

OPEN

Compute Engineering Workshop

March 9, 2015
San Jose

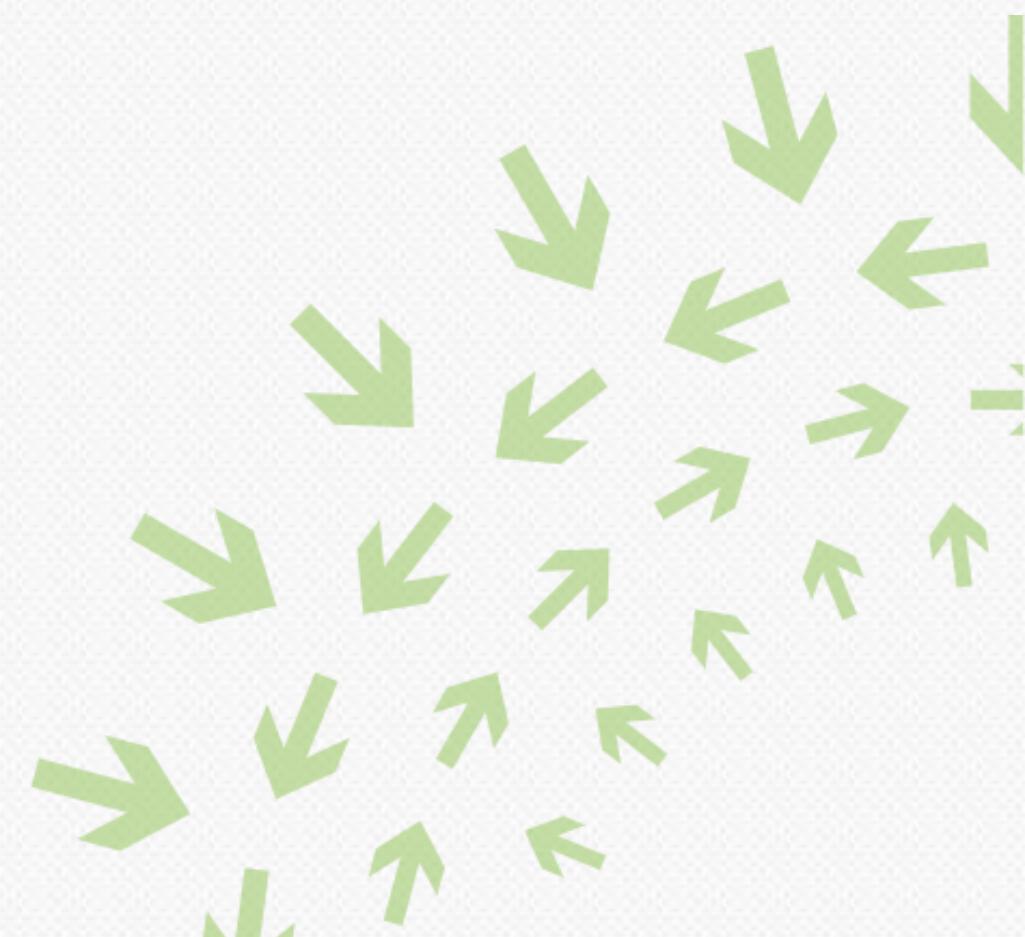


OPEN
Compute Engineering Workshop

Filling UEFI/FW Gaps in the Cloud

Vincent Zimmer
Intel Corporation
Sr. Principal Engineer
vincent.zimmer@intel.com

Mallik Bulusu
Microsoft Corporation
UEFI Lead
mallikb@microsoft.com



ENGINEERING WORKSHOP

Agenda

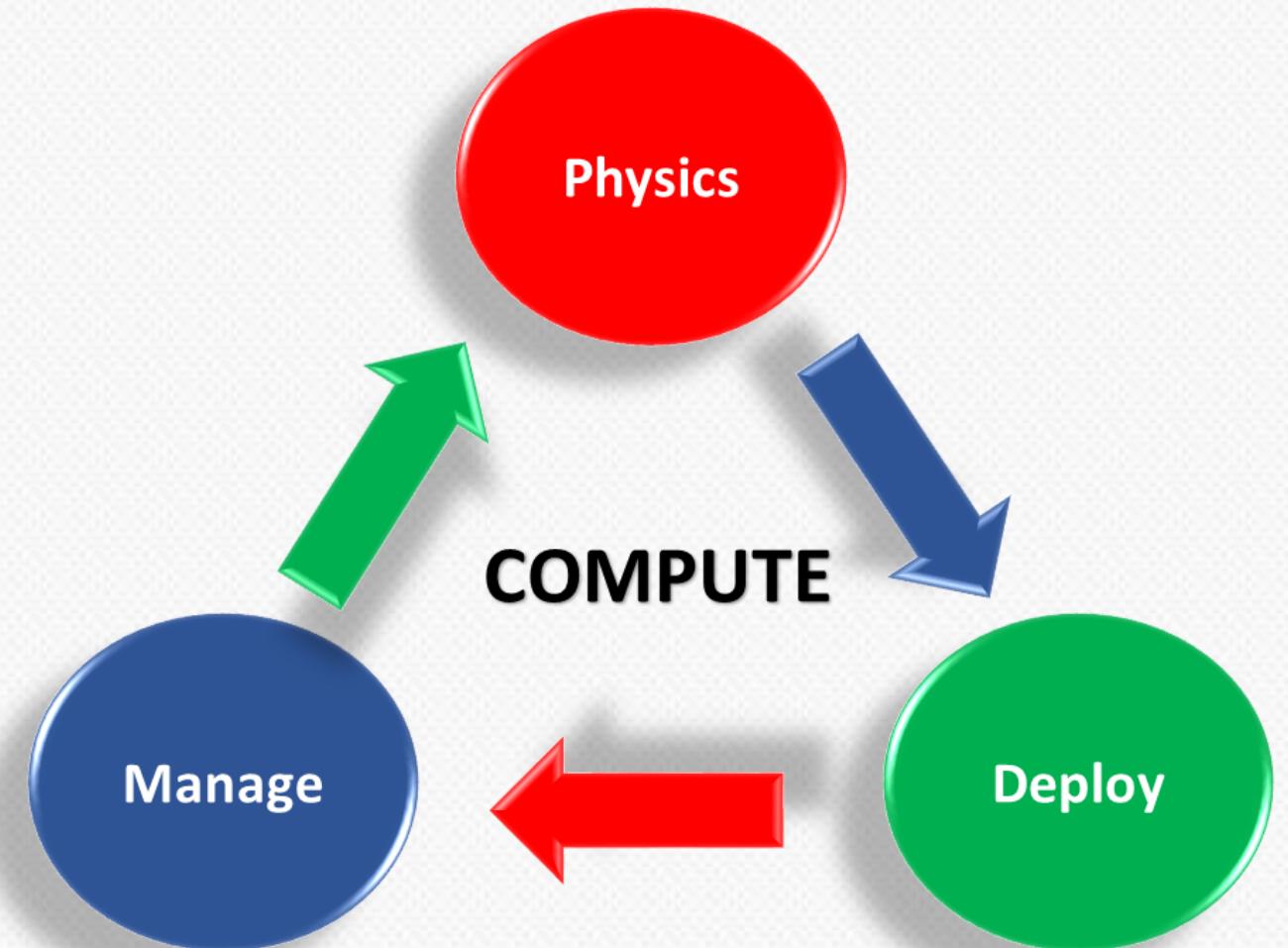
- Cloud Server Firmware Challenges
- Intro to UEFI
- Firmware Update
- Provisioning
- Tools & Diagnostics
- Security
- Conclusion



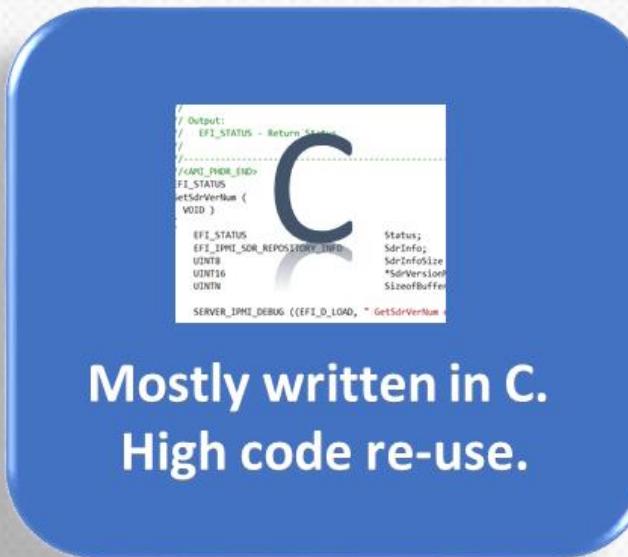
Introduction

Firmware Challenges in the Cloud

- Design constraints for the Compute –
 - Physics challenge?
 - How to **deploy** the compute engine?
 - How to **manage** the compute?
- Key Cloud Firmware Challenges
 - Firmware Updates
 - Bare Metal Provisioning
 - Security
 - Tools & Diagnostics



Why UEFI?



Mostly written in C.
High code re-use.



Emphasis on
Specifications.
Standards compliance.



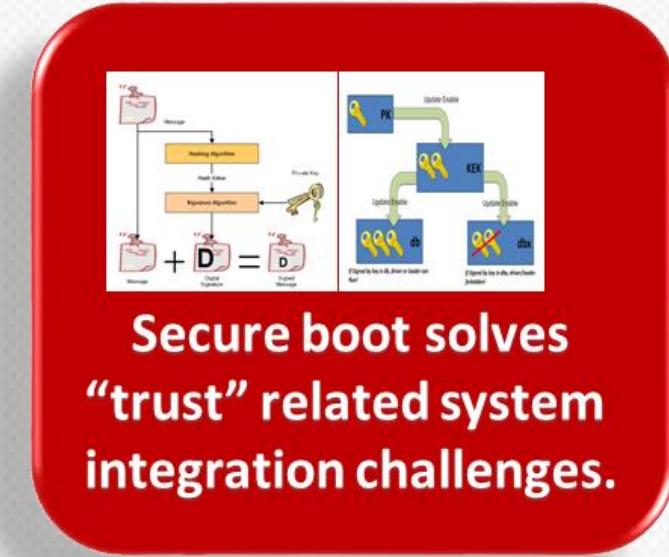
Better platform
scaling. For e.g. removes
shadow ROM limits.



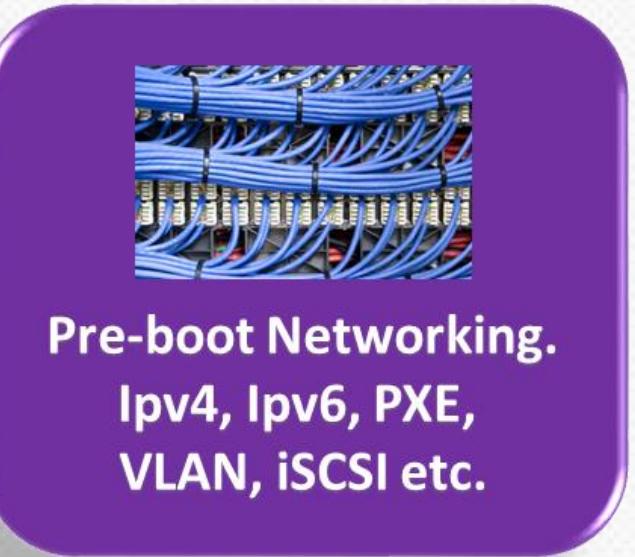
Storage.
GPT removes 2.2 TB
MBR restriction.



CPU Architecture
independent. Platform
design flexibility.



Secure boot solves
“trust” related system
integration challenges.



