presented by

Firmware in the datacenter: Goodbye PXE and IPMI. Welcome HTTP Boot and Redfish!

UEFI Spring Plugfest – May 18-22, 2015

Samer El-Haj-Mahmoud
Master Technologist
Hewlett Packard

Agenda

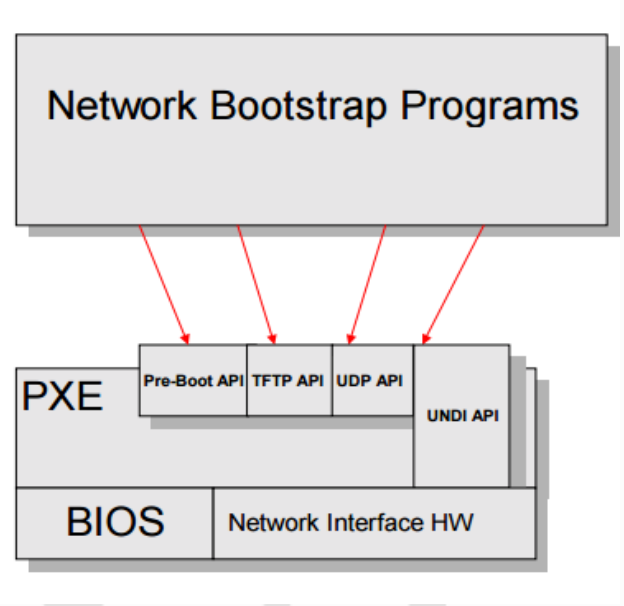
- PXE and HTTP Boot
  - PXE Challenges
  - UEFI 2.5 HTTP Boot
- IPMI and Redfish
  - IPMI Challenges
  - Redfish and REST APIs
- Putting it all together
  - Case study of HP ProLiant Servers

Goodbye PXE. Welcome HTTP Boot!

