Redfish for Liquid Cooling Equipment

DMTF Redfish Forum
November 2023

Introduction

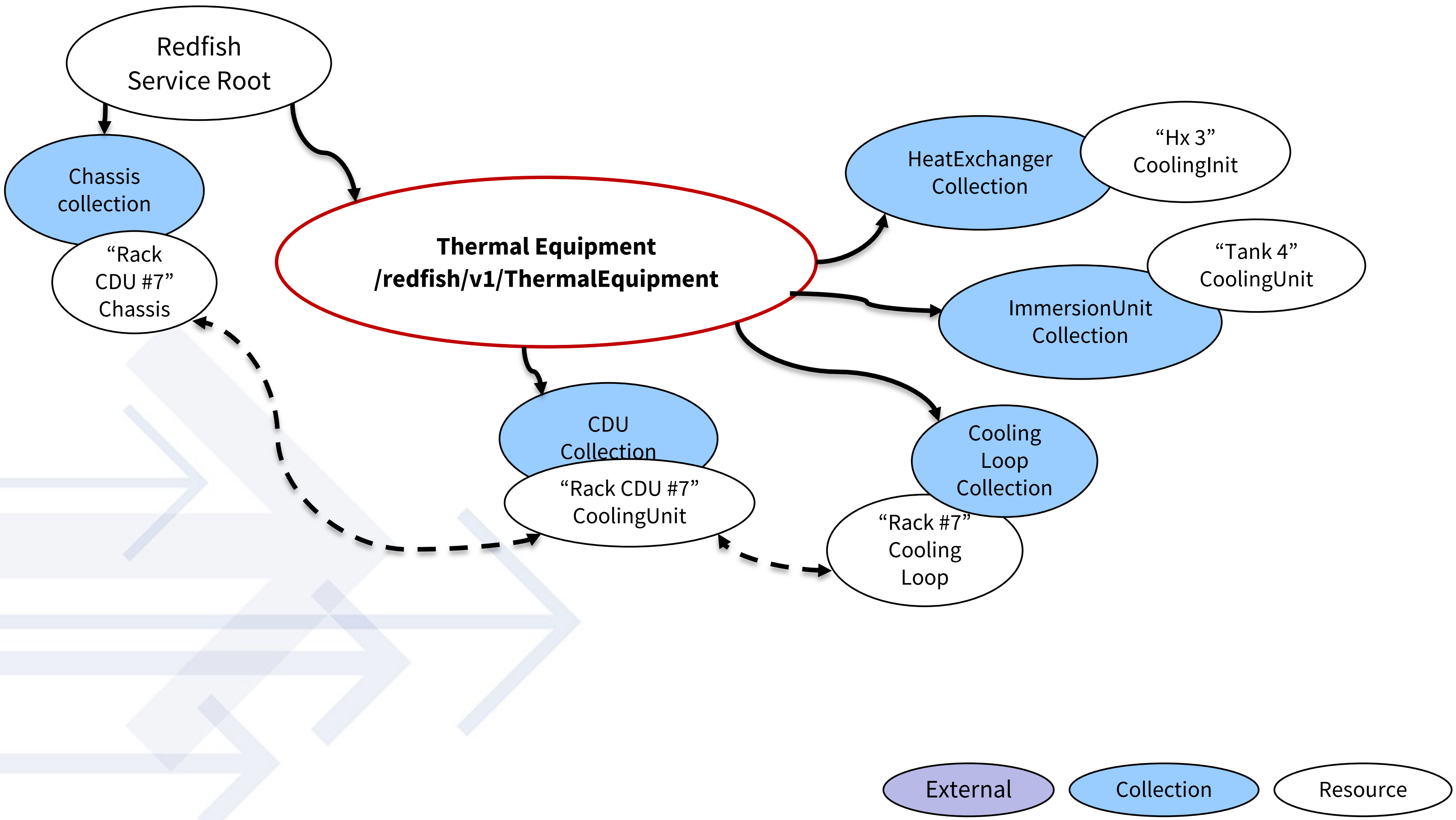
- Redfish extended into DCIM in 2019
  - Initial support for power infrastructure and power distribution units
  - Liquid cooling systems added in release 2023.1
- Benefits industry in several ways
  - Client software obtains data using a single protocol, represented in a standard data model, avoiding ongoing work to adapt vendor-specific
  - Equipment vendors leverage Redfish ecosystem and freely-available open source tools for “housekeeping” software functions
    - Includes: event configuration, firmware update, security & certificates mgmt, etc.
  - With a consistent model, DCIM equipment data can be correlated with the associated IT gear, enabling more advanced, integrated management functionality
    - Numerous methods to increase energy efficiency and equipment utilization if DCIM and IT controls can be tightly coupled
    - But we have to be able to measure and monitor first…

