Response to PICMG Proposal Redfish for Industrial IoT

DMTF Redfish Forum

April 21, 2023

Feedback received from PICMG (Feb 2023)

- DMTF Redfish Forum received feedback from PICMG regarding proposed extensions to DSP8010 (Redfish Schema bundle) and DSP8011 (Redfish Registry bundle)
- This presentation includes diagrams and mockups copied from the PICMG feedback for comment
    - PICMG material replicated to show call-out annotations for clarity
- The Forum appreciates the feedback from PICMG and looks forward to working with the group to fully incorporate this feedback into future versions of the Redfish Schema

Summary

- PICMG's feedback contained proposals in three areas:
  - Addition of schema support for Industrial IoT "node" controllers / devices
  - Addition of IIoT Job Service, Job Document, and Job Executor schemas to model industrial and factory processes
  - Additional standard messages for reporting Job state changes
- The Forum concurs with PICMG's assessment, and believe these topics should be incorporated and supported by the Redfish Schema
- Some concepts in the proposal have similar, existing definitions in Redfish schema
  - Those schemas (e.g. Job and JobService) may be extended or at least leveraged to cover these new use cases
  - Further investigation and joint effort would reveal the correct path to incorporate these concepts into schema

Response to “IIoT Node” proposal

- The Forum agrees a new device-centric schema is warranted, and would like to discuss the design further
  - A “Node” is essentially a simple “computer system” instance in the data model, where one or more may be controlled from a single IP address
- The **IoTNode** concept shows promise and may be able to further leverage recent additions and concepts in the Redfish data model
  - The **Sensor** model, coupled with schema “excerpts” can craft purpose-built resources which combine data from multiple, frequently changing sensors
  - Those concepts allow surfacing of **Sensor** and **Control** ‘metadata’ without burdening the frequently-retrieved resources

Response to IIoT Job-related proposals

- Suggest we extend the existing **Job** and **JobService**
  - Many of the proposed additions likely apply to containers / workloads
  - Allows Job-related toolchain to work with a single definition
- Add proposed properties to **JobService**
  - *JobServiceCapabilities* – these are broadly applicable
  - *JobPolicies* – also broadly applicable
  - Add resource collections for **JobExecutors** and **JobDocuments**
- Add the proposed “bulk” actions to **JobService**
  - Service-wide actions: e.g. *SuspendAllJobs, ResumeAllJobs*
- Add proposed new schemas to extend coverage
  - **JobDocument** – Recipe or “template” for a Job instance
    - This concept also seems to fit future use cases of containers and IT workloads
  - **JobExecutor** – an assembly line, machine, or a compute platform servicing items from the job collection / work queue

PICMG Proposed “IIoT Job Service” mockup

```
{
        “@odata.type” : “IIoTJobService.1_0_0.IIoTJobService”,
        “@odata.id” : “/redfish/v1/IIoTJobService”,
        “Actions” : {
                “@odata.type”: “#IIoTJobService.v1_0_0.Actions”,
                “#IIoTJobService.DeleteJob": { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJobService.DeleteJob” },
                “#IIoTJobService.InvalidateAllJobs": { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJobService.InvalidateAllJobs” },
                “#IIoTJobService.ResubmitJob": { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJ
                “#IIoTJobService.SubmitJob": { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJob
                “#IIoTJobService.ResumeAllJobs”: { “target”: “/redfish/v1/IIoTJobService/Actions/IIo
                “#IIoTJobService.SuspendAllJobs”: { “target”: “/redfish/v1/IIoTJobService/Actions/IIo
                “#IIoTJobService.ValidateJob”: { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJobService.ValidateJob” }
        },
        “Id” : “JobService”,
        “Name” : “IIoT Job Servive”,
        “Description” : “IIoT Job Service”,
        “Executors” : {
                “@odata.id” : “/redfish/v1/IIoTJobService/Executors”
        },
        “JobDocuments” : {
                “@odata.id” : “/redfish/v1/IIoTJobService/JobDoc
        },
        “JobPolicies” : {
                “AutoValidation” : true,
                “BypassValidation” : true
        },
        “JobServiceCapabilites” : { “MaxJobs” : 200, “MaxSteps” : 20, “Scheduling” : true },
        “Jobs” : {
                “@odata.id” : “/redfish/v1/IIoTJobService/Jobs”
        },
        “ServiceEnabled” : true,
        “Status” : { “State” : “Enabled”, “Health” : “OK” }
}
```

Some of these actions map to REST requests: POST (submit), DELETE (delete)

Need more detail on what the “validate” process entails and the results

Executors assigned to the service

Job documents serve as “templates” for creating new jobs

The policy and capability properties should be broadly applicable to **JobService** and would be straightforward additions

Job resources tracked by the system

DMTF comments in red

Response to PICMG "JobDocument" proposal

- *MandatoryParameters* and *OptionalParameters*
  - These can directly leverage the *Parameters* definition from **ActionInfo**
  - May need an additional tuple map of "values" (text) to integer/hex/etc.
    - E.g. "Strawberry": 0x53, "Vanilla": 0x5F, "Mango": 0xE9
    - Provide human-readable values, while preserving the machine codes
    - Would this remove need for executor to have text decode tables?
- Link to **Jobs**
  - Prefer to avoid bi-directional link definitions where possible
    - But must ensure the use case can be accomplished from the **Job** collection
    - A **Job** instance should absolutely point to its **JobDocument**
  - Expect the reverse link from **Job** to **JobDocument** to exist

Response to PICMG "JobDocument" proposal, continued

- What happens if this **JobDocument** is modified?
    - How are **Job** instances in progress affected?
        - Replace / halt items in progress?
    - Is a new document instance a new "version"?
- *Creator*
    - Is that always the "parent job"?
    - **Job** has *Steps* to show "sub-jobs" contained by a parent
    - Is this also intended to show the user that created the Job instance itself?
        - We would define a separate property to show that relationship, so that the link(s) can be bound to a specific resource type for interoperability

Response to “JobMetrics” and message proposals

- The contents of the proposed **JobMetrics** are contained in the existing **Job** schema definition
  - Expect the **Job** instance to be the primary monitoring point
  - But if more detailed reporting is needed, that can certainly be discussed as a potential ‘metrics’ resource
- Job-related message registry additions
  - Fully support the addition of new messages to further flesh out the Job message registry
    - PICMG should submit the proposed message registry file additions so the Forum can incorporate and release as a “work in progress” for review
  - Initial release of Job registry was intended to cover FW update use cases
    - Kept the list relatively small to discourage “chatty” update processes

Next Steps

- Schedule joint meeting(s)
  - Provide DMTF opportunity to present response to feedback
  - Present latest applicable Redfish releases and concepts to PICMG
  - Discuss the IoT node concepts in-depth with subject matter experts
- Update the PICMG / DMTF work register to call out these efforts:
  - Extensions to JobService / Job models
  - IIoT Node definition
  - Additions to DMTF standard Message Registries
- Develop timeline / schedule for the work
  - Propose schedule for first work-in-progress release

Redfish Industrial IoT Schema Proposal

Submission from PICMG to DMTF Redfish Forum

February 7, 2023

Context

- Factory 4.0 represents a significant shift in factory automation from the current state. Key attributes are:
  - Convergence of Information Technology and Operations Technology.
  - Internet of Things (smart sensors and effecters)
  - Big data and analytics
  - Custom and reconfigurable machinery
  - Cyber-physical such as digital twinning
- This proposal extends the Redfish schemas to support Factory 4.0 in two main areas:
  - Composable hardware configuration through I/O building blocks
  - Robust job management suitable for factory scheduling

DMTF Feedback

- “Node” concept makes sense as essentially a simple “system”
  - One or more per managed network address…
- We understand the concepts, and want to apply and leverage the Sensor, Control models, and further extend
  - Our “excerpt” concept may work to pull the readings together in the Metrics resource for easy retrieval / modification
- Think this conceptually looks like our “ResourceBlock” model, where a generic blob of stuff is listed as unconfigured, and once configured, also shows up in the tree as a “device” that matches its’ config role…

Where does Redfish Industrial IoT fit in the Factory?

