
Hello, and welcome to this presentation that details the TF-M offer in STM32U5

## STM32U5 TFM overview

- Modular/configurable from basic SBSFU example to full TFM example
- Provides a wide set of security concepts:
  - Single-entry point at reset: force code execution to start with Secure Boot code
  - TFM_SBSFU_Boot code and “secrets” immutable: no possibility to modify or alter them
  - Three protected/isolated domains:
    - Secure & privileged to execute PSA immutable Root of Trust (RoT)
    - Secure & privileged to execute PSA updatable RoT
    - Secure & unprivileged to execute application updatable RoT
  - Limit execution surface according to the application state:
    - TFM-SBSFU code: from Reset until installed application
    - Application secure/non-secure code: once the installed application has been verified
  - Remove JTAG access to the device
- Protection against software and physical attacks

life.augmented

2

Cryptography ensures integrity, authentication and confidentiality.
However, the use of cryptography alone is not enough: a set of measures and a system-level strategy are needed to protect critical operations, sensitive data (such as a secret key), and the execution flow, in order to resist possible attacks.
The Secure Boot and Secure Firmware Update (or SBSFU) solution based on Trusted Firmware for Cortex-M (or TF-M) provides a modular and configurable framework whose security concepts are described hereafter.
Three protected and isolated domains are created:

- Secure / privileged: to execute PSA immutable RoT code using its associated secrets and to use secure privileged STM32U5 peripherals. This domain is hidden once the execution of immutable PSA RoT code is complete.

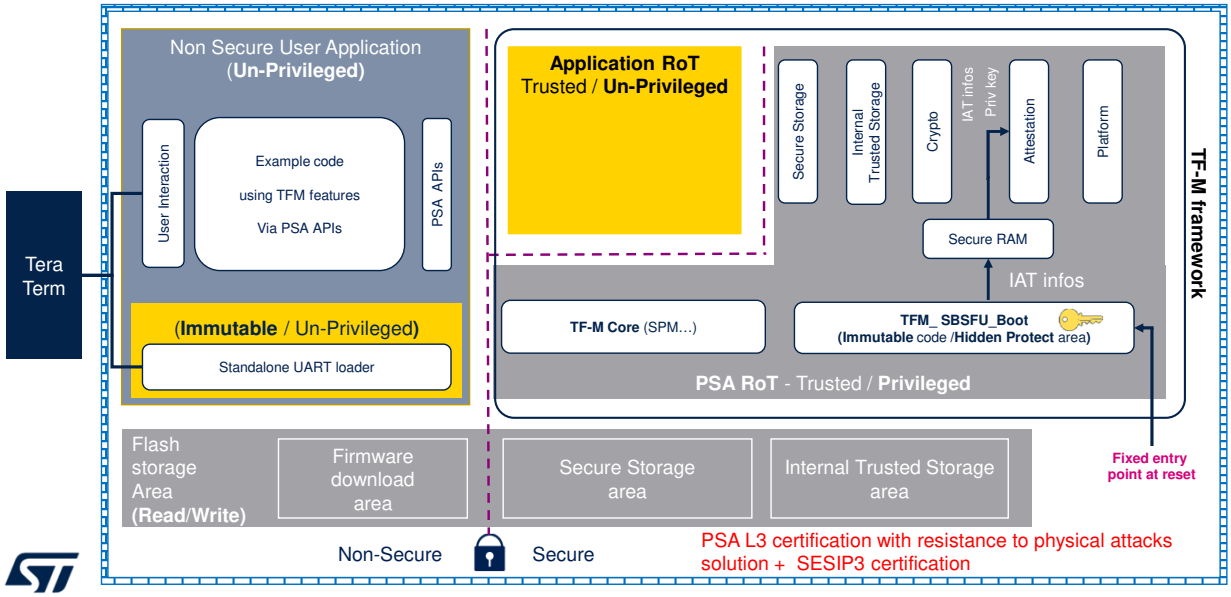
- Secure / privileged: to execute PSA updatable RoT code using its associated secrets and to use secure privileged STM32U5 peripherals.
- Secure / unprivileged: to execute application updatable RoT and its associated secrets and to use secure unprivileged STM32U5 peripherals.

