UEFI Update 5/5/11

Harry Hsiung

Agenda

- UEFI Specification Update
- Intel® UDK2010 Firmware Development Platform
- UEFI Training/Plugfest Schedule
- UEFI Resources
- Opens

UEFI 2.3.1 Specification Update/Key New Features

Security

- Authenticated Variable & Signature Data Base
- Key Management Service (KMS)
- Storage Security Command Protocol for encrypted HDD

Network

IPV6 support

Netboot6 client use DUID-UUID to report platform identifier

Interoperability

- New FC and SAS Device Path
- FAT32 data region alignment
- HII clarification & update
- HII Modal Form

Performance

Fastboot

Non-blocking interface for BLOCK oriented devices

Technology

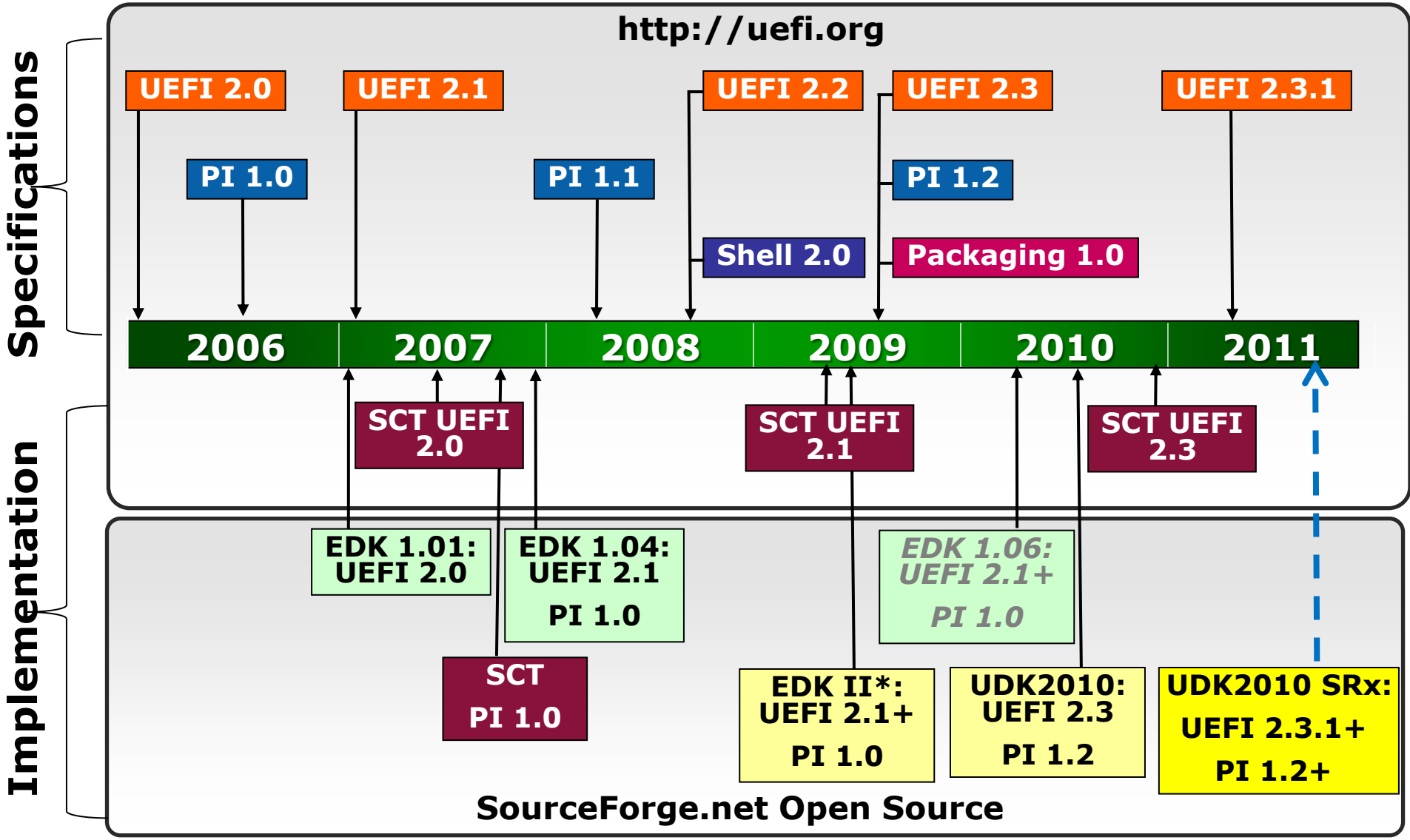
USB 3.0

Maintenance

User Identifier, etc.

The UEFI Board unanimously approved UEFI 2.3.1 final release in mid April 2011

UEFI Specification & Tianocore.org Timeline



EDK II is same code base as UDK2010

UEFI Platforms

- UEFI class 2
  - Contains firmware with legacy bios boot support (CSM)
  - Contains firmware with UEFI boot support
    - New platforms will move from csm priority boot to UEFI priority boot
  - ACPI, Smbios still present
  - Most commercial systems moving to UEFI class 2 at this time
- UEFI class 3
  - Contains firmware with only UEFI boot support (ie UEFI 2.3 and UEFI only GOP, GPT drivers)
  - Will not support boot to legacy bios or any OS that uses legacy hardware interfaces and firmware (int10, int13, int19, MBR etc.)
  - ACPI, SMBIOS still present

