Introduction and Overview of Redfish

John Leung

OCP IC representative to the Hardware Management Project

DMTF - VP of Alliances

Intel – Principal Engineer (system manageability)

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

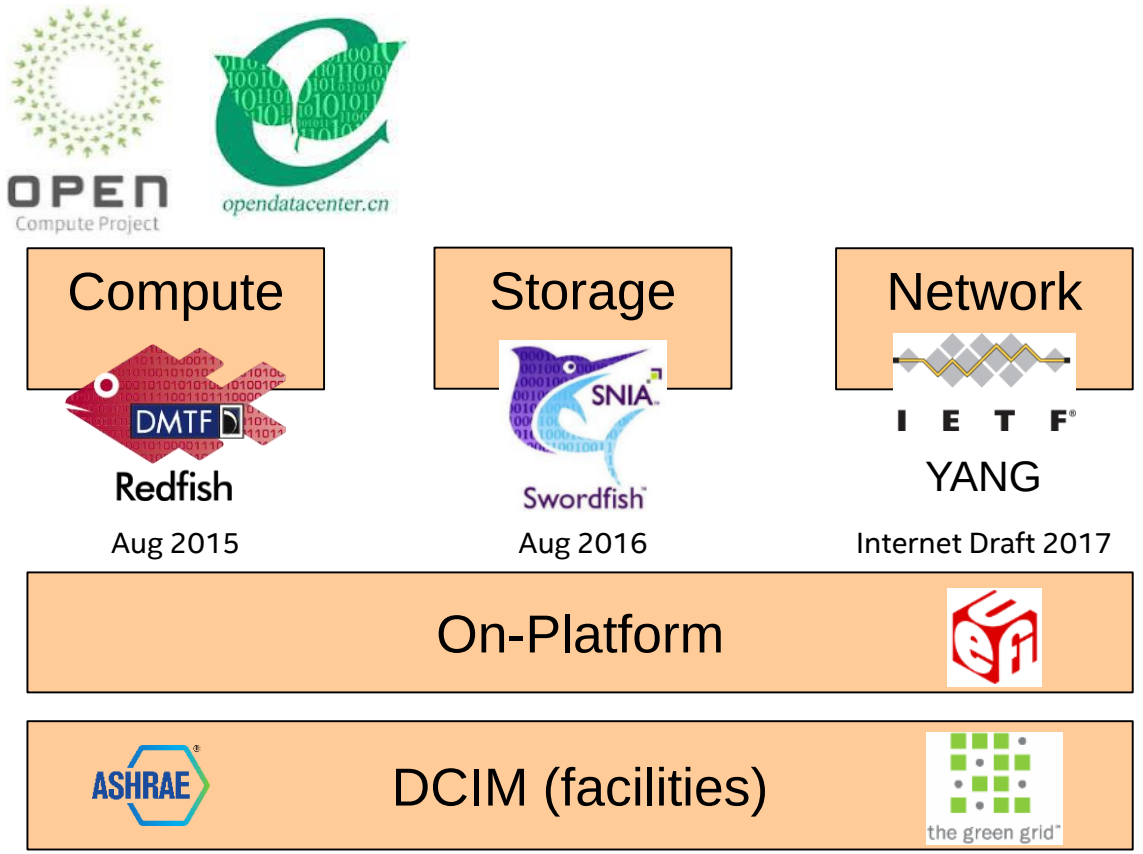
"Redfish – a modern interface for managing the data center"

