ESPRESSIF

Segurança e Conformidade do Dispositivo ESP32

Um panorama de como proteger seu projeto

Amey Inamdar

Diretor de Marketing Técnico

ESPRESSIF

ESP32 Device Security and Compliance

An overview of how to secure your project

Amey Inamdar

Technical Marketing Director

Why security and compliance matter

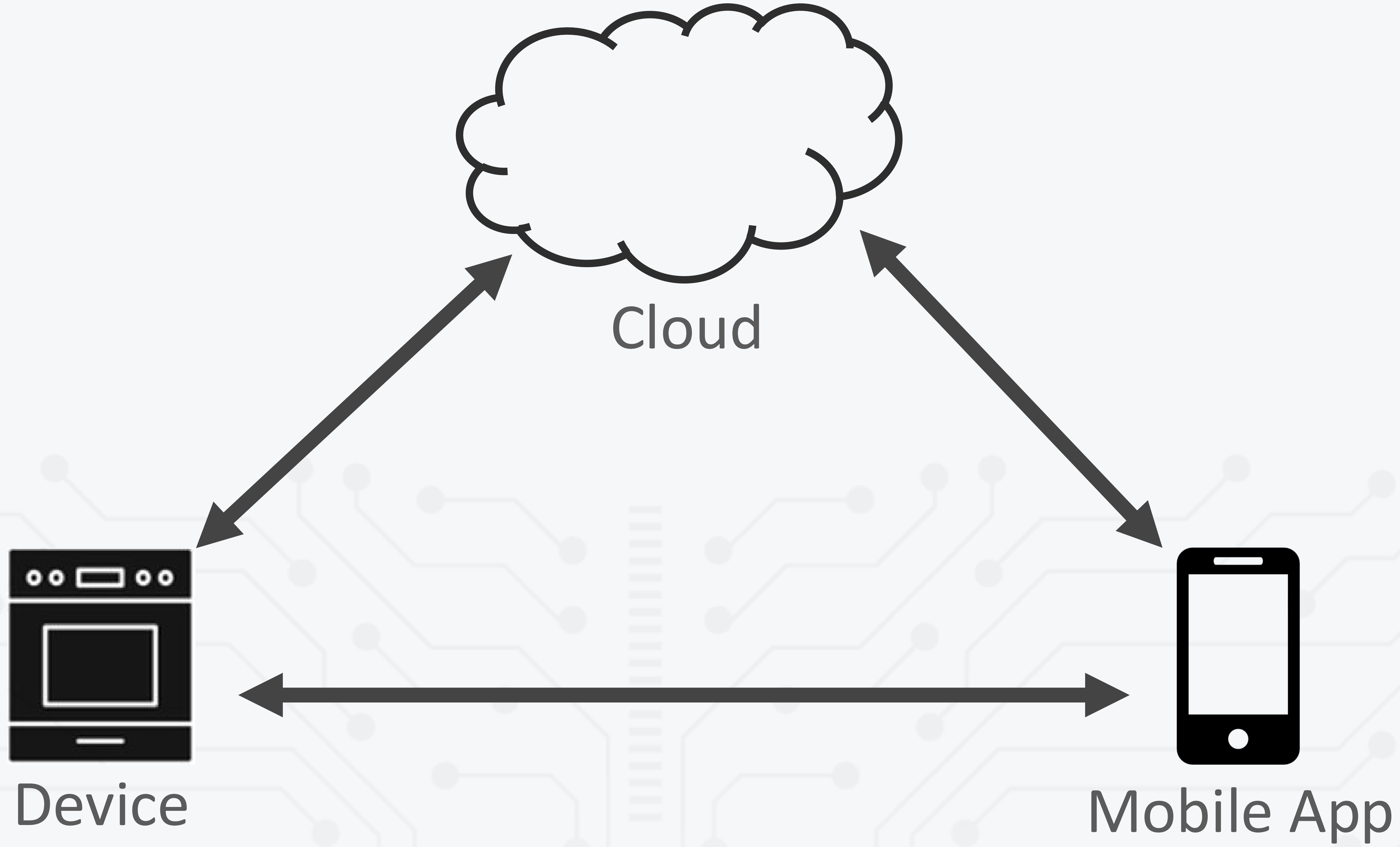
✓ Business and Technical Risks

- IoT breaches may take several months to identify and contain
- Attacks can persist into unintended physical processes
- High containment and downtime costs
- Customer fear and tarnished company image

