IDF16
INTEL DEVELOPER FORUM

Accelerating Firmware Development With UEFI Advanced Features

Dong Wei
VP and Fellow, Hewlett Packard Enterprise

Ting Ye
UEFI Firmware Architect, Intel Corporation

STTS001

intel
experience what's inside™

Agenda

- Latest UEFI & ACPI Specifications
- Redfish RESTful Use Case in Data Center
- Apply Key Features to UEFI Development
- Summary

Agenda

- Latest UEFI & ACPI Specifications
- Redfish RESTful Use Case in Data Center
- Apply Key Features to UEFI Development
- Summary

Latest UEFI & ACPI Specifications

- UEFI 2.6
- ACPI 6.1
- UEFI Shell 2.2
- UEFI PI 1.4
- UEFI PI Packaging 1.1

http://uefi.org/specifications

UEFI 2.5 Networking

- Boot from HTTP(S) (HTTP API, HTTP Helper API, DNS v4/v6, RAMDISK, …)
- Wi-Fi (EAP, Extensible Authentication Protocol, Support)
- TLS, Transport Layer Security
- Bluetooth®
- Redfish REST Protocol

www.uefi.org

What’s New – UEFI 2.6

Network Enhancements

- Wireless MAC Connection II Protocol
- RAMDISK Protocol

RAS

- Common Platform Error Record (CPER) Extension for ARM*

User Interface

- Human Interface Infrastructure (HII) Font Ex, Glyph Generator, Image Ex and Image Generator Protocols

I/O

- SD/eMMC Pass Thru Protocol
- Non-identity Mapped Address Translations in PCI Root Bridge and I/O Protocols

www.uefi.org

What’s New – ACPI 6.1

Persistent Memory

- NFIT Updates
- NFIT Root Device _DSM

RAS

- APEI Extension for ARM*
- ERST/EINJ max wait time

Management

- Graceful Shutdown Clarifications
- Wireless Power Calibration Device

I/O

- Interrupt-signaled Events

UEFI & ACPI specification updates help in accelerating firmware development

Agenda

- Latest UEFI & ACPI Specifications
- Redfish RESTful Use Case in Data Center
- Apply Key Features to UEFI Development
- Summary

Redfish RESTful Use Case in Data Center

What is Redfish?

- Industry standard - www.dmtf.org/standards/redfish
- DMTF* Scalable Platforms Management Forum (SPMF) provides specification, schema, mockup, whitepaper, FAQ & resource browser

Managing multi-code servers via a RESTful API

- Built on modern tool chain (HTTPS, JSON, OData)

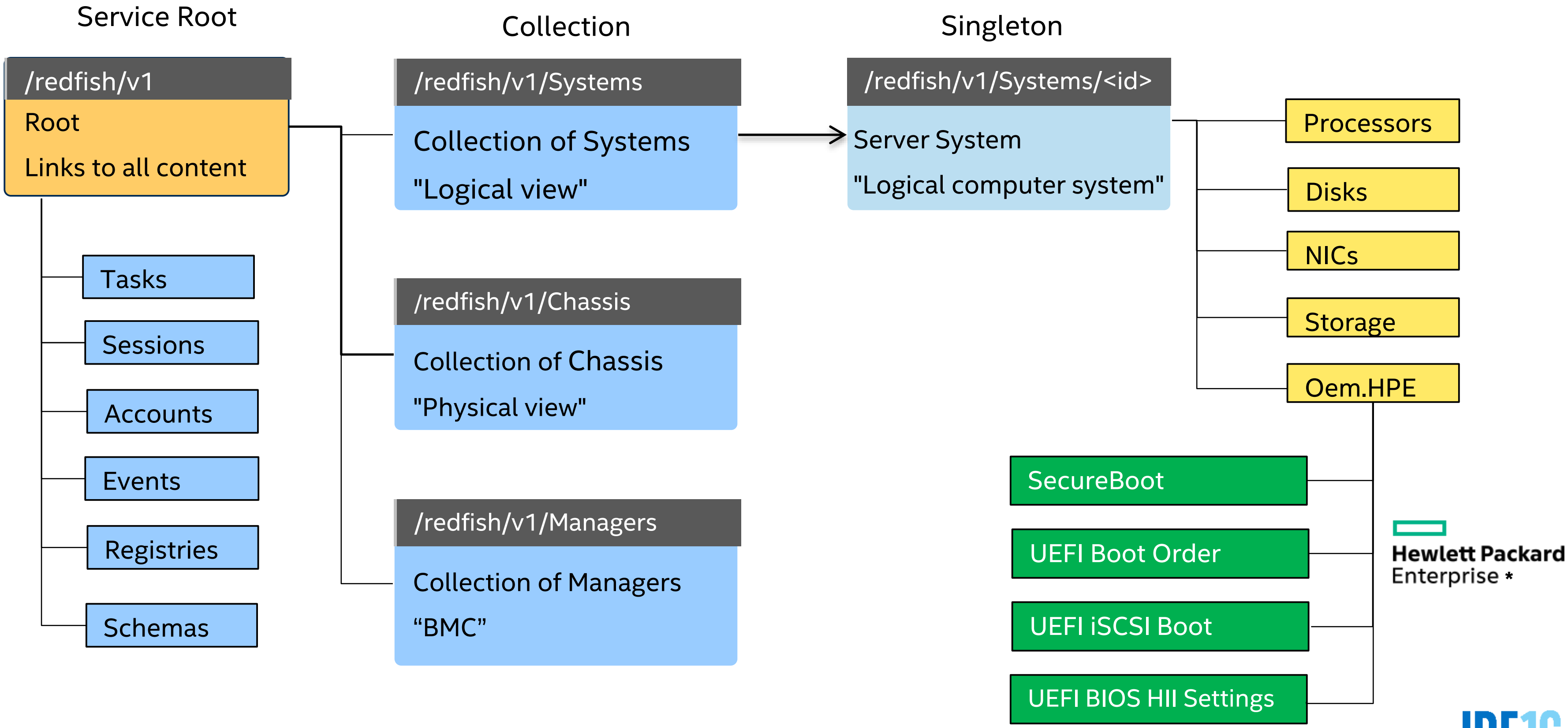
Client Python* code

```
rawData = urllib.urlopen(‘https://192.168.0.1/redfish/v1/Systems/1’)
jsonData = json.loads(rawData)
print( jsonData[‘SerialNumber’] )
```

