Public Key Infrastructure at ABN AMRO Bank

Real world cryptography in practice

Benoît Viguier – May 6th, 2024

ABN·AMRO

Who am I

PhD in Cryptography in 2021

Working at ABN AMRO since 2021:

- Crypto Services
- Secure Coding

On my (limited) free time:

- Main dev of FOSS Lychee (❤️ php)
- ildyria
- Photographer
- Top sport ballroom dancer

> « Il faut qu’il n’exige pas le secret, et qu’il puisse sans inconvénient tomber entre les mains de l’ennemi. »

Jean Guillaume Auguste Victor François Hubert Kerckhoffs — 1883

NL: Het ontwerp van het systeem behoort niet geheim te hoeven zijn, en moet zonder schadelijke gevolgen in vijandelijke handen kunnen vallen.

EN: The design of a system should not require secrecy, and compromise of the system should not inconvenience the correspondents

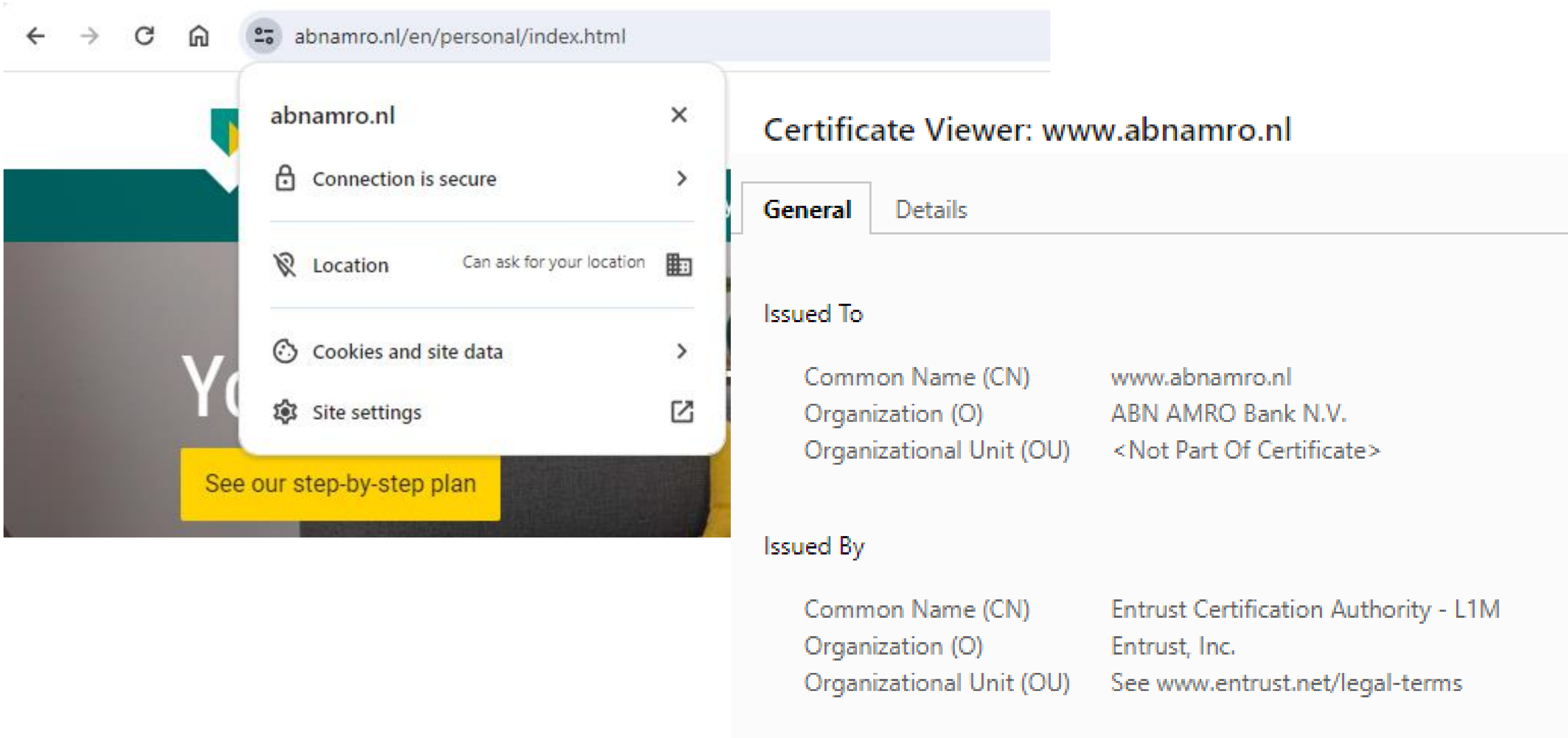
Agenda

1. Brief intro to TLS
2. Certificate Management at ABN AMRO
3. Certificate Authorities
4. Anatomy of a Key Ceremony
5. Things that could go wrong…

1 Brief intro to TLS

ABN·AMRO

Trusting a website



abnamro.nl/en/personal/index.html

abnamro.nl

Connection is secure

Location — Can ask for your location

Cookies and site data

Site settings

Certificate Viewer: www.abnamro.nl

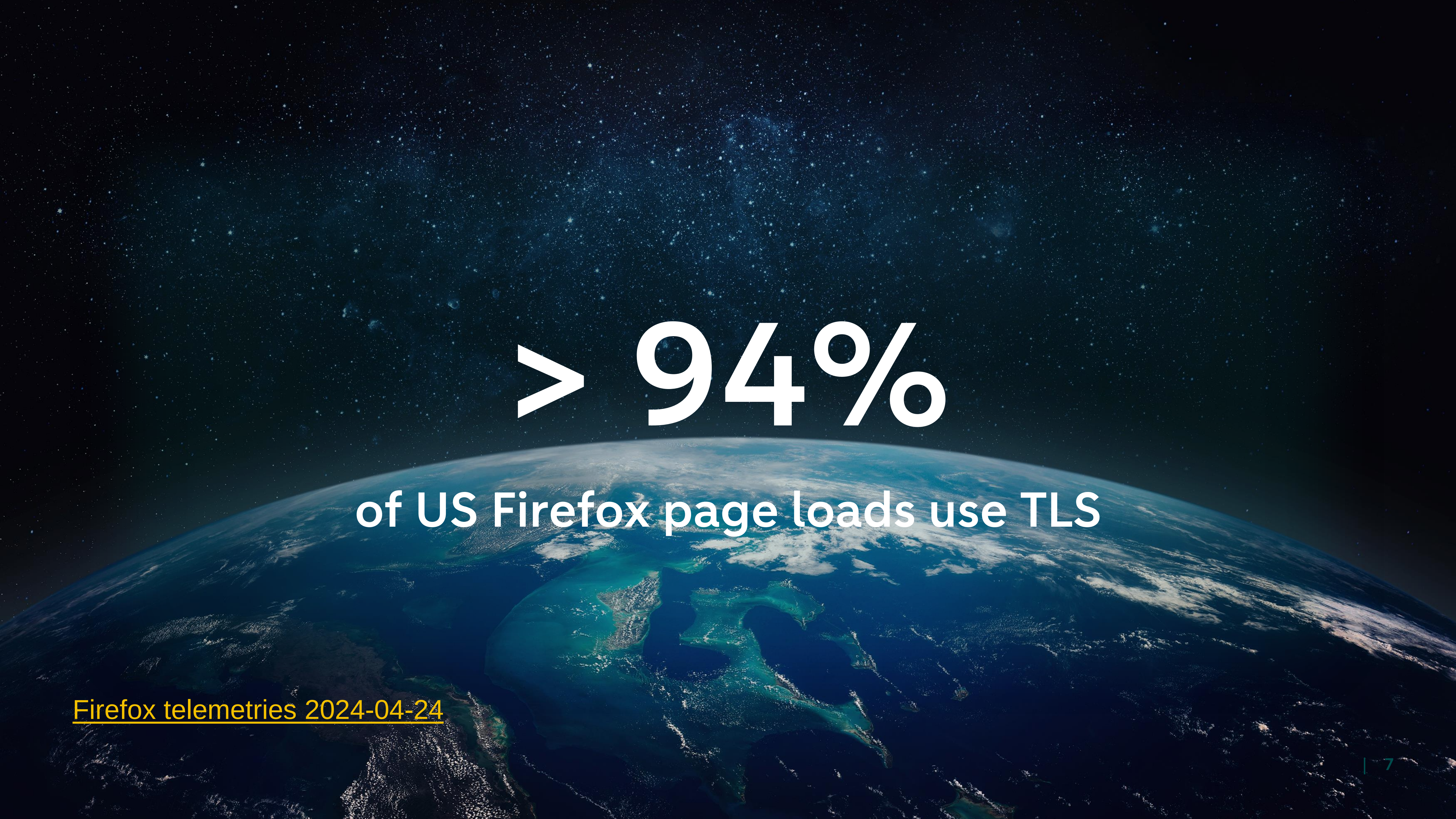
General | Details

Issued To

Common Name (CN)	www.abnamro.nl
Organization (O)	ABN AMRO Bank N.V.
Organizational Unit (OU)	<Not Part Of Certificate>

Issued By

Common Name (CN)	Entrust Certification Authority - L1M
Organization (O)	Entrust, Inc.
Organizational Unit (OU)	See www.entrust.net/legal-terms



> 94%

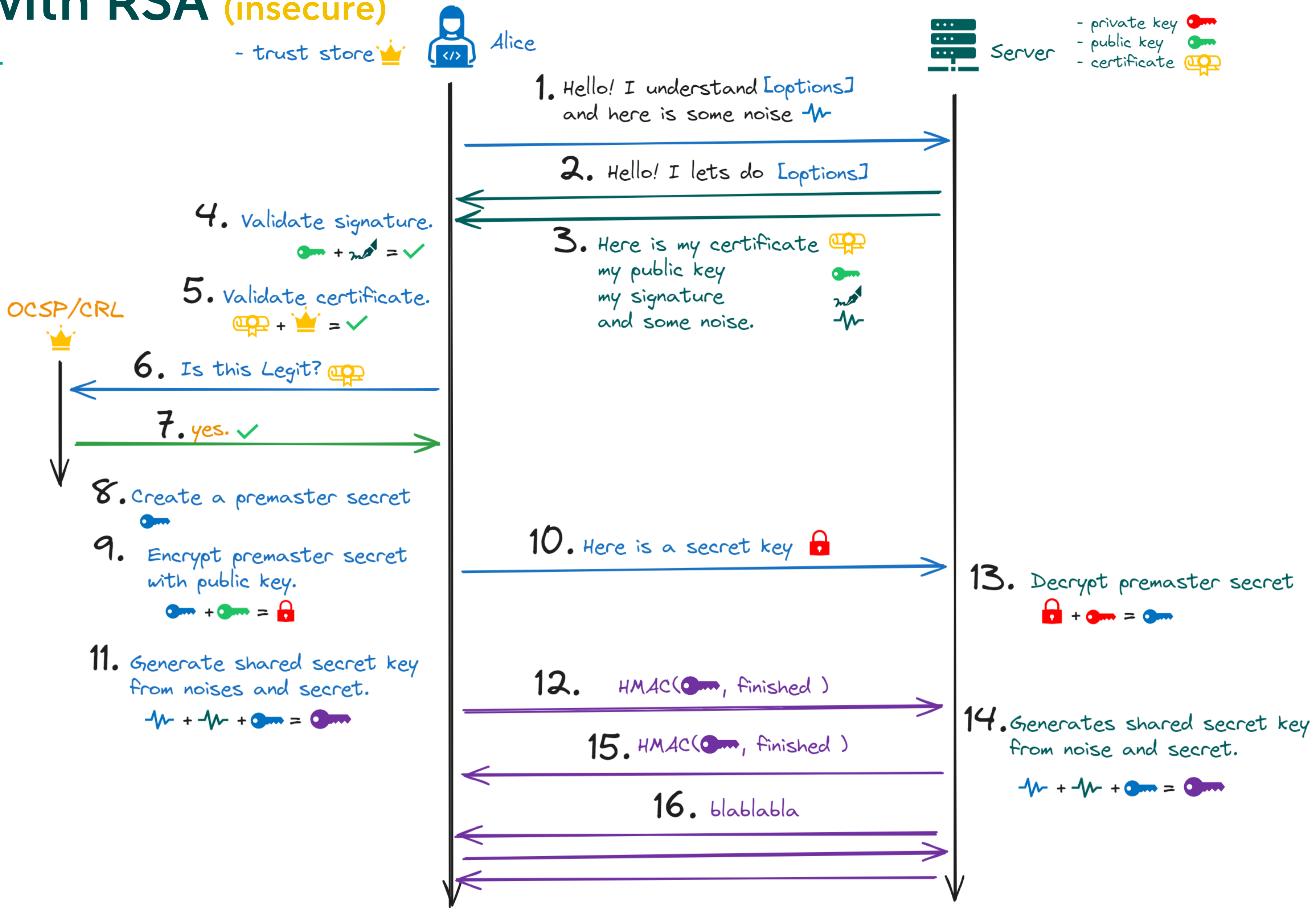
of US Firefox page loads use TLS

Firefox telemetries 2024-04-24

Transport Layer Security

- Colloquially still also known as “SSL”
- Often equated with https:// , but TLS is much more
  - OpenVPN, Cisco AnyConnect, Citrix NetScaler, Zscaler, and more VPNs are based on (D)TLS
  - WPA Enterprise has TLS auth modes,
  - Encrypted email transport (SMTPS),
  - VoIP, RTSP (streaming video), XMPP, …
- SSL 3.0 (RFC 6101$^{\text{(historic)}}$) ☠️
- TLS 1.0 (RFC 2246) — 1999 💀
- TLS 1.1 (RFC 4346) — 2006 💀
- TLS 1.2 (RFC 5246) — 2008 😬
- TLS 1.3 (RFC 8446) — 2018 💪

TLS 1.2 with RSA (insecure)



1. Hello! I understand [options]
and here is some noise

2. Hello! I lets do [options]

3. Here is my certificate
my public key
my signature
and some noise.

4. Validate signature.
+ = ✓

5. Validate certificate.
+ = ✓

6. Is this Legit?

7. yes. ✓

8. Create a premaster secret

9. Encrypt premaster secret
with public key.
+ =

10. Here is a secret key

11. Generate shared secret key
from noises and secret.
+ + =

12. HMAC(, finished)

13. Decrypt premaster secret
+ =

14. Generates shared secret key
from noise and secret.
+ + =

15. HMAC(, finished)

16. blablabla

TLS 1.2 problems

- **Too many round trips:**
  - Options
  - Randomness
  - Encryption of premaster key
  - HMAC
  - and more if ECDH…

- **Certificate sent in the clear** (everybody knows where you are connecting to).

- **Lots of legacy crypto** (which should no longer be used).

- **Lots of patches against attacks…**

TLS 1.2 problems

341 Cipher combinations

Only 20 **Recommended by IANA**

TLS_NULL_WITH_NULL_NULL
TLS_RSA_WITH_NULL_MD5
TLS_RSA_WITH_NULL_SHA
TLS_RSA_EXPORT_WITH_RC4_40_MD5
TLS_RSA_WITH_RC4_128_MD5
TLS_RSA_WITH_RC4_128_SHA
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
TLS_RSA_WITH_IDEA_CBC_SHA
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_RSA_WITH_DES_CBC_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA
TLS_DH_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_DSS_WITH_DES_CBC_SHA
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_RSA_WITH_DES_CBC_SHA
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA
TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_DSS_WITH_DES_CBC_SHA
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_RSA_WITH_DES_CBC_SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
TLS_DH_anon_EXPORT_WITH_RC4_40_MD5
TLS_DH_anon_WITH_RC4_128_MD5
TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_anon_WITH_DES_CBC_SHA
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA
TLS_KRB5_WITH_DES_CBC_SHA
TLS_KRB5_WITH_3DES_EDE_CBC_SHA
TLS_KRB5_WITH_RC4_128_SHA
TLS_KRB5_WITH_IDEA_CBC_SHA
TLS_KRB5_WITH_DES_CBC_MD5
TLS_KRB5_WITH_3DES_EDE_CBC_MD5
TLS_KRB5_WITH_RC4_128_MD5
TLS_KRB5_WITH_IDEA_CBC_MD5
TLS_KRB5_EXPORT_WITH_DES_CBC_40_SHA
TLS_KRB5_EXPORT_WITH_RC2_CBC_40_SHA
TLS_KRB5_EXPORT_WITH_RC4_40_SHA
TLS_KRB5_EXPORT_WITH_DES_CBC_40_MD5
TLS_KRB5_EXPORT_WITH_RC2_CBC_40_MD5
TLS_KRB5_EXPORT_WITH_RC4_40_MD5
TLS_PSK_WITH_NULL_SHA
TLS_DHE_PSK_WITH_NULL_SHA
TLS_RSA_PSK_WITH_NULL_SHA
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_DH_DSS_WITH_AES_128_CBC_SHA
TLS_DH_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_DSS_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DH_anon_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DH_DSS_WITH_AES_256_CBC_SHA
TLS_DH_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DH_anon_WITH_AES_256_CBC_SHA

TLS_RSA_WITH_NULL_SHA256
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_DH_DSS_WITH_AES_128_CBC_SHA256
TLS_DH_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA
TLS_DH_DSS_WITH_CAMELLIA_128_CBC_SHA
TLS_DH_RSA_WITH_CAMELLIA_128_CBC_SHA
TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA
TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA
TLS_DH_anon_WITH_CAMELLIA_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DH_DSS_WITH_AES_256_CBC_SHA256
TLS_DH_RSA_WITH_AES_256_CBC_SHA256
TLS_DHE_DSS_WITH_AES_256_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
TLS_DH_anon_WITH_AES_128_CBC_SHA256
TLS_DH_anon_WITH_AES_256_CBC_SHA256
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA
TLS_DH_DSS_WITH_CAMELLIA_256_CBC_SHA
TLS_DH_RSA_WITH_CAMELLIA_256_CBC_SHA
TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA
TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA
TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA
TLS_PSK_WITH_RC4_128_SHA
TLS_PSK_WITH_3DES_EDE_CBC_SHA
TLS_PSK_WITH_AES_128_CBC_SHA
TLS_PSK_WITH_AES_256_CBC_SHA
TLS_DHE_PSK_WITH_RC4_128_SHA
TLS_DHE_PSK_WITH_3DES_EDE_CBC_SHA
TLS_DHE_PSK_WITH_AES_128_CBC_SHA
TLS_DHE_PSK_WITH_AES_256_CBC_SHA
TLS_RSA_PSK_WITH_RC4_128_SHA
TLS_RSA_PSK_WITH_3DES_EDE_CBC_SHA
TLS_RSA_PSK_WITH_AES_128_CBC_SHA
TLS_RSA_PSK_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_SEED_CBC_SHA
TLS_DH_DSS_WITH_SEED_CBC_SHA
TLS_DH_RSA_WITH_SEED_CBC_SHA
TLS_DHE_DSS_WITH_SEED_CBC_SHA
TLS_DHE_RSA_WITH_SEED_CBC_SHA
TLS_DH_anon_WITH_SEED_CBC_SHA
TLS_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DH_RSA_WITH_AES_128_GCM_SHA256
TLS_DH_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256
TLS_DHE_DSS_WITH_AES_256_GCM_SHA384
TLS_DH_DSS_WITH_AES_128_GCM_SHA256
TLS_DH_DSS_WITH_AES_256_GCM_SHA384
TLS_DH_anon_WITH_AES_128_GCM_SHA256
TLS_DH_anon_WITH_AES_256_GCM_SHA384
TLS_PSK_WITH_AES_128_GCM_SHA256
TLS_PSK_WITH_AES_256_GCM_SHA384

TLS_DHE_PSK_WITH_AES_128_GCM_SHA256
TLS_DHE_PSK_WITH_AES_256_GCM_SHA384
TLS_RSA_PSK_WITH_AES_128_GCM_SHA256
TLS_RSA_PSK_WITH_AES_256_GCM_SHA384
TLS_PSK_WITH_AES_128_CBC_SHA256
TLS_PSK_WITH_AES_256_CBC_SHA384
TLS_PSK_WITH_NULL_SHA256
TLS_PSK_WITH_NULL_SHA384
TLS_DHE_PSK_WITH_AES_128_CBC_SHA256
TLS_DHE_PSK_WITH_AES_256_CBC_SHA384
TLS_DHE_PSK_WITH_NULL_SHA256
TLS_DHE_PSK_WITH_NULL_SHA384
TLS_RSA_PSK_WITH_AES_128_CBC_SHA256
TLS_RSA_PSK_WITH_AES_256_CBC_SHA384
TLS_RSA_PSK_WITH_NULL_SHA256
TLS_RSA_PSK_WITH_NULL_SHA384
TLS_RSA_WITH_CAMELLIA_128_CBC_SHA256
TLS_DH_DSS_WITH_CAMELLIA_128_CBC_SHA256
TLS_DH_RSA_WITH_CAMELLIA_128_CBC_SHA256
TLS_DHE_DSS_WITH_CAMELLIA_128_CBC_SHA256
TLS_DHE_RSA_WITH_CAMELLIA_128_CBC_SHA256
TLS_DH_anon_WITH_CAMELLIA_128_CBC_SHA256
TLS_RSA_WITH_CAMELLIA_256_CBC_SHA256
TLS_DH_DSS_WITH_CAMELLIA_256_CBC_SHA256
TLS_DH_RSA_WITH_CAMELLIA_256_CBC_SHA256
TLS_DHE_DSS_WITH_CAMELLIA_256_CBC_SHA256
TLS_DHE_RSA_WITH_CAMELLIA_256_CBC_SHA256
TLS_DH_anon_WITH_CAMELLIA_256_CBC_SHA256
TLS_ECDH_ECDSA_WITH_NULL_SHA
TLS_ECDH_ECDSA_WITH_RC4_128_SHA
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_NULL_SHA
TLS_ECDHE_ECDSA_WITH_RC4_128_SHA
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDH_RSA_WITH_NULL_SHA
TLS_ECDH_RSA_WITH_RC4_128_SHA
TLS_ECDH_RSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_NULL_SHA
TLS_ECDHE_RSA_WITH_RC4_128_SHA
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_ECDH_anon_WITH_NULL_SHA
TLS_ECDH_anon_WITH_RC4_128_SHA
TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA
TLS_ECDH_anon_WITH_AES_128_CBC_SHA
TLS_ECDH_anon_WITH_AES_256_CBC_SHA
TLS_SRP_SHA_WITH_3DES_EDE_CBC_SHA
TLS_SRP_SHA_RSA_WITH_3DES_EDE_CBC_SHA
TLS_SRP_SHA_DSS_WITH_3DES_EDE_CBC_SHA
TLS_SRP_SHA_WITH_AES_128_CBC_SHA

