

Redfish Overview for ODIM TSC

John Leung

- Intel - Principal Engineer in DCG
- DMTF - Redfish Forum, VP of Alliances
- OCP - Incubation Committee (Rep to HW Mgmt Project)

DMTF's Redfish



- A modern RESTful interface for manageability
 - Uses cloud/web protocols, structures, and security models (HTTPS, JSON)
 - Datacenter can use existing development expertise
- Manageability is specified by resource URIs and the JSON response formats
- Machine-readable schemas for code-generation tool chains
 - Available in json-schema, OpenAPI, OASIS OData CSDL

HTTP/S

```
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'] )
```

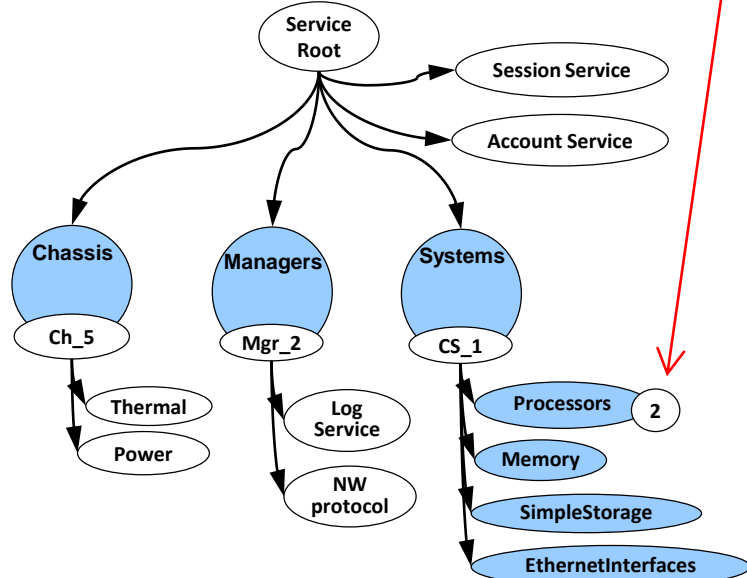
Output

```
1A87CA442K
```

Example Resource response - JSON



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



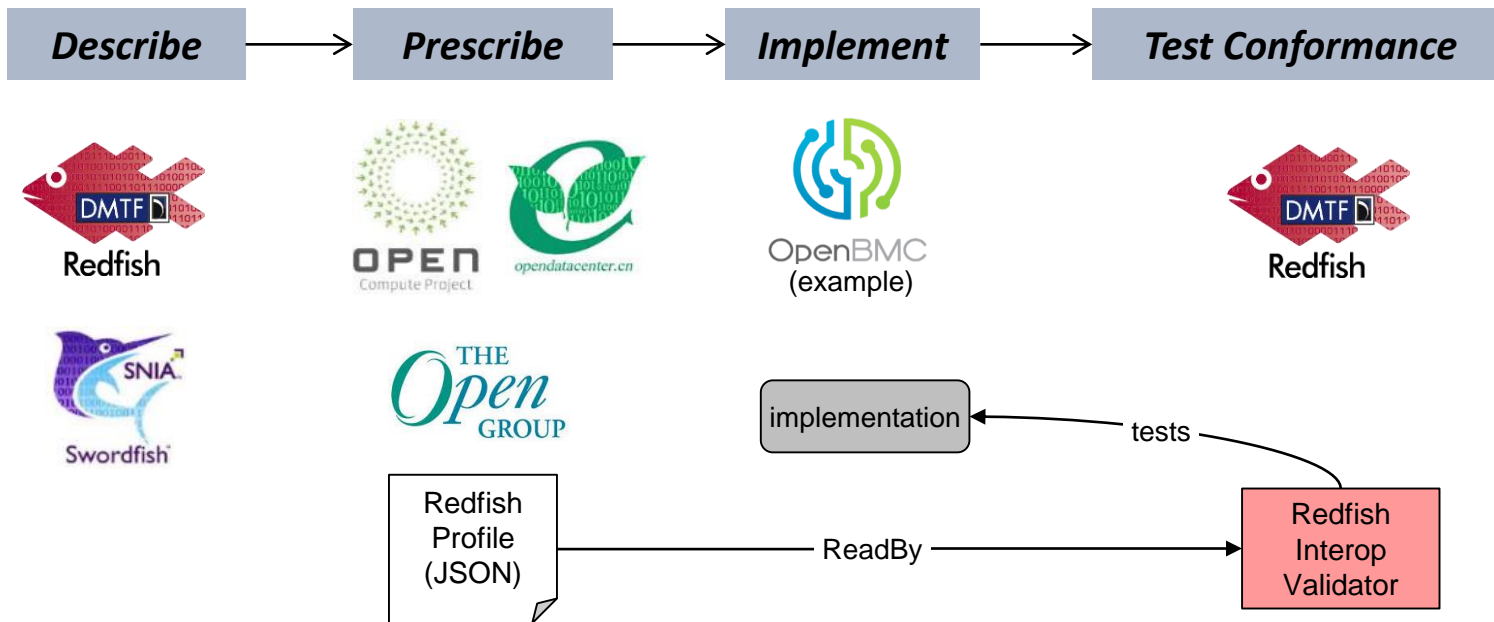
● Collection resource
 ○ Singleton resource

