CAPSULE UPDATE & LVFS IMPROVING SYSTEM FIRMWARE UPDATES

Presented by Brian Richardson, Intel Corporation

Materials prepared by Vincent Zimmer (Intel), Mike Kinney (Intel) and Richard Hughes (LVFS Maintainer)

FOSDEM (Feb 2020)

Topics

The Update Problem

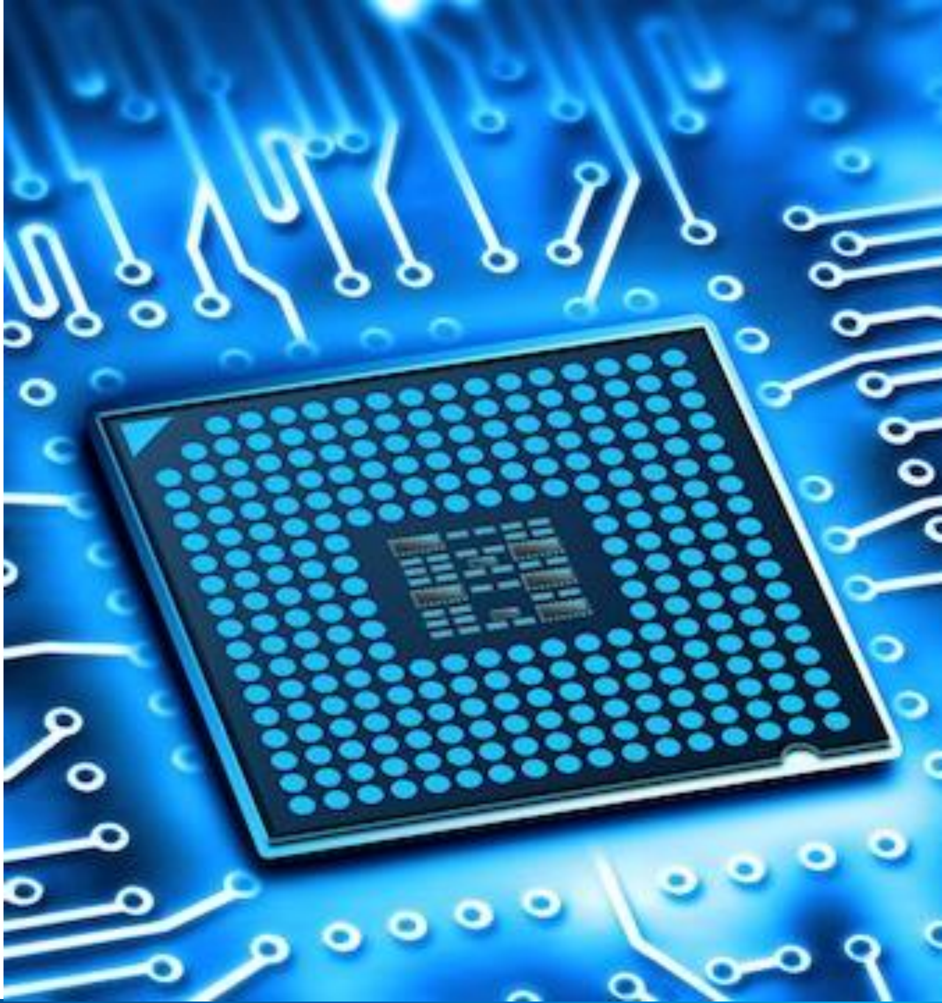
Using UEFI Capsules for Firmware Update

Firmware Management Protocol

Modularization

Distribution using LVFS

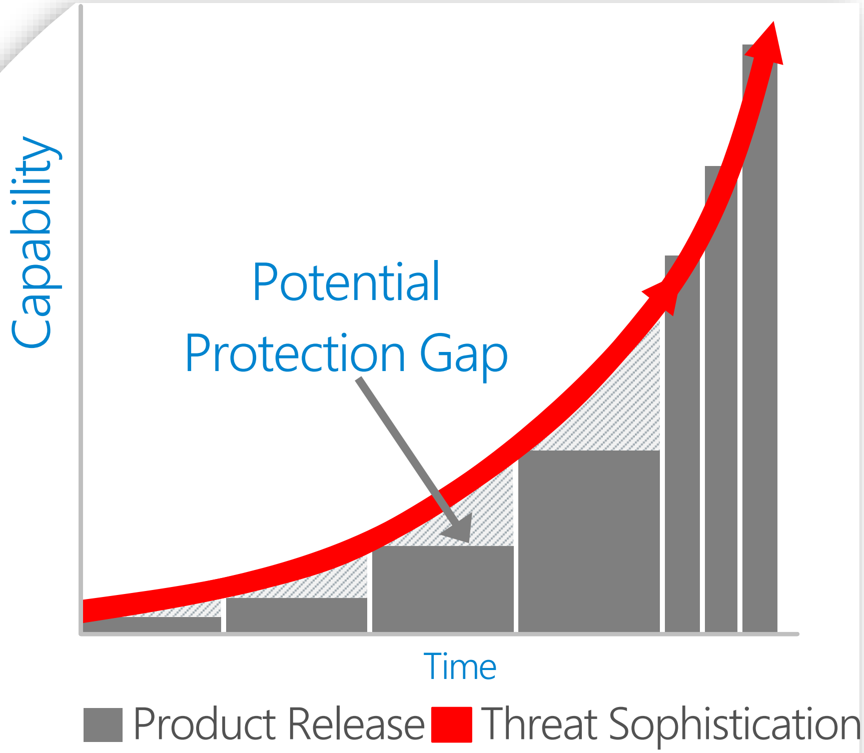
Summary & Call to Action



THE UPDATE PROBLEM

PROBLEM STATEMENT

- Low-attach rate for firmware updates on end-user systems
- Firmware process is traditionally designed for experts, not users
- Creates an environment where released updates are never applied



CHALLENGES FOR LINUX

- OEM update process typically targets users of Microsoft Windows
- Running an update utility at Linux runtime has technical complexities
- Creates an environment where released updates are never applied

Image by OpenClipart-Vectors from Pixabay

Current Solution Space

Standardized Delivery Format

- OS-independent payload (Capsule)

Infrastructure for Update Delivery

- Consistent protocols and data formats
- OS-based staging infrastructure

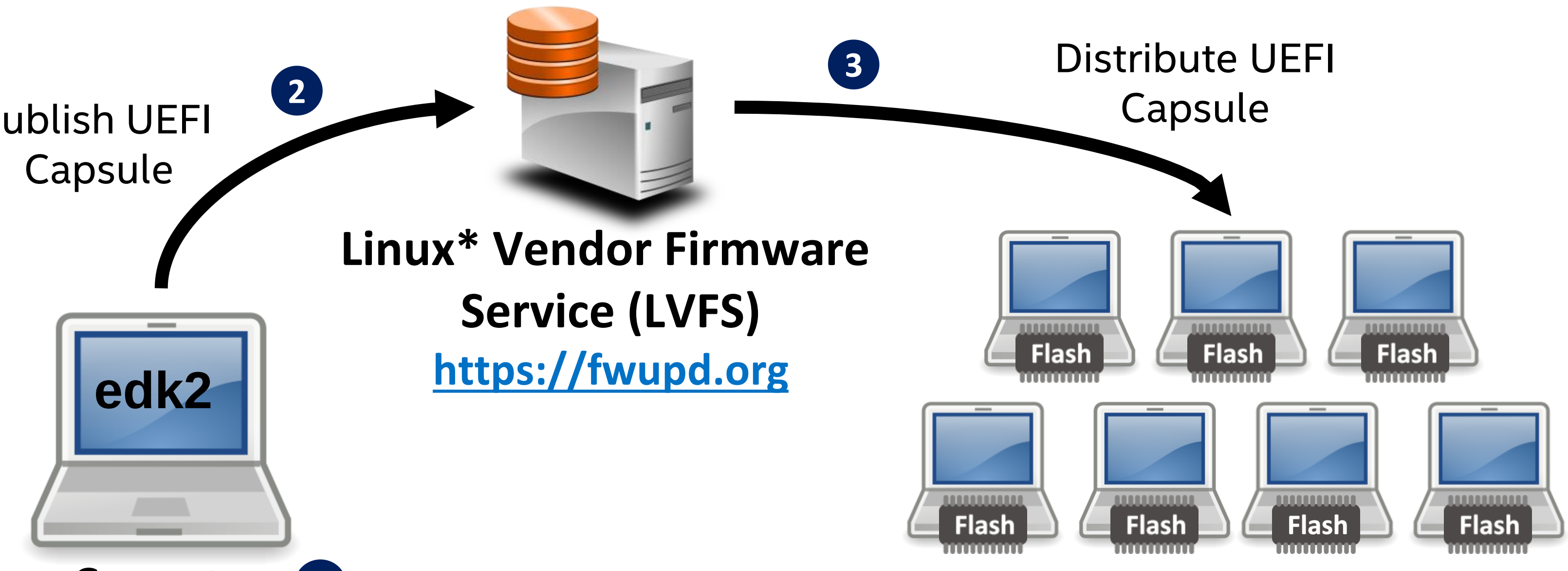
Leverage Modular Firmware Infrastructure

