

# Standalone Matrix-based apps using the Widget API

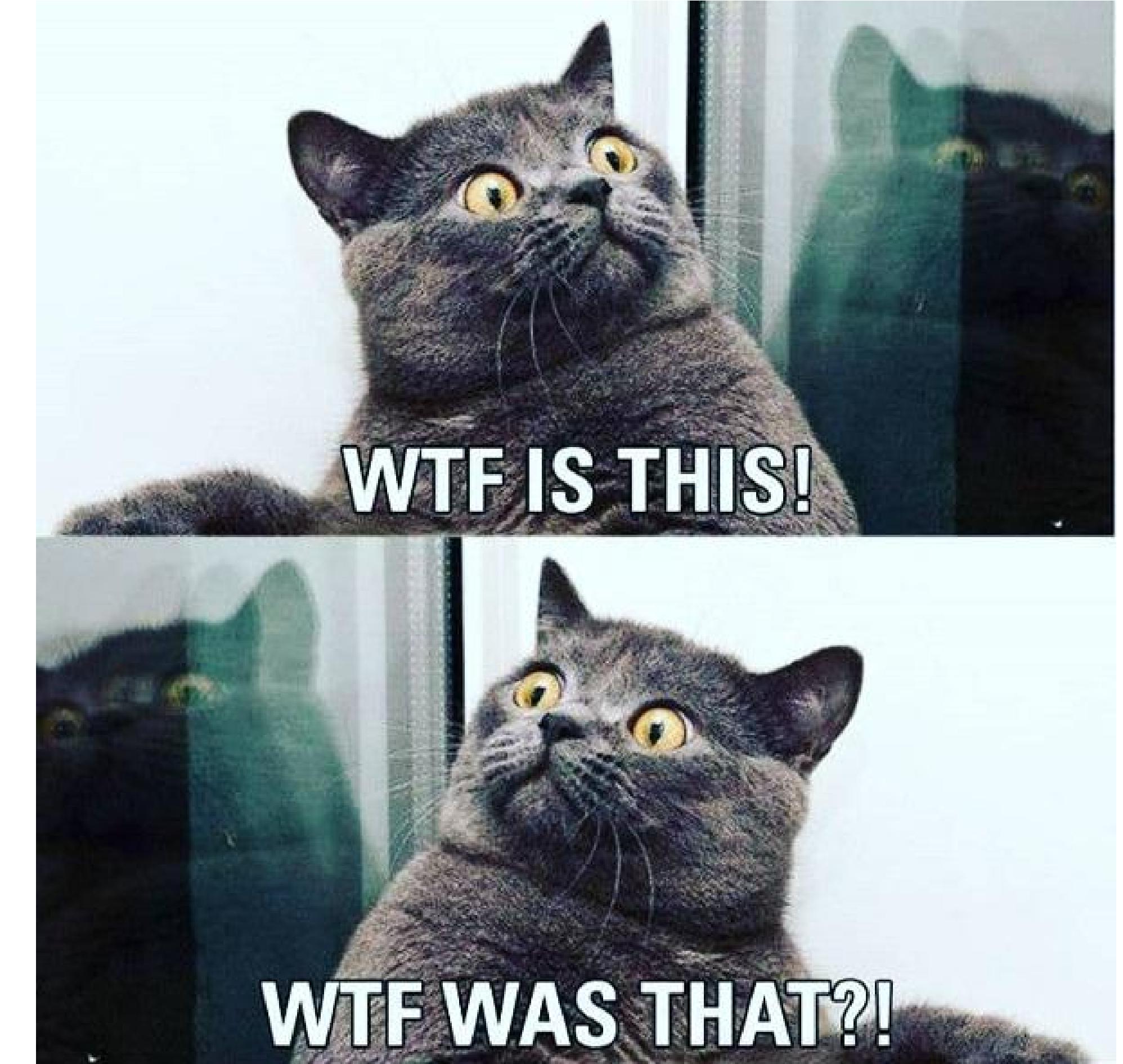
Michael Weimann  
Matrix Developer @ Nordeck IT + Consulting GmbH  
@michael.weimann:nordeck.io

the  
**matrix**  
conference



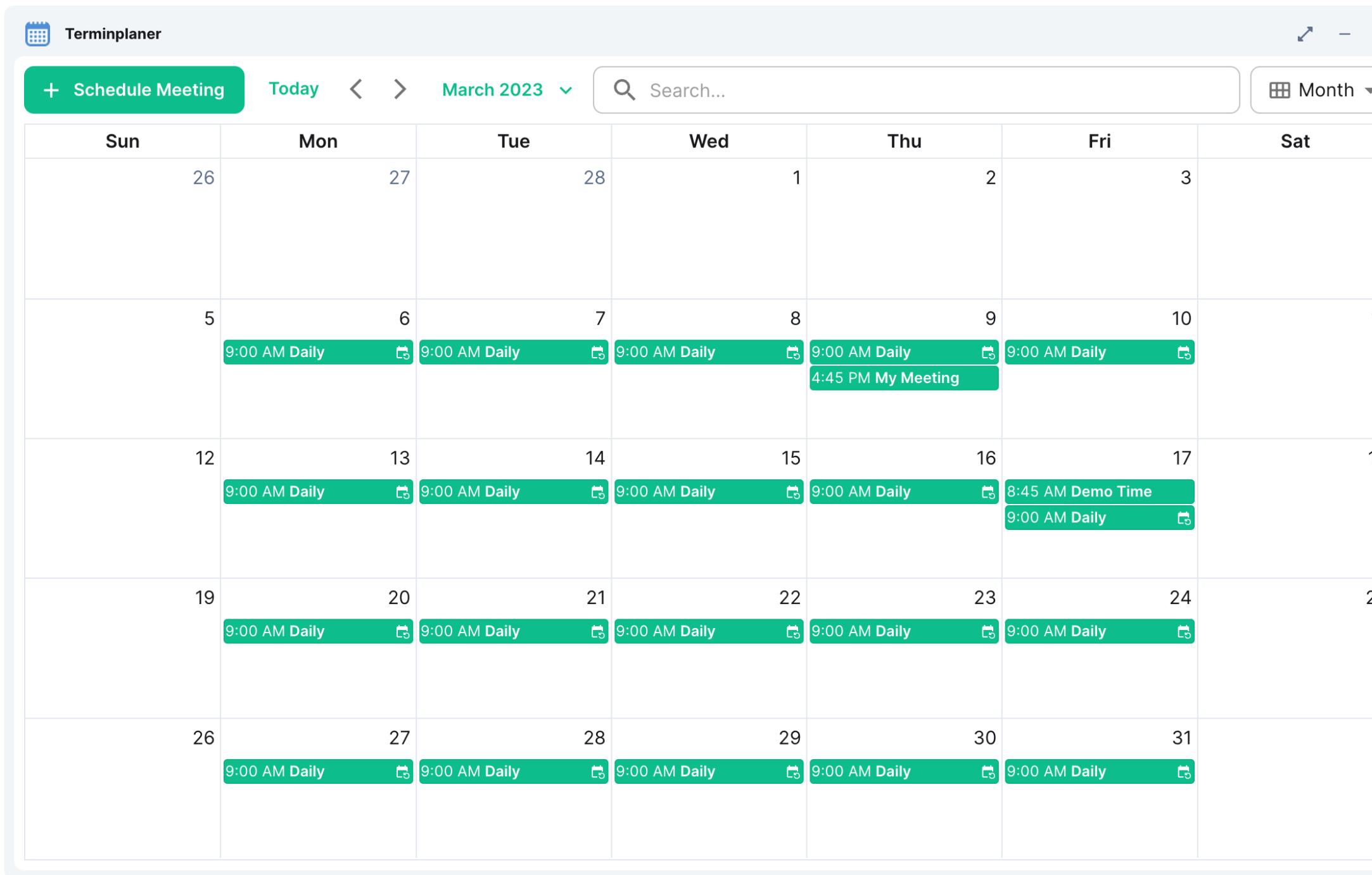
Berlin, 19th-22nd September 2024

We will take a look  
at code and spec  
documents!



