



Introduction to Redfish

Jeff Autor, co-Chair
DMTF Scalable Platforms Management Forum
July 2015



Disclaimer

- The information in this presentation represents a snapshot of work in progress within the DMTF.
- This information is subject to change without notice. The standard specifications remain the normative reference for all information.
- For additional information, see the Distributed Management Task Force (DMTF) website.



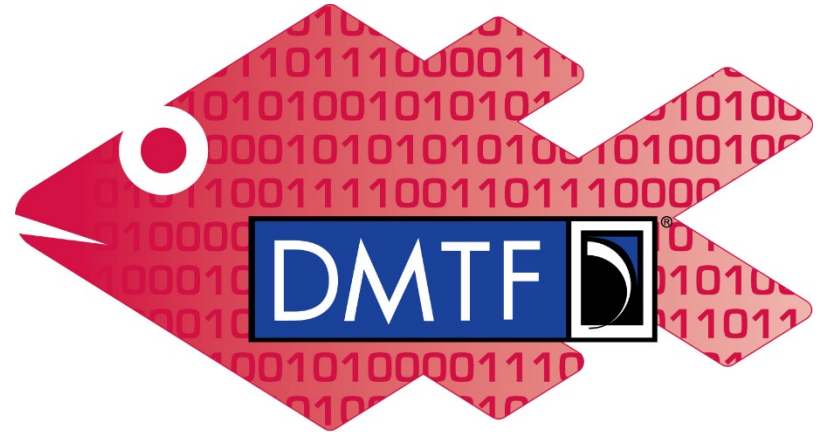
Scalable Platforms Management Forum

- Created in September 2014 – now 16 member companies
- Co-Chairs: Jeff Autor (HP), Paul Vancil (Dell)
- Promoters: Broadcom, Dell, Emerson, Hewlett-Packard, Intel, Lenovo, Microsoft, Supermicro, VMWare
- Supporters: AMI, Fujitsu, Huawei, Mellanox, Oracle, Seagate
- Charter: *Create and publish an open industry-standard specification and schema that meets the expectations of Cloud and Web-based IT professionals for scalable platform hardware management utilizing existing tool chains as well as being usable by personnel with minimal experience.*
- Technology submissions from RedfishSpecification.org and Microsoft formed the basis of the forum's work
- Forum leveraging the “Redfish” name for the public specification
 - Significant press and industry usage over the last year
 - Existing domain names and logo support DMTF effort



Redfish Scope and Goals

- *The DMTF's Scalable Platforms Management Forum (SPMF) is working to create and publish an open industry standard specification and schema that meets the expectations of end users for simple, modern and secure management of scalable platform hardware.*



Redfish

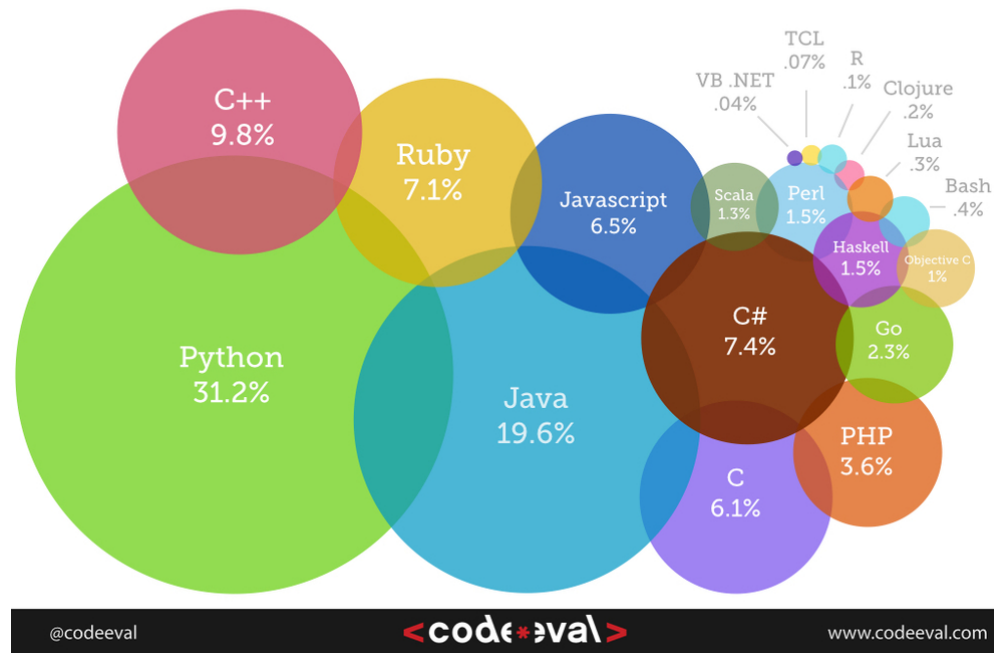
- RESTful interface over HTTPS in JSON format based on OData v4
- Usable by client applications and browser-based GUIs
- A secure, multi-node capable replacement for previous interfaces
- Schema-backed human-readable output
- Covers popular use cases and customer requirements
- Intended to meet OCP Remote Machine Management requirements