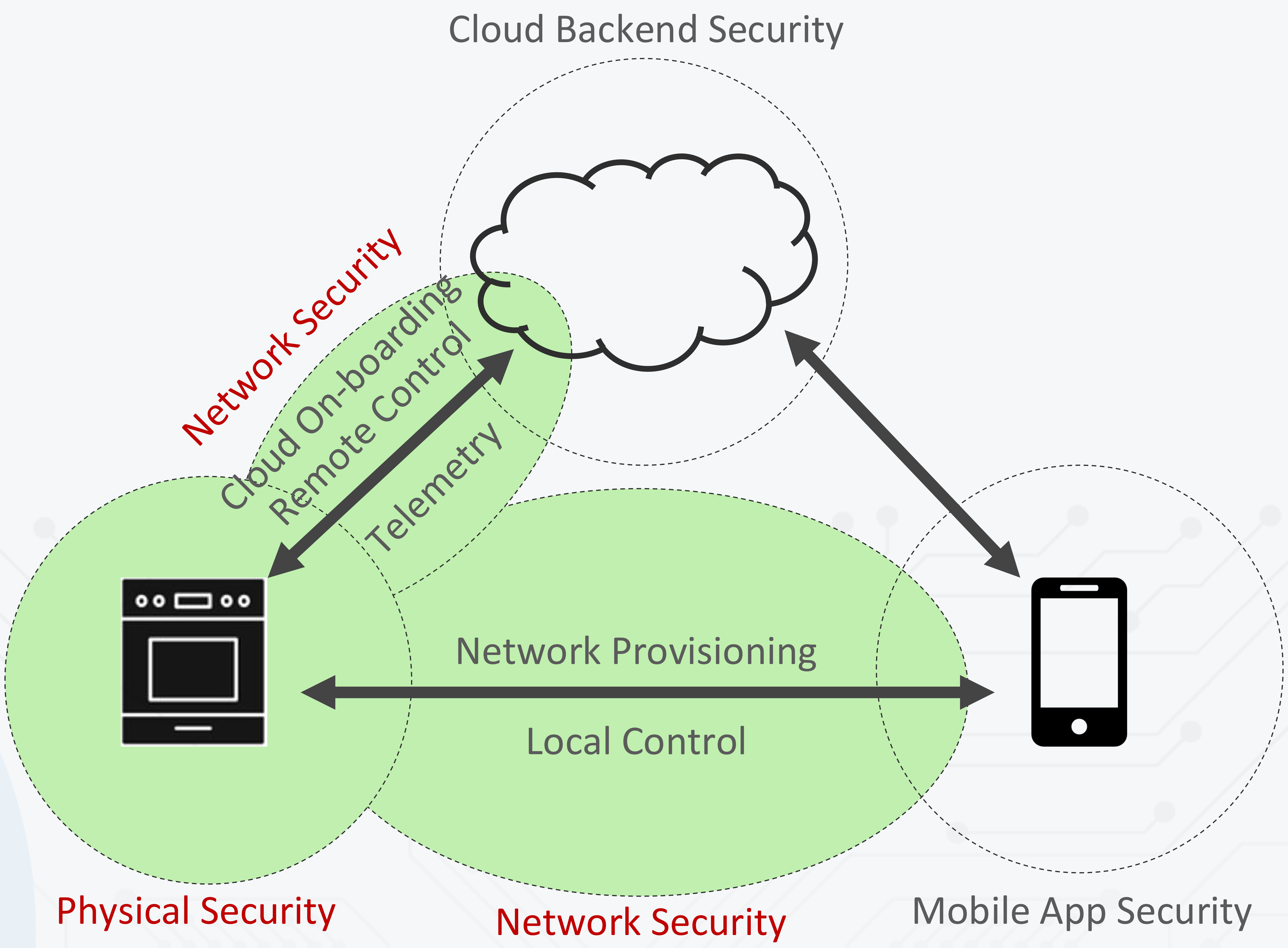
✓ Global Regulatory Pressure

- Many countries in process of defining Cybersecurity compliance for devices
- Non-compliance may result in severe fines
- Compliance provides peace-of-mind to consumer and helps in your product branding

