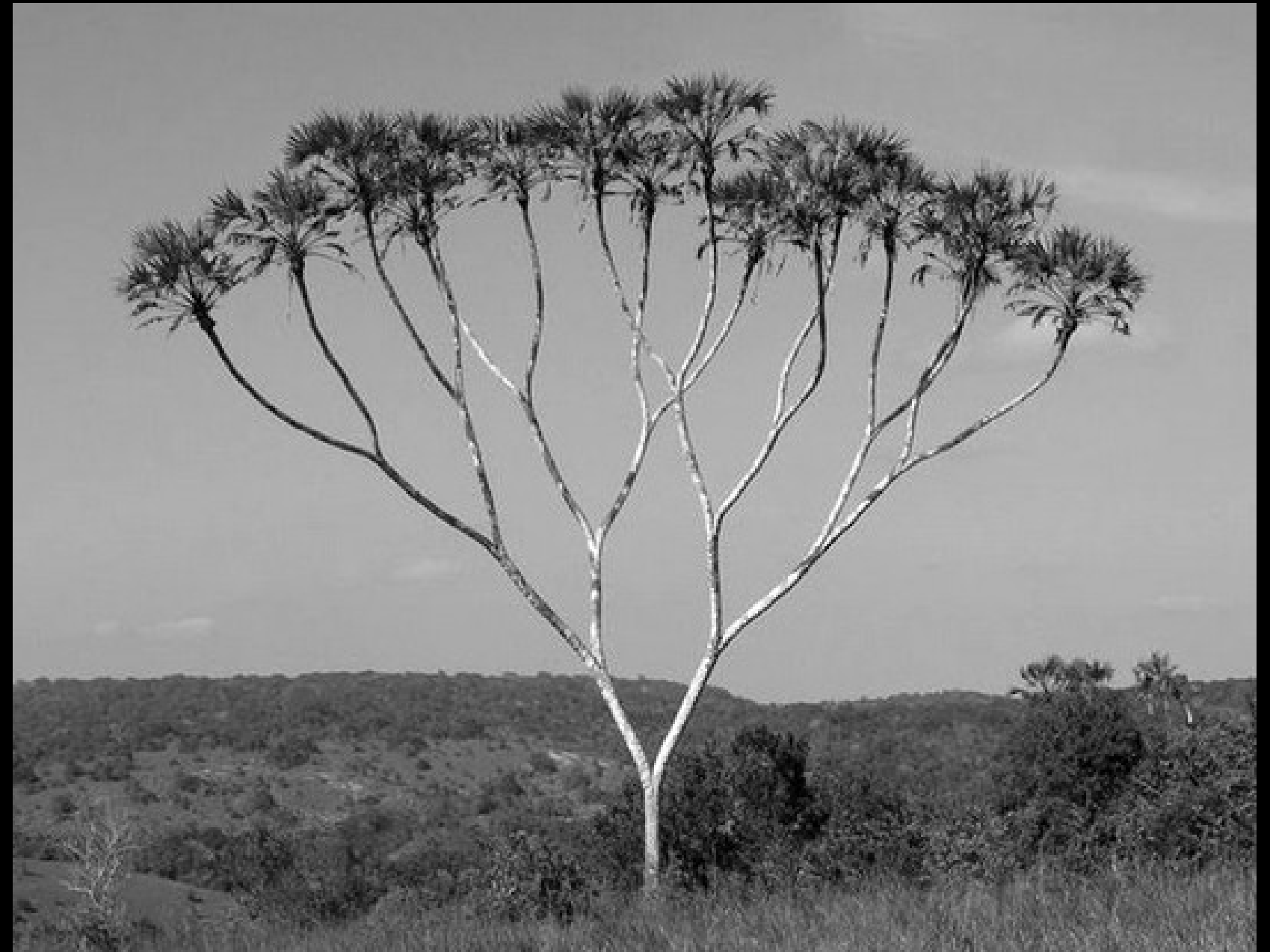


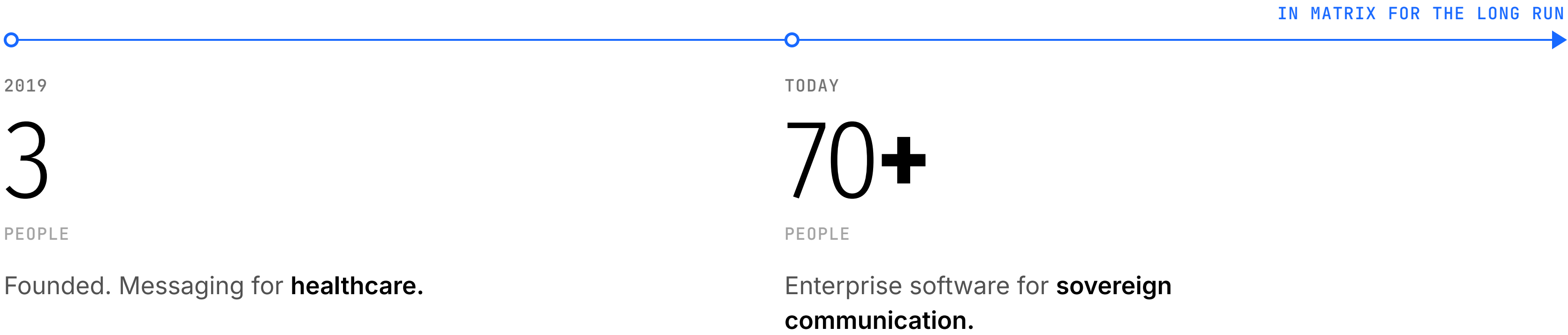
“The binary tree
actually **exists!!**”

@MrAhmadAwais · 18 May 2022



Hi.

From healthcare to **Matrix for the enterprise.**



Niklas Zender
CEO · Famedly

Matrix made this possible.
Thank you to everyone behind it.



HOW THE ECOSYSTEM IS DRIFTING

Secure Matrix shouldn't be a **paid tier**.

When security only ships proprietary, new companies stop building on Matrix.
And the protocol loses the people who carry it into new markets.



WHAT SOVEREIGN MEANS

Sovereignty needs an **open stack**.

Every layer open to read, run, fork and audit.
No single vendor that everything depends on.
Open isn't the cheap tier. It's the secure one.

Built in Germany. **Open for everyone.**

01

MLS and Matrix

- Group trust the client can verify
- RFC 9420, Matrix spec unchanged
- This talk · 10:30

02

Enterprise Readiness

- Security and operations first
- Why is there Rust in our Matrix?
- Talk 2 · 12:45

03

App, Adapters, Locks

- Bring it all together
- Demo time
- Talk 3 · 14:15

Group crypto, **freshly served.**

01



David Lassig
Platform Product Owner Backend · BWI

02



Timm Behner
Security Engineer (Cryptography) · BWI

CRYSPEN

CE · Labs

 **Famedly**

WATCH · MATRIX CONF 2025

Matrix and MLS: **Water, Oil and Mayonnaise**

Nico Werner · Famedly · on YouTube



MSC4256 · THE LAST COMMENT

Silence.

"Is this still being worked on?" The last comment on MSC4256, 8 February 2026.
No reply since.