TLS_SRP_SHA_RSA_WITH_AES_128_CBC_SHA
TLS_SRP_SHA_DSS_WITH_AES_128_CBC_SHA
TLS_SRP_SHA_WITH_AES_256_CBC_SHA
TLS_SRP_SHA_RSA_WITH_AES_256_CBC_SHA
TLS_SRP_SHA_DSS_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_PSK_WITH_RC4_128_SHA
TLS_ECDHE_PSK_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA
TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA
TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA256
TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA384
TLS_ECDHE_PSK_WITH_NULL_SHA
TLS_ECDHE_PSK_WITH_NULL_SHA256
TLS_ECDHE_PSK_WITH_NULL_SHA384
TLS_RSA_WITH_ARIA_128_CBC_SHA256
TLS_RSA_WITH_ARIA_256_CBC_SHA384
TLS_DH_DSS_WITH_ARIA_128_CBC_SHA256
TLS_DH_DSS_WITH_ARIA_256_CBC_SHA384
TLS_DH_RSA_WITH_ARIA_128_CBC_SHA256
TLS_DH_RSA_WITH_ARIA_256_CBC_SHA384
TLS_DHE_DSS_WITH_ARIA_128_CBC_SHA256
TLS_DHE_DSS_WITH_ARIA_256_CBC_SHA384
TLS_DHE_RSA_WITH_ARIA_128_CBC_SHA256
TLS_DHE_RSA_WITH_ARIA_256_CBC_SHA384
TLS_DH_anon_WITH_ARIA_128_CBC_SHA256
TLS_DH_anon_WITH_ARIA_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_ARIA_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_ARIA_256_CBC_SHA384
TLS_ECDH_ECDSA_WITH_ARIA_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_ARIA_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_ARIA_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_ARIA_256_CBC_SHA384
TLS_ECDH_RSA_WITH_ARIA_128_CBC_SHA256
TLS_ECDH_RSA_WITH_ARIA_256_CBC_SHA384
TLS_RSA_WITH_ARIA_128_GCM_SHA256
TLS_RSA_WITH_ARIA_256_GCM_SHA384
TLS_DHE_RSA_WITH_ARIA_128_GCM_SHA256
TLS_DHE_RSA_WITH_ARIA_256_GCM_SHA384
TLS_DH_RSA_WITH_ARIA_128_GCM_SHA256
TLS_DH_RSA_WITH_ARIA_256_GCM_SHA384
TLS_DHE_DSS_WITH_ARIA_128_GCM_SHA256

TLS_DHE_DSS_WITH_ARIA_256_GCM_SHA384
TLS_DH_DSS_WITH_ARIA_128_GCM_SHA256
TLS_DH_DSS_WITH_ARIA_256_GCM_SHA384
TLS_DH_anon_WITH_ARIA_128_GCM_SHA256
TLS_DH_anon_WITH_ARIA_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_ARIA_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_ARIA_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_ARIA_128_GCM_SHA256
TLS_ECDH_ECDSA_WITH_ARIA_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_ARIA_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_ARIA_256_GCM_SHA384
TLS_ECDH_RSA_WITH_ARIA_128_GCM_SHA256
TLS_ECDH_RSA_WITH_ARIA_256_GCM_SHA384
TLS_PSK_WITH_ARIA_128_CBC_SHA256
TLS_PSK_WITH_ARIA_256_CBC_SHA384
TLS_DHE_PSK_WITH_ARIA_128_CBC_SHA256
TLS_DHE_PSK_WITH_ARIA_256_CBC_SHA384
TLS_RSA_PSK_WITH_ARIA_128_CBC_SHA256
TLS_RSA_PSK_WITH_ARIA_256_CBC_SHA384
TLS_PSK_WITH_ARIA_128_GCM_SHA256
TLS_PSK_WITH_ARIA_256_GCM_SHA384
TLS_DHE_PSK_WITH_ARIA_128_GCM_SHA256
TLS_DHE_PSK_WITH_ARIA_256_GCM_SHA384
TLS_RSA_PSK_WITH_ARIA_128_GCM_SHA256
TLS_RSA_PSK_WITH_ARIA_256_GCM_SHA384
TLS_ECDHE_PSK_WITH_ARIA_128_CBC_SHA256
TLS_ECDHE_PSK_WITH_ARIA_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_CBC_SHA384
TLS_ECDH_ECDSA_WITH_CAMELLIA_128_CBC_SHA256
TLS_ECDH_ECDSA_WITH_CAMELLIA_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_CAMELLIA_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_CAMELLIA_256_CBC_SHA384
TLS_ECDH_RSA_WITH_CAMELLIA_128_CBC_SHA256
TLS_ECDH_RSA_WITH_CAMELLIA_256_CBC_SHA384
TLS_RSA_WITH_CAMELLIA_128_GCM_SHA256
TLS_RSA_WITH_CAMELLIA_256_GCM_SHA384
TLS_DHE_RSA_WITH_CAMELLIA_128_GCM_SHA256
TLS_DHE_RSA_WITH_CAMELLIA_256_GCM_SHA384
TLS_DH_RSA_WITH_CAMELLIA_128_GCM_SHA256
TLS_DH_RSA_WITH_CAMELLIA_256_GCM_SHA384
TLS_DHE_DSS_WITH_CAMELLIA_128_GCM_SHA256
TLS_DHE_DSS_WITH_CAMELLIA_256_GCM_SHA384
TLS_DH_DSS_WITH_CAMELLIA_128_GCM_SHA256
TLS_DH_DSS_WITH_CAMELLIA_256_GCM_SHA384
TLS_DH_anon_WITH_CAMELLIA_128_GCM_SHA256
TLS_DH_anon_WITH_CAMELLIA_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_CAMELLIA_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_CAMELLIA_256_GCM_SHA384
TLS_ECDH_ECDSA_WITH_CAMELLIA_128_GCM_SHA256
TLS_ECDH_ECDSA_WITH_CAMELLIA_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_CAMELLIA_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_CAMELLIA_256_GCM_SHA384
TLS_ECDH_RSA_WITH_CAMELLIA_128_GCM_SHA256
TLS_ECDH_RSA_WITH_CAMELLIA_256_GCM_SHA384
TLS_PSK_WITH_CAMELLIA_128_GCM_SHA256
TLS_PSK_WITH_CAMELLIA_256_GCM_SHA384

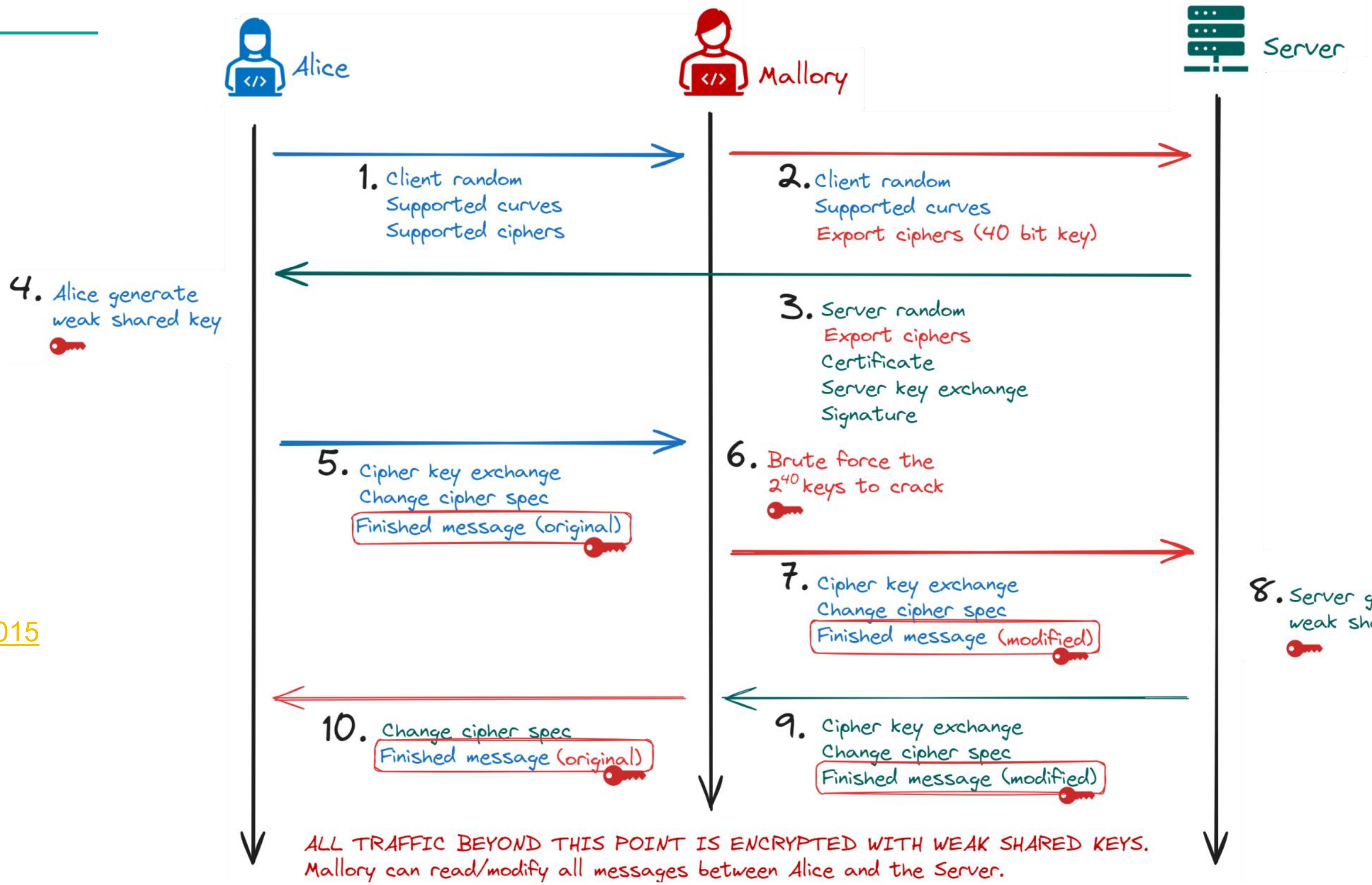
TLS_DHE_PSK_WITH_CAMELLIA_128_GCM_SHA256
TLS_DHE_PSK_WITH_CAMELLIA_256_GCM_SHA384
TLS_RSA_PSK_WITH_CAMELLIA_128_GCM_SHA256
TLS_RSA_PSK_WITH_CAMELLIA_256_GCM_SHA384
TLS_PSK_WITH_CAMELLIA_128_CBC_SHA256
TLS_PSK_WITH_CAMELLIA_256_CBC_SHA384
TLS_DHE_PSK_WITH_CAMELLIA_128_CBC_SHA256
TLS_DHE_PSK_WITH_CAMELLIA_256_CBC_SHA384
TLS_RSA_PSK_WITH_CAMELLIA_128_CBC_SHA256
TLS_RSA_PSK_WITH_CAMELLIA_256_CBC_SHA384
TLS_ECDHE_PSK_WITH_CAMELLIA_128_CBC_SHA256
TLS_ECDHE_PSK_WITH_CAMELLIA_256_CBC_SHA384
TLS_RSA_WITH_AES_128_CCM
TLS_RSA_WITH_AES_256_CCM
TLS_DHE_RSA_WITH_AES_128_CCM
TLS_DHE_RSA_WITH_AES_256_CCM
TLS_RSA_WITH_AES_128_CCM_8
TLS_RSA_WITH_AES_256_CCM_8
TLS_DHE_RSA_WITH_AES_128_CCM_8
TLS_DHE_RSA_WITH_AES_256_CCM_8
TLS_PSK_WITH_AES_128_CCM
TLS_PSK_WITH_AES_256_CCM
TLS_DHE_PSK_WITH_AES_128_CCM
TLS_DHE_PSK_WITH_AES_256_CCM
TLS_PSK_WITH_AES_128_CCM_8
TLS_PSK_WITH_AES_256_CCM_8
TLS_PSK_DHE_WITH_AES_128_CCM_8
TLS_PSK_DHE_WITH_AES_256_CCM_8
TLS_ECDHE_ECDSA_WITH_AES_128_CCM
TLS_ECDHE_ECDSA_WITH_AES_256_CCM
TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8
TLS_ECCPWD_WITH_AES_128_GCM_SHA256
TLS_ECCPWD_WITH_AES_256_GCM_SHA384
TLS_ECCPWD_WITH_AES_128_CCM_SHA256
TLS_ECCPWD_WITH_AES_256_CCM_SHA384
TLS_SHA256_SHA256
TLS_SHA384_SHA384
TLS_GOSTR341112_256_WITH_KUZNYECHIK_CTR_OMAC
TLS_GOSTR341112_256_WITH_MAGMA_CTR_OMAC
TLS_GOSTR341112_256_WITH_28147_CNT_IMIT
TLS_GOSTR341112_256_WITH_KUZNYECHIK_MGM_L
TLS_GOSTR341112_256_WITH_MAGMA_MGM_L
TLS_GOSTR341112_256_WITH_KUZNYECHIK_MGM_S
TLS_GOSTR341112_256_WITH_MAGMA_MGM_S
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256
TLS_PSK_WITH_CHACHA20_POLY1305_SHA256
TLS_ECDHE_PSK_WITH_CHACHA20_POLY1305_SHA256
TLS_DHE_PSK_WITH_CHACHA20_POLY1305_SHA256
TLS_RSA_PSK_WITH_CHACHA20_POLY1305_SHA256
TLS_ECDHE_PSK_WITH_AES_128_GCM_SHA256
TLS_ECDHE_PSK_WITH_AES_256_GCM_SHA384
TLS_ECDHE_PSK_WITH_AES_128_CCM_8_SHA256
TLS_ECDHE_PSK_WITH_AES_128_CCM_SHA256

IANA Cipher list

Attacks on TLS (non-exhaustive list)

- 1998, 2006: Bleichenbacher — breaks RSA encryption and RSA signatures using errors as side-channel
- 2011: BEAST — breaks SSL 3.0 and TLS 1.0 (nobody was using TLS 1.1 (2006) or 1.2 (2008)…)
  - avoid attack by using RC4 (but since 2013 RC4 is considered ☠…)
- 2012/2013: CRIME / BREACH — compression in TLS is bad
- 2013: Lucky Thirteen — timing attack on encrypt-then-MAC
- 2014: POODLE — destroys SSL 3.0
- 2014: Bleichenbacher again (BERserk) — signature forgery
- 2015/2016: FREAK / Logjam
  - implementation flaws downgrade to EXPORT cryptography
- 2016: DROWN — use the server’s SSLv2 support to break SSLv3/TLS 1.{0,1,2}
- 2018: ROBOT — Bleichenbacher’s 1998 attack is still valid on many TLS 1.2 implementations
- 2023: Everlasting ROBOT — Bleichenbacher’ s 1998 attack is still, still valid on many TLS 1.2 implementations

Downgrade Attack (FREAK) — 2015



1. Client random
Supported curves
Supported ciphers

2. Client random
Supported curves
Export ciphers (40 bit key)

3. Server random
Export ciphers
Certificate
Server key exchange
Signature

4. Alice generate
weak shared key

5. Cipher key exchange
Change cipher spec
Finished message (original)

6. Brute force the
2^{40} keys to crack

7. Cipher key exchange
Change cipher spec
Finished message (modified)

8. Server generate
weak shared key

9. Cipher key exchange
Change cipher spec
Finished message (modified)

10. Change cipher spec
Finished message (original)

ALL TRAFFIC BEYOND THIS POINT IS ENCRYPTED WITH WEAK SHARED KEYS.
Mallory can read/modify all messages between Alice and the Server.

Read more:
IEEE S&P 2015
BBDFKPSZ

TLS 1.3 (rfc8446) — 2018

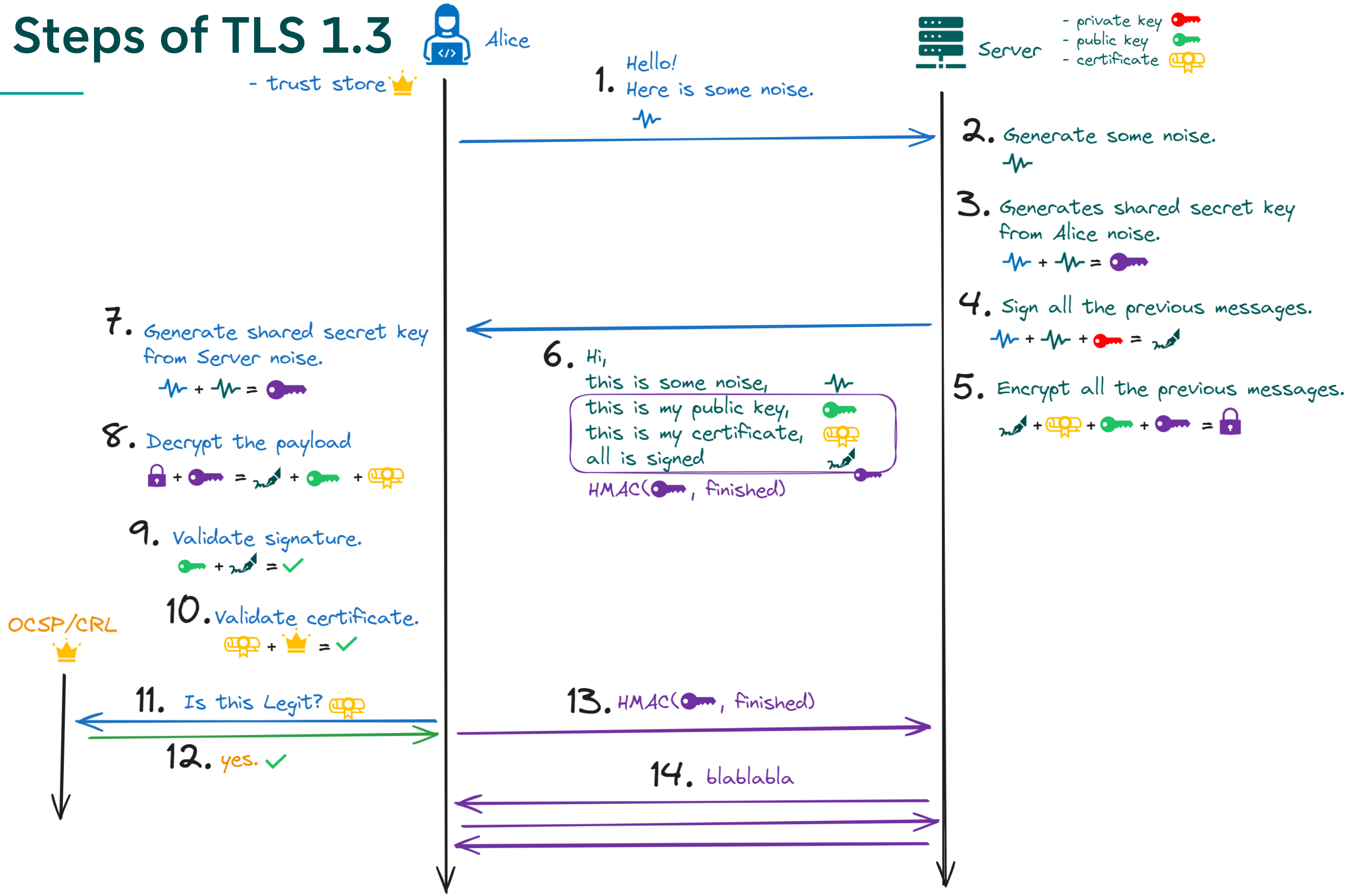
Faster — 1 round trip.

More private — Encrypt as much as possible.

Safer — Key exchange with ECDHE

Simpler — Only AES-GCM or ChaCha20-Poly1305

Basic Steps of TLS 1.3



1. Hello! Here is some noise.

2. Generate some noise.

3. Generates shared secret key from Alice noise.

4. Sign all the previous messages.

5. Encrypt all the previous messages.

6. Hi,
this is some noise,
this is my public key,
this is my certificate,
all is signed
HMAC(, finished)

7. Generate shared secret key from Server noise.

8. Decrypt the payload

9. Validate signature.

10. Validate certificate.

11. Is this Legit?

12. yes. ✓

13. HMAC(, finished)

14. blablabla

Certificate Revocation List

- rfc5280

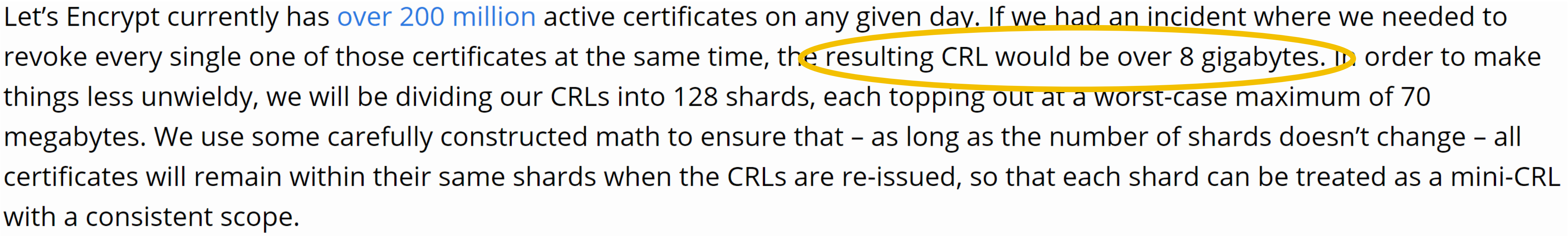
- **List of all the currently revoked certificates (by serial number).**
  - Do not include expired certs.
  - Include reason of revocation (not that we really care)

- **Downloaded once by the client.**
  - Allow offline verification

- **Published & Signed by the Certificate Authority.**
  - Complex process in certain situations (see later).

Certificate Revocation List

One *big* problem: Size.



Let’s Encrypt currently has over 200 million active certificates on any given day. If we had an incident where we needed to revoke every single one of those certificates at the same time, the resulting CRL would be over 8 gigabytes. In order to make things less unwieldy, we will be dividing our CRLs into 128 shards, each topping out at a worst-case maximum of 70 megabytes. We use some carefully constructed math to ensure that – as long as the number of shards doesn’t change – all certificates will remain within their same shards when the CRLs are re-issued, so that each shard can be treated as a mini-CRL with a consistent scope.

Online Certificate Status Protocol (OCSP)

- rfc5019

- Client ask the CA for the certificate status with a signed response.

- Response has a max-age.

- Client must cache the response to minimize bandwidth usage.

Online Certificate Status Protocol (OCSP)

Problems:

- Privacy issues
- Latency cost
- Only works for online clients/servers, not *e.g.* cars
- Single point of availability/failure

High-profile failure in 2020 for Apple developer certs.

Used OCSP for code signing certs: couldn’t launch desktop apps!

