Intel® Rack Scale Design (Intel® RSD) POD Manager (PODM) Representational State Transfer (REST)

API Specification
Software v2.4

April 2019

Revision 001

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express, implied, by estoppel, or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and noninfringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software, or service activation. Performance varies depending on system configuration. No computer system can be secure. Check with your system manufacturer or retailer or learn more at *www.intel.com*.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

This document contains information on products, services, and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications, and roadmaps.

The products and services described may contain defects or errors, known as errata, which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents that have an order number and are referenced in this document may be obtained by calling 1 800 548 4725 or by visiting *www.intel.com/design/literature.htm*.

Contents

Table of Contents

Figures

Tables

Revision History

Revision	Description	Date
001	Initial release for Intel® RSD software release v2.4	April 2019

§

1.0 Introduction

This document contains information about the Intel® Rack Scale Design (Intel® RSD) POD Manager (PODM) Representational State Transfer (REST) API, which was designed and implemented for Intel® Rack Scale Design Software v2.4.

1.1 Scope

The interface is based on the *Distributed Management Task Force's (DMTF) Redfish* Interface Specification v1.5.0*, and Redfish* schema #DSP8010 v2018.1; refer to Table 2. The exceptions are as follows:

- The interface for Manager resource is based on the *Distributed Management Task Force's (DMTF) Redfish* Interface Specification v1.5.0, and Redfish** #DSP8010 v2018.2.
- The Composed Node interface is an Intel® Rack Scale Design extension to Redfish.

For the location and titles of documents mentioned here, refer to Table 2.

1.2 Intended Audience

The intended audience for this document is designers and engineers working with the Intel® Rack Scale Design Software v2.4 release.

1.3 Conventions

The key words/phrases "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119 (refer to Table 2).

1.4 Notes and Symbol Conventions

- Symbol and note conventions are similar to typographical conventions used in CIMI specification
- Notation used in JavaScript Object Notation (JSON) serialization description:
  - Mandatory in italics indicate data types instead of literal Mandatory
  - Characters are appended to items to indicate cardinality:
  - – "?" (0 or 1)
  - – "*" (0 or more)
  - – "+" (1 or more)
  - Vertical bars, "|", denote choice. For example, "a|b" means a choice between "a" and "b"
  - Parentheses, "(" and ")", are used to indicate the scope of the operators "?", "*", "+" and "|"
  - Ellipses (i.e., "...") indicate points of extensibility

Note: The lack of ellipses doesn't mean no extensibility point exists; it's just not explicitly called out.

1.5 Terminology

Table 1. Terminology

Term	Definition
BMC	Baseboard Management Controller
CIMI	Cloud Infrastructure Management Interface
HTTP	Hypertext Transfer Protocol
iSCSI	Internet Small Computer Systems Interface. Specification available at RFC 3720 and RFC 3721
iQN	iSCSI Qualified Name format defined in RFC 3720 and RFC 3721
JSON*	JavaScript Object Notation*
NIC	Network Interface Card
NQN	NVMe Qualified Name described in Section 7.9 of the NVMe Base specification, available at *http://www.nvmexpress.org/specifications*
NVMe*	Non-volatile Memory Express*. Specification available at *http://www.nvmexpress.org/*
OCCI	Open Cloud Computing Interface
OEM	Original Equipment Manufacturer
OData	Open Data Protocol
OVF	Open Virtualization Format
PODM	POD Manager
PSME	Pooled System Management Engine
TPM	Trusted Platform Module
REST	Representational State Transfer
URI	Uniform Resource Identifier
UUID	Universally Unique Identifier
XML	Extensible Markup Language

1.6 References

Table 2. Reference Documents and Resources

Doc ID	Title	Location
608486	*Intel® Rack Scale Design (Intel® RSD) Pooled System Management Engine (PSME) User Guide Software v2.4*	Note: *https://www.intel.com/content/www/us/en/architecture-and-technology/rack-scale-design/rack-scale-design-resources.html*
608487	*Intel® Rack Scale Design (Intel® RSD) Conformance and Software Reference Kit Getting Started Guide v2.4*	
608488	*Intel® Rack Scale Design (Intel® RSD) POD Manager (PODM) Release Notes Software v2.4*	
608489	*Intel® Rack Scale Design (Intel® RSD) POD Manager (PODM) User Guide Software v2.4*	
608490	*Intel® Rack Scale Design (Intel® RSD) Pooled System Management (PSME) Release Notes Software v2.4*	
608491	*Intel® Rack Scale Design Storage Services API Specification Software v2.4*	
608492	*Intel® Rack Scale Design (Intel® RSD) Architecture Specification Software v2.4*	
608494	*Intel® Rack Scale Design (Intel® RSD) Rack Management Module (RMM) Representatinal State Transfer (REST) API Specification Software v2.4*	
608495	*Intel® Rack Scale Design (Intel® RSD) Generic Assets Management Interface (GAMI) API Specification v2.4*	

Doc ID	Title	Location
608496	*Intel® Rack Scale Design (Intel® RSD) Pooled System Management Engine (PSME) REST API Specification Software v2.4*	
608497	*Intel® Rack Scale Design (Intel® RSD) Conformance Test Suite (CTS) Release Notes*	
608298	*Field Programmable Gate Array (FPGA) over Fabric Protocol Architecture Specification*	https://cdrdv2.intel.com/v1/dl/getContent/608298
596167	*Intel® Rack Scale Design (Intel® RSD) for Cascade Lake Platform Firmware Extension Specification*	https://cdrdv2.intel.com/v1/dl/getContent/596167
DSP0266	*Redfish* Scalable Platforms Management API Specification, v1.5.0*	https://www.dmtf.org/sites/default/files/standards/documents/DSP0266_1.5.0.pdf
DSP8010	*Redfish Schema v2018.1*	https://www.dmtf.org/sites/default/files/standards/documents/DSP8010_2018.1.zip
DSP8010	*Redfish Schema v2018.2*	https://www.dmtf.org/sites/default/files/standards/documents/DSP8010_2018.2.zip
DSP8011	*Redfish Base Registry v1.0.0*	https://www.dmtf.org/sites/default/files/standards/documents/DSP8011_1.0.0a.json
RFC2119	*Key Words for Use in RFCs to Indicate Requirement Levels, March 1997*	https://ietf.org/rfc/rfc2119.txt
RFC2616	*Hypertext Transfer Protocol - HTTP/1.1*	https://tools.ietf.org/html/rfc2616
RFC3986	*Uniform Resource Identifier (URI): Generic Syntax*	https://tools.ietf.org/html/rfc3986
RFC5789	*PATCH Method for HTTP*	https://tools.ietf.org/html/rfc5789
N/A	*Swordfish* Scalable Storage Management API Specification v1.0.6*	https://www.snia.org/sites/default/files/technical_work/Swordfish/Swordfish_v1.0.6_specification.pdf
N/A	*Scalable Platforms Management Forum (SPMF)*	https://www.dmtf.org/standards/spmf

Note: Documents referenced in this table which have a Doc ID, but cannot be accessed, can be obtained by calling 1-800-548-4725 or by visiting *www.intel.com/design/literature.htm* obtain a copy.

§

2.0 PODM API

2.1 PODM API Structure and Relations

The PODM Representational State Transfer (REST) API provides the REST-based interface that allows full management of the Intel® RSD POD, including asset discovery, configuration, and composed node assembly.

2.1.1 PODM API Resource Hierarchy

The PODM API provides a set of its resources, as well as resources aggregated from the POD that it manages. Figure 1 presents the PODM's resources only.

Figure 1. PODM REST API Hierarchy

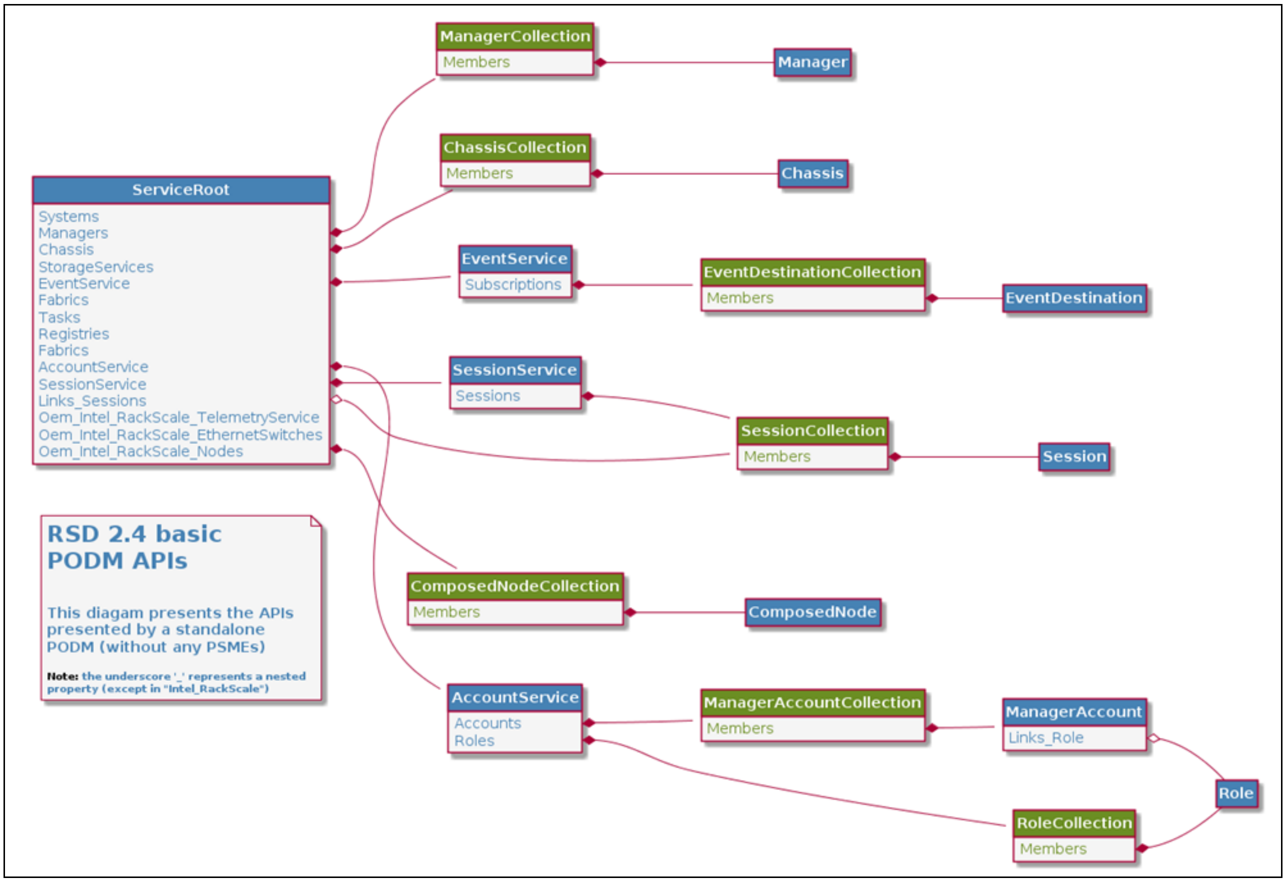


Table 3. Resources and URI

Resource	URI
Service Root	`/redfish/v1`
Chassis Collection	`/redfish/v1/Chassis`
Chassis	`/redfish/v1/Chassis/{chassisID}`
Computer Systems Collection	`/redfish/v1/Systems`
Computer Systems	`/redfish/v1/Systems/{systemID}`

Resource	URI
Processors Collection	/redfish/v1/Systems/{systemID}/Processors
Processors	/redfish/v1/Systems/{systemID}/Processors/{processorID}
Acceleration Functions Collection	/redfish/v1/Systems/{systemID}/Processors/{processorID}/Functions
Acceleration Function	/redfish/v1/Systems/{systemID}/Processors/{processorID}/Functions/{functionID}
Memory Collection	/redfish/v1/Systems/{systemID}/Memory
Memory	/redfish/v1/Systems/{systemID}/Memory/{memoryID}
Managers Collection	/redfish/v1/Managers
Managers	/redfish/v1/Managers/{managerID}
NetworkProtocol	/redfish/v1/Managers/{managerID}/NetworkProtocol
Network Interface Collection	/redfish/v1/Systems/{systemID}/EthernetInterfaces /redfish/v1/Managers/{managerID}/EthernetInterfaces

Network Interface	/redfish/v1/Systems/{systemID}/EthernetInterfaces/{nicID}
	/redfish/v1/Managers/{managerID}/EthernetInterfaces/{nicID}

EthernetSwitches Collection	/redfish/v1/EthernetSwitches
EthernetSwitches	/redfish/v1/EthernetSwitches/{switchID}
EthernetSwitches Ports Collection	/redfish/v1/EthernetSwitches/{switchID}/Ports
EthernetSwitchea Ports	/redfish/v1/EthernetSwitches/{switchID}/Ports/{portID}
VLAN Network Interface Collection	/redfish/v1/EthernetSwitches/{switchID}/Ports/{portID}/VLANs /redfish/v1/Systems/{systemID}/EthernetInterfaces/{nicID}/VLANs /redfish/v1/Managers/{managerID}/EthernetInterfaces/{nicID}/VLANs

VLAN Network Interface	/redfish/v1/EthernetSwitches/{switchID}/Ports/{portID}/VLANs/{vlanID}
	/redfish/v1/Systems/{systemID}/EthernetInterfaces/{nicID}/VLANs/{vlanID}
	/redfish/v1/Managers/{managerID}/EthernetInterfaces/{nicID}/VLANs/{vlanID}

Composed Nodes Collection	/redfish/v1/Nodes
Composed Nodes	/redfish/v1/Nodes/{nodeID}
Power	/redfish/v1/Chassis/{chassisID}/Power
Thermal	/redfish/v1/Chassis/{chassisID}/Thermal
Storage Subsystem Collection	/redfish/v1/Systems/{systemID}/Storage
Storage Subsystem	/redfish/v1/Systems/{systemID}/Storage/{storageID}
Drives	/redfish/v1/Chassis/{chassisID}/Drives/{driveID}
Fabrics collection	/redfish/v1/Fabrics
Fabrics	/redfish/v1/Fabrics/{fabricID}
Fabrics Switches collection	/redfish/v1/Fabrics/{fabricID}/Switches
Fabrics Switches	/redfish/v1/Fabrics/{fabricID}/Switches/{switchID}
Fabrics Switches Ports collection	/redfish/v1/Fabrics/{fabricID}/Switches/{switchID}/Ports
Fabrics Switches Ports	/redfish/v1/Fabrics/{fabricID}/Switches/{switchID}/Ports/{portID}
Fabrics Zones collection	/redfish/v1/Fabrics/{fabricID}/Zones
Fabrics Zones	/redfish/v1/Fabrics/{fabricID}/Zones/{zoneID}
Endpoints Collection	/redfish/v1/Fabrics/{fabricID}/Endpoints
Endpoints	/redfish/v1/Fabrics/{fabricID}/Endpoints/{endpointID}
PCIeDevices	/redfish/v1/Chassis/{chassisID}/PCIeDevices/{deviceID}
PCIeDevices Function	/redfish/v1/Chassis/{chassisID}/PCIeDevices/{deviceID}/Functions/{functionID}
StorageServices Collection	/redfish/v1/StorageServices/

Resource	URI
StorageService	/redfish/v1/StorageServices/{serviceId}
Volumes Collection	/redfish/v1/StorageServices/Volumes
Volumes	/redfish/v1/StorageServices/{serviceId}/Volumes/{volumeId}
StoragePools Collection	/redfish/v1/StorageServices/{storageId}/StoragePools
StoragePools	/redfish/v1/StorageServices/{serviceId}/{storageId}/StoragePools/{poolId}
Account Service	/redfish/v1/AccountService
Role Collection	/redfish/v1/AccountService/Roles/
Role	/redfish/v1/AccountService/Roles/{roleID}
Session Service	/redfish/v1/SessionService
Session Collection	/redfish/v1/SessionService/Sessions/
Session	/redfish/v1/SessionService/Sessions/{sessionID}
Manager Account Collection	/redfish/v1/AccountService/Accounts/
Manager Account	/redfish/v1/AccountService/Accounts/{accountID}
TelemetryService	/redfish/v1/Oem/Intel_RackScale/TelemetryService
Metric Definition Collection	/redfish/v1/Oem/Intel_RackScale/TelemetryService/MetricDefinitions
Metric Definition	/redfish/v1/Oem/Intel_RackScale/TelemetryService/MetricDefinitions/{metricDefinitionId}

§

3.0 Support for Requests with Query Parameters

The PODM REST API supports GET requests with query parameters as described in section "Query parameters" of the Redfish specification (refer to Table 2). The following attributes are supported:

```
filter
```

For queries with unsupported attributes, the service shall respond as required by the Redfish specification.

