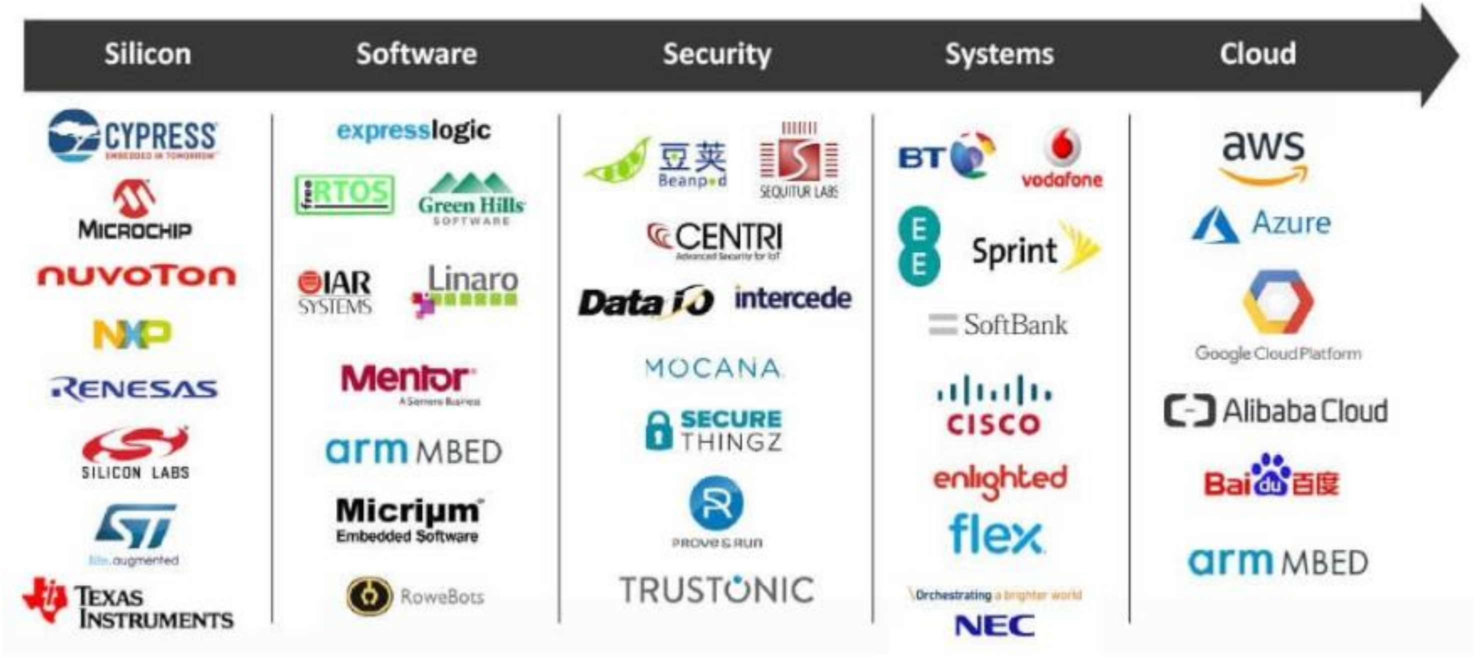
life.augmented

STM32 Security Workshop

PSA-TFM presentation

ARM Platform Security Architecture

- PSA : is an ARM initiative which establishes the general method for securing devices from the very start of the product lifecycle.
- This standard pushed covering the entire IoT ecosystem, from chip designers and device developers to cloud and network infrastructure providers and software vendors.



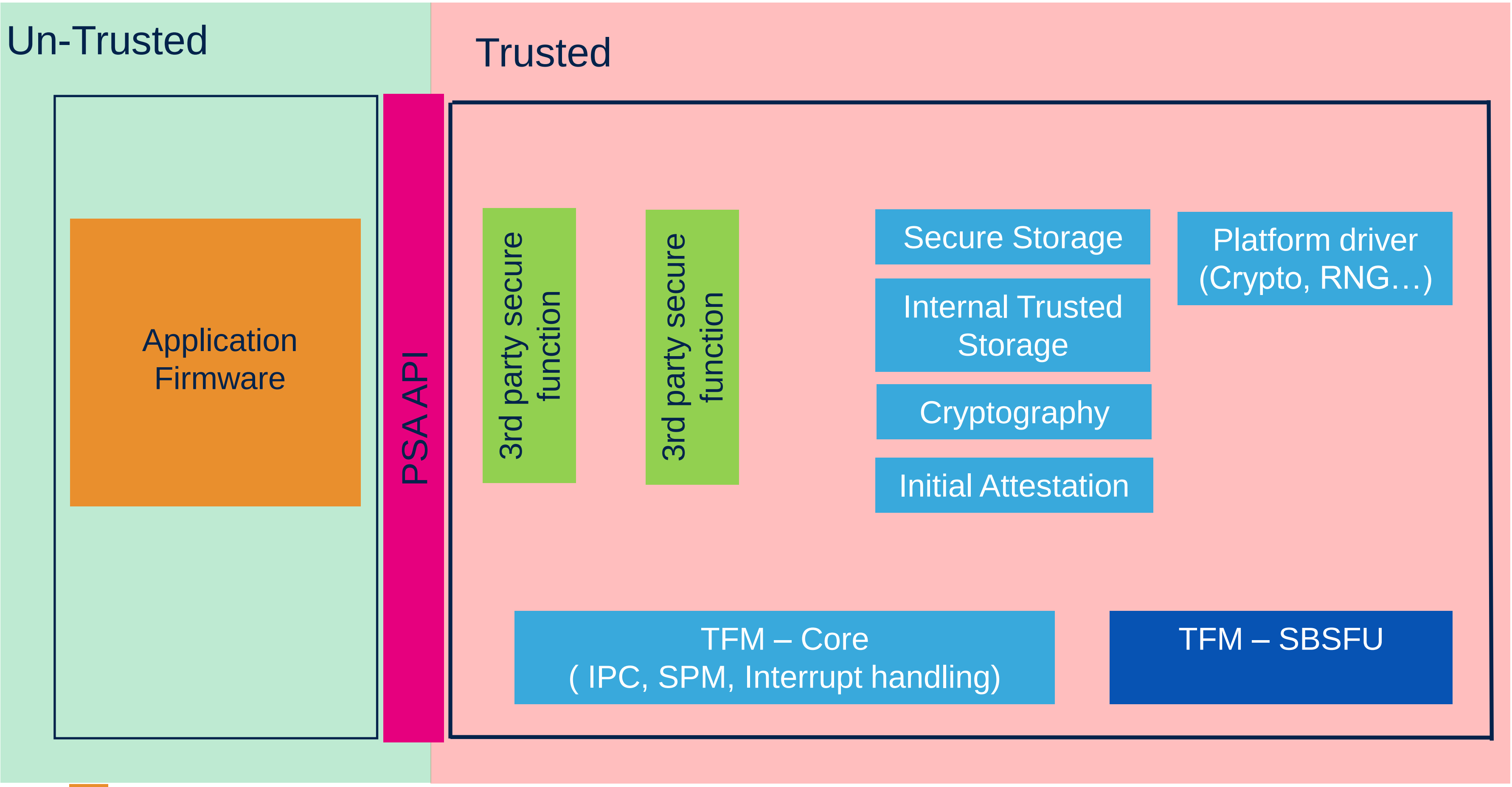
ARM Platform Security Architecture

- In October 2018 ARM announce a PSA compliance program and publish the first version of PSA certification
- The Platform Security Architecture (PSA) is made up of four key stages:

Analyze
Architect
01001
Implement
Certify

https://developer.arm.com/architectures/security-architectures/platform-security-architecture

