# Practical PKI

## A hands-on X.509 workshop

Fraser Tweedale

Everything Open 2026

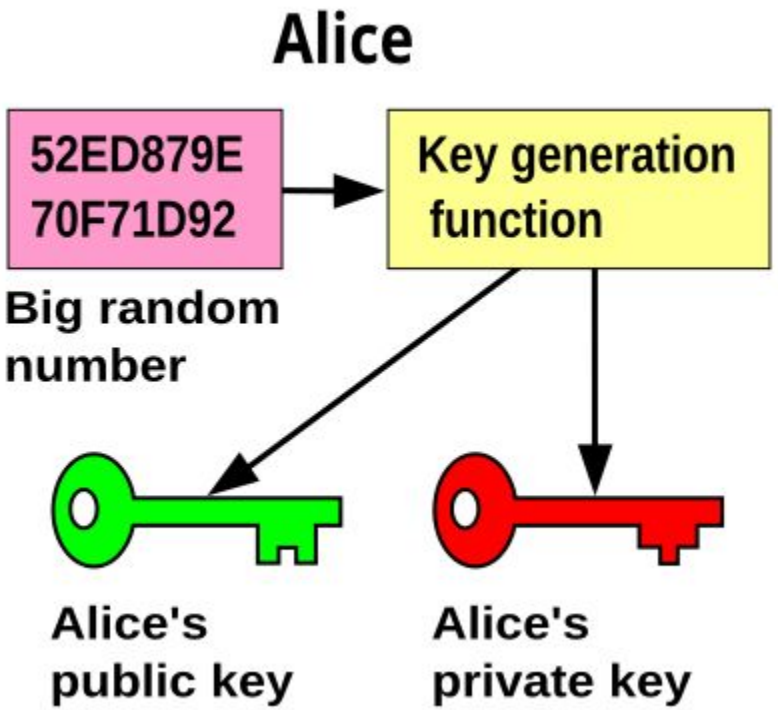
# This workshop

- PKI overview
- FreeIPA overview
- The workshop environment
- Do the workshop!
- **Ask questions throughout**

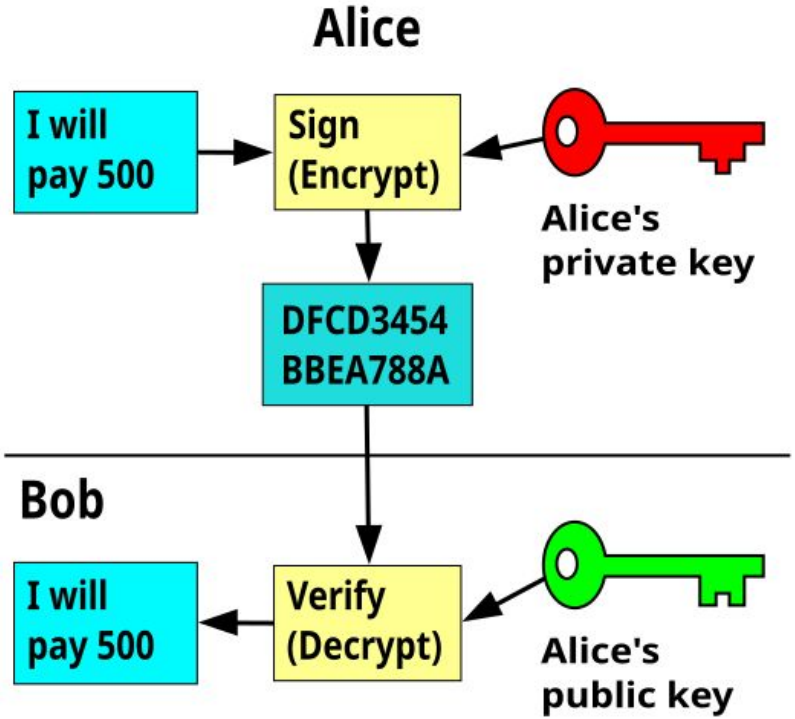
# For the impatient...