Note: The queries should be encoded using Percent-encoding (a.k.a. URL encoding) as defined in the *Uniform Resource Identifier (URI): Generic Syntax*, Section 2.1 of RFC3986 (refer to Table 2).

3.1 Query with *Filter* Attribute

GET requests with `filter` query parameter are supported on all Collection resources except for Composed Node Collection. Section Query parameters define the syntax of the queries for Filter of the Redfish specification (refer to Table 2).

3.1.1 Example Queries

Note: The examples below are presented before encoding.

Query for all Volumes which are not assigned to a remote host:

```
GET /redfish/v1/StorageServices/{ssid}/Volumes$filter=Oem/Intel_RackScale/Assigned eq
false
```

Query for all Physical Computer Systems with State Enabled and Health OK:

```
GET /redfish/v1/Systems$filter=Status.State eq 'Enabled' and Status.Health eq 'OK’ or
SystemType eq 'Physical'
```

§

4.0 PODM REST API Error Codes

This section contains descriptions of all error codes that may be returned by the REST calls implemented in the PODM REST API of the Intel® RSD v2.4 release.

4.1 API Error Response

In case of an error, the 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 enable deterministic error semantics. For this reason, wherever required by the Redfish specification (refer to Table 2), the PODM REST API shall return an extensive error resource, which is represented as a JSON object with a single property named "error". The value of the property shall be a JSON object with the properties shown in Table 4.

Table 4. API Error Response Attributes

Attribute	Description
code	A string indicating a specific MessageId from the message registry. "Base.1.0.GeneralError" should be used only when no other message is better.
message	A human-readable error message is corresponding to the message in the message registry.
@Message.ExtendedInfo	An array of message objects describing one or more error message(s).

4.1.1 Message Object

Message Objects provide additional information about an object, property, or error response. Messages are represented as a JSON object with the properties shown in Table 5.

Table 5. API Error Response Attributes

Attribute	Description
MessageId	A string indicating a specific error or message (not to be confused with the HTTP status code). This code can be used to access a detailed message from a message registry.
Message	A human-readable error message indicating the semantics associated with the error. This is the complete message, and it does not rely on substitution variables.
MessageArgs	An optional array of strings representing the substitution parameter values for the message. This is included in the response if a MessageId is specified for a parameterized message.
Severity	An optional string representing the severity of the error.
Resolution	An optional string describing recommended action(s) to take to resolve the error.
RelatedProperties	An optional array of JSON Pointers defining the specific properties in a JSON payload described by the message.

4.1.2 Error Message Definitions

The messages returned by a Redfish service are defined in Message Registries. In the current implementation, the PSME REST API responds with messages from two registries:

- The Redfish Base Registry v1.0.0, refer to Table 2.
- The Intel RackScale Registry, presented in the next section.

The URIs of the registries may also be obtained from the service by querying the Message Registry File API at `/redfish/v1/Registries`.

4.1.3 Intel RackScale Message Registry

The registry contains two RSD-specific error messages.

Request:

```
GET /registries/Intel_RackScale
Content-Type: application/json
```

Response:

```
{
  "@odata.type": "#MessageRegistry.v1_0_0.MessageRegistry",
  "Id": "Intel_RackScale.1.0.0",
  "Name": "Intel RackScale Message Registry",
  "Language": "en",
  "Description": "This registry defines messages specific to Intel RackScale",
  "RegistryPrefix": "Intel_RackScale",
  "RegistryVersion": "1.0.0",
  "OwningEntity": "Intel Corporation",
  "Messages": {
    "PropertyNotModifiable": {
      "Description": "Indicates that a property cannot be modified even though the
metadata specifies it as writable",
      "Message": "The service is unable to modify the property %1 even though metadata
specifies it as writeable.",
      "Severity": "Warning",
      "NumberOfArgs": 1,
      "ParamTypes": [
        "string"
      ],
      "Resolution": "Remove the unmodifiable property from the request body and
resubmit the request."
    },
    "PropertyValueRestricted": {
      "Description": "Indicates that the value given for a property is not within
restrictions imposed by the Service (even though it may be correct according to
metadata)",
      "Message": "The value %1 for property %2 is not within restrictions imposed by
the Service.",
      "Severity": "Warning",
      "NumberOfArgs": 1,
      "ParamTypes": [
        "string",
        "string"
      ],
      "Resolution": "Correct the value for the property in the request body and
resubmit the request."
    }
  }
}
```

4.1.4 Example Error JSON Object

```
{
    "error": {
        "code": "Base.1.0.GeneralError",
        "message": "A general error has occurred. See ExtendedInfo for more
information.",
        "@Message.ExtendedInfo": [
            {
                "@odata.type" : "/redfish/v1/$metadata#Message.v1_0_5.Message",
                "MessageId": "Base.1.0. MalformedJSON",
                "Message": "The request body submitted was malformed JSON and could 
not be parsed by the receiving service",
                "Severity": "Error"
            }
            {
                "@odata.type" : "/redfish/v1/$metadata#Message.v1_0_0.Message",
                "MessageId": "Base.1.0.PropertyNotWriteable",
                "RelatedProperties": [
                    "#/Name"
                ],
                "Message": "The property Name is a read property and cannot be 
assigned a value",
                "MessageArgs": [
                    "Name"
                ],
                "Severity": "Warning",
                "Resolution": "Remove the property from the request body and resubmit 
the request if the operation failed"
            }
        ]
    }
}
```

4.2 API Error Codes

In general, if an error isn't described in any of the following tables, it is mapped into an HTTP 500 Internal Error Code. (refer to Table 2). The client should be prepared to handle the error codes shown in Table 6.

4.2.1 General Error Codes

For a detailed list of Error Codes, refer to *Redfish* Scalable Platforms Management API Specification*, Section 6.5.2 (refer to Table 2).

The client should be prepared to handle the following error codes:

Table 6. HTTP Error Status Codes

HTTP Status Code	Description	Extended Error required
400 Bad Request	The request could not be processed because it contains missing or invalid information (such as validation error on an input field, a missing required value, and so on). An extended error shall be returned in the response body.	Yes
404 Not Found	The request specified a URI of a resource that does not exist.	No
405 Method Not Allowed	The HTTP verb specified in the request (for example, DELETE, GET, HEAD, POST, PUT, PATCH) is not supported for the request URI. The response shall include an Allow header, which provides a list of methods supported by the resource identified by request URI.	No
406 Not Acceptable	The Accept header was specified in the request and the resource identified by this request is not capable of generating a representation corresponding to one of the media types in the Accept header.	No
409 Conflict	A creation or update request could not be completed, because it would cause a conflict in the current state of the resources supported by the platform (for example, an attempt to set multiple attributes that work in a linked manner using incompatible values).	No

HTTP Status Code	Description	Extended Error required
500 Internal Server Error	The server encountered an unexpected condition that prevented it from fulfilling the request. An extended error shall be returned in the response body	Yes
501 Not Implemented	The server does not (currently) support the functionality required to fulfill the request. This is the appropriate response when the server does not recognize the request method and is not capable of supporting it for any resource.	No
503 Service Unavailable	The server is currently unable to handle the request due to temporary overloading or maintenance of the server.	No

4.2.2 PATCH Method Error Codes

For the PATCH method error codes, the Intel® RSD service conforms to the *PATCH Method for HTTP*, RFC 5789 standard, Table 2. The service responds with the following error codes in the cases listed:

- **400 Bad Request** - Malformed JSON in the request (such as values not in range, an unknown property and so on). The code, message and extended information within the error response explain why a request was rejected.

  Of particular concern are the RSD-specific messages from the `Intel_RackScale` registry. `PropertyNotModifiable` is returned when a PATCH request was sent for a property which, while writable according to metadata, is read-only on the POD Manager API. `PropertyValueRestricted` is returned when a PATCH request contains a value for a property which is compliant with metadata, but the service has additional restrictions on the acceptable values for that property which were not met by request.
- **405 Method Not Allowed** - Resource does not support PATCH method.
- **409 Conflict** - Update cannot be executed at this moment. The user might be able to resolve the conflict and resubmit the request.
- **501 Not Implemented** - Resource supports PATCH method, but current implementation does not (for example, underlying hardware does not support the functionality).
- **500 Internal Server Error** - All other situations in which the previous codes do not fit (for instance, underlying hardware does not allow to execute a particular request).

4.2.3 POST Method Error Codes

The `POST` method is used to create new resources (`POST` request is submitted to the resource collection in which the new resource will belong) or to initiate an operation on the object (sending `POST` method to the URI of the action).

Service responds with the following error codes in the cases listed below:

- **400 Bad Request** – malformed JSON in the request (values not in range, unknown property, etc.)
- **405 Method Not Allowed** – the resource does not support `POST` method
- **409 Conflict** – update cannot be executed at this moment. moment or `POST` method payload refers to a resource that does not exist. The user might be able to resolve the conflict and resubmit the request.
- **501 Not Implemented** – resource supports the `POST` method, but current implementation doesn't (e.g., underlying hardware doesn't support such functionality)
- **500 Internal Server Error** – all other situations where any of the above codes do not fit (e.g., underlying hardware does not allow execution of this particular request). Extended error information should provide information on why the operation failed.

§

5.0 PODM REST API Definition

The JSON* examples in this document are informative, not normative. Metadata files that are referenced by this specification are normative.

5.1 Odata* Support

Intel® RSD supports Odata* v4.0 as it is defined in *Redfish* Schema v2018.1, DSP8010* (refer to Table 2).

All resources within this REST API are identified by unique identifier property named `"@odata.id"`. Resource Identifiers are represented in JSON* payloads as Uniform Resource Identifier (URI) paths relative to the Redfish Schema portion of the URI. That is, they always start with `"/redfish/"`. The resource identifier is the canonical URL for the resource and can be used to retrieve or edit the resource, as appropriate.

5.2 Protocol Version

The protocol version is separate from the version of the resources or the version of the Redfish Schema supported by them.

Each version of the Redfish protocol is strongly typed. This is accomplished using the URI of the Redfish service in combination with the resource obtained at that URI, called the `ServiceRoot`.

The root URI for this version of the Redfish protocol is `"/redfish/v1/"`.

While the primary version of the protocol is represented in the URI, the major, minor, and errata version of the protocol are represented in the version property of the `ServiceRoot` resource, as defined in *Redfish* Schema v2018.1, DSP8010* (refer to Table 2) for that resource. The protocol version is a string of the form:

```
MajorVersion.MinorVersion.Errata
```

Where:

- **MajorVersion** = integer: something in the class changed in a backward incompatible way.
- **MinorVersion** = integer: a minor update. New functionality may have been added, but nothing removed. Compatibility is preserved with previous minor versions.
- **Errata** = integer: something in the prior version was broken and needed to be fixed.

Any resource discovered through links found by accessing the root service or any service or resource referenced using references from the root service conforms to the same version of the protocol supported by the root service.

5.2.1 Operations

5.2.1.1 GET

Request:

```
GET /redfish
Content-Type: application/json
```

Response:

```
{
        "v1": "/redfish/v1/"
}
```

5.3 Odata Service Document

This service document provides a standard format for enumerating the resources exposed by the service, enabling generic hypermedia-driven Odata* clients to navigate to the resources of the service.

5.3.1 Operations

5.3.1.1 GET

Request:

```
GET /redfish/v1/odata
Content-Type: application/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata",
    "value": [
        {
            "name": "Service",
            "kind": "Singleton",
            "url": "/redfish/v1/"
        },
        {
            "name": "Systems",
            "kind": "Singleton",
            "url": "/redfish/v1/Systems"
        },
        {
            "name": "Chassis",
            "kind": "Singleton",
            "url": "/redfish/v1/Chassis"
        },
        {
            "name": "Managers",
            "kind": "Singleton",
            "url": "/redfish/v1/Managers"
        },
        {
            "name": "Nodes",
            "kind": "Singleton",
            "url": "/redfish/v1/Nodes"
        },
        {
            "name": "StorageServices",
            "kind": "Singleton",
            "url": "/redfish/v1/StorageServices"
        },
        {
            "name": "EthernetSwitches",
            "kind": "Singleton",
            "url": "/redfish/v1/EthernetSwitches"
        },
        {
            "name": "Fabrics",
            "kind": "Singleton",
            "url": "/redfish/v1/Fabrics"
        },
        {
            "name": "AccountService",
            "kind": "Singleton",
            "url": "/redfish/v1/AccountService"
        },
        {
            "name": "SessionService",
            "kind": "Singleton",
            "url": "/redfish/v1/SessionService"
        },
        {
        }
      ]
 }
```

5.4 Tagged Values Oem extension

Intel® RSD PODM allows for assigning persistent user-defined tags to any resource exposed on the API by modifying the TaggedValues property.

5.4.1 PATCH operation

Request:

```
PATH /redfish/v1/ExampleResource
Content-Type: application/json
{
    "Oem": {
        "Intel_RackScale": {
            "TaggedValues": {
                "SKU": "2018/07BYH",
                "Inventory": "Canada/Ontario"
            }
        }
    },
}
```

Response (for a resource without a resource-specific Oem section):

```
{
    …
    (the rest of the resource)
    …
    "Oem": {
        "Intel_RackScale": {
            "@odata.type": "#Intel.Oem.Resource",
            "TaggedValues": {
                "SKU": "2018/07BYH",
                "Inventory": "Canada/Ontario"
            }
        }
    },
}
```

Response (for a resource with a specific Oem section):

```
{
    …
    (the rest of the resource)
    …
    "Oem": {
        "Intel_RackScale": {
            "@odata.type": "#Intel.Oem.ExampleResource",
            "TaggedValues": {
                "SKU": "2018/07BYH",
                "Inventory": "Canada/Ontario"
            },
            …
            (the rest of Oem properties of the resource)
            …
        }
    },
}
```

5.5 Intel® RSD PODM Service Root

Intel® RSD PODM Service Root resource – entry point.

Property details available in `ServiceRoot_v1.xml` metadata file.

