# Security and Networking Advancements Today's UEFI and Intel® UEFI Development Kit 2010 (Intel® UDK2010)

Dong Wei, Hewlett Packard
Ting Ye, Intel
Jeff Bobzin, Insyde

EFIS005

Sponsors of Tomorrow.™ intel

# Agenda

- Latest UEFI specs releases
- Intel® UEFI Development Kit 2010 (Intel® UDK2010) Key Features
- Key UEFI Security and Network features Intel® UDK2010
- Implementing a Secure Boot Path with UEFI 2.3.1

# Agenda

- Latest UEFI specs releases
- Intel® UEFI Development Kit 2010 (Intel® UDK2010) Key Features
- Key UEFI Security and Network features Intel® UDK2010
- Implementing a Secure Boot Path with UEFI 2.3.1

# Industry BIOS Transition

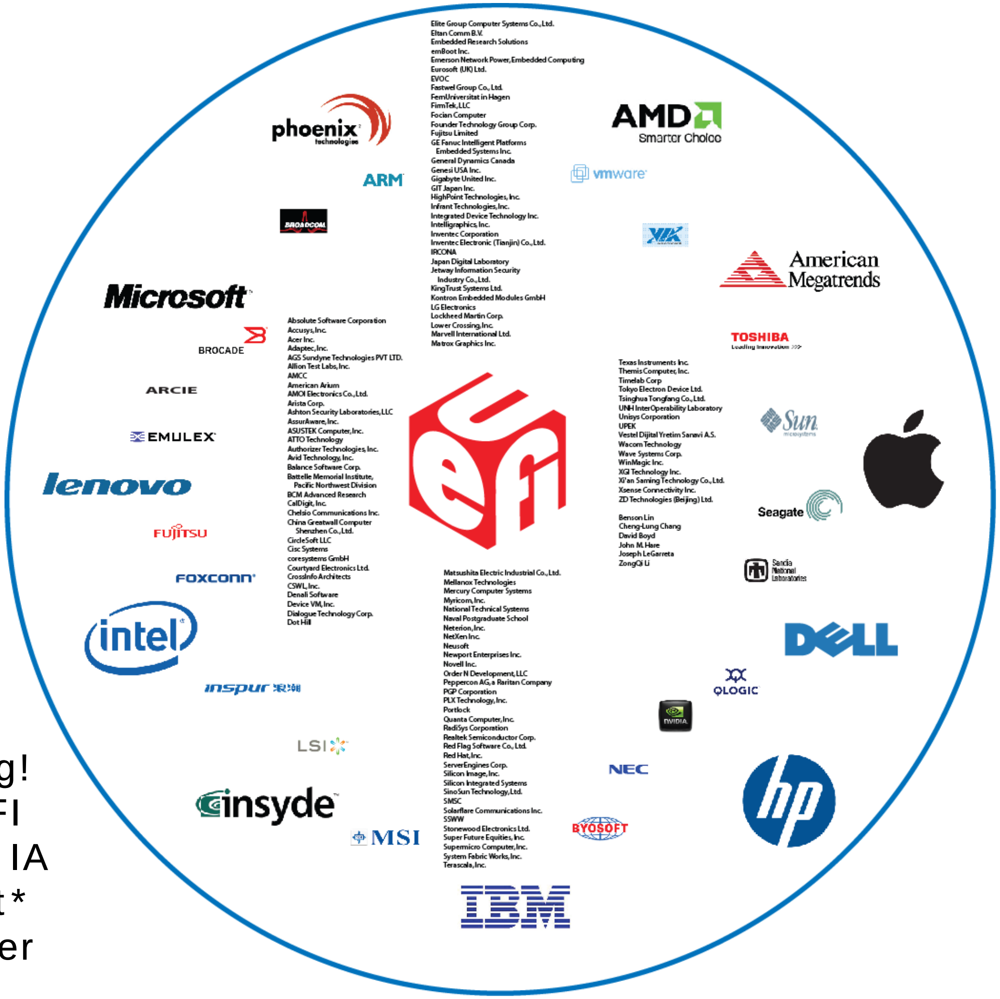
**Pre-2000** All Platforms BIOS were proprietary

**2000** Intel invented the Extensible Firmware Interface (EFI) and provided sample implementation under free BSD terms

**2004** **tianocore.org**, open source EFI community launched

**2005** **Unified EFI (UEFI)** Industry forum, with 11 promoters, was formed to standardize EFI

**2011** 170 members and growing! Major MNCs shipping; UEFI platforms crossed 50% of IA worldwide units; Microsoft* UEFI x64 support in Server 2008, Vista* and Win7*; RedHat* and Novell* OS support



Absolute Software Corporation
Accusys, Inc.
Acer Inc.
Adaptec, Inc.
AGS Sundyne Technologies PVT LTD.
Allion Test Labs, Inc.
AMCC
American Arium
AMOI Electronics Co., Ltd.
Arista Corp.
Ashton Security Laboratories, LLC
AssurAware, Inc.
ASUSTEK Computer, Inc.
ATTO Technology
Authorizer Technologies, Inc.
Avid Technology, Inc.
Balance Software Corp.
Battelle Memorial Institute, Pacific Northwest Division
BCM Advanced Research
CalDigit, Inc.
Chelsio Communications Inc.
China Greatwall Computer Shenzhen Co., Ltd.
CircleSoft LLC
Cisc Systems
coresystems GmbH
Courtyard Electronics Ltd.
CrossInfo Architects
CSWL, Inc.
Denali Software
Device VM, Inc.
Dialogue Technology Corp.
Dot Hill

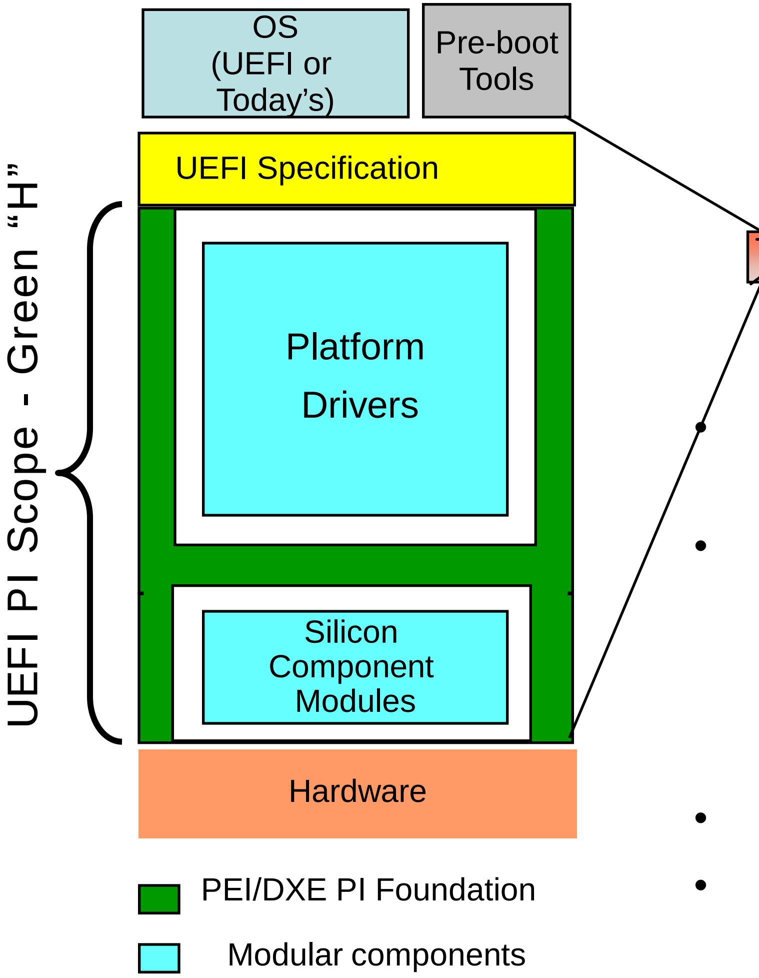
Elite Group Computer Systems Co., Ltd.
Eltan Comm B.V.
Embedded Research Solutions
emBoot Inc.
Emerson Network Power, Embedded Computing
Eurosoft (UK) Ltd.
EVOC
Fastwel Group Co., Ltd.
FernUniversitat in Hagen
FirmTek, LLC
Focian Computer
Founder Technology Group Corp.
Fujitsu Limited
GE Fanuc Intelligent Platforms Embedded Systems Inc.
General Dynamics Canada
Genesi USA Inc.
Gigabyte United Inc.
GIT Japan Inc.
HighPoint Technologies, Inc.
Infrant Technologies, Inc.
Integrated Device Technology Inc.
Intelligraphics, Inc.
Inventec Corporation
Inventec Electronic (Tianjin) Co., Ltd.
IRCONA
Japan Digital Laboratory
Jetway Information Security Industry Co., Ltd.
KingTrust Systems Ltd.
Kontron Embedded Modules GmbH
LG Electronics
Lockheed Martin Corp.
Lower Crossing, Inc.
Marvell International Ltd.
Matrox Graphics Inc.

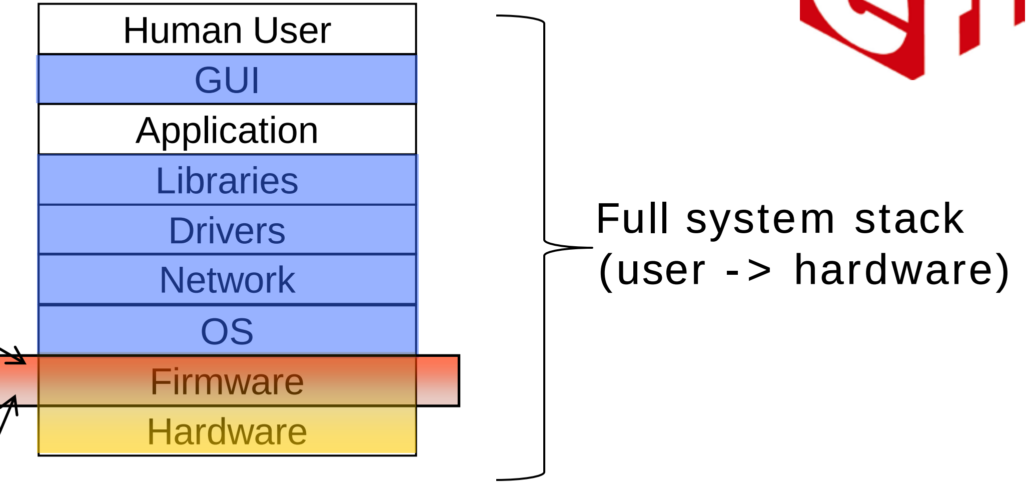
Matsushita Electric Industrial Co., Ltd.
Mellanox Technologies
Mercury Computer Systems
Myricom, Inc.
National Technical Systems
Naval Postgraduate School
Neterion, Inc.
NetXen Inc.
Neusoft
Newport Enterprises Inc.
Novell Inc.
Order N Development, LLC
Peppercon AG, a Raritan Company
PGP Corporation
PLX Technology, Inc.
Portlock
Quanta Computer, Inc.
RadiSys Corporation
Realtek Semiconductor Corp.
Red Flag Software Co., Ltd.
Red Hat, Inc.
ServerEngines Corp.
Silicon Image, Inc.
Silicon Integrated Systems
SinoSun Technology, Ltd.
SMSC
Solarflare Communications Inc.
SSWW
Stonewood Electronics Ltd.
Super Future Equities, Inc.
Supermicro Computer, Inc.
System Fabric Works, Inc.
Terascala, Inc.

