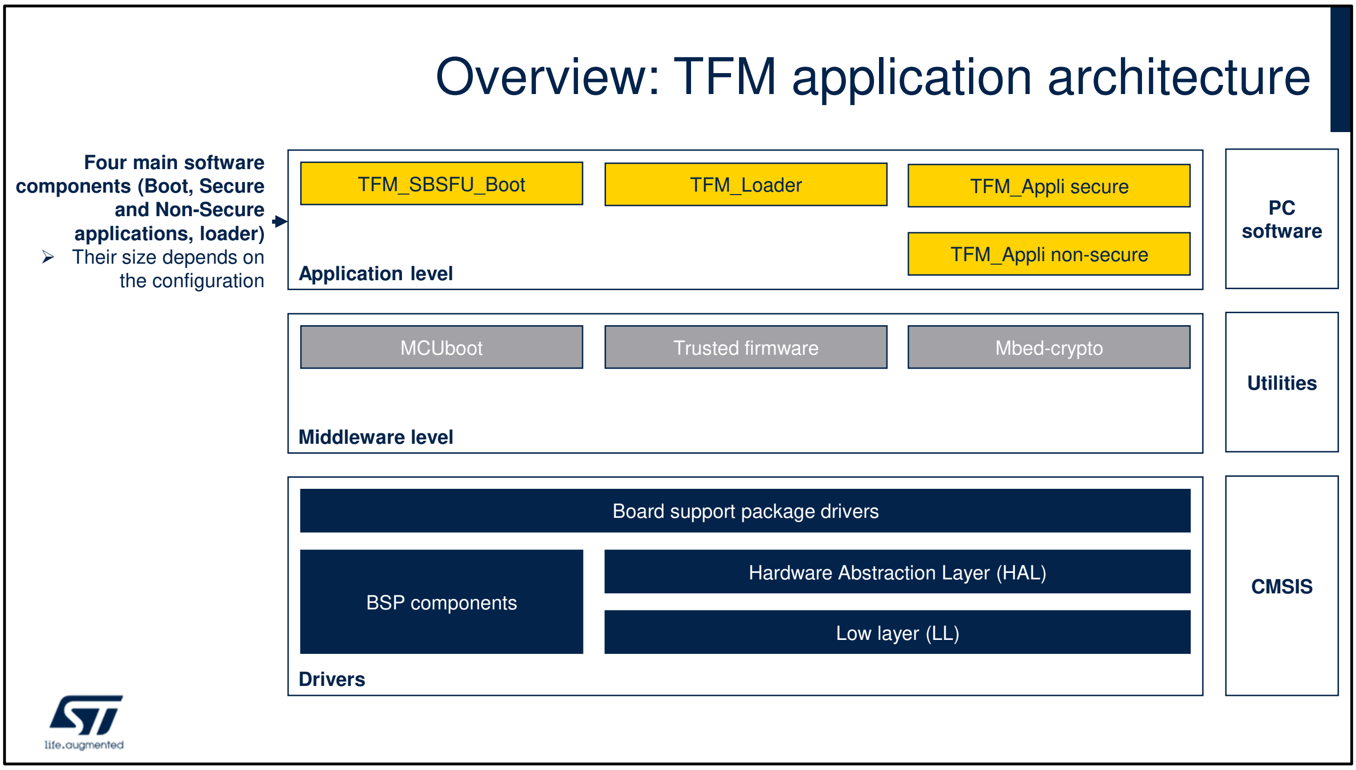
STM32U5

TFM flash memory footprint

Rev 1.0

Hello, and welcome to this presentation that details the parameters that affect the TFM flash memory footprint.