Market update

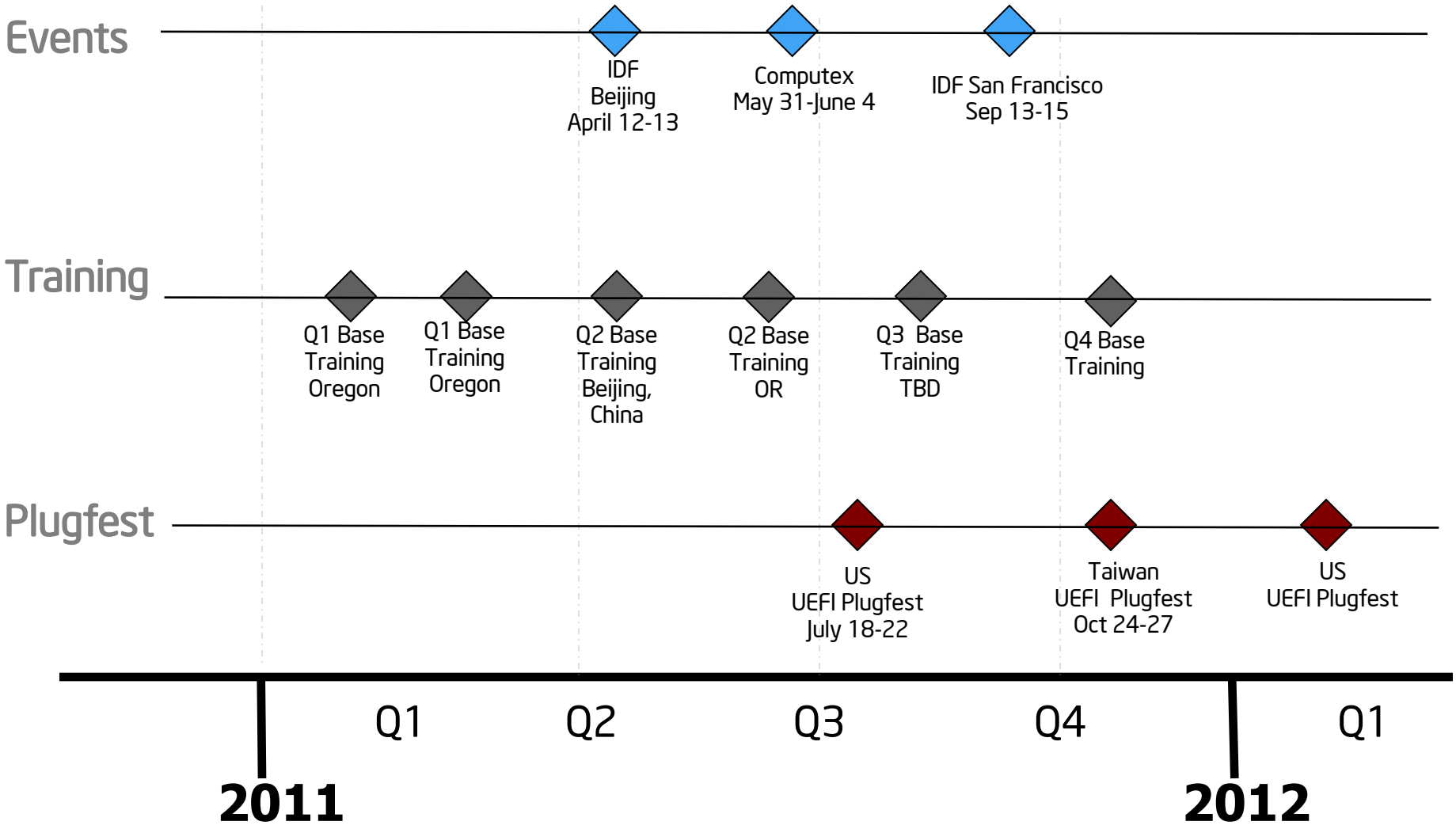
- Most systems shipping systems today are UEFI 2.1
  - Derived from tianocore.org project EDKI (11/17/2007)
  - Some implementations missing GOP driver and/or UEFI pxe boot
- Class 2 systems with legacy or non-UEFI bios mode boot preffered over UEFI mode x64.
- Magic setup switch to UEFI is buried in the bios setup screens somewhere (not std).
- Some will boot to UEFI shell when enabled to UEFI boot mode or select a particular removable device to boot to (ie dvdrom, usb etc.) in UEFI x64 mode and expect the boot file to be \EFI\boot\bootx64.efi on the removable media. Designed to boot to install media to UEFI OS installer.\
- >2.2Tb hard drives causing some users to look at UEFI installation
  - 1st in a series of new hardware starting to cause shift to UEFI.

Intel® UDK2010 firmware development platform

- Enables developers to write, debug, and validate drivers and applications on UEFI 2.3+
- Benefits
  - All H/W commercially available, NDA not required
  - Build platform yourself or purchase an pre-assembled platform
  - UDK2010 Compatible, supports UEFI 2.3+
  - Long lifetime hardware platform support from Intel
- It's easy to do: Purchase Parts from supported H/W list, assemble, download UEFI 2.3. BIOS Image, and flash BIOS to motherboard using a SPI Flash programmer
- Pre-assembled systems available at HDNW, visit http://www.Tunnelmountain.net
- http://shop.hdnw.com/default.aspx or contact tomk@hdnw.com, (425) 943-5515 xt 4223. Use product name "Tunnel Mountain" when ordering

Visit www.intel.com/technology/efi for the latest

UEFI Enabling Schedule

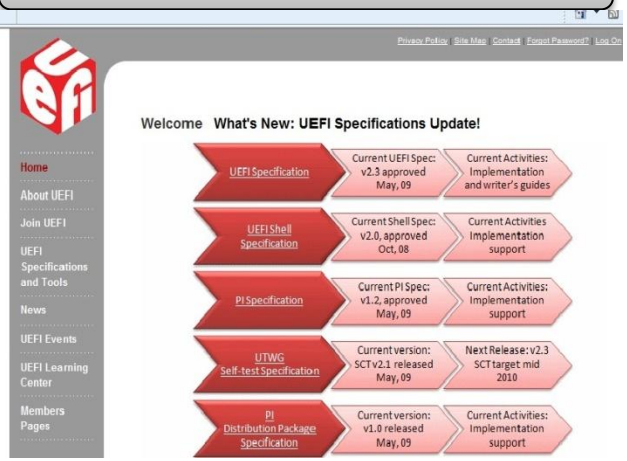


Opens

- Which UEFI features Ubuntu is interested to support
- Business model and deployment
  - how do you support customers - support line, contract
  - Other distros/spins for specific customers
- Ubuntu events that oems/odms can participate in implementing on a UEFI platform?
- Any plan for Ubuntu UEFI 32 bit support (for Atom and smaller platforms)
- Will Ubuntu distros support Secure boot?
  - Who is going to sign Linux distro boot loader and Kernel for secure signed boot?

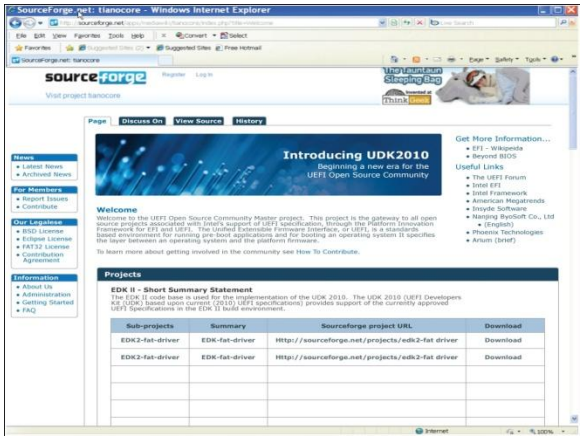
UEFI Industry Resources

UEFI Forum



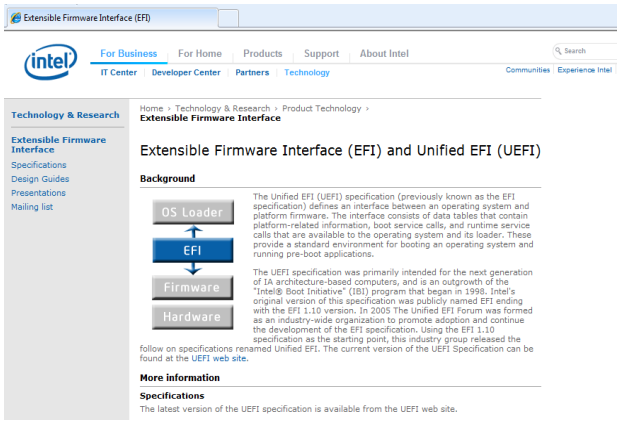
www.uefi.org

UEFI Open Source



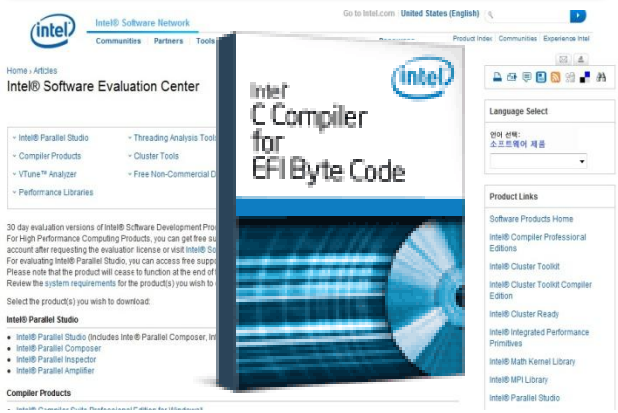
www.tianocore.org

Intel UEFI Resources



http://developer.intel.com/technology/efi

Intel EBC Compiler



http://software.intel.com/en-us/articles/intel-software-evaluation-center/#compilers

UEFI Books

www.intel.com/intelpress

Training/OSVs/IHVs Contact

Laurie Jarlstrom
- Intel UEFI Training
- Laurie.Jarlstrom@intel.com

Harry Hsiung
- Intel OSVs UEFI Support
- harry.L.Hsiung@intel.com

Bailey Cross
- Intel IHVs UEFI Support
- Bailey.T.Cross@intel.com

10

Backup Slides

Your UEFI requirements of IHVs?

