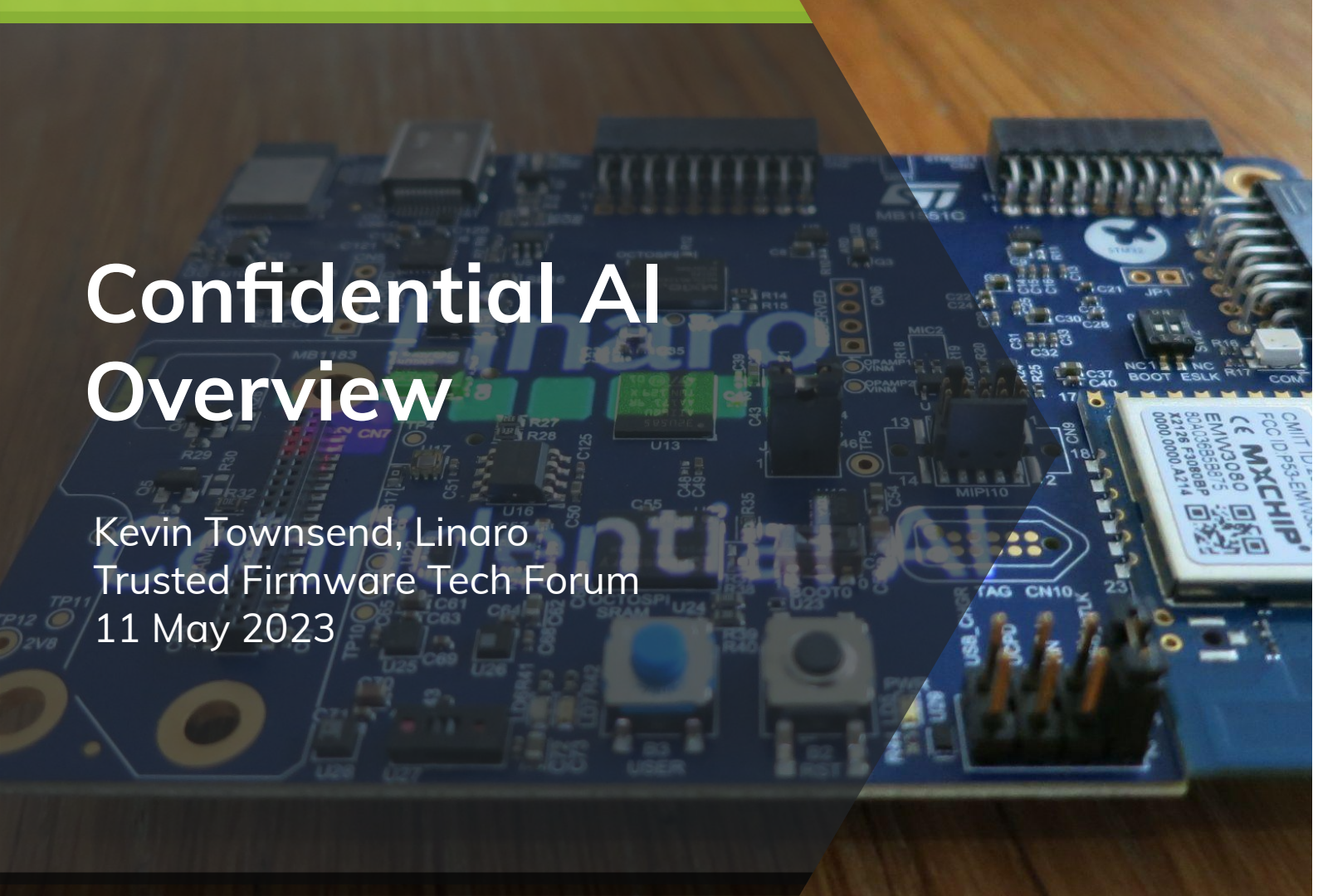


Confidential AI Overview

Kevin Townsend, Linaro

Trusted Firmware Tech Forum

11 May 2023

Linaro

What is ‘Confidential AI’?

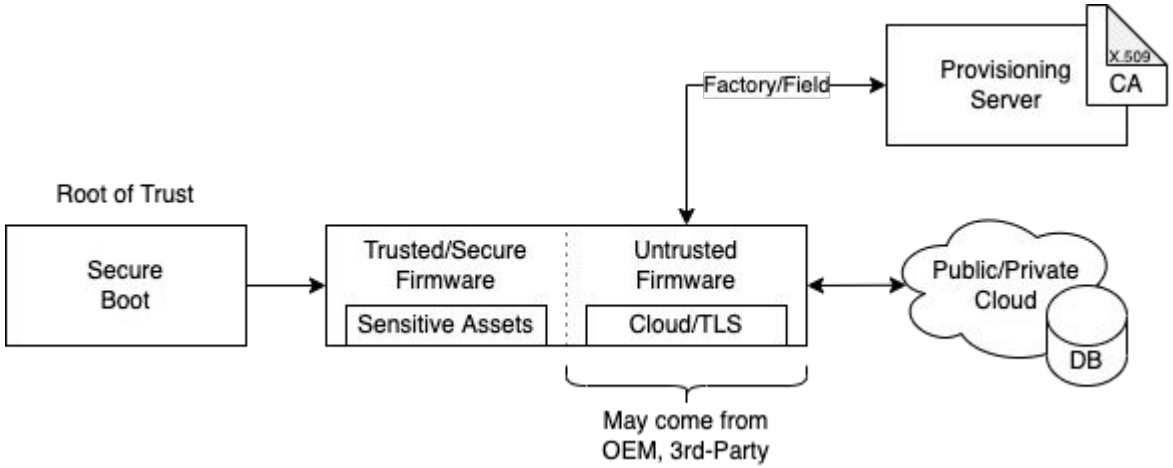
- An attempt to demonstrate end-to-end security best practices
- Making use of the security features on modern Cortex-M hardware
- Based on open source software and open standards
- With AI/ML workloads as a test case