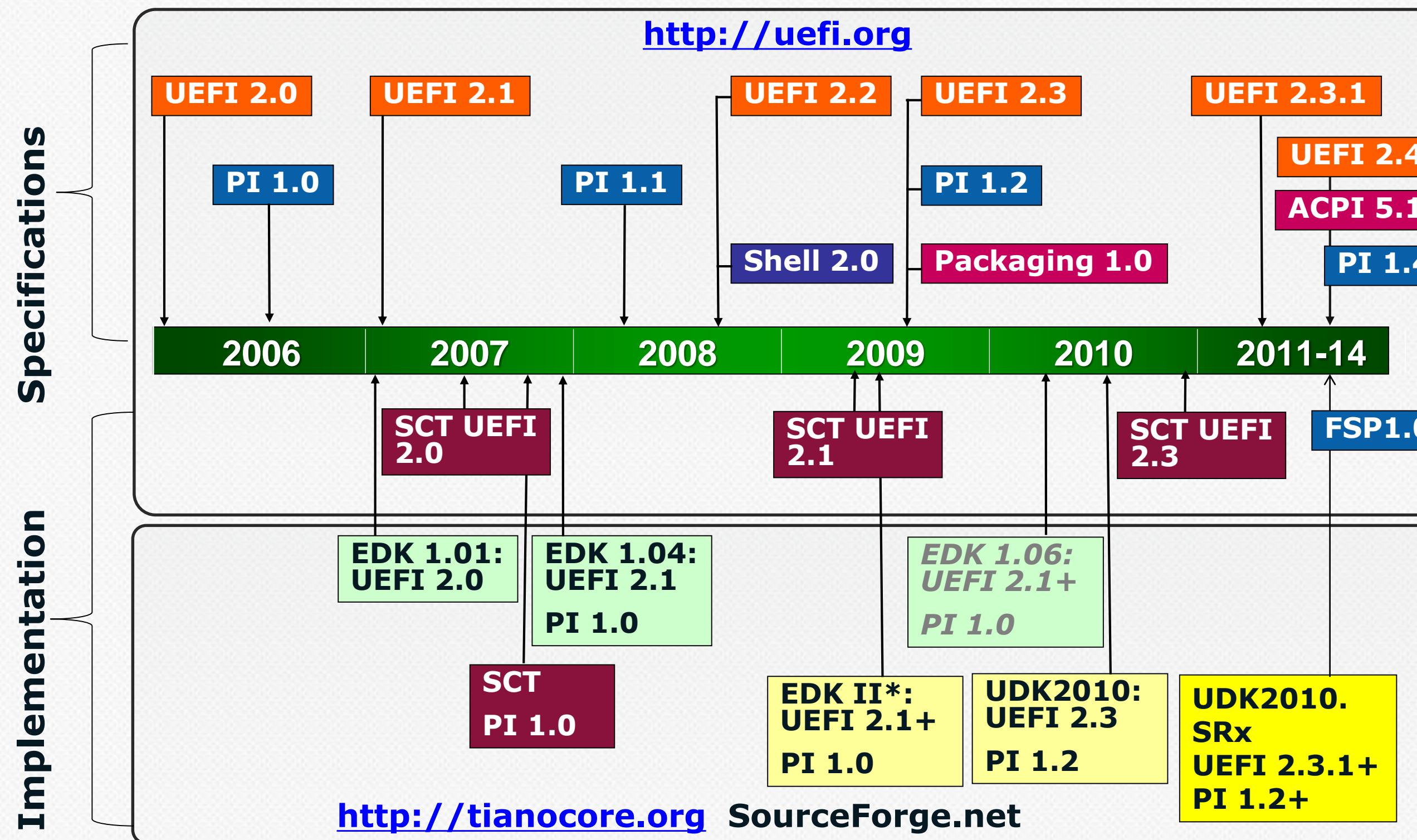
Pre-boot Networking.
Ipv4, Ipv6, PXE,
VLAN, iSCSI etc.



UEFI shell improves
pre-boot testing &
diagnostics experience.



Specification & Tianocore.org Timeline



Building UEFI

Industry Standards Compliance

- UEFI 2.0, UEFI 2.1, UEFI 2.2, UEFI 2.3, UEFI 2.4; PI 1.0, PI 1.1, PI 1.2, PI 1.3; ACPI 5.1

Extensible Foundation for Advanced Capabilities

- Pre-OS Security
- Rich Networking
- Manageability

Support for UEFI Packages

- Import/export modules source/binaries to many build systems

Maximize Re-use of Source Code**

- Platform Configuration Database (PCD) provides “knobs” for binaries
- ECP provides for reuse of EDK1117 (EDK I) modules
- Improved modularity, library classes and instances
- Optimize for size or speed

Multiple Development Environments and Tool Chains**

- Windows, Linux, OSX
- VS2003, VS2005, WinDDK, Intel, GCC

Fast and Flexible Build Infrastructure**

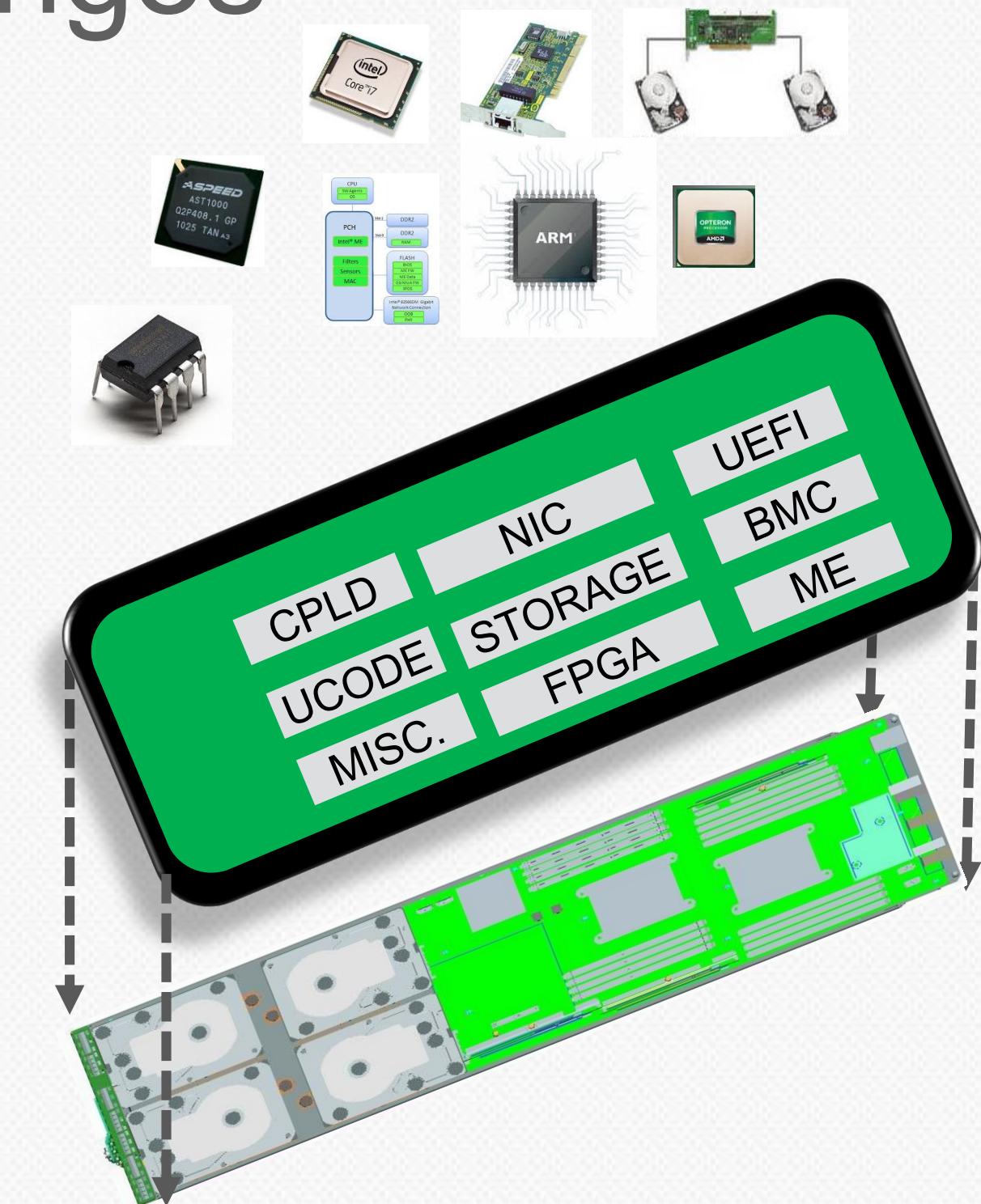
- 4X+ Build Performance Improvement (vs EDK I)
- Targeted Module Build Flexibility

* benefit of EDK II codebase

Firmware Updates

Firmware Update Challenges

- Components from multiple vendors
- Delivering firmware
- Different types of devices
- Recovery from failures
- Node equivalence across datacenter
- Security, security, security.....