Output

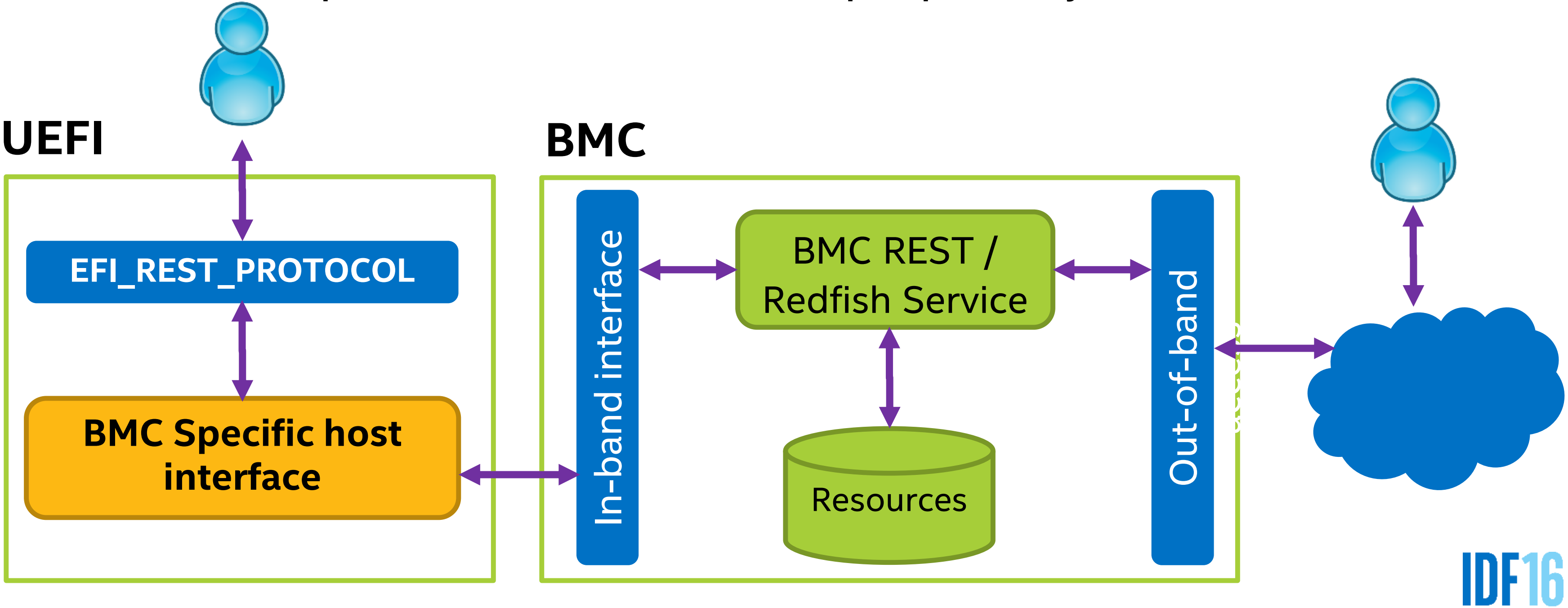
```
0AB8012GQ0
```

Redfish Resource Map

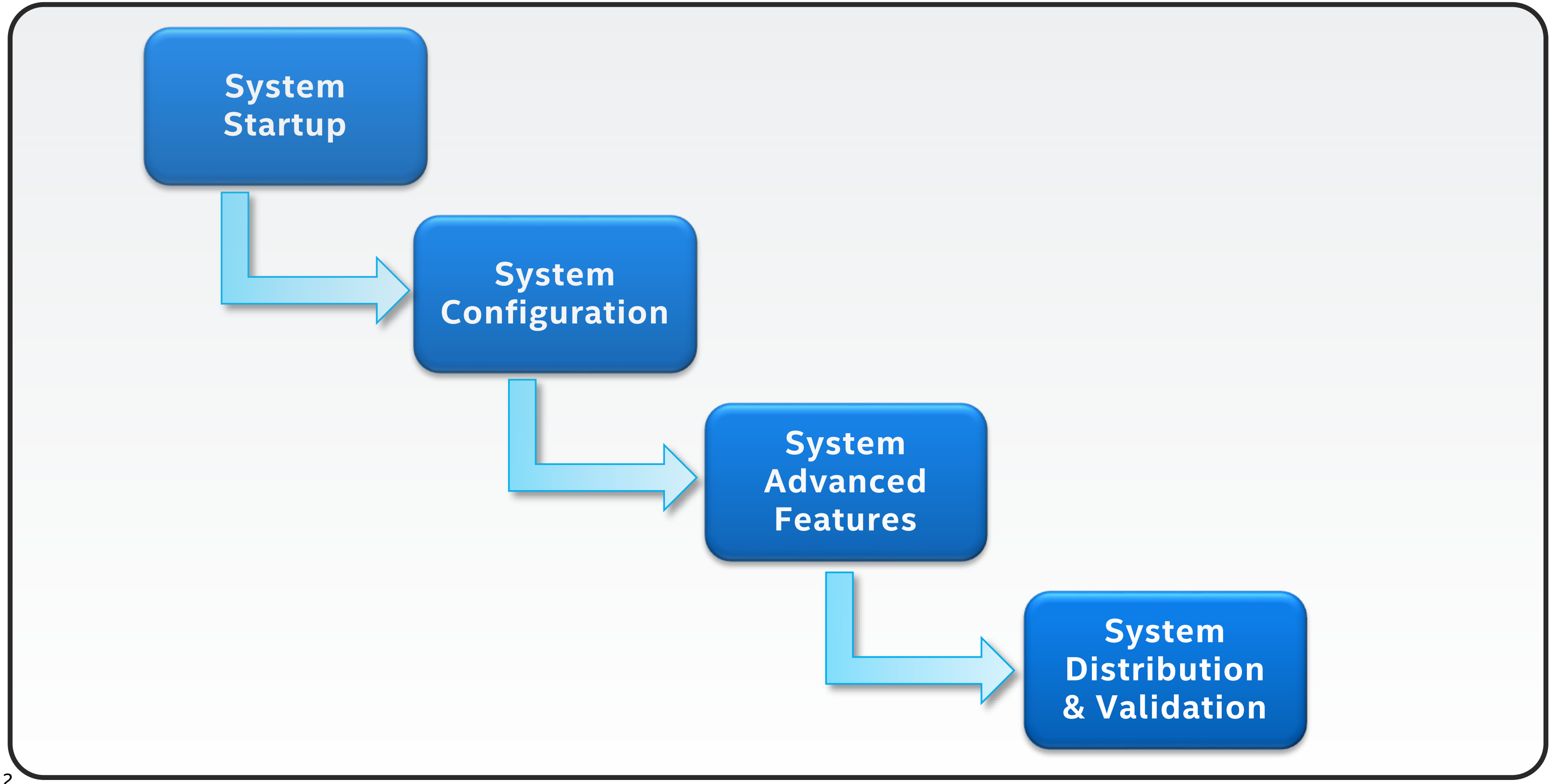


UEFI REST Protocol

- New in UEFI v2.5
- Standard pre-boot in-band access to a RESTful API, like Redfish
- Abstracts BMC-specific access methods (proprietary)



UEFI Firmware Development Process

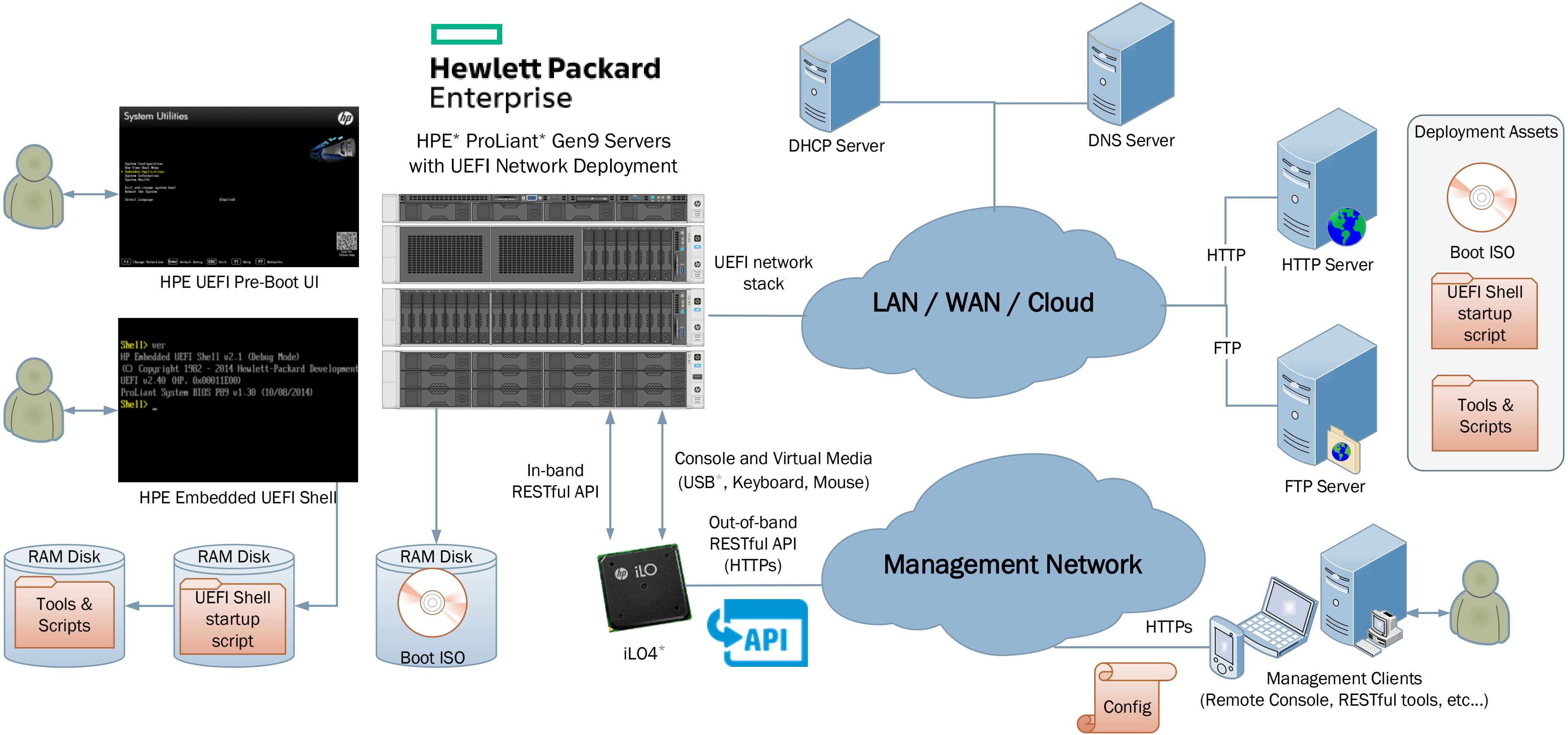

System Startup

System Configuration

System Advanced Features

System Distribution & Validation

UEFI Deployment Solution on HPE* Servers

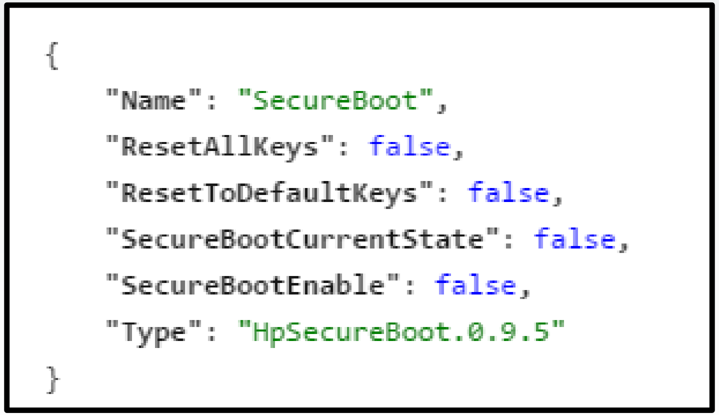


```
Shell> ver
HP Embedded UEFI Shell v2.1 (Debug Mode)
(C) Copyright 1982 - 2014 Hewlett-Packard Development
UEFI v2.40 (HP, 0x00011E00)
ProLiant System BIOS P89 v1.30 (10/08/2014)
Shell> _
```

Hewlett Packard Enterprise* Redfish Example: Secure Boot

GET @ /redfish/v1/systems/1/secureboot

- Enable/Disable Secure Boot
- Reset all Secure Boot variables to defaults
- Clear all keys (Setup Mode)

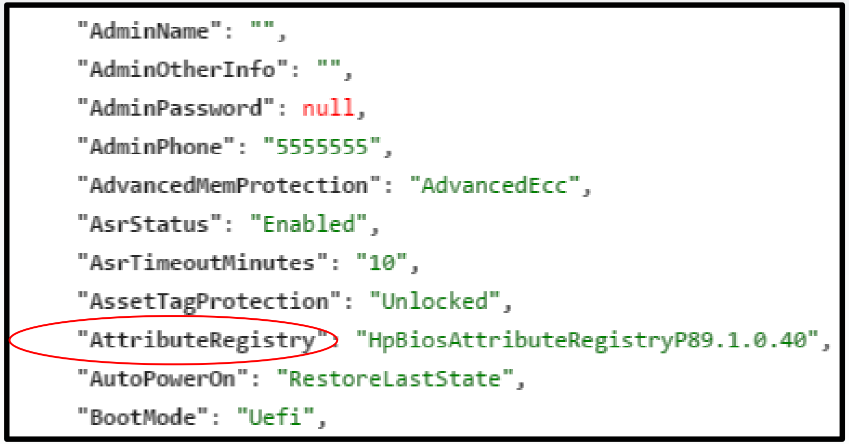


```
{
    "Name": "SecureBoot",
    "ResetAllKeys": false,
    "ResetToDefaultKeys": false,
    "SecureBootCurrentState": false,
    "SecureBootEnable": false,
    "Type": "HpSecureBoot.0.9.5"
}
```