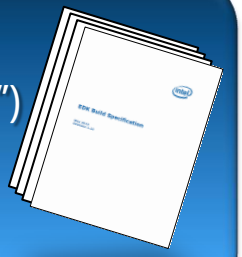
- Migrating legacy Option ROMs into UEFI Option ROMs
- Providing manufacturing tools and device drivers with UEFI support
- UEFI setup and self diagnostics for UEFI option rom drivers(configuration and self test in preboot space)
- Large disk support with UEFI
- UEFI 2.3.1 Secure Boot (UEFI driver in option rom must be signed with an OEM or UEFI CA or it will not be allowed to execute in boot path).
- Encrypted hard disk drives
- Optimized boot
- Who is going to sign Linux distro boot loader and Kernel for secure signed boot?
- Etc

Intel UEFI Convergence Direction Summary

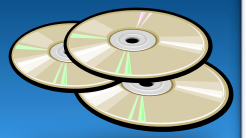
- Intel will unify internal teams on EDK II / Intel® UDK 2010
  - Intel will remain aligned on UDK (UEFI Development Kit) as it evolves

- Intel Client Group, Server Group and SSG will align in 2011/2012
  - Other Intel BIOS alignment activities will occur on varying timelines

- Intel will publish the minimum baseline source compatibility definition
  - Intel to deliver silicon support assuming this compatibility definition (the "Green H")
  - Intel will set a reasonable pace for change and strongly encourage customers to align



- Intel will deliver shared code based on Intel® UDK 2010
  - Contact business unit on specifics for shared code
  - Intel® UDK2010 code base available for evaluation purposes on Tianocore.org



Basis for Transition Plan

- Evolution of our current strategy to align Intel's team on a common code base
  - Intel Product Groups will use a common BIOS codebase to reduce development and validation costs
  - Improve code sharing, efficiency and enabling
- Strong industry desire to accelerate adoption of new features and technologies
  - Intel® UDK2010 offers significant compelling Security, Networking, and Interface improvements
- Maintain Industry Software Stability
  - Industry specification is stable
  - Reduce the number of codebase transitions for Intel, IBVs and OEMs

Why UDK 2010?

- Our Fastest & Lowest Risk Path to EDKII Feature Set
  - Strong internal & industry desire to accelerate adoption of current UEFI specifications
  - Prior common core presented challenges resolved by UDK 2010
  - UDK 2010 is a stable and modular code base which is well suited for cross segment usage

Intel IDF 2011 Beijing UEFI track slides

http://intel.wingateweb.com/bj11/scheduler/public.jsp

enter UEFI for search box

Session ID	Title
EFIS001	Microsoft* Windows* Platform Evolution and UEFI
EFIS002	UEFI Development and Innovations for System-On-Chip (SoC)
EFIS003	UEFI and Transparent Computing Technology
EFIS004	Intel® UEFI Development Kit 2010 and Intel® Boot Loader Development Kit: Foundations for Advanced Embedded Development
SPCQ001	Hot Topic Q&A: Intel® Boot Loader Development Kit (Intel® BLDK)
EFIS005	Security and Networking Advancements Today's UEFI and Intel® UEFI Development Kit 2010 (Intel® UDK2010)

UEFI 2.1 Features

- Added protocols
  - HII (several protocols)
  - Absolute pointer protocol
- New member functions or equivalent
  - Driver Supported Version (for option roms)
  - Extended Simple Text In (more function keys supported)
  - Authenticated Variables
  - Extended SCSI Pass through
  - Signal on configuration change
  - EHCI exclusive ownership
  - Firmware storage device path
  - Hot key registration support
  - Run-time services with interrupts enabled
- Clean-up: e.g. error returns, * vs ** in declarations in several protocols introduced as a result of implementation

Required	Optional

UEFI 2.2 Significant Features

- Networking – IPv6
  - IPv6 stack corresponding to existing IPv4 stack
  - Replacement for PXE protocols which are IPv6 compliant and large network friendly
    - Now being worked through IETF
  - Support for more LAN protocols: EAP and VLAN
- Security – Driver signing
  - Added optional ability to create firmware / OS trust relationships
    - Via key exchange
  - More signature combinations
  - Good / Bad list support
  - Platform owner control of denial response
  - Pre-Boot Authentication (PBA) Framework
    - Passwords, Smart cards, Fingerprint sensors, etc.

Required	Optional

UEFI 2.2 Other Features

- HII
  - Additional operators for mapping to other standards
  - Page by page security control
  - Animation updates
- EFI_ATA_PASS_THRU Protocol
  - Gives direct access to ATA devices
- UEFI Driver Health
  - Allow for a driver to fix/re-configure (e.g. rebuild RAID set)
- ABI Updates/Clarifications
  - Floating Point/MMX/XMM
  - 16-Byte stack alignment
- EFI_LOAD_FILE2 Protocol
  - Loads non-boot-option EXEs (PCI option ROMs & apps)
  - Modifies LoadImage() behavior
- EFI_LOADED_IMAGE Protocol
  - Associates entire device path with EXE image
- Bug fixes in spec for rest of document

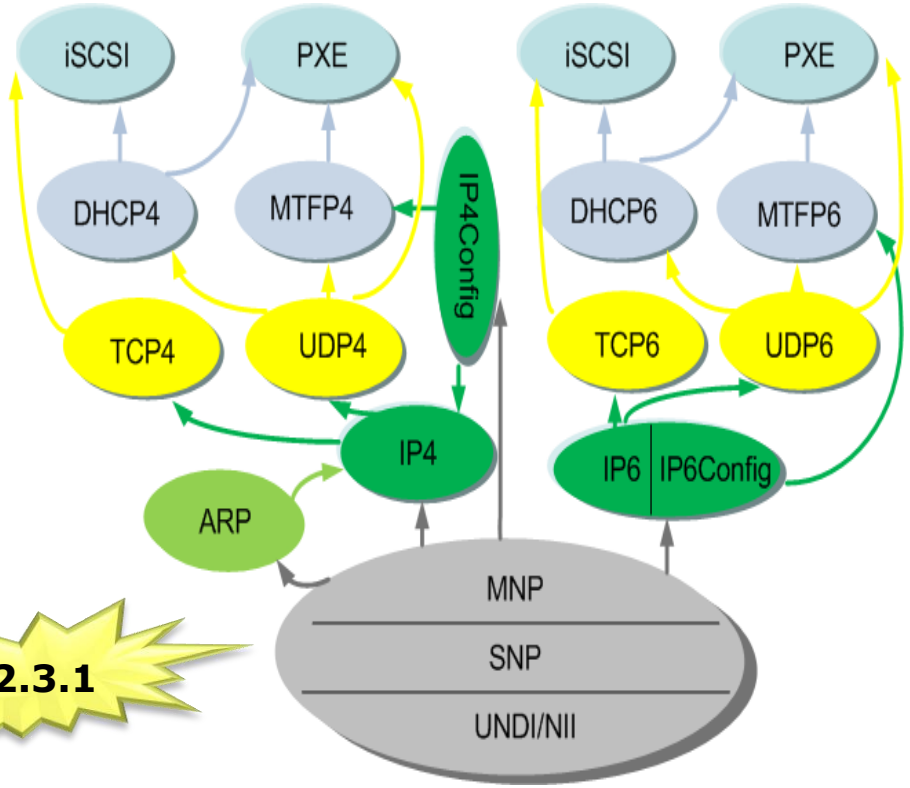
Required	Optional

IP6 Networking

- **IPv6 protocols compliance**
  - **IPv6 ready logo approved**

    http://www.ipv6ready.org/db/index.php/public/
  - **Requirements for IPv6 transition http://www.antd.nist.gov/usgv6/usgv6-v1.pdf**