The Execution surface is limited according to the application state:

- From product reset until the installed application is verified: only TFM_SBSFU_Boot code execution is allowed
- Once the installed application has been verified: application code (secure part and non-secure part) execution is allowed.

The STM32U5 also features protection against software and physical attacks.

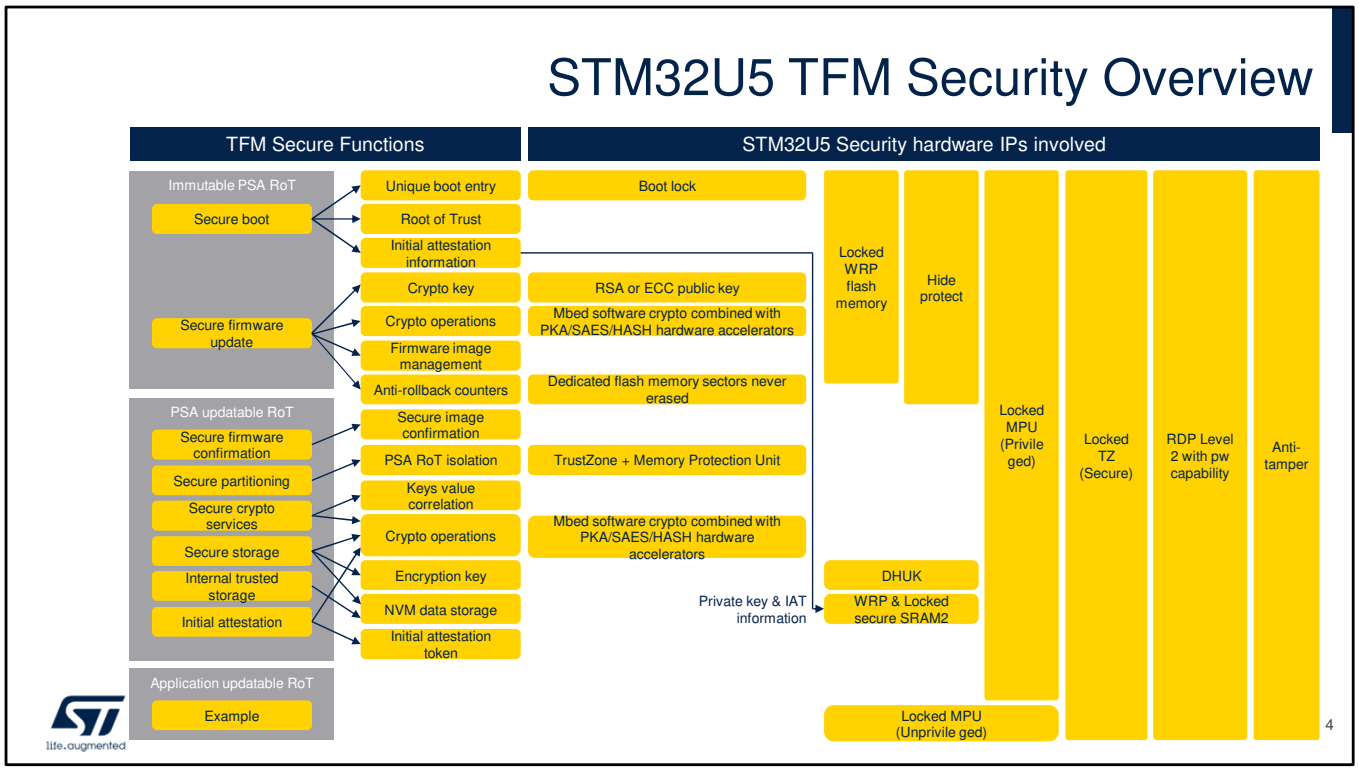
## STM32U5 TFM top level view




TFM Implementation in the STM32U5Cube firmware is based on the ARM TF-M reference implementation.
This figure presented by this top-level view summarizes all the TFM components described in the previous section.
The STM32CubeU5 package proposes 2 applications

- The TFM: application with full TFM services
- The SBSFU: application with only the Secure Boot and Secure Firmware Update services of the ARM TF-M.