Sensor Nodes
- Smart Sensors report their capabilities to a Sensor Node
- Sensors are exposed using Redfish

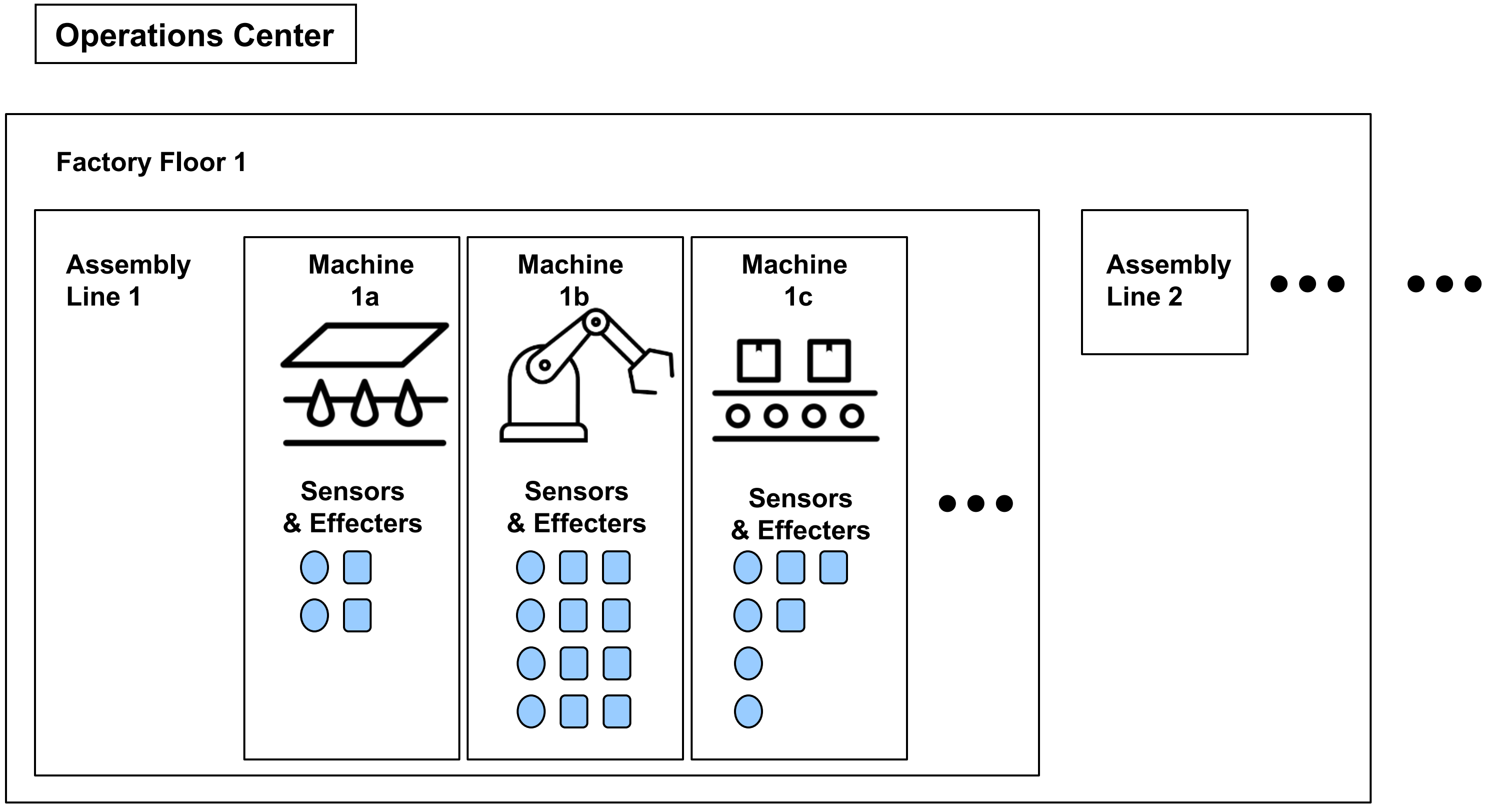
Effector Nodes
- Accept a “Setpoint”
- Controls Motion, Temperature, Pressure, etc.
- Controls are exposed using Redfish

Endpoint Controller
- Integrates multiple Sensor and Effector Nodes
- Can synchronize motion of multiple Effector Nodes
- Sensors and Controls exposed using Redfish

Aggregator/Bridge
- Can control multiple Endpoint Controllers
- Endpoint Controllers exposed using Redfish

Higher levels of integration supported to build a complete factory

Representative Factory 4.0 Installation



All interconnects may use Redfish services

Assumptions on usage

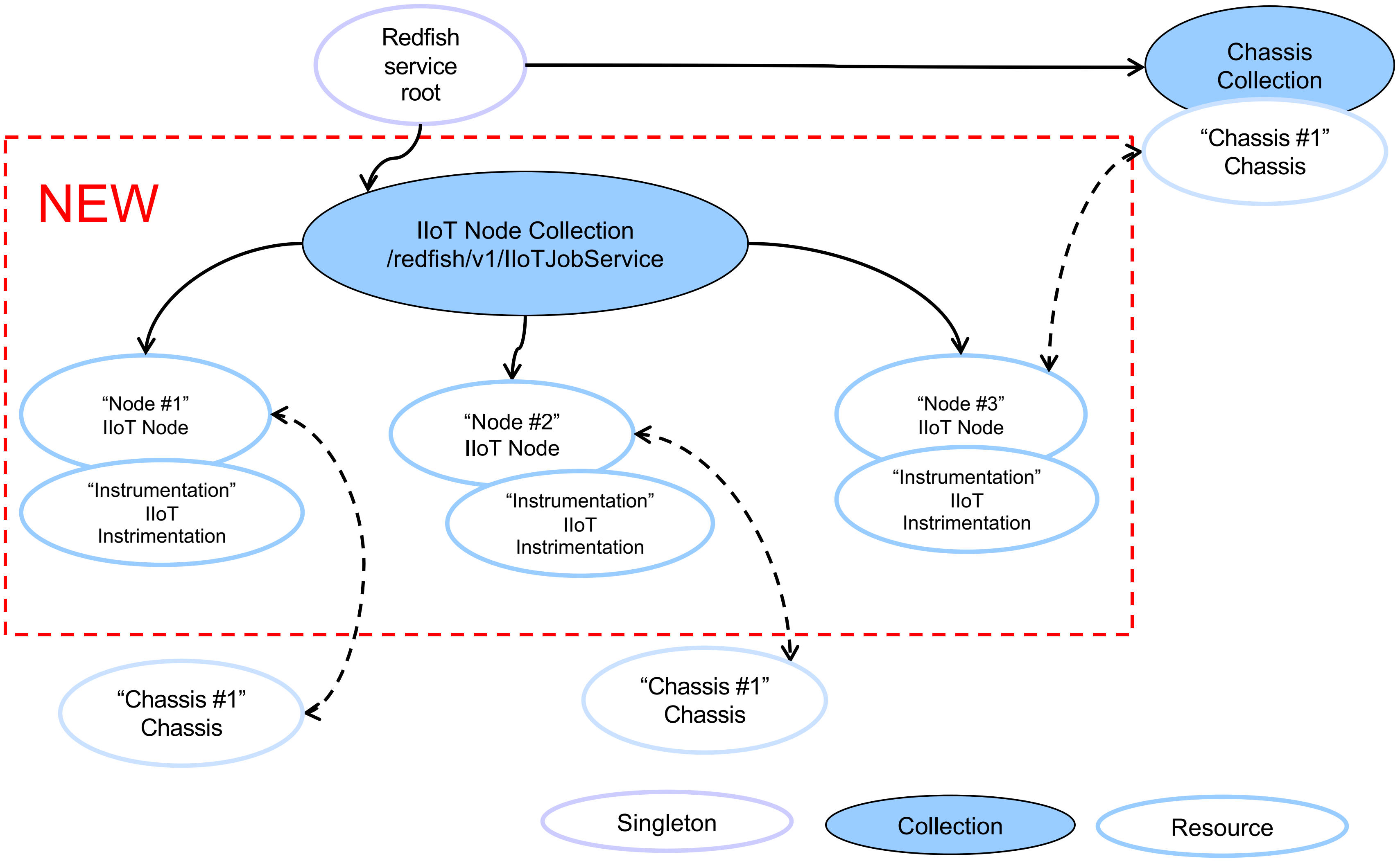
- Complex factory equipment will be composed of collections of IIoTNode hardware building blocks
- IIoTNode objects will expose control and operational metrics to users whose role and permissions allows them to interact with them
  - Direct control of motors will normally be disallowed except in cases of equipment development and debugging
  - Operational metrics typically will have less stringent access requirements
- IIoTJobs comprise factory “recipies” that enable configuration and operation of factory equipment
  - Jobs are interpreted by the equipment’s executor(s) to coordinate the operation of IIoTNodes and other endpoints

IIOTNODE SCHEMA

IIoT Node Purpose

- Provides a single object that encapsulates Smart Sensor or Smart Controller metrics and behaviors for industrial IoT applications
  - Simple sensing
  - PID control
  - Single or multi-axis synchronized motion
- Schema is compatible with PICMG IoT.1 device firmware specification
- Complex factory equipment can be composed of multiple IIoTNode objects.

