presented by

System Firmware and Device Firmware Updates using Unified Extensible Firmware Interface (UEFI) Capsules

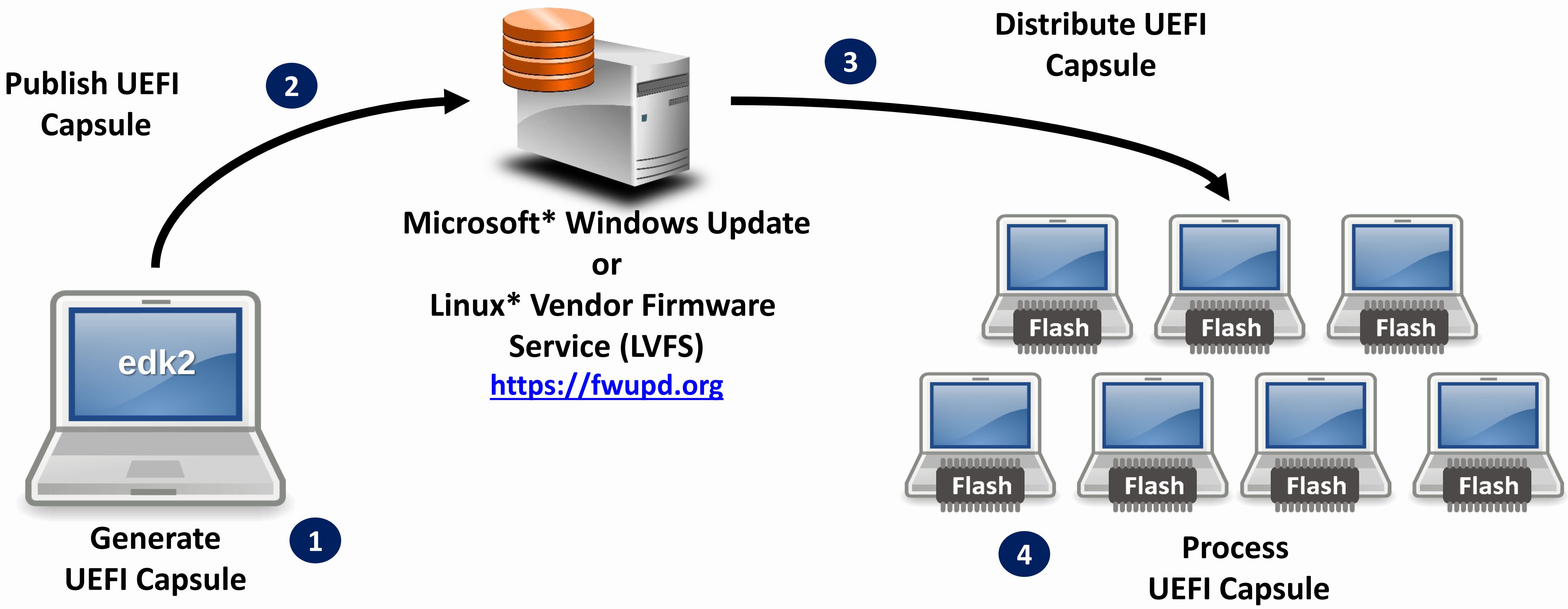
Fall 2018 UEFI Plugfest

October 15 – 19, 2018

Presented by Brian Richardson (Intel)

Materials by Michael Kinney (Intel)

Building and Distributing UEFI Capsules for Firmware Update



1 Generate UEFI Capsule

2 Publish UEFI Capsule

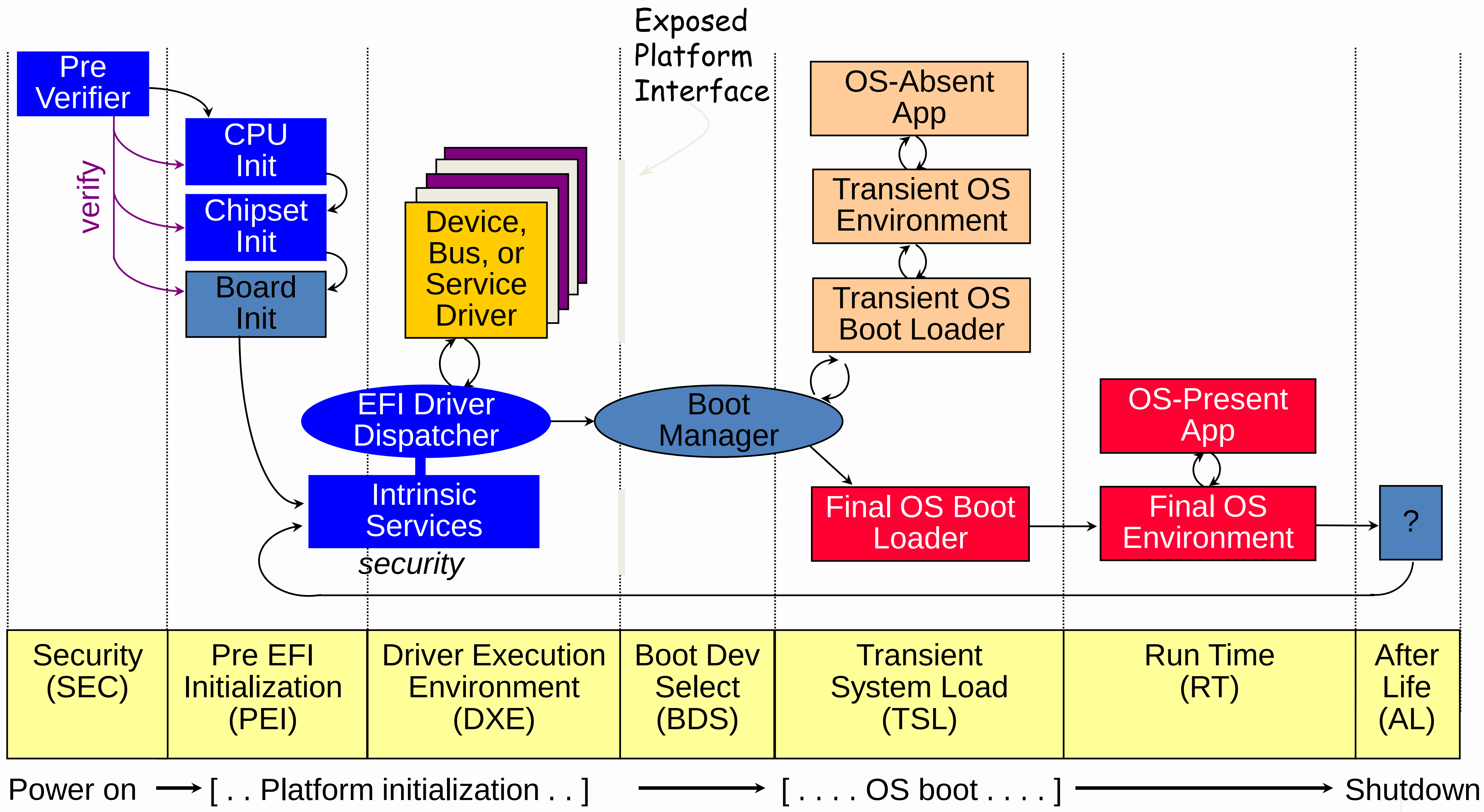
Microsoft* Windows Update
or
Linux* Vendor Firmware Service (LVFS)
https://fwupd.org

