Robrix

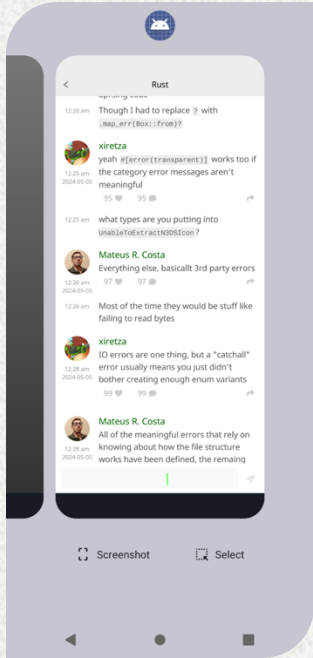A pure Rust multi-platform app for [matrix] chat and beyond

Kevin Boos

Project Robius Tech Lead

Principal Architect @ Futurewei
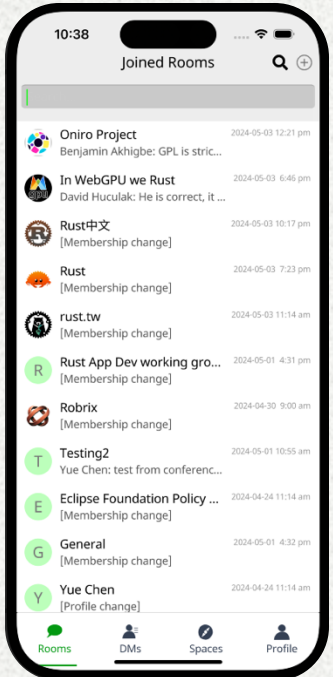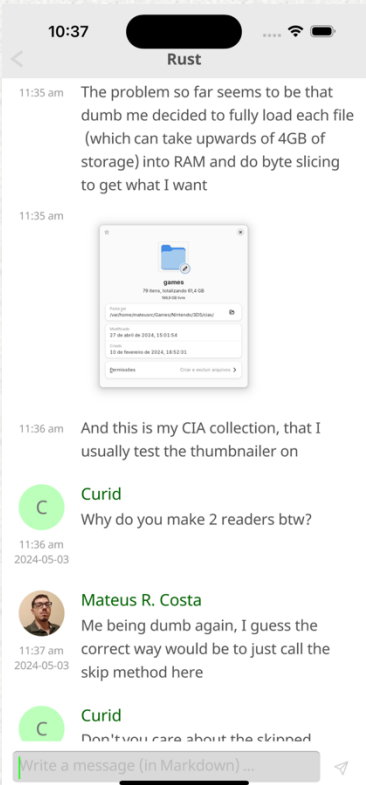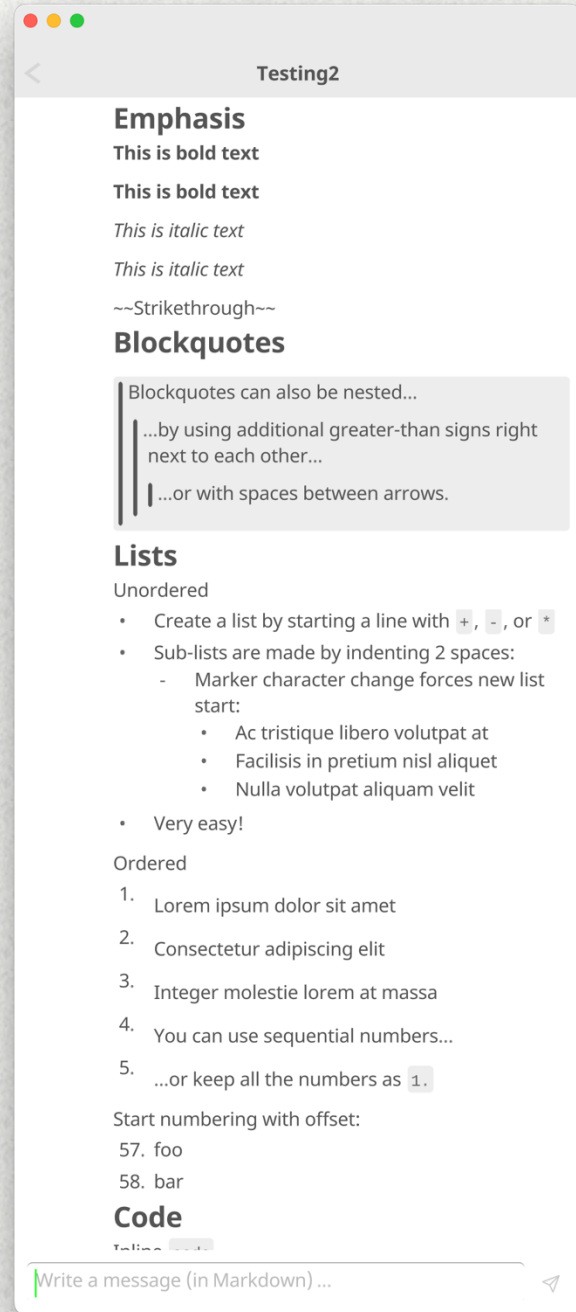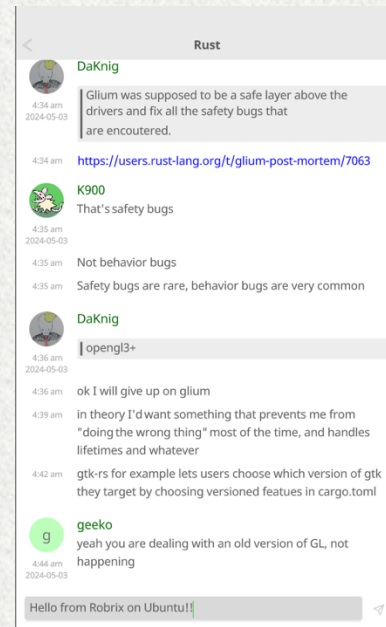
FOSDEM

