Matrix: Otevřený standard pro bezpečnou komunikaci

Marian Rychtecký
CSNOG 2026

20-21. 01. 2026

NIX·CZ

Matrix – zrychlená verze

- Víte, jak funguje SMTP?
- „Pojďme to šifrovat.“ a vzniklo SMTPS
- „Pojďme šifrovat i obsah.“ a vzniklo např. PGP

- Matrix je ten samý příběh
  - Jen v reálném čase.
  - Federovaně.
  - Bez ruční výměny klíčů.

Děkuji za pozornost.

NIX·CZ

Proč se o Matrix vůbec bavit?

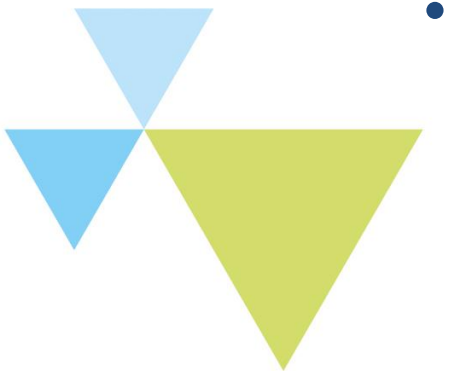
- Centralizované messengery = single point of failure
- Metadata jsou často cennější než obsah
- Vendor lock-in & compliance
- Potřeba otevřeného, federovaného, E2EE řešení

matrix

NIX·CZ

Proč se o Matrix vůbec bavit?

- Kritická infrastruktura:
  - provozní chat
  - incident response
  - interní koordinace
- Požadavek:
  - otevřený protokol
  - federace
  - E2EE mimo server



Stejně jako neradi jeden globální route-server, máme neradi jeden chat server.

NIX·CZ

Co je Matrix?

- Otevřený federovaný komunikační protokol
- Decentralizace: vlastní server
- Real-time + async (chat, VoIP, IoT)
- Otevřená specifikace + open-source implementace
- JSON-based komunikace nad HTTPS

Architektura Matrixu

- **Homeserver** – ukládá a přeposílá **eventy**
- **Client** – UI + kryptografie
- **Federace** – server  server (HTTPS + signing)
- **Event** – založen na JSON struktuře

Eventy


```
{
  "type": "m.room.encrypted",
  "sender": "xz:matrix.abc.cz",
  "content": {
    "algorithm": "m.megolm.v1.aes-sha2",
    "ciphertext": "AwgVEqABFcNUJ4xiLuD.....Ag",
    "device_id": "GMFHYKTRAV",
    "sender_key": "Y6W0HkL7FSGRKU5oIhCwA6V4xNzf2VsuYB",
    "session_id": "7P3gQhX2a6w5MIsjO4TnH9k+8sPdlRtbh"
  },
  "origin_server_ts": 1768651682846,
  "unsigned": {
    "membership": "join",
    "age": 68
  },
  "event_id": "$D6Y9kH7l0s3VGt2LZME4yP-a8oKfGd1V9H0Y3lT",
  "room_id": "!cz:matrix.abc.az"
}
```



```
{
  "content": {
    "body": "Hura uz je lip!",
    "m.mentions": {},
    "msgtype": "m.text"
  },
  "event_id": "$7-VLkD79......6Hl0GGy3d0s",
  "origin_server_ts": 1768651682846,
  "room_id": "!ABC:matrix.abc.cz",
  "sender": "@xz:matrix.abc.cz",
  "type": "m.room.message",
  "unsigned": {}
}
```



```
{
"type": "m.room.message",
"content": {
  "body": "image.png",
  "info": {
    "size": 255943,
    "mimetype": "image/png",
    "w": 680,
    "h": 371,
    "xyz.amorgan.blurhash": "LROz6-3-#r#DASxS|",
    "org.matrix.msc4230.is_animated": false
  },
  "msgtype": "m.image",
  "m.mentions": {},
  "file": {
    "v": "v2",
    "key": {
      "alg": "A256CTR",
      "ext": true,
      "k": "Nwto4MRpKvoJGdQKLyBCn_VFyc8qAU",
      "key_ops": [
        "encrypt",
        "decrypt"
      ],
      "kty": "oct"
    },
    "iv": "OEmDAiKnHUoAAAAAAAAAAA",
    "hashes": {
      "sha256": "0BAtdOFzd/lSNddaozmqQbwo7Y"
    },
    "url": "mxc://matrix.abc.cz/sPORMlTKMs"
  }
}
```


