## Contents

1 Introduction ........................................................................................................................................................................... 7
   1.1 Scope .............................................................................................................................................................................. 7
   1.2 Intended audience ............................................................................................................................................................ 7
   1.3 Terminology ..................................................................................................................................................................... 7
   1.4 References ........................................................................................................................................................................ 8

2 PODM API ............................................................................................................................................................................. 9
   2.1 PODM API structure and relations ................................................................................................................................ 9
      2.1.1 PODM API physical resource hierarchy .................................................................................................................. 9

3 PODM REST API Error Codes ............................................................................................................................................. 11
   3.1 API error response .......................................................................................................................................................... 11
      3.1.1 Example error JSON object ........................................................................................................................................ 11
   3.2 API error codes ................................................................................................................................................................. 11
      3.2.1 General error codes ..................................................................................................................................................... 12
      3.2.2 Request error codes ................................................................................................................................................... 12
      3.2.3 Node assembly error codes ....................................................................................................................................... 12
      3.2.4 Node action error codes ............................................................................................................................................. 12

4 PODM REST API Definition .................................................................................................................................................. 13
   4.1 Pod manager service root .................................................................................................................................................. 13
      4.1.1 Metadata ..................................................................................................................................................................... 13
      4.1.2 JSON serialization ....................................................................................................................................................... 13
      4.1.3 Operations .................................................................................................................................................................. 14
   4.2 Pod collection ..................................................................................................................................................................... 14
      4.2.1 Metadata ..................................................................................................................................................................... 15
      4.2.2 JSON serialization ....................................................................................................................................................... 15
      4.2.3 Operations .................................................................................................................................................................. 15
   4.3 Pod ....................................................................................................................................................................................... 16
      4.3.1 Metadata ..................................................................................................................................................................... 16
      4.3.2 JSON serialization ....................................................................................................................................................... 16
      4.3.3 Operations .................................................................................................................................................................. 17
   4.4 Rack collection ................................................................................................................................................................. 18
      4.4.1 Metadata ..................................................................................................................................................................... 18
      4.4.2 JSON serialization ....................................................................................................................................................... 18
      4.4.3 Operations .................................................................................................................................................................. 19
   4.5 Rack ....................................................................................................................................................................................... 20
      4.5.2 Metadata ..................................................................................................................................................................... 22
      4.5.3 JSON serialization ....................................................................................................................................................... 22
      4.5.4 Operations .................................................................................................................................................................. 24
   4.6 Storage service collection .................................................................................................................................................. 27
      4.6.1 Metadata ..................................................................................................................................................................... 27
      4.6.2 JSON serialization ....................................................................................................................................................... 27
      4.6.3 Operations .................................................................................................................................................................. 28
   4.7 Storage service ................................................................................................................................................................. 28
   4.8 Remote target collection ................................................................................................................................................... 28
   4.9 Logical drive collection ..................................................................................................................................................... 28
   4.10 Physical drive collection ................................................................................................................................................... 29
   4.11 Drawer collection .............................................................................................................................................................. 29
   4.12 Drawer ................................................................................................................................................................................ 29
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.13 Compute module collection</td>
<td>29</td>
</tr>
<tr>
<td>4.14 Compute module</td>
<td>29</td>
</tr>
<tr>
<td>4.15 Blade collection</td>
<td>29</td>
</tr>
<tr>
<td>4.16 Blade</td>
<td>29</td>
</tr>
<tr>
<td>4.17 Processor collection</td>
<td>29</td>
</tr>
<tr>
<td>4.18 Processor</td>
<td>29</td>
</tr>
<tr>
<td>4.19 Manager collection</td>
<td>29</td>
</tr>
<tr>
<td>4.20 Manager</td>
<td>29</td>
</tr>
<tr>
<td>4.20.1 Metadata</td>
<td>30</td>
</tr>
<tr>
<td>4.20.2 JSON serialization</td>
<td>30</td>
</tr>
<tr>
<td>4.20.3 Operation</td>
<td>32</td>
</tr>
<tr>
<td>4.21 Storage manager</td>
<td>33</td>
</tr>
<tr>
<td>4.22 Composed node collection</td>
<td>33</td>
</tr>
<tr>
<td>4.22.1 Metadata</td>
<td>34</td>
</tr>
<tr>
<td>4.22.2 JSON serialization</td>
<td>34</td>
</tr>
<tr>
<td>4.22.3 Operation</td>
<td>34</td>
</tr>
<tr>
<td>4.23 Composed node</td>
<td>37</td>
</tr>
<tr>
<td>4.23.1 Metadata</td>
<td>38</td>
</tr>
<tr>
<td>4.23.2 JSON serialization</td>
<td>39</td>
</tr>
<tr>
<td>4.23.3 Operations</td>
<td>40</td>
</tr>
<tr>
<td>4.24 Network service</td>
<td>43</td>
</tr>
<tr>
<td>4.25 Network interface collection</td>
<td>43</td>
</tr>
<tr>
<td>4.26 Network interface</td>
<td>43</td>
</tr>
<tr>
<td>4.27 VLAN network interface collection</td>
<td>43</td>
</tr>
<tr>
<td>4.28 VLAN network interface</td>
<td>43</td>
</tr>
</tbody>
</table>
Figures

Figure 1  PODM REST API hierarchy .......................................................... 9
Figure 2  ComposedNodeState changes during assembly process .............. 38

Tables

Table 1  Terminology .................................................................................................. 7
Table 2  Reference documents .................................................................................. 8
Table 3  Resources and URI ..................................................................................... 9
Table 4  API error response .................................................................................... 11
Table 5  General error codes ................................................................................... 12
Table 6  Request error codes ................................................................................... 12
Table 7  Node assembly error codes ....................................................................... 12
Table 8  Node disassembly error codes .................................................................. 12
Table 9  Pod manager service root attributes ......................................................... 13
Table 10 Pod collection attributes .......................................................................... 15
Table 11 Pod attributes ........................................................................................... 16
Table 12 Rack collection attributes ....................................................................... 18
Table 13 Rack attributes ........................................................................................ 20
Table 14 TrayPresent attributes ............................................................................. 20
Table 15 Thermal zone attributes ........................................................................... 21
Table 16 Power zone attributes .............................................................................. 21
Table 17 Storage service collection attributes ....................................................... 27
Table 18 Manager attributes .................................................................................... 30
Table 19 Composed node collection attributes ..................................................... 34
Table 20 Composed node attributes ...................................................................... 37
## Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Initial public release.</td>
<td>August 3, 2015</td>
</tr>
<tr>
<td>002</td>
<td>Minor updates.</td>
<td>August 17, 2015</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Scope
This document contains information about the Intel® Rack Scale Architecture Pod Manager REST API, which was designed and implemented for the v1.1 release for the Bulldog Creek SDV.

1.2 Intended audience
The intended audiences for this document include designers and engineers working with the v1.1 release.