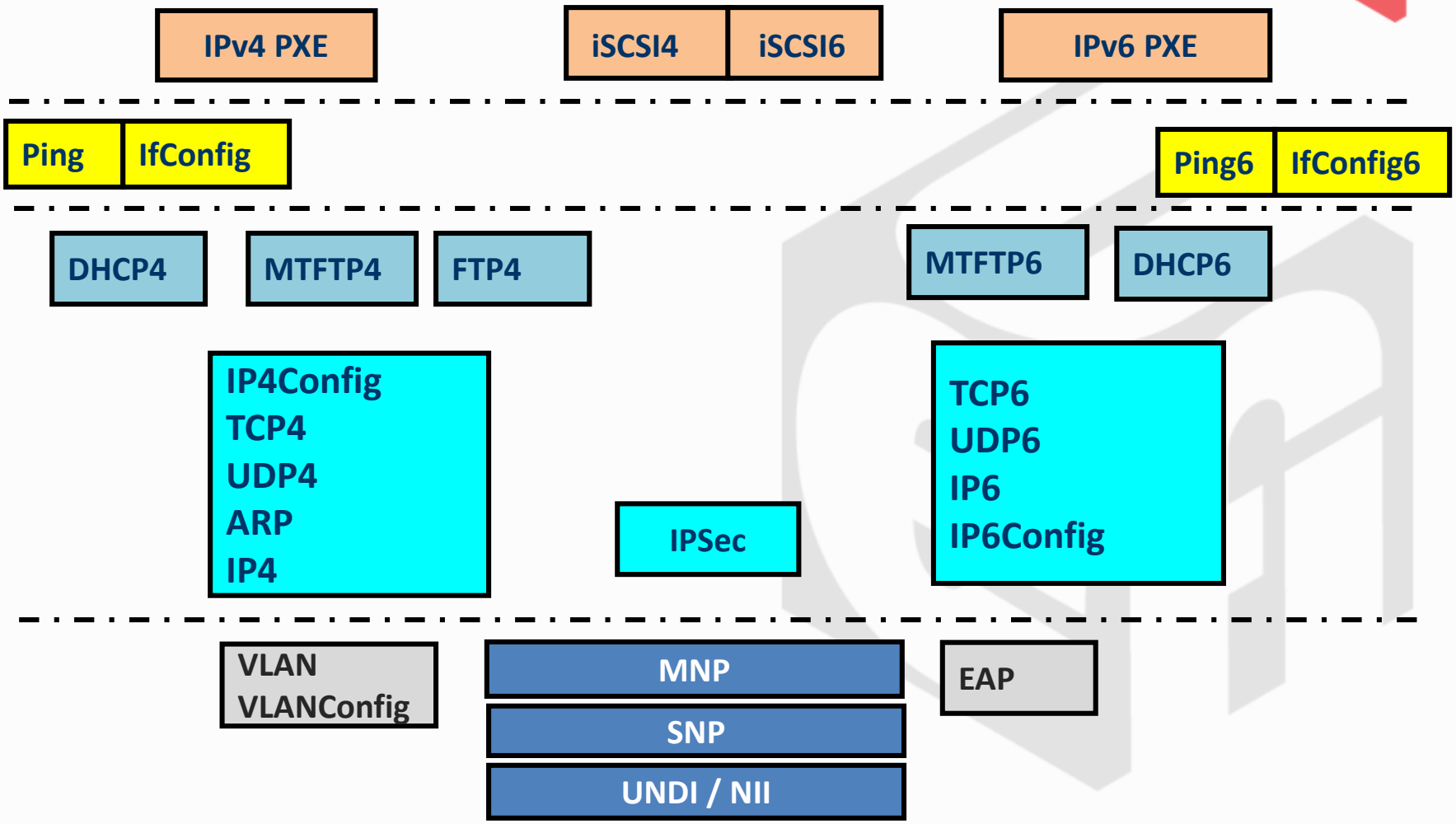
PXE Boot Challenges

Preboot e**X**ecution **E**nvironment

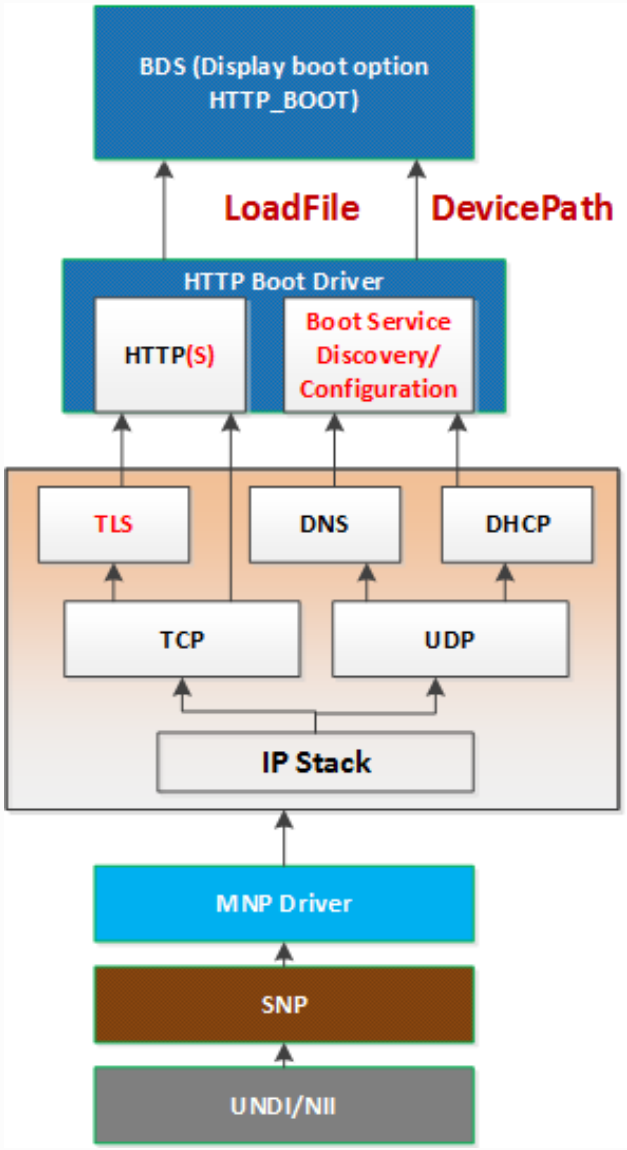
- Security Issues
  - Only physical. No encryption or authentication
  - Rouge DHCP servers, man-in-the-middle attacks
- Scaling issues
  - Circa 1998
  - TFTP timeouts
  - UDP packet loss
  - Download time = Deployment time = $$$
  - Aggravated in density-optimized data datacenters
- OEMs and Users “duct-tape”
  - Retry (DHCP request, download, reboot, etc…)
  - Chain-load 3rd party boot loaders (iPXE, mini-OS, etc…)
  - Alternate net-booting (Boot from SAN, iSCSI, etc…)



UEFI 2.4 Network Stack



UEFI 2.5 Network Stack

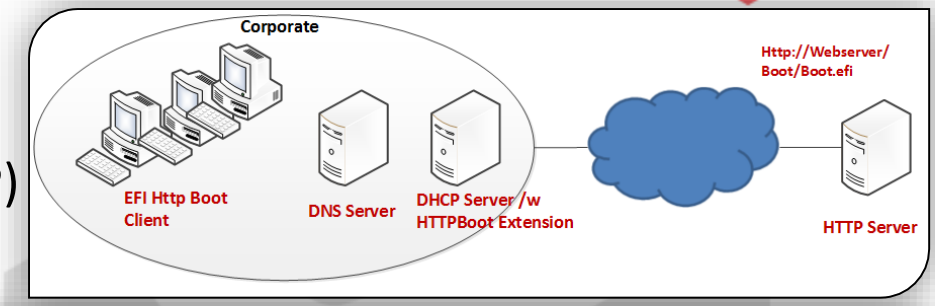


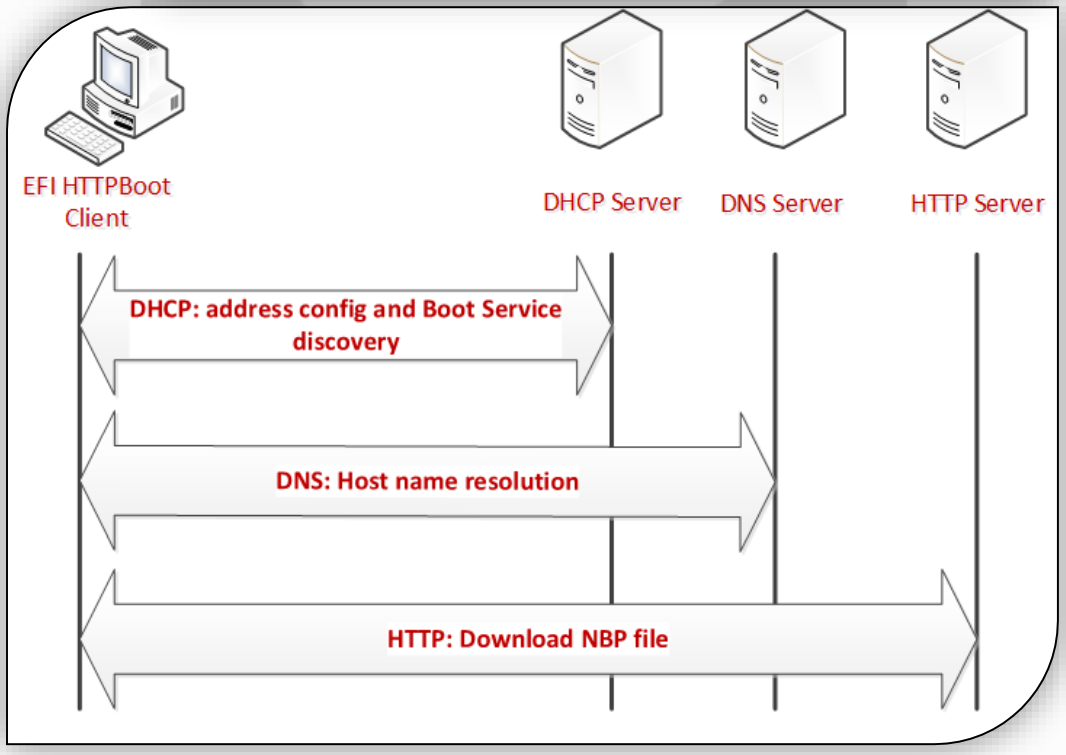
- DNS support
  - EFI_DNS4_SERVICE_BINDING_PROTOCOL
  - EFI_DNS6_SERVICE_BINDING_PROTOCOL
  - EFI_DNS4_PROTOCOL
  - EFI_DNS6_PROTOCOL
  - EFI_IP4_CONFIG2_PROTOCOL
- HTTP support
  - EFI_HTTP_SERVICE_BINDING_PROTOCOL
  - EFI_HTTP_PROTOCOL
  - EFI_HTTP_UTILITIES_PROTOCOL
  - HTTP Boot Wire Protocol
- TLS support
  - EFI_TLS_SERVICE_BINDING_PROTOCOL
  - EFI_TLS_PROTOCOL
  - EFI_TLS_CONFIGURATION_PROTOCOL

UEFI 2.5 HTTP Boot

- **HTTP Boot Wire Protocol**
  - Boot from configured 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 “Architectural Types”
- http://www.iana.org/assignments/dhcpv6-parameters/dhcpv6-parameters.xml
- IPv4/IPv6 DHCP Discover request
  - DHCP Option 93: Client system Architecture
  - DHCPv6 Option 61: Client system Architecture

    0x10 = x64 EFI boot from HTTP

    0x13 = AArch64 EFI boot from HTTP

- Server responds with DHCPOFFER that includes the boot file HTTP URI for the requested processor architecture

