Proposed Redfish Telemetry Model

Ver. 1.0
April 2017

Disclaimer

- The information in this presentation represents a snapshot of work in progress within the DMTF.
- This information is subject to change. The Standard Specifications remain the normative reference for all information.
- For additional information, see the Distributed Management Task Force (DMTF) Web site.

www.dmtf.org

Providing Feedback and Comments

Feedback to the proposal can be provided

- During a meeting or during a ballot (via ballot comment)
  - By members of the DMTF Scalable Platform Management Forum
  - http://www.dmtf.org/standards/redfish
- Via the DMTF feedback portal
  - http://www.dmtf.org/standards/feedback

Goal

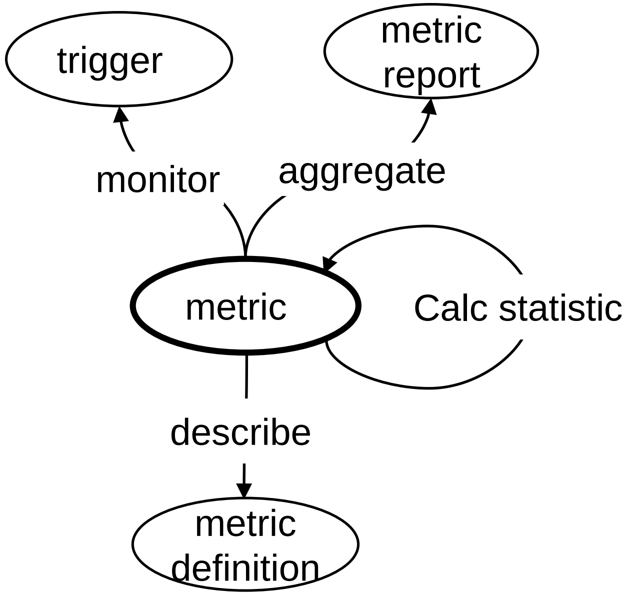
- The Redfish represent metrics as resource properties (readings, statistics)
  - Resource properties used for metric characteristics
  - Resource properties used from triggers thresholds
- Users have use cases that require
  - Additional metric characteristics
  - Ability to obtain a group of metrics
  - Ability to configure triggers for a metric
- Propose a general model for metrics characteristics, metric reports and triggers that work with existing metric properties

/redfish/v1/Chassis/1/Power

```
"Voltages": [
   {
      "@odata.id": "",
      "MemberId": "0",
      "Name": "VRM1 Voltage",
      "SensorNumber": 11,
      "Status": {
         "State": "Enabled",
         "Health": "OK"
      },

      "ReadingVolts": 12,
      "UpperThresholdNonCritical": 12.5,
      "UpperThresholdCritical": 13,
      "UpperThresholdFatal": 15,
      "LowerThresholdNonCritical": 11.5,
      "LowerThresholdCritical": 11,
      "LowerThresholdFatal": 10,
      "MinReadingRange": 0,
      "MaxReadingRange": 20,
      "PhysicalContext": "VoltageRegulator",
   }
]
```

Use cases

- Get value of a metric
  - Get resource, query metric property
  - Statistical metrics (e.g. min, max, average)
- Get metric metadata
  - Describes characteristics of the metric
  - E.g. physical sensor or digital meter, numeric vs discrete, range, etc.
- Aggregate metrics
  - Specify a set of metrics reported together
  - Report can be logged and/or transmitted
  - On obtained, able to correlate metrics, by time
- Configure Triggers
  - Delegate the monitoring of a metric against a set of triggers



Modeling elements

Get a metric value

- Get resource, query metric property
- E.g. Power#/Voltages/0/ReadingVolts

Get metric metadata

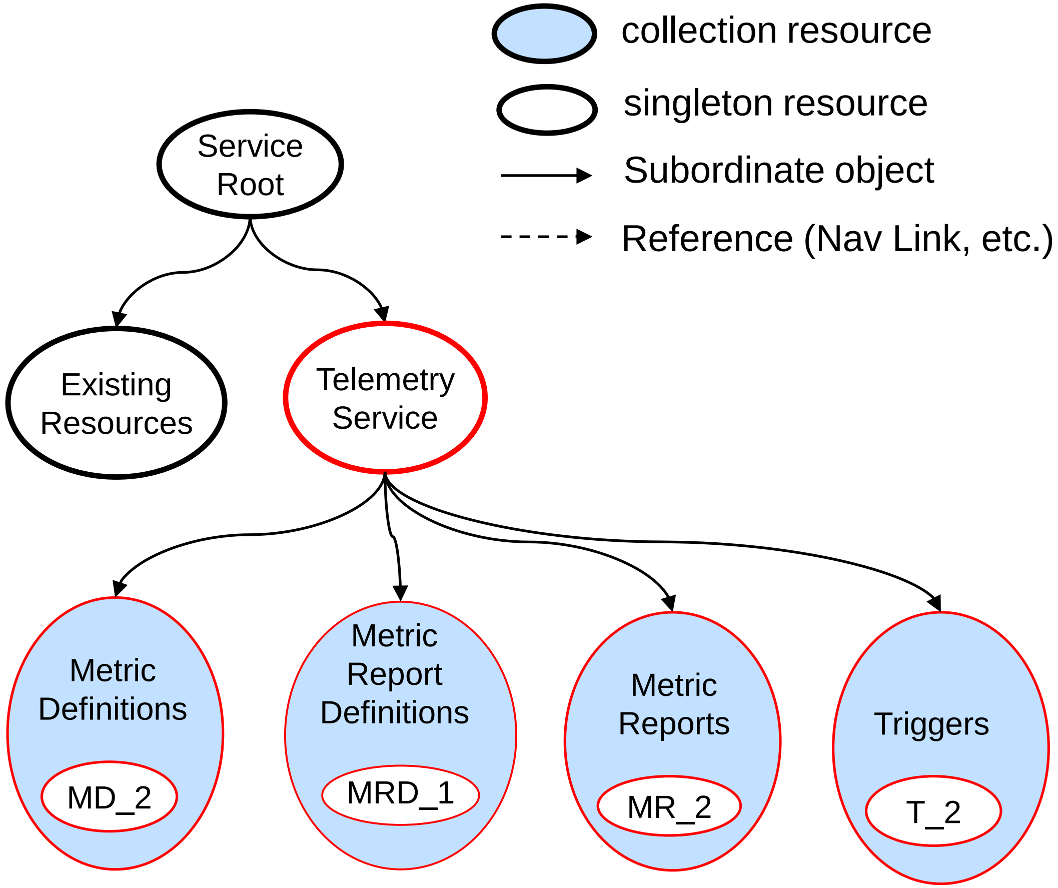
- Get associated MetricDefinition
- Contains characteristics of metrics
- Defined the calculation of statistics

