presented by

Filling UEFI/FW Gaps in the Cloud

UEFI Spring Plugfest – May 18-22, 2015

Presented by Mallik Bulusu – Microsoft

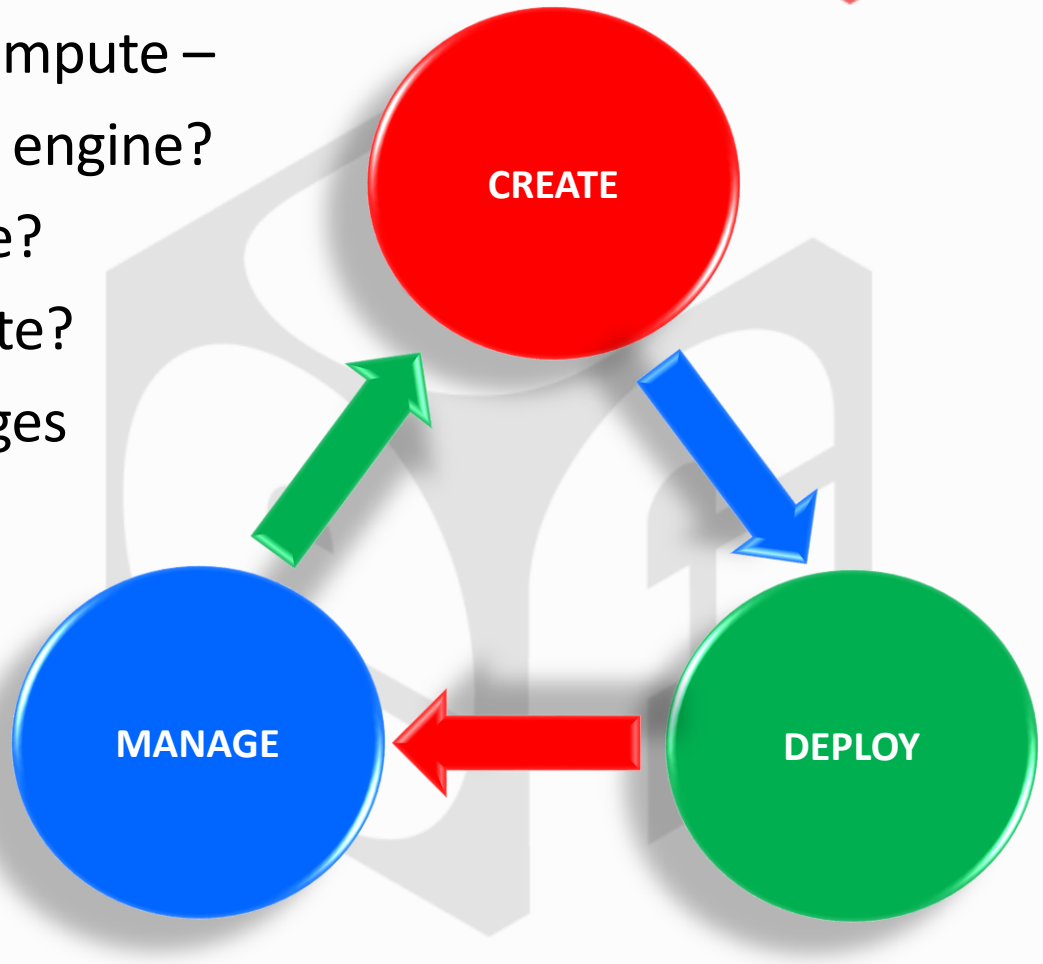
and Vincent Zimmer - Intel

Agenda

- Cloud Server Firmware Challenges
- What is Open Compute
- Intro to UEFI
- Firmware Update
- Provisioning
- Tools & Diagnostics
- Security
- Conclusion

Firmware Challenges in the Cloud

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



What is OCP?

