Secure Foundation for connected devices – Platform Security Architecture

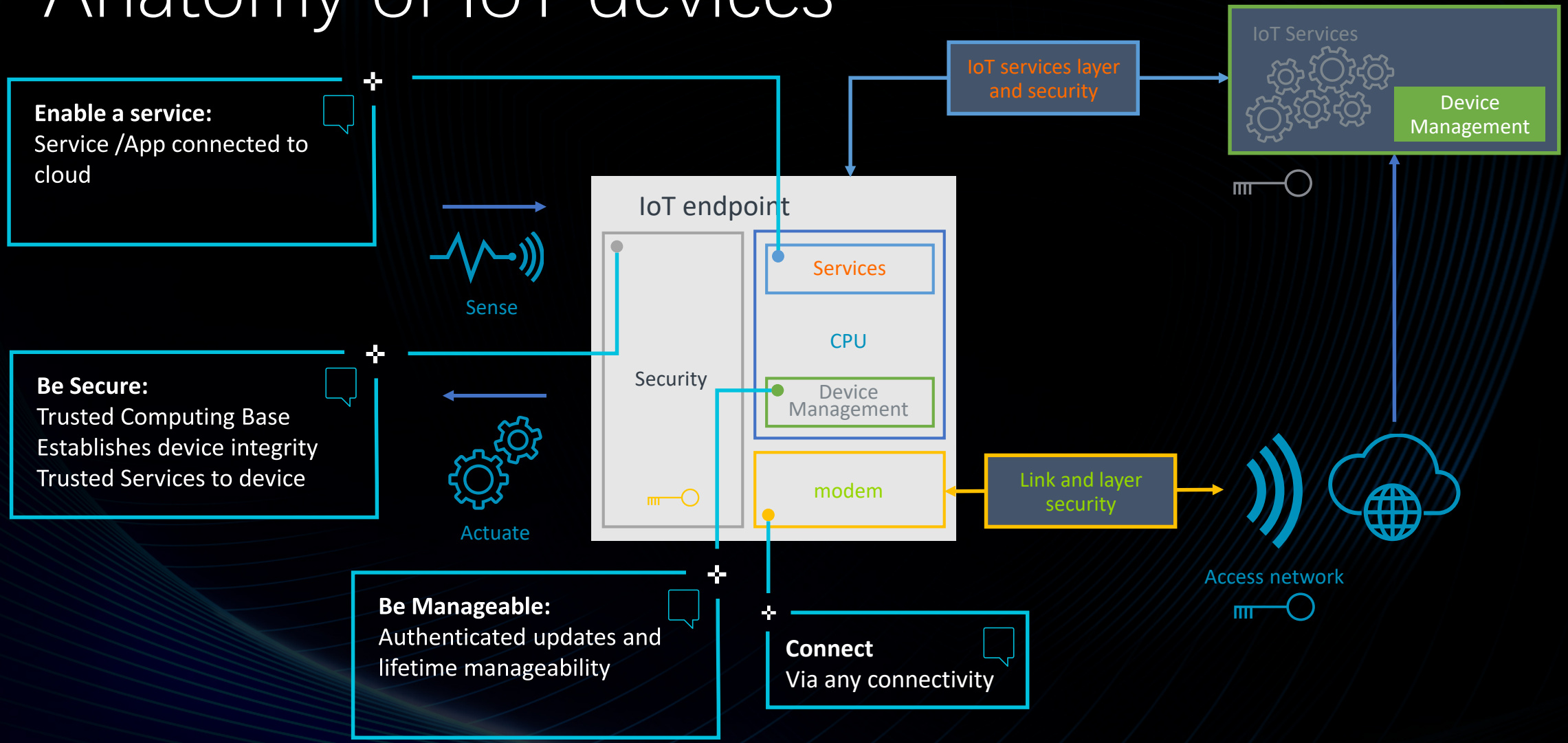
Samuel Chiang

Sn BD Director, IoT Line of Business

Agenda

- IoT – What is an IoT device
- IoT attack surfaces
- Introducing Platform Security Architecture
- Security counter-measures
- Development platforms for PSA designs

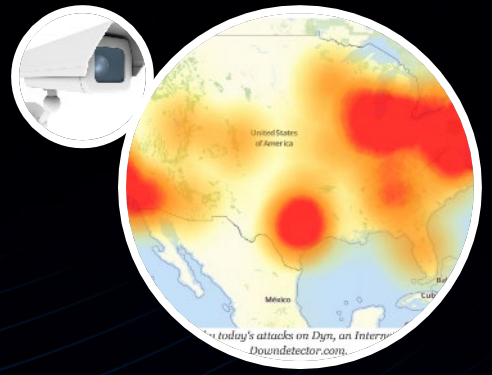
Anatomy of IoT devices



IoT – Still the wild west?

- Unregulated, no common standards
- Inconsistent approach to security
- Immature and fragmented end markets with diverse requirements

Jeep hack



Mirai Botnet DDoS attack

Owlet baby monitor

St Jude's pacemaker

What is Platform Security Architecture?

Introducing Platform Security Architecture (PSA)