Overview: TFM application architecture

Four main software components (Boot, Secure and Non-Secure applications, loader)

- Their size depends on the configuration

Application level

TFM_SBSFU_Boot

TFM_Loader

TFM_Appli secure

TFM_Appli non-secure

Middleware level

MCUboot

Trusted firmware

Mbed-crypto

Drivers

Board support package drivers

BSP components

Hardware Abstraction Layer (HAL)

Low layer (LL)

PC software

Utilities

CMSIS

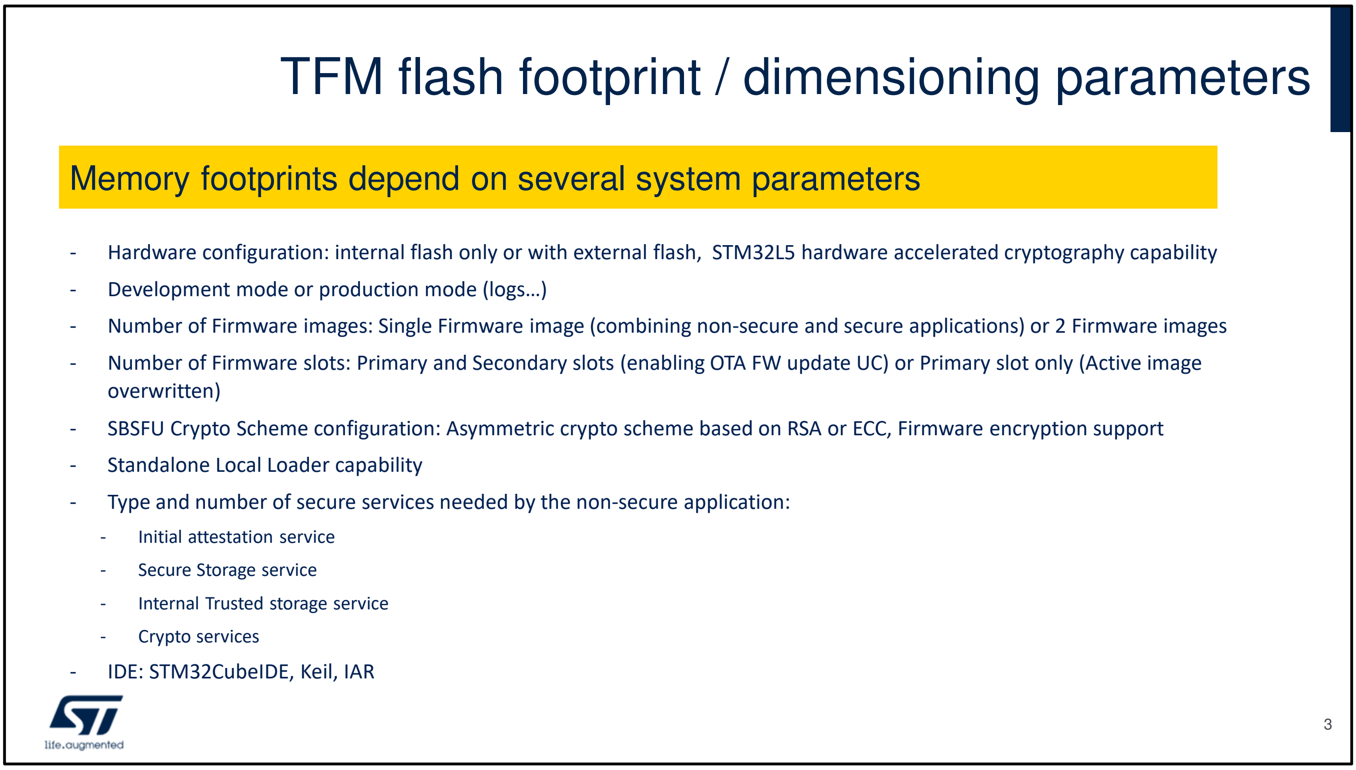
life.augmented

This figure summarizes the software layers on which TFM relies, the utilities for middleware and CMSIS for the drivers.

The TFM based application examples provided by STM32CubeU5 consist of four main software components, which can be configured according to user needs:

- TFM_SBSFU_Boot: Secure Boot and Secure Firmware Update application
- TFM_Loader : Application loader based on the Ymodem protocol over USART
- TFM_Appli_Secure: Secure application providing secure services to the non-secure user application (at run-time)
- TFM_Appli_NonSecure: Non-secure user application.

The following slides provide an estimation of the footprints for minimal and full configuration of each of these components.



