Hello and welcome to this presentation describing the STM32H5 product life cycle.
The product life-cycle allows to control access to different assets (code and data) of the product, including during development, manufacturing, and after sales.
It allows to provision the product with different distribution models taking care on the code and data provisioned.
This presentation is part of a set of presentations, that describe the security mechanisms offered by the STM32H5 and related secure firmware.

Agenda

- # Multi-OEM distribution use cases
- # Main PRODUCT_STATES
- # Debug Authentication PRODUCT_STATES

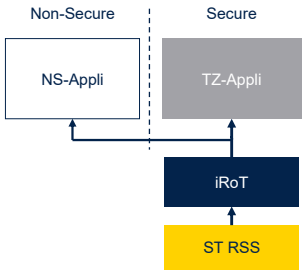
life.augmented

2

The goal of this presentation is to provide some details on the usage of the PRODUCT_STATE, including the Multi-OEM provisioning aspect, and the Debug Authentication Control.
The following topics are going to be explained:
First, the multi-OEM model that enables firmware images coming from different OEMs to be installed securely.
Second the various states of the life cycle will be described.
At last, debug authentication according to the current product state will be explained, particularly in case of field return.

Multi-OEMs Model

- The STM32H5 product lifecycle is designed to support up to three main third parties
  - Only one third party
    - All the product can be configured in only one step
      - Product set to **Closed** or **Locked**
  - Two third parties
    - In that case different splits are possible
      - **iROT** then **(TrustZone + Non-Secure)**
      - Or **(iROT + TrustZone)** then **Non-Secure**
  - Three third parties
    - **iROT** then **TrustZone** then **Non-Secure**



life.augmented

The following images have to be provisioned:
- The immutable root of trust (iROT)
- The secure image containing secure kernel and applications
- The non-secure image containing non-secure kernel and applications.

These images can be provided by independent OEMs.
Either one OEM provides the three images.
Or one OEM provides the iROT and secure image, while another one provides the non-secure image
Or one OEM provides the iROT, while another one provides the secure and non-secure images
Or one OEM provides the iROT, another one the secure image and a last one the non-secure image.

The product life cycle is designed to support this model, based on up to three OEMs.
A typical product configuration taking benefit of TrustZone, is shown in the figure.

Multi-OEMs Model

- Device Provisioning
  - Initial Provisioning
    - Initial setup of the product, that will determine who controls the ROT
      - Could be only the **iROT**
        - In that case the product will be set in **iROT-Provisioned** state
      - Could be **iROT+TrustZone**
        - In that case the product will be set in **TZ-Closed** state
      - Could be **iROT+TrustZone+Non-Secure**
  - Updates
    - When several third parties are considered
      - The Initial Provisioning is not complete, then the installed firmware oversees new firmware & keys to install

life.augmented

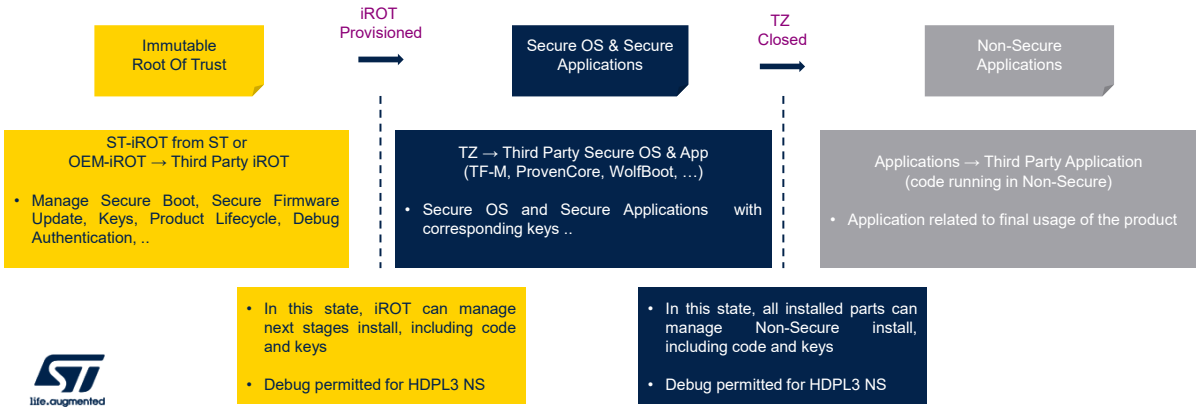
4

The relationship between this model, based on up to three OEMs, and product life cycle is explained on this slide.
Once iROT is provisioned, the product life cycle becomes the iROT-Provisioned state.
Once iROT + Secure image are provisioned, the product life cycle transitions to TrustZone-Closed or TZ-Closed state.
Once iROT + Secure image + non-secure images are provisioned, the product life cycle becomes either the closed or locked state.
This state machine supports backwards transitions only in case of field return, as explained in the following slides.
When several OEMs contribute to install all images, a sequence of installs is required, starting with iROT

provisioning.
The last installed firmware oversees new firmware and keys to install.

Multi-OEMs Model

- ST Lifecycle consider 3 main bricks to be installed in the product that could rely on 3 different parties