- Workshop curriculum: https://eo2026.frase.id.au/
- You (should) have a card with environment access info
- You need:
  - Internet access + browser
  - An SSH client
  - (optional) an RDP client
- It's OK if you get stuck!  Ask for help!  (after the presentation)
- I will walk through the workshop modules

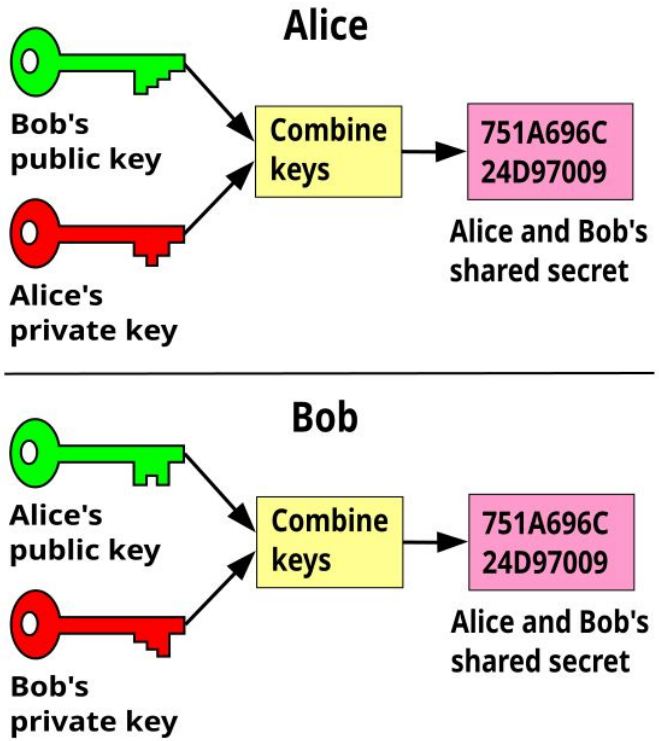
# Public key cryptography



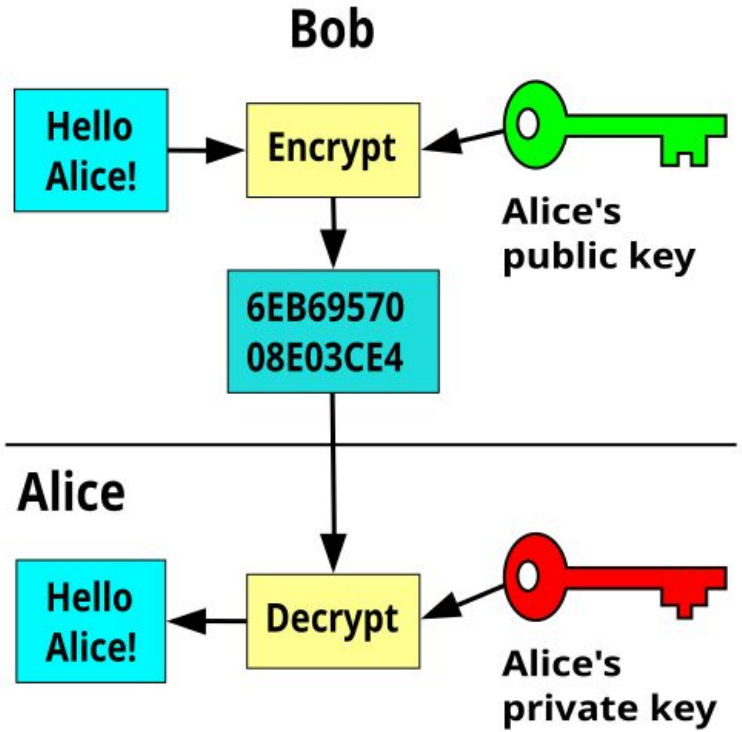
# Public key cryptography



# Public key cryptography



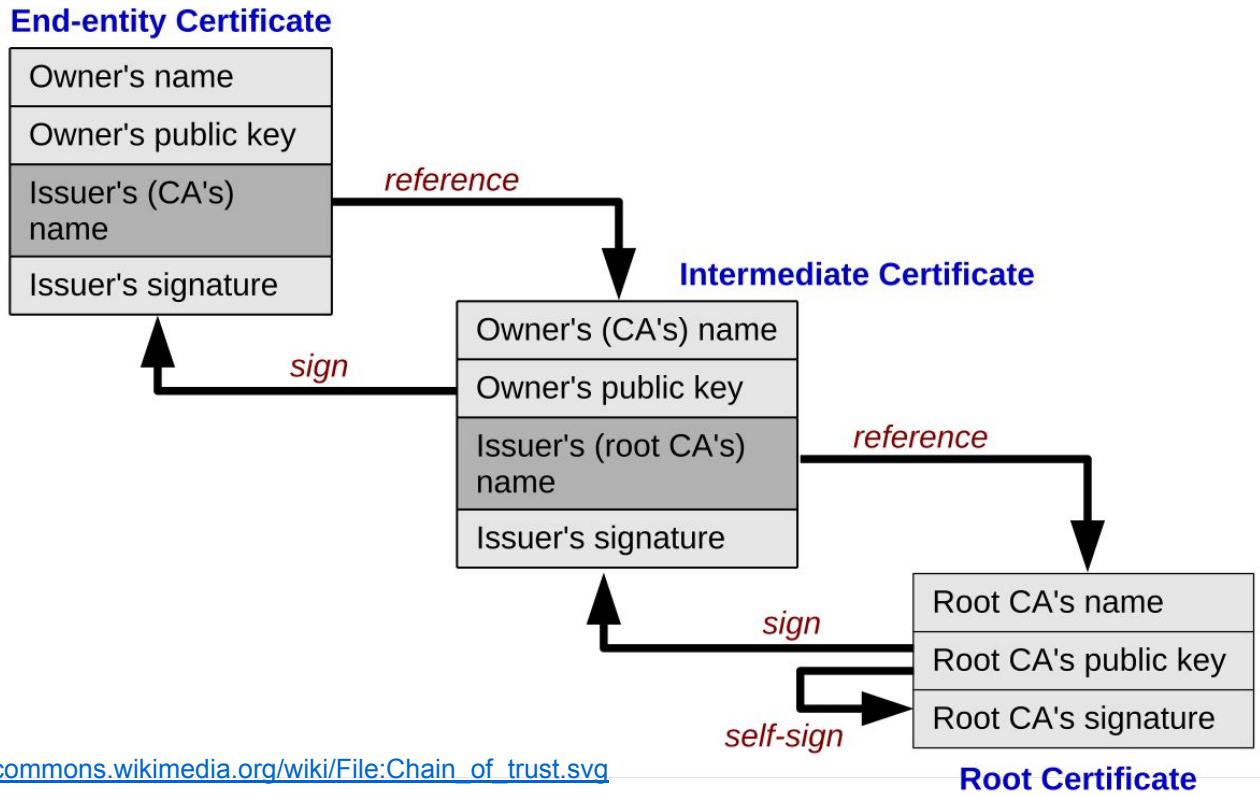
# Public key cryptography



# Public Key Infrastructure (PKI)

- Solve trust, privacy and authentications problems for the internet
- Uses public key cryptography (**no shared secrets**)
- **Certificate Authority** - verifies identities and **signs** certificates
- **End Entity (EE)** - present certificate for signing, encryption, key agreement
- **Relying party (RP)** - verifies peer certificate

# Chain of trust



# Certificate life cycle

- EE proves identity or domain control to CA
- CA issues certificate (with a fixed validity period, possibly short)
- EE must **renew** certificate before expiry
  - Update server config, certificate stores, etc