Why REST, HTTP and JSON?

- **REST:** The API architecture
 - Rapidly replacing SOAP
- **HTTPS:** The Web protocol
 - Well-understood by admins
 - Known security model
 - Known network configuration
- **JSON:** Modern data format
 - Human-readable
 - Simpler than XML
 - Modern language support

Most Popular Coding Languages of 2015



- The combination of language support and ubiquity of REST, HTTP and JSON means that systems management tasks can be performed using the same skill set and tool chain as all other IT and dev/ops tasks.



How simple is REST using JSON?

Example Python code to retrieve serial number from a server:

```
rawData = urllib.urlopen('http://192.168.1.135/redfish/v1/Systems/1')
jsonData = json.loads(rawData)
print( jsonData['serialNumber'] )
```

Output is:

```
1A87CA442K
```

*Example uses Redfish v0.96 ComputerSystem resource



Redfish v1.0 Specification & Schema

Retrieve “IPMI class” data

- Basic server identification and asset info
- Health state
- Temperature sensors and fans
- Power supply, power consumption and thresholds

Discovery

- Service endpoint (network-based discovery)
- System topology (rack/chassis/server/node)

Basic I/O infrastructure data

- Host NIC MAC address(es) for LOM devices
- Simple hard drive status / fault reporting

Security

- Session-based leverages HTTPS

Perform Common Actions

- Reboot / power cycle server
- Change boot order / device
- Set power thresholds

Access and Notification

- Serial console access via SSH
- Event notification method(s)
- Logging method(s)

BMC infrastructure

- View / configure BMC network settings
- Manage local BMC user accounts



Expected Open Source Efforts

Client Library

- Common utility support functions
 - Discovery, Enumeration, etc.
 - Event subscription
- Typical tasks
 - Power on/off/reboot
 - Gather thermal data
- Languages under consideration
 - Python
 - Java
 - PowerShell
 - Other possibilities...

Command Line Utility

- Similar to IPMItool
- Designed for end users
- Calls Client library
- Likely written in Python

Conformance Test Suite

- Schema-aware tool for testing
- Checklist for vendors and customers
- Avoid spec interpretation conflicts



Progress towards a Redfish v1.00 release

- v0.93 *Work in Progress* released in October 2014
 - Packaging of technology submissions for feedback
- v0.94 *Work in Progress* released in January 2015
 - Aligned the JSON payload with OData v4 constructs
- v0.95 *Work in Progress* released March 31st
 - Completed the key collections and resources
- v0.96 *Work in Progress* released April 28th
 - Entire data model stable (only bug fixes beyond)
- v0.99 *Work in Progress* released July 2nd
 - Specification clean-up, additional schema text
 - Same as the v1.00 artifacts delivered to TC
- **v1.00 Specification** (target publication: August 2015)
 - Final specification submitted to Technical Committee June 26th