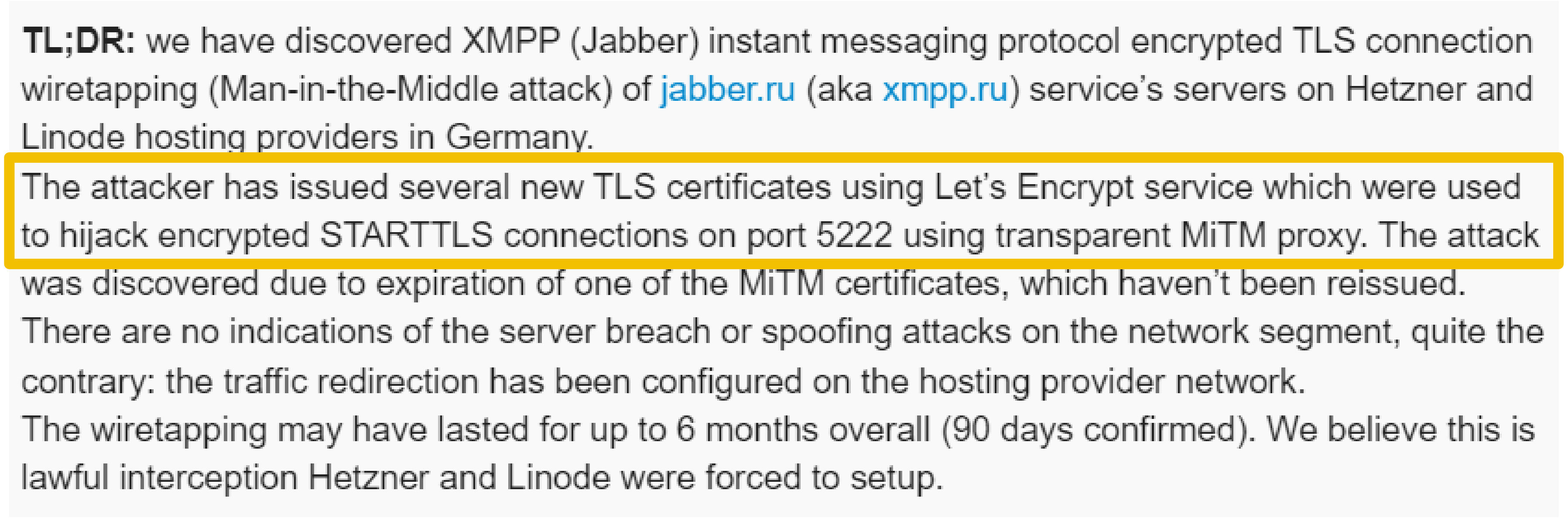
Solution: OCSP Stapling

Server does the OCSP request, attach the response from the CA, and forwards it to the client.

- **OCSP response can be cached by the server.**
- **Short lived (max 7 days).**
- **Privacy friendly (client no longer does the request).**
- **Resilient to OCSP server outage.**
- **But stricter behaviour on client side.**
  - e.g., Firefox reject OCSP stapling for bad stapled response but does not fail for bad OCSP responses.

Problem: Any CA can sign for anyone.

Your CA is not the only one that can issue certificates for your domain…

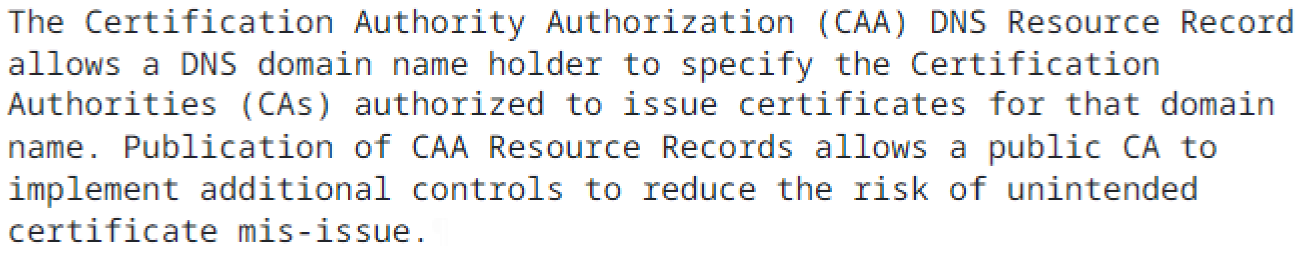


TL;DR: we have discovered XMPP (Jabber) instant messaging protocol encrypted TLS connection wiretapping (Man-in-the-Middle attack) of jabber.ru (aka xmpp.ru) service's servers on Hetzner and Linode hosting providers in Germany.
The attacker has issued several new TLS certificates using Let's Encrypt service which were used to hijack encrypted STARTTLS connections on port 5222 using transparent MiTM proxy. The attack was discovered due to expiration of one of the MiTM certificates, which haven't been reissued.
There are no indications of the server breach or spoofing attacks on the network segment, quite the contrary: the traffic redirection has been configured on the hosting provider network.
The wiretapping may have lasted for up to 6 months overall (90 days confirmed). We believe this is lawful interception Hetzner and Linode were forced to setup.

Source: https://community.letsencrypt.org/t/presumed-gov-mitm-discovered-due-to-expired-le-certs/206966
Read more: https://notes.valdikss.org.ru/jabber.ru-mitm/

CAA & Certificate Transparency

- CAA record in DNS records (says who can issue certificate for a domain) — rfc8659.

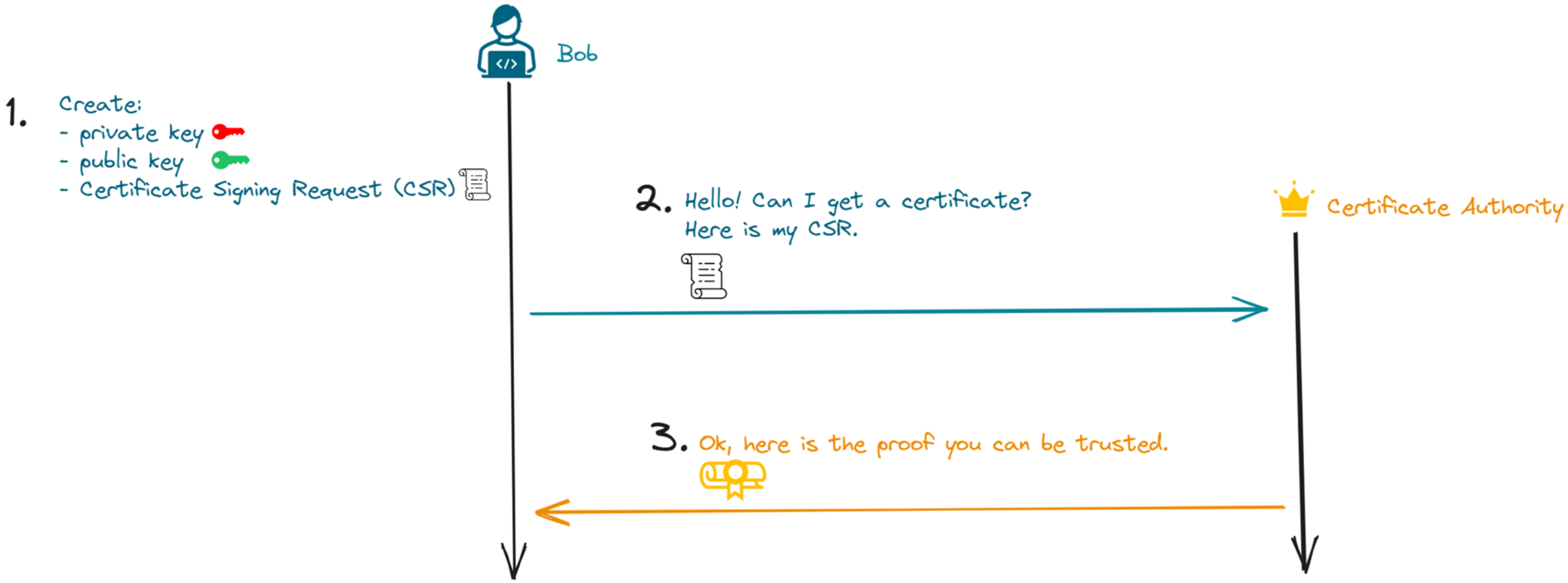


The Certification Authority Authorization (CAA) DNS Resource Record allows a DNS domain name holder to specify the Certification Authorities (CAs) authorized to issue certificates for that domain name. Publication of CAA Resource Records allows a public CA to implement additional controls to reduce the risk of unintended certificate mis-issue.

- Public log for all issued certificates (Certificate Transparency) — rfc9162.
  Not applicable within ABN AMRO.

  See Thom’s talk in a few weeks. 😊

How to get a certificate?


Bob
1.
Create:
- private key
- public key
- Certificate Signing Request (CSR)
2.
Hello! Can I get a certificate?
Here is my CSR.
Certificate Authority
3.
Ok, here is the proof you can be trusted.

"Trust but verify."

Establishing trust with the CA

1. **Are we sure we are talking to the right domain?**

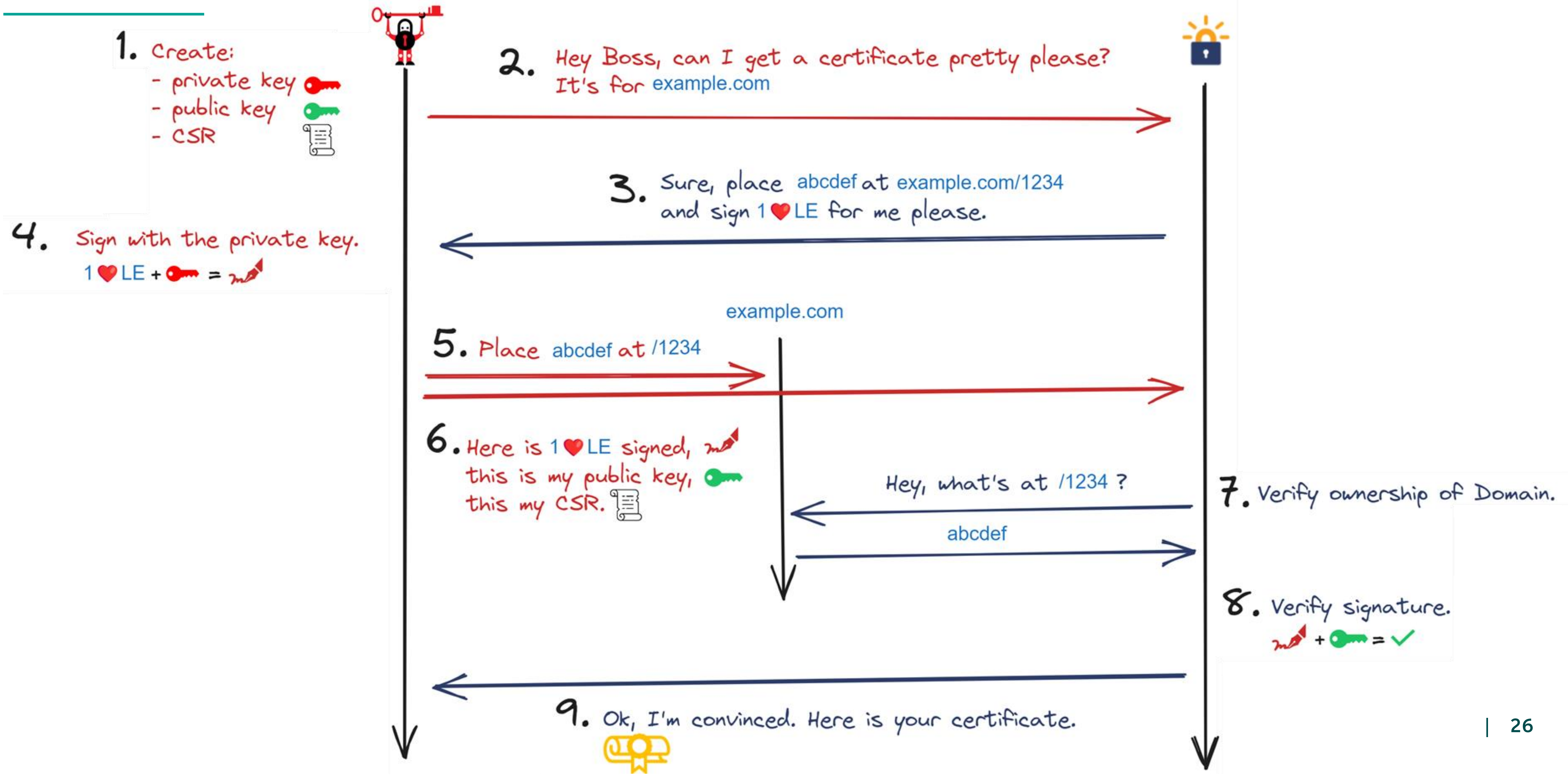
2. **Are we sure that the domain is in possession of the private key?**

Solution: ACME* Protocol

Most well-known implementation: Certbot

*Automatic Certificate Management Environment
Read More: rfc 8555

Getting an https certificate with Let's Encrypt



1. Create:
- private key 🔑
- public key 🔑
- CSR 📜

2. Hey Boss, can I get a certificate pretty please? It's for example.com

3. Sure, place abcdef at example.com/1234 and sign 1 ❤ LE for me please.

4. Sign with the private key.
1 ❤ LE + 🔑 = ✍

5. Place abcdef at /1234

6. Here is 1 ❤ LE signed, ✍
this is my public key, 🔑
this my CSR. 📜

Hey, what's at /1234 ?

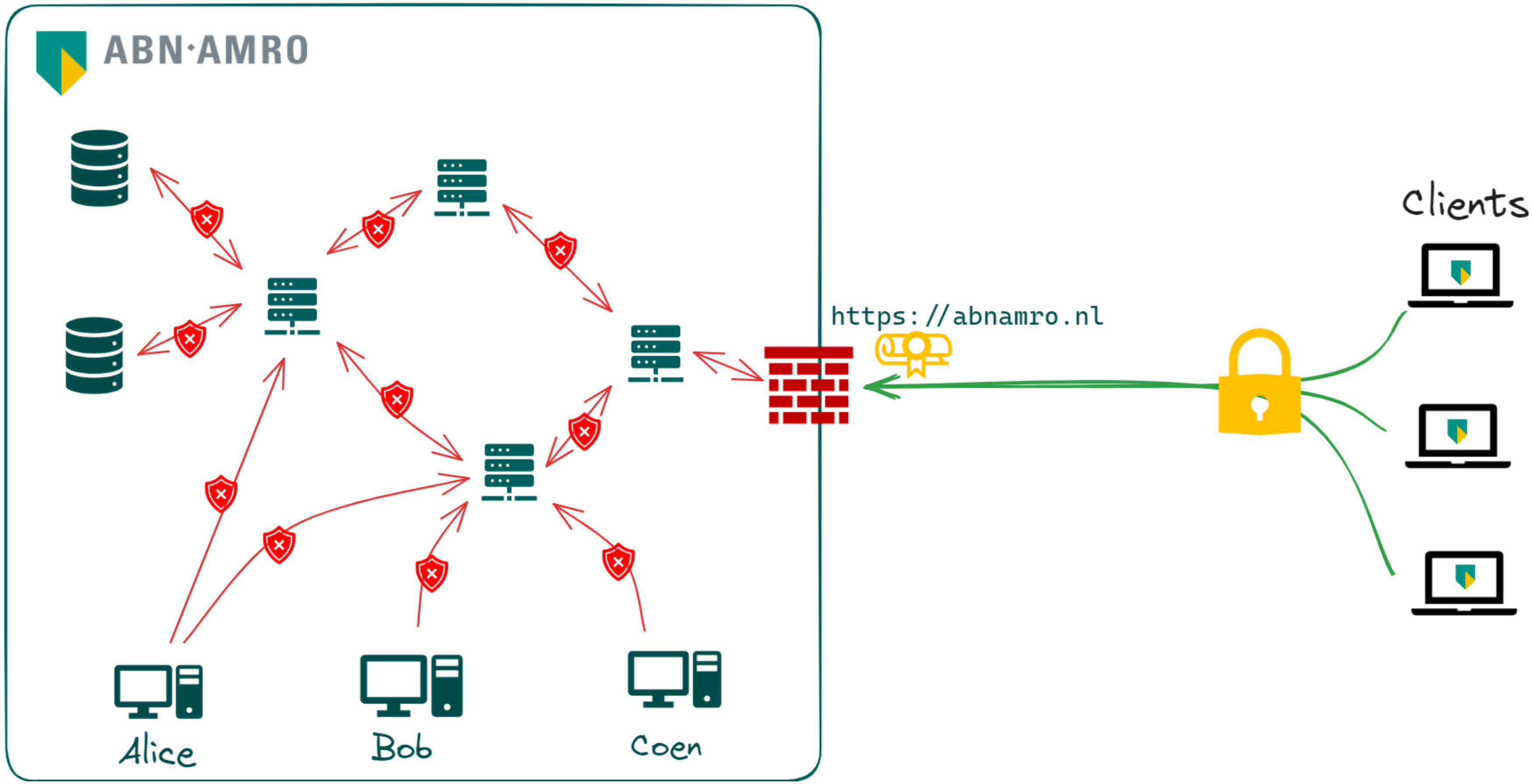
abcdef

7. Verify ownership of Domain.

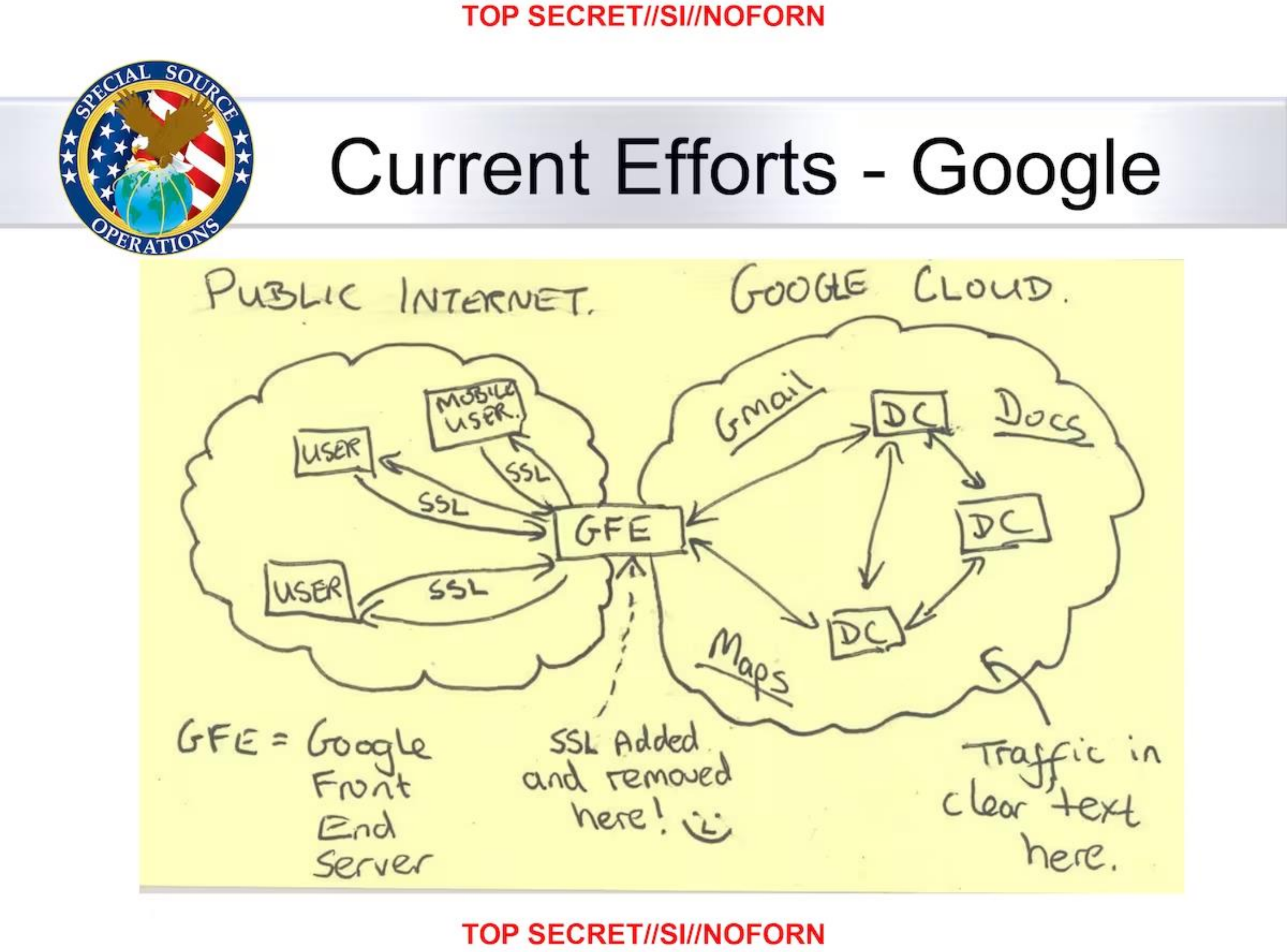
8. Verify signature.
✍ + 🔑 = ✓

9. Ok, I'm convinced. Here is your certificate.

Within ABN AMRO network



Project BULLRUN (2013)



TOP SECRET//SI//NOFORN

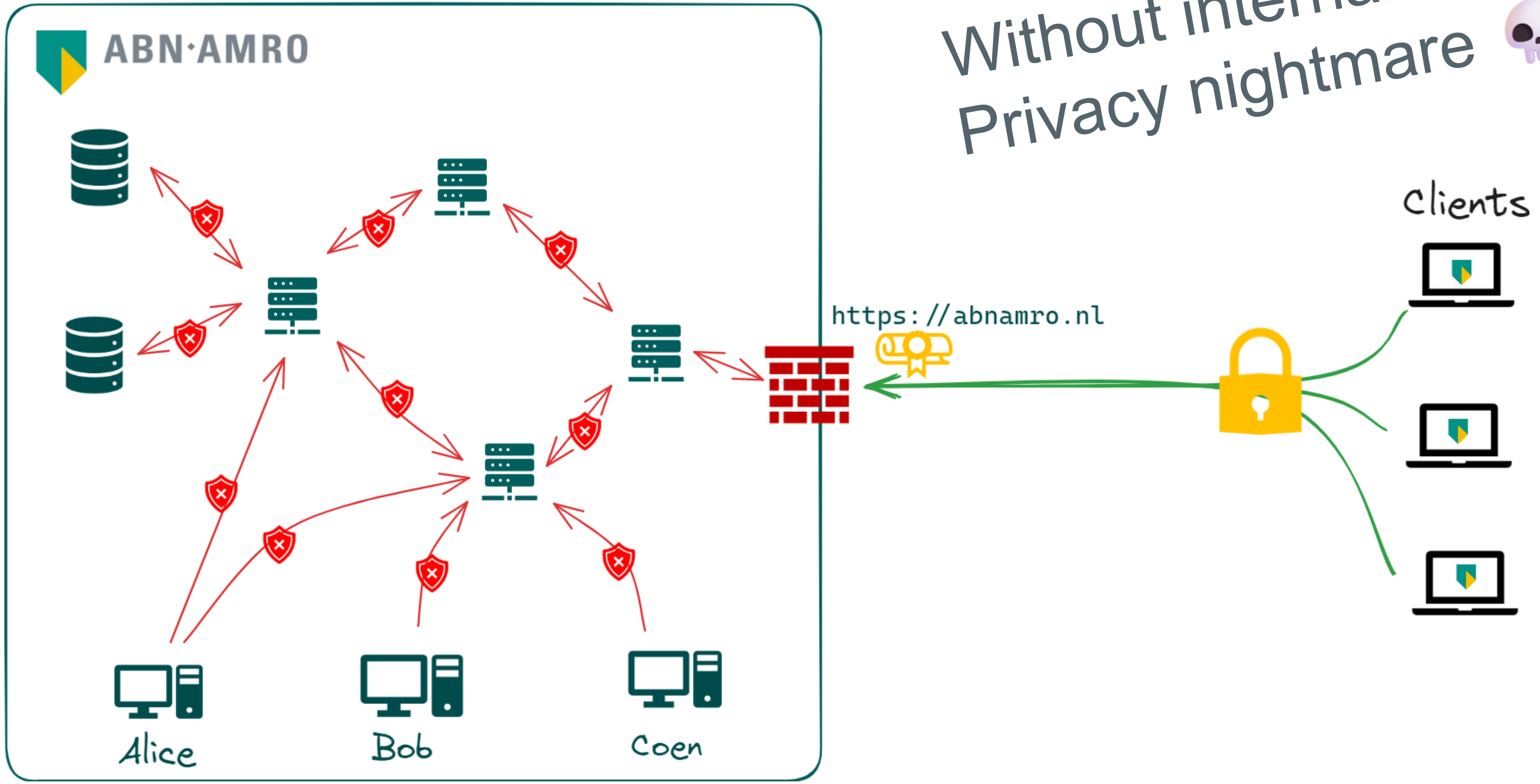
Current Efforts - Google

SPECIAL SOURCE OPERATIONS
PUBLIC INTERNET.
GOOGLE CLOUD.
MOBILE USER.
USER
USER
SSL
SSL
SSL
GFE
Gmail
DC
Docs
DC
DC
Maps
GFE = Google Front End Server
SSL Added and removed here! :)
Traffic in clear text here.