Hewlett Packard Enterprise* Redfish Example: UEFI BIOS HII Settings

GET @ /redfish/v1/systems/1/bios

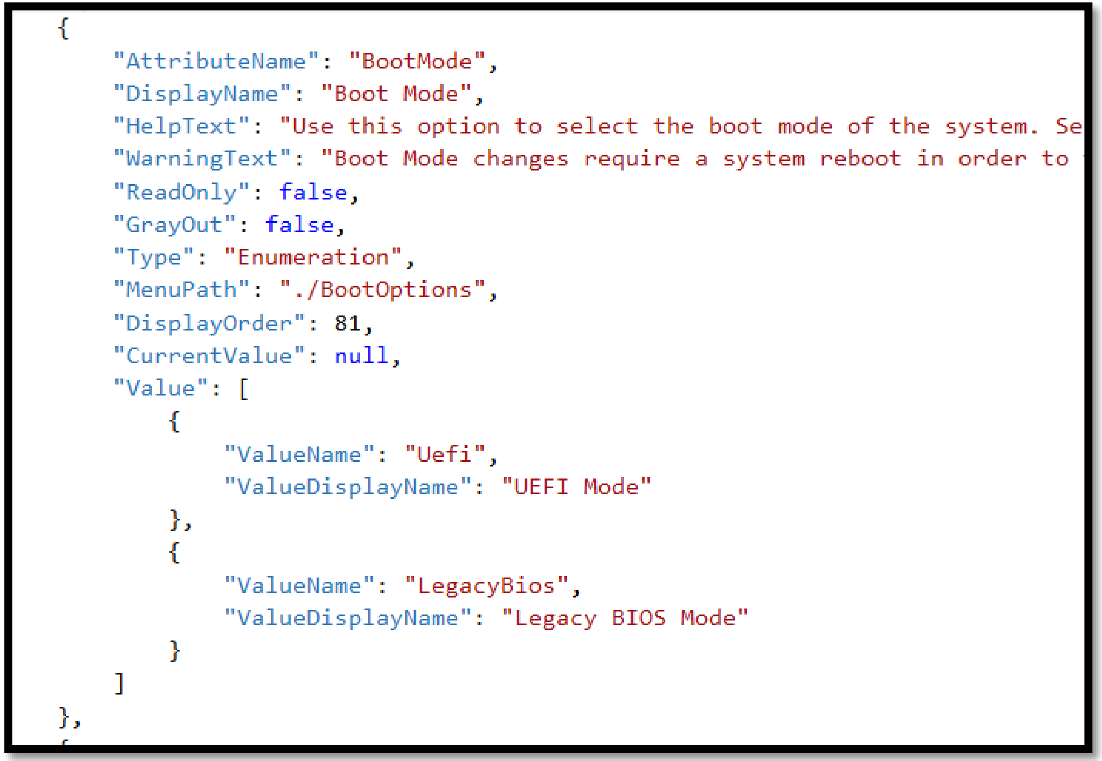
- All UEFI BIOS settings HII (name/values)
- HII meta-data in Attribute Registry
- Name/value pairs used to lookup meta-data in Attribute Registry



```
"AdminName": "",
"AdminOtherInfo": "",
"AdminPassword": null,
"AdminPhone": "5555555",
"AdvancedMemProtection": "AdvancedEcc",
"AsrStatus": "Enabled",
"AsrTimeoutMinutes": "10",
"AssetTagProtection": "Unlocked",
"AttributeRegistry": "HpBiosAttributeRegistryP89.1.0.40",
"AutoPowerOn": "RestoreLastState",
"BootMode": "Uefi",
```

Hewlett Packard Enterprise* Redfish Example: UEFI BIOS HII Settings

GET @ /redfish/v1/registries/HpBiosAttributeRegistryP89.1.0.40



```
{
    "AttributeName": "BootMode",
    "DisplayName": "Boot Mode",
    "HelpText": "Use this option to select the boot mode of the system. Se
    "WarningText": "Boot Mode changes require a system reboot in order to
    "ReadOnly": false,
    "GrayOut": false,
    "Type": "Enumeration",
    "MenuPath": "./BootOptions",
    "DisplayOrder": 81,
    "CurrentValue": null,
    "Value": [
        {
            "ValueName": "Uefi",
            "ValueDisplayName": "UEFI Mode"
        },
        {
            "ValueName": "LegacyBios",
            "ValueDisplayName": "Legacy BIOS Mode"
        }
    ]
},
```

Sample UEFI Shell Deployment Script (startup)

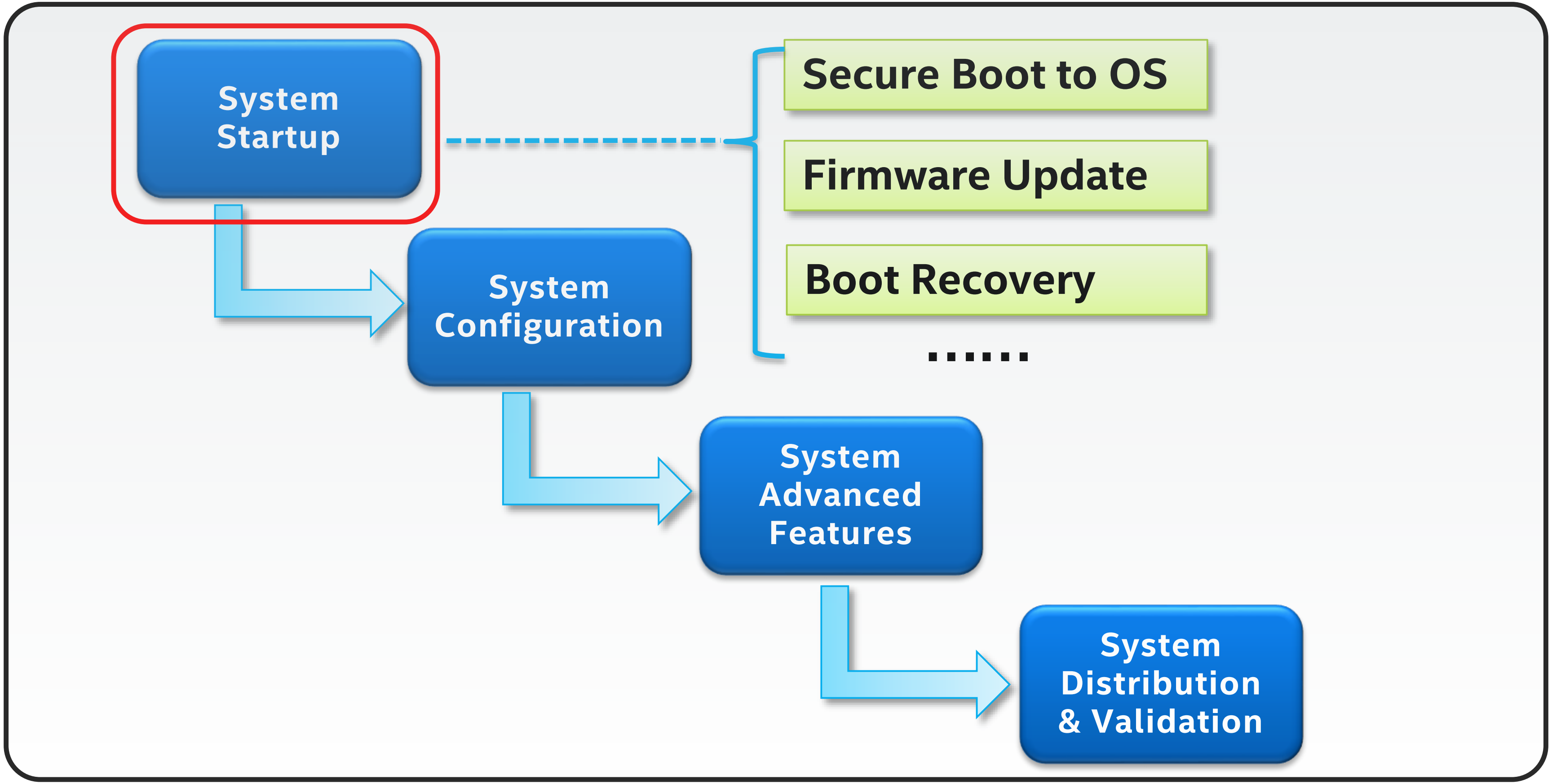
```
# Create FAT32 RAM Disk
ramdisk -c -s 512 -v MYRAMDISK -t F32
FS0:


# Download provisioning OS files from HTTP to RAM Disk
webclient -g http://repo.hpe.com/deploy/efilinux.efi
webclient -g http://repo.hpe.com/deploy/deploy.kernel
webclient -g http://repo.hpe.com/deploy/deploy.ramdisk


# Start provisioning OS
efilinux.efi -f deploy.kernel initrd=deploy.ramdisk
```

A use case of accelerating firmware development with UEFI advanced features

Agenda

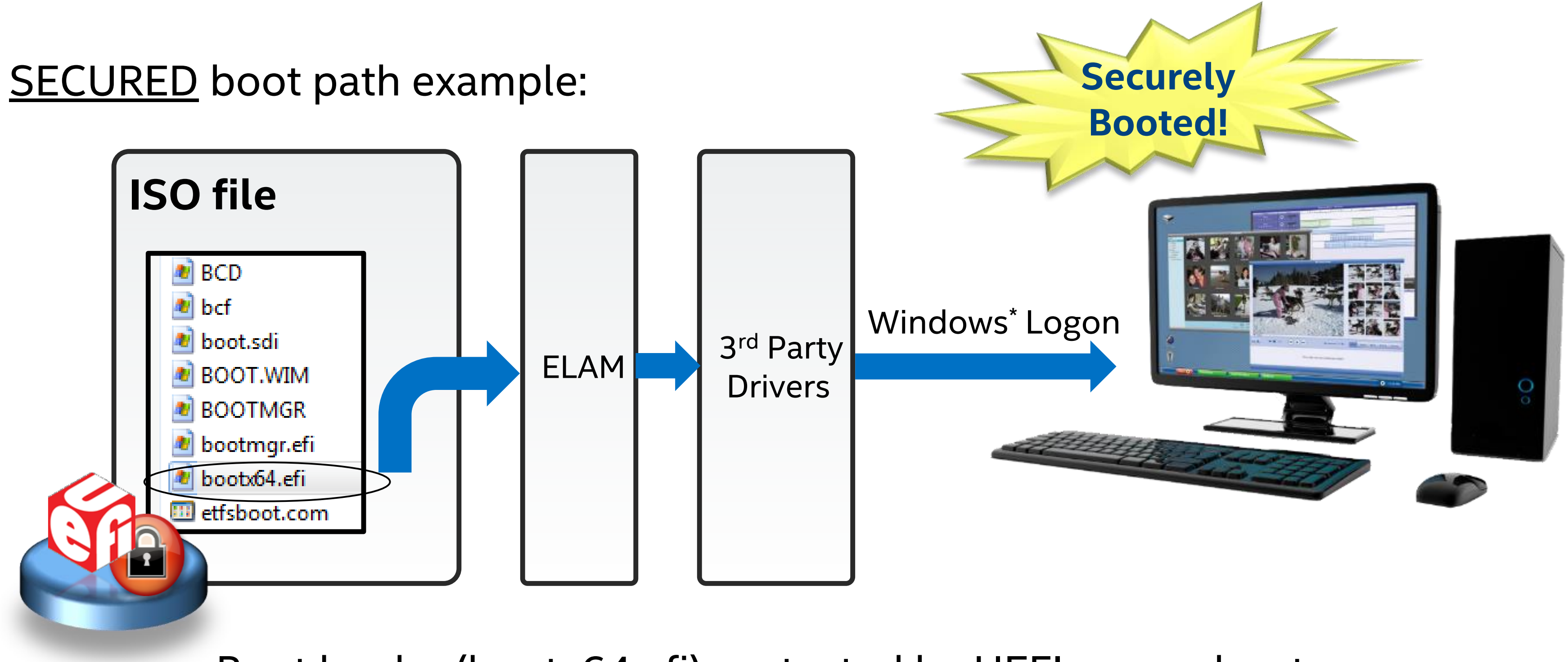
- Latest UEFI & ACPI Specifications
- Redfish RESTful Use Case in Data Center
- Apply Key Features to UEFI Development
- Summary

Apply Key Features to UEFI Development



Initial - UEFI Secure Boot

SECURED boot path example:



Securely Booted!

