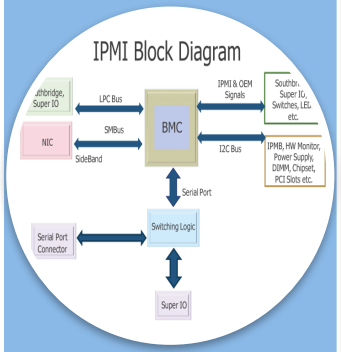
The Case for Redfish

Jeff Hilland
President, DMTF
CT Manageability, Hewlett Packard Enterprise DCIG
June 2017

Disclaimer

- The information in this presentation represents a snapshot of work in progress within the DMTF.
- This information is subject to change without notice. The standard specifications remain the normative reference for all information.
- For additional information, see the Distributed Management Task Force (DMTF) website.

The Status Quo



Inefficient architecture

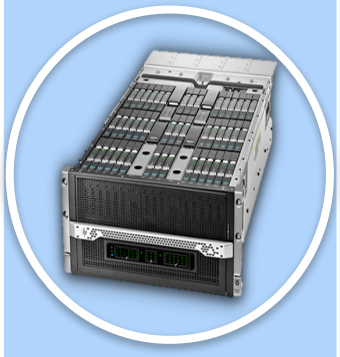
- Designed for 8-bit microarchitectures of the past
- Increased development cost from using multiple incompatible protocols & tools

High barrier to entry

- Protocols not human readable
- Significant expertise required to develop for legacy protocols
- Proprietary protocols and fragmentation from OEM extensions
- Lack of interoperability

Security Risks

- Not developed with security focus
- No security best practices deployed



Scaling Limitations

- Can't describe modern systems (i.e. multi-node)
- Current specs do not address scale data centers
- As scale increases, the need to monitor and manage efficiently increases exponentially