- A recipe for building more secure systems from analysis to implementation

Analyse

- Threat Models and Security Analyses

Architect

- Firmware and hardware architecture specifications

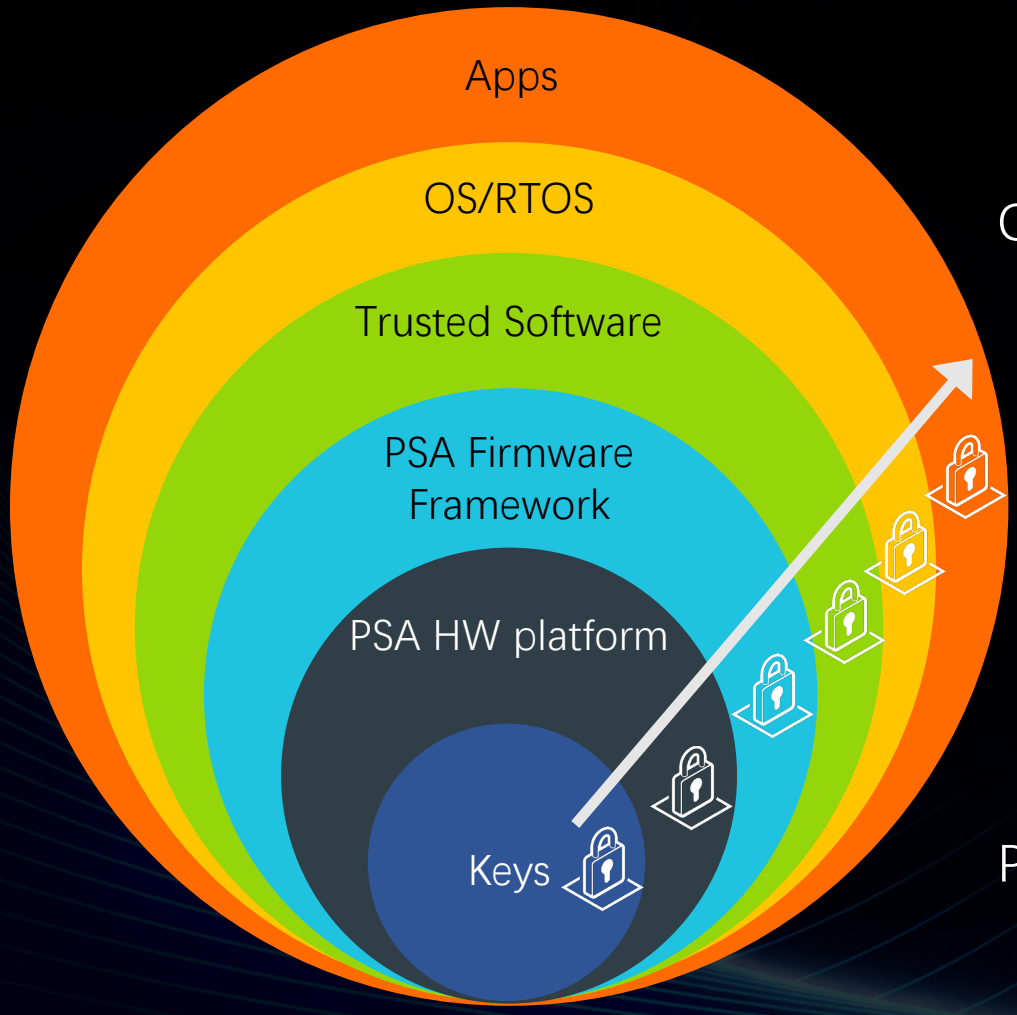
Implement

- SoC/MCU + Open Source FW

PSA documents

Arm IP and enablers

The PSA onion model of security by isolation



OS/RTOS

Trusted Apps/Libs

Extended Root of Trust

Initial Root of Trust: Dependable Security functions

Provisioned keys/certs

PSA consists of three parts

Platform Security Architecture

Device Security Model

- Hardware system architecture
- Trusted device initialisation
- Trusted boot and firmware update
- Firmware framework
- Trusted functions

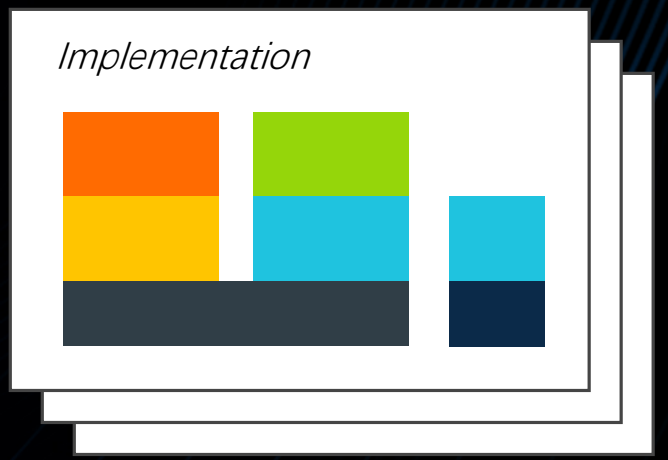
HW & FW architecture specifications

Inform

Specify

Threat Models and Security Analyses (TMSAs)

Assess



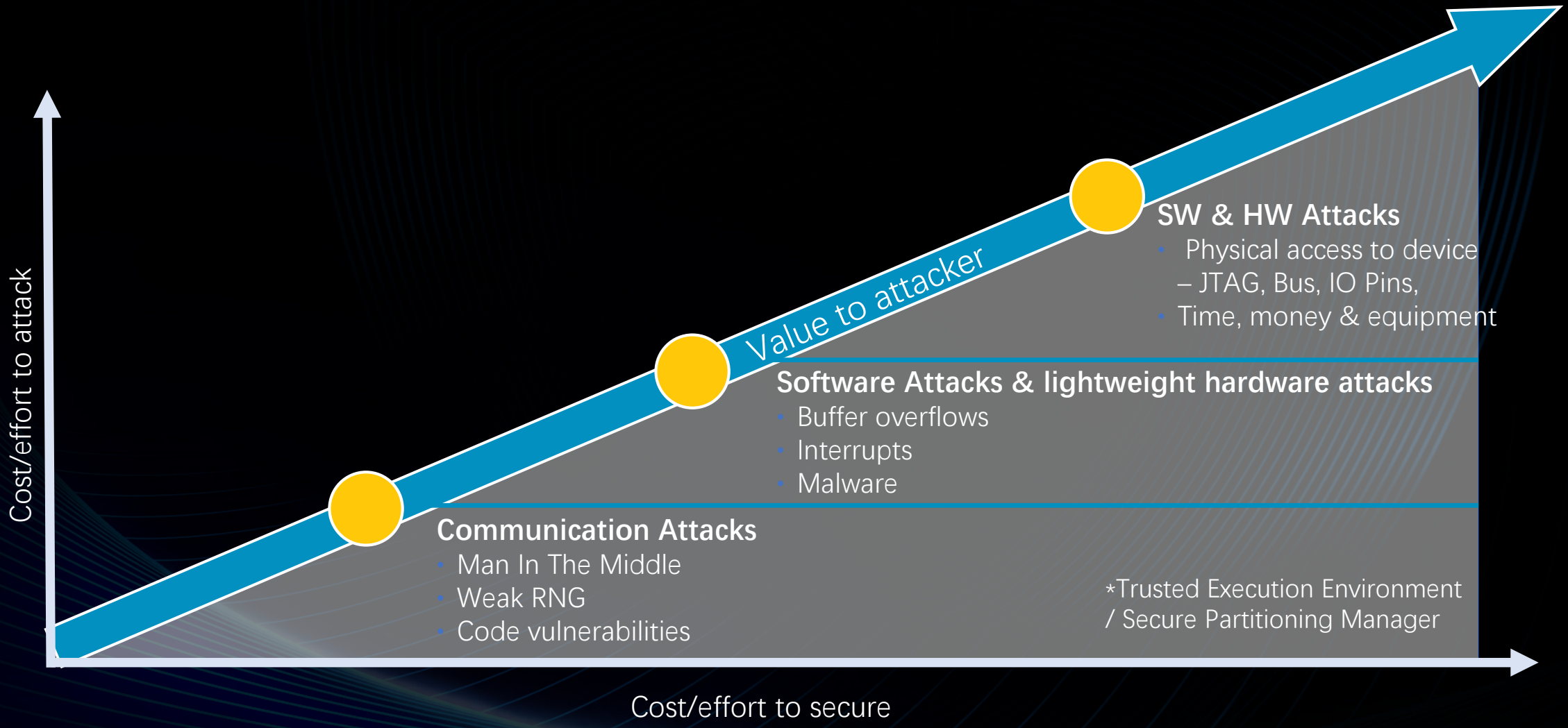
OSS FW implementation
(Trusted Firmware – M)

