arm

Arm Research Summit 2019

The Evolution of Security Solutions for Microcontrollers

Joseph Yiu, distinguished engineer, Arm

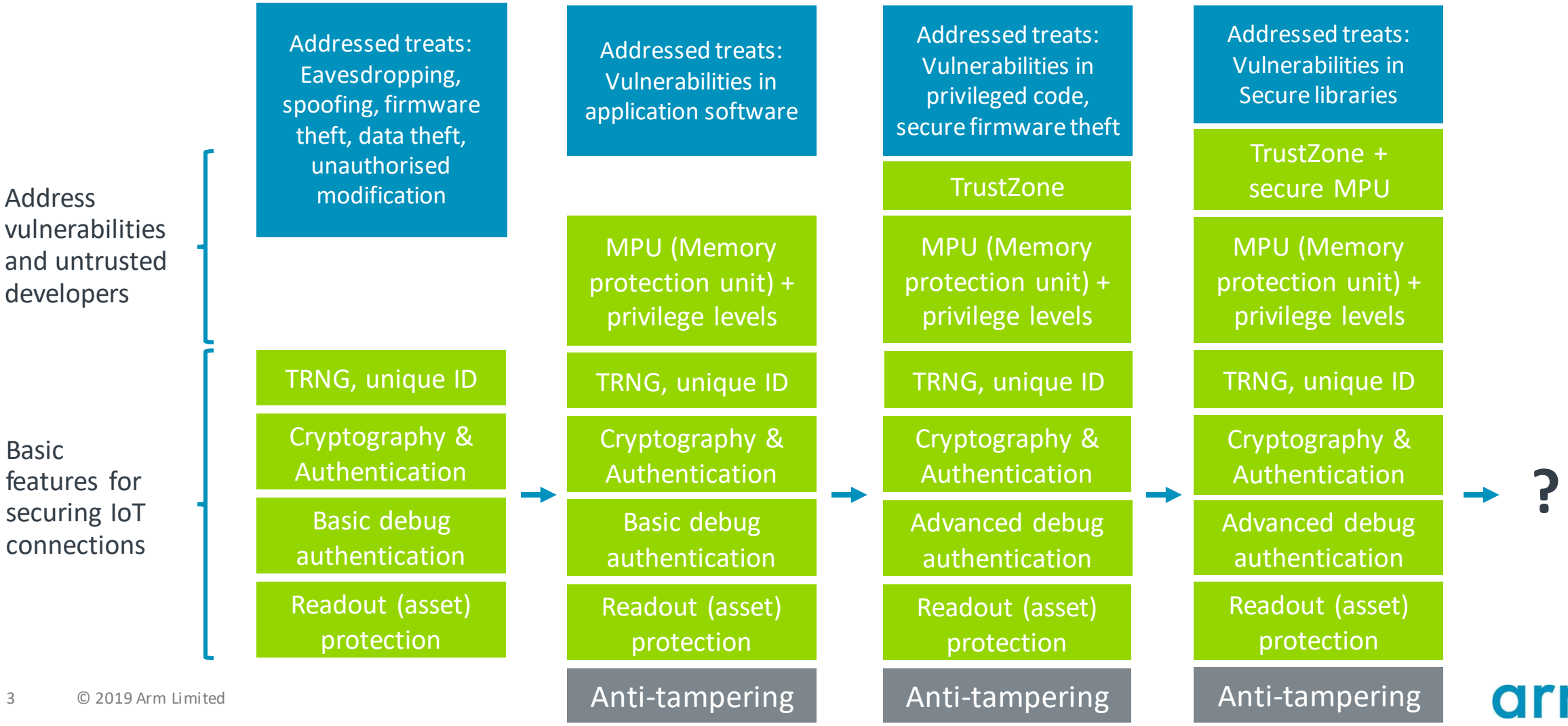
September 2019

Agenda

- The changing landscape of security requirements
- New requirements from chip vendors and the Arm ecosystem
- How Armv8.1-M security features address some of the requirements
- Secure delivery of middleware components
- Summary