- **Technology includes**
  - **IP4/6, UDP4/6, TCP4/6, DHCP4/6, MTFP4/6, iSCSI, PXE**
    - **Allows for concurrent network applications via design based upon MNP**

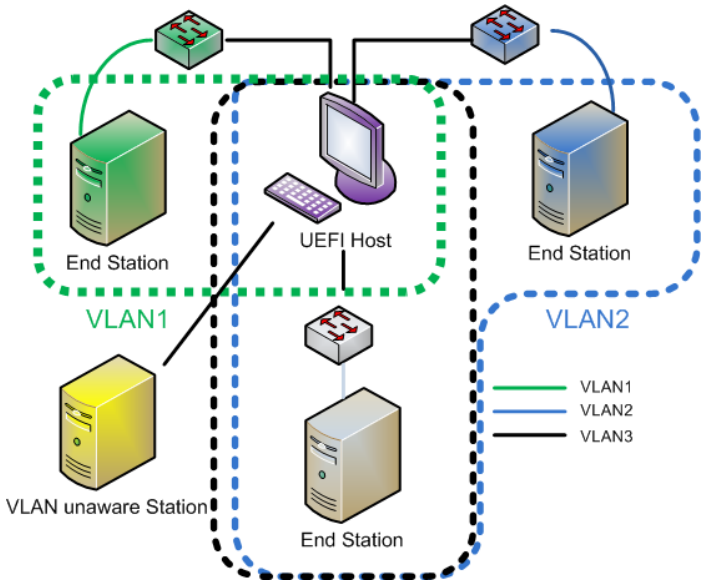


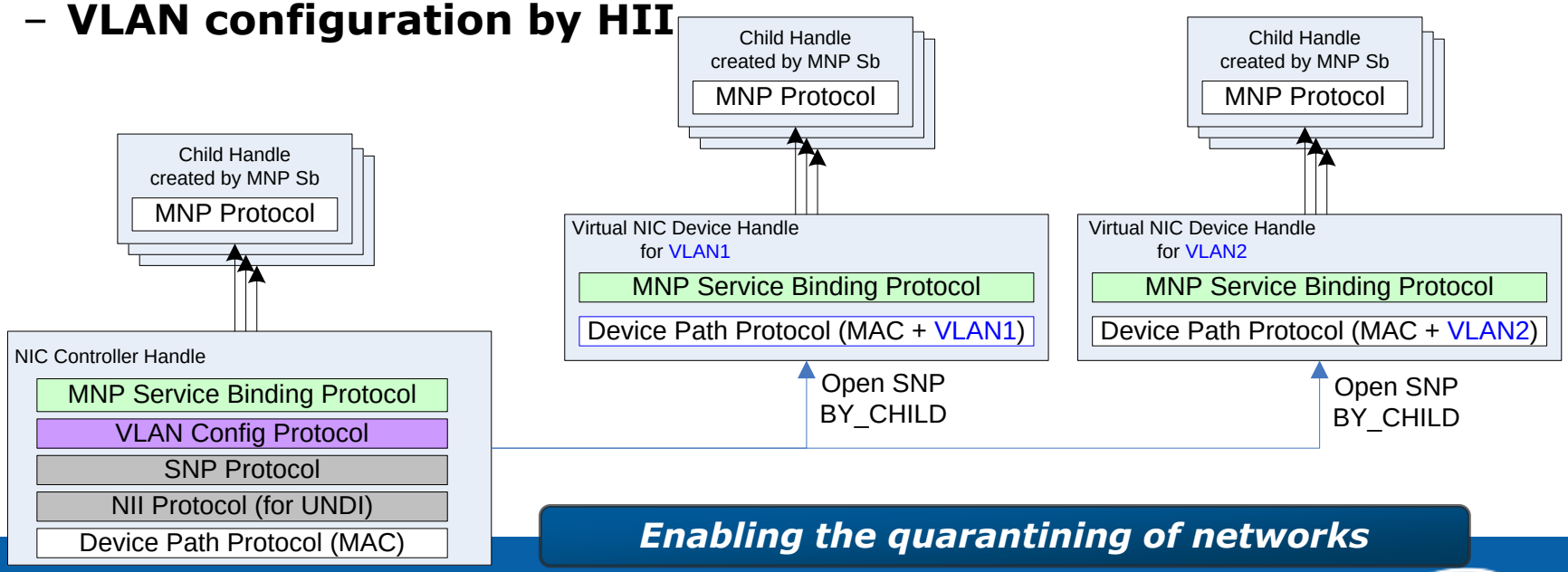
2.3.1

Industry moving to IPv6 for equipment procurement

VLAN Support

- **Virtual Local Area Network**
- **– Defined in IEEE 802.1Q, to create logical groups of stations**
- **– Increased performance, security and improved manageability**
  - **– Support Hybrid LAN topology**
  - **– Multiple VLAN for one station**
    - **– VLAN configuration by HII**

