IDF15
INTEL DEVELOPER FORUM

Firmware in the Data Center: Building a Modern Deployment Framework Using Unified Extensible Firmware Interface (UEFI) and Redfish REST APIs

Mark Doran – Intel Fellow, Chief Platform Software Architect, Intel Corporation

Dong Wei – Fellow and VP, UEFI Forum, HP

Samer El-Haj-Mahmoud – Master Technologist, HP

STTS001

intel
experience what's inside™

Agenda

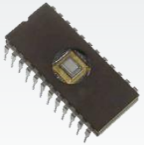
- Challenges of Firmware in the Data Center
- PXE and HTTP Boot
- UEFI Shell Scripting
- Data Center Manageability: Redfish and REST APIs
- Putting it all together: HP* ProLiant* Servers
- Summary and Q&A

Challenges of Firmware in the Data Center

Firmware Challenges in the Data Center

Bare Metal Provisioning

Deployment



Firmware Updates

Firmware Configuration

Automation

Security

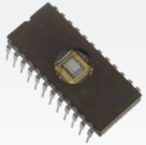
Scalability

Ecosystem

The UEFI Solution

Bare Metal Provisioning

- Pre-Boot Networking
- IPv4, IPv6 TCP/UDP
- PXE, iSCSI, HTTP, FTP



Firmware Updates

- Firmware Management Protocol
- Capsule Updates

Deployment

- Boot Device Selection
- Boot Order control
- OS install & recovery

Firmware Configuration

- Human Interface Infrastructure (HII)
- Platform-To-Driver Config (CLP)
- REST Protocol

Automation

- UEFI Shell
- Scripting language

Scalability

- New Hardware abstraction with UEFI Protocols
- UEFI Driver model
- UEFI Device Path

The UEFI Solution

Security

- Secure Boot and Driver Signing
- Security technologies (OpenSSL*, RNG, etc...)
- Encrypted Disks and Key Management
- Interoperability with TCG standards

Eco-system

- Standards (UEFI Forum)
- Compliance: Self Certification Test (SCT), Linux* UEFI Validation (LUV)
- Open source code (EDK2 - http://tianocore.org)
- Ubiquitous vendor support (OEMs, ISVs, IHVs, OSVs)

UEFI offers solutions to today's data center firmware challenges

Data Center Manageability Interface Requirements

- **Use security best practices**
- **Support modern architectures**
  - Describe modern architectures (multi-node servers)
  - UEFI-aware (boot order selection, Secure Boot)
- **Scaling**
  - Scale-out servers usage model drastically different from traditional/enterprise servers
  - Management complexities grow exponentially
- **Interoperability for “OEM extensions”**

Today’s Data Center Manageability Interfaces do not meet all of these needs

PXE and HTTP Boot

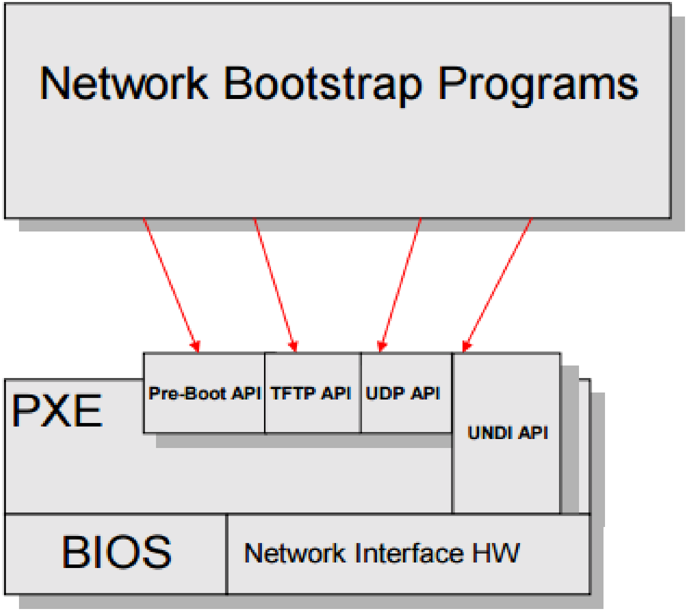
Bare Metal Provisioning

Deployment

Security

PXE Boot Challenges

- **P**reboot e**X**ecution **E**nvironment
- Security Issues
  - Only physical. No encryption or authentication.
  - Rouge DHCP servers, man-in-the-middle attacks
- Scaling issues
  - Circa 1998
  - TFTP timeouts / UDP packet loss
  - Download time = deployment time = $$$
  - Aggravated in density-optimized data centers
- OEMs and users workarounds
  - Chain-load 3rd party boot loaders (iPXE, mini-OS)



