[matrix]

An end-to-end encrypted ecosystem for open decentralised communication

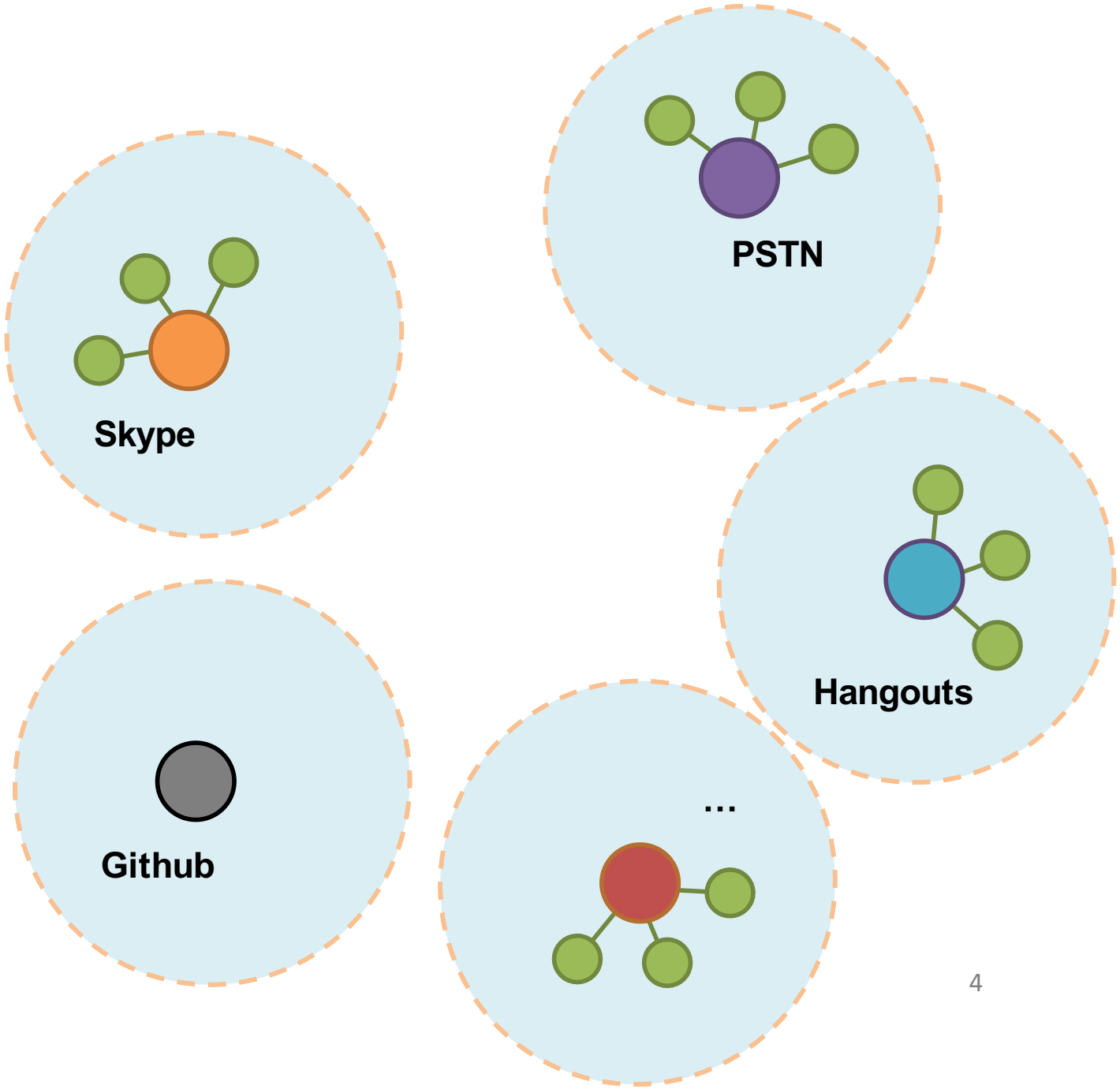
matthew@matrix.org
http://www.matrix.org