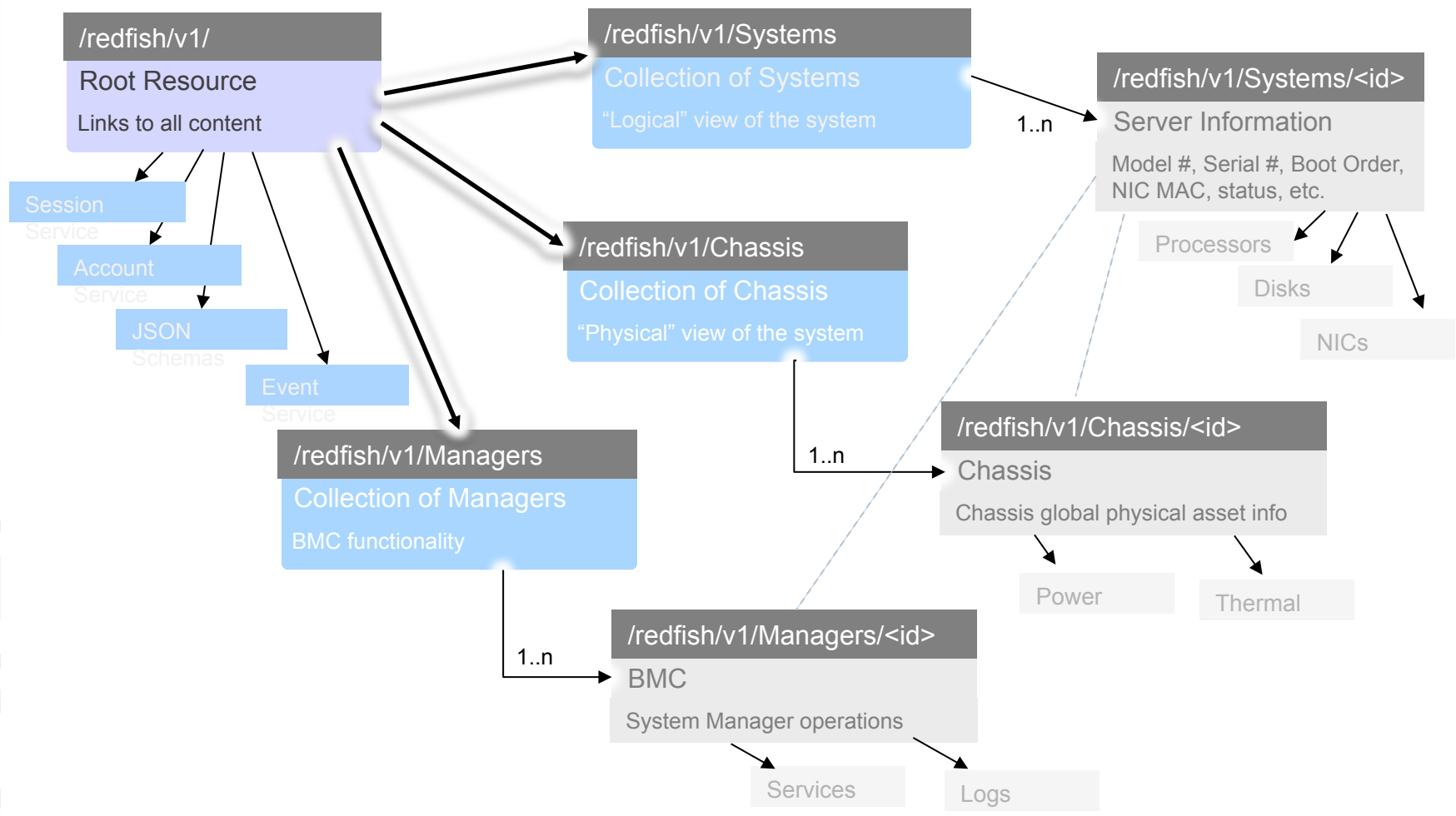


Introduction to the Redfish data model

- All resources linked from a Service Entry point (root)
 - Always located at URL: /redfish/v1/
- Major resource types structured in 'collections' to allow for standalone, multi-node, or aggregated rack-level systems
 - Additional related resources fan out from members within these collections
- **ComputerSystem**: properties expected from an OS console
 - Items needed to run the “computer”
 - Roughly a logical view of a computer system as seen from the OS
- **Chassis**: properties needed to locate the unit with your hands
 - Items needed to identify, install or service the “computer”
 - Roughly a physical view of a computer system as seen by a human
- **Managers**: properties needed to perform administrative functions
 - aka: the systems management subsystem (BMC)



