

SNIA DEVELOPER CONFERENCE



BY Developers FOR Developers

September 16-18, 2024
Santa Clara, CA

Redfish for Storage Management

Presented by Jeff Hilland
President, DMTF
Distinguished Technologist, Hewlett Packard Labs

Disclaimer

- The information in this presentation represents a snapshot of work in progress within DMTF & SNIA
- This information is subject to change without notice.
- For additional information, see the DMTF & SNIA websites: www.snia.org/swordfish, www.dmtf.org



Agenda

- DMTF Background
 - Who we are and what we do
- Redfish
 - Local Storage Model
 - Fabric Model for Networks (CXL)
 - Log changes: Auditing, Debug
 - Storage Metrics
 - Replacement for Power/Thermal: EnvironmentalMetrics

DMTF – An Industry Standards Organization

WHO Led by innovative, industry-leading companies, DMTF has a global presence with members in multiple countries.

WHAT DMTF standards support diverse emerging and traditional IT infrastructures including cloud, virtualization, network, servers and storage. A complete list is available at www.dmtf.org/standards.

WHY Nationally and internationally recognized by ANSI and ISO, DMTF standards enable a **more integrated and cost-effective approach to management through interoperable solutions**.

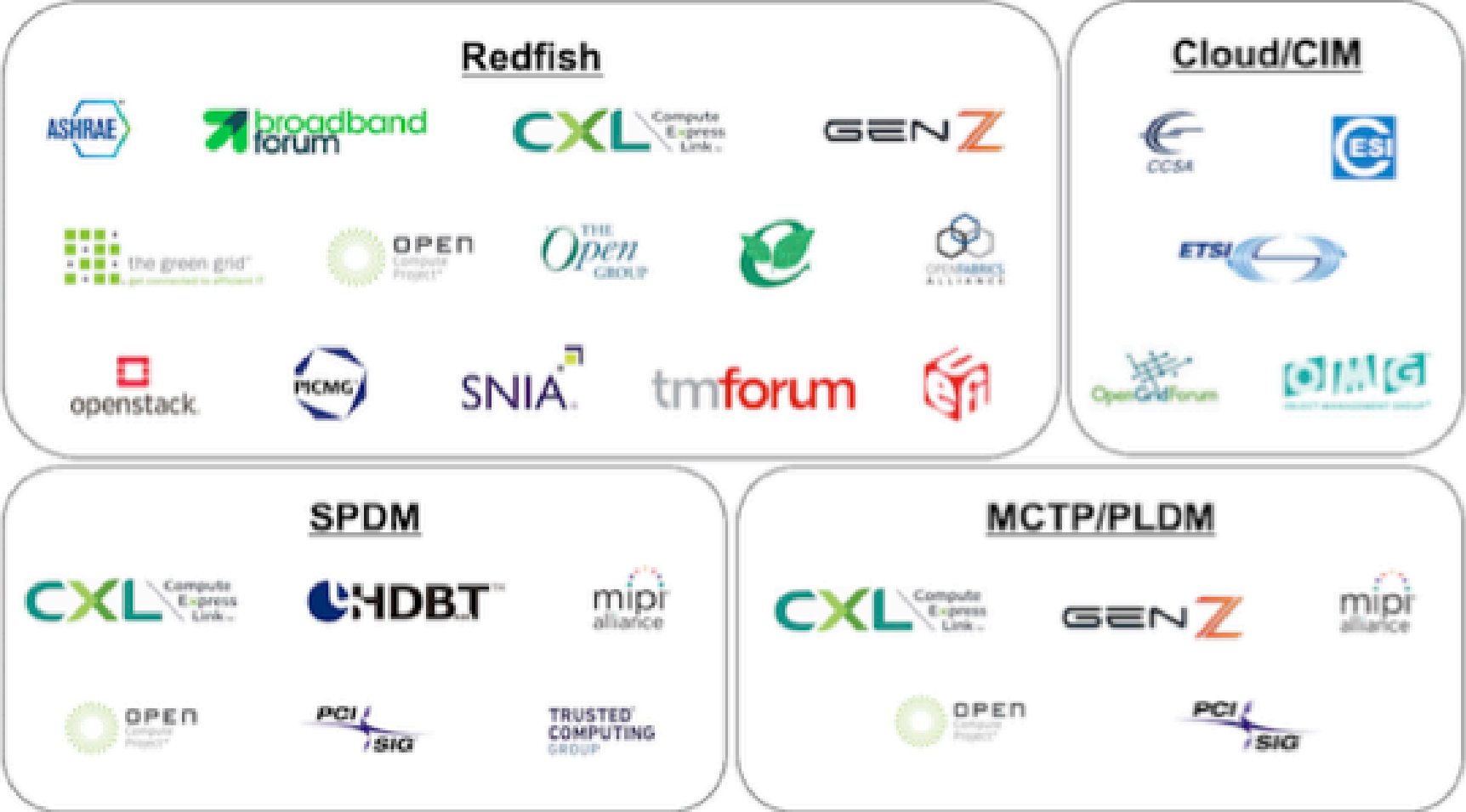
HOW Simultaneous development of Open Source and Open Standards is made possible by DMTF, which has the support, tools and infrastructure for efficient development and collaboration.