NIX·CZ

Architektura Matrixu

- **Specifikace**
  - Definuje:
    - JSON struktury
    - REST endpointy
    - stavový model místností
  - Vendor-neutral
  - Interoperabilní
- **Implementace**
  - Konkrétní software:
    - server: Synapse
    - klient: ElementX , FluffyChat

https://matrix.org/ecosystem/clients/

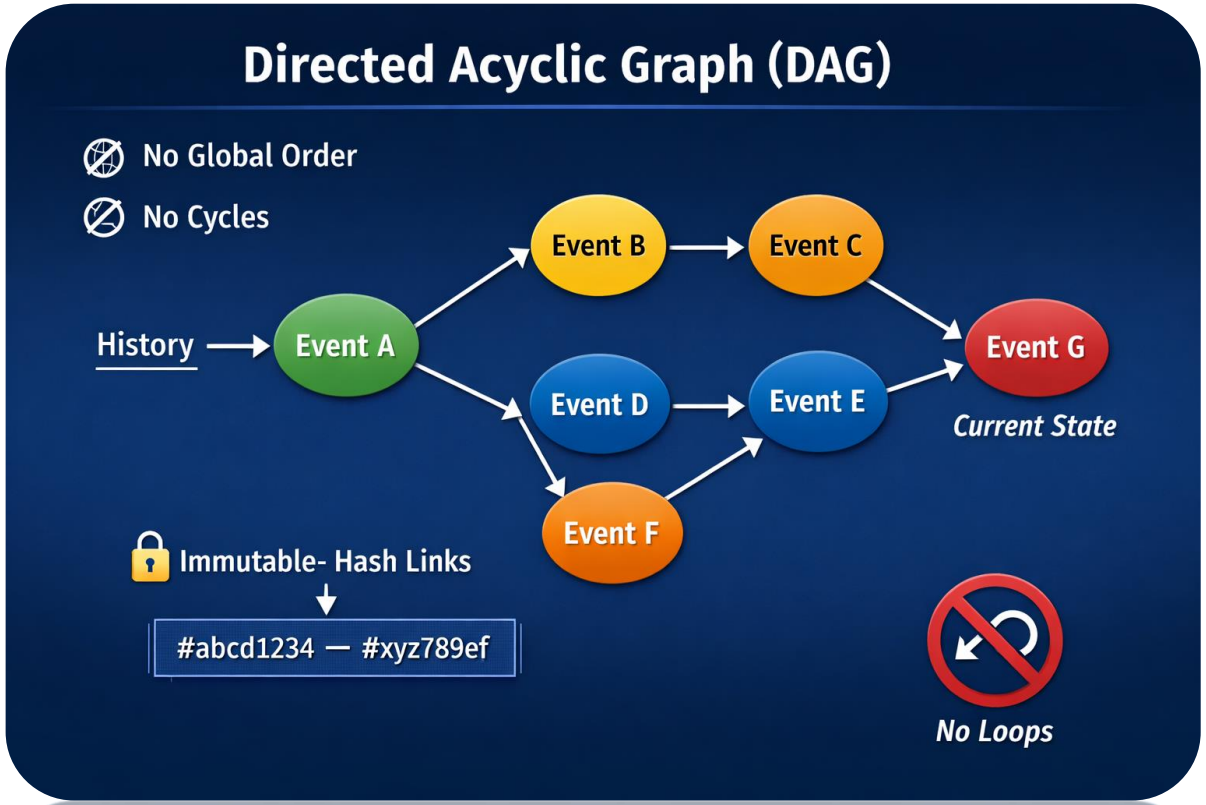
Transportní a aplikační vrstva

- Transport
  - TCP/443
  - HTTPS
- Aplikační vrstva:
  - REST (/sync, /send, /keys/query)
  - JSON payload
- Vyřešeno
  - Firewally
  - Proxy

BGP over TLS, ale pro messaging.