Texas Instruments Inc.
Themis Computer, Inc.
Timelab Corp
Tokyo Electron Device Ltd.
Tsinghua Tongfang Co., Ltd.
UNH InterOperability Laboratory
Unisys Corporation
UPEK
Vestel Dijital Yretim Sanavi A.S.
Wacom Technology
Wave Systems Corp.
WinMagic Inc.
XGI Technology Inc.
Xi'an Saming Technology Co., Ltd.
Xsense Connectivity Inc.
ZD Technologies (Beijing) Ltd.

Benson Lin
Cheng-Lung Chang
David Boyd
John M. Hare
Joseph LeGarreta
ZongQi Li

# UEFI Platform Initialization Overview





- UEFI 2.3.1 specifies how firmware boots OS loader
- UEFI's Platform Initialization (PI) 1.2 Architecture specifies how Driver Execution Environment (DXE) Drivers and Pre-EFI Initialization (PEI) Modules (PEIMs) initialize SI and the platform
- DXE is preferred UEFI Implementation
- PEIMs, UEFI and DXE drivers implements networking, Update, other security features

# UEFI 2.3.1 Specification Update

**Security**

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

**Network**

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**

Non-blocking interface for BLOCK oriented devices

**Technology**

USB 3.0

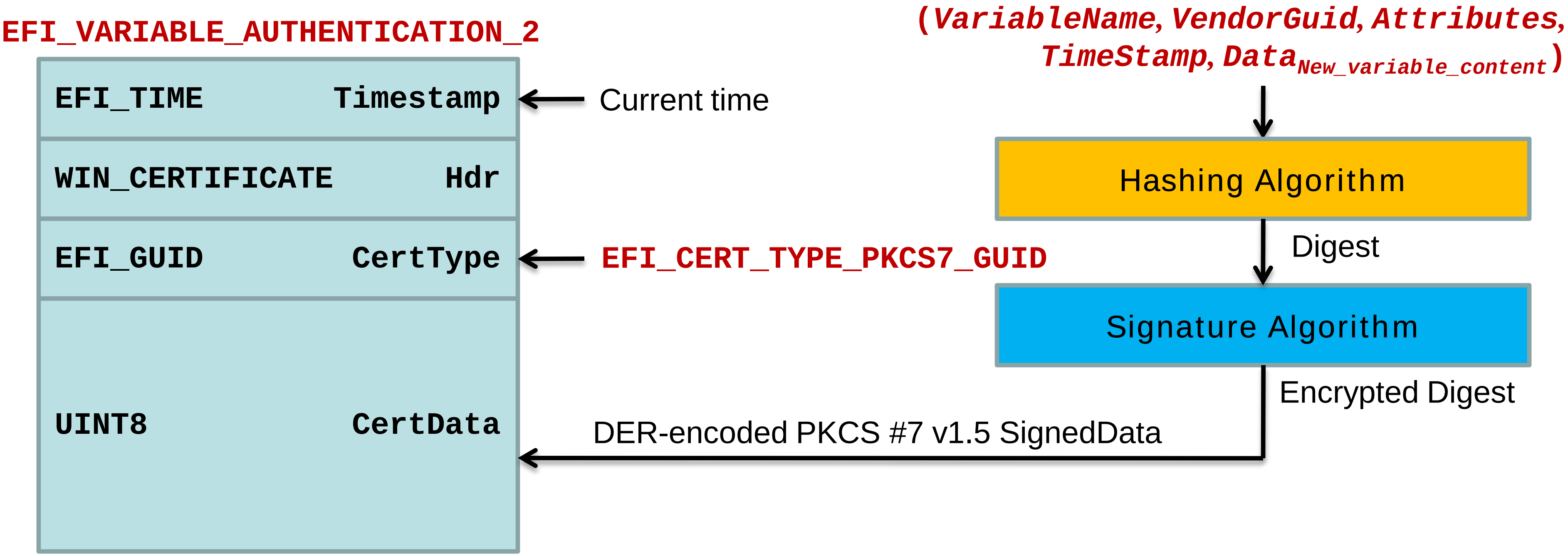
**Maintenance**

User Identifier, etc.

UEFI 2.3.1 Enabling More Security Support

# Security Update

- Time-based authenticated Variable
  - Certificate chaining infrastructure
  - Absolute time for rollback protection
  - Append operation for Signature Databases



EFI_VARIABLE_AUTHENTICATION_2

EFI_TIME Timestamp ← Current time

WIN_CERTIFICATE Hdr

EFI_GUID CertType ← EFI_CERT_TYPE_PKCS7_GUID

UINT8 CertData

(VariableName, VendorGuid, Attributes, TimeStamp, Data$_{New\_variable\_content}$)

Hashing Algorithm

Digest

Signature Algorithm

Encrypted Digest

DER-encoded PKCS #7 v1.5 SignedData

***Better support servicing of UEFI Secure Boot in a large ecosystem with many actors***

# UEFI 2.3.1 Security Spec Update

- Key Management Service (KMS)
  - Services to generate, store, retrieve, and manage cryptographic keys
  - Based on remote key server, or local Hardware Security Module (HSM), or software
- Storage Security Command Protocol
  - Send/receive security protocol defined data to/from mass storage devices
  - Supported command set
    - **TRUSTED SEND/RECEIVE** (ATA8-ACS)
    - **SECURITY PROTOCOL IN/OUT** (SPC-4)

# UEFI 2.3.1 HII Spec Update

- Forms Browser Default Behavior
  - Series of clarifications and guidance for proper handling of default information
- Modal Form Support
  - Provide methods to better support UI abstractions that resemble error or confirmation dialogs
- New opcode for event initiated refresh of browser
  - Allows for a periodic event  to occur which can make the browser aware of the need to refresh context
  - This avoids impractical periodic refreshes which otherwise might affect performance of the underlying firmware
- Series of errata/clarifications
  - Proper clarification of questions with no variable storage

# UEFI Deployment @HP

Collaborate on HP UEFI features providing enhanced manageability, security and ease of use with shared UEFI-based diagnostics

- Embedded
  - Printers and Scanners including: Scanjet Enterprise 7000n*, Color Laserjet CM4540 MFP*, Color LaserJet CP5525*, LaserJet M4555 MFP*
  - Network and Storage
- Client PC
  - Notebooks and Tablets with HP Platform Innovations
    - Shipping Class 2 systems from 2008: Latest EliteBook* and ProBook (8560p/8560b/8460p/ 8460w/6460b/6360b)
- Desktops and Workstations
  - Adopted a common UEFI codebase
  - Shipping Class 2 systems
    - HP Z210* and Z210 SFF* Workstations
    - HP Compaq Elite 6200* and 8200* Desktop PCs
- Servers
  - HP Integrity Superdome 2* and Integrity Server Blades*
  - HP-UX, OpenVMS, HP Integrity Virtual Machine operating environments

- **UEFI / PI framework has enabled code sharing opportunities among business entities and with partners/vendors**
- **Working with industry partners for the next generation products**

Hewlett-Packard Company makes no warranty as to the accuracy or completeness of the foregoing material and hereby disclaims any responsibility therefor.

# Agenda

- Latest UEFI specs releases
- **Intel® UEFI Development Kit 2010 (Intel® UDK2010) Key Features**
- Key UEFI Security and Network features Intel® UDK2010
- Implementing a Secure Boot Path with UEFI 2.3.1

# Intel® UDK2010 Enables a Common Firmware Development Foundation Across the Compute Continuum

Gadgets
Smartphones
TVs
Intel® UDK2010
Networks
Desktop PCs
Notebooks
Netbooks
Data Center / Servers
Embedded: Auto, Signage, Printers, etc.

Intel® UEFI Development Kit 2010 (Intel® UDK2010)

# Intel® UDK2010 Key Features

### Industry Standards Compliance

- UEFI 2.0, UEFI 2.1, UEFI 2.2, UEFI 2.3; PI 1.0, PI 1.1, PI 1.2

### 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[1]

- 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[1]

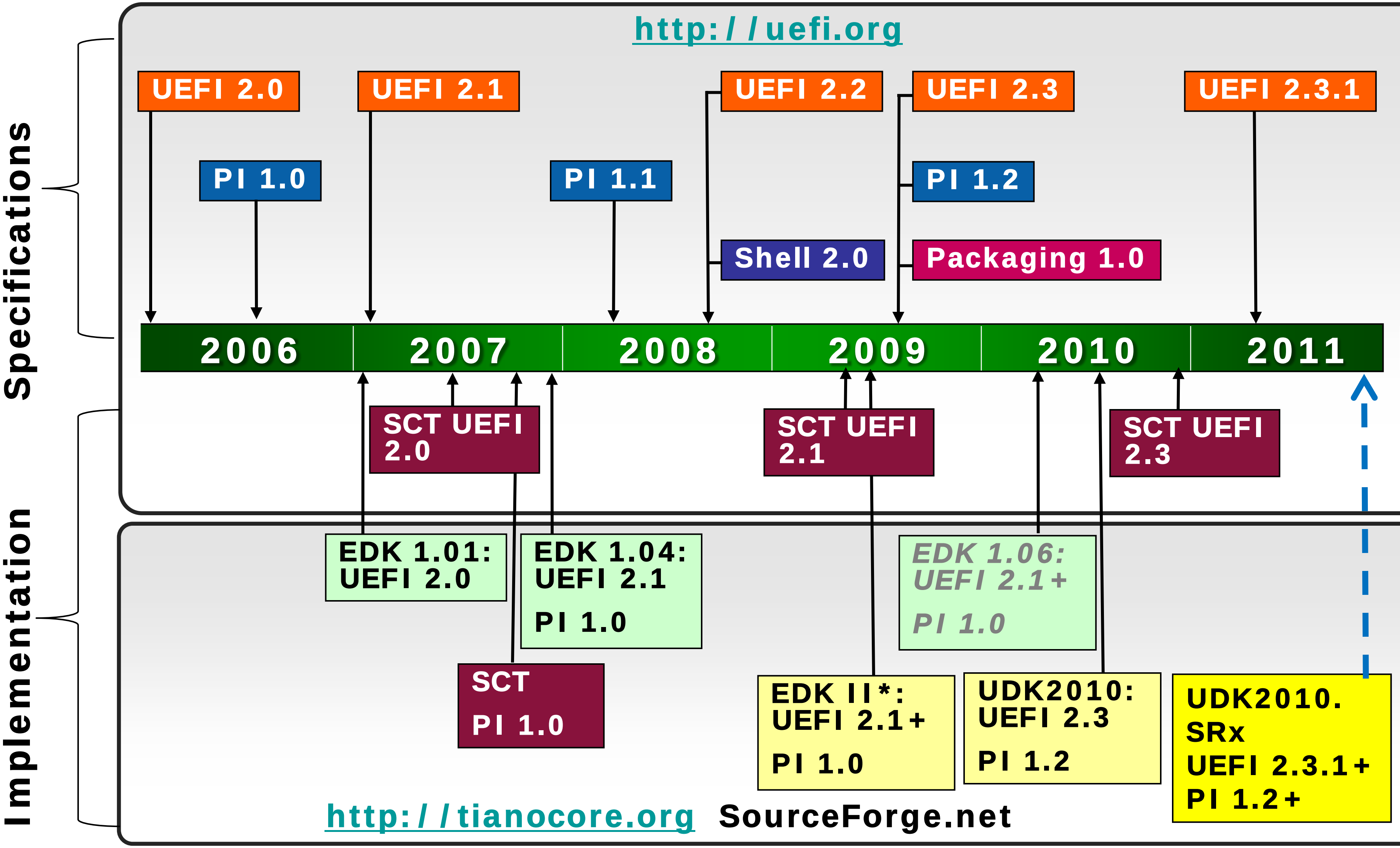
- Windows*, Linux*, OSX*
- VS2003, VS2005, WinDDK, Intel, GCC

