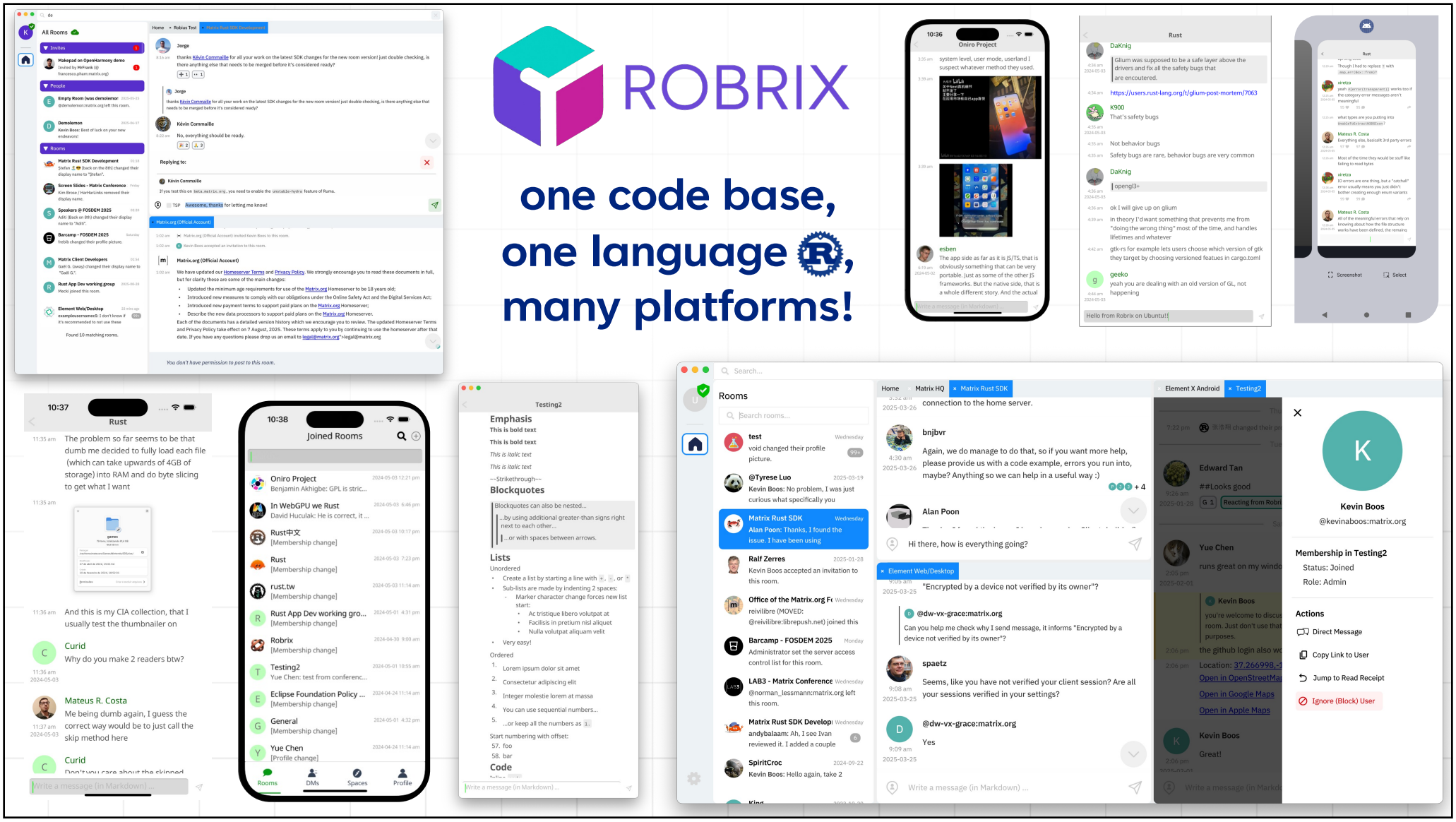
RUST GLOBAL × 

RustChinaConf 2025
Rust Global China

# ROBRIX

## a complex, multi-platform app in Rust for secure chat using [matrix]

**Kevin Boos, PhD**
Architect @ Futurewei Technologies


ROBRIX
one code base,
one language,
many platforms!

# You can now build complex apps in pure !

- Robrix is built on a **strong Rust-only foundation**:

Makepad – high-performance GPU-based UI toolkit with live reload, custom shader styling

+

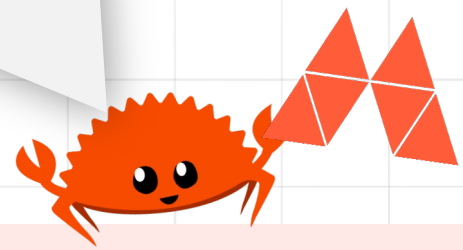
Robius – abstractions of platform features / OS services, build tooling, and more



# Background & Motivation

1. High-performance UI framework in pure Rust
   - Custom DSL for UI design: mixed immediate + retained mode
   - Shader-based styling with built-in hot reload
2. Lightweight & efficient; ~10-second compile time
3. Supports mobile + desktop + web + VR:  iOS & Android, macOS, Windows, Linux, WASM, Meta Quest
   - And recently, OpenHarmony too! (in progress)

# Makepad at a glance



1. Provide everything else needed for app dev ***beyond the UI***
   - Fill in missing support for many platform-provided features
2. Make Mobile a first-class concern in the Rust community
   - The Rust experience has felt neglected on Mobile
3. Bring frontend devs to Rust, and app dev to Rust users

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

# Project Robius at a glance

# Makepad + Robius = better Rust app dev


- Makepad recently released v1.0 (May 2025)
  - 5+ years of active development, still improving every day
  - However, there's more to app dev than just drawing a UI
  - This need fostered a joint collaboration with Robius, ongoing since 2023
- Both are fully open-source, decentralized, and community-driven
  - Independent collaborators across US, Europe, South America, China
- All components written entirely in Rust, *for* Rust devs
  - Direct, lightweight use of native platform APIs,
    no heavy deps like underlying web engines or bridges
  - Subverts the older app style of "less Rust, more other code"
    (a small Rust core surrounded by big platform-specific wrappers)

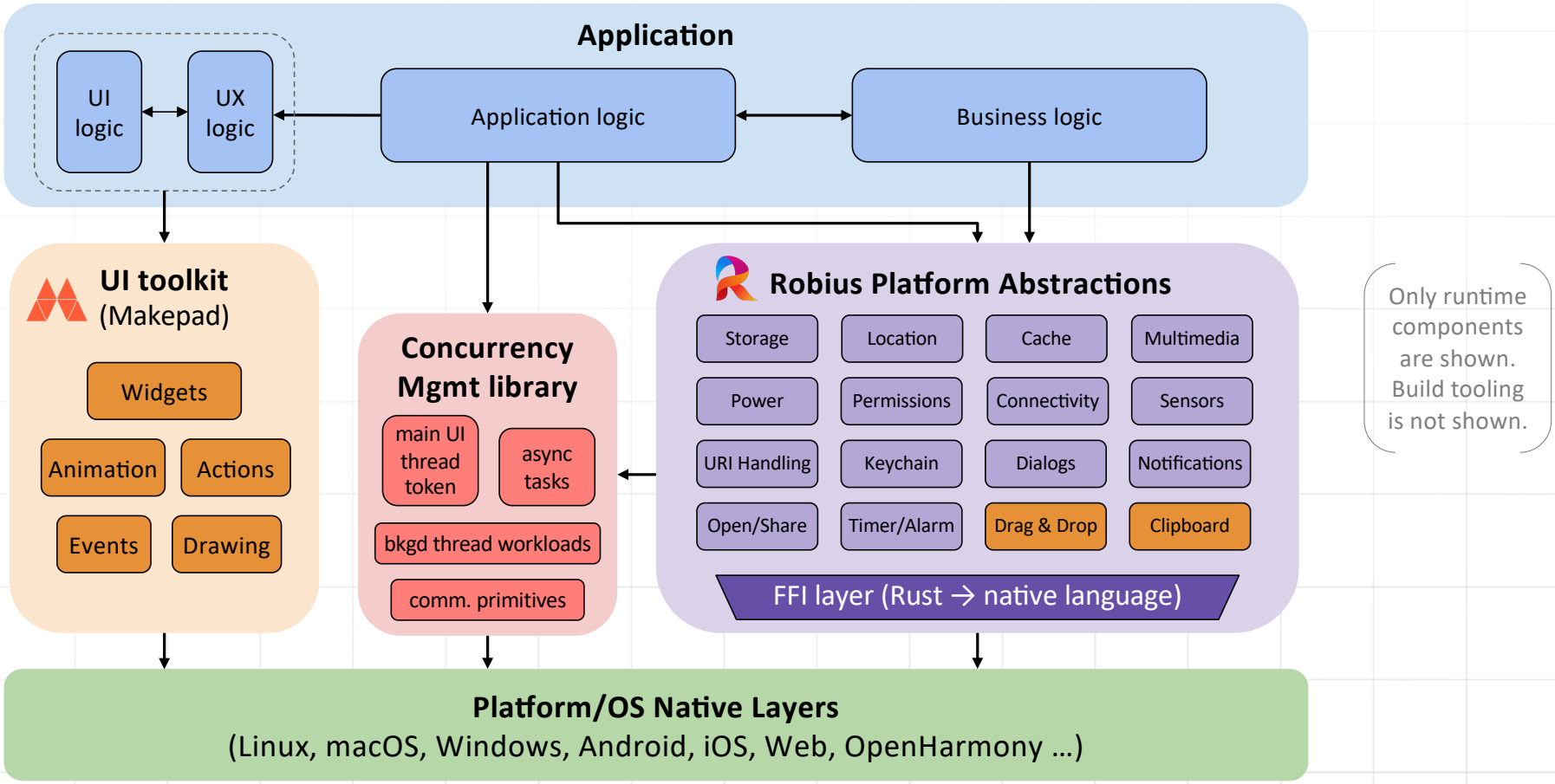
# What is [matrix] and why did we choose it?