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



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
```

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

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

Composition

- Specific composition
- Enumerated composition

The Redfish Standard

Redfish is composed of

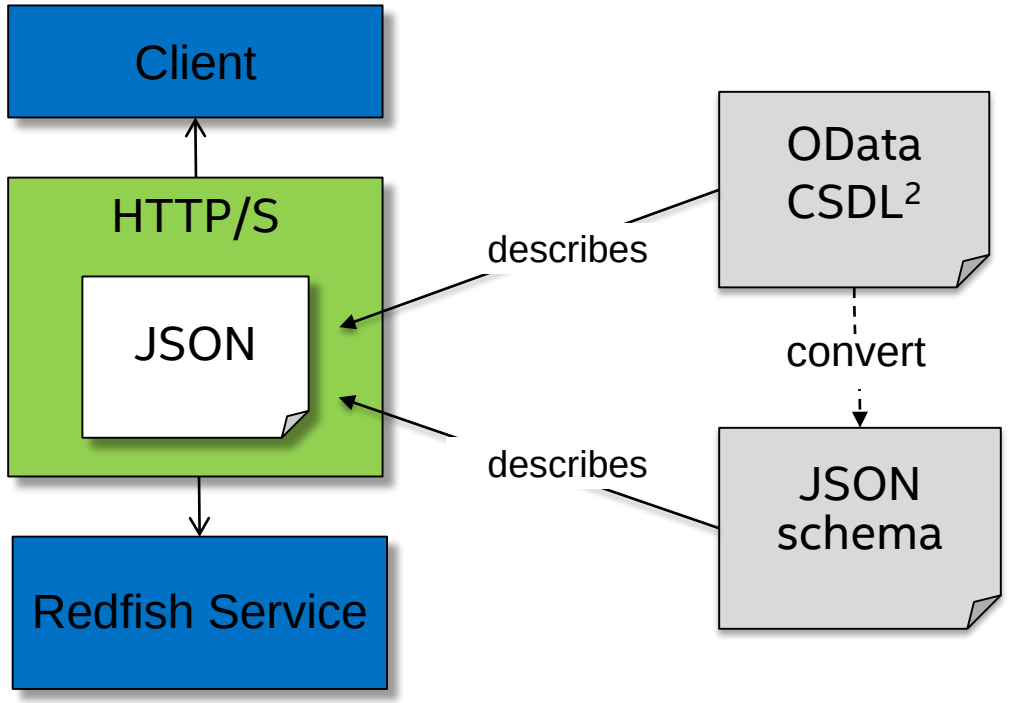
- An Interface definition
- Various Model schema

Redfish Interface (RESTful)

- HTTP/HTTPS - protocol
- JSON – format of content
- Schemas for JSON responses (enables tool chain and introspection)

Redfish Models and Schema

- DMTF publishes the models for platforms and compute/servers
