



Introduction and Overview of Redfish

John Leung

DMTF - VP of Alliances

Intel – Principal Engineer



The Distributed Management Task Force

- **An Industry Standards Organization**
 - Developing manageability standards for 24 years (est. 1992)
 - Membership includes 65 companies and industry organizations
 - With active chapters in China and Japan
- **Allied with**
 - 14 standard development organizations (alliance partners)
 - 80+ universities and research organizations (academic alliance partners)
- **Focused on manageability standards**
 - For the management of on-platform, off-platform, network services and infrastructure domains
 - Which are recognized nationally (ANSI/US) and internationally (ISO)



Agenda

- Redfish - a modern manageability interface for the data center
 - Why a new interface?
 - Redfish capabilities
 - Elements of the Redfish standard
 - A JSON Response
- Redfish manageability models
 - Server/Compute model
 - Storage model
 - Network Device model
- Redfish tools
 - The conformance tool chain and OCP



"Redfish – a modern interface for managing the data center"

- A RESTful interface
 - To manage compute, storage, network and DCIM
 - Leverages existing Internet standards and tool chains
 - Usable by professions and amateurs
- Resource models for managing
 - Common platform manageability
 - (Power, thermal, cooling, inventory, reboot, firmware update, get telemetry, etc.)
 - Domain specific capabilities



Compute



Redfish

Aug 2015



Storage



Swordfish

Aug 2016



Network

I E T F
YANG

Internet Draft 2017

On-Platform



DCIM (facilities)



DCIM = Data Center Infrastructure Management



Redfish: Why a New Interface?

- Market shifting to scale-out solutions
 - Datacenters have a sea of simple servers and multi-node servers
- Customers exhausting the functionality of current manageability interfaces
- Customers asked for a modern interface
 - Single simple interface for managing all datacenter platforms and devices
 - An interface which uses cloud/web protocols, structures, security models and tool chains
 - Schemas to allow introspect of interface and programmatic enablement

HTTP

```
HTTP GET https://<ip_addr>/redfish/v1/Systems/CS_1
```

**Python
code**

```
rawData = urllib.urlopen('https://<ip_addr>/redfish/v1/Systems/CS_1')  
jsonData = json.loads(rawData)  
print( jsonData['SerialNumber'] )
```

Output

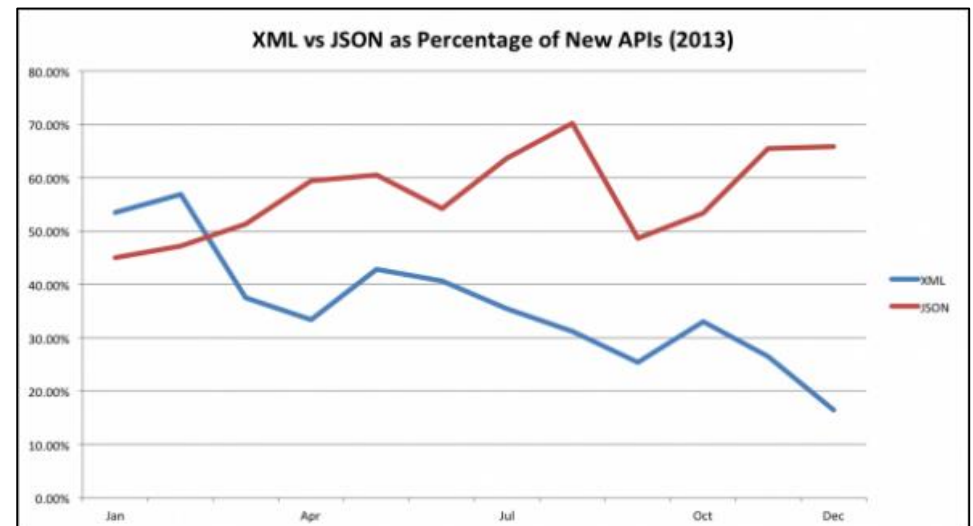
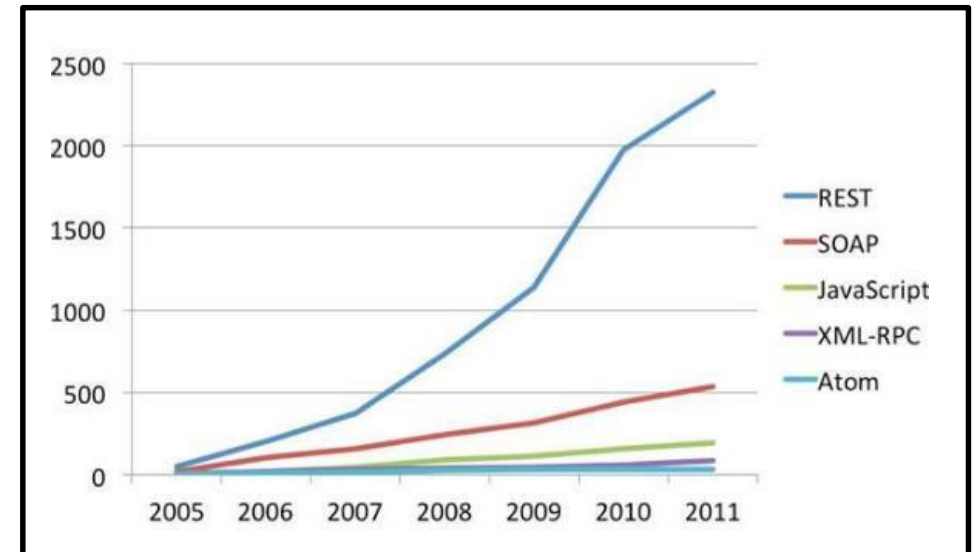
```
1A87CA442K
```

Why HTTP and JSON?