Matrix devroom

February 2nd, 2025

ROBRIX

One code base, many platforms
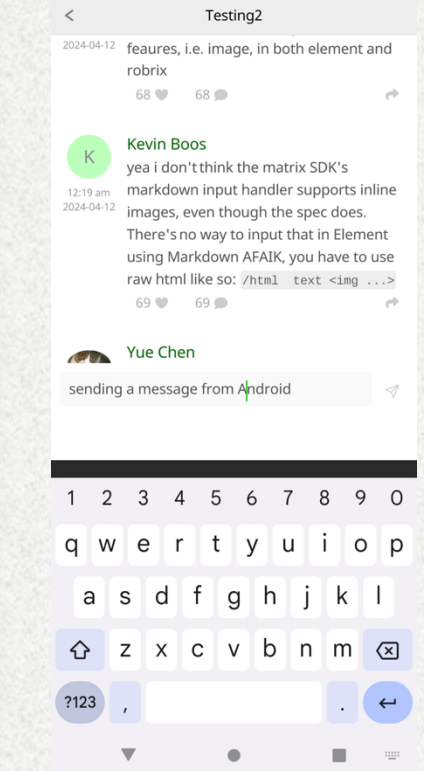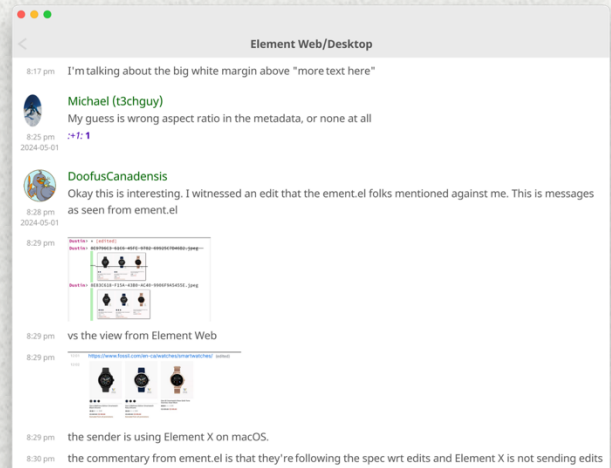
Background & Motivation

1. Create a multi-platform app dev experience fully in Rust

   Devs can:
   ➢ leverage the robust, safe, and performant Rust ecosystem
   ➢ avoid dealing with multiple languages/environments
   ➢ never write a single line of platform-specific code

2. Make Mobile a first-class concern in the Rust community
   • the Rust experience on Mobile has felt neglected

Project Robius goals at a glance

Background — Project Robius app dev framework

- Fully open-source, decentralized, and community-driven
  - Independent collaborators across US, Europe, Aus/NZ, S. America, China
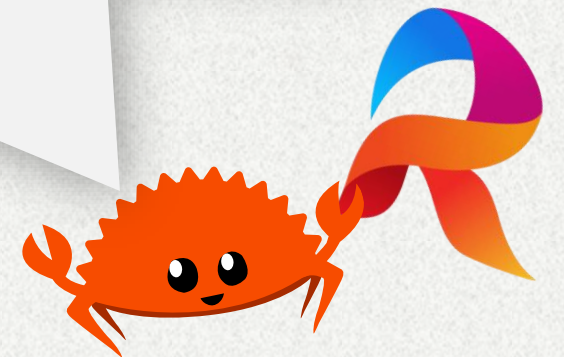- All components are also written entirely in Rust

But why Robius? ... there are already great Rust UI toolkits out there
- egui, Iced, Slint, Leptos, Tauri, Dioxus, Makepad, Xilem, Freya, Ribir, ...
→ There's **more to app dev** than just drawing a GUI!

- Traditionally, Rust apps have been "less Rust, more other code"
(a small Rust core surrounded by big platform-specific wrappers

Why *Robrix,* though?

1. Needed a key app to drive Robius development
   - "What kind of app is highly **demanding**?" (of UI & platform abstractions)
   - ➤ A modern chat client!
     - Requires network, geolocation, file/image access, audio/video capture & playback, system notifications, clipboard, rich text formatting & input, persistent device storage & cache, connectivity mgmt, biometric auth, secret key storage, and so much more!
   - Why Matrix?
     - Shares our values: open-source, decentralized, community-driven, and now focused on Rust

Why *Robrix,* though?

1. Needed a key app to drive Robius development
   - ➢ A modern chat client is quite demanding
     - Matrix aligns with our values & goals

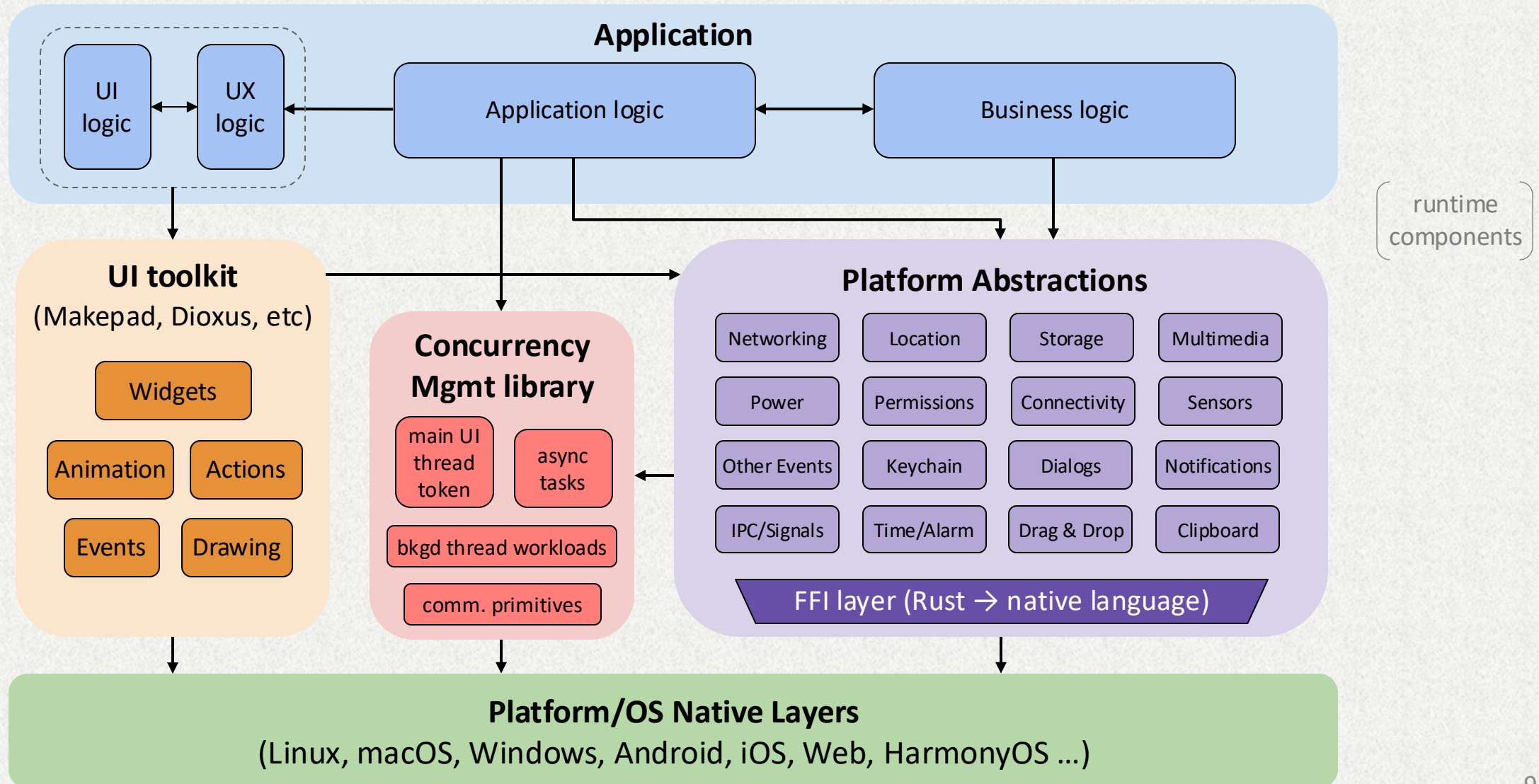2. But then... since we've already started, why stop there? Let's go further!
   - Can we make a unique Matrix client for power users / productivity?
   - Can we make Robrix into a central "hub" for federated services?
     - Focus on the needs of open-source community & developers
     - Combine Matrix, ActivityPub, local AI LLMs, source code tools, etc.

