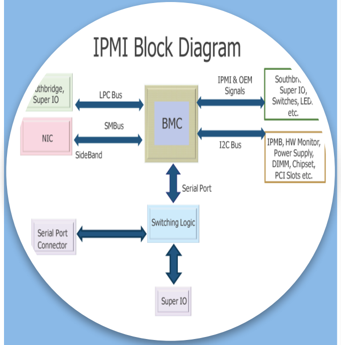
Introduction to Redfish

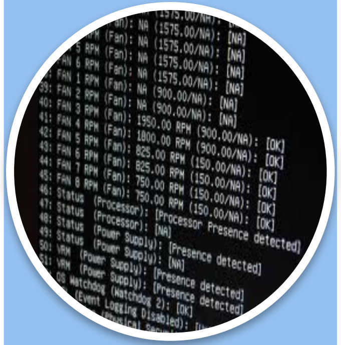
December, 2017

The Status Quo



Inefficient architecture

- Designed for 8-bit microarchitectures of the past
- Increased development cost from using multiple incompatible protocols & tools
- In-band access differs from out-of-band



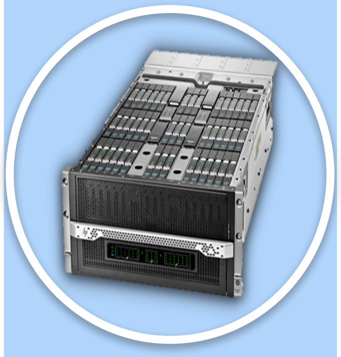
```
Fan): NA (1575.00/NA): [NA]
5 RPM (Fan): NA (1575.00/NA): [NA]
FAN 6 RPM (Fan): NA (1575.00/NA): [NA]
: FAN 1 RPM (Fan): NA (1575.00/NA): [NA]
39: FAN 2 RPM (Fan): NA (900.00/NA): [NA]
40: FAN 3 RPM (Fan): NA (900.00/NA): [NA]
41: FAN 4 RPM (Fan): 1950.00 RPM (900.00/NA): [OK]
42: FAN 5 RPM (Fan): 1800.00 RPM (900.00/NA): [OK]
43: FAN 6 RPM (Fan): 825.00 RPM (150.00/NA): [OK]
44: FAN 7 RPM (Fan): 825.00 RPM (150.00/NA): [OK]
45: FAN 8 RPM (Fan): 750.00 RPM (150.00/NA): [OK]
                     750.00 RPM (150.00/NA): [OK]
46: Status (Processor): [Processor Presence detected]
47: Status (Processor): [NA]
48: Status (Power Supply): [Presence detected]
49: Status (Power Supply): [NA]
50: VRM (Power Supply): [Presence detected]
51: VRM (Power Supply): [Presence detected]
 OS Watchdog (Watchdog 2): [OK]
 (Event Logging Disabled): [
 (Physical Secu
```

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Outdated Tools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