Project Goals

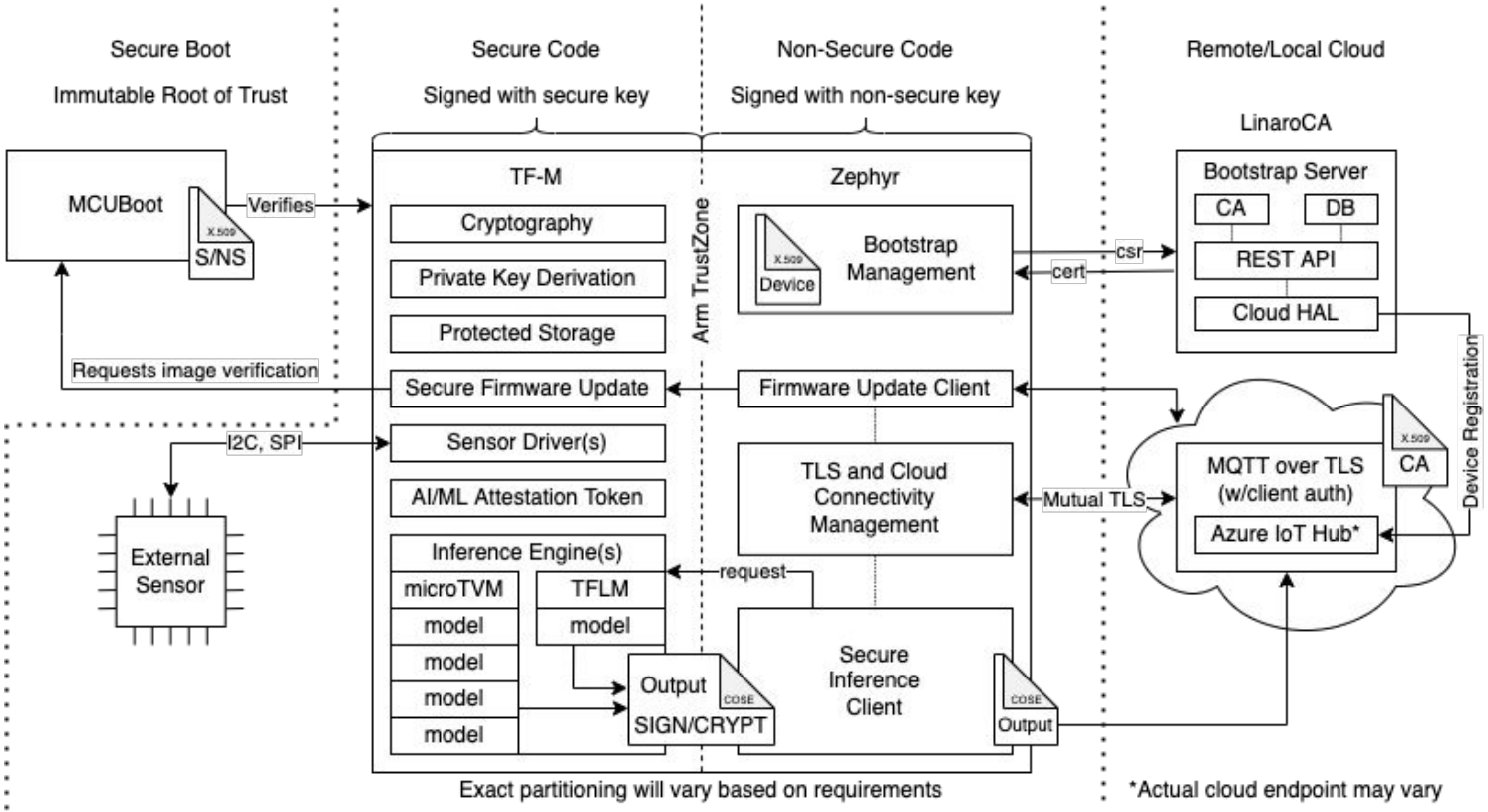
- Vendor neutral:
  TFLM or MicroTVM
  Cloud vendor neutral
- Emulation-friendly:
  mps2_an521 (M33)
  mps3_an547 (M55)
- Open source firmware:
  MCUBoot, TF-M, Zephyr
- Open standards:
  TLS, X.509 certificates, COSE

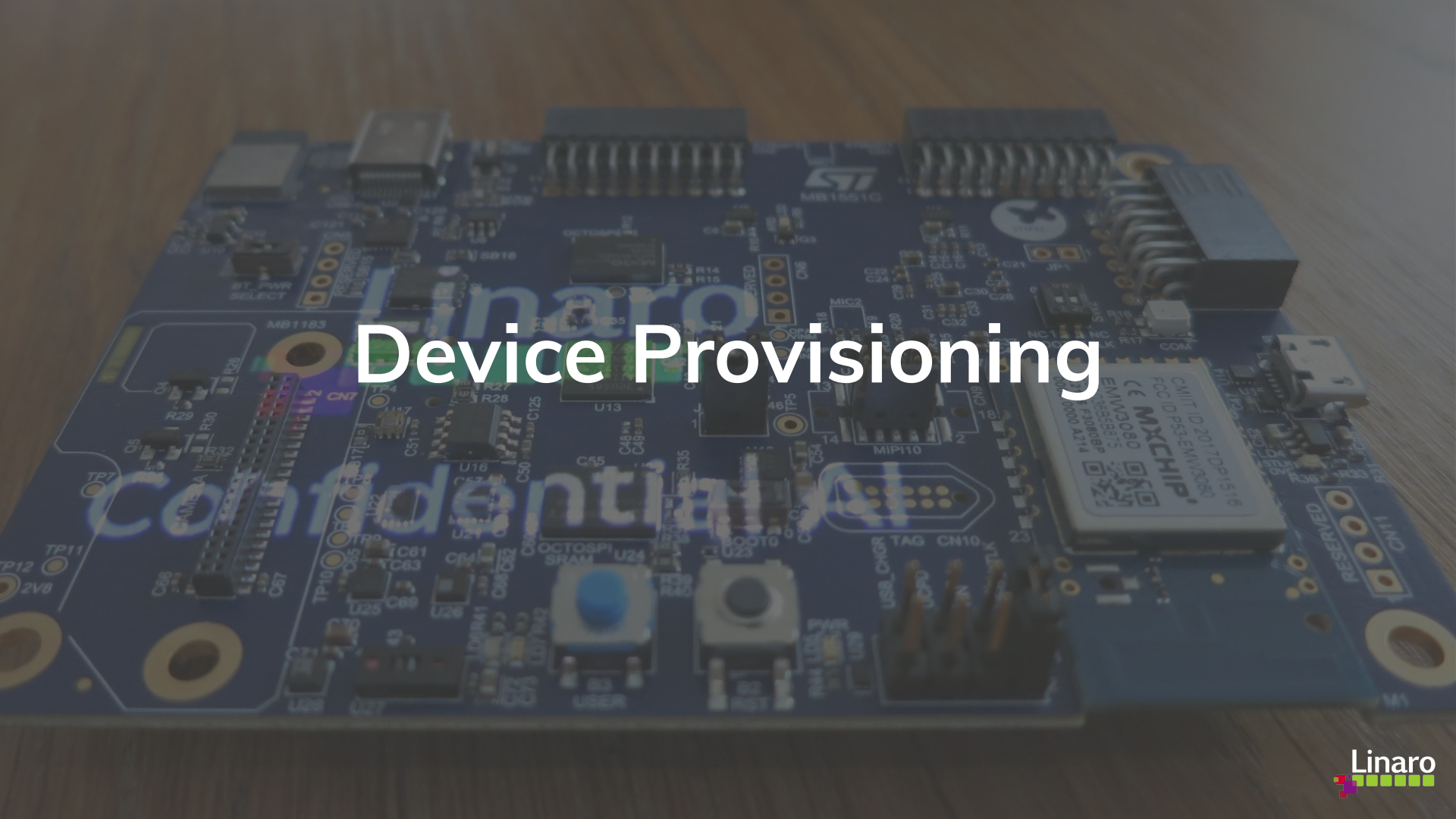
What would that look like … ?