Threat Modelling

Analyze: Analysis of security threats and risks

- *Understand application security requirements*
  - What do I need to protect against ?
  - What are the attack vectors?
  - What are the security objectives?
  - What are the security requirements?
  - What solutions can provide the requirements?
  - **Security designed in from the start**

Security is a balance…



Threat modelling considers “How much security

PSA Analysis Stage

Assess the potential vulnerabilities

Physical
(including non-invasive and invasive)

Software
(Including buffer overflows, interrupts and malware)

Communication
(including man-in-the-middle, weak RNG and code vulnerabilities)

Lifecycle
(including code downgrade, ownership changes and unauthorized overproduction)

Physical mitigation

SecurCore, Cortex-M35P, CryptoCell-312P, CryptoIsland-300P

Software mitigation

Arm TrustZone and Arm processors with TrustZone support

Communication mitigation

Arm CryptoCell & CryptoIsland, Arm Mbed IoT Device Platform

Lifecycle mitigation

Arm CryptoCell & CryptoIsland, Arm Mbed IoT Device Platform

STRIDE threat model

Spoofing identity

- illegally accessing and then using another user's authentication information

Tampering with data

- malicious modification
- unauthorized changes

Repudiation

- deny performing an malicious action
- non-repudiation refers to the ability of a system to counter repudiation threats

Elevation of privilege

- unprivileged user gains privileged access to compromise the system
- effectively penetrated and become part of the trusted system

Denial of service

- deny service to valid users
- threats to system availability

Information disclosure

- exposure of information to individuals not supposed to access

Analysis: Establishing the “right” level of security

- www.arm.com/psa-resources

Three example Threat Models and Security Analyses (TMSA) documents available now

	Asset tracker	Smart water meter	Network camera
Use case	Long range asset-tracking device used for tracking people, vehicles, containers or valuables	Smart water meters used in domestic and business locations	Network-connected cameras found in homes and offices. Can be used for high security use cases
Security considerations	Cellular communication, SIM-based network authentication, real-time tracking	Battery-powered, limited over-the-air maintenance, long lifecycles, large deployments	Provisions to automatically connect to network

