Scaling Realtime in NeoBoard

Our journey from P2P WebRTC to MatrixRTC

Milton Moura
Matrix Specialist @ Nordeck
@milton.moura:nordeck.io

the matrix conference

Nordeck

Strasbourg, 19th October 2025

What is NeoBoard

openDesk Whiteboard ⇒ NeoBoard

openDesk

Good afternoon, Milton.

+ Create new in Files

Applications

Files
Email
Calendar
Contacts
Tasks
Projects
Knowledge
Videoconfer-ence
Chat
Notes
Whiteboard

openDesk

Applications

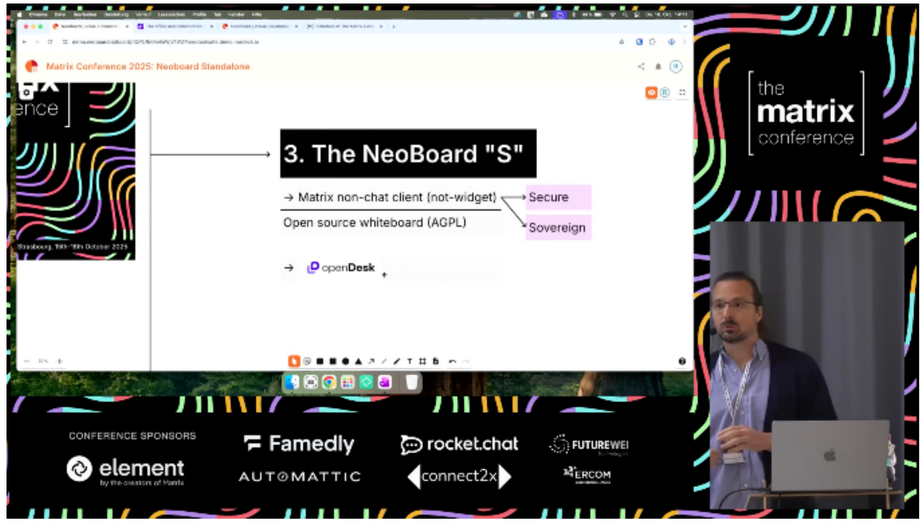
Files
Email
Calendar
Contacts
Tasks
Projects
Knowledge
Videoconference
Chat

Sort by:

matrixRTC and PDF import
Last view 16 hours ago

Sprint 14 Starship review
Last view -

Sprint 14 Starship retrospective
Last view -



NeoBoard S: a whiteboard matrix client offering endless possibilities for real-time collaboration

Thursday, October 16, 2:10 PM, Lynn Conway, Beyond Chat

Last year, we introduced NeoToolSuite, Nordeck's productivity suite of Matrix widgets designed for efficient collaboration and communication.

