Document Number: DSP0263

Date: 2013-08-13

Version: 1.1.0a

Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

An Interface for Managing Cloud Infrastructure

Information for Work-in-Progress version:

IMPORTANT: This specification is not a standard. It does not necessarily reflect the views of the DMTF or all of its members. Because this document is a Work in Progress, this specification may still change, perhaps profoundly. This document is available for public review and comment until the stated expiration date.

It expires on: 2014-01-22

Provide any comments through the DMTF Feedback Portal:
http://www.dmtf.org/standards/feedback

Document Type: Specification

Document Status: Work In Progress

Document Language: en-US

CONTENTS

150
151

152 Figures

159

Foreword

The *Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol* specification (DSP0263) was prepared by the DMTF Cloud Management Working Group. It defines a logical model for the management of resources within the Infrastructure as a Service domain.

DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability.

Acknowledgments

The DMTF acknowledges the following individuals for their contributions to this document:

Editors (past and present):

- Davis, Doug – IBM
- Pilz, Gilbert - Oracle
- Andreou, Marios – Red Hat
- Durand, Jacques - Fujitsu

Contributors:

- Ali, Ghazanfar - ZTE Corporation
- Andreou, Marios - Red Hat
- Bankston, Keith - Microsoft Corporation
- Bumpus, Winston - VMware Inc.
- Burkhart, Nathan - Microsoft Corporation
- Carlson, Mark - Oracle
- Carter, Steve - Novell
- Chu, Junsheng - ZTE Corporation
- Cohen, Josh - Microsoft Corporation
- Coleman, Derek - Hewlett-Packard Company
- Crandall, John - Brocade Communications Systems
- Davis, Doug - IBM
- Davis, Jim - WBEM Solutions
- de la Iglesia, Fernando - Telefónica
- Dempo, Hiroshi - NEC Corporation
- Durand, Jacques - Fujitsu
- Edery, Yigal - Microsoft Corporation
- Ericson, George - EMC
- Evans, Colleen - Microsoft Corporation
- Floeren, Norbert - Ericsson AB
- Freund, Robert - Hitachi, Ltd.
- Galán, Fermín - Telefónica
- Gopalan, Krishnan - Microsoft Corporation
- Iwasa, Kazunori - Fujitsu
- Johnson, Mark - IBM
- Khasnabish, Bhumip - ZTE Corporation
- Köper, Dies - Fujitsu
- Kowalski, Vincent - BMC Software
- Krishnaswamy, Ruby - France Telecom Group
- Lamers, Lawrence - VMware Inc.
- Lipton, Paul - CA Technologies
- Livingston, James - NEC Corporation

206 • Lubsey, Vince - Virtustream Inc.
207 • Lutterkort, David - Red Hat
208 • Maciel, Fred - Hitachi, Ltd.
209 • Maier, Andreas - IBM
210 • Malhotra, Ashok - Oracle
211 • Mischkinsky, Jeff - Oracle
212 • Molina, Jesus - Fujitsu
213 • Moscovich, Efraim - CA Technologies
214 • Murray, Bryan - Hewlett-Packard Company
215 • Neely, Steven - Cisco
216 • Ogawa, Ryuichi - NEC Corporation
217 • Parchem, John - Microsoft Corporation
218 • Pardikar, Shishir - Citrix Systems Inc.
219 • Peñalvo, Miguel - Telefónica
220 • Pilz, Gilbert - Oracle
221 • Polo, Alvaro - Telefónica
222 • Ronco, Enrico - Telecom Italia
223 • Rossini, Federico - Telecom Italia
224 • Rutkowski, Matthew - IBM
225 • Rutt, Tom - Fujitsu
226 • Shah, Hemal - Broadcom
227 • Shah, Nihar - Microsoft Corporation
228 • Sill, Alan - Texas Tech University
229 • Song, Zhexuan - Huawei
230 • Waschke, Marvin - CA Technologies
231 • Wells, Eric - Hitachi, Ltd.
232 • Wheeler, Jeff - Huawei
233 • Wiggers, Maarten – Fujitsu
234 • Wilson, Daniel – Ericsson AB
235 • Winkler, Steve - SAP AG
236 • Yu, Jack - Oracle
237 • Zhang, Aaron - Huawei
238 • Zhang, HengLiang – Huawei
239

240

Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

1 Scope

This specification describes the model and protocol for management interactions between a cloud Infrastructure as a Service (IaaS) Provider and the Consumers of an IaaS service. The basic resources of IaaS (machines, storage, and networks) are modeled with the goal of providing Consumer management access to an implementation of IaaS and facilitating portability between cloud implementations that support the specification. This document specifies a Representational State Transfer (REST)-style protocol using HTTP. However, the underlying model is not specific to HTTP, and it is possible to map it to other protocols as well.

CIMI addresses the management of the lifecycle of infrastructure provided by a Provider. CIMI does not extend beyond infrastructure management to the control of the applications and services that the Consumer chooses to run on the infrastructure provided as a service by the Provider. Although CIMI may be to some extent applicable to other cloud service models, such as Platform as a Service ("PaaS") or Storage as a Service ("SaaS"), these uses are outside the design goals of CIMI.

1.1 Document structure

This document defines a model and a RESTful HTTP-based protocol.

The core REST patterns are defined first and, after each resource is defined, any HTTP-specific information for that resource will be specified.

1.2 Document versioning scheme

This document will adhere to the versioning scheme defined in clause 6.3 of DSP4004.

As the specification changes over time certain features might be deprecated. These will be identified in the specification and should not be supported. Each of these deprecated features will be clearly denoted in the clause in which they were previously defined.

1.3 Typographical conventions

This specification uses the following conventions inside tables describing the resource data model:

- Resource names, and any other name that is usable as a type (i.e., names of embedded structures as well as atomic types such as "integer", "string"), are in *italic*.
- Attribute names are in regular font.
- Names that are just placeholders for actual names that may vary with each model instance, are between < > (e.g., <componentTemplate>).

In addition, this specification uses the following syntax to define the serialization of resources:

- Values in *italics* indicate data types instead of literal values.
- Characters are appended to items to indicate cardinality:
  - "?" (0 or 1)
  - "*" (0 or more)

277 – "+" (1 or more)

278 • Vertical bars, "|", denote choice. For example, "a|b" means a choice between "a" and "b".

279 • Parentheses, "(" and ")", are used to indicate the scope of the operators "?", "*", "+" and "|".

280 • Ellipses (i.e., "...") indicate points of extensibility. Note that the lack of an ellipses does not mean
281 no extensibility point exists, rather it is just not explicitly called out - usually for the sake of
282 brevity.

283 2 Normative references

284 The following referenced documents are indispensable for the application of this document. For dated or
285 versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies.

286 DMTF DSP0223, *Generic Operations 1.0*,
287 http://www.dmtf.org/standards/published_documents/DSP0223_1.0.pdf

288 DMTF DSP0243, Distributed Management Task Force, Inc., *Open Virtualization Format Specification 1.1*,
289 http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.pdf

290 DMTF DSP0259, Distributed Management Task Force, Inc., *Cloud Infrastructure Management Interface -*
291 *CIM Model (CIMI-CIM) 0.0.1*, http://members.dmtf.org/apps/org/workgroup/cmwg/download.php/yyyy

292 DMTF DSP1001, *Management Profile Specification Usage Guide 1.1,*
293 http://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf

294 DMTF DSP4004, Distributed Management Task Force, Inc., DMTF Release Process 2.4,
295 http://www.dmtf.org/sites/default/files/standards/documents/DSP4004_2.4.pdf

296 IANA HTTP Header Registry, http://www.iana.org/assignments/message-headers/perm-headers.html

297 IEC 80000-13:2008, International Organization for Standardization, Geneva, Switzerland, *Quantities and*
298 *units – Part 13: Information science and technology*, April 2008,
299 http://www.iso.org/iso/catalogue_detail?csnumber=31898

300 IETF RFC2616, R. Fielding et al, *Hypertext Transfer Protocol -- HTTP/1.1*,
301 http://www.ietf.org/rfc/rfc2616.txt

302 IETF RFC3986, T.Berners-Lee et al, *Uniform Resource Identifiers (URI): Generic Syntax*, August 1998,
303 http://www.ietf.org/rfc/rfc3986.txt

304 IETF RFC4627, D. Crockford, *The application/json Media Type for JavaScript Object Notation (JSON)*,
305 July 2006, http://www.ietf.org/rfc/rfc4627.txt

306 IETF RFC5246, T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*,
307 http://www.ietf.org/rfc/rfc5246.txt

308 ISO 8601:20044, International Organization for Standardization, Geneva, Switzerland, *Data elements and*
309 *interchange formats -- Information interchange - - Representation of dates and times*, March 2008,
310 http://www.iso.org/iso/iso_catalogue/ catalogue_tc/catalogue_detail.htm?csnumber=40874

311 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,
312 http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype

313 NIST Special Publication 800-145, Peter Mell and Timothy Grance, *The NIST Definition of Cloud*
314 *Computing*, Sept. 2011, http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf

315 NIST Special Publication 500-292, Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee
316 Badger and Dawn Leaf, *NIST Cloud Computing Reference Architecture*, Sept. 2011,

http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/ReferenceArchitectureTaxonomy/NIST_SP_500-292_-_090611.pdf

Representational State Transfer, Roy Fielding, Doctoral dissertation, University of California, *Architectural Styles and the Design of Network-based Software Architectures (Chapter 5)*, 2000, http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm

XMLSchema - Part 1, World Wide Web Consortium (W3C) Recommendation, H. Thompson, et al., Editors, *XML Schema Part 1: Structures Second Edition*, 28 October 2004, http://www.w3.org/TR/xmlschema-1/

XMLSchema - Part 2, World Wide Web Consortium (W3C) Recommendation, P. Biron, A. Malhotra, Editors, *XML Schema Part 2: Datatypes (Second Edition)*, 28 October 2004, http://www.w3.org/TR/xmlschema-2/

3 Terms and definitions

In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause.

The terms "shall" ("required"), "shall not," "should" ("recommended"), "should not" ("not recommended"), "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Annex H. The terms in parenthesis are alternatives for the preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that ISO/IEC Directives, Part 2, Annex H specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning.

The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 5.

The terms "normative" and "informative" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 3. In this document, clauses, subclauses, or annexes labeled "(informative)" do not contain normative content. Notes and examples are always informative elements.

The terms defined in DSP4004, DSP0223, and DSP1001 apply to this document. The following additional terms are used in this document.

3.1

authentication

The process of verifying a claim, made by a subject, that it should be allowed to act on behalf of a given principal (person, service, etc.). Typical authentication mechanisms involve the use of username/password combination or public/private key pairs.

3.2

authorization

(also known as Access Control) The process of verifying that an authenticated principal (person, service, etc.) has permission to perform certain operations (e.g., read, update) on specific resources.

3.3

cloud

Synonymous with “cloud computing” as defined in section 2 of the NIST Definition of Cloud Computing [SP800-145].

3.4

Cloud Service Consumer

A category of actors that includes the Consumer Business Manager (who approves business and financial expenditures for consumed services; accounts for used service instances; establishes business relationships; sets up accounts, budget, and terms; etc.); the Consumer Service Administrator (who requests service instances and changes to service instances; purchases services within the business relationship; creates Service Users (including policies); allocates resources, such as computer and storage; generates reports, such as usage; etc.); and Service Users (who use service instances provided by a Cloud Service Provider). The term "Consumer" is used when the indicated action or activity could involve one or more of the above actors. In cases where the distinction between the actors in this category is relevant, the more detailed term will be used.

For purposes of comparison and alignment, it should be noted that a Cloud Service Consumer is equivalent to the “Cloud Consumer” actor defined in the NIST Reference Architecture [SP500-292].

3.5

Cloud Service Provider

A category of actors that includes the Service Operations Manager (who manages the technical infrastructure required for providing cloud services; monitors and measures performance and utilization against SLAs; provides reports from monitoring and measurement; etc.); Service Business Manager (who offers all types of services developed by cloud service developers; accounts for services potentially offered by service Providers themselves and services offered on behalf of cloud service developers; establishes a portfolio of business relationships; and sets up accounts and terms for Consumers, etc.); and Service Transition Manager (who enables a customer to use the cloud service, including "onboarding", integration, and process adoption; defines and creates service offerings based on Templates and Configurations that can be used by Consumers and are populated into the catalog; etc.). The term "Provider" is used when the indicated action or activity could involve one or more of the above actors. In cases where the distinction between the actors in the category is relevant, the more detailed term will be used.

For purposes of comparison and alignment, it should be noted that a Cloud Service Provider is equivalent to the “Cloud Provider” actor defined in the NIST Reference Architecture [SP500-292].

3.6

configuration

A Configuration is a set of metadata, the values of which serve as the parameters of a discrete conformation of a specific type of virtual resource. For example, a Machine Configuration may define a Machine with the equivalent of a 2.66 GHz processor, 4 GB of memory, and 320 GB of local disk storage.

3.7

Infrastructure as a Service (IaaS)

A cloud computing service model defined in section 2 of the NIST Definition of Cloud Computing [SP800-145].

3.8

message confidentiality

A quality of a message that prevents anyone but the intended receiver(s) from viewing its contents.

3.9

message integrity

A quality of a message that allows a receiver of that message to determine whether the contents of the message have been altered since its creation.

3.10

Template

A Template is the resource that represents the set of metadata and instructions used to instantiate resources (e.g., a Machine Template is used to create Machines). Templates may aggregate other metadata resources such as other Templates, Configurations and Images. For example, a Machine Template refers to a Machine Configuration and a Machine Image.

How a specific protocol mapping, or implementation, chooses to supply Templates as inputs to the instantiation process may vary. However, some common patterns should be considered:

1. By reference - allow Consumers to reference a Template (that exists as a resource in the Provider) as part of the instantiation operation.

2. By value - allow Consumers to dynamically provide the Template information as part of the instantiation operation.

3. Reference with overrides - allow Consumers to reference a Template (that exists as a resource in the Provider) and provide additional values that override the attributes of that Template as part of the instantiation operation.

4 HTTP-Based protocol

4.1 Introduction

All operations are based on the *HyperText Transfer Protocol (HTTP)*, version 1.1 [RFC2616]. Each request is sent using an HTTP verb such as PUT, GET, DELETE, HEAD, or POST and includes a message body in either JSON or XML format. Each response uses a standard HTTP status code, whose semantics are interpreted in the context of the particular request that was made. Each resource in the model has a MIME type that further contextualizes the payload of requests and responses.

Resources in the model are identified by URIs, and each resource's representation shall contain an "id" attribute, of type URI, that acts as a "self pointer." This URI shall be unique within the context of the Provider's implementation. Dereferencing (via an HTTP GET) the URI of an resource will yield a representation of the resource containing attributes and links to associated resources. To begin operations, a client shall know the URI to the main entry point of a Provider - also known as the "Cloud Entry Point" resource. All other resources within the environment shall then be discoverable via the iterative following of links to associated resource within each resource retrieved.

4.1.1 Protocol evolution and client expectations

Future versions of this specification will structure changes in such a way that clients that conform to an earlier version of this specification will continue to work, and will not be adversely affected by the evolution of the protocol. Clients are expected to follow a few simple rules to ensure this.

1. Clients shall not assume that the serializations shown for responses in this specification are complete. In particular, clients shall accept responses that contain data mixed in with the serializations shown here, and shall ignore such data. However, per clause 4.2.1.3, clients shall include unknown data in PUT requests to update resources.

2. Clients shall not assume anything about the operations supported by a server. They are expected to discover which operations are supported (and permissible) by navigating to resources from the cloud entry point. The serializations of resources encountered will indicate which operations are supported by the server.

4.1.2 XML namespaces

The following table lists the XML namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

Prefix	XML Namespaces	Specification
cimi	http://schemas.dmtf.org/cimi/1	This specification
xs	http://www.w3.org/2001/XMLSchema	XML Schema Part2

4.1.3 URI space

While URIs returned by Providers are to be treated as opaque by Consumers, and Consumers shall not make assumptions about the layout of the URIs or the structures of the URIs for the resources, Consumer may augment URIs with any well-defined query parameters that are supported by the Provider as defined in clause 4.1.6.

The sample URIs used in this specification are not normative and the patterns used shall not be interpreted as guidance for implementations. For example, any of the following URIs might be used by Providers to reference a particular Machine resource:

```
http://example.com/machines/12345
http://example.com/machines?id=12345
http://example.com/12345
http://example.com/Cloud/resource?id=12345
```

4.1.4 Media types

In this specification, resource and response representations are encoded either in JSON, as specified in RFC4627 or in XML. When serialized in JSON, the media-type for CIMI resources shall be "application/json." When serialized in XML the media-type shall be "application/xml."

In the JSON serialization of CIMI representations sent by Providers there shall be an additional attribute on the root object called "resourceURI" that will contain the unique URI that is associated with the type of CIMI resource being serialized.

Note that this requirement applies even when $select is used to subset the resource being acted upon.

In the XML serialization of Collection representations sent by Providers shall contain this "resourceURI" attribute, as shown in the example XML serialization of Collections in clause 5.5.12.

This attribute is optional for Consumers to include. When included, this attribute's value shall match the "typeURI" attribute of the corresponding ResourceMetadata resource (see clause 5.11), if ResourceMetadata is supported. This value shall also be equivalent to the wrapping element of the XML serialization; in other words, the namespace of the wrapper element concatenated a "/" and then its localName.

. Any CIMI resource implemented by a Provider shall have representations in JSON and XML. The client implementation may thus use either JSON or XML in requests with any server implementation, and may request a specific serialization using server-driven content negotiation (using the Accept request header).

4.1.5 Request headers

This specification uses general-header, request-header, and entity-header headers as defined in RFC2616 in request messages to provide metadata about the message. Applications using messages defined in this specification shall use headers consistent with the requirements of RFC2616.

4.1.6 Request query parameters

Providers may choose to include query parameters as part of the URIs returned to Consumers. Consumers shall include those query parameters when sending messages to those URIs. If Providers choose to define query parameters care should be taken to avoid conflicts with CIMI defined query parameters.

To modify the behavior of the Provider when processing request messages, Consumers may augment request URIs as described in the following clauses. As stated in clause 4.1.3, URIs returned from Providers are to be treated as opaque by Consumers, however, it is the responsibility of the Consumer to understand the use of the query parameters defined in the following clauses and ensure correctness when making a request.

Unsupported, or unknown, query parameters shall be silently ignored by Providers. Consumer may examine the CloudEntryPoint's capabilities to determine whether support of these query parameters is enabled.

4.1.6.1 Filtering collections

When retrieving the representation of a collection, Consumers may include the $filter query parameter to reduce the number of entries of the collection that are returned based on the data within the entries of the collection. The $filter parameter shall be of the form:

```
?$filter=expression
```

Where "expression" represents a mathematical expression denoting how the top-level attributes of the resources within the collection shall be filtered. The expression is defined by the following EBNF grammar:

```
Filter      ::= AndExpr ( 'or' Filter )* ;
AndExpr     ::= Comp ( 'and' AndExpr )*
Comp        ::= Attribute Op Value
              | Value Op Attribute
              | PropExpr
              | '(' Filter ')'
Op          ::= '<' | '<=' | '=' | '>=' | '>' | '!='
Attribute   ::= ? resource attribute name ?
Value       ::= IntValue | DateValue | StringValue | BoolValue
IntValue    ::= /[0-9]+/
DateValue   ::= ? as defined by XML Schema ?
StringValue ::= "..." | '...'
BoolValue   ::= 'true' | 'false'
PropExpr    ::= 'property[' StringValue ']' Op StringValue
```

Where "PropExpr" is used to find resources that contain a property with a certain key/value combination. Where the "key" is the "StringValue" within the square brackets ([ ]) and the "value" is the "StringValue"

after the "Op". The resource shall be considered to satisfy the search criteria if any of the properties in the resources match the specified "PropExpr".

Each of these shall be percent encoded in the URL as appropriate.

The choice of which operator (including 'and' and 'or') is limited based on the type of the value and attribute. The following describes the allowable operators:

```
'or', 'and'                       : Boolean value/attribute
'<', '<=', '=', '>=', ">', '!='   : Integer and date value/attribute
'=', '!='                         : String value/attribute
```

Consumer may include multiple filters within a single URI. Provider shall treat multiple filters as a series of "and" expressions where an entry of the collection shall only be included in the response message if it satisfies all of the filter expressions specified.

Examples:

In the following examples the following sample base URIs are used:

- /machines is the URI to the Machines Collection
- /machines/123 is the URI to a Machine
- /machines/123/disks is the URI to the DiskCollection of a Machine
- /machines/123/volumes is the URI to the MachineVolumeCollection of a Machine

To filter the "Machines Collection" so that just Machines with a "name" attribute of "mine" are returned, the following filter would be used:

```
GET /machines?$filter=name='mine'
```

To filter a "DiskCollection" of a Machine so that just Disks with a format of "ntfs" are returned, the following filter would be used:

```
GET /machines/123/disks?$filter=format='ntfs'
```

When $filter is used, the collection's "count" attribute shall contain the number of resources matching the filter expression.

4.1.6.2 Subsetting Collections

When retrieving the representation of a collection, Consumers may include query parameters to subset the number of entities of the collection that are returned. While the previous clause discussed how to perform a filter over the data within the collection, this clause uses ordinal position within the collection to achieve the desired reduction.

This specification defined two query parameters that, when used, shall indicate the first and last ordinal positions of the entities within the collection that are returned. The query parameters shall be of the form:

```
?$first=number
?$last=number
```

Where "$first" indicates the (1-based) ordinal position of the first entity of the collection to return. And "$last" indicates the (1-based) ordinal position of the last entity of the collection to return. Consumer are not required to use both at the same time. When $first is specified but $last is not, then the implied value for $last shall be the ordinal position of the last entity in the collection. Conversely, when $last is specified but $first is not, the implied value for $first shall be 1.

If any part of the range as expressed by $first and $last is outside of the bounds of the collection then just the resources (if any) in the collection that are contained within that range shall be returned. A fault shall not be generated if any part, or all, of the expressed range is outside the bounds of the collection. Note that if $first is larger than $last then the range shall represent an empty range and therefore no resources returned.

When either $first or $last are specified, and a filter expression (as defined in clause 4.1.6.1) is also specified, then the filter expression shall be performed first and then the ordinal constraints of $first and $last shall be applied.

4.1.6.3 Subsetting resources

The $select query parameter may be used to specify a subset of a resource to be acted upon. This shall have the semantic equivalence of referencing a different resource whose attributes are a subset of the original resource as specified by the attribute names listed in the $select query parameter. The format of a $select query parameter is:

```
?$select=attributeName,...
```

The value of the $select query parameter shall be a comma separated list of top-level attribute names of the resource, possibly including the string “operations” in case the intent is to select the operations available to the Consumer for this resource. Any attribute name erroneously appearing in the list that is not part of the resource shall be ignored by the Provider. An attribute name of "*" is equivalent to specifying all of the attributes of the resource including its operations. Any attribute name explicitly appearing more than once in a URI shall have its second (and subsequent) appearances ignored.

The $select query parameter may appear more than once in a URI that is semantically equivalent to all of the attribute names appearing as values of a single $select query parameter. For example:

```
?$select=name&$select=state
```

is equivalent to:

```
?$select=name,state
```

The order of attribute names in the $select query parameter is not relevant for serialization purposes. The attributes will be serialized per the serialization rules/order as specified by the resource definition.

Note that per clause 4.1.4, when a resource representation is sent by a Provider it shall always include the "resourceURI" attribute even if it is not specified in the $select query parameter.

For example, to subset the list of Machine attributes being acted upon to just the "name" and "description", the following query parameter would be used:

```
?$select=name,description
```

See clause 4.2.1.3.1 for more information on the impact of using this query parameter when updating a resource.

When $select is used in the URI for a collection resource, the subsettings shall apply to the attributes of the collection resource itself as for any other resource. For example, to subset a collection resource in order to only return the number of its items, plus the operations available on this collection:

```
?$select=count,operations
```

However, exceptionally for collection resources, if some attribute provided in the $select list is not a top-level attribute of the collection resource but instead is an attribute of the entities that are items of the collection then the subsetting shall apply to each item of the collection regarding this attribute. For example, when retrieving the DiskCollection, the following query parameter:

```
?$select=name,capacity
```

would return a collection of the Disks associated with a Machine but each entity of the collection would just have the "name" and "capacity" attributes and nothing else, not even the "operations" or "id" attributes.

Optionally, an implementation may also support the alternative attribute name notation: <collectionName>/<attributeName> for subsetting the items inside a collection. For example, the following subsetting on items of a Disks collection is equivalent to the one done in the previous example, while in addition listing the operations of the collection resource itself (not of its items):

```
?$select=disks/name,disks/capacity,operations
```

This notation, when supported (see the “QueryPathNotation” capability in 5.11.2), allows for disambiguating subsettings when the same attribute name can be found for the collection and for each item in the collection (which is always the case for “id” and “operations”).

4.1.6.4 Expanding references

The $expand query parameter specified which of the top-level "reference" attributes of a resource shall be "expanded" when retrieving this resource. To expand a reference means that the attributes of the resource being referenced shall be included in the serialization of that attribute. This feature allows for a more optimized retrieval of resources.

The serialization shall be performed as follows:

JSON serialization:

```
"name": { "href": string }
```

shall be expanded to be:

```
"name": {
  "href": string,
  ... attributes of referenced resource...
}
```

XML serialization:

```
<name href="xs:anyURI"/>
```

shall be expanded to be:

```
<name href="xs:anyURI">
  ... attributes of the referenced resource...
</name>
```

Note that in the XML case the nested elements shall not contain the wrapper element of the referenced resource (e.g., <Machine> in the case of a reference to a Machine resource).

The format of a $expand query parameter shall be:

```
?$expand=attributeName,...
```

The value of the $expand query parameter is a comma separated list of attribute names. Any attribute name erroneously appearing in the list that is not part of the resource, or is not a reference, shall be ignored by the Provider. An attribute name of "*", or no attribute name list at all, is equivalent to specifying all of the attributes. Any attribute name explicitly appearing more than once in a URI shall have its second (and subsequent) appearances ignored.

The $expand query parameter may appear more than once in a URI, which is semantically equivalent to all of the attribute names appearing as values of a single $expand query parameter.

When the resource being retrieved is a collection, the attribute names listed in the $expand shall apply to the attributes of the entities within the collection. For example, specifying:

```
?$expand=volumes
```

when retrieving the Machine Collection shall have the same net effect as applying the "expand" semantics to the specified attribute ("volumes" in this example) of each Machine within the collection. To be clear, $expand acts on the attributes of the resources in the collection, not on the wrapping collection resource itself.

4.1.6.5 Specifying the resource format

When retrieving the representation of a resource the HTTP Accept header is used to specify the encoding style of the response. While it is recommended that Consumer use the Accept header, there might be situations where Consumer are unable to control the values specified in that header. In these cases Consumers may use the $format query parameter to override the Accept header values.

The $format parameter shall be of the form:

```
?$format=encoding
```

Where "encoding" is the requested representation of the response. This specification defines two possible values: "json" and "xml". Provider may support others. The value of the $format query parameter shall be case insensitive.

When both an Accept header and $format query parameter are present in a request message then the $format value shall take precedence. If the $format query parameter appears more than once then the second, and subsequent, appearances shall be ignored.

4.1.6.6 Sorting collections

When retrieving the representation of a collection, Consumers may include the $orderby query parameter to sort the entries of the collection that are returned based on different attributes or in a different order (descending). The $orderby parameter shall be of the form:

```
?$orderby=attributeName[:asc|:desc], …
```

The $orderby expression may include multiple, comma separated attribute names. Each attribute name may be optionally followed immediately by a colon and “asc” to denote ascending order (default), or “desc” to denote descending order for that attribute. When neither "asc" nor "desc" is specified then the implied order shall be "ascending”.

The attributes included in the $orderby shall be of the following types as defind in clause 5.5: boolean, dateFormat, duration, integer or string.

The sort shall be performed based on the attribute type.

The following applies to the ‘ascending’ sort order:

boolean – ‘false’ shall come before ‘true’

dateTime – earlier datetime shall come before a later datetime

duration – a shorter duration shall come before a longer duration

679 interger – smaller integer shall come before larger integers. Negative integers shall come before positive
680 integers.

681 string – based on Unicode/UTF-8 sort order

682 For the ‘desc’ sort order the reverse of the above shall be performed.

683

684 **Examples:**

685 To sort the result set of the "Machines Collection" on the “created” attribute in descending order, the
686 following expression would be used:

687
```
GET /machines?$orderby=created:desc
```
688

689 To sort the result set of the "Machines Collection" on the “cpu” attribute in descending order then followed
690 by the “memory” attribute in ascending order, the following expression would be used:

691
```
GET /machines?$orderby=cpu:desc,memory:asc
```
692

693

694 Response headers

695 As defined in RFC2616, this specification uses general-header, response-header, and entity-header
696 headers in response messages to provide metadata about the message. Applications that use messages
697 defined in this specification shall use headers consistent with the IANA HTTP Header Registry.

698 #### 4.1.6.7 Job header

699 If the server supports the Job resource, response messages shall include a header defined by this
700 specification to indicate the URI for the job created to process the associated request message.

701
```
CIMI-Job-URI = "CIMI-Job-URI" ":" string
```

702 In cases where an error occurs during the processing of a request, the Provider shall include a
703 representation of a Job resource describing the status of the failed operation. This representation of a Job
704 shall be included even in cases where the Provider does not normally support Job resources to ensure
705 that Consumers are provided with sufficient information, in a consistent manner, as to the reason for the
706 failure regardless of whether the Provider supports Jobs. When Jobs are not supported in general, any of
707 the references in the Job representation (e.g., "id" or the "href" for nestedJobs) shall be empty paths (i.e.,
708 "") and the "nestedJobs" array shall be expanded (see 4.1.6.4) to inline the representation of the pseudo
709 subordinate Jobs.

710 #### 4.1.6.8 ETag support

711 An Etag header may be provided by a Provider with each resource as specified in RFC2616. If a Provider
712 does provide an Etag header, it shall also support If-Match header processing on behalf of the Consumer.

713 ## 4.2 Protocol operations

714 This clause defines the set of common HTTP operations that a Provider might expose. At its core there
715 are four basic CRUD (Create, Read, Update, and Delete) operations. The manner in which these are
716 used is consistent across all resources within the model; therefore, their use is defined once and is to be
717 applied consistently. Some resources support specialized operations that do not fit well into a CRUD style
718 of operation and those will all follow a similar high-level pattern but each operation is allowed to have

slight variations to accommodate its specific needs. The specifics of these special operations are detailed within the clause that defines the resource.

When appropriate some of the resource representations will include an "operations" attribute. Providers shall only include the "operations" attribute when the specified operations are accessible to the current client for that particular resource. This situation means that based on many factors (e.g., authorization rights of the clients, current state of the resource, etc.), a different set of "operations" shall be returned on each serialization of the resource. Each operation shall include a "rel" and an "href" field. The "rel" field shall uniquely identify the operation name (e.g., "add", "edit"), while the "href" field is the URI to which the operation's request message shall be sent. Note that the "href" field's URI may be different from the URI of the resource itself. The "operations" attribute shall be serialized as follows:

JSON serialization:

```
{ "operations": [
    { "rel": "string", "href": "string" }, +
  ]
}
```

XML serialization:

```
<Resource xmlns="http://schemas.dmtf.org/cimi/1">
  <operation rel="xs:anyURI" href="xs:anyURI"/> *
</Resource>
```

For example, the "edit" operation would appear as:

JSON serialization:

```
{ "operations": [
    { "rel": "edit", "href": "<editURI>" }
  ]
}
```

XML serialization:

```
<Resource xmlns="http://schemas.dmtf.org/cimi/1">
  <operation rel="edit" href="<editURI>I"/>
</Resource>
```

Additional "rel" values may be defined by Providers; however, they shall be fully qualified URIs and not relative URIs.

4.2.1 Common CRUD operations

Each of the resources supported by this protocol shall adhere to the interaction patterns defined in the following clauses.

4.2.1.1 Creating a new resource

To create a new instance of a resource type, an HTTP POST request is sent to a designated "addURI" for that resource type. In many cases, the Collection resource that maintains, or groups, all instances of that resource type will include an "add" operation. The "add" operation references the "addURI" that is to be used.

The HTTP POST request shall include:

- CIMI serialization of the request to create a new resource in the HTTP Body
- HTTP Content-Type header
- HTTP Content-Length header

For example, the request can be:

```
POST <addURI> HTTP/1.1
Host: <hostname>
Accept: application/(json|xml)
Content-Type: application/(json|xml)
Content-Length: <length>

<serialization of request to create a new resource>
```

This example has an Accept header with one of the CIMI supported media types: application/json or application/xml. If the Provider chooses to reply with a serialization, then this serialization should be of the specified media type. Omission of the Accept header allows the Provider to reply with a serialization of any media type.If the resource has a “State” attribute, then its value shall be “CREATING” while the Provider is processing this operation.

Many of the create requests are defined such that a Template of the new resource is passed in. These create requests allow for the Template to be passed in "by-reference" or "by-value." For example, creating a new Machine looks like this (here using XML):

```
<MachineCreate xmlns="http://schemas.dmtf.org/cimi/1">
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <property key="xs:string"> xs:string </property> *
  <machineTemplate href="xs:anyURI"? >
    ... template attributes ... ?
  </machineTemplate>
</MachineCreate>
```

Note that in the XML case the creation of a new Machine requires a wrapper element named "MachineCreate" per the rules specified in clause 5.5.12.1.