3 Distribute UEFI Capsule

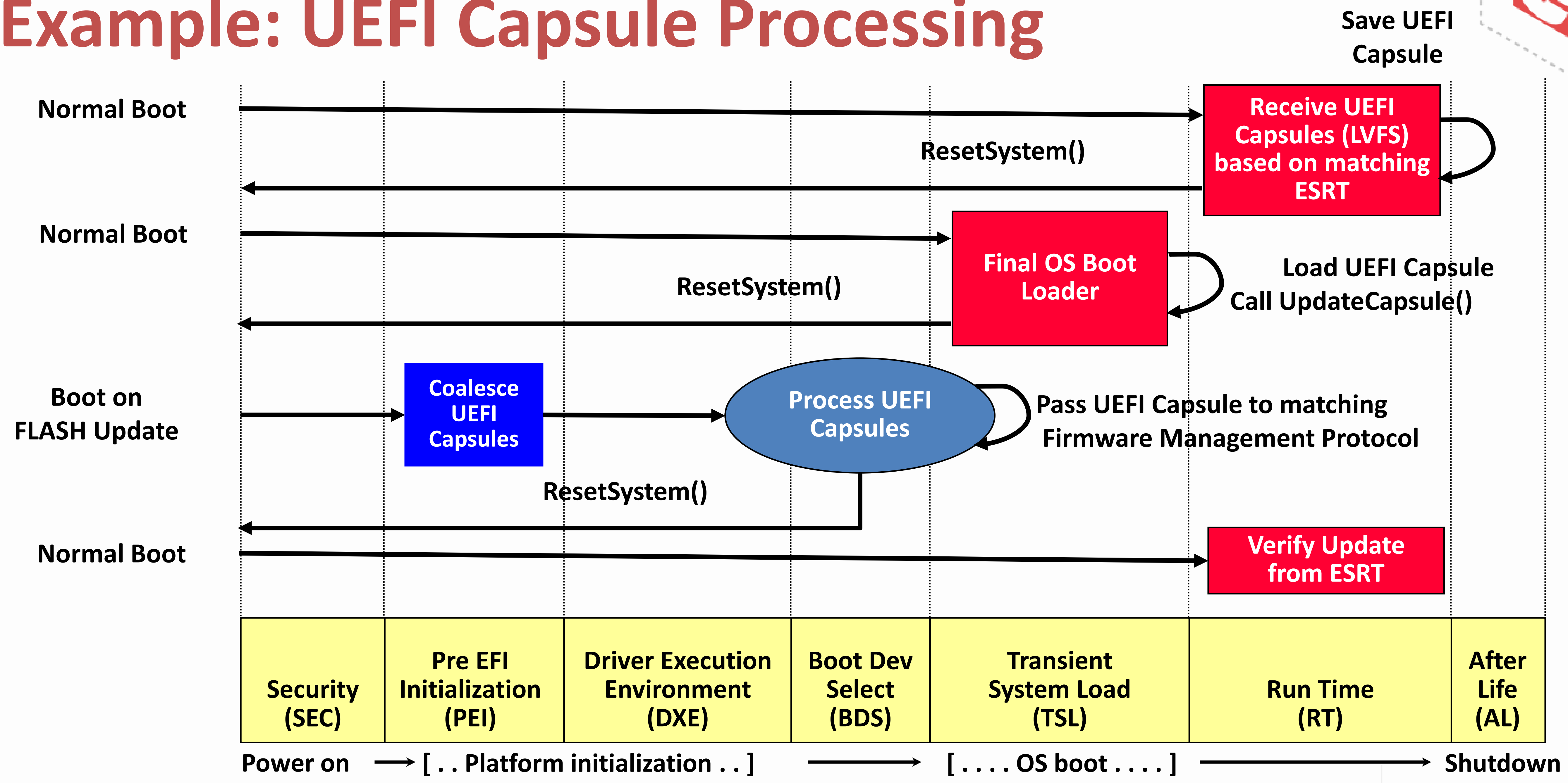
4 Process UEFI Capsule

* Other names and brands may be claimed as property of others

Platform Initialization (PI) Architecture Firmware Phases



PI Architecture Firmware Phases Example: UEFI Capsule Processing



Save UEFI Capsule

Normal Boot

Receive UEFI Capsules (LVFS) based on matching ESRT

ResetSystem()

Normal Boot

Final OS Boot Loader

Load UEFI Capsule
Call UpdateCapsule()

ResetSystem()