- *"An open network for secure, decentralized communication"*
  - Similar to 钉钉/DingTalk, Microsoft Teams, Discord, Slack, **but open & federated**
- We needed a "killer app" to drive Robius development
  - "What kind of app is highly **demanding**?" (of UI & platform abstractions)
  - ➢ A modern chat client!
    - Requires network, geolocation, storage access for files/images, multimedia capture & playback, system notifications, clipboard, connectivity mgmt., rich text formatting & input, persistent device storage & cache, biometric authentication, secret key storage, and so much more!
  - Why Matrix? It's also open-source, decentralized, and Rust-oriented ✅



# Structure of a Robius app

# How do Makepad + Robius fit together?

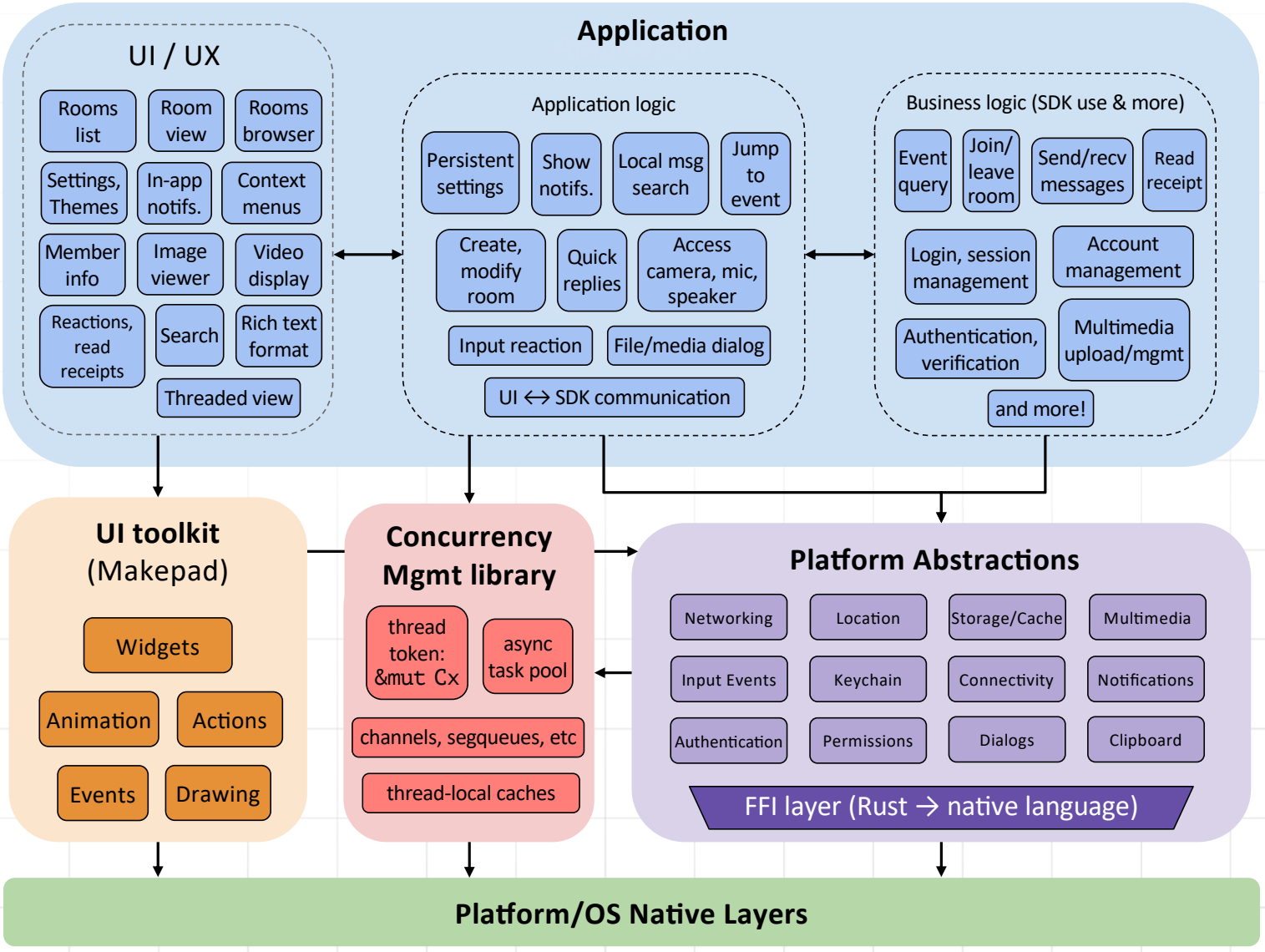



Only runtime components are shown. Build tooling is not shown.

# Architecture of the Robrix App

a Matrix chat client

- Robrix's needs **guide** Robius development and implementation priority
- Writing & using Robrix forces us to **dogfood** Robius components
  - and Makepad, of course
  - Added many new features to Makepad over the past ~2 years to support Robrix



# demo time!

for future/offline viewers:

the demo included an overview of Robrix's features
running on macOS & other platforms.

Check out our various apps to run them yourself:
https://github.com/project-robius/robrix
https://github.com/moxin-org/moly
https://github.com/makepad/makepad

# Code Examples & Implementation Details

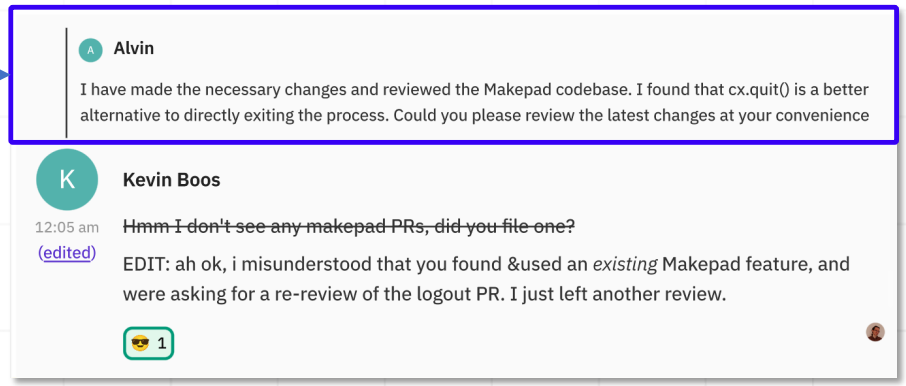
# A glimpse into Robrix's code

- What does a Makepad app look like?
  - Defining the UI via live DSL
  - Connecting the UI to Rust logic
- How do we leverage Rust for *safe*, high performance?
- Easily use Robius crates for deep platform integration



