Appendix. Software security based on Isolation

ARM TrustZone and other technologies

Agenda

- What is isolation? Why is it important?
- What means of isolation we have today on STM32?
- What is Trustzone? How it works? What are the benefits over current solution?
- Show system integration of TrustZone on STM32L5
- Introduce development flow. CMSIS support of TrustZone

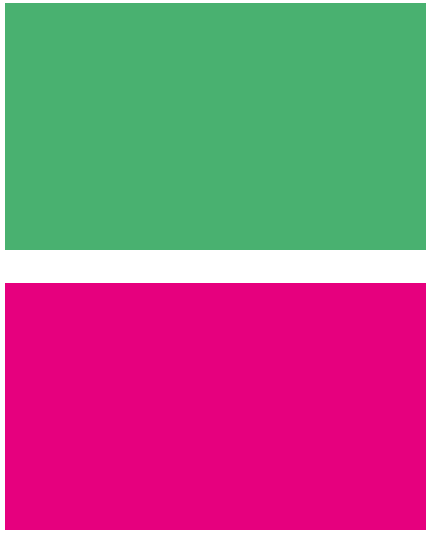
Isolation concept

life.augmented

Convention

- Secure is GREEN (ST2 Lite Green)
- Non-Secure is RED (ST2 Pink)

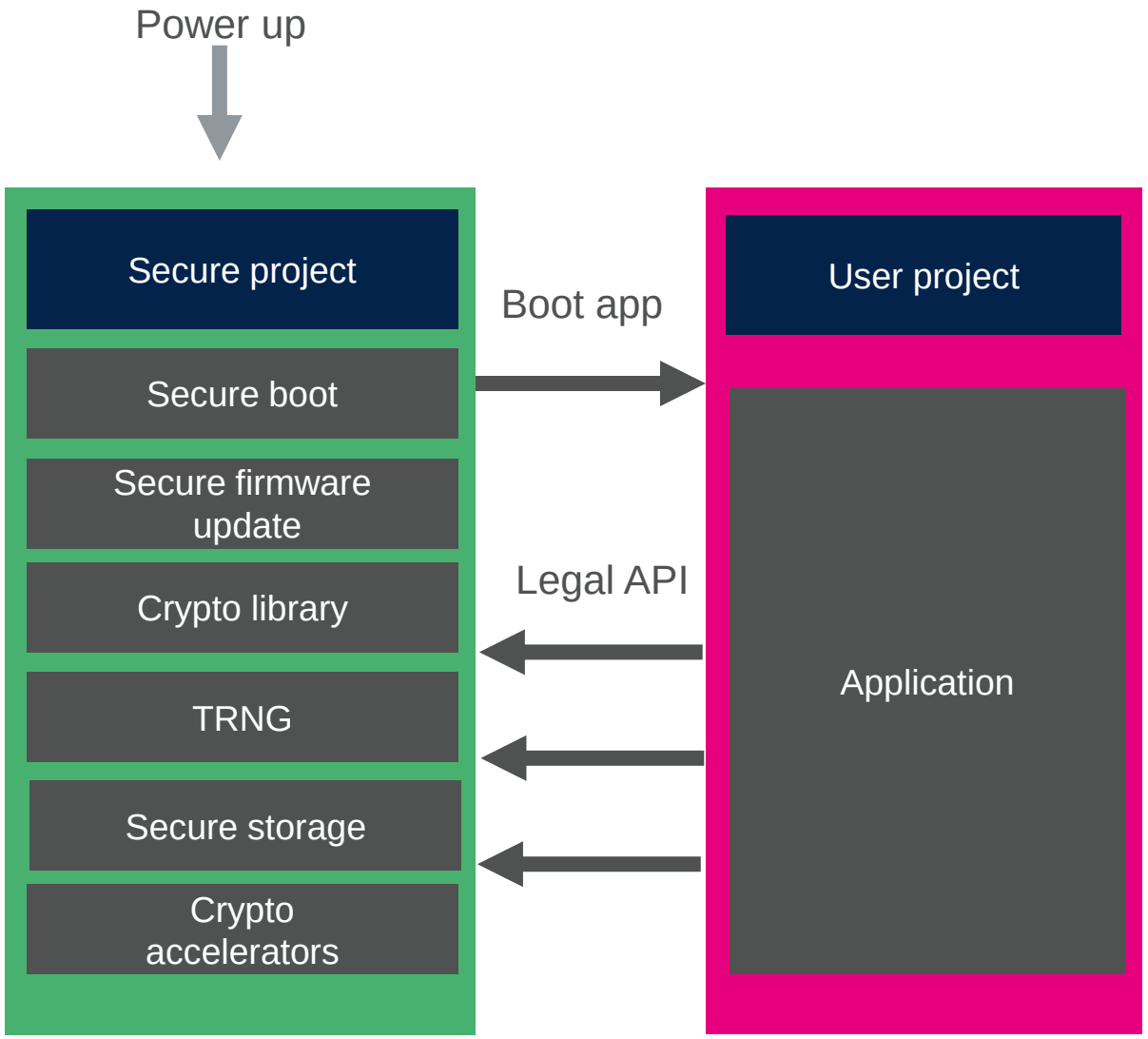
Horizontal alignment
Secure to the left



Vertical alignment
Secure on top

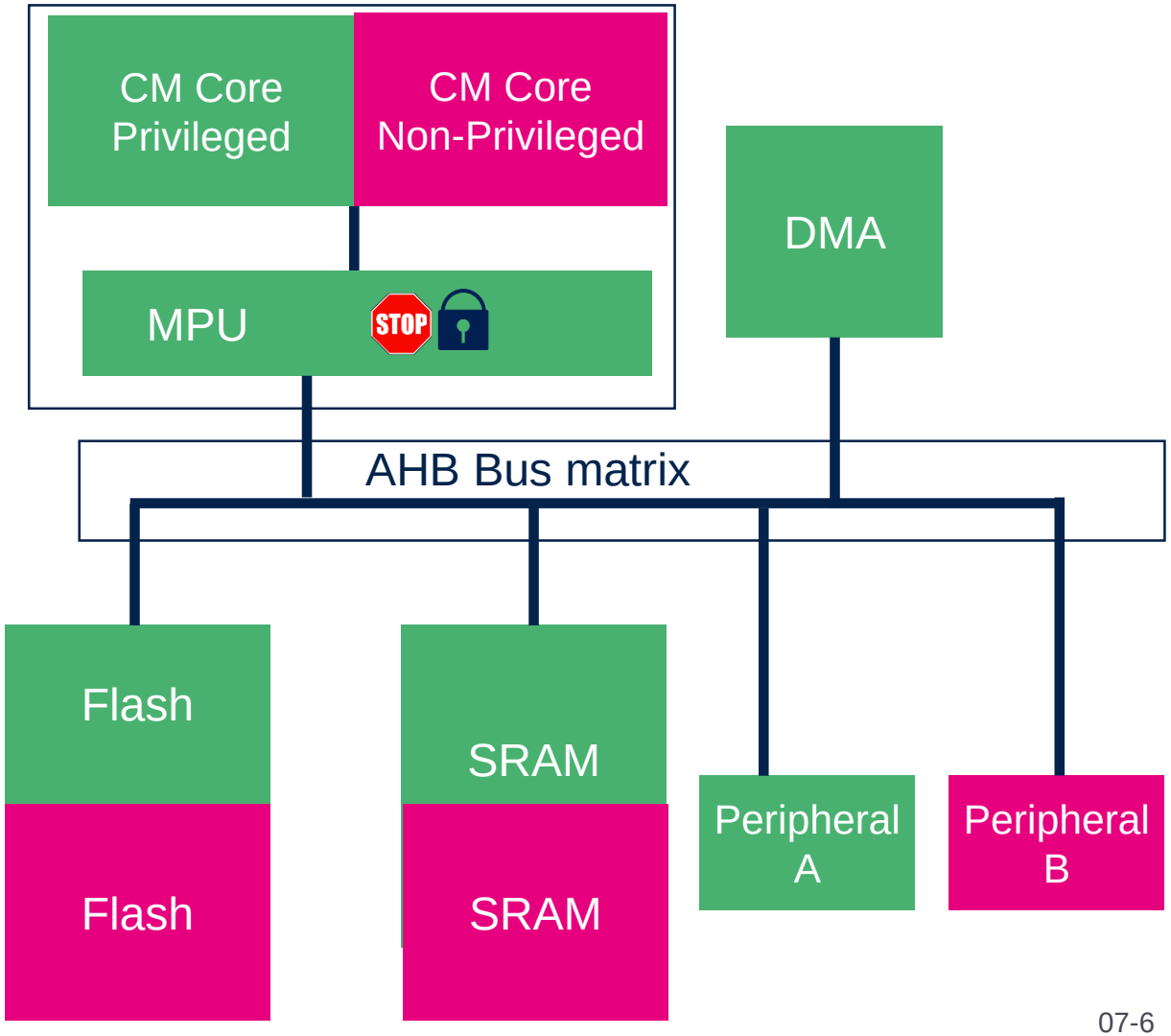
Isolate runtime environments

- Separate responsibilities
- Hardware based isolation
- Restrict access to code and data