Boot on FLASH Update

Coalesce UEFI Capsules

Process UEFI Capsules

Pass UEFI Capsule to matching Firmware Management Protocol

ResetSystem()

Normal Boot

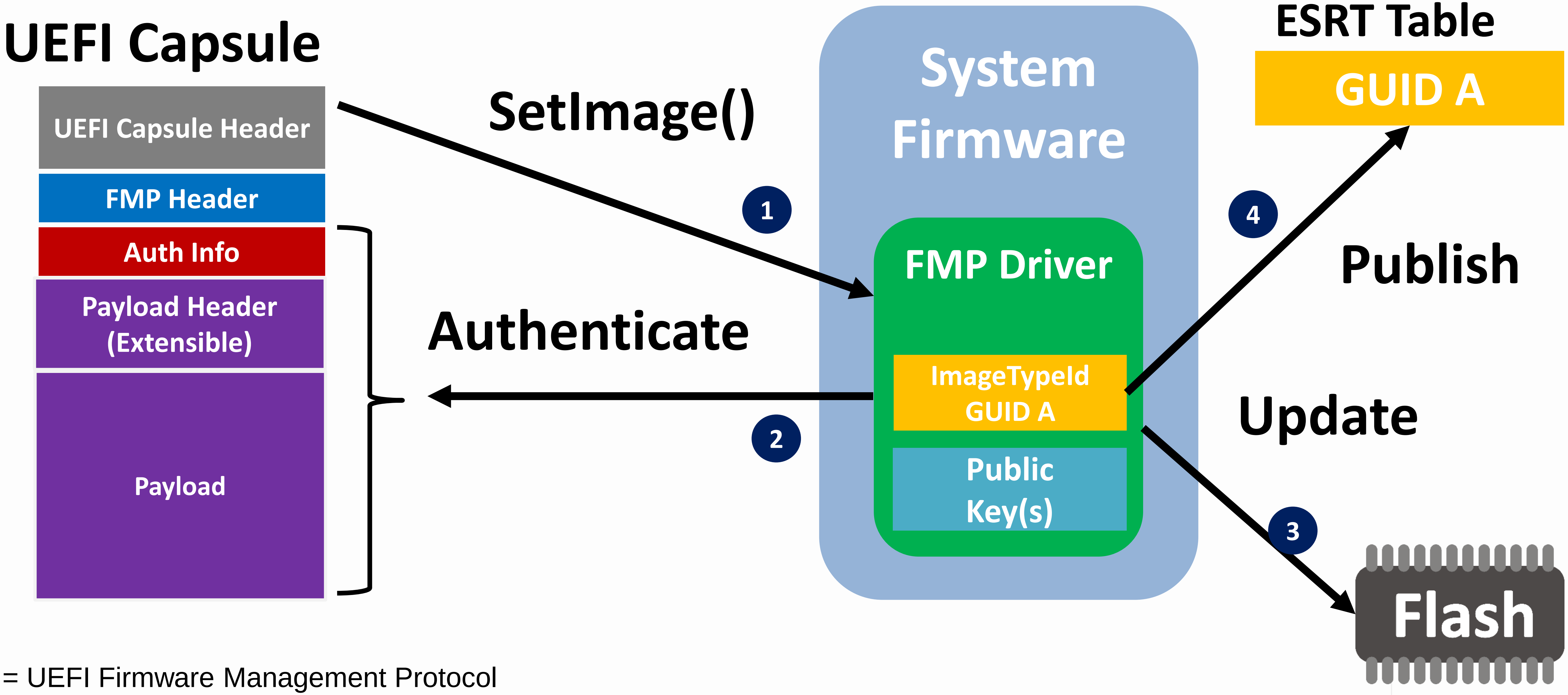
Verify Update from ESRT

Security (SEC)	Pre EFI Initialization (PEI)	Driver Execution Environment (DXE)	Boot Dev Select (BDS)	Transient System Load (TSL)	Run Time (RT)	After Life (AL)

Power on → [. . Platform initialization . .] → [. . . . OS boot . . . .] → Shutdown

ESRT = EFI System Resource Table

Process UEFI Capsule



FMP = UEFI Firmware Management Protocol
GUID = Globally Unique Identifier

EDK II UEFI Capsule Features

EFI Development Kit II (https://www.tianocore.org)