In this figure the Tera Term HyperTerminal is used to interface with a toolset to configure the example, run it and display the execution results.
Examples and help is available in the UM2851 user's manual entitled: Getting started with STM32CubeU5 TFM application.




This Figure details the secure functions of TFM, and the hardware security IPs integrated into the STM32U5 devices to reinforce the protection mechanisms against outer and inner attacks.

TF-M is an open-source software framework driven by Arm Limited that provides a reference implementation of the PSA standard on the Arm® Cortex®-M33 processor:

- The PSA immutable RoT (Root of Trust) is an immutable *"Secure Boot and Secure Firmware Update"* application executed after any reset.
- The PSA updatable RoT is a *"secure"* application implementing a set of secure services isolated in the secure/privileged environment that can be called by the non-secure application at non-secure application run-time via the following PSA APIs
  - Secure storage service

    - Internal trusted storage service
    - Cryptography service
    - Initial attestation service
- The application updatable RoT are third-party secure services that are isolated in the secure/unprivileged environment and that can be called by the non-secure application at non-secure application run-time.

The right-hand side of the figure details the security hardware IPs involved in the various secure functions.

# Protection against outer attacks

- Outer attacks definition = attacks triggered by external tools such as debuggers or probes
- STM32U5 TFM SBSFU example implements 4 protection mechanisms (in yellow):

| Device lifecycle | Boot lock | Protected SRAM2 |
|---|---|---|
| • 4 RDP levels<br>• RDP2 highest protection<br>• No JTAG access to the device at RDP2<br>• Possible regression with OEM password if provisioned | • **BOOT_LOCK** is a device user option byte (OB)<br>• Fixed entry point to a memory location defined by OB<br>• In TFM: boot Entry point after reset is fixed to **TFM_SBSFU_Boot code** | • Against intrusion at RDP1<br>• Erased when an intrusion is detected<br>• SRAM2 WRP: if enabled the content is frozen until next the reset<br>• In TFM: used to share and freeze initial attestation information between TFM_SBSFU_Boot and the secure application |

| Anti-tamper | DAP disable + IWDG |
|---|---|
| • Used to protect sensitive data from physical attacks<br>• Activated at the start of TFM_SBSFU_Boot<br>• Remains active during TFM_Appli and TFM_Loader applications<br>• If Tamper detected: SRAM2, caches and cryptography peripherals are erased, then a reboot is forced<br>• Tampers can be internal events or external pins | Security features available in STM32U5 devices but not used by TFM:<br>• DAP deactivation<br>• IWDG to control the boot duration |

life.augmented

5

This slide describes the mechanisms used to protect against outer attacks triggered by tools such as debuggers and probes.
The Device lifecycle feature is based on Read protection level 2 to achieve the highest protection level.
Read protection level 2 with OEM2 password capability is used to ensure that the JTAG debugger cannot access the device, except to inject the OEM2 password.
In RDP level 2, when OEM2 password is injected on the JTAG port, the RDP level is regressed to level 1.
The OEM2 password must first have been provisioned when the RDP level is 0.
The Boot lock feature is based on the BOOT_LOCK option byte, used to fix the entry point to a memory location defined by the Option byte.
In the TFM application example, the boot Entry point after