DMTF Board Member Companies



DMTF Alliance Partnership

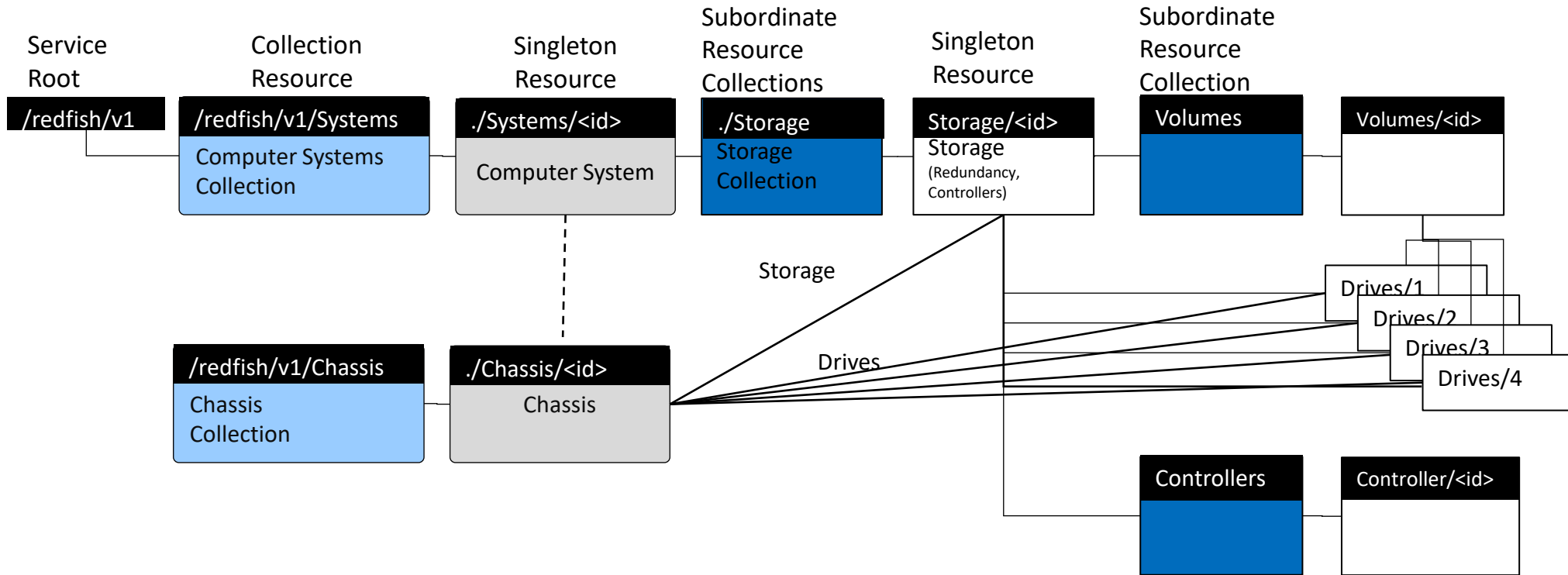
DMTF and its Alliance Partners develop a common dialogue and work together for the good of the industry, avoiding overlap and helping ensure interoperability