THERMAL EQUIPMENT MODEL

ThermalEquipment resource

- Single resource under **ServiceRoot**
  - Follows design pattern used for **PowerEquipment**
  - Contains links to all cooling systems and related equipment
  - Used primarily for discovery of managed equipment
  - Can expand to include other equipment categories, such as heat re-use
- Links to Resource Collections of:
  - Coolant Distribution Units (CDU’s)
  - Immersion Cooling Units
  - Heat Exchangers
    - Also leverages the **ThermalSubsystem** under **Chassis** for **Fan** resources
  - Cooling Loops
    - Both facility-level (FWS) and rack/secondary (TCS) loops

Thermal Equipment Model

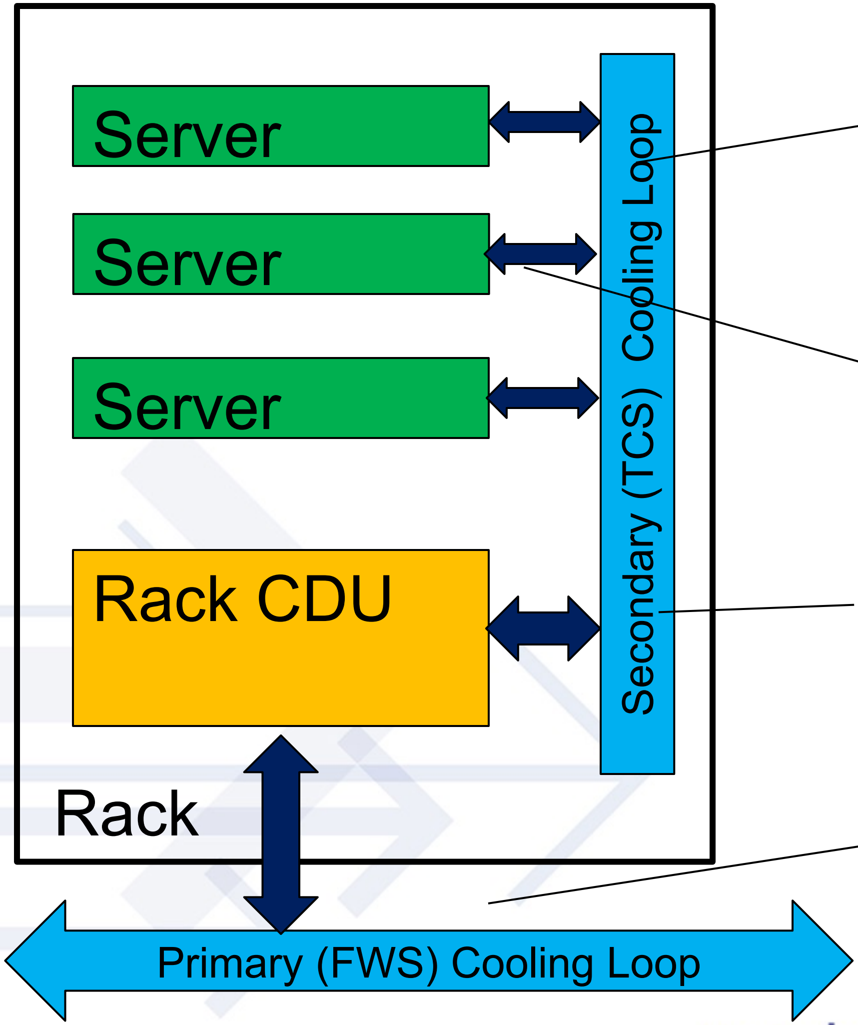


COOLING LOOP MODEL

Cooling Loop and Coolant Connector Models

- **CoolingLoop** model
  - Describes the physical characteristics and capacities of a cooling loop
  - Loop can be self-contained (within a rack or group of racks)
    - Or can be facility-wide (primary loops from external chillers, etc.)
  - Shows connectivity to equipment
    - Provides means for both “names” (strings) and links to resources
- **CoolantConnector** models connections to a **CoolingLoop**
  - Describes the “supply” and / or “return” side of the managed equipment
    - An instance is either a connection pair, or an individual supply or return
  - Metrics are gathered at these connection points
    - Allows independent metrics for each piece of equipment connected to the loop
    - Can also support “in line” measurement points using the consistent data model
  - Provide information about the connected loop if available
    - User-entered “loop name” provides a connection path through the infrastructure