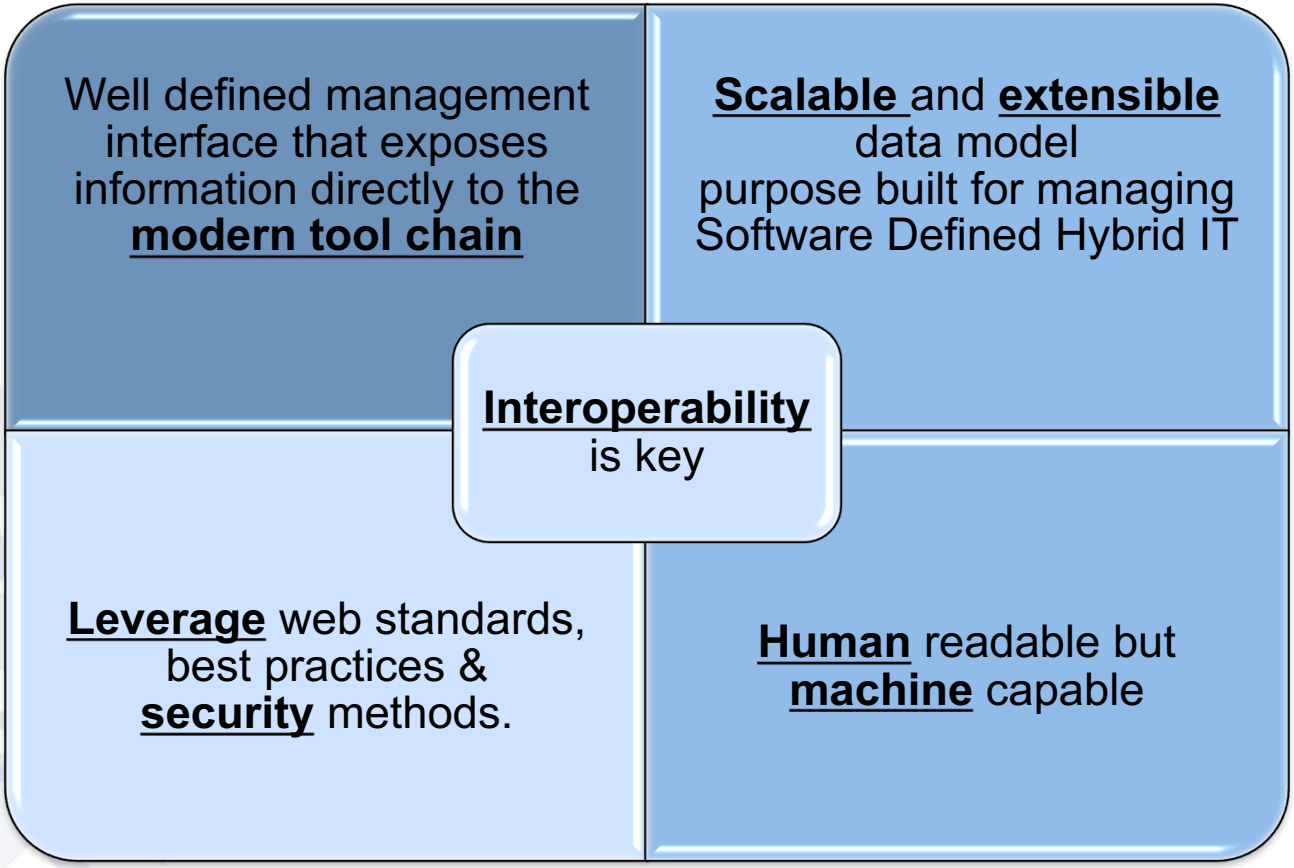
Outdated Tools

- Layers needed to adapt to the current tool chain.
- Special Utilities, libraries & reformatting needed to meet customer needs.
- Layers on layers
- Inefficiencies in representation and number of IOs

What we did to fix the Status Quo

- Bring forth a modern standard that advances capabilities started by IPMI, SMASH, DASH, etc.
- Build a consortium of industry leaders using an existing standards body
- Strive for broad adoption of specification across industry to meet growing customer demand
- Seed ecosystem with open source
- Unprecedented level of interoperability
- Expand scope to rest of Software Defined IT Infrastructure over time

How to Re-invent IT Management



A Hybrid IT Management Solution

Design Tenets

- Leverage common Internet / Web Services standards, other standards where appropriate
- Represent modern hardware designs (standalone to scale-out, current silicon, OCP)
- Does not require a PhD to design or use.
- Separation of protocol from data model, allowing them to be revised independently

Protocol Suite

- HTTPS / SSL: Primary data transport
- SSDP from uPnP: Service Discovery
- HTTP-based alert subscription
- Leverage OData v4

REST & JSON

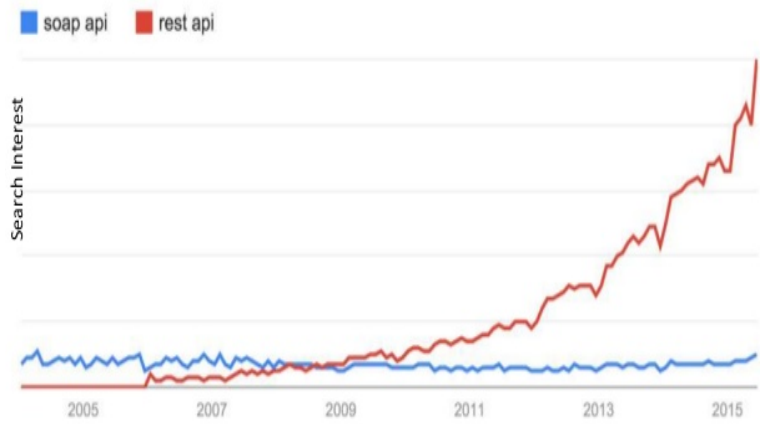
- Modern, standards-based
- Widely used for web services, software defined and public APIs
- Easy for IT professionals and amateurs to utilize

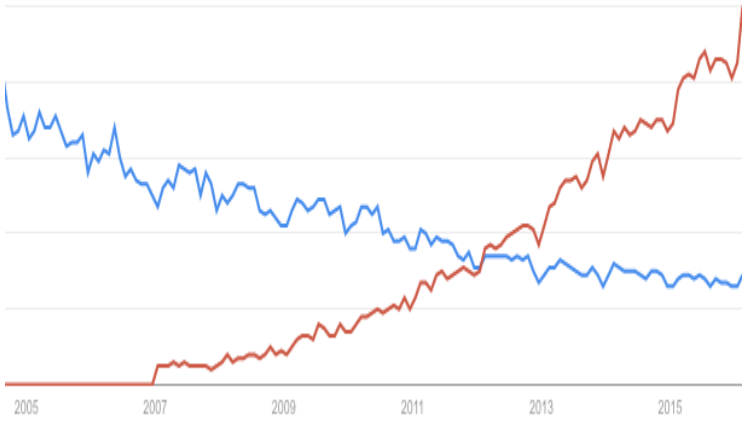
Data Model

- Schema-based, starting with CSDL & JSON Schema
  - Prepare to add schema language definitions as market changes
- An easy to use data model that a human can just read
- Create new modeling tenants to facilitate ease of design (inheritance by copy, polymorphism by union)

Why REST, HTTP and JSON?

