Introduction and Overview of Redfish

John Leung
Distributed Management Task Force - VP of Alliances
Intel Corporation – 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
  - Standards are recognized nationally (ANSI/US) and internationally (ISO)
Agenda

- Redfish - a modern manageability interface for the data center
- Why a new interface?
- Redfish capabilities
- The Redfish standard
  - A JSON Response
- Redfish Models
  - Compute, platform, storage, network models
  - PCIe and memory model
- Redfish tool-chain
- Public Redfish collateral
"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

Compute

Storage

Network

On-Platform

DCIM (facilities)

DMTF

Redfish

Aug 2015

SNIA

Swordfish

Aug 2016

IETF

YANG

Internet Draft 2017
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

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

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
The Redfish Standard

- Redfish is composed of
  - Interface definition
  - Model schema
- Redfish Interface (RESTful)
  - HTTP/HTTPS - protocol
  - JSON – format of content
- Redfish Models and Schema
  - Schema format for JSON
  - DMTF publishes the models for platforms and compute/servers

1OData is an OASIS Standard
2CSDL = Common Schema Definition Language
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
HTTP GET /redfish/v1/Systems/CS_1

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

**Simple properties**

- `Id`: "CS_1"
- `Name": "My Computer System",
- `SystemType": "Physical",
- `AssetTag": "free form asset tag",
- `Manufacturer": "Manufacturer Name",
- `Model": "Model Name",
- `SerialNumber": "2M2201005L",
- `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"},
  "PCleDevices": [ {"@odata.id": "/redfish/v1/Chassis/CS_1/PCleDevices/NIC" }],
  "PCleFunctions": [ {"@odata.id": "/redfish/v1/Chassis/CS_1/PCleDevices/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/PCleEndpoints/HostRootComplex1" }],
Redfish Model – Compute and Platform

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

- **Service Root**
  - `/redfish/v1`
    - **Root**
    - Tasks
    - Sessions
    - Accounts
    - Events
    - Registries
    - Schemas

- **Collection Resource**
  - `/redfish/v1/Systems`
    - Collection of Systems
      - "Logical view"
      - `/Systems/<id>`
      - Computer System
        - Processors
        - Memory
        - Disks
        - NICs
        - Power
        - Thermal
        - LogService
        - NW protocol

- **Single Resource**
  - `/Systems/<id>`
    - Computer System
    - Managed By
      - Computer Systems
        - `/redfish/v1/Chassis`
          - Collection of Chassis
            - "Physical view"
            - `/Chassis/<id>`
            - Chassis
              - Power
              - Thermal

- **Collection of Managers**
  - `/redfish/v1/Managers`
    - Collection of Managers
      - "Mgmt hierarchy"
      - `/Managers/<id>`
      - BMC
        - Managed By
          - Computer Systems
            - `/redfish/v1/Systems`
              - Collection of Systems
                - "Logical view"
                - `/Systems/<id>`
                - Computer System
                  - Managed By
                    - Computer Systems
                      - `/redfish/v1/Chassis`
                        - Collection of Chassis
                          - "Physical view"
                          - `/Chassis/<id>`
                          - Chassis
                            - Managed By
Storage Model

- Reuses chassis model
- Adds StorageServices & StorageSystems

Other SDC sessions
“Dip your Toe in the Water: A Swordfish Introduction”
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

Ethernet Switch (Phase 1)
- RFC6991 (YANG types)
- RFC7223 (Interfaces)
- RFC7224 (IANA Interface types)
- RFC7277 (IPv4 and IPv6)
- RFC7317 (system, system_state, platform, clock, ntp)
PCle Model

- The Fabric model is used to model PCle, SAS, and other Fabrics.
- A fabric includes collections of zones, endpoints, and switches.
- A switch includes a collection of ports.
- Fabric mockups exist for PCle, PCleMesh, and ComplexPCle.
Memory Model

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

- **Initiator**
  - Service Root
  - Systems
    - CS_1
  - Fabric
    - Fabric1
  - Endpoints
    - Initiator
    - Target

- **Target**
  - Zones
    - Zone1
  - Storage
    - Stor_1
    - Volumes
      - Vol_1
    - Drives
      - Drv_1

- Ethernet Interfaces
  - EIF_1
- Collection resource
- Singleton resource
- Subordinate resource
- Associated resource
- Bi-directional associated resource

**Notes**:
- Compute node
- Storage node
Redfish Tools

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

Redfish Service Implementation

- Redfish files
- DMTF open source
  - http://github.com/DMTF

- Tests
  - Service Conformance
  - Service Validator

- Client
  - Redfish Lib
  - Redfish Tool

- Working Service
  - Mockup Creator
  - Mockup Server (GET)
  - Profile Simulator
  - Interface Emulator (PATCH, POST)

- Redfish Lib

- Redfish Tools
  - Tools to enable Redfish modeling
  - Tools to enable Redfish clients
  - Tools being added regularly

- YANG (RFC)
  - YANG to Redfish Converter
- CSDL Validator
- OData CSDL
- CSDL to JSON-schema Converter
- JSON (mockup)
- CSDL to JSON-schema
- json-schema
- Document Generator
- Redfish Documentation

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

- DMTF open source
  - http://github.com/DMTF
Public Redfish Collateral

- Github
- Community Forum
- Developer's Hub
- Specs, presentation
- Redfish Forum (SPMF)

<table>
<thead>
<tr>
<th>Redfish Specification Forum</th>
<th>Redfish™ Developer Hub</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Specs, Protocol, Schemas and Tools</strong></td>
<td><strong>Welcome to the Redfish Developer Hub</strong></td>
</tr>
<tr>
<td><strong>Board</strong></td>
<td><strong>DMTF® API</strong> 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 using JSON, JSON-LD and OData, Redfish helps system administrators 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.</td>
</tr>
<tr>
<td><strong>Threads</strong></td>
<td><strong>Welcome Developers</strong></td>
</tr>
<tr>
<td>1</td>
<td><strong>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.</strong></td>
</tr>
<tr>
<td><strong>Posts</strong></td>
<td><strong>How to use the schema or API under Rev 1</strong></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td><strong>Last Post</strong></td>
<td><strong>By revia, Aug 15, 2020 at 10:30am</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Creating a RedfishServer/Controller: examples or examples for RedfishServer</strong></td>
</tr>
</tbody>
</table>

---

SDC 2017 Storage Developer Conference. © Distributed Management Task Force. All Rights Reserved.
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
- Academic research is underway (academic alliance partner members)
  - Texas Tech University - Cloud and Autonomic Computing Center
  - Barcelona Supercomputing Center
Thank you