Cooling Loop – Rack-level self-contained example



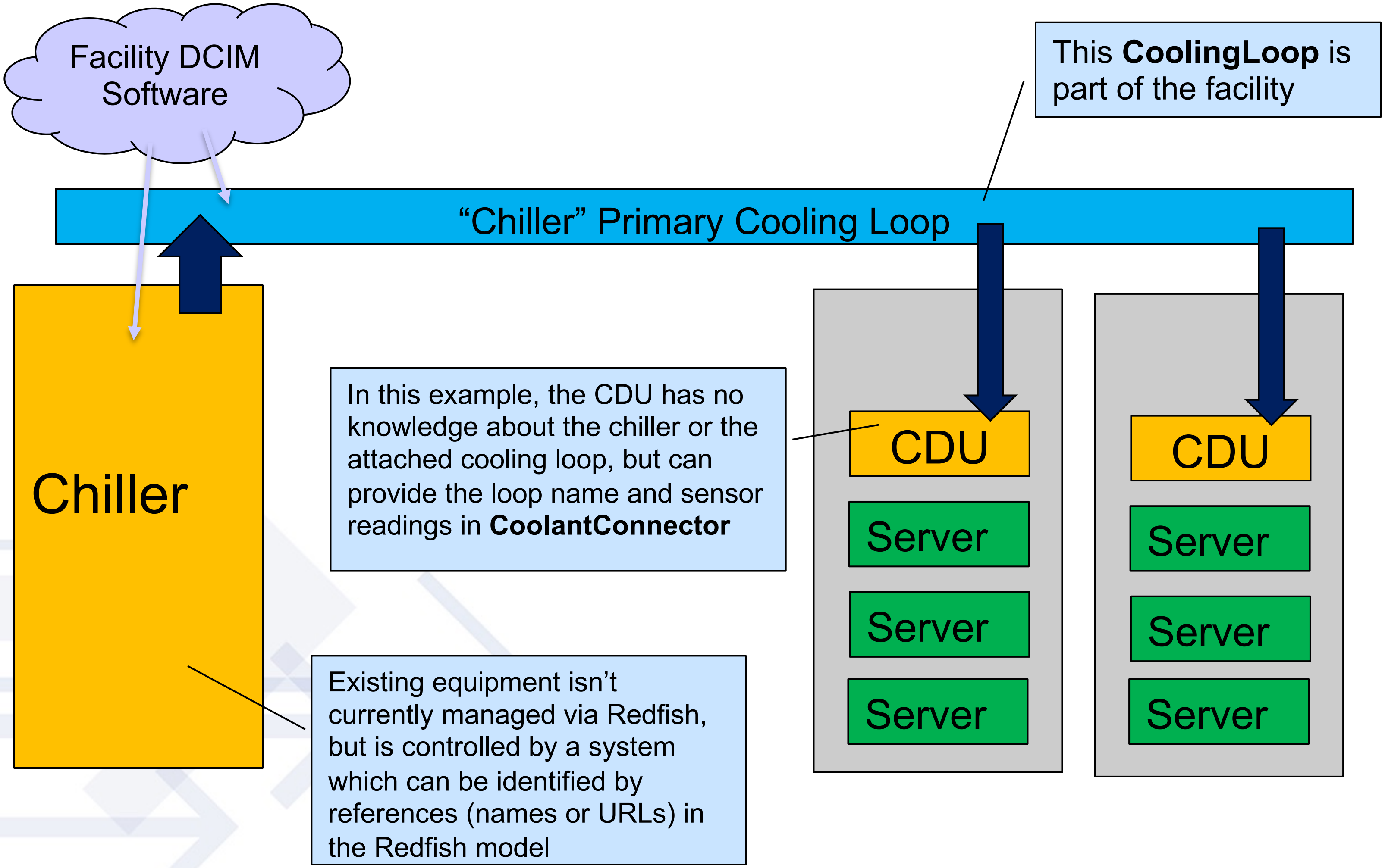
This **CoolingLoop** is within the rack, and can be modeled and populated by the RackCDU's Redfish service

