OPEN. FOR BUSINESS.
OCP Profiles for Platform Hardware Management

John Leung
Intel Corporation - Data Center Group - Principal Engineer
OCP Incubation Committee Representative to the HW Management Project
OCP Profiles

• The vision: the OCP Profile specifies conformance requirements for an OCP submission as a checklist item
• Initially, the OCP Profile will contain the requirements for the manageability interface
  – Manageability interface is based on Redfish¹
  – OCP profile is specified using the Redfish Profile format
  – Conformance is tested with the Redfish open source tools

¹A manageability interface standard from the DMTF (dmtf.org)
Platform Manageability based on Redfish

- The OCP Hardware Management Project
  - Has approved the "OCP Baseline Hardware Management Profile"
  - Specifies the manageability common across OCP platforms
- Other OCP projects
  - Create platform level profiles by extending the "Baseline Hardware Management Profile"
  - To include platform specific requirements
- Start with the OCP Server platform
OCP Profile Status

• Server Project
  - Reviewing the "OCP Server Hardware Management Profile" v0.2.0

• Storage Project
  - There is interest in creating an "OCP Storage Hardware Management Profile"

• Rack and Power Project
  - Active work - "The majority of the descriptive work that we need for Rack & Power management has been already been done for the DCIM domain." - Mike
<table>
<thead>
<tr>
<th>Name</th>
<th>Status</th>
<th>Contributor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rack and Power Redfish Profile</td>
<td>In Development</td>
<td>Shared</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Activity</th>
<th>Target Completion Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Generate initial list of elements for the rack and power schema/mock-up</td>
<td>Done</td>
</tr>
<tr>
<td>2. Generate mock-up and test with validator to create initial feel of the validation process</td>
<td>Done</td>
</tr>
<tr>
<td>3. Consolidate additional elements needed for the schema/mock-up</td>
<td>Sept 7, 2018</td>
</tr>
<tr>
<td>- Upload updated checklist</td>
<td></td>
</tr>
<tr>
<td>- Solicit inputs from members</td>
<td></td>
</tr>
<tr>
<td>4. Schedule semi-monthly calls</td>
<td>By August Rack&amp;Power Meeting</td>
</tr>
<tr>
<td>5. Finalize process for determining the baseline for the Rack&amp;Power profile</td>
<td>Will be done on semi-monthly calls</td>
</tr>
</tbody>
</table>
Redfish Overview

- A modern RESTful interface for manageability
- Uses cloud/web protocols, structures, security models and tool chains
- Schemas are accessible so an interface can be introspected and enabled programmatically
- Models specified for managing datacenter platforms and devices (compute, storage, network, facilities)

**HTTP/S**

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

**Python code**

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

**Output**

```
1A87CA442K
```
Redfish interface and schema

- Redfish interface
  - HTTP/HTTPS - GET, POST, PATCH, DELETE
  - JSON – format of content

- Redfish model schema
  - Describes the content of the JSON response
  - DMTF develops the models for platforms and compute/servers
  - Other organization create models for their management domain