Processor Architecture Types

Registration Procedure(s)
Expert Review
Expert(s)
Vincent Zimmer
Reference
[RFC5970]
Available Formats
CSV

Type	Architecture Name	Reference
0x00 0x00	x86 BIOS	[RFC5970][RFC4578]
0x00 0x01	NEC/PC98 (DEPRECATED)	[RFC5970][RFC4578]
0x00 0x02	Itanium	[RFC5970][RFC4578]
0x00 0x03	DEC Alpha (DEPRECATED)	[RFC5970][RFC4578]
0x00 0x04	Arc x86 (DEPRECATED)	[RFC5970][RFC4578]
0x00 0x05	Intel Lean Client (DEPRECATED)	[RFC5970][RFC4578]
0x00 0x06	x86 UEFI	[RFC5970][RFC4578]
0x00 0x07	x64 UEFI	[RFC5970][RFC4578]
0x00 0x08	EFI Xscale (DEPRECATED)	[RFC5970][RFC4578]
0x00 0x09	EBC	[RFC5970][RFC4578]
0x00 0x0a	ARM 32-bit UEFI	[RFC5970]
0x00 0x0b	ARM 64-bit UEFI	[RFC5970]
0x00 0x0c	PowerPC Open Firmware	[Thomas_Huth]
0x00 0x0d	PowerPC ePAPR	[Thomas_Huth]
0x00 0x0e	POWER OPAL v3	[Jeremy_Kerr]
0x00 0x0f	x86 uefi boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x10	x64 uefi boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x11	ebc boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x12	arm uefi 32 boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x13	arm uefi 64 boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x14	pc/at bios boot from http	[Samer_El-Haj-Mahmoud]
0x00 0x15	arm 32 uboot	[Joseph_Shifflett]
0x00 0x16	arm 64 uboot	[Joseph_Shifflett]
0x00 0x17	arm uboot 32 boot from http	[Joseph_Shifflett]
0x00 0x18	arm uboot 64 boot from http	[Joseph_Shifflett]
0x00 0x19-0xff 0xff	Unassigned	

UEFI 2.5 / ACPI 6.0 RAM Disk

- **UEFI 2.5 defined RAM Disk device path nodes**
  - Standard access to a RAM Disk in UEFI
  - Supports Virtual Disk and Virtual CD (ISO) 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

Type: 4 (Media Device Path) SubType: 9 (RAM Disk)	**RamDisk** (*StartingAddress*,*EndingAddress*,*DiskInstance*,*DiskTypeGuid*) The *StartingAddress* and *EndingAddress* are both 64-bit integers and are both required. The *DiskInstance* is a 16-bit integer and is optional. The default value is 0. The *DiskTypeGuid* is a GUID and is required.

Goodbye IPMI. Welcome Redfish!

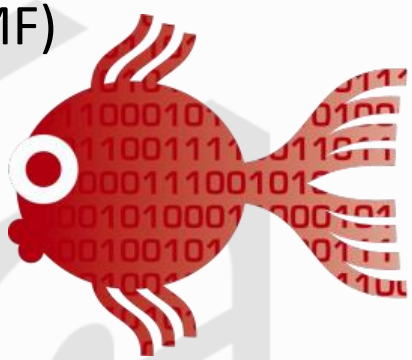
IPMI Challenges

Intelligent **P**latform **M**anagement **I**nterface

- Security (or lack of)
  - Lacks modern security best practices
  - *"IPMI 2.0 RAKP Authentication Remote Password Hash Retrieval"* vulnerability (IPMI 2.0 spec "feature")
- Out-of-Date
  - Lacks the ability to describe modern architectures (e.g. multi-node servers)
  - Not UEFI-aware (boot device/order selection, Secure Boot, etc…)
- Scaling Limits
  - Scale-out servers usage model drastically different from traditional and enterprise servers
  - Management complexities grow exponentially at scale
- Unique "OEM extensions" create inconsistencies

What is Redfish?

- Industry standard replacement to IPMI
  - DMTF Scalable Platforms Management Forum (SPMF)
- RESTful interface over HTTPs
  - JSON format (Based on OData v4)
  - Secure (HTTPs)
  - Multi-node and aggregated rack-level servers capable
  - Schema-backed, human readable output
- Current Specification
  - Version 0.97.0 (Redfish API Specification , Schema, Mockup Data)
  - Targeting 1.0.0 in June, 2015



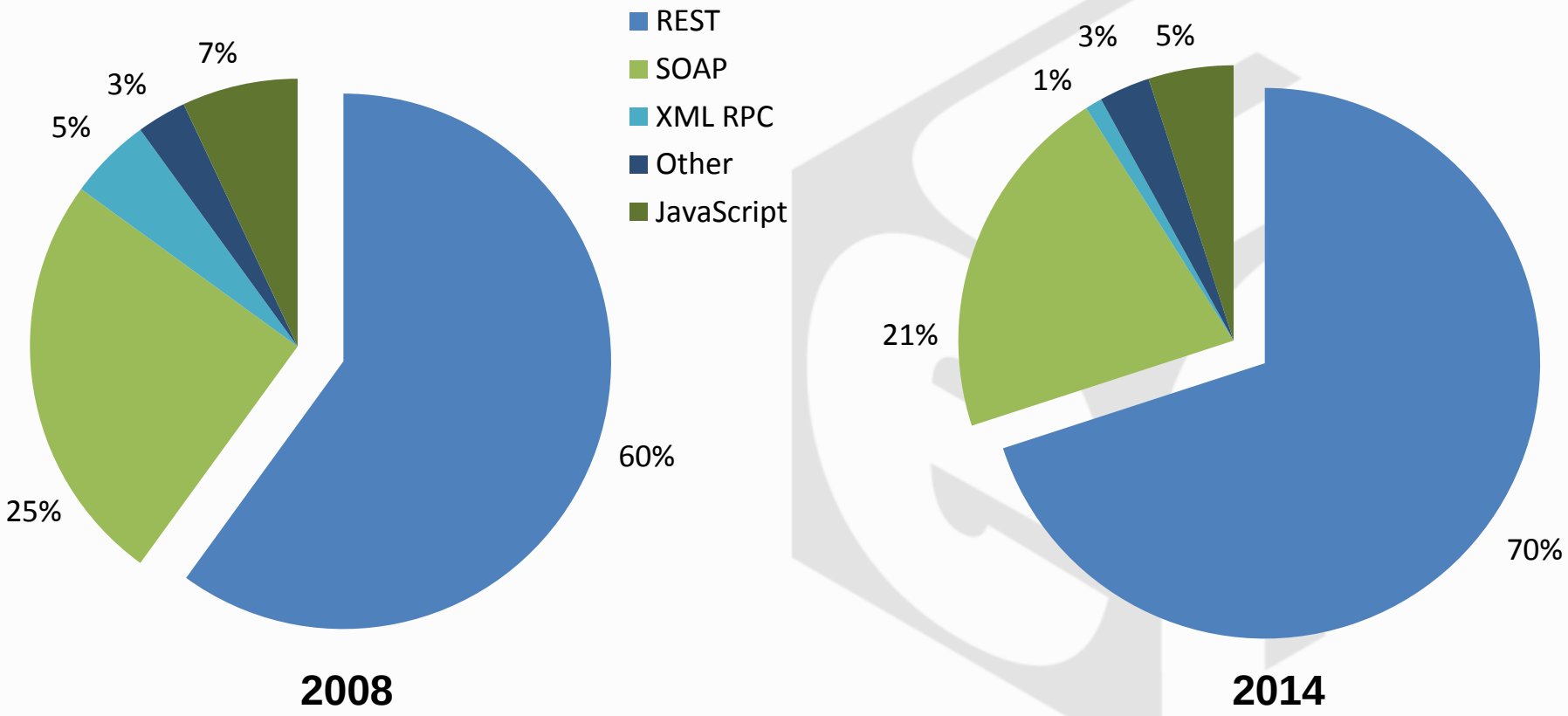
What is REST?