ISO file

- BCD
- bcf
- boot.sdi
- BOOT.WIM
- BOOTMGR
- bootmgr.efi
- bootx64.efi
- etfsboot.com

ELAM

3rd Party Drivers

Windows* Logon

- Boot loader (bootx64.efi) protected by UEFI secure boot
- Early Launch Anti-Malware (ELAM) protected by Boot loader
- Rootkit malware can no longer bypass anti-malware inspection

IDF16 INTEL DEVELOPER FORUM

Advanced – Customized UEFI Secure Boot

Deployment

Initial	Advanced
Platform Specific PK$_{pub}$ Clear	Standardized solution to customize the secure boot keys
Setup Mode User Mode	Setup Mode User Mode Audit Mode Deployed Mode

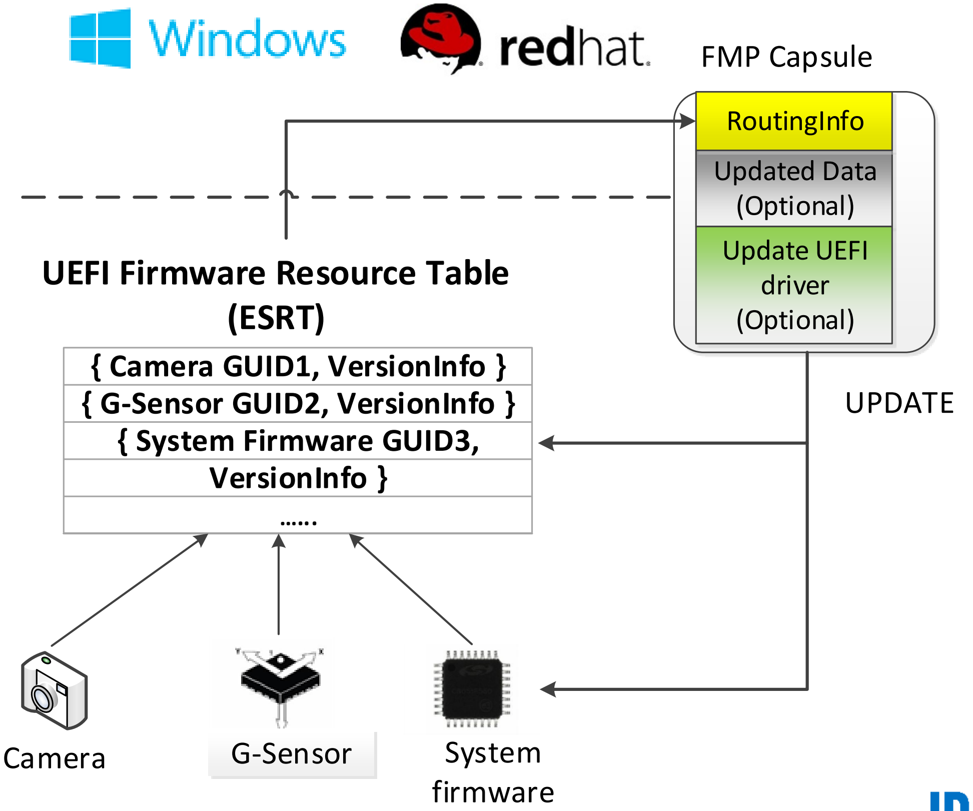
Benefits

• No specific solution	Security
• Higher utilization	Flexibility
• Verification status	Extensibility

Customized UEFI Secure Boot reduces the security risk introduced by platform specific solutions. Working w/ OS vendors on interoperability and readiness.

Secure Firmware Update

- Firmware update protected by:
  - OS verify the update driver when creating capsule
  - UEFI secure boot verify capsule payload before performing update
- What's new:
  - ESRT
  - FMPv3
  - FMP capsule



UEFI Firmware Resource Table (ESRT)

{ Camera GUID1, VersionInfo }

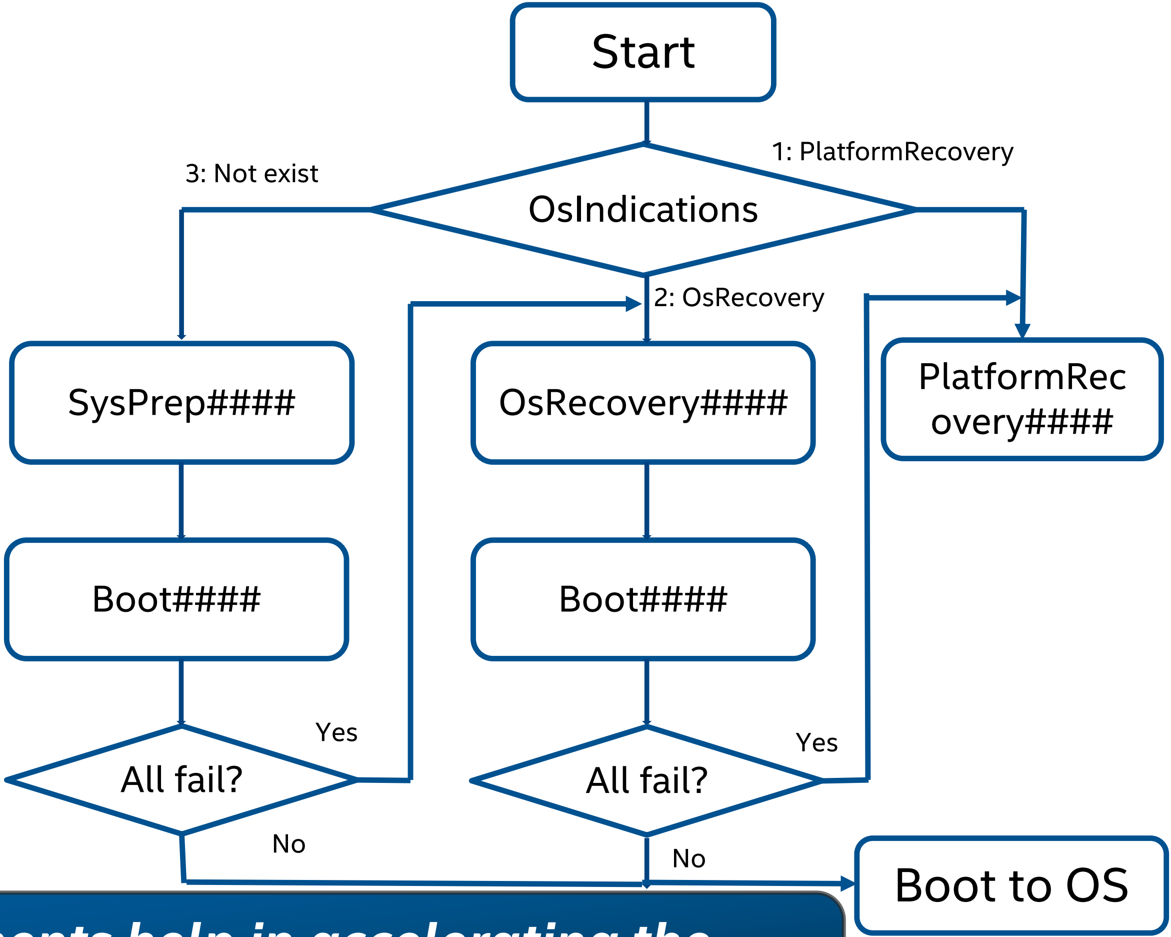
{ G-Sensor GUID2, VersionInfo }

{ System Firmware GUID3, VersionInfo }

……

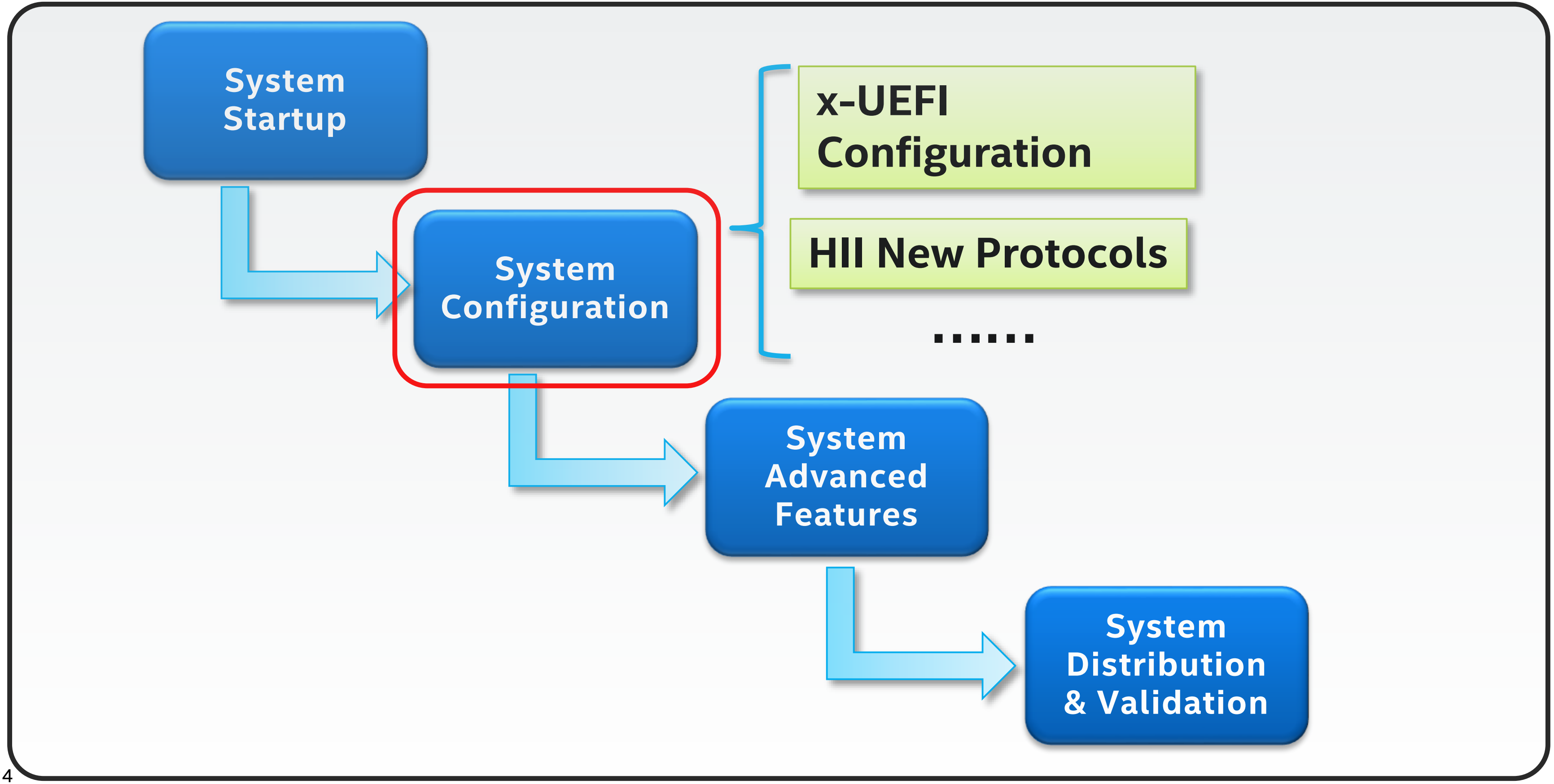
Boot Recovery

- What's new
  - OS defined recovery
  - Platform defined recovery
  - Recovery policy protected by authentication
    - OsRecoveryOrder
    - dbrDefault, dbr
  - Default platform recovery supported



Security enhancements help in accelerating the system startup stage

Apply Key Features to UEFI Development



System Startup

System Configuration

x-UEFI Configuration

HII New Protocols

......