- Demos
- How widgets work
- Matrix Widget Toolkit
- Standalone clients

# NeoDateFix



<https://github.com/nordeck/matrix-meetings/>

# NeoBoard

The screenshot shows the NeoBoard application interface. It features a central workspace with a grid pattern. Above the workspace, there are three main sections: "Demos" (black background), "NeoDateFix" (calendar view), and "NeoBoard" (calendar view). Below these sections, there are two smaller calendar windows. At the bottom of the interface, there are two URLs: "https://github.com/nordeck/matrix-meetings/" and "https://github.com/nordeck/matrix-neoboard". The application has a minimalist design with a white background and blue accents.

<https://github.com/nordeck/matrix-neoboard>

# Schedule Meeting

! Data on this screen is shared with meetings-widget.dpx-nordeck.at-univention.de

Title (required)

Start date  Start time

End date  End time

Description

Widgets

Repeat meeting

 End of the recurring meeting

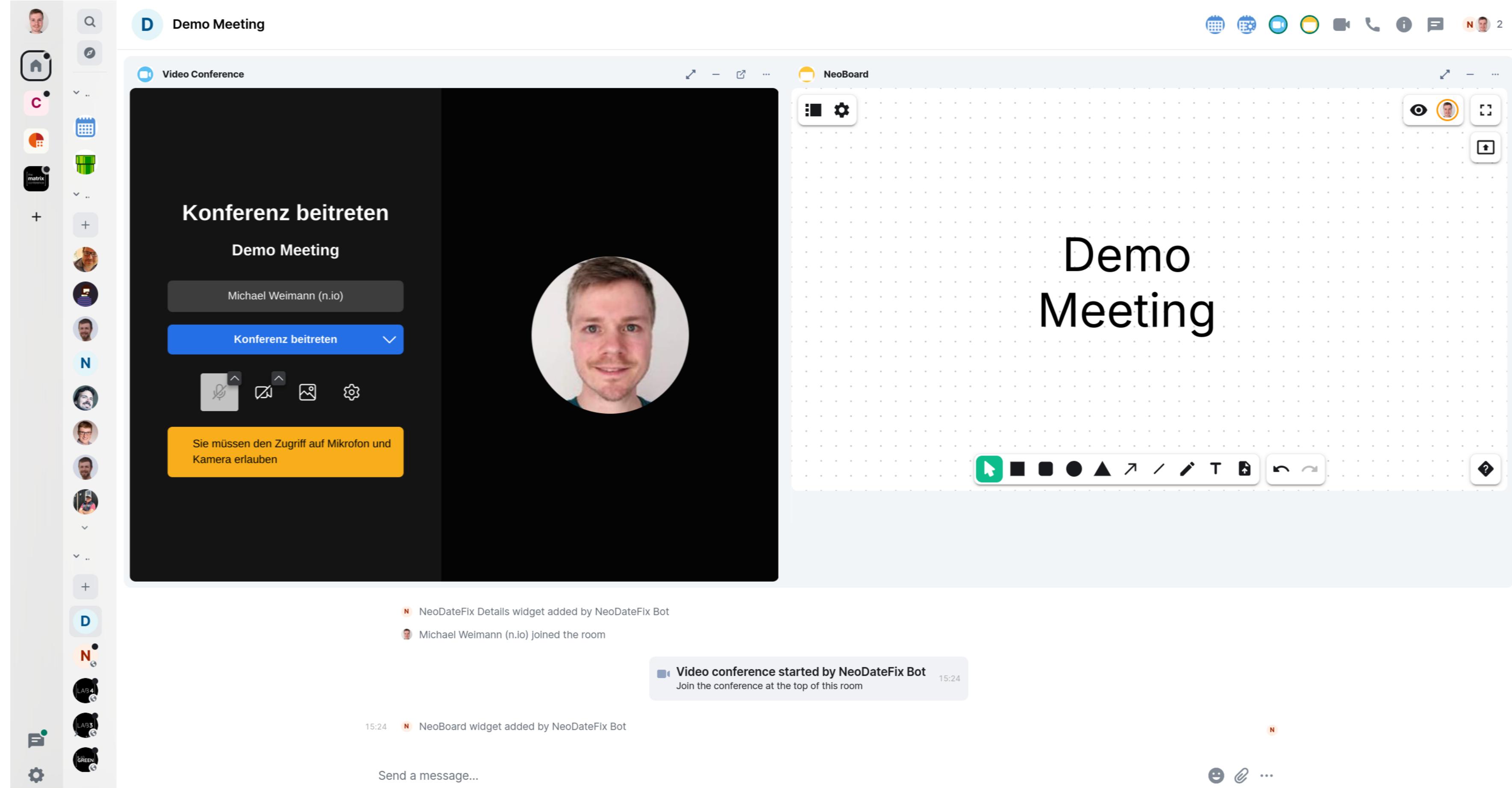
Cancel Create Meeting

<https://github.com/nordeck/matrix-meetings/>

- Calendar view with planned events

- Schedule events
  - Attendees
  - Widgets

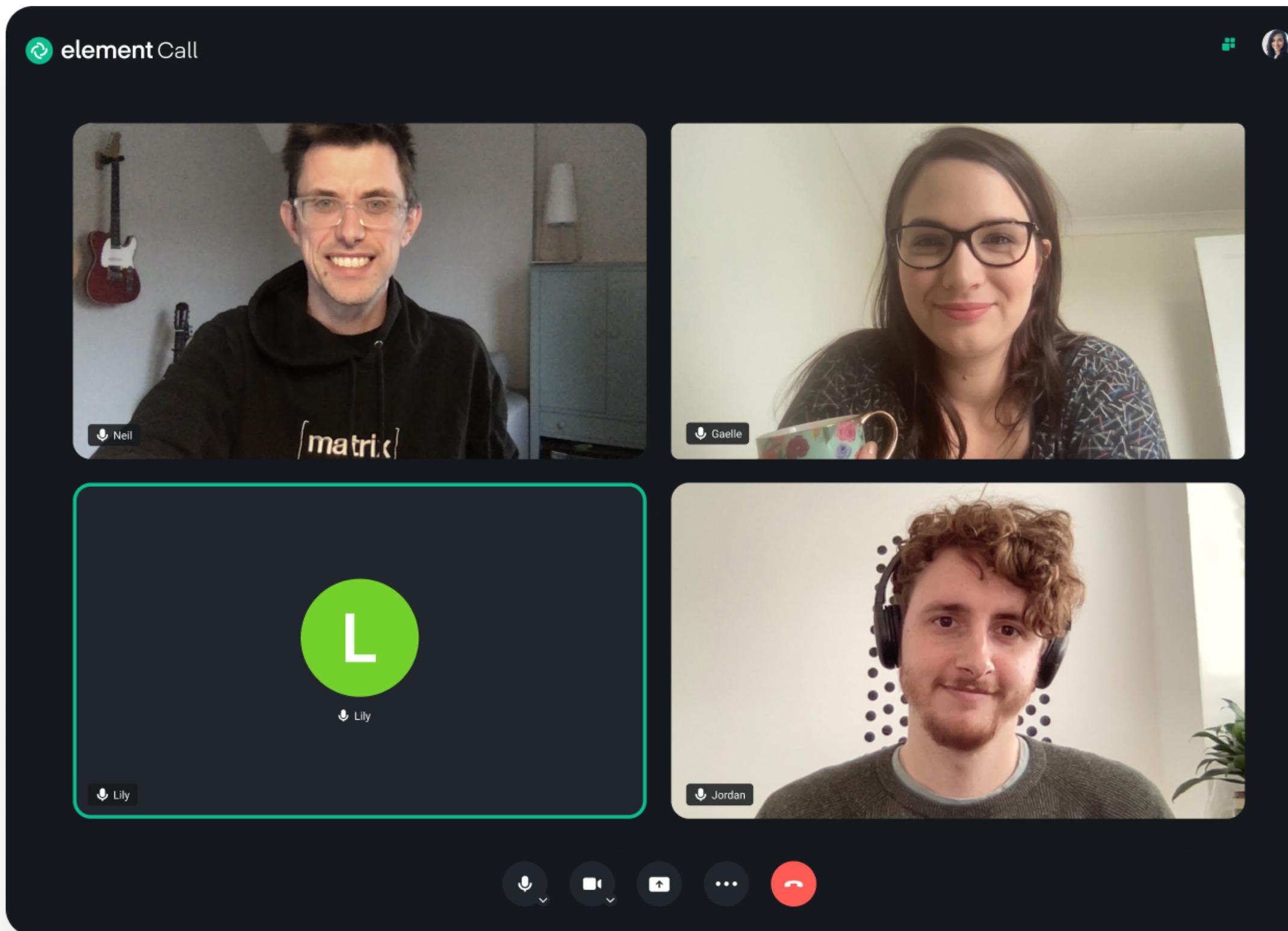
- Creates a room
  - Invites people
  - Sets up widgets



