

# Element Call...

Open, Federated, and Decentralised

Redefining conferencing for privacy, scale, and sovereignty



Florian Heese

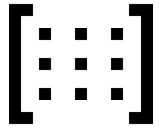


Aaron Thornburgh

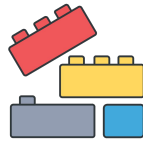


Timo Kandra

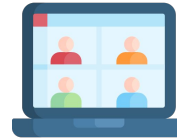
# Agenda



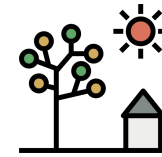
Intro to Matrix



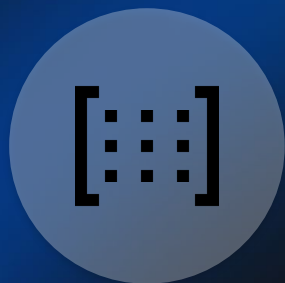
MatrixRTC



Element Call



Ecosystem



# Recap | Matrix

# Matrix is an open network for secure, decentralised real-time communication.



Interoperable chat



Interoperable VoIP



Open comms for  
VR/AR



Real-time IoT  
data fabric

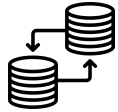
# What Matrix is really good at



Open Standard



Decentral & Federated



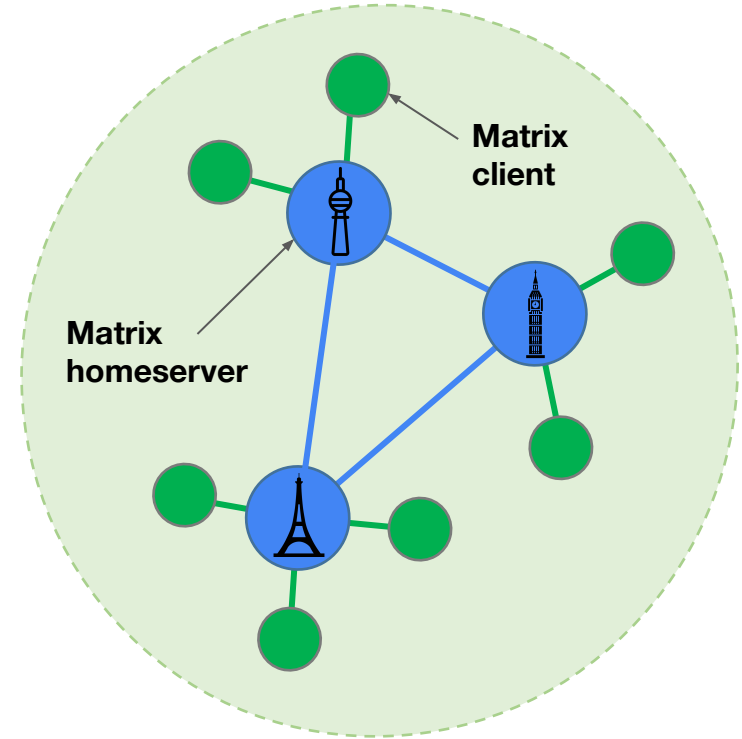
Replicated / Persistent



**End-2-End Encryption**

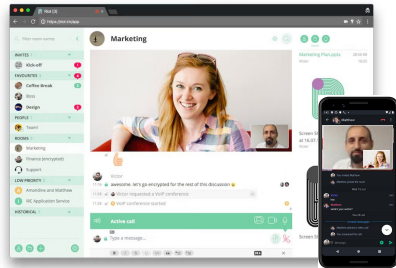


**Verified Identities**



Does it Fit Calling Requirements?

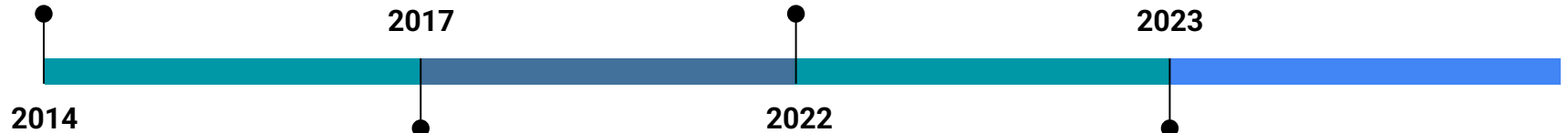
# Brief History of Matrix Calling



Native 1:1 Calling [MSC2746](#)



Native P2P Group Calling [MSC3401](#)



2014

2017

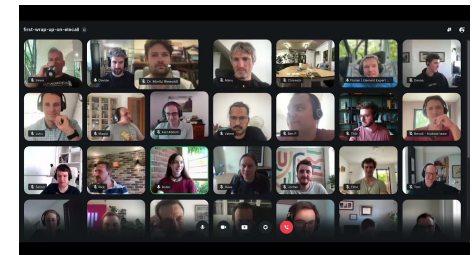
2022

2023

Jitsi via Widget (iFrame)



Group Calling Using SFU Backend for Scaling



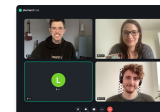
# Does it Fit Matrix DNA



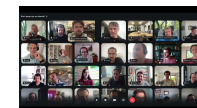
Jitsi



1:1 Calling



P2P Calling



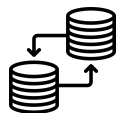
Element Call

	Jitsi	1:1 Calling	P2P Calling	Element Call
Federation	X	✓	✓	✓
(Large) group calls	✓	X	~ 5-8	✓
E2EE / Verified Identities	(✓) / X	✓	✓	✓
Interoperable	X	✓	✓	✓
Matrix compatible permissions and roles	X	✓	✓	✓
Open Standard	X	✓	✓	✓