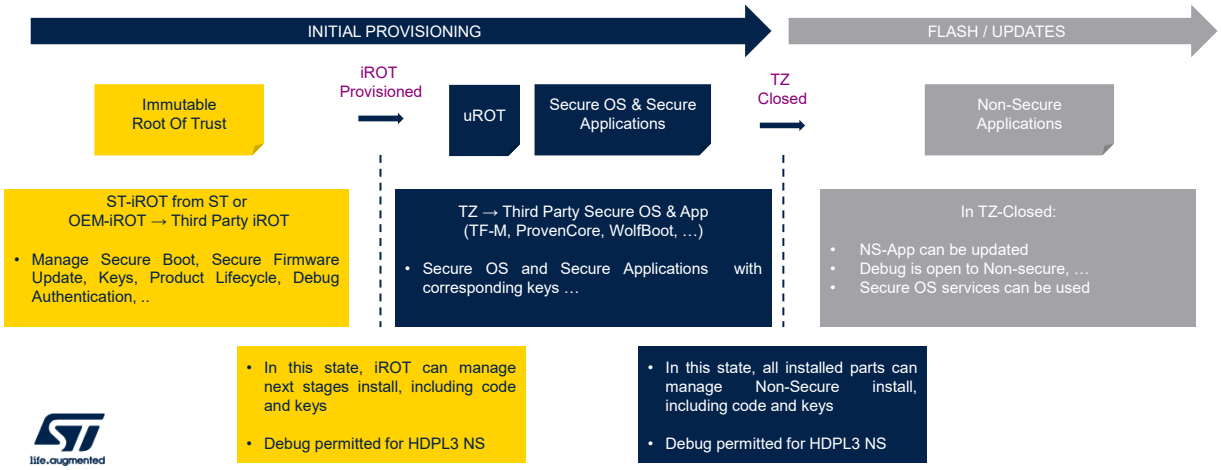


A typical MCU device life-cycle is based on three separate installs.
For each step, the STM32H5 proposes secure life-cycle management mechanisms embedded in the hardware.
First, the Immutable Root-of-Trust (iROT) firmware is programmed.
Second, the secure operating system and secure applications are programmed.
Third, the non secure operating system and applications are programmed.
The installs are chained: iROT offers services to program the secure image and keys and the secure OS offer services to program the non-secure image and keys.
Debug permissions are also reduced all along the

sequence.
When iROT performs the handover to the secure OS, it also restricts the debug permissions to HDPL2 and non-secure.
When secure OS performs the handover to the Non-Secure OS, it also restricts the debug permissions to HDPL3 non-secure.

Multi-OEMs Model

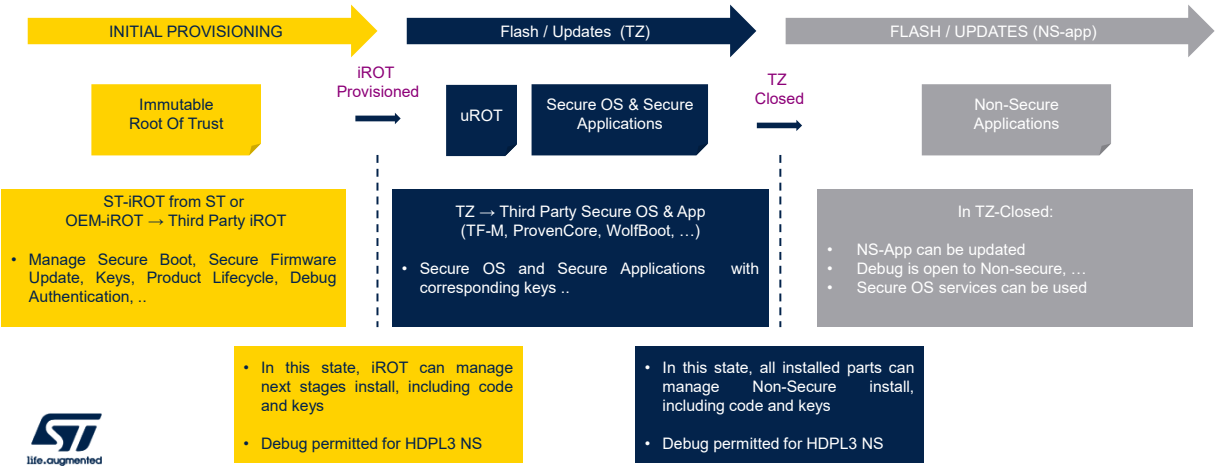
- Example considering 2 third parties: (iROT + TrustZone) then Non-Secure



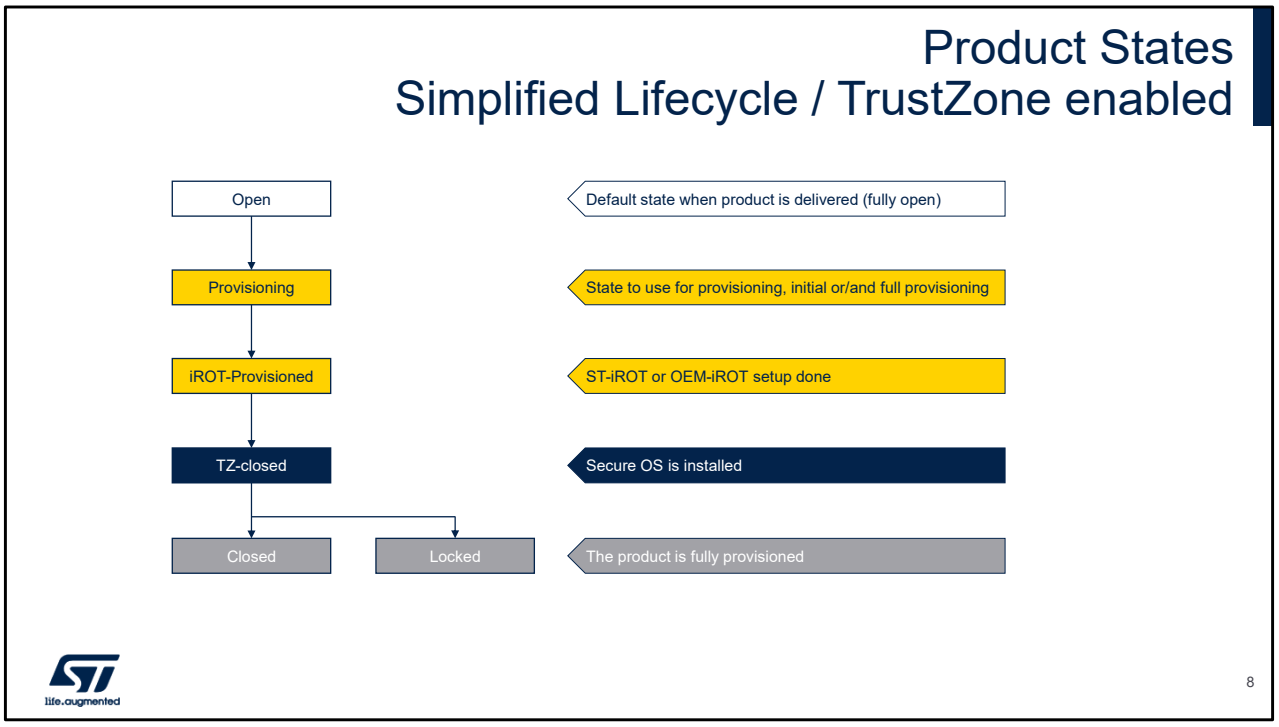
In this example, a first OEM provides both iROT and secure image and a second OEM provides the non-secure image.
Once iROT and secure image are installed, the first OEM transitions the life cycle to TZ-closed state.
Thus, the non-secure image has no visibility on what happened before it is launched.

