CoMatrix

How to overcome kernel panic, deal with buffer limitations and LOL at random

RIOT Summit 2021, 10.09.2021

Tobias Buchberger and Ines Kramer

Competence Centre for IT-Security, FH Campus Wien

The CoMatrix project was funded by netidee PROJEKTE

[matrix]

- (Messaging) services have increasingly been centralized into the hands of a few companies
- Matrix[1] is an **open standard** for **interoperable**, **decentralised**, **real-time** communication over IP.
  - Synchronizes decentralized conversation history
    - Bridges between different messaging systems[2]
  - *Matrix Foundation* provides (amongst others):
    - Specifications: **Client-Server API**[3]
    - Reference homeserver: *Synapse*[4]
    - Client: *Element*[5]

Matrix behind the Scenes

> Matrix is built on common web standards. All communication happens via HTTP calls to a RESTful API. Transport security is provided by TLS, data is exchanged in the JSON format.

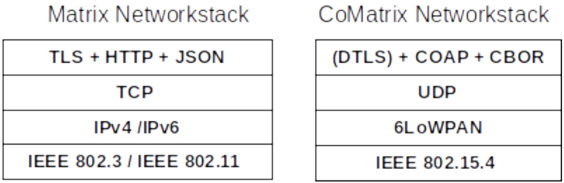
```
curl -X PUT -d ’{"msgtype":"m.text", "body":"hello"}’ ’https://alice.com
     :8448/_matrix/client/r0/rooms/<ROOM_ID>/send/m.room.message/<txnId>
      ?access_token=<ACCESS_TOKEN>’

{"event_id": "YUwRidLecu"}
```

One of the proposed use cases of Matrix is IoT, but how?

CoMatrix

- **CoMatrix** (Constrained Matrix) tries to combine the powerful semantics of Matrix with the restrictions of constrained environments. We aim to enhance Matrix by making it accessible for constrained IoT devices.



Matrix Networkstack
TLS + HTTP + JSON
TCP
IPv4 /IPv6
IEEE 802.3 / IEEE 802.11

CoMatrix Networkstack
(DTLS) + COAP + CBOR
UDP
6LoWPAN
IEEE 802.15.4

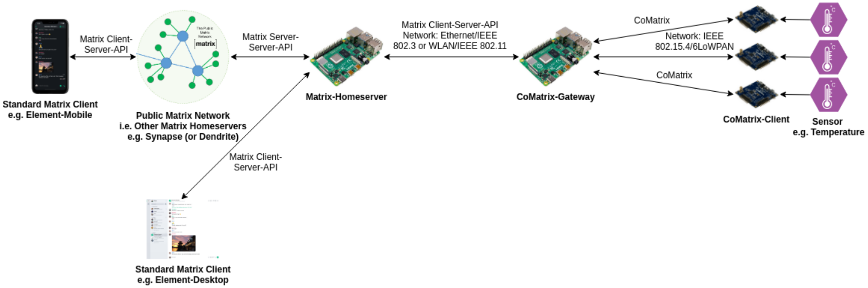
CoMatrix: Gateway and Client Library

CoMatrix consists of:

> **CoMatrix Gateway:** this gateway communicates with constrained IoT devices on one side via CoAP+CBOR and translates to the Matrix protocol on the other side (i.e. HTTP+JSON).

> **CoMatrix Client:** an external RIOT-OS module and 2 example applications. Interaction with Synapse-Matrix homeserver is enabled via the gateway.

CoMatrix Overview



Standard Matrix Client
e.g. Element-Mobile

Matrix Client-Server-API

Public Matrix Network
i.e. Other Matrix Homeservers
e.g. Synapse (or Dendrite)

Matrix Server-Server-API

Matrix-Homeserver

Matrix Client-Server-API
Network: Ethernet/IEEE 802.3 or WLAN/IEEE 802.11

CoMatrix-Gateway

CoMatrix

Network: IEEE 802.15.4/6LoWPAN

CoMatrix

CoMatrix-Client

Sensor
e.g. Temperature

Matrix Client-Server-API

Standard Matrix Client
e.g. Element-Desktop

CoMatrix Features

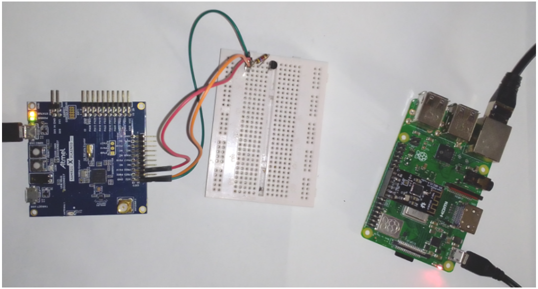
- Sending of messages to a Matrix room
- Receiving of the last message of a Matrix room
- User registration at a Matrix-Synapse HS
- Joining of a Matrix room upon invitation
- Login of a user at a Matrix-Synapse HS
- Logout of a user at a Matrix-Synapse HS