- Other organization may create models for their management domain



[1]OData is an OASIS Standard
[2]CSDL = 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

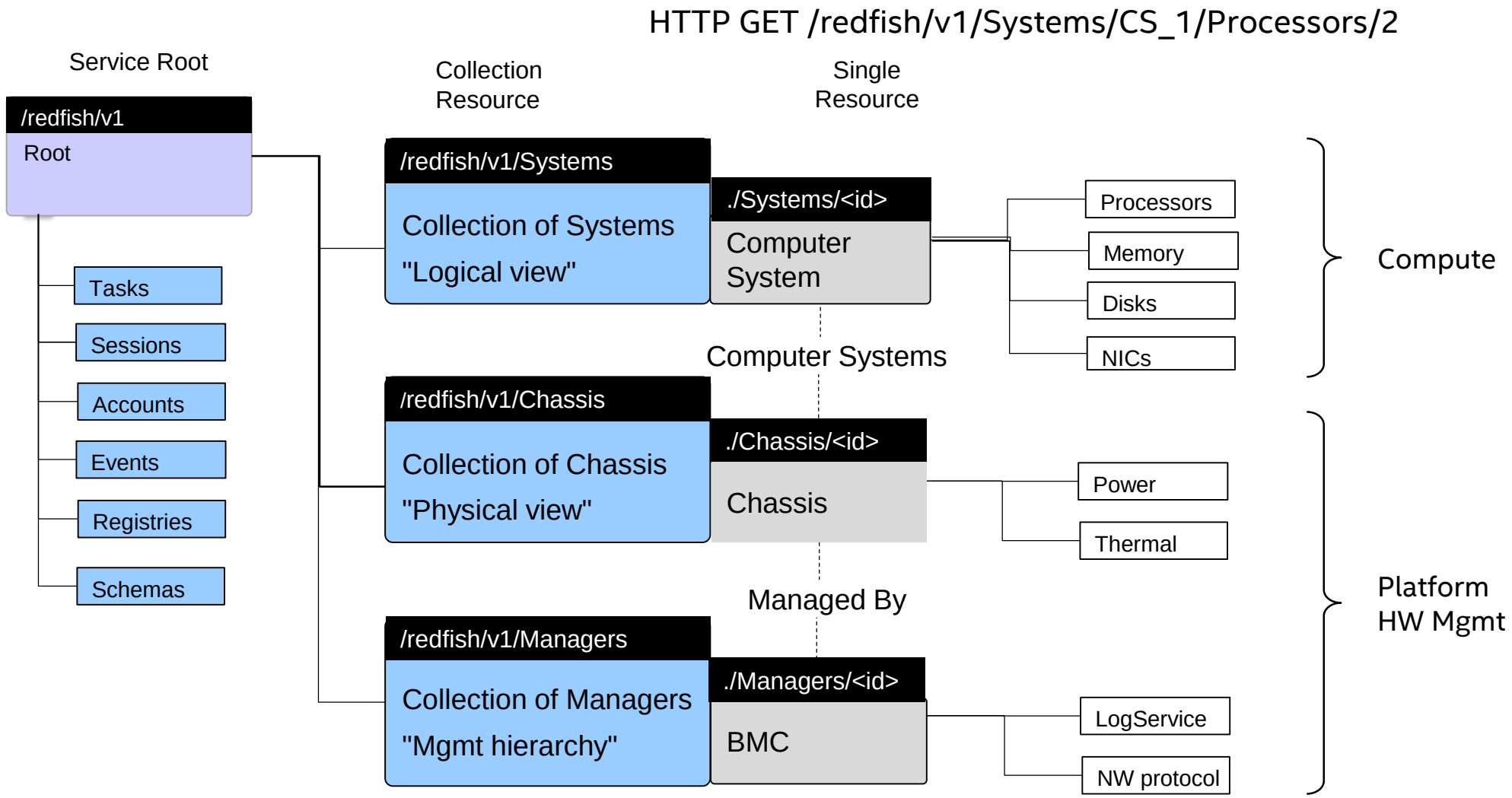
Subordinate resources

Associated resources

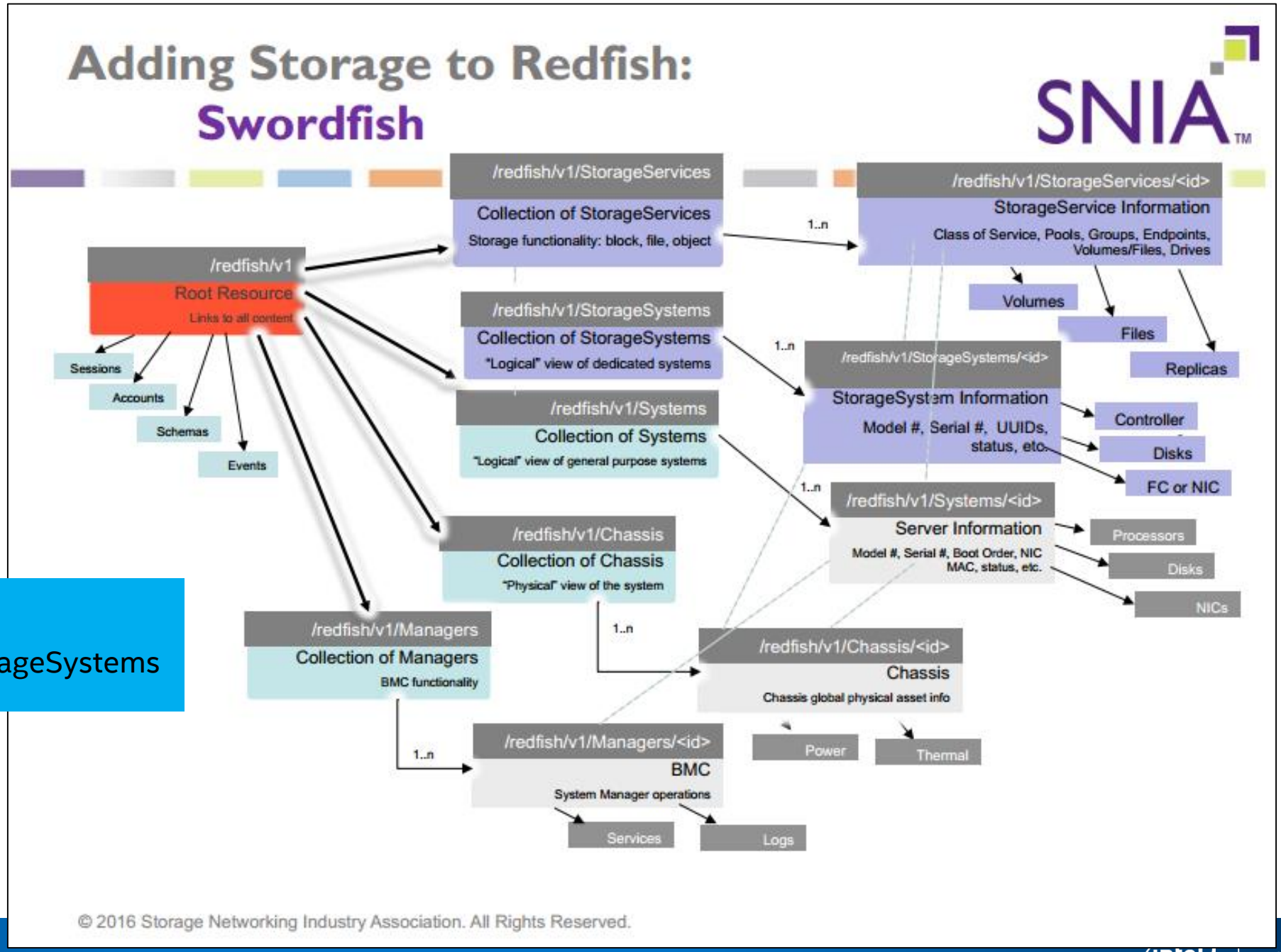
Actions