DMTF does more than Redfish

- **SMBIOS is everywhere**
- **PMCI**
 - MCTP & PLDM: FW Update, Monitoring & Control, FRU, RDE, PLDM, Mappings and Bindings, and NC-SI
 - PMCI Tools TF
- **SPDM WG**
 - Leveraged by PCIe, OCP, JEDEC, MIPI, CXL, HDBaseT and others
 - Provides Authentication, Attestation and Encryption Key Exchange
 - Includes mapping for MCTP & encrypted MCTP
 - Adding PQC Support
 - Security Code TF
 - Libspdm on GitHub has thousands of clones monthly
- **CIM**
 - Consolidated efforts under a single CIM Forum

Storage in Redfish: the basics



Note that the Volumes are in Collections off of the Storage resource, drives are in arrays off of the chassis and associated with storage resource (URI can be in either location).

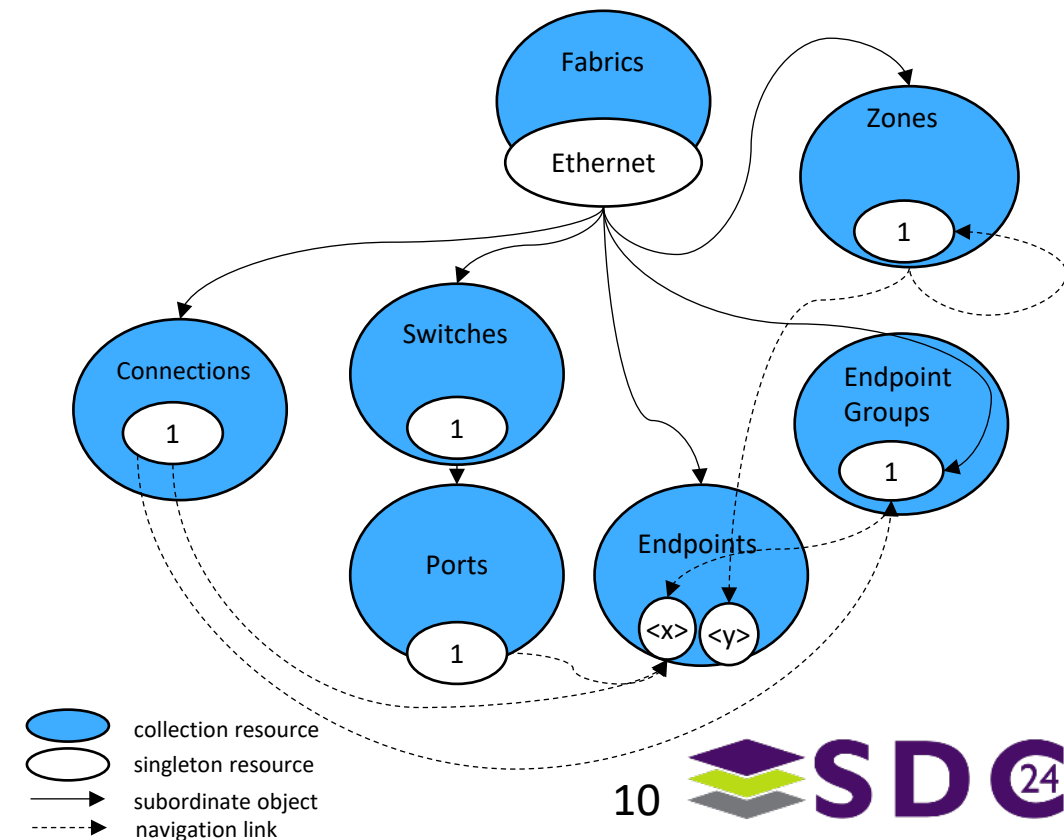
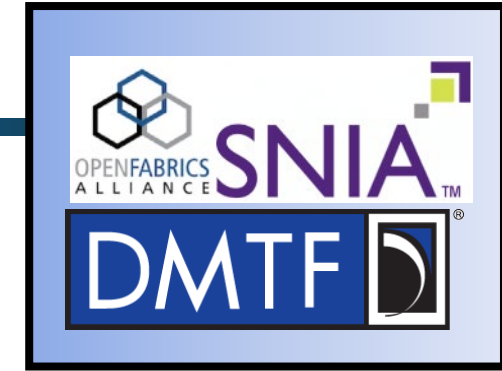
Controllers can either be an object in Storage or their own object (like for NVMe)