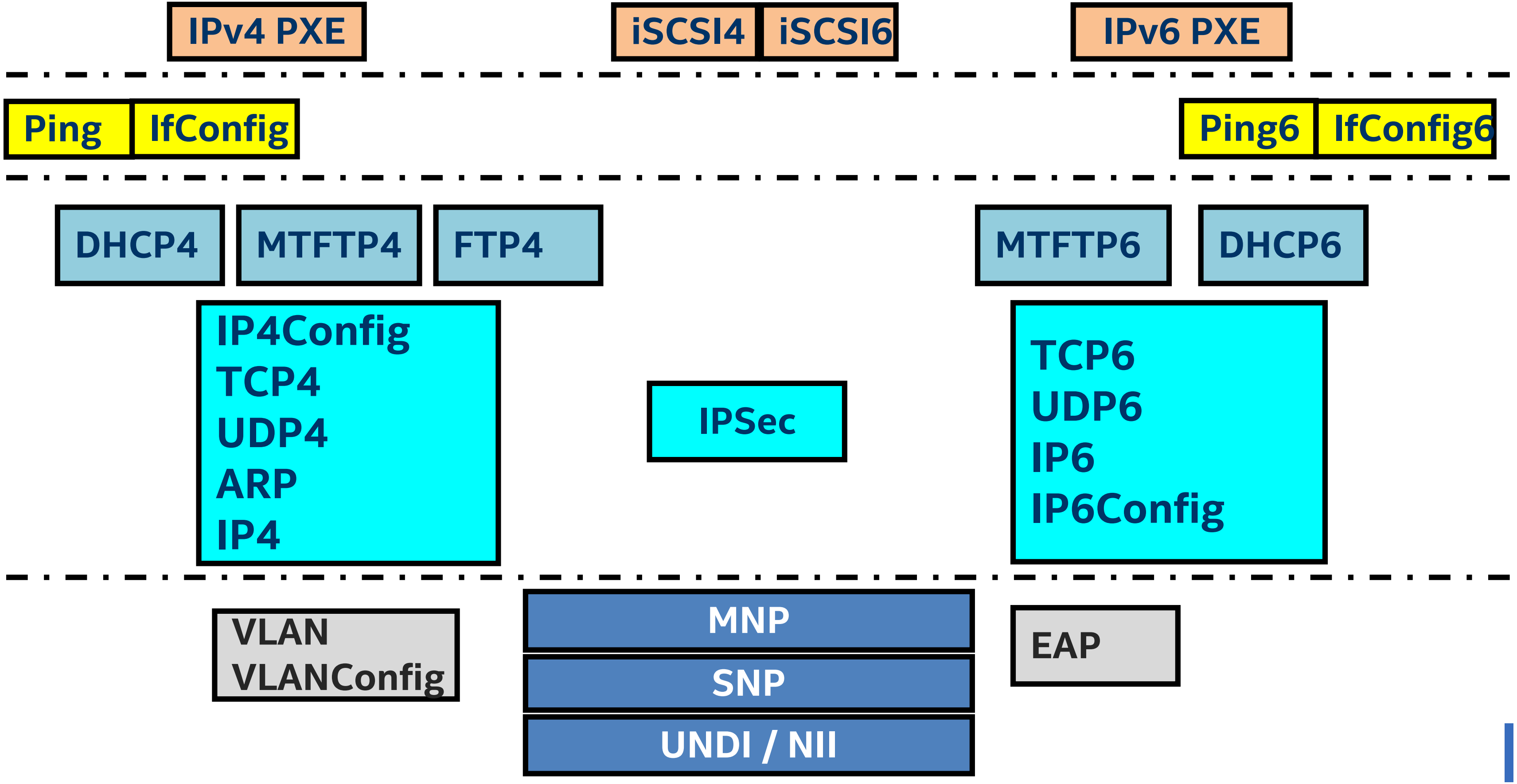
PXE is not keeping up with the modern data centers requirements

iPXE (http://ipxe.org)

- Open-source PXE client and bootloader
- Adds support of HTTP Boot, but currently:
  - Only works with Traditional BIOS
  - Only provides low-level SNP interface (no HTTP Boot) in UEFI
  - Users have to choose between **HTTP Boot** and **UEFI Secure Boot**
- iPXE UEFI vision
  - *"Provide the same advanced features within the UEFI environment as are currently provided within the Traditional BIOS environment"*
    - http://ipxe.org/efi/vision

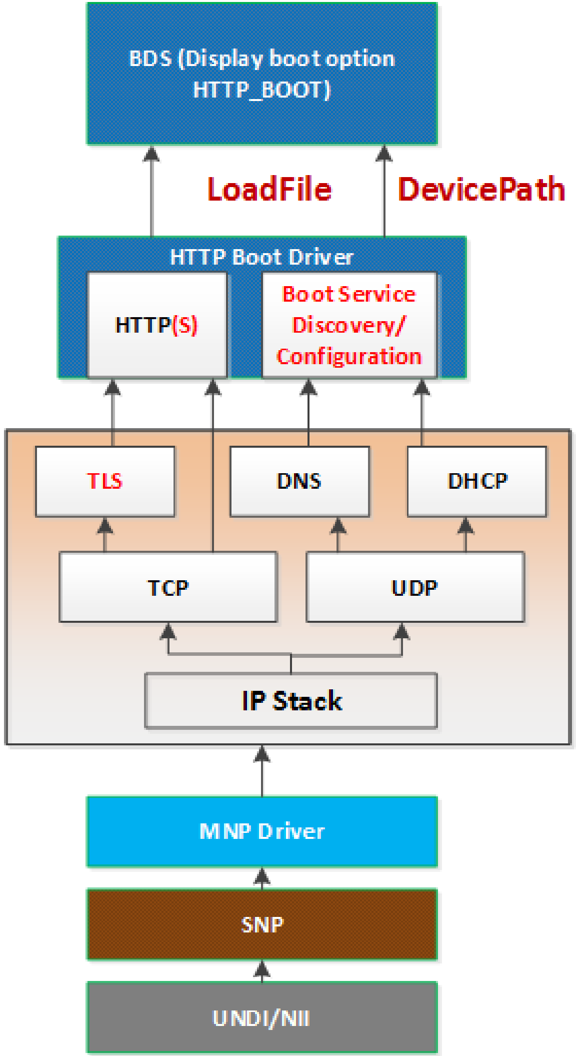
Why not solve the PXE boot challenges natively in a standard way in UEFI?

Network Stack in UEFI v2.4



Network Stack in UEFI v2.5

- Builds on top of UEFI 2.4
- DNS (IPv4 / IPv6)
- HTTP (IPv4 / IPv6)
- TLS (for HTTPs)
- HTTP Boot Wire Protocol
- Bluetooth® technology
- Wi-Fi*