1.3 Terminology

Table 1 Terminology

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC</td>
<td>Baseboard management controller</td>
</tr>
<tr>
<td>CIMI</td>
<td>Cloud Infrastructure Management Interface</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>JSON</td>
<td>JavaScript Object Notation</td>
</tr>
<tr>
<td>NIC</td>
<td>Network interface card</td>
</tr>
<tr>
<td>OCCI</td>
<td>Open Cloud Computing Interface</td>
</tr>
<tr>
<td>OData</td>
<td>Open Data Protocol</td>
</tr>
<tr>
<td>OVF</td>
<td>Open virtualization format</td>
</tr>
<tr>
<td>pod</td>
<td>A physical collection of multiple racks.</td>
</tr>
<tr>
<td>PODM</td>
<td>Pod Manager</td>
</tr>
<tr>
<td>PSME</td>
<td>Pooled System Management Engine</td>
</tr>
<tr>
<td>REST</td>
<td>Representational state transfer</td>
</tr>
<tr>
<td>SDV</td>
<td>Software development vehicle</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform resource identifier</td>
</tr>
<tr>
<td>UUID</td>
<td>Universally unique identifier</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
1.4 References

<table>
<thead>
<tr>
<th>Number</th>
<th>Title</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>332971</td>
<td>Intel® Rack Scale Architecture Getting Started Guide</td>
<td><a href="https://github.com/01org/RSA">https://github.com/01org/RSA</a></td>
</tr>
<tr>
<td>332868</td>
<td>Intel® Rack Scale Architecture GAMI API Specification</td>
<td><a href="https://github.com/01org/RSA">https://github.com/01org/RSA</a></td>
</tr>
<tr>
<td>332869</td>
<td>Intel® Rack Scale Architecture Pod Manager API Specification</td>
<td>This document.</td>
</tr>
<tr>
<td>332870</td>
<td>Intel® Rack Scale Architecture Pod Manager Release Notes</td>
<td><a href="https://github.com/01org/RSA">https://github.com/01org/RSA</a></td>
</tr>
<tr>
<td>332871</td>
<td>Intel® Rack Scale Architecture Pod Manager User Guide</td>
<td><a href="https://github.com/01org/RSA">https://github.com/01org/RSA</a></td>
</tr>
<tr>
<td>332872</td>
<td>Intel® Rack Scale Architecture PSME Release Notes</td>
<td><a href="https://github.com/01org/RSA">https://github.com/01org/RSA</a></td>
</tr>
<tr>
<td>332873</td>
<td>Intel® Rack Scale Architecture PSME API Specification</td>
<td><a href="https://github.com/01org/RSA">https://github.com/01org/RSA</a></td>
</tr>
<tr>
<td>332875</td>
<td>Intel® Rack Scale Architecture RMM Installation Guide</td>
<td><a href="https://github.com/01org/RSA">https://github.com/01org/RSA</a></td>
</tr>
<tr>
<td>332876</td>
<td>Intel® Rack Scale Architecture RMM Release Notes</td>
<td><a href="https://github.com/01org/RSA">https://github.com/01org/RSA</a></td>
</tr>
<tr>
<td>332877</td>
<td>Intel® Rack Scale Architecture RMM API Specification</td>
<td><a href="https://github.com/01org/RSA">https://github.com/01org/RSA</a></td>
</tr>
<tr>
<td>332878</td>
<td>Intel® Rack Scale Architecture Storage Services API Specification</td>
<td><a href="https://github.com/01org/RSA">https://github.com/01org/RSA</a></td>
</tr>
<tr>
<td></td>
<td>Scalable Platforms Management</td>
<td><a href="http://dmtf.org/standards/wip">http://dmtf.org/standards/wip</a></td>
</tr>
<tr>
<td></td>
<td>Intel® Rack Scale Architecture System Manageability Architecture Specification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Intel® Rack Scale Architecture Platform Design Guide</td>
<td></td>
</tr>
</tbody>
</table>

Scalable Platforms Management
2   PODM API

2.1   PODM API structure and relations

The PODM REST API provides the REST-based interface that allows full management of pod including asset discovery, configuration, and composed node assembly.

2.1.1   PODM API physical resource hierarchy

Table 3   Resources and URI

<table>
<thead>
<tr>
<th>Resource</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Root</td>
<td>/rest/v1</td>
</tr>
<tr>
<td>PodCollection</td>
<td>/rest/v1/Pods</td>
</tr>
<tr>
<td>POD</td>
<td>/rest/v1/Pods/{podID}</td>
</tr>
<tr>
<td>Rack Collection</td>
<td>/rest/v1/Pods/{podID}/Racks</td>
</tr>
<tr>
<td>Rack</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}</td>
</tr>
<tr>
<td>Drawer Collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers</td>
</tr>
<tr>
<td>Drawer</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}</td>
</tr>
<tr>
<td>Compute Module Collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules</td>
</tr>
<tr>
<td>Compute Module</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleId}/Blades</td>
</tr>
<tr>
<td>Blade Collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleId}/Blades</td>
</tr>
<tr>
<td>Blade</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleId}/Blades/{bladeID}</td>
</tr>
<tr>
<td>Processor Collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleId}/Blades/{bladeID}/Processors</td>
</tr>
<tr>
<td>Processor</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleId}/Blades/{bladeID}/Processors/{processorID}</td>
</tr>
<tr>
<td>Resource</td>
<td>URI</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Memory collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleID}/Blades/{bladeID}/Memory</td>
</tr>
<tr>
<td>Memory</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleID}/Blades/{bladeID}/Memory/{memoryID}</td>
</tr>
<tr>
<td>Storage controller collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleID}/Blades/{bladeID}/StorageControllers</td>
</tr>
<tr>
<td>Storage controller</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleID}/Blades/{bladeID}/StorageControllers/{controllerID}</td>
</tr>
<tr>
<td>Drive collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleID}/Blades/{bladeID}/StorageControllers/{controllerID}/Drives</td>
</tr>
<tr>
<td>Drive</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleID}/Blades/{bladeID}/StorageControllers/{controllerID}/Drives/{driveID}</td>
</tr>
<tr>
<td>Manager Collection</td>
<td>/rest/v1/Managers</td>
</tr>
<tr>
<td>Manager</td>
<td>/rest/v1/Managers/{managerID}</td>
</tr>
<tr>
<td>Network service</td>
<td>/rest/v1/Managers/{managerID}/NetworkService</td>
</tr>
<tr>
<td>Network interface collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/ComputeModules/{moduleID}/Blades/{bladeID}/EthernetInterfaces</td>
</tr>
<tr>
<td>Network interface</td>
<td>/rest/v1/Managers/{managerID}/EthernetInterfaces</td>
</tr>
<tr>
<td>Fabric Module Collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/FabricModules</td>
</tr>
<tr>
<td>Fabric Module</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/FabricModules/{moduleID}</td>
</tr>
<tr>
<td>Fabric Switch Collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/FabricModules/{moduleID}/Switches</td>
</tr>
<tr>
<td>Fabric Switch</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/FabricModules/{moduleID}/Switches/{switchID}</td>
</tr>
<tr>
<td>Fabric Switch Port Collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/FabricModules/{moduleID}/Switches/{switchID}/Ports</td>
</tr>
<tr>
<td>Fabric Switch Port</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/FabricModules/{moduleID}/Switches/{switchID}/Ports/{portID}</td>
</tr>
<tr>
<td>VLAN network interface collection</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/FabricModules/{moduleID}/Switches/{switchID}/Ports/{portID}/VLANs</td>
</tr>
<tr>
<td>VLAN network interface</td>
<td>/rest/v1/Pods/{podID}/Racks/{rackID}/Drawers/{drawerID}/FabricModules/{moduleID}/Blades/{bladeID}/EthernetInterfaces/{nicID}/VLANs</td>
</tr>
<tr>
<td>Storage Service Collection</td>
<td>/rest/v1/Services</td>
</tr>
<tr>
<td>Storage Service</td>
<td>/rest/v1/Services/{serviceID}</td>
</tr>
<tr>
<td>Remote Target Collection</td>
<td>/rest/v1/Services/{serviceID}/Targets</td>
</tr>
<tr>
<td>Remote target</td>
<td>/rest/v1/Services/{serviceID}/Targets/{targetID}</td>
</tr>
<tr>
<td>Logical Drive Collection</td>
<td>/rest/v1/Services/{serviceID}/LogicalDrives</td>
</tr>
<tr>
<td>Logical Drive</td>
<td>/rest/v1/Services/{serviceID}/LogicalDrives/{driveID}</td>
</tr>
<tr>
<td>Physical Drive Collection</td>
<td>/rest/v1/Services/{serviceID}/Drives</td>
</tr>
<tr>
<td>Physical Drive</td>
<td>/rest/v1/Services/{serviceID}/Drives/{driveID}</td>
</tr>
<tr>
<td>Composed Node Collection</td>
<td>/rest/v1/Systems</td>
</tr>
<tr>
<td>Composed Node</td>
<td>/rest/v1/Systems/{nodeID}</td>
</tr>
</tbody>
</table>
This chapter describes all error codes that may be returned by the REST calls implemented in the PODM REST API of the Intel® Rack Scale Architecture software v1.1 release.