Core Components in a Secure IoT System



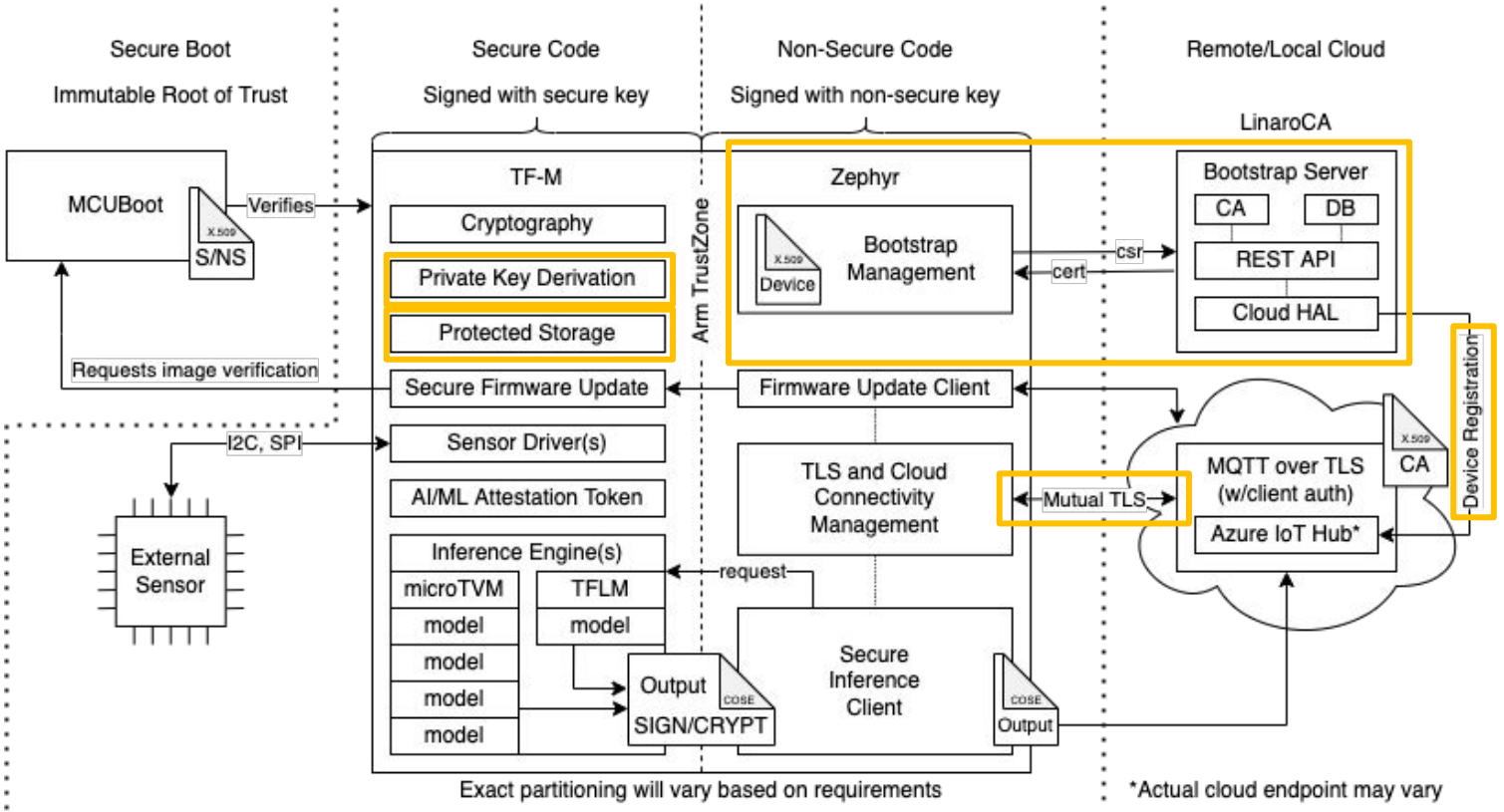
Confidential AI Proof of Concept





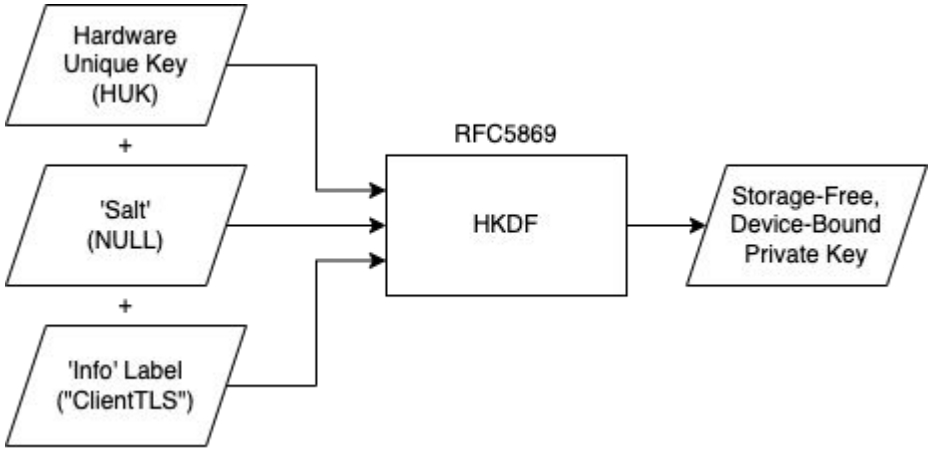
Device Provisioning

Device Provisioning



Best Practice: Storage-Free Key Derivation

- Safest way to store a key is never store it!
- Private key storage is high risk
- Derive device-bound key w/HUK
- Key regenerated at boot
- Persistent across updates
- Generate a CSR (MbedTLS, etc.)
- Send CSR to CA for signing
- X.509 cert stored in the open



* This same approach is also be used to derive a device UUID

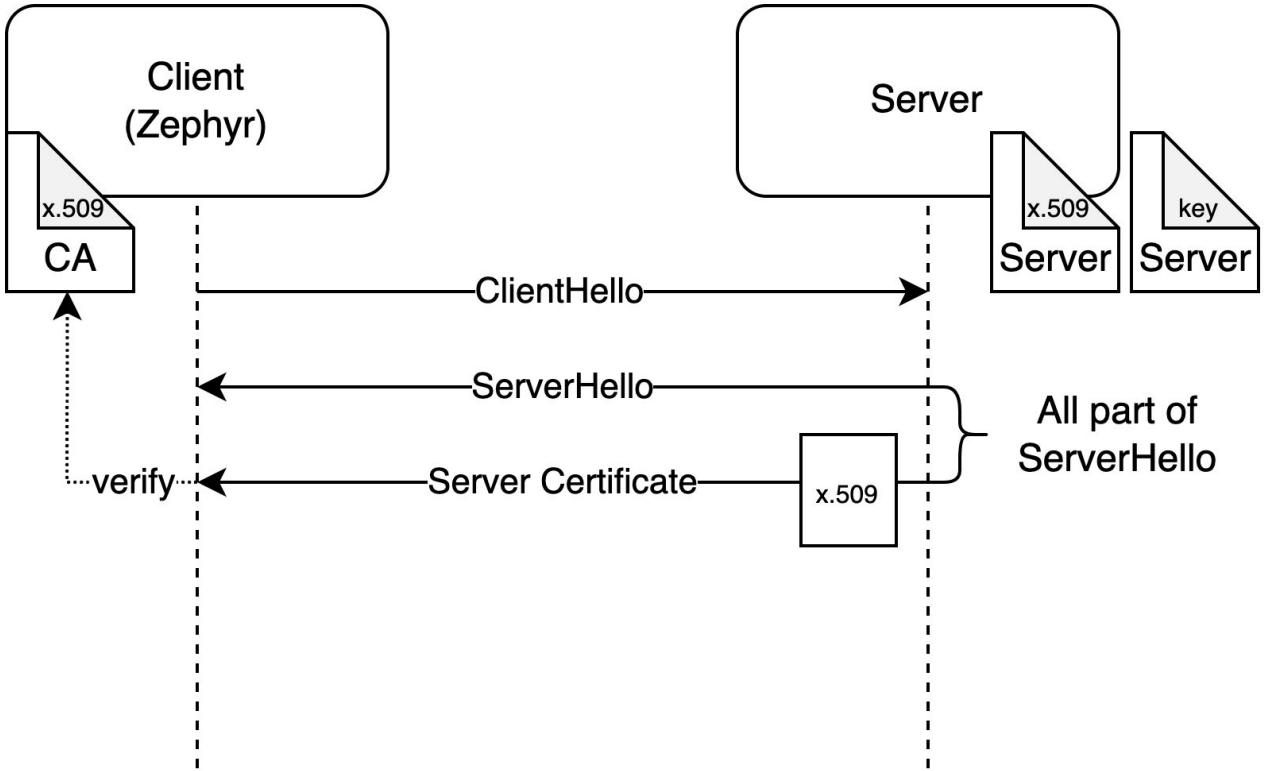
Linaro

Best Practice: Mutual TLS

- Basic TLS authentication only validates the **SERVER** identity
- TLS optionally includes **Client Authentication**, where the server also asks the client device to provide proof of it's identity