TFM on Cortex-M33



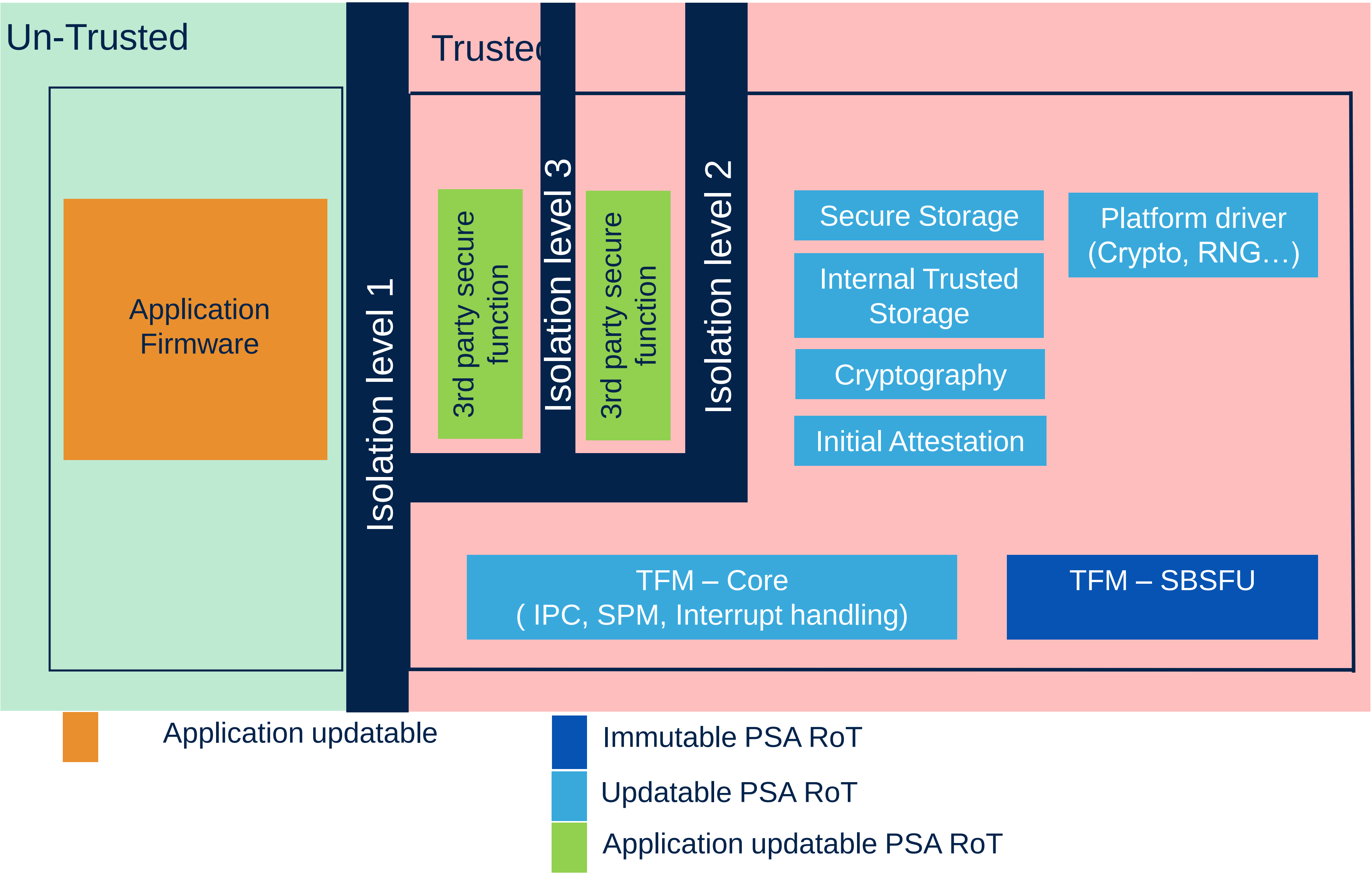
🟧 Application updatable

🟦 Immutable PSA RoT

🟦 Updatable PSA RoT

🟩 Application updatable PSA RoT

TFM isolation on Cortex-M33



Un-Trusted

Application Firmware

Isolation level 1

Trusted

3rd party secure function

Isolation level 3

3rd party secure function

Isolation level 2

Secure Storage

Internal Trusted Storage

Cryptography

Initial Attestation

Platform driver (Crypto, RNG…)

TFM – Core (IPC, SPM, Interrupt handling)

TFM – SBSFU

🟧 Application updatable

🟦 Immutable PSA RoT

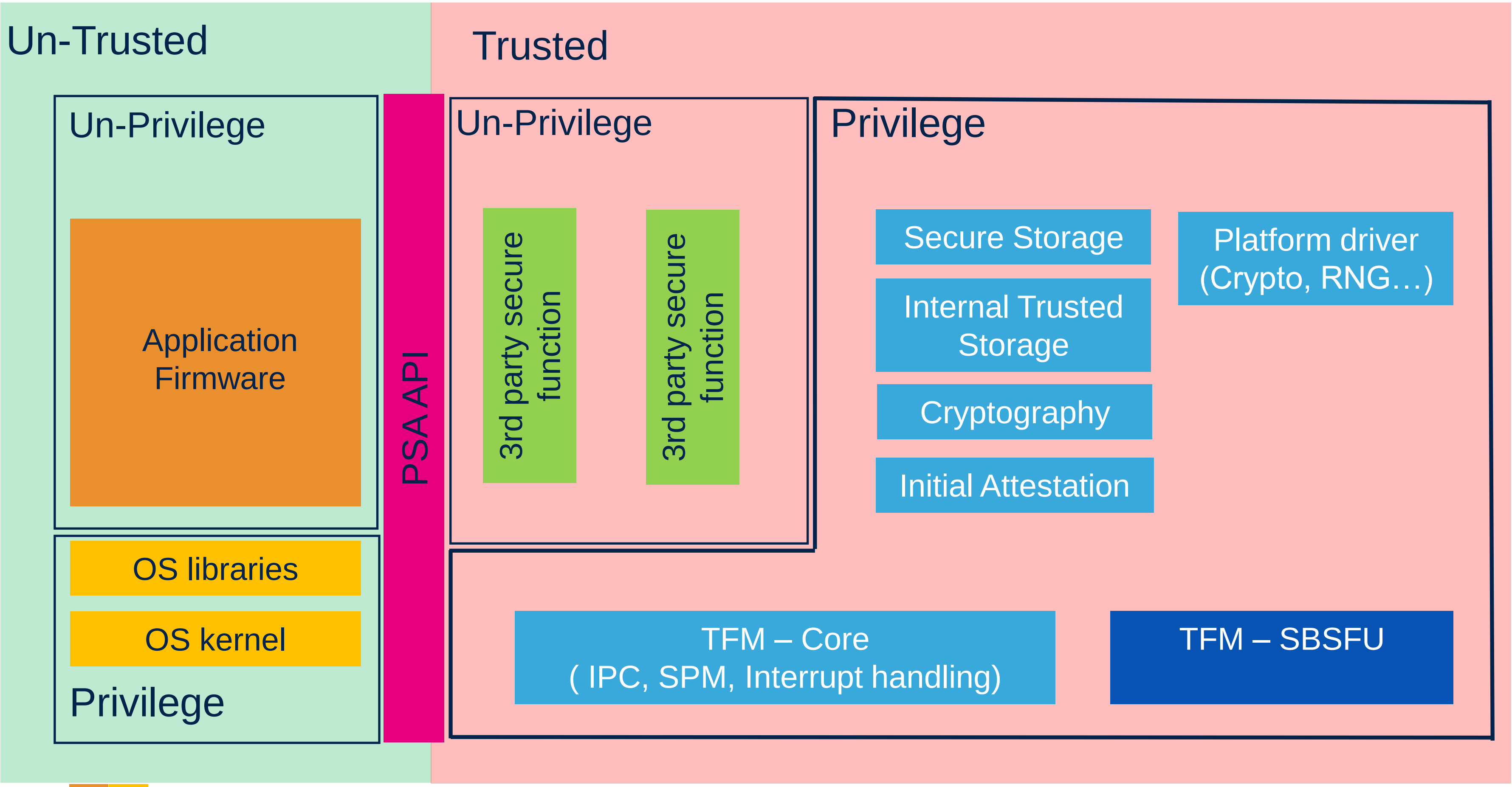
🟦 Updatable PSA RoT

🟩 Application updatable PSA RoT

ARM Trusted Firmware-M (TF-M)