1 OData is an OASIS Standard
2 CSDL = Common Schema Definition Language
3 AKA Swagger
Redfish JSON response

- Obtained by issuing an HTTP GET
  - URL = /redfish/v1/Systems/<member>
- JSON response contains
  - Simple properties
  - Complex properties
  - References to subordinate resources
  - References to associated resources
  - Actions
  - References to schema
- Redfish is a hypertext model
  - Resources are accessible by traversing references

```json
{
  "@odata.context": "/redfish/v1/$metadata#ComputerSystem.ComputerSystem",
  "@odata.type": "#ComputerSystem.v1_3_0.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": "78899498CLF-7",
  "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/PCIeDevices/NIC" } ],
  "PCleFunctions": [ { "@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 Compute Model

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

Service Root
- /redfish/v1
  - Root

Collection Resource
- /redfish/v1/Systems
  - Collection of Systems
    - "Logical view"
- /redfish/v1/Chassis
  - Collection of Chassis
    - "Physical view"
- /redfish/v1-Managers
  - Collection of Managers
    - "Mgmt hierarchy"

Single Resource
- /Systems/<id>
  - Computer System
- /Chassis/<id>
  - Chassis
- /Managers/<id>
  - BMC

Compute
- Processors
- Memory
- Disks
- NICs

Platform HW Mgmt
- Power
- Thermal

Managed By
- Computer Systems

Open. For Business.
Capabilities of Compute Model

• **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
  - Change boot order and device
  - Configure BIOS settings
  - 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**
  - Compute (servers)
  - Physical hierarchy (rack/chassis/server/node)
  - Management hierarchy (rack mgr, tray mgr, BMC)

• **Security**
  - HTTPS
  - Map roles to privileges

• **Access and Notification**
  - Subscribe to published events
  - Inspect Logs
  - Host interface for in-band access

• **Composition**
  - Specific composition
  - Constrained composition

Red font = notable capabilities
Extending Redfish manageability

- The Redfish Forum enabling other SDOs to create and extend models into new management domains
  - Networked storage, storage services, and non-volatile storage (SNIA, NVMExpress)
  - Ethernet Switch (IETF) - map YANG to Redfish
  - Industrial IoT (PICMG)
  - Customer Premise Equipment (Broadband Forum)
  - BIOS interface (UEFI)
  - DC facilities infrastructure devices (The Green Grid, ASHRAE)
Conformance Testing with Redfish Tools

- Redfish Interop Validator
  - Runs conformance test against an implementation (pass/fail)
  - The Profile file is read to determine which tests to perform
  - Open source application\(^1\)

- Profile file
  - Specified by OCP projects
  - HW mgmt project - baseline requirements
  - Other projects - requirements beyond the baseline

\(^1\)github.com/DMTF/Redfish-Interop-Validator
OCP Redfish Profile file

- A JSON formatted file
- File contains requirements for
  - Supported protocols
  - Supported resources
    - Create, delete, etc.
  - Supported properties
    - read-only or read/write, Conditional, MinCount, Value, Action
"OCP Baseline Hardware Management Profile"

The server profile references the baseline profile¹

¹http://www.opencompute.org/wiki/Hardware_Management/SpecsAndDesigns#Baseline_and_Server_profile
OCP Profile Specification

- The specification is a readable version of the profile
- Contains:
  - Sample with requirements bold-faced
  - Requirement Tables

<table>
<thead>
<tr>
<th>Property</th>
<th>Rqmt</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UUID</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>RedfishVersion</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>AccountService</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>SessionService</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>Chassis</td>
<td>Mandatory</td>
<td></td>
</tr>
<tr>
<td>Managers</td>
<td>Mandatory</td>
<td></td>
</tr>
</tbody>
</table>
"OCP Baseline Hardware Management Profile"
Power and Thermal Resource

Service Root

- Chassis
  - Power
    - ≥ 1
  - Thermal
    - ≥ 1
- Managers
  - ≥ 1
- Account Service
- Session Service
  - ≥ 1
- Accounts

"OCP Server Hardware Management Profile"

1http://www.opencompute.org/wiki/Hardware_Management/SpecsAndDesigns#Baseline_and_Server_profile
"OCP Server Management Profile": System Resource

```json
{  "@odata.id": "/redfish/v1/Systems/1",  "Id": "1",  "Name": "My Computer System",  "SystemType": "Physical",  "AssetTag": "free form asset tag",  "Manufacturer": "Manufacturer Name",  "Model": "Model Name",  "SKU": "",  "SerialNumber": "2M220100SL",  "PartNumber": "",  "Description": "Description of server",  "UUID": "00000000-0000-0000-0000-000000000000",  "HostName": "web-srv344",  "Status": { "State": "Enabled", "Health": "OK", "HealthRollup": "OK" },  "IndicatorLED": "Off",  "PowerState": "On",  "Boot": {    "BootSourceOverrideEnabled": "Once",    "BootSourceOverrideMode": "UEFI",    "BootSourceOverrideTarget": "Pxe",    "BootSourceOverrideTarget@Redfish.AllowableValues": [ ... ],    "UefiTargetBootSourceOverride": "uefi device path"  },  "BiosVersion": "P79 v1.00 (09/20/2013)",  "ProcessorSummary": {    "Count": 8,    "Model": "Multi-Core Intel(R) Xeon(R) processor 7xxx Series",    "Status": { "State": "Enabled", "Health": "OK", "HealthRollup": "OK" }  },  "MemorySummary": {    "TotalSystemMemoryGiB": 16,    "MemoryMirroring": "System",    "Status": { "State": "Enabled", "Health": "OK", "HealthRollup": "OK" }  },  ...
```

System (continue)

```
```
Next Steps

• Attend the "OpenBMC Status Update" at 14:00
  - Open source implementation with Redfish support
• Participate in OCP project's efforts to create OCP profiles
  - Hardware Mgmt, Server, Rack & Power & Storage
• Test your platforms for conformance with the OCP Server Profile
  1. Setup Python execution environment
  2. Download the Redfish Interop Validator
  3. Execute the Validator with the OCP Server Profile