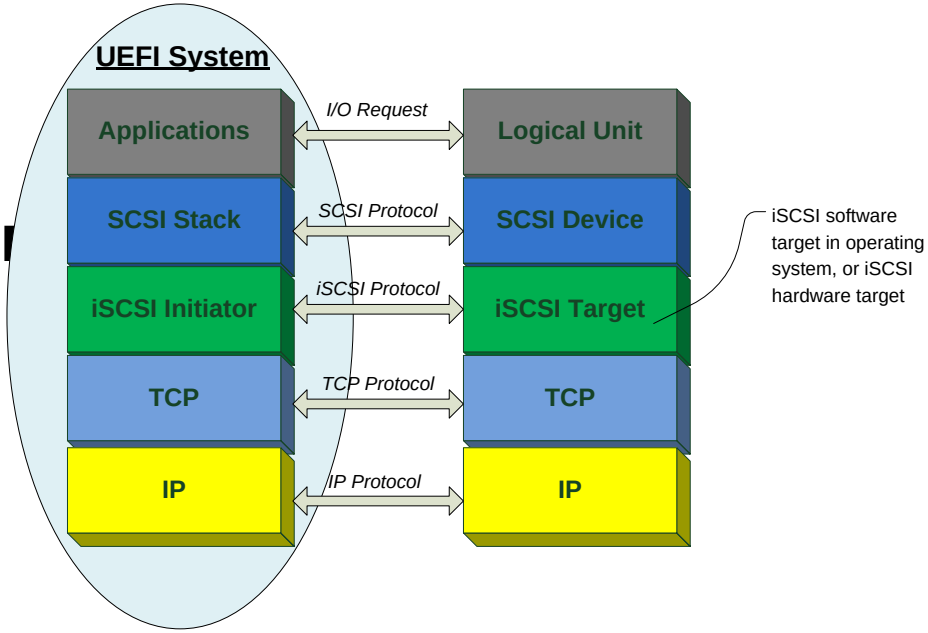


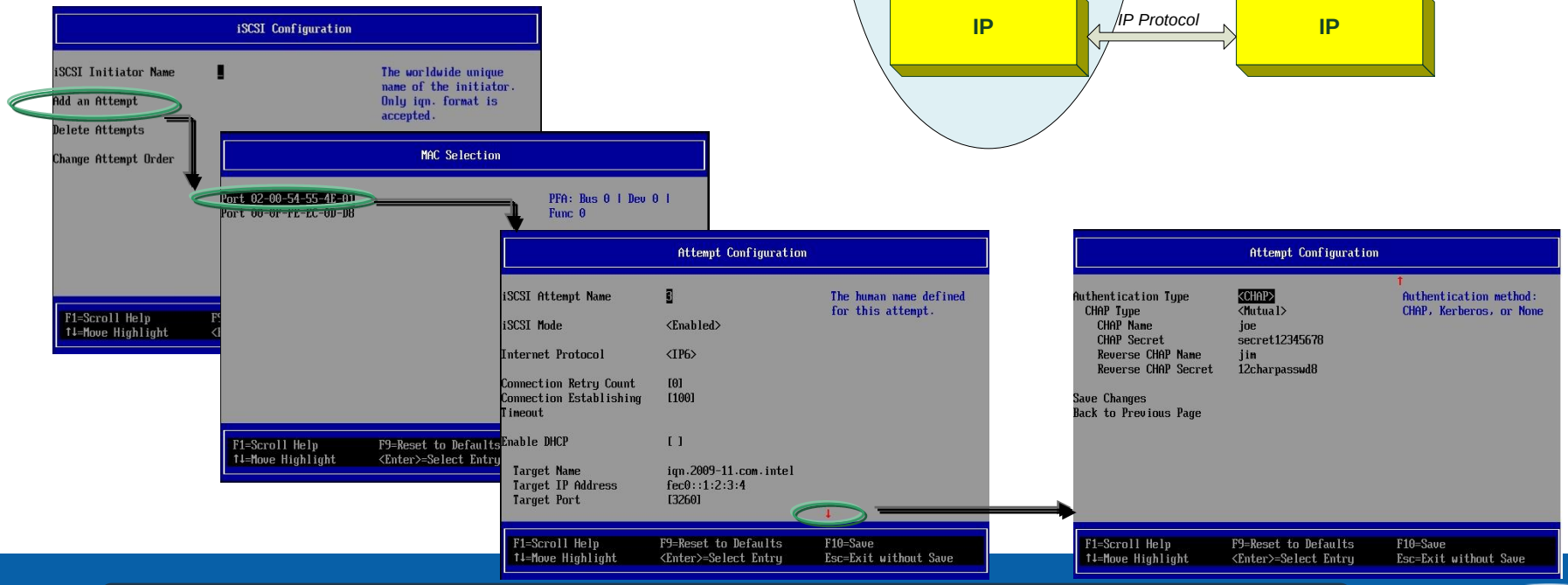


Enabling the quarantining of networks

UEFI iSCSI Solutions

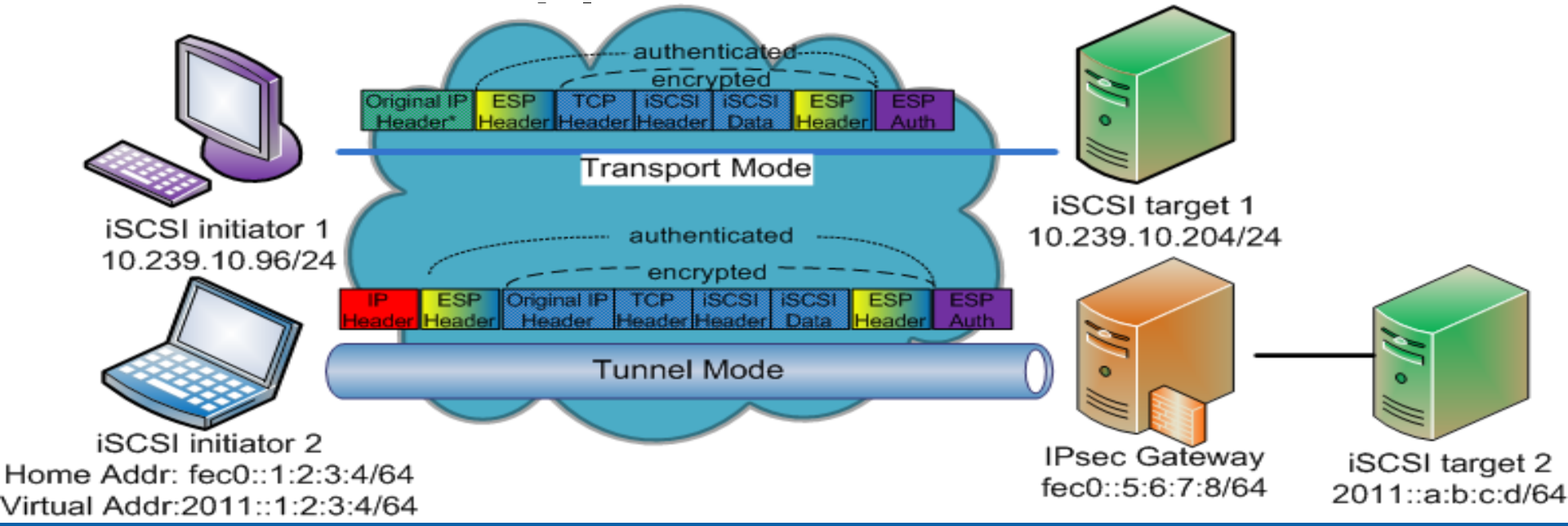
- **SAN/Data center boot over iSCSI**
  - **Manual/DHCP based configuration allowed**
  - **Cryptographic logon with CHAP**
  - **Multi-path/fail-over capable**
  - **User Interface using HII**





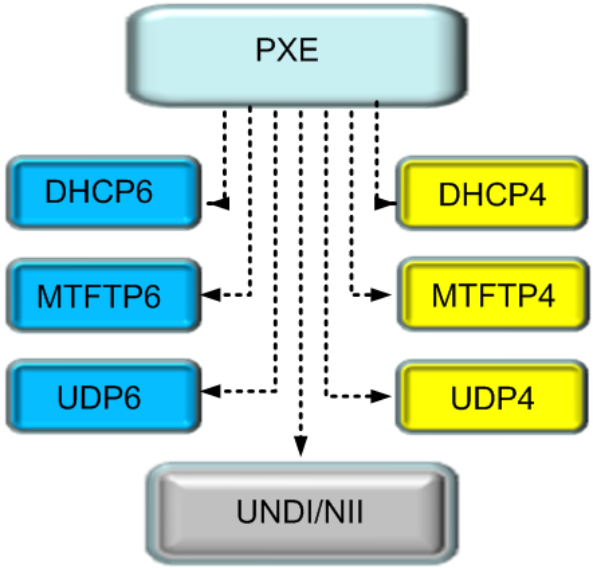
IPsec - Network Security

- **Secure Internet Protocol Communication**
  - **Protects any application traffic across an IP network**
  - **Mandatory for IPv6**
- **Features include**
  - **AH, ESP, IKE version 2**