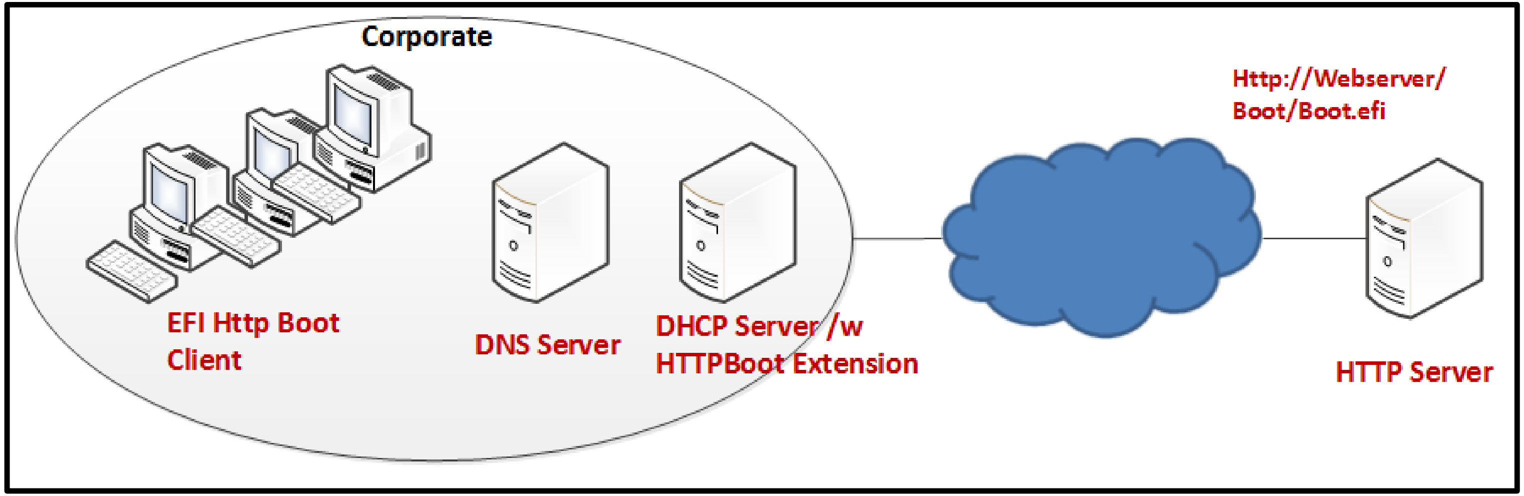
UEFI Native HTTP Boot

HTTP Boot Wire Protocol

- Boot from a URL
- Target can be:
  1. EFI Network Boot Program (NBP)
  2. Shrink-wrapped ISO image
- URL pre-configured or auto-discovered (DHCP)

Addresses PXE issues

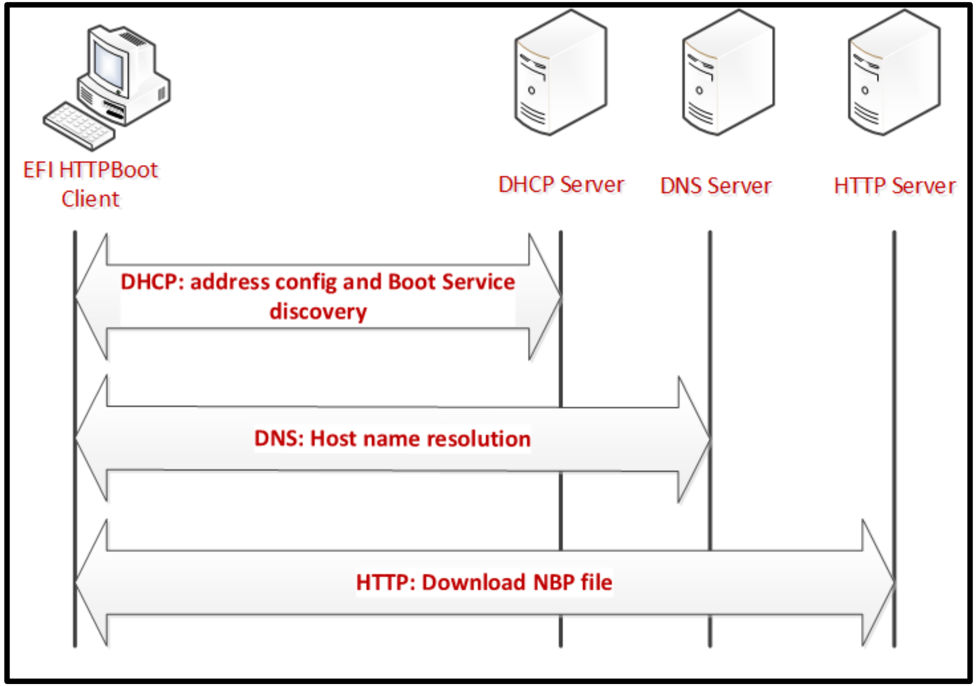
- HTTPs addresses security
- TCP reliability
- HTTP load balancing



HTTP Boot DHCP Discovery

HTTP Boot DHCP Discovery

- New HTTP Boot "Architectural Types" to distinguish from PXE
- Client sends DHCP Discover request
- DHCP Server responds with offer that includes the boot file URL
- Clients resolves URL server name from DNS
- Client downloads boot image from HTTP server using HTTP(s)



RAM Disk Standard

- UEFI 2.5 defined RAM Disk device path nodes
  - Standard access to a RAM Disk in UEFI
  - Supports Virtual Disk and Virtual CD (ISO image) in persistent or volatile memory

- ACPI 6.0 NVDIMM Firmware Interface Table (NFIT)
  - Describe the RAM Disks to the OS
  - Runtime access of the ISO boot image in memory

HTTP Boot is the emerging solution for modern data centers!

UEFI Shell Scripting

script...

Automation

UEFI Shell

- UEFI Pre-boot command line interface (CLI)
  - Much like DOS* or Linux*/Unix* Shell environment
- Interactive prompt and scriptable
- Built-in commands
  - **Standard Commands:** File manipulations, driver management, device access, scripting control, system information, basic network operations
  - **Extensible:** OEMs can provide value-add commands
- Can be embedded as a boot option or bootable from storage
- Fully documented
  - Latest UEFI Shell Specification v2.1

```
Shell> _
```

UEFI Shell Standard Commands

Scripting

- echo, stall, set, shift, pause, parse, if / else / endif, for/endfor, reset, exit, cls
- **startup.nsh** auto-start script
- Parsable comma-separated output (-sfo)

File Operations

- dir cd, md, rd, mv, copy, del, type, edit, touch, attrib, setsize, comp, compress
- Read/Write files (FAT/FAT32)
- Console/file redirection and piping

Debug and Test

- **UEFI Drivers Debug:** load, unload, connect, disconnect, drivers, devices, devtree, dh, openinfo
- **System debug:** memmap, dmem, smbiosview, pci, dblk

Data Center Manageability: Redfish and REST APIs

Firmware Configuration

Scalability

Security

Data Center Manageability Interface Requirements

- **Use security best practices**
- **Support modern architectures**
- **Scaling**
- **Interoperability for “OEM extensions”**

Today’s Data Center Manageability Interfaces do not meet all of these needs

What is Redfish?