More generally, creating a new resource shall follow one of these two serialization patterns (here illustrated in JSON):

(1) Resource creation by passing a template by value:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceCreate",
  "name": "myResourceName", ?
  "description": "My resource description", ?
  "properties": { “prop1name” : “prop1value” , + }, ?
  "resourceTemplate": {
    <here the template is passed by value>
  }
}
```

(2) Resource creation by passing a template by reference:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceCreate ",
  "name": "myResourceName", ?
  "description": "My resource description", ?
  "properties": { "prop1name" : "prop1value" , + }, ?
  "resourceTemplate": { "href": string ,
    <here some template attribute/value pairs may be added to override values
in the referenced template>
  }
}
```

In case the created resource is itself a template, only the first creation pattern - by value - applies.

In both patterns (1) and (2) the “resourceURI” specifies the operation here generically identified as “ResourceCreate”,, e.g. MachineCreate.

In both patterns (1) and (2) an element corresponding to the resource template (here identified generically as “resourceTemplate” e.g. machineTemplate) is specifying the template to be used, either by value (1) or by reference (2).

Direct setting of attributes in the new resource:

In a creation request it is possible to set the value of some attributes of the newly created resource, regardless of what values the template instantiation might have set if used alone. Three common attributes of the new created resource may be set: “name”, "description" and “properties”.

The semantics shall be same as of a partial update of the resource for these attributes (described in a next sub-section), immediately following the resource creation from the template alone.

Defining or referring to the resource template:

In pattern (1) above, the Provider may choose to create a template resource from the value given but such creation is temporal in nature. The Provider shall not expose such a transient resource to the Consumer and no such transient resource shall be included in any query results back to the Consumer.

In pattern (2) above, additional attribute name/value pairs may be given inside the resourceTemplate element that also contains the reference to the external (pre-existing) template in order to override similar attributes defined in the template. More precisely:

- Any top-level attribute of complex or simple type in the referred template shall be overridden by providing its name / value pair in the create request inside the resourceTemplate element and immediately under it. For a top-level attribute of complex type (e.g. arrays, collections, structures) the provided complex value shall also set all underlying attributes – e.g. array elements.

- The semantics shall be same as of modifying (overriding) parts of the referred template just before it is used for instantiation, but these overrides shall not persist in the referred template and shall only concern this particular instantiation.

In pattern (2) above Consumer may erase any Template attributes by specifying either

```
"attribute": null
```

for the attribute in the JSON serialization, or

```
<attribute/>
```

in the XML serialization for that attribute.

Examples:

Here is an example of creation pattern (1) using a MachineTemplate by value (in JSON):

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate ",
  "name": "myMachine123",
  "description": "A machine to be connected to a pre-existing network",
  "machineTemplate": {
    <here a template passed "by value" i.e. the attribute/value pairs for the
MachineTemplate template. An example is of the networkInterfaces below: >
    "networkInterfaces": [
      { "addresses": [ { "address": { "href": "http://example.com/addresses/add1"
}},{ "address": { "href": "http://example.com/addresses/add2" }} ],
       "network": { "href": "http://example.com/networks/net1" },
       "state": "ACTIVE" }
      ]

  }
}
```

In the previous example:

The attributes "name" and "description" are instance-level settings because outside the machineTemplate element (i.e. they will set attribute values in the new created resource, not in the template used to create the resource). The name of the new Machine will be "myMachine123".

This Machine will be connected to an existing Network of reference (http://example.com/networks/net1), as specified in the template complex attribute .

Here is an example of creation pattern (2) using a MachineTemplate by reference:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate ",
  "name": "myMachine456",
  "description": "A machine connected to a pre-existing volume",
  "machineTemplate": { "href": "http://example.com/machineTemplates/72000",
    "credential": { "href": "http://example.com/myCredential" }
    "networkInterfaces": [
      { "addresses": [ { "address": { "href": "http://example.com/addresses/add4"
}},{ "address": { "href": "http://example.com/addresses/add5" }} ],
       "network": { "href": "http://example.com/networks/net1" },
       "state": "ACTIVE" }
      ]

  }
}
```

In the above example, a new machine named "myMachine456" is created, also connected to the same existing Network as in example (1) but with a different set of Addresses. Two kinds of attributes are provided with values at creation time in this example:

- Instance-level attribute settings: these shall directly update similar attributes in the created resource, here “name” and “description”.
- Template-level overrides: The referred MachineTemplate is used for creating the Machine, but the “credential” attribute in this template is (temporarily) overridden by the credential provided in the creation request. So is the networkInterfaces array. In case such attributes were not present in the referred template, they will be added (temporarily) just for this Machine creation.

Some of the create requests allow for configuration type of resources to be passed by-reference or by-value as well - e.g., Credential on a Machine create operation. The processing rules defined above applies in those cases as well.

If the response has a 201 status code, then the response shall include:

- HTTP Location header with a reference to the new resource

If the response to a create request includes a serialization of the new resource, then the response shall additionally include:

- HTTP Content-Type header
- HTTP Content-Length header

For example, the response can be:

```
HTTP/1.1 201 Created
Location: <location>
Content-Type: application/(json|xml)
Content-Length: <length>

<serialization of new resource>
```

4.2.1.2 Retrieving a representation of a resource

To retrieve a representation of resource, an HTTP GET request is sent to the resource's URI.

For example, the request can be:

```
GET <ResourceURI> HTTP/1.1
Host: <hostname>
Accept: application/(json|xml)
```

If the response has a 200 status code, then the response shall include:

- HTTP Content-Type header
- HTTP Content-Length header

For example, the response can be:

```
HTTP/1.1 200 OK
Content-Type: application/(json|xml)
Content-Length: <length>
```

```
<serialization of resource>
```

4.2.1.3 Updating a resource

To update a resource's state, an HTTP PUT request containing the complete, updated representation is sent to a designated "editURI" for that resource type. Clients shall include all non-empty attributes of the resource in the PUT request - including ones that it might not support or understand that were returned in a GET response. This is to ensure that a client does not inadvertently modify (erase) data in a resource by excluding it from the full representation of the resource.

In many cases, this "editURI" will be the same as the URI of resource itself. Retrieving the resource representation shall include an "edit" operation, which contains the "editURI" that is to be used, if the requester is allowed to modify the resource.

While processing a PUT request, if the server detects that an attempt is being made to update a read-only, or immutable, attribute, it shall silently ignore that attribute update request and shall not generate an error. This rule applies to resource partial updates as well.

Because of potential conflicts that might occur due to multiple concurrent updates, Consumers should use the partial update mechanism, defined in 4.2.1.3.1, to reduce the chances of mistakenly updating attributes with out-of-date data.

The HTTP PUT request shall include:

- CIMI serialization of the updated resource in the HTTP Body
- HTTP Content-Type header
- HTTP Content-Length header

For example, the request can be:

```
PUT <editURI> HTTP/1.1
Host: <hostname>
Accept: application/(json|xml)
Content-Type: application/(json|xml)
Content-Length: <length>

<serialization of request to update a resource>
```

If the response includes a serialization of the updated resource and has a status code of 200, then this response shall include:

- HTTP Content-Type header
- HTTP Content-Length header

For example, the response can be:

```
HTTP/1.1 200 OK
Content-Type: application/(json|xml)
Content-Length: <length>

<serialization of updated resource>
```

966 #### 4.2.1.3.1 Partial updates to a resource

967 For clarity, this clause explains how to use the $select query parameter (see clause 4.1.6.3) to subset a
968 resource for the purposes of only operating on a selected set of top-level attributes.

969 To update only certain top-level attributes of a resource, a Consumer may include only the altered
970 attributes in the representation of the resource within the HTTP request body. When this request is made,
971 the URI to the resource shall include the attributes to be modified as a comma separated list of query
972 parameters; in other words, the URI shall be of the form:

973 ```
http://example.com/resource?$select=attribute1,attribute2,...
```

974 Only the attributes listed in the URI's query parameters shall be modified; attributes not listed in the URI
975 shall not be directly modified by the request. Note that this circumstance does not preclude the
976 modification of one attribute causing side-effects that result in the modification of an attribute not listed in
977 the query parameters.

978 Any attribute listed in the URI but not included within the HTTP request body shall be reset to a resource
979 specific value (e.g., removed).

980 From an HTTP perspective, the updated subsetted resource is a distinct one. The semantics of a normal
981 HTTP PUT are adhered to; it is a complete replacement update of the specified resource. From the
982 Consumer's perspective, the partial update is interpreted and executed by the Cloud Service Provider,
983 and some part of the resource is changed.

984 Adhering to the generic PUT semantics defined previously, any attribute of the original (full) resource
985 included within the HTTP request body shall result in an error being generated if that attribute is not listed
986 in the $select query parameter - see clause 5.4. Note that this is due to these attributes being unknown to
987 this subsetted resource.

988 The following sample request updates just the name and description attributes of a Machine:

989 ```
PUT /machines/myMachine?$select=name,description HTTP/1.1
990 Host: <hostname>
991 Accept: application/xml
992 Content-Type: application/xml
993 Content-Length: <length>
994
995 <Machine>
996   <name>My New Machine</name>
997 </Machine>
```

998 The "name" attribute is set to "My New Machine" and the "description" attribute is erased.

999 #### 4.2.1.4 Deleting a resource

1000 To delete a resource, an HTTP DELETE request is sent to a designated "deleteURI" for that resource
1001 type. In many cases, this "deleteURI" will be the same as the URI of resource itself. Retrieving the
1002 resource representation shall include a "delete" operation, which contains the "deleteURI" that is to be
1003 used, if the requester is allowed to delete the resource.

1004 For example, the request can be:

1005 ```
DELETE <deleteURI> HTTP/1.1
1006 Host: <hostname>
```
```

If the resource has a “State” attribute, then its value shall be “DELETING”, while the Provider is processing this operation.

For example, the response can be::

```
HTTP/1.1 200 OK
```

4.2.1.5 Other operations

While some modifications to the resources in the model can be done via a simple update (PUT) operation to the resource's "editURI", sometimes a more complex set of actions need to be taken. In these cases, the operations shall be modeled as HTTP POSTs to the operation specific URI of the resource.

For each of the resources that define additional operations, a description of the HTTP request and response bodies will be provided. However, the general HTTP interaction will be as described below.

The request shall be of the following form:

```
POST <operationLinkURI> HTTP/1.1
Host: <hostname>
Accept: application/(json|xml)
Content-Type: application/(json|xml)
Content-Length: <length>

<serialization of request to perform some action>
```

The form of the response will vary depending on the operation and will be defined by the operation itself.

Note that the definition of the "Create" operation (see clause 4.2.1.1) follows this same pattern. It is just called out for ease of reference.

4.2.1.6 Synchronous operations

If a Provider supports the Job resource, each incoming PUT, DELETE, POST request shall result in a Job resource being created and an absolute URI reference to that Job resource shall be returned back to the client via the CIMI-Job-URI HTTP Header in the HTTP response message:

```
CIMI-Job-URI: <uri-to-Job>
```

In this case, the requested operation shall be complete and the Job URI shall point to a completed Job. If the Job is not complete, the server shall return a 202 and follow the instructions for Asynchronous operations.

4.2.1.7 Asynchronous operations

In some cases, an operation requested by the client may take an undetermined amount of time to complete. For example, creating a new Machine or starting an existing Machine, may take a relatively long time to complete. In these cases, it is not practical to complete these operations within a reasonable HTTP request timeout interval, so the Provider shall return an HTTP "202 Accepted" response code.

As with synchronous operations, if a Provider supports the Job resource, it shall create a Job resource for the incoming request and return a reference to that Job resource back to the client via the CIMI-Job-URI HTTP Header in the HTTP response message. Additionally, in the case of a "202 Accepted" response code, the Provider may also return any of the following in the HTTP response body:

- a representation of the Job resource, if one was created.

- a partial representation of the response message as if the operation were a synchronous operation. For example, when creating a new Machine the response message may include a partial representation of the new Machine in the response message. The list of attributes of the resource that are returned will be implementation specific and based upon how much information is available at the time the response message is generate, but it shall be consistent with the definition of the full resource representation. In the case of a create operation, the Provider may also include an HTTP Location header referencing the "to be created" resource if it is known.

- an empty response body.

Note that the decision as to whether any particular operation will be synchronous or asynchronous is at the server's discretion.

4.3 OVF support

The *Open Virtualization Format (OVF) Specification* describes an open, secure, portable, efficient, and extensible format for the packaging and distribution of software to be run in virtual machines. OVF support in CIMI allows an OVF package to be used to create CIMI management resources by importing the package. Additionally, CIMI management resources can be exported into an OVF package. The actual support for the OVF package will typically be provided by a hypervisor being managed by the CIMI provider. The import of an OVF package exposes CIMI specific constructs and parameters as a result of the import without altering the original OVF package. Thus the CIMI resources that are created as a result of the import form a “View” of what the hypervisor did; however, other (non-CIMI mapped) information from the OVF package may have been used by the hypervisor in its import. This other information is implementation dependent and is not further touched upon by this standard.

An OVF package can support single virtual machines (VMs) corresponding to a single CIMI Machine or Machine Template (see clause 5.14.1) or may also support a complex hierarchy of VMs and their related resources corresponding to a CIMI System or System Template (see clause 5.13.1) and related CIMI management resources.

OVF Support is covered in more detail in ANNEX A.

5 Model

This model assumes that a business relationship has already been established between the Consumer and the Provider. This relationship may include financial terms, creating separately administered clouds that the consuming organization is paying for, and the establishment of authentication credentials to access the administrative entry point for each cloud. The scope of this model is one separately administered cloud.

The CIMI model is described here by using a tabular representation. It is inspired from Entity-Relationship modeling, where each entity is modeling a significant cloud resource for which independent access and manipulation is expected. Relationships between resources use a referential mechanism based on unique identifiers that is expected to be already supported by the implementation environment and protocol (e.g., URIs for HTTP).

The model is self-describing and allows for querying its own metadata, e.g., to discover which extensions have been implemented. The model is also extensible in different ways (see clause 5.1).

Along with this model, a serialization of its entities is defined (both in XML and JSON).

An alternative UML diagram representation is provided for each major group of resources

5.1 Resource wrappers

The serialization of resource instances in the model will follow these conventions. Consider the serialization of a resource named “MyResource”:

JSON serialization:

The resource is serialized as an object wrapping all its attributes, but without a wrapper name. The resource includes an "resourceURI" with a URI for the type of resource being serialized. For example:

```
{ "resourceURI": "http://example.com/MyResource",
  "attribute": "value"
}
```

XML serialization:

The resource is serialized as an element with name equal to the Resource name; for example:

```
<MyResource xmlns="http://example.com">
  <attribute> value </attribute>
</MyResource>
```

5.2 Extensibility

There are two types of extensibility mechanisms defined by the CIMI model; one is intended for use by Consumers whilst the other is to be used by Providers.

The first allows for a CIMI Consumer to add additional data to a resource. Each resource in the CIMI model has an attribute called "properties." Consumers, when creating or updating a resource, may store any name/value pair in the "properties" attribute. CIMI Providers shall store and return these values to the Consumer. There is no obligation for the Provider to understand or take any action based on these values; they are there for the Consumer's convenience. Providers shall not add elements to this "properties" attribute.

The second type of extensibility mechanism allows for Provider defined extensions and this specification includes the ResourceMetadata resource for this purpose. ResourceMetadata may be used to:

- Express constraints on the existing CIMI defined resource attributes (e.g., express a maximum for the 'cpu' attribute of the MachineConfiguration resource)
- Introduce new attributes for CIMI defined resources together with any constraints governing these (e.g., a new 'location' attribute for the Volume resource that takes values from a defined set of strings)
- Introduce new operations for any of the CIMI defined resources (e.g., define a new 'compress' operation for the Volume resource)
- Express any Provider specific capabilities or features (e.g., the length of time that a Job resource will be retained after Job completion and before this is deleted).

It is recommended that Providers use the ResourceMetadata resource to advertise these attributes, operations, and capabilities along with any constraints that might need to be understood by Consumers. The ResourceMetadata resource is defined in clause 5.11.

When a Provider receives a message containing an unknown or unsupported attribute, it shall reject the request. When a Consumer receives a message containing an unknown or unsupported attribute, it shall silently ignore the attribute. However, Consumers are required to include those attributes in messages

sent back to the Provider. Note in these cases the Consumer is not required to understand or process the unsupported attribute, merely echo it back to the Provider.

5.3 Identifiers

All identifiers (e.g., resource names, attributes, operations, parameter names) defined by this specification, or defined via an extension, shall adhere to the following:

- Identifier names shall be treated as case sensitive.
- Identifier names shall only use the following set of characters:
  - Uppercase ASCII (U+0041 through U+005A)
  - Lowercase ASCII (U+061 through U+007A)
  - Digits (U+0030 through U+0039)
  - Underscore (U+005F)
- Identifier names shall not start with a Digit (U+0030 through U+0039).

Note that these rules do not apply to the "name" common attribute defined in clause 5.10.1.

5.4 Attribute constraints

Each attribute of the resources in the CIMI model is augmented by a set of "Constraints" that further qualify the attribute being defined. For each attribute there is a Provider and a Consumer set of constraints because each might differ. The following describes the possible "Constraints."

support optional:

This constraint indicates that support for this attribute is optional. If supported, Providers should advertise its support via ResourceMetadata. See clause 5.2 for information concerning the processing of unsupported and unknown attributes. See clause 5.5.14 regarding empty attribute values.

Non-empty Consumer supported writeable (i.e., read-write and write-only) attributes shall always be included as part of the resource representation sent from Consumers to Providers, including create requests.

Non-empty Provider supported attributes shall always be included as part of the resource representation sent from Providers to Consumers.

support mandatory:

This constraint indicates that support for this attribute is required by compliant implementations. When present on a nested attribute, this attribute is required to be supported only if the parent attribute is supported. See clause 5.5.14 regarding empty attribute values.

Non-empty mandatory writeable (i.e., read-write and write-only) attributes shall always be included as part of the resource representation sent from Consumers to Providers - including create requests.

Non-empty Provider mandatory attributes shall always be included as part of the resource representation sent from Providers to Consumers.

immutable:

This Provider constraint indicates that the attribute, once set, shall never change for the lifetime of the resource.

mutable:

This Provider constraint indicates that the attribute may be modified. Providers shall always have the ability to modify these attributes. Whether Consumers have the ability to modify these attributes shall be indicated by the read-only, read-write, and write-only constraints.

read-only:

This Consumer constraint indicates that the attribute may be retrieved but not updated by Consumers. Read-only attributes are not required to appear in the serialization of resources in create or update request messages. If present, they shall be silently ignored by the Provider. Read-only attributes shall appear in the serialization of resources sent from Providers.

read-write:

This Consumer constraint indicates that the attribute may be retrieved and/or updated by Consumers. Read-write attributes shall appear in the serialization of resources sent to and from Providers. Providers may further constrain whether Consumers can update these attributes and should indicate this via ResourceMetadata.

write-only:

This Consumer constraint indicates that the attribute may be updated by Consumers but are not retrievable by Consumers, typically for security reasons. Write-only attributes shall appear in the serialization of resources sent to Providers but shall never appear in the serialization of resources sent from Providers.

5.5 Data types and their serialization

Unless specifically asked to not include certain attributes in the resource representation, the absence of an optional attribute in the representation means that the attribute has no value (i.e., is undefined); meaning there is no notion of an optional attribute having an implied value. Note that a client cannot distinguish (from just looking at the returned representation) whether a particular attribute is not supported from one that does not exist. Likewise, an absent attribute from a resource representation as the input to an update operation means that the Consumer is requesting that the Provider remove that attribute.

The following describes the data types and values that are used within the model definition tables.

5.5.1 boolean

A value as defined by xs:boolean per XML Schema – Part 2, with the exception that the only allowable values are either "true" or "false." The value is case sensitive.

When serialized in JSON these values shall be of JSON type: *boolean*

When serialized in XML these values shall be of XML Schema type: *xs:boolean*

5.5.2 dateTime

A value as defined by xs:dateTime per XML Schema – Part 2 which is consistent with DMTF DSP4004 and ISO 8601. The timestamp should preserve time zone information, i.e., include a local time component and an offset from UTC.

Any constraints on the specific ranges allowed for any particular attribute will be specified by that attribute's definition or at runtime by the Provider via the metadata discovery mechanisms defined by this specification.

For example, Monday, May 25, 2012, at 1:30:15 PM EST is represented as:

```
2012-05-25T13:30:15-05:00
```

When serialized in JSON these values shall be of JSON type: *string*

When serialized in XML these values shall be of XML Schema type: *xs:dateTime*

5.5.3 duration

A value as defined by xs:duration per XML Schema – Part 2. Any constraints on the specific ranges allowed for any particular attribute shall be specified by that attribute's definition or at runtime by the Provider via the metadata discovery mechanisms defined by this specification.

When serialized in JSON these values shall be of JSON type: *string*

When serialized in XML these values shall be of XML Schema type: *xs:duration*

5.5.4 integer

A value as defined by xs:integer per XML Schema – Part 2. Any constraints on the specific ranges allowed for any particular attribute shall be specified by that attribute's definition or at runtime by the Provider via the metadata discovery mechanisms defined by this specification.

When serialized in JSON these values shall be of JSON type: *number*

When serialized in XML these values shall be of XML Schema type: *xs:integer*

5.5.5 string

A value as defined by xs:string per XML Schema – Part 2. Any constraints on this type for any particular attribute shall be specified by that attribute's definition or at runtime by the Provider via the metadata discovery mechanisms defined by this specification.

When serialized in JSON these values shall be of JSON type: *string*

When serialized in XML these values shall be of XML Schema type: *xs:string*

When serializing an attribute of type string, the serialization shall omit this attribute in case of an empty string.

5.5.6 ref

A reference to another resource.

References allow for Consumers to navigate to resources. By starting at the Cloud Entry Point and following the references that appear in the retrieved resources, Consumers will be able to recursively discover and navigate to all other resources.

As a general rule, when an attribute is of type "ref", its value shall be held by an attribute named "href" (both in JSON and XML).

JSON serialization:

In the JSON serialization the "href" property appears as of type "string." When an attribute is of type "ref", the name of this attribute shall appear as a key, with the "href" property as it a nested value. For example, a resource attribute "myvolume" of type "ref" is serialized as:

```
"myvolume": { "href": string }
```

XML serialization:

In the XML serialization the "href" attribute appears as type "xs:anyURI." When an attribute is of type "ref," the name of this attribute shall appear as name of an XML element with the "href" property as an (XML) attribute. For example, a resource attribute "myvolume" of type "ref" is serialized as:

```
<myvolume href="xs:anyURI"/>
```

References in both JSON and XML have an extensibility point that allows for additional information (such as the target resource to be included "by value") if supported. For convenience the JSON and XML representations, as shown above, exclude the implicit extensibility points that would allow for the attributes of the target resource to be included if desired. So, more accurately the above representations might be written as follows:

For JSON:

```
"myvolume": { "href": string, ... }
```

and in XML:

```
<myvolume href="xs:anyURI"> xs:any* </myvolume>
```

However, for brevity the extensibility points are excluded in the serialization of the resources.

5.5.7 map

A list of key/value pairs. The same "key" shall not be used more than once within an attribute. The "key" is case sensitive.

When serializing an attribute of type map, the serialization shall omit this attribute in case of an empty map.

5.5.8 structure

Attributes of this type are complex attributes made up of a set of nested attributes. For each attribute of this type there will be an additional table defining those nested attributes.

A nested structure can be considered a complex type definition. Structures may be named or unnamed. Here is an example of named structure:

Name	*summary*	
Attribute	**Type**	**Description**
low	*number*	Number of “low” occurrences
medium	*number*	Number of “medium” occurrences
high	*number*	Number of “high” occurrences
critical	*number*	Number of “critical” occurrences

JSON serialization:

In JSON, the name of the structure (i.e., of the type it represents) never appears. In other words, whether the structure is named or not does not matter. An attribute named “systemIncidents” of type "summary" (as above) is serialized as follows:

```
"systemIncidents": {
  "low": number,
```

```
  "medium": number,
  "high": number,
  "critical": number
}
```

XML serialization:

In XML, the name of the structure (i.e., of the type it represents) never appears. In other words, whether the structure is named or not does not matter. The same previous "systemIncidents" example will be serialized so that the structure sub-attributes become XML attributes of a <systemIncidents> XML element wrapper:

```
<systemIncidents low="xs:integer" medium="xs:integer" high="xs:integer"
         critical="xs:integer"/>
```

NOTE: A large number of sub-attributes of atomic type in a structure may be represented alternatively as XML child elements for better readability. Both options are available; however, the same structure shall be serialized the same way across resources.

5.5.9 byte[]

An arbitrary set of bytes meant to represent a block of binary data. Any constraints on this type for any particular attribute shall be specified by that attribute's definition or at runtime by the Provider via the metadata discovery mechanisms defined by this specification.

When serialized in JSON these values shall be of JSON type: *string*

When serialized in XML these values shall be of XML Schema type: *xs:hexBinary*

5.5.10 URI

The format and syntax of the attributes of type "URI" is defined by RFC3986.

Unless otherwise noted, this specification does not mandate whether Providers use relative or absolute URI in the HTTP response bodies.

When URIs are specified as relative URIs, they shall be relative to the "baseURI".

The algorithm used for converting a relative URI to an absolute URI shall be as described in section 5.2 of RFC3986. The table below illustrated how relative URIs are resolved against base URIs:

Base URI	Relative URI	Absolute URI
http://example.com/	p1/file	http://example.com/p1/file
http://example.com/c1/	p1/file	http://example.com/c1/p1/file
http://example.com/c1/c2/	p1/file	http://example.com/c1/c2/p1/file

If relative URIs are used, the "baseURI" shall end with a trailing slash and relative URIs shall not begin with a leading slash. This format will be consistent with most URI resolve utilities and will produce the same results as a simple string concatenation algorithm.

When serialized in JSON these values shall be of JSON type: *string*

When serialized in XML these values shall be of XML Schema type: *xs:anyURI*

5.5.11 Arrays

An array represents an ordered list of items of the same type. An array shall always appear as an attribute of a resource, and is only accessible as such (it is not a separately addressable resource). When

a resource is deleted, the items in its arrays shall also be deleted. However, in case these items were just references to other resources, these referred resources are not affected (see the semantics of references in 5.7)

Attributes that are arrays are defined by using the notation "itemType[]," where itemType is the type name for each item of the array. When the type is a structure, not a simple data type, it is recommended as a convention in the model that the name of an array be the plural of a name that characterizes each item. For example, an array of volume items or of references to these may be named "volumes."

When an attribute is of type of references ("ref[]") – and more generally array of an atomic type - the definition in the model shall include an "Array item name", that may be used in its serialization.

JSON serialization:

Within this specification, arrays in JSON are serialized with a wrapper property. The wrapper name shall be same as the attribute name for the array. For example, a "things" attribute of type "thing[]" is serialized as:

```
"things" : [
  { ... }, +
] ?
```

When the items in the array are structures then the structure name shall not be present in the JSON serialization.

In the case of an array of references, i.e., where the "ref" type applies to each element of the array, each element shall simply be serialized as an "href" property within a JSON array. For example, an array "things" of type "ref[]" is serialized as:

```
"things": [
  { "href": string },  +
] ?
```

When serializing arrays, conformant implementations shall not include empty arrays (i.e., arrays that contain no child properties) in the JSON serialization. Notice that the child of the "things" property is defined with a "+", meaning at least one child is required. This requirement ensures that the JSON serialization is minimized and only includes the wrapping "things" element if, and only if, there is at least one "thing" in the array.

XML serialization:

The XML serialization of arrays requires each item of the array to be represented as an element. These elements shall be consecutive and contiguous in the serialization and the name of each element (tag name) shall be the name of the element type (the name that appears before "[]" in the array type). For example, a "things" attribute shall be serialized as a list of items named "thing", where "thing" is the name of a structure:

```
<thing>
  ...
</thing> *
```

There is no wrapper element for an array in XML.

In the case of an array of references, i.e., where the "ref" type applies to each element of the array, the array is serialized as a list of XML elements without wrapper. Each element is named per the "Array item name" value specified in the attribute's definition. For example, an array "things" of type "ref[]" where the "Array item name" is "thing", is serialized as:

```
<thing href="xs:anyURI"/> +
```

5.5.12 Collections

Like arrays, collections are groupings of resources of the same type. In contrast with arrays, collections are themselves resources that have their own URI and can be independently accessed. Collections also allow for an optimized and convenient interaction pattern by providing a specialized set of operations that avoid replacing a large number of items when updating the set.

This specification uses collections when the set of items in the list will most likely be modified often and potentially by multiple Consumers. Conversely, arrays are used when it is expected that the list of items will not be modified often or can be easily modified by substitution of the entire list, and thus the overhead of managing these items as separate resources might be burdensome.

Attributes that are collections are represented as type "collection[itemType]." The resource type of the collection items are specified inside the brackets; for example an attribute that is a collection of Machines is expressed as "collection[Machine]." These will be serialized as a reference to a collection resource. For brevity, while these attributes are "references" the word "ref" or "reference" does not appear in the model definition tables - simply the type "collection[*itemType*]" appears.

To each one of these resource items, shall correspond an entry in the collection. These resources items are assumed to be of a complex type and are separately addressable and manageable. While different collections will contain entries of different resource types, all collections follow the pattern described below:

- Collections shall contain an "id" attribute that acts as a "self pointer." Retrieving the data at this reference shall return the collection. In the XML representation, each collection shall be wrapped by a `<Collection>` element.
- Collections shall contain a "count" attribute which indicates the number of resources in the collection at the time the collection was queried.
- Collections shall contain a list of resources that make up the collection. As with all arrays, if there are no resources in the collection, the serialization of the list shall be omitted.
- As with all resources in the CIMI model, each resource in the collection shall have an "id" attribute that acts as a "self pointer." Retrieving the data at this reference shall return just that one resource and not any parent resource, such as the collection or array attribute.
- Adding new resources to the collection shall be done via the "add" operation defined within the collection. Note that lack of an "add" operation on the collection indicates that new resources are not permitted at that time.
- Deleting resources from the collection shall be done via a "delete" operation on the resource itself.
- Unless otherwise specified, deleting a collection shall also delete all of the resources that make up the collection, but shall not delete any tertiary resources referenced by the to-be deleted collection resources.
- Collections shall be deleted when their owning resource is deleted.

The resources in a collection are of two kinds:

- either the resource is an infrastructure resource (such as those listed in the Cloud Entry Point, or those embedded in an entity such as the disks inside a Machine),

- or the resource is just an intermediary resource that holds a reference to an infrastructure resource, called the “target resource”. By convention, intermediary resources have a name that concatenates the name of the resource owning the collection, with the name of the target resource, e.g. “MachineVolume” is the name of the intermediary resource that is used to connect a Machine to a Volume.

Collections of intermediary resources allow for decoupling the lifecycle of a collection (and of its owning entity) from the lifecycle of the actual target resources. For example, deleting a collection shall delete its intermediary resources but not its target resources. In case the reference to the target resource is a mandatory attribute of the intermediary resource, then the intermediary resource cannot have a longer lifecycle than the target resource.

- If a target resource is deleted, then the Provider shall also delete any intermediary resource that has a reference to this resource as the value of a mandatory attribute.

The serialization of collections shall adhere to the following pattern:

JSON serialization:

```
{ "resourceURI": string,
  "id": string,
  "count": number,
  "resourceSpecificGroupingName": [
    { "resourceURI": string,
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      ... entry specific data ...
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    } +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="xs:anyURI" xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <ResourceSpecificElementName>
```

```
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    ... entry specific data ...
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </ResourceSpecificElementName> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

Where the "resourceURI" attributes shall contain the collection or resource specific URIs for that type of collection, and "resourceSpecificGroupingName" and "ResourceSpecificElementName" shall be replaced with the name of the collection-specific resource name, e.g. "machines" in JSON or "Machine" in XML.

5.5.12.1 Adding items to collections

Adding new resources to collections shall be done by invoking the "add" operation of the collection. The contents of the request body shall be either a representation of the new resource being added to the collection, or a representation of the Template associated with the new resource being created. Each resource that requires the use of a Template indicates this in its definition.

For example, to add a new Volume to a Machine's "volumes" collection, the "add" operation's request body shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolume",
  "initialLocation": string,
  "volume": { "href": string }
}
```

XML serialization:

```
<MachineVolume xmlns="http://schemas.dmtf.org/cimi/1">
  <initialLocation> xs:string </initialLocation>
  <volume href="xs:string"/>
</MachineVolume>
```

Note that while deleting this type of resource from the collection will delete and remove the resource from the collection, it shall not delete the referenced target resource itself - in this case the Volume.

When creating a new resource that requires the use of a Template, the "add" operation shall contain:

- The "common attributes" as defined by clause 5.10.1.

- The resource specific data needed to create it. This data shall either be a reference to the resource-specific Template resource or the resource-specific Template resource itself inlined.
- In the XML case, a wrapper element (named <*ResourceName*Create>).

For example, to create a new Machine (which requires the use of a Template) and add it to the MachineCollection, the "add" operation of the MachineCollection shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate", ?
  "name": string, ?
  "description": string, ?
  "properties": { string: string, + }, ?
  "machineTemplate": { "href": string ?}
  ...
}
```

XML serialization:

```
<MachineCreate xmlns="http://schemas.dmtf.org/cimi/1">
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <property key="xs:string"> xs:string </property> *
  <machineTemplate href="xs:anyURI"? />
  <xs:any>*
</MachineCreate>
```

The MachineCollection will have a new Machine:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
  "id": string,
  "name": string,
  ...
}
```

XML serialization:

```
<Machine xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name
  ...
</Machine>
```

The processing of the "add" operation shall adhere to the semantics defined in clause 4.2.1.1.