- Drive innovation through expandability & flexibility

USING UEFI CAPSULES FOR FIRMWARE UPDATE

Using UEFI Capsules for Firmware Update

(Unified Extensible Firmware Interface)



1 Generate UEFI Capsule

2 Publish UEFI Capsule

Linux* Vendor Firmware Service (LVFS)

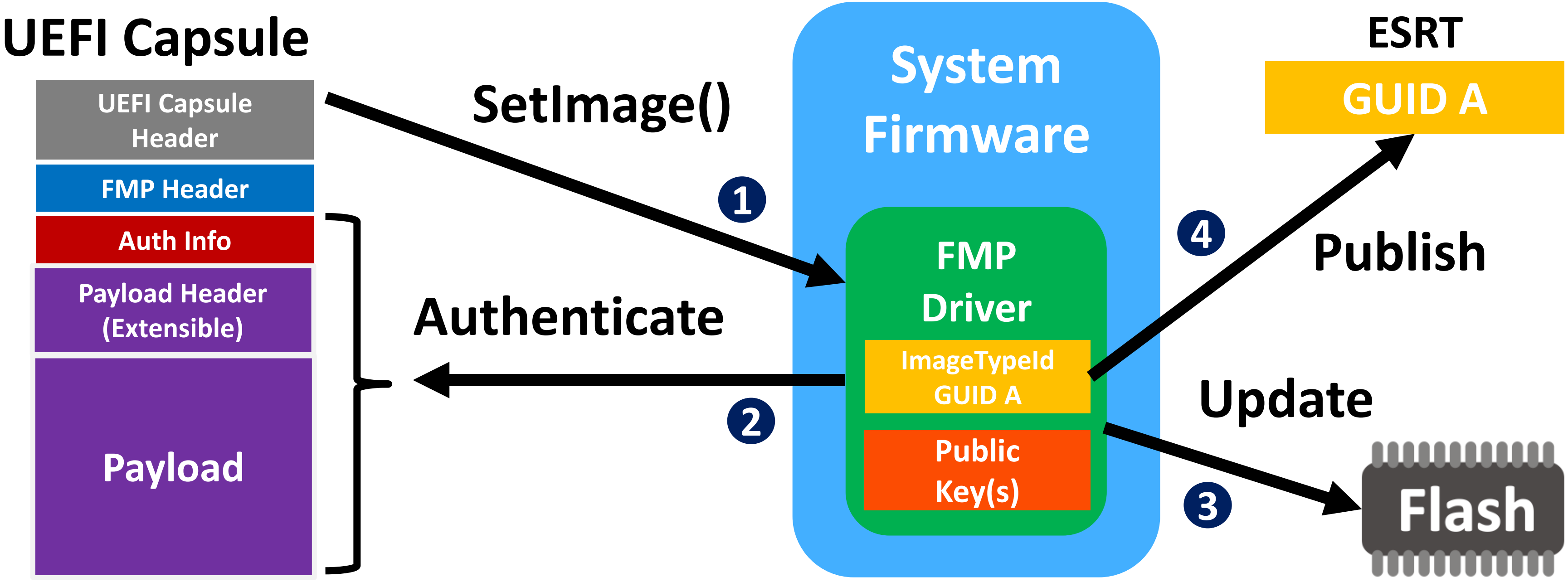
https://fwupd.org

3 Distribute UEFI Capsule

4 Process UEFI Capsule

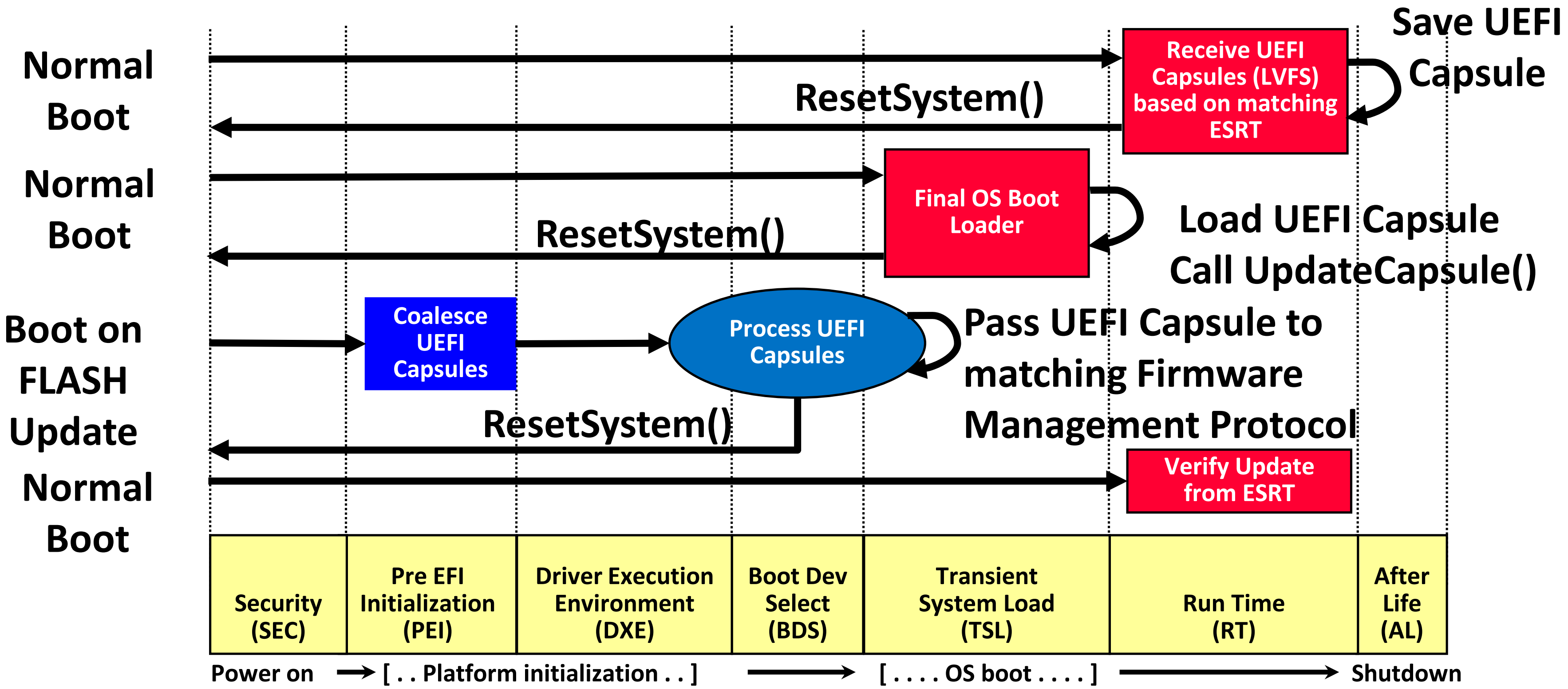
* Other names and brands may be claimed as property of others

Process UEFI Capsule



ESRT = EFI System Resource Table
FMP = Firmware Management Protocol
GUID = Globally Unique Identifier