- **Architectural successor to previous manageability interfaces**
- **Industry Standard**
  - DMTF* Scalable Platforms Management Forum (SPMF)
  - www.dmtf.org/standards/redfish
  - Specification, schema, mockup, whitepaper, FAQ, resource browser
- **RESTful interface over HTTPs**
  - JSON format
  - Secure (HTTPs)
  - Multi-node and aggregated rack-level servers capable
  - Schema-backed, human readable output

What is REST?

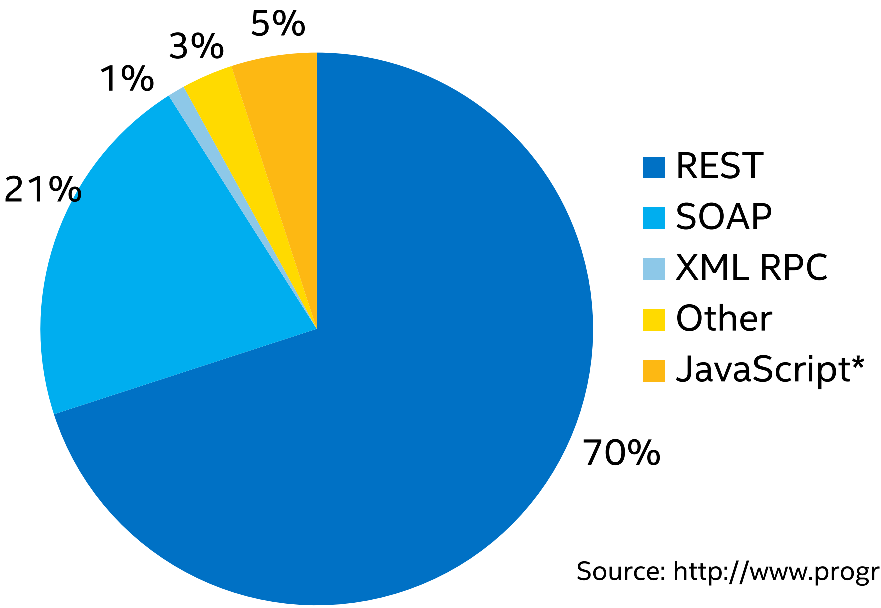
- **RE**presentational **S**tate **T**ransfer
- Scalable Software Architectural “style”
- Standardized operations (verbs)
  - HTTP GET, POST, PUT, and DELETE
  - Practical implementations add HTTP PATCH, HEAD
- Standardized operands (nouns)
  - Resources uniquely identified by URIs
- Stateless, atomic operations
  - No client/application context stored

What is JSON?

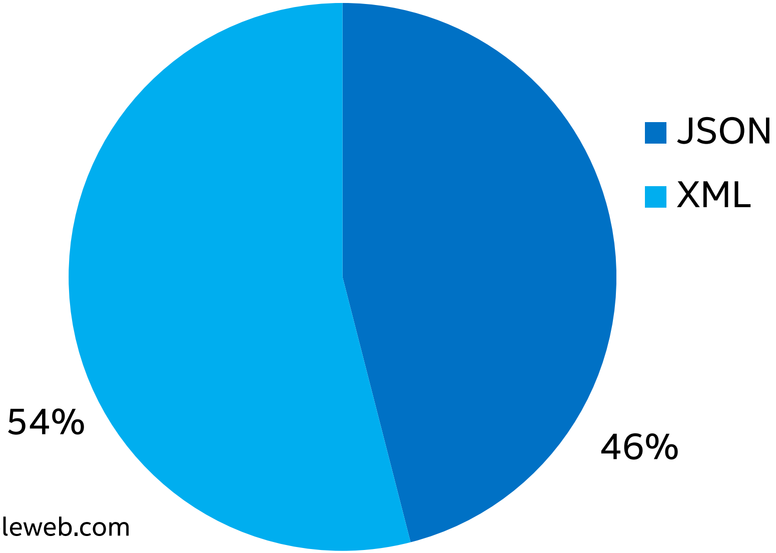
- **J**ava **S**cript **O**bject **N**otation
- Lightweight data-interchange format
  - Easy for humans to read and edit
  - Easy for machines to parse and generate
- Much smaller grammar than XML
  - XML good for “documents”
  - JSON better for “data structures” used in programming languages