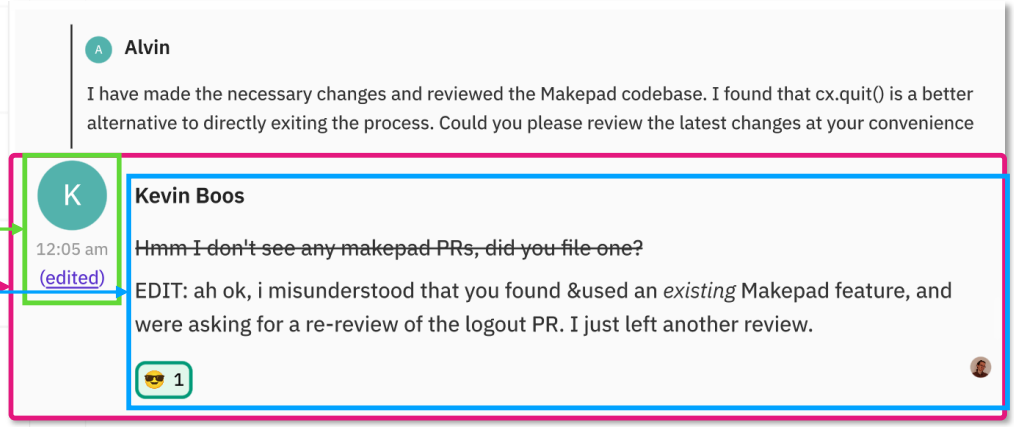
# Example: The UI code for a Message, using Makepad's *live DSL*

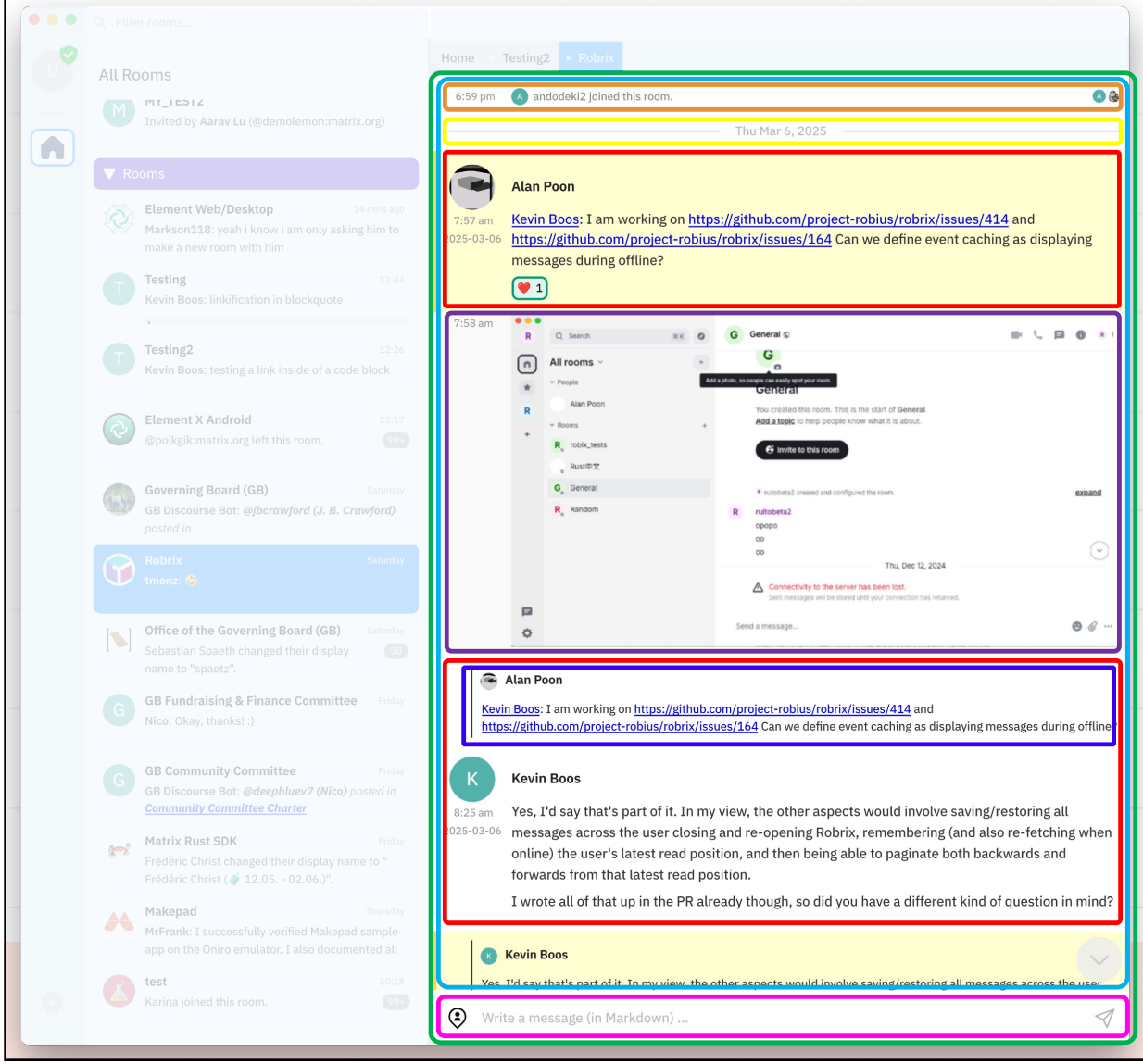
```
live_design! {
    Message = {{Message}} {
        width: Fill, height: Fit, flow: Down,

        show_bg: true,
        draw_bg: { ... }

        animator: {
            on_highlight = { ... }
            on_hover = { ... }
        }

        // Preview of an earlier message that this message replied to.
        replied_to_message = <RepliedToMessage> {
            visible: false
            flow: Right
            margin: { bottom: 3, top: 10 }

            replied_to_message_content = { ... }
        }

        body = <View> {
            width: Fill, height: Fit
            padding: {top: 0, bottom: 10, left: 10, right: 10},
            flow: Right,

            profile = <View> {
                align: {x: 0.5, y: 0.0}
                width: 65.0,  height: Fit,
                margin: {top: 5, right: 10}
                flow: Down,

                avatar = <Avatar> { width: 48, height: 48 }
                timestamp = <Timestamp> { }
                edited_indicator = <EditedIndicator> { }
            }
```



# The UI code for a Message, using Makepad's *live DSL*

```
body = <View> {
    width: Fill, height: Fit
    padding: {top: 0, bottom: 10, left: 10, right: 10},
    flow: Right,

    profile = <View> {
        align: {x: 0.5, y: 0.0}
        width: 65.0,  height: Fit,
        margin: {top: 5, right: 10}
        flow: Down,

        avatar = <Avatar> { width: 48, height: 48 }
        timestamp = <Timestamp> { }
        edited_indicator = <EditedIndicator> { }
    }

    content = <View> {
        width: Fill,  height: Fit
        flow: Down,

        username = <Label> {
            draw_text: {
                text_style: <USERNAME_TEXT_STYLE> {},
                color: (USERNAME_TEXT_COLOR)
                wrap: Ellipsis,
            }
            text: "<Username here>"
        }

        message = <Html> { }

        <View> {
            width: Fill, height: Fit, flow: Right,

            reaction_list = <ReactionList> { }
            avatar_row = <AvatarRow> {}
        }
```



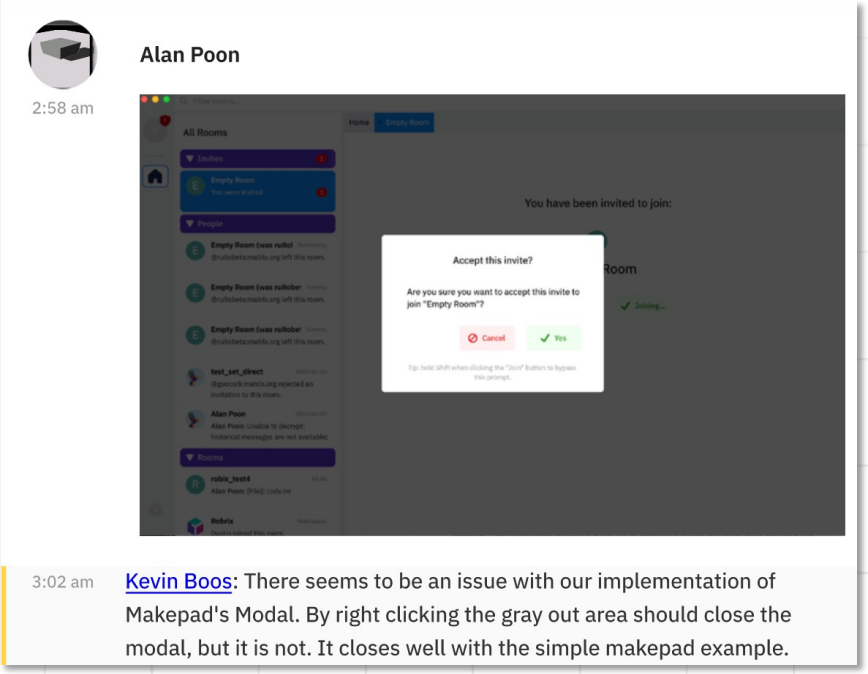



# Views, custom Widgets, and composing them

- RoomScreen
  - Timeline (PortalList)
    - SmallStateEvent
    - DayDivider
    - CondensedImageMessage
    - Message
      - ReplyPreview
      - Avatar, Username, Timestamp
      - HTML or plaintext content
      - Reactions
      - Seen-by users (AvatarSet)
  - RoomInputBar
    - Location icon button
    - MentionableTextInput
    - Send button