TFM flash footprint / dimensioning parameters

Memory footprints depend on several system parameters

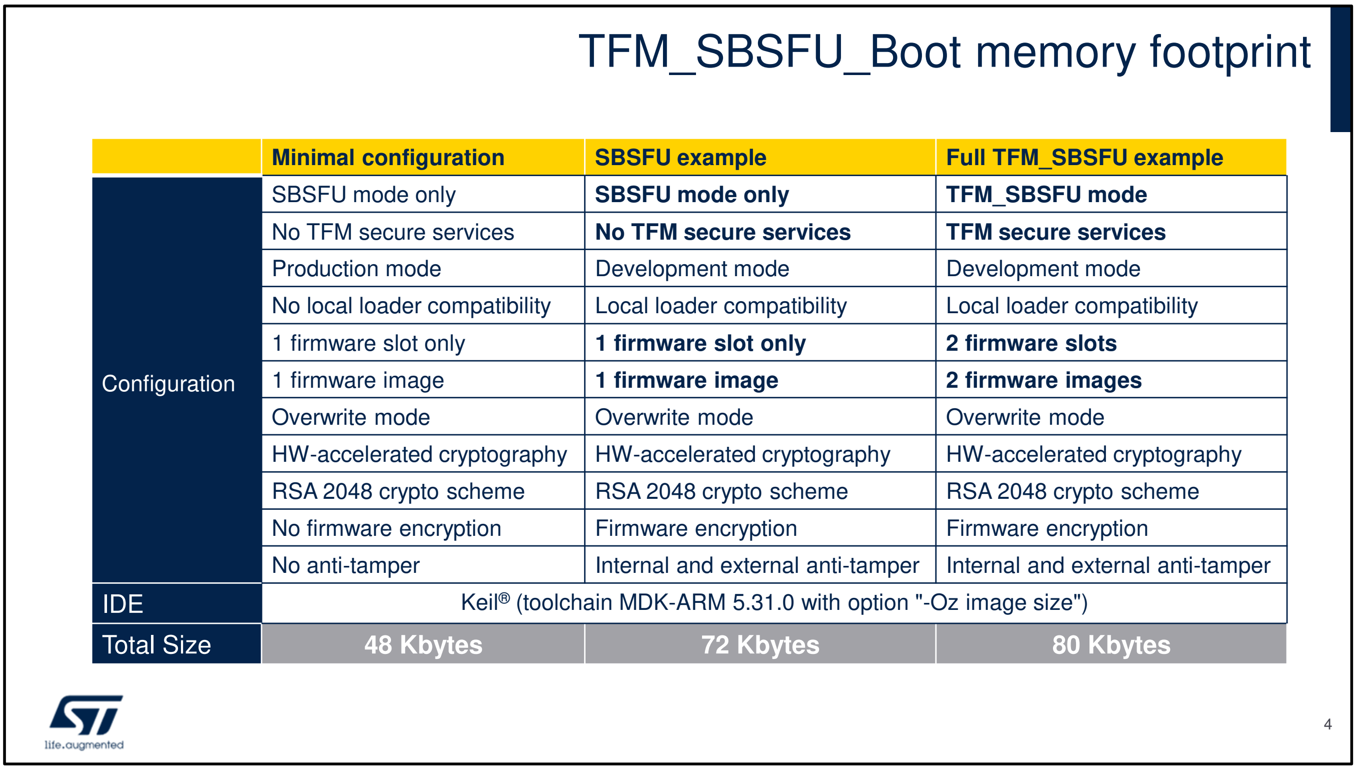
- Hardware configuration: internal flash only or with external flash, STM32L5 hardware accelerated cryptography capability
- Development mode or production mode (logs…)
- Number of Firmware images: Single Firmware image (combining non-secure and secure applications) or 2 Firmware images
- Number of Firmware slots: Primary and Secondary slots (enabling OTA FW update UC) or Primary slot only (Active image overwritten)
- SBSFU Crypto Scheme configuration: Asymmetric crypto scheme based on RSA or ECC, Firmware encryption support
- Standalone Local Loader capability
- Type and number of secure services needed by the non-secure application:
  - Initial attestation service
  - Secure Storage service
  - Internal Trusted storage service
  - Crypto services
- IDE: STM32CubeIDE, Keil, IAR

