black hat ASIA 2024

APRIL 18-19, 2024
BRIEFINGS

What the TrustZone-M Doesn't See, the MCU Does Grieve Over

Lessons Learned from Assessing a Microcontroller TEE

Cristiano Rodrigues | Sandro Pinto, PhD

(Centro ALGORITMI / LASI, Universidade do Minho)

What the TrustZone-M Doesn't See, the MCU Does Grieve Over

Lessons Learned from Assessing a Microcontroller TEE

Cristiano Rodrigues | Sandro Pinto, PhD

(Centro ALGORITMI / LASI, Universidade do Minho)

AGENDA

01 **Introduction**
Background and Motivation

02 **A Bumpy but Revealing Journey**
Weak Protections, TEE Assessment and our Responsible Disclosure Journey

03 **What Can Go Wrong**
Attack Examples and “Live” Demo

04 **Lessons Learned**
Advices for HW & SW providers and System Designers

05 **Summary**
Final Thoughts and BH Sound Bytes

Introduction

Background and Motivation

INTERNET OF THINGS

AI-ENABLED EDGE DEVICES

SMART FACTORIES

SMART CITIES

DRONES

SMART AGRICULTURE

HARDWARE WALLETS

MEDICAL DEVICES

WEARABLES

HOME APPLIANCES

AUTONOMOUS VEHICLES

The route to a trillion devices

The outlook for IoT investment to 2035

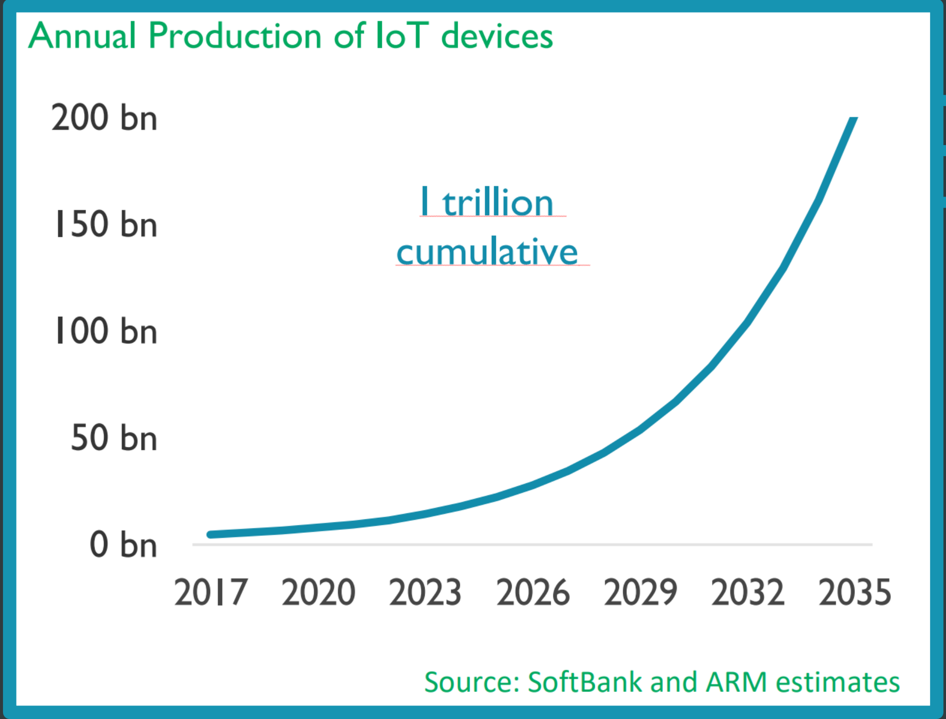
AI-ENABLED
EDGE DEVICES

SMART
CITIES

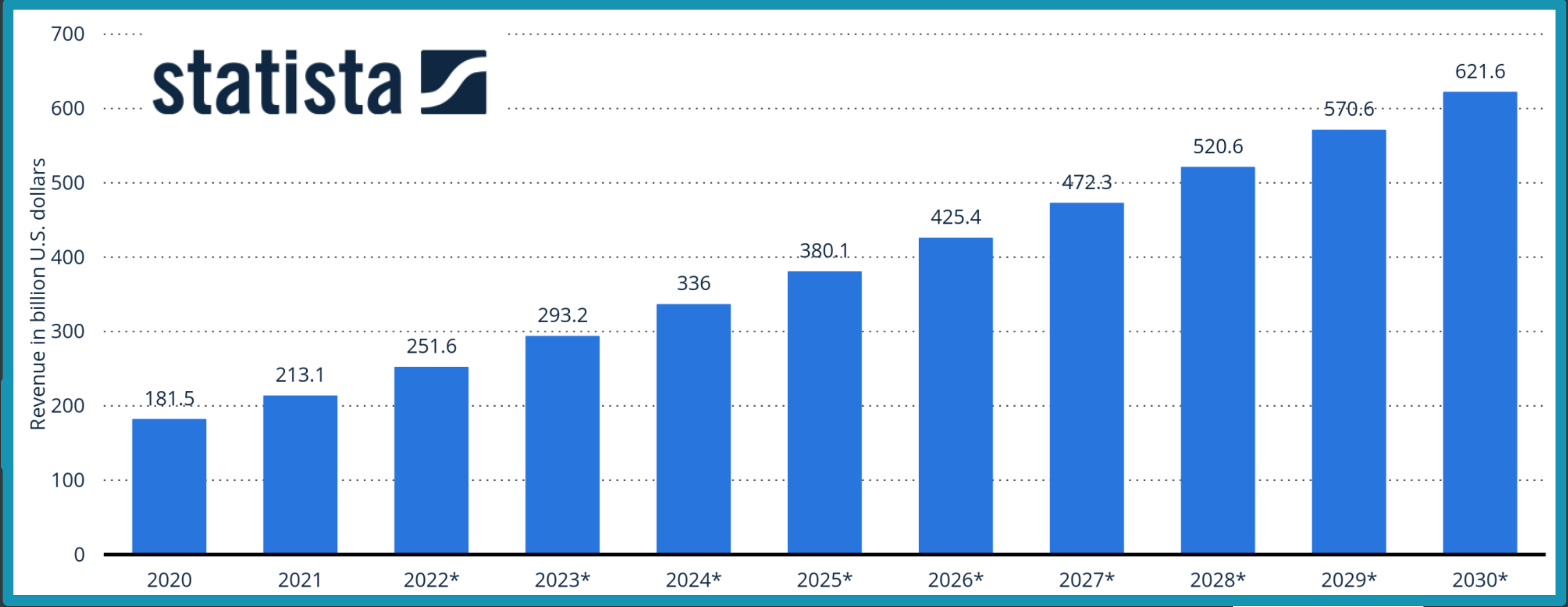
SMART
FACTORIES

DRONES

SMART
AGRICULTURE



Source: SoftBank and ARM estimates

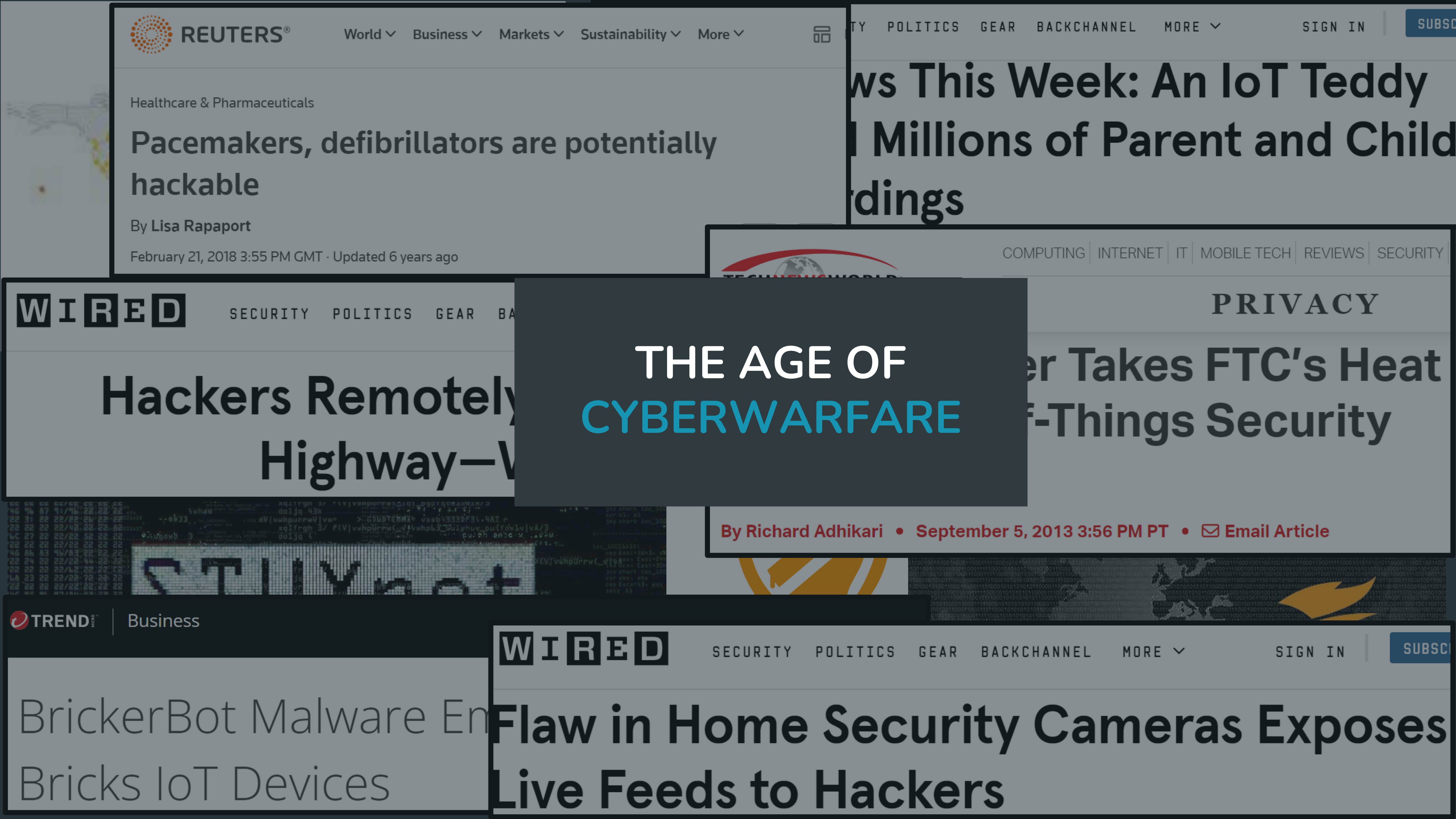


MEDICAL
DEVICES

WEARAB

VEHICLES

THE AGE OF CYBERWARFARE



REUTERS® World Business Markets Sustainability More

Healthcare & Pharmaceuticals

Pacemakers, defibrillators are potentially hackable

By Lisa Rapaport

February 21, 2018 3:55 PM GMT · Updated 6 years ago

TY POLITICS GEAR BACKCHANNEL MORE SIGN IN SUBSC

ws This Week: An IoT Teddy

Millions of Parent and Child

dings

WIRED SECURITY POLITICS GEAR BA

Hackers Remotely

Highway—V

COMPUTING INTERNET IT MOBILE TECH REVIEWS SECURITY

PRIVACY

er Takes FTC's Heat

-Things Security

By Richard Adhikari • September 5, 2013 3:56 PM PT • ✉ Email Article

TREND MICRO Business

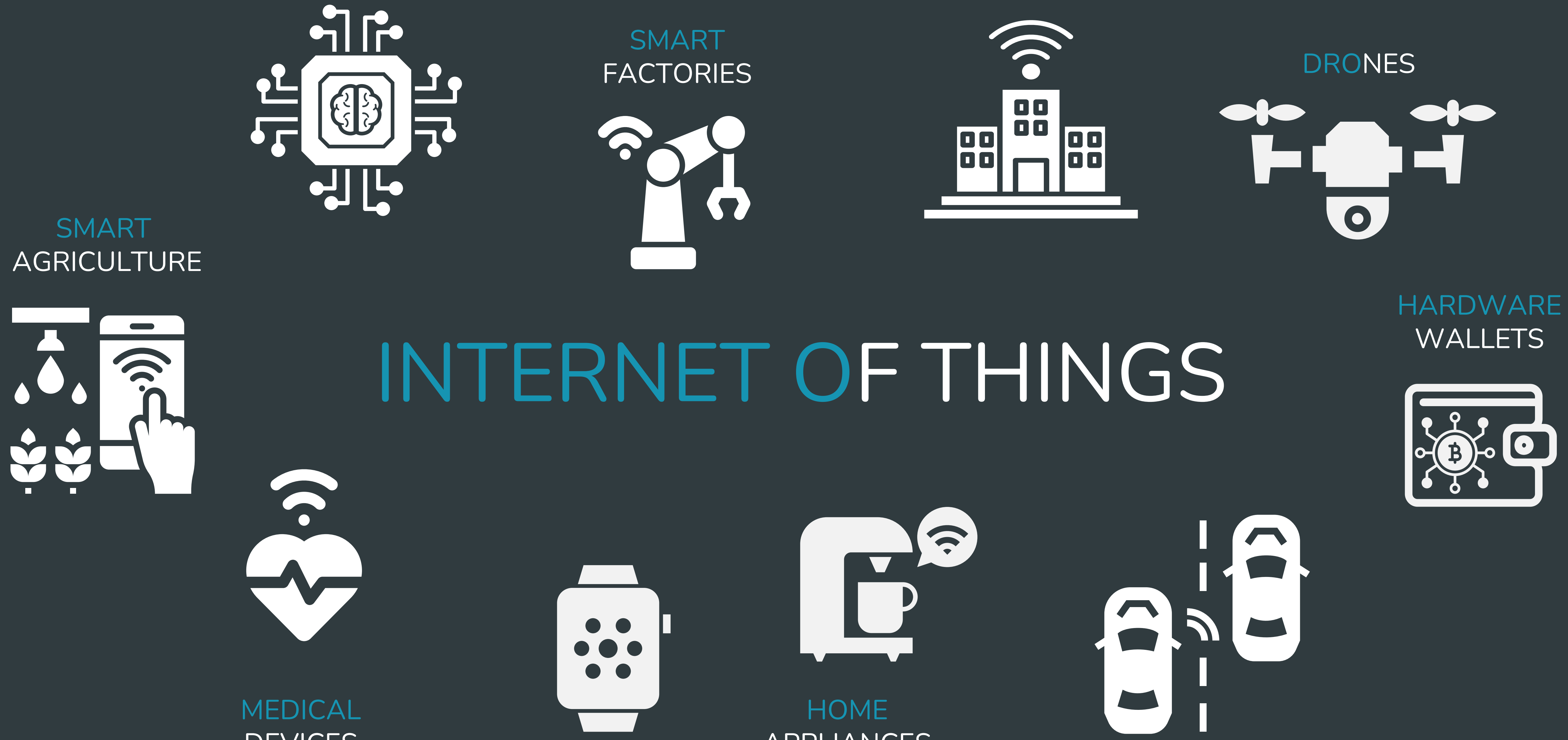
BrickerBot Malware En

Bricks IoT Devices

WIRED SECURITY POLITICS GEAR BACKCHANNEL MORE SIGN IN SUBSC

Flaw in Home Security Cameras Exposes Live Feeds to Hackers

INTERNET OF THINGS



AI-ENABLED EDGE DEVICES

SMART FACTORIES

SMART CITIES

DRONES

SMART AGRICULTURE

HARDWARE WALLETS

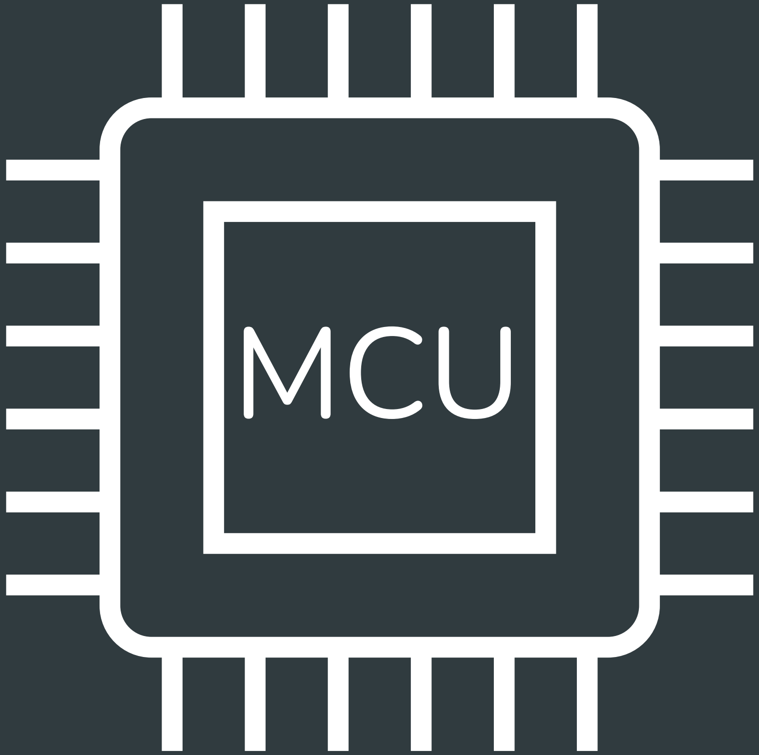
MEDICAL DEVICES

WEARABLES

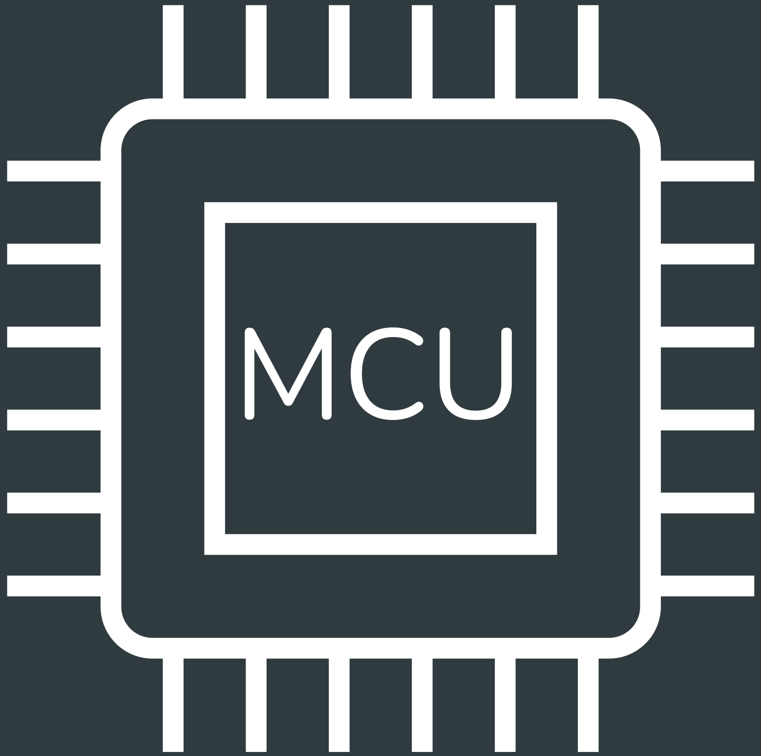
HOME APPLIANCES

AUTONOMOUS VEHICLES

INTERNET OF THINGS



INTERNET OF THINGS



arm

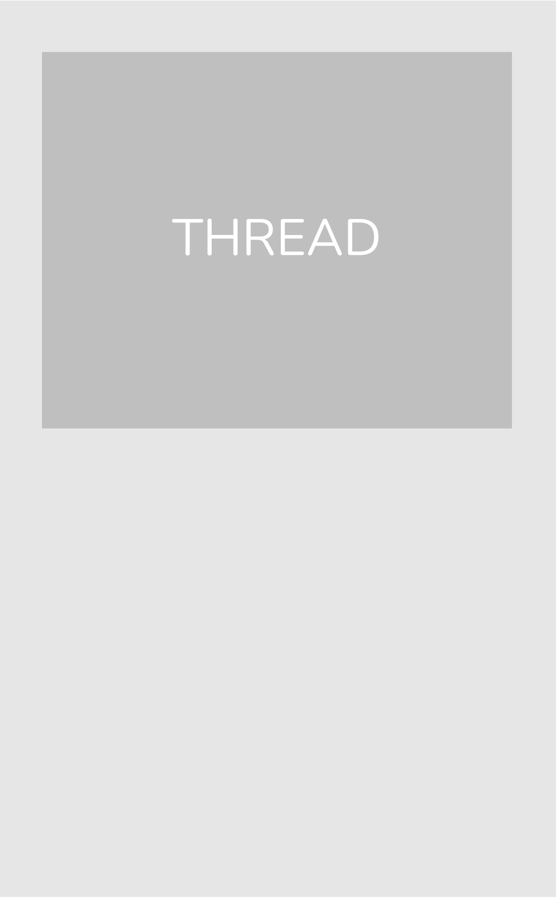
TRUSTZONE

Armv8-M TrustZone

Armv6/7-M Processor Modes

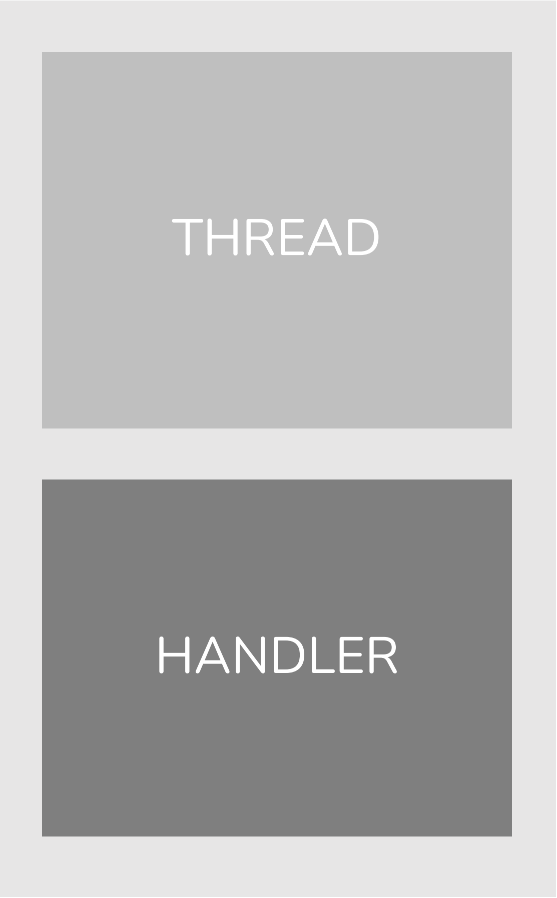
Armv8-M TrustZone

Armv6/7-M Processor Modes



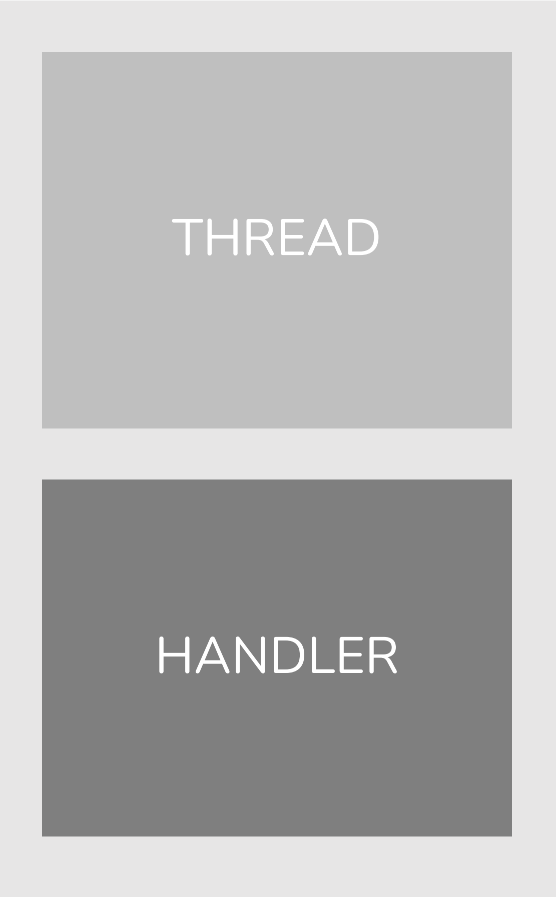
Armv8-M TrustZone

Armv6/7-M Processor Modes



Armv8-M TrustZone

Armv6/7-M Processor Modes

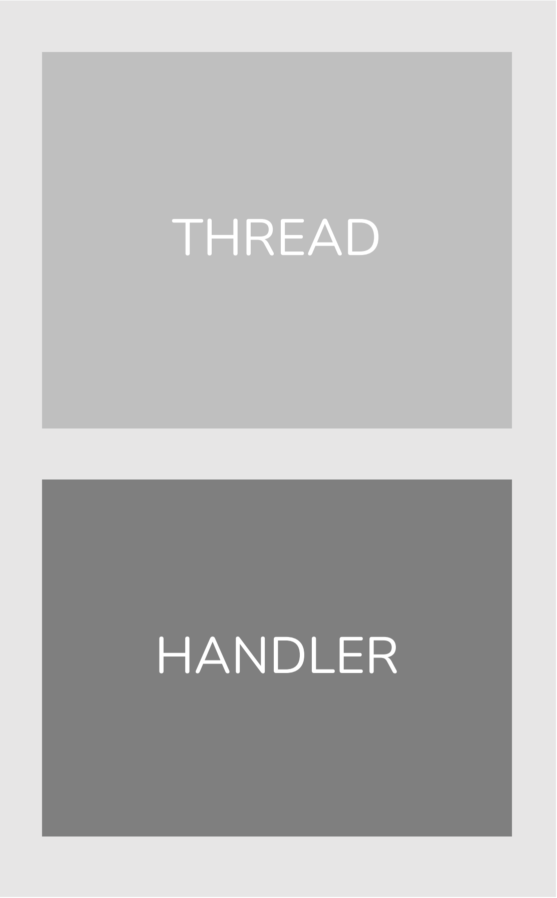

THREAD

HANDLER

Armv6/7-M Privileges Levels

Armv8-M TrustZone

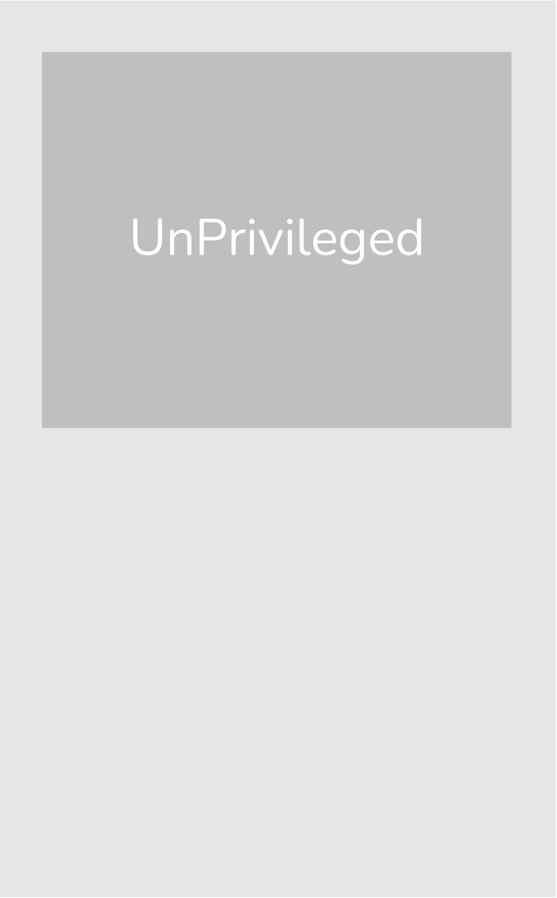
Armv6/7-M Processor Modes



THREAD

HANDLER

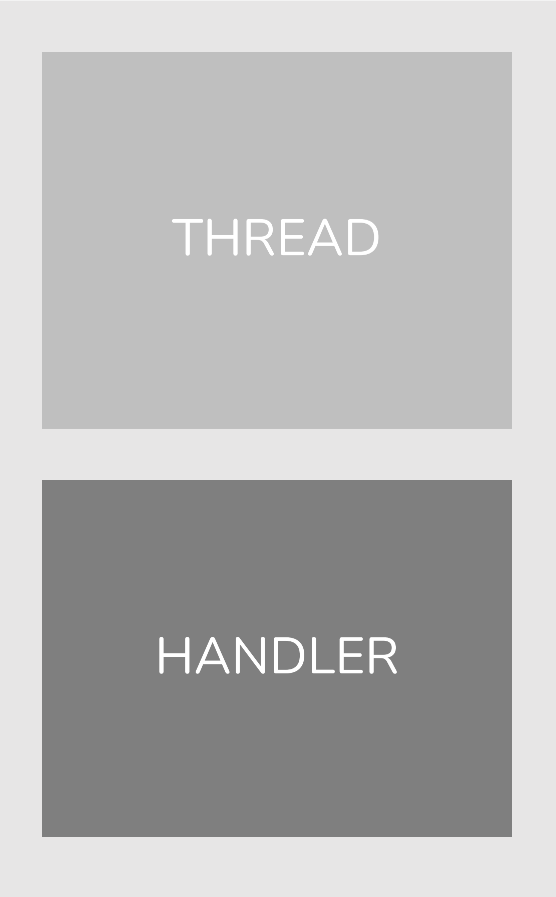
Armv6/7-M Privileges Levels



UnPrivileged

Armv8-M TrustZone

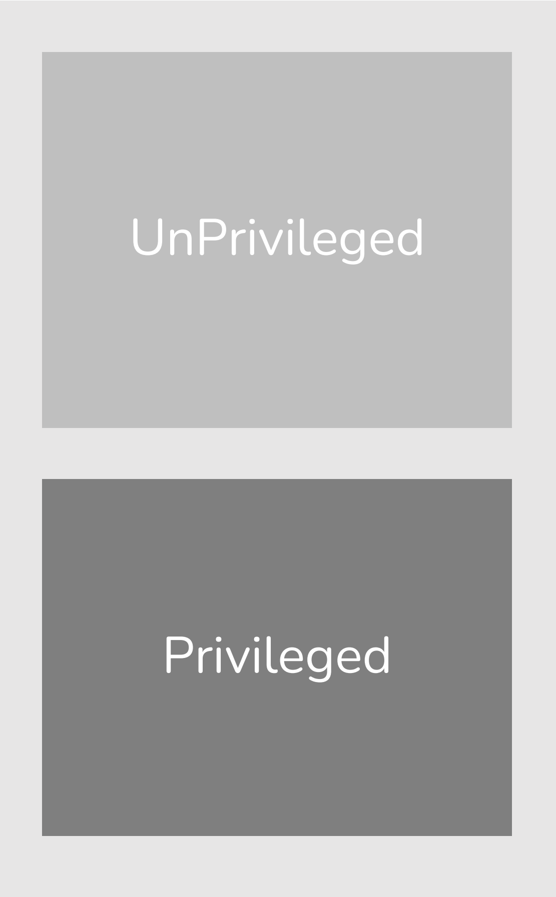
Armv6/7-M Processor Modes



THREAD

HANDLER

Armv6/7-M Privileges Levels



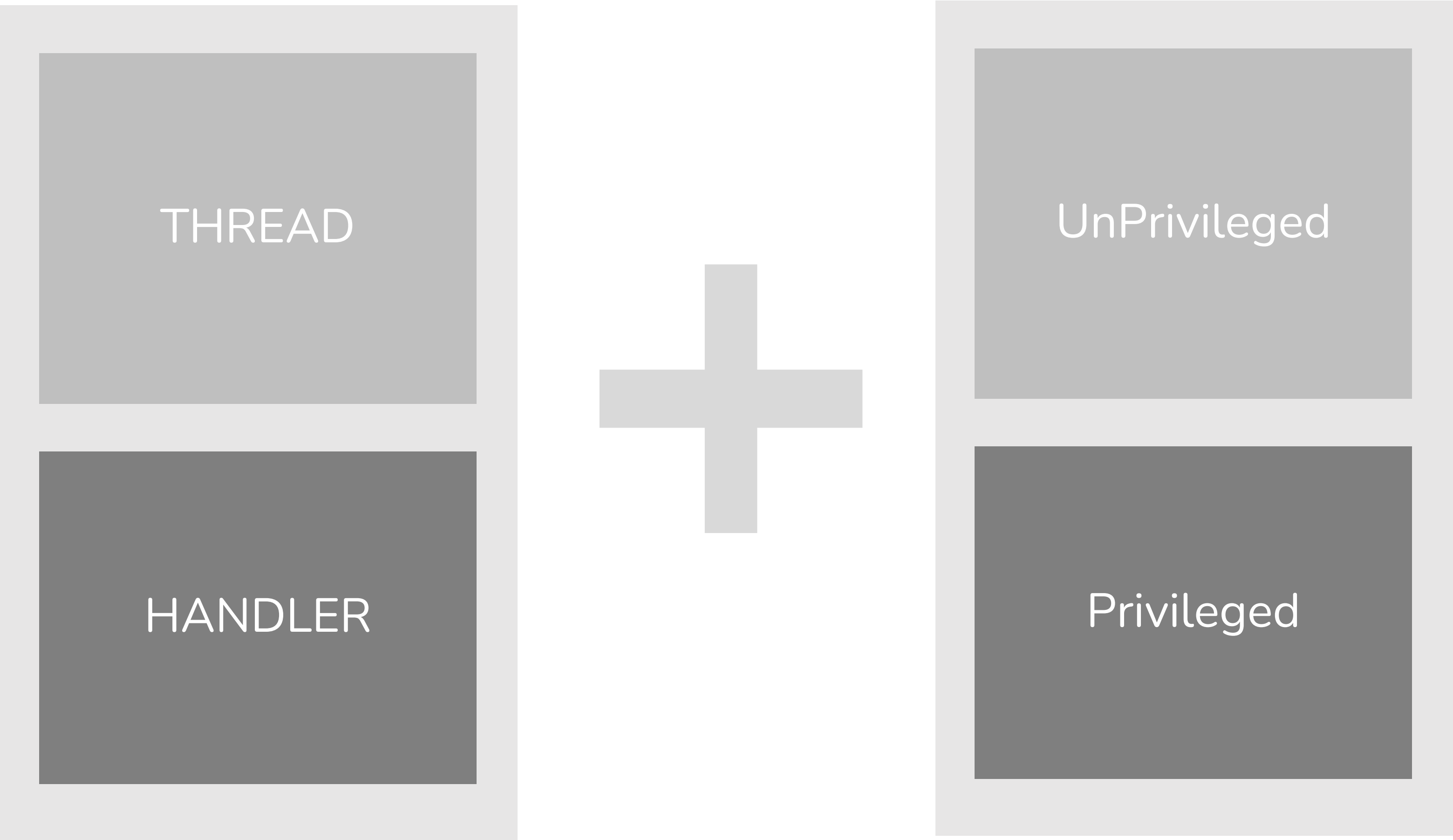
UnPrivileged

Privileged

Armv8-M TrustZone

Armv6/7-M Processor Modes

Armv6/7-M Privileges Levels



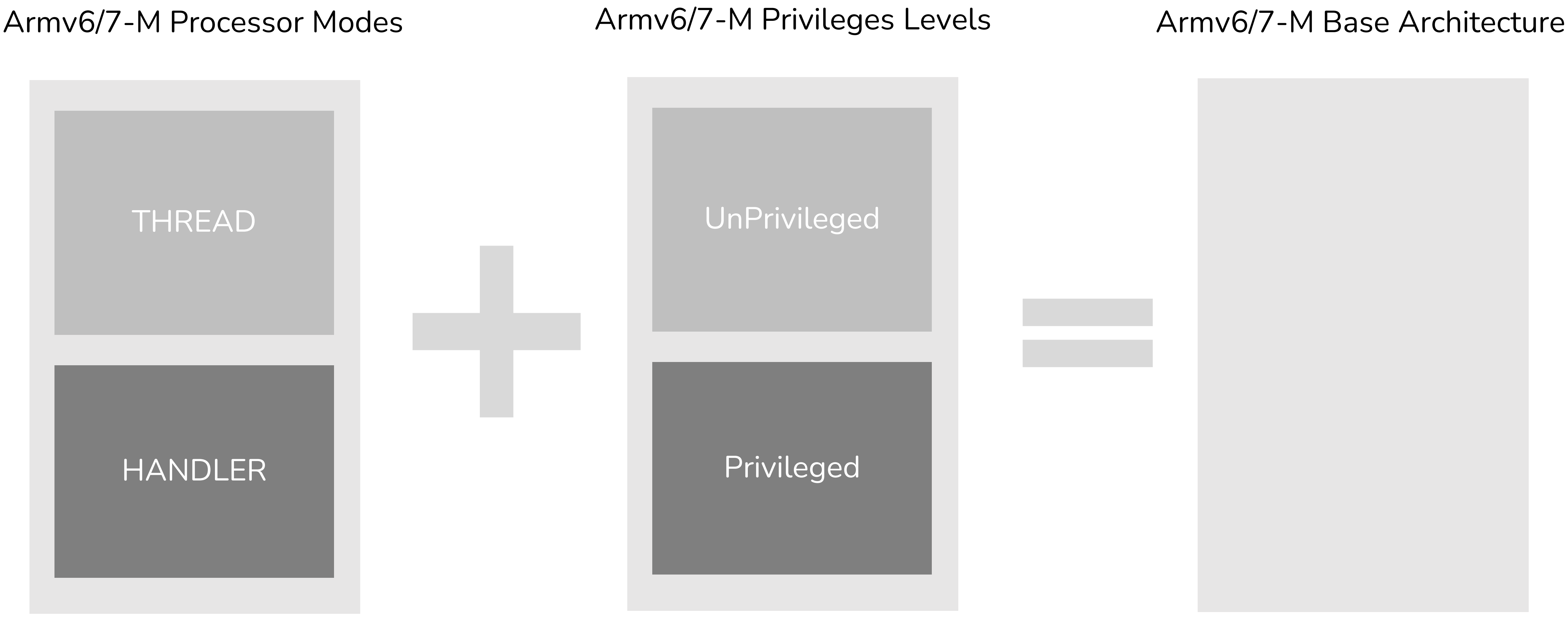
THREAD

HANDLER

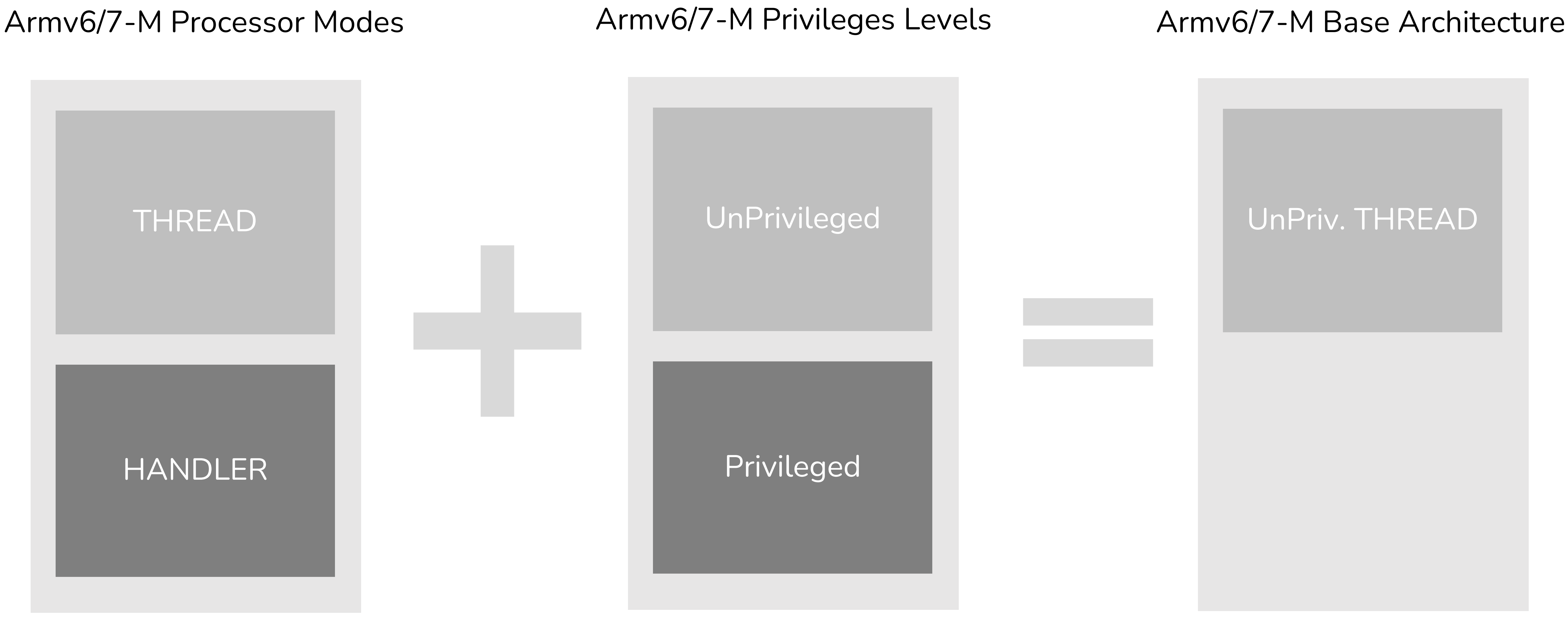
UnPrivileged

Privileged

Armv8-M TrustZone

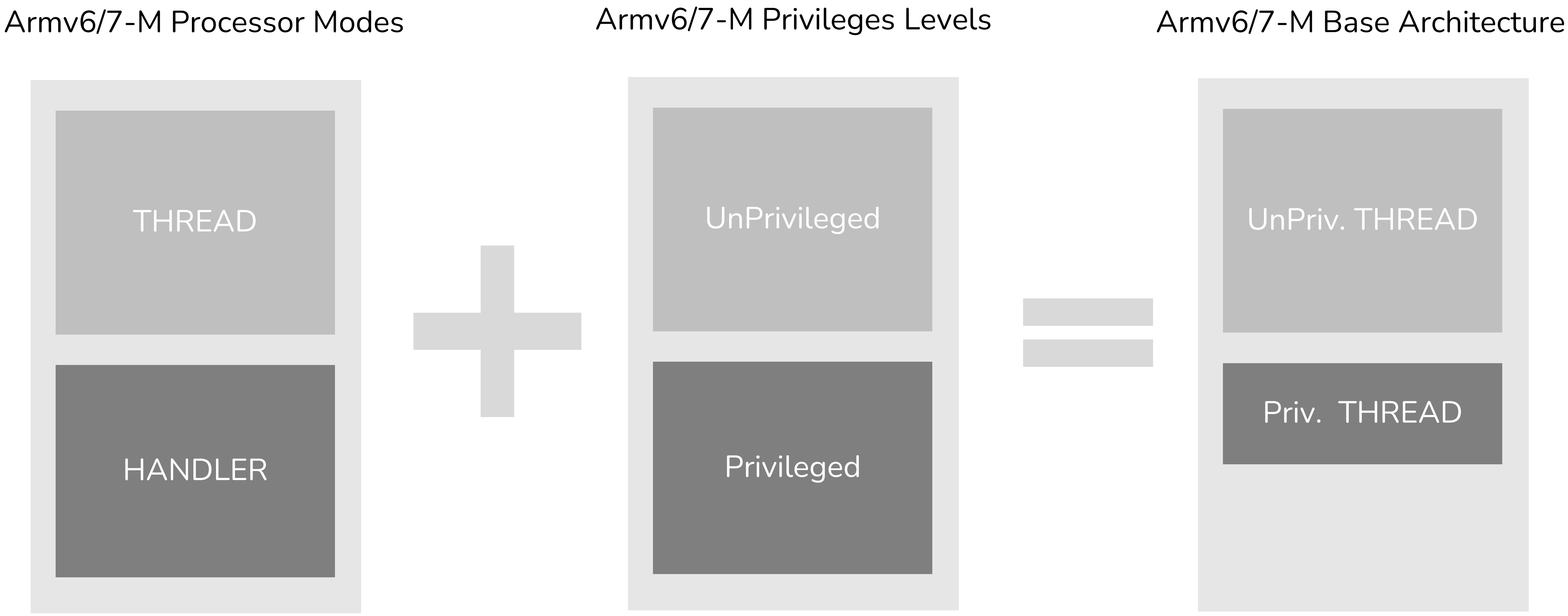

Armv6/7-M Processor Modes
THREAD
HANDLER
Armv6/7-M Privileges Levels
UnPrivileged
Privileged
Armv6/7-M Base Architecture

Armv8-M TrustZone

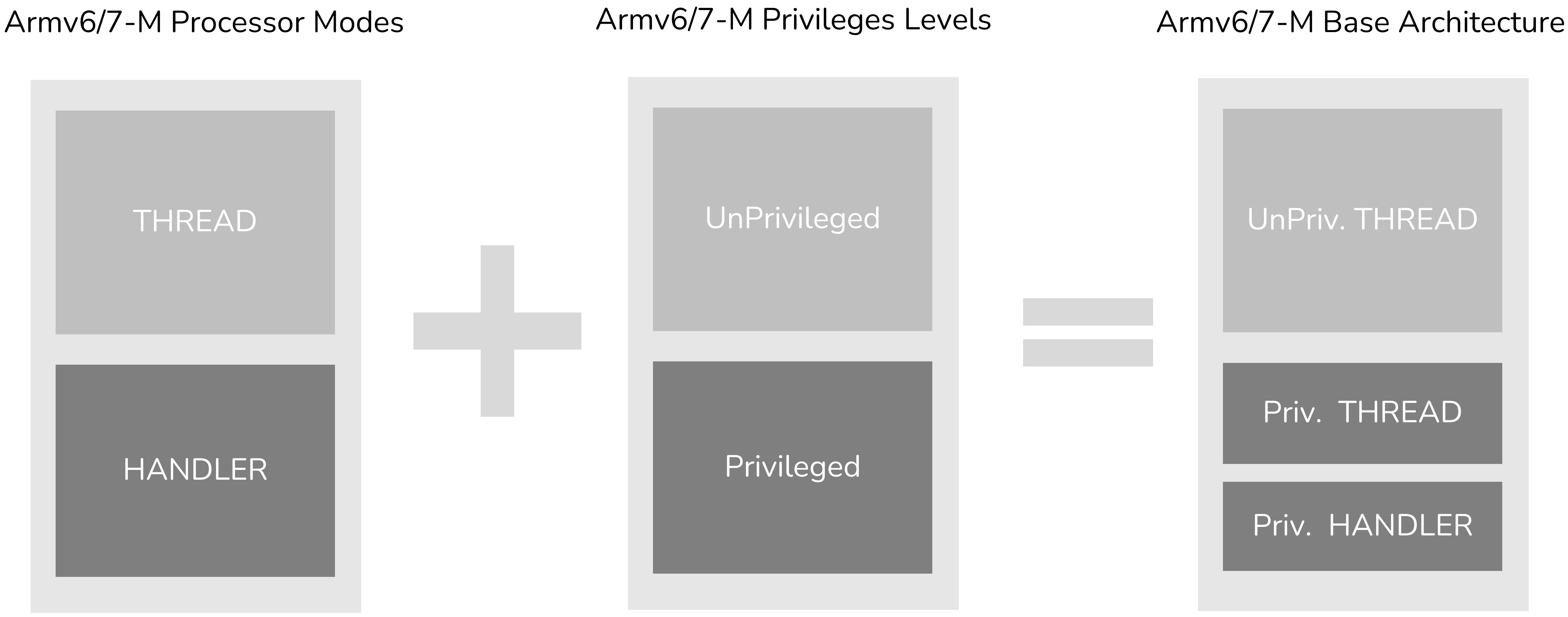


Armv6/7-M Processor Modes
THREAD
HANDLER
Armv6/7-M Privileges Levels
UnPrivileged
Privileged
Armv6/7-M Base Architecture
UnPriv. THREAD

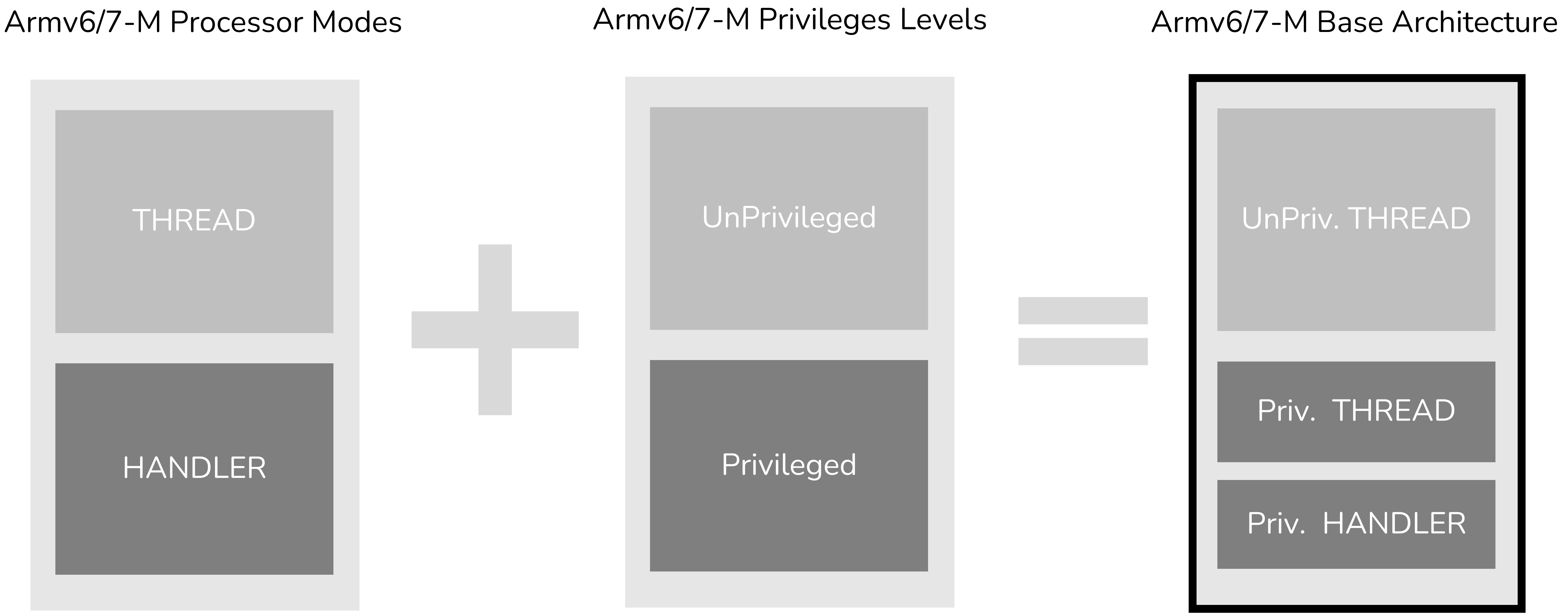
Armv8-M TrustZone


Armv6/7-M Processor Modes
THREAD
HANDLER
Armv6/7-M Privileges Levels
UnPrivileged
Privileged
Armv6/7-M Base Architecture
UnPriv. THREAD
Priv. THREAD

Armv8-M TrustZone

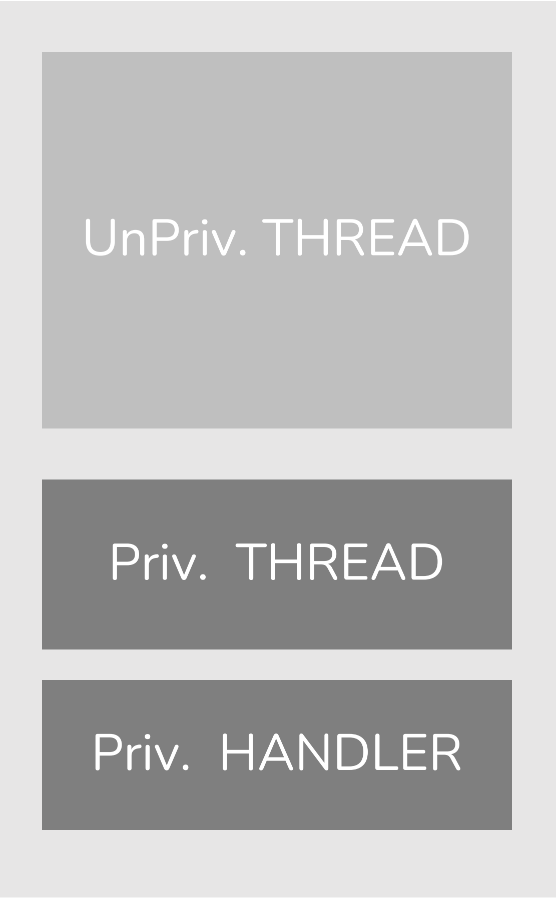

Armv6/7-M Processor Modes
THREAD
HANDLER
Armv6/7-M Privileges Levels
UnPrivileged
Privileged
Armv6/7-M Base Architecture
UnPriv. THREAD
Priv. THREAD
Priv. HANDLER

Armv8-M TrustZone


Armv6/7-M Processor Modes
THREAD
HANDLER
Armv6/7-M Privileges Levels
UnPrivileged
Privileged
Armv6/7-M Base Architecture
UnPriv. THREAD
Priv. THREAD
Priv. HANDLER