```
{
  "@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



Storage Model



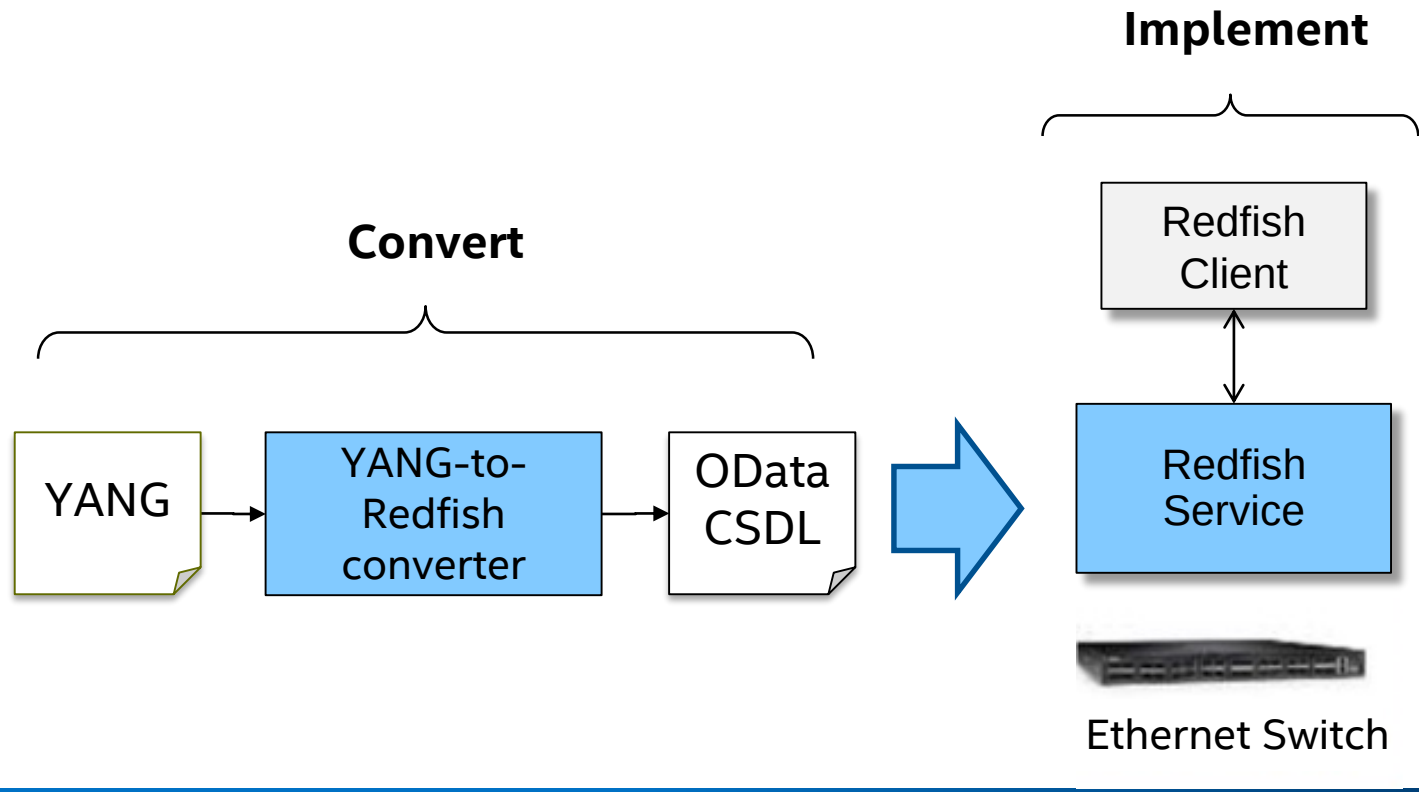
- Reuses chassis model
- Adds StorageServices & StorageSystems