IIoTNode Architecture



IIoT Node mockup

```
{
    "@odata.type": "#IIoTNode.v1_0_0.IIoTNode",
    "@odata.context": "/redfish/v1/$metadata#IIoTNode.IIoTNode",
    "Id": "XAxisMover",
    "Name": "X-Axis positioner",
    "Description": "",
    "Actions": {
        "@odata.type": "#IIoTNode.v1_0_0.Actions",
        "#IIoTNode.Reset": {
            "target": "/redfish/v1/IIoTNodes/XAxisMover/Actions/Reset"
        },
        "#IIoTNode.SendTrigger": {
            "target": "/redfish/v1/IIoTNodes/XAxisMover/Actions/SendTrigger"
        },
        "#IIoTNode.Start": {
            "target": "/redfish/v1/IIoTNodes/XAxisMover/Actions/Start"
        },
        "#IIoTNode.Stop": {
            "target": "/redfish/v1/IIoTNodes/XAxisMover/Actions/Stop"
        },
        "#IIoTNode.Wait": {
            "target": "/redfish/v1/IIoTNodes/XAxisMover/Actions/Wait"
        }
    },
```

Actions for IIoT Node. These are used to interact with the node's control state machine

IIoT Node mockup (continued)

```
"IIoTFruFields": [
    {
        "RecordNumber": 0,
        "Type": "Model",
        "FieldDescription": "model number",
        "FieldValue": "MicroSAM.001"
    },
    {
        "RecordNumber": 0,
        "Type": "SerialNumber",
        "FieldDescription": "Serial Number of the device",
        "FieldValue": "3"
    }
],
"IIoTInstrumentation" : {
    "@odata.id" : "/redfish/v1/IIoTNodes/XAxisMover/IIoTInstrumentation"
},
```

FRU data from node

Instrumentation (parameters, control and sensor readings) for the node.

IIoT Node mockup (continued)

```
    "Links": {
        "Chassis": {
            "@odata.id" : "/redfish/v1/Chassis/Node1"
        },
        "Controls": {
            "Output": {
                "@odata.id" : "/redfish/v1/Chassis/Node1/Controls/Output"
            }
        },
        "Sensors": {
            "Position": {
                "@odata.id" : "/redfish/v1/Chassis/Node1/Sensors/Position"
            },
            "PositionError": {
                "@odata.id" : "/redfish/v1/Chassis/Node1/Sensors/Perror"
            },
            "Velocity": {
                "@odata.id" : "/redfish/v1/Chassis/Node1/Sensors/Velocity"
            },
            "VelocityError": {
                "@odata.id" : "/redfish/v1/Chassis/Node1/Sensors/Verror"
            }
        }
    },
    "NodeType": "ProfiledMotion",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    }
}
```

Link to the chassis that contains this node

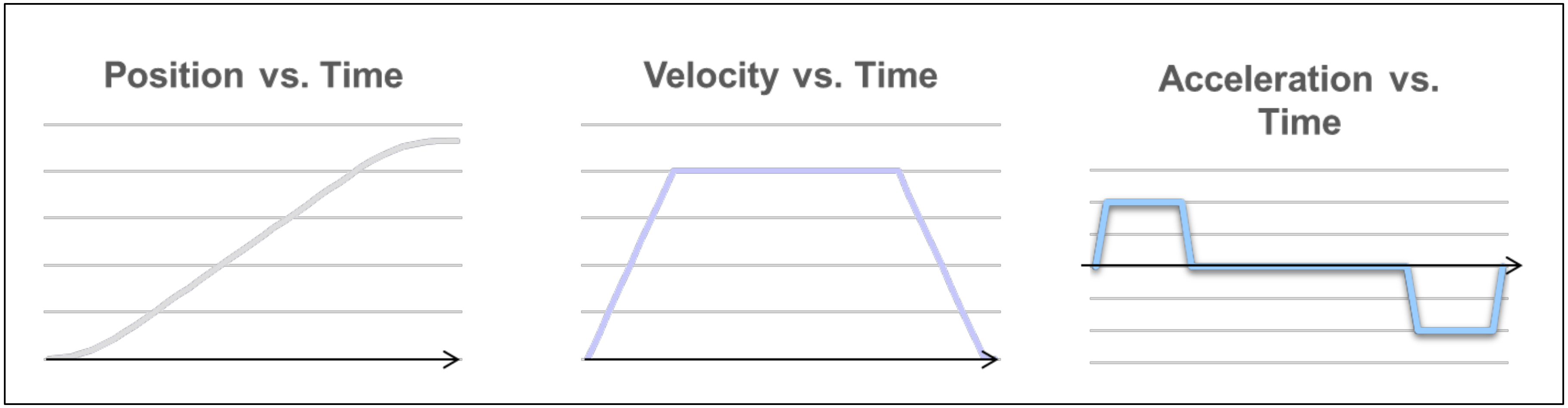
Links to controls associated with this node

Links to sensors associated with this node

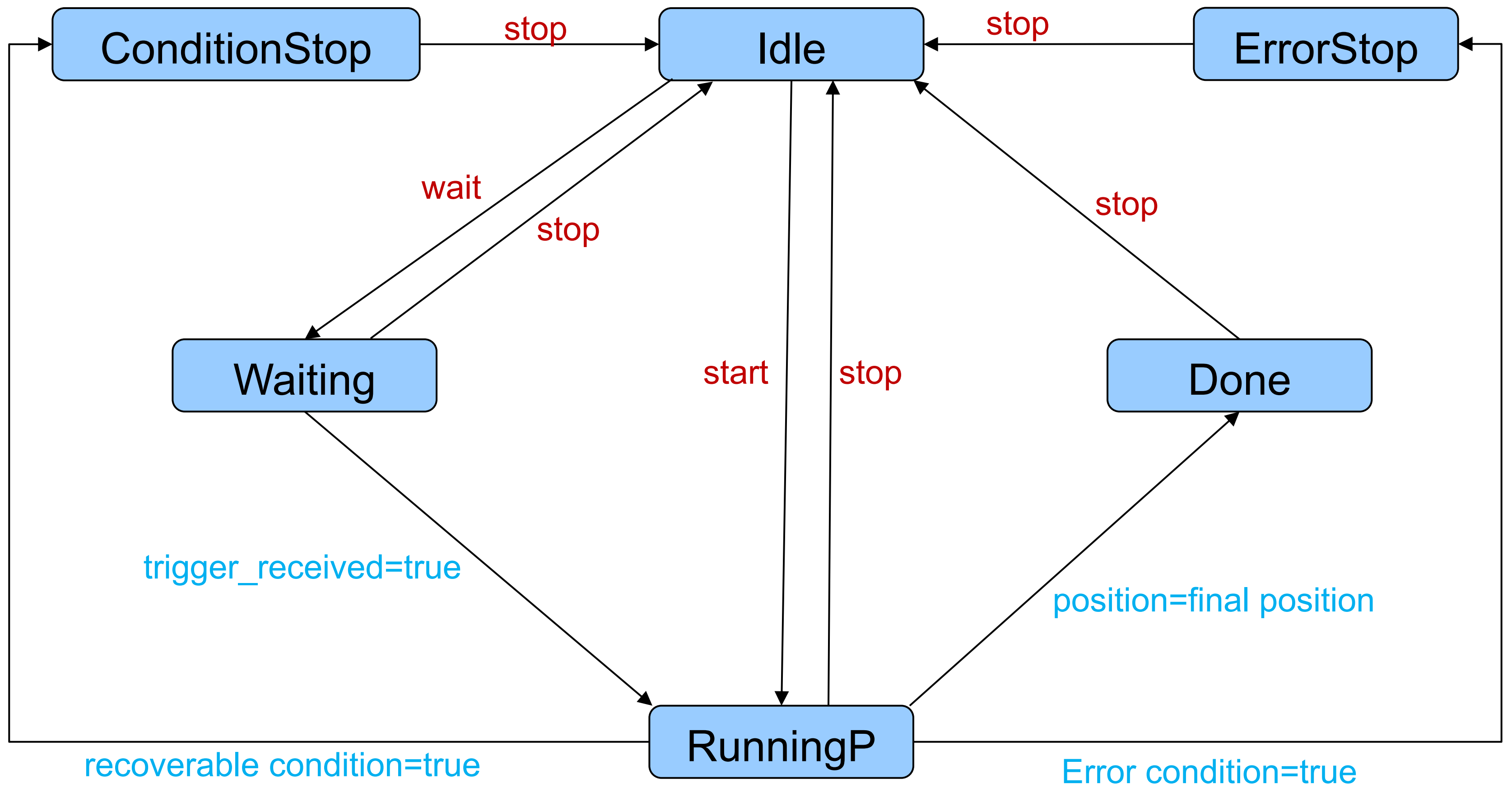
The node type: Profiled motion for synchronized multi-axis control, PID for standard control loop, Simple for sensor nodes.