# Does Matrix Fit Calling Requirements Yet?



Decentral & Federated



Replicated / Persistent



End-2-End Encryption



Verified Identities

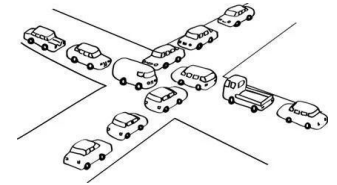


**Reliable RTC**



# The RTC Challenge...

- Getting Calls right is hard due to Matrix distributed nature
  - Eventual consistency & race conditions
- Typical symptoms in the past
  - Neverending call ringing on your second device
  - **Ghosts** / Split brains / partitions / multiple parallel calls
  - Glare causing a conflict in state machines
    - Similar to two people speaking at the same time
    - Unlike humans, computers are unable to recover gracefully



→ We Need New Matrix Primitives



# MatrixRTC



**In depth talk MatrixRTC**

Friday, 15:00h, Lynn Conway

# Make RTC a great and central part of [m]

Think beyond calls

**third room**

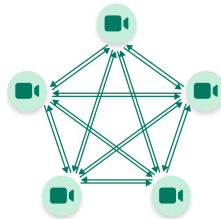
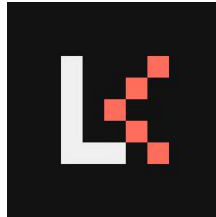


NeoBoard (Nordeck)

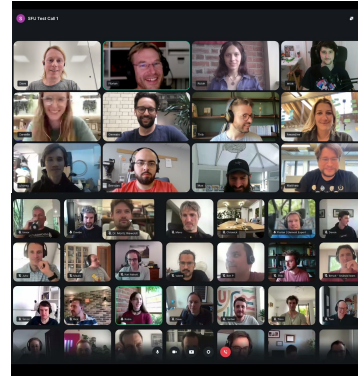


Scaling Real-Time in Neoboard  
Sat, 12.10h, Alan Turing

Pluggable  
RTC backend



Large, interop.,  
secure group calls



Support many  
clients



+

**Others?**

(widget support EC)  
Or any livekit sdk

# MatrixRTC A Structured RTC Framework

## Applications

Defines activity type and logic



## Slots

Govern what kind of app runs, permissions & config



## Membership

record of who is actively participating, and under which transports and devices.



## End-to-End Encryption

Ensures secure media exchange, mechanisms for key sharing and rotation



## Transport

How participants exchange media



# Replacing Room State with Sticky Events



✓ E2E Encrypted Membership

✓ Reliable Event Delivery

✓ No Room State Bloating

✓ No State Resolution Overhead

Sticky events reduce metadata and increase efficiency

# MatrixRTC Ready for Spec Review



Concluded R&D



[MSC4140](#) Cancellable Delayed Events



**[New]** Slots → Management of RTC Apps



**[New]** Sticky Events [MSC4354](#) → Reliable & Lightweight

# MatrixRTC at a Glance



Base Layer [MSC4143](#) MatrixRTC



Separation of concerns

- Pluggable RTC Transports
  - [MSC4195](#) livekit - a backend using the [LiveKit](#) SFU
  - [MSC3401](#) full\_mesh - a backend using a pure WebRTC full-mesh approach
- Applications
  - [MSC4196](#) MatrixRTC voice and video conferencing application
  - [MSC4075](#) MatrixRTC Call Ringing

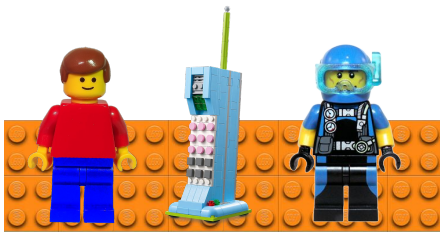


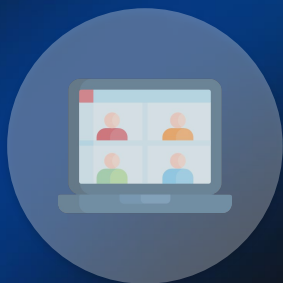
**New:** Reliable RTC state (implemented in Synapse)

- [MSC4140](#) Cancellable delayed events (aka Futures)
- [MSC4354](#) Sticky Events

# The m.call Application

- [MSC4196](#) first standardized MatrixRTC application
  - Familiar & Interoperable Calling Experience
  - **Aim:** Lowest common denominator for Calling in Matrix
- Supported Call Types
  - 1:1 calls in direct messages – private, peer-to-peer conversations.
  - Huddles – Adding a call to an existing room for spontaneous group discussions.
  - Persistent “social” rooms – act as a permanent hangout