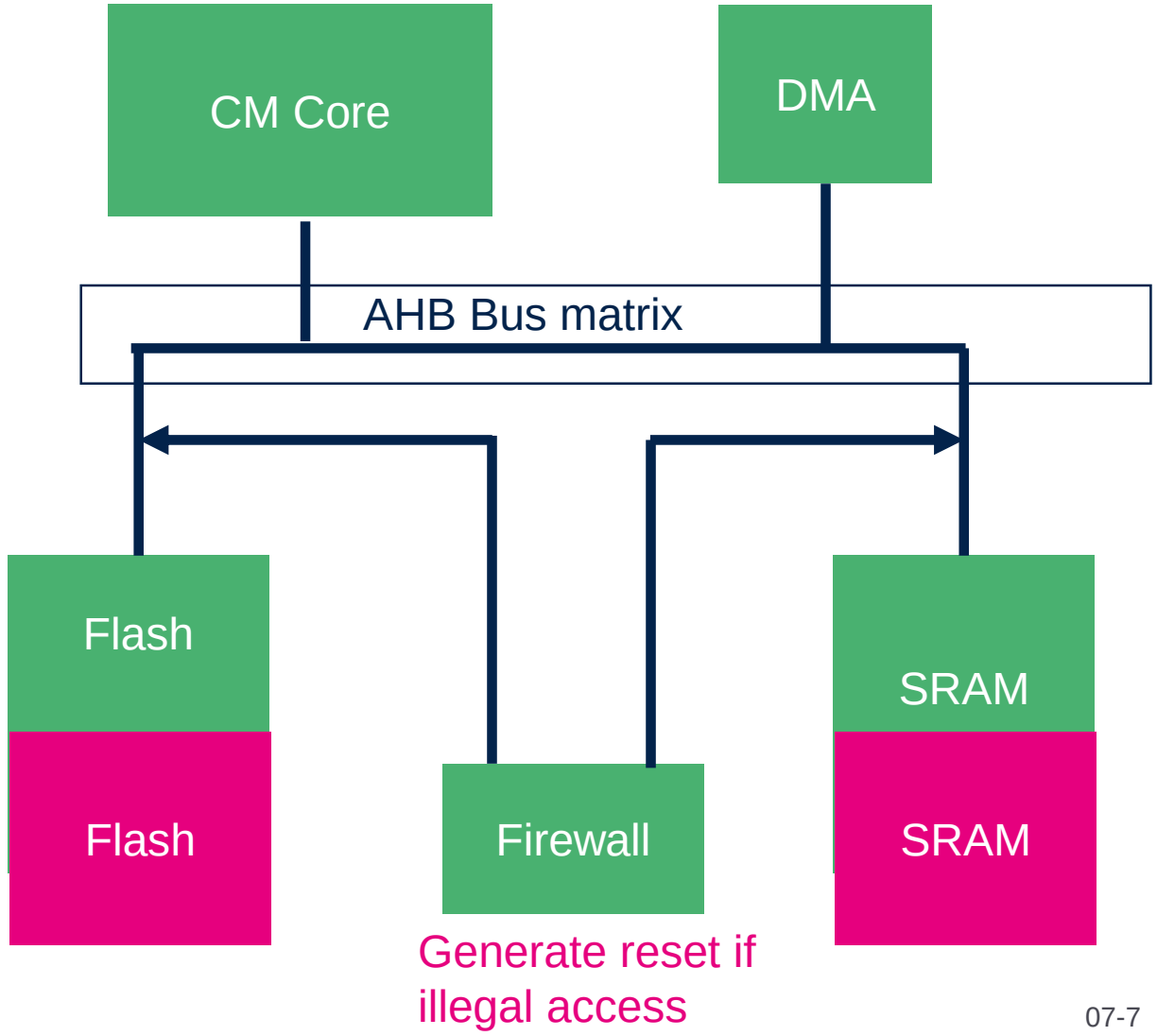
MPU based isolation

- MPU is mechanism which define access rights of the core for memory regions (SRAM, FLASH, Peripherals).
- Constraint :
  - As MPU can be configured only in privileged mode. Handler mode is always privileged:
    all interrupt service routines must be trusted
  - DMA has full access on the memory map. Secure project must restrict access to DMA itself by MPU



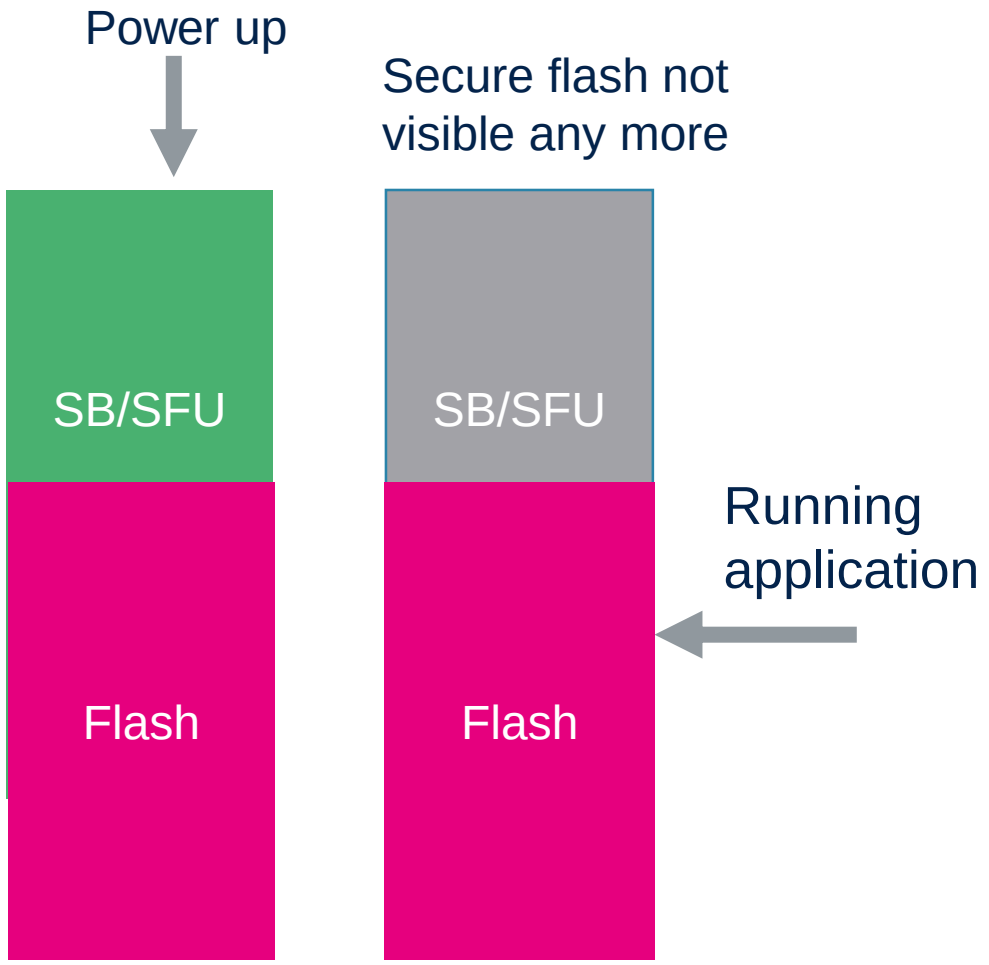
Firewall based isolation

- Hardware IP which snoops bus transaction to SRAM and Flash on the AHB bus matrix
- Configured by Secure project and once enabled it cannot be disabled until next reset
- Constraint :
  - Interrupts must be disabled befor calling secure service
  - Securing peripherals is not possible
- Rmk : Protects against DMA access