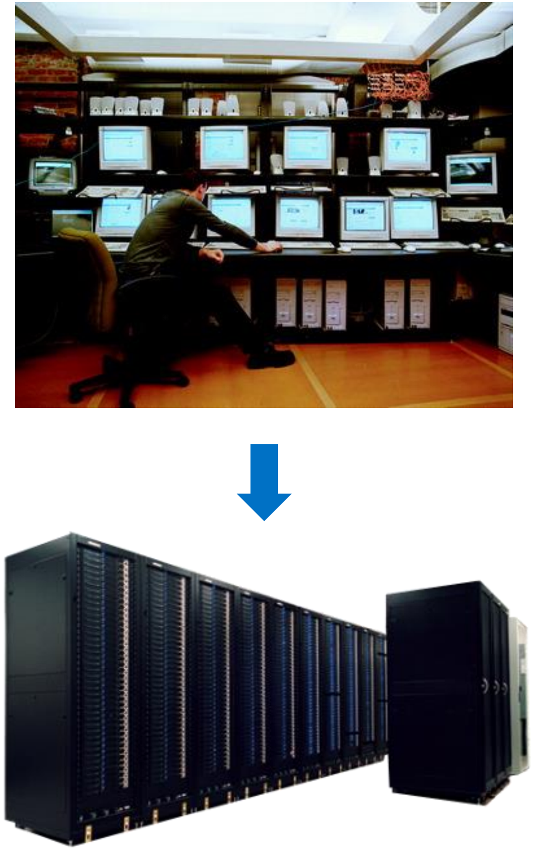
System Advanced Features

System Distribution & Validation

x-UEFI Scriptable Configuration

- Based on keywords defined in different namespaces
- Leverages existing UEFI HII infrastructure

- Key elements:
  - x-UEFI language
  - Keyword Handler Protocol



x-UEFI Usage Example

```
iSCSIInitiatorName
```

VFR file

```
string  varid    = ISCSI_CONFIG_IFR_NVDATA.InitiatorName,
            prompt   = STRING_TOKEN(STR_ISCSI_CONFIG_INIT_NAME),
```

UNI file

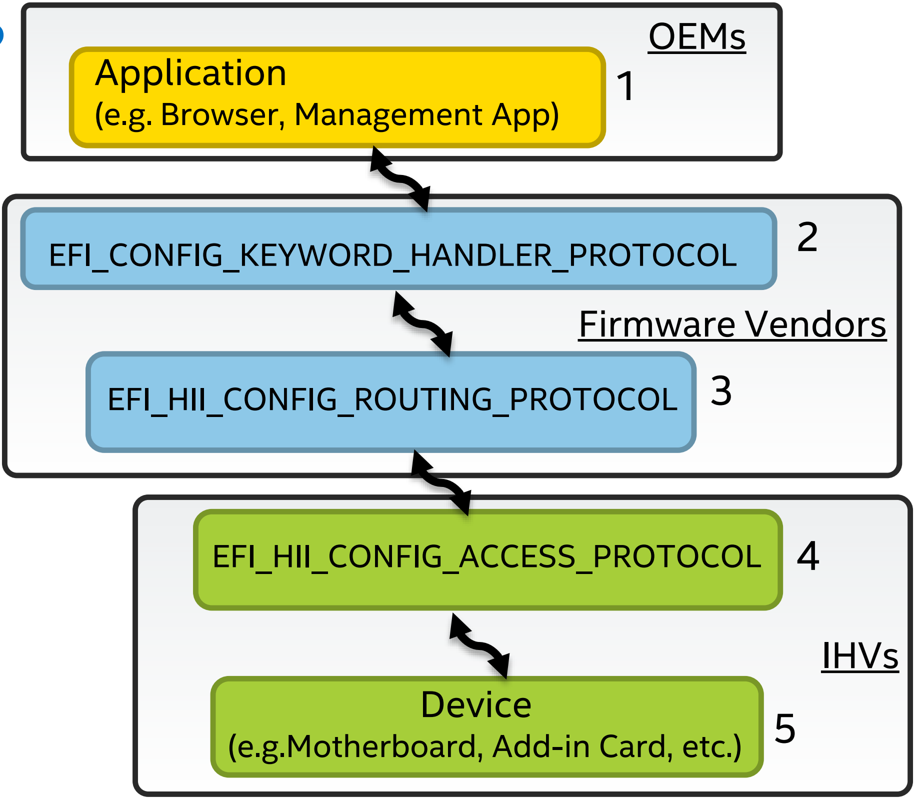
```
#string STR_ISCSI_CONFIG_INIT_NAME   #language en-US "iSCSI Initiator Name"
#string STR_ISCSI_CONFIG_INIT_NAME   #language x-UEFI "iSCSIInitiatorName"
```

Script file

```
IScsiScript -i iqn.edkii.intel.com
```

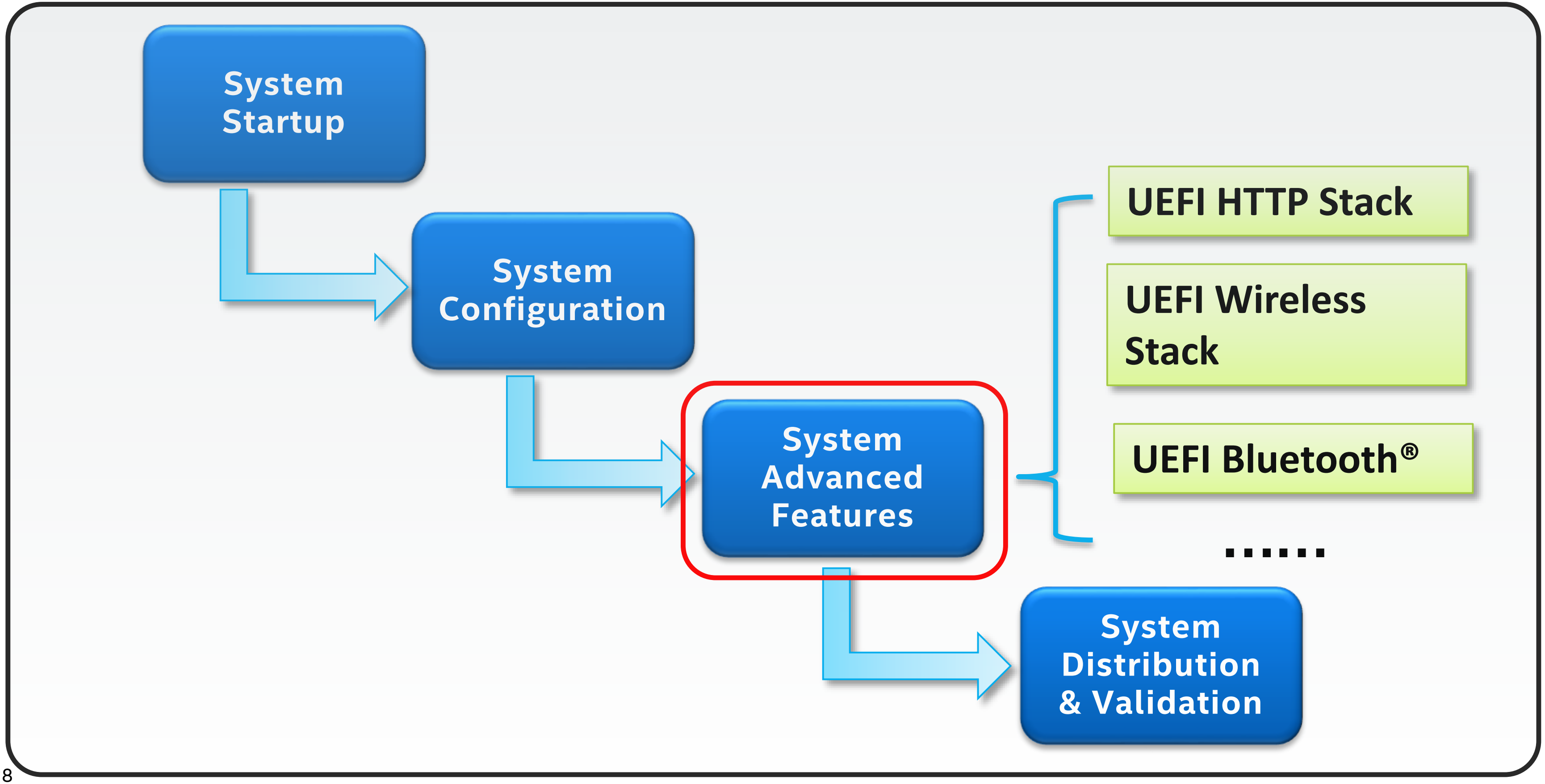
How to Implement x-UEFI?