REST and JSON in WWW APIs

WWW Programmable APIs



WWW APIs Data formats

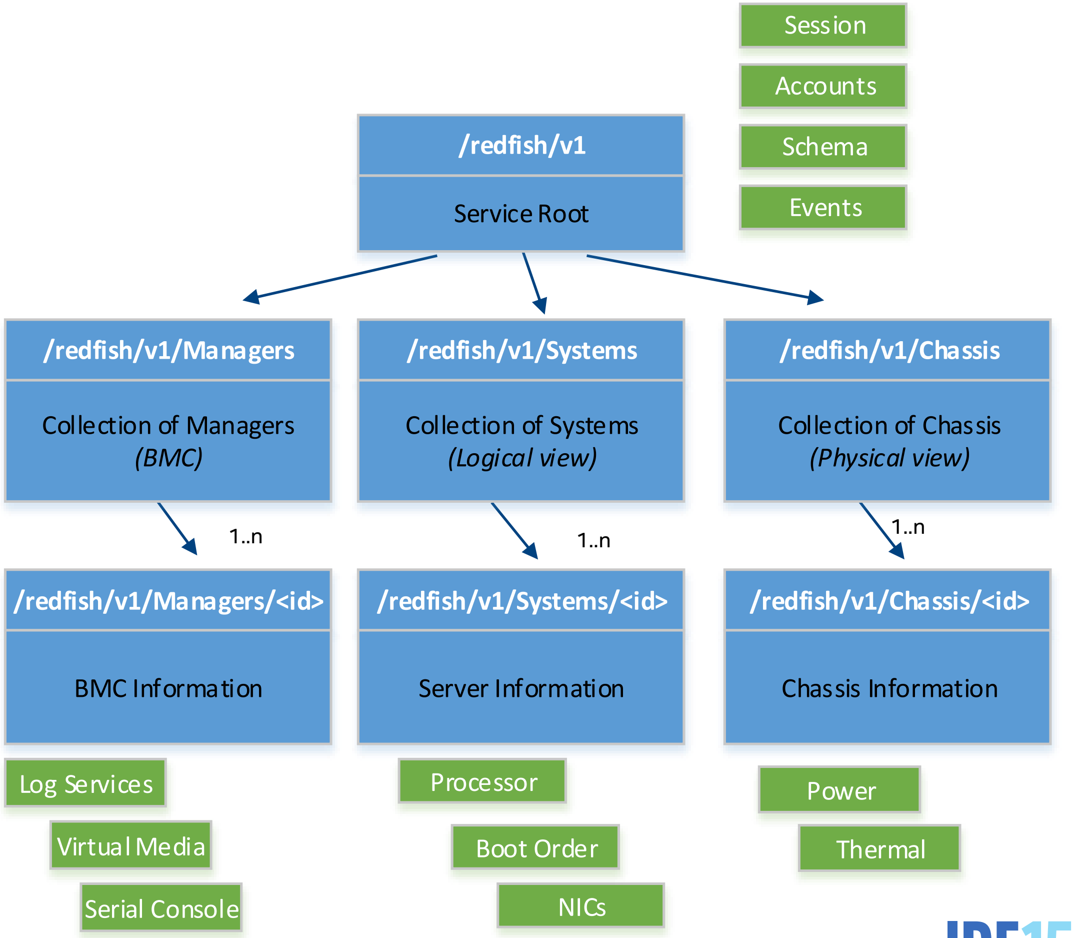


Source: http://www.programmableweb.com

REST and JSON: Simple Wins!

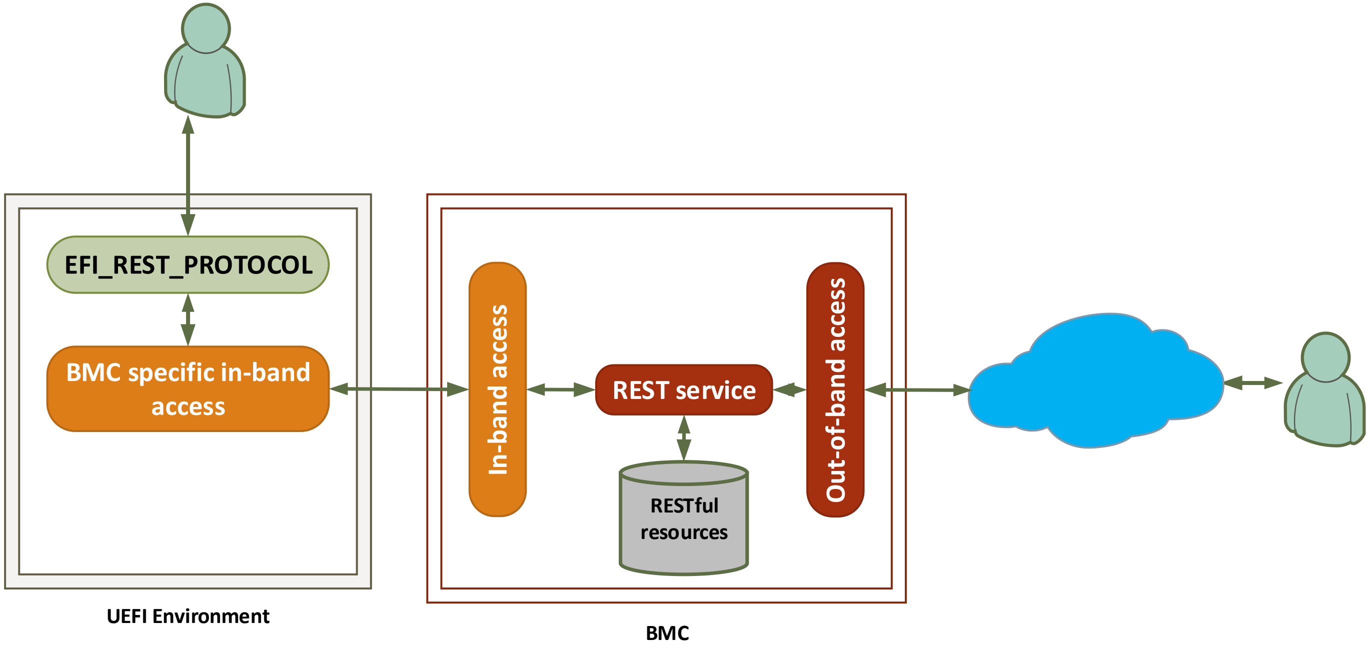
Redfish Data Model

- Root of service "/redfish/v1"
- Each resource has a type
  - Versioned schema
  - Meta-data
  - OEM extensions
- Collections to describe versatile server hardware architectures
  - Stand-alone
  - Multi-node
  - Rack-level aggregated



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)