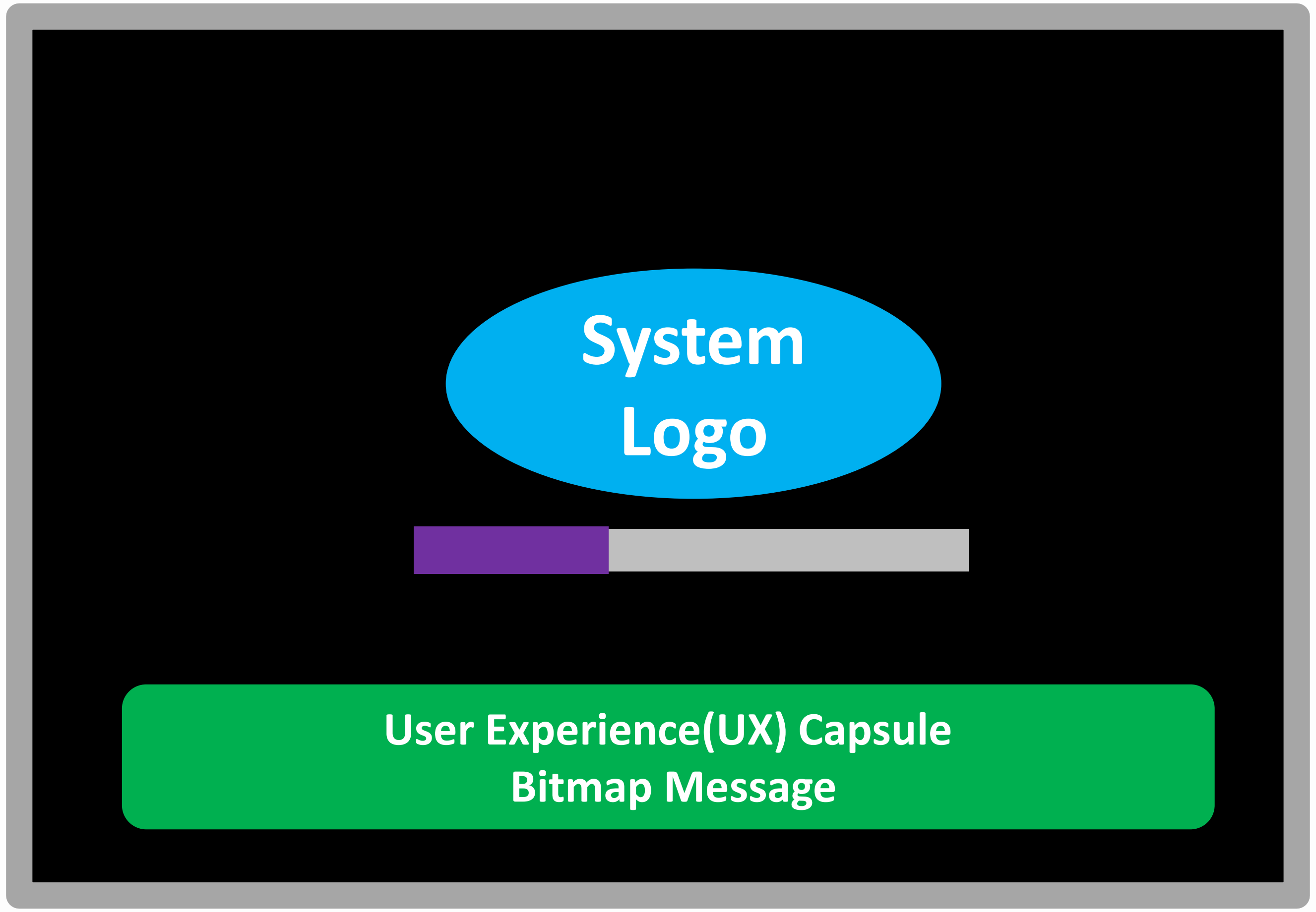
Feature	UDK2017 / UDK2018	edk2-stable201808
Generate UEFI Capsule	Integrated EDK II Build	Standalone Python* Script
Update Granularity	Focused on Monolithic	Designed to support Multiple Components
Authentication	PKCS7 Single Key	PKCS7 Multiple Keys
Pre Check	N/A	Power/Battery, Thermal, System
Update Indicator	Requires platform code	Built-in with Consistent UX and Progress Bar
Firmware Management Protocol (FMP)	Requires full implementation	Produced by FmpDxe module customized using configuration data and small libraries
Test Key Detection	Requires platform code	Built-in
Watchdog	Requires platform code	Built-in
ESRT Driver	Legacy + FMP	Smaller/Simpler FMP only version