The better the app, the stronger the case for Rust app dev!
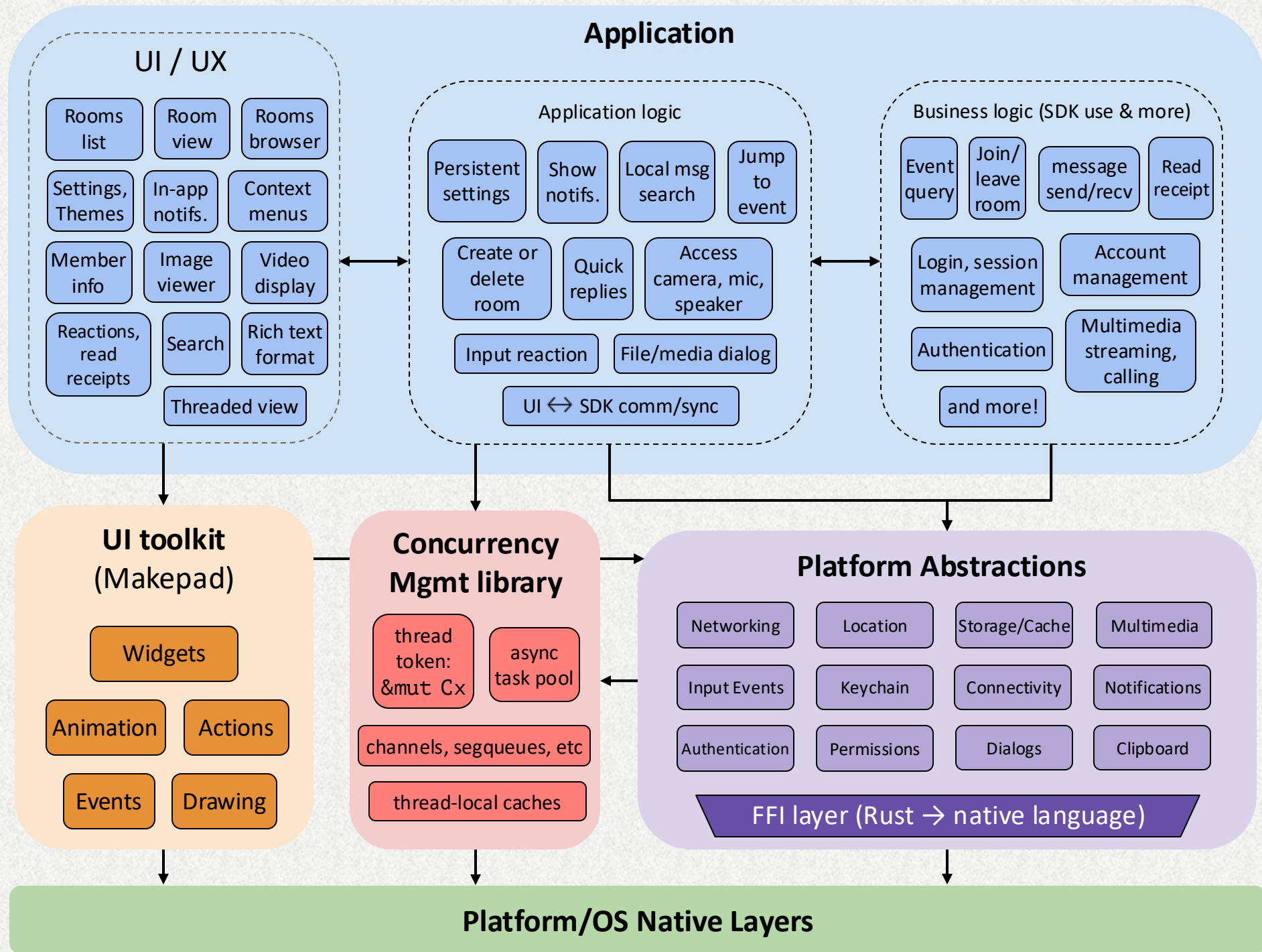
Goals & Roadmap

Main Robius goal: create reference design of full system stack

Application

UI logic ↔ UX logic → Application logic ↔ Business logic

runtime components

UI toolkit
(Makepad, Dioxus, etc)

- Widgets
- Animation
- Actions
- Events
- Drawing

Concurrency Mgmt library

- main UI thread token
- async tasks
- bkgd thread workloads
- comm. primitives

Platform Abstractions

Networking	Location	Storage	Multimedia
Power	Permissions	Connectivity	Sensors
Other Events	Keychain	Dialogs	Notifications
IPC/Signals	Time/Alarm	Drag & Drop	Clipboard

FFI layer (Rust → native language)

Platform/OS Native Layers
(Linux, macOS, Windows, Android, iOS, Web, HarmonyOS …)

Robrix App Architecture

(Matrix parts)

- Robrix's needs **guide** Robius development and implementation priority

- Writing/using Robrix forces us to **dogfood** Robius components
  - and Makepad, of course, which needed many new features to support Robrix

Application

UI / UX

- Rooms list
- Room view
- Rooms browser
- Settings, Themes
- In-app notifs.
- Context menus
- Member info
- Image viewer
- Video display
- Reactions, read receipts
- Search
- Rich text format
- Threaded view

Application logic

- Persistent settings
- Show notifs.
- Local msg search
- Jump to event
- Create or delete room
- Quick replies
- Access camera, mic, speaker
- Input reaction
- File/media dialog
- UI ↔ SDK comm/sync

Business logic (SDK use & more)

- Event query
- Join/ leave room
- message send/recv
- Read receipt
- Login, session management
- Account management
- Authentication
- Multimedia streaming, calling
- and more!