Regardless of whether a Template is used, the "add" operation shall create the new resource and add it to the collection and a reference (URI) to the new entry shall be returned in the response message in the HTTP Location header.

5.5.13 "Any" type

Some attributes are polymorphic and can hold various data types, the list of which is indicated in their description. In such cases, the type of the attribute shall be indicated as "any" in the model representation.

5.5.14 Empty attribute values

Attributes of the following types are omitted in cases where they have an empty value: string, map, array, and Collection. Apart from being “Provider optional” or “Consumer optional” , an empty value is the third reason that the serialization schema contains an ‘?’ or an ‘*’ for an attribute.

Other attribute types do not have empty values and shall not be omitted from the serialization for this reason.

5.6 Units

Some of the resources defined by this specification have attributes that describe an amount of something that belongs to, or is associated with, that resource. For example, the `Machine` resource has a `memory` attribute that describes "the size of the memory allocated to this machine." The allowable units of these attributes are listed in the following table. Their meaning is defined in IEC 80000-13:2008. Their numerical equivalents are provided here for convenience:

String	Numerical Value	String	Numerical Value
kilobyte	`10^3`	kibibyte	`2^10`
megabyte	`10^6`	mebibyte	`2^20`
gigabyte	`10^9`	gibibyte	`2^30`
terabyte	`10^12`	tebibyte	`2^40`
petabyte	`10^15`	pebibyte	`2^50`
exabyte	`10^18`	exbibyte	`2^60`
zettabyte	`10^21`	zebibyte	`2^70`
yottabye	`10^24`	yobibyte	`2^80`

5.7 Relationship semantics

A reference between two resource instances has the semantics of a simple "association." In particular, unless specified otherwise, (a) the same referred instance can be referred by other resource instances, i.e., be "shared," and (b) the referred resource instance is not affected when deleting the referring resource instance (i.e., the Delete operation is a "shallow delete" by default).

The embedding of a sub-resource inside another resource, has the semantics of a "composition" (or whole-part relationship in UML). In particular, unless specified otherwise, (a) an embedded sub-resource cannot be shared by several resource instances, and (b) when deleting an embedding resource instance, the embedded sub-resource instances are also deleted.

5.8 Operations

All resource operations defined by this specification are optional for Providers to support. Consumers, via examination of an resource's ResourceMetadata, will be able to determine which operations are supported. However, even for those operations that are supported Consumers will still need to examine each resource's representation to determine which operations are supported at that moment. Whether an

operation is supported will be based on a number of factors, including state of the resource and access control rights of the Consumer. Also see clause 4.2.Operations and states are coupled; i.e., when implementing a state-changing resource operation defined in this specification the corresponding state(s) shall also be implemented. See the resource specific “Operations” sections for additional detail.

The "State" attribute of resources that have this attribute shall only change value if

an operation is performed on this resource and this operation requires a state change, or

an error occurred, in this case the “State” attribute shall obtain the value “ERROR”.

For example, for a ‘start’ operation on a Machine both the STARTING and the STARTED states are required to be supported by the Machine, while the Machine can only leave the STARTED state after another state changing operation is requested, unless an error occurs.

Providers can define additional operations and states. Such extensions shall fall into one of these categories:

(a) A new operation that starts from a CIMI-defined state, or leads to a CIMI-defined state, or both. In the latter case, if a CIMI-defined operation already exists for this transition between two CIMI-defined states, then it shall also be supported by the Provider in addition to the new operation.

A new resource state. In that case, a new operation that leads to that state shall also be created. In other words, a Provider-defined operation has to be performed before a Provider-defined state can be reached.”

(b) A new operation that transitions between two Provider-defined states.

5.9 Alternative model formats

Because it is expected that this specification will be implemented by using a variety of technologies, as a convenience, the definition of the model elements are provided in alternative formats that are easily consumable by technology-specific tooling.

This model is also available in a CIM/MOF format [CIMI-CIM].

In the event of inconsistencies between the various formats, the normative text within this specification takes precedence over the XML Schemas and alternative formats, which in turn take precedence over examples.

5.10 Resources

The following clauses detail the attributes of the resources defined by the CIMI model.

5.10.1 Common attributes

Except for ResourceMetadata and Collection resources (see 5.5.12), the resources described by this document share the following common attributes. There are different requirements for primary and secondary CIMI resources. All resources that are element types of collections in the CloudEntryPoint shall be primary CIMI resources. All other resources shall be secondary CIMI resources. An exception to this rule is that the CloudEntryPoint shall be considered a primary resource.

For example, Machine is a primary CIMI resource as the CloudEntryPoint has a collection with Machine as its element type. However, for example, MachineVolume is a secondary CIMI resource, because the CloudEntryPoint does not have a collection with MachineVolume as its element type.

<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>id</td><td>URI</td><td>The unique URI identifying this resource; assigned upon resource creation. This attribute value shall be unique in the Provider's cloud.

<u>Constraints for primary and secondary resources:</u>
Provider: support mandatory; immutable
Consumer: support mandatory; read-only</td></tr>
<tr><td>name</td><td>string</td><td>The human readable name of this resource; assigned by the creator as a part of the resource creation input.

<u>Constraints for primary resources:</u>
Provider: support mandatory; mutable
Consumer: support optional; read-write

<u>Constraints for secondary resources:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>description</td><td>string</td><td>The human readable description of this resource; assigned by the creator as a part of the resource creation input.

<u>Constraints for primary resources:</u>
Provider: support mandatory; mutable
Consumer: support optional; read-write

<u>Constraints for secondary resources:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>created</td><td>dateTime</td><td>The timestamp when this resource was created. The format should be unambiguous, and the value is immutable.

<u>Constraints for primary and secondary resources:</u>
Provider: support optional; immutable
Consumer: support optional; read-only</td></tr>
<tr><td>updated</td><td>dateTime</td><td>The time at which the last explicit attribute update was made on the resource. Note, while operations such as "stop" do implicitly modify the 'state' attribute it does not change the 'updated_time'.

<u>Constraints for primary and secondary resources:</u>
Provider: support optional; mutable
Consumer: support optional; read-only</td></tr>
<tr><td>properties</td><td>map</td><td>A map of key/value pairs (each entry called a "property"), some of which may control one or more aspects this resource. Properties may also serve as an extension point, allowing Consumers to record additional information about the resource.

The same "key" shall not be used more than once within a "properties" attribute.

Each property shall contain the following nested data:
<table>
<tr><th>Name</th><td colspan="2">property</td></tr>
<tr><th>Data</th><th>Type</th><th>Description</th></tr>
<tr><td>key</td><td>string</td><td>The name of the property.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>value</td><td>string</td><td>The value of the property.

<u>Constraints:</u>
Provider: support mandatory; mutable</td></tr>
</table>
</td></tr>
</table>

Attribute	Type	Description
		Consumer: support mandatory; read-write **Constraints for primary resources:** **Provider:** support mandatory; mutable **Consumer:** support optional; read-write **Constraints for secondary resources:** **Provider:** support optional; mutable **Consumer:** support optional; read-write

The following describes the serialization of these attributes in both JSON and XML:

JSON serialization:

```
"id": string,
"name": string, ?
"description": string, ?
"created": string, ?
"updated": string, ?
"properties": { string: string, + }, ?
```

XML serialization:

```
<id> xs:anyURI </id>
<name> xs:string </name> ?
<description> xs:string </description> ?
<created> xs:dateTime </created> ?
<updated> xs:dateTime </updated> ?
<property key="xs:string"> xs:string </property> *
```

5.11 Resource Metadata

Implementations of this specification should allow for Consumers to discover the metadata associated with each supported resource type. Doing so allows for the discovery of Provider defined constraints on the CIMI defined attributes as well as discovery of any new extension attributes or operations that the Provider may have defined. A ResourceMetadata instance contains metadata describing a particular Resource type – e.g. Network, or Machine – including any Provider specific capabilities or features. The mechanism by which this metadata is made available will be protocol specific.

Note that while this specification does not restrict the editability of the ResourceMetadata attributes, it is expected that these types of features will be reserved for administrative type of Consumers, which means that these attributes will be read-only for most Consumers.

Each resource's metadata shall contain the following pieces of information:

Name	ResourceMetadata	
Type URI	http://schemas.dmtf.org/cimi/1/ResourceMetadata	
Attribute	**Type**	**Description**
id	*URI*	The unique URI identifying this resource; assigned upon resource creation. This attribute value is **immutable**, and shall be **unique** in the Provider's cloud. **Constraints:** **Provider:** support mandatory; immutable

<table>
<tr><td></td><td></td><td>Consumer: support mandatory; read-only</td></tr>
<tr><td>typeURI</td><td>URI</td><td>A unique URI associated with, and denoting, the described resource type.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>name</td><td>string</td><td>The name of the described resource type.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>attributes</td><td>attribute[]</td><td>A set of Provider-defined metadata that can be used by clients to discover any metadata associated with each attribute of the described resource type, including the set of extension attributes not defined in this specification.

Each attribute shall contain the following nested data:
<table>
<tr><th>Name</th><td colspan="2">attribute</td></tr>
<tr><th>Data</th><th>Type</th><th>Description</th></tr>
<tr><td>name</td><td>string</td><td>The name of the attribute.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>namespace</td><td>URI</td><td>The namespace in which this attribute is defined. It is recommended that a dereference of this URI returns information about the attribute. This shall not be present when describing a CIMI defined attribute, but shall be present when describing a non-CIMI defined attribute.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>type</td><td>string</td><td>The data type of the attribute. This shall not be present when describing a CIMI defined attribute, but shall be present when describing a non-CIMI defined attribute.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>required</td><td>boolean</td><td>Indicates whether this resource requires this attribute to be present. When absent the implied value is "false."

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>value constraints</td><td>any</td><td>Type specific data that describes any constraints on values of this attribute. When absent there are no constraints.

Note that the serialization of these "value constraints" shall be determined by the type of the attribute - see clause 5.11.1.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td><u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>capabilities</td><td>capability[]</td><td>A set of Provider defined metadata that can be used by Consumer to discover any capability or feature provided by this Provider.

Each capability shall contain the following nested data:
<table>
<tr><th>Name</th><td colspan="2">capability</td></tr>
<tr><th>Data</th><th>Type</th><th>Description</th></tr>
<tr><td>name</td><td>string</td><td>The name of the capability.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>uri</td><td>URI</td><td>A URI that uniquely identifies the capability at a global level.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>description</td><td>string</td><td>The human readable description of the semantic of the capability.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>value</td><td>any</td><td>The value of the capability. The specific type will vary depending on the definition of the capability. When not present the capability defaults to a "boolean" type with a value of "true" indicating that the specific capability is supported by the Provider.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
</table>
<u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>actions</td><td>action[]</td><td>A set of Provider defined operations that can be used by consumers to act on the resource. This set represents all operations defined for this described resource type, which may be a superset of those operations a particular Consumer is actually allowed to use. The subset of allowed operations for a particular Consumer shall be those operations returned to this Consumer when querying an instance of the described resource type. Note that this attribute is called "actions" so as not to conflict with the ResourceMetadata resource's own operations.

Each operation shall contain the following nested data:
<table>
<tr><th>Name</th><td colspan="2">action</td></tr>
<tr><th>Data</th><th>Type</th><th>Description</th></tr>
<tr><td>name</td><td>string</td><td>The name of the operation.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>uri</td><td>URI</td><td>A URI that uniquely identifies the operation at a global level.</td></tr>
</table></td></tr>
</table>

				Constraints: **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
		description	*string*	The human readable description of the semantic of the operation. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support optional; read-write
		method	*string*	The protocol dependent verb to use to perform the operation. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
		inputMessage	*string*	The body mimeType of the request message; it may depend on the model format chosen by the Provider. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
		outputMessage	*string*	The body mimeType of the response message; it may depend on the model format chosen by the Provider. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
		Constraints: **Provider:** support optional; mutable **Consumer:** support optional; read-write		

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadata",
  "id": string,
  "typeURI": string,
  "name": string,
  "attributes" : [
    { "name": string,
      "namespace": string, ?
      "type": string, ?
      "required": boolean, ?
      ...value constraints...? } *
  ], ?
  "capabilities": [
    { "name": string, ?
      "uri": string,
```

```
      "description": string, ?
      "value": any } *
  ], ?
  "actions" : [
    { "name": string,
      "uri": string,
      "description": string, ?
      "method": string,
      "inputMessage": string, ?
      "outputMessage": string ? }, *
  ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name>
  <typeURI> xs:anyURI </typeURI>
  <attribute name="xs:string" namespace="xs:anyURI"? type="xs:string"
             required="xs:boolean"? >
    ...value constraints...?
  </attribute> *
  <capability name="xs:string"? uri="xs:anyURI" description="xs:string"?>
    xs:any*
  </capability> *
  <action name="xs:string" uri="xs:anyURI" description="xs:string"?
          method="xs:string" inputMessage="xs:string"?
          outputMessage="xs:string"? /> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</ResourceMetadata>
```

Additional metadata about the resource or attributes may be included by the Provider.

5.11.1 Serialization of Attribute value constraints

The following describes the values, syntax, and serialization of the "value constraints" attribute (sub-attribute of "attributes"), which has a type of "any."

type="string"

The JSON shall be of the form:

```
"values": [ string, + ] ?
```

The XML shall be of the form:

```
<value> xs:string </value> *
```

type="integer"

The JSON shall be of the form:

```
"values": [ number, + ], ?
"ranges": [ { "low": number, "high": number }, + ] ?
```

The XML shall be of the form:

```
<value> xs:integer </value> *
<range low="xs:integer" high="xs:integer"/> *
```

The total value space of an 'integer' attribute is the accumulation of all values and ranges.

type="boolean"

The JSON shall be of the form:

```
"value": boolean ?
```

The XML shall be of the form:

```
<value> xs:boolean </value> ?
```

Only one 'value' is permitted. It indicates whether the attribute is required to be either 'true' or 'false'.

5.11.1.1 Examples

The following shows a sample metadata document for a VolumeConfiguration resource in XML that lists the allowable values for the "format" attribute and has been extended with a "Location" string attribute:

```
<ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/1">
  <id> http://example.org/types/VC </id>
  <typeURI> http://schemas.dmtf.org/cimi/1/VolumeConfiguration </typeURI>
  <name> VolumeConfiguration </name>
  <attribute name="format" type="string" required="false">
    <value> ext4 </value>
    <value> ntfs </value>
  </attribute>
  <attribute name="Location" namespace="http://example.org/" type="string"/>
</ResourceMetadata>
```

The following shows the same VolumeConfiguration but the "Location" attribute is restricted to a set of values and is required:

```
<ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/1">
  <id> http://example.org/types/VC </id>
  <typeURI> http://schemas.dmtf.org/cimi/1/VolumeConfiguration </typeURI>
  <name> VolumeConfiguration </name>
  <attribute name="format" type="string" required="false">
    <value> ext4 </value>
    <value> ntfs </value>
  </attribute>
  <attribute name="Location" namespace="http://example.org/" type="string"
             required="true">
    <value> NYC </value>
    <value> LAX </value>
  </attribute>
</ResourceMetadata>
```

The following shows the same VolumeConfiguration serialized in JSON:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
  "id": "http://example.org/types/VC",
  "typeURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
  "name": "VolumeConfiguration",
  "attributes": [
    { "name": "format",
      "type": "string",
      "required": false,
      "values": [ "ext4", "ntfs" ]
    },
    { "name": "Location",
      "namespace": "http://example.org",
      "type": "string",
      "required": true,
      "values": [ "NYC", "LAX" ]
    }
  ]
}
```

The following shows a Volume serialized in JSON which provides an action of data compression. In this specific example the method returned (POST) is for the CIMI HTTP protocol; should another protocol be implemented (e.g. SOAP) the "method" will be different:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
  "id": "http://example.org/types/V",
```

```
  "typeURI": "http://schemas.dmtf.org/cimi/1/Volume",
  "name": "Volume",
  "actions": [
    {
      "name": "compress",
      "uri": "http://example.org/cimi/action/compress"
      "description": "Compress the data stored in the volume",
      "method": "POST"
    }
  ]
}
```

5.11.2 Capabilities

The following table describes the capability URIs defined by this specification. Providers may define new URIs and it is recommended that these URIs be dereferencable such that Consumers can discover the details of the new capability. The "Resource Name" column contains the name of the resource that may contain the specified capability within its ResourceMetadata. The "Capability Name" column contains the name of the specified capability and shall be unique within the scope of the corresponding resource. Each capability's URI shall be constructed by appending the "Resource Name", a slash(/), and the "Capability Name" to "http://schemas.dmtf.org/cimi/1/capability/". For example, the Machine's "InitialState" capability shall have a URI of:

```
http://schemas.dmtf.org/cimi/1/capability/Machine/InitialState
```

Capabilities that apply to the Provider in general, and are not specific to any one resource, shall be associated with the Cloud Entry Point resource (in case a capability would apply only to the CloudEntryPoint resource itself, its definition would say so).

Each one of these capabilities may be set to some value, or may be absent. The meaning of an absent capability is defined as follows:

- For boolean-valued capabilities: same as a “false” value.
- For other capabilities that use a single value or a list of values among an enumeration: same as no particular preference or restriction being enforced for this value.

Resource Name	Capability Name	Description
CloudEntryPoint	ExpandParameter	If true, then the Provider shall support the $expand query parameter.
CloudEntryPoint	FilterParameter	If true, then the Provider shall support the $filter query parameter.
CloudEntryPoint	FirstParameter	If true, then the Provider shall support both the $first and $last query parameters.
CloudEntryPoint	SelectParameter	If true, then the Provider shall support the $select query parameter.
CloudEntryPoint	FormatParameter	If true, then the Provider shall support the $format query parameter.
CloudEntryPoint	OrderByParameter	If true, then the Provider shall support the $orderby query parameter.

Resource Name	Capability Name	Description
CloudEntryPoint	QueryPathNotation	If true, then the Provider shall support the use of path-like notation with query parameter $select (see 4.1.6.3) to disambiguate between attributes of a collection resource and attributes of each items in the collection when subsetting.
CloudEntryPoint	MaxPropertyItems	When set, the Provider shall support a 'Properties' attribute with a number of elements less than or equal to the size specified by this capability.
System	SystemComponentTemplateByValue	If true, then the Provider shall support the specification of ComponentTemplates by value in SystemTemplates.
Machine	DefaultInitialState	When this capability is set, unless otherwise provided (e.g. via a MachineTemplate "initialState" attribute), the Provider shall set a new Machine to this state value, assuming the value is compatible with the InitialStates capability if set.
Machine	InitialStates	When this capability is set, and when using a MachineTemplate that has an "initialState" attribute, then a Consumer shall use an initialState value from the set of values of this capability..
Machine	MachineConfigByValue	If true, then the Provider shall support specifying MachineConfigurations by value. If true, then the MachineTemplateByValue shall also have the value true.
Machine	MachineCredentialByValue	If true, then the Provider shall support specifying Credentials by value in Machine create operations. If true, then the MachineTemplateByValue capability shall also have the value true.
Machine	MachineImageByValue	If true, then the Provider shall support specifying MachineImages by value in Machine create operations. If true, then the MachineTemplateByValue capability shall also have the value true.
Machine	MachineVolumeTemplatesByValue	If true, then the Provider shall support specifying VolumeTemplates by value in Machine create operations. If true, then the MachineTemplateByValue capability shall also have the value true.
Machine	MachineTemplateByValue	If true, then the Provider shall support specifying MachineTemplates by value in Machine create operations.
Machine	MachineStopForce	If true, then the Provider shall support the "force" option on the stop and restart operations on Machines.
Machine	MachineStopForceDefault	If true, then the Provider shall forcefully stop Machines if no other indication is provided. Otherwise, the Provider shall gracefully stop Machines.
Machine	RestoreFromImage	If true, then the Provider supports restoring Machines from MachineImages that are not SNAPSHOT MachineImages.
Machine	UserData	When set, indicates which userData injection method shall be used by the Provider.
Machine	MachineAvailabilityLevel	If true, then the Provider supports the notion of an availability level for the Machine resource. The availability level and its value constraints will be advertised as an extension attribute via the Machine and

Resource Name	Capability Name	Description
		MachineTemplate ResourceMetadata.
Credential	CredentialTemplateByValue	If true, then the Provider shall support specifying CredentialTemplates by value in Credential create operations.
Volume	SharedVolumeSupport	If true, then the Provider shall support that a single Volume resource can be shared by multiple Machines.
Volume	VolumeConfigByValue	If true, then the Provider shall support specifying VolumeConfigurations by value in the Volume create operation. If true, then the VolumeTemplateByValue capability shall have the value true.
Volume	VolumeImageByValue	If true, then the Provider shall support specifying VolumeImages by value in the Volume create operation. If true, then the VolumeTemplateByValue capability shall have the value true.
Volume	VolumeSnapshot	If true, then the Provider shall support creating a new VolumeImage by referencing an existing Volume.
Volume	VolumeTemplateByValue	If true, then the Provider shall support specifying the VolumeTemplates by value in Volume create operations.
Volume	VolumeAvailabilityLevel	If true, then the Provider supports the notion of an availability level for the Volume resource. The availability level and its value constraints will be advertised as an extension attribute via the Volume and VolumeTemplate ResourceMetadata.
Network	NetworkConfigByValue	If true, then the Provider shall support specifying NetworkConfigurations by value in Network create operations.
Network	NetworkTemplateByValue	If true, then the Provider shall support specifying Network Templates by value in Network create operations.
Network	DefaultInitialState	When this capability is set, unless otherwise provided (e.g. via a NetworkTemplate "initialState" attribute), the Provider shall set a new Network to this state value, assuming the value is compatible with the InitialStates capability if set.
Network	InitialStates	When this capability is set, and when using a NetworkTemplate that has an “initialState” attribute, then a Consumer shall use an initialState value from the set of values of this capability.
NetworkPort	NetworkPortConfigByValue	If true, then the Provider shall support specifying NetworkPortConfigurations by value in NetworkPort create operations.
NetworkPort	NetworkPortTemplateByValue	If true, then the Provider shall support specifying NetworkPortTemplates by value in NetworkPort create operations.
NetworkPort	DefaultInitialState	When this capability is set, unless otherwise provided (e.g. via a NetworkPortTemplate "initialState" attribute), the Provider shall set a new NetworkPort to this state value, assuming the value is compatible with the InitialStates capability if set.
NetworkPort	InitialStates	When this capability is set, and when using a NetworkPortTemplate that has an “initialState” attribute, then a Consumer shall use an initialState value from the

Resource Name	Capability Name	Description
		set of values of this capability.
ForwardingGroup	MixedNetwork	If true, then a Provider shall support ForwardingGroups that can have both private and public connections at the same time. Otherwise, ForwardingGroups shall have only private or public connections at the same time.
Job	JobRetention	When set, the value of this capability shall indicate the minimum number of minutes a job shall be retained by the Provider before it is deleted.
Meter	MeterConfigByValue	If true, then the Provider shall support specifying MeterConfigurations by value in Meter create operations.
Meter	MeterTemplateByValue	If true, then the Provider shall support specifying MeterTemplates by value in Meter create operations.
EventLog	Linked	If true, then the Provider shall delete EventLogs that are associated with resources when the resource is deleted.

The following example shows the ResourceMetadata for a Machine that advertises some of its capabilities:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadata",
  "id": "http://example.com/types/Machine",
  "typeURI": "http://schemas.dmtf.org/cimi/1/Machine",
  "name": "Machine",
  "capabilities": [
    { "uri":
      "http://schemas.dmtf.org/cimi/1/capability/Machine/MachineConfigByValue",
      "value": true },
    { "uri":
      "http://schemas.dmtf.org/cimi/1/capability/Machine/MachineImageByValue",
      "value": true },
    { "uri":
      "http://schemas.dmtf.org/cimi/1/capability/Machine/DefaultInitialState",
      "value": "STARTED" }
  ]
}
```

XML serialization:

```
<ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/1">
  <id> http://example.org/types/Machine </id>
  <typeURI> http://schemas.dmtf.org/cimi/1/Machine </typeURI>
  <name> Machine </name>
  <capability
uri="http://schemas.dmtf.org/cimi/1/capability/Machine/MachineConfigByValue">
    true
```

```
  </capability>
  <capability
uri="http://schemas.dmtf.org/cimi/1/capability/Machine/MachineImageByValue">
    true
  </capability>
  <capability
uri="http://schemas.dmtf.org/cimi/1/capability/Machine/DefaultInitialState">
    STARTED
  </capability>
</ResourceMetadata>
```

5.11.3 ResourceMetadata Collection

A ResourceMetadata Collection resource represents the collection of ResourceMetadata resources within a Provider and follows the Collection pattern defined in clause 5.5.12. Note that modifications of the resources within this collection will typically be reserved for administrator type of CIMI Consumers. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadataCollection",
  "id": string,
  "count": number,
  "resourceMetadatas": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadata",
      "id": string,
      ... remaining ResourceMetadata attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/ResourceMetadataCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <ResourceMetadata>
    <id> xs:anyURI </id>
    ... remaining ResourceMetadata attributes ...
  </ResourceMetadata> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
```

```
</Collection>
```

5.12 Cloud Entry Point

The Cloud Entry Point represents the entry point into the cloud defined by the CIMI Model. The Cloud Entry Point implements a catalog of resources, such as Systems, System Templates, Machines, Machine Templates, etc., that can be queried and browsed by the Consumer.

Figure 1 illustrates the Cloud Entry Point and its relationship to other resources. Although this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor normative.

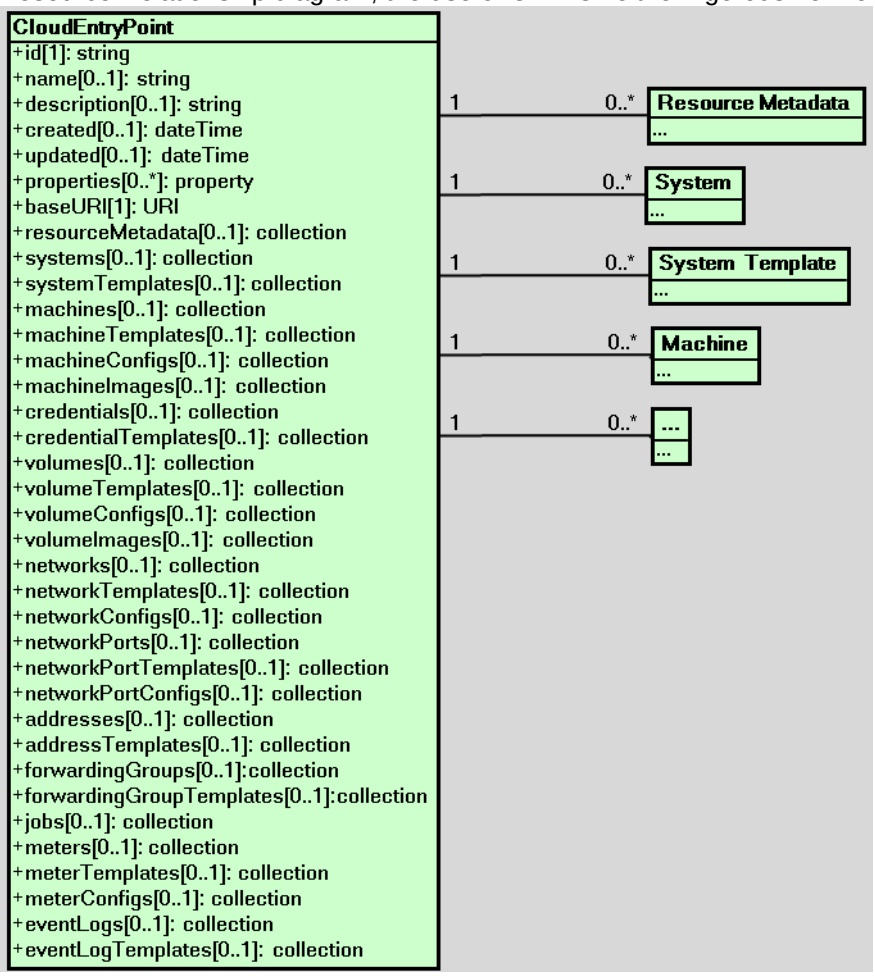


Figure 1 - Cloud Entry Point

When a Consumer issues a read on the Cloud Entry Point resource, then the Provider shall return a Cloud Entry Point resource that only catalogs resources that this Consumer is allowed to perform operations on.

Name	CloudEntryPoint	
Type URI	http://www.dmf.org/cimi/CloudEntryPoint	
Attribute	**Type**	**Description**
baseURI	*URI*	An absolute URI that references the "base URI" of the Provider. This URI shall be used to convert relative URIs to resources within this Provider to absolute URIs. See the "URIs" clause of 5.5.

		Constraints: **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
resourceMetadata	*collection [ResourceMetadata]*	A reference to ResourceMetadata Collection of this Cloud Entry Point. The collection contains the resources supported by the Provider. If an resource does not have any metadata, it shall not appear in this list, e.g., it has no constraints beyond what the CIMI specification defines nor does it have any extension attributes. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
systems	*collection [System]*	A reference to the System Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
systemTemplates	*collection [SystemTemplate]*	A reference to the System Template Collection of this CloudEntry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
machines	*collection [Machine]*	A reference to the Machine Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
machineTemplates	*collection [MachineTemplate]*	A reference to the Machine Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
machineConfigs	*collection [MachineConfiguration]*	A reference to the Machine Configuration Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
machineImages	*collection [MachineImage]*	A reference to the Machine Image Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
credentials	*collection [Credential]*	A reference to the Credential Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
credentialTemplates	*collection [CredentialTemplate]*	A reference to the Credential Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
volumes	*collection [Volume]*	A reference to the Volume Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only

volumeTemplates	*collection [VolumeTemplate]*	A reference to the Volume Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
volumeConfigs	*collection [VolumeConfiguration]*	A reference to the Volume Configuration Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
volumeImages	*collection [VolumeImage]*	A reference to the Volume Image Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
networks	*collection [Network]*	A reference to the Network Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
networkTemplates	*collection [NetworkTemplate]*	A reference to the Network Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
networkConfigs	*collection [NetworkConfiguration]*	A reference to the Network Configuration Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
networkPorts	*collection [NetworkPort]*	A reference to the Network Port Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
networkPortTemplates	*collection [NetworkPortTemplate]*	A reference to the Network Port Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
networkPortConfigs	*collection [NetworkPortConfiguration]*	A reference to the Network Port Configuration Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
addresses	*collection [Address]*	A reference to the Address Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
addressTemplates	*collection [AddressTemplate]*	A reference to the Address Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
forwardingGroups	*collection*	A reference to the Forwarding Group Collection of this Cloud Entry Point.

	[ForwardingGroup]	**Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
forwardingGroupTemplates	*collection [ForwardingGroupTemplate]*	A reference to the Forwarding Group Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
jobs	*collection [Job]*	A reference to the Jobs Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
meters	*collection [Meter]*	A reference to the Meter Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
meterTemplates	*collection [MeterTemplate]*	A reference to the Meter Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
meterConfigs	*collection [MeterConfiguration]*	A reference to the Meter Configuration Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
eventLogs	*collection [EventLog]*	A reference to the Event Log Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
eventLogTemplates	*collection [EventLogTemplate]*	A reference to the Event Log Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only

Each of the collections mentioned above will be defined within the related resource definition clauses. For example, the MachineCollection resource will be defined in clause 5.14.2 as part of the Machine related resources.

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
```

```
  "baseURI": string,
  "resourceMetadata": { "href": string }, ?
  "systems": { "href": string }, ?
  "systemTemplates": { "href": string }, ?
  "machines": { "href": string }, ?
  "machineTemplates": { "href": string }, ?
  "machineConfigs": { "href": string }, ?
  "machineImages": { "href": string }, ?
  "credentials": { "href" string }, ?
  "credentialTemplates": { "href" string }, ?
  "volumes": { "href": string }, ?
  "volumeTemplates": { "href": string }, ?
  "volumeConfigs": { "href": string }, ?
  "volumeImages": { "href": string }, ?
  "networks": { "href": string }, ?
  "networkTemplates": { "href": string }, ?
  "networkConfigs": { "href": string }, ?
  "networkPorts": { "href": string }, ?
  "networkPortTemplates": { "href": string }, ?
  "networkPortConfigs": { "href": string }, ?
  "addresses": { "href": string }, ?
  "addressTemplates": { "href": string }, ?
  "forwardingGroups" { "href": string }, ?
  "forwardingGroupTemplates" { "href": string }, ?
  "jobs": { "href": string }, ?
  "meters": { "href": string }, ?
  "meterTemplates": { "href": string }, ?
  "meterConfigs": { "href": string }, ?
  "eventLogs": { "href": string }, ?
  "eventLogTemplates": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<CloudEntryPoint xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
```

```
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <baseURI> xs:anyURI </baseURI>
  <resourceMetadata href="xs:anyURI"/> ?
  <systems href="xs:anyURI"/> ?
  <systemTemplates href="xs:anyURI"/> ?
  <machines href="xs:anyURI"/> ?
  <machineTemplates href="xs:anyURI"/> ?
  <machineConfigs href="xs:anyURI"/> ?
  <machineImages href="xs:anyURI"/> ?
  <credentials href="xs:anyURI"/> ?
  <credentialTemplates href="xs:anyURI"/> ?
  <volumes href="xs:anyURI"/> ?
  <volumeTemplates href="xs:anyURI"/> ?
  <volumeConfigs href="xs:anyURI"/> ?
  <volumeImages href="xs:anyURI"/> ?
  <networks href="xs:anyURI"/> ?
  <networkTemplates href="xs:anyURI"/> ?
  <networkConfigs href="xs:anyURI"/> ?
  <networkPorts href="xs:anyURI"/> ?
  <networkPortTemplates href="xs:anyURI"/> ?
  <networkPortConfigs href="xs:anyURI"/> ?
  <addresses href="xs:anyURI"/> ?
  <addressTemplates href="xs:anyURI"/> ?
  <forwardingGroups href="xs:anyURI"/> ?
  <forwardingGroupTemplates href="xs:anyURI"/> ?
  <jobs href="xs:anyURI"/> ?
  <meters href="xs:anyURI"/> ?
  <meterTemplates href="xs:anyURI"/> ?
  <meterConfigs href="xs:anyURI"/> ?
  <eventLogs href="xs:anyURI"/> ?
  <eventLogTemplates href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</CloudEntryPoint>
```

5.12.1 Operations

This resource supports the Read and Update operations.