* Other names and brands may be claimed as property of others

Firmware Update Indicators

UEFI Graphics Console

EFI_GRAPHICS_OUTPUT_PROTOCOL

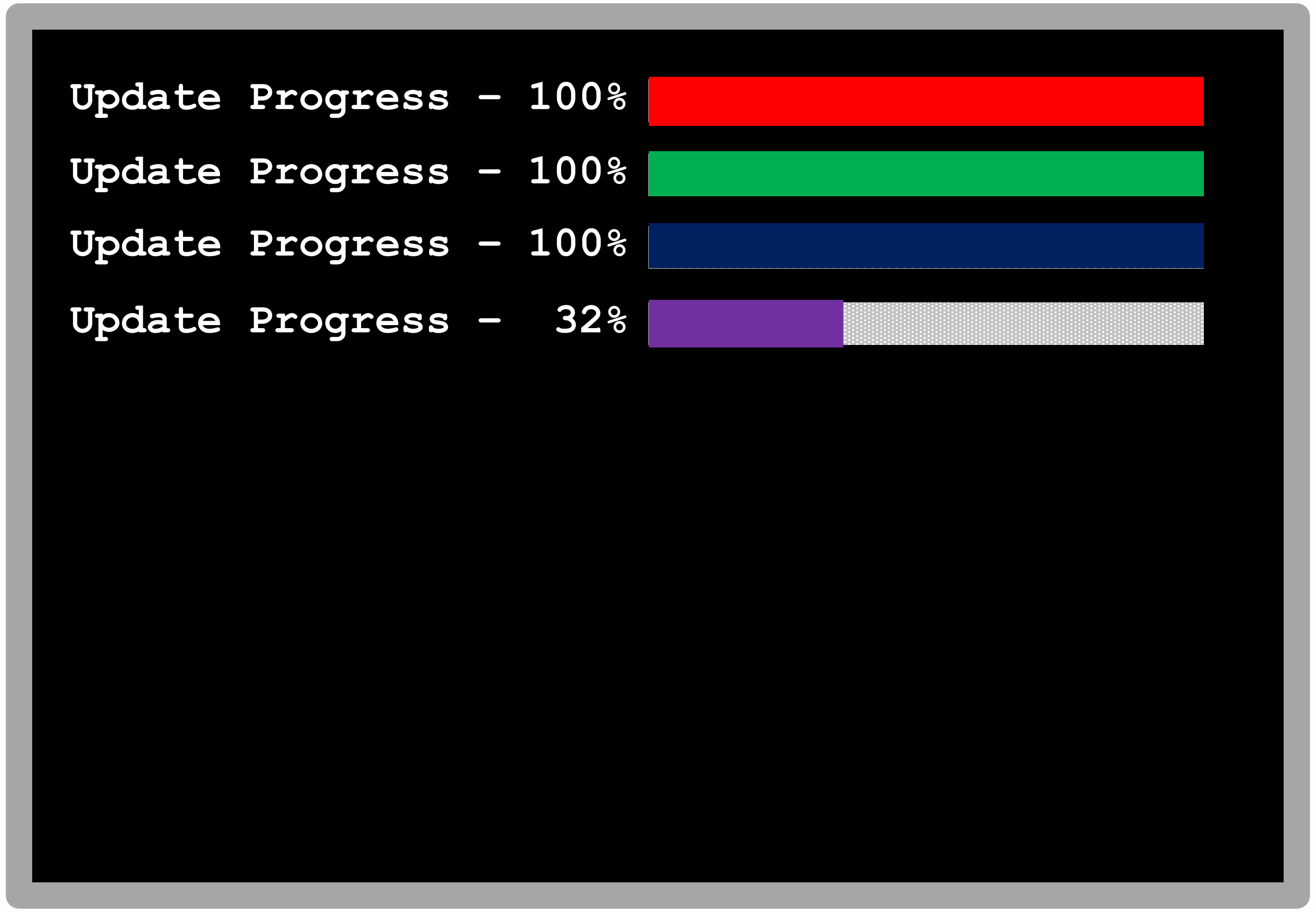


System Logo

User Experience(UX) Capsule Bitmap Message

UEFI Text Console

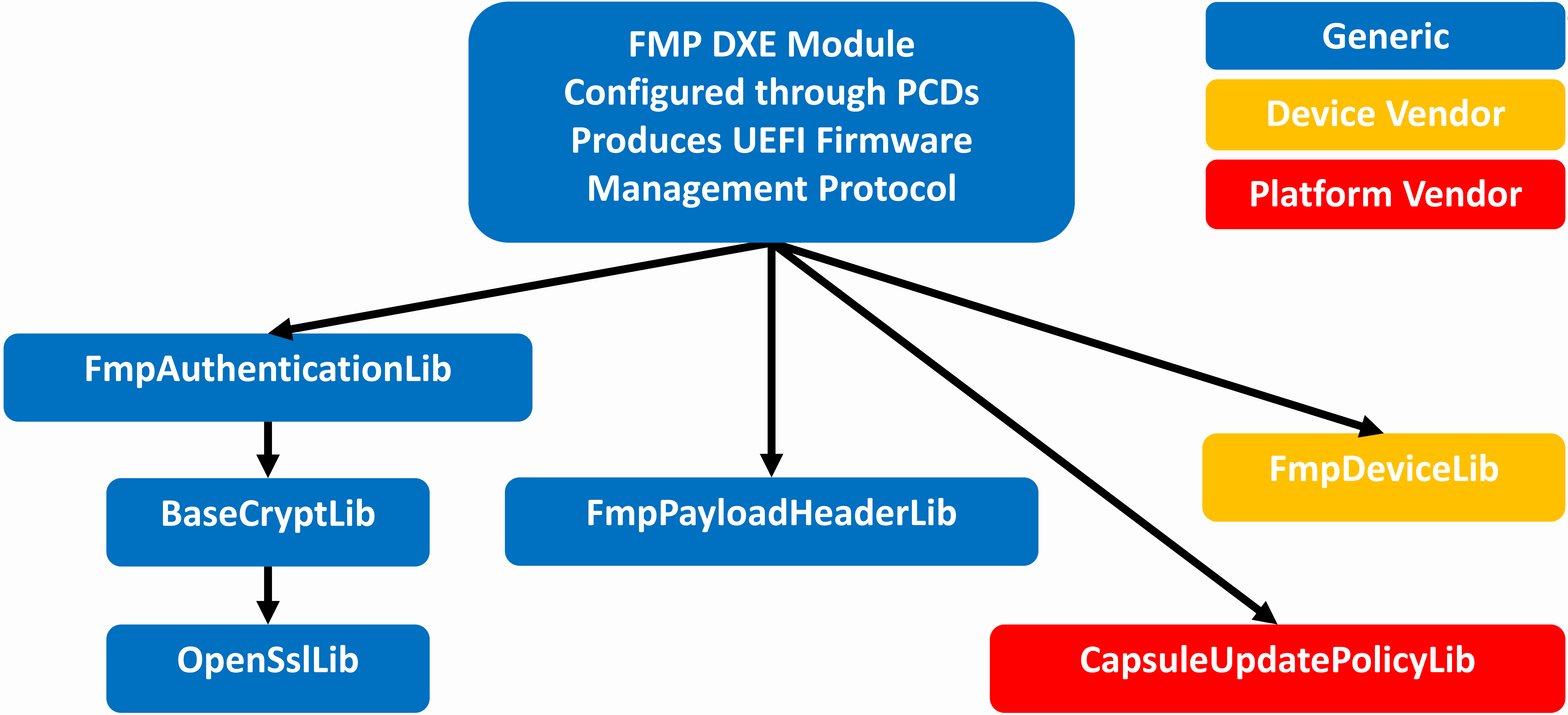
EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL



```
Update Progress – 100%
Update Progress – 100%
Update Progress – 100%
Update Progress –  32%
```

Customize with a new DisplayUpdateProgressLib instance

FmpDxe Module Overview



PCD = Platform Configuration Database

FmpDxe Module Configuration

Name	Description
`FILE_GUID`	ESRT GUID Value
`PcdFmpDeviceImageIdName`	FMP Image Descriptor - Unicode string
`PcdFmpDeviceBuildTimeLowestSupportedVersion`	Build time FMP/ESRT default value
`PcdFmpDeviceLockEventGuid`	Event GUID to lock FW storage device. Default is End of DXE.
`PcdFmpDeviceProgressWatchdogTimeInSeconds`	Watchdog armed on each progress update
`PcdFmpDeviceProgressColor`	24-bit Progress Bar Color (0x00rrggbb)
`PcdFmpDevicePkcs7CertBufferXdr`	One or more PKCS7 Certs in XDR format. Encode w/ `BaseTools/Scripts/BinToPcd`
`PcdFmpDeviceTestKeySha256Digest`	Set to `{0}` to disable test key detection

XDR = External Data Representation using Variable-Length Opaque Data format from RFC 4506