UI toolkit
(Makepad)

- Widgets
- Animation
- Actions
- Events
- Drawing

Concurrency Mgmt library

- thread token: &mut Cx
- async task pool
- channels, segqueues, etc
- thread-local caches

Platform Abstractions

- Networking
- Location
- Storage/Cache
- Multimedia
- Input Events
- Keychain
- Connectivity
- Notifications
- Authentication
- Permissions
- Dialogs
- Clipboard

FFI layer (Rust → native language)

Platform/OS Native Layers

Overall Roadmap

Stage 1: publish app with fundamental Matrix client features   **Q1**
- Basic messaging, login, app persistence, E2EE support
- Flesh out sufficient App Dev ecosystem and UI toolkit for major platforms

Stage 2: Matrix client for "power" users, plus AI integration   **Q3**
- Feature parity with existing major clients (incl. administrative features)
- Responsive UI design with productive (IDE-like) multi-room view
- Local LLM runtime for AI chat bots & conversation summaries
  - Key point: does not jeopardize E2EE or data sovereignty

Stage 3: central "hub" for federated & open-source services   **> Q4**
- Collect multiple services: chat, discussion forums, code + issues + PRs, announcements/microblogs, feed of notifications/activity/news
- Identity management via OpenWallet

Roadmap Stage 1

Stage 1: the basics ✅

- Realize Matrix client fundamentals
  - Login, rooms list (sliding sync),
    basic messaging, replies, reactions,
    rich text formatting, images,
    E2EE support, app persistence

- Publish pre-alpha versions of Robrix
  - Requires enhanced build tooling for
    releasing, signing, packaging

- Establish app framework for
  high performance and efficiency
  - Despite inherent async/concurrency
  - Key: *do "nothing"* on main UI thread

Stage 1: basic Matrix features

▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬

📅 Due by January 31, 2025   **93% complete**

This milestone is for the first alpha release of Robrix, i.e., Stage 1. This includes implementing fundamental Matrix chat features that make Robrix suitable for basic daily usage as a Matrix client. As such, the completion of this milestone should make Robrix eligible for inclusion in [Matrix's list of clients](#).

The rough goal for this milestone is to be (mostly) completed by Q1 of 2025. We intend to package and publish the Robrix app for macOS, Windows, Linux, Android, and (potentially) iOS.

Features being targeted for this milestone include (but are not limited to):

- ✅ ~~Displaying and sending reactions to messages~~
- Editing messages
- ✅ ~~Deleting messages~~
- ✅ ~~Sending and receiving standard text messages, including rich text formatting~~
- ✅ ~~Receiving image messages~~
- ✅ ~~Full pagination of historical timeline events~~
- ✅ ~~Viewing inline replies~~
- ✅ ~~Sending replies to specific messages~~
- ✅ ~~Correctly handling end-to-end encryption (E2EE)~~
- ✅ ~~Supporting basic user login and session management~~
- ✅ ~~Persistence of app data and Matrix SDK client data~~

☐ ⊙ **3 Open** ✓ 44 Closed

⠿ ☐ ⊙ **Display reactions as emojis**

#111 opened on Aug 19, 2024 by 👤 *kevinaboos*

Gradually checking items off our feature tracker

- Complete platform support from the beginning

Host OS	Target Platform	Builds?	Runs?
macOS	macOS	✅	✅
macOS	Android	✅	✅
macOS	iOS	✅	✅
Linux	Linux	✅	✅
Linux	Android	✅	✅
Windows	Windows	✅	✅
Windows	Android	✅	✅

Basic room views and fundamental actions

- ☑ View list of joined rooms
- ☑ View timeline of events in a single room
- ☑ Fetch and display room avatars
- ☑ Fetch user profiles (displayable names)
- ☑ Cache user profiles and avatars
- ☑ Cache fetched media on a per-room basis
- ☑ Fetch and display user profile avatars
- ☑ Backwards pagination to view a room's older history
- ☑ Dynamic backwards pagination based on scroll position/movement: #109
- ☑ Loading animation while waiting for pagination request: #109
- ☑ Stable vertical position of events during timeline update
- ☑ Display simple plaintext messages
- ☑ Display image messages (PNG, JPEG)
- ☑ HTML (rich text) formatting for message bodies
- ☑ Display reactions (annotations)
- ☑ Handle opening links on click
- ☑ Linkify plaintext hyperlinks
- ☑ Show reply previews above messages: #82
- ☑ Send standalone messages
- ☑ Interactive reaction button, send reactions: #115
- ☑ Show reply button, send reply: #83
- ☑ E2EE device verification, decrypt message content: #116
- ☐ Display multimedia (audio/video/gif) message events: #120
- ☐ Re-spawn timeline as focused on an old event after a full timeline clear: #103

Targeted platform abstractions / OS service APIs

- System notifications
- Biometrics & authentication
- File/image picker dialogs
- Context menus (custom, native)
- Menu bar (desktop only)
- Rich clipboard
- Drag & Drop
- Default URI/Intent handling
- Content sharing to foreign app
- Native font access
- Input Method Editors (IME)
- Native gesture recognition

- Connectivity management
  - WiFi, Bluetooth, NFC, nearby devices
- GPS/Location access
- Multimedia capture
  - Device discovery & configuration
  - Camera – snapshots, video, settings
  - Audio – input, MIDI, playback
- Sensors
- Native alarms, timers
- Keychain (secret storage)
- Power management/status
- App lifecycle state transitions
- OS-standard data directories

and so much more …

Current status (EOY 2024)

- System notifications
- Biometrics & authentication
- File/image picker dialogs
- Context menus (custom, native)
- Menu bar (desktop only)
- Rich clipboard
- Drag & Drop
- Default URI/Intent handling
- Content sharing to foreign app
- Native font access
- Input Method Editors (IME)
- Native gesture recognition

- Connectivity management
  - WiFi, Bluetooth, NFC, nearby devices
- GPS/Location access
- Multimedia capture
  - Device discovery & configuration
  - Camera – snapshots, video, settings
  - Audio – input, MIDI, playback
- Sensors
- Native alarms, timers
- Keychain (secret storage)
- Power management/status
- App lifecycle state transitions
- OS-standard data directories

and so much more …

demo time!

for future/offline viewers:
the demo included an overview of Robrix running on macOS and Android.