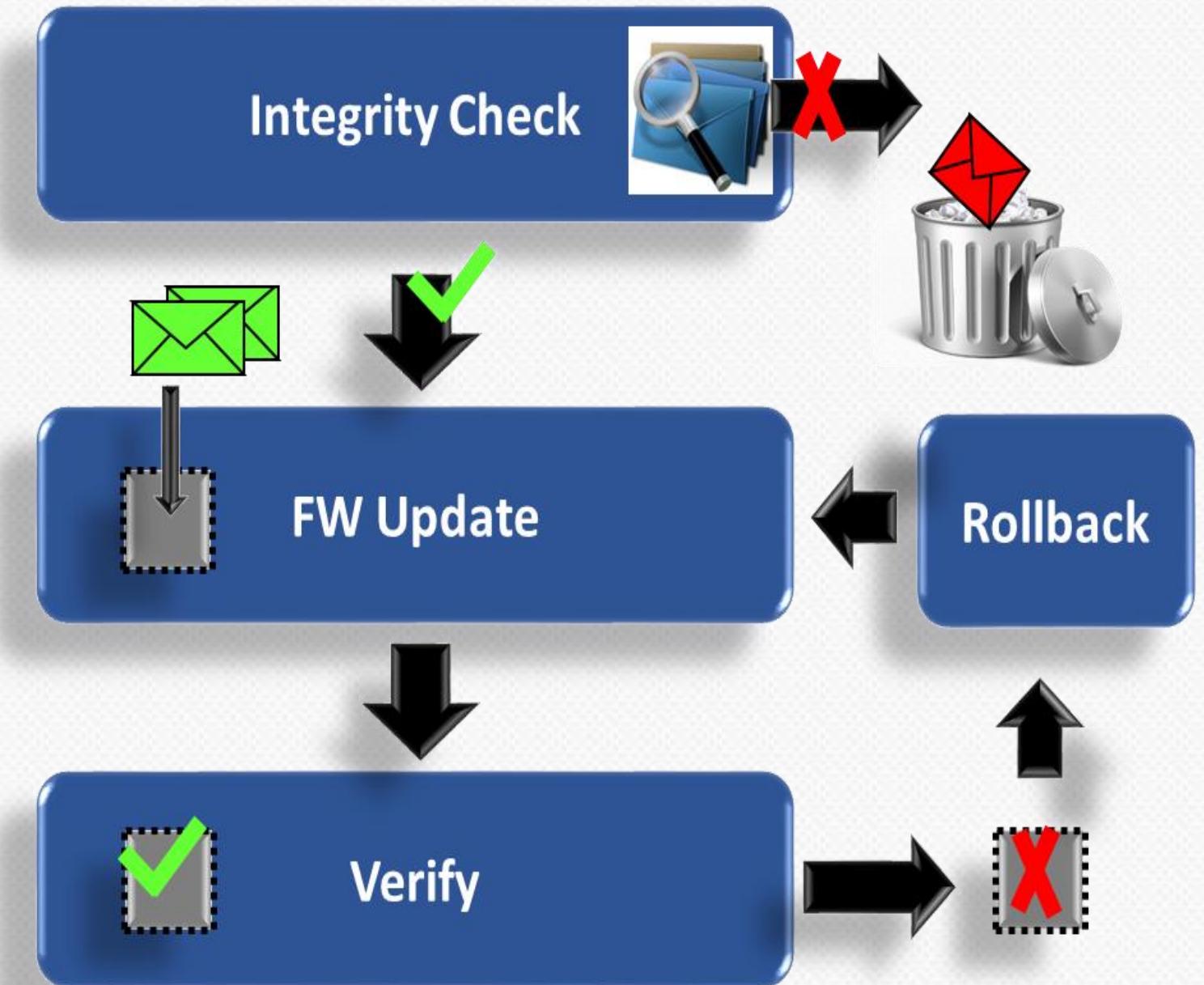


• Note: BMC based FW updates not covered here



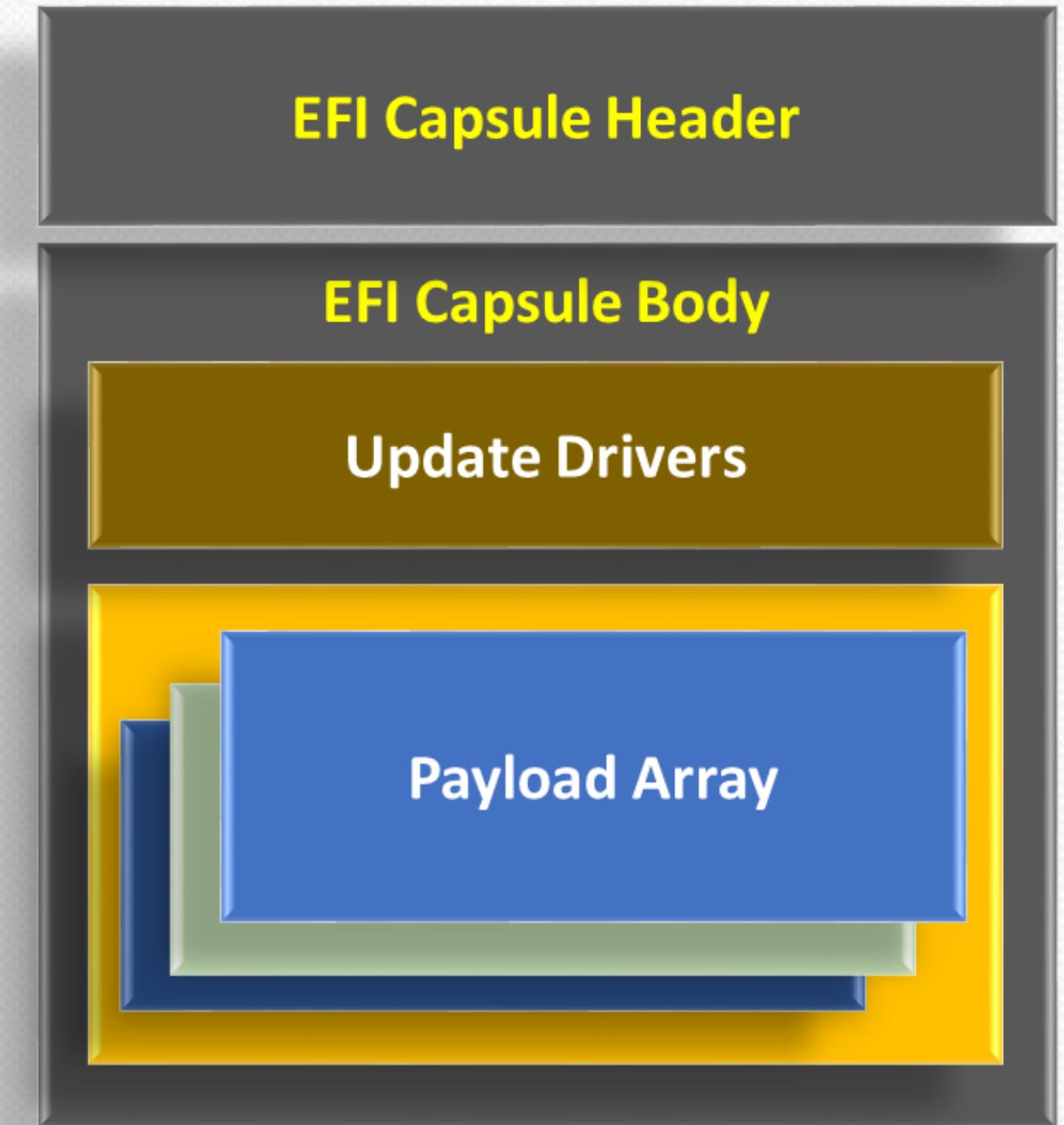
Solving Firmware Update

- 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



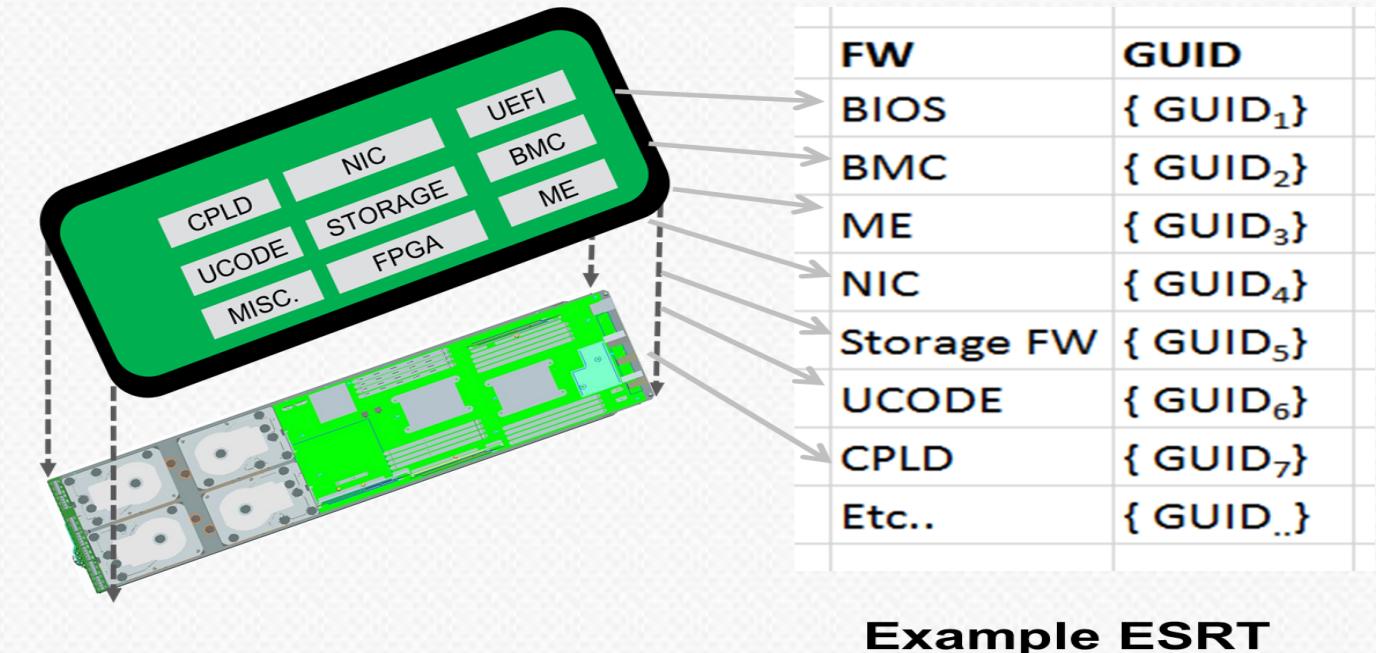
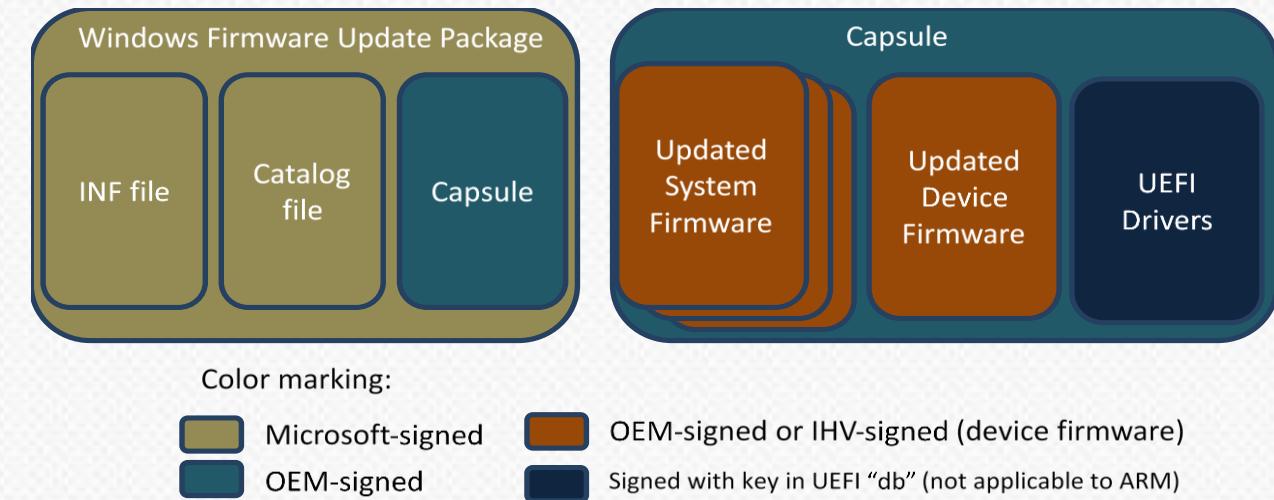
Delivering Firmware Binaries

- UEFI supports Capsule format
 - Tools for capsule generation
 - Core logic for capsule handling
- Extensible Capsule format
 - Self-contained
 - Discrete updates
 - Composite updates
- Firmware Management Protocol allows
 - Reading / updating firmware
 - Integrity checks