CapsuleUpdatePolicyLib APIs Platform Specific Library

Name	Description
`CheckSystemPower()`	Is system power/battery ok for FW update?
`CheckSystemThermal()`	Is system temperature ok for FW update?
`CheckSystemEnvironment()`	Is the system environment ok for FW update?
`IsLowestSupportedVersionCheckRequired()`	Skip lowest supported version check? (e.g. Service Mode)
`IsLockFmpDeviceAtLockEventGuidRequired()`	Skip firmware storage device lock action? (e.g. Manufacturing Mode)

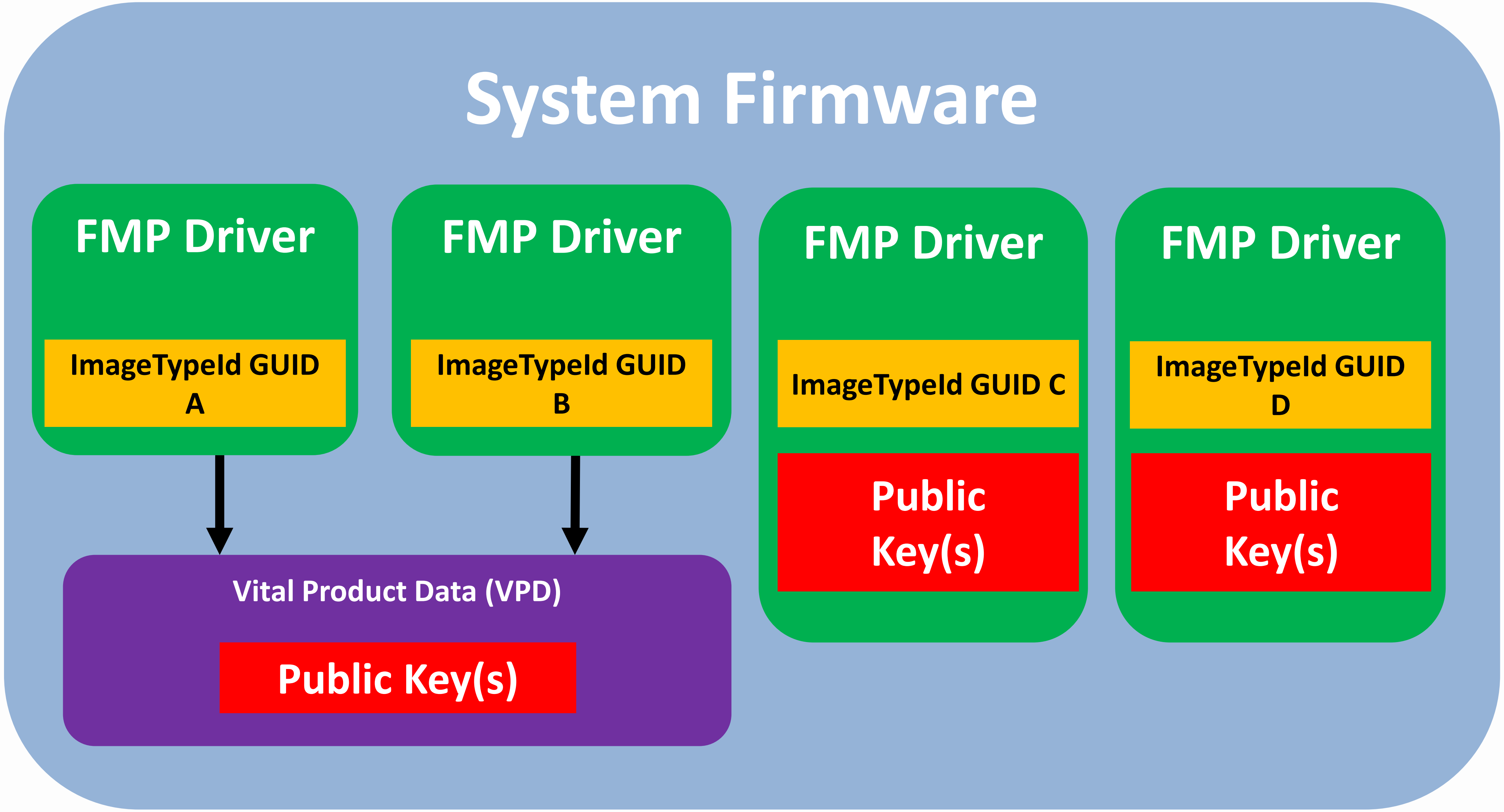
FmpDeviceLib APIs

Device Specific Library

Name	Description
`RegisterFmpInstaller()`	Future expansion for add-in controllers.
`FmpDeviceGetSize()`	Size of ***currently stored FW image.***
`FmpDeviceGetImageTypeIdGuidPtr()`	ESRT/FMP GUID. Overrides FILE_GUID value.
`FmpDeviceGetAttributes()`	FMP Attributes Supported/Settings.
`FmpDeviceGetLowestSupportedVersion()`	LSV from ***currently stored FW image.***
`FmpDeviceGetVersionString()`	Unicode version string from ***currently stored FW image.***
`FmpDeviceGetVersion()`	32-bit version value from ***currently stored FW image.***
`FmpDeviceGetImage()`	Retrieve copy of ***currently stored FW image.***
`FmpDeviceCheckImage()`	Check if a new FW image is valid for this device.
`FmpDeviceSetImage()`	Update FW storage with a new FW image.
`FmpDeviceLock()`	Lock FW storage to prevent any further changes.