# X.509 anatomy

- Issuer DN *("DN" = Distinguished Name)*
- Serial number *(big random number)*
- Subject DN
- Subject Public Key Info (SPKI)
- **Subject Alternative Name (SAN)**
  - DNS name, email address, IP address, URI, ...
- Basic Constraints - *is the subject a CA?*
- CRL Info + Authority Information Access (AIA) - *where to find revocation info*
- Signed Certificate Timestamps (SCT) - *Certificate Transparency proof*

# Algorithmic agility

- X.509 easily adapt to new cryptography
- Just define public key data and signature representations
- Chain of trust can use different algorithms at different levels
- Current efforts to standardise **Post-Quantum algorithms**:
  ML-DSA, ML-KEM and hybrid KEM.

# Certificate use cases

- TLS authentication (everyone knows this one)
- Smart cards / user authentication
- VPN / IPSec / 802.1x authentication
- Email signing and encryption (S/MIME ; cf OpenPGP)
- Passports (yes, really)
- … the list goes on

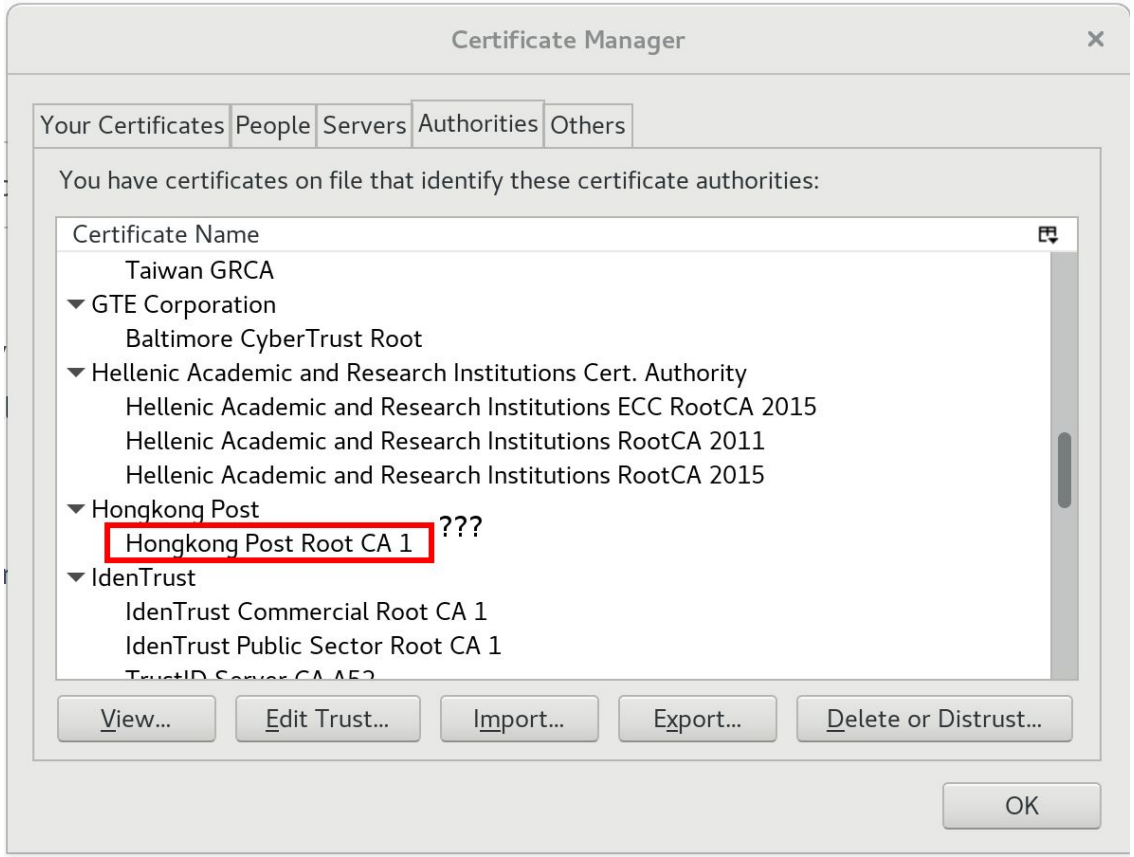
# PKI for the Internet

- Operating systems and browsers have a bundle of **trusted CAs**
  - CA/Browser Forum defines rules for CAs to be "publicly trusted"
  - **Baseline Requirements** defines the X.509 certificate profile
- A few BIG CAs and a long tail of small and esoteric ones
- **TLS (HTTPS)** server certificates chain up to a trusted CA
- Other use cases: code signing, email security
- Challenges: key security, revocation, threat intel

# TLS handshake

- Server sends a message with its certificate
- Server also signs its *ephemeral key*
- Client certificate authentication is also possible, server verifies
  - a.k.a. *mutual TLS / mTLS*
  - client cert is often signed by private CA (e.g. Bank or Gov authn)

# PKI for the Internet



# Revocation

- Your CA (at your request or own initiative) can ***revoke*** a certificate
- How do RPs (e.g. browsers) find out?
- Certificate Revocation List (**CRL**): big & slow, offline
- Online Certificate Status Protocol (**OCSP**): load & reliability, privacy issues
  - OCSP Stapling mitigates these concerns
- **CRLite**: small, offline, public CAs only
- Alternative: **don't revoke**; use short lifetimes and regular validation

# Certificate Transparency

- Publicly-trusted CAs are required to log all issued certs to a public log
- Certificates contain evidence of the logging (SCTs)
- Purpose: detect CA **compromise or misbehaviour**
- Extra benefit: domain owners can **detect unexpected issuance**
  - CT log **monitors:** e.g. https://crt.sh/ and others