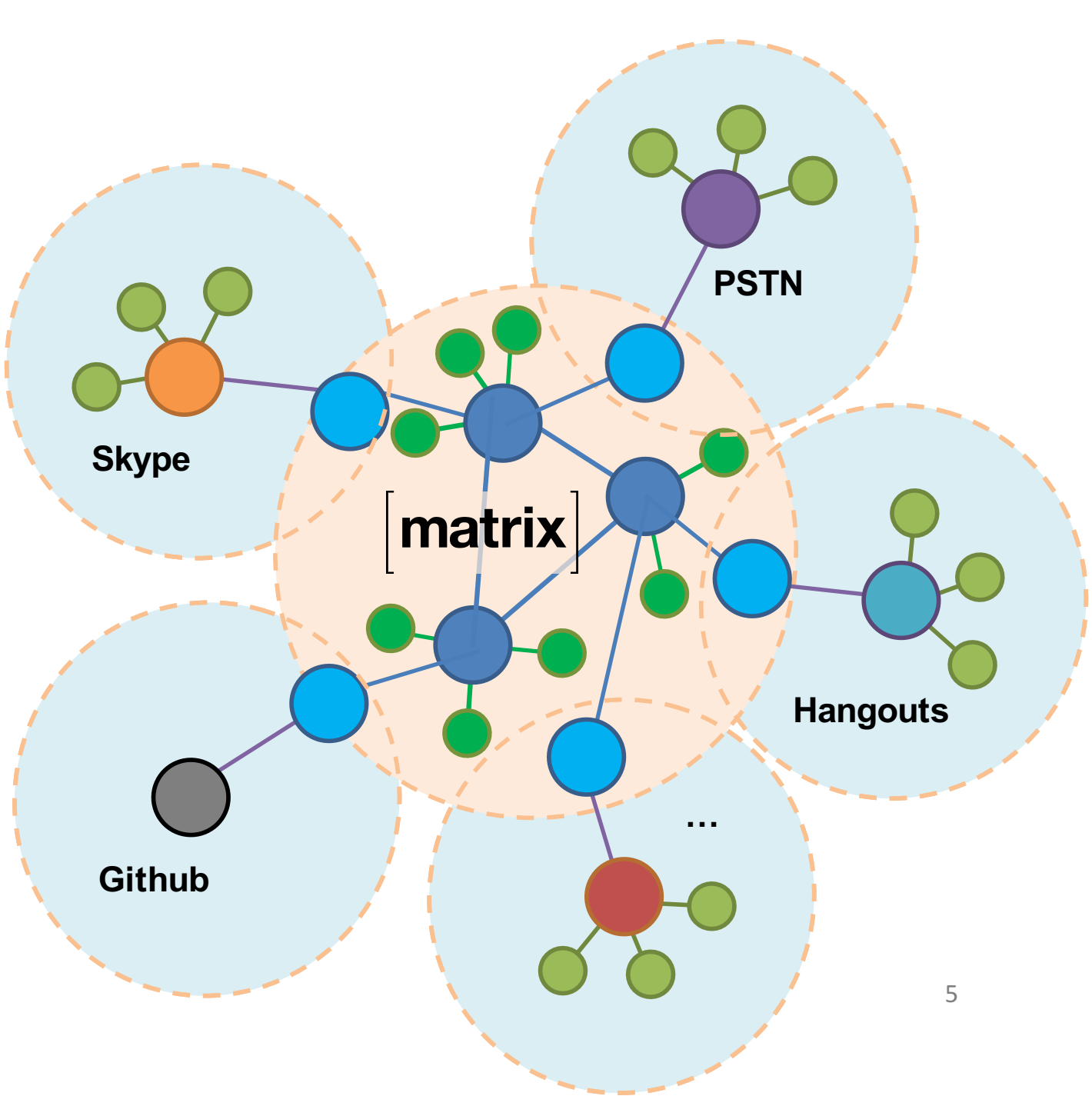
[matrix]

A non-profit open standard for defragmenting communication

[matrix]

To create a global encrypted communication meta-network that bridges all the existing silos & liberates our communication to be controlled only by us.


PSTN
Skype
Hangouts
Github
...


[matrix]
PSTN
Skype
[matrix]
Hangouts
Github
...

No single party own your conversations – they are shared over all participants.

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

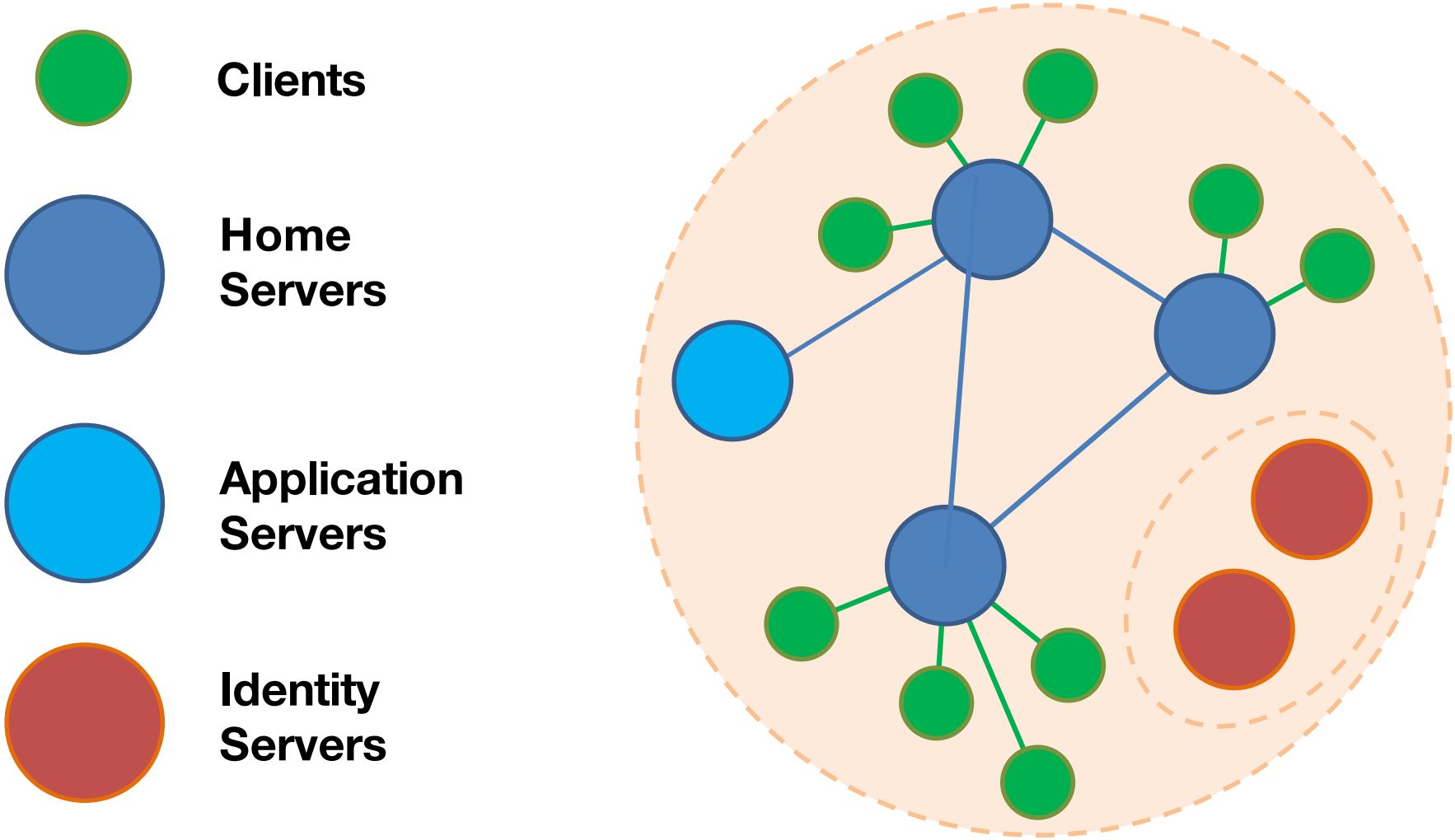
Why are you re-inventing XMPP!?!?