EFI System Resource Table

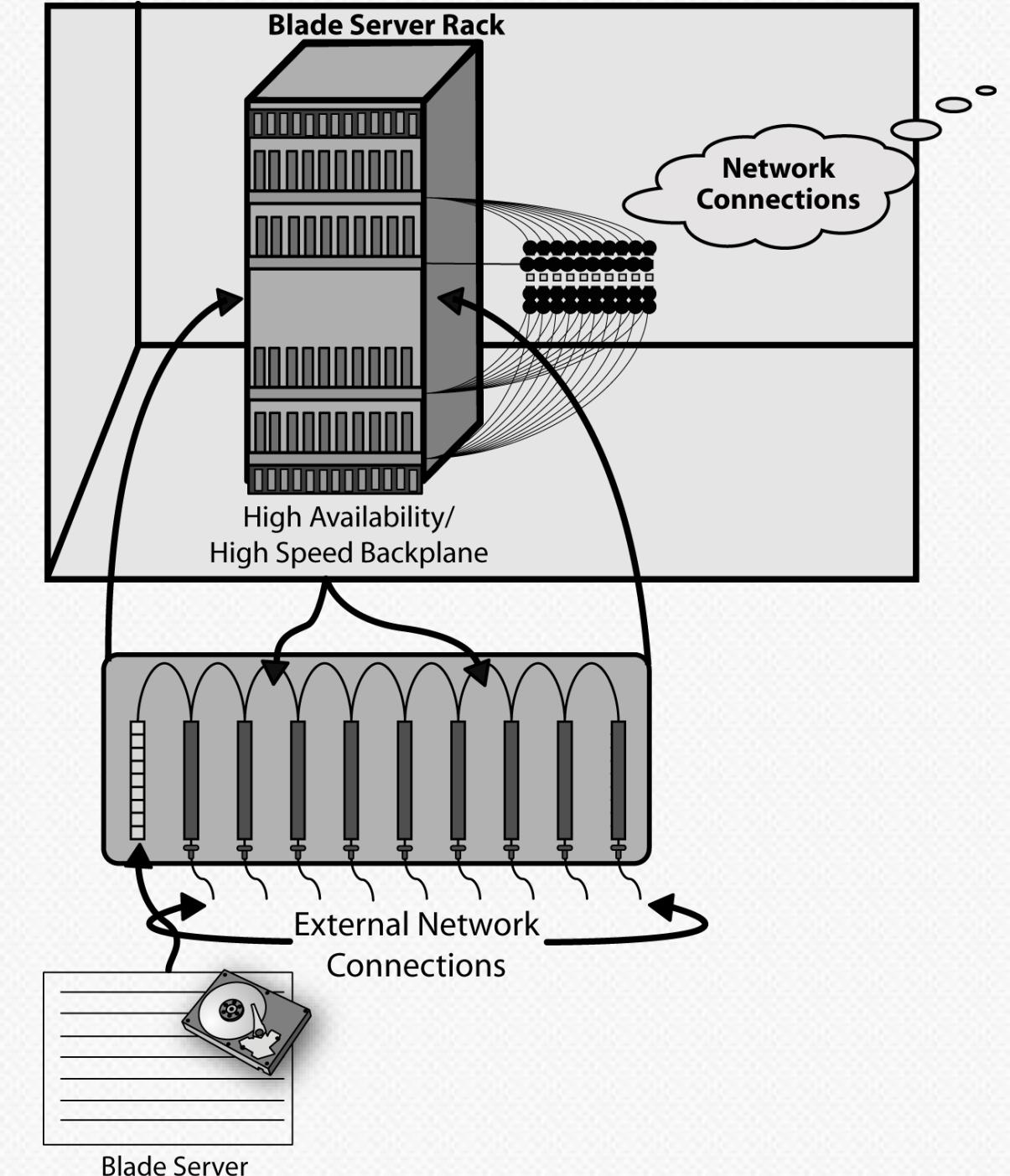
- Update types
 - Largely OS assisted
 - Largely BIOS assisted
- FW updateability rules can be encoded into the capsule
 - Least version
 - Signing
- Describe various updateable components on the platform



Bare Metal Provisioning

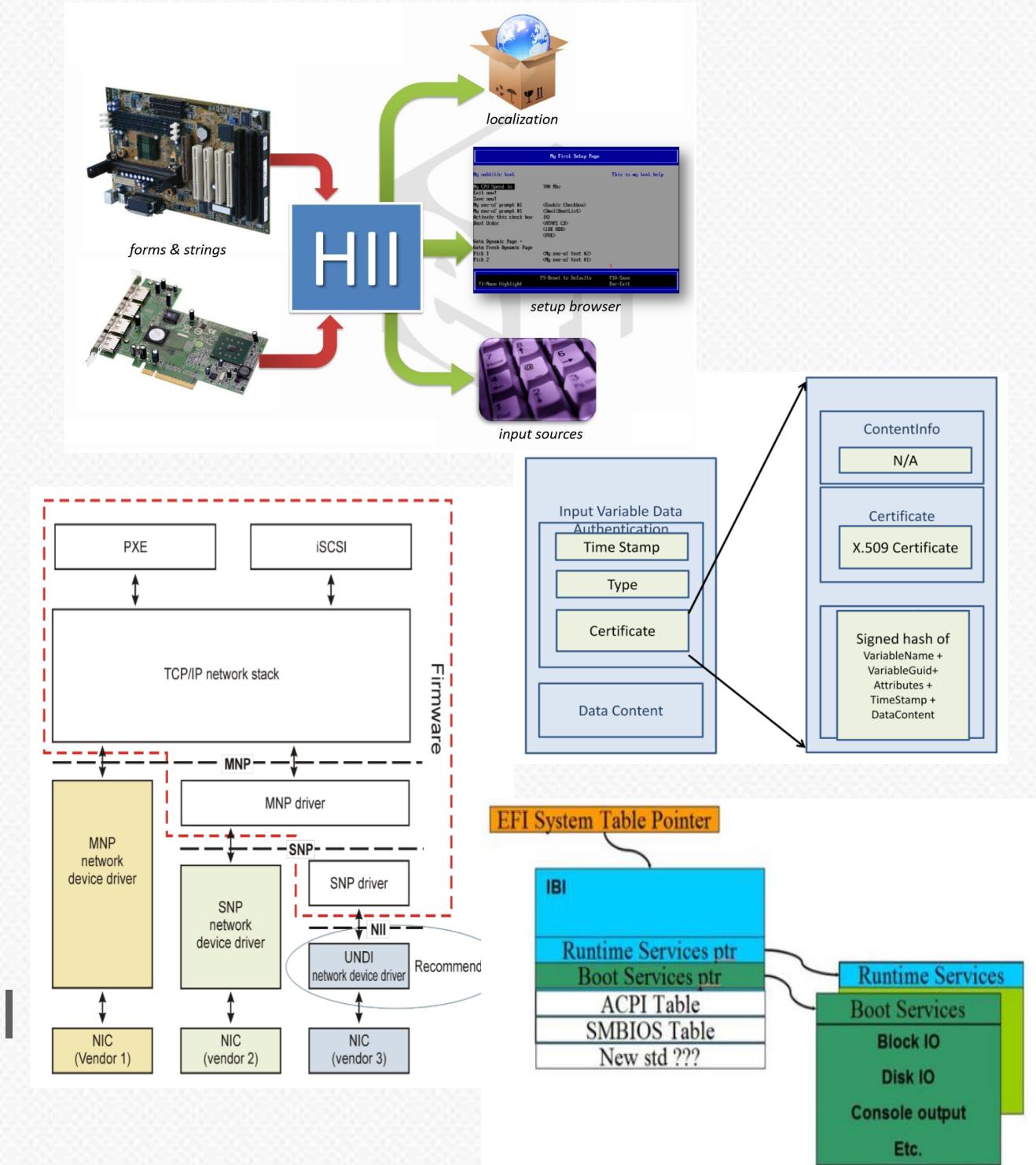
Bare Metal Provisioning Challenges

- Hardware Detection
- Installation
 - Local / Remote
- Configuration
 - Local / Remote / Scriptable
- Cloning
 - Automated
- Backup / Recovery
 - Local / Remote / Automated



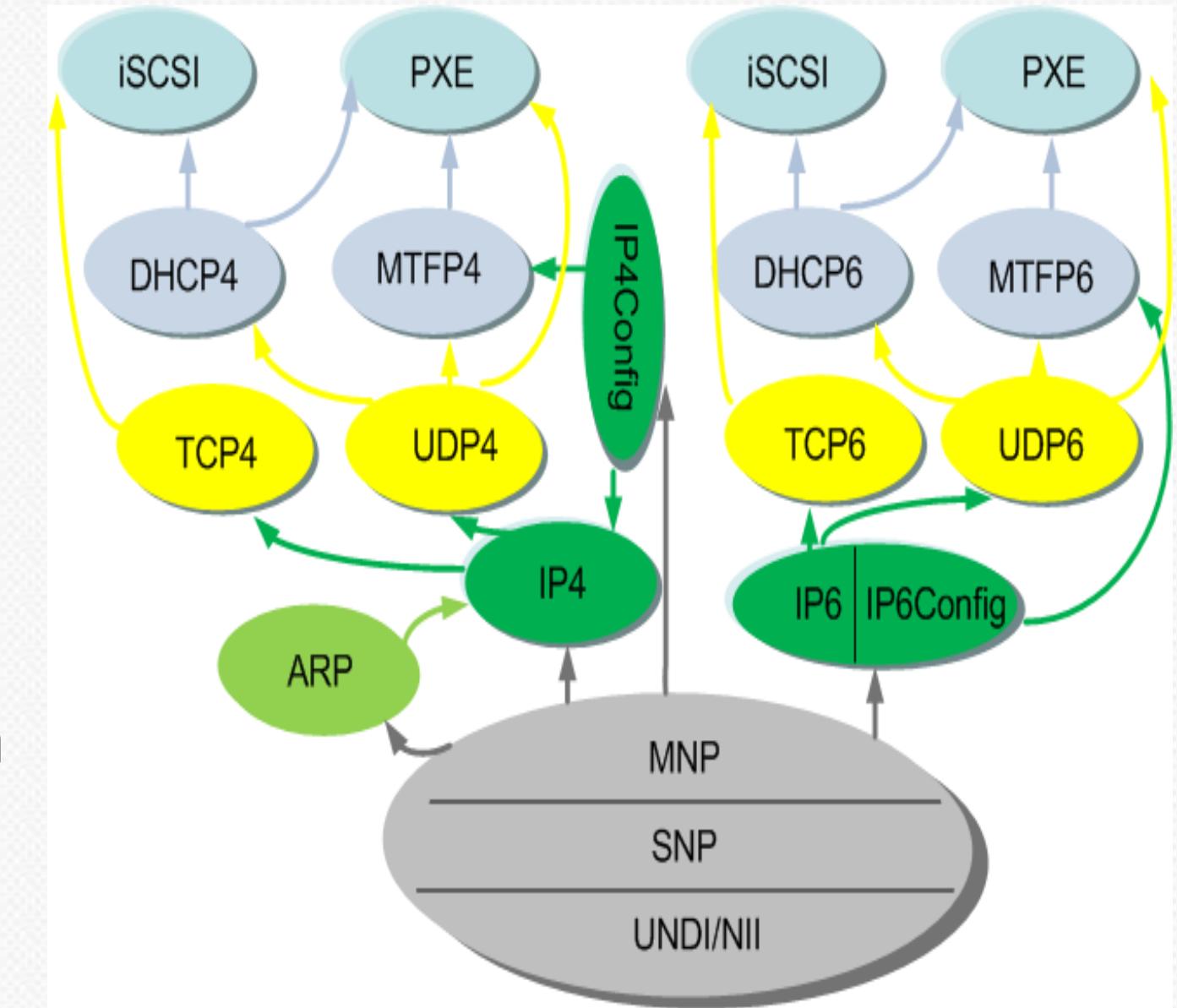
Bare Metal Provisioning Solutions

- Need a ‘no-touch’, automated installation mechanism
 - Repurpose / Configure / Recover
- HII and IFR for consistent & scriptable configuration
- Non-blocking local disk and networking services for high throughput image delivery and recovery
- UEFI Variables for booting and Authenticated Variables for safe storage of settings, like UEFI secure boot database



Networking in UEFI

- UEFI offers rich set of Networking Features during pre-boot
 - PXE boot support for network boot, OS installations, provisioning etc.
 - Native support for IPv4 as well as IPv6
 - Network file system support
 - Virtual LAN support, iSCSI
 - IpSec for supporting secure communication
- Evolution of networking –
 - RFC 5970 allows for ‘boot from URI’
 - HTTP, NFS,...