Secure flash based isolation

- Secure memory is visible at boot time, then can disappear from the rest of the system until the next reset.
- Constraint :
  - Interactions between Secure and Non-Secure are limited. Must go through reset
- Rmk : address Secure boot and Secure firmware update isolation requirement



Isolation Features by STM32 Series

STM32 Series	Isolation features				
	Secure mem/ HDP	MPU	Firewall	Trustzone	Arm Cortex®
STM32 **F0**					**M0**
STM32 **F1**		■			**M3**
STM32 **F2**		■			**M3**
STM32 **F3**		■			**M4**
STM32 **F4**		■			**M4**
STM32 **F7**		■			**M7**
STM32 **L0**		■	■		**M0+**
STM32 **L1**		■			**M3**
STM32 **L4**		■	■		**M4**
STM32 **L5**	■	■		■	**M33**
STM32 **H7**	▒	■			**M7/M4**
STM32 **G0**	▒	■			**M0+**
STM32 **G4**	▒	■			**M4**
STM32 **WB**		■			**M4/M0+**

■ Available on all devices

▒ Depends on device part number

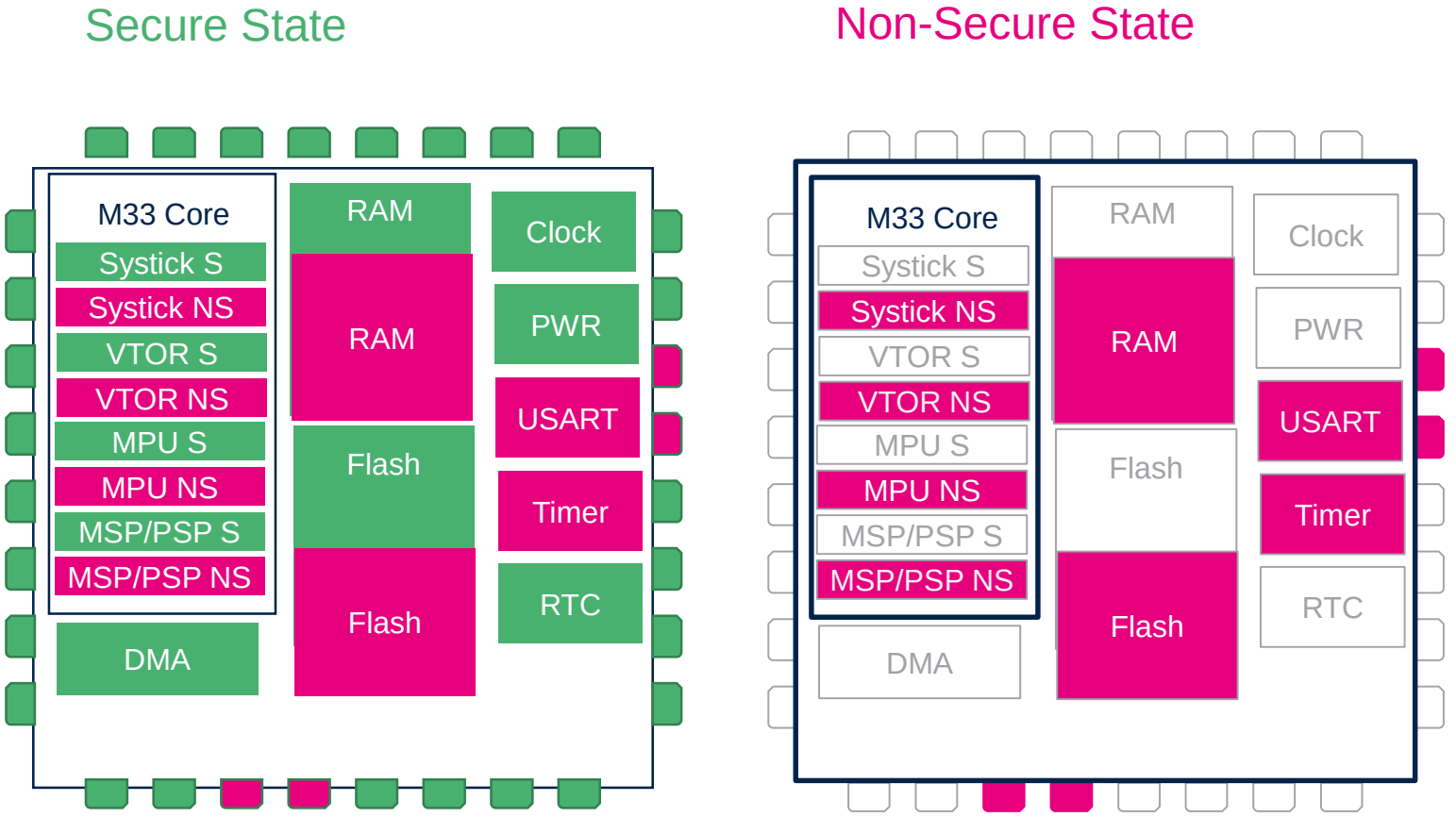
life.augmented

TrustZone introduction on STM32L5

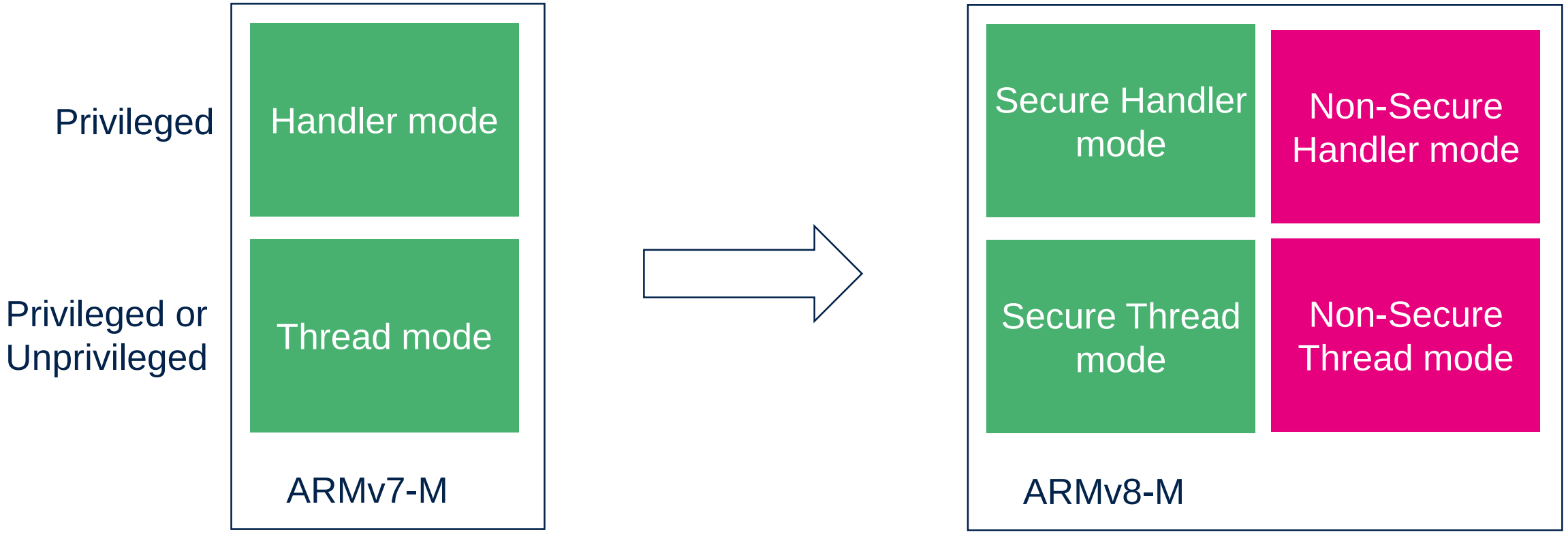
life.augmented

Cortex M33 and TrustZone Key features

- New processor state
- Real time
  - Low interrupt latency
  - Low state switching overhead
  - Deterministic