Open Compute: Industry Collaboration

facebook

Open Compute: Industry Collaboration

intel

hyve solutions

facebook

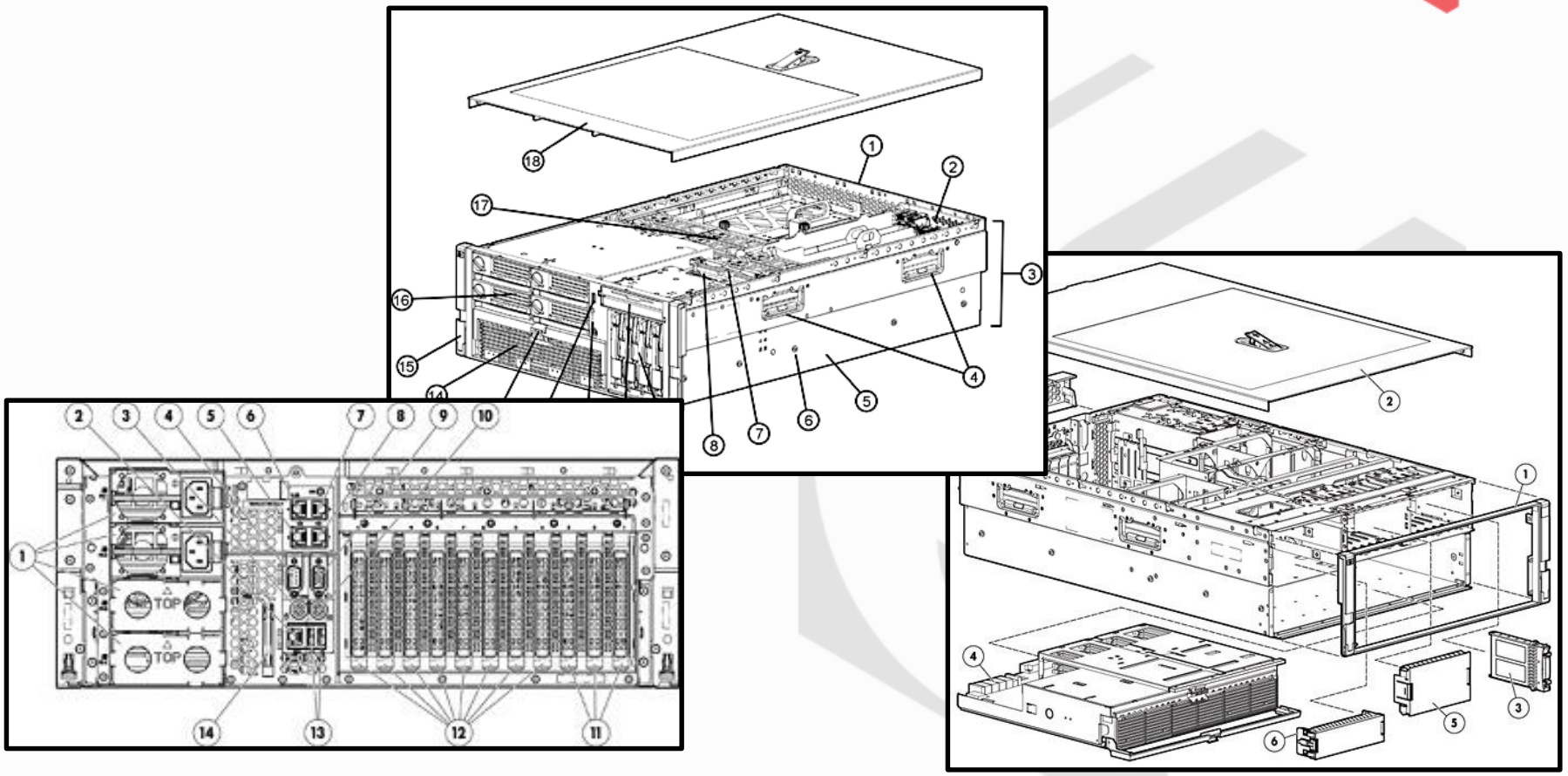
Quanta
Optimize Your Datacenter

power-one®

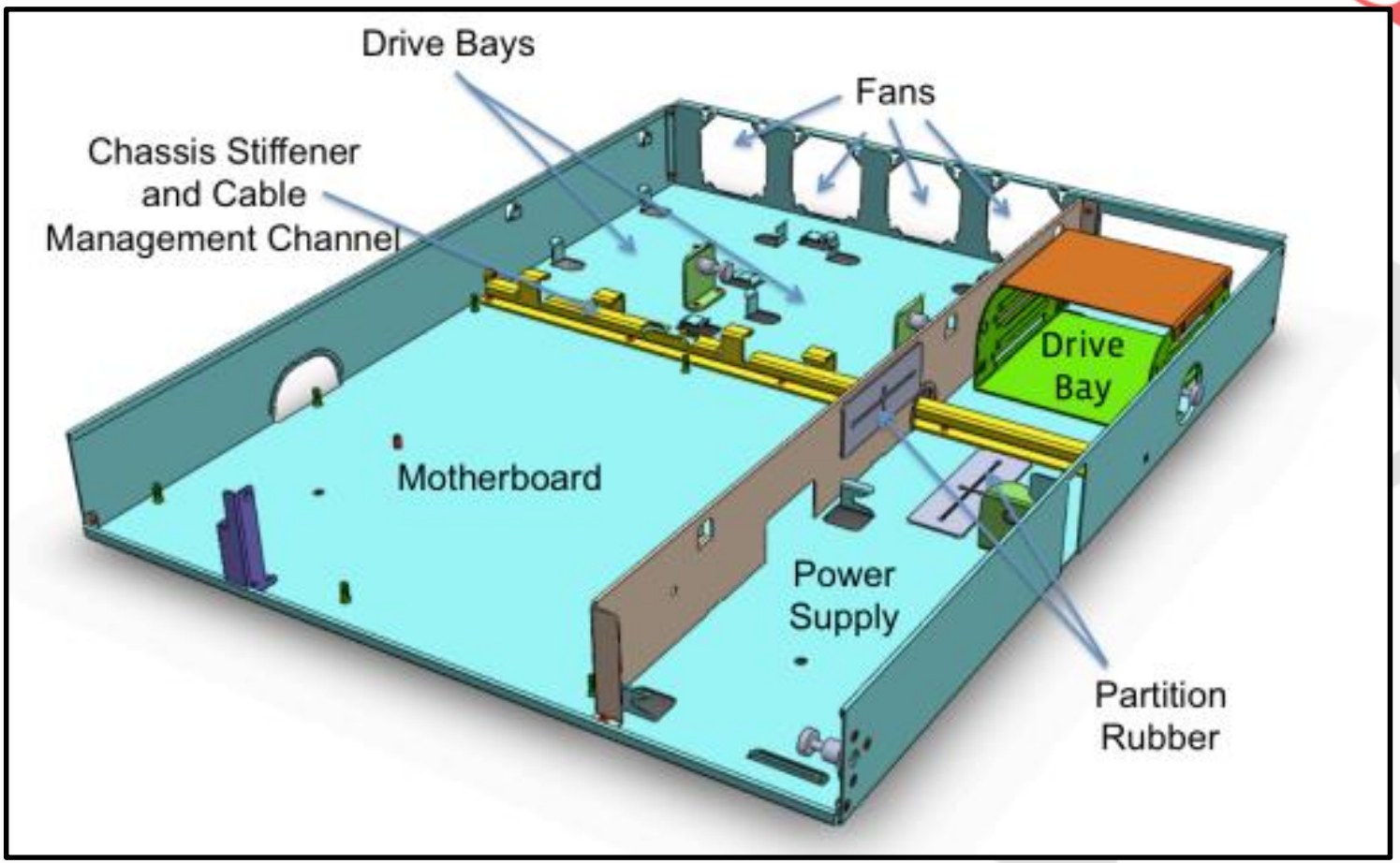
Open Compute: Industry Collaboration

Open Compute: Industry Collaboration

Open Compute: Complex Designs

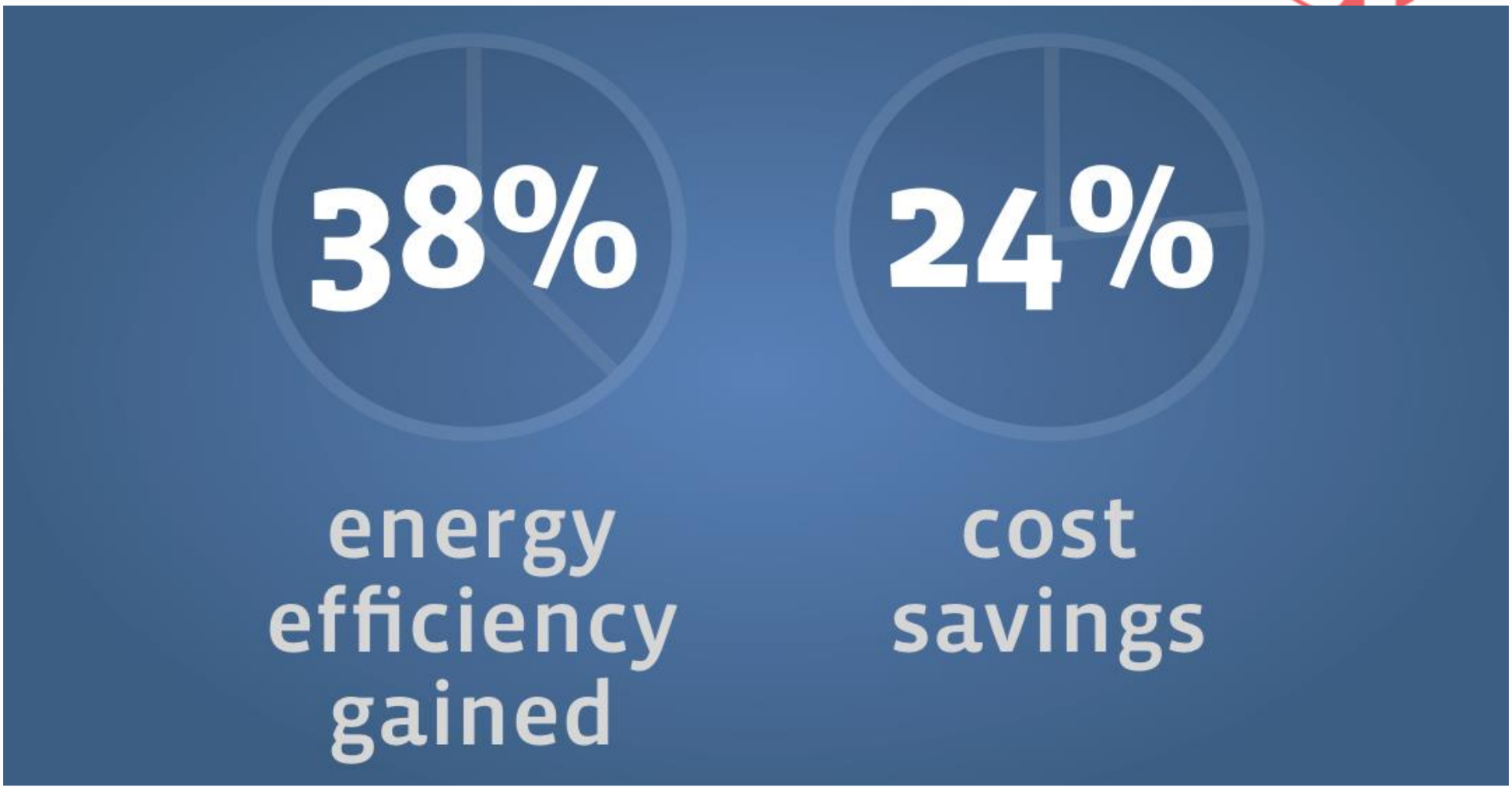


Open Compute: ~~Complex~~ Simple Designs

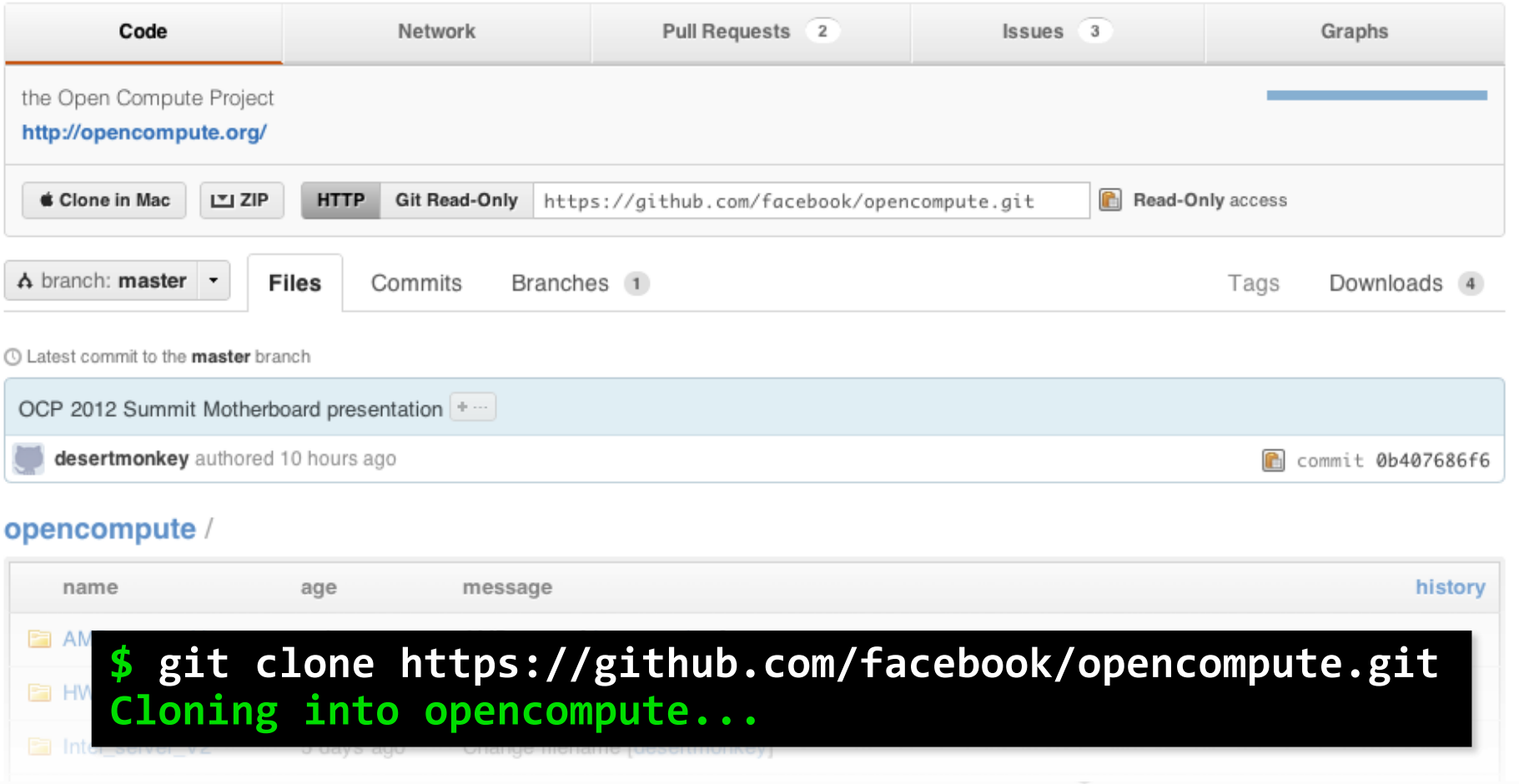


Open Compute: Designed for Scale

Open Compute: Why Facebook loves it



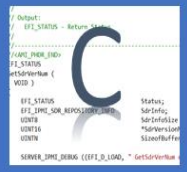
Open Compute: Open Source