System's **CoolantConnector** links to the external **CoolingLoop** (in the Rack CDU)

Rack CDU *secondary* **CoolantConnector** connects to a **CoolingLoop** resource

Rack CDU *primary* **CoolantConnector** links to a **CoolingLoop** resource in the facility, or provides just the *LoopName*

Cooling Loop – facility level example



CoolingLoop schema

- **CoolingLoopCollection** placed under **ThermalEquipment**
- Reports product, location, and capacity for the loop
- Describes *Coolant* properties
  - *CoolantType* – Water, Dielectric, Hydrocarbon, Fluorocarbon
  - *AdditiveName* and *AdditivePercent* – based on *CoolantType*
  - *ServicedDate* and *ServiceHours* – Track maintenance requirements
  - *SpecificHeatkJoulesPerKgK* and *DensitykgPerCubicMeter* – info for clients
- Cooling loop-level metrics
  - *CoolantQuality* – Normal or Abnormal
  - *CoolantLevelStatus* (OK, Warning, Critical) and *CoolantLevelPercent*
- Methods to represent connections to related equipment
  - *ConsumingEquipmentNames[]* – User-defined strings for unmanaged gear
  - *SupplyEquipmentNames[] – User-defined strings for upstream gear*
  - *CoolingManagerUri* – User-defined link to a management console

CoolingLoop resource example

```
{
    "@odata.type": "#CoolingLoop.v1_0_0.CoolingLoop",
    "Id": "BuildingChiller",
    "Name": "Feed from building chiller",
    "Status": {
        "Health": "OK"
    },
    "UserLabel": "Building Chiller",
    "RatedFlowLitersPerMinute": 900,
    "RatedPressurekPa": 1600,
    "Coolant": {
        "AdditiveName": "Generic cooling water biocide",
        "AdditivePercent": 0.25,
        "CoolantType": "Water"
    },
    "CoolantLevelStatus": "OK",
    "CoolantQuality": "Normal",
    "CoolantLevelPercent": {
        "Reading": 95
    },
    "SupplyEquipmentNames": ["Chiller"],
    "ConsumingEquipmentNames": ["Rack #1 CDU", "Rack #2 CDU", "Rack #3 CDU", "Rack #4 CDU"],
    "PrimaryCoolantConnectors: {
        "@odata.id": "/redfish/v1/ThermalEquipment/CoolingLoops/BuildingChiller/PrimaryCoolantConnectors"
    },
    "SecondaryCoolantConnectors: {
        "@odata.id": "/redfish/v1/ThermalEquipment/CoolingLoops/BuildingChiller/SecondaryCoolantConnectors"
    },
    << TRUNCATED >>
```

