



YANG Device Profile for Redfish Network Management

**Joe White, David Black, John Leung
Distributed Management Task Force**



Disclaimer

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



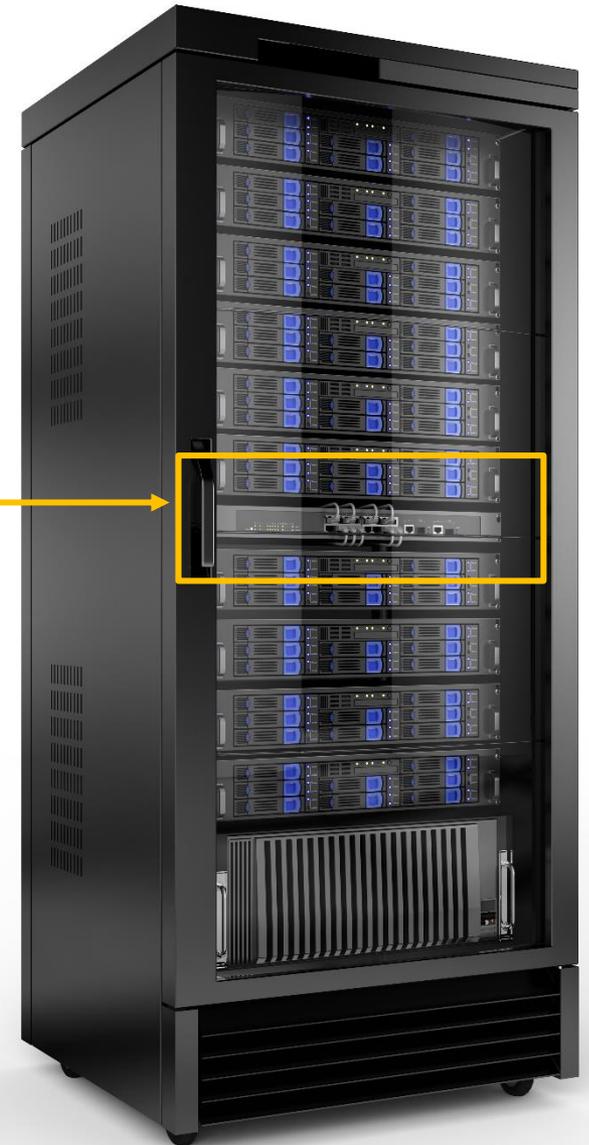


Overview

- Redfish standard: Converged Infrastructure (CI) management interface
 - CI = rack-integrated compute, network and storage for data centers
 - CI data center networks (e.g., top of rack switches) are crucial
- Redfish network management – based on YANG module reuse
 - Translate YANG to Redfish interface schema
- Our (initial) goal: baseline data center switch/router device profile for top-of-rack CI switches
 - List of applicable YANG modules
 - Structure for how modules go together to manage a top-of-rack CI switch
- Result: IETF YANG device profile for use by IETF, DMTF, etc.
 - Device profile should not be not Redfish-specific
 - Looking for help and guidance

What is Converged Infrastructure?

- Server, storage and networking:
 - Assembled into a larger system
 - Managed together via unified management
 - Usually part of an orchestrated solution
- One to four network switches per rack
- Datacenter grows by adding racks





Redfish™ : Why a New Interface?



- Market shifting to scale-out solutions
 - Sea of simple servers; reliability via software
- Customers exhausting existing server management interface (IPMI)
 - IPMI = Intelligent Platform Management Interface
 - Inadequate security, limited functionality and hard to extend
- Customer requirements:
 - Single interface for managing all datacenter platforms and devices
 - Expect APIs to use cloud/web protocols, structures and security model
- DMTF released Redfish 1.0.0 in Aug 2015
 - v1.0.0 included interface spec and initial models. Released Interface Spec v1.1 on Dec 2016. Released three supplemental models in 2016.
 - Standard organizations are aligning to Redfish (SNIA, OCP, TGG, and UEFI)
 - Redfish Forum has 24 member companies, including most major server vendors

SNIA = Storage Network Industry Association
TGG = The Green Grid
UEFI = Unified Extensible Firmware Interface
OCP = Open Compute Project



Why Redfish for Networking?