3.1 API error response

In the case of an error, PODM REST API responds with an HTTP status code, as defined by the HTTP 1.1 specification, and constrained by additional requirements defined in this specification.

HTTP response status codes alone often do not provide enough information to determine the cause of an error. The PODM REST API returns extended error information as a JSON object with a single property named "error". The value of this property shall be the JSON object with the properties listed in Table 4.

Table 4 API error response

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>code</td>
<td>String indicating a specific error or message (not to be confused with the HTTP status code).</td>
</tr>
<tr>
<td>message</td>
<td>A human readable error message indicating the semantics associated with the error.</td>
</tr>
<tr>
<td>target</td>
<td>An optional string defining the target of the particular error.</td>
</tr>
<tr>
<td>details</td>
<td>An optional array of JSON objects with code, message, target, severity, and resolution properties, providing more detailed information about the error.</td>
</tr>
<tr>
<td>message.parameters</td>
<td>An optional array of strings representing the substitution parameter values for the message. The Severity attribute is an annotation specified in the DMTF namespace and shall be prefixed with the alias “message”.</td>
</tr>
<tr>
<td>message.severity</td>
<td>An optional string representing the severity of the error. The Severity attribute is an annotation specified in the DMTF namespace and shall be prefixed with the alias “message”.</td>
</tr>
<tr>
<td>message.resolution</td>
<td>An optional string describing recommended action(s) to take to resolve the error. The Resolution attribute is an annotation specified in the DMTF namespace and shall be prefixed with the alias “message”.</td>
</tr>
</tbody>
</table>

3.1.1 Example error JSON object

```
{
    "error": {
        "code": "Base.1.0.InvalidEndpoint",
        "message": "Invalid endpoint in /rest/v1 namespace",
        "details": [
        ]
    }
}
```

3.2 API error codes

In general, if an error is not described in any of the following tables, it is to be mapped into HTTP 500 Internal Error code.
3.2.1 General error codes

Table 5 General error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
<th>HTTP status code</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnknownException</td>
<td>A generic error message, given when an unexpected condition was encountered and no more specific message is suitable.</td>
<td>500 Internal Error</td>
</tr>
</tbody>
</table>

3.2.2 Request error codes

Table 6 Request error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
<th>HTTP status code</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidEndpoint</td>
<td>Invalid endpoint in /rest/v1 namespace.</td>
<td>404 Not Found</td>
</tr>
<tr>
<td>InvalidHttpMethod</td>
<td>Invalid HTTP request method.</td>
<td>405 Method Not Allowed</td>
</tr>
<tr>
<td>MalformedUri</td>
<td>Malformed URI.</td>
<td>400 Bad Request</td>
</tr>
<tr>
<td>BadAcceptHeader</td>
<td>Bad Accept field in request header.</td>
<td>406 Not Acceptable</td>
</tr>
<tr>
<td>InvalidPayload</td>
<td>Request payload is invalid or missing.</td>
<td>400 Bad Request</td>
</tr>
</tbody>
</table>

3.2.3 Node assembly error codes

Table 7 Node assembly error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
<th>HTTP status code</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotSufficientResources</td>
<td>Not sufficient resources for creating machine.</td>
<td>500 Internal Server Error</td>
</tr>
<tr>
<td>UnsupportedCreationRequest</td>
<td>Unable to create machine due to error in machine template or bad creation request.</td>
<td>400 BadRequest</td>
</tr>
<tr>
<td>ResourcesStateMismatch</td>
<td>Conflict during allocation.</td>
<td>409 Conflict</td>
</tr>
</tbody>
</table>

3.2.4 Node action error codes

Table 8 Node disassembly error codes

<table>
<thead>
<tr>
<th>Error code</th>
<th>Description</th>
<th>HTTP status code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ComposedNodeActionException</td>
<td>Unable to perform action on actual state of the composed node.</td>
<td>409 Conflict</td>
</tr>
</tbody>
</table>

Each resource available on REST API has its own list of properties that describe it. Properties can be mandatory, and in such case, they must be present in the resulting JSON. When a property is not mandatory, it may not be present in the JSON.
4  PODM REST API Definition

4.1  Pod Manager service root

This section provides resource information about the entry point of the Pod Manager service root. Table 9 lists Pod Manager service root attributes.

Table 9  Pod Manager service root attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>String</td>
<td>Yes</td>
<td>Provides an ID of this resource.</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Yes</td>
<td>Name of service root.</td>
</tr>
<tr>
<td>Modified</td>
<td>String</td>
<td>Yes</td>
<td>The date-time stamp that the object was last modified.</td>
</tr>
<tr>
<td>Time</td>
<td>String</td>
<td>No</td>
<td>The current time as reported by the owner of this SPMA Service.</td>
</tr>
<tr>
<td>ServiceVersion</td>
<td>String</td>
<td>Yes</td>
<td>Version of Redfish service in majorversion.minorversion.errata format.</td>
</tr>
<tr>
<td>UUID</td>
<td>String</td>
<td>Yes</td>
<td>Unique identifier for a service instance.</td>
</tr>
</tbody>
</table>

4.1.1  Metadata

4.1.2  JSON serialization

```json
{
   "@odata.context": string,
   "@odata.id": string,
   "@odata.type": string,
   "Id": string,
   "Name": string,
   "Modified": string,
   "Time": string,
   "ServiceVersion": string,
   "UUID": string,
   "Links": {
      "Chassis": {
         "@odata.id": string
      },
      "Services": {
         "@odata.id": string
      },
      "Systems": {
         "@odata.id": string
      },
      "Managers": {
         "@odata.id": string
      }
   }
}
```
4.1.3  Operations

4.1.3.1  GET

Request:

GET /rest/v1
Content-Type: application/json

Response:

```
{
    "@odata.context": "/rest/v1/$metadata#RSAPODMServiceRoot",
    "@odata.id": "/rest/v1",
    "@odata.type": "#RSAPODMServiceRoot.1.0.0.RSAPODMServiceRoot",
    "Id": "RootService",
    "Name": "RSA POD Manager Service Root",
    "Modified": "2015-02-20T14:44:00+00:00",
    "Time": "2015-03-04T13:45:04+00:00",
    "ServiceVersion": "1.0.0",
    "UUID": "00000000-0000-0000-0000-000000000000",
    "Links": {
        "Chassis": {
            "@odata.id": "/rest/v1/Pods"
        },
        "Services": {
            "@odata.id": "/rest/v1/Services"
        },
        "Systems": {
            "@odata.id": "/rest/v1/Systems"
        },
        "Managers": {
            "@odata.id": "/rest/v1/Managers"
        }
    }
}
```

4.1.3.2 PUT

Operation is not allowed on this resource.

4.1.3.3 PATCH

Operation is not allowed on this resource.

4.1.3.4 POST

Operation is not allowed on this resource.

4.1.3.5 DELETE

Operation is not allowed on this resource.