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

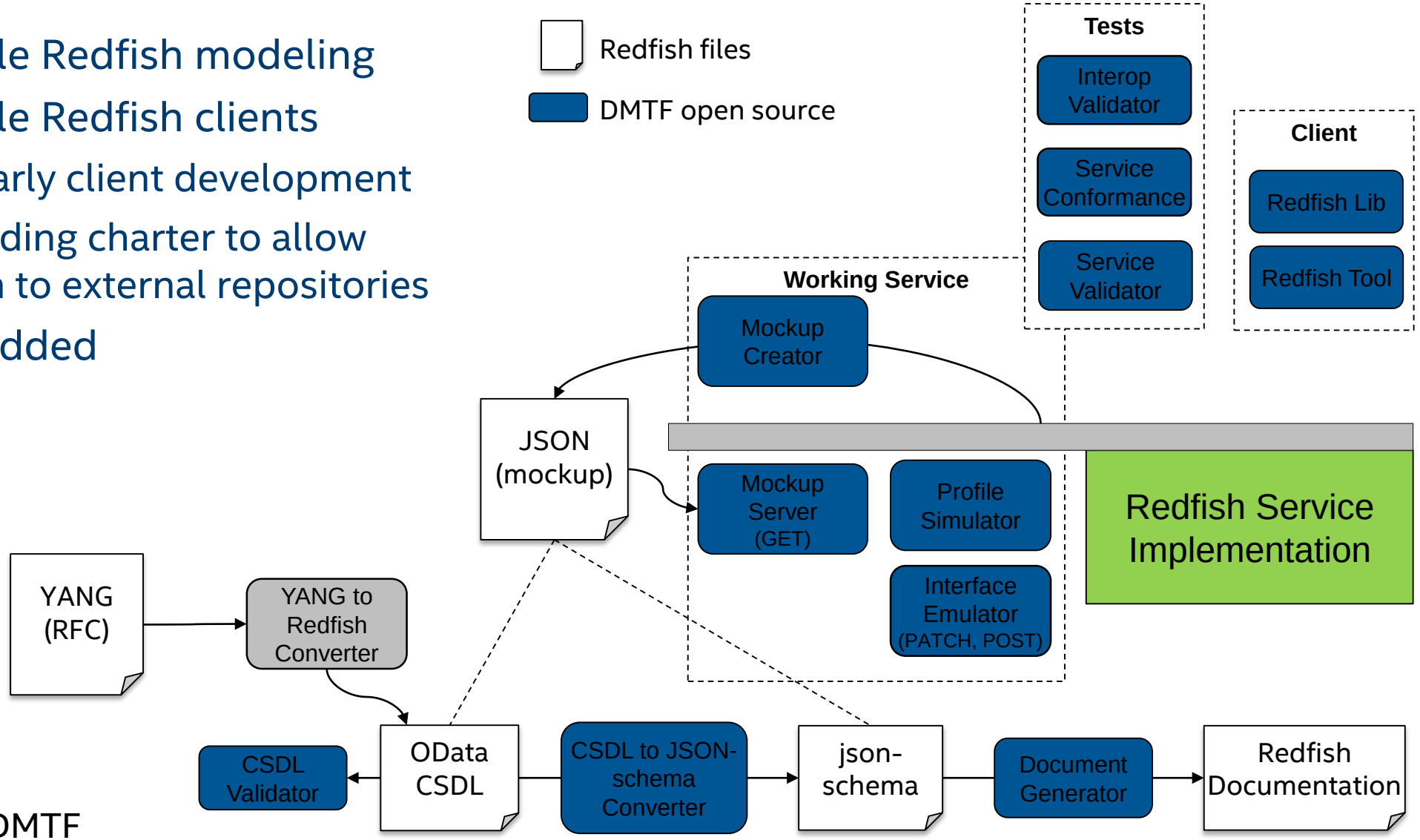
Phase 2 – larger list of YANG models

Phase 1 (Ethernet Switch)
- RFC7223 (Interfaces)
- RFC7224 (IANA Interface types)
- RFC7277 (IPv4 and IPv6)
- RFC7317 (system, system_state, platform, clock, ntp)