WE ARE NOT.

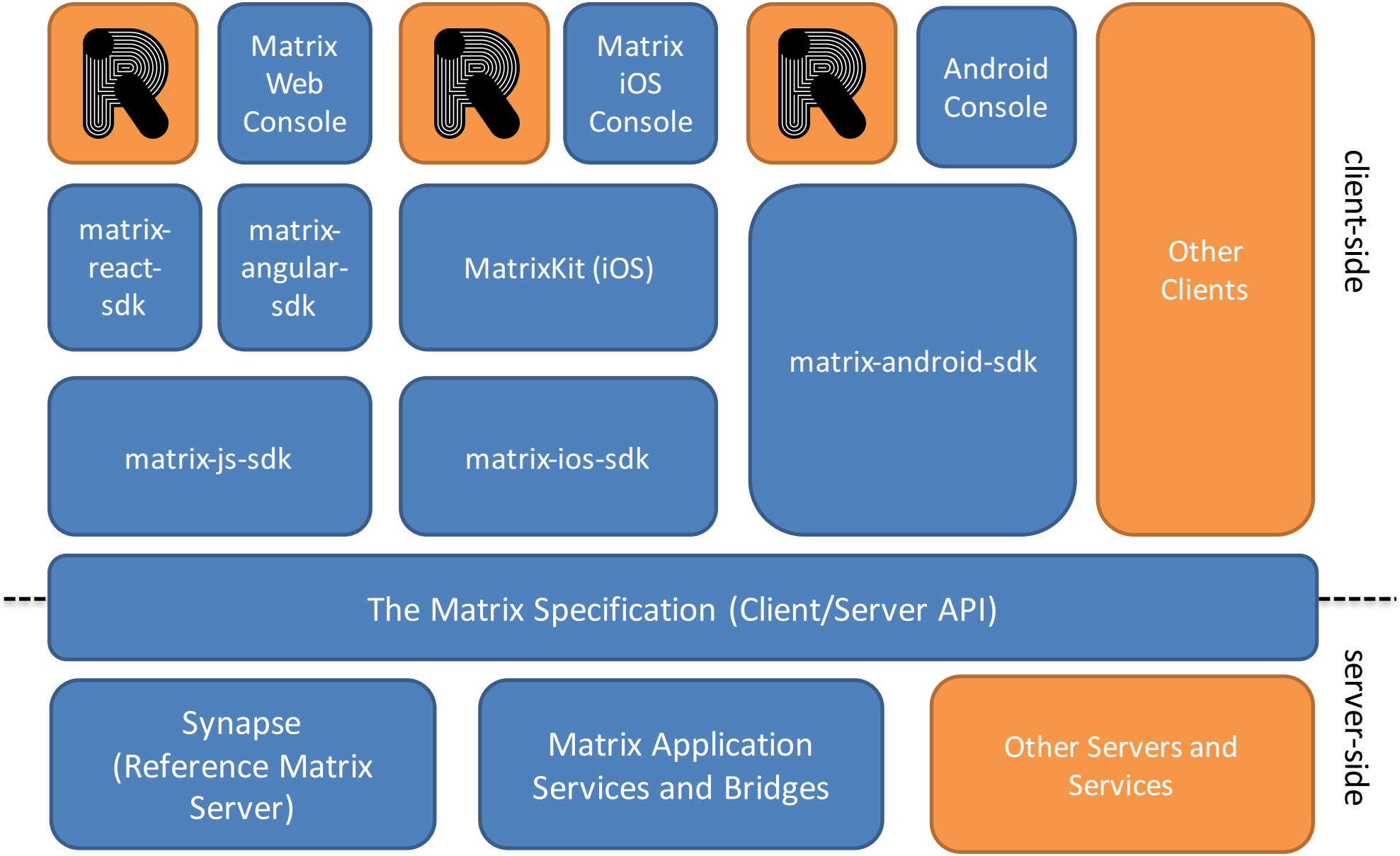
Why not XMPP?

- **Completely** different philosophy & architecture:
  - A single, monolithic, consistent, spec.
  - Different primitives:
    - Syncing decentralised conversation history (not message passing / pubsub)
    - Group conversation as a first class citizen
    - E2E crypto as a first class citizen (beta)
  - HTTP+JSON as the baseline API **(but you can use other transports too!)**
  - Core focus on defragmentation and bridging (hence the name “matrix”).