UEFI PXE Solutions

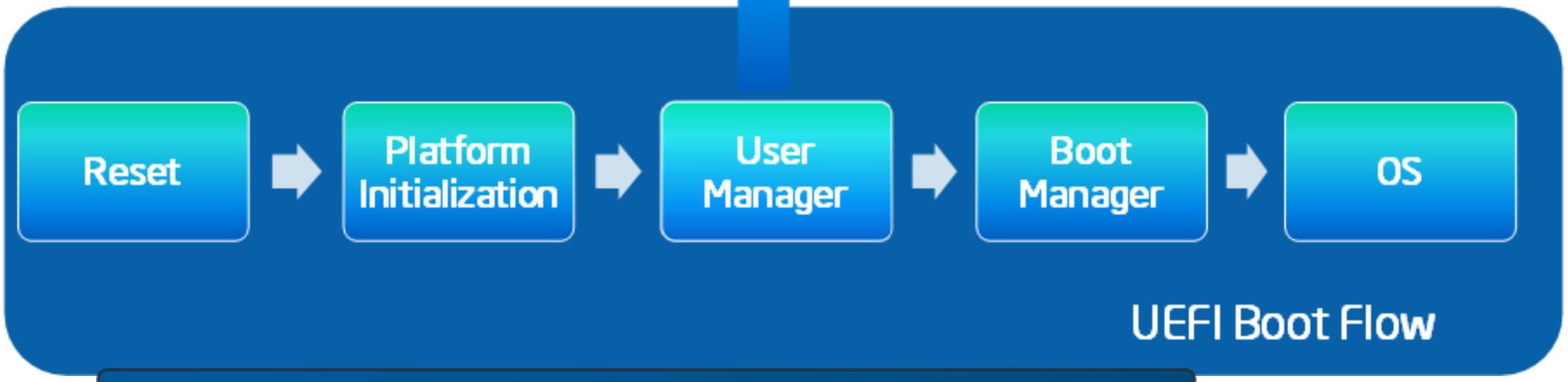
- **Preboot eXecution Environment**
- **– General network booting**
  - **Independent of data storage device**
  - **– IPv4 based PXE defined in PXE 2.1**
  - **– IPv6 based PXE defined in UEFI 2.3**



- **Technology includes** **2.3.1**
  - **– Dual network stack support**
    - **Evolution of network boot to IPv6 defined in IETF RFC 5970**
    - **IPV6 pxe boot called netboot6**
    - **Support of multiple preboot applications using MNP layer (ie ftp, telnet, ssh etc.)**
  - **– DUID-UUID support**
    - **Use SMBIOS system GUID as UUID**

UEFI User Identification

- Pre-boot Authentication
  - Facilitates appropriate user and platform administrator existence
  - A standard framework for user-authentication devices
    - Static password, Network auth protocols, Smart cards, USB key & fingerprint sensors



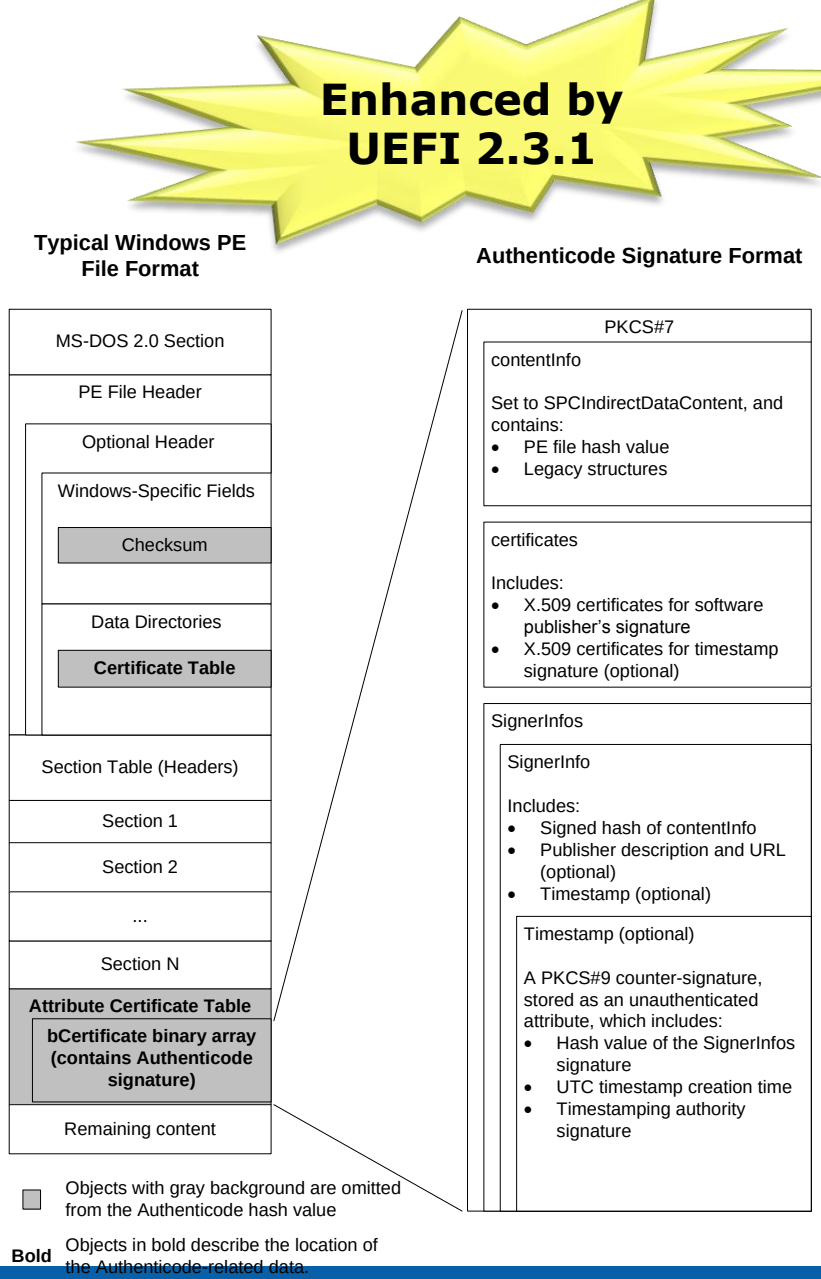
Support for various pre-boot authenticators