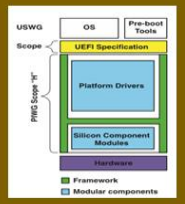
Open Compute and Firmware

- No recommendation for firmware
- Various solutions for boot and network provisioning
- Align on UEFI based technology

Why UEFI?



Mostly written in C.
High code re-use.

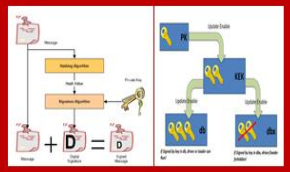


Emphasis on Specifications.
Standards compliance.

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

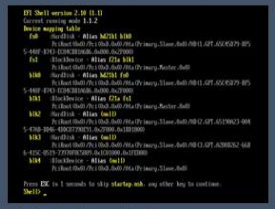
Storage.
GPT removes 2.2 TB MBR restriction.

CPU Architecture independent. Platform design flexibility.



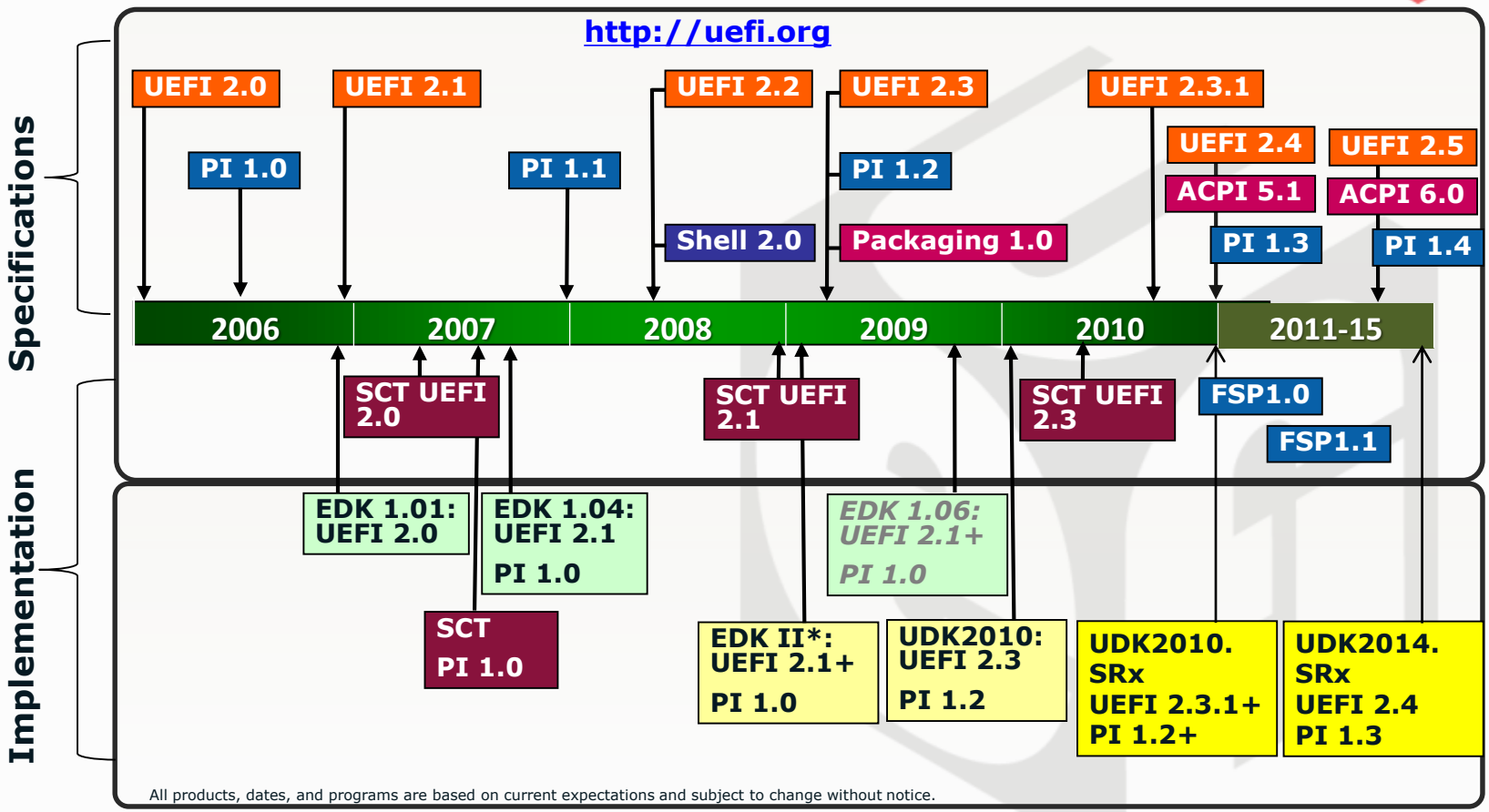
Secure boot solves "trust" related system integration challenges.