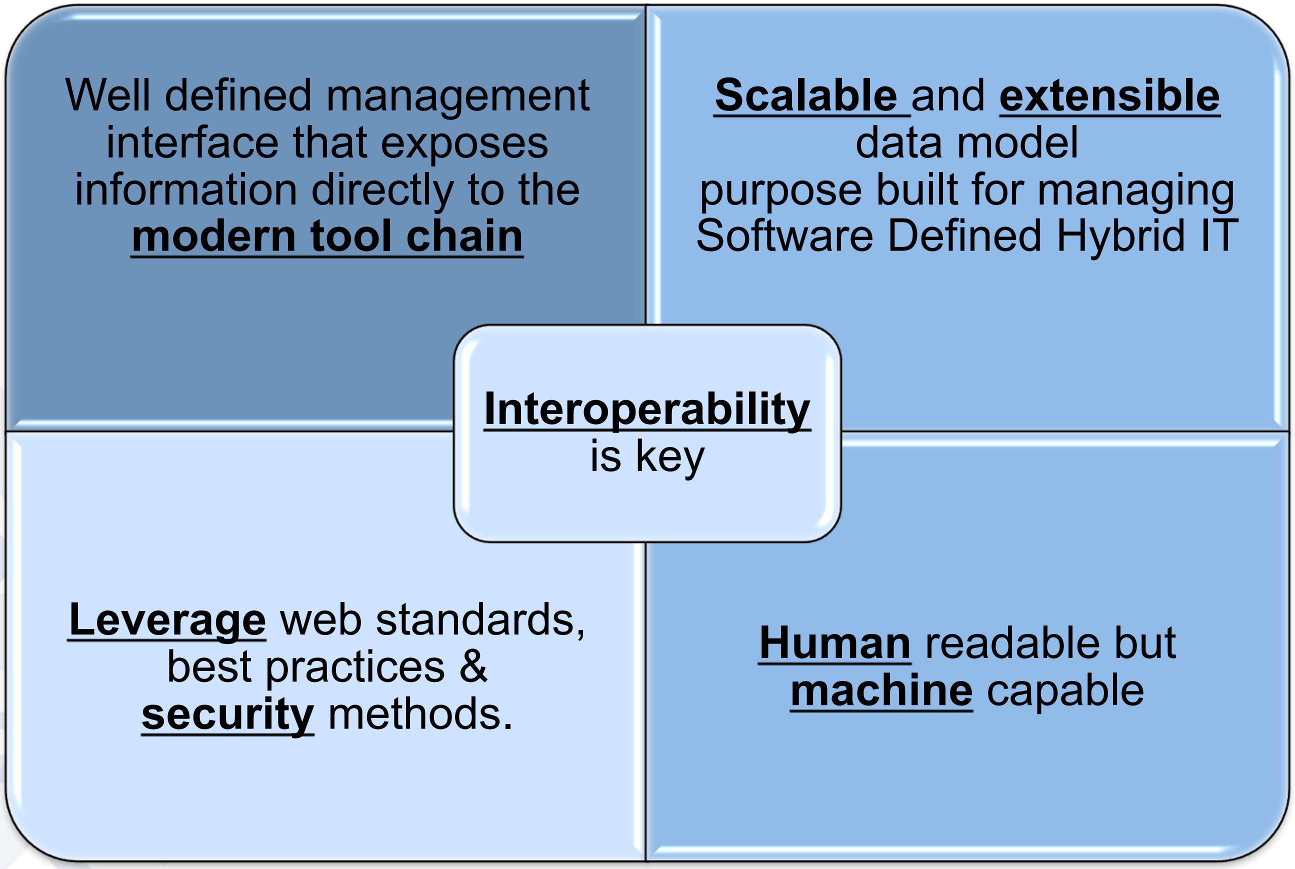
Strive for broad adoption of specification across industry to meet growing customer demand

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



Well defined management interface that exposes information directly to the **modern tool chain**