- Trusted Firmware-M (TF-M) providing a reference implementation of PSA standard on ARM-CM33
- Current version support PSA Level 1 and 2 isolation on Armv8-M.
- ST ported TF-M code on STM32L5 with isolation level 2

TFM isolation on Cortex-M33



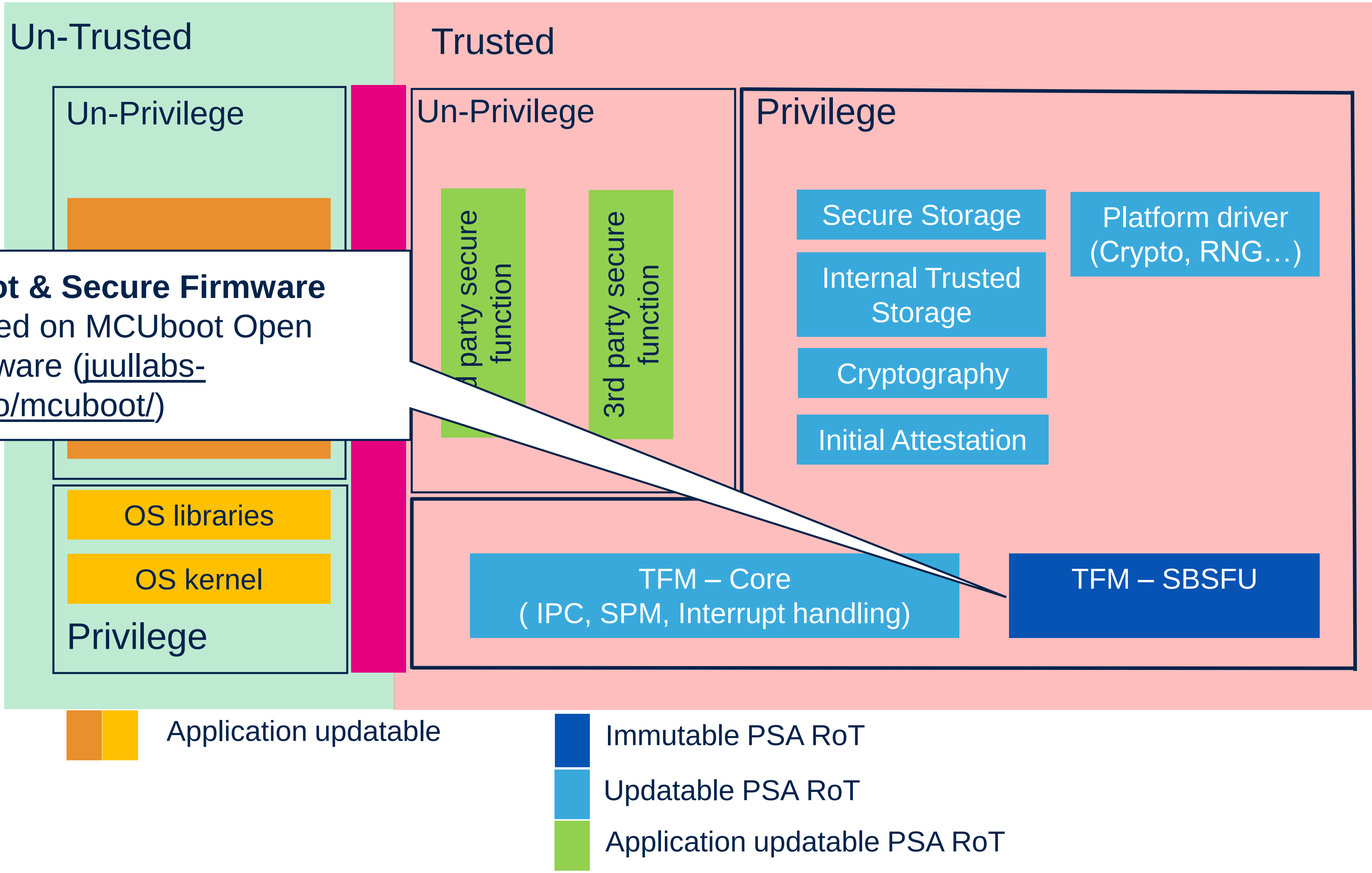
🟧🟨 Application updatable

🟦 Immutable PSA RoT

🟦 Updatable PSA RoT

🟩 Application updatable PSA RoT

TFM isolation on Cortex-M33



Secure Boot & Secure Firmware Update based on MCUboot Open Source software (juullabs-oss.github.io/mcuboot/)

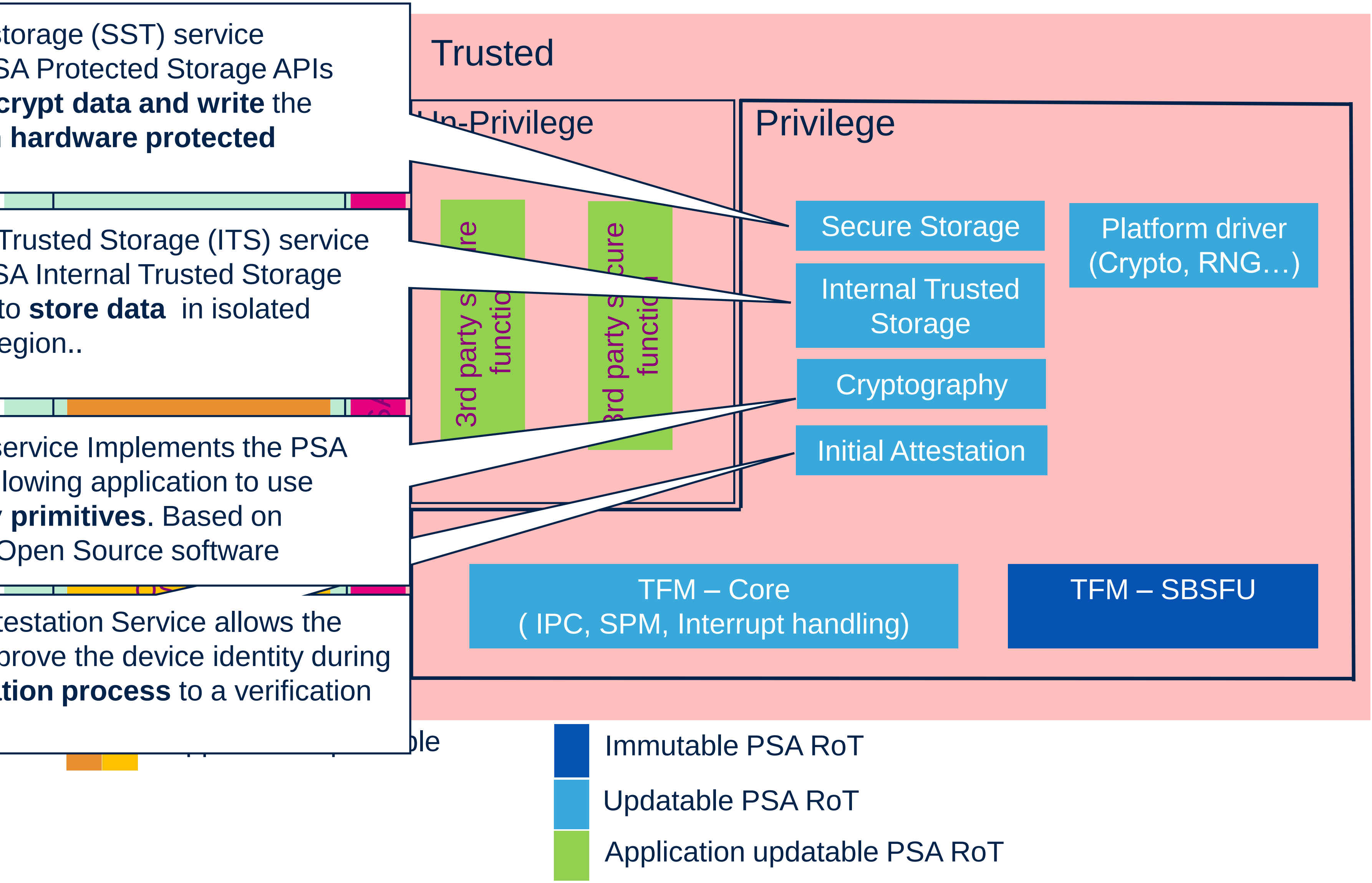
TFM isolation on Cortex-M33

TF-M secure storage (SST) service implements PSA Protected Storage APIs allowing to **encrypt data and write** the result in a **non hardware protected storage.**

TF-M Internal Trusted Storage (ITS) service implements PSA Internal Trusted Storage APIs allowing to **store data** in isolated internal flash region..