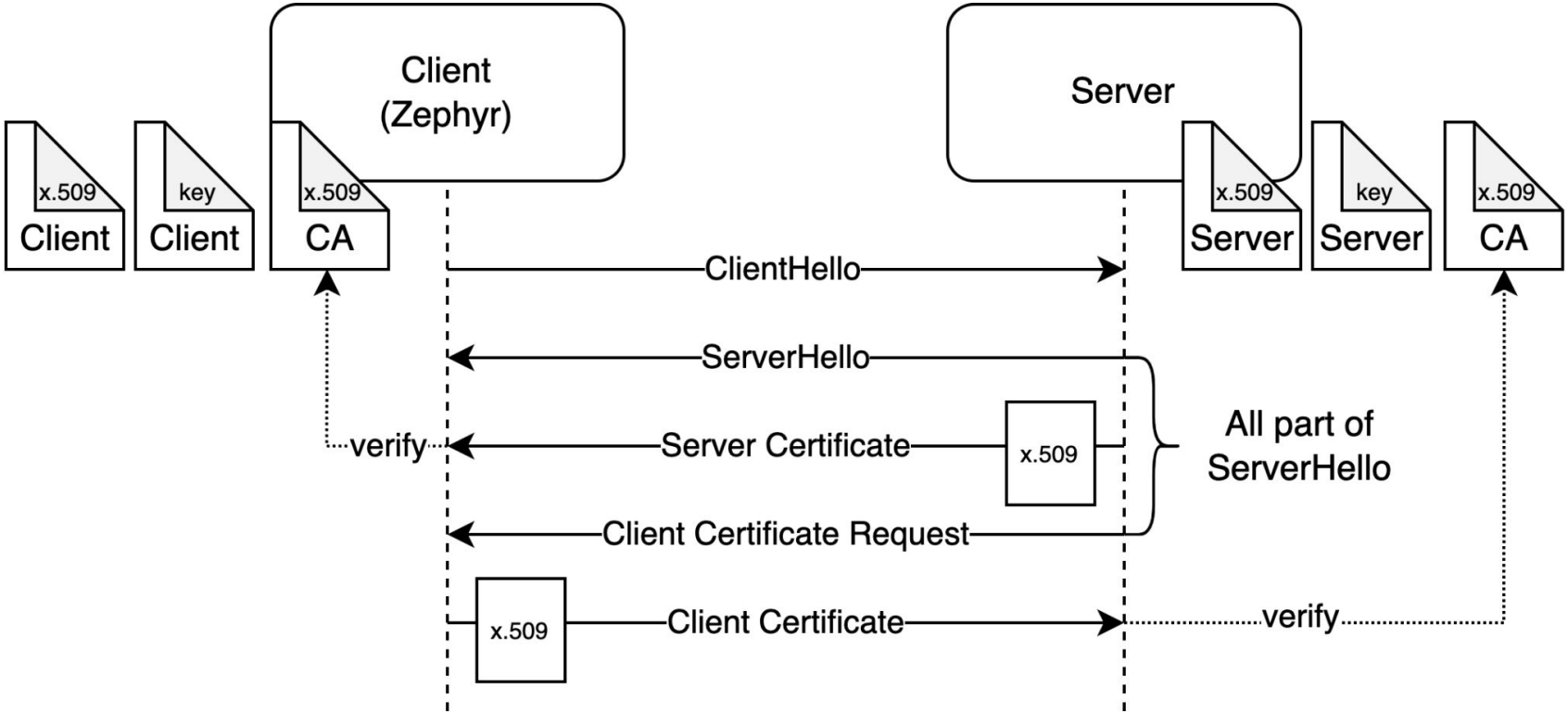
Linaro

Basic TLS

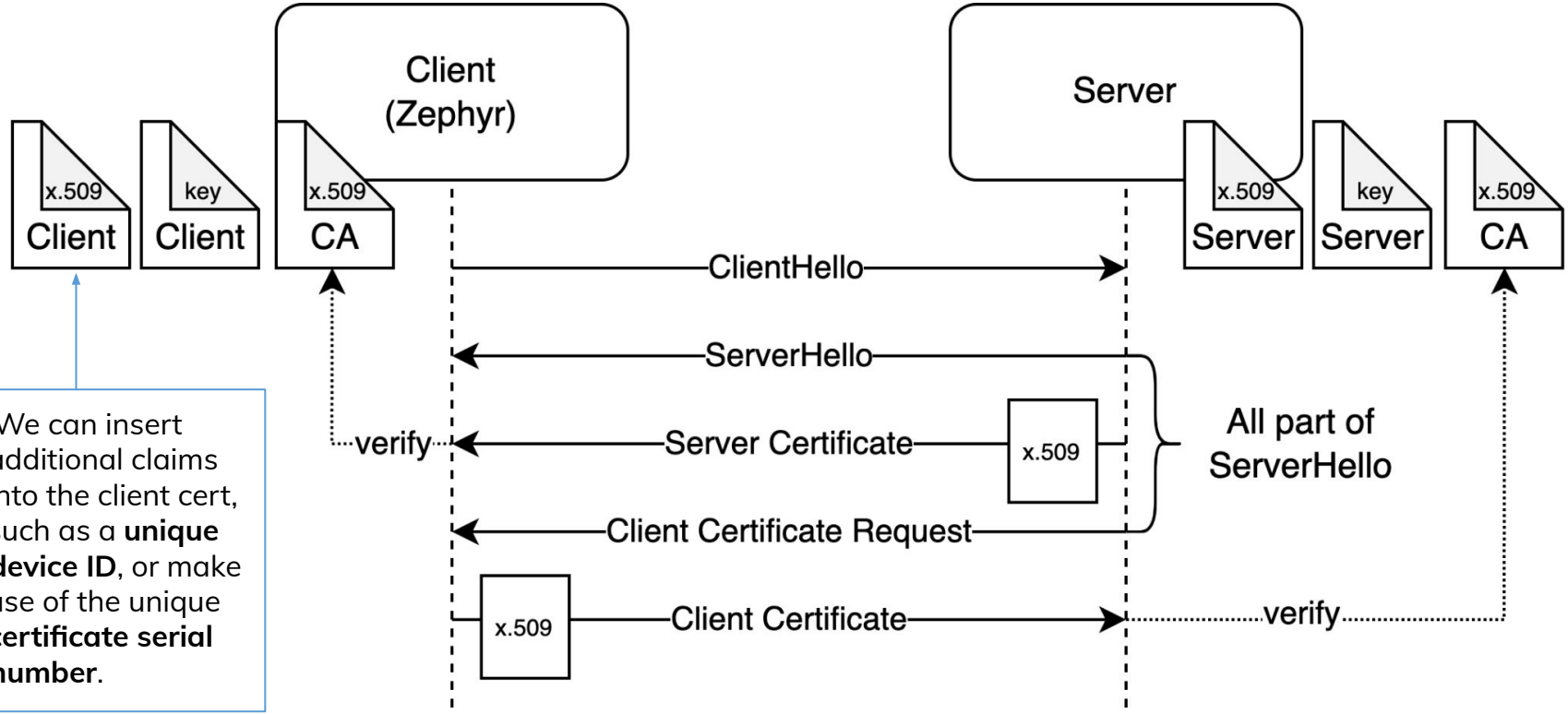


Linaro

Mutual TLS



Client Authentication?



⚠ We can insert additional claims into the client cert, such as a **unique device ID**, or make use of the unique **certificate serial number**.