TOP SECRET//SI//NOFORN

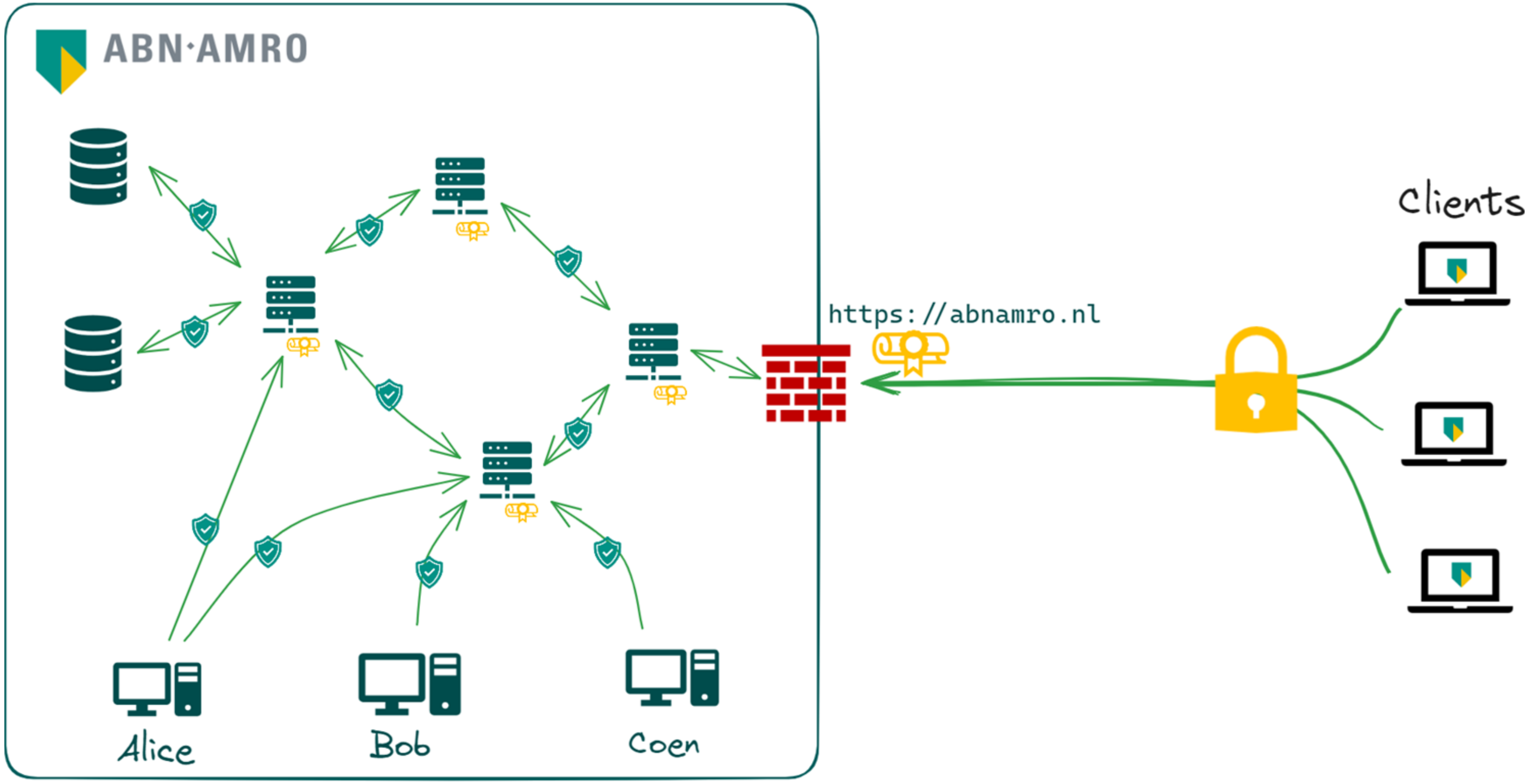
NSA infiltrates links to Yahoo, Google data centers worldwide

Within ABN AMRO network



Without internal HTTPS
Privacy nightmare 💀

Within ABN AMRO network



Trust level of Certificates

DV

Domain Validation

Basic

90~420* €
per annum

Or Free via Let's Encrypt

OV

Organization Validation

Standard

165~800*€
per annum

EV

Extended Validation

Premium

230~430€
per annum

*For wildcard certificates.

2 Certificate Management at ABN AMRO

Why can’t we use Let’s Encrypt at ABN AMRO

- Very complex network (with multiple NAT).
- Would need to pass through the Firewalls.
- Exposes our internal hostnames…
- A lot of our services are internal only.
- Legal requires us to use OV certificate (LE only provides DV).

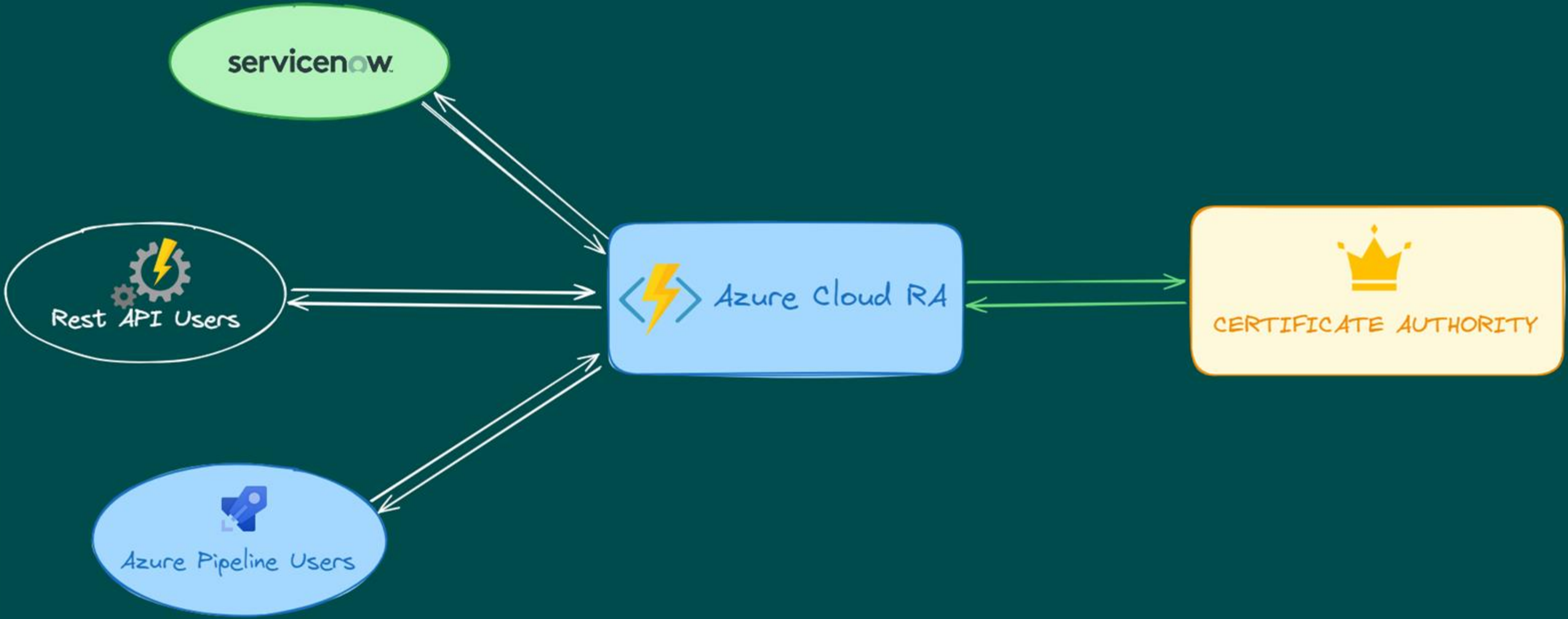
Managing certificates in a bank... (old)

Each team is responsible for their own certificate.

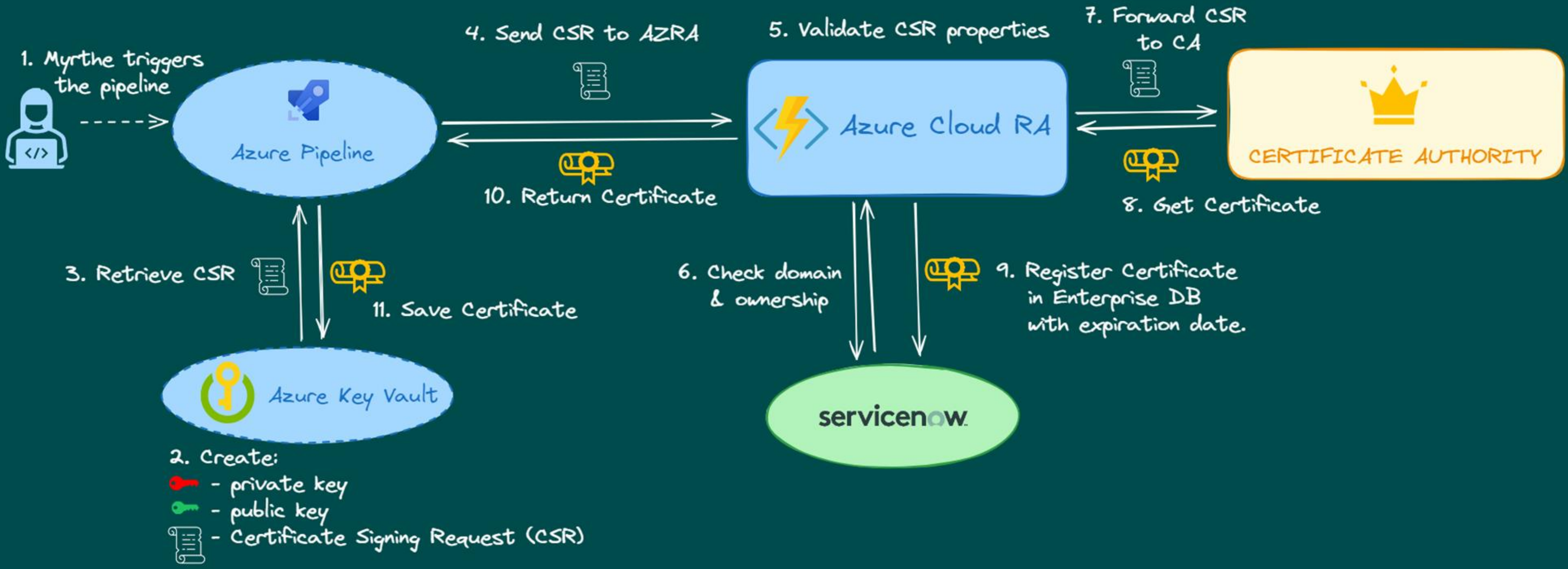
Crypto Services **manually** sends expiration reminder mails.

Crypto Service **manually** manages the access to the Certificate Portal.

Moving to Azure Cloud Registration Authority



Certificate Request Process



Some of the validation rules applied.

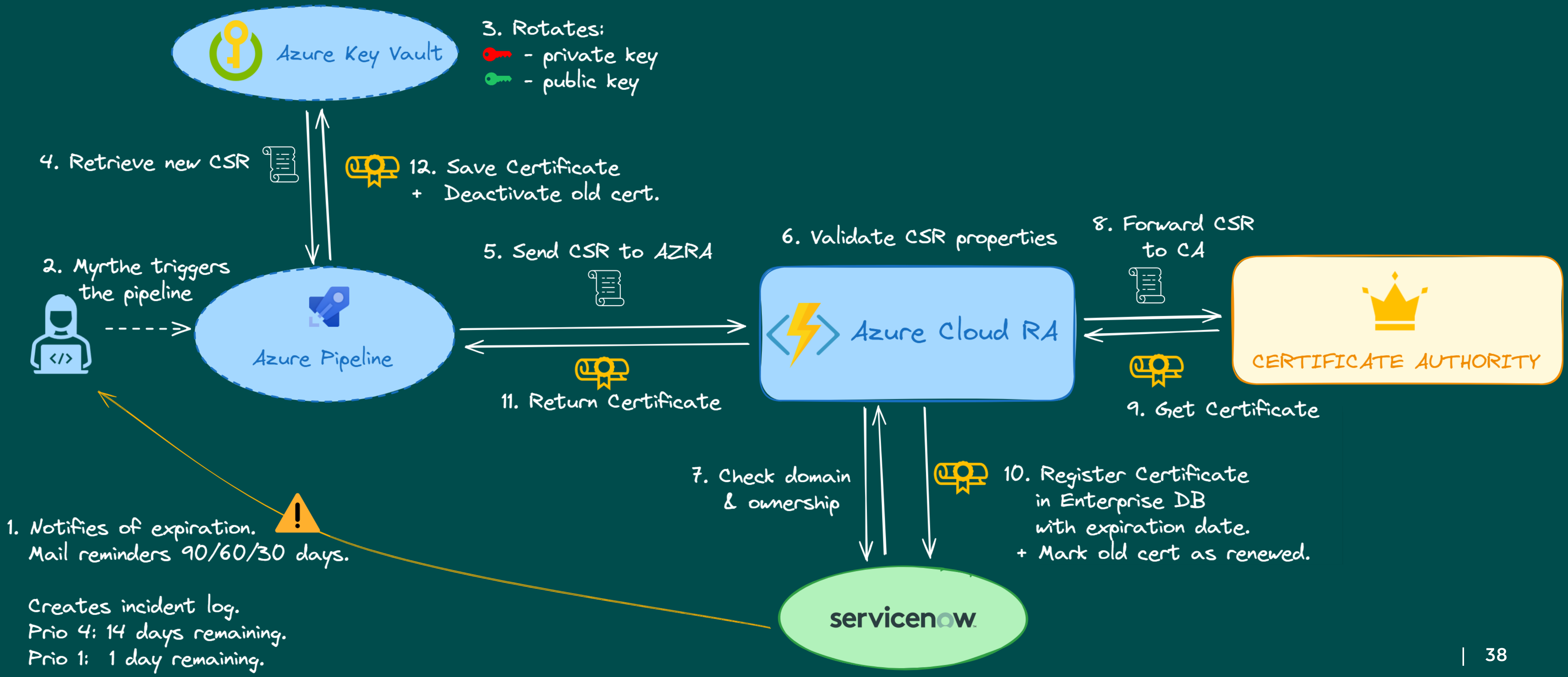
- **CN must follow a regex pattern.**
  - e.g. [A-Za-z0-9]+.abnamro.nl

- **Wildcard are forbidden.**
  - No *.abnamro.nl

- **Algorithm limitations.**
  - RSA key size must be between 2048 and 4096 bits.
  - ECC field size must be greater than 256 bits: P384 or P521.
  - No SHA1 (and no MD5, duh...)

- **SAN fields must not contain different environment.**
  - frontend-test.abnamro.nl and frontend-acceptance.abnamro.nl cannot use the same certificate.

Certificate Renewal Process



Azure Key Vault

3. Rotates:
🟥 - private key
🟩 - public key

4. Retrieve new CSR

12. Save Certificate
+ Deactivate old cert.

2. Myrthe triggers
the pipeline

Azure Pipeline

5. Send CSR to AZRA

11. Return Certificate

6. Validate CSR properties

Azure Cloud RA

8. Forward CSR
to CA

CERTIFICATE AUTHORITY

9. Get Certificate

7. Check domain
& ownership

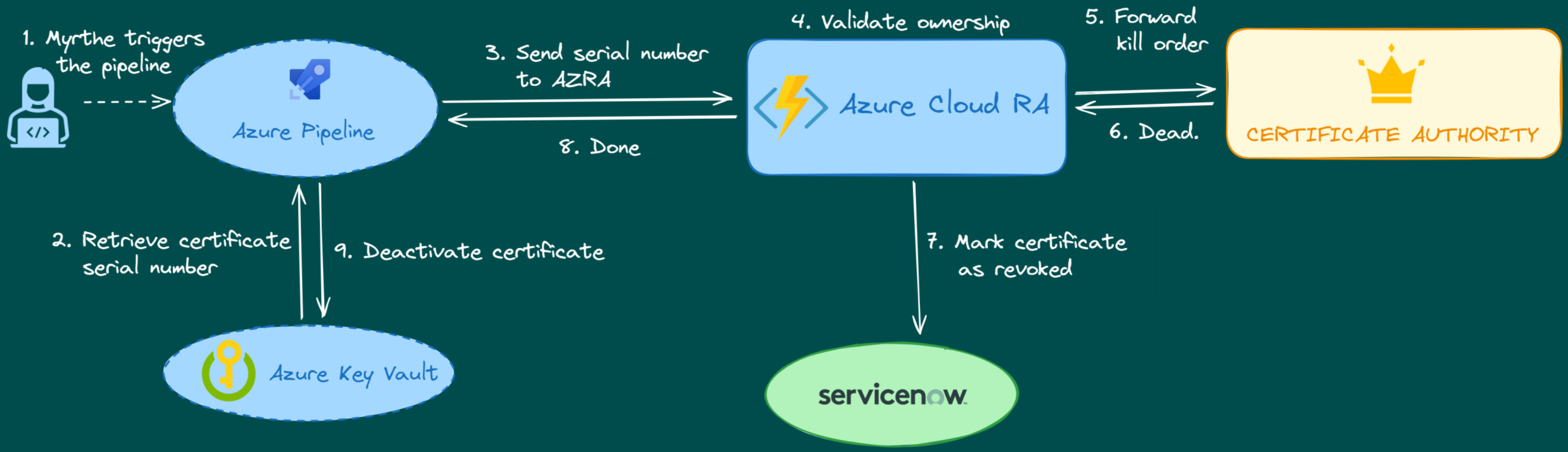
10. Register Certificate
in Enterprise DB
with expiration date.
+ Mark old cert as renewed.

servicenow

1. Notifies of expiration.
Mail reminders 90/60/30 days.

Creates incident log.
Prio 4: 14 days remaining.
Prio 1: 1 day remaining.

Certificate Revocation Process


1. Myrthe triggers the pipeline
Azure Pipeline
3. Send serial number to AZRA
8. Done
4. Validate ownership
Azure Cloud RA
5. Forward kill order
6. Dead.
CERTIFICATE AUTHORITY
2. Retrieve certificate serial number
9. Deactivate certificate
7. Mark certificate as revoked
Azure Key Vault
servicenow

24 000+

active certificates
within ABN AMRO BANK

Certificates per key type

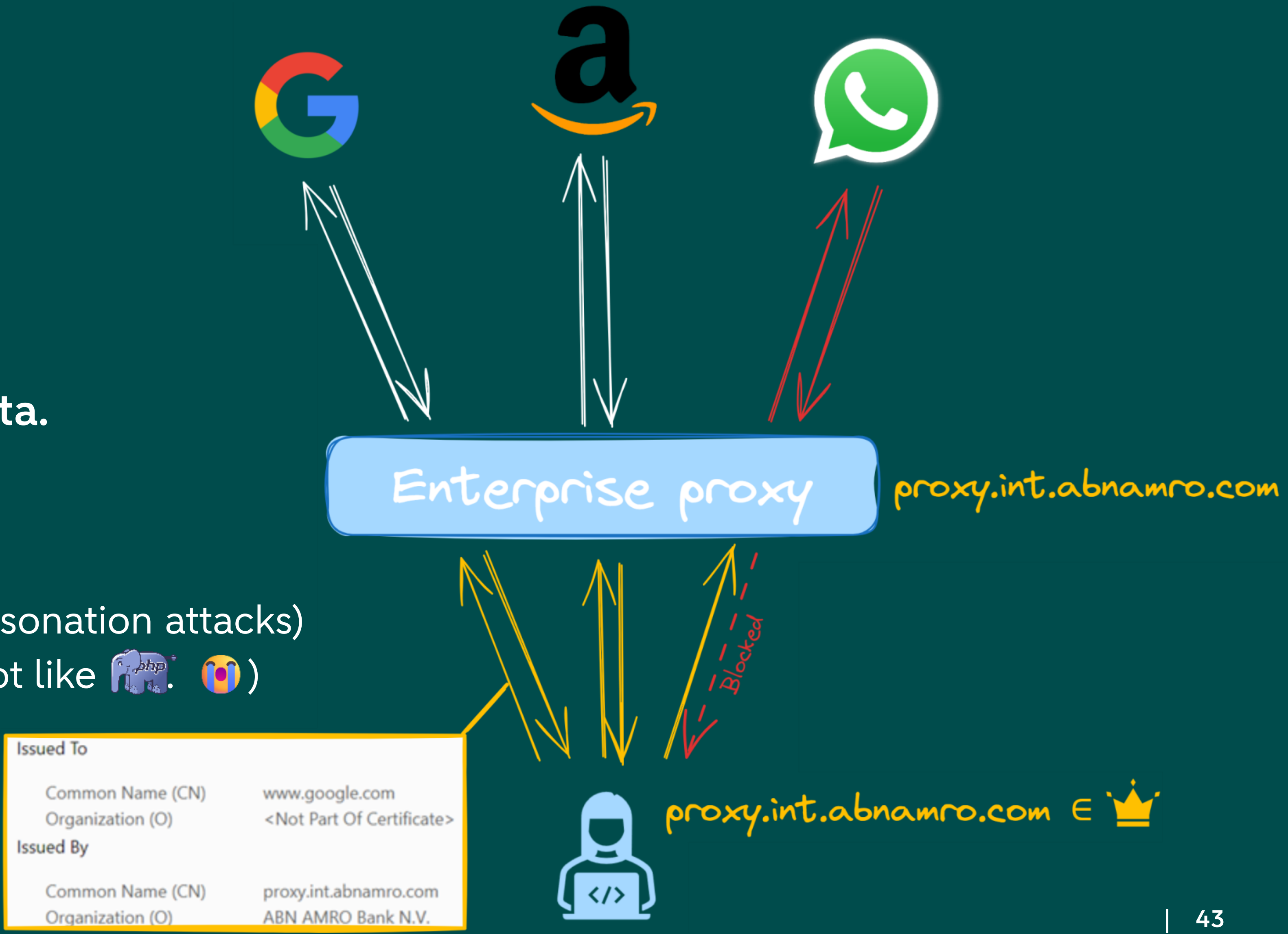
ECC 384:	25 (0,1%)
RSA 2048:	11 671 (48%)
RSA 4096:	12 592 (52%)