Aggregated reporting metrics

- Create MetricReportDefinition resource
- List of metrics to include a report and periodicity of report
- Reports can be placed in MetricReport resource

Configure Triggers (2 methods)

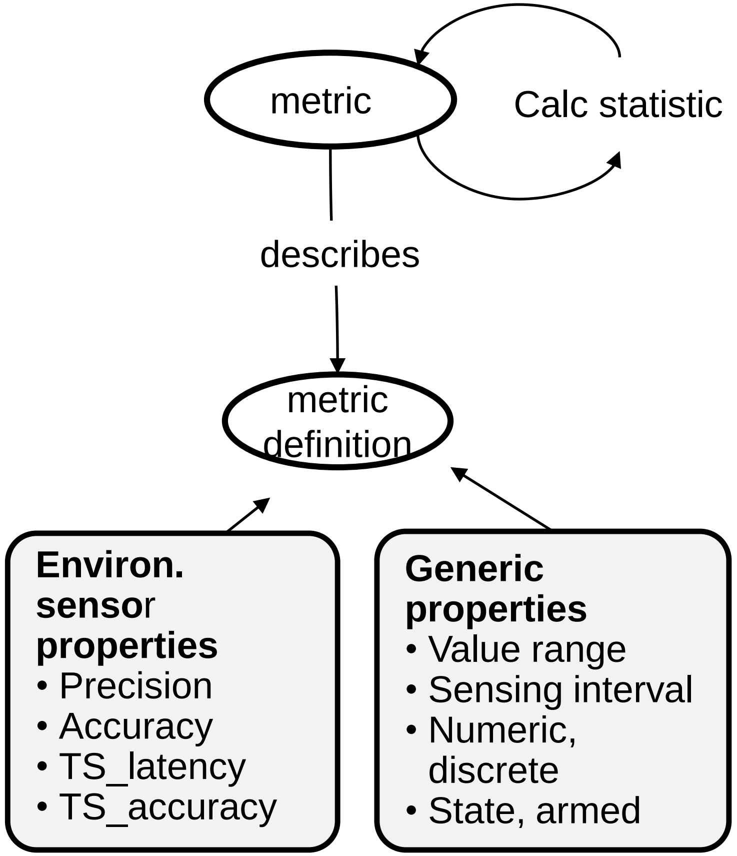
- Specify in Triggers resource, or
- Specify in MetricReportDefinition



Metric Definitions

Metric Definitions

- Metrics can be obtained in several ways
  - A sensor reading (whether a physical sensor or synthesized sensor)
  - A digital meter (reading from a register)
  - Metrics can be numeric or discrete
- The MetricDefinition resource
  - Associated to the metric
  - Describes the characteristics of the metric
  - If the metric is a statistic, describes the statistical calculation
- Characteristics and metadata
  - Generic metadata are meaningful to all sources
  - Sensor metadata characterizes the gap between the reading and physical reality



Environ. sensor properties
- Precision
- Accuracy
- TS_latency
- TS_accuracy

Generic properties
- Value range
- Sensing interval
- Numeric, discrete
- State, armed

Metric Definition properties

Property	Values	Description
MetricType	Counter	The metric is a counter metric (rollover, etc.)
	Gauge	The metric has integer or float values that can increase and decrease arbitrarily
	Numeric	The metric has a numeric value, a real number present in float format.
	Discrete	The metric has discrete values, which have discrete values or states, not continuous. (see DiscreteValues property)
ImplementationType	PhysicalSensor	The metric is implemented as a physical sensor.
	DigitalMeter	The metric is implemented as digital meter.
	Synthesized	The metric is implemented by applying a calculation on the readings from one or more physical sensors
	Calculated	The metric is implemented by applying a calculation on one more metric properties.
CalculationAlgorithm		Enum: AverageOverInterval, MinumumOverInterval, MaximumOverInterval
CalculationTimeInterval		The time interval over which the calculation is performed
SensorType		Enumerated in LogEntry.xml
PhysicalContext		Enumerated in PhysicalContext.xml
Units		As defined by Unified Code for Units of Measure (UCUM)
DiscreteValues[]		Array of discrete values that the metric value may take

Metric Definition properties

Property	Values	Description
Calculable		Specifies whether algebraic operations can be performed on the metric value
IsLinear		Specifies whether the metric value can be compared
PowerAPI, Table 4.2		
Precision		Number of significant digits in values
Accuracy		Estimated percent error +/- of measured vs. actual values.
TimeStampLatency		Estimate of the time required to get or set an attribute. This is useful to estimate completion time for an operation a priori. A value of zero should be returned when the get/set is instantaneous.
TimeStampAccuracy		Estimated accuracy of returned timestamps, represented as +/- the PWR_Time value returned.
TimeWindow (CalculationTimeInterval)		The time window used to calculate the value returned or relevant to an attribute. For example, the “instantaneous” PWR_- ATTR_POWER values reported may actually be averaged over a short time window. Power caps are also enforced with respect to a target time window.
UpdateRate		Rate values become visible to user, in updates per second. Getting or setting a value at a rate higher than this is not useful.
SampleRate		Rate of underlying sampling, in samples per second. This is only relevant for values derived over time (e.g., PWR_ATTR_- ENERGY)
MeasureMethod		Denotes the measurement method: an actual measurement (returned value = 0) or a model based estimate (return value = 1). Other values > 1 may be used to denote multiple vendor specific models in the situation where multiple models may exist.