5.5.1 Operations

5.5.1.1 GET

Request:

```
GET /redfish/v1
Content-Type: application/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#ServiceRoot.ServiceRoot",
    "@odata.id": "/redfish/v1/",
    "@odata.type": "#ServiceRoot.v1_3_1.ServiceRoot",
    "Id": "RootService",
    "Name": "Root Service",
    "Description": "description-as-string",
    "RedfishVersion": "1.5.0",
    "UUID": "92384634-2938-2342-8820-489239905423",
    "Systems": {
        "@odata.id": "/redfish/v1/Systems"
    },
    "Chassis": {
        "@odata.id": "/redfish/v1/Chassis"
    },
    "Managers": {
        "@odata.id": "/redfish/v1/Managers"
    },
    "EventService": {
        "@odata.id": "/redfish/v1/EventService"
    },
    "Fabrics": {
      "@odata.id": "/redfish/v1/Fabrics"
    },
    "StorageServices": {
      "@odata.id": "/redfish/v1/StorageSerices"
    },
    "Oem": {
        "Intel_RackScale": {
            "@odata.type": "#Intel.Oem.ServiceRoot",
            "ApiVersion": "2.4.0",
            "Nodes": {
                "@odata.id": "/redfish/v1/Nodes"
            },
            "EthernetSwitches": {
                "@odata.id": "/redfish/v1/EthernetSwitches"
            },
            "TelemetryService": {
                "@odata.id": "/redfish/v1/Oem/Intel_RackScale/TelemetryService"
            }
        }
    },
   "Links": {},
   "AccountService": {
       "@odata.id": "/redfish/v1/AccountService"
   },
   "SessionService": {
       "@odata.id": "/redfish/v1/SessionService"
   }
}
```

5.5.1.2 PUT

Operation not allowed on this resource.

5.5.1.3 PATCH

Operation not allowed on this resource.

5.5.1.4 POST

Operation not allowed on this resource.

5.5.1.5 DELETE

Operation not allowed on this resource.

5.6 Managers Collection

Table 7. Managers Collection Attributes

Name	Managers		
Type URI	/redfish/v1/ Managers		
Attribute	Type	Required	Description
`Name`	String	Yes	Name of collection
`Members@odata.count`	Number	No	Collection members count
`Members`	Array	No	Contains the members of this collection.

5.6.1 Operations

5.6.1.1 GET

Request:

```
GET /redfish/v1/Managers
Content-Type: application/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#ManagerCollection.ManagerCollection",
    "@odata.id": "/redfish/v1/Managers",
    "@odata.type": "#ManagerCollection.ManagerCollection",
    "Name": "Manager Collection",
    "Members@odata.count": 3,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Managers/ManagementController1"
        },
        {
            "@odata.id": "/redfish/v1/Managers/RackManager1"
        },
        {
            "@odata.id": "/redfish/v1/Managers/EnclocureManager1"
        }
    ]
}
```

5.6.1.2 PUT

Operation not allowed on this resource.

5.6.1.3 PATCH

Operation not allowed on this resource.

5.6.1.4 POST

`POST` operation on the Manager Collection on the PODM REST API is used to register Redfish services to be discovered and managed by the PODM.

Table 8. Attributes of POST action on Manager Collection

Attribute	Type	Required	Description
`RemoteRedfishServiceUri`	`Edm.String`	Yes	Defines the URI of the Service Root for the remote Redfish service managed by the created Manager resource
`ServiceEntryPointUUID`	`Resource.UUID`	Yes	This property shall contain the UUID of the remote Redfish service

Request:

```
POST /redfish/v1/Managers
Content-Type: application/json
{
  "RemoteRedfishServiceUri": "https://10.2.234:443/redfish/v1",
  "ServiceEntryPointUUID": "123e4567-e89b-12d3-a456-426655440000
}
```

Response:

```
HTTP/1.1 201 Created
Location: http://<IP>:<PORT>/redfish/v1/Manager/4
((created resource body))
```

5.6.1.5 DELETE

Operation not allowed on this resource.

5.7 Manager

This schema defines a manager resource and its respective properties.

Table 9. Manager Attributes

Attribute	Type	Nullable	Description
`ManagerType`	`Manager.v1_0_0.ManagerType`	false	The value of this property shall describe the function of this manager. The value `EnclosureManager` shall be used if this manager controls one or more services through aggregation. The value BMC shall be used if this manager represents a traditional server management controller. The value ManagementController shall be used if none of the other enumerations apply.
`Links`	`Manager.v1_0_0.Links`	false	The Links property, as described by the Redfish Specification, shall contain references to resources that are related to, but not contained by (subordinate to), this resource.
`ServiceEntryPointUUID`	`Resource.UUID`	True	This property shall contain the UUID of the Redfish Service provided by this manager. Each Manager providing an Entry Point to the same Redfish Service shall report the same UUID value (even though the name of the property may imply otherwise). This property shall not be present if this manager does not provide a Redfish Service Entry Point.
`UUID`	`Resource.UUID`	True	The value of this property shall contain the universally unique identifier number for the manager.
`Model`	`Edm.String`	True	The value of this property shall contain information about how the manufacturer references this manager.

Attribute	Type	Nullable	Description
DateTime	Edm.DateTimeOffset	True	The value of this property shall represent the current DateTime value for the manager, with the offset from UTC, in Redfish Timestamp format.
DateTimeLocalOffset	Edm.String	True	The value is property shall represent the offset from UTC time that the current value of DateTime property contains.
FirmwareVersion	Edm.String	True	This property shall contain the firmware version as defined by the manufacturer for the associated manager.
SerialConsole	Manager.v1_0_0.SerialConsole	false	The value of this property shall contain information about the Serial Console service of this manager.
CommandShell	Manager.v1_0_0.CommandShell	false	The value of this property shall contain information about the Command Shell service of this manager.
GraphicalConsole	Manager.v1_0_0.GraphicalConsole	false	The value of this property shall contain information about the Graphical Console (KVM-IP) service of this manager.
Actions	Manager.v1_0_0.Actions	false	The Actions property shall contain the available actions for this resource.
Status	Resource.Status	false	This property shall contain any status or health properties of the resource.
EthernetInterfaces	EthernetInterfaceCollection.EthernetInterfaceCollection	false	The value of this property shall be a link to a collection of type EthernetInterfaceCollection.
SerialInterfaces	SerialInterfaceCollection.SerialInterfaceCollection	false	The value of this property shall be a link to a collection of type SerialInterfaceCollection which are for the use of this manager.
NetworkProtocol	ManagerNetworkProtocol.ManagerNetworkProtocol	false	The value of this property shall contain a reference to a resource of type ManagerNetworkProtocol which represents the network services for this manager.
LogServices	LogServiceCollection.LogServiceCollection	false	The value of this property shall contain a reference to a collection of type LogServiceCollection which are for the use of this manager.
VirtualMedia	VirtualMediaCollection.VirtualMediaCollection	false	The value of this property shall contain a reference to a collection of type VirtualMediaCollection which are for the use of this manager.
Redundancy	Collection(Redundancy.Redundancy)	True	The values of the properties in this array shall be used to show how this manager is grouped with other managers to form redundancy sets.
PowerState	Resource.PowerState	True	The value of this property shall contain the power state of the Manager.
HostInterfaces	HostInterfaceCollection.HostInterfaceCollection	false	The value of this property shall be a link to a collection of type HostInterfaceCollection.

Attribute	Type	Nullable	Description
AutoDSTEnabled	Edm.Boolean	false	The value of this property shall contain the enabled status of the automatic Daylight Saving Time (DST) adjustment of the manager's DateTime. It shall be true if Automatic DST adjustment is enabled and false if disabled.
RemoteRedfishServiceUri	Edm.String	True	This property shall contain the URI of the Redfish Service Root for the remote Manager represented by this resource. This property shall only be present when providing aggregation of Redfish services.
RemoteAccountService	AccountService.AccountService	false	This property shall contain a reference to the AccountService resource for the remote Manager represented by this resource. This property shall only be present when providing aggregation of Redfish services.

Table 10. Links Attributes

Attribute	Type	Nullable	Description
ManagerForServers	Collection(ComputerSystem.ComputerSystem)	True	This property shall contain an array of references to ComputerSystem resources of which this Manager instance has control.
ManagerForChassis	Collection(Chassis.Chassis)	True	This property shall contain an array of references to Chassis resources of which this Manager instance has control.
ManagerInChassis	Chassis.Chassis	false	This property shall contain a reference to the chassis that this manager is located in.

Intel® RSD OEM extensions:

Table 11. Manager Attributes

Attribute	Type	Nullable	Description
Trusted	Edm.Boolean	true	Indicates if the service represented by this Manager resource has been authenticated by the PODM

Table 12. ManagerLinks Attributes

Attribute	Type	Nullable	Description
ManagerForServices	Collection(StorageService.StorageService)	True	This property is an array of references to services that this manager has control over.
ManagerForEthernetSwitches	Collection(EthernetSwitch.v1_0_0.EthernetSwitch)	True	This property is an array of references to Ethernet switches that this manager has control over.
ManagerForFabrics	Collection(Fabric.Fabric)	True	This property is an array of references to fabrics that this manager has control over.

5.7.1 Operations

5.7.1.1 GET

Request:

```
GET /redfish/v1/Managers/PSME
Content-Type: application/json
```

Response:

```
{
  "@odata.context": "/redfish/v1/$metadata#Manager.Manager",
  "@odata.id": "/redfish/v1/Managers/PSME",
  "@odata.type": "#Manager.v1_4_0.Manager",
  "Id": "1",
  "Name": "Manager",
  "ManagerType": "BMC",
  "Description": "BMC",
  "ServiceEntryPointUUID": "92384634-2938-2342-8820-489239905423",
  "UUID": "00000000-0000-0000-0000-000000000000",
  "Model": "Joo Janta 200",
  "DateTime": "2015-03-13T04:14:33+06:00",
  "DateTimeLocalOffset": "+06:00",
  "Status": {
    "State": "Enabled",
    "Health": "OK",
    "HealthRollup": null
  },
  "AutoDSTEnabled": false,
  "GraphicalConsole": {
    "ServiceEnabled": true,
    "MaxConcurrentSessions": 2,
    "ConnectTypesSupported": [
      "KVMIP"
    ]
  },
  "HostInterfaces": [],
  "SerialConsole": {
    "ServiceEnabled": true,
    "MaxConcurrentSessions": 1,
    "ConnectTypesSupported": [
      "Telnet",
      "SSH",
      "IPMI"
    ]
  },
  "CommandShell": {
    "ServiceEnabled": true,
    "MaxConcurrentSessions": 4,
    "ConnectTypesSupported": [
      "Telnet",
      "SSH"
    ]
  },
  "FirmwareVersion": "1.00",
  "NetworkProtocol": {
    "@odata.id": "/redfish/v1/Managers/PSME/NetworkProtocol"
  },
  "EthernetInterfaces": {
    "@odata.id": "/redfish/v1/Managers/PSME/EthernetInterfaces"
  },
  "Links": {
    "@odata.type": "#Manager.v1_4_0.Links",
    "ManagerForServers": [],
    "ManagerForChassis": [
      {
        "@odata.id": "/redfish/v1/Chassis/FabricModule1"
      }
    ],
    "ManagerInChassis": {
      "@odata.id": "/redfish/v1/Chassis/Drawer1"
    },
    "ManagerForSwitches": [
      {
        "@odata.id": "/redfish/v1/Fabrics/PCIe/Switches/1"
      }
    ],
    "ManagerForPCIeSwitches": [],
    "ManagerForPCIeDevices": [],
    "Oem": {
      "Intel_RackScale": {
        "@odata.type": "#Intel.Oem.ManagerLinks",
        "ManagerForServices": [
          {
            "@odata.id": "/redfish/v1/StorageServices/NVMeoE1"
          }
        ],
        "ManagerForFabrics": [
          {
            "@odata.id": "/redfish/v1/Fabrics/PCIe"
          }
        ],
        "ManagerForEthernetSwitches": [
          {
            "@odata.id": "/redfish/v1/EthernetSwitches/Switch1"
          }
        ]
      }
    }
  },
  "Actions": {
    "#Manager.Reset": {
      "target": "/redfish/v1/Managers/PSME/Actions/Manager.Reset"
    },
    "Oem": {}
  },
  "Oem": {
    "Intel_RackScale": {
        "@odata.type": "#Intel.Oem.Manager",
        "Trusted": true
  }
}
```

5.7.1.2 PUT

The operation is not allowed on this resource.

5.7.1.3 PATCH

The operation is not allowed on this resource.

5.7.1.4 POST

The operation is not allowed on this resource.

5.7.1.5 DELETE

Request:

```
DELETE redfish/v1/Managers/PSME
```

Response:

```
HTTP/1.1 204 No Content
```

Or (when a task is created):

```
HTTP/1.1 202 Accepted
Location: http://<ip:port>/redfish/v1/TaskService/Tasks/1/TaskMonitor
{
  "@odata.context": "/redfish/v1/$metadata#Task.Task",
  "@odata.id": "/redfish/v1/TaskService/Tasks/1",
  "@odata.type": "#Task.v1_0_0.Task",
  "Id": "1",
  "Name": "Task 1",
  "TaskState": "New",
  "StartTime": "2017-12-06T04:45+01:00",
  "TaskStatus": "OK",
  "Messages": []
}
```

5.8 Chassis Collection

Table 13. Chassis Collection Attributes

Name	**Chassis**		
Type URI	**/redfish/v1/Chassis**		
Attribute	**Type**	**Required**	**Description**
`Name`	String	Yes	Name of collection
`Members@odata.count`	Number	No	Collection members count
`Members`	Array	No	Contains the members of this collection.

5.8.1 Operations

5.8.1.1 GET

Request:

```
GET /redfish/v1/Chassis
Content-Type: application/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#Chassis",
    "@odata.id": "/redfish/v1/Chassis",
    "@odata.type": "#ChassisCollection.ChassisCollection",
    "Name": "Chassis Collection",
    "Members@odata.count": 6,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Chassis/Pod"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/Rack1"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/Drawer1"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/FabricModule1"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/Sled1"
        },
        {
            "@odata.id": "/redfish/v1/Chassis/Blade1"
        }
    ]
}
```

5.8.1.2 PUT

Operation not allowed on this resource.

5.8.1.3 PATCH

Operation not allowed on this resource.

5.8.1.4 POST

Operation not allowed on this resource.

5.8.1.5 DELETE

Operation not allowed on this resource.

5.9 Chassis

This is the Schema Definition for the Chassis resource, which represents properties for physical components for any system. This one object is intended to represent Racks, Rackmount Servers, Blades, Standalone, Modular Systems, Enclosures, and all other containers. The non-CPU/device centric parts of the schema are all accessed either directly or indirectly through this resource.

Details of this resource are described in metadata file: `Chassis_v1.xml`. Original Equipment Manufacturer (OEM) extensions details are available in `IntelRackScaleOem_v1.xml`.

5.9.1 Operations

5.9.1.1 GET

Request:

```
GET /redfish/v1/Chassis/1
Content-Type: application/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#Chassis/Members/$entity",
    "@odata.id": "/redfish/v1/Chassis/Rack1",
    "@odata.type": "#Chassis.v1_7_0.Chassis",
    "Id": "Rack1",
    "ChassisType": "RackMount",
    "Name": "name-as-string",
    "Description": "description-as-string",
    "Manufacturer": "Intel Corporation",
    "Model": "model-as-string",
    "SKU": "sku-as-string",
    "SerialNumber": "serial-number-as-string",
    "PartNumber": "part-number-as-string",
    "AssetTag": null,
    "IndicatorLED": "Unknown",
    "Status": {
        "State": "Enabled",
        "Health": "OK"
    },
    "Oem": {
        "Intel_RackScale": {
            "@odata.type": "#Intel.Oem.RackChassis",
            "Location": {
                "Id": "Rack1",
                "ParentId": "Pod1"
            },
            "RackSupportsDisaggregatedPowerCooling": true,
            "UUID": "Unique ID",
            "GeoTag": "54.348103, 18.645172"
        }
    },
    "PowerState": null,
    "Thermal": {
      "@odata.id": "/redfish/v1/Chassis/Rack1/Thermal"
    },
    "Power": {
      "@odata.id": "/redfish/v1/Chassis/Rack1/Power"
    },
    "Links": {
        "@odata.type": "#Chassis.v1_7_0.Links",
        "Contains": [
            {
                "@odata.id": "/redfish/v1/Chassis/Drawer1"
            }
        ],
        "ContainedBy": null,
        "ComputerSystems": [],
        "ManagedBy": [
            {
                "@odata.id": "/redfish/v1/Managers/RMM"
            }
        ],
        "ManagersInChassis": [
            {
                "@odata.id": "/redfish/v1/Managers/RMM"
            }
        ],
        "Oem": {
           "Intel_RackScale": {
            "@odata.type": "#Intel.Oem.ChassisLinks",
             "Switches": []
           }
        }
    }
}
```

5.9.1.2 PUT

Operation not allowed on this resource.