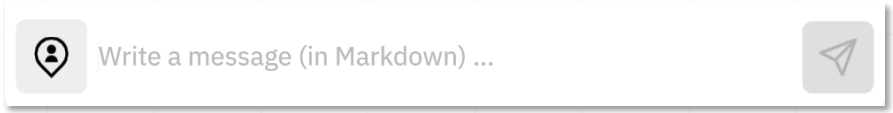
```
// A regular Message that shows an image, not text.
ImageMessage = <Message> {
    body = {
        // The `profile` view is inherited from `Message`,
        // so we don't need to change it here.

        content = {
            width: Fill, height: Fit
            padding: { left: 10.0 }
            message = <Image> { }

            v = <View> {
                width: Fill, height: Fit, flow: Right

                reaction_list = <ReactionList> { }
                avatar_row = <AvatarRow> {}
            }
        }
    }
}

// A condensed message comes right after another message
// from the same sender, so it doesn't show their avatar/name.
CondensedMessage = <Message> {
    body = {
        profile = <View> {
            align: {x: 0.5, y: 0.0}
            width: 65.0, height: Fit, flow: Down,

            timestamp = <Timestamp> { }
            edited_indicator = <EditedIndicator> { }
            tsp_sign_indicator = <TspSignIndicator> { }
        }

        // The `content` view is inherited from `Message`,
        // so we don't need to change it here.
    }
```

# DSL inheritance makes it easy to extend, customize, or re-style your widgets



# Connecting UI DSL code to Rust logic:

## ⟶ Event handling is simple via DSL *queries*




```
pub RoomInputBar = {{RoomInputBar}} {

    location_button = <RobrixIconButton> {
        spacing: 0,
        draw_icon: { svg_file: (ICON_LOCATION_PERSON) }
        icon_walk: { width: Fit, height: 23 }
    }

    message_input = <TextInput> {
        empty_text: "Write a message (in Markdown)..."
    }

    send_message_button = <RobrixIconButton> {
        // Only enabled when text is inputted
        enabled: false,
        draw_icon: {
            svg_file: (ICON_SEND),
            color: (COLOR_FG_DISABLED),
        }
        draw_bg: { color: (COLOR_BG_DISABLED) }
    }
}
```

```
impl Widget for RoomInputBar {
  fn handle_event(&mut self, cx: &mut Cx, event: &Event, ...) {
    if let Event::Action(actions) = event {
      // Handle the send message button click or Enter being pressed.
      let msg_input = self.text_input(id!(message_input));
      if self.button(id!(send_message_button)).clicked(actions)
          || msg_input.returned(actions).is_some()
      {
        let text = msg_input.text().trim();
        if !text.is_empty() {
          let message = create_message_with_mentions(&text);
          submit_async_request(MatrixRequest::SendMessage {
            room_id,
            message,
            replied_to: self.tl_state.as_mut()
              .and_then(|tl| tl.replying_to.take()),
          })

          self.clear_replying_to(cx);
          msg_input.set_text(cx, "");
          room_input_bar.enable_send_message_button(cx, false);
        }
      }
    }
  }
```

# Leverage Rust for max performance: *safely* and easily offload I/O & computation off of the main UI thread

```
submit_async_request(MatrixRequest::SendMessage {
    room_id,
    message,
    replied_to: self.tl_state.as_mut()
        .and_then(|tl| tl.replying_to.take()),
})
```

```
static REQUEST_SENDER:
    OnceLock<UnboundedSender<MatrixRequest>> = ...;

pub fn submit_async_request(req: MatrixRequest) {
    REQUEST_SENDER.get().send(req)
}
```

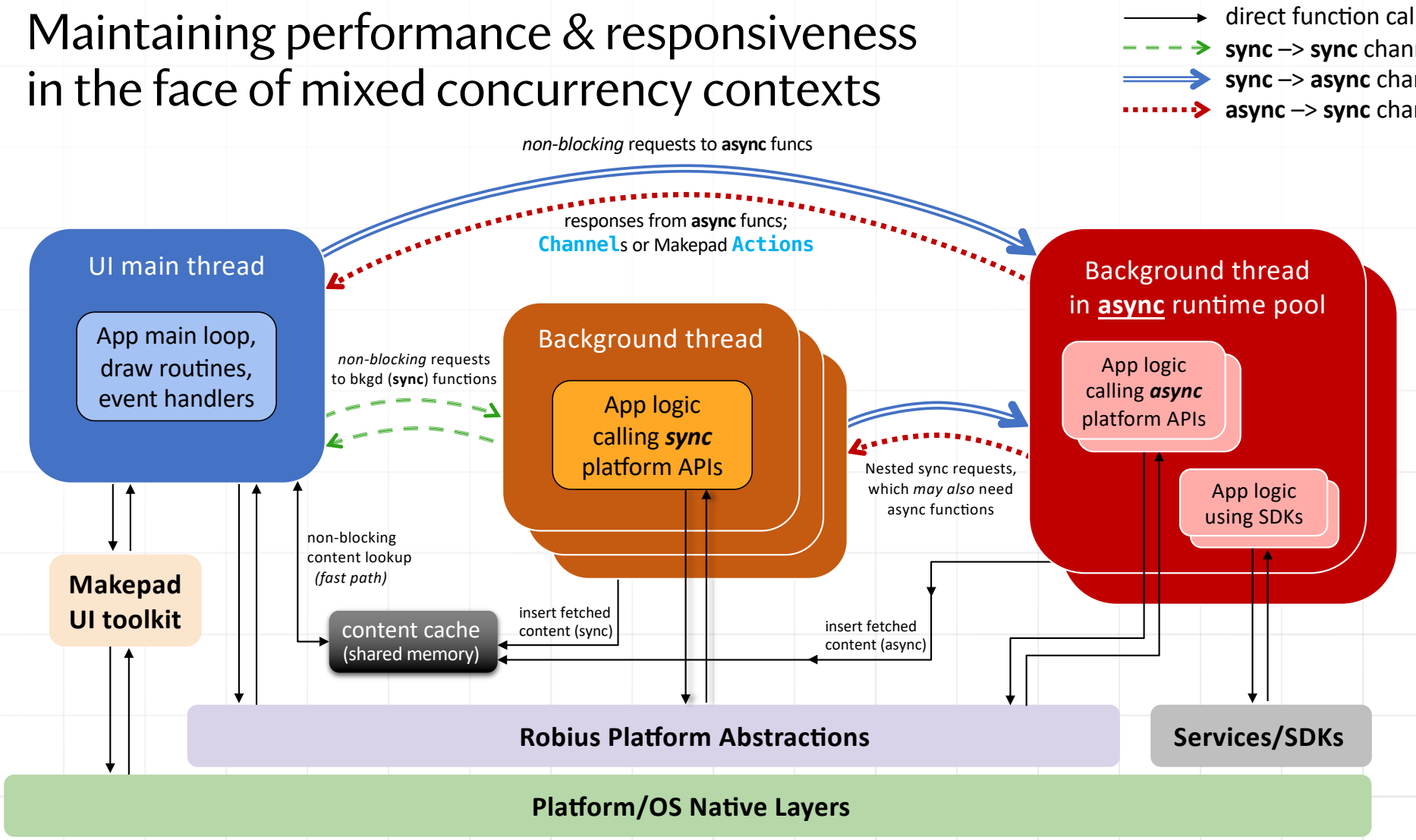
```
impl Widget for RoomInputBar {
  fn handle_event(&mut self, event: &Event, ...) {
    if let Event::Action(actions) = event {
      match actions.cast() {
        TimelineAction::SentMessage(res) => {
          // Update UI here with send result
        }
```

```
async fn async_matrix_worker(
    mut request_receiver: UnboundedReceiver<MatrixRequest>,
) -> Result<()> {
  while let Some(request) = request_receiver.recv().await {
    match request {
      MatrixRequest::SendMessage { room_id, msg, replied_to } => {
        let timeline = get_room_timeline(&room_id) else { ... };

        // Spawn a new async task that will send the actual message.
        Handle::current().spawn(async move {
          let result = if let Some(replied_to_info) = replied_to {
            timeline.send_reply(msg, replied_to_info).await
          } else {
            timeline.send(msg).await
          };
          Cx::post_action(TimelineUpdate::SentMessage(result));
        });
    }
```

**Main UI thread**

**Background async task**

# Maintaining performance & responsiveness in the face of mixed concurrency contexts



# Robius crates offer easy access to native platform APIs

## Easily query current location

```
struct LocationHandler;

impl robius_location::Handler for LocationHandler {
  fn handle_update(&self, location: Location<'_>) {
     match location.coordinates() {
        Ok(coordinates) => {
           Cx::post_action(LocationAction::Update {
             coordinates,
             time: location.time().ok(),
           });
        }
        Err(e) => Cx::post_action(LocationAction::Error(e)),
     }
}
...

let mgr = Manager::new(LocationHandler)?;
mgr.request_auth(Access::Foreground, Accuracy::Precise)?;
mgr.update_once(); // or start_updates(), stop_updates()
```

## Get & manage canonical directories for each platform, including mobile

```
let project_dirs = robius_directories::ProjectDirs::from(
  "org", "robius", "robrix",
);
let app_data_dir = project_dirs.data_dir();
let cache_dir    = project_dirs.cache_dir();


// Easily access storage & cache
save_persistent_app_data_to(&app_data_dir);

clear_app_cache(&cache_dir);
```

and many more features...