Redfish Tools

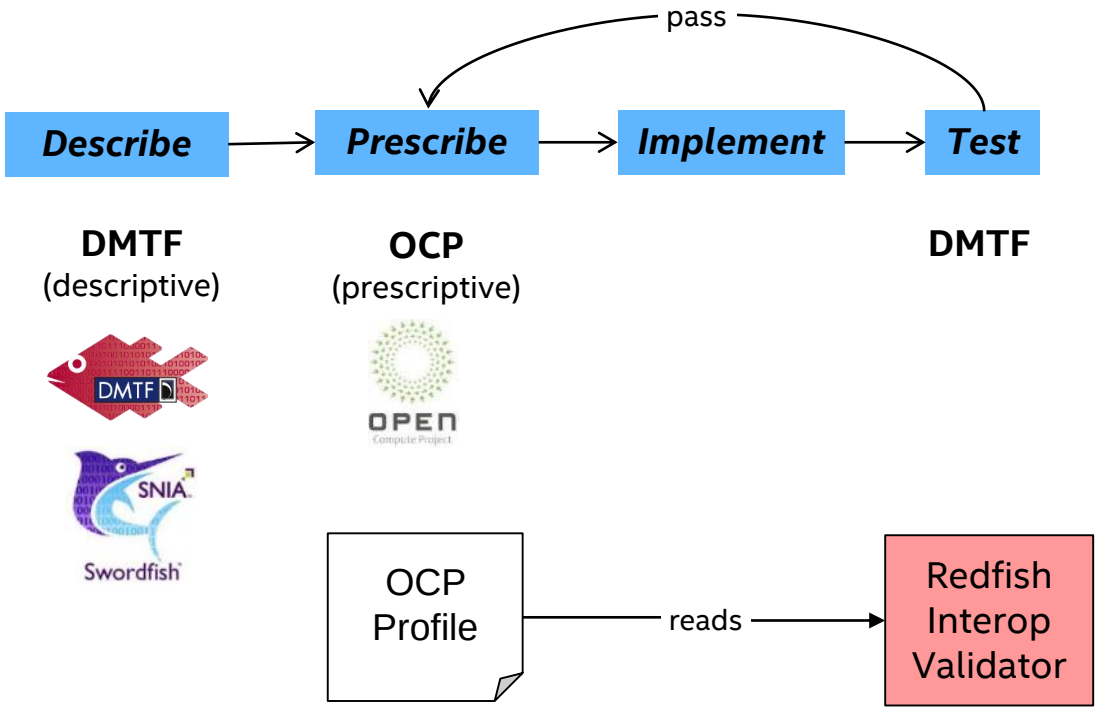
- Tools to enable Redfish modeling
- Tools to enable Redfish clients
  - Ability for early client development
  - DMTF extending charter to allow contribution to external repositories
- Tools being added



http://github.com/DMTF

Conformance

- 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
- DMTF develops a conformance test which reads a profile document
  - https://github.com/DMTF/Redfish-Interop-Validator



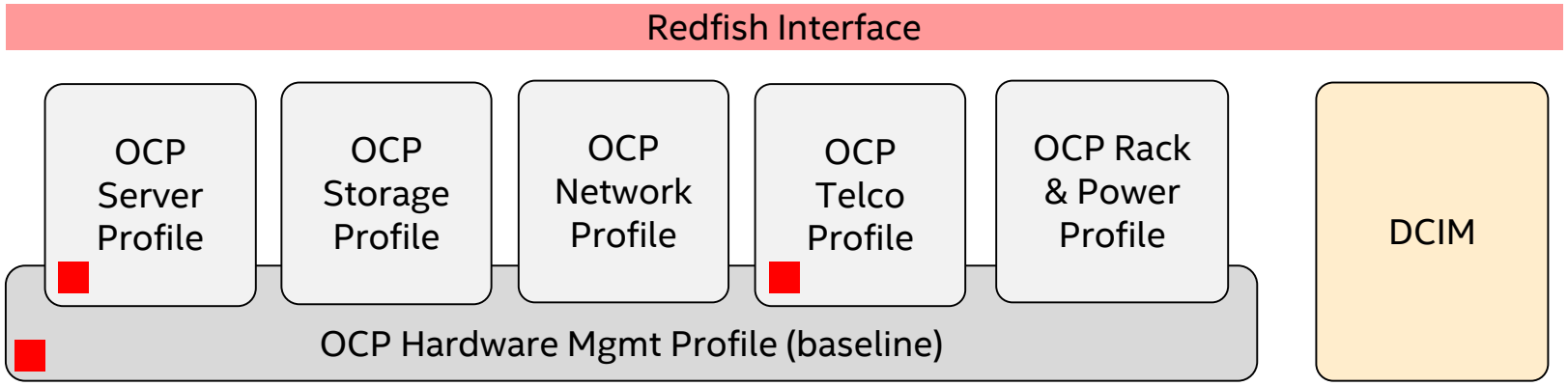
Baseline and Platform specific profiles

OCP HW Management project

- Specifies the baseline which is common across all OCP platforms
- DMTF members have been participating in the HW Mgmt project