Linaro

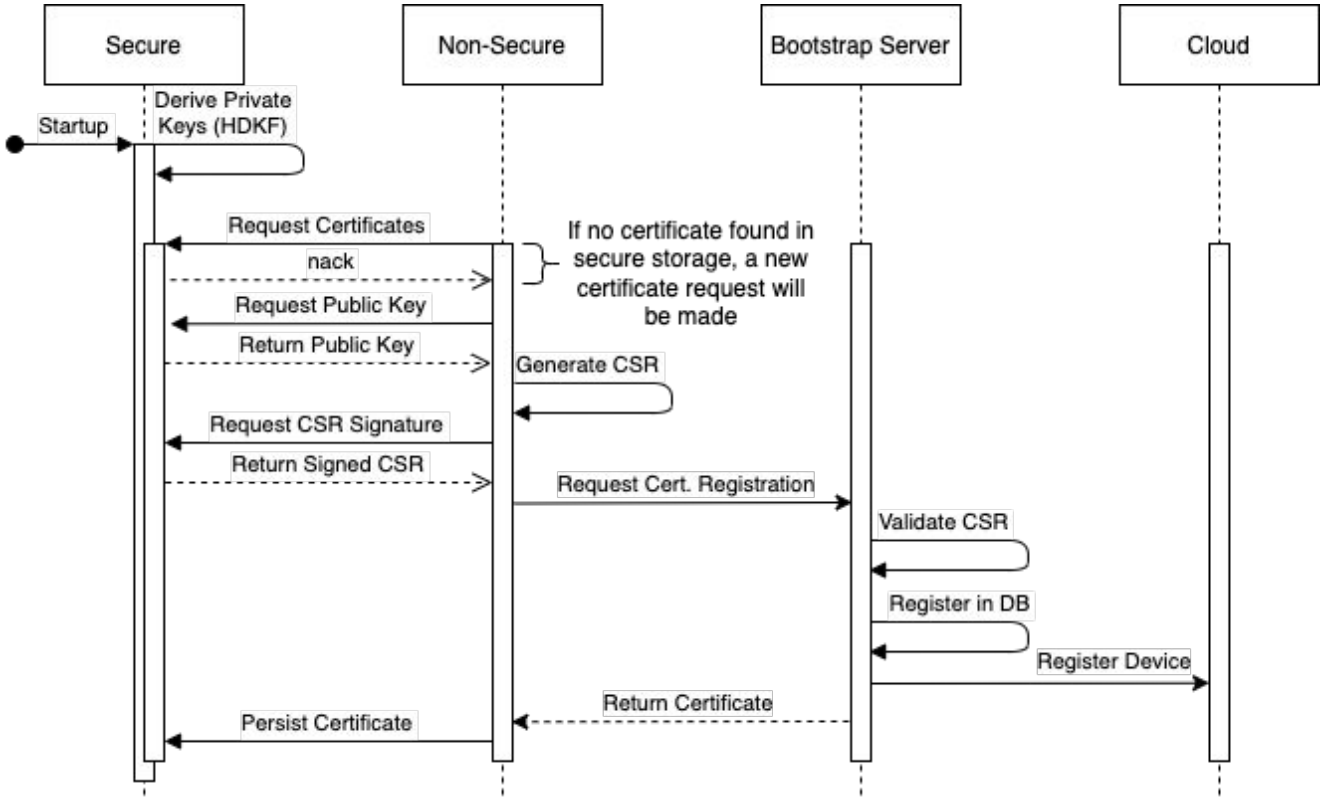
LITE Bootstrap Server

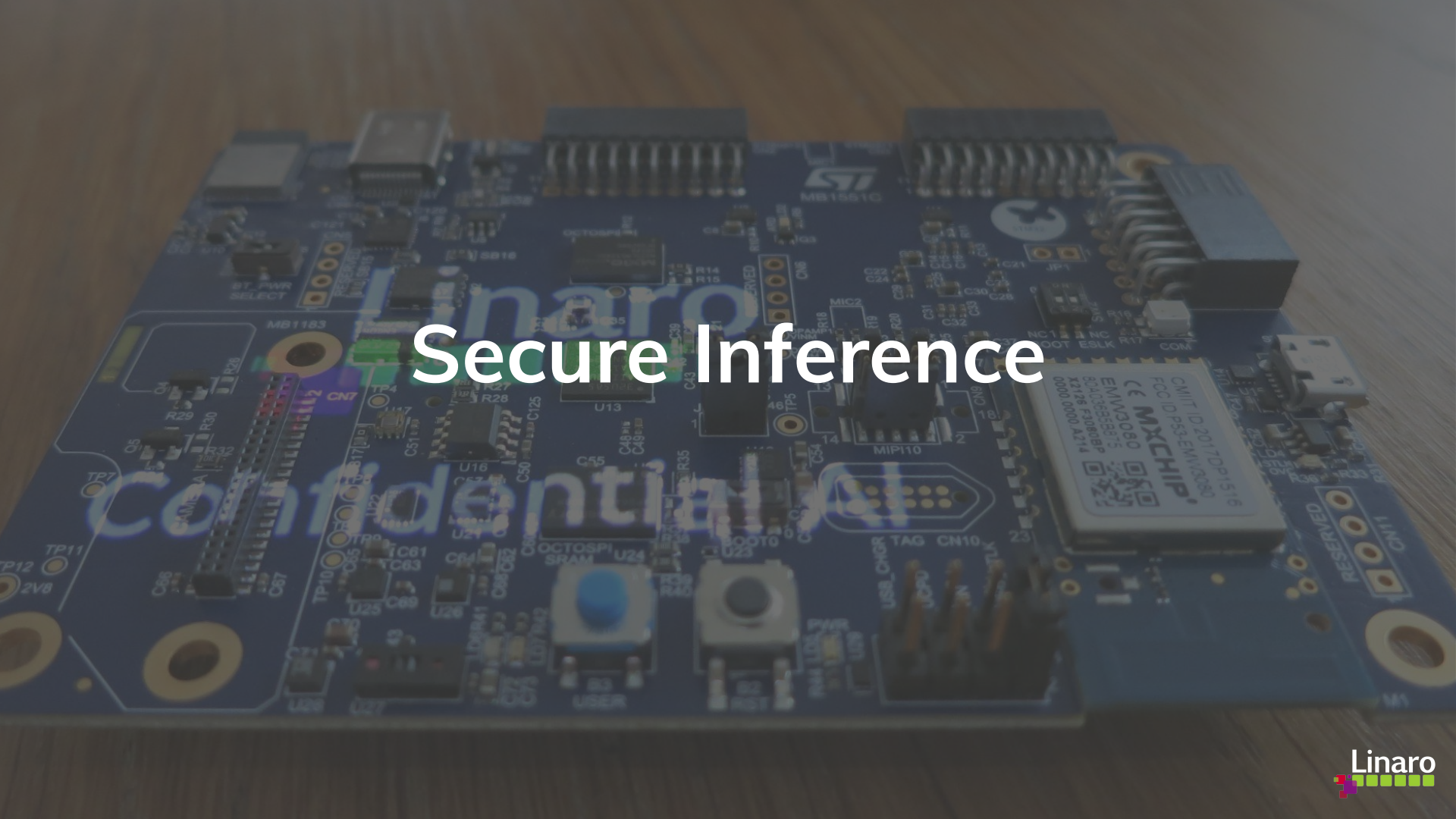
- Written in golang
- Basic Certificate Authority (CA) functionality:
  - Certificate Signing Request (CSR) processing
  - Verify certificate status
  - Certificate updates, etc.
- Authenticated REST API
  - Requiring a pre-shared X.509 certificate and key to connect to REST API
- Callback to register devices on **cloud provider(s)** during CSR processing
- Secondary TCP server demonstrating Mutual TLS on the server side
- **Proof of concept!**