MetricDefinition resource example (sensor)

```
{
"@odata.context": "/redfish/v1/$metadata#MetricDefinition.MetricDefinition",
    "@odata.type": "#MetricDefinition.v1_0_0.MetricDefinition",
    "@odata.id": "/redfish/v1/TelemetryService/MetricDefinitions/PowerConsumedWatts",
    "Id": "PowerConsumedWatts",
    "Name": "Power Consumed Watts Metric Definition",

    "MetricType": "Numeric",
     "SensorType": "PowerConsumption",
     "Implementation": "PhysicalSensor",
     "PhysicalContext": "PowerSupply",

     "SensingIntervalms": "1000",
     "ReadingUnits": "W",
     "Precision": "4",
     "Accuracy": "0",
     "Calibration": "2",
     "TimeStampAccuracyms": "1",
     "TimeStampLatencyms": "10",
     "MinReadingRange": "0",
     "MaxReadingRange": "50",

     "AppliesTo": [
         { "@odata.id": "/redfish/v1/Chassis/1/Power#/PowerControl/0/PowerConsumedWatts " },
         { "@odata.id": "/redfish/v1/Chassis/1/Power#/PowerControl/1/PowerConsumedWatts " }
    ]
}
```

PowerAPI Spec
- Precision (signf digits)
- Accuracy
- TS_latency
- TS_accuracy

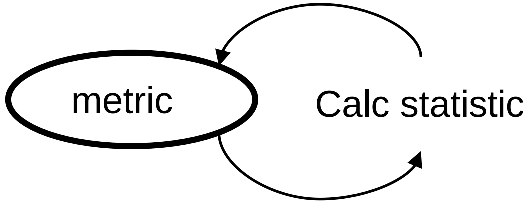
"Metrics" points to each sensor reading specified by this MetricDefinition

MetricDefinition resource example (digital meter)

```
{
    "@odata.context": "/redfish/v1/$metadata#MetricDefinition.MetricDefinition",
    "@odata.type": "#MetricDefinition.v1_0_0.MetricDefinition",
    "@odata.id": "/redfish/v1/TelemetryService/MetricDefinitions/Memory_CurrentPeriod_BlocksRead",
    "Id": "Memory_CurrentPeriod_BlocksRead",
    "Name": "Current Memory Blocks Read Metric Definition",

    "MetricType": "Numeric",
    "SensorType": "Memory",
    "Implementation": "DigitalMeter",

    "SensingIntervalms": "1000",
    "ReadingUnits": "Blocks",
    "MinReadingRange": "0",
    "MaxReadingRange": "1000000",

    "AppliesTo": [
        { "@odata.id": "/redfish/v1/Systems/1/Memory/1/MemoryMetric#/CurrentPeriod/BlocksRead" },
        { "@odata.id": "/redfish/v1/Systems/1/Memory/2/MemoryMetric#/CurrentPeriod/BlocksRead" },
        { "@odata.id": "/redfish/v1/Systems/1/Memory/3/MemoryMetric#/CurrentPeriod/BlocksRead" },
        { "@odata.id": "/redfish/v1/Systems/1/Memory/4/MemoryMetric#/CurrentPeriod/BlocksRead" }
    ]
}
```

Statistics

- Some resources already have properties for statistics
  - The MetricDefinition for a statistics metric property should describe the calculation, and reference both the source metric properties and resultant metric property
- But statistics may be desired for metrics where a statistics metric property doesn't exist
  - Specific the statistics calculation in MetricReportDefinition



metric

Calc statistic

/redfish/v1/Chassis/1/Power

```
"PowerControl": [
   {
      "@odata.id": "…",
      "MemberId": "0",
      "Name": "System Power Control",
      "PowerConsumedWatts": 8000,
      "PowerRequestedWatts": 8500,
      "PowerAvailableWatts": 8500,
      "PowerCapacityWatts": 10000,
      "PowerAllocatedWatts": 8500,
      "PowerMetrics": {
         "IntervalInMin": 30,
         "MinConsumedWatts": 7500,
         "MaxConsumedWatts": 8200,
         "AverageConsumedWatts": 8000
      },
      . . .
   }
]
```