3 Certificate Authorities

ABN·AMRO

MITM & Enterprise proxy

- Intercept all traffic In-&-Out.
- Deep packet inspection.
- Verify non-exfiltration of PII data.
- Block some websites, *e.g.*:
    - LinkedIn (avoid phishing/impersonation attacks)
    - laravel-news.com (AAB does not like php. 😭)
    - WhatsApp…



Issued To

Common Name (CN)	www.google.com
Organization (O)	<Not Part Of Certificate>

Issued By

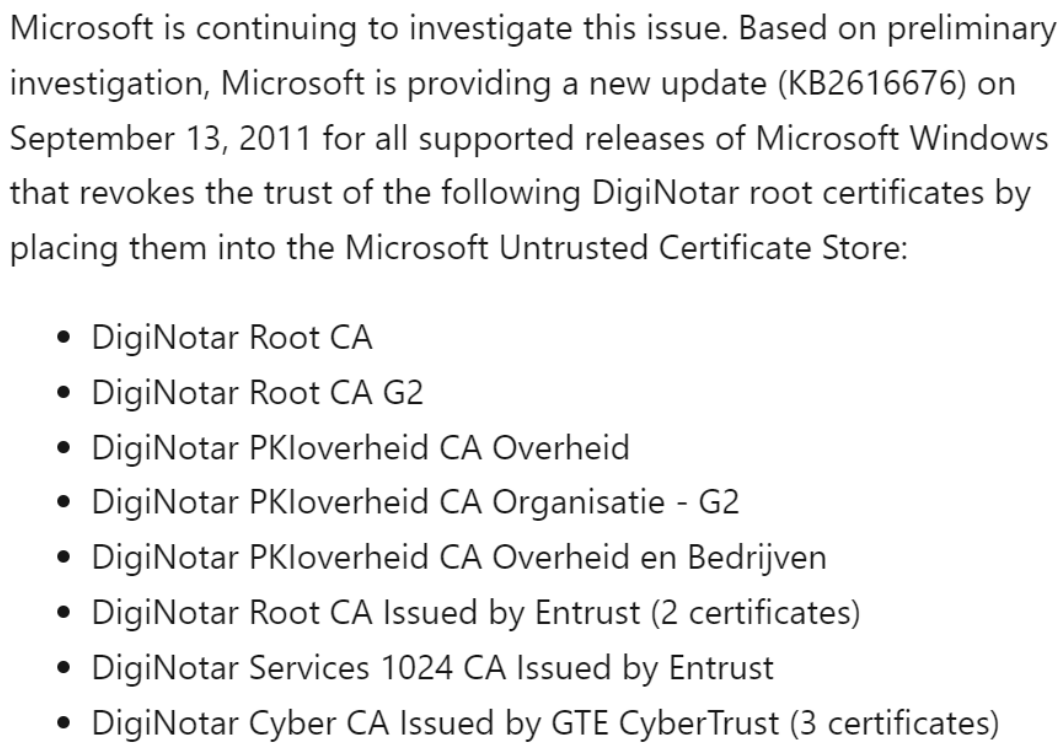
Common Name (CN)	proxy.int.abnamro.com
Organization (O)	ABN AMRO Bank N.V.

CA Compromise: DigiNotar

- Fully compromised in July 2011
- Had to be removed from all Trust stores.
- Issued compromised certificates for e.g. google.com
- Notable use to spy on Gmail users in Iran.

Read more:

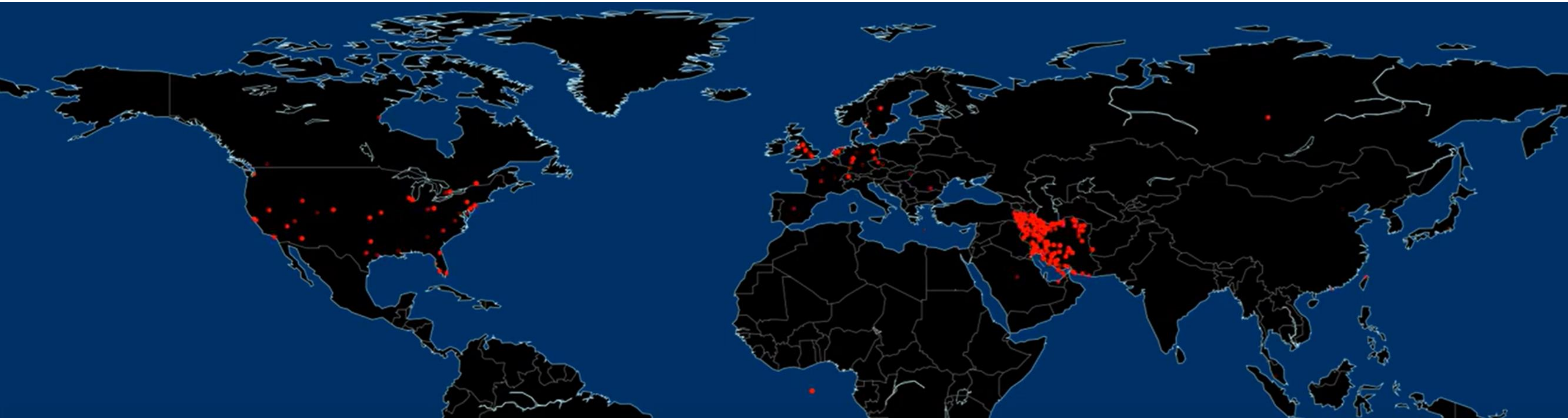
- https://blog.gerv.net/2011/09/diginotar-compromise/
- Blog mozilla
- MS security advisories



Microsoft is continuing to investigate this issue. Based on preliminary investigation, Microsoft is providing a new update (KB2616676) on September 13, 2011 for all supported releases of Microsoft Windows that revokes the trust of the following DigiNotar root certificates by placing them into the Microsoft Untrusted Certificate Store:

- DigiNotar Root CA
- DigiNotar Root CA G2
- DigiNotar PKIoverheid CA Overheid
- DigiNotar PKIoverheid CA Organisatie - G2
- DigiNotar PKIoverheid CA Overheid en Bedrijven
- DigiNotar Root CA Issued by Entrust (2 certificates)
- DigiNotar Services 1024 CA Issued by Entrust
- DigiNotar Cyber CA Issued by GTE CyberTrust (3 certificates)

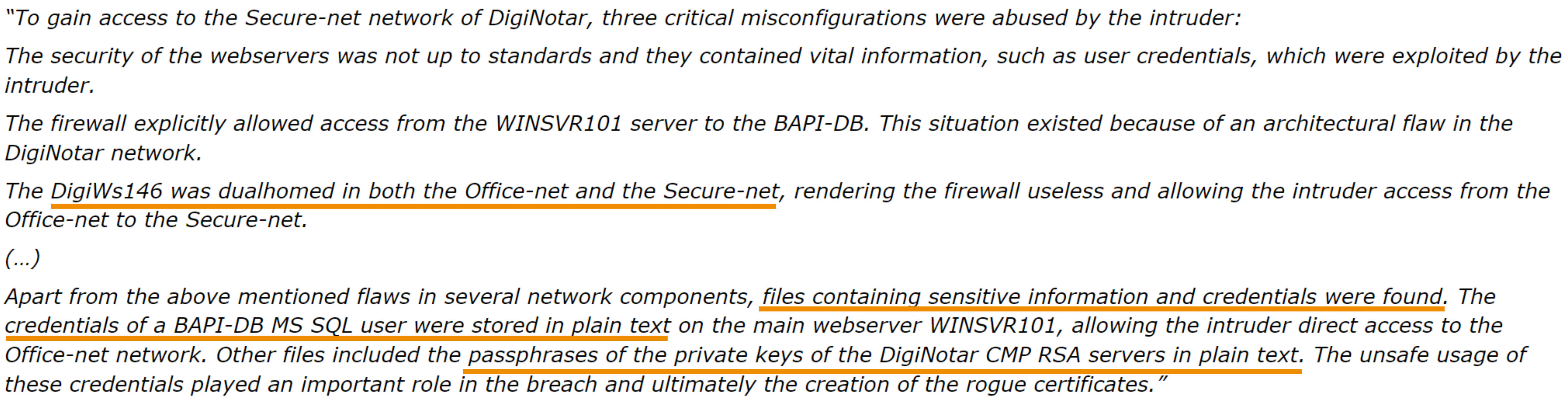
CA Compromise: DigiNotar



OCSP requests for the rogue *.google.com certificate

Source: https://www.youtube.com/watch?v=wZsWoSxxwVY

CA Compromise: DigiNotar



"To gain access to the Secure-net network of DigiNotar, three critical misconfigurations were abused by the intruder:

The security of the webservers was not up to standards and they contained vital information, such as user credentials, which were exploited by the intruder.

The firewall explicitly allowed access from the WINSVR101 server to the BAPI-DB. This situation existed because of an architectural flaw in the DigiNotar network.

The DigiWs146 was dualhomed in both the Office-net and the Secure-net, rendering the firewall useless and allowing the intruder access from the Office-net to the Secure-net.

(...)

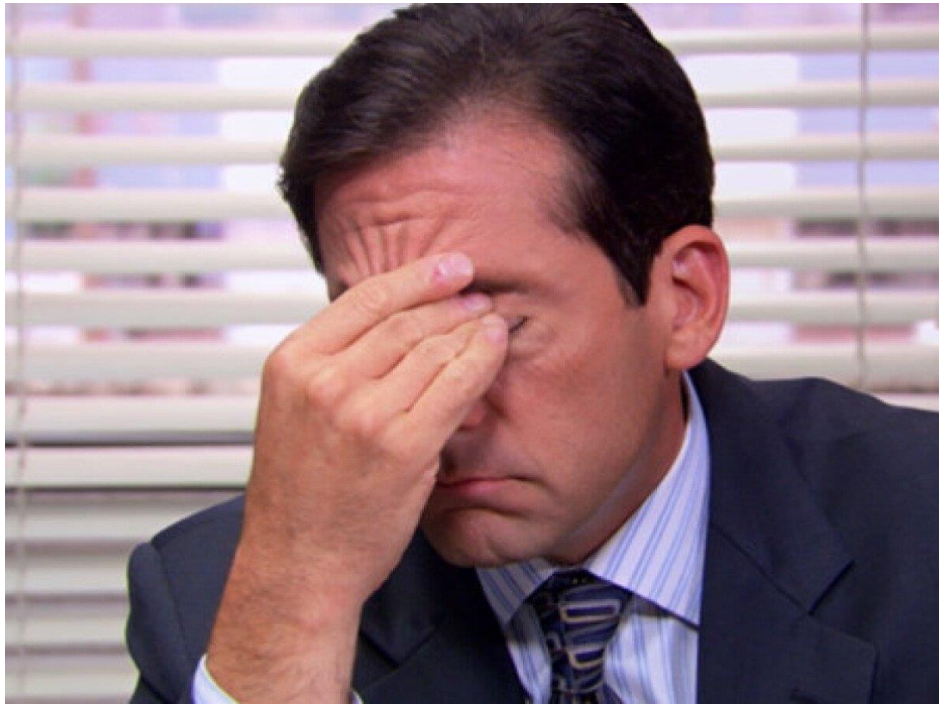
Apart from the above mentioned flaws in several network components, files containing sensitive information and credentials were found. The credentials of a BAPI-DB MS SQL user were stored in plain text on the main webserver WINSVR101, allowing the intruder direct access to the Office-net network. Other files included the passphrases of the private keys of the DigiNotar CMP RSA servers in plain text. The unsafe usage of these credentials played an important role in the breach and ultimately the creation of the rogue certificates."

Source: https://uitspraken.rechtspraak.nl/details?id=ECLI:NL:RBAMS:2014:4888

CA Compromise: DigiNotar

The BAD:

- Network/Firewall mis-configuration.
- Credentials stored in plaintext…
- Passphrases of private key stored in plaintext…
- Single-tier CA.

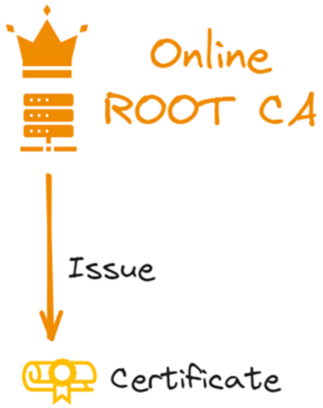


Read more:

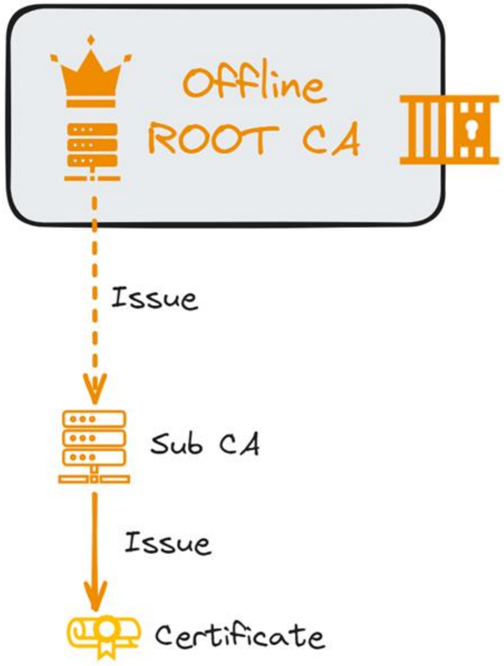
- https://roselabs.nl/files/audit_reports/Fox-IT_-_DigiNotar.pdf