This slide lists the parameters that affect the TFM flash footprint.
The footprint depends on the hardware configuration of internal or external flash, especially the page size, and also the possible encryption of flash regions.
The number of firmware images and the number of firmware slots used to update the images impact the footprint.
The size is also affected by the SBSFU crypto scheme configuration: asymmetric based on RSA or ECC.
If an image loader is needed, it also consumes part of the flash.
The type and number of secure services obviously impact the memory footprint.
The code compactness also depends on the level of

optimization provided by the compiler.

In the following slides, metrics will be provided with the following settings:

- The size is aligned taking into account 8-Kbyte Flash memory sector alignment constraints
- The IDE is a Keil® toolchain MDK-ARM 5.31.0 with option "-Oz image size“.



TFM_SBSFU_Boot memory footprint

	Minimal configuration	SBSFU example	Full TFM_SBSFU example
Configuration	SBSFU mode only	**SBSFU mode only**	**TFM_SBSFU mode**
	No TFM secure services	**No TFM secure services**	**TFM secure services**
	Production mode	Development mode	Development mode
	No local loader compatibility	Local loader compatibility	Local loader compatibility
	1 firmware slot only	**1 firmware slot only**	**2 firmware slots**
	1 firmware image	**1 firmware image**	**2 firmware images**
	Overwrite mode	Overwrite mode	Overwrite mode
	HW-accelerated cryptography	HW-accelerated cryptography	HW-accelerated cryptography
	RSA 2048 crypto scheme	RSA 2048 crypto scheme	RSA 2048 crypto scheme
	No firmware encryption	Firmware encryption	Firmware encryption
	No anti-tamper	Internal and external anti-tamper	Internal and external anti-tamper
IDE	Keil® (toolchain MDK-ARM 5.31.0 with option "-Oz image size")		
Total Size	**48 Kbytes**	**72 Kbytes**	**80 Kbytes**

life.augmented

This table indicates the size of the TFM SBSFU boot program.

The TFM_SBSFU_Boot application consists of the following sections in Flash memory:

- BL2 NVCNT data: area used to store firmware version information for the anti-rollback feature.
- SCRATCH region: used by TFM_SBSFU_Boot to temporarily store image data during the image swap process (not used in overwrite mode)
- Integrator personal data: area used to store the SBSFU application keys and the keys and information used by the TFM secure application.
- SBSFU code: code managing the "Secure Boot" and the "Secure Firmware Update“ functions.