5.9.1.3 PATCH

Following properties can be updated by using the PATCH operation:

Table 14. Chassis Attributes

Attribute	Type	Required	Description
`AssetTag`	String	No	The user assigned asset tag for this chassis.
`Oem->Intel_RackScale->Location->Id`	String	No	The user assigned location id for this chassis can be changed only for the Rack Chassis. Support for changing this attribute is not mandatory in this API Specification v2.4, but if implemented it needs to be aligned to this specification.

Request:

```
PATCH /redfish/v1/Chassis/Rack1
Content-Type: application/json
{
    "AssetTag": "Rack#1",
    "Oem": {
        "Intel_RackScale": {
            "Location": {
                "Id": "1234"
            }
        }
    }
}
```

Response:

```
HTTP/1.1 204 No Content
```

Or

```
HTTP/1.1 200 OK
```

With full resource representation.

5.9.1.4 POST

Operation not allowed on this resource.

5.9.1.5 DELETE

Operation not allowed on this resource.

5.10 StorageServiceCollection

Intel® Rack Scale Design `StorageServiceCollection` resource – provides a collection of available storage services. Refer to Table 15 for a list of attributes.

Table 15. Storage Service Collection Attributes

Name	**Storage Service**		
Type URI	**/redfish/v1/StorageServices**		
Attribute	**Type**	**Mandatory**	**Description**
`Name`	String	Yes	Name of service collection.
`Members@odata.count`	Number	No	Collection members count
`Members`	Array	No	Contains the members of this collection.

5.10.1 Operations

5.10.1.1 GET

Request:

```
GET /redfish/v1/Services
Content-Type: application/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#Services",
    "@odata.id": "/redfish/v1/Services",
    "@odata.type": "#StorageServiceCollection.StorageServiceCollection",
    "Name": "Storage Services Collection",
    "Description": "Collection of Storage Services",
    "Members@odata.count": 1,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Services/RSS1"
        }
    ]
}
```

5.10.1.2 PUT

Operation not allowed on this resource.

5.10.1.3 PATCH

Operation not allowed on this resource.

5.10.1.4 POST

Operation not allowed on this resource.

5.10.1.5 DELETE

Operation not allowed on this resource.

5.11 ComposedNodeCollection

Intel® RSD ComposedNodeCollection resource – provides a collection of all logical nodes. Refer to Table 16 for a list of attributes.

Table 16. ComposedNodeCollection Attributes

Name	ComposedNodeCollection		
Type URI	/redfish/v1/Nodes		
Attribute	Type	Mandatory	Description
Name	String	Yes	Name of collection
Members@odata.count	Number	No	Collection members count
Members	Array	No	Contains the members of this collection.
Actions	Object	No	Actions available: **Allocate** – this action is the first mandatory step to create a composed node. In response to this action, proper resources are found and allocated for node composition. Node resource is created, and URL (link) of this node is returned. To allocate a Composed Node using PODM REST API it is necessary to create a JSON template describing requested resources. The JSON template contains various details concerning resources to be used in Composed Node. All JSON template elements are optional, but they should not be mutually exclusive. It is possible to supply the PODM with a JSON template containing no specific requirements (e.g. {} – a pair of empty curly braces in HTTP request body) thus allowing PODM to propose a Composed Node containing resources chosen arbitrarily by PODM. The format of JSON template (action payload) is described in Table 17

Table 17. ComposedNodeCollection Allocate Action Attributes

<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>Name</td><td>String</td><td>No</td><td>Name of the composed node.
Because ComposedNode is a Redfish resource - its Name field is mandatory, so an attempt to directly set a <b>null</b> value results in an expected error. PODM sets a default name for newly created ComposedNode resource only upon not supplying the Name attribute.</td></tr>
<tr><td>Description</td><td>String</td><td>No</td><td>Description of ComposedNode</td></tr>
<tr><td>Processors</td><td>Array</td><td>No</td><td>An array of requirements for a processor for ComposedNode. Each processor requirement may contain one or more optional attributes:
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>Model</td><td>String</td><td>Processor model that should be used for ComposedNode (exact model name)</td></tr>
<tr><td>TotalCores</td><td>Number</td><td>Minimum number of processor cores</td></tr>
<tr><td>AchievableSpeedMHz</td><td>Number</td><td>Minimum achievable processor operating frequency</td></tr>
</table></td></tr>
</table>

Attribute	Type	Mandatory	Description		
			InstructionSet	String	Processor supported instruction set. "x86" – x86 32-bit "x86-64" – x86 64-bit "IA-64" – Intel IA-64 "ARM-A32" – ARM 32-bit "ARM-A64" – ARM 64-bit "MIPS32" – MIPS 32-bit "MIPS64" – MIPS 64-bit "OEM" – OEM-defined
			Resource	Object	Reference to a particular processor that should be used in ComposedNode
			Chassis	Object	Link to chassis object within this processor should be contained.
			Oem -> Intel_RackScale -> Brand	String	Brand of CPU that should be used to allocate node. Allowable values: New Intel® Xeon® branding: "Silver", "Gold", "Platinum" Intel® Xeon® family: E3, E5, E7 SoC/Intel Atom® family: X3 (Avoton), X5 (Broadwell-DE), X7 Core family: I3, I5, I7 "Unknown" – processor doesn't fit to any above categories
			Oem -> Intel_RackScale -> Capabilities	Array	Array of strings describing processor capabilities (like reported in /proc/cpuinfo flags), such as: "sse" - Streaming SIMD Extensions "avx" - Advanced Vector Extensions
			ProcessorType	String	This property contains the string which identifies the type of processor: "CPU" "FPGA" "GPU" "DSP" "Accelerator" "OEM"
			Connectivity	Array	An array of values determining how a processor should be connected to the node: "Local" "Ethernet" "RemotePCIe"

<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>Memory</td><td>Array</td><td>No</td><td>An array of requirements for memory for ComposedNode.
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>CapacityMiB</td><td>Number</td><td>Minimum memory capacity requested for ComposedNode</td></tr>
<tr><td>MemoryDeviceType</td><td>String</td><td>Type details of DIMM:
"DDR"<br>"DDR2"<br>"DDR3"<br>"DDR4"<br>"DDR4 SDRAM"
"DDR4E SDRAM"
"LPDDR4SDRAM"
"DDR3 SDRAM"
"LPDDR3 SDRAM"
"DDR2 SDRAM"
"DDR2 SDRAM-FB-DIMM"
"DDR2SDRAM-FB-DIMM PROBE"
"DDR SGRAM"
"DDR SDRAM"
"ROM"<br>"SDRAM"<br>"EDO"<br>"FastPageMode"
"PipelinedNibble"</td></tr>
<tr><td>SpeedMHz</td><td>Number</td><td>Minimum supported memory speed.</td></tr>
<tr><td>Manufacturer</td><td>String</td><td>Requested memory manufacturer.</td></tr>
<tr><td>DataWidthBits</td><td>Number</td><td>Requested memory data width in bits.</td></tr>
<tr><td>Resource</td><td>Object</td><td>Reference to a particular memory module that should be used in ComposedNode</td></tr>
<tr><td>Chassis</td><td>Object</td><td>Link to chassis object within this memory DIMM should be contained.</td></tr>
</table></td></tr>
<tr><td>RemoteDrives</td><td>Array</td><td>No</td><td>An array of requirements for remote drives that should be created/connected to ComposedNode
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>CapacityGiB</td><td>Number</td><td>This property is mutually exclusive with Resource - only one can be provided at the time.
Minimum drive capacity requested for ComposedNode</td></tr>
<tr><td>Protocol</td><td>String</td><td>Defines which protocol should be used to communicate with the attached drive. Currently supported:</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>"iSCSI",
"NVMeOverFabrics"</td></tr>
<tr><td>Master</td><td>Object</td><td>Defines master volume that should be taken to create a new remote drive. It contains properties described in Table 18.</td></tr>
<tr><td>Resource</td><td>Object</td><td>Reference to a particular remote drive (represented by Volume or target Endpoint resource) that should be used for this ComposedNode.</td></tr>
</table>
</td></tr>
<tr><td>LocalDrives</td><td>Array</td><td>No</td><td>Array of requirements for local drives for ComposedNode.
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>CapacityGiB</td><td>Number</td><td>Minimum drive capacity requested for ComposedNode</td></tr>
<tr><td>Type</td><td>String</td><td>Drive type
"HDD"<br>"SSD"</td></tr>
<tr><td>MinRPM</td><td>Number</td><td>The minimum rotation speed of the requested drive</td></tr>
<tr><td>SerialNumber</td><td>String</td><td>Serial number of requested drive</td></tr>
<tr><td>Interface</td><td>String</td><td>The interface of the requested drive:
"SAS"<br>"SATA"<br>"NVMe"</td></tr>
<tr><td>Resource</td><td>Object</td><td>Reference to a particular local drive that should be used in ComposedNode</td></tr>
<tr><td>Chassis</td><td>Object</td><td>Link to chassis object within this drive should be contained.</td></tr>
<tr><td>FabricSwitch</td><td>Boolean</td><td>Determines if the local drive should be connected using a fabric switch or locally</td></tr>
</table>
</td></tr>
<tr><td>EthernetInterfaces</td><td>Array</td><td>No</td><td>An array of requirements for Ethernet interfaces of ComposedNode
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>SpeedMbps</td><td>Number</td><td>Minimum speed of Ethernet interface requested for ComposedNode</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>VLANs</td><td>Array</td><td>An array of VLANs that should be configured on the connected switch port for ComposedNode for given Ethernet interface in the following format:
<b>VLANId</b> – a number indicating VLAN Id
<b>Tagged</b> – Boolean value describing if given VLAN is tagged.
<b>Deprecated</b>: This is going to be removed in future releases of Intel® RSD. Equivalent functionality will be provided via CRUD operation on Ethernet Switch port VLANs.</td></tr>
<tr><td>PrimaryVLAN</td><td>Number</td><td>Primary VLAN ID that should be set for given Ethernet Interface.
<b>Deprecated</b>: This is going to be removed in future releases of Intel® RSD. Equivalent functionality will be provided via CRUD operation on Ethernet Switch port.</td></tr>
<tr><td>Resource</td><td>Object</td><td>Reference to a particular Ethernet interface that should be used in ComposedNode.</td></tr>
<tr><td>Chassis</td><td>Object</td><td>Link to chassis object within this network interface should be contained.</td></tr>
</table>
</td></tr>
<tr><td>TotalSystemMemoryMiB</td><td>Number</td><td>No</td><td>The minimum total amount of memory in ComposedNode.</td></tr>
<tr><td>TotalSystemCoreCount</td><td>Number</td><td>No</td><td>Minimum total processor core count in ComposedNode.</td></tr>
<tr><td>SupportedPerformanceConfigurations</td><td>Array</td><td>No</td><td>List of performance configurations which should be supported in the ComposedNode:
"StaticSpeedSelect"
"PrioritizedBaseFrequency"</td></tr>
<tr><td>Security</td><td>Object</td><td>No</td><td>Security requirements for ComposedNode
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>TpmPresent</td><td>Boolean</td><td>This is used to specify if ComposedNode should have a TPM module present.</td></tr>
<tr><td>TpmInterfaceType</td><td>String (enum)</td><td>This is used to specify the requested TPM interface type. It overrides TpmPresent attribute.</td></tr>
</table>
</td></tr>
</table>

Attribute	Type	Mandatory	Description		
			TxtEnabled	Boolean	This is used to specify if ComposedNode should have TXT mode enabled.
			ClearTPMOnDelete	Boolean	This is used to specify if TPM module should be cleared on ComposedNode DELETE request.
			ClearOptaneDCPersistentMemoryOnDelete	Boolean	This property is used to specify if Intel® Optane™ DC Persistent Memory should be erased on ComposedNode DELETE request.

Table 18. RemoteDrives Target Properties

Attribute	Type	Description
Type	String	Type of replication of master drive: Clone – volume should be cloned Snapshot – Copy on Write should be created from indicated volume (does not work for Protocol NVMeOverFabrics) ***Note:*** For iSCSI protocol, the newly created drive will be marked by PODM as Bootable
Resource	Object	Reference to a volume that should be used as a master for replication.

5.11.1 Operation

5.11.1.1 GET

Request:

```
GET /redfish/v1/Nodes
Content-Type: application/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#Nodes",
    "@odata.id": "/redfish/v1/Nodes",
    "@odata.type": "#ComposedNodeCollection.v1_1_0.ComposedNodeCollection",
    "Name": "ComposedNodes Collection",
    "Members@odata.count": 1,
    "Members": [
        {
            "@odata.id": "/redfish/v1/Nodes/Node1"
        }
    ],
    "Actions": {
        "@odata.type": "#ComposedNodeCollection.v1_1_0.Actions",
        "#ComposedNodesCollection.Allocate": {
            "target": "/redfish/v1/Nodes/Actions/Allocate"
        }
    }
}
```

5.11.1.2 PUT

Operation not allowed on this resource.

5.11.1.3 PATCH

Operation not allowed on this resource.

5.11.1.4 POST

Note: Currently, a user can request allocation of a single node with a single request. Node components—CPU, Memory, Local Storage, and Network Interface—must be located on a single physical Sled. Remote storage can be located anywhere in the POD.

Note: Below JSON is just an example. For more details refer to Section 7.0, Composed Node Additional Info.