Profiled Motion

- Used to control velocity and acceleration while moving to a specific set point position.
- This mode is important for a variety of robotics operations
- Multiple axis can be synchronized together for complex motions



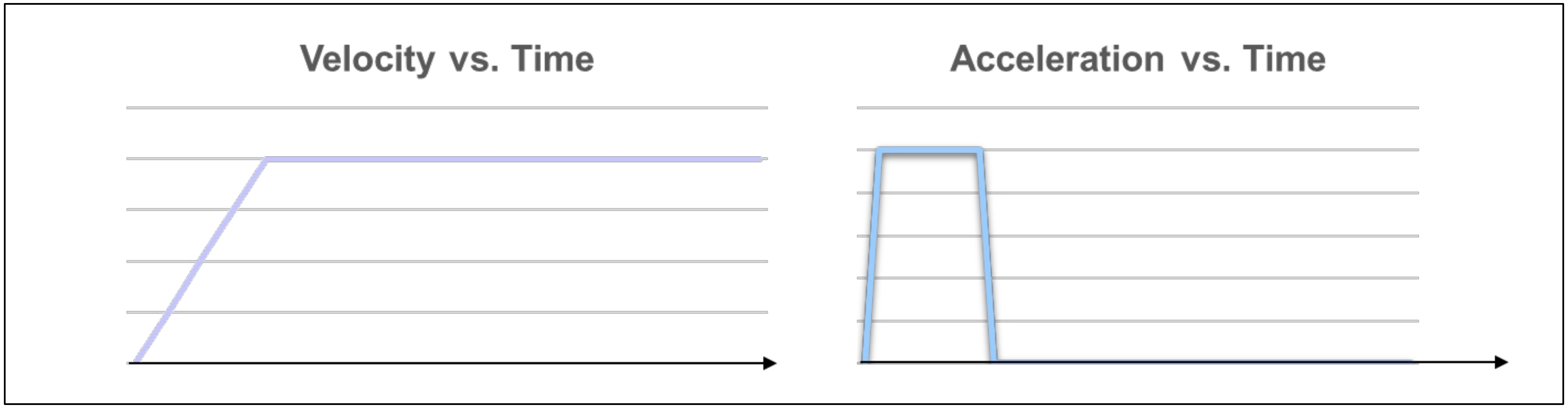
IIoT Node Behaviors for ProfiledMotion position control



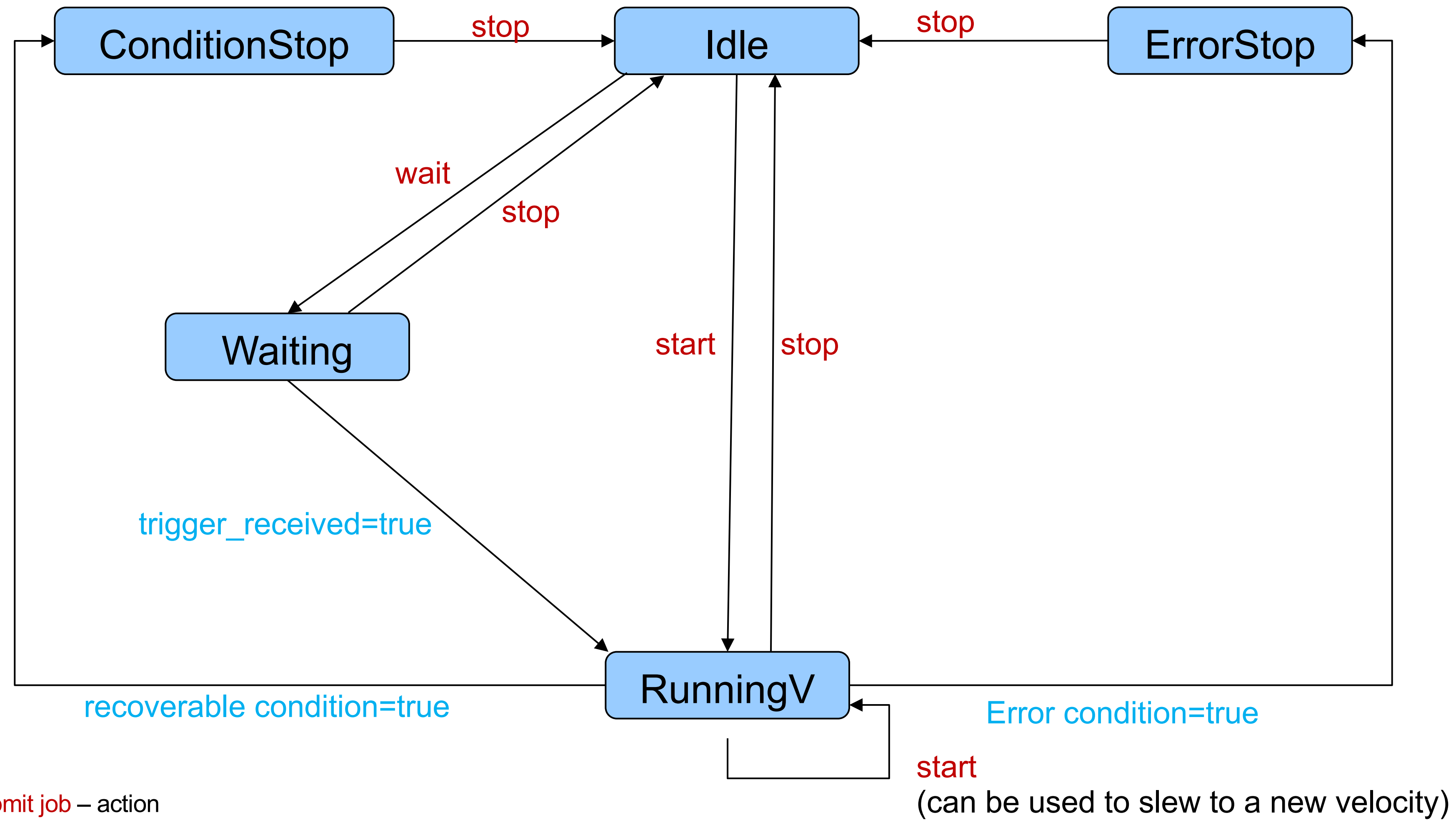
submit job – action
auto-validation=true - condition

Velocity Control (Profiled Motion)

- Used to control an output to a constant velocity (no final position)
- This mode is important for a variety of robotics control operations
- Multiple axis can be synchronized together for complex motions