Simple properties
 Complex properties
 Subordinate resources
 Associated resources
 Actions

```

{
  "@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",
  "UUID": "00000000-0000-0000-0000-000000000000",
  "HostName": "web-srv344",
  "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",
                  "Links": {
                    "Chassis": [ { "@odata.id": "/redfish/v1/Chassis/Ch_1",
                      "ManagedBy": [ { "@odata.id": "/redfish/v1/Managers/Mgr_1",
                        "Endpoints": [ { "@odata.id": "/redfish/v1/Fabrics/PCle/Endpoints/HostRootComplex1",
                          "Actions": {
                            "#ComputerSystem.Reset": {
                              "target": "/redfish/v1/Systems/CS_1/Actions/ComputerSystem.Reset",
                              "@Redfish.ActionInfo": "/redfish/v1/Systems/CS_1/ResetActionInfo"
                            }
                          }
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  ]
}
    
```

From Specification to Conformance

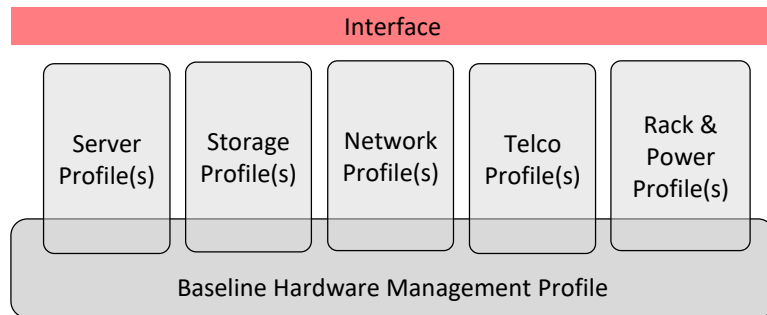


- Conformance to the Redfish interface is verified with Redfish Service Validator* and Redfish Service Conformance Check*
- Conformance of the Redfish model is verified with the Redfish Interop Validator*

