

UEFI & EDK II Training

UEFI Capsule Update

tianocore.org

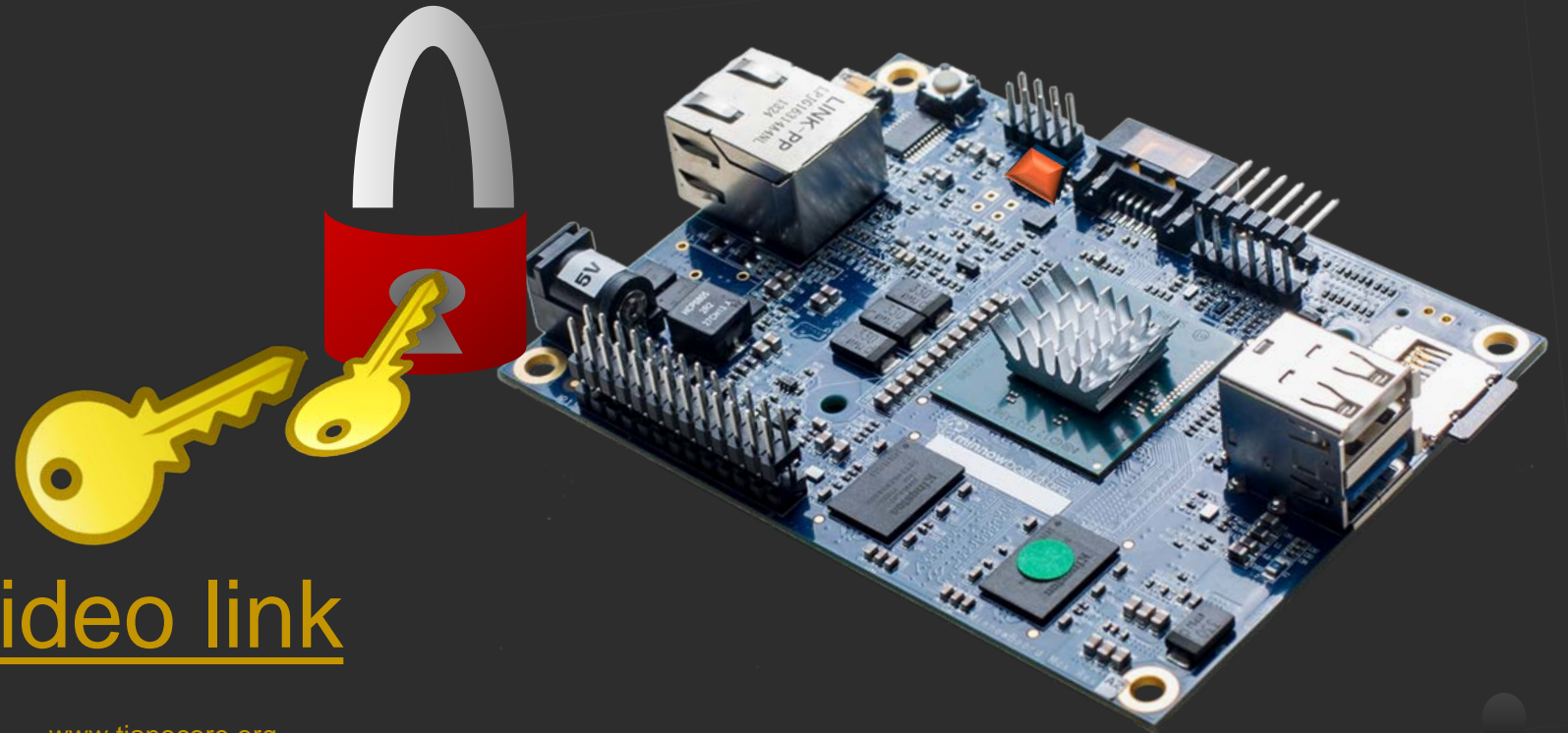
Lesson Objective

- ★ What is Capsule Update
- ★ Why is Capsule Update needed
- ★ How to enable Capsule Update in Edk II platforms

UEFI CAPSULE UPDATE OVERVIEW

What is Capsule Update ?

- A more secure way to update firmware
- OS Agnostic



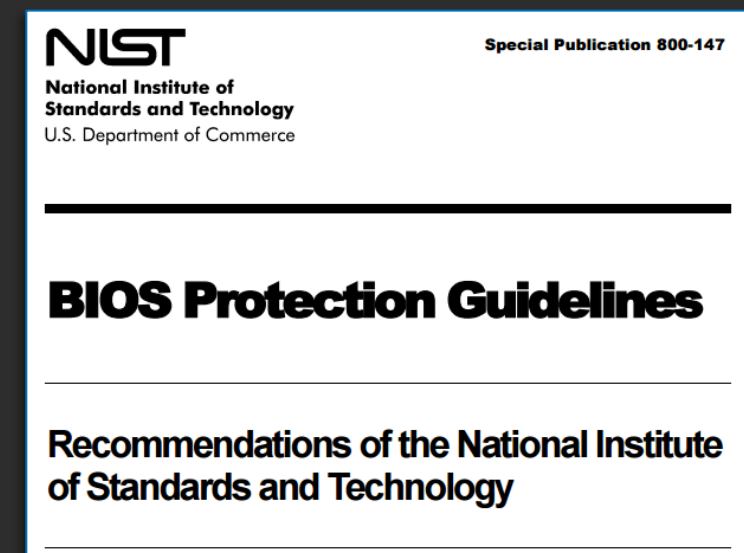
[Video link](#)

Why is Capsule Update Needed?