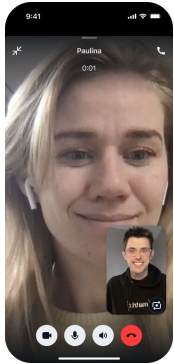




# Element Call

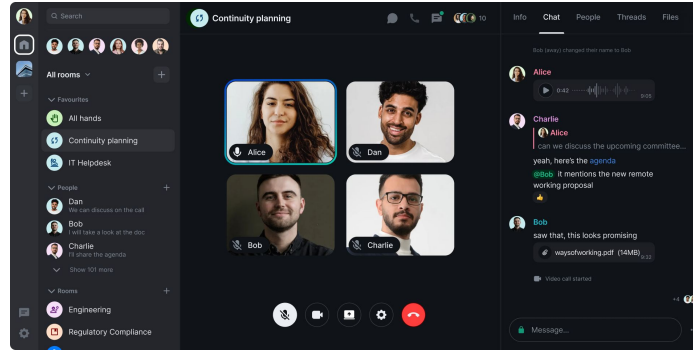
# We Cover all Call & Video Scenarios

1:1



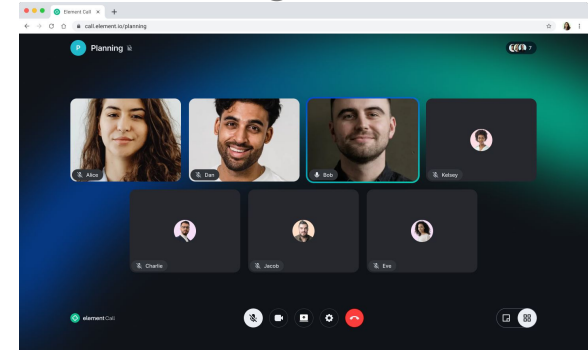
- Telephone call
- Ringing by default

Ad hoc in-chat



- Start a call in existing matrix room
- Ringing as part of notification settings