OCP Platform Manageability based on Redfish

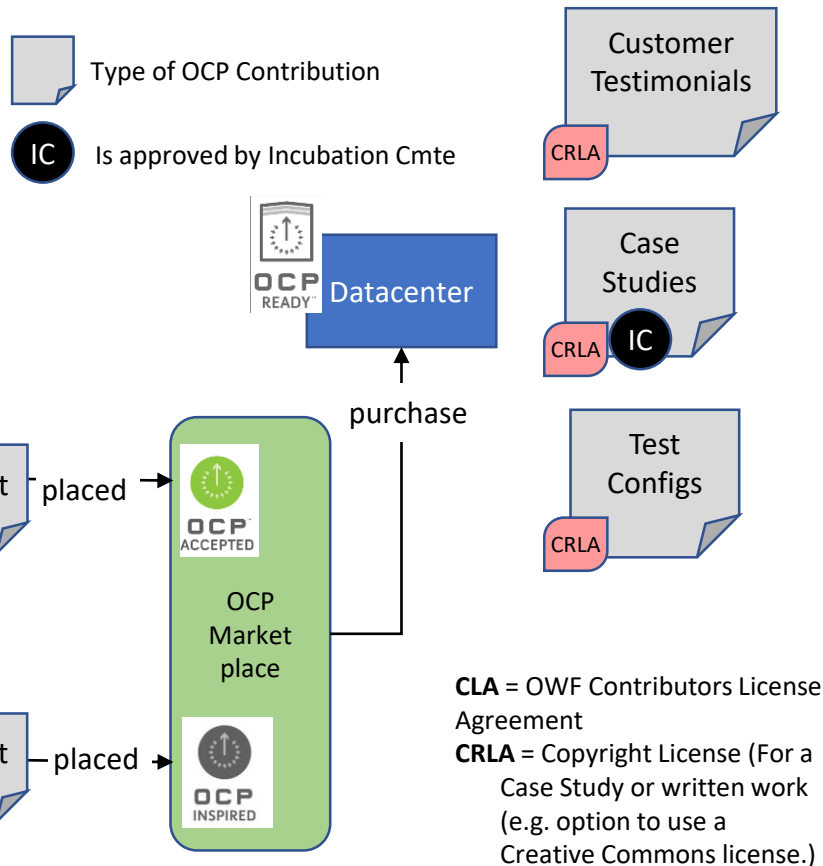


- The OCP HW Management Project created a baseline hardware management profile
 - Includes manageability common across OCP platforms
- Other OCP projects platform level profiles
 - May extend the baseline hardware management profile to include platform specific requirements



OCP Contribution Flow

- In the OCP Market Place, the OCP label is determined by whether the Manufacturer contributes a Product Design with their product
- The Design Files and a Product Design can be contributed concurrently