TF-M Crypto service Implements the PSA Crypto APIs allowing application to use **cryptography primitives**. Based on **MbedCrypto** Open Source software

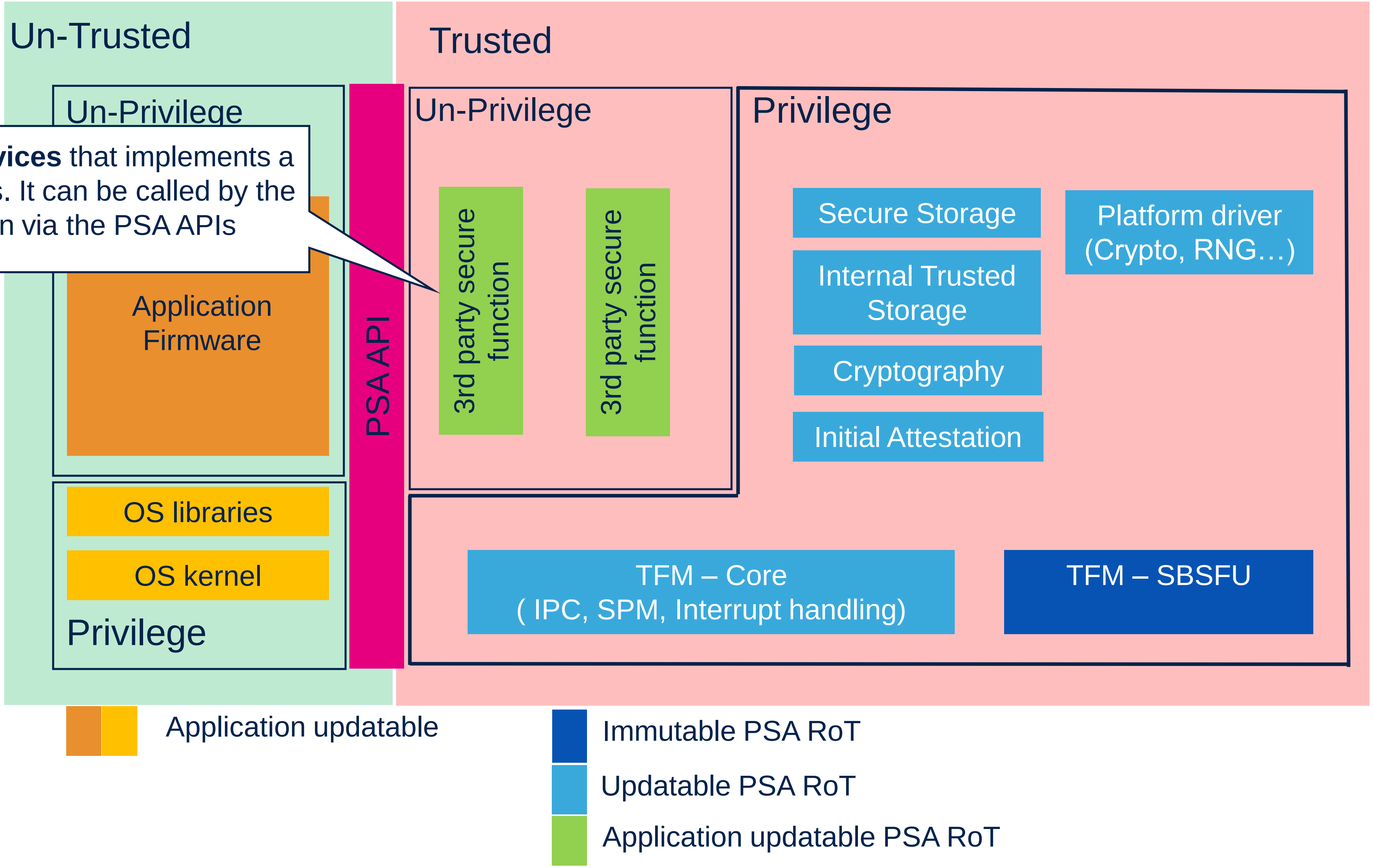
TF-M Initial Attestation Service allows the application to prove the device identity during an **authentication process** to a verification entity.



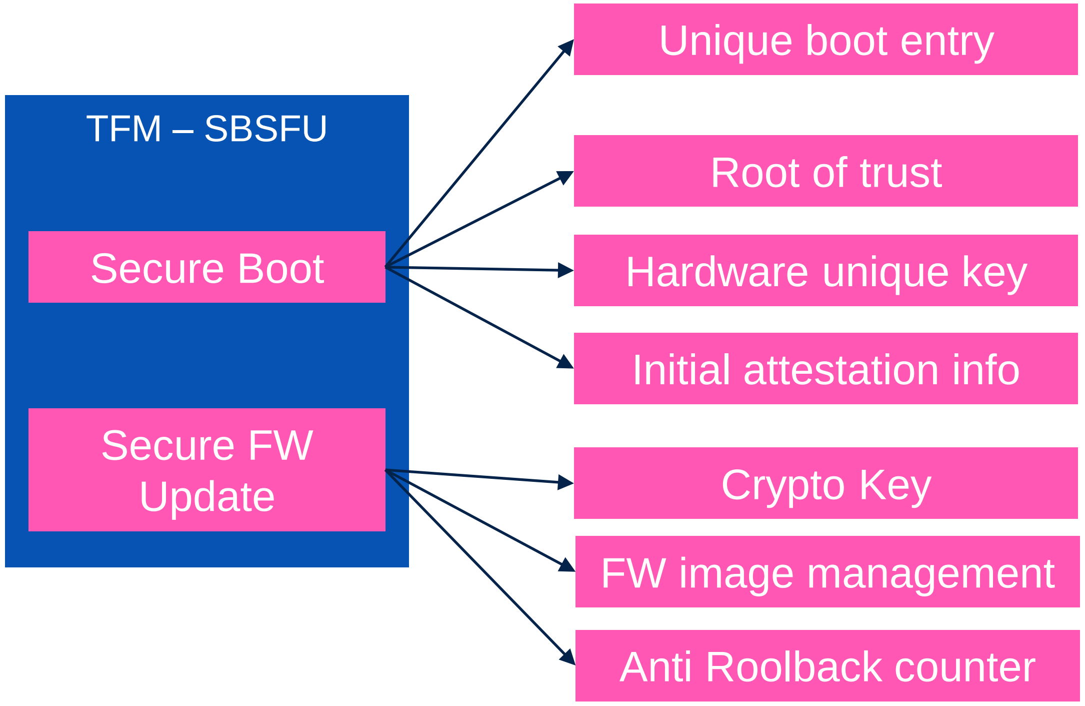
Trusted
Un-Privilege
Privilege
3rd party secure functions
3rd party secure functions
Secure Storage
Internal Trusted Storage
Cryptography
Initial Attestation
Platform driver (Crypto, RNG…)
TFM – Core (IPC, SPM, Interrupt handling)
TFM – SBSFU
ple
Immutable PSA RoT
Updatable PSA RoT
Application updatable PSA RoT

TFM isolation on Cortex-M33

3rd party **secure services** that implements a set of secure services. It can be called by the non-secure application via the PSA APIs