Multi-OEMs Model

- Example considering 2 third parties: [iROT + TrustZone] then [Non-Secure]



In this example, a first OEM provides the iROT, a second OEM provides the secure image and a third OEM provides the non-secure image.
Once iROT is installed, the first OEM transitions the life cycle to iROT-provisioned state.
Once the secure image is installed, the second OEM transitions the life cycle to TZ-closed state.



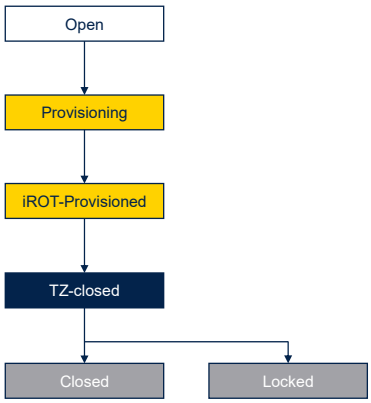
This slide introduces the life cycle state diagram, assuming that trustzone is enabled.
It is simplified, because it does not include the field return additional states.
This is an overview; each state will be detailed in the next slides.
The list of product states from top to bottom is:

- Open, default state when no firmware is installed
- Provisioning
- iROT-Provisioned
- TZ-closed
- Closed
- Locked.

The supported transitions can be requested through the

debug interface or via the system bootloader.

Product States
Simplified Lifecycle / TrustZone enabled



Default state when product is delivered (fully open)

This state is useful for development
- Debug is fully open
- Boot configuration can be SET
  - TZEN = Enabled/Disabled
  - UBE = ST-iROT (only H57x)/OEM-iROT
- HDPL isolation available
- TrustZone isolation available
- Bootloader can be used to provision firmware, Option-Byte keys, …

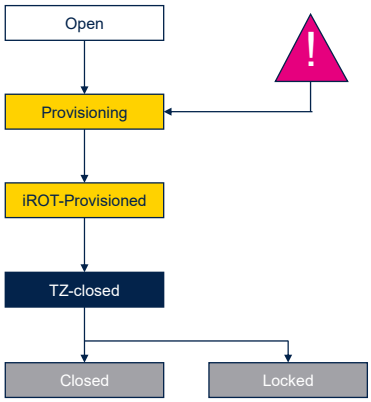
WARNING:
- **RHUK not set (similar to all products)**
- Means provisioning in this state will not be valid for others

life.augmented

9

This slide details the features of the Open state.
It allows to develop the product, as it allows the debug of the code in both secure and non-secure states.
Boot address must target a secure area when TrustZone is enabled.
Using boot pin and Unique Boot Entry (UBE) allow to launch user flash code through ST-iROT.
In this open state, Hide Protect Level (HDPL) and TrustZone isolations are available. They enable to protect data according to current HDPL and secure states.
Since the Root Hardware Unique Key (RHUK) is not activated, provisioning in this state will not be valid of others.

Product States
Simplified Lifecycle / TrustZone enabled



WARNING:
- Debug Authentication only possible if DA-config provisioned
- Take care to provision DA-config!

State to use for provisioning, initial or/and full provisioning

This state is to be used to Provision the device
- RHUK is unique per Device from this state
- **Provisioning MUST be done in this state**
- Debug is only available to HDPL3-NS
- Boot configuration can be set according to
- TZEN = Enabled/Disabled
- UBE = ST-iROT (only H57x)/OEM-iROT
- HDPL isolation available
- TrustZone isolation available
- Bootloader can be used to provision firmware, Option-Byte keys, …
- SFI can be launch in this state, no more in the next

- **WARNING:**
- **The DA-config provisioning MUST be done in this state**

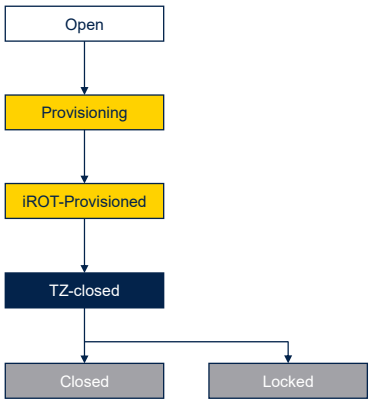
life.augmented

10

The Provisioning state allows to manage the provisioning of the product, partial or full.
The microcontroller is provisioned with ST security services in system flash but can be configured to let the full control of the boot chain to OEMs.
Security services are provisioned by ST in system flash, which is immutable.
They provides the root of trust of the platform managing the verification and the update of the first updatable code (uROT).
The Root Hardware Unique Key (RHUK) is programmed in this state.
Debug authentication is natively supported by STM32H5 platform.

It means that the data used by ST Debug Authentication (ST-DA) must be provisioned at a defined location in option byte keys.
The debug authentication configuration must be done only when the product state is “Provisioning”, it cannot be performed when product state is “Open”.