Security

Attacks on Firmware

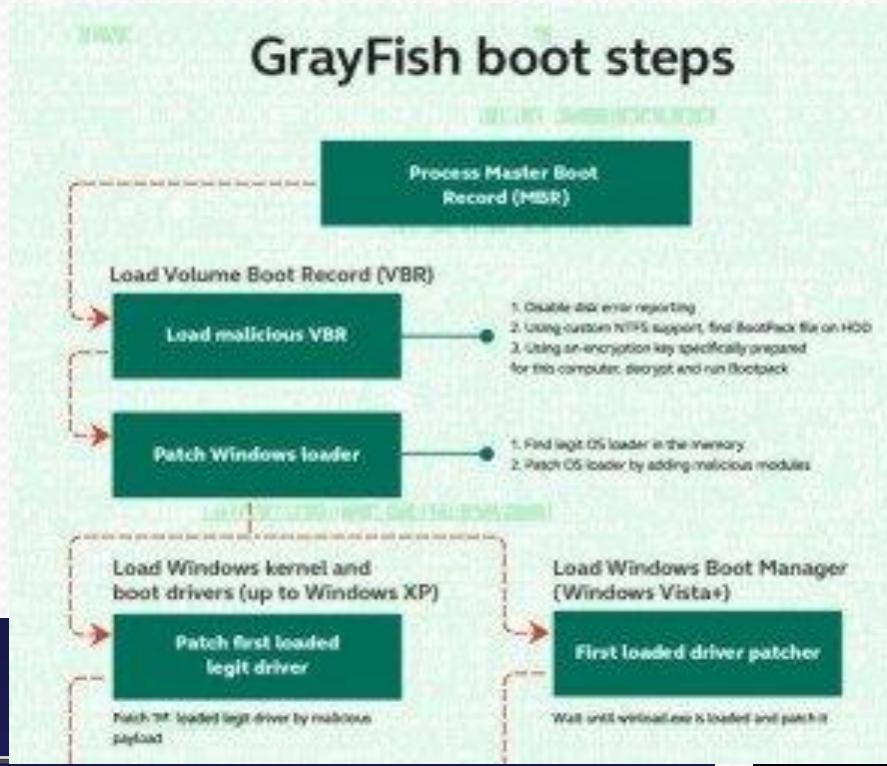
Hacking the Extensible Firmware Interface



John Heasman, Director of Research

Code Injection Attacks

- Important when firmware verifies digital signatures
 - Depends on implementation flaw in driver
 - e.g. stack overflow, heap overflow
 - or incorrect signature verification
- Plenty of targets:
 - File system drivers (e.g. FAT32, HFS+)
 - PE parsing code
 - Crypto code (Data in certs, ASN.1 decoding)
 - Network interaction (PXE)



Stephen Cobb, senior security researcher at ESET North America, says that hacking firmware can be particularly effective because it is so hard to eliminate.

It's also particularly challenging to do, says Jean Taggart, security researcher at Malwarebytes.

"Doing this on just one brand of hard drive would be an almost Herculean task," he says. "You have to understand the hardware as well—if not more—than the original manufacturer."

[Stan Alcorn, Marketplace. Feb 17, 2015](#)



DE MYSTERIIS DOM JOBSIVS: MAC EFI ROOTKITS

SNARE
@ SYSCAN SINGAPORE
APRIL 2012



IN CONCLUSION...
I HAD FUN.

► So basically we're all screwed

- ▶ What should you do?
 - ▶ Glue all your ports shut
 - ▶ Use an EFI password to prevent basic local attacks
 - ▶ Stop using computers, go back to the abacus
- ▶ What should Apple do?
 - ▶ Implement UEFI Secure Boot (actually use the TPM)
 - ▶ Use the write-enable pin on the firmware data flash properly
 - ▶ NB: They may do this on newer machines, just not my test one
 - ▶ Audit the damn EFI code (see Heasman/ITL)
 - ▶ Sacrifice more virgins

De Mysteriis Dom Jobsivs - SyScan

April, 2012



Network Nightmare

Ruling the nightlife between shutdown and boot with pxesplloit

Bootkits

Stoned Bootkit

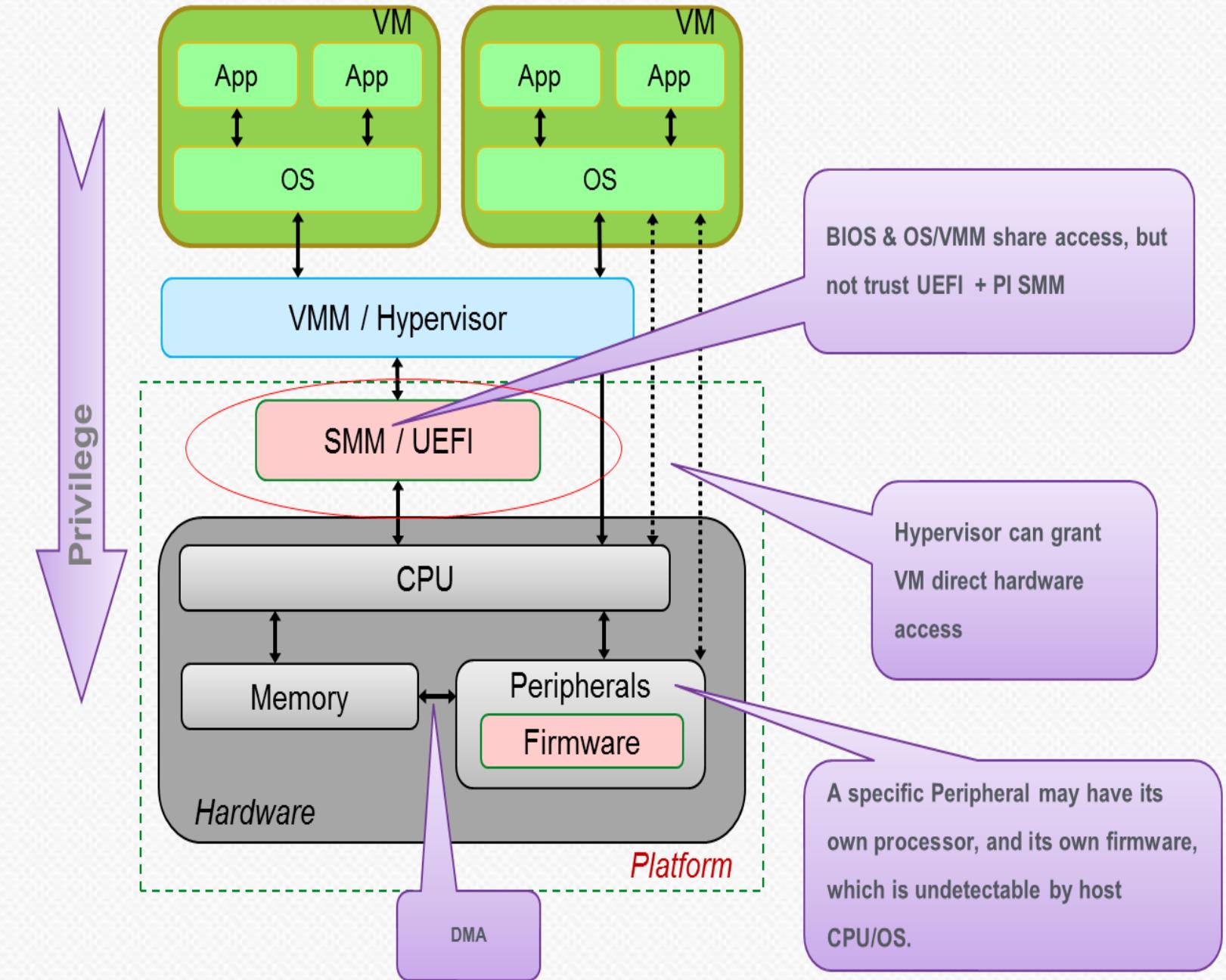


ENGINEERING WORKSHOP

Peter Kleissner

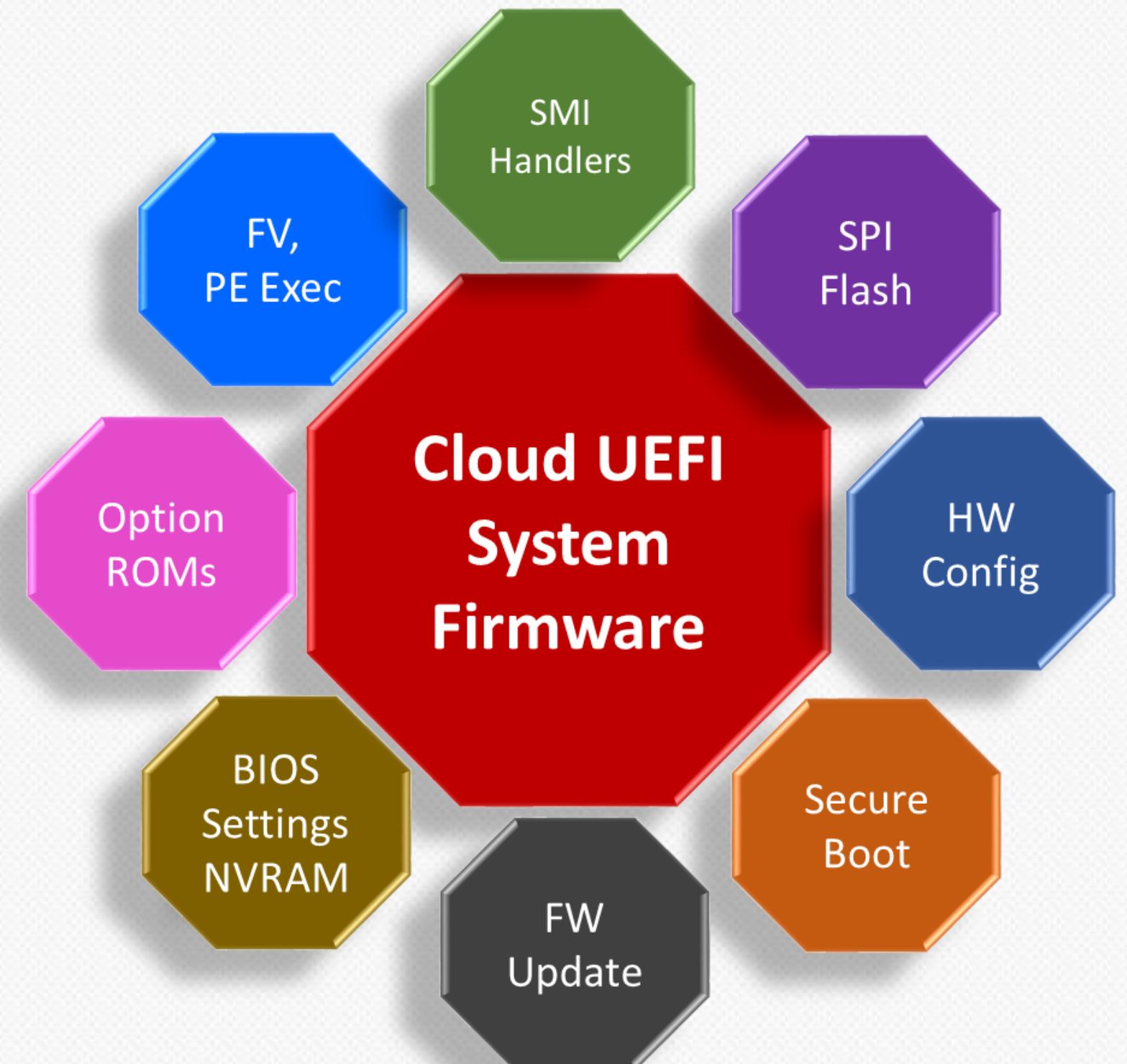
Security Challenges

- Different elements in platform from many vendors
- How to establish trust anchor in the hardware
- How to protect elements
- How to protect the platform
- How to allow platform scaling



