

[matrix]

Strategies for OTT Federation

matthew@matrix.org

**The OTT ecosystem
isn't doing too badly.**

WhatsApp:

>600M MAU

60% MAU:DAU

97% penetration (in Spain)

**How do we harness this
success?**

Drives data...

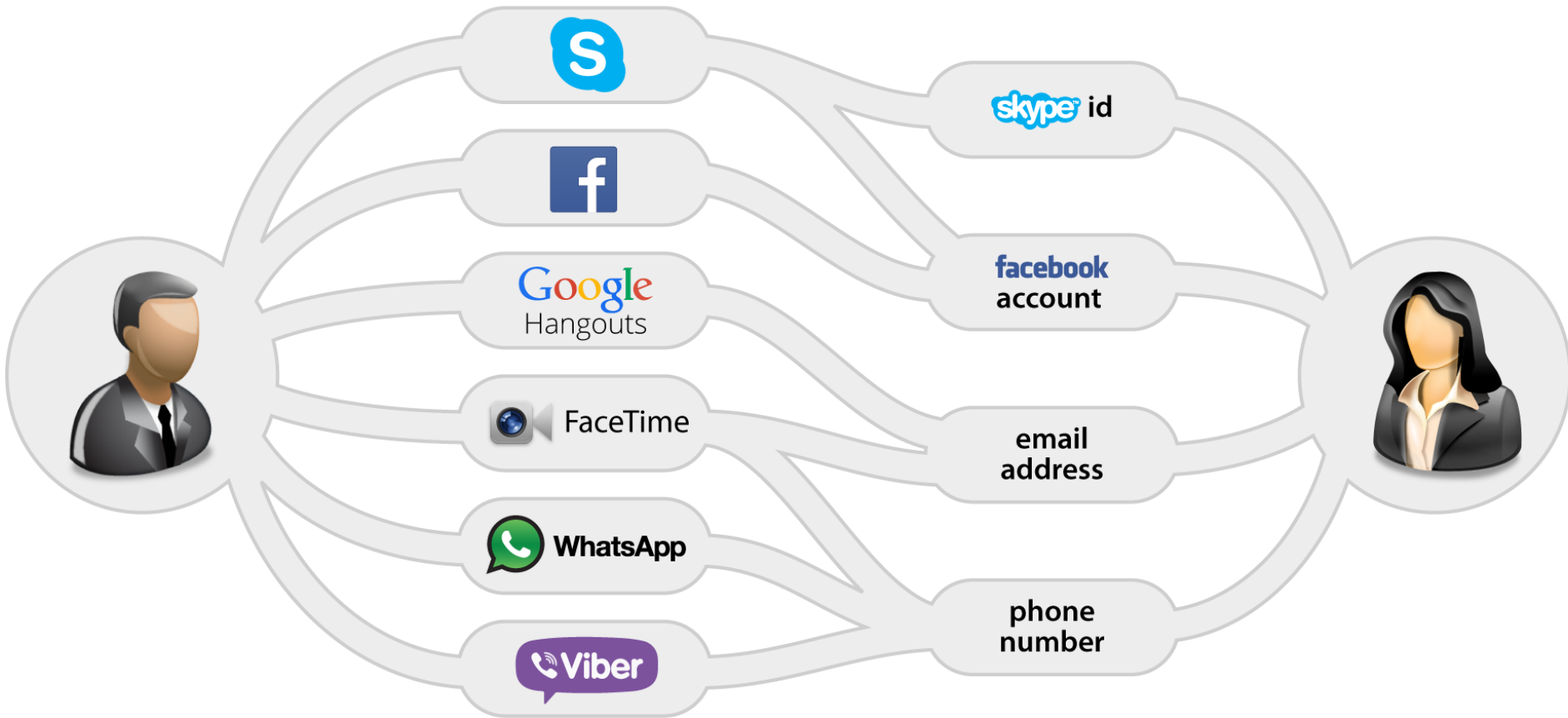
...but eats messaging/voice revenue.

So how many are there?

Google	Apple	Facebook	WhatsApp	Telegram	Blah
Hike	Groupme	Viber	Libon	Line	Gadu Gadu
Kik	Kakao Talk	Mxit	Talko	Tango	Threema
Tox	Snapchat	WeChat	QQ	Skype	AIM
ICQ	Yahoo!	BBM	Path	Pinterest	Twitter

...and all the others too...

**What's the end-user
experience like?**



**With so many apps you might
think there was great
consumer choice...**

**...but your contacts dictate
which app to use.**

**You can't pick the ones you
actually prefer.**

It's not "channel surfing TV"...

... it's like having 20 television sets in your living room, one per channel 😞

So why put up with it?

**Users seem to
blindly accept the situation.**

"It's just how it is."

If email suddenly became this fragmented, users would go apocalyptic.

So how did it end up like this?

1. The standards simply didn't work robustly on the real-world internet, encouraging startups to build proprietary closed solutions.

- **SIP lacked firewall traversal until ICE & TURN**
- **RTP lacked standardised dynamic bitrate control and FEC**
- **SDP lacks exhaustive capability negotiation**
- **XMPP provides too low a baseline featureset**

2. Federation of the old IM networks was not successful - by the time AIM/ICQ/MSN/Google Talk eventually managed to federate, they had been all but killed by Facebook.

3. The big OTTs don't have any incentive to federate: silos can themselves be financially successful.

It's "just" the end user who suffers...