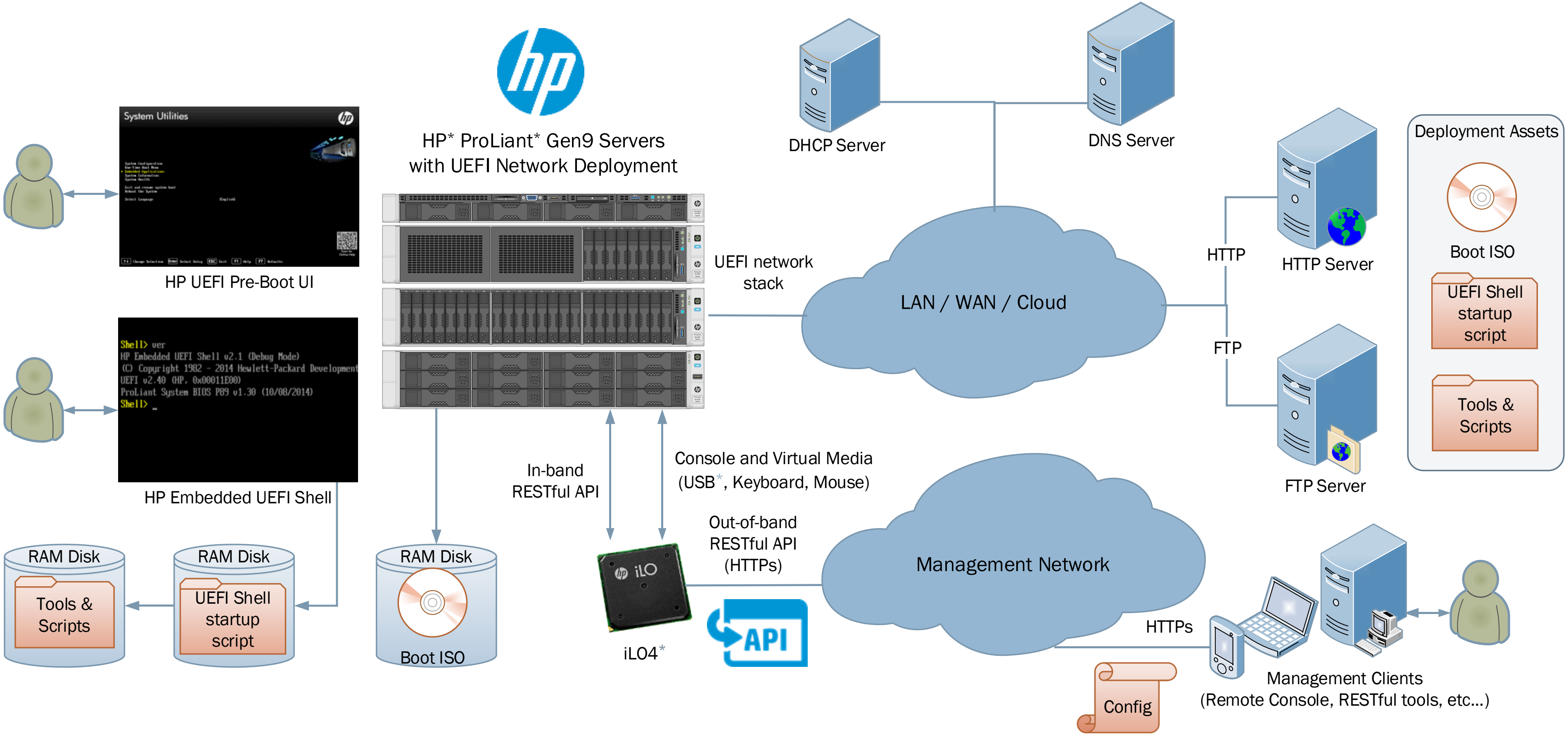
Putting it all together : HP* ProLiant* Servers

UEFI Deployment Solution on HP* ProLiant* Servers

- **UEFI Network Stack Extensions**
  - HTTP, FTP, DNS
  - "Boot from URL" to EFI file or ISO image
  - UEFI iSCSI Software Initiator
- **HP RESTful API**
  - Accessible in-band (from OS) or out-of-band (iLO4* HTTPs). Redfish conformance soon.
  - HP* OEM extensions including support for UEFI BIOS configuration

- **Embedded UEFI Shell**
  - Built into the system firmware
  - HP value-add commands for bare-metal deployment
  - Startup script loading from media or network location

UEFI Deployment Solution on HP* ProLiant* Servers

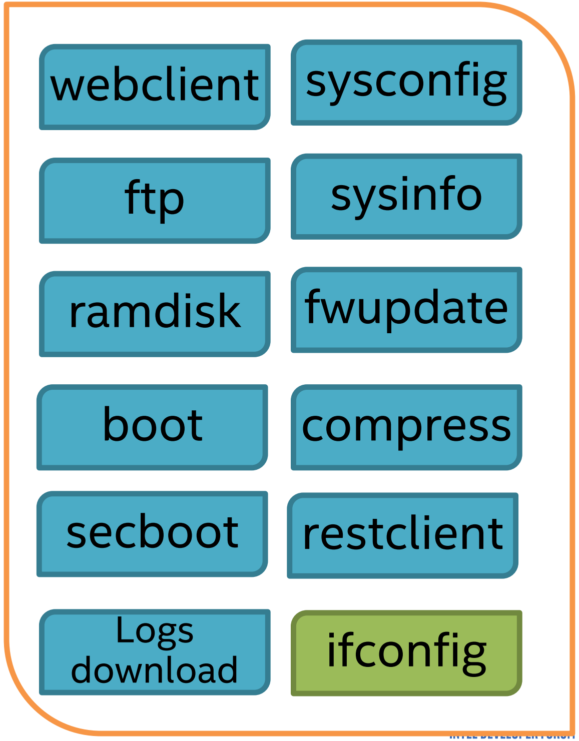


```
Shell> ver
HP Embedded UEFI Shell v2.1 (Debug Mode)
(C) Copyright 1982 - 2014 Hewlett-Packard Development
UEFI v2.40 (HP, 0x00011E00)
ProLiant System BIOS P89 v1.30 (10/08/2014)
Shell> _
```