REpresentational **S**tate **T**ransfer

- Scalable Software Architectural “style” for the WWW
  - Defined by Roy Fielding, principal author of HTTP specification
- Standardized operations (verbs)
  - HTTP GET, POST, PUT, PATCH, HEAD and DELETE
- Standardized operands (nouns)
  - Resources (and resource states), uniquely identified by URIs
- Stateless, atomic operations
  - No client/application context stored on server

REST: Simple Wins

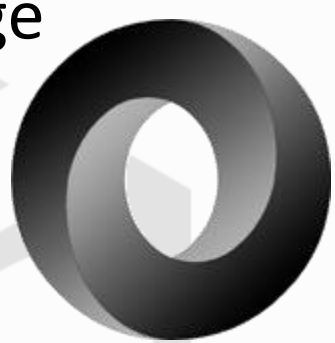
70% of WWW APIs



Source: http://www.programmableweb.com

What is JSON?

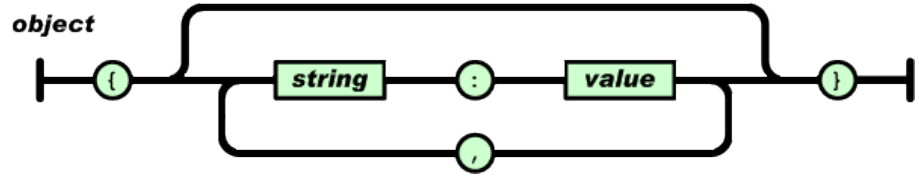
Java **S**cript **O**bject **N**otation

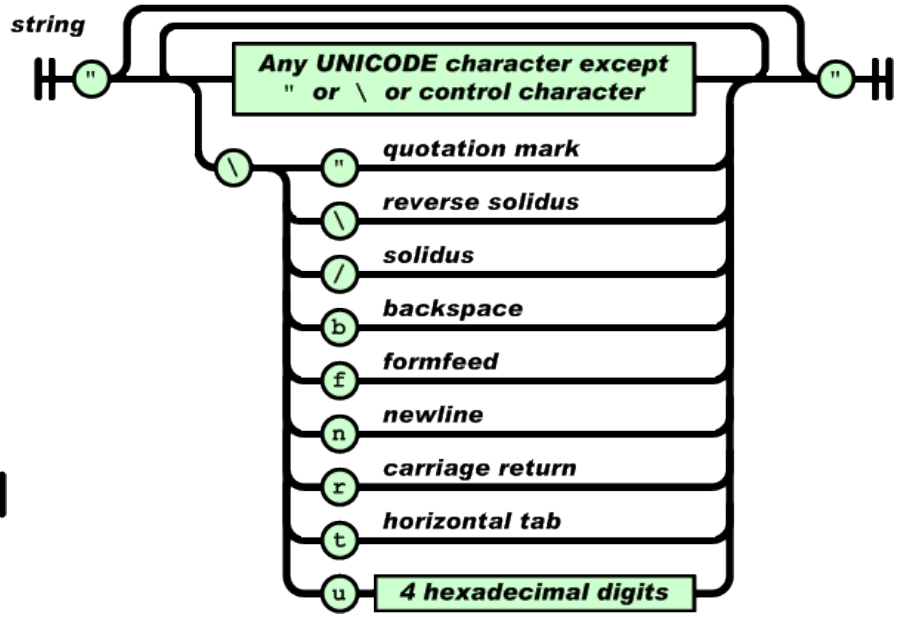


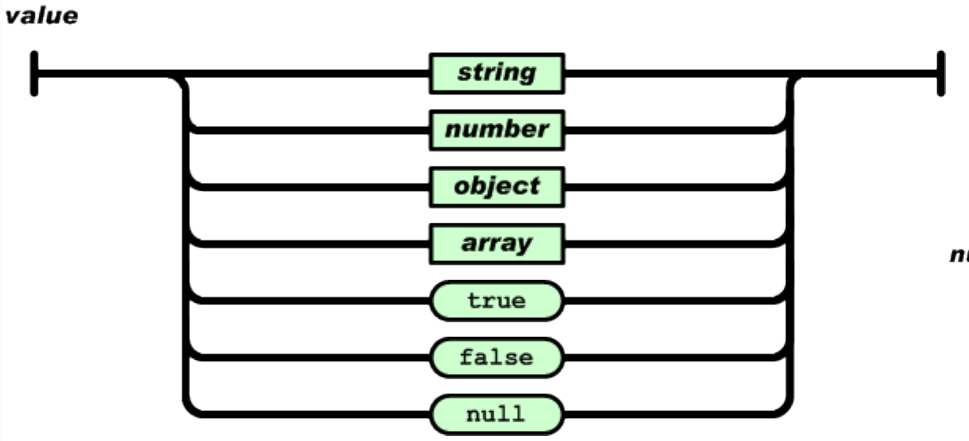
- Lightweight human readable data-interchange format
  - Easy for humans to read and write
  - Easy for machines to parse and generate
- Much smaller grammar than XML
  - XML good for “documents”
  - JSON better for “data structures” used in programming languages