Check out our git repo and run it on your own machine here:
https://github.com/project-robius/robrix

Major Robrix-driven implementation challenges

- Realizing all the practical, unexciting UI features
- Exposing & abstracting platform APIs / OS services
- Handling complex multi-step builds (Rust + others)
  - Standardizing compilation of non-Rust system glue layers
  - Supporting platform-specific toolchains and linking conventions
  - Defining and executing pre- and post-compilation steps
- Packaging: bundling, codesigning, distributing apps
  - Required for system notifications, geolocation (and so much more)

all across disparate platforms

- Maintaining high performance in the face of concurrency and async
  - Key objective: **do "nothing"** on main UI thread
  - But, UI & other select platform functions **must** run on "main" thread
  - Certain APIs *require* async, others *forbid* async
- Contending with multiple *vastly* different UI toolkits

We manually manage mixed concurrency contexts

direct function call
sync –> **sync** channel
sync –> **async** channel
async –> **sync** channel

non-blocking requests to **async** funcs

responses from **async** funcs;
visual updates from backend server

UI main thread

App main func,
draw routines,
event handlers

Background thread

App logic
calling *sync*
platform APIs

**Background thread
in async runtime pool**

App logic
calling *async*
platform APIs

App logic
using Matrix

non-blocking requests
to bkgd (**sync**) functions

Nested sync requests,
which *may also (rarely)*
require async functions

non-blocking
media lookup
(fast path)

**Makepad
UI toolkit**

insert fetched
media (sync)

insert fetched
media (async)

media cache
(shared memory)

Robius Platform Abstractions

Matrix SDK

Platform/OS Native Layers

Major platforms already supported; others to come

- Desktop
  - macOS
  - Linux (Debian-based, X11)
  - Windows
- Mobile
  - Android
  - iOS
- Web/WASM
  - Waiting on Matrix SDK
- OpenHarmony OS
  - Waiting on availability and platform integration

OpenHarmony

Q2-Q3

WA

WEB ASSEMBLY

Roadmap Stages 2 & 3

Stage 2 — a Matrix client for power users

A. Feature parity with existing clients

B. Responsive UI with side-by-side panes & dockable tabs
- Visually beautiful and customizable across desktop, tablet, foldable, mobile
- Fast navigation with keyboard-driven interaction mode

C. Integration with *local* LLMs to leverage AI benefits
- Automated topic analysis, chat synopses of "what you missed", etc.
- Chatbot response channels to help newcomers in large open-source project communities
- *Local-only* LLM runtime preserves E2EE & data sovereignty

A. Reach feature parity with existing clients

Features that don't drive Project Robius development

Auxiliary features, login, registration, settings

- ☑ Persistence of app session to disk: #112
- ☑ Username/password login screen: #113
- ☑ SSO, other 3rd-party auth providers login screen: #114
- ☑ Side panel showing detailed user profile info (click on their Avatar)
- ☑ Ignore and unignore users (see known issues)
- ☑ Display read receipts besides messages: #162
- ☐ Collapsible/expandable view of contiguous "small" events: #118
- ☐ User settings screen
- ☐ Dedicated view of spaces
- ☐ Link previews beneath messages: #81
- ☑ Keyword filters for the list of all rooms: #123
- ☐ Search messages within a room: #122
- ☐ Room browser, search for public rooms
- ☐ Join, invite, and create rooms
- ☐ Room settings/info screen
- ☐ Room members pane

B. Multi-pane dockable tab UI

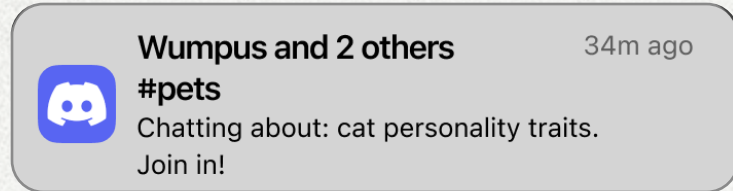(the original design proposal)

Robrix's actual "IDE" UI

- Auto-adapts to screen sizes
  - Same code works on Desktop, Mobile, Tablet...

Unlock productivity for work/social chats (just like IDEs)

C. AI capabilities via local-only LLM integration

Wumpus and 2 others   34m ago
#pets
Chatting about: cat personality traits.
Join in!

◄── Discord's chat topic summaries

Wouldn't it be great to enable this (and more) for Matrix chats?
  - But how can this work without giving an AI service access to all of your data?

- **Local LLMs** — benefit from AI _without_ jeopardizing E2EE, data sovereignty
  - **Auto-handle public FAQs** in an open-source project community
  - **"Action items":** AI identifies important messages & to-dos from a long backlog
  - **"What's happening":** AI-driven conversation summaries & topical analysis

- Longer-term: Matrix as a medium to share LLM agent recipes, chatbot demos

Robrix will leverage Moxin's existing featureset

Robrix will leverage Moxin's existing featureset

- **Moxin is another Robius app**
  - Also drove development of widgets, platform crates, packaging logic
  - Models run locally using WasmEdge, a cross-platform WASM LLM engine

- **Work has already begun to separate the discovery, download, execution, and configuration of LLMs from the app's UI frontend**

