Building an IoT Platform with Matrix

matthew@matrix.org
http://www.matrix.org
What is Matrix?
An open decentralised conversation store and message bus.
Why?
To create a global communication meta-network that bridges all the existing silos & liberates our communication to be controlled only by us.
Matrix is for:

- Group Chat (and 1:1)
- WebRTC Signalling
- Bridging Comms Silos
- Internet of Things Data

...and anything else which needs to pubsub persistent data to the world.
Matrix Architecture

- Clients
- Home Servers
- Application Servers
- Identity Servers
The Matrix Ecosystem

- Matrix Web Console
- Matrix iOS Console
- MatrixKit (iOS)
- Matrix Android Console
- Matrix iOS SDK
- Matrix React SDK
- Matrix Angular SDK
- Matrix JS SDK
- Matrix iOS Web Console
- Matrix React Web Console
- Synapse (Reference Matrix Server)
- Matrix Application Services and Bridges
- Other Clients
- Other Servers and Services
How does it work?

https://matrix.org/#about
Internet of Things

• Lots and lots of vendor silos.
• Lots of data that can be liberated into Matrix – both for control and telemetry.
• Matrix provides a trivial standard HTTP+JSON API for publishing or subscribing to data
• Supports arbitrary alternative transports and encodings too (CoAP+CBOR, MQTT, WebSockets etc)
Let’s bust some silos!
Matrix to IOT...

Parrot Bebop Drone

Janus WebRTC Gateway (from MeetEcho)

Web Matrix example client (Firefox)

synapse (on laptop)

classified

matrix
Janus Architecture

- `janus_audiobridge.c`
- `janus_echotest.c`
- `janus_recordplay.c`
- `janus_sip.c`
- `janus_streaming.c`
- `janus_matrix.c`
- `janus_videocall.c`
- `janus_videoroom.c`
- `janus_voicemail.c`
Ryan Rix’s IoT exploits

- Lightrix: https://youtu.be/4YG9Fk5aP24
- http://rix.si/blog/2015/11/22/body-computing-system-continuations
A random CoAP example

echo '{"msgtype":"m.text", "body":"hello"}' | perl -MCBOR::XS -MJSON -pe '$_=encode_cbor decode_json' | coap-client -m post \ coaps://alice.com/_m/c/a/v1/r/ROOM_ID/s/m.room.message?a=ACCESS_TOKEN

is the same as...

curl -XPOST -d '{"msgtype":"m.text", "body":"hello"}' "https://alice.com:8448/_matrix/client/api/v1/rooms/ROOM_ID/send/m.room.message?access_token=ACCESS_TOKEN"
End to End Encryption with Olm

- Apache License C++11 implementation of an Axolotl-style ratchet, exposing a C API.
- Axolotl is Open Whisper System's better-than-OTR cryptographic ratchet, as used by TextSecure, Pond, WhatsApp etc.
- Supports encrypted asynchronous group communication.
- 130KB x86-64 .so, or 208KB of asm.js
- https://matrix.org/git/olm
The end goal:

IOT Vendor 1

New services!

IOT Vendor 2

New platforms!

New services!
What’s left?

• Rolling out End-to-end Encryption
• Building more bridges
• Group ACLs
• File tagging and management
• Decentralised identity
• Abuse mitigation
We need help!!
• We need people to try running their own servers and join the federation.
• We need people to run gateways to their existing services
• We need feedback on the APIs.
• Consider native Matrix support for new apps
• Follow @matrixdotorg and spread the word!
Thank you!

matthew@matrix.org

http://matrix.org

@matrixdotorg