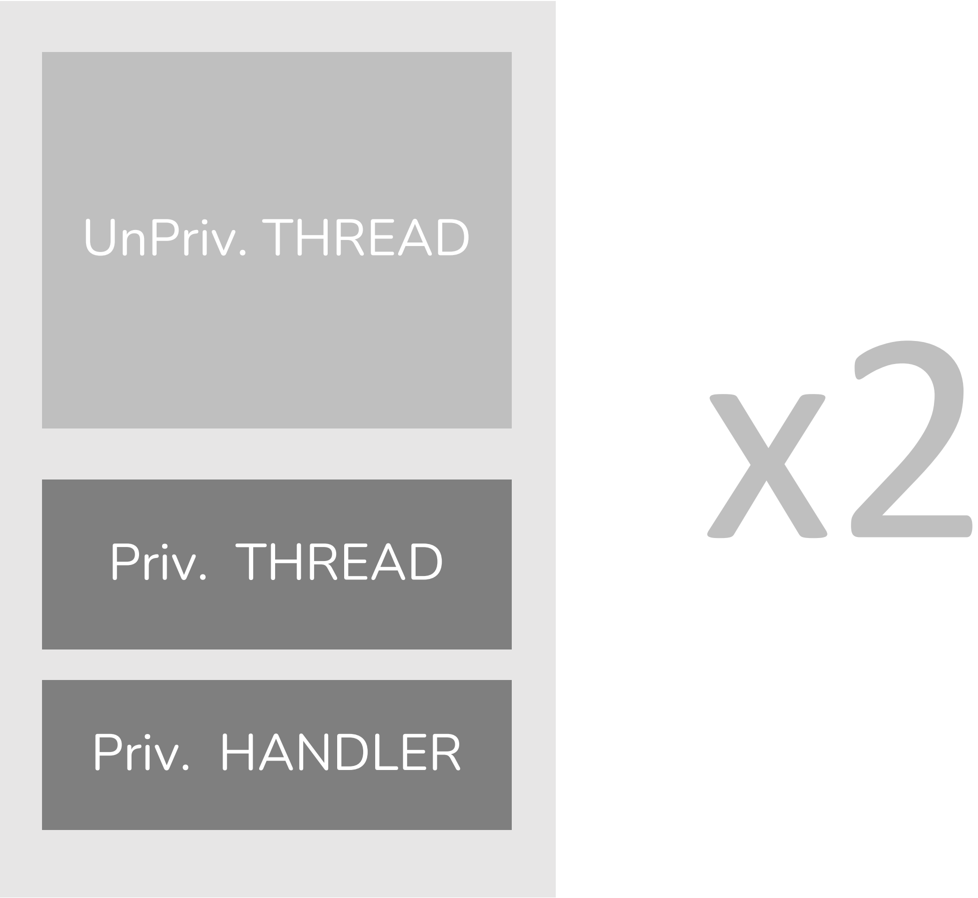
Armv8-M TrustZone

Armv6/7-M Base Architecture



Armv8-M TrustZone

Armv6/7-M Base Architecture



UnPriv. THREAD

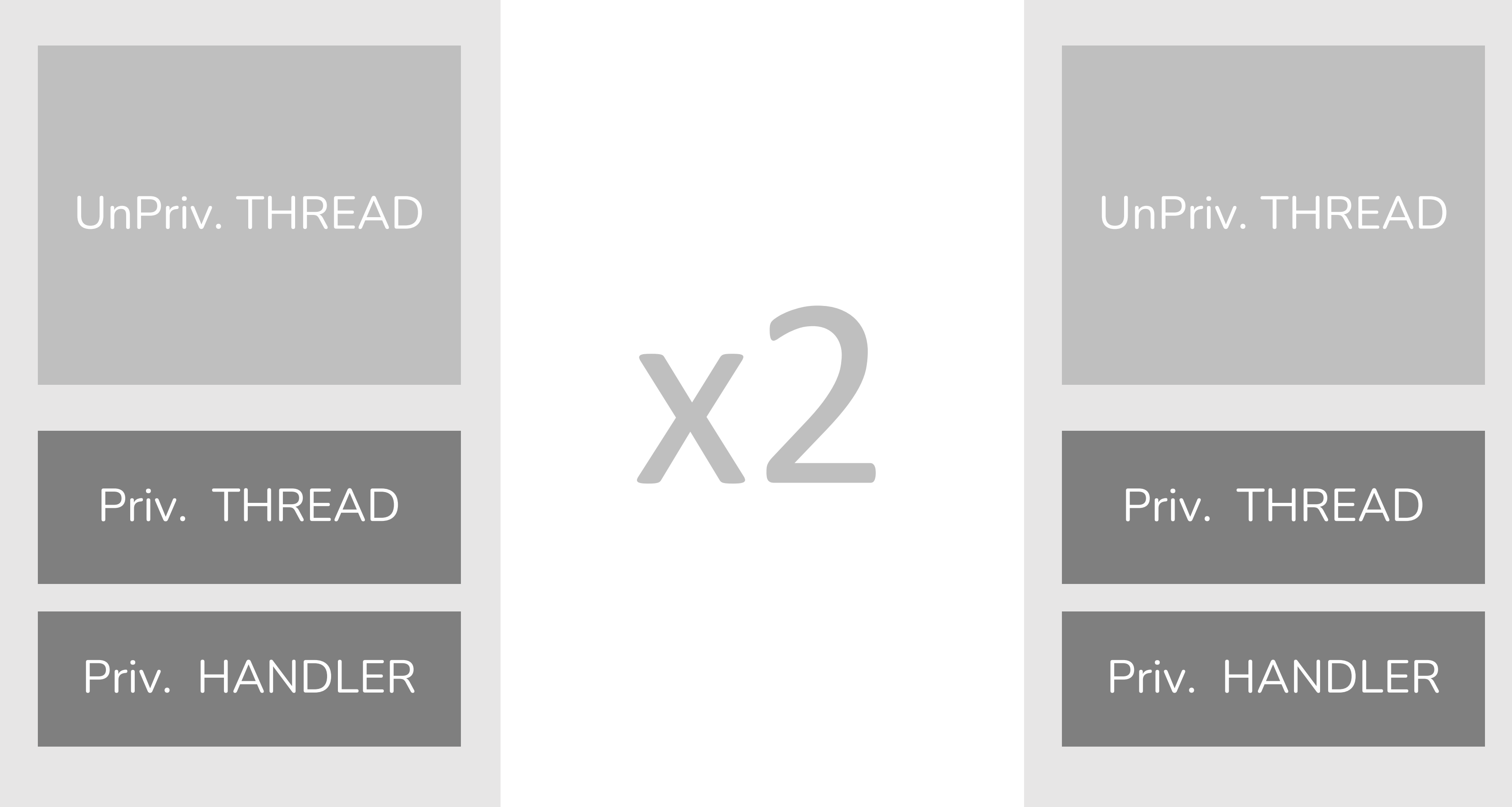
Priv. THREAD

Priv. HANDLER

x2

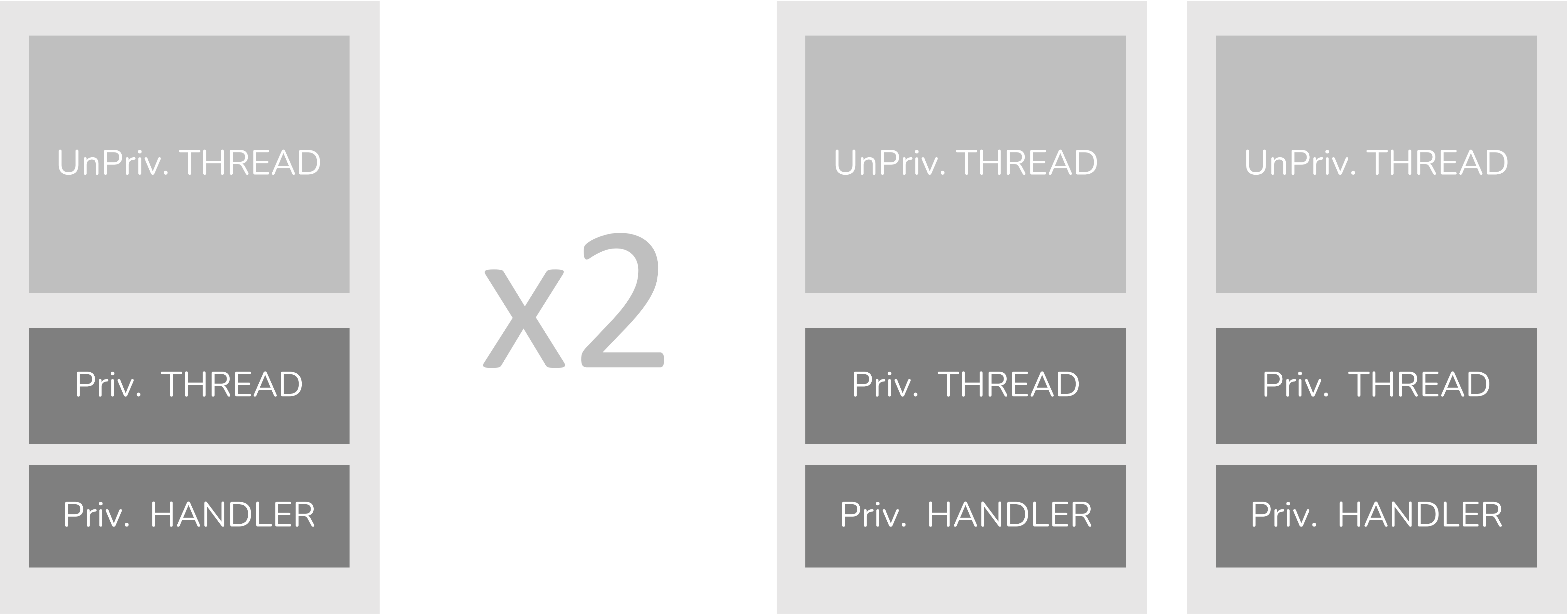
Armv8-M TrustZone

Armv6/7-M Base Architecture



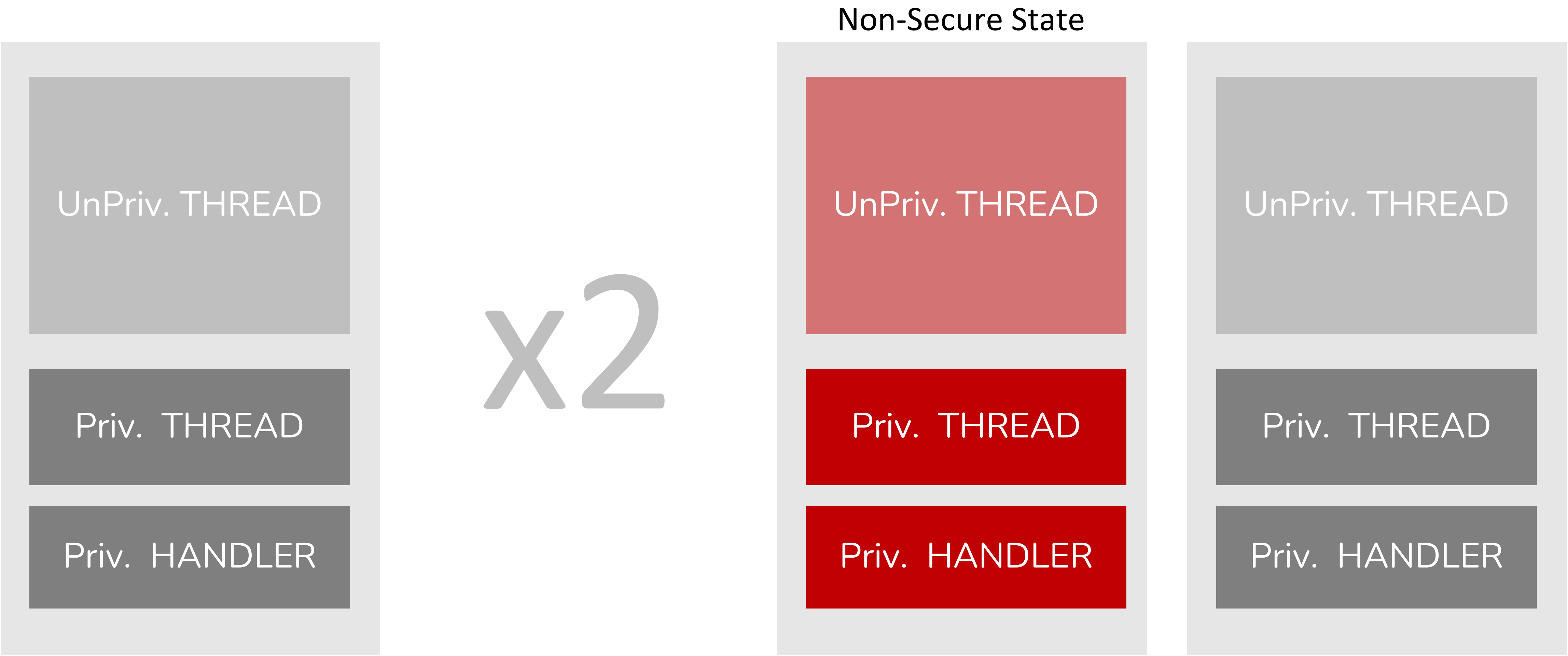
Armv8-M TrustZone

Armv6/7-M Base Architecture

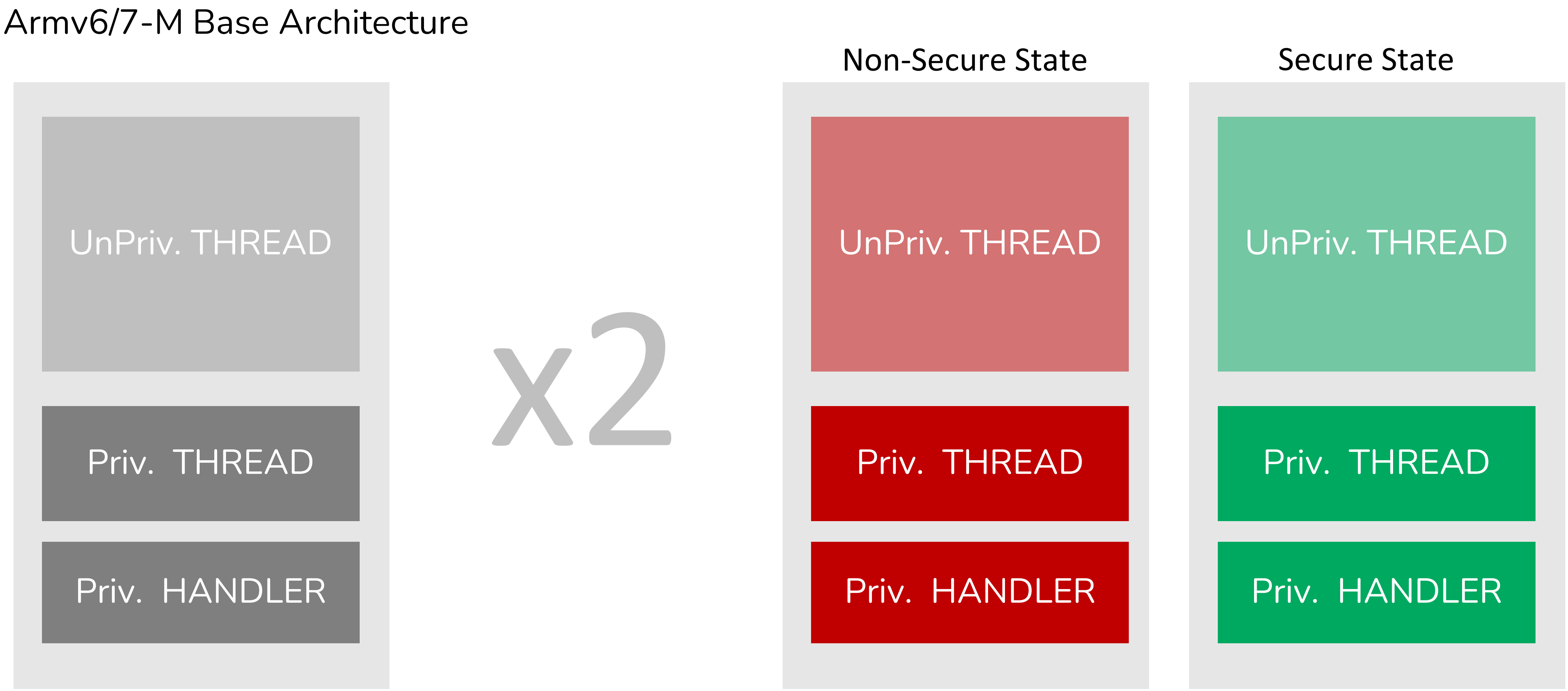


Armv8-M TrustZone

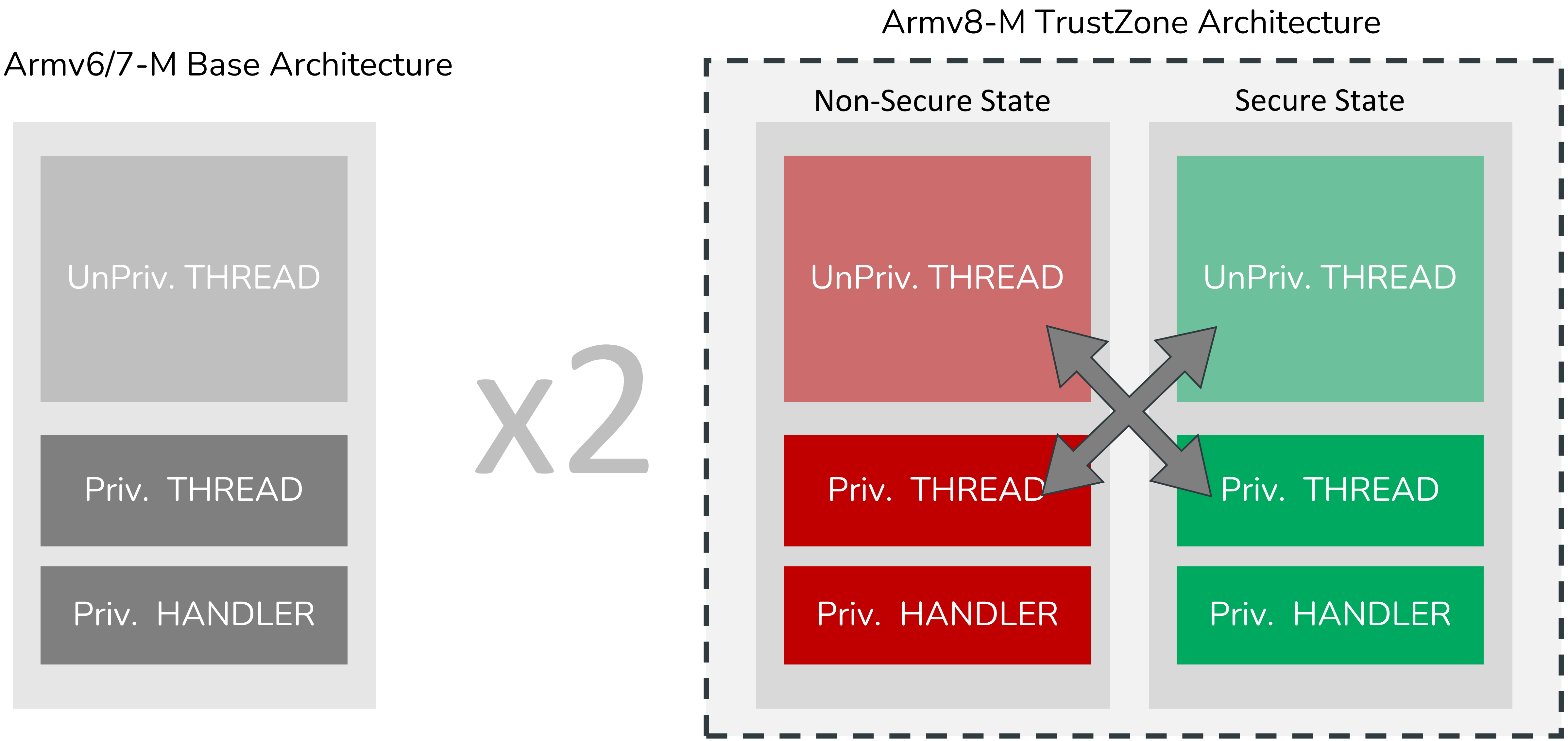
Armv6/7-M Base Architecture



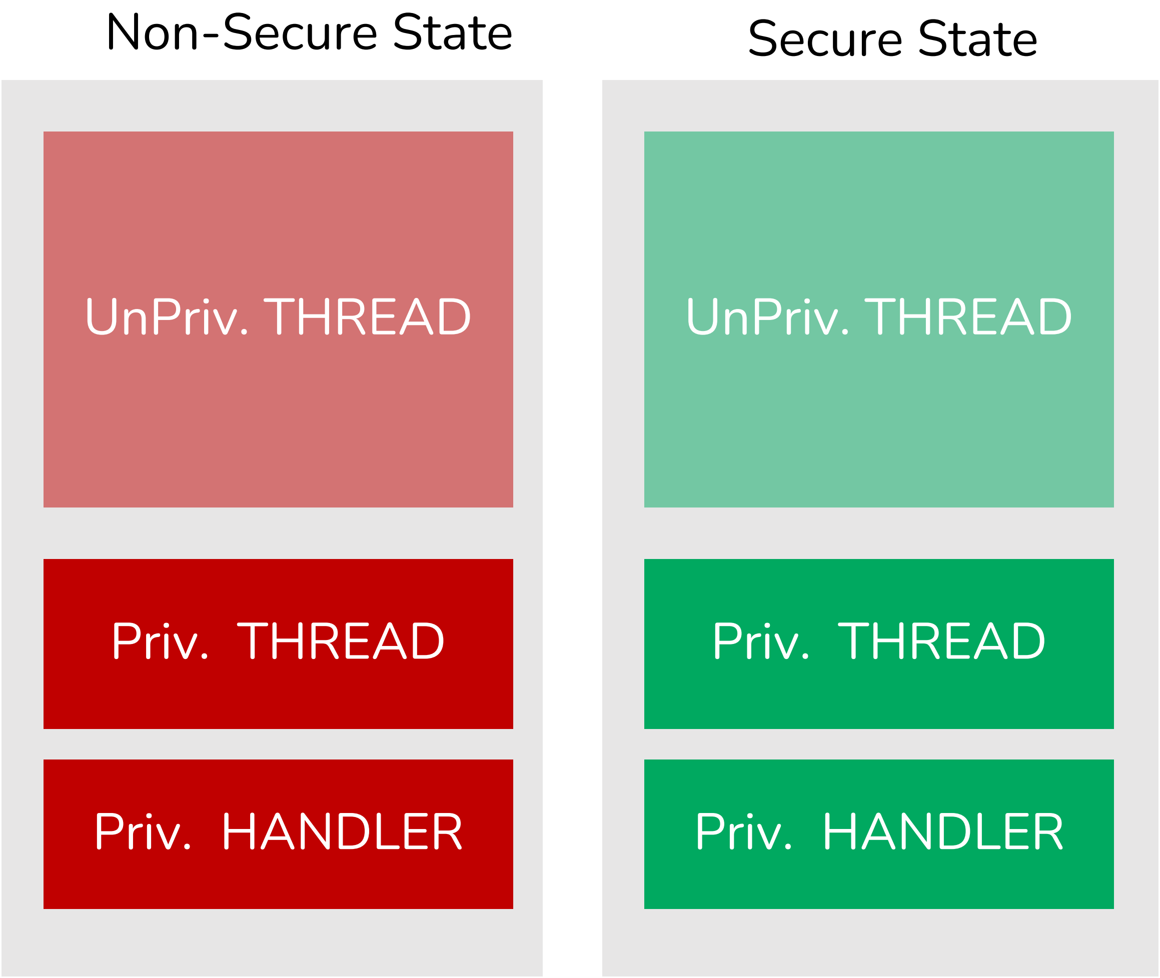
Armv8-M TrustZone



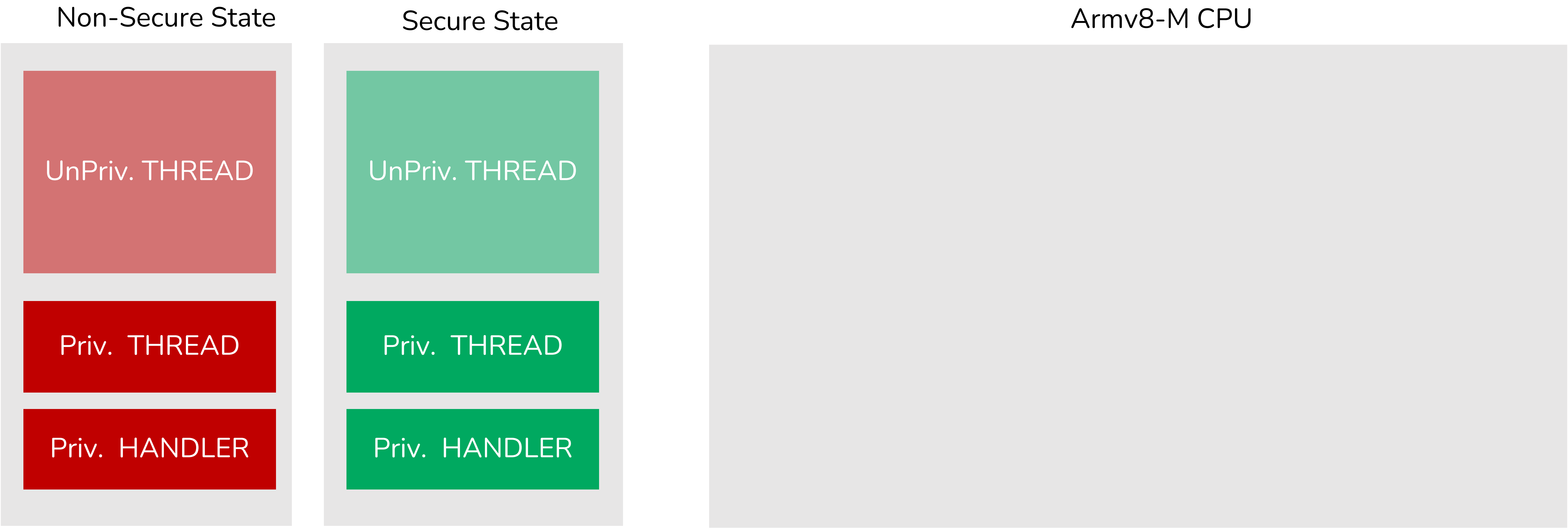
Armv8-M TrustZone



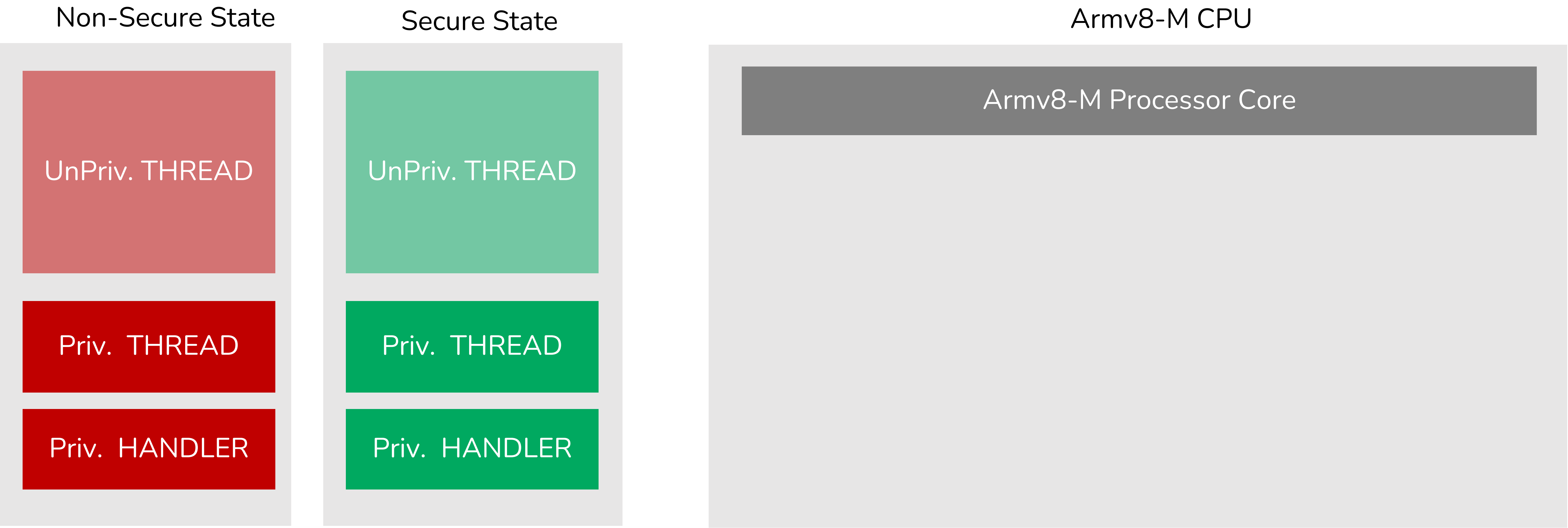
Armv8-M TrustZone



Armv8-M TrustZone



Armv8-M TrustZone



Non-Secure State

UnPriv. THREAD

Priv. THREAD

Priv. HANDLER

Secure State

UnPriv. THREAD

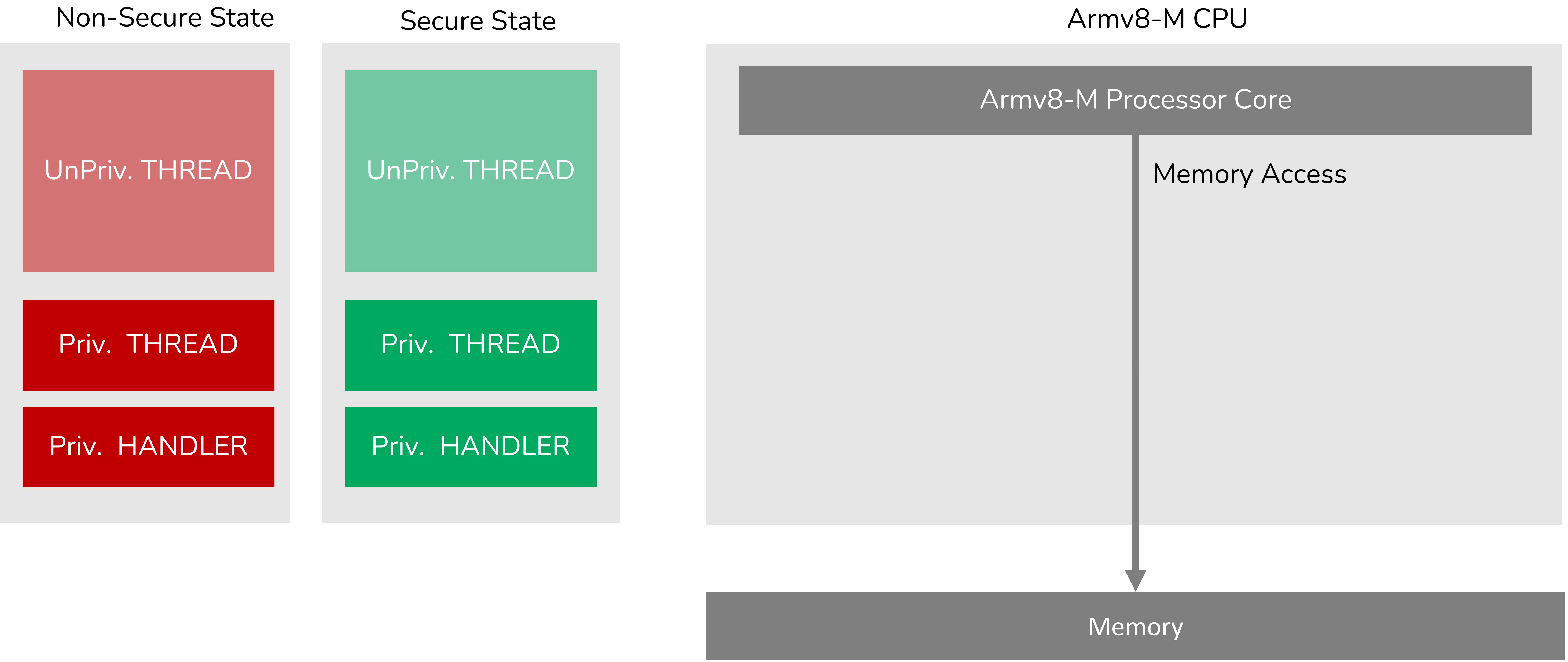
Priv. THREAD

Priv. HANDLER

Armv8-M CPU

Armv8-M Processor Core

Armv8-M TrustZone



Non-Secure State

UnPriv. THREAD

Priv. THREAD

Priv. HANDLER

Secure State

UnPriv. THREAD

Priv. THREAD

Priv. HANDLER

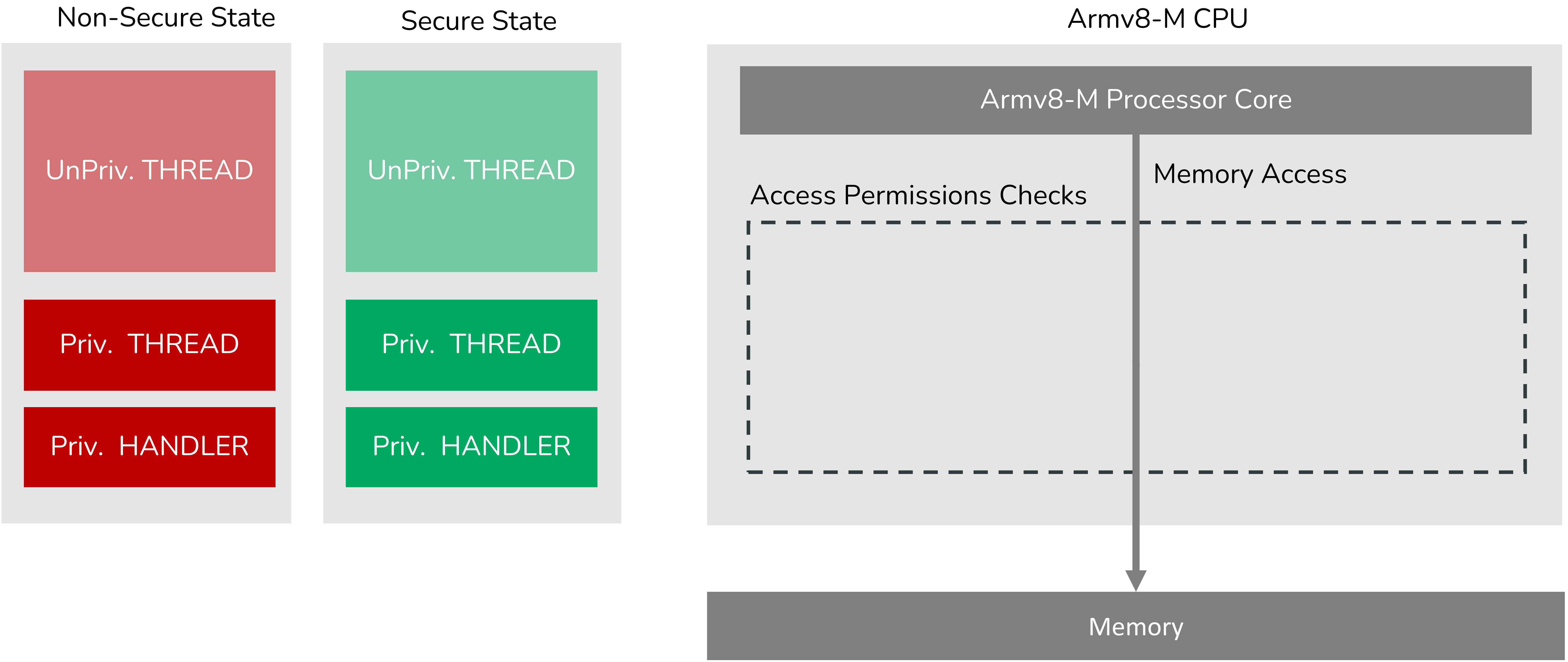
Armv8-M CPU

Armv8-M Processor Core

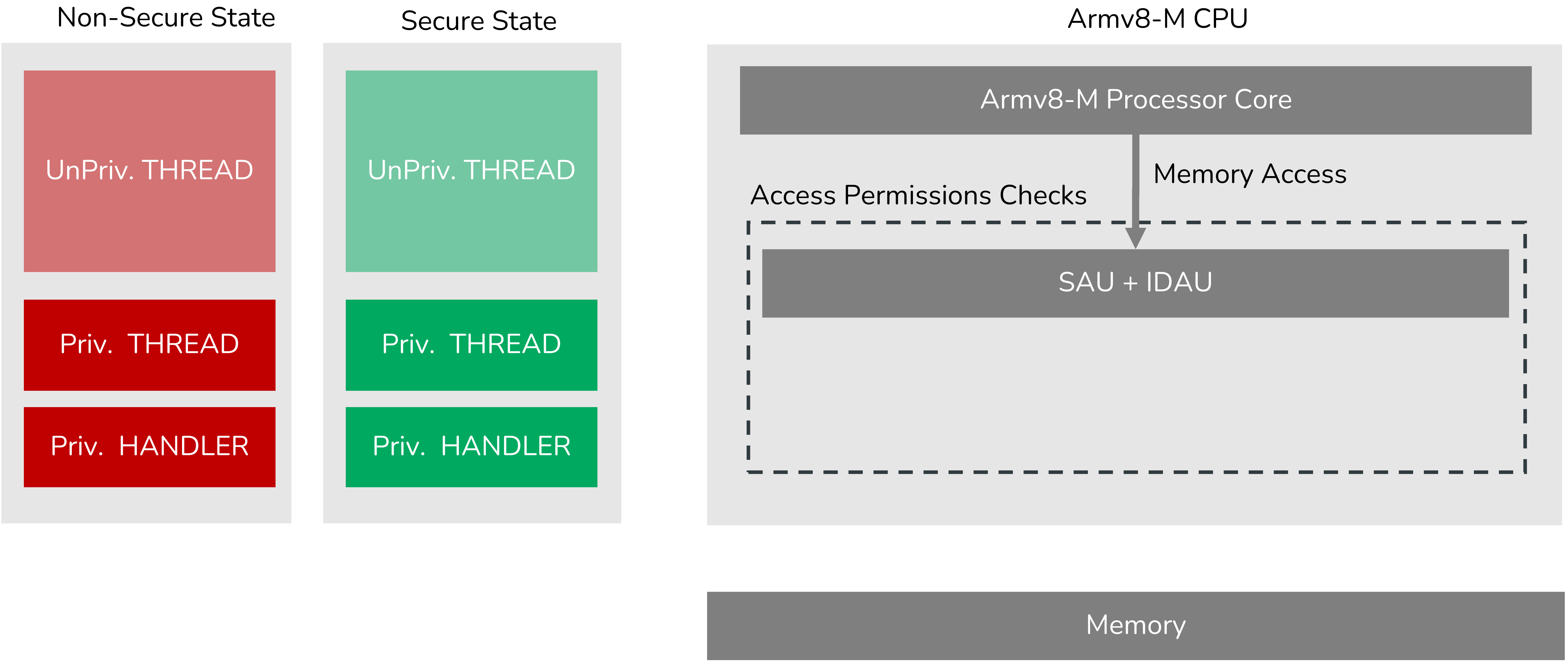
Memory Access

Memory

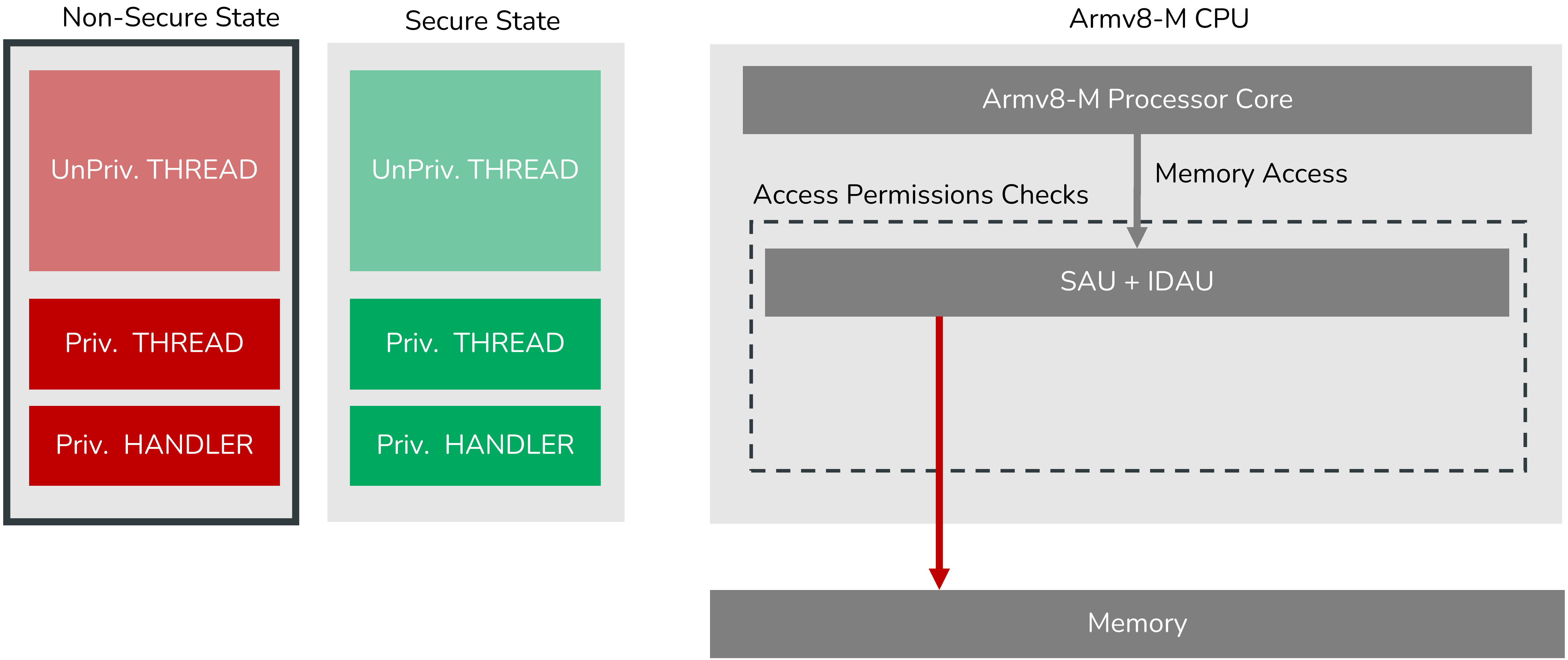
Armv8-M TrustZone



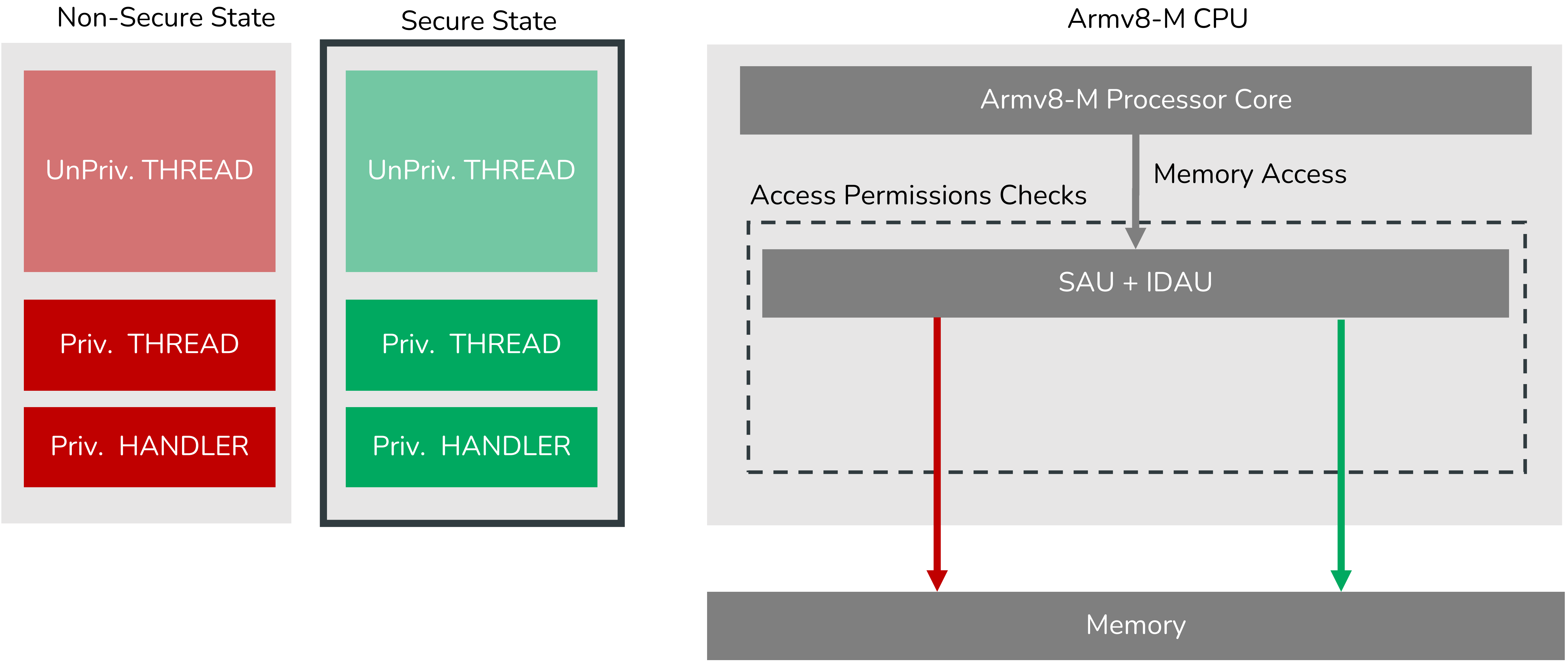
Armv8-M TrustZone



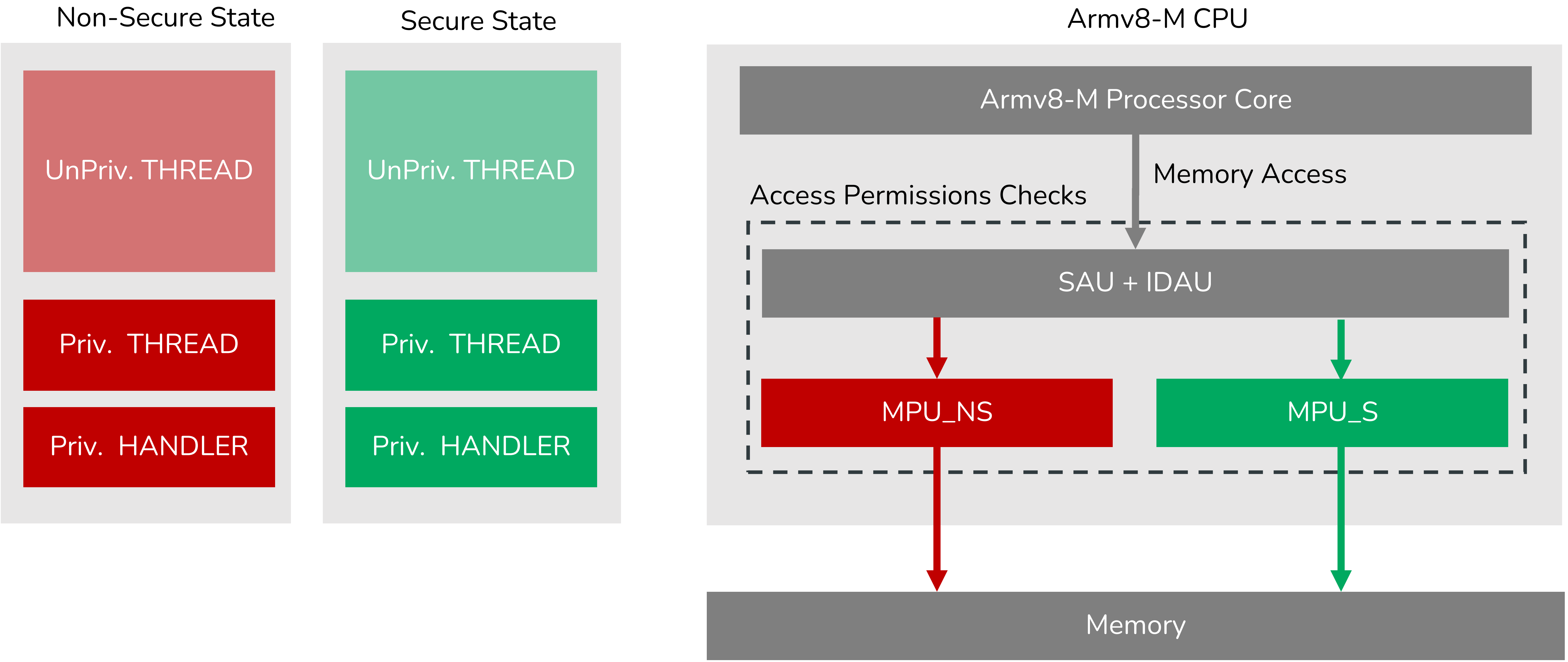
Armv8-M TrustZone



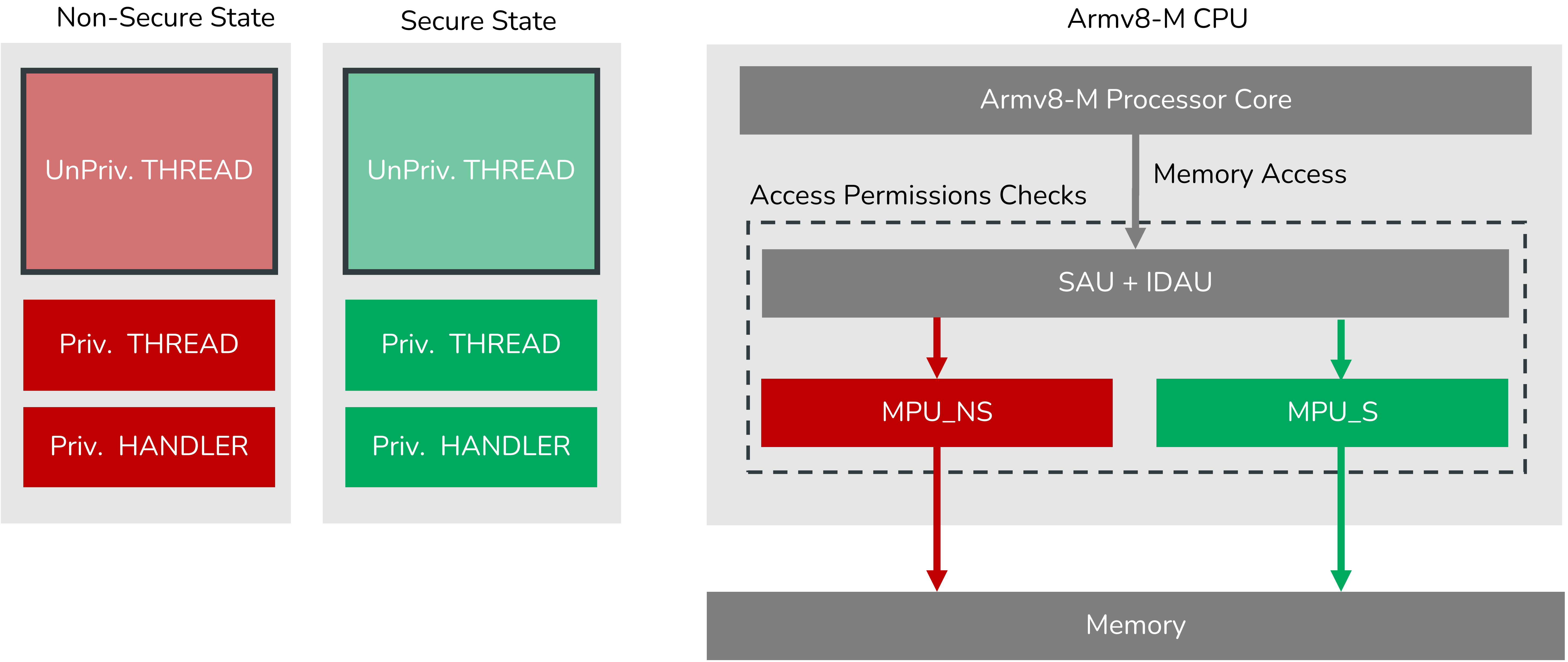
Armv8-M TrustZone



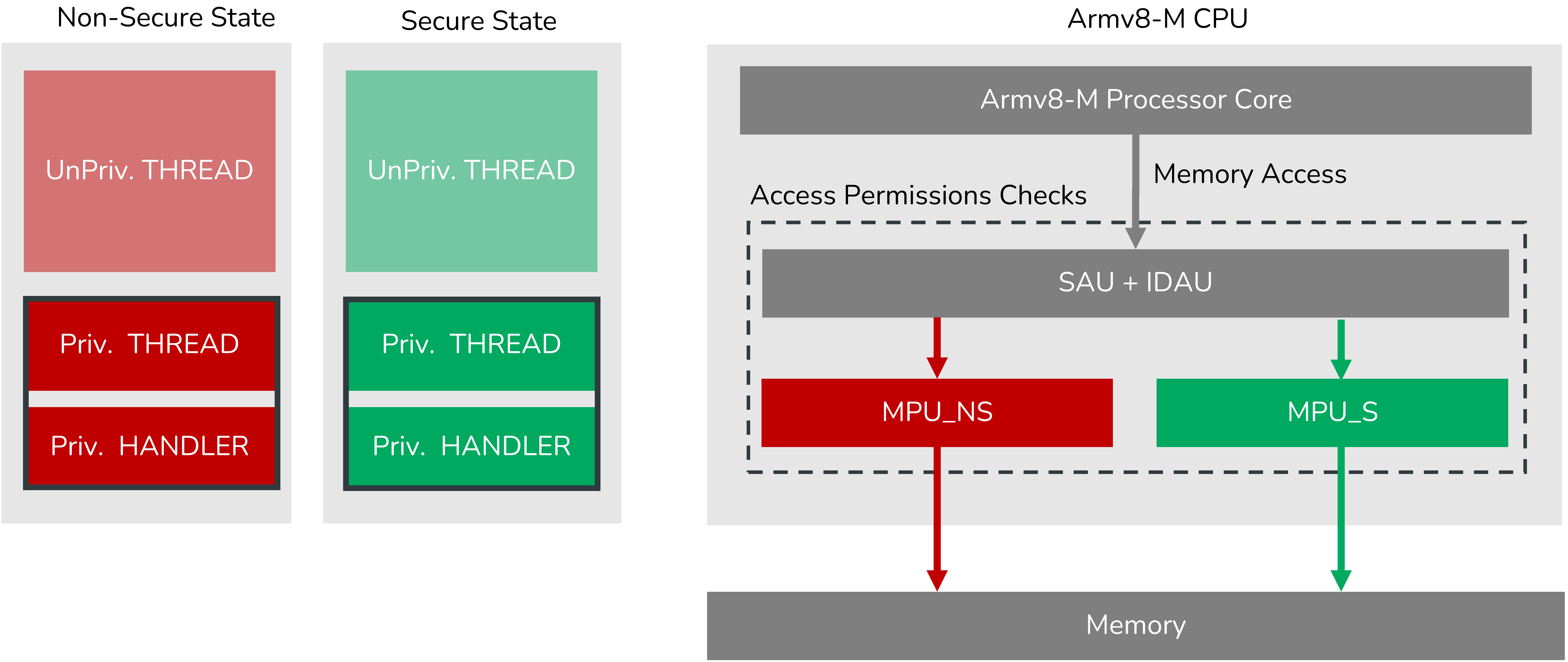
Armv8-M TrustZone



Armv8-M TrustZone



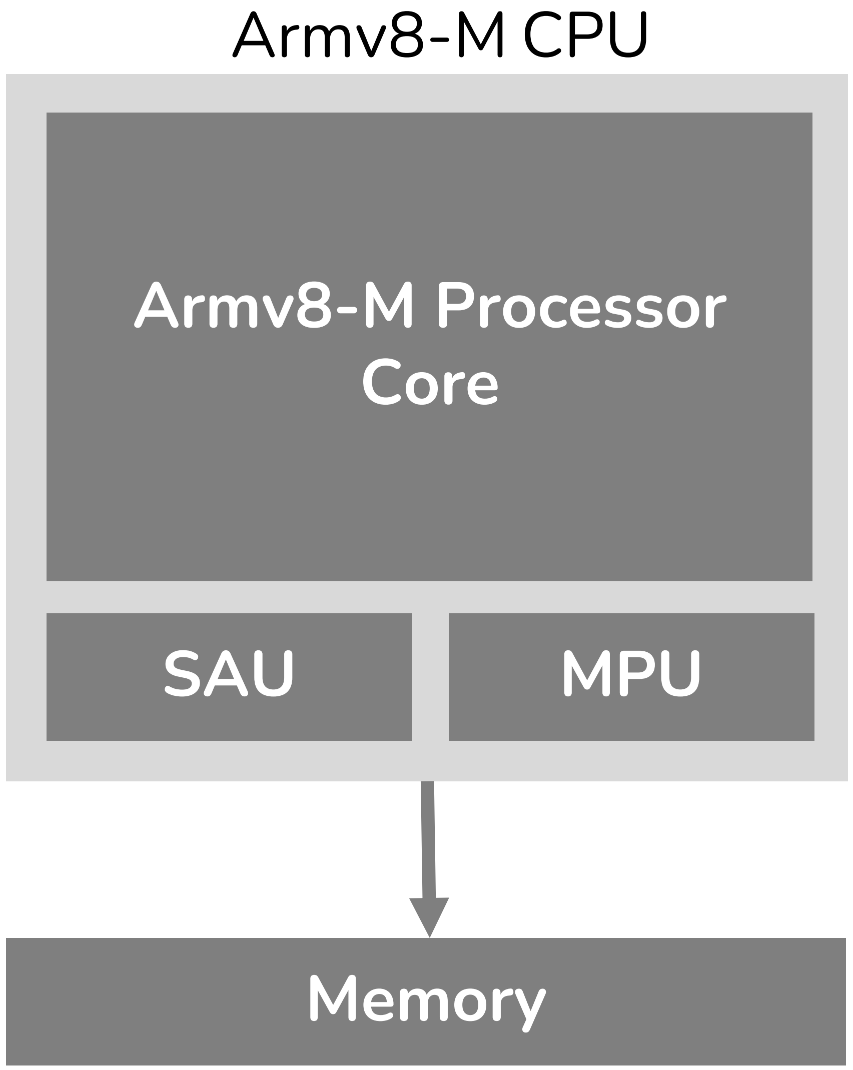
Armv8-M TrustZone



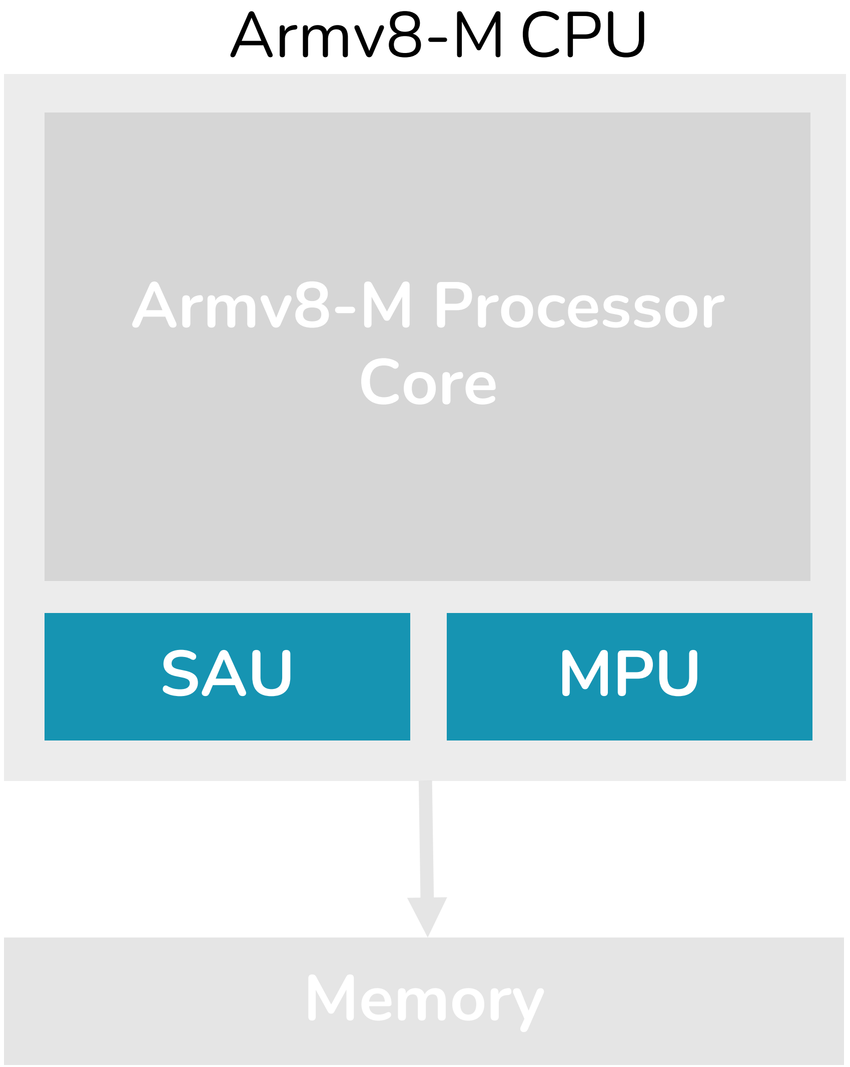
CPU Protection vs System Protection

CPU Protection vs System Protection

CPU Protection vs System Protection

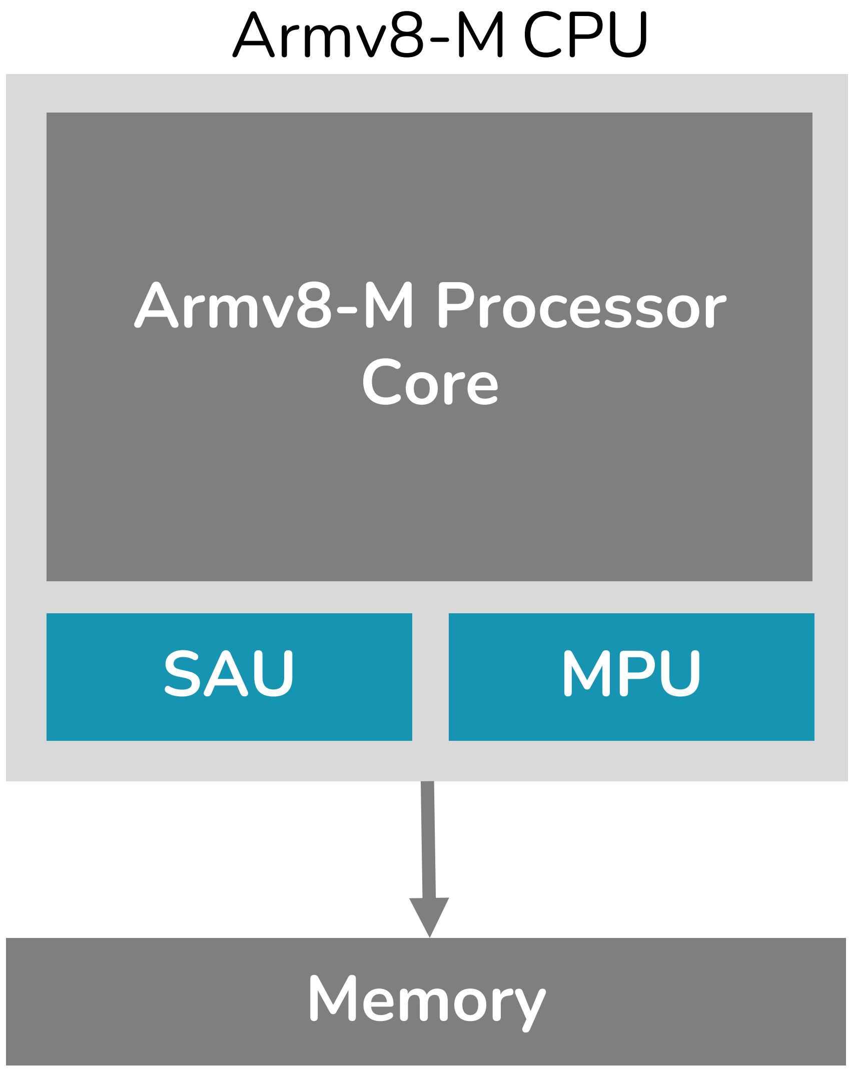


CPU Protection vs System Protection

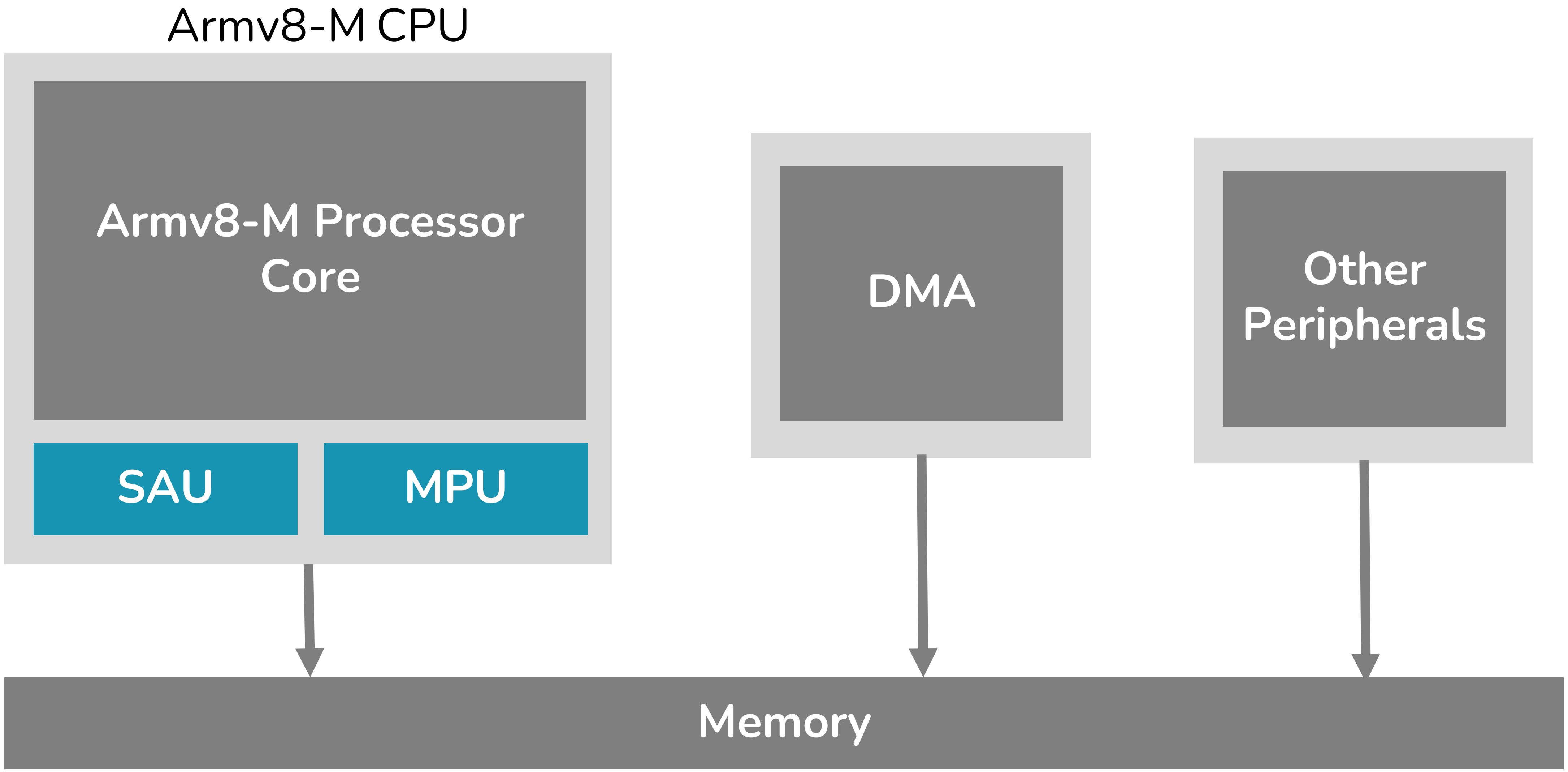


Armv8-M
Memory Protection Controllers

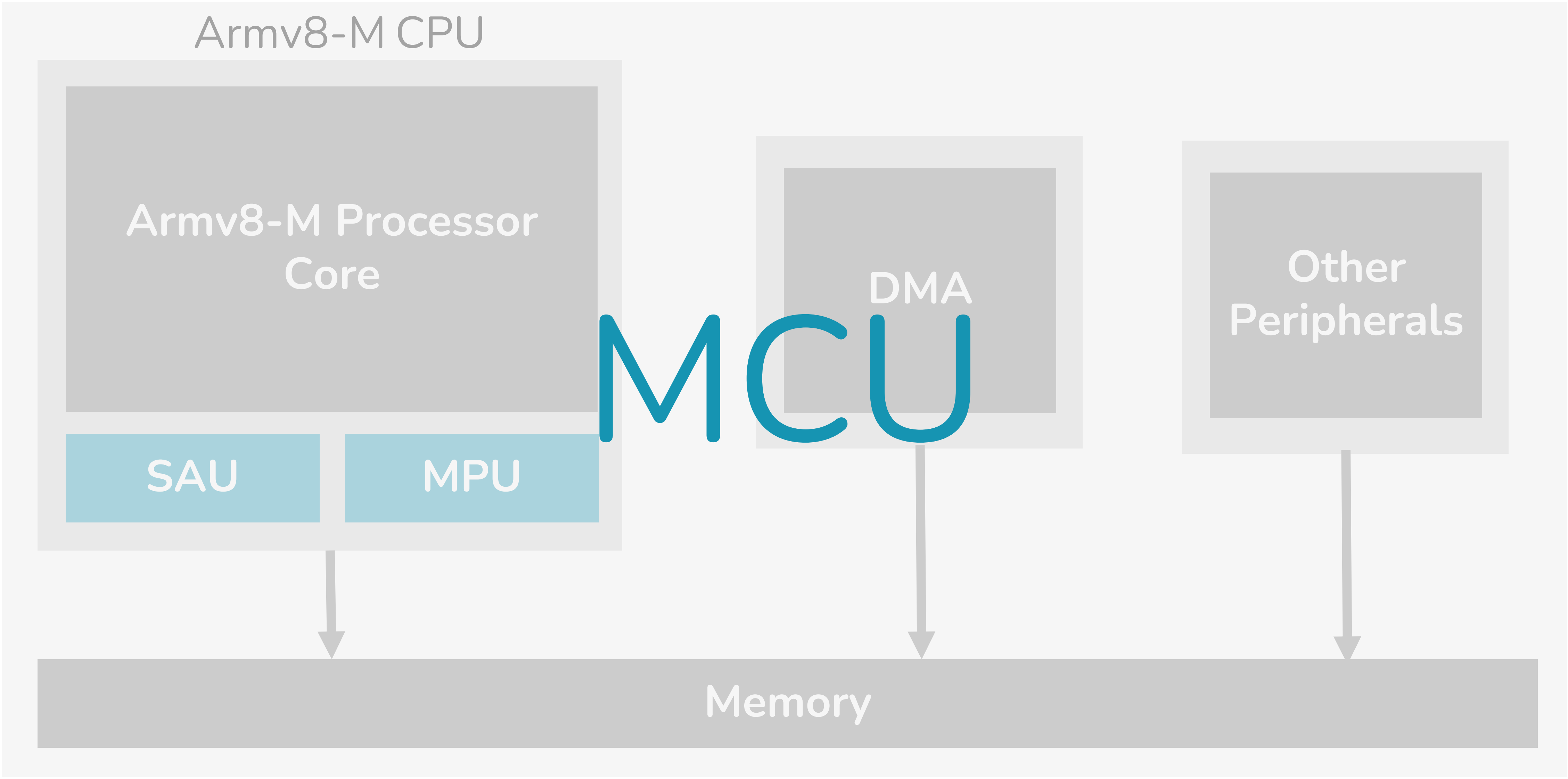
CPU Protection vs System Protection



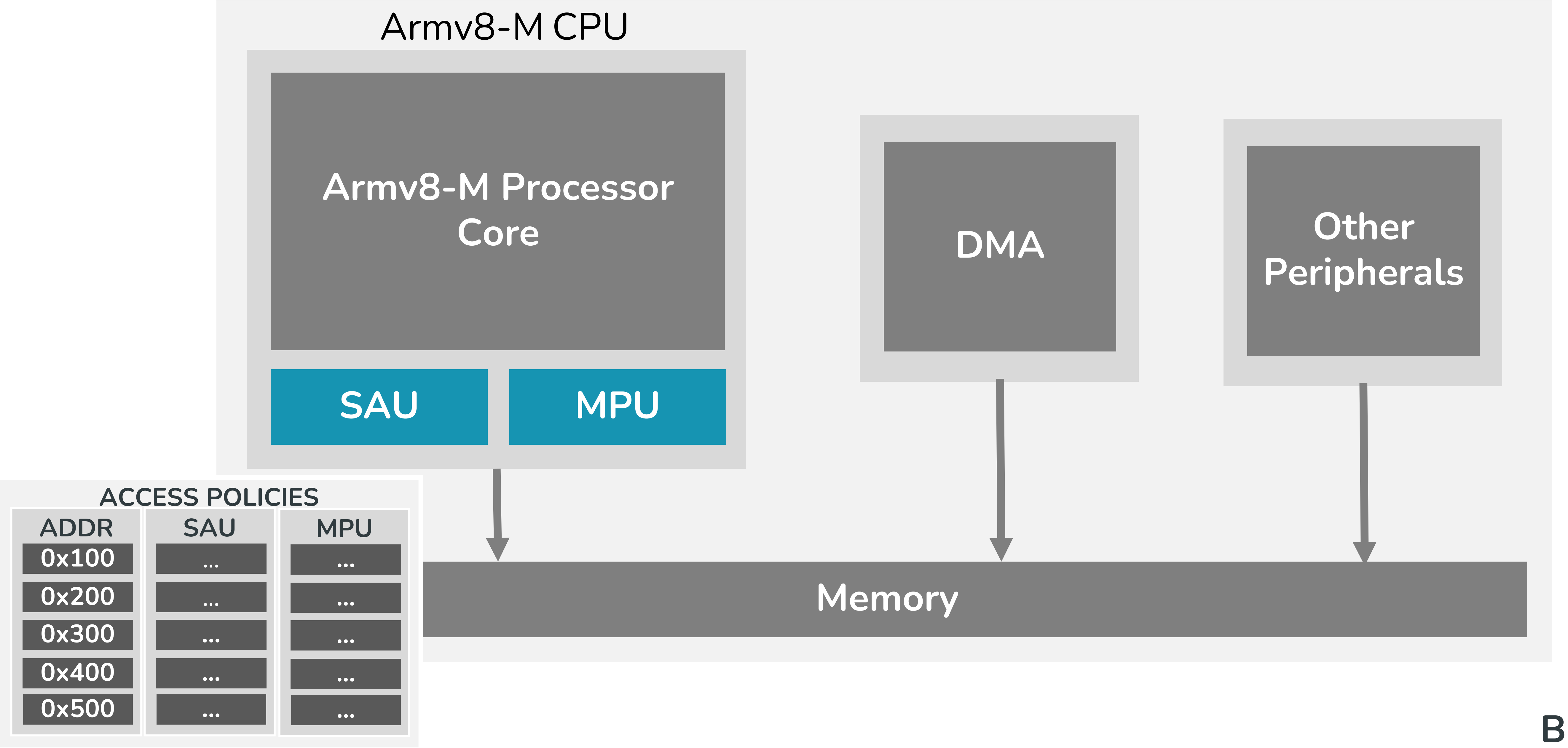
CPU Protection vs System Protection



CPU Protection vs System Protection



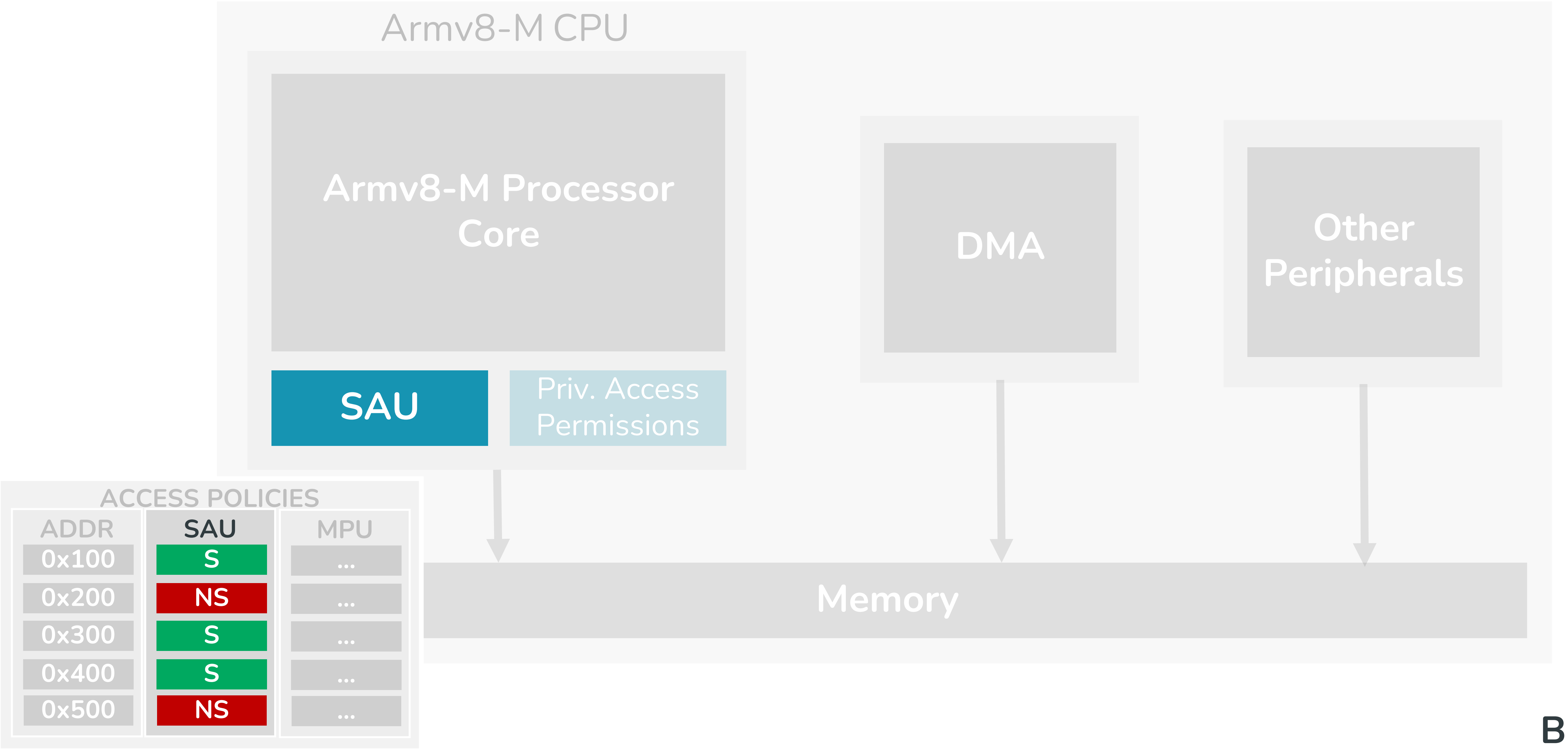
CPU Protection vs System Protection



ACCESS POLICIES

ADDR	SAU	MPU
0x100	...	...
0x200	...	...
0x300	...	...
0x400	...	...
0x500	...	...

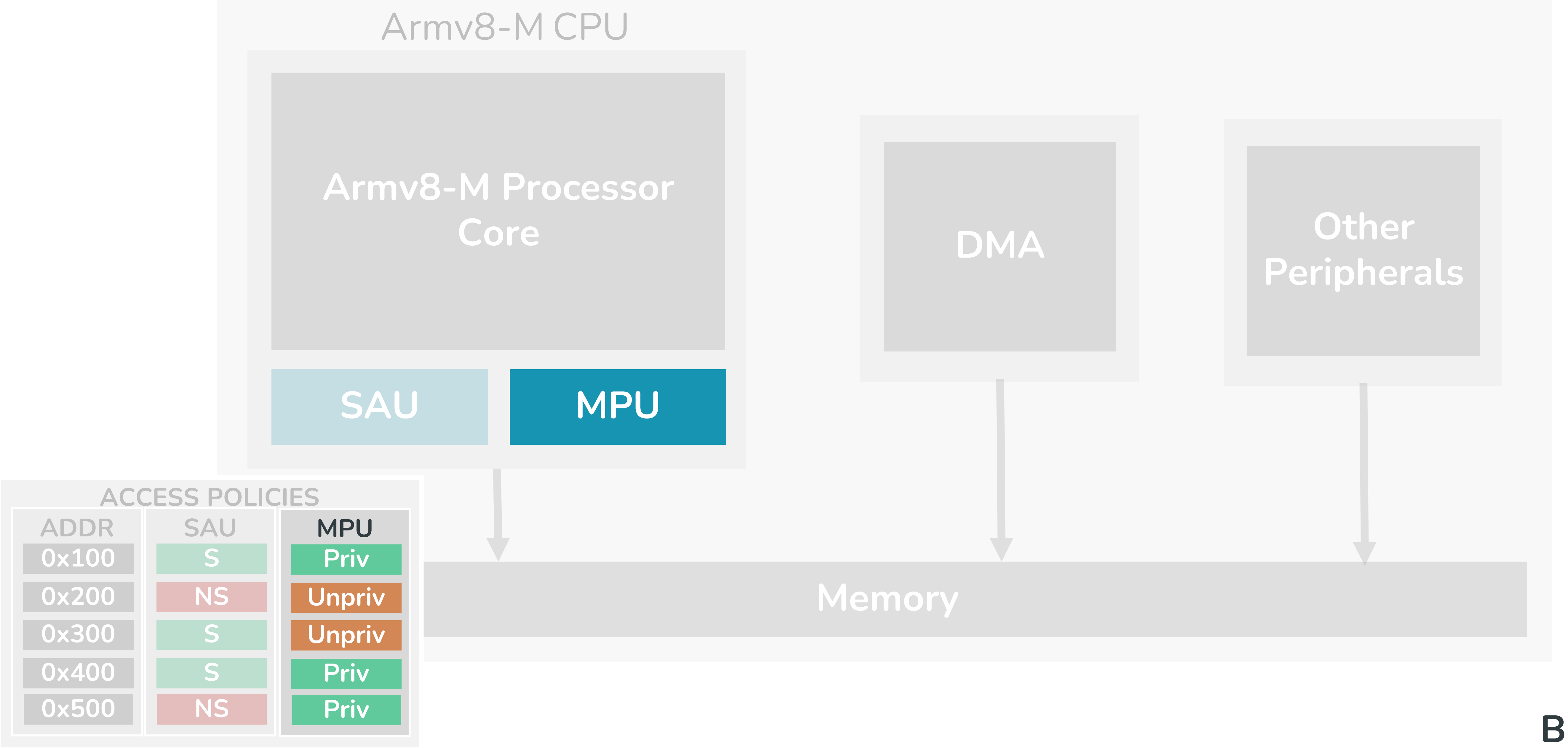
CPU Protection vs System Protection



ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	...
0x200	NS	...
0x300	S	...
0x400	S	...
0x500	NS	...

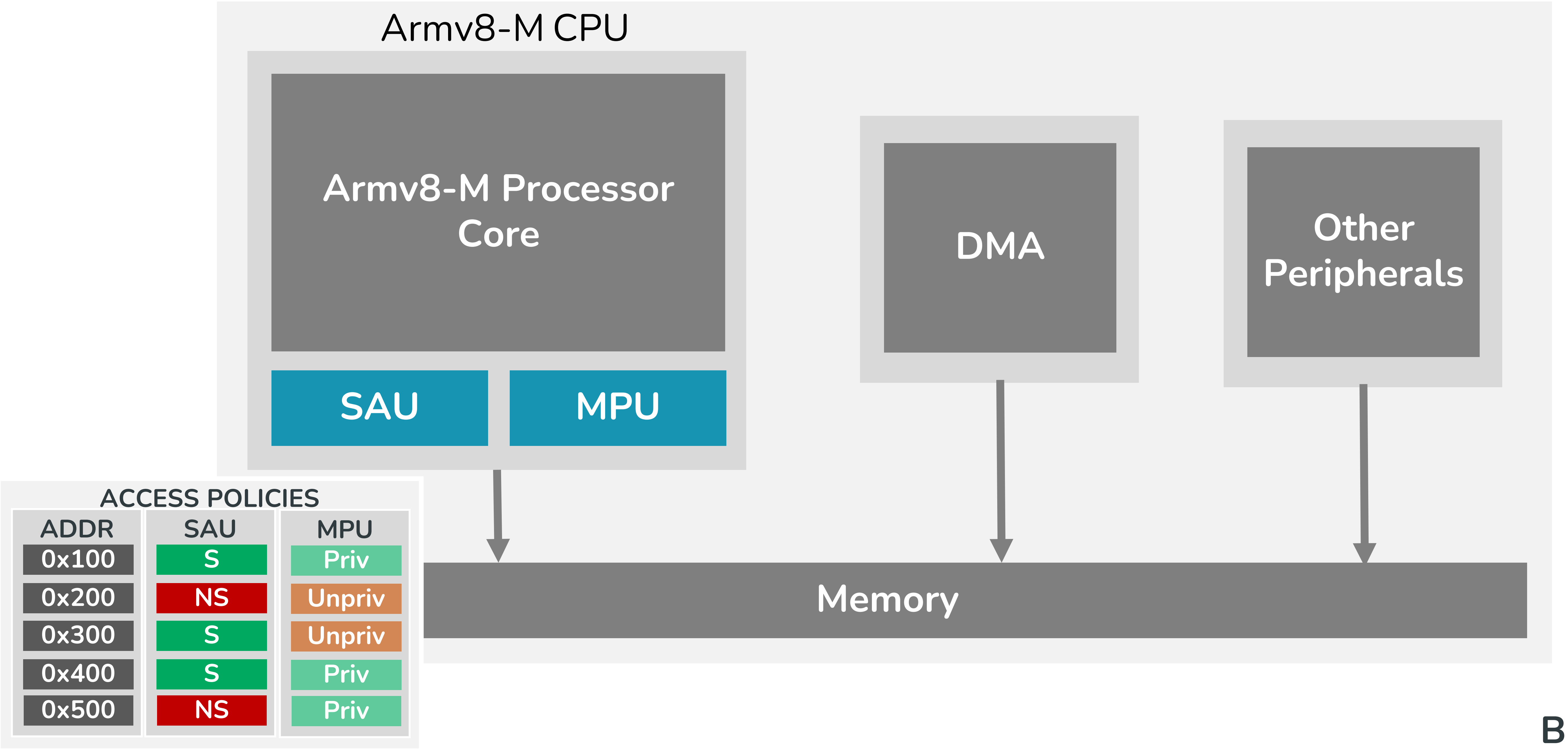
CPU Protection vs System Protection



ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	Priv
0x200	NS	Unpriv
0x300	S	Unpriv
0x400	S	Priv
0x500	NS	Priv

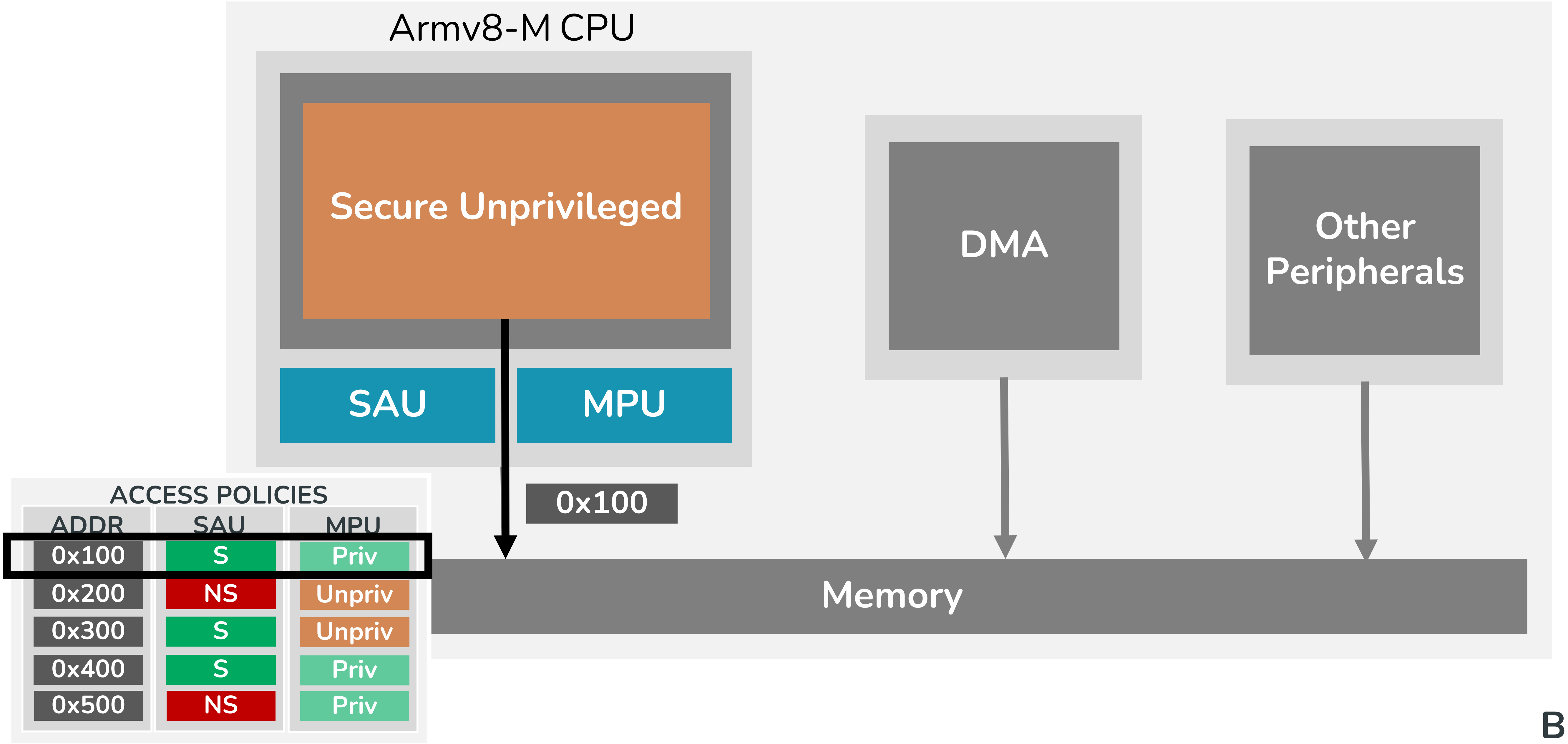
CPU Protection vs System Protection



ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	Priv
0x200	NS	Unpriv
0x300	S	Unpriv
0x400	S	Priv
0x500	NS	Priv

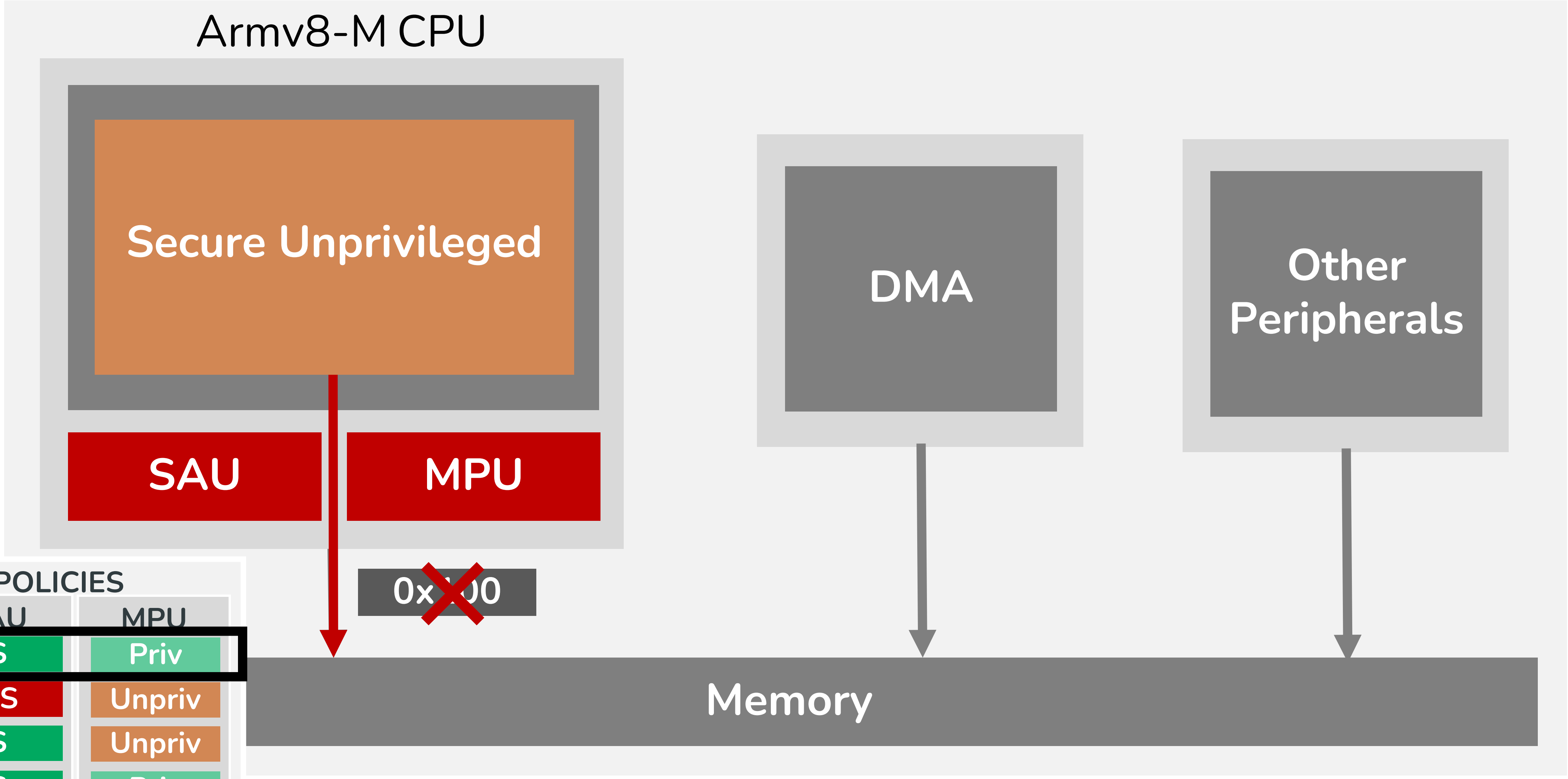
CPU Protection vs System Protection



ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	Priv
0x200	NS	Unpriv
0x300	S	Unpriv
0x400	S	Priv
0x500	NS	Priv

CPU Protection vs System Protection

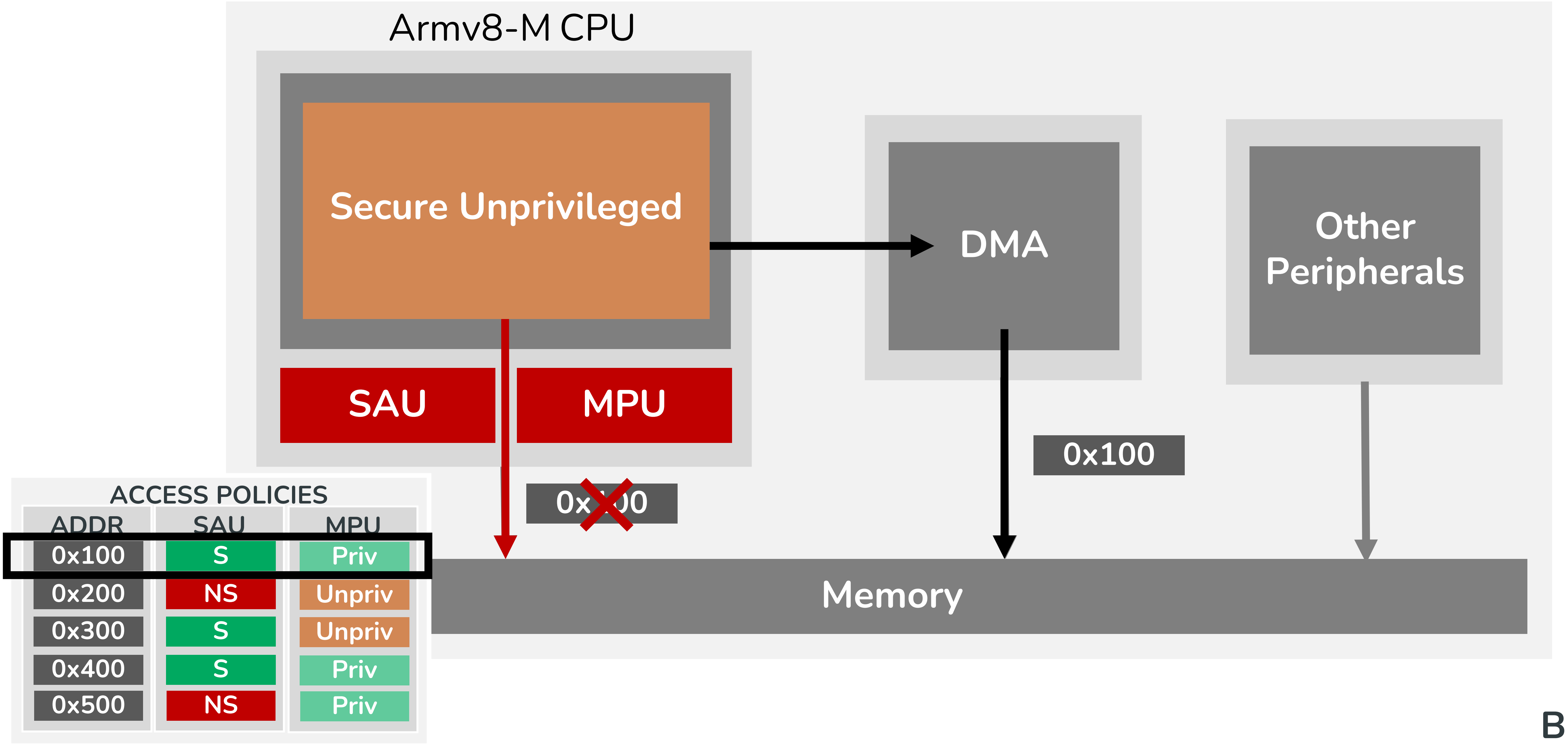


ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	Priv
0x200	NS	Unpriv
0x300	S	Unpriv
0x400	S	Priv
0x500	NS	Priv

BLACKHAT24

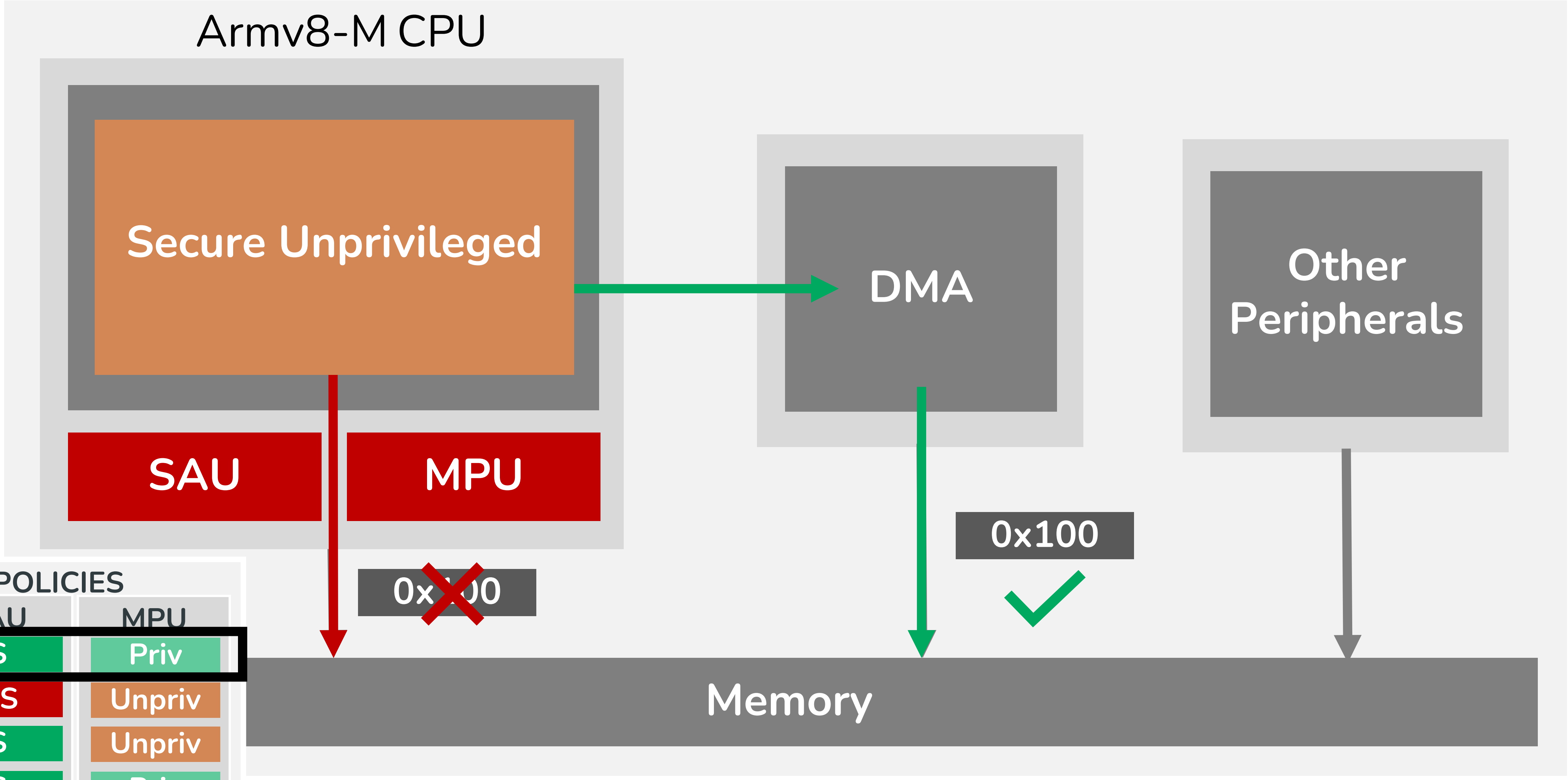
CPU Protection vs System Protection



ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	Priv
0x200	NS	Unpriv
0x300	S	Unpriv
0x400	S	Priv
0x500	NS	Priv

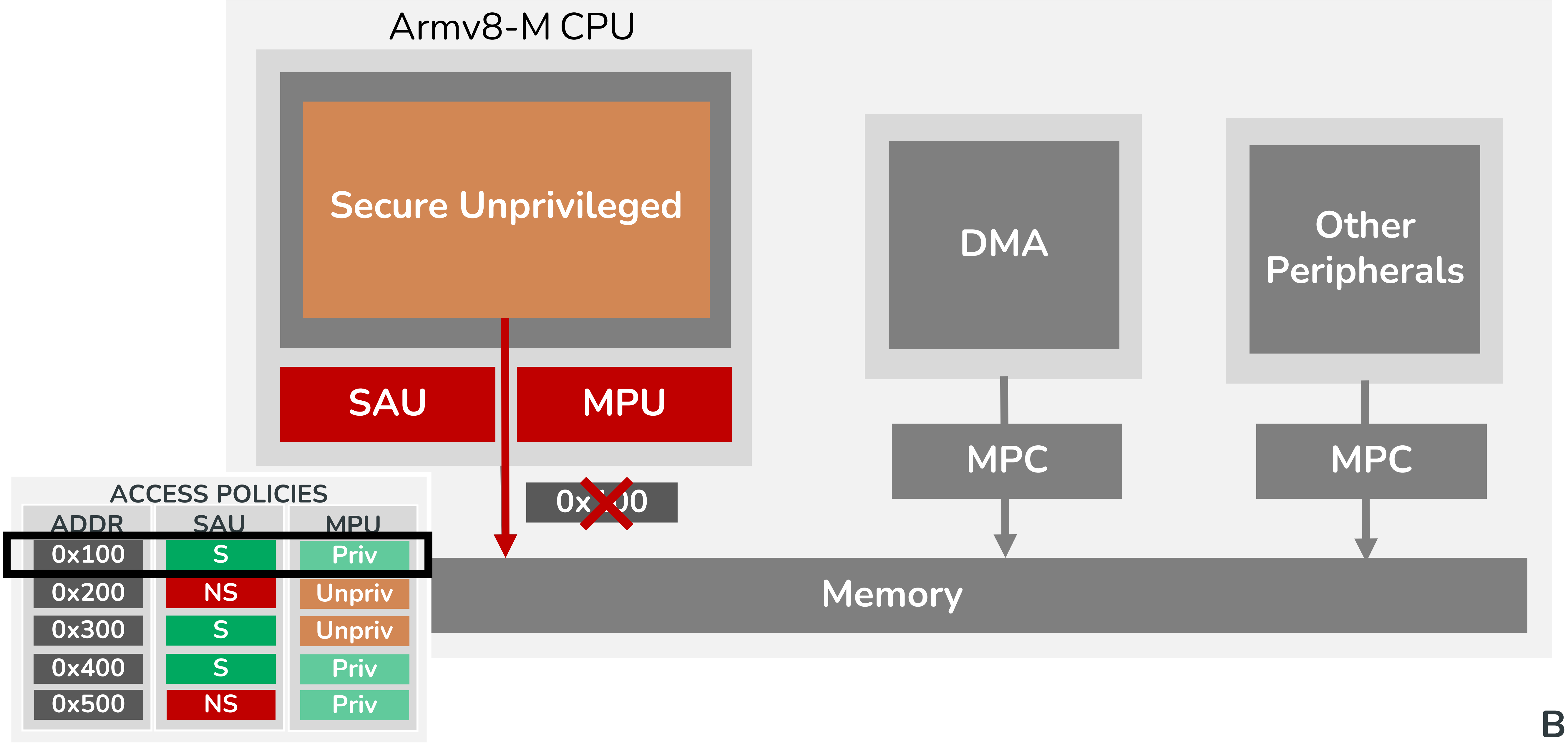
CPU Protection vs System Protection



ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	Priv
0x200	NS	Unpriv
0x300	S	Unpriv
0x400	S	Priv
0x500	NS	Priv

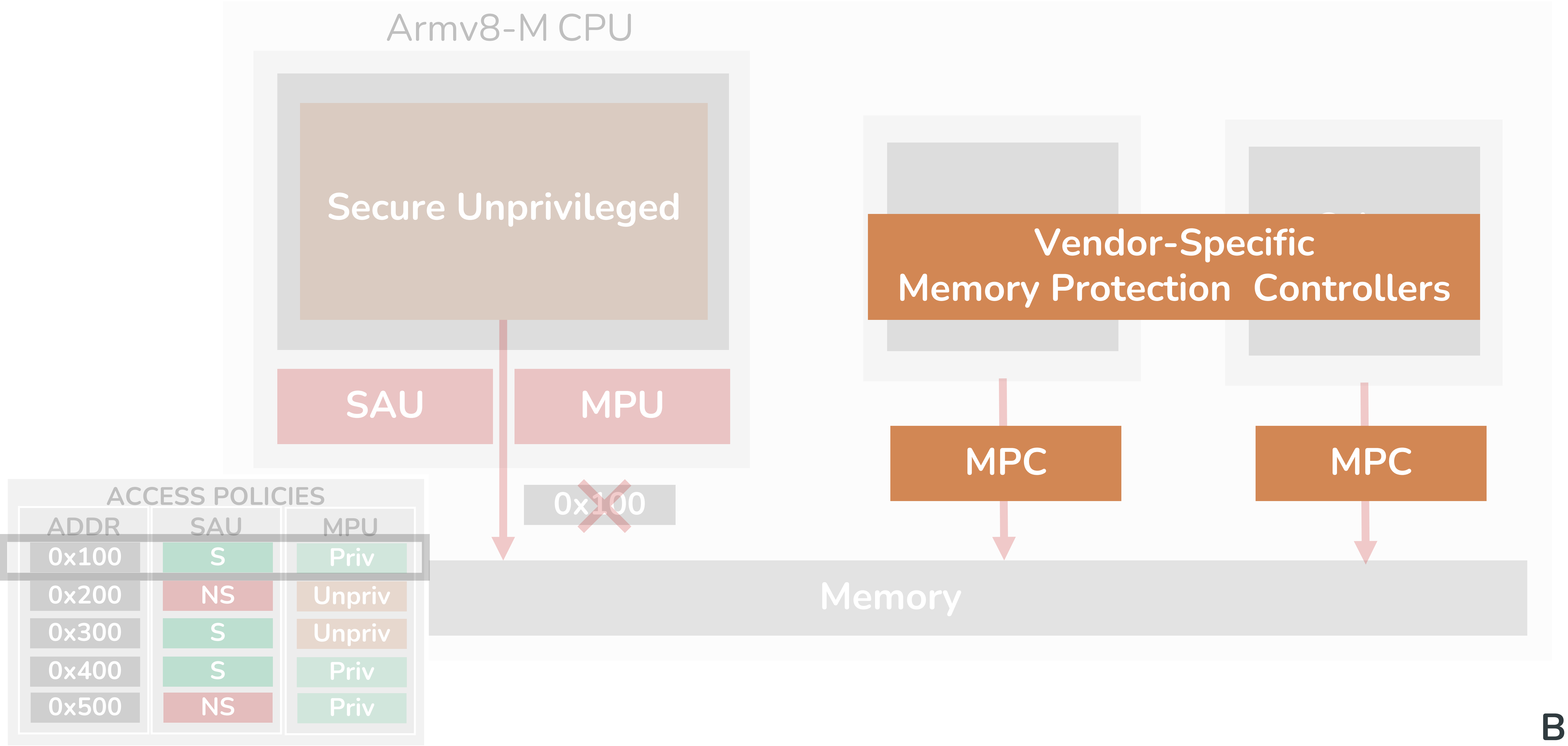
CPU Protection vs System Protection



ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	Priv
0x200	NS	Unpriv
0x300	S	Unpriv
0x400	S	Priv
0x500	NS	Priv

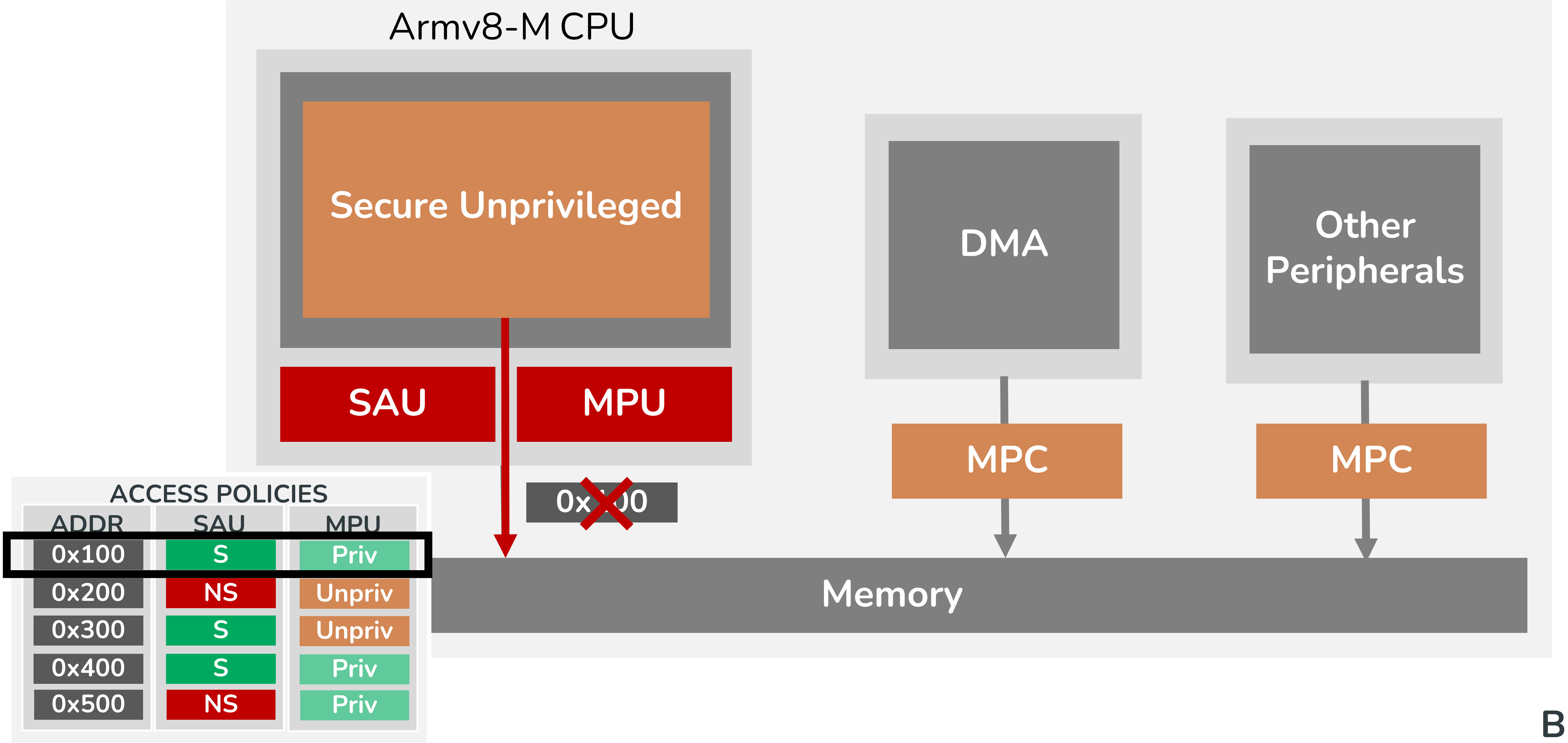
CPU Protection vs System Protection



ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	Priv
0x200	NS	Unpriv
0x300	S	Unpriv
0x400	S	Priv
0x500	NS	Priv

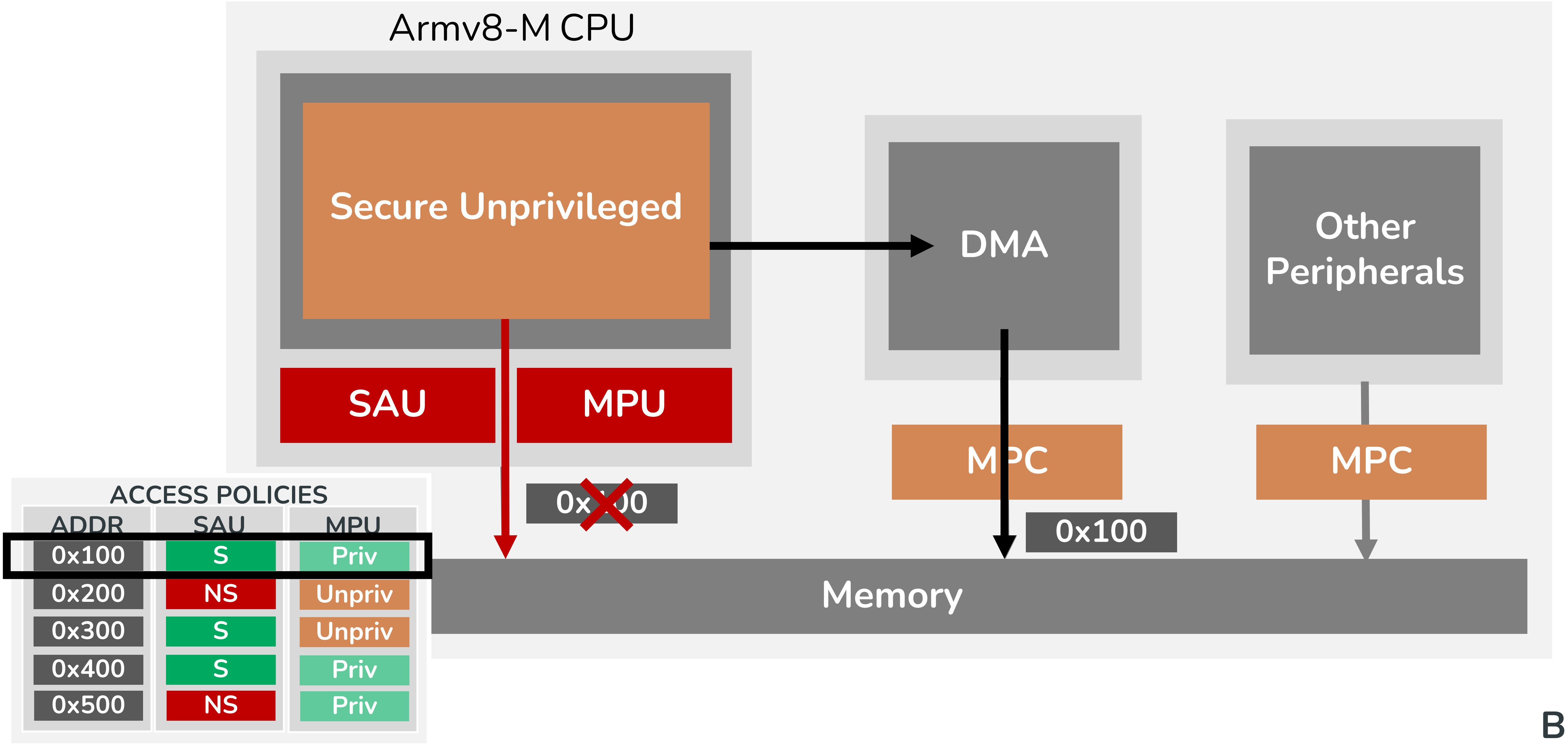
CPU Protection vs System Protection



ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	Priv
0x200	NS	Unpriv
0x300	S	Unpriv
0x400	S	Priv
0x500	NS	Priv

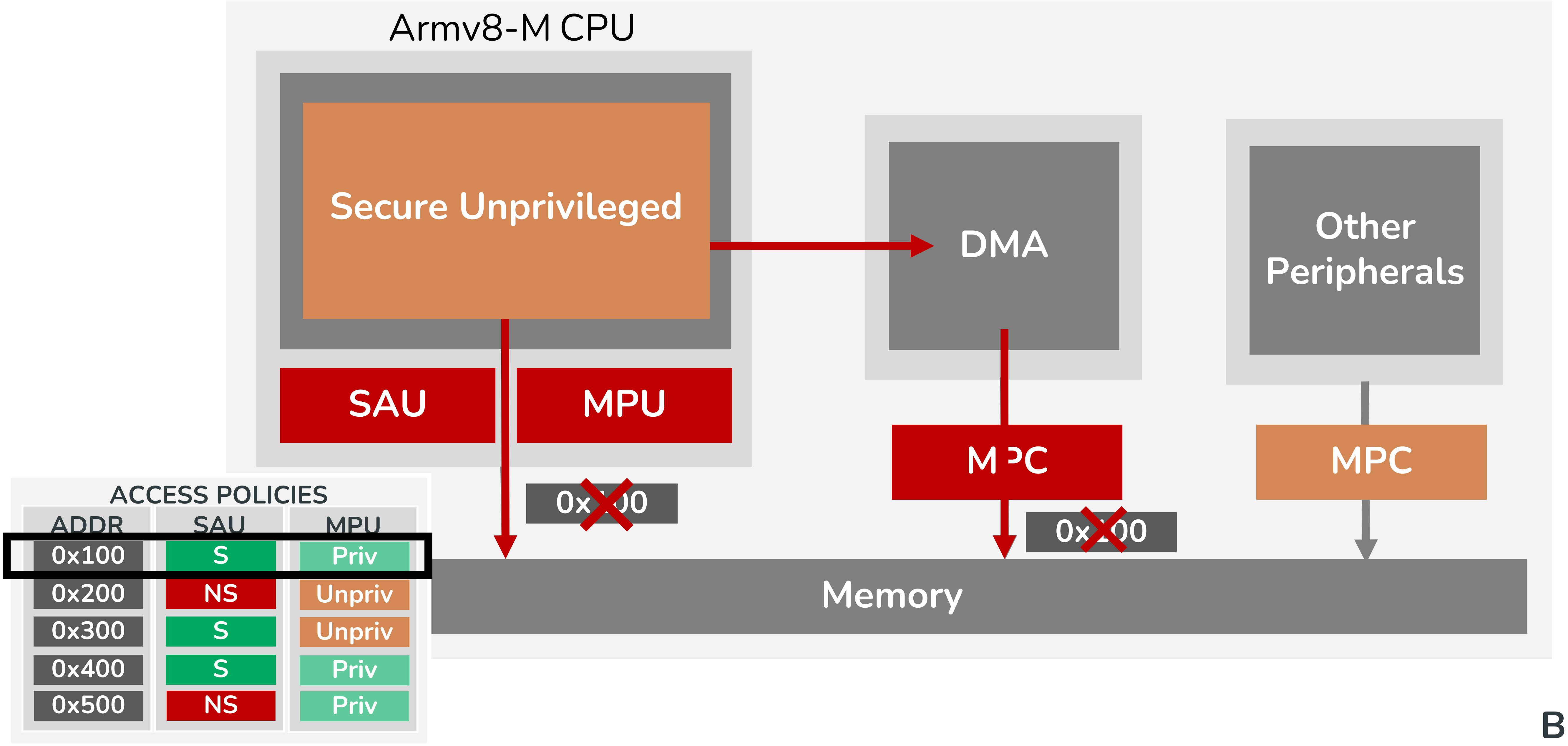
CPU Protection vs System Protection



ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	Priv
0x200	NS	Unpriv
0x300	S	Unpriv
0x400	S	Priv
0x500	NS	Priv

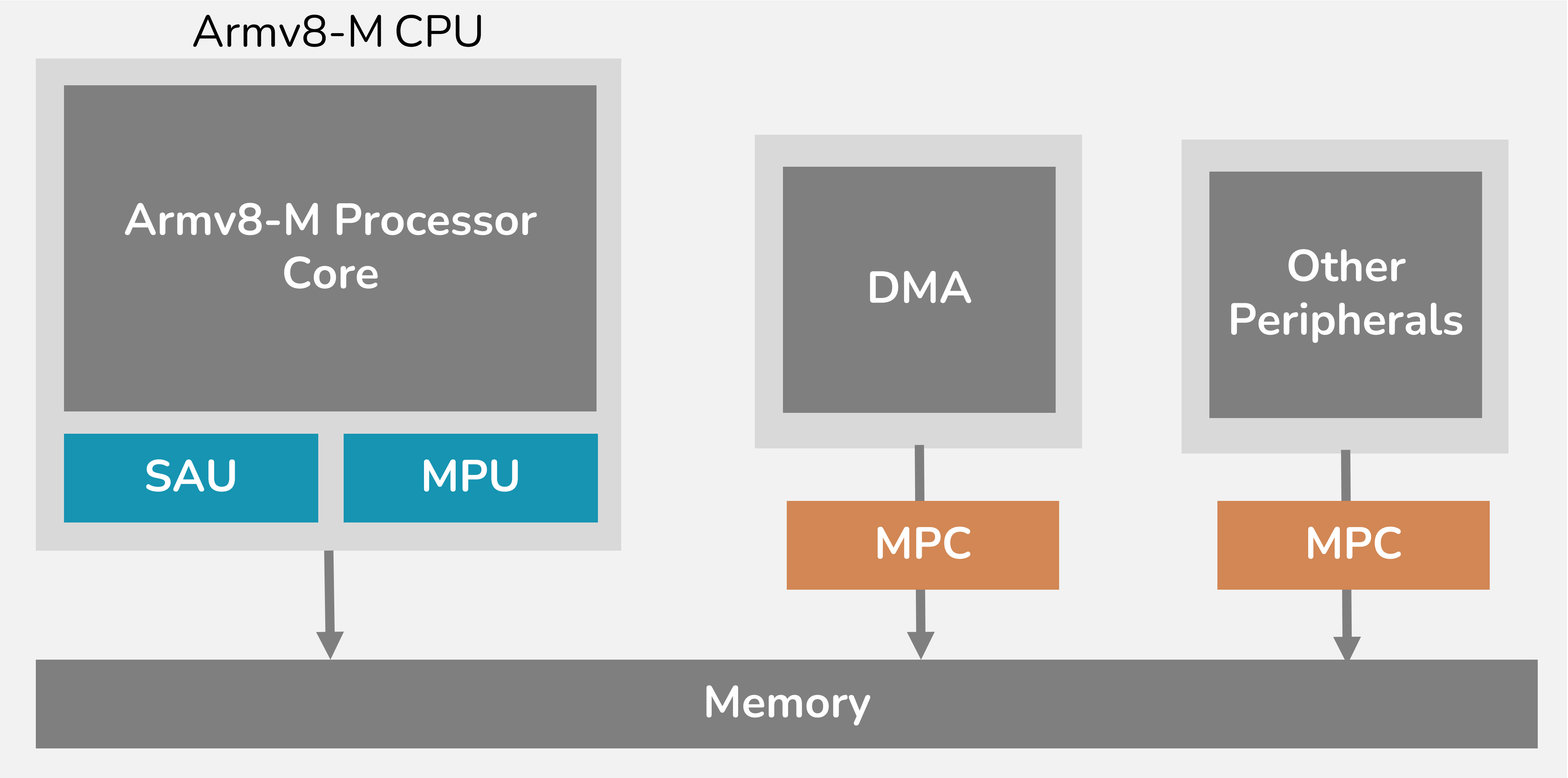
CPU Protection vs System Protection



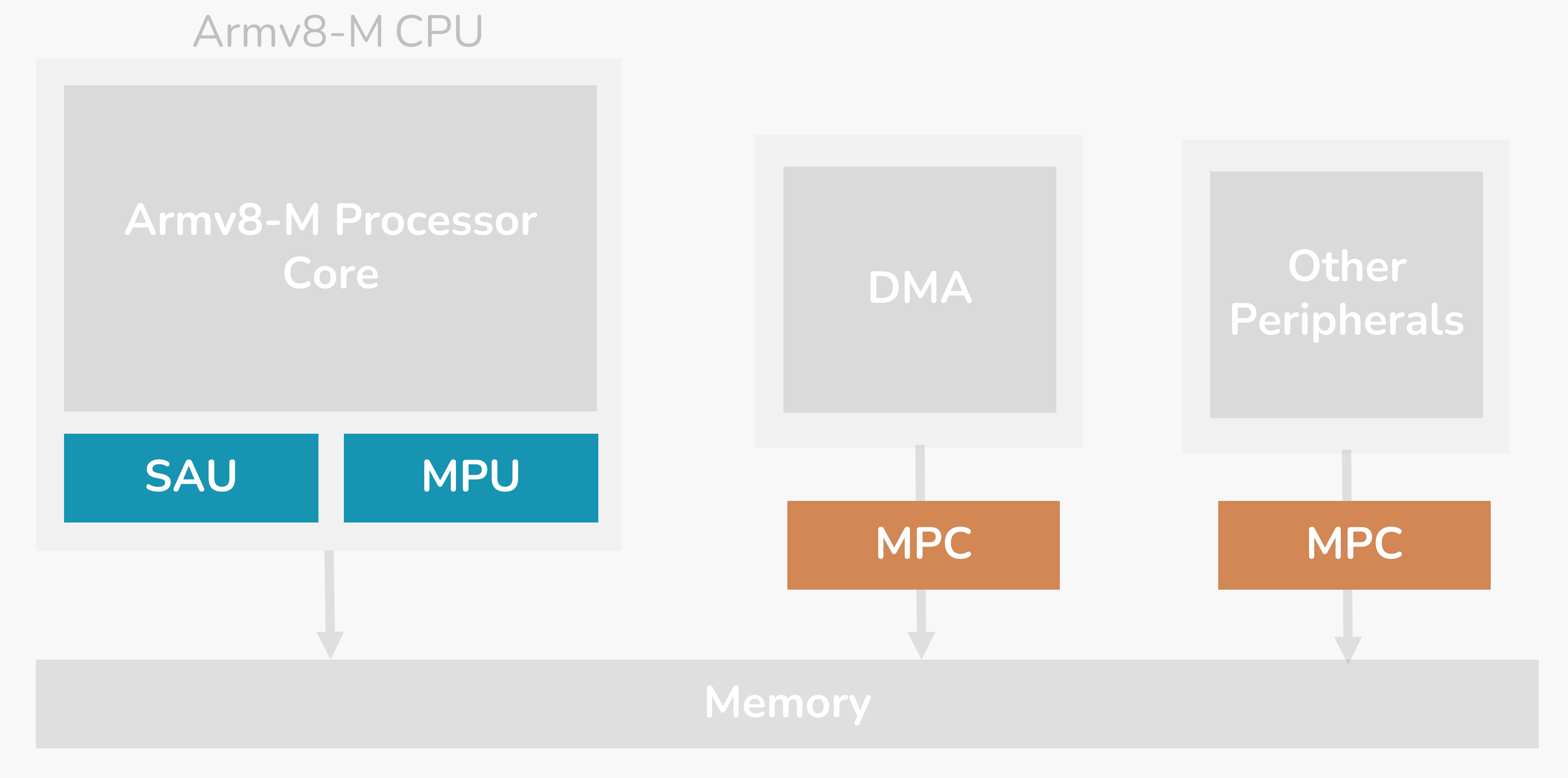
ACCESS POLICIES

ADDR	SAU	MPU
0x100	S	Priv
0x200	NS	Unpriv
0x300	S	Unpriv
0x400	S	Priv
0x500	NS	Priv

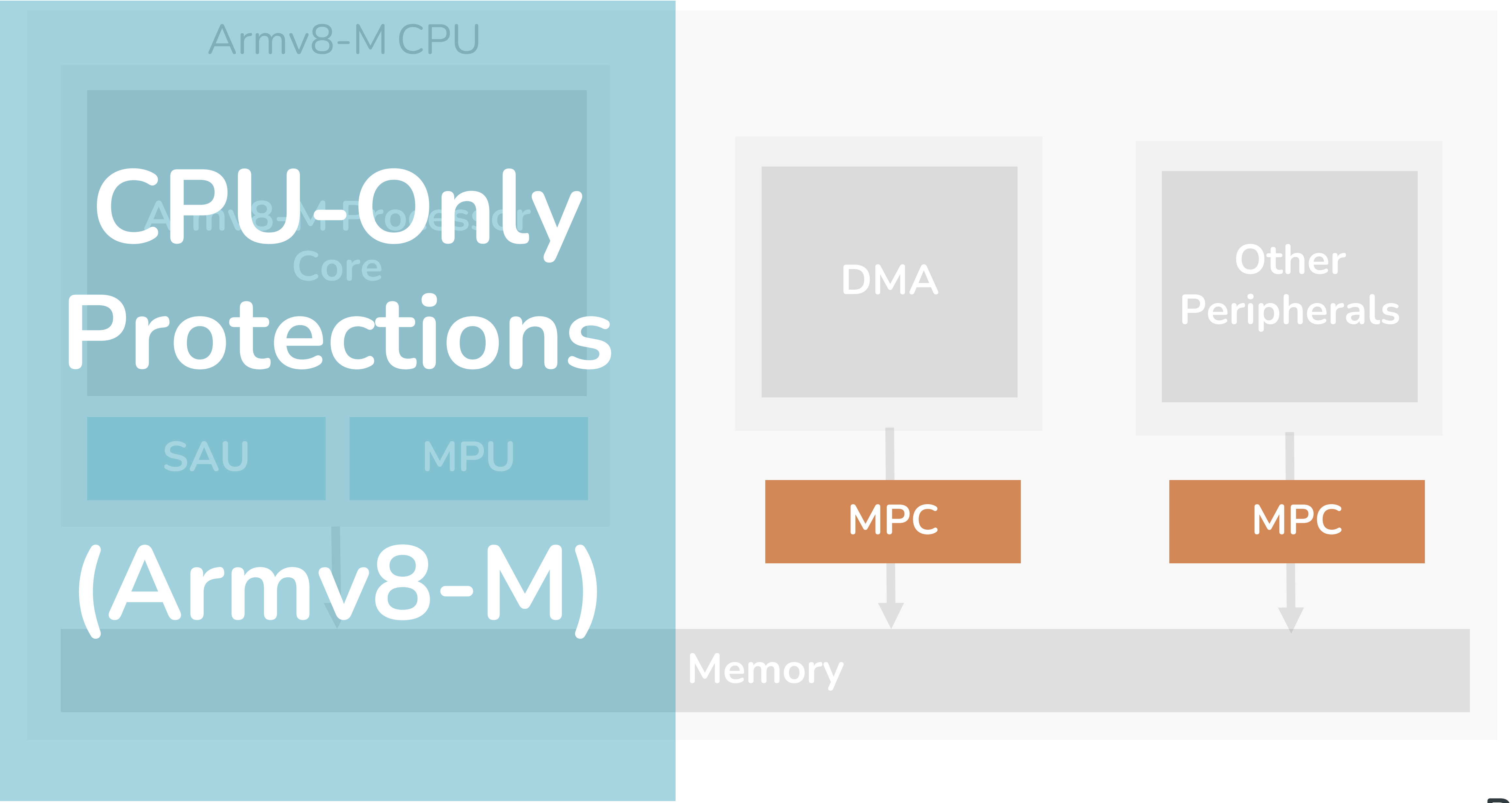
CPU Protection vs System Protection



CPU Protection vs System Protection



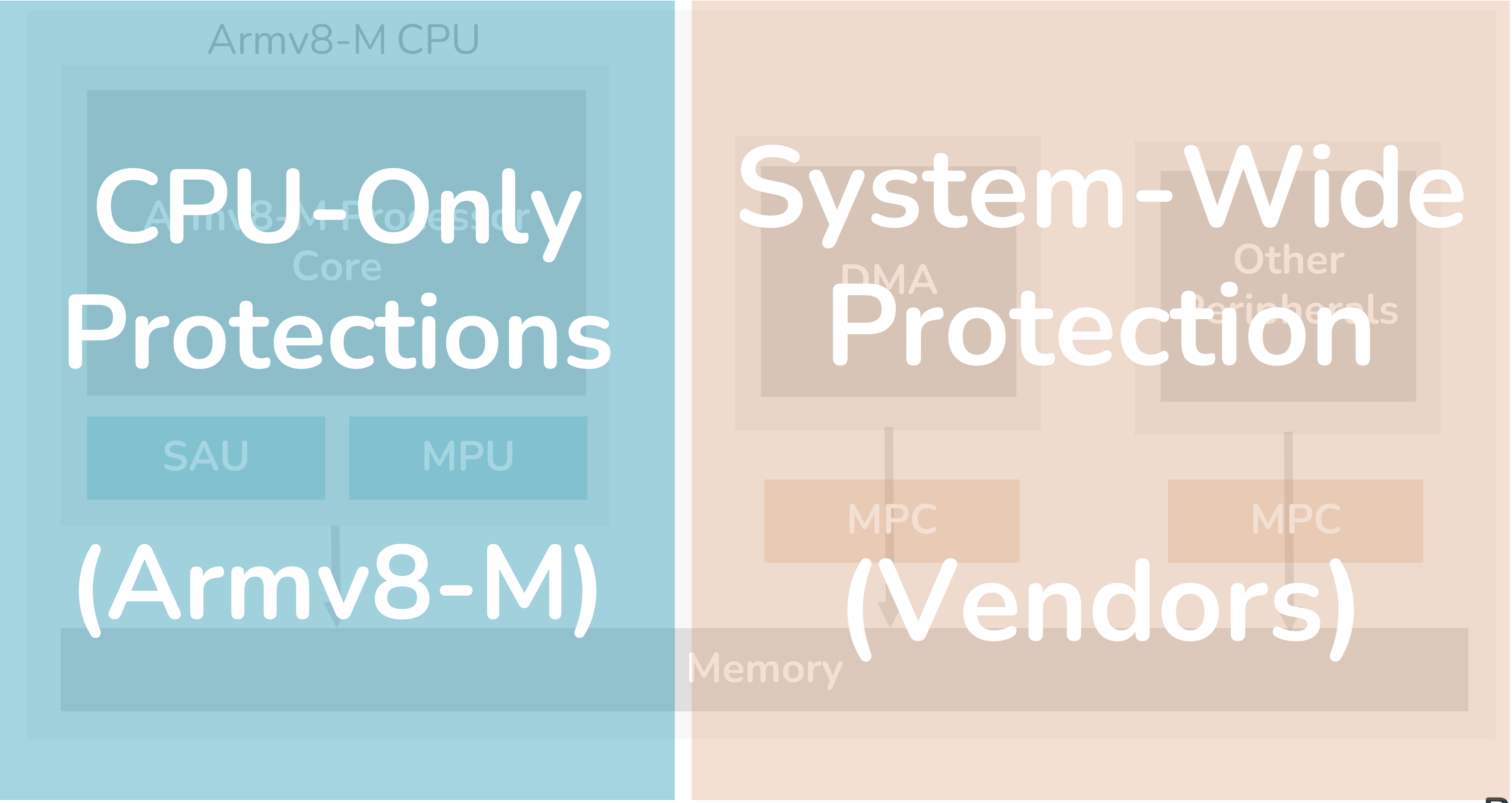
CPU Protection vs System Protection



CPU-Only Protections

(Armv8-M)

CPU Protection vs System Protection

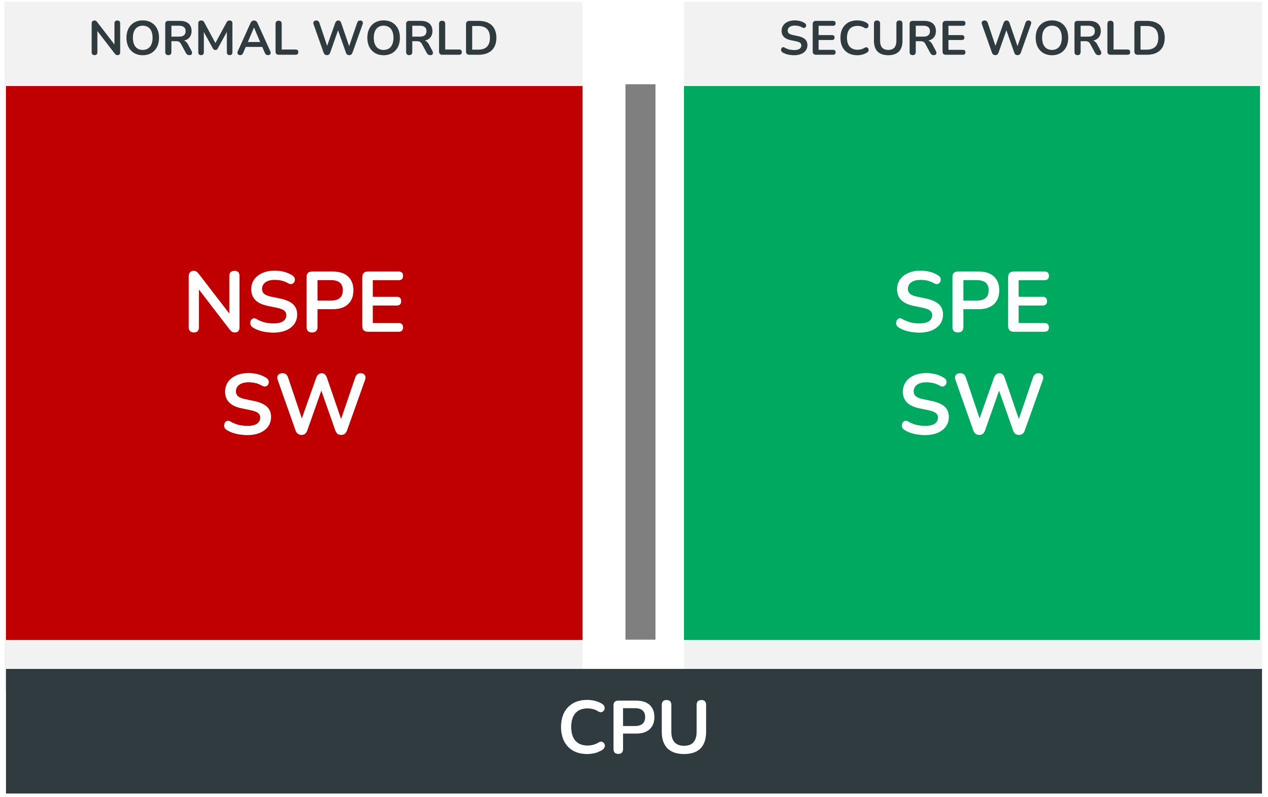


CPU-Only Protections (Armv8-M)

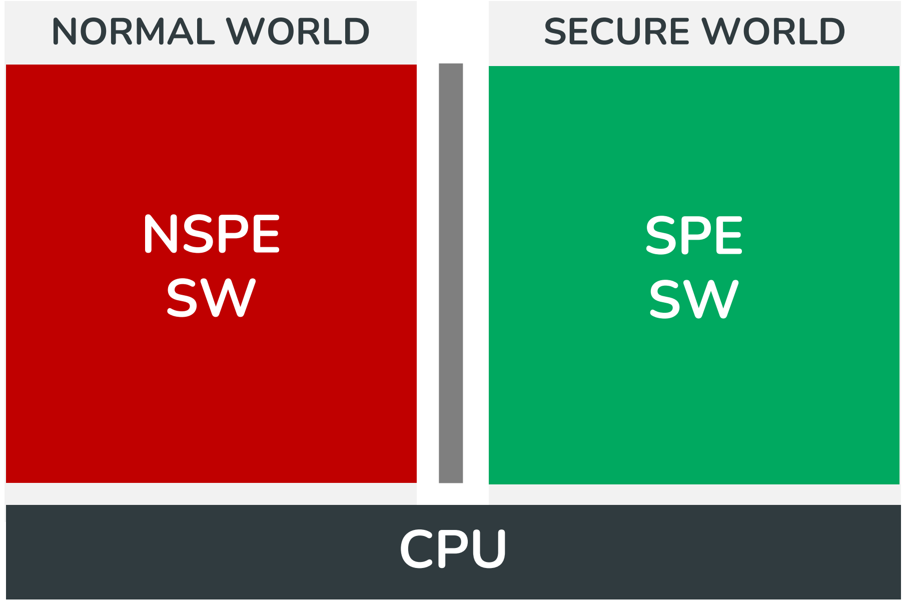
System-Wide Protection (Vendors)

Platform Security Architecture (PSA)

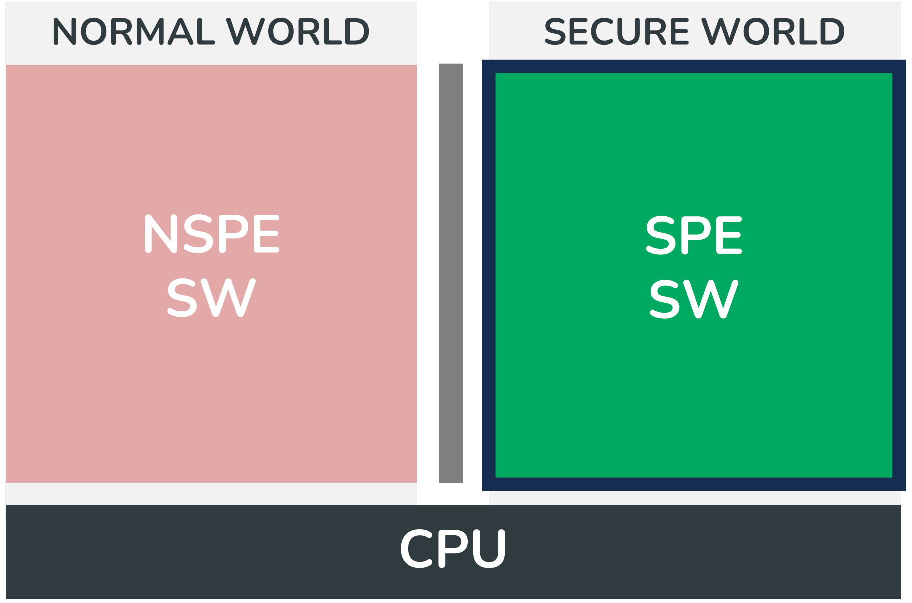
Platform Security Architecture (PSA)



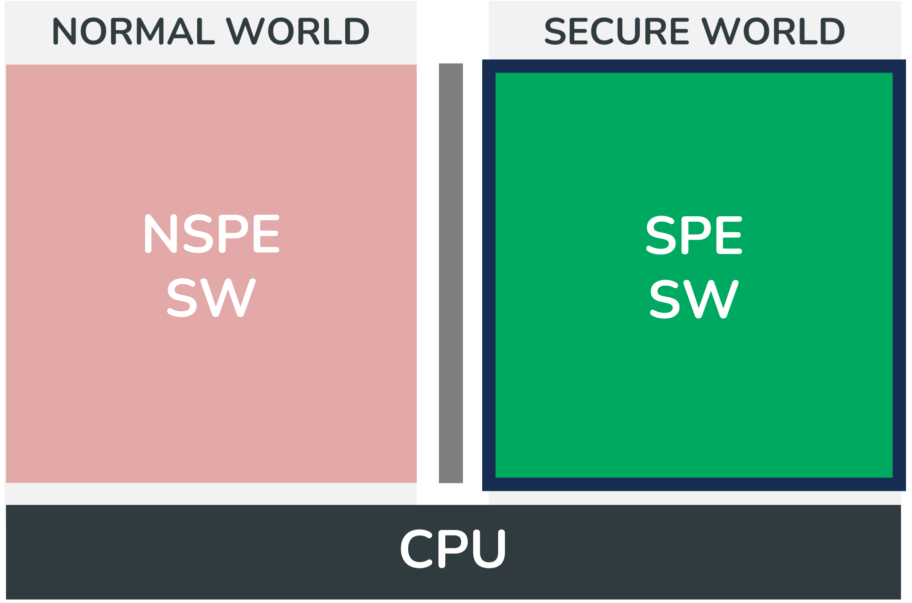
Platform Security Architecture (PSA)



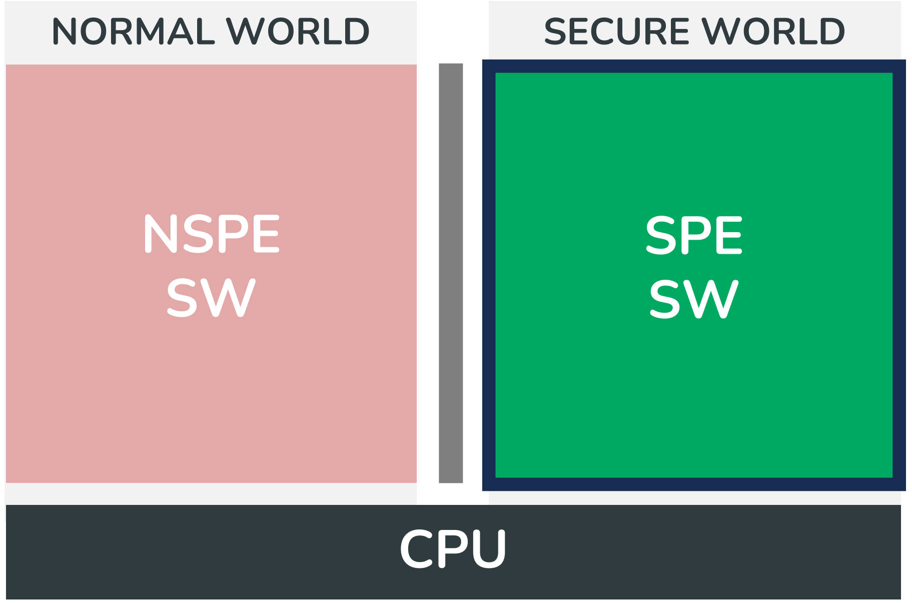
Platform Security Architecture (PSA)

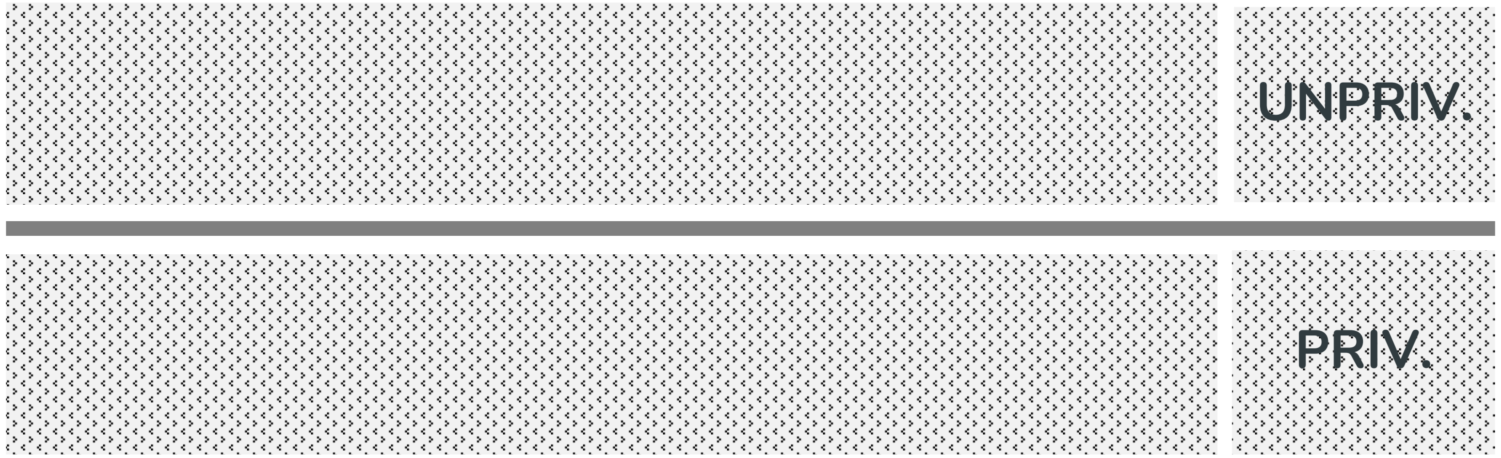


Platform Security Architecture (PSA)