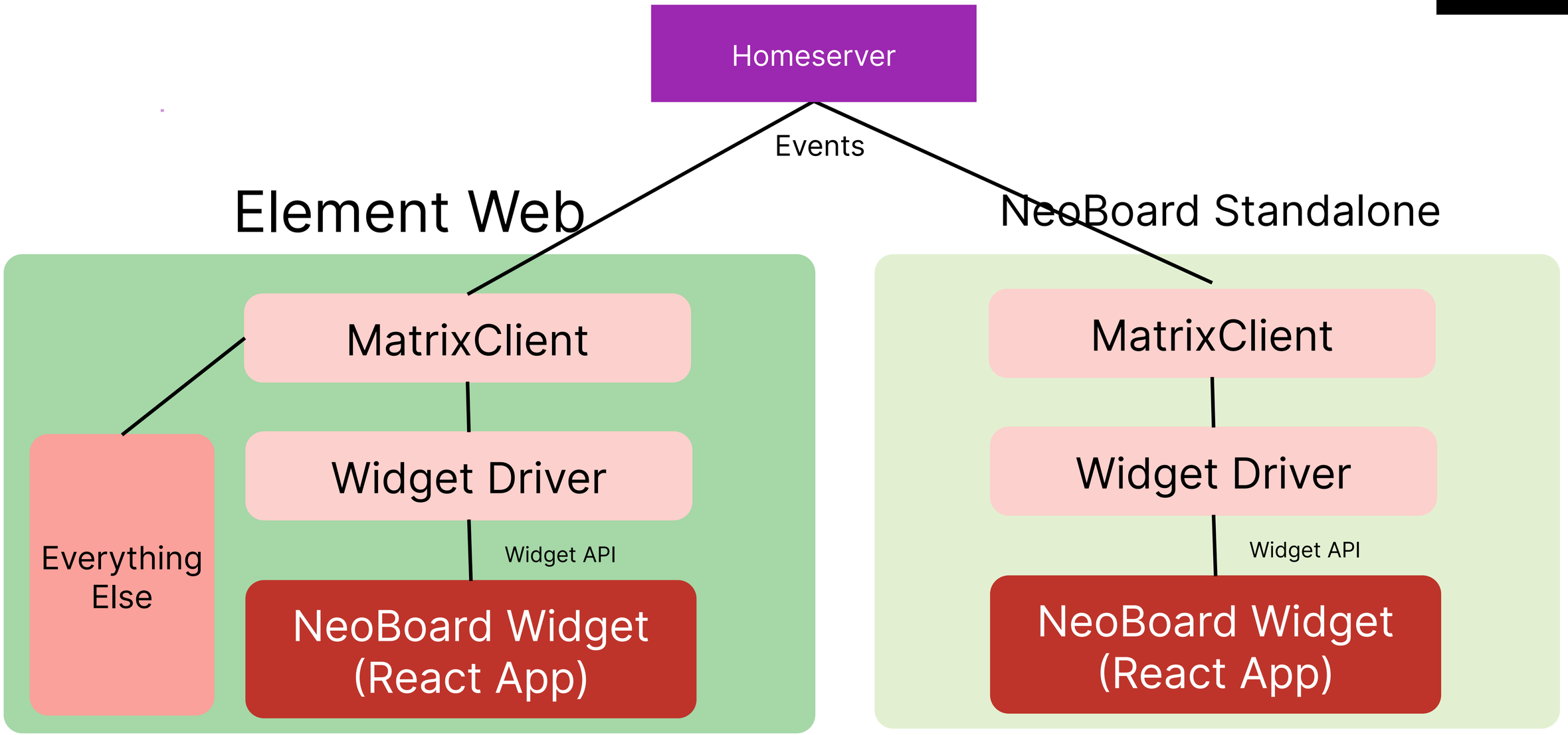
This year, our focus shifts to our flagship product: NeoBoard Standalone. This real-time collaborative whiteboard matrix client has evolved through continuous development and iteration and now it features an immersive infinite canvas that redefines what's possible for open-source productivity and creative tools.

Join us for a tour of NeoBoard's latest features and share our vision for the future of real-time collaboration, powered by the Matrix protocol.

Robert Gerbauld

Neoboard S product owner. Nordeck's product owner and product development lead for Matrix-based products.

How it uses Matrix


Homeserver
Events
Element Web
NeoBoard Standalone
MatrixClient
Widget Driver
Widget API
Everything Else
NeoBoard Widget (React App)
MatrixClient
Widget Driver
Widget API
NeoBoard Widget (React App)

the matrix conference

Room as a data container

1 Room -> 1 Whiteboard
(or more)

WhiteBoard Configuration

Custom State Events
(net.nordeck.whiteboard)