OpenRMC goals & status

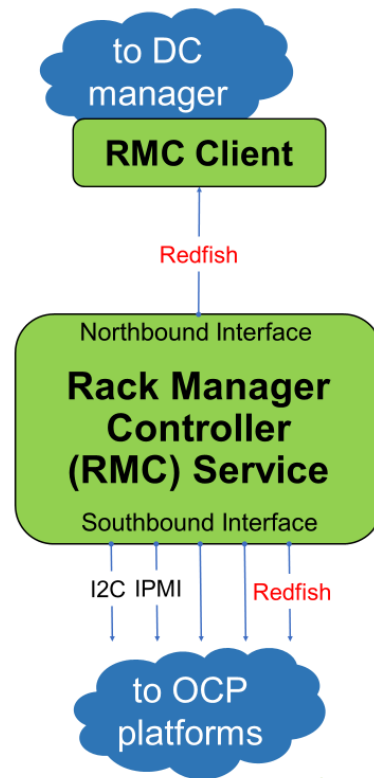
Specify the RMC northbound interface

- Northbound interface spec and profile approved at v1 approved

Host RMC Firmware contributions

- Hosting three FW-only contributions¹
- Received a FW/HW bundled contribution

¹github.com/opencompute/OpenRMC



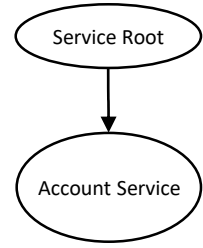
Capabilities of Compute Model



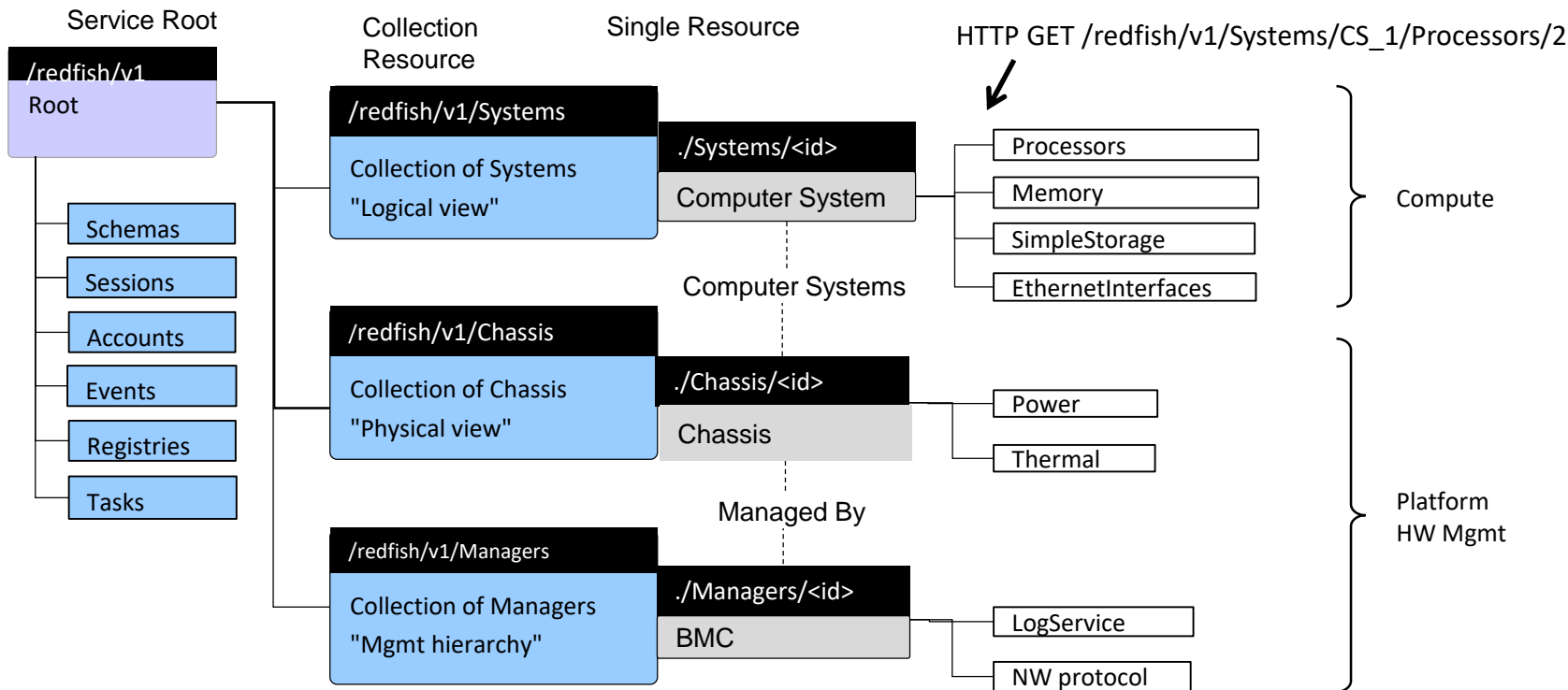
- **Chassis Information**
 - Identification and asset information
 - State and status
 - Temperature sensors and fans
 - Power supply, power consumption and thresholds
 - Set power thresholds
- **Compute Manageability**
 - Reboot and power cycle server
 - Configure BIOS settings
 - Change boot order and device
 - Update firmware (BIOS, BMC, etc)
 - 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**
 - HTTPS (certificate mgmt.)
 - Map roles to privileges
- **Access and Notification**
 - Subscribe to published events
 - Inspect Logs
 - Host interface for in-band access
- **Composition**
 - Specific composition
 - Constrained composition

Redfish Services

- Account Service - roles and privileges
- Event Service - asynchronous events (SSE for stream)
- Session Service - connections to the Redfish Service
- Task Service - tasks created by the Redfish Service
- Job Service - jobs are created by the user
- Update Service - firmware inventory & update and software inventory
- Certification Service - manage component and system certificates
- Composition Service - create components and systems
- Aggregation Service - manage a group of resources, as a single entity