Request:

```
POST /redfish/v1/Nodes/Actions/Allocate
Content-Type: application/json
{
        "@odata.type": "#ComposedNodeCollection.v1_1_0.Allocate",
        "Name": "My first composed node",
        "Description": "Test node",
        "Processors": [{
                 "@odata.type" : "AllocationComposedNodeRequest.v1_1_0.Processor"
                 "Model": "Multi-Core Intel(R) Xeon(R) processor 7xxx Series",
                 "TotalCores": 2,
                 "AchievableSpeedMHz": 2000,
                 "InstructionSet": "x86",
                 "Oem": {
                    "Intel_RackScale": {
                      "@odata.type":
"AllocationComposedNodeRequest.v1_1_0.ProcessorExtensions",
                      "Brand" : "E5",
                      "Capabilities": [ "sse" ]
              }
       },
       "Resource":{
                         "@odata.id": "/redfish/v1/Systems/System1/Processors/CPU1"
                 }
        }],
        "Memory": [{
                 "@odata.type": "AllocationComposedNodeRequest.v1_1_0.Memory",
                 "CapacityMiB": 16000,
                 "MemoryDeviceType": "DDR3",
                 "SpeedMHz": 1600,
                 "Manufacturer": "Intel",
                 "DataWidthBits": 64,
                 "Resource": {
                         "@odata.id": "/redfish/v1/Systems/System1/Memory/Dimm1"
                 },
                 "Chassis": {
                         "@odata.id": "/redfish/v1/Chassis/Rack1"
                 }
        }],
        "RemoteDrives": [{
                 "@odata.type" : "AllocationComposedNodeRequest.v1_1_0.RemoteDrive",
                 "CapacityGiB": 80,
                 "Protocol": "iSCSI",
                 "Master": {
                        "Type": "Snapshot",
                        "Resource": {
                                 "@odata.id":
"/redfish/v1/StorageServices/iSCSI/Volumes/1"
                        }
                 }
        }],
        "LocalDrives": [{
                 "CapacityGiB": 500,
                 "Type": "HDD",
                 "MinRPM": 5400,
                 "SerialNumber": "12345678",
                 "Interface": "SATA",
                 "Resource": {
                        "@odata.id": "redfish/v1/Chassis/Blade1/Drives/Disk1"
                 },
                 "FabricSwitch": false
        }],
        "EthernetInterfaces": [{
                 "SpeedMbps": 1000,
                 "PrimaryVLAN": 100,
                 "VLANs": [{
                        "VLANId": 100,
                        "Tagged": false
                 }],
                 "Resource": {
                        "@odata.id":
"/redfish/v1/Systems/System1/EthernetInterfaces/LAN1"
                 }
        }],
        "Security": {
          "@odata.type" : "AllocationComposedNodeRequest.v1_1_0.Security",
          "TpmPresent": true,
          "TpmInterfaceType": "TPM2_0",
          "TxtEnabled": false,
          "ClearTPMOnDelete": true,
          "ClearOptaneDCPersistentMemoryOnDelete": true
        },
        "Oem": {
        },
        "TotalSystemMemoryMiB": 32000,
        "TotalSystemCoreCount": 2,
        "SupportedPerformanceConfigurations": ["StaticSpeedSelect"]
}
```

Response:

```
HTTP/1.1 201 Created
Location: http://<IP>:<Port>/redfish/v1/Nodes/2
```

5.11.1.5 DELETE

Operation not allowed on this resource.

5.12 Composed Node

Composed Node resource – provides detailed information about an assembled logical node identified by `{nodeID}`. Table 19 lists the attributes.

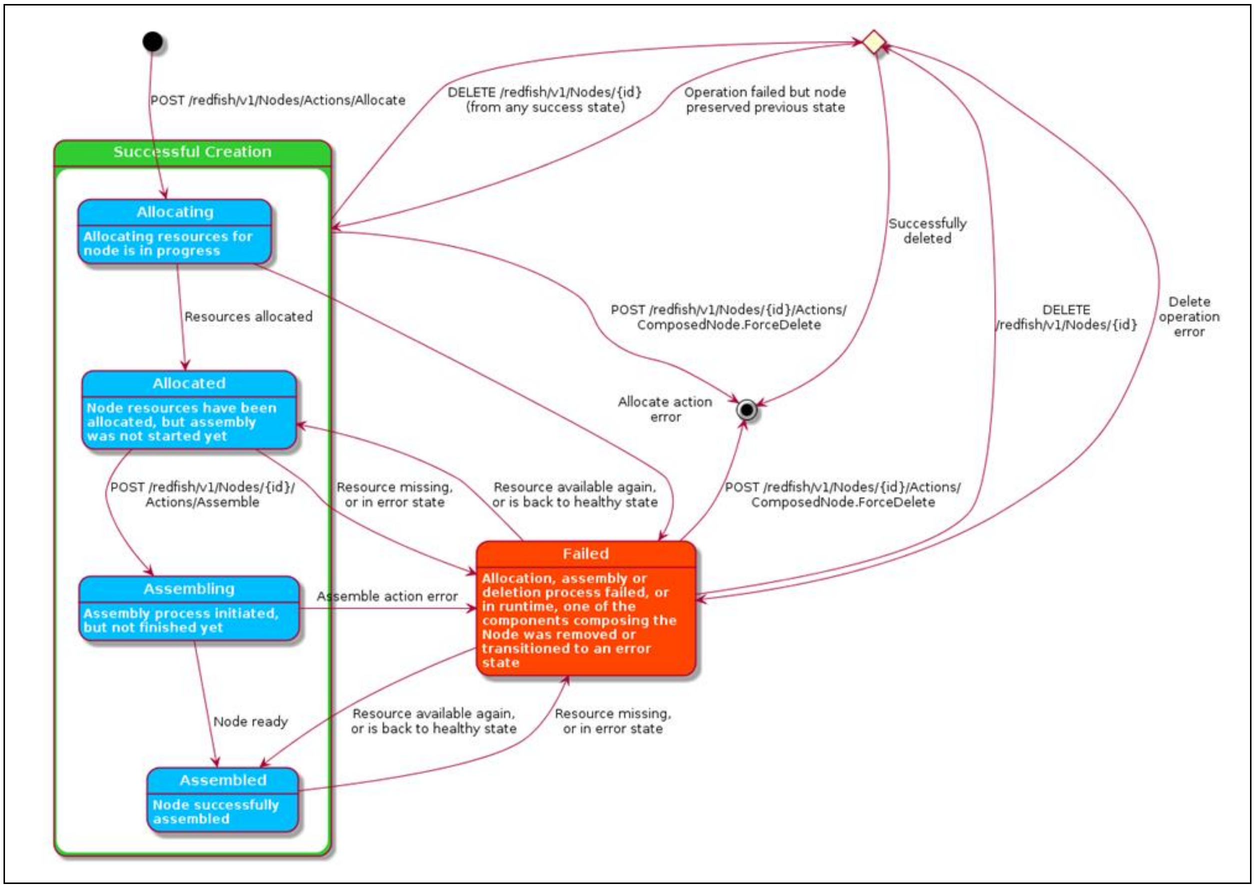
Table 19. Composed Node Attributes

Name	**Composed Node**		
Type URI	**/redfish/v1/Nodes/{nodeID}**		
Attribute	**Type**	**Mandatory**	**Description**
`Id`	String	Yes	Provides an ID of this resource
`Name`	String	Yes	Name of composed Node
`Description`	String	No	User provided node description
`UUID`	String	No	UUID of a computer system used as a base for this node.
`PowerState`	String (enum)	No	This is the current power state of the node: • `"On"` - The system is powered on • `"Off"` - The system is powered off, although some components may continue to have AUX power such as management controller • "`PoweringOn`" - A temporary state between Off and On. This temporary state can be very short. • "`PoweringOff`" - A temporary state between on and off. The power-off action can take time while the OS is in the shutdown process.
`Status`	Object, null	No	Refer to Section 5.1 for the status of the resource.
`ComposedNodeState`	String (enum)	Yes	The current state of the assembly process for this node. • Allocating: Allocating resources for the node is in progress. Next state can be Allocated or Failed. • Allocated: Node resources have been allocated, but assembly was not started yet after `ComposedNode.Assemble` action state will progress to Assembling. • Assembling: Assembly process initiated, but not finished yet. When the assembly is done, it will change into Assembled. • Assembled: Node successfully assembled. • Failed: Allocation or assembly process failed, or in runtime, one of composing components was removed or transitioned in error state.

<table>
<tr><th>Name</th><th colspan="3">Composed Node</th></tr>
<tr><th>Type URI</th><th colspan="3">/redfish/v1/Nodes/{nodeID}</th></tr>
<tr><th>Attribute</th><th>Type</th><th>Mandatory</th><th>Description</th></tr>
<tr><td>Boot</td><td>Object</td><td>No</td><td>
<table>
<tr><th>Name</th><th>Type</th><th>Required</th><th>Description</th></tr>
<tr><td>BootSourceOverrideEnabled</td><td>String, Null</td><td>No</td><td>State of the Boot Source Override feature.
Proper values:
"Disabled"
"Once"<br>"Continuous"</td></tr>
<tr><td>BootSourceOverrideTarget</td><td>String, Null</td><td>No</td><td>The current boot source to be used at next boot instead of the normal boot device if BootSourceOverrideEnabled is true.</td></tr>
<tr><td>BootSourceOverrideTarget@Redfish.AllowableValues</td><td>Array</td><td>No</td><td>An array of supported boot sources.</td></tr>
<tr><td>BootSourceOverrideMode</td><td>String, Null</td><td>No</td><td>The BIOS Boot Mode (either Legacy or UEFI) to be used when BootSourceOverrideTarget boot source is booted from.</td></tr>
<tr><td>BootSourceOverrideMode@Redfish.AllowableValues</td><td>Array</td><td>No</td><td>An array of supported boot modes.</td></tr>
</table>
</td></tr>
<tr><td>Oem</td><td>Object, Null</td><td>No</td><td>OEM defined object</td></tr>
<tr><td>ClearTPMOnDelete</td><td>Boolean</td><td>No</td><td>This is used to specify if the TPM module should be cleared on composed node DELETE request. If not specified in the allocation request, it is assumed that it is set to "true".</td></tr>
<tr><td>ClearOptaneDCPersistentMemoryOnDelete</td><td>Boolean</td><td>No</td><td>This property is used to specify if Intel® Optane™ DC Persistent Memory should be erased on ComposedNode DELETE request. If not specified in the allocation request, it is assumed that it is set to "true".</td></tr>
<tr><td>PersistentMemoryOperationOnDelete</td><td>String (enum)</td><td>No</td><td>This property is used to specify what operation should be performed on Intel® Optane™ DC Persistent Memory on ComposedNode DELETE Request:
• PreserveConfiguration
• SecureErase
• OverwritePCD</td></tr>
<tr><td>Links</td><td>Object</td><td>No</td><td>Link section:
<table>
<tr><th>Name</th><th>Type</th><th>Required</th><th>Description</th></tr>
<tr><td>ComputerSystem</td><td>Object, null</td><td>Yes</td><td>Reference to ComputerSystem resource used to compose this node.</td></tr>
<tr><td>Processors</td><td>Array</td><td>No</td><td>An array of references to Processor resources.</td></tr>
<tr><td>Memory</td><td>Array</td><td>No</td><td>An array of references to Memory resources.</td></tr>
<tr><td>Storage</td><td>Array</td><td>No</td><td>An array of references to the storage resources used by this composed node.</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Name</th><th colspan="3">Composed Node</th></tr>
<tr><td><b>Type URI</b></td><td colspan="3"><b>/redfish/v1/Nodes/{nodeID}</b></td></tr>
<tr><td><b>Attribute</b></td><td><b>Type</b></td><td><b>Mandatory</b></td><td><b>Description</b></td></tr>
<tr><td></td><td></td><td></td><td><table><tr><td>EthernetInterfaces</td><td>Array</td><td>No</td><td>An array of links to Ethernet Interface collection associated with this Composed Node.</td></tr></table></td></tr>
<tr><td>Actions</td><td>Object</td><td>Yes</td><td>Actions available for this node:
• <b>Reset</b> action with the 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
– "GracefulShutdown" – initiate a soft-shutdown of OS via ACPI
• <b>Assemble</b> action: does not consume any parameters. The second step of creating a composed node (after Allocate Action on Nodes collection). That action will assemble the logical node – initiate ComposedNodeState change from the Allocated state into Assembling state
<b><i>Note:</i></b> After assembly, the PowerState of the composed node is not changed, and it will match the power state of the node's computer system.
• <b>AttachResource</b> – this action allows attaching a pooled resource like remote drive, FPGA, NIC or existing endpoint etc. to this composed node. Action can be performed when Composed Node is Assembled or Failed. Requires below parameter:
– Resource – link to the device from the attachable device pool
– Protocol - defines protocol (fabric) used for connecting the device if multiple protocols are supported by the device.
• <b>DetachResource</b> – action used to detach already connected device. This action can be used to preserve the user's data on a drive/volume (otherwise, it will be deleted on the Node's deletion). Action can be performed when Composed Node is Assembled or Failed. It takes one argument:
– Resource – link to an already connected device or endpoint, which needs to be detached.
• <b>ForceDelete</b> – This action executes the same steps as the DELETE operation. The difference is that the action proceeds even if any DELETE steps fail.</td></tr>
</table>

Figure 2. ComposedNodeState State Diagram



5.12.1 Operations

5.12.1.1 GET

Request:

```
GET /redfish/v1/Nodes/{nodeID}
Content-Type: application/json
```

Response:

```
{
    "@odata.context": "/redfish/v1/$metadata#Nodes/Members/$entity",
    "@odata.id": "/redfish/v1/Nodes/Node1",
    "@odata.type": "#ComposedNode.v1_1_0.ComposedNode",
    "Id": "Node1",
    "Name": "Composed Node",
    "Description": "Node #1",
    "UUID": "00000000-0000-0000-0000-000000000000 - the same as Computer System",
    "PowerState": "On",
    "Status": {
        "State": "Enabled",
        "Health": "OK",
        "HealthRollup": "OK"
    },
    "ComposedNodeState": "Allocated",
    "Boot": {
        "BootSourceOverrideEnabled": "Disabled",
        "BootSourceOverrideTarget": "None",
        "BootSourceOverrideTarget@Redfish.AllowableValues": [
            "None",
            "Pxe",
            "Hdd",
            "RemoteDrive"
        ],
        "BootSourceOverrideMode": "Legacy",
        "BootSourceOverrideMode@Redfish.AllowableValues":  ["Legacy", "UEFI"]
    },
    "Oem": {},
    "ClearTPMOnDelete": true,
    "ClearOptaneDCPersistentMemoryOnDelete": true,
    "Links": {
        "ComputerSystem": {
            "@odata.id": "/redfish/v1/Systems/System1"
        },
        "Processors": [
            {
                "@odata.id": "/redfish/v1/Systems/System1/Processors/CPU1"
            }
        ],
        "Memory": [
            {
                "@odata.id": "/redfish/v1/Systems/System1/Memory/Dimm1"
            }
        ],
        "EthernetInterfaces": [
            {
                "@odata.id": "/redfish/v1/Systems/System1/EthernetInterfaces/LAN1"
            }
        ],
        "Storage": [
            {
                "@odata.id": "/redfish/v1/Chassis/Blade1/Drives/1"
            },
            {
                "@odata.id": "/redfish/v1/StorageServices/NVMeoE1/Volumes/1"
            }
        ],
        "Oem": {}
    },
    "Actions": {
        "@odata.type": "#ComposedNode.v1_1_0.Actions",
        "#ComposedNode.Reset": {
            "target": "/redfish/v1/Nodes/Node1/Actions/ComposedNode.Reset",
            "@Redfish.ActionInfo":
            "/redfish/v1/Nodes/Node1/Actions/ComposedNodeResetActionInfo"
        },
        "#ComposedNode.Assemble": {
            "target": "/redfish/v1/Nodes/Node1/Actions/ComposedNode.Assemble"
        },
        "#ComposedNode.AttachResource": {
            "target": "/redfish/v1/Nodes/Node1/Actions/ComposedNode.AttachResource",
            "@Redfish.ActionInfo":
              "/redfish/v1/Nodes/Node1/Actions/AttachResourceActionInfo"
        },
        "#ComposedNode.DetachResource": {
```

```
      "target": "/redfish/v1/Nodes/Node1/Actions/ComposedNode.DetachResource",
      "@Redfish.ActionInfo":
        "/redfish/v1/Nodes/Node1/Actions/DetachResourceActionInfo"
    },
    "#ComposedNode.ForceDelete": {
      "target": "/redfish/v1/Nodes/Node1/Actions/ComposedNode.ForceDelete"
    }
  }
}
```

5.12.1.2 PUT

Operation not allowed on this resource.

5.12.1.3 PATCH

Table 20 lists the properties that can be updated by the PATCH operation.

Table 20. Properties Updated by PATCH Operation

Attribute	Type	Required	Description
`Boot`	Object	No	Boot override properties, details in Table 21
`ClearTPMOnDelete`	Boolean	No	This is used to specify if the TPM module should be cleared on composed node `DELETE` request.
`ClearOptaneDCPersistentMemoryOnDelete`	Boolean	No	This property is used to specify if Intel® Optane™ DC Persistent Memory should be erased on `ComposedNode DELETE` request.