MSC4256 · SINCE 8 FEBRUARY 2026 · VIA TENOR.COM

01

CHAPTER ONE • THE EFFORT

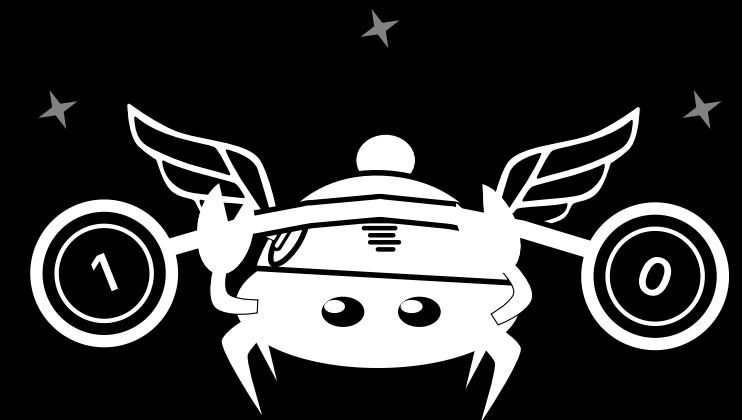


BILD: BUNDESWEHR

Why so much effort **for MLS*?**

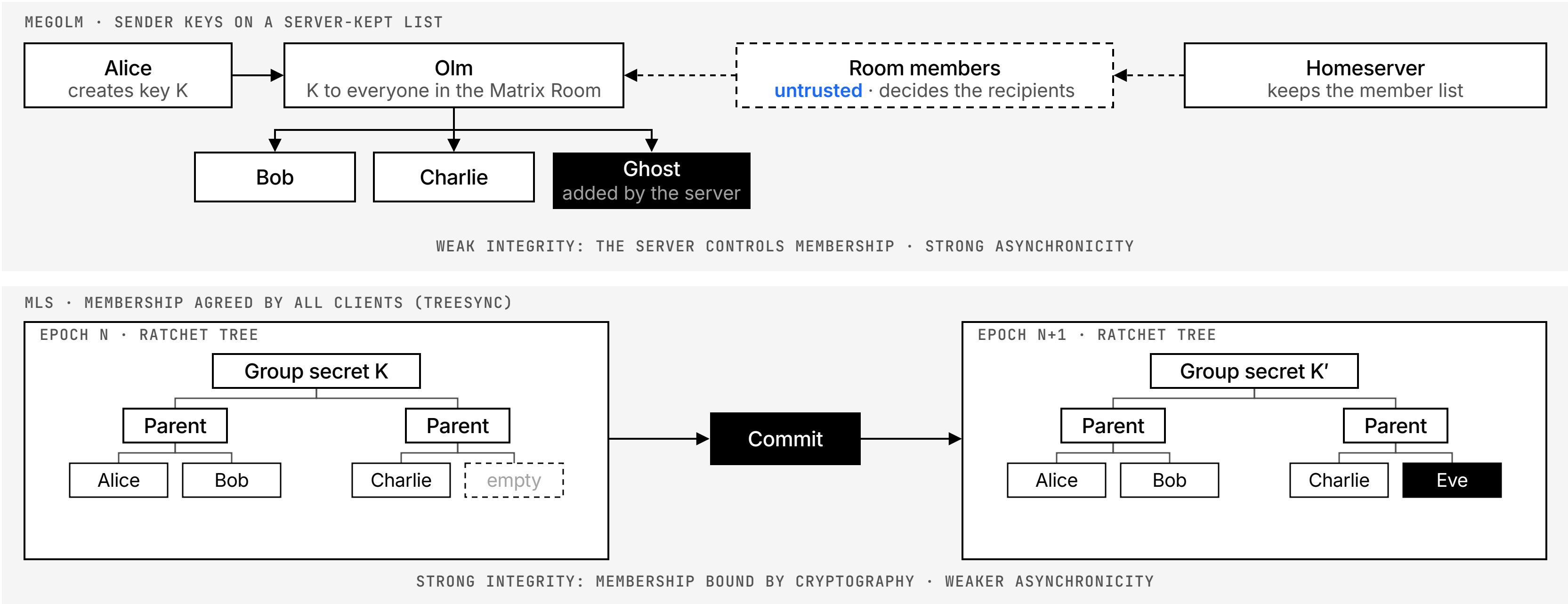
Matrix encrypts messages. MLS lets every member verify who is in the group.

That is the trust we came for.



VS-AP0024 IS NOT AN OPTION

Who decides **who is in the group?**



02

CHAPTER TWO • SEGIMER SDK

Segimer SDK.

One core, every client. An MLS abstraction layer for messaging apps.
Integrates with the Matrix Rust SDK and speaks MLS only.

Groups become **trees**.