reset is fixed to TFM_SBSFU_Boot code.
SRAM2 is automatically protected against intrusion once the system is configured in RDP level 1.
The SRAM2 content is erased as soon as an intrusion is detected. Moreover, SRAM2 content can be write protected until the next reset by activating a lock bit.
In the TFM application example, the system has been configured to use the protected SRAM2 to share and freeze the initial attestation information between the TFM_SBSFU_Boot application and the secure application.
The anti-tamper protection is used to protect sensitive data from physical attacks. It is activated at the start of TFM_SBSFU_Boot and remains active during TFM_Appli and TFM_Loader applications.
If tampering is detected, sensitive data in SRAM2, caches and cryptographic peripherals are immediately erased, and a reboot is forced.
Both external active tamper pins and internal tamper events are used.
Other STM32U5 peripherals could be used to protect product against outer attacks, but the current TFM example does not use them:

- The debug protection consists in disabling the Debug Access Port. Once disabled, the JTAG pins are no longer connected to the STM32U5 internal bus. DAP is automatically disabled with RDP level 2.
- Independent Watchdog (IWDG) is a free-running down-counter. Once running, it cannot be stopped. It must be serviced periodically, otherwise it causes a reset. This mechanism could be used to control the TFM_SBSFU_Boot execution duration.

## Protection against inner attacks

- Inner attacks definition = attacks triggered by code running in the STM32
  - Attacks may be due to:
    - Malicious firmware exploiting bugs or security breaches
    - unwanted operations

- Hardware protections against inner attacks used by TFM:
  - **TZ** (TrustZone®)
  - **MPU** (memory protection unit)
  - **SAU** (security attribution unit)
  - **GTZC** (global TrustZone® controller)
  - **WRP** (write protect)
  - **HDP** (hide protection)

life.augmented

6

Inner attacks refer to attacks triggered by code running in the STM32.
Attacks may be due to either malicious firmware exploiting bugs or security breaches, or unwanted operations.
TF-M provides the following protections against inner attacks:

- ARM Trustzone enables two execution environments: secure and non-secure with a strict isolation between them.
- The MPU is used to make an embedded system more robust by splitting the memory map for Flash and SRAMs into regions with their own privileged access permissions.
- The SAU assigns security attributes to address ranges
- The GTZC is a firewall that checks the secure and privileged attributes of transactions targeting peripherals and memories.

- Write protection is used to protect trusted code from external attacks or even internal modifications such as unwanted write/erase operations on critical code/data.
- The code executed in this HDP area, with its related associated data and keys, can be hidden after boot until the next system reset.

# STM32CubeU5 TFM configurability

- The STM32Cube U5 MCU package proposes two different examples of applications:
  - TFM: application with full TF-M services
  - SBSFU: application with only the Secure Boot and Secure Firmware Update services of the TF-M

| Feature | Full TFM_SBSFU_Boot |
|---|---|
| Crypto schemes | RSA 2048, RSA 3072, EC 256 |
| Image encryption | AES-CTR, None |
| Cryptography modes | Software, Mix hardware/software,<br>DPA Hardware crypto SAES+PKA, HUK direct connection to SAES |
| Slot modes | Primary only slot (Active image overwritten)<br>Primary and secondary slots (Enabling OTA FW Update UC) |
| Images number modes | 1 image (sec+nsec), 2 images (separate sec + nsec images) |
| Flash memory configuration | Internal Flash memory + external flash capability |
| Image upgrade strategy | Overwrite only, Swap |
| Local loader | Ymodem, None |
| Anti-tamper | None, Internal tampers only, Internal and external tampers |