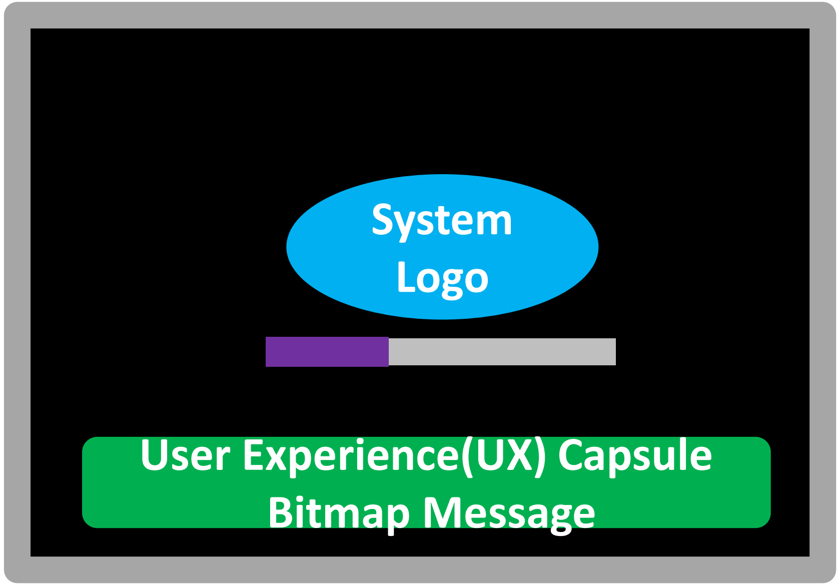
UEFI Capsule Processing using UEFI PI



Firmware Update Indicators

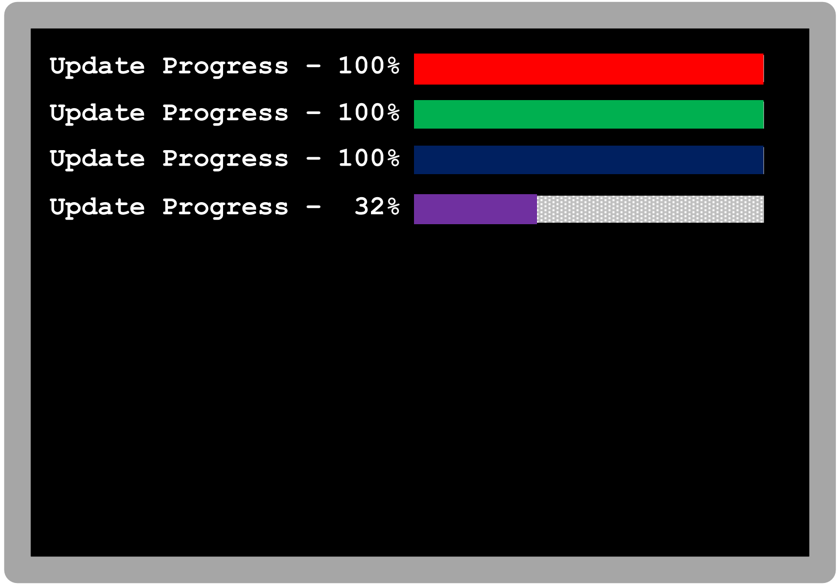
UEFI Graphics Console

EFI_GRAPHICS_OUTPUT_PROTOCOL



UEFI Text Console

EFI_SIMPLE_TEXT_OUTPUT_PROTOCOL



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

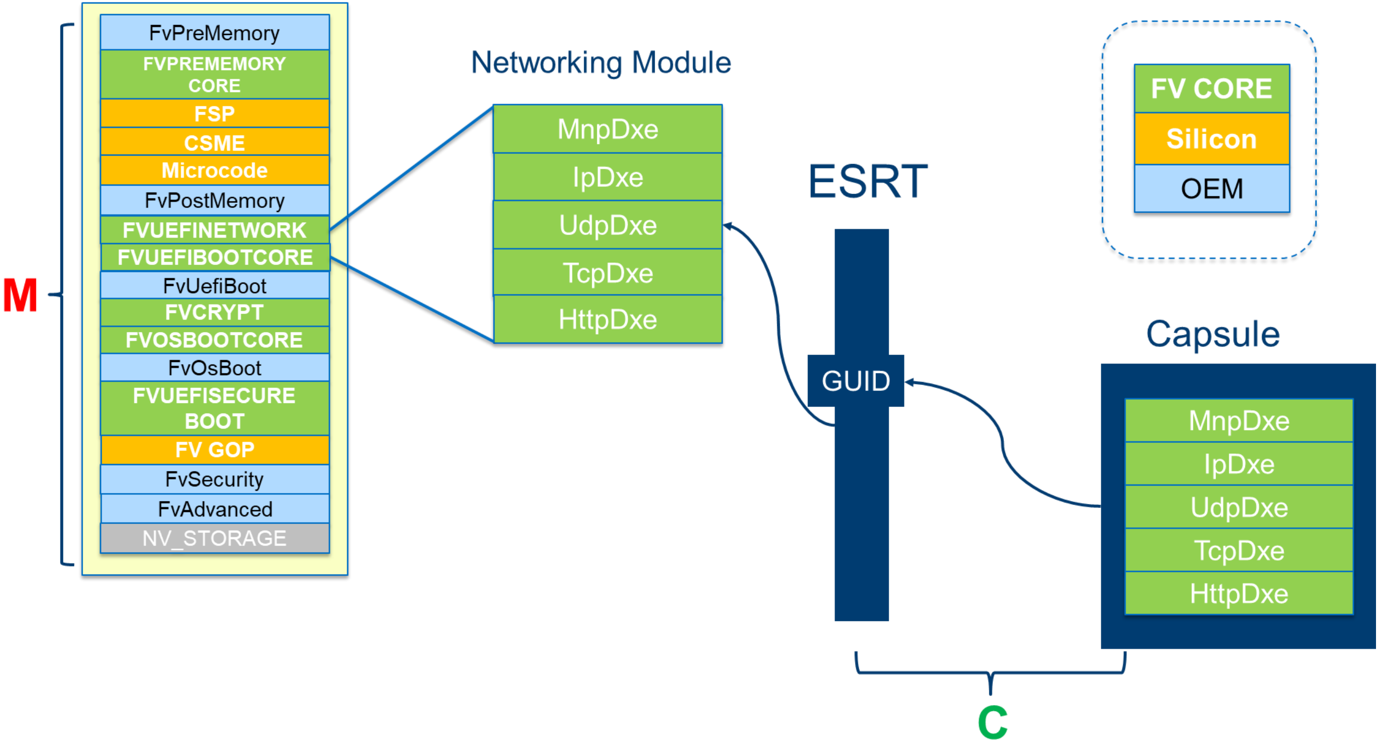
Customize with a new DisplayUpdateProgressLib instance

MODULARIZATION

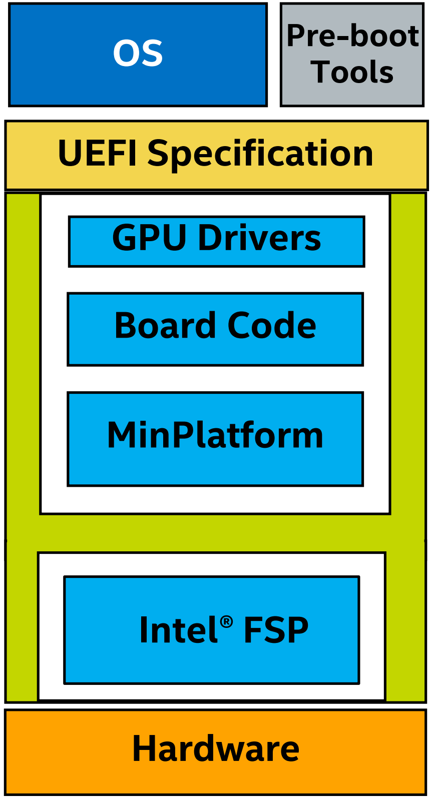
The Modular Philosophy

Make firmware component integration easy during Manufacturing.

Make firmware update easy using Capsules.