Details about coolant used in the loop

Sensor excerpts for coolant level and total heat removed

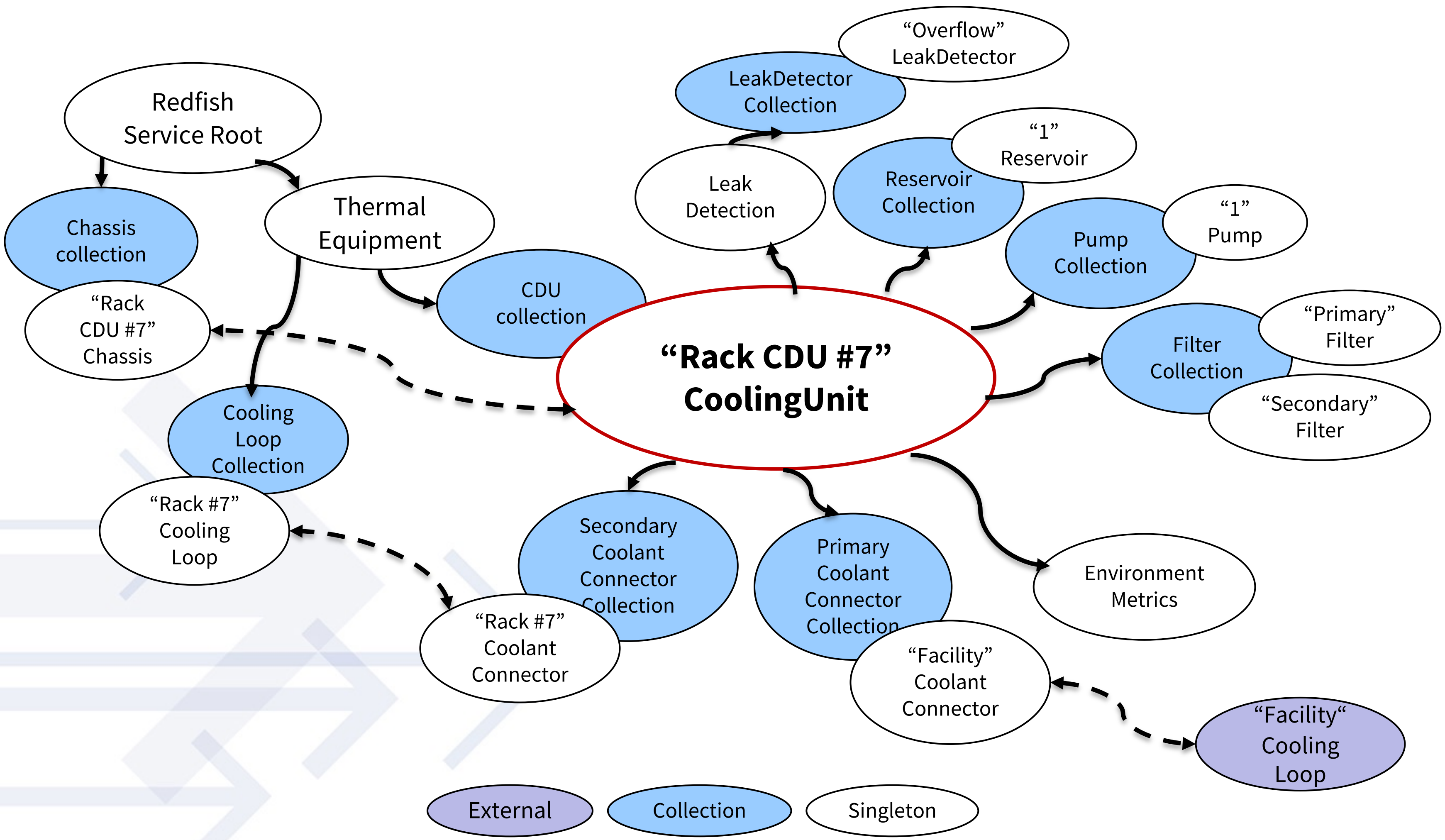
EquipmentNames allow users to manually add non-Redfish devices to help complete the model

COOLING UNIT MODEL

CoolingUnit schema and resources

- Unified schema covers many types of cooling gear
  - Equipment that cannot be modeled by a **Chassis** and **ThermalSubsystem**
    - Simple heat exchangers or manifolds may be covered as a Chassis alone
    - **CoolingUnit** equipment will have a containing Chassis resource
  - Share common modeling and property definitions
  - *EquipmentType* property provides specific identification
- Resource contents
  - General product identification – model, manufacturer, serial number, etc.
  - Versioning – Hardware revision, firmware version, date of manufacture
- *Links* to subordinate and related resources
  - **Sensors**, **LeakDetection**, **EnvironmentMetrics**
  - Primary (input) and Secondary (output) **CoolantConnectors**
  - Subsystems: **Pumps, Filters, Reservoirs**
  - **Chassis** that contains the equipment

Cooling Unit Model



CoolingUnit resource example


```
{
    "@odata.type": "#CoolingUnit.v1_0_0.CoolingUnit",
    "Id": "1",
    "EquipmentType": "CDU",
    "Name": "Rack #4 Cooling Distribution Unit",
    "FirmwareVersion": "3.2.0",
    "Version": "1.03b",
    "ProductionDate": "2020-12-24T08:00:00Z",
    "Manufacturer": "Contoso",
    "Model": "BRRR4000",
    "SerialNumber": "29347ZT536",
    "PartNumber": "ICE-9",
    “Coolant”: {
        “CoolantType”: “Hydrocarbon”
    },
    "AssetTag": "PDX5-92381",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    “LeakDetection”: { “@odata.id”: < Link to LeakDetection  > },
    "PrimaryCoolantConnectors": { “@odata.id”: < Link to CoolantConnectorCollection > },
    "SecondaryCoolantConnectors": { “@odata.id”: < Link to CoolantConnectorCollection > },
    "Pumps": { “@odata.id”: < Link to PumpCollection > },
    "Filters": { “@odata.id”: < Link to FilterCollection > },
    "EnvironmentMetrics": { “@odata.id”: < Link to EnvironmentMetrics > },
     < TRUNCATED >
```