Establish a **Root-of-Trust** at the low-level platform initialization

National Institute of Standards and Technology (NIST) provides guidelines on BIOS update, [\[800-147\]](#)

- BIOS Update Authentication
- Secure Local Update Method
- Integrity Protection
- Non-Bypassability



NIST : [Nist SP 800-147 .pdf](#)

Why is Capsule Update Needed?

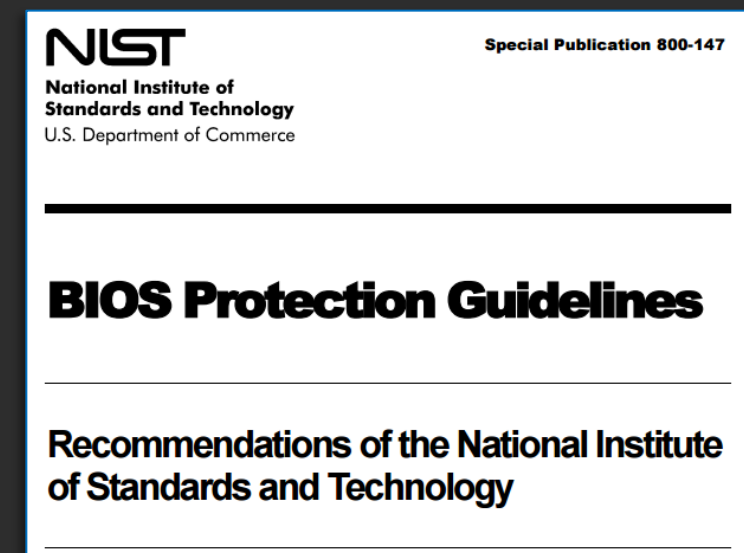
Establish a **Root-of-Trust** at the low-level platform initialization

National Institute of Standards and Technology (NIST) provides guidelines on BIOS update, [\[800-147\]](#)

- BIOS Update Authentication
- Secure Local Update Method
- Integrity Protection
- Non-Bypassability

Does not describe implementation – the

“HOW?”



NIST : [Nist SP 800-147 .pdf](#)

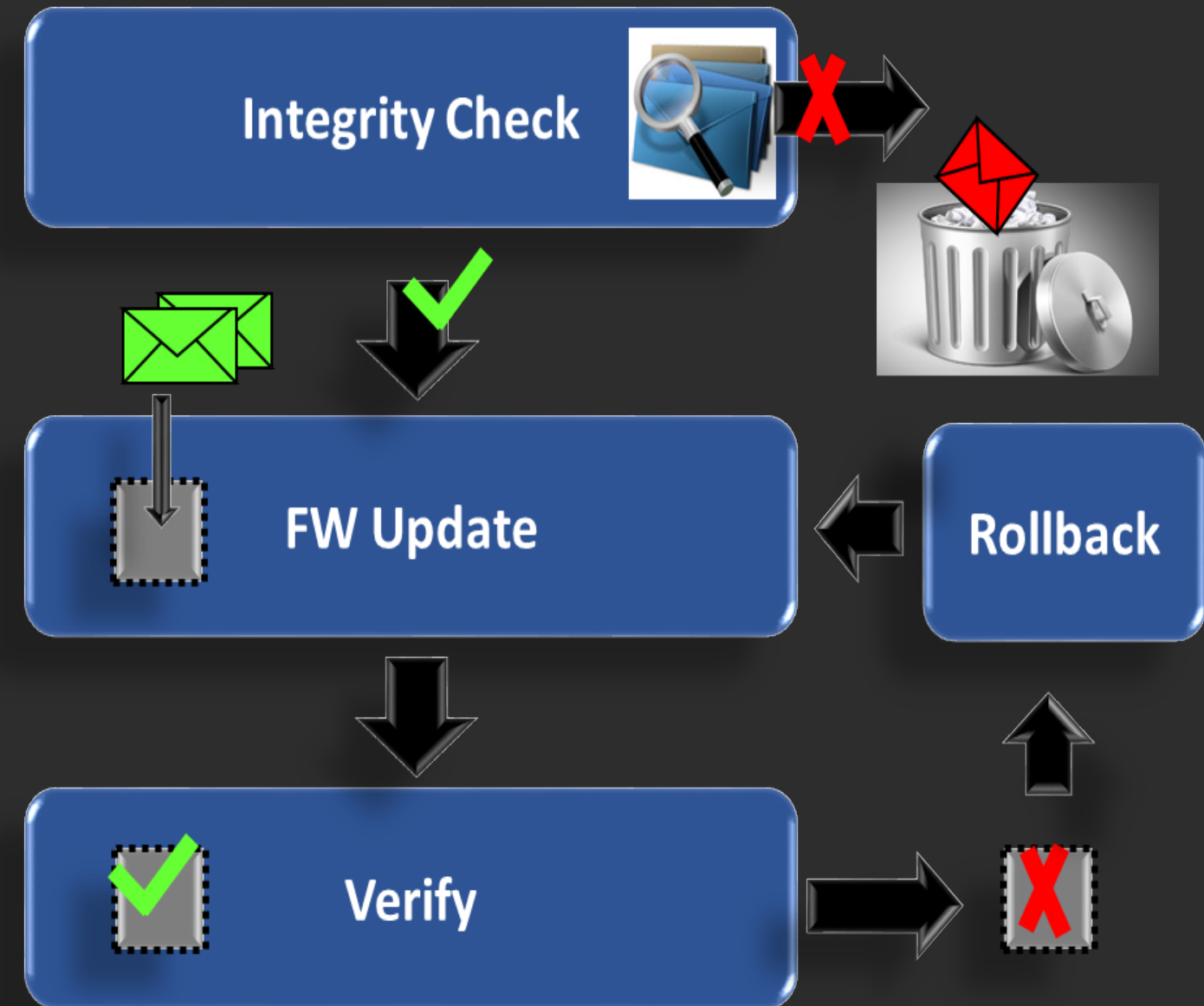
Solving Firmware Update

Reliable update story

- Fault tolerant
- Scalable & repeatable

How can UEFI Help?