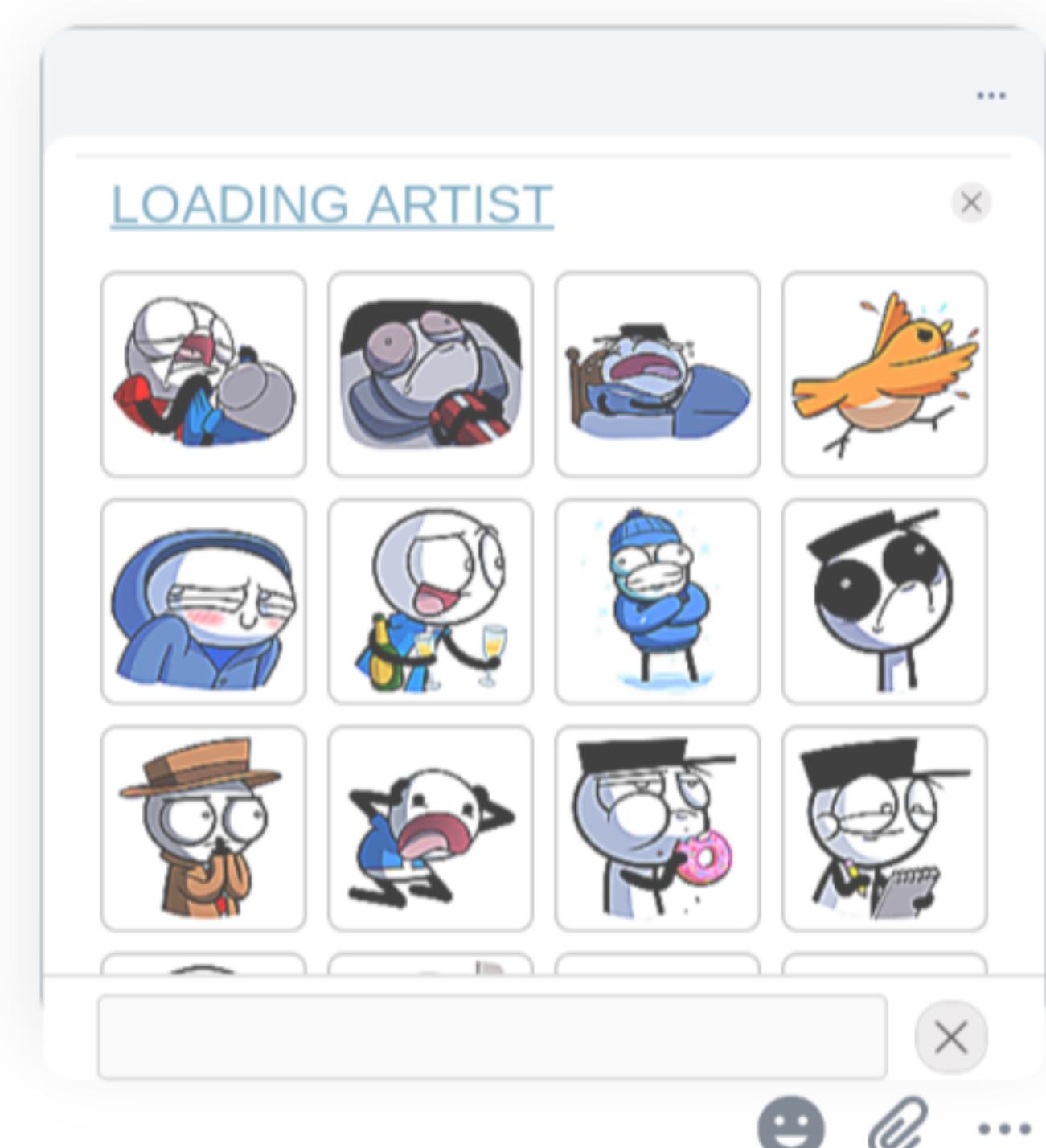
# NeoDateFix can even do more

- Recurring Meetings
- Breakout rooms
- Clean up after meetings
- Rest API (to manage meetings)

# Element Call



# Stickers



## Room widgets

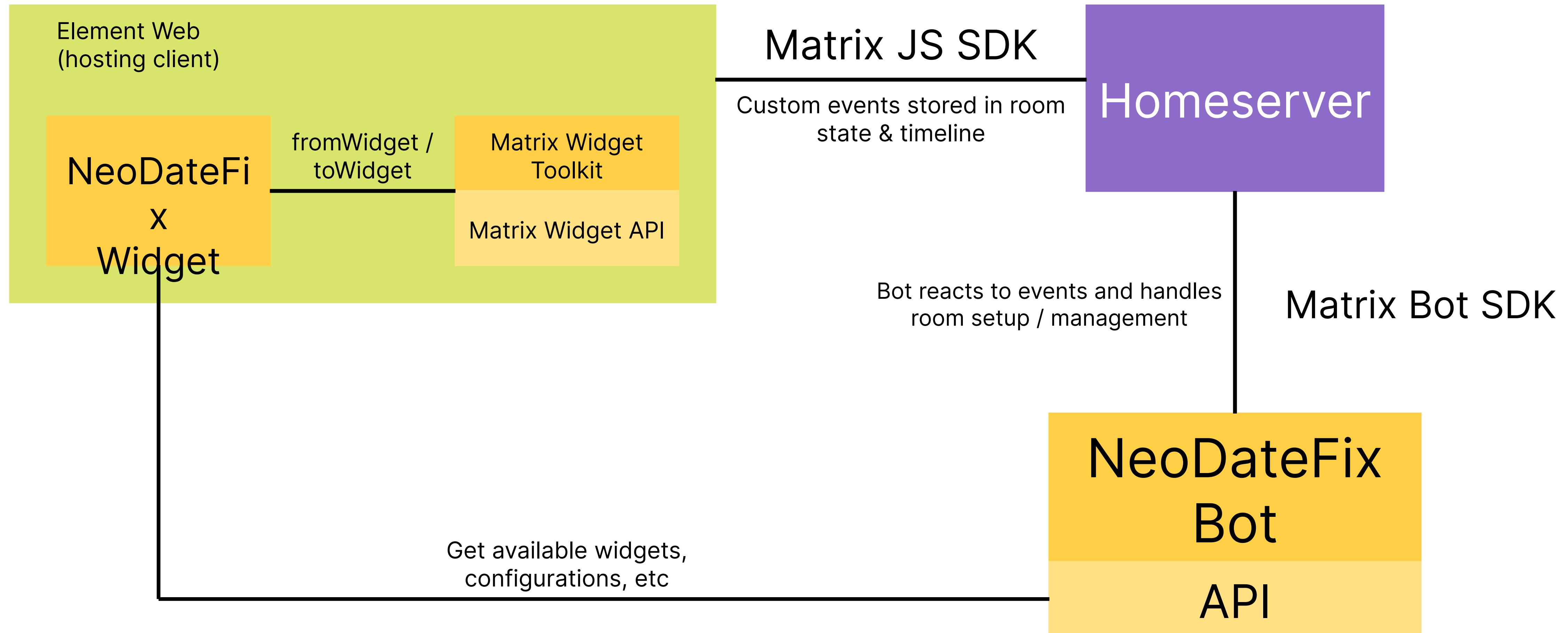
- Per room widgets
- Stored in room state  
(m.widget)