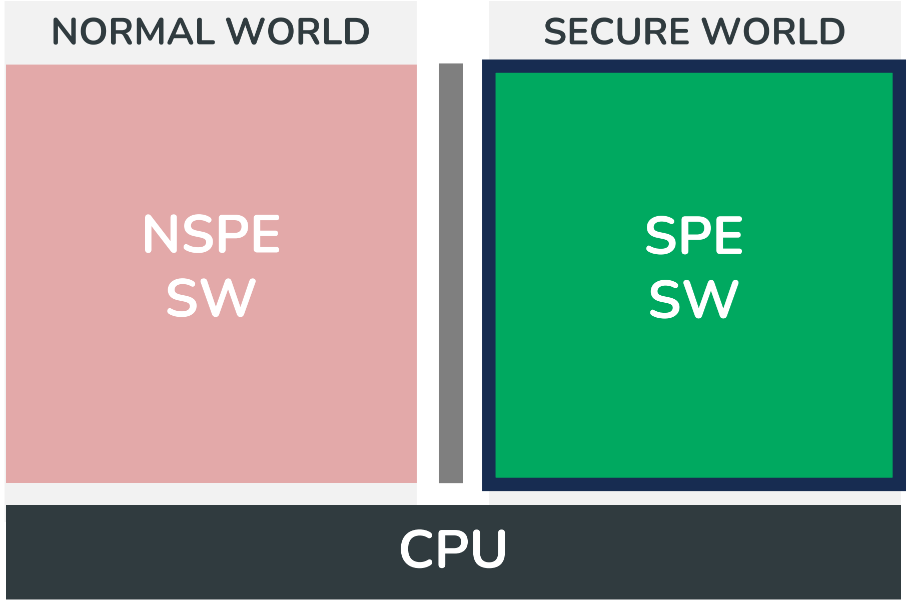


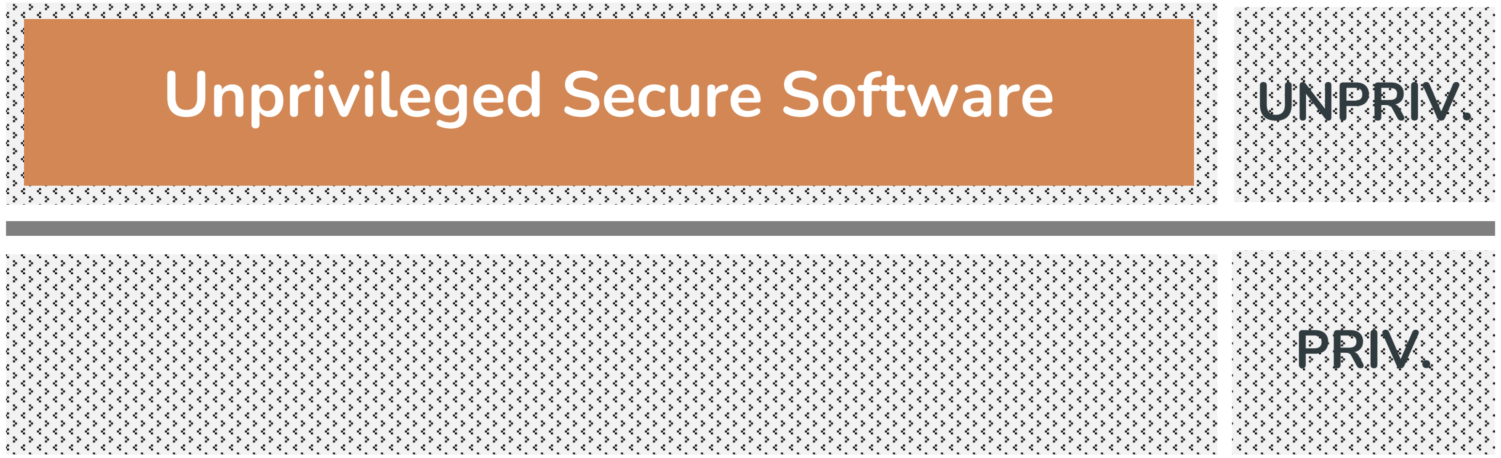
Platform Security Architecture (PSA)



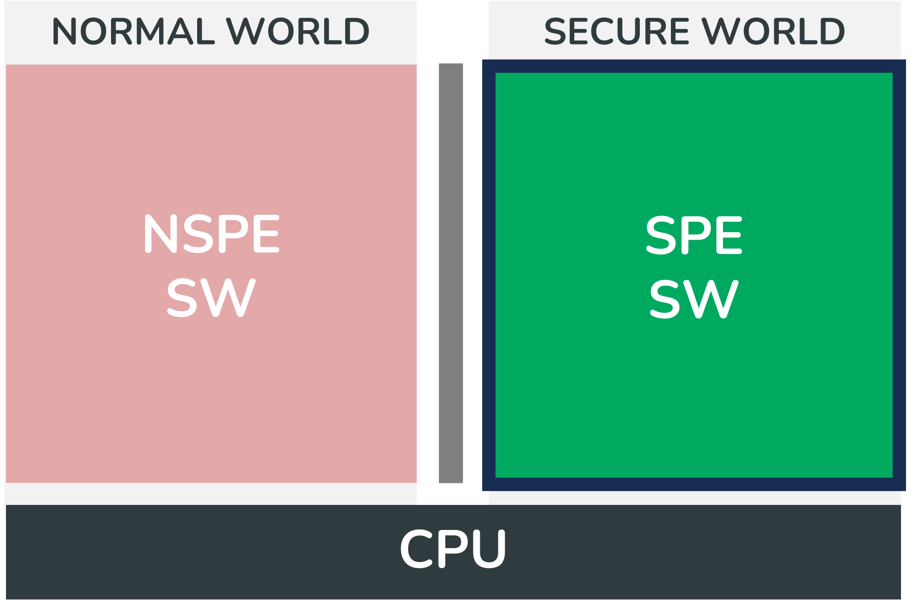


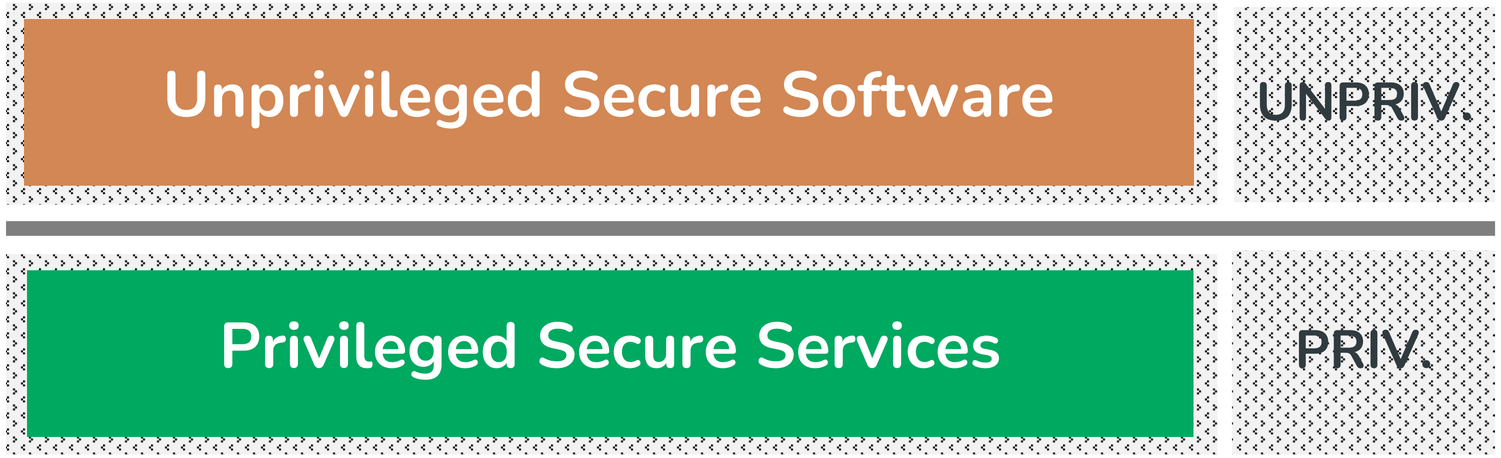
Platform Security Architecture (PSA)



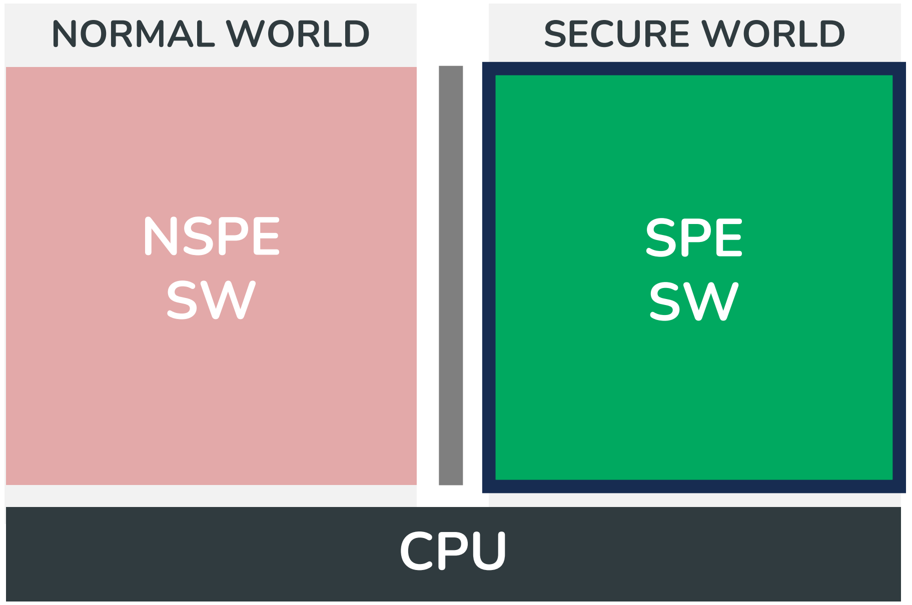


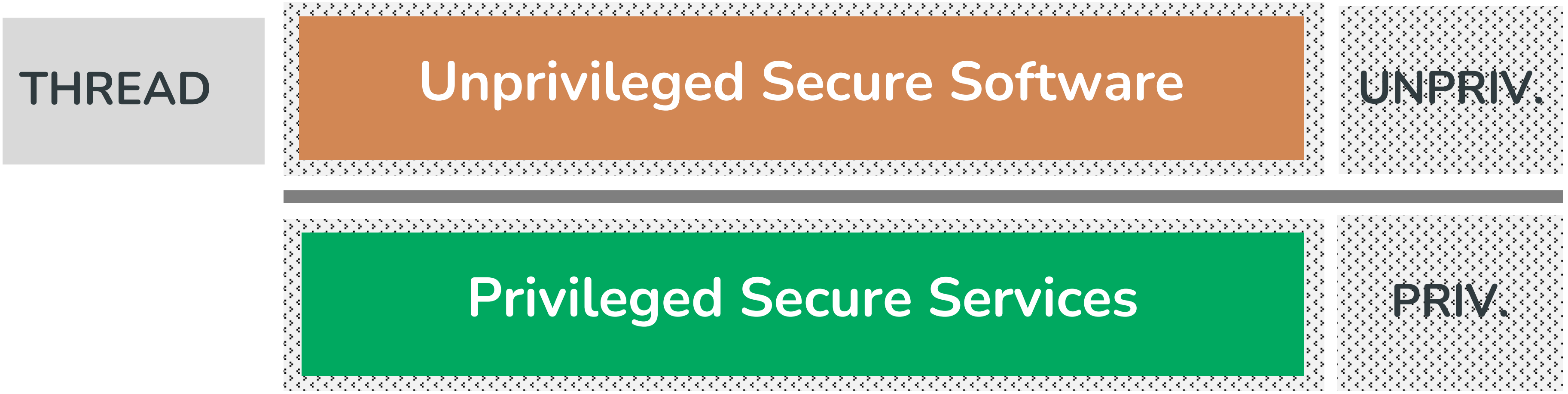
Platform Security Architecture (PSA)



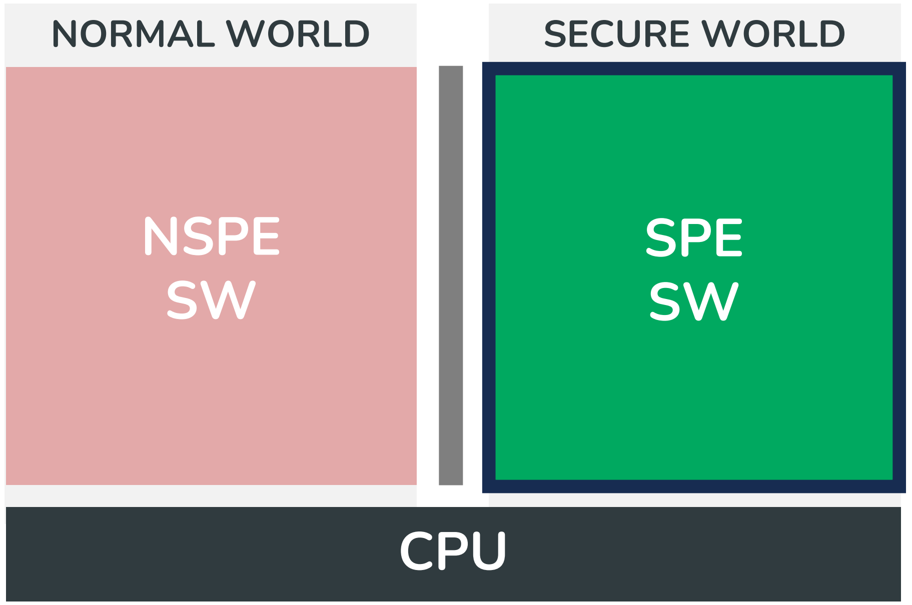


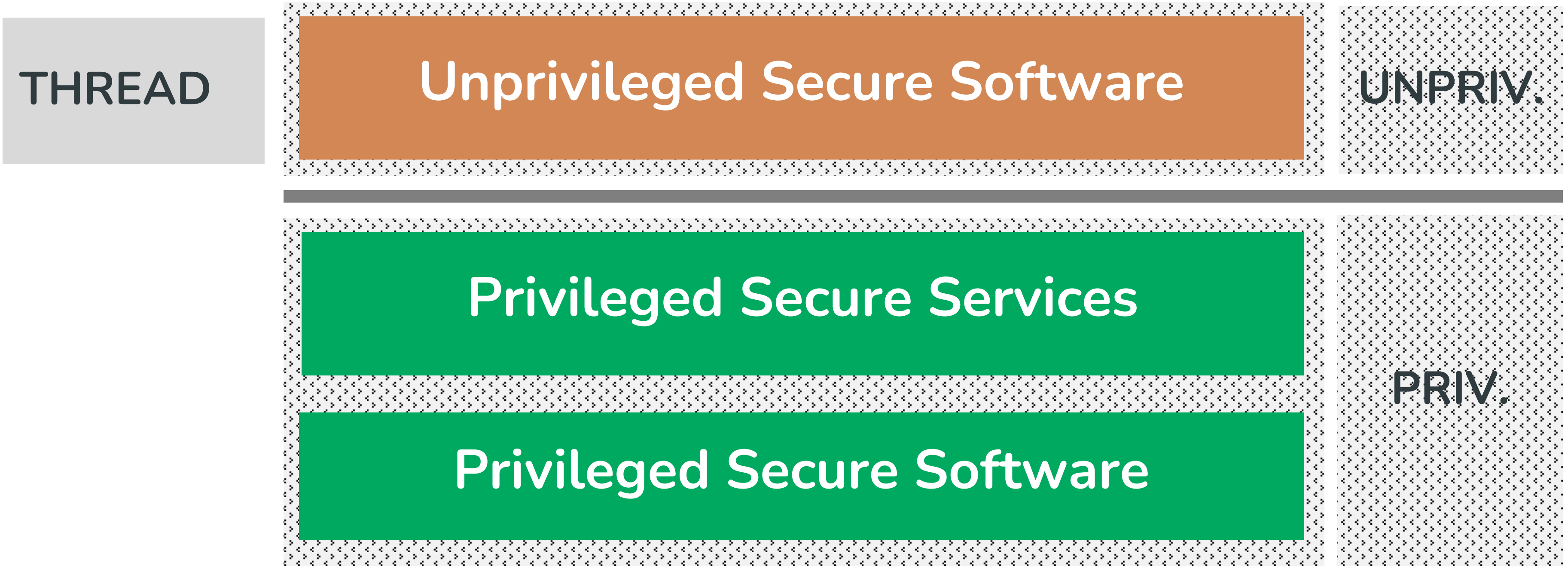
Platform Security Architecture (PSA)



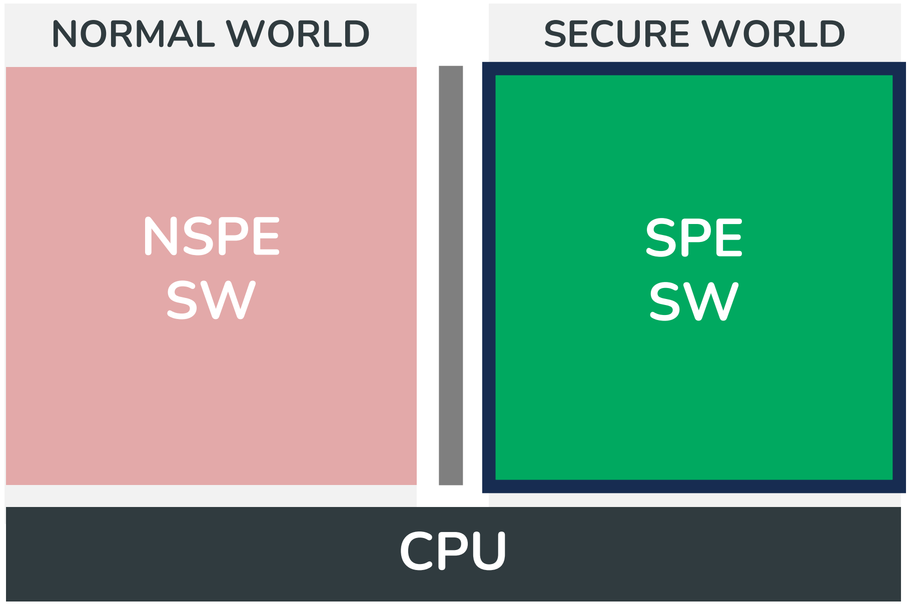


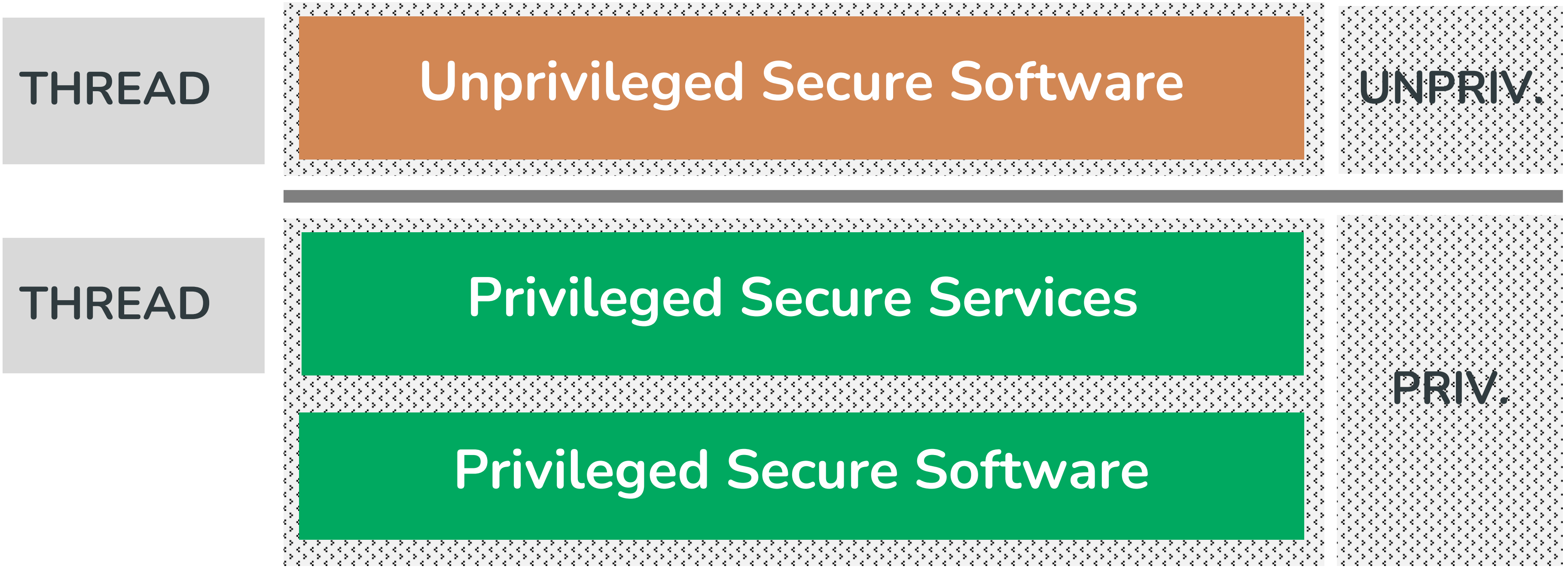
Platform Security Architecture (PSA)



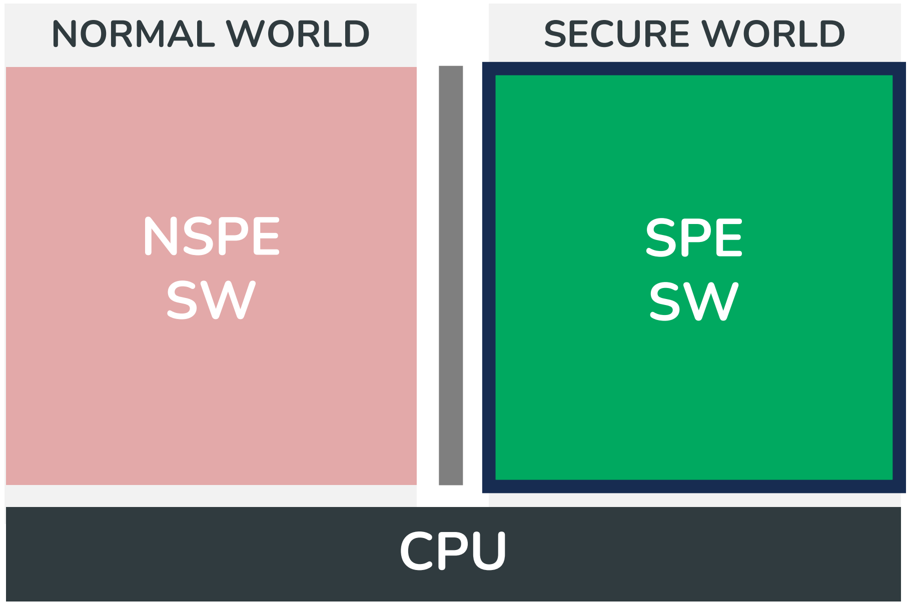

THREAD
Unprivileged Secure Software
UNPRIV.
Privileged Secure Services
Privileged Secure Software
PRIV.

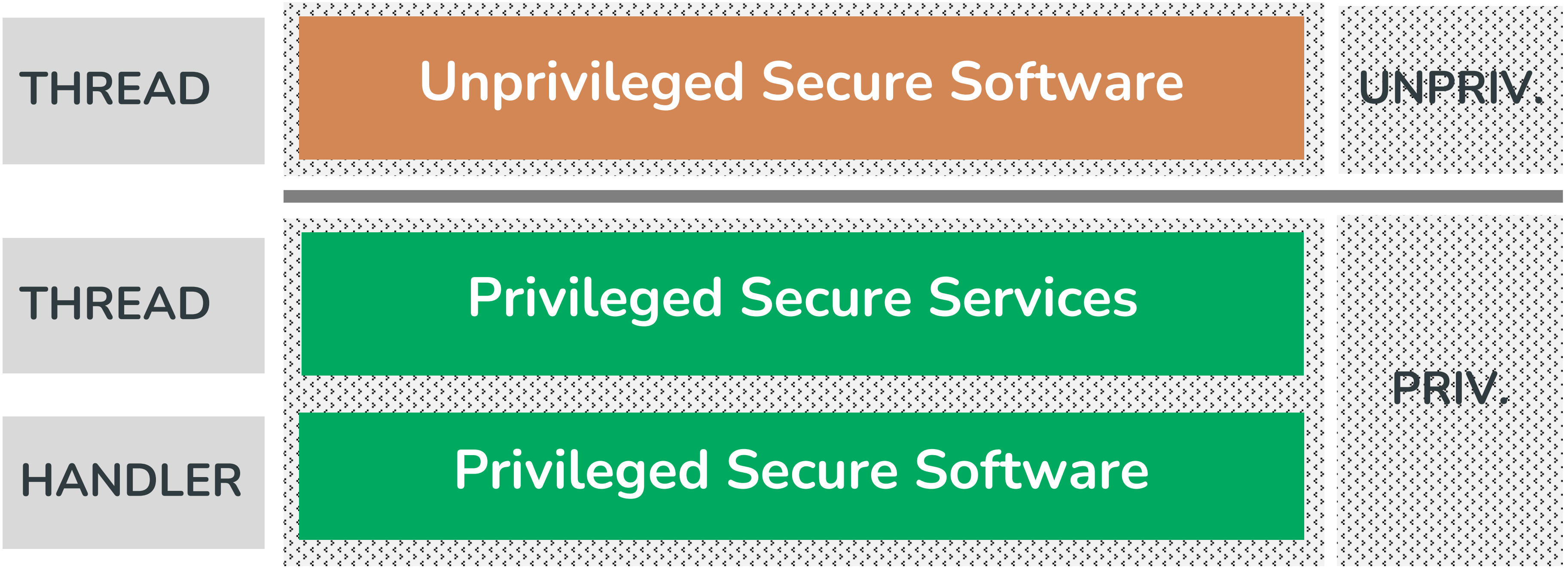
Platform Security Architecture (PSA)



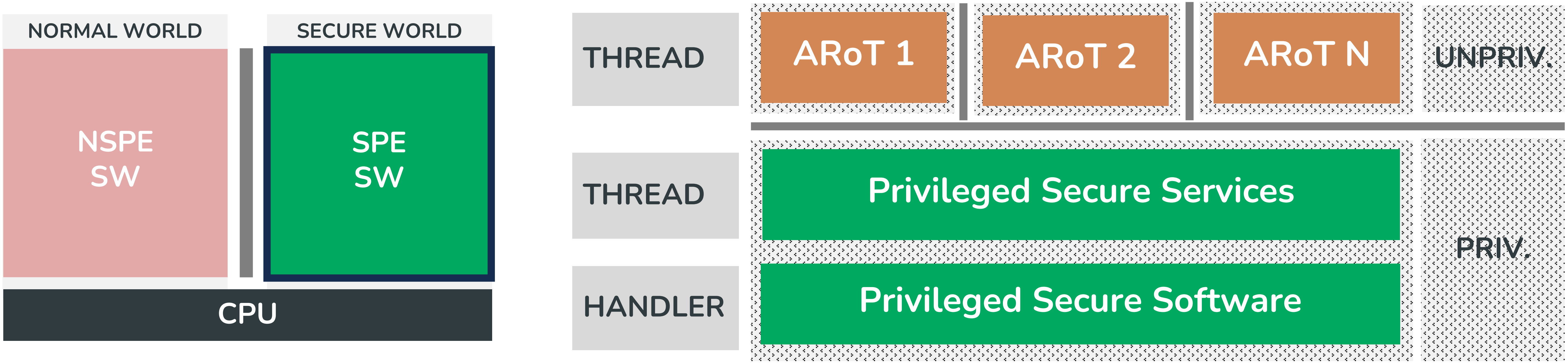

THREAD
Unprivileged Secure Software
UNPRIV.
THREAD
Privileged Secure Services
Privileged Secure Software
PRIV.

Platform Security Architecture (PSA)

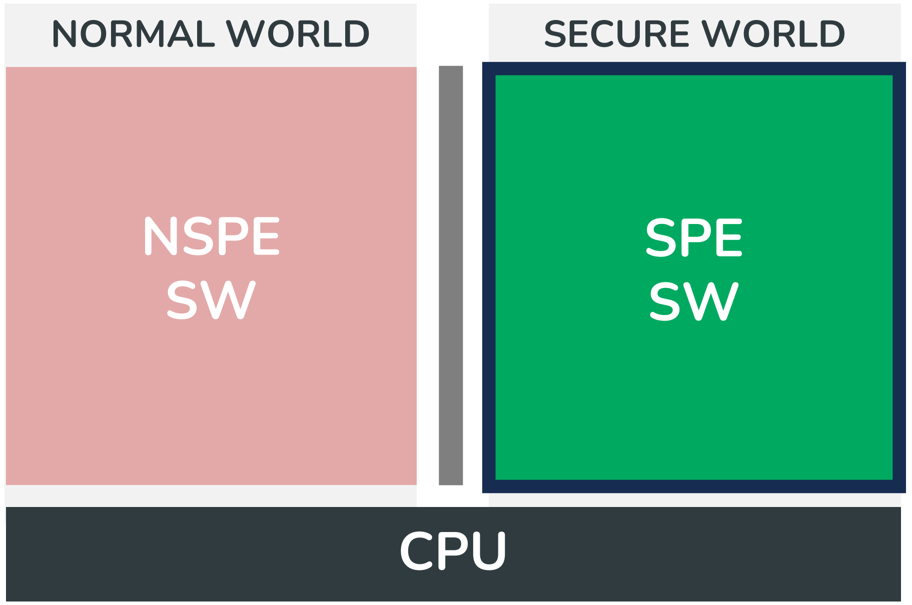


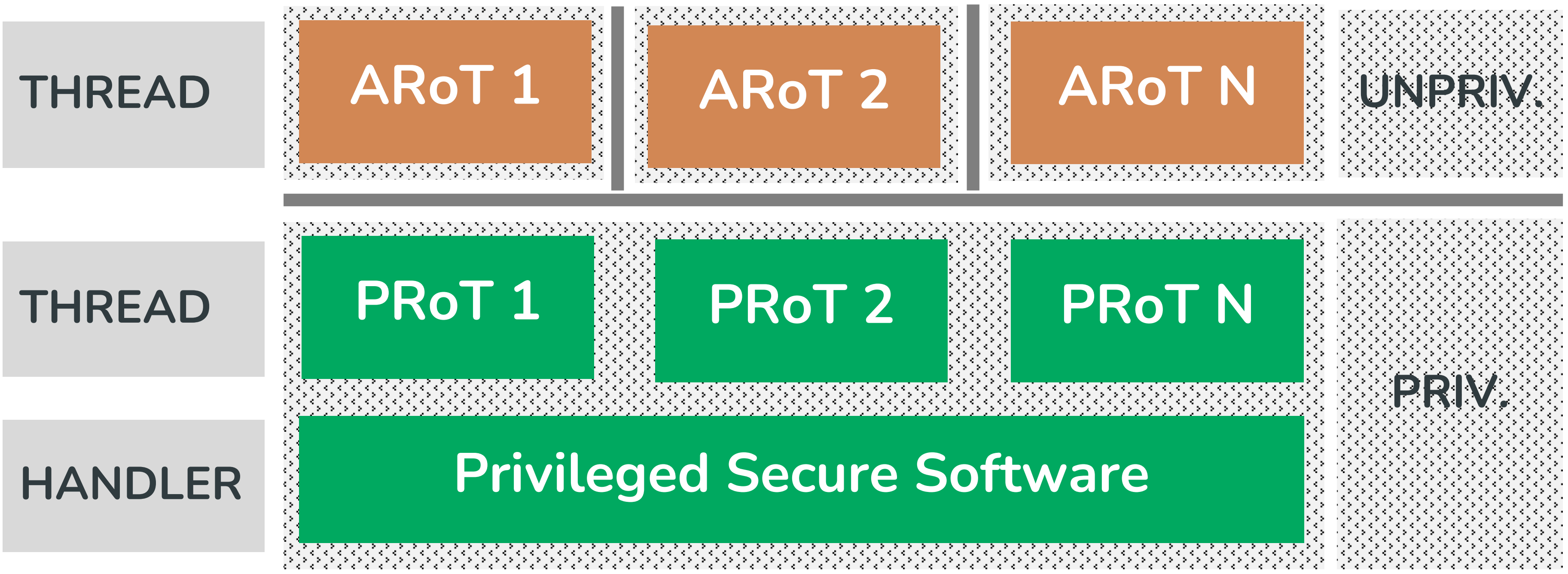

THREAD
Unprivileged Secure Software
UNPRIV.
THREAD
Privileged Secure Services
HANDLER
Privileged Secure Software
PRIV.

Platform Security Architecture (PSA)

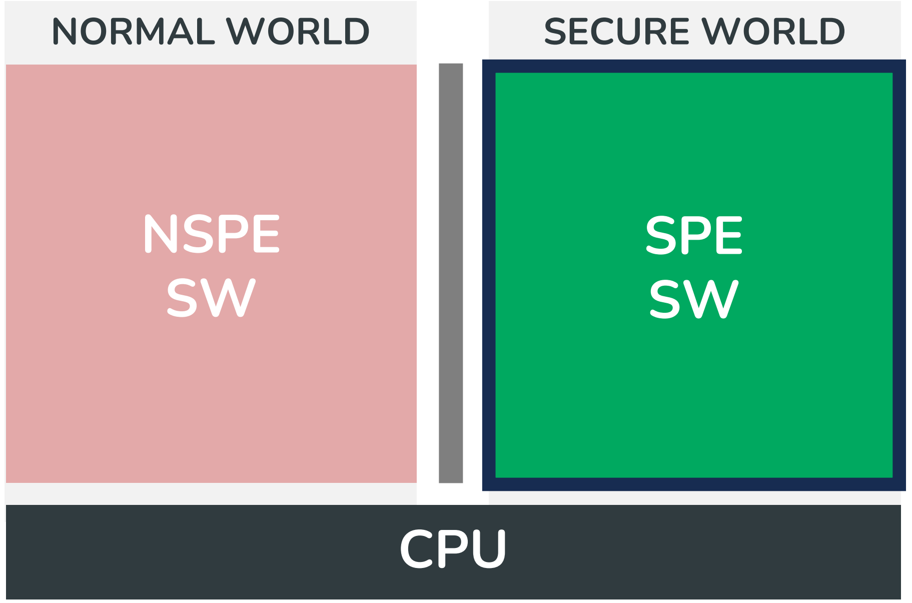


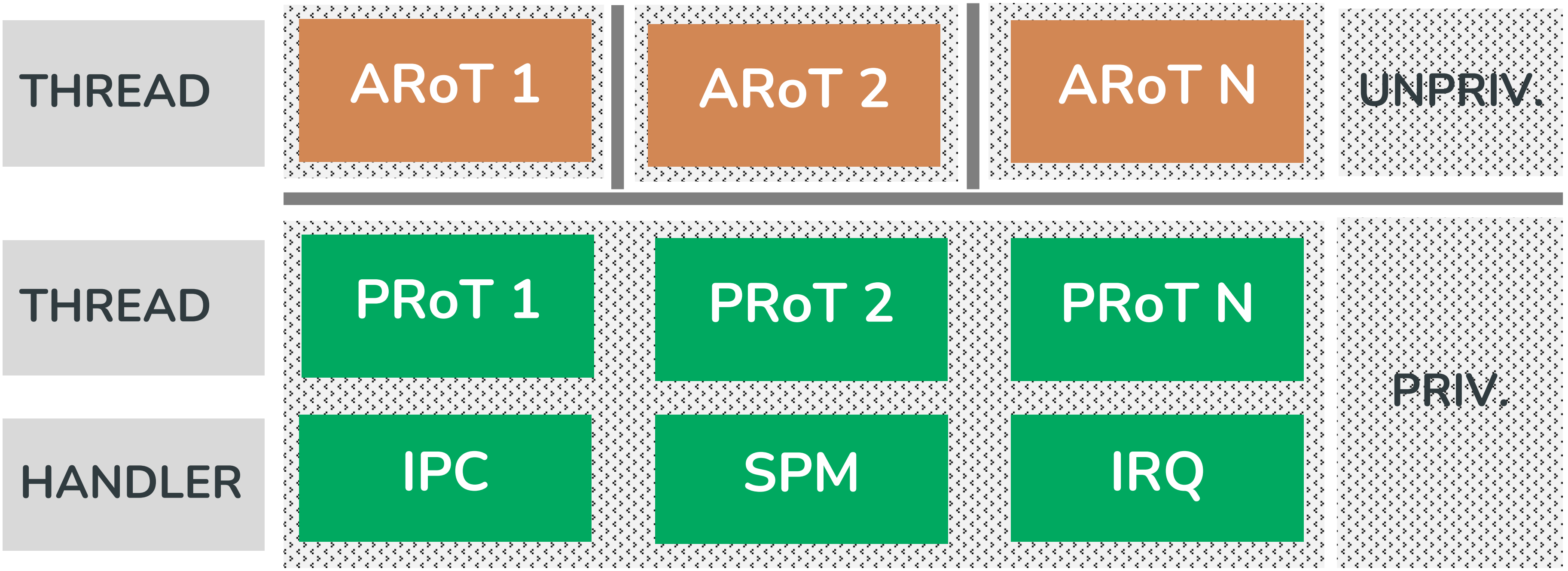
Platform Security Architecture (PSA)





Platform Security Architecture (PSA)



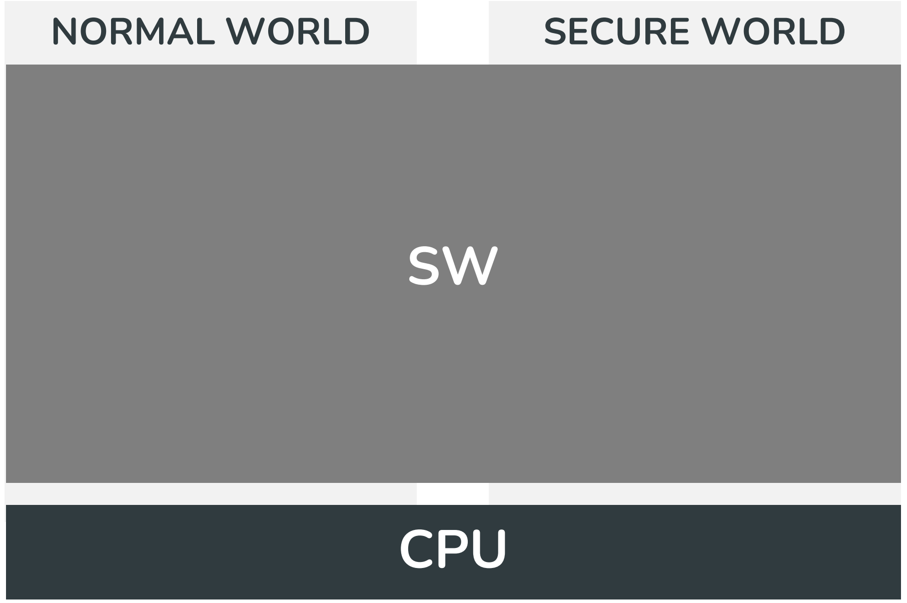


Platform Security Architecture (PSA)

PSA Level 1

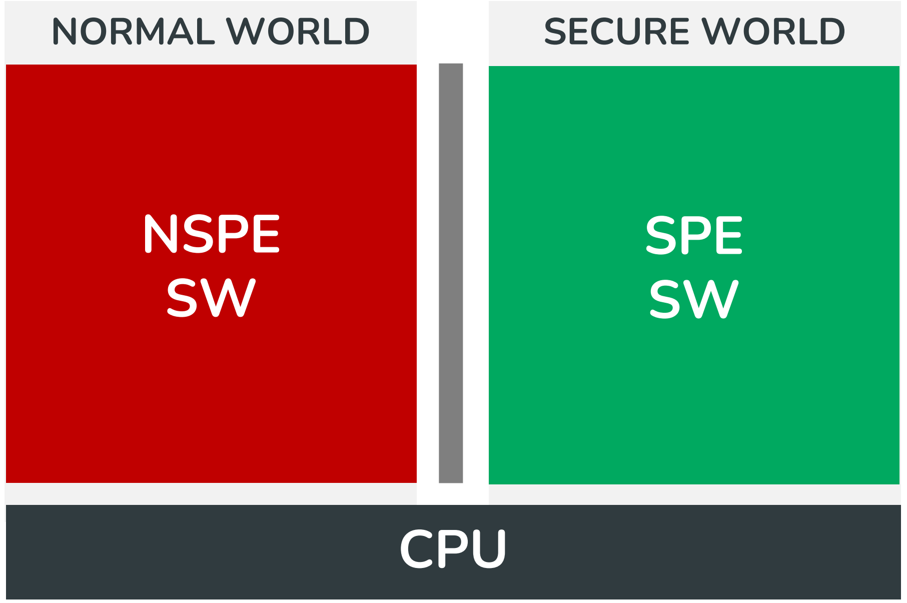
Platform Security Architecture (PSA)

PSA Level 1



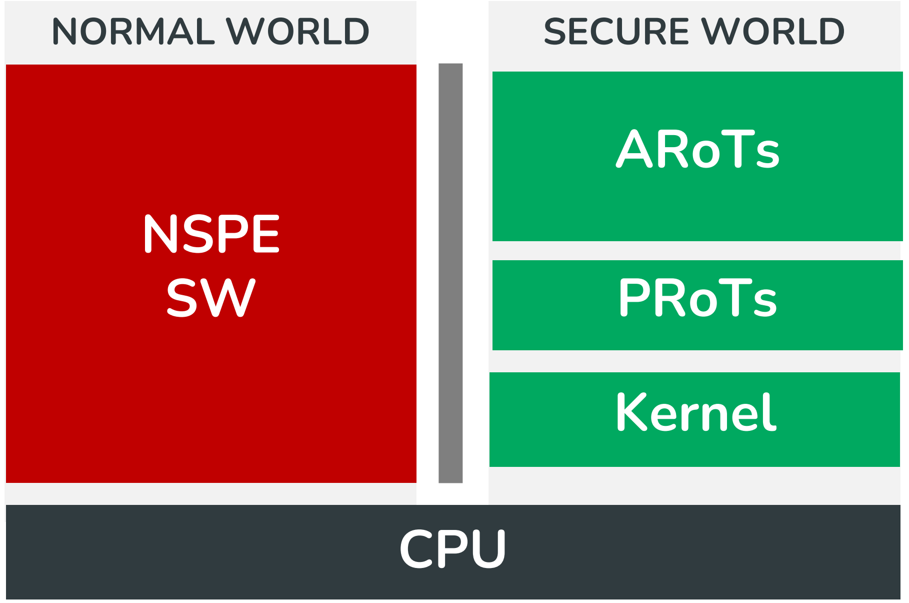
Platform Security Architecture (PSA)

PSA Level 1



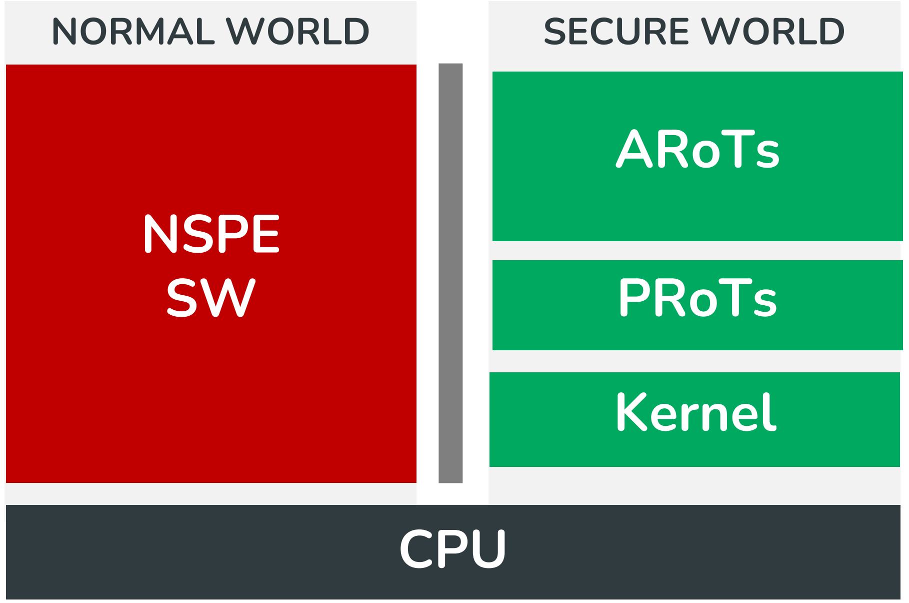
Platform Security Architecture (PSA)

PSA Level 1



Platform Security Architecture (PSA)

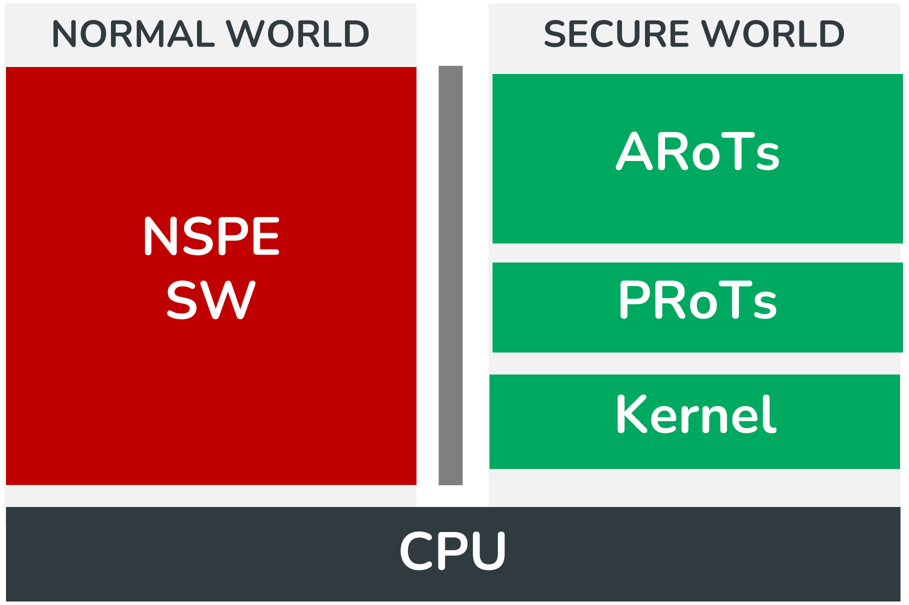
PSA Level 1



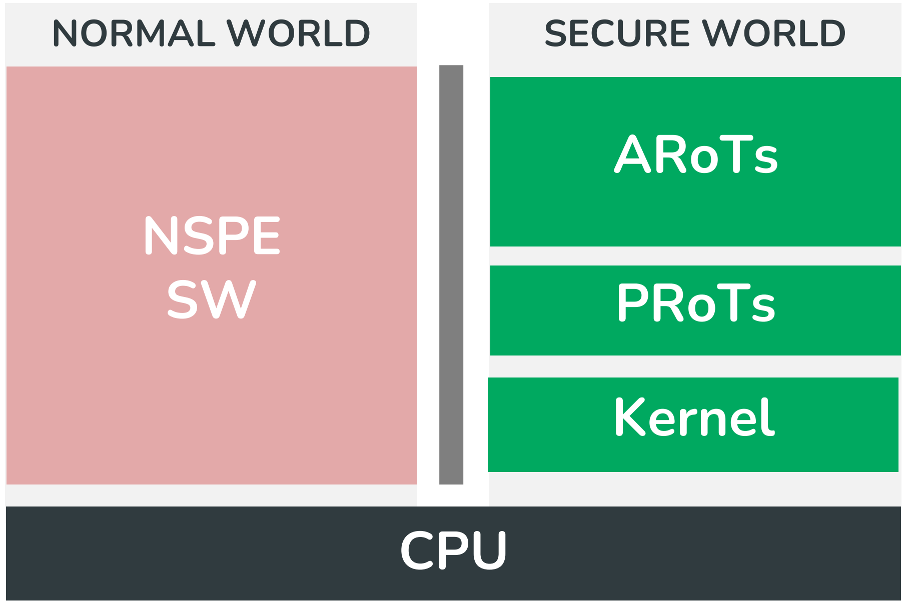
PSA Level 2

Platform Security Architecture (PSA)

PSA Level 1

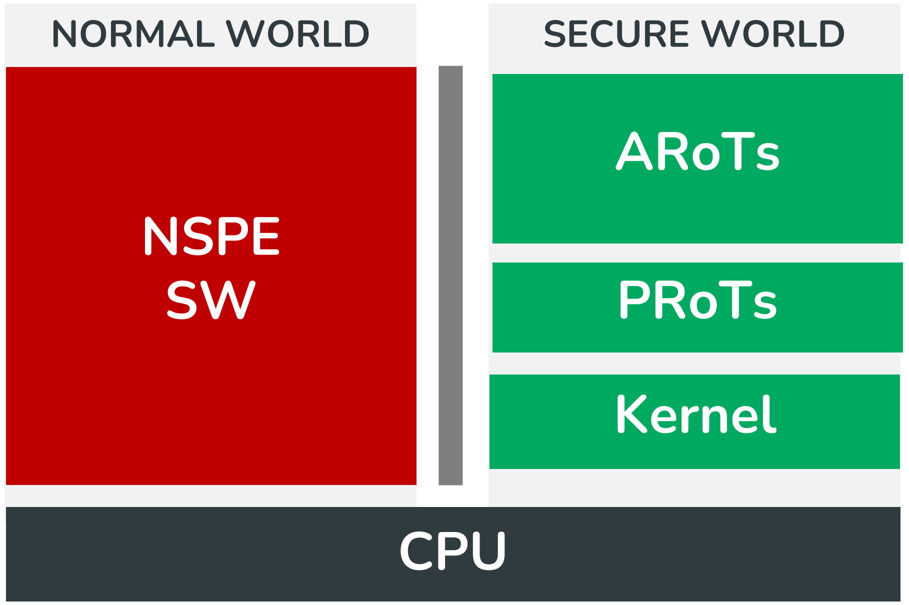


PSA Level 2

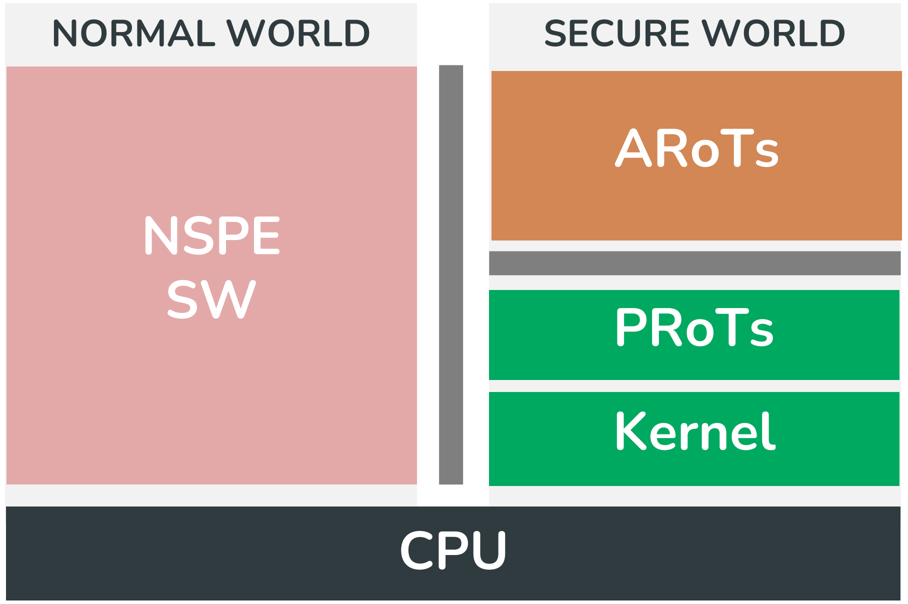


Platform Security Architecture (PSA)

PSA Level 1

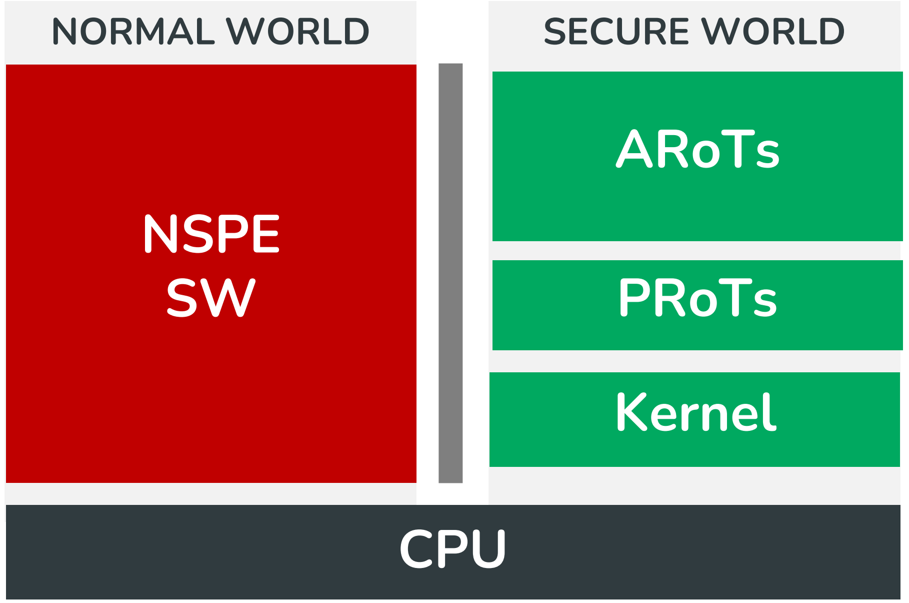


PSA Level 2

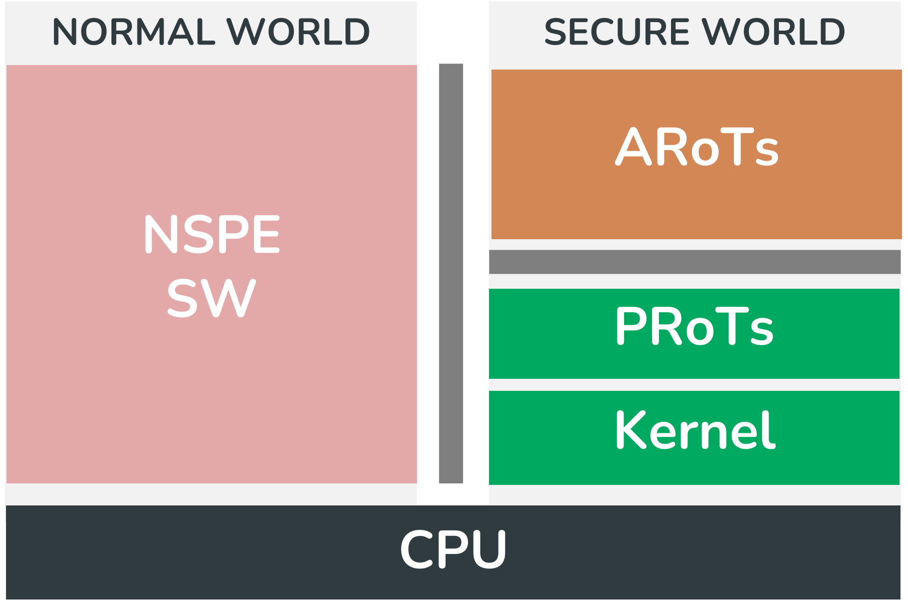


Platform Security Architecture (PSA)

PSA Level 1

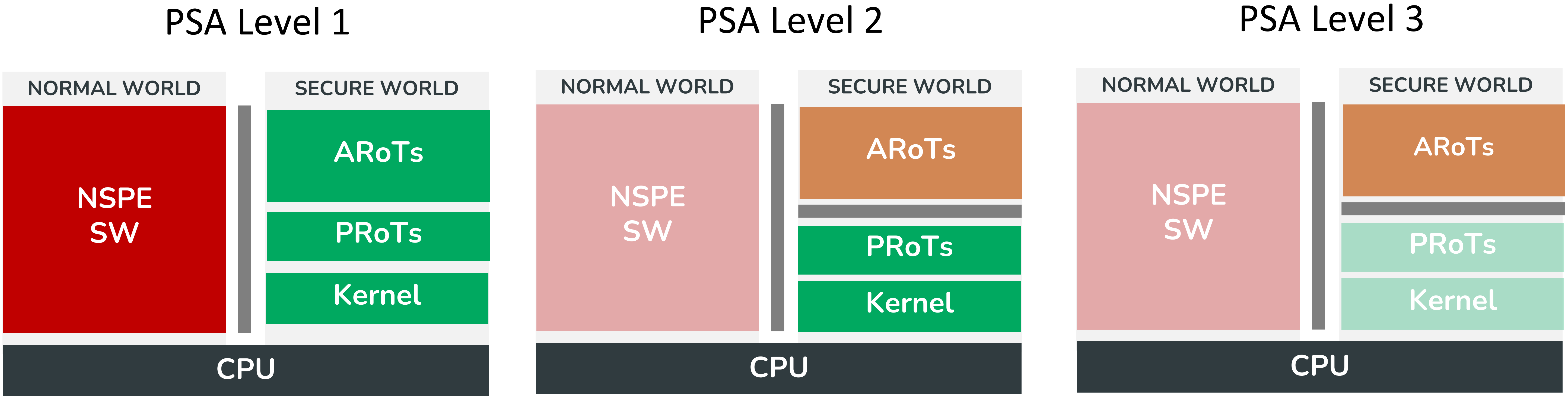


PSA Level 2

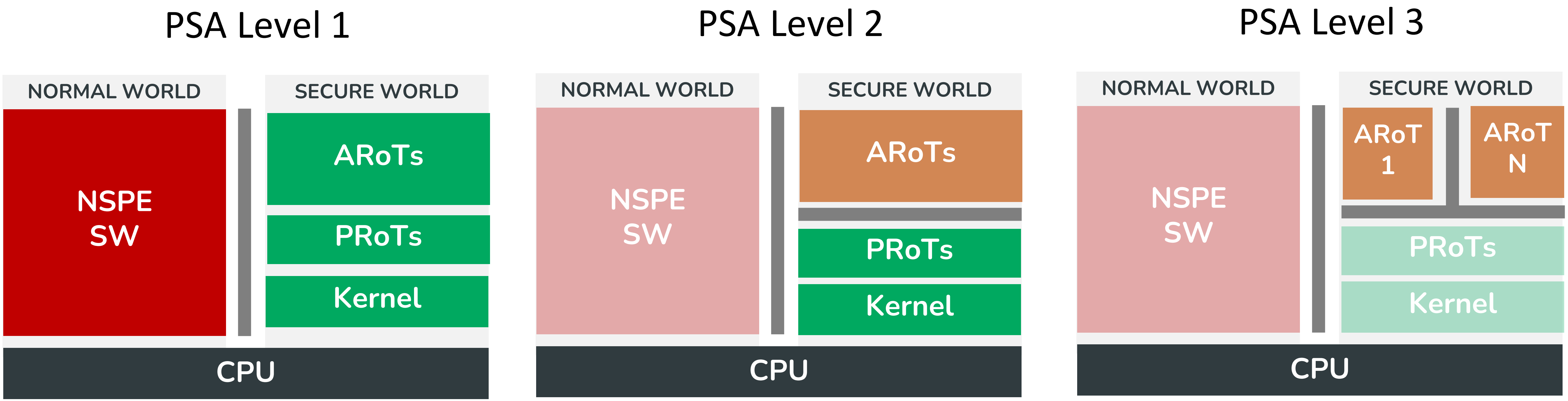


PSA Level 3

Platform Security Architecture (PSA)



Platform Security Architecture (PSA)



PARADOXAL OBSERVATIONS

01 **TRUSTZONE-M HAS A CPU-CENTRIC VIEW**

Armv8-M Only Defines Protection Controllers at The CPU-level (MPU, SAU, IDAU)

02 **SYSTEM-WIDE PROTECTIONS ARE PROPRIETARY**

Vendors Are Forced to Develop System Protection Controllers (PPCs, MPCs)

03 **MISSMATCH BETWEEN TZ-M AND PSA LEVELS**

PSA Level 2/3 Need CPU- and System-level Memory Protection Controllers (the latter isn’t defined by Armv8-M)

While System-Wide protections are a must, Armv8-M only defines CPU-level memory protections. We hypothesize that this dichotomy (together with a lack of understanding of the PSA isolation levels) may open security holes in modern TrustZone-M systems

Hypothesis

A Bumpy but Revealing Journey

Weak Protections, TEE Assessment and our Responsible Disclosure Journey

MICROCHIP

MICROCHIP

TRUSTONIC

MICROCHIP

SAML11

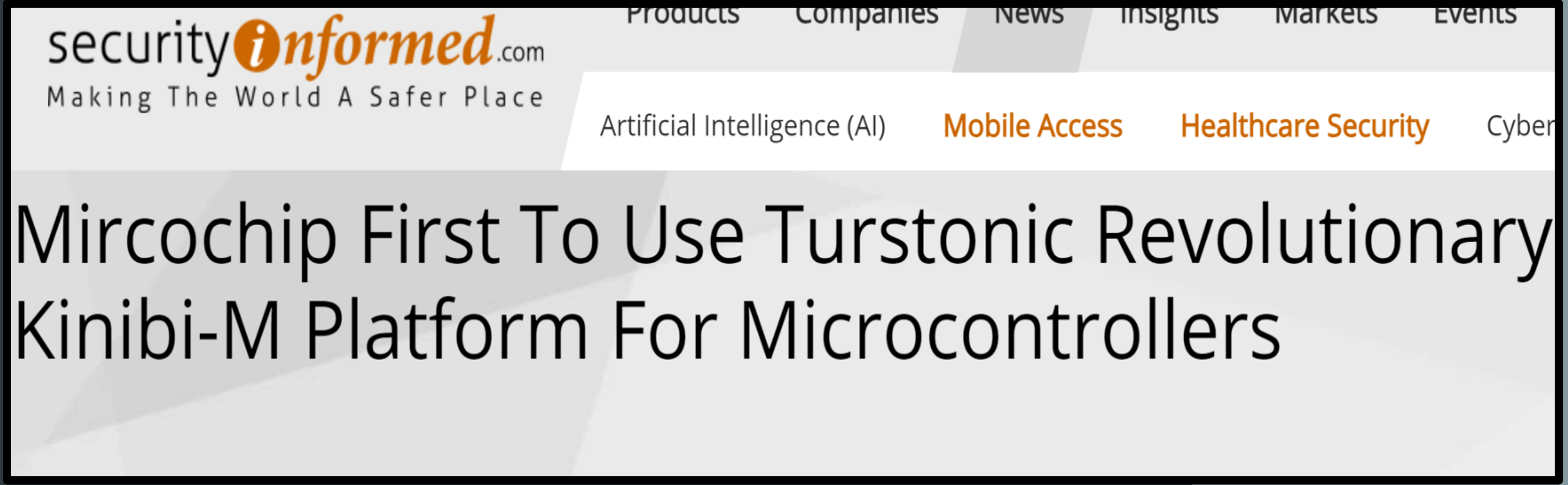
TRUSTONIC

MICROCHIP

SAML11

TRUSTONIC

Kinibi-M



securityinformed.com
Making The World A Safer Place

Products Companies News Insights Markets Events

Artificial Intelligence (AI) Mobile Access Healthcare Security Cyber

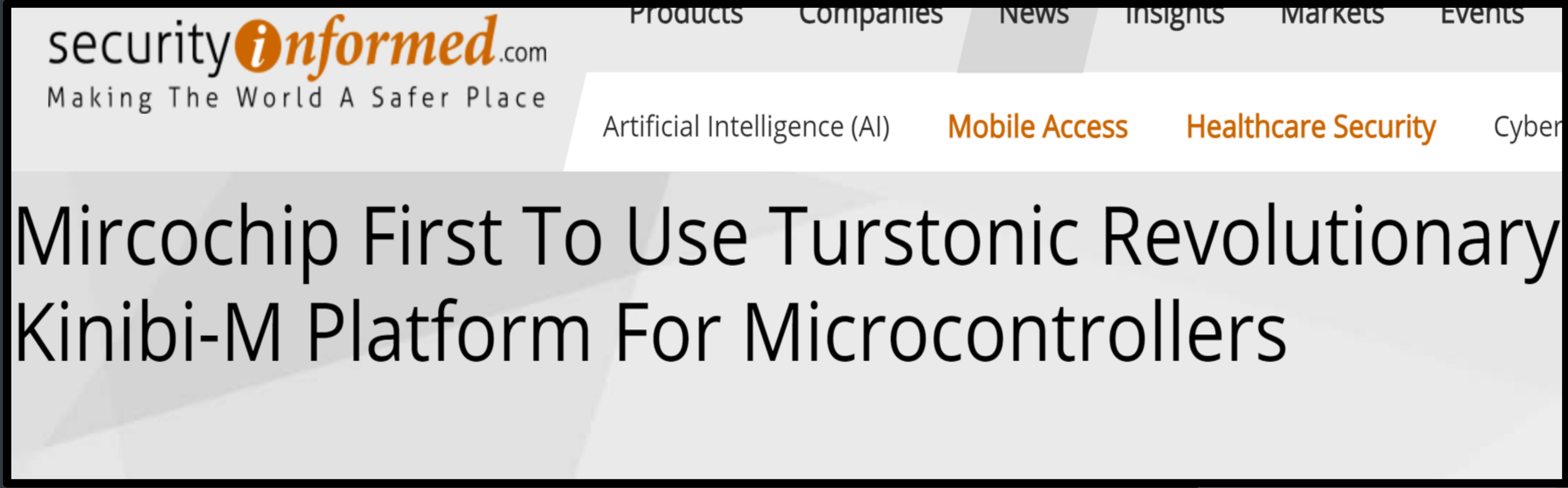
Mircochip First To Use Turstonic Revolutionary Kinibi-M Platform For Microcontrollers

MICROCHIP

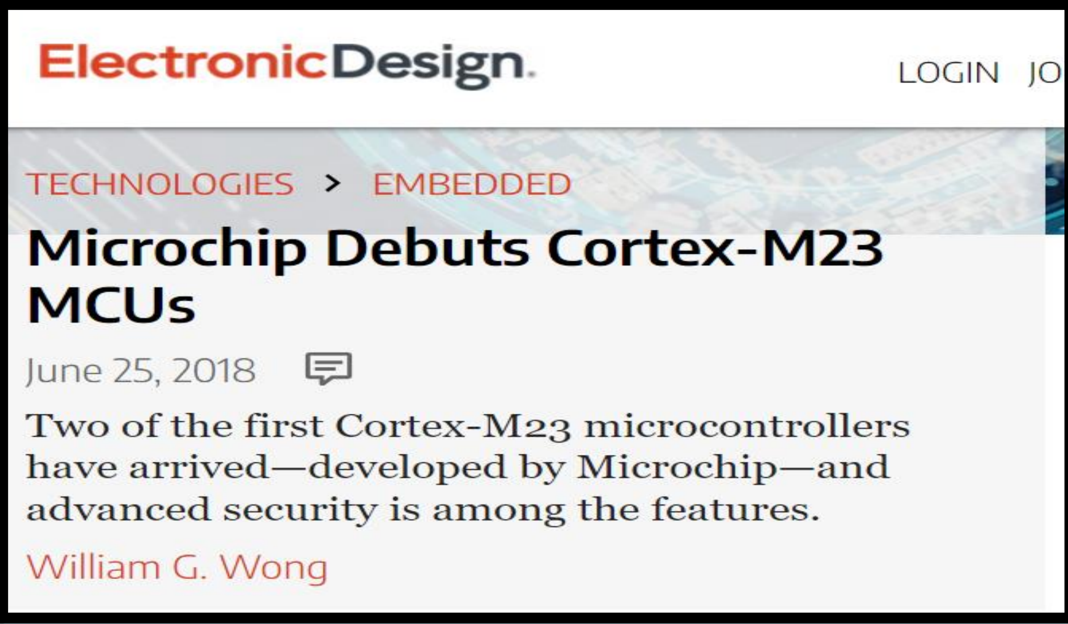
TRUSTONIC

SAML11

Kinibi-M



Mircochip First To Use Turstonic Revolutionary Kinibi-M Platform For Microcontrollers



Microchip Debuts Cortex-M23 MCUs

June 25, 2018

Two of the first Cortex-M23 microcontrollers have arrived—developed by Microchip—and advanced security is among the features.

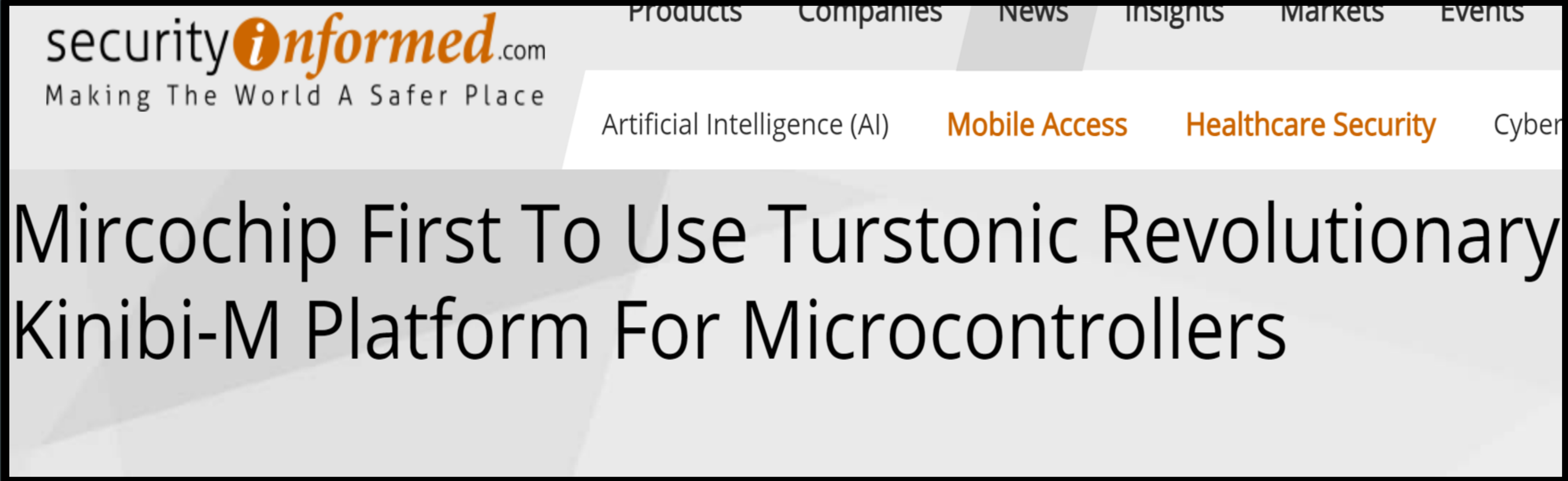
William G. Wong

MICROCHIP

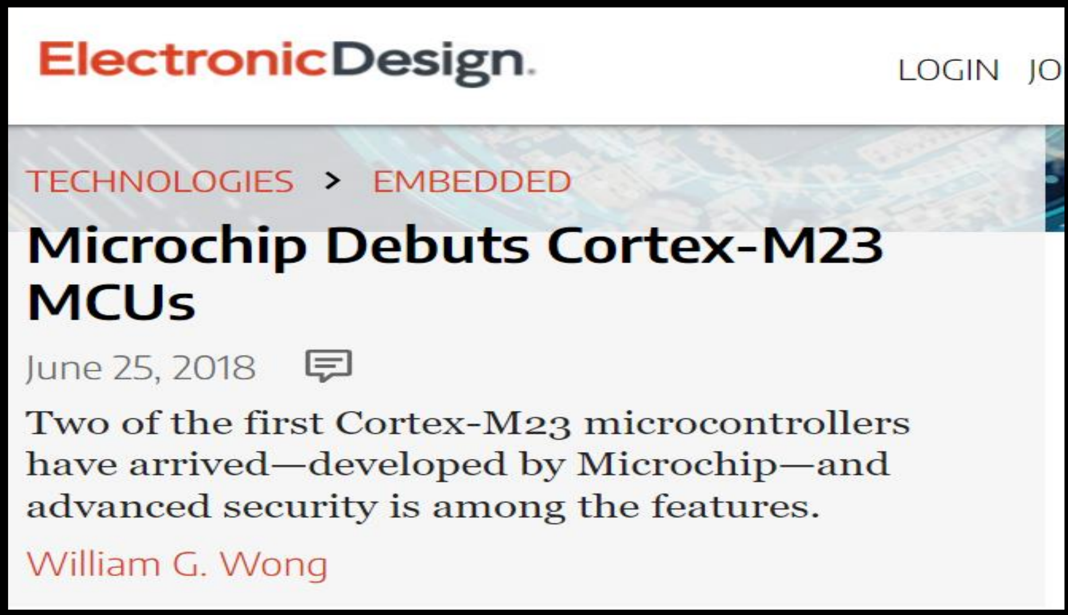
TRUSTONIC

SAML11

Kinibi-M



Mircochip First To Use Turstonic Revolutionary Kinibi-M Platform For Microcontrollers



Microchip Debuts Cortex-M23 MCUs

June 25, 2018

Two of the first Cortex-M23 microcontrollers have arrived—developed by Microchip—and advanced security is among the features.

William G. Wong

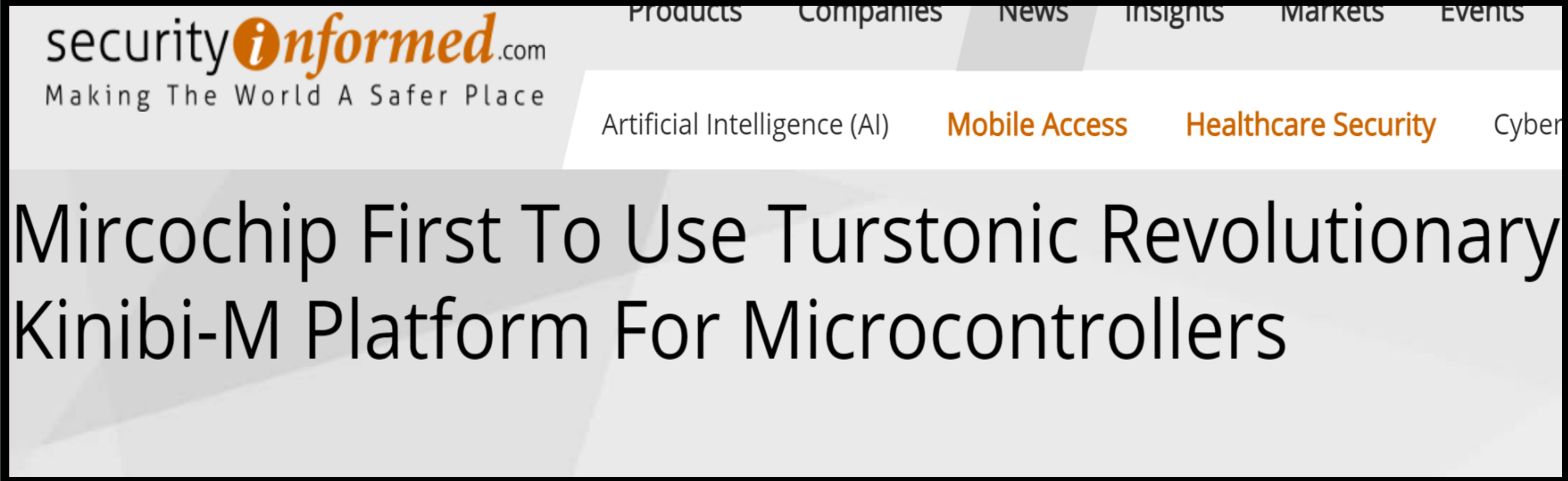
Not just droning on! The rise of Kinibi-M

31 OCTOBER 2017

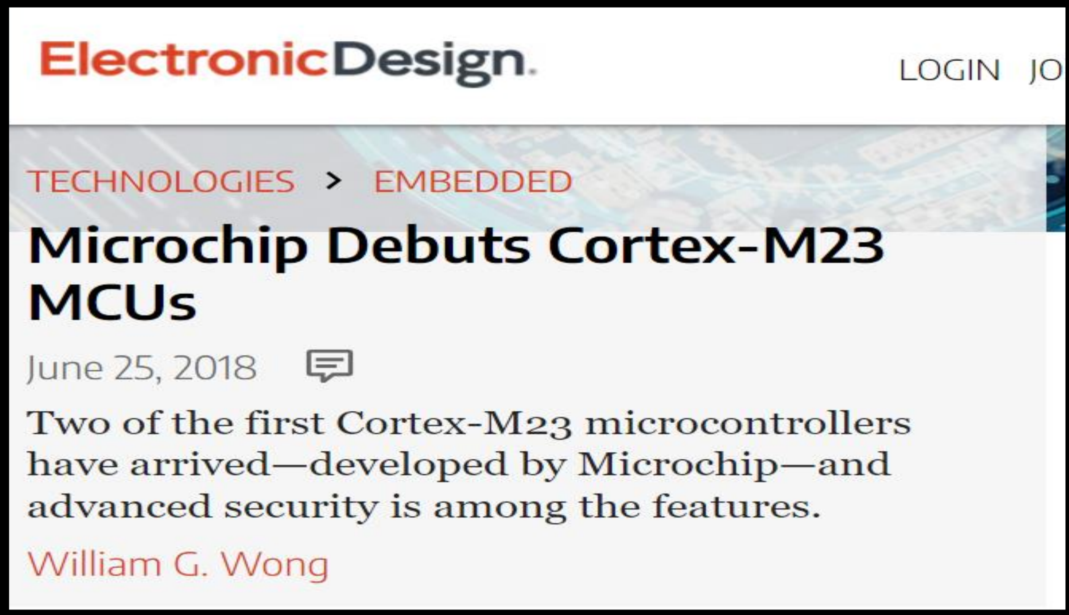
TRUSTONIC

SAML11

Kinibi-M



Mircochip First To Use Turstonic Revolutionary Kinibi-M Platform For Microcontrollers

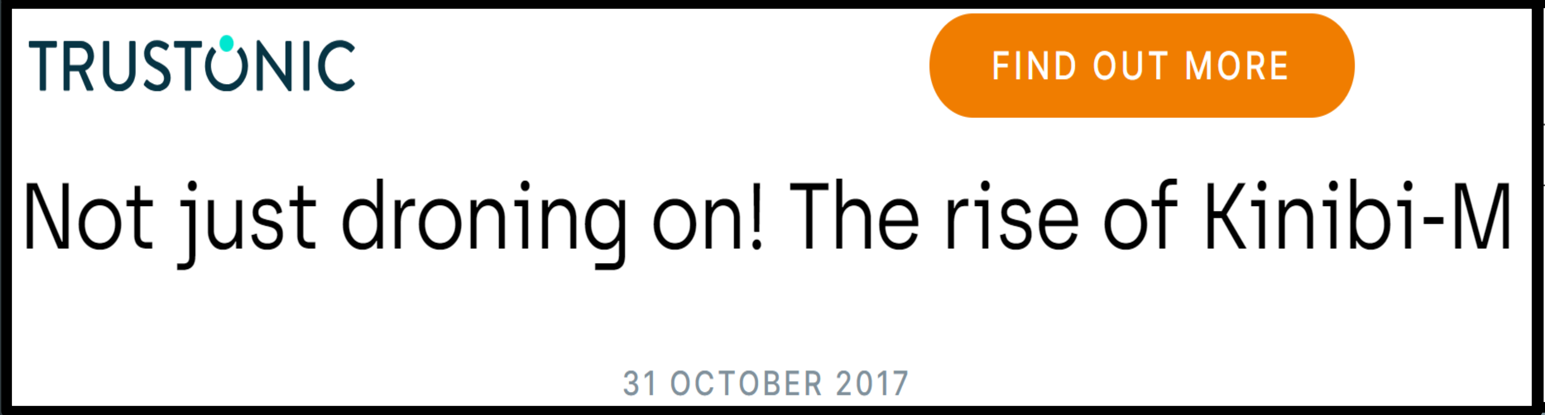


Microchip Debuts Cortex-M23 MCUs

June 25, 2018

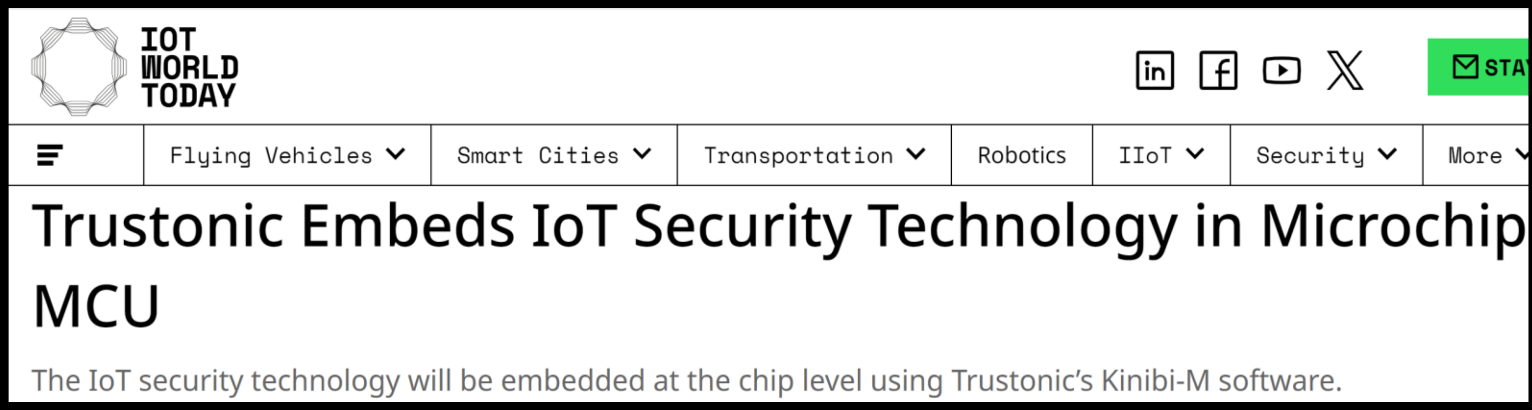
Two of the first Cortex-M23 microcontrollers have arrived—developed by Microchip—and advanced security is among the features.

William G. Wong



Not just droning on! The rise of Kinibi-M

31 OCTOBER 2017

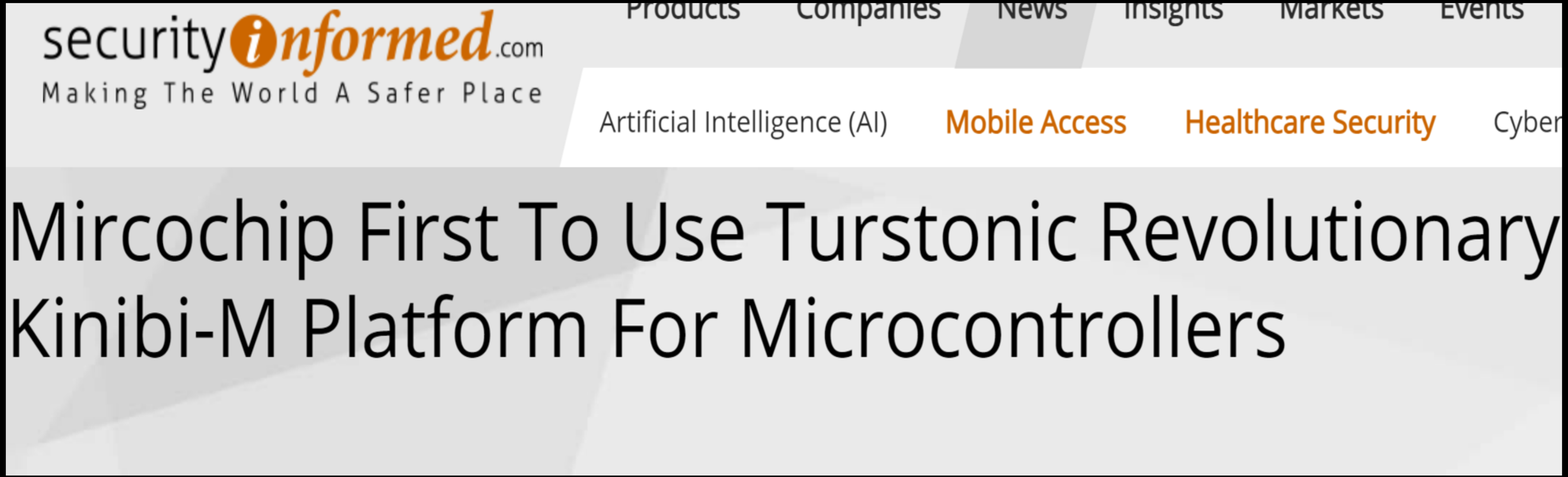


Trustonic Embeds IoT Security Technology in Microchip MCU

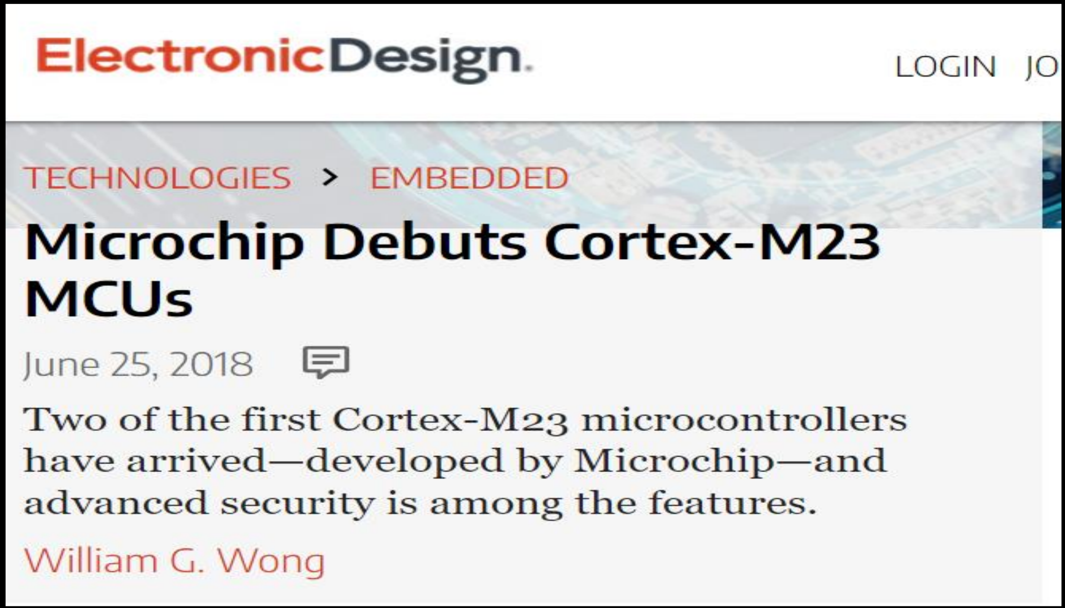
The IoT security technology will be embedded at the chip level using Trustonic's Kinibi-M software.

SAML11

Kinibi-M



Mircochip First To Use Turstonic Revolutionary Kinibi-M Platform For Microcontrollers

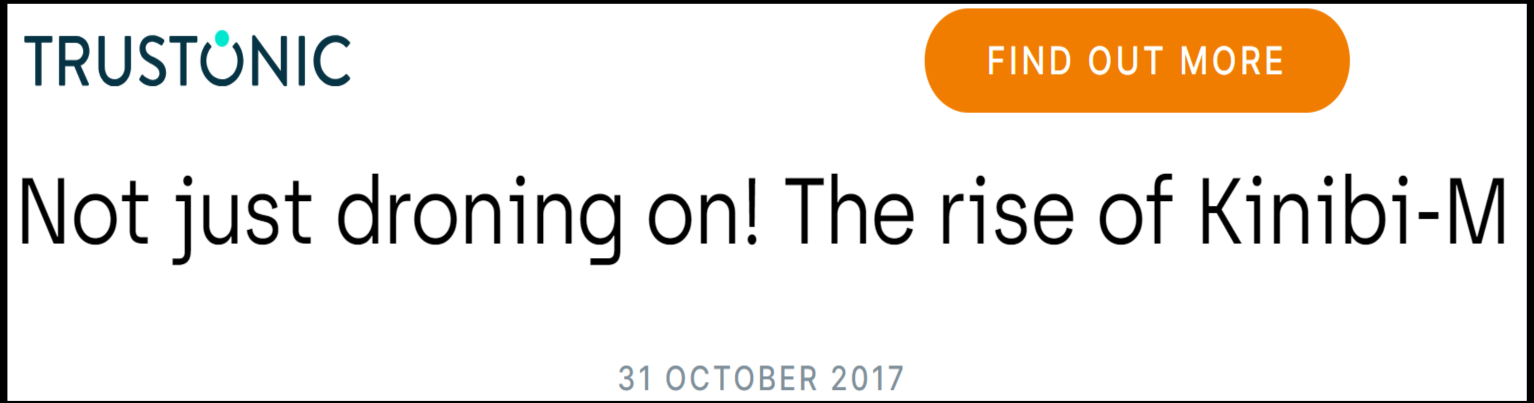


Microchip Debuts Cortex-M23 MCUs

June 25, 2018

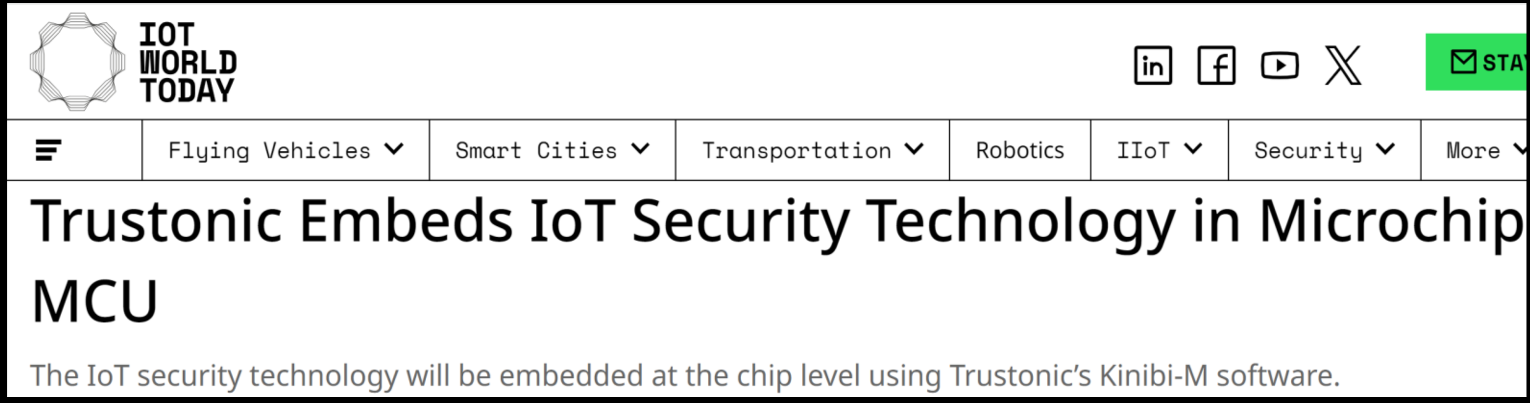
Two of the first Cortex-M23 microcontrollers have arrived—developed by Microchip—and advanced security is among the features.

William G. Wong



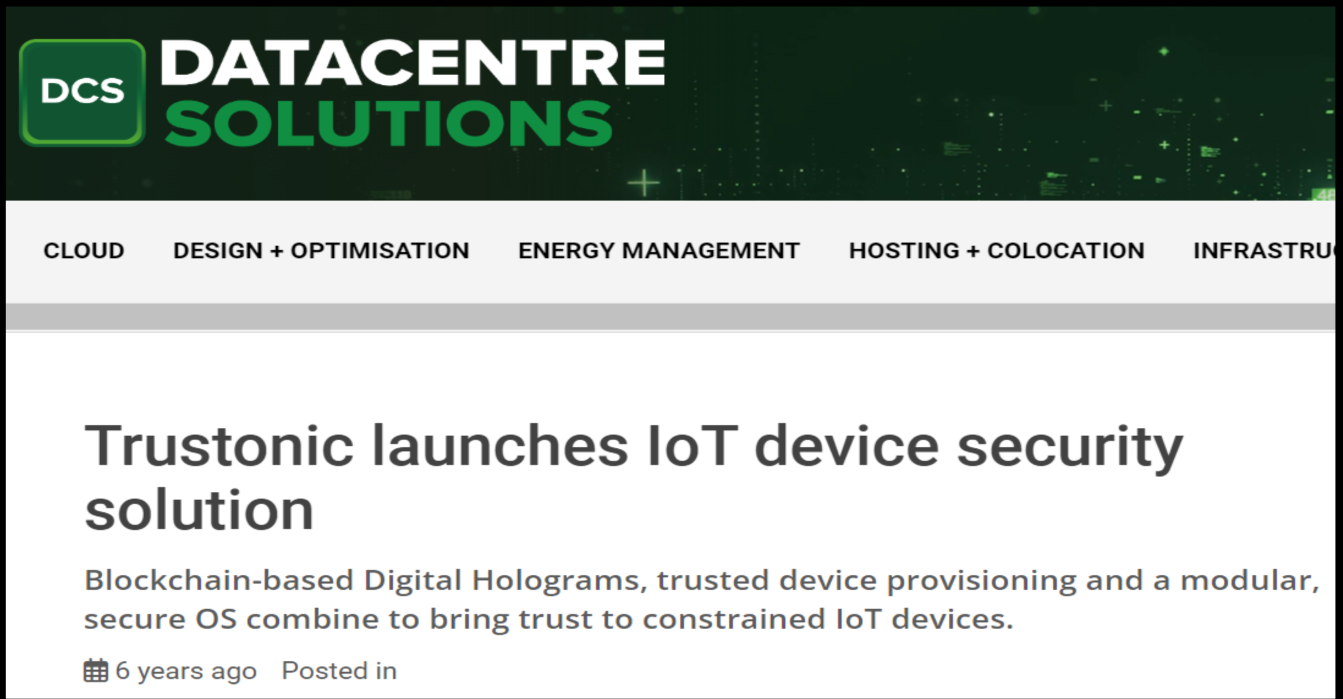
Not just droning on! The rise of Kinibi-M

31 OCTOBER 2017



Trustonic Embeds IoT Security Technology in Microchip MCU

The IoT security technology will be embedded at the chip level using Trustonic's Kinibi-M software.

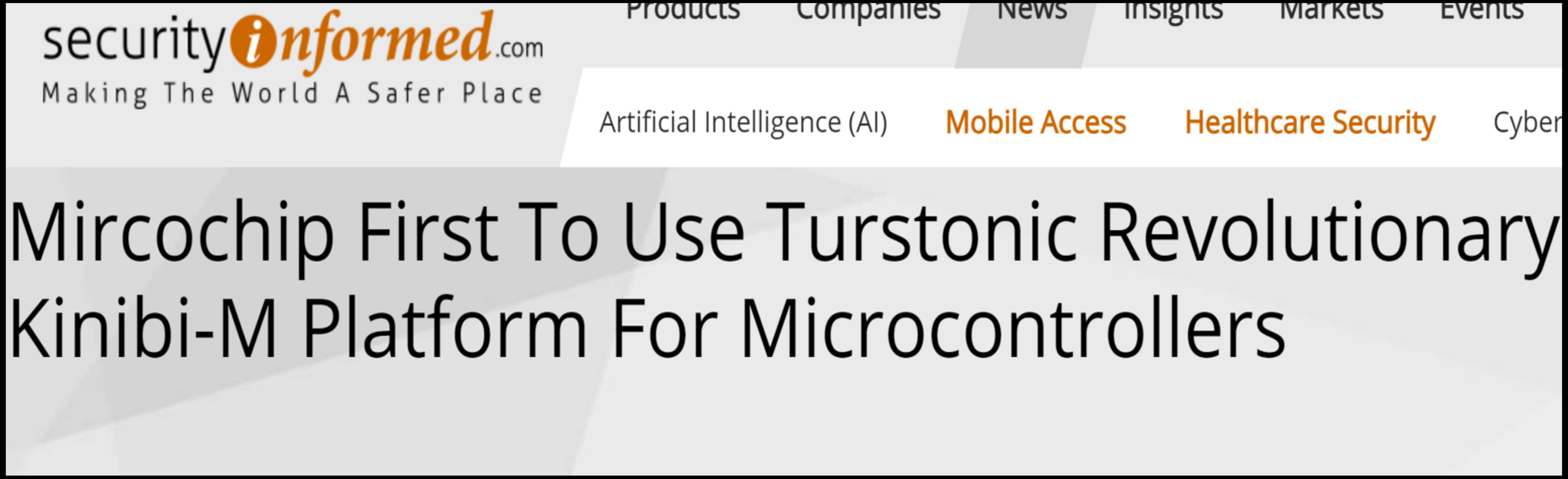


Trustonic launches IoT device security solution

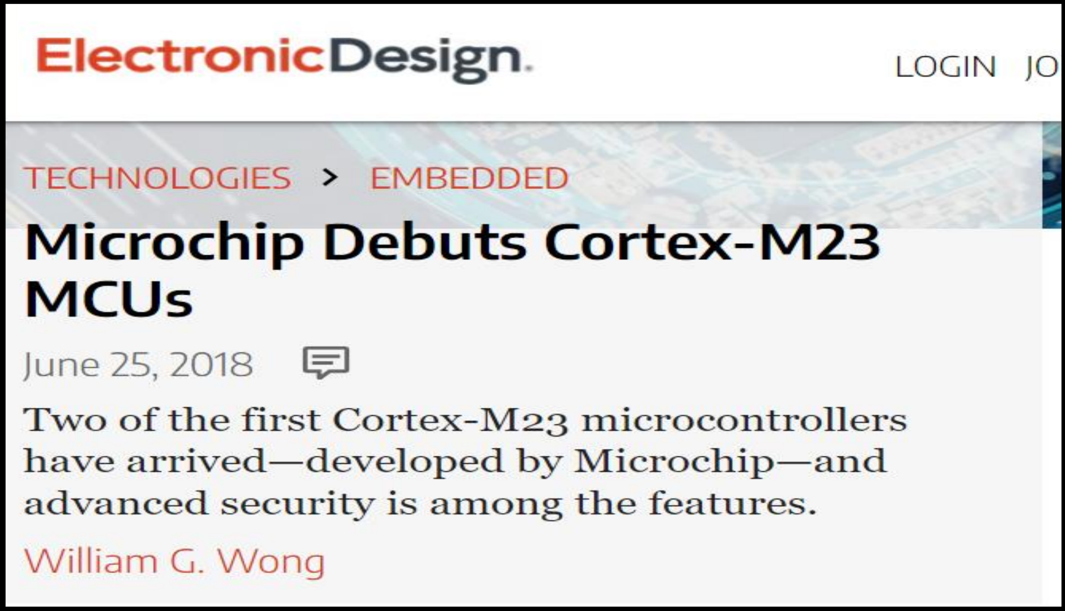
Blockchain-based Digital Holograms, trusted device provisioning and a modular, secure OS combine to bring trust to constrained IoT devices.

6 years ago Posted in

Kinibi-M



Mircochip First To Use Turstonic Revolutionary Kinibi-M Platform For Microcontrollers

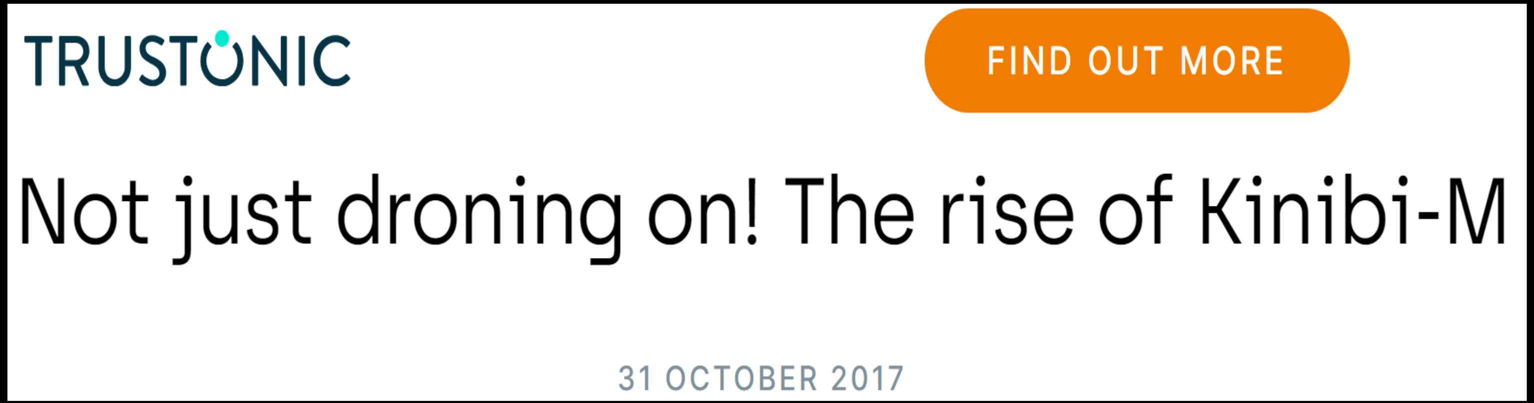


Microchip Debuts Cortex-M23 MCUs

June 25, 2018

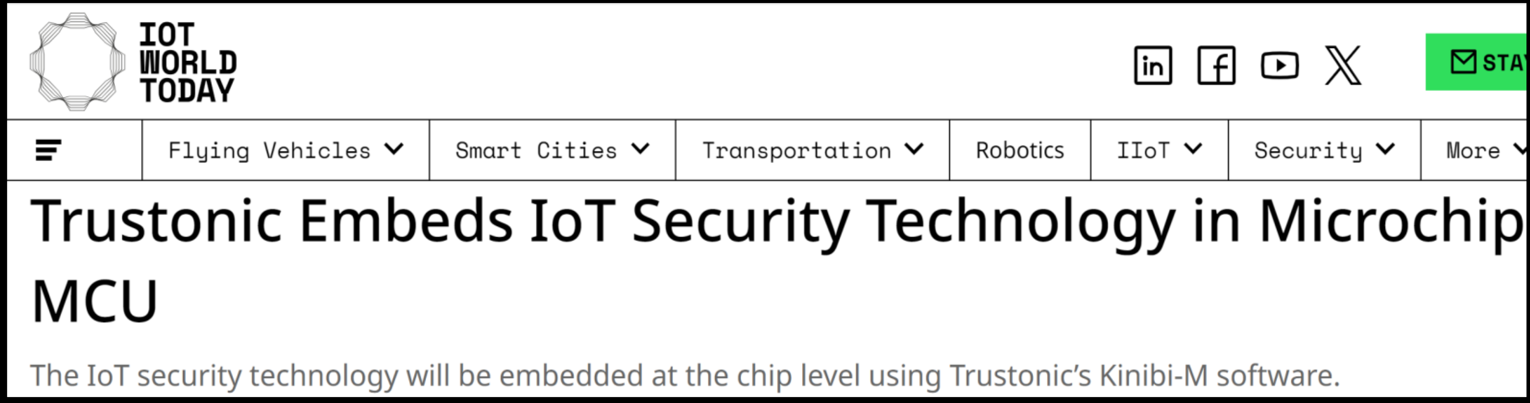
Two of the first Cortex-M23 microcontrollers have arrived—developed by Microchip—and advanced security is among the features.