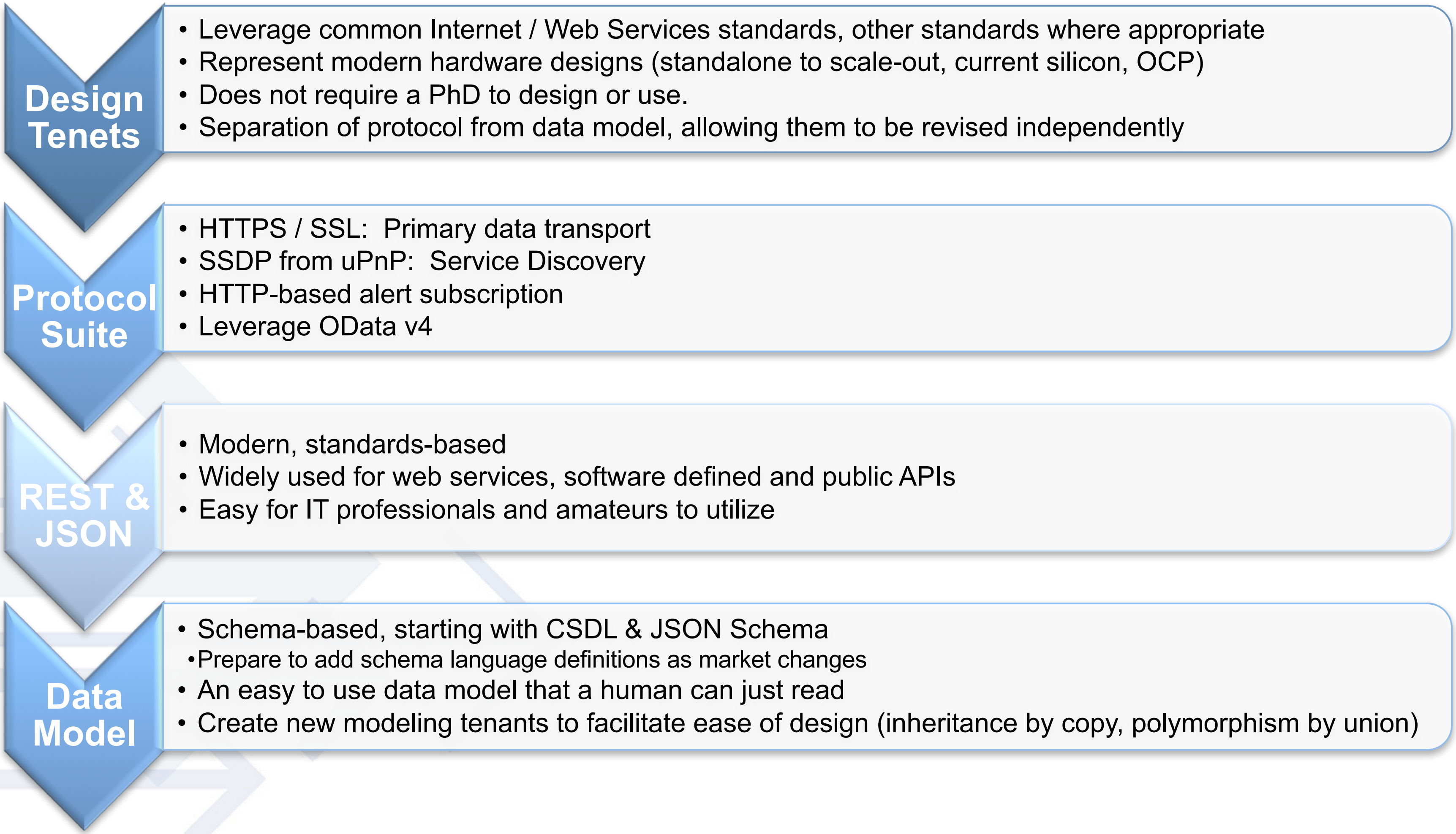
Scalable and **extensible** data model purpose built for managing Software Defined Hybrid IT

Interoperability is key

Leverage web standards, best practices & **security** methods.

Human readable but **machine** capable

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

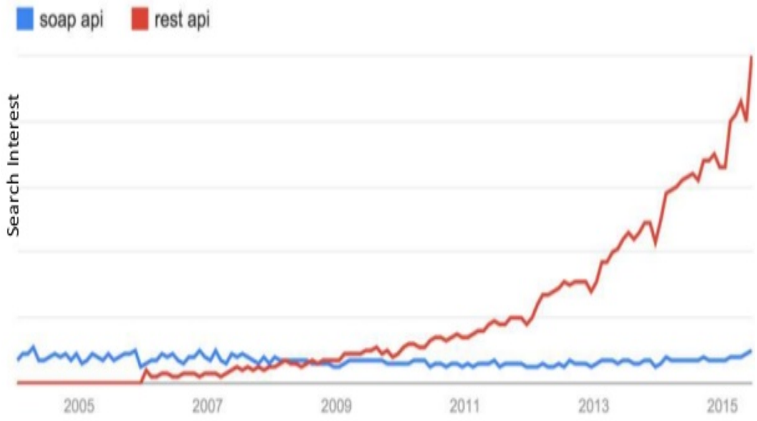
- Modern, standards-based
- Widely used for web services, software defined and public APIs
- Easy for IT professionals and amateurs to utilize