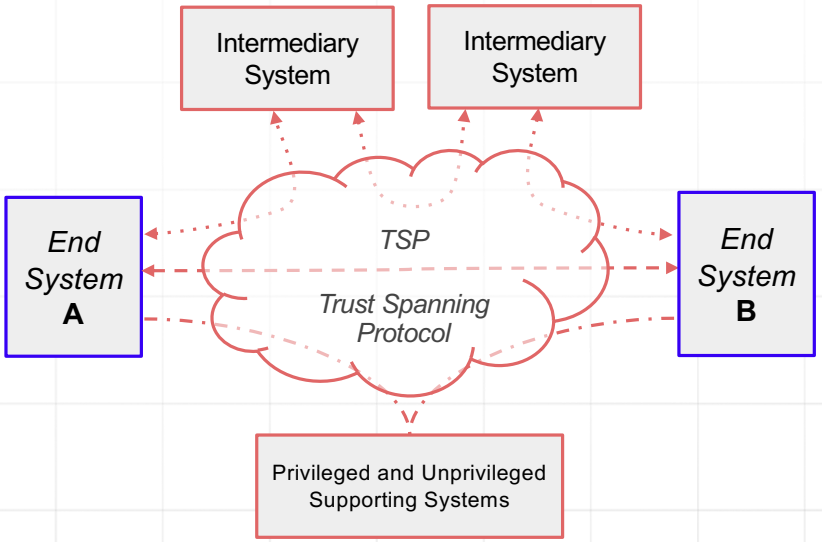


# Integrating TSP into Robrix

## the Trust Spanning Protocol

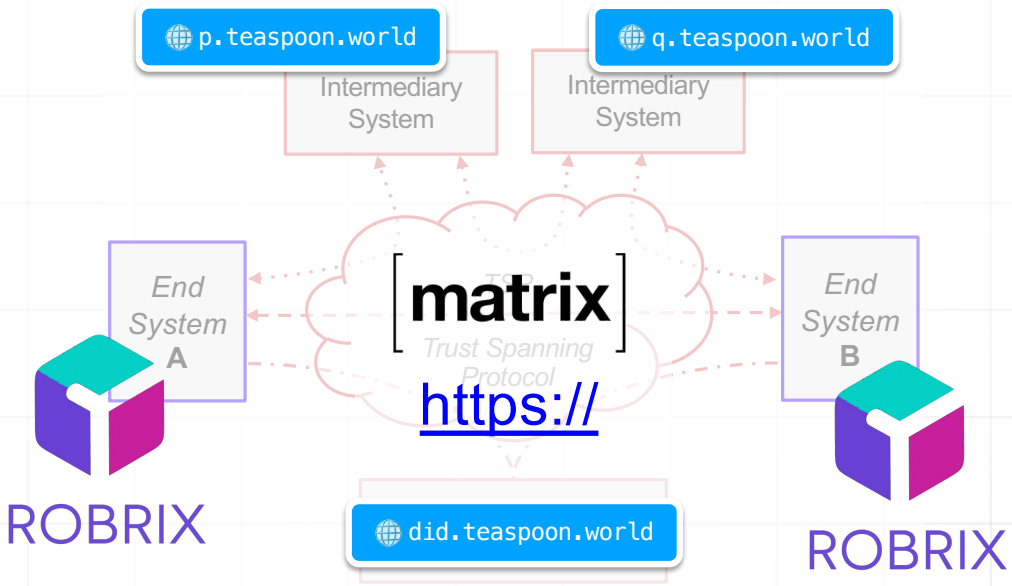
# Decentralized Identity Verification with TSP

- TSP facilitates effective trust relationships over an untrusted network (e.g., the internet)
  - Authenticity: sender is who they say they are; message contents are untampered
  - Confidentiality: Only the chosen receiver can read the message (if encrypted)
  - Metadata privacy, including protection through untrusted intermediaries
- No central authority required to act as the "trust provider"
- Persistent long-term identities are managed by wallets ***local*** to each endpoint system

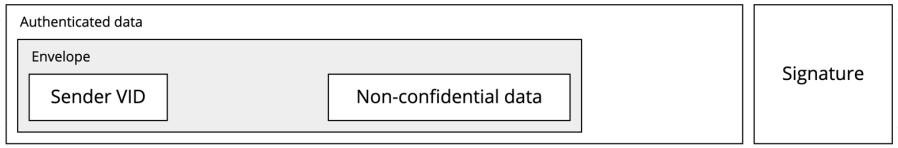



Diagrams courtesy of Wenjing Chu

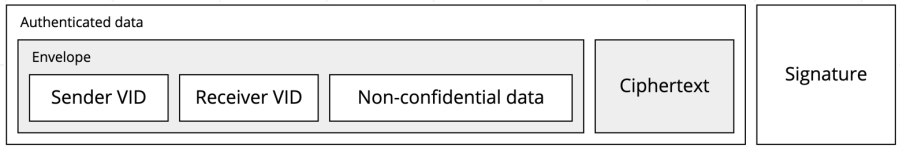
# TSP as used in Robrix




We use non-confidential signed messages in public rooms:




We can use confidential encrypted messages in direct rooms:




- TSP Spec: https://trustoverip.github.io/tswg-tsp-specification/
- TSP SDK: https://github.com/openwallet-foundation-labs/tsp