William G. Wong



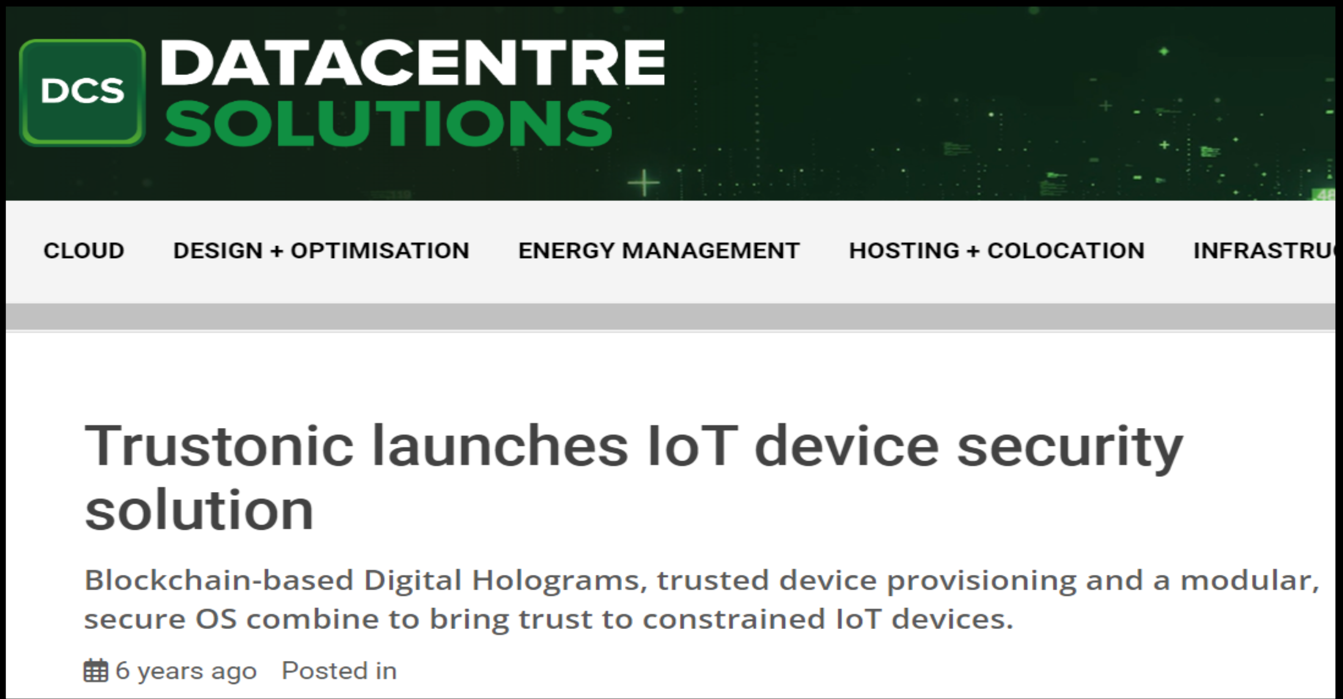
Not just droning on! The rise of Kinibi-M

31 OCTOBER 2017



Trustonic Embeds IoT Security Technology in Microchip MCU

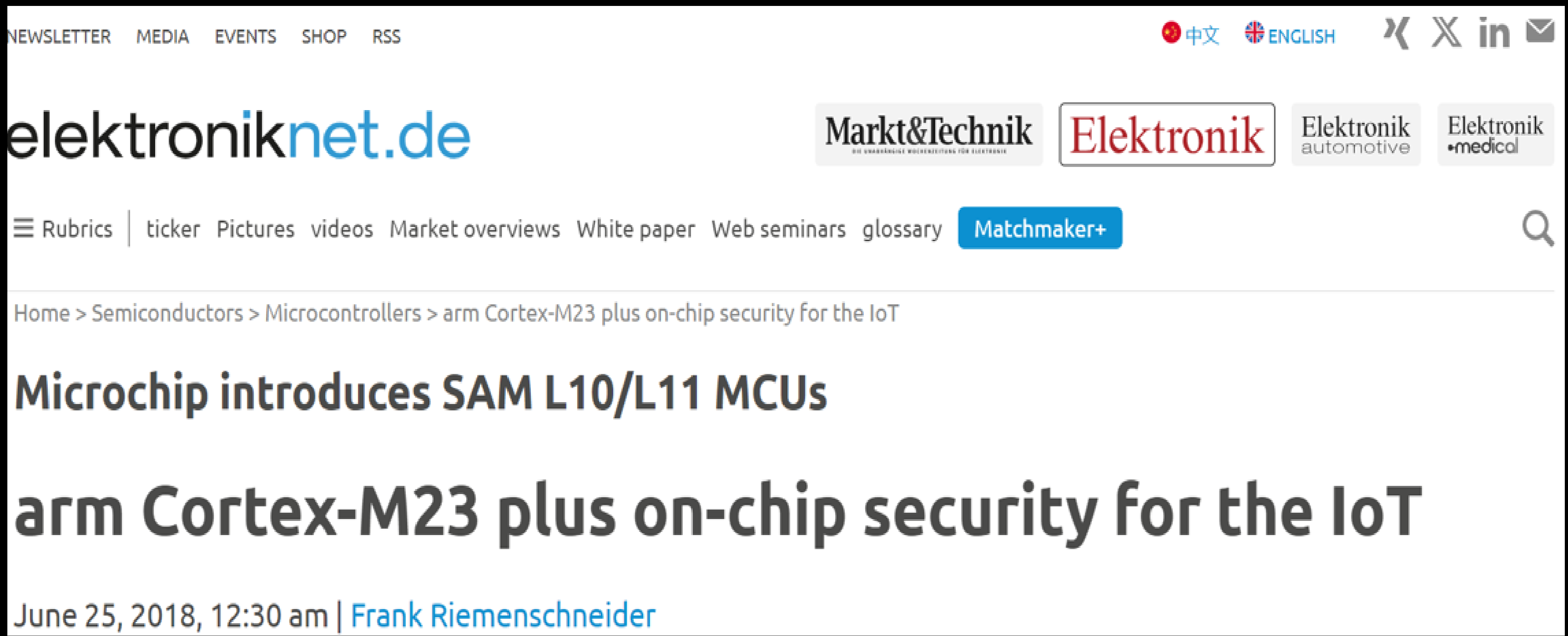
The IoT security technology will be embedded at the chip level using Trustonic's Kinibi-M software.



Trustonic launches IoT device security solution

Blockchain-based Digital Holograms, trusted device provisioning and a modular, secure OS combine to bring trust to constrained IoT devices.

6 years ago Posted in



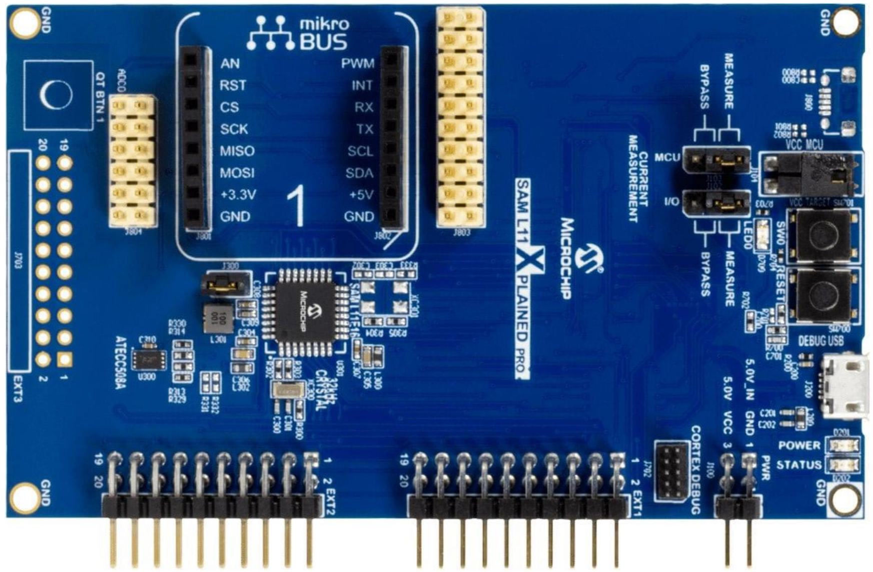
Home > Semiconductors > Microcontrollers > arm Cortex-M23 plus on-chip security for the IoT

Microchip introduces SAM L10/L11 MCUs

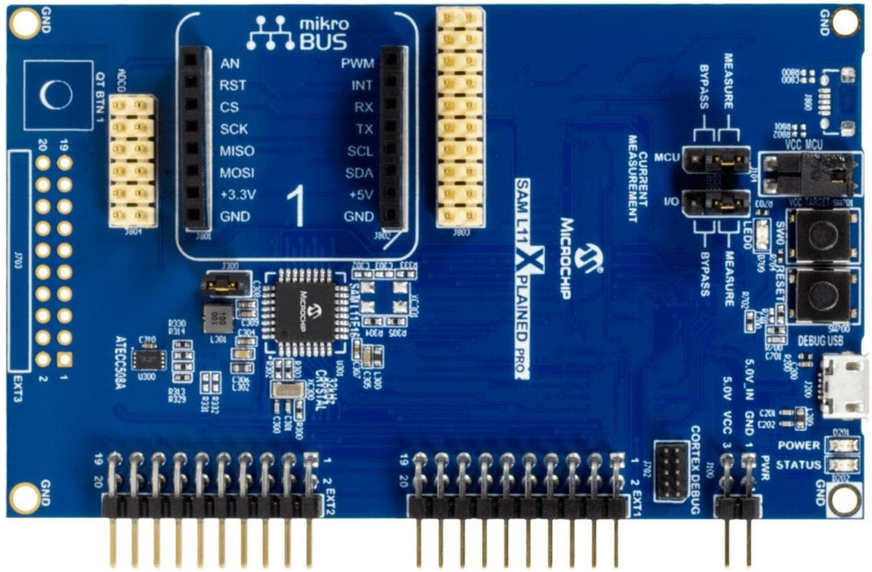
arm Cortex-M23 plus on-chip security for the IoT

June 25, 2018, 12:30 am | Frank Riemenschneider

MICROCHIP SAML11



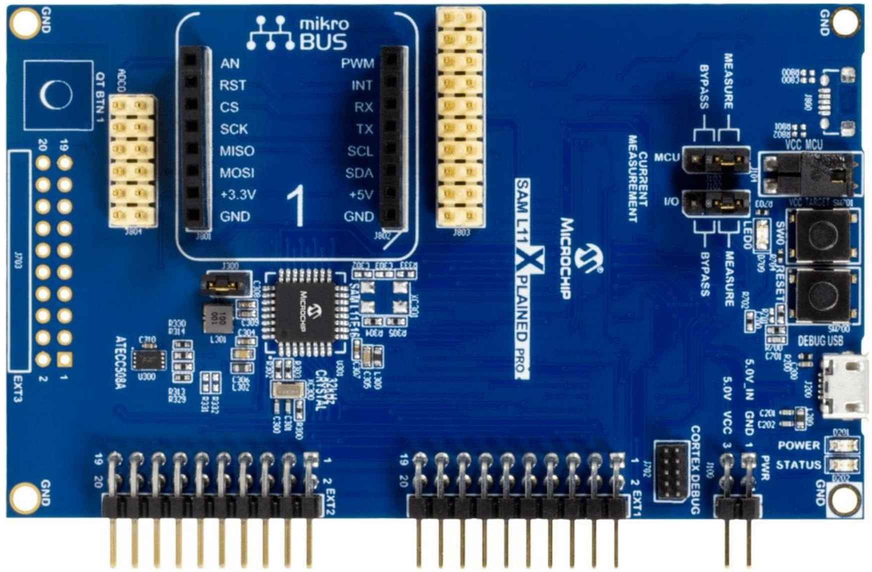
MICROCHIP SAML11

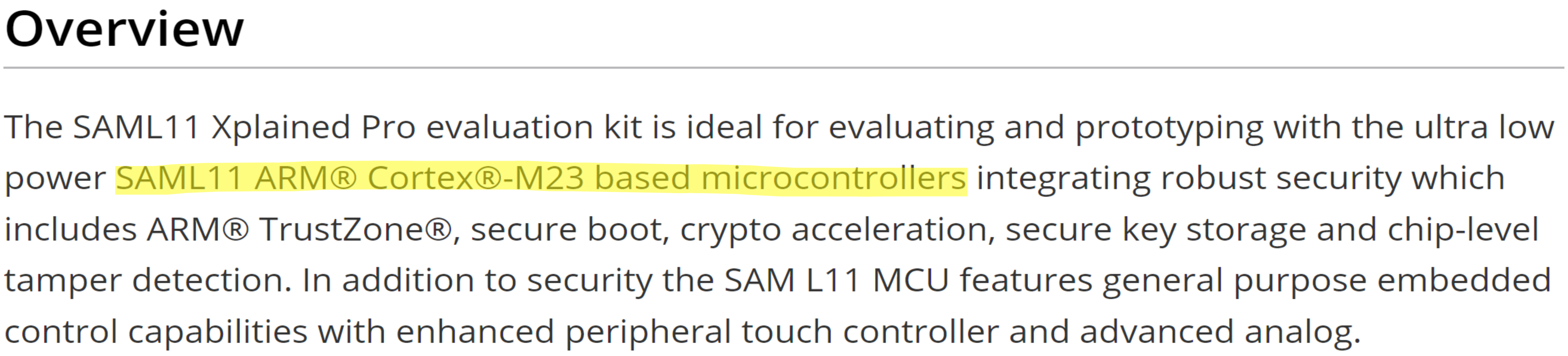


Overview

The SAML11 Xplained Pro evaluation kit is ideal for evaluating and prototyping with the ultra low power SAML11 ARM® Cortex®-M23 based microcontrollers integrating robust security which includes ARM® TrustZone®, secure boot, crypto acceleration, secure key storage and chip-level tamper detection. In addition to security the SAM L11 MCU features general purpose embedded control capabilities with enhanced peripheral touch controller and advanced analog.

MICROCHIP SAML11

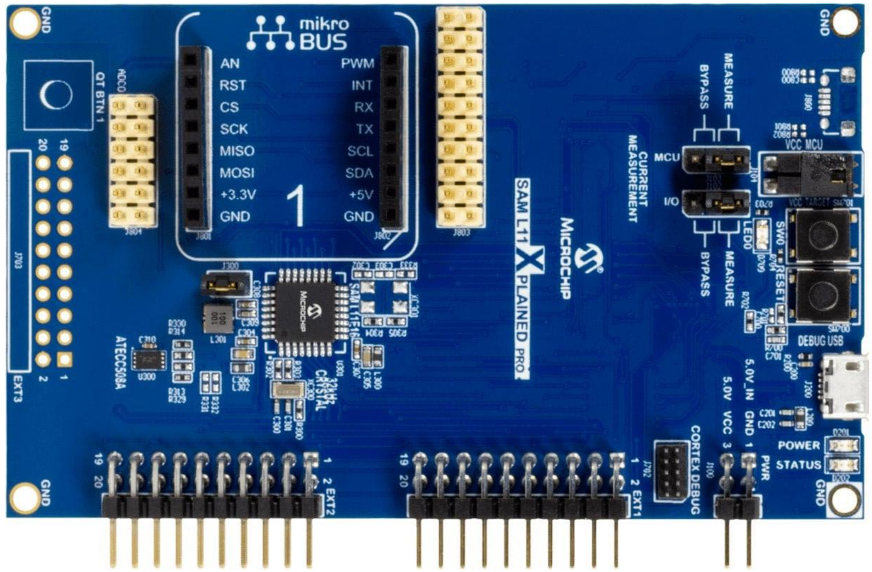




Overview

The SAML11 Xplained Pro evaluation kit is ideal for evaluating and prototyping with the ultra low power SAML11 ARM® Cortex®-M23 based microcontrollers integrating robust security which includes ARM® TrustZone®, secure boot, crypto acceleration, secure key storage and chip-level tamper detection. In addition to security the SAM L11 MCU features general purpose embedded control capabilities with enhanced peripheral touch controller and advanced analog.

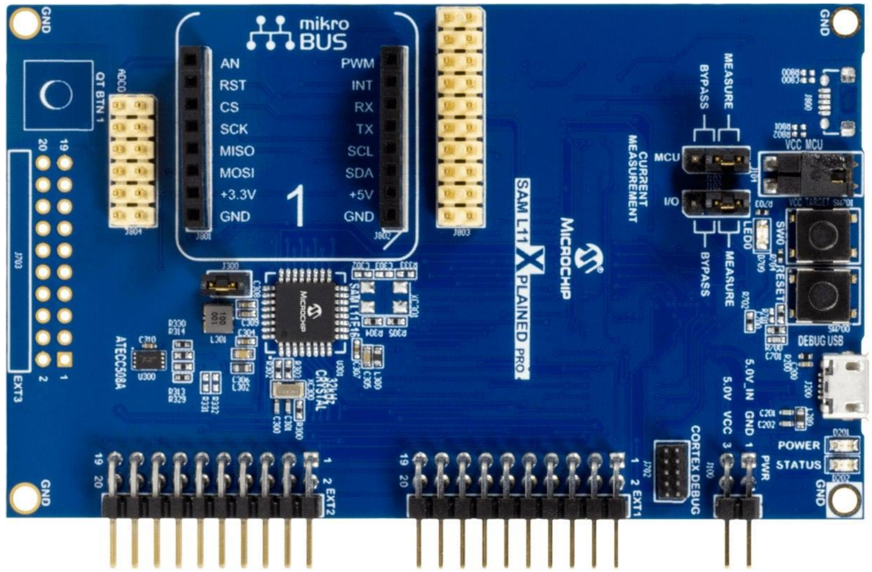
MICROCHIP SAML11

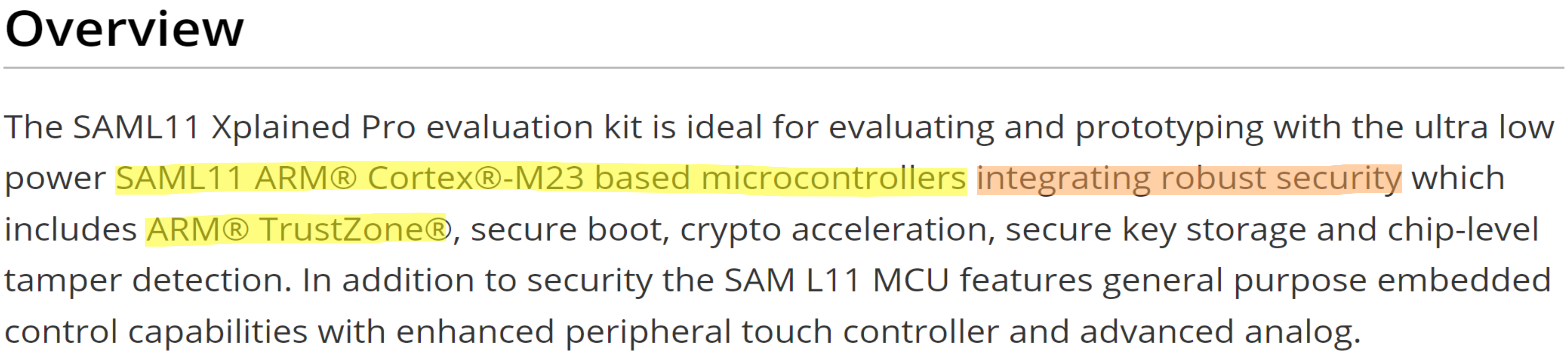


Overview

The SAML11 Xplained Pro evaluation kit is ideal for evaluating and prototyping with the ultra low power SAML11 ARM® Cortex®-M23 based microcontrollers integrating robust security which includes ARM® TrustZone®, secure boot, crypto acceleration, secure key storage and chip-level tamper detection. In addition to security the SAM L11 MCU features general purpose embedded control capabilities with enhanced peripheral touch controller and advanced analog.

MICROCHIP SAML11

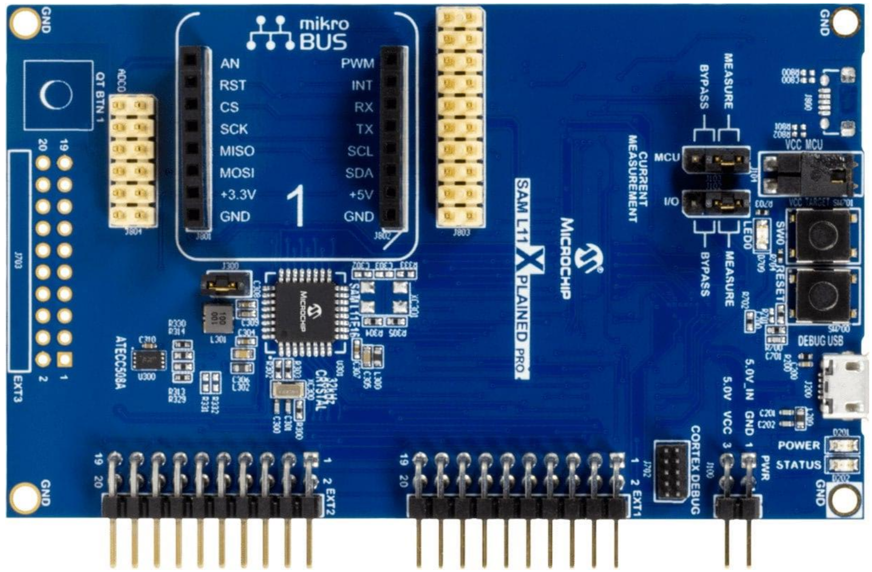


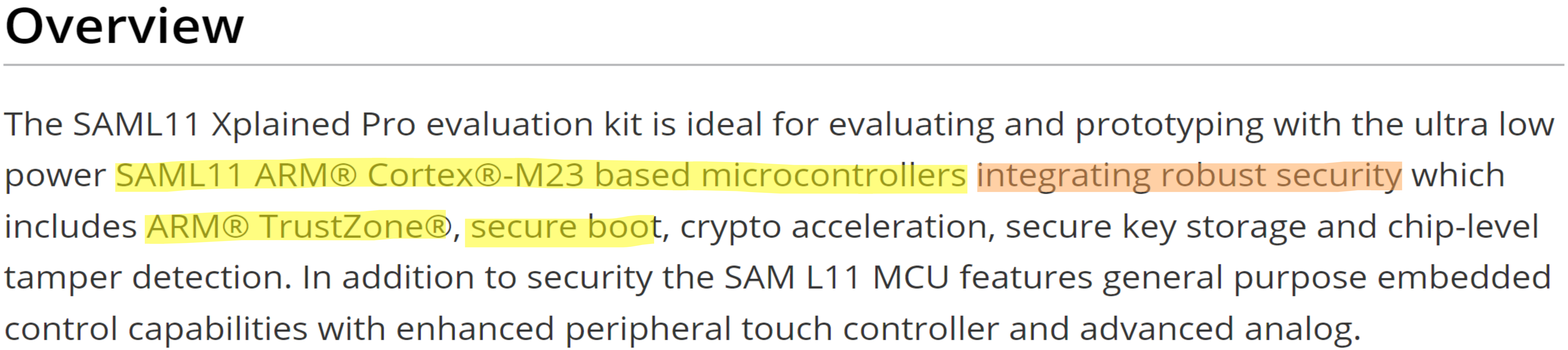


Overview

The SAML11 Xplained Pro evaluation kit is ideal for evaluating and prototyping with the ultra low power SAML11 ARM® Cortex®-M23 based microcontrollers integrating robust security which includes ARM® TrustZone®, secure boot, crypto acceleration, secure key storage and chip-level tamper detection. In addition to security the SAM L11 MCU features general purpose embedded control capabilities with enhanced peripheral touch controller and advanced analog.

MICROCHIP SAML11

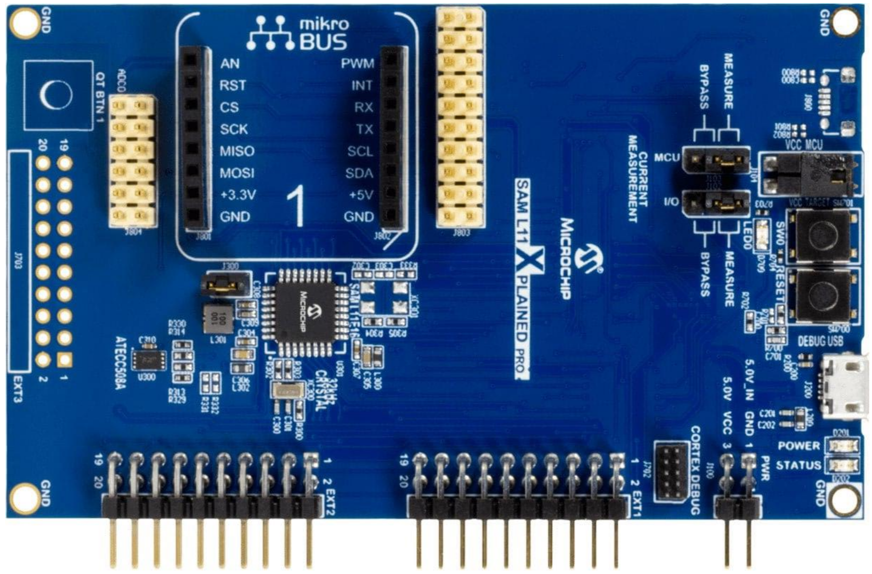


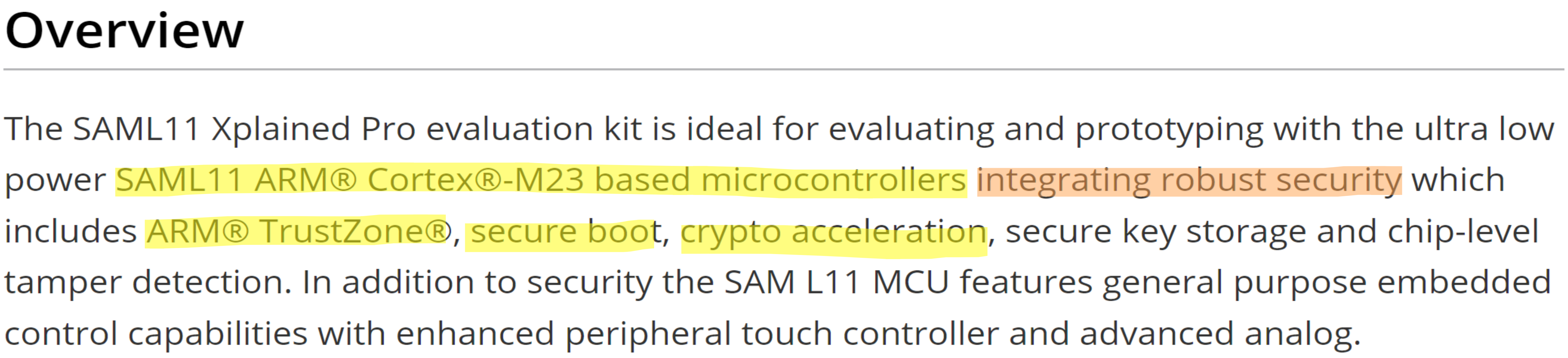


Overview

The SAML11 Xplained Pro evaluation kit is ideal for evaluating and prototyping with the ultra low power SAML11 ARM® Cortex®-M23 based microcontrollers integrating robust security which includes ARM® TrustZone®, secure boot, crypto acceleration, secure key storage and chip-level tamper detection. In addition to security the SAM L11 MCU features general purpose embedded control capabilities with enhanced peripheral touch controller and advanced analog.

MICROCHIP SAML11

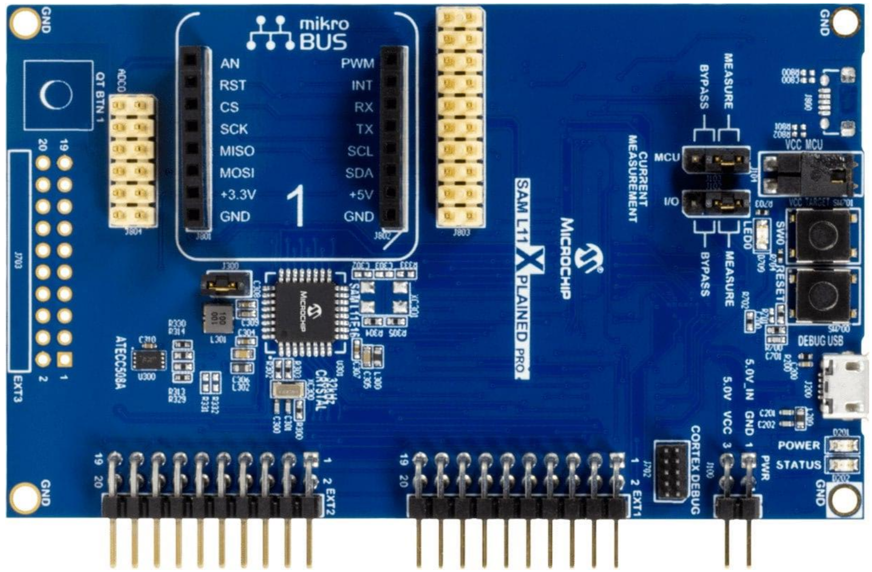




Overview

The SAML11 Xplained Pro evaluation kit is ideal for evaluating and prototyping with the ultra low power SAML11 ARM® Cortex®-M23 based microcontrollers integrating robust security which includes ARM® TrustZone®, secure boot, crypto acceleration, secure key storage and chip-level tamper detection. In addition to security the SAM L11 MCU features general purpose embedded control capabilities with enhanced peripheral touch controller and advanced analog.

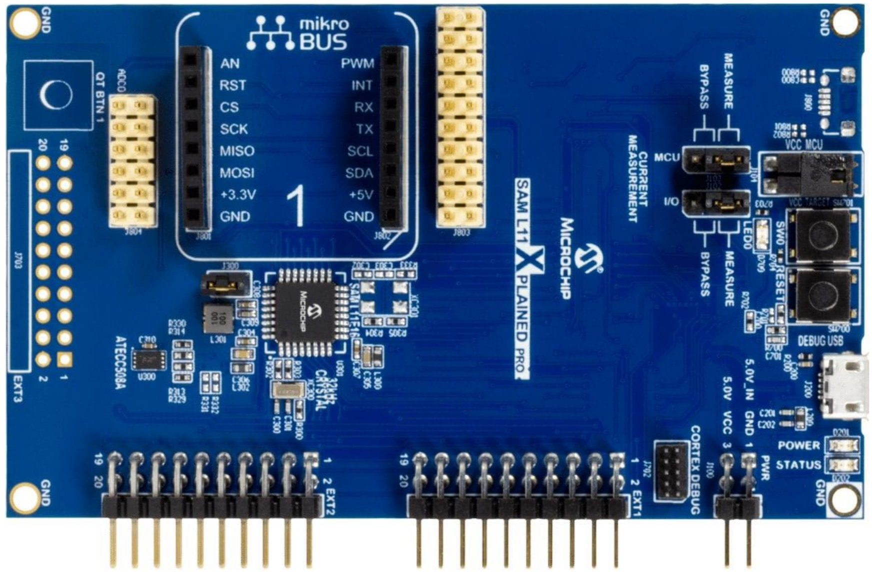
MICROCHIP SAML11

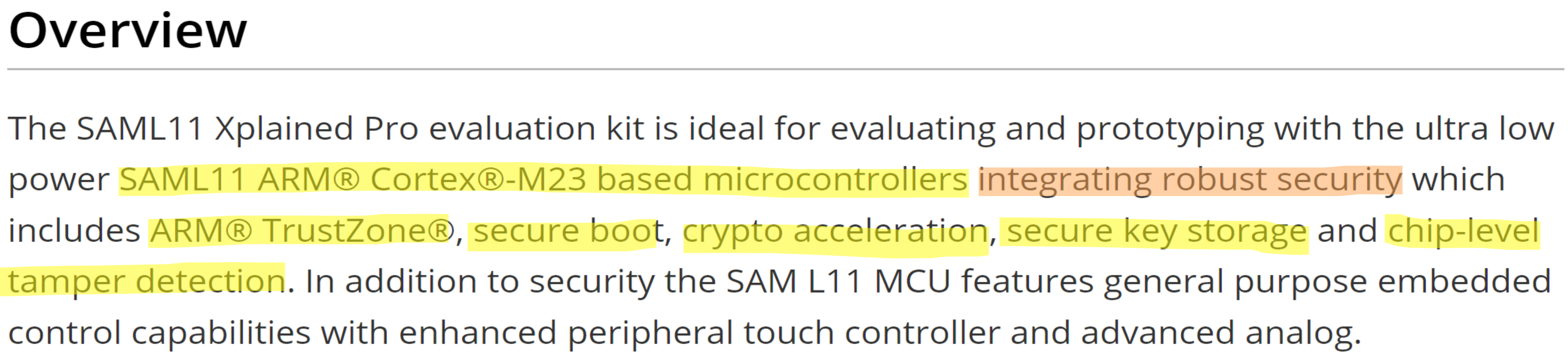


Overview

The SAML11 Xplained Pro evaluation kit is ideal for evaluating and prototyping with the ultra low power SAML11 ARM® Cortex®-M23 based microcontrollers integrating robust security which includes ARM® TrustZone®, secure boot, crypto acceleration, secure key storage and chip-level tamper detection. In addition to security the SAM L11 MCU features general purpose embedded control capabilities with enhanced peripheral touch controller and advanced analog.

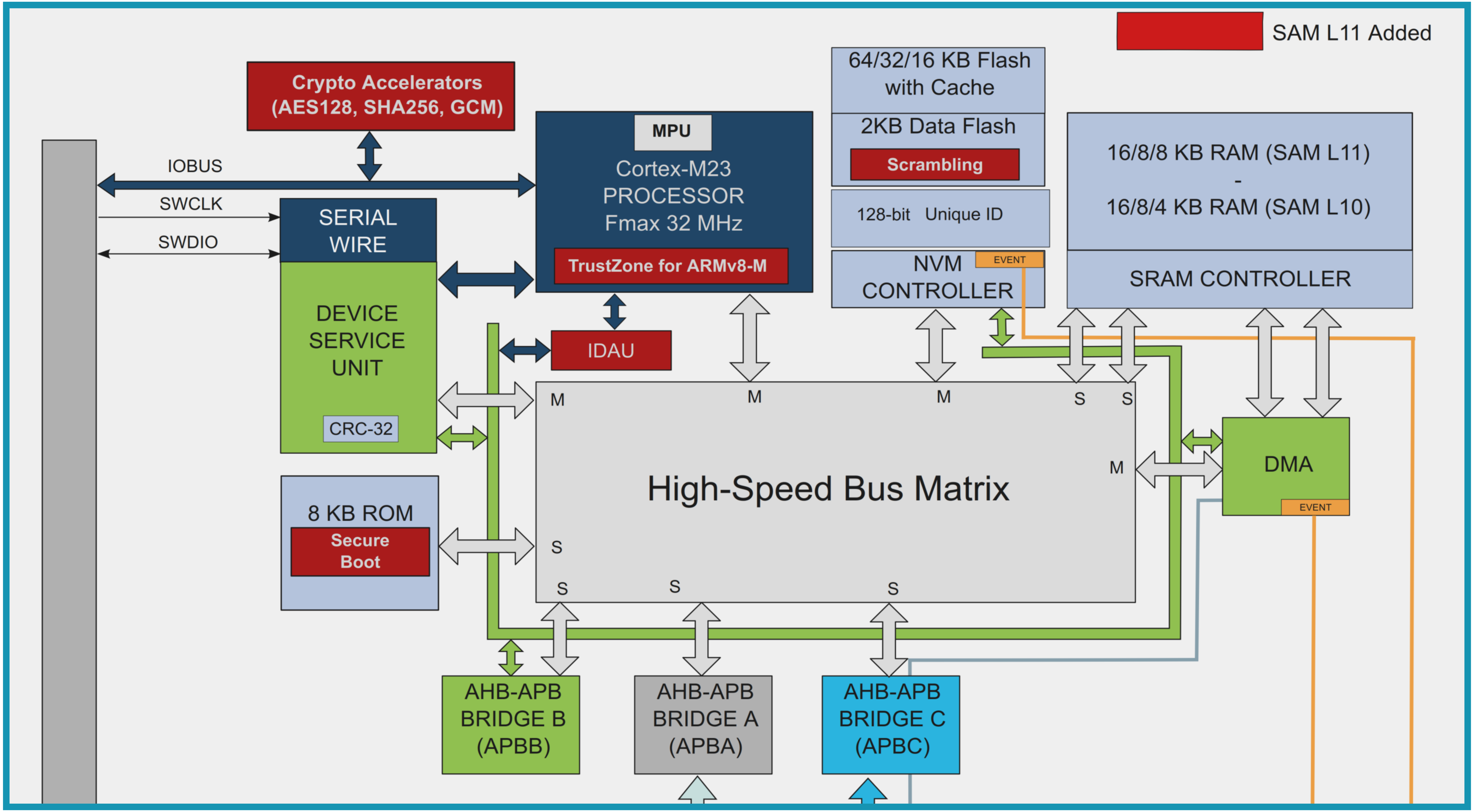
MICROCHIP SAML11



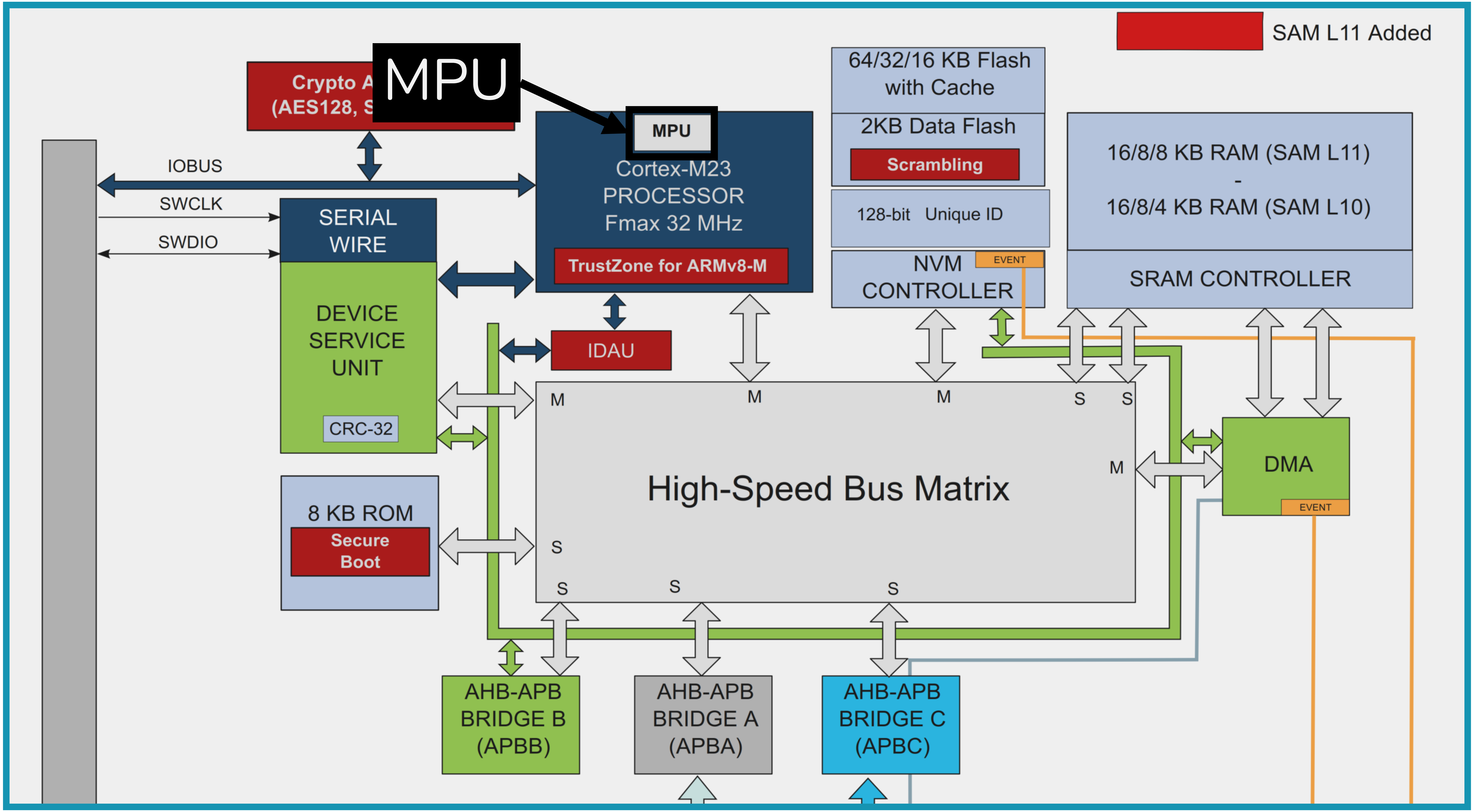


Overview

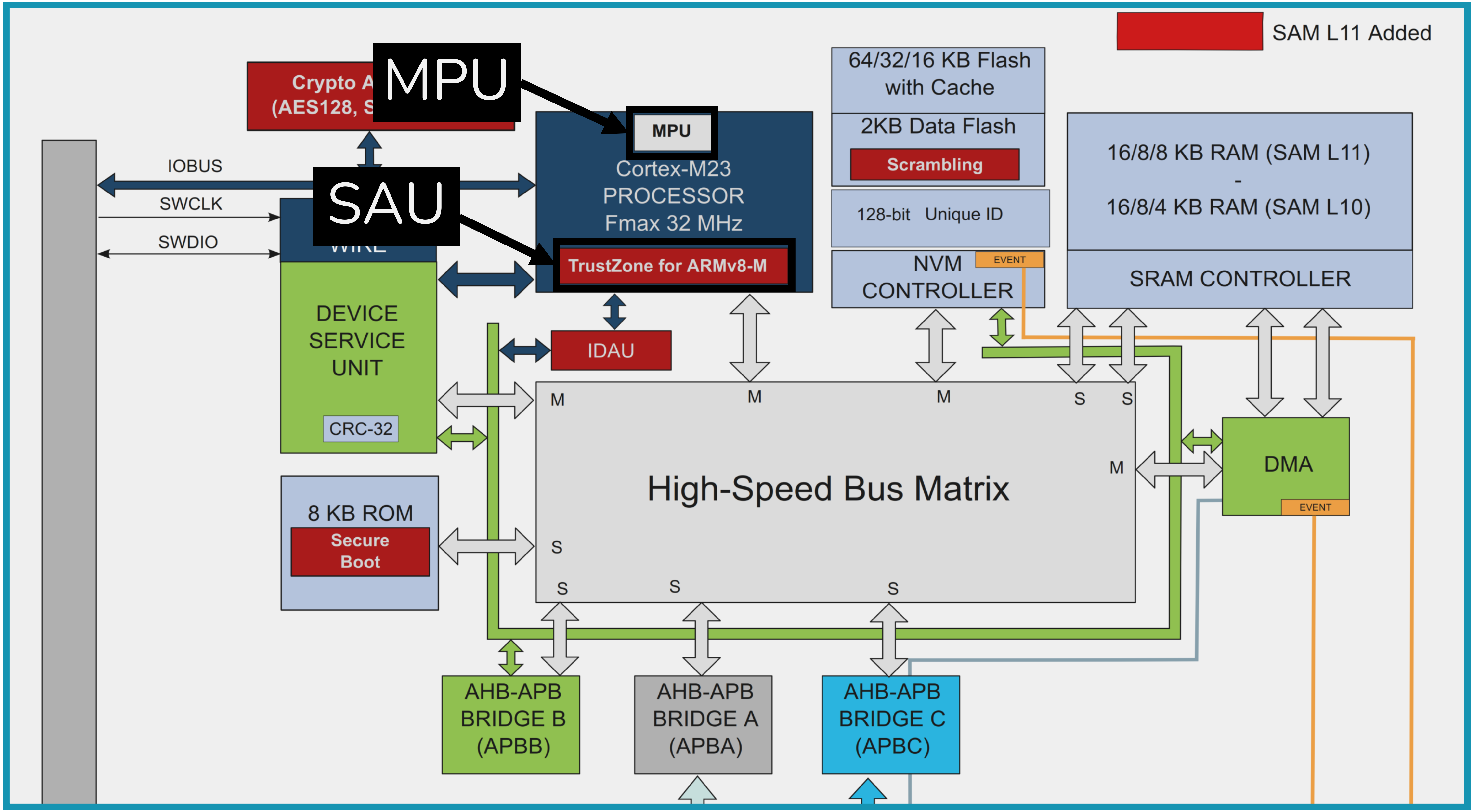
The SAML11 Xplained Pro evaluation kit is ideal for evaluating and prototyping with the ultra low power SAML11 ARM® Cortex®-M23 based microcontrollers integrating robust security which includes ARM® TrustZone®, secure boot, crypto acceleration, secure key storage and chip-level tamper detection. In addition to security the SAM L11 MCU features general purpose embedded control capabilities with enhanced peripheral touch controller and advanced analog.



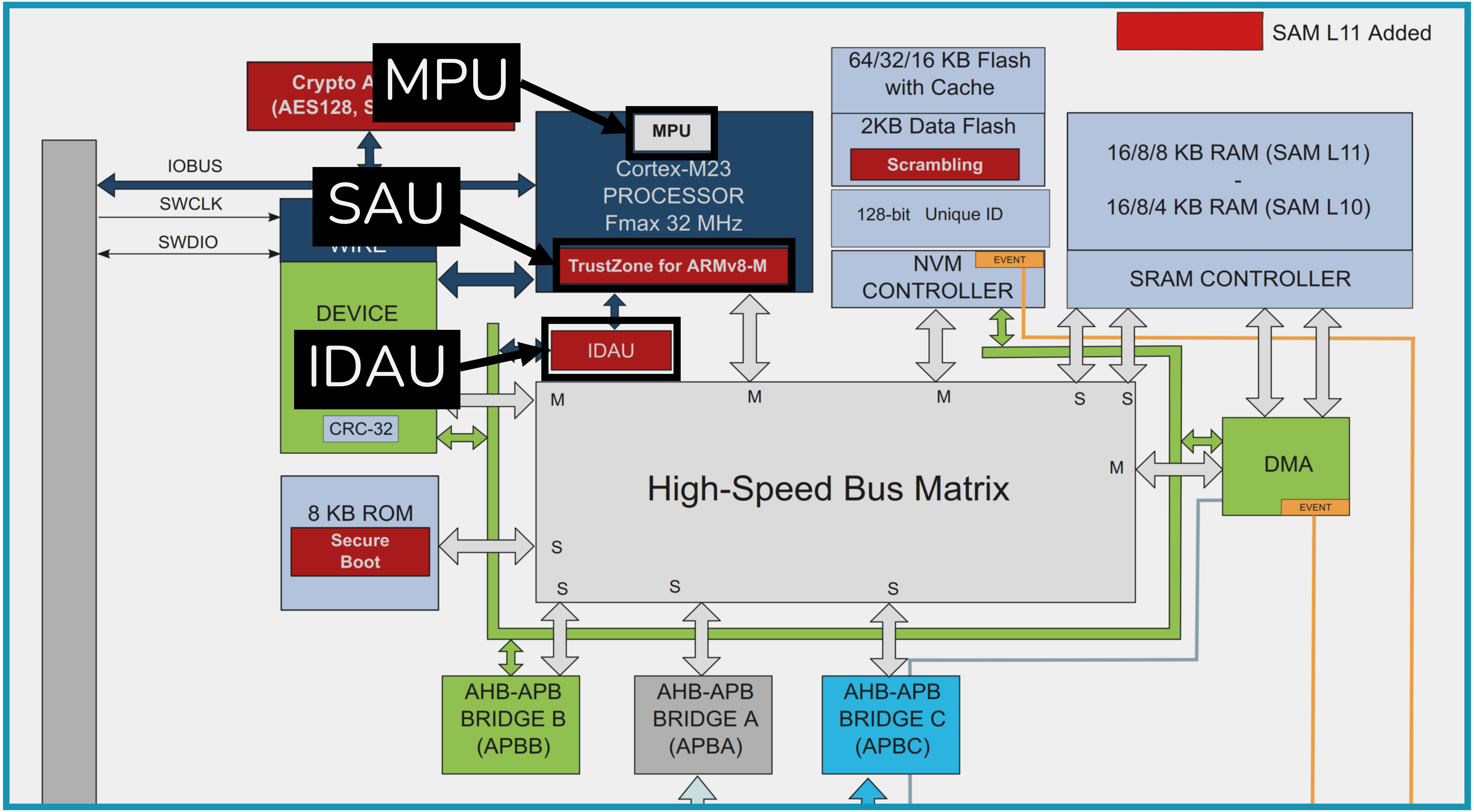
Pag. 17 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



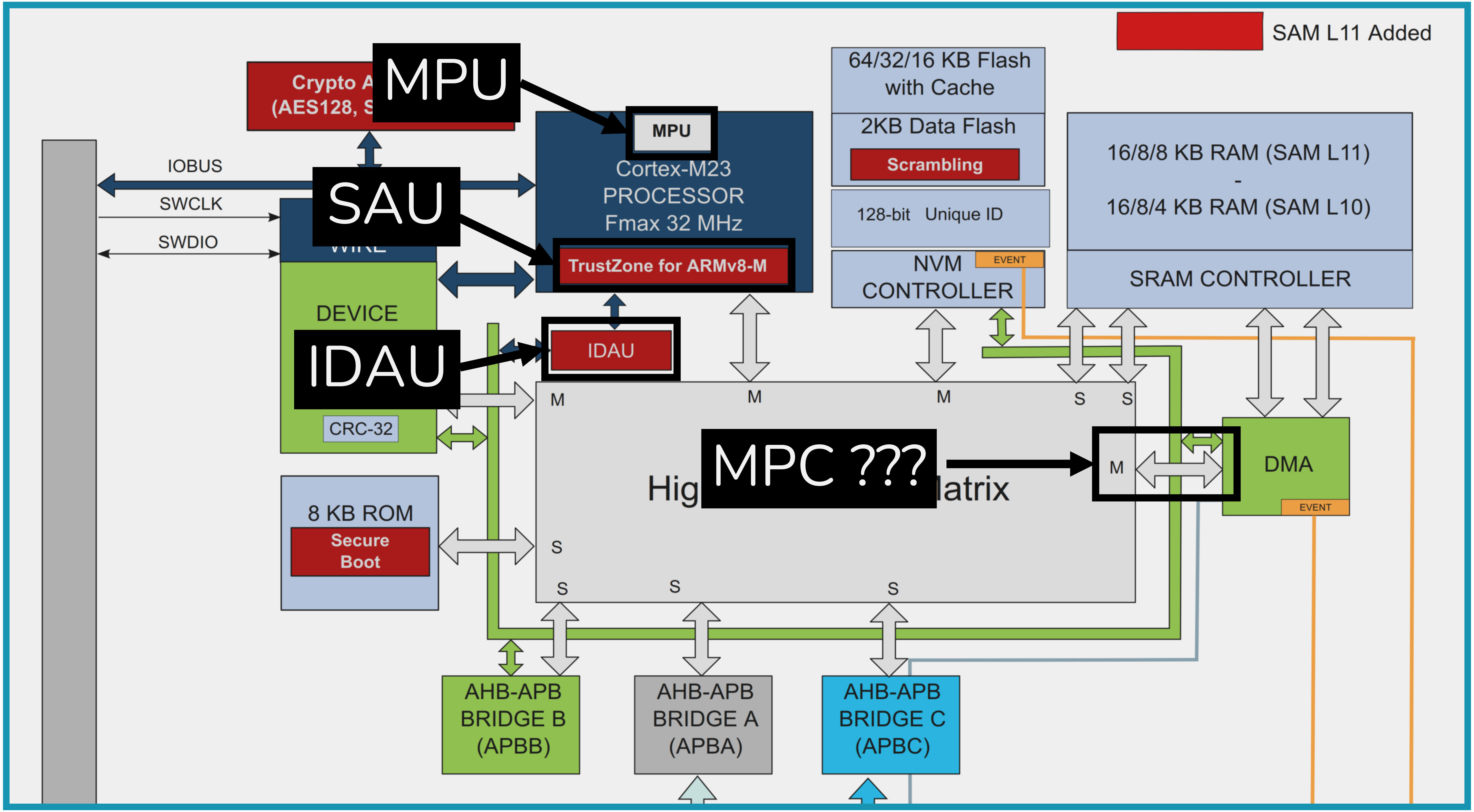
Pag. 17 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



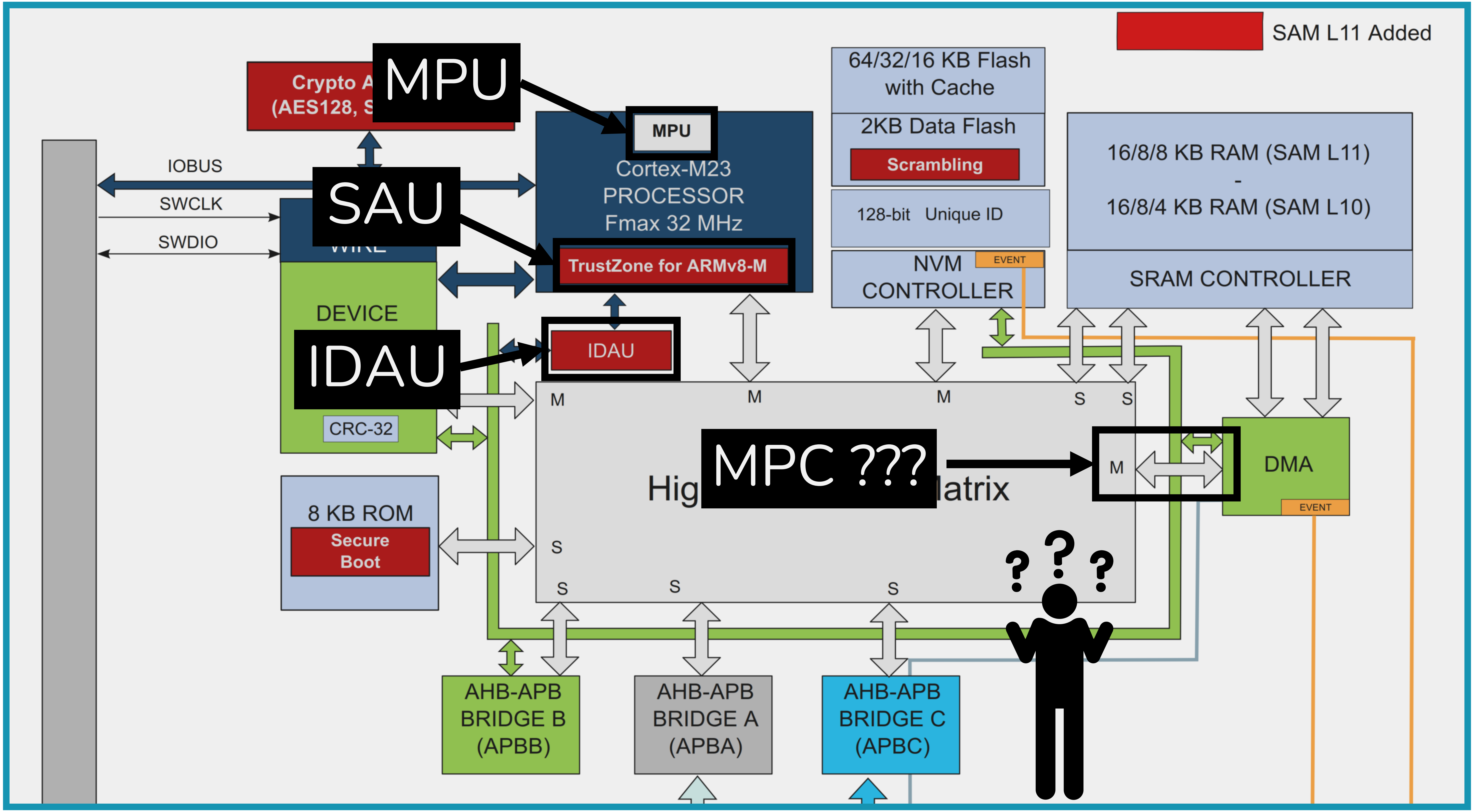
Pag. 17 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



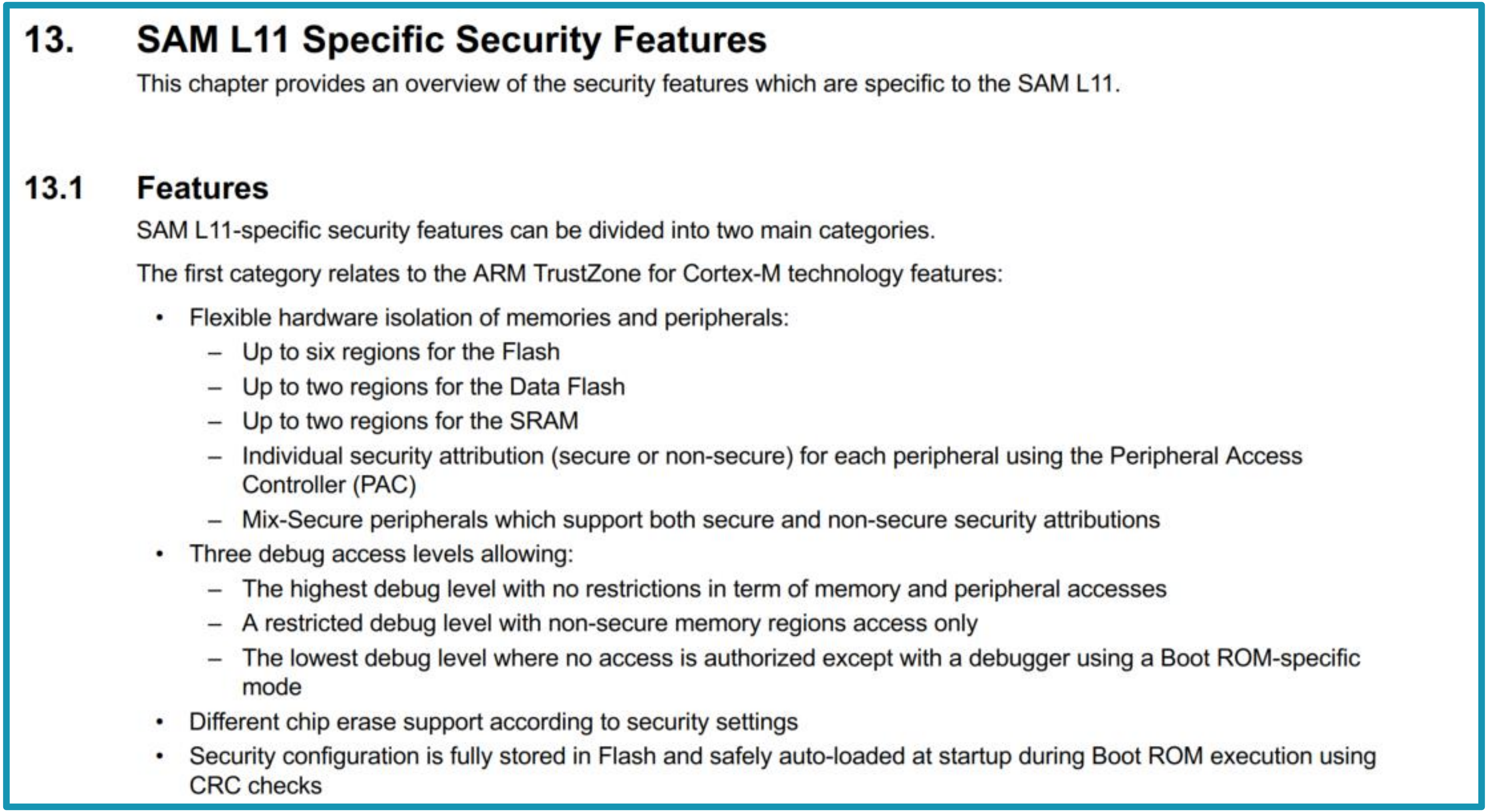
Pag. 17 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



Pag. 17 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



Pag. 17 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



13. SAM L11 Specific Security Features

This chapter provides an overview of the security features which are specific to the SAM L11.

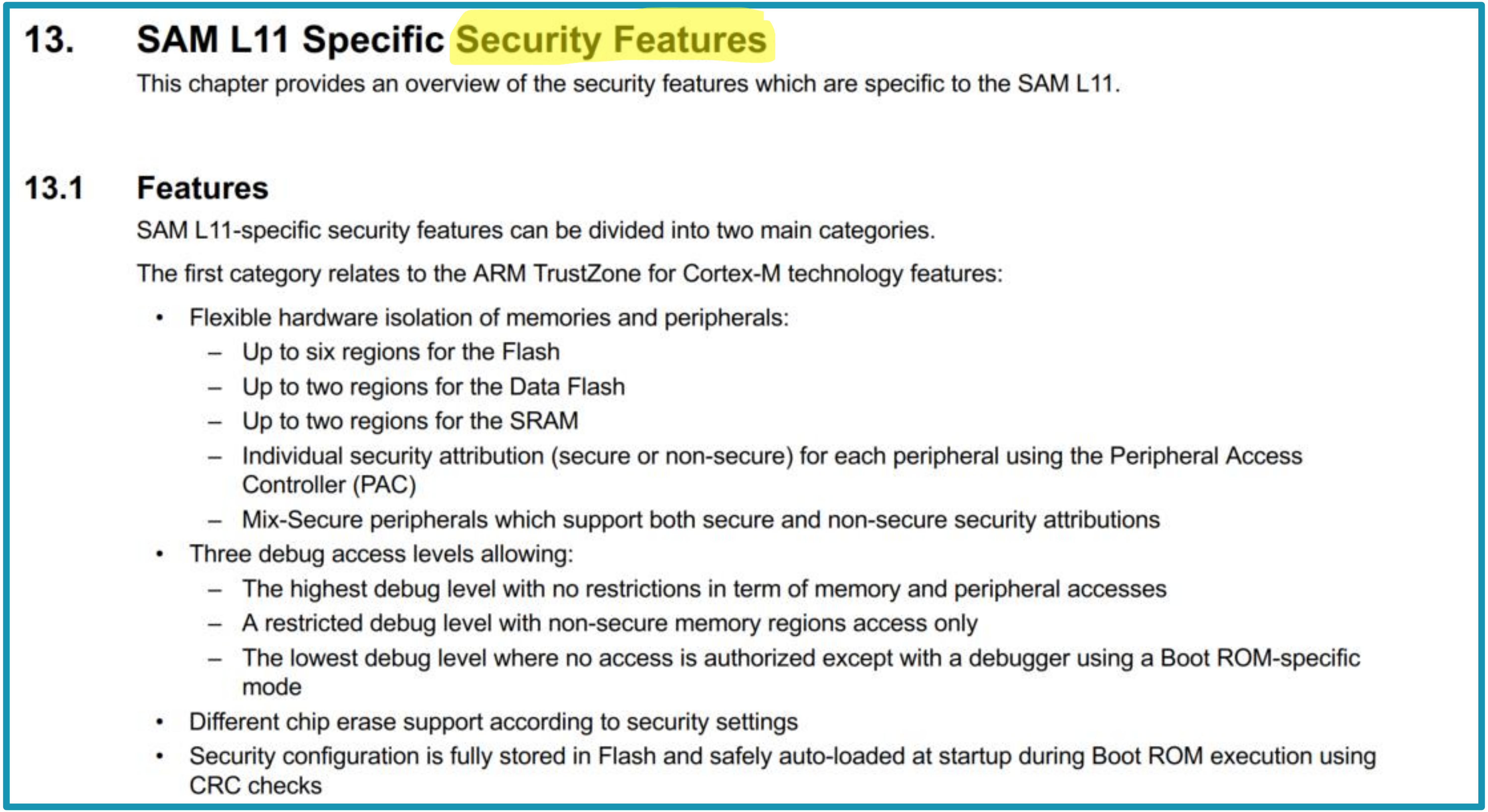
13.1 Features

SAM L11-specific security features can be divided into two main categories.

The first category relates to the ARM TrustZone for Cortex-M technology features:

- Flexible hardware isolation of memories and peripherals:
  - Up to six regions for the Flash
  - Up to two regions for the Data Flash
  - Up to two regions for the SRAM
  - Individual security attribution (secure or non-secure) for each peripheral using the Peripheral Access Controller (PAC)
  - Mix-Secure peripherals which support both secure and non-secure security attributions
- Three debug access levels allowing:
  - The highest debug level with no restrictions in term of memory and peripheral accesses
  - A restricted debug level with non-secure memory regions access only
  - The lowest debug level where no access is authorized except with a debugger using a Boot ROM-specific mode
- Different chip erase support according to security settings
- Security configuration is fully stored in Flash and safely auto-loaded at startup during Boot ROM execution using CRC checks

Pag. 53 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



13. SAM L11 Specific Security Features

This chapter provides an overview of the security features which are specific to the SAM L11.

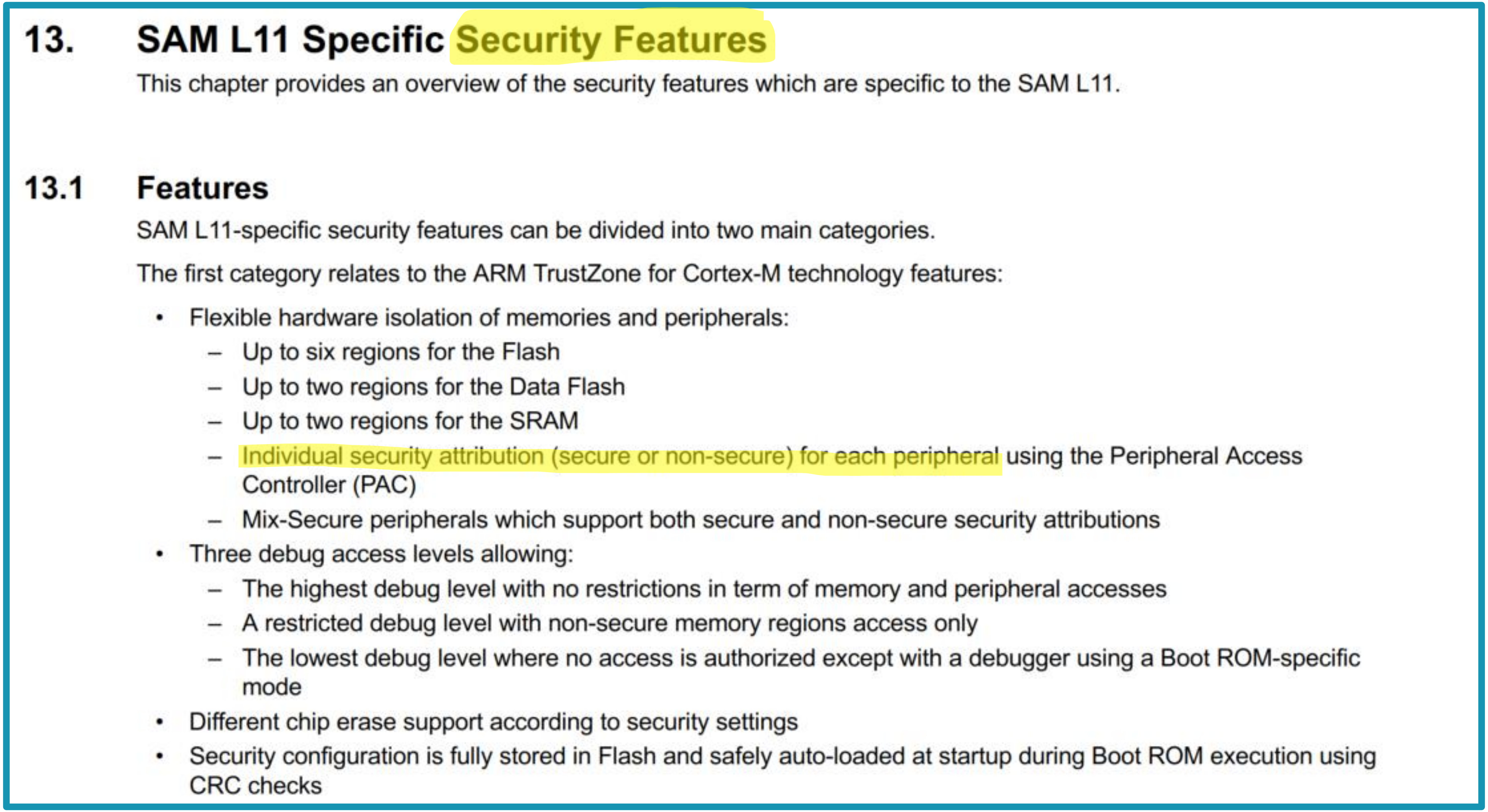
13.1 Features

SAM L11-specific security features can be divided into two main categories.

The first category relates to the ARM TrustZone for Cortex-M technology features:

- Flexible hardware isolation of memories and peripherals:
  - Up to six regions for the Flash
  - Up to two regions for the Data Flash
  - Up to two regions for the SRAM
  - Individual security attribution (secure or non-secure) for each peripheral using the Peripheral Access Controller (PAC)
  - Mix-Secure peripherals which support both secure and non-secure security attributions
- Three debug access levels allowing:
  - The highest debug level with no restrictions in term of memory and peripheral accesses
  - A restricted debug level with non-secure memory regions access only
  - The lowest debug level where no access is authorized except with a debugger using a Boot ROM-specific mode
- Different chip erase support according to security settings
- Security configuration is fully stored in Flash and safely auto-loaded at startup during Boot ROM execution using CRC checks

Pag. 53 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



13. SAM L11 Specific Security Features

This chapter provides an overview of the security features which are specific to the SAM L11.

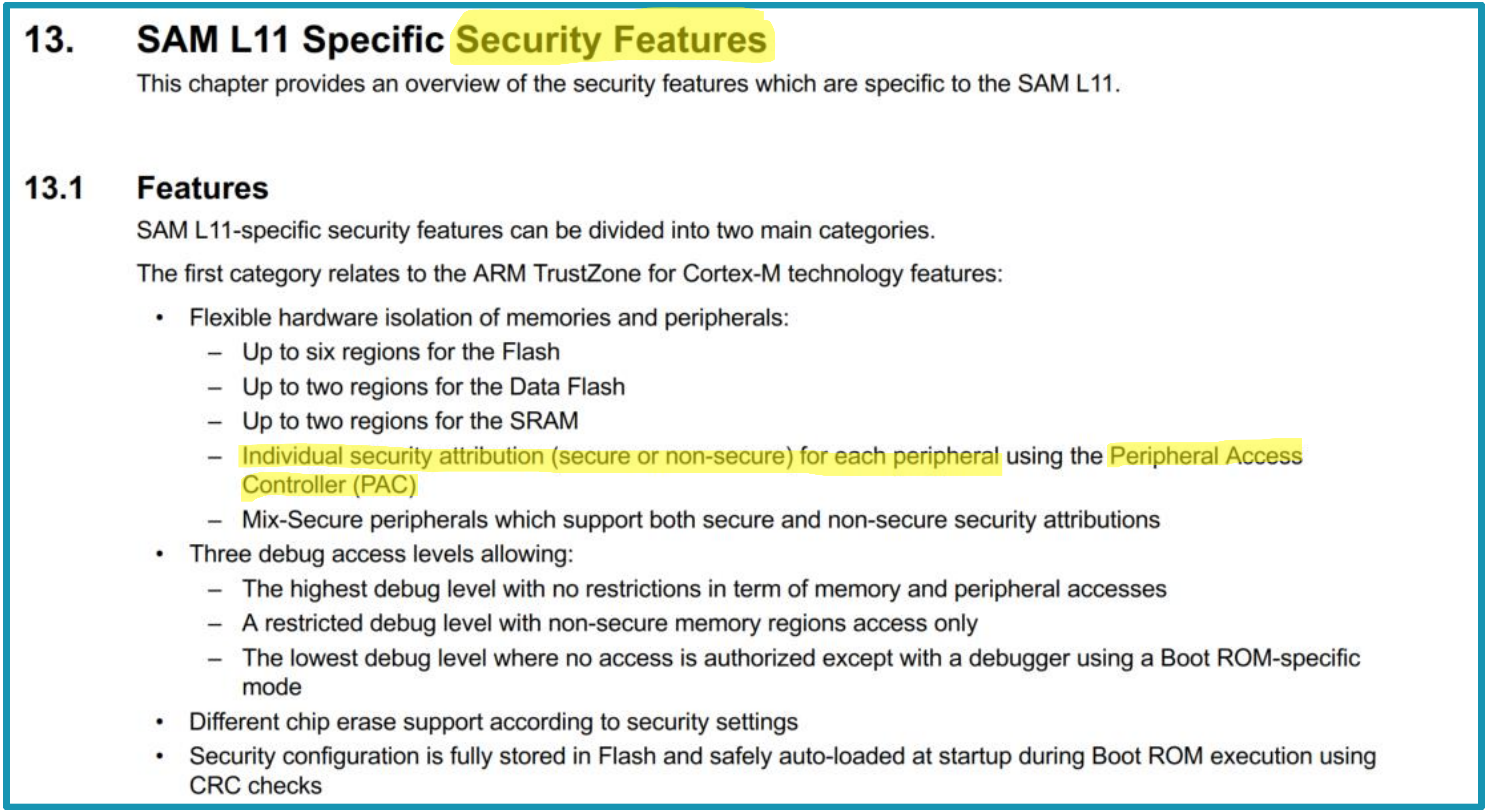
13.1 Features

SAM L11-specific security features can be divided into two main categories.

The first category relates to the ARM TrustZone for Cortex-M technology features:

- Flexible hardware isolation of memories and peripherals:
  - Up to six regions for the Flash
  - Up to two regions for the Data Flash
  - Up to two regions for the SRAM
  - Individual security attribution (secure or non-secure) for each peripheral using the Peripheral Access Controller (PAC)
  - Mix-Secure peripherals which support both secure and non-secure security attributions
- Three debug access levels allowing:
  - The highest debug level with no restrictions in term of memory and peripheral accesses
  - A restricted debug level with non-secure memory regions access only
  - The lowest debug level where no access is authorized except with a debugger using a Boot ROM-specific mode
- Different chip erase support according to security settings
- Security configuration is fully stored in Flash and safely auto-loaded at startup during Boot ROM execution using CRC checks

Pag. 53 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



13. SAM L11 Specific Security Features

This chapter provides an overview of the security features which are specific to the SAM L11.

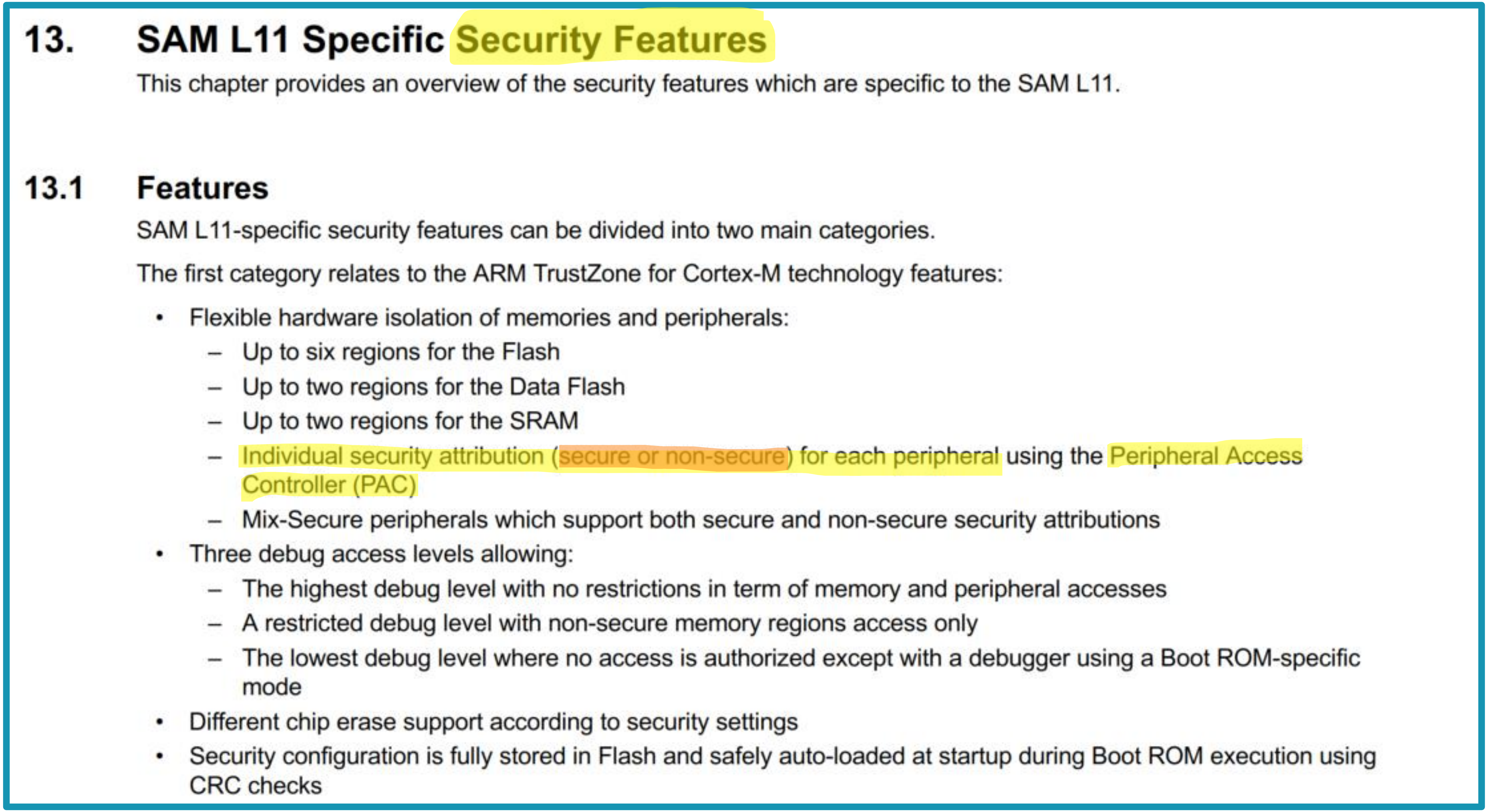
13.1 Features

SAM L11-specific security features can be divided into two main categories.

The first category relates to the ARM TrustZone for Cortex-M technology features:

- Flexible hardware isolation of memories and peripherals:
  - Up to six regions for the Flash
  - Up to two regions for the Data Flash
  - Up to two regions for the SRAM
  - Individual security attribution (secure or non-secure) for each peripheral using the Peripheral Access Controller (PAC)
  - Mix-Secure peripherals which support both secure and non-secure security attributions
- Three debug access levels allowing:
  - The highest debug level with no restrictions in term of memory and peripheral accesses
  - A restricted debug level with non-secure memory regions access only
  - The lowest debug level where no access is authorized except with a debugger using a Boot ROM-specific mode
- Different chip erase support according to security settings
- Security configuration is fully stored in Flash and safely auto-loaded at startup during Boot ROM execution using CRC checks

Pag. 53 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



13. SAM L11 Specific Security Features

This chapter provides an overview of the security features which are specific to the SAM L11.

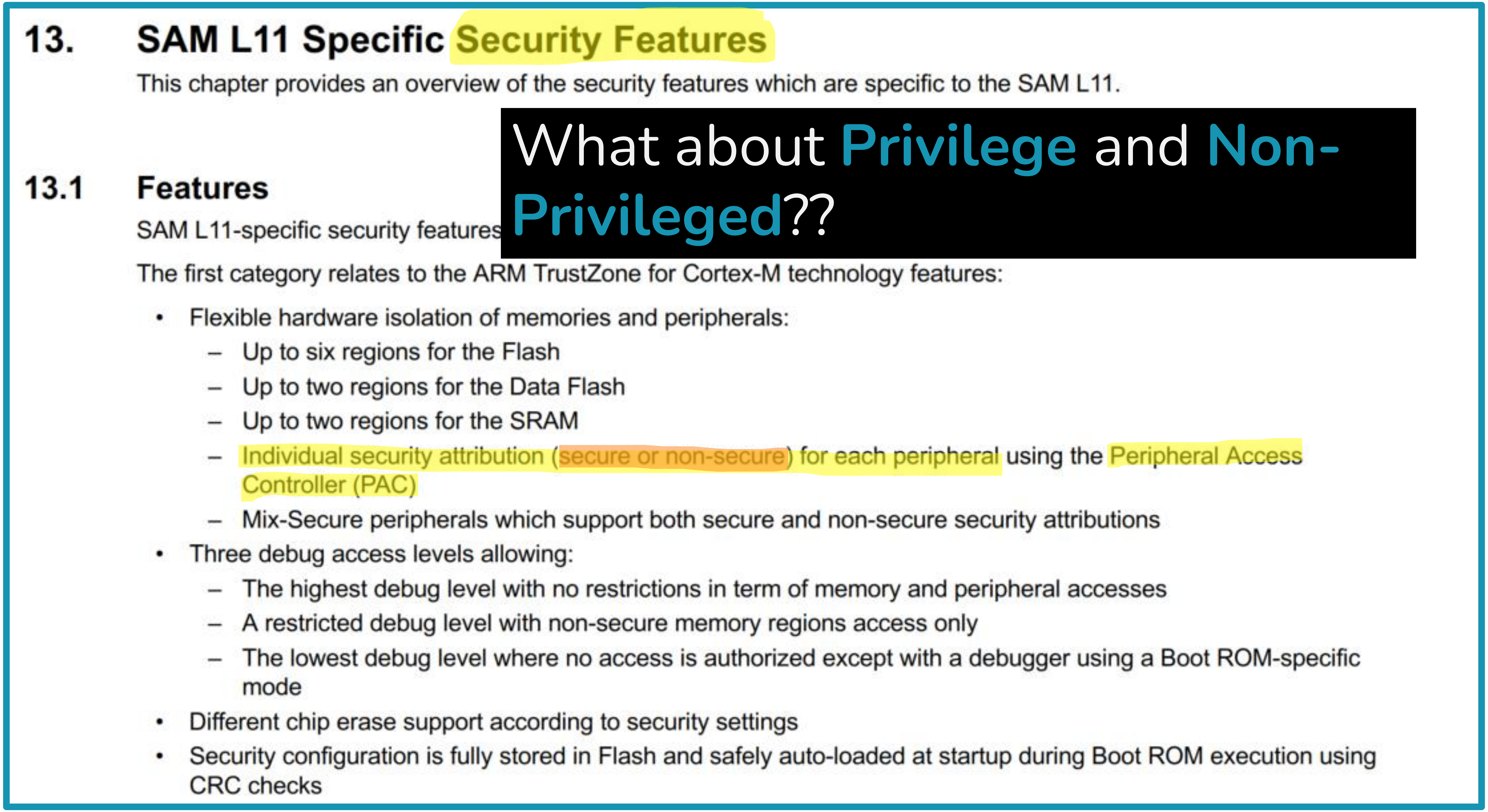
13.1 Features

SAM L11-specific security features can be divided into two main categories.

The first category relates to the ARM TrustZone for Cortex-M technology features:

- Flexible hardware isolation of memories and peripherals:
  - Up to six regions for the Flash
  - Up to two regions for the Data Flash
  - Up to two regions for the SRAM
  - Individual security attribution (secure or non-secure) for each peripheral using the Peripheral Access Controller (PAC)
  - Mix-Secure peripherals which support both secure and non-secure security attributions
- Three debug access levels allowing:
  - The highest debug level with no restrictions in term of memory and peripheral accesses
  - A restricted debug level with non-secure memory regions access only
  - The lowest debug level where no access is authorized except with a debugger using a Boot ROM-specific mode
- Different chip erase support according to security settings
- Security configuration is fully stored in Flash and safely auto-loaded at startup during Boot ROM execution using CRC checks

Pag. 53 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



13. SAM L11 Specific Security Features

This chapter provides an overview of the security features which are specific to the SAM L11.

What about **Privilege** and **Non-Privileged**??

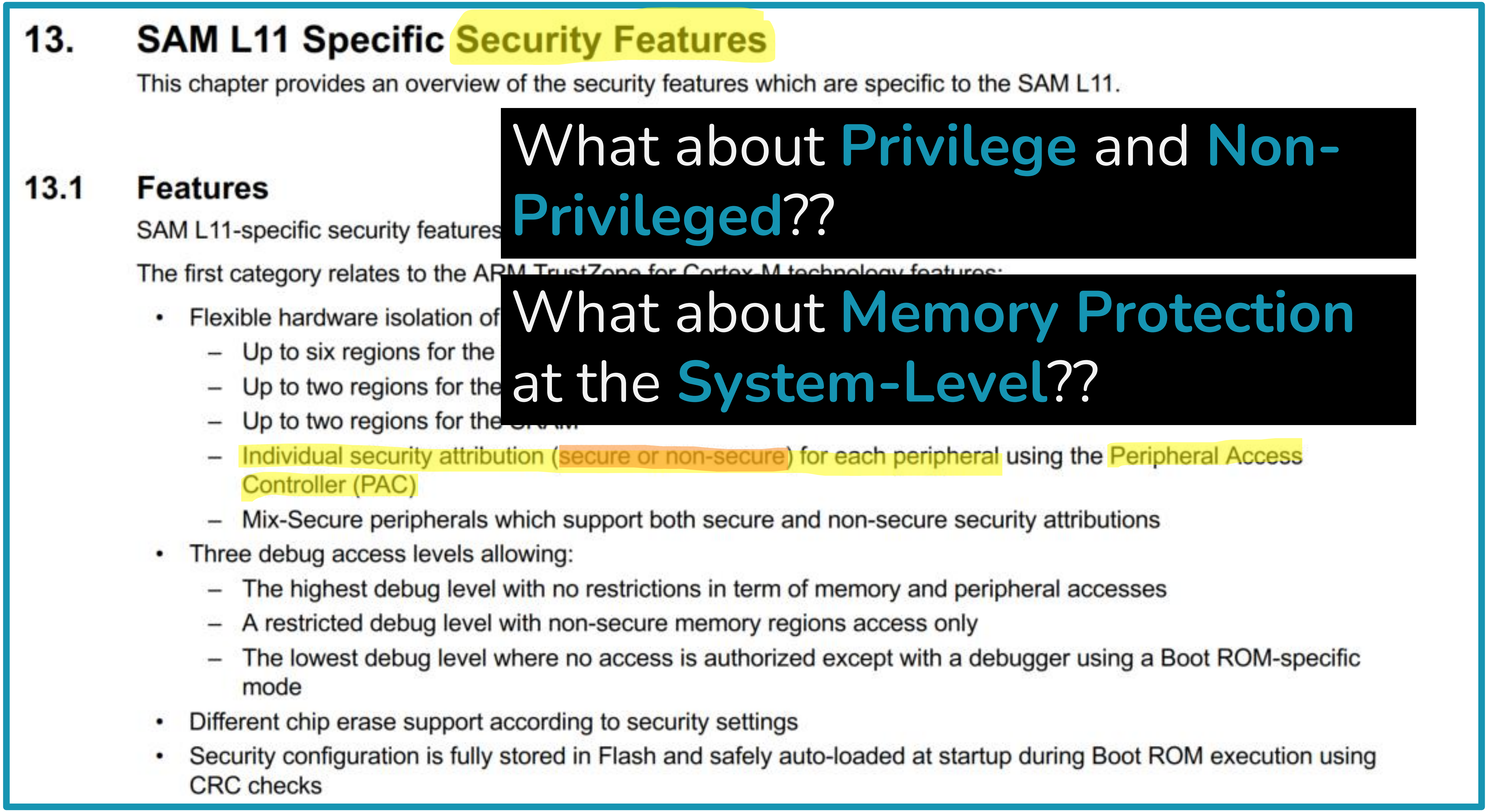
13.1 Features

SAM L11-specific security features

The first category relates to the ARM TrustZone for Cortex-M technology features:

- Flexible hardware isolation of memories and peripherals:
  - Up to six regions for the Flash
  - Up to two regions for the Data Flash
  - Up to two regions for the SRAM
  - Individual security attribution (secure or non-secure) for each peripheral using the Peripheral Access Controller (PAC)
  - Mix-Secure peripherals which support both secure and non-secure security attributions
- Three debug access levels allowing:
  - The highest debug level with no restrictions in term of memory and peripheral accesses
  - A restricted debug level with non-secure memory regions access only
  - The lowest debug level where no access is authorized except with a debugger using a Boot ROM-specific mode
- Different chip erase support according to security settings
- Security configuration is fully stored in Flash and safely auto-loaded at startup during Boot ROM execution using CRC checks

Pag. 53 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



13. SAM L11 Specific Security Features

This chapter provides an overview of the security features which are specific to the SAM L11.

13.1 Features

SAM L11-specific security features

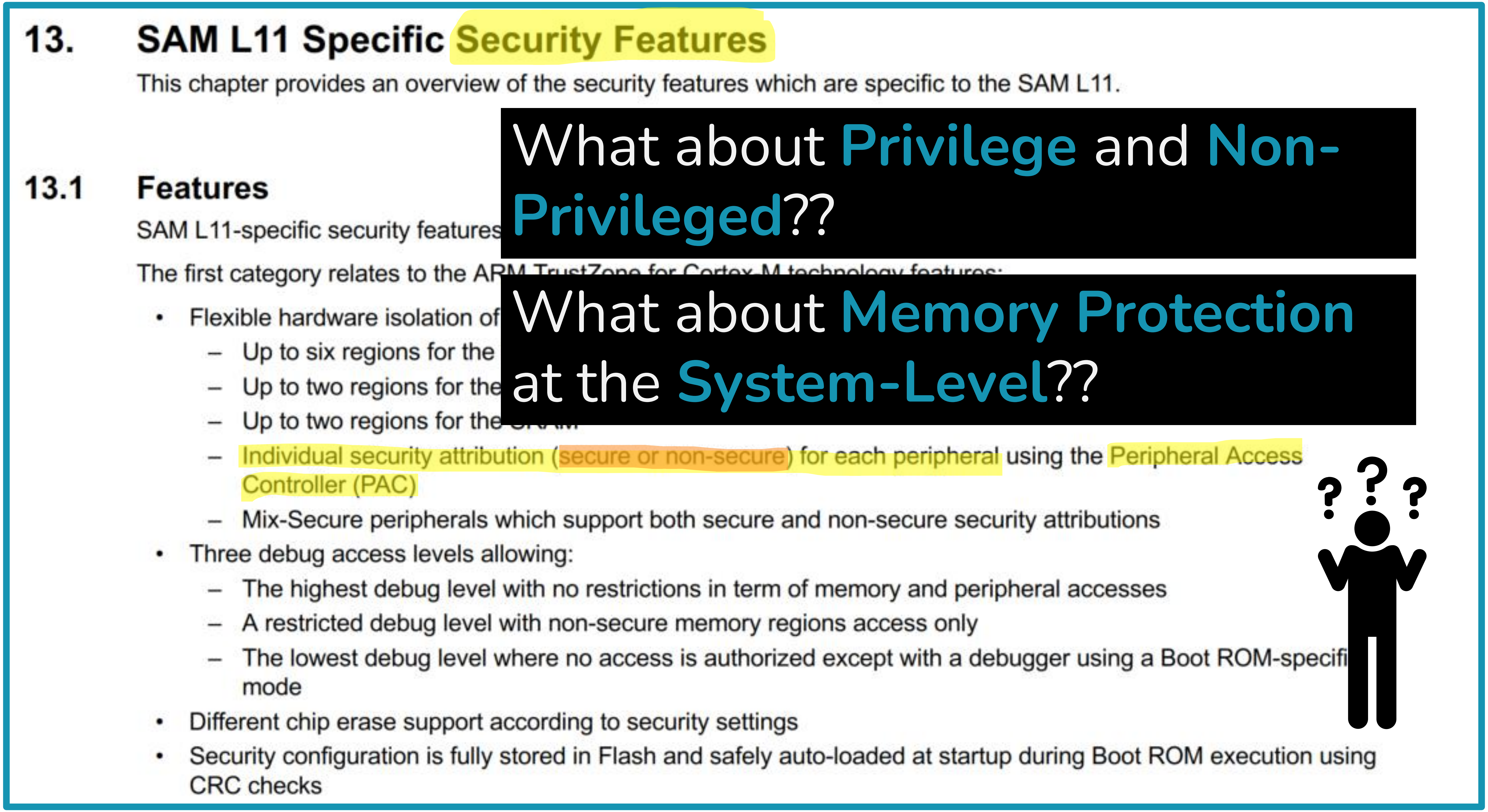
The first category relates to the ARM TrustZone for Cortex-M technology features:

- Flexible hardware isolation of
  - Up to six regions for the
  - Up to two regions for the
  - Up to two regions for the SRAM
  - Individual security attribution (secure or non-secure) for each peripheral using the Peripheral Access Controller (PAC)
  - Mix-Secure peripherals which support both secure and non-secure security attributions
- Three debug access levels allowing:
  - The highest debug level with no restrictions in term of memory and peripheral accesses
  - A restricted debug level with non-secure memory regions access only
  - The lowest debug level where no access is authorized except with a debugger using a Boot ROM-specific mode
- Different chip erase support according to security settings
- Security configuration is fully stored in Flash and safely auto-loaded at startup during Boot ROM execution using CRC checks

What about **Privilege** and **Non-Privileged**??

What about **Memory Protection** at the **System-Level**??

Pag. 53 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.



13. SAM L11 Specific Security Features

This chapter provides an overview of the security features which are specific to the SAM L11.

13.1 Features

SAM L11-specific security features

The first category relates to the ARM TrustZone for Cortex-M technology features:

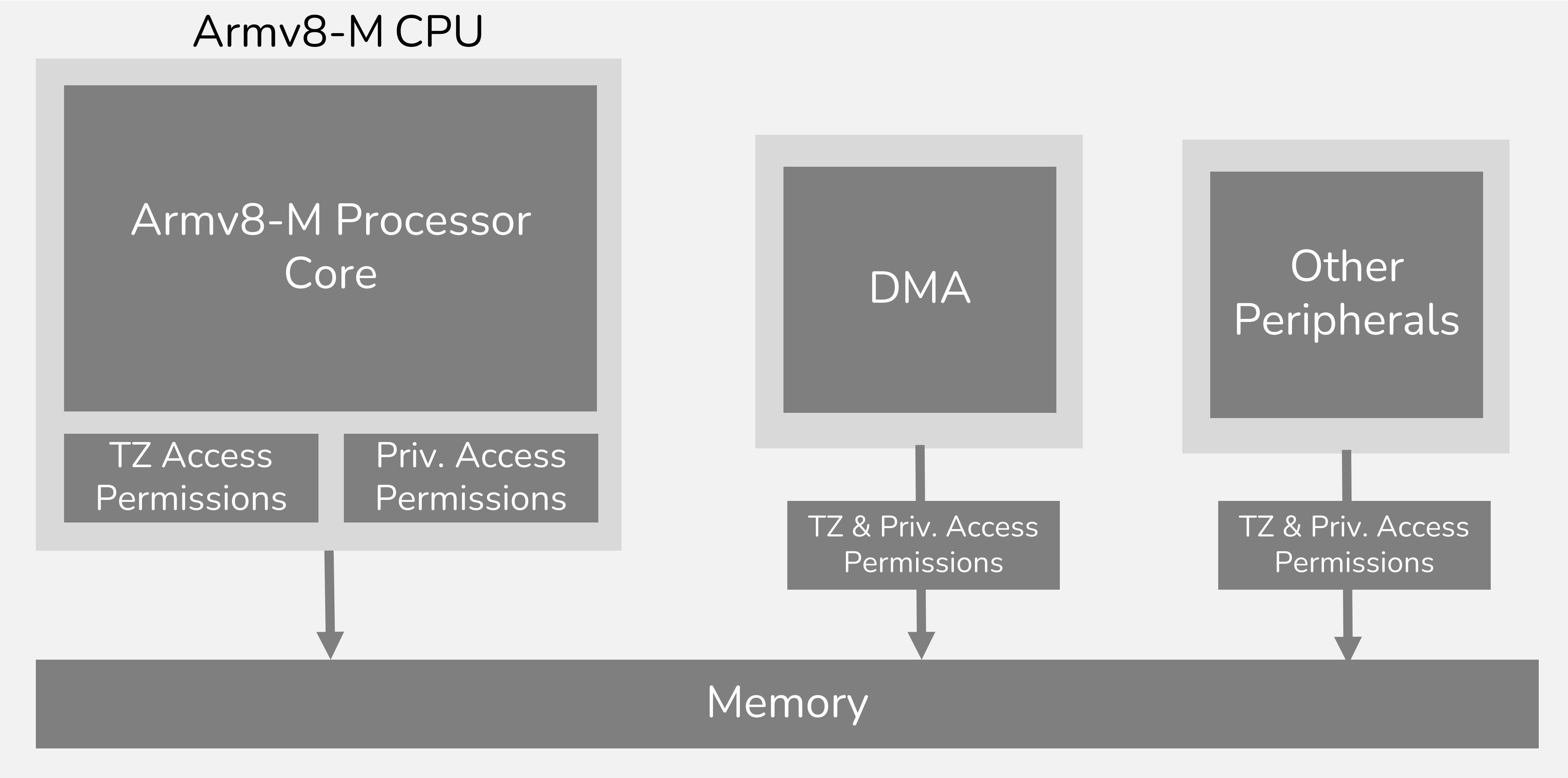
- Flexible hardware isolation of
  - Up to six regions for the
  - Up to two regions for the
  - Up to two regions for the SRAM
  - Individual security attribution (secure or non-secure) for each peripheral using the Peripheral Access Controller (PAC)
  - Mix-Secure peripherals which support both secure and non-secure security attributions
- Three debug access levels allowing:
  - The highest debug level with no restrictions in term of memory and peripheral accesses
  - A restricted debug level with non-secure memory regions access only
  - The lowest debug level where no access is authorized except with a debugger using a Boot ROM-specifi mode
- Different chip erase support according to security settings
- Security configuration is fully stored in Flash and safely auto-loaded at startup during Boot ROM execution using CRC checks

What about **Privilege** and **Non-Privileged**??

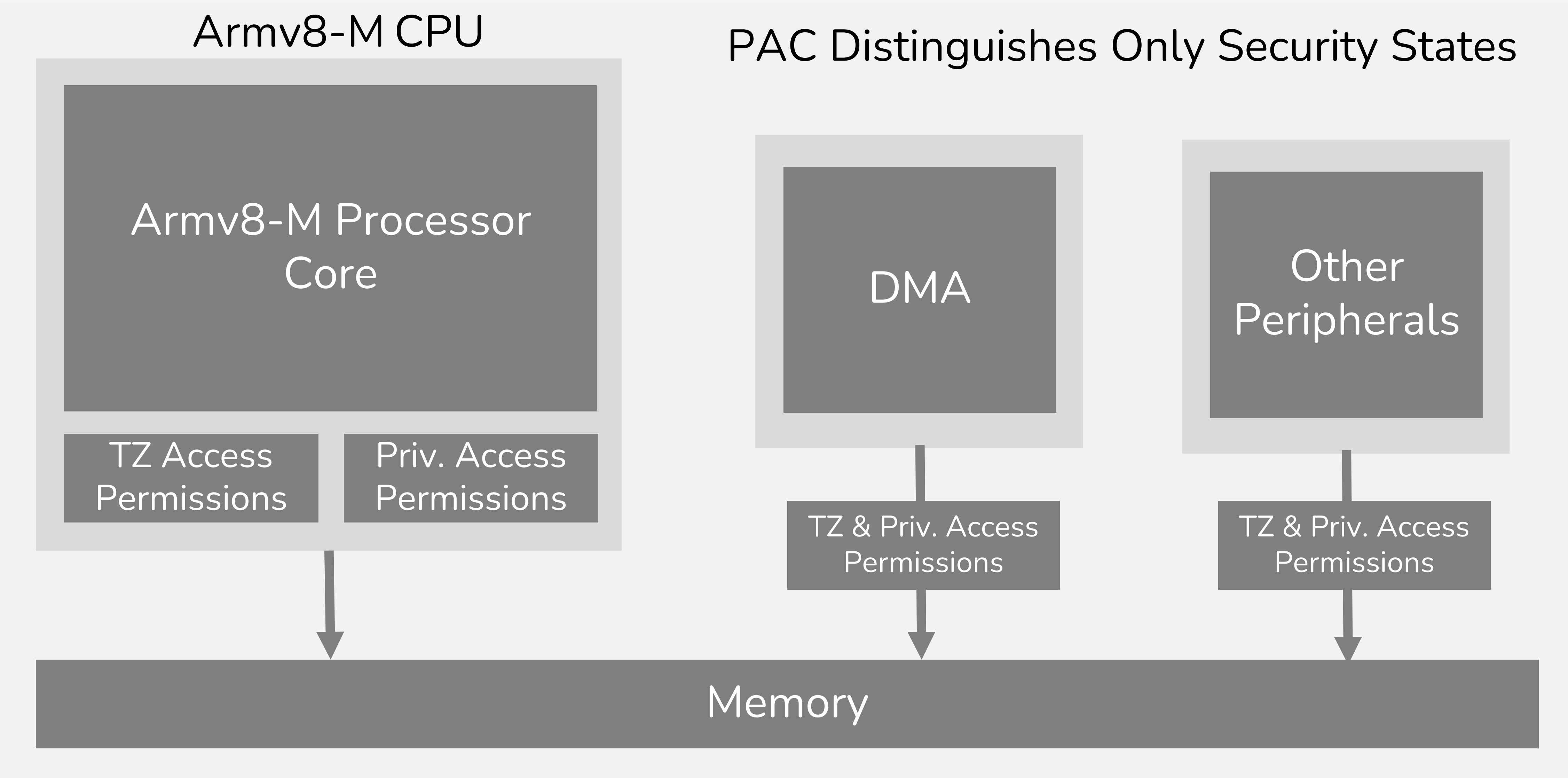
What about **Memory Protection** at the **System-Level**??

Pag. 53 - Microchip. SAM L10/L11 Family Data Sheet. Tech. rep. Microchip, June 2020.