- OEMs …
  - Get keywords definition from http://www.uefi.org/confignamespace
  - Use KeywordHandler.GetData/SetData
- Firmware vendors …
  - Get HII updates from Intel® UEFI Development Kit (Intel® UDK) 2015
- IHVs …
  - Define and register x-UEFI keywords
  - Support keyword setting in ConfigAccess.RouteConfig


OEMs
Application
(e.g. Browser, Management App)
1
EFI_CONFIG_KEYWORD_HANDLER_PROTOCOL
2
Firmware Vendors
EFI_HII_CONFIG_ROUTING_PROTOCOL
3
EFI_HII_CONFIG_ACCESS_PROTOCOL
4
IHVs
Device
(e.g.Motherboard, Add-in Card, etc.)
5

Configuration enhancements help in accelerating the in-band startup during the system configuration stage

Apply Key Features to UEFI Development



System Startup

System Configuration

System Advanced Features

System Distribution & Validation

UEFI HTTP Stack

UEFI Wireless Stack

UEFI Bluetooth®

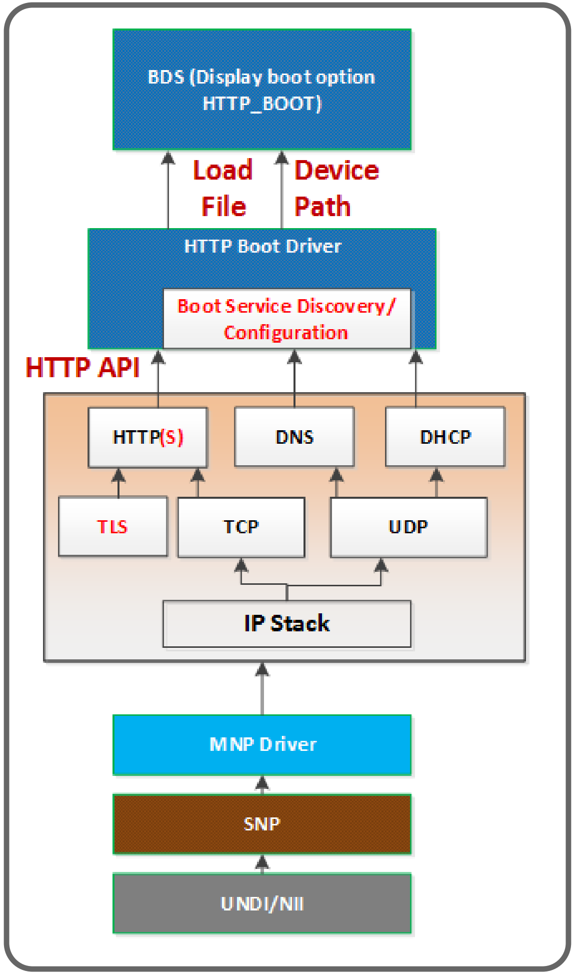
……

UEFI HTTP Stack

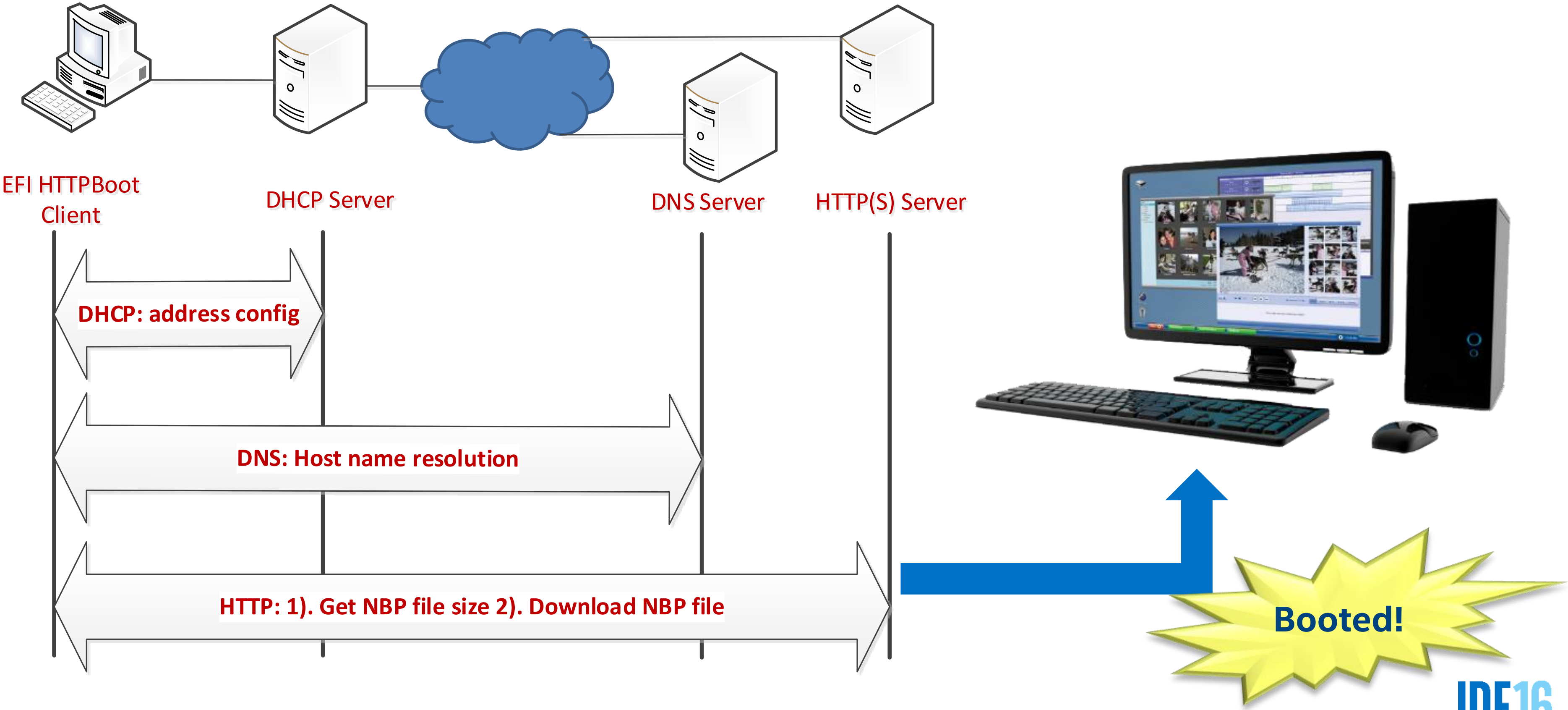
New Modules

Driver	Library
HTTP Boot Driver HTTP Driver HTTP Utilities Driver TLS Driver	HTTP Library TlsLib Library OpenslTlsLib Library

- Flexible Network Deployment
- Home Environment Support
- Corporate Environment Support



HTTP(S) Boot Flow



EFI HTTPBoot Client

DHCP Server

DNS Server

HTTP(S) Server

DHCP: address config

DNS: Host name resolution

HTTP: 1). Get NBP file size 2). Download NBP file

Booted!

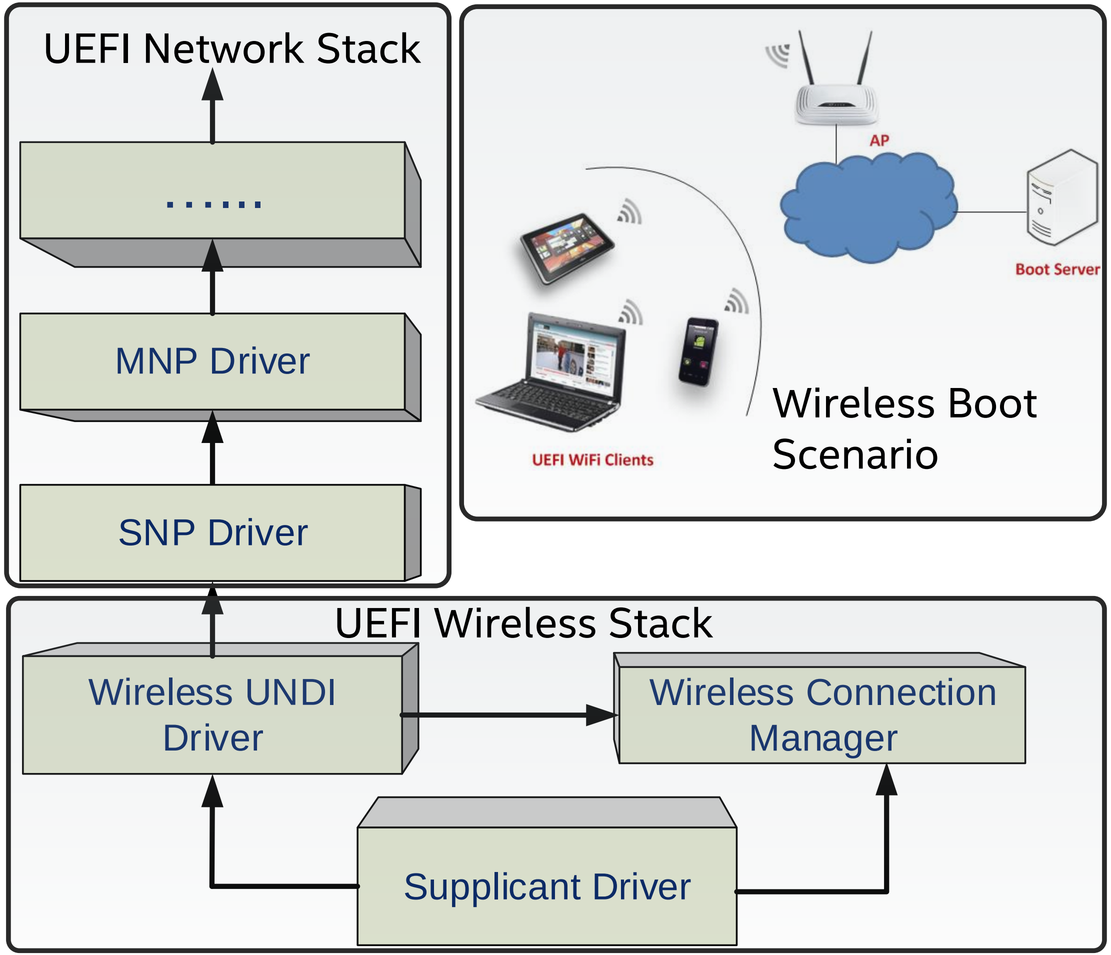
DEMO – UEFI HTTP(S) Boot

- STEP 0: Configure TLS certificate
  - For HTTPS
- STEP 1: Configure Boot URI
  - Enter Device Manager
  - Select a particular NIC
  - Enter HTTP boot Configuration
  - Enter Boot URI and save changes
- STEP 2: Find boot option
  - Enter Boot Manager
  - Select new added boot option
- STEP 3: Boot to Windows* Pre-installation Environment image

TLS is still a patch. Cert management is not secured so far. Want to work w/ the community to harden and OS vendors for interoperability.