Security Requirements in IoT Era Change Constantly

Security capability is now a product differentiation method

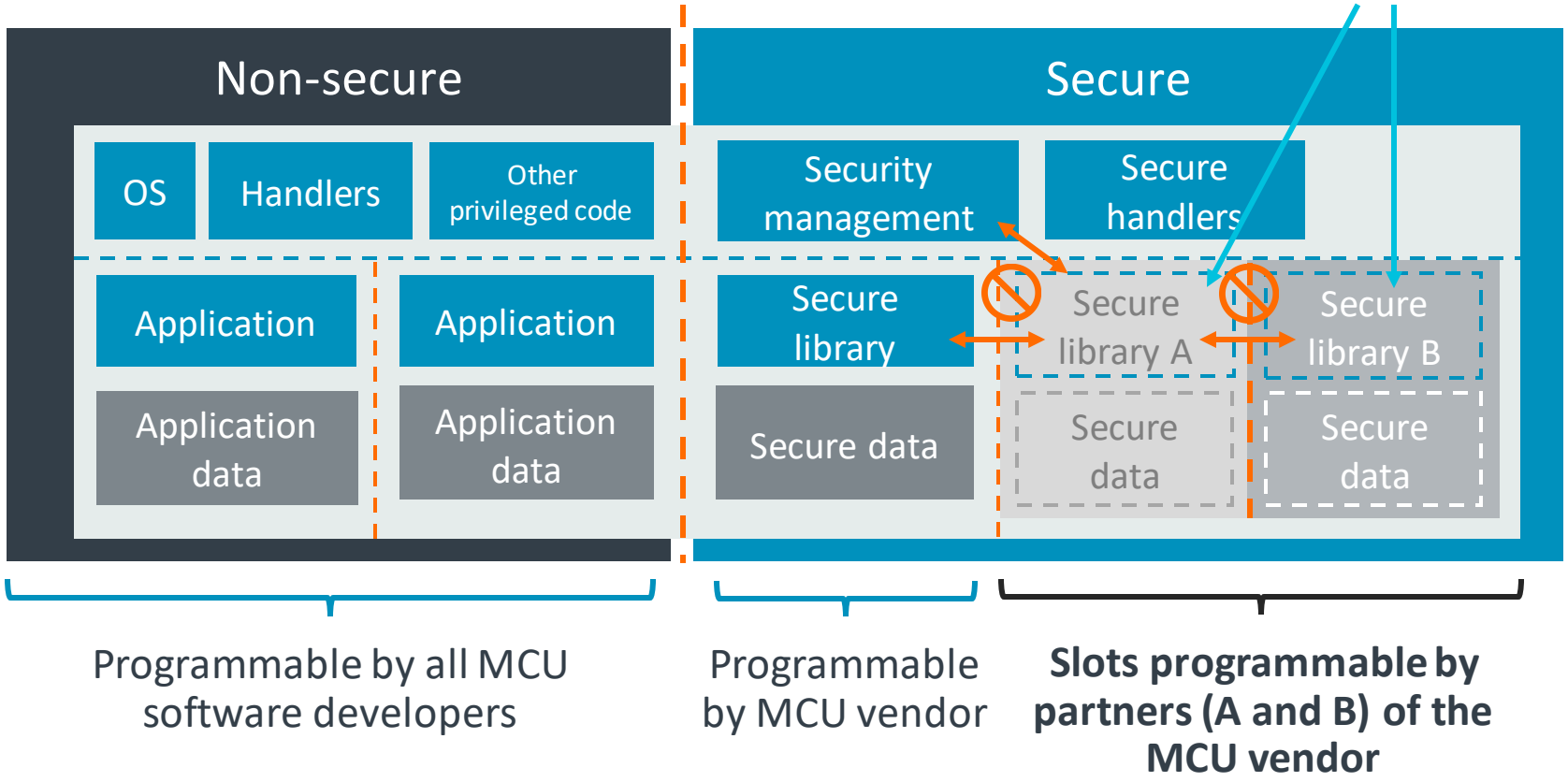


Is My Device Secure Enough?

… Remember: requirements keep changing!

What's new?

Chip vendors want to isolate **third-party software partition(s)** in Secure world **during development**



Example:

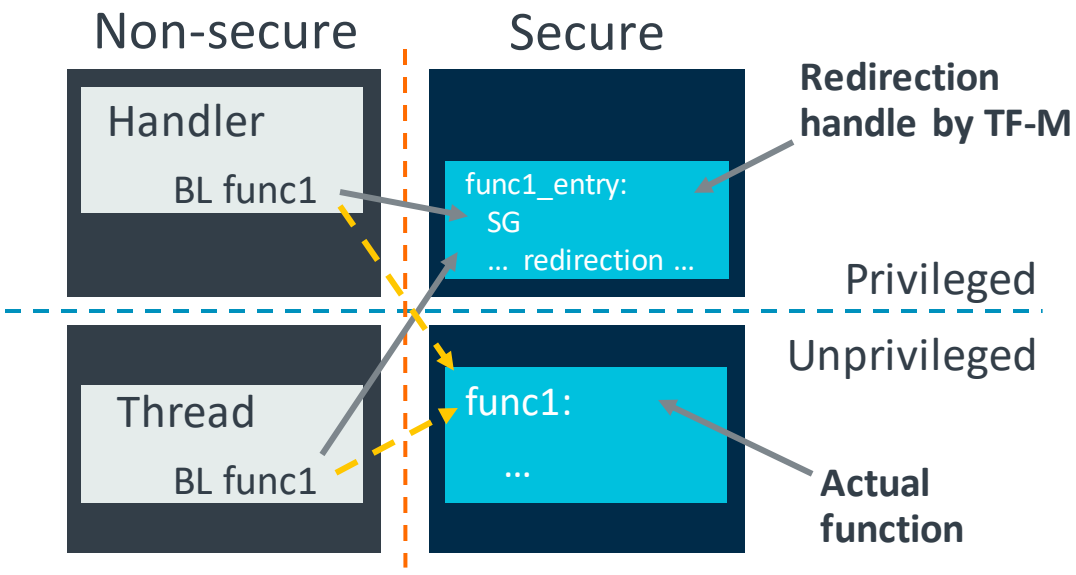
- MCU vendors do not want partners A & B, such as cloud solution providers, to see their secure codes
- Partner A and B are not mutually trusted

(A different concept from previous TrustZone usage)

arm

Additional Challenges – Third-party Secure API Entry Points

Chip vendors may say, “Untrusted Secure APIs in third-party partition must not be allowed to be executed in privileged state”… but they do need to have their own Secure entry points



Current TrustZone redirection for secure library entry-points

For Armv8-M today, the Secure world has redirection code for Secure unprivileged APIs

- Ensure the APIs execute as Unprivileged (bounded by MPU) even it is called by Non-secure Handler.

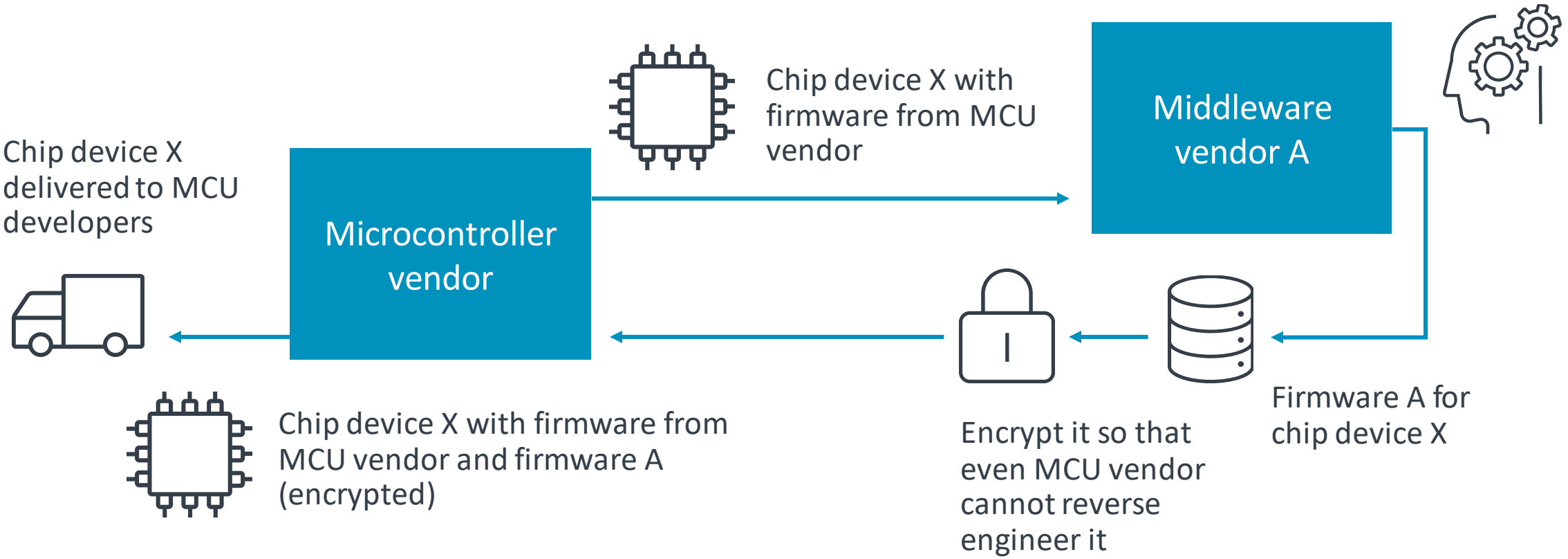
This is fine if entry points in the Secure library is available when the Secure world is compiled.