### Fast and Flexible Build Infrastructure[1]

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

[1] benefit of EDK II codebase

# Specification & Tianocore.org Timeline

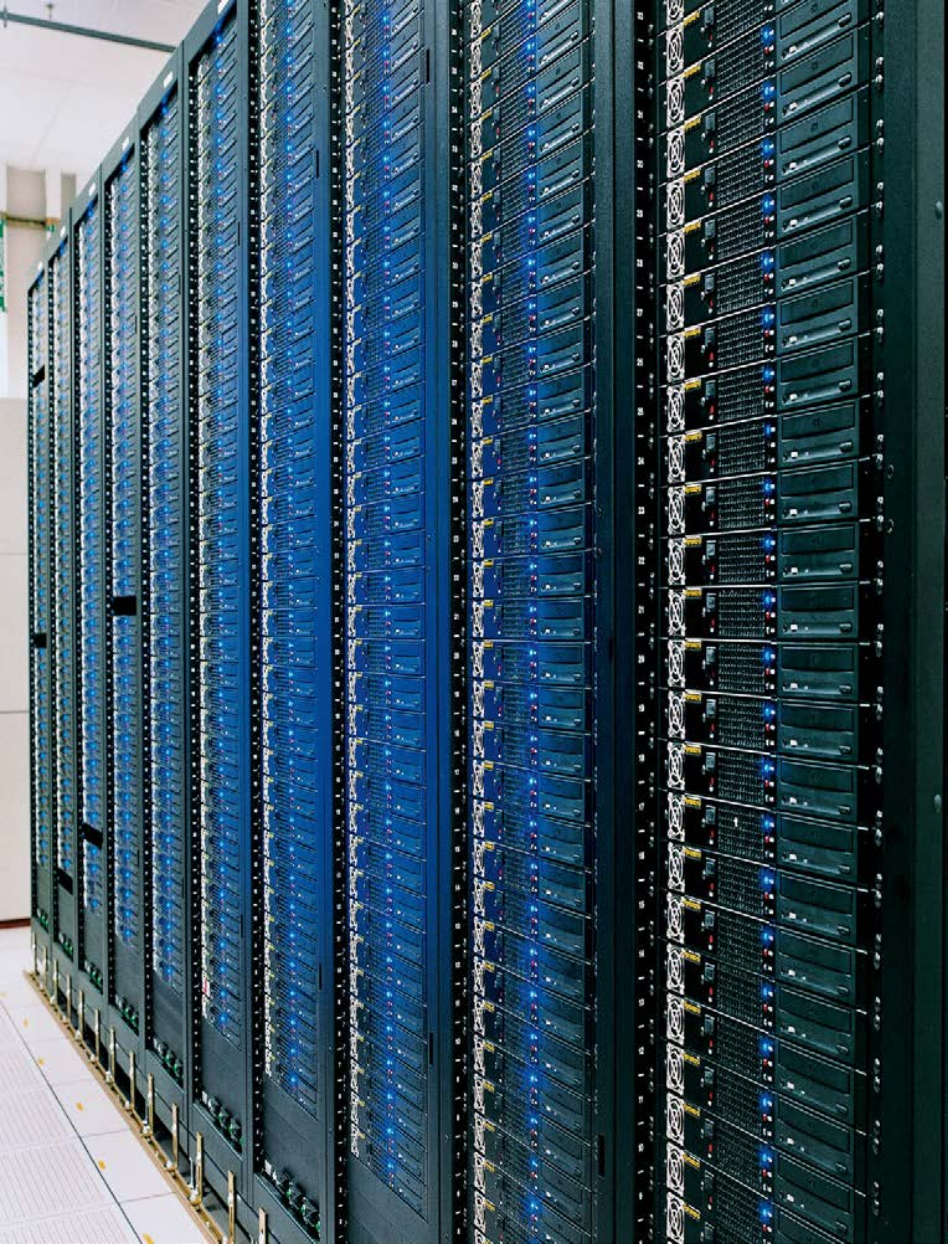


All products, dates, and programs are based on current expectations and subject to change without notice.

* EDK II is same code base as UDK2010

# Agenda

- Latest UEFI specs releases
- Intel® UEFI Development Kit 2010 (Intel® UDK2010) Key Features
- **Key UEFI Security and Network features Intel® UDK2010**
- Implementing a Secure Boot Path with UEFI 2.3.1

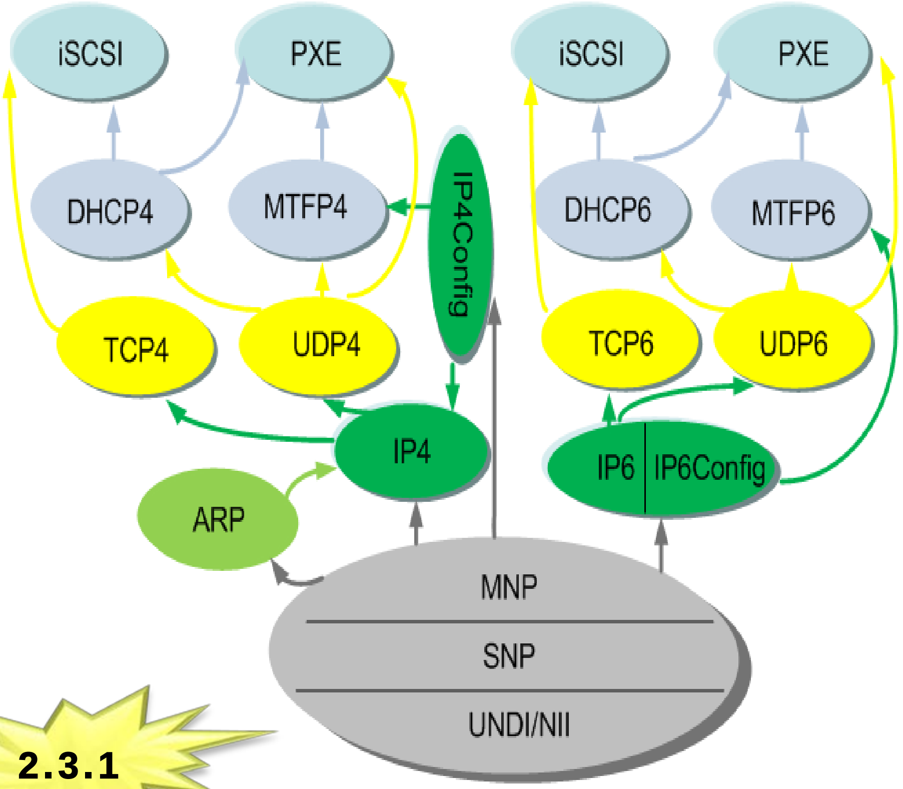


# Rich Networking

# 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
  - No IPv4 Addresses available

- Technology includes
  - IP4/6, UDP4/6, TCP4/6, DHCP4/6, MTFP4/6, iSCSI, PXE
    - Allows for concurrent network applications via design based upon MNP
    - Features dual stack: IP4, IP6, or both
- DUID-UUID support (UEFI 2.3.1)
  - Use SMBIOS system GUID as UUID

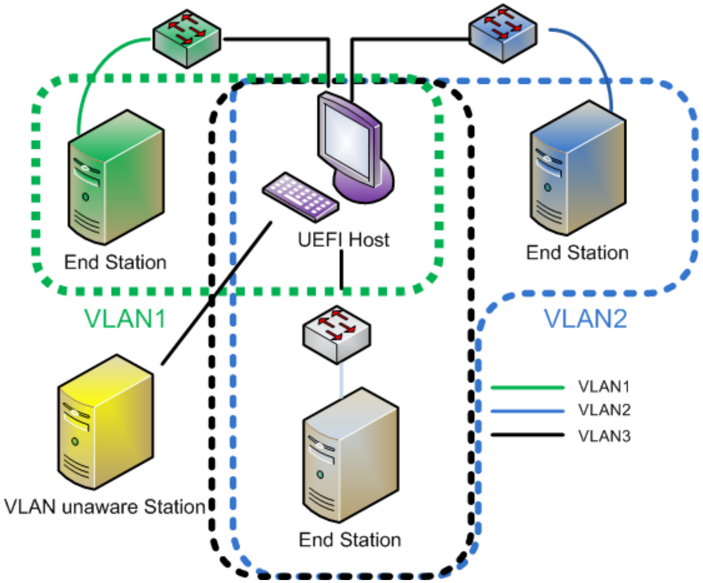


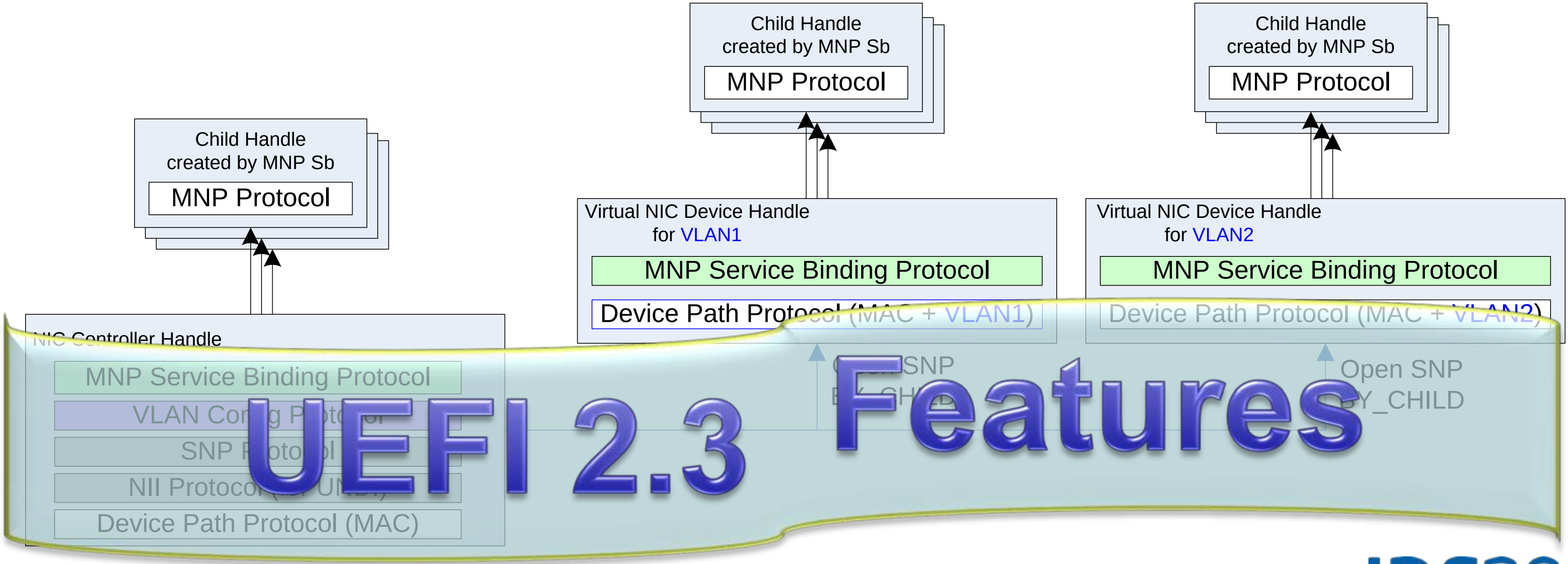
2.3.1

***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
- Technology includes
  - Support Hybrid LAN topology
  - Multiple VLAN for one station
  - VLAN configuration by HII