Completes the converged infrastructure management API story

- A single interface protocol to manage compute, storage and network
 - CI will need common manageability for compute and networking
- Network device management reuses Redfish hardware component models
 - Chassis, thermal, power, fan, embedded compute, etc.
- Results in a common interface for inventory, control, etc.



Redfish

Aug 2015



Swordfish

Aug 2016



YANG



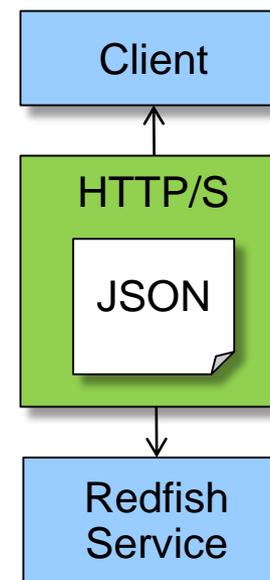
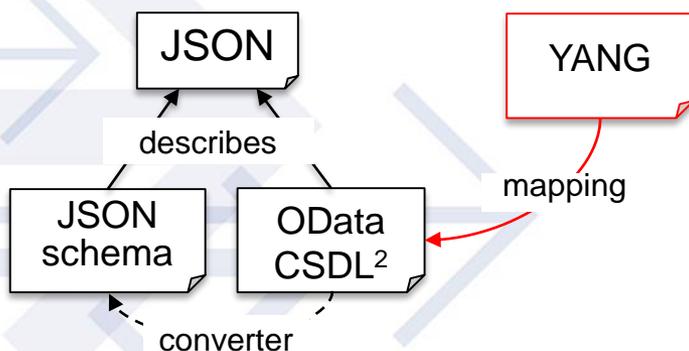
How does Redfish manage network devices?

- Expose IETF YANG models as Redfish schema
 - YANG modules (and their "container" nodes) become Redfish resources
 - The internal node structure of each YANG module is retained
 - YANG "Lists" are transformed into Redfish collections
 - Multiple devices can be managed as elements of the network device collection
 - /redfish/v1/NetworkDevices/MySwitch1
 - /redfish/v1/NetworkDevices/MySwitch2
 - Also supports different views of the same device
- Access those models via Redfish JSON
 - Consistent with compute and storage management



DMTF Redfish Standard

- Redfish Interface (RESTful)
 - HTTP/HTTPS - protocol
 - JSON – format of content
- Redfish Models
 - JSON content described in OData CSDL¹
 - Tool converts OData CSDL to json-schema



¹OData is an OASIS Standard

²CSDL = Common Schema Definition Language



Mapping YANG-to-CSDL

Translate YANG models to Redfish CSDL schema

YANG data model (RFC7223)

```
+--rw interfaces
| +--rw interface* [name]
| +--rw name string
| +--rw description? string
| +--rw type identityref
| +--rw enabled? boolean
| +--rw link-up-down-trap-enable? enumeration
+--ro interfaces-state
+--ro interface* [name]
+--ro name string
+--ro type identityref
+--ro admin-status enumeration
```

Redfish JSON HTTP GET response

```
{
  "Id": "ethernet1",
  "Name": "ethernet1",
  "Description": "Ethernet interface on slot 1",
  "type": "iana_if_type:ethernetCsmacd",
  "enabled": "true",
  "link_up_down_trap_enable": "true"

  "@odata.context": "...",
  "@odata.type": "#interface_v1_0_0.interfaces",
  "@odata.id": "/redfish/v1/NetworkDevices/Switch1/ietf_interfaces/interfaces/ethernet1"
}
```