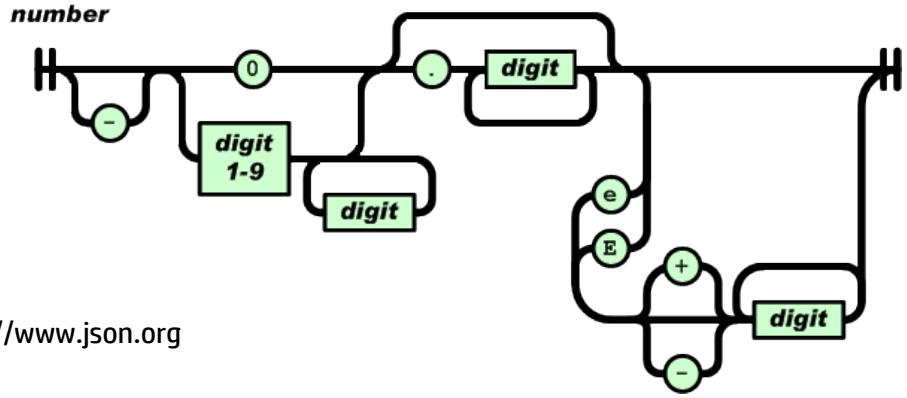
JSON Grammar

JSON One-Pager





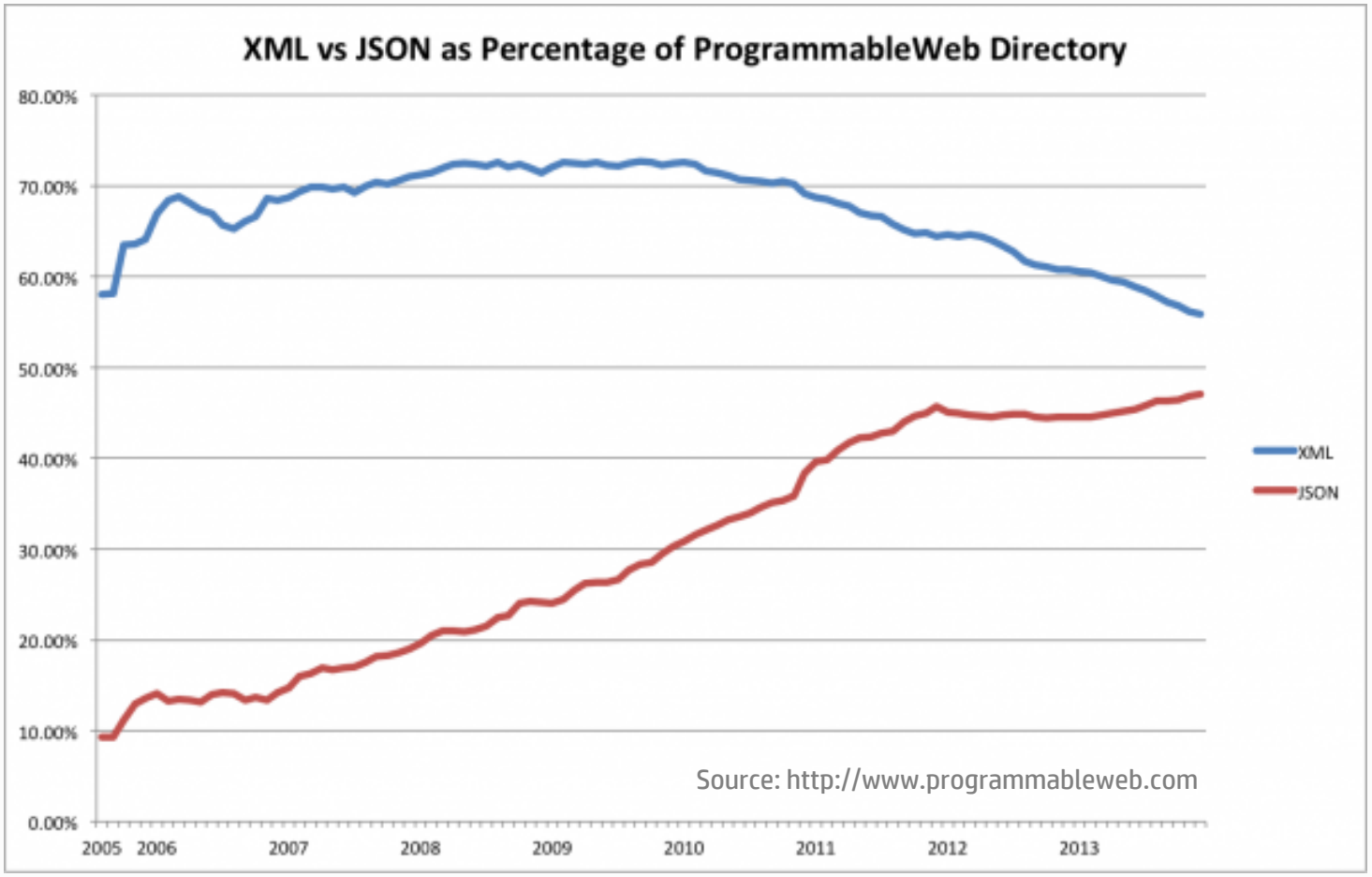




Source: http://www.json.org

JSON on the Rise

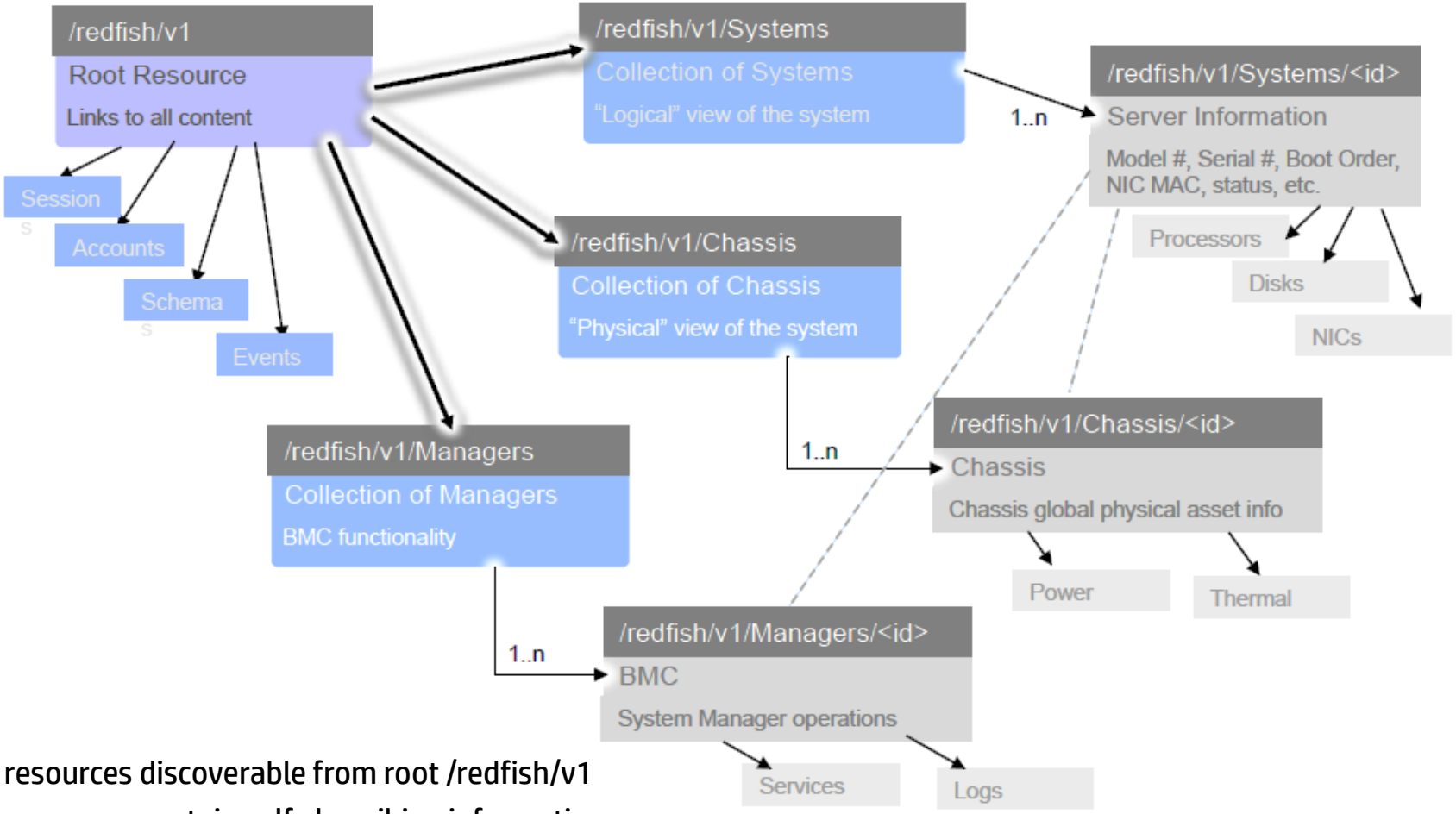
~50% of WWW APIs data



Redfish Data Model

- Root of service “/redfish/v1”
  - All resources linked from root using “href”
- Each resource has a type
  - Versioned, schema-backed
  - Standard or OEM extensions
  - Self-describing meta-data
- Collections to describe versatile hardware architectures
  - Stand-alone, multi-node, rack-level aggregated systems
- Major types
  - **ComputerSystem**: Logical view of the host (as seen by software)