- HTTP(S): The Web protocol
 - Well-understood by IT admin
 - Known security model
 - Known network configuration
- JSON: A modern data format
 - Human-readable
 - Simpler than XML
 - Modern language support (json-schema)
- For manageability, IT can use their
 - Existing DEV/OPS skill set
 - Tool chain ecosystem

<http://www.infoq.com/articles/rest-soap>

<http://www.programmableweb.com/news/jsons-eight-year-convergence-xml/2013/12/26>



Redfish Capabilities

Chassis Information

- Identification and asset information
- State and status
- Temperature sensors and fans
- Power supply, power consumption and thresholds
- Set power thresholds

Compute Manageability

- Reboot and power cycle server
- Configure BIOS settings
- Change boot order and device
- Update BIOS and firmware
- Memory and NVDIMMs
- Local network interface
- Local storage
- State and status

Composition

- Specific composition
- Enumerated composition

Management Infrastructure

- View / configure BMC network settings
- Manage local BMC user accounts
- Configure serial console access (e.g. SSH)

Discovery

- Physical hierarchy (rack/chassis/server/node)
- Compute service (servers)
- Management hierarchy (rack mgr, tray mgr, BMC)

Security

- Use HTTPS
- Map roles to privileges

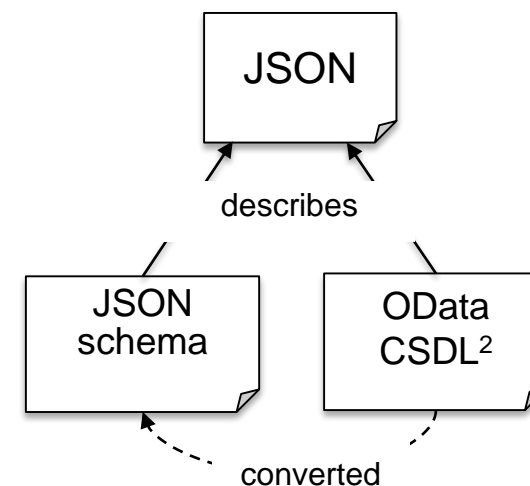
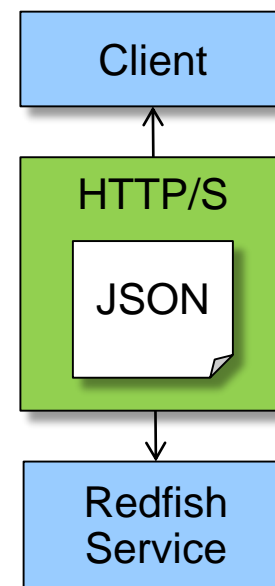
Access and Notification

- Subscribe to published events
- Inspect Logs
- Access via host interface



The Redfish Standard

- Redfish is composed of
 - An Interface definition
 - Various Model schema
- Redfish Interface (RESTful)
 - HTTP/HTTPS - protocol
 - JSON – format of content
- Redfish Models and Schema
 - Schema format for JSON responses
 - DMTF publishes the models for platforms and compute/servers
 - Other standard bodies can publish their own