- But, if Secure unprivileged APIs are to be added later and are unknown in the development phase of the Secure firmware, as in the case of the new requirement, there is a problem.

arm

Bonus Challenge…Protecting Middleware in MCU Production

For middleware companies that work with a MCU vendor, how can they have their firmware preloaded on the MCU devices, while still protecting their software IP asset?



arm

Solutions: Security Features in Armv8.1-M and System Solutions

- PXN (Privileged eXecute Never) attribute in MPU (Memory Protection Unit)
- Unprivileged Debug Extension (UDE)

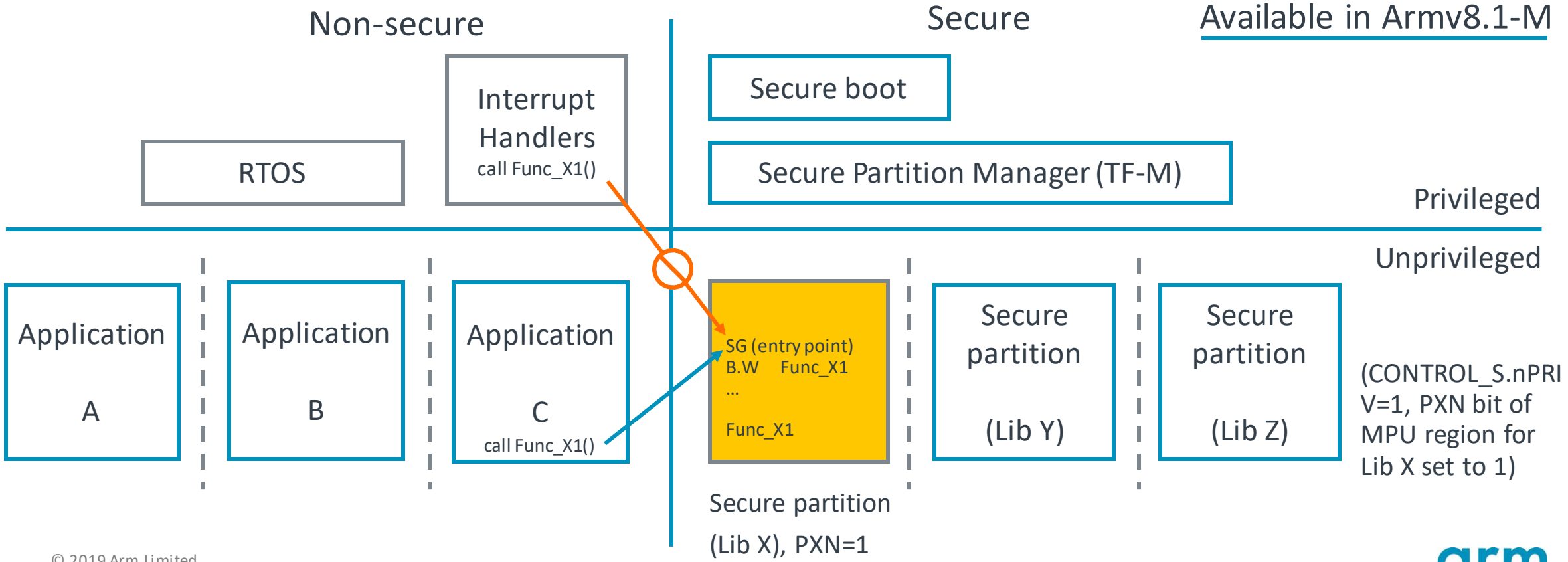
New features in Armv8.1-M architecture

- System-level arrangements for software delivery protection

arm

PXN – Privileged eXecute Never in MPU Region Attribute

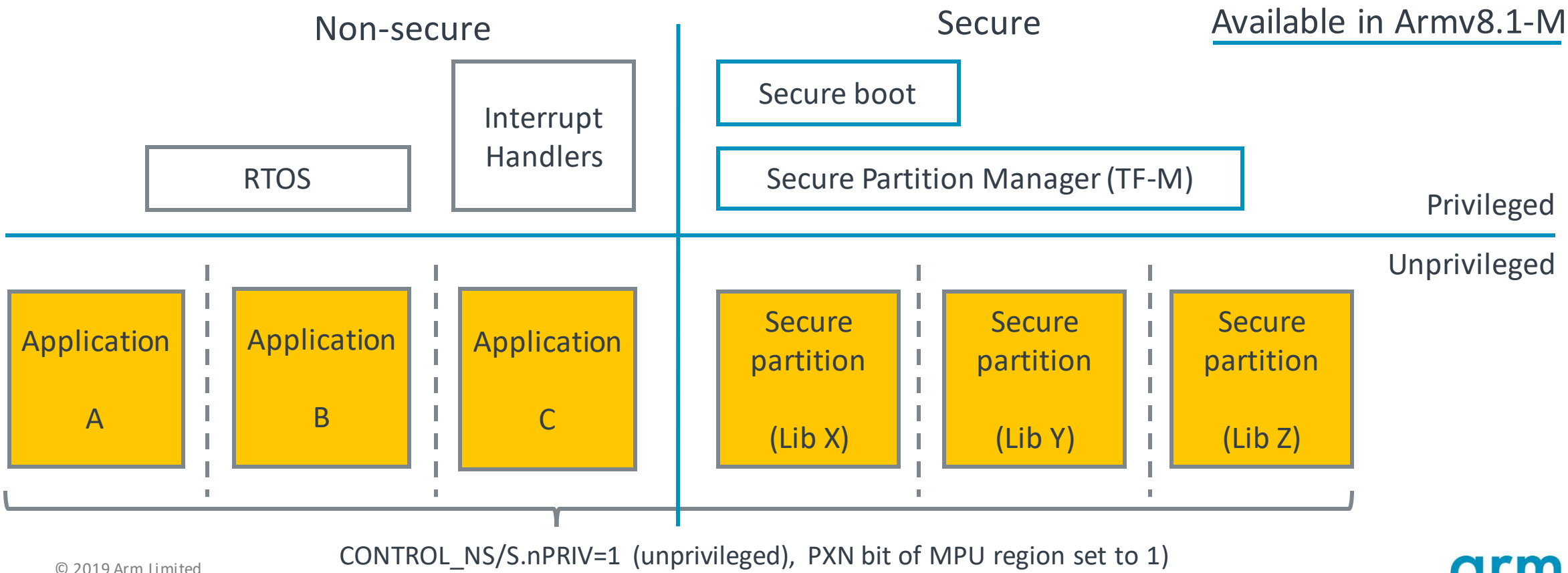
- Secure unprivileged libraries can have entry points – faster and flexible API definitions
- If the APIs are called by the Non-secure handler, then MemFault handler can intercept, switch to unprivileged state and resume.