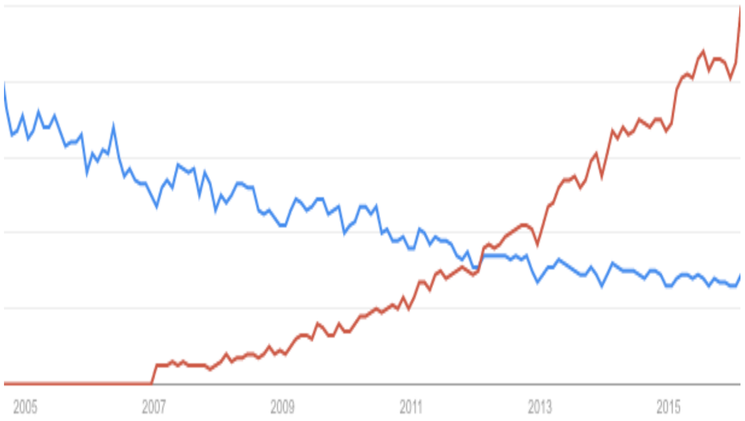
Data Model

- 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

intel · Lenovo · SUPERMICRO · VERTIV · vmware

SPMF Supporting Companies

American Megatrends, Inc., ARM, Inc., Artesyn Embedded Systems, Cray, Inc., Flex, Fujitsu, Huawei, IBM, Insyde Software Corp., Mellanox Technologies, Microsemi, MPSTOR, NetApp, OSIsoft, LLC, Quanta Computer, Solarflare Communications Toshiba, Western Digital Corporation

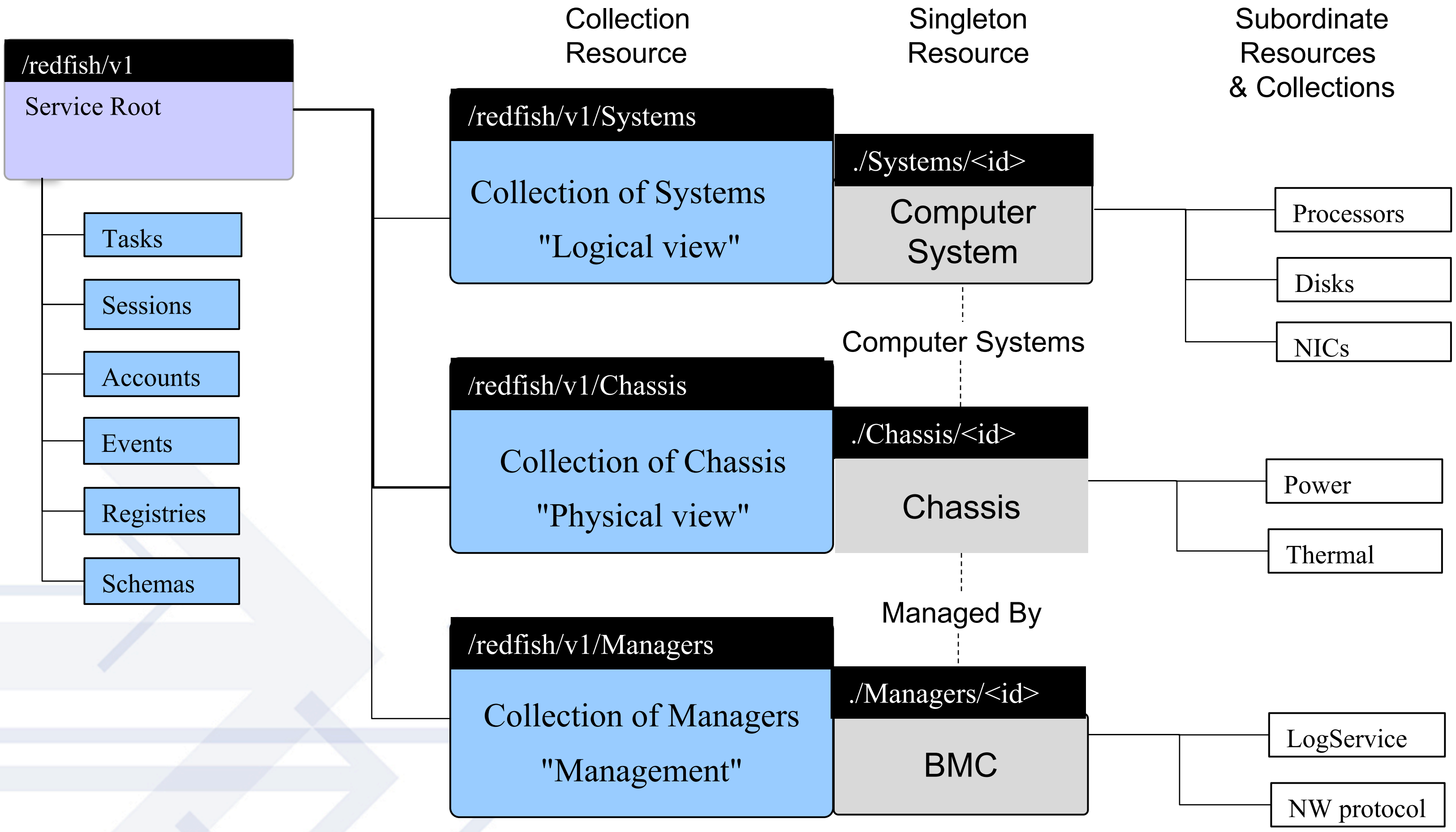
SPMF Industry Alliance Partners & efforts