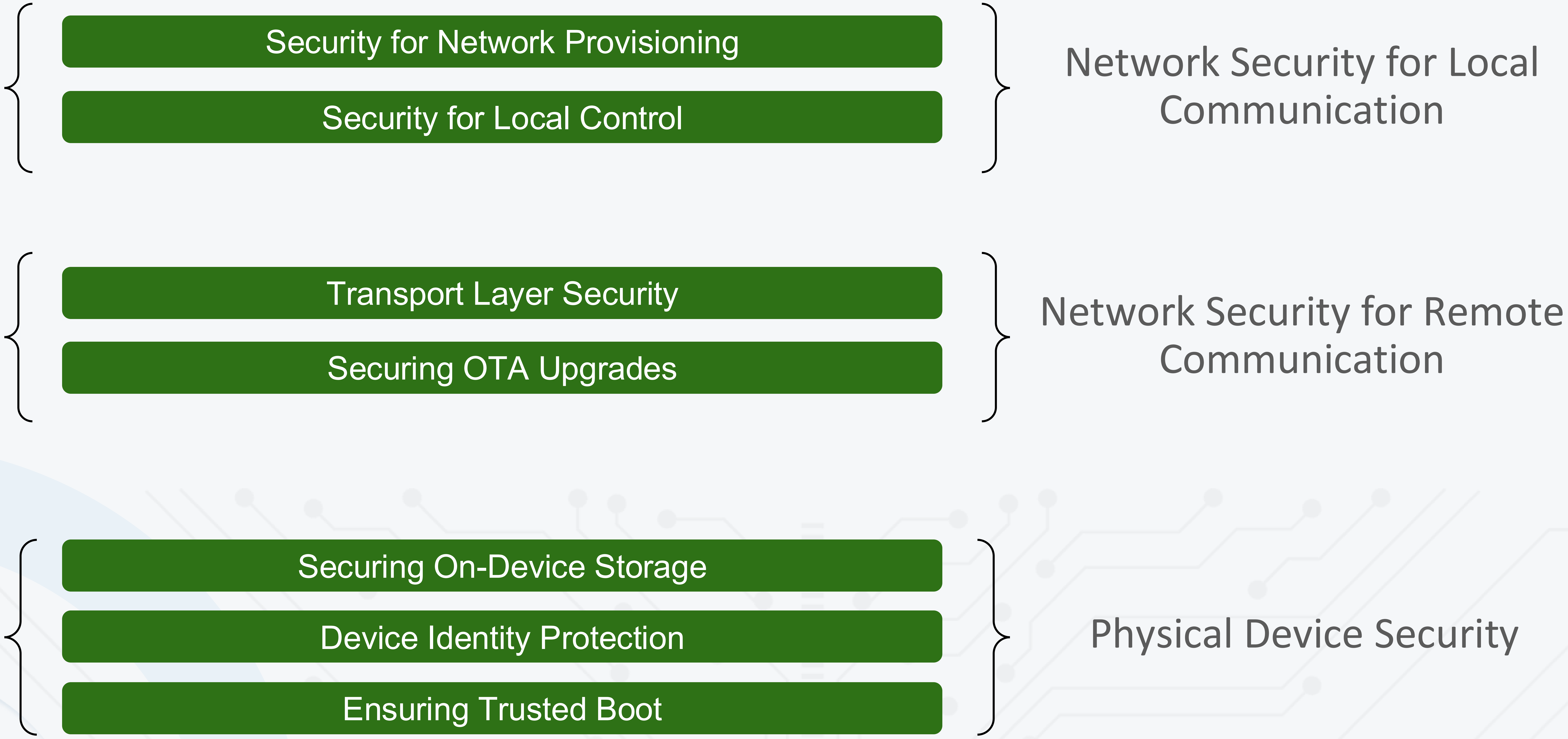
Anatomy of the IoT Project



Security of the IoT Project



Security for Each Layer



Network Security for Local Communication

- Security for Network Provisioning
- Security for Local Control

Network Security for Remote Communication

- Transport Layer Security
- Securing OTA Upgrades

Physical Device Security

- Securing On-Device Storage
- Device Identity Protection
- Ensuring Trusted Boot

Secure Network Provisioning

- ✓ Key Design Considerations
  - Ensuring Transport Security to avoid snooping
  - Ensuring Proof-of-possession to avoid MITM attacks and unintended provisioning
  - Retrieved Network Credentials need to be securely stored
- ✓ Espressif offers Unified Provisioning, SmartConfig, Blu-Fi as provisioning methods that provide out-of-box security
- ✓ Standard network provisioning methods include Matter, Apple WAC (HomeKit), WiFi EasyConnect (DPP)

Secure Local Control

- ✓ Key Design Considerations
  - Trust model and User Authentication
  - Transport security
- ✓ Unified Provisioning offers application level authenticated sessions
- ✓ Wi-Fi – TLS + Certificate Pinning – A common security practice
- ✓ BLE – Secure Pairing
- ✓ Matter, HomeKit offer their own secure local control

Transport Layer Security

- ✓ TLS provides a standardized authentication and privacy for internet connectivity
- ✓ IoT devices typically use MQTT over TLS with certificate-based mutual authentication and server authenticated HTTPs (for file transfers) protocols
- ✓ ESP-IDF provides TLS protocol implementation using MbedTLS and WolfSSL (third-party component)
- ✓ Device Identity – X.509 certificate for the device
- ✓ Certificate Bundle – For server authentication

Securing OTA Upgrades

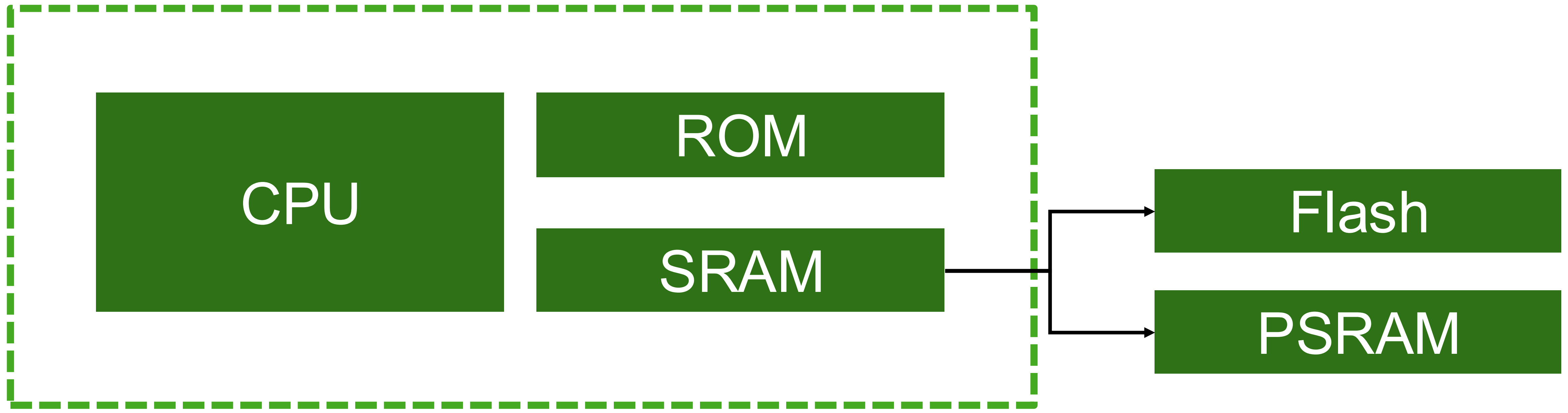
- ✓ OTA Upgrades are essential for connected devices for bug fixes, security vulnerability fixes and feature improvements
- ✓ Key Design Considerations
  - Fetch OTA upgrade images from trusted source
  - Ensure that the upgrade image is authentic
  - Ensure no forced rollbacks are possible
- ✓ OTA Upgrades over HTTPs or MQTTs with authenticated server
- ✓ Espressif OTA implementation checks the image for valid secure boot signature
- ✓ Anti-rollback mechanism in Espressif SW Bootloader to disallow roll-back to insecure OTA images

Securing OTA Upgrades (cont..)