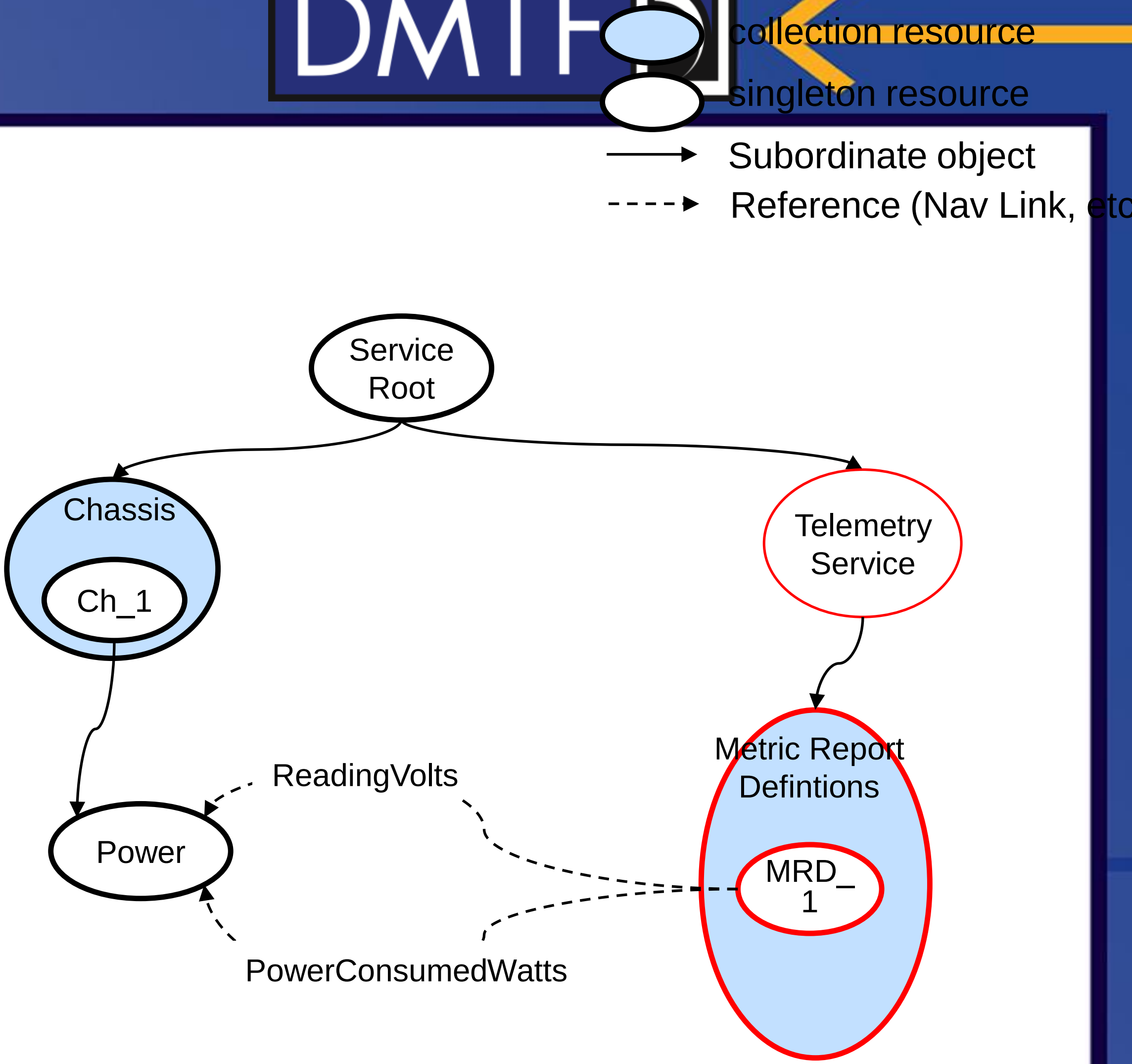
MetricDefinition resource example (statistic)

```
{
    "@odata.context": "/redfish/v1/$metadata#TelemetryService/MetricDefinitions/AverageConsumedWatts",
    "@odata.id": "/redfish/v1/TelemetryService/MetricDefinitions/AverageConsumedWatts",
    "@odata.type": "#MetricDefinition.v1_0_0.MetricDefinition",
    "Id": "AverageConsumedWatts",
    "Name": "Average Consumed Watts Metric Definition",

    "MetricType": "Numeric",
    "SensorType": "PowerConsumption",
    "Implementation": "Calculated",
    "PhysicalContext": "PowerSupply",

    "CalculationAlgorithm": "AverageOverInterval",
    "CalculationTimeInterval": "PT1S",

    "Wildcards": [
        { "ChassisID": [ "1", "2", "3" ] }
    ],

    "CalculationParameters": [
        {
            "SourceMetric": { "@odata.id": "/redfish/v1/{ChassisID}/1/Power#/PowerControl/0/PowerConsumedWatts " },
            "ResultMetric": { "@odata.id": "/redfish/v1/{ChassisID}/1/Power#/PowerControl/0/AverageConsumedWatts " }
        }
    ]
}
```

Metric Report Definition

Metric Report Definition

- Client defines report
  - Select or create a MetricReportDefinition with a list of metrics to include
  - The MetricReportDefinition contains a reference to each metric property



MetricReportDefinition resource example

```
    {
        . . .
        "@odata.id": "/redfish/v1/TelemetryService/MetricReportDefinitions/PowerMetrics",
        "Id": "PowerMetrics",
        "Name": "Power Metrics",

        "MetricReportType": "Periodic",
        "Schedule": {
            "RecurrenceInterval": "PT01S"
        },
        "ReportAction": ["Transmit", "Log"],
        "MetricReport": {"@odata.id": "/redfish/v1/TelemetryService/MetricReports/PowerMetrics"},
        "Volatile": true,
        "Status": {
            "State": "Enabled"
        },
         "Wildcards": [
            { "TrayID": [ "Tray_1", "Tray_2", "Tray_3" ],
            { "PwrCtrl": [ "0", "1" ] }
        ]
        "MetricProperties": [
           { "@odata.id": "/redfish/v1/Chassis/{TrayID}/Power#/PowerControl/{PwrCtrl}/PowerConsumedWatts" },
           { "@odata.id": "/redfish/v1/Chassis/{TrayID}/Power#/PowerControl/{PwrCtrl}/PowerMetrics/MinConsumedWatts" },
           { "@odata.id": "/redfish/v1/Chassis/{TrayID}/Power#/PowerControl/{PwrCtrl}/PowerMetrics/MaxConsumedWatts" },
           { "@odata.id": "/redfish/v1/Chassis/{TrayID}/Power#/PowerControl/{PwrCtrl}/PowerMetrics/AverageConsumedWatts" }
        ]
    }
```

MetricReportDefinition resource example (statistics)

```
{
    "@odata.id": "/redfish/v1/TelemetryService/MetricReports/PowerMetrics",
    "Id": "PowerMetrics", "Name": "PowerMetrics",
    "Schedule": {
        "Lifetime": "P05D",
        "RecurrenceInterval": "PT0.001S"
    },
    "MetricReport": {"@odata.id": "/redfish/v1/TelemetryService/MetricReports/PowerMetrics"},
    "Volatile": true,
    "Wildcards": [
        { "PWild": ["0", "1"] },
        { "TWild": ["Tray_1", "Tray_2", "Tray_3"] }
    ],
    "Metrics": [
        {
            "MemberID": "AverageConsumedWatts",
            "CollectionFunction": "Avg",
            "MetricProperties": ["/redfish/v1/Chassis/{TWild}/Power/PowerControl/{PWild}/PowerConsumedWatts"]
        }, {
            "MemberID": "MaximumConsumedWatts",
            "CollectionFunction": "Max",
            "MetricProperties": ["/redfish/v1/Chassis/{TWild}/Power/PowerControl/{PWild}/PowerConsumedWatts"]
        }, {
            "MemberID": "MinimumConsumedWatts",
            "CollectionFunction": "Min",
            "MetricProperties": ["/redfish/v1/Chassis/{TWild}/Power/PowerControl/{PWild}/PowerConsumedWatts"]
        }
    ]
}
```