- Capsule model for binary delivery
- Bus / Device Enumeration
- Managing updates via
 - EFI System Resource Table
 - Firmware Management Protocol
 - Capsule Signing



HOW DOES THE CAPSULE UPDATE WORK?

UEFI Spec defines Capsule Services to meet NIST Requirement

- **EFI_FIRMWARE_MANAGEMENT_PROTOCOL, (FMP)** capsule format
- **EFI System Resource Table (ESRT)** to support system firmware and device firmware update
- An OS agent may call the UEFI service **UpdateCapsule()** to pass the capsule image from the OS to the firmware. Based upon the capsule flags, the firmware may process the capsule image immediately, or the firmware may reset the system and process the capsule image on the next boot

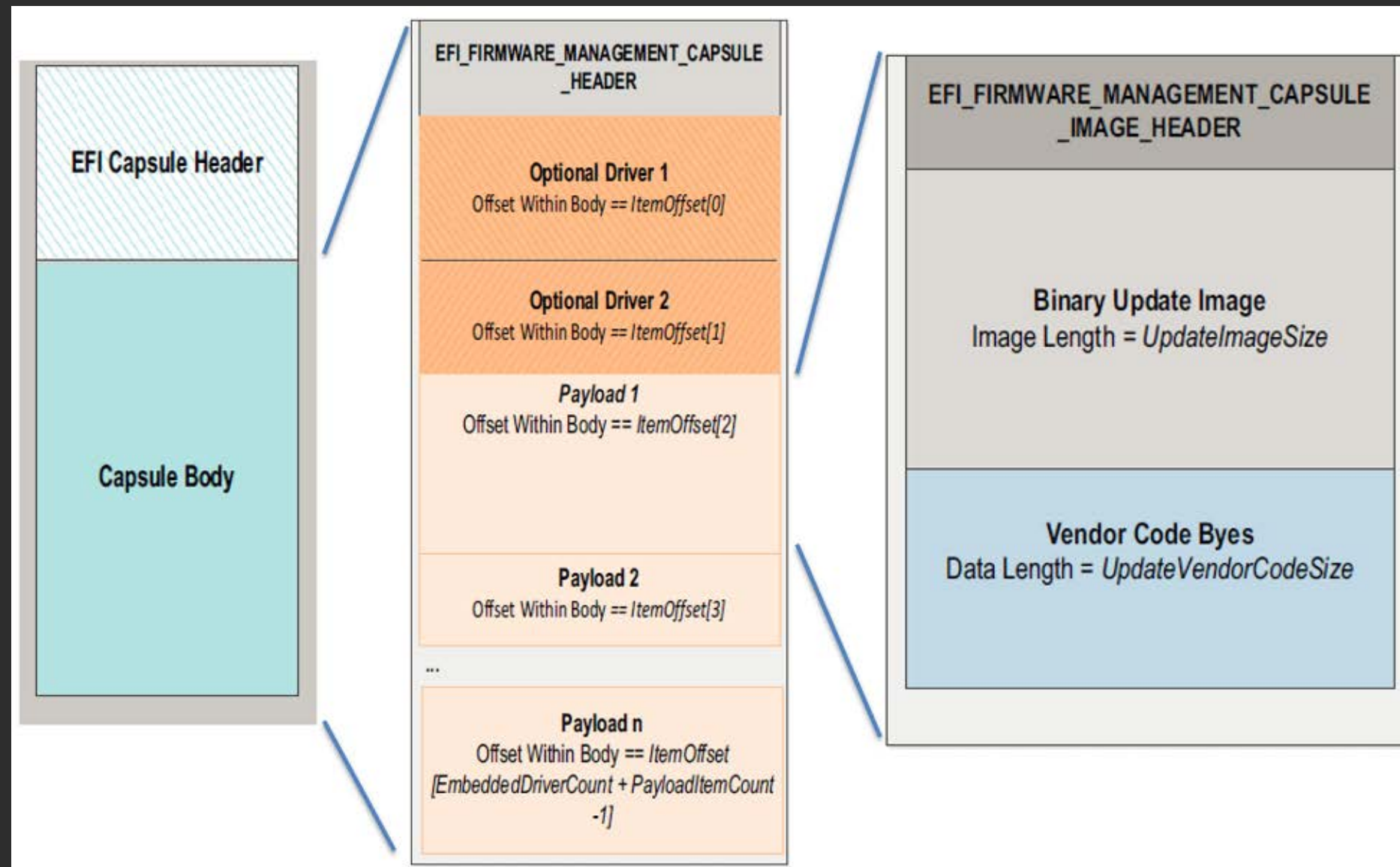
UEFI Capsule Update – Firmware Management Protocol (FMP)

FMP capsule image format

Update FMP drivers

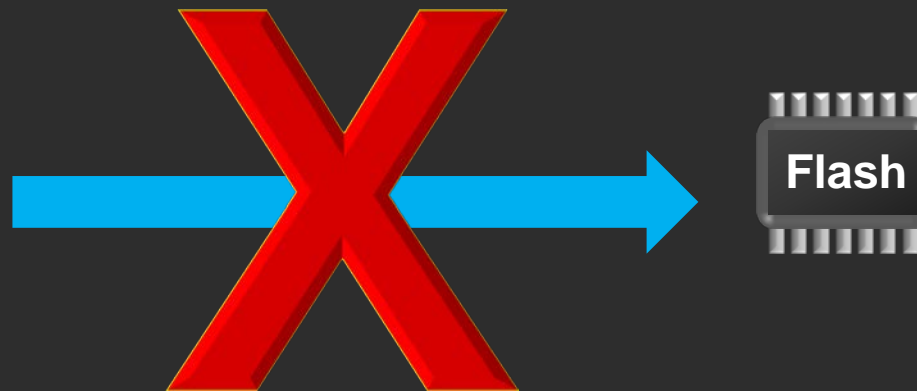
FMP payloads

- binary update image and optional vendor code
- The platform may consume a FMP protocol to update the firmware image



