Telemetry Service

Clients gather metrics remotely as the first part of the management cycle: sense, analyze, decide, and control. Telemetry are metrics that are obtained from a remote system. The Redfish standard is composed of an interface protocol and a data model. The data model contains resources that express manageability capabilities and services. The Redfish telemetry model defines resources that a Redfish client can use to understand and obtain telemetry from a Redfish service. This enables one to

- Obtain the characteristics and details of a metric (metadata).
- Specify metric reports that periodically report a set of metrics (aggregation).
- Specify trigger thresholds against a metric that is monitored (monitoring).

Metric definitions, defined by the MetricDefinition schema, contain the definition, metadata, or characteristics for a metric. A metric definition contains links to the metric properties to which the definition applies.

Metric report definitions, defined by the MetricReportDefinition schema, specify the metric reports that are generated. The MetricReportDefinition resource specifies the contents and periodicity of the metric report. It also contains links to the metric properties to which the definition applies.

The Redfish service can support the ability to specify a set of triggers or thresholds for a list of metric properties. The Triggers resource specifies the trigger thresholds that apply to the listed metrics. A trigger can result in one or more actions, such as an alert being transmitted using the event service or an event logged in the log service.

The telemetry service for ODIM should implement the following APIs:

- /redfish/v1/TelemetryService
- /redfish/v1/TelemetryService/MetricDefinitions
- /redfish/v1/TelemetryService/MetricReportDefinitions
- /redfish/v1/TelemetryService/MetricReports
- /redfish/v1/TelemetryService/Triggers

Additionally ODIM will have to implement event forwarding in case of triggers that need forwarding reports as events. The details are covered in DSP2051_1.0.0.pdf.

Open issues that need discussion:

The current API are in /redfish/v1/TelemetryService... so that this is suited for individual BMCs to implement. If ODIMRA being a manager implements this then it has to act on behalf of all the managed BMCs. The implication will include

- ODIM will have to take one set of metric/report definitions and reports and triggers and propagate the same to all BMCs
- This implies we cannot have different set of metric/reports across machines and groups of machines
- ODIM will have to be more involved in collections in case it needs to reconcile behaviors across BMC implementations
- ODIM will have scalability issues if BMCs do not support all the needed metrics natively. Then ODIM has to collect related metrics from BMC, calculate the needed metric and possibly trigger events based on the values
- ODIM will also have to disallow the high frequency polling - metric collection can be configured for sub second intervals. However ODIM maybe able to only poll once a few minutes like 10-15 minutes.

One way we can solve this is to make these services a part of the actions under Aggregation Service. The current plugins may not be best fit for collecting this information. ODIM may implement data collector plugins (as suggested by Alex) which will focus only on managing and collecting metrics/reports from BMC.

Initial Release

As a first step we want to release an implementation of the Telemetry Service that is basically just exposing the existing metric reports for servers and allows northbound clients to set up subscriptions for those reports and retrieve them through the Event Service.

Future items

Add AI for analyzing and actioning the telemetry gathered. This may happen in the device, in the plugin, in ODIMRA or north bound clients. Doing this closer to the device helps in quicker actions and this also has the advantage of these systems are better aware of the problem domain. Whereas north bound systems may be used to do long term planning.

Investigate supporting gRPC for NB client Telemetry gathering. grpc is widely seen as the ‘industry standard’ for various eventing usage. This could also be used to deliver events directly from plugins bypassing the message bus and ODIM event services. Software implementations using grpc to deliver events have reported very good performance compared to solutions using message bus and http(s) delivery.