- **REST:** The API architecture
  - Rapidly replacing SOAP
- **HTTPS:** The Web protocol
  - Well-understood by admins
  - Known security model
  - Known network configuration
- **JSON:** Modern data format
  - Human-readable
  - Simpler than XML
  - Modern language support
- The combination of language support and ubiquity of REST, HTTP and JSON means that IT management tasks can be performed using the same skill set and tool chain as all other IT and dev/ops tasks.





JSON vs XML

Scalable Platforms Management Forum (DMTF group that defines Redfish)

Co-Chairs: Jeff Autor (HPE), Mike Raineri (Dell)

SPMF Leadership Companies

BROADCOM, CISCO, DELL, ERICSSON, Hewlett Packard Enterprise, inspur, intel, Lenovo, SUPERMICRO, VERTIV, vmware

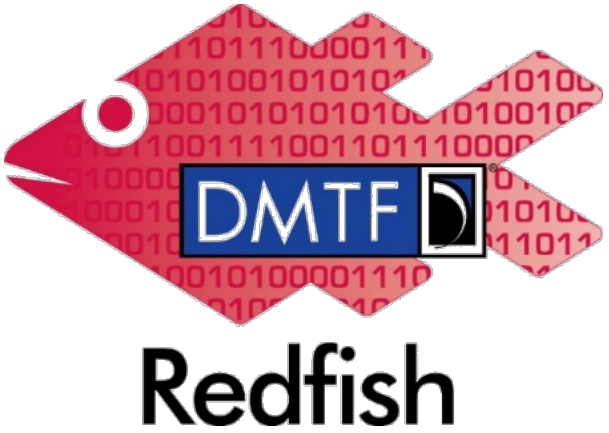
SPMF Supporting Companies

American Megatrends, Inc., Fujitsu, Huawei, IBM, Insyde Software Corp., Mellanox Technologies, Microsemi, NetApp, Oracle, OSIsoft, LLC, Quanta Computer, Toshiba, Western Digital Corporation

SPMF Industry Alliance Partners & efforts

OpenCompute Project – Collaborating on profile definition
UEFI – Collaborating on Firmware Update and Host Interface work
SNIA – Collaborating on Storage modeling / alignment between SNIA SSM and Redfish
TGG – Pursuing relationship to work on Power/Cooling (existing DMTF Alliance Partner)
IETF – working on Switch modeling (no official alliance)

What is Redfish?



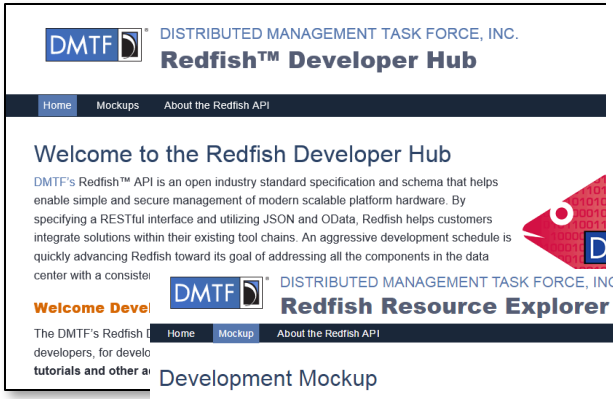
- **Industry Standard Software Defined Management for Converged, Hybrid IT**
  - HTTPS in JSON format based on OData v4
  - Schema-backed but human-readable
  - Equally usable by Apps, GUIs and Scripts
  - Extensible, Secure, Interoperable
- **Version 1 focused on Servers**
  - A secure, multi-node capable replacement for IPMI-over-LAN
  - Represent full server category: Rackmount, Blades, HPC, Racks, Future
  - Intended to meet OCP Remote Machine Management requirement
- **Expand scope over time to rest of IT infrastructure**
  - Additional features coming out approximately every 4 months
  - Working with SNIA to cover more advanced Storage (Swordfish)
  - Working with The Green Grid to cover Facilities (Power/Cooling)
  - Work with the IETF to cover some level of Ethernet Switching