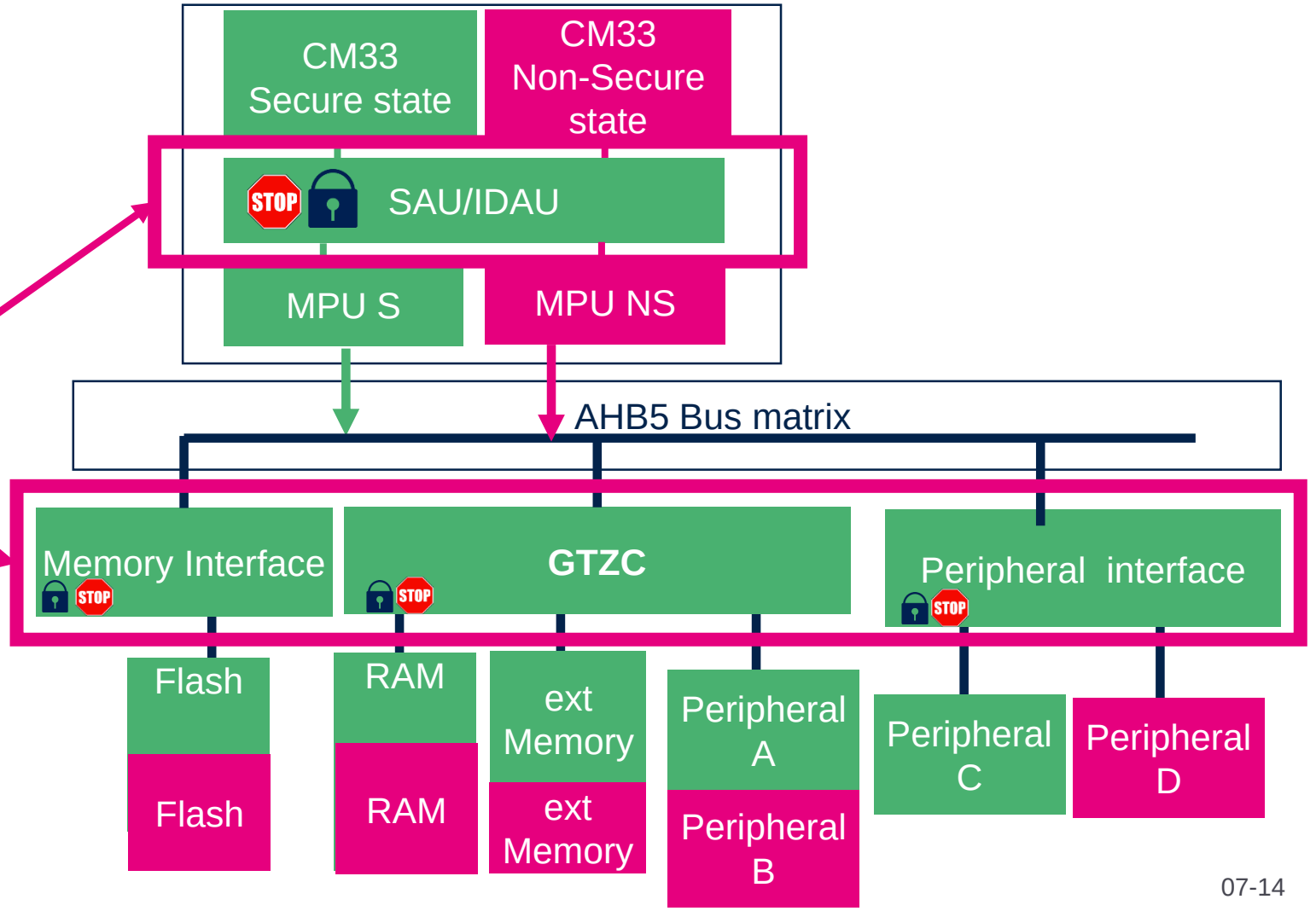


Security state in ARMv8-M



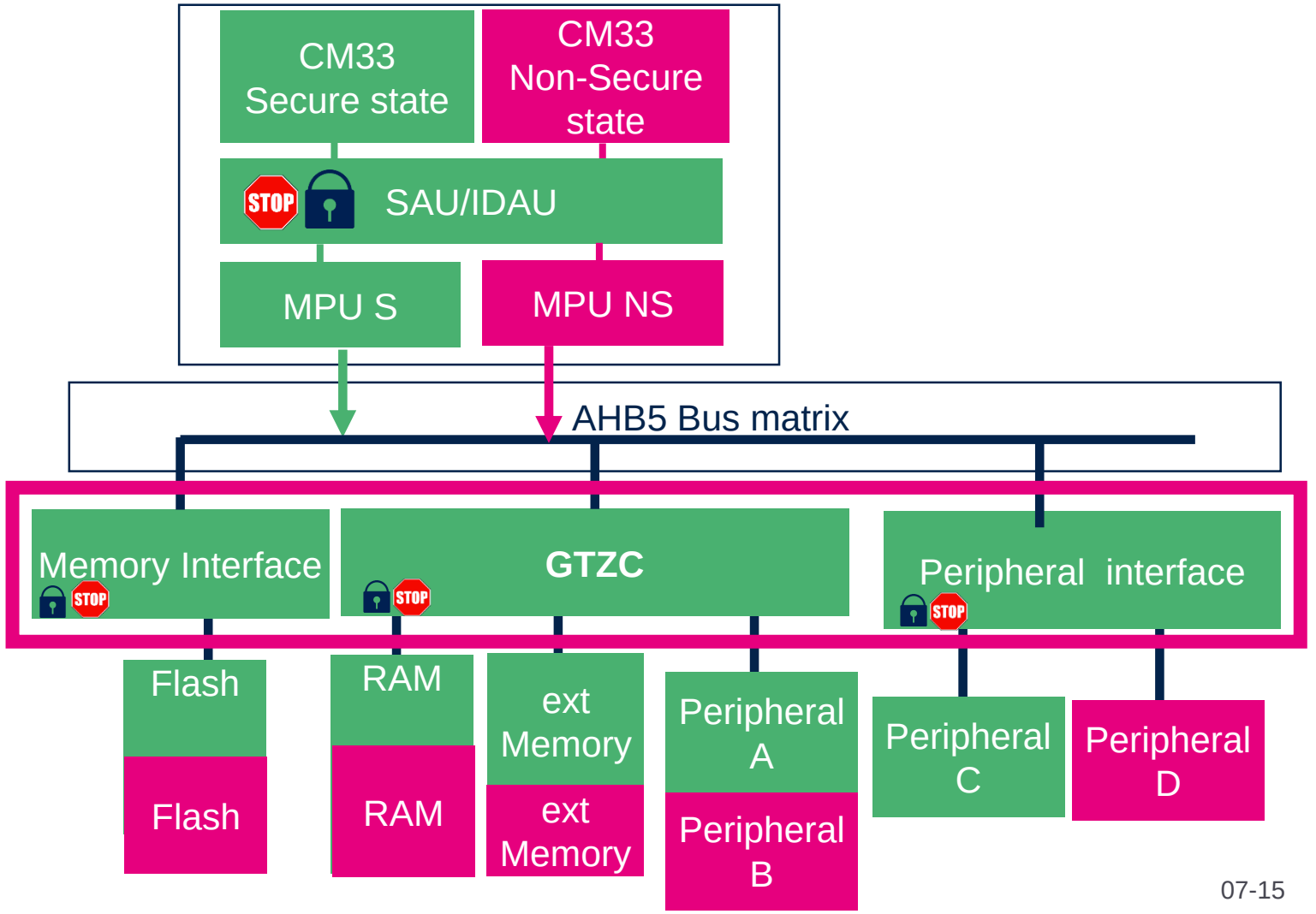
TrustZone based isolation

- Memory and peripheral access are controlled based on Core state and Peripheral/Memory security attribute
- 2 level of security filtering :
  - SAU/IDAU (attribution units)
  - Protection controllers
- Security attribute is fully configurable



