

# X.509 Certificate Management with Zephyr/TF-M


David Vincze



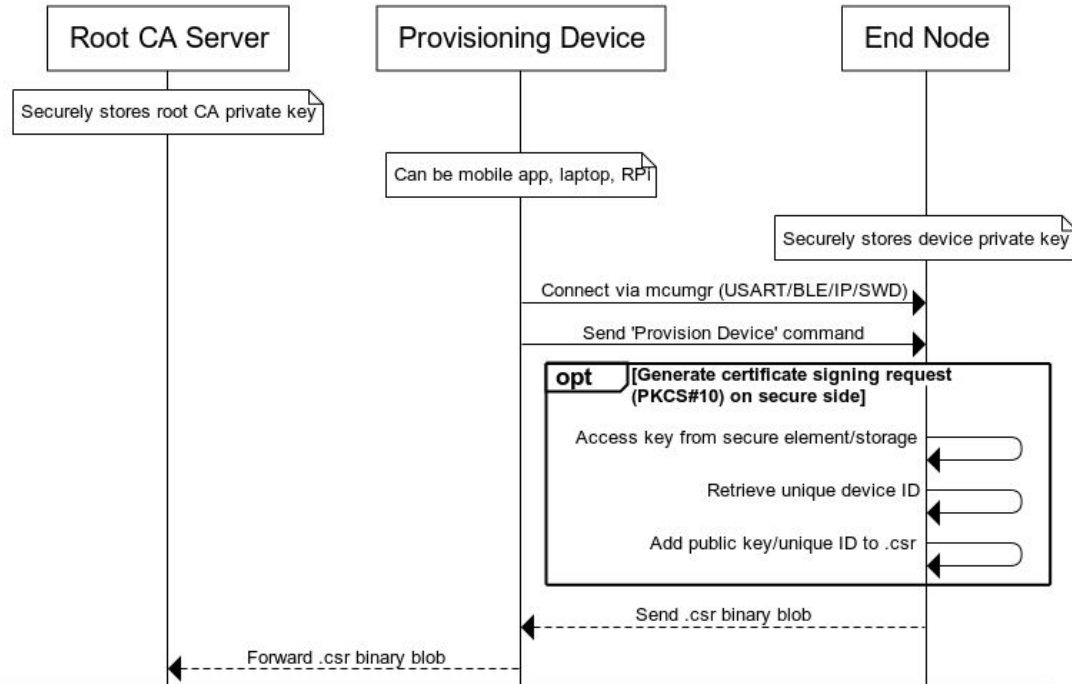
# Agenda

- Certificates and provisioning
- Workflow of provisioning
- Available components
- Zephyr sample application

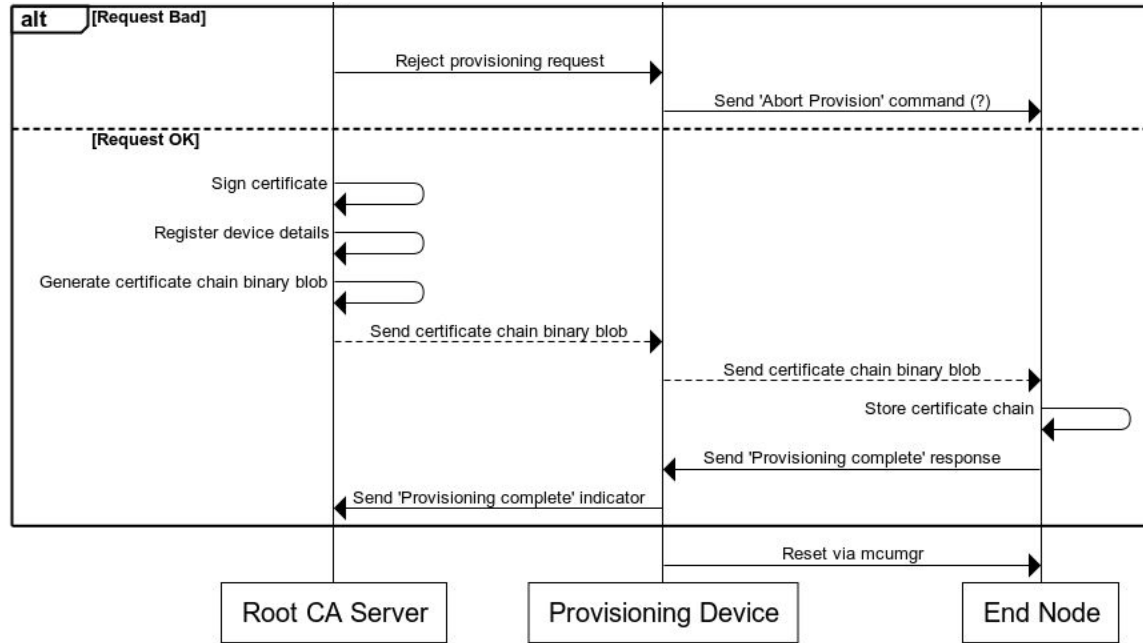
# Certificates and Provisioning

- Public key certificates
  - X.509 standard, public key infrastructure (PKI)
  - Revoking certificates if needed
  - Multi-level certificate chains
- IoT device provisioning into cloud services
  - Credential provisioning at scale?
  - Manual, during manufacturing (trust?), cellular network, security module
  - Provisioning/User Device  End Device communication