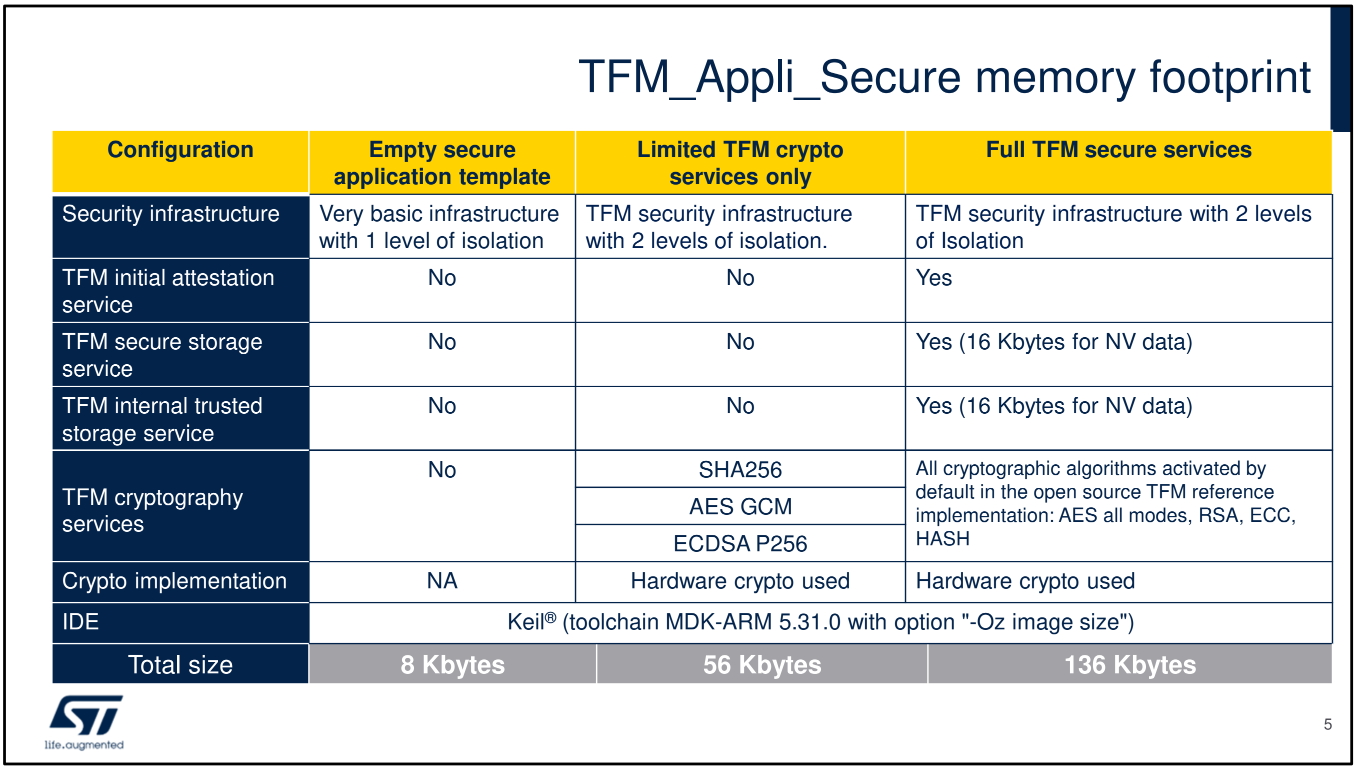
- HDP activation code: code that hides all SBSFU code and secrets before launching the application.

For more details about the footprint of each section refer to the UM2851 user’s manual entitled: Getting started with STM32CubeU5 TFM application.

This Table describes three examples as follows:

- Minimum configuration example
- SBSFU_Boot example delivered in the STM32CubeU5 MCU Package
- TFM_SBSFU_Boot example delivered in the STM32CubeU5 MCU Package.

The differences between the SBSFU and TFM examples are highlighted in bold.



TFM_Appli_Secure memory footprint

Configuration	Empty secure application template	Limited TFM crypto services only	Full TFM secure services
Security infrastructure	Very basic infrastructure with 1 level of isolation	TFM security infrastructure with 2 levels of isolation.	TFM security infrastructure with 2 levels of Isolation
TFM initial attestation service	No	No	Yes
TFM secure storage service	No	No	Yes (16 Kbytes for NV data)
TFM internal trusted storage service	No	No	Yes (16 Kbytes for NV data)
TFM cryptography services	No	SHA256 AES GCM ECDSA P256	All cryptographic algorithms activated by default in the open source TFM reference implementation: AES all modes, RSA, ECC, HASH
Crypto implementation	NA	Hardware crypto used	Hardware crypto used
IDE	Keil® (toolchain MDK-ARM 5.31.0 with option "-Oz image size")		
Total size	8 Kbytes	56 Kbytes	136 Kbytes

This table indicates the size of the TFM secure application.