5.13 System resources and relationships

Figure 2 illustrates the resources involved in constructing a System and their relationships. Although this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor normative.

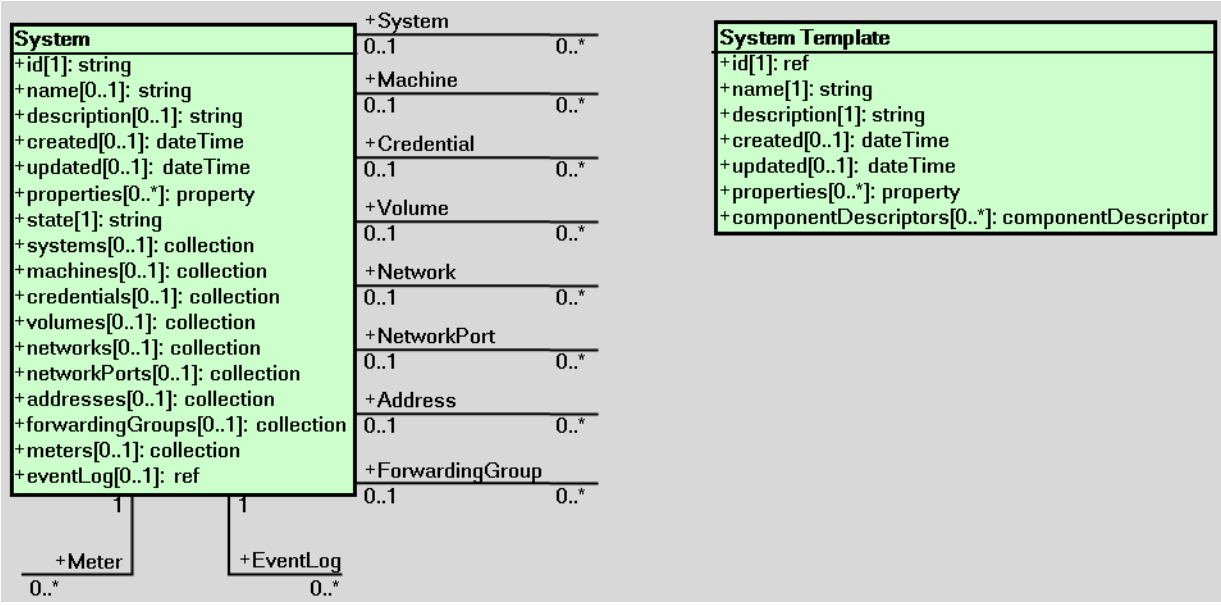


Figure 2 - System resources

5.13.1 System

A System is a realized resource that consists of one or more Networks, Volumes, Machines, (and others) that could be connected and associated with each other. A System can be created from the interpretation of a SystemTemplate. A System can be operated and managed as a single resource and usually forms a stack of service. For example, a shopping cart system consists of machines for web servers and databases, network addresses for public access, and volumes for database files. A System may directly provide a user-facing component, or may provide an infrastructure component.

A System has several "top-level" attributes that are collections of references to resources that are owned by the System. A resource that is owned by a System has its lifecycle directly tied to the lifecycle of the System. In particular, when a System is deleted, all of its owned resources shall also be deleted. Generally, operations on a System will translate into operations on its owned resources.

However, a resource owned by a System may in turn refer to some other resources that are not owned by this System, e.g., a Machine in a System can refer to a Volume that is not owned by this System. More precisely, the following rules apply:

- By default, all resources that are created as the result of a System creation are also owned by the System. (This rule can be overridden by subsequent modifications to the top-level System collection attributes.)
- Ownership of a resource by a System is expressed by including the reference to the resource in the appropriate top-level System collection attribute, or by ownership to a sub-System of this System (i.e., ownership is transitive across hierarchies of Systems).

- When a resource other than a System is added to an existing System (i.e., becomes owned by the System by insertion of its reference to the appropriate top-level System collection attribute) other resources already referred by this added resource are by default not owned by the System. (This rule can be overridden by subsequent modifications to the top-level System collection attributes.)

A resource shall not be owned by more than one System at any point in time (unless there is an ownership relationship between these Systems). Note that a resource does not need to owned by a System (i.e. part of any of its collection attributes) to be references/used by a resource in the System. By not including it in any of the collections, the resource will simply not be part of any actions performed on the System.

Name	System	
Type URI	http://schemas.dmtf.org/cimi/1/System	
Attribute	**Type**	**Description**
state	*string*	The operational state of the System. Allowable values include:5.14.1 **CREATING**: The System is in the process of being created. **STARTING/STARTED/STOPPING/STOPPED/PAUSING/PAUSED/SUSPENDING /SUSPENDED**: The System shall be in one of these states if all the Machines referenced by the System are in that state. See clause 5.14.1for the list of available actions based on the state of a Machine. Such transitional states may just indicate that all Machines in a System are undergoing the same operation (e.g. “start”), without the System being actually operated on (e.g. no “start” done at System level). An actual operation on a System may be traced by querying the “job” entity. **MIXED**: The System shall be in this state if either no Machines are referenced by this System or the Machines referenced by this System are in varying states. Such varying states are likely to occur when an operation is in progress on a system, resulting in transitions of its Machine states toward a new common state (e.g. STOPPED, STARTED) but at a different pace, or sequentially one after the other. **DELETING**: The System is in the process of being deleted. **ERROR**: The Provider has detected an error in the System. The operations that result in transitions to the above defined states are defined in Section 5.13.1.2 **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
systems	*collection [SystemSystem]*	A reference to the list of references to nested Systems owned by this System. Adding an item (of type System) to this list is logically equivalent to associating the referenced System to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the referenced System from this System. Note: the SystemSystem resource type is representing an association between the System and another System. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
machines	*collection [SystemMachine]*	A reference to the list of references to Machines owned by this System. Adding an item (of type Machine) to this list is logically equivalent to associating the Machine to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the Machine from this System. Note: the SystemMachine resource type is representing an association between

		the System and a Machine. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
credentials	*collection [SystemCredential]*	A reference to the list of references to Credentials owned by this System. Adding an item (of type Credential) to this list is logically equivalent to associating the Credential to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the Credential from this System. Note: the SystemCredential resource type is representing an association between the System and a Credential. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
volumes	*collection [SystemVolume]*	A reference to the list of references Volumes owned by this System. Adding an item (of type Volume) to this list is logically equivalent to associating the Volume to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the Volume from this System. Note: the SystemVolume resource type is representing an association between the System and a Volume. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
networks	*collection [SystemNetwork]*	A reference to the list of references Networks owned by this System. Adding an item (of type Network) to this list is logically equivalent to associating the Network to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the Network from this System. Note: the SystemNetwork resource type is representing an association between the System and a Network. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
networkPorts	*collection [SystemNetworkPort]*	A reference to the list of references NetworkPorts owned by this System. Adding an item (of type NetworkPort) to this list is logically equivalent to associating the NetworkPort to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the NetworkPort from this System. Note: the SystemNetworkPort resource type is representing an association between the System and a NetworkPort. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
addresses	*collection [SystemAddress]*	A reference to the list of references Addresses owned by this System. Adding an item (of type Address) to this list is logically equivalent to associating the Address to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the Address from this System. Note: the SystemAddress resource type is representing an association between the System and a Address. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
forwardingGroups	*collection [SystemFor*	A reference to the list of references Forwarding Groups owned by this System. Adding an item (of type ForwardingGroup) to this list is logically equivalent to

	wardingGroup]	associating the Forwarding Group to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the Forwarding Group from this System. Note: the SystemForwardingGroup resource type is representing an association between the System and a ForwardingGroup. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
meters	*collection [Meter]*	A reference to the list of Meters monitored for this System. Note that these Meters are for the System and not for any individual component in the System. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
eventLog	*ref*	A reference to the EventLog of this System. Note that this EventLog is for the System and not for any individual component in the System. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "systems": { "href": string }, ?
  "machines": { "href": string }, ?
  "credentials": { "href": string }, ?
  "volumes": { "href": string }, ?
  "networks": { "href": string }, ?
  "networkPorts": { "href": string }, ?
  "addresses": { "href": string }, ?
  "forwardingGroups": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
```

```
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/restart", "href": string },
?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/pause", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/suspend", "href": string },
?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/export", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<System xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <systems href="xs:anyURI"/> ?
  <machines href="xs:anyURI"/> ?
  <credentials href="xs:anyURI"/> ?
  <volumes href="xs:anyURI"/> ?
  <networks href="xs:anyURI"/> ?
  <networkPorts href="xs:anyURI"/> ?
  <addresses href="xs:anyURI"/> ?
  <forwardingGroups href="xs:anyURI"/> ?
  <meters href="xs:anyURI"/> ?
  <eventLog href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
             href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
             href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/restart"
             href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/pause"
```

```
              href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/suspend"
              href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/export"
              href="xs:anyURI"/> ?
  <xs:any>*
</System>
```

5.13.1.1 Collections

The following describes the collection resources owned by Systems.

5.13.1.1.1 SystemSystem Collection

The resource type for each item of this collection is “SystemSystem”, defined as follows:

Name	SystemSystem	
Type URI	http://schemas.dmtf.org/cimi/1/SystemSystem	
Attribute	**Type**	**Description**
system	*ref*	Reference to a System resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemSystemCollection",
  "id": string,
  "count": number,
  "systemSystems": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemSystem",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "system": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
```

```
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemSystemCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <SystemSystem>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <system href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </SystemSystem> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.1.1.2 SystemMachine Collection

The resource type for each item of this collection is “SystemMachine”, defined as follows:

Name	SystemMachine	
Type URI	http://schemas.dmtf.org/cimi/1/SystemMachine	
Attribute	**Type**	**Description**
machine	*ref*	Reference to a Machine resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemMachineCollection",
  "id": string,
  "count": number,
  "systemMachines": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemMachine",
      "id": string,
```

```
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "machine": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemMachineCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <SystemMachine>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <machine href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </SystemMachine> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.1.1.3 SystemCredential Collection

The resource type for each item of this collection is “SystemCredential”, defined as follows:

Name	SystemCredential	
Type URI	http://schemas.dmtf.org/cimi/1/SystemCredential	
Attribute	**Type**	**Description**
credential	*ref*	Reference to a Credential resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCredentialCollection",
  "id": string,
  "count": number,
  "systemCredentials": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCredential",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "credential": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemCredentialCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <SystemCredential>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
```

```
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <credential href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </SystemCredential> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.1.1.4 SystemVolume Collection

The resource type for each item of this collection is “SystemVolume”, defined as follows:

Name	SystemVolume	
Type URI	http://schemas.dmtf.org/cimi/1/SystemVolume	
Attribute	**Type**	**Description**
volume	*ref*	Reference to a Volume resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemVolumeCollection",
  "id": string,
  "count": number,
  "systemVolumes": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemVolume",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "volume": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
```

```
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemVolumeCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <SystemVolume>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <volume href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </SystemVolume> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.1.1.5 SystemNetwork Collection

The resource type for each item of this collection is “SystemNetwork”, defined as follows:

Name	SystemNetwork	
Type URI	http://schemas.dmtf.org/cimi/1/SystemNetwork	
Attribute	**Type**	**Description**
network	*ref*	Reference to a Network resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetworkCollection",
  "id": string,
  "count": number,
```

```
  "systemNetworks": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetwork",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "network": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemNetworkCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <SystemNetwork>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <network href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </SystemNetwork> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.1.1.6 SystemNetworkPort Collection

The resource type for each item of this collection is “SystemNetwork”, defined as follows:

Name	SystemNetworkPort	
Type URI	http://schemas.dmtf.org/cimi/1/SystemNetworkPort	
Attribute	**Type**	**Description**
networkPort	*ref*	Reference to a NetworkPort resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetworkPortCollection",
  "id": string,
  "count": number,
  "systemNetworkPorts": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetworkPort",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "networkPort": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemNetworkPortCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <SystemNetworkPort>
```

```
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <networkPort href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </SystemNetworkPort> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.1.1.7 SystemAddress Collection

The resource type for each item of this collection is “SystemAddress”, defined as follows:

Name	SystemAddress	
Type URI	http://schemas.dmtf.org/cimi/1/SystemAddress	
Attribute	**Type**	**Description**
address	*ref*	Reference to a Address resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemAddressCollection",
  "id": string,
  "count": number,
  "systemAddresses": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemAddress",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "address": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
```

```
      ] ?
      ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemAddressCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <SystemAddress>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <address href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </SystemAddress> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.1.1.8 SystemForwardingGroup Collection

The resource type for each item of this collection is “SystemForwardingGroup", defined as follows:

Name	SystemForwardingGroup	
Type URI	http://schemas.dmtf.org/cimi/1/SystemForwardingGroup	
Attribute	**Type**	**Description**
forwardingGroup	*ref*	Reference to a ForwardingGroup resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

JSON serialization:

```
{ "resourceURI":
```

```
    "http://schemas.dmtf.org/cimi/1/SystemForwardingGroupCollection",
  "id": string,
  "count", number,
  "systemForwardingGroups": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemForwardingGroup",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "forwardingGroup": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
 resourceURI="http://schemas.dmtf.org/cimi/1/SystemForwardingGroupCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <SystemForwardingGroup>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <forwardingGroup href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </SystemForwardingGroup> *
```

```
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.1.1.9 SystemMeter Collection

The resource type for each item of this collection is “Meter” as defined in clause 5.17.3.

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemMeterCollection",
  "id": string,
  "count": number,
  "meters": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
      "id": string,
      ... remaining Meter attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/SystemMeterCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Meter>
    <id> xs:anyURI </id>
    ... remaining Meter attributes ...
  </Meter> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.1.2 Operations

The System resource supports the Read, Update, and Delete operations. Create is supported via the System Collection resource.

The following custom operations are also defined:**start/stop/restart/pause/suspend**

/link@rel: http://schemas.dmtf.org/cimi/1/action/xxx

Where "xxx" is either "start", "stop", "restart", "pause", or "suspend".

This operation shall recursively perform the requested operation on each component of the System (Machine or sub-System). Note that not all Machines need to be in the same state for this operation to be

available and the impact that this operation will have will vary depending on the component's current state; see clause 5.14.1.2 for more details about performing operations on Machines If the operation fails for a Machine, then that Machine shall not be affected by the operation.

export

/link@rel: http://schemas.dmtf.org/cimi/1/action/export

This operation shall export a System. If an export package exists at that URI, it is updated with the values of the System and any component management resources. Otherwise, a new export package is created at that URI with a Media Type as specified by the "format" parameter. Other formats may be used if supported, but are not specified by this standard.

Input parameters:

(1) "format" - type: string - optional
Indicates the Media Type of the exported data. If not present, the default value shall be "application/ovf."

(2) "destination" - type: URI - optional
The location to where the exported data is placed. If not present, the HTTP response Location header shall contain the URL to the exported data. Based on the specific protocol specified within the URI, the Consumer might need to provide additional information (such as credentials) in the "properties" field. In the case of HTTP, a PUT shall be used to place the data at the specified location.

Output parameters: None.

HTTP protocol

To export a System, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/export" URI of the System where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "action": "http://schemas.dmtf.org/cimi/1/action/export",
  "format": string, ?
  "destination": string, ?
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/export </action>
  <format> xs:string </format> ?
  <destination> xs:anyURI </destination> ?
  <property key="xs:string"> xs:string </property> *
```

```
  <xs:any>*
</Action>
```

5.13.2 System Collection

A System Collection resource represents the collection of System resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCollection",
  "id": string,
  "count", number,
  "systems": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
      "id": string,
      ... remaining System attributes ...
    }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/import", "href": string } ?
  ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/SystemCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <System>
    <id> xs:anyURI </id>
    ... remaining System attributes ...
  </System> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/import" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.2.1 Operations

NOTE: The "add" operation requires a SystemTemplate to be used (see 4.2.1.1).

Resources created during the process of creating a System shall be "owned" by the System (see 5.13.1). For example, a "componentDescriptor" that references a MachineTemplate, and within that MachineTemplate is a reference to a VolumeTemplate, will result in a reference to the new Machine

being added to the System.machines attribute and a reference to the new Volume being added to the System.volumes attribute. However, if this MachineTemplate refers to an existing Volume, this Volume shall not be added to the top-level System attributes.

The following custom operations are also defined:**import**

/link@rel: http://schemas.dmtf.org/cimi/1/action/importThis operation shall import a System. Not only will a System be created, but Machines, Volumes, and Networks and possibly recursive Systems and their components may also be created corresponding to imported descriptor entries. More detail about this process is in ANNEX A.

(1) Input parameters:"source" - type: URI - mandatory
The location from which the imported data will be retrieved. Based on the specific protocol specified within the URI, the Consumer might need to provide additional information (such as credentials) in the "properties" field.

Output parameters: None.

HTTP protocol

To import a System, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/import" URI of the System Collection where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "action": "http://schemas.dmtf.org/cimi/1/action/import",
  "source": string, ?
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/import </action>
  <source> xs:anyURI </source> ?
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

5.13.3 System Template

The System Template contains the set of individual descriptors that are necessary to create the components of a System. Each component descriptor can be considered to be the persisted view of the create operation that instantiates the component. In practice, the Provider will interpret the set of component descriptors as a set of creation operations to be executed in an order compatible with the dependencies (e.g., attachments or references between components) that are manifest between these components.

A System Template may include component references in the descriptors, used to express links between components of the resulting System. A component reference uses the "name" of the target (referred) component. For example, <volume href="#newVolume"/> would reference a Volume named

2566 "newVolume." The reference name – here #newVolume – will be replaced by the actual resource URL in
2567 the instantiated System.

2568 A SystemTemplate shall not contain two component descriptors of the same type that would result in the
2569 same non-null value for the "name" attribute of resulting components. Attempting to create or to update a
2570 SystemTemplate that fails this rule shall result in an error.

2571

<table>
<tr><td>Name</td><td colspan="2">SystemTemplate</td></tr>
<tr><td>Type URI</td><td colspan="2">http://schemas.dmtf.org/cimi/1/SystemTemplate</td></tr>
<tr><td>Attribute</td><td>Type</td><td>Description</td></tr>
<tr><td>component Descriptors</td><td>component Descriptor[]</td><td>The list of component descriptors describing the components of a System instance realized from this SystemTemplate. For each component descriptor, the corresponding component is created when a System instance is created. Each component descriptor refers to a template (either by reference or value), and may also provide additional metadata (name, description, properties). The creation order of components is not specified in SystemTemplate, in particular the order of the component descriptors in this array is not meaningful in terms of creation order.
<table>
<tr><td>Name</td><td colspan="2">componentDescriptor</td></tr>
<tr><td>Data</td><td>Type</td><td>Description</td></tr>
<tr><td>name</td><td>string</td><td>The value of the "name" attribute that will be associated with a System component created from this component descriptor. Note: This name is not to be confused with the name that may be present in the component template – e.g., a MachineTemplate – from which this component will be instantiated.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>description</td><td>string</td><td>The value of the "description" attribute that will be associated with a System component created from this component descriptor.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>properties</td><td>map</td><td>The key/value pairs that will be associated with a System component created from this component descriptor.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>type</td><td>URI</td><td>The TypeURI of the component to be created from this component descriptor, e.g., for a machine:

http://schemas.dmtf.org/cimi/1/Machine

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>component Template</td><td>any</td><td>Reference either to a component Template or the Template data itself inlined (i.e., the Template "value").</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>Note that the exact name of this attribute will vary depending on the type of resource being created, e.g., MachineTemplate for a Machine.

This attribute shall contain either:

• A template that is provided inline. Such an embedded template may contain component references, each one of which shall resolve to the URI of a component with same name once created from this SystemTemplate.

• A reference to an externally defined template, Some attribute name / value pairs may be added inside the componentTemplate element to override similar attributes in the referred template (as described in 4.2.1.1). This is how component references can be added to an external template.

Example (JSON):

<pre>"machineTemplate": {
    "href":
"http://example.com/machineTemplates/72000",
     "credential": { "href":
"#MyCredential" }

  }</pre>
This “credential” attribute assumes that there is another componentDescriptor item named “MyCredential" of type “Credential” in the SystemTemplate. It shall set or override similar attribute in the referred MachineTemplate when instantiating the Machine component.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>quantity</td><td>integer</td><td>Number of component instances to be created from this component descriptor. By default, this number is equal to 1. When the value is 2 or more, the actual name assigned to each instance will be the "name" value concatenated with a sequential number (e.g., if name="mymachine", and quantity=3, the names will be: mymachine1, mymachine2, mymachine3.)

<u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
</table>
<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>meterTempl</td><td>meterTemp</td><td>A list of references to Meter Templates that shall be used to create and connect a set of</td></tr>
</table>

ates	*lates[]*	new Meters to the new System. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
eventLogTemplate	*ref*	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new System. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
importImage	*ref*	When the template is the result of an import – e.g. of an OVF package - this attribute should be used. When present it shall reference the import source (e.g. OVF package) used to create this template. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "componentDescriptors": [
    { "name": string, ?
      "description": string, ?
      "properties": { string: string, + }, ?
      "type": string,
      "componentTemplate": {
        "href": string, ?
        ... ComponentTemplate attributes ... ?
      },
       "quantity": number ?
    }, +
  ], ?
  "meterTemplates": [
    { "href": string, ?
      ... MeterTemplate attributes ... ?
```

```
    }, *
  ], ?
  "eventLogTemplate": {
    "href": string, ?
    ... EventLogTemplate attributes ... ?
  }, ?
  "importImage": { "href": string }, ?

  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/export", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<SystemTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <componentDescriptor>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <property key="xs:string"> xs:string </property> *
    <type> xs:anyURI </type>
    <componentTemplate href="xs:anyURI"? >
      ... ComponentTemplate attributes ... ?
    </componentTemplate> *

    <quantity> xs:integer </quantity>
  </componentDescriptor> *
  <meterTemplate href="xs:anyURI"? >
    ... MeterTemplate attributes ... ?
  </meterTemplate> *
  <eventLogTemplate href="xs:anyURI"? >
```

```
    ... EventLogTemplate attributes ... ?
  </eventLogTemplate> ?
  <importImage href="xs:anyURI"? >
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/export"
href="xs:anyURI"/> ?
  <xs:any>*
</SystemTemplate>
```

5.13.3.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the System Template Collection resource.

The following custom operations are also defined:**export**

/link@rel: http://schemas.dmtf.org/cimi/1/action/exportThis operation shall export a System Template. If an export package exists at that URI, it is updated with the values of the System Template and any component management resources. Otherwise a new export package is created at that URI with a Media Type as specified by the "format" parameter. Other formats may be used if supported, but are not specified by this standard.

Input parameters:

(1) "format" - type: string - optional
Indicates the Media Type of the exported data. If not present, the default value shall be "application/ovf."

(2) "destination" - type: URI - optional
The location to where the exported data is placed. If not present, the HTTP response Location header shall contain the URL to the exported data. Based on the specific protocol specified within the URI, the Consumer might need to provide additional information (such as credentials) in the "properties" field. In the case of HTTP, a PUT shall be used to place the data at the specified location.

Output parameters: None.

HTTP protocol

To export a SystemTemplate, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/export" URI of the System Template where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "action": "http://schemas.dmtf.org/cimi/1/action/export",
  "format": string, ?
  "destination": string, ?
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/export </action>
  <format> xs:string </format> ?
  <destination> xs:anyURI </destination> ?
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

5.13.4 System Template Collection

A System Template Collection resource represents the collection of System Template resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplateCollection",
  "id": string,
  "count": number,
  "systemTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
      "id": string,
      ... remaining SystemTemplate attributes ...
    }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/import", "href": string } ?
  ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <SystemTemplate>
    <id> xs:anyURI </id>
    ... remaining SystemTemplate attributes ...
  </SystemTemplate> *
```

```
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/import" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.4.1 Operations

The following custom operations are defined:**import**

/link@rel: http://schemas.dmtf.org/cimi/1/action/import

This operation shall import a SystemTemplate. Not only will a System Template be created, but Machine Templates, Volume Templates, and Network Templates and possibly recursive System Templates and their components may also be created, corresponding to imported descriptor entries. More detail about this process is in ANNEX A.

Input parameters:

(1) "source" - type: URI - mandatory
The location from which the imported data will be retrieved. Based on the specific protocol specified within the URI, the Consumer might need to provide additional information (such as credentials) in the "properties" field.

Output parameters: None.

HTTP protocol

To import a SystemTemplate, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/import" URI of the System Template Collection where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "action": "http://schemas.dmtf.org/cimi/1/action/import",
  "source": string, ?
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/import </action>
  <source> xs:anyURI </source> ?
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

5.14 Machine resources and relationships

Figure 3 illustrates the resources involved in constructing a Machine and their relationships. Although this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor normative.

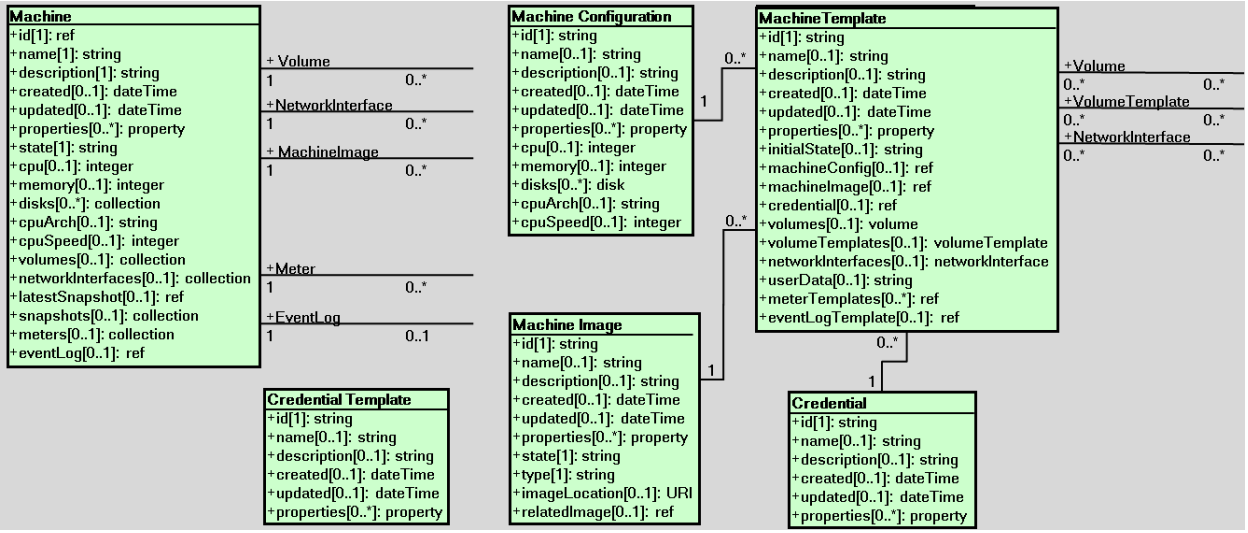


Figure 3 - Machine resources

5.14.1 Machine

An instantiated compute resource that encapsulates both CPU and Memory.

Name	Machine	
Type URI	http://schemas.dmtf.org/cimi/1/Machine	
Attribute	**Type**	**Description**
state	*string*	The operational state of the Machine. Allowable values include: **CREATING**: The Machine is in the process of being created. **STARTING**: The Machine is in the process of being started. **STARTED**: The Machine is available and ready for use. **STOPPING**: The Machine is in the process of being stopped. **STOPPED**: This value is the virtual equivalent of powering off a physical Machine. There is no saved CPU or memory state. Section 5.14.2.1 defines the initial state of a Machine. **PAUSING**: The Machine in the process of being PAUSED. **PAUSED**: In this state the Machine and its virtual resources remain instantiated and resources remain allocated, similar to the "STARTED" state, but the Machine and its virtual resources are not enabled to perform tasks. **SUSPENDING**: The Machine is in the process of being suspended. **SUSPENDED**: In this state the Machine and its virtual resources are stored on non-volatile storage. The Machine and its resources are not enabled to perform tasks. **DELETING**: The Machine is in the process of being deleted.

		ERROR: The Provider has detected an error in the Machine. The operations that result in transitions to the above defined states are defined in Section 5.14.1.2. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
cpu	*integer*	The amount of CPU that this Machine has. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
memory	*integer*	The size of the memory (RAM) in kibibytes allocated to this Machine. When this value is increased, it implies that the Machine is allocated more RAM, and vice versa when the value is decreased. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
disks	*collection [Disk]*	A reference to the list of disks (local storage) that are part of the Machine. Adding an element to this list creates a disk. Note: the Disk resource type is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
cpuArch	*string*	The CPU architecture that will be supported by Machines created by using this configuration. Allowable values include: **68000**, **Alpha**, **ARM**, **Itanium**, **MIPS**, **PA_RISC**, **POWER**, **PowerPC**, **x86**, **x86_64**, **z/Architecture**, **SPARC**. Providers may define additional values. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only
cpuSpeed	*integer*	The approximate CPU speed of this Machine - in megahertz. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
volumes	*collection [MachineVolume]*	A reference to the list of references to Volumes that are connected to this Machine. Adding a Volume to this list means that the Machine has some access to the data on the Volume. Removing a Volume from this list means that the Machine no longer has access to the data on the Volume. Note: the MachineVolume resource type is representing an association between the Machine and a Volume. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
networkInterfaces	*collection [MachineNetwork Interface]*	A reference to the list of MachineNetworkInterfaces on this Machine. Note: the MachineNetworkInterface resource type is representing an association between the Machine and a NetworkInterface. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable

		Consumer: support optional; read-only
latestSnapshot	*ref*	A reference to the SNAPSHOT representing the latest state captured for this Machine (either most recent Snapshot or the last Snapshot reverted to). **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
snapshots	*collection [MachineSnapshot]*	A reference to the list of references to the SNAPSHOT Machine Images taken of this Machine. Note: the MachineSnapshot resource type is representing an association between the Machine and a Snapshot. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
meters	*collection [Meter]*	A reference to the list of Meters monitored for this Machine. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
eventLog	*ref*	A reference to the EventLog of this Machine. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "cpu": number,
  "memory": number,
  "disks" : { "href": string }, ?
  "cpuArch": string, ?
  "cpuSpeed": number, ?
  "volumes": { "href": string }, ?
  "networkInterfaces": { "href": string }, ?
  "latestSnapshot": { "href": string }, ?
  "snapshots": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
```

```
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/restart", "href": string },
?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/pause", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/suspend", "href": string }
?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/snapshot", "href": string }
?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/restore", "href": string }
?
  ]
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Machine xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <cpu> xs:integer </cpu>
  <memory> xs:integer </memory>
  <disks href="xs:anyURI"/> ?
  <cpuArch> xs:string </cpuArch> ?
  <cpuSpeed> xs:integer </cpuSpeed> ?
  <volumes href="xs:anyURI"/> ?
  <networkInterfaces href="xs:anyURI"/> ?
  <latestSnapshot href="xs:anyURI"/> ?
  <snapshots href="xs:anyURI"/> ?
  <meters href="xs:anyURI"/> ?
  <eventLog href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
```

```
href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/restart"
href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/pause"
href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/suspend"
href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/capture"
href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/snapshot"
href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/restore"
href="xs:anyURI"/> ?
  <xs:any>*
</Machine>
```

5.14.1.1 Collections

The following describes the collection resources owned by Machines.

5.14.1.1.1 Disk Collection

The resource type for each item of this collection is "Disk", as defined as follows:

Name	Disk	
Type URI	http://schemas.dmtf.org/cimi/1/Disk	
Attribute	**Type**	**Description**
capacity	*integer*	The initial capacity, in kilobytes, of the disk. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
initialLocation	*string*	Operating System specific location(path) in its namespace where this disk will first appear. Note, once deployed Consumers might move where this Disk is located. Support of this attribute indicates that the Provider can report this information back to the Consumer. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/DiskCollection",
  "id": string,
  "count": number,
  "disks": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Disk",
      "id": string,
      "name": string, ?
```

```
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "capacity": number,
      "initialLocation": string, ?
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/DiskCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Disk>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <capacity> xs:integer </capacity>
    <initialLocation> xs:string </initialLocation> ?
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </Disk> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.1.1.2 MachineVolume Collection

The resource type for each item of this collection is “MachineVolume”, defined as follows:

Name	MachineVolume	
Type URI	http://schemas.dmtf.org/cimi/1/MachineVolume	
Attribute	**Type**	**Description**
initialLocation	*string*	Operating System specific location(path) in its namespace where this Volume will first appear. Note, once deployed Consumers might move where this Volume is located. Support of this attribute indicates that the Provider can report this information back to the Consumer. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only
volume	*ref*	A reference to the Volume that will be connected. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolumeCollection",
  "id": string,
  "count": number,
  "machineVolumes": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineVolume",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "initialLocation": string, ?
      "volume": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/MachineVolumeCollection"
```

```
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <MachineVolume>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <initialLocation> xs:string </initialLocation> ?
    <volume href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </MachineVolume> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.1.1.3 MachineNetworkInterface Collection

The resource type for each item of this collection is “MachineNetworkInterface”, defined as follows:

Name	MachineNetworkInterface	
Type URI	http://schemas.dmtf.org/cimi/1/MachineNetworkInterface	
Attribute	**Type**	**Description**
addresses	*collection [MachineNetworkInterfaceAddress]*	A reference to the list of references to the Addresses for this network interface. Note: the MachineNetworkInterfaceAddress resource type is representing an association between the MachineNetworkInterface and an Address. It is defined following this resource's definition. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
network	*ref*	A reference to a Network for this network interface. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
networkPort	*ref*	A reference to the NetworkPort for this network interface. If this attribute is provided, the "network" attribute in the referenced NetworkPort shall have the same value as the "network" attribute in this networkInterface. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
state	*string*	The state of the MachineNetworkInterface. Allowable values include:

		ACTIVE: An active interface is the primary interface, able to forward traffic. **PASSIVE**: A passive interface is in a standby mode ready to forward traffic if the primary interface fails. **DISABLED: A disabled interface is one that is not able to forward traffic.** **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
macAddress	*string*	Address assigned by the hypervisor when a machine is created or a unique address can be manually assigned. While this attribute can be specified, in most cases it is expected to be supplied by the Provider. Specifying this value is typically only done when the Template is only used for one particular Machine. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
mtu	*integer*	To set the largest supported maximum transmission unit packet size. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write

JSON serialization:

```
{ "resourceURI":
    "http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceCollection",
  "id": string,
  "count": number,
  "machineNetworkInterfaces": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineNetworkInterface",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "addresses": { "href": string },
      "network": { "href": string },
      "networkPort": { "href": string }, ?
      "state": string, ?
      "macAddress": string, ?
      "mtu": number, ?
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
```

```
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
resourceURI="http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <MachineNetworkInterface>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <addresses href="xs:anyURI"/>
    <network href="xs:anyURI"/>
    <networkPort href="xs:anyURI"/> ?
    <state> xs:string </state> ?
    <macAddress> xs:string </macAddress> ?
    <mtu> xs:integer </mtu> ?
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </MachineNetworkInterface> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.1.1.4 MachineNetworkInterfaceAddress Collection

The resource type for each item of this collection is “MachineNetworkInterfaceAddress”, defined as follows:

Name	MachineNetworkInterfaceAddress	
Type URI	http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddress	
Attribute	**Type**	**Description**
address	*ref*	Reference to an Address resource. **Constraints:** **Provider:** support mandatory; mutable

		Consumer: support mandatory; read-only

JSON serialization:

```
{ "resourceURI":
"http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddressCollection",
  "id": string,
  "count": number,
  "machineNetworkInterfaceAddresses": [
    { "resourceURI":
        "http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddress",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "address": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
resourceURI="http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddressColle
ction"
   xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <MachineNetworkInterfaceAddress>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
```

```
    <address href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </MachineNetworkInterfaceAddress> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.1.1.5 MachineSnapshot Collection

The resource type for each item of this collection is “MachineSnapshot”, defined as follows:

Name	MachineSnapshot	
Type URI	http://schemas.dmtf.org/cimi/1/MachineSnapshot	
Attribute	**Type**	**Description**
snapshot	*ref*	Reference to a SNAPSHOT MachineImage resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineSnapshotCollection",
  "id": string,
  "count": number,
  "machineSnapshots": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineSnapshot",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "snapshot": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    }, +
  ] ?
  ...
}
```

XML serialization:

```
<Collection
resourceURI="http://schemas.dmtf.org/cimi/1/MachineSnapshotCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <MachineSnapshot>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <snapshot href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </MachineSnapshot> *
  <xs:any>*
</Collection>
```

Note: Previous versions of this specication included an "add" operation on this resource, it is now deprecated in favor of creating a new MachineImage with the imageLocation attribute pointing to the Machine to be snapshotted.