Product States
Simplified Lifecycle / TrustZone enabled



ST-iROT or OEM-iROT setup done

This state is to be used when almost 2 OEMs are used and one of the OEM manage the iROT

- In this state, the iROT must allow the update of the next level(s): code & data
- Debug is only available to HDPL3-NS
- iROT can launch Bootloader if next level not OK
- Debug Authentication can be launched

life.augmented

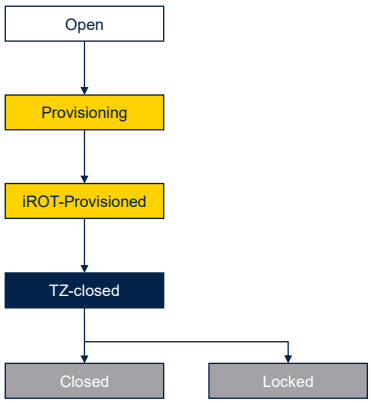
11

The product state becomes iROT-Provisioned, when it is assumed that immutable root of trust is installed, including its configuration: code, option bytes, secure storage.
This state is relevant when a first OEM provisions the iROT and another one provisions the secure and possibly non-secure images.
The iROT allows the update of the image containing both the secure and non-secure images.
It can launch the bootloader if the next level is not OK.
The embedded bootloader is located in the system memory, programmed by ST during production.
It is used to reprogram the flash memory by using USART, I2C, I3C, SPI, FDCAN, or USB_FS in device mode through the Device Firmware Upgrade (DFU).

The Debug Authentication protocol can be launched in iROT-Provisioned product state.
The protocol is based on:

- Initial message: posted by the host combined with a reset to launch the debug authentication process on the device.
- Challenge message: the device generates a random value, to be signed by the host, when sending back the response.
- Response: the host sends a message to the device proving its authenticity. This is done using a tool to generate a token.

Product States
Simplified Lifecycle / TrustZone enabled

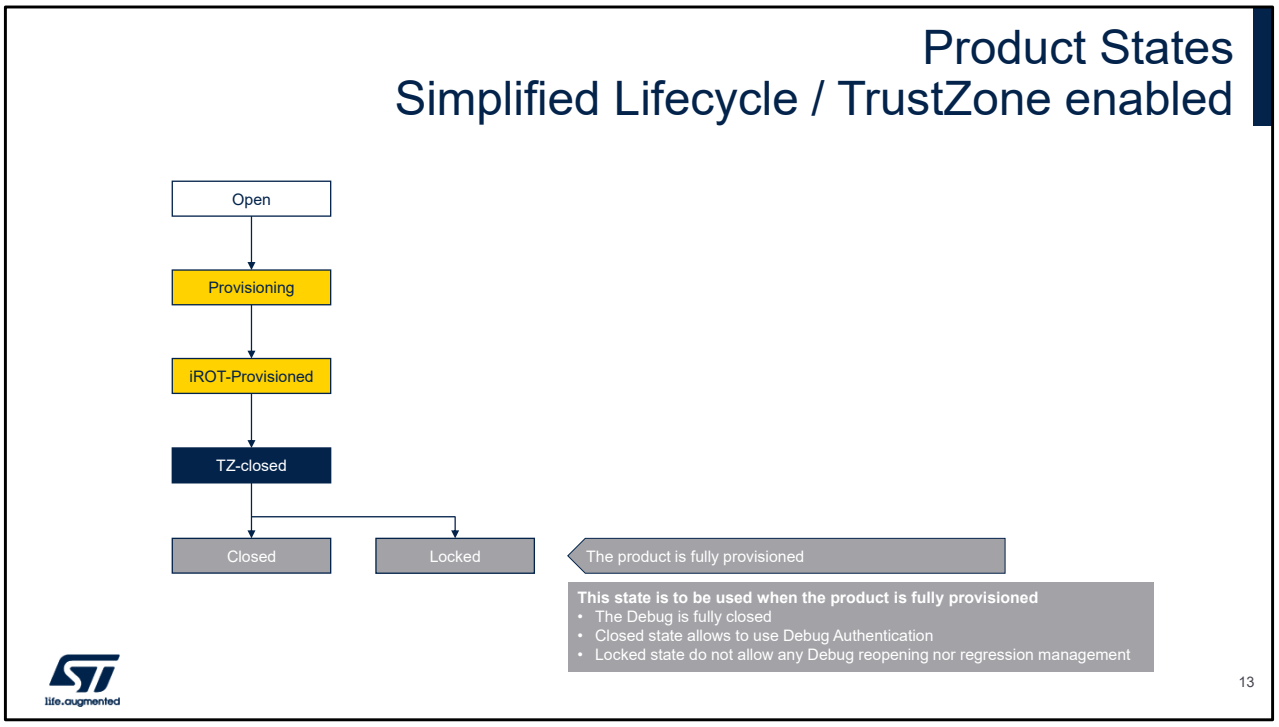


Secure OS is installed

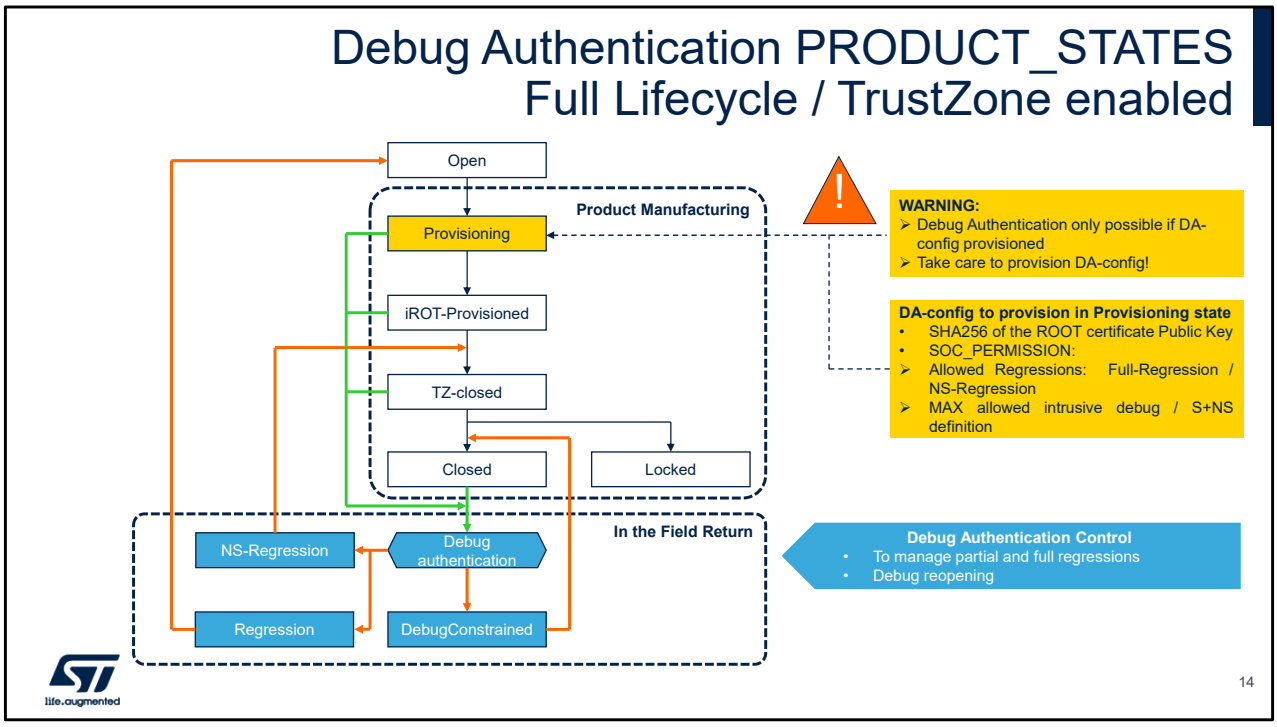
This state is to be used when iROT and (uROT+SecureOS) are provisioned
- In this state, the uROT is supposed to allow the update of the next level(s): code & data
- Debug is only available to HDPL3-NS
- uROT can launch Bootloader if next level not OK
- Debug Authentication can be launched