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



UEFI shell improves pre-boot testing & diagnostics experience.

Timeline



http://tianocore.org

https://github.com/tianocore/edk2

How to build it

Industry Standards Compliance
- UEFI 2.0, UEFI 2.1, UEFI 2.2, UEFI 2.3, UEFI2.4; PI 1.0, PI 1.1, PI 1.2, PI1.3; 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 EDKI)
- Targeted Module Build Flexibility

*** benefit of EDK II codebase*

Maximize the open source at www.tianocore.org

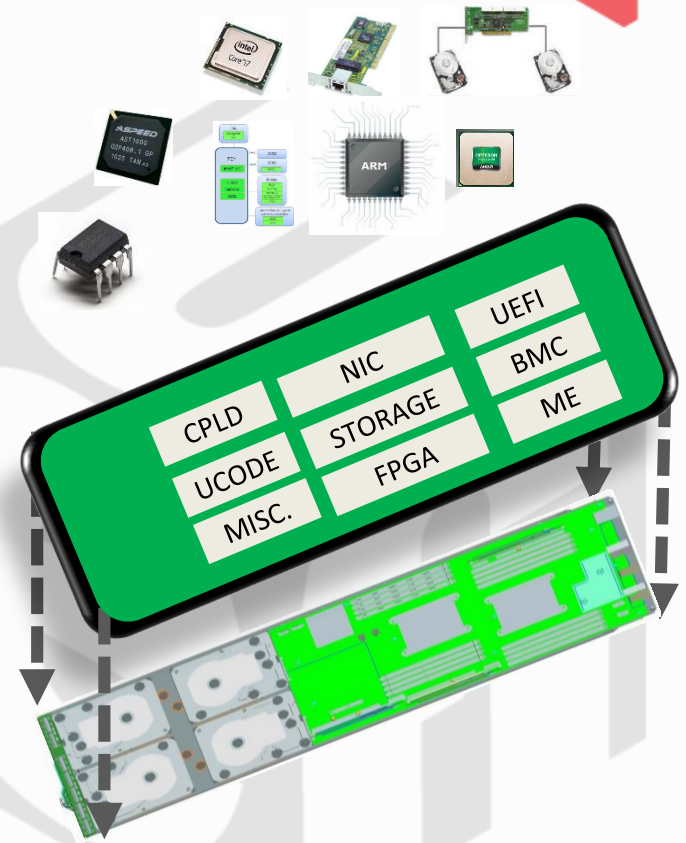
New Specification advances for Cloud

- Error support
  - CPER
- New memory toplogy
  - NVDIMM
  - Reliability
- Boot from HTTP

Firmware Updates

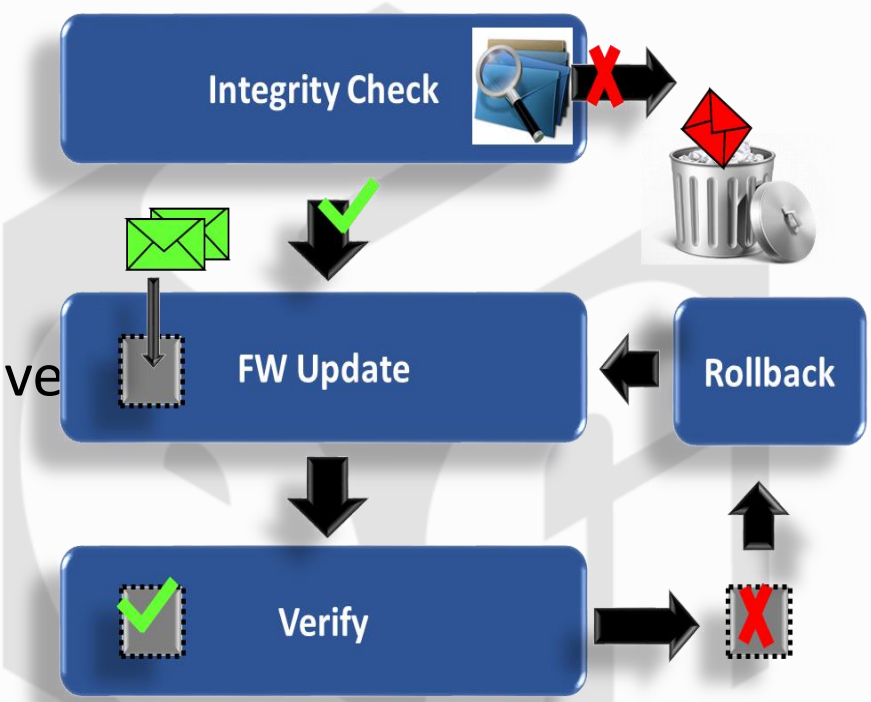
Firmware Update Challenges

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



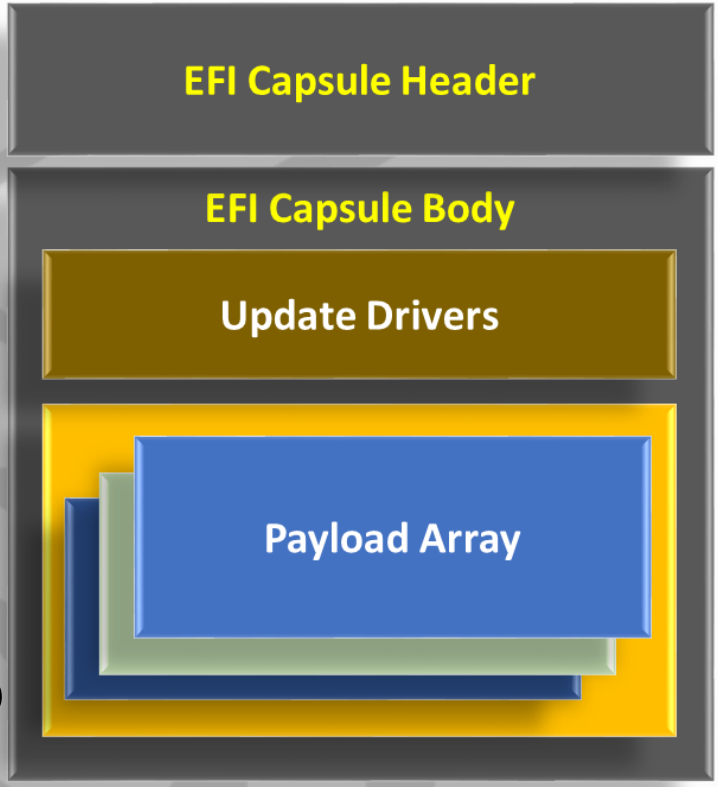
Solving the firmware update Challenge

- Reliable update story
  - Fault tolerant
  - Scalable & repeatable
- How can UEFI Help?
  - Capsule model for binary delive
  - Bus / Device Enumeration
  - Managing updates via
    - EFI System Resource Table
    - Firmware Management Protocol
    - Capsule Signing