CAPSULE UPDATE - FLOW

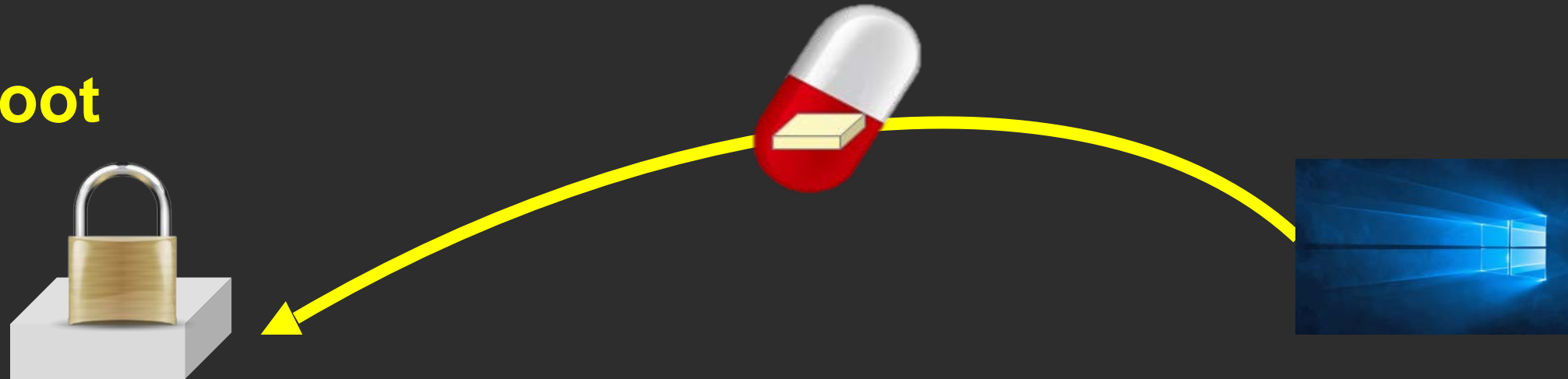
Flash Update



UEFI specification also defines the capsule image format for the EFI_FIRMWARE_MANAGEMENT_PROTOCOL (FMP).

CAPSULE UPDATE - FLOW

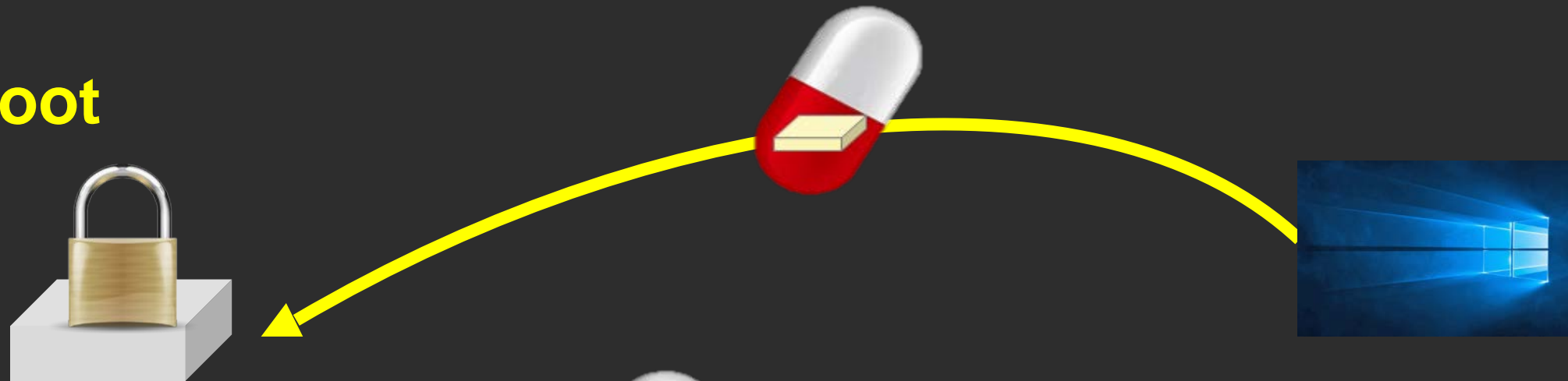
Normal Boot



UEFI specification also defines the capsule image format for the EFI_FIRMWARE_MANAGEMENT_PROTOCOL (FMP).