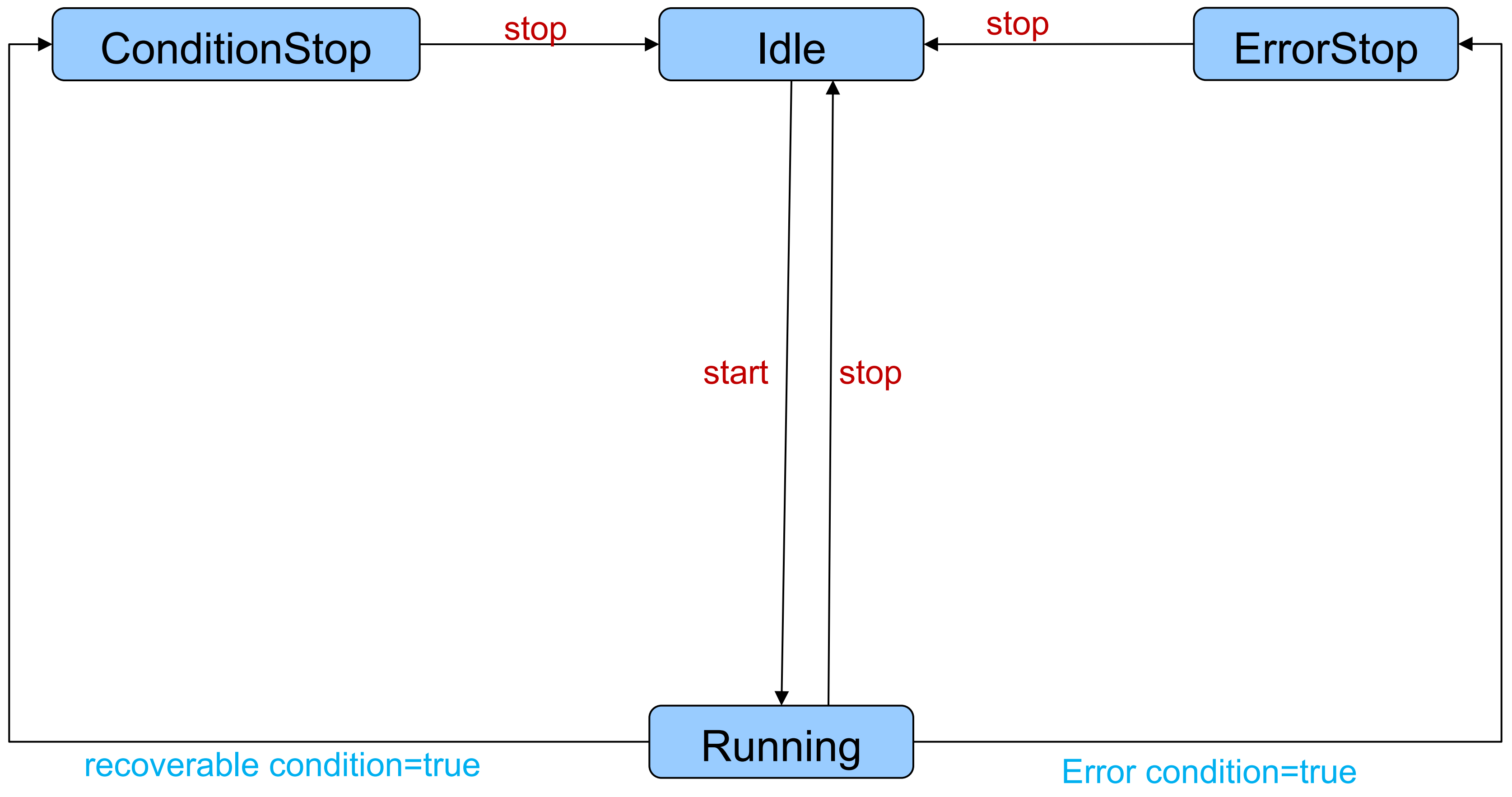


IIoT Node Behaviors for ProfiledMotion velocity control



submit job – action
auto-validation=true - condition

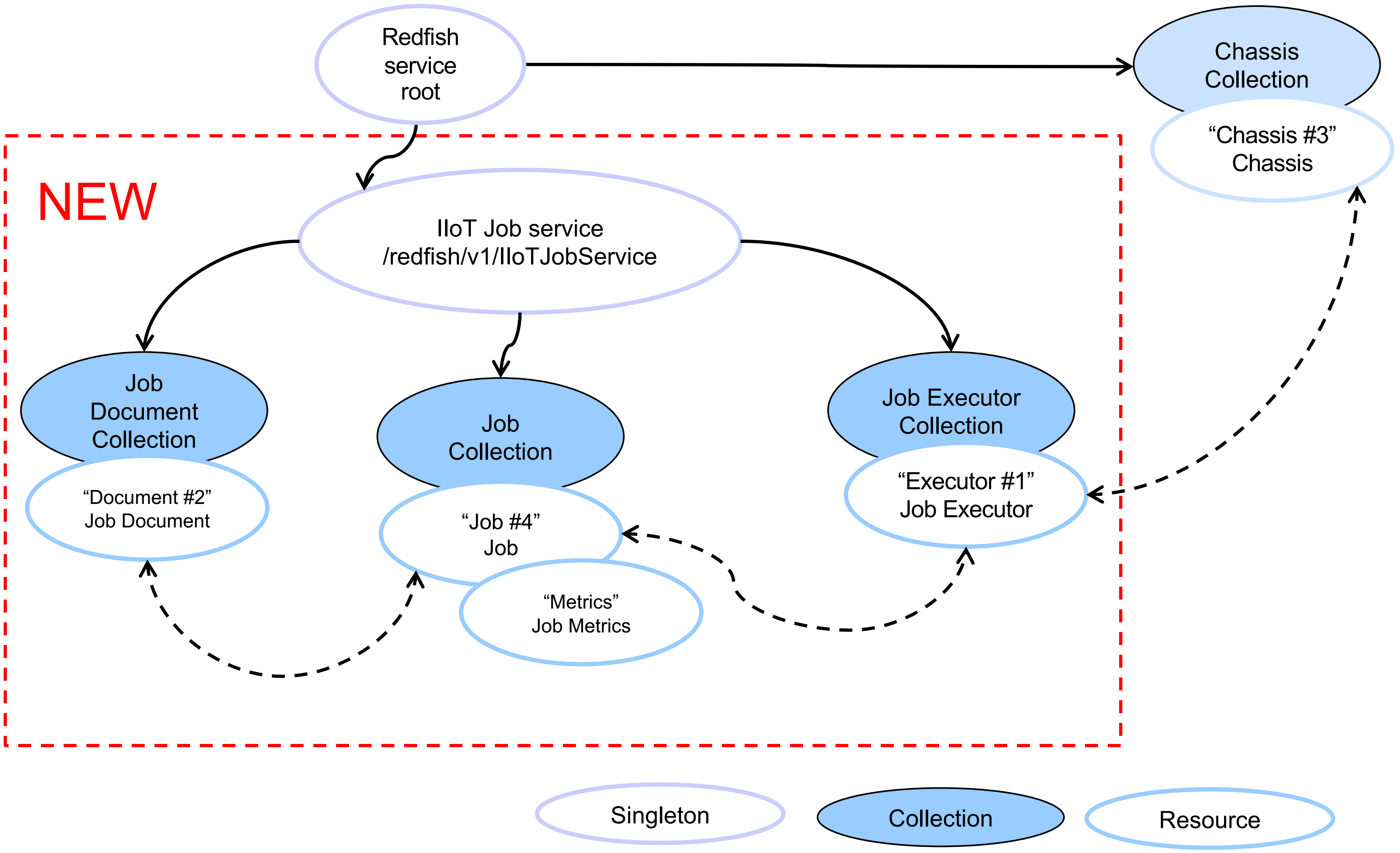
IIoT Node Behavior for PID NodeTypes



submit job – action
auto-validation=true - condition

IIOTJOB AND RELATED SCHEMA

IIoT Job Service Architecture



IIoT Job Service mockup

```
{
        “@odata.type” : “IIoTJobService.1_0_0.IIoTJobService”,
        “@odata.id” : “/redfish/v1/IIoTJobService”,
        “Actions” : {
                “@odata.type”: “#IIoTJobService.v1_0_0.Actions”,
                “#IIoTJobService.DeleteJob": { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJobService.DeleteJob” },
                “#IIoTJobService.InvalidateAllJobs": { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJobService.InvalidateAllJobs” },
                “#IIoTJobService.ResubmitJob": { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJobService.ResubmitJob” },
                “#IIoTJobService.SubmitJob": { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJobService.SubmitJob” },
                “#IIoTJobService.ResumeAllJobs”: { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJobService.ResumeAllJobs” },
                “#IIoTJobService.SuspendAllJobs”: { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJobService.SuspendAllJobs” },
                “#IIoTJobService.ValidateJob”: { “target”: “/redfish/v1/IIoTJobService/Actions/IIoTJobService.ValidateJob” }
        },
        “Id” : “JobService”,
        “Name” : “IIoT Job Servive”,
        “Description” : “IIoT Job Service”,
        “Executors” : {
                “@odata.id” : “/redfish/v1/IIoTJobService/Executors”
        },
        “JobDocuments” : {
                “@odata.id” : “/redfish/v1/IIoTJobService/JobDocuments”
        },
        “JobPolicies” : {
                “AutoValidation” : true,
                “BypassValidation” : true
        },
        “JobServiceCapabilites” : { “MaxJobs” : 200, “MaxSteps” : 20, “Scheduling” : true },
        “Jobs” : {
                “@odata.id” : “/redfish/v1/IIoTJobService/Jobs”
        },
        “ServiceEnabled” : true,
        “Status” : { “State” : “Enabled”, “Health” : “OK” }
}
```