✓ For privacy of OTA image stored on the upgrade server, Espressif offers an implementation of pre-encrypted OTA that is based on a PKI

Espressif SoC Architecture

- Typically the application is executed from flash using XIP
- Flash and PSRAM are accessed through cache which is part of SRAM
- PSRAM and Flash can be in-package, in-module or sometimes outside the module



Securing On-Device Storage

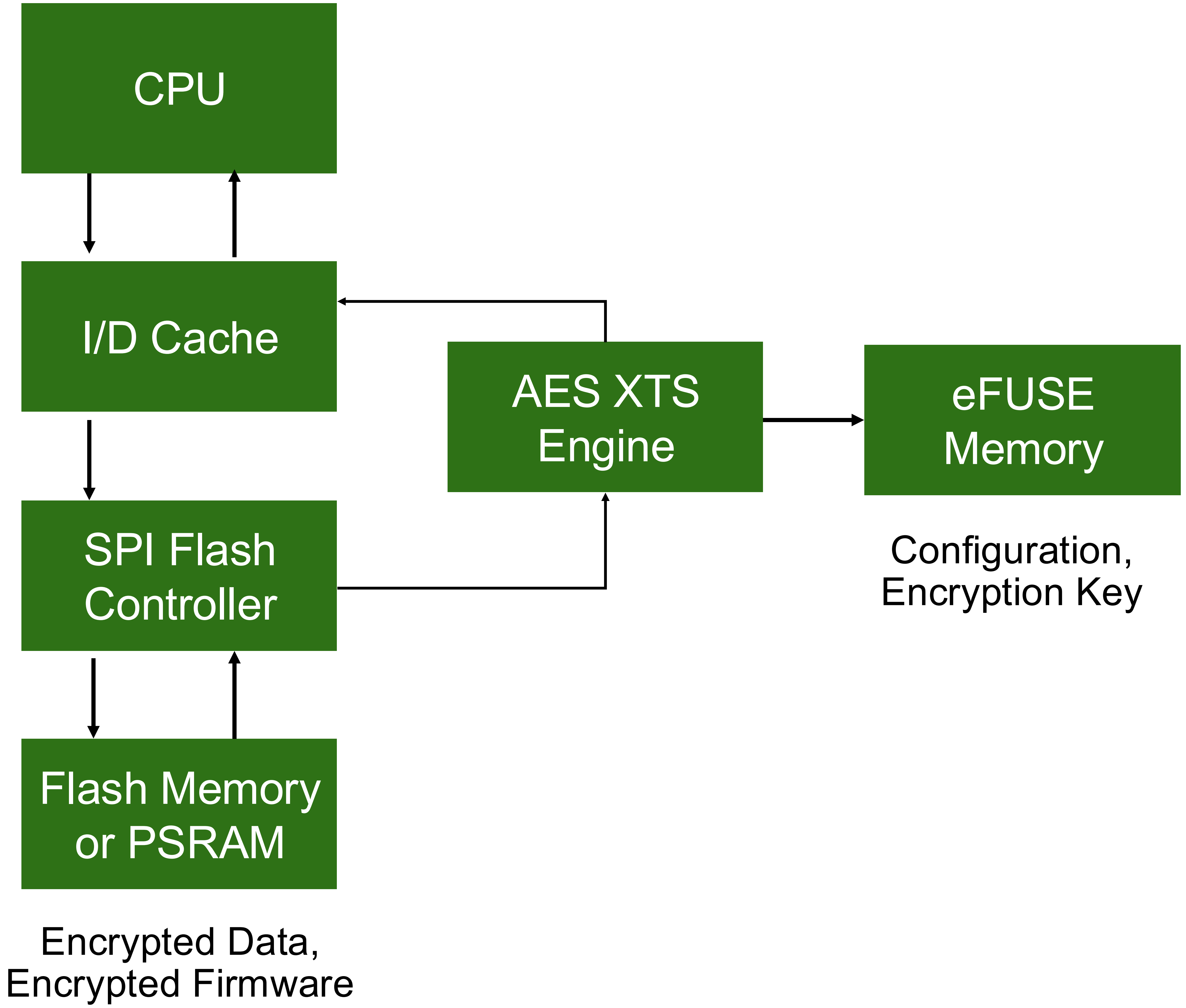
✓ Following critical data resides on the device and needs to be secured from unauthorized direct access

- Network Credentials
- Device Identity (Private Key of the device)
- Application specific critical data
- Application Firmware (some customers want it to be protected to protect IP)

✓ Flash Encryption (external memory encryption) offers transparent on-the-fly encryption/decryption of the data stored in the flash (and PSRAM)

Flash Encryption

- XTS-AES block cipher mode with a 256-bit key size for flash encryption
- Flash encryption key is not accessible to the firmware and only accessible to AES XTS hardware
- All memory-mapped read/write accesses to flash are transparently encrypted or decrypted



Flash Encryption - Considerations

- ❑ ESP-IDF and tools provide flexibility to configure flash encryption and encryption of the plain text firmware on-chip or on-host

- ❑ Good practice - Ensure that flash encryption key is randomly generated unique to each device

- ❑ Good practice – If the manufacturing line is not fully trusted, generate flash encryption key on the chip; otherwise, you can generate flash encryption key on the host, encrypt the firmware image and program flash faster – BUT ensure that the key is still unique to each device

NVS Encryption

- □ NVS stores name-value pairs in fail-safe on-flash object store
- □ As it relies on flash erase-write property, flash encryption directly can’t be used to store NVS data
- □ ESP-IDF offers tools to generate and encrypt NVS partitions on the host or create empty encrypted NVS partition with key on-the-fly in the SDK

NVS Encryption (cont..)

- ❑ NVS encryption key can be protected using one of the two mechanisms

- ❑ NVS encryption key protected using flash encryption:
  - NVS Key is stored in a separate flash partition is a that holds the AES-XTS key. The key partition itself is secured using flash encryption
  - NVS data partition has metadata in plaintext and data is encrypted at the software layer using NVS encryption key and AES-XTS algorithm

- ❑ NVS encryption key protected using HMAC peripheral
  - In this scheme HMAC peripheral generates the AES-XTS NVS encryption key using the programmed HMAC key that is not software accessible