- **Coming soon: more modalities beyond LLM agents**
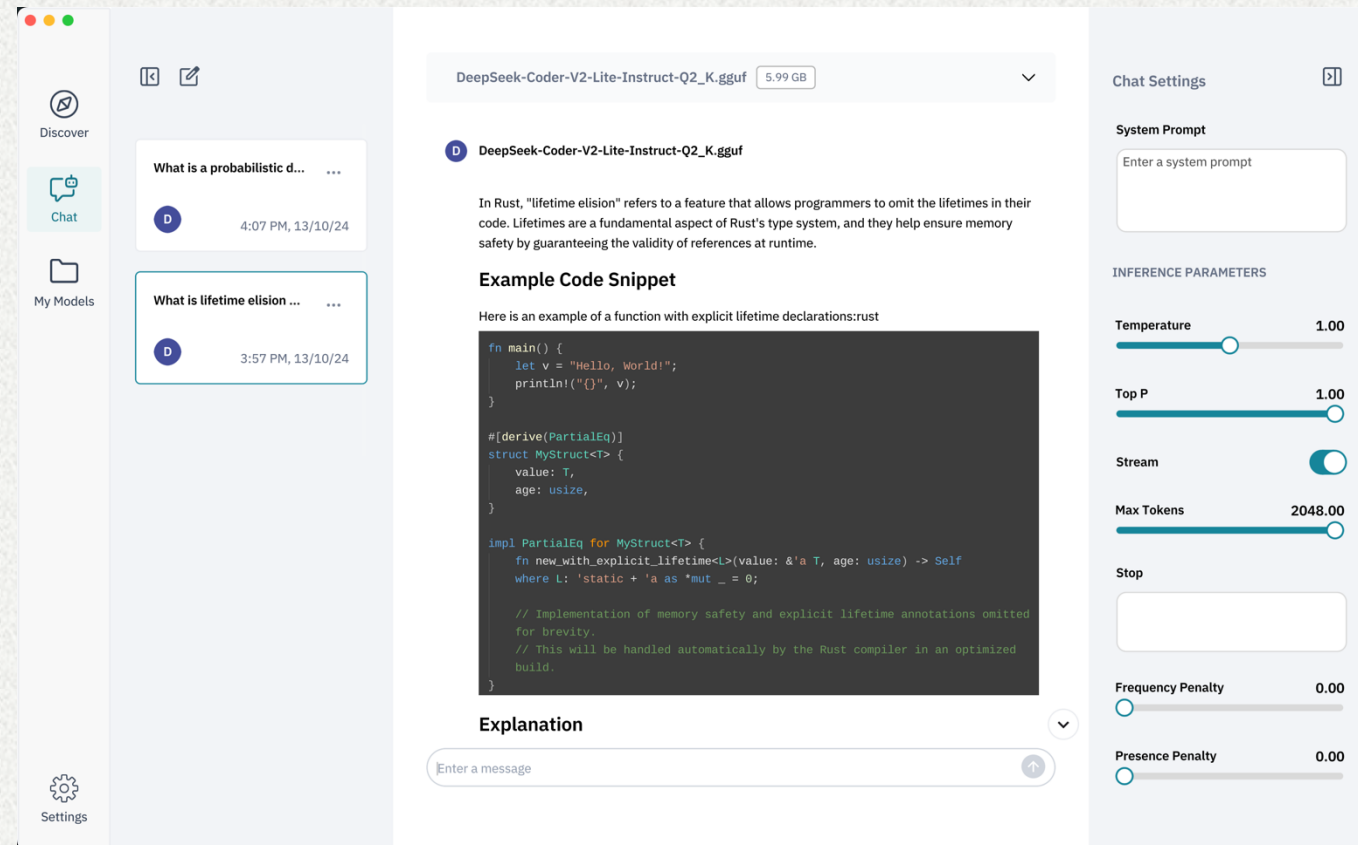
Robrix in-app AI LLMs

via a local Moxin installation

Robius: App Dev in Rust

Futurewei ⌄

People +

Alex Robins

Patryk

Yue

Kristie

David

Channels +

P   python   39
T   test-dev   12
N   neutron-dev   26

🔍 Search

P python   Alex Robins ✕

Brad Hugh 04/09/2024 11:06 AM
Hi! I'm locked out of my account for our testing env. Can you help sort this out for me?

Alex Robins 04/09/2024 12:22 PM
Yeah no problem. Can you send me your email and github username? I'll need some help from **@Emily** for tracking down the ID you use so feel free to ping her as well. Thanks!

Can you send me your email and github username?

Brad Hugh 01/09/2024 12:22 PM
Here you go:

Github.com

Github: @Brad_Hugh

Developer with various interests including python, rust, javascript, etc.

Message Alex Robins

AI Chat

Meta-Llama-3-8B-Instruct-Q2_K.gguf 2.96 GB

You

summarize the last two days of conversations from #matrix:matrix.org

Meta-Llama-3-8B-Instruct-Q2_K.gguf

Since May 1, 2024, users have been discussing CLI clients, how to export chat history, how to program Matrix bots, and more.

You

what else were they discussing, more recently?

Meta-Llama-3-8B-Instruct-Q2_K.gguf

I'd need to dig deeper into the conversation history... Okay, let me see...

Recently, users were discussing topics such as: better client-side implementations of the Matrix protocol, suggestions for improving the Matrix documentation, and high ping from the Matrix homeserver.
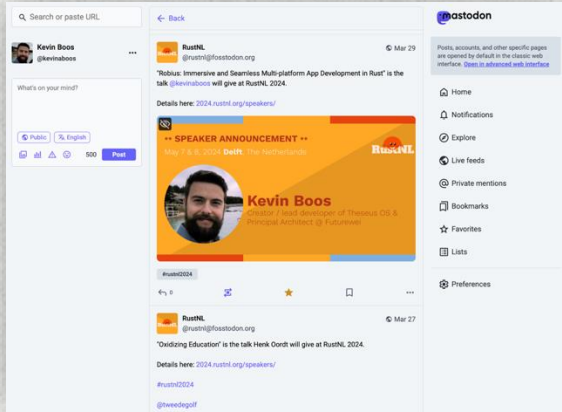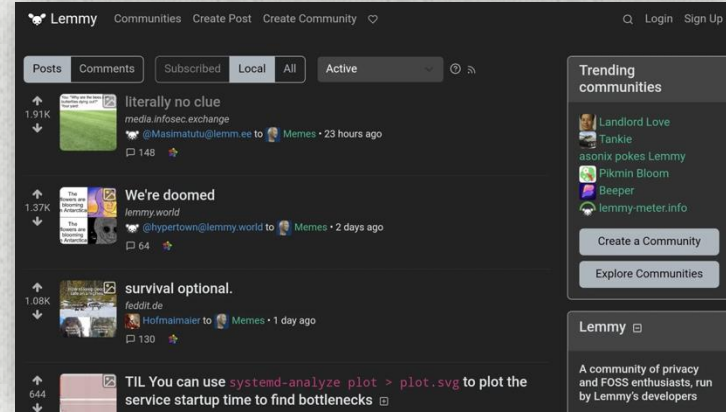
Enter a message

AI model is user's choice

31

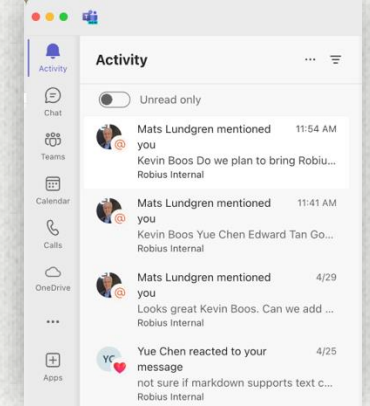# Stage 3 – a community hub (for open-source)
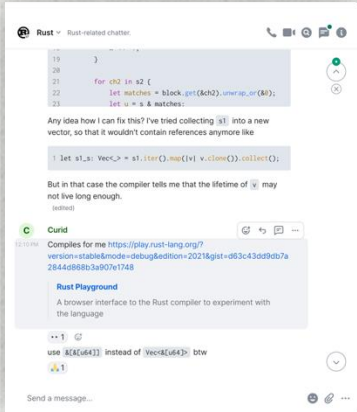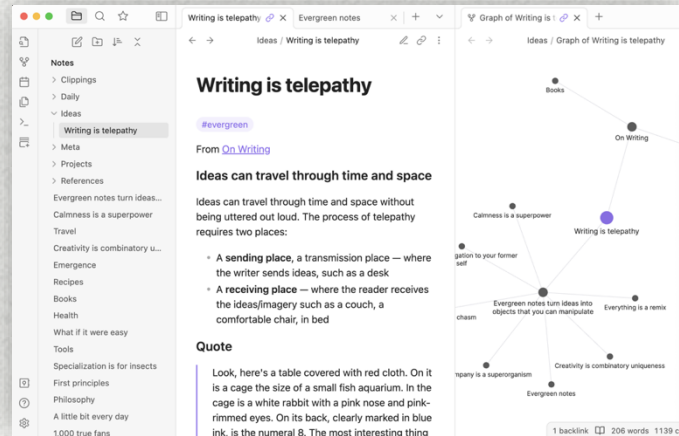
Chat (Matrix)