TFM-STM32L5 Protection



TFM – SBSFU

Secure Boot

Secure FW Update

Unique boot entry

Root of trust

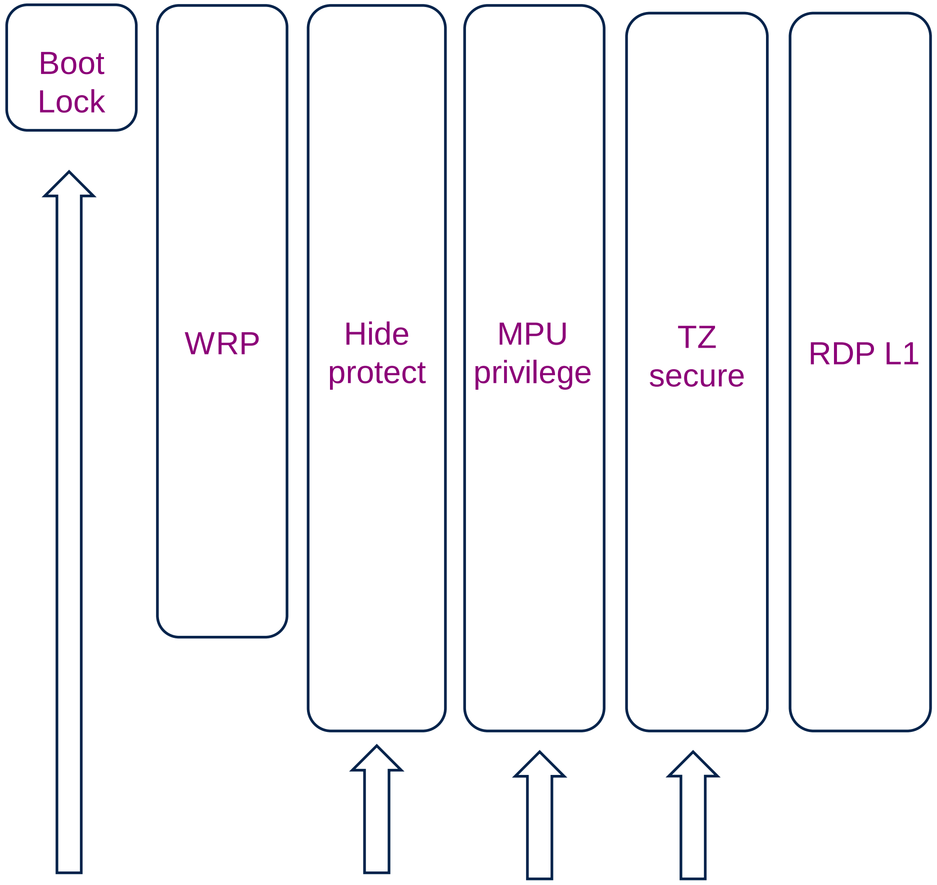
Hardware unique key

Initial attestation info

Crypto Key

FW image management

Anti Roolback counter



Boot Lock

WRP

Hide protect

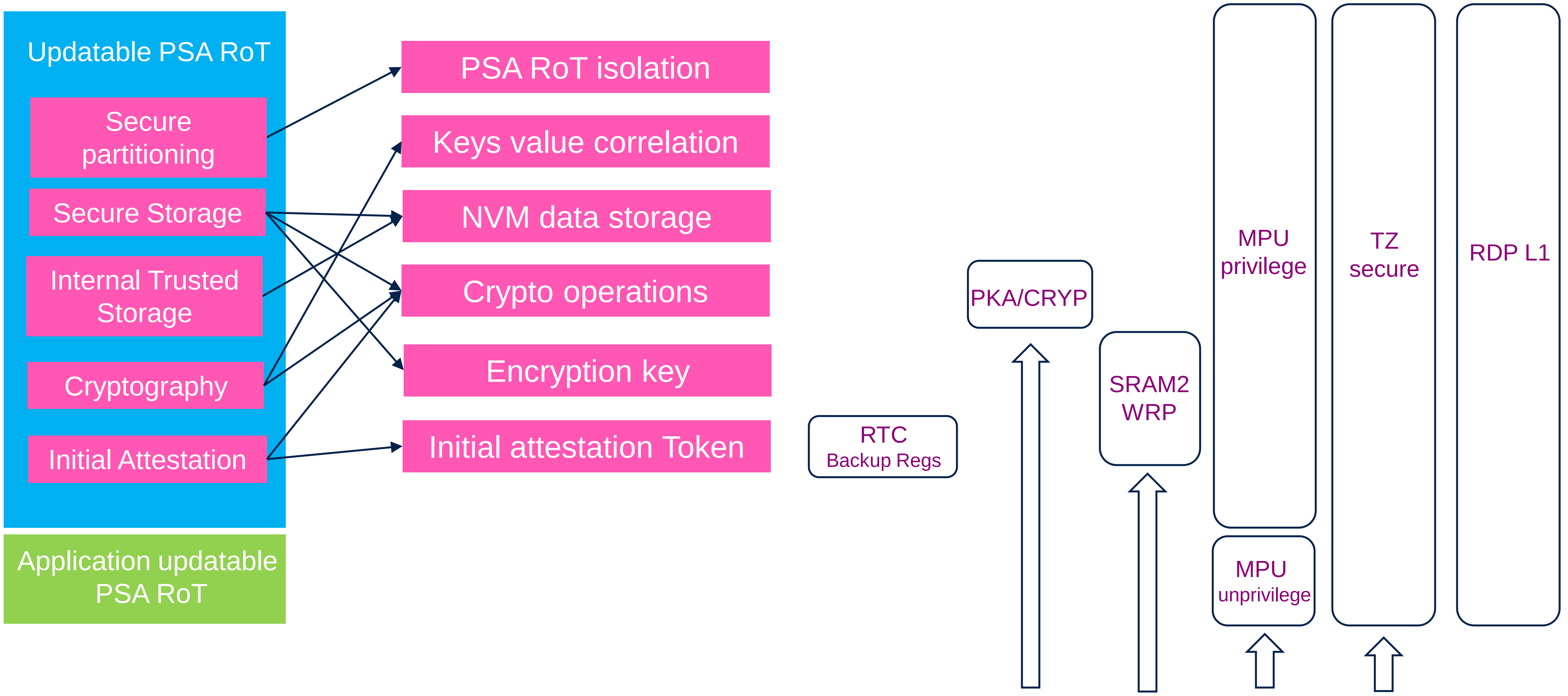
MPU privilege

TZ secure

RDP L1

L5 improvement (Vs L4)

TFM-STM32L5 Protection



L5 improvement (Vs L4)

TFM-STM32L5 delivery feature

- TFM-SBSFU (TFM_SBSFU_Boot)
  - Authentication : RSA 2048, RSA3072 or ECC256
  - SHA256 integrity check
  - Confidentiality : AES CTR with symmetric key encrypted in RSA-OAP or ECIES-P256
  - Dual slot mode
  - Dual image mode (1 image for secure application / 1 image for non secure application)
  - External memory support via OTFDEC (only the secure application primary slot in internal flash)