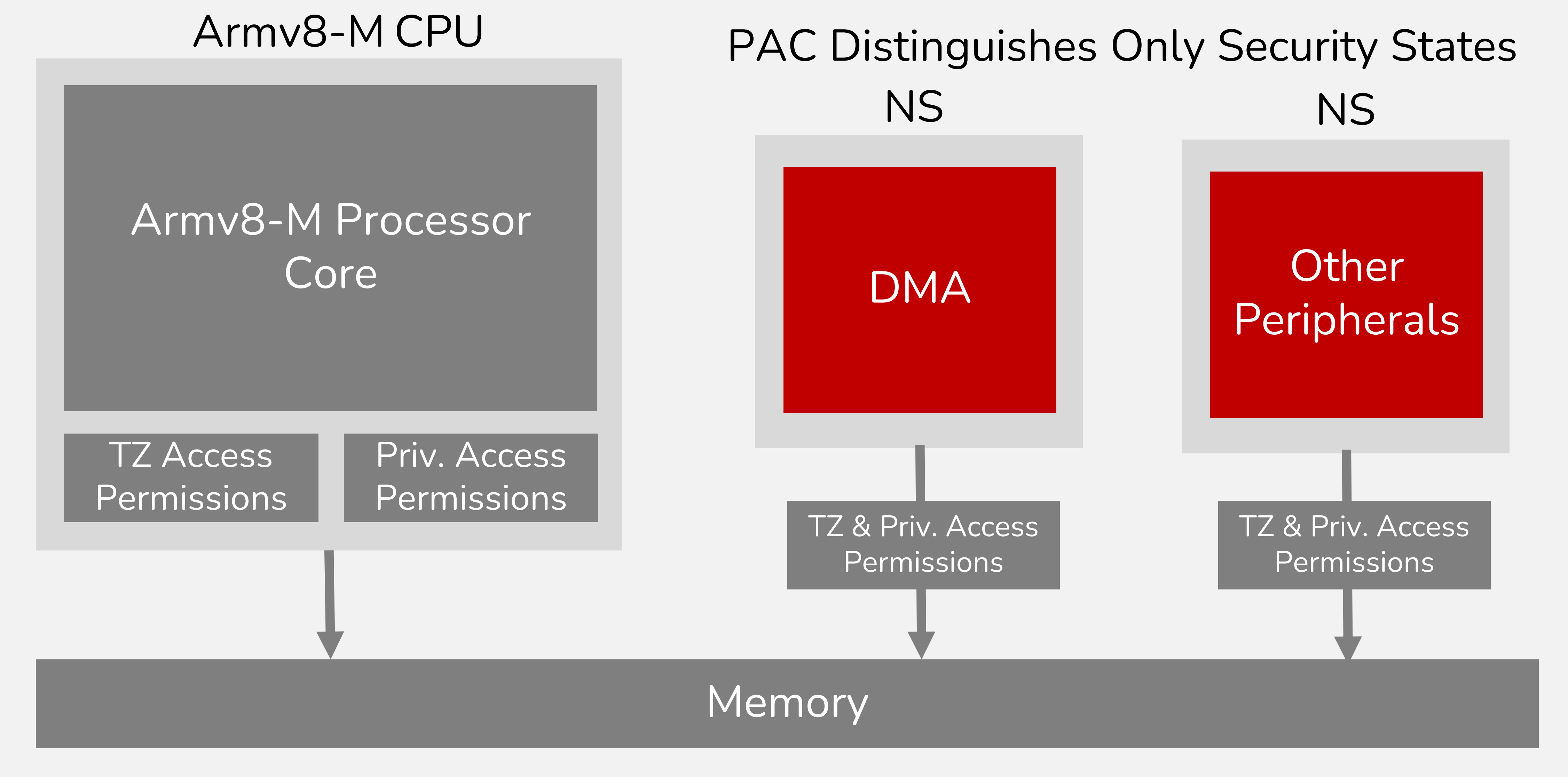
SAML11 WEAK PROTECTIONS



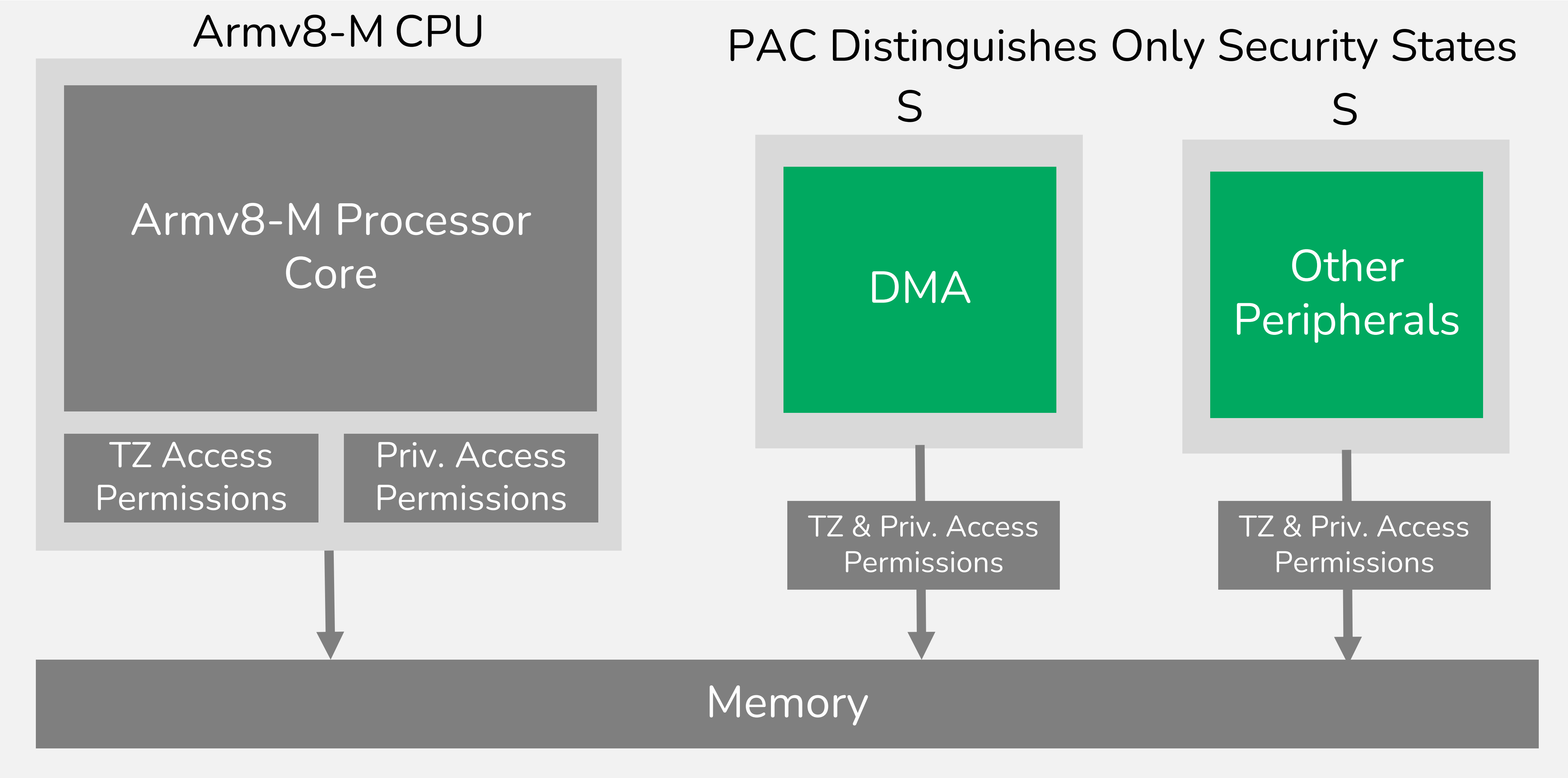
SAML11 WEAK PROTECTIONS



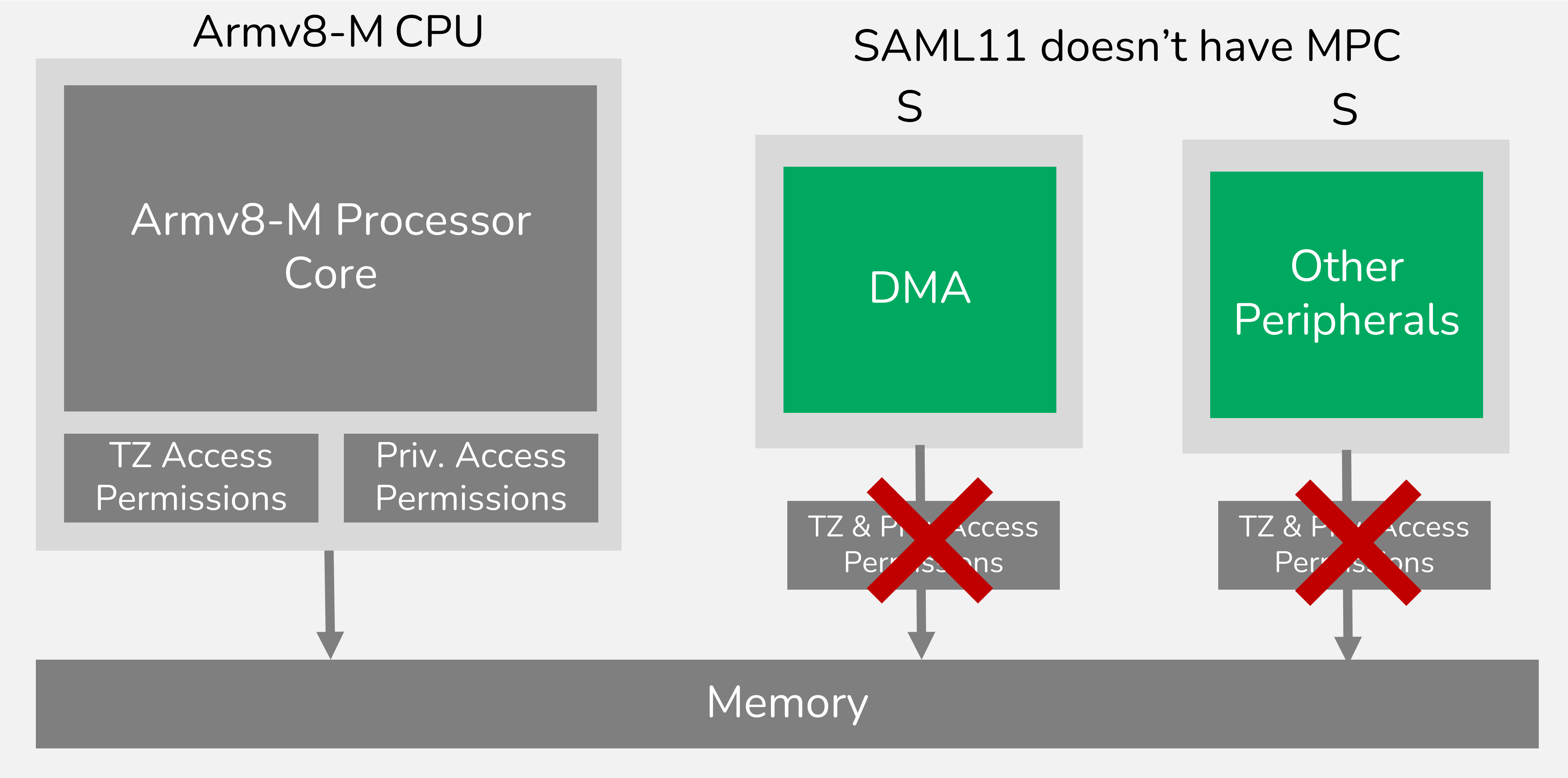
SAML11 WEAK PROTECTIONS



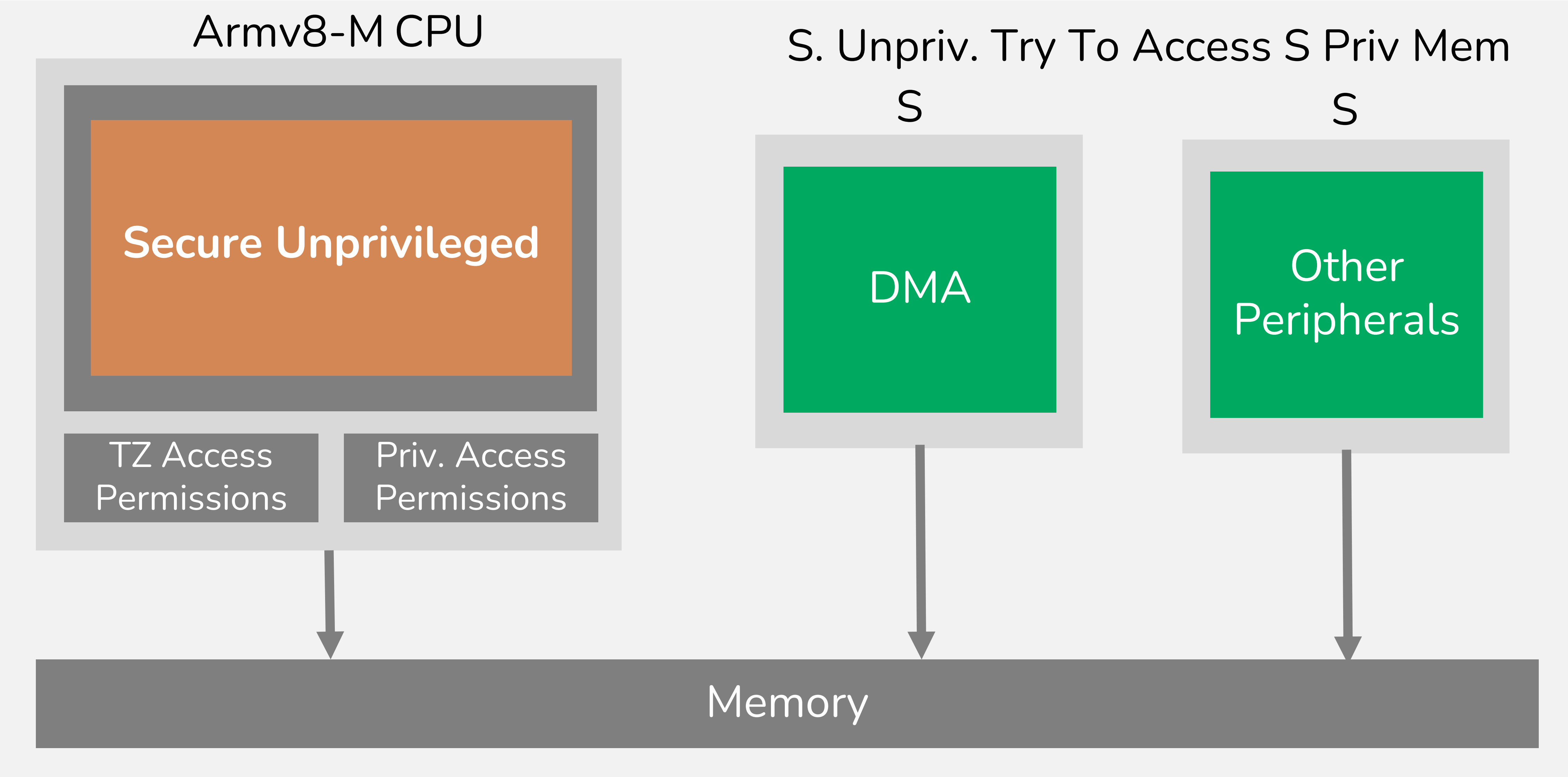
SAML11 WEAK PROTECTIONS



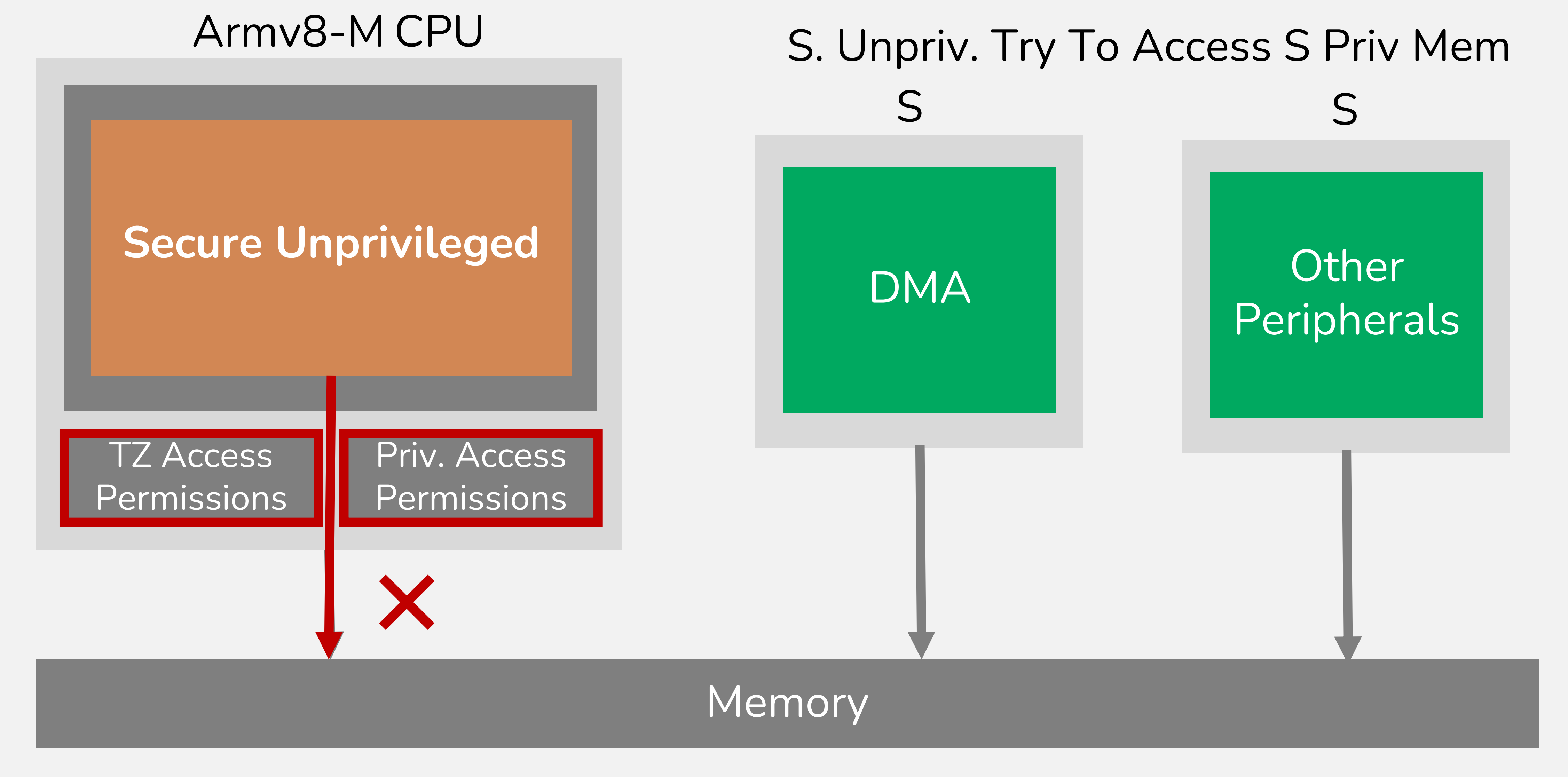
SAML11 WEAK PROTECTIONS



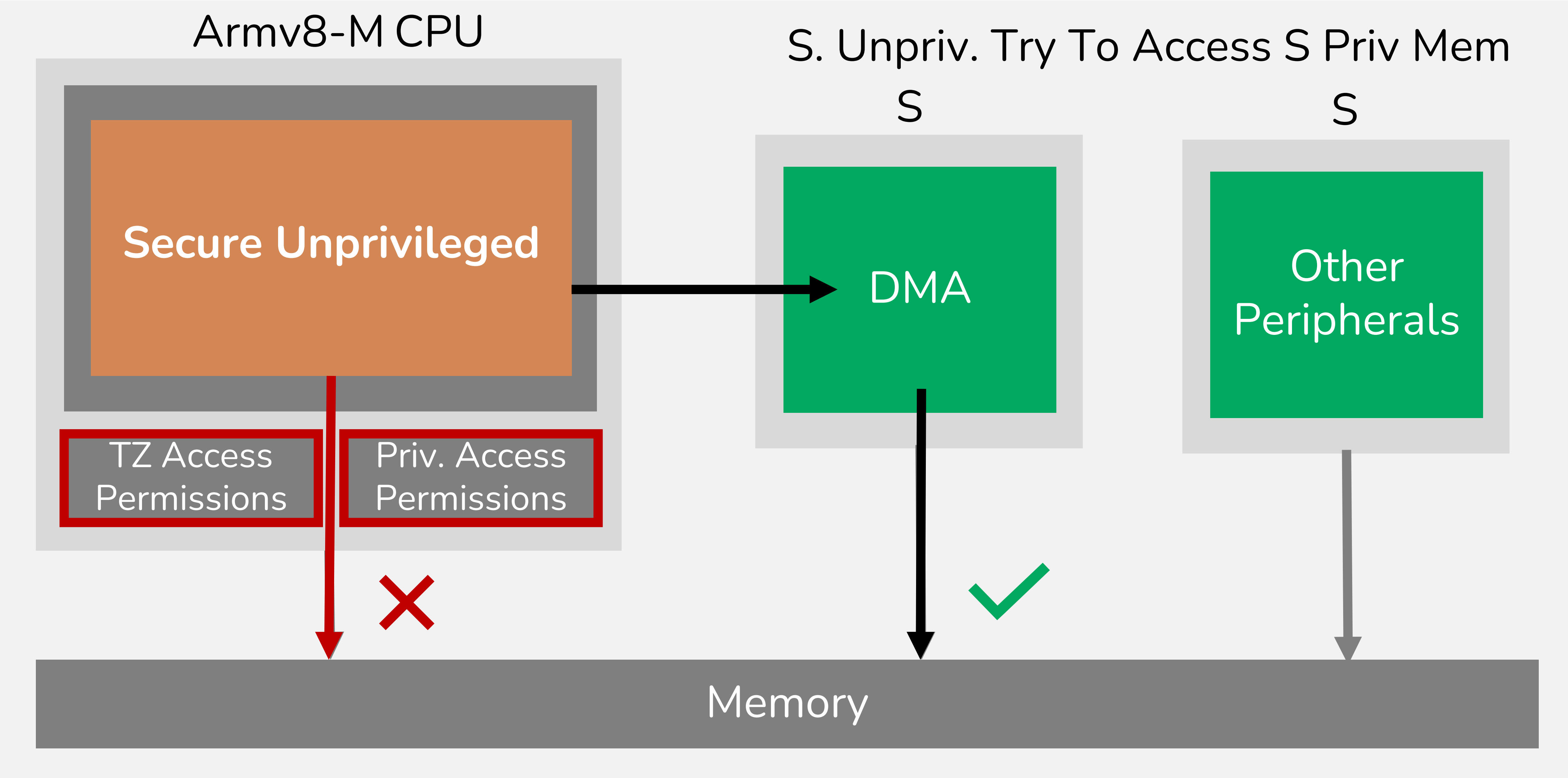
SAML11 WEAK PROTECTIONS



SAML11 WEAK PROTECTIONS



SAML11 WEAK PROTECTIONS



PSA Certification

SAML11

PSA Certification

SAML11

SILICON

SAML11-KPH

MICROCHIP

psacertified™
level one

PSA Certification

SAML11

SILICON

SAML11-KPH

MICROCHIP

psacertified™
level one

PSA Certification

SAML11

SILICON

SAML11-KPH

MICROCHIP

psacertified™
level one

PSA Certification

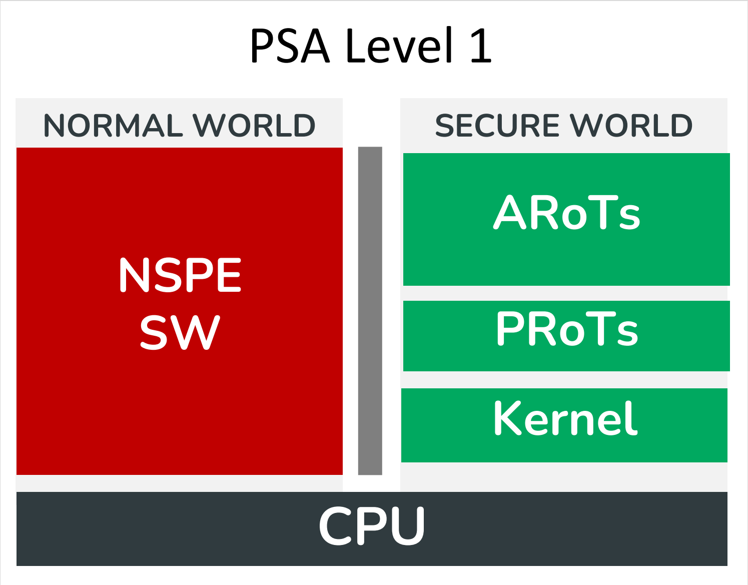
SAML11

SILICON

SAML11-KPH

MICROCHIP

psacertified™
level one



PSA Level 1

NORMAL WORLD

NSPE SW

SECURE WORLD

ARoTs

PRoTs

Kernel

CPU

PSA Certification

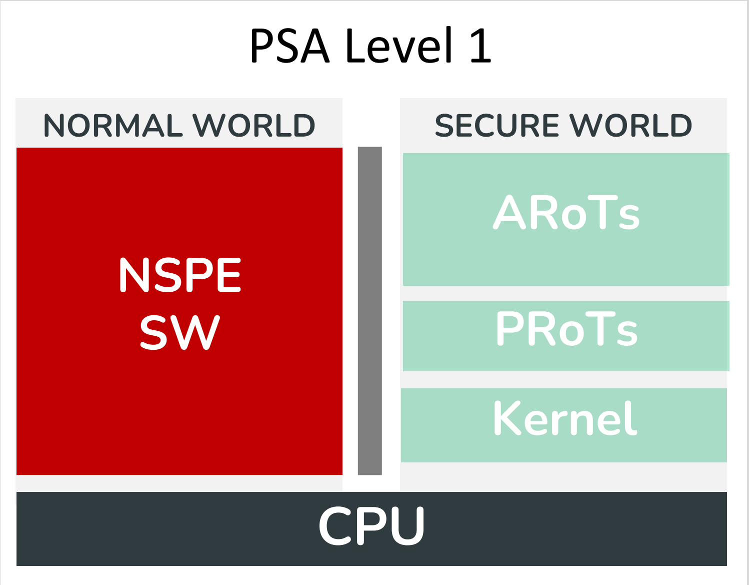
SAML11

SILICON

SAML11-KPH

MICROCHIP

psacertified™
level one



PSA Certification

SAML11

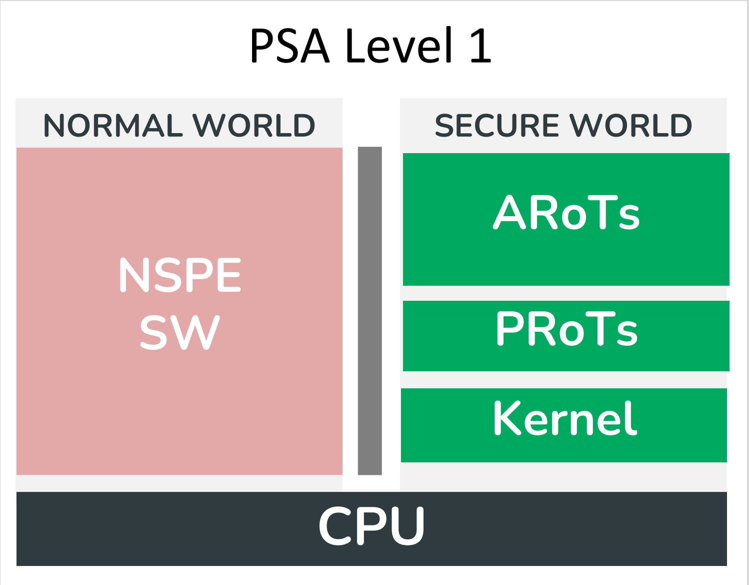
SILICON

SAML11-KPH

MICROCHIP

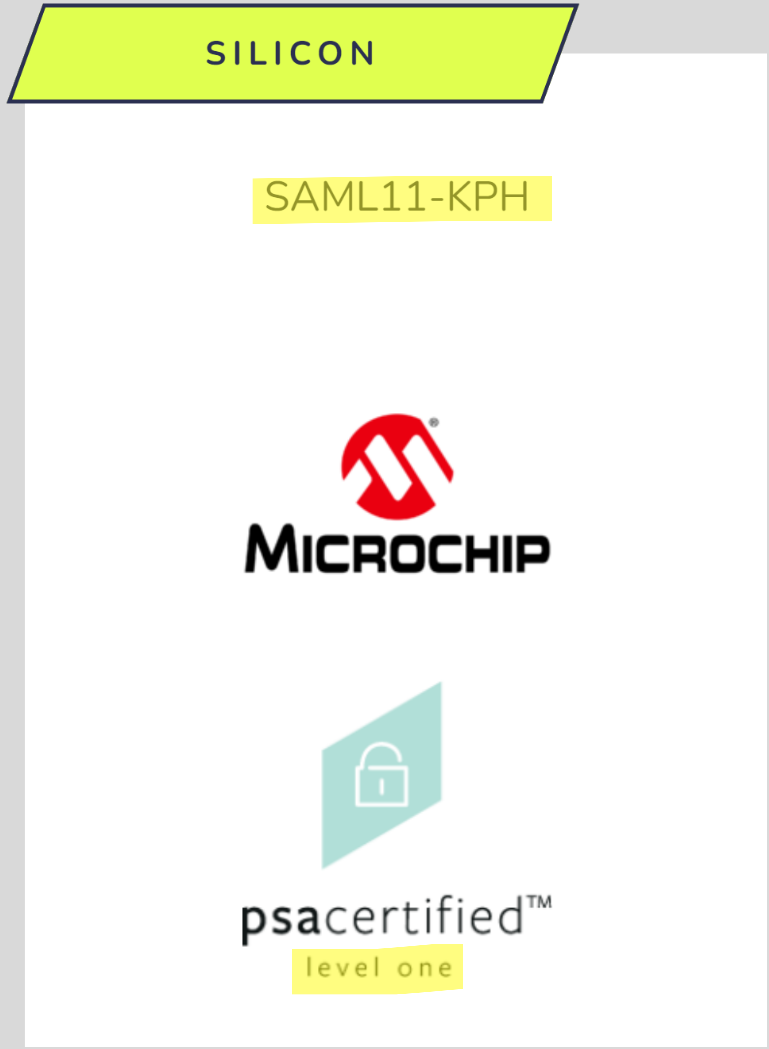
psacertified™

level one



PSA Certification

SAML11 + Kinibi-M

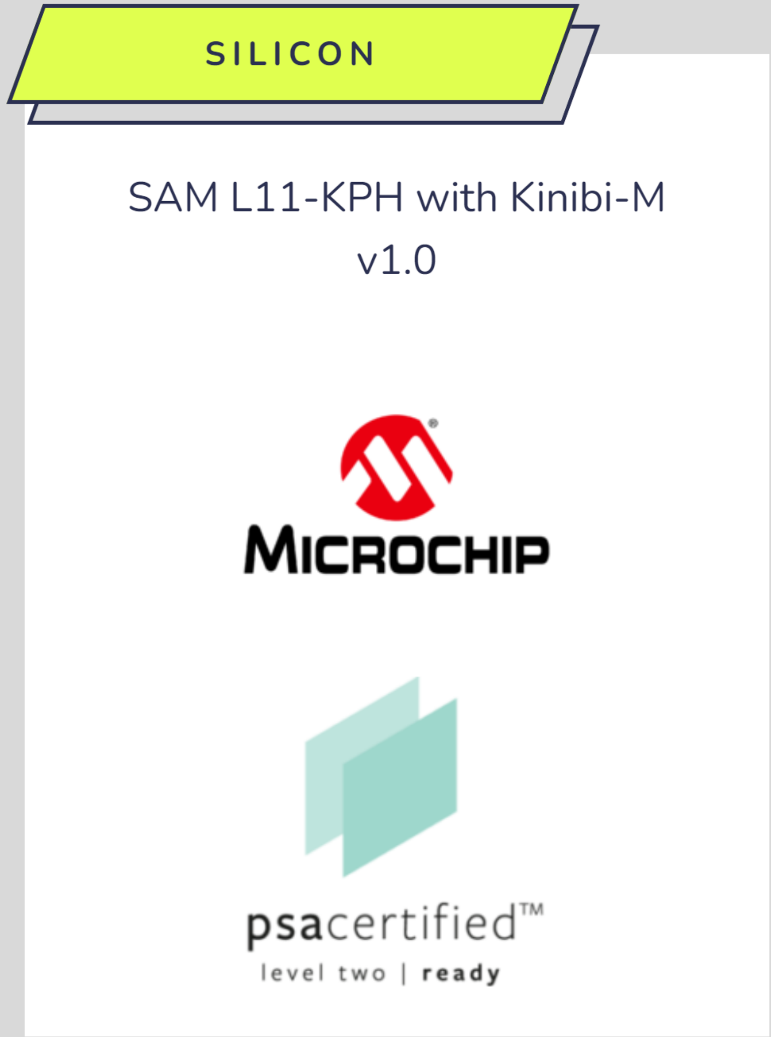


SILICON

SAML11-KPH

psacertified™

level one

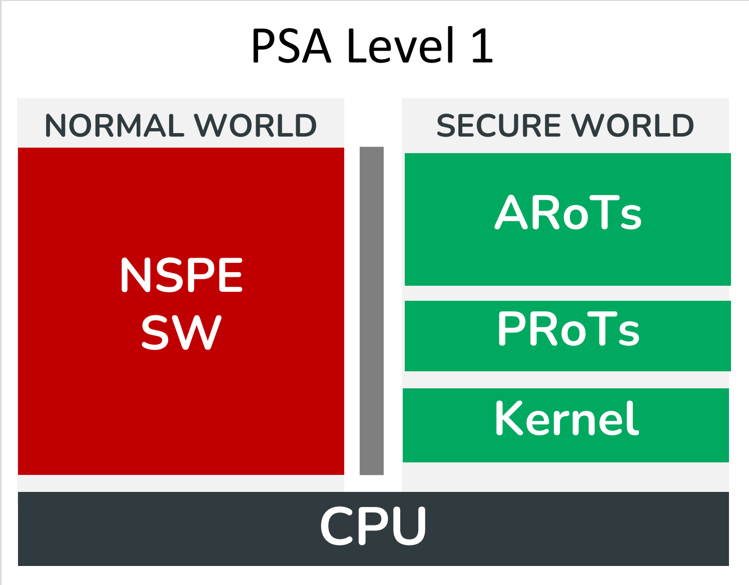


SILICON

SAM L11-KPH with Kinibi-M v1.0

psacertified™

level two | ready



PSA Level 1

NORMAL WORLD

NSPE SW

SECURE WORLD

ARoTs

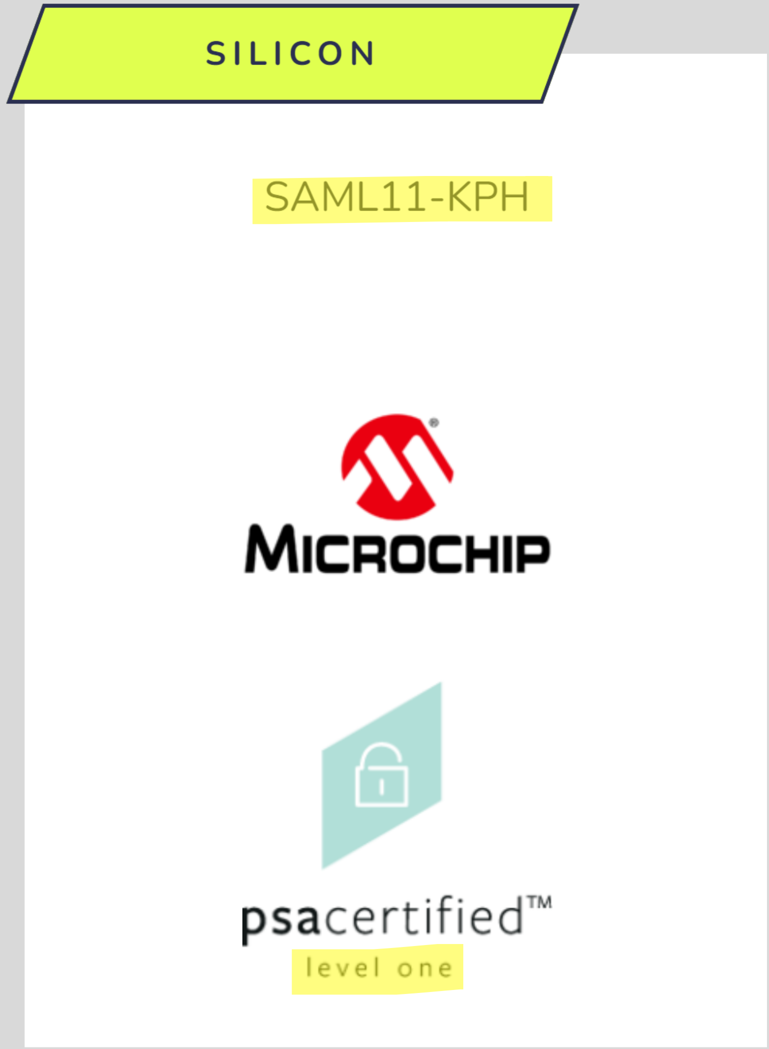
PRoTs

Kernel

CPU

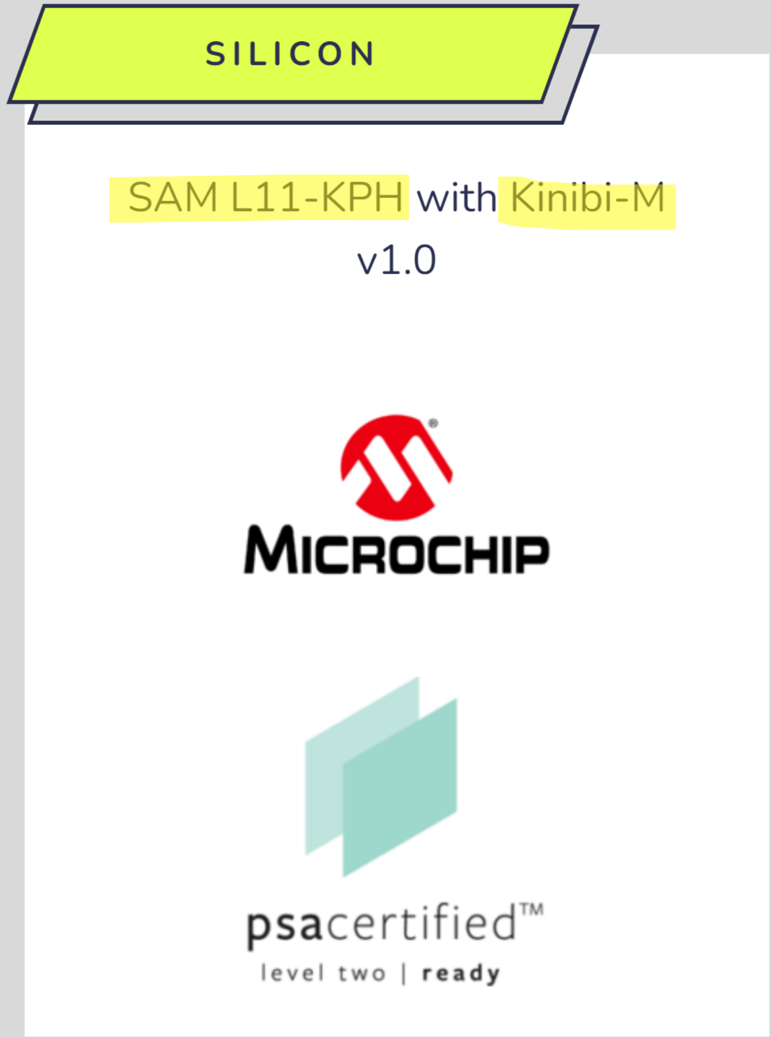
PSA Certification

SAML11 + Kinibi-M



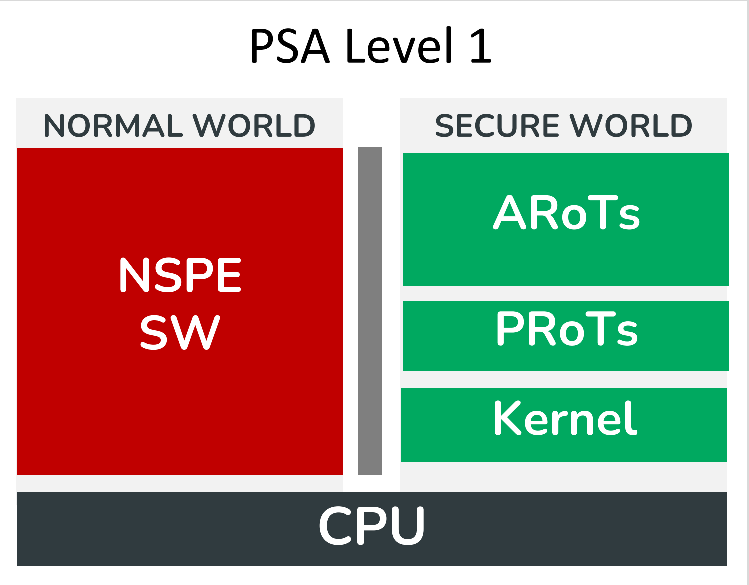
SILICON

SAML11-KPH



SILICON

SAM L11-KPH with Kinibi-M v1.0



PSA Level 1

NORMAL WORLD

SECURE WORLD

NSPE SW

ARoTs

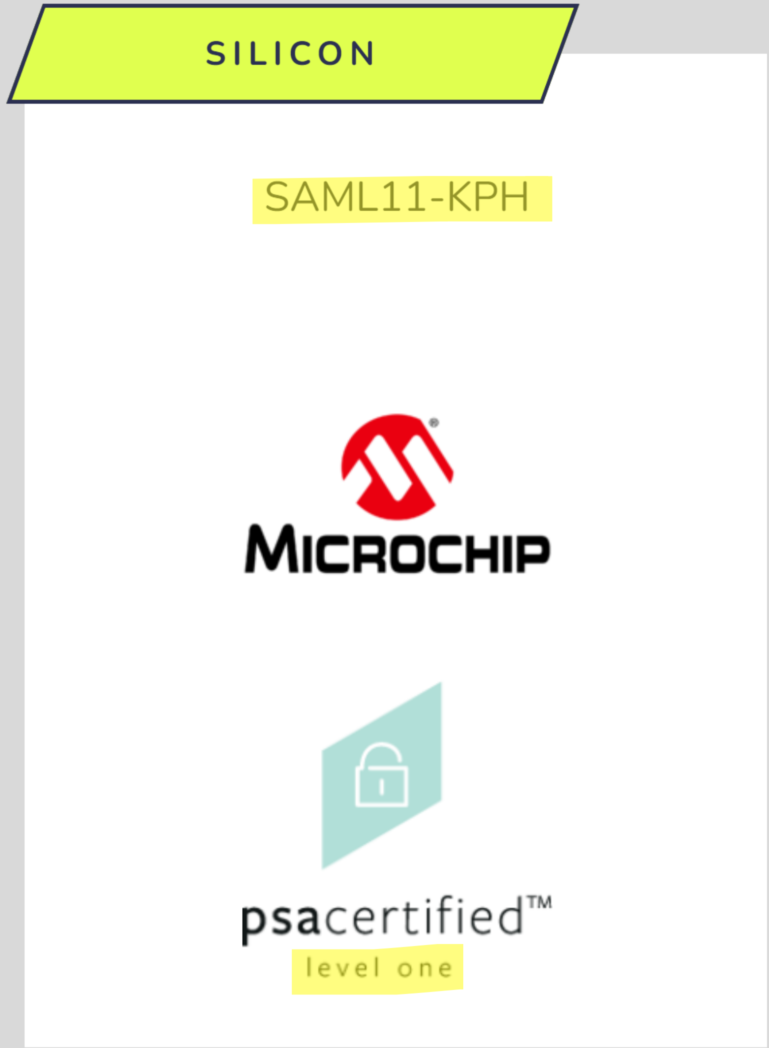
PRoTs

Kernel

CPU

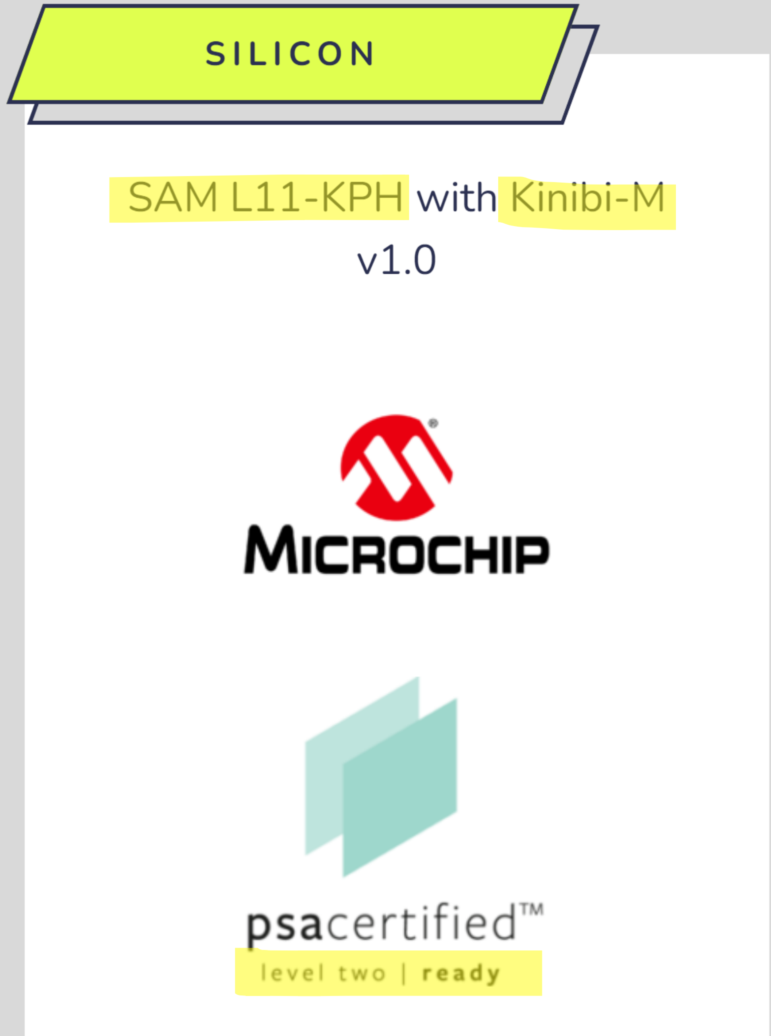
PSA Certification

SAML11 + Kinibi-M



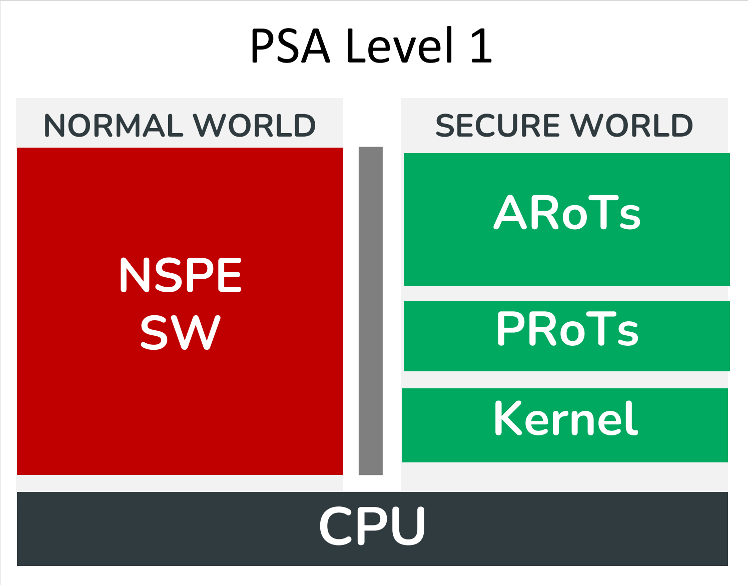
SILICON

SAML11-KPH



SILICON

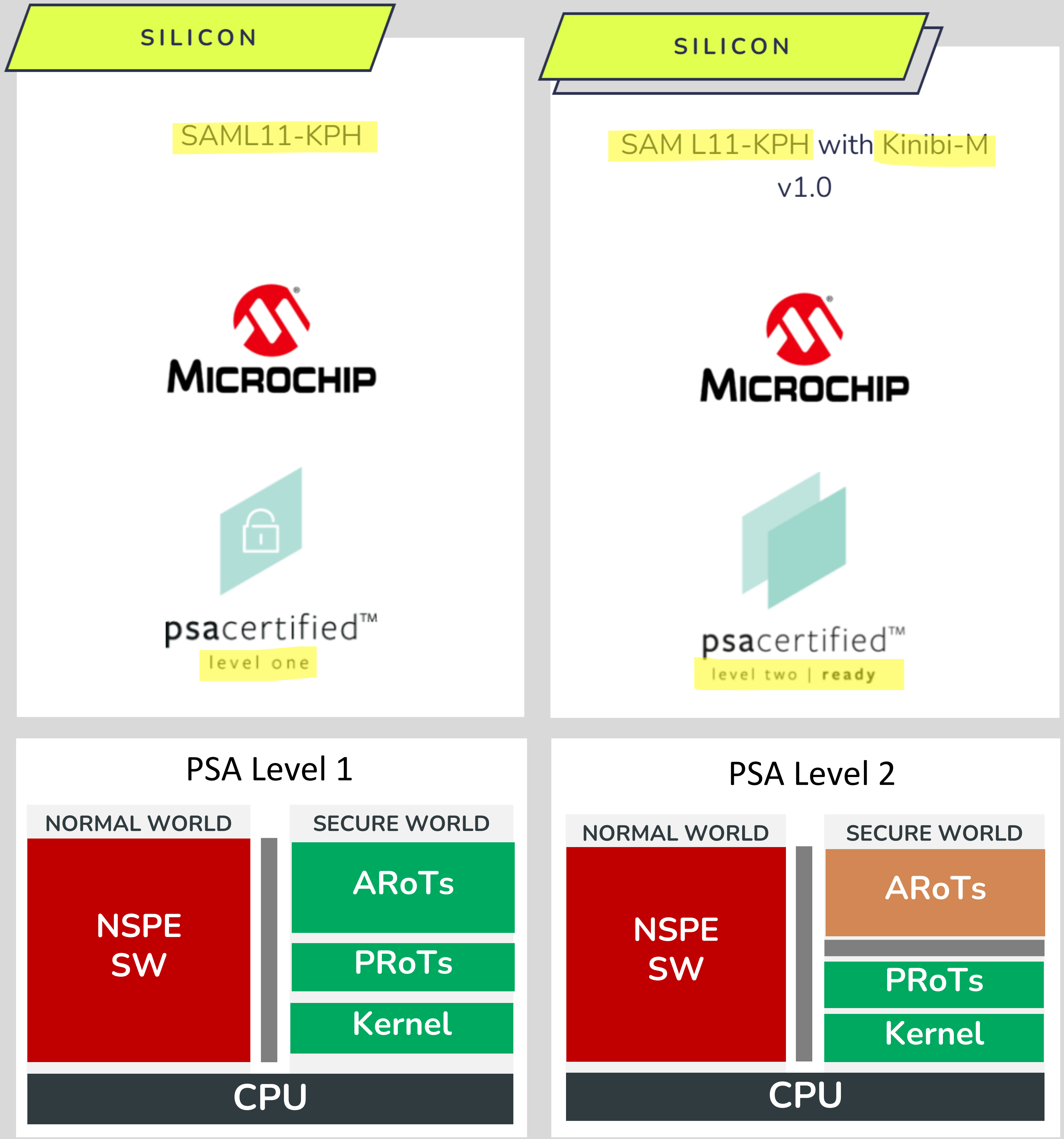
SAM L11-KPH with Kinibi-M v1.0



PSA Level 1

PSA Certification

SAML11 + Kinibi-M



SILICON

SAML11-KPH

level one

SILICON

SAM L11-KPH with Kinibi-M v1.0

level two | ready

PSA Level 1

NORMAL WORLD

NSPE SW

SECURE WORLD

ARoTs

PRoTs

Kernel

CPU

PSA Level 2

NORMAL WORLD

NSPE SW

SECURE WORLD

ARoTs

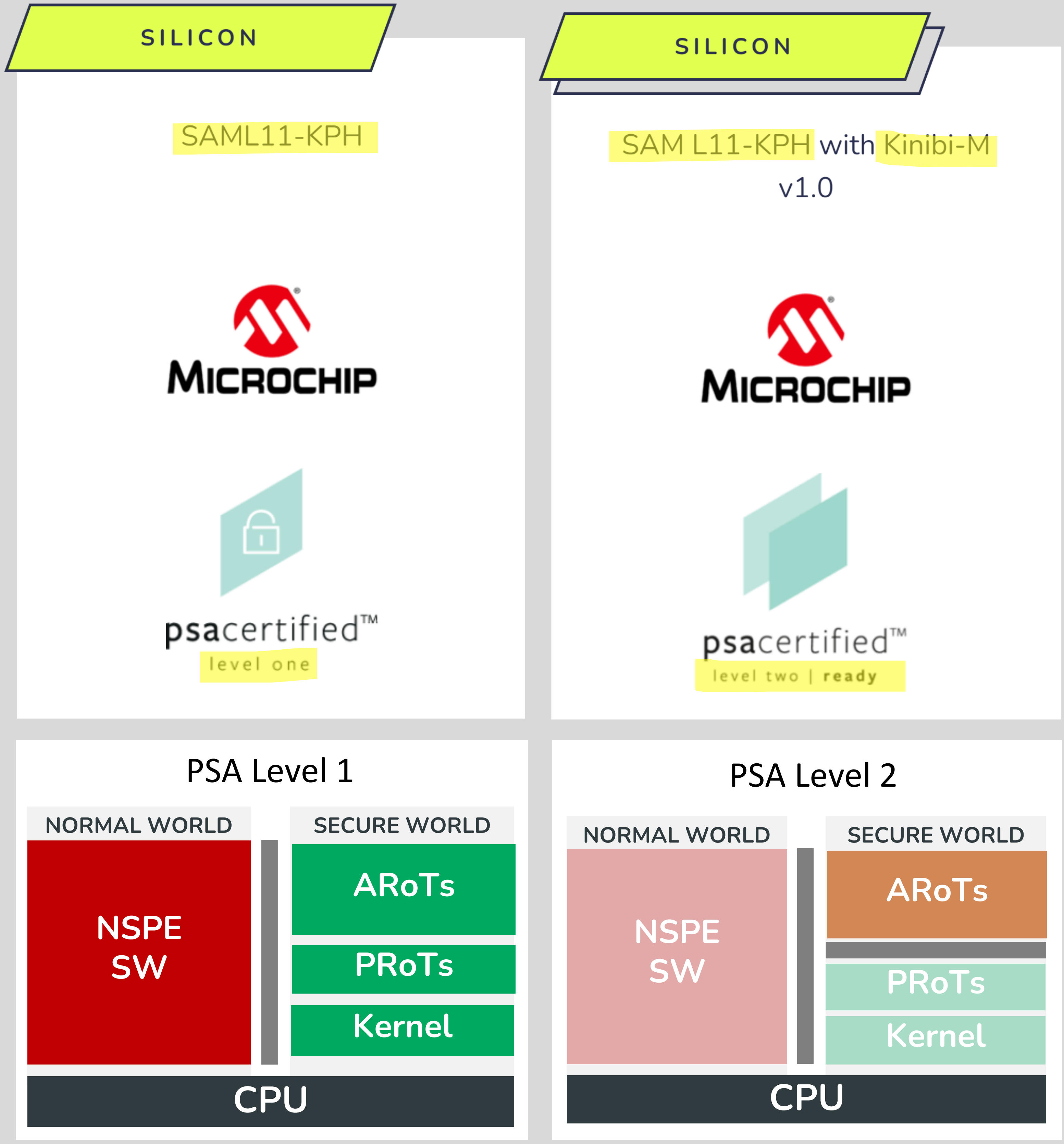
PRoTs

Kernel

CPU

PSA Certification

SAML11 + Kinibi-M



SILICON

SAML11-KPH

psacertified™ level one

SILICON

SAM L11-KPH with Kinibi-M v1.0

psacertified™ level two | ready

PSA Level 1

NORMAL WORLD

NSPE SW

SECURE WORLD

ARoTs

PRoTs

Kernel

CPU

PSA Level 2

NORMAL WORLD

NSPE SW

SECURE WORLD

ARoTs

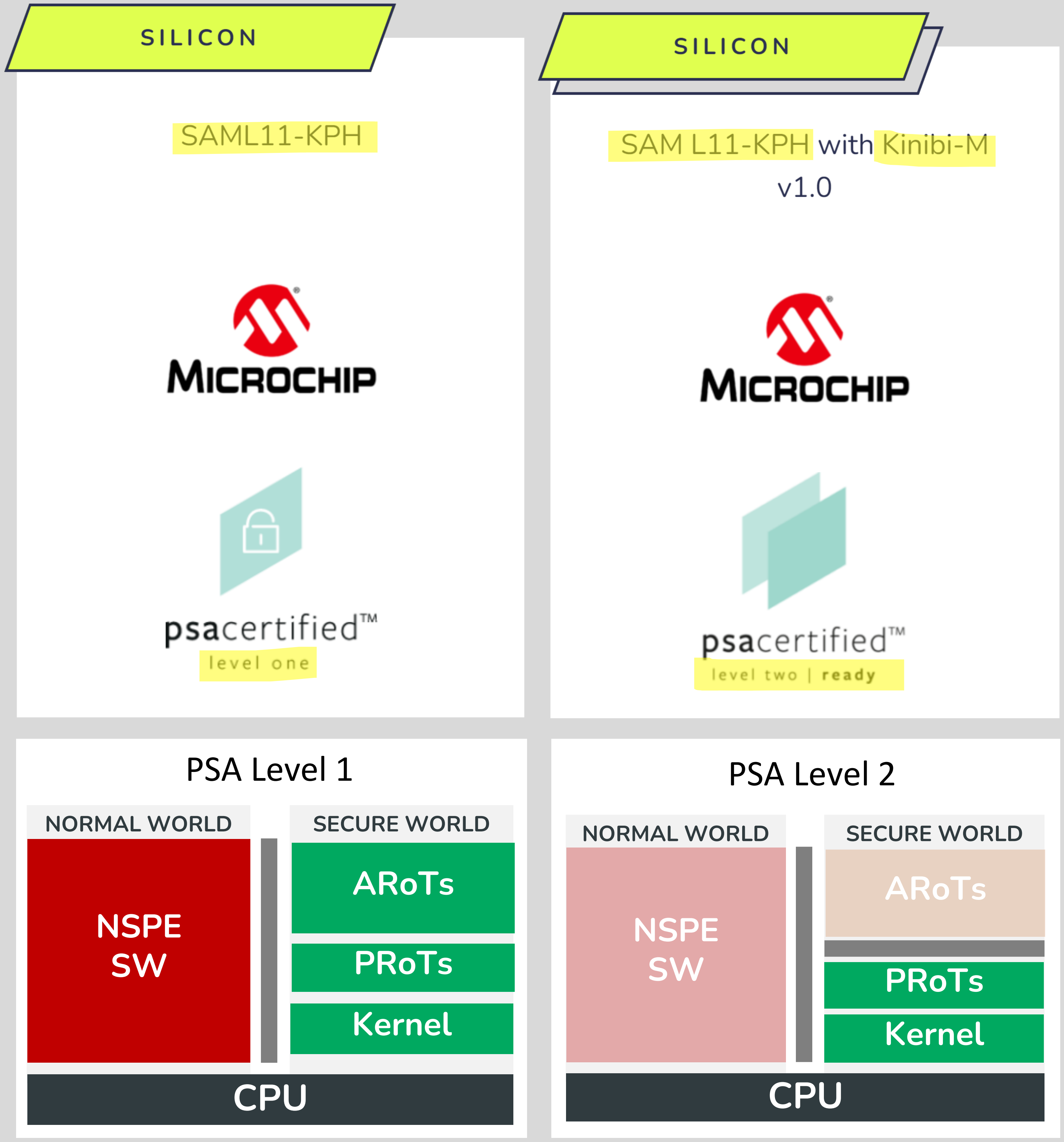
PRoTs

Kernel

CPU

PSA Certification

SAML11 + Kinibi-M



SILICON

SAML11-KPH

psacertified™

level one

SILICON

SAM L11-KPH with Kinibi-M v1.0

psacertified™

level two | ready

PSA Level 1

NORMAL WORLD

NSPE SW

SECURE WORLD

ARoTs

PRoTs

Kernel

CPU

PSA Level 2

NORMAL WORLD

NSPE SW

SECURE WORLD

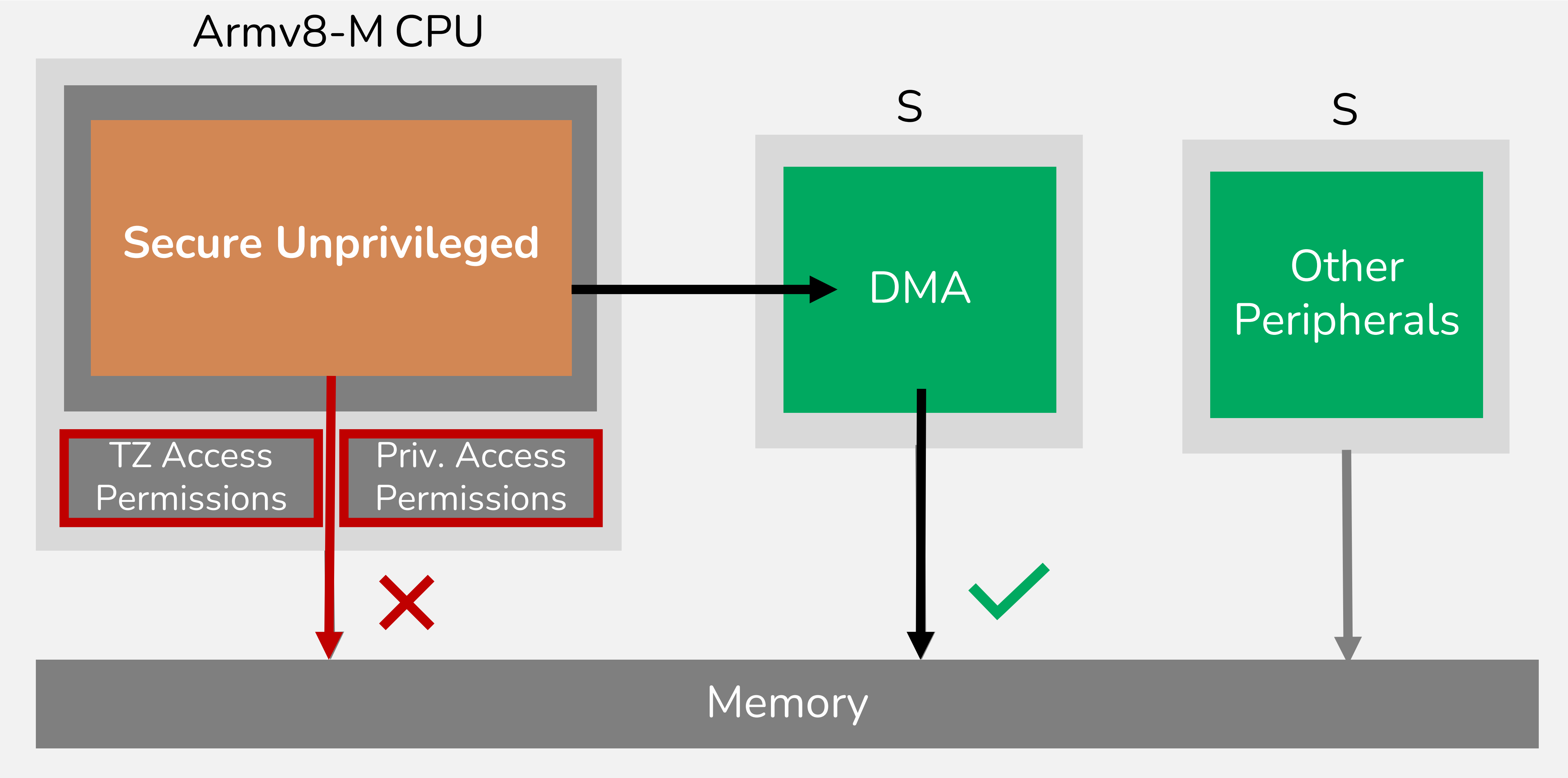
ARoTs

PRoTs

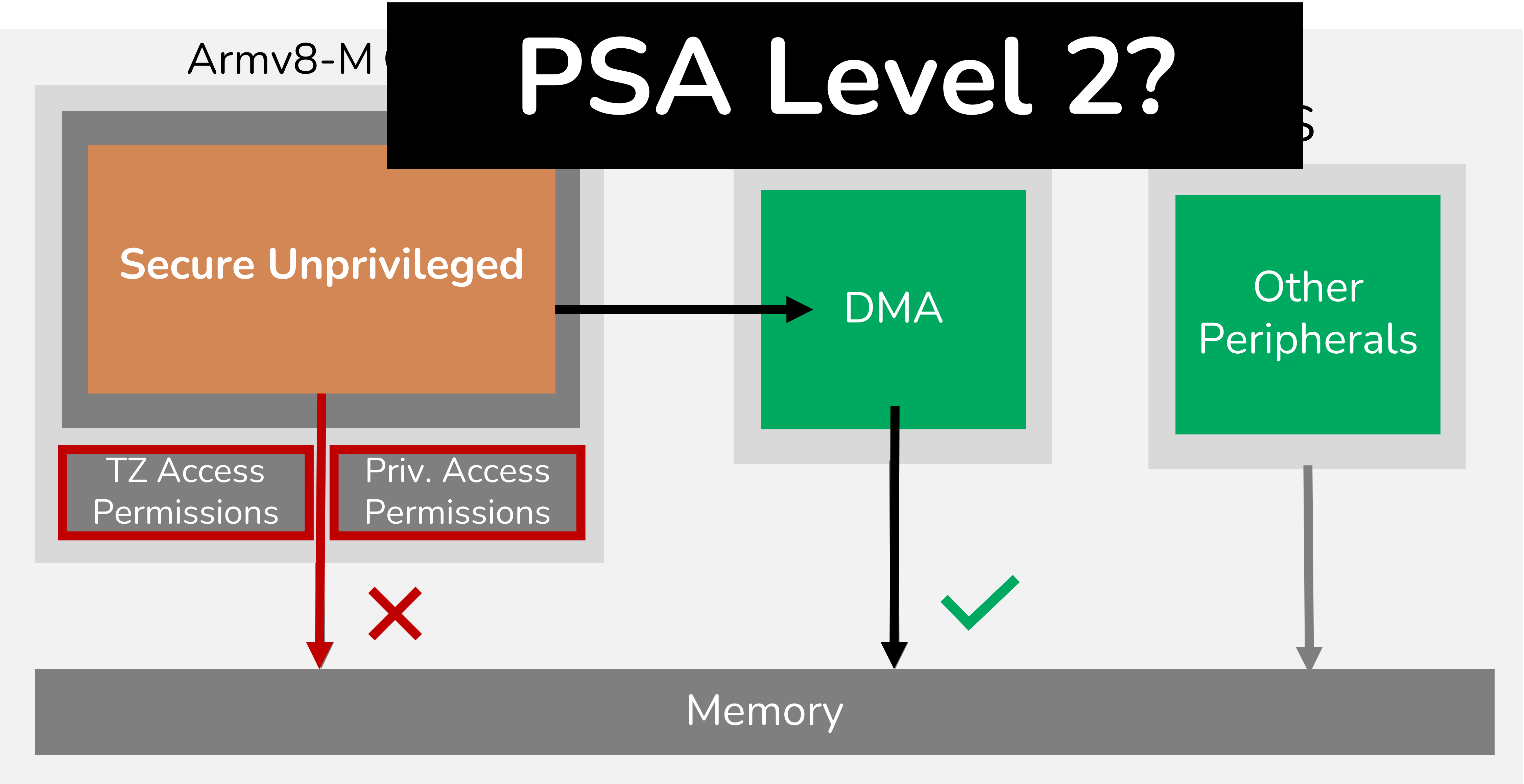
Kernel

CPU

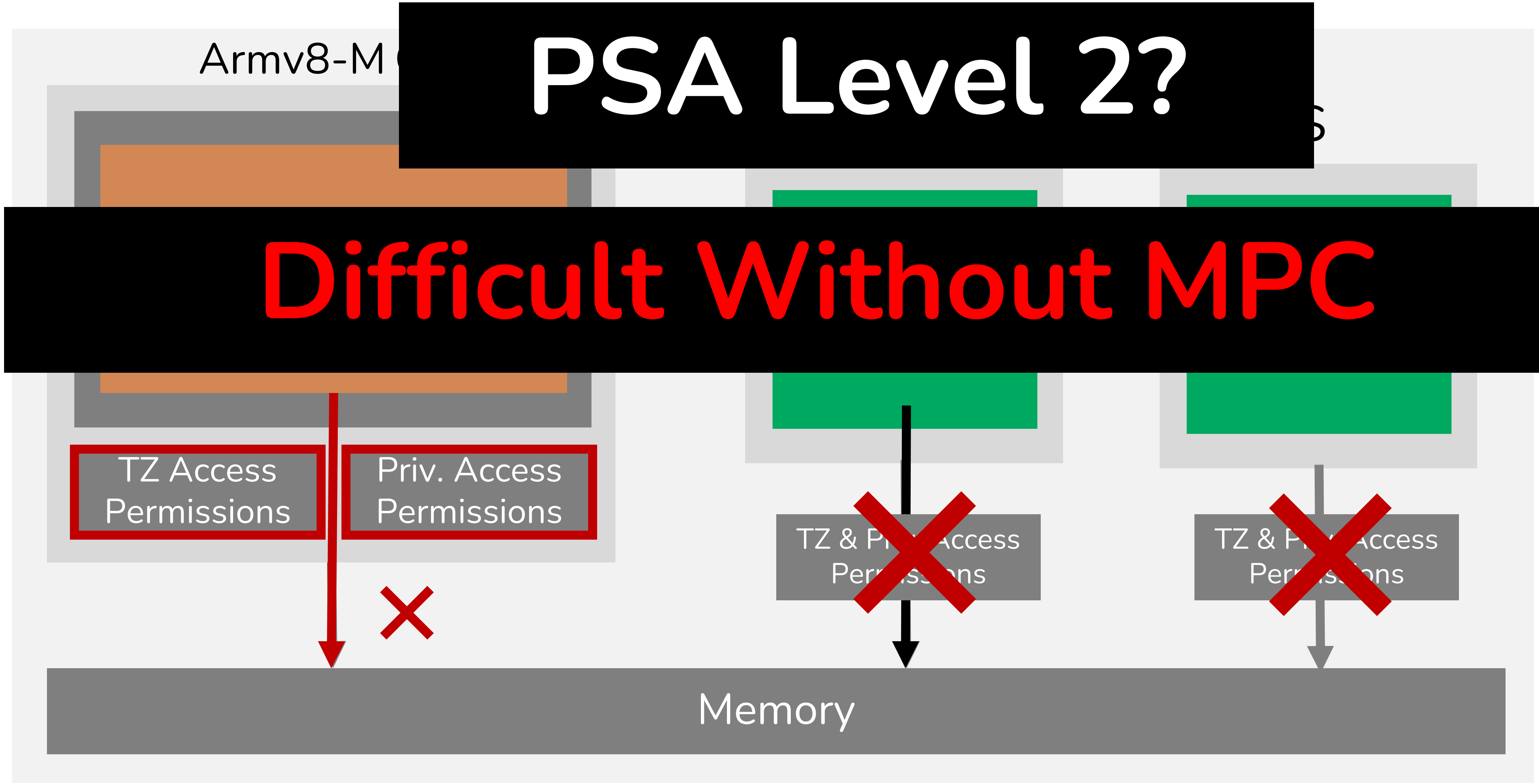
SAML11 WEAK PROTECTIONS



SAML11 WEAK PROTECTIONS

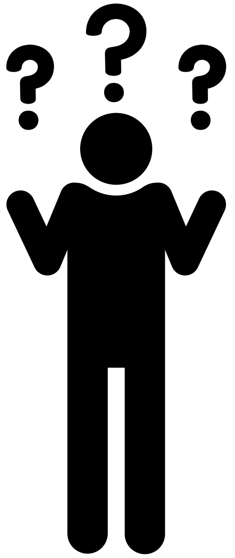


SAML11 WEAK PROTECTIONS



PSA Level 2?

Difficult Without MPC



We report to Microchip that the lack of a MPC may create security issues, special in PSA level 2/3, Microchip didn’t take any actions!

Responsible Disclosure: Microchip

Trustonic

Kinibi-M

TRUSTONIC KINIBI-M

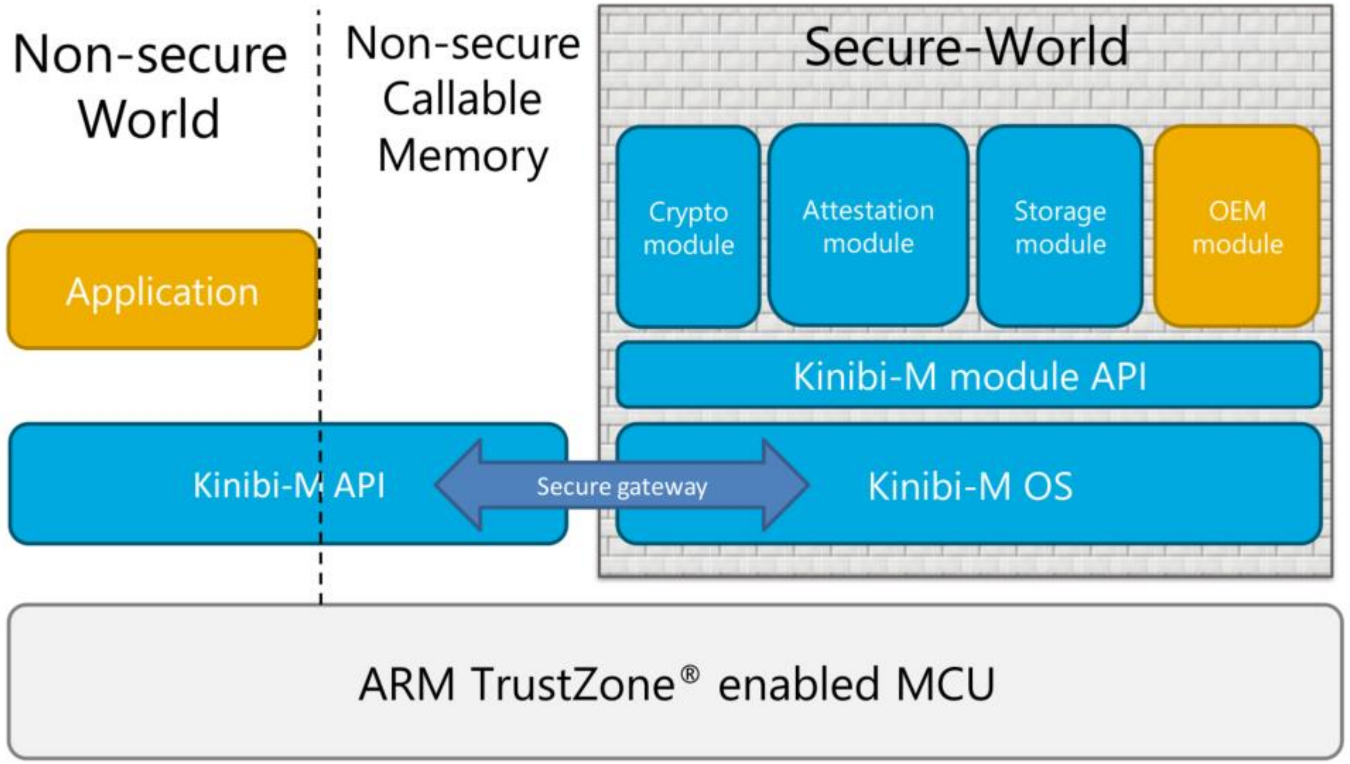


Figure 1: Kinibi-M Architecture Overview.

Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M

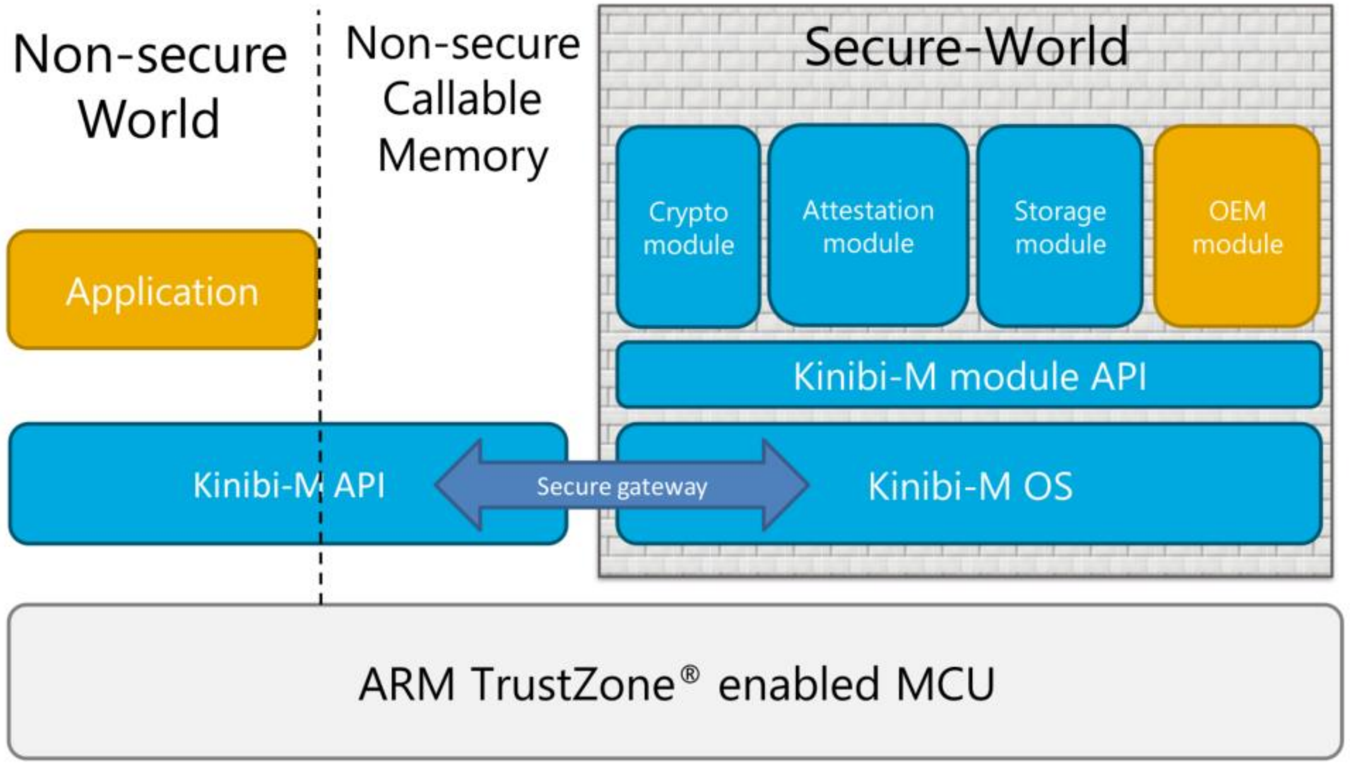


Figure 1: Kinibi-M Architecture Overview.

PSA Level 2

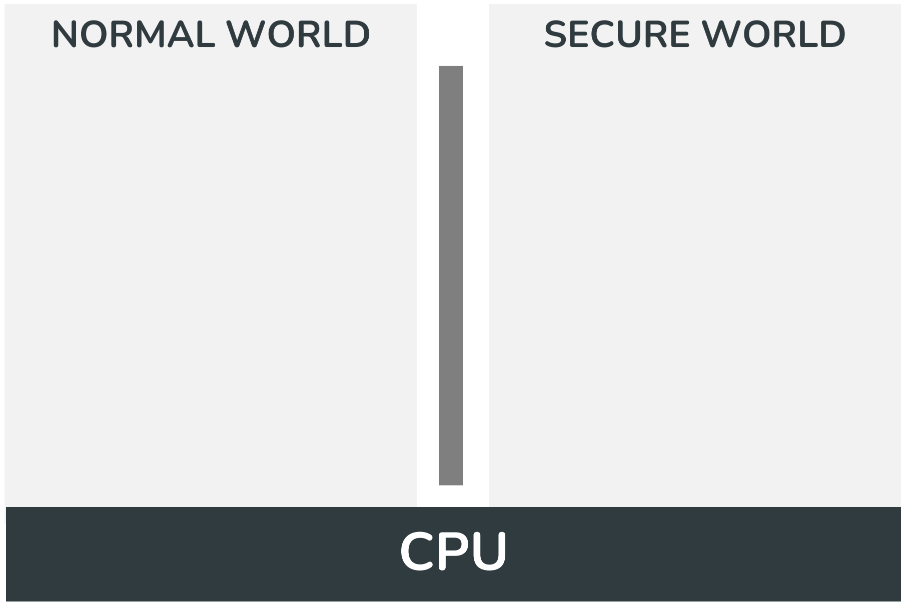


Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M

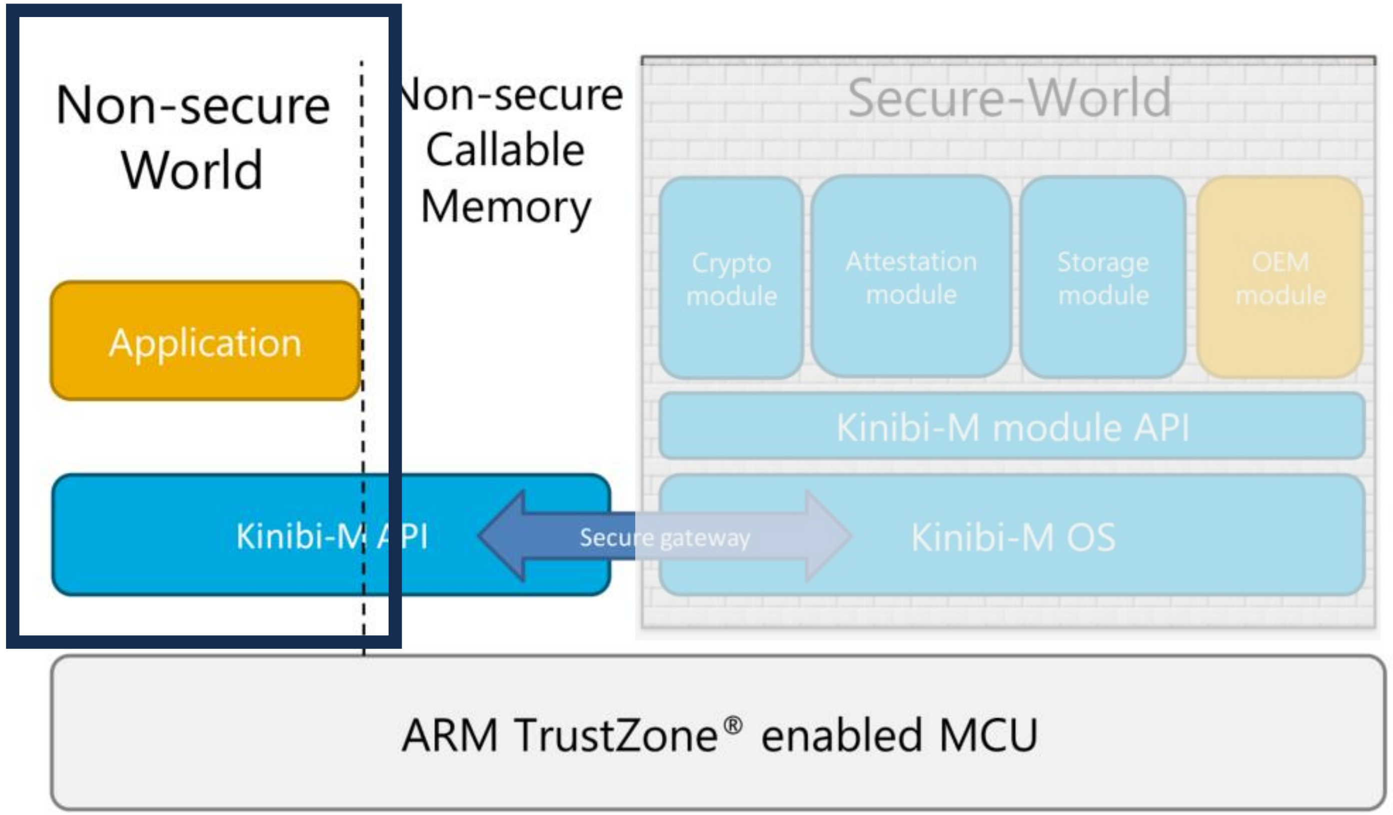


Figure 1: Kinibi-M Architecture Overview.

PSA Level 2

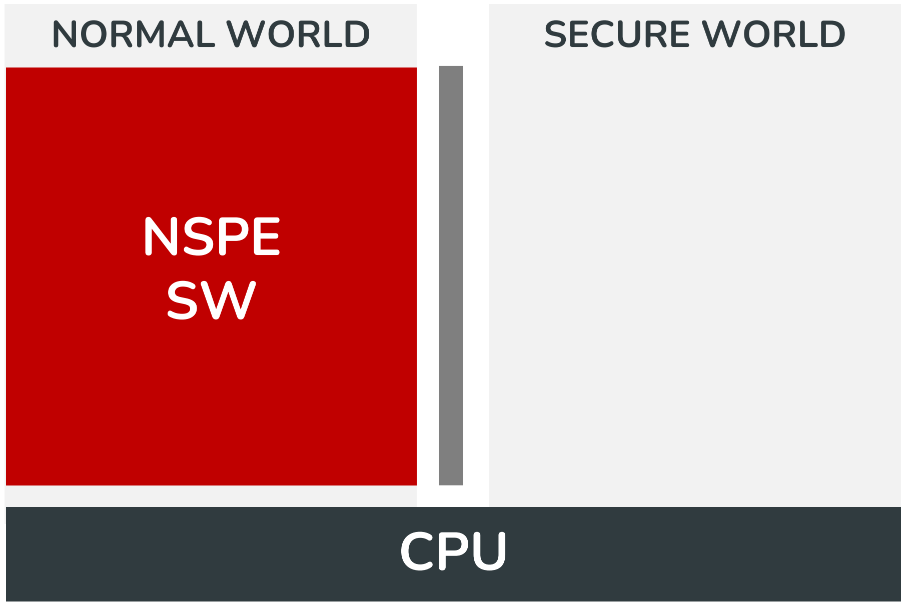


Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M

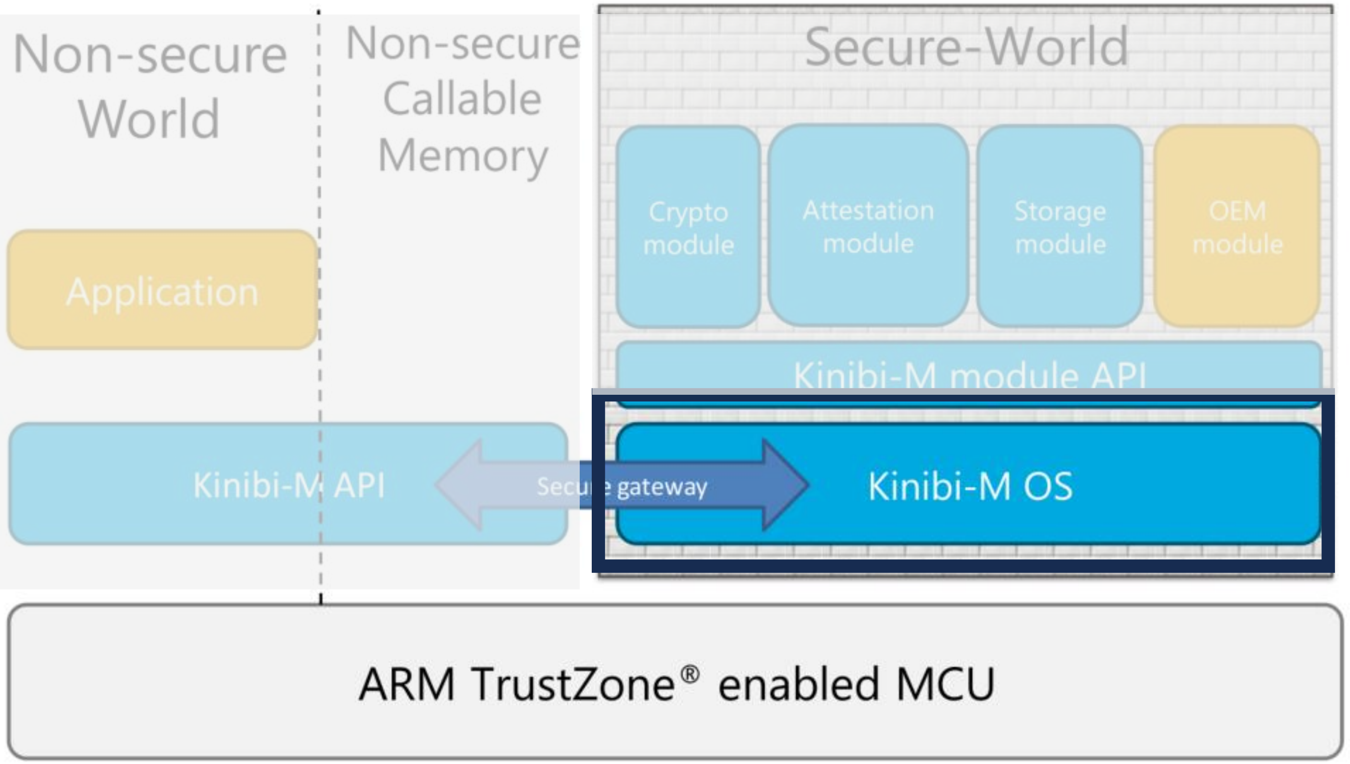


Figure 1: Kinibi-M Architecture Overview.

PSA Level 2

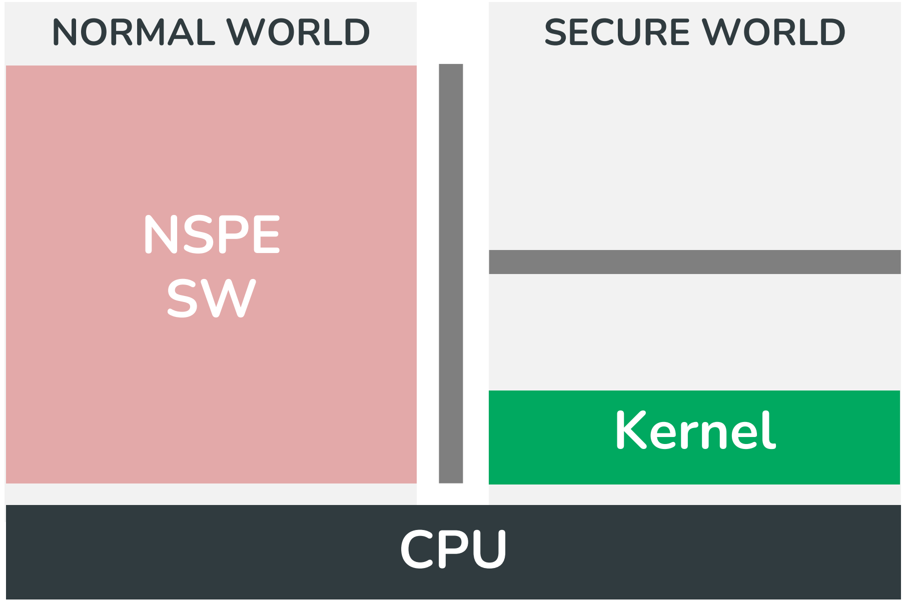


Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M

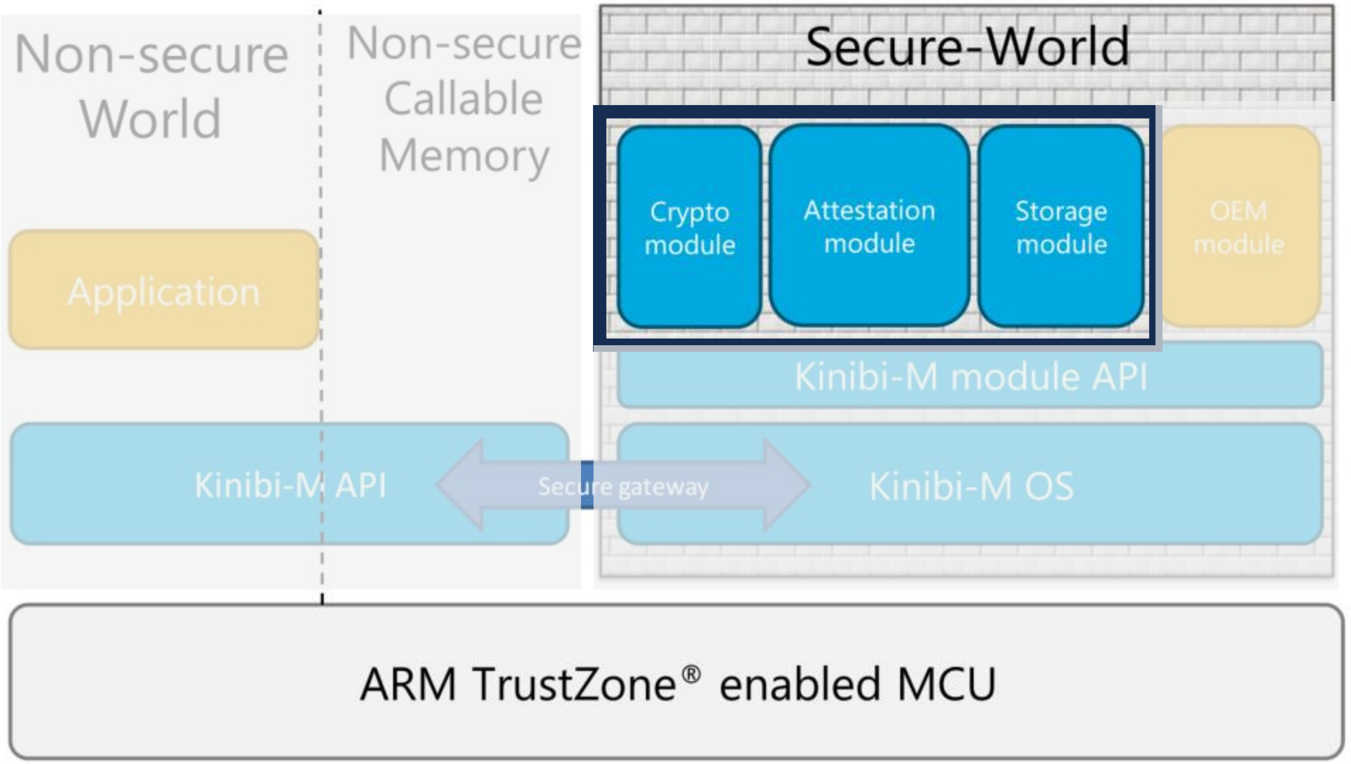


Figure 1: Kinibi-M Architecture Overview.

PSA Level 2

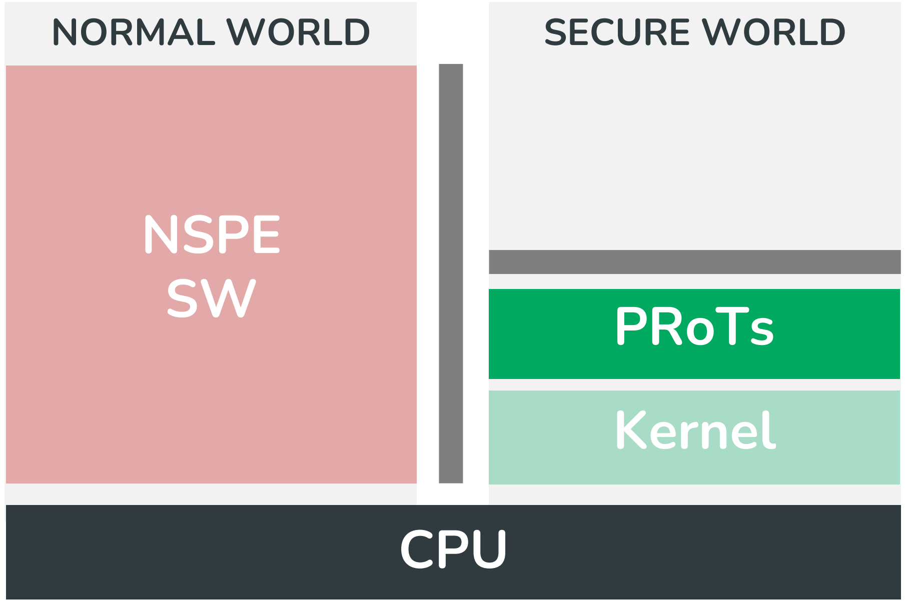


Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M

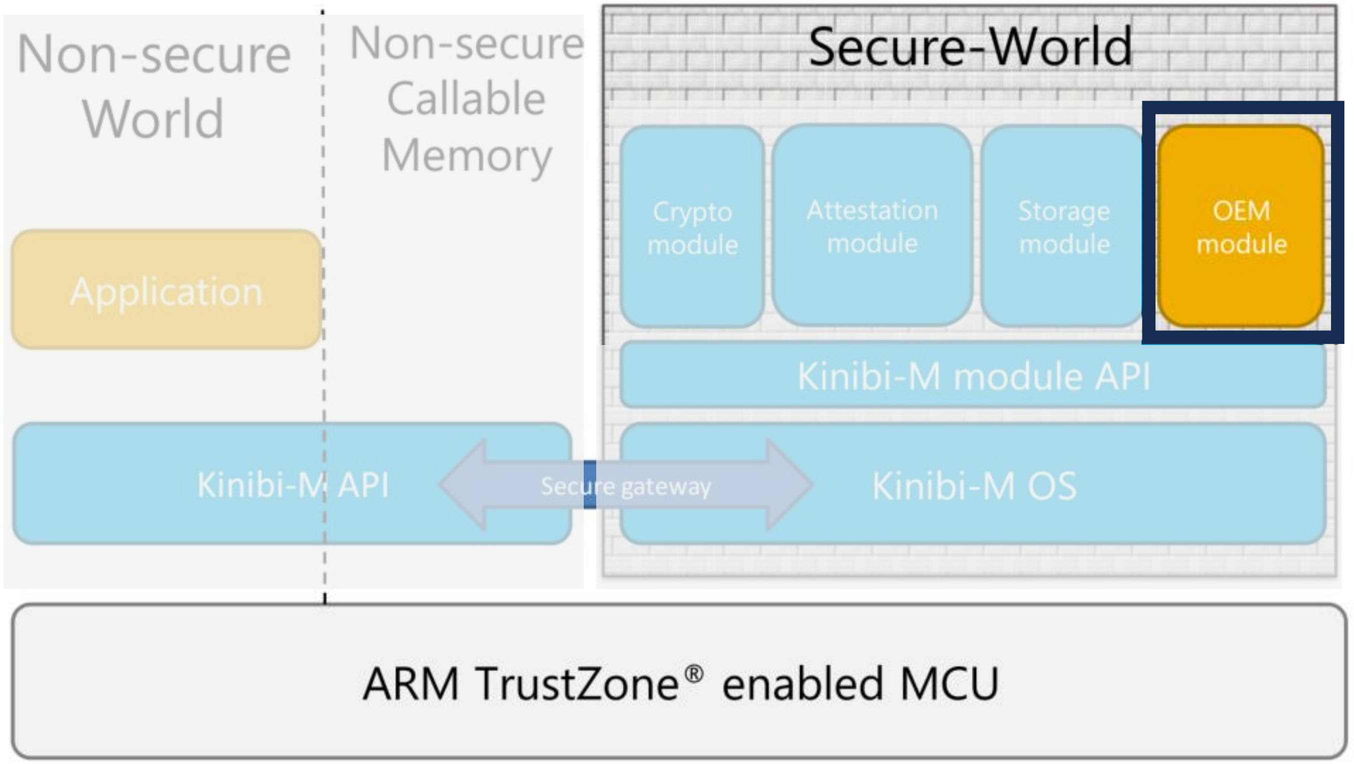


Figure 1: Kinibi-M Architecture Overview.