life.augmented

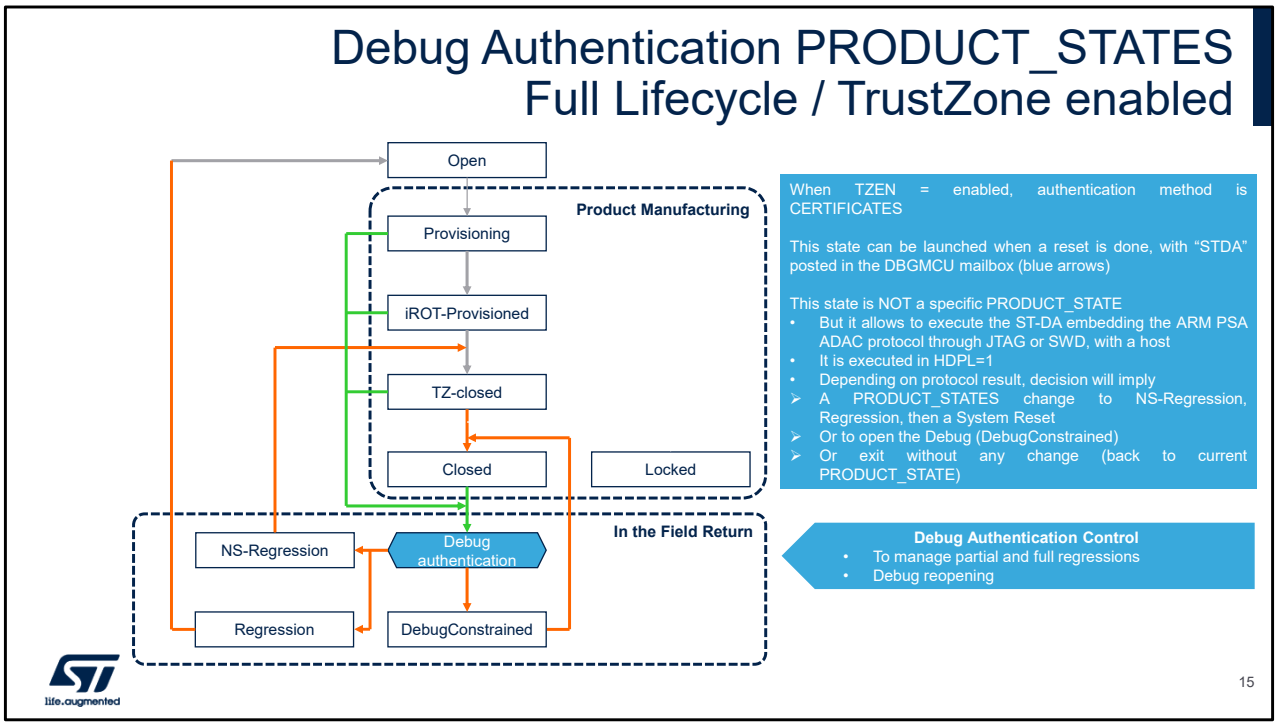
The product state becomes TZ-closed, when it is assumed that the Secure OS is installed in TrustZone.
Debug is opened for non-secure applications.
All peripherals and memories mapped as secure during secure boot cannot be dumped, debugged or traced.
The uROT must allow the update of the non-secure image.
It can launch the bootloader if the next level is not OK.



The product state becomes Closed, when it is assumed that the product configuration is finalized.
No debug is possible except through debug authentication, when field return is enabled.
Progressing to more closed state is the normal product life cycle, that does not require security measures.
Transition in direction to open state is a regression, controlled by the debug authentication control. See explanations hereafter in the presentation.
If debug unlock policy is set to “locked”, no regression is accepted.
Debug Authentication is fully deactivated in Locked state. It means neither regression, nor debug opening are possible