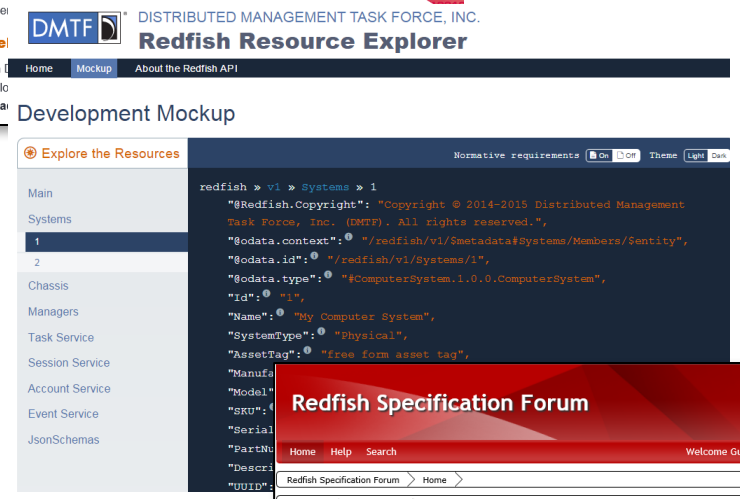
Timeline of Redfish™ Specification

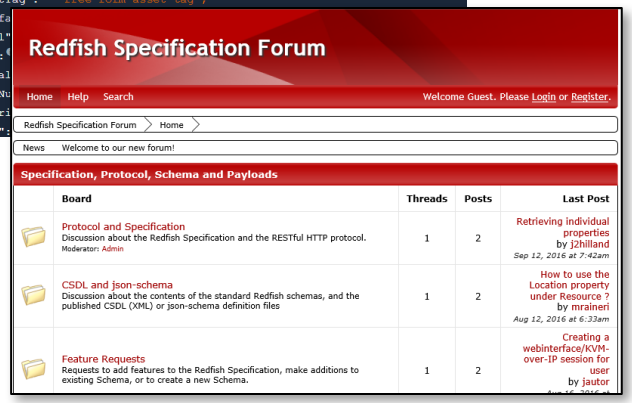
- The DMTF Redfish technology
  - Sep 2014: SPMF Formed in DMTF.
    - Released multiple work-in-progress for public feedback
  - Aug 2015: Redfish Specification with base models (v1.0)
  - May 2016: Models for BIOS, disk drives, memory, storage, volume (2016.1)
  - Aug 2016: Models for endpoint, fabric, switch, PCIe device, zone, software/firmware inventory & update (2016.2)
  - Dec 2016: Adv. communications devices (multi-function NICs), host interface (KCS replacement), privilege mapping (2016.3)
  - May 2017: Composability (2017.1)
- Alignment with other standard organizations
  - Aug 2016: SNIA releases model for network storage services (Swordfish)
  - DMTF created work registers with UEFI Forum, TGG and OCP
  - Working with IETF to create a programmatic Redfish mapping

Redfish Developer Hub: redfish.dmtf.org

- Resources
  - Schema Index
  - Specifications
  - GitHub for Redfish Tools
  - Registries
  - Other Documentation
- Mockups
  - Simple Rack-mounted Server
  - Bladed System
  - Proposed OCP Redfish Profile
  - More being added
- Education/Community
  - Redfish User Forum
  - Whitepapers, Presentations
  - YouTube shorts & Webinars







Redfish Task Forces

Task forces created when there is sufficient need or interest

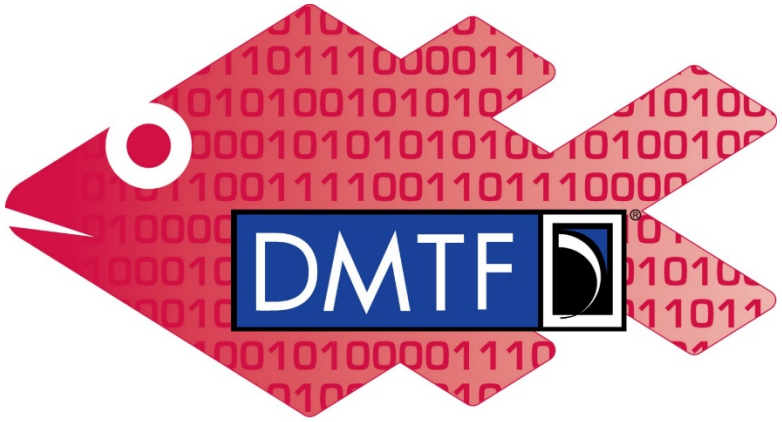
Task Forces	Product	Comment
Host Interface	Interface for in-band entity to access the Redfish service	Complete
Firmware Update	Resource models for performing firmware updates	Complete
Storage	Resource models for managing local storage devices	Complete
Advanced Comm Device	Resource models for managing network ports	Complete
Privilege Mapping	Model extensions for expressing user privilege	Complete
Composability	Resource models for composing resource	Model released, ongoing
Network Infrastructure	Resource models for managing an Ethernet Switch model, derived from YANG models	WIP released, ongoing
Tools	Open source for using and developing Redfish models	Released 10 tools to date
Interoperability	Develop interoperability tools & hold plugfests	Ongoing