u2022 Examples: NeoBoard,  
Jitsi, Element Call

## Account widgets

- Per account widgets
- Stored in account data  
(m.widgets)

u2022 Examples: Stickers

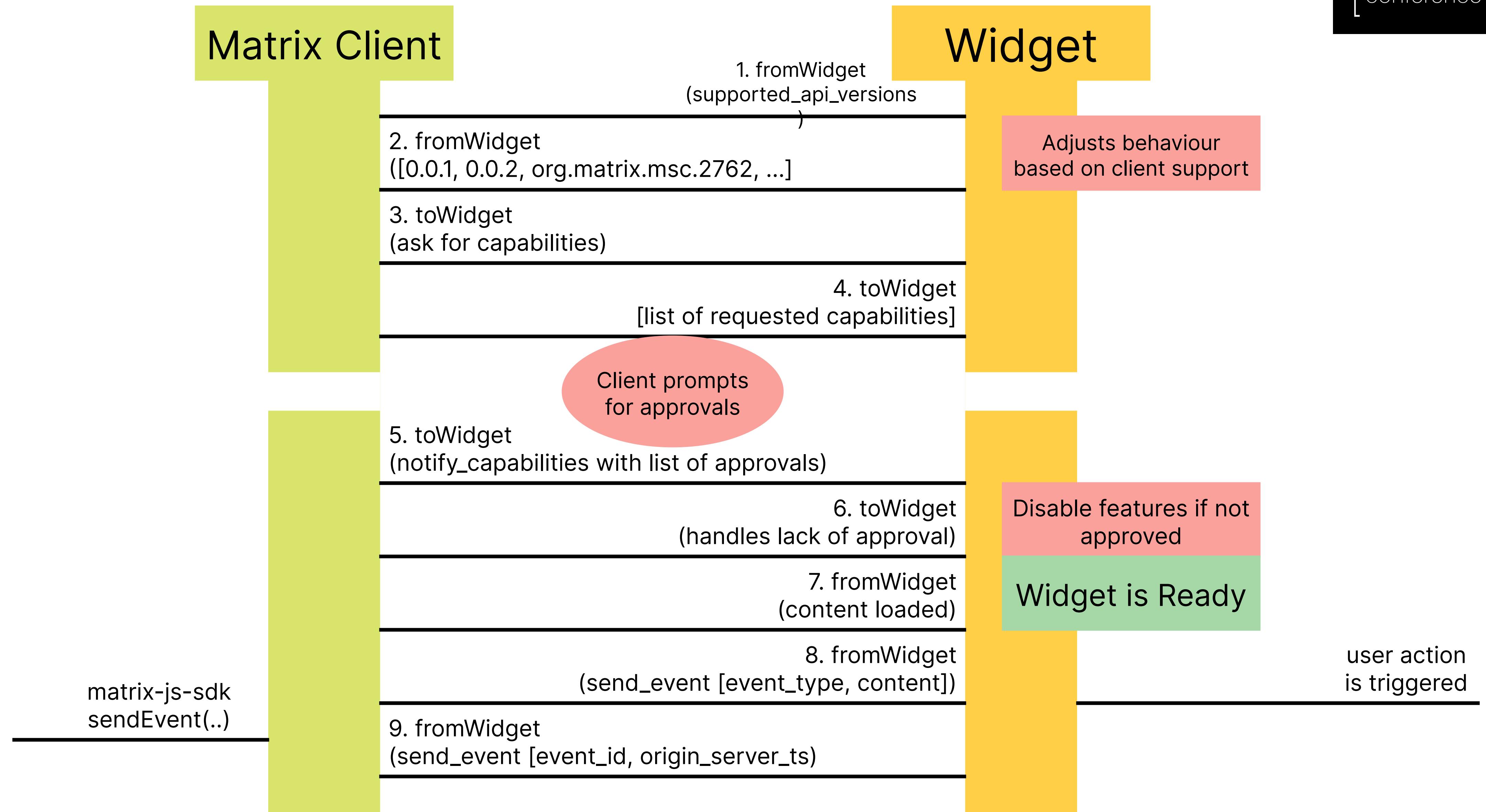


## Request

```
{  
  "api": "fromWidget",  
  "requestId": "request123",  
  "action": "send_event",  
  "widgetId": "widget123",  
  "data": {  
    "type": "m.room.message",  
    "content": {  
      "msgtype": "m.text",  
      "body": "Hello"  
    }  
  }  
}
```

## Response

```
{  
  "api": "fromWidget",  
  "requestId": "request123",  
  "action": "send_event",  
  "widgetId": "widget123",  
  "data": {  
    "type": "m.room.message",  
    "content": {  
      "msgtype": "m.text",  
      "body": "Hello"  
    }  
  },  
  "response": {  
    "event_id": "$0NkHkXBMX..."  
  }  
}
```