Child Handle created by MNP Sb

MNP Protocol

Child Handle created by MNP Sb

MNP Protocol

Child Handle created by MNP Sb

MNP Protocol

Virtual NIC Device Handle for VLAN1

MNP Service Binding Protocol

Device Path Protocol (MAC + VLAN1)

Virtual NIC Device Handle for VLAN2

MNP Service Binding Protocol

Device Path Protocol (MAC + VLAN2)

Open SNP BY_CHILD

Open SNP BY_CHILD

NIC Controller Handle

MNP Service Binding Protocol

VLAN Config Protocol

SNP Protocol

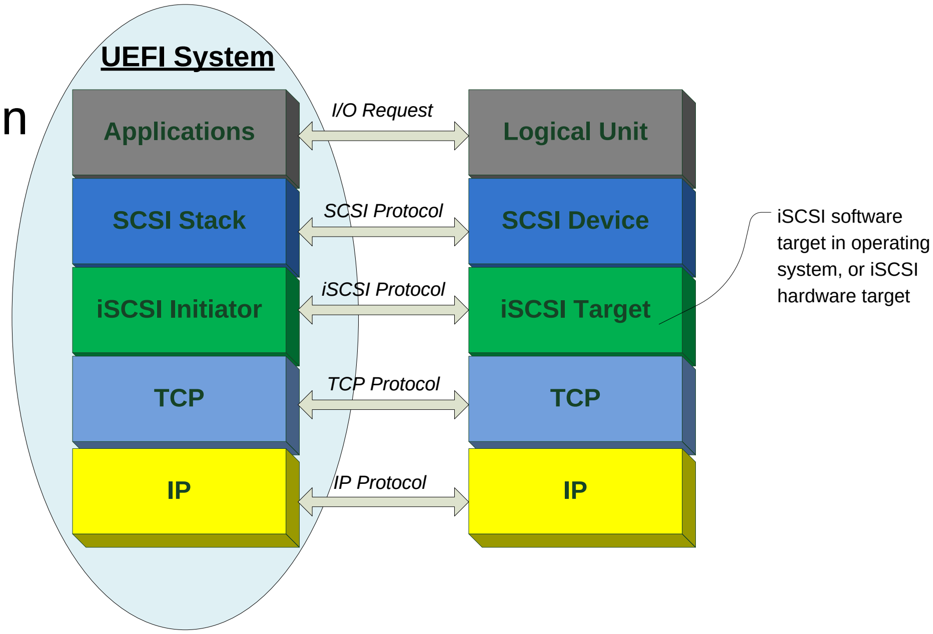
NII Protocol (UNDI)

Device Path Protocol (MAC)

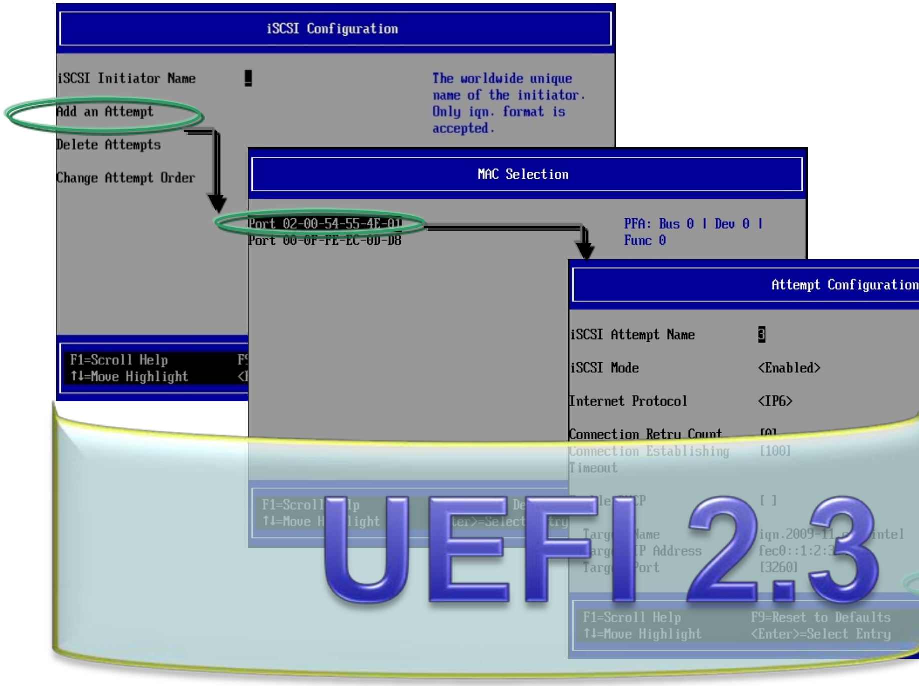
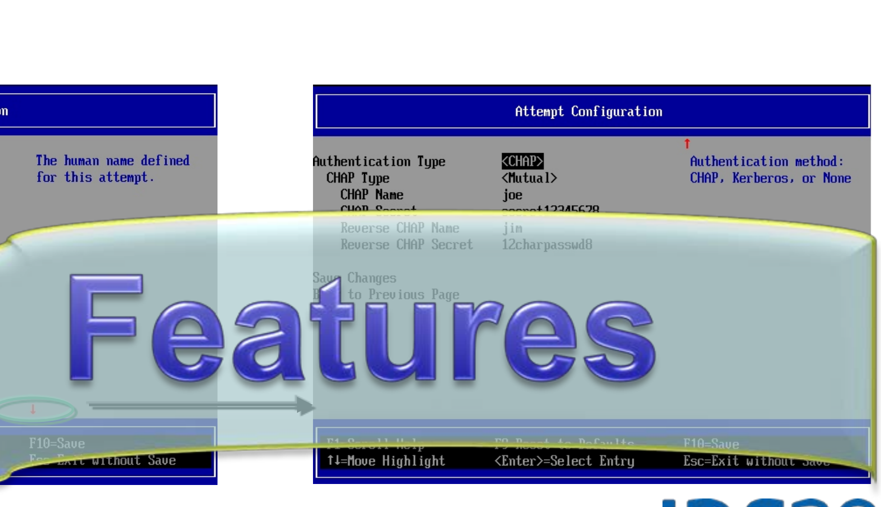
***Enabling the quarantining of networks***

# 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



iSCSI software target in operating system, or iSCSI hardware target

```
iSCSI Configuration

iSCSI Initiator Name                     The worldwide unique
                                         name of the initiator.
Add an Attempt                           Only iqn. format is
                                         accepted.
Delete Attempts

Change Attempt Order

F1=Scroll Help            F9
↑↓=Move Highlight         <I
```

```
MAC Selection

Port 02-00-54-55-4E-01                   PFA: Bus 0 | Dev 0 |
Port 00-0F-FE-EC-0D-D8                   Func 0

F1=Scroll Help               De
↑↓=Move Highlight         <Enter>=Select Entry
```

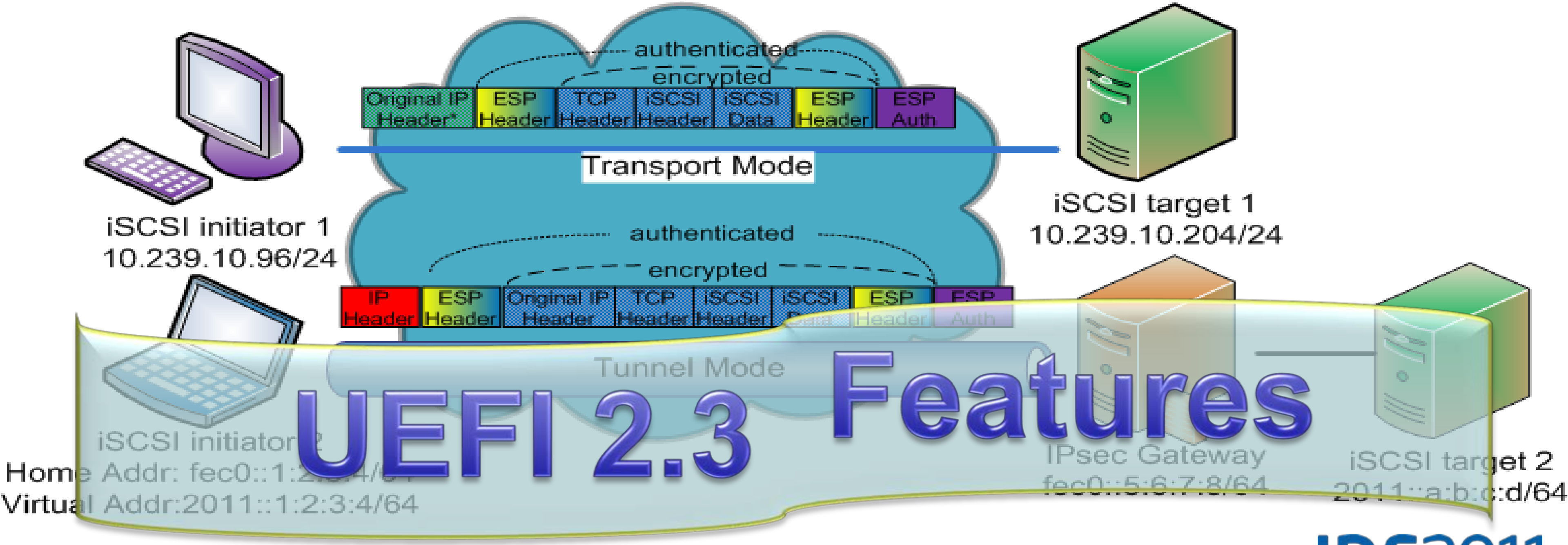
```
Attempt Configuration

iSCSI Attempt Name            3                 The human name defined
                                                for this attempt.
iSCSI Mode                    <Enabled>

Internet Protocol             <IP6>

Connection Retry Count        [0]
Connection Establishing       [100]
Timeout

Enable DHCP                   [ ]

  Target Name                 iqn.2009-11.com.intel
  Target IP Address           fec0::1:2:3
  Target Port                 [3260]

F1=Scroll Help            F9=Reset to Defaults     F10=Save
↑↓=Move Highlight         <Enter>=Select Entry     Esc=Exit without Save
```

```
Attempt Configuration

Authentication Type           <CHAP>            Authentication method:
  CHAP Type                   <Mutual>          CHAP, Kerberos, or None
    CHAP Name                 joe
    CHAP Secret               secret12345678
    Reverse CHAP Name         jim
    Reverse CHAP Secret       12charpasswd8

Save Changes
Back to Previous Page

F1=Scroll Help            F9=Reset to Defaults     F10=Save
↑↓=Move Highlight         <Enter>=Select Entry     Esc=Exit without Save
```