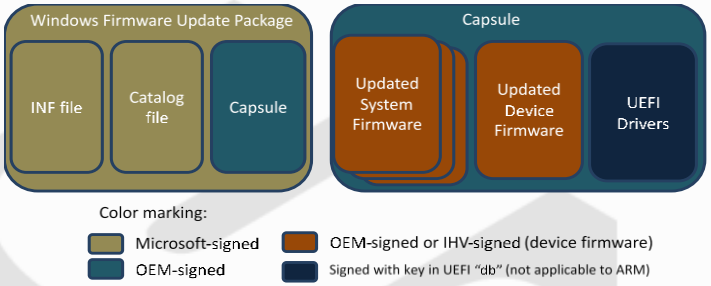
Delivering firmware updates

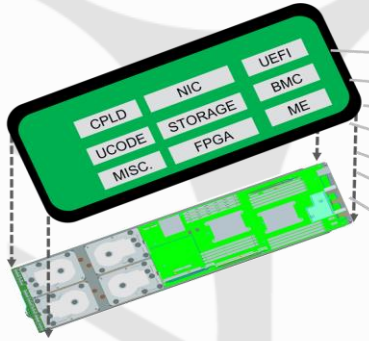
- UEFI supports Capsule format
  - Tools for capsule generation
  - Core logic for capsule handling
- Extensible Capsule format
  - Self-contained
  - Discrete updates
  - Composite updates
- Firmware Management Protoco
  - Reading / updating firmware
  - Integrity checks



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





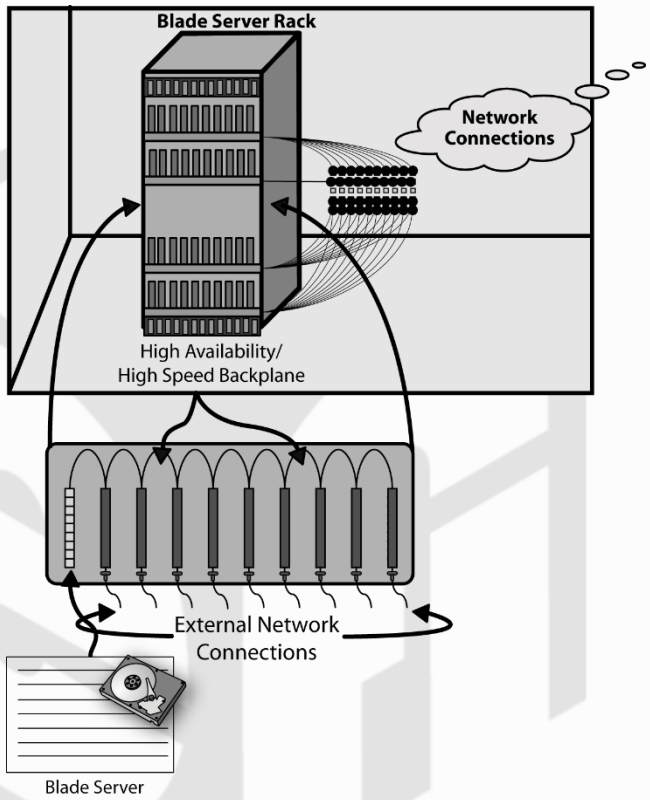
FW	GUID
BIOS	{ $GUID_1$}
BMC	{ $GUID_2$}
ME	{ $GUID_3$}
NIC	{ $GUID_4$}
Storage FW	{ $GUID_5$}
UCODE	{ $GUID_6$}
CPLD	{ $GUID_7$}
Etc..	{ GUID..}

Example ESRT

Bare Metal Provisioning

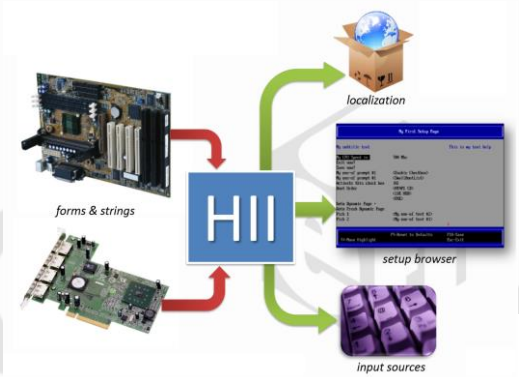
Bare Metal Provisioning Challenges

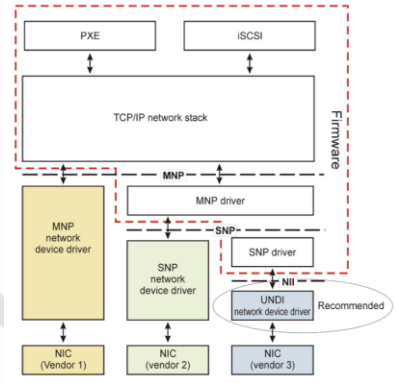
- Hardware Detection
- Installation
  - Local / Remote
- Configuration
  - Local / Remote / Scriptable
- Cloning
  - Automated
- Backup / Recovery
  - Local / Remote / Automated

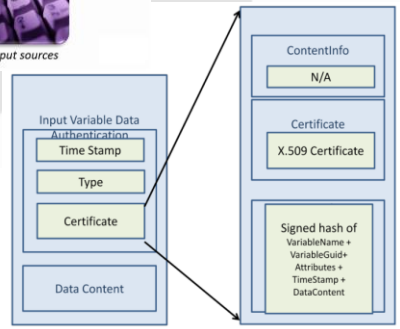


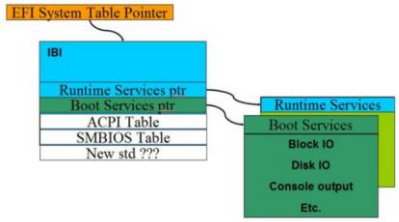
Bare Metal Provisioning Solutions

- 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





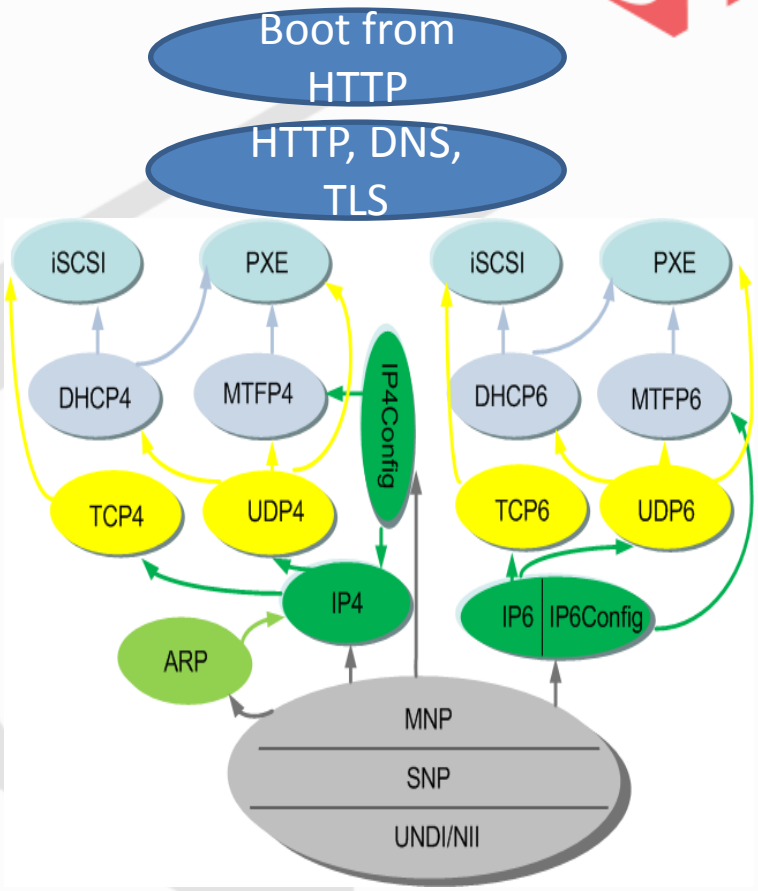




Networking in UEFI

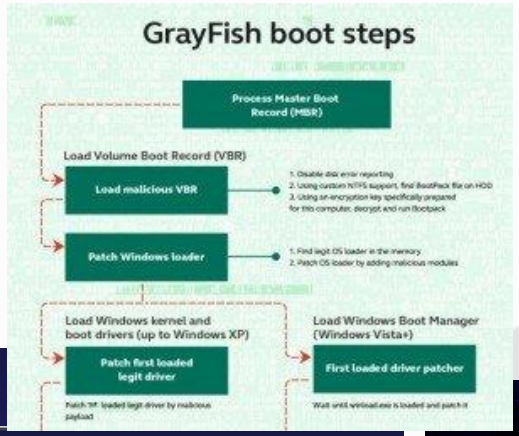
Networking

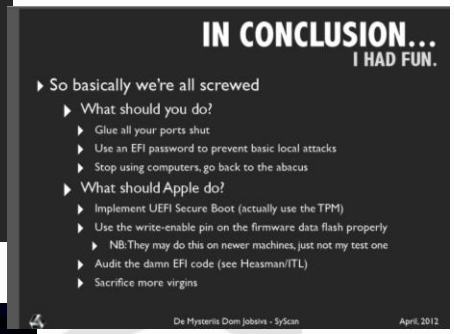
- UEFI offers rich set of Networking Features during pre-boot
  - PXE boot support for network boot, OS installations, provisioning etc.
  - Native support for IPv4 as well as IPv6
  - Network file system support
  - Virtual LAN support, iSCSI
  - IpSec for supporting secure communication
- Evolution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