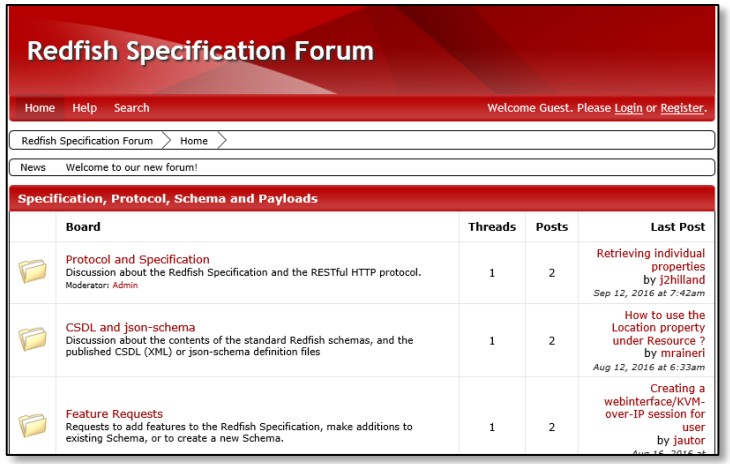
OCP ‘platform’ projects

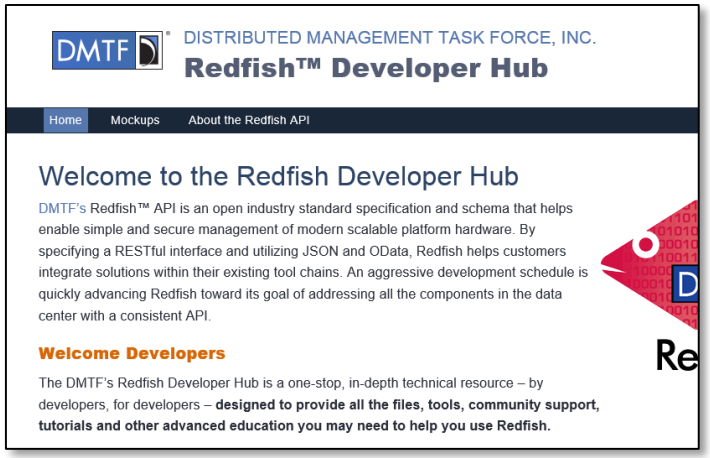
- Specifies platform specific profile, which references and extends baseline profile (proper superset?)

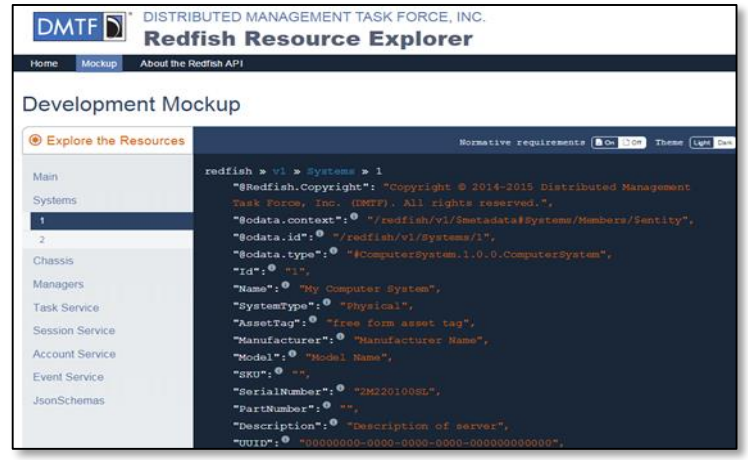


Public Redfish Collateral

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







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

Questions?