YANG model

```
RFC7223
<CODE BEGINS>
module ietf-interfaces {
  ...
}
<CODE ENDS>
```

Translate (mapping)

Redfish CSDL

```
./ietf_interfaces.xml
./ietf_interfaces.interfacesCollection.xml
./ietf_interfaces.interfaces.xml
...
```

XML Schemas define JSON payloads



Example Mapping – Resultant Resources

Complete YANG structure reuse

```
/redfish/v1/NetworkDevices/{id}/ietf_interfaces  
/redfish/v1/NetworkDevices/{id}/ietf_interfaces/interfaces  
/redfish/v1/NetworkDevices/{id}/ietf_interfaces/interfaces/{id}  
/redfish/v1/NetworkDevices/{id}/ietf_interfaces/interfaces_state  
/redfish/v1/NetworkDevices/{id}/ietf_interfaces/interfaces_state/{id}  
/redfish/v1/NetworkDevices/{id}/ietf_interfaces/interfaces_state/{id}/statistics
```

RFC7223 (interfaces)
RFC7224 (IANA)
RFC7277 (IPv4/v6)

```
/redfish/v1/NetworkDevices/{id}/ietf_system/system  
/redfish/v1/NetworkDevices/{id}/ietf_system/system/authentication  
/redfish/v1/NetworkDevices/{id}/ietf_system/system/clock  
/redfish/v1/NetworkDevices/{id}/ietf_system/system/dns_resolver  
/redfish/v1/NetworkDevices/{id}/ietf_system/system/dns_resolver/options  
/redfish/v1/NetworkDevices/{id}/ietf_system/system/ntp  
/redfish/v1/NetworkDevices/{id}/ietf_system/system/ntp/udp  
/redfish/v1/NetworkDevices/{id}/ietf_system/system_state  
/redfish/v1/NetworkDevices/{id}/ietf_system/system_state/clock  
/redfish/v1/NetworkDevices/{id}/ietf_system/system_state/platform
```

RFC7317 (system)



Overview

- Redfish standard: Converged Infrastructure (CI) management interface
 - CI = rack-integrated compute, network and storage for data centers
 - CI data center networks (e.g., top of rack switches) are crucial
- Redfish network management – based on YANG module reuse
 - Translate YANG to Redfish interface schema
- Our (initial) goal: baseline data center switch/router device profile for top-of-rack CI switches
 - List of applicable YANG modules
 - Structure for how modules go together to manage a top-of-rack CI switch
- Result: IETF YANG device profile for use by IETF, DMTF, etc.
 - Device profile should not be not Redfish-specific
 - Looking for help and guidance



Backup





Redfish Capabilities (v1.0+)

Discovery

- Chassis
- Computer systems
- Managers

Server/Platform Information

- Server identification and asset info
- Host network MAC addresses
- Local storage
- Power supply and fans
- State and Status

Common Manageability

- Change boot order / device
- Reboot / power cycle server
- Power usage and thresholds
- Temperature
- Config serial console access via SSH