WhiteBoard Content

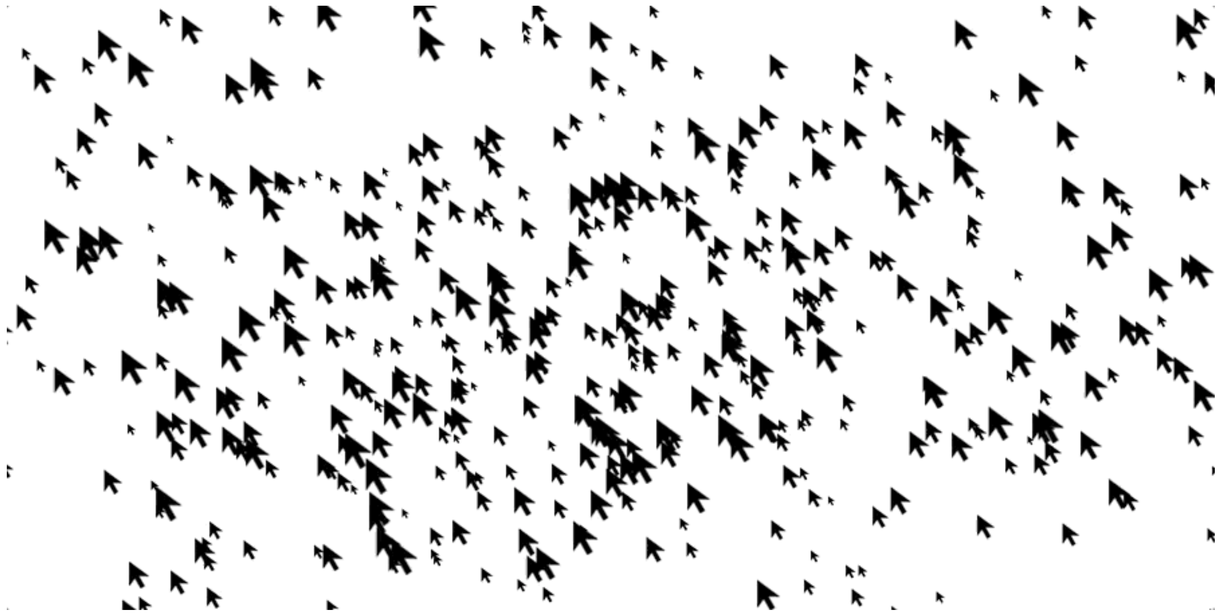
Custom Timeline Events
(net.nordeck.whiteboard.snapshot)
(net.nordeck.whiteboard.chunk)

Nordeck

```
{
  "type": "net.nordeck.whiteboard",
  "sender": "@user-id",
  "state_key": "<whiteboard-id>",
  "content": {
    "documentId": "$H1-nssrxUGbrMdKSDJcACCpmc4PrClb2WDSOrGUv6bs"
  },
  "event_id": "$event-id",
  "room_id": "!room-id",
  "origin_server_ts": 1665134498391
}
```

```
{
  "type": "net.nordeck.whiteboard.document.snapshot",
  "sender": "@user-id",
  "state_key": "<whiteboard-id>",
  "content": {
    "chunkCount": 1,
    "m.relates_to": {
      "rel_type": "m.reference",
      "event_id": "$document-create-event-id"
    }
  },
  "event_id": "$document-snapshot-event-id",
  "room_id": "!room-id",
  "origin_server_ts": 1665134498392
}
```

Why we need realtime data



Persistent data

- content snapshots and chunks

Transient data

- mouse pointers
- enter / leave presentation mode
- active slide / frame
- realtime text typing
- etc...

the matrix conference

BUT

We need low 100s ms latency (synapse avg 500ms)

There is an overhead to JSON event metadata transport and serialization (~350 bytes)

Also a real chance of having rate limiting issues on stricter policy homeservers

Nordeck

the matrix conference

Peer-to-Peer WebRTC

Nordeck

WebRTC

standard web technology for peer to peer streaming of media but also data

following Element Call / MSC3401 Native Group VoIP signalling

Discover active participants

Use custom session state event
(instead of m.call)

Own state event has all the user's device sessions
(statekey = @mxid:hs.org)