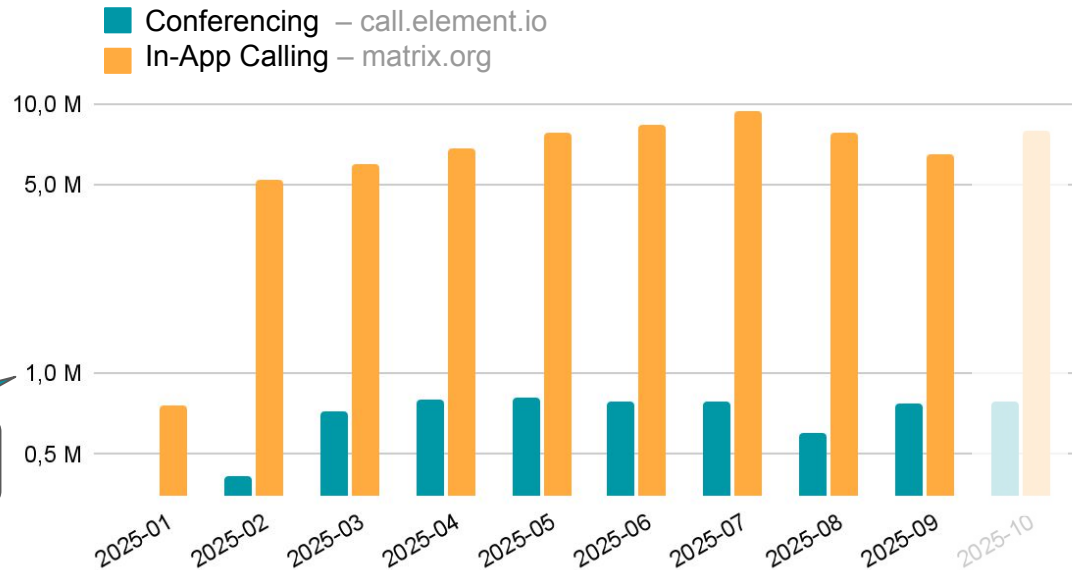
Conferencing / Standalone



- Easy to share
- Features **Guest Access**

# Is Matrix Calling a Thing?

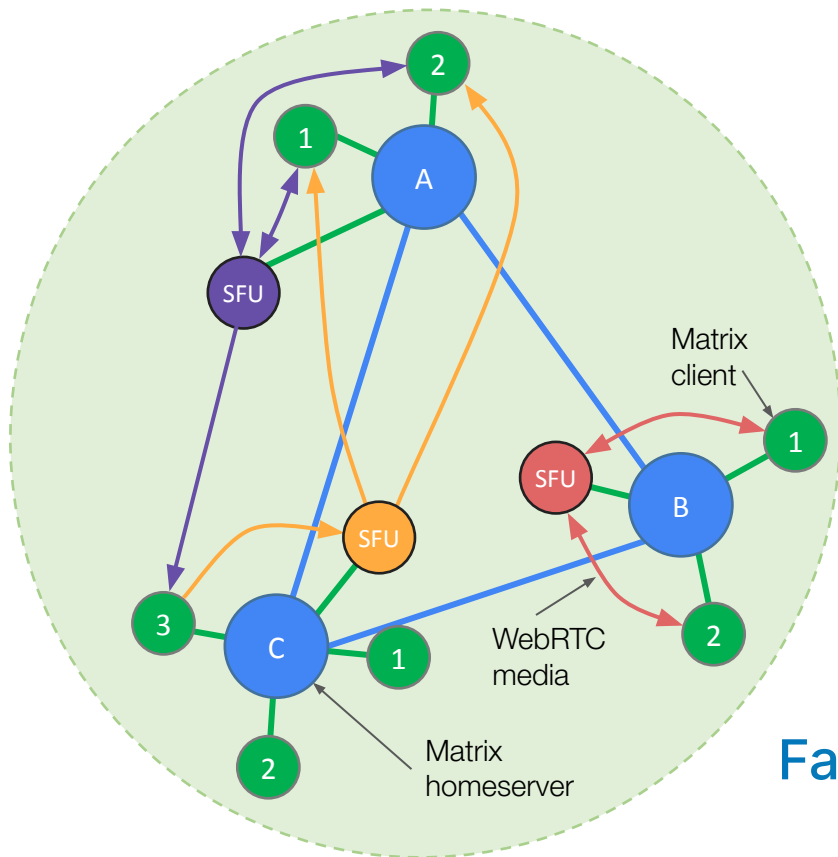
Meeting Minutes per Month



694 days ≈ **1,9 years**  
of nonstop meetings

  
6944 days ≈ **19 years**  
of nonstop meetings

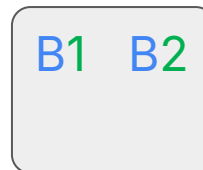
# New: Multi SFU Federated Setup



## Mental model for MatrixRTC Backend

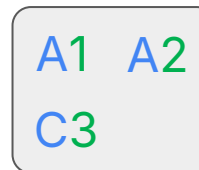
- SFU alongside each homeserver
- Serves all rooms of that server (also federated ones!)

RoomID: 123



SFU B

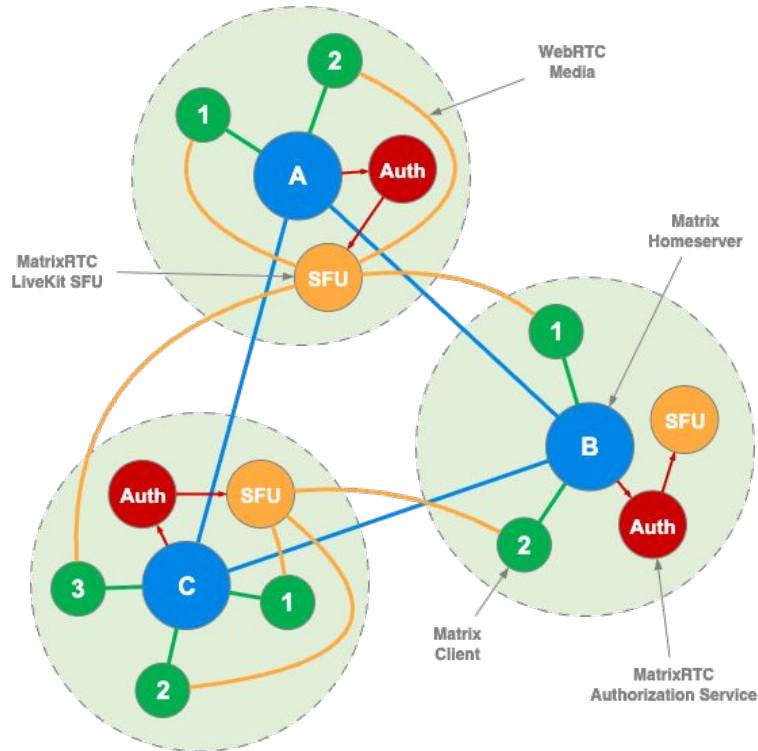
RoomID: 456



SFU A  
SFU C

Fair distribution of backend resources

# New: LiveKit RTC Transport Access



## Full Access Users

- Users from related Matrix Site(s)
- Trigger LiveKit room creation
- Can Publish and Subscribe Media

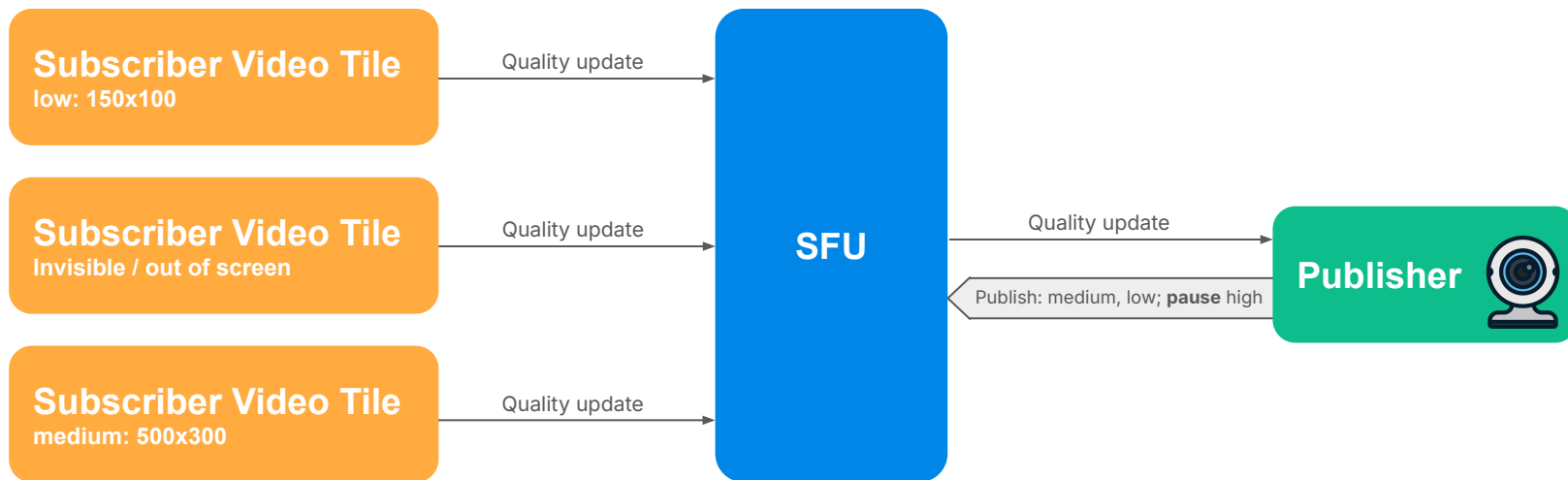
## Restricted Access Users

- All other Users
- DO NOT trigger LiveKit room creation
- Can only Subscribe Media