BMC Infrastructure

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

Eventing, Logs, Accounts

- Subscribe/publish event model
- Logs and entries
- Accounts

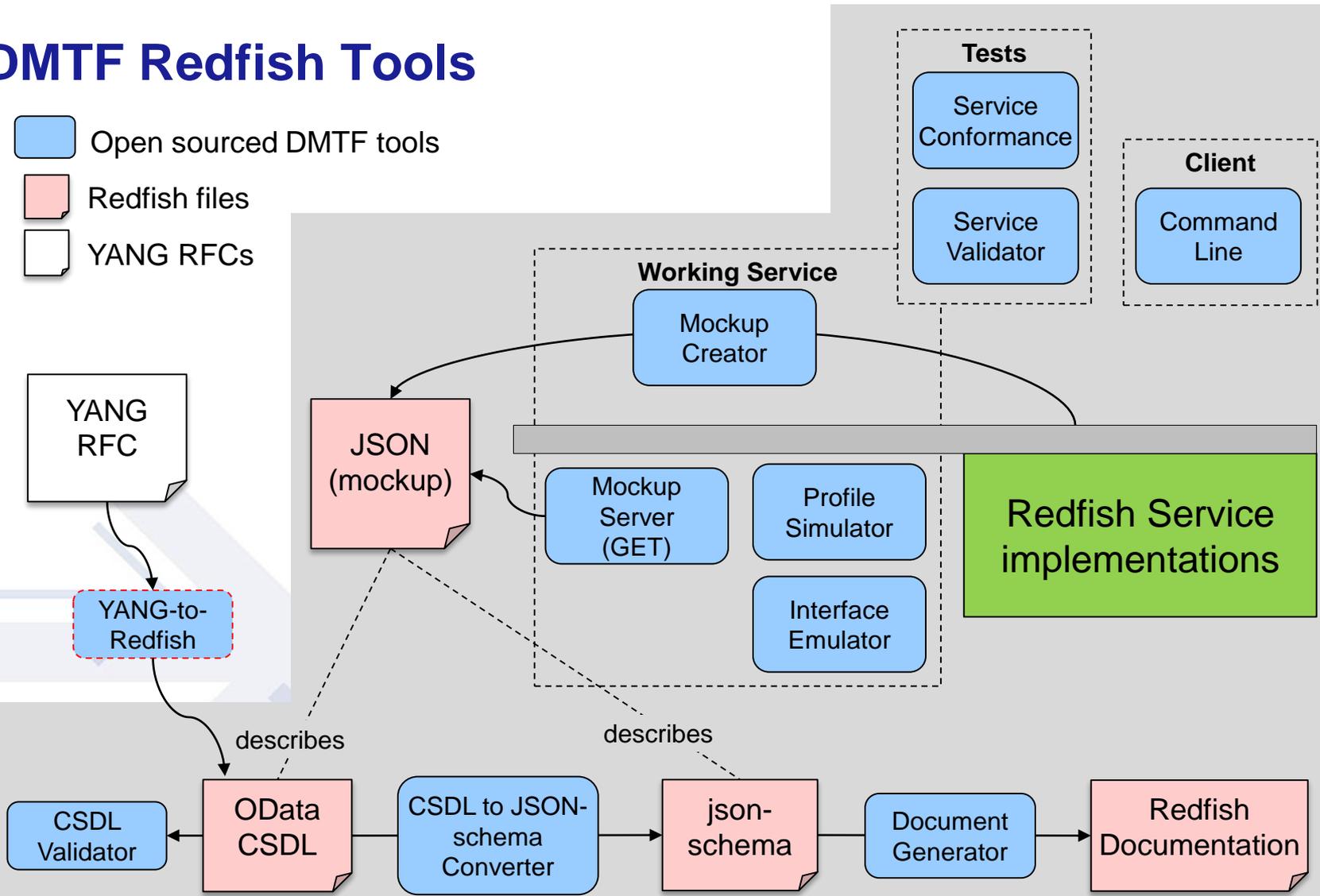
New Manageability¹

- BIOS
- Memory
- Disk drives, Storage & Volume
- Endpoints & fabric
- PCIe switch, device & zone
- Firmware/Software inventory & Update