Open Compute 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)
ASHRAE – American Society of Heating, Refrigerating and Air Conditioning Engineers
BBF – Broadband Forum

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 & ASHRAE to cover Facilities (Power/Cooling)
  - Work with the IETF to cover some level of Ethernet Switching

Redfish Resource Map (simplified)



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

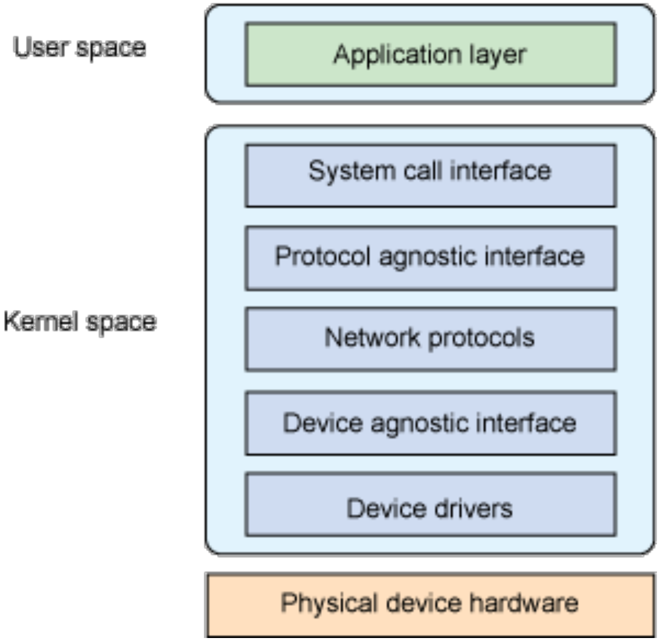
Use the Redfish Resource Explorer (redfish.dmtf.org) to explore the resource map