PXN is Useful to the Non-secure World, too

An OS can mark program memories of applications as PXN to reduce the security risk

- Be careful of run-time library functions handling



Unprivileged Debug – a New "Mode" of Debugging

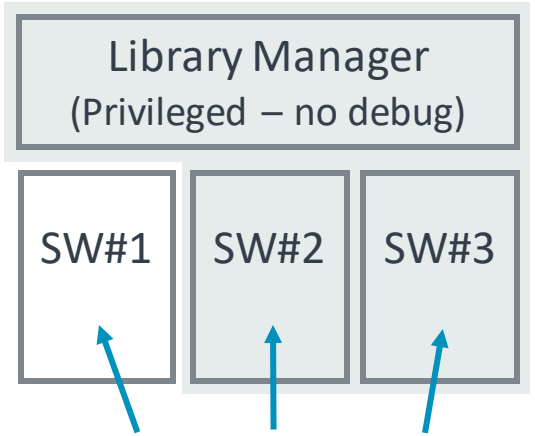
If enabled, only allow debugging of unprivileged code

- Privileged software control debug and trace visibility at context switches
  - Separated control bits for Secure and Non-secure worlds
- Debug accesses
  - Debug accesses are checked by MPU in this "mode" – in exactly the same permission view as the unprivileged application
  - Debug accesses to memories (except debug components) is not permitted when the processor is running

Existing debug operations are not affected if unprivileged debug is disabled

Debug tools and software need to be updated to support this

- New debug register fields



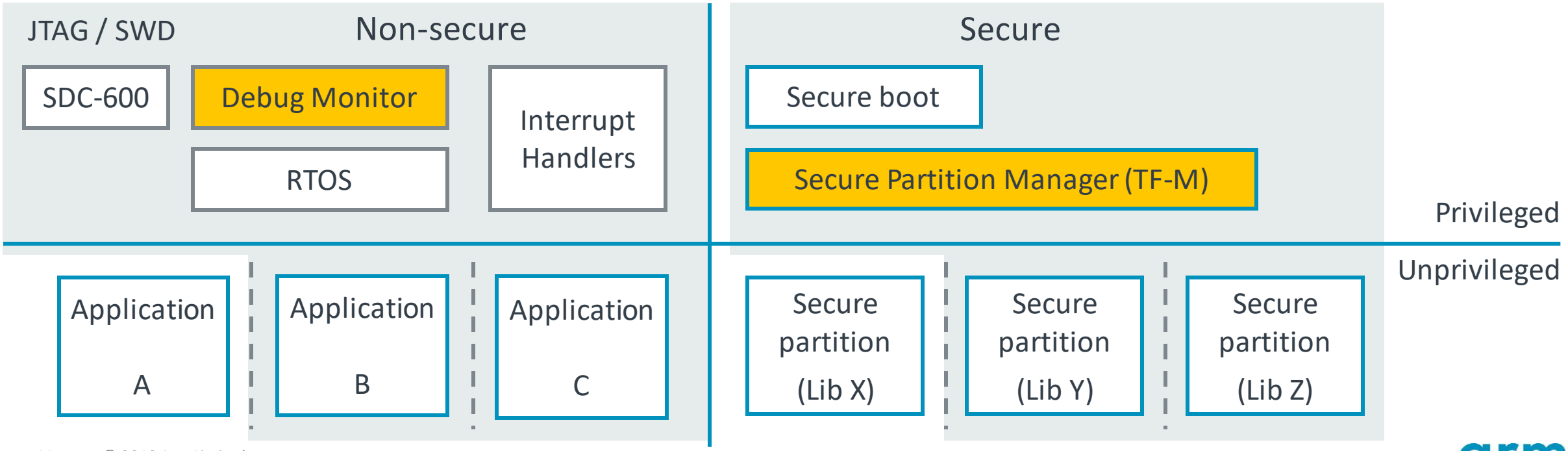
Unprivileged software threads/tasks

(programmable debug permission)

arm

Software Enhancement Opportunities with UDE

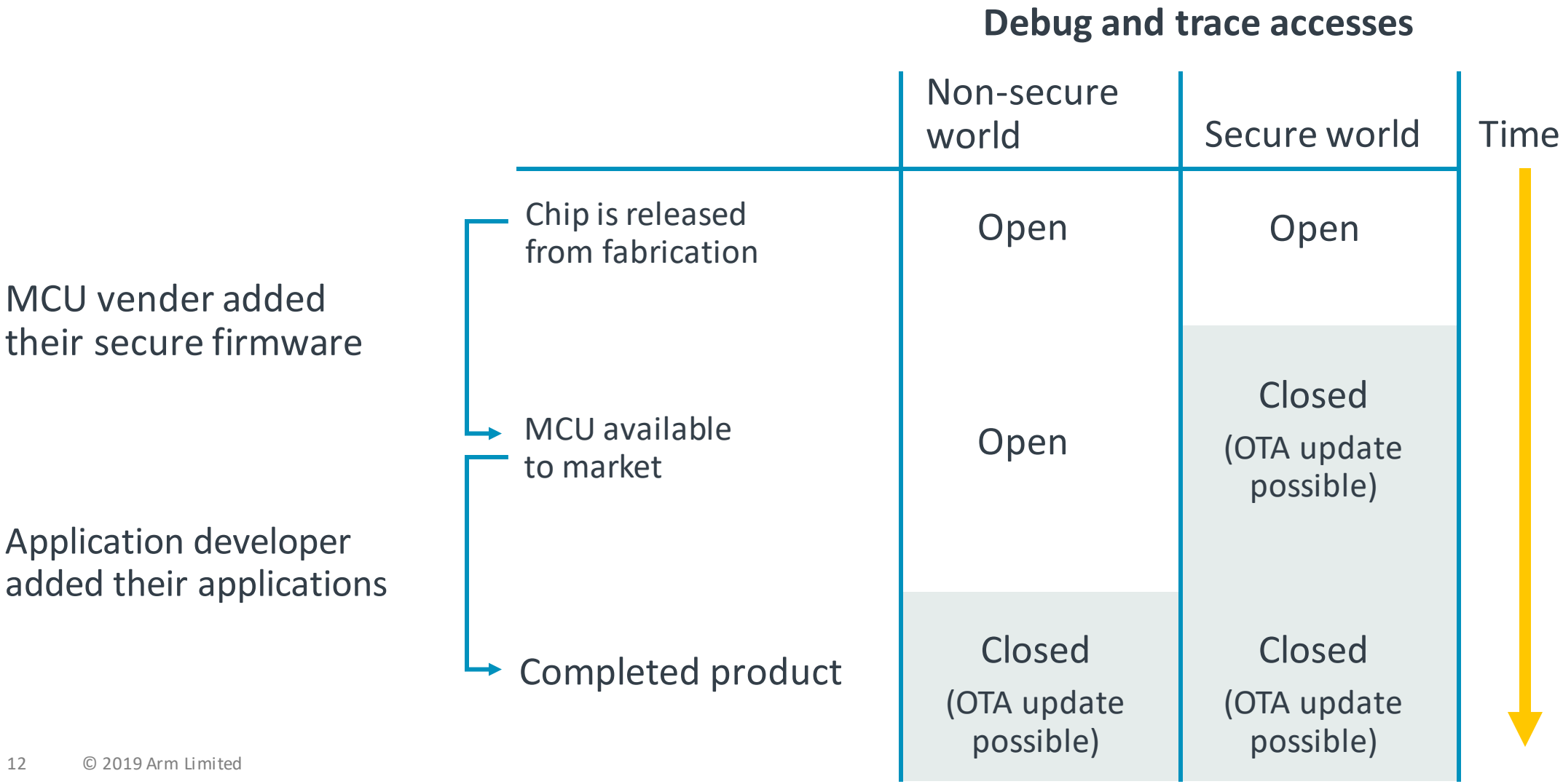
- Secure world: SPM can be enhanced with UDE to allow debugging to be restricted to the individual partition
- Non-secure world: with debug monitor support, OS and other applications can remain running and available while debugging an application
  - Permission check by software – need link to OS's permission control rather than using MPU