- Not a new cipher: MLS takes over three jobs at once
- Every change is a commit, every commit a new epoch
- Post-compromise security, enforced by the protocol and not by a rotation policy

RFC 9420 • THREE LAYERS • AND MATRIX TODAY

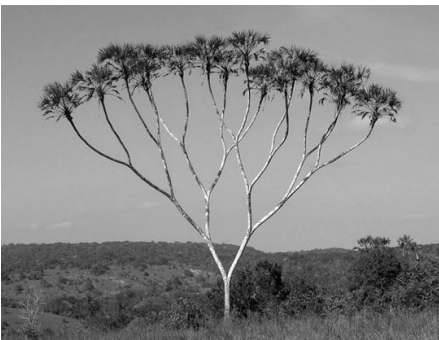
TreeSync	group state changes room state events
TreeKEM	key agreement Megolm keys sent over Olm
TreeDEM	secure channel Megolm ratchet + AES-CBC + HMAC

```
// Segimer SDK • segimer-app/src/group.rs
// The Welcome seals the group secrets to Bob's new leaf.
let welcome = group.add_user(provider, bob).await?;

// Sending is one call
let ct = group.encrypt_application_message(provider, &msg)?;

// Mallory's laptop is stolen. One commit, and the group heals:
// a Remove forces a fresh path up the tree (RFC 9420 §12.4).
group.remove_user(provider, mallory).await?;

// 47 → 48. Every secret Mallory ever held stops at that
// boundary, and nobody had to rebuild the room.
self.mls_group.epoch().as_u64()
```



03

CHAPTER THREE · METADATA AVOIDANCE

How do we keep identities
and membership verification
inside the MLS group?

Spoiler: **it's cryptography.**

Links users to their devices in the group, cryptographically. The Delivery Service and outsiders never learn who is in a group.

It is all **written down**.



NOTATION

u user_id d device_id π pseudonym vk/sk verify / sign key as Authentication Service

ALICE · THE CHAIN

01 DeviceCertificate The only signature a service makes.

$$c_{\text{dev}} = ((u, d, vk_d), \sigma_1), \quad \sigma_1 = \text{Sign}(sk_{\text{as}}, (u, d, vk_d))$$

02 RoomCertificate Fresh π and fresh (sk_r, vk_r) per group.

$$c_{\text{room}} = ((u, \pi, vk_r), \sigma_2), \quad \sigma_2 = \text{Sign}(sk_d, (u, \pi, vk_r))$$

03 Credential · LeafNode All the tree ever carries.

$$\text{cred} = (\pi, \sigma_3), \quad \sigma_3 = \text{Sign}(sk_r, \pi)$$

→ **Delivery Service · the wire**

$$E = \text{Enc}(k, (c_{\text{dev}}, c_{\text{room}})), \quad k = \text{Exp}(\text{epoch}, \text{certificate_entry})$$

$$L = (vk_{\text{leaf}}, \pi)$$

k is derivable by group members only. L goes in a PrivateMessage, and even in the clear names nobody.

BOB · ACCEPTS IFF

$$\begin{aligned} & \text{Vfy}(vk_{\text{as}}, (u, d, vk_d), \sigma_1) \\ & \wedge \text{Vfy}(vk_d, (u, \pi, vk_r), \sigma_2) \\ & \wedge \text{Vfy}(vk_r, \pi, \sigma_3) \\ & \wedge \pi = \pi_L \wedge vk_{\text{leaf}} = vk_r \end{aligned}$$