MSC2764  
Widgets, the  
spec

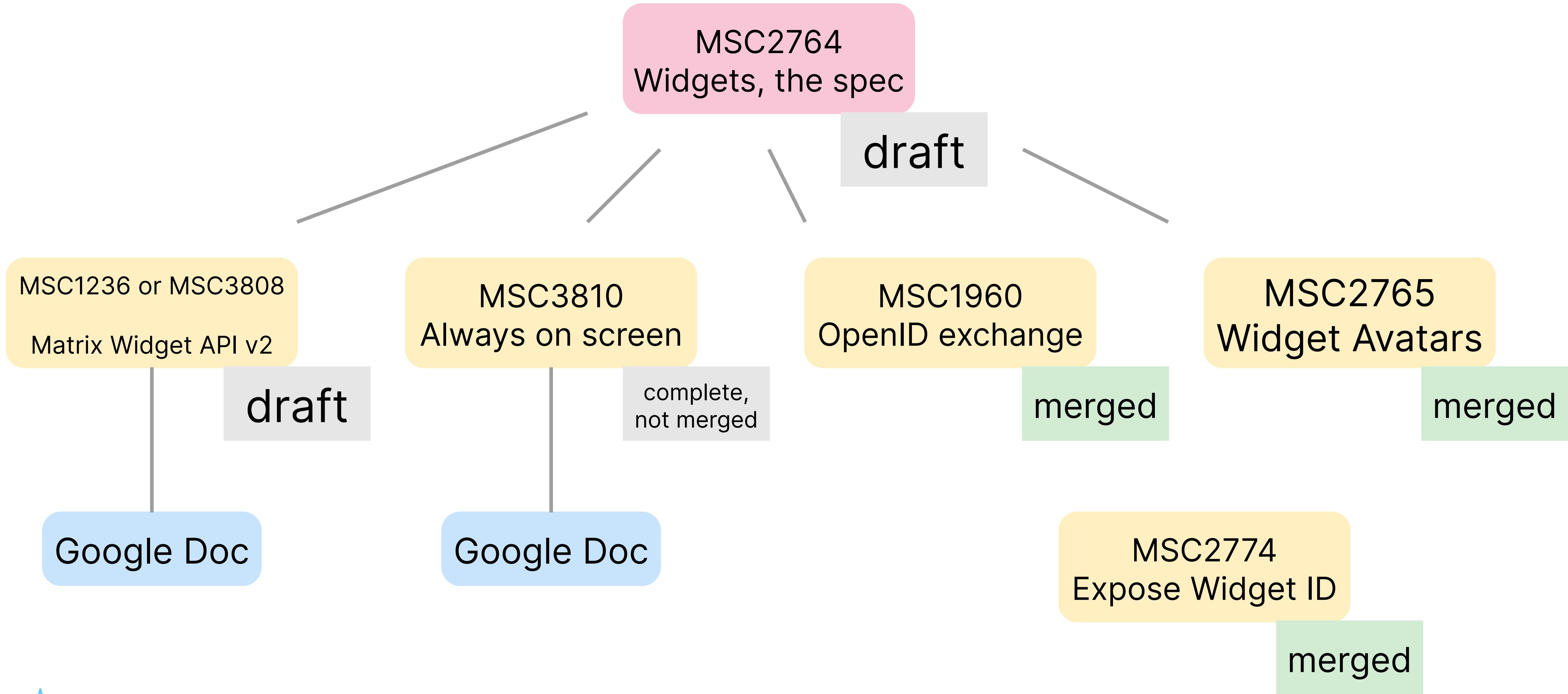
- Links to other MSCs
- Proposes a plan to get Widgets into the spec

Widgets,  
the book

- Guide to Widgets implementation
- Not an MSC

<https://github.com/matrix-org/matrix-spec-proposals/pull/2764>

<https://github.com/matrix-org/matrix.org/pull/825>



# More MSCs around widgets

MSC4039

Access content repo

MSC3973

User directory

MSC3869

Event relations

MSC3846

TURN servers

MSC3819

To-device messages

MSC2873

Settings (lang, theme...)

MSC2762

Send/receive events

MSC4157

Delayed events

MSC3751

Read account data

MSC2931

Navigate matrix.to URLs

Without implementation / drafts

MSC3817

Create rooms

MSC3008

Scoped access

MSC2192

Inline widgets

MSC2790

Modal widgets

MSC4039

Access content repo

<https://github.com/nordeck/matrix-spec-proposals/blob/nic/feat/widgetapi-upload-files/proposals/4039-widget-api-media.md>

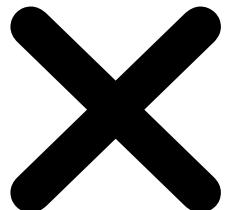
# Client Support

Web



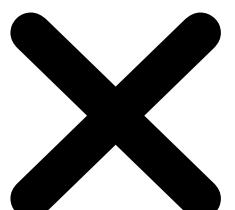
Full support

Mobile

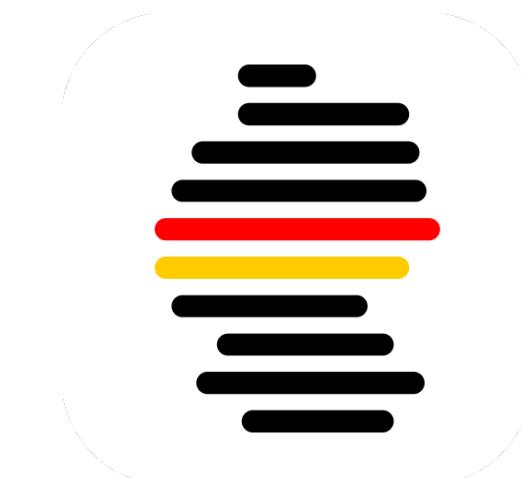


Only web views loading the page,  
effectively unusable

X

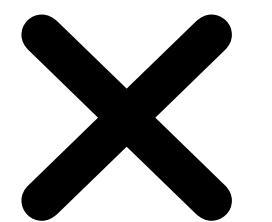


Only for Element Call



Same for Bundes-  
Messenger and other forks

# Other clients



Most likely - no support

# Convenience helpers to create widgets

- Ease of use API with RxJS
- Themed MUI components
- React integration
- Widget server image

<https://github.com/nordeck/matrix-widget-toolkit/>

Willkommen

Example on widget parameters

Identity

Example on using the OIDC token to check the user's identity

Dice

Example on how room events are send and processed

Room Admin Tool

Example on how state events are processed

All Rooms

Example on how to access events from a different room

Modal

Example on how models are opened

Power Levels (Redux)

Example on how to interact with redux

Event relations

Example on how to read related events

# Examples demo

```
const widgetApiPromise = WidgetApiImpl.create({
  // You can specify which capabilities should be requested at startup. One
  // can also request capabilities after the application started.
  capabilities: [
    WidgetEventCapability forStateEvent(
      EventDirection.Send,
      PLANNING_POKER_ROUND_EVENT,
    ),
    WidgetEventCapability forStateEvent(
      EventDirection.Receive,
      PLANNING_POKER_ROUND_EVENT,
    ),
    WidgetEventCapability forRoomEvent(
      EventDirection.Send,
      PLANNING_POKER_CARD_EVENT,
    ),
    WidgetEventCapability forRoomEvent(
      EventDirection.Receive,
      PLANNING_POKER_CARD_EVENT,
    ),
  ],
});
```

## Create Widget API

```
export const App: React.FC<AppParams> = ({ widgetApiPromise }) => {
  return (
    <MuiThemeProvider>
      <Suspense fallback={<></>}>
        <MuiWidgetApiProvider
          widgetApiPromise={widgetApiPromise} />
          widgetRegistration={{
            name: "Planning Poker",
            type: "info.weeman.planning_poker",
          }}
        >
          <Box sx={{ padding: "8px" }}>
            <PlanningPoker />
          </Box>
        </MuiWidgetApiProvider>
      </Suspense>
    <MuiThemeProvider>
    );
  };
};
```



## API provider

```
useEffect(() => {
  setDices([]);

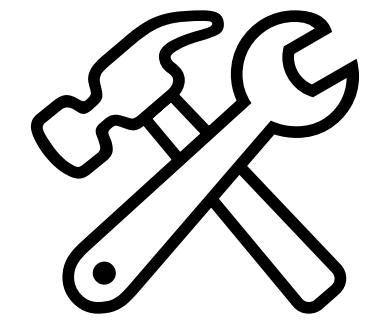
  const subscription = widgetApi
    .observeRoomEvents(STATE_EVENT_THROW_DICE)
    .pipe(
      filter(isValidThrowDiceEvent),
      map((r) => r.content.pips),
    )
    .subscribe((d) => {
      setDices((l) => [...l, d]);
    });

  return () => {
    subscription.unsubscribe();
  };
}, [widgetApi]);
```

