NeoBoard

Building a Real-Time Collaborative Whiteboard Widget on Matrix

#neoboard-froscon23:matrix.org
Dominik Henneke

Fullstack Software Engineer
Works with Matrix since 2021

https://github.com/dhenneke
@dhenneke:matrix.org
About the NeoBoard

• It's a whiteboard.
• Alternative to commercial SaaS tools.
• Matrix-native: embedded in your messenger.
• Uses the Matrix room as database.
• Not a drawing tool; it's a working tool.
Widget Architecture

Browser

Element

Messenger

Widget – web app

Homeserver

(Client-Server API)

(Synapse)

#neoboard-froscon23:matrix.org
Features

• Fixed sized slides
• Add shapes and lines
• Add text to shapes
• Collaborative work

#neoboard-froscon23:matrix.org
Data Storage

Updates every 5-10 seconds

- Persistence (whiteboard content) stored in room for later
Data Storage

- **WebRTC**
  - User Alice
  - User Bob
- **Matrix**
  - User Alice
  - User Bob
- **Local Storage**
  - User Alice
  - User Bob

**Real-Time updates, some ms**
- Persistence (whiteboard content)
- Temporary (cursors, ...)

**Updates every 5-10 seconds**
- Persistence (whiteboard content)
  - stored in room for later

**Stored locally, immediately**
- Whiteboard content, for offline use
  - or users leaving before data is stored

**WebRTC**
- User Alice
- User Bob

**Matrix**
- User Alice
- User Bob

**Local Storage**
- User Alice
- User Bob
WebRTC over Matrix

- Similar approach as Element Call (MSC3401).
- WebRTC signalling using toDevice messages.
- Full mesh connection.
- May switch to MatrixRTC with SFU once fully available.

#neoboard-froscon23:matrix.org
Conflict-free Replicated Data Type (CRDT)

- Different users can edit the state individually.
- Automatic (and deterministic!) conflict resolution.
- Yjs as a CRDT implementation.
- Can provide delta-updates for real-time updates.
- Our implementation is not yet reusable for us or others.

#neoboard-froscon23:matrix.org
Challenges

Solved:
• Size limit of events
• How to read all events from the rooms via the widget api?
• How to efficiently search for whiteboard content in an encrypted room?

Unsolved:
• History visibility
• Rate limiting
• We spam the room with invisible events
Event Relations & Chunking

- `net.nordeck.whiteboard` (state_key: `<whiteboard-id>`) ➔ one state event for every whiteboard in the room
- `net.nordeck.whiteboard.document.create` (event_id -> documentId) ➔ one event for each whiteboard
- `net.nordeck.whiteboard.document.snapshot` (event_id -> snapshotId) ➔ n snapshots for each whiteboard
- `net.nordeck.whiteboard.document.chunk` ➔ m chunks for each snapshot
Notable upstream contributions

- MSC3869: Read event relations with the Widget API
- Implement MSC3869
- 🚧 MSC4039: Upload files with the Widget API 🚧
- 🚧 Implement MSC4039 🚧

- Extend url template variables with device_id
- Correct the dir parameter of MSC3715
- Allow downloads from widgets

#neoboard-froscon23:matrix.org
Next Steps

- Upload images
- Improved rights and role model
- Standalone mode
- Use Content Repository for snapshot storage
- More formatting options
- Accessibility of the slide contents
- Mobile Support

…
More Info

- https://nordeck.net
- https://github.com/nordeck
- https://github.com/nordeck/matrix-neoboard

- Demo deployment (or self-host it)
- Read our architecture docs and ADRs in the docs/ folder