Tools Task Force: Redfish Tools Description

SPMF Tools TF open source tools to enable Redfish http://www.dmtf.org/standards/opensource

	Tool	Description
Extend	CSDL Validator	Validates the CSDL conforms to Redfish requirements
	CSDL-to-JSON schema convertor	Generates json-schema files from CSDL
	Document Generator	Generates documentation from json-schema
Working Svc	Mockup Server	Exposes a mockup as a static HTTP service (GETs only)
	Mockup Creator	Creates a mockup from a Redfish service
	Profile Simulator	Dynamic simulator of the proposed Redfish profile for OCP
	Interface Emulator	Dynamic simulator which can rapid
Test	Service Validator	Validates a Redfish service is conformant
	Service Conformance Tool	Verifies the conformance of a Redfish service to assertions extracted from the Redfish Specification
Client	Command Line (redfishtool)	A command line tool for interacting with a Redfish service (similar to ipmitool)
	Python Utility & Python Library	A Command line tool with UI and python libraries for interacting with Redfish services

Thank you!



Redfish

Backup Material

How simple is Redfish?

Example Python code to retrieve serial number from a server:

```
rawData = urllib.urlopen('http://192.168.1.135/redfish/v1/Systems/1')
jsonData = json.loads(rawData)
print( jsonData['SerialNumber'] )
```