[**matrix**]

**Matrix provides
a possible solution...**

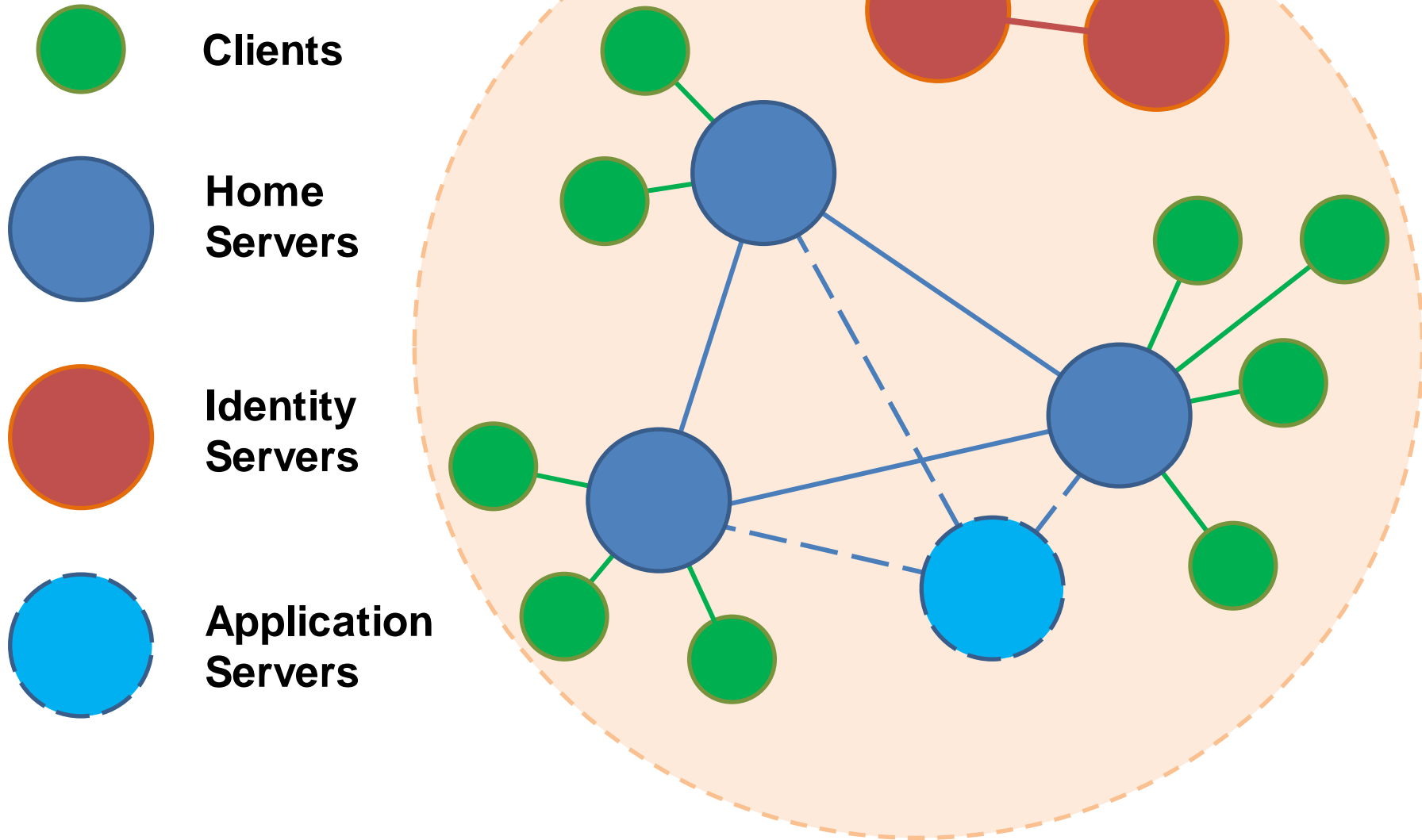
Introducing Matrix

- New Open Source project (launched Sept 2014)
- Setting up as non-profit org (matrix.org)
- Publishing pragmatic simple **HTTP** API standard for federated VoIP (**WebRTC**), IM and generic messaging.
- Defines client-server and server-server APIs (and, shortly, server<->application-server APIs).
- Provides Apache-Licensed reference implementations of the server and clients (web, iOS, Android, Python, Perl...)

Key Characteristics

- Entirely open:
 - open standard; open source; open project.
- Message History as first-class citizen
- Group communication as first-class citizen
 - Fully distributed room state (cryptographically signed) - no SPOFs or SPOCs.
- Strong cryptographic identity to prevent spoofing
- Identity agnostic
- End-to-end encryption (RSN)

Architecture



Strategy:

- Chase the long tail of:
 - Emerging OTTs
 - Telco OTTs
 - Tier 2-3 OTTs
- ...and glue them into one great big meta-OTT.
- Encourage vendors to build gateways to the PSTN (e.g. RCS, IMS, SS7)
- Try to convince the Big OTTs to expose their lowest common denominator service via Matrix.

**What does this have to do
with RCS!?**

RCS is great for Telco interworking.

OTT interworking is a very different problem domain.

RCS isn't exactly web- or internet-friendly technology.

Just for IM you need to understand MSRP, SDP, IMS, SIP, SIMPLE, XCAP...

**So we think both RCS and
OTT federation will co-exist.**

**Telcos will benefit from
extending RCS's reach to
OTT federation –
interoperating via gateways
and hubs.**

Finally, operators will be fully harnessing OTTs...

...and end-users will enjoy a fully seamless experience over PSTN and OTT services.

[**matrix**]

<http://matrix.org>

THANK YOU!

@matthew:matrix.org

matthew@matrix.org