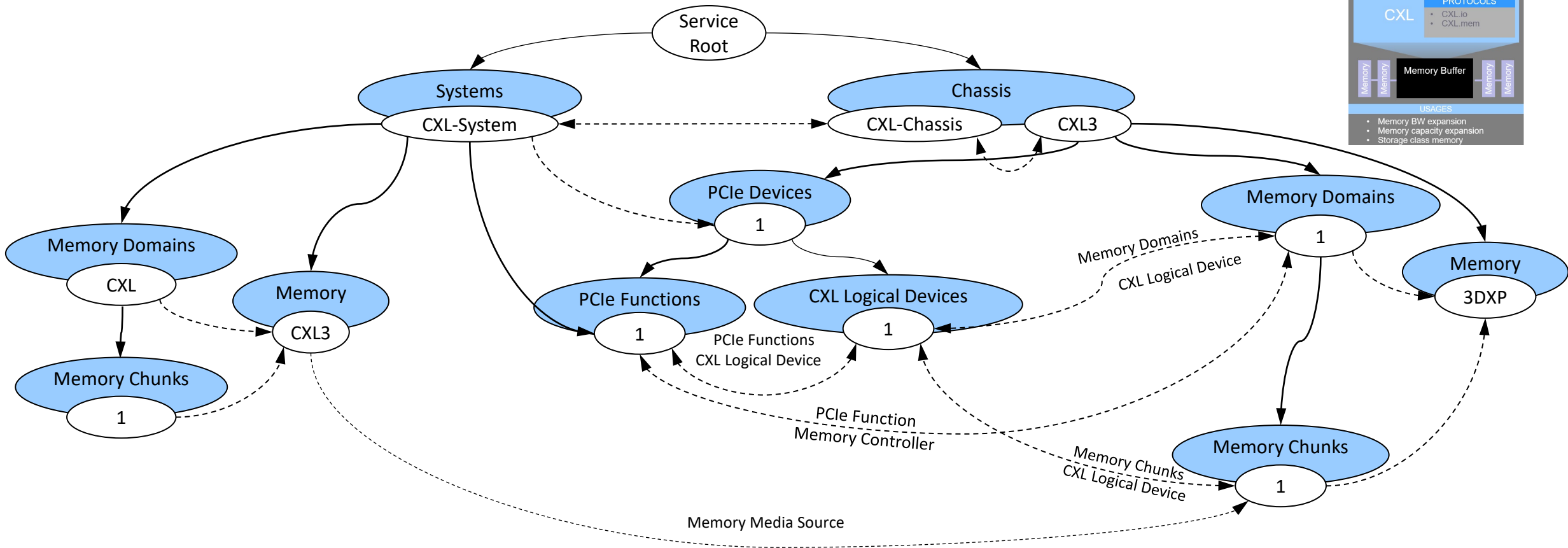
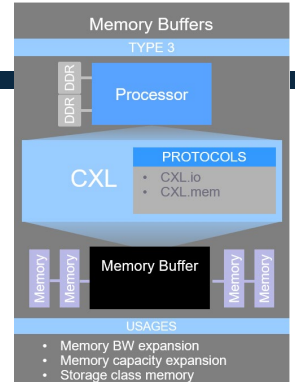
Fabric Model and CXL

Redfish Common Fabric Model

- Goal is to unify the representation of Fabrics, regardless of fabric type
 - SAS, PCIe, CXL, Ethernet Based, FC all can be shown in this structure
 - Enable client to walk from the controller to the port to the switch port, on through the switches and ports to the target's port and controller.
 - DMTF, OFA & SNIA joint collaboration on the OFMF (Open Fabric Management Framework) in the OFA.
- Simple Representation
 - Collection of Fabrics off of the Service Root
 - Switch
 - Switches have Ports that represent the connection
 - Endpoint
 - Represent the “logical” endpoint, not where the cable ends.
 - Parts of the protocol stack/standard that determine source or destination
 - Zone
 - Represents routing & default behavior.
 - AddressPool (not needed for CXL so not shown)
 - Show address allocation, eBGP, DNS and common endpoint settings
 - Connection
 - Which endpoints are allowed to communicate