Timeline of Redfish® Specification

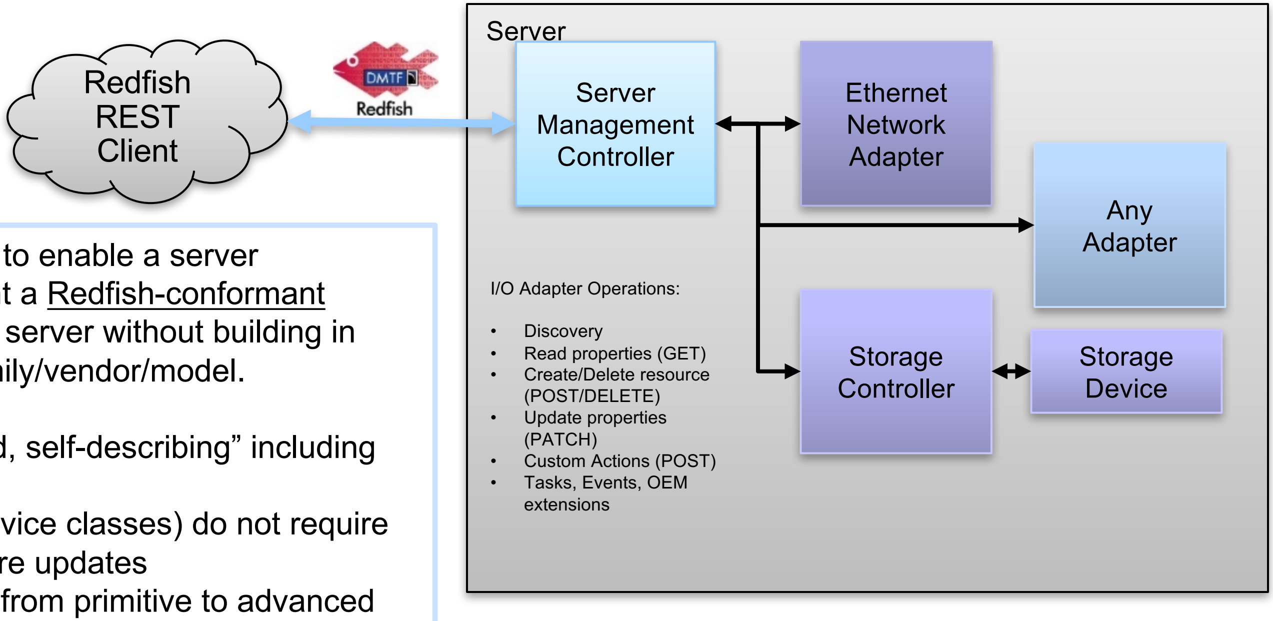
- The DMTF Redfish technology
  - Sep 2014: SPMF Formed in DMTF.
    - Released multiple work-in-progress for public feedback
  - Aug 2015: Redfish Specification with base models (v1.0)
  - May 2016: Models for BIOS, disk drives, memory, storage, volume (2016.1)
  - Aug 2016: Models for endpoint, fabric, 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)
    - WIP for Telemetry
  - Aug 2017: Location, errata (2017.2)
    - WIPs for Ethernet Switching, DCIM, OCP & Profiles
- Alignment with other standard organizations
  - Aug 2016: SNIA releases first model for network storage services (Swordfish)
  - Working with IETF to create an open YANG Redfish mapping algorithm
  - DMTF created work registers with UEFI Forum, TGG, OCP, ASHRAE, Broadband Forum, ETSI-NFV, ODCA for work on applying Redfish

Host Interface

- Replacement for IPMI KCS, etc.
- Exposes a NIC from Management Controller to OS
  - SMBIOS records provide information to allow kernel access
- Same access in-band as out-of-band
  - Kernel mode or user mode accessible
  - Encouraging OS vendors to begin consuming Redfish data.
  - This means you can get to the iLO homepage from the OS.
  - This means you can write your tools for the iLO homepage or Redfish and run them in the host OS.
  - Anything that accesses the out of band can be run on the host OS to access to local management subsystem.



Redfish Device Enablement: PLDM Redfish Providers



I/O Adapter Operations:

- Discovery
- Read properties (GET)
- Create/Delete resource (POST/DELETE)
- Update properties (PATCH)
- Custom Actions (POST)
- Tasks, Events, OEM extensions

PMCI WG developing a standard to enable a server Management Controller to present a Redfish-conformant management of I/O Adapters in a server without building in code specific to each adapter family/vendor/model.

- Support adapter “self-contained, self-describing” including value-add (OEM) properties
- New managed devices (and device classes) do not require Management Controller firmware updates
- Support a range of capabilities from primitive to advanced devices (lightweight/low bandwidth options)
- Leveraging PLDM, a provider architecture is being specified that can binary encode the data in a small enough format for devices to understand and support.
- MC acts as a broker to encode/decode the data to/from the provider
- PLDM works over I2C & PCIe VDM. Additional mappings under consideration.

Interoperability Profiles: Goals

- An “Interoperability Profile” provides a common ground for Service implementers, client software developers, and users
  - A profile would apply to a particular category or class of product (e.g. “Front-end web server”, “NAS”, “Enterprise-class database server”)
  - It specifies Redfish implementation requirements, but **is not** intended to mandate underlying hardware/software features of a product
  - Provides a target for implementers to meet customer requirements
  - Provide baseline expectations for client software developers utilizing Redfish
  - Enable customers to easily specify Redfish functionality / conformance in RFQs
- Create a machine-readable Profile definition
  - Document must be human-readable
  - Can be created by dev/ops personnel and non-CS professionals
- Enable authoring of Profiles by DMTF, partner organizations, and others
- Create open source tools to document and test conformance