CAPSULE UPDATE - FLOW

Normal Boot



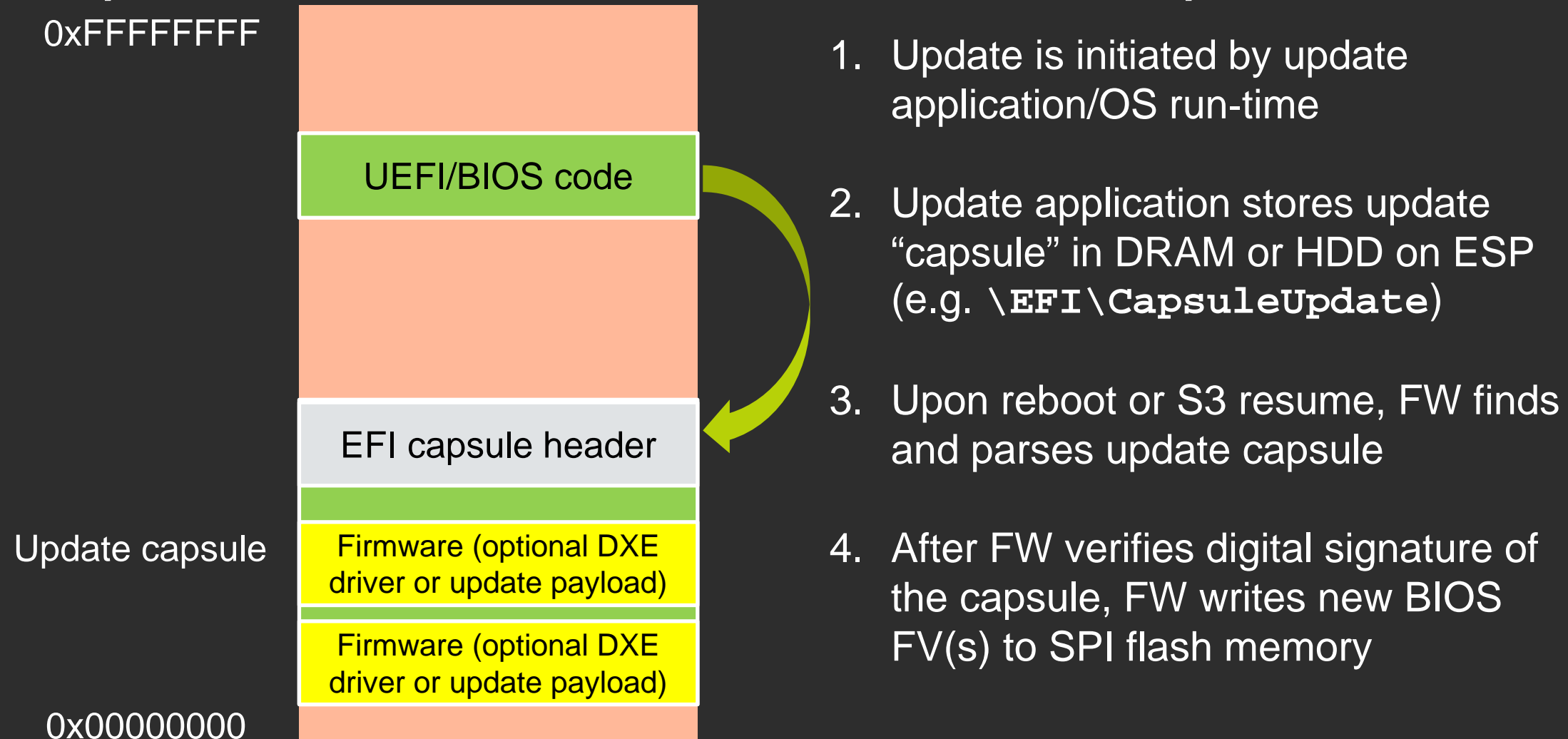
Capsule Update



UEFI specification also defines the capsule image format for the EFI_FIRMWARE_MANAGEMENT_PROTOCOL (FMP).

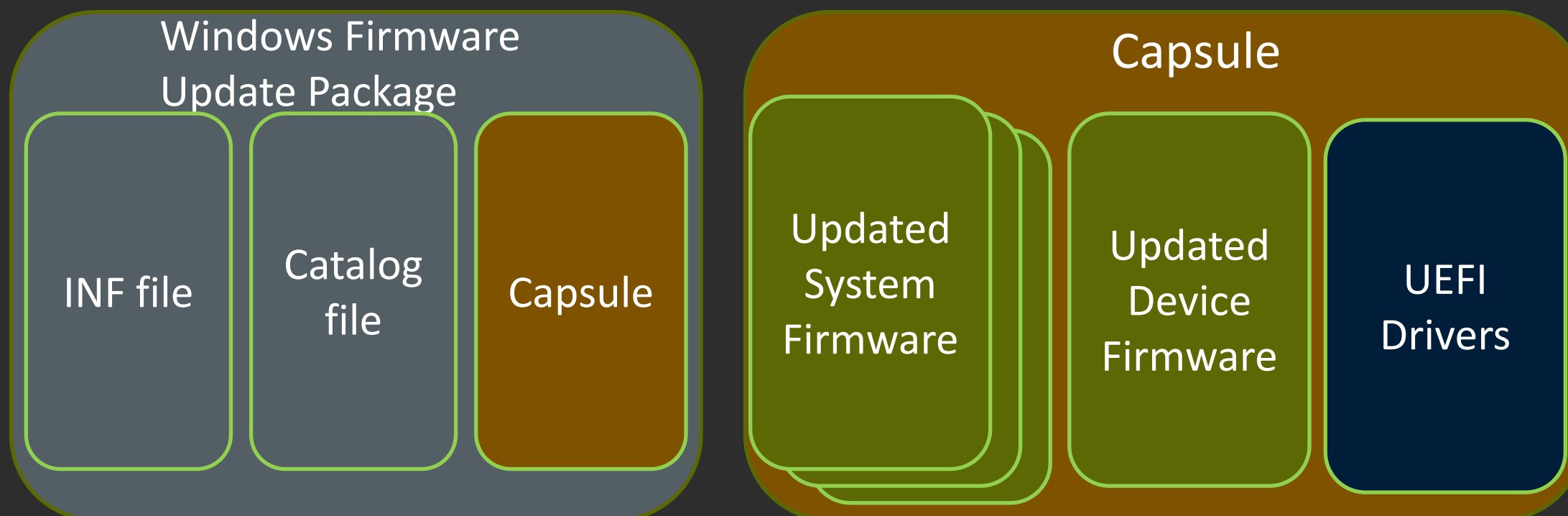
UEFI Firmware Secure “Capsule” Update

Capsule update is a runtime service used to update UEFI FW



Source: UEFI Spec Version 2.4 Facilitates Secure Update UEFI Summerfest – July 15-19, 2013