5.14.1.1.6 MachineMeter Collection

The resource type for each item of this collection is “Meter” as defined in clause 5.17.3.

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineMeterCollection",
  "id": string,
  "count": number,
  "meters": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
      "id": string,
      ... remaining Meter attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/MachineMeterCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Meter>
    <id> xs:anyURI </id>
    ... remaining Meter attributes ...
  </Meter> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.1.2 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Machine Collection resource.

The following custom operations are also defined:**start**

/link@rel: http://schemas.dmtf.org/cimi/1/action/startThis operation shall start a Machine.

Input parameters: None.

Output parameters: None.

During the processing of this operation, the Machine shall be in the “STARTING” state.

Upon successful completion of this operation, the Machine shall be in the "STARTED" state.

When a Machine is in the "STOPPED" state, starting it shall be the virtual equivalent of powering on a physical machine. There is no restored CPU or Memory state, so the guest OS will typically perform boot or installation tasks.

If the Machine was in the "SUSPENDED" or "PAUSED" state, starting it shall have the effect of resuming it.

HTTP protocol

To start a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/start" URI of the Machine where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/start",
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/start </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.**stop**

/link@rel: http://schemas.dmtf.org/cimi/1/action/stopThis operation shall stop a Machine.

Input parameters:

(3) "force" - type: boolean - optional
A flag to indicate whether the Provider shall simulate a power off condition (force=true) or shall simulate a shutdown operation that allows applications to save their state and the file system to be made consistent (force=false). Inclusion of this parameter by Consumers is optional and when not specified, the Provider may choose either mechanism. Providers are encouraged to advertise this choice via the MachineStopForceDefault capability.

Output parameters: None.

During the processing of this operation, the Machine shall be in the "STOPPING" state.

Upon successful completion of this operation, the Machine shall be in the "STOPPED" state. Stopping a Machine with force=true shall be the virtual equivalent of powering off a physical machine. There is no saved CPU or Memory state. Stopping a Machine with force=false shall result in a machine with consistent file systems.

A Consumer may reissue a stop operation when the state is STOPPING, perhaps with force=true, but Providers shall not issue a force=true stop operation on their own.

HTTP protocol

To stop a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the Machine where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/stop",
  "force": boolean, ?
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
```

```
  <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
  <force> xs:boolean </force> ?
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.**restart**

/link@rel: http://schemas.dmtf.org/cimi/1/action/restart

This operation shall restart a Machine. If the Machine is in the "STARTED" state, this operation shall have the effect of executing the "stop" and then "start" operations. If the Machine is in the "STOPPED" state, this operation shall have the effect of executing the "start" operation.

Input parameters:

(4) "force" - type: boolean - optional
A flag to indicate whether the Provider shall simulate a power off condition (force=true) or shall simulate a shutdown operation that allows applications to save their state and the file system to be made consistent (force=false). Inclusion of this parameter by Consumers is optional and when not specified, the Provider may choose either mechanism. Providers are encouraged to advertise this choice via the MachineStopForceDefault capability.

Output parameters: None.

During the processing of this operation, the Machine shall be in the “STOPPING” and/or “STARTING” states, as appropriate depending on its initial state.

Upon successful completion of this operation, the Machine shall be in the "STARTED" state. Restarting a Machine shall be the virtual equivalent of powering off, and then powering on a physical machine. There is no restored CPU or Memory state, so the guest OS will typically perform boot or installation tasks.

HTTP protocol

To restart a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/restart" URI of the Machine where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/restart",
  "force": boolean, ?
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/restart </action>
  <force> xs:boolean </force> ?
```

```
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.**pause**

/link@rel: http://schemas.dmtf.org/cimi/1/action/pause

This operation shall pause a Machine.

Input parameters: None.

Output parameters: None.

During the processing of this operation, the Machine shall be in the "PAUSING" state.

Upon successful completion of this operation, the Machine shall be in the "PAUSED" state. Pausing a Machine shall keep the Machine and its resources instantiated, but the Machine shall not be available to perform any tasks. The current state of the CPU and Memory shall be retained in volatile memory.

HTTP protocol

To pause a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action.pause" URI of the Machine where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/pause",
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/pause </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.**suspend**

/link@rel: http://schemas.dmtf.org/cimi/1/action/suspend

This operation shall suspend a Machine.

Input parameters: None.

Output parameters: None.

During the processing of this operation, the Machine shall be in the "SUSPENDING" state.

Upon successful completion of this operation, the Machine shall be in the "SUSPENDED" state. Suspending a Machine shall keep the Machine and its resources instantiated, but the Machine shall not be available to perform any tasks. The current state of the CPU and Memory shall be retained in non-volatile memory.

HTTP protocol

To suspend a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/suspend" URI of the Machine where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/suspend",
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/suspend </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.**capture**

/link@rel: http://schemas.dmtf.org/cimi/1/action/capture

This operation shall create a new Machine Image from an existing Machine. This operation is defined within the Machine Image resource; see 5.14.7.1 for more details. Note that while this operation is performed against a Machine Image, its presence in the Machine serialization is used to advertise support for the operation.

Snapshotting a Machine

/link@rel: http://schemas.dmtf.org/cimi/1/action/snapshot

This operation shall create a new SNAPSHOT Machine Image from an existing Machine. This operation is defined within the Machine Image resource; see 5.14.7.1 for more details. Note that while this operation is performed against a Machine Image, its presence in the Machine serialization is used to advertise support for the operation.

Restoring a Machine

/link@rel: http://schemas.dmtf.org/cimi/1/action/restore

This operation shall restore a Machine from a previously created Machine Image.

Input parameters:

(5) "image" - type: URI - mandatory
A reference to the Machine Image.

Output parameters: None.

During the processing of this operation, the Machine shall be in the "RESTORING" state.

Upon successful completion of this operation, the Machine shall be in the same state as the specified in the Machine Image, if specified. See 5.14.2.1 for more details.

Note that Providers can indicate support for restoring from non-SNAPSHOT Machine Images via the Machine "RestoreFromImage" capability. When this capability is not supported, but the restore operation is supported, then that indicates it only supports restoring from SNAPSHOT Machine Images.

HTTP protocol

To restore a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/restore" URI of the Machine where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/restore",
  "image": string,
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/restore </action>
  <image href="xs:anyURI"/>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Where the "image" URI is a reference to the Machine Image to be used.

Upon successful processing of the request, the HTTP response body may be empty.

5.14.2 Machine Collection

A Machine Collection resource represents the collection of Machine resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCollection",
  "id": string,
  "count": number,
  "machines": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
      "id": string,
      ... remaining Machine attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/MachineCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Machine>
    <id> xs:anyURI </id>
    ... remaining Machine attributes ...
  </Machine> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.2.1 Operations

NOTE: The "add" operation requires a MachineTemplate to be used (see 4.2.1.1).

Within the NetworkInterface portion of the MachineTemplate, there may be a reference to an Address resource. If one is not provided, the Provider shall create one on the Consumer's behalf. In these cases, and unless some action is taken to change this behavior, the Address will be bound to the new Machine that is created and shall be deleted by the Provider when the Machine is deleted. Additionally, if these Provider-created Address resources are disassociated from the Machine, the Provider shall delete them. If the Consumer does provide an Address resource, the Address shall not be deleted when the Machine is deleted and it is then up to the Consumer to delete the Address through some other mechanism.

Upon successful processing of the "add" operation, unless otherwise specified via the MachineTemplate "initialState" attribute, the state of the new Machine shall be the value of the DefaultInitialState capability, if defined. If no DefaultInitialState capability is defined the default value shall be "STOPPED." The semantics of "initialState"shall be equivalent to the Provider issuing the appropriate actions against the new Machine to move it into that state. Note that this controls the actions of the hypervisor and the state of the resources within the Machine (e.g. the operating system) will also be influenced by the data within the MachineImage used to create the new Machine. For example, if a new Machine's initialState is

"STARTED" and a SNAPSHOT MachineImage was used to create the new Machine then the Machine would not be "booted" but rather resume executing from the saved state in the MachineImage.

If a Provider is unable to change the state of the new Machine to the appropriate "initialState" (either as specified by the MachineTemplate or as implied by the previous stated rules), then the Machine creation shall fail.

If a Provider is unable to create the new Machine due to invalid or inconsistent credentials in the MachineTemplate then the Machine creation process shall fail. If any credentials are included in the MachineTemplate then they shall be part of the new Machine regardless of the type of MachineImage used.

5.14.3 Machine Template

A Machine Template represents the set of metadata and instructions used in the creation of a Machine.

Name	MachineTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/MachineTemplate	
Attribute	**Type**	**Description**
initialState	*string*	The initial state of the new Machine. Possible values include the non-transient states as specified by the Machine "state" attribute (eg. STARTED, STOPPED) and will be determined by the actions supported by the Provider. Providers should advertise the list of available values via the Machine's "initialStates" capability. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
machineConfig	*ref*	A reference to the Machine Configuration that will be used to create a Machine from this Machine Template. Note that the attributes of the MachineConfiguration may be specified rather than a reference to an existing MachineConfiguration resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
machineImage	*ref*	A reference to the Machine Image that will be used to create a Machine from this Machine Template. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
credential	*ref*	A reference to the Credential that will be used to create the initial login credentials for the new Machine. Note that the attributes of the Credential may be specified rather than a reference to an existing Credential resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
volumes	*volume[]*	A list of structures, each containing a reference to an existing Volume and potentially describing aspects of the way that the given Volume is to be connected to the Machine during its creation from this Machine Template. Each volume structure has the following attributes:

<table>
<tr>
<td></td>
<td></td>
<td>
<table>
<tr><th>Name</th><td colspan="2">volume</td></tr>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>initialLocation</td><td>string</td><td>An Operating System specific location(path) in its namespace where the Volume will appear.

Support of this attribute indicates that the Provider allows for Consumers to choose where the Volume will appear.

<u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>volume</td><td>ref</td><td>Reference to the Volume that will be connected.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
</table>
<u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write
</td>
</tr>
<tr>
<td>volumeTemplates</td>
<td>volumeTemplate[]</td>
<td>
A list of structures, each containing a reference to a Volume Template from which a Volume will be created and connected to the Machine resulting from this Machine Template. Each structure can potentially also include aspects of the way in which each created Volume will be connected to the created Machine.

If the Machine is created as part of a System creation, the Volumes created from these templates will be considered as part of that System without the need for these Volume Templates to also be listed in the volumeTemplates attribute of the relevant System Template. If the same Volume Template reference is listed in both the volumeTemplates attribute of a System Template and in the <code>volumeTemplates</code> attribute of a Machine Template contained by that System Template, this means that multiple, distinct Volume instances will be created as part of the overall System creation.Each volumeTemplate structure has the following attributes:
<table>
<tr><th>Name</th><td colspan="2">volumeTemplate</td></tr>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>initialLocation</td><td>string</td><td>An Operating System specific location(path) in its namespace where the Volume will appear.

Support of this attribute indicates that the Provider allows for Consumers to choose where the Volume will appear.

<u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>volumeTemplate</td><td>ref</td><td>Reference to the Volume Template that will be used to create a new Volume.

Note that the attributes of the VolumeTemplate may be specified rather than a reference to an existing VolumeTemplate resource.</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td><u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
</table>
<u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>networkInterfaces</td><td>networkInterface[]</td><td>
A list of structures, each containing references to the resources and attributes defining a network interface to be created on a Machine instantiated from this Machine Template. The resources referenced by each networkInterface structure are a Network, a NetworkPort and a list of Addresses:
<table>
<tr><th>Name</th><td colspan="2">networkInterface</td></tr>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>addresses</td><td>ref[]</td><td>A list of references to the Addresses for this network interface.

Array item name: address

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-only</td></tr>
<tr><td>network</td><td>ref</td><td>A reference to the Network for this network interface.

It is expected that NetworkPorts and Networks will be defined separately and prior to the Machines that connect to them.

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>networkPort</td><td>ref</td><td>A reference to the NetworkPort for this network interface.

Note this is a reference to a NetworkPort and not a NetworkPortTemplate. It is expected that NetworkPorts and Networks will be defined separately and prior to the Machines that connect to them.

If this attribute is provided, the "network" attribute in the referenced NetworkPort shall have the same value as the "network" attribute in this networkInterface.

<u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>state</td><td>string</td><td>The state of the network interface. Allowable values include:

ACTIVE: An active interface is the primary interface, able to forward traffic.

PASSIVE: A passive interface is in a standby mode ready to forward traffic if the primary interface fails.</td></tr>
</table>
</td></tr>
</table>

		<table><tr><td></td><td></td><td>**DISABLED: A disabled interface is one that is not able to forward traffic.** **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write</td></tr><tr><td>mtu</td><td>*integer*</td><td>To set the largest supported packet size. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write</td></tr></table> **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
userData	*string*	A Base64 encoded string whose decoded version is to be injected into Machines created by using this template. See the discussion of injection of user-defined data below. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
meterTemplates	*meterTemplates[]*	A list of references to Meter Templates that shall be used to create and connect a set of new Meters to the new Machine. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
eventLogTemplate	*ref*	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new Machine. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "initialState": string, ?
  "machineConfig": {
    "href": string | ... MachineConfiguration attributes ...
```

```
  }, ?
  "machineImage": {
    "href": string | ... MachineImage attributes ...
  }, ?
  "credential": {
    "href": string | ... CredentialTemplate attributes ...
  }, ?
  "volumes": [
    { "initialLocation": string?, "href": string }, +
  ], ?
  "volumeTemplates": [
    { "initialLocation": string?,
      "href": string, ?
      ... VolumeTemplate attributes ... ?
    }, +
  ], ?
  "networkInterfaces": [
    { "addresses": [
        {"href": string}, +
      ],
      "network": {"href": string},
      "networkPort": {"href": string}, ?
      "state": string,
      "mtu": number ?
    }, +
  ], ?
  "userData": string, ?
  "meterTemplates": [
    { "href": string, ?
      ... MeterTemplate attributes ... ?
    }, *
  ], ?
  "eventLogTemplate": {
    "href": string, ?
    ... EventLogTemplate attributes ... ?
  }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
```

```
  ...
}
```

XML media type: application/xml

XML serialization:

```
<MachineTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <initialState> xs:string </initialState> ?
  <machineConfig href="xs:anyURI"?>
    ... MachineConfiguration attributes ... ?
  </machineConfig> ?
  <machineImage href="xs:anyURI"?>
    ... MachineImage attributes ... ?
  </machineImage> ?
  <credential href="xs:anyURI"?>
    ... CredentialTemplate attributes ... ?
  </credential> ?
  <volume initialLocation="xs:string"? href="xs:anyURI" /> *
  <volumeTemplate initialLocation="xs:string"? href="xs:anyURI"? >
    ... VolumeTemplate attributes ... ?
  </volumeTemplate> *
  <networkInterface>
    <address href="xs:anyURI"/> *
    <network href="xs:anyURI"/>
    <networkPort href="xs:anyURI"/> ?
    <state> xs:string </state>
    <mtu> xs:integer </mtu> ?
  </networkInterface> *
  <userData> xs:string </userData> ?
  <meterTemplate href="xs:anyURI"? >
    ... MeterTemplate attributes ... ?
  </meterTemplate> *
  <eventLogTemplate href="xs:anyURI"? >
    ... EventLogTemplate attributes ... ?
  </eventLogTemplate> ?
```

```
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineTemplate>
```

Injection of user-defined data

To simplify the customization of individual Machines, it is possible to pass arbitrary data into the new Machine by using the userData parameter. The value of this parameter shall be the Base64-encoded payload. The Provider shall arrange for this data to be available from inside the Machine by using one of the following three methods:

1. *Metadata server*: The data can be retrieved from within the instance by using an HTTP GET request to http://169.254.169.254/cimi/latest/user-data.
2. *Disk*: The Machine will have access to a Disk with an ISO 9660 file system on it. The data can be found in a file at <*location*>/cimi/user-data.
3. *Image modification*: The Provider modifies the root file system of the machine image just before launching the machine. In UNIX-like operating systems, the data can be found in the file /var/lib/cimi/user-data.

It is strongly recommended that Providers implement a metadata server, or, failing that, injection via Disk, as image modification is brittle and may not work for every operating system in use. The Provider shall indicate which of these three methods is supported with the Machine 'UserData' capability in the ResourceMetadata for Machines. The value for this feature shall be one of metadata, disk, or imgmod, corresponding to the three methods listed above.

The Provider shall preserve this data across restarts of the machine. The data will be the Base64-decoded version of the data that was passed into the MachineCreate request.

5.14.3.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Machine Template Collection resource.

5.14.4 Machine Template Collection

A Machine Template Collection resource represents the collection of Machine Template resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplateCollection",
  "id": string,
  "count": number,
  "machineTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplate",
      "id": string,
      ... remaining MachineTemplate attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
```

```
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/MachineTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <MachineTemplate>
    <id> xs:anyURI </id>
    ... remaining MachineTemplate attributes ...
  </MachineTemplate> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.4.1 Operations

This resource supports the Read and Update operations. Creation of new Machine Template resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.14.5 Machine Configuration

The Machine Configuration resource represents the set of configuration values that define the (virtual) hardware resources of a to-be-realized Machine Instance. Machine Configurations are created by Providers and may, at the Providers discretion, be created by Consumers.

Name	MachineConfiguration	
Type URI	http://schemas.dmtf.org/cimi/1/MachineConfiguration	
Attribute	**Type**	**Description**
cpu	*integer*	Indicates the amount of CPU that a Machine realized from this configuration will have. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
memory	*integer*	Indicates the amount of RAM, in kibibytes, that a Machine realized from this configuration will have. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
disks	disk[]	. A list of structures, each containing the attributes defining the disks to be created for the Machine instantiated with this MachineConfiguration resource. The disks are local storage to the Machine. Each disks attribute has the following sub-attributes: **Name** | *disk*

<table>
<tr><td></td><td></td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>capacity</td><td>integer</td><td>Indicates the initial capacity, in kilobytes, of the disk described by this attribute. <u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>format</td><td>string</td><td>The format/type of this disk (e.g., ext4, NTFS).

<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-write</td></tr>
<tr><td>initialLocation</td><td>string</td><td>An Operating System specific location(path) in its namespace where this disk will first appear. Note, once deployed Consumers might move where this Disk is located.

<u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
</table>
<u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write
</td></tr>
<tr><td>cpuArch</td><td>string</td><td>This property indicates the CPU architecture that will be supported by Machines created by using this configuration.

Allowable values include: 68000, Alpha, ARM, Itanium, MIPS, PA_RISC, POWER, PowerPC, x86, x86_64, z/Architecture, SPARC. Providers may define additional values.

<u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
<tr><td>cpuSpeed</td><td>integer</td><td>The approximate CPU speed of this Machine - in megahertz.

<u>Constraints:</u>
Provider: support optional; mutable
Consumer: support optional; read-write</td></tr>
</table>

NOTE: The disk attributes "format" will not appear on Machine resources because after the Machine is created, the user of the Machine will be able to modify this attribute of a disk, possibly without the Provider's knowledge. Therefore these attributes might not be an aspect of the Machine that the Provider can reliably manage.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "cpu": number,
  "memory": number,
  "disks" : [
    { "capacity": number,
```

```
      "format": string,
      "initialLocation": string?
    }, +
  ], ?
  "cpuArch": string, ?
  "cpuSpeed": number, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<MachineConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <cpu> xs:integer </cpu>
  <memory> xs:integer </memory>
  <disk>
    <capacity> xs:integer </capacity>
    <format> xs:string </format>
    <initialLocation> xs:string </initialLocation> ?
  </disk> *
  <cpuArch> xs:string </cpuArch> ?
  <cpuSpeed> xs:integer </cpuSpeed> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineConfiguration>
```

5.14.5.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Machine Configuration Collection resource.

5.14.6 Machine Configuration Collection

A Machine Configuration Collection resource represents the collection of Machine Configuration resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI":
    "http://schemas.dmtf.org/cimi/1/MachineConfigurationCollection",
  "id": string,
  "count": number,
  "machineConfigurations": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
      "id": string,
      ... remaining MachineConfiguration attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/MachineConfigurationCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <MachineConfiguration>
    <id> xs:anyURI </id>
    ... remaining MachineConfiguration attributes ...
  </MachineConfiguration> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.6.1 Operations

This resource supports the Read and Update operations. Creation of new Machine Configuration resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.14.7 Machine Image

This resource represents the information necessary for hardware virtualized resources to create a Machine Instance; it contains configuration data such as startup instructions, including possible combinations of the following items, depending on the 'type' of Machine Image created:

- the software image (i.e., a copy of an installed Machine), which is to be instantiated on the disk and other virtual resources. The image can be a snapshot that consists of disk images plus memory and other resource state information.
- installation software, which, when executed on the hardware (virtual) resources, builds the machine instance
- both a disk image and a set of software and parameters in order to install new components not included in the original disk image

Name	MachineImage	
Type URI	http://schemas.dmtf.org/cimi/1/MachineImage	
Attribute	**Type**	**Description**
state	*string*	The operational state of the MachineImage. Allowable values include: **CREATING**: The MachineImage is in the process of being created. **AVAILABLE**: The MachineImage is available and ready for use. **Unless otherwise specified, the MachineImage shall initially be in this state after successful creation.** **DELETING**: The MachineImage is in the process of being deleted. **ERROR**: The Provider has detected an error in the MachineImage. **The operations that result in transitions to the above defined states are defined in Section 5.14.7.1** **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
type	*string*	The type of Machine Image that is represented by this resource. This specification defines the following values: **IMAGE**: This type represents the persisted data of a stopped Machine. Unlike "snapshots", it does not contain any runtime information. When this value is used the "relatedImage" attribute shall not be present. **SNAPSHOT**: This type represents the persisted data of a Machine. If the Machine was not in a stopped state when this Image was created, it will also contain runtime information. When this value is used, the "relatedImage" attribute shall reference the most recently created (or reverted to) snapshot Image for that Machine, which allows for easy discovery of the "previous" snapshot. The "relatedImage" attribute shall not be set by Consumers. **PARTIAL_SNAPSHOT**: This type follows the same semantics as the "SNAPSHOT" Machine Image except that it will contain just the changes (deltas) made to the Machine based on the referenced "relatedImage" Machine Image rather than a complete representation of the Machine. When a Machine Image is deleted, the following semantics shall apply: Any "SNAPSHOT" Machine Images that have a "relatedImage" value that references the deleted Machine Image shall have that value changed to the "relatedImage" attribute of the delete Machine Image. Any "PARTIAL_SNAPSHOT" Machine Images that have a "relatedImage" value that references the deleted Machine Image shall also be deleted. This detail applies recursively to any subsequent "PARTIAL_SNAPSHOT" Machine Images as well. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only

imageLocation	*URI*	A reference to the location of the binary data that makes up this image. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
relatedImage	*ref*	A reference to another Machine Image resource that is related to this one. The specific meaning of this value will vary depending on the type of Machine Image. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only

3630 The following describes the serialization of the resource in both JSON and XML:

3631 **JSON media type:** application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "type": string,
  "imageLocation": string,
  "relatedImage": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<MachineImage xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <type> xs:string </type>
  <imageLocation> xs:anyURI </imageLocation>
  <relatedImage href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineImage>
```

5.14.7.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Machine Image Collection resource.

When creating a new Machine Image the representation of the new Machine Image may include a reference in the "imageLocation" attribute. Providers shall inspect this reference (most likely via an HTTP HEAD) to determine if any special processing is required. This specification defines the following additional steps that Providers shall take depending on the type of resource being referenced:

http://schemas.dmtf.org/cimi/1/Machine

If the "imageLocation" is a reference to a Machine, the Provider shall create a new SNAPSHOT Machine Image based on the Machine being referenced. Upon completion of the create operation, the Machine Image's "imageLocation" attribute shall not reference the Machine (as the Machine might change over time), but instead it shall reference the (or contain the data of a) static representation of the Machine. Additionally, the referenced Machine's MachineSnapshot Collection shall be updated to include a reference to this newly created SNAPSHOT MachineImage resource.

5.14.8 Machine Image Collection

A Machine Image Collection resource represents the collection of Machine Image resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImageCollection",
  "id": string,
  "count": number,
  "machineImages": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
      "id": string,
      ... remaining MachineImage attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/MachineImageCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <MachineImage>
    <id> xs:anyURI </id>
    ... remaining MachineImage attributes ...
  </MachineImage> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.8.1 Operations

This resource supports the Read and Update operations. Creation of new Machine Image resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1, where the request body and the way it is processed is described in clause 5.14.7.1.

5.14.9 Credential

A Credential resource contains the information required to create the initial administrative superuser of a newly created Machine or to represent the credentials needed to perform some operation. Due to the variation between operating systems and Providers, this specification does not mandate one particular set of attributes that all implementations need to support. However, Providers are expected to extend this resource with additional attributes to meet their requirements.

For example, a Provider might extend this resource with username and password attributes, which would then be the login information for new Machines. These extension attributes would appear as siblings to the common attributes like "name"' and "description."

Name	Credential	
Type URI	http://schemas.dmtf.org/cimi/1/Credential	
Attribute	**Type**	**Description**
TBD		The exact set of attributes will be determined by the Provider.

Some common extension attributes that Providers might use include:

UserName/Password:

Attribute	**Type**	**Description**
userName	*string*	The initial superuser's user name. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
password	*string*	Initial superuser's password. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; write-only

Public key:

Attribute	**Type**	**Description**
key	*byte[]*	The digit of the public key for the initial superuser. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Credential",
  "id": string,
  "name": string, ?
```

```
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Credential xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</Credential>
```

5.14.9.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Credential Collection resource.

5.14.10 Credential Collection

A Credential Collection resource represents the collection of Credential resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialCollection",
  "id": string,
  "count": number,
  "credential": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Credential",
      "id": string,
      ... remaining Credential attributes ...
    }, +
```

```
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/CredentialCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Credential>
    <id> xs:anyURI </id>
    ... remaining Credential attributes ...
  </Credentials> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.10.1 Operations

NOTE: The "add" operation requires a CredentialTemplate to be used (see 4.2.1.1).

5.14.11 Credential Template

This resource captures the configuration values for realizing a Credential resource. A Credential Template may be used to create multiple Credentials.

Name	CredentialTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/CredentialTemplate	
Attribute	**Type**	**Description**
TBD		The exact set of attributes will be determined by the provider.

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
```

```
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<CredentialTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</CredentialTemplate>
```

5.14.11.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Credential Template Collection resource.

5.14.12 Credential Template Collection

A Credential Template Collection resource represents the collection of CredentialTemplate resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI":
    "http://schemas.dmtf.org/cimi/1/CredentialTemplateCollection",
  "id": string,
  "count": number,
  "credentialTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialTemplate",
      "id": string,
      ... remaining CredentialTemplate attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/CredentialTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <CredentialTemplate>
    <id> xs:anyURI </id>
    ... remaining CredentialTemplate attributes ...
  </CredentialTemplate> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.12.1 Operations

This resource supports the Read and Update operations. Creation of new Credential Template resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.15 Volume resources and relationships

Figure 4 illustrates the resources involved in constructing a Volume and their relationships. Although this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor normative.

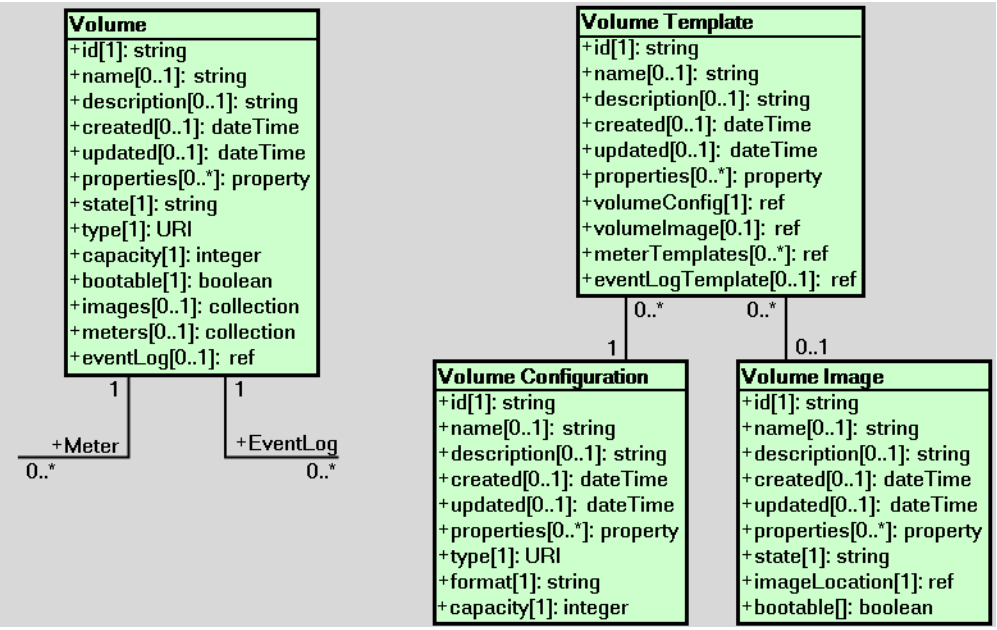


Figure 4 - Volume resources

5.15.1 Volume

A Volume represents storage at either the block or the file-system level. Volumes can be connected to Machines. Once connected, Volumes can be accessed by processes on that Machine.

Name	Volume	
Type URI	http://schemas.dmtf.org/cimi/1/Volume	
Attribute	**Type**	**Description**
state	*string*	Indicates the operational state of the Volume. Allowable values include: **CREATING**: The Volume is in the process of being created. **AVAILABLE**: The Volume is available and ready for use. Unless otherwise specified, the Volume shall initially be in this state after successful creation. **CAPTURING**: The Volume is in the process of being captured (snapshotted) into a new VolumeImage. Allowable action when in this state is: **delete**. **DELETING**: The Volume is in the process of being deleted. **ERROR**: The Provider has detected an error in the Volume. **The operations that result in transitions to the above defined states are defined in Section 5.15.1.2** **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
type	*URI*	A URI that indicates the type of Volume to be created. This specification defines the following URI: **http://schemas.dmtf.org/cimi/1/mapped**: Indicates a Volume that shall be used for shared storage that might be available to multiple Machines, but which does not require an explicit mount operation from within the guest operating system. Additional values may be defined. If certain types of Volumes require additional data then it is expected that this resource will be extended. For example, a "sharedFileSystem" type might require additional networking information and credentials to be specified. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
capacity	*integer*	The maximum size, when limited, of the Volume in kilobytes. When this value is increased, the Volume can contain more data. Decreasing this value may require evaluations. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
bootable	*boolean*	This property indicates whether this Volume is bootable. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
images	*collection [VolumeVolumeImage]*	A reference to the list of references to Volume Images that represent snapshots taken from the Volume. Note: the VolumeVolumeImage resource type is representing an association between the Volume and a VolumeImage. It is defined in the following clause. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only

meters	*collection [Meter]*	A reference to the list of Meters monitored for this Volume. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
eventLog	*ref*	A reference to the EventLog of this Volume. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Volume",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "type": string,
  "capacity": number,
  "bootable": boolean,
  "images": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Volume xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
```

```
  <state> xs:string </state>
  <type> xs:anyURI </type>
  <capacity> xs:integer </capacity>
  <bootable> xs:boolean </bootable>
  <images href="xs:anyURI"/> ?
  <meters href="xs:anyURI"/> ?
  <eventLog href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</Volume>
```

5.15.1.1 Collections

The following describes the collection resources owned by Volumes.