Event model & state resolution

- Vše je jako immutable event
- Eventy tvoří Directed Acyclic Graph (DAG) – „větvený blockchain“
- Stav místnosti je deterministický výpočet z eventů
- Replikace přes federaci mezi servery



Routing table = funkce nad BGP UPDATEs.

NIX·CZ

Identity & zařízení

- Uživatel má více zařízení
- Každé zařízení má:
  - vlastní identity keypair - Curve25519 (key agreement)
  - vlastní kryptografický stav - Ed25519 (signing)
- Ztráta zařízení neznamená ztrátu identity (recovery)

Olm - 1:1 kryptografie

- Použití pro
  - DM
  - distribuce klíčů
- Vlastnosti Olm
  - double ratchet
  - forward secrecy
  - post-compromise security

Olm ≈ TLS session mezi dvěma endpointy.

Olm - ratcheting

- Jak probíhá Olm ratchet:
  - Každá zpráva má ID
    - posune symetrický klíč (nová výměna klíče pomcí DH)
  - Každý směr má vlastní chain key
  - Při kompromitaci:
    - staré zprávy zůstávají bezpečné
    - nové se časem znovu zabezpečí

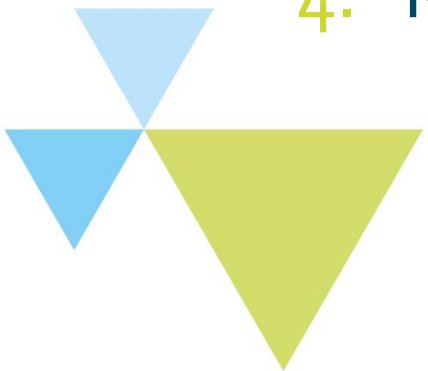
Megolm - skupinové šifrování

- V místnosti 1:N je nutné použít Megolm
  - Olm by musel používat N^2 - 1 klíčů
- Jeden klíč ⟶ mnoho příjemců
- Optimalizován pro velké místnosti a nízký overhead
- Používá symetrický ratchet (jednosměrný)

Megolm – kdo je generuje

- Každé odesílající zařízení
- Paralelně existuje více Megolm sessions

1. Odesílatel vytvoří „**Megolm outbound session**“
2. Klíč je zabalen a poslán přes **Olm session**
3. Event „`m.room_key`“ obsahuje session key + index
4. Receiver uloží jako inbound session



Control-plane (Olm) distribuuje data-plane klíče (Megolm).

Megolm – rotace a bezpečnost

- Rotace klíčů při
  - join nebo leave
  - čas nebo počet zpráv
- Odebraní členové nemůžou číst nové zprávy
  (pokud jim někdo přepošle nový klíč)
- Noví členové nemůžou číst staré zprávy
  (pokud jim někdo přepošle původní klíč)

Key rollover po změně topologie.

Instalace a monitoring

- Synapse server – Python balík + konfigurace v YAML
  - 1h včetně čtení dokumentace
  - povolení uživatelů – auto registrace / LDAP / OAUTH2
  - federace serverů – TCP/8448 (HTTPs)