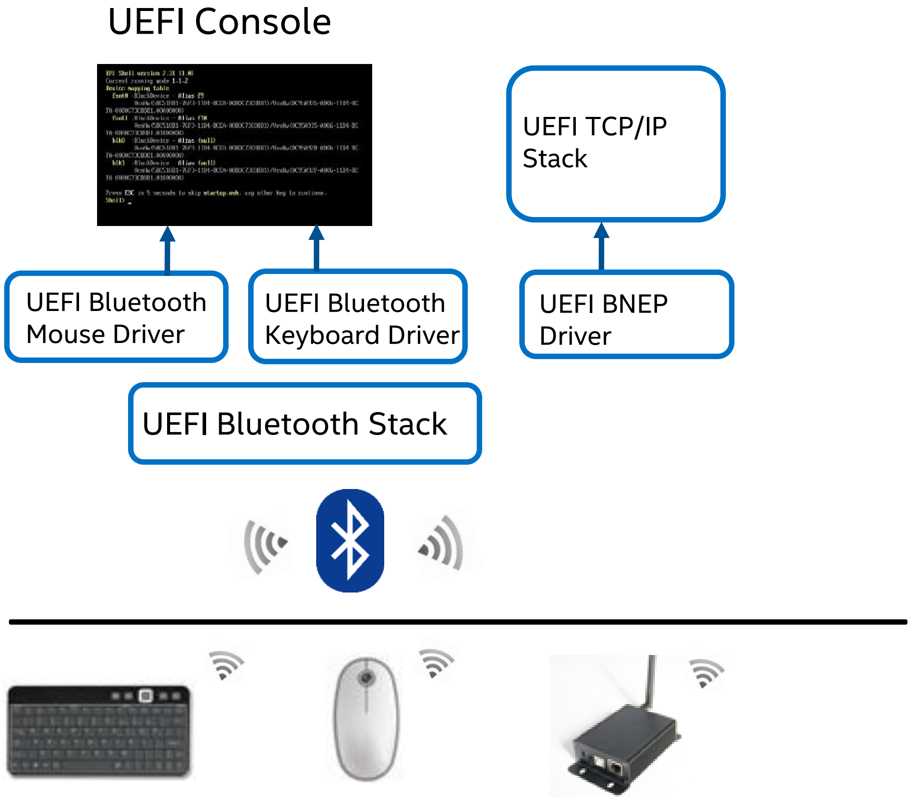
UEFI Wireless Stack

- 802.11 compliant wireless stack:
  - Connection manager using HII
  - Generic supplicant capability includes
    - PSK authentication
    - EAP 802.1x authentication
  - CCMP encryption



UEFI Bluetooth®

- Produce generic I/O interface:
  - UEFI device drivers can easily deliver rich services
- UEFI Bluetooth® Stack Layer:
  - Bluetooth® host controller
  - Bluetooth® bus
  - Bluetooth® service

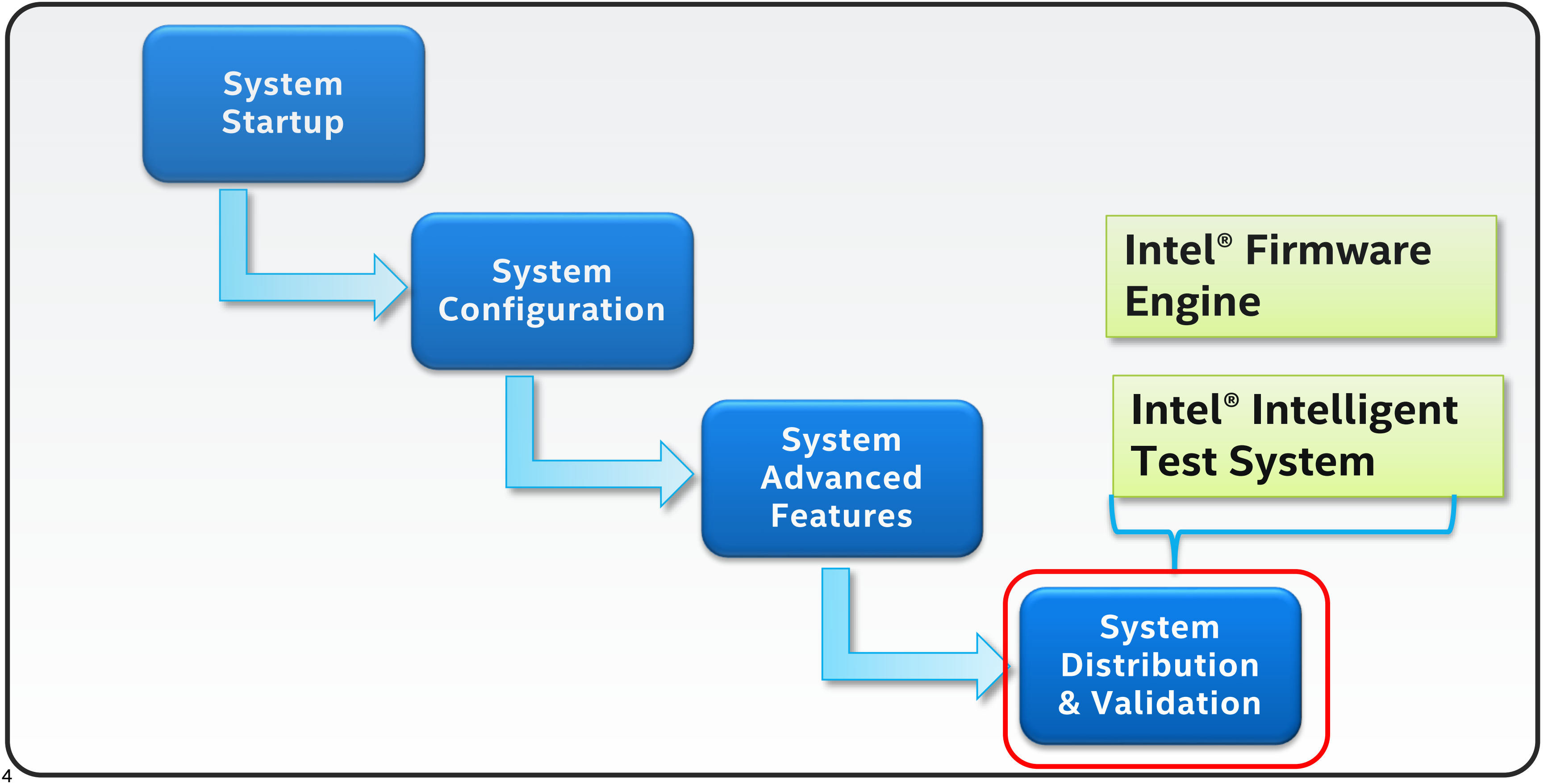


UEFI Console

```
EFI Shell version 2.31 [1.0]
Current running mode 1.1.2
Device mapping table
  fsnt0 :BlockDevice - Alias f9
          VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)/VenHw(0C95A935-A006-11D4-BC
FA-0080C73C8881,00000000)
  fsnt1 :BlockDevice - Alias f10
          VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)/VenHw(0C95A935-A006-11D4-BC
FA-0080C73C8881,01000000)
  blk0  :BlockDevice - Alias (null)
          VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)/VenHw(0C95A928-A006-11D4-BC
FA-0080C73C8881,00000000)
  blk1  :BlockDevice - Alias (null)
          VenHw(58C518B1-76F3-11D4-BCEA-0080C73C8881)/VenHw(0C95A92F-A006-11D4-BC
FA-0080C73C8881,01000000)

Press ESC in 5 seconds to skip startup.nsh, any other key to continue.
Shell> _
```

Leverage the connectivity enhancement during the stage of enabling advanced feature

Apply Key Features to UEFI Development



System Startup

System Configuration

System Advanced Features

System Distribution & Validation

Intel® Firmware Engine

Intel® Intelligent Test System

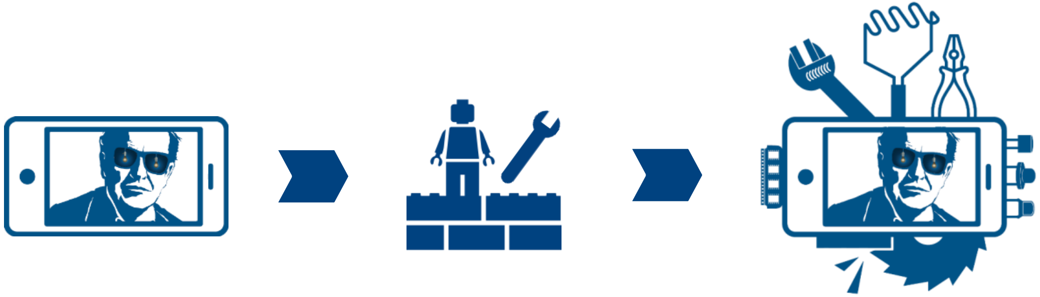
Intel® Firmware Engine

Quickly generate royalty-free firmware for IoT devices without source code

Extensible binary firmware framework

Start from validated reference designs

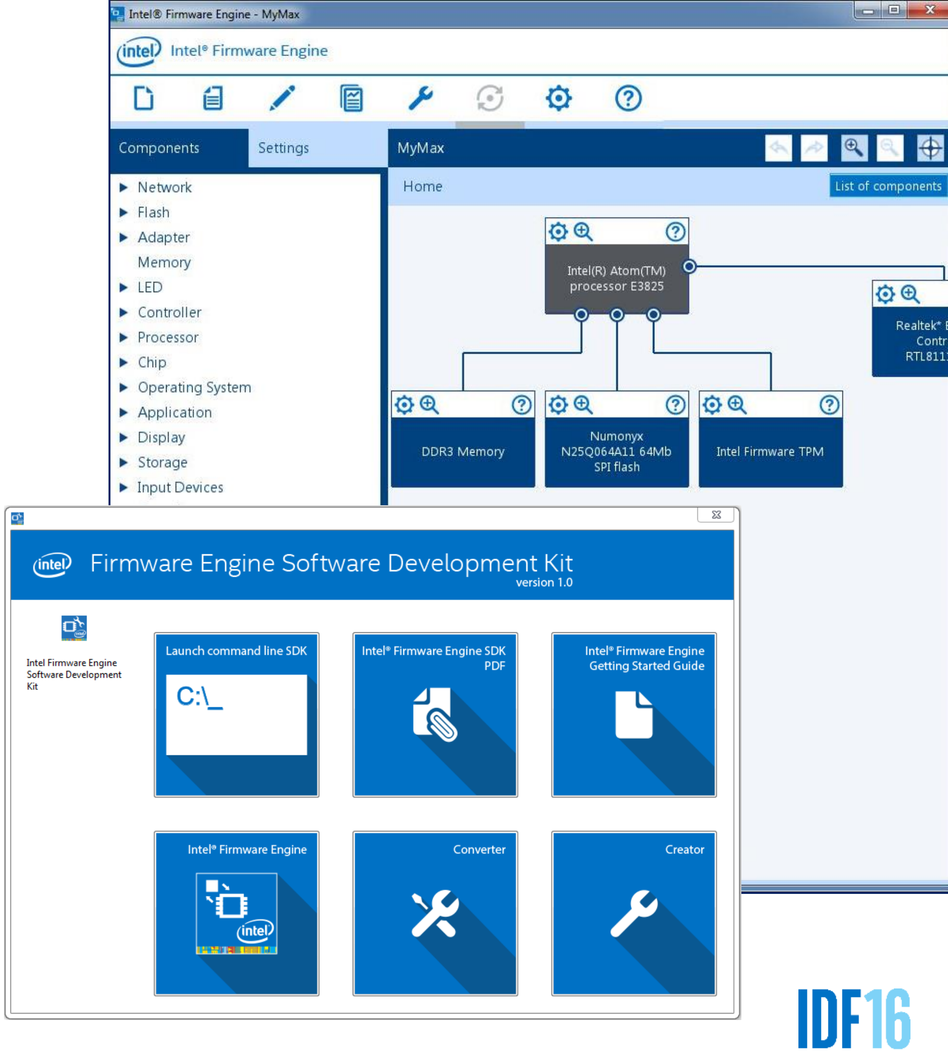
GUI development for faster time to market