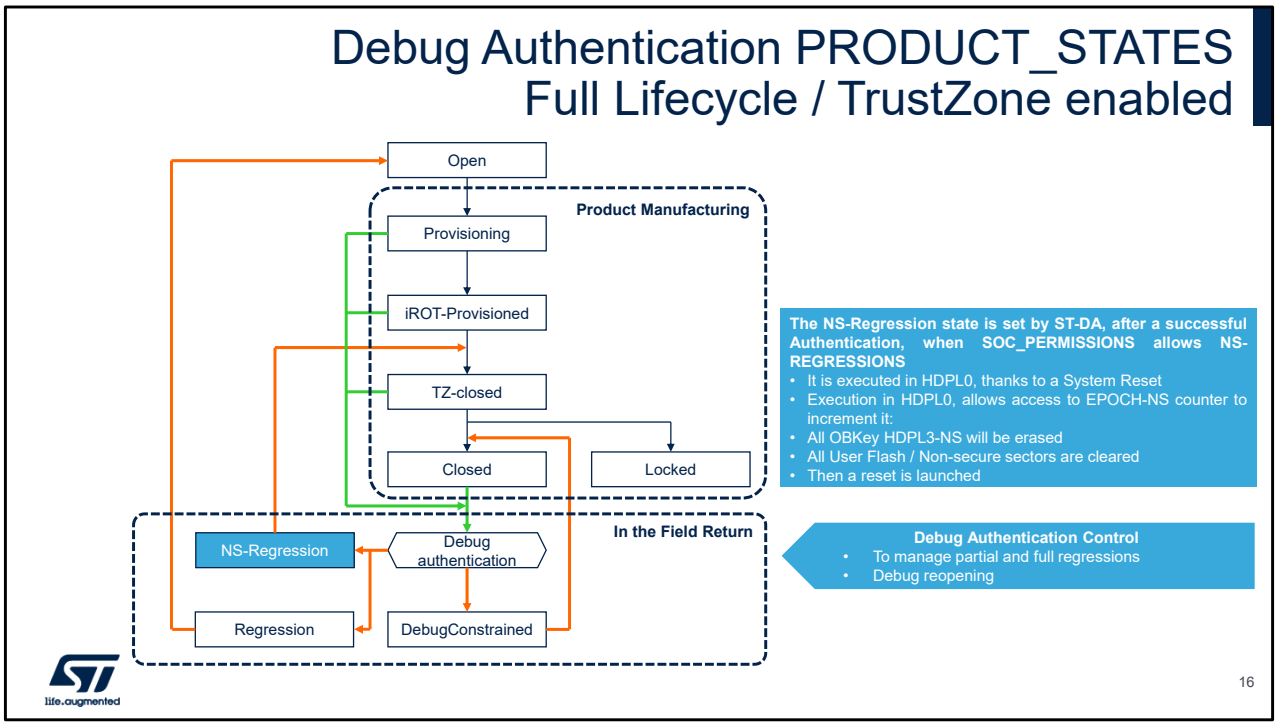


In this figure, all states are represented, including the ones associated with field return.
ST-Debug Authentication (DA) manages the debug authentication control feature, allowing to control the debug reopening and regressions of the product for after sales of it.
Field return is possible only if Debug Authentication configuration has been performed in the Provisioning state.
This configuration includes:
- The SHA256 of the ROOT certificate public key
- SOC permissions that define the allowed regressions, full or non-secure and the maximum allowed intrusive debugs for secure and non-secure states.



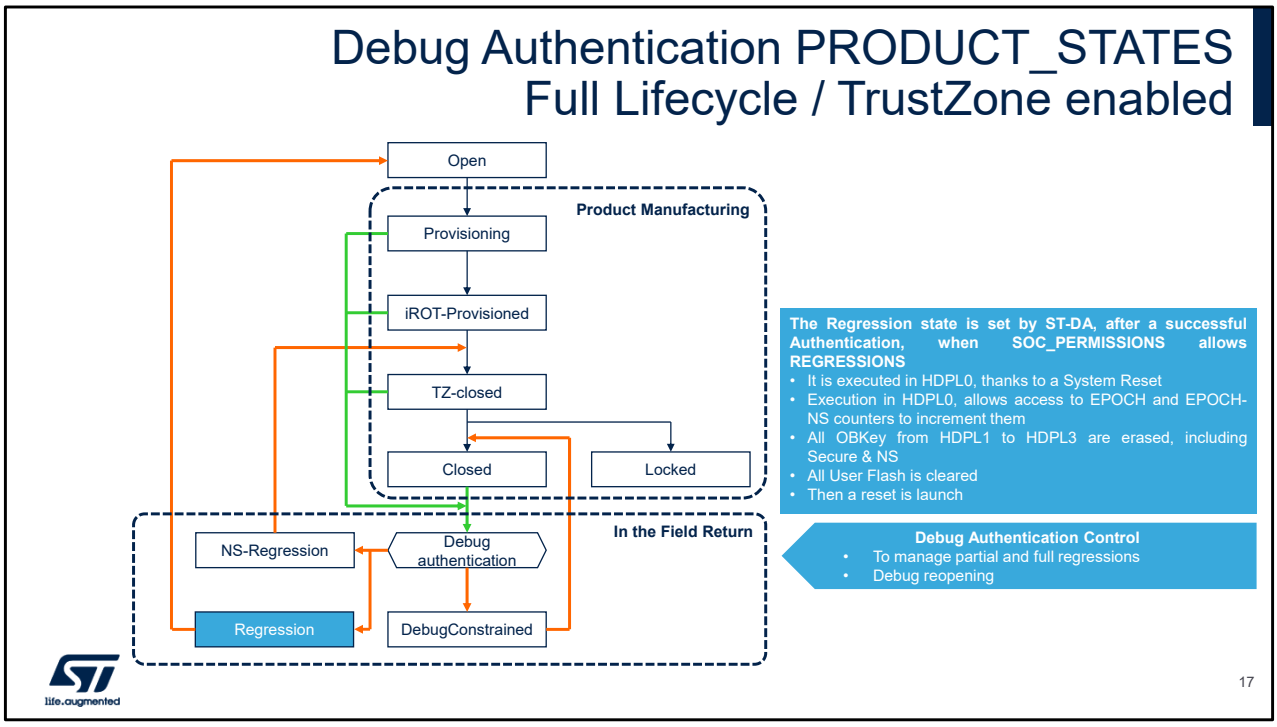
The debug authentication control is ensured thanks to a protocol based on Arm PSA ADAC specification.
Platform Security Architecture (PSA) is a security certification scheme for Internet of Things (IoT) hardware, software and devices.
The Authenticated Debug Access Control (ADAC) specification defines the protocol that allows a target to securely authenticate a debug host.
This Debug Authentication state allows controlling that the host has a trusted certificate with permissions.
Permissions definition allow a lot of flexibility, in the default model ST considers: Full or Partial regression, or to open the debug for non-secure.
The protocol implements a challenge response

mechanism based on asymmetric cryptography to authenticate the host.
It relies on a key pair, with a Public Key stored in the device, and a Private Key from the host library, used to sign a random value (the challenge) generated by the device.
The protocol implements a bidirectional communication between the host and the device through a mailbox interface located in the DBGMCU.