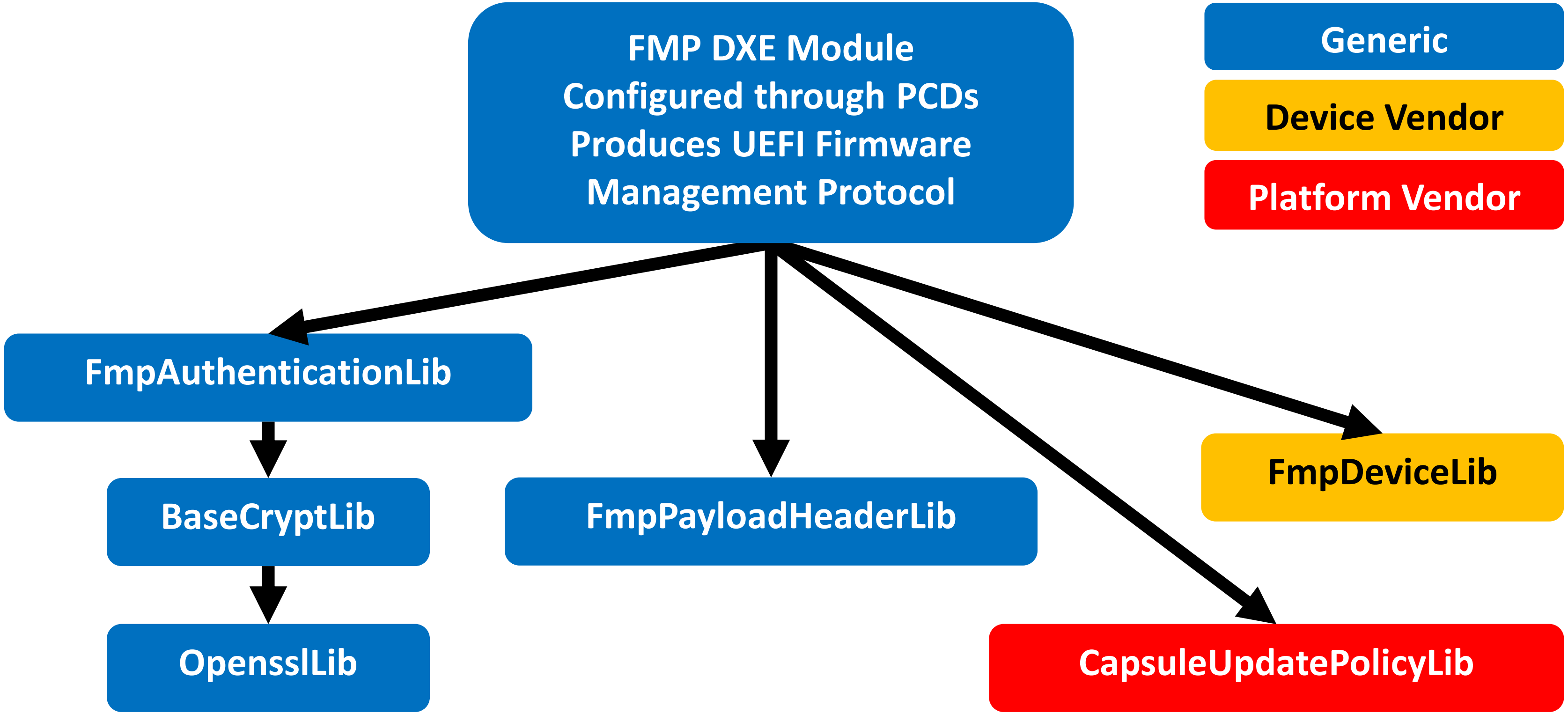


Intel Open Platform – Minimum Platform + Intel® FSP



- UEFI is built with the PC supply chain in mind.
  - Open & closed modules co-exist in a system.
  - Minimum Platform increases overall share of open source UEFI firmware code available.
  - Increases open source firmware for community engagement, development & testing.
- UEFI component-based design gives OEMs choices:
  - Wide array of peripherals and components:
    - CPU, GPU, I/O Controllers (USB, Disk, etc.)
- Silicon vendors can provide pluggable UEFI components that adhere to specifications.

FmpDxe Module Overview



PCD = Platform Configuration Database

DISTRIBUTION USING LVFS

Distribution Using LVFS

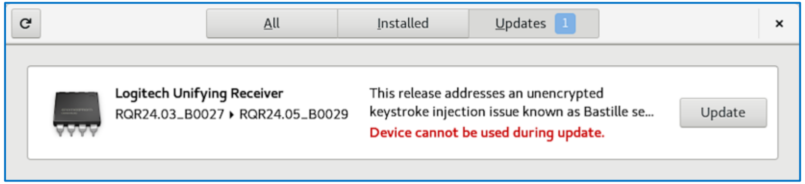
Two Major Components

fwupd - Mechanism

- 100% free software (LGPLv2+)
- Used by users, typically with a GUI

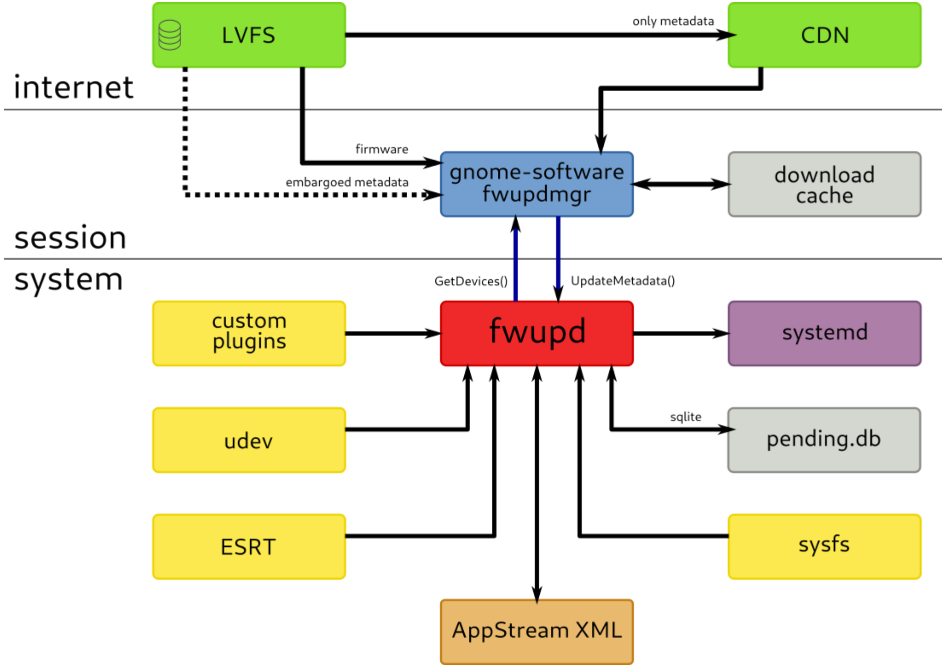
lvfs-website – Data Source

- 100% free software (GPLv2+)
- Used by vendors: OEMs and ODMs

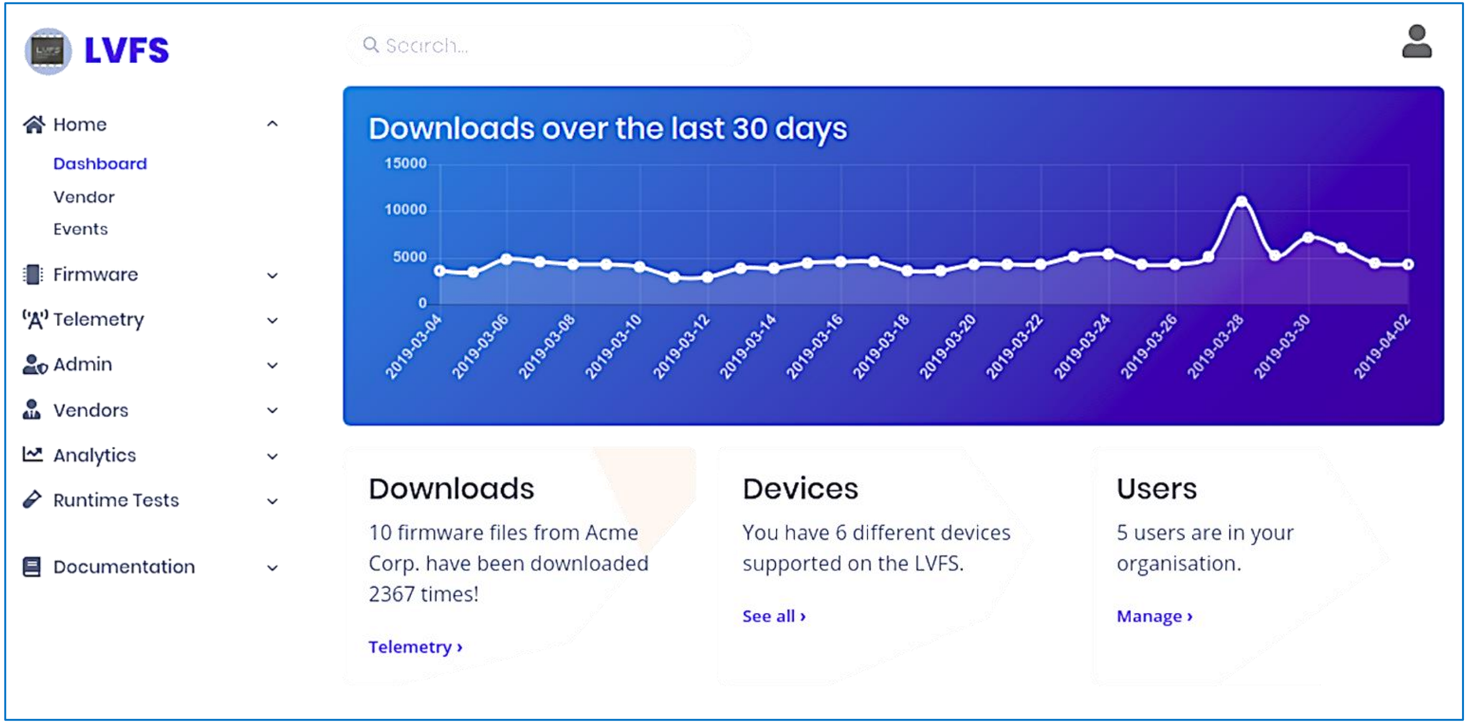


Logitech Unifying Receiver
RQR24.03_B0027 ▸ RQR24.05_B0029

This release addresses an unencrypted keystroke injection issue known as Bastille se...
Device cannot be used during update.