4.2  Pod collection

This section provides resource information about the collection of all pods available. Table 10 lists pod collection attributes.
Table 10  Pod collection attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Yes</td>
<td>Name of collection.</td>
</tr>
<tr>
<td>Modified</td>
<td>String</td>
<td>Yes</td>
<td>The date-time stamp that the object was last modified.</td>
</tr>
</tbody>
</table>

**4.2.1 Metadata**

**4.2.2 JSON serialization**

```json
{
    "@odata.context": string,
    "@odata.id": string,
    "@odata.type": string,
    "Name": string,
    "Modified": string,
    "Links": {
        "Members@odata.count": Number,
        "Members": [
            {
                "@odata.id": string
            }
        ]
    }
}
```

**4.2.3 Operations**

**4.2.3.1 GET**

Request:

```
GET /rest/v1/Pods
Content-Type: application/json
```

Response:

```json
{
    "@odata.context": "/rest/v1/$metadata#RSAPods",
    "@odata.id": "/rest/v1/Pods",
    "@odata.type": "#RSAPod.1.0.0.RSAPodCollection",
    "Name": "RSA POD Collection",
    "Modified": "2015-02-23T14:44:00+00:00",
    "Links": {
        "Members@odata.count": 1,
        "Members": [
            {
                "@odata.id": "/rest/v1/Pods/1"
            }
        ]
    }
}
```

**4.2.3.2 PUT**

Operation is not allowed on this resource.
4.2.3.3 **PATCH**

Operation is not allowed on this resource.

4.2.3.4 **POST**

Operation is not allowed on this resource.

4.2.3.5 **DELETE**

Operation is not allowed on this resource.

4.3 **Pod**

This section provides detailed resource information about single pods. Table 11 lists pod attributes.

<table>
<thead>
<tr>
<th>Table 11</th>
<th>Pod attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Pod</td>
</tr>
<tr>
<td>Type URI</td>
<td>/rest/v1/Pods/{PodId}</td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>Id</td>
<td>String</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
</tr>
<tr>
<td>Modified</td>
<td>String</td>
</tr>
<tr>
<td>ChassisType</td>
<td>String (enum)</td>
</tr>
<tr>
<td>Status</td>
<td>Object</td>
</tr>
<tr>
<td></td>
<td>Name</td>
</tr>
<tr>
<td>State</td>
<td>String</td>
</tr>
<tr>
<td>Health</td>
<td>String, Null</td>
</tr>
<tr>
<td>HealthRollup</td>
<td>String, Null</td>
</tr>
<tr>
<td>EnumStatus</td>
<td>String</td>
</tr>
<tr>
<td>Location</td>
<td>Object</td>
</tr>
<tr>
<td>Pod</td>
<td>Number</td>
</tr>
<tr>
<td>Oem</td>
<td>Object, Null</td>
</tr>
</tbody>
</table>

4.3.1 **Metadata**

4.3.2 **JSON serialization**

```json
{
    "@odata.context": string,
    "@odata.id": string,
    "@odata.type": string,
    "Id": string,
    "Name": string,
    "Modified": string,
    "ChassisType": enum ("Pod"),
    "Status": {
        "State": enum ("Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent")
    }
```

4.3.3 Operations

4.3.3.1 GET

Request:

GET /rest/v1/Pods/1
Content-Type: application/json

Response:

```json
{
    "@odata.context": "http://schemas.dmtf.org/industrystandard/RSAPod-v1.0.0/RSAPod/$entity",
    "@odata.id": "http://schemas.dmtf.org/industrystandard/RSAPod-v1.0.0/RSAPod/1",
    "@odata.type": "#RSAPod.1.0.0.RSAPod",
    "Id": "1",
    "Name": "RSA POD",
    "Modified": "2015-02-20T14:44:00+00:00",
    "ChassisType": "Pod",
    "Status": {
        "State": < { "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent" } >,
        "Health": < { "OK", "Warning", "Critical" } >,
        "HealthRollup": < { "OK", "Warning", "Critical" } >,
    },
    "EnumStatus": "Enumerated",
    "Location": { "Pod": 1 },
    "Links": {
        "Racks": { "@odata.id": "http://schemas.dmtf.org/industrystandard/RSAPod-v1.0.0/RSAPod/1/Racks"
    }
}
```
### 4.3.2 PUT
Operation is not allowed on this resource.

### 4.3.3 PATCH
Operation is not allowed on this resource.

### 4.3.4 POST
Operation is not allowed on this resource.

### 4.3.5 DELETE
Operation is not allowed on this resource.

## 4.4 Rack collection

This section provides resource information about the collection of all available racks. Table 12 lists rack collection attributes.

### Table 12 Rack collection attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Yes</td>
<td>Name of collection.</td>
</tr>
<tr>
<td>Modified</td>
<td>String</td>
<td>Yes</td>
<td>The date-time stamp that the object was last modified</td>
</tr>
</tbody>
</table>

### 4.4.1 Metadata

### 4.4.2 JSON serialization

```json
{
    "@odata.context": string,
    "@odata.id": string,
    "@odata.type": string,
    "Name": string,
    "Modified": string,
    "Links": {
        "Members@odata.count": Number,
        "Members": [
            {
                "@odata.id": string
            }
        ]
    }
}
```
4.4.3 Operations

4.4.3.1 GET

Request:

GET /rest/v1/Pods/1/Racks
Content-Type: application/json

Response:

```
{
   "@odata.context": "/rest/v1/$metadata#RSAPods/Links/Members/l/Racks",
   "@odata.id": "/rest/v1/Pods/1/Racks",
   "@odata.type": "#RSARack.1.0.0.RSARackCollection",
   "Name": "RSA Rack Collection",
   "Modified": "2015-02-23T14:44:00+00:00",
   "Links": {
      "Members@odata.count": 1,
      "Members": [
         {
            "@odata.id": "/rest/v1/Pods/1/Racks/1"
         }
      ]
   }
}
```

4.4.3.2 PUT

Operation is not allowed on this resource.

4.4.3.3 PATCH

Operation is not allowed on this resource.

4.4.3.4 POST

Operation is not allowed on this resource.

4.4.3.5 DELETE

Operation is not allowed on this resource.
### 4.5 Rack

This section provides detailed resource information about single racks, as shown in Table 13.

#### Table 13 Rack attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Rack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>URI</td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>Id</td>
<td>String</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
</tr>
<tr>
<td>Modified</td>
<td>String</td>
</tr>
<tr>
<td>ChassisType</td>
<td>String (enum)</td>
</tr>
<tr>
<td>Status</td>
<td>Object</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>EnumStatus</td>
<td>String</td>
</tr>
<tr>
<td>Location</td>
<td>Object</td>
</tr>
<tr>
<td></td>
<td>Number</td>
</tr>
<tr>
<td>RSARackAttributes</td>
<td>Object</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>AssetTag</td>
<td>String, Null</td>
</tr>
<tr>
<td>RackUUID</td>
<td>String, Null</td>
</tr>
<tr>
<td>GeoTag</td>
<td>String, Null</td>
</tr>
<tr>
<td>TrayPresent</td>
<td>Object</td>
</tr>
<tr>
<td>FRUInfo</td>
<td>Object, Null</td>
</tr>
<tr>
<td>SerialNumber</td>
<td>String, Null</td>
</tr>
<tr>
<td>Manufacturer</td>
<td>String, Null</td>
</tr>
<tr>
<td>ThermalZones</td>
<td>Array, Null</td>
</tr>
<tr>
<td>PowerZones</td>
<td>Array, Null</td>
</tr>
<tr>
<td>Oem</td>
<td>Object, Null</td>
</tr>
</tbody>
</table>

#### 4.5.1.1 TrayPresent attributes

#### Table 14 TrayPresent attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TraysNumber</td>
<td>Number</td>
<td>Yes</td>
<td>Number of trays installed in rack</td>
</tr>
</tbody>
</table>
### 4.5.1.2 Thermal zone attributes