Security starts with analysis

Analysis leads to requirements

System description

↓

Assets

↓

Threats

↓

Security objectives

↓

Security requirements

↓

Arm will deliver representative IoT device security analyses & requirements

- **Example**

Asset: metering data to be protected in integrity & confidentiality

↓

Threat: Remote SW attacks

↓

Security objective: Strong Crypto

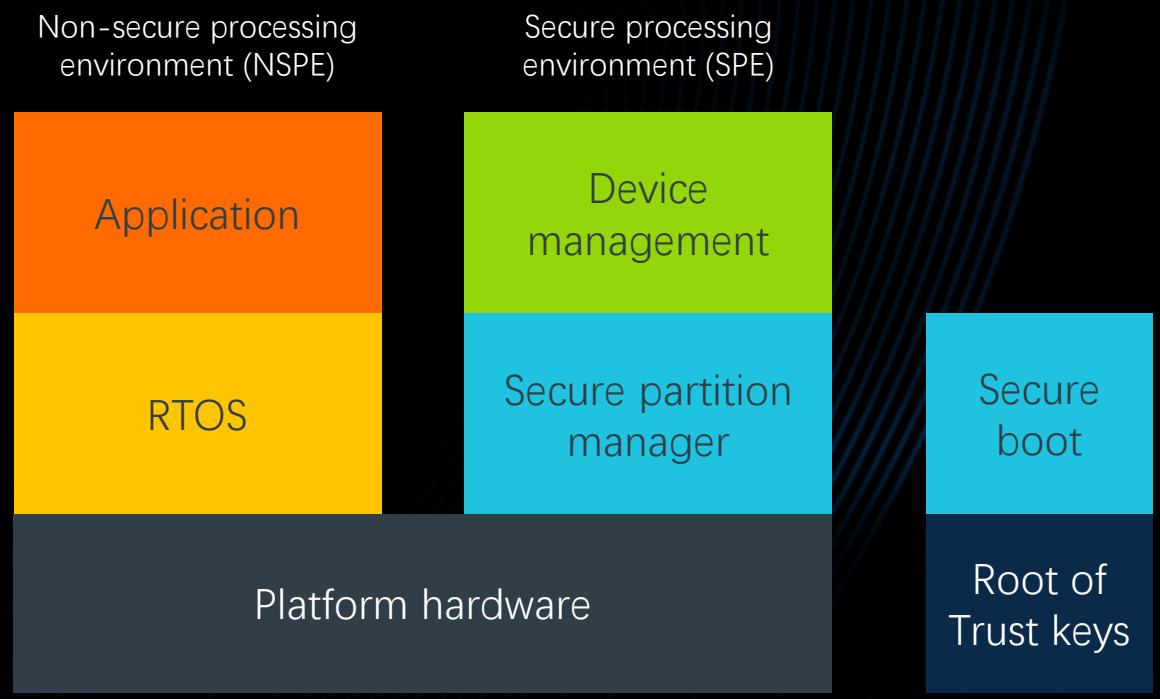
↓

Security requirement: Hardware based key store

Mitigation methods

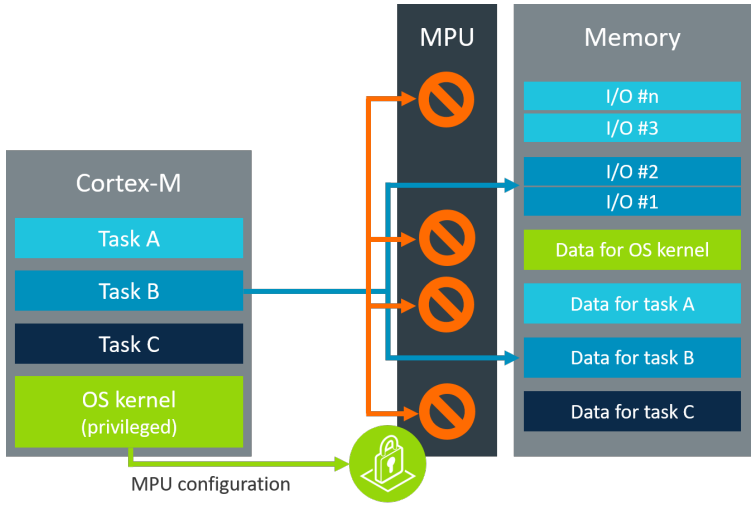
Security by separation / isolation

- PSA protects sensitive assets (keys, credentials and firmware) by separating these from the application firmware and hardware
- PSA defines a Secure Processing Environment (SPE) for this data, the code that manages it and its trusted hardware resources
- The application firmware runs in the Non-secure Processing Environment (NSPE)
- PSA defines a secure boot process so only authentic, trusted firmware runs in the SPE
- PSA depends on secure installation of the initial keys and firmware during manufacture

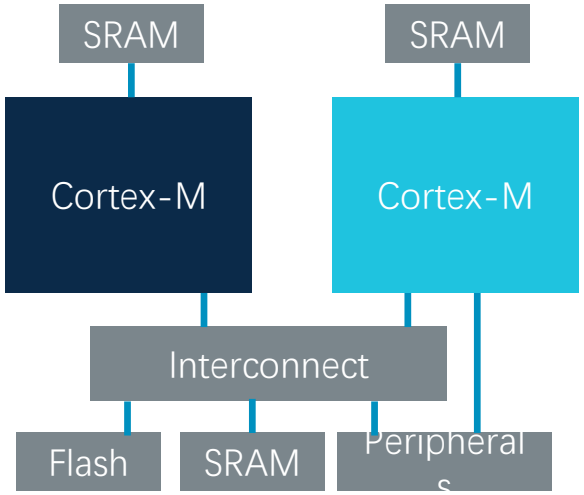


Isolation techniques

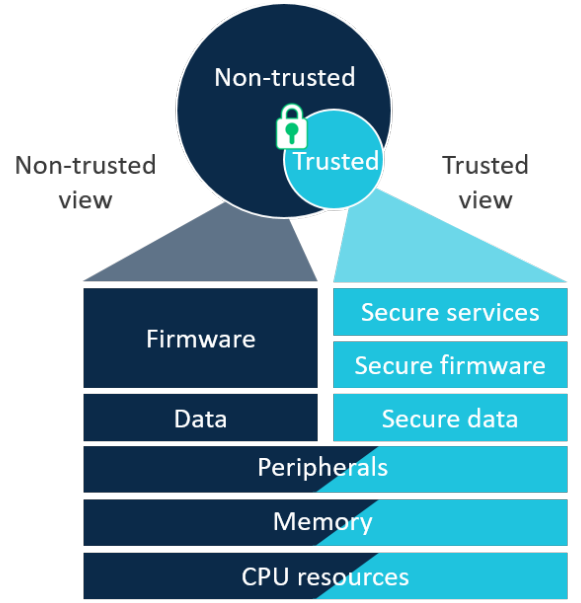
Memory protection unit



Two Cortex-M processors



TrustZone for Armv8-M



TBSA-M

TBSA-M comprises a set of requirements aimed at providing a secure hardware foundation for M-Class MCUs.

Types of requirements :

- Enforce TrustZone system principles
- Supports baseline S/W secure services
- Prohibitions on bad practices
- Propagation of good practices

Builds on TBSA-Client (for A class)

- Authored Specification
- Requirements
- Recommendations

Draft

- ☑ System view
- ☑ Infrastructure
- ☑ Fuses
- ☑ Cryptographic keys
- ☑ Trusted boot
- ☑ Trusted timers
- ☑ Version counters
- ☑ Entropy source
- ☑ Cryptographic acceleration
- ☑ Debug
- ☑ External interface peripherals
- ☑ DRAM protection
- ☑ Device lifecycle

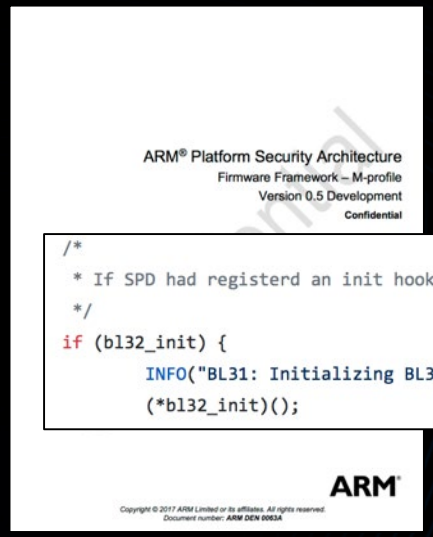
Trusted Firmware-M

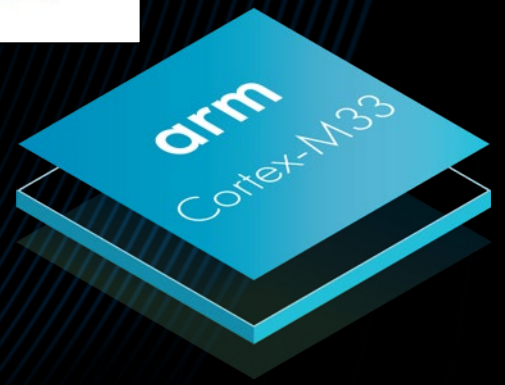
What is Trusted Firmware?