React hook using  
RxJS to subscribe  
to the Widget API

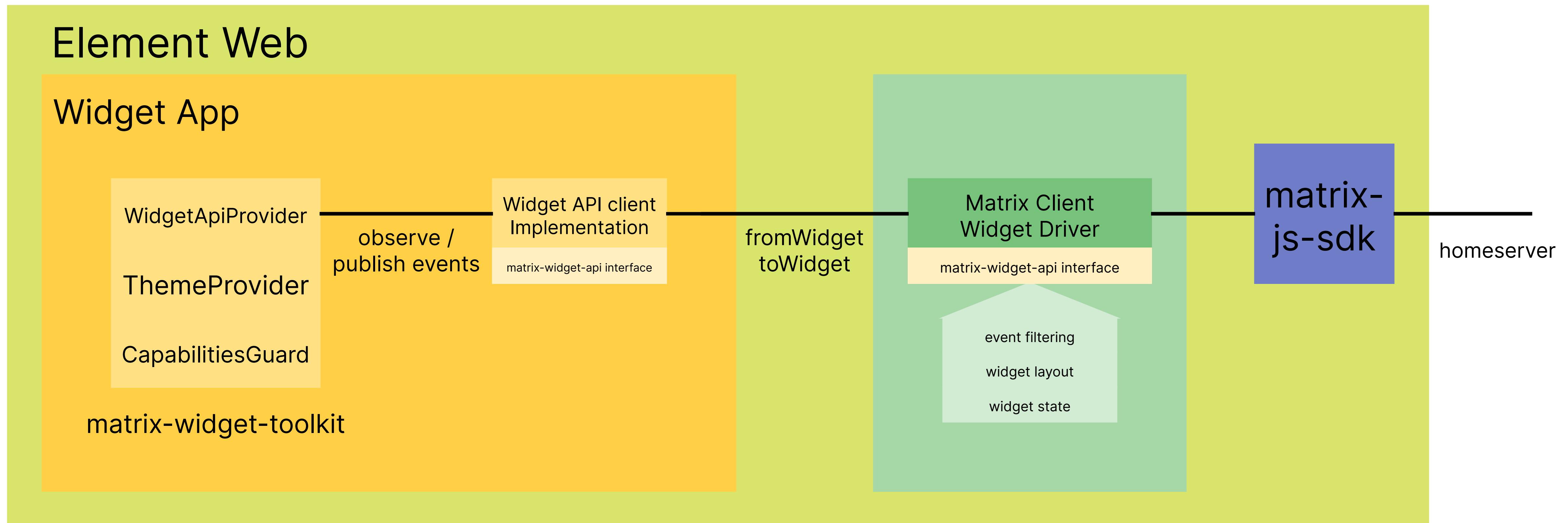
<https://github.com/nordeck/matrix-widget-toolkit/blob/main/example-widget-mui/src/DicePage/DicePage.tsx>

# Want to try it yourself?



Join the workshop  
right after this  
talk!





@matrix-widget-toolkit/



@matrix- @matrix-  
widget-toolkit/ + widget-toolkit/  
react mui

@matrix-widget-toolkit/  
testing



widget server

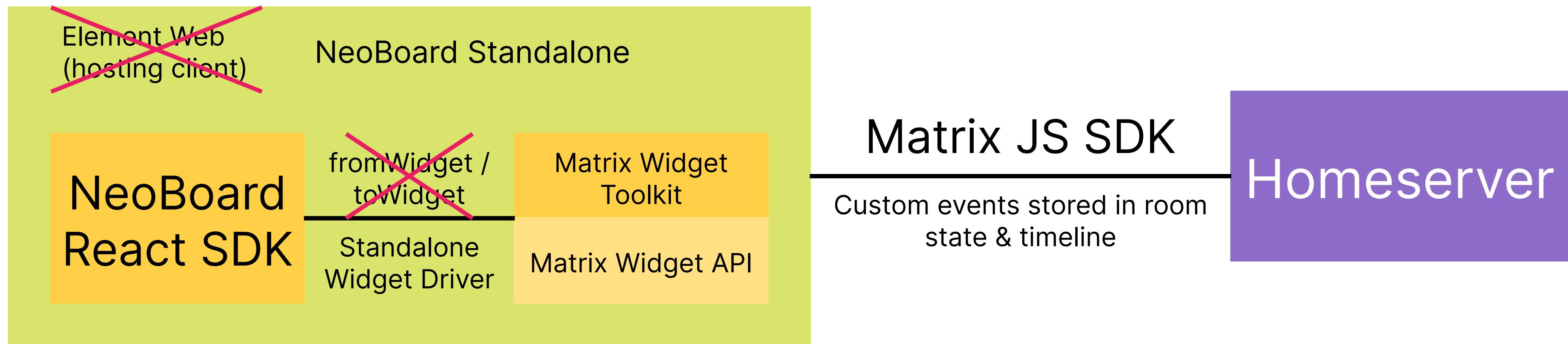
RxJS-based Widget API implementation that any widget app can use.  
Simplifies widget registration, capabilities negotiation and assorted helper functions to streamline development.

Framework-specific components for developing React based apps with a MUI design language.

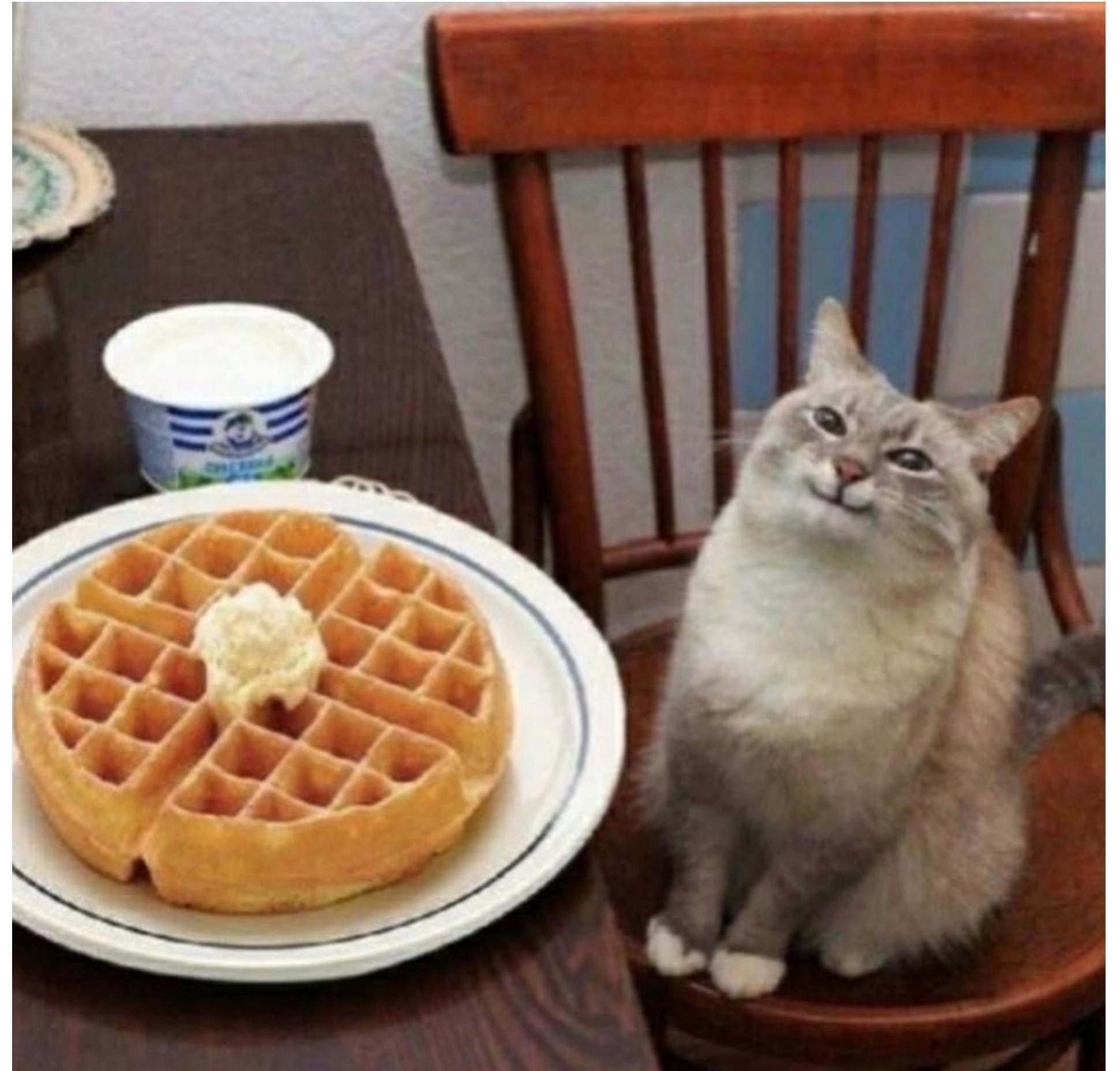
{Mui}WidgetApiProvider -> Makes the Widget API available to React components  
{Mui}ThemeProvider -> Custom theme with dark / light mode support  
{Mui}CapabilitiesGuard -> Only renders the widget content if requested caps are satisfied