Available now at intel.com/firmwareengine

Intel® Firmware Engine

- Application, SDK and open hardware platforms available for download at intel.com/firmwareengine
- Intel® Firmware Engine 2.0 just released
- Updated SDK due April 2016
- Additional platforms from the Intel IoT roadmap are under development



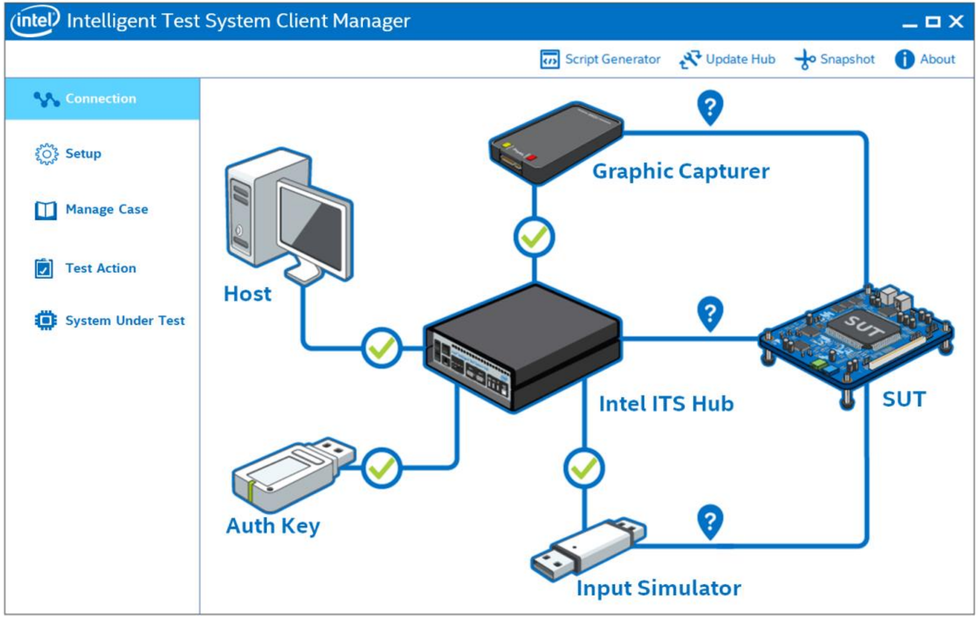
Intel® Intelligent Test System (Intel® ITS)

Scalable hardware/software test framework

Test automation, device control & UEFI code coverage

Reduce costs & improve validation efficiency

Intel® Firmware Engine and Intel® ITS simplify firmware distribution and validation



Available now at intel.com/intel-its

Agenda

- Latest UEFI & ACPI Specifications
- Redfish RESTful Use Case in Data Center
- Apply Key Features to UEFI Development
- Summary

Summary and Next Steps

- UEFI & ACPI specification updates help in accelerating firmware development
- Redfish used RESTful management in modern data center is a good use case of accelerating firmware development with UEFI advanced features
- More enhancements in security, configuration, networking are ready to be adopted
- Intel® Firmware Engine and Intel® Intelligent Test System simplify firmware distribution and validation

Next Steps:

- Adopt UEFI 2.6 implementations with UEFI advanced features
- Adopt Redfish implementations in servers and management software
- Working with the community more deeply to continue improving security, interoperability and readiness of UEFI advanced features

Additional Sources of Information

- A PDF of this presentation is available from our Technical Session Catalog: www.intel.com/idfsessionsSZ
- This URL is also printed on the top of Session Agenda Pages in the Pocket Guide.

- More web-based info:
  - Intel® Architecture Firmware Resource Center: firmware.intel.com
  - UEFI Forum Learning Center: uefi.org/learning_center
  - UEFI and ACPI Specifications: www.uefi.org/specs/
  - Redfish Specification: www.dmtf.org/standards/redfish

Intel EDK II & UEFI Developer Survey

Intel Software is conducting a survey to improve EDK II & UEFI development tools. We want to know about your compiler preferences, debug methods, and what we can do for the future of firmware.

http://intelcustomer.az1.qualtrics.com/SE/?SID=SV_6lJbxG5BYFFMPSl&Source=IDF

Other Technical Sessions

Session ID	Title	Day	Time	Room
✓ STTS001	Accelerating Firmware Development With UEFI Advanced Features	Wed	13:15	Auditorium
STTS002	Enhancing Real-time Communication User Experience on Internet with Intel® Collaboration Suite for WebRTC	Wed	14:30	Auditorium
STTS003	Planning and Predicting Big Data and IoT Solutions	Wed	15:45	Auditorium

✓ = DONE

WHAT WILL YOU DEVELOP?

Legal Notices and Disclaimers

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure.

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

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

Statements in this document that refer to Intel's plans and expectations for the quarter, the year, and the future, are forward-looking statements that involve a number of risks and uncertainties. A detailed discussion of the factors that could affect Intel's results and plans is included in Intel's SEC filings, including the annual report on Form 10-K.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

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

Intel does not control or audit third-party benchmark data or the web sites referenced in this document. You should visit the referenced web site and confirm whether referenced data are accurate.

Intel, and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

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

The Bluetooth® word mark and logos are registered trademarks owned by Bluetooth SIG, Inc. and any use of such marks by Intel is under license.

Risk Factors

The above statements and any others in this document that refer to future plans and expectations are forward-looking statements that involve a number of risks and uncertainties. Words such as "anticipates," "expects," "intends," "goals," "plans," "believes," "seeks," "estimates," "continues," "may," "will," "should," and variations of such words and similar expressions are intended to identify such forward-looking statements. Statements that refer to or are based on projections, uncertain events or assumptions also identify forward-looking statements. Many factors could affect Intel's actual results, and variances from Intel's current expectations regarding such factors could cause actual results to differ materially from those expressed in these forward-looking statements. Intel presently considers the following to be important factors that could cause actual results to differ materially from the company's expectations. Demand for Intel's products is highly variable and could differ from expectations due to factors including changes in business and economic conditions; consumer confidence or income levels; the introduction, availability and market acceptance of Intel's products, products used together with Intel products and competitors' products; competitive and pricing pressures, including actions taken by competitors; supply constraints and other disruptions affecting customers; changes in customer order patterns including order cancellations; and changes in the level of inventory at customers. Intel's gross margin percentage could vary significantly from expectations based on capacity utilization; variations in inventory valuation, including variations related to the timing of qualifying products for sale; changes in revenue levels; segment product mix; the timing and execution of the manufacturing ramp and associated costs; excess or obsolete inventory; changes in unit costs; defects or disruptions in the supply of materials or resources; and product manufacturing quality/yields. Variations in gross margin may also be caused by the timing of Intel product introductions and related expenses, including marketing expenses, and Intel's ability to respond quickly to technological developments and to introduce new products or incorporate new features into existing products, which may result in restructuring and asset impairment charges. Intel's results could be affected by adverse economic, social, political and physical/infrastructure conditions in countries where Intel, its customers or its suppliers operate, including military conflict and other security risks, natural disasters, infrastructure disruptions, health concerns and fluctuations in currency exchange rates. Results may also be affected by the formal or informal imposition by countries of new or revised export and/or import and doing-business regulations, which could be changed without prior notice. Intel operates in highly competitive industries and its operations have high costs that are either fixed or difficult to reduce in the short term. The amount, timing and execution of Intel's stock repurchase program could be affected by changes in Intel's priorities for the use of cash, such as operational spending, capital spending, acquisitions, and as a result of changes to Intel's cash flows or changes in tax laws. Product defects or errata (deviations from published specifications) may adversely impact our expenses, revenues and reputation. Intel's results could be affected by litigation or regulatory matters involving intellectual property, stockholder, consumer, antitrust, disclosure and other issues. An unfavorable ruling could include monetary damages or an injunction prohibiting Intel from manufacturing or selling one or more products, precluding particular business practices, impacting Intel's ability to design its products, or requiring other remedies such as compulsory licensing of intellectual property. Intel's results may be affected by the timing of closing of acquisitions, divestitures and other significant transactions. We completed our acquisition of Altera on December 28, 2015 and risks associated with that acquisition are described in the “Forward Looking Statements” paragraph of Intel’s press release dated June 1, 2015, which risk factors are incorporated by reference herein. A detailed discussion of these and other factors that could affect Intel's results is included in Intel's SEC filings, including the company's most recent reports on Form 10-Q, Form 10-K and earnings release.

Rev. 1/14/16

Backup

What’s New – UEFI Shell 2.2

- Network updates
- Allow **Execute()** to not nest new shells
- Add command line parameter to auto exit
- **setvar** command re-factor
- New command features

  `dh, disconnect, comp, dmem, cls, reset, pci, bcfg, dmpstore`

What’s New – PI Packaging 1.1

- Convey PCD settings with discrete sub-settings
- Localized name to a package
- Convey detailed Protocol/PPI/GUIDs produces information
- Convey usage for PCDs from binary modules
- Convey detailed Protocol/PPI/GUIDs consumes information
- Convey PCD display information
- Convey enumeration-like information for PCD (allow string)
- Abstract type support
- Convey detailed BY_START/TO_START interaction
- Convey install/produce limit information about Protocol/PPI/GUIDs

Sample Configuration Script Using HPREST Tool

```
# Login to iLO
hprest login https://clientilo.domain.com -u username -p password

# Configure UEFI network settings (Use Auto and DHCP defaults)
hprest set PreBootNetwork=Auto --selector HpBios.
hprest set Dhcpv4=Enabled

# Configure UEFI Shell startup script from URL
hprest set UefiShellStartup=Enabled
hprest set UefiShellStartupLocation=NetworkLocation
hprest set UefiShellStartupUrl=http://192.168.1.1/deploy/startup.nsh

# Set one-time-boot to Embedded UEFI Shell
hprest set Boot/BootSourceOverrideEnabled=Once --selector ComputerSystem.
hprest set Boot/BootSourceOverrideTarget=UefiShell

# Save and reboot server
hprest commit --reboot=ON
```