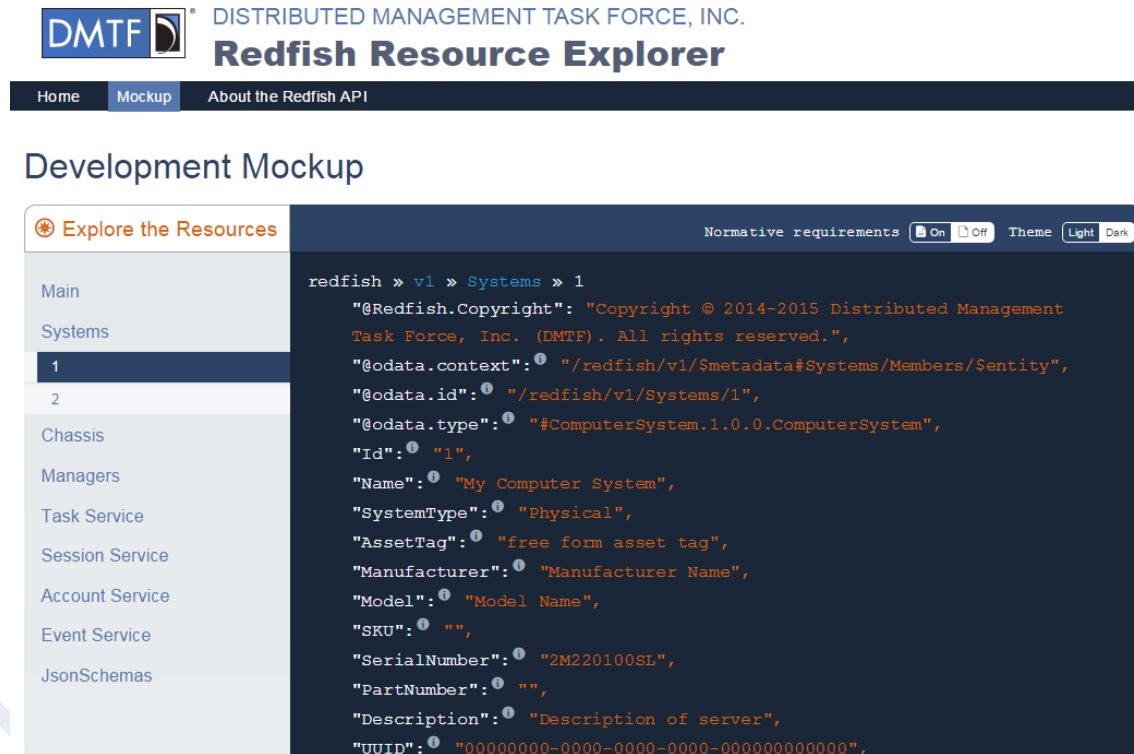
Resource map (highlights)





Redfish Resource Explorer

- Browser-based Educational tool part of the DMTF web site for Redfish
- Explore “mockups” of the Redfish data model
- Navigate via links through the model to various resources
- Text descriptions are taken directly from the schema files for consistency