A mocked Widget API implementation that you can use in your app's unit tests.

An alpine based image ready to deploy Widgets SPAs using NGINX, with required configurations (CSP, Cache Control headers, etc).



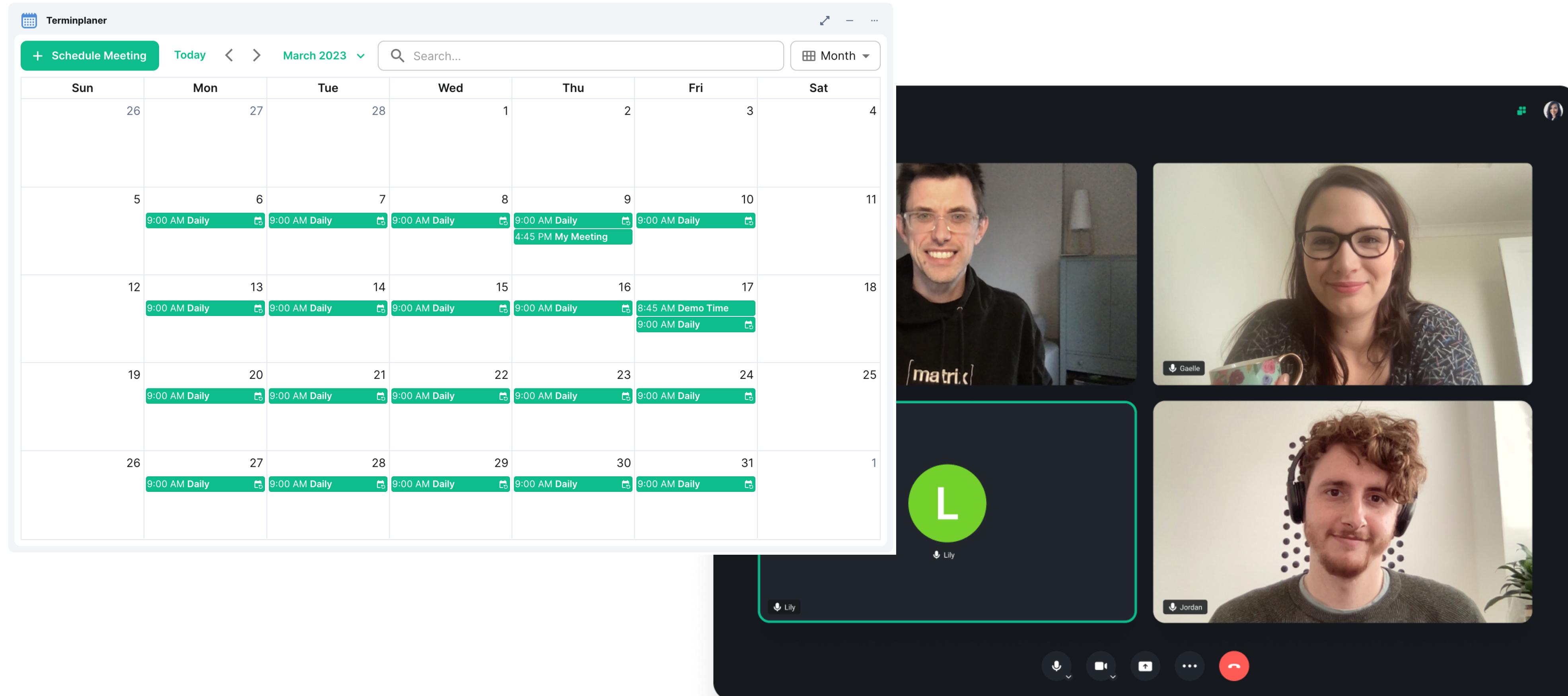
- Hybrid widgets for free™
  - Works in a Matrix Client
  - An on it's own
- Improved development workflow
  - No need to wrap the widget with a client
- Go beyond widget API
  - Use all Matrix and other APIs



- Inconsistent feature set
  - Widget vs. standalone
- New sessions
  - A standalone widget needs it's own session
  - Inconvenient crypto setup



# Widgets can be a powerful way to extend Matrix...



...however

Unclear MSC  
state

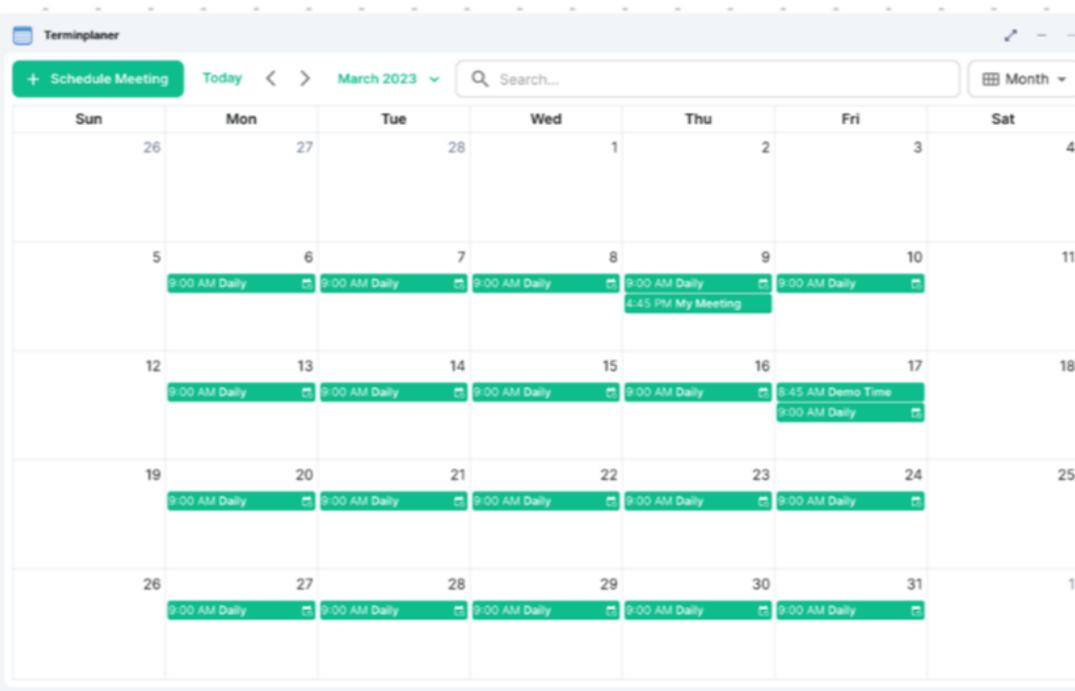
?