5.15.1.1.1 VolumeVolumeImage Collection

The resource type for each item of this collection is “VolumeVolumeImage”, defined as follows:

Name	VolumeVolumeImage	
Type URI	http://schemas.dmtf.org/cimi/1/VolumeVolumeImage	
Attribute	**Type**	**Description**
volumeImage	*ref*	Reference to a Volume Image resource, which represents a snapshot of this Volume. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeVolumeImageCollection",
  "id": string,
  "count": number,
  "volumeVolumeImages": [
    { "resourceURI":
        "http://schemas.dmtf.org/cimi/1/VolumeVolumeImage",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "volumeImage": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
```

```
      ] ?
      ...
    }, +
  ] ?
  ...
}
```

XML serialization:

```
<Collection
resourceURI="http://schemas.dmtf.org/cimi/1/VolumeVolumeImageCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <VolumeVolumeImage>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <volumeImage href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </VolumeVolumeImage> *
  <xs:any>*
</Collection>
```

Note: Previous versions of this specification included an "add" operation on this resource, it is now deprecated in favor of creating a new VolumeImage with the imageLoation attribute pointing to the Volume to be captured.

5.15.1.1.2 VolumeMeter Collection

The resource type for each item of this collection is “Meter” as defined in clause 5.17.3.

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeMeterCollection",
  "id": string,
  "count": number,
  "meters": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
      "id": string,
      ... remaining Meter attributes ...
    }, +
```

```
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/VolumeMeterCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Meter>
    <id> xs:anyURI </id>
    ... remaining Meter attributes ...
  </Meter> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.15.1.2 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Volume Collection resource.

5.15.2 Volume Collection

A Volume Collection resource represents the collection of Volumes within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeCollection",
  "id": string,
  "count": number,
  "volumes": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Volume",
      "id": string,
      ... remaining Volume attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/VolumeCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
```

```
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Volume>
    <id> xs:anyURI </id>
    ... remaining Volume attributes ...
  </Volume> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.15.2.1 Operations

NOTE: The "add" operation requires a VolumeTemplate to be used (see 4.2.1.1).

5.15.3 Volume Template

This resource captures the configuration values for realizing a Volume. A Volume Template may be used to create multiple Volumes.

Name	VolumeTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/VolumeTemplate	
Attribute	**Type**	**Description**
volumeConfig	*ref*	A reference to the Volume Configuration that will be used to create a Volume from this Volume Template. Note that the attributes of the VolumeConfiguration may be specified rather than a reference to an existing VolumeConfiguration resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
volumeImage	*ref*	A reference to the Volume Image that will be used to create a Volume from this Volume Template. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
meterTemplates	*meterTemplates[]*	A list of references to Meter Templates that shall be used to create and connect a set of new Meters to the new Volume. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
eventLogTemplate	*ref*	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new Volume. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "volumeConfig": {
    "href": string | ... VolumeConfiguration attributes ...
  },
  "volumeImage": { "href": string }, ?
  "meterTemplates": [
    { "href": string, ?
      ... MeterTemplate attributes ... ?
    }, *
  ], ?
  "eventLogTemplate": {
    "href": string, ?
    ... EventLogTemplate attributes ... ?
  }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<VolumeTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <volumeConfig href="xs:anyURI"?>
```

```
    ... VolumeConfiguration attributes ... ?
  </volumeConfig>
  <volumeImage href="xs:anyURI"/> ?
  <meterTemplate href="xs:anyURI"? >
    ... MeterTemplate attributes ... ?
  </meterTemplate> *
  <eventLogTemplate href="xs:anyURI"? >
    ... EventLogTemplate attributes ... ?
  </eventLogTemplate> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</VolumeTemplate>
```

5.15.3.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Volume Template Collection resource.

5.15.4 Volume Template Collection

A Volume Template Collection resource represents the collection of VolumeTemplate resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeTemplateCollection",
  "id": string,
  "count": number,
  "volumeTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeTemplate",
      "id": string,
      ... remaining volumeTemplate attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/VolumeTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <VolumeTemplate>
    <id> xs:anyURI </id>
    ... remaining VolumeTemplates attributes ...
  </VolumeTemplate> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.15.4.1 Operations

This resource supports the Read and Update operations. Creation of new Volume Template resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.15.5 Volume Configuration

The Volume Configuration resource represents the set of configuration values needed to create a Volume with certain characteristics. Volume Configurations are created by Providers and may, at the Providers discretion, be created by Consumers.

Name	VolumeConfiguration	
Type URI	http://schemas.dmtf.org/cimi/1/VolumeConfiguration	
Attribute	**Type**	**Description**
type	*URI*	A URI that indicates the type of Volume to be created. This specification defines the following URI: **http://schemas.dmtf.org/cimi/1/mapped**:Indicates a Volume that shall be used for shared

		storage that might be available to multiple Machines, but which does not require an explicit mount operation from within the guest operating system. Additional values may be defined. If certain types of Volumes require additional data then it is expected that this resource will be extended. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
format	*string*	The format of the file system that will be placed on Volumes created from this configuration. This attribute is only meaningful for Volume Configurations that describe block devices. This attribute is optional; the absence of this attribute indicates that Volumes created from this configuration will not be formatted with a file system. Example values: "ext4," "ntfs." **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
capacity	*integer*	The default size in kilobytes, when limited, of the Volume created from this Volume Configuration. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "type": string,
  "format": string,
  "capacity": number,
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<VolumeConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
```

```
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <type> xs:anyURI </type>
  <format> xs:string </format>
  <capacity> xs:integer </capacity>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</VolumeConfiguration>
```

5.15.5.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Volume Configuration Collection resource.

5.15.6 Volume Configuration Collection

A Volume Configuration Collection resource represents the collection of Volume Configuration resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI":
    "http://schemas.dmtf.org/cimi/1/VolumeConfigurationCollection",
  "id": string,
  "count": number,
  "volumeConfigurations": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
      "id": string,
      ... remaining VolumeConfiguration attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/VolumeConfigurationCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <VolumeConfiguration>
```

```
    <id> xs:anyURI </id>
    ... remaining VolumeConfiguration attributes ...
  </VolumeConfiguration> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.15.6.1 Operations

This resource supports the Read and Update operations. Creation of new Volume Image resources are supported via a POST to the "add" operations' URI as described in clause 4.2.1.1.

5.15.7 Volume Image

This resource represents an image that could be placed on a pre-loaded volume.

Name	VolumeImage	
Type URI	http://schemas.dmtf.org/cimi/1/VolumeImage	
Attribute	**Type**	**Description**
state	*string*	Indicates the operational state of the VolumeImage. Allowable values include: **CREATING**: The VolumeImage is in the process of being created. **AVAILABLE**: The VolumeImage is available and ready for use. Unless otherwise specified, the VolumeImage shall initially be in this state after successful creation. **DELETING**: The VolumeImage is in the process of being deleted. **ERROR**: The Provider has detected an error in the VolumeImage. **The operations that result in transitions to the above defined states are defined in Section 5.15.7.1** **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
imageLocation	*ref*	A reference to the location of the binary data that makes up this image. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
bootable	*boolean*	This property indicates whether Volumes created from this Volume Image will be bootable. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeImage",
  "id": string,
  "name": string, ?
  "description": string, ?
```

```
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "imageLocation": { "href": string },
  "bootable": boolean,
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<VolumeImage xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <imageLocation href="xs:anyURI"/>
  <bootable> xs:boolean </bootable>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</VolumeImage>
```

5.15.7.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Volume Image Collection resource.

5.15.8 Volume Image Collection

A Volume Image Collection resource represents the collection of Volume Image resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeImageCollection",
  "id": string,
  "count": number,
```

```
  "volumeImages": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeImage",
      "id": string,
      ... remaining VolumeImage attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/VolumeImageCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <VolumeImage>
    <id> xs:anyURI </id>
    ... remaining VolumeImage attributes ...
  </VolumeImage> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

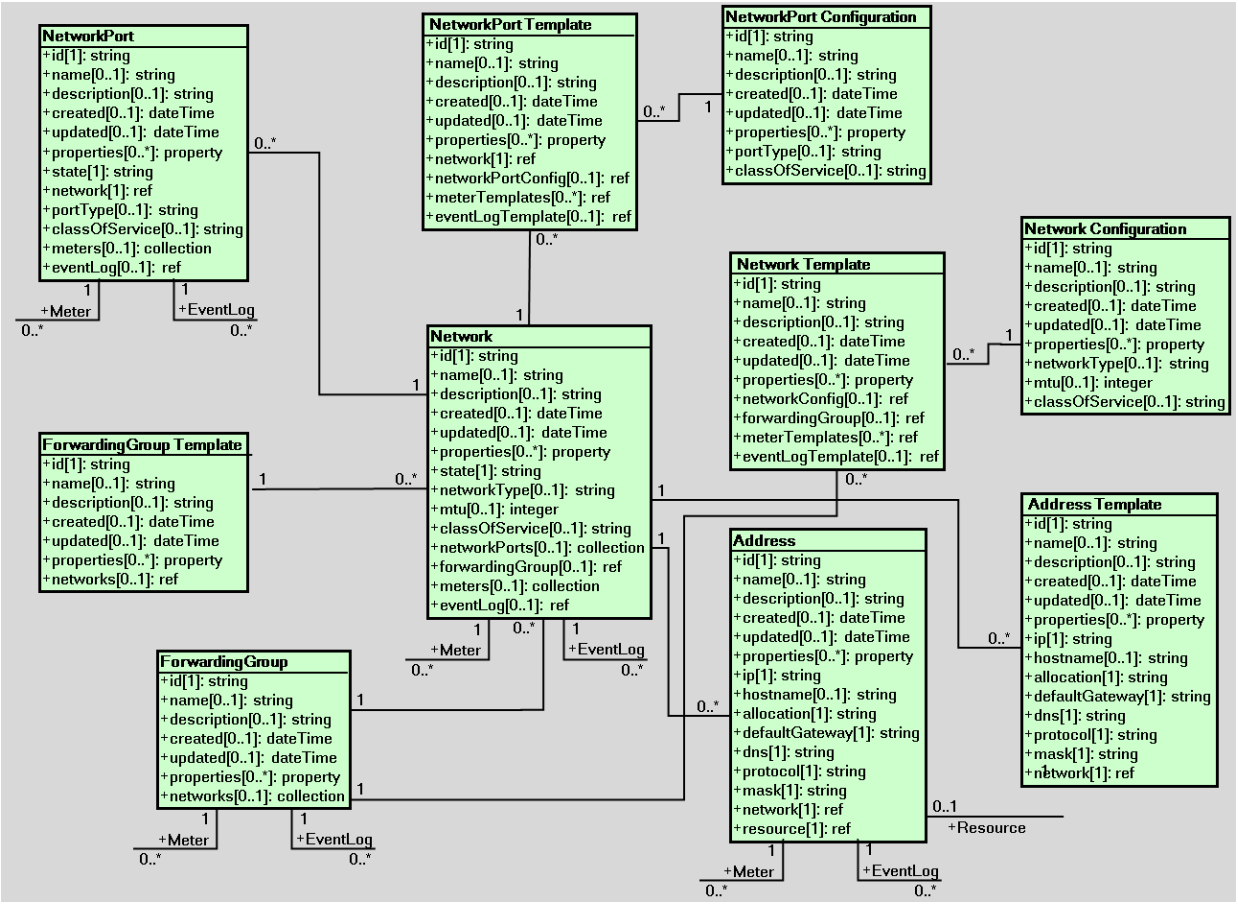
5.15.8.1 Operations

This resource supports the Read and Update operations. Creation of new Volume Image resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

During the creation of a new Volume Image resource, if the "imageLocation" attribute refers to an existing Volume, this shall be interpreted as a request to create a snapshot of the Volume. Once completed, the "imageLocation" attribute of the new Volume Image resource shall not refer to the original Volume, instead it shall refer to a static copy of the Volume. Additionally, the referenced Volume's VolumeVolumeImage Collection shall be updated to include a reference to this newly created snapshot Volume Image resource. During this process, the Provider may put the Volume into a "CAPTURING" state if necessary.

5.16 Network resources and relationships

Figure 5 illustrates the resources involved in constructing Networks and their Network Ports and their relationships. Although this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor normative.



4268 **Figure 5 - Network resources**

4269 ### 5.16.1 Network

4270 A network is a collection of interconnected logical services with the purpose of forwarding data traffic
4271 between end points.

4272 Networks in a ForwardingGroup should all have the same "networkType" attributes, which prevents a
4273 Network with a "private" access attribute from being publicly forwarded because it is a member of a
4274 ForwardingGroup that also contains Networks with a "public" access attribute.

Name	Network	
Type URI	http://schemas.dmtf.org/cimi/1/Network	
Attribute	**Type**	**Description**
state	*string*	The operational state of the Network. Allowable values include: **CREATING**: The Network is in the process of being created. **STARTING**: The Network is in the process of being started. **STARTED**: The Network is available and ready for use. **STOPPING**: The Network is in the process of being stopped. **STOPPED**: The Network is stopped and not available for use.

		DELETING: The Network is in the process of being deleted. **ERROR**: The Provider has detected an error in the Network.The operations that result in transitions to the above defined states are defined in Section **5.16.1.2**. Section **5.16.2.1** defines the initial state of a Network. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
networkType	*string*	An indicator of whether the Machine resource has access to a Public or Private network. Allowable values include: **PUBLIC**: represents an open and Internet routable network. **PRIVATE**: identifies a local non-routed network. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support optional; read-write
mtu	*integer*	Maximum Transmission Unit. Indicates The largest Packet size supported on this network. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
classOfService	*string*	Indicates the Provider's supported category, associated with a collection of attributes characterizing a level of a quality experience Example values: **GOLD**: High bandwidth, low latency, low jitter **SILVER**: An improved service experience over bronze for voice or video traffic **BRONZE**: Best effort The list of possible values, and their implied quality of service, is out of scope of this specification. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
networkPorts	*collection [Network Network Port]*	A reference to the list of NetworkPorts that are associated with this Network. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
forwardingGroup	*ref*	A reference to a ForwardingGroup of which this Network is a part. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
meters	*collection [Meter]*	A reference to the list of Meters monitored for this Network. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only
eventLog	*ref*	A reference to the EventLog of this Network. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Network",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "networkType": string, ?
  "mtu": number, ?
  "classOfService": string, ?
  "networkPorts": { "href": string }, ?
  "forwardingGroup": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Network xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <networkType> xs:string </networkType> ?
  <mtu> xs:integer </mtu> ?
  <classOfService> xs:string </classOfService> ?
```

```
  <networkPorts href="xs:anyURI"/> ?
  <forwardingGroup href="xs:anyURI"/> ?
  <meters href="xs:anyURI"/> ?
  <eventLog" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
href="xs:anyURI"/> ?
  <xs:any>*
</Network>
```

5.16.1.1 Collections

The following describes the collection resources owned by Networks.

5.16.1.1.1 NetworkNetworkPort Collection

When NetworkPorts are created via a Network's NetworkPortCollection's "add" operation, they shall added to the global (Cloud Entry Point) NetworkPortCollection as well.

As specified in clause 5.5.12, when a Network is deleted all of its collections, and resources in those collections, shall also be deleted. This means that all of the NetworkPorts related to that Network shall also be deleted.

The resource type for each item of this collection is “NetworkPort” as defined in clause 5.16.7.

JSON serialization:

```
{ "resourceURI":
    "http://schemas.dmtf.org/cimi/1/NetworkNetworkPortCollection",
  "id": string,
  "count": number,
  "networkNetworkPorts": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkNetworkPort",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "networkPort": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
```

```
      ...
    }, +
  ] ?
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/NetworkNetworkPortCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <NetworkNetworkPort>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <networkPort href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </NetworkNetworkPort> *
  <xs:any>*
</Collection>
```

5.16.1.1.2 NetworkMeter Collection

The resource type for each item of this collection is “Meter” as defined in clause 5.17.3.

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkMeterCollection",
  "id": string,
  "count": number,
  "meters": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
      "id": string,
      ... remaining Meter attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
```

```
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/NetworkMeterCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Meter>
    <id> xs:anyURI </id>
    ... remaining Meter attributes ...
  </Meter> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.1.2 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Network Collection resource.

The following custom operations are also defined:**start**

/link@rel: http://schemas.dmtf.org/cimi/1/action/start

This operation shall start a Network.

Input parameters: None.

Output parameters: None.

During the processing of this operation, the Network shall be in the "STARTING" state.

Upon successful completion of this operation, the Network shall be in the "STARTED" state.

HTTP protocol

To start a Network, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/start" URI of the Network where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/start",
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/start </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.**stop**

/link@rel: http://schemas.dmtf.org/cimi/1/action/stop

This operation shall stop a Network. When stopped, a Network shall not allow data to flow through it.

Input parameters: None.

Output parameters: None.

During the processing of this operation, the Network shall be in the "STOPPING" state.

Upon successful completion of this operation, the Network shall be in the "STOPPED" state.

HTTP Protocol

To stop a Network, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the Network where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/stop",
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.

5.16.2 Network Collection

A Network Collection resource represents the collection of Networks within a Provider and follows the Collection pattern that is defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkCollection",
  "id": string,
  "count": number,
  "networks": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Network",
      "id": string,
      ... remaining Network attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/NetworkCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Network>
    <id> xs:anyURI </id>
    ... remaining Network attributes ...
  </Network> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.2.1 Operations

NOTE: The "add" operation requires a NetworkTemplate to be used (see 4.2.1.1).

Upon successful processing of the "add" operation, unless otherwise specified via the NetworkTemplate "initialState" attribute, the state of the new Network shall be the value of the DefaultInitialState capability of the Network resource's ResourceMetadata, if defined. If no DefaultInitialState capability is defined the default value shall be "STOPPED." The semantics of "initialState"shall be equivalent to the Provider issuing the appropriate actions against the new Network to move it into that state.

If a Provider is unable to change the state of the new Network to the appropriate "initialState" (either as specified by the NetworkTemplate or as implied by the previous stated rules), then the Network creation shall fail.

5.16.3 Network Template

The Network Template is a set of configuration values for realizing a Network. An instance of Network Template may be used to create multiple Networks.

Name	NetworkTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkTemplate	
Attribute	**Type**	**Description**
initialState	*string*	The initial state of the new Network. Possible values include the non-transient states as specified by the Network "state" attribute (i.e., STARTED, STOPPED) and shall be determined by the actions supported by the Provider. Providers should advertise the list of available values via the Network ResourceMetadata "initialStates" capability. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
networkConfig	*ref*	A reference to the Network Configuration that will be used to create a Network from this Network Template. Note that the attributes of the NetworkConfiguration may be specified rather than a reference to an existing NetworkConfiguration resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
forwardingGroup	*ref*	A reference to a ForwardingGroup of which this Network will be a part. Note that Networks forward to themselves; therefore, this attribute will only appear in cases where the Network that will be created from this template forwards to one or more additional Networks. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
meterTemplates	*meterTemplates[]*	A list of references to Meter Templates that shall be used to create and connect a set of new Meters to the new Network. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
eventLogTemplate	*ref*	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new Network. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
```

```
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "initialState": string, ?
  "networkConfig": {
    "href": string |... NetworkingConfiguration attributes ...
  }, ?
  "forwardingGroup": { "href": string }, ?
  "meterTemplates": [
    { "href": string, ?
      ... MeterTemplate attributes ... ?
    }, *
  ], ?
  "eventLogTemplate": {
    "href": string, ?
    ... EventLogTemplate attributes ... ?
  }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<NetworkTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <initialState> xs:string </initialState> ?
  <networkConfig href="xs:anyURI"?>
    ... NetworkConfiguration attributes ... ?
  </networkConfig> ?
  <forwardingGroup href="xs:anyURI"/> ?
  <meterTemplate href="xs:anyURI"? >
    ... MeterTemplate attributes ... ?
```

```
  </meterTemplate> *
  <eventLogTemplate href="xs:anyURI"? >
    ... EventLogTemplate attributes ... ?
  </eventLogTemplate> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</NetworkTemplate>
```

5.16.3.1 Operations

This resource supports the Read, Update and Delete operations. Create is supported via the Network Template Collection resource.

5.16.4 Network Template Collection

A Network Template Collection resource represents the collection of NetworkTemplates within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkTemplateCollection",
  "id": string,
  "count": number,
  "networkTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkTemplate",
      "id": string,
      ... remaining NetworkTemplate attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/NetworkTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <NetworkTemplate>
    <id> xs:anyURI </id>
    ... remaining NetworkTemplate attributes ...
  </NetworkTemplate> *
  <operation rel="add" href="xs:anyURI"/> ?
```

```
  <xs:any>*
</Collection>
```

5.16.4.1 Operations

This resource supports the Read and Update operations. Creation of new Network Template resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.16.5 Network Configuration

The following set of configuration values represent the information needed to create a Network with certain characteristics.

Name	NetworkConfiguration	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkConfiguration	
Attribute	**Type**	**Description**
networkType	*string*	An indicator of whether or not the Network will be a Public or Private network. Allowable values include: **PUBLIC**: represents an open and Internet routable network. **PRIVATE**: identifies a local non-Internet network. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
mtu	*integer*	Maximum Transmission Unit. Size Indicates the largest supported packet size. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
classOfService	*string*	Indicates the Provider's supported category, associated with a collection of attributes characterizing a level of a quality experience Example values: **GOLD**: High bandwidth, low latency, low jitter **SILVER**: An improved service experience over bronze for voice or video traffic **BRONZE**: Best effort The list of possible values, and their implied quality of service, is out of scope of this specifications. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkConfiguration",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
```

```
  "updated": string, ?
  "properties": { string: string, + }, ?
  "networkType": string, ?
  "mtu": number, ?
  "classOfService": string, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<NetworkConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <networkType> xs:string </networkType> ?
  <mtu> xs:integer <mtu> ?
  <classOfService> xs:string </classOfService> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</NetworkConfiguration>
```

5.16.5.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Network Configuration Collection resource.

5.16.6 Network Configuration Collection

A Network Configuration Collection resource represents the collection of Network Configurations within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI":
    "http://schemas.dmtf.org/cimi/1/NetworkConfigurationCollection",
  "id": string,
  "count": number,
```

```
  "networkConfigurations": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkConfiguration",
      "id": string,
      ... remaining NetworkConfiguration attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/NetworkConfigurationCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <NetworkConfiguration>
    <id> xs:anyURI </id>
    ... remaining NetworkConfiguration attributes ...
  </NetworkConfiguration> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.6.1 Operations

This resource supports the Read and Update operations. Creation of new Network Configuration resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.16.7 Network Port

A NetworkPort is a realized connection point between a Network and a resource - such as a Machine.

Name	NetworkPort	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkPort	
Attribute	**Type**	**Description**
state	*string*	The operational state of the NetworkPort. Allowable values include: **CREATING**: The NetworkPort is in the process of being created. **STARTED**: The NetworkPort is available (enabled) and ready for use. **STOPPED**: The NetworkPort is stopped(disabled) and not available for use. **DELETING**: The NetworkPort is in the process of being deleted. **ERROR**: The Provider has detected an error in the NetworkPort.The operations that result in transitions to the above defined states are defined in Section **5.16.7.2.** Section **5.16.8.1 defines the initial state of a NetworkPort.** **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
network	*ref*	A reference to the network associated with this NetworkPort. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
portType	*string*	Indicates that a port will be used as an Access port (a member of the network) or a Trunk port that becomes a transport for multiple networks. Allowable values include: **ACCESS**: a member of a network. **TRUNK**: transport more than one network. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
classOfService	*string*	Indicates the Provider supported category, associated with a collection of attributes characterizing a level of a quality experience Example values: **GOLD**: High bandwidth, low latency, low jitter **SILVER**: An improved service experience over bronze for voice or video traffic **BRONZE**: Best effort The list of possible values, and their implied quality of service, is out of scope of this specifications. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
meters	*collection [Meter]*	A reference to the list of Meters monitored for this NetworkPort.

		Constraints: **Provider:** support optional; mutable **Consumer:** support optional; read-only
eventLog	*ref*	A reference to the EventLog of this NetworkPort. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-only

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPort",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "network": { "href": string },
  "portType": string, ?
  "classOfService": string, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<NetworkPort xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
```

```
  <state> xs:string </state>
  <network href="xs:anyURI"/>
  <portType> xs:string </portType> ?
  <classOfService> xs:string </classOfService> ?
  <meters href="xs:anyURI"/> ?
  <eventLog" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
href="xs:anyURI"/> ?
  <xs:any>*
</NetworkPort>
```

5.16.7.1 Collections

The following describes the collection resources owned by NetworkPorts.

5.16.7.1.1 NetworkPortMeter Collection

The resource type for each item of this collection is “Meter” as defined in clause 5.17.3.

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortMeterCollection",
  "id": string,
  "count": number,
  "meters": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
      "id": string,
      ... remaining Meter attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/NetworkPortMeterCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Meter>
    <id> xs:anyURI </id>
```

```
    ... remaining Meter attributes ...
  </Meter> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.7.2 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Network Port Collection resource.

Deleting a NetworkPort shall remove that NetworkPort from the global (Cloud Entry Point) NetworkPort Collection as well as from its corresponding Network's NetworkPorts collection.

The following custom operations are also defined:**start**

/link@rel: http://schemas.dmtf.org/cimi/1/action/start

This operation shall start a NetworkPort.

Input parameters: None.

Output parameters: None.

Upon successful completion of this operation, the NetworkPort shall be in the "STARTED" state.

HTTP Protocol

To start a NetworkPort, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/start" URI of the NetworkPort where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/start",
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/start </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.**stop**

/link@rel: http://schemas.dmtf.org/cimi/1/action/stop

This operation shall stop a NetworkPort. When stopped, the NetworkPort shall not be available for use and no network traffic shall flow through it.

Input parameters: None.

Output parameters: None.

Upon successful completion of this operation, the NetworkPort shall be in the "STOPPED" state.

HTTP Protocol

To stop a NetworkPort, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the NetworkPort where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/stop",
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.

5.16.8 Network Port Collection

A NetworkPortCollection resource represents the collection of NetworkPorts within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortCollection",
  "id": string,
  "count": number,
  "networkPorts": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPort",
      "id": string,
      ... remaining NetworkPort attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
```

```
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/NetworkPortCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <NetworkPort>
    <id> xs:anyURI </id>
    ... remaining NetworkPort attributes ...
  </NetworkPort> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.8.1 Operations

NOTE: The "add" operation requires a NetworkPortTemplate to be used (see 4.2.1.1).

When NetworkPorts are created via the global (Cloud Entry Point) NetworkPortCollection's "add" operation, they are automatically added to the corresponding Network's "NetworkPort" collection resource as well.

Upon successful processing of the "add" operation, unless otherwise specified via the NetworkPortTemplate "initialState" attribute, the state of the new NetworkPort shall be the value of the DefaultInitialState capability of the NetworkPort resource's ResourceMetadata, if defined. If no DefaultInitialState capability is defined the default value shall be "STOPPED." The semantics of "initialState"shall be equivalent to the Provider issuing the appropriate actions against the new NetworkPort to move it into that state.

If a Provider is unable to change the state of the new NetworkPort to the appropriate "initialState" (either as specified by the NetworkPortTemplate or as implied by the previous stated rules), then the NetworkPort creation shall fail.

5.16.9 Network Port Template

The Network Port Template is a set of Configuration values for realizing a NetworkPort. A NetworkPort Template may be used to create multiple NetworkPorts.

Name	NetworkPortTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkPortTemplate	
Attribute	**Type**	**Description**
initialState	*string*	The initial state of the new NetworkPort. Possible values include the non-transient states as specified by the NetworkPort "state" attribute (i.e., STARTED, STOPPED) and shall be determined by the actions supported by the Provider. Providers should advertise the list of available values via the NetworkPort ResourceMetadata "initialStates" capability. **Constraints:** **Provider:** support optional; mutable

		Consumer: support optional; read-write
network	*ref*	A reference to the network to be associated with this NetworkPort. When this Template is used to create a new NetworkPort via the global (Cloud Entry Point) NetworkPort Collection, this attribute shall be present. When this Template is used to create a new NetworkPort via a Network's NetworkPorts Collection then this attribute shall either be absent or shall have the same value as the "id" of the Network to which this NetworkPort is being added. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
networkPortConfig	*ref*	A reference to the NetworkPortConfiguration that will be used to create a NetworkPort from this NetworkPort Template. Note that the attributes of the NetworkPortConfiguration may be specified rather than a reference to an existing NetworkPortConfiguration resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
meterTemplates	*meterTemplates[]*	A list of references to Meter Templates that shall be used to create and connect a set of new Meters to the new NetworkPort. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
eventLogTemplate	*ref*	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new NetworkPort. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate resource. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "initialState": string, ?
  "network": { "href": string }, ?
  "networkPortConfig": {
    "href": string | ... NetworkPortConfiguration attributes ...
```

```
  },
  "meterTemplates": [
    { "href": string, ?
      ... MeterTemplate attributes ... ?
    }, *
  ], ?
  "eventLogTemplate": {
    "href": string, ?
    ... EventLogTemplate attributes ... ?
  }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<NetworkPortTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <initialState> xs:string </initialState> ?
  <network href="xs:anyURI"/> ?
  <networkPortConfig href="xs:anyURI"?>
    ... NetworkPortConfiguration attributes ... ?
  </networkPortConfig>
  <meterTemplate href="xs:anyURI"? >
    ... MeterTemplate attributes ... ?
  </meterTemplate> *
  <eventLogTemplate href="xs:anyURI"? >
    ... EventLogTemplate attributes ... ?
  </eventLogTemplate> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
```

```
</NetworkPortTemplate>
```

5.16.9.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Network Port Template Collection resource.

5.16.10 Network Port Template Collection

A Network Port Template Collection resource represents the collection of Network port Templates within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI":
    "http://schemas.dmtf.org/cimi/1/NetworkPortTemplateCollection",
  "id": string,
  "count": number,
  "networkPortTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortTemplate",
      "id": string,
      ... remaining NetworkPortTemplate attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/NetworkPortTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <NetworkPortTemplate>
    <id> xs:anyURI </id>
    ... remaining NetworkPortTemplate attributes ...
  </NetworkPortTemplate> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.10.1 Operations

This resource supports the Read and Update operations. Creation of new Network Port Template resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.16.11 Network Port Configuration

The set of configuration values representing the information needed to create a NetworkPort with certain characteristics.