# Example Workflow of Provisioning



# Example Workflow of Provisioning

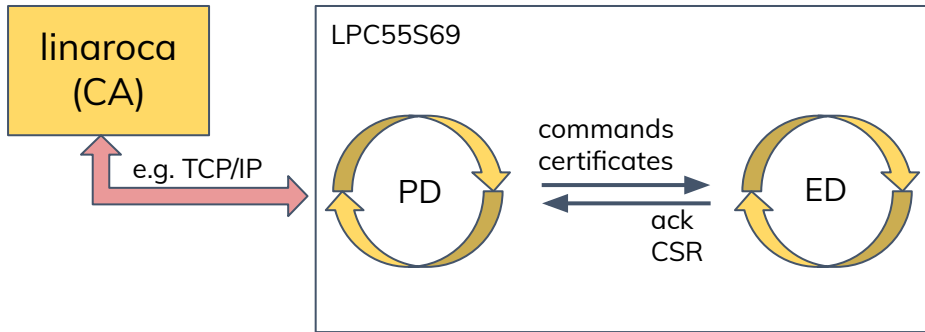


# Available Components

- “We don’t want to start from scratch... show us examples.”
- Root CA server: **linaroca** (<https://github.com/microbuilder/linaroca>)
- Provisioning device:
  - Separate thread in the application (early stage)
  - Mobile/laptop + mcumgr
- End node:
  - **NXP LPC55S69-EVK** (Cortex-M33 based board)
  - **Zephyr** ([RTOS](#), support and examples)
  - **Trusted Firmware-M** ([TF-M](#), secure processing environment)
  - **MCUboot** ([secure bootloader](#))

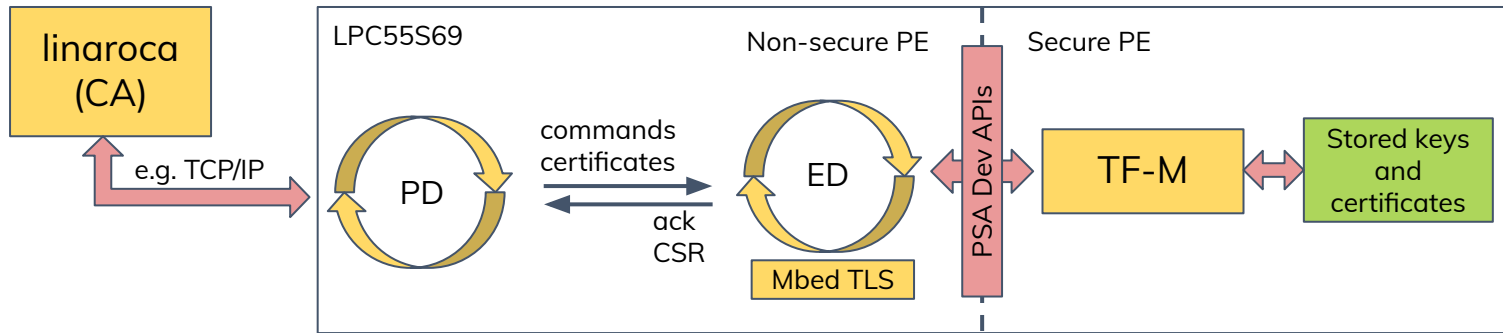
# Zephyr Sample Application

- Located in <zephyr>/samples/tfm\_integration/psa\_handshake\_simple, currently under [development](#)
- Separate threads for: provisioning device (PD), end device (ED)
- PD thread:
  - Commands for end device (ED)  
(e.g. wake up, generate key/CSR, receive certifications)
  - Communication with root CA  
(send CSR and receive certificates, ACK provisioning)



# Zephyr Sample Application

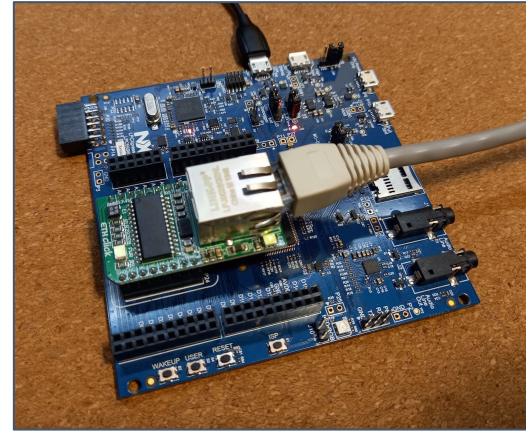
- ED thread:
  - Generate CSR (certificate signing request, PKCS#10) using Mbed TLS: X.509 module
  - TF-M provides secure services (e.g. Crypto, Storage)
    - Generate persistent prime256v1 EC key
    - Provide public key / unique device ID
    - Sign modified CSR with EC private key (doesn't leave the secure side)
  - Send CSR for PD, wait for certificates and store





# What is missing?

- CSRs are verified manually,
- Communication with CA server (linaroca) is missing
  - e.g. TCP/IP connection with ETH Click shield (support and example in Zephyr)
- Store CA's public key in a persistent secure storage for verification,
- Addition: support certificate revocation.



# Thank you

Accelerating deployment in the Arm Ecosystem