Default config in the package which could be modified

life.augmented

TFM-STM32L5 delivery feature

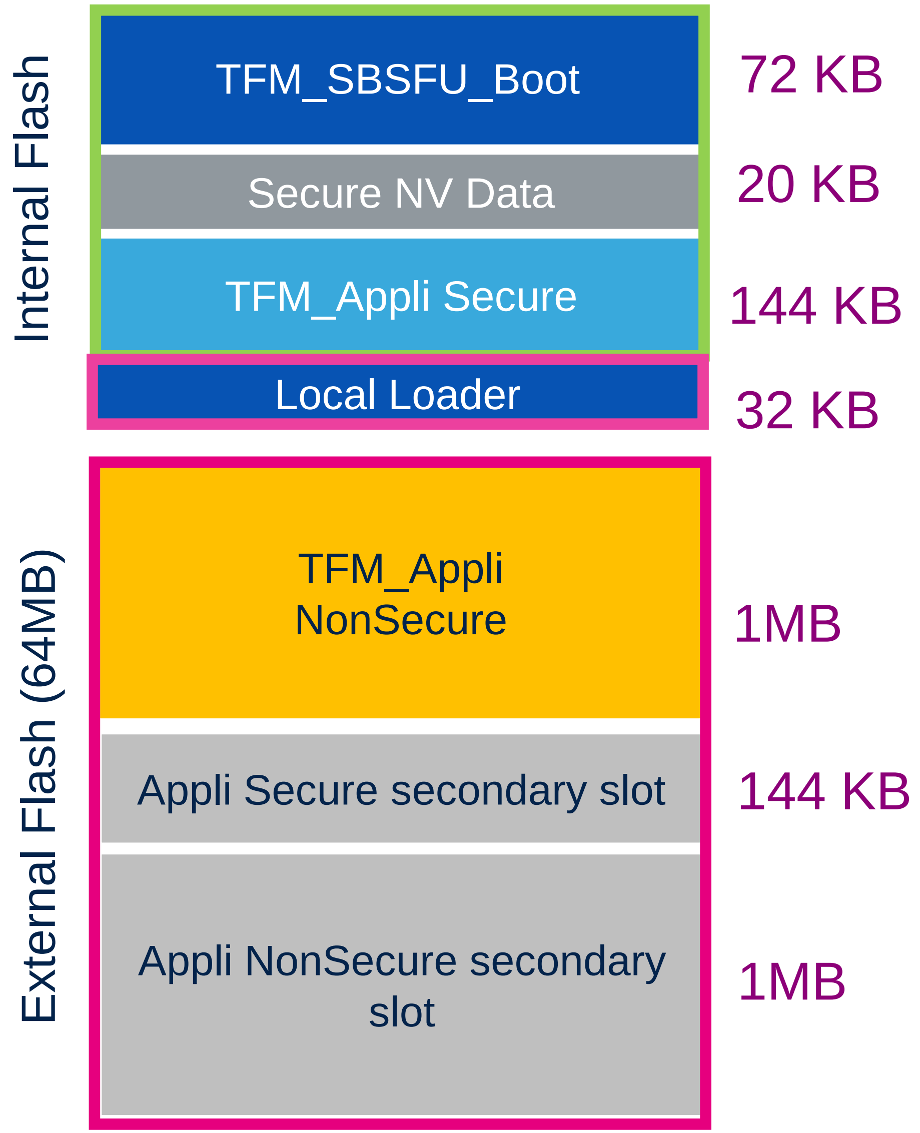
- TFM secure services (TFM_Appli secure)
  - PSA isolation level 2
  - Cryptography
    AES-CBC, AES-CFB, AES-CTR, AES-OFB, AES-CCM, AES-GCM, RSA, ECDSA, ECDH, SHA1, SHA256, SHA512
    software crypt or mix of software/hardware
  - Initial attestation
    Entity token encoded CBOR (RFC7049)/signature SHA256 and ECDSA
  - Secure Storage
    AES GCM based AEAD encryption
  - Internal Trusted Storage
- TFM local loader (TFM_Loader)
  - Non secure application immutable
  - YModem

Default config in the package which could be modified

life.augmented

TFM STM32L5 Memory Layout

TFM-STM32L5 package default config



🟩 Secure

🟥 Non Secure

SBSFU-STM32L5 delivery

- SBSFU-STM32L5 package:
  ST deliver a SBSFU_Boot which in fact is TFM example code where security services has been removed.
- SBSFU_Boot
  - Authentication : RSA 2048, RSA3072 or ECC256
  - SHA256 integrity check
  - Confidentiality : AES CTR with symmetric key encrypted in RSA-OAP or ECIES-P256
  - Single mode or Dual slot mode :

    Single slot : the download slot is also the execution slot
    Dual slot   : the download slot and execution slot are distinct
  - One image mode or dual image mode :

    One image : secure and non secure application are combined in one binary with one set of metadata
    Dual image : secure and non secure application are distinct

Default config in the package which could be modified

life.augmented

SBSFU-STM32L5 delivery

- Secure service (SBSFU_Appli Secure)
  - LED blinking example
- SBSFU local loader (SBSFU_Loader)
  - Secure and Non secure application
  - Immutable
  - YModem

Default config in the package which could be modified

life.augmented

SBSFU STM32L5 Memory Layout

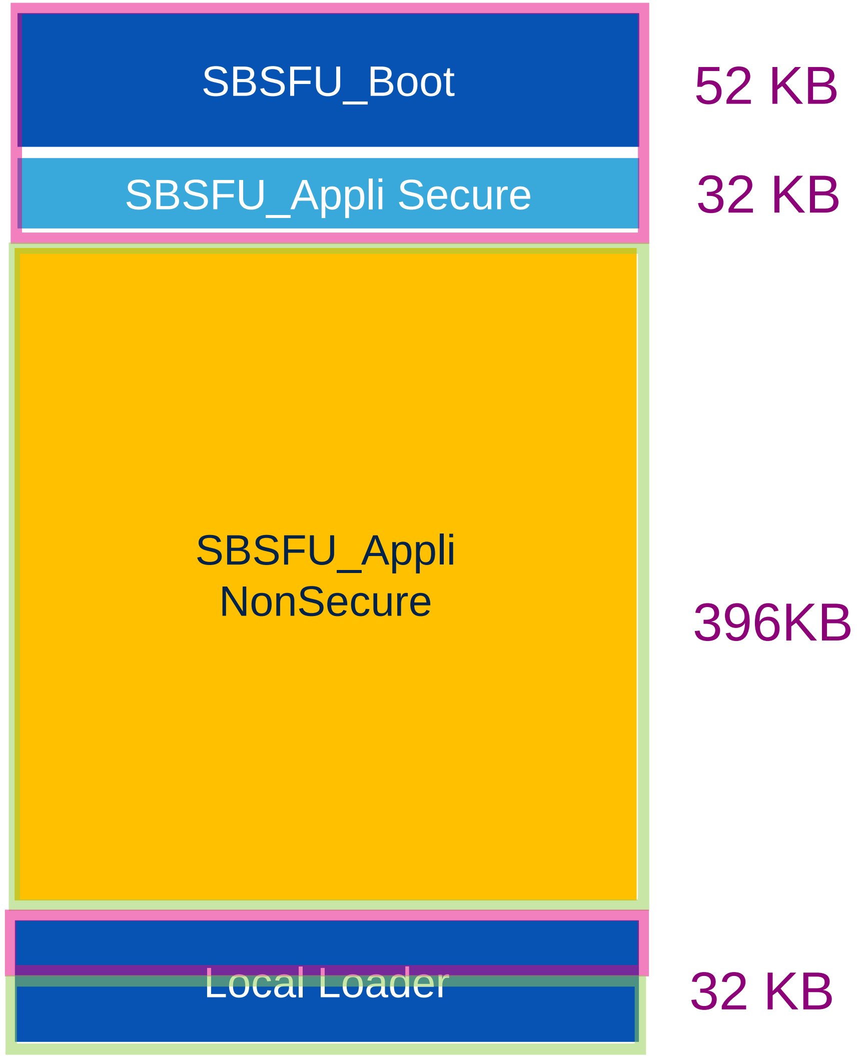
SBSFU-STM32L5 package

- single slot mode :
the download slot is the execution slot

- one image :
1 image for SBSFU_Appli secure + Non secure and 1 signature for this binary