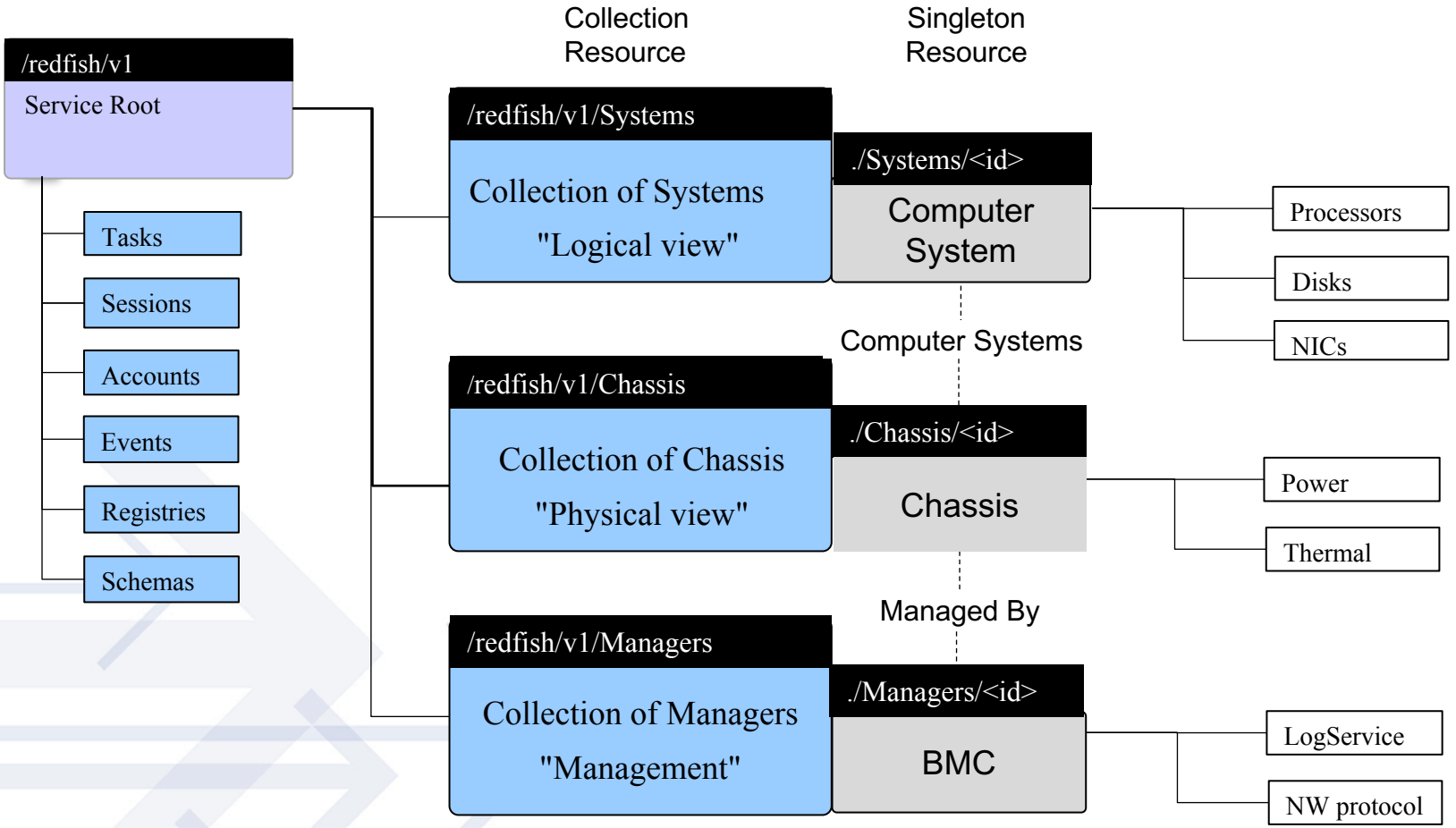
Output is:

```
1A87CA442K
```

Three lines of code: point to the resource, get the data, print the serial number.

***Example uses Redfish ComputerSystem resource**

Redfish Resource Map

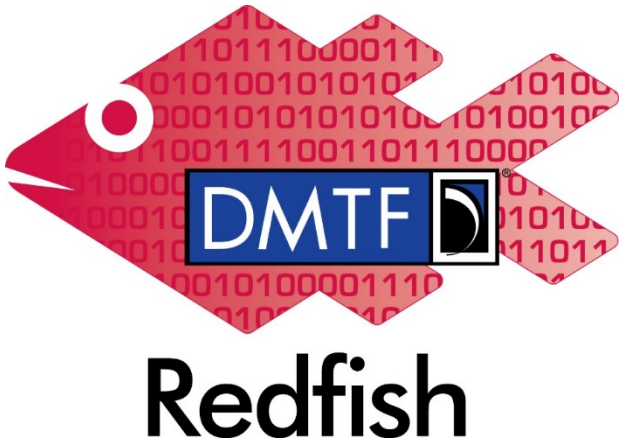


GET http://<ip-addr>/redfish/v1/Systems/{id}/Processors/{id}

Use the Redfish Resource Explorer (redfish.dmtf.org) to explore the resource map

Redfish Releases 2016

- **2016.1**:
  - BIOS Configuration
  - Memory (DIMM) Inventory
  - Secure Boot
  - Advanced Storage (SAS, SATA, RAID, NVMe)
- **2016.2**
  - PCIe Devices
  - Switched Fabrics
  - Persistent Memory Management
  - Installed Firmware and Firmware Update
  - ActionInfo to describe Action Parameters
- **2016.3**
  - Advanced Communications Devices (multi-function NICs)
  - Redfish Host Interface (KCS replacement)
  - Resource and Property-level Privilege Mapping
- **2017.1**
  - Composability



Redfish v1.0 Feature Set

Retrieve "IPMI class" data

- Basic server identification and asset info
- Health state
- Temperature sensors and fans
- Power supply, power consumption and thresholds

Basic I/O infrastructure data

- Host NIC MAC address(es) for LOM devices
- Simple hard drive status / fault reporting

Discovery

- Service endpoint (network-based discovery)
- System topology (rack/chassis/server/node)

Security

- Session-based leverages HTTPS

Perform Common Actions

- Reboot / power cycle server
- Change boot order / device
- Set power thresholds

Access and Notification

- Serial console access via SSH
- Alert / event notification method(s)
- Event Log access method(s)

BMC infrastructure

- View / configure BMC network settings
- Manage local BMC user accounts

Working on more…

Network Infrastructure task force: YANG to Redfish

- Enable converged infrastructure management
  - One interface (one tool chain) to manage compute, storage and network
  - Switches have platform components in common with servers and storage
  - Network Functions Virtualization (NFV) will need common manageability for compute and networking
- DMTF wants to leverage the networking industry's expertise
  - YANG is the basis for general network industry manageability
  - Large body of existing YANG work
  - Model driven approach to network management