Konfigurace Caddy

```
matrix.abc.cz {
  encode zstd gzip
  reverse_proxy /_matrix/* localhost:8008
  reverse_proxy /_synapse/client/*
localhost:8008
}
matrix.abc.cz:8448 {
  encode zstd gzip
  reverse_proxy /_matrix/* localhost:8008
}
```

```
abc.cz {
  @matrix_server path /.well-
known/matrix/server
    handle @matrix_server {
    header Content-Type application/json
    header Access-Control-Allow-Origin *
    respond
`{"m.server":"matrix.abc.cz:443"}` 200
    }
}
```

mr@abc.cz -> mr@matrix.abc.cz

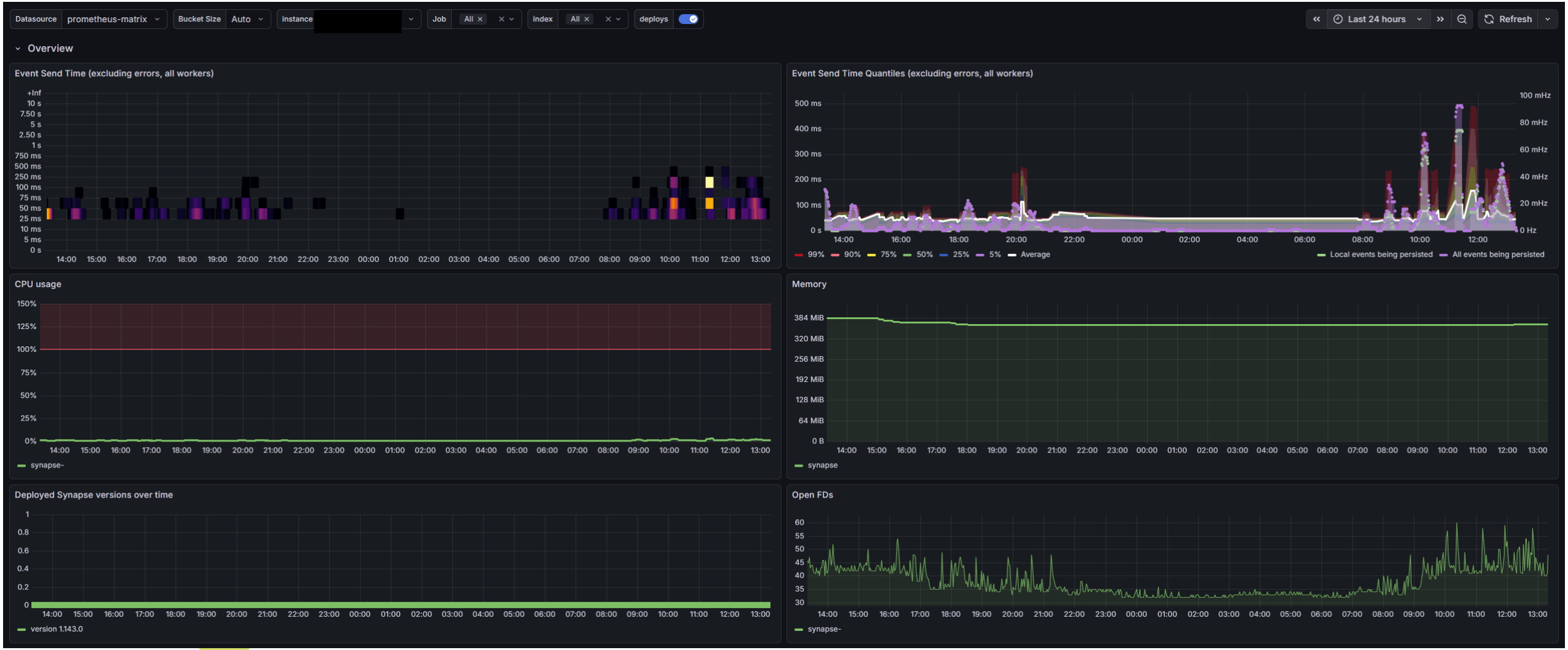
NIX·CZ

Konfigurace Synapse (pro testy)