Name	NetworkPortConfiguration	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration	
Attribute	**Type**	**Description**
portType	*string*	Indicates that a port will be used as an Access port (a member of the network) or a Trunk port that becomes a transport for multiple networks. Allowable values include: **ACCESS**: a member of a network. **TRUNK**: transport more than one network. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
classOfService	*string*	Indicates the Provider supported category, associated with a collection of attributes characterizing a level of a quality experience Example values: **GOLD**: High bandwidth, low latency, low jitter **SILVER**: An improved service experience over bronze for voice or video traffic **BRONZE**: Best effort The list of possible values, and their implied quality of service, is out of scope of this specifications. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "portType": string, ?
  "classOfService": string, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
```

```
  ...
}
```

XML media type: application/xml

XML serialization:

```
<NetworkPortConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <portType> xs:string </portType> ?
  <classOfService> xs:string </classOfService> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</NetworkPortConfiguration>
```

5.16.11.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Network Port Configuration Collection resource.

5.16.12 Network Port Configuration Collection

A NetworkPort Configuration Collection resource represents the collection of NetworkPortConfigurations within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI":
    "http://schemas.dmtf.org/cimi/1/NetworkPortConfigurationCollection",
  "id": string,
  "count": number,
  "networkPortConfigurations": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration",
      "id": string,
      ... remaining NetworkPortConfiguration attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
resourceURI="http://schemas.dmtf.org/cimi/1/NetworkPortConfigurationCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <NetworkPortConfiguration>
    <id> xs:anyURI </id>
    ... remaining NetworkPortConfiguration attributes ...
  </NetworkPortConfiguration> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.12.1 Operations

This resource supports the Read and Update operations. Creation of new NetworkPortConfiguration resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.16.13 Address

An Address represents an IP address, and its associated metadata, for a particular Network. When a Consumer creates an Address resource it is the semantic equivalent of asking for a static IP address that can then be associated with resources at a later point in time. Addresses that are manually created by Consumers shall not be automatically deleted when the resource (e.g., a Machine) that is using that Address is deleted because these manually created Addresses are expected to have a lifetime that is different from the resources that use them. Addresses that are created by Providers on the Consumer's behalf shall be deleted at the Provider's discretion. In particular, the Provider shall delete Addresses that it created on behalf of the Consumer when the resource that is using that Address is deleted or when the Address becomes disassociated from the resource.

Addresses that are created by Providers may be converted to ones that are under the Consumer's control (i.e., will not be deleted until explicitly requested by Consumers) by changing the "allocation" attribute from "dynamic" to "static," if this feature supported by Providers.

Name	Address	
Type URI	http://schemas.dmtf.org/cimi/1/Address	
Attribute	**Type**	**Description**
ip	*string*	The IP address assigned to a virtual interface. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
hostname	*string*	The DNS resolvable name associated with this network interface. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
allocation	*string*	The value is either "**dynamic**" or "**static**". Expresses whether this address is controlled by the Provider or Consumer. **Constraints:**

		Provider: support mandatory; mutable **Consumer:** support mandatory; read-only
defaultGateway	*string*	An IP address of a router that serves other networks. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
dns	*string[]*	The IP addresses of the Domain Name Services for host name to IP resolution. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
protocol	*string*	The selected network protocol, such as IPv4 or IPv6. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
mask	*string*	The network mask associated with this Address. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
network	*ref*	A reference to the Network with which this Address will be associated. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
resource	*ref*	A reference to the resource that is using this Address. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Address",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "ip": string,
  "hostname": string, ?
  "allocation": string,
  "defaultGateway": string, ?
  "dns": [ string, + ], ?
  "protocol": string,
  "mask": string, ?
```

```
  "network": { "href": string },
  "resource": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Address xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <ip> xs:string </ip>
  <hostname> xs:string </hostname> ?
  <allocation> xs:string </allocation>
  <defaultGateway> xs:string </defaultGateway> ?
  <dns> xs:string </dns> *
  <protocol> xs:string </protocol>
  <mask> xs:string </mask> ?
  <network href="xs:anyURI"/>
  <resource href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</Address>
```

5.16.13.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Address Collection resource.

5.16.14 Address Collection

An Address Collection resource represents the collection of Addresses within a Provider that are owned/managed by the Consumer or Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/AddressCollection",
```

```
  "id": string,
  "count": number,
  "addresses": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Address",
      "id": string,
      ... remaining Address attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/AddressCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Address>
    <id> xs:anyURI </id>
    ... remaining Address attributes ...
  </Address> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.14.1 Operations

NOTE: The "add" operation requires an AddressTemplate to be used (see 4.2.1.1).

5.16.15 Address Template

This resource captures the configuration values for realizing an Address. An Address Template may be used to create multiple Addresses.

Name	AddressTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/AddressTemplate	
Attribute	**Type**	**Description**
ip	*string*	The IP address assigned to a virtual interface. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
hostname	*string*	The DNS resolvable name associated with this network interface. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
allocation	*string*	A value of either "**dynamic**" or "**static**". Expresses whether this address is controlled by

		the Provider or Consumer. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
defaultGateway	*string*	An IP address of a router that serves other networks. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
dns	*string[]*	The IP addresses of the Domain Name Services for host name to IP resolution. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
protocol	*string*	The selected network protocol, such as IPv4 or IPv6. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
mask	*string*	The network mask associated with this Address. **Constraints:** **Provider:** support optional; mutable **Consumer:** support optional; read-write
network	*ref*	A reference to the Network with which this Address will be associated. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/AddressTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "ip": string,
  "hostname": string, ?
  "allocation": string,
  "defaultGateway": string, ?
  "dns": [ string, + ], ?
  "protocol": string,
  "mask": string, ?
  "network": { "href": string },
  "operations": [
```

```
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<AddressTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <ip> xs:string </ip>
  <hostname> xs:string </hostname> ?
  <allocation> xs:string </allocation>
  <defaultGateway> xs:string </defaultGateway>
  <dns> xs:string </dns> +
  <protocol> xs:string </protocol>
  <mask> xs:string </mask>
  <network href="xs:anyURI"/>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</AddressTemplate>
```

5.16.15.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Address Template Collection resource.

5.16.16 Address Template Collection

An Address Template Collection resource represents the collection of Address Template resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/AddressTemplateCollection",
  "id": string,
  "count": number,
  "addressTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/AddressTemplate",
```

```
      "id": string,
      ... remaining AddressTemplate attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/AddressTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <AddressTemplate>
    <id> xs:anyURI </id>
    ... remaining AddressTemplate attributes ...
  </AddressTemplate> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.16.1 Operations

This resource supports the Read and Update operations. Creation of new Address Template resources are supported via a POST to the "addLink" URI as described in clause 4.2.1.1.

5.16.17 Forwarding Group

A Forwarding Group represents a collection of Networks that route to each other.

Networks in a ForwardingGroup should all have the same "networkType" attributes, which prevents a Network with a "private" networkType attribute from being publicly forwarded because it is a member of a ForwardingGroup that also contains Networks with a "public" networkType attribute.

Providers shall not allow two Networks to be forwardable to each other unless they are explicitly connected by being part of a common ForwardingGroup.

Name	ForwardingGroup	
Type URI	http://schemas.dmtf.org/cimi/1/ForwardingGroup	
Attribute	**Type**	**Description**
networks	*collection [ForwardingGroupNetwork]*	A reference to the list of references to the Networks in this Forwarding Group. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroup",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "networks": [
    { "href": string }, +
   ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<ForwardingGroup xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <network href="xs:anyURI"> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</ForwardingGroup>
```

5.16.17.1 Collections

The following describes the collection resources owned by ForwardingGroups.

5.16.17.1.1 ForwardingGroupNetwork Collection

The resource type for each item of this collection is "ForwardingGroupNetwork", as defined as follows:

Name	ForwardingGroupNetwork
Type URI	http://schemas.dmtf.org/cimi/1/ForwardingGroupNetwork

Attribute	Type	Description
network	*ref*	A reference to a Network in the ForwardingGroup. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

JSON serialization:

```
{ "resourceURI":
    "http://schemas.dmtf.org/cimi/1/ForwardingGroupNetworkCollection",
  "id": string,
  "count": number,
  "forwardingGroupNetworks": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroupNetwork",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "network": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
      ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
 resourceURI="http://schemas.dmtf.org/cimi/1/ForwardingGroupNetworkCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <ForwardingGroupNetwork>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
```

```
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <network href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </ForwardingGroupNetwork> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.17.2 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the ForwardingGroup Collection resource.

5.16.18 Forwarding Group Collection

A Forwarding Group Collection resource represents the collection of Forwarding Groups within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroupCollection",
  "id": string,
  "count": number,
  "forwardingGroups": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroup",
      "id": string,
      ... remaining ForwardingGroup attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/ForwardingGroupCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <ForwardingGroup>
    <id> xs:anyURI </id>
    ... remaining ForwardingGroup attributes ...
```

```
  </ForwardingGroup> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.18.1 Operations

NOTE: The "add" operation requires a ForwardingGroupTemplate to be used (see 4.2.1.1).

5.16.19 Forwarding Group Template

This resource captures the configuration values for realizing a ForwardingGroup. A Forwarding Group Template may be used to create multiple ForwardingGroup.

Name	ForwardingGroupTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/ForwardingGroupTemplate	
Attribute	**Type**	**Description**
networks	*ref[]*	An array of references to the networks in this Forwarding Group. Array item name: network **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroupTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "networks": [
    { "href": string }, +
   ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<ForwardingGroupTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <network href="xs:anyURI"> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</ForwardingGroupTemplate>
```

5.16.19.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Forwarding Group Template Collection resource.

5.16.20 Forwarding Group Template Collection

A Forwarding Group Template Collection resource represents the collection of Forwarding Group Template resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI":
    "http://schemas.dmtf.org/cimi/1/ForwardingGroupTemplateCollection",
  "id": string,
  "count": number,
  "forwardingGroupTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroupTemplate",
      "id": string,
      ... remaining ForwardingGroupTemplate attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
 resourceURI="http://schemas.dmtf.org/cimi/1/ForwardingGroupTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
```

```
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <ForwardingGroupTemplate>
    <id> xs:anyURI </id>
    ... remaining ForwardingGroupTemplate attributes ...
  </ForwardingGroupTemplate> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.20.1 Operations

This resource supports the Read and Update operations. Creation of new Forwarding Group Template resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.17 Monitoring resources and relationships

Figure 6 illustrates the resources involved in tracking the progress of operations, as well as, metering and monitoring the status of other resources. Although this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor normative.

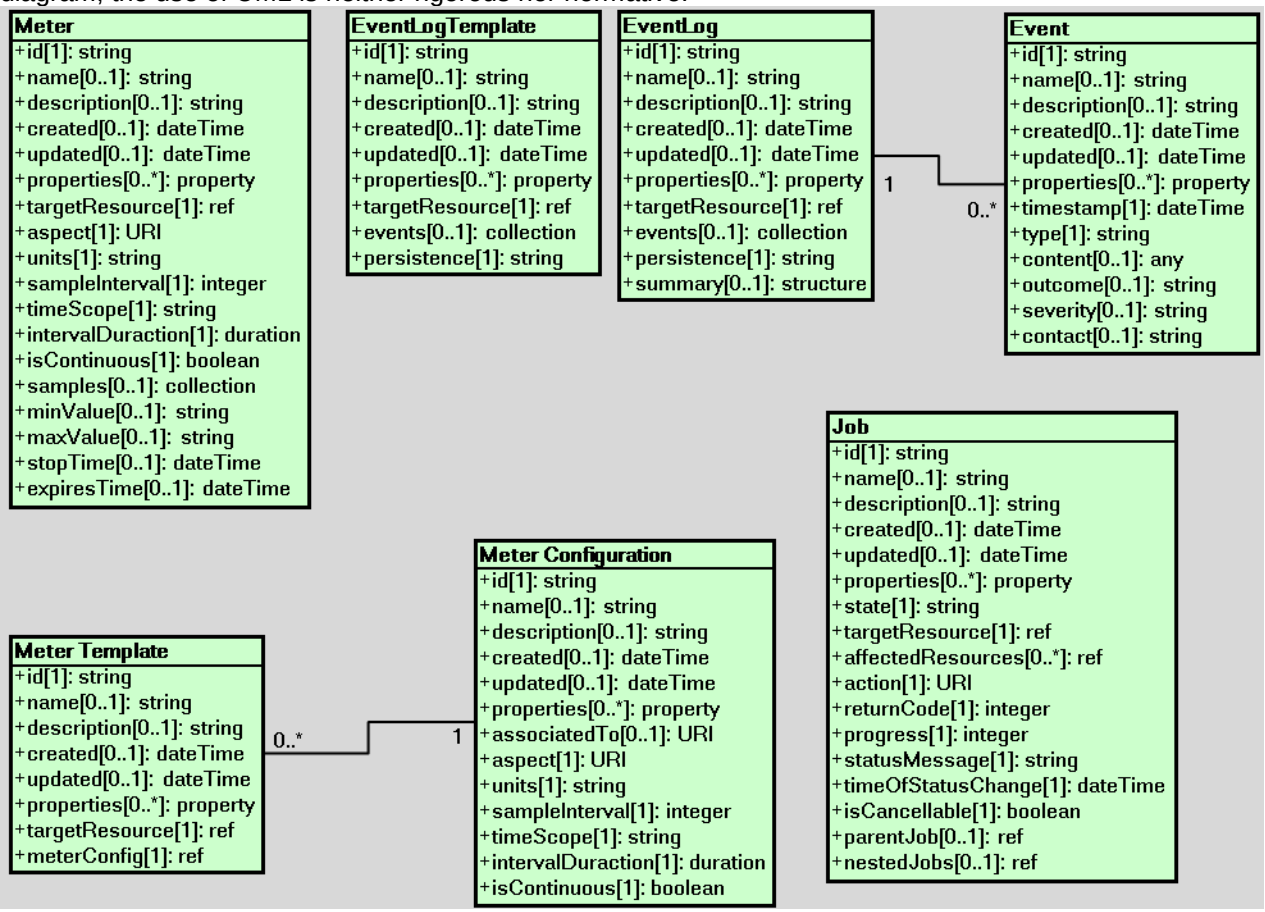


Figure 6 - Monitoring resources

5.17.1 Job

This resource represents a process (i.e., a sequence of one or more operations directed to accomplish a specific goal) that is performed by the Provider.

If a Provider supports exposing Job resources to Consumers, each request from a Consumer that would result in a change to the environment shall result in a Job resource being created and an absolute URI reference to that Job resource shall be made available to the requesting Consumer. Providers may create additional Job resources for Provider initiated operations if the Provider chooses to expose these Jobs to Consumers.

When a Job does not complete successfully (e.g., it is in the FAILED or STOPPED state), this specification does not place any requirements on the Provider to ensure that the affected resources are left in certain states. Based on the environmental conditions at that time, the Provider might choose to "undo" any impact of the operation; simply halt processing; attempt some kind of "cleanup" action; or choose to do something else. However, Providers shall list all resources impacted by the Job in the "affectedResources" attribute, thus allowing Consumers an opportunity to examine the state of each resource themselves. In cases where a resource has been deleted, references to that resource shall not appear in the "affectedResources" attribute.

The Job resource allows for nesting of Jobs. The determination of when a single operation is converted into multiple nested Jobs is out of scope of this specification. However, if there are nested Jobs, the top-most Job resource shall report the overall status of all Jobs and shall only be in a "SUCCESS" state if all nested Jobs are also in "SUCCESS" state. When nested Jobs are created, there is no requirement for the top-most Job resource to reference all affected resources in its "affectedResources" attribute. The Consumer will need to traverse the entire set of nested Jobs to determine the complete list of resources impacted by the Jobs.

Name	Job	
Type URI	http://schemas.dmtf.org/cimi/1/Job	
Attribute	**Type**	**Description**
state	*string*	The state of the process associated with this operation. Allowable values include: **QUEUED**: Indicates that the operation has not yet begun processing. **RUNNING**: Indicates that the operation is still being executed. **FAILED**: Indicates that the operation failed to complete successfully. **SUCCESS**: Indicates that the operation successfully completed. **STOPPING**: Indicates that the operation is in the process of being stopped. **STOPPED**: Indicates that the operation was stopped before completion. **The operations that result in transitions to the above defined states are defined in Section 5.17.1.1** **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
targetResource	*ref*	A reference to the top-level resource upon which the operation is being performed. Typically, this resource would be the resource on which the operation was invoked. Note that when an "add" Job is executed against a "Collection" resource (e.g. MachineCollection), the targetResource attribute shall reference the Collection

		resource - as that is the resource on which the operation was performed. Additionally, the newly created resource shall appear in the "affectedResources" attribute. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
affectedResources	*ref[]*	A list of references to resources that have been impacted by this Job. Note that this list shall always contain the "targetResource" reference. Array item name: affectedResource **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
action	*URI*	A URI that indicates the type of action being performed. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
returnCode	*integer*	The operation return code. The specific value will be specific to the implementation. Values in the range of 0 to 9999 are reserved for use by this specification. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
progress	*integer*	An integer value in the range 0 … 100 that indicates the progress of this Job. This value shall be 100 when the Job is no longer executing, regardless of the outcome. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
statusMessage	*string*	This attribute is a human-readable string that provides information about the operation. It is used to further qualify or provide additional information about the current status of the operation. For example, this attribute may indicate the reason why the operation failed, or whether the operation was cancelled by the Consumer or the Provider. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
timeOfStatusChange	*dateTime*	A timestamp indicating the last time that the status of the operation changed. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
parentJob	*ref*	A reference to the Job of which this resource is a subordinate. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
nestedJobs	*ref[]*	An array of references to a set of subordinate Job resources. Array item name: nestedJob **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Job",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "targetResource": { "href": string },
  "affectedResources": [ { "href": string }, + ],
  "action": string,
  "returnCode": number,
  "progress": number,
  "statusMessage": string,
  "timeOfStatusChange": date,
  "parentJob": { "href": string }, ?
  "nestedJobs": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Job xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateIime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
```

```
  <targetResource href="xs:anyURI"/>
  <affectedResource href="xs:anyURI"/> +
  <action> xs:anyURI </action>
  <returnCode> xs:integer </returnCode>
  <progress> xs:integer <progress>
  <statusMessage> xs:string </statusMessage>
  <timeOfStatusChange> xs:dateTime </timeOfStatusChange>
  <parentJob href="xs:anyURI"/> ?
  <nestedJob href="xs:anyURI"/> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
href="xs:anyURI"/> ?
  <xs:any>*
</Job>
```

5.17.1.1 Operations

This resource supports the Read, Update and Delete operations.Deleting a Job that is in the "RUNNING" state shall be the equivalent of first stopping the Job and then deleting it. A request to delete a running Job that does not support the "stop" action shall fail.

The following custom operations are also defined:**stop**

/link@rel: http://schemas.dmtf.org/cimi/1/action/stop

This operation shall stop a Job.

Input parameters: None.

Output parameters: None.

During the processing of this operation, the Job shall be in the "STOPPING" state.

Upon successful completion of this operation, the Job shall be in the "STOPPED" state.

HTTP protocol

To stop a Job, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the Job where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/stop",
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.

5.17.2 Job Collection

A Job Collection resource represents the collection of Jobs within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/JobCollection",
  "id": string,
  "count": integer,
  "jobs": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Job",
      "id": string,
      ... remaining Job attributes ...
    }, +
  ] ?
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/JobCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Job>
    <id> xs:anyURI </id>
    ... remaining Job attributes ...
  </Job> *
  <xs:any>*
</Collection>
```

5.17.3 Meter

This resource represents an available Meter of some property associated to a given resource.

When a Meter's "targetResource" is deleted all Meters associated with that resource shall also be deleted. In other words, deleting a resource-specific MetersCollection (e.g. a Machine's MetersCollection) shall also result in the deletion of the Meters referenced from that collection.

Name	Meter	
Type URI	http://schemas.dmtf.org/cimi/1/Meter	
Attribute	**Type**	**Description**
targetResource	*ref*	A reference to the resource to which the Meter is related. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
aspect	*URI*	A unique identifier representing the aspect of the resource being metered. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
units	*string*	The name of the used units, e.g., kilobits per second, CPU usage percentage, etc. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
sampleInterval	*integer*	The time between consecutive samples in seconds. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
timeScope	*string*	The time scope to which this meter's value applies. Two possible values: "Point" indicates that the Meter applies to a point in time. "Interval" indicates that the Meter applies to a time interval. For instance, it would be possible to define a Meter whose purpose is to provide the daily average CPU usage. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
intervalDuration	*duration*	The interval duration when the timeScope is set to "Interval". Possible values: hourly, daily, weekly, monthly or yearly. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
isContinuous	*boolean*	This value indicates whether or not the Meter value is continuous or scalar. Performance Meters are an example of a linear metric. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
samples	*collection [Sample]*	A reference to the list of taken samples **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
minValue	*string*	The expected minimal measure value. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only

maxValue	*string*	The expected maximum measure value. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
stopTime	*dateTime*	The time from which the meter stops tracking samples. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
expiresTime	*dateTime*	The time from which the Meter is not monitored anymore. It implies the deletion of the Meter after this time. Note that a Meter might be deleted before this time if the resource being metered is deleted. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "targetResource": { "href": string },
  "aspect": string,
  "units": string,
  "sampleInterval": number,
  "timeScope": string,
  "intervalDuration": string,
  "isContinuous": boolean,
  "samples": { "href": string }, ?
  "minValue": string, ?
  "maxValue": string, ?
  "stopTime": string, ?
  "expiresTime": string, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
```

```
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Meter xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <targetResource href="xs:anyURI"/>
  <aspect> xs:anyURI </aspect>
  <units> xs:string </units>
  <sampleInterval> xs:integer </sampleInterval>
  <timeScope> xs:string <timeScope>
  <intervalDuration xs:duration </intervalDuration>
  <isContinuous> xs:boolean </isContinuous>
  <samples href="xs:anyURI"/> ?
  <minValue> xs:string </minValue> ?
  <maxValue> xs:string </maxValue> ?
  <stopTime> xs:dateTime </stopTime> ?
  <expiresTime> xs:dateTime </expiresTime> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
href="xs:anyURI"/> ?
  <xs:any>*
</Meter>
```

5.17.3.1 Collections

The following describes the collection resources owned by Meters.

5.17.3.1.1 Sample Collection

The resource type for each item of this collection is “Sample”, defined as follows:

Name	Sample	
Type URI	http://schemas.dmtf.org/cimi/1/Sample	
Attribute	**Type**	**Description**
timestamp	*dateTime*	It indicates when the measure was taken (timeScope="Point"). When the timeScope is "Interval", it indicates the end of the time interval. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
value	*string*	It indicates the sampled value of the measure. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SampleCollection",
  "id": string,
  "count": number,
  "samples": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Sample",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "timestamp": string,
      "value": string
      ...
    }, +
  ], ?
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SampleCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Sample>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
```

```
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <sample timestamp="xs:dateTime" value="xs:string"/>
    <xs:any>*
  </Sample> *
  <xs:any>*
</Collection>
```

5.17.3.2 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Meter Collection resource. The deletion of a Meter shall remove the Meter from the targetResource's "meter" attribute.

The following custom operations are also defined:**start**

/link@rel: http://schemas.dmtf.org/cimi/1/action/start

This operation shall start a Meter.

Input parameters: None.

Output parameters: None.

Upon successful completion of this operation, the Meter shall start recording samples related to its associated resource.

HTTP protocol

To start a Meter, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/start" URI of the Meter where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/start",
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/start </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.**stop**

/link@rel: http://schemas.dmtf.org/cimi/1/action/stop

This operation shall stop a Meter.

Input parameters: None.

Output parameters: None.

Upon successful completion of this operation, the Meter shall no longer be recording samples related to its associated resource.

HTTP protocol

To stop a Meter, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the Meter where the HTTP request body shall be as described below.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/stop",
  "properties": { string: string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.

5.17.4 Meter Collection

A Meter Collection resource represents the collection of Meters within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterCollection",
  "id": string,
  "count": number,
  "meters": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
      "id": string,
      ... remaining Meter attributes ...
    }, +
```

```
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/MeterCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Meter>
    <id> xs:anyURI </id>
    ... remaining Meter attributes ...
  </Meter> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.17.4.1 Operations

NOTE: The "add" operation requires a MeterTemplate to be used (see 4.2.1.1).

When Meters are created via the global (Cloud Entry Point) MeterCollection's "add" operation, they shall be automatically added to the corresponding targetResource's "Meters" collection resource as well.

5.17.5 Meter Template

A Meter Template represents the information needed to create a new Meter.

Name	MeterTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/MeterTemplate	
Attribute	**Type**	**Description**
targetResource	*ref*	A reference to the resource that will be metered. The type of the resource shall be one of the "associatedTo" types listed in the Meter Configuration referenced. When this Template is used to create a new Meter via the global (Cloud Entry Point) Meters Collection, this attribute shall be present. When this Template is used to create a new Meter via a targetResource's Meters Collection then this attribute shall either be absent or shall have the same value as the "id" of the targetResource to which this Meter is being added. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
meterConfig	*ref*	A reference to the Meter Configuration that will be used to create a Meter from this Meter Template. Note that the attributes of the MeterConfiguration may be specified rather than a reference to an existing MeterConfiguration resource. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "targetResource": { string },
  "meterConfig": {
    "href": string | ... MeterConfiguration attributes ...
  },
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<MeterTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <targetResource href="xs:anyURI"/>
  <meterConfig href="xs:anyURI"?>
    ... MeterConfiguration attributes ... ?
  </meterConfig>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MeterTemplate>
```

5.17.6 Meter Template Collection

A Meter Template Collection resource represents the collection of MeterTemplate resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterTemplateCollection",
  "id": string,
  "count": number,
  "meterTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterTemplate",
      "id": string,
      ... remaining MeterTemplate attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/MeterTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <MeterTemplate>
    <id> xs:anyURI </id>
    ... remaining MeterTemplate attributes ...
  </MeterTemplate> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.17.6.1 Operations

This resource supports the Read and Update operations. Creation of new Meter Template resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.17.7 Meter Configuration

A Meter Configuration represents the definition of a Meter.

Name	MeterConfiguration	
Type URI	http://schemas.dmtf.org/cimi/1/MeterConfiguration	
Attribute	**Type**	**Description**
associatedTo	*URI[]*	An array of URIs that indicate the types of resources to which a Meter created from this configuration can be applied. The value space of these URIs is identical to that of ResourceMetadata.typeURI, which is a URI that uniquely identifies a resource type. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
aspect	*URI*	A unique identifier representing the aspect of the resource being metered. See the table below for the set of CIMI defined URIs. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
units	*string*	The human-readable name of the used units, e.g., kilobits per second, CPU usage percentage, etc. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
sampleInterval	*integer*	The time between consecutive samples in seconds. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
timeScope	*string*	The time scope to which the Meter value applies. Two possible values: "Point" indicates that the Meter applies to a point in time. "Interval" indicates that the Meter applies to a time interval. For instance, it would be possible to define a MeterConfiguration whose purpose is to provide the daily average CPU usage. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
intervalDuration	*duration*	The interval duration when the timeScope is set to "Interval." Possible values: hourly, daily, weekly, monthly, or yearly. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
isContinuous	*boolean*	This value indicates whether the Meter value is continuous or scalar. Performance Meters are an example of a linear metric. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterConfiguration",
  "id": string,
```

```
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "associatedTo": [
    { "href": string }, +
  ], ?
  "aspect": string,
  "units": string,
  "sampleInterval": number,
  "timeScope": string,
  "intervalDuration": string,
  "isContinuous": boolean,
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<MeterConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <associatedTo href="xs:anyURI"/> *
  <aspect> xs:anyURI </aspect>
  <units> xs:string </units>
  <sampleInterval> xs:integer </sampleInterval>
  <timeScope> xs:string </timeScope>
  <intervalDuration> xs:duration </intervalDuration>
  <isContinuous> xs:boolean </isContinuous>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
```

```
</MeterConfiguration>
```

The following table describes the "aspect" URIs defined by this specification. Providers may define new aspect URIs and it is recommended that these URIs be dereferencable such that Consumers can discover the details of the new aspect. For brevity the "URI" column in the table only shows the last part of the URI. It should be appended to: "http://schemas.dmtf.org/cimi/1/aspect/".

Aspect	Description
cpu	The percentage CPU usage of the resource. Typically associated with CEP, System, and Machine resources. For resources that group other resources (e.g., CEP or System resources), this aspect provides the aggregated percentage usage of the CPU.
memory	The amount of memory being used by the resource. Typically associated with CEP, System, and Machine resources. For resources that group other resources (e.g., CEP or System resources), this aspect provides the aggregated usage of the memory.
disk	The amount of disk being used by the resource. Typically associated with CEP, System, Machine, and Volume resources. For resources that group other resources (e.g., CEP or System resources), this aspect provides the aggregated disk usage.
bandwidth	The amount of network traffic. Typically associated with CEP, System, and Network resources. For CEP and System resources, this aspect provides the aggregated bandwidth of all the networks under them.
inputBandwidth	The amount of input bandwidth used by the resource. Typically associated with Machine, NetworkPort, and Volume resources. For Machine resources, this aspect provides the aggregated input bandwidth usage of all its network interfaces .
outputBandwidth	The amount of output bandwidth used by the resource. Typically associated with Machine, NetworkPort, and Volume resources. For Machine resources, this aspect provides the aggregated input bandwidth usage of all its network interfaces.

5.17.7.1 Operations

This resource supports the Read, Update, and Delete operations. Create is supported via the Meter Configuration Collection resource.

5.17.8 Meter Configuration Collection

A Meter Configuration Collection resource represents the collection of Meter Configurations within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterConfigurationCollection",
  "id": string,
  "count": number,
  "meterConfigurations": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterConfiguration",
      "id": string,
      ... remaining MeterConfiguration attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
```

```
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/MeterConfigurationCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <MeterConfiguration>
    <id> xs:anyURI </id>
    ... remaining MeterConfiguration attributes ...
  </MeterConfiguration> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.17.8.1 Operations

This resource supports the Read and Update operations. Creation of new Meter Configuration resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.17.9 Event Log

An resource that represents a registry of Events.

When an EventLog's "targetResource" is deleted the EventLog associated with that resource may also be deleted. In other words, deleting a resource (e.g. a Machine) may also result in the deletion of the EventLog referenced from that resource. This behavior is denoted by the EventLog.Linked capability.

When an EventLog is deleted all of its Events shall also be deleted.

Name	EventLog	
Type URI	http://schemas.dmtf.org/cimi/1/EventLog	
Attribute	**Type**	**Description**
targetResource	*ref*	A reference to the resource to which the Events are related. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
events	*collection [Event]*	A reference to the list of occurred Events. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-only
persistence	*string*	A value that indicates the persistence of the Events within the EventLog. For instance, daily, weekly, monthly, or yearly. Events that exceed the persistence duration may be deleted. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
summary	*<unnamed*	A summary of all the events present in the EventLog when the read operation is

<table>
<tr>
<td></td>
<td>structure></td>
<td>performed, grouped by severity.
Each summary attribute is an (unnamed) structure that has the following sub-attributes:
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>low</td><td>integer</td><td>Number of occurred Events with a low severity.
<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-only</td></tr>
<tr><td>medium</td><td>integer</td><td>Number of occurred Events with a medium severity.
<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-only</td></tr>
<tr><td>high</td><td>integer</td><td>Number of occurred Events with a high severity.
<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-only</td></tr>
<tr><td>critical</td><td>integer</td><td>Number of occurred Events with a critical severity.
<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-only</td></tr>
</table>
<u>Constraints:</u>
Provider: support mandatory; mutable
Consumer: support mandatory; read-only</td>
</tr>
</table>

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLog",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "targetResource": { "href": string },
  "events": { "href": string },
  "persistence": string,
  "summary": {
    "low": number,
    "medium": number,
    "high": number,
    "critical": number
  }, ?
  "operations": [
```

```
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<EventLog xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <targetResource href="xs:anyURI"/>
  <events href="xs:anyURI"/>
  <persistence> xs:string </persistence>
  <summary>
    <low> xs:integer </low>
    <medium> xs:integer </medium>
    <high> xs:integer <high>
    <critical> xs:integer </critical>
  </summary>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</EventLog>
```

5.17.9.1 Collections

The following describes the collection resources owned by EventLogs.

5.17.9.1.1 Event Collection

The resource type for each item of this collection is “Event” as defined in clause 5.17.13.

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/EventCollection",
  "id": string,
  "count": number,
  "events": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Event",
      "id": string,
```

```
      ... remaining Event attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/EventCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Event>
    <id> xs:anyURI </id>
    ... remaining Event attributes ...
  </Event> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.17.9.2 Operations

This resource supports the Read, Update, and Delete operations.

5.17.10 Event Log Collection

A Event Log Collection resource represents the collection of Event Logs within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogCollection",
  "id": string,
  "count": number,
  "eventLogs": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLog",
      "id": string,
      ... remaining EventLog attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/EventLogCollection"
```

```
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <EventLog>
    <id> xs:anyURI </id>
    ... remaining EventLog attributes ...
  </EventLog> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.17.11 Event Log Template

An EventLog Template represents the information needed to create a new EventLog.

Name	EventLogTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/EventLogTemplate	
Attribute	**Type**	**Description**
targetResource	*ref*	A reference to the resource to which the EventLog shall be connected. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write
persistence	*string*	A value that indicates the persistence of the Events in the new EventLog. For instance, daily, weekly, monthly, or yearly. Events that exceed the persistence duration may be deleted. **Constraints:** **Provider:** support mandatory; mutable **Consumer:** support mandatory; read-write

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "targetResource": { string },
  "persistence": string,
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
```

```
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<EventLogTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <targetResource href="xs:anyURI"/>
  <persistence> xs:string </persistence>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MeterTemplate>
```

5.17.12 Event Log Template Collection

A EventLog Template Collection resource represents the collection of EventLogTemplate resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This resource shall be serialized as follows:

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogTemplateCollection",
  "id": string,
  "count": number,
  "eventLogTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogTemplate",
      "id": string,
      ... remaining EventLogTemplate attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

XML serialization:

```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/EventLogTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
```

```
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <EventLogTemplate>
    <id> xs:anyURI </id>
    ... remaining EventLogTemplate attributes ...
  </EventLogTemplate> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.17.12.1 Operations

This resource supports the Read and Update operations. Creation of new EventLog Template resources are supported via a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.17.13 Event

An resource that represents the occurrence of an event within the managed infrastructure. Some examples of Events may be:

- Machine X has been rebooted by guest OS.
- Machine X is not responding to platform services.
- A new vCPU has been added to machine X following defined elasticity rules.

The scope of the Event concept is any information that the Provider is able to track within its infrastructure and that can constitute useful information for the Consumer. Possible examples include, but are not limited to, errors and inconveniences that occur in the (virtual) resources assigned to Consumers; Provider-initiated actions, such as maintenance tasks; etc.

Name	Event	
Type URI	http://schemas.dmtf.org/cimi/1/Event	
Attribute	**Type**	**Description**
timestamp	*dateTime*	The time of occurrence of the actual event. NOTE: This attribute should not be confused with the time of creation of the Event resource instance, which is captured in the common "created" attribute. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support optional; read-only
type	*URI*	A URI that uniquely identifies the type of the event. When the "content" attribute is present, this URI determines the actual data structure used for this content, e.g., to which schema it is associated. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
content	*any*	A polymorphic attribute that represents detailed event data, the type of which will vary with the event "type." Typically, a data structure; for example: In the case of a monitoring event, the content shall hold the target resource ID and type, measured attribute(s), and status value(s). In the case of an audit event conforming to the CADF model, the content shall hold the

		detailed event structure that complies with CADF event schema. In the case of a CIM Indication, the content shall hold the structure and attributes defined for such events. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support mandatory; read-only
outcome	*string*	A string value that characterizes the general significance of the event. A core set is defined that may be used regardless of the event type. For each event **type**, the definition of a core outcome value maybe refined in the context of this type, provided it does not conflict with the general meaning of the outcome given below. Core outcomes are: **Pending**: The event is about an action or process that is still ongoing. **Unknown**: The event is about a request or action that is not known by the Provider. **Status**: The event reports on the state or status of a resource. **Success**: The event reports on a successful outcome of some action or process. **Warning**: The event reports on a situation that requires attention or remedial action. **Failure**: The event reports on a failed outcome of some action or process. This set of core outcome values may be extended to accommodate possible outcomes of a specific event type. In this case, the extended set of values shall apply to all events of this type. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only
severity	*string*	A value indicating the Event severity. Possible values are: **critical** **high** **medium** **low** The meaning of the severity level may vary depending on the event "type." When such an attribute is not relevant to a particular type of event, it should be omitted. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only
contact	*string*	A reference to a contact point or processing point to handle the event. The actual type of this content (e.g., email address, phone# of helpdesk or staff, message queue, URL…) is dependent on, and determined by the event "type." This attribute is mutable as it may be determined after event creation by the Provider. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only

NOTE: There exists a legacy of several event models that have been standardized or designed for various domains relevant to IT. The objective in CIMI is not to elect one particular event model, but to select as top-level event attributes the most immediately relevant data useful for event processing in a Cloud environment. Additional event data may still be represented in the variable content attribute that allows for mapping other event models into a CIMI event.

The following describes the serialization of the resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Event",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "timestamp": string,
  "type": string,
  "content": any, ?
  "outcome": string, ?
  "severity": string, ?
  "contact": string, ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Event xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <timestamp> xs:dateTime </timestamp>
  <type> xs:string </type>
  <content> xs:any* </content> ?
  <outcome> xs:string </outcome> ?
  <severity> xs:string </severity> ?
  <contact> xs:string </contact> ?
  <xs:any>*
</Event>
```

The following table describes the "type" URIs that are defined or acknowledged by this specification. Additional types may be added by a Provider, for example to characterize external events mapped into CIMI events. It is recommended that these URIs be dereferencable such that Consumers can discover a more detailed description of the type. Event types defined by this specification will share the same base URI: http://schemas.dmtf.org/cimi/1/event/. For brevity, when the "Event Type" column in the table only shows a relative URI (e.g., state) it shall be appended to the end of this base URI.