  - **Chassis**: Physical view of the system (as seen by human)
  - **Manager**: Service processor – BMC subsystem

RedFish Resource Map



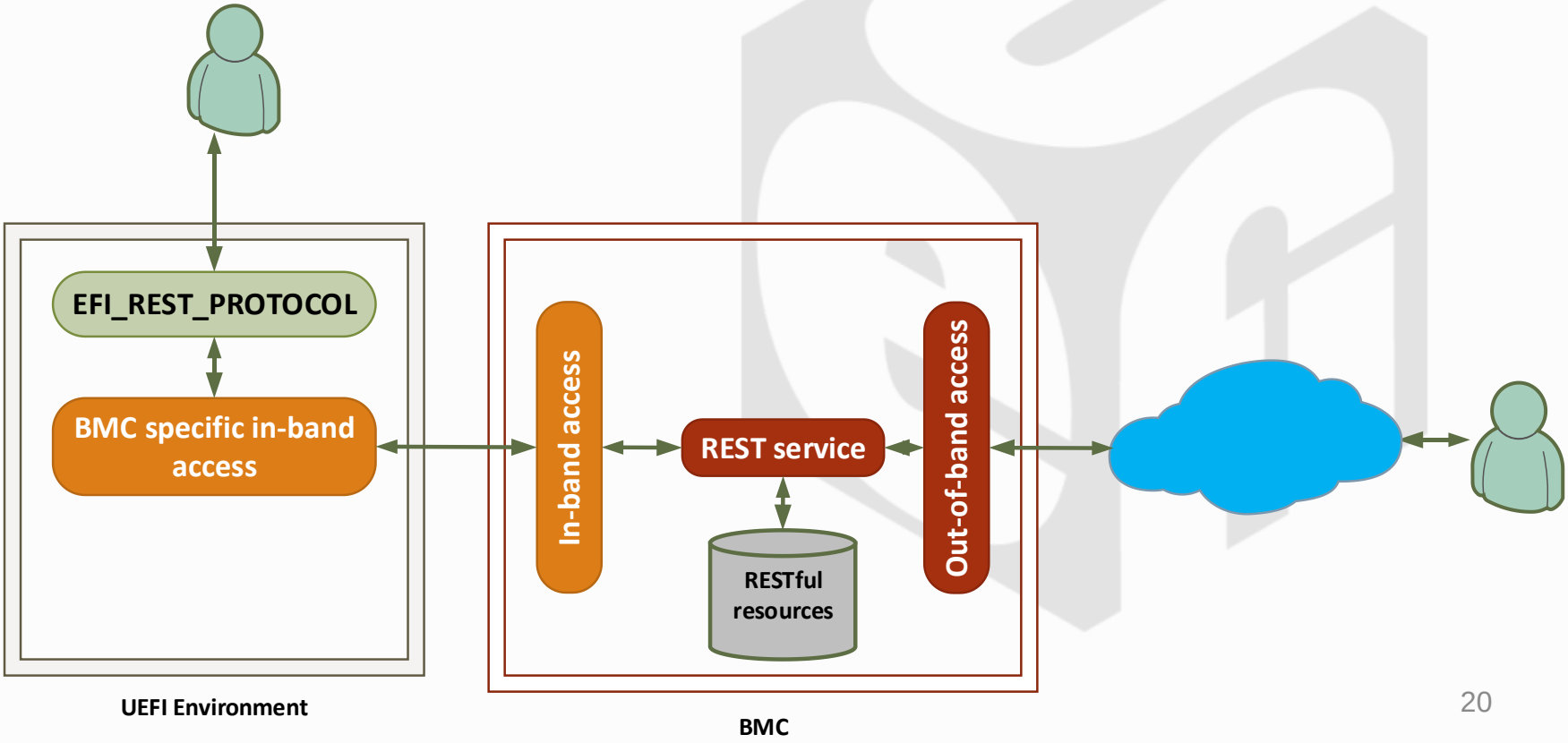
All resources discoverable from root /redfish/v1

All resources contain self-describing information

Source: http://dmtf.org/sites/default/files/DMTF-SPMF-Status-Presentation%203-9-15-2.pdf

UEFI 2.5: REST Protocol

- UEFI 2.5 EFI_REST_PROTOCOL and BMC Device Path
- Standard in-band access to a RESTful API, like Redfish
- Abstracts BMC-specific access methods from UEFI (proprietary)
- Uses EFI_HTTP_UTILITY_PROTOCOL to build/parse HTTP headers