life.augmented

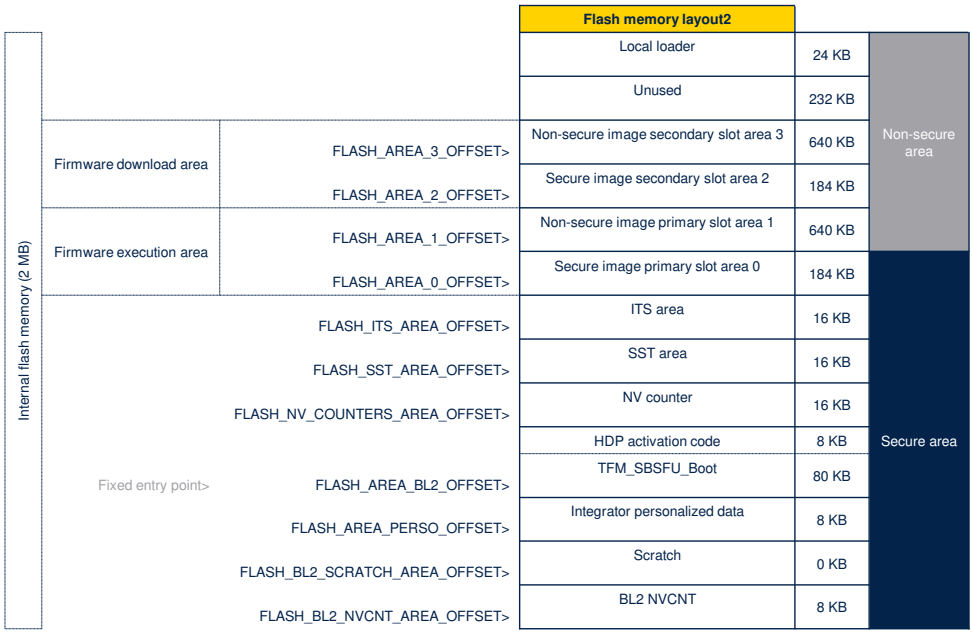
7

Two different examples are provided in the STM32Cube U5 MCU package:

- The TFM application is a complete implementation of [TF-M]
- A second application implementing only the Secure Boot and Secure Firmware Update functionalities of TF-M, named STM32CubeU5 SBSFU, is also available.

The table indicates the main features of the Secure Boot and Secure Firmware Update application.

## TFM example Flash memory layout (default configuration)




The STM32CubeU5 TFM application relies on a Flash memory layout defining different regions.
The Flash memory layout depends on the slot mode, the number of images, the image upgrade strategy and the local loader activation.
The default configuration of these features in the TFM application is the following:

- Slot mode: primary and secondary slots
- Image number mode: two images
- Image upgrade strategy: overwrite only mode
- Local loader: Ymodem.

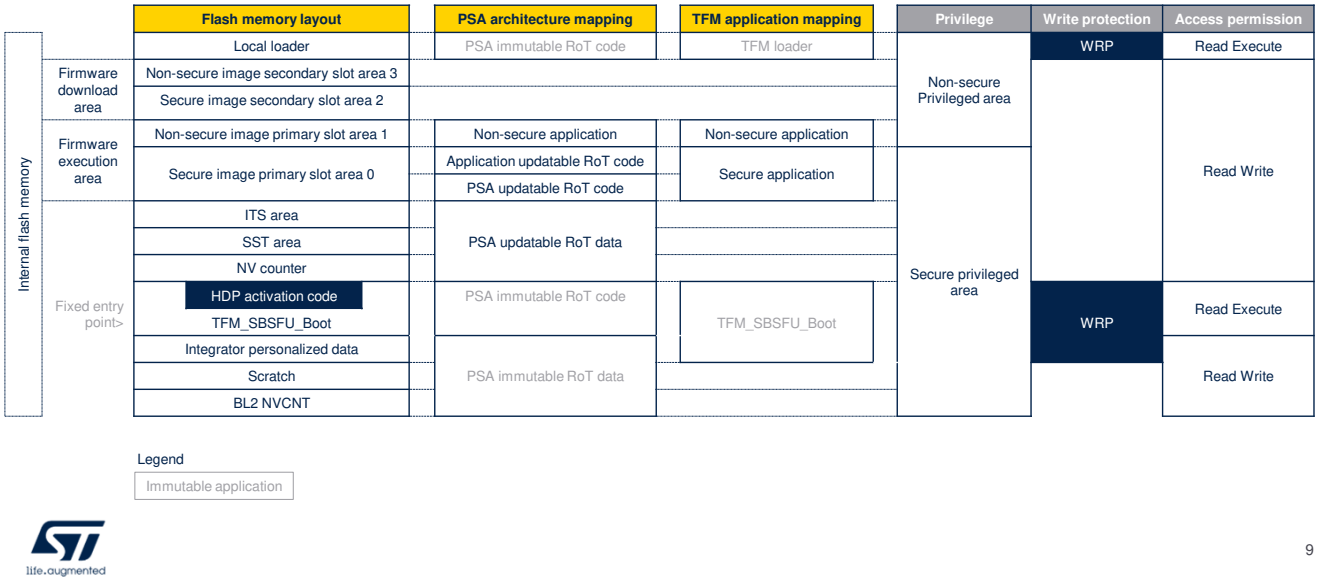
Each region has a specific usage:

- **BL2 NVCNT region**: to get non-volatile information about the lastest installed (Sec/Nsec) images versions.
- **SCRATCH region**: used by TFM_SBSFU_Boot to temporarily store image data during the image swap

process.

- **Integrator personalized data** : to personalize integrator-specific or STM32U5-specific TF-M data.
- **TFM_SBSFU_Boot binary** : to program the TFM_SBSFU_Boot code binary.
- **NV COUNTER** : where secure application manages the non-volatile counters used by the SST services.
- **SST area**: region where the encrypted data of the secure storage service are stored.
- **ITS area** : region where the data of the internal trusted storage service are stored in the clear.
- **Secure image primary slot** : region for programming the secure image of the “active” firmware.
- **Non-secure image primary slot** : region for programming the non secure image of the “active” firmware.
- **Secure image secondary slot** : region for programming the secure image of the “new” firmware.
- **Non-secure image secondary slot** : region for programming the non secure image of the “new” firmware.
- **Non-secure local loader**: region for programming the TFM_Loader non secure code binary.
- **Secure local loader**: region for programming the TFM_Loader secure code binary.

# Protection scheme during TFM_SBSFU_Boot application execution




During execution of TFM_SBSFU_Boot, the TFM_SBSFU_Boot code area is the only Flash memory area that is allowed to be executed, with the immutable local loader.

This figure highlights the protection features per TF-M region.

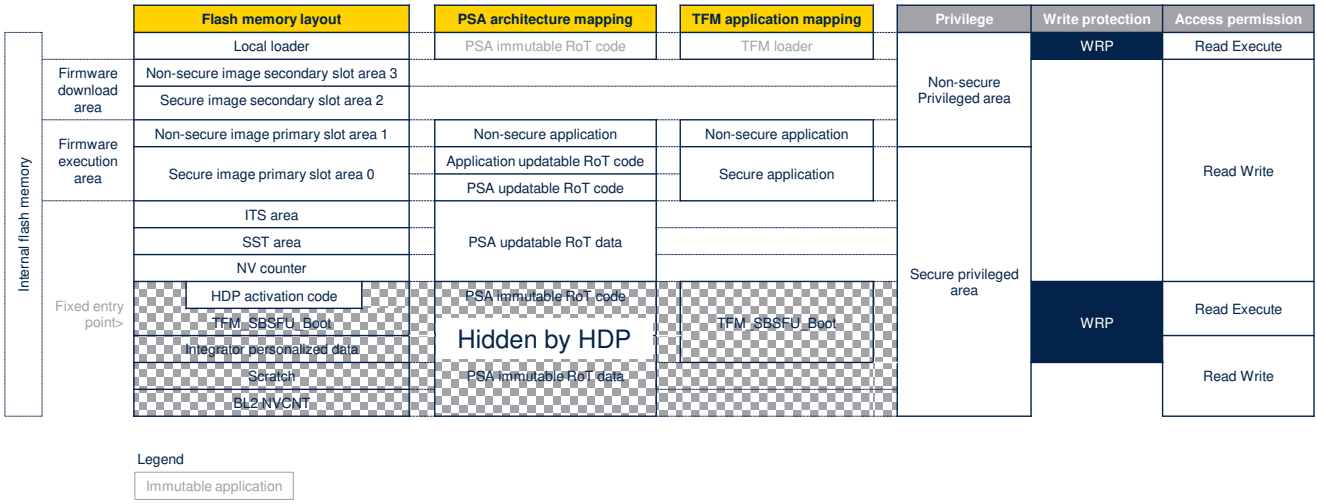
The local loader and firmware download areas as well as the non-secure application region are marked as non-secure & privileged.