Embedded UEFI Shell HP* Commands

```
Shell> _
```

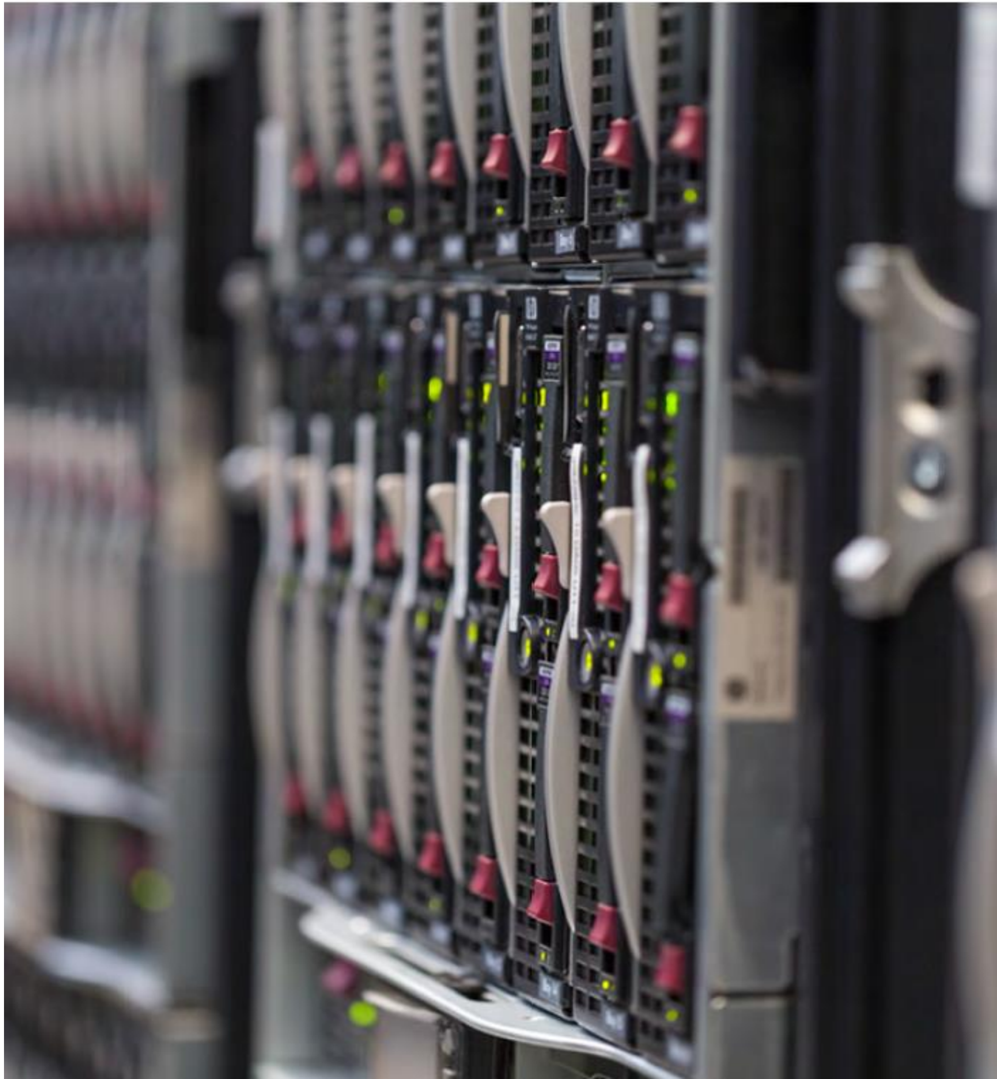
- **HP* value-add commands for bare-metal deployment**
- **ramdisk** : Provision memory disks and mount ISO files
- **webclient** and **ftp :** Scriptable network download/upload
- **restclient**: In-band client for the HP RESTful API
- **sysconfig** : Configuration CLI (integrates with HP* RESTful API)
- **secboot** : Secure Boot management (physical presence)
- **boot** : Transition to OS/boot targets without rebooting
- **sysinfo** : System hardware/firmware inventory
- **fwupdate** : Firmware updates
- **compress** : ZIP/UNZIP archives
- **ifconfig** : UEFI network stack configuration
- Commands to collect server service/troubleshooting logs



webclient
sysconfig
ftp
sysinfo
ramdisk
fwupdate
boot
compress
secboot
restclient
Logs download
ifconfig

HP* RESTful API

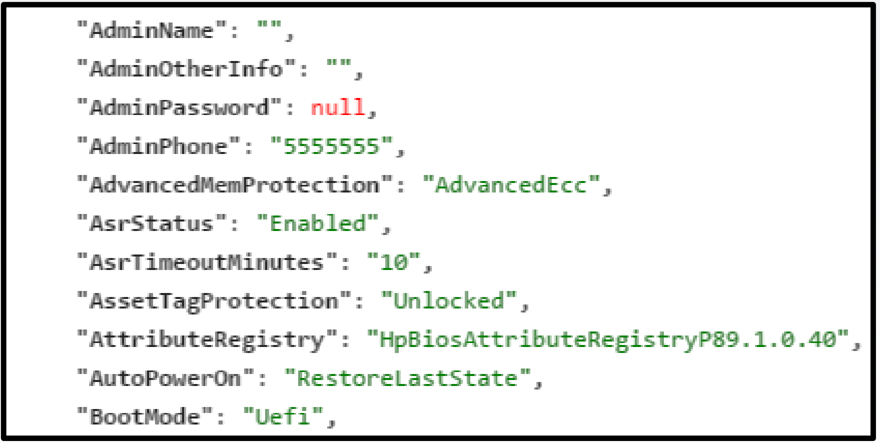
- HP* RESTful API in iLO4*
  - Modern management API for HP ProLiant* and Moonshot servers
  - Comprehensive inventory and server configuration
- Integrated with UEFI
  - UEFI BIOS settings configuration
  - UEFI Boot Order and Secure Boot configuration
  - UEFI iSCSI Software Initiator configuration



HP* RESTful API Example: UEFI BIOS Settings

GET @ /rest/v1/systems/1/bios

- Get a list of all UEFI BIOS settings (name/values)

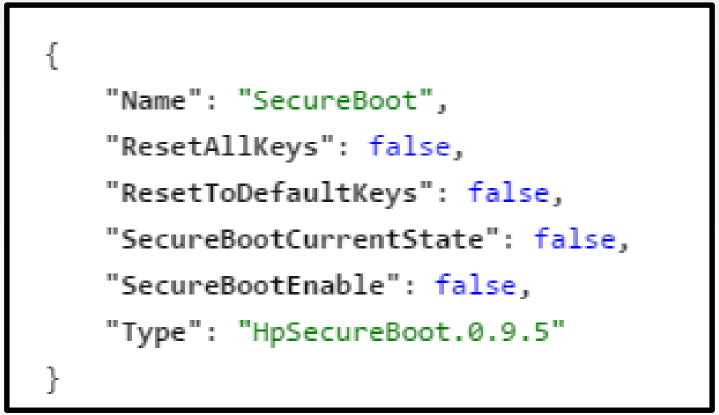


```
"AdminName": "",
"AdminOtherInfo": "",
"AdminPassword": null,
"AdminPhone": "5555555",
"AdvancedMemProtection": "AdvancedEcc",
"AsrStatus": "Enabled",
"AsrTimeoutMinutes": "10",
"AssetTagProtection": "Unlocked",
"AttributeRegistry": "HpBiosAttributeRegistryP89.1.0.40",
"AutoPowerOn": "RestoreLastState",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

HP* RESTful API Example: Secure Boot

GET @ /rest/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"
}
```