LVFS – “It’s Just a Website”



Downloads

10 firmware files from Acme Corp. have been downloaded 2367 times!

Devices

You have 6 different devices supported on the LVFS.

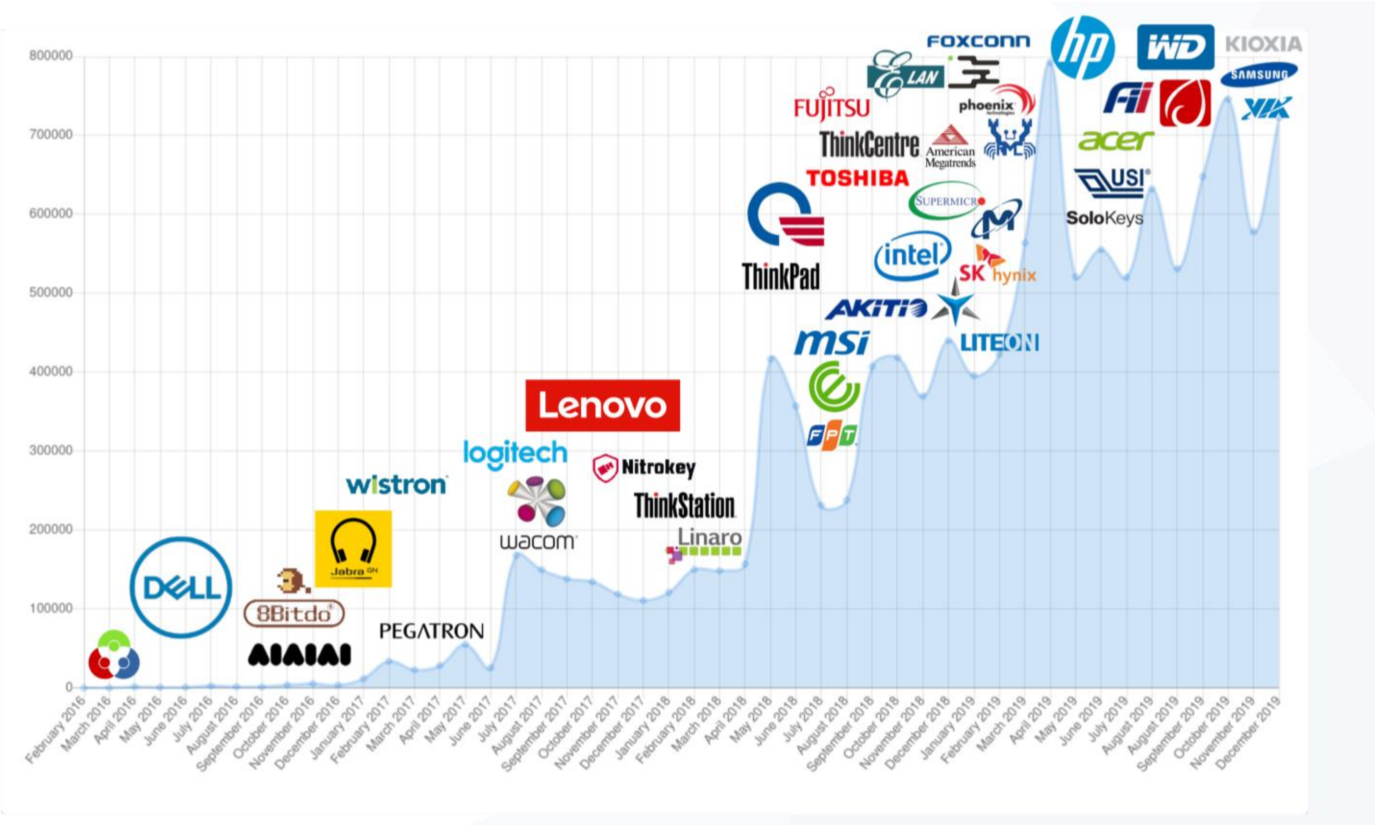
Users

5 users are in your organisation.

Designed for vendor secrecy (permissions system)
LVFS can be mirrored using PULP

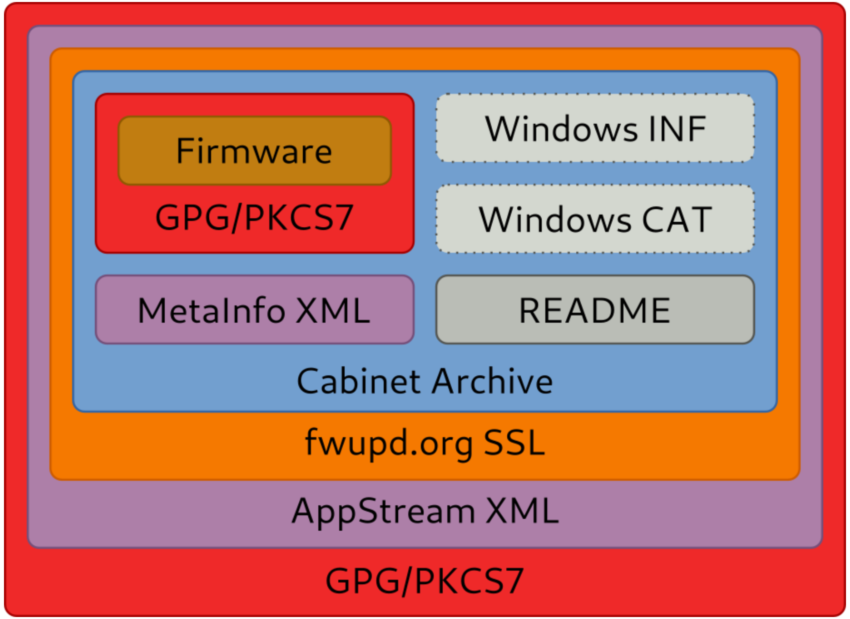
Material contributed by rhughes@redhat.com

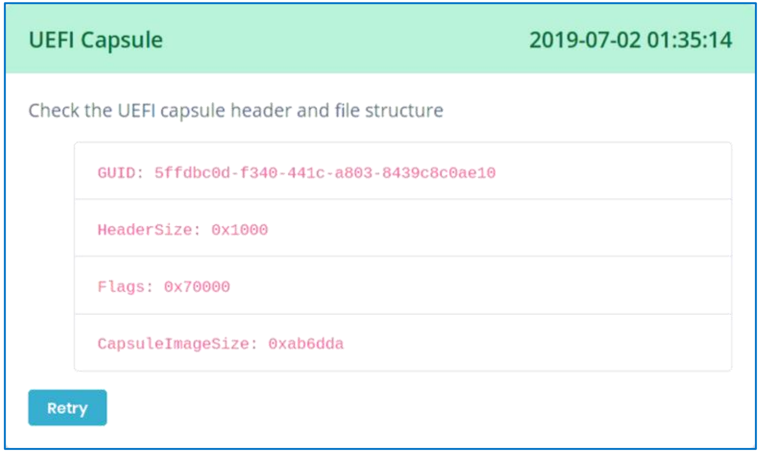
Vendor Support for LVFS



Material contributed by rhughes@redhat.com

Layers of Security & UEFI Capsule Verification

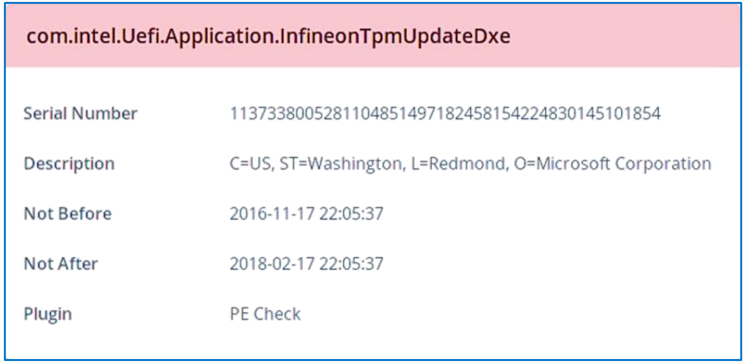




UEFI Capsule 2019-07-02 01:35:14

Check the UEFI capsule header and file structure

```
GUID: 5ffdbc0d-f340-441c-a803-8439c8c0ae10
HeaderSize: 0x1000
Flags: 0x70000
CapsuleImageSize: 0xab6dda
```