Executors assigned to the service

Job documents serve as “templates” for creating new jobs

Job resources tracked by the system

IIoT Job Document mockup

```
{
	“@odata.type” : “IIoTJobDocument.1_0_0.IIoTJobDocument”,
	“@odata.id” : “/redfish/v1/IIoTJobService/Documents/Recipe1”,
	“Actions” : {
		“@odata.type”: “#IIoTJobDocument.v1_0_0.Actions”,
	},
	“Id” : “Recipe1”,
	“Name” : “Sweet Candy”,
	“Description” : “Produce specified amount of the sweet candies”,
	“DocumentType” : “Script”,
	“MandatoryParameters” : [{
		“Name” : “AmountPieces”,
		“Type” : “edm.UInt64”,
		“Description” : “The number of pieces of sweet candy to produce when running this recipe.  This value must be a positive integer.”
	}, {
		“Name” : “Flavor”:,
		“Type” : “edm.String”,
		“AllowableValues”: [“None”,”Orange”,”Strawberry”,”Lemon”,”Lime”,”Raspberry”,”Vanilla”],
	…
	}],
	“OptionalParameters” : [{
		“Name” : “SecondaryFlavor”,
		“AllowableValues”: [“None”,”Orange”,”Strawberry”,”Lemon”,”Lime”,”Raspberry”,”Vanilla”],
		“DefaultValue” : “None”
	}],
	“Version” : “1.0.0-20210215”,
	“CreationTime” : “2022-01-16T04:14:33+06:00”,
	“DocumentDataUri” : “https://recipestore.vendor.com/recipes?Action=Download&RecipeId=SweetCandy”,
	“Status”: { “State”: “Enabled”, “Health”: “OK” },
	“Links” : {
		“Jobs” : [
			{“@odata.id" : "/redfish/v1/IIoTJobService/Jobs/1” }
		],
	},
}
```

Parameters accepted at job creation (mandatory + optional)

Where the document was obtained from

Registered jobs created from this document

IIoT Job mockup

```
{
        “@odata.type” : “IIoTJob.1_0_0.IIoTJob”,
        “@odata.id” : “/redfish/v1/IIoTJobService/Jobs/1”,
        “Actions” : {
                “@odata.type”: “#IIoTJob.v1_0_0.Actions”,
                “#IIoTJob.CancelJob”: { “target”: “/redfish/v1/IIoTJobService/Jobs/1/Actions/IIoTJob.CancelJob” },
                “#IIoTJob.InvalidateJob": { “target”: “/redfish/v1/IIoTJobService/Jobs/1/Actions/IIoTJob.InvalidateJob” },
                “#IIoTJob.ResumeJob”: { “target”: “/redfish/v1/IIoTJobService/Jobs/1/Actions/IIoTJob.ResumeJob” },
                “#IIoTJob.SuspendJob”: { “target”: “/redfish/v1/IIoTJobService/Jobs/1/Actions/IIoTJob.SuspendJob” }
        },
        “Id” : “1”,
        “Name” : “Job 1”,
        “Description” : “Bake Candies”,
        “Parameters” : {
                “AmountPieces”: 10000,
                “Flavor”: “Orange”,
                “Shape”: [“Cylinder”],
                “SecondaryFlavor”: “None”
        },
        “IIoTJobMetrics”: { “@odata.id”: “/redfish/v1/IIoTJobService/Jobs/1/Metrics” },
        “CreationTime”: “2022-01-16T04:14:33+06:00”,
        “StartTime”: “2022-01-16T04:14:33+06:00”,
        “FinishTime”: null,
        “Status”: { “State”: “Enabled", “Health”: “OK” },
        “Links” : {
                “Creator” : { “@odata.id” : “https://10.1.0.114/redfish/v1/IIoTJobService/Jobs/35” },
                “JobDocument” : { “@odata.id” : “/redfish/v1/IIoTJobService/Documents/Recipe1” },
                “SubsidiaryJobs” : [
                        { “@odata.id”: “https://10.1.13.15/redfish/v1/IIoTJobService/Jobs/54” },
                        { “@odata.id”: “https://10.1.13.16/redfish/v1/IIoTJobService/Jobs/16” },
                        { “@odata.id”: “https://10.1.13.17/redfish/v1/IIoTJobService/Jobs/158” }],
                “Executor” : { “@odata.id” : “/redfish/v1/IIoTJobService/Executors/1” },
                “PreferredExecutors” : [ { “@odata.id” : “/redfish/v1/IIoTJobService/Executors/1” } ],
                “ValidatedExecutors” : [
                        { “@odata.id” : “/redfish/v1/IIoTJobService/Executors/1” } ,
                        { “@odata.id” : “/redfish/v1/IIoTJobService/Executors/2” }
                ]
        }
}
```

Parameters are captured at job submission, following the Document resource

Metrics resource contains the job’s run-time changing status data.

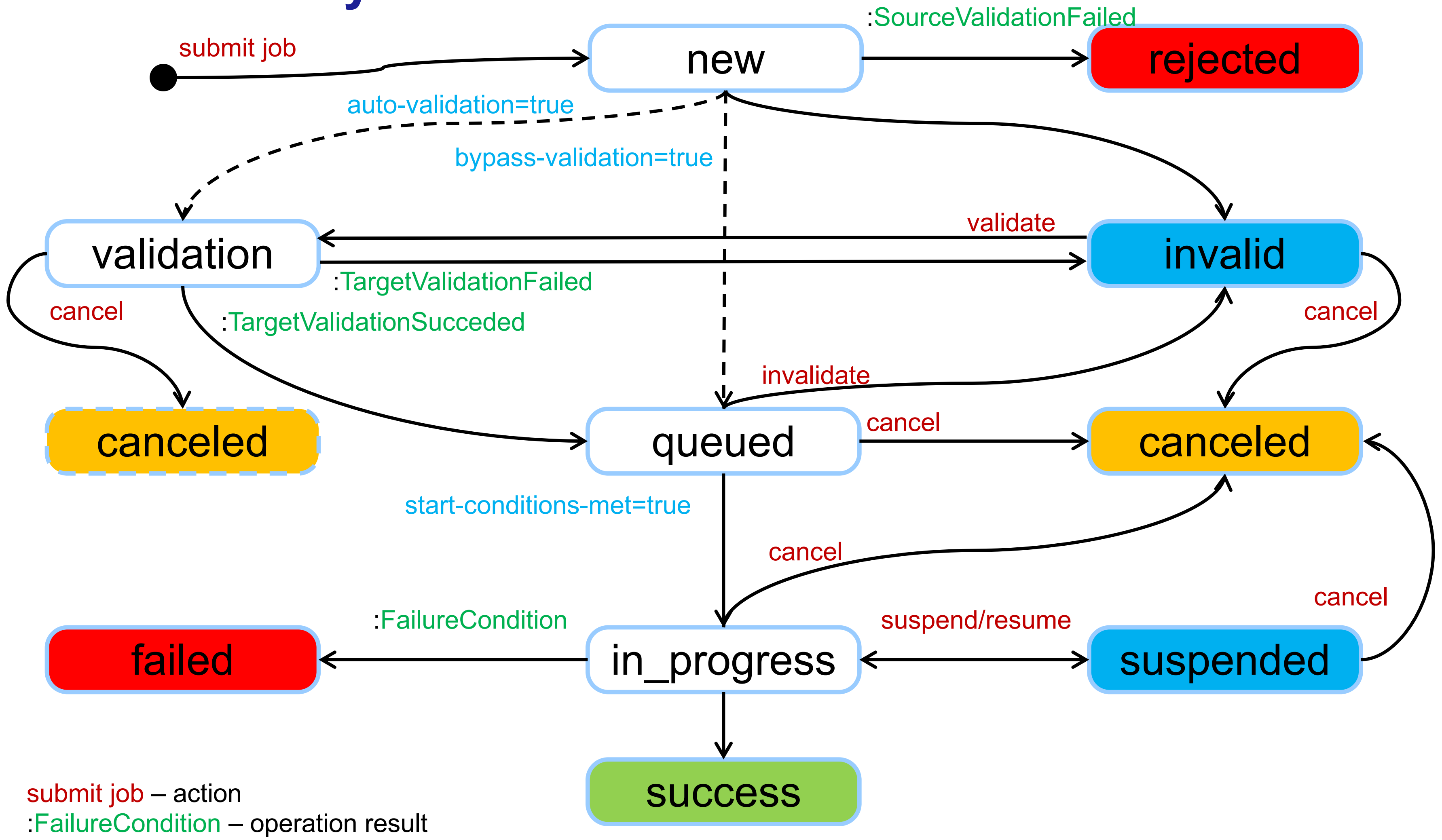
A cross-system link to a job created this one