Client iterates all state events of type net.nordeck.whiteboard.sessions

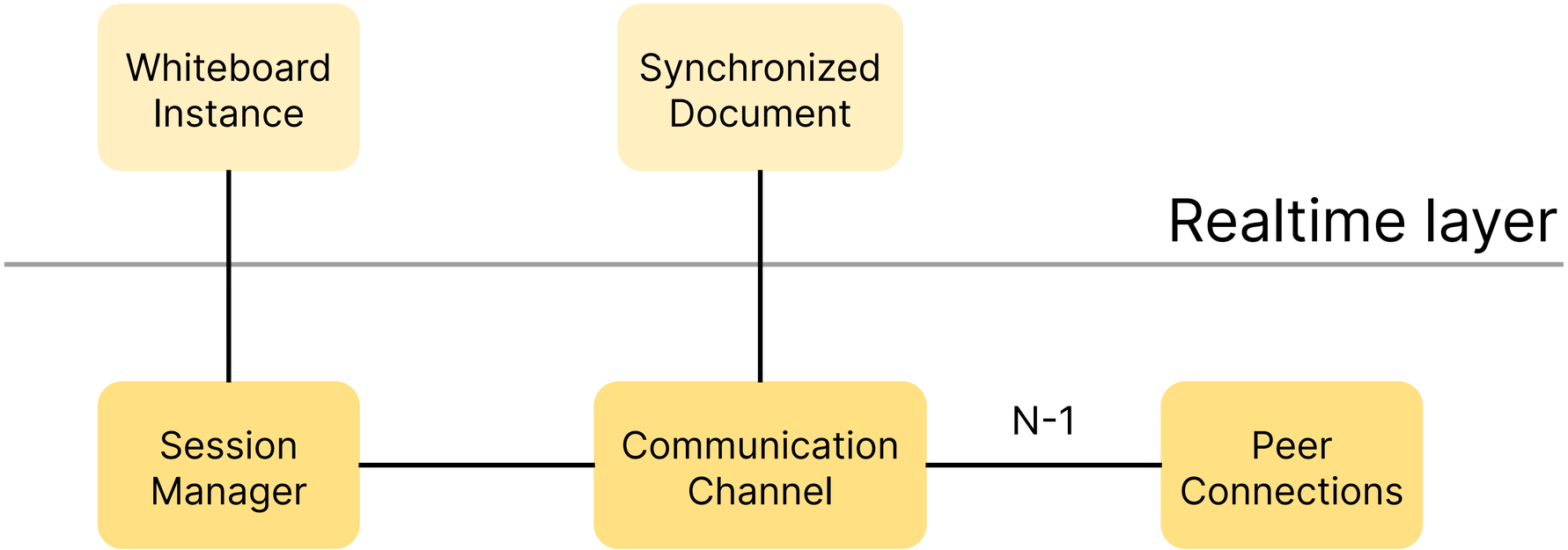
Clients remove their user's expired sessions

Establish N-1 data channels

Client gets public network routes from browser WebRTC API implementation (and TURN/STUN servers)

Shares ICE candidates with everyone in the room by sending toDevice messages
(non-persistent but reliable delivery)

If peers are reachable, data channels are established.


Whiteboard Instance
Synchronized Document
Realtime layer
Session Manager
Communication Channel
N-1
Peer Connections

the matrix conference

BUT

STUN only works for 75% of online devices (optimistic!)

TURN (relaying) - self-hosting is complex, requires considerable bandwidth and public servers have strict quota limitations

Doesn't scale, eg. Chrome has a 500 peers hard limit, and the client needs to handle N-1 data channels

Nordeck

the matrix conference

Hello MatrixRTC (and LiveKit)

Nordeck

RTC in Matrix isn't limited to audio / video calling

Element Call

NeoBoard

Third Room

MatrixRTC
(MSC4143 early 2025)

OBSOLETE

Session Membership State

Application Types with metadata

Pluggable RTC Backends

state event
m.rtc.member
+
state key
mxid_deviceid


```
{
  "type": "org.matrix.msc3401.call.member",
  "sender": "@alice:matrix.internal",
  "content": {
    "application": "net.nordeck.whiteboard",
```


full_mesh
(P2P)
+
livekit
(SFU)

What changed?

the matrix conference

Custom Session Membership State

```
{
  "type": "net.nordeck.whiteboard.sessions",
  "sender": "@user-id",
  "state_key": "@user-id",
  "content": {
    "sessions": [
      {
        "whiteboardId": "whiteboard-id",
        "sessionId": "session-id",
        "expiresTs": 1665134598391
      }
    ]
  },
  "event_id": "$event-id",
  "room_id": "!room-id",
  "origin_server_ts": 1665134498391
}
```

MatrixRTC Session Membership State

```
{
  "type": "org.matrix.msc3401.call.member",
  "sender": "@alice:matrix.internal",
  "content": {
    "application": "net.nordeck.whiteboard",
    "call_id": "whiteboard-id",
    "device_id": "SDXDZRNDJA",
    "focus_active": {
      "type": "livekit",
      "focus_selection": "oldest_membership"
    },
    "foci_preferred": [
      {
        "type": "livekit",
        "livekit_service_url": "https//livekit-jwt.matrix.internal"
      }
    ],
    "scope": "m.room",
    "expires": 1743778636001
  },
  "state_key": "_@alice:matrix.internal_SDXDZRNDJA",
  "origin_server_ts": 1743764236021,
```