🟥 Secure

🟩 Non Secure



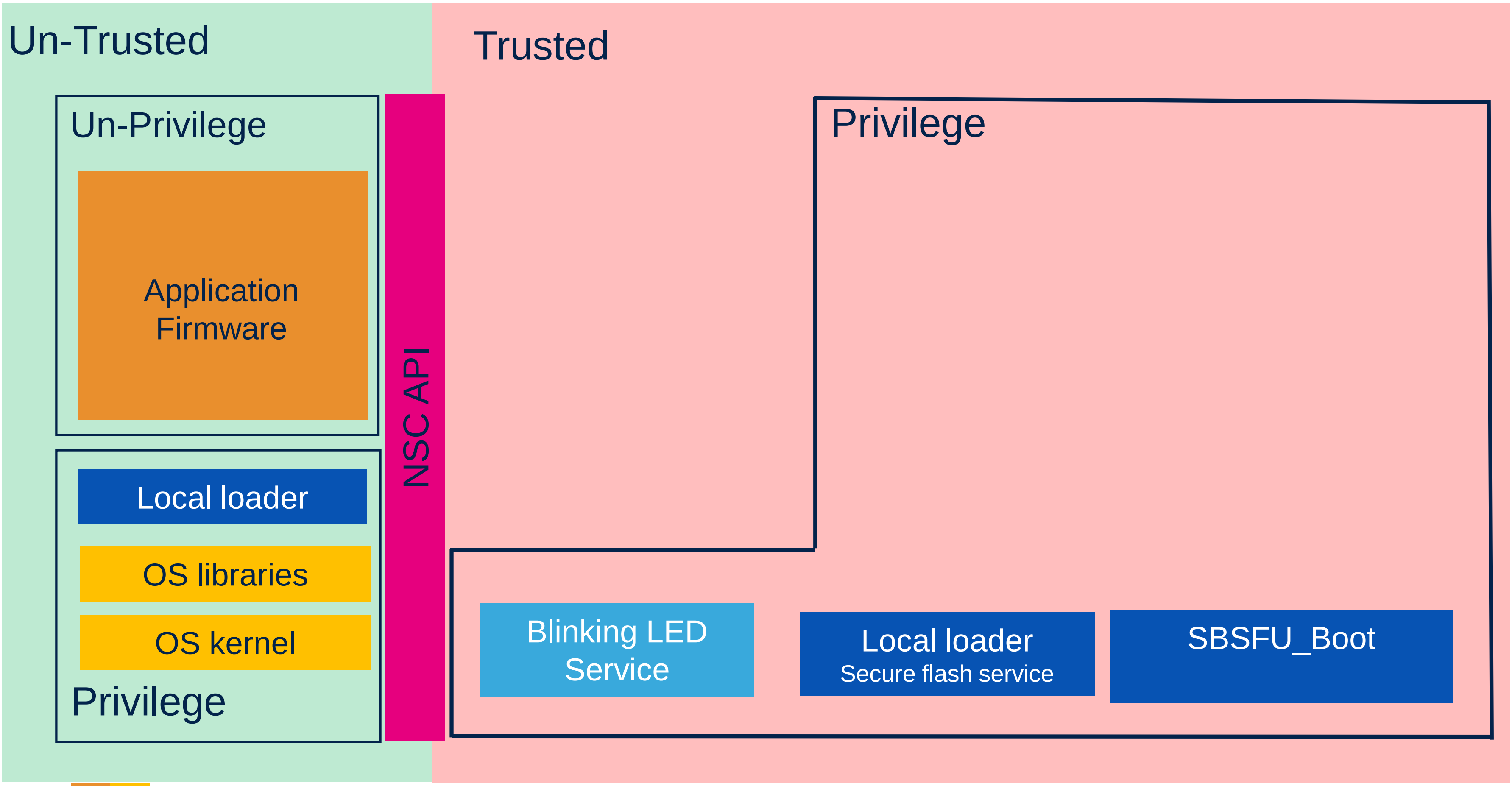
SBSFU_Boot 52 KB

SBSFU_Appli Secure 32 KB

SBSFU_Appli NonSecure 396KB

Local Loader 32 KB

SBSFU-STM32L5



Un-Trusted

Un-Privilege

Application Firmware

Local loader

OS libraries

OS kernel

Privilege

NSC API

Trusted

Privilege

Blinking LED Service

Local loader
Secure flash service

SBSFU_Boot

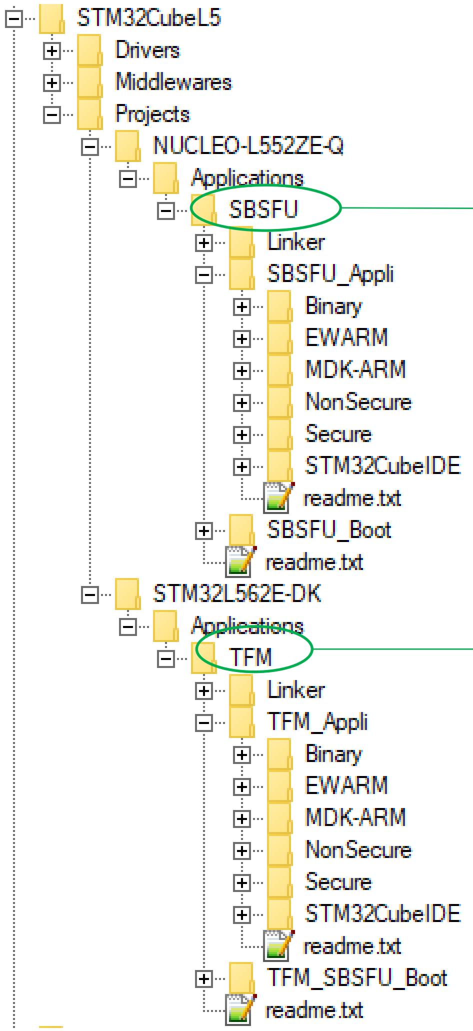
🟧🟨 Application updatable

🟦 Immutable PSA RoT

🟦 Updatable PSA RoT

🟩 Application updatable PSA RoT

SBSFU/TFM-STM32L5 package

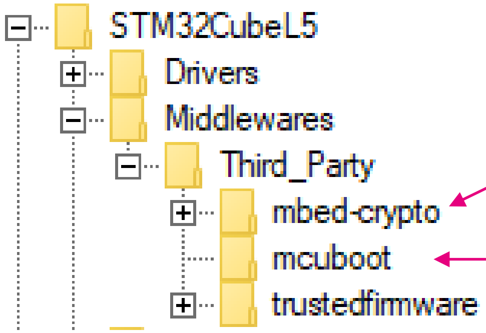


```
STM32CubeL5
    Drivers
    Middlewares
    Projects
        NUCLEO-L552ZE-Q
            Applications
                SBSFU
                    Linker
                    SBSFU_Appli
                        Binary
                        EWARM
                        MDK-ARM
                        NonSecure
                        Secure
                        STM32CubeIDE
                        readme.txt
                    SBSFU_Boot
                    readme.txt
        STM32L562E-DK
            Applications
                TFM
                    Linker
                    TFM_Appli
                        Binary
                        EWARM
                        MDK-ARM
                        NonSecure
                        Secure
                        STM32CubeIDE
                        readme.txt
                    TFM_SBSFU_Boot
                    readme.txt
```

Secure Boot & Secure Firmware Update
Starting example when coming from standard X-CUBE-SBSFU example.

Secure Boot & Secure Firmware Update + TFM secure services

SBSFU-STM32L5 package

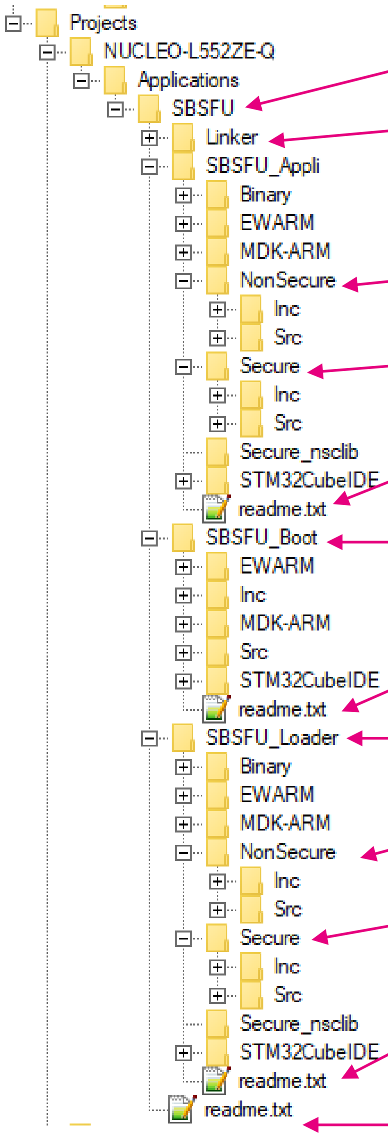


Open Source Crypto library, used by TFM_SBSFU_Boot, TFM_Appli Secure application, and SBSFU_Boot