- It is a set of foundational software components which implement the services required for a secure platform:
- Reference software for partners to build on which creates a trusted execution environment
- Secure Function invocation (Software Interface to TrustZone)
- Secure Device Initialisation and Setup
- Trusted Boot (image verification derived from RoT)
- PSA Compliance

Trusted Firmware-M Open Source Project

- Trusted Firmware-M
- Reference firmware for PSA architecture specification
- Targeting M-profile SoCs (Initially Armv8-M)
- Available on www.trustedfirmware.org



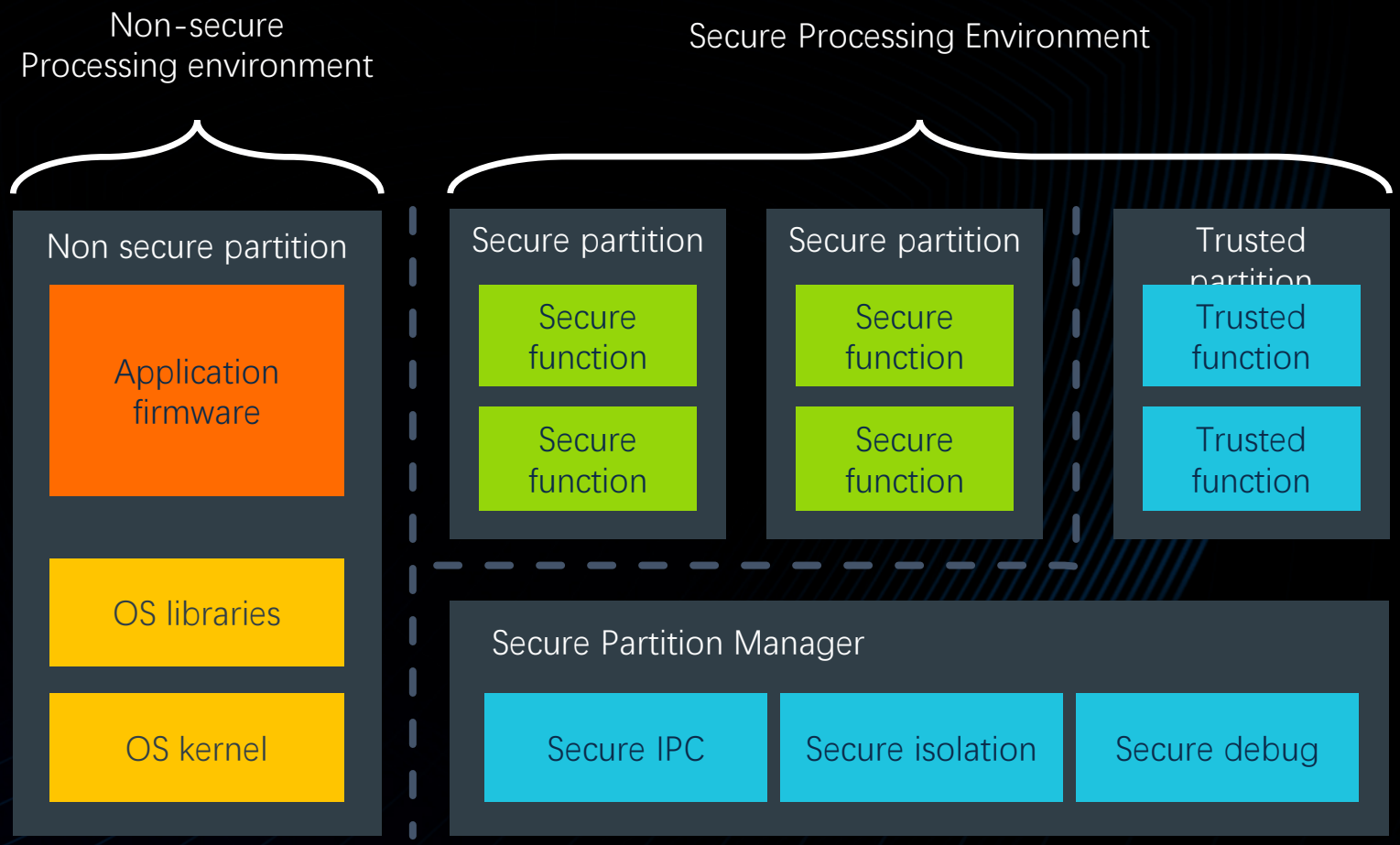


- Arm Mbed OS will include an implementation of PSA
- Based on TF-M for secure services
- Used by Mbed TLS, Pelion Device Mgmt & Mbed OS
- Components being introduced now to future Mbed releases

https://connect.linaro.org/resources/hkg18/hkg18-212/

PSA Firmware Framework Concepts

- Secure Partition Manager (SPM)
  - provides the boot, isolation and IPC services to the SPE
- Partition
  - the unit of execution
- Secure function
  - a set of related APIs invoked through secure IPC
- Trusted function
  - a secure function that provides a Root of Trust service