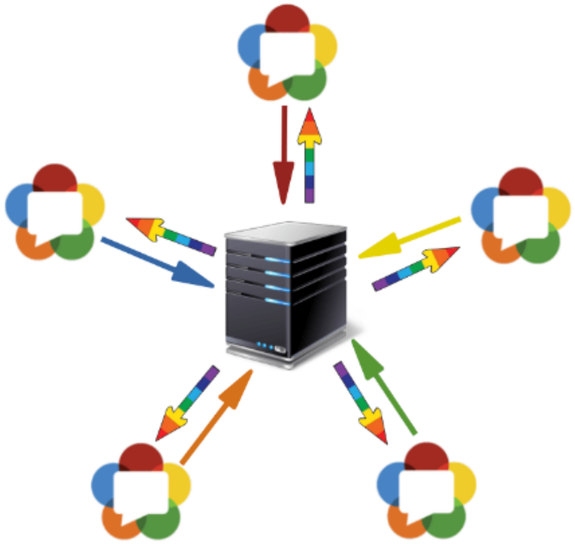
Nordeck

Transport

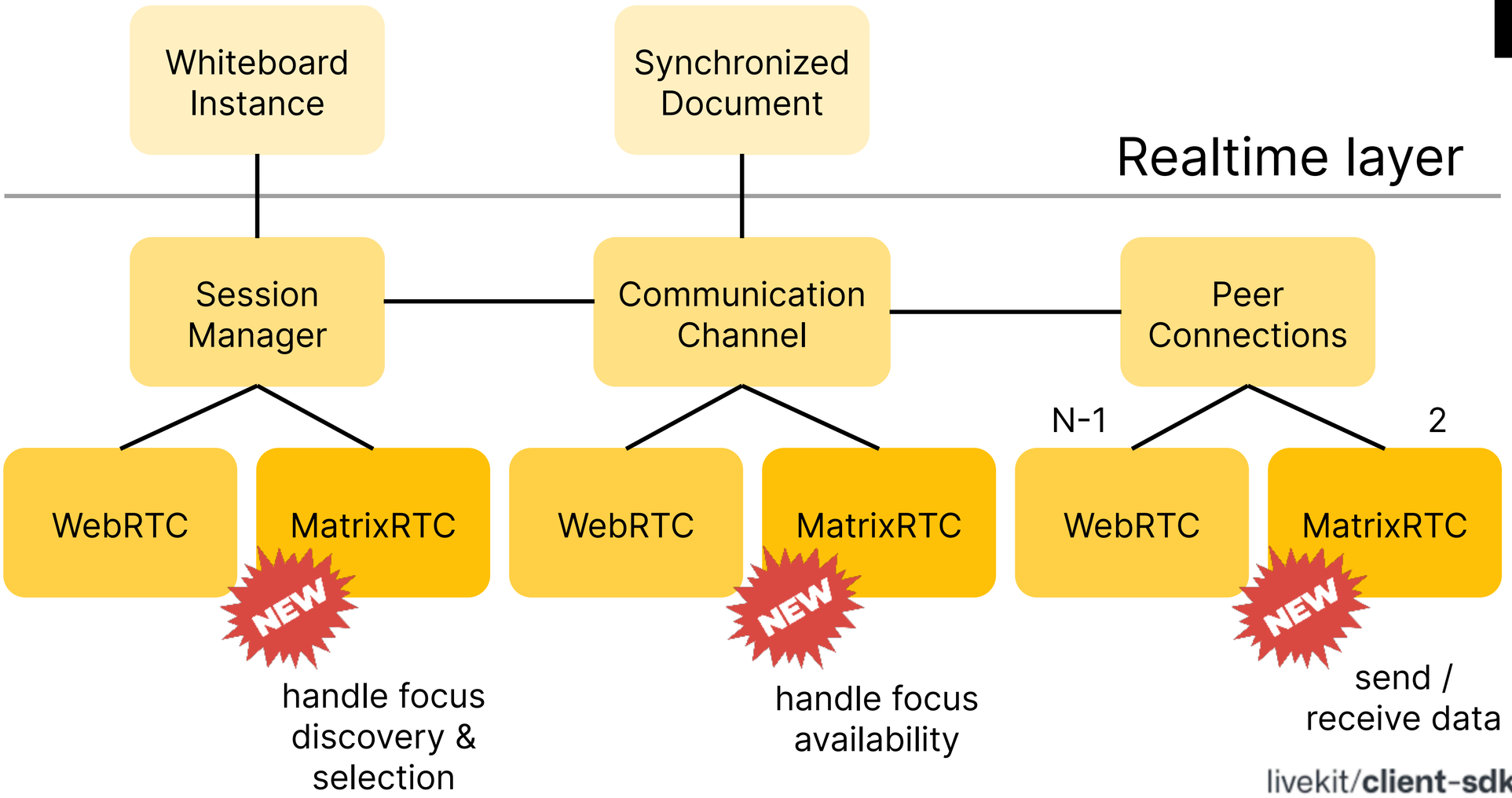
Full Mesh

N-1 WebRTC data channels
N(n-1)/2 total
quadratic growth

SFU



N-1 WebRTC data channels
(N*2) total
linear growth


Whiteboard Instance
Synchronized Document
Realtime layer
Session Manager
Communication Channel
Peer Connections
N-1
2
WebRTC
MatrixRTC
WebRTC
MatrixRTC
WebRTC
MatrixRTC
NEW
NEW
NEW
handle focus discovery & selection
handle focus availability
send / receive data
livekit/client-sdk-js

the matrix conference

Positive 😀

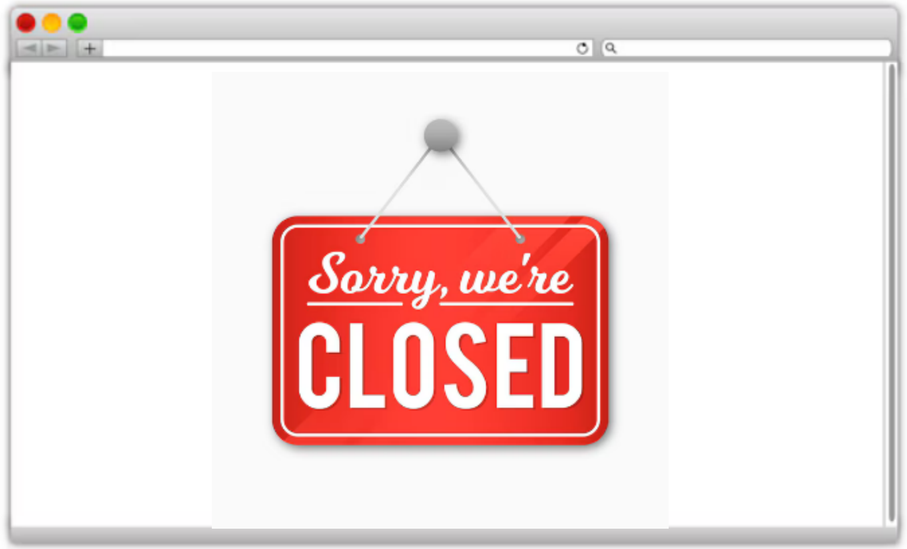
Simpler implementation
Reliable realtime connections
Greater scalability, mostly limited by backend resources

Not So Positive (but still) 🤔

More complex deployments (LiveKit + mRTC Auth Service)
Hopefully becomes a standard for matrix deployments
Not reusing the RTC Membership logic from the JS SDK

Nordeck

Hangups and other challenges