***Enabling Data Storage Scalability***

# 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
  - HMAC-SHA1, TripleDES-CBC, AES-CBC
  - Transport/Tunnel mode
  - Pre shared Key/X.509 certificate authentication

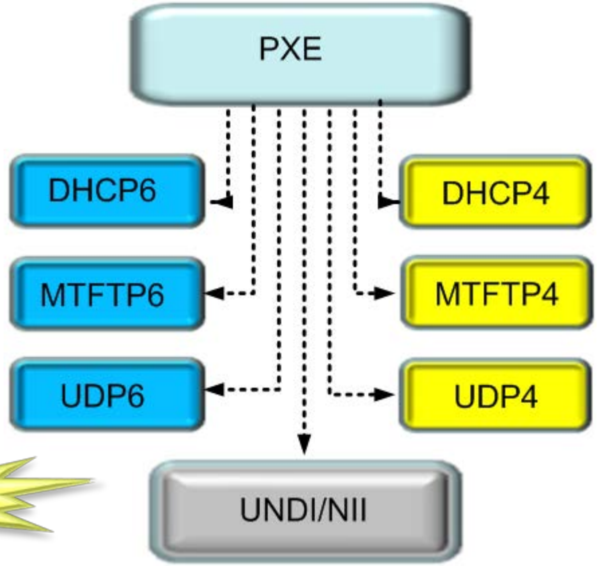


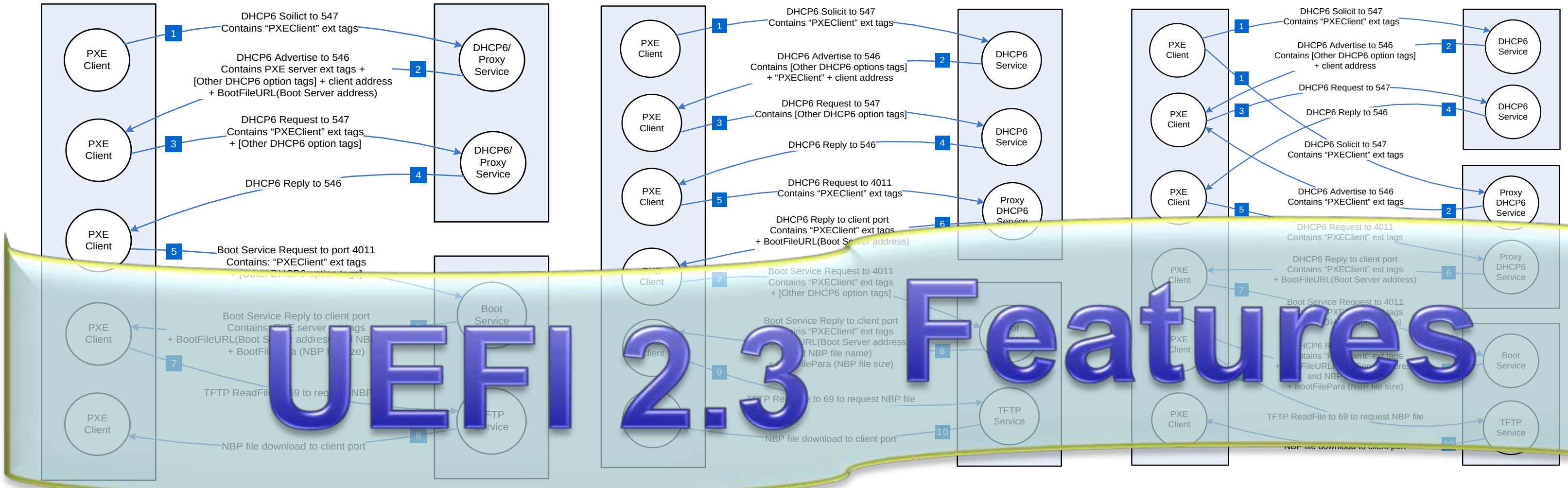
**_Improved Network Integrity_**

# UEFI PXE Solutions

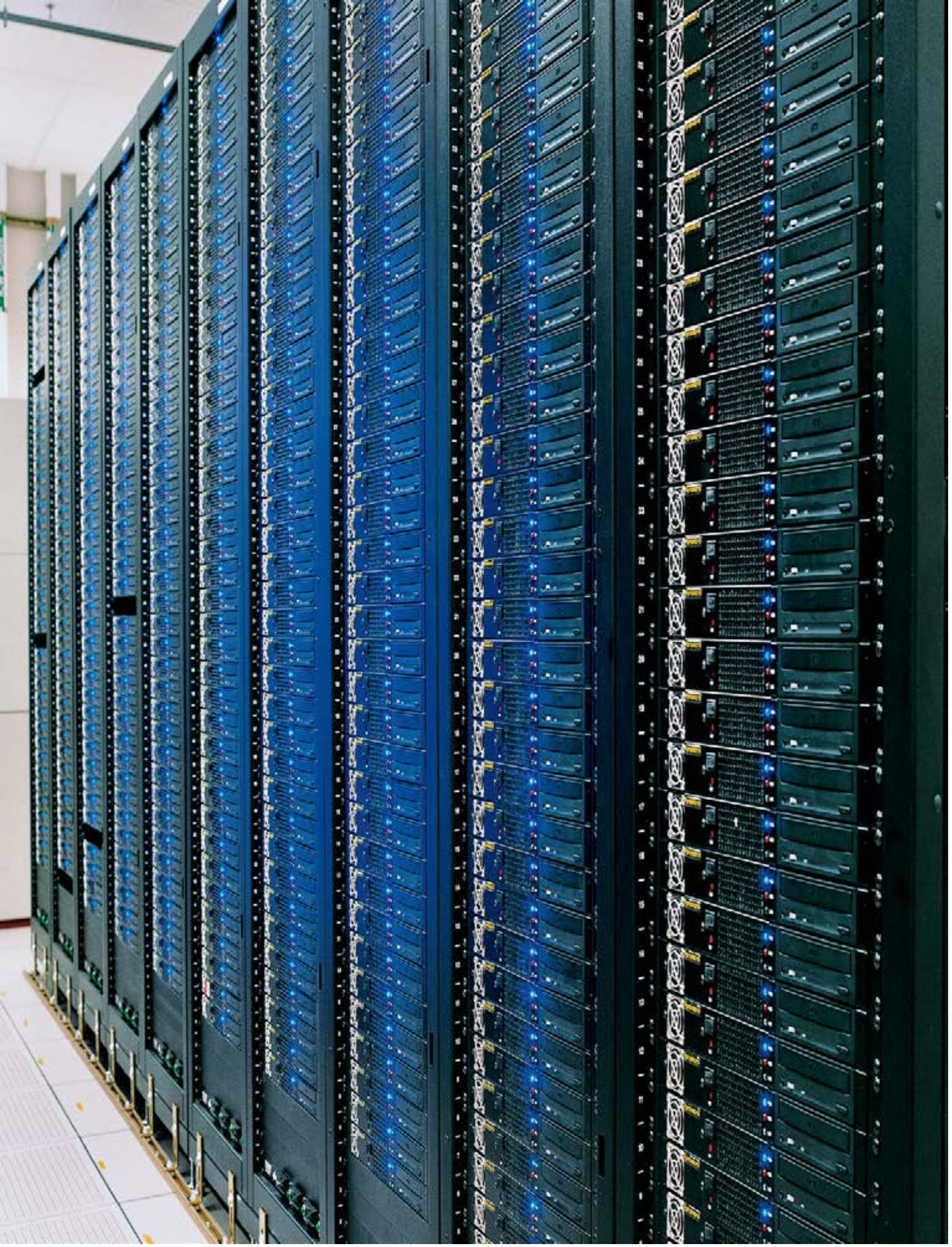
- Preboot eXecution Environment
  - General network booting
    - Independent of data storage device
  - IPv4 based PXE defined in PXE 2.1
  - IPv6 based PXE is defined in UEFI 2.3
- Technology includes
  - Dual network stack support
    - Evolution of network boot to IPv6 defined in IETF RFC 5970
  - DUID-UUID support
    - Use SMBIOS system GUID as UUID

2.3.1




PXE Client
DHCP6/ Proxy Service
1
DHCP6 Soilict to 547
Contains "PXEClient" ext tags
2
DHCP6 Advertise to 546
Contains PXE server ext tags + [Other DHCP6 option tags] + client address + BootFileURL(Boot Server address)
3
DHCP6 Request to 547
Contains "PXEClient" ext tags + [Other DHCP6 option tags]
4
DHCP6 Reply to 546
5
Boot Service Request to port 4011
Contains: "PXEClient" ext tags + [Other DHCP6 option tags]
Boot Service
Boot Service Reply to client port
Contains PXE server ext tags + BootFileURL(Boot Server address) + NBP + BootFileParam (NBP file size)
7
TFTP ReadFile to 69 to request NBP
TFTP Service
8
NBP file download to client port
DHCP6 Solicit to 547
Contains "PXEClient" ext tags
DHCP6 Service
DHCP6 Advertise to 546
Contains [Other DHCP6 options tags] + "PXEClient" + client address
DHCP6 Request to 547
Contains [Other DHCP6 option tags]
DHCP6 Reply to 546
DHCP6 Request to 4011
Contains "PXEClient" ext tags
Proxy DHCP6 Service
6
DHCP6 Reply to client port
Contains "PXEClient" ext tags + BootFileURL(Boot Server address)
Boot Service Request to 4011
Contains "PXEClient" ext tags + [Other DHCP6 option tags]
Boot Service Reply to client port
Contains "PXEClient" ext tags + BootFileURL(Boot Server address) + NBP file name) + BootFilePara (NBP file size)
9
TFTP ReadFile to 69 to request NBP file
10
NBP file download to client port
DHCP6 Advertise to 546
Contains [Other DHCP6 option tags] + client address
DHCP6 Request to 547
DHCP6 Reply to 546
DHCP6 Solicit to 547
Contains "PXEClient" ext tags
DHCP6 Advertise to 546
Contains "PXEClient" ext tags
DHCP6 Request to 4011
Contains "PXEClient" ext tags
DHCP6 Reply to client port
Contains "PXEClient" ext tags + BootFileURL(Boot Server address)
Boot Service Request to 4011
Boot Service Reply to client port
+ BootFilePara (NBP file size)
TFTP ReadFile to 69 to request NBP file
NBP file download to client port