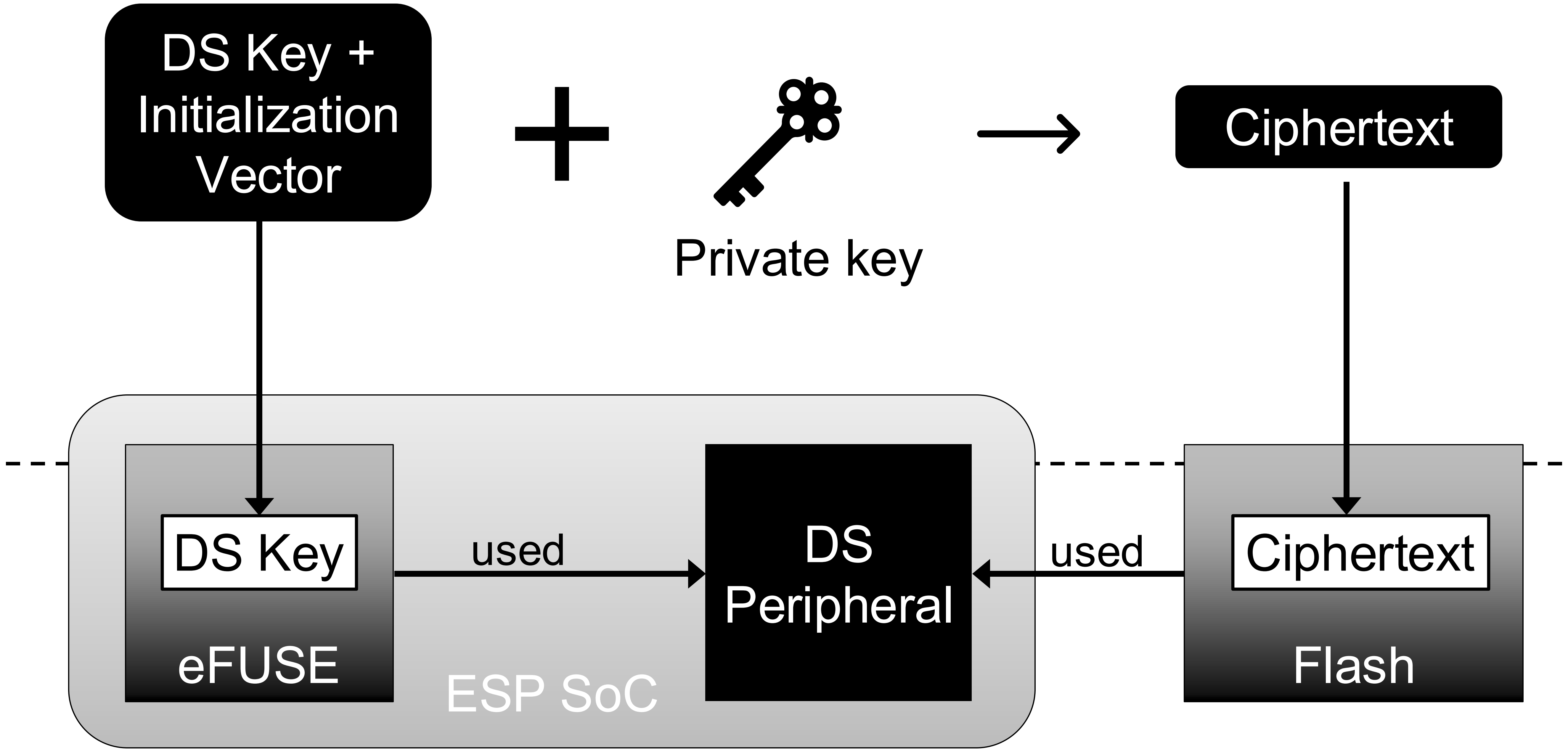
Device Identity Protection

- ❑ Devices typically authenticate themselves using PKI based digital certificate making it a device identity

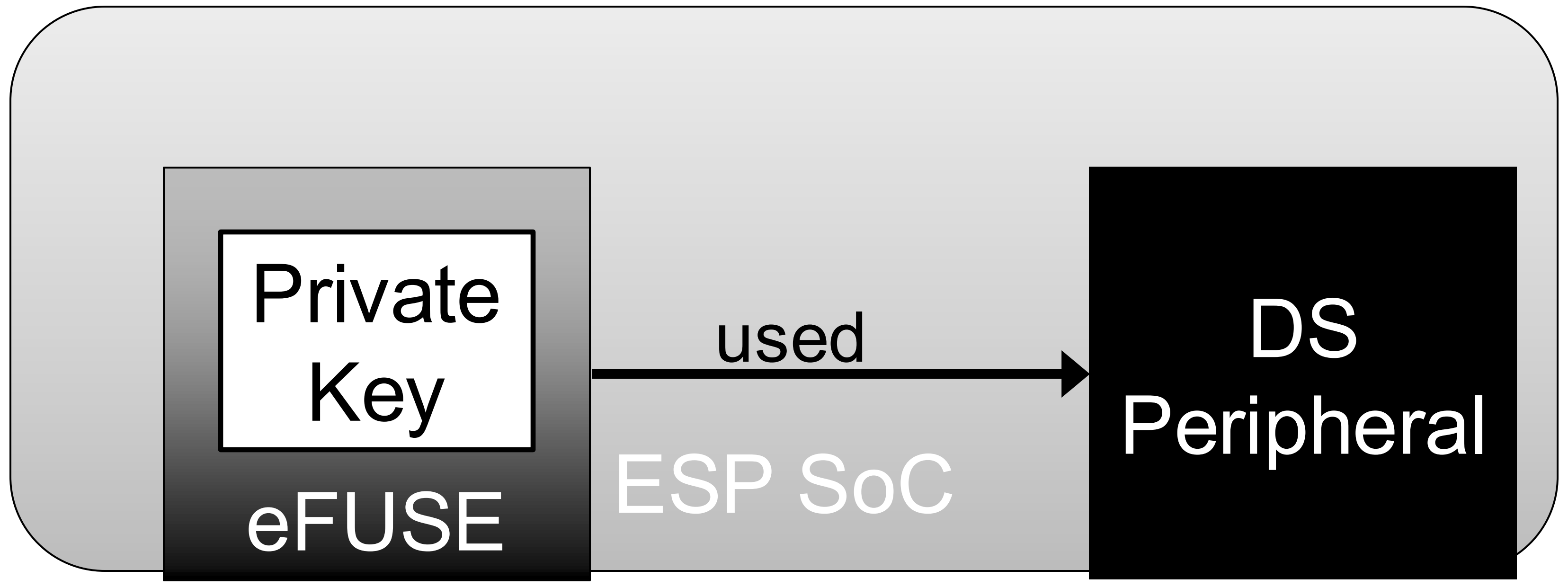
- ❑ While flash or NVS encryption can be used to protect private key, software vulnerabilities can still make the private key accessible

