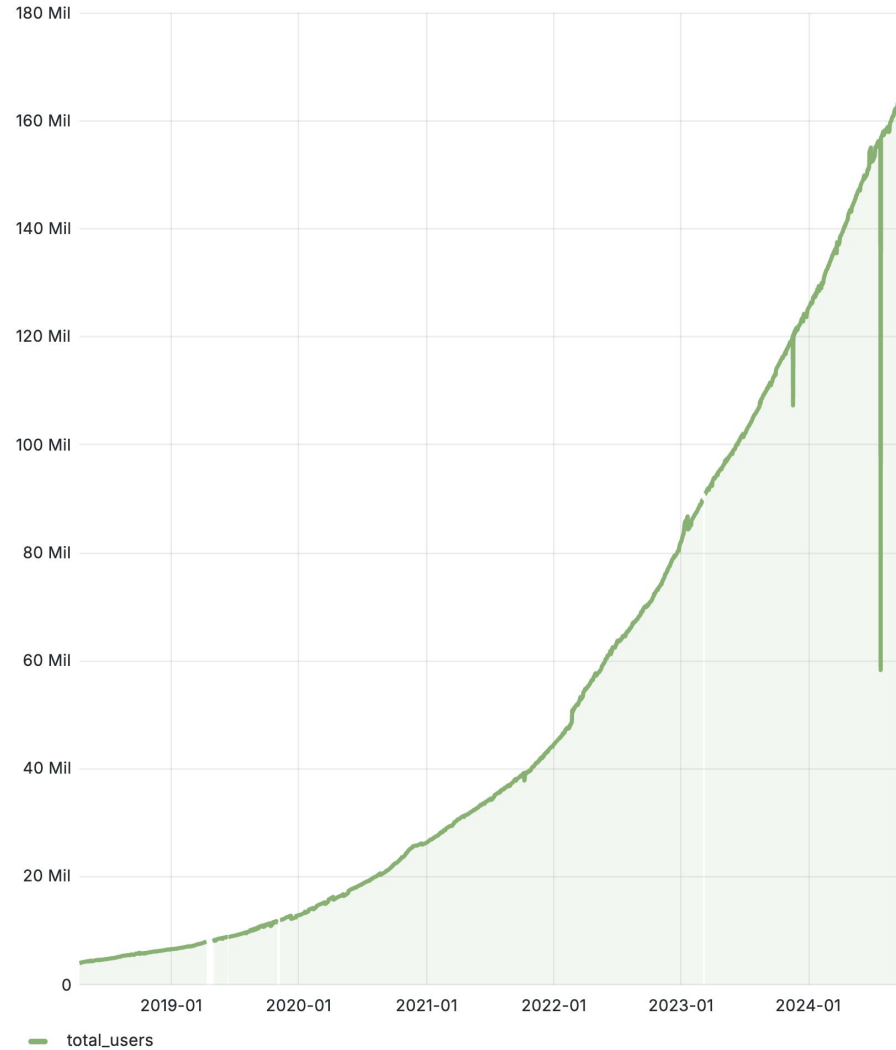
Try it all at beta.matrix.org today
or run it yourself;
rolling out incrementally on matrix.org


Spread the word!

- Matrix 1.0 has ended up with its fair share of complaints - whether that's due to client UX problems, encryption problems, or Synapse being resource heavy.
- **This is our chance to fix that story.**
- Please come out of here telling everyone to give Matrix another shot, and that performance and usability is solved - and, what's more, they can play with it today (at last)!
- Matrix 2.0 is not just for powerusers and early adopters any more - it's for everyone :)

Meanwhile in Matrix...

Total Federated Known Users 



Monthly Active Users 



Meanwhile in Matrix...

- Funding continues to be tough.
- The Foundation is not cash positive, even for its current small set of activities.
- Trust and Safety is a particular challenge - while volunteers are helping (thank you!) we need more folks working on this full time.
- The economics of Matrix are still not clear: open source in general still feels like it is seen as a resource to be exploited, rather than funded.
- Please buy Matrix solutions from vendors who financially support the underlying open source projects...
- ...or directly support the Foundation yourself.
- <https://matrix.org/support>

What's next?

- State resolution redux.
- Finishing invisible crypto
- Keep polishing performance and usability
- Get enough funding to work on account portability & multihoming again.
- Matrix 3.0 options?
 - “Make it safe” - focus entirely on fixing Matrix’s trust & safety tooling?
 - MLS + Matrix - hand over cryptographic group membership to MLS?
 - Converge with MIMI?
- Place your bets!

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!

@matthew:matrix.org

matthew@matrix.org

<https://matrix.org>

@matrixdotorg

@matrix@mastodon.matrix.org