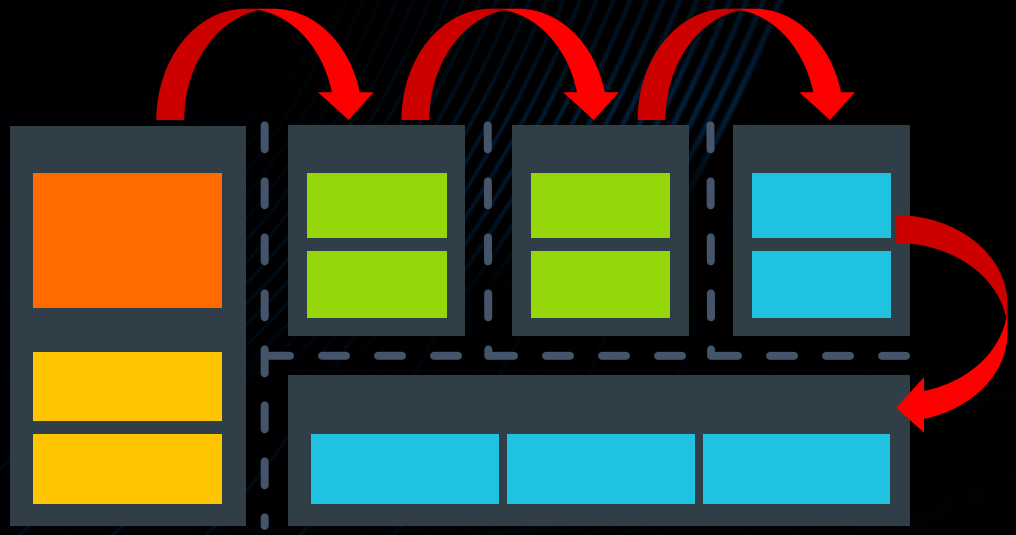


PSA firmware isolation levels

- **Level 2**
- Separate Root of Trust from Secure Partitions within

- **Level 1**
- Lower cost hardware – only isolate the

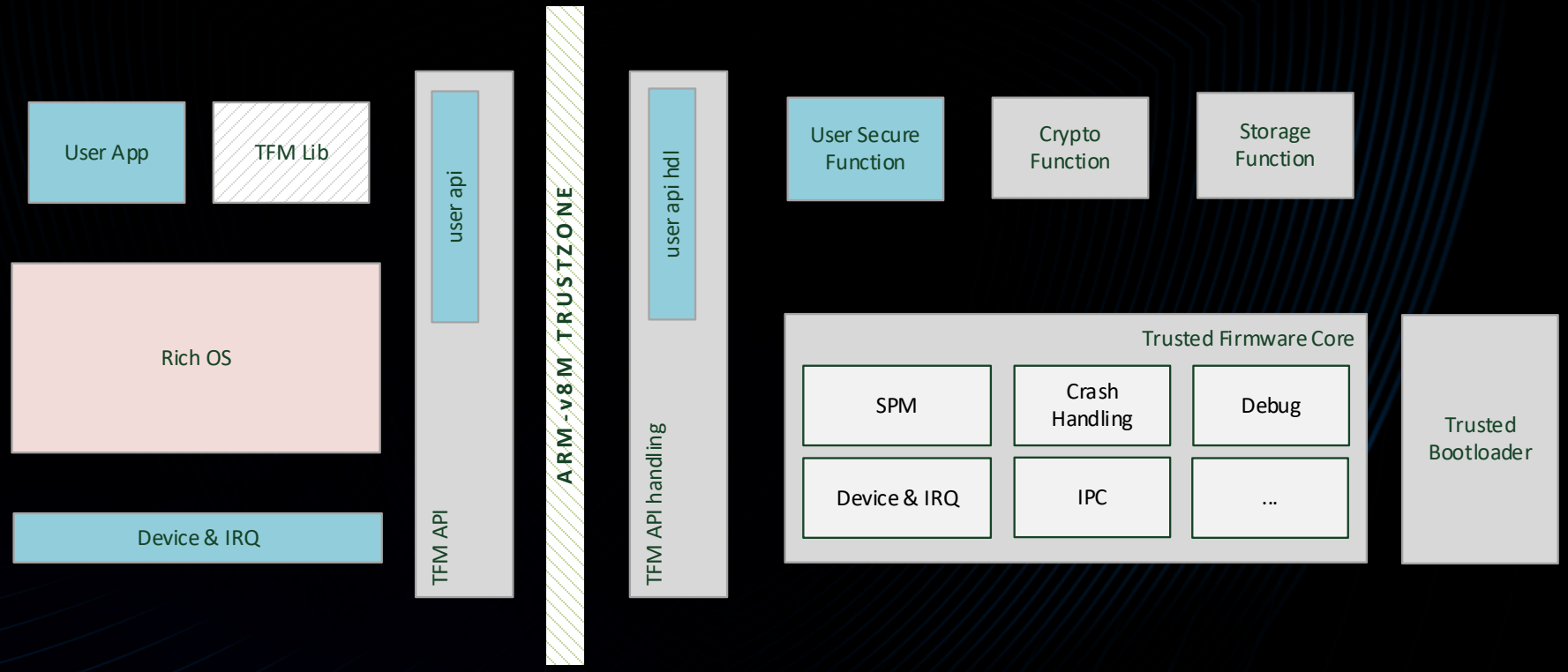
- **Level 3**
- More robustness – isolate all partitions from each other