arm

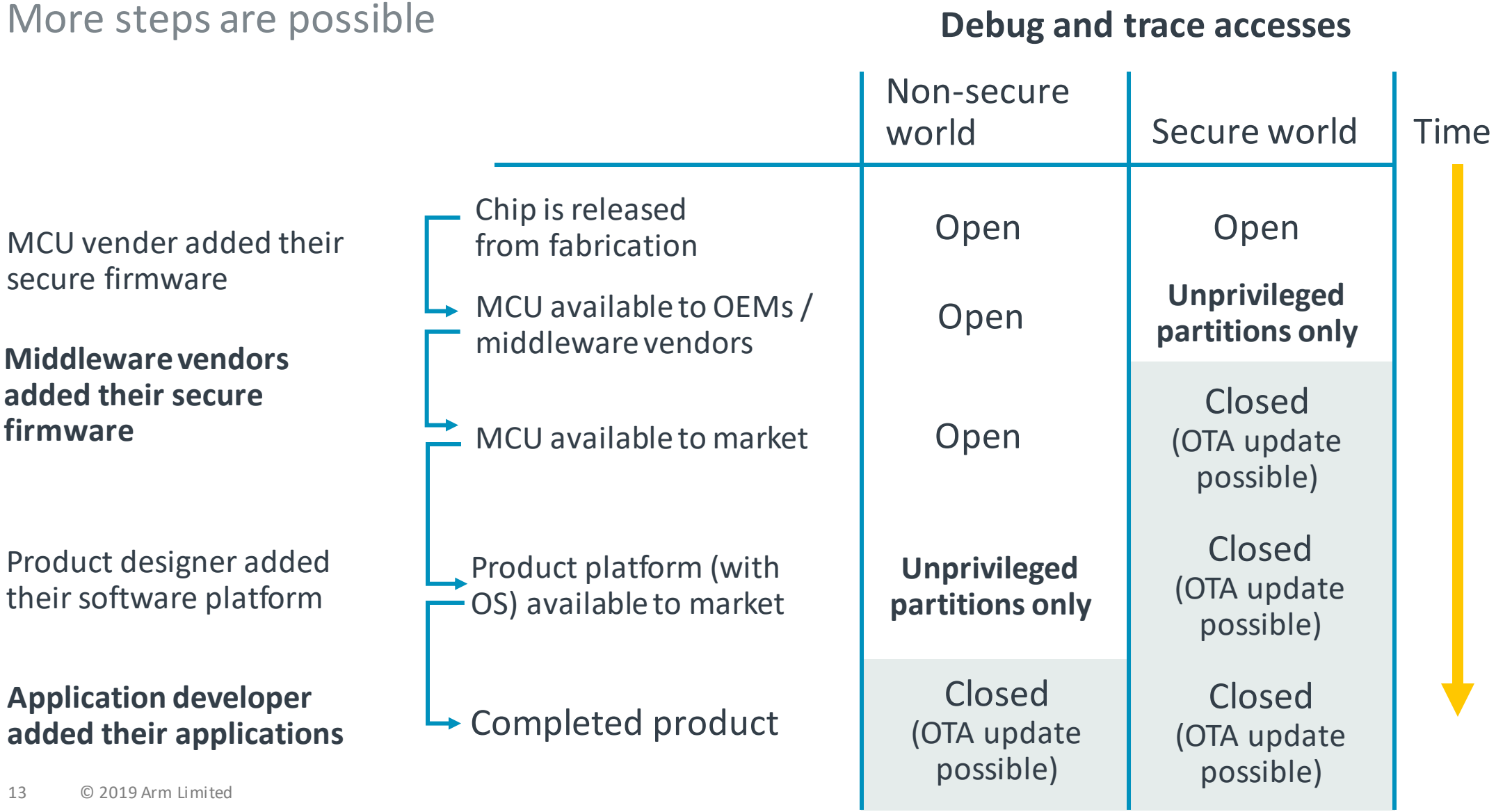
Debug Authentication for TrustZone Before UDE

Secure firmware is protected from Non-secure developers



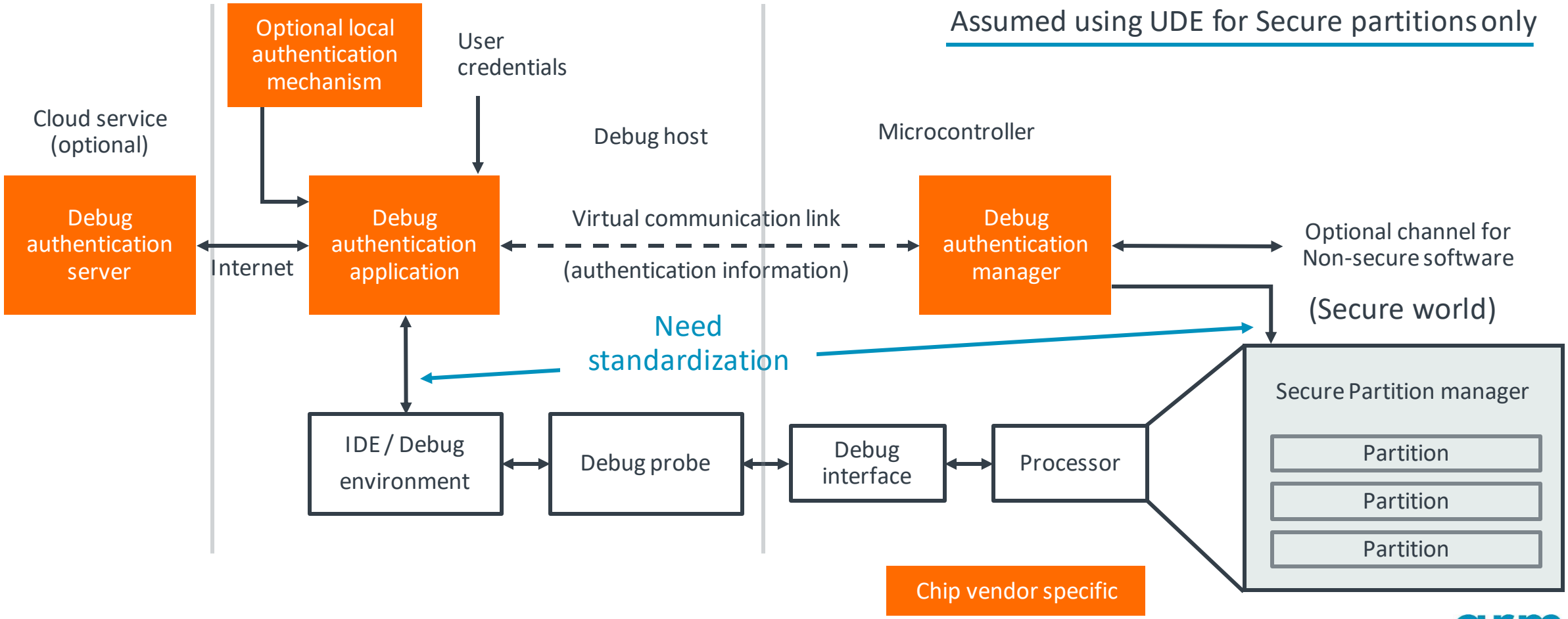
Debug Authentication for TrustZone with UDE