Security Solutions

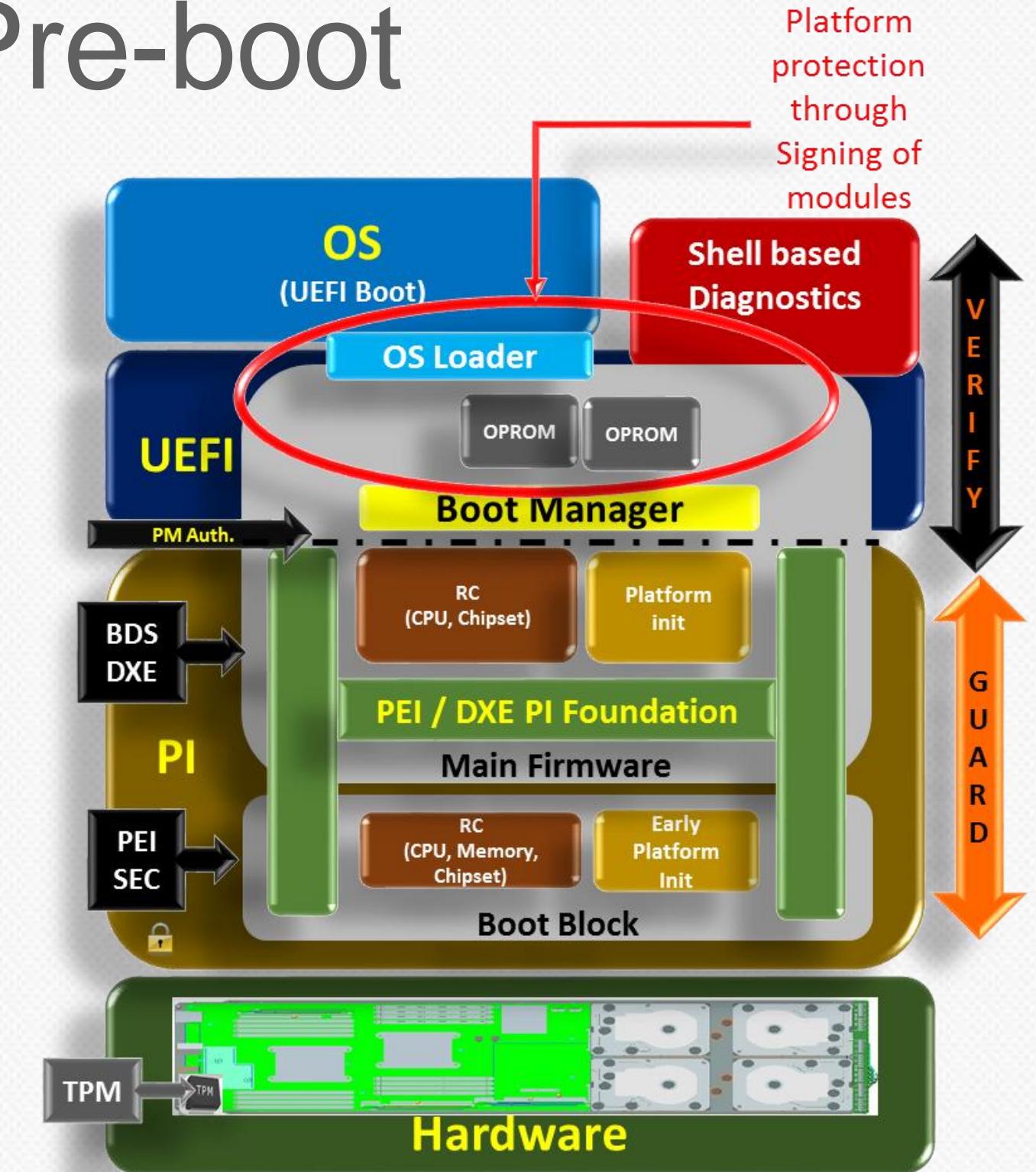
- Signed capsule updates
- UEFI Secure boot
 - local / network
- TPM on the platform
 - Measured boot
 - Root of Trust for Reporting
 - Storage
- Protect machine configuration & UEFI Secure boot trust anchors



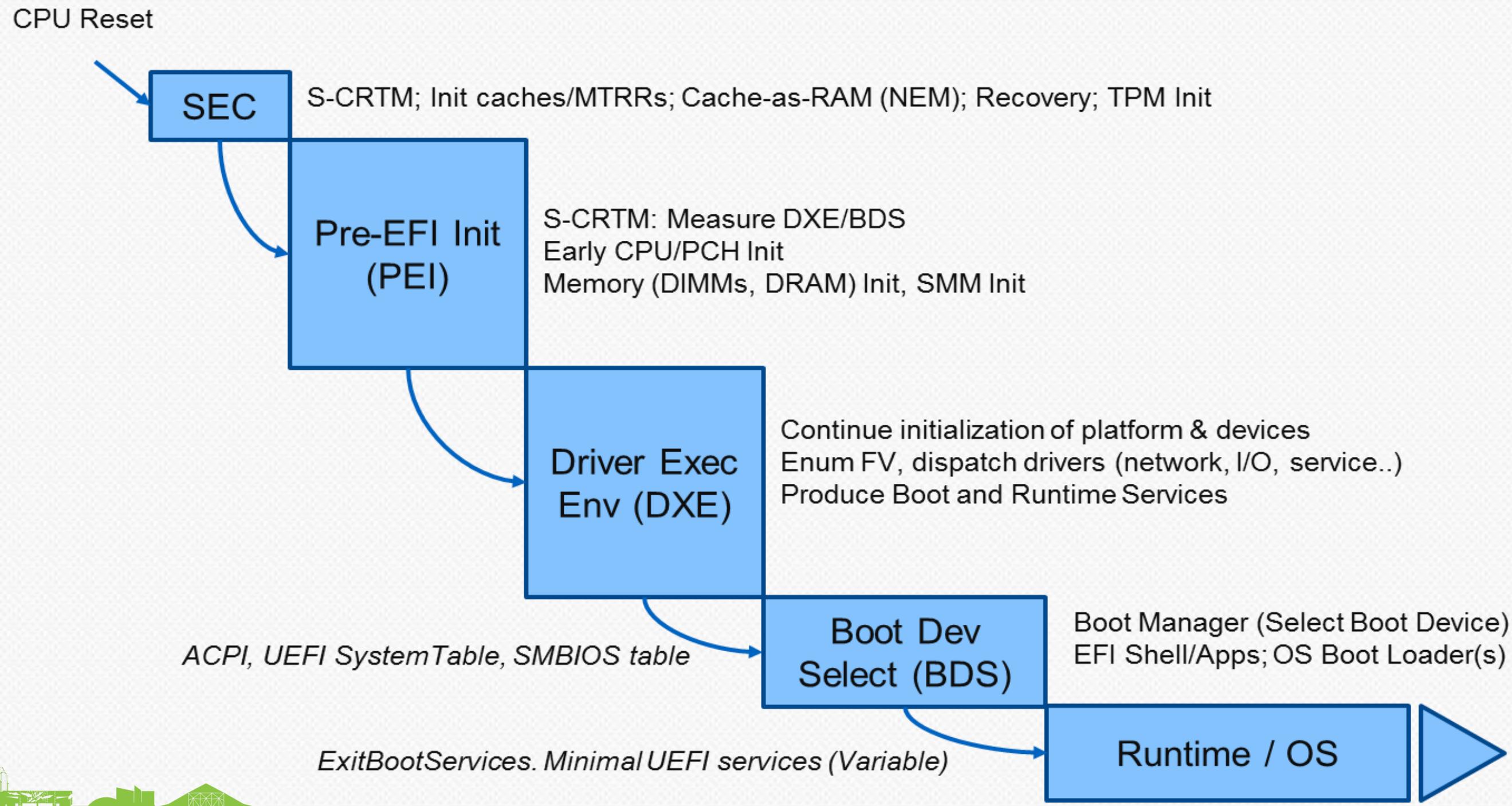
In-band and out-of-band network security

Guarding & Verifying in Pre-boot

- PI & UEFI complement each other to impart **platform security** through guarding and verification during pre-boot.
- PI facilitates **platform hardening** by guarding internal firmware ingredients that consume reset vector, initialization of CPU, Memory, Chipset etc.
- UEFI signing allows **robust platform scaling** through verified inclusion of external firmware ingredients such as OPROMS into the trust chain