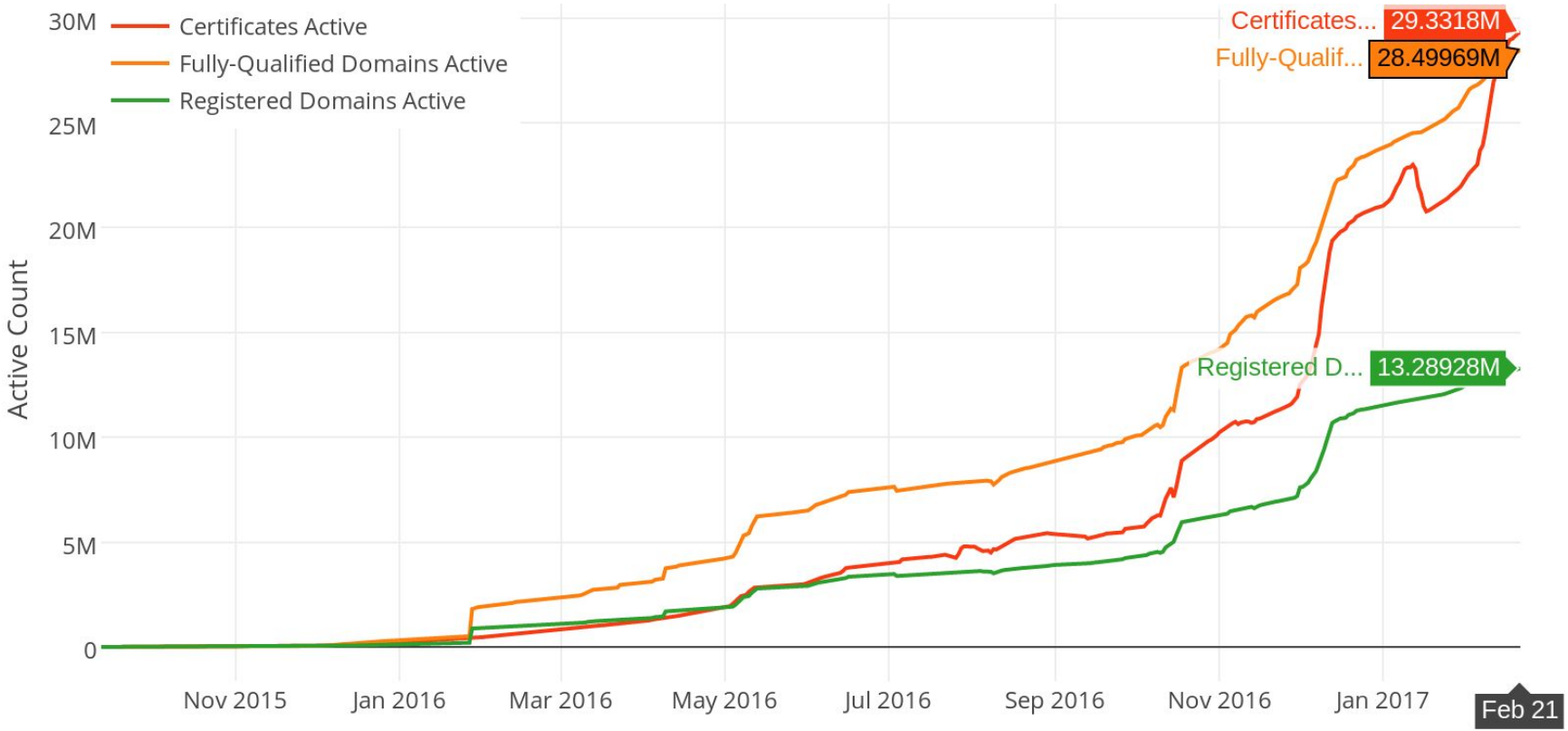
# Interactions with DNS

- **CAA** – *which CAs do you authorise? (RFC 8659)*
- **TXT** – *used for domain validation, including ACME "dns" challenge*
- **TLSA** – *directly publish your certs / fingerprints ; requires DNSSEC*
  - Mail servers are adopting it, and OpenSSL (has some support)
  - Hardly supported anywhere else, but quite interesting! RFC 6698

# ACME

- = *Automated Certificate Management Environment* (RFC 8555)
- **Automate** *domain validation* and *certificate request / renewal*
- Free public CA: **Let's Encrypt** (and others)
- Enables **short lifetimes** (reduce impact of compromise)
- Validation challenges:
  - Http: drop a file in a .well-known/ location
  - Dns: publish a TXT record
  - Alpn: too complicated to explain :)
- Clients: certbot, mod_md (Apache), caddy, ...

# ACME


30M
25M
20M
15M
10M
5M
0
Active Count
Certificates Active
Fully-Qualified Domains Active
Registered Domains Active
Certificates... 29.3318M
Fully-Qualif... 28.49969M
Registered D... 13.28928M
Nov 2015
Jan 2016
Mar 2016
May 2016
Jul 2016
Sep 2016
Nov 2016
Jan 2017
Feb 21

# PKI in the enterprise

- Large organisations often run an internal PKI
- Active Directory Certificate Services (AD-CS) is common
- **FreeIPA / RHEL Identity Management** is another option
- Applications: TLS, VPN, 802.1x, user auth
- Challenges: key security, automation, revocation, cost