Open Source MCUboot middleware, used by TFM_SBSFU_Boot and SBSFU_Boot

Open source TF-M middleware, used by TFM_Appli

SBSFU-STM32L5 package



```
Projects
  NUCLEO-L552ZE-Q
    Applications
      SBSFU
        Linker
        SBSFU_Appli
          Binary
          EWARM
          MDK-ARM
          NonSecure
            Inc
            Src
          Secure
            Inc
            Src
          Secure_nsclib
          STM32CubeIDE
          readme.txt
        SBSFU_Boot
          EWARM
          Inc
          MDK-ARM
          Src
          STM32CubeIDE
          readme.txt
        SBSFU_Loader
          Binary
          EWARM
          MDK-ARM
          NonSecure
            Inc
            Src
          Secure
            Inc
            Src
          Secure_nsclib
          STM32CubeIDE
          readme.txt
        readme.txt
```

SBSFU Application directory

Memory mapping shared between SBSFU_Boot and SBSFU_Appli applications

Non Secure Application (user application example)

Secure Application (only “secure GPIO toggle” service example)

Secure and Non Secure application Implementation Information

Secure Boot and Secure Firmware Update application

Secure Boot implementation Information

Loader application (Ymodem loader application example)

Non Secure part of Loader Application

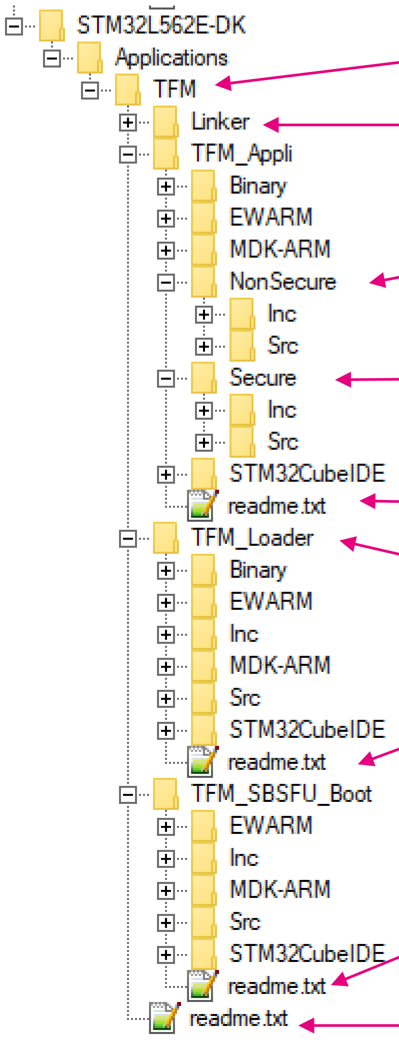
Secure part of the Loader Application (required for primary only slot mode)

Loader application Implementation information

How to prepare the setup and use the SBSFU application

NUCLEO-L552ZE-Q

TFM-STM32L5 package



```
STM32L562E-DK
  Applications
    TFM
      Linker
      TFM_Appli
        Binary
        EWARM
        MDK-ARM
        NonSecure
          Inc
          Src
        Secure
          Inc
          Src
        STM32CubeIDE
        readme.txt
      TFM_Loader
        Binary
        EWARM
        Inc
        MDK-ARM
        Src
        STM32CubeIDE
        readme.txt
      TFM_SBSFU_Boot
        EWARM
        Inc
        MDK-ARM
        Src
        STM32CubeIDE
        readme.txt
      readme.txt
```

TF-M Application directory

Memory mapping shared between TFM_SBSFU_Boot and TFM_Appli

Non Secure Application (code example)

Secure Application (TF-M Core, TFM secure services…)

Secure and Non Secure application Implementation Information

Loader application (Ymodem loader application example)

Loader application Implementation information

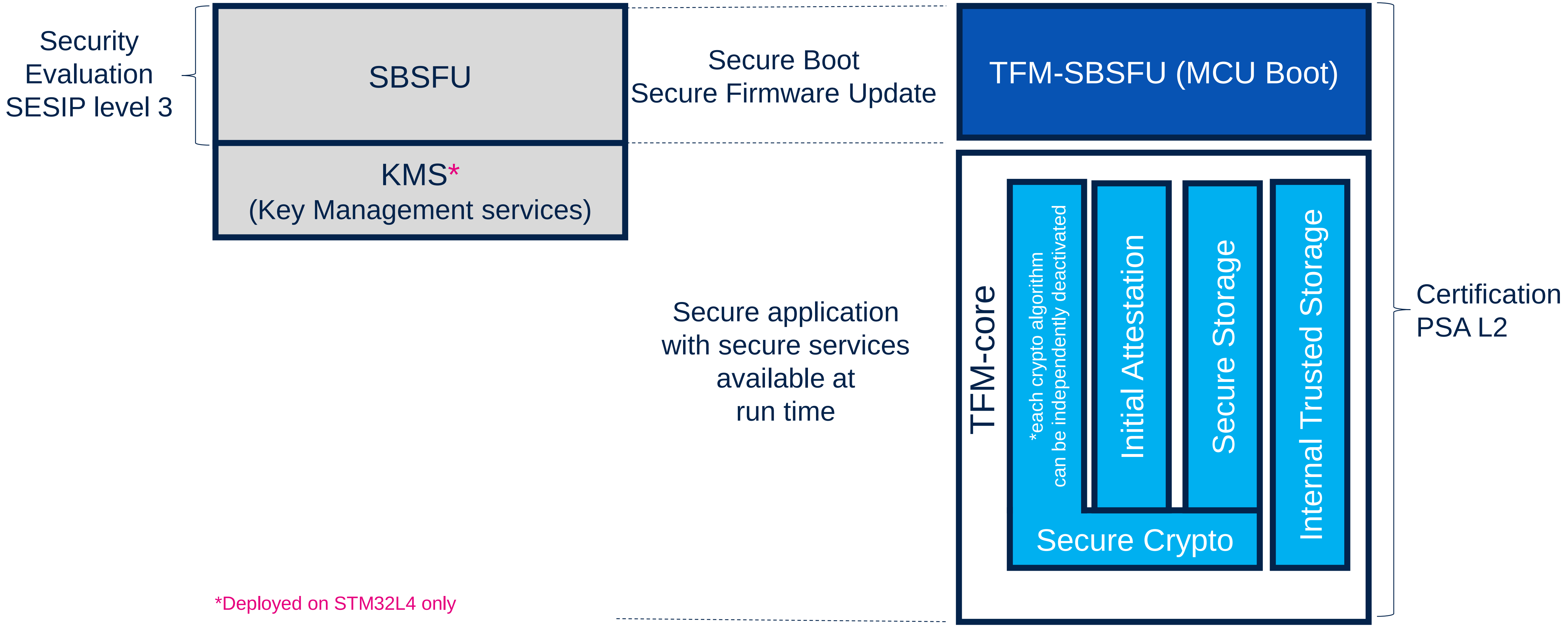
Secure Boot and Secure Firmware Update application

Secure Boot implementation Information

How to prepare the setup and use the TFM applications (Secure Boot and Secure Firmware Update application, Secure application, Non Secure application)

X-CUBE-SBSFU versus TFM

Certifications



Security Evaluation SESIP level 3

SBSFU

KMS*
(Key Management services)

Secure Boot
Secure Firmware Update

TFM-SBSFU (MCU Boot)

Secure application with secure services available at run time

TFM-core

*each crypto algorithm can be independently deactivated

Initial Attestation

Secure Storage

Internal Trusted Storage

Secure Crypto

Certification PSA L2

*Deployed on STM32L4 only

Useful Links

- UM2671 : Getting started with STM32CubeL5 TFM Application
- AN5447 : Overview of Secure Boot and Secure Firmware Update solution on Arm® TrustZone® STM32L5 Series microcontrollers
- https://developer.arm.com/architectures/security-architectures
- https://www.trustedfirmware.org/about/

Conclusion

- TFM is a secure boot with secure firmware update capabilities but its also allows many security services during run-time. It’s relying on PSA standard define by ARM and is open source based project.
- STM32L5 TFM implementation show usage of all the STM32L5 security features to achieve a PSA certification level 2.

Thank you

life.augmented