Matrix Architecture



The Matrix Ecosystem



matrix

What do you get in the spec?

- Decentralised conversation history (timeline and key-value stores)
- Group Messaging
- End-to-end Encryption **(new!)**
- VoIP signalling for WebRTC
- Server-side push notification rules
- Server-side search
- Read receipts, Typing Notifs, Presence
- Synchronised read state and unread counts
- Decentralised content repository
- “Account data” for users per room

How does it work?

https://matrix.org/#about

Clients

- >30 matrix clients (that we know about)
  - Ranging from text UIs (**Weechat**, Emacs(!))
  - …to desktop apps (**Quaternion**, NaChat, Pidgin)
  - …to glossy web and mobile clients (**Riot**)
  - …to protocol proxies (**matrix-ircd**)
- Over 15 client-side SDKs:
  - Official: JS, React, iOS, Android
  - Semi-official: Python, Perl5, Go
  - Community: Erlang, Ruby, Lisp, Elixir, Haskell, Rust…

matrix

Home servers

- **Synapse**: the original reference Matrix home server implementation.
  - 50K lines of Python/Twisted.
  - Some perf and maintainability challenges…
- **Ruma**: Community project Rust implementation… early but promising!
- **Dendron**: skeleton Golang reference impl
  - Wraps synapses, incrementally migrating endpoints
- BulletTime (Go), Pallium (Go), jSynapse (Java) experiments from the community

What does it look like?

https://riot.im

The client-server API

To send a message:

```
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"
```

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

The client-server API

To set up a WebRTC call:

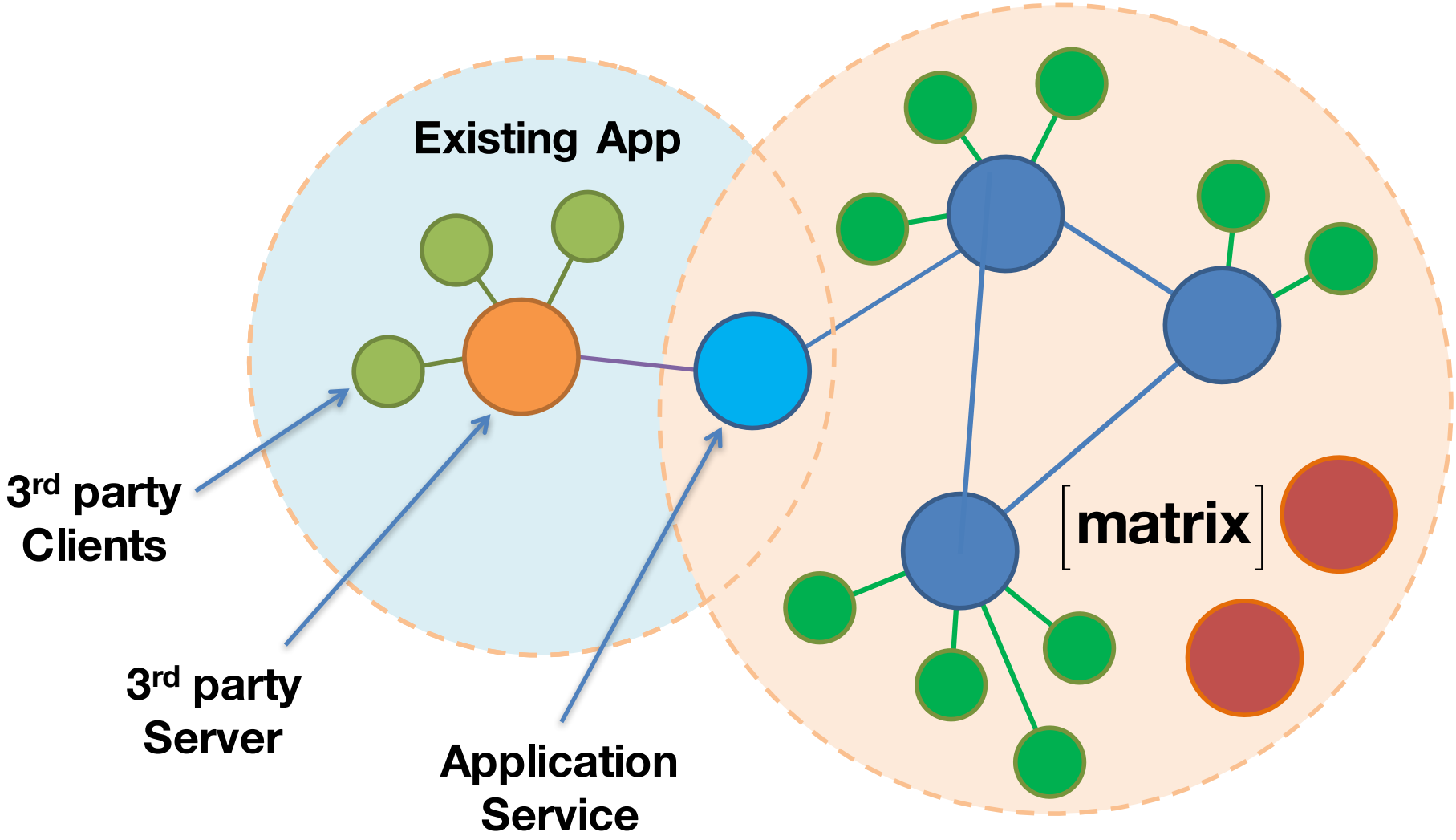
```
curl -XPOST -d '{\
  "version": 0, \
  "call_id": "12345”, \
  "offer": {
    "type" : "offer”,
    "sdp" : "v=0\r\no=- 658458 2 IN IP4 127.0.0.1…"
  }
}'
"https://alice.com:8448/_matrix/client/api/v1/rooms/ROOM_
ID/send/m.call.invite?access_token=ACCESS_TOKEN"

{ "event_id": "ZruiCZBu” }
```