Table 15 Thermal zone attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ThermalzoneId</td>
<td>Number</td>
<td>Yes</td>
<td>Number identification of thermal zone.</td>
</tr>
<tr>
<td>RackLocation</td>
<td>Object, Null</td>
<td>No</td>
<td>Location in rack in units.</td>
</tr>
<tr>
<td>PresentTemperatureD</td>
<td>Number, Null</td>
<td>No</td>
<td>Current temperature.</td>
</tr>
<tr>
<td>MaxFansSupported</td>
<td>Number, Null</td>
<td>No</td>
<td>Maximum number of fans supported in thermal zone.</td>
</tr>
<tr>
<td>NumberOfFansPresent</td>
<td>Number</td>
<td>Yes</td>
<td>Number of installed fans.</td>
</tr>
<tr>
<td>Policy</td>
<td>Enum, Null</td>
<td>No</td>
<td>Type of thermal policy</td>
</tr>
<tr>
<td>Fans</td>
<td>Array, Null</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>DesiredSpeedRPM</td>
<td>Number</td>
<td>Yes</td>
<td>Desired fan speed in rotations per minute.</td>
</tr>
<tr>
<td>PresentTachMeterReading</td>
<td>Number</td>
<td>No</td>
<td>Current fan speed in RPM.</td>
</tr>
<tr>
<td>HealthState</td>
<td>Number</td>
<td>Yes</td>
<td>Health status of fan.</td>
</tr>
<tr>
<td>AssetTag</td>
<td>String</td>
<td>No</td>
<td>Customer settable tag.</td>
</tr>
<tr>
<td>FRUInfo</td>
<td>Object</td>
<td>No</td>
<td>Object containing FRU information (i.e. serial number).</td>
</tr>
<tr>
<td>MeterState</td>
<td>Enum</td>
<td>No</td>
<td>State of thermal zone meter.</td>
</tr>
<tr>
<td>State</td>
<td>Enum</td>
<td>Yes</td>
<td>Thermal zone state.</td>
</tr>
</tbody>
</table>

### 4.5.1.3 Power zone attributes

Table 16 Power zone attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PowerzoneId</td>
<td>Number</td>
<td>Yes</td>
<td>Number identification of thermal zone.</td>
</tr>
<tr>
<td>RackLocation</td>
<td>Object, Null</td>
<td>No</td>
<td>Location in rack in units.</td>
</tr>
<tr>
<td>PresentPowerOutputW</td>
<td>String, Null</td>
<td>No</td>
<td>Current power zone power output in watts.</td>
</tr>
<tr>
<td>MaxPSUsSupported</td>
<td>Number, Null</td>
<td>No</td>
<td>Maximum number of PSUs supported in power zone.</td>
</tr>
<tr>
<td>NumberOfPSUsPresent</td>
<td>Number</td>
<td>Yes</td>
<td>Number of installed PSUs.</td>
</tr>
</tbody>
</table>
### 4.5.2 Metadata

### 4.5.3 JSON serialization

```json
{
    "@odata.context": string,
    "@odata.id": string,
    "@odata.type": string,
    "Id": string,
    "Name": string,
    "Modified": string,
    "ChassisType": enum ("Rack"),
    "Location": {
        "Pod": Number,
        "Rack": Number,
    },
    "Status": {
        "State": enum ("Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent")
        "Health": enum ("OK", "Warning", "Critical")
        "HealthRollup": enum ("OK", "Warning", "Critical")
    },
    "EnumStatus": enum ("Enumerated", "NotEnumerated")
    "RSARackAttributes": {
        "RMMPresent": boolean,
        "RackSupportsDisaggregatedPowerCooling": boolean,
        "RackUUID": string,
        "AssetTag": string,
        "GeoTag": string,
        "TrayPresent": {
            "TraysNumber": Number,
            "Trays": [
                {
                    "TrayRUID": Number
                }
            ]
        },
    }"FRUInfo": {
        "SerialNumber": string,
        "Manufacturer": string
    },
    "ThermalZones": [
        {
            "ThermalzoneId": Number
        }
    ]
}
```
"RackLocation": { "ULocation": Number, "UHeight": Number, "Unit": enum ("RU", "OU") }, "PresentTemperatureDegC": Number, "MaxFansSupported": Number, "NumberOfFansPresent": Number, "Pwm": Number, "Policy": enum ("normal", "aggressive", "none") } "Fans": [ { "DesiredSpeedRPM": Number, "PresentTachMeterReading": Number, "HealthState": Number, "AssetTag": string, "FRUInfo": { "SerialNumber": string }}, "MeterState": enum ("LowerNonCritical", "UpperNonCritical", "LowerCritical", "UpperCritical", "Normal"), "State": enum ("Unknown", "Enabled", "Disabled", "NA") } ] }, "PowerZones": [ { "PowerzoneId": Number, "RackLocation": { "ULocation": Number, "UHeight": Number, "Units": enum ("RU", "OU") }, "PresentPowerOutputWatt": string, "MaxPSUsSupported": Number, "NumberOfPSUsPresent": Number, "PSUs": [ { "PsuId": Number, "PresentPowerOutputWatt": string, } ] } ]}
4.5.4 Operations

4.5.4.1 GET

Request:

GET /rest/v1/Pods/1/Racks/1
Content-Type: application/json

Response:

```json
{
   "@odata.context":
   "/rest/v1/$metadata#RSAPods/Links/Members/1/Links/Racks/Links/Members/$entity",
   "@odata.id": "/rest/v1/Pods/1/Racks/1",
   "@odata.type": "#RSA201.0.0.RSARack",
   "Id": "1",
   "Name": "RSA Rack",
   "Modified": "2015-02-20T14:44:00+00:00",
   "ChassisType": "Rack",
   "Location": {
      "Pod": 1,
      "Rack": 1
   },
   "Status": {
      "State": < { "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent" } >,
      "Health": < { "OK", "Warning", "Critical" } >,
      "HealthRollup": < { "OK", "Warning", "Critical" } >
   }
}
```
"EnumStatus": "Enumerated",
"RSARackAttributes": {
  "RMPPresent": true,
  "RackSupportsDisaggregatedPowerCooling": false,
  "RackUUID": "00000000-0000-0000-0000-000000000000",
  "AssetTag": "CustomerWritableThing",
  "GeoTag": "54.348103, 18.645172",
  "TrayPresent": {
    "TraysNumber": 1,
    "Trays": [
      {
        "TrayRUID": 1
      }
    ],
  },
  "FRUInfo": {
    "SerialNumber": "12345678",
    "Manufacturer": "Foo Corp.",
  },
  "ThermalZones": [
    {
      "ThermalzoneId": 1,
      "RackLocation": {
        "ULocation": 2,
        "UHeight": 6,
        "Unit": < { "RU", "OU" } >
      },
      "PresentTemperatureDegC": 22,
      "MaxFansSupported": 4,
      "NumberOfFansPresent": 2,
      "Pwm": 80,
      "Policy": < { "normal", "aggressive", "none" } >
    },
    {
      "DesiredSpeedRPM": 1200,
      "PresentTachMeterReading": 1200,
      "HealthState": 0,
      "AssetTag": "CustomerWritableThing",
      "FRUInfo": {
        "SerialNumber": "12345678"
      },
      "MeterState": < {
        "LowerNonCritical", "UpperNonCritical", "LowerCritical", "UpperCritical", "Normal" } >,
      "State": < { "Unknown", "Enabled", "Disabled", "NA" } >
    }
  ]
}
"PresentTachMeterReading": 1200,
"HealthState": 0,
"AssetTag": "CustomerWritableThingy",
"FRUInfo": {
  "SerialNumber": "12345678"
},
"FRUInfo": {
  "SerialNumber": "87654321"
},
"Normal" } >,
"State": < { "Unknown",
"Enabled", "Disabled", "NA" } >
],
"PowerZones": [ ]
]
]"presentPowerOutputWatt": 1500",
"Disabled", "NA", "Offline" } >
"
"12345678"

"PresentPowerOutputWatt": 4500",
"MaxPSUsSupported": 4,
"NumberOfPSUsPresent": 2,
"PSUs": [
  {
    "PsuId": 1,
    "PresentPowerOutputWatt": 1500",
    "State": < { "Enabled",
    "Disabled", "NA", "Offline" } >
    "FRUInfo": {
      "SerialNumber": "12345678"
    },
    },
    {
      "PsuId": 2,
      "PresentPowerOutputWatt": 1500",
      "State": < { "Enabled",
      "Disabled", "NA", "Offline" } >
      "FRUInfo": {
        "SerialNumber": "87654321"
      },
      }
    ]}
4.5.4.2 **PUT**  
Operation is not allowed on this resource.