Limited client  
support

# Standalone clients

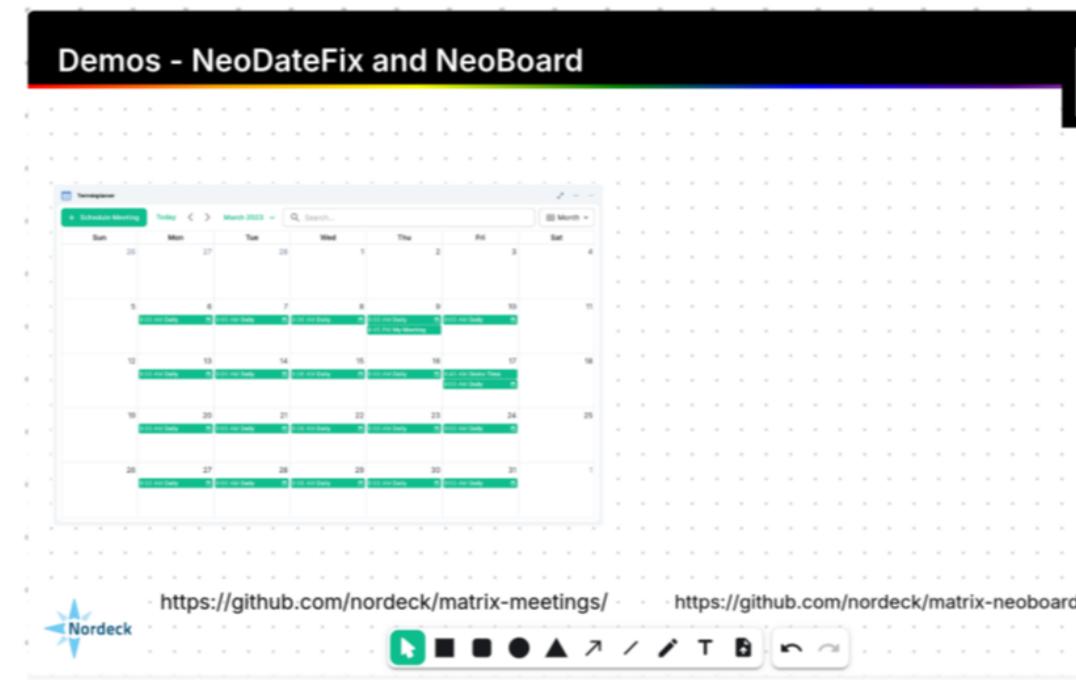
Demos

## NeoDateFix



<https://github.com/nordeck/matrix-meetings/>

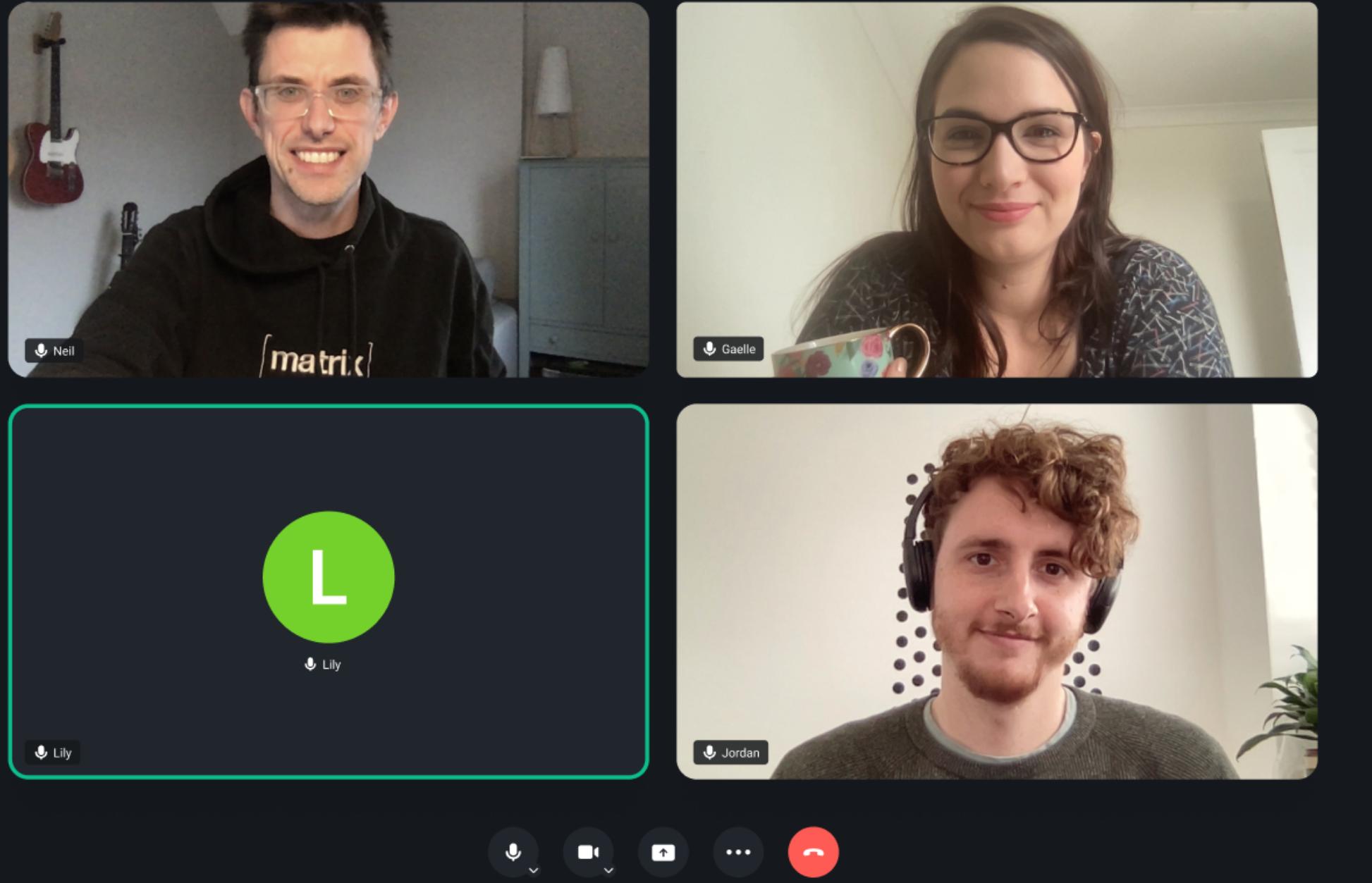
## NeoBoard



<https://github.com/nordeck/matrix-meetings/>   <https://github.com/nordeck/matrix-neoboard>

Nordeck

element Call



Neil   Gaelle   Jordan

Lily

# Wishlist

Stabilise the spec

- Tidy up MSCs
- Reflect current implementations

Extend the spec

- Inline widgets
- Scoped access tokens

# Questions?

## Workshop



short  
break

[https://t1p.de/mc-  
widget](https://t1p.de/mc-widget)