The secure application provides secure services that can be used by the non-secure application at run-time:

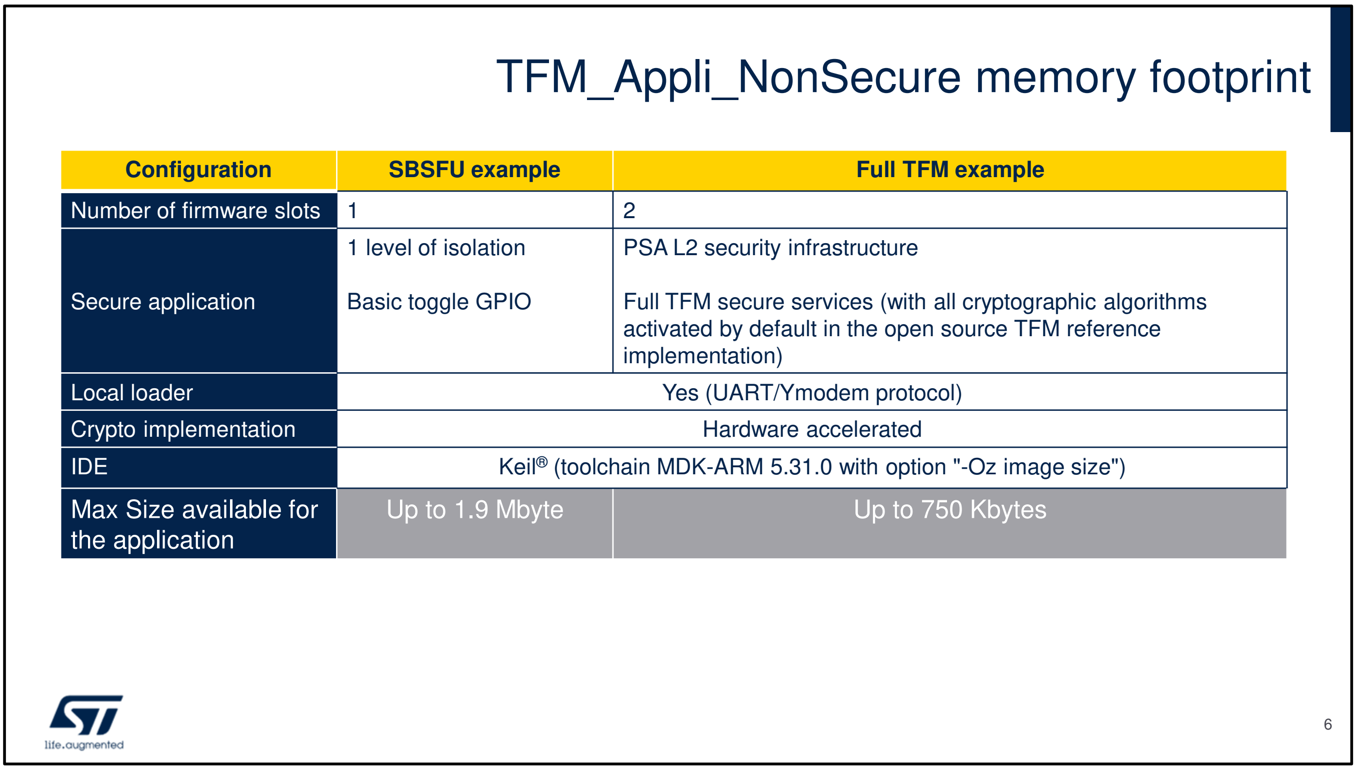
- Configuration of the security architecture with the isolation of the different domains and with the secure APIs mechanisms
- Providing secure services needed by the non-secure user application.

The secure application binary is encapsulated in a firmware image, which contains some metadata that are used in the context of the "Secure Boot" or "Secure Firmware Update" functions.

The size of the secure application image is affected by the configuration.

This Table describes three examples:

- Empty secure application template
- Limited TFM crypto services only
- Full TFM secure services.