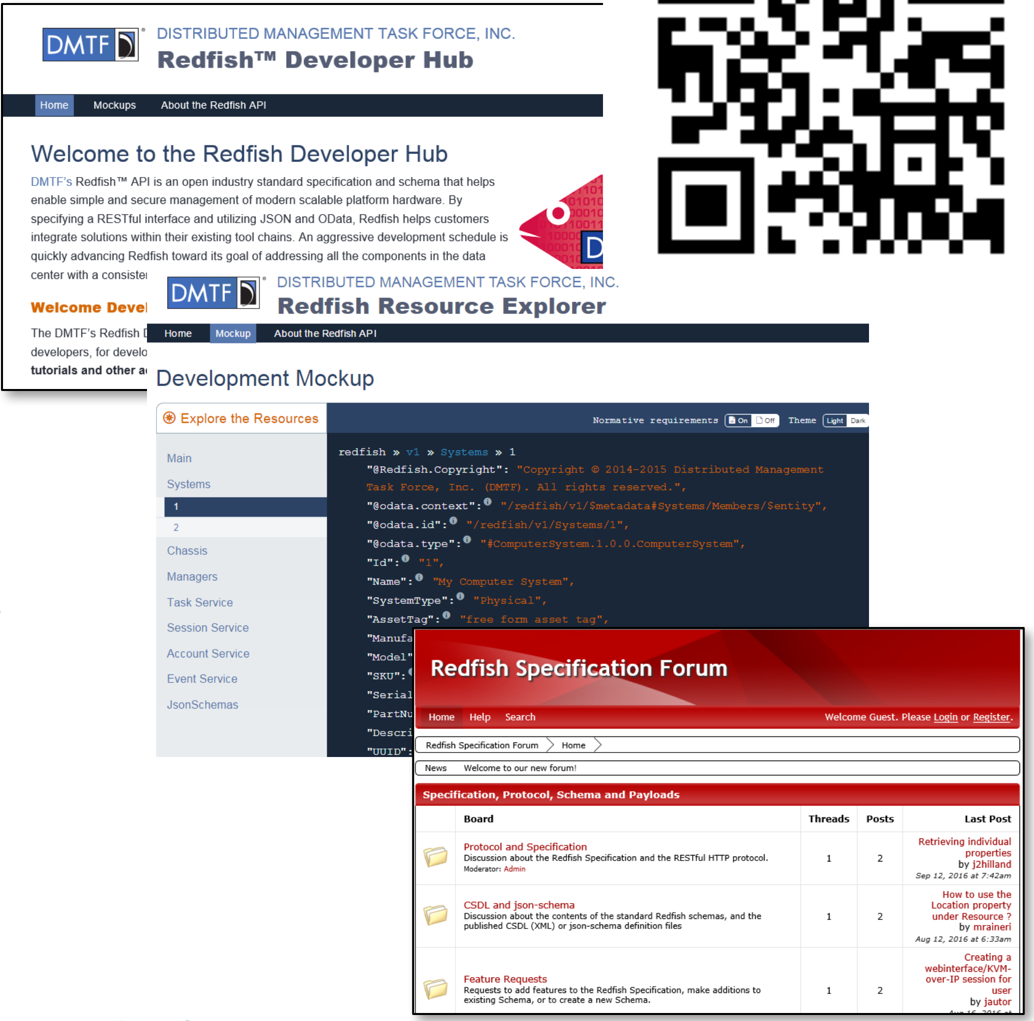
Tools Task Force: Redfish Tools Description

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
	YANG to Redfish	Converts a YANG model into a set of Redfish CSDL files, enabling Ethernet switching standard access via Redfish
	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	Emulate a Redfish interface statically or dynamically.
Test	Service Validator	Validates a Redfish service is conformant
	JSON Schema Response Validator	Validates any JSON resource against DMTF provided JSON schemas
	Reference Checker	Validates the reference URLs in CSDL files
	Use Case Checker	Collection of tools to validate common use cases for Redfish Services.
	Service Conformance Tool	Verifies conformance of a Redfish service to assertions in the Redfish Specification
Client	CLI (redfishtool)	A command line tool for interacting with a Redfish service (similar to ipmitool)
	Event Listener	A lightweight HTTPS server that can be executed to read and record events from a Redfish Service
	C Library (libRedfish)	C libraries for interacting with Redfish services
	Python Utility & Library	A Command line tool with UI and python libraries for interacting with Redfish services