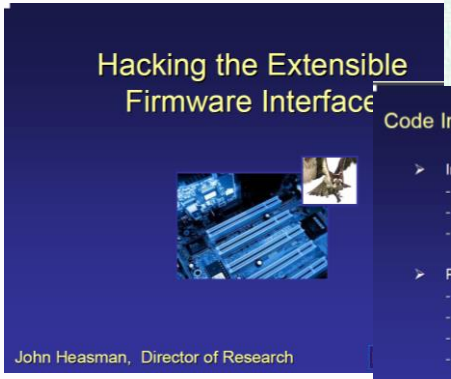


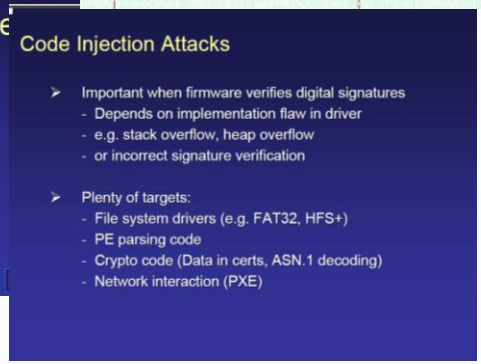
Security

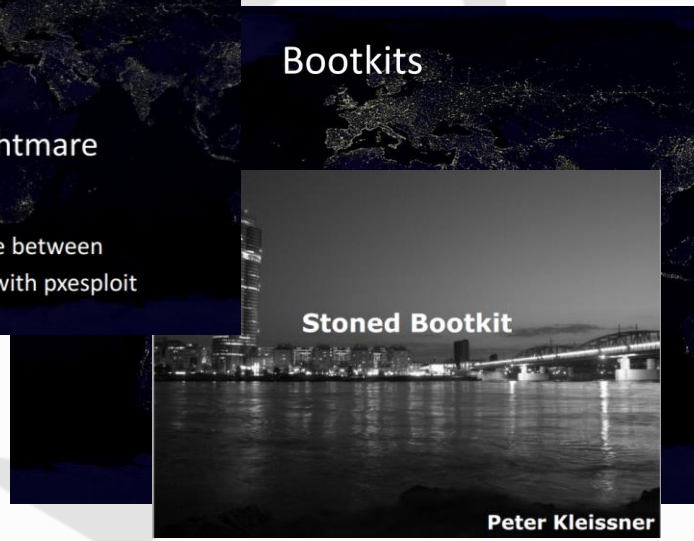
Attacks on Firmware







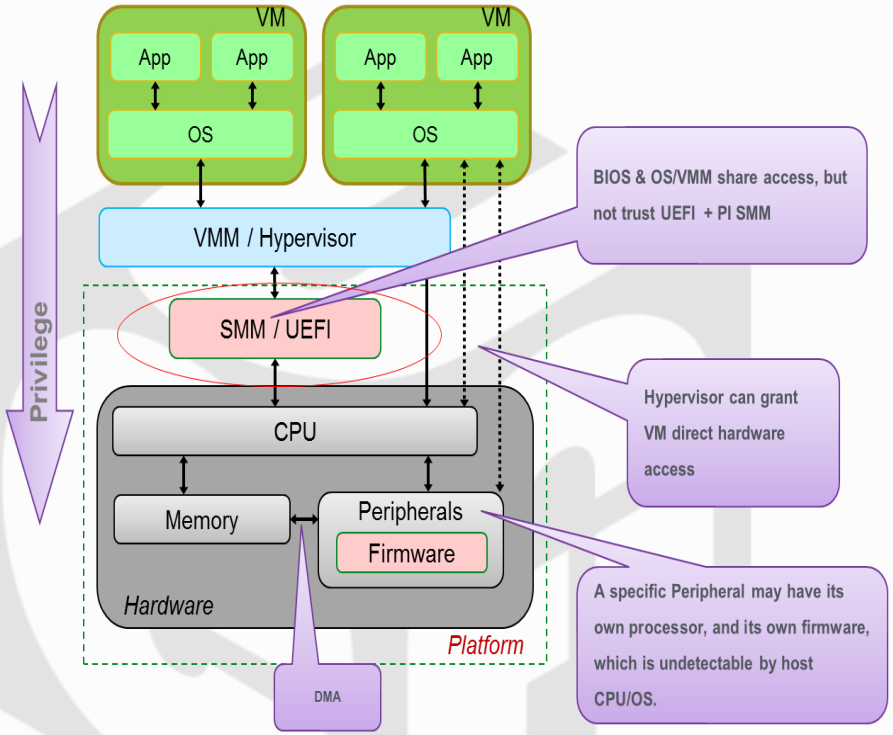




Stephen Cobb, senior security researcher at ESET North America, says that hacking firmware can be particularly effective because it is so hard to eliminate.
It's also particularly challenging to do, says Jean Taggart, security researcher at Malwarebytes. "Doing this on just one brand of hard drive would 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