Multi-tier Certificate Authority

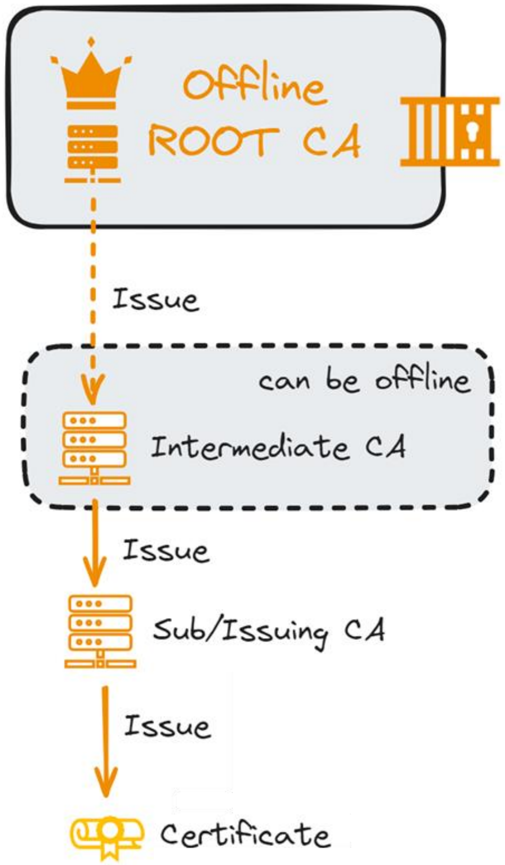
Single-tier hierarchy



Two-tier hierarchy

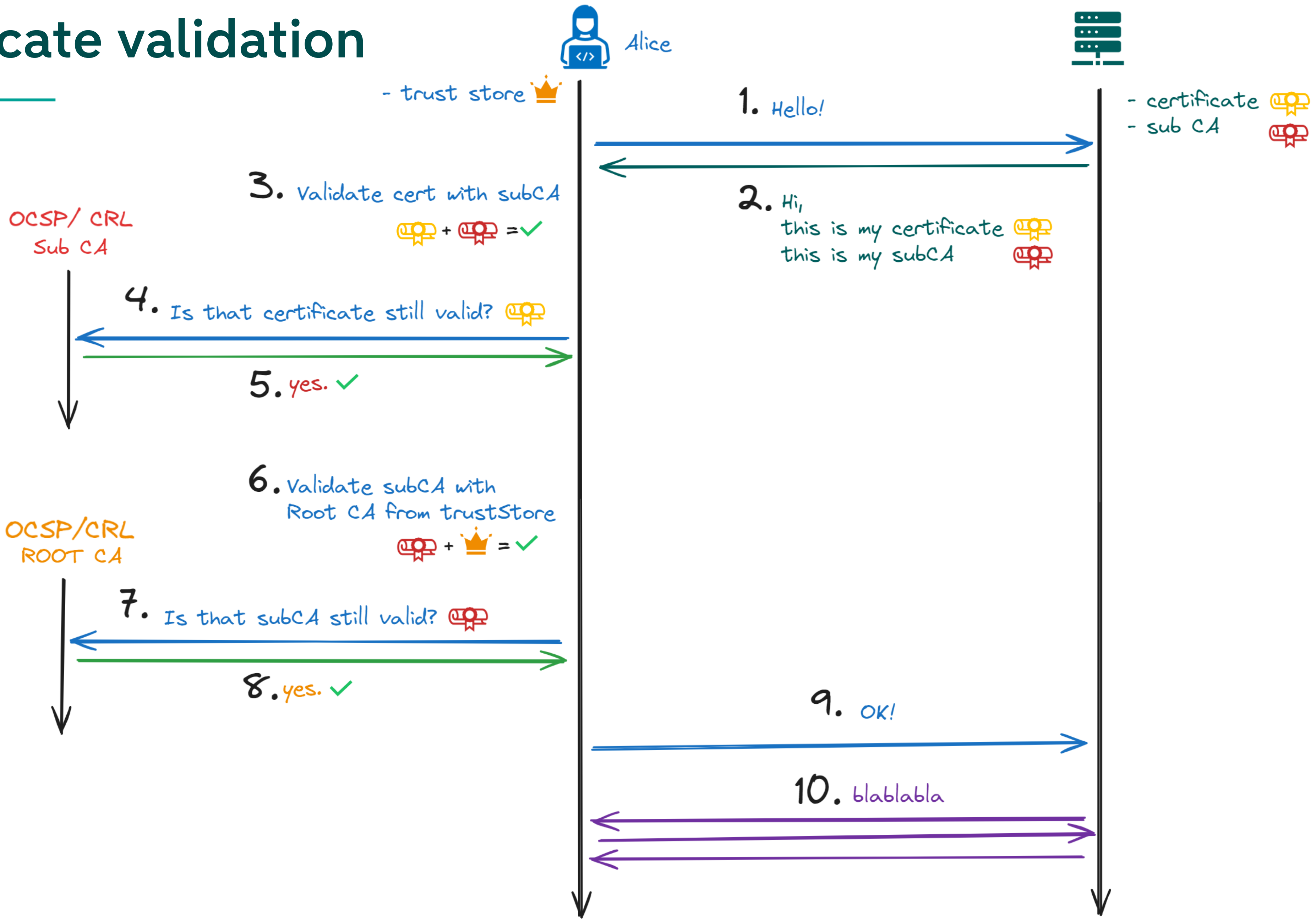


Three-tier hierarchy

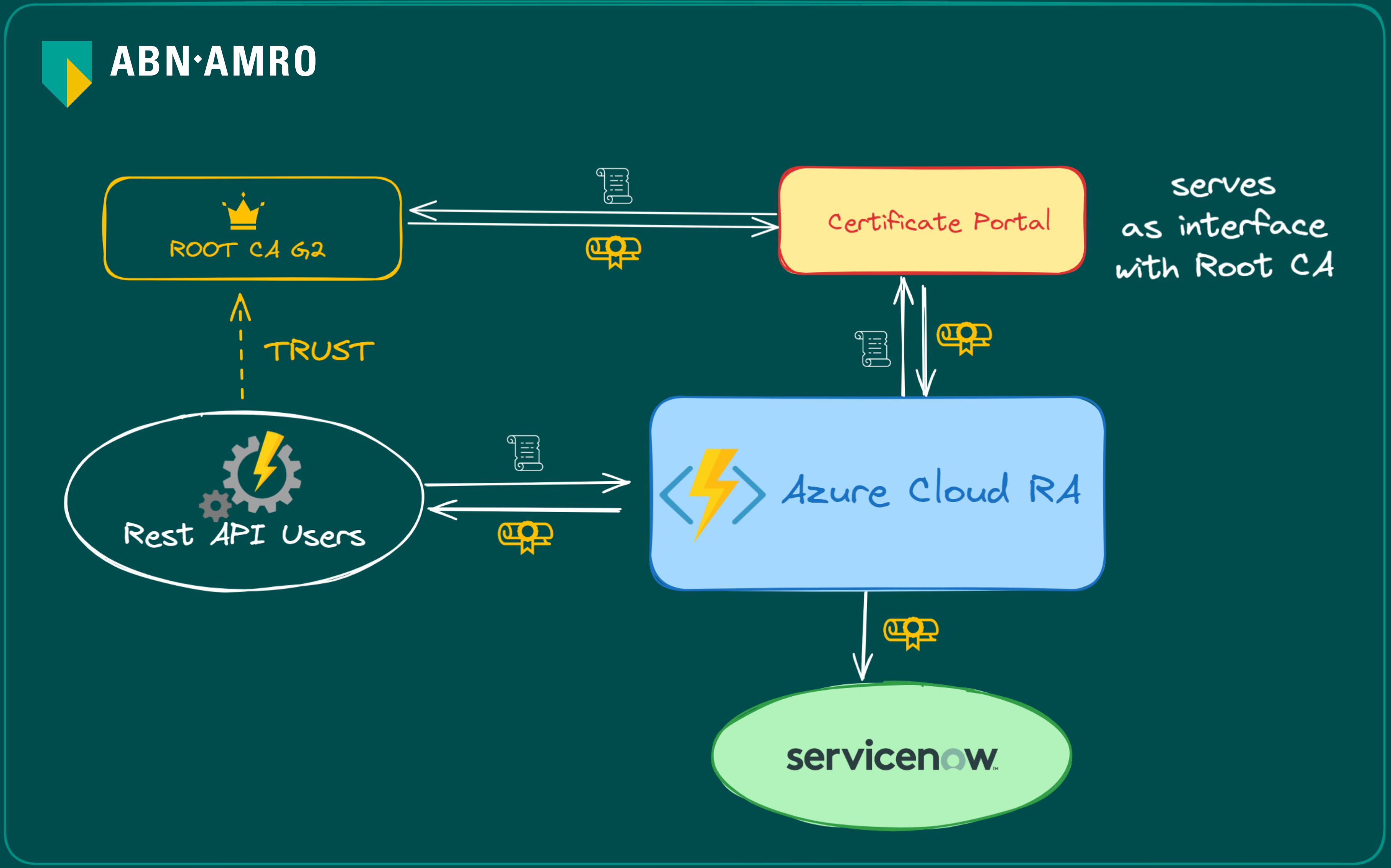


Offline = Air gapped

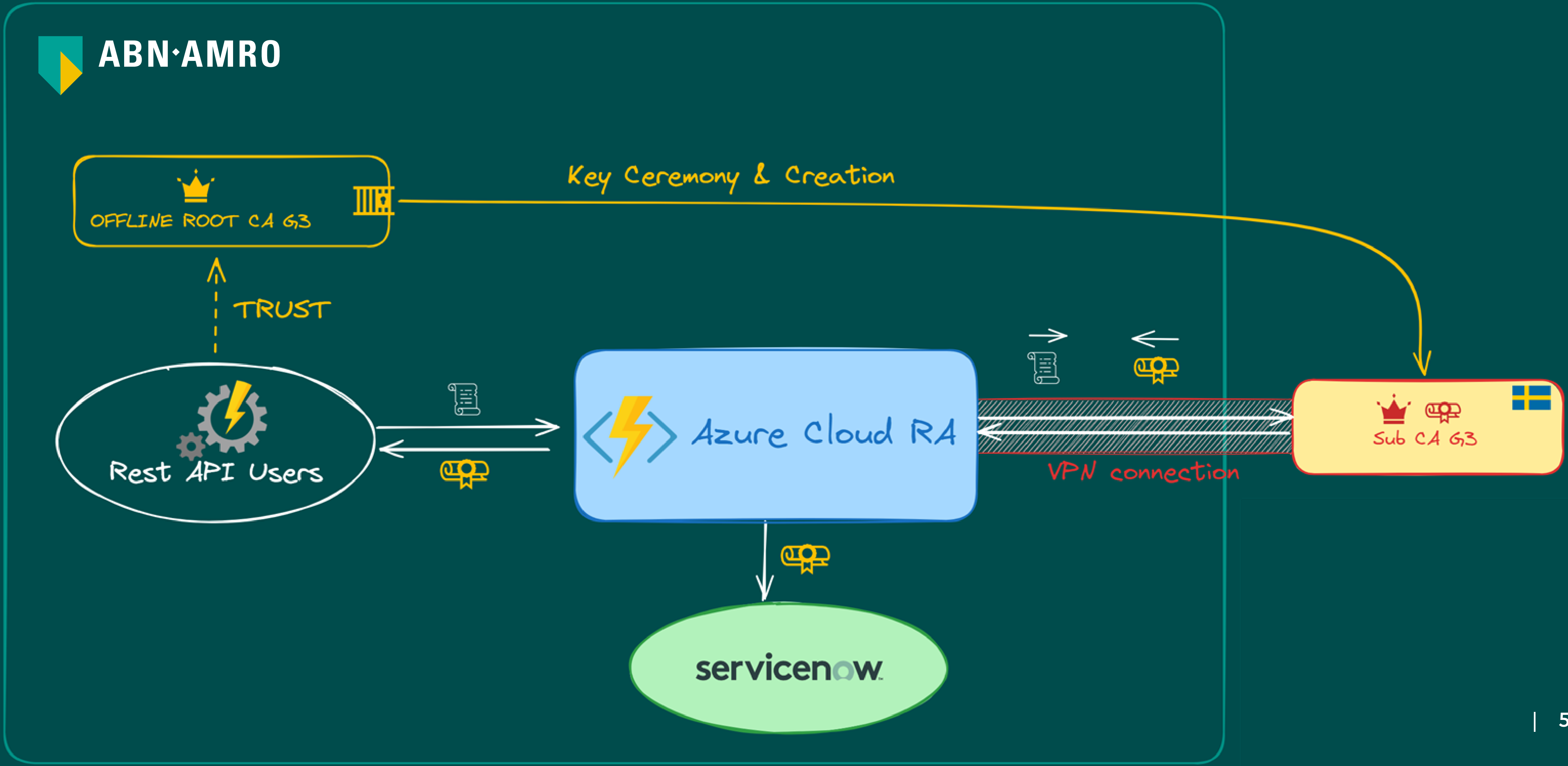
Certificate validation



Infra CA G2 — Single tier



Infra CA G3 — Two tier



Upgrade from G2 to G3 in Five Steps

01 Design of Infra CA G3

Update VPN connection
Network Security Framework
Order new HSM
Documentation

02 VPN Connection and installation

HSM installation for root CA in Netherlands
Connection via VPN to Sweden

03 Key Ceremony & Tests

ET and PR
Testing configuration G3

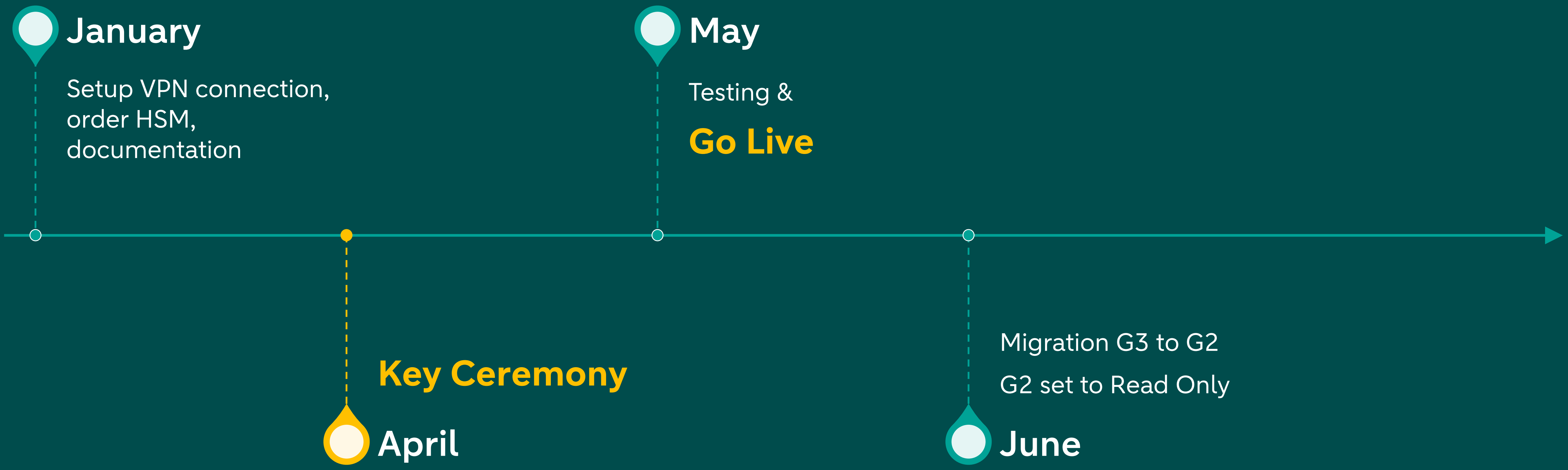
04 Go Live G3

New certificates issued by G3
Renewed certificates issued by G3, revoked in G2

05 Decom Infra G2

All certificates issued by G2 have been revoked
Decom VPN connections, retire old HSM

Planned Timeline of the migration G2 to G3 — April



January

Setup VPN connection,
order HSM,
documentation

Key Ceremony

April

May

Testing &

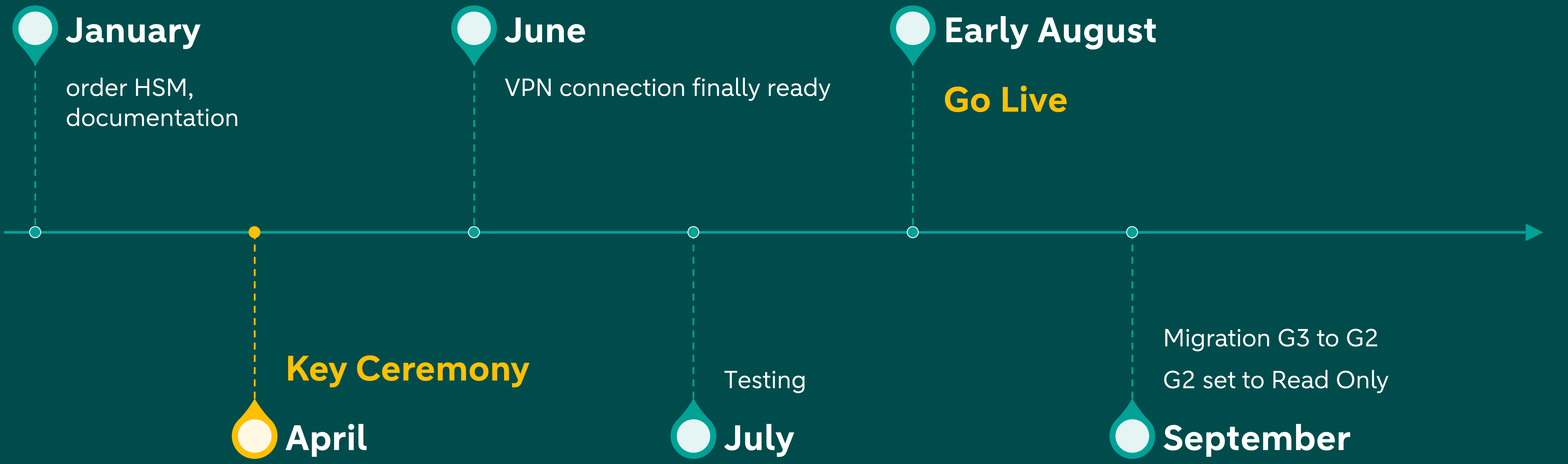
Go Live

Migration G3 to G2

G2 set to Read Only

June

Real timeline of the migration G2 to G3 — June



January

order HSM,
documentation

Key Ceremony

April

June

VPN connection finally ready

Testing

July

Early August

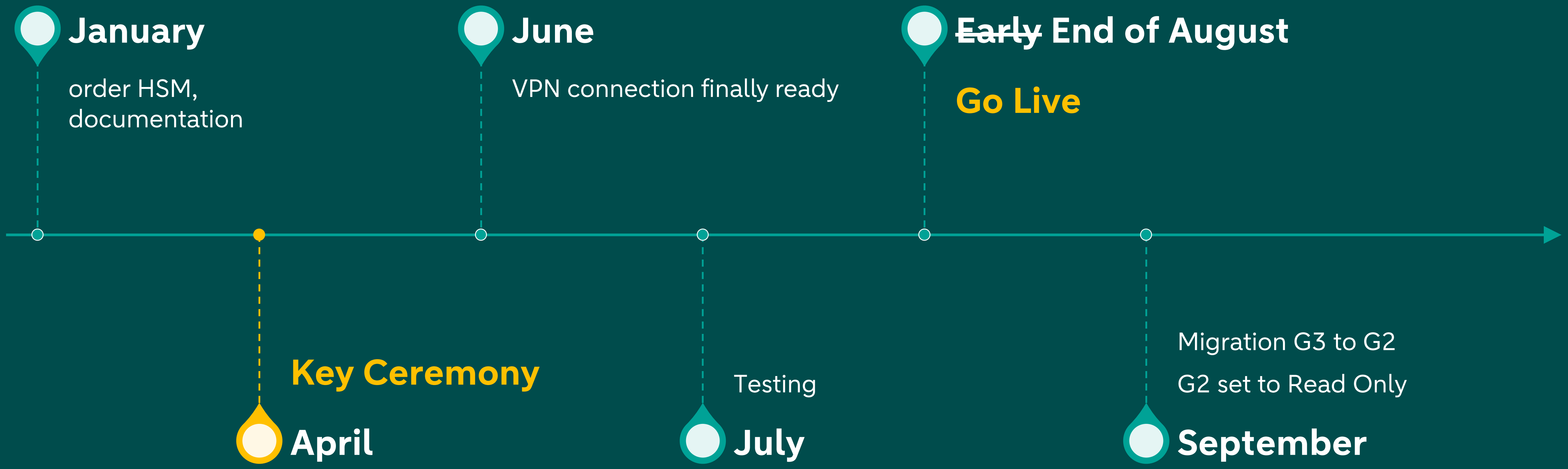
Go Live

Migration G3 to G2

G2 set to Read Only

September

Real timeline of the migration G2 to G3 — July



January

order HSM,
documentation

Key Ceremony

April

June

VPN connection finally ready

Testing

July

~~Early~~ End of August

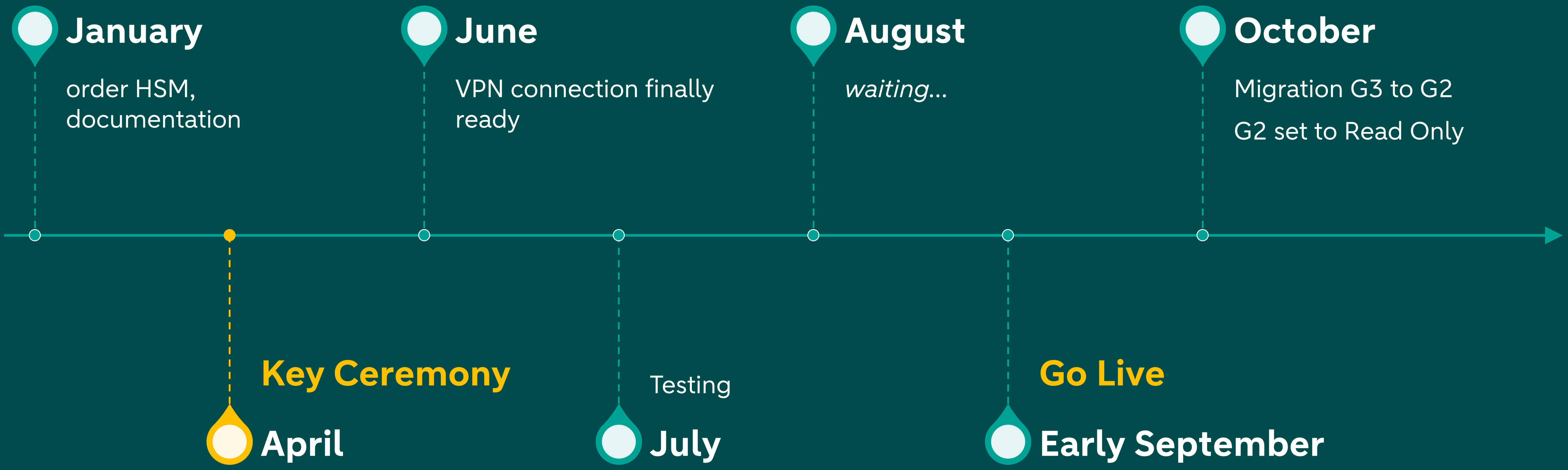
Go Live

Migration G3 to G2

G2 set to Read Only

September

Real timeline of the migration G2 to G3 — August



January

order HSM, documentation

Key Ceremony

April

June

VPN connection finally ready

Testing

July

August

waiting…

Go Live

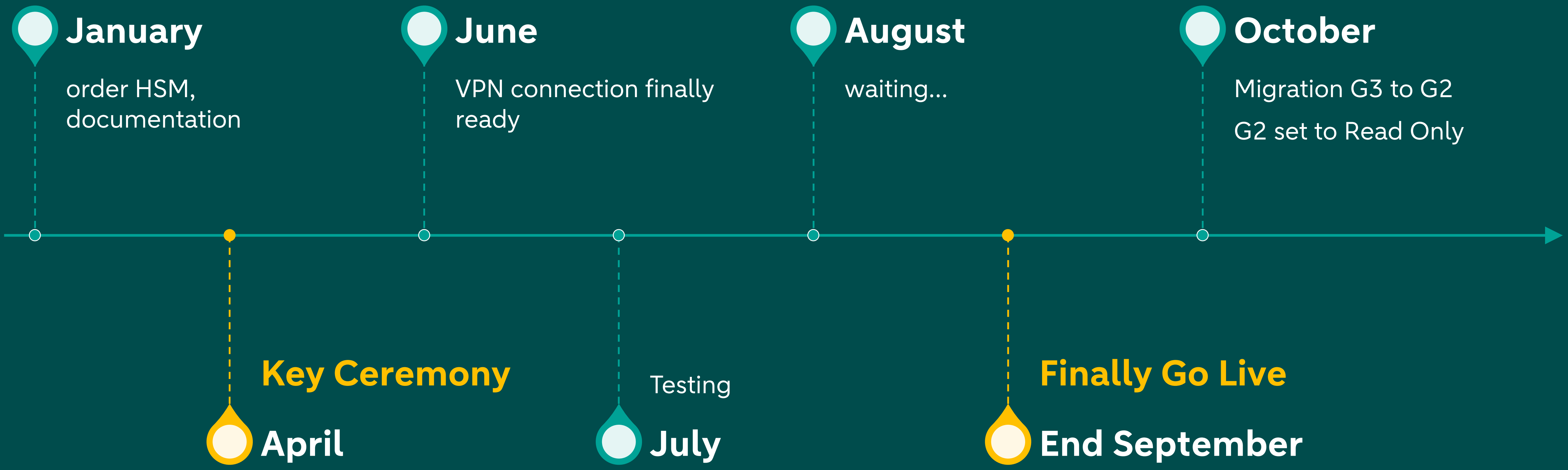
Early September

October

Migration G3 to G2

G2 set to Read Only

Real timeline of the migration G2 to G3 — September



January

order HSM, documentation

Key Ceremony

April

June

VPN connection finally ready

Testing

July

August

waiting...

Finally Go Live

End September

October

Migration G3 to G2

G2 set to Read Only

Go Live INFRA CA G3 Delays - Analysis

Multiple pushback due to other teams' lack of readiness.

Knowledge
problem

Communication
problem

Fear
of outages

CA management — PED keys (simplified)

	Black key	Red key	Blue key	White key
HSM management (user etc.)			✓ ✓ ✓	
Create new private key	✓ ✓ ✓			
Sign a CSR	✓ ✓			
Back up private key	✓	✓		
Revoke certificate/CA	✓ ✓			
Audit				✓

Usually, a quorum of M of N PED Keys is created, where M is the number of keys necessary to complete run a command as that role and N is the total number of keys created.

4 Anatomy of a Key Ceremony

Picture from AP Photo/John Raoux

Anatomy of a Key Ceremony

INFRA CA G3

Tuesday: Preparation.

Inventory.

Checking run books.

Wednesday: Key Ceremony Acceptance

Missing VGA cable. 😅

Hiccup in Sub CA signing. 😡

Swedish Keyboard stuck. 😡

Missing run books… 😑

Thursday: Key Ceremony Production

All run books provided. ✅

No hiccup. ✅

Keyboard ➡ US. ✅

Black/Blue/Red Keys safe. ✅

ET fixed. ✅

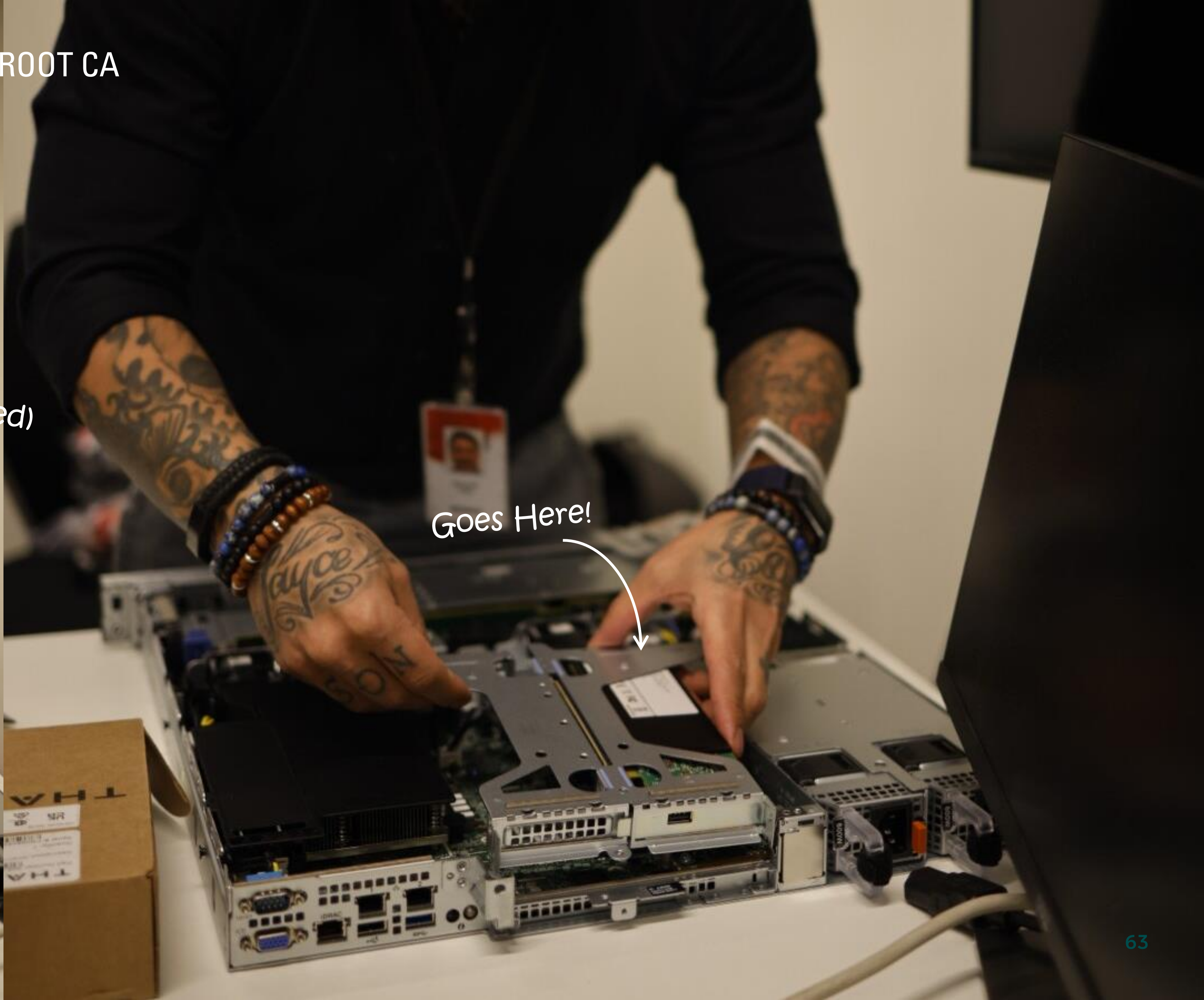
- Keyboard ➡ US.
- Faulty Sub CA cleaned.