Proper and Secure use of infrastructure

# Smart Selective Forwarding Unit

Economical and efficient use of the available network bandwidth










- Up to 500 participants per machine (e.g., AWS, ARM64, 4 Cores, 8GB RAM)
- Horizontal scaling using k8s

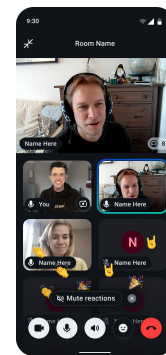
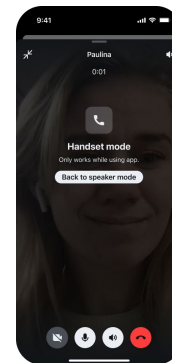
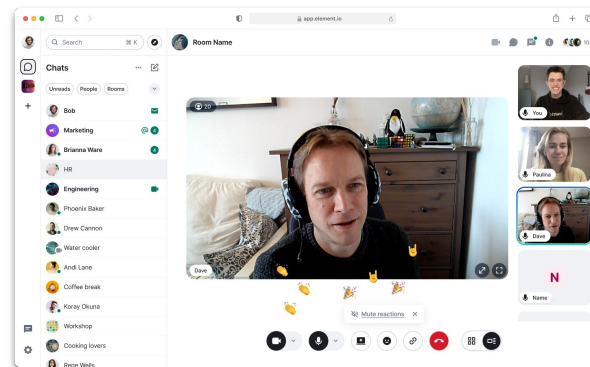
# Out Of Beta – We are Close

## MatrixRTC Mature Enough

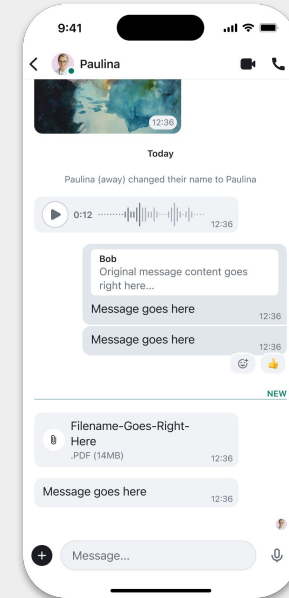
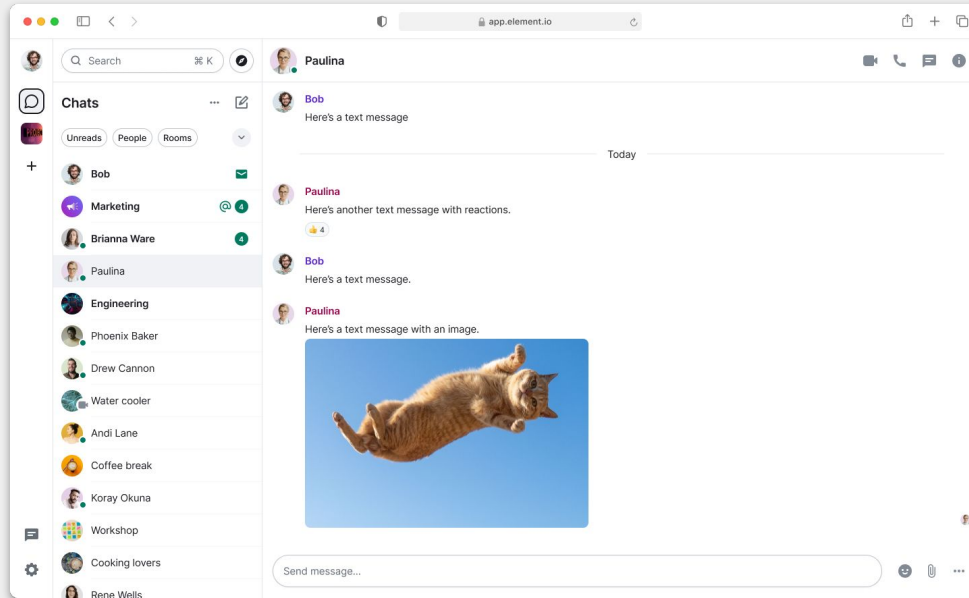
- ✓ Call Signaling (Ringing, Notification)
- ✓ Slots, Multi SFU
- ✓ Cancable Delayed Events, Sticky Events

## Feature Support

-  Conference meta-space
-  Screen Sharing
-  Emoji Reactions &  Raise Hand
-  Native CallKit integration
-  Handset Mode
-  Picture-in-picture mode