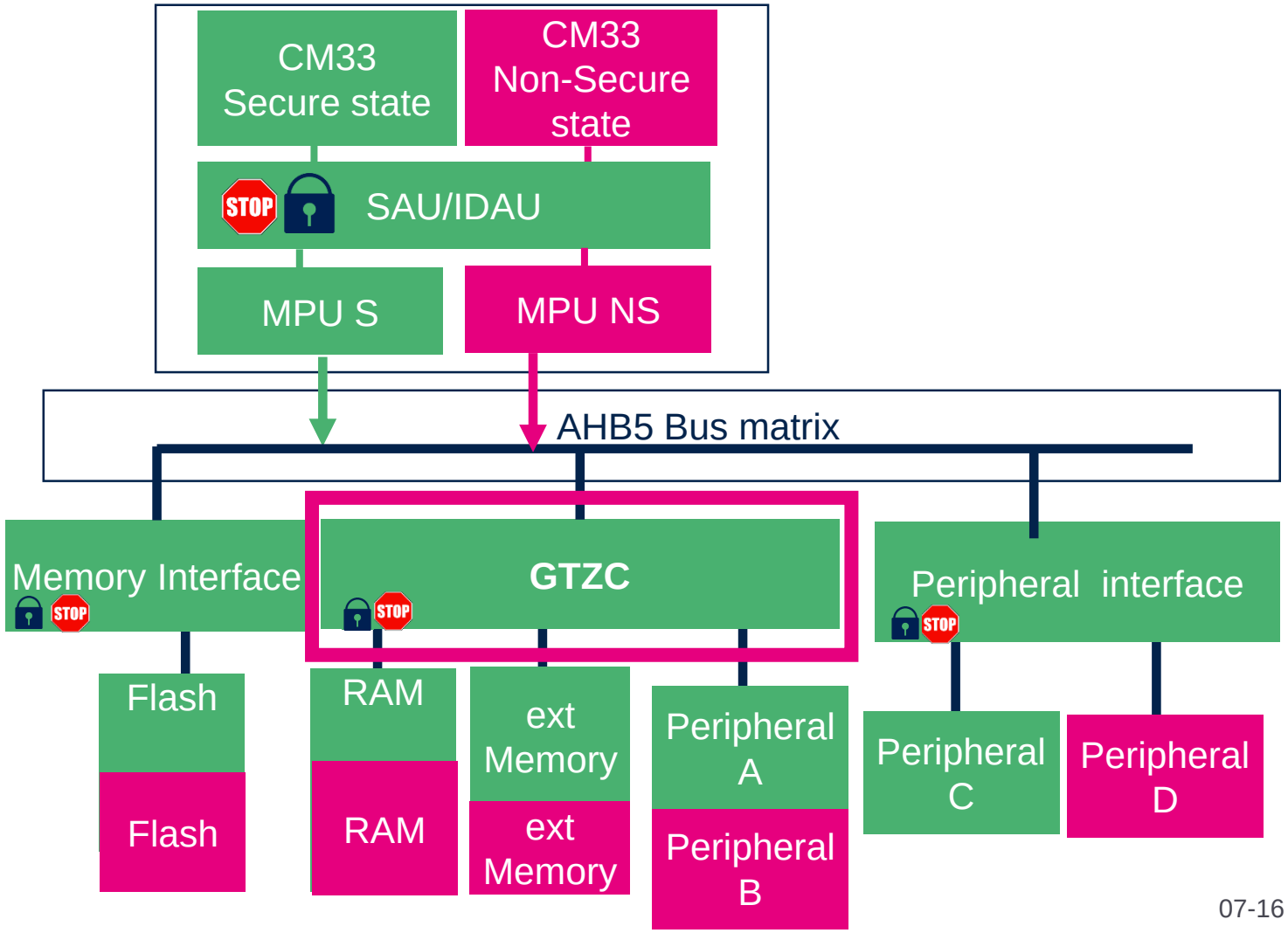
Protection controller based isolation

- Protection controllers are distributed on many levels
  - Inside the Flash memory interface
  - On the peripheral level TrustZone-aware peripherals
  - In GTZC (SRAM, ext Memory, Securable peripherals)



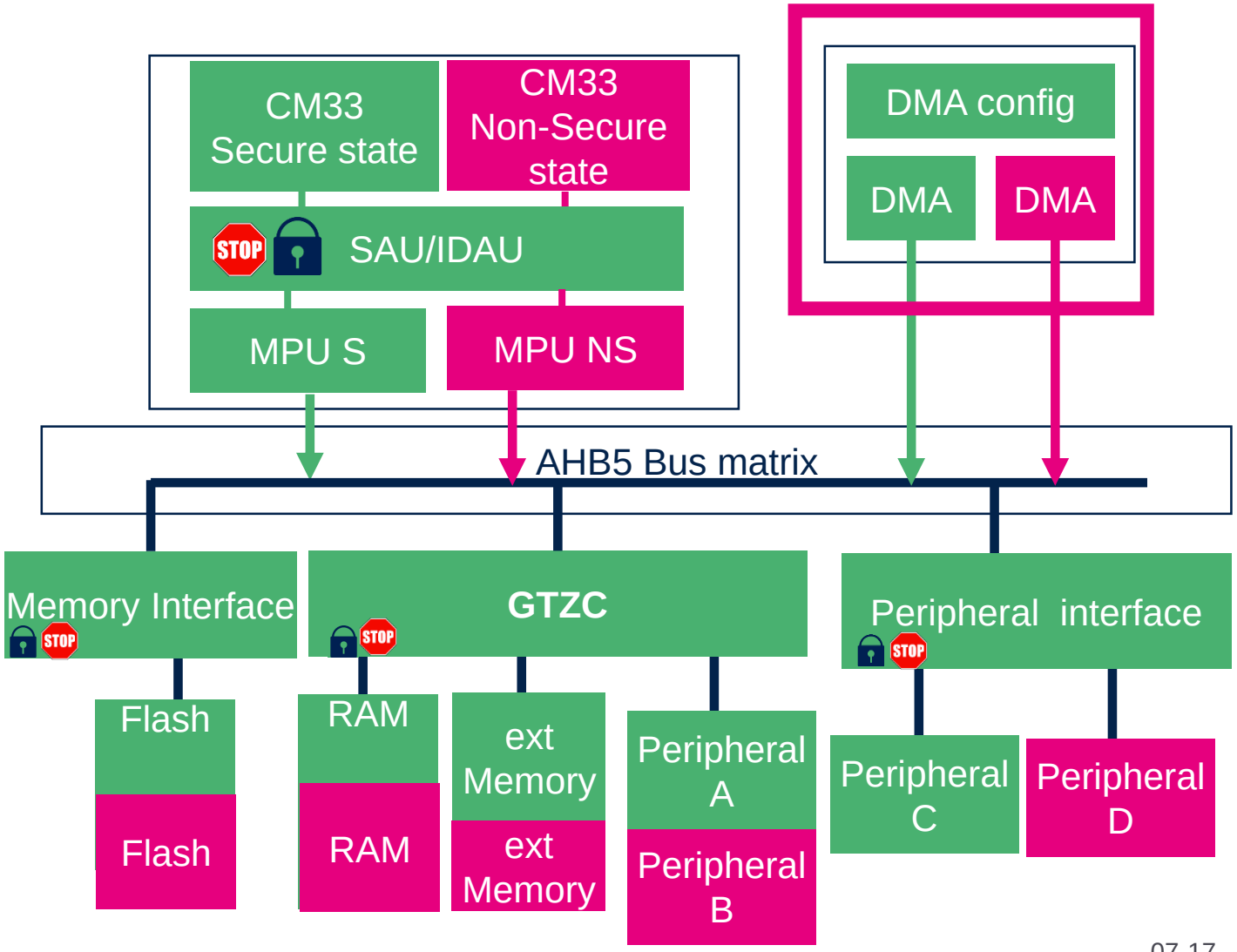
Global TrustZone security controller (GTZC)

- ST proprietary peripheral
- Configures secure areas of
  - SRAM
  - External memories
  - Securable peripherals
- Aggregate illegal access events and generates secure interrupt towards the NVIC