[redacted]

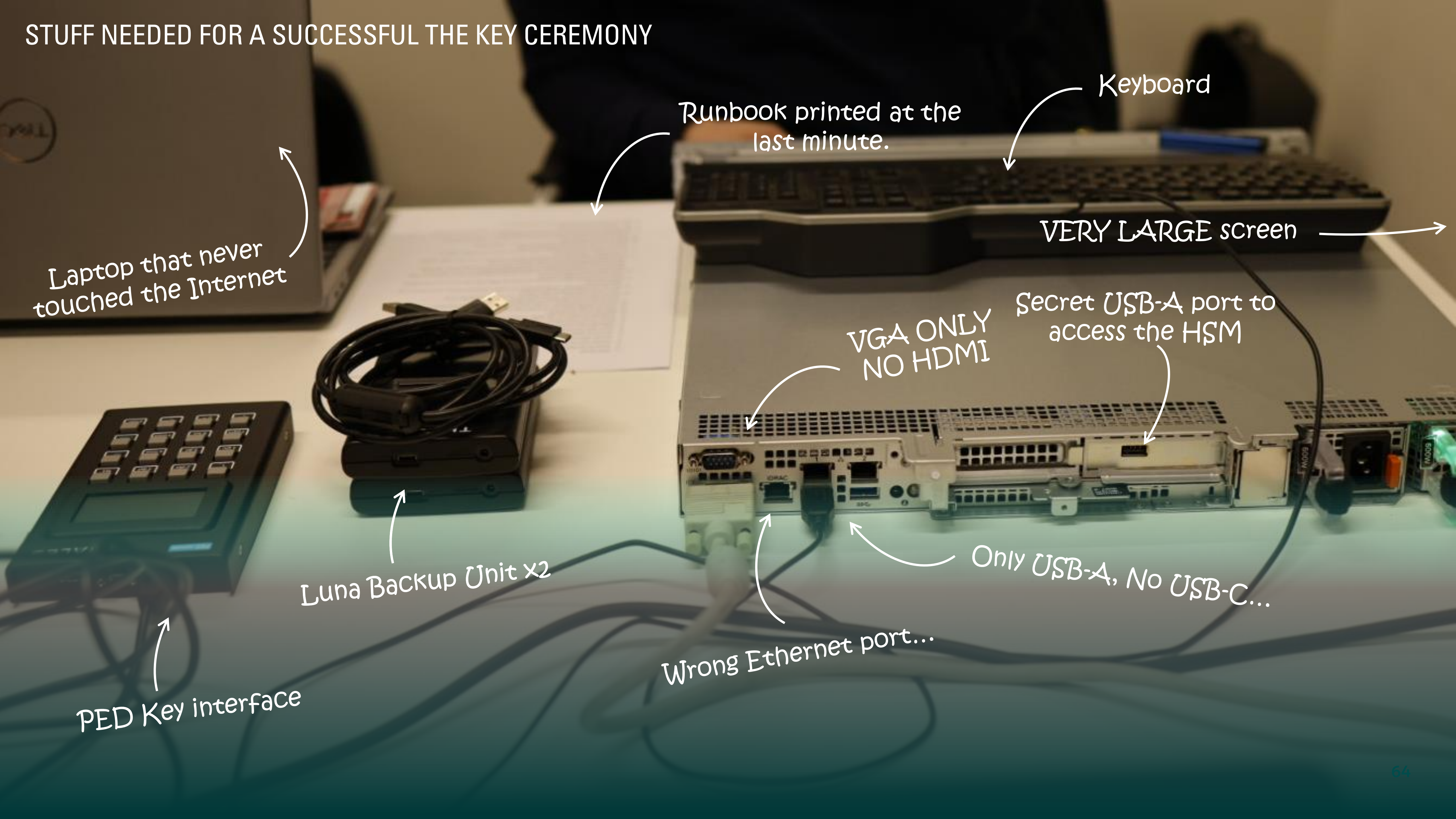
[redacted]

INSTALLATION OF THE HSM FOR THE ROOT CA

[redacted]



STUFF NEEDED FOR A SUCCESSFUL THE KEY CEREMONY



Runbook printed at the last minute.

Keyboard

Laptop that never touched the Internet

VERY LARGE screen

Secret USB-A port to access the HSM

VGA ONLY NO HDMI

Only USB-A, No USB-C...

Luna Backup Unit x2

Wrong Ethernet port...

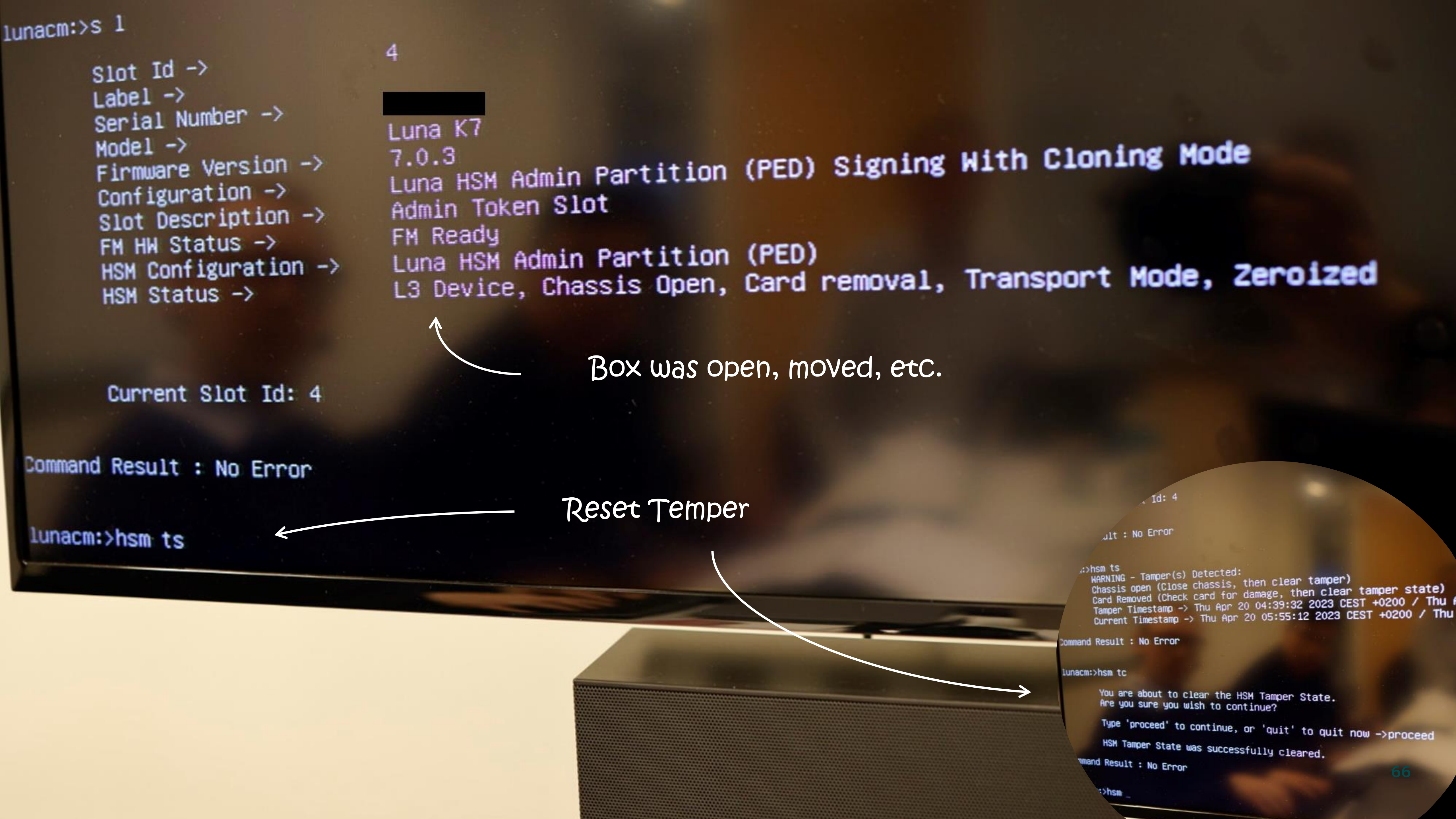
PED Key interface

INSTALLATION OF THE OPERATING SYSTEM (Slim version of Debian)

[redacted]

[redacted]

```
Scanning installation media
76%
Scanning /cdrom/pool/main/c...
```



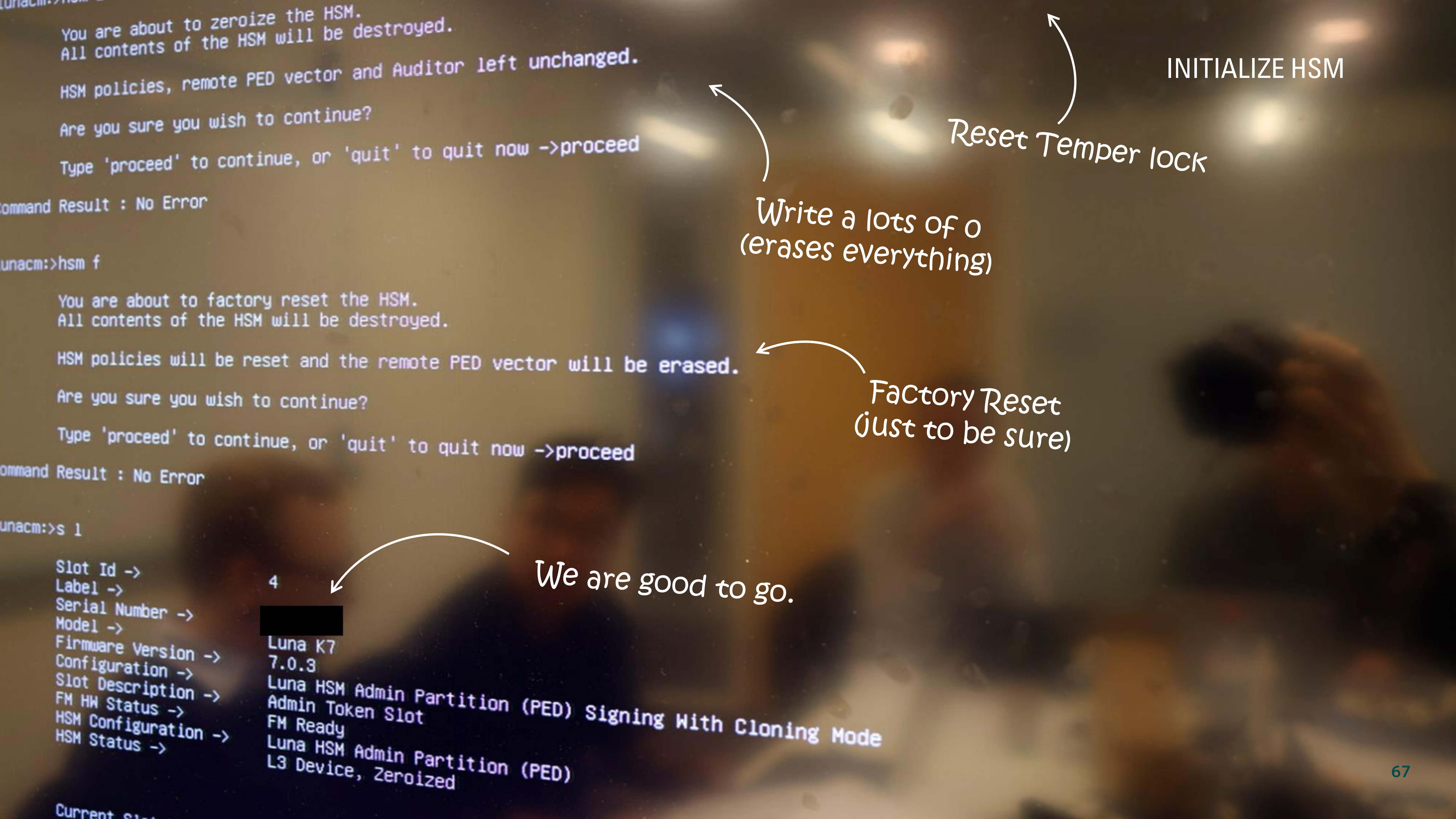
```
lunacm:>s l

        Slot Id ->                  4
        Label ->                    [redacted]
        Serial Number ->
        Model ->                    Luna K7
        Firmware Version ->         7.0.3
        Configuration ->            Luna HSM Admin Partition (PED) Signing With Cloning Mode
        Slot Description ->         Admin Token Slot
        FM HW Status ->             FM Ready
        HSM Configuration ->        Luna HSM Admin Partition (PED)
        HSM Status ->               L3 Device, Chassis Open, Card removal, Transport Mode, Zeroized


         Current Slot Id: 4


Command Result : No Error


lunacm:>hsm ts
```

Box was open, moved, etc.

Reset Temper

```
 Id: 4

ult : No Error

:>hsm ts
    WARNING - Tamper(s) Detected:
    Chassis open (Close chassis, then clear tamper)
    Card Removed (Check card for damage, then clear tamper state)
    Tamper Timestamp -> Thu Apr 20 04:39:32 2023 CEST +0200 / Thu
    Current Timestamp -> Thu Apr 20 05:55:12 2023 CEST +0200 / Thu

Command Result : No Error

lunacm:>hsm tc

        You are about to clear the HSM Tamper State.
        Are you sure you wish to continue?

        Type 'proceed' to continue, or 'quit' to quit now ->proceed

        HSM Tamper State was successfully cleared.

mand Result : No Error

:>hsm _
```

INITIALIZE HSM



```
unacm:>hsm

      You are about to zeroize the HSM.
      All contents of the HSM will be destroyed.

      HSM policies, remote PED vector and Auditor left unchanged.

      Are you sure you wish to continue?

      Type 'proceed' to continue, or 'quit' to quit now ->proceed

ommand Result : No Error


unacm:>hsm f

      You are about to factory reset the HSM.
      All contents of the HSM will be destroyed.

      HSM policies will be reset and the remote PED vector will be erased.

      Are you sure you wish to continue?

      Type 'proceed' to continue, or 'quit' to quit now ->proceed

ommand Result : No Error


unacm:>s l

      Slot Id ->                4
      Label ->
      Serial Number ->          [redacted]
      Model ->                  Luna K7
      Firmware Version ->       7.0.3
      Configuration ->          Luna HSM Admin Partition (PED) Signing With Cloning Mode
      Slot Description ->       Admin Token Slot
      FM HW Status ->           FM Ready
      HSM Configuration ->      Luna HSM Admin Partition (PED)
      HSM Status ->             L3 Device, Zeroized

      Current Sl
```

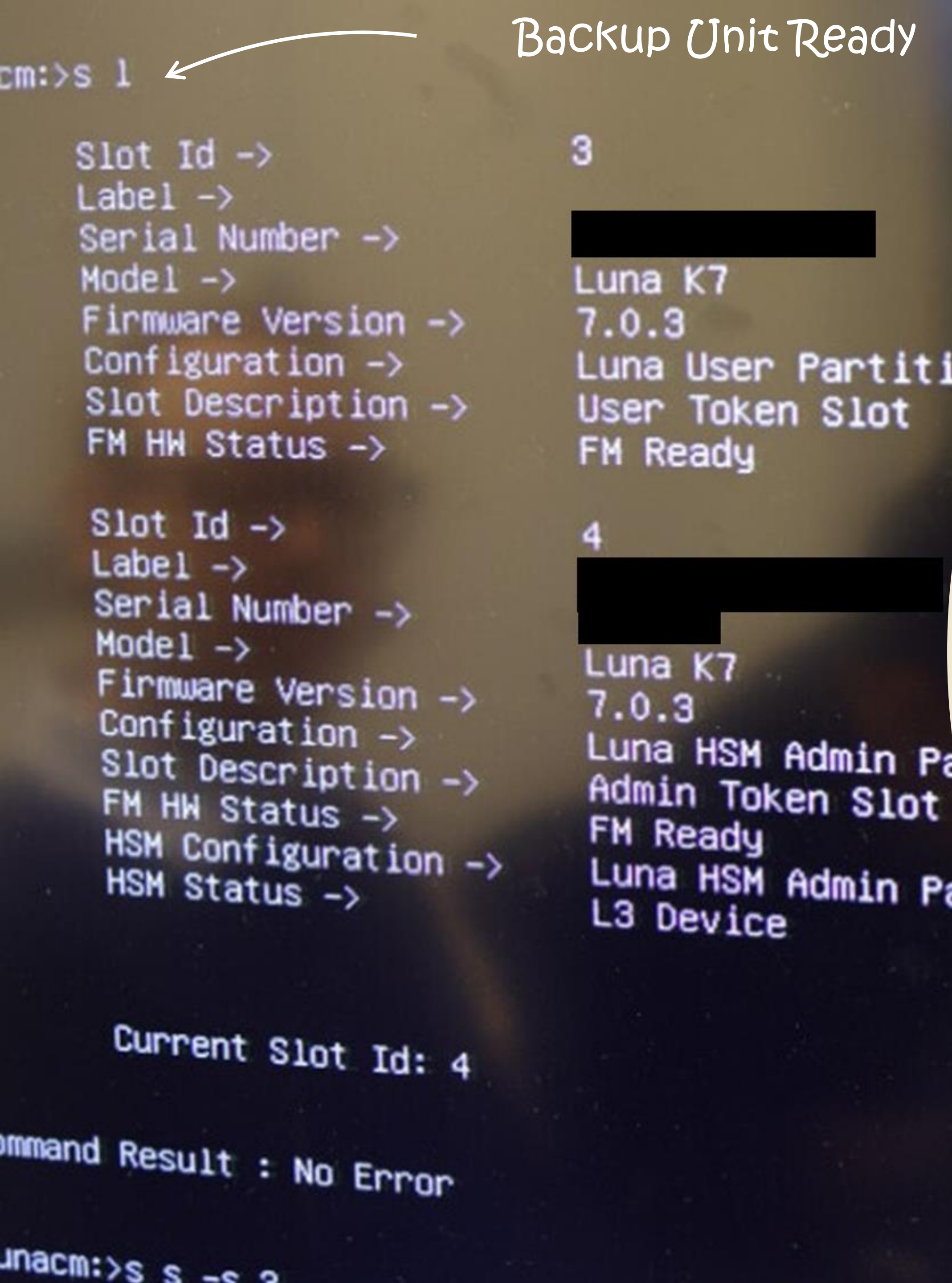
Reset Temper lock

Write a lots of 0
(erases everything)

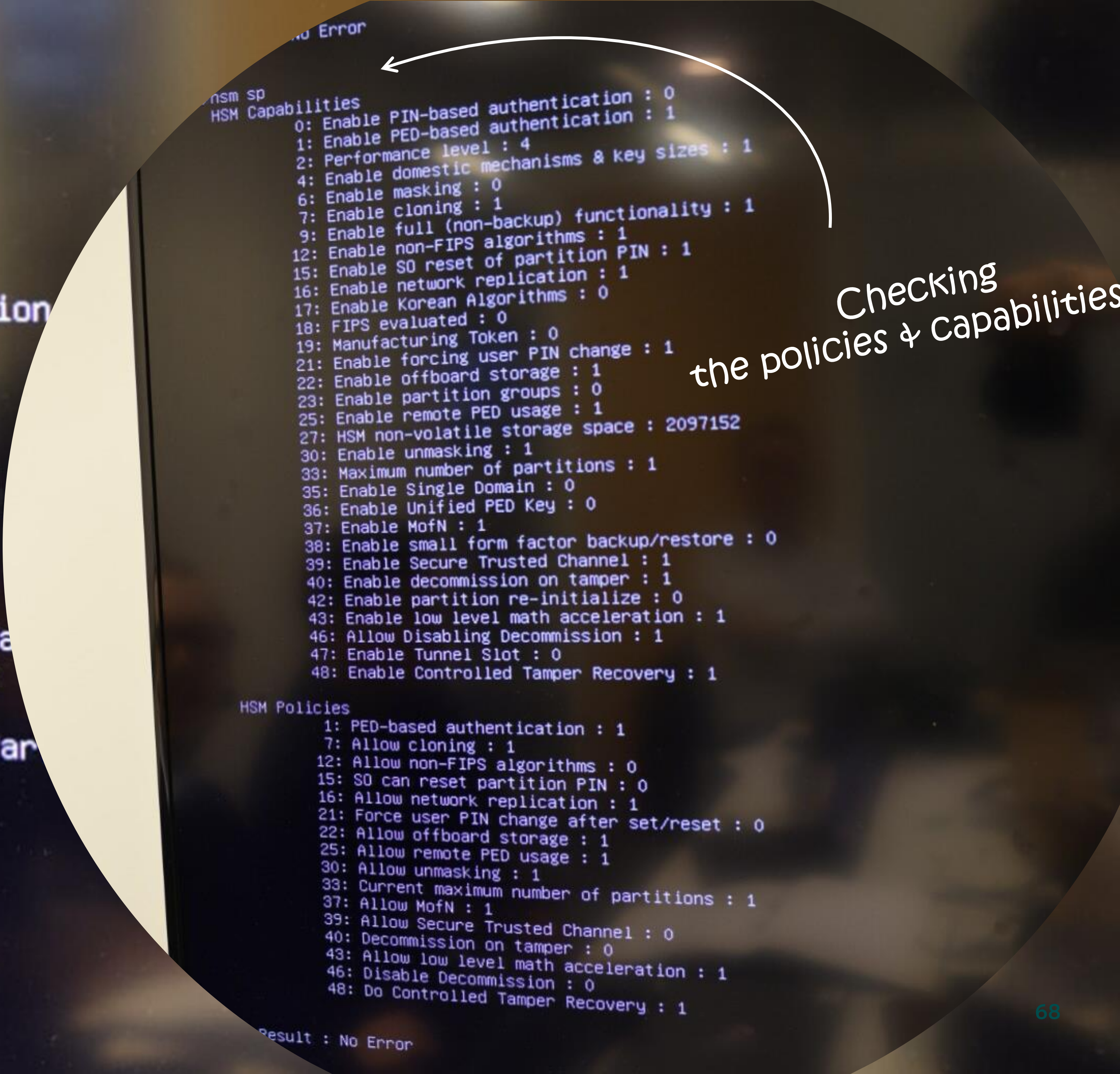
Factory Reset
(just to be sure)

We are good to go.