4.5.4.3 **PATCH**  
Operation is not allowed on this resource.

4.5.4.4 **POST**  
Operation is not allowed on this resource.

4.5.4.5 **DELETE**  
Operation is not allowed on this resource.

### 4.6 Storage service collection

This section provides resource information about the collection of available storage services. Table 17 lists the attributes.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Yes</td>
<td>Name of service collection.</td>
</tr>
<tr>
<td>Modified</td>
<td>String</td>
<td>Yes</td>
<td>The date-time stamp that the object was last modified.</td>
</tr>
</tbody>
</table>

#### 4.6.1 Metadata

#### 4.6.2 JSON serialization

```json
{
    "@odata.context": "string",
    "@odata.id": "string",
    "@odata.type": "string",
    "Name": "string",
    "Modified": "string",
    "Links": {
        "@odata.id": "string",
        "@odata.type": "string",
        "$httpAnchor": "string",
        "Members": ["string"]
    }
}
```
4.6.3 **Operations**

4.6.3.1 **GET**

Request:

GET /rest/v1/Services
Content-Type: application/json

Response:

```json
{
    "@odata.context": "/rest/v1/$metadata#RSAStorageServices",
    "@odata.id": "/rest/v1/Services",
    "@odata.type": "#RSAStorageService.1.0.0.RSAStorageServiceCollection",
    "Name": "Storage Services Collection",
    "Modified": "2015-02-23T14:44:00+00:00",
    "Links": {
        "Members@odata.count":1,
        "Members": [
            {
                "@odata.id": "/rest/v1/Services/1"
            }
        ]
    }
}
```

4.6.3.2 **PUT**

Operation is not allowed on this resource.

4.6.3.3 **PATCH**

Operation is not allowed on this resource.

4.6.3.4 **POST**

Operation is not allowed on this resource.

4.6.3.5 **DELETE**

Operation is not allowed on this resource.

4.7 **Storage service**

Refer to the *Intel® Rack Scale Architecture Storage Services API Specification.*

4.8 **Remote target collection**

Refer to the *Intel® Rack Scale Architecture Storage Services API Specification.*
4.9 **Logical drive collection**
Refer to the *Intel® Rack Scale Architecture Storage Services API Specification*.

4.10 **Physical drive collection**
Refer to the *Intel® Rack Scale Architecture Storage Services API Specification*.

4.11 **Drawer collection**
Refer to the *Intel® Rack Scale Architecture PSME API Specification*.

4.12 **Drawer**
Refer to the *Intel® Rack Scale Architecture PSME API Specification*.

4.13 **Compute module collection**
Refer to the *Intel® Rack Scale Architecture PSME API Specification*.

4.14 **Compute module**
Refer to the *Intel® Rack Scale Architecture PSME API Specification*.

4.15 **Blade collection**
Refer to the *Intel® Rack Scale Architecture PSME API Specification*.

4.16 **Blade**
Refer to the *Intel® Rack Scale Architecture PSME API Specification*.

4.17 **Processor collection**
Refer to the *Intel® Rack Scale Architecture PSME API Specification*.

4.18 **Processor**
Refer to the *Intel® Rack Scale Architecture PSME API Specification*.

4.19 **Manager collection**
Refer to the *Intel® Rack Scale Architecture PSME API Specification*.

4.20 **Manager**
This section provides detailed resource information about managers identified by (managerID). Table 18 lists the attributes.
### Table 18 Manager attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Id</td>
<td>String</td>
<td>Yes</td>
<td>Provides an ID of this resource.</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Yes</td>
<td>Name of resource.</td>
</tr>
<tr>
<td>Modified</td>
<td>String</td>
<td>Yes</td>
<td>The date-time stamp that the object was last modified.</td>
</tr>
<tr>
<td>ManagerType</td>
<td>String (enum)</td>
<td>Yes</td>
<td>Type of manager.</td>
</tr>
<tr>
<td>Model</td>
<td>String, Null</td>
<td>No</td>
<td>Manager model.</td>
</tr>
<tr>
<td>Status</td>
<td>Object</td>
<td>Yes</td>
<td>Resource status represented by following object:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- State</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Health</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- HealthRollup</td>
</tr>
<tr>
<td>Firmware</td>
<td>Object</td>
<td>No</td>
<td>Resource status represented by following object:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Current</td>
</tr>
<tr>
<td>Oem</td>
<td>Object, Null</td>
<td>No</td>
<td>OEM defined object</td>
</tr>
<tr>
<td>GraphicalConsole</td>
<td>Object</td>
<td>Yes</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Enabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- MaxConcurrent Sessions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- ConnectTypesSupported</td>
</tr>
<tr>
<td>SerialConsole</td>
<td>Object</td>
<td>Yes</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Enabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- MaxConcurrent Sessions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- ConnectTypesSupported</td>
</tr>
<tr>
<td>CommandShell</td>
<td>Object</td>
<td>Yes</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Enabled</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- MaxConcurrent Sessions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- ConnectTypesSupported</td>
</tr>
</tbody>
</table>

### 4.20.1 Metadata

TBD

### 4.20.2 JSON serialization

```json
{
    "@odata.context": string,
    "@odata.id": string,
}
```
"@odata.type": string,
"Id": string,
"Name": string,
"Modified": string,
"ManagerType":
  enum ("ManagementController", "EnclosureManager", "BMC")
"Model": string,
"Status": {
  "State":
    enum ("Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent")
  "Health":
    enum ("OK", "Warning", "Critical")
},
"Oem": object,
"GraphicalConsole": {
  "Enabled": boolean,
  "MaxConcurrentSessions": Number,
  "ConnectTypesSupported": enum ("KVMIP", "Oem")
},
"SerialConsole": {
  "Enabled": boolean,
  "MaxConcurrentSessions": Number,
  "ConnectTypesSupported": enum ("SSH", "Telnet", "IPMI", "Oem")
},
"CommandShell": {
  "Enabled": boolean,
  "MaxConcurrentSessions": Number,
  "ConnectTypesSupported": enum ("SSH", "Telnet", "IPMI", "Oem")
},
"Firmware": {
  "Current": {
    "VersionString": string
  }
},
"Links": {
  "ManagerForChassis": [ {
    "@odata.id": string
  } ],
  "ManagerForComputeModules": [ {
    "@odata.id": string
  } ],
  "ManagerForFabricModules": [ {
    "@odata.id": string
  } ],
  "ManagerForBlades": [ {
    "@odata.id": string
  } ]
}
"ManagerForSwitches": [  
  {  
    "@odata.id": "string"  
  }  
],
"ManagerForComposedNodes": [  
  {  
    "@odata.id": "string"  
  }  
],
"NetworkService": {  
  "@odata.id": "string"  
},
"SimpleNetwork": {  
  "@odata.id": "string"  
},
"Oem": object
}