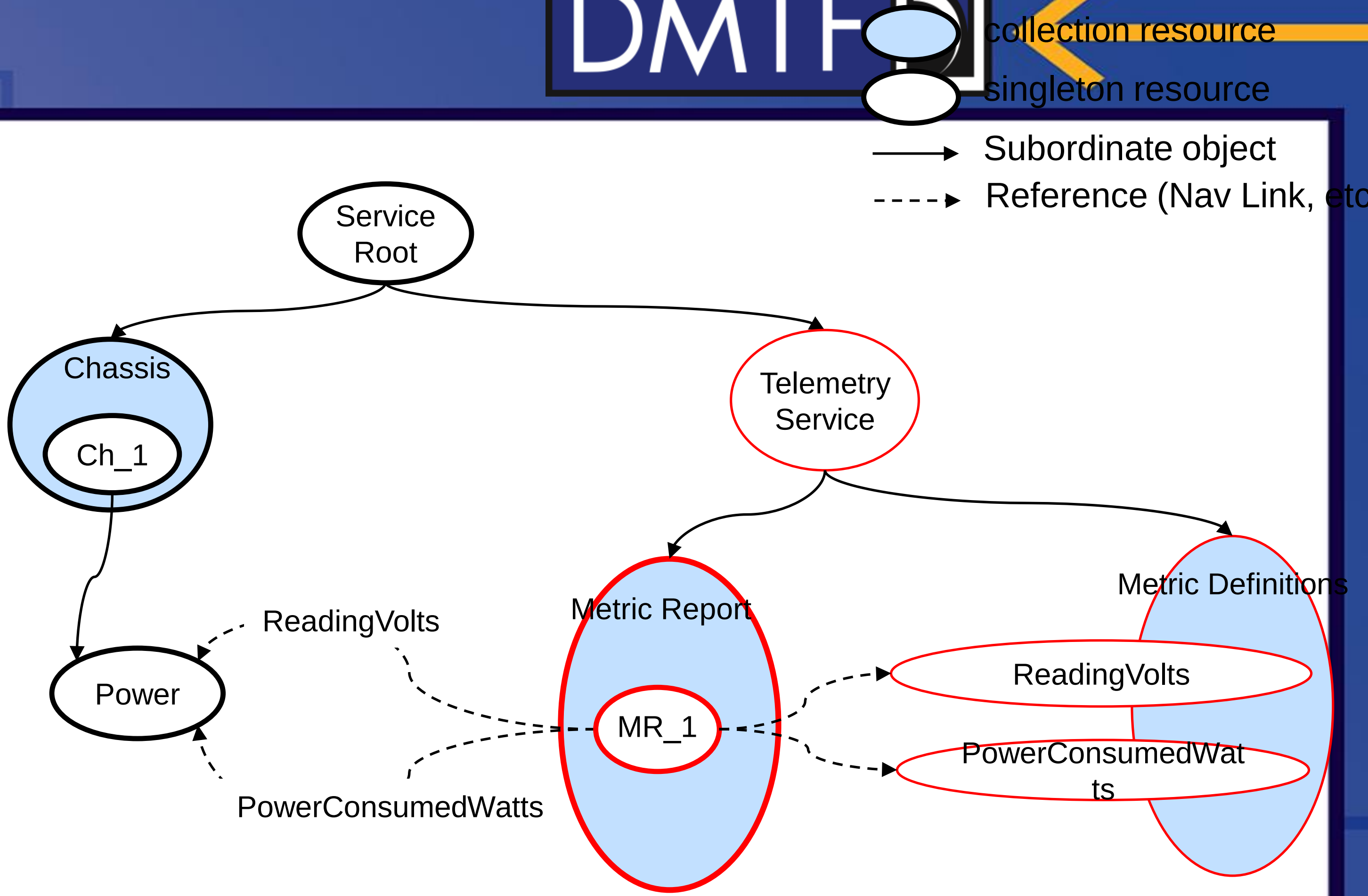
Metric Report Definition properties

Property	Values	Description
MetricReportType	Periodic	The metric report shall be updated periodically. See Schedule property.
	OnChange	The metric report shall be updated when values inside of the measured resource change
	OnRequest	The metric report shall be updated each time a client application reads it
ReportActions[]		An array property to actions to perform • Log - Place a metric report in the location specified by the MetricReport property • Transmit – Send the event as an Metric event (include the metric report)
CollectionTimeScope	Point	The metric value(s) are a point in time
	Interval	The metric value(s) are over a time interval
	StartupInterval	The metric value(s) are over a time interval, that began at the startup of the measured resource
Schedule		Complex property which contains the ReoccurenceInterval property
MetricReport		Reference to the location to place the metric report
Volatile		A binary value with indicates whether the MetricReport is overwritten, instead of appended
Wildcard[]		An array property with wildcards to substitute in the MetricProperties array property
MetricProperties[]		An array of URIs to metric properties

Metric Report

Metric Report

- Metric report (MR)
  - Created by the Redfish service based on MetricReportDefintion
- MetricReport resource
  - Includes links to the source metric
  - Includes a link to the metric definition for each metric (if metric definitions are present)



MetricReport resource example

```
{
    "@Redfish.Copyright": "Copyright 2014-2016 Distributed Management Task Force, Inc. (DMTF). All rights reserved.",
    "@odata.context": "/redfish/v1/$metadata#MetricReport.MetricReport",
    "@odata.type": "#MetricReport.v1_0_0.MetricReport",
    "@odata.id": "/redfish/v1/TelemetryService/MetricReports/PowerMetrics",
    "Id": "PowerMetrics",
    "Name": "PowerMetricsReport",
    "MetricReportDefinition": {"@odata.id": "/redfish/v1/TelemetryService/MetricReportDefinitions/PowerMetrics"},

    "Metrics": {
        {
             "Value": "2.1",
            "Timestamp": "2016-07-25T11:27:59.795513984+02:00",
             "Property": { "@odata.id": "/redfish/v1/Chassis/1/Power#/PowerControl/0/PowerConsumedWatts" }
             "Definition": { "@odata.id": "/redfish/v1/TelemetryService/MetricDefinitions/PowerConsumedWatts" }
        },
        {
             "Value": "2.2",
            "Timestamp": "2016-07-25T11:27:59.795513984+02:00",
             "Property": { "@odata.id": "/redfish/v1/Chassis/1/Power#/PowerControl/1/PowerConsumedWatts" }
             "Definition": { "@odata.id": "/redfish/v1/TelemetryService/MetricDefinitions/PowerConsumedWatts" }
        },
        . . .
    }
}
```