Basic 1:1 VoIP Matrix Signalling

```
    Caller                          Callee
m.call.invite ------------>
m.call.candidate --------->
[more candidates events]
                                User answers call
                   <------ m.call.answer
                [media flows]
                   <------ m.call.hangup
```

Bridges and Integrations

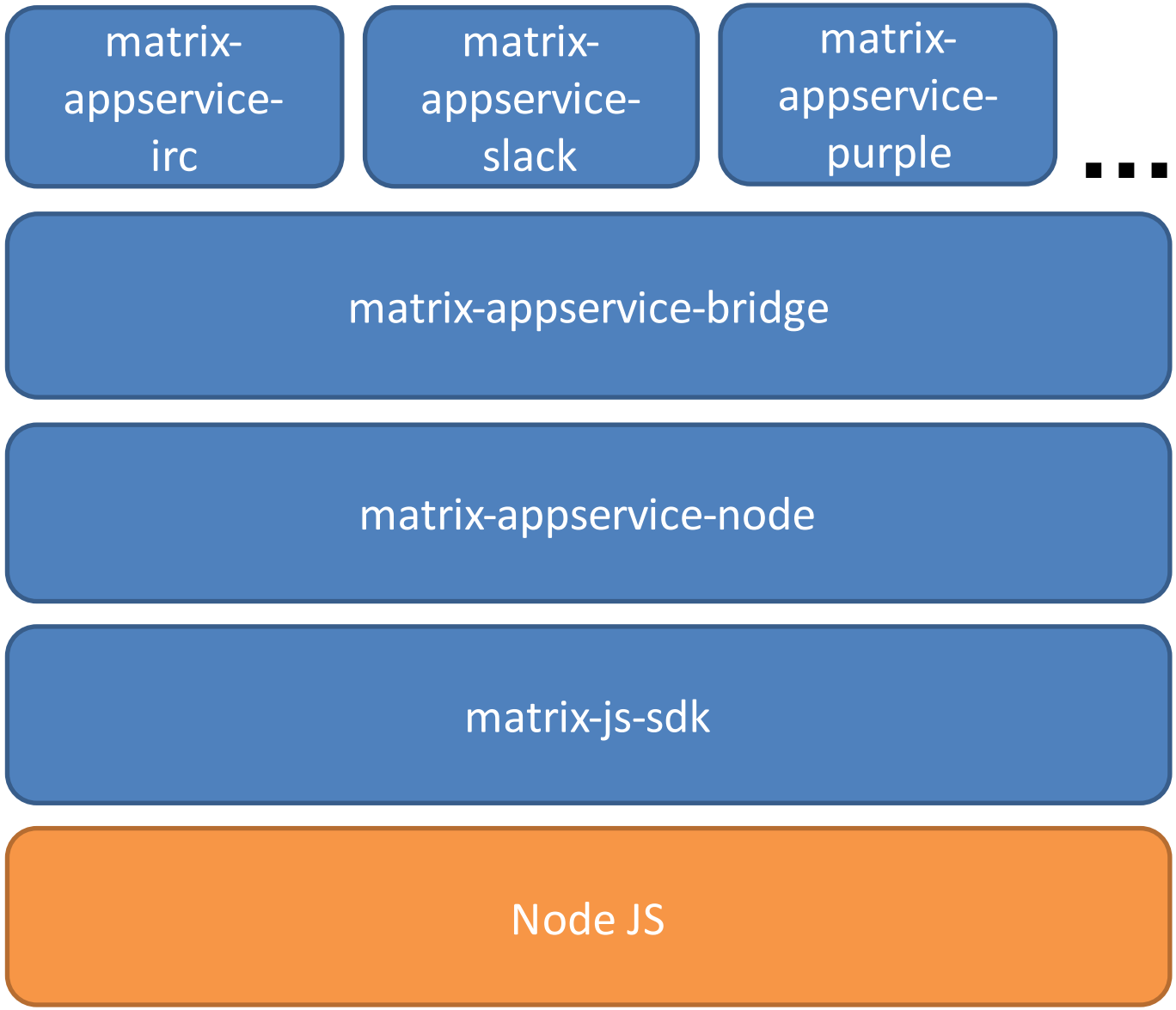


matrix

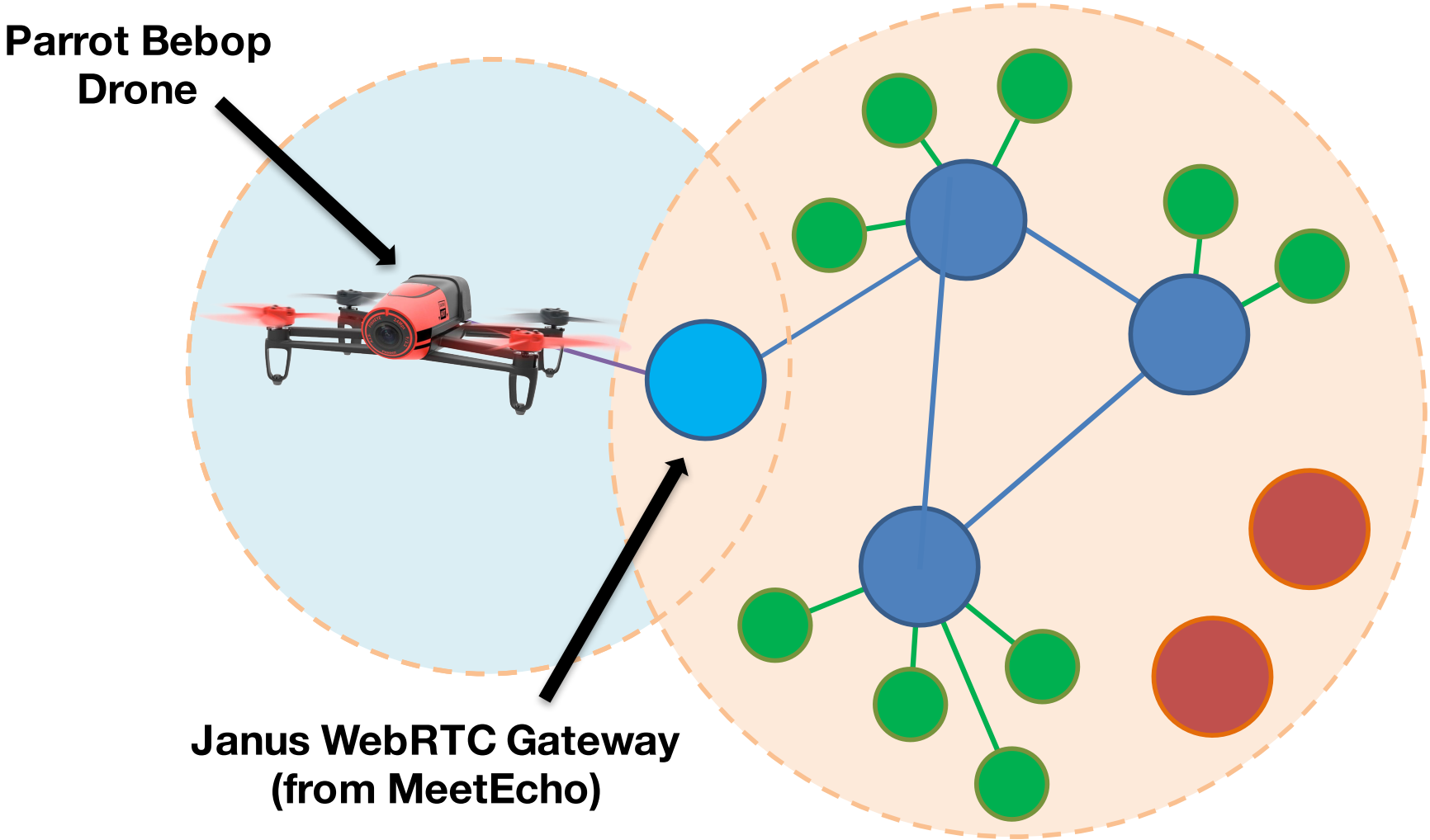
Latest Bridges!

- Official ones:
  - IRC
  - Slack
  - Gitter
  - Rocket.Chat
  - MatterMost
  - FreeSWITCH
  - Asterisk (Respoke)
  - libpurple
- Community ones
  - Twitter
  - Telegram
  - Hangouts
  - Slack webhooks
  - Gitter (‘sidecar’)
  - ~8 IRC ones…
  - ~4 XMPP ones...