More steps are possible



UDE Needs a Lot of Things Working Together

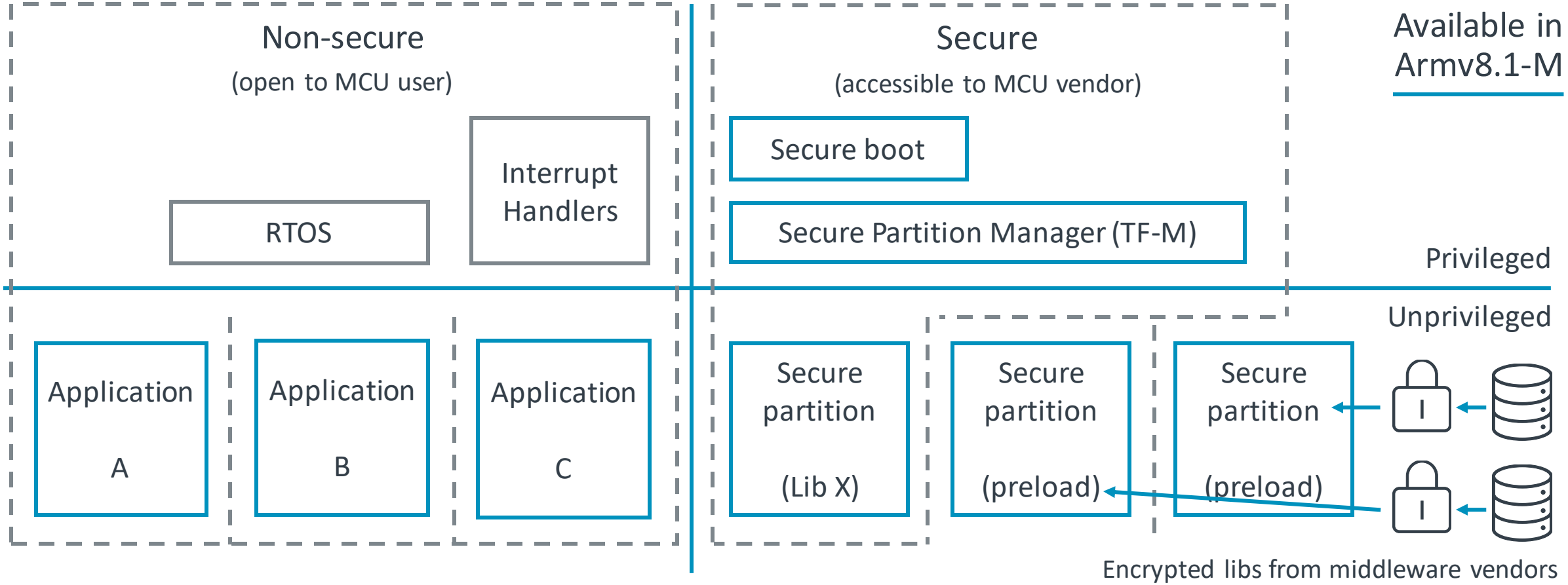
Ecosystem enablement work is ongoing



The Need for Secure Delivery of Secure Software Libraries

Middleware vendors deliver firmware libraries to MCU vendors in encrypted form

- To ensure **only** privileged parties from MCU vendors can access, as production lines may be outsourced