Table 21 describes `"Boot"` properties that can be patched:

Table 21. Boot Override Update Properties

Attribute	Type	Required	Description
`BootSourceOverrideEnabled`	String	No	Describes the state of the Boot Source Override feature. Allowed values: `"Disabled"` - The system will boot as normal `"Once"` - On its next boot cycle, the system will boot (one time) to the `BootSourceOverrideTarget` `"Continuous"` - The system will boot to the target specified in the `BootSourceOverrideTarget` until this property is set to Disabled
`BootSourceOverrideTarget`	String	No	The current boot source to be used at next boot instead of the normal boot device if `BootSourceOverrideEnabled` is true. Supported values: `"None"` - Boot from the normal boot device `"Pxe"` - Boot from the Pre-Boot EXecution (PXE) environment `"Hdd"` - Boot from a hard drive `"RemoteDrive"` – Boot from the remotely attached drive (iSCSI)
`BootSourceOverrideMode`	String	No	The BIOS Boot Mode (either Legacy or UEFI) to be used when `BootSourceOverrideTarget` boot source is booted from: `"Legacy"` - The system will boot in non-UEFI boot mode to the Boot Source Override Target `"UEFI"` - The system will boot in UEFI boot mode to the Boot Source Override Target

Note: Boot property is representing only the override values, not current boot source configured on the system. To make sure that the correct boot source/mode will be applied, it is recommended to send PATCH to boot property after node assembly, before powering it on.

Request:

```
PATCH /redfish/v1/Nodes/Node1
Content-Type: application/json
{
        "Boot": {
                 "BootSourceOverrideEnabled": "Once",
                 "BootSourceOverrideTarget": "Pxe",
                 "BootSourceOverrideMode": "Legacy"
        }
}
```

Response:

```
HTTP/1.1 204 No Content
```

Or:

```
HTTP/1.1 200 OK
```

with full resource representation.

5.12.1.4 POST

5.12.1.4.1 Reset Action

Request:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.Reset
Content-Type: application/json
{
        "ResetType": "On"
}
```

Response:

```
HTTP/1.1 204 No Content
```

5.12.1.4.2 Assemble action

Request:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.Assemble
```

Response:

```
HTTP/1.1 204 No Content
```

5.12.1.4.3 AttachResource action – specific device

Attaching specific device to existing `ComposedNode`.

Request:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.AttachResource
Content-Type: application/json
{
    "Resource": {
        "@odata.id": "/redfish/v1/StorageServices/NVMeoE1/Volumes/1"
    }
}
```

Response:

```
HTTP/1.1 204 No Content
```

5.12.1.4.4 AttachResource action – specifying Protocol

Attaching the specific device to the existing ComposedNode with protocol specified.

Request:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.AttachResource
Content-Type: application/json
{
    "Resource": {
        "@odata.id": "/redfish/v1/StorageServices/NVMeoE1/Volumes/1"
    },
    "Protocol": "NVMeOverFabrics"
}
```

Response:

```
HTTP/1.1 204 No Content
```

5.12.1.4.5 DetachResource Action

Action used to detach specified resource from ComposedNode. Resource attribute can point to the Drive, Volume, Processor, Endpoint etc.

This action will trigger the following operations:

- If the resource is a PCIe* Drive connected via a PCIe switch, the associated Target endpoint is deleted. Also, the Drive's property DriveErased is set to *false.*
- If the resource is an FPGA processor connected via a PCIe switch or via Ethernet, the associated Target endpoint is deleted. Also, the FPGA's property Erased is set to false.
- If the resource is a remotely attached Volume (iSCSI, NVMeOverFabrics), or an FPGA connected over PCIe*, the associated Target endpoint is deleted.
- If the resource is an Endpoint with a PCIe* Drive as it's ConnectedEntity, then the Drive's property DriveErased is set to ***false***.
- If the resource is an Endpoint with an FPGA processor as it's ConnectedEntity, then the FPGA's property Erased is set to false.

Request:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.DetachResource
Content-Type: application/json
{
    "Resource": {
       "@odata.id": "/redfish/v1/StorageServices/NVMeoE1/Volumes/1"
    }
}
```

Response:

```
HTTP/1.1 204 No Content
```

5.12.1.5 ForceDelete action

This action is used to remove a ComposedNode. This action executes the same steps as the DELETE. However, if any DELETE steps fail, the deletion proceeds and all references to the ComposedNode are removed. Manual cleanup of the Node's resources may be required.

Request:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.ForceDelete
```

Response:

```
HTTP/1.1 204 No Content
```

5.12.1.6 DELETE

`DELETE` operation performs disassembly of a `ComposedNode` and performs deallocation (removes component reservations and puts them back to the resource pool). The following disassembly steps are performed:

1. If the Node Composer configuration variable `disassembly.node.forceOff` is set to **true**, a Reset request of type `ForceOff` is sent to the Computer System.

   To set this variable, follow the steps in Section 3.1, *Configuring Properties for Spring Boot-based Applications* in the PODM User Guide, refer to Table 2.
2. All VLANs (except for reserved ones – refer to Section 7.1.8.1, Allocation with VLAN Requirements) are removed from Ethernet switch ports associated with the Computer System's Ethernet interfaces.
3. All PCIe* devices connected to the Node through the PCIe switch, are detached. All Drives attached through the PCIe switch are securely erased and their property `DriveErased` is set to true. Also, all FPGAs attached through the PCIe switch are securely erased and their property `Erased` is set to true. All FPGA processors attached over Ethernet are detached, securley erased, and their property `Erased` is set to **True**.
4. All remotely attached Volumes (iSCSI, NVMe-over-Fabrics) are disconnected, securely erased, and deleted.
5. All Endpoints and Zones associated with the Node are deleted.
6. If a Computer System used by a Composed Node has any Intel® Optane™ DC Persistent Memory Modules and `ClearOptaneDCPersistentMemoryOnDelete` property is set to **true**, the PODM will issue an `EraseOptaneDCPersistentMemory` action with the `ResetConfiguration` parameter set to **true**.
7. If a Computer System used by a Composed Node has any TPM module in the Enabled state, and the `ClearTPMOnDelete` property is set to **true**, the PODM will issue a `ChangeTPMState` action with the `ClearOwnership` parameter set to **true** to clear the TPM ownership.

Note: If any of the preceding steps fail, the node `DELETE` operation returns an *error,* and no further deletion steps are attempted.

Note: If the TPM is not present, or all of the TPM modules are in a "`Disabled state`", the TPM *will not be cleared*.

Request:

```
DELETE /redfish/v1/Nodes/1
```

Response:

```
HTTP/1.1 204 No Content
```

5.13 ActionInfo

`ActionInfo` describes the parameters and other information necessary to perform a Redfish Action to a particular Action target. As parameter support may differ between implementations, and even among instances of a resource, this data can be used to ensure Action requests from applications contain supported parameters.

5.13.1 Operations

5.13.1.1 GET (ComposedNode::AttachResource ActionInfo)

Request:

```
GET /redfish/v1/Nodes/Node1/Action/AttachResourceActionInfo
Content-Type: application/json
```

Response:

```
{
  "@odata.type": "#ActionInfo.v1_0_0.ActionInfo",
  "@odata.context": "/redfish/v1/$metadata#ActionInfo.ActionInfo",
  "@odata.id": "/redfish/v1/Nodes/Node1/Actions/AttachResourceActionInfo",
  "Id": "AttachResourceActionInfo",
  "Name": "Attach Resource ActionInfo",
  "Description": null,
  "Parameters": [
    {
      "Name": "Resource",
      "Required": true,
      "DataType": "Object",
      "ObjectDataType": "#Resource.Resource",
      "AllowableValues": [
        {"@odata.id": "/redfish/v1/StorageServices/NVMeoE1/Volumes/10"},
        {"@odata.id": "/redfish/v1/Fabrics/iSCSI/Endpoints/5"},
        {"@odata.id": "/redfish/v1/Fabrics/NVMe/Endpoints/8"}
      ]
    },
    {
      "Name": "Protocol",
      "Required": false,
      "DataType": "String",
      "AllowableValues": [ "iSCSI", "NVMeOverFabrics"]
    }
  ],
  "Oem": {}
}
```

5.13.1.2 GET (ComposedNode::DetachResource ActionInfo)

Request:

```
GET /redfish/v1/Nodes/Node1/Actions/DetachResourceActionInfo
Content-Type: application/json
```

Response:

```
{
  "@odata.type": "#ActionInfo.v1_0_0.ActionInfo",
  "@odata.context": "/redfish/v1/$metadata#ActionInfo.ActionInfo",
  "@odata.id": "/redfish/v1/Nodes/Node1/Actions/DetachResourceActionInfo",
  "Id": "DetachResourceActionInfo",
  "Name": "Detach Resource ActionInfo",
  "Description": null,
  "Parameters": [
    {
      "Name": "Resource",
      "Required": true,
      "DataType": "Object",
      "ObjectDataType": "#Resource.Resource",
      "AllowableValues": [
        {"@odata.id": "/redfish/v1/StorageServices/NVMeoE1/Volumes/1"}
      ]
    }
  ],
  "Oem": {},
}
```

5.13.1.3 GET (ComposedNode::Reset ActionInfo)

Request:

```
GET /redfish/v1/Nodes/Node1/Actions/ResetActionInfo
Content-Type: application/json
```

Response:

```
{
  "@odata.type": "#ActionInfo.v1_0_0.ActionInfo",
  "@odata.context": "/redfish/v1/$metadata#ActionInfo.ActionInfo",
  "@odata.id": "/redfish/v1/Nodes/Node1/Actions/ResetActionInfo",
  "Id": "ResetActionInfo",
  "Name": "Reset ActionInfo",
  "Description": null,
  "Parameters": [
    {
      "Name": "Resource",
      "Required": true,
      "DataType": "String",
      "AllowableValues": [
                        "On",
                        "ForceOff",
                        "GracefulShutdown",
                        "GracefulRestart",
                        "ForceRestart",
                        "Nmi",
                        "ForceOn",
                        "PushPowerButton"
                 ]
    }
  ],
  "Oem": {},
}
```

5.13.1.4 PUT

Operation not allowed on this resource.

5.13.1.5 PATCH

Operation not allowed on this resource.

5.13.1.6 POST

Operation not allowed on this resource.

5.13.1.7 DELETE

Operation not allowed on this resource.

5.14 PSME, RMM and Storage Services Resources

The PODM supports the PSME, RMM, and Storage Services resources. Table 22 describes which resources and their operations are included as a part of the PODM API. For additional details, refer to *Intel® Rack Scale Design (Intel® RSD) Pooled System Management Engine (PSME) Representational State Transfer (REST) API Specification Software v2.4* and *Intel® Rack Scale Design (Intel® RSD) Storage Services API Specification Software v2.4, and Intel® Rack Scale Design (Intel® RSD) Rack Management Module (RMM) Representational State Transfer (REST) API Specification Software v2.4*. Refer to Table 2.

Table 22. PODM API—Resources and Supported Operations

Resource Name	Supported Operations				
	GET	PATCH	POST	DELETE	Actions
`Computer System`	X	X	-	-	X
`ComputerSystemMetrics`	-	-	-	-	-
`Processor`	X	-	-	-	-
`ProcessorMetrics`	X	-	-	-	-
`Memory`	X	-	-	-	-
`MemoryMetrics`	X	-	-	-	-
`Storage`	X	-	-	-	-
`Drive`	X	X	-	X	X
`VLAN`	X	-	X	X	
`Network Protocol`	X	-	-	-	-
`Ethernet Switch`	X	X	-	-	-
`Ethernet Switch Port`	X	X	X	X	-
`Ethernet Switch ACL`	X	X	X	X	X
`Ethernet Switch ACL rules`	X	X	X	X	-
`Ethernet Switch Port static MACs`	X	X	X	X	-
`Fabric`	X				-
`Zone`	X	X	X	X	-
`PCIe Device`	X	X	-	-	-
`Fabric Switch`	X	-	-	-	-
`PCIe Port`	X	-	-	-	-
`Fabric Port Metrics`	-	-	-	-	-
`PCIe Function`	X	-	-	-	-
`Endpoint`	X	X	X	X	-
`Storage Service`	X	-	-	-	-
`Storage Pool`	X	-	-	X	-
`Volume`	X	X	X	X	X
`Volume Metrics`	X	-	-	-	-
`NetworkInterface`	X	-	-	-	-

Resource Name	Supported Operations				
	GET	PATCH	POST	DELETE	Actions
NetworkDeviceFunction	X	X	-	-	-
MetricDefinition	X	-	-	-	-
Thermal	X	-	-	-	-
Power	X	-	-	-	-
TelemetryService	X	-	-	-	-
MetricDefinitionCollection	X	-	-	-	-
MetricDefinition	X	-	-	-	-
MessageRegistryFileCollection	X	-	-	-	-
MessageRegistryFile	X	-	-	-	-

5.15 PODM Required resources

The PODM service is required to present:

- One Manager resource of type Service and one Chassis resource of type Pod. These two resources represent, respectively, the PODM service and all of its resources (its pod).
- The ComposedNodeCollection as well as all existing ComposedNodes.
- The AccountService with at least one ManagerAccount that can be used to authenticate to the PODM.
- The SessionService with SessionCollection for creating authentication tokens.
- The EventService.

The resources on the AccountService or SessionService are not aggregated from any underlying service. They represent the authentication to the PODM service itself. Similarly, the EventService contains destinations for events generated by the PODM service.

Table 23 lists the resources required of PODM API in details.

Table 23. PODM API—Resources and Supported Operations

Resource Name	Supported Operations				
	GET	PATCH	POST	DELETE	Actions
ComposedNodeCollection	X	-	X	-	-
ComposedNode	X	-	-	X	X
ManagerCollection	X	-	X	-	-
Manager	X	-	-	-	-
ChassisCollection	X	-	-	-	-
Chassis	X	-	-	-	-
EventService	X	-	-	-	-
EventDestinationCollection	X	-	X	-	-
EventDestination	X	-	-	X	-
ManagerAccountCollection	X	-	-	-	-
ManagerAccount	X	-	-	-	-
RoleCollection	X	-	-	-	-
Role	X	-	-	-	-
AccountService	X	-	-	-	-
SessionCollection	X	-	X		-
Session	X	-	-	X	-

Resource Name	Supported Operations				
	GET	PATCH	POST	DELETE	Actions
`SessionService`	X	-	-	-	-

§

6.0 Common Property Description

6.1 Status

Table 24. Status Attributes

Attribute	Type	Nullable	Description
`State`	String	Yes	Indicates the known state of the resource, as if it is enabled. Allowed values: Refer to Section 5.2, Protocol Version.
`Health`	String	Yes	Represents the health state of this resource in the absence of its dependent resources. Allowed values: Refer to Section 5.3, Odata Service Document.
`HealthRollup`	String	Yes	Represents the overall health state from the view of this resource. Allowed values: Refer to Section 5.3, Odata Service Document.