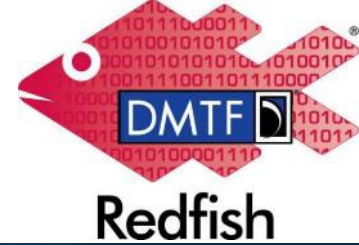


Local CXL Type 3 Device Model

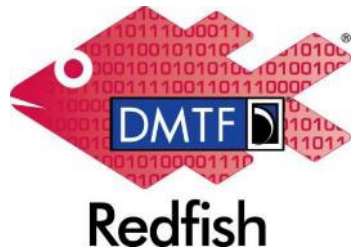


- Collection resource
- Singleton resource
- Subordinate relationship
- Associate relationship

Educational Resources on Fabrics



- **Model Specification (generated from schema)**
 - Redfish Data Model Specification DSP0268
- **Mapping Guides**
 - CXL “Rosetta Stone” is DSP0288: CXL to Redfish Mapping Specification
- **Redfish School**
 - Support for Compute Express Link (CXL)
 - Fabrics Configuration and Routing
 - Fabrics Example
- **White Paper**
 - Fabrics White Paper DSP2066



Changes to LogService

LogService, LogEntry and Event enhancements

- Add `AutoClearResolvedEntries` in 2024.1
- Added CXL log entry type
 - Raw CXL log format
- Diagnostic Data (dumps)
 - Add *DiagnosticData* property to allow inclusion of a small-to-moderate amount of binary data within the *LogEntry* or *Event* resource
 - Value is a Base64-encoded string of data
 - Type of data follows value of existing *DiagnosticDataType*
 - Provide guidance for maximum size of this data
 - Initiated by an Action on the *LogService*
 - If *DiagnosticData* received (from an external data provider) is too large given payload guidance, service will provide a URI for retrieval
 - Use the existing *AdditionalDataURI* and *AdditionalDataSizeBytes*
 - Designed to work with PLDM File Transfer
- Add *CPER* object to hold *NotificationType* and *SectionType*
 - Can add other decoded information from a CPER record or section for UEFI



Redfish

Metrics

& Proposed Telemetry Update

Metrics Abound for Storage!

- **DriveMetrics**
 - Has Correctable & Uncorrectable IO error counts, queue depth, powerOnHourse, NVMeSmart and now has ReadIOKiBytes & WriteIOKiBytes
- **StorageControllerMetrics**
 - There are many metrics, including Correctable & Uncorrectable errors like ECC & parity & state change count. Most are around NVMe Smart
- **Port Metrics**
 - General metrics for RX & TX as well as specific ones for CXL, FC, Ethernet, Gen-Z, SAS & Transceivers.
- **Memory Metrics**
 - General metrics for bandwidth, errors, dirty, health, speed but also CXL specific.
- **Telemetry Service revamp**
 - Session at OCP, participate to help shape this important usability improvement