The remaining part of the flash is secure and privileged.

The local loader and the TFM SBSFU boot program as well as integrator personalized data areas are write protected.

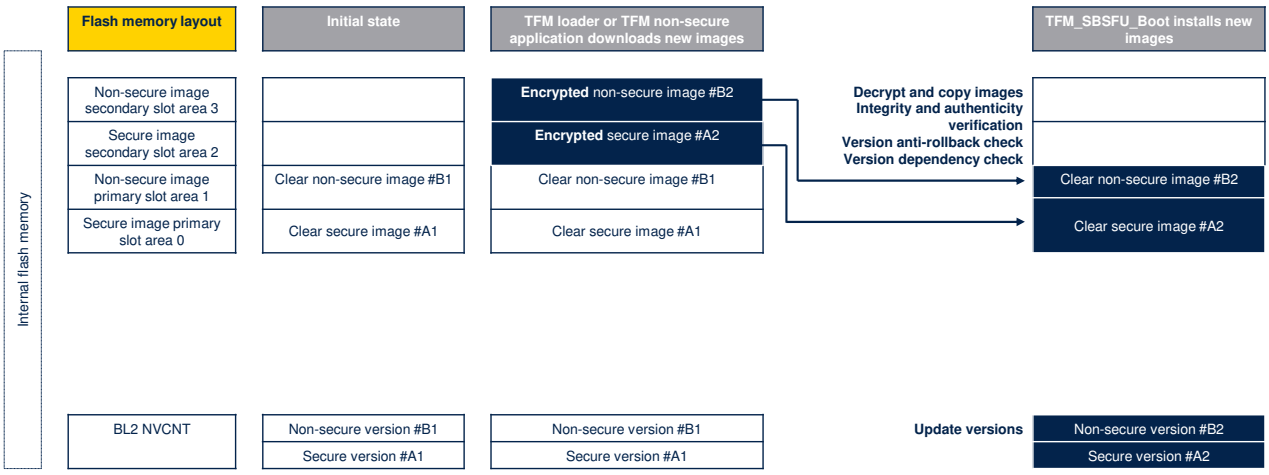
The local loader and the TFM SBSFU are the only regions for which execution is allowed.

# Protection scheme during TFM_SBSFU_Boot application execution




When exiting the TFM_SBSFU_Boot application to a secure application, all Flash memory areas dedicated to the execution of TFM_SBSFU_Boot are hidden, and execution is allowed in secure and non-secure primary slot areas. Detailed protection schemes covering all execution and transition cases can be found in UM2851.

# Firmware images update using default TFM configuration




The mechanisms for updating firmware images depends on the number of images, the image upgrade strategy, and the configuration of the slots mode.
The procedure is described here based on the default configuration.
It describes the procedure for downloading and installing new firmware for overwrite mode, the configuration of 2 firmware images and the configuration of the primary and secondary slots.
The loader downloads encrypted images which are decrypted and authenticated before being programmed in clear in the corresponding slot area.
The BL2 NVCNT region stores the data used to manage firmware version information for anti-rollback feature.

# Thank you



life.augmented

Thank you for attending this presentation!
You can now refer to the presentations that detail the operation of the TFM

- TFM flash memory footprint
- TFM pointers.