The screenshot displays the Redfish Resource Explorer interface. At the top, the DMTF logo and the text "DISTRIBUTED MANAGEMENT TASK FORCE, INC. Redfish Resource Explorer" are visible. Below this is a navigation bar with "Home", "Mockup", and "About the Redfish API" links. The main content area is titled "Development Mockup" and features a sidebar on the left with a tree view of resources: Main, Systems (selected), Chassis, Managers, Task Service, Session Service, Account Service, Event Service, and JsonSchemas. The main pane shows a breadcrumb "redfish » v1 » Systems » 1" and a JSON representation of a system resource. The JSON includes fields such as "@Redfish.Copyright", "@odata.context", "@odata.id", "@odata.type", "Id", "Name", "SystemType", "AssetTag", "Manufacturer", "Model", "SKU", "SerialNumber", "PartNumber", "Description", and "UUID".

```
redfish » v1 » Systems » 1
"@Redfish.Copyright": "Copyright © 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": "2M220100SL",
"PartNumber": "",
"Description": "Description of server",
"UUID": "00000000-0000-0000-0000-000000000000",
```



Resource Explorer Demo





How to participate – we want your input!

1. Download the “Work in Progress” from the Redfish site:

<http://www.dmtf.org/standards/redfish>

2. Join the SPMF!

By Joining the DMTF and SPMF, you can shape the standard

<http://www.dmtf.org/join/spmf>

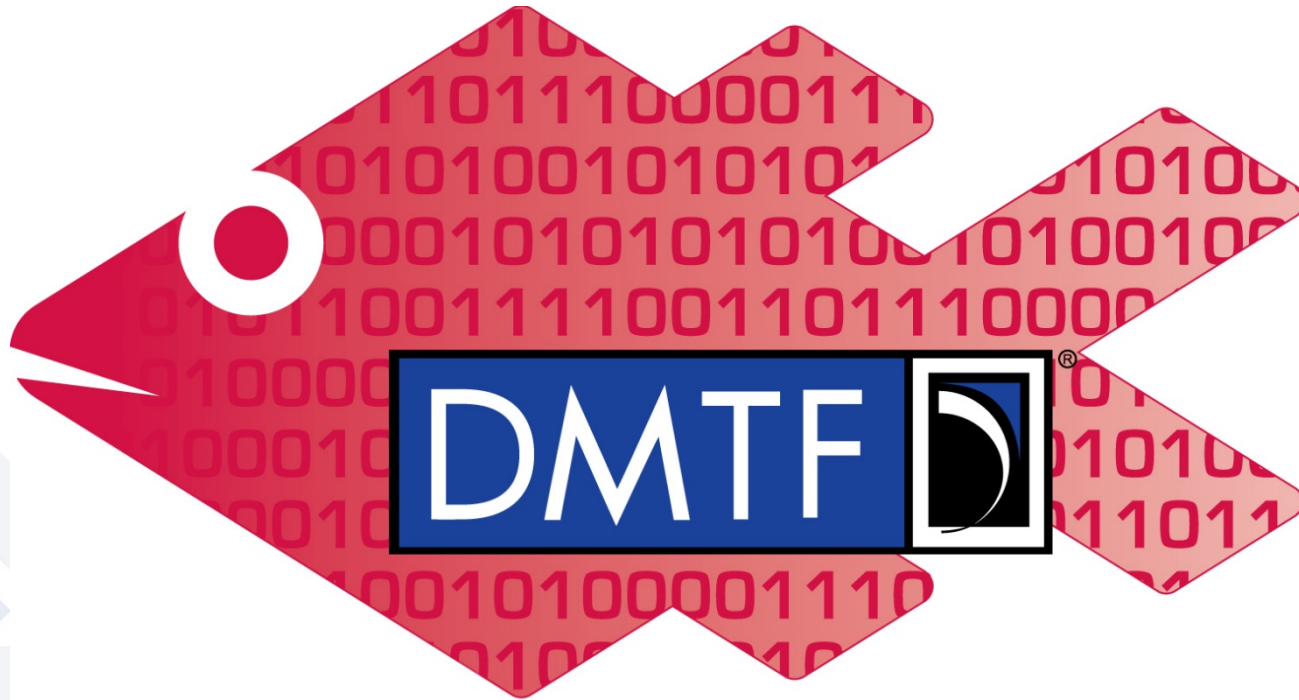
3. DMTF feedback portal, providing feedback on “Work In Progress”

As the group produces “Works In Progress”, you can provide feedback at

<http://www.dmtf.org/standards/feedback>



Q&A & Discussion



Redfish