Redfish System Model

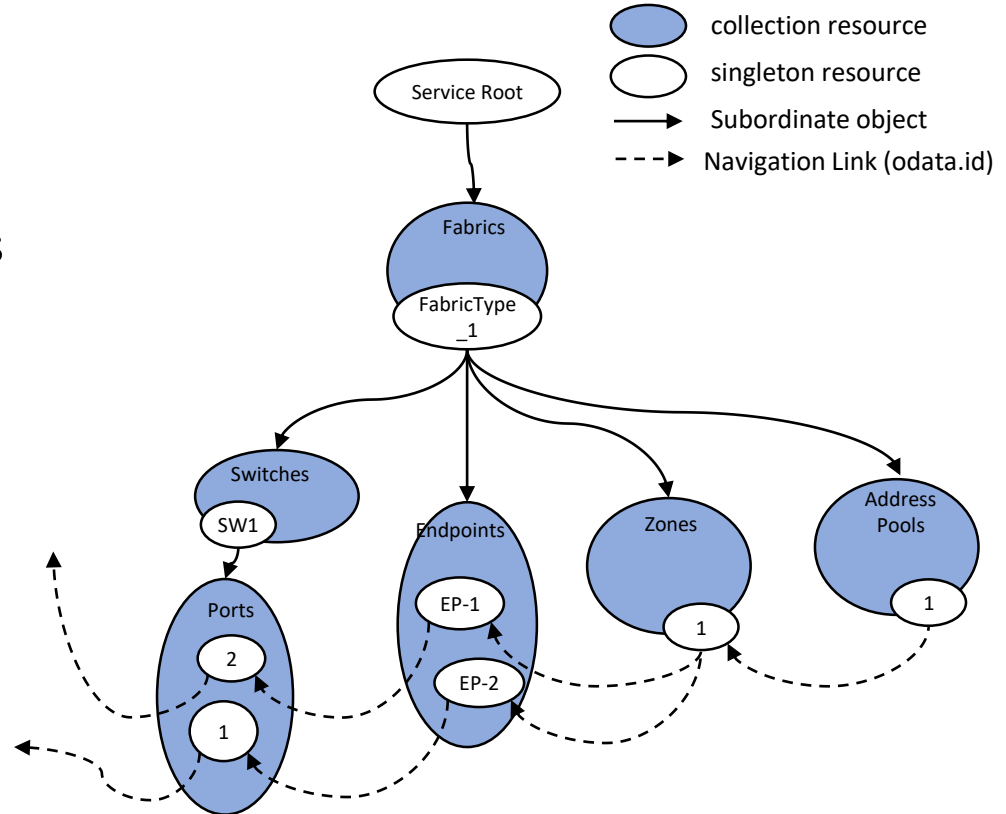


Redfish Models

- Fabric Model
 - Redfish Client manages a connectivity between components & systems
- Aggregation Model
 - Redfish Client manages a group of resources, as a single entity
 - Groups may be temporary or persistent; may be homogeneous or heterogeneous
- Composability Model
 - Redfish Client creates a system from components

Fabric Model

- The Fabric has a **zone** which contains two **endpoints**
- The **endpoints** are manifested as **ports** on a **switch** (SW1)
- The **endpoints** have addresses which are within the **address pool**



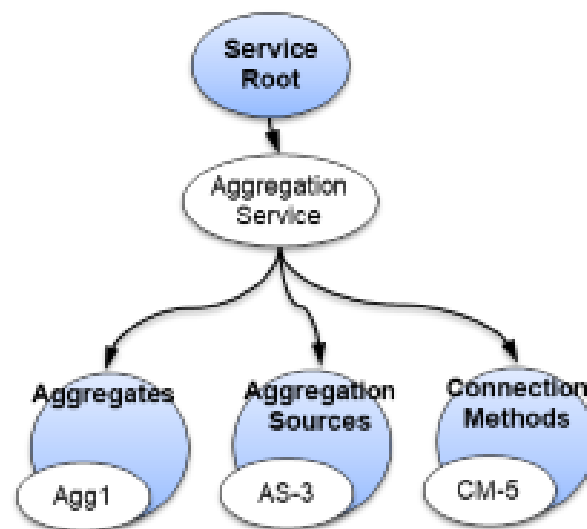
Aggregation Model

AggregationService

- Supports persistent and temporary aggregates (i.e. groups)
- Supports temporary aggregates within a Action parameter

Info access the system/platform which is part of the aggregate

- **AggregationSource** resource represent the sources of information for a subset of the resources provided by a Redfish service. It can be thought of as a provider of information. Includes info for gathering of information like address and account used to access the information.
- **ConnectionMethod** resource describes the protocol, provider, or other method used to communicate to a given access point for a Redfish aggregation service.



Redfish Feedback & Contributions

- Bulletin Board - redfishforum.com
 - Public community discussion and Q&A
- Developer Hub - redfish.dmtf.org
 - One-stop, in-depth technical resource
 - Mockups
 - Education (whitepapers, presentations, YouTube shorts)
 - Community
- DMTF Github - github.com/dmtf
 - Tools to simulate and emulate a Redfish service
 - Libraries and toolbox for Redfish client development
 - Tests for Redfish implementations
 - Tools for creating Redfish models
- Redfish webpage - dmtof.org/redfish
 - Latest work from the Redfish Forum



Redfish Specification Forum			
Home Help Search Welcome Guest, Please Login or Register			
Redfish Specification Forum Home			
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. Moderator: Admin	1	2	Retrieving individual properties by ZhiLiang Sep 12, 2018 at 7:42am
 CSDL and json-schema Discussions 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 1 by mraimani Aug 12, 2018 at 6:33am
 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/OCF over-IP session for user by Jandur