¹OData is an OASIS Standard

²CSDL = Common Schema Definition Language

JSON response

HTTP GET /redfish/v1/Systems/CS_1

Note

- Redfish is hyper-media
- Cannot presume a resource hierarchy

Simple properties

Complex properties

Subordinate resources

Associated resources

Actions

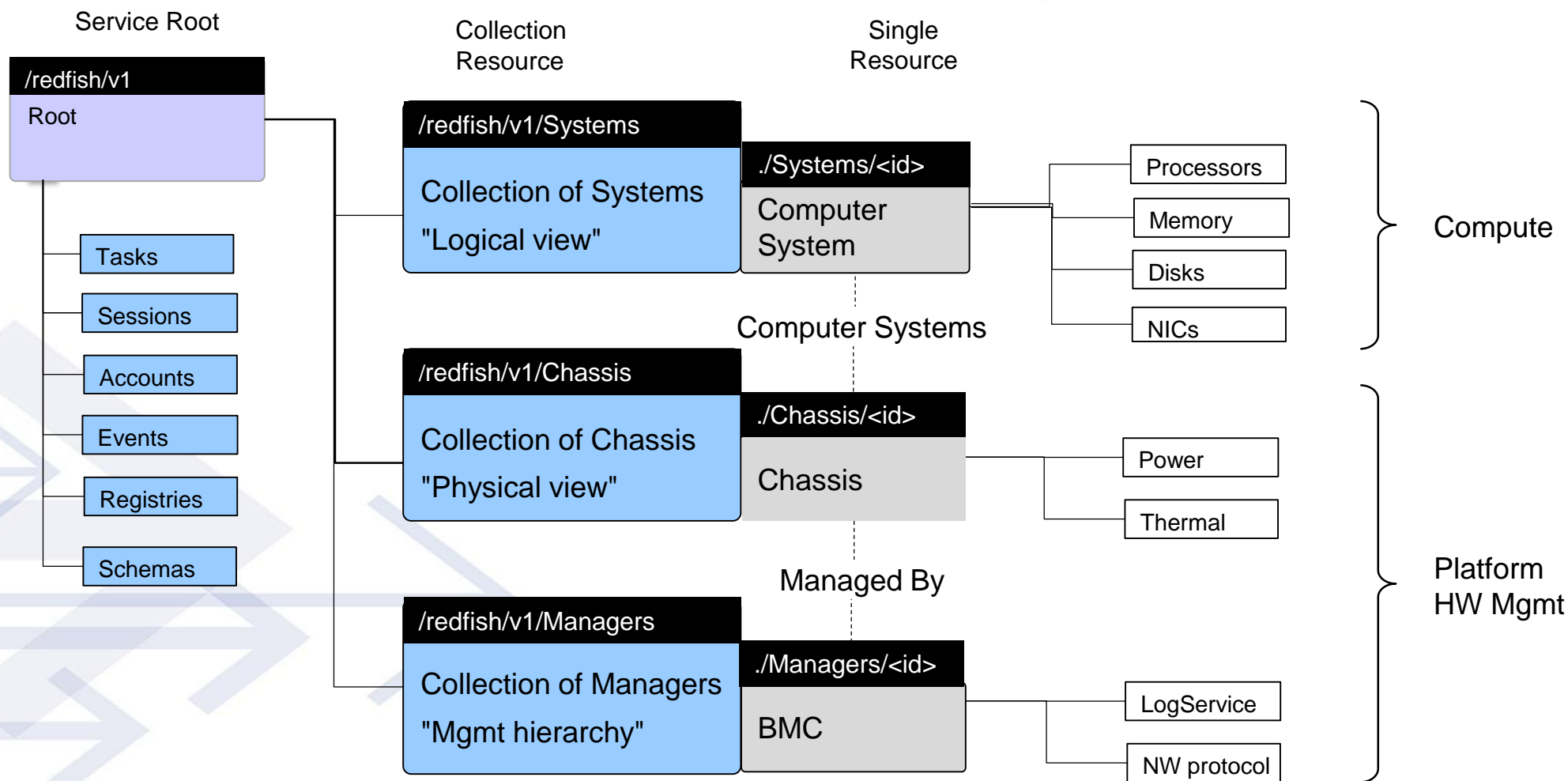
```
{
  "@odata.context": "/redfish/v1/$metadata#ComputerSystem.ComputerSystem",
  "@odata.id": "/redfish/v1/Systems/CS_1",
  "Id": "CS_1",
  "Name": "My Computer System",
  "SystemType": "Physical",
  "AssetTag": "free form asset tag",
  "Manufacturer": "Manufacturer Name",
  "Model": "Model Name",
  "SerialNumber": "2M220100SL",
  "PartNumber": "",
  "Description": "Description of server",
  "UUID": "00000000-0000-0000-0000-000000000000",
  "HostName": "web-srv344",
  "IndicatorLED": "Off",
  "PowerState": "On",
  "BiosVersion": "P79 v1.00 (09/20/2013)",
  "Status": { "State": "Enabled", "Health": "OK", "HealthRollup": "OK" },
  "Boot": { . . . },
  "ProcessorSummary": { . . . },
  "MemorySummary": { . . . },
  "TrustedModules": [ { . . . } ],
  "Processors": { "@odata.id": "/redfish/v1/Systems/CS_1/Processors" },
  "Memory": { "@odata.id": "/redfish/v1/Systems/CS_1/Memory" },
  "EthernetInterfaces": { "@odata.id": "/redfish/v1/Systems/CS_1/EthernetInterfaces" },
  "SimpleStorage": { "@odata.id": "/redfish/v1/Systems/CS_1/SimpleStorage" },
  "LogServices": { "@odata.id": "/redfish/v1/Systems/CS_1/LogServices" },
  "SecureBoot": { "@odata.id": "/redfish/v1/Systems/CS_1/SecureBoot" },
  "Bios": { "@odata.id": "/redfish/v1/Systems/CS_1/Bios" },
  "PCIeDevices": [ { "@odata.id": "/redfish/v1/Chassis/CS_1/PCIeDevices/NIC" } ],
  "PCIeFunctions": [ { "@odata.id": "/redfish/v1/Chassis/CS_1/PCIeDevices/NIC/Functions/1" } ],
  "Links": {
    "Chassis": [ { "@odata.id": "/redfish/v1/Chassis/Ch_1" } ],
    "ManagedBy": [ { "@odata.id": "/redfish/v1/Managers/Mgr_1" } ],
    "Endpoints": [ { "@odata.id": "/redfish/v1/Fabrics/PCIe/Endpoints/HostRootComplex1" } ],
  },
  "Actions": {
    "#ComputerSystem.Reset": {
      "target": "/redfish/v1/Systems/CS_1/Actions/ComputerSystem.Reset",
      "@Redfish.ActionInfo": "/redfish/v1/Systems/CS_1/ResetActionInfo"
    }
  }
}
```