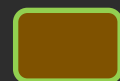
Delivering Update “Capsules” in OS



Color marking:



Microsoft-signed



OEM-signed

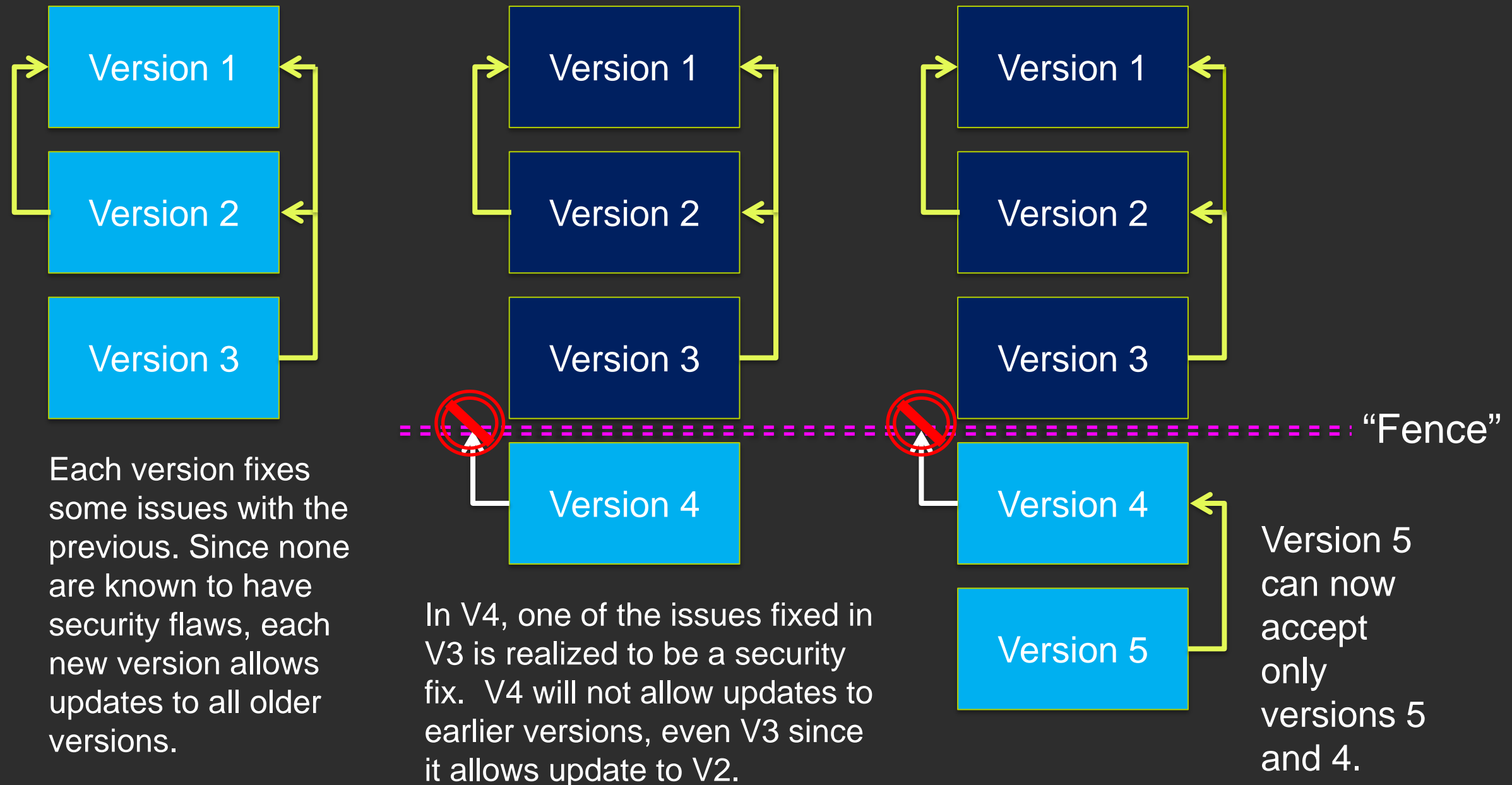


OEM-signed or IHV-signed (device firmware)



Signed with key in UEFI “db” (not applicable to ARM)

Firmware Update Rollback Protection



HOW TO ENABLE?

How to Enable Capsule Update on a EDK II Platform?

UEFI Capsule Implementation in EDK II

SignedCapsulePkg. Uses OpenSSL to sign and authenticate firmware update capsules and firmware recovery images