Extensible integrity architecture

25

UEFI Driver Signing

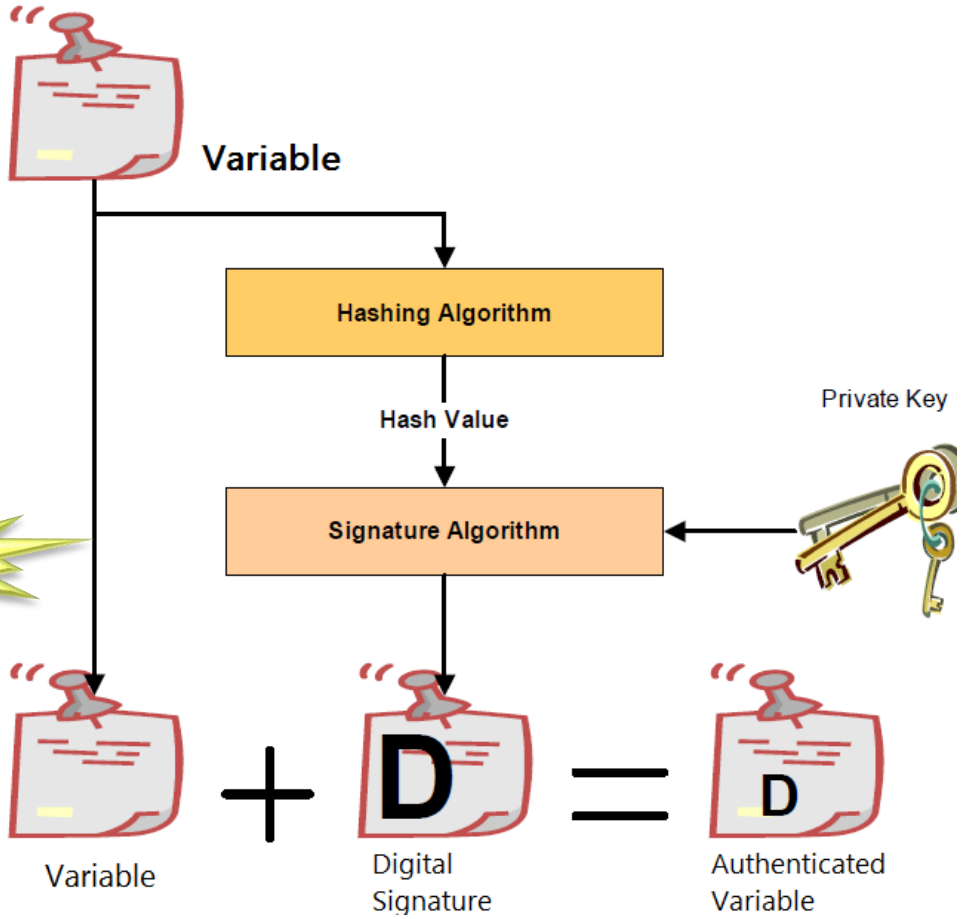
- Adds policy around UEFI and its 3rd party image extensibility
  - Admixture of OS loaders, apps, and drivers in system
  - Gives IT control around these executables
  - Detects/prevents malware
- Technology includes
  - Supports "known-good" and "known-bad" signature databases
  - Policy-based updates to list
  - Authenticode* signature types (Windows Authenticode Portable Executable Signature Format)



Extensible integrity architecture

UEFI Authenticated Variable

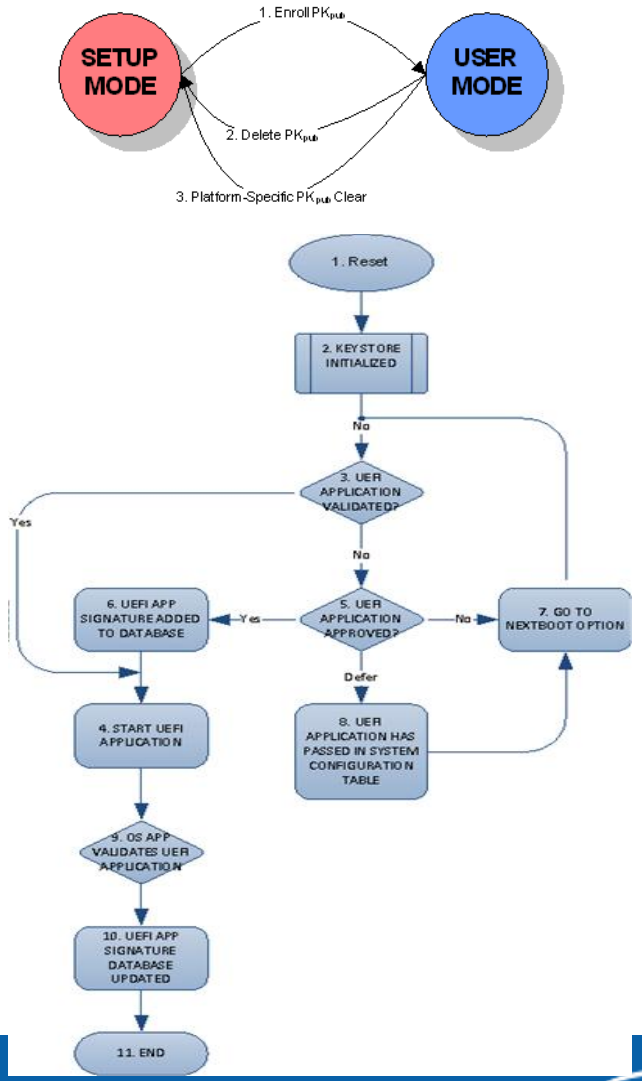
- Counter-based authenticated variable (UEFI 2.3)
  - Uses monotonic count to against suspicious replay attack
  - Hashing algorithm – SHA256
  - Signature algorithm – RSA-2048
- Time-based authenticated variable (UEFI 2.3.1)
  - Use EFI_TIME as rollback protection mechanism
  - Hashing algorithm – MD5/SHA1/SHA224/SHA256
  - Signature algorithm – X.509 certificate chains
    - Complete X.509 certificate chain
    - Intermediate certificate support (non-root certificate as trusted certificate.



UEFI Secure Boot

-Extensive Improvement to UEFI 2.3.1

- Platform security and integrity
  - Allows firmware to authenticate UEFI images, such as OS loader
  - Ensures firmware drivers are loaded in an owner-authorized fashion
- Technology includes
  - Global defined variables
    - Platform Key (PK)
    - Key Exchange Key (KEK)
  - Authenticated variable service, an enhancement on runtime variable service in UEFI
  - Driver signing, a means of embedding a digital signature of a UEFI executable, and verifying the signature from an authorized source.
- Authentication process



Why Implement UEFI Secure Boot?

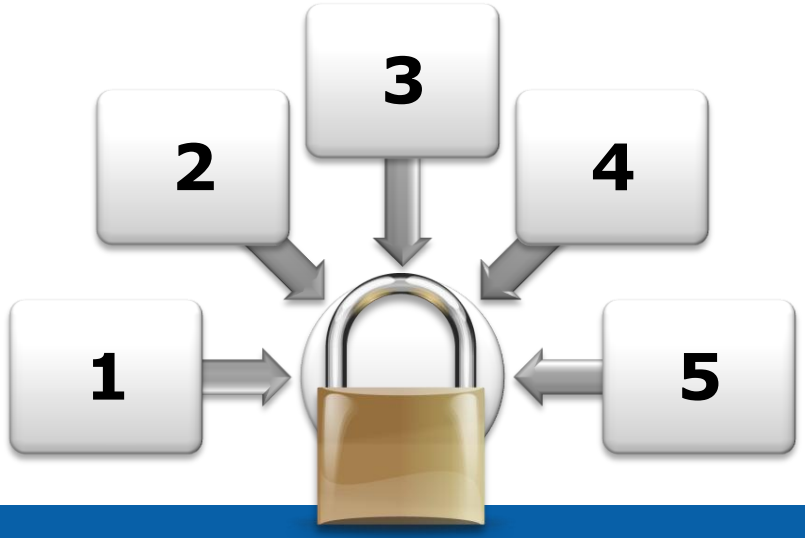
- As OS becomes more resistant to attack the threat targets the weakest element in the chain
- And 16-bit Legacy Boot is not secure!