# Design Philosophy In-App Calling





# Ecosystem

# Element Call | Embedding

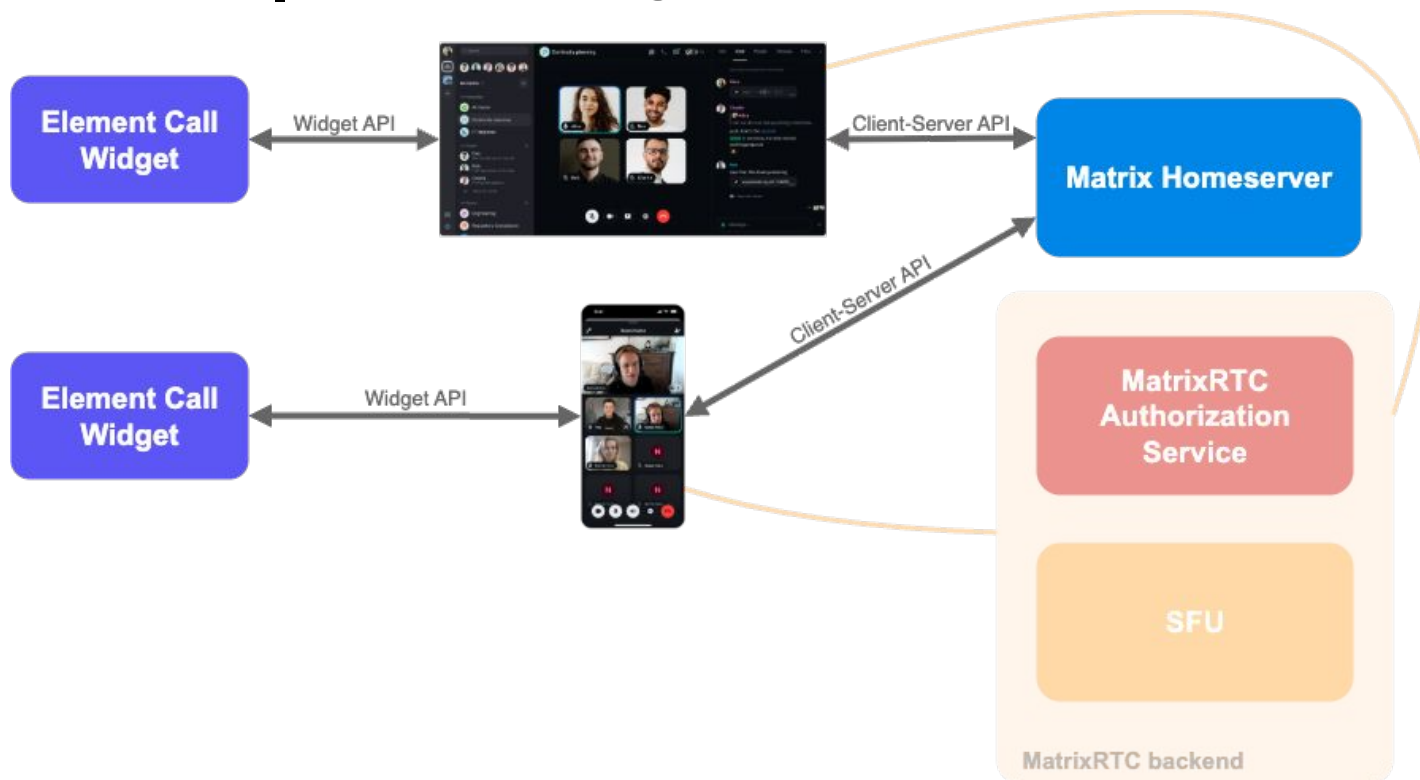
From the MatrixRTC Spec [MSC4143](#)

“

To facilitate interoperability, **ideally** each application type should **provide** a Matrix **widget** that can serve as a reference implementation for clients. This allows client developers to support new application types by embedding or integrating the widget, without having to implement the full application logic themselves.

”

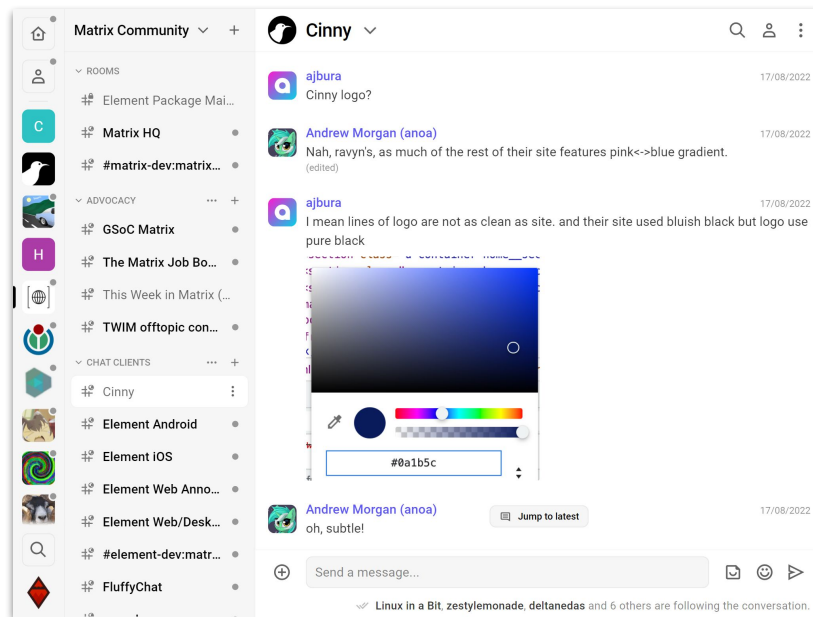
# Element Call | Embedding



 Requires Widget driver on each platform (👍 js-sdk/react-sdk, 👍 rust-sdk)