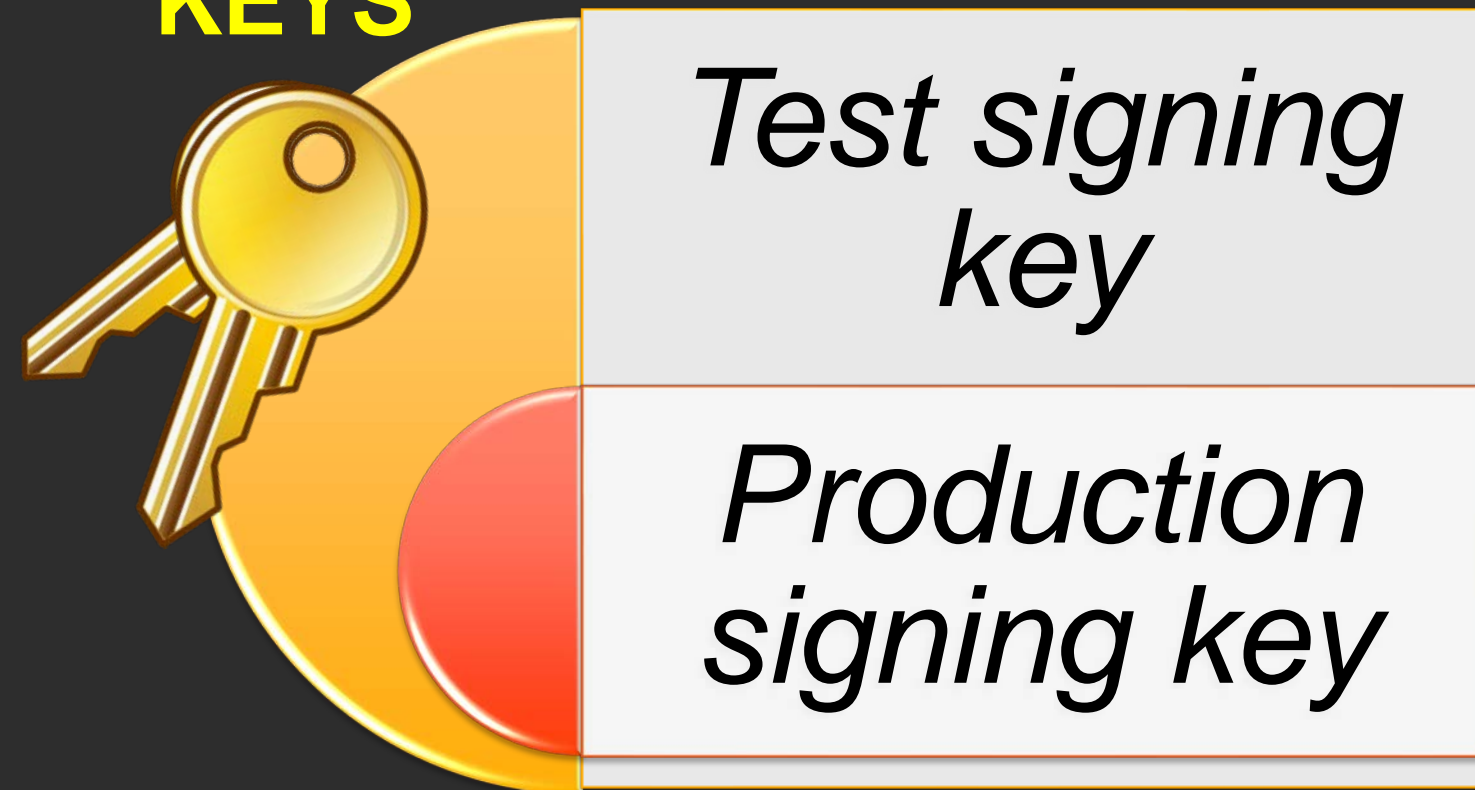
KEYS



UEFI Capsule Implementation in EDK II

SignedCapsulePkg. Uses OpenSSL to sign and authenticate firmware update capsules and firmware recovery images

KEYS



- Used for firmware development and debug
- Used by OEM to create and manage their own
OpenSSL utilities can be used to create key

Enable Capsule Based System Firmware Update

The following wiki pages provide details on how to add the system firmware update using Signed UEFI Capsules

- Implement Platform Components for UEFI Capsule: [Link](#)
- Add `CAPSULE_ENABLE` feature to Platform DSC/FDF Files : [Link](#)
- Verify `CAPSULE_ENABLE` Feature using Test Signing Keys: [Link](#)
- Change System Firmware Update Version : [Link](#)
- Change ESRT System Firmware Update GUID: [Link](#)
- How to Generate Signing Keys using OpenSSL Command Line Utilities: [Link](#)
- Verify `CAPSULE_ENABLE` Feature using Generated Signing Keys: [Link](#)

Implement Platform Components for UEFI Capsule

Requirements for EDK II Projects to support `SignedCapsulePkg`

Library

An instance of `PlatformFlashAccessLib` must be implemented to provide API to update a portion of the non-volatile storage device.

`<Your Platform
Package>/Feature/Capsule/Library/PlatformFlashAccessLib`

Example from Minnowboard Max/Turbot platform: [github link](#)

Implement Platform Components for UEFI Capsule

Requirements for EDK II Projects to support `SignedCapsulePkg`

Library

An instance of `PlatformFlashAccessLib` must be implemented to provide API to update a portion of the non-volatile storage device.

`<Your Platform
Package>/Feature/Capsule/Library/PlatformFlashAccessLib`

Descriptor

PEIM System Firmware Descriptor

`<Your Platform Package>/Feature/Capsule/SystemFirmwareDescriptor`
- Requires `.aslc` (C structure syntax)

Example from Minnowboard Max/Turbot platform: [github link](#)

Implement Platform Components for UEFI Capsule

Requirements for EDK II Projects to support **SignedCapsulePkg**

Library

An instance of `PlatformFlashAccessLib` must be implemented to provide API to update a portion of the non-volatile storage device.

