Redfish – Simple and Secure Management for Converged, Hybrid IT

Introduction

DMTF's Redfish® is a standard designed to deliver simple and secure management for converged, hybrid IT and the Software Defined Data Center (SDDC).

Both human readable and machine capable, Redfish leverages common Internet and web services standards to expose information directly to the modern tool chain.

Delivering both in-band and out-of-band manageability, Redfish continues to expand to address customer use cases and technology for a holistic data center management experience. This work includes data center infrastructure management (DCIM) power and cooling schemas, as well as YANG mapping that extends Redfish into managing the network device domain. In addition, the Storage Networking Industry Association (SNIA), a long-time DMTF Alliance Partner, is building on Redfish's local storage management capabilities with its Swordfish standard, which addresses enterprise storage services.

This Technical Note provides an overview of the standard. Developers are encouraged to visit the Redfish Developer Hub - http://redfish.dmtf.org - for more in-depth information.

Why Redfish?

Until Redfish, interoperable management standards were lacking for modern data center environments. As organizations shift to scale-out solutions, legacy standards are insufficient to successfully manage numerous simple and multi-node servers or hybrid infrastructures.

IPMI, an older standard for out-of-band management, is limited to a "least common denominator" set of commands (e.g. Power On/Off/Reboot, temperature value, text console, etc.), forcing customers to use a reduced set of functionality because vendor extensions are not common across all platforms. Many users developed their own tools for tight integration, often having to rely on in-band management software.

Seeking a modern interface that builds on widely-used tools to accelerate development, today's customers demand a well-defined standard that uses the protocols, structures and security models that are common in Internet and web services environments.

Designed from its inception to provide interoperable management for converging infrastructures, Redfish delivers powerful simplicity that integrates within existing tool chains.

DMTF's Redfish Standard

An open industry standard specification and schema, Redfish specifies a RESTful interface and utilizes defined JSON payloads - usable by existing client applications and browser-based GUI. Redfish also supports OpenAPI, an open specification that describes API services and offers a rich ecosystem of tools for developers and end users.

Release 1 of Redfish focused on servers, providing a secure, multi-node capable replacement for IPMI-over-LAN. Subsequent Redfish releases have added management of network interfaces (such as NIC, CNA, and FC HBA), PCIe switching, local storage, NVDIMMs, multifunction adapters, composability, and telemetry. The standard also provides eventing, firmware and standardized push-style software updates, as well as certificate management and privileges mapping for security.

In addition, the Redfish Host Interface Specification allows applications and tools running on an Operating System - including in the pre-boot (firmware) stage – to communicate with the Redfish management service.

In defining the Redfish standard, the protocol is separate from the data model, allowing them to be revised independently. The schema-based data model is scalable and extensible, and will continue to develop with additional human-readable definitions as the industry evolves.

Why REST, HTTP and JSON?

Combining language support with the ubiquity of REST, HTTP and JSON, Redfish enables IT management tasks to be performed using the same skill set and tool chain as other IT and dev/ops tasks.

RESTful protocols are rapidly replacing SOAP as the cloud ecosystem is adopting REST, and the web API community has followed suit. RESTful protocols are much quicker to learn than SOAP, and they have the simplicity of being a data pattern (as REST is not strictly a protocol) mapped to HTTP operations directly.

HTTP, with its commonly used security model and network configuration, is well-understood by admins.

JSON is fast becoming the modern data format. It is inherently human readable, more concise than XML, has a plethora of modern language support and is the most rapidly growing data format for web service APIs.

JSON has an additional advantage for embedded manageability environments: most Baseboard Management Controllers (BMCs) already support a web server and managing a server through a browser is common (typically via a Java script-driven interface). By utilizing JSON in Redfish, the data from a Redfish service is viewed directly in the browser, ensuring the data and the programmatic interface is uniform in semantics and value.

In addition to REST, HTTP and JSON, Redfish adopts common OData v4 conventions for describing schema, URL conventions, and naming, as well as the structure of common properties in a JSON payload. This further enables Redfish services to be consumed by a growing ecosystem of generic client libraries, applications, and tools.

How simple is it?