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



DISTRIBUTED MANAGEMENT TASK FORCE, INC.
Redfish™ Developer Hub

Welcome to the Redfish Developer Hub

DMTF's Redfish™ API is an open industry standard specification and schema that helps enable simple and secure management of modern scalable platform hardware. By specifying a RESTful interface and utilizing JSON and OData, Redfish helps customers integrate solutions within their existing tool chains. An aggressive development schedule is quickly advancing Redfish toward its goal of addressing all the components in the data center with a consiste

Welcome Devel

The DMTF's Redfish D developers, for develo **tutorials and other a**

DISTRIBUTED MANAGEMENT TASK FORCE, INC.
Redfish Resource Explorer

Development Mockup

```
redfish » v1 » Systems » 1
    "@Redfish.Copyright": "Copyright © 2014-2015 Distributed Management
    Task Force, Inc. (DMTF). All rights reserved.",
    "@odata.context": "/redfish/v1/$metadata#Systems/Members/$entity",
    "@odata.id": "/redfish/v1/Systems/1",
    "@odata.type": "#ComputerSystem.1.0.0.ComputerSystem",
    "Id": "1",
    "Name": "My Computer System",
    "SystemType": "Physical",
    "AssetTag": "free form asset tag",
    "Manufa
    "Model"
    "SKU":
    "Serial
    "PartNu
    "Descri
    "UUID":
```

Redfish Specification Forum

Welcome Guest. Please Login or Register.

Redfish Specification Forum > Home

News Welcome to our new forum!

Specification, Protocol, Schema and Payloads

Board	Threads	Posts	Last Post
Protocol and Specification Discussion about the Redfish Specification and the RESTful HTTP protocol. Moderator: Admin	1	2	Retrieving individual properties by j2hilland Sep 12, 2016 at 7:42am
CSDL and json-schema Discussion about the contents of the standard Redfish schemas, and the published CSDL (XML) or json-schema definition files	1	2	How to use the Location property under Resource ? by mraineri Aug 12, 2016 at 6:33am
Feature Requests Requests to add features to the Redfish Specification, make additions to existing Schema, or to create a new Schema.	1	2	Creating a webinterface/KVM-over-IP session for user by jautor

Thank you!

Redfish

www.dmtf.org

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 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)

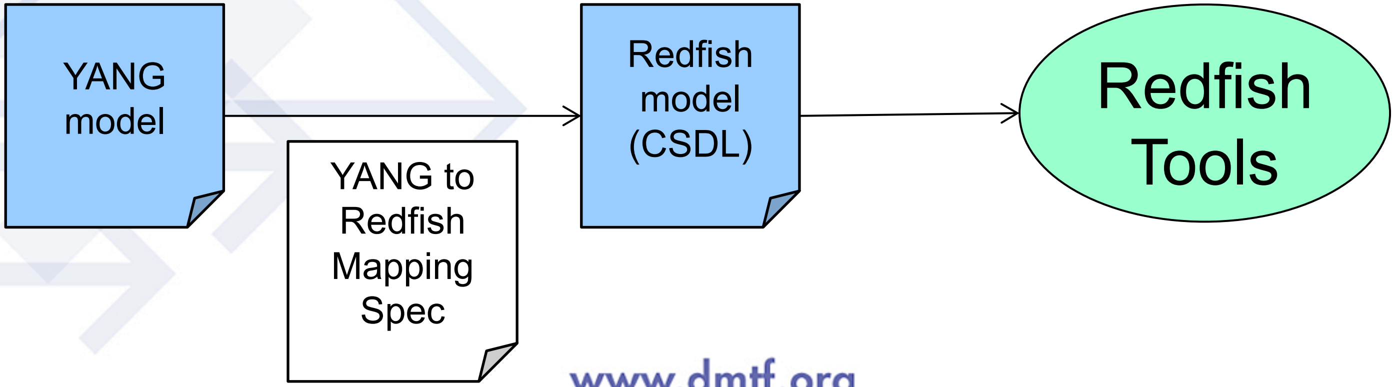
BMC infrastructure

- 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



YANG model

YANG to Redfish Mapping Spec

Redfish model (CSDL)

Redfish Tools