Different session semantics in NeoBoard

You just join and leave

What happens when you don't explicitly leave?

Delayed Events (MSC4140)*

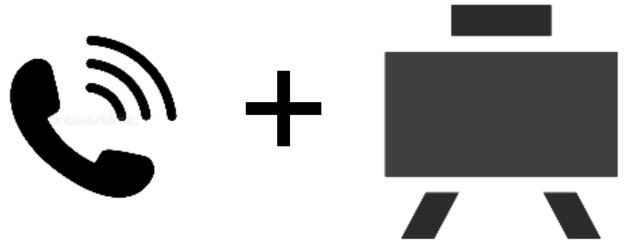
RTC membership event content is cleared, sent after joining

Timer refreshes the delay, avoiding it being sent

Timer expired? Event is sent and membership cleared, terminating session

* use carefully, may have scalability issues

Multiple RTC Apps in the same Room



OBSOLETE

Overrides of shared RTC state

Workaround to include APPID in the session membership state key

Apps can read all shared RTC state and metadata is leaked

Especially when using widgets

Ongoing work to improve this

IT WORKS!
NOW WHAT?

Next Steps

MatrixRTC Spec update this week!

Slots, Sticky Events, Multi-SFU

MSC for Whiteboard application type

WebSockets as an RTC transport

the matrix conference

Milton Moura
milton.moura@nordeck.net
@milton.moura:nordeck.io

https://neoboard.io

Nordeck