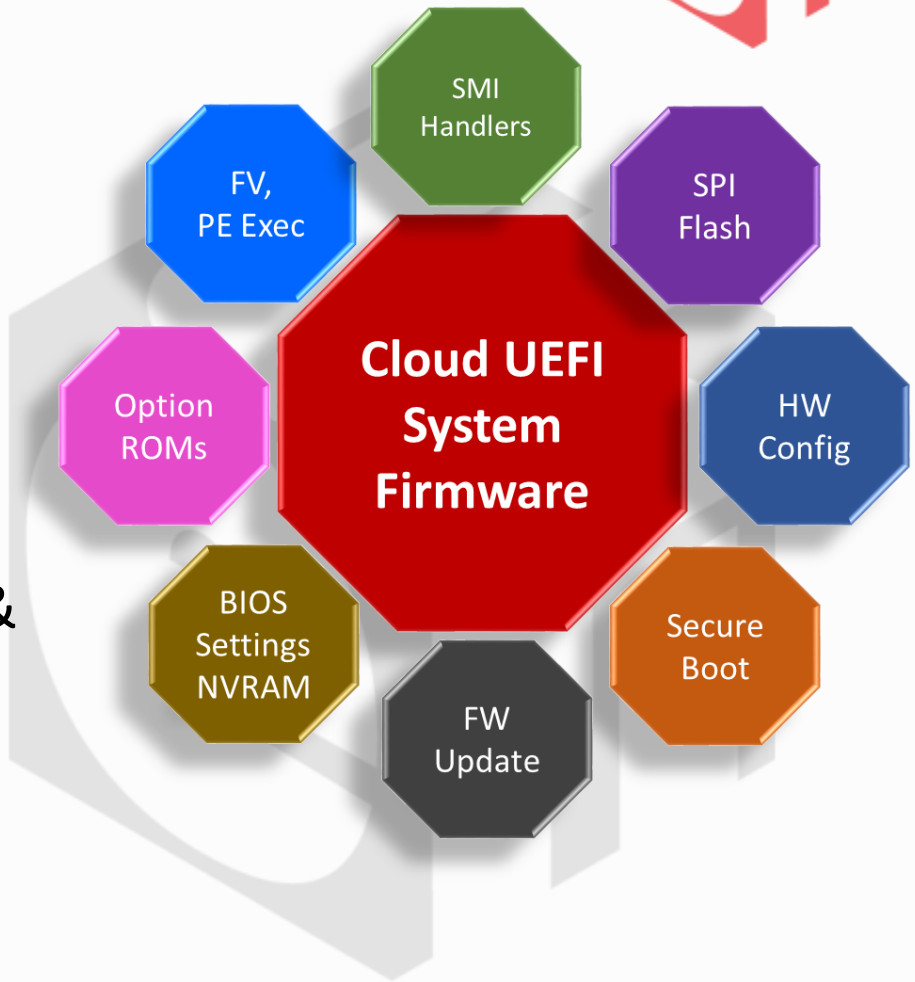
Security

- 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



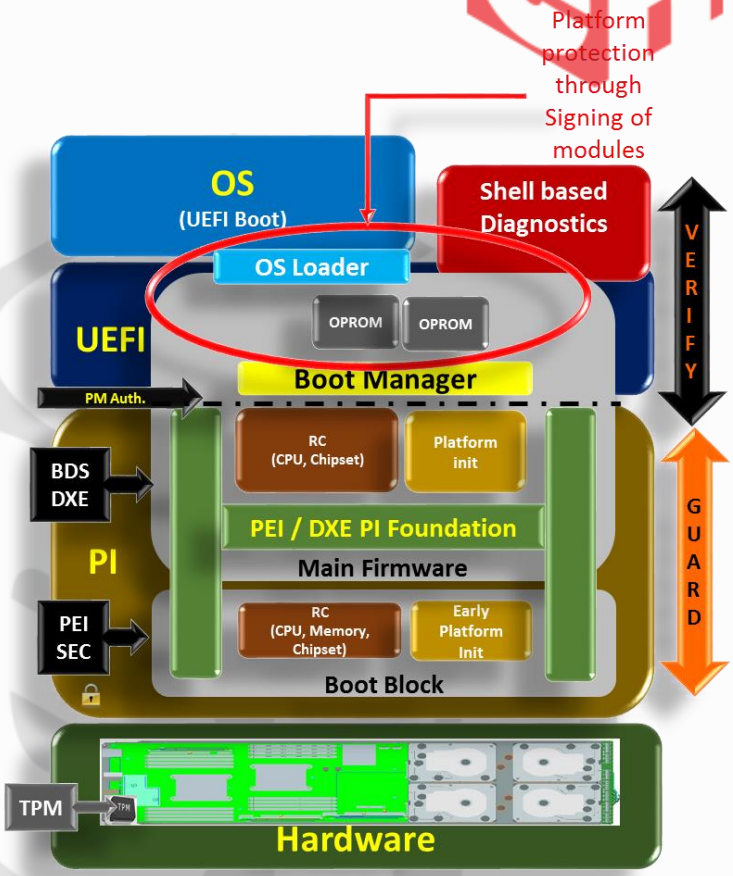
Security Solutions

- 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

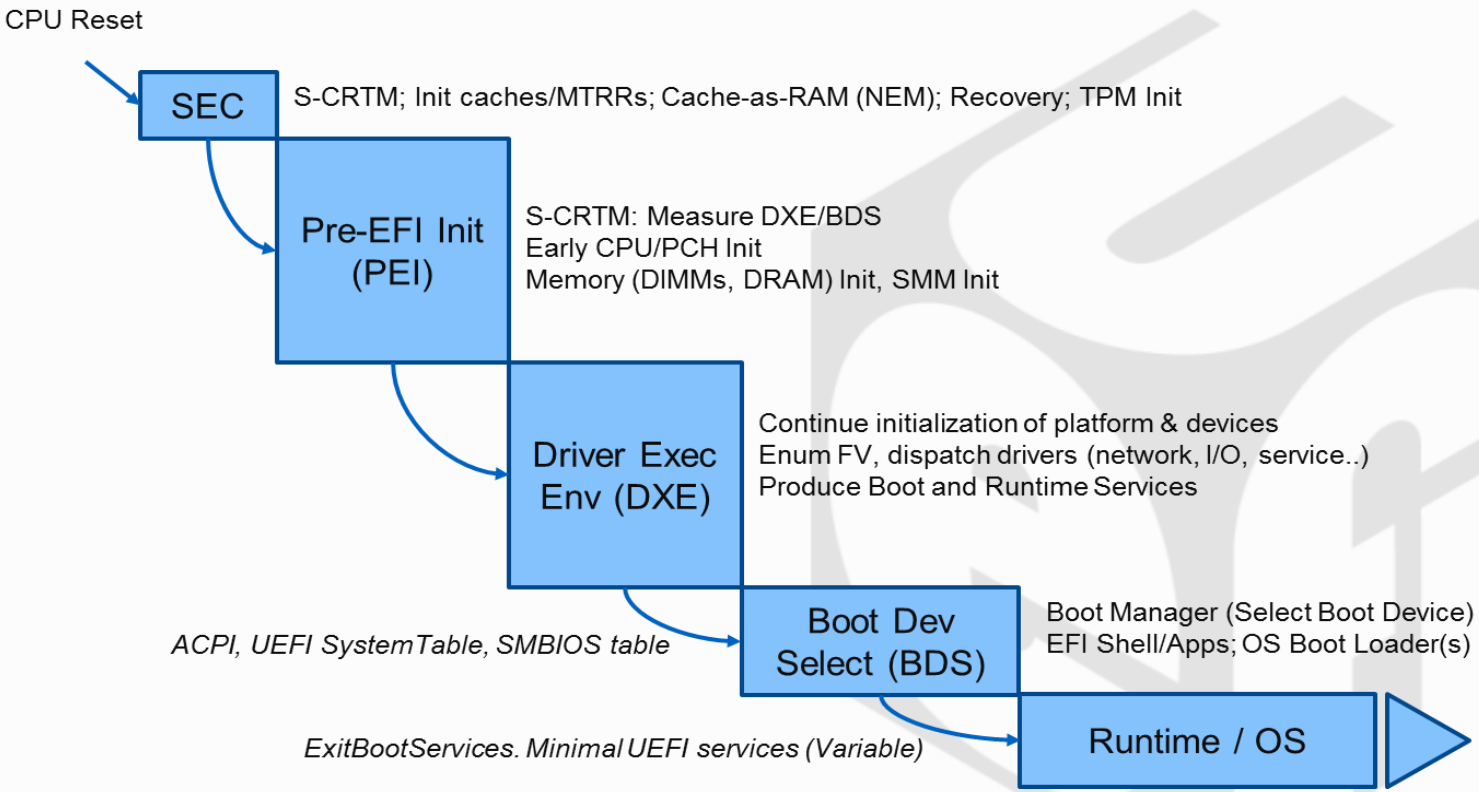


Guarding and Verifying

- 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



Full UEFI Boot Flow



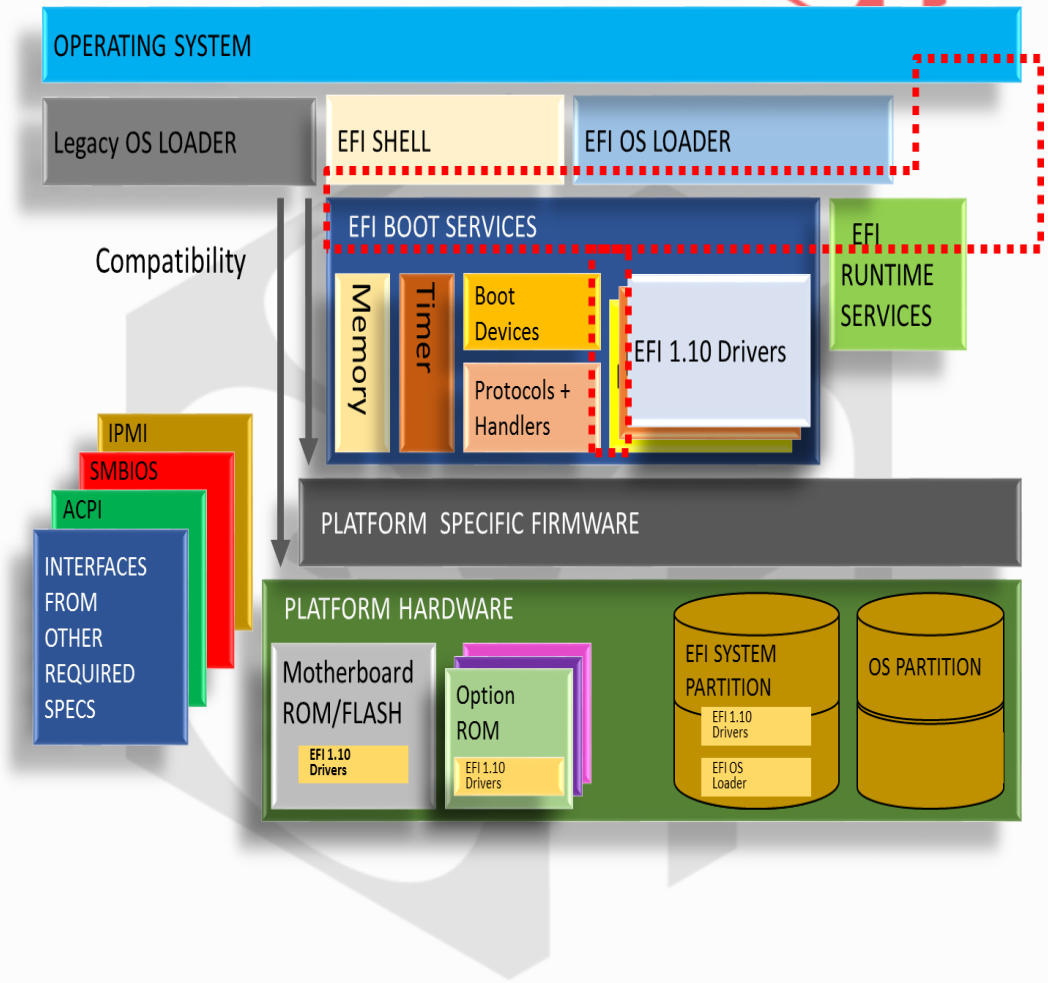
Tools and Diagnostics

Tools and diagnostics 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

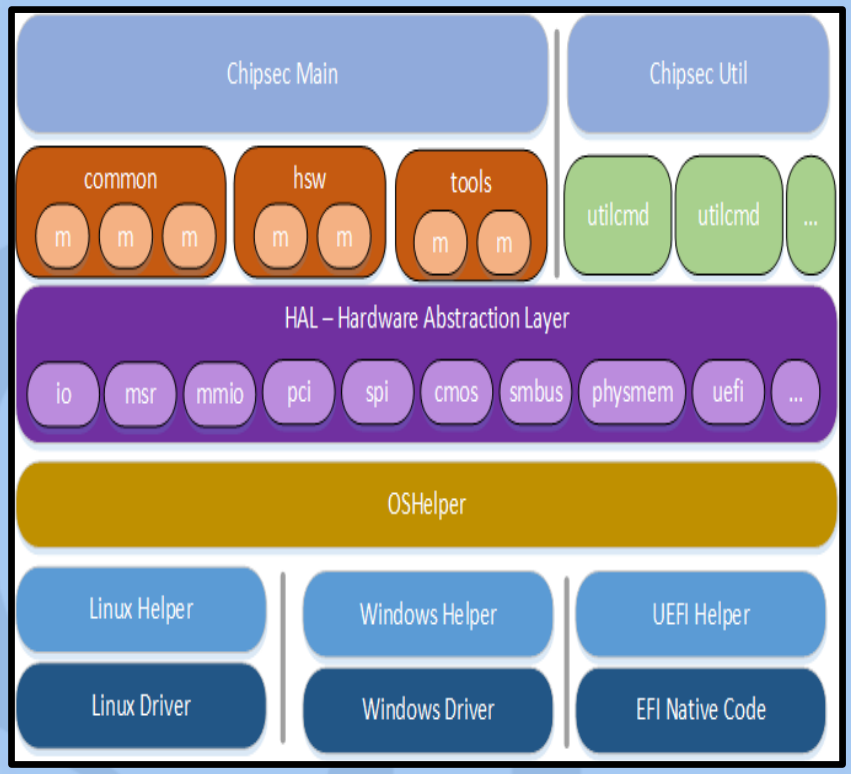
Tools solutions

- 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



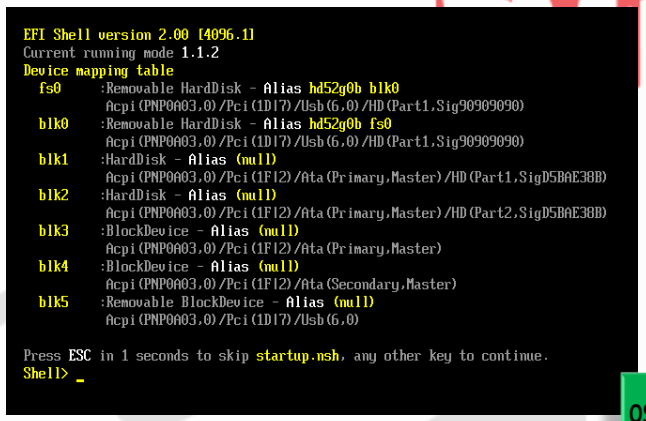
Chipsec tool

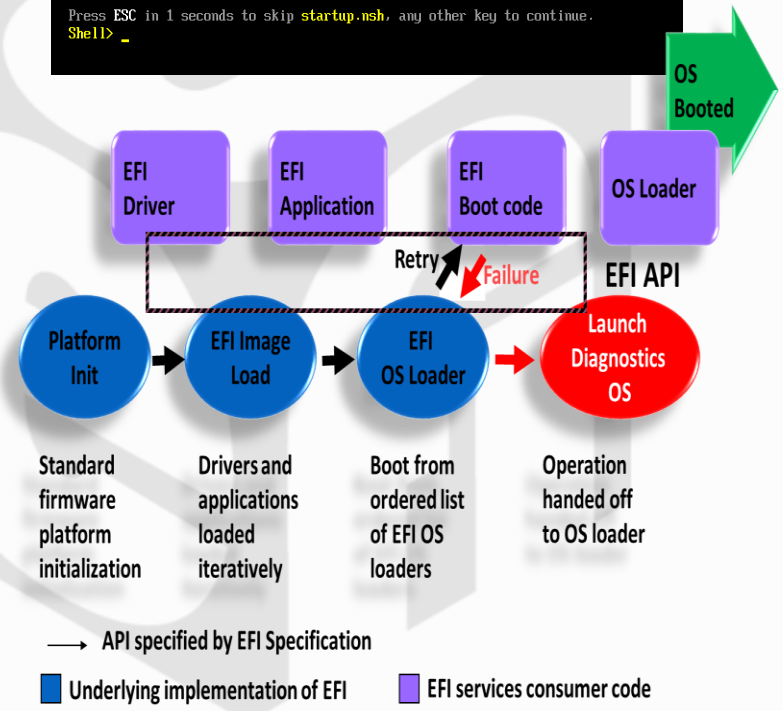
- Platform security assessment framework for risk assessment
- Can be extended to meet specific platform security concerns
- Open sourced
  - https://github.com/chipsec/chipsec
- Supported Environments
  - Windows
  - Linux
  - UEFI (over Python)



Diagnostic 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





Can we do more?

- Yes
- Working group in OCP on updates/management
- Liaison
- Group subteam in UEFI Forum?
- More open source oppty and collaboration

Call To Action

- Get involved in the cloud
- Talk to Mallik and Vincent about how to do more in OCP and the UEFI Forum for Cloud
- The best ideas come from the people who do the work everyday.

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
- chipsec https://github.com/chipsec/chipsec

Thanks for attending the UEFI Spring Plugfest 2015

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

presented by