- ❑ ESP SoCs offer more secure mechanisms to let developers store and use the private keys using hardware protection
    - Use of Digital Signature Peripheral
    - Use of Trusted Execution Environment (explained later)

Digital Signature Peripheral



RSA DS Peripheral



ECDSA DS Peripheral

- SoC generates private keys on-chip, inaccessible to software, or physical attacks in plaintext

- Hardware-accelerated digital signatures are produced, allowing applications to perform signing operations with the encrypted device's private key without exposing

- Provisions to choose between an RSA or ECDSA-based certificates

Ensuring Trusted Boot

- ❑ Many of the security bets are off if the malicious firmware is executed on the device

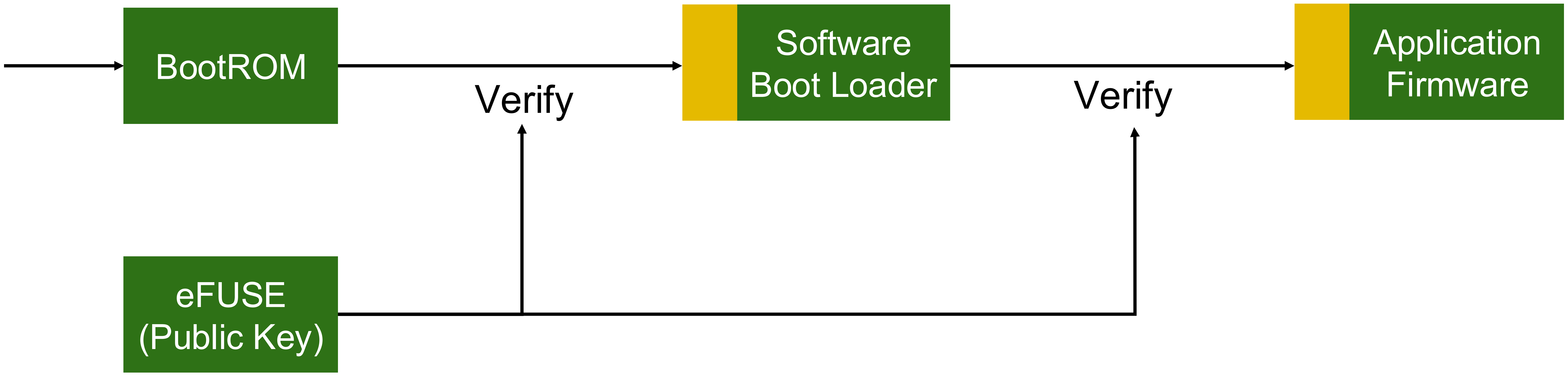
- ❑ Attackers can use existing firmware upgrade channels or software vulnerabilities that can invoke remote execution and then make the attack persistent by modifying the firmware

- ❑ ESP SoCs offer secure boot hardware feature that helps ensuring that only trusted code can be executed on the chip

Secure Boot

- BootROM uses the Public Key present in the eFUSE memory to check signature of the software boot loader

- Once authenticated, software boot loader uses the same mechanism to ensure authenticity of the application firmware