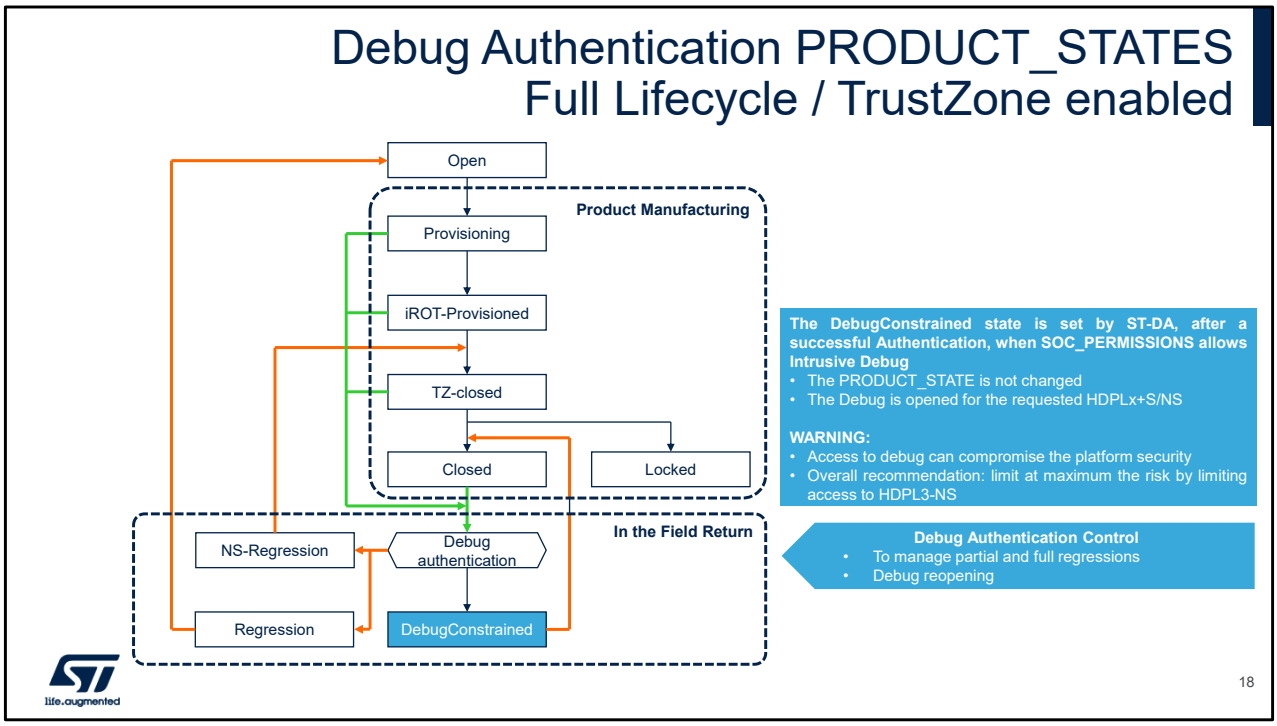
NS-Regression is the temporary state initiated by the debug authentication system in transition to TZ-Closed.
The EPOCH_NS counter is incremented.
This transition opens the device to authorize the debug of the non-secure application software without compromising the security of the ROT functions.
Note that the Option-Byte key related to HDPL3-NS and all user flash non-secure sectors are erased.
The starting state is Closed.
The debug tools are used to authenticate debug regression access rights with the debug authentication library, running on the device in the HDPL1.
After verifying the credentials, they put the device into intermediate state NS-Regression.

From this state, the device regresses securely in HDPL0 to TZ-closed.
If the implementation uses some sectors in EDATA, to allow erasing their state is changed to no-EDATA, erase is done, then reconfigured to EDATA.
As we are not able to distinguish secure from non-secure EDATA sectors, the erase if done whatever the sector is in secure or not.