Metric Report

- A resource generated by the creation of a MetricReportDefinition resource.
- Can be examined via GET or be the OriginResource target of a EventDestination
  - EventDestination/Destination may indicate a log or other target for the event.
- Each MetricReport has
  - A reference to the defining MetricReportDefinition
  - An array of metric values
- Each MetricValue has
  - A reference to the corresponding Metric
  - A MetricValue containing the reported value
  - A TimeStamp reporting the time when the reported value was computed.
  - A URI to the property in the entity that this metric was sourced from.

Triggers

Triggers

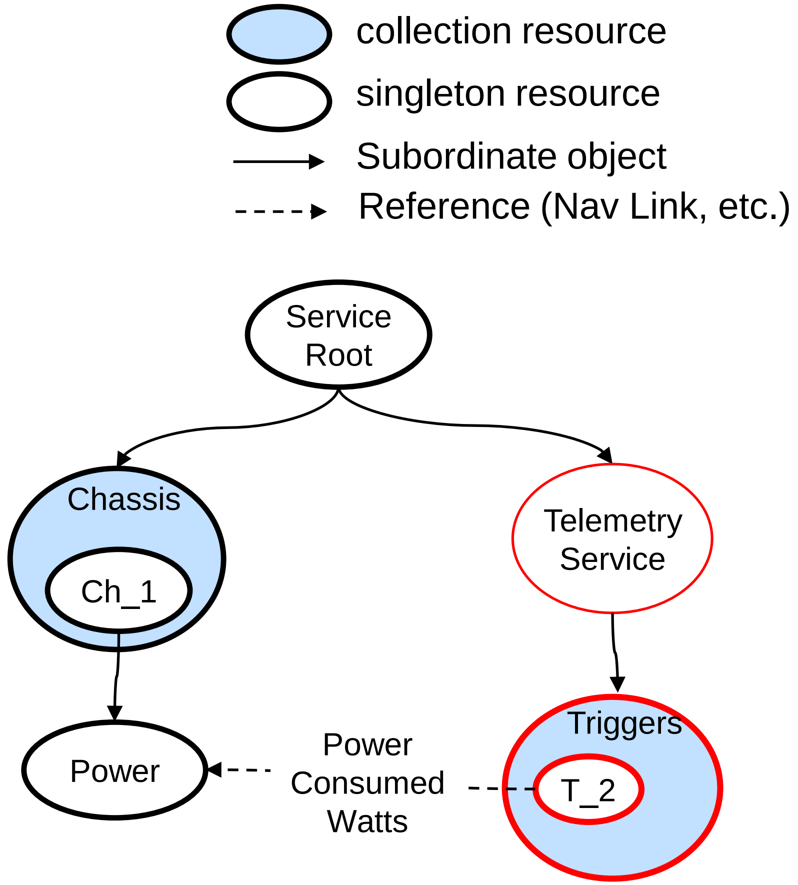
- Some resources already have properties for triggers (e.g Power.xml)
  - Below are two options for alternative locations for trigger information
- Triggers resource
  - A TriggersCollection resource with Triggers members
  - Each member specifies triggers and the list of metrics to which the triggers apply
- MetricReportDefinition resource
  - In the Metrics listing, include a TriggerCondition property to specific a trigger for the metric of interest

/redfish/v1/Chassis/1/Power

```
"Voltages": [
    {
        "@odata.id": ". . .",
        "MemberId": "0",
        "Name": "VRM1 Voltage",
        "SensorNumber": 11,
        "Status": {
            "State": "Enabled",
            "Health": "OK"
        },
        "ReadingVolts": 12,
        "UpperThresholdNonCritical": 12.5,
        "UpperThresholdCritical": 13,
        "UpperThresholdFatal": 15,
        "LowerThresholdNonCritical": 11.5,
        "LowerThresholdCritical": 11,
        "LowerThresholdFatal": 10,
        "MinReadingRange": 0,
        "MaxReadingRange": 20,
        "PhysicalContext": "VoltageRegulator",
        . . .
},
```

Triggers in Triggers Resource

```
{
    "@Redfish.Copyright":"Copyright 2014-2016 Distributed Management Task Force, Inc. (DMTF). All rights reserved.",
    "@odata.context": "/redfish/v1/$metadata#Triggers.Triggers",
    "@odata.type": "#Triggers.v1_0_0.Triggers",
    "@odata.id": "/redfish/v1/TelemetryService/Triggers/PlatformPowerCapTriggers",

    "Id": "PlatformPowerCapTriggers",
    "Name": "Triggers for platform power consumed",
    "MetricType": "Numeric",
    "TriggerActions": ["Transmit"],

    "NumericTriggers": [
        {
            "Name": "UpperThresholdCritical",
            "Value": "50.0",
            "DirectionOfCrossing": "Increasing",
            "DwellTime": "1",
            "Severity": "Critical"
        }
    ],

    "MetricProperties": [
        "/redfish/v1/Chassis/1/Power#/PowerControl/0/PowerConsumedWatts",
        "/redfish/v1/Chassis/1/Power#/PowerControl/1/PowerConsumedWatts"
    ]
}
```



Triggers resource properties

Property	Values	Description
TriggerType	Numeric	The metric report shall be updated periodically. See Schedule property.
	Discrete	The metric report shall be updated when values inside of the measured resource change
NumericTriggers[]		An array of numeric triggers • Name • Value – the threshold value • DirectionOfCrossing – the direction in which the threshold is crossed • DwellTime - The duration in the triggering state before the trigger is invoked • Severity – The severity to use for the Alert event
DiscreteTriggerCondition	Specified	A trigger occurs when the value of the metric becomes one of the values listed in the DiscreteTriggers property
	Change	A trigger occurs whenever the value of the metric changes.
DiscreteTriggers[]		An array of discrete triggers • Name • Value – the discrete value • DwellTime - The duration the reading has the trigger value before the trigger is invoked • Severity – The severity to use for the Alert event
ReportActions[]		An array property to actions to perform • Log - Place a metric report in the location specified by the MetricReport property • Transmit – Send the event as an Alert event
Wildcards[]		An array property with wildcards to substitute in the MetricProperties array property
MetricProperties[]		An array of URIs to metric properties