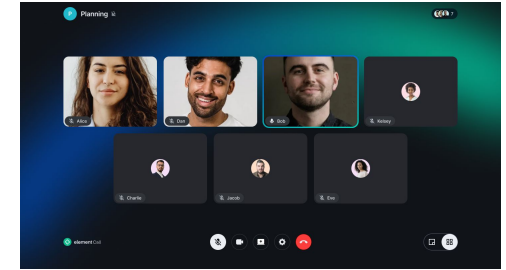
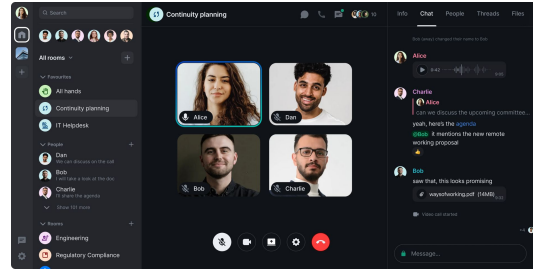
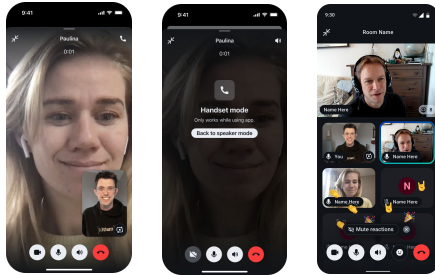
# Adding Element Call to \$Client

Let's add the Element Call Widget to Cinny...

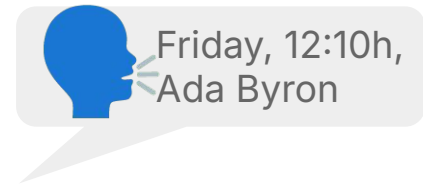
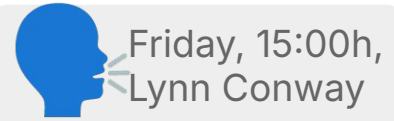


# Summary

# Summary



- ✔ MatrixRTC is ready for Matrix calling 🎉 🎊 🚀
  - We delivered **more than 139 years** of non-stop meetings
- ✔ Lets push `m.call` across all Matrix Clients
  - [MSC4196](#) MatrixRTC voice and video conferencing application
- ✔ Element Call Widget implements `m.call`
- ✔ Deep integration in Element Apps incl. CallKit and PiP



MatrixRTC compatible Site Deployment via Element Server Suite Pro

# Element Call...

Open, Federated, and Decentralised

Redefining conferencing for privacy, scale, and sovereignty



Florian Heese



Timo Kandra



Aaron Thornburgh

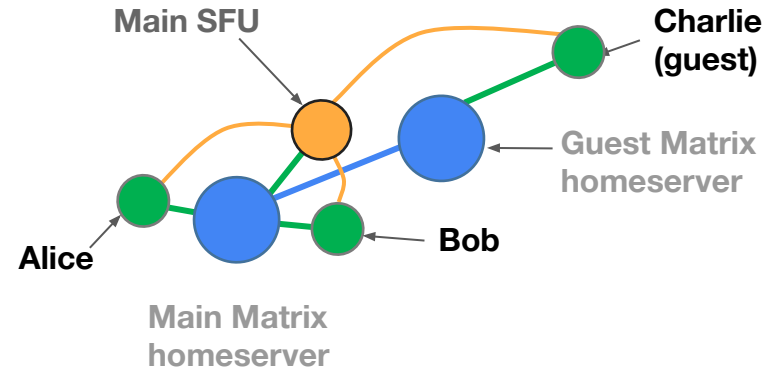




# Guest Access

# Matrix Allows Separation of Guest Users

- A Call happens in a (new) Matrix room
  - No guest users on main homeserver
- Guests join via separate homeserver
  - Temporary accounts created on the fly
  - Account retention possible
  - User only required to provide a displayname
  - Main homeserver joined via federation
- Guest user access control is **secured**
  - Knocking request granted on the fly by call moderators / admins



# Call Member Events

## Call Member event

```
"content": {  
  "memberships": [  
    {
```

```
      "device_id": "DEVICEID123",  
      "expires": 3600000,  
      "foci_active": [  
        {  
          "livekit_alias": "!NXHzTvNOwsFiZaAvTT:matrix.org",  
          "livekit_service_url": "https://livekit-jwt.call.element.dev",  
          "type": "livekit"  
        }  
      ],  
      "application": "m.call",
```

**MatrixRTC**

```
    }  
  ],  
  "call_id": "",  
  "scope": "m.room",  
  "linked_event": "$Azt5QD7kRbOq19IyqWoNUtmk6ulGsUSCgKLT6Bvzs-Y",  
  "other_shared_or_individual_data": "100,200"  
}
```