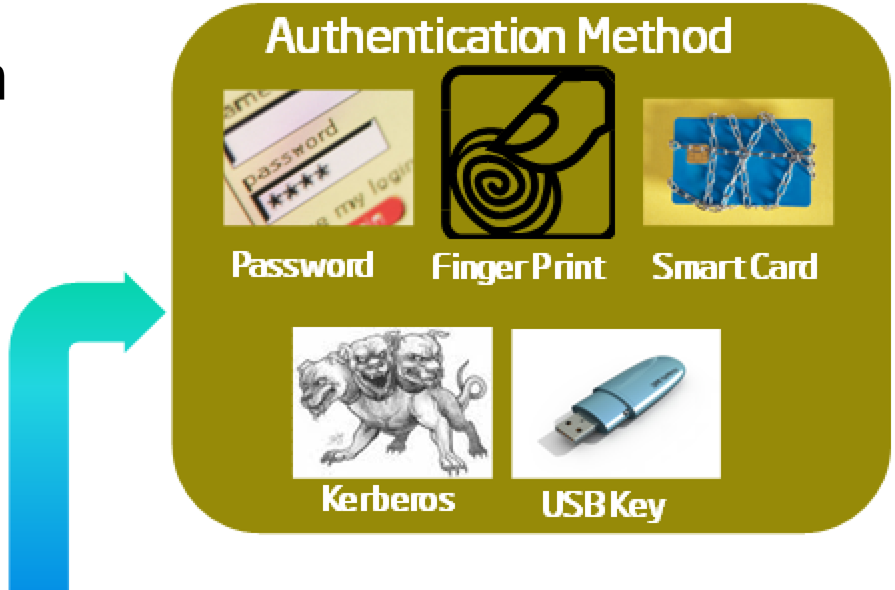
UEFI 2.3 Features

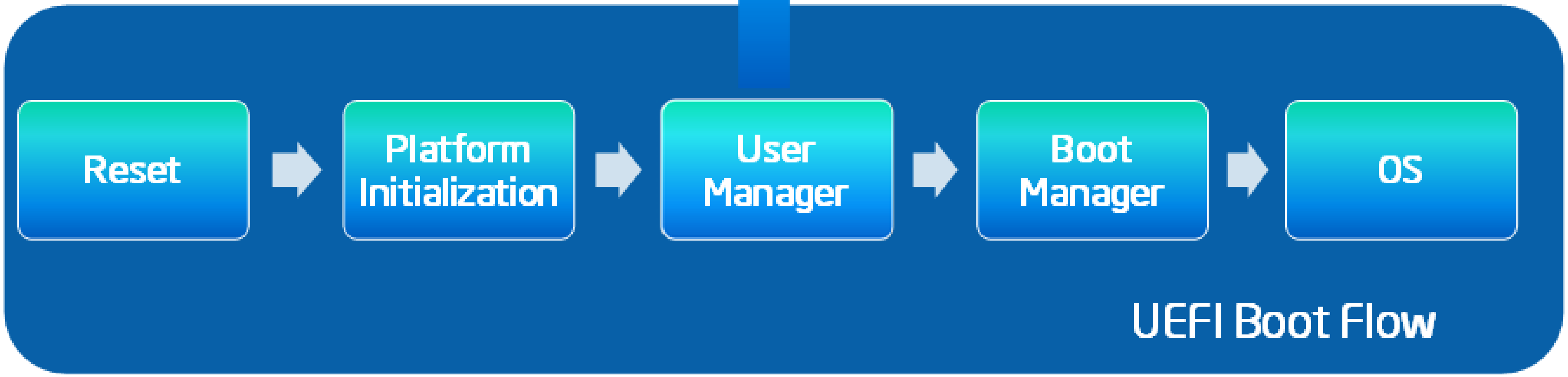


# Security Features

# 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



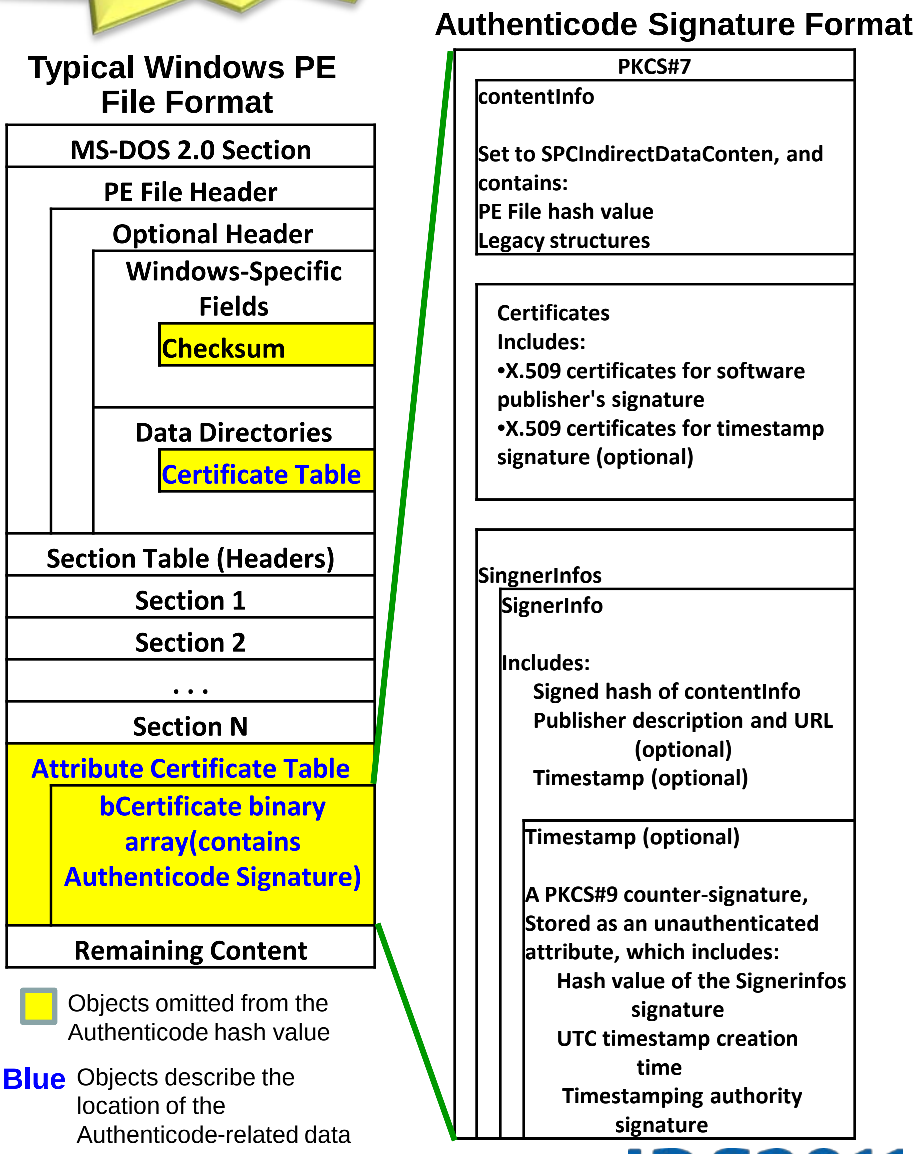


***Support for various pre-boot authenticators***

# UEFI Driver Signing

Enhanced by UEFI 2.3.1

- 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)



Typical Windows PE File Format

MS-DOS 2.0 Section

PE File Header

Optional Header

Windows-Specific Fields

Checksum

Data Directories

Certificate Table

Section Table (Headers)

Section 1

Section 2

. . .

Section N

Attribute Certificate Table

bCertificate binary array(contains Authenticode Signature)

Remaining Content

🟨 Objects omitted from the Authenticode hash value

**Blue** Objects describe the location of the Authenticode-related data

Authenticode Signature Format

PKCS#7

contentInfo

Set to SPCIndirectDataConten, and contains:
PE File hash value
Legacy structures

Certificates
Includes:
•X.509 certificates for software publisher's signature
•X.509 certificates for timestamp signature (optional)

SingnerInfos

SignerInfo

Includes:
Signed hash of contentInfo
Publisher description and URL (optional)
Timestamp (optional)

Timestamp (optional)

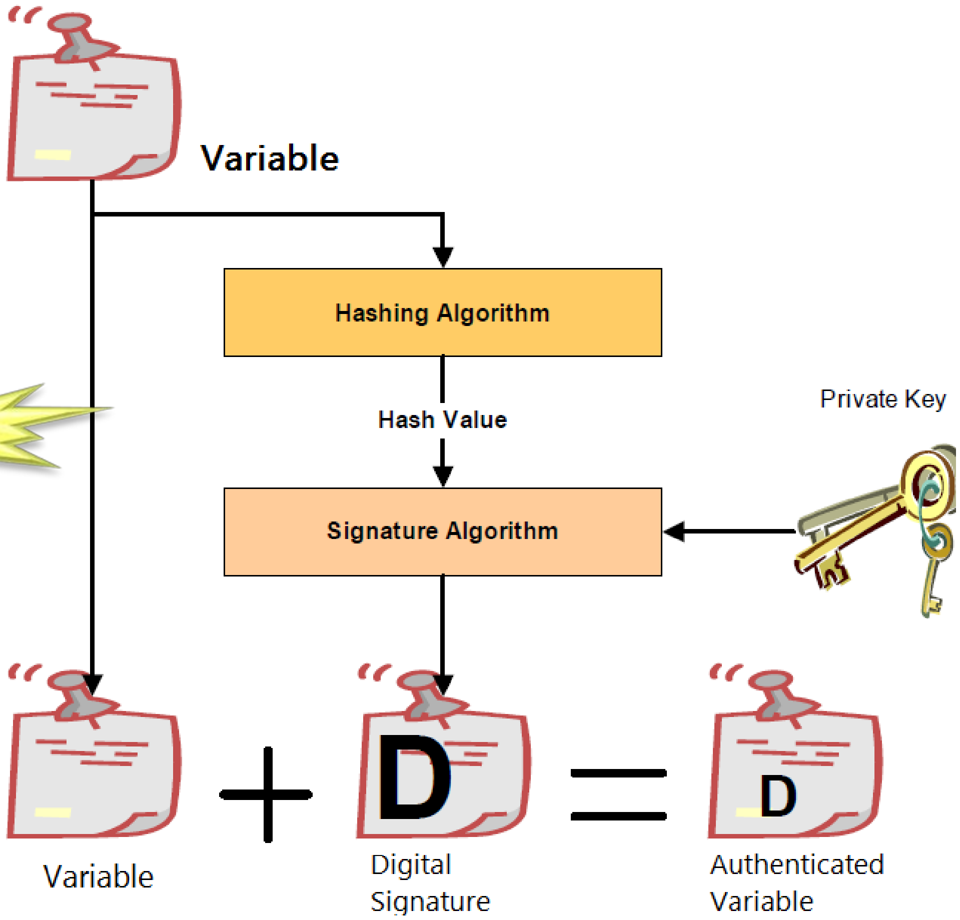
A PKCS#9 counter-signature, Stored as an unauthenticated attribute, which includes:
Hash value of the Signerinfos signature
UTC timestamp creation time
Timestamping authority signature