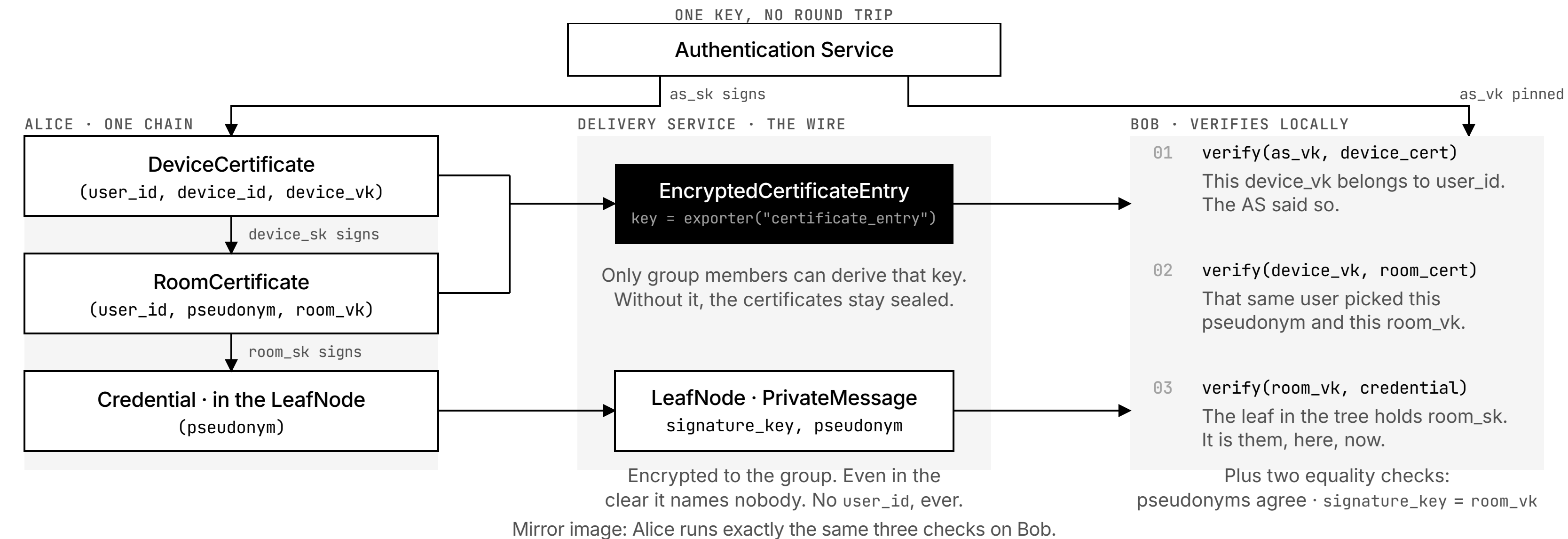
No network, no round trip: vk_as is pinned, the rest is signature maths. The last two equalities stop anyone stapling a foreign certificate to their own leaf.

Alice evaluates the same predicate on Bob.

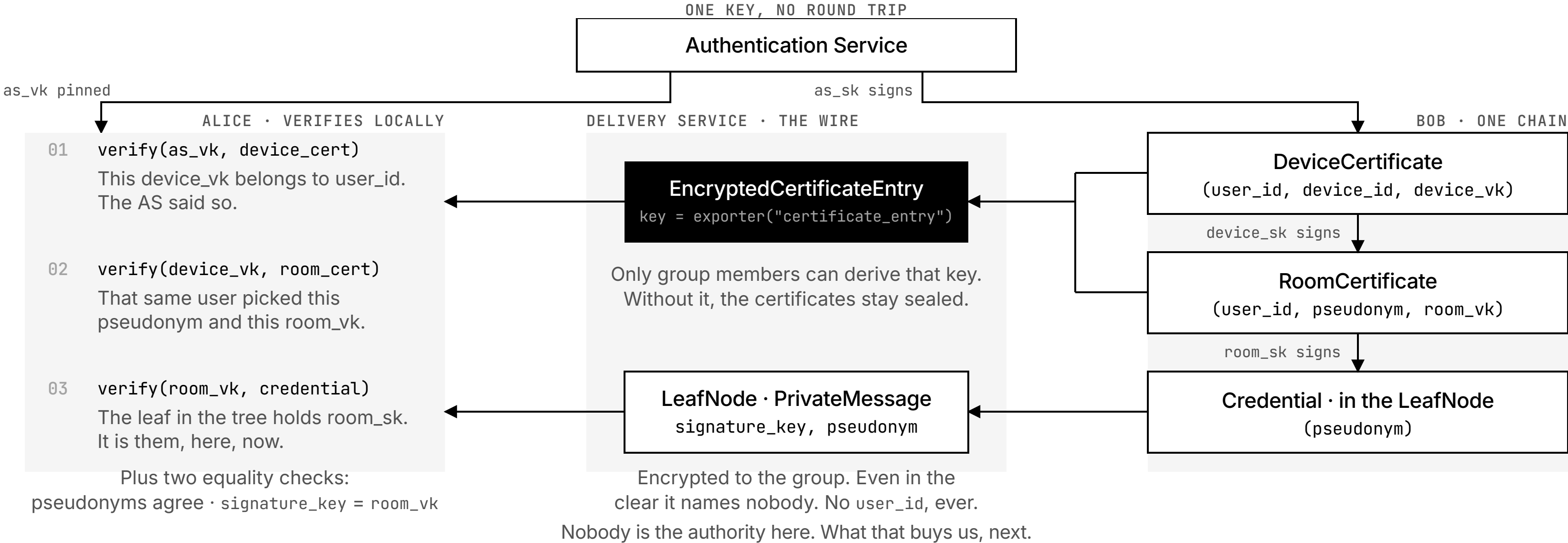
$$\text{view}_{\text{DS}} \subseteq (vk_{\text{leaf}}, \pi, E) \quad u \notin \text{view}_{\text{DS}}$$