TFM_Appli_NonSecure memory footprint

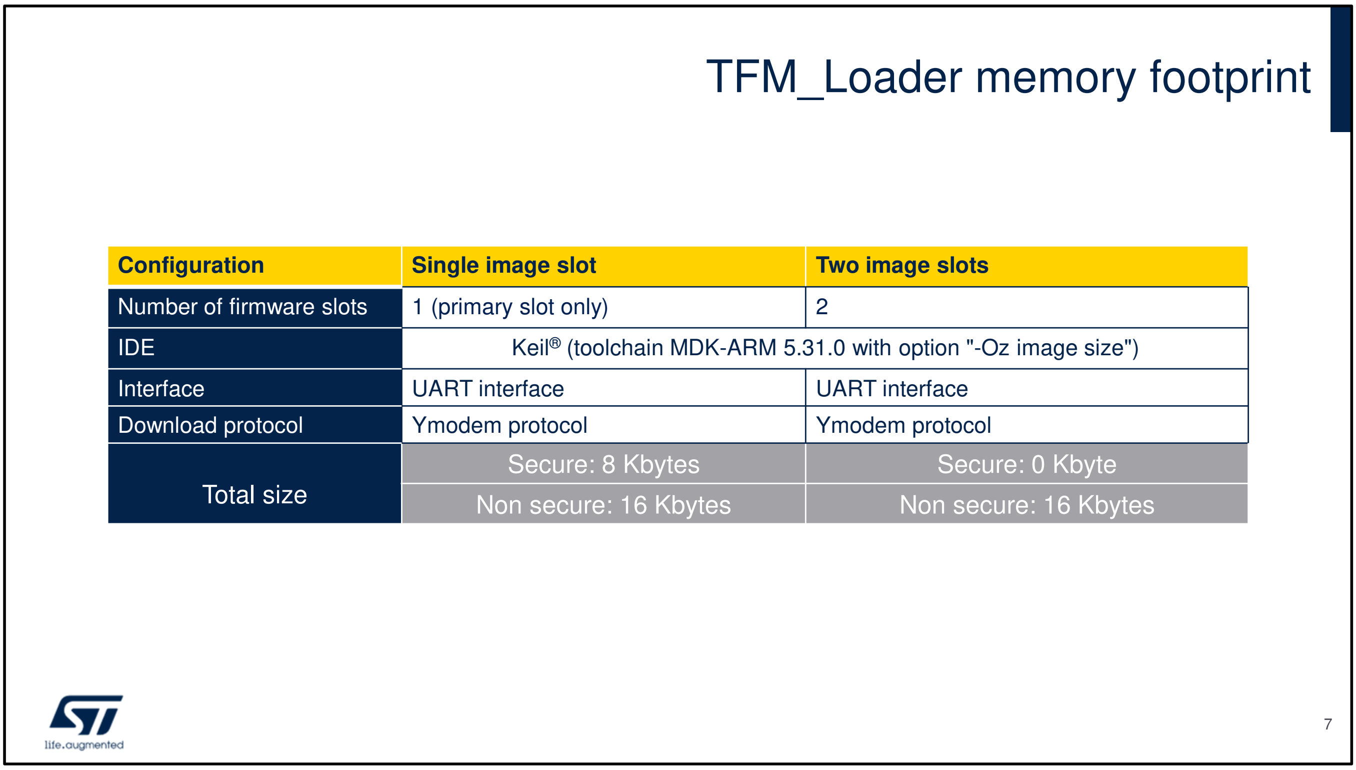
Configuration	SBSFU example	Full TFM example
Number of firmware slots	1	2
Secure application	1 level of isolation Basic toggle GPIO	PSA L2 security infrastructure Full TFM secure services (with all cryptographic algorithms activated by default in the open source TFM reference implementation)
Local loader	Yes (UART/Ymodem protocol)	
Crypto implementation	Hardware accelerated	
IDE	Keil® (toolchain MDK-ARM 5.31.0 with option "-Oz image size")	
Max Size available for the application	Up to 1.9 Mbyte	Up to 750 Kbytes

This table indicates the size of the TFM non-secure application.

If internal Flash memory is used, the size available for the non-secure application area depends on the configurations.

This Table describes the two examples provided in the STM32CubeU5 MCU Package:

- SBSFU example
- Full TFM example.