**_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.

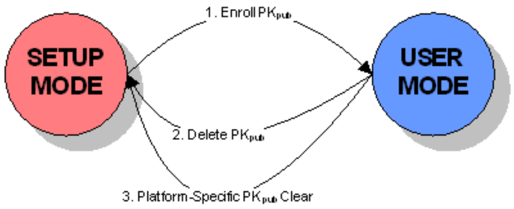
2.3.1

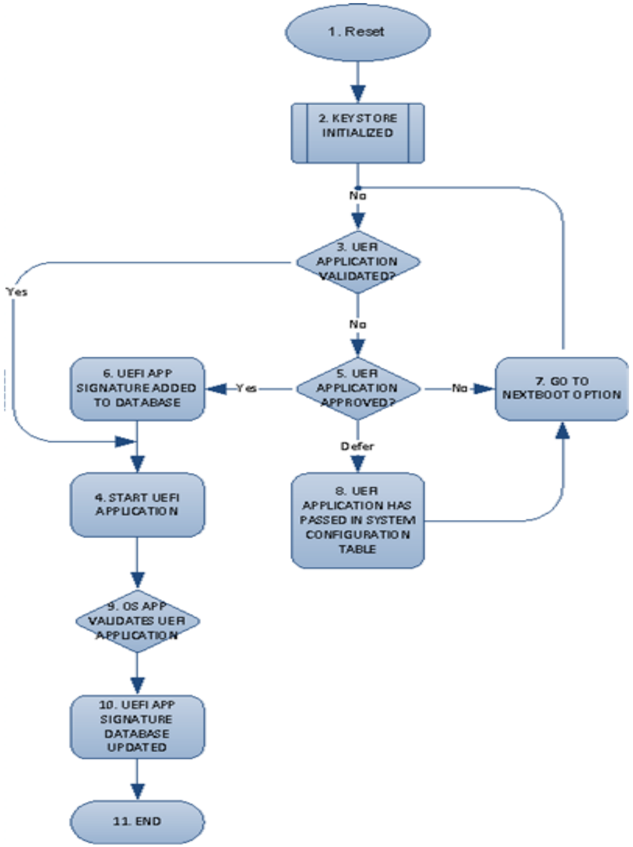


# UEFI Secure Boot
## Extensive Improvement to UEFI 2.3.1

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




1. Reset
2. KEYSTORE INITIALIZED
No
3. UEFI APPLICATION VALIDATED?
Yes
No
5. UEFI APPLICATION APPROVED?
Yes
6. UEFI APP SIGNATURE ADDED TO DATABASE
No
7. GO TO NEXTBOOT OPTION
Defer
8. UEFI APPLICATION HAS PASSED IN SYSTEM CONFIGURATION TABLE
4. START UEFI APPLICATION
9. OS APP VALIDATES UEFI APPLICATION
10. UEFI APP SIGNATURE DATABASE UPDATED
11. END

IDF2011
INTEL DEVELOPER FORUM

# Agenda

- Latest UEFI specs releases
- Intel® UEFI Development Kit 2010 (Intel® UDK2010) Key Features
- Key UEFI Security and Network features
- **Implementing a Secure Boot Path with UEFI 2.3.1**

# Why Implement UEFI Secure Boot?

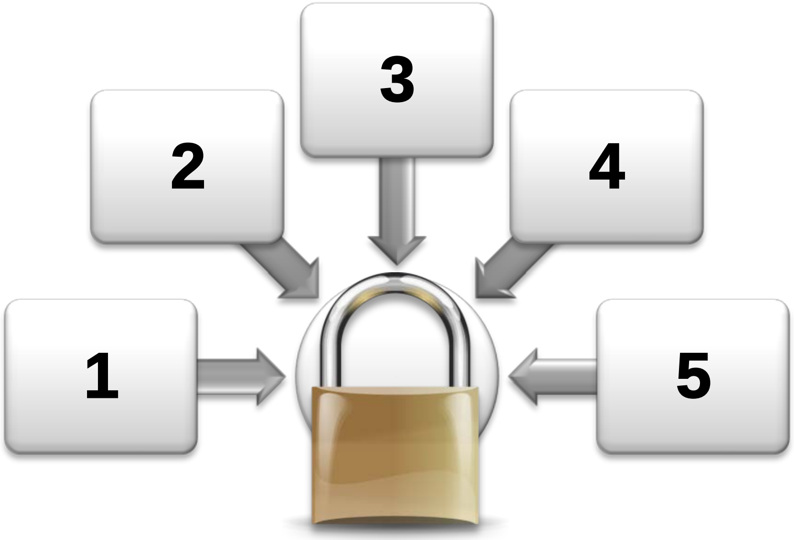
- As OS becomes more resistant to attack the threat targets the weakest element in the chain
- And 16-bit Legacy Boot is not secure!

> *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

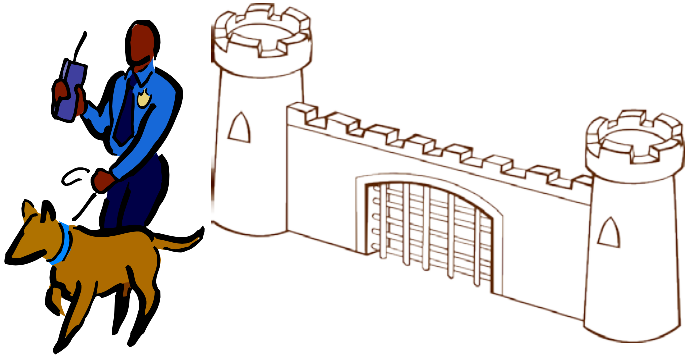
# OEM / IHV Guide to UEFI 2.3.1 Secure Boot

- 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



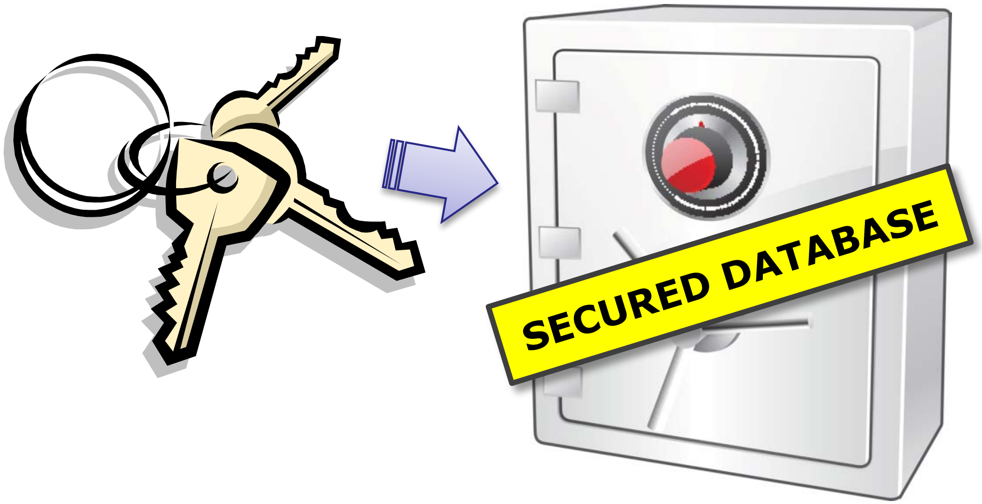
# Element #1: UEFI Platform Firmware with 2.3.1 And Strong Firmware Security Policies

- 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?*



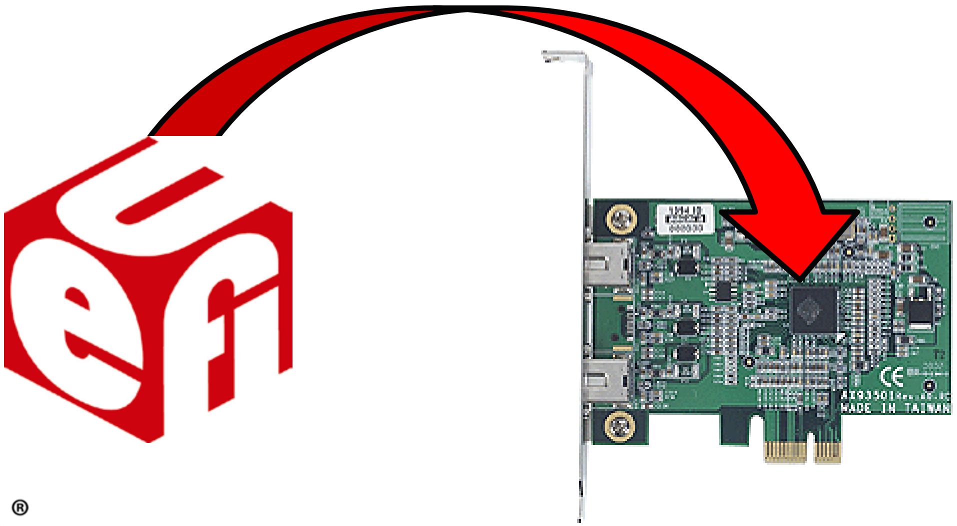
# Element #2: Hardware Protection of Critical Data

- 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

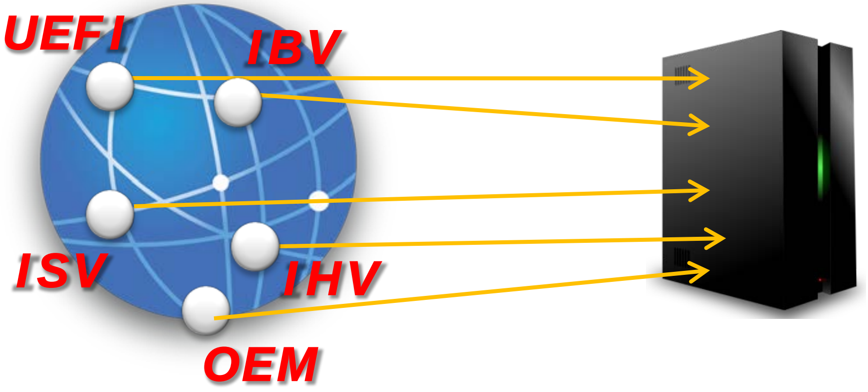
# DEMO
# Signing Test Tool

# Summary

- Industry transition from Legacy to UEFI will impact all industry segments this year
- UEFI 2.3.1 spec update adds significant new value allowing improved protection of the UEFI systems
- Driver signing and authenticated variables are key tools for constructing UEFI Secure Boot
- OEMs need to implement UEFI Secure Boot as part of an integrated strategy in concert with IHV and ISV partners

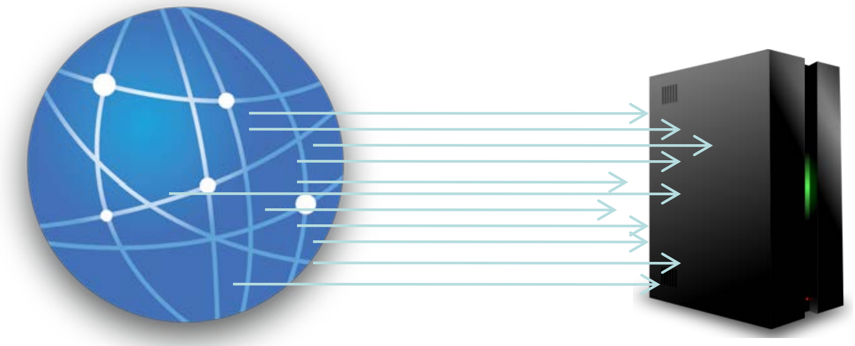
# Next Steps

- Join UEFI if not already a member
- Download the new UEFI 2.3.1 Spec from www.uefi.org
- OEMs need to implement UEFI boot and use UEFI 2.3.1 security features to harden their systems
- OEMs must work with IBV, IHV and ISV partners in coordinated approach



# Next Steps

- Join UEFI if not already a member
- Download the new UEFI 2.3.1 Spec from www.uefi.org
- OEMs need to implement UEFI boot and use UEFI 2.3.1 security features to harden their systems
- OEMs must work with IBV, IHV and ISV partners in coordinated approach



# Additional resources on UEFI:

- Other UEFI Sessions – Please download the PDFs
- More web based info:
  - Specifications sites www.uefi.org, www.intel.com/technology/efi
  - EDK II Open Source Implementation: www.tianocore.org

- Technical book from Intel Press: "Beyond BIOS: Implementing the Unified Extensible Firmware Interface with Intel's Framework" www.intel.com/intelpress