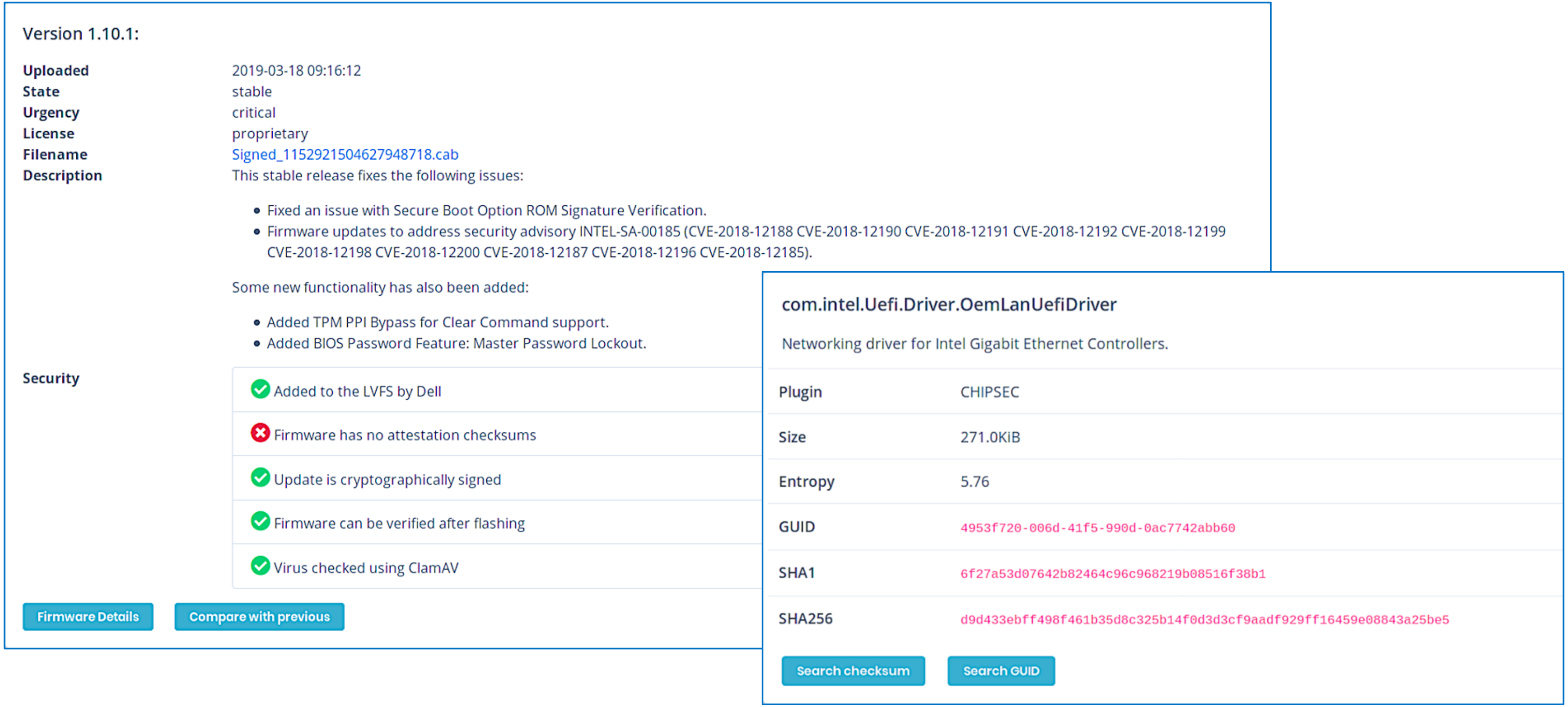


com.intel.Uefi.Application.InfineonTpmUpdateDxe

Serial Number	1137338005281104851497182458154224830145101854
Description	C=US, ST=Washington, L=Redmond, O=Microsoft Corporation
Not Before	2016-11-17 22:05:37
Not After	2018-02-17 22:05:37
Plugin	PE Check

Material contributed by rhughes@redhat.com

Firmware Analysis (LVFS Server Side)



Version 1.10.1:

Uploaded	2019-03-18 09:16:12
State	stable
Urgency	critical
License	proprietary
Filename	Signed_1152921504627948718.cab
Description	This stable release fixes the following issues: • Fixed an issue with Secure Boot Option ROM Signature Verification. • Firmware updates to address security advisory INTEL-SA-00185 (CVE-2018-12188 CVE-2018-12190 CVE-2018-12191 CVE-2018-12192 CVE-2018-12199 CVE-2018-12198 CVE-2018-12200 CVE-2018-12187 CVE-2018-12196 CVE-2018-12185). Some new functionality has also been added: • Added TPM PPI Bypass for Clear Command support. • Added BIOS Password Feature: Master Password Lockout.
Security	✓ Added to the LVFS by Dell ✗ Firmware has no attestation checksums ✓ Update is cryptographically signed ✓ Firmware can be verified after flashing ✓ Virus checked using ClamAV

com.intel.Uefi.Driver.OemLanUefiDriver

Networking driver for Intel Gigabit Ethernet Controllers.

Plugin	CHIPSEC
Size	271.0KiB
Entropy	5.76
GUID	4953f720-006d-41f5-990d-0ac7742abb60
SHA1	6f27a53d07642b82464c96c968219b08516f38b1
SHA256	d9d433ebff498f461b35d8c325b14f0d3d3cf9aadf929ff16459e08843a25be5

Material contributed by rhughes@redhat.com

Way too much LVFS info for one presentation!

Looking to the Future

- Dashboard, albeit with caveats
- Get adoption from a few remaining vendors
- More tests, possibly using external companies

Per Richard… “Question Everything! (except asking what vendors are testing in secret!)”

- https://www.fwupd.org/
- https://github.com/fwupd/lvfs-website

Material contributed by rhughes@redhat.com

SUMMARY & CALL TO ACTION

Summary

EDK II supports UEFI Capsule Infrastructure for Firmware Update

- Simplifies FMP support for system firmware and integrated devices.
- Multiple authentication keys with flexible key storage options.
- System update pre-check (Power/battery, thermal, and system).
- Improved UX with progress indicators during update.
- Built-in support for test key detection & watchdog timer.
- Simplified ESRT driver using FMP instances

Open Source Developers can Generate Signed UEFI Capsules

Infrastructure Simplifies Distribution and Adoption of Firmware Updates

Call to Action

Platform Designers & OEMs

- Use Signed Capsules to distribute firmware updates
  - Guidance: NIST 800-147/800-147B
- Post updates to LVFS & Microsoft Windows Update
- Require device vendors create capsules for their components
- Platforms should implement a firmware recovery solution (NIST 800-193)

Developers

- Engage with open source communities supporting modern update solutions (examples: LVFS, EDK II) to ensure compatibility with future products