Redfish Model – Compute and Platform



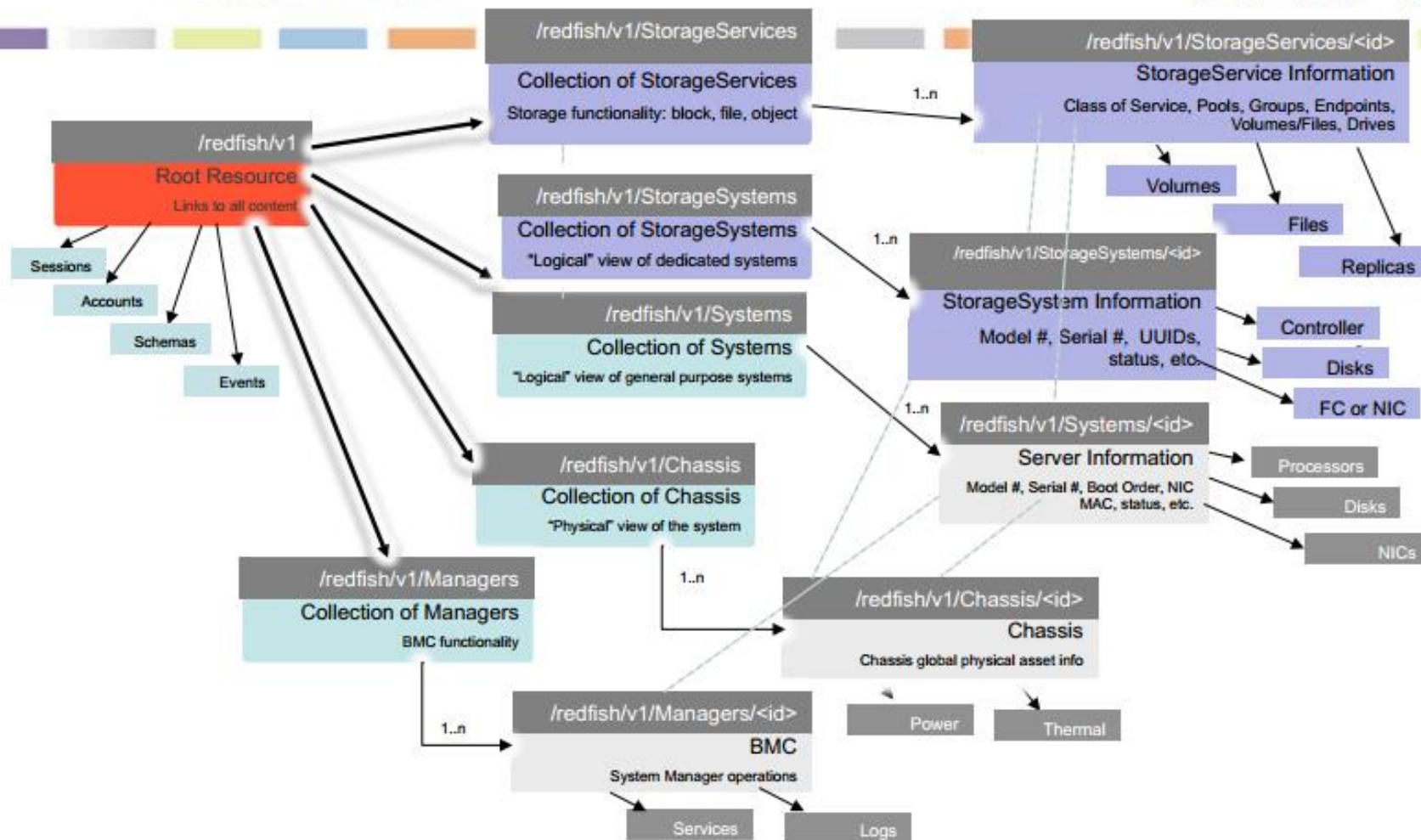
HTTP GET /redfish/v1/Systems/CS_1/Processors/2