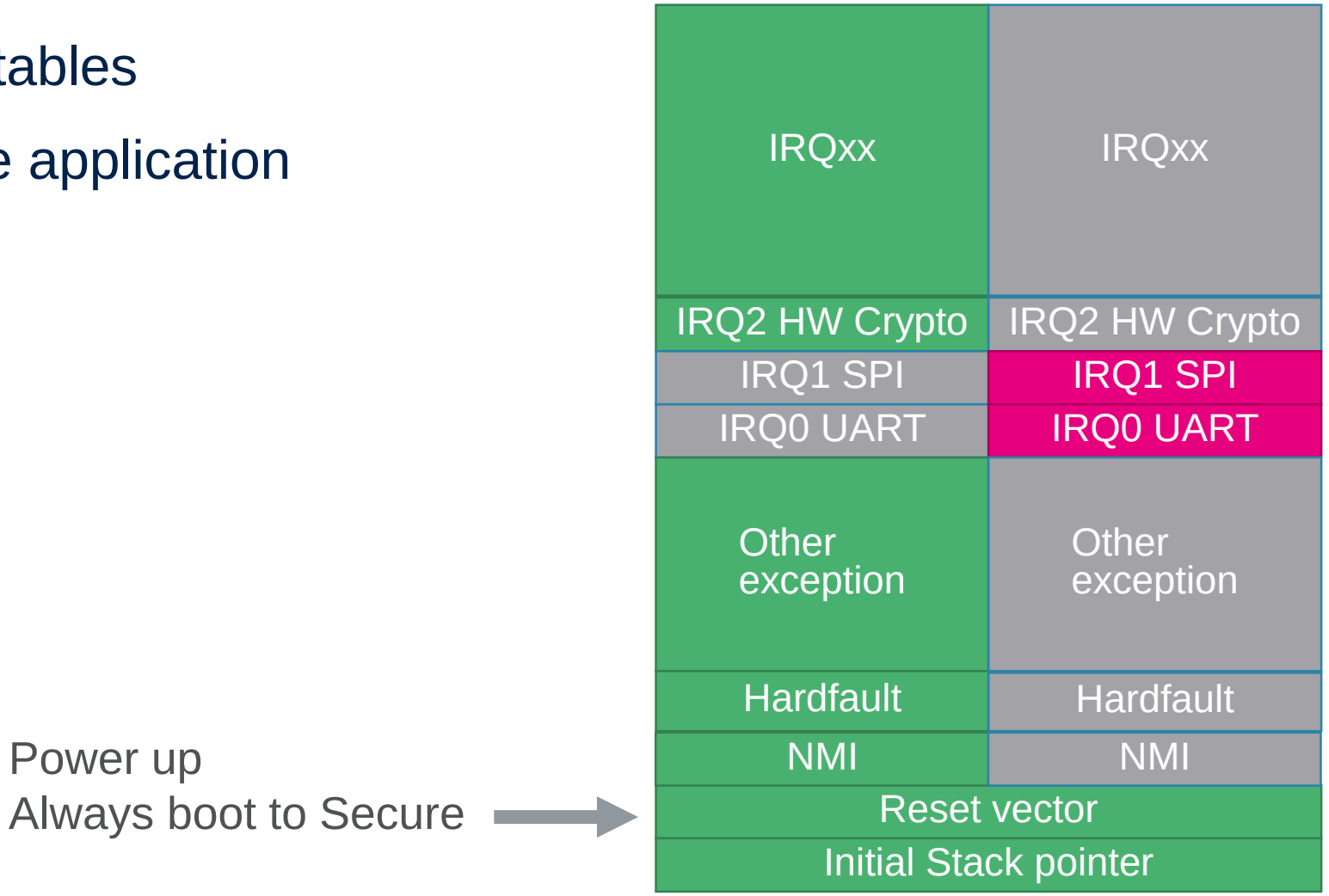
DMA

- DMA is AHB Master
- 2 x DMA each with 8 channels
- Support of secure/non-secure DMA transfers independently at
  - channel level
  - source and destination address level
- DMA is
  also TrusZone aware peripheral.
  Security configuration can be only done by CM33 in Secure state



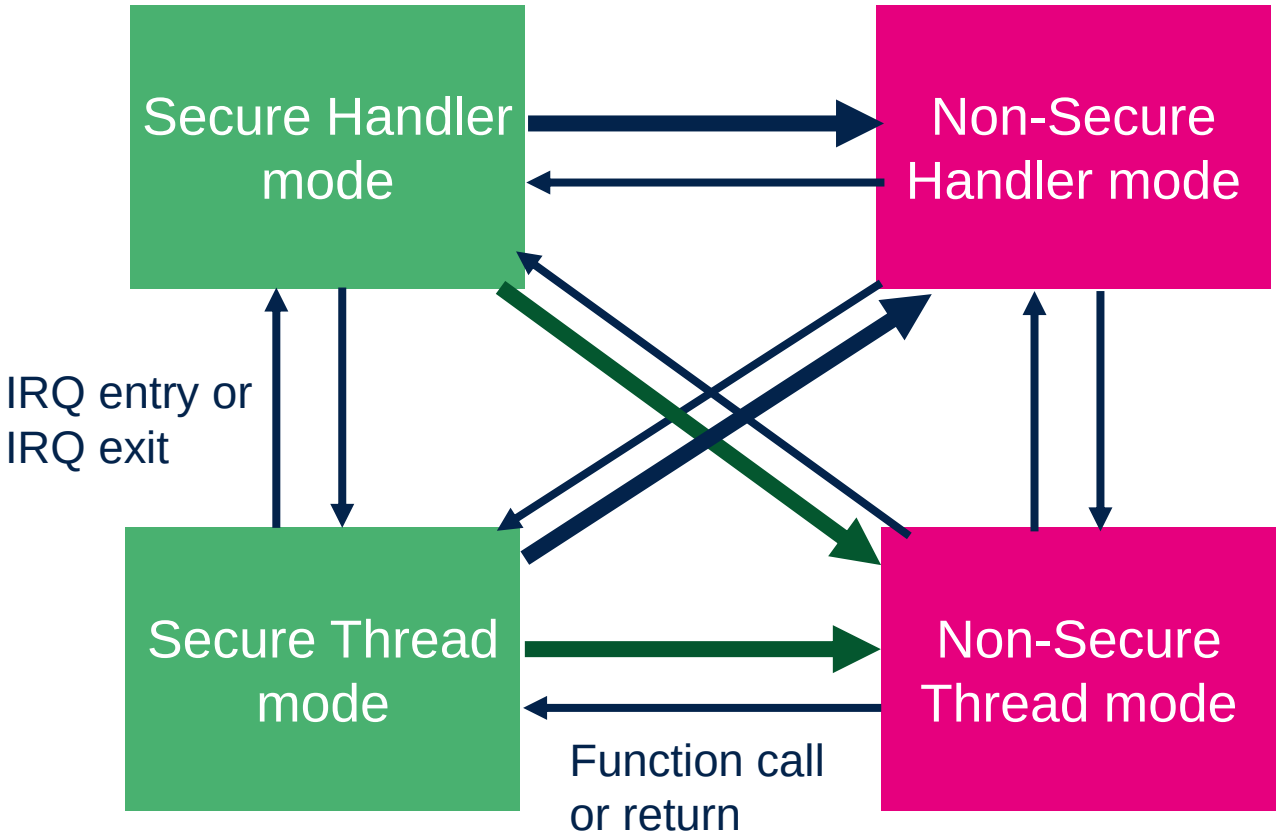
Interrupt assignment

- Two separate vector tables
- Configured by Secure application
- Flexible assignment
- Priority management



Security state transitions

- Transition possible at any time
- Interrupt latency increased when switching from
  Secure to Non-Secure Handler.
  Context safe and register clean up done by HW (21 cycles, 12 cycles otherwise)
- Same for transition from
  Secure to Non-Secure Thread
  but here compiler adds instruction to do the clean up