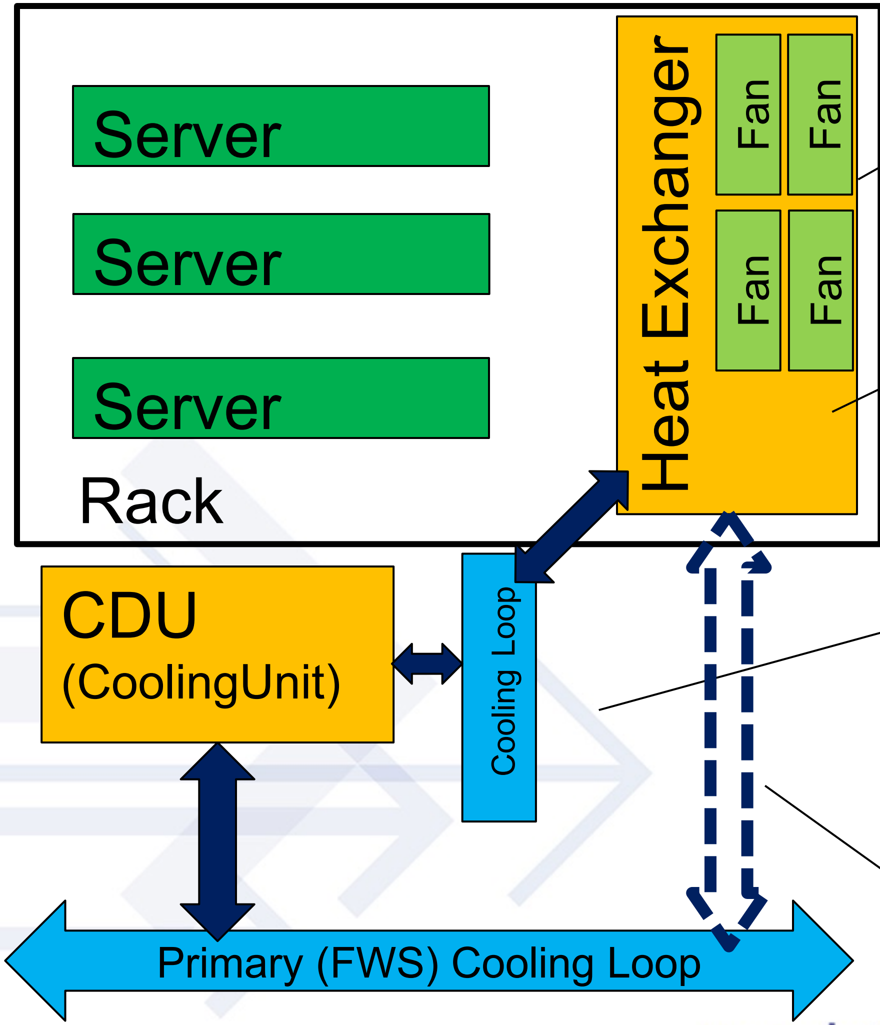
CoolantConnector schema

- Describes a coolant-carrying connector and its equipment connections
  - Modeled either as a connector pair, or an individual “supply” or “return”
  - Provides numerous sensor readings
    - Flow, Temperature, Pressure on both supply and return
  - If known, provide link to **CoolingLoop**
    - Or the loop name and Manager URI if known and populated by end user
- Main monitoring resource for the cooling unit’s functionality
  - Primary coolant connectors – input from facility chillers or other sources
  - Secondary coolant connectors– output from the cooling unit to feed “consuming” equipment

CoolantConnector resource example

```
{
    "@odata.type": "#CoolantConnector.v1_0_0.CoolantConnector",
    "Id": "Chiller",
    "Name": “Primary Input from Chiller",
    "Status": {
        "Health": "OK"
    },
    "CoolantConnectorType": “Pair",
    “CoolingLoopName”: “Building Chiller”,
    “Coolant”: {
        “CoolantType”: “Water”,
        “AdditiveName”: “Generic cooling water biocide”,
        “AdditivePercent”: 0.25
    }
    "RatedFlowLitersPerMinute": 120,
    “RatedPressurekPa”: 1600,
    "SupplyTemperatureCelsius": {
        "DataSourceUri": "/redfish/v1/CoolingEquipment/RackCDUs/1/Sensors/LoopASupplyTemp",
        "Reading": 14.8
    },
   “SupplyPressurekPa": {
        "Reading": 319.6
    },
    “DeltaTemperatureCelsius”: {
        “Reading”: 19.3
    },
    < TRUNCATED >
```