Environmental Metrics

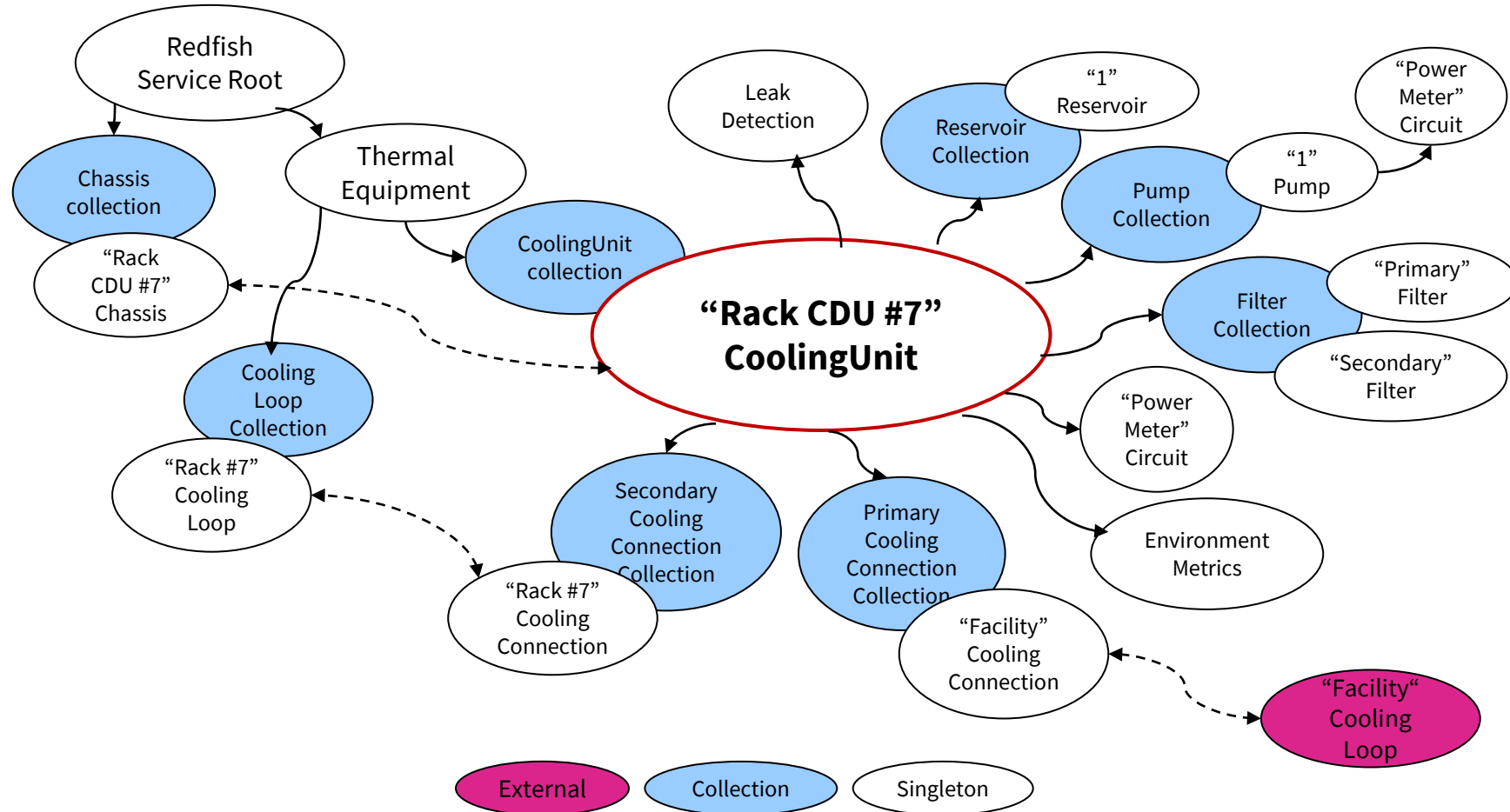
Power & Thermal became Sensors & Metrics

- Power and Thermal resources are completely replaced (deprecated)
- Created a concept call “Excerpt”
 - A subset of data that is copied from one resource and presented in another resource.
 - Added Sensor off of Chassis with dozens of properties, most of which are only needed for drill down
 - Excerpts of Sensors are represented in other resources
- ***Fans[]* and *PowerSupplies[]* become Resource Collections**
 - Contain mostly static data for these devices (and their “bays”)
 - *PowerSupply* gains a Metrics resource due to large sensor count
- ***Voltages[]* and *Temperature[]* sensors become Sensor instances**
 - Temperature readings now summarized in ThermalMetrics
 - Voltage readings move to PowerSupplyMetrics or Sensor collection
- ***PowerControl[]* object split along functional lines**
 - Power limits move to the appropriate resources
- Sensor excerpts added to various “Metrics” resources where desired
 - Much easier to correlate “CPU readings” by starting at Processor instead of searching Sensor collection
- “Subsystem” resources lay groundwork for further model expansion
 - Add them where needed for Liquid cooling, external power sources, etc.

EnvironmentMetrics resource

- A summary of sensor data related to a specific component
 - Environment readings are shown as Sensor excerpts
 - Performance metrics read directly from component, not a Sensor resource
 - Separates “performance” metrics from the environmental metrics
 - Different use cases, with “other half” of data thrown away
- Includes Reading / Sensor excerpts for:
 - Power, Temperature, Fan, Voltage
- Single schema definition used for instrumented components
 - Processors, Memory, Drive, Adapters, Controllers, Heat Exchangers, etc.