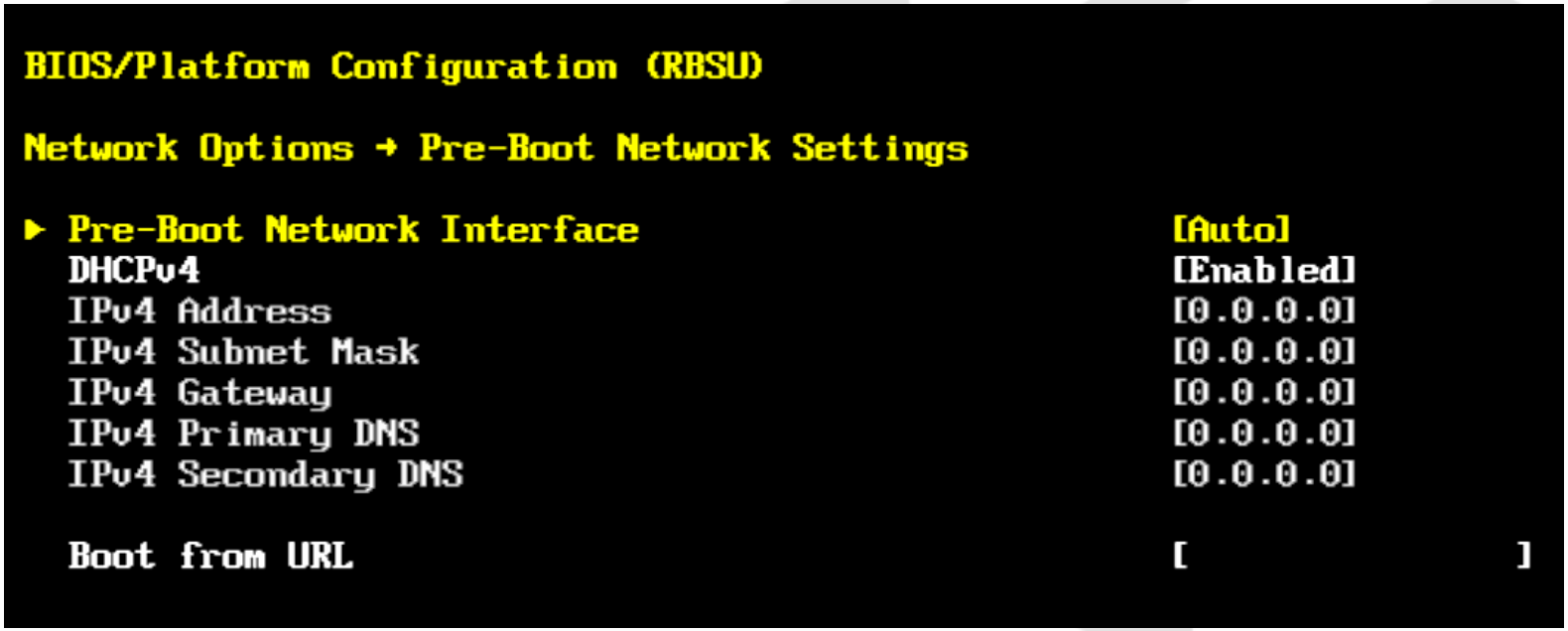
Putting it together:
Case Study of HP ProLiant Servers

UEFI Deployment solution on HP ProLiant Servers

- UEFI Network Stack Extensions
  - HTTP, FTP, DNS
  - “Boot from URL”
  - Target can be NBP or ISO image
- Embedded UEFI Shell
  - Integrates with the HP UEFI network stack
  - HP value-add commands for bare-metal deployment
- HP RESTful API
  - Accessible in-band (from OS) or out-of-band (iLO4 HTTPs)
  - HP OEM extensions including support for UEFI BIOS configuration and Secure Boot

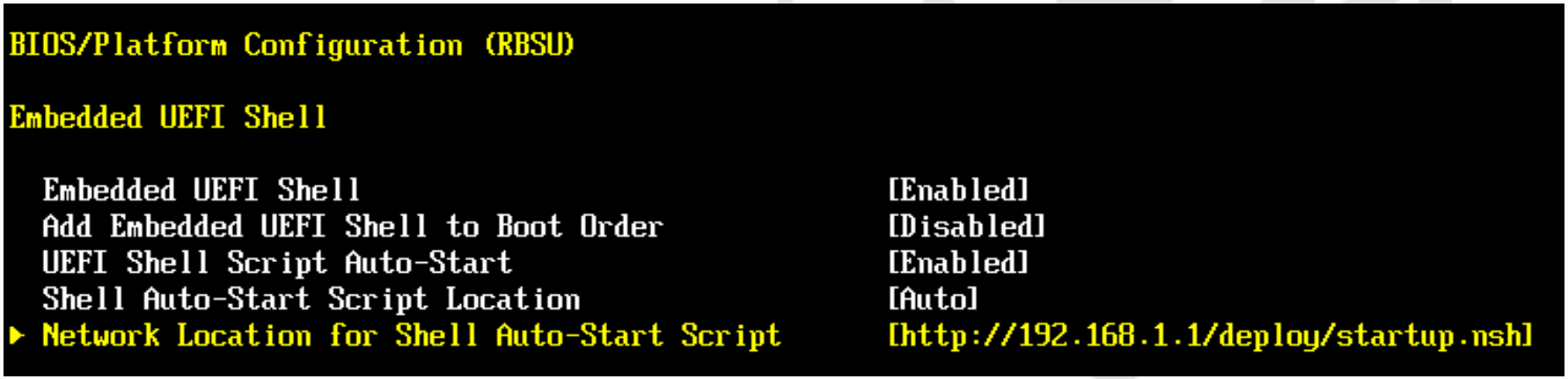
UEFI Network configuration

- Select pre-boot NIC (or auto select)
- Static / DHCP configuration
- Boot from URL configuration



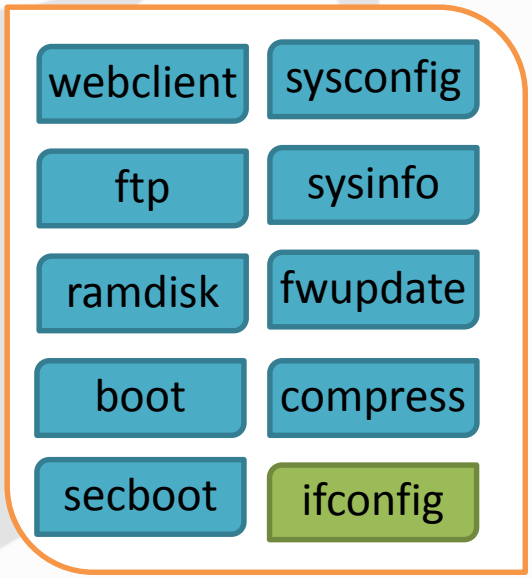
Embedded UEFI Shell

- Embedded in the BIOS image
- Fully documented and supported
- Dangerous commands removed (mm, hexedit, etc…)
- Configurable:
  - Enable / Disable
  - Policy for “startup.nsh” file location
    1. Local/virtual media
    2. URL for script location