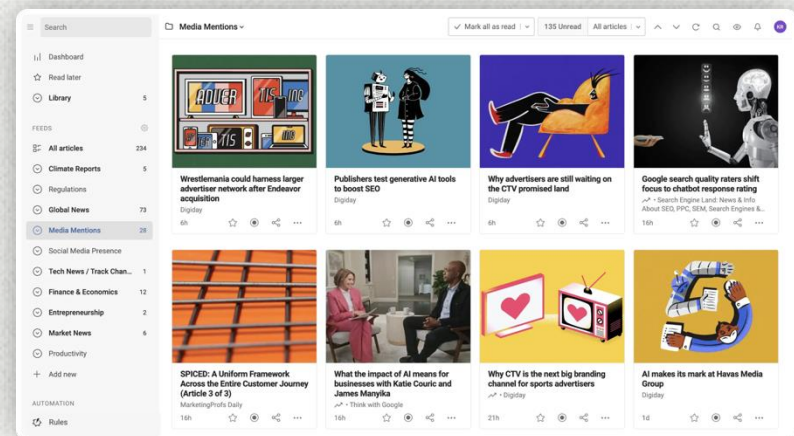**Microblogs** (Mastodon)

Community Forums (Lemmy)

Notifications, activity pane
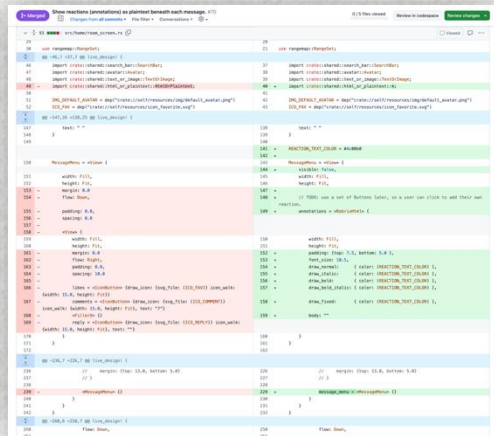
Code [re]view (git)

Notes

News feeds (RSS)

Microblogs
(Mastodon)

Community Forums
(Lemmy)

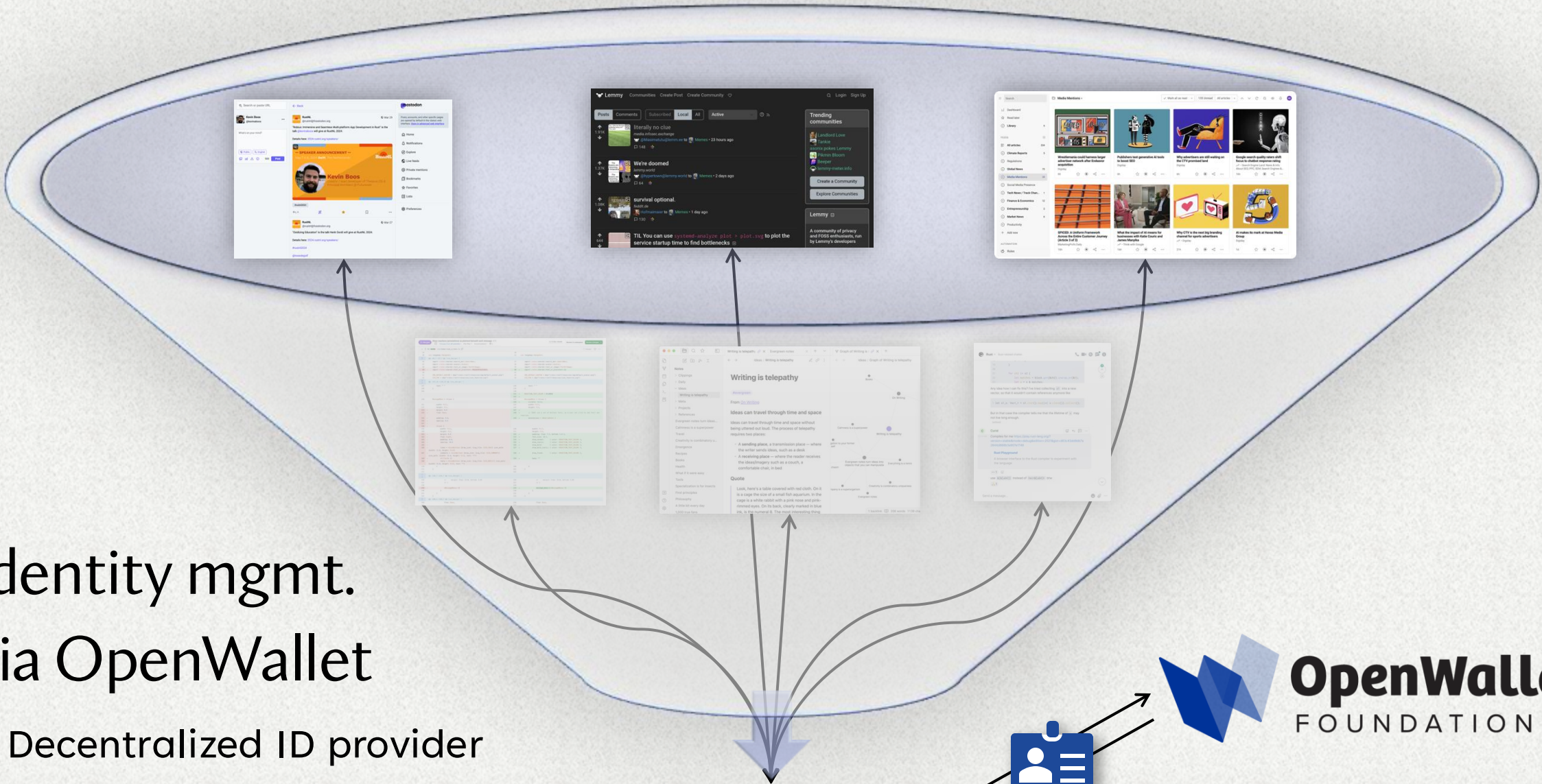**News feeds** (RSS)