Cooling Unit Model (DCIM Work In Progress)



In Summary

- Redfish, along with the other DMTF WGs and DMTF alliance partners like SNIA, is working to define interoperable software defined hybrid IT management for servers, storage, networking, power/cooling, fabrics and more
- And is solving problems from security attestation to key exchange, composition to resource managers, aggregation engines to fabric management
- As well as plumbing the mechanisms inside the box to be self contained and self describing
- And enabling a zero-trust model in the platform

Redfish Developer Hub: redfish.dmtf.org



- Resources

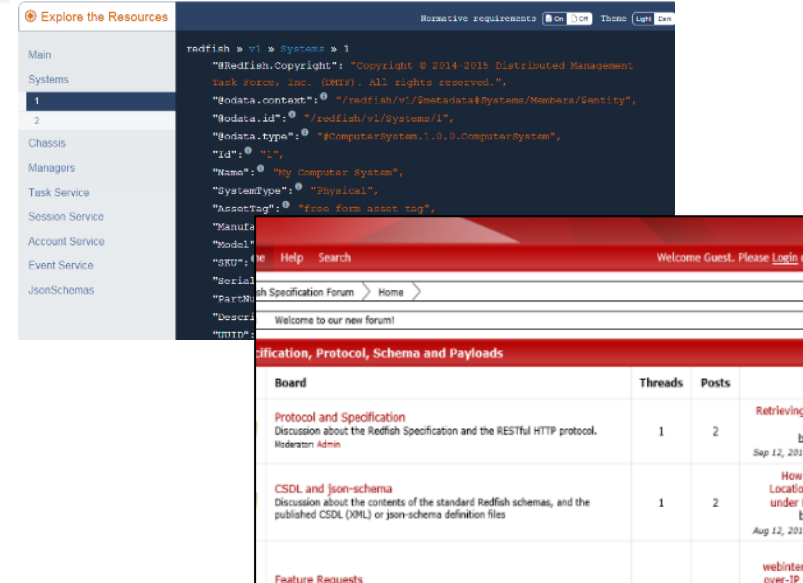
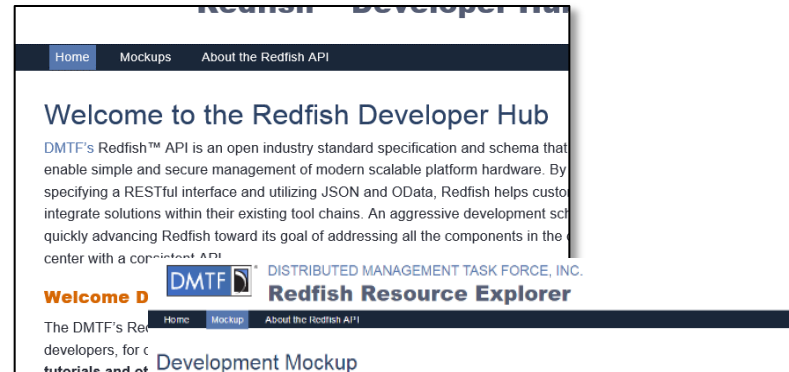
- Schema Index
- Specifications
- GitHub for Redfish Tools
- Registries
- Other Documentation

- Mockups

- Simple Rack-mounted Server
- Bladed System
- Proposed OCP Redfish Profile
- More being added

- Education/Community

- Redfish User Forum
- Whitepapers, Presentations
- YouTube shorts & Webinars





Please take a moment to rate this session.

Your feedback is important to us.

Background

Storage Resource Overview

- **Storage:** A representation of a storage sub-system
 - Contains sets of Volumes, Drives, and Storage Controllers
 - Storage Controller information is an array of objects in the Storage resource
 - Describes the protocols supported by the controller, the speed of the controller interface, and manufacturer information about the controller
- **Drive:** The physical media for the data
 - Manufacturer information about the drive (part number, serial number, etc.)
 - Capability information about the drive (size, protocol, encryption, etc.)
 - Contains control aspects (secure erase and LED setting)
- **Volume:** The logical construct used by the OS/hypervisor
 - Contains status about a volume (what drives contribute to the volume, size information, identifier information, etc.)
 - Allows a client to control the volume (initialization, encryption settings, etc.)
- **Controller:** The physical or logical storage controller
 - Was an array inside of storage (deprecated)
 - Broken out as its own object now (needed for NVMe)