Storage Model

- Reuses chassis model
- Adds StorageServices & StorageSystems

Adding Storage to Redfish: Swordfish

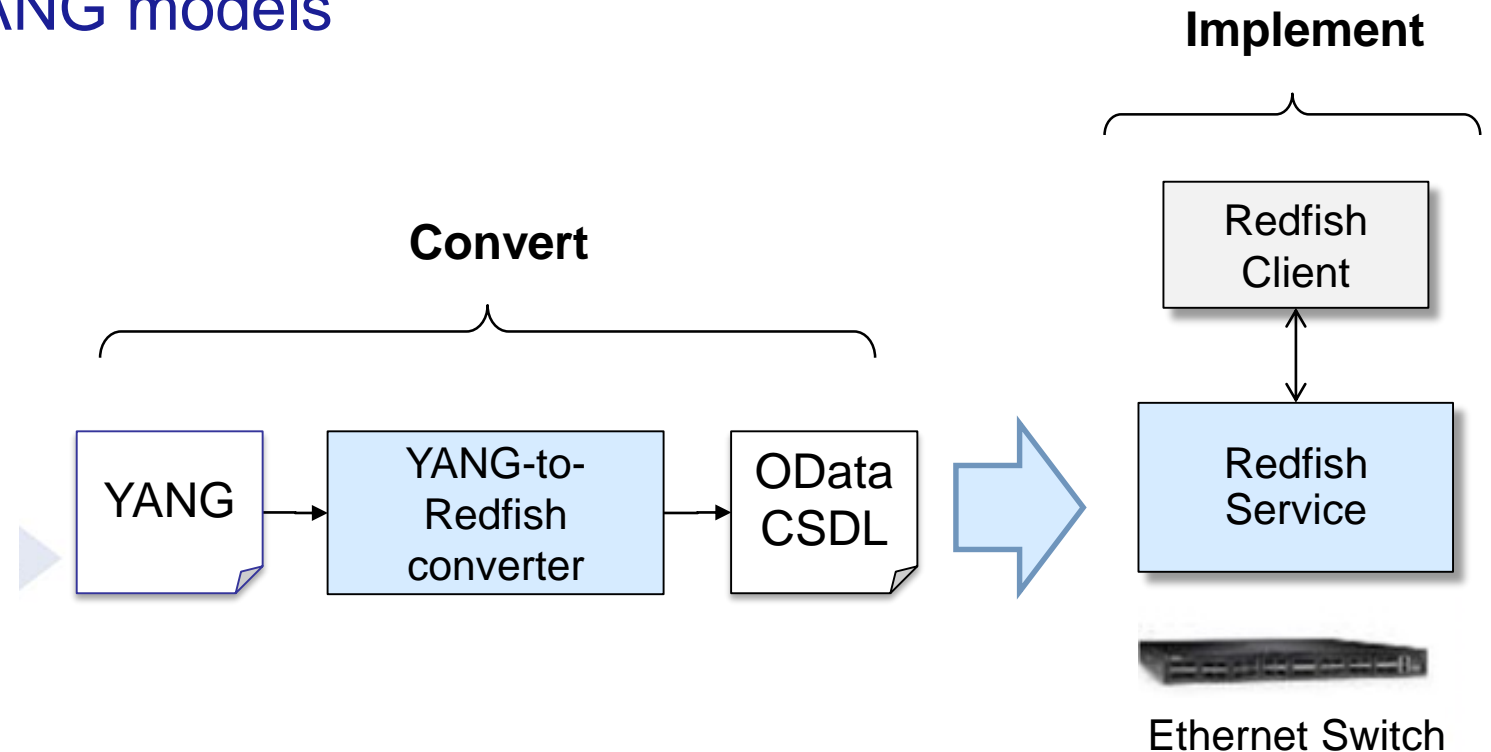


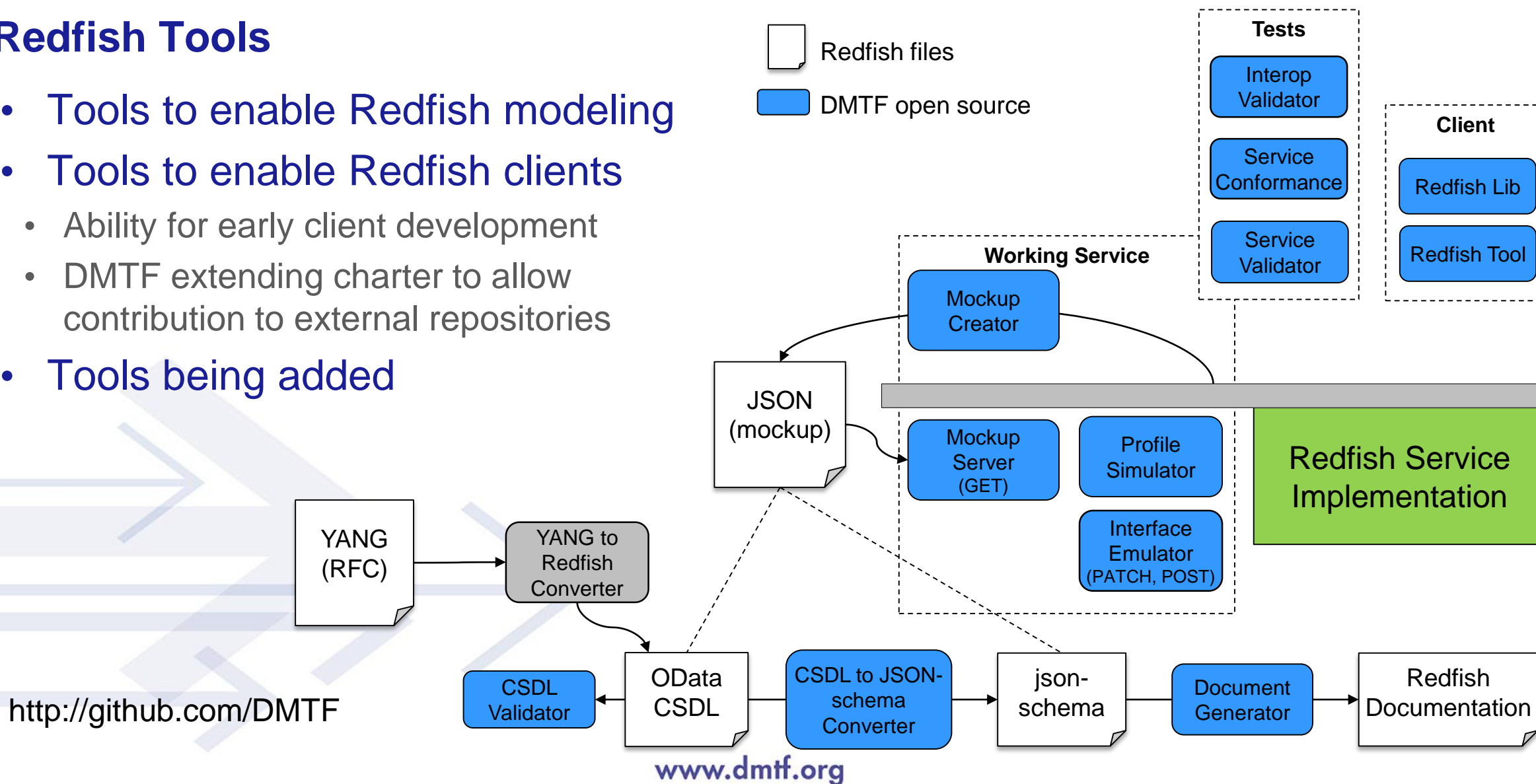
Network Model – Convert from YANG models

- ✓ Phase 1 - convert a small set of YANG models to Redfish models
 - Proves out the process, and validates the converter
- Phase 2 – larger list of YANG models

Phase 1 (Ethernet Switch)

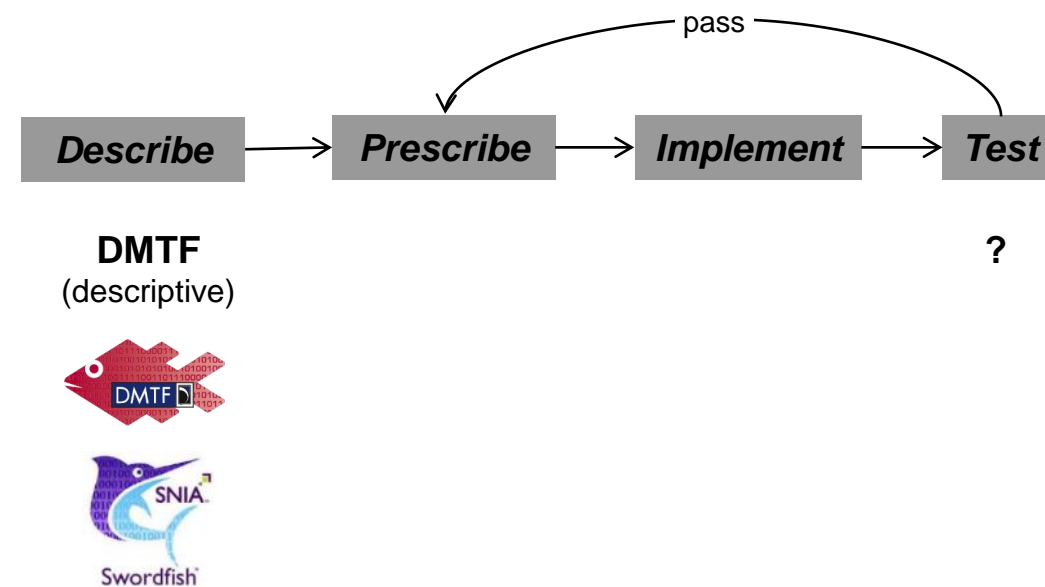
- RFC6991 (YANG types)
- RFC7223 (Interfaces)
- RFC7224 (IANA Interface types)
- RFC7277 (IPv4 and IPv6)
- RFC7317 (system, system_state, platform, clock, ntp)





The Redfish Conformance Tool-chain

- DMTF describes a standard for a manageability interface and models
- OCP could prescribe conformance requirements for the Redfish model elements (resources, properties, actions) in a profile document
- Implementations are tested for conformance to the profile to claim conformance to OCP specifications