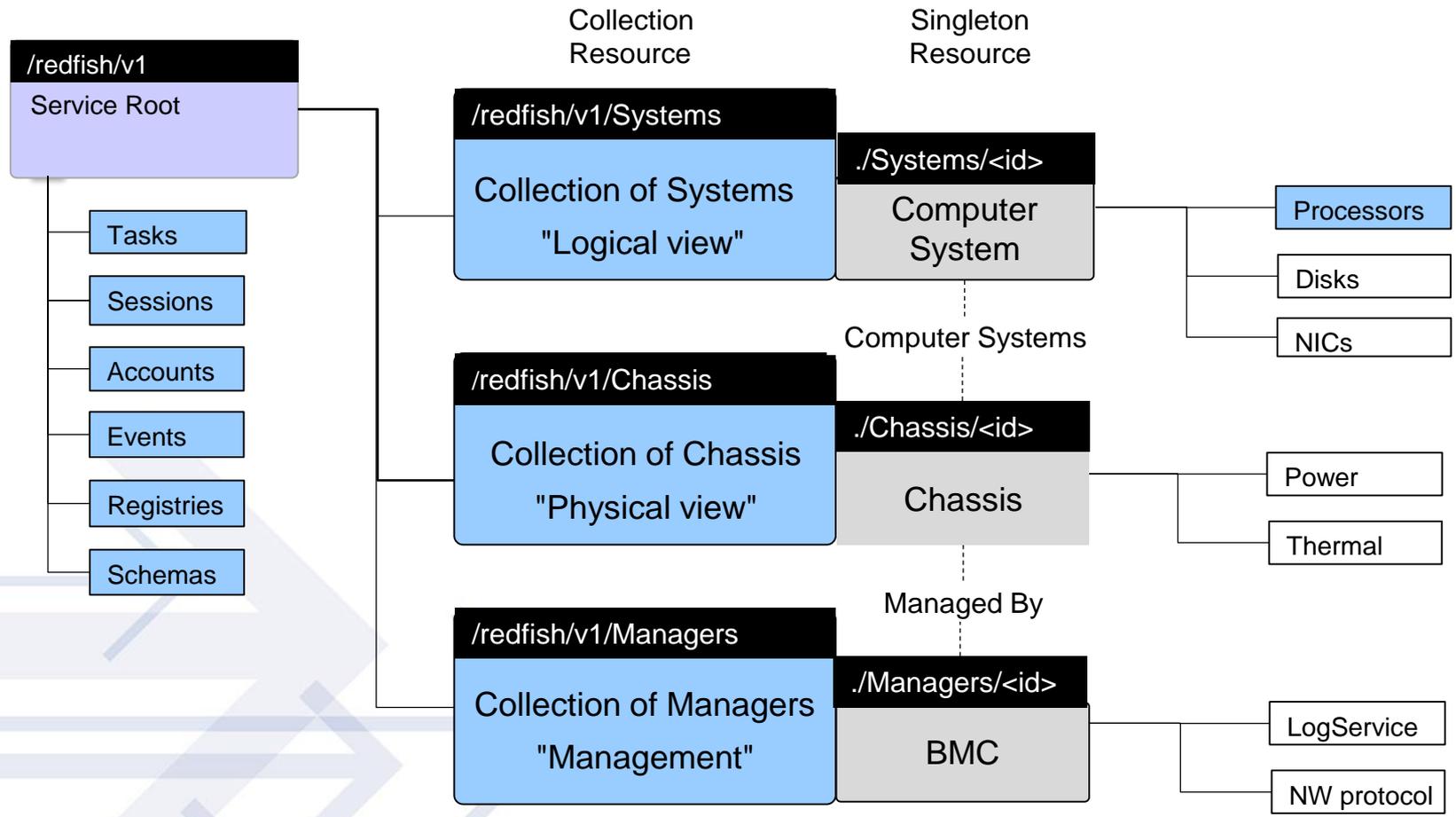
DMTF Redfish Tools

-  Open sourced DMTF tools
-  Redfish files
-  YANG RFCs





Redfish Resource Map



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

The Redfish Resource Explorer (redfish.dmtf.org) allows exploration the resource map



Mapping Guidelines

- Retain all naming (spelling and capitalization)
- When required, synthesize names to retain YANG scoping
- Align to Redfish string convention
 - Convert dashes ("-") to underlines ("_"), when used in an identifier
 - Convert colons (":") to periods ("."), when used in an identifier
- Map RFC's as-is - suppress desire to optimize
- Convert everything in the RFC – don't worry about feature exposure exclusion
- Treat some YANG statements as a pre-processor style directive (e.g. uses, grouping)