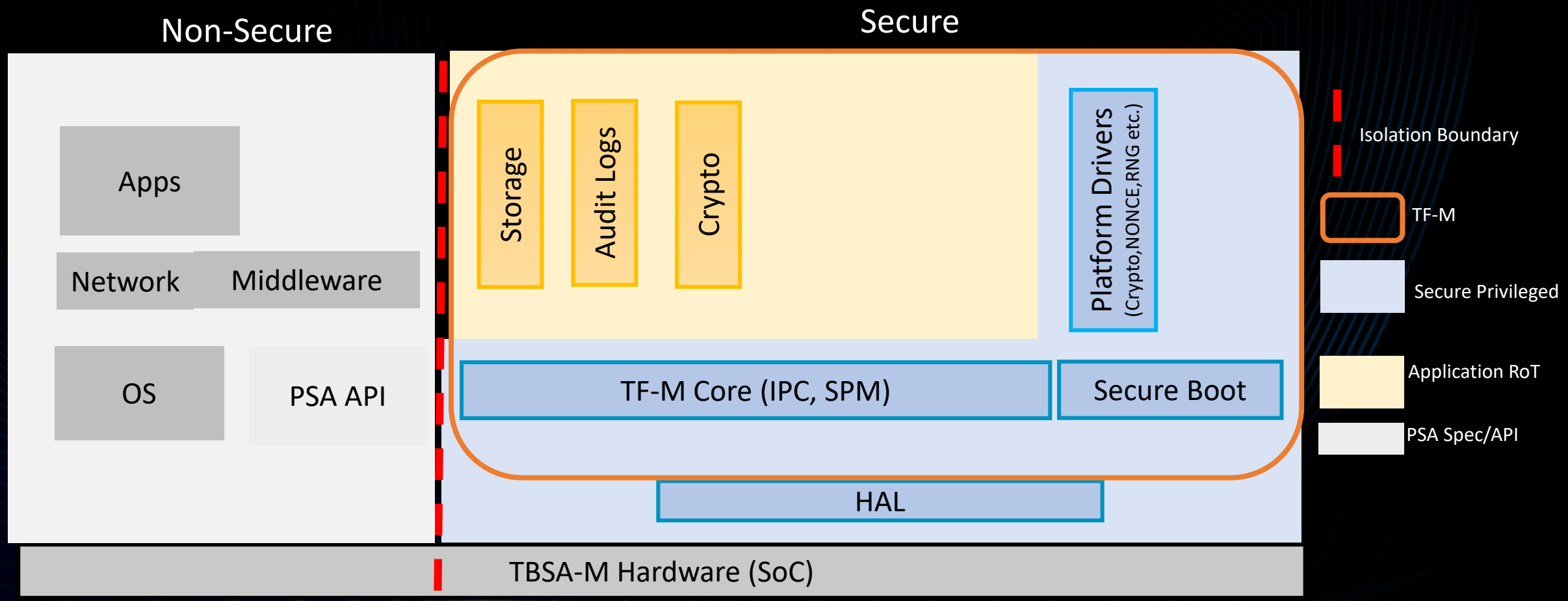
TF-M approach for ARMv8M based system

Constrained implementation with PSA alignment

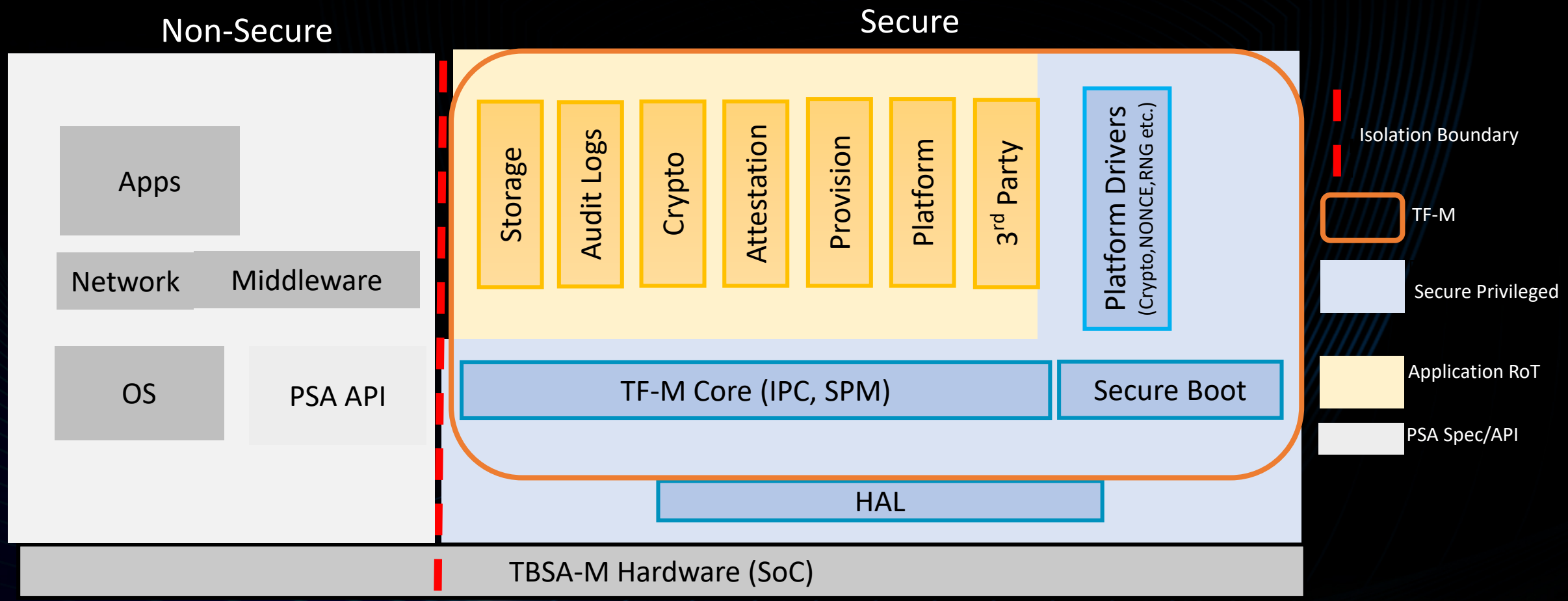
- This figure represents the current view of TF-M architecture at PSA level 1 isolation.
- TF-M project will provide mechanisms to include existing APIs in the TF-M API handlers.
- Existing apps and secure libs will most likely need some updates.
- Extent of rework will depend upon the design of existing software.