Embedded UEFI Shell

- HP value-add commands for bare metal deployment
- **ramdisk** : Provision temporary staging locations, and mount ISO files
- **webclient** and **ftp :** Scriptable Network download/upload
- **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 components updates
- **compress** : ZIP/UNZIP archives
- **ifconfig** : Extensions to support DNS
- Commands to collect HP service/system logs



HP RESTful API

- RESTful API in iLO4
  - Main management API for iLO and iLO Chassis Manager based servers
  - Comprehensive inventory and server configuration
- Integrated with UEFI
  - UEFI BIOS settings configuration
  - Includes standard settings such as UEFI Boot Order and Secure Boot

HP RESTful API
UEFI Secure Boot Settings

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"
}
```

HP RESTful API
Get UEFI BIOS Settings

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


```
{
    "AcpiRootBridgePxm": "Enabled",
    "AcpiSlit": "Enabled",
    "AdjSecPrefetch": "Enabled",
    "AdminEmail": "",
    "AdminName": "",
    "AdminOtherInfo": "",
    "AdminPassword": null,
    "AdminPhone": "5555555",
    "AdvancedMemProtection": "AdvancedEcc",
    "AsrStatus": "Enabled",
    "AsrTimeoutMinutes": "10",
    "AssetTagProtection": "Unlocked",
    "AttributeRegistry": "HpBiosAttributeRegistryP89.1.0.40",
    "AutoPowerOn": "RestoreLastState",
    "BootMode": "Uefi",
    "BootOrderPolicy": "RetryIndefinitely",
    "ChannelInterleaving": "Enabled",
    "CollabPowerControl": "Enabled",
    "ConsistentDevNaming": "LomsOnly",
```

HP RESTful API
Set UEFI BIOS Settings

PATCH @ /rest/v1/systems/1/bios/settings

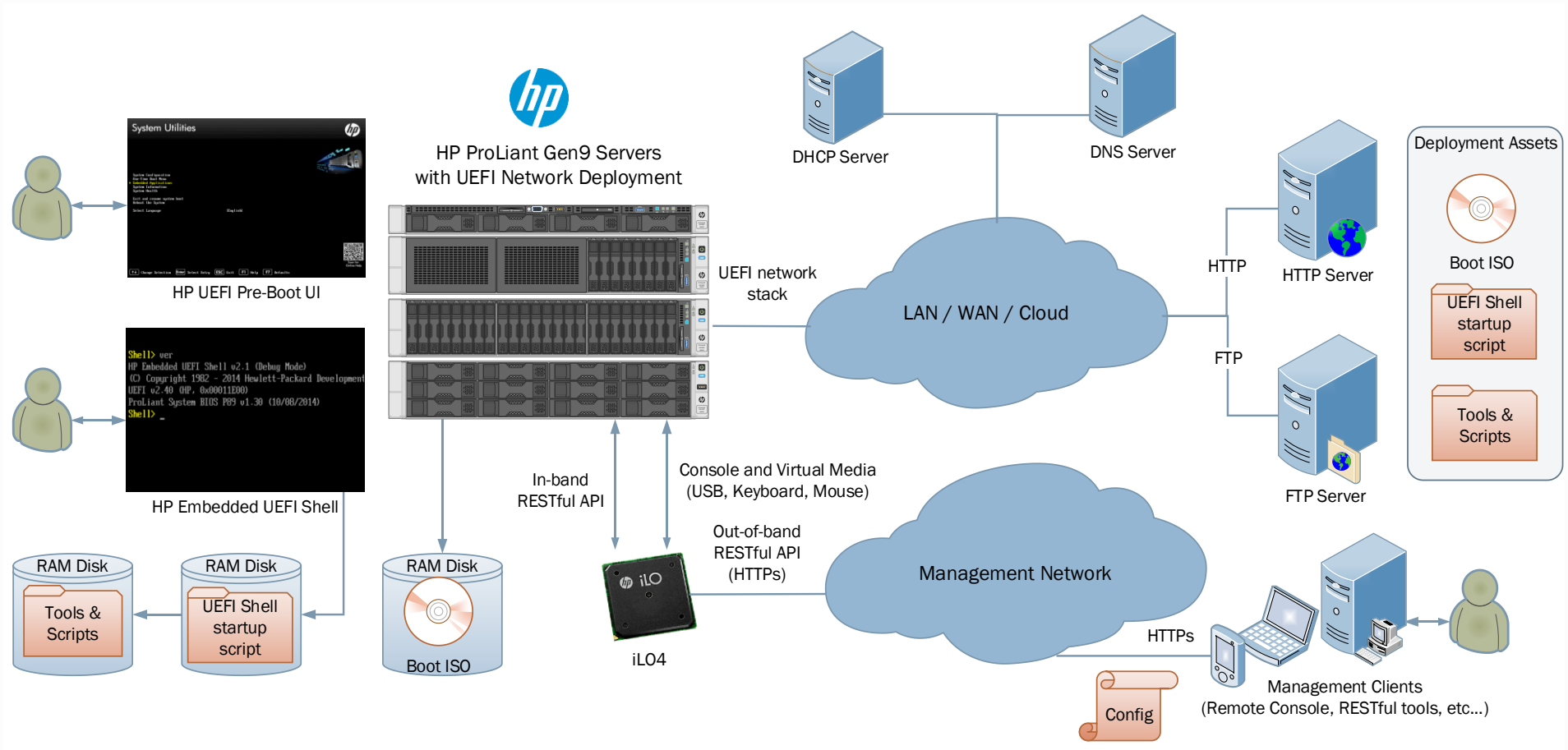
form-data | x-www-form-urlencoded | raw | binary | JSON (application/json)

```
1  {"AdminName":"Samer"}
```

Response in Body:


```
{
  "Messages": [
    {
      "MessageID": "iLO.0.10.SystemResetRequired"
    }
  ],
  "Name": "Extended Error Information",
  "Type": "ExtendedError.0.9.6"
}
```

UEFI Deployment solution on HP ProLiant Servers



Sample HPREST config script

```
# 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 to start 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 Shell
hprest set Boot/BootSourceOverrideEnabled=Once --selector ComputerSystem.
hprest set Boot/BootSourceOverrideTarget=UefiShell

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

Sample startup.nsh

```
# Create RAM Disk

ramdisk -c -s 512 -v MYRAMDISK -t F32
FS0:

# Download provisioning OS files

webclient -g http://192.168.1.1/deploy/efilinux.efi
webclient -g http://192.168.1.1/deploy/deploy.kernel
webclient -g http://192.168.1.1/deploy/deploy.ramdisk

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

References

- **UEFI 2.5 and ACPI 6.0 Specifications**
  - http://www.uefi.org/specs/
- **Redfish Specification**
  - http://www.redfishspecification.org/
- **UEFI on HP ProLiant Servers**
  - http://hp.com/go/proliant/uefi
- **Redfish-python github module**
  https://github.com/devananda/python-redfish

Thanks for attending the
UEFI Spring Plugfest 2015

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

presented by