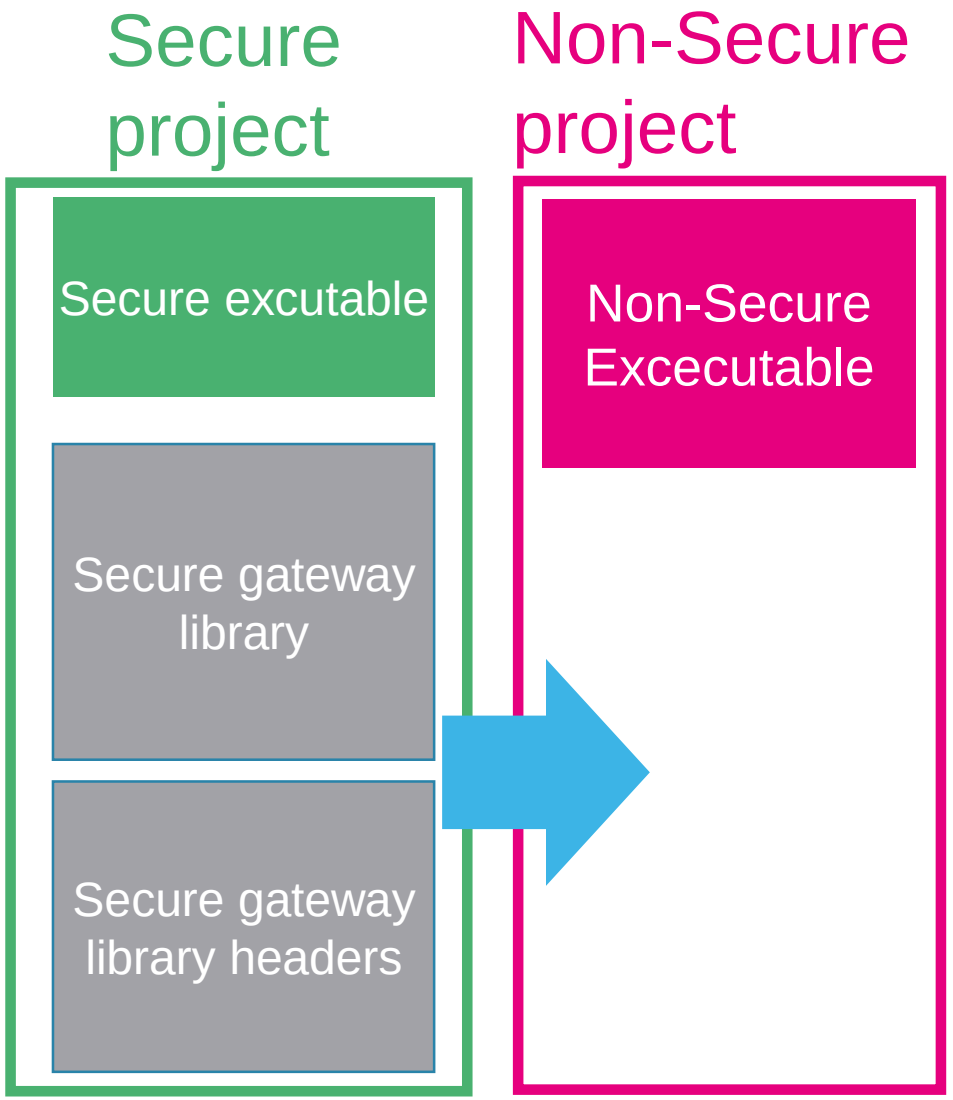


TrustZone
Development flow

life.augmented

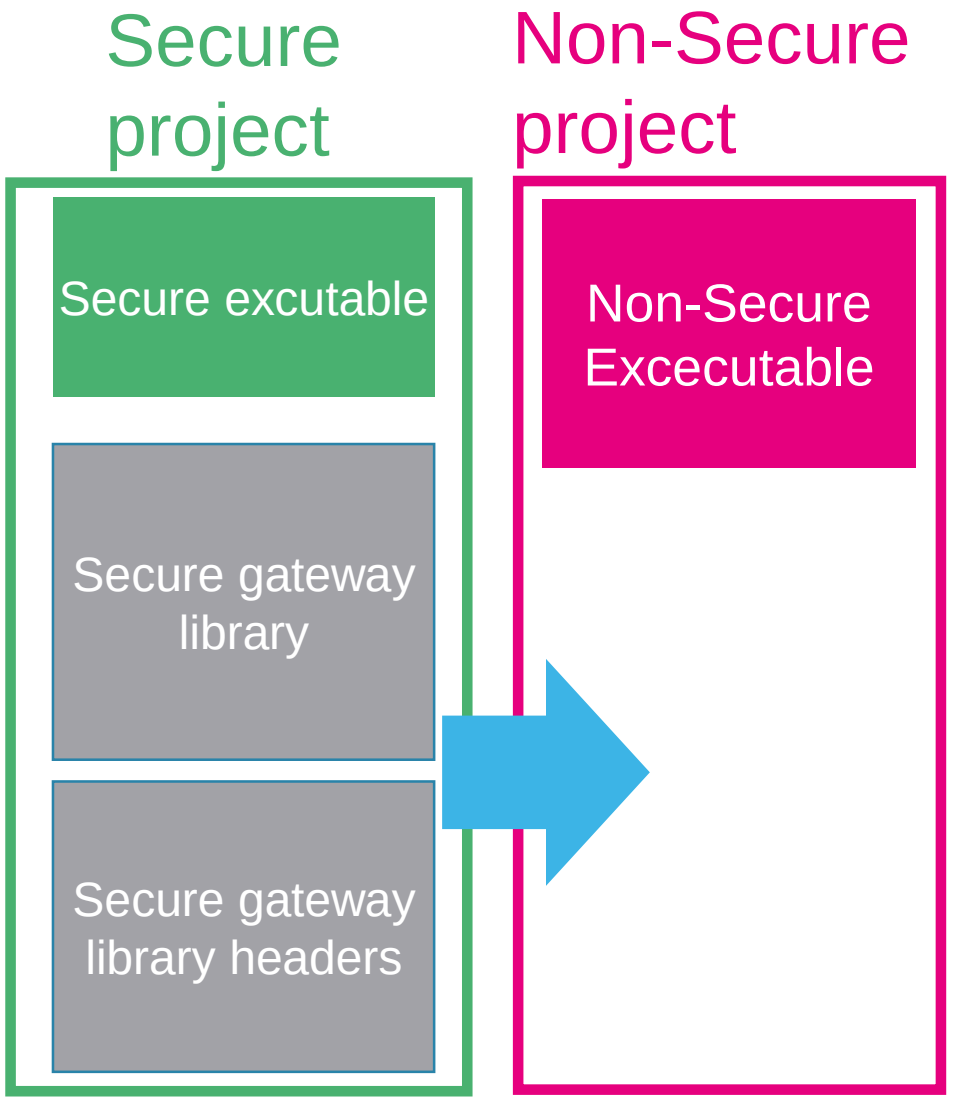
Development flow

- Independent development of Secure and Non-Secure
- Secure project exports legal API via Secure gateway library
- Non-Secure project is linked against Secure gateway library
- Rmk: Build order



Toolchain requirements

- Secure project does require
  - Toolchain support
  - C Language extension
  - Extra instructions for state switch
  - Reason: Prevent information leakage or Secure state exception
- Non-Secure project does NOT require any toolchain support for TrustZone