**Application Specific**  
Fields defined by `m.call`

```
]  
}  
"state_key": "@user:matrix.org"
```

## Room

New

Call Member  
event

Call Member  
event

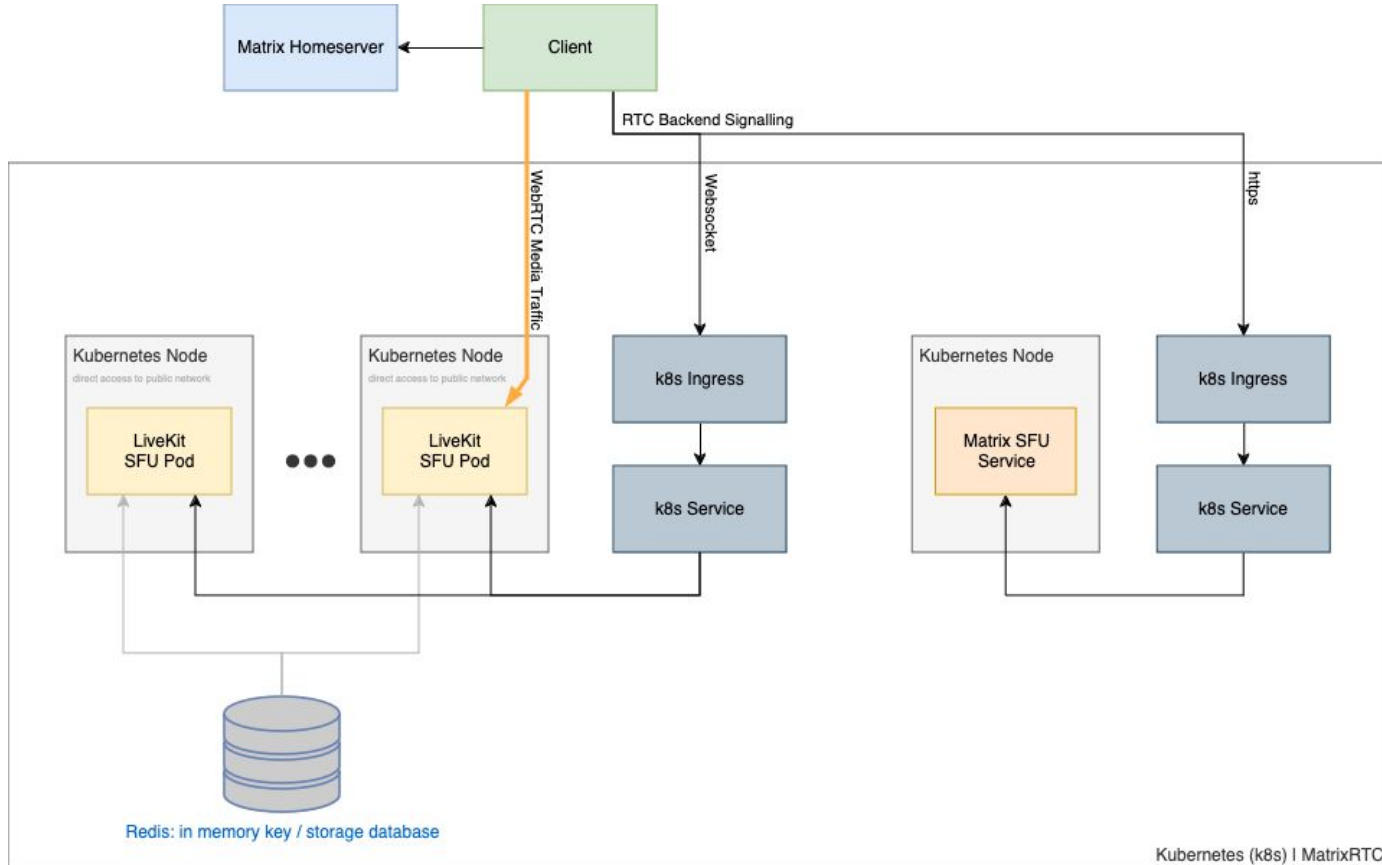
Call Member  
event

Call Member  
event

Call Member  
event

Call Member  
event

# Horizontal Scaling



# Element Call | Guest Access UX

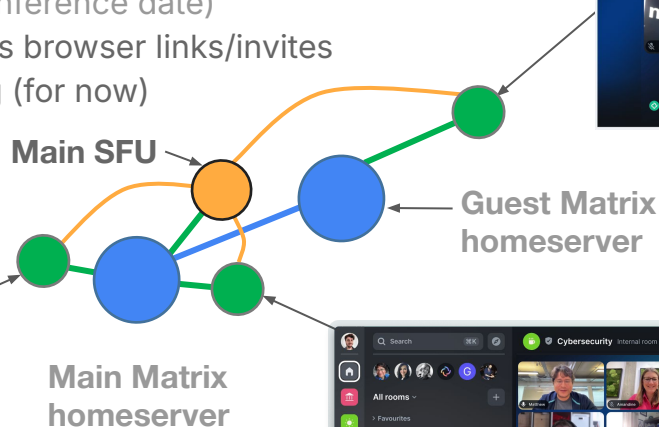
- Guest access allows joining via web browser
  - New UX flow in the Messenger App
  - Dedicated Calls/Conference section
    - Each new Call/Conference is a new matrix room
    - Video first UX (sorted by conference date)
    - Allows to share guest access browser links/invites
    - Access control via Knocking (for now)



Guest via web browser



Messenger App



Messenger App

# More Use-Cases for Guest Access

- Knocking
- Guest Access works the same way for other MatrixRTC applications like NeoBoard