6.2 Status -> State

- Enabled: This function or resource has been enabled
- Disabled: This function or resource has been disabled
- StandbyOffline: This function or resource is enabled, but awaiting an external action to activate it
- StandbySpare: This function or resource is part of a redundancy set and is awaiting a failover or other external action to activate it
- InTest: This function or resource is under test
- Starting: This function or resource is starting
- Absent: This function or resource is not installed
- UnavailableOffline: This function or resource is present but cannot be used
- Deferring: The element will not process any commands but will queue new requests
- Quiesced: The element is enabled but only processes a restricted set of commands
- Updating: The element is updating and may be unavailable or degraded

6.3 Status -> Health

- OK: Normal
- Warning: A condition exists that requires attention
- Critical: A critical condition exists that requires immediate attention

6.4 ComputerSystem.Reset

- `On`: Turn the system on
- `ForceOff`: Turn the system off immediately (nongraceful) 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 nonmaskable interrupt to cause an immediate system halt
- `ForceOn`: Turn the system on immediately
- `PushPowerButton`: Simulate the pressing of the physical power button on this system
- `GracefulShutdown`: Perform a graceful system shutdown and power off

6.5 BootSourceOverrideTarget/Supported

- None: Boot from the normal boot device
- Pxe: Boot from the preboot execution (PXE) environment
- Floppy: Boot from the floppy disk drive
- Cd: Boot from the CD/DVD disc
- Usb: Boot from a USB device as specified by the system BIOS
- Hdd: Boot from a hard drive
- `BiosSetup` - Boot to the BIOS Setup Utility
- Utilities: Boot the manufacturer's Utilities programs
- Diags: Boot the manufacturer's Diagnostics program
- `UefiShell`: Boot to the UEFI Shell
- `UefiTarget`: Boot to the UEFI Device specified in the `UefiTargetBootSourceOverride` property
- `SDCard`: Boot from an SD Card
- `UefiHttp`: Boot from a UEFI HTTP network location
- `RemoteDrive`: Boot from a remote drive (e.g., iSCSI)

§

7.0 Composed Node Additional Info

7.1 Creating New Composed Node – Explanation

This section explains how to create a Composed Node that satisfies a particular set of requirements using the PODM REST API.

7.1.1 Creating Composed Node Using JSON Template

To create a Composed Node using PODM REST API it is necessary to create a JSON template that describes the requested resources. It needs to be supplied to the PODM by performing an HTTP `POST` request using the Composed Node Collection Action URI located at `"/redfish/v1/Nodes/Actions/Allocate"` on the PODM service.

The JSON template may contain various details of resources used in the Composed Node. All JSON template elements are optional, but each requirement should be coherent itself. It is possible to supply the PODM with a JSON template containing no specific requirements (e.g. { } – a pair of empty curly braces in HTTP request body) thus allowing the PODM to propose a Composed Node containing resources chosen arbitrarily by the PODM.

7.1.2 Specifying Requirements for a Composed Node

The JSON template contains requirements for a single Composed Node. Basic customization covers setting a `"Name"` and `"Description"` of such System (both being of type *String*). As `"Name"` parameter is required by Redfish for all resources if it is not supported the PODM will use the default name. The example shown below will allocate a single Composed Node with the requested name and description:

```
{
        "Name": "Customized Composed Node name",
        "Description": "Description of a customized Composed Node."
}
```

The JSON template may contain requirements for Processors, Memory, Remote Drives, Local Drives, and Ethernet Interfaces. To specify requirements for those resources, a proper section must appear in JSON template.

7.1.3 General Assumptions for Allocation

Requirements are treated as a minimal required value, so the resulting Composed Node may have better parameters than requested. Composed Node customization and resource customization sections described below can be used jointly.

Each resource type description has an associated table, which contains details about specific requirements. **Key** is the JSON object field. **JSON type** contains data type as defined at *https://www.json.org/*. **Allowed values** contain additional restrictions to JSON type or hints (e.g., for enumerations or *Boolean* values). **Nullable** indicates if a *null* value can be passed for the specified key — the **Notes, limitations** column, where present, provides additional hints about specific requirements.

7.1.3.1 Location Requirements

Processor, Memory, Local Drive, and Ethernet interface sections may contain Resource and Chassis objects. The Resource must contain the PODM URI (presented as `"@odata.id"`) of the discovered resource (Processor's URI in Processor section, URI to Memory resource in the Memory section and so on). The Chassis must contain the PODM URI of the discovered Chassis in which applicable resources will be searched for.

7.1.3.2 Allocation Limitations

Due to the limitations of the PODM service, resource availability for allocation is only guaranteed after at least one full cycle of the discovery process after the resource has changed. This interval between the end of the previous discovery and start of next can be configured using the procedures in Section 3.0, *Intel® RSD PODM User Guide*, Table 2. The default value for the interval is 60 seconds.

Note: An “Allocate request“ using a resource may fail if the resource appeared, was modified, or released to the pool of available resources since the last discovery process.

7.1.4 Specifying Processor Requirements

The JSON template may contain requirements for multiple Processors. The example below specifies requirements for a single Processor to be used in the Composed Node.

```
{
        "Processors": [{
                  "@odata.type": "AllocationComposedNodeRequest.v1_1_0.Processor",
                  "Model": "Multi-Core Intel(R) Xeon(R) processor 7xxx Series",
                  "TotalCores": 2,
                  "AchievableSpeedMHz": 3700,
                  "InstructionSet": "x86-64",
                  "Oem" : {
                    "Intel_RackScale" : {
                      "@odata.type":
"AllocationComposedNodeRequest.v1_1_0.ProcessorExtensions",
                          "Brand" : "E5",
                          "Capabilities": [ "sse" ]
         }
           },
                  "Resource": {
                         "@odata.id": "/redfish/v1/Systems/1/Processors/1"
                  },
                  "Chassis": {
                         "@odata.id": "/redfish/v1/Chassis/1"
                  },
                  "ProcessorType": "CPU",
                  "Connectivity": ["Local", "RemotePCIe"]
        }]
}
```

Table 25. Processor Requirements

Attribute	Type	Allowed values	Nullable	Description
`Model`	String	-	Yes	A string representing Processor model.
`TotalCores`	Number	-	Yes	Positive integer value expected
`AchievableSpeedMHz`	Number	-	Yes	Positive integer value expected
`InstructionSet`	String	"x86", "x86-64", "IA-64", "ARM-A32", "ARM-A64", "MIPS32", "MIPS64", "OEM"	Yes	One of allowed, enumerated values
`Oem`	Object	-	Yes	-

Attribute	Type	Allowed values	Nullable	Description
Oem -> Intel_RackScale -> Brand	String	"E3", "E5", "E7", "X3", "X5", "X7", "I3", "I5", "I7", "Silver", "Gold", "Platinum", "Unknown"	Yes	One of allowed, enumerated values
Oem -> Intel_RackScale -> Capabilities	Array	CPU capabilities string	Yes	List of processor capabilities (like "sse")
Resource	Object	The exact location of a single Processor.	Yes	Refer to Section 7.1.3.1, Location Requirements.
Chassis	Object	The exact location of a single chassis.	Yes	Refer to Section 7.1.3.1, Location Requirements.
ProcessorType	String	"CPU", "FPGA", "GPU", "DSP", "Accelerator", "OEM"	Yes	One of allowed enumerated values
Connectivity	Array of String	[("Local"| "Ethernet"| "RemotePCIe")*]	Yes	List specifying how the processor should be connected to the node. If more than one item is given, the service shall use the following priorities: 1. Local 2. RemotePCIe 3. Ethernet If this property is not specified, the service shall use the same priorities.

The template can also provide the requirement for total processor cores available in the composed node:

```
        " TotalSystemCoreCount ": 32
```

Allocation assumptions:

- Which Processors will meet supplied requirements?
  - located on the same computer system as other resources
  - with exact match on Model
  - with exact match on Brand
  - with at least TotalCores
  - with at least AchievableSpeedMHz
  - with exact match on InstructionSet
  - with exact match on ProcessorType
  - with superset of processor capabilities specified Capabilities array
  - will have a match on one of the methods of connection specified in Connectivity
- If a computer system contains processors with cores that number at least TotalSystemCoreCount, it meets the requirements.

7.1.4.1 Pooled FPGA over PCIe

Pooled FPGA functionality provides a capability of connecting FPGA processors to compute host through the PCIe fabric. The PCIe fabric module contains a PSME, which enables the PODM to assign FPGA to the compute modules. If the PCIe fabric module is enabled in the rack, then the PODM has to provide support to PSME API that covers the PCIe fabric module.

Example request that allocates the node with the FPGA processors:

```
{
        "Processors": [{
                "@odata.type": "AllocationComposedNodeRequest.v1_1_0.Processor",
                "ProcessorType": "FPGA",
                "Chassis": {
                        "@odata.id": "/redfish/v1/Chassis/PCIeSwitchChassis"
                },
                "Connectivity": ["RemotePCIe"]
        }]
}
```

After node allocation and assembly (in "`Assembled`" and "`Failed`" `ComposedNodeState`), the user is able to attach and remove FPGA processors using one of the action: `AttachResource, DetachResource.`

Example attaching FPGA processor to existing node:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.AttachResource
Content-Type: application/json
{
    "Resource": {
       "@odata.id":"/redfish/v1/Chassis/PCIeSwitchChassis/Processors/PCIe.fpga.1"
    }
}
```

Example removing FPGA processor from existing node:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.DetachResource
Content-Type: application/json
{
    "Resource": {
        "@odata.id":"/redfish/v1/Chassis/PCIeSwitchChassis/Processors/PCIe.fpga.1"
    }
}
```

7.1.4.2 Remote FPGA Processor Lifecycle

Given an FPGA processor connected to a Composed Node over Ethernet or over a PCIe Switch, if the FGPA is Detached from the Node using the Detach action, then it will not be erased and it's property `"Erased"` will change to `‘false’`.". Such a processor will not be available in the pool of resources for node composition, though it may still be explicitly selected by using the `"Resource" ProcessorRequirement`.

Alternatively, calling `SecureErase` action on the FPGA resource will return it to the pool.

If a `ComposedNode` with an FPGA connected over a PCIe Switch or over Ethernet is `DELETE-d`, the FPGA is securely erased, it's `"Erased"` property is changed to **True**, and it's returned to the pool of resources for composition.

7.1.5 Specifying Memory Requirements

This section will be updated in the final version of the document.

The JSON template may contain requirements for multiple Memory Modules. The example below specifies requirements for a single Memory Module used in the Composed Node.

```
{
        "Memory": [{
                "@odata.type": "AllocationComposedNodeRequest.v1_1_0.Memory",
                "CapacityMiB": 16000,
                "MemoryType": "DRAM",
                "MemoryDeviceType": "DDR3",
                "SpeedMHz": 1600,
                "Manufacturer": "Intel",
                "DataWidthBits": 64,
                "Resource": {
                       "@odata.id": "/redfish/v1/Systems/1/Memory/1"
                },
                "Chassis": {
                       "@odata.id": "/redfish/v1/Chassis/1"
                }
        }]
}
```

Table 26. Memory Module Requirements

Attribute	Type	Allowed values	Nullable	Notes, limitations
CapacityMiB	Number	-	Yes	Positive value expected
MemoryType	String	"DRAM", "NVDIMM_N", "NVDIMM_F", "NVMDIMM_P", "IntelOptane"	Yes	One of allowed, enumerated values
MemoryDeviceType	String	"DDR", "DDR2", "DDR3", "DDR4", "DDR4_SDRAM", "DDR4E_SDRAM", "LPDDR4_SDRAM", "DDR3_SDRAM", "LPDDR3_SDRAM", "DDR2_SDRAM", "DDR2_SDRAM_FB_DIMM", "DDR2_SDRAM_FB_DIMM_PROBE", "DDR_SGRAM", "DDR_SDRAM", "ROM", "SDRAM", "EDO", "FastPageMode", "PipelinedNibble", "Logical"	Yes	One of allowed, enumerated values
SpeedMHz	Number	-	Yes	Positive integer value expected
Manufacturer	String	-	Yes	A string representing Memory Module manufacturer name
DataWidthBits	Number	-	Yes	Positive integer value expected.
Resource	Object	The exact location of a single Memory Module.	Yes	Refer to section 7.1.3.1, Location Requirements.
Chassis	Object	The exact location of a single chassis.	Yes	Refer to section 7.1.3.1, Location Requirements.

The template can also provide a requirement for total memory available in the composed node, without dividing it into memory modules:

```
" TotalSystemMemoryMiB ": 32000
```

Allocation assumptions:

- Which Memory Modules will meet supplied requirements?
  - With at least CapacityMiB
  - located on the same computer system as other resources
  - with exact match on MemoryType
  - with exact match on MemoryDeviceType
  - with at least SpeedMHz
  - with exact match on Manufacturer
  - with at least DataWidthBits

- If a computer system contains Memory Modules of least the same size as the `TotalSystemMemoryMiB`, it will meet requirements.

7.1.6 Specifying Remote Drive Requirements

The JSON template may contain requirements for multiple Remote Drives, but currently supports only one set of requirements. The example below specifies requirements for a single Remote Drive used in the Composed Node.

```
{
        "RemoteDrives": [{
                  "CapacityGiB": 80,
                  "Protocol": "iSCSI",
                  "Master": {
                          "@odata.type": "AllocationComposedNodeRequest.v1_1_0.Master",
                          "Type": "Snapshot",
                          "Resource": {
                                    "@odata.id":
"/redfish/v1/StorageServices/iscsi1/Volumes/1"
                          }
                  },
                  "Resource": null
        }]
}
```

Table 27. Requirements for a Single Remote Drive

Attribute	Type	Allowed Values	Nullable	Description
`CapacityGiB`	Number	-	Yes	Positive value expected, required if Master Drive supplied. Should be at least the size of Logical Drive used as Master Drive.

`Protocol`	String	"iSCSI"
		"NVMeOverFabrics"
			Yes	Defines protocol used to communicate with an attached drive. Currently supported:
				`"iSCSI"`
				`"NVMeOverFabrics"`

`Master`	Object	-	Yes	-
`Master` →`Type`	String	`"Snapshot"`, `"Clone"`	No	One of the allowed, enumerated values. Required if Master Drive supplied
`Master` → `Resource`	Object	-	No	URI of Master volume that should be used for clone or snapshot operation
`Resource`	Object	-	Yes	Reference to a particular volume or already created endpoint.

7.1.6.1 Using existing iSCSI Remote Drive

To use an existing Volume, it is necessary to point to the resource (Volume) directly:

```
{
        "RemoteDrives": [{
                  "Protocol": "iSCSI",
                  "Resource": {
                     "@odata.id": "/redfish/v1/StorageServices/iSCSI/Volumes/1"
                  }
        }]
}
```

7.1.6.2 Using existing NVMe* over Fabrics Remote Drive

To use an existing Volume, it is necessary to point to the resource (Volume) directly:

```
{
  "RemoteDrives": [{
    "Protocol": "NVMeOverFabrics",
     "Resource": {
       "@odata.id": "/redfish/v1/StorageServices/NVMeoE1/Volumes/1"
     }
  }]
}
```

7.1.6.3 Using a Master Drive for fresh Remote Drive Creation

To use a fresh Drive created from Master Drive, it is necessary to:

- Set `CapacityGiB` to define the capacity of the new Remote Drive that is at least of Master Drive's size,
- Set Master → Type to `"Snapshot"` or `"Clone"`
- Set Master → Resource to valid PODM URI of Volume resource used as source Drive.
- Set Protocol to the desired protocol that should be used.

Note: Only iSCSI RemoteDrives can be allocated with specified Master Drive. Allocating with an `NVMeOverFabrics` Clone or Snapshot is not supported.

```
{
  "RemoteDrives": [{
    "CapacityGiB": 80,
    "Protocol": "iSCSI",
    "Master": {
      "Type": "Snapshot",
      "Resource": {
        "@odata.id": "/redfish/v1/StorageServices/iSCSI1/Volumes/102"
      }
    }
  }]
}
```

7.1.6.4 Creating a New Empty Drive

To create a new empty Drive without any data:

- Set `CapacityGiB` to define the capacity of the new Remote Drive,
- Set Protocol to the desired protocol that should be used.