Github: https://github.com/Linaro/lite_bootstrap_server

Linaro

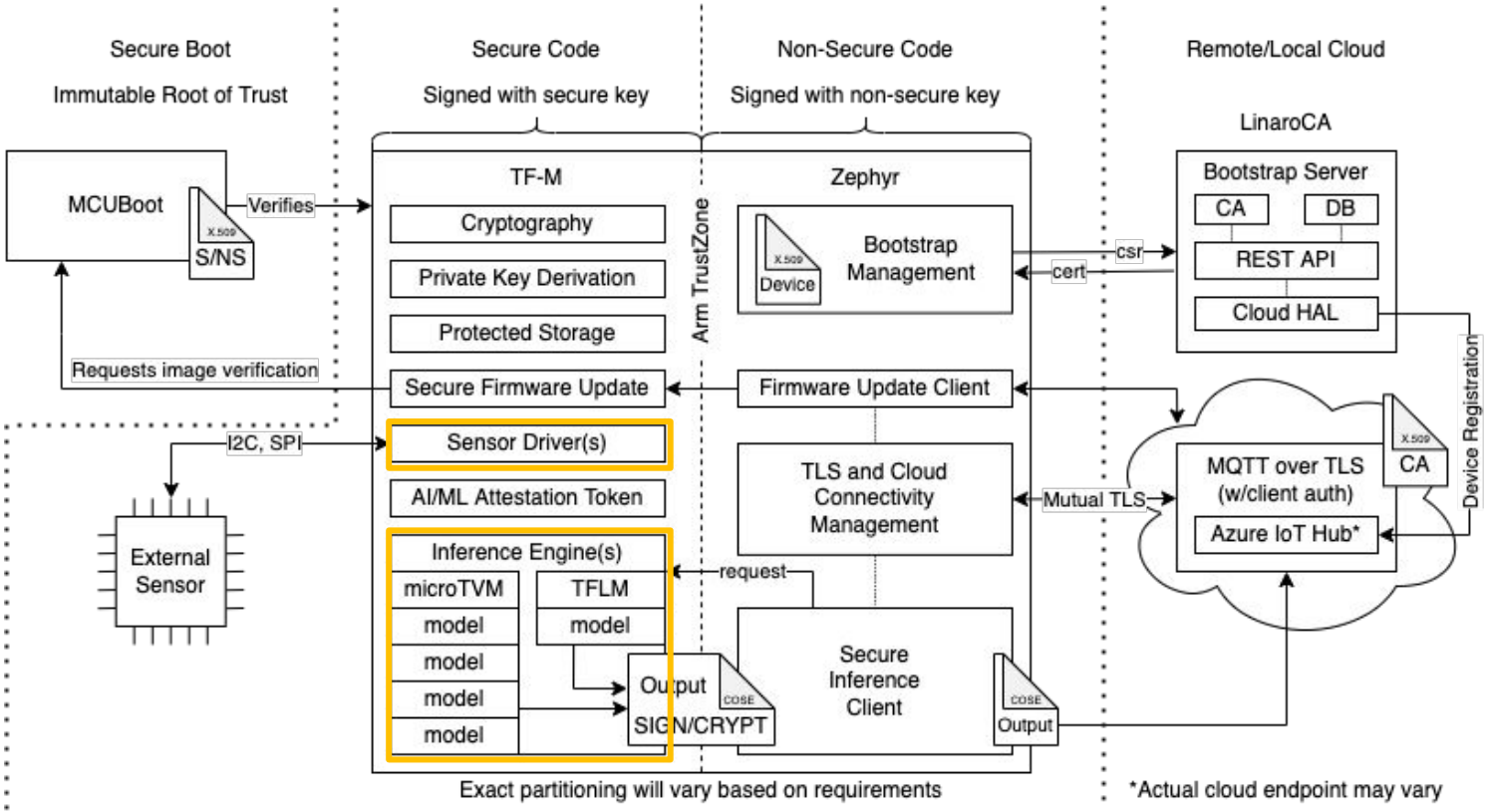
X.509 Certificate Provisioning Workflow

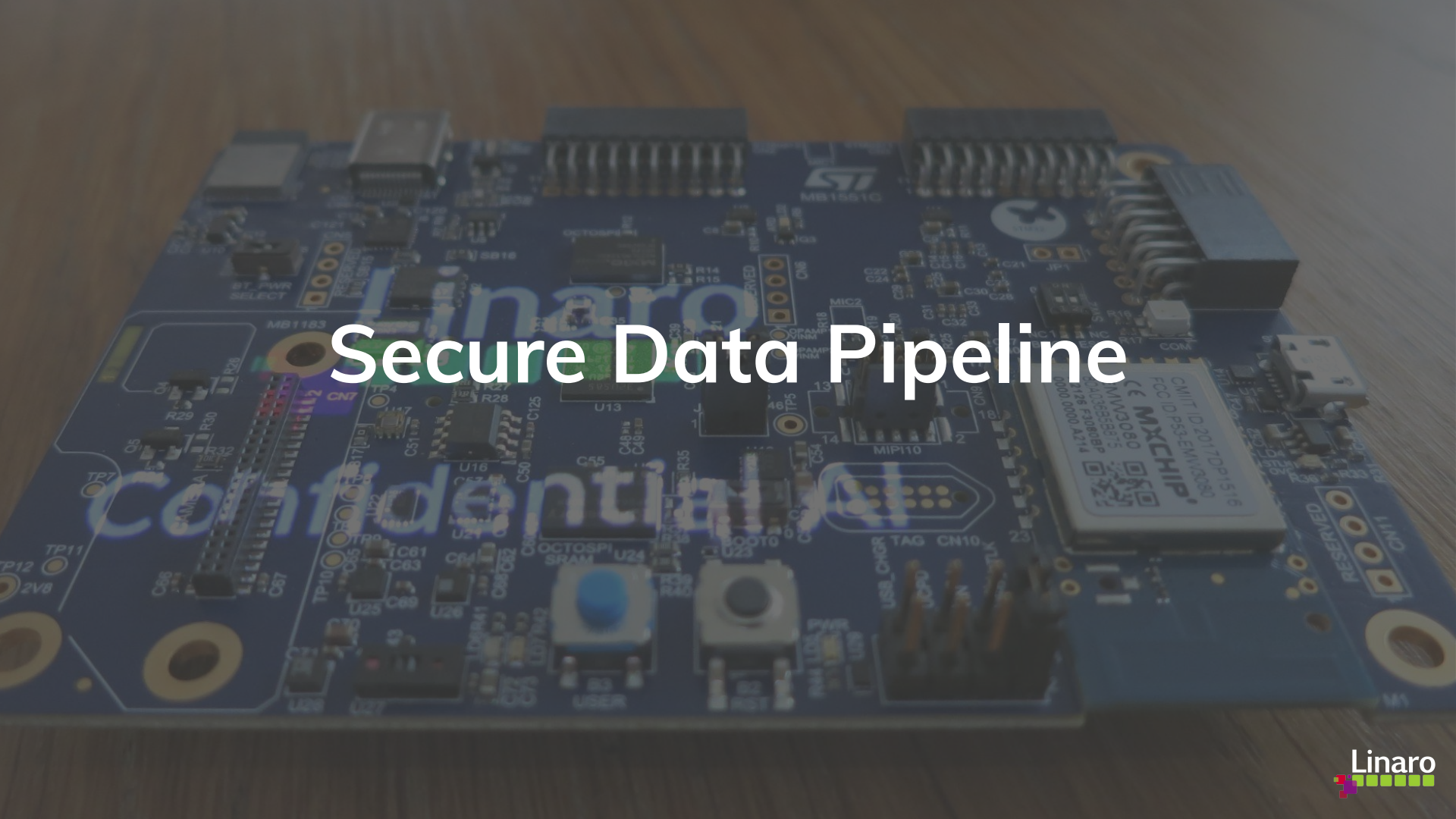




Secure Inference

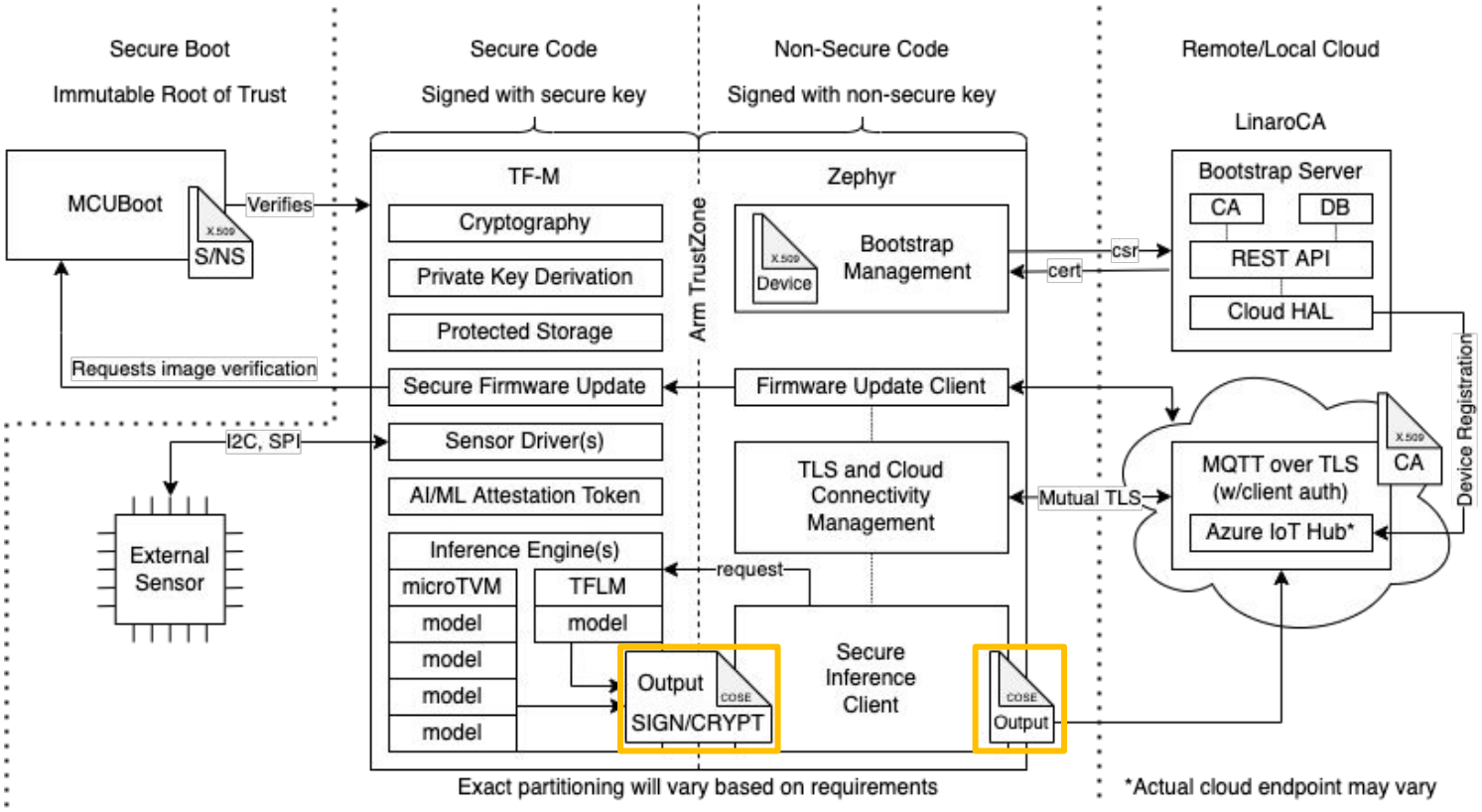
Secure Inference



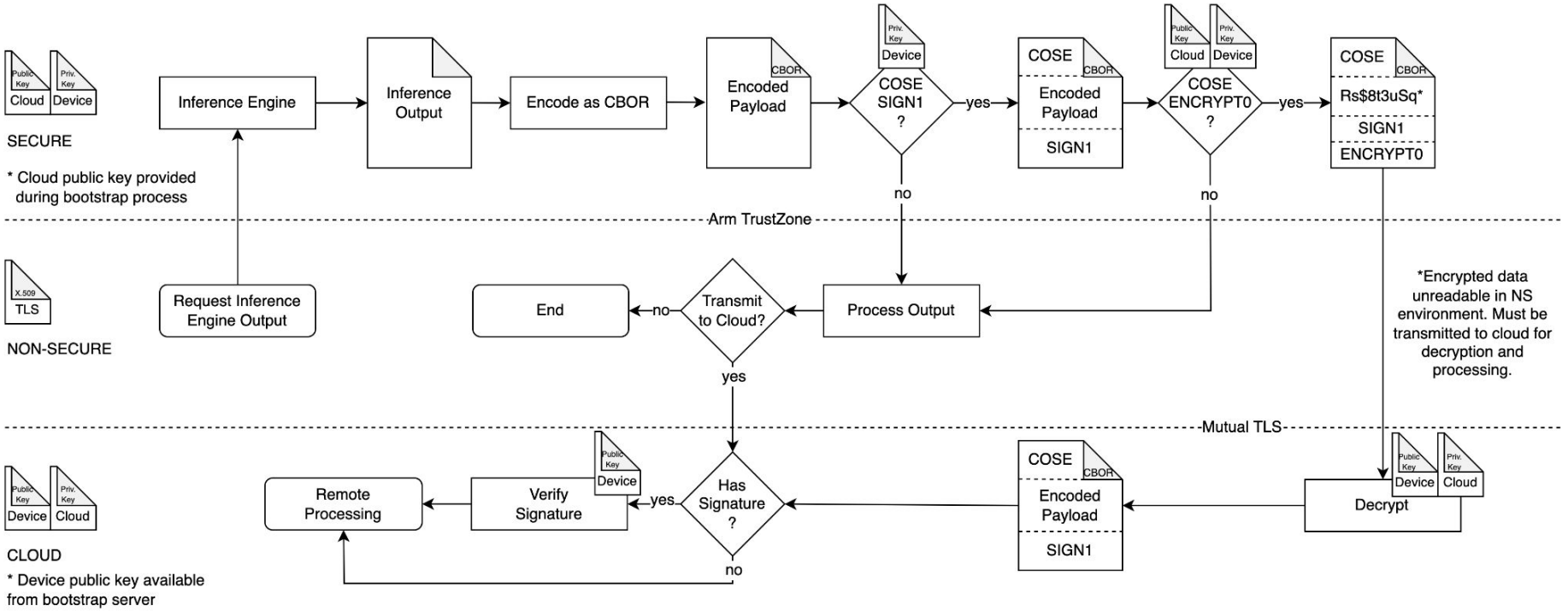


Secure Data Pipeline

Secure Data Pipeline



Secure Data Path



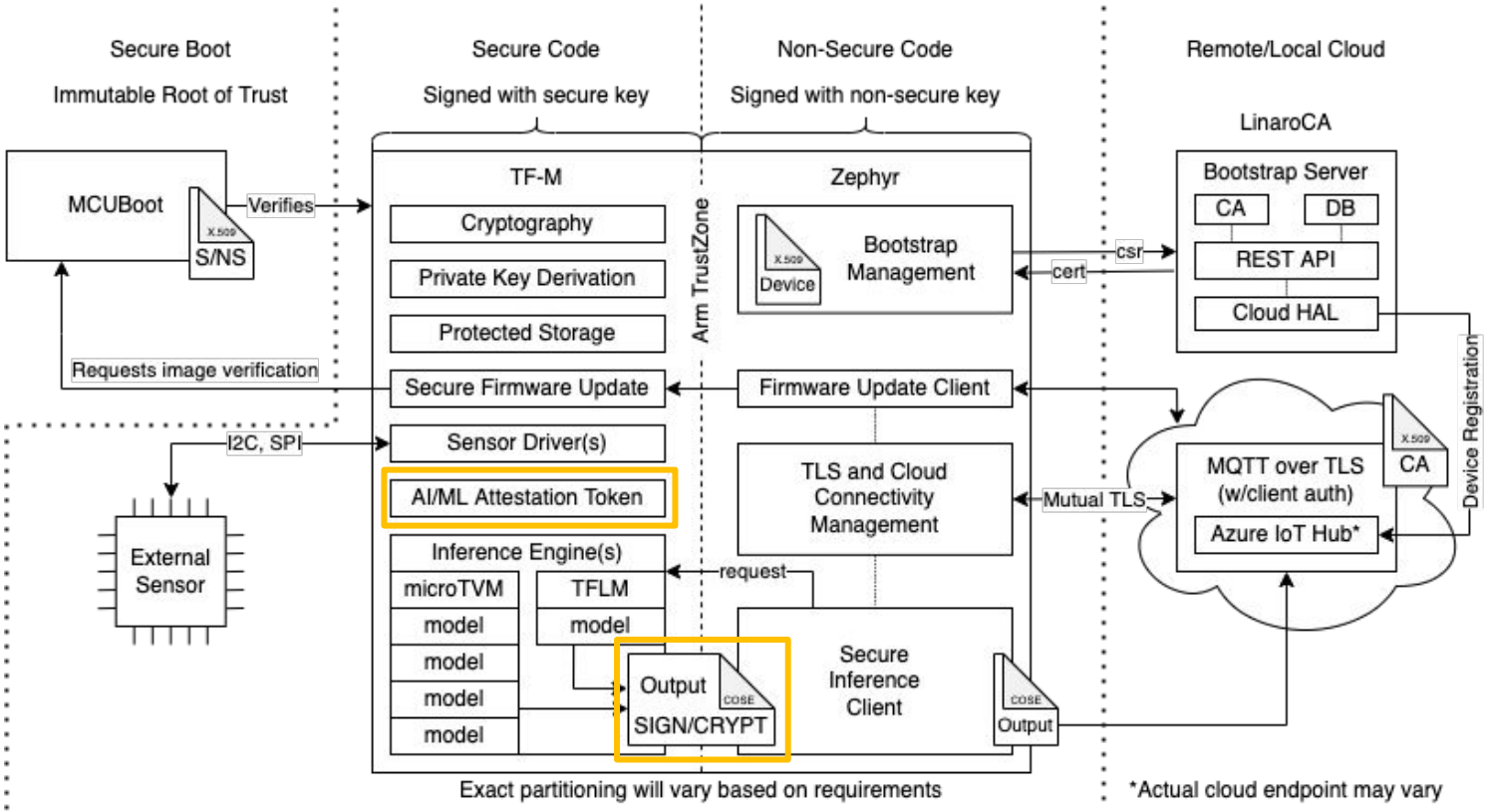
COSE Payload Encryption

- **TLS may not always be available**, or you may need to protect payloads before or after the TLS connection (untrusted NS firmware, public MQTT endpoint, etc.)
- COSE allow for flexible encryption using modern cyphers, but **performance is problematic** (particularly signing operations)
- COSE encryption is new compared to COSE signing (as used in attestation tokens)
  - Library support for ENCRYPT/ENCRYPT0 is still very poor
  - C libraries like t_cose are making an effort to improve this, but still a WIP
- Our initial proposal for **Efficient COSE encryption** was presented at Linaro Connect 2023: LHR23-313: Secure IoT Data Flow