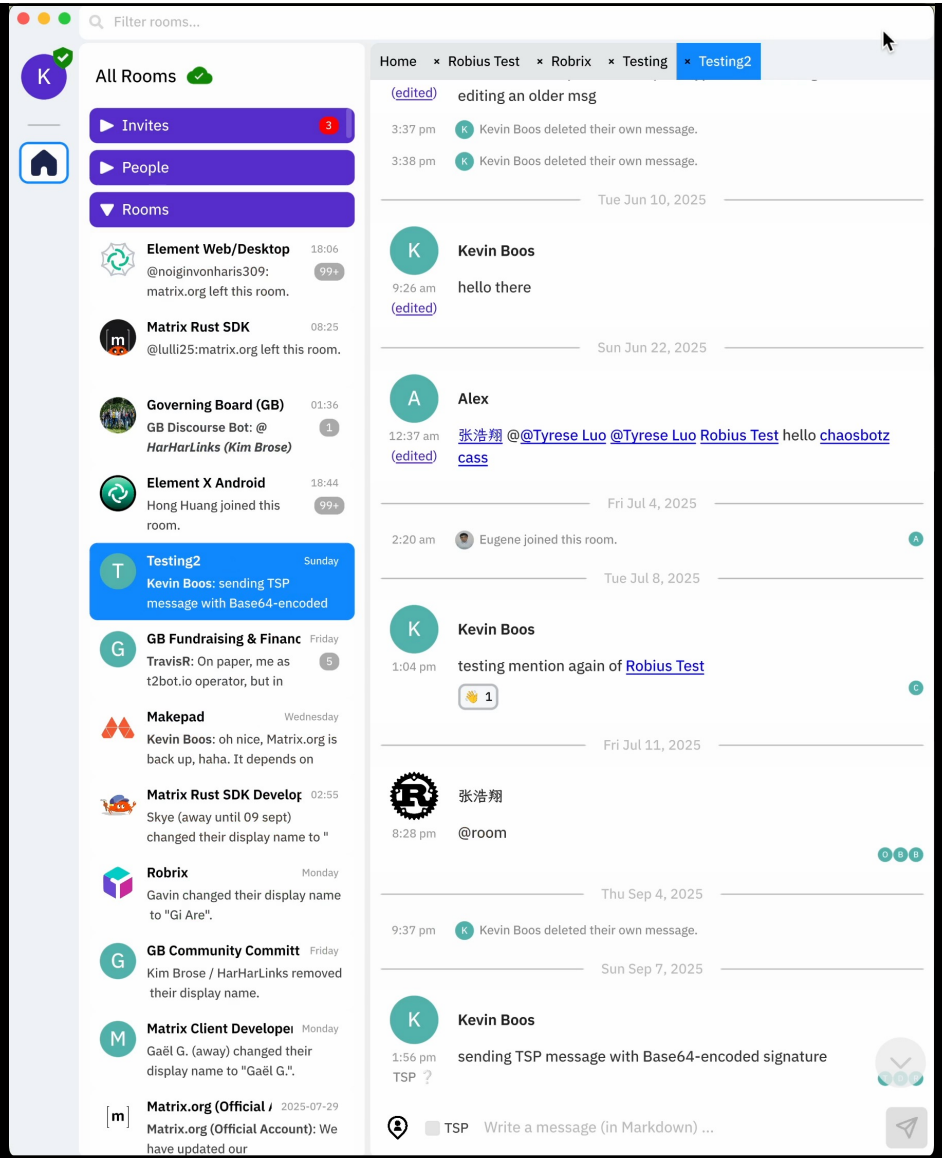
# TSP features implemented by Robrix

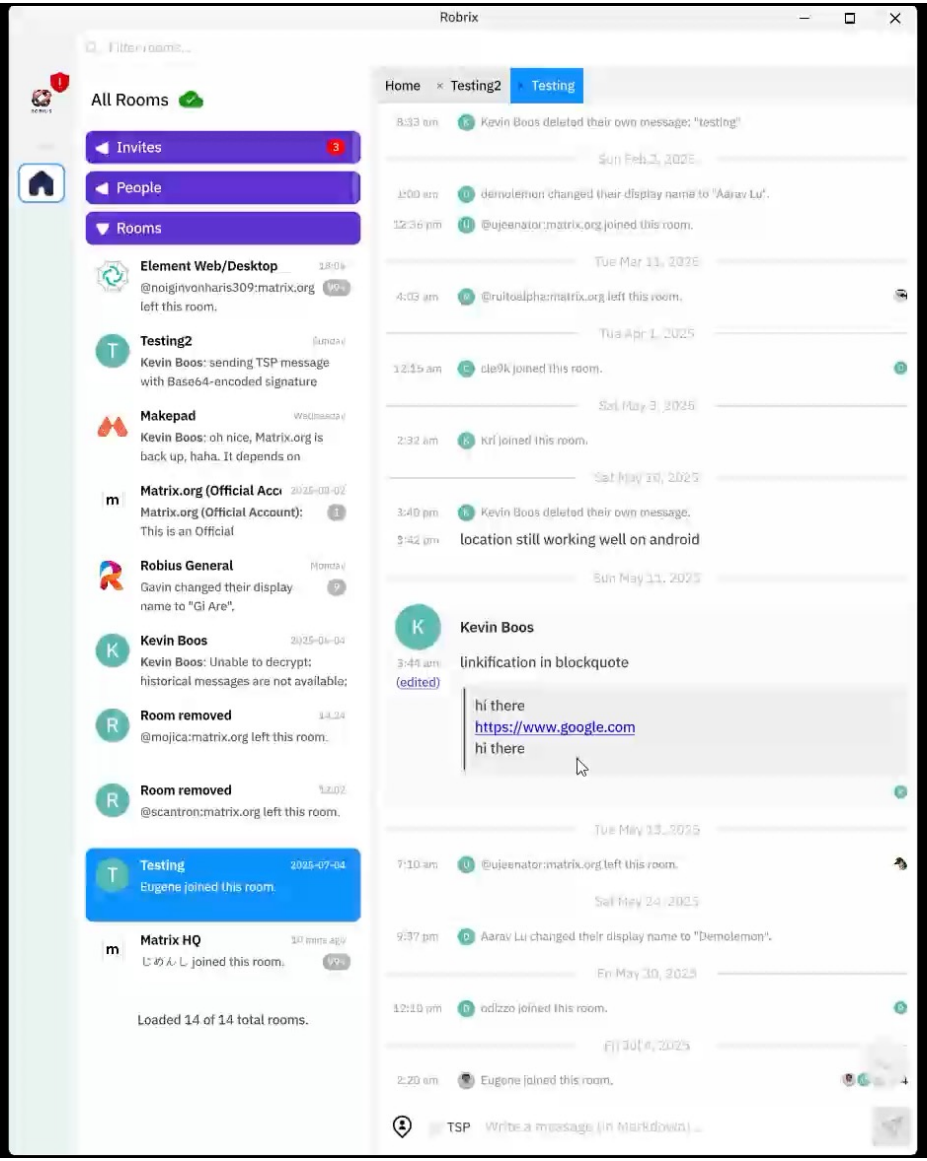
- Create & manage wallets
  - Persist/recover wallets to/from secure storage
- Create and (re)publish **D**ecentralized **ID**entities (DIDs)
- Verify other users' identities into your local wallet
- Associate a verified ID with a Matrix User ID
- Sign a Matrix message with your TSP ID when sending it out
- Send/receive direct TSP messages over TCP/HTTPS
  - As well as TSP data via Matrix as a transport