Example: Rear Door Heat Exchanger



The heat exchanger model includes a **Chassis** with a **ThermalSubsystem** that describes the **Fans**

ThermalMetrics has new sensor excerpts to describe airflow and delta pressure

CDU *primary* **CoolantConnector** attaches to the facility cooling loop and its *secondary* **CoolantConnector** connects to a technology loop, which in turn connects to the heat exchanger via **CoolantConnector**

OR the heat exchanger directly connects to the facility loop via a **CoolantConnector**

Subsystems for CoolingUnit and ThermalEquipment

- Equipment that may appear as a subsystem of a **CoolingUnit**
  - Model this equipment as a subordinate resources to an individual **CoolingUnit**
- These schemas contain basic inventory and identification data
  - More specific properties are easily added as feedback is received
- **Pump** resource collection
  - Pump speed, product information
- **Reservoir** resource collection
  - Fill level, capacity, internal pressure, product information
- **Filter** resource collection
  - Service time / install time, product information

LEAK DETECTION

LeakDetection schema

- Resource to describe leak detection equipment and report leaks
  - Allows discovery of detection equipment to validate customer requirements
- *DetectorGroups* supports multiple "zones" of detection
  - Each group represents a detection zone
  - Made up of one or more *LeakDetector instances*
  - Can also include a humidity sensor
  - "Policy" for what constitutes a reported leak is left to implementation
    - Assumes this is manufacturer or configuration based, not user-defined
- *Status* object provides means to report leaks
  - Will define messages for reporting leaks as *Conditions*

LeakDetection resource example

```
{
    "@odata.type": "#LeakDetection.v1_0_0.LeakDetection",
    "Name": "Leak Detection Systems",
    "Status": {
        "State": "Enabled",
        "Health": “OK",
        "Conditions": []
    },
    "LeakDetectorGroups": [{
        "GroupName": "Detectors under and around the CDU",
        "HumidityPercent": {
            "Reading": 45
        },
        "Detectors": [{
                "DataSourceURI": "/redfish/v1/ThermalEquipment/CDUs/1/LeakDetection/LeakDetectors/1",
                “PhysicalContext": “Chassis",
                "DetectorState": "OK"
            },
            {
                "DataSourceURI": "/redfish/v1/ThermalEquipment/CDUs/1/LeakDetection/LeakDetectors/2",
                “PhysicalContext”: “Chassis”,
                "DetectorState": “OK"
            }
        ]
    }],
```

In this example with one LeakDetectorGroup, there are three sensors, with the implementation deciding the policy under which a leak is reported

Humidity reading, with an internal threshold to indicate a leak

Two types of Detectors, which will indicate a leak with a DetectorState of “Alert”

Call to Action

- Implement Redfish support on your cooling equipment
  - Leverage the existing Redfish software ecosystem
  - Enable higher-level client software functions
    - Link infrastructure power & cooling to IT gear performance data
    - Share status information and provide event notification
- Provide feedback on the data model
  - Redfish 2023.2 release incorporated feedback from early adopters
    - CoolingLoop, CoolantConnector redundancy support added
    - Redfish releases typically occur 3-4 times annually
  - Simple questions or suggestions are welcome on the public forum
  - Feedback can be submitted to the DMTF directly

Getting involved in Redfish

- Redfish Standards page
  - Schemas, Specs, Mockups, White Papers & more
  - https://www.dmtf.org/standards/redfish
- Redfish Developer Portal
  - Redfish Interactive Resource Explorer
  - Educational material, documentation & other links
  - https://redfish.dmtf.org
- Redfish User Forum
  - User forum for questions, suggestions and discussion
  - https://www.redfishforum.com
- DMTF Feedback Portal
  - Provide feedback or submit proposals for Redfish standards
  - https://www.dmtf.org/standards/feedback
- DMTF GitHub organization
  - Open source tools and libraries for DMTF standards
  - https://www.github.com/DMTF

Redfish

Q&A & Discussion