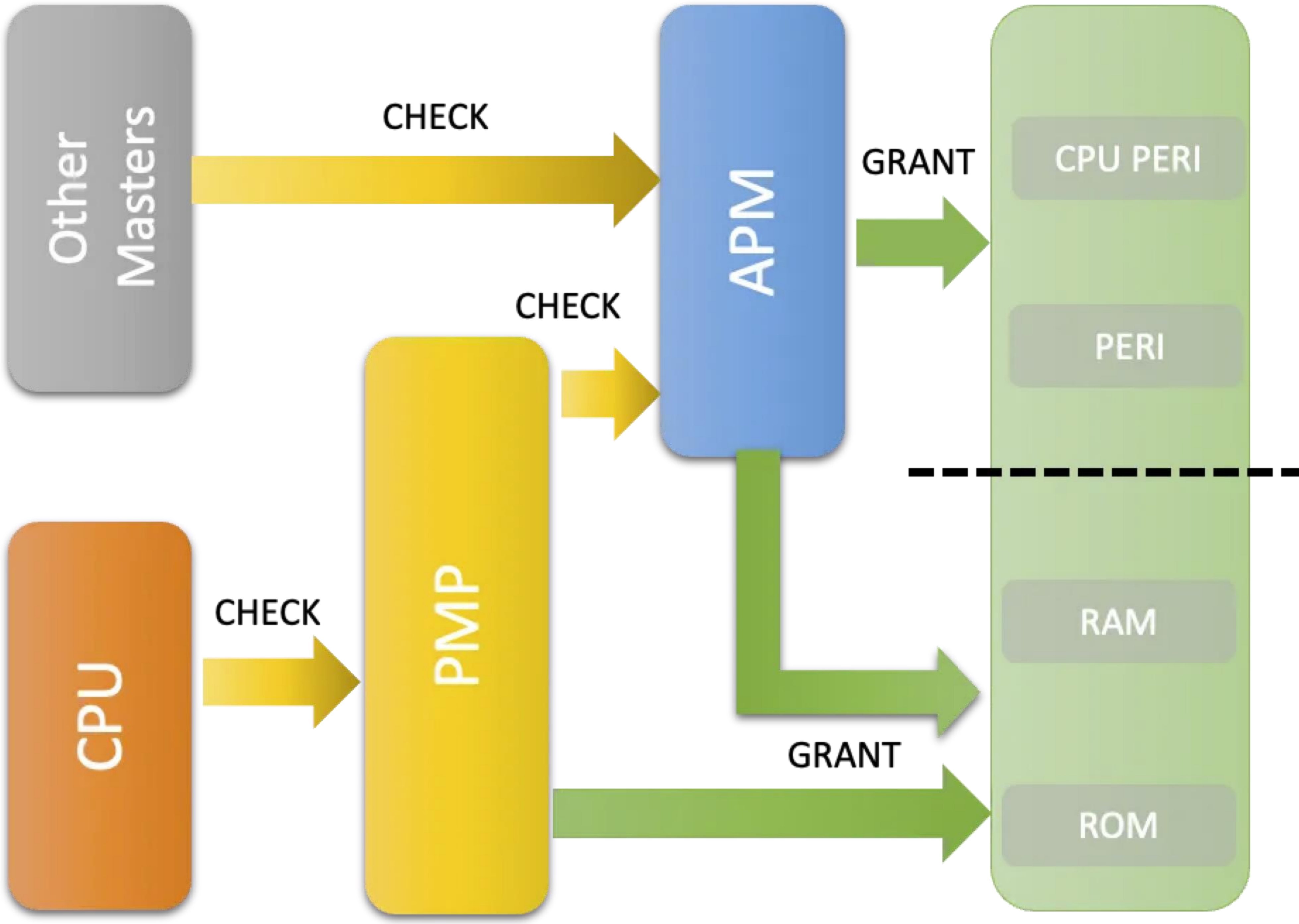
Secure Boot Considerations

- ❑ RSA-3072 or ECDSA-P256 algorithms are used for secure boot
- ❑ ESP-IDF and tools allow different ways to enable secure boot – bootloader enabling secure boot or manual eFUSE programming
- ❑ ESP SoCs allow up to 3 public key signatures configured in eFUSE facilitating key revocation – Developers must ensure security of signing keys
- ❑ For boot-time sensitive devices, bootloader customizations are possible

Hardware Enforced Isolation

- ❑ Hardware facilitated access permission management facilitating Trusted Execution Environment and privilege separation

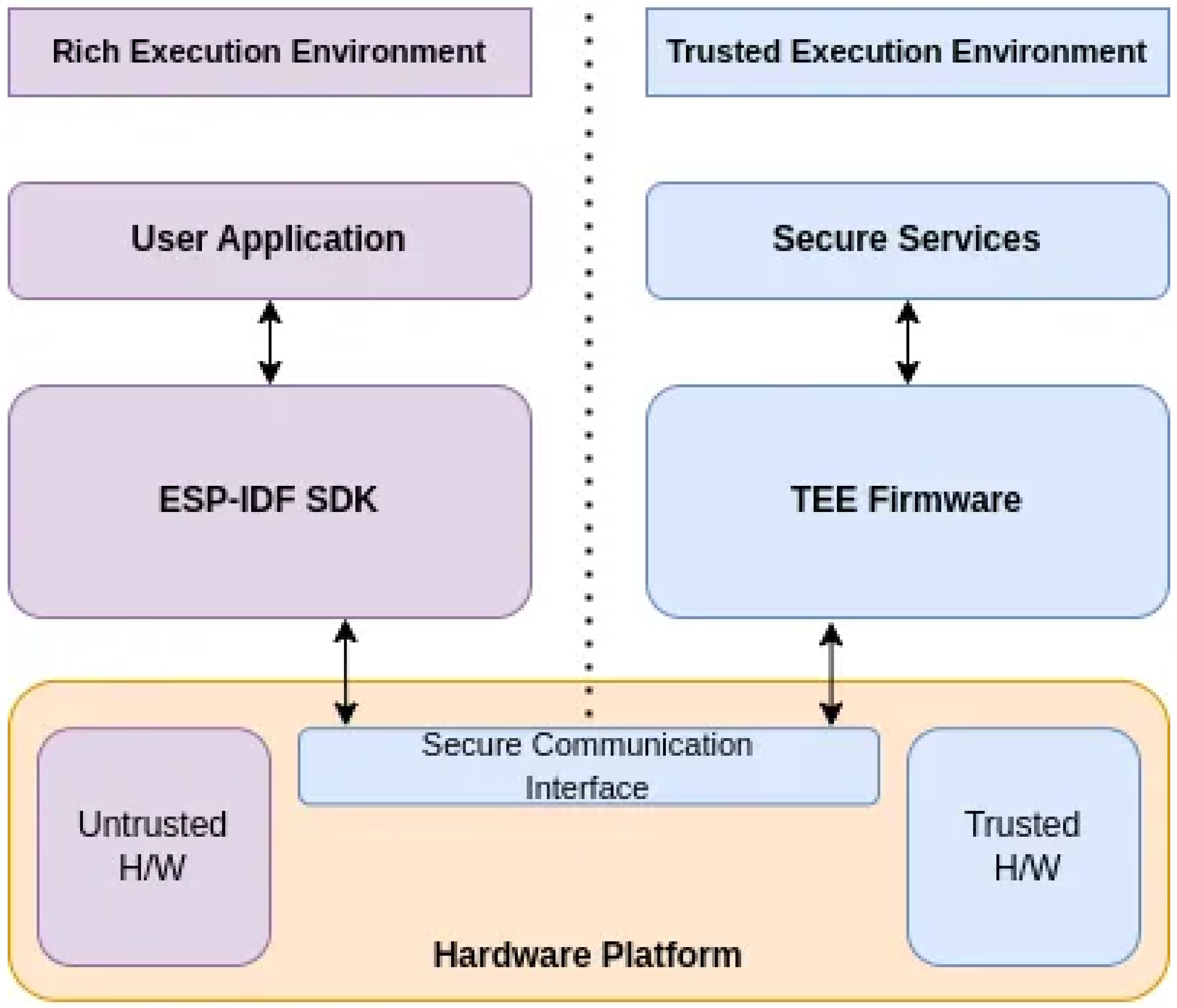
- ❑ PMP manages CPU access to SRAM and ROM, with APM handling peripheral access for CPU and other masters (e.g. DMA)