*It should be no surprise that a TDL Gang botnet climbed into the **number one** position in the Damballa Threat Report – Top 10 Botnets of 2010. "RudeWarlockMob" … applied effective behaviors of old viruses and kits. It combined techniques that have been effective since the days of 16-bit operating systems, like Master Boot Record (MBR) infection … with newer malware techniques.*
(from http://blog.damballa.com)

- secure Boot based on UEFI 2.3.1 removes the Legacy Threat and provides software identity checking at every step of boot – Platform Firmware, Option Cards, and OS Bootloader
- Users want to be able to only run software they installed or purchased on the system and not what they happen to catch

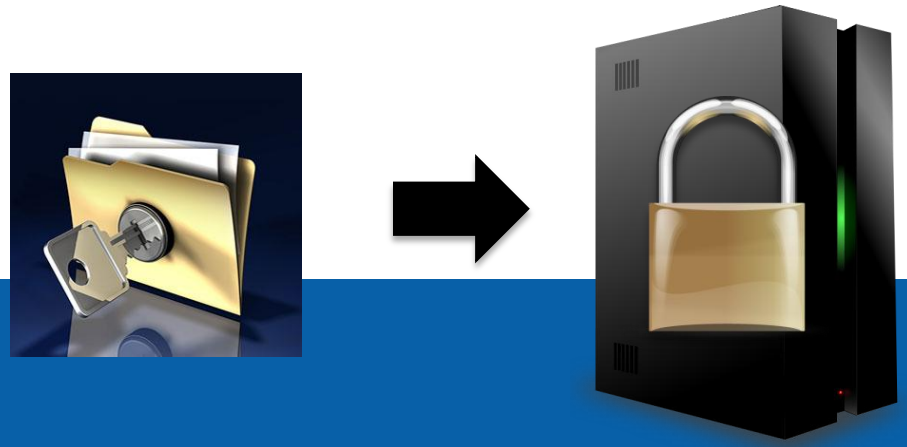
OEM/IHV Guide to UEFI 2.3.1 Secure Boot

- The Five Elements of Secure Boot Strategy:
  1. UEFI Platform Firmware with 2.3.1 implemented and backed by Strong Firmware Security Policies
  2. Hardware protection of critical security data
  3. Coordination from IBV, IHV and ISV partners
  4. UEFI Factory Provisioning and Field Support Tools
  5. Secure Firmware Update



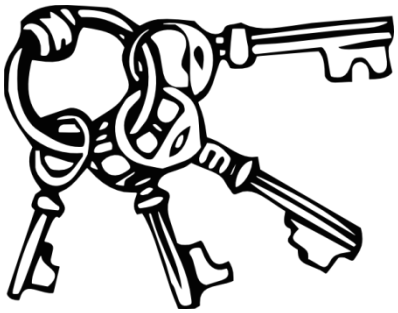
Element #1: UEFI Platform Firmware with 2.3.1 And Strong Firmware Security Policies

- UEFI 2.3.1 is an architectural specification
- But real security strength is in the policy enforcement
- **OEM-ACTION**→ Policy must lock-out untrusted code including all legacy 16-bit code
- But User Experience is key to acceptance:
  - *We ship locked-down secure systems but how much freedom should I give users to reconfigure?*
  - *How does my UI design minimize confusion from users used to "less secure" systems?*



Element #2: Hardware Protection of Critical Data

- Hardware protection of the key database is integral to a secure implementation
- **OEM-ACTION**→ Work with your chipset provider and IBV to implement strong protection of critical data



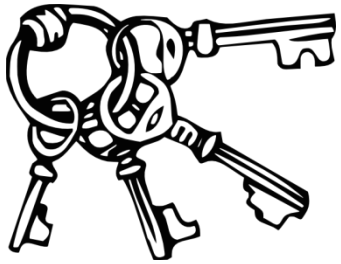
Element #3: Support from IBV, IHV & ISV Partners

- **OEM-ACTION**→ System ROM will need to contain UEFI drivers for all onboard devices (and no legacy drivers)
- **IHV-ACTION**→ Expansion cards will need Signed UEFI drivers
- **ISV-ACTION**→ Pre-boot software tools, for example bootable recovery disk, will need to be Signed

Element #4: Factory Provisioning

- Several new steps at the end of the factory flow will be required

- **OEM-ACTION**→ Provision with:
  - UEFI Key
  - OS Partner Key
  - OEM Support and Update Key
  - Install Platform Key to lock system



Element #4: . . . And Field Support Tools

- Any field support tools should be:
  - Signed UEFI executable (using UEFI Shell, not DOS)
  - Shipped pre-signed by the OEM key

- **OEM-ACTION**→ Examine field support flow, for example
  - Consider what users will do to reinitialize replacement motherboards?

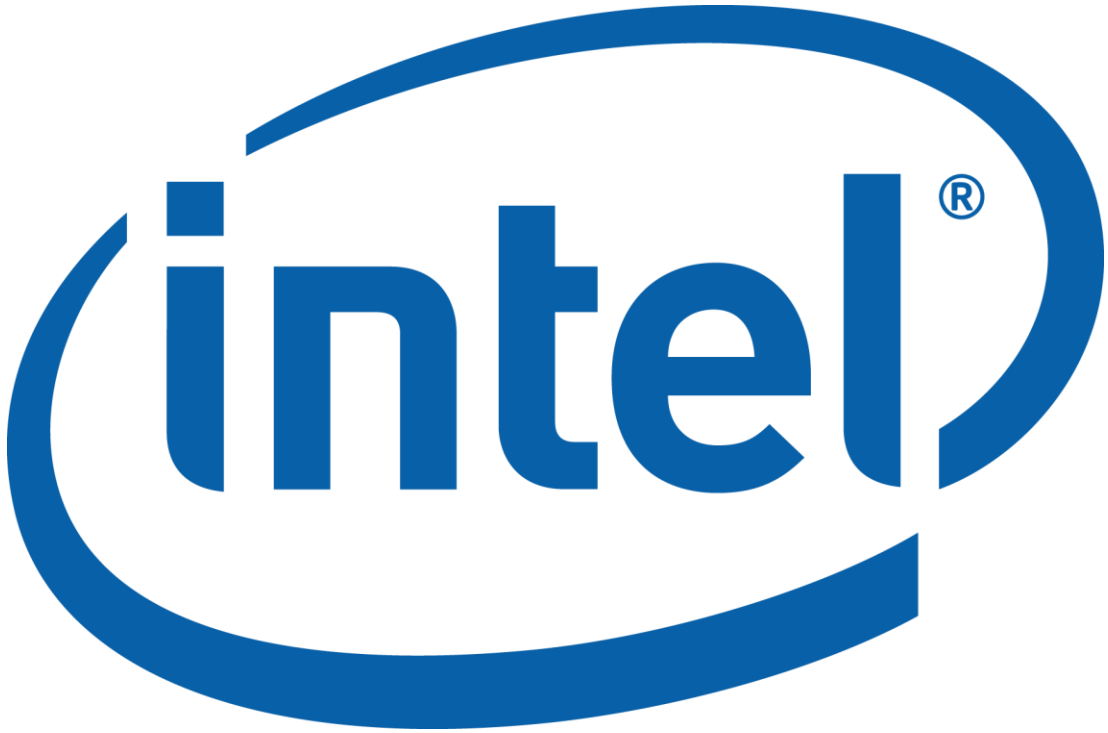
- Support the future - Enterprise Administrator install of Enterprise key
  - Can Enterprise buyer unlock new system and re-provision using your tools?

Element #5: Secure Firmware Update

- Security level of the Firmware Update must match system goals for security

OEM-ACTION→

1. Sign all Firmware Updates images
2. Firmware Update process must occur under control of secure firmware (not in OS)
3. H/W Flash Protection must reject any flash writes from unauthorized sources


intel


intel