# Smart cards

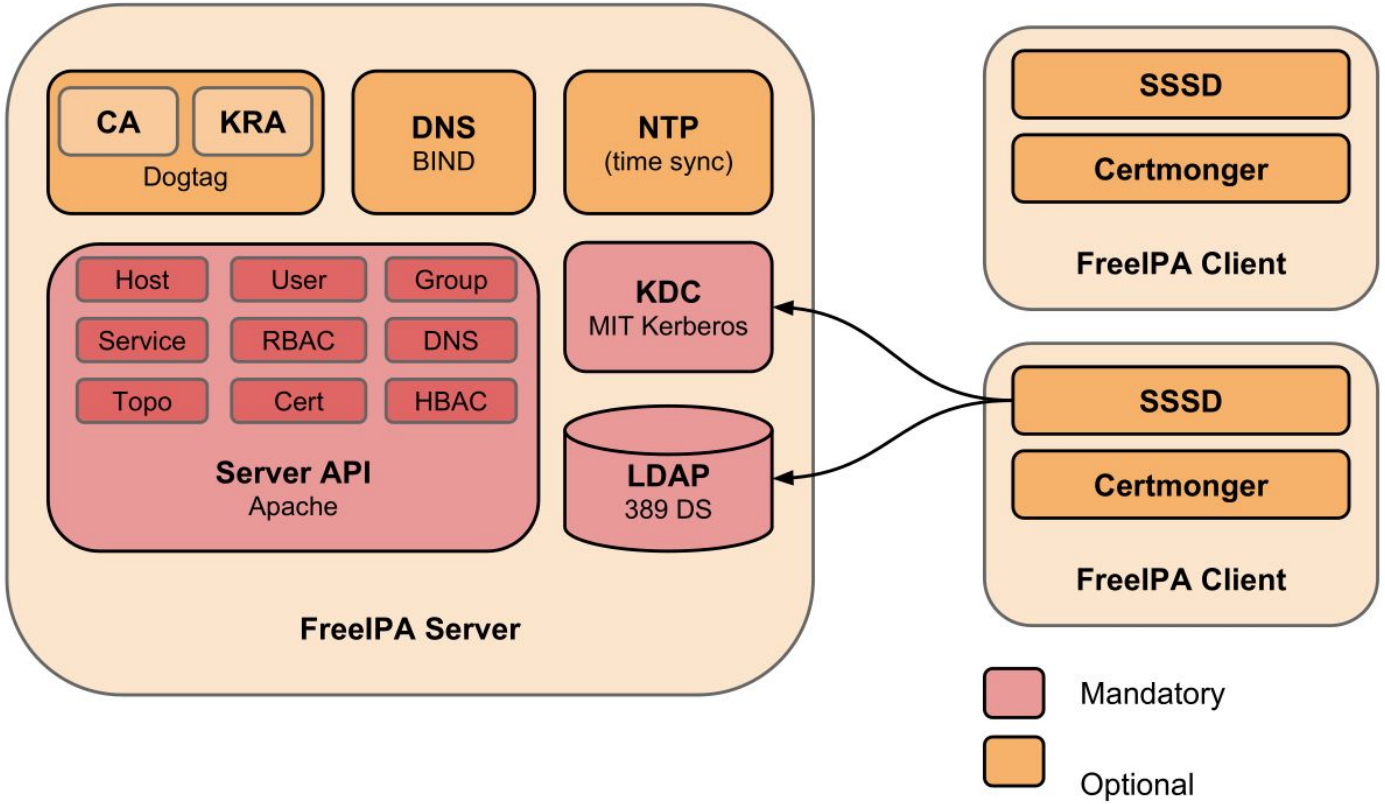
- Cryptographic token with (optional) PIN
- Private key + **X.509 certificate**
- Sign a challenge to prove identity
- **Phishing-resistant**
- Challenges: issuance, **renewal** and **revocation**
- USB, NFC, "SIM-esque", ...

# FreeIPA overview

- **Identity and access management** for Linux environments
- Includes a Certificate Authority
- Supported authentication methods:

  Password, OTP, external IdP, Passkeys, **certificate / smart card**

# FreeIPA architecture



# Workshop time!

- Workshop curriculum: https://eo2026.frase.id.au/
- You (should) have a card with environment access info
- You need:
  - Internet access + browser
  - An SSH client
  - (optional) an RDP client
- It's OK if you get stuck!  Ask for help!  (after the presentation)
- I will walk through the workshop modules
- Feedback: https://eo2026.frase.id.au/feedback