Note: Current iSCSI implementation requires the remote drive to be bootable. For this reason, this functionality is intended to work with the NVMe-oF Volumes only.

```
{
        "RemoteDrives": [{
                 "CapacityGiB": 80,
                 "Protocol": "NVMeOverFabrics",
        }]
}
```

7.1.7 Specifying Local Drive Requirements

The JSON template may contain requirements for multiple Local Drives (represented by Drive resource). The example below specifies requirements for a single Local Drive to be used in the Composed Node.

```
{
        "LocalDrives": [{
                 "CapacityGiB": 100,
                 "Type": "HDD",
                 "MinRPM": 5400,
                 "SerialNumber": "12345678",
                 "Interface": "SATA",
                 "Resource": {
                        "@odata.id": "redfish/v1/Chassis/Blade1/Drives/Disk1"
                 },
                 "Chassis": {
                        "@odata.id": "/redfish/v1/Chassis/Blade1"
                 },
                 "FabricSwitch": false
        }]
}
```

Table 28. Local Drive Requirements

Attribute	Type	Allowed values	Nullable	Description
`CapacityGiB`	Number	-	Yes	Positive value expected

`Type`	String	`"HDD",`
		`"SSD"`
			Yes	One of allowed, enumerated values

`MinRPM`	Number	-	Yes	Positive integer value expected
`SerialNumber`	String	-	Yes	-
`Interface`	String	`"SAS", "SATA", "NVMe"`	Yes	One of allowed, enumerated values
`Resource`	Object	The exact location of a single Device.	Yes	Refer to section 7.1.3.1, Location Requirements
`Chassis`	Object	The exact location of a single Chassis.	Yes	Refer to section 7.1.3.1, Location Requirements.
`FabricSwitch`	Boolean	`"True", "false"`	Yes	Determines if the drive should be connected using the fabric switch (PNC) or directly attached to the computer system.

Allocation assumptions:

- Which Local Drives will meet supplied requirements?
  - drives located on the same computer system as other resources
  - drives with at least CapacityGiB
  - drives with exact match on Type
  - drives with at least MinRPM
  - drives with exact SerialNumber
  - drives with exact Interface.

7.1.7.1 Pooled NVMe* Controller (PNC) Drives

If the PNC is available in the PODM, and there is no system fulfilling Local Drive requirements, PNC drives are attached to the Composed Node from the pool of available the PNC drives.

Note: PNC drives that were detached from Composed Node resource via Detach action are not erased, and their property `"DriveErased"` is changed to false. They will not be available in the pool of the PNC drives ready for node composition. These PNC drives need to be selected using the Resource property. Alternatively, they may be erased by using the action "SecureErase" on drive resource.

Note: PNC drives that were detached from Composed Node resource via DELETE operation on the node are securely erased, and their property `"DriveErased"` is changed to true. Drives with property `"DriveErased"` set to true are available for composition without the need to specify their URI in the Resource property.

Example request that allocates the node with the PNC drive:

```
{
        "LocalDrives": [{
                "CapacityGiB": 100,
                "Type": "SSD",
                "Interface": "NVMe",
                "Chassis": {
                        "@odata.id": "/redfish/v1/Chassis/PCIeSwitchChassis"
                }
        }]
}
```

After node allocation and assembly (in "`Assembled`" and "`Failed`" `ComposedNodeState`), the user is able to attach and remove PNC devices (drives) using the `AttachEndpoint` and `DetachEndpoint` actions:

Example attaching drive to existing node:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.AttachResource
Content-Type: application/json
{
    "Resource": {
       "@odata.id":"/redfish/v1/Chassis/PCIeSwitchChassis/Drives/Disk.Bay.2"
    }
}
```

Example removing drive from existing node:

```
POST /redfish/v1/Nodes/1/Actions/ComposedNode.DetachResource
Content-Type: application/json
{
    "Resource": {
        "@odata.id":"/redfish/v1/Chassis/PCIeSwitchChassis/Drives/Disk.Bay.3"
    }
}
```

7.1.8 Specifying Ethernet Interface Requirements

The JSON template may contain requirements for multiple Ethernet Interfaces. The example below specifies requirements for a single Ethernet Interface used in the Composed Node.

```
{
        "EthernetInterfaces": [{
                "SpeedMbps": 1000,
                "PrimaryVLAN": 100,
                "VLANs": [{
                        "VLANId": 100,
                        "Tagged": false
                }],
                "Resource": {
                        "@odata.id": "/redfish/v1/Systems/1/EthernetInterfaces/1"
                },
                "Chassis": {
                        "@odata.id": "/redfish/v1/Chassis/1"
                }
        }]
}
```

Table 29. Ethernet Interface Requirements

Attribute	Type	Allowed values	Nullable	Description
SpeedMbps	Number	-	Yes	Positive integer value expected
PrimaryVLAN	Number	-	Yes	Positive integer value expected
VLANs	Array[Object]	-	Yes	A null value will be interpreted as the absence of this key. Empty array [ ] will clear all existing VLANs, excluding Reserved VLANs.
VLANs → VLANId	Number	-	No	Positive integer value expected
VLANs → Tagged	Boolean	true, false	No	Boolean value
Resource	Object	The exact location of a single Ethernet Interface.	Yes	Refer to section 7.1.3.1, Location Requirements.
Chassis	Object	The exact location of a single Chassis.	Yes	Refer to section 7.1.3.1, Location Requirements.

Allocation assumptions:

- Which Ethernet Interfaces will meet supplied requirements?
  - located on the same Computer System as other resources
  - with at least SpeedMbps
  - ones that are connected to SwitchPorts (when VLANs section is provided)

7.1.8.1 Allocation with VLAN Requirements

Note: Allocating a node with EthernetInterface requirements may have an immediate effect on the selected node's ethernet connectivity (even before Assemble action is triggered). For more details refer to Section 7.0, Composed Node Additional Info.

7.1.8.2 Reserved VLANs

It is possible to restrict the usage of some VLANs by marking them as reserved.

Reserved VLANs:

- Cannot be used in an Allocate Request. A request containing one of the reserved VLANs in the EthernetInterfaces section will result in an error.
- Reserved VLANs are not deleted during disassembly.

The list of Reserved VLANs can be modified though the Node Composer configuration variable allocation.reserved-vlan-ids. To set this variable, Section 3.1, *Configuring Properties for Spring Boot-based Applications* in the PODM User Guide, refer to Table 2.

The default Reserved VLANs are 1, 170, 4088, 4091 and 4094.

7.1.9 Specifying Security Requirements

The JSON template may contain requirements for the security of Composed Node. The example below specifies the requirements for a Trusted Platform Module (TPM) v2.0 present in the Composed Node.

```
{
        "Security": {
          "@odata.type" : "AllocationComposedNodeRequest.v1_1_0.Security",
          "TpmPresent": true,
          "TpmInterfaceType": "TPM2_0",
          "ClearTPMOnDelete": true,
          "ClearOptaneDCPersistentMemoryOnDelete": true
```

```
            }
}
```

Table 30. Security Requirements

Attribute	Type	Allowed values	Nullable	Description
`TpmPresent`	Boolean	-	Yes	Boolean expected
`TpmInterfaceType`	String	-	Yes	A string value (enum) expected. Possible values for this field are defined in metadata file `ComputerSystem_v1.xml` as a `InterfaceType`
`TxtEnabled`	Boolean	-	Yes	Boolean expected
`ClearTPMOnDelete`	Boolean	-	Yes	Boolean expected
`ClearOptaneDC PersistentMemory OnDelete`	Boolean	-	Yes	This property is used to specify if Intel® Optane™ DC Persistent Memory should be erase upon `ComposedNode DELETE` request.

The PODM will allocate the Composed Node that meets security requirements listed in Section 7.1.8.2, Reserved VLANs

It is possible to restrict the usage of some VLANs by marking them as reserved.

Reserved VLANs:

- Cannot be used in an Allocate Request. A request containing one of the reserved VLANs in the `EthernetInterfaces` section will result in an error.
- Reserved VLANs are not deleted during disassembly.

The list of Reserved VLANs can be modified though the Node Composer configuration variable `allocation.reserved-vlan-ids`. To set this variable, Section 3.1, *Configuring Properties for Spring Boot-based Applications* in the PODM User Guide, refer to Table 2.

The default Reserved VLANs are 1, 170, 4088, 4091 and 4094.

Specifying Security Requirements.

- `TpmPresent` – determine if Composed Node should be equipped with TPM module
- `TpmInterfaceType` – overrides `TpmPresent` parameter (if specified TPM module expected). The system must be equipped with the only defined TPM interface type.
- `TxtEnabled` – determine if Composed Node should have Trusted Execution Technology (TXT) mode enabled.
- `ClearTPMOnDelete` - used to specify if the TPM module should be cleared on composed node `DELETE` request.
- `•ClearOptaneDCPersistentMemoryOnDelete –` used to specify whether Intel® Optane™ DC Persistent Memory should be cleared on `ComposedNode DELETE` request.

7.1.10 Allocation Algorithm

Node composition starts with HTTP `POST` request of JSON template on "`/redfish/v1/Nodes/Actions/Allocate`" Composed Node Collection Action URI on PODM Service. If the JSON template is well-formed, and contains a supported set of requirements, the allocation process begins. Four major scenarios are currently supported:

- Allocating resources for Composed Node to be booted from Local Drive.
- We are allocating resources for Composed Node to be booted from existing iSCSI Remote Drive.

- We are allocating resources for Composed Node to be booted from an iSCSI Remote Drive that needs to be created.
- We are allocating resources for Composed Node with VLAN requirements specified. This scenario is used with one of the other three.

The allocation process is preceded by a general verification of JSON template that checks if requested node can be realized by available resources and consists of:

- Selecting and allocating a Computer System that contains resources matching template requirements for Processors, Memory, Local Drives, and Ethernet Interfaces.
- You are selecting or creating Remote Drive to be used with previously selected Computer System and allocating it.

7.1.10.1 Detailed Process of Selecting and Allocating a Computer System for a Composed Node:

1. Find all Computer Systems that are not yet allocated (not used by any other allocated Composed Node) with State Enabled and Health OK.
2. Filter Computer Systems by specified Resource and Chassis (if supplied in the template)
3. Filter Computer Systems by available Performance Configuration. If `SupportedPerformanceConfigurations` is supplied in the template, a Computer System should have at least one `PerformanceConfiguration` of each type listed in the array.
4. Filter Computer Systems by Processors: return all Computer Systems that contain at least requested quantity of Processors that meet requirements (if supplied in the template):

   a. Exactly matching requested model,
   b. Exactly matching requested brand,
   c. With at least requested number of cores,
   d. With at least requested frequency,
   e. Exactly matching requested instruction set,
   f. Exactly matching requested processor type,
   g. With a superset of requested capabilities,
   h. With a match on one of the requested methods of connection.

5. **Filter Computer Systems by Memory:** return all Computer Systems with at least total requested size of memory located on Memory Modules that each of them meets requirements (if supplied in the template):

   a. The memory of exactly requested type,
   b. With exactly matching DIMM device type,
   c. With at least requested speed MHz,
   d. With exact requested manufacturer,
   e. With at least requested data width bits,
   f. With at least the requested capacity.

6. **Filter Computer Systems by Local Drives:** return all Computer Systems that contain for each requested Drive one distinct Device meeting requirements (if supplied in the template):

   a. With at least requested capacity specified
   b. Exactly matching requested Drive type
   c. With at least requested min RPM
   d. With exact requested serial number
   e. With exact Interface

7. **Filter Computer Systems by Ethernet Interfaces:** return all Computer Systems that contain for each requested Ethernet Interface one distinct Ethernet Interface meeting requirements (if supplied in the template):

   a. With at least requested speed.

   b. If VLANs section is provided, then Computer Systems with Ethernet Interfaces which are not connected with `EthernetSwitchPorts` are filtered out (as described below)

8. A first Computer System from resulting filtered collection is then allocated to be used in the Composed Node.

7.1.10.2 Connection between Computer System Ethernet Interface and Ethernet Switch Port

To enable any particular VLAN usage on the Composed Node, there is a need to map the Ethernet Switch Port and Computer System's Ethernet interface. This mapping is done using the MAC address as an identifier. Fields used for this mapping are:

- `NeighborMAC` on `EthernetSwitchPort` resource
- `MacAddress` on `EthernetInterface` resource

If those two properties contain the same value, the Computer System's Ethernet Interface and Ethernet Switch Port are treated as connected. Only Computer Systems with the Ethernet Interface, connected to Ethernet Switch Ports, could be used in allocation with specified VLANs requirement.

7.1.10.3 Detailed Process of Selecting Remote Drives:

1. If a specific existing Remote Drive is requested by specifying Resource:
   a. The requested Drive will be attached to the Node during the Assemble action. For this purpose, the PODM will select the Storage Service that contains the Remote Drive and create
      1. a target Endpoint representing the Drive.
      2. an initiator Endpoint is representing the ComputerSystem in the Node.
      3. a Zone is containing both endpoints.
2. Otherwise, attempt to create a new Remote Drive matching the requirements during Assemble action. If a Master is specified, select the Storage Service which contains the Master Volume.
   a. If a Snapshot was requested, check if the StoragePool containing the Master Volume has enough capacity for the new Remote Drive. If yes, select this Pool.
   b. If a Clone was requested, check if any of the StoragePools on the Storage Service have enough capacity. for the new Remote Drive. Select the first Pool with available capacity.
   c. Create a new Volume on the StorageService using the requested Master Volume and the selected Pool.
   d. Attach the Volume as in step 1a.
3. If a Master is not specified, attempt to create a new Remote Drive on any StorageService with a Fabric matching the requested RemoteDrive protocol:
   a. For each StorageService with a matching fabric, check if any of the StoragePools have enough capacity for the new Remote Drive. Select first Pool with available capacity.
   b. Create a new Volume on the StorageService using the selected Pool.
   c. Attach the Volume as in step 1a

7.1.10.4 POST-allocation Scenarios

A Composed Node is created as a new REST resource at `/redfish/v1/Nodes/{NodeId}` when a proper Computer System was found and was successfully allocated. The state of Composed Node is set to `"Allocated"`. An `"Allocated"` Composed Node is a PODM proposition that can be either accepted or rejected.

- If accepted, the user has to send an HTTP `POST` request on `ComposedNode.Assemble` action of the proposed Composed Node to assemble it:
  – If no Remote Drive was requested, a Composed Node's state is set to `"Assembled"`.

  - When the Remote Drive is requested, Composed Node remains "Assembling" until Target creation finishes. When the Target is successfully assembled to be used with the Composed Node, the node's state is set to "Assembled".
  - The assembly process does not end with sending power on request, so it is necessary to perform a ComposedNode.Reset action to power on a Composed Node after assembly.
- If rejected, a user can continue sending HTTP POST requests of the JSON template on /redfish/v1/Nodes/Actions/Allocate to create more proposals from which to pick. When finding the right pick, it is recommended to send HTTP DELETE on all rejected proposals of the Composed Nodes to free resources allocated by them.

7.2 Remote Drive lifecycle

Remote drives using NVMeOverFabrics or iSCSI protocols can be attached and detached to/from Composed Node during allocation and removal. In addition, it is possible in runtime using Attach/DetachResource actions.

To make sure user data are safe and not provided to other users, all remote drives used by a Composed Node are erased and deleted when the Composed Node is deleted. If data on a drive should be preserved, the user should detach the resource. Below table summarizes what actions are made on each scenario.

Table 31. RemoteDrive Lifecycle

Action	Effect on Remote Drive
POST Node.DetachResource	Remote Drive not erased Remote Drive not deleted
DELETE node	Remote Drive erased Remote Drive deleted

Note: When the remote drive is detached using the Node.DetachResource action is pointing to the Endpoint, the associated Endpoint will not be deleted by the PODM and will prevent the Volume from being erased and deleted.

§