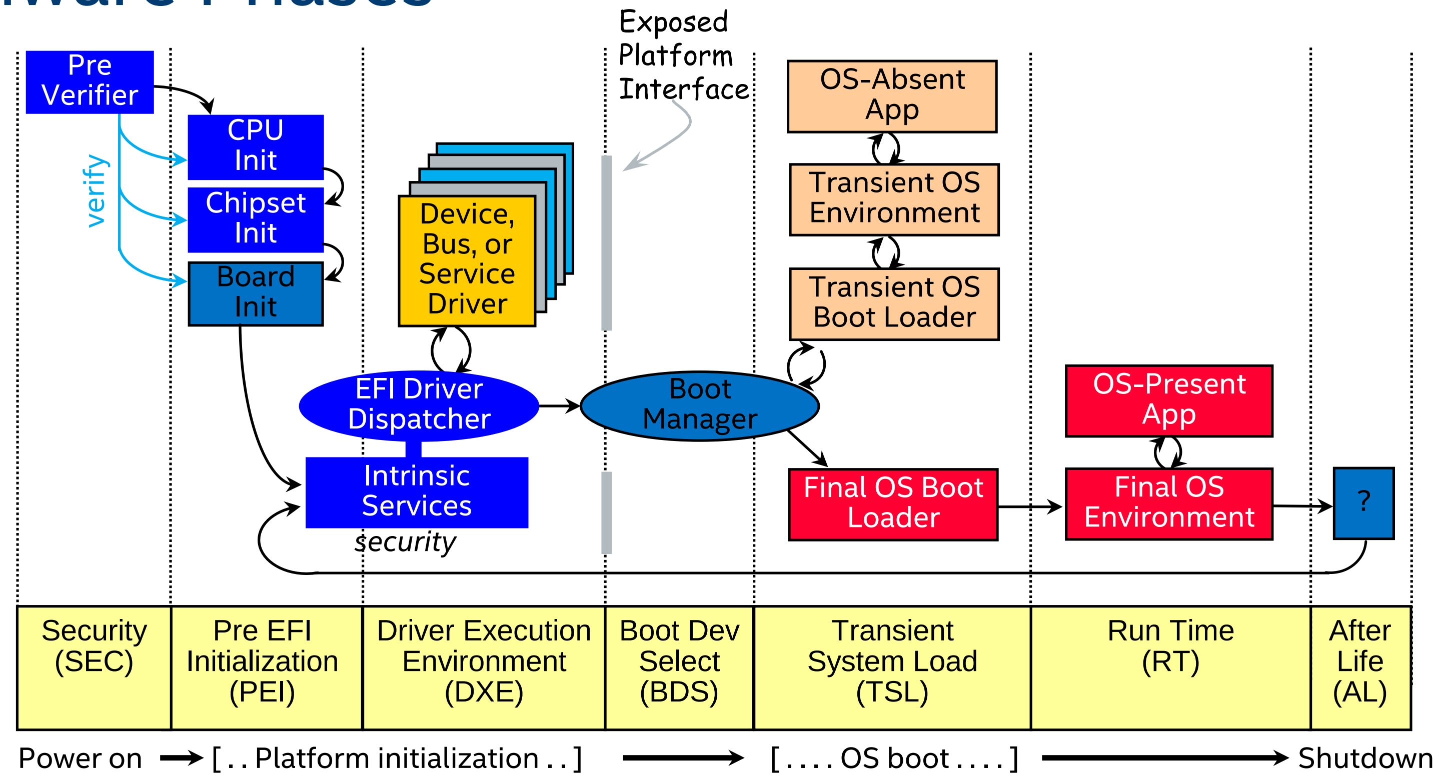
More Information

- Firmware threat model information:
  - https://edk2-docs.gitbooks.io/edk-ii-secure-coding-guide/content/appendix_threat_model_for_edk_ii/asset_flash_content.html
  - https://edk2-docs.gitbooks.io/understanding-the-uefi-secure-boot-chain/content/secure_boot_chain_in_uefi/boot_chain__putting_it_all_together/signed-capsule-update.html
- LVFS: https://fwupd.org/
- Microsoft Windows Update: https://docs.microsoft.com/en-us/windows-hardware/drivers/bringup/windows-uefi-firmware-update-platform
- UEFI Specifications: https://uefi.org/specifications
- EDK II MinPlatform Specification: https://legacy.gitbook.com/book/edk2-docs/edk-ii-minimum-platform-specification/details
- Additional Resources:
  - https://firmware.intel.com/sites/default/files/resources/UEFI_Plugfest_2015_Challenges_in_the_Cloud_Whitepaper_0.pdf
  - https://uefi.org/sites/default/files/resources/OCPsummit2016_Towards%20a%20Firmware%20Update%20Standard.pdf

intel®

BACKUP

UEFI Platform Initialization (PI) Architecture Firmware Phases



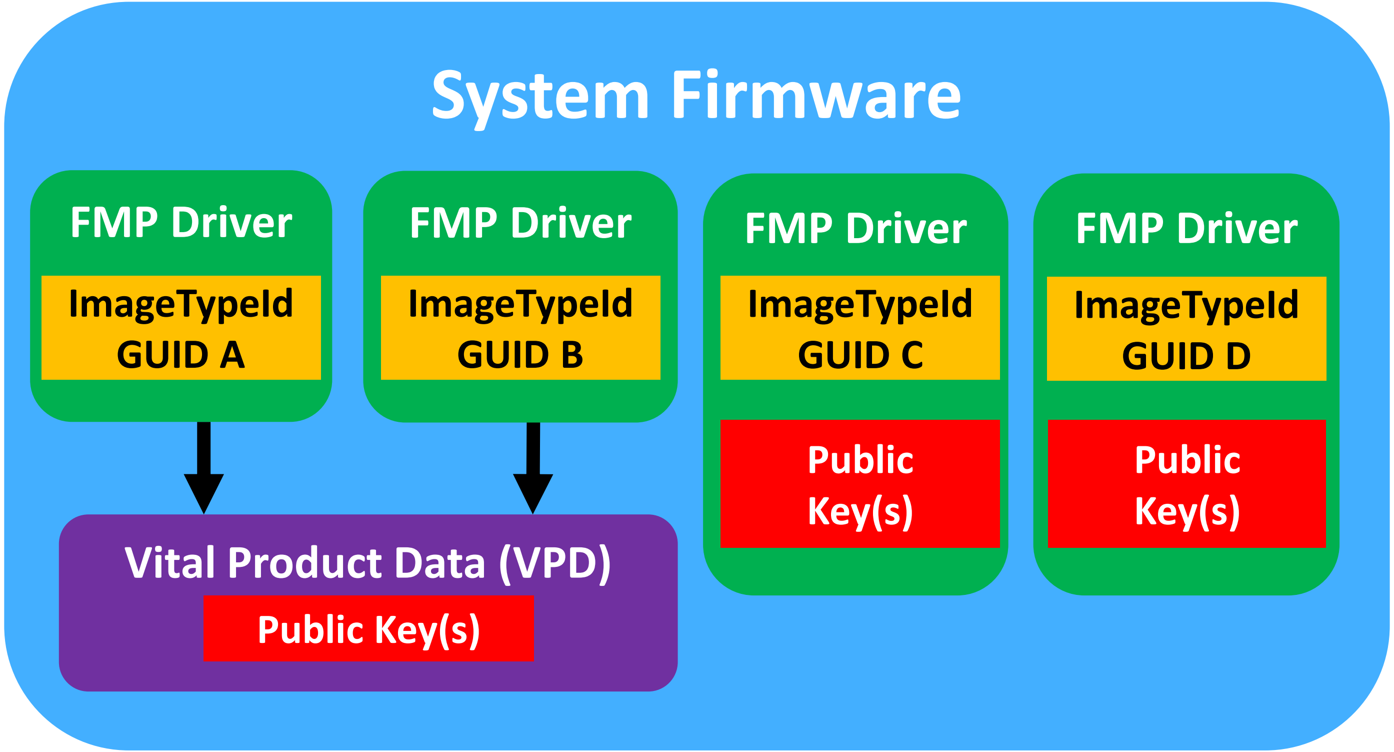
EDK II UEFI Capsule Features

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	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

ESRT GUIDs and Keys
Multiple Components



System Firmware

FMP Driver

ImageTypeId
GUID A

FMP Driver

ImageTypeId
GUID B

FMP Driver

ImageTypeId
GUID C

Public
Key(s)