Code [re]view
(git)

Notes

Chat
(Matrix)

Collect multiple federated services in a single experience

ROBRIX

Potential for powerful combo features & actions

Identity mgmt. via OpenWallet

- Decentralized ID provider

- Integrate with client for extra out-of-band functions
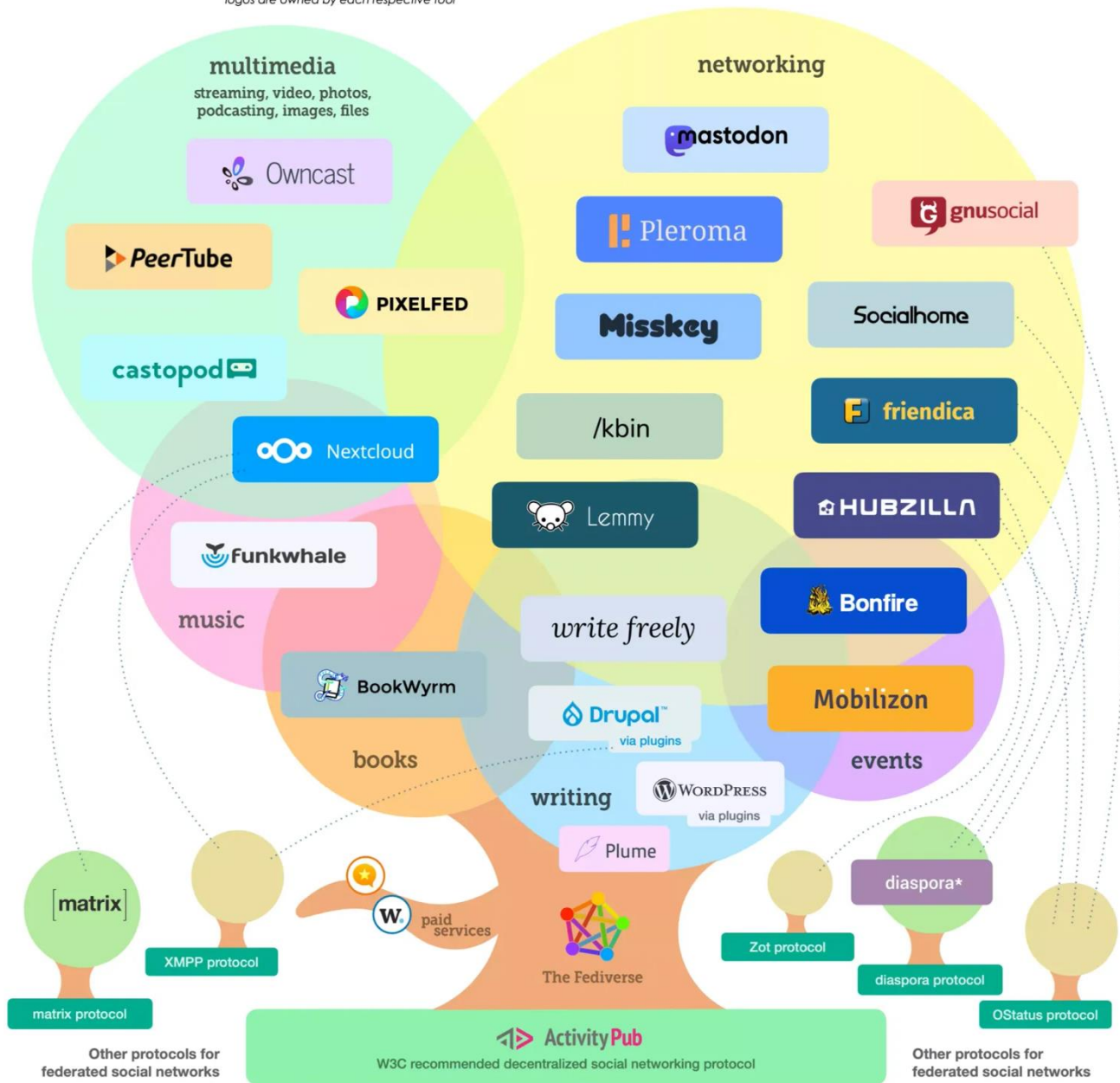
ROBRIX

OpenWallet FOUNDATION

The many branches of the Fediverse
axbom.com/fediverse • CC BY-SA Per Axbom
version 3.0 • January 17, 2023
logos are owned by each respective tool
Note: this is an overview and not a complete mapping of the Fediverse.

Many other service categories exist

- Most based on ActivityPub

Image sharing	Pixelfed
Music sharing	Funkwhale
Podcasting	Castopod
Video hosting	PeerTube
Full blogging	Wordpress, Plume, etc.

35

Wrapping up, but looking forward

- Robrix & Robius are just getting started!
  - Many challenges remain: UI design, platform support, build tooling
  - Continue driving collaboration with Rust UI/App ecosystem, lang/libs team

- Robrix aims to provide value *back* to the Matrix community
  - Not just an app — an **external** user of matrix-rust-sdk
    - An **independent, multi-platform** test case for the SDK that shows off its power!
    - Also a bit of a forcing function for correctness + documentation clarity 😬 😬
  - Strives to be a good answer for "how can I use this SDK function"?
    - Simple, "non-enterprise" coding style within a fully-fledged app environment

- For more, check out two recent blog posts: robius.rs/blog/
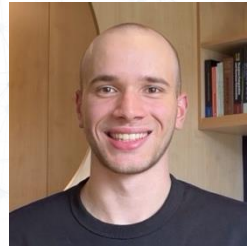 - A 2024 retrospective, and 2025 roadmaps for Robius and Robrix

Acknowledgments

Rik Arends

Edward Tan

Klim Tsoutsman

Alex Zhang

Sebastian
Michailidis

Julian
Montes de Oca
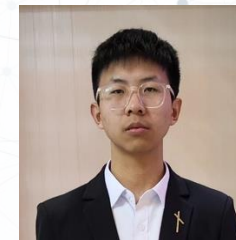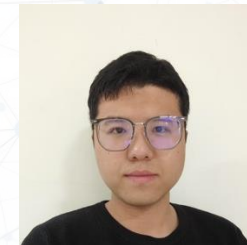
Poon Yong
Quan (Alan)

Tyrese Luo

Eddy Bruël

Jorge Bejar

Cassaundra
Smith

Yiming
(Aarav) Lu

Guo KeZhen
(Cork)

Thanks!

Interested? Please reach out:

@project-robius/robrix

#robrix@matrix.org

#robius@matrix.org

@kevinaboos