ESP-TEE

- ❑ ESP-TEE implementation offers flexible separation between trusted and rich applications running on the same chip

- ❑ Trusted application remains protected and prevents privacy of sensitive cryptographic data or device identity from possible vulnerabilities in the rich application



ESP-TEE Architecture - ESP32-C6

Other Key Hardware Settings

- ❑ Disable JTAG in production or use HMAC authenticated JTAG
- ❑ Disable UART programming interface
- ❑ Burn Security eFUSE configurations as described in documentation
- ❑ Enable Differential Power Analysis protection and AES engine’s side channel attach protection in project configuration
- ❑ Disable console logs

Secure Manufacturing

- Review the cryptographic material generated or programmed in the manufacturing line and review the access
- Ensure zero-trust (or minimal-trust) model for manufacturing to ensure security
- Espressif offers secure pre-provisioning of device certificates using flexible Certificate Signing Authority

Security Processes

❑ Always keep SBOM available for your firmware using Espressif provided tool

❑ Monitor security bugfixes in the software libraries and ensure in-field devices are kept up-to-date

❑ Monitor Espressif published advisories and PCNs

Regulatory Requirements

Common Security principles from various regulatory requirements

ESPRESSIF

Security Requirements Definition

Summary of requirements

REQUIREMENT	Definition
Authentication	Implement appropriate authentication and access control mechanisms
Configuration	Allows security-relevant configuration changes via a network or other interface, the related configuration change shall only be accepted after authentication
Cryptography	Use of cryptographic algorithms, modes and protocols, key generation and random number generation approved by a government or by an industry body in the intended deployment market
Secure Communication	The System Software shall provide the ability to ensure the authentication of connection, confidentiality and integrity of data exchanged with remote devices and servers
Hardening	Deployed (production) devices shall be protected against unauthorized use of debug or test features, with rules depending on device lifecycle state.
Logging	The device should support audit logging of security relevant events and errors. The log should include enough details to determine what happened.
Privacy	The device must ensure that any stored personal data, including that in any log files, shall only be accessible by the owner or an authorized entity.
Secure Storage	The chip shall support the secure storage or derivation of minimum set, or equivalent, of critical security parameters:
Secure Updates	The firmware, software and data that can be securely updated following manufacture.

Reference: https://www.psacertified.org/app/uploads/2024/10/PSA_Certified_Regulations_Whitepaper_Oct_2024.pdf

Regulatory Requirements Summary

REQUIREMENT	NIST 8425	PSA-L1	RED (EU)	CRA (EU)	CLS-Ready (SG)	Cyber Trust Mark (USA)	PTSI (UK)
Authentication	✅	✅	✅	✅	✅	✅	☑
Configuration	✅	✅	☑	✅	✅	✅	☑
Cryptography	✅	✅	☑	✅	✅	✅	☑
Secure Communication	✅	✅	✅	✅	✅	✅	☑
Hardening	✅	✅	✅	✅	✅	✅	☑
Logging	✅	✅	✅	✅			
Privacy	✅	✅	✅	✅	✅	✅	☑
Secure Storage	✅	✅	✅	✅	✅	✅	☑
Secure Updates	✅	✅	✅	✅	✅	✅	☑

Standards

Reference: https://www.psacertified.org/app/uploads/2024/10/PSA_Certified_Regulations_Whitepaper_Oct_2024.pdf

Mapping to Espressif MCU

Mapping of the common Security principles and the from various regulatory requirements on the various Espressif MCU

Requirement Mapping

Mapping of regulatory requirements to Espressif MCU

ESPRESSIF

REQUIREMENT	ESP32-C2	ESP32-C3	ESP32-C6	ESP32-C61	ESP32-C5	ESP32-S2	ESP32-S3	ESP32-H2	ESP32-H4	ESP32-P4
Authentication	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
Configuration	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
Cryptography	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
Secure Communication	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
Hardening		✅	✅	✅	✅	✅	✅	✅	✅	✅
Logging		☑	☑		☑		☑	☑	☑	☑
Privacy	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
Secure Storage	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
Secure Updates	✅	✅	✅	✅	✅	✅	✅	✅	✅	✅
Regulatory Requirements Met	-	ALL	ALL	PTSI, CLS, CTM	ALL	PTSI, CLS, CTM	ALL	ALL	ALL	ALL

ALL: *CRA & RED(EU), CTM (US), CLS(SG), PTSI(UK)*

RED-DA Compliance Requirements

- ❑ Decide if your product needs to follow self-assessment or notified body assessment process

- ❑ Determine the applicable EN18031 standards for your product

- ❑ For self-assessment, maintain Technical Specifications, Product Risk Assessment and most importantly self-signed Declaration of Conformity – **Espressif provides readymade templates to simplify this work**

- ❑ For notified body assessment, work with an authorized notified body (security lab) to ensure compliance for your product

Brazil’s IoT Security Compliance

❑ Brazil PNCiber defines principles for national cyber-security activity (CNCiber) with IoT as one of the focus areas

❑ It is expected to follow and promote security-by-design culture making it parallel to other compliance standards

❑ The basic feature mapping will continue to remain useful

❑ No specific timeline yet

Q&A