FMP Driver

ImageTypeId
GUID D

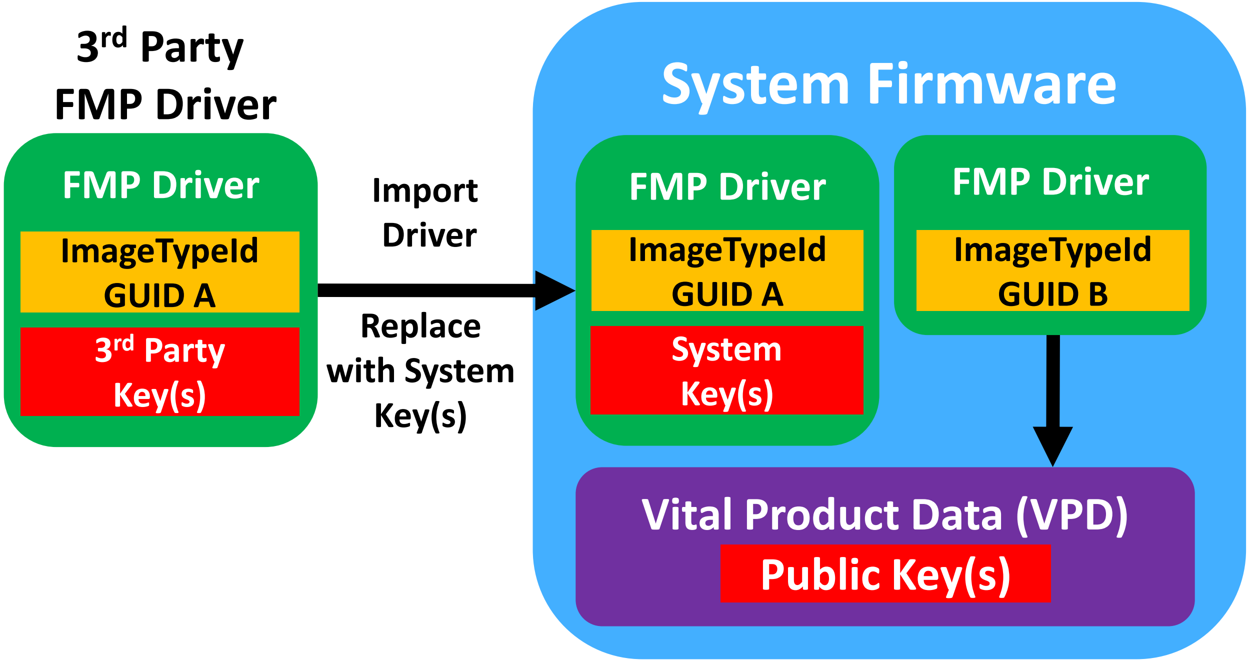
Public
Key(s)

Vital Product Data (VPD)

Public Key(s)

ESRT Table
GUID A
GUID B
GUID C
GUID D

ESRT GUIDs and Keys
3rd Party FMP Driver

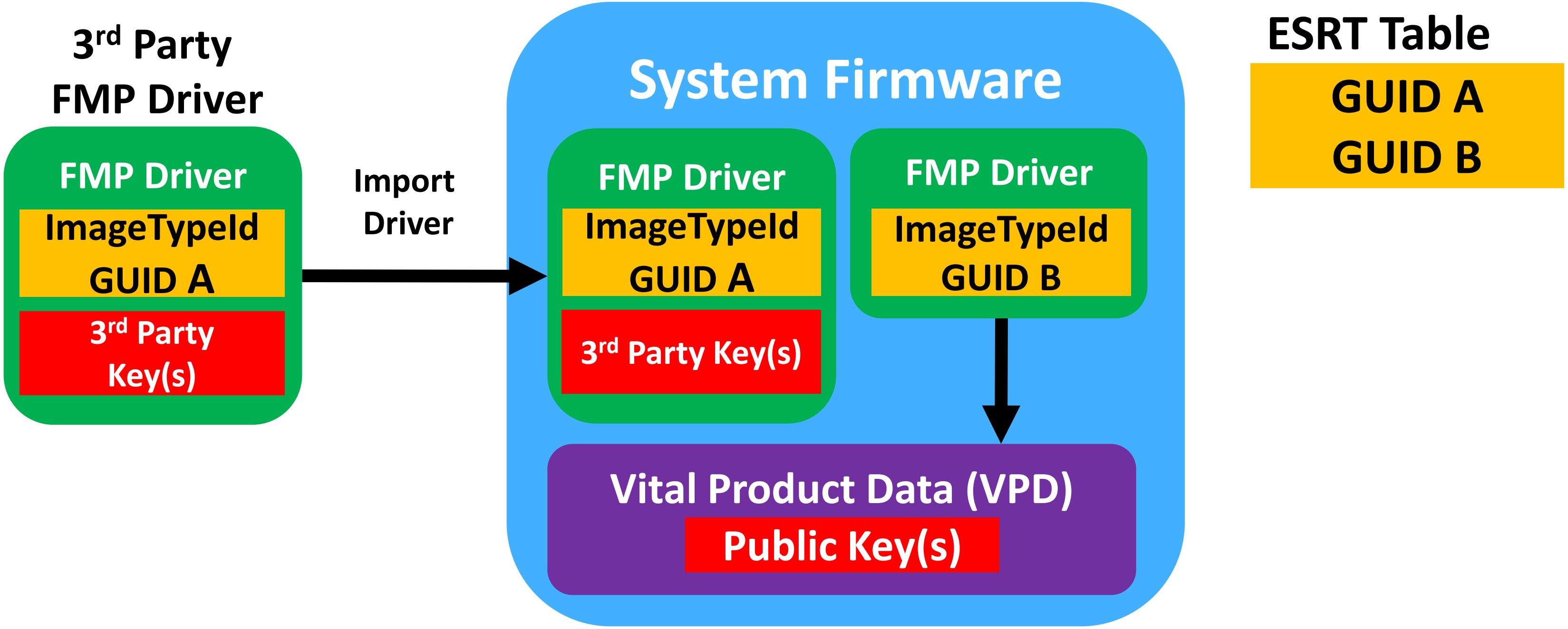


ESRT Table

GUID A
GUID B

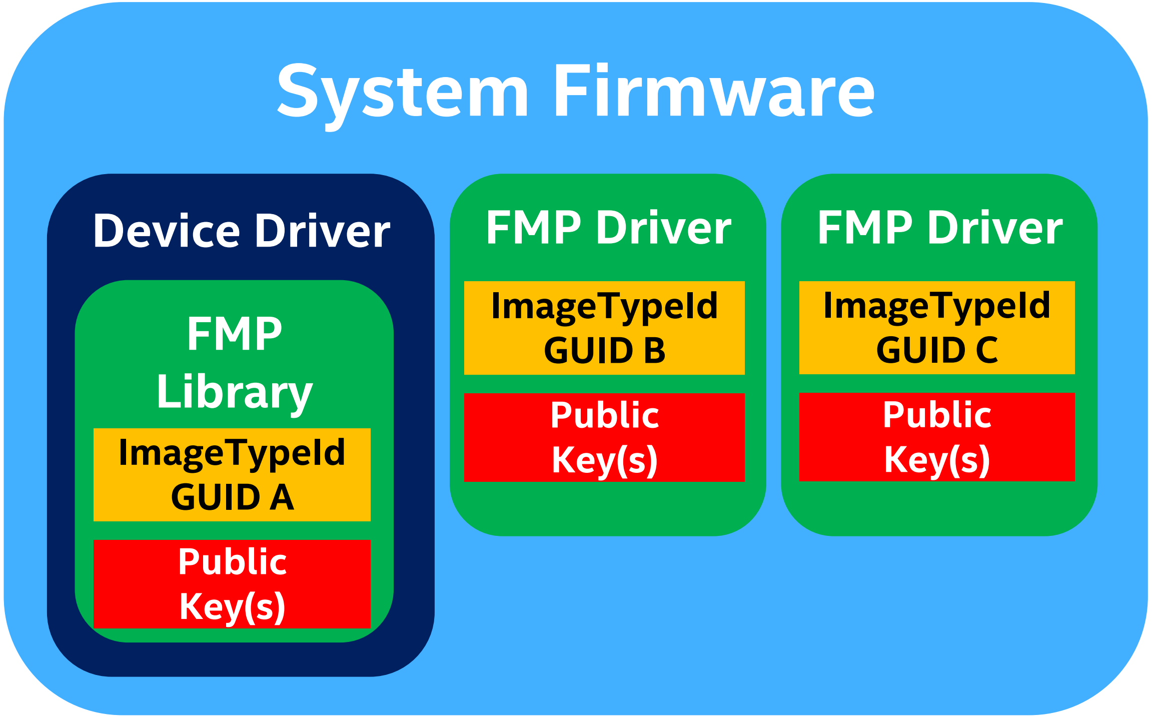
3rd Party UEFI Capsules must be resigned with System Key

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 Table
GUID A
GUID B
GUID C

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 with `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.

Legal Disclaimer

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request. No product or component can be absolutely secure.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.

Intel, the Intel logo, Intel Firmware Support Package (Intel FSP), Intel Server Platform Services (Intel SPS), Intel Slim Bootloader, and Intel Trusted Execution Technology (Intel TXT) are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

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