arm

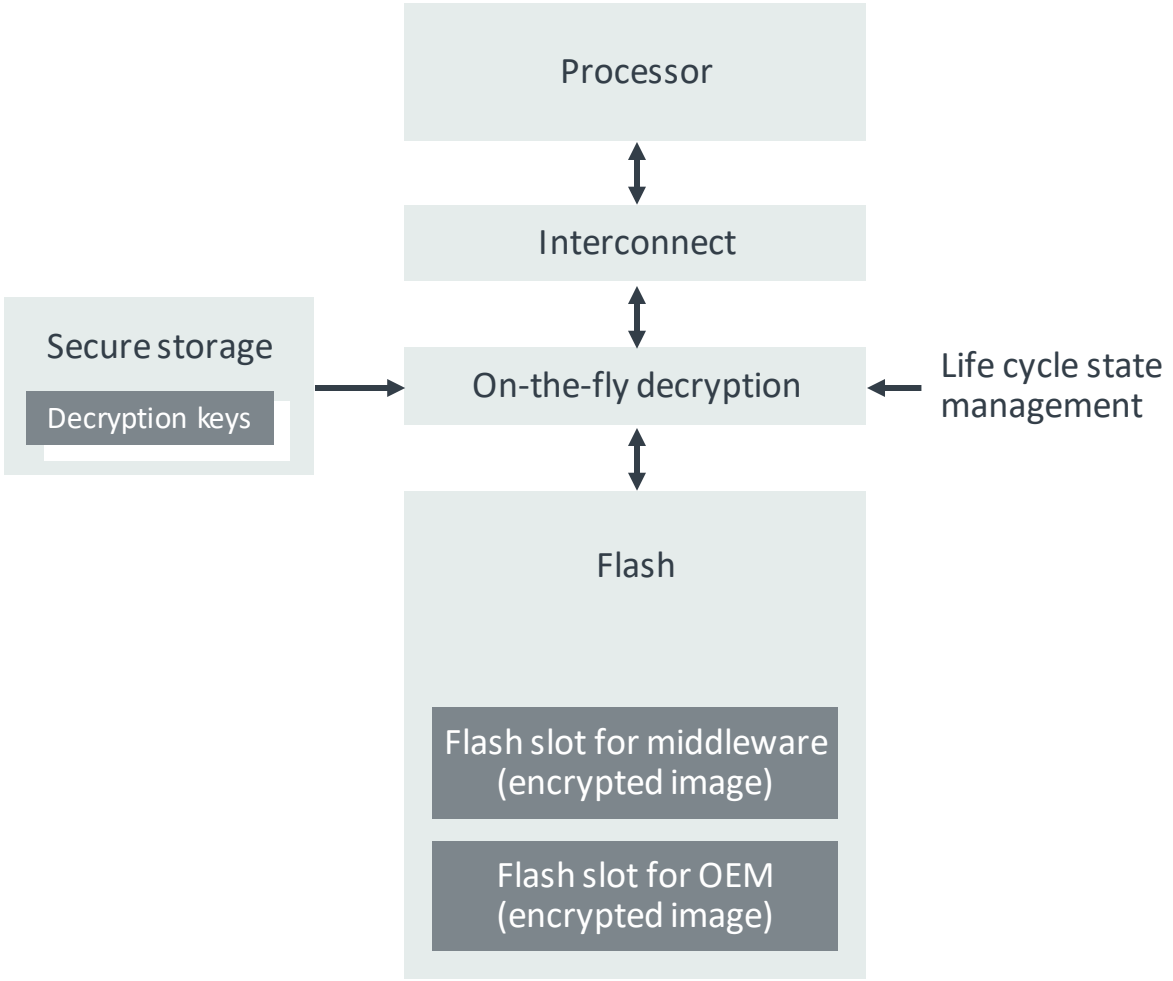
On-the-fly Decryption Engine is Needed

Security requirements

- On-the-fly decryption (library image on chip is encrypted)
- Decryption key to be installed later when the end product is activated
- Middleware vendor deliver single encrypted library image to MCU vendor
- Decryption key need to be erased at end of product life

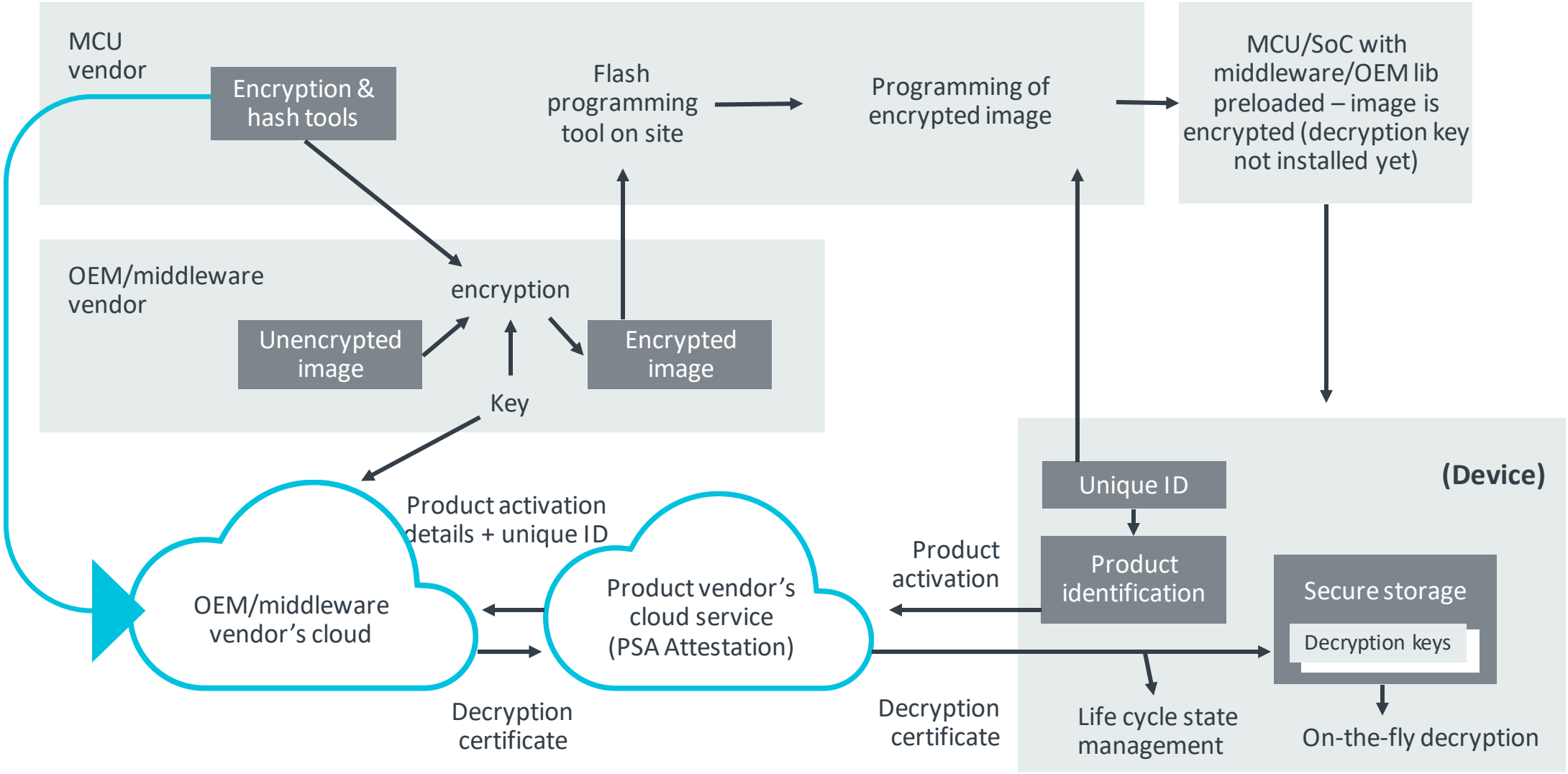
Solution

- Life cycle state management
- Secure storage
- Flash programming and key generation utilize unique device ID