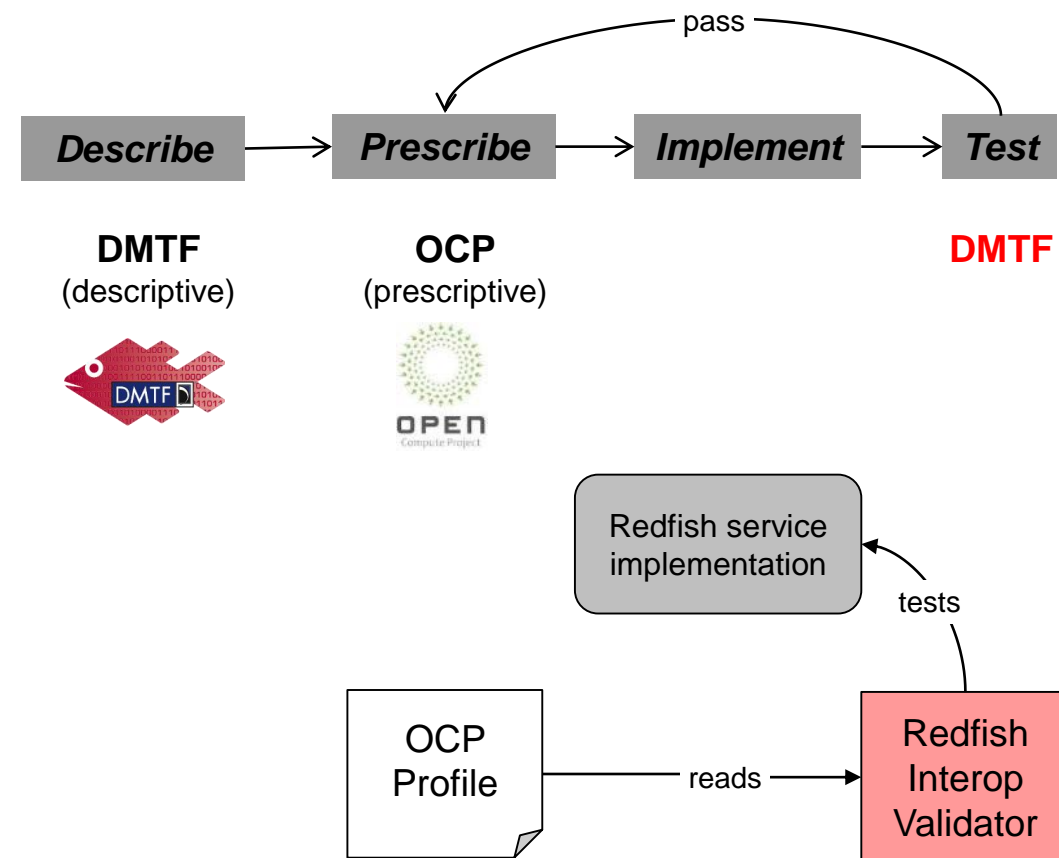
OCP Conformance of Redfish Profile

The DMTF is

- Defining a JSON file format for a prescriptive profile
- Developing a Redfish Interop Validator
 - Reads the profile file
 - Profile drives the testing of a Redfish service implementation
- The Redfish Interop Validator has been open sourced
 - <https://github.com/DMTF/Redfish-Interop-Validator>

OCP could




- Create a baseline manageability profile
- Create project specific profiles




Public Redfish Collateral

- Redfish Github github.com/DMTF
- Redfish Community Forum redfishforum.com
- Redfish Developer's Hub redfish.dmtf.org
- Specs, presentation dmtof.org/standards/redfish
- Redfish Forum (SPMF) dmtof.org/standards/spmf



Redfish Specification Forum				
Home Help Search 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. <small>Moderator: Admin</small>	1	2	Retrieving individual properties by j2hilland <small>Sep 12, 2016 at 7:42am</small>	
 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 mraierl <small>Aug 12, 2016 at 6:33am</small>	
 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 <small>Aug 16, 2016 at 10:10am</small>	


 DISTRIBUTED MANAGEMENT TASK FORCE, INC.
Redfish™ Developer Hub


[Home](#)
[Mockups](#)
[About the Redfish API](#)

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 consistent API.

Welcome Developers

The DMTF's Redfish Developer Hub is a one-stop, in-depth technical resource – by developers, for developers – **designed to provide all the files, tools, community support, tutorials and other advanced education you may need to help you use Redfish.**


 DISTRIBUTED MANAGEMENT TASK FORCE, INC.
Redfish Resource Explorer

[Home](#)
[Mockup](#)
[About the Redfish API](#)

Development Mockup

[Explore the Resources](#)

Main
 Systems
 1
 2
 Chassis
 Managers
 Task Service
 Session Service
 Account Service
 Event Service
 JsonSchemas

```

redfish > v1 > Systems > 1
{"$Redfish.Copyright": "Copyright \u00a9 2014-2015 Distributed Management Task Force, Inc. (DMTF). All rights reserved.",
"$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",
"Manufacturer": "Manufacturer Name",
"Model": "Model Name",
"SKU": "",
"SerialNumber": "2M22010086",
"PartNumber": "",
"Description": "Description of server",
"UUID": "00000000-0000-0000-0000-000000000000"}