Backup Unit Ready

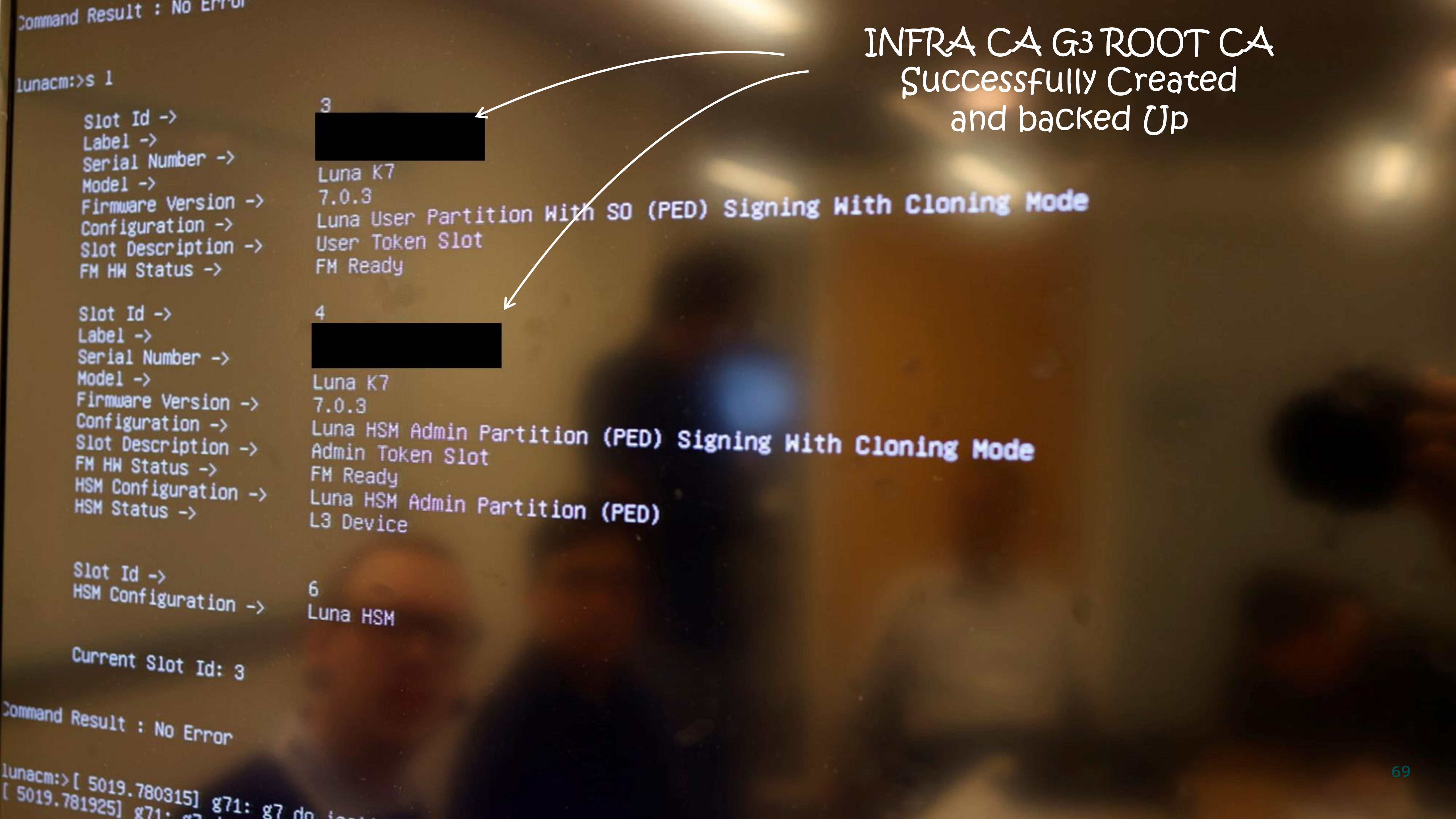


```
cm:>s l

    Slot Id ->                  3
    Label ->                    [redacted]
    Serial Number ->
    Model ->                    Luna K7
    Firmware Version ->         7.0.3
    Configuration ->            Luna User Partition
    Slot Description ->         User Token Slot
    FM HW Status ->             FM Ready

    Slot Id ->                  4
    Label ->                    [redacted]
    Serial Number ->
    Model ->                    Luna K7
    Firmware Version ->         7.0.3
    Configuration ->            Luna HSM Admin Pa
    Slot Description ->         Admin Token Slot
    FM HW Status ->             FM Ready
    HSM Configuration ->        Luna HSM Admin Par
    HSM Status ->               L3 Device


    Current Slot Id: 4

ommand Result : No Error

unacm:>s s -s 3
```



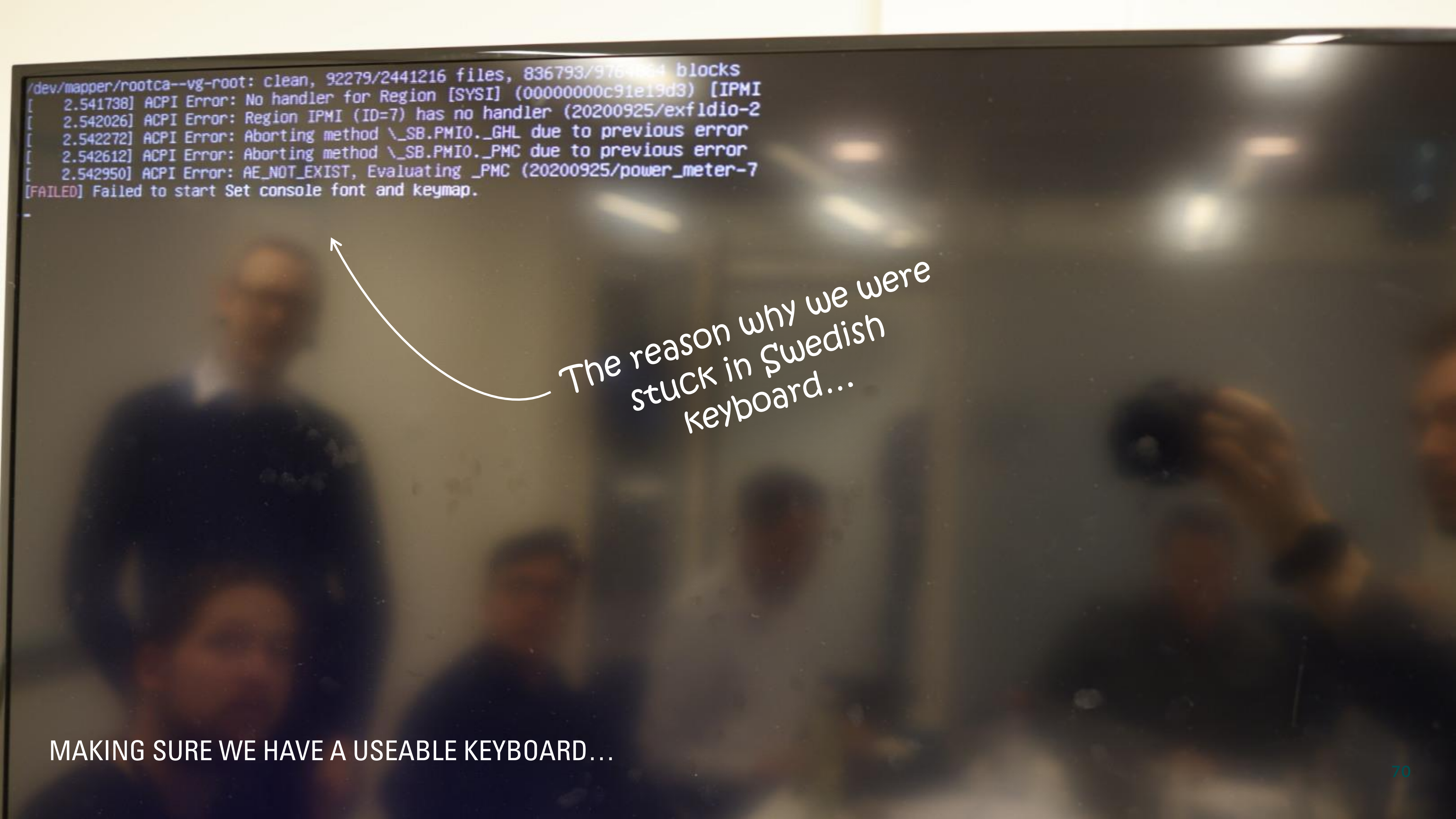
Checking
the policies & capabilities

```
o Error


nsm sp
HSM Capabilities
        0: Enable PIN-based authentication : 0
        1: Enable PED-based authentication : 1
        2: Performance level : 4
        4: Enable domestic mechanisms & key sizes : 1
        6: Enable masking : 0
        7: Enable cloning : 1
        9: Enable full (non-backup) functionality : 1
       12: Enable non-FIPS algorithms : 1
       15: Enable SO reset of partition PIN : 1
       16: Enable network replication : 1
       17: Enable Korean Algorithms : 0
       18: FIPS evaluated : 0
       19: Manufacturing Token : 0
       21: Enable forcing user PIN change : 1
       22: Enable offboard storage : 1
       23: Enable partition groups : 0
       25: Enable remote PED usage : 1
       27: HSM non-volatile storage space : 2097152
       30: Enable unmasking : 1
       33: Maximum number of partitions : 1
       35: Enable Single Domain : 0
       36: Enable Unified PED Key : 0
       37: Enable MofN : 1
       38: Enable small form factor backup/restore : 0
       39: Enable Secure Trusted Channel : 1
       40: Enable decommission on tamper : 1
       42: Enable partition re-initialize : 0
       43: Enable low level math acceleration : 1
       46: Allow Disabling Decommission : 1
       47: Enable Tunnel Slot : 0
       48: Enable Controlled Tamper Recovery : 1

HSM Policies
        1: PED-based authentication : 1
        7: Allow cloning : 1
       12: Allow non-FIPS algorithms : 0
       15: SO can reset partition PIN : 0
       16: Allow network replication : 1
       21: Force user PIN change after set/reset : 0
       22: Allow offboard storage : 1
       25: Allow remote PED usage : 1
       30: Allow unmasking : 1
       33: Current maximum number of partitions : 1
       37: Allow MofN : 1
       39: Allow Secure Trusted Channel : 0
       40: Decommission on tamper : 0
       43: Allow low level math acceleration : 1
       46: Disable Decommission : 0
       48: Do Controlled Tamper Recovery : 1

Result : No Error
```



INFRA CA G3 ROOT CA
Successfully Created
and backed Up

```
Command Result : No Error

lunacm:>s l

        Slot Id ->              3
        Label ->                [redacted]
        Serial Number ->
        Model ->                Luna K7
        Firmware Version ->     7.0.3
        Configuration ->        Luna User Partition With SO (PED) Signing With Cloning Mode
        Slot Description ->     User Token Slot
        FM HW Status ->         FM Ready

        Slot Id ->              4
        Label ->                [redacted]
        Serial Number ->
        Model ->                Luna K7
        Firmware Version ->     7.0.3
        Configuration ->        Luna HSM Admin Partition (PED) Signing With Cloning Mode
        Slot Description ->     Admin Token Slot
        FM HW Status ->         FM Ready
        HSM Configuration ->    Luna HSM Admin Partition (PED)
        HSM Status ->           L3 Device

        Slot Id ->              6
        HSM Configuration ->    Luna HSM

        Current Slot Id: 3

Command Result : No Error

lunacm:>[ 5019.780315] g71: g7 do
[ 5019.781925] g71: g7
```



```
/dev/mapper/rootca--vg-root: clean, 92279/2441216 files, 836793/9764864 blocks
[    2.541738] ACPI Error: No handler for Region [SYSI] (00000000c91e19d3) [IPMI
[    2.542026] ACPI Error: Region IPMI (ID=7) has no handler (20200925/exfldio-2
[    2.542272] ACPI Error: Aborting method \_SB.PMI0._GHL due to previous error
[    2.542612] ACPI Error: Aborting method \_SB.PMI0._PMC due to previous error
[    2.542950] ACPI Error: AE_NOT_EXIST, Evaluating _PMC (20200925/power_meter-7
[FAILED] Failed to start Set console font and keymap.
_
```

The reason why we were stuck in Swedish keyboard…

MAKING SURE WE HAVE A USEABLE KEYBOARD…

FINALLY…
BLACK/RED/BLUE KEY
ARE SECURELY
STORED IN THEIR
RESPECTIVE VAULT.

Picture from AP Photo/Chris O'Meara

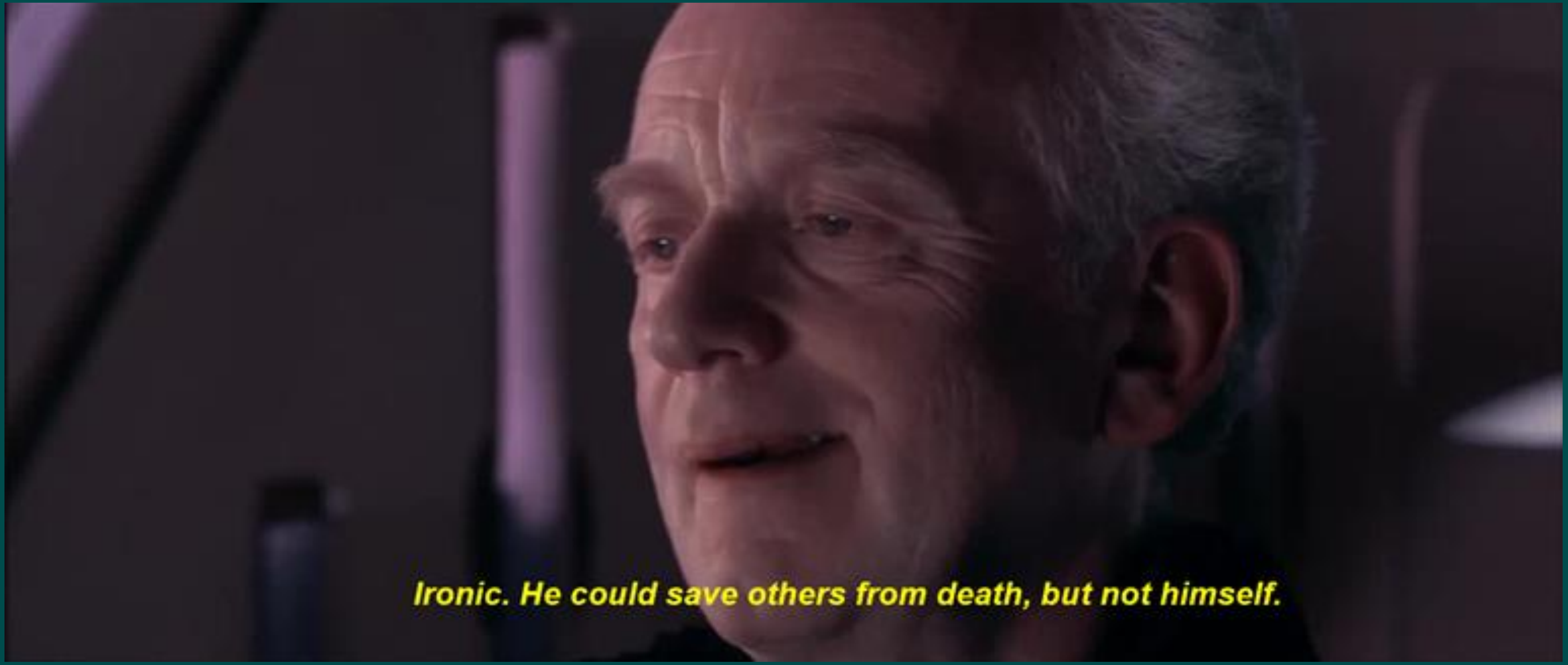
INFRA G3 ROOT CA KEY CEREMONY COMPLETED SUCCESFULLY.

Picture from gibsoncreativefl.com

5 Things that ~~could go~~ *went* wrong...

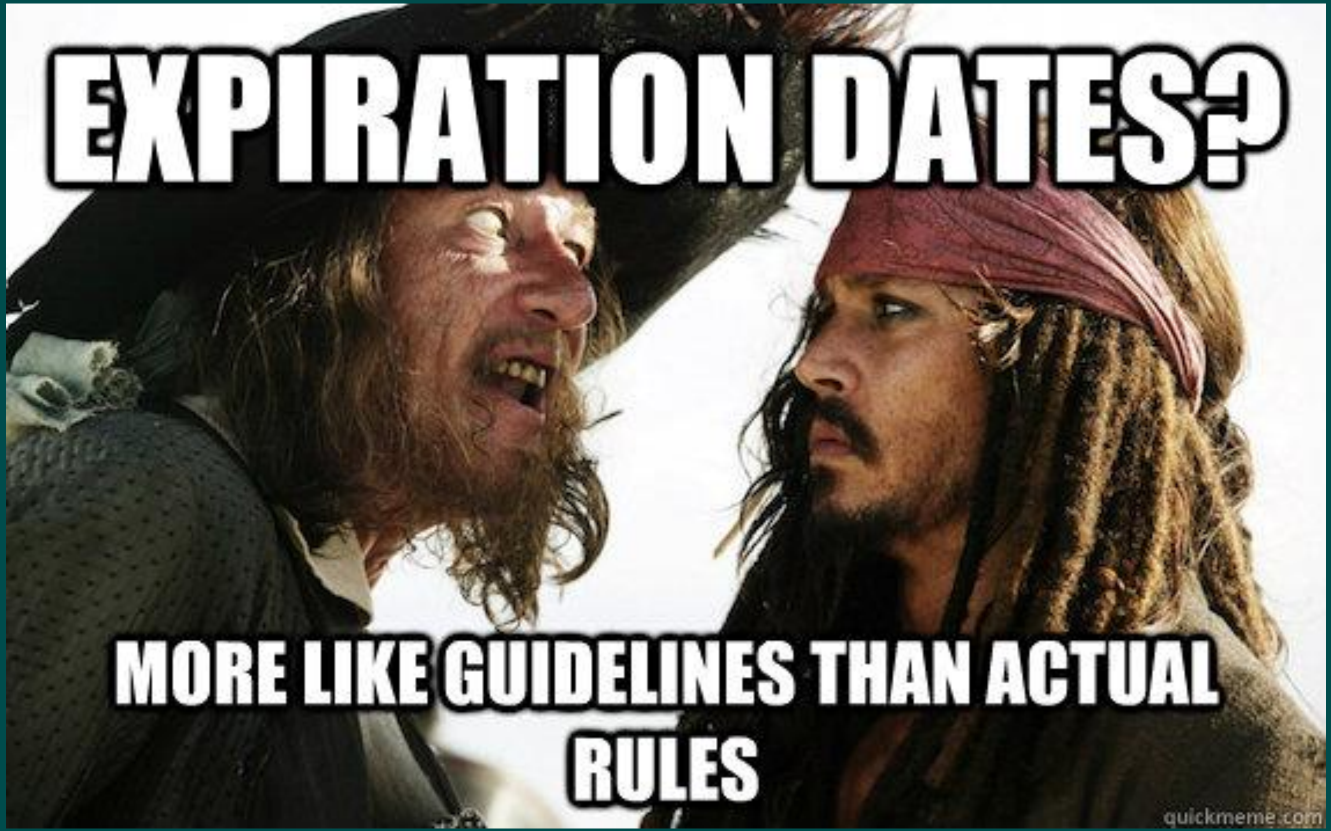
ABN·AMRO

Our certificate in the signing store expired.



Lesson learned: we need reminders for our OWN certificates... 😅

Signing certificate from the CA expired.



Lesson learned: Also track our providers’ certificate expiration dates...

Connection ServiceNow Prod — AZRA A is down.

Our Product Owner onboarded a new client and removed an IP range from the firewall white-listing...

Apparently everything's down?
Yeah... they planned a security drill this morning...
They wanted to test the backup and recovery system in emergency conditions
In real emergency conditions?
Yeah, real emergency conditions...
So you mean they've rolled back to a corrupt backup point and now they don't know how to get everything working again?
Like I said... emergency conditions...
CommitStrip.com

Lesson learned: Use infra as Code to avoids such errors.

Key rotation fail…

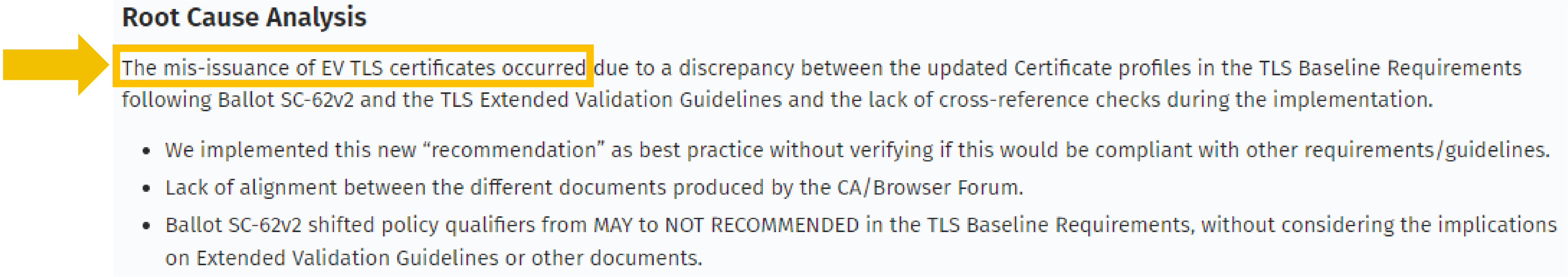
Colleague who shall not be named…

12H down time, both Acceptance and Production at the same time.

Reason: Transparent Database Encryption key expired AGAIN in HSM…

Errors on the External CA side...

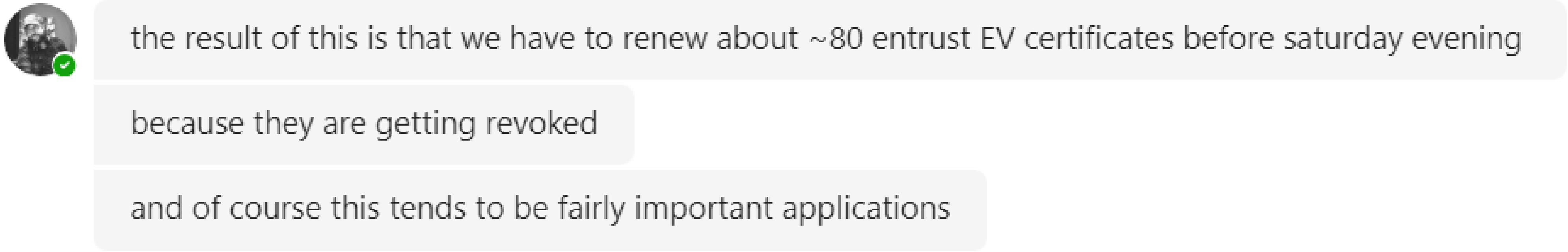
https://bugzilla.mozilla.org/show_bug.cgi?id=1883843#c19



Root Cause Analysis

The mis-issuance of EV TLS certificates occurred due to a discrepancy between the updated Certificate profiles in the TLS Baseline Requirements following Ballot SC-62v2 and the TLS Extended Validation Guidelines and the lack of cross-reference checks during the implementation.

- We implemented this new “recommendation” as best practice without verifying if this would be compliant with other requirements/guidelines.
- Lack of alignment between the different documents produced by the CA/Browser Forum.
- Ballot SC-62v2 shifted policy qualifiers from MAY to NOT RECOMMENDED in the TLS Baseline Requirements, without considering the implications on Extended Validation Guidelines or other documents.



the result of this is that we have to renew about ~80 entrust EV certificates before saturday evening

because they are getting revoked

and of course this tends to be fairly important applications

6 Takeaways.

Or the 3 largest problems of Enterprise Cryptography.

ABN·AMRO

Problem 1

Certificate Management & Key agility

Problem 2

Trust store Management & CA rotation

Problem 3

Post Quantum migration.

See Thom’s talk.

7 Anyway…

ABN·AMRO

Working at ABN

- Extremely Internationalized.
- Super cool environment.
- Nice benefits (Altijd Vrij OV, learning budget, WFH).
- Decent salaries.

DO NOT APPLY FOR INTERNSHIPS (it’s shit).
APPLY FOR JOBS, it’s cool ! 😊

https://www.werkenbijabnamro.nl/

ABN·AMRO