PSA Level 2

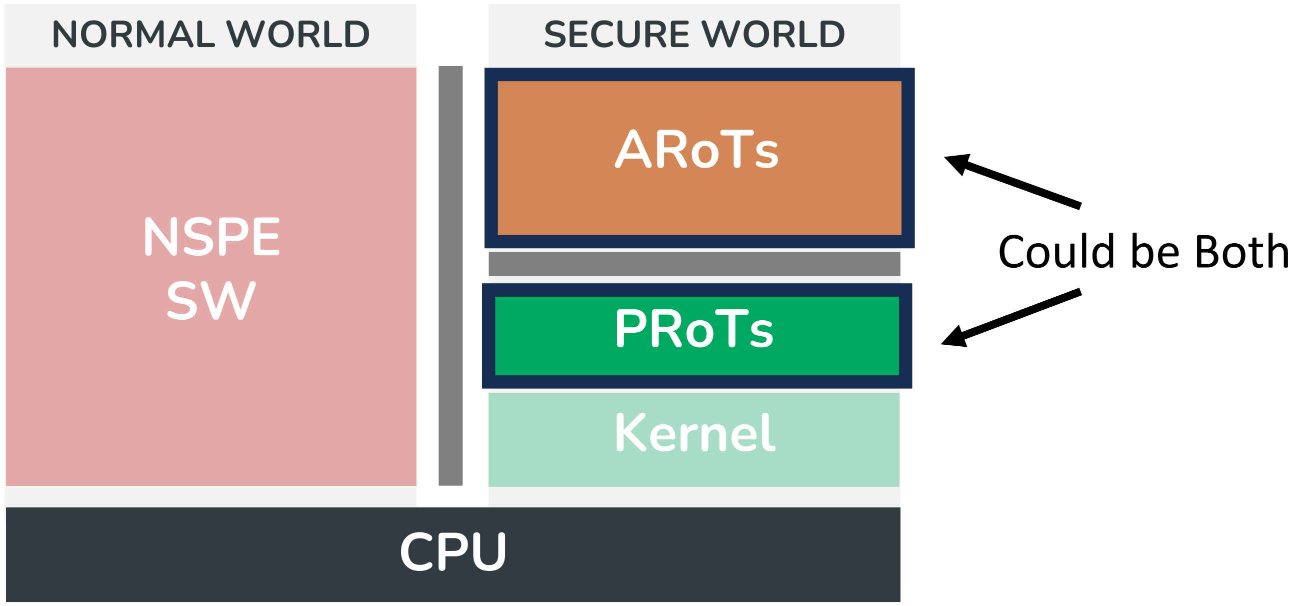


Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M

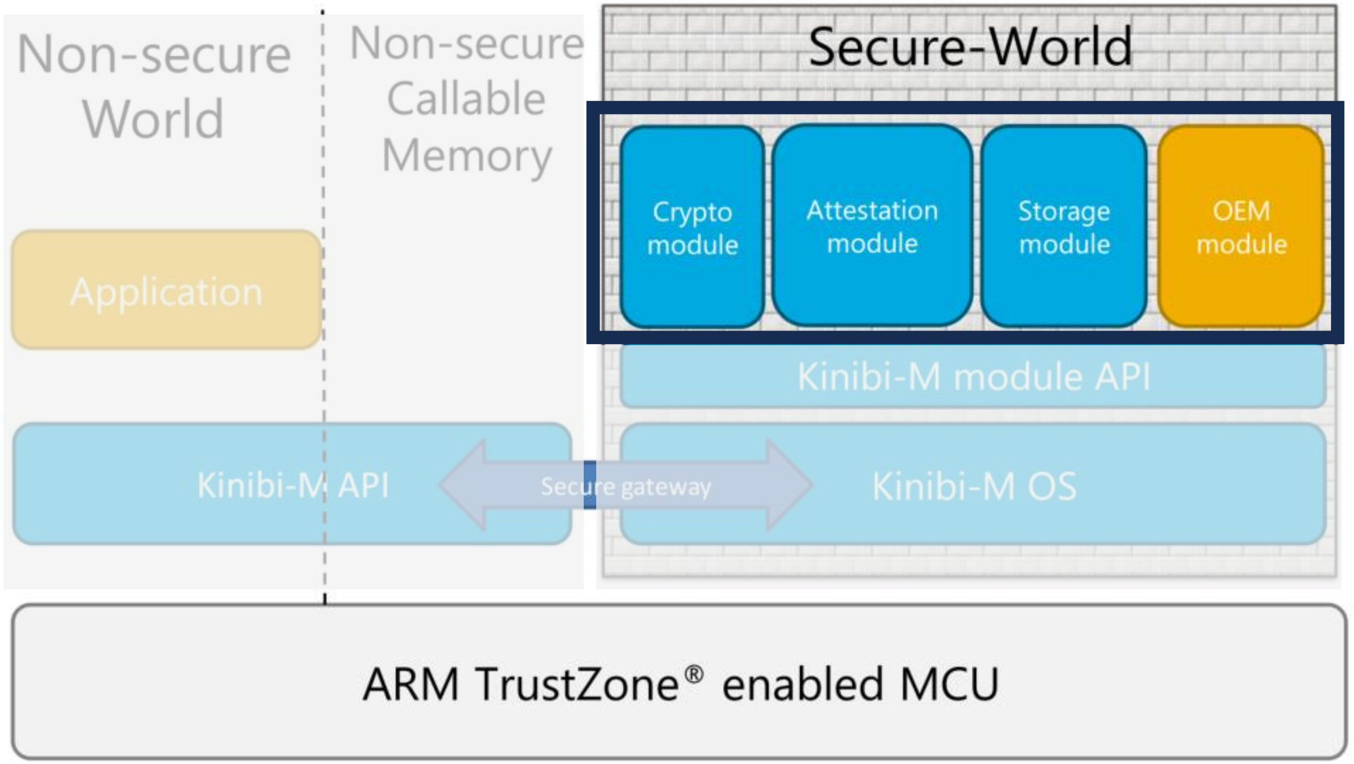
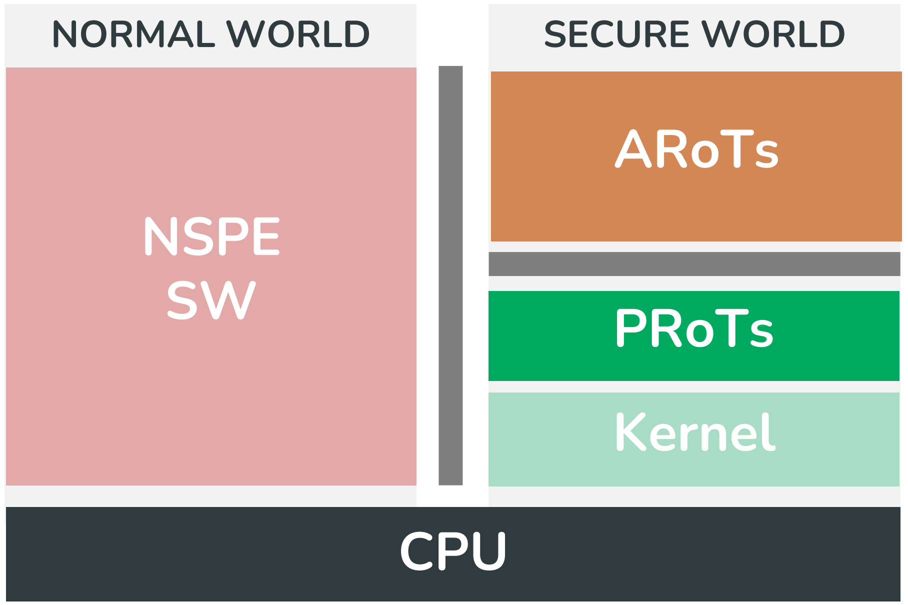


Figure 1: Kinibi-M Architecture Overview.

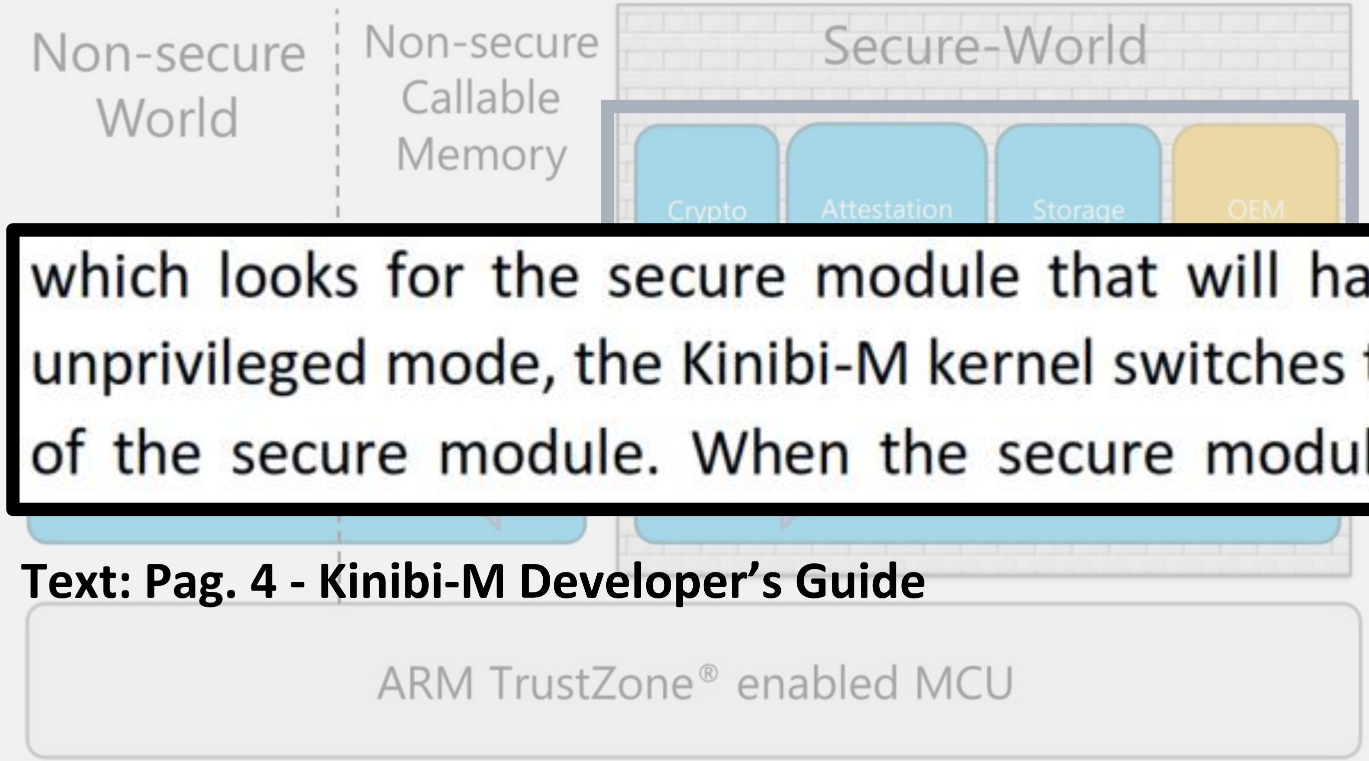
PSA Level 2



Kinibi-M Refers to PRoT and ARoT as a Secure Module

Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M

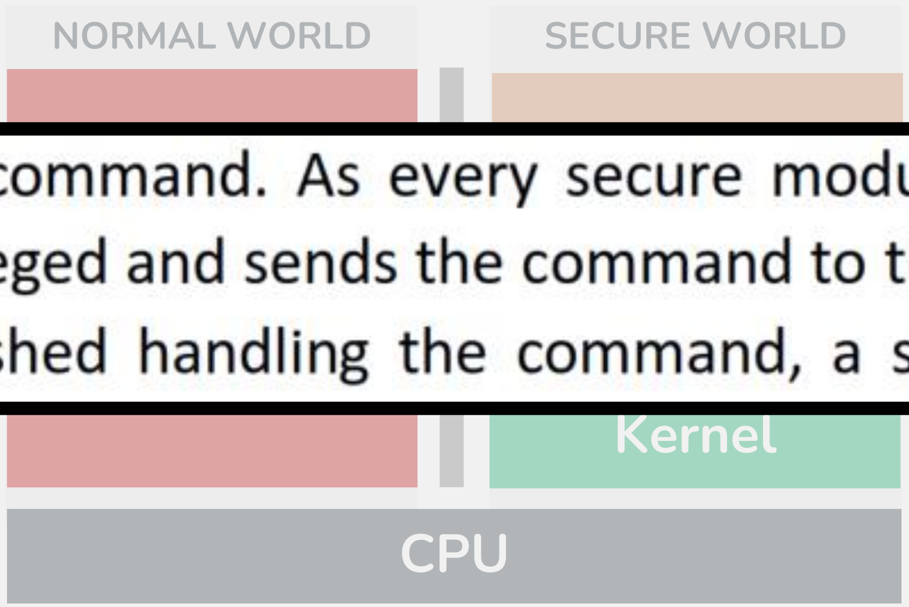


which looks for the secure module that will handle this command. As every secure module runs in unprivileged mode, the Kinibi-M kernel switches to unprivileged and sends the command to the handler of the secure module. When the secure module has finished handling the command, a system call

Text: Pag. 4 - Kinibi-M Developer's Guide

Figure 1: Kinibi-M Architecture Overview.

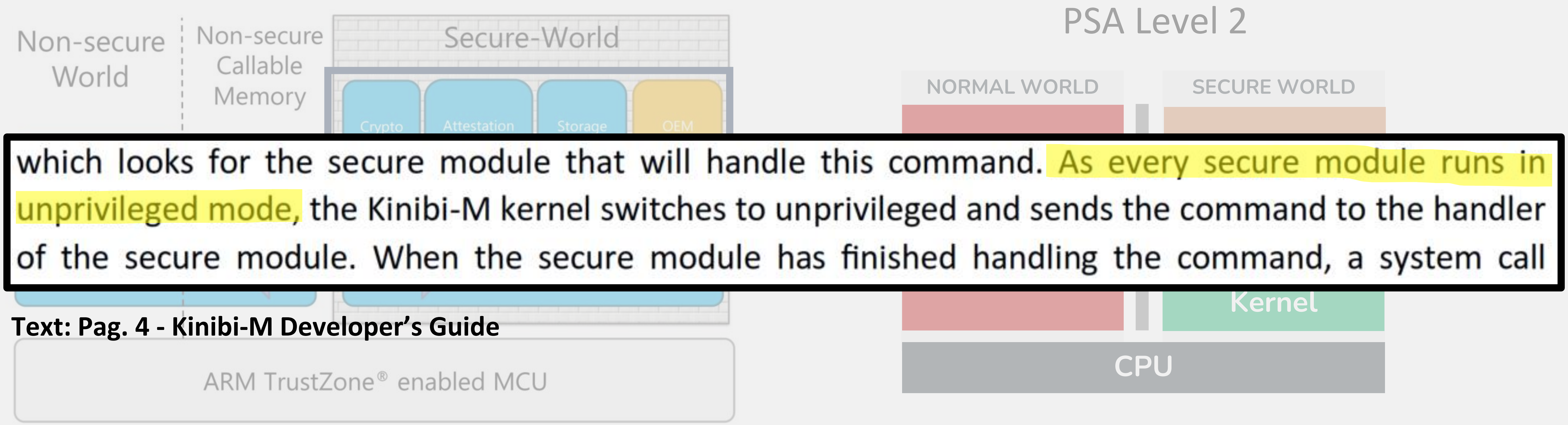
PSA Level 2



Kinibi-M Refers to PRoT and ARoT as a Secure Module

Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M



which looks for the secure module that will handle this command. As every secure module runs in unprivileged mode, the Kinibi-M kernel switches to unprivileged and sends the command to the handler of the secure module. When the secure module has finished handling the command, a system call

Text: Pag. 4 - Kinibi-M Developer's Guide

Figure 1: Kinibi-M Architecture Overview.

Kinibi-M Refers to PRoT and ARoT as a Secure Module

Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M

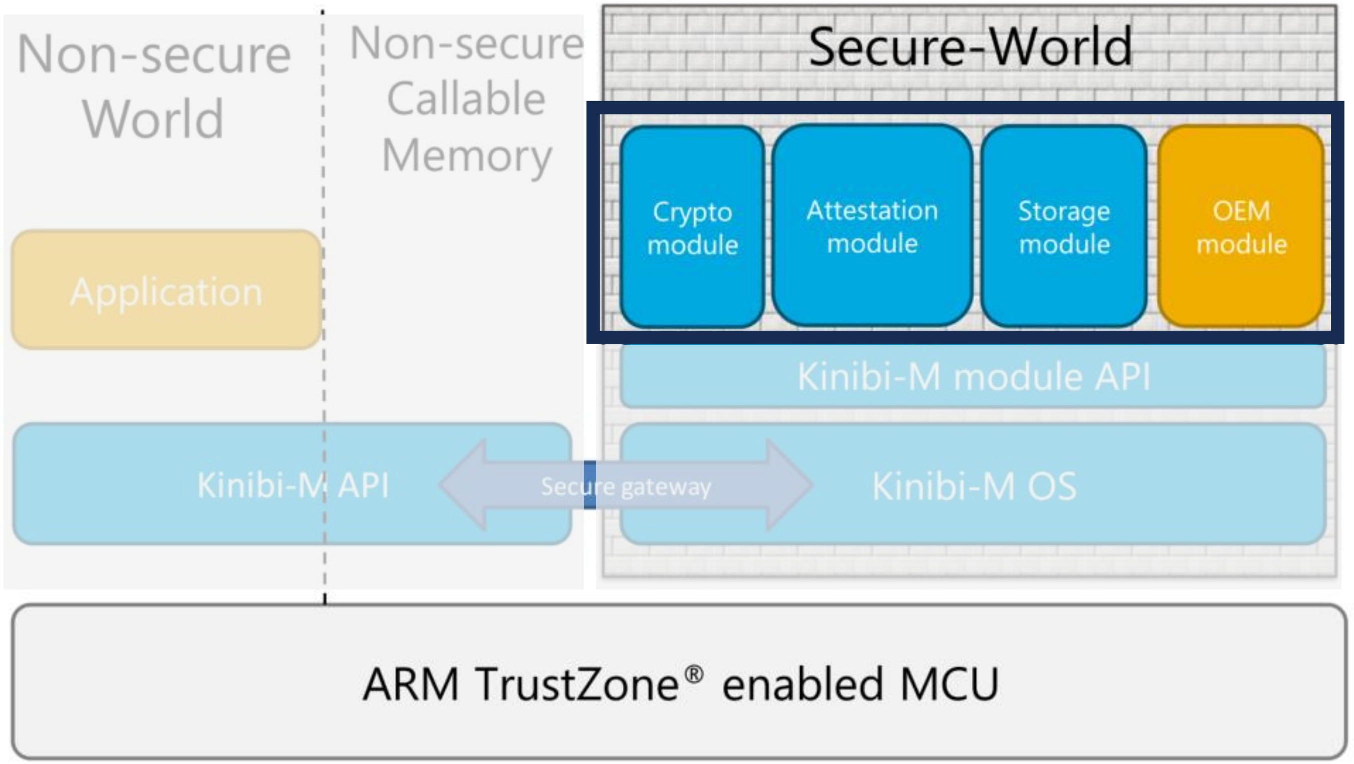
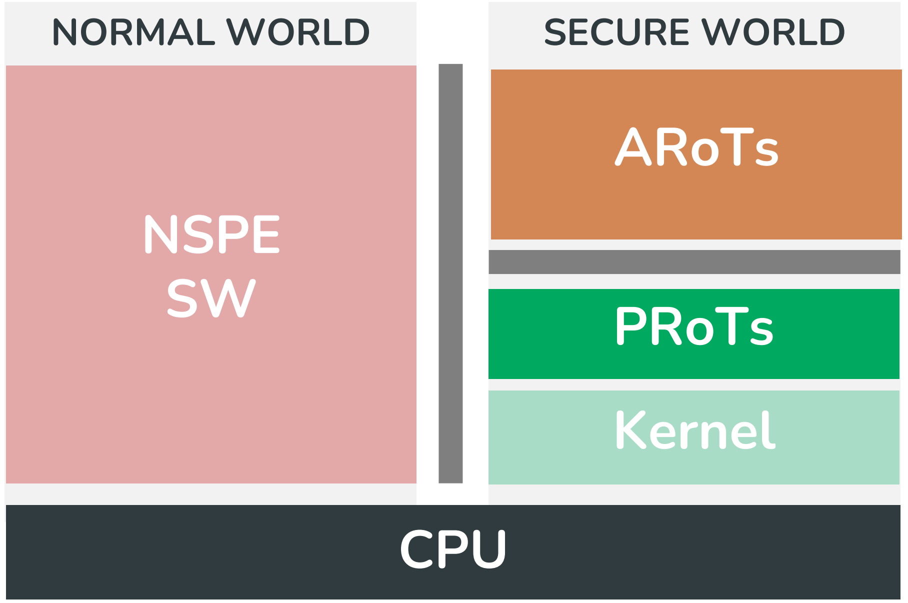


Figure 1: Kinibi-M Architecture Overview.

PSA Level 2



Kinibi-M Refers to PRoT and ARoT as a Secure Module

Image: Pag. 3 - Kinibi-M Developer’s Guide

TRUSTONIC KINIBI-M

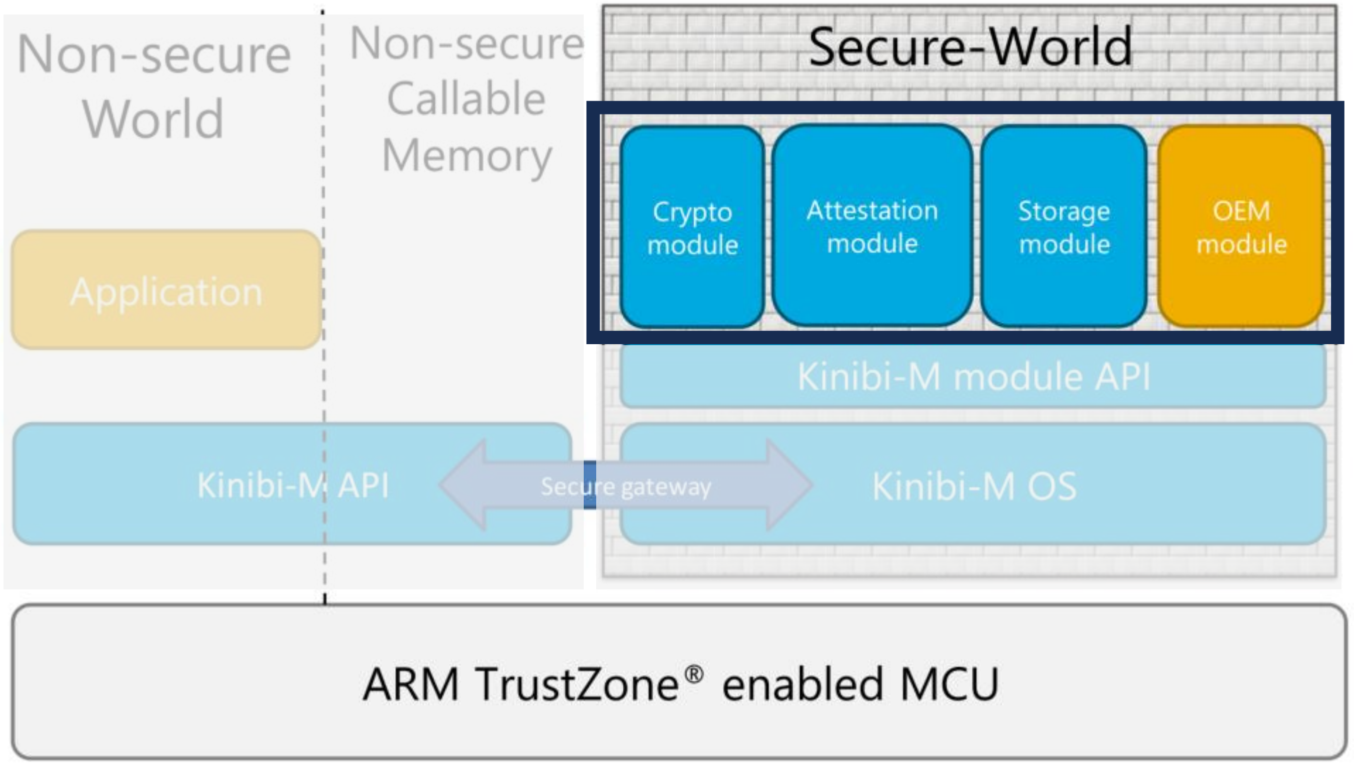
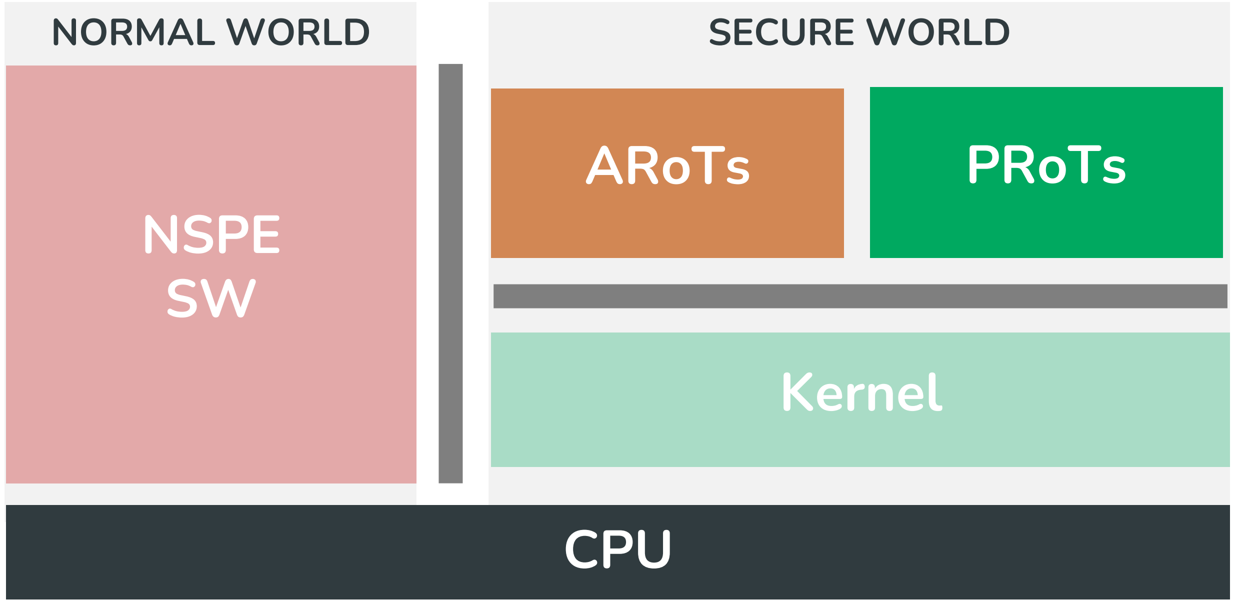


Figure 1: Kinibi-M Architecture Overview.

PSA Level 2



Kinibi-M Refers to PRoT and ARoT as a Secure Module

Image: Pag. 3 - Kinibi-M Developer’s Guide

TRUSTONIC KINIBI-M

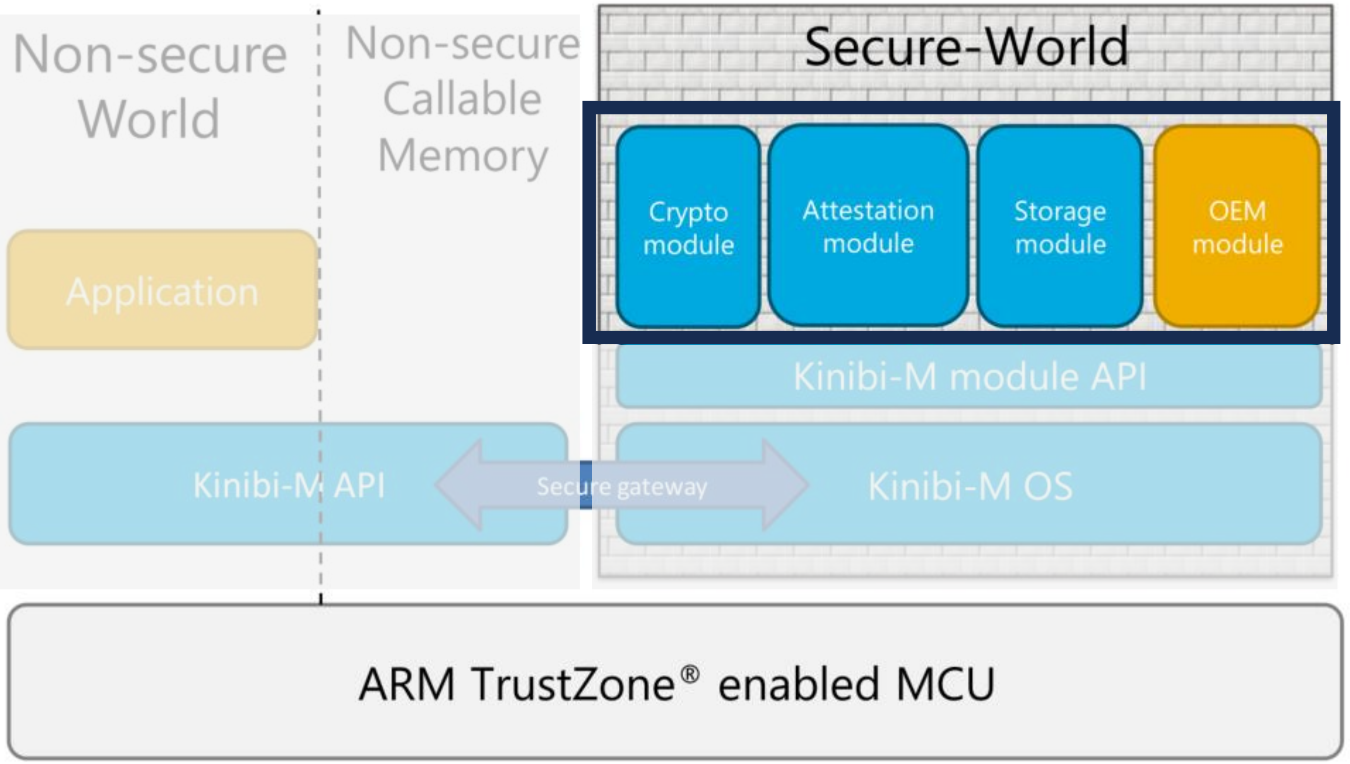
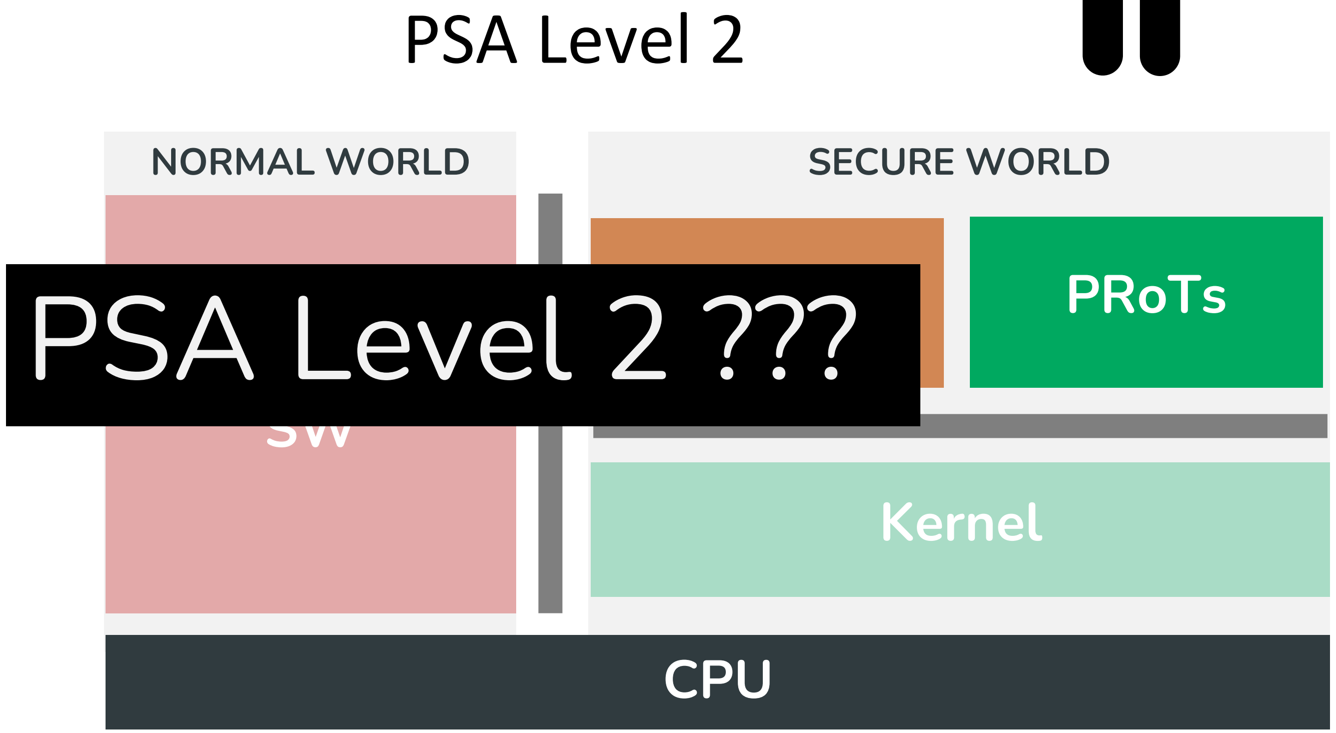


Figure 1: Kinibi-M Architecture Overview.



PSA Level 2 ???

Kinibi-M Refers to PRoT and ARoT as a Secure Module

Image: Pag. 3 - Kinibi-M Developer’s Guide

TRUSTONIC KINIBI-M

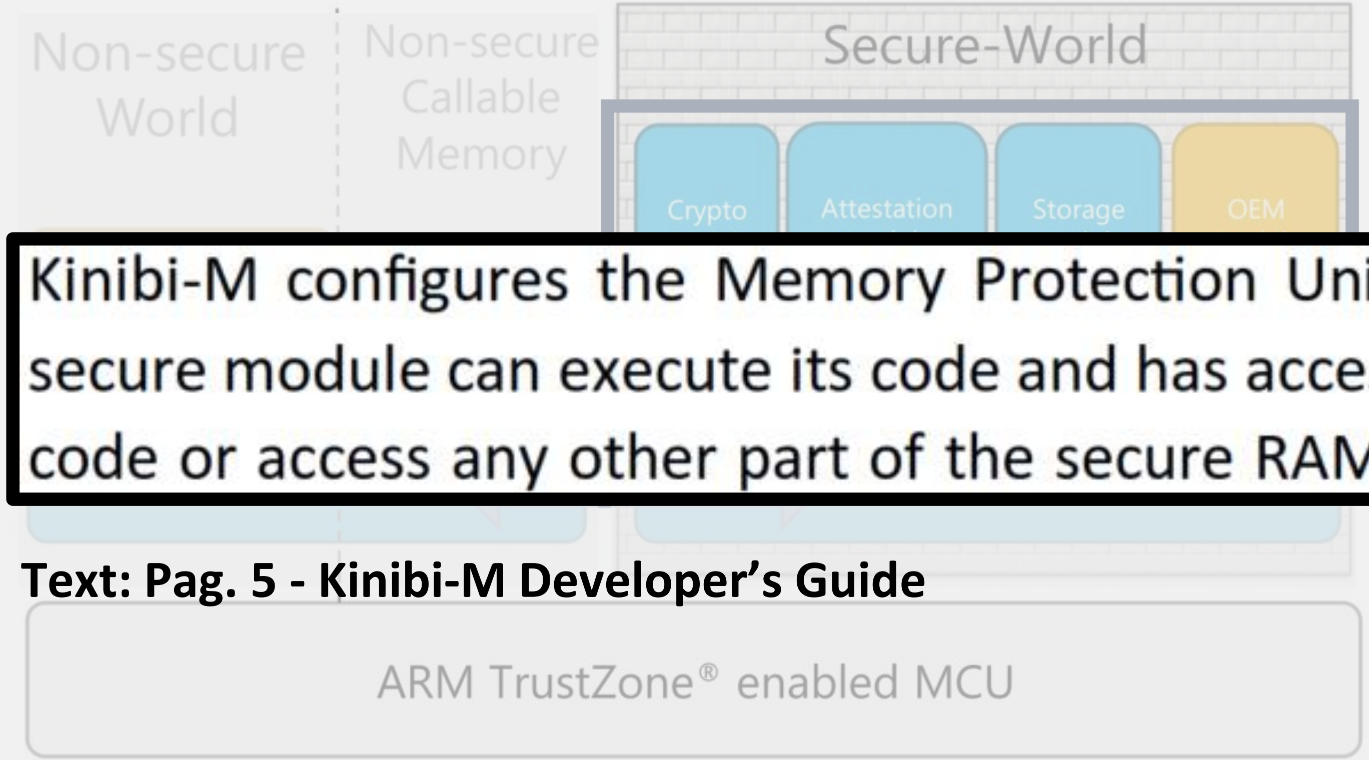
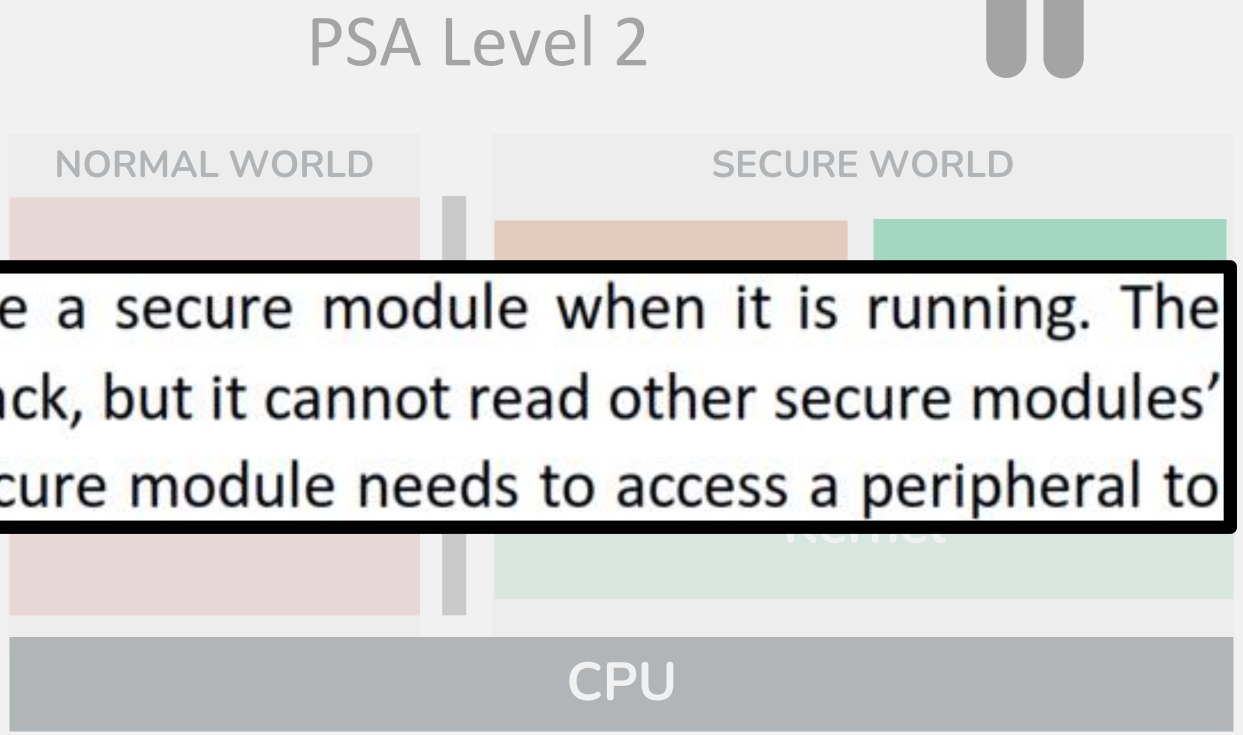


Figure 1: Kinibi-M Architecture Overview.



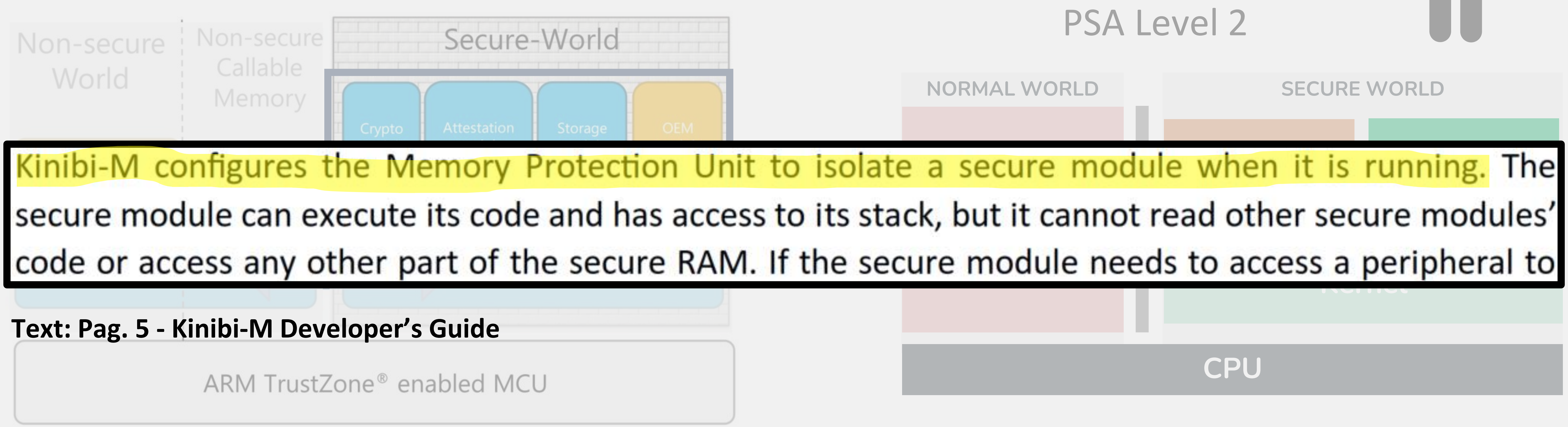
Kinibi-M configures the Memory Protection Unit to isolate a secure module when it is running. The secure module can execute its code and has access to its stack, but it cannot read other secure modules' code or access any other part of the secure RAM. If the secure module needs to access a peripheral to

Text: Pag. 5 - Kinibi-M Developer's Guide

Kinibi-M Refers to PRoT and ARoT as a Secure Module

Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M



Kinibi-M configures the Memory Protection Unit to isolate a secure module when it is running. The secure module can execute its code and has access to its stack, but it cannot read other secure modules' code or access any other part of the secure RAM. If the secure module needs to access a peripheral to

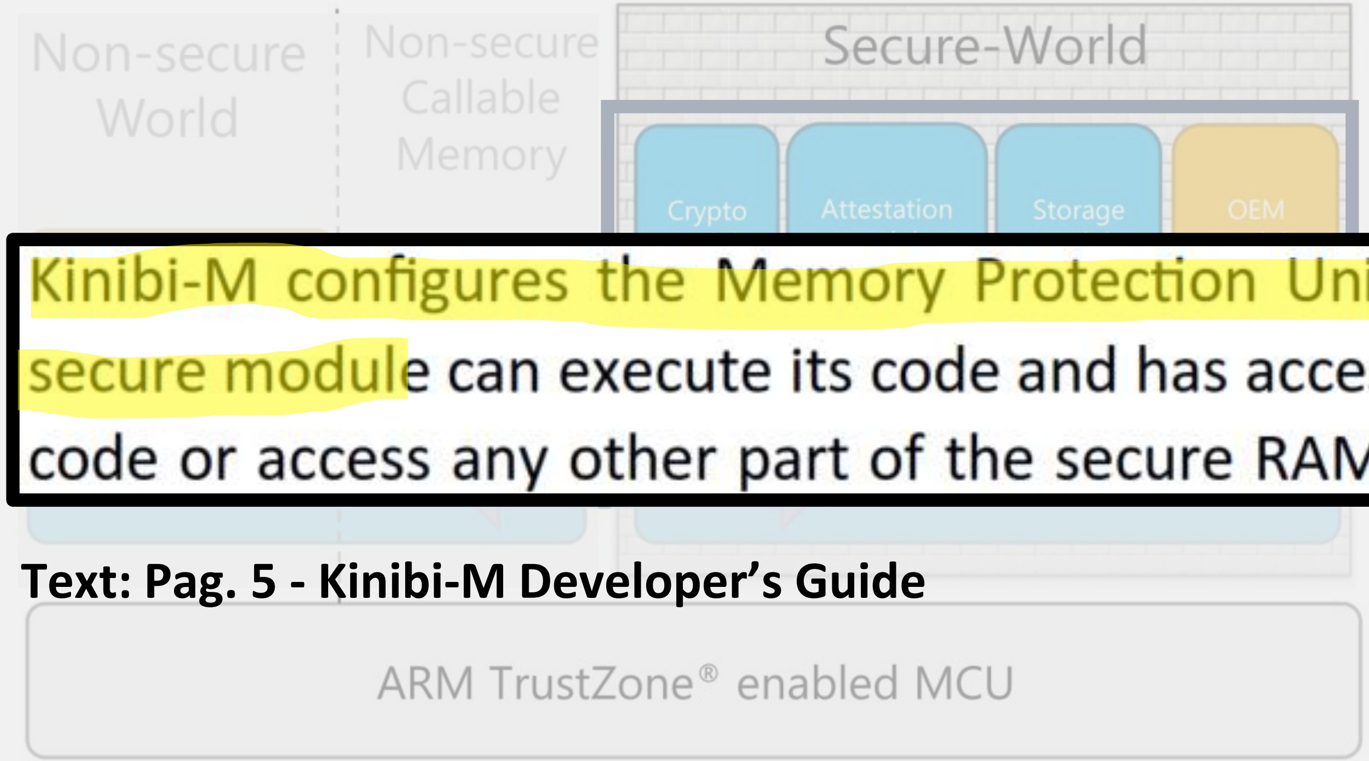
Text: Pag. 5 - Kinibi-M Developer's Guide

Figure 1: Kinibi-M Architecture Overview.

Kinibi-M Refers to PRoT and ARoT as a Secure Module

Image: Pag. 3 - Kinibi-M Developer's Guide

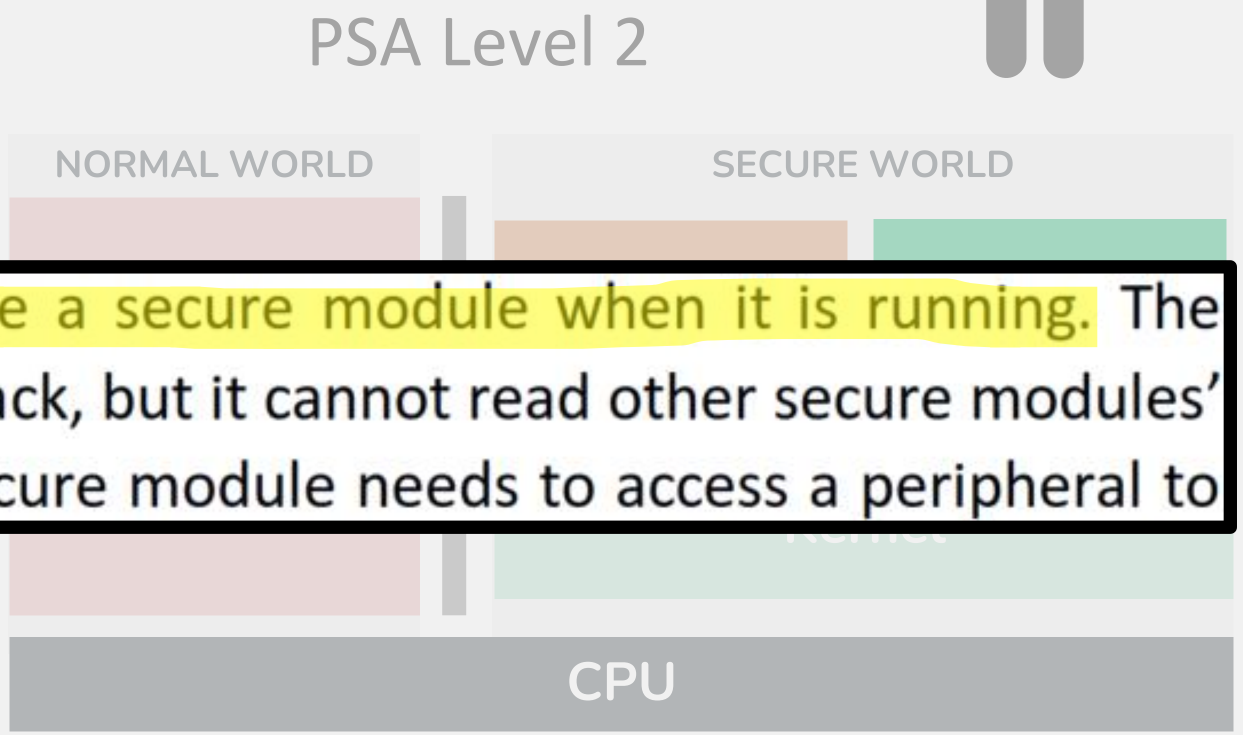
TRUSTONIC KINIBI-M



Kinibi-M configures the Memory Protection Unit to isolate a secure module when it is running. The secure module can execute its code and has access to its stack, but it cannot read other secure modules' code or access any other part of the secure RAM. If the secure module needs to access a peripheral to

Text: Pag. 5 - Kinibi-M Developer's Guide

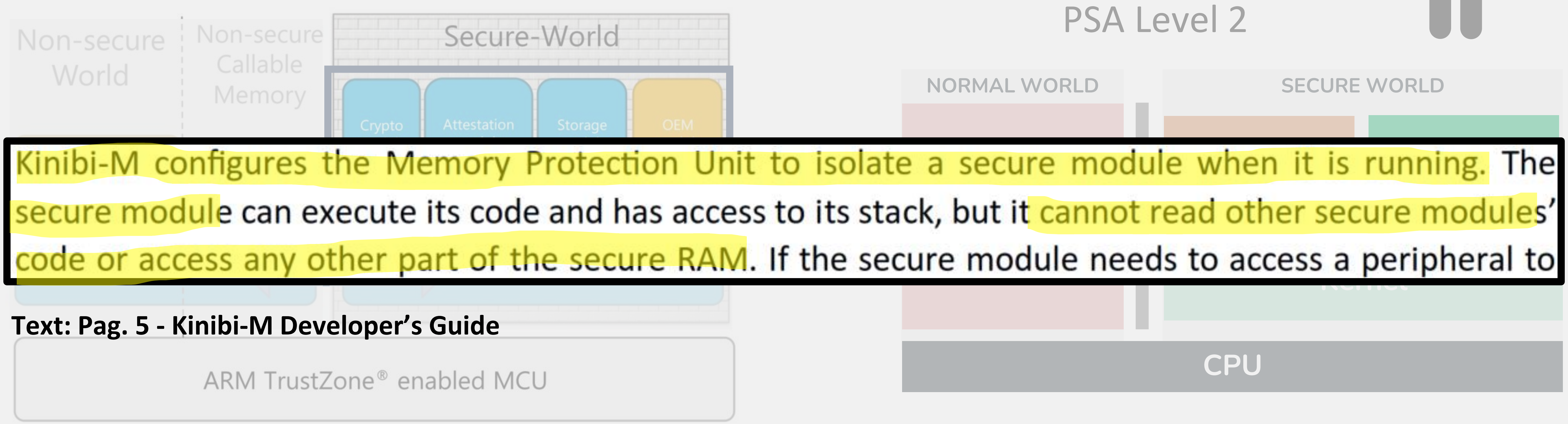
Figure 1: Kinibi-M Architecture Overview.



Kinibi-M Refers to PRoT and ARoT as a Secure Module

Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M



Kinibi-M configures the Memory Protection Unit to isolate a secure module when it is running. The secure module can execute its code and has access to its stack, but it cannot read other secure modules' code or access any other part of the secure RAM. If the secure module needs to access a peripheral to

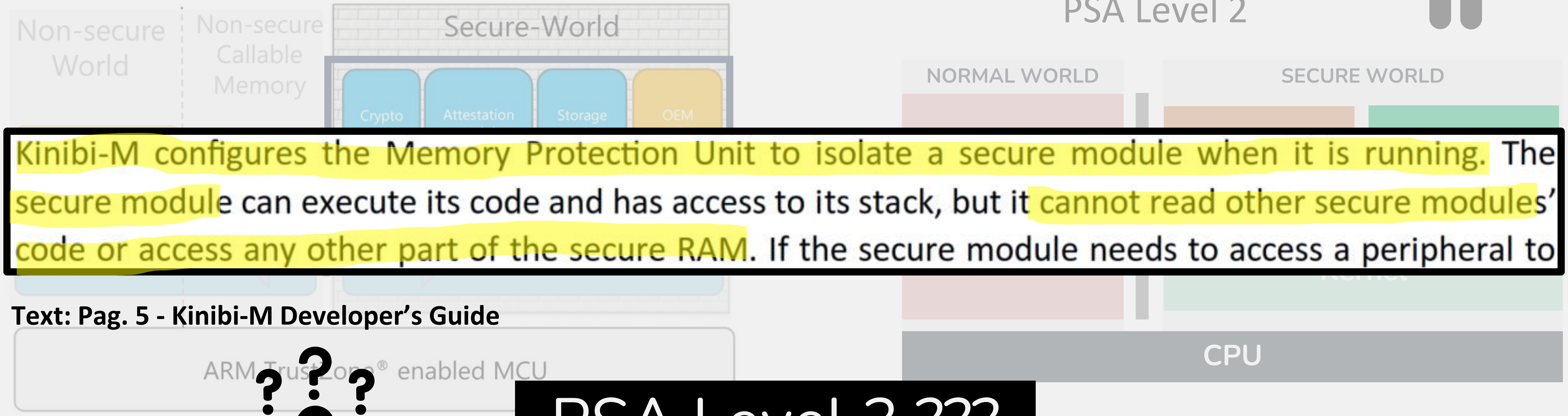
Text: Pag. 5 - Kinibi-M Developer's Guide

Figure 1: Kinibi-M Architecture Overview.

Kinibi-M Refers to PRoT and ARoT as a Secure Module

Image: Pag. 3 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M



Kinibi-M configures the Memory Protection Unit to isolate a secure module when it is running. The secure module can execute its code and has access to its stack, but it cannot read other secure modules' code or access any other part of the secure RAM. If the secure module needs to access a peripheral to

Text: Pag. 5 - Kinibi-M Developer's Guide

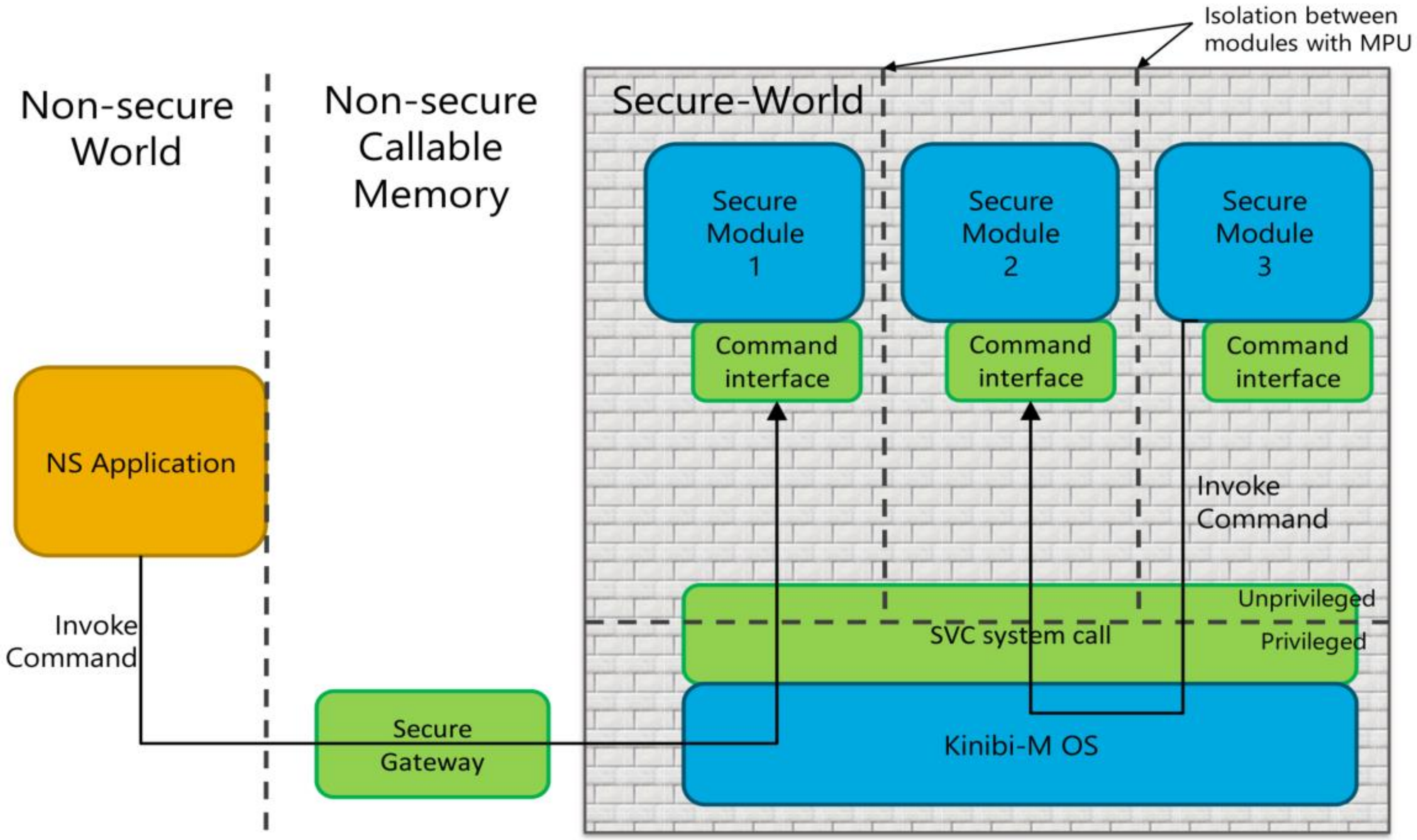
Figure 1 … chitecture Ove

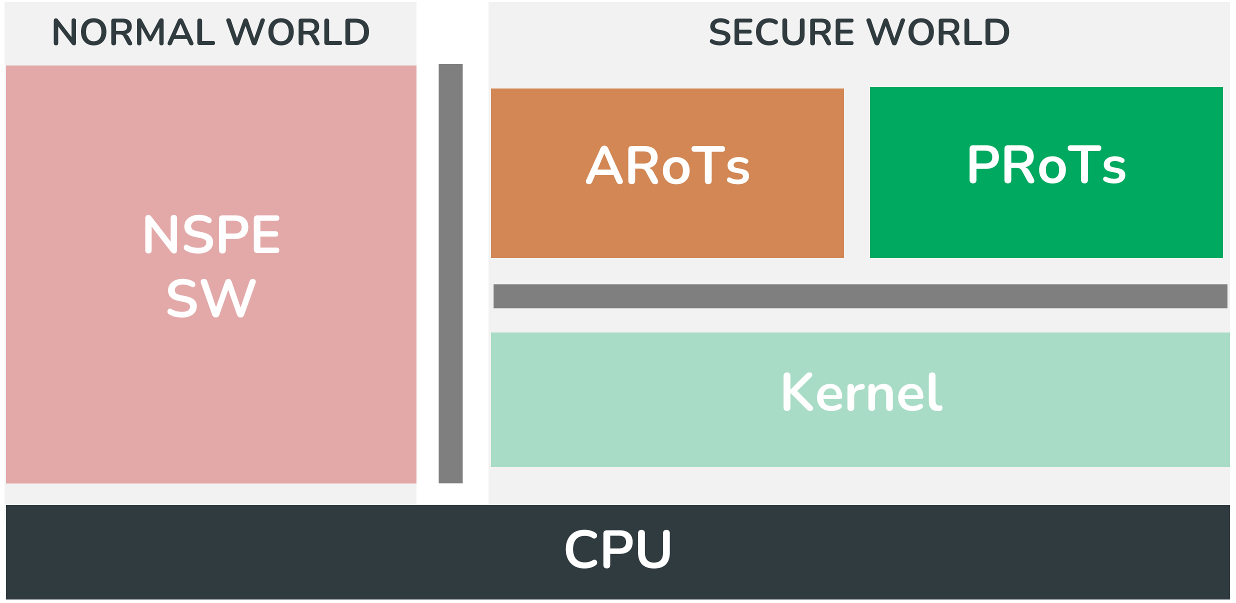
PSA Level 3 ???

Kinibi-M Refers to PRoT and ARoT as a Secure Module

Image: Pag. 3 - Kinibi-M Developer's Guide

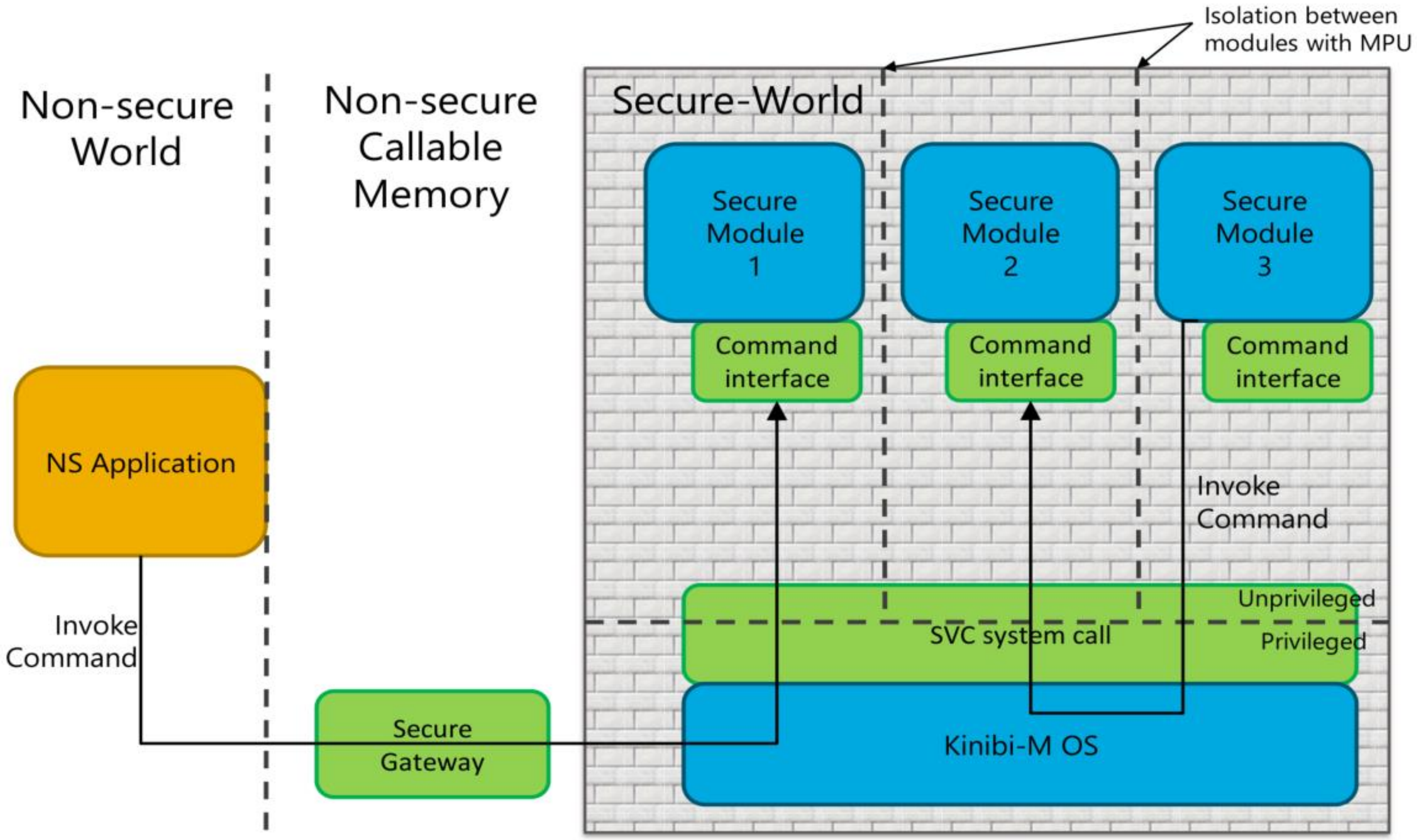
TRUSTONIC KINIBI-M

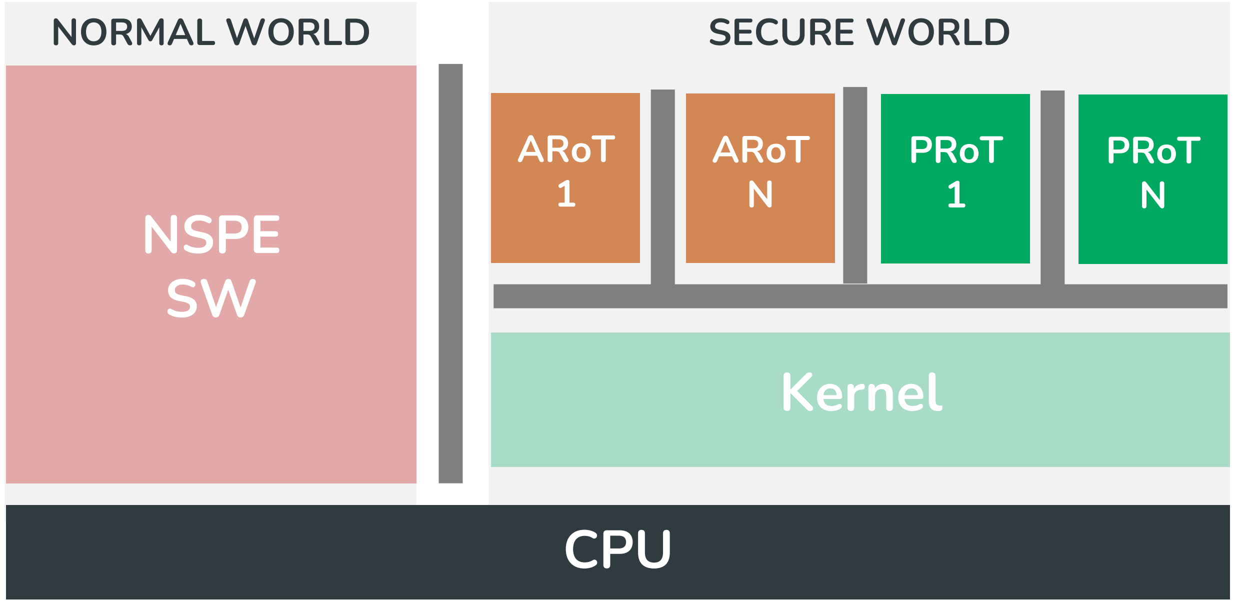




Text: Pag. 4 - Kinibi-M Developer’s Guide

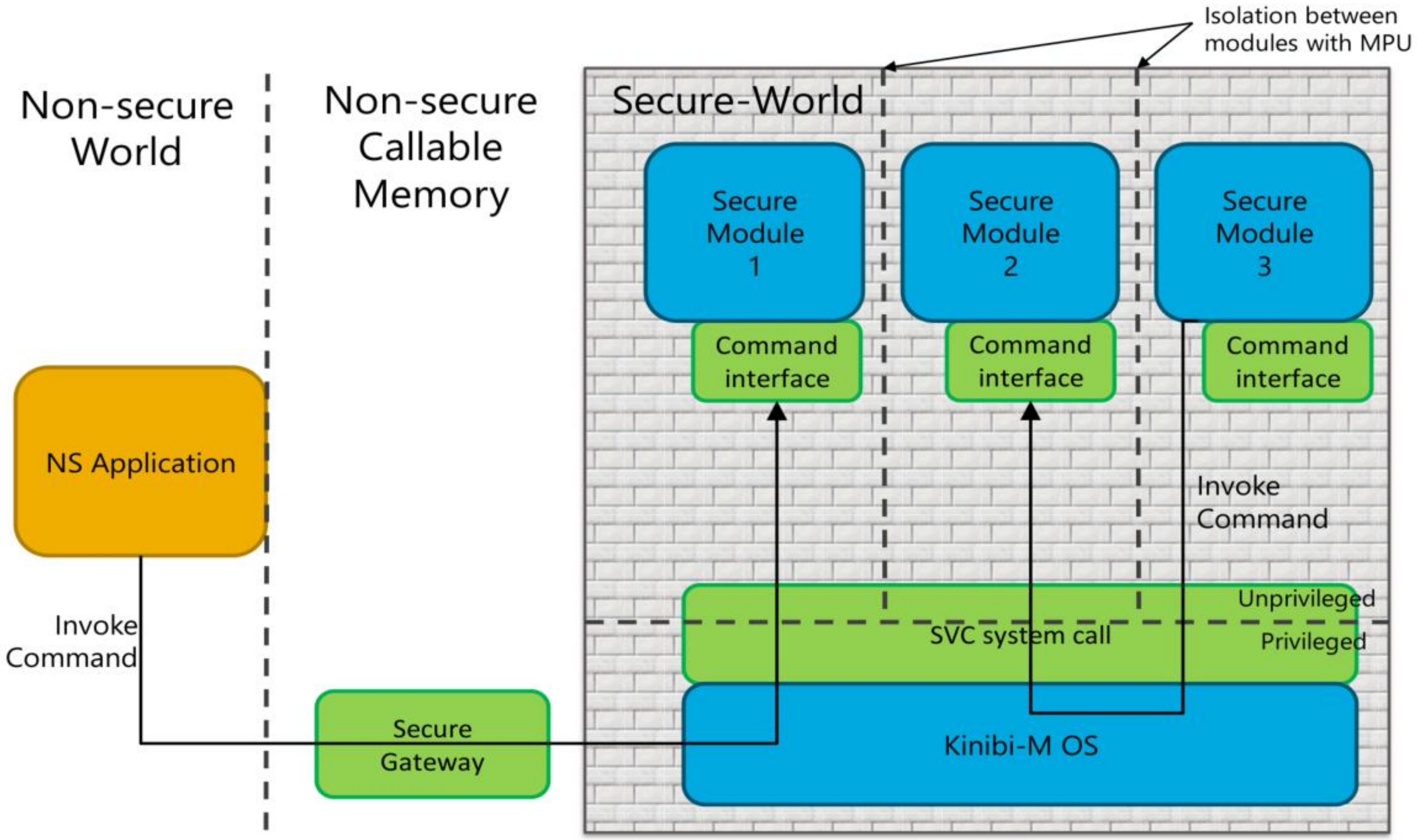
TRUSTONIC KINIBI-M

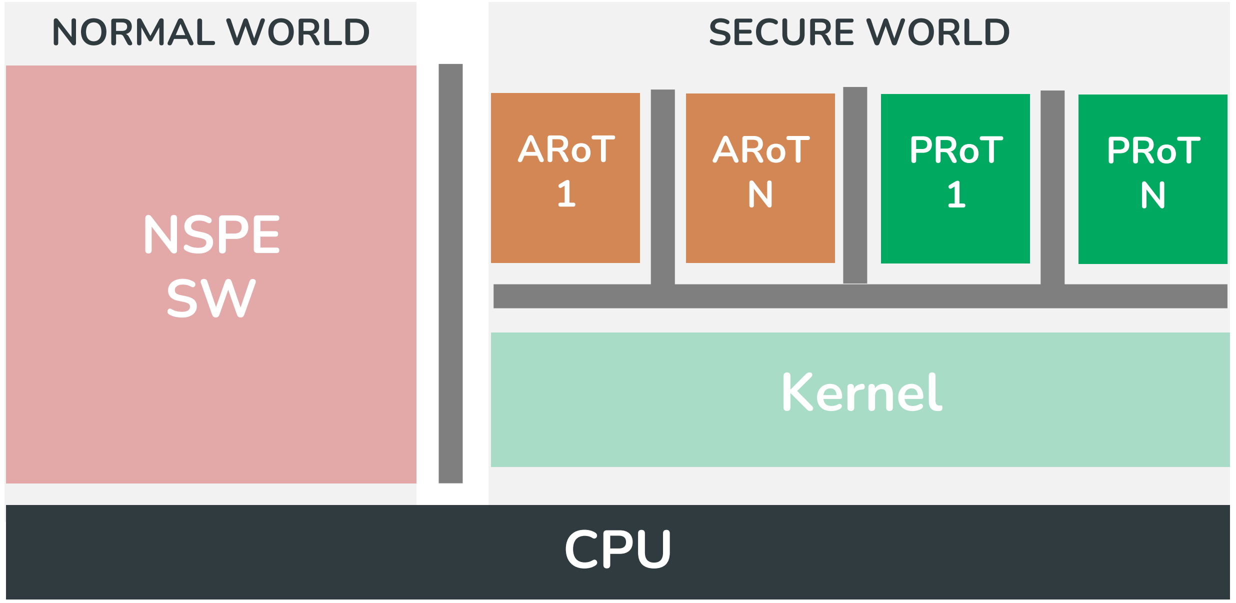




Text: Pag. 4 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M

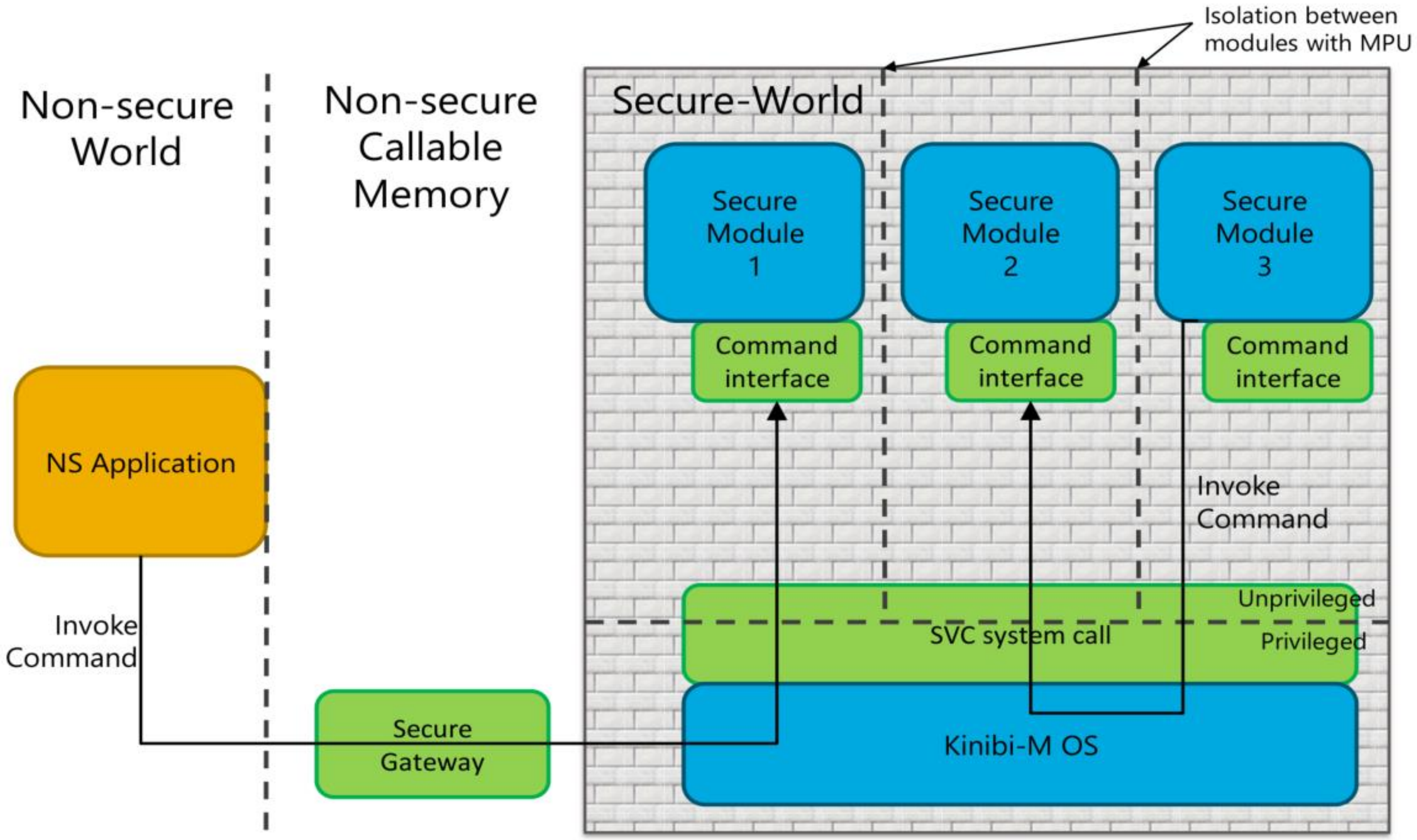


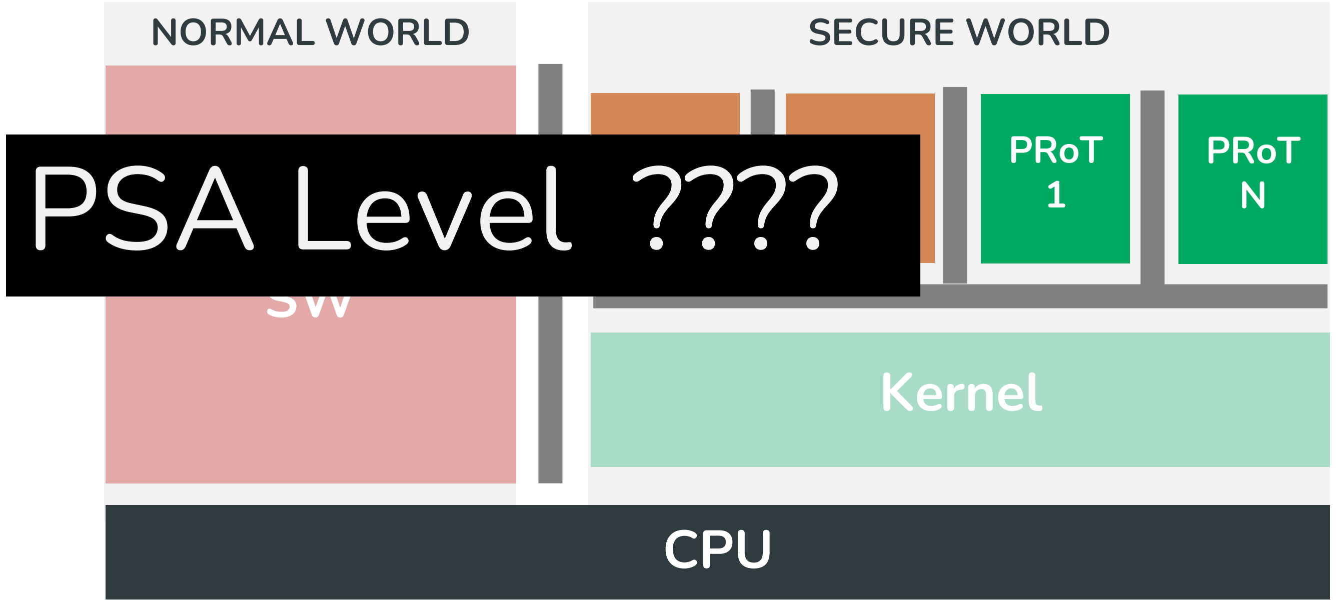


Microkernel-like Architecture

Text: Pag. 4 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M



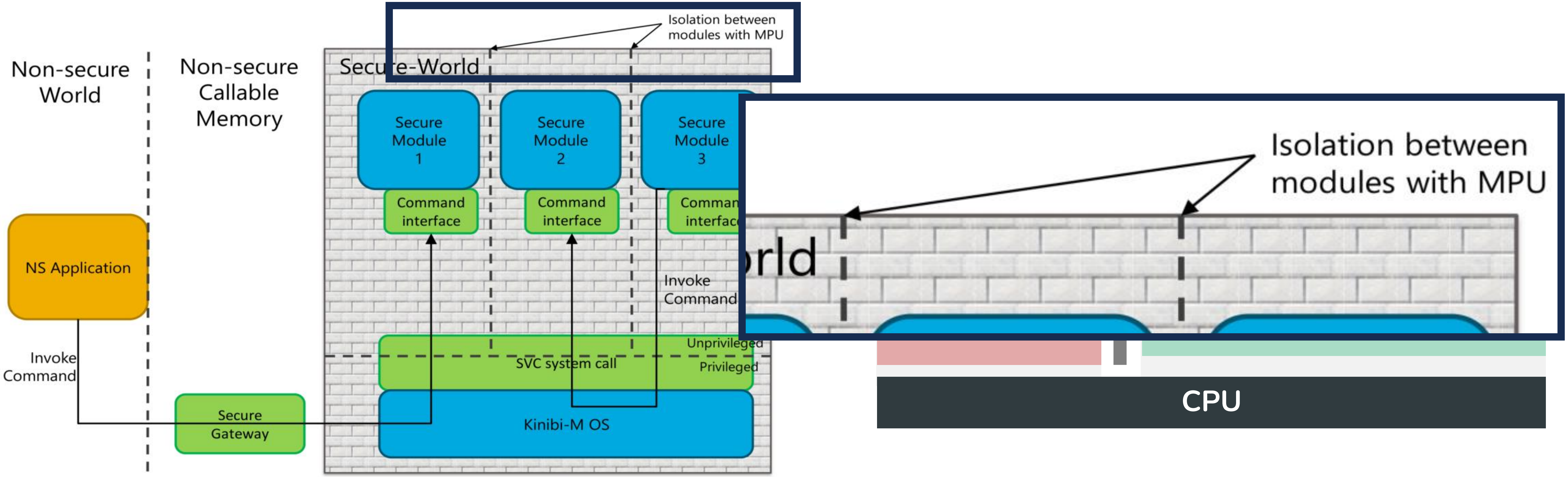


PSA Level ????

Microkernel-like Architecture

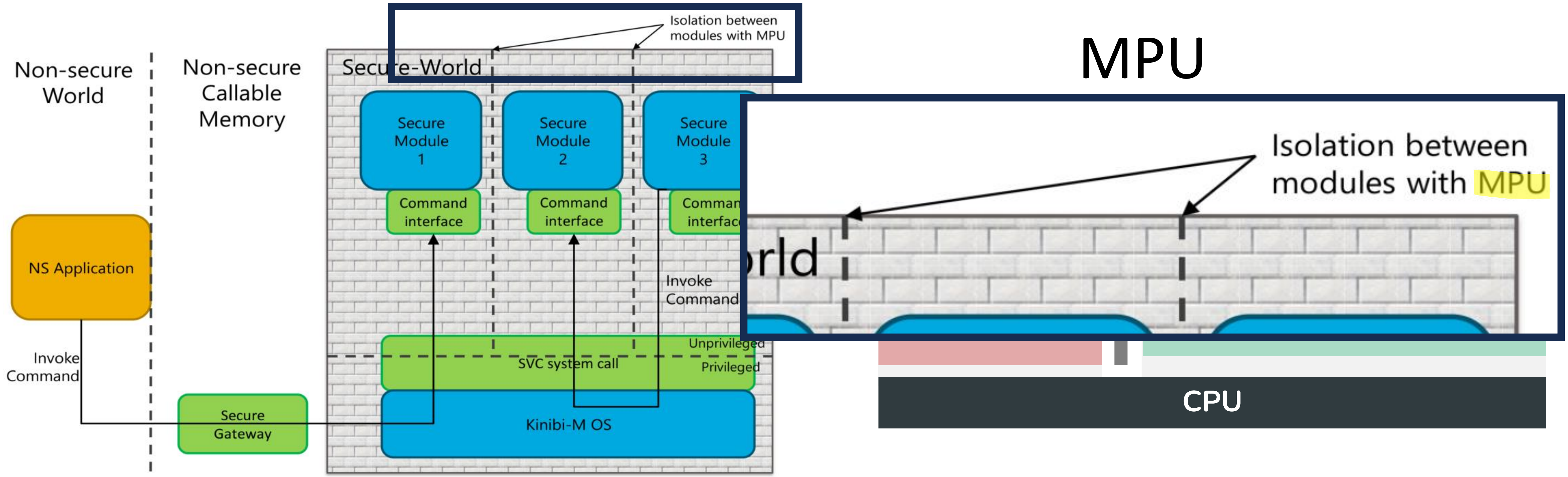
Text: Pag. 4 - Kinibi-M Developer’s Guide

TRUSTONIC KINIBI-M



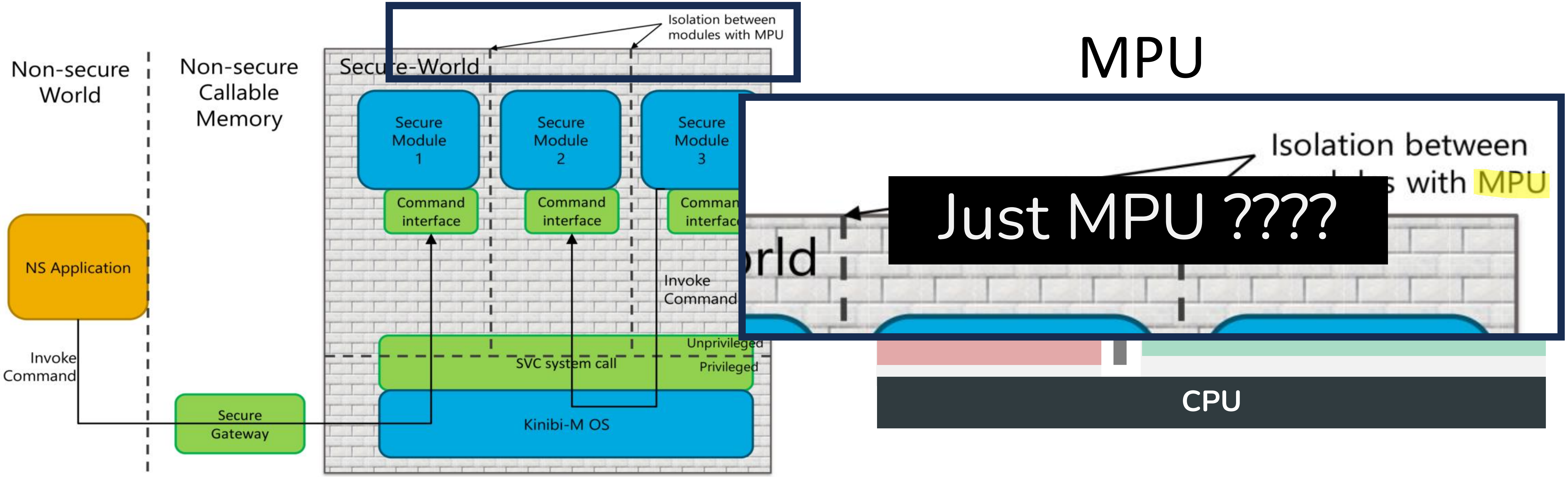
Text: Pag. 4 - Kinibi-M Developer’s Guide

TRUSTONIC KINIBI-M



Text: Pag. 4 - Kinibi-M Developer’s Guide

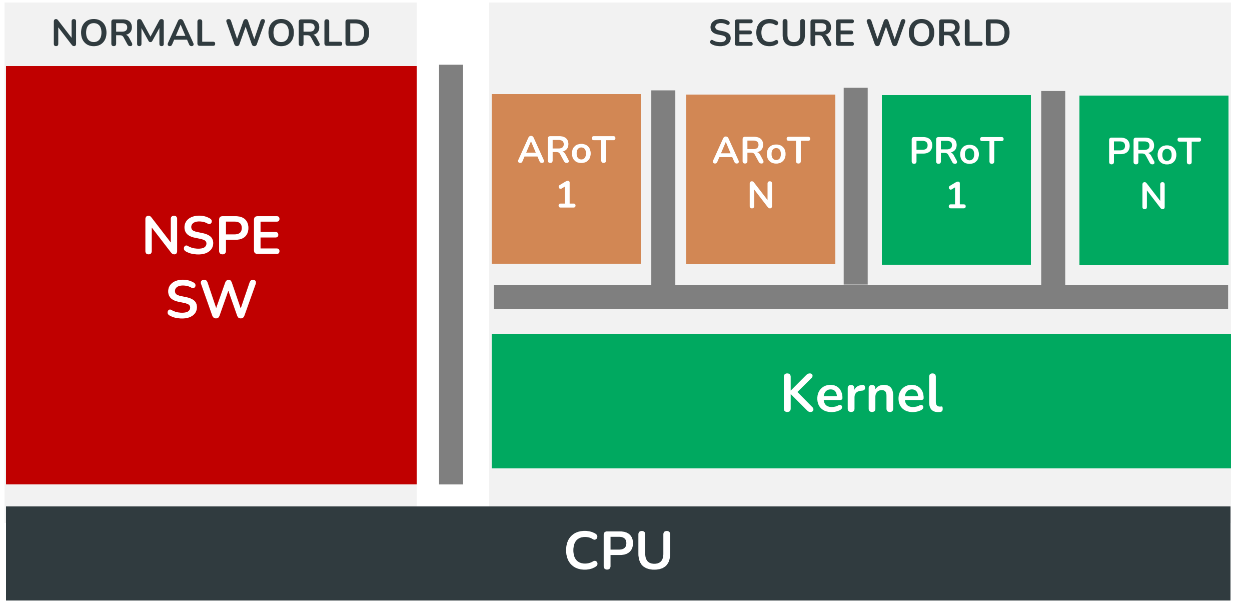
TRUSTONIC KINIBI-M



Just MPU ????

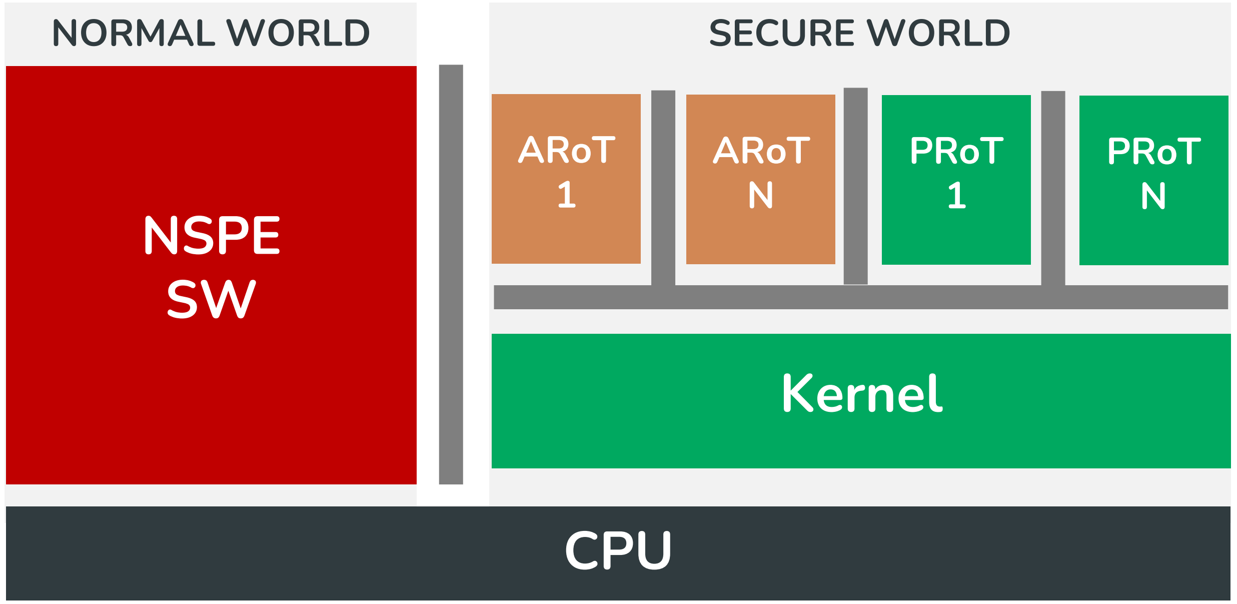
Text: Pag. 4 - Kinibi-M Developer's Guide