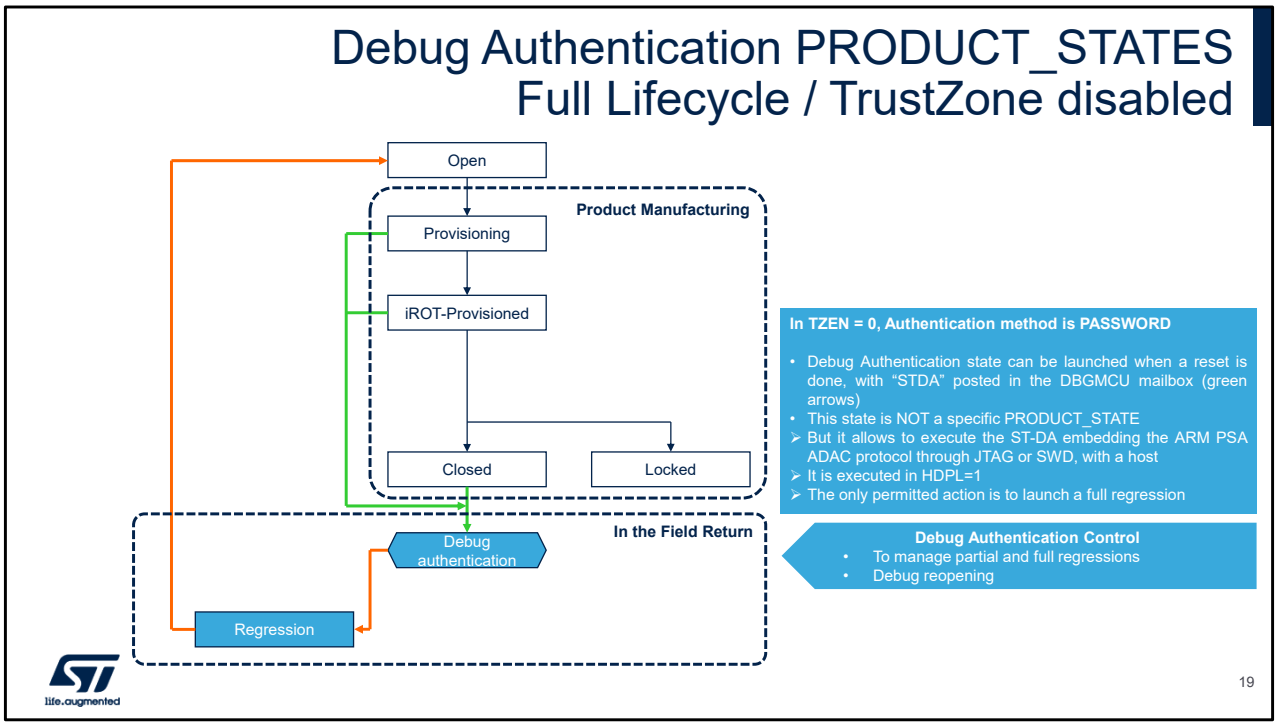


Regression is a temporal state, but non-volatile, to manage the full regression to Open state, removing all user flash code and data including in secure storage, from HDPL1 to HDPL3.
This transition is a full regression.
The EPOCH_NS and EPOCH_S counters are incremented.
Note that the Option-Byte key address ranges related to HDPL1 to HDPL3 and all user flash sectors are erased.
The starting state is any except Open and Locked.
The debug tools are used to authenticate debug regression access rights with the debug authentication library, running on the device in HDPL1.
After verifying the credentials, it puts the device into

intermediate Regression state.
From this state the device regresses securely in HDPL0 to Open.



DebugConstrained is a temporal state, until power-on-reset, to manage debug based on the permissions acquired by the debug authentication protocol.
The product state is not changed.
Debug is re-opened for the requested HDPL and security states.
Access to debug can therefore compromise the platform security. This can be limited by restricting the debug access to HDPL3 non-secure.

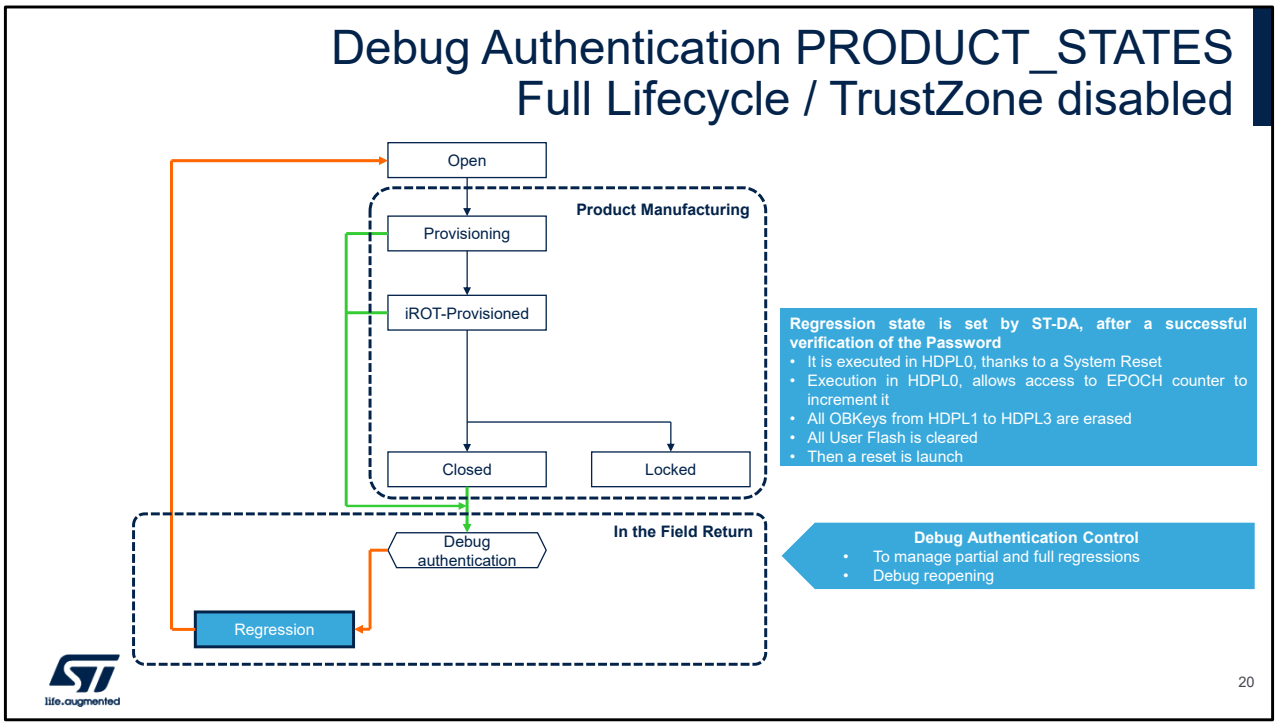


When TrustZone is disabled, the state TZ-Closed does not exist and the debug authentication is based on a password authentication method.

Only the HASH of the password has to be provisioned.

This state allows to execute the ST-DA embedding the ARM PSA ADAC protocol through JTAG or SWD, with a host.

The only permitted action is to launch a full regression.



When TrustZone is disabled, the full regression is done to open state after a successful verification of the password.
The EPOCH_NS counter is incremented.
All option byte keys from HDPL1 to HDPL3 are erased, as well as the complete user flash.

Thank you

© STMicroelectronics - All rights reserved.
ST logo is a trademark or a registered trademark of STMicroelectronics International NV or its affiliates in the EU and/or other countries.
For additional information about ST trademarks, please refer to www.st.com/trademarks.
All other product or service names are the property of their respective owners.

life.augmented

Thanks for having attended this presentation.
You can also refer to the following presentations:

- Security overview
- Secure data storage
- Debug authentication.