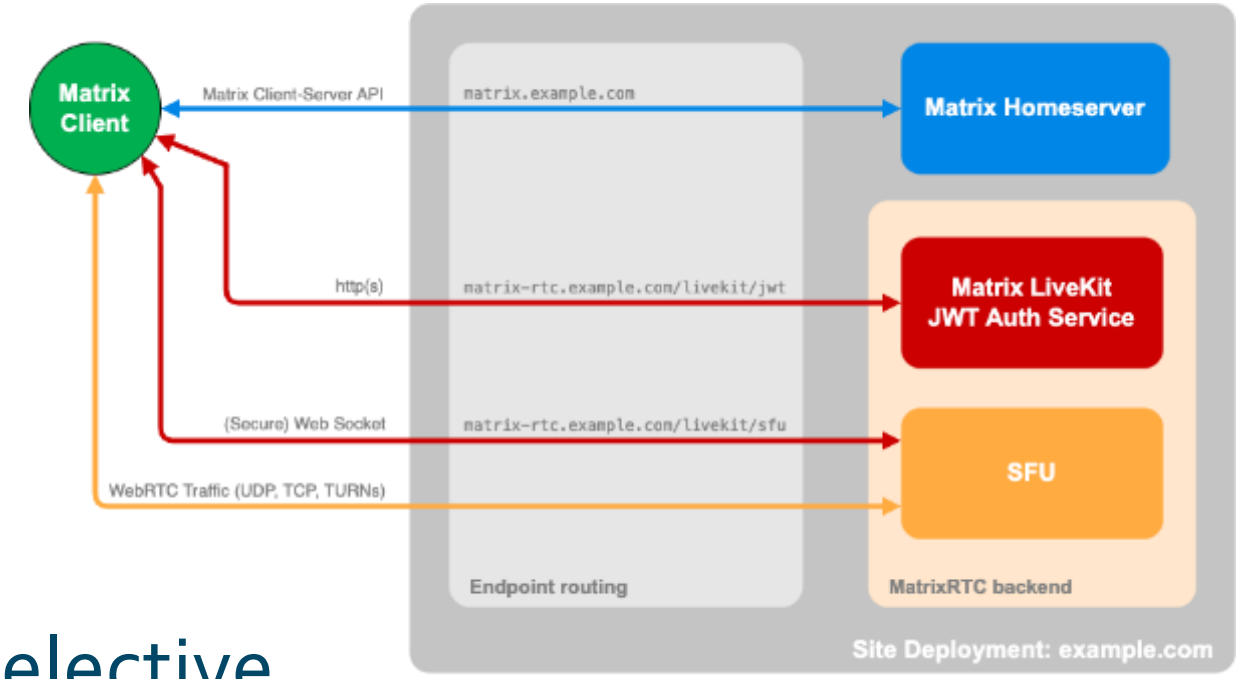
```
listeners:
  - port: 8008
    tls: false
    type: http
    x_forwarded: true
    bind_addresses: ['::1', '127.0.0.1']
    resources:
      - names: [client, federation]
        compress: false

database:
  name: sqlite3
  args:
    database: /var/lib/matrix-synapse/myhomeserver.db

media_store_path: /var/lib/matrix-synapse/media

signing_key_path: "/etc/matrix-synapse/myhomeserver.signing.key"
```

NIX·CZ

Monitoring


Datasource
prometheus-matrix
Bucket Size
Auto
instance
Job
All
index
All
deploys
Last 24 hours
Refresh
Overview
Event Send Time (excluding errors, all workers)
Event Send Time Quantiles (excluding errors, all workers)
99%
90%
75%
50%
25%
5%
Average
Local events being persisted
All events being persisted
CPU usage
synapse-
Memory
synapse
Deployed Synapse versions over time
version 1.143.0
Open FDs
synapse-

Video RTC



- JWT server pro tokeny
- LiveKit – server WebRTC Selective Forwarding Unit (SFU)
- TURNs / STUN server