DMTF 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

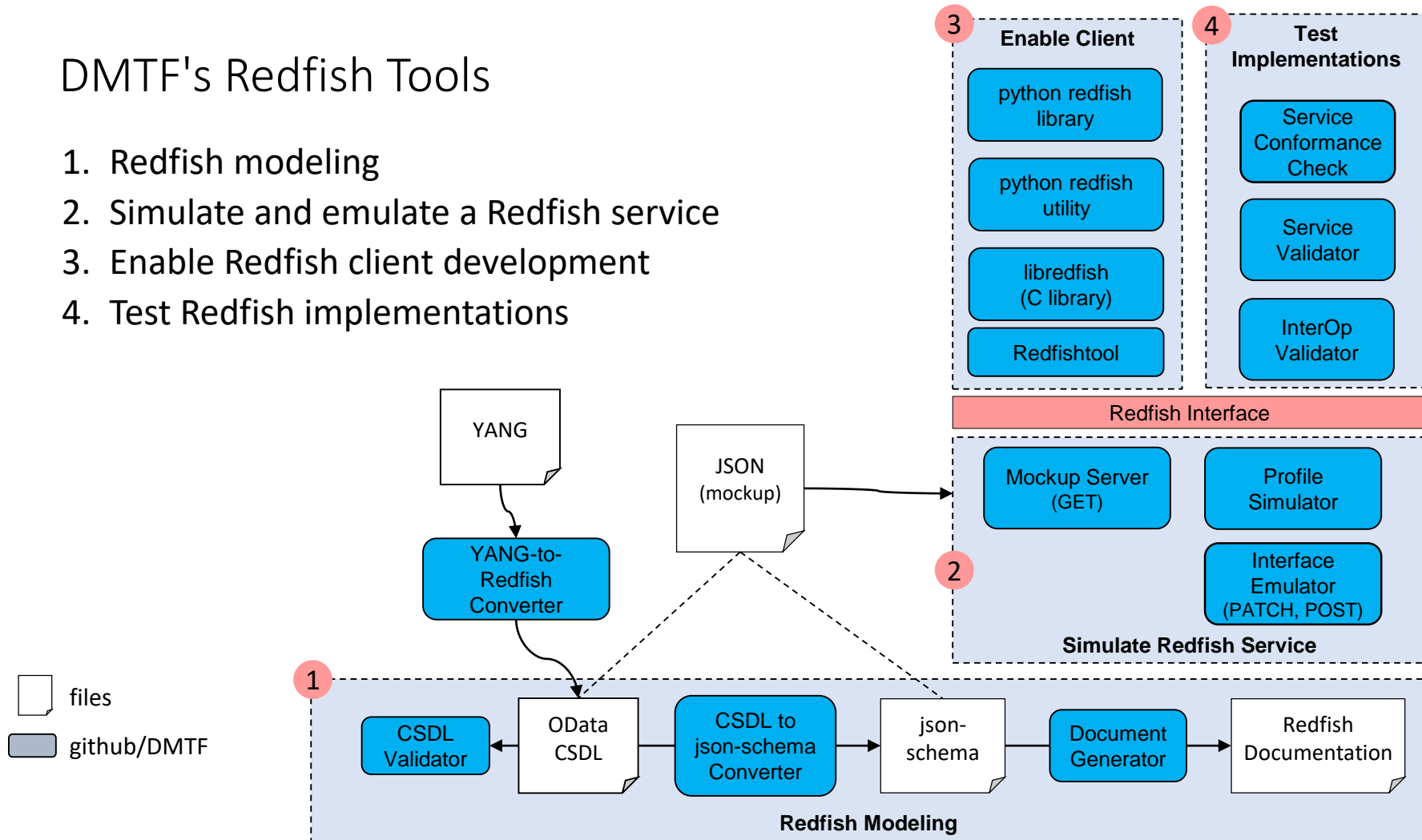
The DMTF's Redfish Developer Hub is a one-stop, in-depth technical resource – by developers, for developers – designed to provide all the files, tools, community support, tutorials and other advanced education you may need to help you use Redfish.



Thank You

DMTF's Redfish Tools

1. Redfish modeling
2. Simulate and emulate a Redfish service
3. Enable Redfish client development
4. Test Redfish implementations



Extending Redfish manageability

- Domain experts from other SDO's are extending Redfish
- Networked storage, storage services, and non-volatile storage (**SNIA, NVMExpress**)
- Ethernet Switch - map YANG to Redfish
- BIOS interface (**UEFI**)
- DC facilities infrastructure devices (**The Green Grid, ASHRAE**)
- Industrial IoT (**PICMG**)
- Customer Premise Equipment (**Broadband Forum**)