Typical Bridging Stack



Matrix to IOT…



https://www.youtube.com/watch?v=D7jZSYkXqt4&t=2649

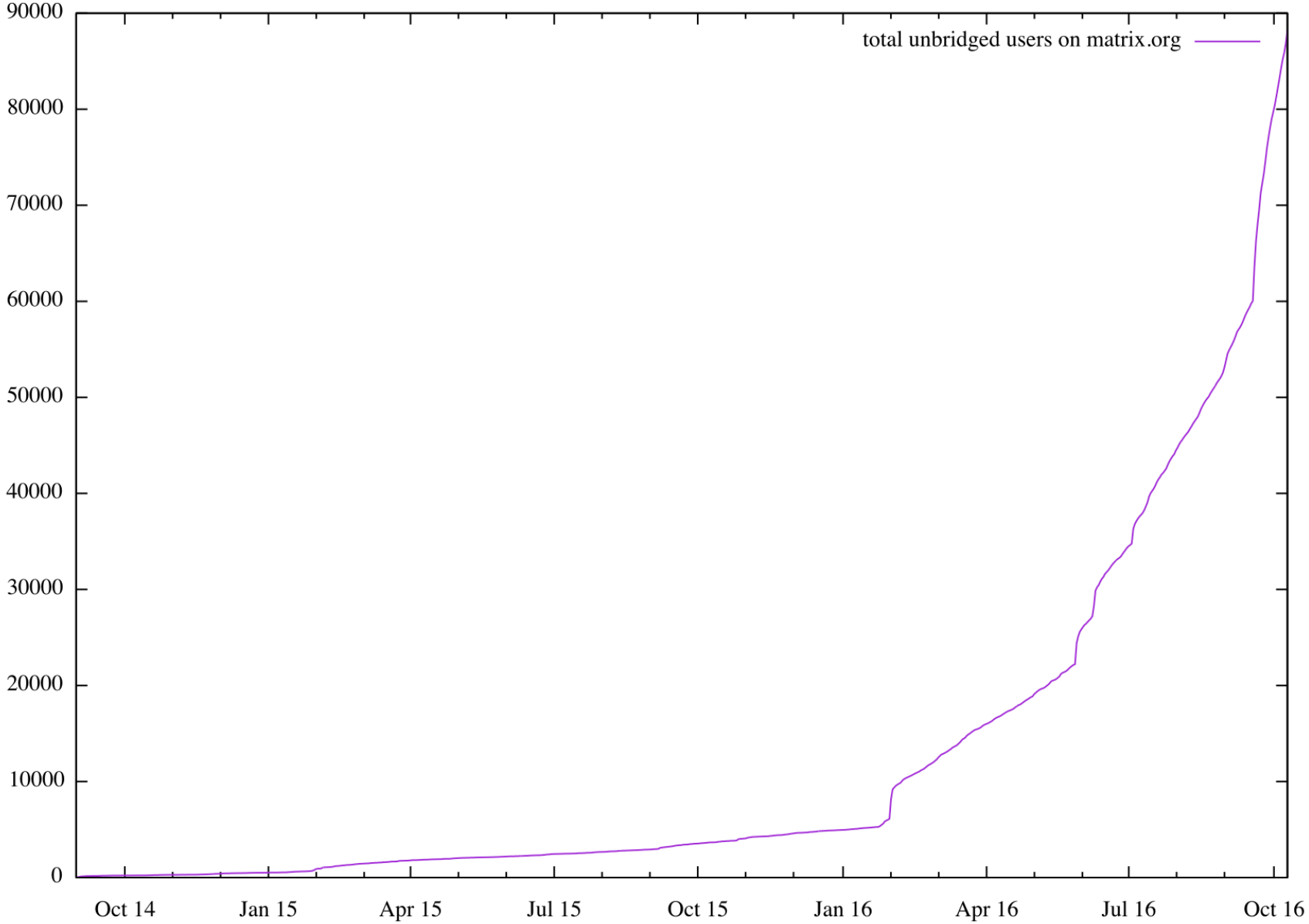
matrix

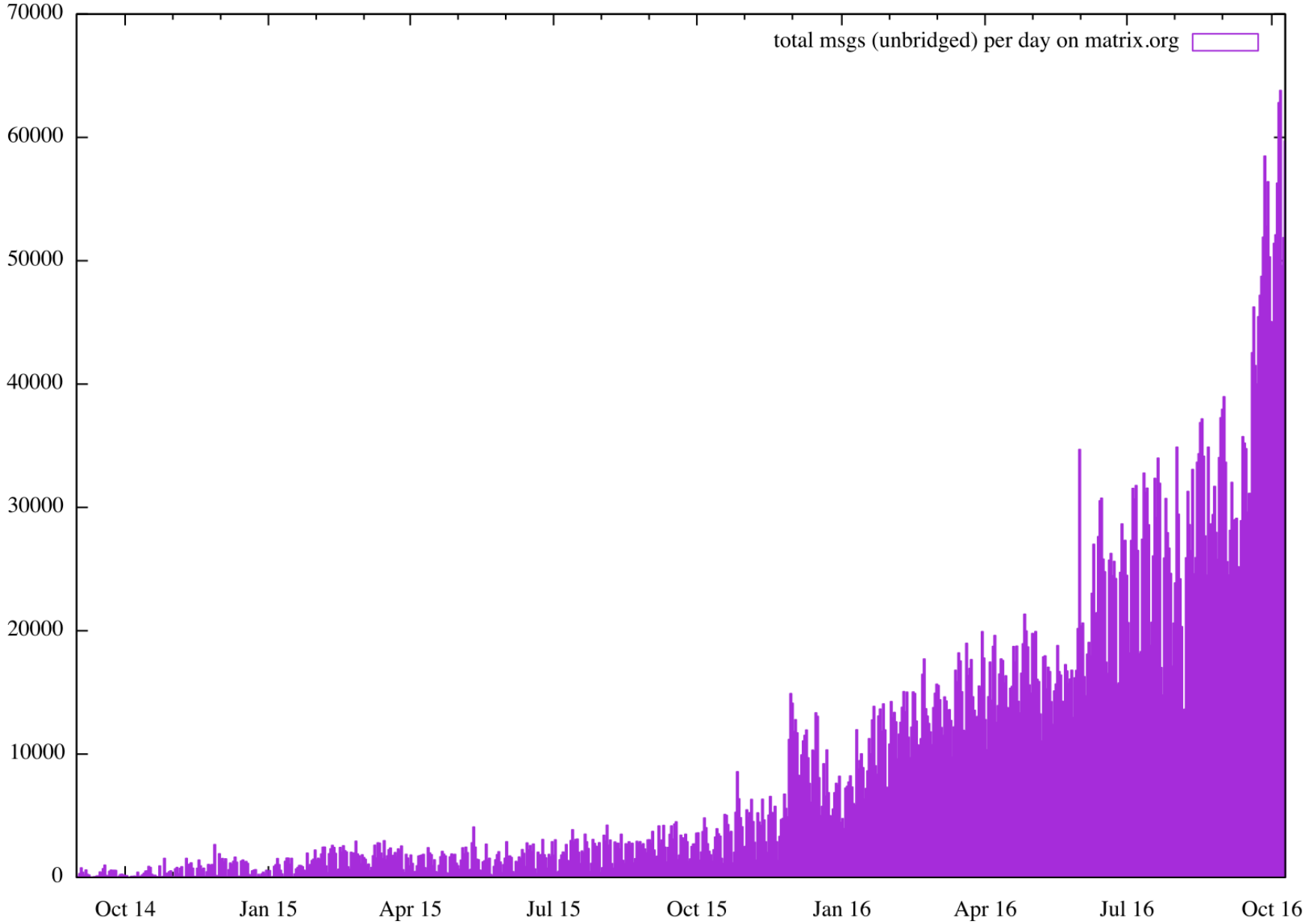
Matrix and VR…

matrix

Community Status

- Started out in Sept 2014
- Currently in very late beta
- ~450K user accounts on the Matrix.org homeserver (many of these are bridged)
- ~400K messages per day
- ~50K rooms that Matrix.org participates in
- ~1000 federated servers
- ~50 companies building on Matrix


total unbridged users on matrix.org
90000
80000
70000
60000
50000
40000
30000
20000
10000
0
Oct 14
Jan 15
Apr 15
Jul 15
Oct 15
Jan 16
Apr 16
Jul 16
Oct 16


total msgs (unbridged) per day on matrix.org
70000
60000
50000
40000
30000
20000
10000
0
Oct 14
Jan 15
Apr 15
Jul 15
Oct 15
Jan 16
Apr 16
Jul 16
Oct 16

matrix

End to End Crypto with Olm

https://matrix.org/git/olm

matrix

End to End Encryption

- 2 years in the making!
- Based on Open Whisper Systems' "Double Ratchet" alg as used in Signal etc.
- Audited by NCC Group
- Started final roll-out in Sept on Web
- Launching next week on iOS & Android (on develop branches currently)
- Supports per-target-device encryption
- Supports flexible history privacy per-room.

Olm

- Apache License C++11 implementation of Trevor Perrin / Moxie Marlinspike's Double Ratchet, exposing a C API.
- Supports encrypted asynchronous 1:1 communication.
- "Megolm" layer adds group communication too.
- 130KB x86-64 .so, or 208KB of asm.js

matrix

Olm + Megolm C API

Megolm Group
Ratchet

Account
- Keys

Session
- Initial Key Exchange

Ratchet

- Encrypt
- Decrypt

Crypto

- Curve25519
- AES
- SHA256

A Double ratchet.
Kinda sorta.

Alice

Bob

Alice and Bob both generate identity (I) & ephemeral (E) elliptic curve key pairs

Initial Shared Secret (ISS) =
ECDH(Ea, Ib) +
ECDH(Ia, Eb) +
ECDH(Ea, Eb)

Discard Ea
Derive chain key from ISS (HMAC)
Derive message key (K_0) from chain key (HMAC)
Derive new chain key ← **hash ratchet**
M_0 = Message plaintext
C_0 = Authenticated Encryption of (M_0, K_0)
Ra_0 = generate random ratchet key pair
Ja_0 = incremental counter for each hash ratchet advancement

→

Ia, Ea, Eb, Ra_0, Ja_0, C_0

A Double ratchet.
Kinda sorta.

Alice

Bob