- Current Rust demonstration of this approach ('**flow**') is available at: https://github.com/Linaro/zephyr_confidential_ai/tree/main/tools/flow

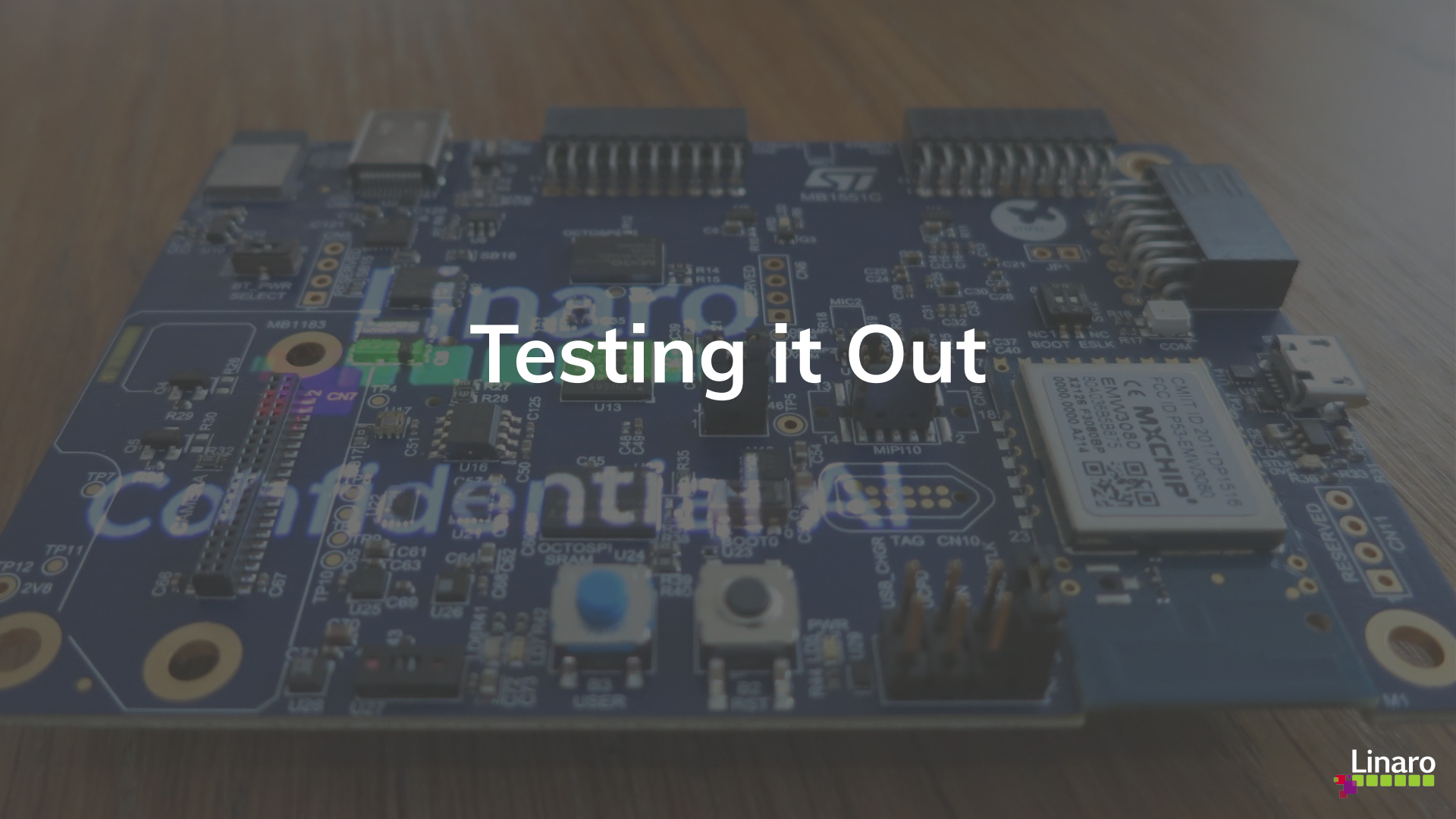
Linaro

Attestation Tokens: What from Where, and When?



AI Attestation Token

- **Entity Attestation Tokens (EAT)** allow devices to make **claims** about their status
- This can include:
  - Device ID
  - Software and hardware versions
  - Inference engine and model versions
- Model outputs are associated with a specific attestation token
- Tokens can be transmitted during device connection, or on system state changes
- Why bother?
  - Defective or malicious models can and eventually will be deployed
  - Attestation tokens allow us to know which outputs to reject or remove in the data aggregation phase once a security issue is identified


Testing it Out
Linaro
Confidential

Platform: QEMU

- QEMU has Arm v8M-profile support including emulated security functions such as:
  - MPU (Memory Protection Unit Extension)
  - PXN (Privileged Execute Never)
  - S (Security Extension)

- We used the following platforms emulated in QEMU, which also include networking support:
- mps2_an521_ns (Arm Cortex M33)
  - MPS2 is an FPGA-based Arm development board
  - It runs an Arm model defined in Application Note AN521
- mps3_an547_ns (Arm Cortex M55)
  - MPS3 is an evolution of MPS2
  - It runs an Arm model defined in Application Note AN547
  - This model is also known as Arm® Corstone™ SSE-300

Application Code and Component Repositories

← Confidential AI Proof of Concept Application:
https://github.com/Linaro/zephyr_confidential_ai

Open Source Components:

- LITE Bootstrap https://github.com/Linaro/lite_bootstrap_server
- MCUBoot https://github.com/mcu-tools/mcuboot
- TF-M https://git.trustedfirmware.org/TF-M/trusted-firmware-m.git/
- Zephyr RTOS https://github.com/zephyrproject-rtos/zephyr
- MicroTVM https://tvm.apache.org/docs/topic/microtvm/index.html
- TFLM https://www.tensorflow.org/lite/microcontrollers
- MbedTLS https://github.com/Mbed-TLS/mbedtls
- COSE https://github.com/laurencelundblade/t_cose

Project contact details: confidential_ai@linaro.org

Linaro

Linaro