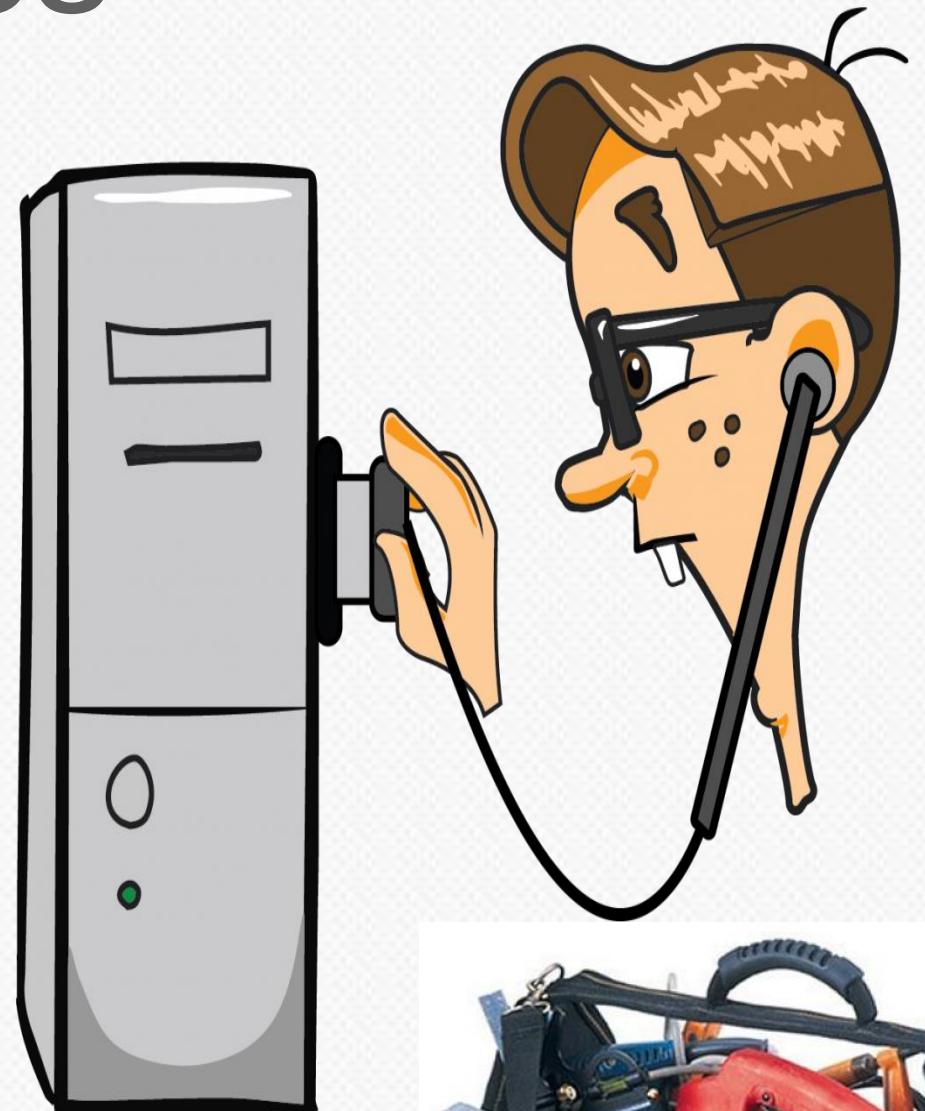
Recommended UEFI Boot Flow



Tools & Diagnostics

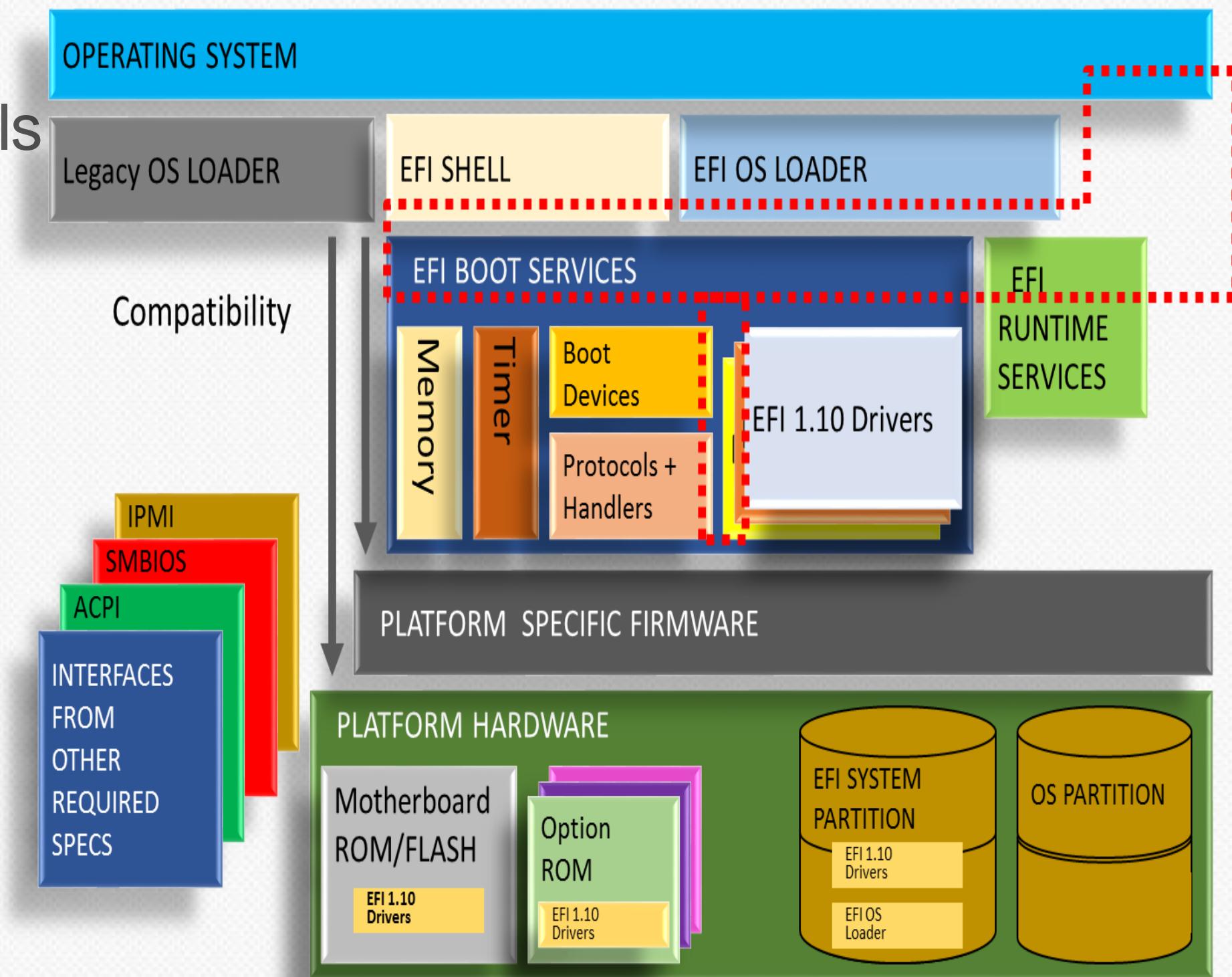
Tools & Diagnostic Challenges

- Platform ingredients from many vendors
- How to assess health, security, compliance of the elements
- Consistent environment to run diagnostics
 - Log / Report / Journal results
- Recovery agent considerations
 - Local / Remote / In-band / Out-of-band



Tools Solutions

- Environment for hosting tools
 - UEFI Shell
 - Linux UEFI Validation project
- Tools for deployment
 - UEFI SCT
 - PI SCT
 - ACPI Compliance
 - SMBIOS Compliance
 - Security
 - Chipsec
 - Copernicus
 - Selftest

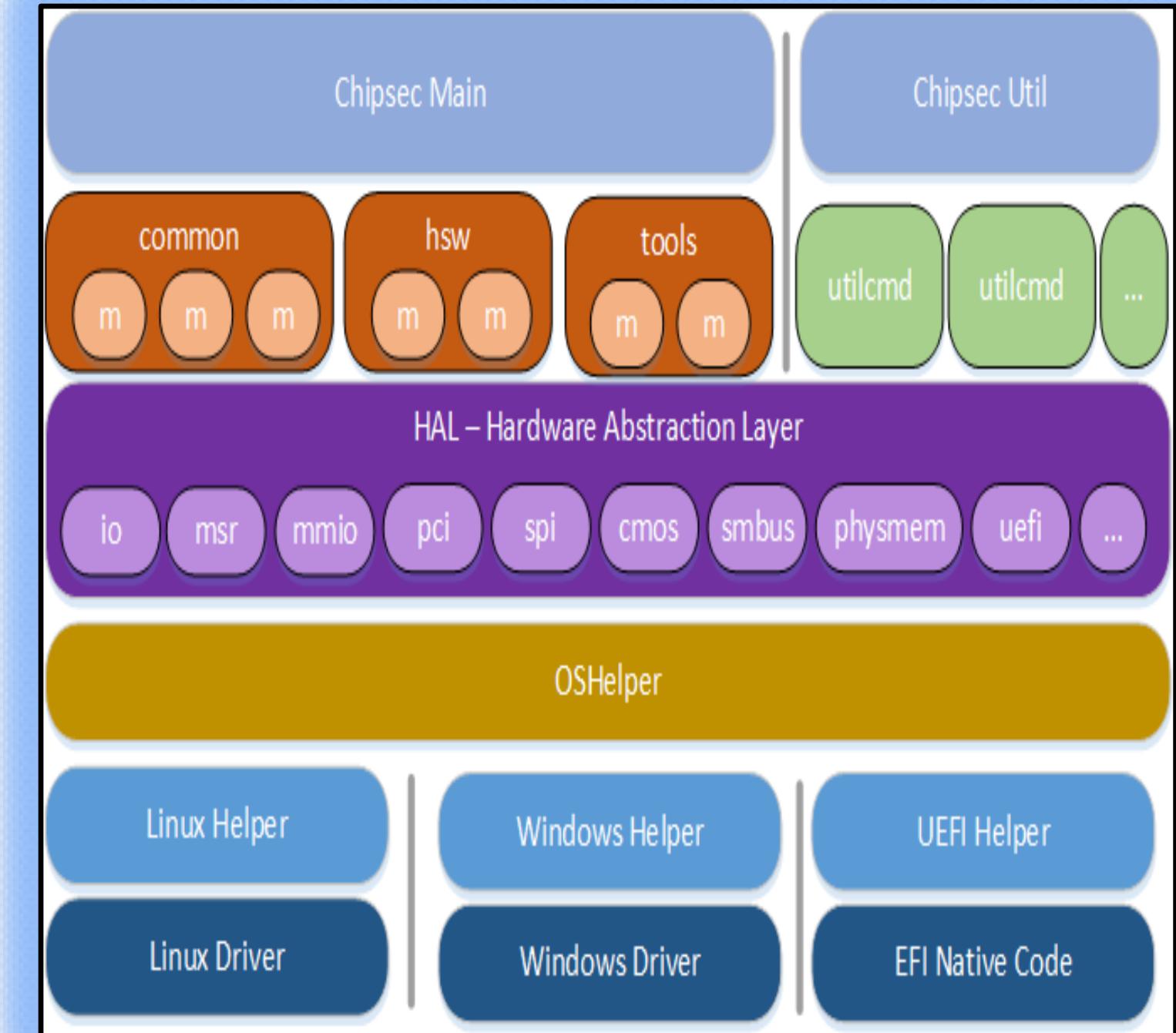


chipsec Tool

- Essentially a platform security assessment framework for risk assessment
- Can be extended to meet specific platform security concerns
- Open sourced
 - <https://github.com/chipsec/chipsec>
- Supported Environments
 - Windows
 - Linux



UEFI (over Python)

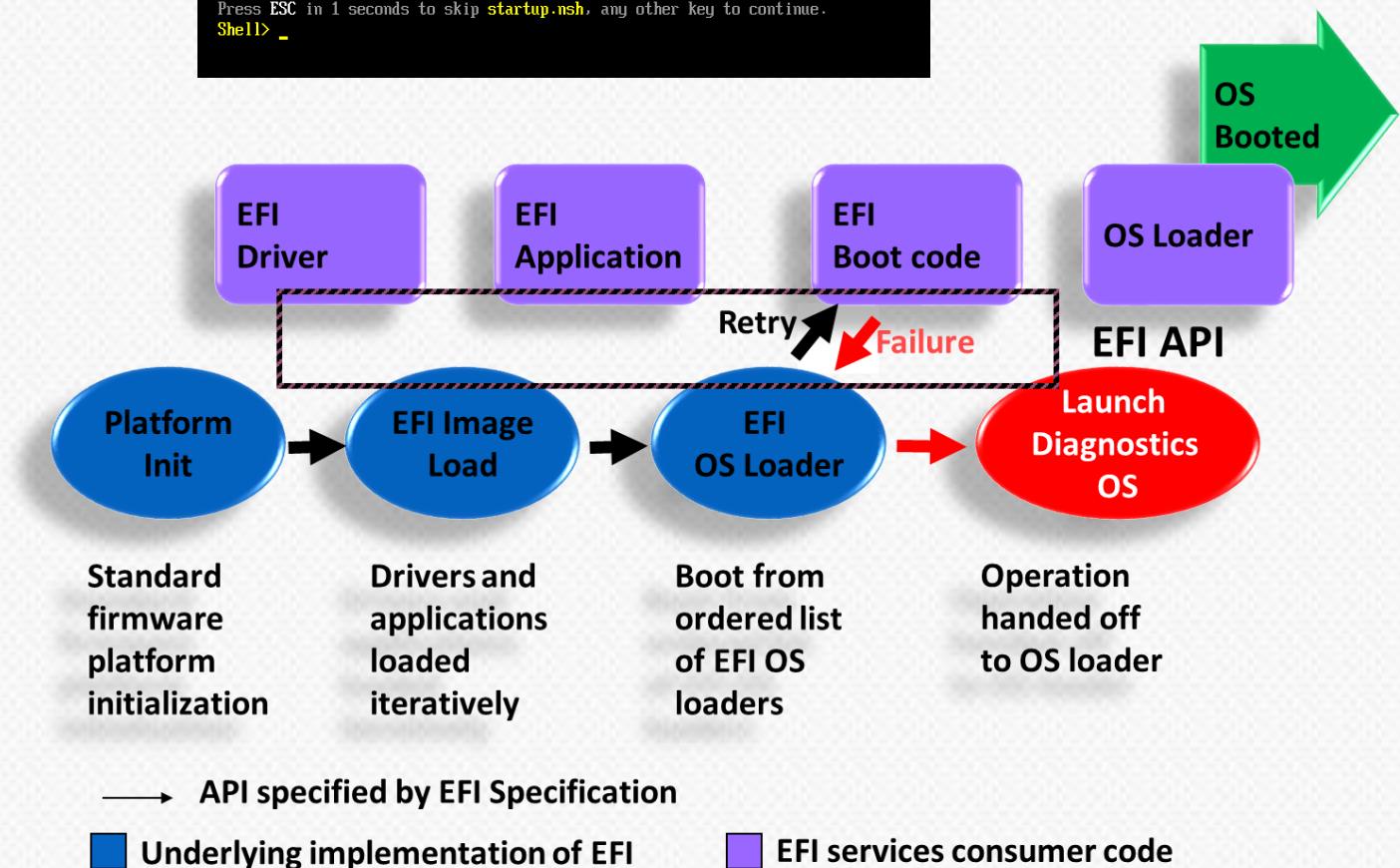


Diagnostics Solutions

- Once in UEFI, how to assess, probe, and prod the system
 - Type15 SMBIOS Records
 - Dmpstore for UEFI variables, incl WHEA variable
 - ACPI CA for executing/dumping/viewing namespace
 - UEFI shell to run above, redirect output to file or ‘virtual file’ (e.g., volatile variable)
 - PCI command to read/write/assess hardware state. Scriptable too
 - Results can be installed in UEFI system table like other hand-off info, or variable, or file on ESP, or sent across the network using UEFI network stack

```
EFI Shell version 2.00 [4096.1]
Current running mode 1.1.2
Device mapping table
fs0 :Removable HardDisk - Alias hd52g0b blk0
      Acpi (PNP0a03,0)/Pci (1D|7)/Usb (6,0)/HD (Part1,Sig90909090)
blk0 :Removable HardDisk - Alias hd52g0b fs0
      Acpi (PNP0a03,0)/Pci (1D|7)/Usb (6,0)/HD (Part1,Sig90909090)
blk1 :HardDisk - Alias (null)
      Acpi (PNP0a03,0)/Pci (1F|2)/Ata (Primary,Master) /HD (Part1,SigD5BAE3BB)
blk2 :HardDisk - Alias (null)
      Acpi (PNP0a03,0)/Pci (1F|2)/Ata (Primary,Master) /HD (Part2,SigD5BAE3BB)
blk3 :BlockDevice - Alias (null)
      Acpi (PNP0a03,0)/Pci (1F|2)/Ata (Primary,Master)
blk4 :BlockDevice - Alias (null)
      Acpi (PNP0a03,0)/Pci (1F|2)/Ata (Secondary,Master)
blk5 :Removable BlockDevice - Alias (null)
      Acpi (PNP0a03,0)/Pci (1D|7)/Usb (6,0)

Press ESC in 1 seconds to skip startup.nsh, any other key to continue.
Shell> -
```



Conclusions & Next Steps

Call to actions

Get involved in the standards

If you're an IHV, implement FMP, Capsules, Security reviews of code

If you're an OSV, build UEFI loader and boot agent

If you're middle ware, leverage HII and shell scripting/config



Conclusion

Cloud has challenges for platform

UEFI for interop

Evolve updates

Provisioning

Diagnostics

Security



More information

www.opencompute.org – OCP specs

www.uefi.org – UEFI, ACPI, Shell, PI Specifications

www.Tianocore.org – open source UEFI

<http://firmware.intel.com> – white papers, training



UEFI Industry Resources

UEFI Forum

Welcome What's New: UEFI Specifications Update!