TRUSTONIC KINIBI-M



Kinibi-M Architecture

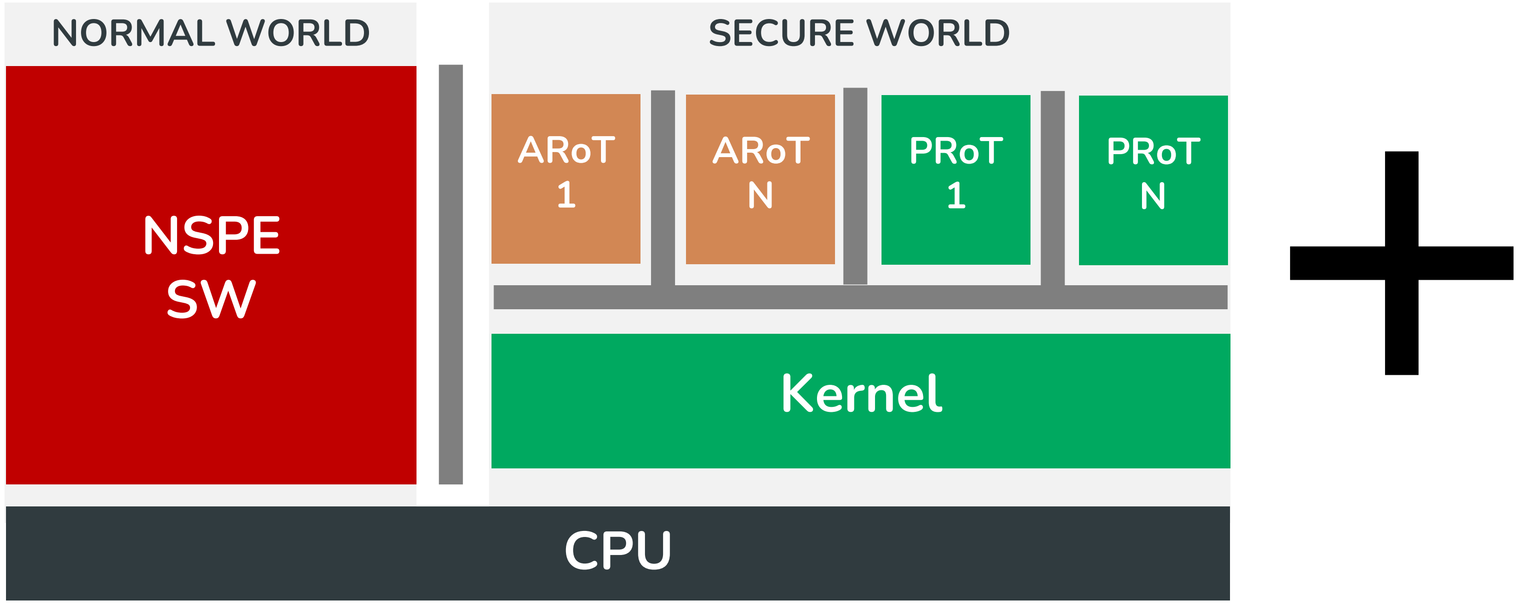
TRUSTONIC KINIBI-M



Kinibi-M Architecture

Seems Probably More then PSA Level 3

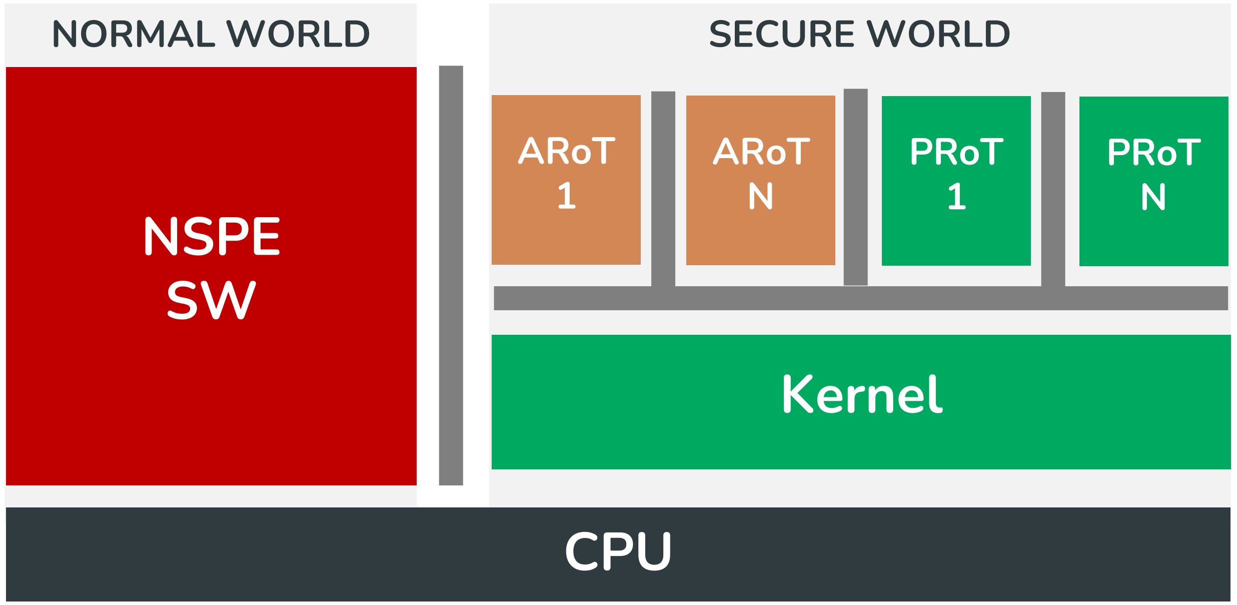
TRUSTONIC KINIBI-M



Kinibi-M Architecture

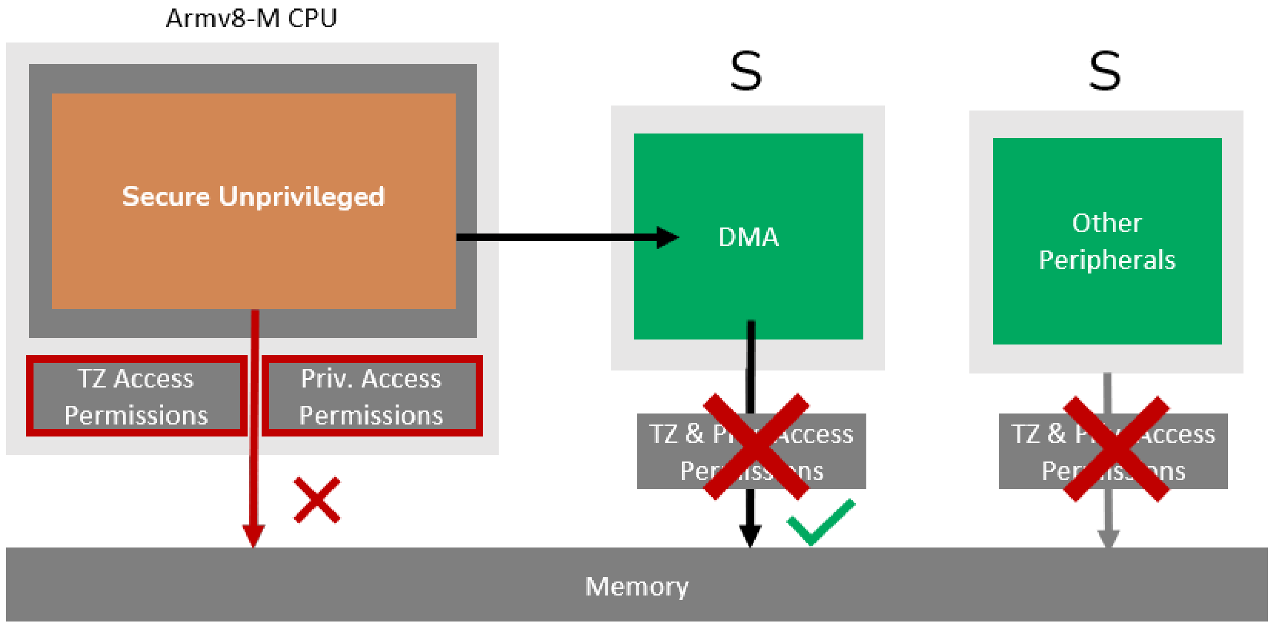
Seems Probably More then PSA Level 3

TRUSTONIC KINIBI-M



Kinibi-M Architecture

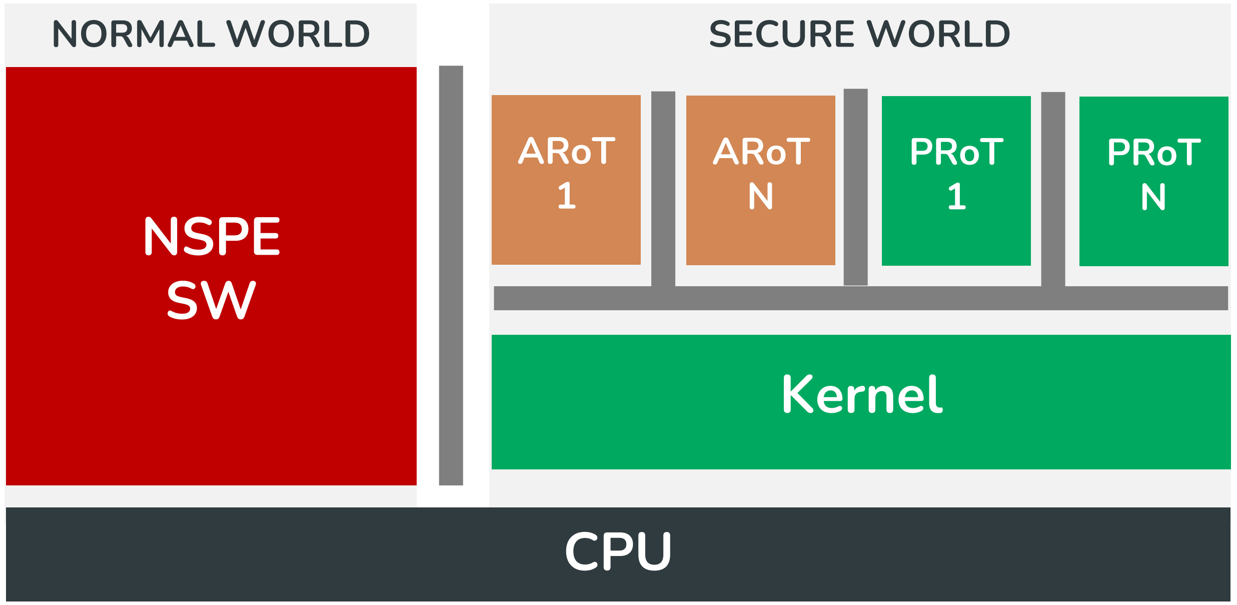
+

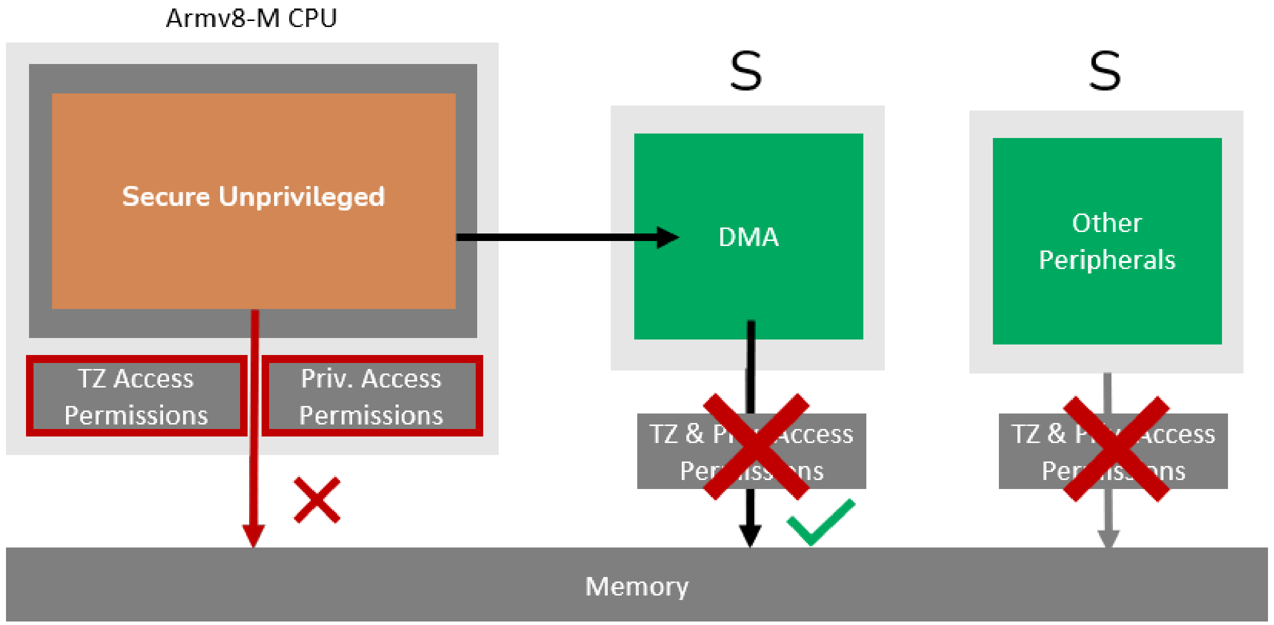


Microchip SAML11

Seems Probably More then PSA Level 3

TRUSTONIC KINIBI-M



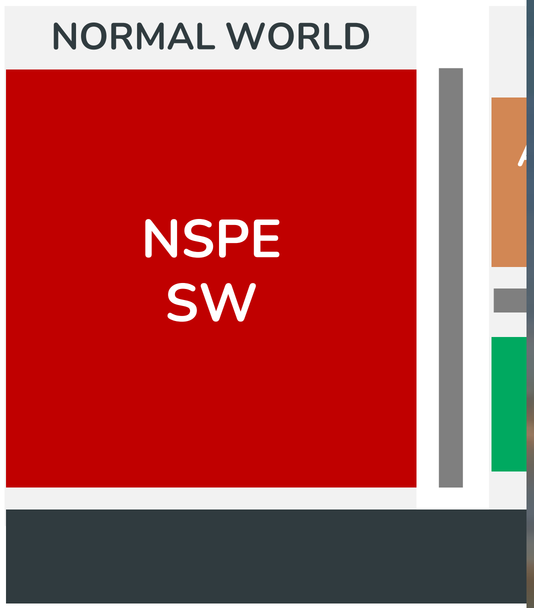


Kinibi-M Architecture

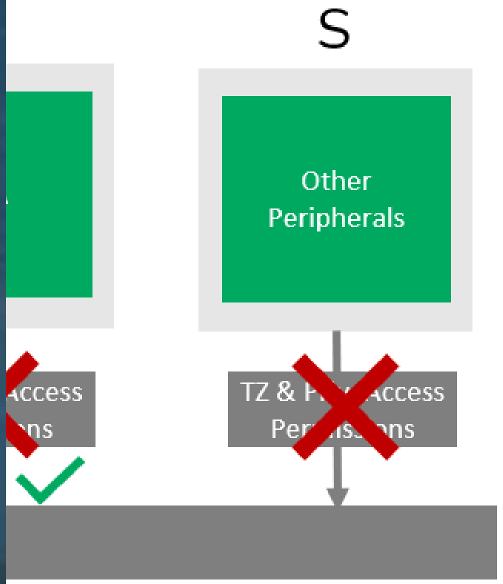
Seems Probably More then PSA Level 3

Microchip SAML11

Only PSA Level 1 & No MPC


NORMAL WORLD

NSPE
SW


S

Other
Peripherals

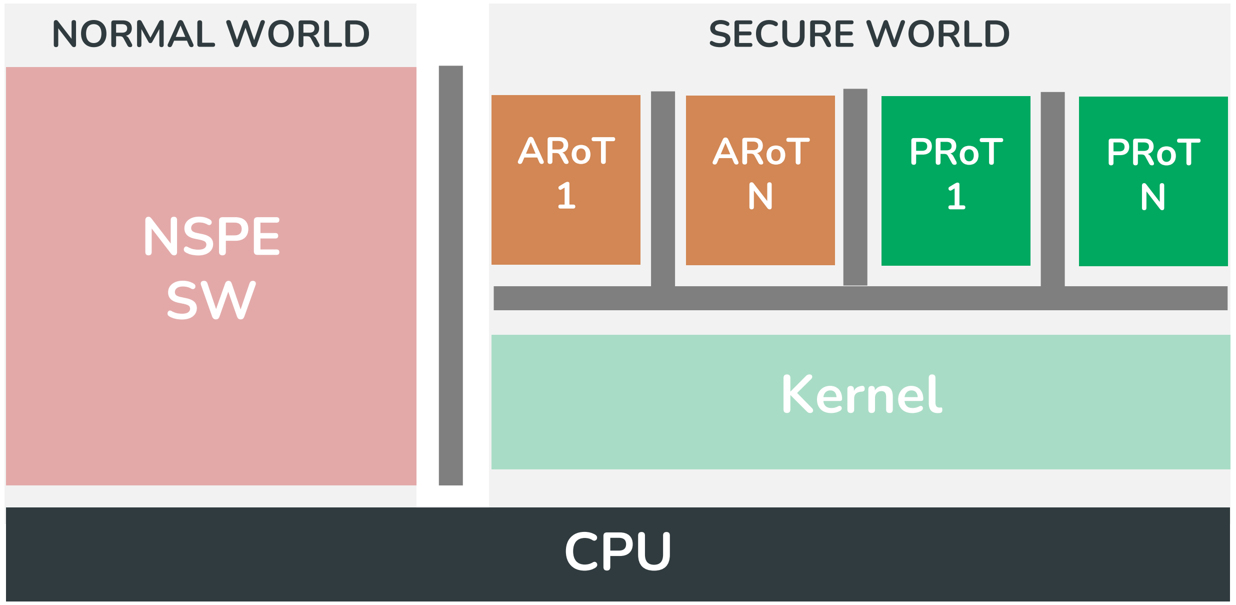
ccess
ns

TZ & P ccess
Per ns

Kinibi-M ML11

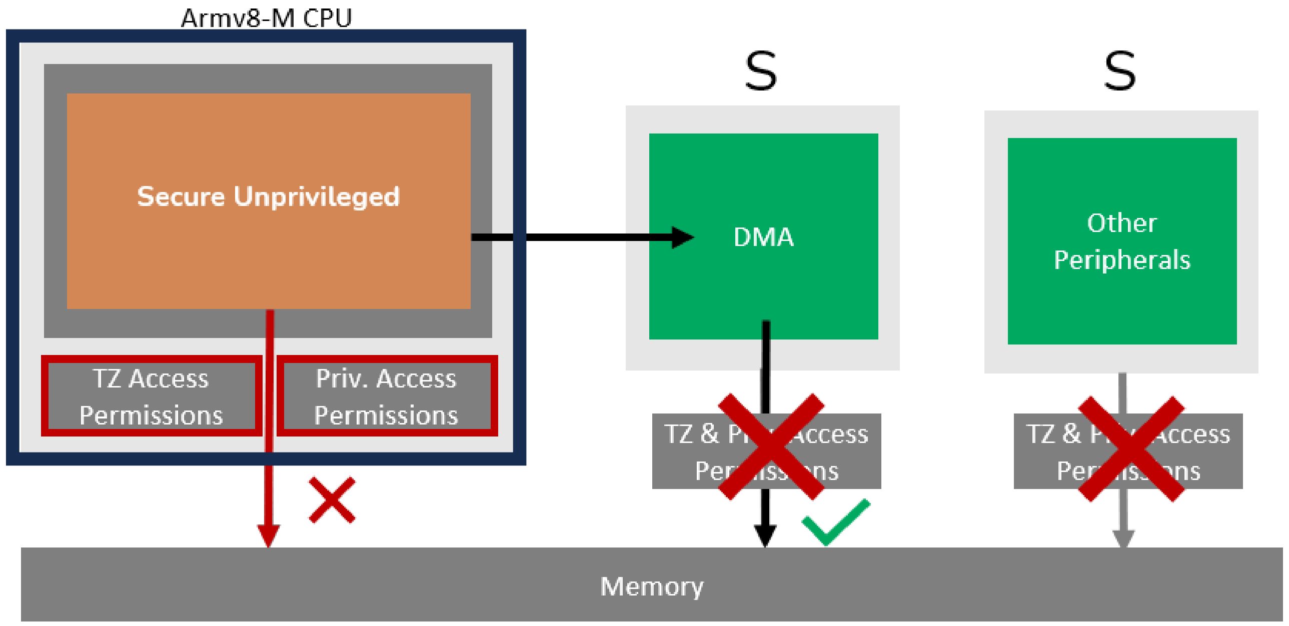
Seems Probably & No MPC

TRUSTONIC KINIBI-M



Kinibi-M Architecture

Seems Probably More then PSA Level 3

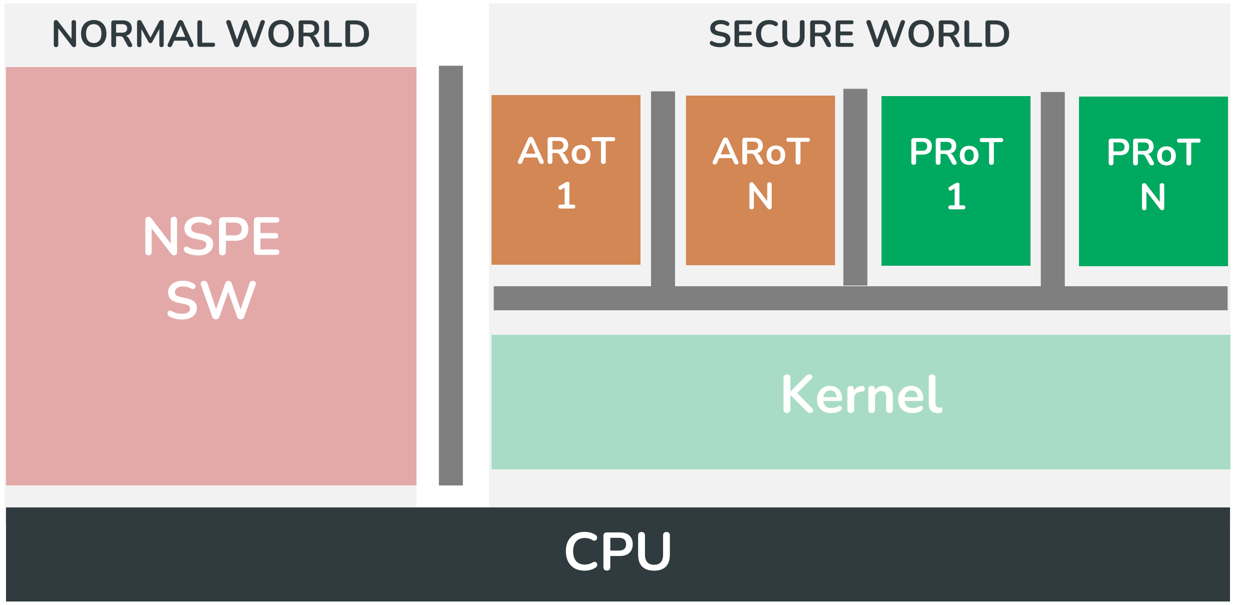


Microchip SAML11

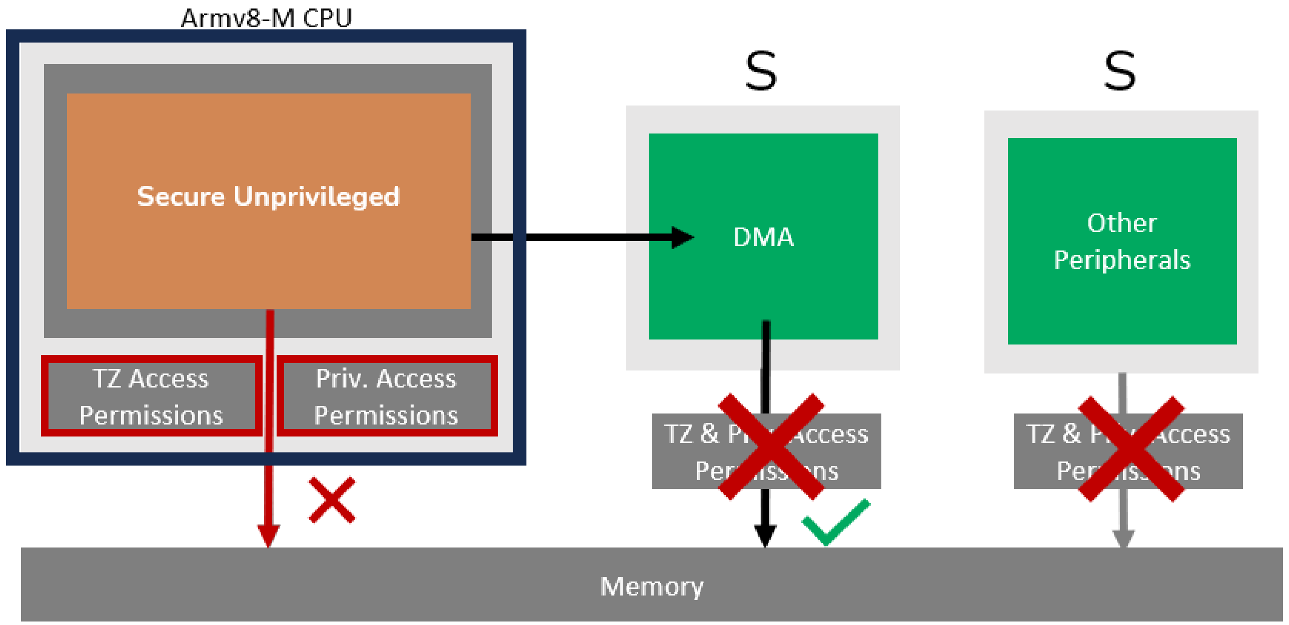
Only PSA Level 1 & No MPC

TRUSTONIC KINIBI-M

✓ SAU+IDAU



+



Kinibi-M Architecture

Seems Probably More then PSA Level 3

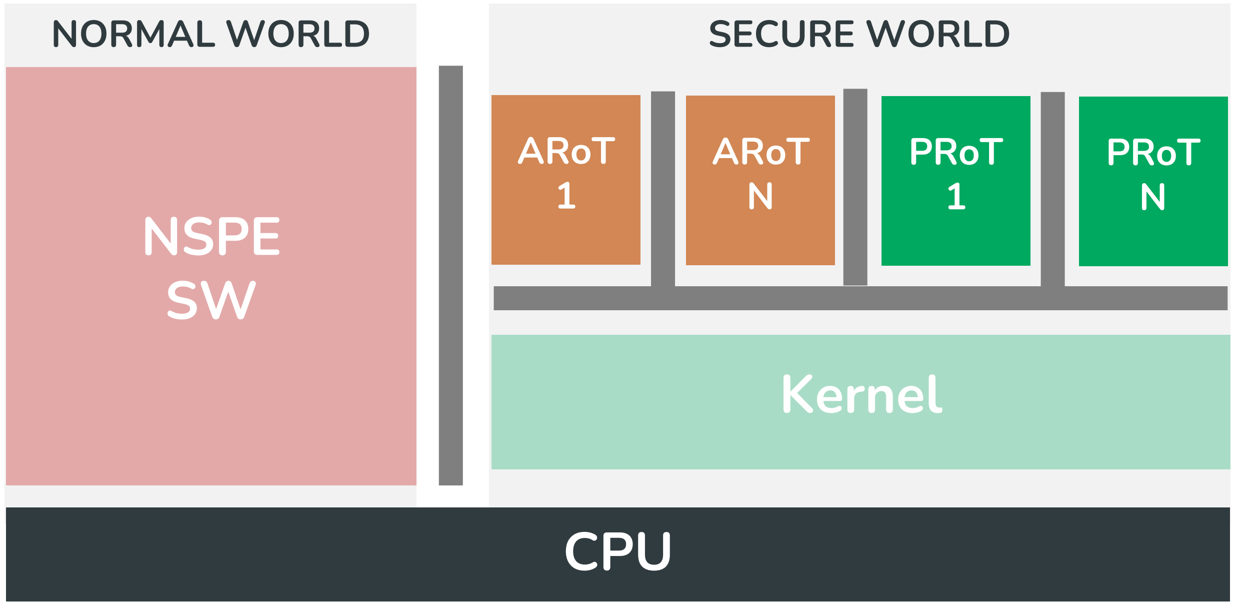
Microchip SAML11

Only PSA Level 1 & No MPC

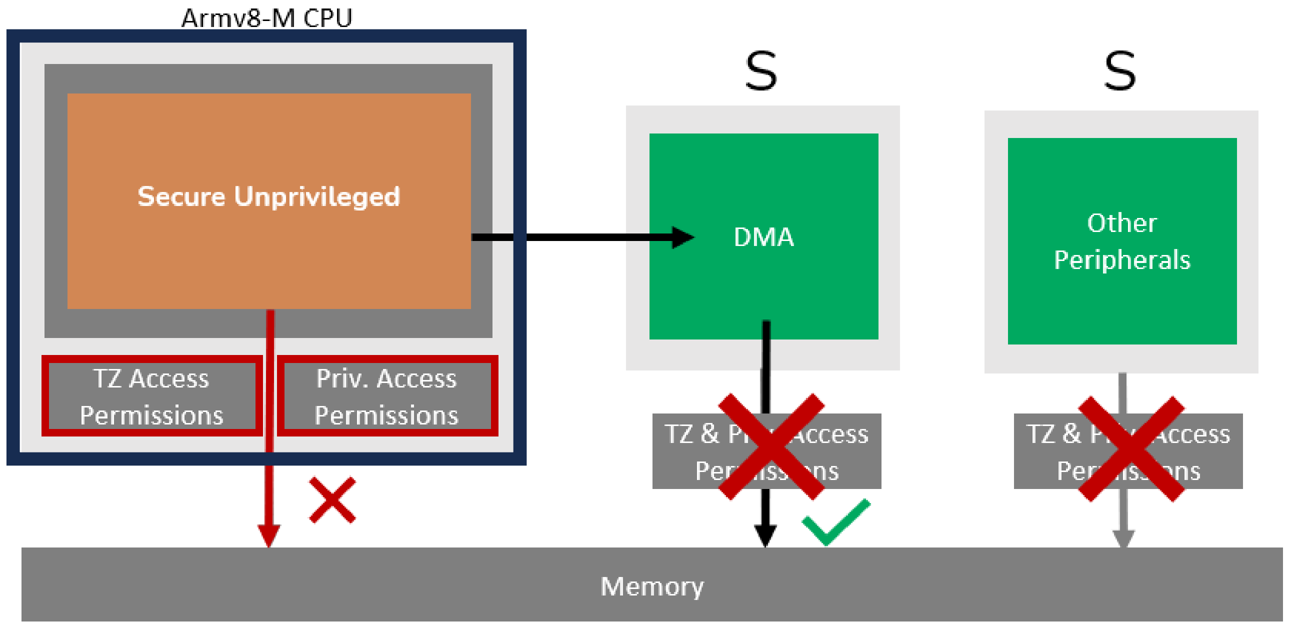
TRUSTONIC KINIBI-M

✓ SAU+IDAU

✓ MPU



+



Kinibi-M Architecture

Seems Probably More then PSA Level 3

Microchip SAML11

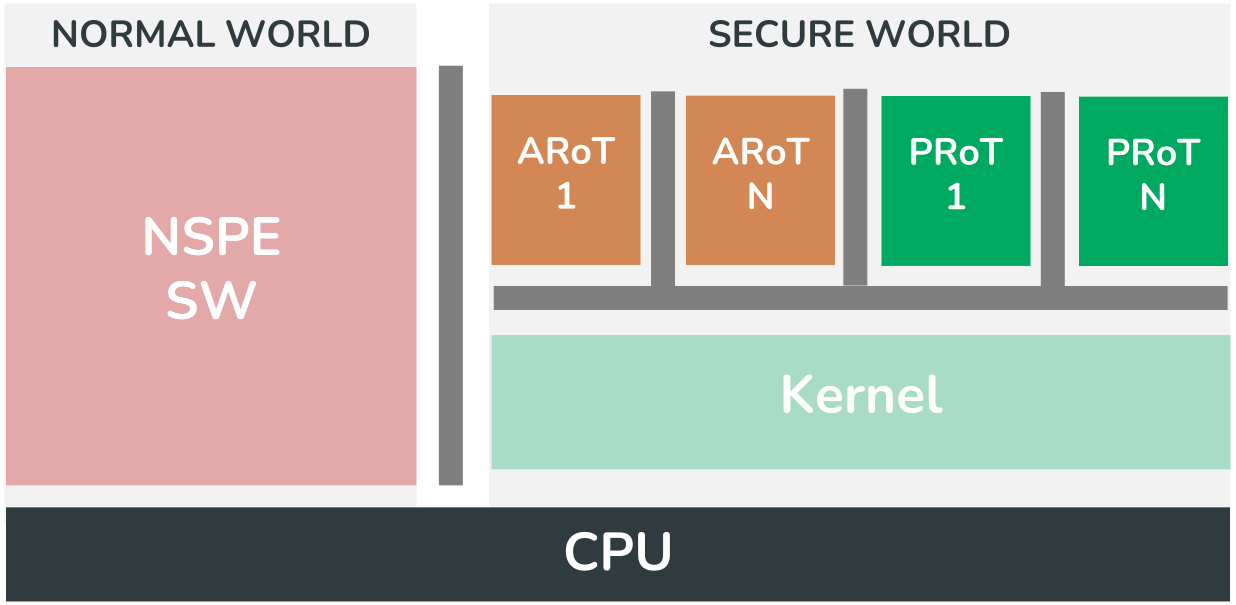
Only PSA Level 1 & No MPC

TRUSTONIC KINIBI-M

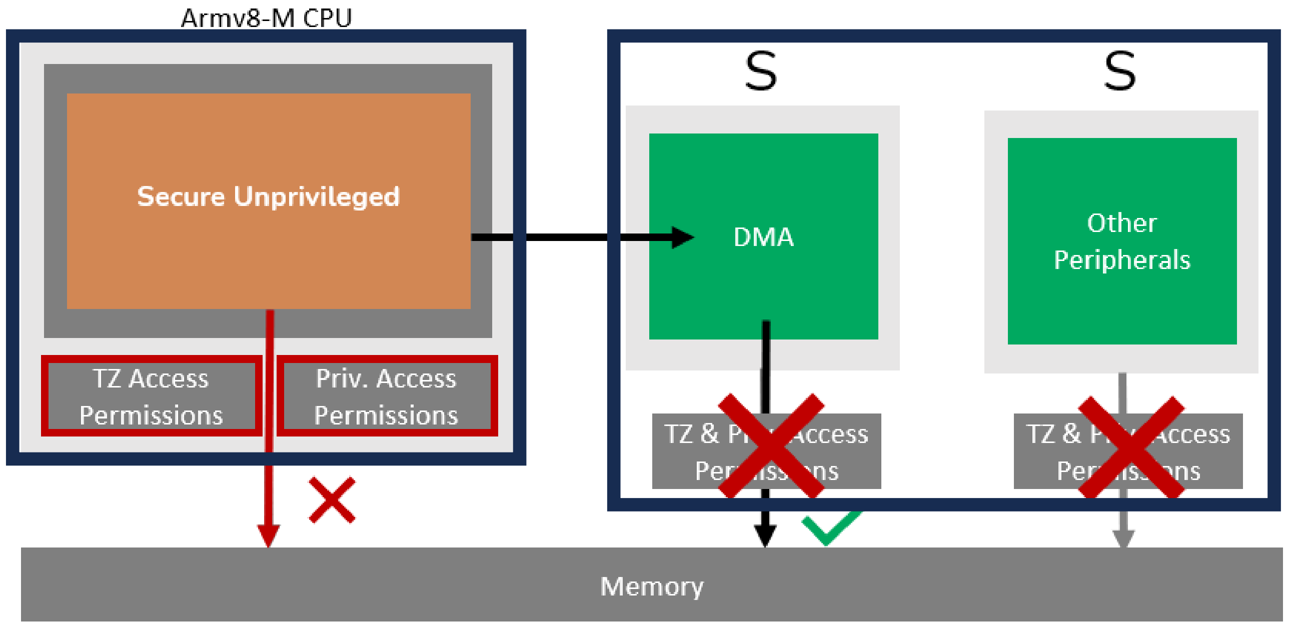
✓ SAU+IDAU

✓ MPU

✗ MPC



+



Kinibi-M Architecture

Seems Probably More then PSA Level 3

Microchip SAML11

Only PSA Level 1 & No MPC

With this gap of protection, a ***Secure Unprivileged application*** *that has been granted a DMA* ***can bypass*** *all* ***Kinibi-M*** *security mechanism and* ***achieve arbitrary read****,* ***write*** *or* ***execute capabilities***

Observation

Responsible Disclosure Trustonic

A Journey

We Contact Trustonic Reporting our Findings

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

Trustonic Security Team Acknowledged the Reception of Our Report

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

Trustonic Security Team Provided 1st Feedback

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

We Respond to 1st Feedback

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

Trustonic Security Team Provided 2nd Feedback

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

We Respond to 2nd Feedback

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

Trustonic Security Team Provided 3rd and last Feedback

Jan 10th

Jan 12th

Jan 30th

Jan 31st

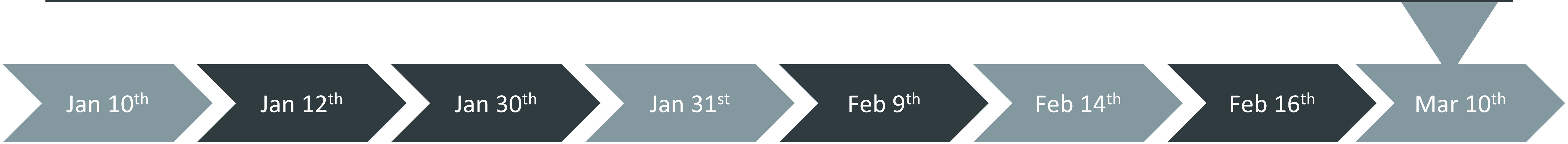
Feb 9th

Feb 14th

Feb 16th

Mar 10th

We Sent a Last Response Wrapping up the Responsible Disclosure



Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

1 **Topic:** Evaluatoin SDK vs Comercial SDK

Jan 10th
Jan 12th
Jan 30th
Jan 31st
Feb 9th
Feb 14th
Feb 16th
Mar 10th

1 **Topic:** Evaluatoin SDK vs Comercial SDK

2 **Topic:** Attestation Secure Modules

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

1 **Topic:** Evaluatoin SDK vs Comercial SDK

2 **Topic:** Attestation Secure Modules

3 **Topic:** DMA Permissions

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 **Topic:** Evaluatoin SDK vs Comercial SDK

2 **Topic:** Attestation Secure Modules

3 **Topic:** DMA Permissions

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 2 3

Topic: Evaluatoin SDK vs Comercial SDK

"We note that you are **using the Kinibi-M evaluation SDK**, **not** the full (**commercial**) **production SDK**. (…) Kinibi-M evaluation (…) is deliberately more flexible than a commercial (…) production SDK"

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

1 2 3

Topic: Evaluatoin SDK vs Comercial SDK

“We note that you are **using the Kinibi-M evaluation SDK**, **not** the full (**commercial**) **production SDK**. (…) Kinibi-M evaluation (…) is deliberately more flexible than a commercial (…) production SDK”

DISCLAIMER

We were only granted access to the **evaluation SDK**, thus all assessments and **conclusions presented on this talk** are derived form documentation and artifacts **from the Evaluation SDK**.

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

1 2 3

Topic: Evaluatoin SDK vs Comercial SDK

"We note that you are **using the Kinibi-M evaluation SDK**, **not** the full (**commercial**) **production SDK**. (...) Kinibi-M evaluation (...) is deliberately more flexible than a commercial (...) production SDK"

DISCLAIMER

We were only granted access to the **evaluation SDK**, thus all assessments and **conclusions presented on this talk** are derived form documentation and artifacts **from the Evaluation SDK**.

We **still think commercial version may suffer from the same problem** (the underlying architecture problem is the same, weak hardware protections on SAML11)

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 **Topic:** Evaluatoin SDK vs Comercial SDK

2 **Topic:** Attestation Secure Modules

3 **Topic:** DMA Permissions

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 **Topic:** Evaluatoin SDK vs Comercial SDK

2 **Topic:** Attestation Secure Modules

3 **Topic:** DMA Permissions

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

1 2 3

Topic: Attestation Secure Modules

You cannot install malicious modules because, “all **modules** must be **signed**, and are **validated** at install time against a protected list of signing keys” (attestation).

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

Topic: Attestation Secure Modules

You cannot install malicious modules because, “all **modules** must be **signed**, and are **validated** at install time against a protected list of signing keys” (attestation).

DISCLAIMER

The **Evaluation SDK doesn’t support attestation** of secure modules so we could freely instantiate secure modules, but in the **Commercial SDK only OEMs can instantiate modules** and they are all **signed** and **validated**.

1 2 3

Topic: Attestation Secure Modules

You cannot install malicious modules because, “all **modules** must be **signed**, and are **validated** at install time against a protected list of signing keys” (attestation).

DISCLAIMER

The **Evaluation SDK doesn’t support attestation** of secure modules so we could freely instantiate secure modules, but in the **Commercial SDK only OEMs can instantiate modules** and they are all **signed** and **validated**.

Attesting OEMs’ **Secure Modules** offers **no guarantees** that the Secure Module has **no defects**.

Jan 10th · Jan 12th · Jan 30th · Jan 31st · Feb 9th · Feb 14th · Feb 16th · Mar 10th

1 2 3

Topic: Attestation Secure Modules

You cannot install malicious modules because, “all **modules** must be **signed**, and are **validated** at install time against a protected list of signing keys” (attestation).

DISCLAIMER

The **Evaluation SDK doesn’t support attestation** of secure modules so we could freely instantiate secure modules, but in the **Commercial SDK only OEMs can instantiate modules** and they are all **signed** and **validated**.

Attesting OEMs’ **Secure Modules** offers **no guarantees** that the Secure Module has **no defects**.

Unless OEMs code is **formally verified** (which, as far as we know, is not the industry standard) **we should** (by probability) **expect bugs** and vulnerabilities.

Jan 10th · Jan 12th · Jan 30th · Jan 31st · Feb 9th · Feb 14th · Feb 16th · Mar 10th

TAKEAWAY

1

We argue that there is a **naive trust in OEM developers**. **Even if** there is **no malicious intent**, unintended **bugs may be introduced in the code** which may lead to a vulnerability, e.g., privileged escalation.

Jan 10th | Jan 12th | **Jan 30th** | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 **Topic:** Evaluatoin SDK vs Comercial SDK

2 **Topic:** Attestation Secure Modules

3 **Topic:** DMA Permissions

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 **Topic:** Evaluatoin SDK vs Comercial SDK

2 **Topic:** Attestation Secure Modules

3 **Topic:** DMA Permissions

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

Topic: DMA Permissions

It’s true that a **Secure Module with access to a DMA** "**can effectively access any part of the system**", it is "**a common limitation** of low-cost hardware, **however** it is **far from an open door**"

1 2 3

Topic: DMA Permissions

It’s true that a **Secure Module with access to a DMA** "**can effectively access any part of the system**", it is "**a common limitation** of low-cost hardware, **however** it is **far from an open door**"

“**Access** to the **DMA** controller **needs to be granted**, and the best practice guidance in the **production SDK** (which we acknowledge you do not have) **explains how to lock down** access to **devices** from less trusted developers”

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 2 3

Topic: DMA Permissions

It’s true that a **Secure Module with access to a DMA** "**can effectively access any part of the system**", it is "**a common limitation** of low-cost hardware, **however** it is **far from an open door**"

“**Access** to the **DMA** controller **needs to be granted**, and the best practice guidance in the **production SDK** (which we acknowledge you do not have) **explains how to lock down** access to **devices** from less trusted developers”

Contradictory ideas, on one side, Trustonic admits that a **Secure Module with DMA** access **has full access to the system**, and, on the other side, Trustonic claims that it **is not an open door**.

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 2 3

Topic: DMA Permissions

It's true that a **Secure Module with access to a DMA** "**can effectively access any part of the system**", it is "**a common limitation** of low-cost hardware, **however** it is **far from an open door**"

"**Access** to the **DMA** controller **needs to be granted**, and the best practice guidance in the **production SDK** (which we acknowledge you do not have) **explains how to lock down** access to **devices** from less trusted developers"

Contradictory ideas, on one side, Trustonic admits that a **Secure Module with DMA** access **has full access to the system**, and, on the other side, Trustonic claims that it **is not an open door**.

DMA access **should not** need to **be granted but MEDIATED** (because lack of hardware mechanisms). **Kinibi-B should mediate** access from **ALL Secure Modules** via DMA interposer.

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 2 3

Topic: DMA Permissions

It’s true that a **Secure Module with access to a DMA** "**can effectively access any part of the system**", it is "**a common limitation** of low-cost hardware, **however** it is **far from an open door**"

“**Access** to the **DMA** controller **needs to be granted**, and the best practice guidance in the **production SDK** (which we acknowledge you do not have) **explains how to lock down** access to **devices** from less trusted developers”

Contradictory ideas, on one side, Trustonic admits that a **Secure Module with DMA** access **has full access to the system**, and, on the other side, Trustonic claims that it **is not an open door**.

DMA access **should not** need to **be granted but MEDIATED** (because lack of hardware mechanisms). **Kinibi-B should mediate** access from **ALL Secure Modules** via DMA interposer.

We proposed to share the **DMA interposer mechanism** to fix the DMA issue.

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

TAKEAWAY

1

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

TAKEAWAY

1

We argue that there is a **lack of understanding of the limitations** of the **underlying hardware** (where Kinibi-M runs) and the necessary **Software mechanisms needed** to **enforce claimed protections**.

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

1 **Topic:** No Native DMA Support

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

1 **Topic:** No Native DMA Support

2 **Topic:** No System MMU & DMA permissions

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

1. **Topic:** No Native DMA Support

2. **Topic:** No System MMU & DMA permissions

3. **Topic:** Native FLASH Access Mediation but not Native DMA mediation.

1. **Topic:** No Native DMA Support

2. **Topic:** No System MMU & DMA permissions

3. **Topic:** Native FLASH Access Mediation but not Native DMA mediation.

Jan 10th · Jan 12th · Jan 30th · Jan 31st · Feb 9th · Feb 14th · Feb 16th · Mar 10th

1 2 3

Topic: No Native DMA Support

“**Kinibi-M** for SAML11 **does not ship with a Secure World DMA module**, and it is **left up to customers** to source one or do without.”

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

1 2 3

Topic: No Native DMA Support

"**Kinibi-M** for SAML11 **does not ship with a Secure World DMA module**, and it is **left up to customers** to source one or do without."

"In our architecture it **would be up to the OEM** provided **DMA module to provide that mediation**"

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

1 2 3

Topic: No Native DMA Support

"**Kinibi-M** for SAML11 **does not ship with a Secure World DMA module**, and it is **left up to customers** to source one or do without."

"In our architecture it **would be up to the OEM** provided **DMA module to provide that mediation**"

OEMs have to source one DMA module if they want to use a DMA. **We don't think is a good approach**, because this **forces OEMs to trust each other** (which they don't).

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 2 3

Topic: No Native DMA Support

“**Kinibi-M** for SAML11 **does not ship with a Secure World DMA module**, and it is **left up to customers** to source one or do without.”

“In our architecture it **would be up to the OEM** provided **DMA module to provide that mediation**”

OEMs have to source one DMA module if they want to use a DMA. **We don’t think is a good approach**, because this **forces OEMs to trust each other** (which they don’t).

It also **increases** the **probability** of a **bug/vulnerability**.

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

TAKEAWAY

1

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

TAKEAWAY

1 We argue that there is a **lack of understanding of multi-OEM threat model**. In a multistakeholder scenario (i.e., multiple OEMs) **OEMs don’t trust each other**.

1. **Topic:** No Native DMA Support

2. **Topic:** No System MMU & DMA permissions

3. **Topic:** Native FLASH Access Mediation but not Native DMA mediation.

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

1 **Topic:** No Native DMA Support

2 **Topic:** No System MMU & DMA permissions

3 **Topic:** Native FLASH Access Mediation but not Native DMA mediation.

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

1 2 3

Topic: No System MMU & DMA permissions

“**You have at most revealed** that this **device has no system MMU** (covered in the data sheet), and that **DMA permissions should not be granted** to untrusted application modules”

Jan 10th | Jan 12th | Jan 30th | Jan 31st | **Feb 9th** | Feb 14th | Feb 16th | Mar 10th

1 2 3

Topic: No System MMU & DMA permissions

“**You have at most revealed** that this **device has no system MMU** (covered in the data sheet), and that **DMA permissions should not be granted** to untrusted application modules”

System MMU is an access control IP used in **platforms with virtual memory**, In **Cortex-M (MCU)** platforms, there are no SMMU, but **MPC** (Memory Protection Controller) and **PPC** (Peripheral Protection Controller)

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 2 3

Topic: No System MMU & DMA permissions

“**You have at most revealed** that this **device has no system MMU** (covered in the data sheet), and that **DMA permissions should not be granted** to untrusted application modules”

System MMU is an access control IP used in **platforms with virtual memory**, In **Cortex-M (MCU)** platforms, there are no SMMU, but **MPC** (Memory Protection Controller) and **PPC** (Peripheral Protection Controller)

The **PPC/MPC** in **SAML11 cannot enforce** access control in terms of **privilege levels**. **If you** directly **assign a DMA** device **to an OEM** you are basically **granting them full control of the system**

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

1 2 3

Topic: No System MMU & DMA permissions

"**You have at most revealed** that this **device has no system MMU** (covered in the data sheet), and that **DMA permissions should not be granted** to untrusted application modules"

System MMU is an access control IP used in **platforms with virtual memory**, In **Cortex-M (MCU**) platforms, there are no SMMU, but **MPC** (Memory Protection Controller) and **PPC** (Peripheral Protection Controller)

The **PPC/MPC** in **SAML11 cannot enforce** access control in terms of **privilege levels**. **If you** directly **assign a DMA** device **to an OEM** you are basically **granting them full control of the system**

Kinibi-M should provide native DMA support once it is a critical piece of infrastructure for Microcontrollers, due to the power and resource-constrained nature of this devices.

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

TAKEAWAY

1

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

TAKEAWAY

1

We argue there is a **lack of understanding** about **the memory protection controllers** of **Microcontrollers** (system wide protection mechanisms).

1 **Topic:** No Native DMA Support

2 **Topic:** No System MMU & DMA permissions

3 **Topic:** Native FLASH Access Mediation but not Native DMA mediation.

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

1. **Topic:** No Native DMA Support

2. **Topic:** No System MMU & DMA permissions

3. **Topic:** Native FLASH Access Mediation but not Native DMA mediation.

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

1 2 3 **Topic:** Native FLASH Access Mediation but not Native DMA mediation.

“Kinibi-M fully supports secure identification of module-to-module caller identity precisely to support this sort of use case. For example this is the pattern we use to **mediated access to flash storage provided by our secure storage module.”**

Jan 10th | Jan 12th | Jan 30th | Jan 31st | **Feb 9th** | Feb 14th | Feb 16th | Mar 10th

1 2 3 **Topic:** Native FLASH Access Mediation but not Native DMA mediation.

“Kinibi-M fully supports secure identification of module-to-module caller identity precisely to support this sort of use case. For example this is the pattern we use to **mediated access to flash storage provided by our secure storage module.”**

Kinibi-M provides mediation for **flash** storage, but **why doesn't** it offer similar **mediation for DMA**? DMA is also a critical service, arguably even more.

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

TAKEAWAY

1

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

TAKEAWAY

1 We argue that there is a **lack of understanding** regarding the **criticality of a core service such as the DMA**. If mismanaged, it can grant full access to all system memory.

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

1 **Topic:** Clarification of Kinibi-M isolation levels

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

1 **Topic:** Clarification of Kinibi-M isolation levels

2 **Topic:** Clarification of who should provide DMA mediator

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

1 **Topic:** Clarification of Kinibi-M isolation levels

2 **Topic:** Clarification of who should provide DMA mediator

3 **Topic:** Requests to Trustonic

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

1 **Topic:** Clarification of Kinibi-M isolation levels

2 **Topic:** Clarification of who should provide DMA mediator

3 **Topic:** Requests to Trustonic

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

1 2 3

Topic: Clarification of Kinibi-M isolation levels

“**Kinibi-M pre-dates Arm PSA** and was not built on the PSA architecture. (...) **In some areas we do more that PSA (any level) in others we do less**. That is why we do not claim PSA Level 3 and have not certified against it.”

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

TAKEAWAY

1

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

TAKEAWAY

1

We argue there is **lack of awareness and mapping** regarding the **PSA isolation levels** on Kinibi-M.

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → **Feb 16th** → Mar 10th

1. **Topic:** Clarification of Kinibi-M isolation levels
2. **Topic:** Clarification of who should provide DMA mediator
3. **Topic:** Requests to Trustonic

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 **Topic:** Clarification of Kinibi-M isolation levels

2 **Topic:** Clarification of who should provide DMA mediator

3 **Topic:** Requests to Trustonic

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

1 2 3

Topic: Clarification of who should provide DMA mediator

"This **device has only** (at most) **64kb of flash** and a **16kb of ram**. There are very few use cases for secure world DMA. In practice **most customers simply disable the use of DMA in the secure world**, preventing any potential abuse."

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → **Feb 16th** → Mar 10th

1 2 3 **Topic:** Clarification of who should provide DMA mediator

"This **device has only** (at most) **64kb of flash** and a **16kb of ram**. There are very few use cases for secure world DMA. In practice **most customers simply disable the use of DMA in the secure world**, preventing any potential abuse."

"If needed, **DMA access should be** provided and **mediated by a "system" module**. That is what we have said all along. **However**, that module needs to be **provided by an OEM**. **It is not provided by Trustonic**."

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

1 2 3

Topic: Clarification of who should provide DMA mediator

"This **device has only** (at most) **64kb of flash** and a **16kb of ram**. There are very few use cases for secure world DMA. In practice **most customers simply disable the use of DMA in the secure world**, preventing any potential abuse."

"If needed, **DMA access should be** provided and **mediated by a "system" module**. That is what we have said all along. **However**, that module needs to be **provided by an OEM**. **It is not provided by Trustonic**."

We strongly believe that **not providing DMA mediation** is **not** a **good security practice**.

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 2 3 **Topic:** Clarification of who should provide DMA mediator

“This **device has only** (at most) **64kb of flash** and a **16kb of ram**. There are very few use cases for secure world DMA. In practice **most customers simply disable the use of DMA in the secure world**, preventing any potential abuse.”

“If needed, **DMA access should be** provided and **mediated by a “system” module**. That is what we have said all along. **However**, that module needs to be **provided by an OEM**. **It is not provided by Trustonic**.”

We strongly believe that **not providing DMA mediation** is **not** a **good security practice**.

DMAs are **key components** in **MCUs** (but bus masters!!). **Not providing** DMA module **is limiting** the **system’s capabilities** from one side and **leaving an open threat vector** on the other side.

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

1 **Topic:** Clarification of Kinibi-M isolation levels

2 **Topic:** Clarification of who should provide DMA mediator

3 **Topic:** Requests to Trustonic

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

1 **Topic:** Clarification of Kinibi-M isolation levels

2 **Topic:** Clarification of who should provide DMA mediator

3 **Topic:** Requests to Trustonic

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

1 2 3

Requests to Trustonic

To **issue** a **Security Advisory**.

Jan 10th Jan 12th Jan 30th Jan 31st Feb 9th Feb 14th Feb 16th Mar 10th

1 2 3

Requests to Trustonic

To **issue** a **Security Advisory**.

Clarify the **documentation** clearly communicating the limitations of **Evaluation** SDK **vs Commercial** SDK.

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

1 2 3

Requests to Trustonic

To **issue** a **Security Advisory**.

Clarify the **documentation** clearly communicating the limitations of **Evaluation** SDK **vs Commercial** SDK.

Provide us **access** to the **Commercial SDK** for internal assessment.

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

1 2 3

Requests to Trustonic

To **issue** a **Security Advisory**.

Clarify the **documentation** clearly communicating the limitations of **Evaluation** SDK **vs Commercial** SDK.

Provide us **access** to the **Commercial SDK** for internal assessment.

NO RESPONSE TO OUR REQUESTS!

Jan 10th | Jan 12th | Jan 30th | Jan 31st | Feb 9th | Feb 14th | Feb 16th | Mar 10th

SUMMING UP

1

2

3

4

5

Jan 10th

Jan 12th

Jan 30th

Jan 31st

Feb 9th

Feb 14th

Feb 16th

Mar 10th

SUMMING UP

1 We **could only validate** our claims on **Evaluation SDK** (the only SDK we were granted permissions);

2

3

4

5

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

SUMMING UP

1. We **could only validate** our claims on **Evaluation SDK** (the only SDK we were granted permissions);

2. **Secure Modules** (from OEMs) are **signed and validated** on the **Commercial** Version;

3.

4.

5.

Jan 10th — Jan 12th — Jan 30th — Jan 31st — Feb 9th — Feb 14th — Feb 16th — Mar 10th

SUMMING UP

1. We **could only validate** our claims on **Evaluation SDK** (the only SDK we were granted permissions);

2. **Secure Modules** (from OEMs) are **signed and validated** on the **Commercial** Version;

3. We think **attestation is orthogonal** to the problem we discussed in this presentation;

4

5

Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

SUMMING UP

1. We **could only validate** our claims on **Evaluation SDK** (the only SDK we were granted permissions);

2. **Secure Modules** (from OEMs) are **signed and validated** on the **Commercial** Version;

3. We think **attestation is orthogonal** to the problem we discussed in this presentation;

4. Official **Kinibi-m claims** only **PSA Level 2** ready, **but** its **secure architecture claims higher protections levels** (not backed by any hardware or software mechanism);

5.

SUMMING UP

1. We **could only validate** our claims on **Evaluation SDK** (the only SDK we were granted permissions);

2. **Secure Modules** (from OEMs) are **signed and validated** on the **Commercial** Version;

3. We think **attestation is orthogonal** to the problem we discussed in this presentation;

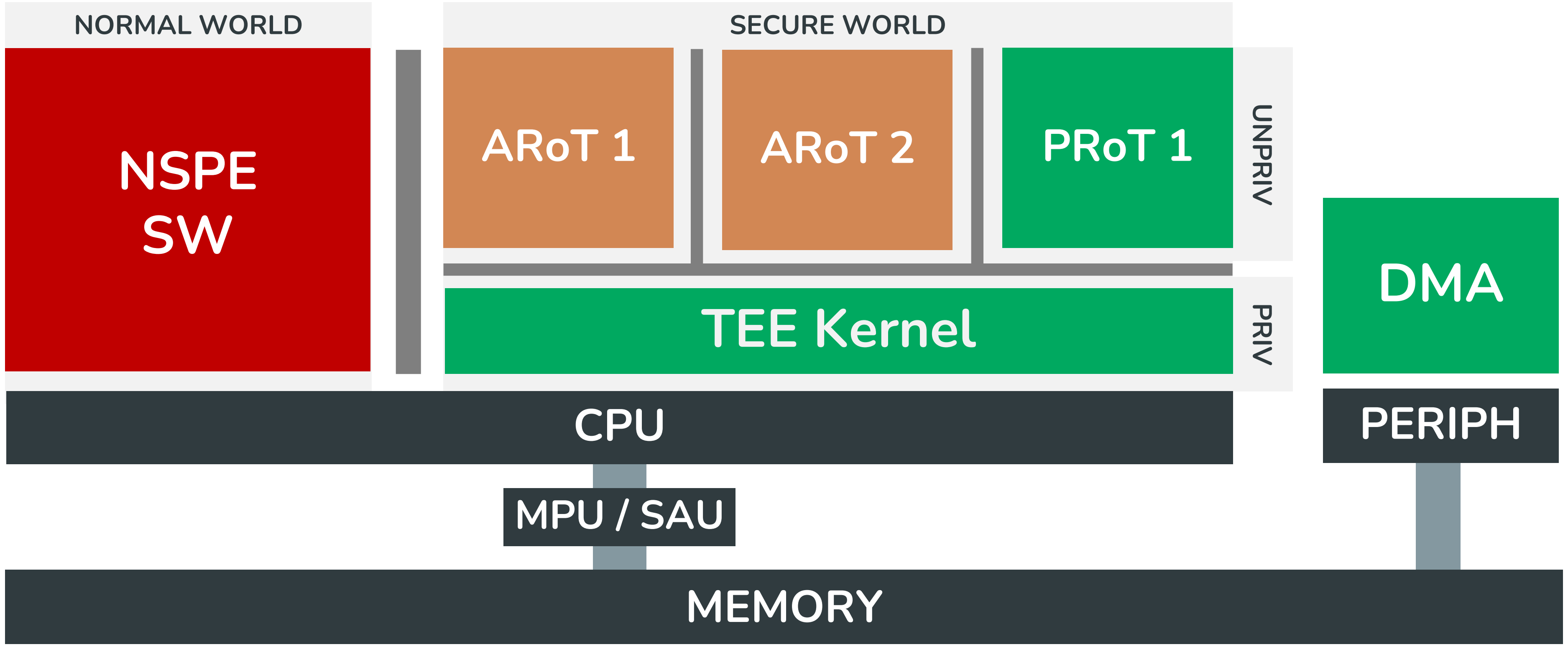
4. Official **Kinibi-m claims** only **PSA Level 2** ready, **but** its **secure architecture claims higher protections levels** (not backed by any hardware or software mechanism);

5. There is **no DMA mediator**, the responsibility is **left to the OEMs**, and by default Kinibi-M has no control of such an import core service, able to disrupt all system;

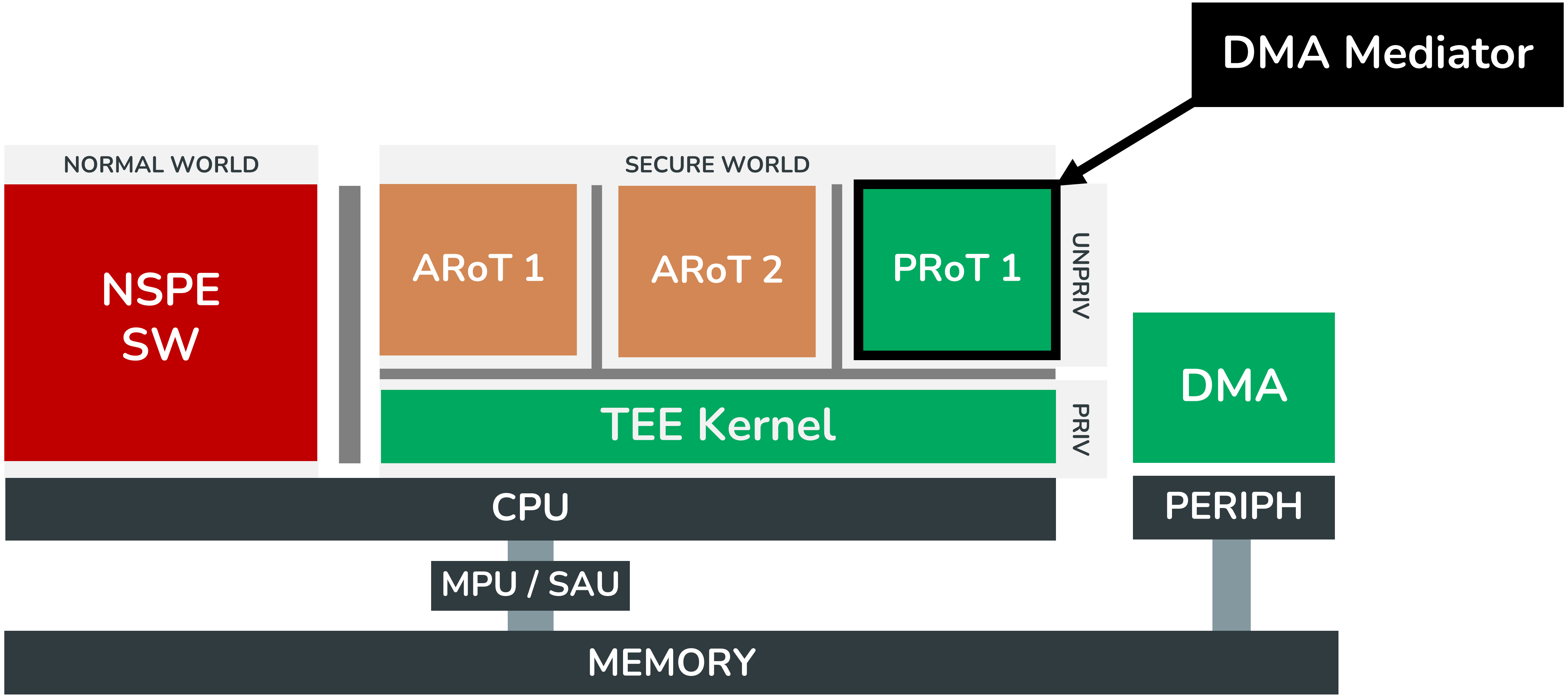
Jan 10th → Jan 12th → Jan 30th → Jan 31st → Feb 9th → Feb 14th → Feb 16th → Mar 10th

DMA Mediation

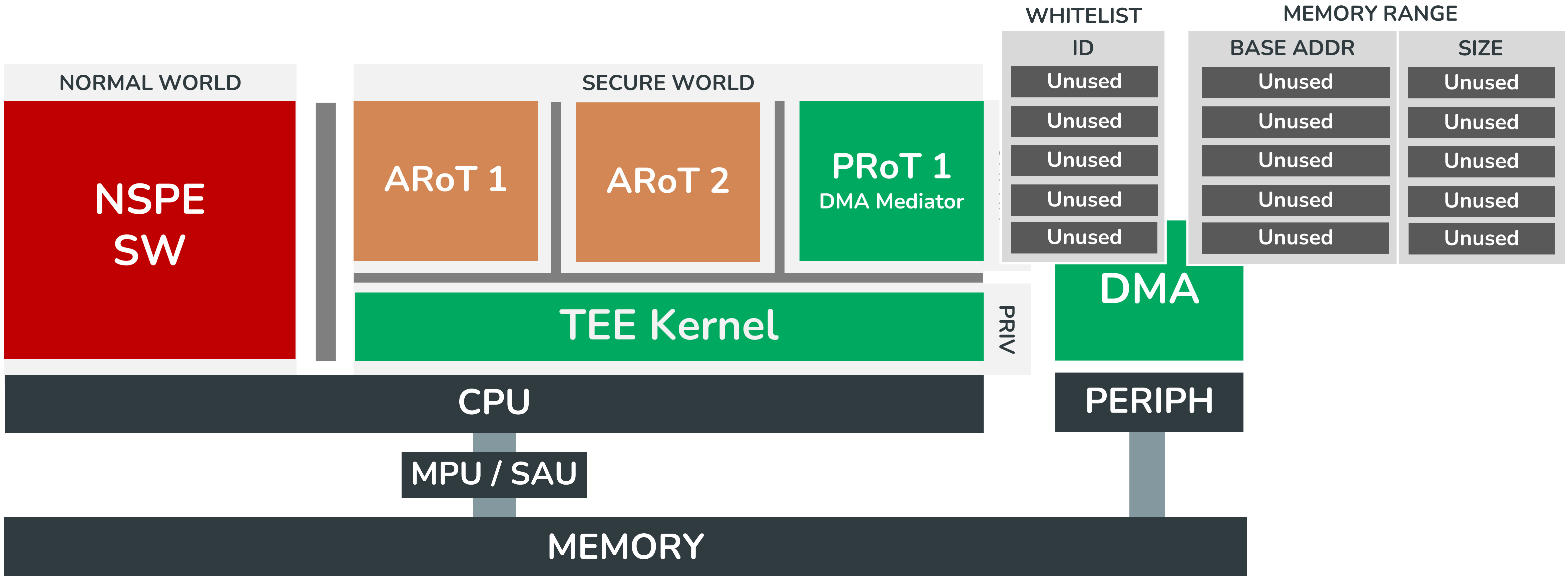
DMA MEDIATION



DMA MEDIATION

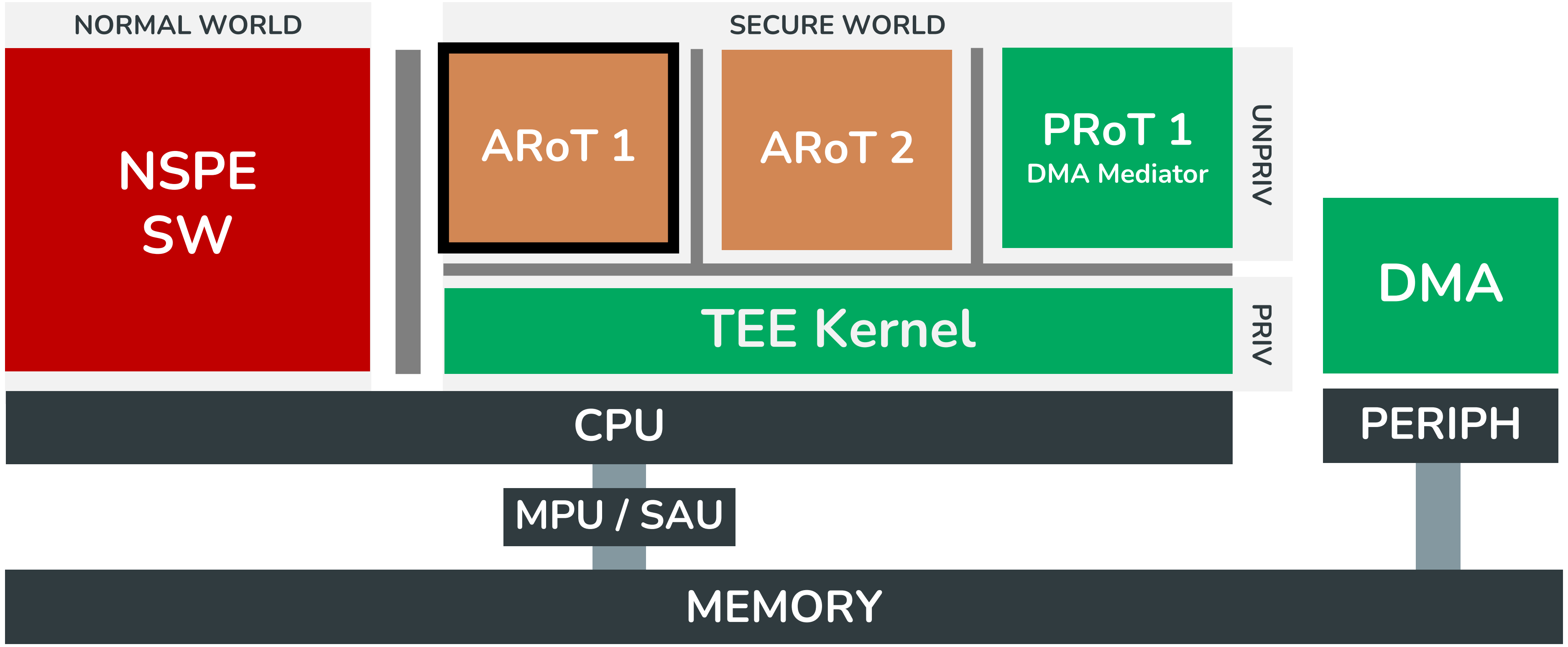


DMA MEDIATION



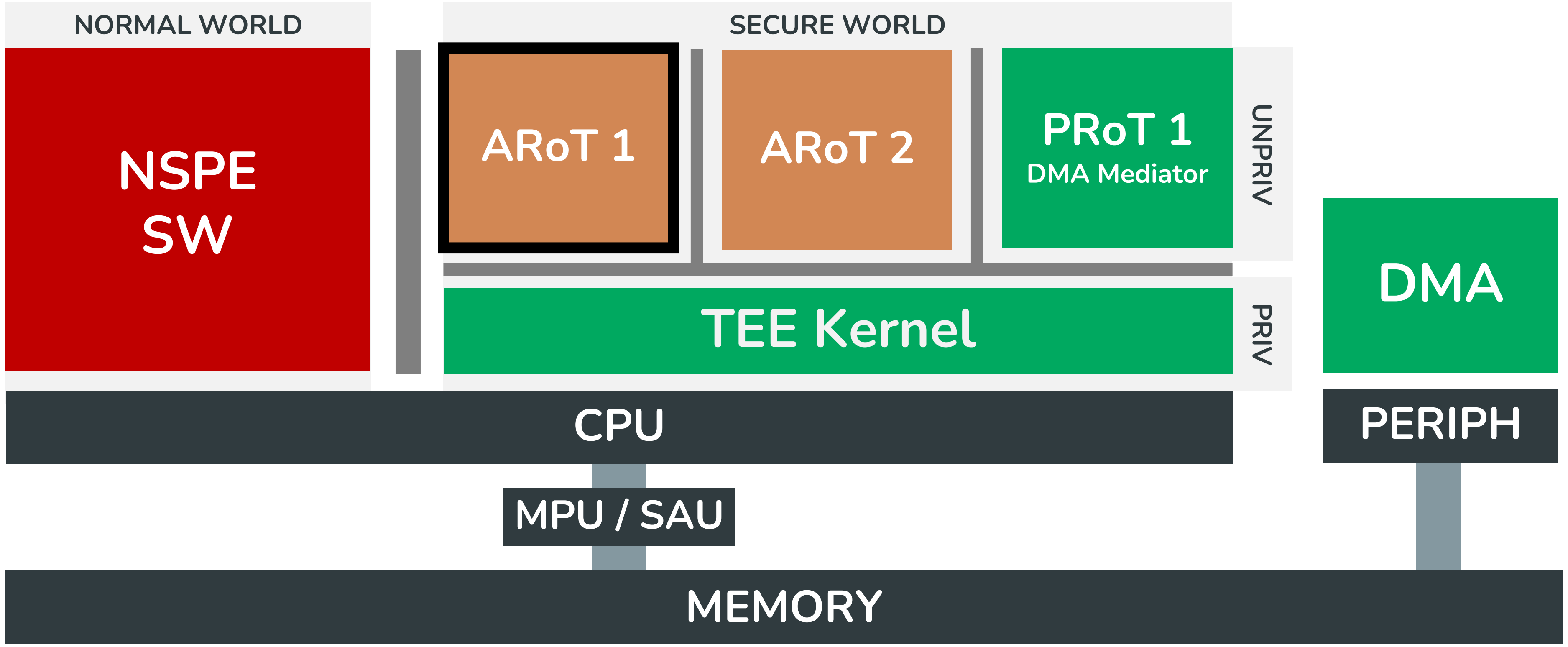
WHITELIST	MEMORY RANGE	
ID	BASE ADDR	SIZE
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused

DMA MEDIATION



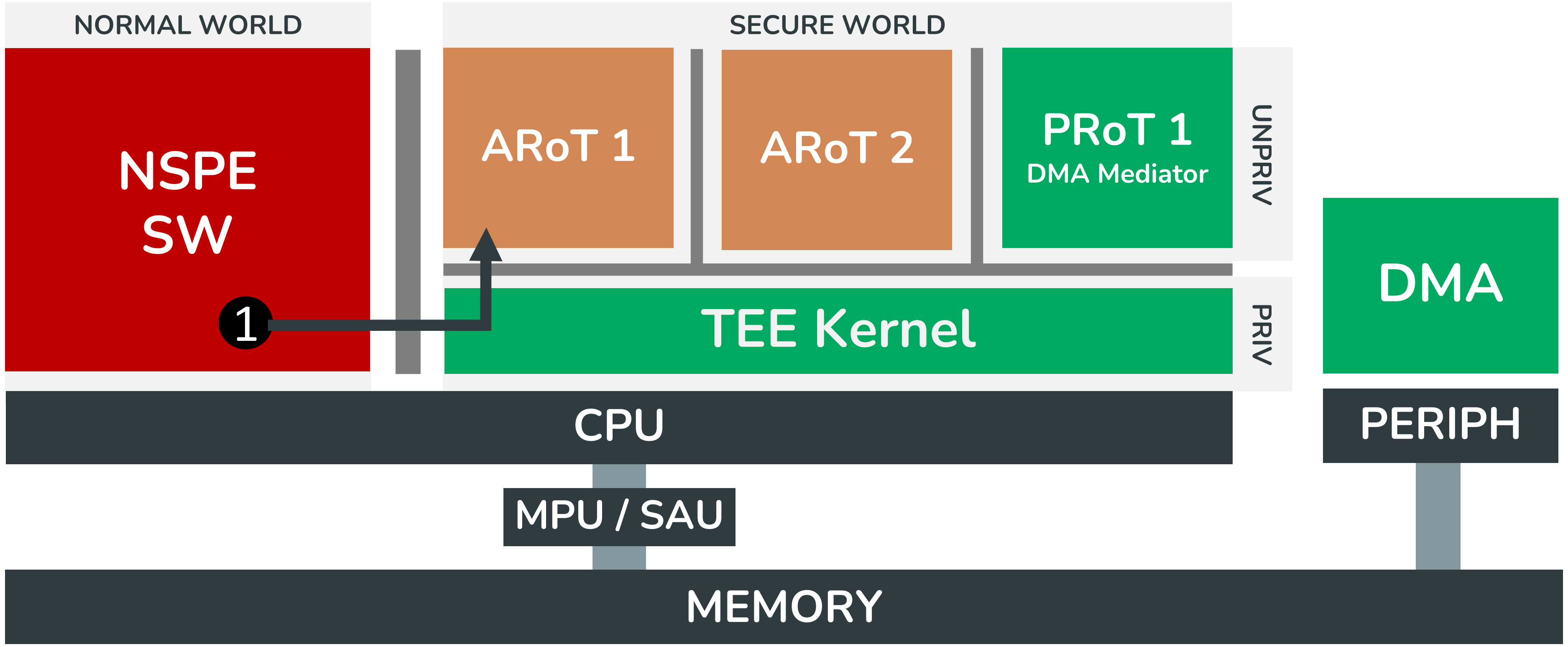
WHITELIST	MEMORY RANGE	
ID	BASE ADDR	SIZE
ARoT 1	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused

DMA MEDIATION



WHITELIST	MEMORY RANGE	
ID	BASE ADDR	SIZE
ARoT 1	0x20000000	0x1000
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused

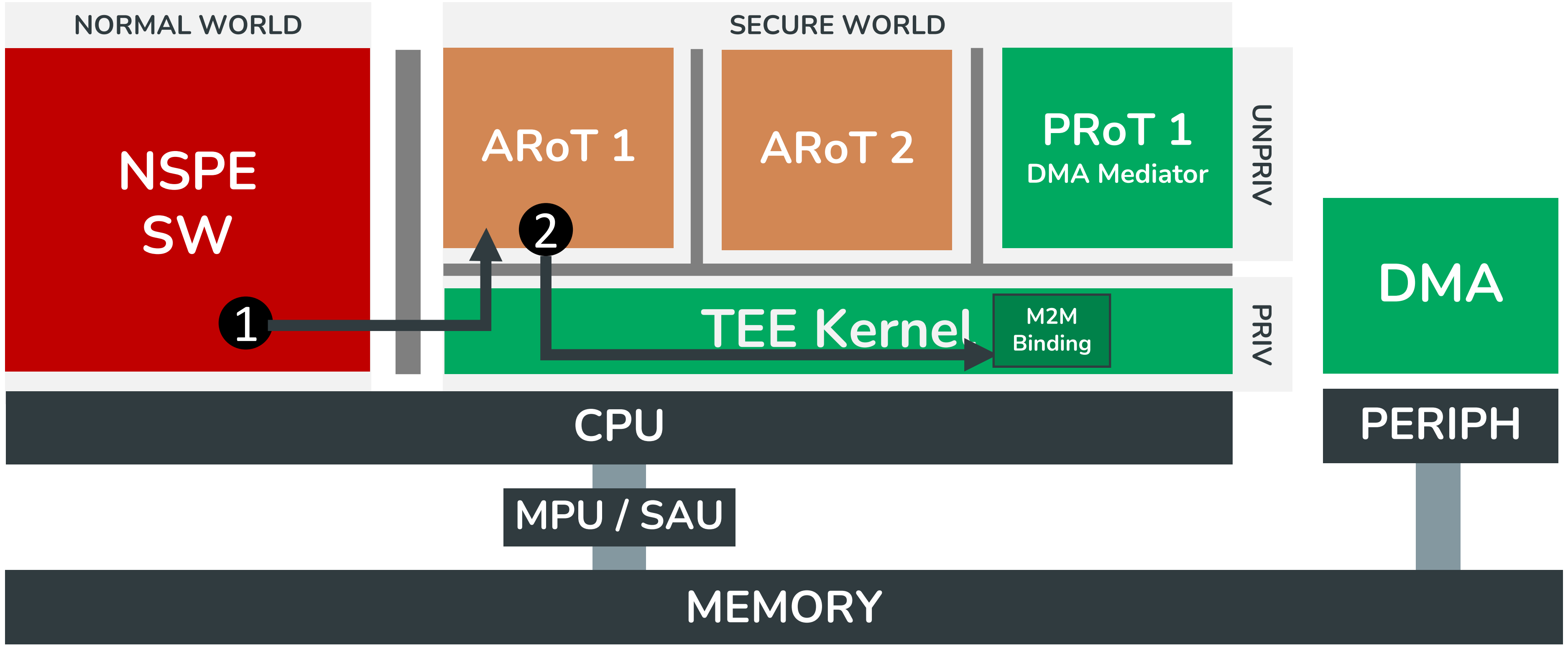
DMA MEDIATION



WHITELIST	MEMORY RANGE	
ID	BASE ADDR	SIZE
ARoT 1	0x20000000	0x1000
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused

❶ NS calls ARoT 1

DMA MEDIATION

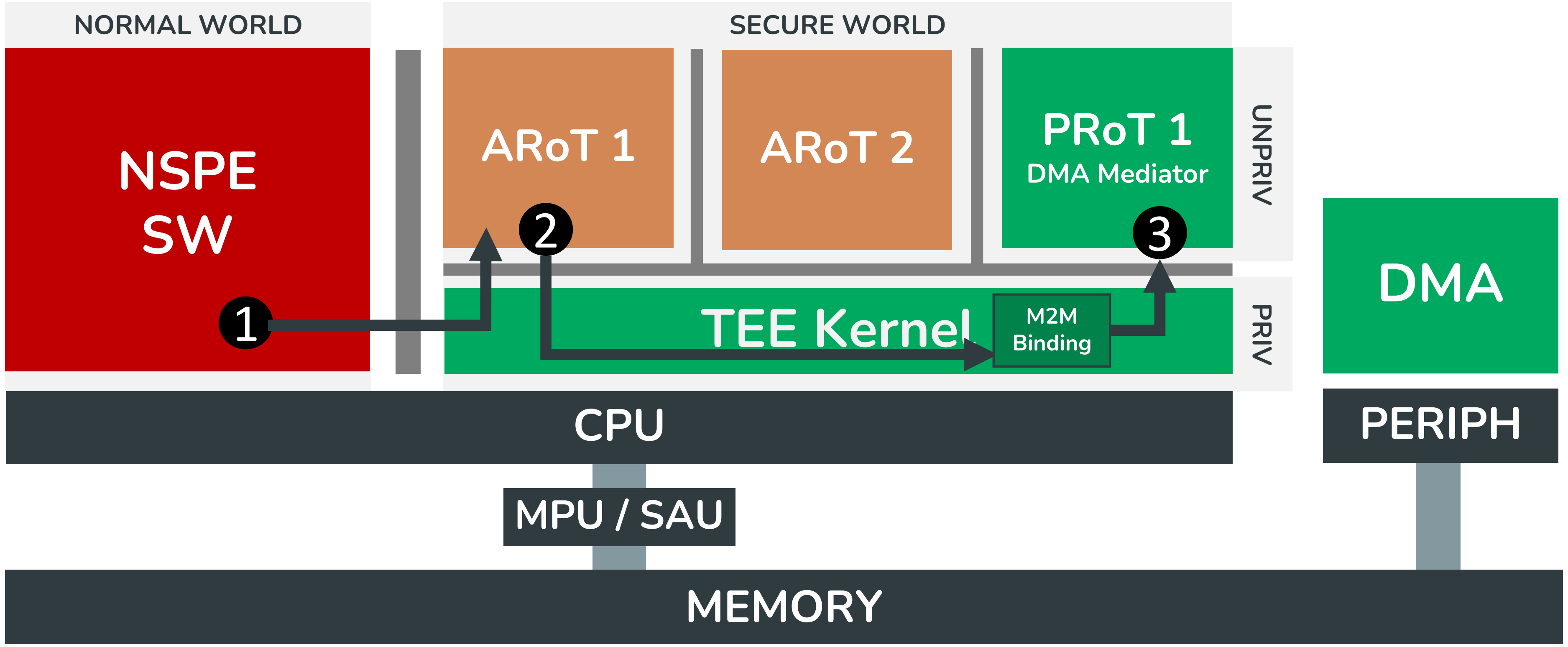


WHITELIST	MEMORY RANGE	
ID	BASE ADDR	SIZE
ARoT 1	0x20000000	0x1000
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused

❶ NS calls ARoT 1

❷ ARoT 1 requests access to DMA mediator

DMA MEDIATION



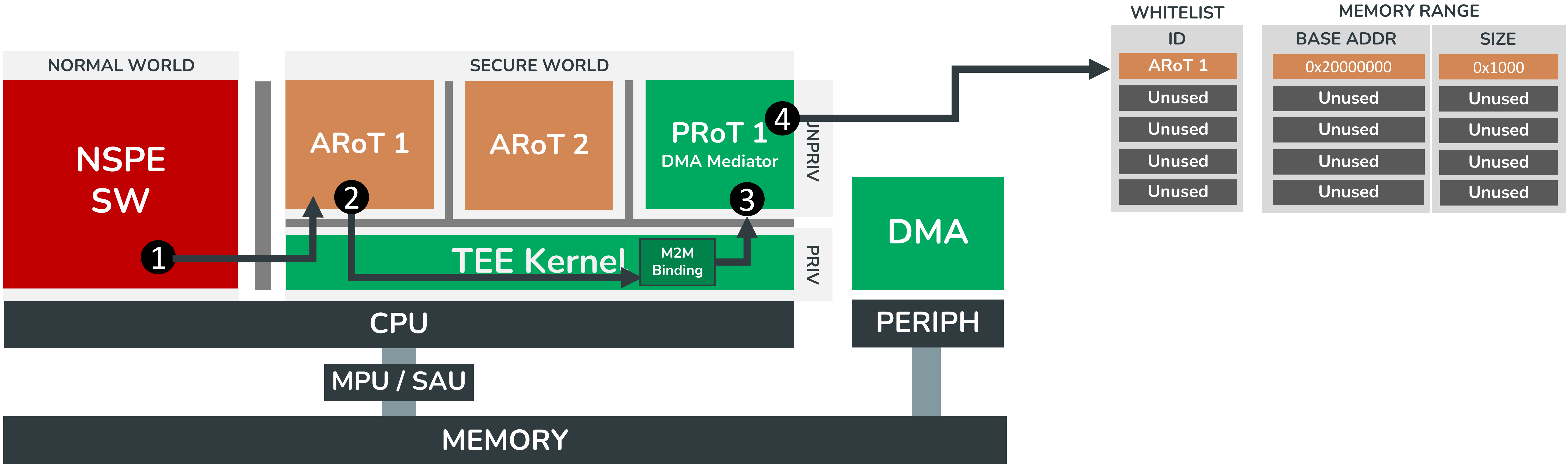
WHITELIST	MEMORY RANGE	
ID	BASE ADDR	SIZE
ARoT 1	0x20000000	0x1000
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused

❶ NS calls ARoT 1

❷ ARoT 1 requests access to DMA mediator

❸ TEE Kernel Invokes DMA Mediator

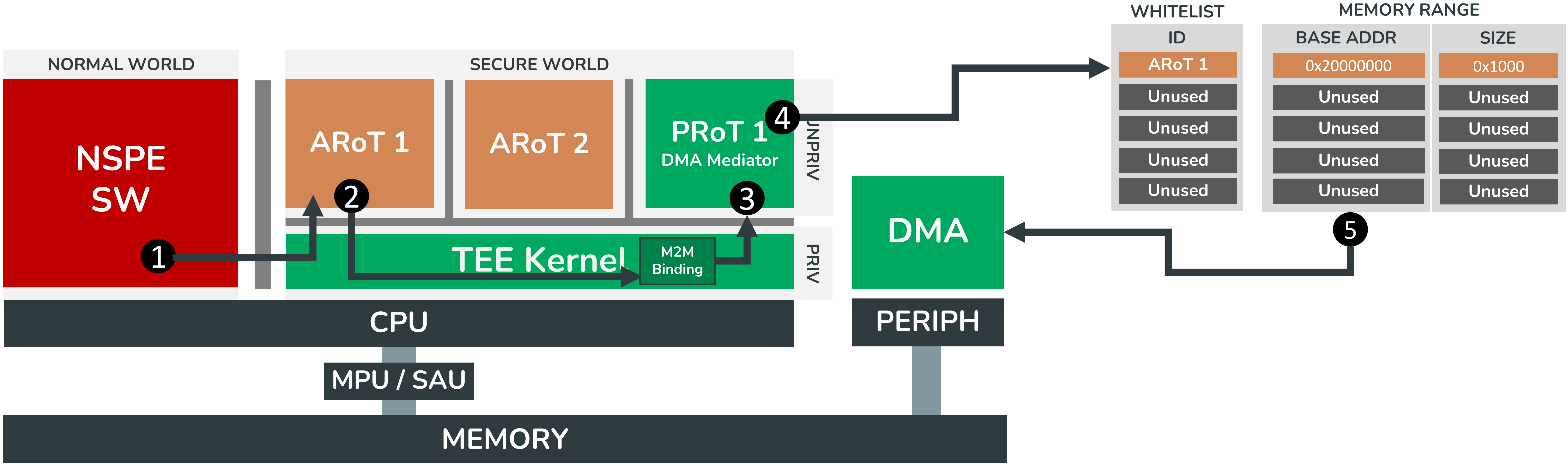
DMA MEDIATION



WHITELIST	MEMORY RANGE	
ID	BASE ADDR	SIZE
ARoT 1	0x20000000	0x1000
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused

1. NS calls ARoT 1
2. ARoT 1 requests access to DMA mediator
3. TEE Kernel Invokes DMA Mediator
4. DMA Mediator Checks Access Permissions and Memory Range

DMA MEDIATION



WHITELIST	MEMORY RANGE	
ID	BASE ADDR	SIZE
ARoT 1	0x20000000	0x1000
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused

1 NS calls ARoT 1

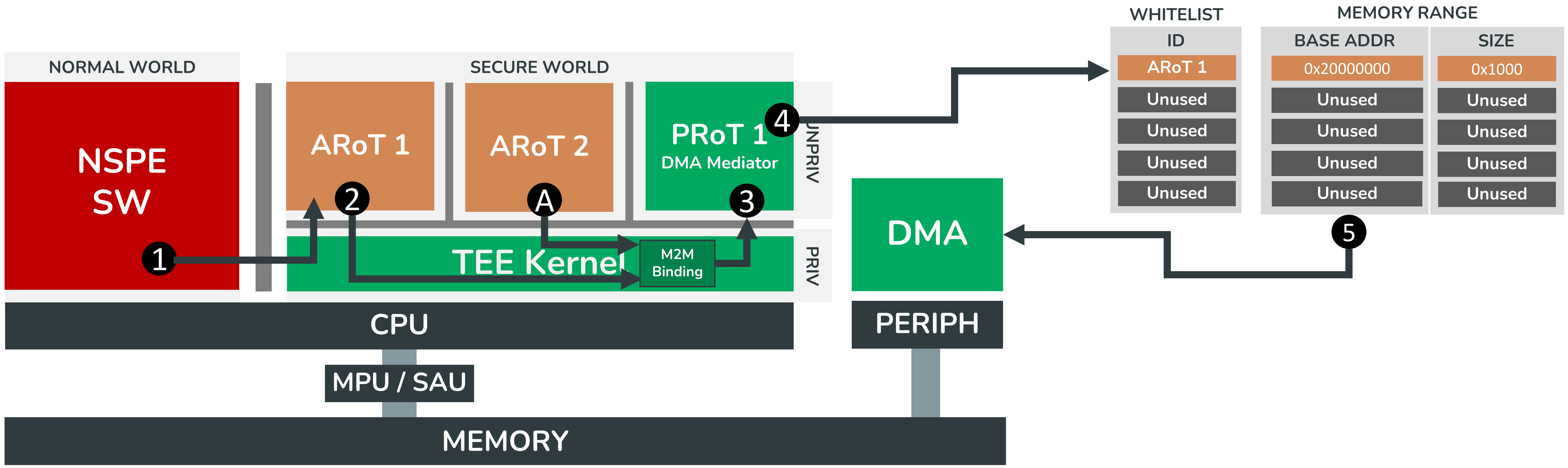
2 ARoT 1 requests access to DMA mediator

3 TEE Kernel Invokes DMA Mediator

4 DMA Mediator Checks Access Permissions and Memory Range

5 DMA Memory Access Granted to ARoT 1

DMA MEDIATION



WHITELIST	MEMORY RANGE	
ID	BASE ADDR	SIZE
ARoT 1	0x20000000	0x1000
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused

❶ NS calls ARoT 1

❷ ARoT 1 requests access to DMA mediator

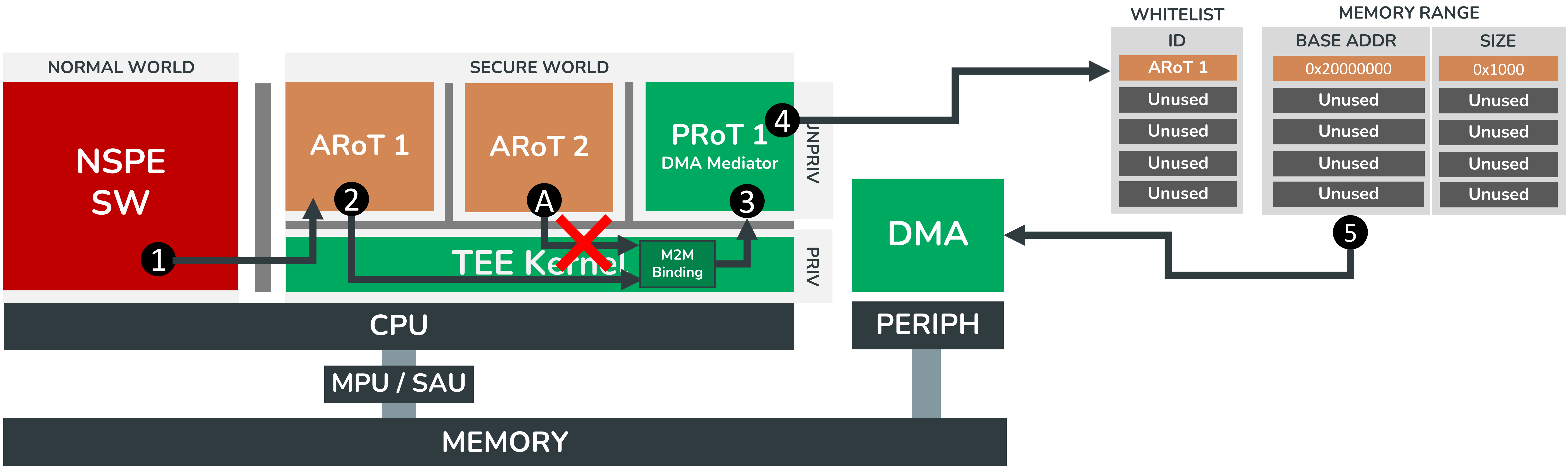
❸ TEE Kernel Invokes DMA Mediator

❹ DMA Mediator Checks Access Permissions and Memory Range

❺ DMA Memory Access Granted to ARoT 1

Ⓐ ARoT 2 requests access to DMA mediator

DMA MEDIATION



WHITELIST	MEMORY RANGE	
ID	BASE ADDR	SIZE
ARoT 1	0x20000000	0x1000
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused
Unused	Unused	Unused

1 NS calls ARoT 1

2 ARoT 1 requests access to DMA mediator

3 TEE Kernel Invokes DMA Mediator

4 DMA Mediator Checks Access Permissions and Memory Range

5 DMA Memory Access Granted to ARoT 1

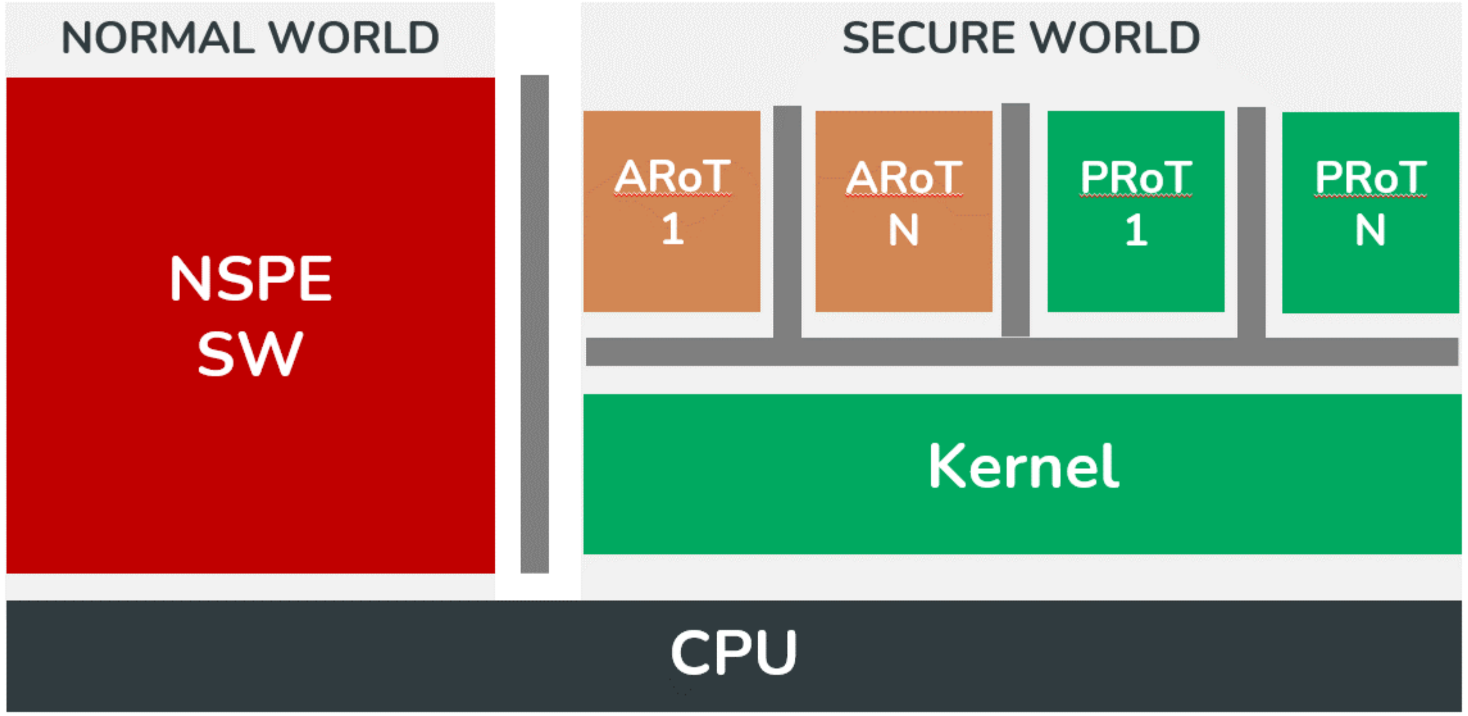
A ARoT 2 requests access to DMA mediator

✗ ARoT 2 is not on the DMA Mediator Whitelist, requested is rejected

What Can Go Wrong

Attack Examples and “Live” Demo

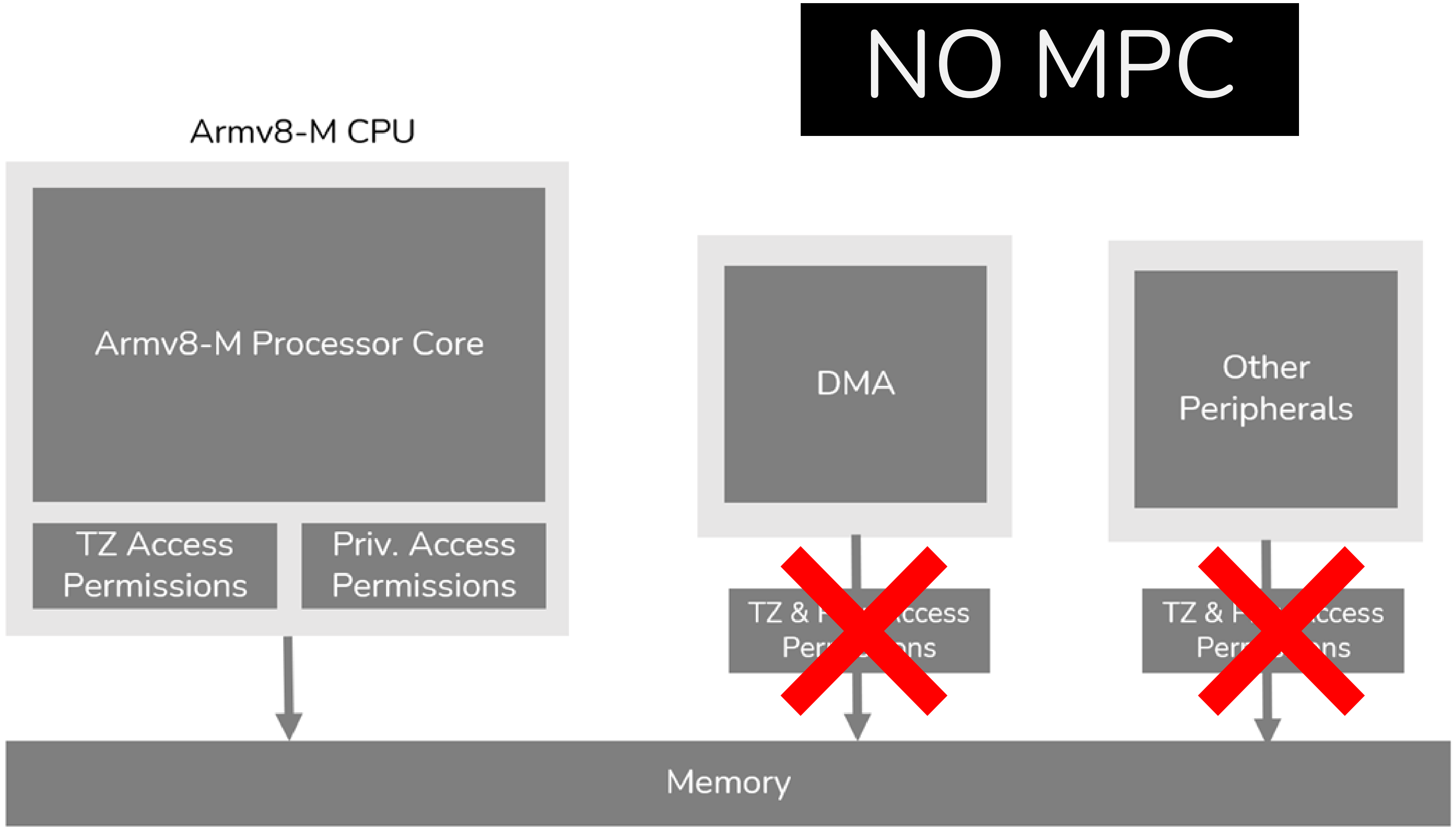
WHEN WE WANT “PSA 3+” ISOLATION



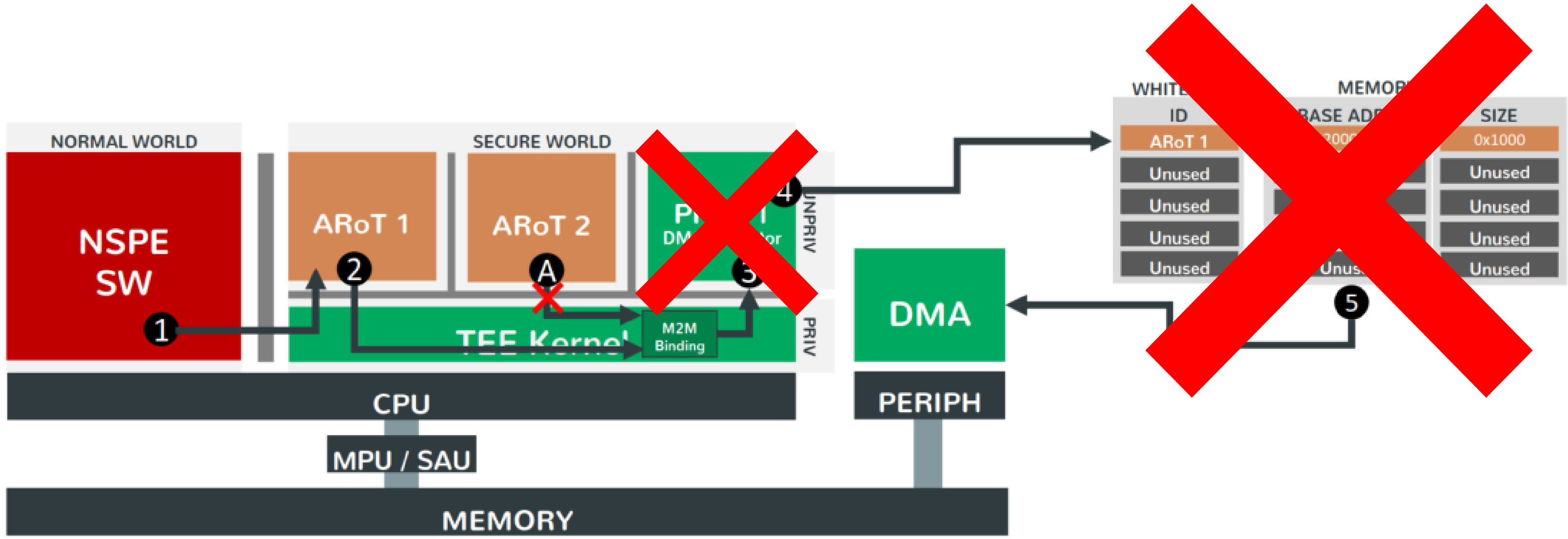
Kinibi-M Architecture

Seems Probably More then PSA Level 3

BUT THE MCU HAS NO MPC

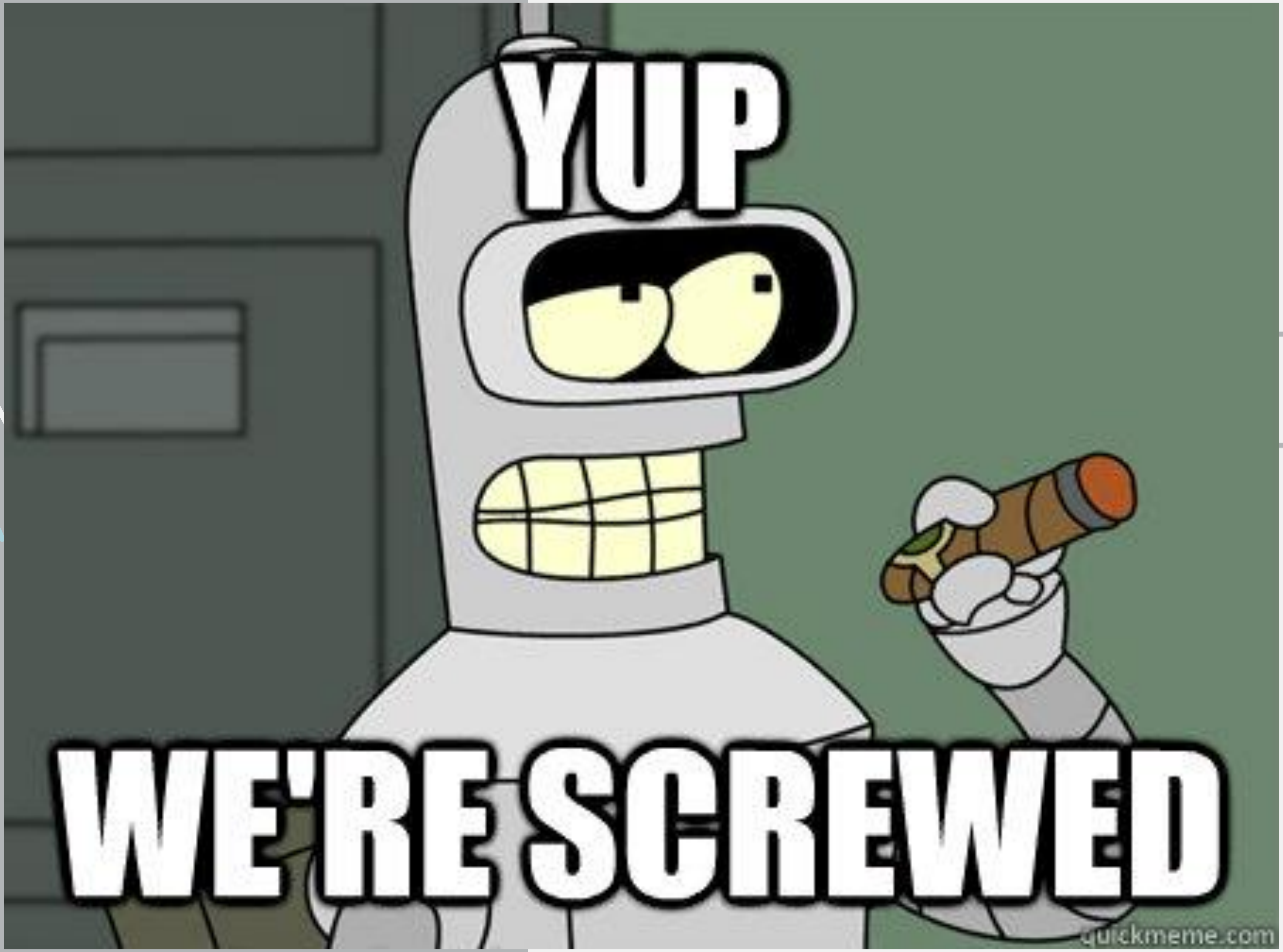


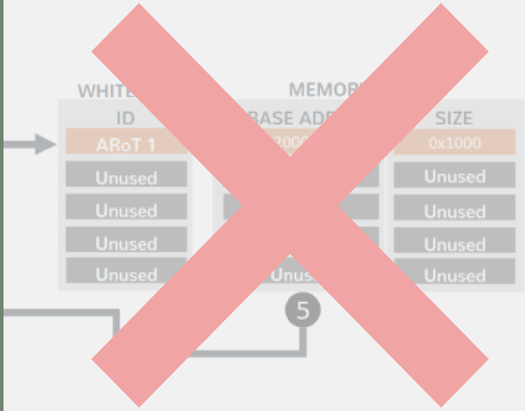
AND FIRMWARE HAS NO DMA MEDIATION



AND FIRN

NO DMA





WHITE ID	MEMOR BASE ADD	SIZE
ARoT 1	[illegible]	0x1000
Unused		Unused
Unused		Unused
Unused		Unused
Unused	Unus	Unused

POTENTIAL EXPLOITS

01

Arbitrary Code Execution in Secure Privilege Mode

Demonstrates the capability to directly tamper with Kinibi-M and achieve **arbitrary code execution** in **secure privileged mode**, rendering all Kinibi-M memory protections ineffective.

Attack 1

02

Steal Proprietary Code from a Secure Module

Demonstrates the capability to **read arbitrary CODE memory** from other secure modules and entirely bypass Kinibi-M's system memory protections.

Attack 2

03

Steal Cryptographic Keys from Kinibi-M Secure Storage

Demonstrates the capability to **read** and **write arbitrary DATA memory** from other secure modules and entirely bypass Kinibi-M's system memory protections.

Attack 3

POTENTIAL EXPLOITS

01

Arbitrary Code Execution in Secure Privilege Mode

Demonstrates the capability to directly tamper with Kinibi-M and achieve **arbitrary code execution** in **secure privileged mode**, rendering all Kinibi-M memory protections ineffective.

Attack 1

02

Steal Proprietary Code from a Secure Module

Demonstrates the capability to **read arbitrary CODE memory** from other secure modules and entirely bypass Kinibi-M's system memory protections.

Attack 2

03

Steal Cryptographic Keys from Kinibi-M Secure Storage

Demonstrates the capability to **read** and **write arbitrary DATA memory** from other secure modules and entirely bypass Kinibi-M's system memory protections.

Attack 3

POTENTIAL EXPLOITS

01

Arbitrary Code Execution in Secure Privilege Mode

Demonstrates the capability to directly tamper with Kinibi-M and achieve **arbitrary code execution** in **secure privileged mode**, rendering all Kinibi-M memory protections ineffective.

Attack 1

02

Steal Proprietary Code from a Secure Module

Demonstrates the capability to **read arbitrary CODE memory** from other secure modules and entirely bypass Kinibi-M's system memory protections.

Attack 2

03

Steal Cryptographic Keys from Kinibi-M Secure Storage

Demonstrates the capability to **read** and **write arbitrary DATA memory** from other secure modules and entirely bypass Kinibi-M's system memory protections.

Attack 3

POTENTIAL EXPLOITS

01

Arbitrary Code Execution in Secure Privilege Mode

Demonstrates the capability to directly tamper with Kinibi-M and achieve **arbitrary code execution** in **secure privileged mode**, rendering all Kinibi-M memory protections ineffective.

Attack 1

02

Steal Proprietary Code from a Secure Module

Demonstrates the capability to **read arbitrary CODE memory** from other secure modules and entirely bypass Kinibi-M's system memory protections.

Attack 2

03

Steal Cryptographic Keys from Kinibi-M Secure Storage

Demonstrates the capability to **read** and **write arbitrary DATA memory** from other secure modules and entirely bypass Kinibi-M's system memory protections.

Attack 3

POTENTIAL EXPLOITS

01

Arbitrary Code Execution in Secure Privilege Mode

Demonstrates the capability to directly tamper with Kinibi-M and achieve **arbitrary code execution** in **secure privileged mode**, rendering all Kinibi-M memory protections ineffective.

Attack 1

02

Steal Proprietary Code from a Secure Module

Demonstrates the capability to **read arbitrary CODE memory** from other secure modules and entirely bypass Kinibi-M's system memory protections.