# Video demo of TSP atop Matrix

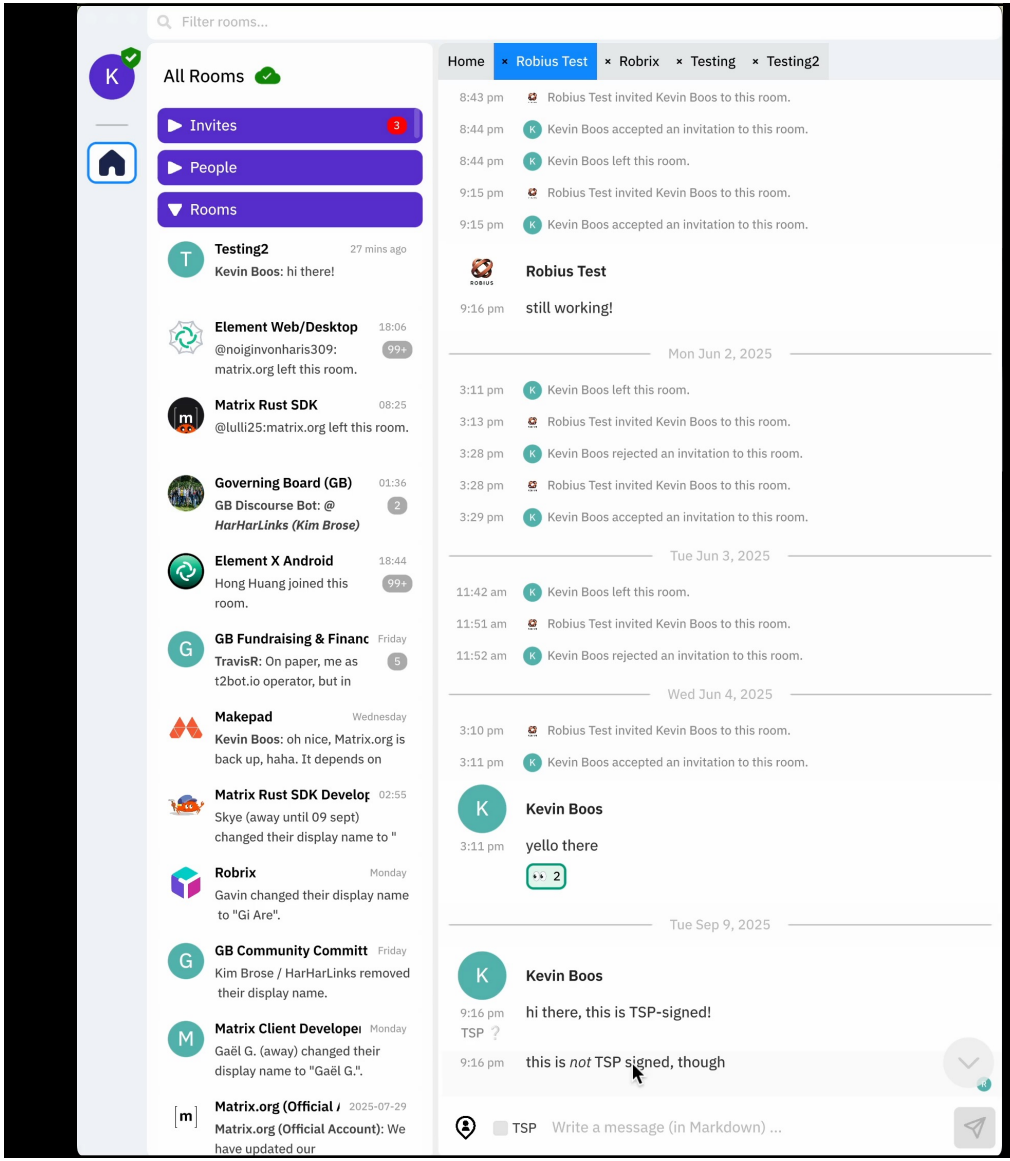


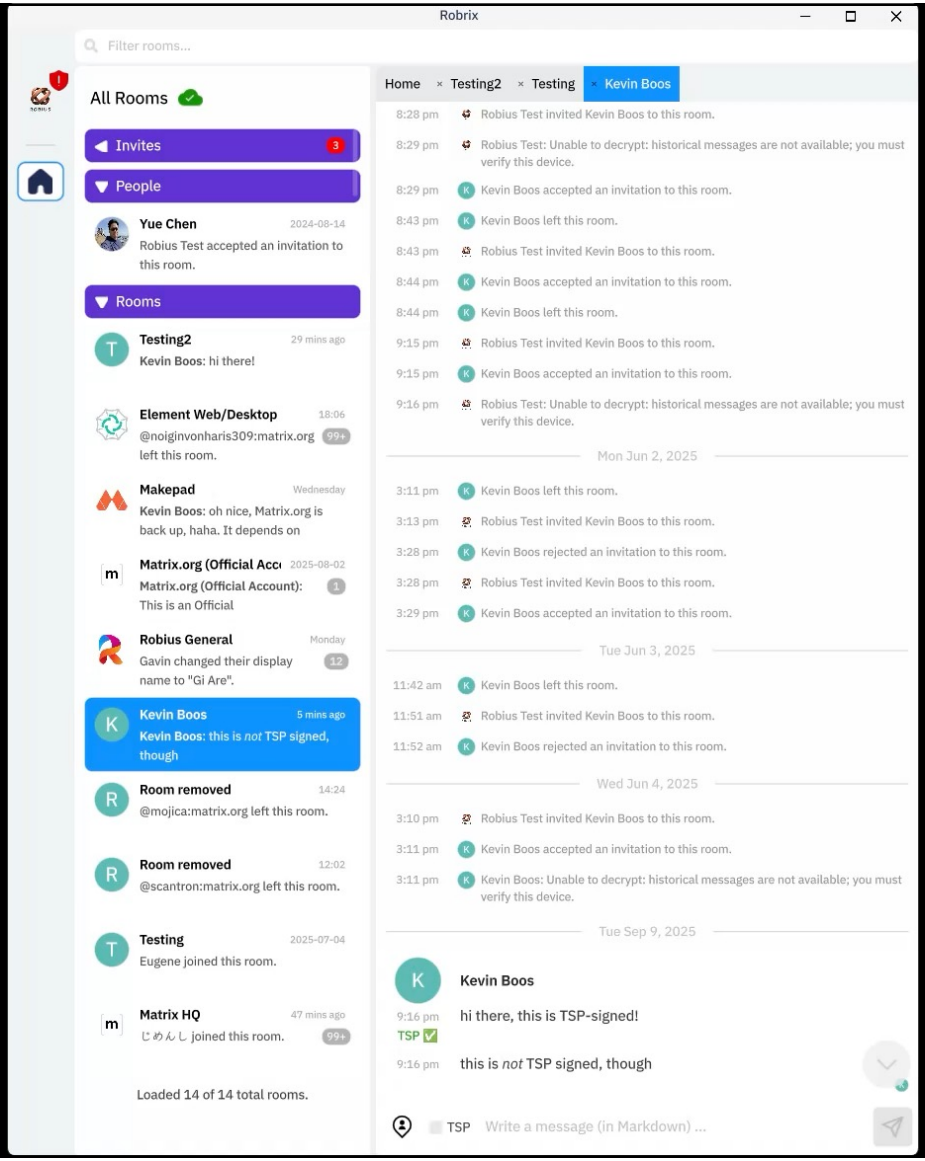

Filter rooms...
All Rooms
Invites 3
People
Rooms
Element Web/Desktop 18:06
@noiginvonharis309: matrix.org left this room. 99+
Matrix Rust SDK 08:25
@lulli25:matrix.org left this room.
Governing Board (GB) 01:36
GB Discourse Bot: @ HarHarLinks (Kim Brose) 1
Element X Android 18:44
Hong Huang joined this room. 99+
Testing2 Sunday
Kevin Boos: sending TSP message with Base64-encoded
GB Fundraising & Financ Friday
TravisR: On paper, me as t2bot.io operator, but in 5
Makepad Wednesday
Kevin Boos: oh nice, Matrix.org is back up, haha. It depends on
Matrix Rust SDK Develop 02:55
Skye (away until 09 sept) changed their display name to "
Robrix Monday
Gavin changed their display name to "Gi Are".
GB Community Committ Friday
Kim Brose / HarHarLinks removed their display name.
Matrix Client Developer Monday
Gaël G. (away) changed their display name to "Gaël G.".
Matrix.org (Official / 2025-07-29
Matrix.org (Official Account): We have updated our
Home
Robius Test
Robrix
Testing
Testing2
(edited) editing an older msg
3:37 pm Kevin Boos deleted their own message.
3:38 pm Kevin Boos deleted their own message.
Tue Jun 10, 2025
Kevin Boos
9:26 am (edited)
hello there
Sun Jun 22, 2025
Alex
12:37 am (edited)
张浩翔 @@Tyrese Luo @Tyrese Luo Robius Test hello chaosbotz cass
Fri Jul 4, 2025
2:20 am Eugene joined this room.
Tue Jul 8, 2025
Kevin Boos
1:04 pm
testing mention again of Robius Test
1
Fri Jul 11, 2025
张浩翔
8:28 pm
@room
Thu Sep 4, 2025
9:37 pm Kevin Boos deleted their own message.
Sun Sep 7, 2025
Kevin Boos
1:56 pm TSP
sending TSP message with Base64-encoded signature
TSP
Write a message (in Markdown) ...


Robrix
All Rooms
Invites 3
People
Rooms
Element Web/Desktop
@noiginvonharis309:matrix.org left this room.
Testing2
Kevin Boos: sending TSP message with Base64-encoded signature
Makepad
Kevin Boos: oh nice, Matrix.org is back up, haha. It depends on
Matrix.org (Official Acc
Matrix.org (Official Account): This is an Official
Robius General
Gavin changed their display name to "Gi Are".
Kevin Boos
Kevin Boos: Unable to decrypt: historical messages are not available;
Room removed
@mojica:matrix.org left this room.
Room removed
@scantron:matrix.org left this room.
Testing
Eugene joined this room.
Matrix HQ
joined this room.
Loaded 14 of 14 total rooms.
Home
Testing2
Testing
Kevin Boos deleted their own message; "testing"
demolemon changed their display name to "Aarav Lu".
@ujeenator:matrix.org joined this room.
@ruitoalpha:matrix.org left this room.
cle9k joined this room.
Kri joined this room.
Kevin Boos deleted their own message.
location still working well on android
Kevin Boos
linkification in blockquote
hi there
https://www.google.com
hi there
@ujeenator:matrix.org left this room.
Aarav Lu changed their display name to "Demolemon".
odizzo joined this room.
Eugene joined this room.
TSP

# ⚠ Warnings when TSP signatures *don't* match! ⚠




Filter rooms...
All Rooms
Invites 3
People
Rooms
Testing2 27 mins ago
Kevin Boos: hi there!
Element Web/Desktop 18:06
@noiginvonharis309: matrix.org left this room. 99+
Matrix Rust SDK 08:25
@lulli25:matrix.org left this room.
Governing Board (GB) 01:36
GB Discourse Bot: @ HarHarLinks (Kim Brose) 2
Element X Android 18:44
Hong Huang joined this room. 99+
GB Fundraising & Financ Friday
TravisR: On paper, me as t2bot.io operator, but in 5
Makepad Wednesday
Kevin Boos: oh nice, Matrix.org is back up, haha. It depends on
Matrix Rust SDK Develop 02:55
Skye (away until 09 sept) changed their display name to "
Robrix Monday
Gavin changed their display name to "Gi Are".
GB Community Committ Friday
Kim Brose / HarHarLinks removed their display name.
Matrix Client Developer Monday
Gaël G. (away) changed their display name to "Gaël G.".
Matrix.org (Official / 2025-07-29
Matrix.org (Official Account): We have updated our
Home
Robius Test
Robrix
Testing
Testing2
8:43 pm Robius Test invited Kevin Boos to this room.
8:44 pm Kevin Boos accepted an invitation to this room.
8:44 pm Kevin Boos left this room.
9:15 pm Robius Test invited Kevin Boos to this room.
9:15 pm Kevin Boos accepted an invitation to this room.
Robius Test
9:16 pm still working!
Mon Jun 2, 2025
3:11 pm Kevin Boos left this room.
3:13 pm Robius Test invited Kevin Boos to this room.
3:28 pm Kevin Boos rejected an invitation to this room.
3:28 pm Robius Test invited Kevin Boos to this room.
3:29 pm Kevin Boos accepted an invitation to this room.
Tue Jun 3, 2025
11:42 am Kevin Boos left this room.
11:51 am Robius Test invited Kevin Boos to this room.
11:52 am Kevin Boos rejected an invitation to this room.
Wed Jun 4, 2025
3:10 pm Robius Test invited Kevin Boos to this room.
3:11 pm Kevin Boos accepted an invitation to this room.
Kevin Boos
3:11 pm yello there
👀 2
Tue Sep 9, 2025
Kevin Boos
9:16 pm hi there, this is TSP-signed!
TSP ?
9:16 pm this is not TSP signed, though
TSP
Write a message (in Markdown) ...