TFM_Loader memory footprint

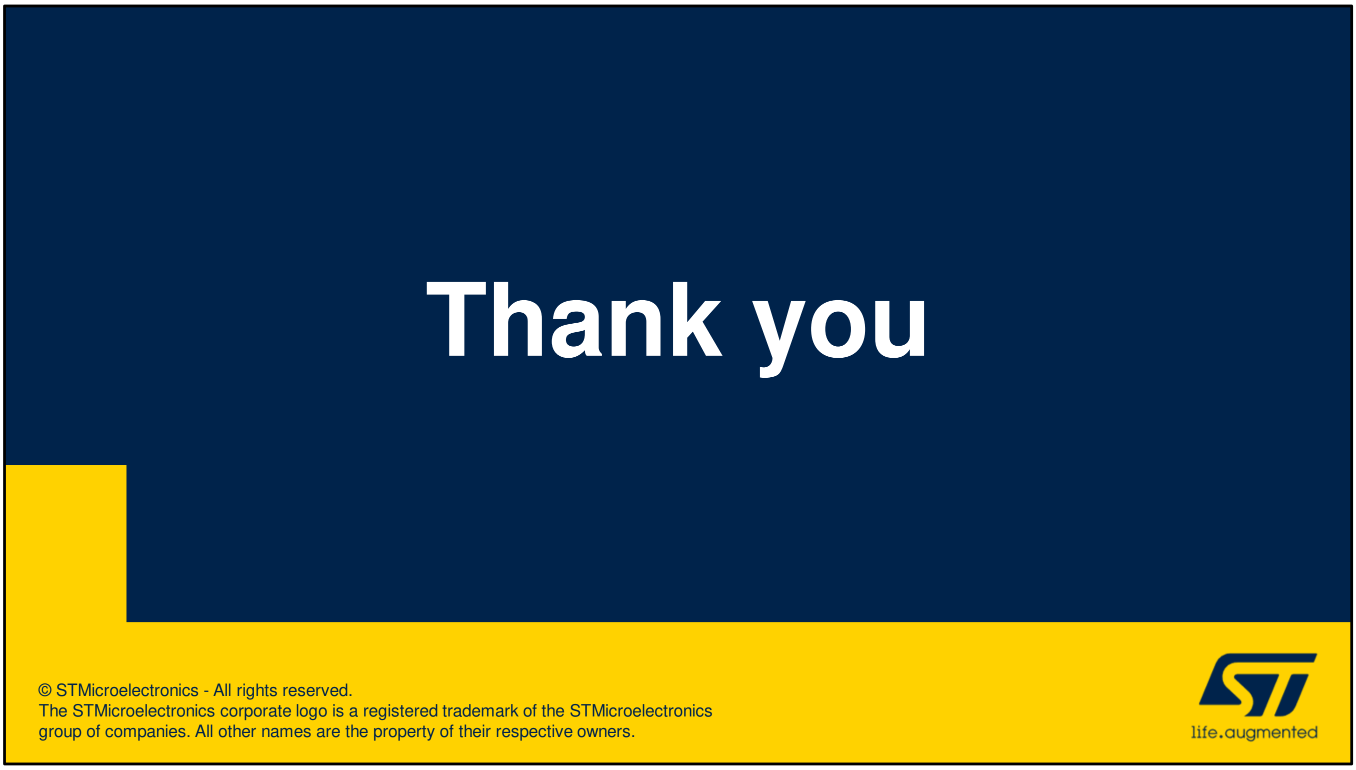
Configuration	Single image slot	Two image slots
Number of firmware slots	1 (primary slot only)	2
IDE	Keil® (toolchain MDK-ARM 5.31.0 with option "-Oz image size")	
Interface	UART interface	UART interface
Download protocol	Ymodem protocol	Ymodem protocol
Total size	Secure: 8 Kbytes	Secure: 0 Kbyte
	Non secure: 16 Kbytes	Non secure: 16 Kbytes

This table indicates the size of the TFM loader

The TFM_Loader application enables the download of new firmware versions using the UART interface with the Ymodem protocol (as an example).
The TFM_Loader application is optional; it can be fully removed if not needed.
Integrators can configure it according to their product specifications and can customize it to support other hardware interfaces or to support other protocols.

The size of the TFM_Loader application can be affected by the configuration.
This Table describes two examples:
- Single image slot

- Two image slots.



Thank you

life.augmented

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

- TFM offer in STM32U5
- TFM pointers.