ESRT GUIDs and Keys

(Multiple Components)



System Firmware

FMP Driver

ImageTypeId GUID A

FMP Driver

ImageTypeId GUID B

Vital Product Data (VPD)

Public Key(s)

FMP Driver

ImageTypeId GUID C

Public Key(s)

FMP Driver

ImageTypeId GUID D

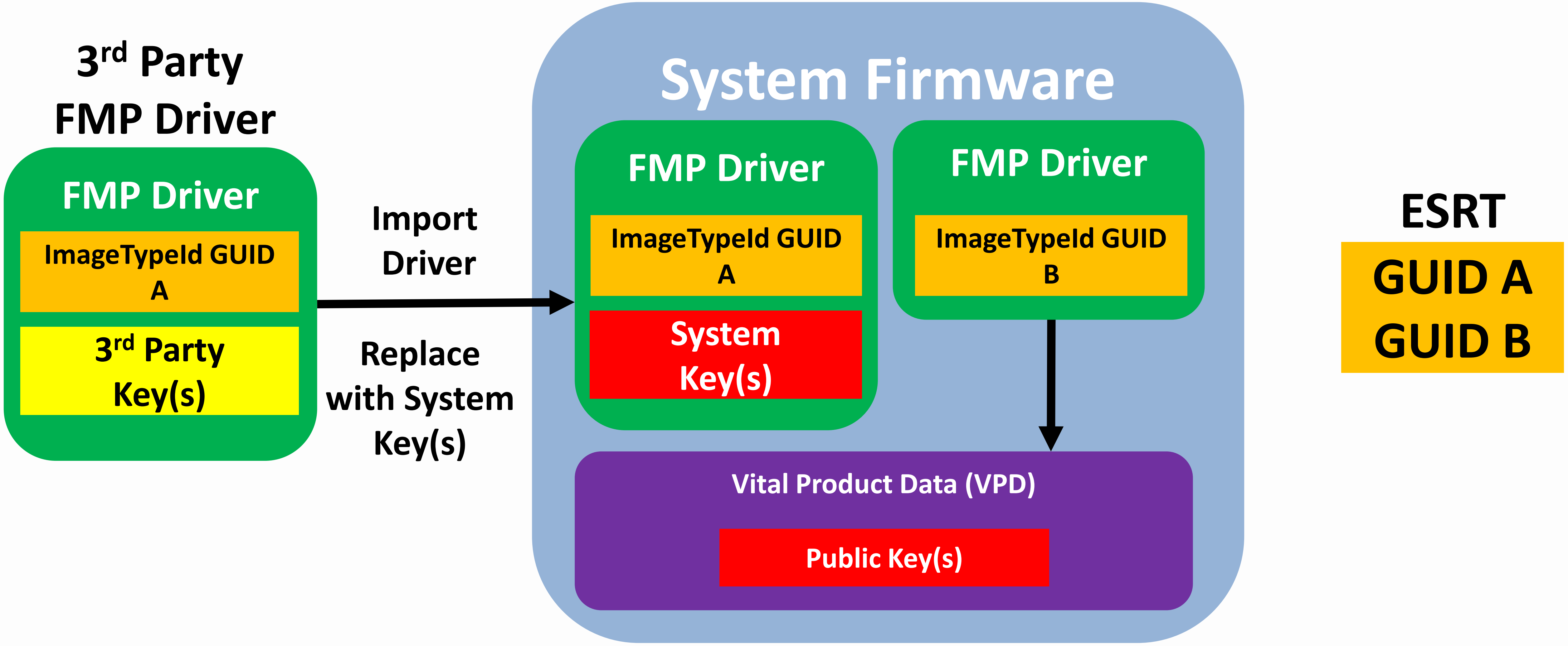
Public Key(s)