# Session Presentations - PDFs

The PDF for this Session presentation is available from our IDF Content Catalog at the end of the day at:

intel.com/go/idfsessionsBJ

URL is on top of Session Agenda Pages in Pocket Guide

# Please Fill out the Session Evaluation Form

## Give the completed form to the room monitors as you exit!

**Thank You for your input, we use it to improve future Intel Developer Forum events**

# Q&A

# Backup Slides

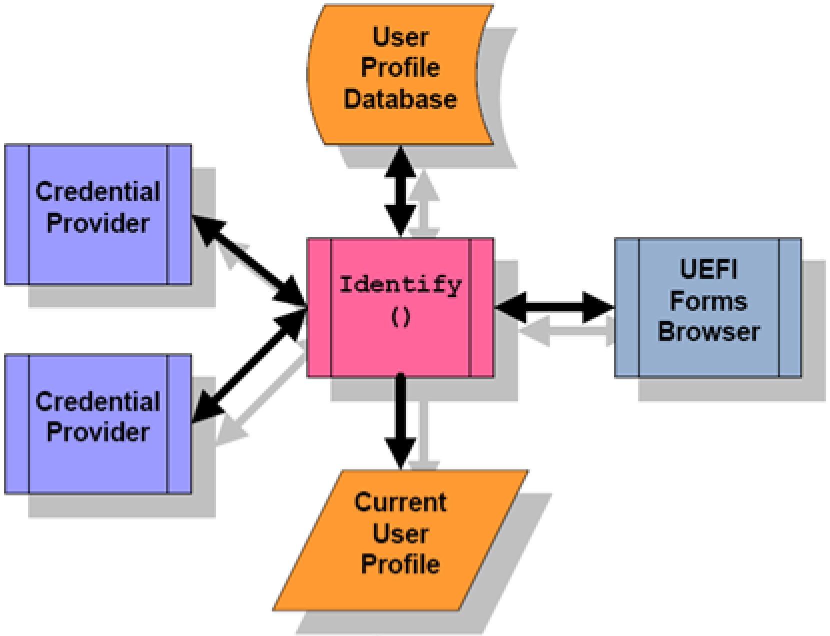
# EFI Track Sessions

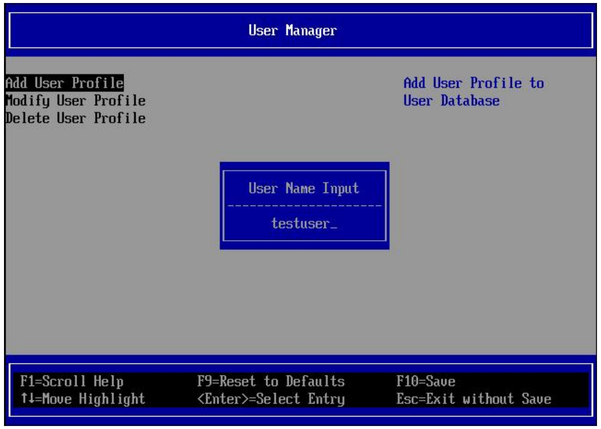
| Session ID | Title | Day/ Time | Room |
|---|---|---|---|
| EFIS001 ✓ | Microsoft* Windows* Platform Evolution and UEFI | Tuesday 11:10 | 306A |
| EFIS002 ✓ | UEFI Development and Innovations for System-On-Chip (SoC) | Tuesday 14:05 | 306A |
| EFIS003 ✓ | UEFI and Transparent Computing Technology | Tuesday 15:10 | 306A |
| EFIS004 ✓ | Intel® UEFI Development Kit 2010 and Intel® Boot Loader Development Kit: Foundations for Advanced Embedded Development | Tuesday 16:10 | 306A |
| SPCQ001 ✓ | Hot Topic Q&A: Intel® Boot Loader Development Kit (Intel® BLDK) | Tuesday 17:00 | 306A |
| EFIS005 ✓ | Security and Networking Advancements Today's UEFI and Intel® UEFI Development Kit 2010 (Intel® UDK2010) | Wednesday 11:10 | 306A |

✓ = DONE

# UEFI User Identification

- Technology includes
  - Uses UEFI Human Interface Infrastructure (HII) to display information to the user
  - Introduces optional policy controls for connecting to devices, loading images and accessing setup pages.





User Manager

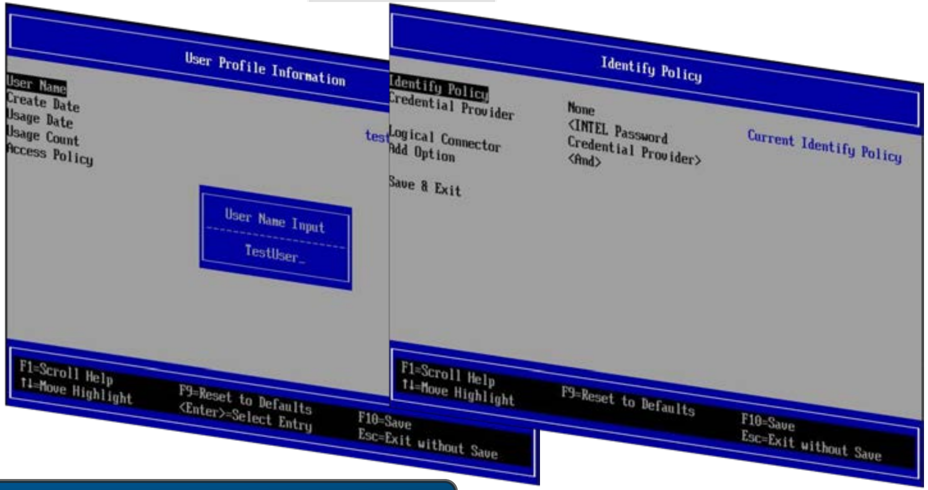
Add User Profile
Modify User Profile
Delete User Profile

Add User Profile to User Database

User Name Input

testuser_

F1=Scroll Help
↑↓=Move Highlight
F9=Reset to Defaults
<Enter>=Select Entry
F10=Save
Esc=Exit without Save



User Profile Information

User Name
Create Date
Usage Date
Usage Count
Access Policy

test

User Name Input

TestUser_

F1=Scroll Help
↑↓=Move Highlight
F9=Reset to Defaults
<Enter>=Select Entry
F10=Save
Esc=Exit without Save

Identify Policy

Identify Policy
Credential Provider
Logical Connector
Add Option

Save & Exit

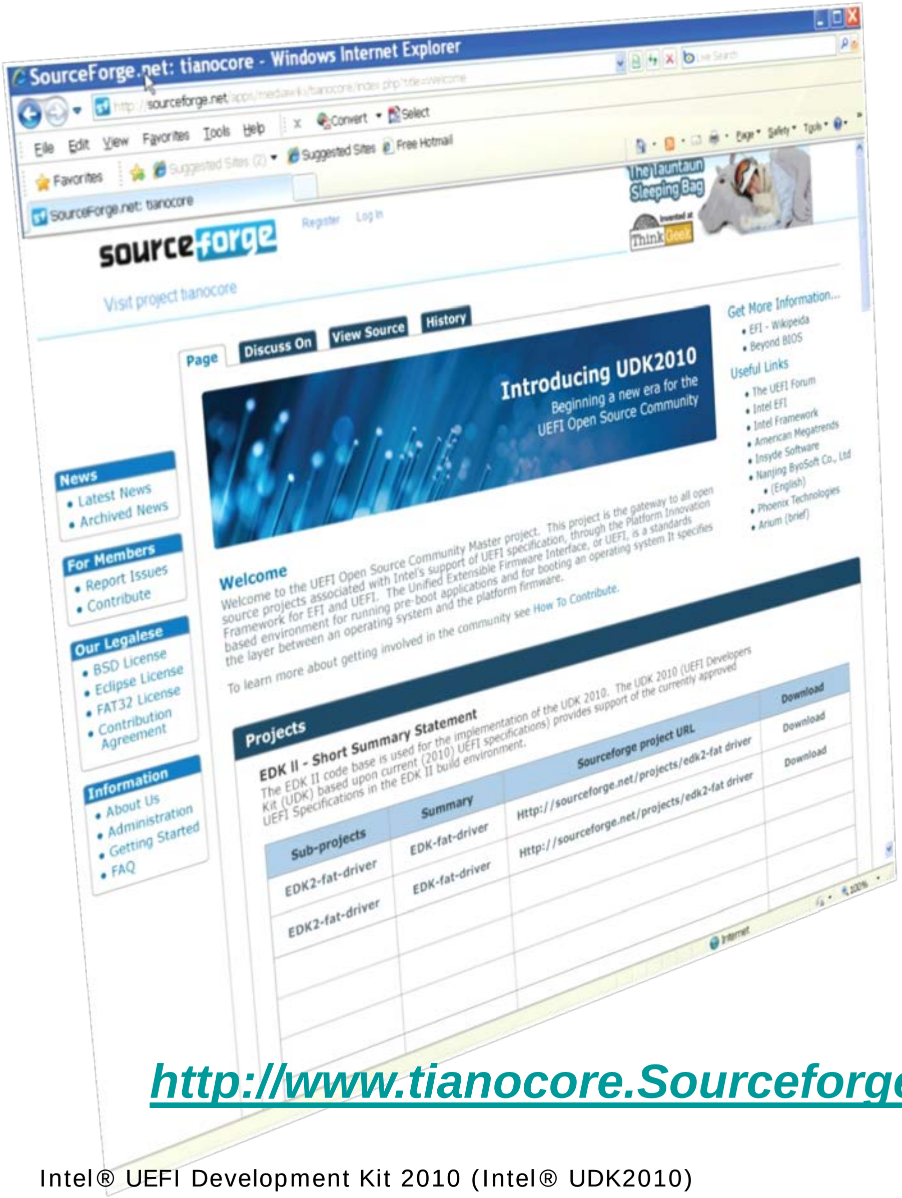
None
<INTEL Password
Credential Provider>
<And>

Current Identify Policy

F1=Scroll Help
↑↓=Move Highlight
F9=Reset to Defaults
F10=Save
Esc=Exit without Save

***Support for various pre-boot authenticators***

# Intel® UDK2010 Available on tianocore.org



## Introducing UDK2010

Beginning a new era for the UEFI Open Source Community

**News**

- Latest News
- Archived News

**For Members**

- Report Issues
- Contribute

**Our Legalese**

- BSD License
- Eclipse License
- FAT32 License
- Contribution Agreement

**Information**

- About Us
- Administration
- Getting Started
- FAQ

**Get More Information...**

- EFI - Wikipeida
- Beyond BIOS

**Useful Links**

- The UEFI Forum
- Intel EFI
- Intel Framework
- American Megatrends
- Insyde Software
- Nanjing ByoSoft Co., Ltd
  - (English)
- Phoenix Technologies
- Arium (brief)

### Welcome

Welcome to the UEFI Open Source Community Master project. This project is the gateway to all open source projects associated with Intel's support of UEFI specification, through the Platform Innovation Framework for EFI and UEFI. The Unified Extensible Firmware Interface, or UEFI, is a standards based environment for running pre-boot applications and for booting an operating system It specifies the layer between an operating system and the platform firmware.

To learn more about getting involved in the community see How To Contribute.

### Projects

**EDK II - Short Summary Statement**

The EDK II code base is used for the 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 EDK II build environment.

| Sub-projects | Summary | 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 |

*tianocore.org*

Intel® UDK2010
*Open Source*
UEFI Development Kit

***Develop. Contribute. Advance.***

*http://www.tianocore.Sourceforge.net*