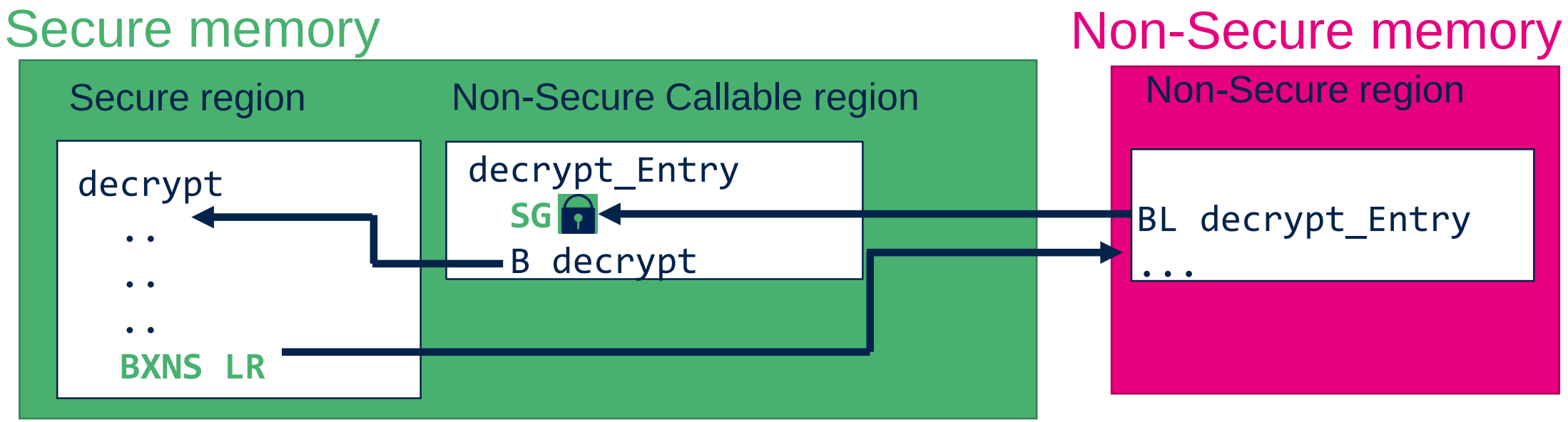


TrustZone
Interactions between the Secure and Non-Secure application

life.augmented

Calling Secure function from Non-Secure world

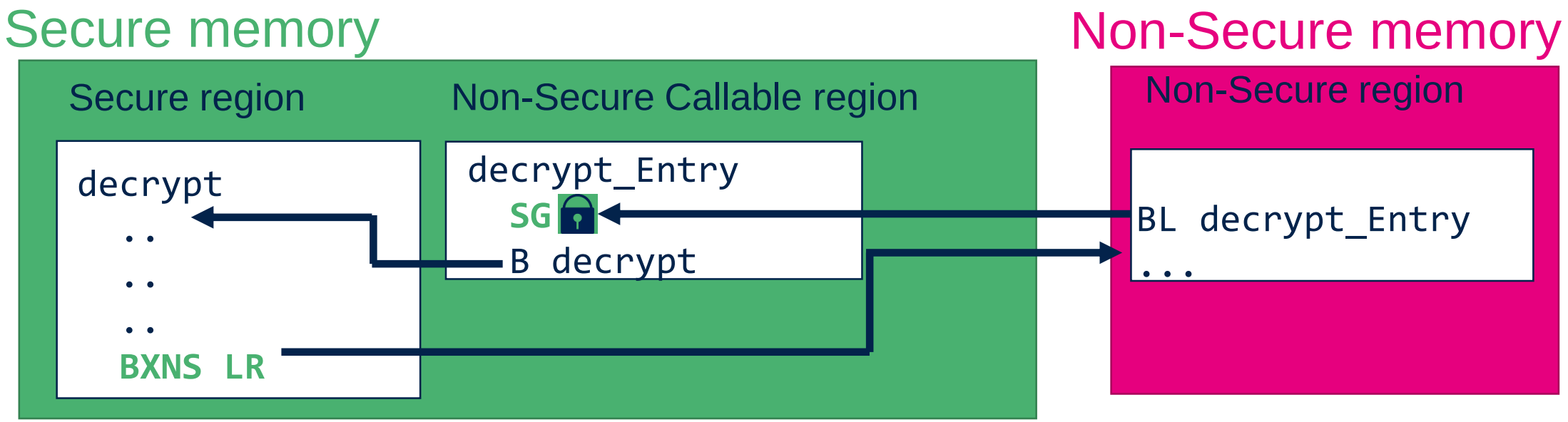
- New instruction for the Secure state
  - SG: Secure gate
  - BXNS: Branch to Non-Secure



ARM C Language Extensions (ACLE) for ARMv8-M

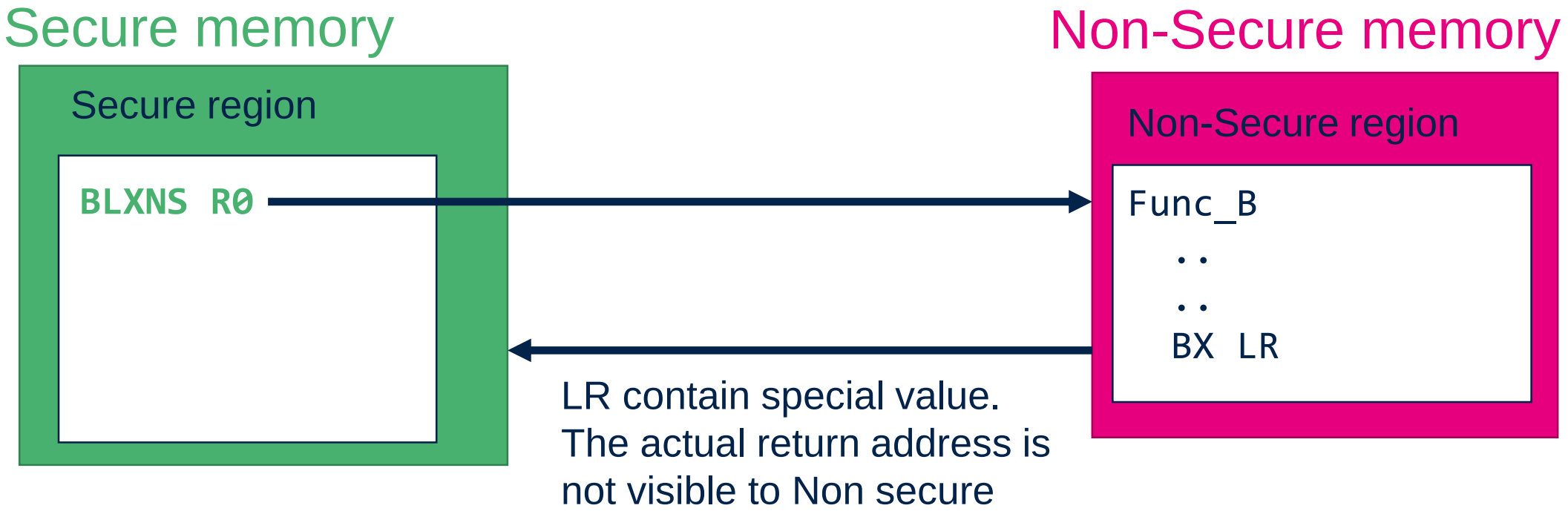
- Set of intrinsic functions and predefined macros.
- Ex1: Creating function callable from Non-Secure aka legal API

```
CMSE_NS_ENTRY void decrypt_Entry(void){}
```



Calling Non-Secure function from Secure world

- New instruction for the Secure state
  - BLXNS: Branch with link to Non-Secure



ARM C Language Extensions (ACLE) for ARMv8-M

- Ex2: Calling NonSecure function

```
funcptr_NS callback_NS; // non-secure callback function pointer
callback_NS = (funcptr_NS)cmse_nsfptr_create(Func_B);
callback_NS();
```

- `Func_B` is often initialized at runtime. Why?

life.augmented

Conclusion

life.augmented

Conclusion

- Isolation means on STM32 families
  - MPU
  - Firewall
  - Secure Memory
  - TrustZone

- TrustZone integration on STM32L5
- Development flow. CMSIS support of TrustZone

ARM Resources

Intro level

- Software Developers' Guide to IoT Security (White Paper)

Functional overview

- TrustZone ® technology for ARM ® v8-M Architecture
- System Design with ARMv8-M
- Enhanced Security and Energy Efficiency of Microcontrollers and SoCs (ARM, EW2016 paper)

Software oriented

- Secure software guidelines for ARMv8-M
- Using TrustZone on Armv8-M
- ACLE Extensions for ARMv8-M

ARM Specifications

- ARM ® v8-M Architecture, Reference Manual
- Arm ® Cortex ® -M33 Devices, Generic User Guide
- ARM ® Cortex ® -M33 Processor Technical Reference Manual
- ARM ® AMBA ® 5 AHB Protocol Specification

STM Resources

- AN5347 STM32L5 Series TrustZone® features (STM)
- AN5156 Introduction to STM32 microcontrollers security (STM)
- RM0438 Reference manual STM32L5x (STM)

Thank you

life.augmented