CoMatrix Gateway

- A CoAP-HTTP proxy based on Python CoAP library *aiocoap*[6]
- Provides different CoAP resources for different features (e.g. `/register`, `/send`)
- CoAP Proxy-Uri needs to contain the Matrix-Synapse homeserver URL in full or short format (cf. "MSC3079: Low Bandwidth Client-Server API"[7])
  - `http://localhost:8008/_matrix/client/r0/register` vs. `http://localhost:8008/4`
- A custom CoAP option with ID 256 is used for Synapse access tokens
- Reduces JSON response payloads to necessary information

CoMatrix Client

- External RIOT-OS Module
- 2 Example Applications
  - Chat client - for testing
  - Temperature sensor application



CoMatrix Client external RIOT-OS Module

> Dependencies:
  - generic network routing module for 802.15.4
  - gcoap, nanocoap
  - tinyCBOR package

> Kconfig - app.conf
  - Gateway IPv6 address: e.g. `fe80::1`
  - Matrix-Synapse HS: e.g. `https://192.168.1.102:8008`
  - opt Matrix room ID: e.g. `!base64string:synapse.name`
  - opt Matrix-Synapse access token
  - opt Switch: enable short-url

CoMatrix Client Implementation Experiences

- Be aware of buffer lengths, free allocated memory, number of threads called
- **Overcoming gcoap packet size limitations**
  - Matrix-Synapse token length >290 bytes BUT default gcoap packet size 128 bytes[8], increased buffer size via Kconfig (token length reduced to ca. 40 bytes since Synapse v1.34[9])
- **Everything but Random**
  - Client needs to create unique message identifier
  - **Solution:** request a timestamp from the CoMatrix gateway on initialization of SAMR21-xpro
  - **BUT** each CoAP request on initialization uses the same CoAP token because of a lack of randomness

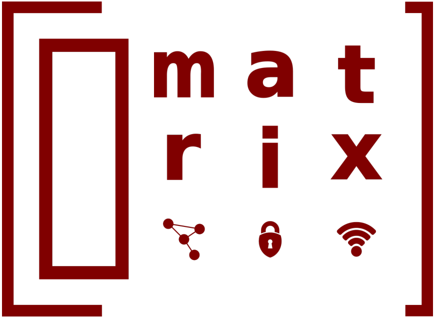
Limitations and Future Work

- **Still not able to talk in secret** - currently only plaintext communication supported, our options are
  - DTLS - aiocoap does not support DTLS on server side
  - Application layer encryption with OSCORE
  - OpenThread - full network stack which supports DTLS
- Unit tests
- Implement callback handler for Matrix errors
- Reduce bandwidth by using CBOR integer keys[10]

CoMatrix: Links and Contact Information

- Website: https://comatrix.eu
- Code repo: https://gitlab.com/comatrix/comatrix
- Contact:
  - Tobias Buchberger: tobias.buchberger@fh-campuswien.ac.at, @tobi_fh:matrix.org / IdleRPG @ #riot-os:matrix.org
  - Ines Kramer: ines.kramer@fh-campuswien.ac.at
- Featured by ELVIS Lab (Embedded Lab Vienna for IoT and Security): https://elvis.science/

CoMatrix was funded by *Netidee*.
https://netidee.at



Thank you for your attention!
Any questions?

References

[1] https://matrix.org/

[2] https://matrix.org/bridges/

[3] https://matrix.org/docs/spec/client_server/r0.6.1

[4] https://github.com/matrix-org/synapse

[5] Formerly called *RIOT* ;-), https://github.com/vector-im

[6] https://github.com/chrysn/aiocoap

[7] https://github.com/matrix-org/matrix-doc/blob/3924ba53ab3a95aa84341
proposals/3079-low-bandwidth-csapi.md#appendix-b-coap-path-enums

[8] https://github.com/RIOT-OS/RIOT/pull/16377

[9] https://matrix.org/blog/2021/05/17/synapse-1-34-0-released#new-acce

[10] https://github.com/matrix-org/matrix-doc/blob/3924ba53ab3a95aa84341
proposals/3079-low-bandwidth-csapi.md#appendix-a-cbor-integer-keys