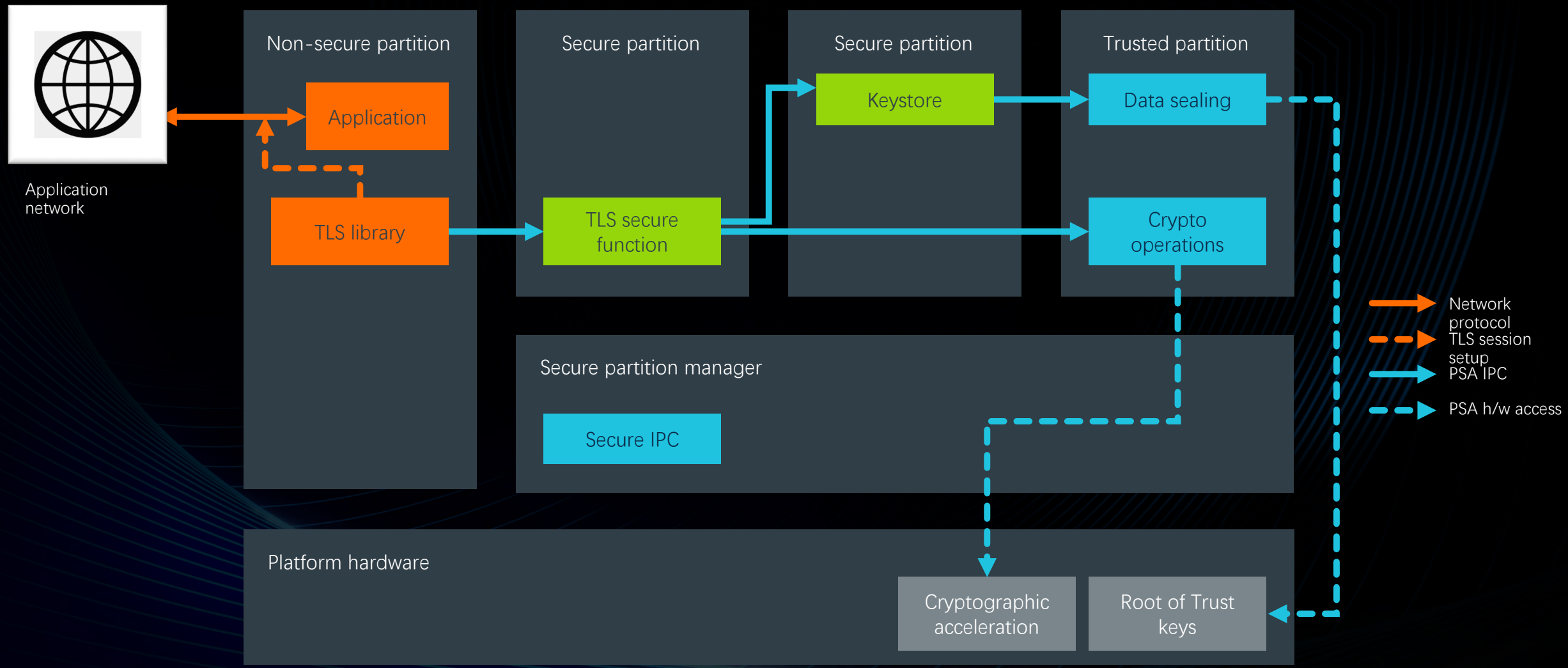
TF-M secure services today



TF-M secure services 2019



PSA In Action – TLS Session Setup



Getting started with a secure system

Get started now: Platform Security Architecture

Deliverables already available - **www.arm.com/psa-resources**

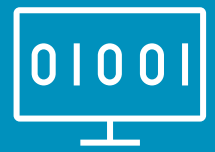
Analyze

- Download example threat models and assess your risks

Threat models
& security analyses

Architect

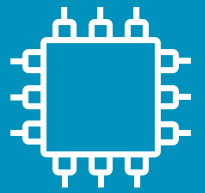
- Specifications available under NDA



Hardware & firmware
architect specifications

Implement

- Download open-source firmware (TF-M) and contribute
- Request a Musca board



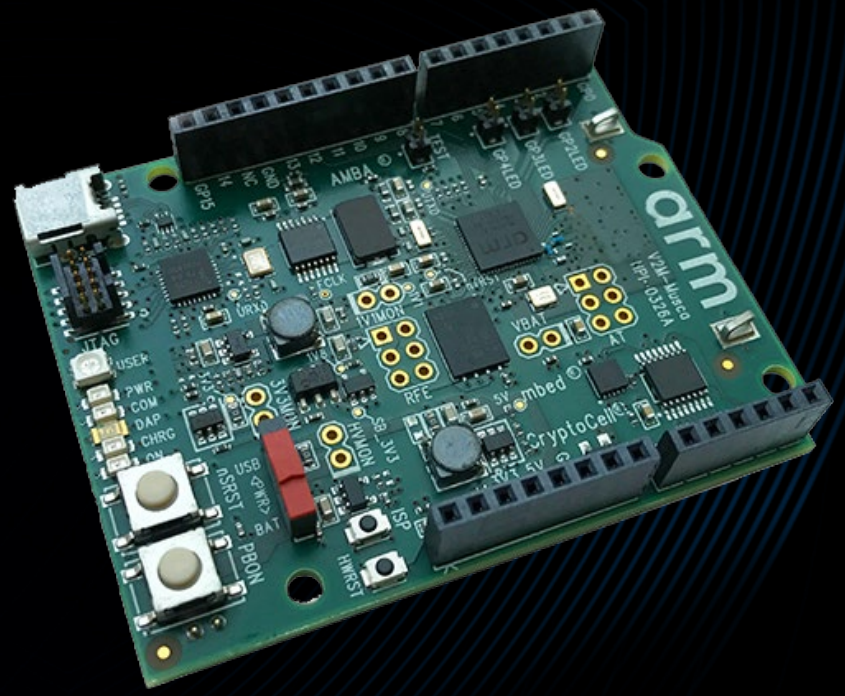
Firmware
source code

Development platforms for PSA

Musca-A1: the development board for PSA

Request your free board (arm.com/musca)

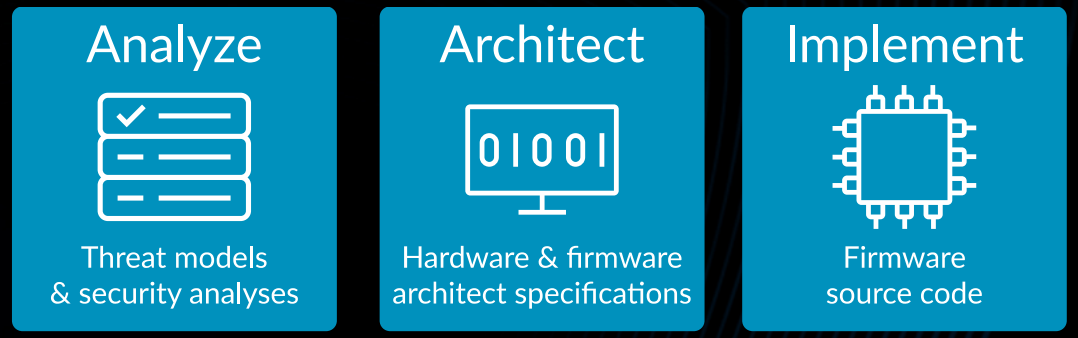
- Arm Cortex-M33 based dev board
- Used for internal software development
- Test chip built on PSA recommendations
- Prototype your system

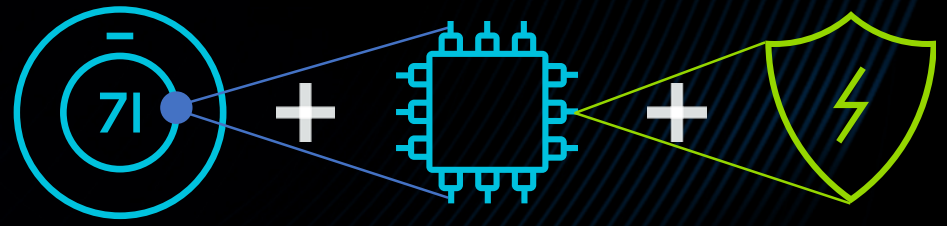


You could also choose to develop via MPS2/MPS3 FPGA boards with fixed virtual platforms

Summary – Secure foundations

- PSA provides a complete set of security deliverables reducing TTM and cost
- PSA makes security easier to implement through a common architecture
- Enabling success for the ecosystem, by providing confidence and trust





THANKS

THANKS

THANKS

THANKS