- UEFI Specification** → Current UEFI Spec: v2.3 approved May, 09 → Current Activities: Implementation and writer's guides
- UEFI Shell Specification** → Current Shell Spec: v2.0, approved Oct, 08 → Current Activities: Implementation support
- PI Specification** → Current PI Spec: v1.2, approved May, 09 → Current Activities: Implementation support
- UTWG Self-test Specification** → Current version: SCTv2.1 released May, 09 → Next Release: v2.3 SCT target mid 2010
- PI Distribution Package Specification** → Current version: v1.0 released May, 09 → Current Activities: Implementation support

www.uefi.org

UEFI Open Source

SourceForge.net - lancore - Windows Internet Explorer

File Edit Favorites Tools Help Convert Select
 Favorites Suggested Sites Suggested Sites Free Hotmail
 SourceForge.net lancore

SOURCEFORGE Register Log In

Visit project lancore

Project Home | SourceForge.net | Help | Log In | Logout | Search

Tags: None
 Latest News
 Archived News
For Members
 Edit Profile
 Contribute
Our Legislate
 BSD License
 Apache License
 FAT32 License
 Community Agreement
Information
 About Us
 Administration
 Getting Started
 FAQ

Page Discuss On View Source History



Introducing UDK2010
 Beginning a new era for the
 UEFI Open Source Community

Welcome

We invite you to the UEFI Open Source Community Master project. This project is the gateway to all open source UEFI development. It contains the latest version of the UEFI 2.0 Specification, the UEFI 2.0 Platform Framework for FDT and UETF. The Unified Extensible Firmware Interface, or UEFI, is a standards-based interface between the operating system and the hardware. It specifies how the system is initialized and how platform services are provided.

To learn more about getting involved in the community see [How To Contribute](#).

Projects

UDK II - Short Summary Statement

The UDK II project is the second implementation of the UDK 2010. The UDK 2010 (UEFI Developers Kit (UDK) based upon current (2010) UEFI specifications) provides support of the currently approved UEFI Specifications in the UOS 1.0 build environment.

Sub-project	Primary	Sourceforge project URL	Download
EDK2-fat-driver	EDK-fat-driver	Http://sourceforge.net/projects/edk2-fat-driver	Download
EDK2-fat-driver	EDK-fat-driver	Http://sourceforge.net/projects/edk2-fat-driver	Download

Get More Information...
 • [UFI - Wikipedia](#)
 • [Hyper-V](#)
Useful Links
 • [The UEFI Forum](#)
 • [Intel](#)
 • [American Megatrends](#)
 • [Microsoft](#)
 • [Phoenix Software Co., Ltd](#)
 • [Phoenix Technologies](#)
 • [Aorus \(brief\)](#)

www.tianocore.org

Intel UEFI Resources

The screenshot shows the homepage of the Intel® UDk Community Resource Center. At the top left is the Intel logo. To its right is a search bar with placeholder text "Search our site for...". Above the search bar are links for "Partners", "Community", and "Support". Below the search bar is a navigation bar with tabs: "Learn", "Communicate", "Collaborate", "Develop", and "Integrate". A "Documentation" dropdown menu is open, showing sub-options like "UDk API Reference", "UDk Examples", "UDk Tutorials", and "UDk Samples". The main content area features a large banner with the text "Welcome to Intel® UDk Community Resource Center" and a subtext "Intel's resources, center of expertise, and partner to the UDk ecosystem for engineers developing the world's first Unified Extensible Firmware Interface (UEFI) Development Kit™ (UDk). Learn more about UDk!". Below the banner are five cards with icons and titles: "Learn", "Communicate", "Collaborate", "Develop", and "Integrate". Each card has a small thumbnail image and a brief description.

Firmware.intel.com

Open Compute Resources



 OPEN
Compute Project

www.opencompute.org

Intel EBC Compiler

INTEL® C Compiler for EFI Byte Code

All prices listed below are Manufacturer Suggested List Prices (MSRP) and subject to Taxes (VAT) or any other state or local taxes or charges.

Customers with 10 or more licenses qualify for the **Intel® Volume Program**. Learn more about the benefits of this program.

Purchase Options

Buy from a reseller
Intel Software Channel Partners have a strong connection to Intel® Software Development Prod. up-to-date technical knowledge about the products.
Find your nearest reseller to get your copy today!

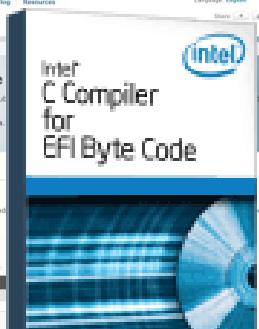
Buy from our Software Store
Click on the "Buy" links in the table below to purchase products directly from our Software Store.

Licenses Types:
Intel® C Compiler for EFI Byte Code

What's Included

Full product:

- New user license and media for Intel® C Compiler for EFI Byte Code
- Technical support, documentation, and more
- One year of support services, which includes technical support (Intel® Premier Support) upgrades and new releases during that term



<http://software.intel.com/en-us/articles/intel-c-compiler-for-efi-byte-code-purchase/>

UEFI Books/ Collateral

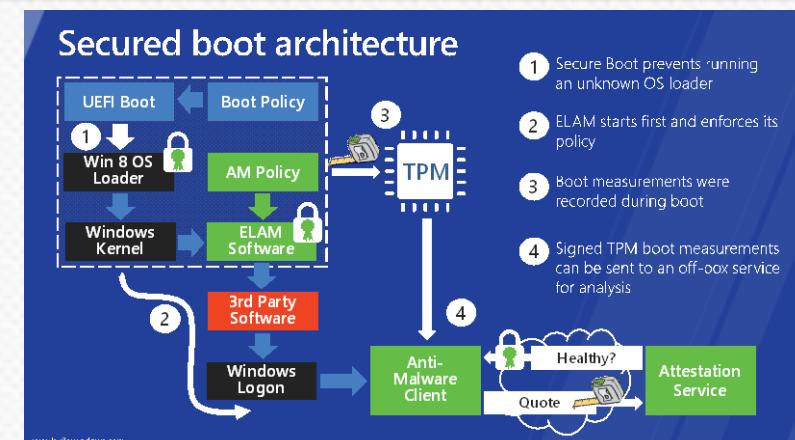


www.intel.com/intelpress

<http://www.intel.com/content/www/us/en/research/intel-technology-journal/2011-volume-15-issue-01-intel-technology-journal.html>

<http://www.apress.com/apressopen>

Microsoft UEFI Resources

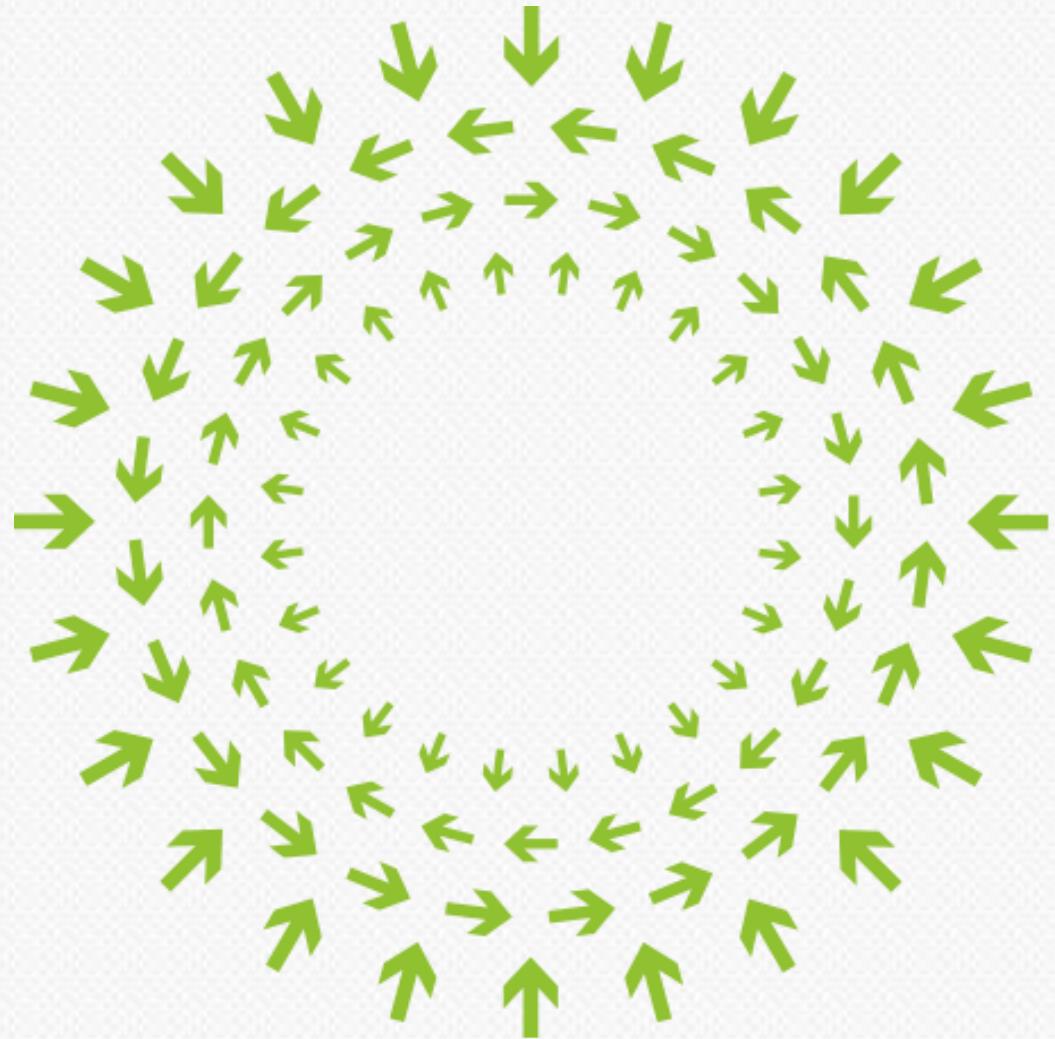


<http://channel9.msdn.com/Events/BUILD/BUILD2011/HW-462T>

<https://msdn.microsoft.com/en-us/library/windows/hardware/hh973604.aspx>

<https://technet.microsoft.com/en-us/library/hh824898.aspx>





OPEN

Compute Engineering Workshop

March 9, 2015
San Jose