Attack 2

03

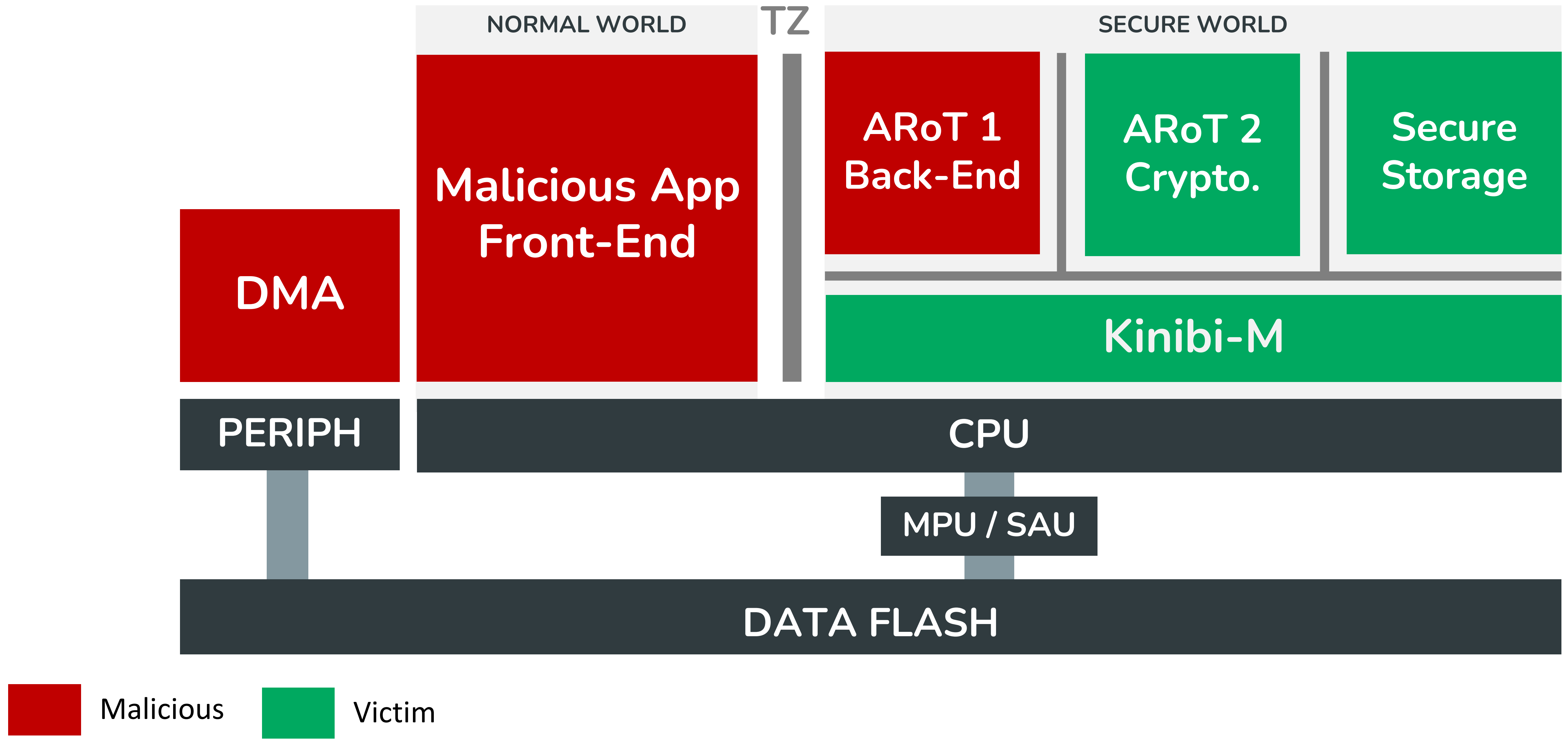
Steal Cryptographic Keys from Kinibi-M Secure Storage

Demonstrates the capability to **read** and **write arbitrary DATA memory** from other secure modules and entirely bypass Kinibi-M's system memory protections.

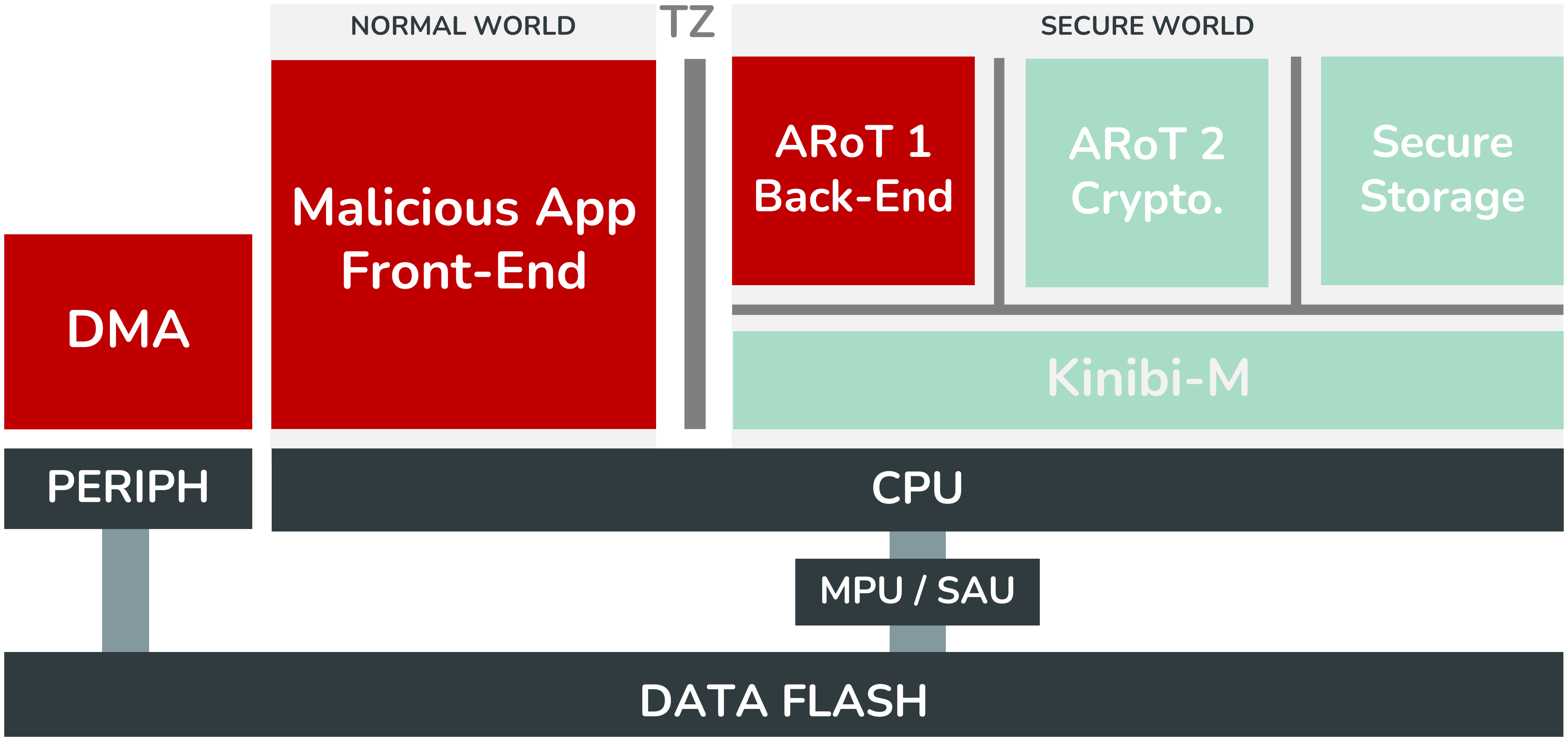
Attack 3

Steal Cryptographic Keys from Kinibi-M Secure Storage

ATTACK 3 STEALING KEYS

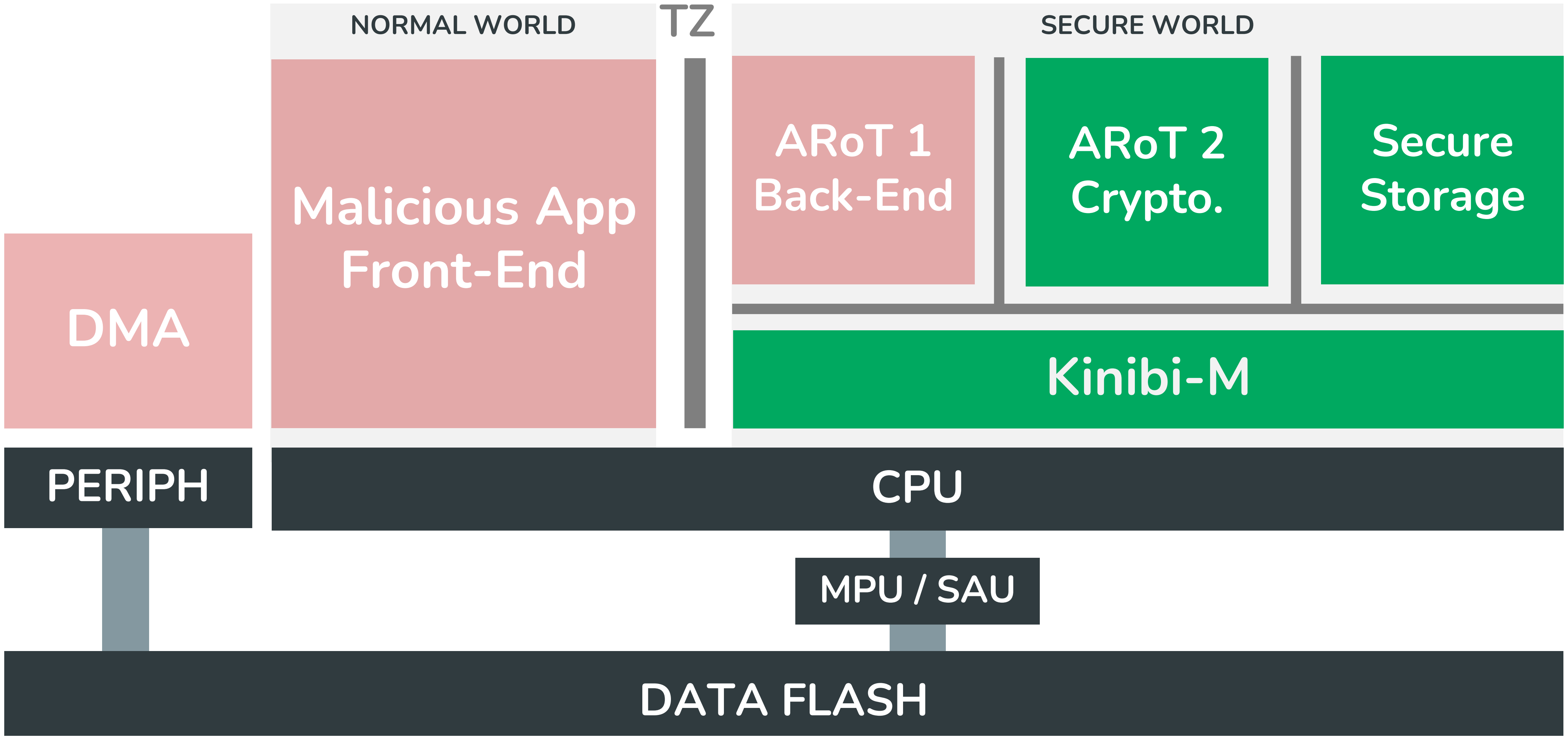


ATTACK 3 STEALING KEYS



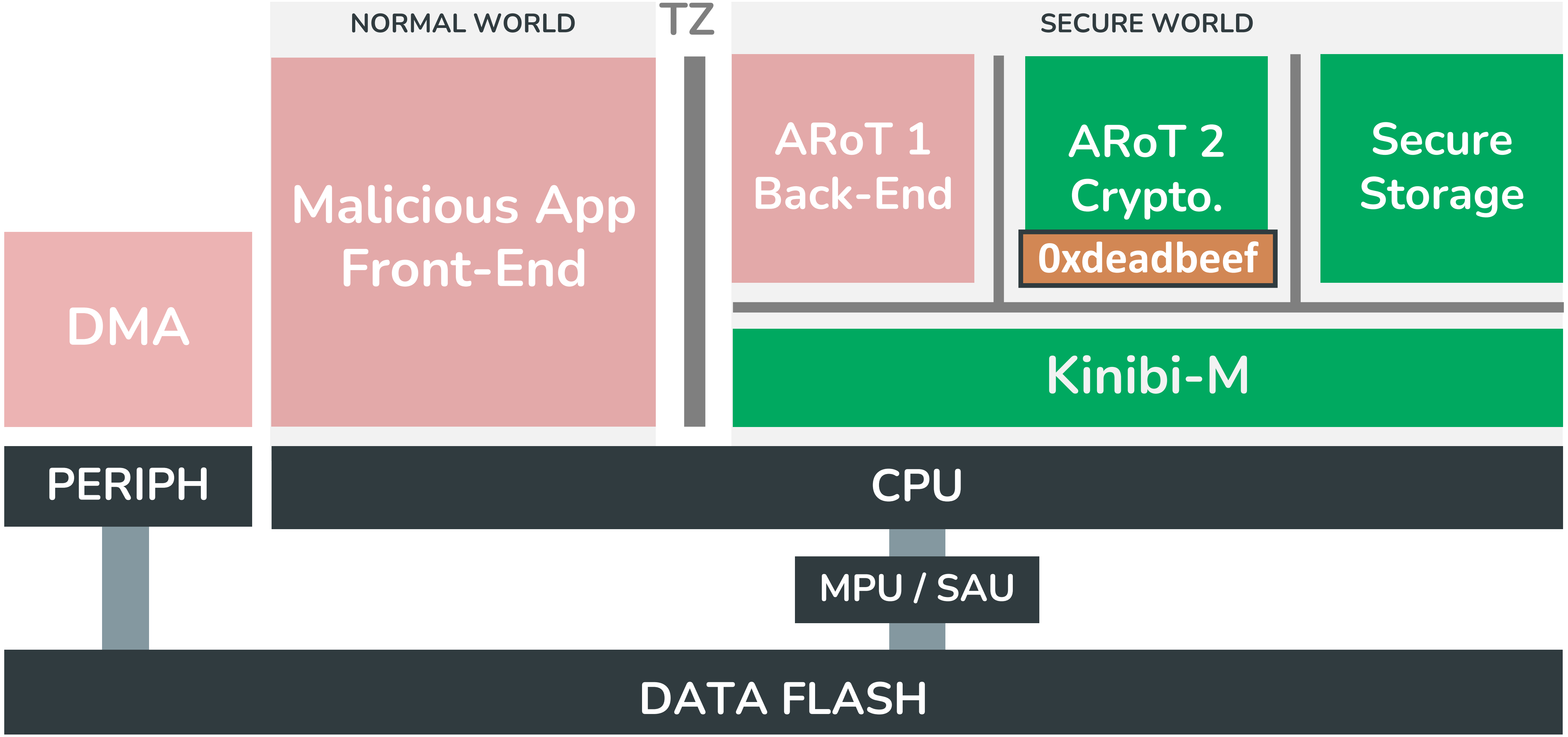
🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS



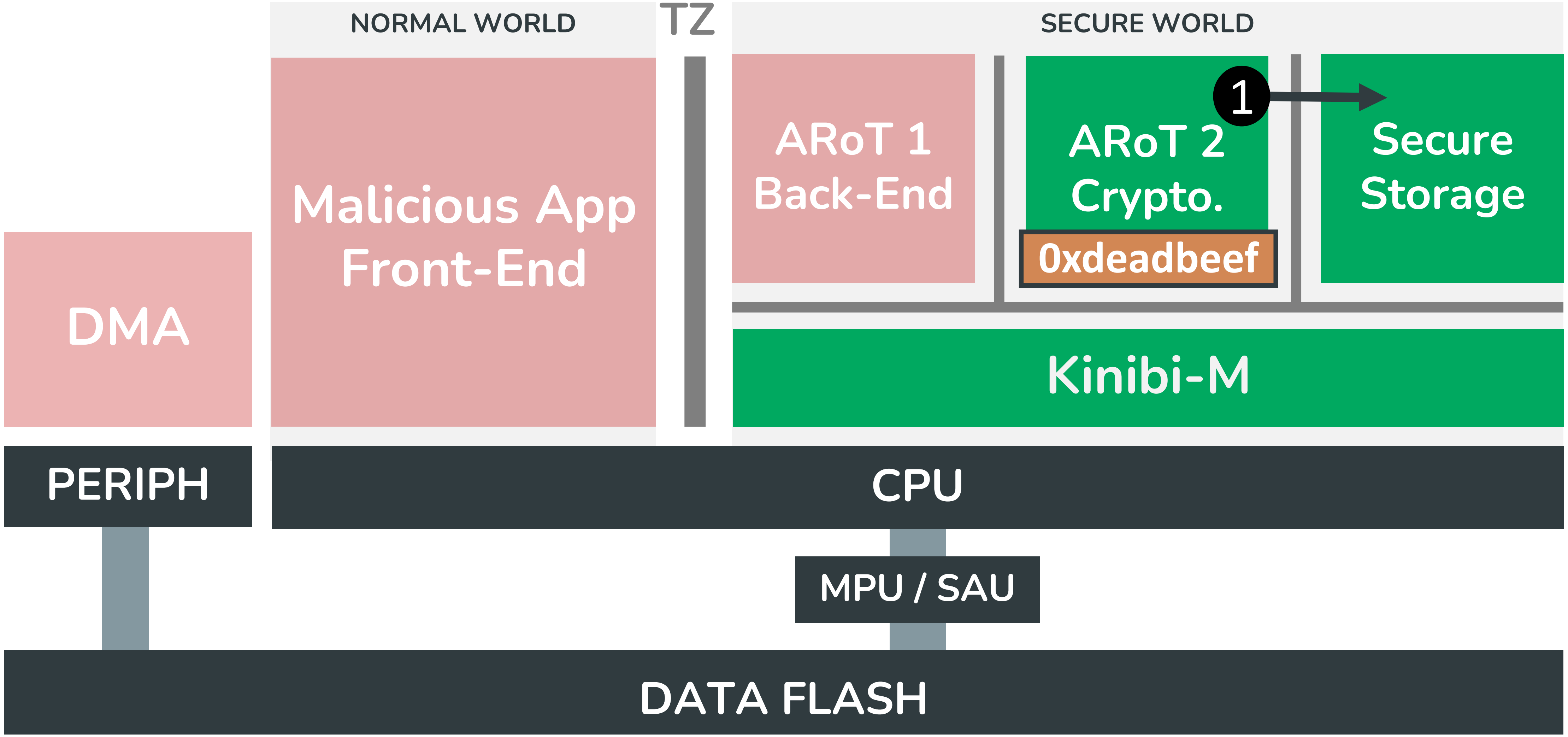
🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS



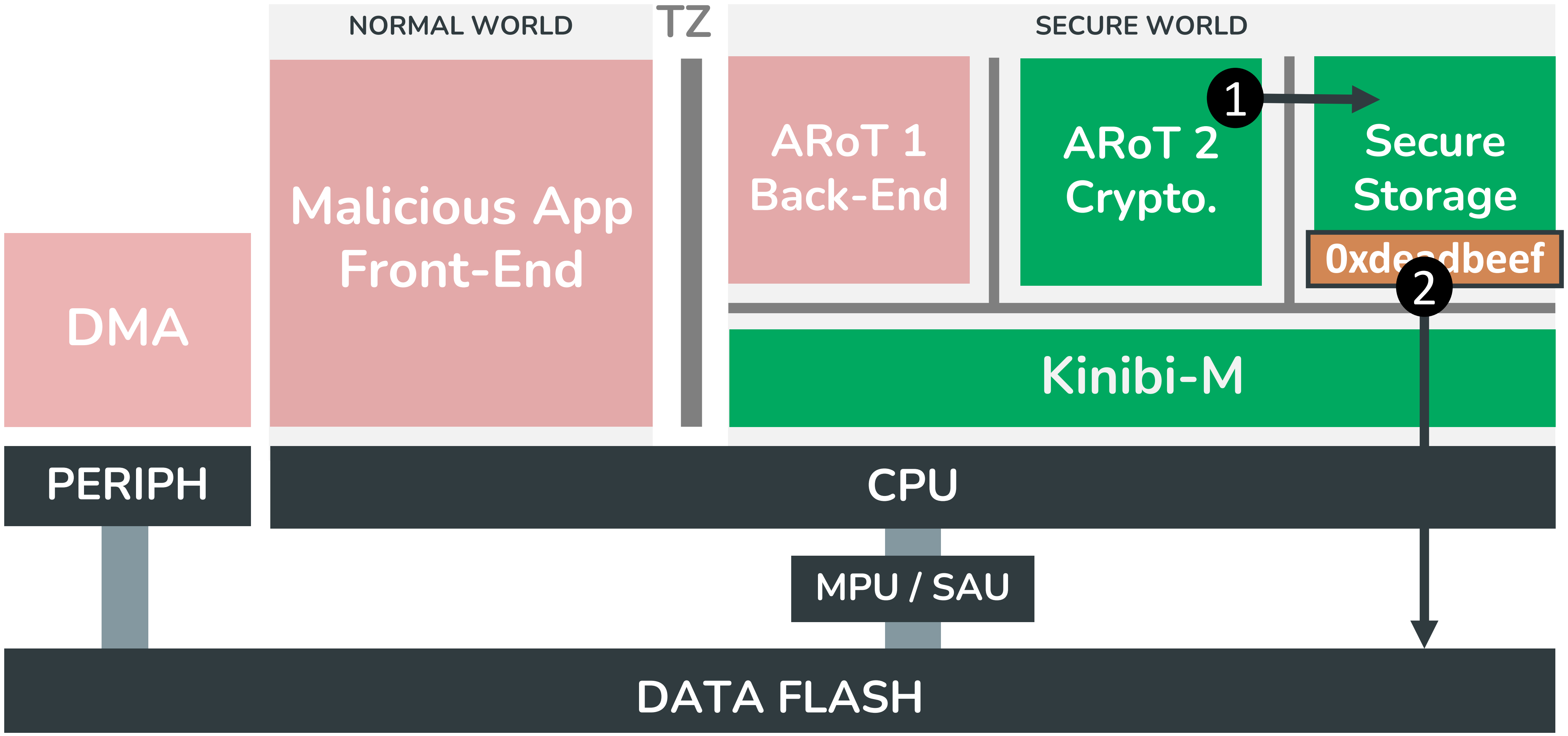
🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS



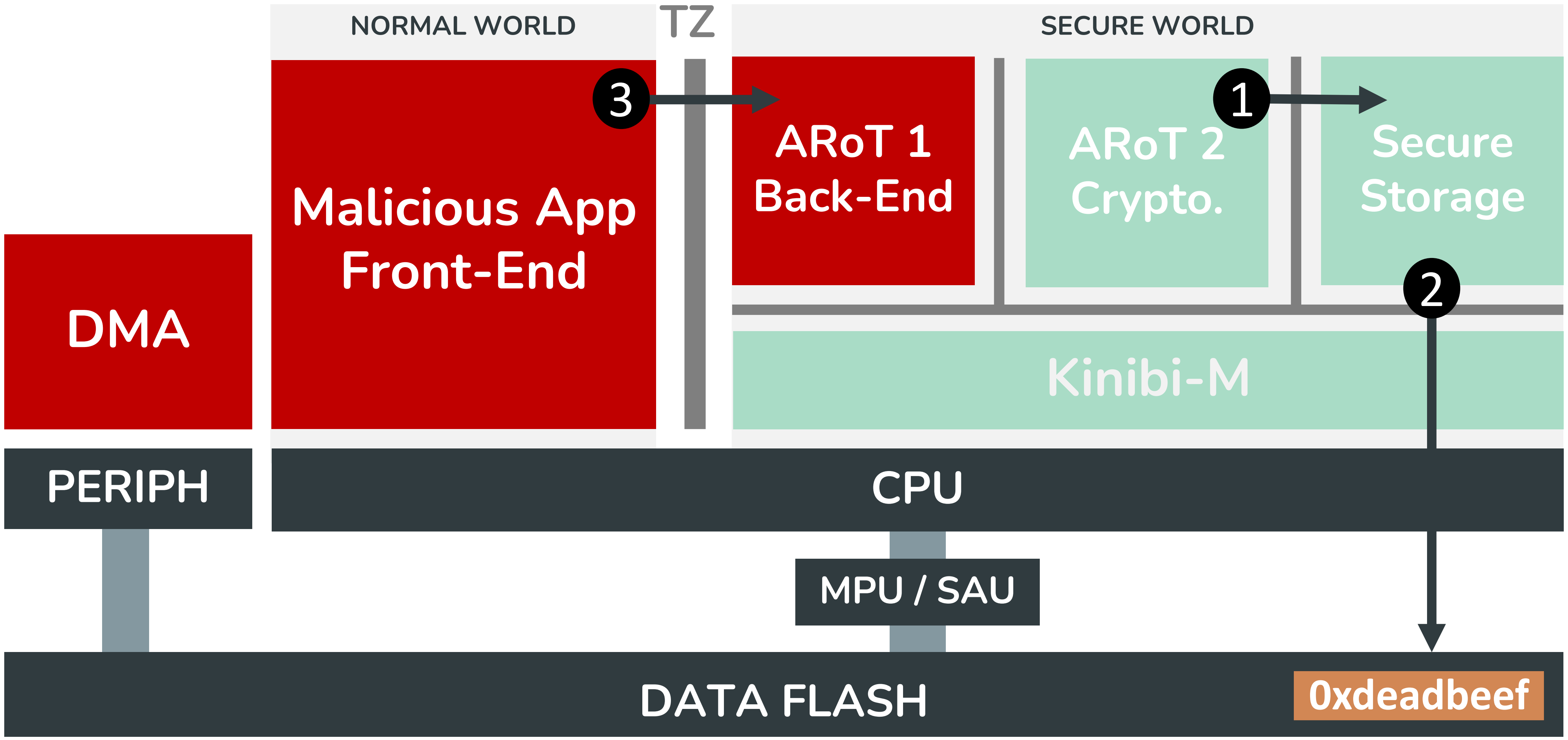
🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS



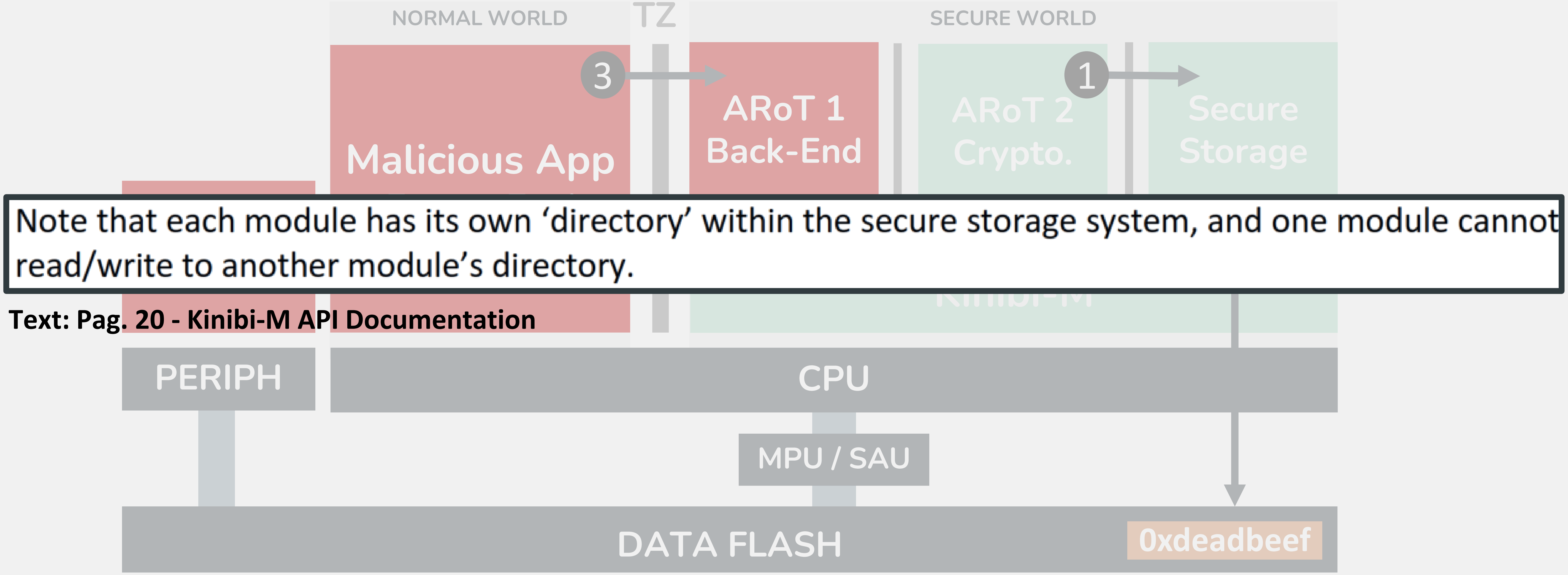
🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS



🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS

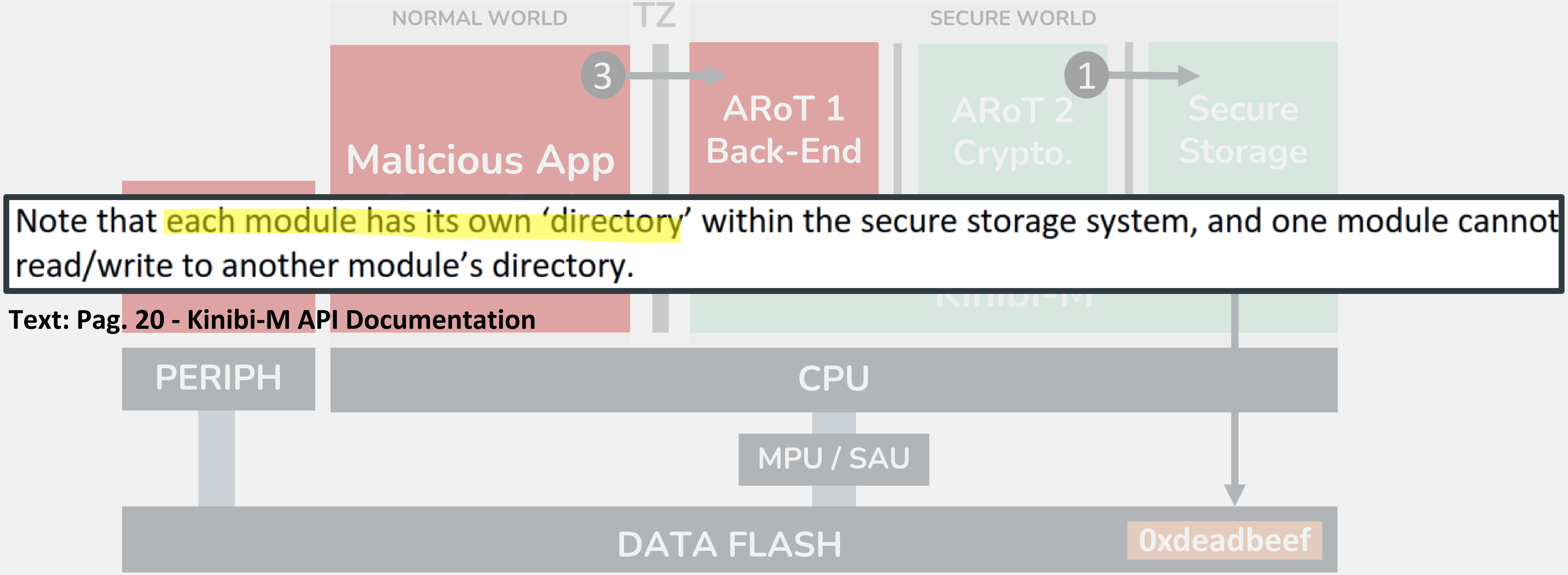


Note that each module has its own 'directory' within the secure storage system, and one module cannot read/write to another module's directory.

Text: Pag. 20 - Kinibi-M API Documentation

🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS

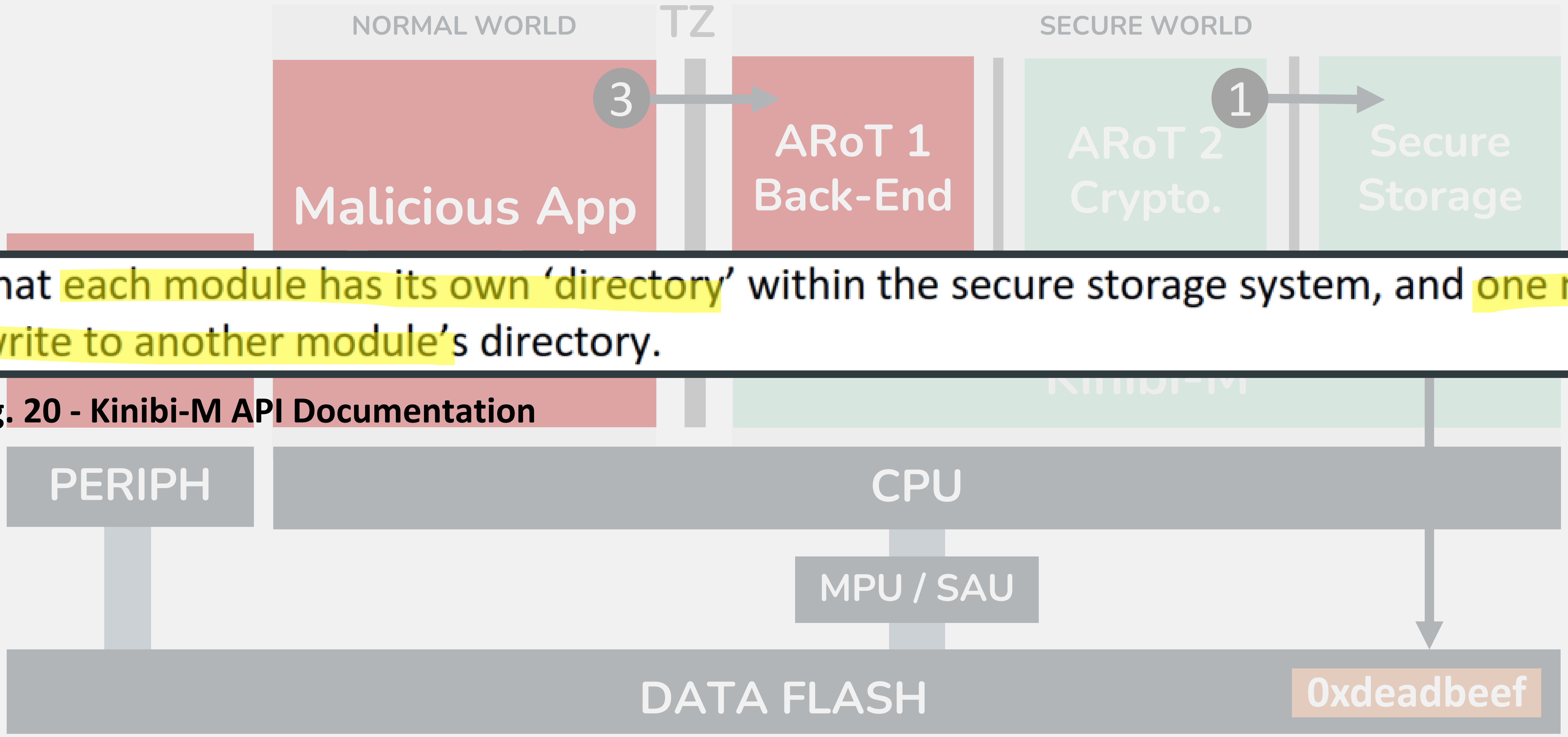


Note that each module has its own 'directory' within the secure storage system, and one module cannot read/write to another module's directory.

Text: Pag. 20 - Kinibi-M API Documentation

🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS

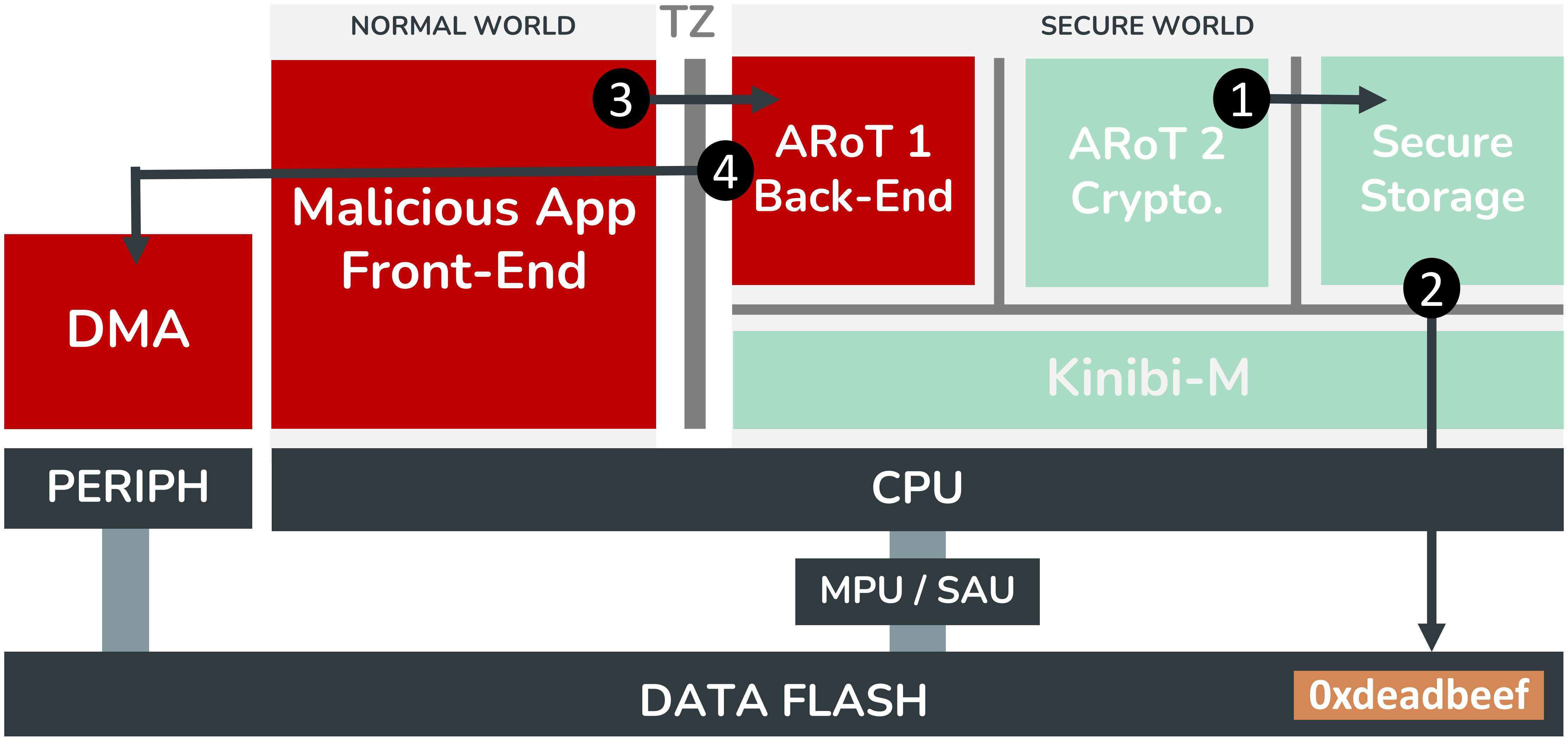


Note that each module has its own 'directory' within the secure storage system, and one module cannot read/write to another module's directory.

Text: Pag. 20 - Kinibi-M API Documentation

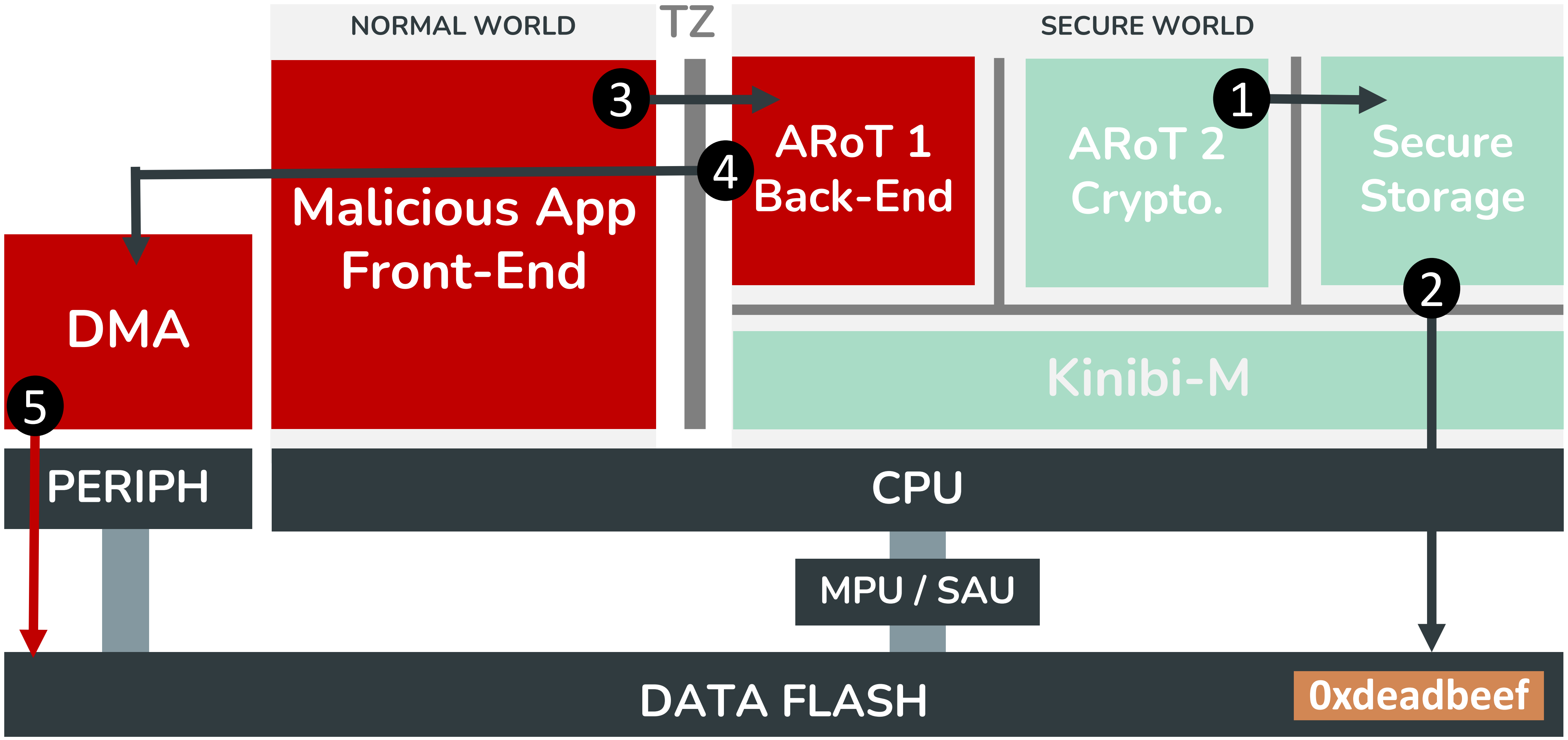
🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS



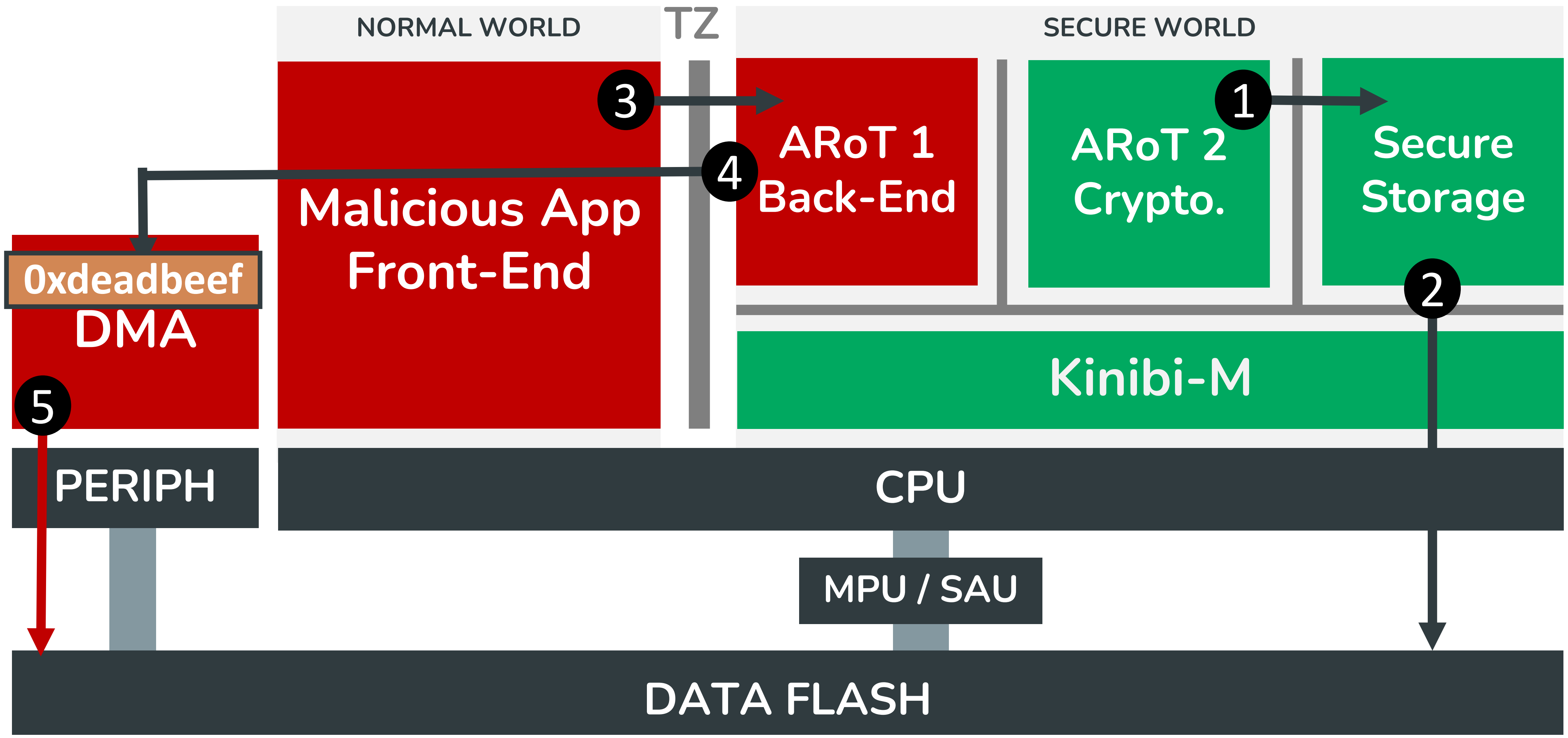
🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS



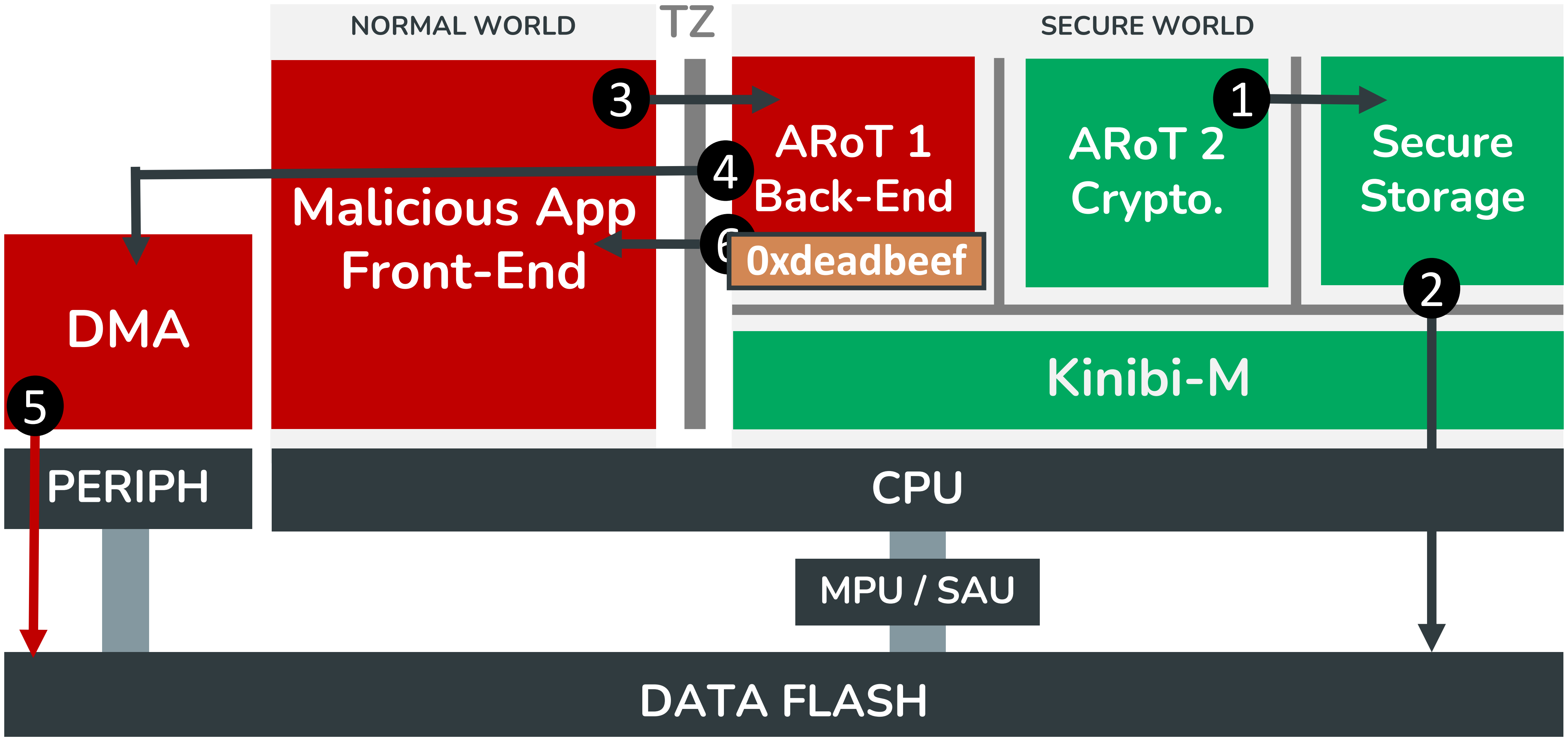
🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS



🟥 Malicious 🟩 Victim

ATTACK 3 STEALING KEYS



🟥 Malicious 🟩 Victim

Live Demo

Video

Lessons Learned

Advices for HW & SW providers and System Designers

LESSONS

LESSONS

#1

For Hardware Providers

LESSONS

#1

For Hardware Providers

For Firmware Providers

#2

LESSONS

#1

For Hardware Providers

#2

For Firmware Providers

#3

For System's Users

LESSONS

#1

For Hardware Providers

Hardware providers should **implement protections** at the **system-level** that takes in account both **privilege levels** and **security states**.

For Firmware Providers

#2

#3

For System's Users

LESSONS

#1

For Hardware Providers

RECOMENDED

Hardware providers should **implement protections** at the **system-level** that takes in account both **privilege levels** and **security states**.

For Firmware Providers

#2

#3

For System's Users

LESSONS

#1

For Hardware Providers

RECOMENDED

Hardware providers should **implement protections** at the **system-level** that takes in account both **privilege levels** and **security states**.

For Firmware Providers

#2

#3

For System's Users

NOT RECOMENDED

LESSONS

NXP LPC5500

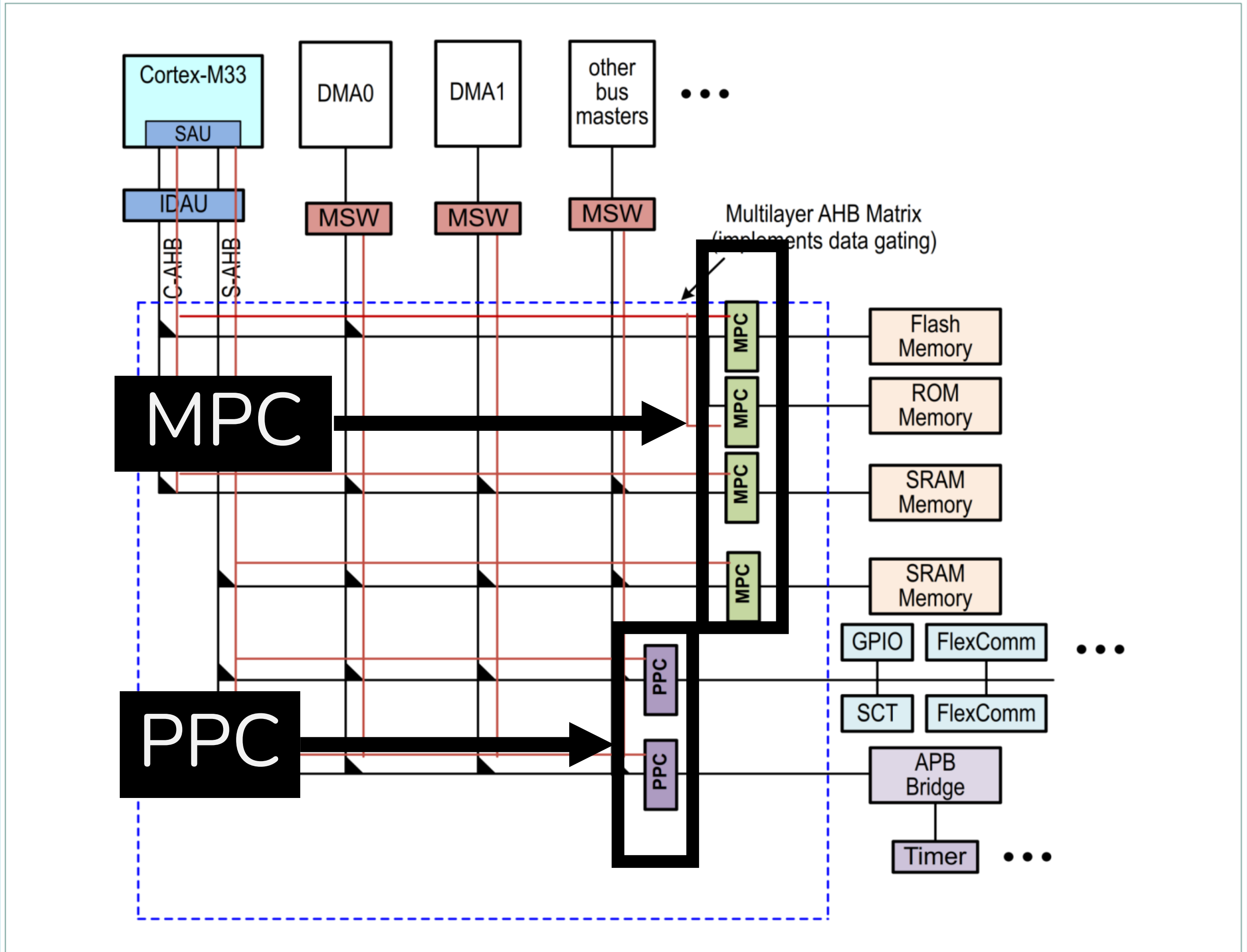


Fig 177. System view with secure AHB bus

#3

For System's Users

#2

Firmware
oviders

NOT RECOMENDED

LESSONS

NXP LPC5500

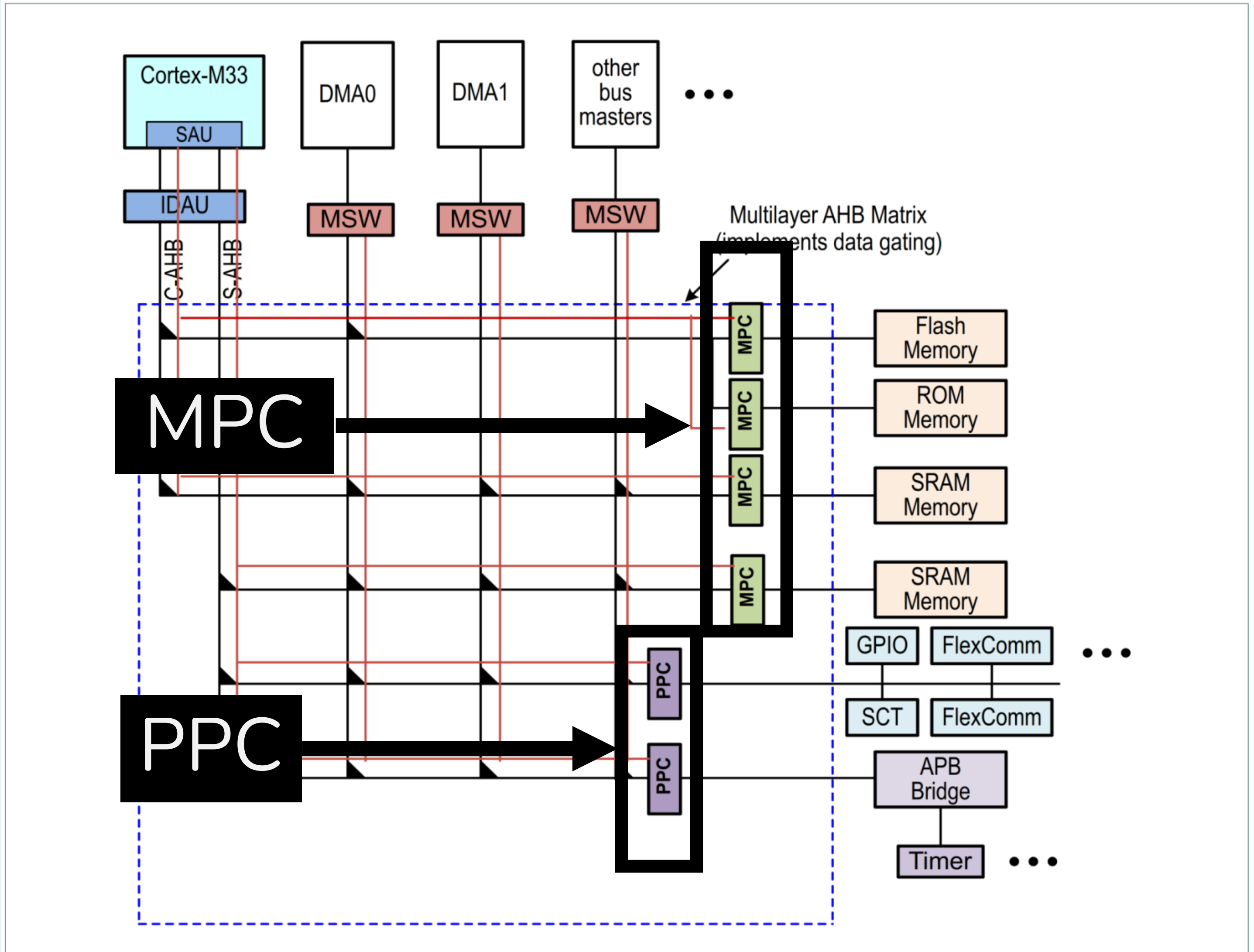
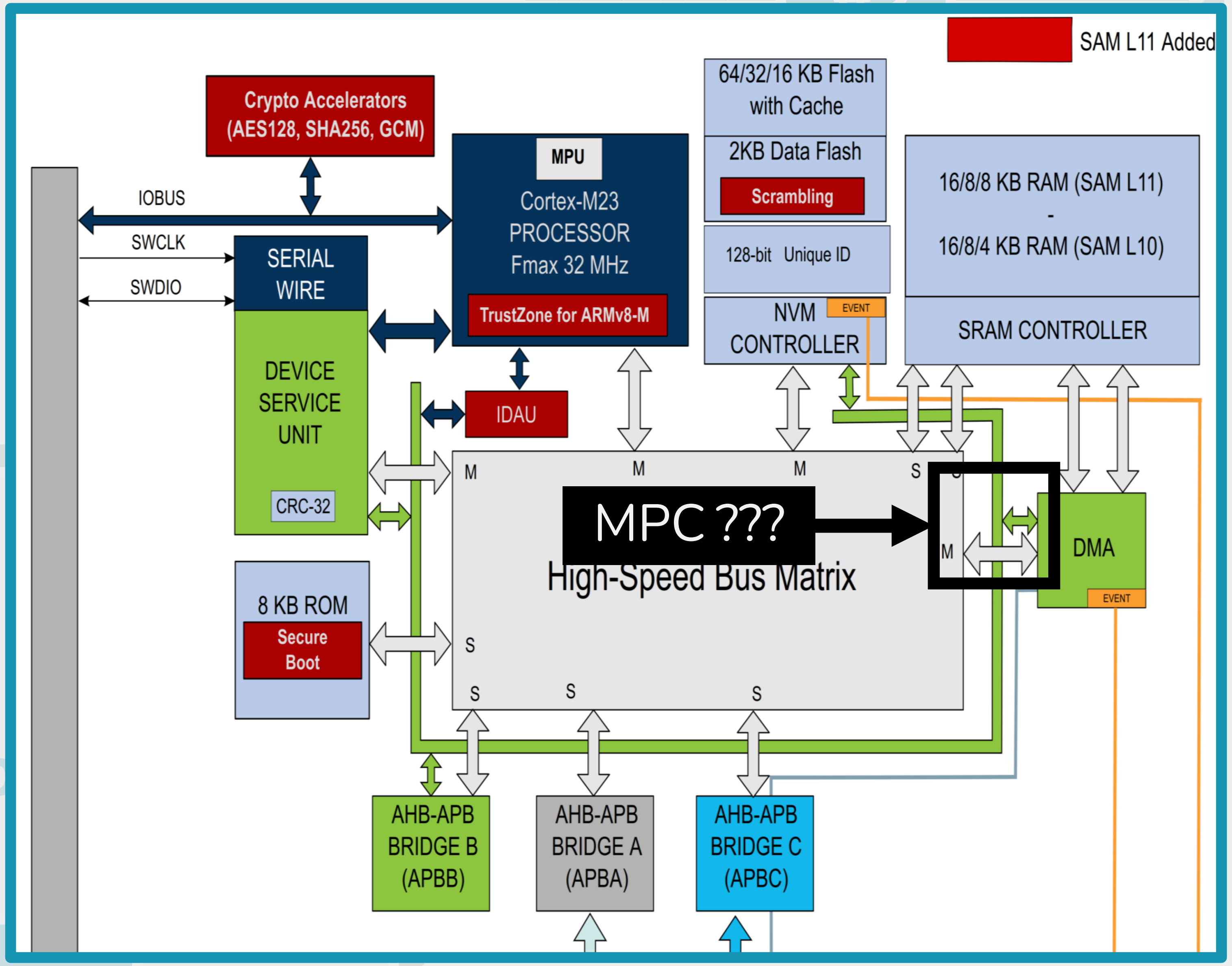


Fig 177. System view with secure AHB bus

MICROCHIP SAML11



LESSONS

#1

For Hardware Providers

Hardware providers should **implement protections** at the **system-level** that takes in account both **privilege levels** and **security states**.

For Firmware Providers

#2

#3

For System's Users

LESSONS

#1

For Hardware Providers

Hardware providers should **implement protections** at the **system-level** that takes in account both **privilege levels** and **security states**.

Firmware providers should implement mechanisms that **enforce isolation defined in the PSA standard.**

For Firmware Providers

#2

#3

For System's Users

LESSONS

RECOMENDED

NOT RECOMENDED

#1

For Hardware Providers

Hardware providers should **implement protections** at the **system-level** that takes in account both **privilege levels** and **security states**.

#2

For Firmware Providers

Firmware providers should implement mechanisms that **enforce isolation defined in the PSA standard.**

#3

For System's Users

LESSONS

MULTIZONE

0x5 HEX–Five Security

“To enforce system separation policies, MultiZone built-in support for protected DMA transfers traps all DMA requests and emulates the PMP logic in software”

Pag. 19 - MultiZone. MultiZone® Security Reference Manual, RISC-V. Tech. rep. MultiZone, Nov 2021.

re providers should
ment mechanisms
enforce isolation
ned in the PSA
standard.

r Firmware
Providers

#2

#3

For System’s
Users

NOT RECOMENDED

LESSONS

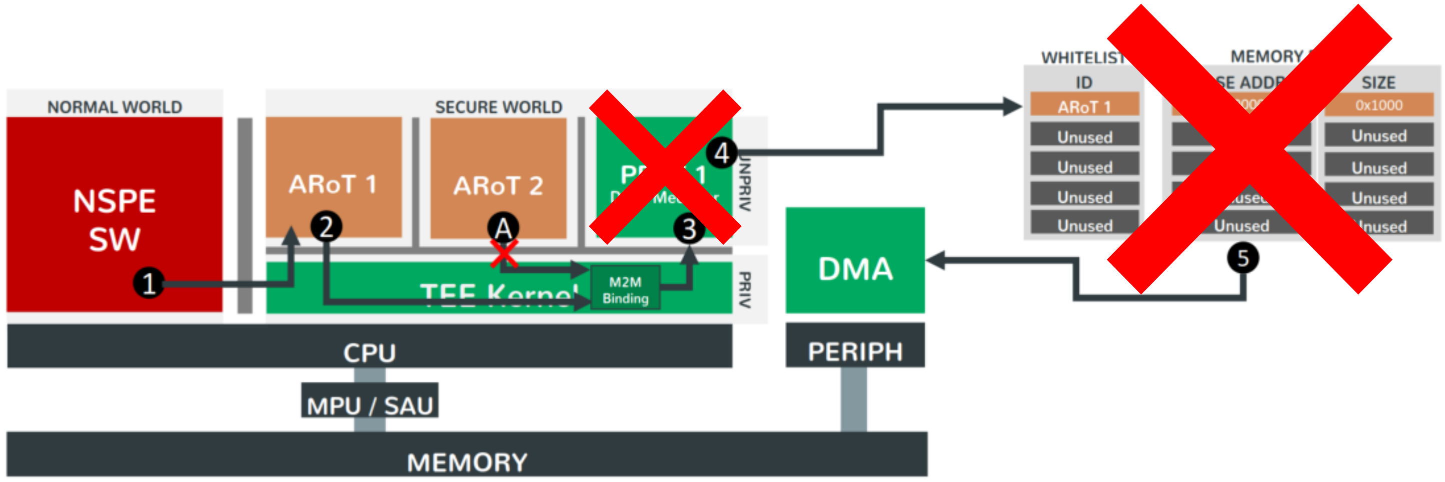
MULTIZONE

0x5 HEX–Five Security

“To enforce system separation policies, MultiZone built-in support for protected DMA transfers traps all DMA requests and emulates the PMP logic in software”

Pag. 19 - MultiZone. MultiZone® Security Reference Manual, RISC-V. Tech. rep. MultiZone, Nov 2021.

KINIBI-M



LESSONS

#1

For Hardware Providers

Hardware providers should **implement protections** at the **system-level** that takes in account both **privilege levels** and **security states**.

Firmware providers should implement mechanisms that **enforce isolation defined in the PSA standard.**

For Firmware Providers

#2

#3

For System’s Users

LESSONS

#1

For Hardware Providers

Hardware providers should **implement protections** at the **system-level** that takes in account both **privilege levels** and **security states**.

#2

For Firmware Providers

Firmware providers should implement mechanisms that **enforce isolation defined in the PSA standard.**

#3

For System's Users

Users (OEMs and software developers) **should be cautious in choosing the system** where they **want to deploy their software**.

WHY NOT AN EXTRA PSA LEVEL?

LESSONS

#1

For Hardware Providers

Hardware providers should **implement protections** at the **system-level** that takes in account both **privilege levels** and **security states.**

#2

For Firmware Providers

Firmware providers should implement mechanisms that **enforce isolation defined in the PSA standard.**

#3

For System's Users

Users (OEMs and software developers) **should be cautious in choosing the system** where they **want to deploy their software.**

LESSONS

#1

For Hard
Provid

Hardware provi
implement prot
the system-le
takes in accou
privilege levels and
security states.

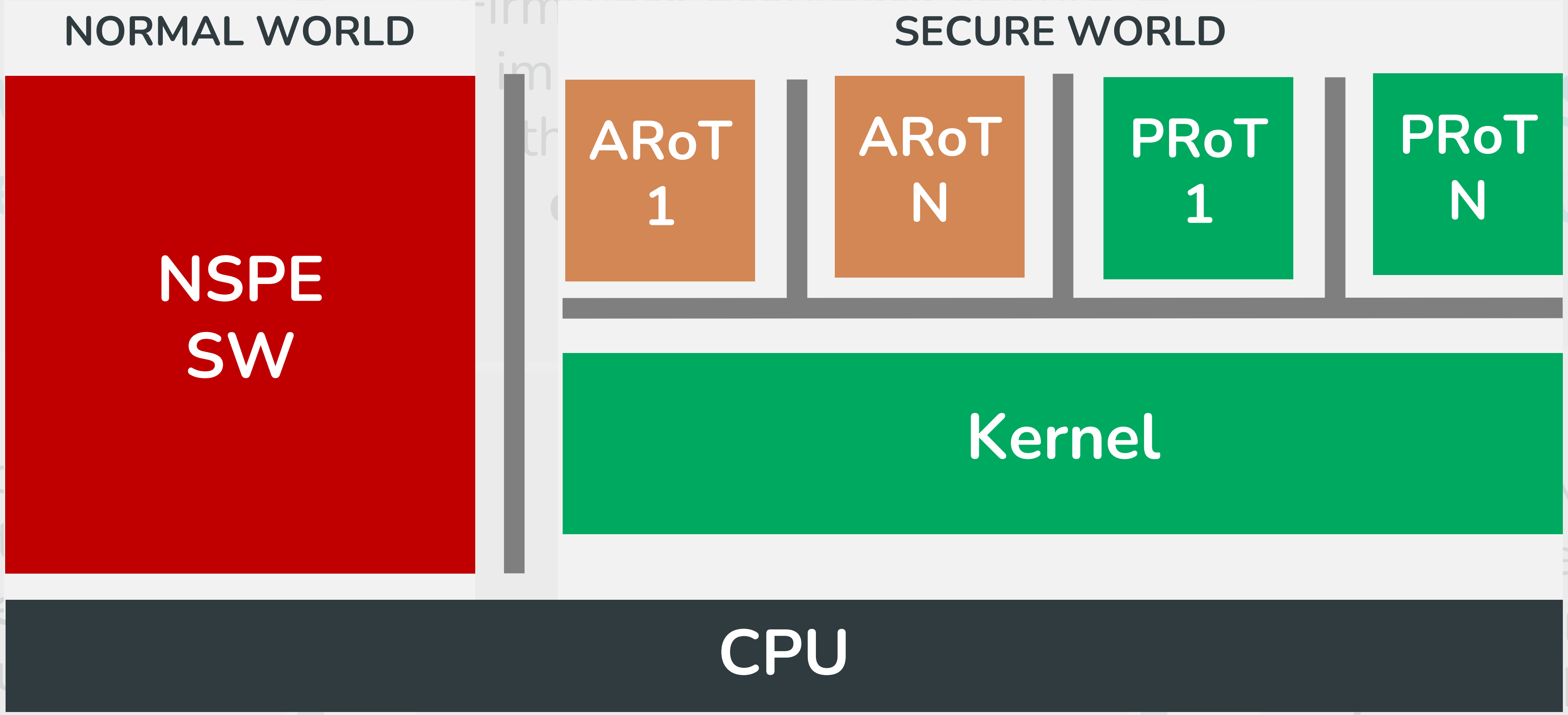
Firmware providers should
im
th

#3

System's
Users

Ms and software
ers) should be
n choosing the
here they want
to deploy their software.

#2



Summary

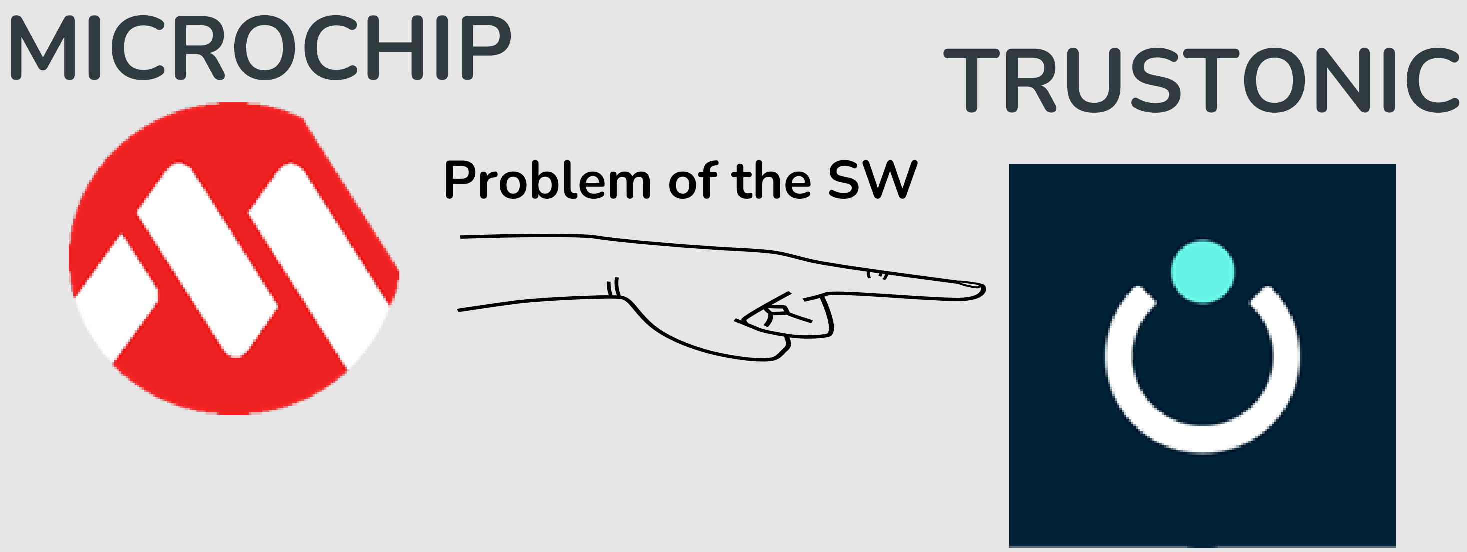
Final Thoughts and BH Sound Bytes

Responsible Disclosure

Responsible Disclosure

MICROCHIP

Responsible Disclosure

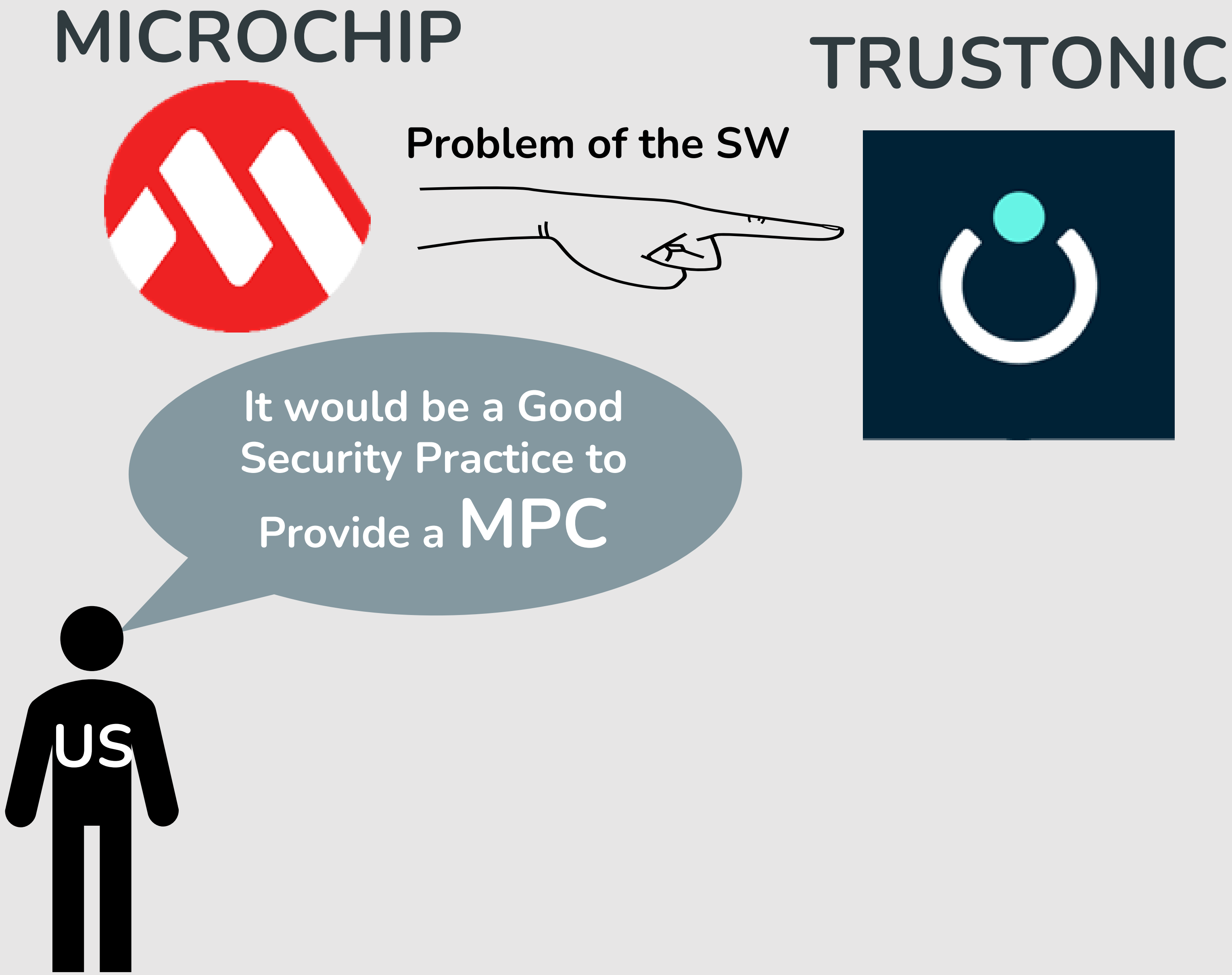


MICROCHIP

Problem of the SW

TRUSTONIC

Responsible Disclosure



MICROCHIP

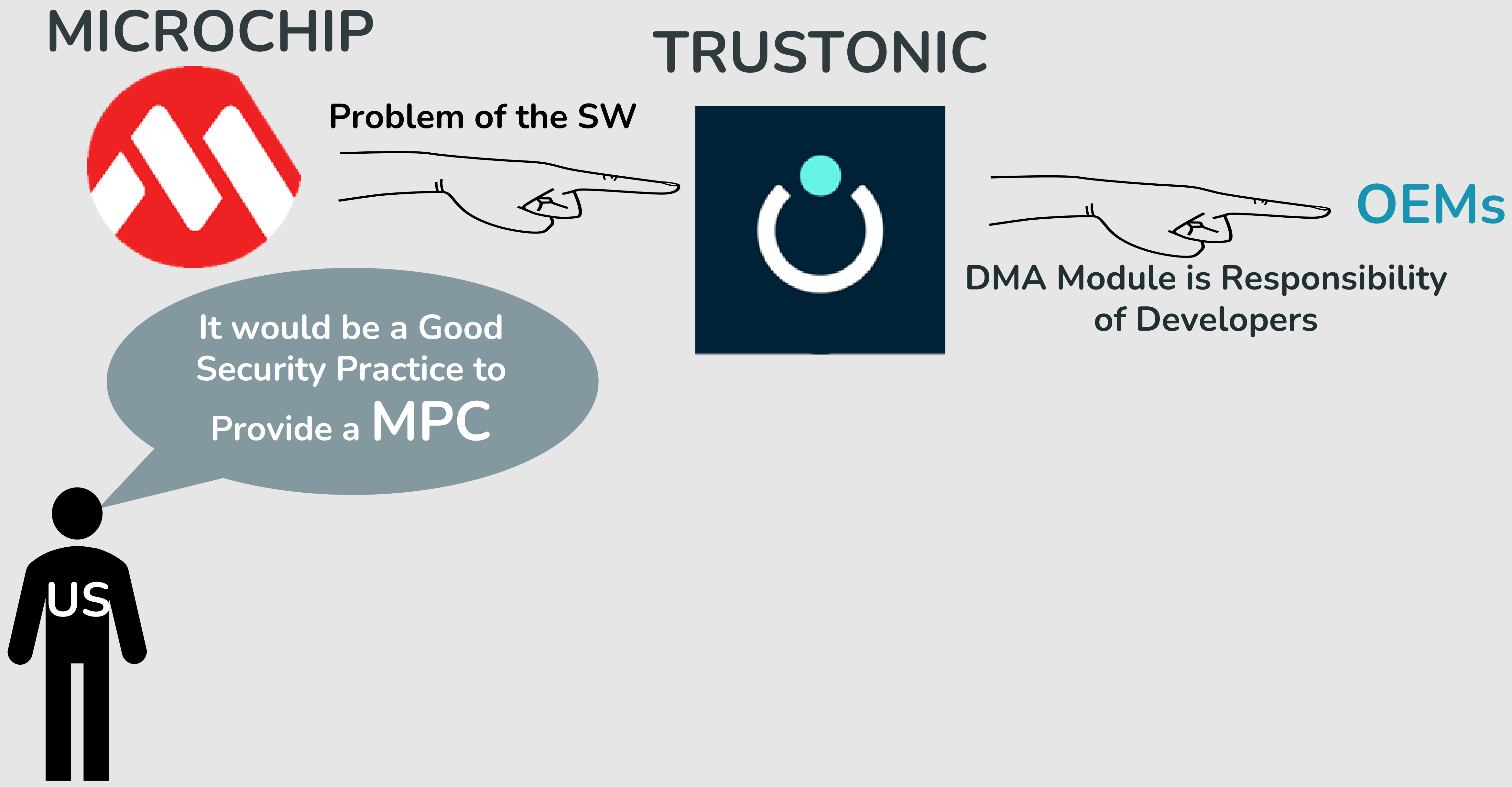
Problem of the SW

TRUSTONIC

It would be a Good Security Practice to Provide a MPC

US

Responsible Disclosure



MICROCHIP

Problem of the SW

TRUSTONIC

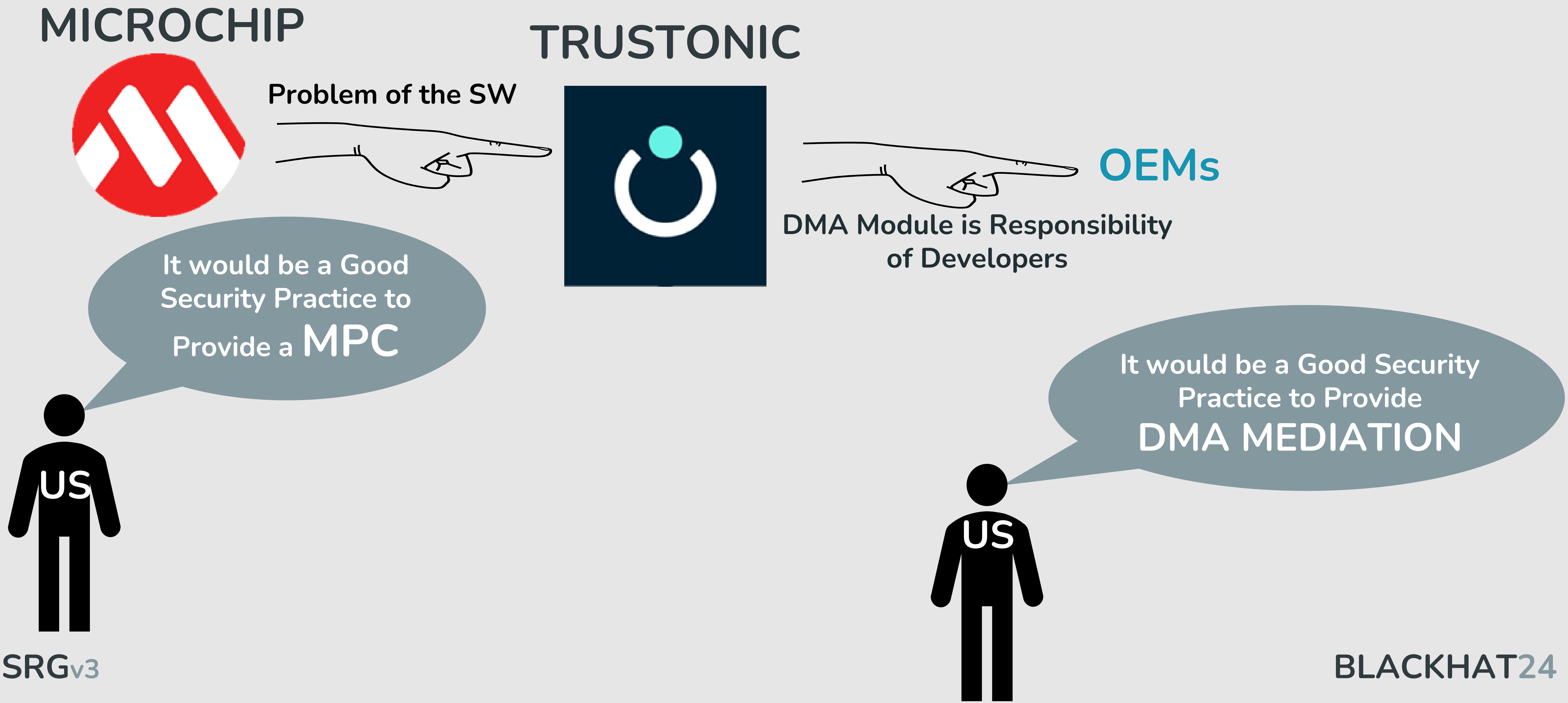
OEMs

DMA Module is Responsibility of Developers

It would be a Good Security Practice to Provide a MPC

US

Responsible Disclosure



MICROCHIP

TRUSTONIC

Problem of the SW

OEMs

DMA Module is Responsibility of Developers

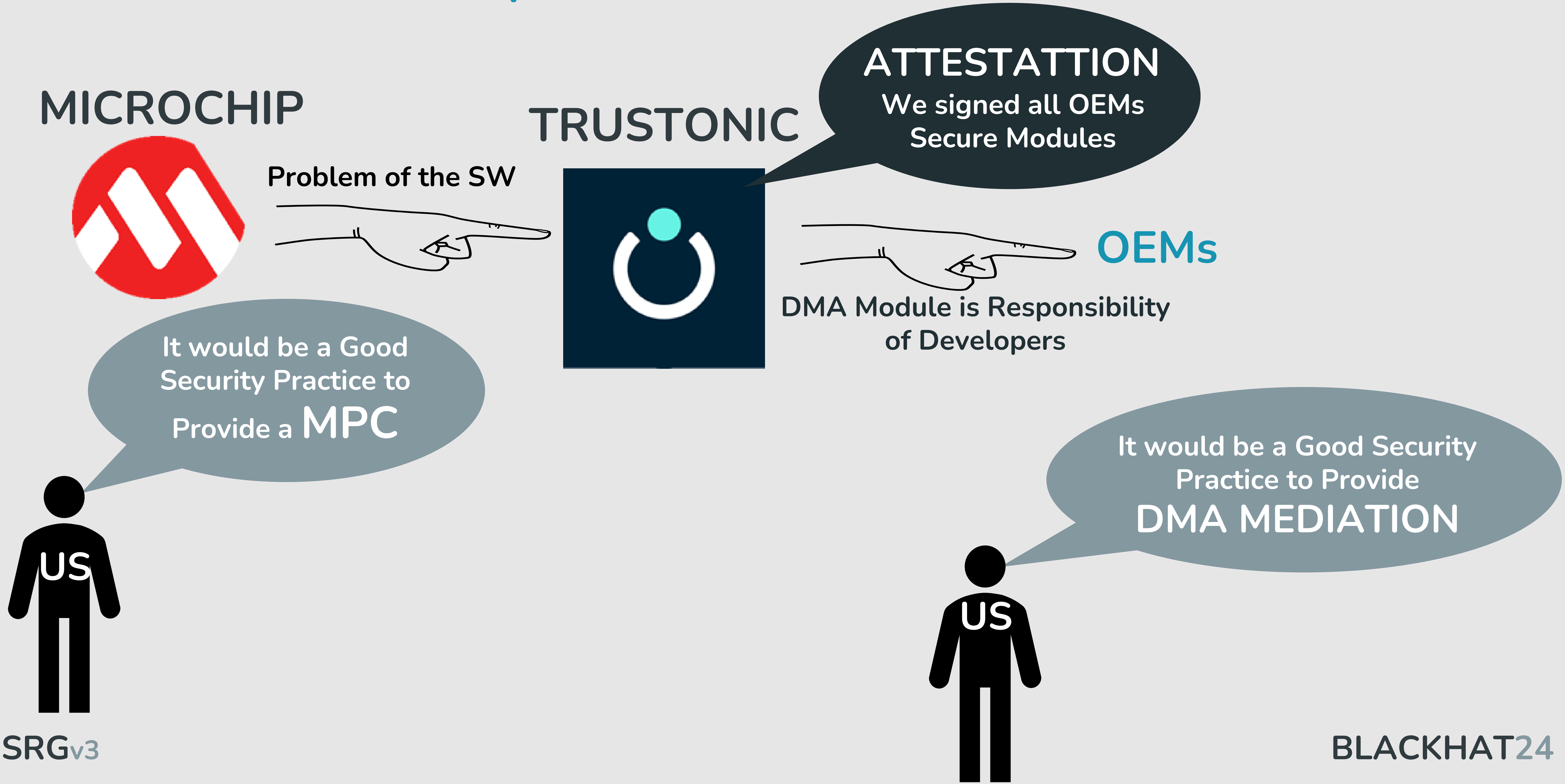
It would be a Good Security Practice to Provide a MPC

US

It would be a Good Security Practice to Provide DMA MEDIATION

US

Responsible Disclosure



MICROCHIP

Problem of the SW

TRUSTONIC

ATTESTATTION

We signed all OEMs Secure Modules

OEMs

DMA Module is Responsibility of Developers

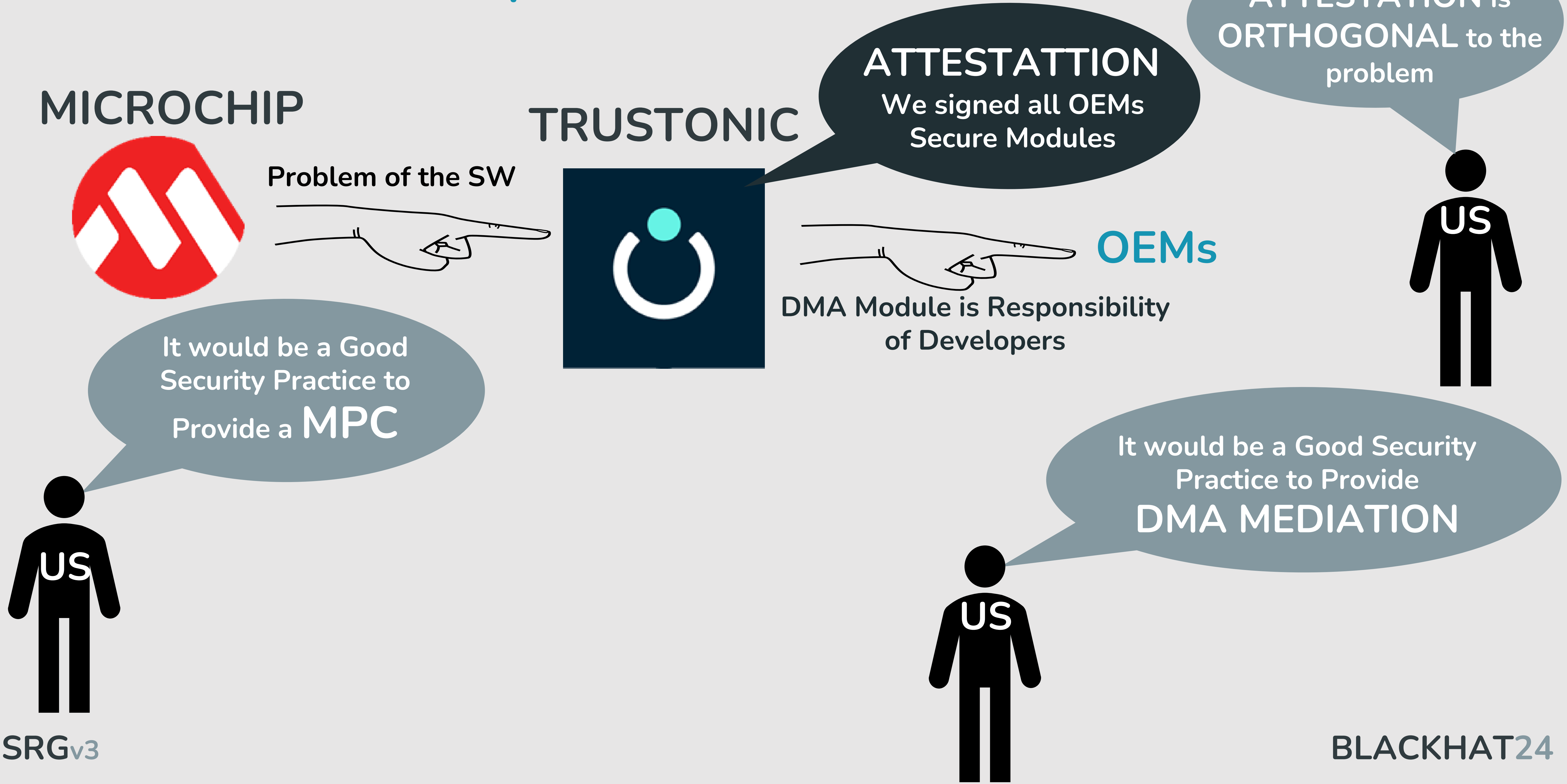
It would be a Good Security Practice to Provide a MPC

US

It would be a Good Security Practice to Provide DMA MEDIATION

US

Responsible Disclosure



MICROCHIP

Problem of the SW

TRUSTONIC

ATTESTATTION
We signed all OEMs
Secure Modules

ATTESTATION is
ORTHOGONAL to the
problem

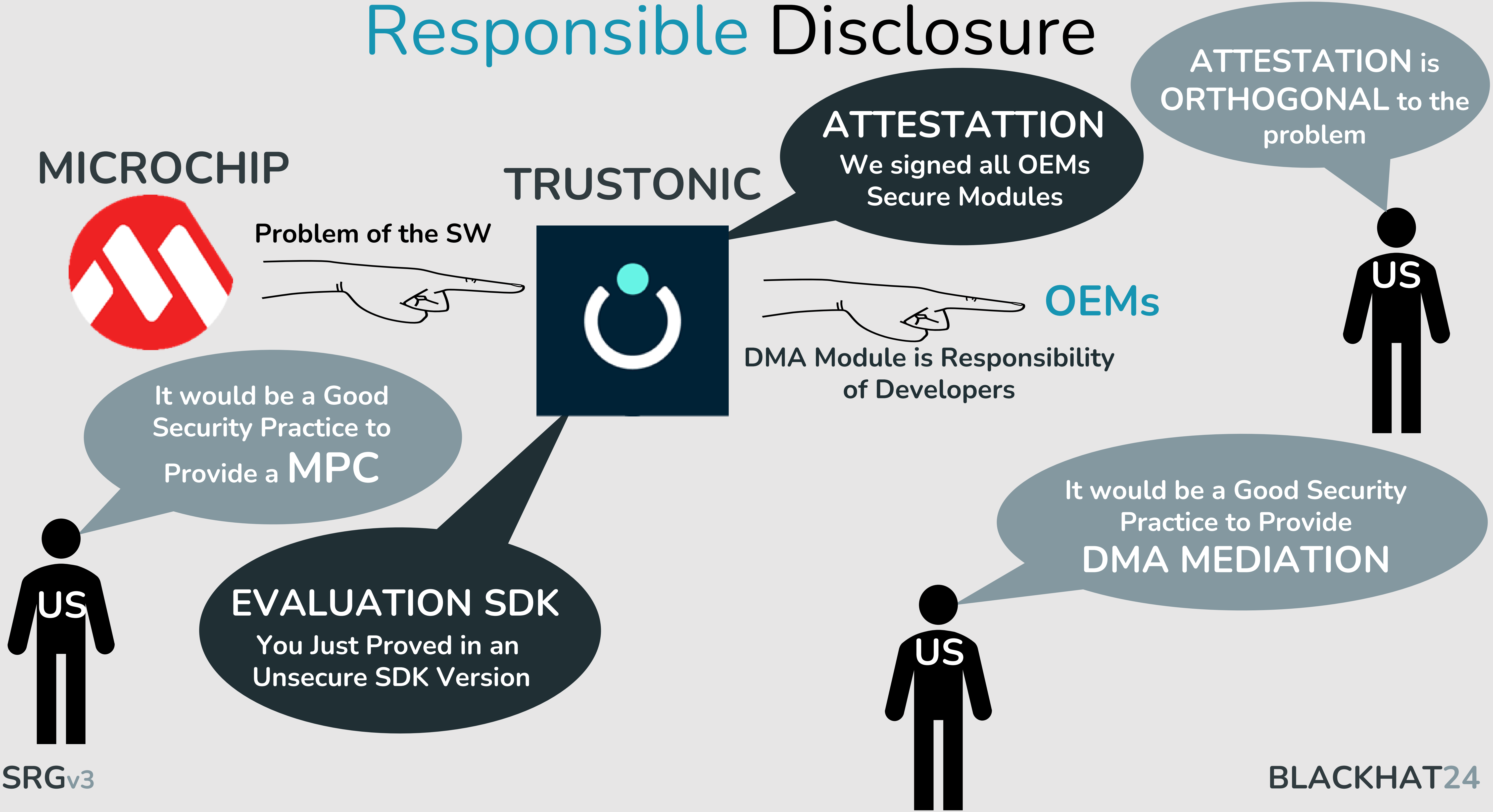
OEMs

DMA Module is Responsibility
of Developers

It would be a Good
Security Practice to
Provide a MPC

It would be a Good Security
Practice to Provide
DMA MEDIATION

Responsible Disclosure



MICROCHIP

Problem of the SW

TRUSTONIC

ATTESTATTION

We signed all OEMs Secure Modules

ATTESTATION is ORTHOGONAL to the problem

US

OEMs

DMA Module is Responsibility of Developers

It would be a Good Security Practice to Provide a MPC

US

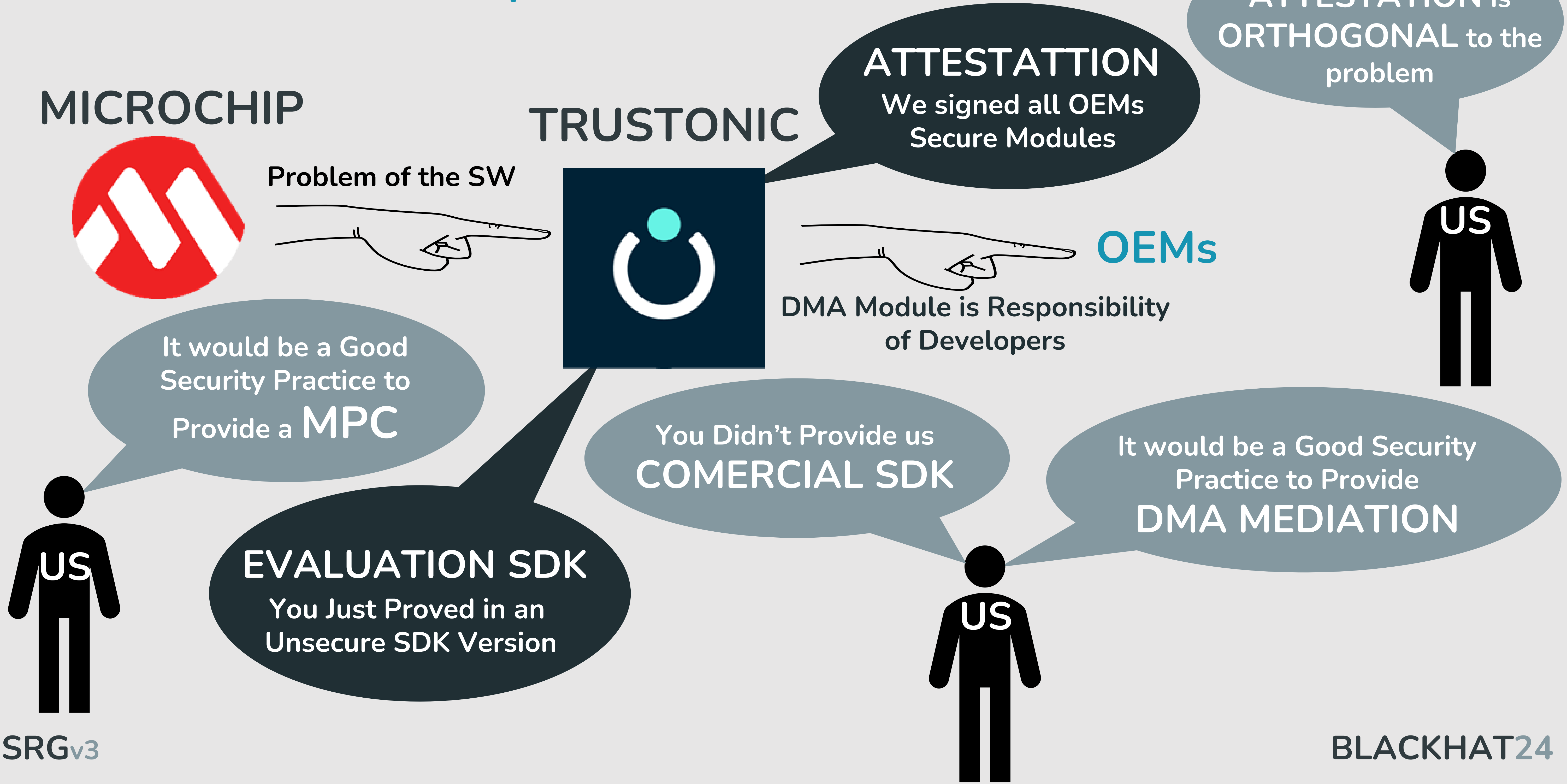
EVALUATION SDK

You Just Proved in an Unsecure SDK Version

It would be a Good Security Practice to Provide DMA MEDIATION

US

Responsible Disclosure



MICROCHIP

Problem of the SW

TRUSTONIC

ATTESTATTION
We signed all OEMs
Secure Modules

ATTESTATION is
ORTHOGONAL to the
problem

US

OEMs

DMA Module is Responsibility
of Developers

It would be a Good
Security Practice to
Provide a **MPC**

US

EVALUATION SDK
You Just Proved in an
Unsecure SDK Version

You Didn't Provide us
COMERCIAL SDK

It would be a Good Security
Practice to Provide
DMA MEDIATION

US

Black Hat
SOUND BYTES

1. We shared our journey on fully assessing an MCU-based TEE (Kinibi-M) targeting a reference TrustZone-M hardware platform (SAML11)

2. We presented how it is possible to bypass CPU-level isolation primitives, and explain the design of a TEE core mechanism (DMA Mediator) to offer such protection;

3. We perform a live demo of one potential exploit that retrieves a cryptographic key from other Secure Partitions bypassing all hardware and software TEE isolation boundaries.

THANK YOU!

Cristiano Rodrigues | Sandro Pinto, PhD
(Centro ALGORITMI / LASI, Universidade do Minho)

id9492@alunos.uminho.pt

𝕏 **@_CRodrigues__**

sandro.pinto@dei.uminho.pt

𝕏 **@sandro2pinto**

Q&A

Cristiano Rodrigues | Sandro Pinto, PhD
(Centro ALGORITMI / LASI, Universidade do Minho)

Cristiano Rodrigues

Sandro Pinto