The graphic below shows example Python code used to retrieve the serial number from a server using Redfish:

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

The output in this example would look like:

```
1A87CA442K
```

Basic Concepts

In Redfish, every URL represents a resource, a service, or a collection of resources. In RESTful terms, these Uniform Resource Identifiers (URIs) point to resources and clients interact with resources.

The resource format is defined by the Redfish Schema, which the client can use to determine the correct semantics, if needed (Redfish semantics are designed to be largely intuitive).

The Redfish Schema is defined in three formats. It is defined in the OData Common Schema Definition Language (CSDL), so generic OData tools and applications can interpret it. The schema is also defined in the JSON Schema format for other environments, such as Python scripts, JavaScript code and visualization. In addition, the schema is provided in YAML, as specified by OpenAPI.

In Redfish, all resources are linked from a Service Entry point (root), which is always located at URL: /redfish/v1.

Major resource types are structured in "collections" to allow for standalone, multi-node, or aggregated rack-level systems. Additional related resources fan out from members within these collections.

Collections

In Redfish, a collection represents a group of similar resources. Examples include Systems, Managers and Chassis.

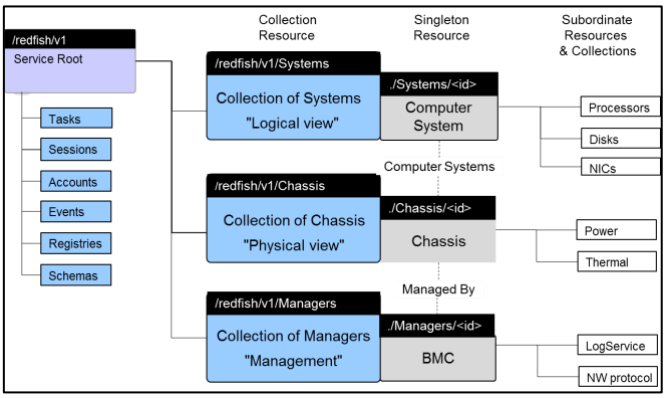

Fig 1: Redfish Resource Map Highlights

A System represents the logical view of a computer system. Any subsystem accessible from the host CPU is represented in a System resource. Each System instance will have CPUs, memory and other components. Computer systems are contained as members of the Systems collection.

The Managers collection contains BMCs, Enclosure Managers or any other component managing the infrastructure. Managers handle various management services and can also have their own components (such as NICs).

The Chassis collection contains resources that represent the physical aspects of the infrastructure. A single Chassis resource can house sensors, fans and the like. Racks, enclosures and blades are examples of Chassis resources included in the Chassis collection. Redfish also provides a method to represent a

Chassis contained within another Chassis.

Open Approach to Development

DMTF welcomes open source contributions to the Redfish ecosystem, as well as collaborative review and feedback on the standard from the industry at large. The organization has released numerous open source tools for testing, validation and other areas on its public GitHub repository - https://github.com/DMTF/.

Conclusion

As the management of platforms and devices continues to overlap and converge in increasingly hybrid IT environments, Redfish enables the management of compute, network, storage and facilities equipment using the same simple interface.

As a result of broad industry collaboration, Redfish reduces vendor lock-in and increases the productivity of system administrators.

Recommended Resources

- Redfish Developer Hub – http://redfish.dmtf.org
- "Redfish School" YouTube Tutorials – https://www.youtube.com/dmtforg
- Redfish Standard Page – www.dmtf.org/standards/redfish

Acknowledgements

Work on the Redfish standard takes place in DMTF's Redfish Forum (the group formerly known as SPMF - http://dmtf.org/standards/spmf). Redfish Forum members contributed to this Technical Note.

Redfish Forum Promoting Members include Cisco; Dell Technologies.; Ericsson AB; Hewlett Packard Enterprise; IBM; Inspur; Intel Corporation; Lenovo; Supermicro; Verizon; Vertiv; and VMware, Inc.

Redfish Forum Supporting Members include Advanced Micro Devices: Alibaba (China) Co., Ltd; American Megatrends, Inc.; ARM, Inc.; Atos; Eaton; Fujitsu; Google LLC; Huawei; Jabil; KIOXIA Corporation; Meta; Microchip Technology Inc.; NetApp; NVIDIA Corporation; Phoenix Technologies Ltd.; Quanta Computer Inc.; Seagate Technology; and Xilinx, Inc..