4.20.3 Operation

4.20.3.1 GET

Request:
GET /rest/v1/Managers/{managerID}
Content-Type: application/json

Response:

{
  "@odata.context": "/rest/v1/$metadata#RSAManagers/Links/Members/Sentity",
  "@odata.id": "/rest/v1/Managers/1",
  "@odata.type": "#RSAManager.1.0.0.RSAManager",
  "Id": "1",
  "Name": "Drawer Manager",
  "Modified": "2015-02-23T14:44:22+00:00",
  "ManagerType": < { "ManagementController", "EnclosureManager", "BMC" } >
},
  "Model": "Avoton CPP",
  "Status": {
    "State": < { "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent" } >,
    "Health": < { "OK", "Warning", "Critical" } >
  },
  "GraphicalConsole": {
    "Enabled": false,
  },
  "SerialConsole": {
    "Enabled": false,
  },
  "CommandShell": {
    "Enabled": false,
  },
  "Firmware": {

4.20.3.2 PUT
Operation is not allowed on this resource.

4.20.3.3 PATCH
Operation is not allowed on this resource.

4.20.3.4 POST
Operation is not allowed on this resource.

4.20.3.5 DELETE
Operation is not allowed on this resource.

4.21 Storage manager
Refer to the Intel® Rack Scale Architecture Storage Services API Specification.

4.22 Composed node collection
This section provides resource information about composed collection of all logical nodes. Table 19 lists the attributes.
### Table 19 Composed node collection attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>String</td>
<td>Yes</td>
<td>Name of collection.</td>
</tr>
<tr>
<td>Modified</td>
<td>String</td>
<td>Yes</td>
<td>The date-time stamp that the object was last modified.</td>
</tr>
</tbody>
</table>

#### 4.22.1 Metadata

TBD

#### 4.22.2 JSON serialization

```json
{
    "@odata.context": string,
    "@odata.id": string,
    "@odata.type": string,
    "Name": string,
    "Modified": string,
    "Links": {
        "Members@odata.count": Number,
        "Members": [
            {
                "@odata.id": string
            }
        ]
    }
}
```

#### 4.22.3 Operation

##### 4.22.3.1 GET

Request:

```
GET /rest/v1/Systems
Content-Type: application/json
```

Response:

```json
{
    "@odata.context": "/rest/v1/$metadata#RSAComposedNodes",
    "@odata.id": "/rest/v1/Systems",
    "@odata.type": #RSAComposedNode.1.0.0.RSAComposedNodeCollection",
    "Name": "Composed Nodes Collection",
    "Modified": "2015-02-23T14:44:00+00:00",
    "Links": {
        "Members@odata.count": 1,
        "Members": [
            {
                "@odata.id": "/rest/v1/Systems/1"
            }
        ]
    }
}
```
4.22.3.2 PUT
Operation is not allowed on this resource.

4.22.3.3 PATCH
Operation is not allowed on this resource.

4.22.3.4 POST
POST request is the first step to assemble the logical server node required to allocate physical resources needed for node composition. After a successful allocation, the link to the newly created logical node is returned. User can send a DELETE request to remove node and return components back to the pool, or POST with “Assembly” action without a parameter, to physically assemble node and power it on.

Both requests are available to many users at the same time (parallel allocation and assembly).

Every property in JSON used for this method is OPTIONAL. It’s possible to send POST command with an empty JSON ({}); in such case the Pod Manager will allocate any available resources to the requested node.

**Note:** Currently a user can request assembly of a single node with a single request. Node components—CPU, memory, local storage, network interface—must be located on a single physical blade. Remote storage can be located anywhere in the pod.

Request:

```plaintext
POST /rest/v1/Systems
Content-Type: application/json
{
   Systems: [{
      "Name": String,
      "Description": String,
      "Processors": [{
         "Quantity": Number,
         "Model": enum <"E3", "E5", "E7", "X3", "X5", "X7", "I3", "I5", "I7">,
         "TotalCores": Number,
         "MaxSpeedMHz": Number,
         "InstructionSet": enum <"x86", "x86-64">,
         "InstructionSetExtensions": [ enum <"x86:SSE2", "x86:VT-d", "x86:VT-x"> ],
         "Location": {
            "Pod": Number,
            "Rack": Number,
            "Drawer": Number,
            "Module": Number,
            "Blade": Number
         }
      }],
      "Memory": [{
         "SizeGB": Number,
         "Type": enum <"DDR-3", "DDR-4", "SXP">,
         "SpeedMHz": Number,
         "Location": {
            "Pod": Number,
            "Rack": Number,
            "Drawer": Number
         }
      }]
   }],
   "Location": {
      "Pod": Number,
      "Rack": Number,
      "Drawer": Number,
      "Module": Number,
      "Blade": Number
   }
}
```
"Module": Number,
"Blade": Number
}

"RemoteDrives": [{
  "CapacityGB": Number,
  "AddressType": enum <"iSCSI">,
  "Address": String,
  "Storage": enum <"Cold", "Hot">,
  "Master": {
    "Type": <"Snapshot", "Clone">,
    "AddressType": enum <"iSCSI">,
    "Address": String
  },
  "BootOrder": Number,
  "Location": Object
}],

"LocalDrives": [{
  "CapacityGB": Number,
  "Type": enum <"HDD", "SSD">,
  "BootOrder": Number,
  "Location": {
    "Pod": Number,
    "Rack": Number,
    "Drawer": Number,
    "Module": Number,
    "Blade": Number
  }
}],

"NetworkInterfaces": [{
  "SpeedMbps": number,
  "VLANs": [{
    "VlanId": Number,
    "Tagged": Boolean
  }],
  "Location": {
    "Pod": Number,
    "Rack": Number,
    "Drawer": Number,
    "Module": Number,
    "Blade": Number
  }
}],

"Oem": {}]

Response:

HTTP/1.1 201 Created
Location: http://<IP>:<Port>/rest/v1/Systems/2

4.22.3.5 DELETE

Operation is not allowed on this resource.
## 4.23 Composed node

This section provides detailed resource information about a composed logical node identified by `{systemID}`. Table 20 lists the attributes.

### Table 20 Composed node attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Mandatory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Composed node</td>
<td>Type URI</td>
<td></td>
<td>/rest/v1/Systems/{systemID}</td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
<td>Mandatory</td>
<td>Description</td>
</tr>
<tr>
<td>Id</td>
<td>String</td>
<td>Yes</td>
<td>Provides an ID of this resource.</td>
</tr>
<tr>
<td>Name</td>
<td>String</td>
<td>Yes</td>
<td>Name of composed node.</td>
</tr>
<tr>
<td>Modified</td>
<td>String</td>
<td>Yes</td>
<td>The date-time stamp that the object was last modified.</td>
</tr>
<tr>
<td>Description</td>
<td>String</td>
<td>No</td>
<td>User provided node description.</td>
</tr>
<tr>
<td>SystemType</td>
<td>String (enum)</td>
<td>Yes</td>
<td>Type of computer system (physical, virtual).</td>
</tr>
<tr>
<td>Processors</td>
<td>Object</td>
<td>Yes</td>
<td>Array of references to Processor resources.</td>
</tr>
<tr>
<td>Memory</td>
<td>Object</td>
<td>Yes</td>
<td>Array of references to Memory resources.</td>
</tr>
<tr>
<td>Status</td>
<td>Object</td>
<td>Yes</td>
<td>Resource status represented by following object:</td>
</tr>
<tr>
<td>ComposedNodeState</td>
<td>String (enum)</td>
<td>Yes</td>
<td>Current state of assembly process for this node.</td>
</tr>
<tr>
<td>Oem</td>
<td>Object, Null</td>
<td>No</td>
<td>OEM defined object.</td>
</tr>
<tr>
<td>Processors</td>
<td>Array</td>
<td>No</td>
<td>Array of references to Processor resources.</td>
</tr>
<tr>
<td>Memory</td>
<td>Array</td>
<td>No</td>
<td>Array of references to Memory resources.</td>
</tr>
</tbody>
</table>

**Processors**

- **Count**: Number of CPUs.
- **ProcessorSockets**: Number of available CPU sockets.
- **ProcessorFamily**: Basic information about processor type.
- **Status**: See Status property below.