A “template” used for job submission

Cross-system links to jobs created by this one

Executor running the job

IIoT Job Lifecycle



submit job – action
:FailureCondition – operation result
auto-validation=true - condition

IIoT JobMetrics mockup

```
{
    “@odata.type” : “IIoTJobMetrics.1_0_0.IIoTJobMetrics”,
    “@odata.id” : “/redfish/v1/IIoTJobService/Jobs/1/Metrics”,
    “Id” : “Metrics”,
    “Name” : “Job 1 Metrics”,
    “Description” : “”,
    “PercentCompleted”: 85.3,
    “LastUpdateTime”: “2022-01-16T04:23:33+06:00”,
    “EstimatedCompletionTime”: “2022-01-16T04:30:00+06:00”,
    “JobState” : “Running”,
    “Status”: { “State”: “Enabled", “Health”: “OK” }
}
```

Decimal format allows updating progress in fractions of a percent.

When we last seen a progress update.

One of the job’s life cycle.

IIoT JobExecutor mockup

```
{
    “@odata.type” : “IIoTJobExecutor.1_0_0.IIoTJobExecutor”,
    “@odata.id” : “/redfish/v1/IIoTJobService/Executors/1”,
    “Id” : “1”,
    “Name” : “CandyMaker #1”,
    “Description” : “Candy recipe executor”,
    “ExecutorType” : “OEMScriptEngine”,
    “MaximumConcurrentJobs” : 1,
    “Status”: { “State”: “Enabled", “Health”: “OK” },
    “Links” : {
     "ContainerChassis" : { "@odata.id" : "/redfish/v1/Chassis/1" },
    "ComputerSystem" : { "@odata.id" : "/redfish/v1/Systems/1" },
    “ExecutingJobs” : [ “@odata.id” : “/redfish/v1/IIoTJobService/Jobs/1” ]
    }
}
```

Specifies type of jobs executed.

NEW EVENT MESSAGES

IIoT Job Event Message Registry

- IIoTJobEvent registry
  - Owning Entity: PICMG
- Defines messages for IIoT Job related events
  - Job has started
  - Job has completed
  - Job has completed with warnings or errors
  - Job has been cancelled
  - Job has been removed
  - Job has been suspended
  - Job has been resumed
  - Job progress has changed
  - Job has been queued
  - Job validation has begun

IIoT Node Event Message Registry

- IIoTNodeEvent Registry
  - Owning Entity: PICMG
- Defines messages for IIoT Node related events
  - Node is idle
  - Node is waiting
  - Node is running
  - Node is done running
  - Node has stopped because of conditions
  - Node has stopped because of warnings or errors

BACKUP

JobStarted Event Message

```
"JobStarted": {
		"Description": "A job has started.",
		"LongDescription": "This message shall be used to indicate that a job has started as the `JobState` property
transitions to `InProgress`.",
		"Message": "The job with Id '%1' has started.",
		"Severity": "OK",
		"MessageSeverity": "OK",
		"NumberOfArgs": 1,
		"ParamTypes": [
				"string"
		],
		"ArgDescriptions": [
				"The `Id` of the job that started."
		],
		"ArgLongDescriptions": [
				"This argument shall contain the value of the `Id` property of the job resource that has started."
		],
		"Resolution": "None."
}
```

JobCompletedOk Event Message

```
"JobCompletedOK": {
		"Description": "A job has completed.",
		"LongDescription": "This message shall be used to indicate that a job has completed and the `JobState` property transitioned to
`Completed`.",
		"Message": "The job with Id '%1' has completed.",
		"Severity": "OK",
		"MessageSeverity": "OK",
		"NumberOfArgs": 1,
		"ParamTypes": [
			"string"
		],
		"ArgDescriptions": [
			"The `Id` of the job."
		],
		"ArgLongDescriptions": [
			"This argument shall contain the value of the `Id` property of the job resource that has completed."
		],
		"Resolution": "None.",
		"ClearingLogic": {
			"ClearsIf": "SameOriginOfCondition",
			"ClearsMessage": [
				"JobStarted"
			]
		}
}
```

JobCompletedException Event Message

```
"JobCompletedException": {
		"Description": "A job has completed with warnings or errors.",
		"LongDescription": "This message shall be used to indicate that a job has completed and the `JobState` property transitioned to `Exception`.",
		"Message": "The job with Id '%1' has completed with warnings or errors.",
		"Severity": "Warning",
		"MessageSeverity": "Warning",
		"NumberOfArgs": 1,
		"ParamTypes": [
				"string"
		],
		"ArgDescriptions": [
				"The `Id` of the job."
		],
		"ArgLongDescriptions": [
				"This argument shall contain the value of the `Id` property of the job resource that has completed with warnings or errors."
		],
		"Resolution": "None.",
		"ClearingLogic": {
				"ClearsIf": "SameOriginOfCondition",
				"ClearsMessage": [
						"JobStarted",
						"JobSuspended"
				]
		}
}
```

JobCancelled Event Message

```
"JobCancelled": {
		"Description": "A job has been cancelled.",
		"LongDescription": "This message shall be used to indicate that a job has been cancelled and the `JobState` property 
transitioned to `Cancelled`.",
		"Message": "The job with Id '%1' has been cancelled.",
		"Severity": "Warning",
		"MessageSeverity": "Warning",
		"NumberOfArgs": 1,
		"ParamTypes": [
			"string"
		],
		"ArgDescriptions": [
			"The `Id` of the job."
		],
		"ArgLongDescriptions": [
			"This argument shall contain the value of the `Id` property of the job resource that has been cancelled."
		],
		"Resolution": "None.",
		"ClearingLogic": {
			"ClearsIf": "SameOriginOfCondition",
			"ClearsMessage": [
				"JobStarted"
			]
		}
}
```

JobRemoved Event Message

```
"JobRemoved": {
		"Description": "A job has been removed.",
		"LongDescription": "This message shall be used to indicate that a job has been removed.",
		"Message": "The job with Id '%1' has been removed.",
		"Severity": "Warning",
		"MessageSeverity": "Warning",
		"NumberOfArgs": 1,
		"ParamTypes": [
				"string"
		],
		"ArgDescriptions": [
				"The `Id` of the job."
		],
		"ArgLongDescriptions": [
				"This argument shall contain the value of the `Id` property of the job resource that has been removed."
		],
		"Resolution": "None.",
		"ClearingLogic": {
				"ClearsIf": "SameOriginOfCondition",
				"ClearsAll": true
		}
}
```

JobSuspended Event Message

```
"JobSuspended": {
		"Description": "A job has been suspended.",
		"LongDescription": "This message shall be used to indicate that a job has been susepended and the 
`JobState` property transitioned to `Suspended`.",
		"Message": "The job with Id '%1' has been suspended.",
		"Severity": "Warning",
		"MessageSeverity": "Warning",
		"NumberOfArgs": 1,
		"ParamTypes": [
			"string"
		],
		"ArgDescriptions": [
			"The `Id` of the job."
		],
		"ArgLongDescriptions": [
			"This argument shall contain the value of the `Id` property of the job resource that has been 
suspended."
		],
		"Resolution": "None."
}
```