NIX·CZ

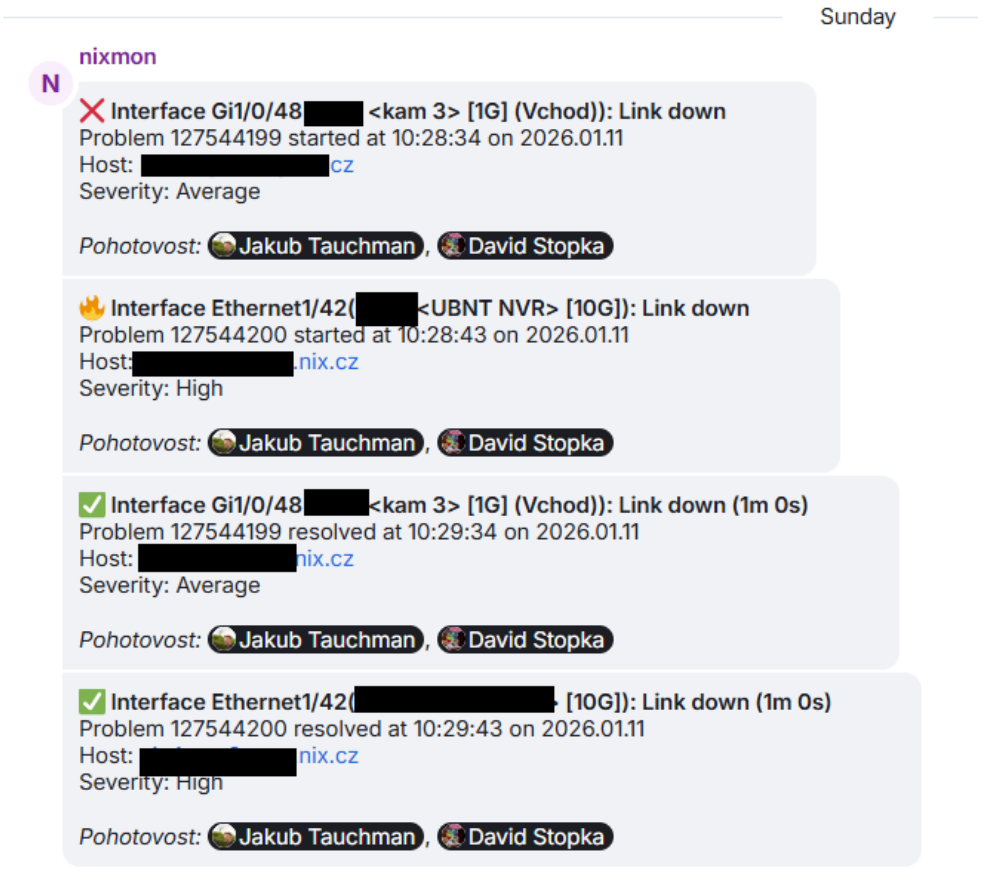
Credit: https://willlewis.co.uk/blog/posts/deploy-element-call-backend-with-synapse-and-docker-compose/

Video RTC

- Vlastnosti WebRTC
  - RTP stream „každý s každým“
  - lze mutnout lokálně „per peer“
  - lze sdílet mnoho video streamů / ploch zároveň

NIX.CZ use case

- Monitoring sítě / zařízení
- 1:1 chat
- Chat místnosti
- Video / audio volání



NIX·CZ

NIX.CZ lessons learned

- Vývoj jde rychle dopředu
- Nativní klienti pro různé OS i platformy
- Nejdále Element-X
  - Element-X klient má mouchy, ale vývoj jde rychle dopředu
- Video / audio má lepší kvalitu
- Problém s mazáním zpráv (lze vyřešit na serveru)

NIX.CZ lessons learned

- Decentralizace má své nevýhody -> federace
  - stovky až tisíce HTTPS spojení
  - geo/firewall vysoké odezvy a timeouty
  - join do místnosti trvá výrazně déle
  - server load

Děkuji za pozornost

mr:nix.cz

mr@nix.cz

NIX·CZ