Compute same Initial Shared Secret =
ECDH(Ea, Ib) +
ECDH(Ia, Eb) +
ECDH(Ea, Eb)

Compute same K_0
M_0 = Authenticated decryption of (C_0, K_0)

To respond, B starts new ratchet chain:
Rb_1 = generate random ratchet key pair
New Initial Shared Secret =
ECDH(Ra_0, Rb_1) ← **ECDH Ratchet**

C_0 = Authenticated Encryption of (M, K_0)
Ra_0 = generate random ratchet key
Ja_0 = incremental counter for each hash ratchet advancement

Rb_1, Jb_1, C_1

```
                Alice

    Sending     |     Receiving

MK      CK      RK      CK      MK
--      --      --      --      --
            ECDH(A0,B0)
                |
                |
    ECDH(A1,B0) +
               /|
              / |
             /  + ECDH(A1,B1)
     CK-A1-B0   |\
         |      | \
MK-0 ----+      |  \
         |      |   CK-A1-B1
MK-1 ----+      |       |
         |      |       +---- MK-0
MK-2 ----+      |       |
                |       +---- MK-1
    ECDH(A2,B1) +
               /|
              / |
             /  |
     CK-A2-B1   |
         |      + ECDH(A2,B2)
MK-0 ----+       \
                  \
                   \
                    CK-A2-B2
                        |
                        +---- MK-0
                        |
                        +---- MK-1
```

Group chat

- Adds a 3rd type of ratchet: “Megolm”, used to encrypt group messages.
- Establish 'normal' 1:1 ratchets between all participants in order to exchange the initial secret for the group ratchet.
- All receivers share the same group ratchet state to decrypt the room.

matrix

Flexible privacy with Olm

- Users can configure rooms to have:
  - No ratchet (i.e. no crypto)
  - Full PFS ratchet
  - Selective ratchet
    - Deliberately re-use ratchet keys to support paginating partial eras of history.
    - Up to participants to trigger the ratchet (e.g. when a member joins or leaves the room)
  - Per-message type ratchets?

Matrix: What's coming up?

- More hosted bridges, bots, services etc
- Threading
- Message tagging (e.g. "Like" support)
- Group ACLs
- File tagging and management
- Decentralised identity
- "Fixing spam"

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!

[matrix]

Thank you!

matthew@matrix.org
http://matrix.org
@matrixdotorg