Robrix
Filter rooms...
All Rooms
Invites 3
People
Yue Chen 2024-08-14
Robius Test accepted an invitation to this room.
Rooms
Testing2 29 mins ago
Kevin Boos: hi there!
Element Web/Desktop 18:06
@noiginvonharis309:matrix.org left this room. 99+
Makepad Wednesday
Kevin Boos: oh nice, Matrix.org is back up, haha. It depends on
Matrix.org (Official Acc 2025-08-02
Matrix.org (Official Account): This is an Official 1
Robius General Monday
Gavin changed their display name to "Gi Are". 12
Kevin Boos 5 mins ago
Kevin Boos: this is not TSP signed, though
Room removed 14:24
@mojica:matrix.org left this room.
Room removed 12:02
@scantron:matrix.org left this room.
Testing 2025-07-04
Eugene joined this room.
Matrix HQ 47 mins ago
じめんし joined this room. 99+
Loaded 14 of 14 total rooms.
Home
Testing2
Testing
Kevin Boos
8:28 pm Robius Test invited Kevin Boos to this room.
8:29 pm Robius Test: Unable to decrypt: historical messages are not available; you must verify this device.
8:29 pm Kevin Boos accepted an invitation to this room.
8:43 pm Kevin Boos left this room.
8:43 pm Robius Test invited Kevin Boos to this room.
8:44 pm Kevin Boos accepted an invitation to this room.
8:44 pm Kevin Boos left this room.
9:15 pm Robius Test invited Kevin Boos to this room.
9:15 pm Kevin Boos accepted an invitation to this room.
9:16 pm Robius Test: Unable to decrypt: historical messages are not available; you must verify this device.
Mon Jun 2, 2025
3:11 pm Kevin Boos left this room.
3:13 pm Robius Test invited Kevin Boos to this room.
3:28 pm Kevin Boos rejected an invitation to this room.
3:28 pm Robius Test invited Kevin Boos to this room.
3:29 pm Kevin Boos accepted an invitation to this room.
Tue Jun 3, 2025
11:42 am Kevin Boos left this room.
11:51 am Robius Test invited Kevin Boos to this room.
11:52 am Kevin Boos rejected an invitation to this room.
Wed Jun 4, 2025
3:10 pm Robius Test invited Kevin Boos to this room.
3:11 pm Kevin Boos accepted an invitation to this room.
3:11 pm Kevin Boos: Unable to decrypt: historical messages are not available; you must verify this device.
Tue Sep 9, 2025
Kevin Boos
9:16 pm hi there, this is TSP-signed!
TSP ✅
9:16 pm this is not TSP signed, though
TSP
Write a message (in Markdown) ...

# TSP signature is injected into Matrix message

- Compliant with Matrix Spec ✅
  - Doesn't affect other clients; can be freely ignored
- Currently includes the full message (Base64-encoded)
  - Will add TSP SDK support for extracting message payload out of signed byte array

Original event source


```
{
  "content": {
    "body": "sending TSP message with Base64-encoded signature",
    "m.mentions": {},
    "msgtype": "m.text",
    "org.robius.tsp_signature":
"+SABXAAAXAAB9VIDAAARAABkaWQ6d2ViOmRpZC50ZWFzcG9vbi53b3JsZDplbmRwb2ludDprZXZpbmFib29
zX21l5BAgAHsibXNndHlwZSI6Im0udGV4dCIsImJvZHkiOiJzZW5kaW5nIFRTUCBtZXNzYWdlIHdpdGggQmF
zZTY0LWVuY29kZWQgc2lnbmF0dXJlIiwibS5tZW50aW9ucyI6e3190BBmEa2Nk9ket8+Z9LIIl1nj+paEcgI
GyPsUJq/TETpRwMMepd8orKQoblhr7BOSU5gcN5EJ3rROyJmCwS9rxfgI"
  },
  "origin_server_ts": 1757278579091,
  "sender": "@kevinaboos:matrix.org",
  "type": "m.room.message",
  "unsigned": {
    "membership": "join",
    "age": 188275837
  },
  "event_id": "$CwJHtN3ZTmGh0DZ41_r57OaEUDPQBXBECDaGitcAtHY",
  "room_id": "!nCULugmcNjxWftdWKg:matrix.org"
}
```




# Use cases & benefits of TSP atop Matrix

- ✅ An additional layer of ID verification beyond Matrix
  - Autonomously *prove* an identity (“I am who I say I am”) and the message’s *authenticity*
- ✅ Show a smoke signal warning if someone’s signature changes
  - Prevents false identity scams like “pig butchering”/杀猪盘
- 🚧 Double encryption for direct (1-on-1) messages
  - Decryption requires locally-available state in secure TSP wallet;
    **attacker can’t read messages even if they compromise your Matrix account**
- Identities are ***portable***, not tied to or limited to Matrix
  - Can take your verified relationships with you to a new Matrix account,
    or port them over to a new service entirely (e.g., Mastodon, Bluesky)
- Relationships (creditable endorsements) can be *referred*:
  - “Your trusted contact Bob verifies that a new user Alice is *actually* Alice”



# Concluding Remarks

# Wrapping up, but looking forward

- You can now use Rust to build complex, real-world apps! 🚀
  - Makepad + Robius enable high performance UI/UX amidst very demanding use cases
  - Offers consistent, deep system integration across multiple platforms
- Many challenges still remain
  - Simplifying UI design, more platform features, better build tooling
  - Deeper TSP integration: referrals, nested relationships, double encryption
- For more, check out our blog: [robius.rs/blog/](robius.rs/blog/)
  - Looking forward to continued collaboration with other Rust UI/App project organizations, and the Rust lang/libs/tools teams



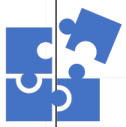
# Roadmap (in no particular order)

Expand Makepad's UI **widget collection** and **input event** support

- Ensure consistent coverage for app lifecycle events

Continue developing key **platform feature abstraction** crates

- Focus on mobile platform APIs, for truly *immersive* apps
- Backfill OpenHarmony support into existing Robius (and other) crates



Better, more automated **build tooling**

- Auto-detection + generation of required app permissions/entitlements

**Publish flagship apps** to app stores & package managers

- Under the stewardship of the GOSIM organization

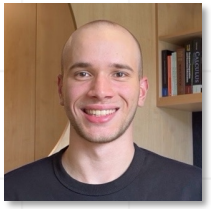
**More docs** and **guided tutorials** for Makepad + Robius app dev

# Acknowledgments

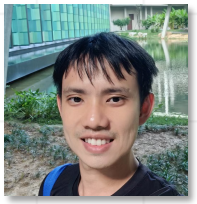
RustChinaConf 2025
Rust Global China


Rik Arends


Edward Tan


Klim Tsoutsman


Alex Zhang


Poon Yong Quan (Alan)


Sebastian Michailidis


Julian Montes de Oca


Alvin

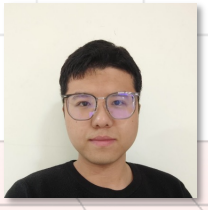

Tyrese Luo


Cassaundra Smith


Eddy Bruël


Jorge Bejar


Yiming (Aarav) Lu


Guo KeZhen (Cork)

RUST GLOBAL × 杭州

RustChinaConf 2025
Rust Global China

# Thanks!

Interested? Please reach out:

- @project-robius
- #robius@matrix.org
- @kevinaboos
- https://makepad.nl
- https://robius.rs