TriggerCondition in MetricReportDefinition resource example

```
    "@odata.id": "/redfish/v1/TelemetryService/MetricReportDefinitions/TransmitPowerUsage",
    "Id": "TransmitPowerUsage",

    "Name": "TransmitPowerUsage",
    "Schedule": {
        "RecurrenceInterval": "PT0.1S"
    },
    "MetricReportType":"OnChange",
    "CollectionTimeScope": "Interval",
    "MetricReport": {"@odata.id": "/redfish/v1/TelemetryService/MetricReports/TransmitPowerUsage"}}
    "Volatile": false,
    "Wildcards": [
       { "Wild": ["0"] }
    ],
    "Metrics": [{
        "MemberID": "PowerUsageReading",
        "MetricProperties": ["/redfish/v1/Chassis/Tray_1/Power/PowerControl/{Wild}/PowerConsumedWatts"],
        "CollectionDuration": "PT0.020S",
        "TriggerCondition": {
            "DwellInterval": "PT0.001S",
            "TriggerType": "Numeric",
            "NumericTriggerConditions": {
                "Name": "UpperThresholdNonCritical",
                "Value": "48.1",
                "DirectionOfCrossing": "Increasing"
            }
        }
    }]
}
```

MetricReportDefinition resource TriggerCondition properties

Property	Values	Description
TriggerType	Numeric	The metric report shall be updated periodically. See Schedule property.
	Discrete	The metric report shall be updated when values inside of the measured resource change
	Filter	The metric report shall be updated each time a client application reads it
NumericTriggerCondition		The condition that constitutes a trigger for a numeric metric • Name • Value – the threshold value • DirectionOfCrossing – the direction in which the threshold is crossed
DiscreteTriggerCondition		The condition that constitutes a trigger for a discrete metric • Name • TriggerValue – the value of the metric • PreviousValue – the previous value of the metric, if a specific transition is of interest
FilterTriggerCondition		The filter condition that constitutes a trigger • A string with an OData filter condition specified relative to the MetricScope
DwellInterval		The time in the triggering state before the trigger is invoked.
Wildcards[]		An array property with wildcards to substitute in the MetricProperties array property
MetricProperties[]		An array of URIs to metric properties

Schema Change or added

Metadata	Description
EventDestination_v1.xml	
TelemetryService_v1.xml	
MetricDefinitionCollection_v1.xml	
MetricDefintion_v1.xml	
MetricReportDefinitionCollection_v1.xml	
MetricReportDefinition_v1.xml	
MetricReportCollection_v1.xml	
MetricReport_v1.xml	
TriggersCollection_v1.xml	
Triggers_v1.xml	
Metric_v1.xml	
Schedule_v1.xml	

Telemetry Service

TelemetryService resource example

```
{
    "@Redfish.Copyright": "Copyright 2014-2016 Distributed Management Task Force, Inc. (DMTF). All rights reserved.",
    "@odata.context": "/redfish/v1/$metadata#TelemetryService.TelemetryService",
    "@odata.type": "#TelemetryService.1.0.0.TelemetryService",
    "@odata.id": "/redfish/v1/TelemetryService",
    "Id": "TelemetryService",
    "Name": "Telemetry Service",

    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },

    "MetricDefinitions":   { "@odata.id": "/redfish/v1/TelemetryService/MetricDefinitions" },
    "MetricReportDefinitions":   { "@odata.id": "/redfish/v1/TelemetryService/MetricReportDefinitions" },
    "MetricReports":   { "@odata.id": "/redfish/v1/TelemetryService/MetricReports" },
    "Triggers":   { "@odata.id": "/redfish/v1/TelemetryService/Triggers" }
}
```

TelemetryService resource properties

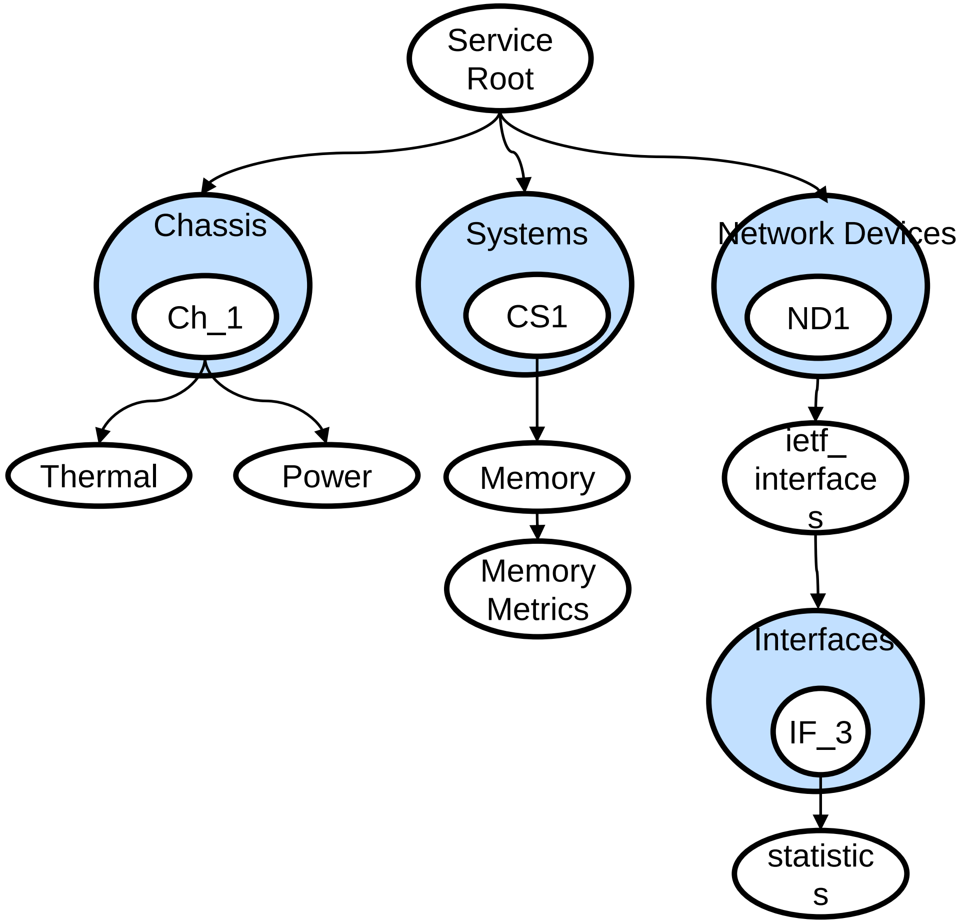
Property	Values	Description
MaxReports		The maximum number of MetricReports that are supported by this service.
MinCollectionInterval		The minimum supported interval between collections.
SupportedCollectionFunctions[]		An array of collection functions • Avg • Max • Min • Sum
Status		Status of Telemetry Service

