Beyond the Matrix

Extend the capabilities of your Synapse homeserver

a.k.a. “Modules go brrr”
Who am I?

Brendan Abolivier
babolivier@matrix.org
@babolivier:element.io

Software developer at Element
Core committer on Synapse
What is Synapse?

The Matrix Specification (Client/Server API)

Other Clients:
- Quaternion (Qt/C++)
- gomuks (CLI/go)
- Fractal (Gtk+/Rust)
- NeoChat (Qt/C++)
- Nheko (Qt/C++)
- weechat-matrix
- ...and many many more

Other Servers:
- conduit (Rust), construct (C++)...
- Bridges, Bots, Integs...

Other Services:
- Bots
- Integs...

client-side

server-side

Synapse (1st gen Matrix Server)

Dendrite (2nd gen Server)

Matrix Application Services and Bridges
What is Synapse?

Synapse (1st gen Matrix Server)

Dendrite (2nd gen Server)

Matrix Application Services and Bridges

Other Servers: conduit (Rust), construct (C++)...

Other Services: Bridges, Bots, Integs...

Other Clients:
- Quaternion (Qt/C++)
- gomuks (CLI/go)
- Fractal (Gtk+/Rust)
- NeoChat (Qt/C++)
- Nheko (Qt/C++)
- weechat-matrix

...and many many more

Hydrogen

The Matrix Specification (Client/Server API)

Other Services:
- Bots, Integs...

Client-side:
- Element Web
- Element iOS
- Element Android
- Hydrogen
- Element R

Other Apps:
- gomuks (CLI/go)
- Quaternion (Qt/C++)
- Fractal (Gtk+/Rust)
- NeoChat (Qt/C++)
- Nheko (Qt/C++)
- weechat-matrix

Server-side:
- matrix-react-sdk
- MatrixKit (iOS)
- matrix-android-sdk2
- matrix-ios-sdk
- Hydrogen SDK
- matrix-rust-sdk
- matrix-rust-sdk-crypto

Other Clients:
- Element Web
- Element iOS
- Element Android
- Hydrogen
- Element R
What is Synapse?

- Main Matrix homeserver implementation
- Written in Python
- Maintained by the Matrix.org Foundation
Modules

Modules add **extra features** to Synapse, which are either **implementation-specific** or don’t fit in the Matrix spec, or need more **flexibility** than config options.

e.g. spam checkers, authentication providers...
A module is a **Python class**, which registers **callback functions** for Synapse to call on specific **operations**, and can **interact with Synapse** using the **ModuleApi** class.
How to write a module

Resources for writing a Synapse module:

- List of callbacks with documentation and examples
- Cookiecutter template for modules
- ModuleApi class with docstrings

All are linked to from the talk’s page on fosdem.org
Let’s write a module!

Let’s write a **simple module** that changes a **room’s avatar** to reflect the **last person** who sent a message into it.

To do this, we’ll use:

- `on_new_event`: callback called by Synapse on every new event
- `create_and_send_event_into_room`: ModuleApi method to send an event into a room
The module system

- Introduced around 2017 to help with custom authentication backends
- Used a lot for spam checking (e.g. Mjölnir)
- Previously modules were split in several types with different ways to configure them with Synapse
- Rewrite started in 2021 to standardise the system into a single interface
- Most module types now migrated to the new system
What’s next?

- Migrating the **remaining interfaces** to the new system (#10469)
  - Media storage providers
  - SSO mapping providers
- Document the **ModuleApi** class in Synapse’s documentation (#10687)
- Deprecate and remove the **legacy interfaces**
- Deprecate and remove **duplicate** callbacks
- **Sanbox** modules so they can’t access Synapse’s internals
- …
Thank You! Any question?

babolivier@matrix.org
@babolivier:element.io