ESRT

GUID A
GUID B
GUID C
GUID D

ESRT GUIDs and Keys

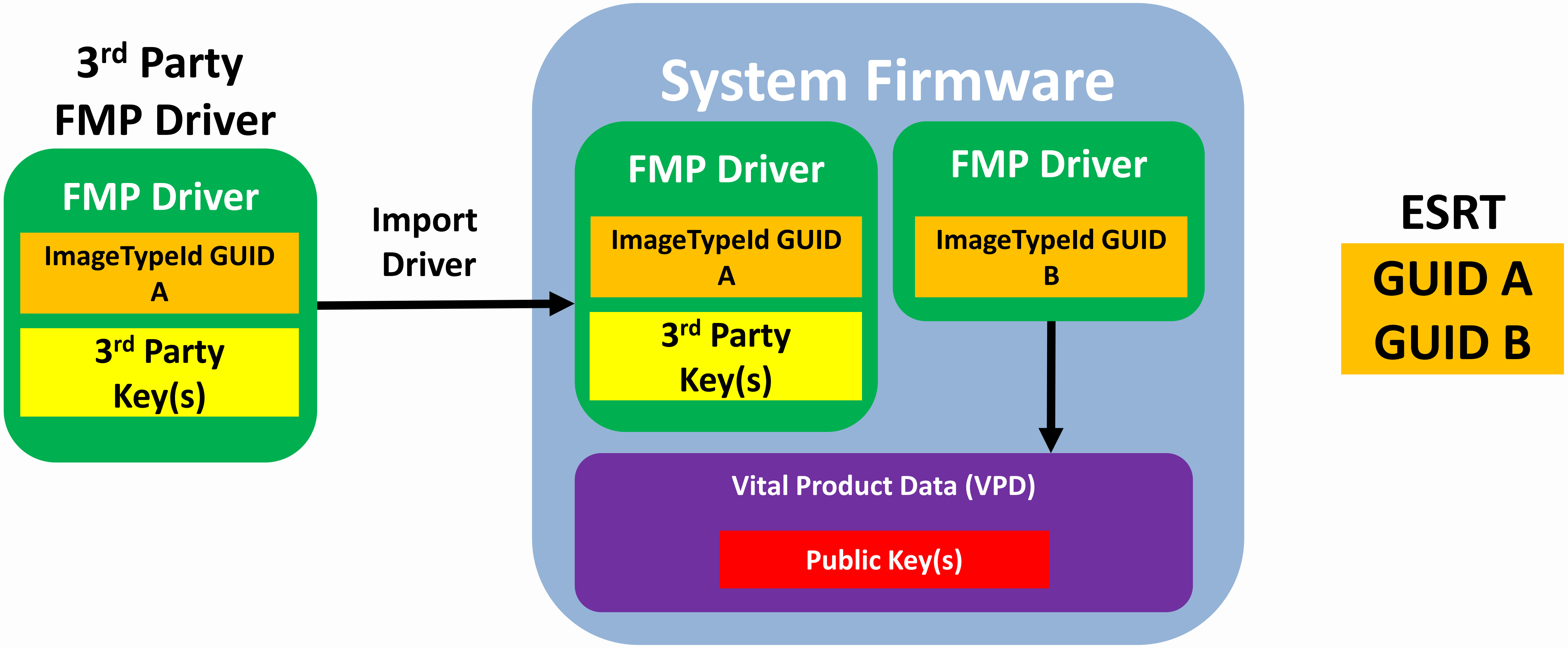
3rd Party FMP Driver



3rd Party UEFI Capsules must be re-signed with System Key

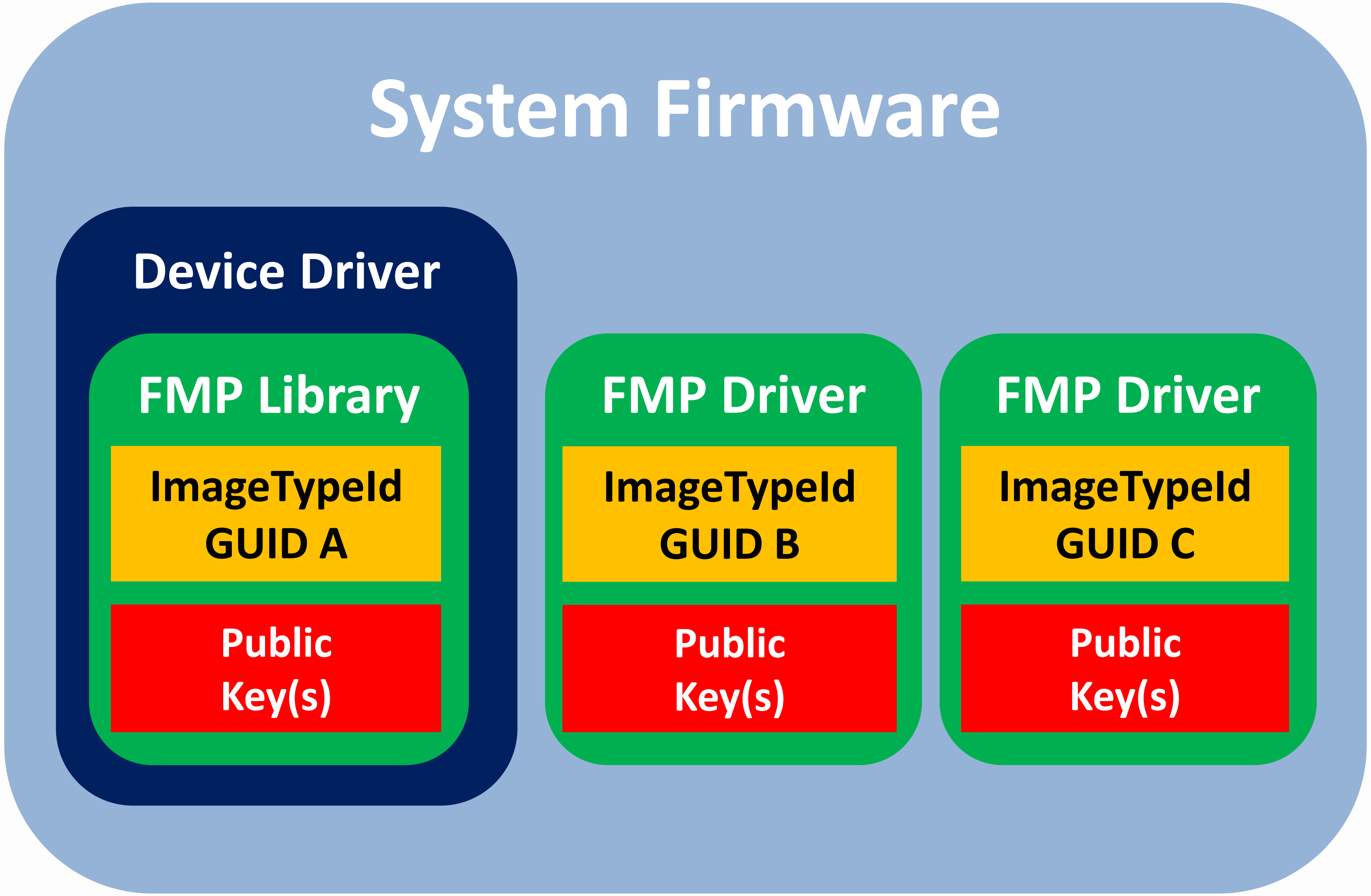
ESRT GUIDs and Keys

3rd Party FMP Driver



System allows UEFI Capsules from 3rd Party to be installed

Add FMP to Existing Device Driver



ESRT

GUID A
GUID B
GUID C

Summary

- New UEFI Capsule Update Features in EDK II
  - Platform firmware and device firmware (ESRT/FMP)
  - Multiple authentication keys & test key detection
  - Improved UX and system update pre-checks
- Simplified capsule generation (Python script)
- Supports OS-based firmware update workflow
  - Model Based Servicing via Microsoft Windows Update
  - Linux Vendor Firmware Service (LVFS) via fwupd.org

Call to Action

- Add UEFI Capsule Support to platforms
- Implement UEFI Capsule Update for devices
- Take advantage of EDK II FmpDevicePkg features
- Use Windows Update & LVFS to simplify distribution of firmware updates
- Provide feedback and contribute!
  - TianoCore - https://www.tianocore.org/
  - LVFS - https://fwupd.org/

Thanks for attending the Fall 2018 UEFI Seminar and Plugfest

For more information on the Unified EFI Forum and UEFI Specifications, visit http://www.uefi.org

presented by

intel®

Intel Legal Notice