`<Your Platform
Package>/Feature/Capsule/Library/PlatformFlashAccessLib`

Descriptor

PEIM System Firmware Descriptor

`<Your Platform Package>/Feature/Capsule/SystemFirmwareDescriptor`
- Requires `.aslc` (C structure syntax)

Config INI

System Firmware Update Configuration INI File

`<Your Platform
Package>/Feature/Capsule/SystemFirmwareUpdateConfig.ini`

Example from Minnowboard Max/Turbot platform: [github link](#)

Add CAPSULE_ENABLE feature to Platform Files

- Add **-D CAPSULE_ENABLE** to the build command line to enable capsule update features.
- The build process generates a capsule update image (.cap file) along with the UEFI application CapsuleApp.efi.
 - Copy .cap file and CapsuleApp.efi to USB thumb drive.
 - Boot to UEFI Shell and use CapsuleApp.efi with .cap signed capsule file.
- Once the system is rebooted, the signed capsule is authenticated, and the firmware is update with the new system firmware version.

Platform DSC Sections:

```
[LibraryClasses]  
[Pcds]  
[Components]
```

Platform FDF Sections:

```
[FV] PEI  
[FV] DXE  
[FV] Platform  
[VmpPayload]  
[Capsule]  
[Rule]
```

Verify CAPSULE_ENABLE Feature w/ Test Signing Keys

- Download the OpenSSL library
- Build the Boot Firmware image with CAPSULE_ENABLE
- Copy the CapsuleApp.efi to USB thumb drive and run on Target system

```
FS0:\> CapsuleApp.efi
```

```
CapsuleApp:  usage
```

```
  CapsuleApp <Capsule...>
```

```
  CapsuleApp -S
```

```
  CapsuleApp -C
```

```
  CapsuleApp -P
```

```
  CapsuleApp -E
```

```
  CapsuleApp -G <BMP> -O <Capsule>
```

```
  CapsuleApp -N <Capsule> -O <NestedCapsule>
```

```
  CapsuleApp -D <Capsule>
```

```
Parameter:
```

```
-S:  Dump capsule report variable (EFI_CAPSULE_REPORT_GUID),  
      which is defined in UEFI specification.
```

```
-C:  Clear capsule report variable (EFI_CAPSULE_RPORT_GUID),  
      which is defined in UEFI specification.
```

```
-P:  Dump UEFI FMP protocol info.
```

```
-E:  Dump UEFI ESRT table info.
```

```
-G:  Convert a BMP file to be a UX capsule,  
      according to Windows Firmware Update document
```

```
-N:  Append a Capsule Header to an existing capsule image,  
      according to Windows Firmware Update document
```

```
-O:  Output new Capsule file name
```

```
-D:  Dump Capsule image header information and FMP header  
      information, if it is an FMP capsule
```

Verify CAPSULE_ENABLE with CapsuleApp -P option

```
FS0:\> CapsuleApp.efi -P
```

```
#####
```

```
# FMP DATA #
```

```
#####
```

```
FMP (0) ImageInfo:
```

```
DescriptorVersion - 0x3
```

```
DescriptorCount - 0x1
```

```
DescriptorSize - 0x70
```

```
PackageVersion - 0xFFFFFFFF
```

```
PackageVersionName - "Unknown"
```

```
ImageDescriptor (0)
```

```
ImageIndex - 0x1
```

```
ImageTypeId - 4096267B-DA0A-42EB-B5EB-FEF31D207CB4
```

```
ImageId - 0x64465F5F32564C56
```

```
ImageIdName - "Vlv2Fd"
```

```
Version - 0x2
```

```
VersionName - "0x00000002"
```

```
Size - 0x800000
```

```
AttributesSupported - 0xF
    IMAGE_UPDATABLE - 0x1
    RESET_REQUIRED - 0x2
    AUTHENTICATION_REQUIRED - 0x4
    IN_USE - 0x8
    UEFI_IMAGE - 0x0
AttributesSetting - 0xF
    IMAGE_UPDATABLE - 0x1
    RESET_REQUIRED - 0x2
    AUTHENTICATION_REQUIRED - 0x4
    IN_USE - 0x8
    UEFI_IMAGE - 0x0
Compatibilities - 0x0
    COMPATIB_CHECK_SUPPORTED - 0x0
LowestSupportedImageVersion - 0x1
LastAttemptVersion - 0x0
LastAttemptStatus - 0x0
HardwareInstance - 0x0
FMP (0) PackageInfo - Unsupported
```


Verify CAPSULE_ENABLE with CapsuleApp -E option

```
FS0:\> CapsuleApp.efi -E
#####
# ESRT TABLE #
#####
EFI_SYSTEM_RESOURCE_TABLE:
FwResourceCount      - 0x1
FwResourceCountMax   - 0x40
FwResourceVersion     - 0x1
EFI_SYSTEM_RESOURCE_ENTRY (0):
  FwClass              - 4096267B-DA0A-42EB-B5EB-FEF31D207CB4
  FwType               - 0x1 (SystemFirmware)
  FwVersion            - 0x2
  LowestSupportedFwVersion - 0x1
  CapsuleFlags         - 0x1
    PERSIST_ACROSS_RESET - 0x0
    POPULATE_SYSTEM_TABLE - 0x0
    INITIATE_RESET      - 0x0
  LastAttemptVersion   - 0x0
  LastAttemptStatus    - 0x0 (Success)
```

Summary

- What is Capsule Update
- Why is Capsule Update needed
- How to enable Capsule Update in Edk II platforms

Questions?



Return to Main Training Page



Return to Training Table of contents for next presentation [link](#)



ACKNOWLEDGEMENTS

Redistribution and use in source (original document form) and 'compiled' forms (converted to PDF, epub, HTML and other formats) with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code (original document form) must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of this file unmodified.

Redistributions in compiled form (transformed to other DTDs, converted to PDF, epub, HTML and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS DOCUMENTATION IS PROVIDED BY TIANOCORE PROJECT "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL TIANOCORE PROJECT BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright (c) 2021, Intel Corporation. All rights reserved.