Existing Redfish sensor/metric structures

(Backup)

Legacy Considerations

- The Redfish model uses resources properties for metric readings throughout the model
- Power and Thermal resources
  - PowerConsumedWatts, ReadVolts, ReadingCelsius properties
  - Also properties for statistics (avg, min & max)
- MemoryMetrics resource
  - Subordinate resource to Memory resource
  - Complex JSON structure (CurrentPeriod, Lifetime)
- Expected statistics resources in YANG
  - Resource with flat name-value pairs



Existing Redfish Structures

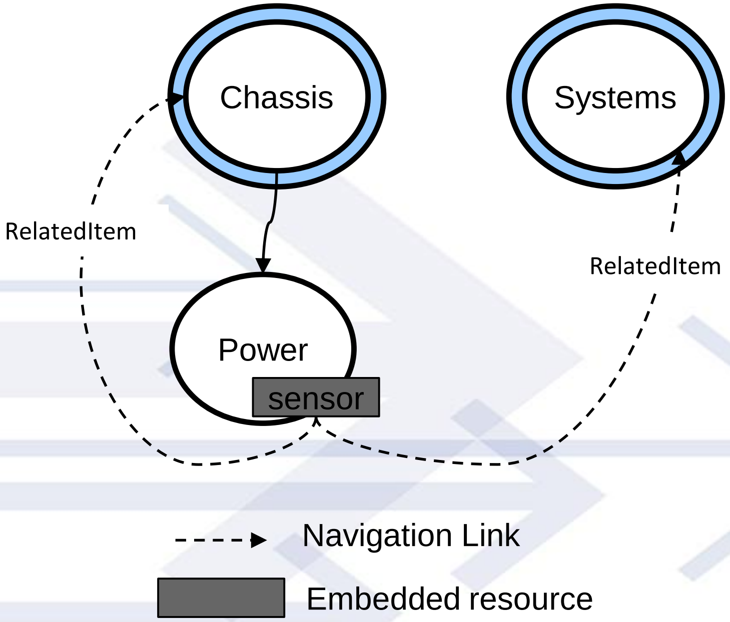
- LogEntry.SensorType
  - Used in ./Chassis/{id}/Power and ./Thermal resources
- LogEntry.LogEntryCode
  - Used ?
- Properties in ./Power and ./Thermal resources
  - PhysicalContext.xml
- MemoryMetrics.xml
- *YANG stat resources*

```
"SensorType": {
  "type": "string",
  "enum": [
  "Platform Security Violation Attempt",
    "Temperature",
  "Voltage",
  "Current",
  "Fan",
```

```
"LogEntryCode": {
   "type": "string",
   "enum": [
      "Assert",
      "Deassert",
      "Lower Non-critical - going low",
      "Lower Non-critical - going high",
      "Lower Critical - going low",
      "Lower Critical - going high",
```

Current Redfish Power resource voltage sensor(s)

- Sensors are embedded within a resource
- Navigation properties used to associate sensor to other resources (chassis, system)



Power Resource

```
{    . . .
     "Voltages": [
        {
            "@odata.id": "",
            "MemberId": "0",
            "Name": "VRM1 Voltage",
            "SensorNumber": 11,
            "Status": { "State": "Enabled", "Health": "OK" },

            "ReadingVolts": 12,
            "UpperThresholdNonCritical": 12.5,
            "UpperThresholdCritical": 13,
            "UpperThresholdFatal": 15,
            "LowerThresholdNonCritical": 11.5,
            "LowerThresholdCritical": 11,
            "LowerThresholdFatal": 10,
            "MinReadingRange": 0,
            "MaxReadingRange": 20,
            "PhysicalContext": "VoltageRegulator",

            "RelatedItem": [
                {"@odata.id": "/redfish/v1/Systems/1" },
                {"@odata.id": "/redfish/v1/Chassis/1" }
            ]
          }
     ]
}
```

MemoryMetrics mockup

```
{
    "@Redfish.Copyright":"Copyright 2014-2016 Distributed Management Task Force, Inc. (DMTF). All rights reserved.",
    "@odata.context": "/redfish/v1/$metadata#MemoryMetrics",
    "@odata.type": "#MemoryMetrics.1.0.0.MemoryMetrics",
    "@odata.id": "/redfish/v1/Systems/CS_1/Memory/Mem_1/Metrics",

    "BlockSizeBytes": "74653",
    "CurrentPeriod": {
        "BlocksRead": 200,
        "BlocksWritten": 100
    }
    "LifeTime": {
        "BlocksRead": 2000,
        "BlocksWritten": 1000
    },
    "HealthData": {
        "RemainingSpareBlocksPercentage": "50.2",
        "LastShutdownSuccess": true,
        "DataLossDetected": false,
        "PerformanceDegraded": false,
        "AlarmTrips": {
            "Temperature": false,
            "SpareBlock": false,
            "UncorrectableECCError": false,
            "CorrectableECCError": true,
            "AddressParityError": false
        }
    }
    . . .
}
```

Backup