No need to read any of this now. Next slide: the same thing, drawn.

Two members, **three signatures.**



Bob signs, **Alice checks.**



What MLS buys over **plain Matrix and Megolm**. And why BSI demands it.

01

Rotate and Scale

- (Theoretically) Efficiency for big groups with frequent membership changes
- No State Resolution necessary for Membership Enforcement

02

Standard and Sovereignty

- RFC 9420, a standard with rising industry backing
- libcrux - (mostly) formally verified crypto provider (by Cryspen)

03

Trust and Control

- Identities and membership stay in the client-side MLS group
- No server-side problematic metadata by design
- PCS and PFS by default - Every new epoch allows for new trust reset
- Ready for post-quantum: ML-KEM ciphersuite built, switch it on when we decide

04

MORE FROM US · TODAY

So, where does
all of this **lead us?**



TALK 2 · TODAY · KATHERINE

12:45–13:10

Why is there **Rust** in our Matrix?

Björn Reiners, David Lassig, Nico Werner

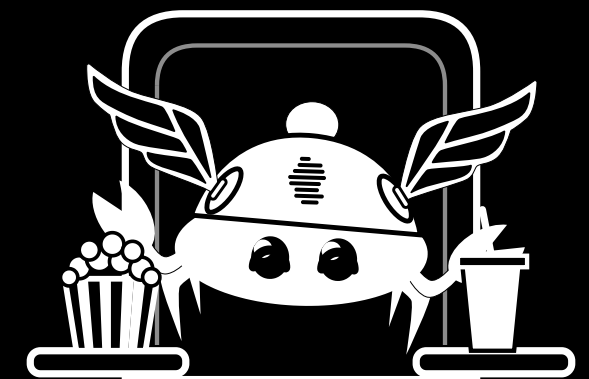


TALK 3 · TODAY · SOFYA

14:15–14:40

When to change **the locks** in your
Matrix.

David Lassig, Timm Behner, Björn Reiners



STICKERS • FREE

Ask us anything.
Leave with a crab.



One question, one sticker. Find us after the talk.

THANKS.

Questions? Please.

PLATFORM

Niklas Zender

CEO • Famedly

n.zender@famedly.com

ARCHITECTURE

David Lassig

Platform Product Owner Backend • BWI

david.lassig@bwi.de

CRYPTOGRAPHY

Timm Behner

Security Engineer (Cryptography) • BWI

tim.behner@bwi.de

BWI
IT für Deutschland

BundesMessenger is a product of BWI GmbH

FIND US ON SOCIALS



LINKEDIN

Famedly

Enterprise software for sovereign communication.

FIND US ON THE WEB



FAMEDLY.COM



BUNDESMESSENGER • MATRIX AND MLS