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
```

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.hp.com/deploy/efilinux.efi
webclient -g http://repo.hp.com/deploy/deploy.kernel
webclient -g http://repo.hp.com/deploy/deploy.ramdisk


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

Summary and Q&A

Summary and Next Steps

- UEFI 2.5 HTTP Boot bridges the gaps of network boot in the data center
- Redfish is emerging RESTful management API to address modern data center requirements
- HP* ProLiant* Servers showcase of a bare-metal UEFI deployment solution using HTTP Boot, Embedded UEFI Shell, and RESTful APIs

Next Steps:

- Adopt UEFI 2.5 implementations with HTTP Boot (now on open source)
- Adopt Redfish implementations in servers and management software
- Transition data centers to use HTTP Boot and Redfish REST APIs

Additional Sources of Information

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

- More web based info:
  - UEFI Forum Learning Center: http://uefi.org/learning_center
  - UEFI 2.5 and ACPI 6.0 Specifications: http://www.uefi.org/specs/
  - Redfish Specification: http://www.dmtf.org/standards/redfish
  - UEFI on HP* ProLiant* Servers: http://hp.com/go/proliant/uefi
  - Open source UEFI EDK II Tianocore.org
  - HTTP Boot in the news

Other Technical Sessions

Session ID	Title	Day	Time	Room
✓ STTS001	Firmware in the Data Center: Building a Modern Deployment Framework Using UEFI and Redfish REST APIs	Tue	11:00	2002
STTS002	Building a Firmware Component Ecosystem with the Intel® Firmware Engine	Tue	1:15	2002
STTS003	Developing Best-in-Class Security Principles with Open Source Firmware	Tue	2:30	2002
STTC003	Tech Chat: Using Intel® Firmware Engine to Generate Simulated Platforms for Wind River Simics*	Wed	1:00	Level 2 Tech Chat Station 5
INFS006	Exploring Redfish - Emerging Manageability Standards	Wed	2:30	2002

✓ = DONE

See also:

- Technical Showcase Booths #763 (Redfish demo), #511 (Intel UEFI)

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.

Risk Factors

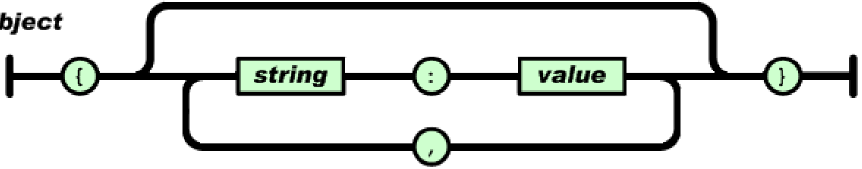
The above statements and any others in this document that refer to plans and expectations for the second quarter, the year and the future are forward-looking statements that involve a number of risks and uncertainties. Words such as "anticipates," "expects," "intends," "plans," "believes," "seeks," "estimates," "may," "will," "should" and their variations identify 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. 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. 4/14/15

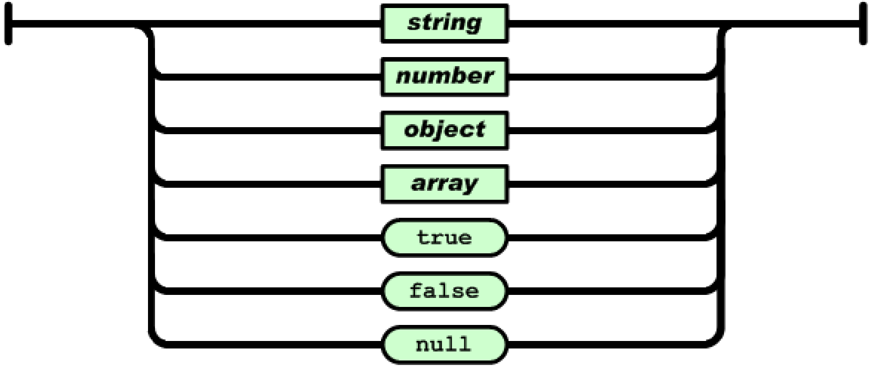
Backup

JSON Grammar

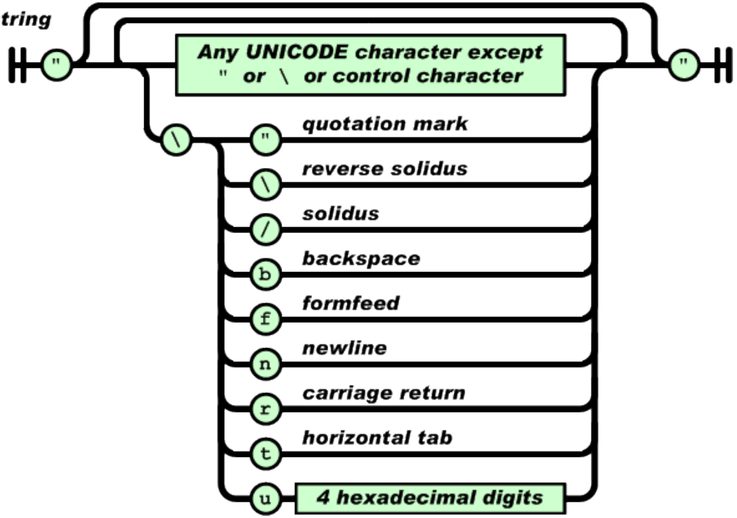
object



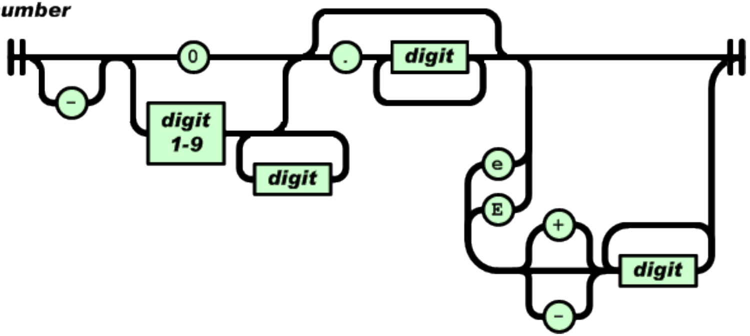
value



string



number



Source: http://www.json.org