**Memory**

- **TotalSystemMemory GB**: Amount of installed memory in GB.
- **MemorySockets**: Number of available memory sockets.
- **Status**: See Status property below.

**Status**

- **State**: Known state of the resource.
- **Health**: Overall health state from the view of this resource.
- **HealthRollup**: Health state of this resource in the absence of its dependent resources.

**ComposedNodeState**

- Allocating: Allocating resources for node is in progress. Next state can be Allocated or Failed.
- Allocated: Node resources have been allocated, but assembly not started yet. After RSAComposedNode.Assemble action state will progress to Assembling.
- Assembling: Assembly process initiated, but not finished yet. When assembly is done, it will change into PoweredOn.
- PoweredOn: Node successfully assembled and powered on.
- PoweredOff: Node successfully assembled and powered off.
- Failed: Allocation or assembly process failed, or in runtime, one of composing components was removed or transitioned in error state.
RemoteDrives | Array | Yes | An array of references to the remote storage drives.
--- | --- | --- | ---
LocalDrives | Array | Yes | An array of references to the blade local storage drives.
EthernetInterfaces | Array | Yes | Array of links to Ethernet Interface collection associated with this Composed Node.
ManagedBy | Array | Yes | An array of references to Managers responsible for this Composed Node.
Actions | Object | Yes | Actions available for this node:
- Reset action with following values:
  - On – Turn the system on.
  - ForceOff – Turn the system off immediately (non-graceful shutdown).
  - GracefulRestart – Perform a graceful system shutdown, followed by a restart of the system.
  - ForceRestart – Perform an immediate (non-graceful) shutdown, followed by a restart of the system.
  - Nmi – Generate a Diagnostic Interrupt (usually an NMI on x86 systems) to cease normal operations, perform diagnostic actions and typically halt the system.
  - ForceOn – Turn the system on immediately.
  - PushPowerButton – Simulate the pressing of the physical power button on this system.
- Assemble: Doesn’t consume any parameters. Second step of creating a composed node (after POST on Systems collection). That action will assembly logical node – initiate ComposedNodeState change from Allocated state into Assembling state.

Figure 2 ComposedNodeState changes during assembly process

4.23.1 Metadata

TBD
4.23.2 JSON serialization

```json
{
    "@odata.context": string,
    "@odata.id": string,
    "@odata.type": string,
    "Id": string,
    "Name": string,
    "Modified": string,
    "Description": string,
    "SystemType": enum ("Physical"),
    "Status": {
        "State": enum ("Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent")
    },
    "Oem": object,
    "Processors": {
        "Count": Number,
        "ProcessorSockets": Number,
        "ProcessorFamily": enum ("E3", "E5", "E7", "X3", "X5", "X7", "I3", "I5", "I7"),
        "Status": {
            "State": enum ("Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent")
        }
    },
    "Memory": {
        "TotalSystemMemoryGB": Number,
        "MemorySockets": Number,
        "Status": {
            "State": enum ("Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent")
        }
    },
    "Links": {
        "Processors": [
            { "@odata.id": string }]
        },
        "Memory": [
            { "@odata.id": string }
        ]
```
4.23.3 Operations

4.23.3.1 GET

Request:

GET /rest/v1/Systems/{systemID}
Content-Type: application/json

Response:

```json
{
    "@odata.context": "/rest/v1/$metadata#RSAComposedNode/Links/Members/$entity",
    "@odata.id": "/rest/v1/Systems/1",
    "@odata.type": "#RSAComposedNodes.1.0.0.RSAComposedNode",
    "Id": "1",
    "Name": "Xeon Compute Module",
    "Modified": "2015-02-20T14:44:11+00:00",
    "Description": "RSA Node #1",
    "SystemType": "Physical",
    "Status": {
        "State": < { "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent" } >,
    },
    "Processors": {
        "Count": 2,
        "ProcessorSockets": 2,
        "ProcessorFamily": "E7",
        "Status": {
```
"State": < { "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent" } >,
"Health": < {"OK", "Warning", "Critical" } >,
"HealthRollup": < {"OK", "Warning", "Critical" } >
}
},
"Memory": {
"TotalSystemMemoryGB": 32.0,
"MemorySockets": 8,
"Status": {
"State": < { "Enabled", "Disabled", "Offline", "InTest", "Starting", "Absent" } >,
"Health": < {"OK", "Warning", "Critical" } >,
"HealthRollup": < {"OK", "Warning", "Critical" } >
}
},
"ComposedNodeState": "Allocated",
"Oem": {},
"Links": {
"Processors": [ {
"@odata.id": "/rest/v1/Pods/1/Racks/1/Drawers/1/ComputeModules/1/Blades/1/Processors/1"
}
],
"Memory": [ {
"@odata.id": "/rest/v1/Pods/1/Racks/1/Drawers/1/ComputeModules/1/Blades/1/Memory/1"
}
],
"RemoteDrives": [ {
"@odata.id": "/rest/v1/Services/1/Targets/1"
}
],
"LocalDrives": [ {
"@odata.id": "/rest/v1/Pods/1/Racks/1/Drawers/1/ComputeModules/1/Blades/1/StorageControllers/1/Drives/1"
}
],
"EthernetInterfaces": [ {
"@odata.id": "/rest/v1/Pods/1/Racks/1/Drawers/1/ComputeModules/1/Blades/1/EthernetInterfaces/1"
}
],
"ManagedBy": [ {
"@odata.id": "/rest/v1-Managers/2"
}
]},
“Oem”: {}  
}  
“Actions”: {}  

“#RSAComposedNode.Reset”: {}  
  “target”:  
“/rest/v1/Systems/1/Actions/RSAComposedNode.Reset”,  
  “ResetType@DMTF.AllowableValues”: [  
    “On”,  
    “ForceOff”,  
    “GracefulRestart”,  
    “ForceRestart”,  
    “Nmi”,  
    “ForceOn”,  
    “PushPowerButton”  
  ]  
},  
“#RSAComposedNode.Assemble”: {}  
  “target”:  
“/rest/v1/Systems/1/Actions/RSAComposedNode.Assemble”,  
}

4.23.3.2 PUT  
Operation is not allowed on this resource.

4.23.3.3 PATCH  
Operation is not allowed on this resource.

4.23.3.4 POST  
Request:  
POST /rest/v1/Systems/1/Actions/RSAComposedNode.Reset  
Content-Type: application/json  
{  
  “ResetType”: “On”  
}

Response:  
HTTP/1.1 204 No Content

4.23.3.5 DELETE  
Request:  
DELETE /rest/v1/Systems/1

Response:  
HTTP/1.1 204 No Content
PODM REST API Definition

4.24 **Network service**
Refer to the *Intel® Rack Scale Architecture PSME API Specification.*

4.25 **Network interface collection**
Refer to the *Intel® Rack Scale Architecture PSME API Specification.*

4.26 **Network interface**
Refer to the *Intel® Rack Scale Architecture PSME API Specification.*

4.27 **VLAN network interface collection**
Refer to the *Intel® Rack Scale Architecture PSME API Specification.*

4.28 **VLAN network interface**
Refer to the *Intel® Rack Scale Architecture PSME API Specification.*

§