arm

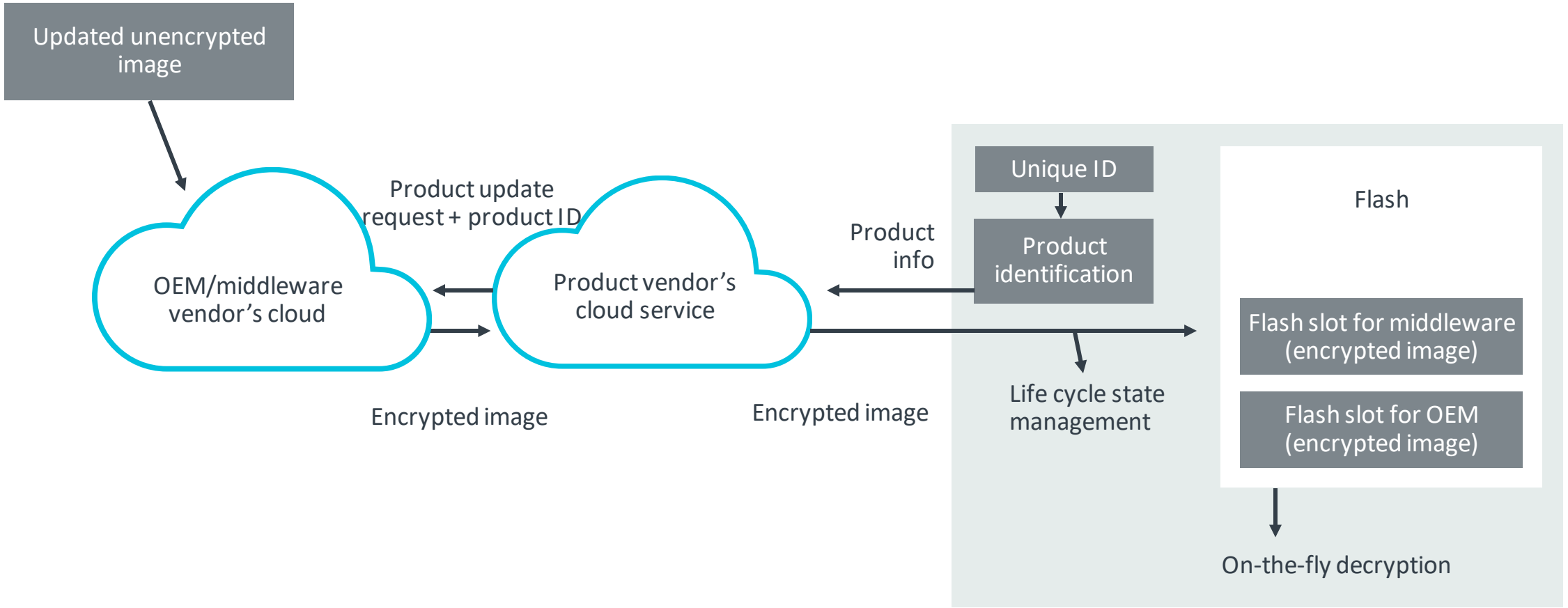
Programming of Encrypted Libraries and Key/Cert Installation



arm

Over-the-air Firmware Update of Encrypted Library

MCU vendor is not involved



arm

Summary

- The changes of security requirements in microcontrollers (MCU)
- PXN (Privileged eXecute Never) feature in Armv8.1-M
- UDE (Unprivileged Debug Extension) feature in Armv8.1-M
- Secure delivery of middleware libraries

Next Steps

Security technology development is a continuous process

TrustZone adoption is growing

- Raising awareness of security requirements
- Enabling ecosystem and open source communities (e.g. Trusted Firmware-M project)
- Enabling more designers to access security technologies
- Learn more at www.arm.com/trustzone-m

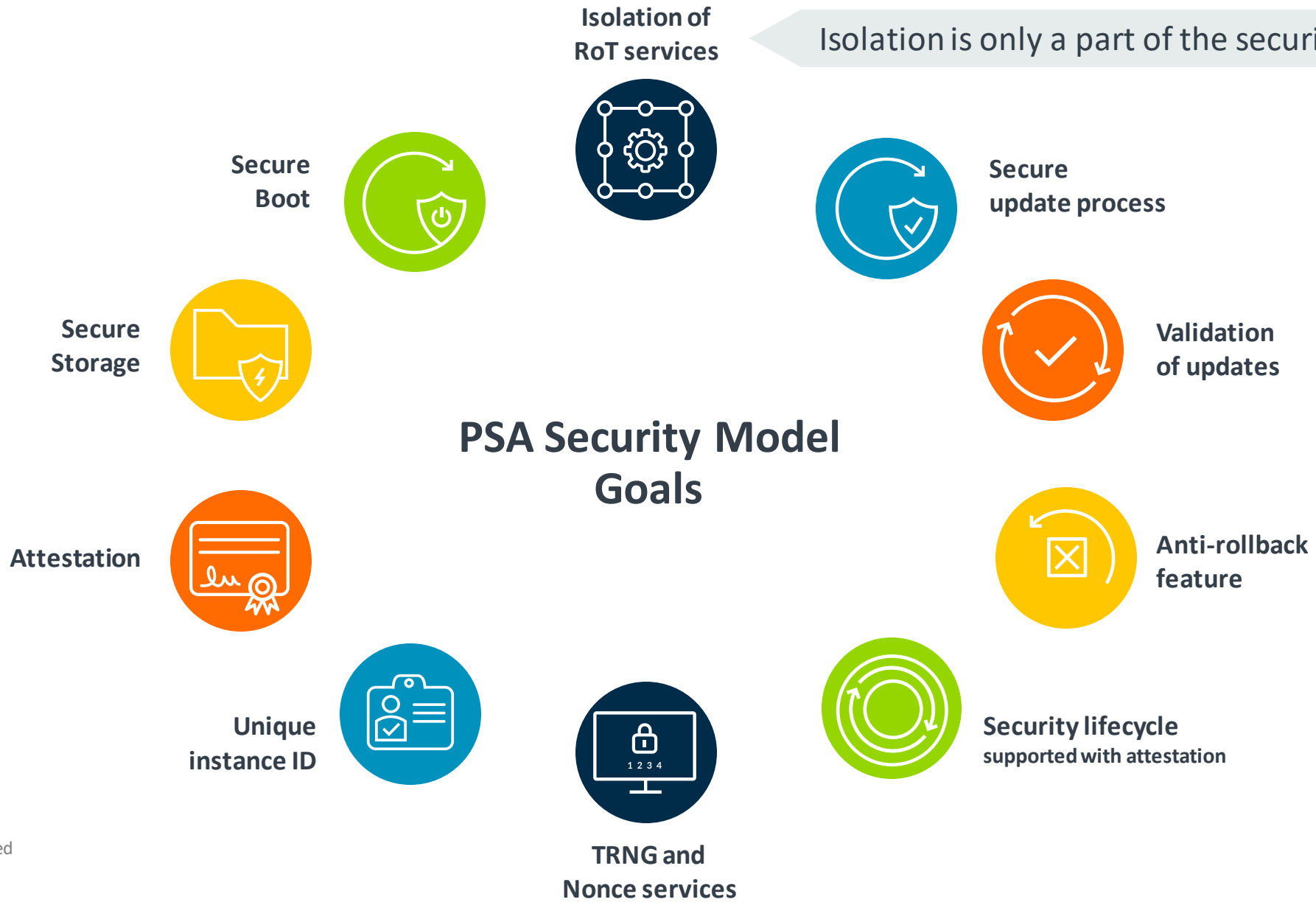
- Educating partners on the new Armv8.1-M security features

- Future technologies
  - Additional security technologies are now in research

To learn more about Armv8.1-M architecture,
please visit www.arm.com/helium

Platform Security Architecture (PSA) - Security Model Goals

Isolation is only a part of the security story



Thank You
Danke
Merci
谢谢
ありがとう
Gracias
Kiitos
감사합니다
धन्यवाद
شكرًا
תודה