JobResumed Event Message

```
"JobResumed": {
		"Description": "A job has resumed.",
		"LongDescription": "This message shall be used to indicate that a job has resumed and the `JobState` property transitioned to `InProgress` from `Suspended`.",
		"Message": "The job with Id '%1' has resumed.",
		"Severity": "OK",
		"MessageSeverity": "OK",
		"NumberOfArgs": 1,
		"ParamTypes": [
			"string"
		],
		"ArgDescriptions": [
			"The `Id` of the job."
		],
		"ArgLongDescriptions": [
			"This argument shall contain the value of the `Id` property of the job resource that has resumed."
		],
		"Resolution": "None.",
		"ClearingLogic": {
			"ClearsIf": "SameOriginOfCondition",
			"ClearsMessage": [
				"JobSuspended"
			]
		}
}
```

JobProgressChanged

```
"JobProgressChanged": {
		"Description": "A job has changed progress.",
		"LongDescription": "This message shall be used to indicate that a job has changed progress.",
		"Message": "The job with Id '%1' has changed to progress %2 percent complete.",
		"Severity": "OK",
		"MessageSeverity": "OK",
		"NumberOfArgs": 2,
		"ParamTypes": [
			"string",
			"number"
		],
		"ArgDescriptions": [
			"The `Id` of the job.",
			"The percent completion of the job."
		],
		"ArgLongDescriptions": [
			"This argument shall contain the value of the `Id` property of the job resource that has changed progress.",
			"This argument shall contain the percent completion of the job that has changed progress and shall be a value from 0 to 100."
		],
		"Resolution": "None."
}
```

JobQueued Event Message

```
"JobQueued": {
		"Description": "A job has been queued.",
		"LongDescription": "This message shall be used to indicate that a job has been queued and the `JobState` 
property transitioned to `Queued`.",
		"Message": "The job with Id '%1' has been queued.",
		"Severity": "OK",
		"MessageSeverity": "OK",
		"NumberOfArgs": 1,
		"ParamTypes": [
			"string"
		],
		"ArgDescriptions": [
			"The `Id` of the job."
		],
		"ArgLongDescriptions": [
			"This argument shall contain the value of the `Id` property of the job resource that has been 
susepended."
		],
		"Resolution": "None."
}
```

JobValidationStarted Event Message

```
"JobValidationStarted": {
		"Description": "A job validation has been started.",
		"LongDescription": "This message shall be used to indicate that a job has started self-validation as the `JobState` property transitions to `Validation`.",
		"Message": "The job with Id '%1' is under validation.",
		"Severity": "OK",
		"MessageSeverity": "OK",
		"NumberOfArgs": 1,
		"ParamTypes": [
				"string"
		],
		"ArgDescriptions": [
				"The `Id` of the job that is being validated."
		],
		"ArgLongDescriptions": [
				"This argument shall contain the value of the `Id` property of the job resource that is being validated."
		],
		"Resolution": "None."
}
```

JobValidatedOK Event Message

```
"JobValidatedOK": {
		"Description": "A job validation has completed.",
		"LongDescription": "This message shall be used to indicate that a job has completed and the `JobState` property transitioned
to `Queued`.",
		"Message": "The job with Id '%1' has been validated.",
		"Severity": "OK",
		"MessageSeverity": "OK",
		"NumberOfArgs": 1,
		"ParamTypes": [
			"string"
		],
		"ArgDescriptions": [
			"The `Id` of the job."
		],
		"ArgLongDescriptions": [
			"This argument shall contain the value of the `Id` property of the job resource that has been validated."
		],
		"Resolution": "None.",
		"ClearingLogic": {
			"ClearsIf": "SameOriginOfCondition",
			"ClearsMessage": [
				"Validation"
			]
		}
}
```

JobValidatedException Event Message

```
"JobValidatedException": {
		"Description": "A job validation has completed with warnings or errors.",
		"LongDescription": "This message shall be used to indicate that a job validation has completed and the `JobState` property transitioned to `Invalid`.",
		"Message": "The job with Id '%1' has completed validation with warnings or errors.",
		"Severity": "Warning",
		"MessageSeverity": "Warning",
		"NumberOfArgs": 1,
		"ParamTypes": [
			"string"
		],
		"ArgDescriptions": [
			"The `Id` of the job."
		],
		"ArgLongDescriptions": [
			"This argument shall contain the value of the `Id` property of the job resource that has completed validation with warnings or errors."
		],
		"Resolution": "None.",
		"ClearingLogic": {
			"ClearsIf": "SameOriginOfCondition",
			"ClearsMessage": [
				"Validation"
			]
		}
}
```

NodeIdle Event Message

```
"NodeIdle": {
        "Description": "A node is idle.",
        "LongDescription": "This message shall be used to indicate that a node `State` property transitions to `Idle`.",
        "Message": "The node with Id '%1' is idle.",
        "Severity": "OK",
        "MessageSeverity": "OK",
        "NumberOfArgs": 1,
        "ParamTypes": [
                "string"
        ],
        "ArgDescriptions": [
                "The `Id` of the node that is idle."
        ],
        "ArgLongDescriptions": [
                "This argument shall contain the value of the `Id` property of the node resource that is idle."
        ],
        "Resolution": "None."
}
```

NodeWaiting Event Message

```
"NodeWaiting": {
            "Description": "A node is waiting.",
            "LongDescription": "This message shall be used to indicate that a node `State` property transitioned to `Waiting`.",
            "Message": "The node with Id '%1' is waiting.",
            "Severity": "OK",
            "MessageSeverity": "OK",
            "NumberOfArgs": 1,
            "ParamTypes": [
                        "string"
            ],
            "ArgDescriptions": [
                        "The `Id` of the node."
            ],
            "ArgLongDescriptions": [
                        "This argument shall contain the value of the `Id` property of the node resource that is waiting."
            ],
            "Resolution": "None."
}
```

NodeRunning Event Message

```
"NodeRunning": {
        "Description": "A node is running.",
        "LongDescription": "This message shall be used to indicate that a node `State` property transitioned to one of the ‘Running’ states.",
        "Message": "The node with Id '%1' is running.",
        "Severity": "OK",
        "MessageSeverity": "OK",
        "NumberOfArgs": 1,
        "ParamTypes": [
                "string"
        ],
        "ArgDescriptions": [
                "The `Id` of the node."
        ],
        "ArgLongDescriptions": [
                "This argument shall contain the value of the `Id` property of the node resource that is running."
        ],
        "Resolution": "None."
},
```

NodeDone Event Message

```
"NodeDone": {
		"Description": "A node is done running.",
		"LongDescription": "This message shall be used to indicate that a node `State` property transitioned to `Done`.",
		"Message": "The node with Id '%1' is done running.",
		"Severity": "OK",
		"MessageSeverity": "OK",
		"NumberOfArgs": 1,
		"ParamTypes": [
				"string"
		],
		"ArgDescriptions": [
				"The `Id` of the node."
		],
		"ArgLongDescriptions": [
				"This argument shall contain the value of the `Id` property of the node resource that is done running."
		],
		"Resolution": "None."
}
```

NodeConditionStop Event Message

```
"NodeConditionStop": {
		"Description": "A node has stopped because of conditions.",
		"LongDescription": "This message shall be used to indicate that a node has stopped and the `State` property transitioned to 
`ConditionStop`.",
		"Message": "The node with Id '%1' has stopped because of conditions.",
		"Severity": "Warning",
		"MessageSeverity": "Warning",
		"NumberOfArgs": 1,
		"ParamTypes": [
				"string"
		],
		"ArgDescriptions": [
				"The `Id` of the node."
		],
		"ArgLongDescriptions": [
				"This argument shall contain the value of the `Id` property of the node resource that has stopped."
		],
		"Resolution": "None."
}
```

NodeErrorStop Event Message

```
"NodeErrorStop": {
        "Description": "A node has stopped with warnings or errors.",
        "LongDescription": "This message shall be used to indicate that a node has stopped and the `State` property transitioned to
`ErrorStop`.",
        "Message": "The node with Id '%1' has stopped with warnings or errors.",
        "Severity": "Warning",
        "MessageSeverity": "Warning",
        "NumberOfArgs": 1,
        "ParamTypes": [
                "string"
        ],
        "ArgDescriptions": [
                "The `Id` of the node."
        ],
        "ArgLongDescriptions": [
                "This argument shall contain the value of the `Id` property of the node resource that has stopped."
        ],
        "Resolution": "None."
}
```