```



Summary

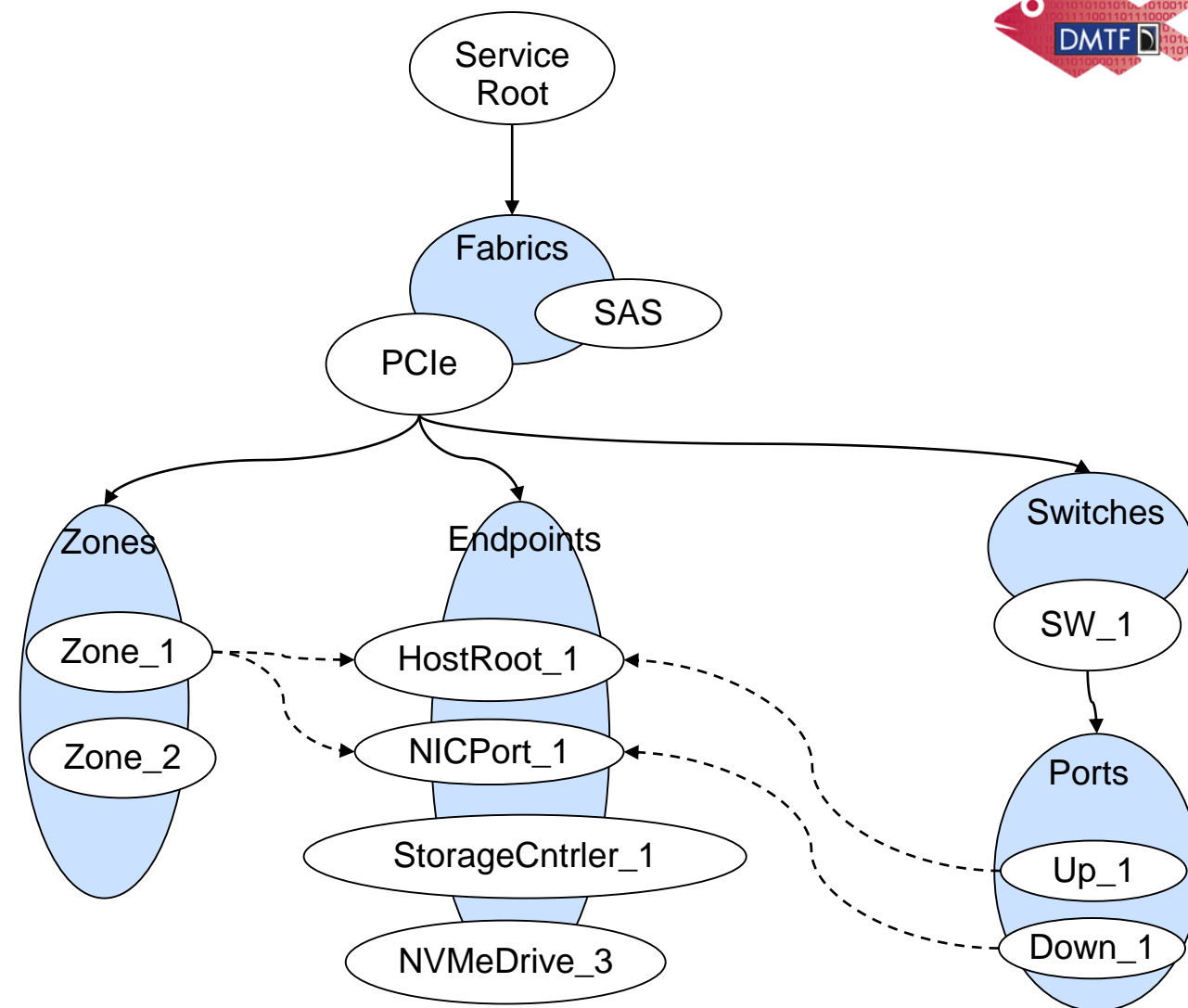
- Redfish has rapidly established itself as the modern interface for data center management
 - Rapid advances in the interface with multiple schema releases
 - Expediting the tool-chain for extensions and usage
- The industry have reacted favorably (standards orgs, companies)
 - Alliance partnerships with SNIA, UEFI, OCP, The Green Grid, ASHRAE, CSCC
- Academic research is underway (with academic alliance partner members)
 - Texas Tech University - Cloud and Autonomic Computing Center
 - Barcelona Supercomputing Center



Thank you

PCIe Model

- The Fabric model is use to model PCIe, SAS, and other Fabrics.
- A fabric includes collections of zones, endpoints and switches
- A switch include a collection of ports
- Fabric mockups exist for PCIe, PCIeMesh and ComplexPCIe





Memory Model

- A computer system has physical memory
- A computer system may have memory domains
 - Each memory domains can be interleaved memory sets and memory chunks
 - Each memory chunks may have interleaved sets