<table>
<tr><th>Event Type</th><th>Description</th></tr>
<tr><td>state</td><td>Events of this type report state information about CIMI run-time resources such as instances of Machines, Systems, Networks, and Volumes. This information includes reports on any change in the "state" of these resources.

The content element associated with this event type has the following structure:
<table>
<tr><th>Data</th><th>Type</th><th>Description</th></tr>
<tr><td>resName</td><td>string</td><td>The name of the resource about the state of which is reported.

<u>Constraints:</u>
Provider: support optional; immutable
Consumer: support optional; read-only</td></tr>
<tr><td>resource</td><td>ref</td><td>The reference to the resource about the state of which is reported. (Note: This reference may become invalid because the event might outlive the resource.)

<u>Constraints:</u>
Provider: support mandatory; immutable
Consumer: support optional; read-only</td></tr>
<tr><td>resType</td><td>URI</td><td>URI denoting this resource type (same as the type URI associated with the Resource type for this resource).

<u>Constraints:</u>
Provider: support optional; immutable
Consumer: support optional; read-only.</td></tr>
<tr><td>state</td><td>string</td><td>The state reported for the resource. Shall be the same as the "state" attribute value (if any) of the run-time resource at the time the event is generated.

<u>Constraints:</u>
Provider: support mandatory; immutable
Consumer: support optional; read-only</td></tr>
<tr><td>previous</td><td>string</td><td>The previous state value, if the event reports a state change.

<u>Constraints:</u>
Provider: support optional; immutable
Consumer: support optional; read-only.</td></tr>
</table></td></tr>
<tr><td>alarm</td><td>Events of this type report errors or alarms occurring during management operations of Cloud resource. This information includes failures to provision resources, failures to fulfill requests to the CIMI interface, and any critical situation that needs be addressed in a timely manner.

The content element associated with this event type has the following structure:
<table>
<tr><th>Data</th><th>Type</th><th>Description</th></tr>
<tr><td>resName</td><td>string</td><td>The name of the resource associated with this alarm, if applicable.

<u>Constraints:</u>
Provider: support optional; immutable
Consumer: support optional; read-only.</td></tr>
<tr><td>resource</td><td>ref</td><td>The reference to the resource associated with this alarm, if applicable. (Note: This reference may become invalid because the event might outlive the resource.)</td></tr>
</table></td></tr>
</table>

			Constraints: **Provider:** support mandatory; immutable **Consumer:** support optional; read-only
	restype	*URI*	URI denoting, this resource type associated with this alarm, if applicable (same as the type URI associated with the Resource type for this resource). **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only
	code	*string*	An alarm code. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support optional; read-only
	detail	*string*	The detailed information associated with the alarm. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only

model

Events of this type report changes in the CIMI resource model, which includes creation, modification, and destruction of resource instances; and updates to metadata (resource extensions, capabilities and constraints, etc.).

The **content** element associated with this event type has the following structure:

Data	Type	Description
resName	*string*	The name of the main model resource affected by the modification. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only
resource	*ref*	The reference to the main model resource affected by the modification. (Note: This reference may become invalid because the event might outlive the resource.) **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support optional; read-only
resType	*URI*	URI denoting, this resource type (same as the type URI associated with the Resource type for this resource). **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only
change	*string*	The kind of modification reported (create/update/delete). **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support optional; read-only
detail	*string*	The detailed information associated with the change, typically the data for an update or creation, as used in a request. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only

access

Events of this type keep track of all requests to access some resource of a CIMI provider.

<table>
<tr><td></td><td>The content element associated with this event type has the following structure:
<table>
<tr><th>Data</th><th>Type</th><th>Description</th></tr>
<tr><td>operation</td><td>string</td><td>The method or name of the operation intended for this access (for the HTTP protocol, the HTTP method for the request).

<u>Constraints:</u>
Provider: support mandatory; immutable
Consumer: support optional; read-only</td></tr>
<tr><td>resource</td><td>ref</td><td>The reference of the primary resource supporting the operation (for the HTTP protocol, the resource URI or the URI associated with the operation). (Note: This reference may become invalid because the event might outlive the resource.)

<u>Constraints:</u>
Provider: support mandatory; immutable
Consumer: support optional; read-only</td></tr>
<tr><td>detail</td><td>string</td><td>The detailed information associated with the change, typically the data for an update or creation, as used in a request

<u>Constraints:</u>
Provider: support optional; immutable
Consumer: support optional; read-only</td></tr>
<tr><td>initiator</td><td>string</td><td>The details identifying the request initiator, in case that information can be associated with the request.

<u>Constraints:</u>
Provider: support optional; immutable
Consumer: support optional; read-only</td></tr>
</table></td></tr>
<tr><td>http://schemas.dmtf.org/cloud/audit/1.0/</td><td>Events of this type represent events that have audit significance, as defined by CADF (…). This type can be subdivided further by extending the URI path (e.g., http://schemas.dmtf.org/cloud/audit/1.0/event/security, for security audit events).

The content element associated with this event type has the same structure as the event serialization defined in CADF[…]:</td></tr>
</table>

The following describes the serialization of the "content" property for various types of events:

"state" event:

JSON serialization:

```
{ "id": string,
  ...
  "type": "http://schemas.dmtf.org/cimi/1/event/state",
  "content": {
    "resName": string,
    "resource" : { "href" : string },
    "resType" : string,
    "state" : string,
    "previous" : string ?
  }
  ...
}
```

XML serialization:

```
<Event xmlns="http://schemas.dmtf.org/cimi/1">
  ...
  <type> http://schemas.dmtf.org/cimi/1/event/state </type>
  <content>
    <resName> xs:string </resName>
    <resource href="xs:anyURI"/>
    <resType> xs:anyURI </resType>
    <state> xs:string </state>
    <previous> xs:string </previous> ?
  </content> ?
  ...
</Event>
```

"alarm" event:

JSON serialization:

```
{ "id": string,
  ...
  "type": "http://schemas.dmtf.org/cimi/1/event/alarm",
  "content": {
    "resName": string ?
    "resource" : { "href" : string }, ?
    "resType" : string ?
    "code" : string,
    "detail" : string ?
  }
  ...
}
```

XML serialization:

```
<Event xmlns="http://schemas.dmtf.org/cimi/1">
  ...
  <type> http://schemas.dmtf.org/cimi/1/event/alarm </type>
  <content>
    <resname> xs:string </resname> ?
    <resource href="xs:anyURI"/> ?
    <restype> xs:anyURI </restype> ?
    <code> xs:string </code>
    <detail> xs:string </detail> ?
  </content> ?
```

```
  ...
</Event>
```

"model" event:

JSON serialization:

```
{ "id": string,
  ...
  "type": "http://schemas.dmtf.org/cimi/1/event/model",
  "content": {
    "resName": string, ?
    "resource" : { "href" : string }, ?
    "resType" : string, ?
    "change" : string,
    "detail" : string ?
  }
  ...
}
```

XML serialization:

```
<Event xmlns="http://schemas.dmtf.org/cimi/1">
  ...
  <type> http://schemas.dmtf.org/cimi/1/event/model </type>
  <content>
    <resname> xs:string </resname> ?
    <resource href="xs:anyURI"/> ?
    <restype> xs:anyURI </restype> ?
    <change> xs:string </change>
    <detail> xs:string </detail> ?
  </content> ?
  ...
</Event>
```

"access" event:

JSON serialization:

```
{ "id": string,
  ...
  "type": "http://schemas.dmtf.org/cimi/1/event/access",
  "content": {
    "operation": string,
    "resource" : { "href" : string },
    "detail" : string, ?
    "initiator" : string ?
```

```
    }
  ...
}
```

XML Serialization:

```
<Event xmlns="http://schemas.dmtf.org/cimi/1">
  ...
  <type> http://schemas.dmtf.org/cimi/1/event/access </type>
  <content>
    <operation> xs:string </operation>
    <resource href="xs:anyURI"/>
    <detail> xs:string </detail> ?
    <initiator> xs:string </initiator> ?
  </content> ?
  ...
</Event>
```

5.17.13.1 Operations

This resource supports the Read, Update, and Delete operations.

6 Security considerations

There are many security mechanisms that can be used in conjunction with this specification. This specification does not mandate any particular mechanism(s). Providers shall provide enough information about their security mechanisms so that the Consumer can implement the necessary algorithms to successfully communicate with the Provider.

ANNEX A
(normative)

OVF support in CIMI

This annex details how elements of the OVF descriptor are mapped to CIMI resources and their attributes. This definition allows the import of an OVF package to create multiple CIMI resources. This is done by specifying a reference to an OVF package in the import operation of a System Collection or System Template Collection (the Media Type at that URI shall be “application/ovf”). Please reference DSP0243 for more information about OVF.

Support for OVF import and export is optional for a Provider and it is an implementation choice as to how many of the attributes in the OVF package are exposed through CIMI resources. A Provider may support the import of OVF package for only Systems, only System Templates or both. Support for the actual import and export of OVF packages will typically be handled by a hypervisor under the management of the CIMI implementation, and thus the CIMI resources that are created reflect what the hypervisor did upon import and form a “View” into the results.

The import of an OVF package can be reflected in the creation of templates that can be later used to create Systems, Machines and other component resources. The import of an OVF package can also be used to directly create Systems, Machines and other component resources, bypassing the step of creating templates.

Clause 5.13.4 details how to import an OVF file to create a System Template (and component resources). The System Template thus created will contain a reference to a Machine Template for every VirtualSystem that is defined in the OVF Descriptor VirtualSystemCollection. Note that CIMI currently allows Systems of Systems, so for each VirtualSystemCollection encountered in a nested set of collections, a separate System Template is created within the parent System Template with Machine Templates for each of the contained VirtualSystems in that VirtualSystemCollection.

The values of the attributes for the Machine Template are taken from the VirtualHardwareSection of the VirtualSystem description (required in OVF). If multiple VirtualHardwareSections are used for a given VirtualSystem (allowed in OVF), the result is implementation dependent, but the implementation might choose a Machine Template from an existing (perhaps static) set that best matches one of the VirtualHardwareSections. Items in the VirtualHardwareSection are mapped to CIMI Machine Configuration properties and the corresponding Machine Configuration resource is created and linked to from the created Machine Template for that VirtualSystem.

The CIMI Volume Templates are created according to the DiskSection of the OVF Descriptor and can be shared among multiple VirtualSystems (CIMI Machine Templates) defined in the OVF Package. In addition, a new CIMI Machine Image resource may be created from the DiskSection if an ovf:fileRef for the virtual disk content is specified.

The CIMI Network Templates are created according to the NetworkSection of the OVF Descriptor along with the Connection elements in the various VirtualHardwareSections that refer to these named networks.

Clause 5.13.2.1 details how to import an OVF file to create a System (and component resources). The System thus created will contain a reference to a Machine for every VirtualSystem that is defined in the OVF Descriptor VirtualSystemCollection. Note that CIMI currently allows Systems of Systems, so for each VirtualSystemCollection encountered in a nested set of collections, a separate System is created within the parent System with Machines for each of the contained VirtualSystems in that VirtualSystemCollection.

The values of the attributes for the Machine are taken from the VirtualHardwareSection of the VirtualSystem description (required in OVF). If multiple VirtualHardwareSections are used for a given VirtualSystem (allowed in OVF), the result is implementation dependent. Items in the VirtualHardwareSection are mapped to CIMI Machine Configuration properties and the corresponding Machine Configuration resource is created and linked to from the created Machine for that VirtualSystem.

The CIMI Volumes are created according to the DiskSection of the OVF Descriptor and can be shared among multiple VirtualSystems (CIMI Machines) defined in the OVF Package. In addition, a new CIMI Machine Image resource may be created from the DiskSection if an ovf:fileRef for the virtual disk content is specified.

The CIMI Networks are created according to the NetworkSection of the OVF Descriptor along with the Connection elements in the various VirtualHardwareSections that refer to these named networks.

ANNEX B
(informative)

XML Schema

The XML Schema for the XML serialization of the CIMI model can be found at:

http://schemas.dmtf.org/cimi/1/DSP8009_1.0.xsd

The schema provided does not intend to reflect every single modeling constraint and requirement specified in the model. This schema is designed to apply more broadly to any model-related serialized material found in Consumer requests as well as in Provider responses, and is intended to provide a preliminary, non-exhaustive syntactic check on these. In particular future updates of this specification may intermix new XML elements into the resources using the current CIMI namespace to resources. The schema that is provided is just a starting-point for those who would find it useful and it might need to be modified based on specific application's needs.

ANNEX C
(informative)

Change log

Version	Date	Who	Description
0.0.1	10/15/10	Gil, Jack	Initial Draft
0.0.2	10/19/10	Jack	Adding the attribute descriptions and high level operational descriptions on the entities
0.0.3	10/29/10	Gil	Add section on the "Initial Scenario" and the mapping of its required use cases to our model.
0.0.4	11/15/10	Gil	Removed 'definition' attribute from System Template, Machine Template, Volume Template, and Network Template (per 912). Added "networkInterfaces" attribute to Machine with sub-properties that define IP address – added "protocol", "subnet_mask", "default_gateway" and "dns_servers to Netowork entity (per 910). Removed inline issues and created issues 928, 929, and 930.
0.0.5	11/17/10	Gil	Change "Cloud Site" to "Site per 882. Added Job entity and removed 'progress' attributes per 911. Added structure to Machine/disks and Machine/volumes per 915.
0.0.6	12/01/10	Gil	Removed "jobs" attributes from System and System Template to complete 911. Added "capacity" and "format", removed "type" from Machine/volumes to complete 915.
0.0.7	12/10/10	Gil	Added Image entity to resolve 935. Added new initial scenario to resolve 994.
0.0.8	12/15/10	Gil	Removed "os" attribute from Machine Template to resolve 1032. Added quantity/units sub-properties to describe memory and disk sizes and capacities for Machines and Machine Templates to resolve 1009. Removed "based_on" attribute from System, Machine, Volume, and Network to resolve 1001. Removed all template property descriptions to the effect that "changes to [this attribute] should correspondingly evaluate the [objects] that have been instantiated based on this [object template]" to resolve 1005.
0.0.9	01/05/11	Gil	Add Update operation to all entities as resolution to 1003. Change "Image" entity to "Machine Image" and updated description to resolve 1026. Fixed some capitalization and terminology inconsistencies.
0.0.10	01/18/11	Gil	Add Provider entity as resolution to 1043. Change "params" attribute to "properties" in all entities with that attribute – resolves 1002.
0.0.11	01/19/11	Gil	Move "format" attribute from the Machine-Volume connection to Volume itself and remove "capacity" attribute from the Machine-Volume connection to resolve 956. Add "properties" attribute to all entities that lacked it as completion of 1002. Add "Read" operation to all entities that lacked it; homogenize the description of the "Read" operation across all entities - 1049
0.0.12	01/26/11	Gil	Further edits to tweak the resolution for 1043.
0.0.13	02/02/11	Gil	Added "Create new Machine Image from existing Machine" scenario as per the decision of the WG on 02/02/2011.
0.0.14	02/09/11	Gil	Added "job_time" property to Provider entity as resolution of 1038. Renamed Machine Template to Machine Configuration and created a new Machine Template entity that reflects the resolution of 1045.
0.0.15	02/21/11	Gil	Added definition of "Template" as resolution of 1063. Changed definitions of Machine Configuration and Machine image as resolution

Version	Date	Who	Description
			of 1069. Updated UML diagram to remove all relationships that weren't explicitly defined as attributes of the relevant entities.
0.0.16	03/07/11	Gil	Add additional text to description of Template in section 3.2 to resolve 1044. Add column to all entity tables to describe attribute data types to resolve 1073. Changed the 'templates' attribute of Site to 'systemtemplates' to resolve 1075. Changed the description of Machine Configuration to resolve 1079.
0.0.17	03/22/11	Gil	Change attributes on Job entity to resolve 1080. Added "guestInterface" attribute to Machine Configuration, Machine, Volume Template, and Volume to resolve 1083. Changed description of the operations on Machine Configuration to resolve 1084. Moved "hostname" attribute of Machine to a sub-property of the "networkinterface" to resolve 1087. Added "volumes" and "volumetemplates" attributes to Provider entity to resolve 1089. Removed "state" attributes from System Template and Machine Template to resolve 1093.
0.0.18	03/23/11	Gil	Add constraint on relative URIs to heading of Section 4.2, "Attributes", to resolve 1100. Add "volume" sub-attribute to "volumes" attribute of Machine to resolve 1110. Applied consistent camelCased naming to resolve 1111. Added definitions for "immutable/mutable" and "writeable/read-only" to heading of Section 4.2, "Attributes"; revised all mentions of immutable and mutable to agree with these definitions to resolve 1126.
0.0.19	03/30/11	Gil	Add table that defines attributes common to all entities and remove those same attributes from the entity tables to resolve 1094. Add Volume Configuration and Volume Image entities to resolve 1096.
0.0.20	04/06/11	Gil	Add sub-section that describes unit attributes in both base-10 and base-2 (e.g., kilobyte, kibibyte), change memory units to base-2 names (e.g., kibibyte, mebibyte), change unit designators to singular, add reference to IEC 80000-13:2008 – all to resolve 1101.
0.0.21	04/13/11	Doug	Renamed title of section 4.2 per 1153. Expanded the text for the create operation of a System per 999. Added the delete operation to the list of ops for Network per 1011. Converted all pointers to URIs instead of the name of the type its points to per 1129.
0.0.22	04/20/11	Doug	Added section 4.2.1 per 1113 and 1115. Modified the definition of Site.Create per 1132. Modified Site description per 1133. Moved 'bootable' from volume Config to Volume Image per 1137. Removed 'local' from Volume and Volume Config per 1138. Modified definition of Volume per 1139. Removed 'autoDelete' from Volume per 1140.
0.0.23	04/27/11	Gil	Changed description of unit values in Section 4.2.2.1 to resolve 1157. Change places where "mutable" is meant as "writeable" to use "writeable"; add Update operation to Provider entity to resolve 1158.
0.0.24	05/11/11	Gil	Added Machine Admin entity to resolve 1164. Added text describing the requirements on when a Job entity is created to resolve 1166.
0.0.25	05/25/11	Gil	Added new Section 4.2 describing the facilities for retrieving metadata about the entities, added "Entity Metadata" to UML diagram, added "Type URI" to each entity type - to resolve 1135. Changed description of VolumeConfiguration:format and removed Volume:format to resolve 1136.
0.0.26	06/01/11	Gil	Added volumes and volumeTemplates attributes to the Machine Templates entity to resolved 1155. Changed description of Provider entity to resolve 1174. Chaged the description of the SystemTemplate.Update operation to resolve 1175. Added Section 6 "Security" (plus relevant definitions) to resolve 1178.

Version	Date	Who	Description
0.0.27	06/07/11	Gil	Merged Site and Provider entities into new Cloud Entry Point entity to resolve 1192 and 1196. Added operations and options to Machine and Machine template to resolve 1204. Replaced "Create and deploy a Machine using a Provider created Machine Template" scenario and added "Create a Machine by passing a Machine Template by value", and "Create a Machine using a User created Machine Template" to resolve 1205. Added "Create a Machine Template by specifying individual components" and "Create a Machine Template from a template file" scenarios to resolve 1206. Added "Create new Machine Image from an image file" and "Create new Machine Image from Machine instance" scenarios to resolve 1207.
0.0.28	06/09/2011	Gil	Added 'MeterTemplate' and 'Meter' entities and references to resolve 908. Added Event and EventLog entities and references to resolve 909. Added 'Cancel' operation and 'isCancellable' attribute to the Job entity to resolve 1012. Added terminology definition for "Configuration" to resolve 1191.
0.0.29	06/20/2011	Gil	Removed "Role in Use Cases" rows from entity tables to resolve 1223. Fixed miscellaneous typos and miscapitalizations to resolve 1226.
0.0.30	07/28/2011	Gil, Doug	Merged CM model and HTTP protocol documents. Added definitions of "Consumer" and "Provider" to Section 3, "Terms and Definitions"; made various changes to use these terms consistently throughout the document to resolve 1180. Fix serializations to resolve 1219. Added support for partial updates to resolve 1154. Fixed the CloudEntryPoint to point to the Collection entities to resolve 1238. Tweaked the uri field of EntityMetadata to resolve 1254. Added resolution of 1171 from HTTP protocol doc. Changed Network Template and Network entities, added VSP Template, VSP Configuration, and VSP to resolve 1010, 1085, 1086, and 1088.
0.0.31	08/01/2011	Doug	Removed text that duplicates the HTTP spec to resolve 1193.
0.0.32	08/22/2011	Doug	Added .../xs:any* to show explicit extensibility points to resolve 1271. Added section 4.1.10 (Serialization of Array) and s/*/+/g on array children to resolve 1270.
0.0.33	08/24/2011	Doug	Used CIMI and Cloud Infrastructure Management Interface where appropriate. Added section 5.1 which points to the CIMI-CIM and CIMI-RNG docs. Add WIP front-matter.
0.0.34	08/28/2011	Doug	s/The follow describes/The following describes/. Fixed the heading style on the "Entities" section - it lost its style at some point. Added the serialization headers for the EntityMetadata JSON/XML - to match the other entities in the model. s/Entity/EntityMetadata/ in the XML to match the entity type and be more descriptive.
0.0.35	08/31/2011	Gil, Doug	Re-factor ER diagram into separate sub-diagrams and re-organize sections around these diagrams. Changed title of doc and added resolution of issue 1310.
0.0.36	09/07/2011	Brightleaf	Various edits as part of the WIP release.
0.0.37	09/09/2011	Gil	Changed description of Job:isCancellable to resolve 1240. Add Volumes and Networks to System entity to resolve 1245. Changed title of Section 6.1.6 to "Control Machine State" to resolve 1246. Added 'status' attribute to Network and changed VSP:state to VSP:status to resolve 1255. Changed descriptions of 'imageLocation' and 'imageData' for both MachineImage and VolumeImage to resolve 1264. Changed wording in 4th paragraph of 4.2.1.3.1 to resolve issue 1266.
0.0.38	09/09/2011	Doug	Removed empty row in Machine table, filled in empty "Optionality"

Version	Date	Who	Description
			cells for CEP.
0.0.39	09/12/2011	Doug	Added section "4.2.1 Operational Principles" to resolve HTTP issue 1172.
0.0.40	09/21/2011	Doug	Lots of minor editorial changes to resolve 1269.
0.0.41	09/21/2011	Doug	Added text about URIs to resolve issue 1267. Modified the "properties" attribute to resolve issue 1352. Fixed SystemTemplate, it was missing some name, description and networkInterface definition attributes in the pseudo-schema. Added "Model Semantics and Conventions" section to resolve issue 1274. Made CEP.EntityMetadata a URI[] instead of a map to resolve issue 1243. Make 'stop' action URIs consistent to resolve issue 1364.
0.0.42	10/04/2011	Doug	Added typographical convention and preamble text to terms & def'n section to resolve issue 1272. Removed "format" and "attachmentPoint" from Machine.disk to resolve issue 1241. Removed disk.guestInterface from Machine and MachineConfig to resolve issue 1242. Changed most uses of URI to "ref" in the model tables to resolve issue 1351. Changed 'uri' to "self" on entities to resolve 1220.
0.0.43	10/04/2011	Doug	Added support for CIMISelect query parameter to resolve issue 1384. Clarified the optionality of the HTTP version header to resolve issue 1363.
0.0.44	10/05/2011	Doug	Added start/restart to Machine's operations resolve issue 1369. Cleaned up some text around Jobs to resolve issue 1194. Tweak the optionality of some attributes to resolve issue 1412. Add support for operations in EntityMetadata per issue 1168. Added the definition of optional, mandatory and condition to resolve issue 1339. Also moved some of the high-level topics about the model (units, identifier, ...) to a common spot at the start of section 5.
0.0.45	10/06/2011	Doug	Moved EntityMetadata into the Entities section to resolve issue 1415. Add some clarifying text about routingGroups to resolve issue 1413. Replace status with state on select resources to resolve issue 1095. Define what 'ref' maps to for REST to resolve issue 1409. Add pious advice about some network properties to resolve issue 1259. Clean up some of the pointers in EventLog, Meter and Event to resolve issue 1383. Add start/stop operations to Meter to resolve issue 1237. Clarify the behavior when updating read-only properties to resolve issue 1118. Adding RoutingGroup as a new entity to resolve issue 1260.
0.0.45a	10/12/2011	Gil	Removed requirement to support TLS NULL cipher to resolve 1244. Updated diagrams to match changes in the text.
0.0.46	10/18/2011	Doug	Added recommendation to use partial updates to avoid overwriting changes to resolve 1360. Removed attr_regex feature to resolve 1418. Clarify just how opaque our URIs really are to resolve 1417.
0.0.47	10/26/2011	Doug	Added an 'extensibility" section to resolve issue 1356. Explain what a missing attributes in the serializations mean, and fix some Optional vs Mandatory flags to resolve issue 1114.
0.0.48	11/03/2011	Doug	Added text around our versioning scheme to resolve issue 1119.
0.0.49	11/09/2011	Doug	Add new scope text to resolve issue 1435. Fix the JSON serialization of "properties" to resolve issue 1436.
0.0.50	11/14/2011	Doug	Added support for enum/query support for collections to resolve issue 1405.
0.0.51	11/28/2011	Doug	Updated description of HTTP error code 501 to resolve issue 1442.
0.0.52	11/30/2011	Doug	Remove Bibliography to resolve issue 1443. Add networkInterfaces to MachineTemplate in machine.create() to resolve issue 1460. Added pause and resume operations to Machine to resolve issue 1434.

Version	Date	Who	Description
0.0.53	12/06/2011	Doug	s/using/used/ to resolve issue 1466. Removed the word "use" to resolve issue 1465. Removed section 6 to resolve issue 1464. Changed the use of the word "avoid" to resolve issue 1469. Add 'aspect' to Meter(Template) to resolve issue 1444. Tweaked the note at the end of MachineConfig to resolve issue 1454.
0.0.54	12/07/2011	Doug	Removed 'protocol' from Machine and Volume entities to resolve issue 1247. Complete the definition of SystemTemplate, add MachineAdminTemplate and RoutingGroupTemplate to resolve issue 1368. Updated the state values on many entities to resolve issue 1446. Use Job as the error response message and allow hierarchical jobs to resolve issue 1452. s/Network/VSP/ in some VSP attributes to resolve issue 1471.
0.0.55	12/08/2011	Doug	Add support for capturing a Machine to a MachineImage to resolve issue 1448.
0.0.56	12/08/2011	Doug	Clarify initial state of new Machines to resolve issue 1478.
0.0.57	12/14/2011	Doug	Minor typos to resolve issue 1486.
0.0.58	01/05/2012	Doug	Added an "Operations" section (5.6) to resolve issue 1257.
0.0.59	01/11/2012	Doug	Added 'suspend' to allowable actions for a Network in STARTED state to resolve issue 1500. s/self/id/ to resolve issue 1496. Add the notion of operations to '5.1 Extensions' to resolve issue 1511. Minor wording fix to resolve issue 1502. Lots of minor typos to resolve issue 1495.
0.0.60	01/18/2012	Doug	Update boilerplate info.
0.0.61	01/19/2012	Doug	Tweak to security text to resolve issue 1521. Add Protocol Authentication section to resolve issue 1520. Added updated time to resolve issue 1485. Add snapshots to machine image to resolve issue 1027.
0.0.62	01/25/2012	Gil, Doug	Modify the Machine UML diagram to resolve issue 1507. Make MachineAdmin.password write-only to resolve issue 1473. Added capabilities to resolve issue 1488.
0.0.63	02/01/2012	Doug	Fix the indenting of a paragraph to resolve issue 1533. Add more details of the types used to resolve issue 1407. Add cpuArch to Machine and MachineConfig to resolve issue 1217. Change Machine Admin to Credentials to resolve issue 1532. Add a forward and ack section to resolve issue 1530. Add serialization rules to resolve issues 1453 and 1195.
0.0.64	02/09/2012	Doug	Fix used of "*Link" phrases to resolve issue 1493.
0.0.65	02/15/2012	Doug	Add consumer and provider constraints to each attribute to resolve issue 1515. Fix specification of "number" in some resources to resolve issue 1501. Clarify pass-by-value attributes and add some related capabilities to resolve issue 1497.
0.0.66	02/23/2012	Doug	Clarify how Meters are created to resolve issue 1547. Clear up whether networks can be connected when not part of the same routingGroup to resolve issue 1508. Clarify routing of RoutingGroups to resolve issue 1499. Use "passive" instead of "standby" for Networks to resolve issue 1556.
0.0.67	02/29/2012	Doug	Add quotes to strings in CIMISelect to resolve issue 1557.
0.0.68	03/07/2012	Doug	Remove job_time to resolve issue 1568. Clarify the use of common attributes to resolve issue 1571. Clarify 'method' to resolve issue 1570. Add MixedNetwork capability to resolve issue 1566. Add stop action to Job to resolve issue 1572. Add DefaultInitialState to MachineTemplate to resolve issue 1573. Add some text to Job description to resolve issue 1574.
0.0.69	03/14/2012	Doug	Disallow digits for identifier startChars to resolve issue 1599. Add precedence to verion header to resolve issue 1594. Minor edits to

Version	Date	Who	Description
			Attribute Constraints to resolve issue 1565. Remove hrefs from CIMISelect examples to resolve issue 1593. Don't duplicate Template attributes to resolve issue 1592. Put types in italics to resolve issue 1470. Add CADF stuff to Events to resolve issue 1541.
0.0.70	03/20/2012	Doug	Clarify that the Job header is an absolute URI to resolve issue 1606. Use application/json and application/xml as the media-types to resolve issue 1456. Add userData support to resolve issue 1483. Define the semantics of System.delete to resolve issue 1558. Add text around URI resolve algorithm to resolve issue 1472. Remove "supportsSnapshot" to resolve issue 1479.
0.0.71	03/21/2012	Doug	Clean-up EntityMetadata to resolve issue 1596. Add networkInterface.network to resolve issue 1578. Add Machine.InitialStates capability to resolve issue 1484. Allow for Templates to be overridden to resolve issue 1516. Add Address entity to resolve issue 1445.
0.0.72	03/23/2012	Doug	Remove references to RelaxNG doc to resolve issue 1635. Add support for OVF import/export to resolve issue 1447.
0.0.73	03/27/2012	Doug	Redo how collections are handled to resolve issue 1359.
0.0.74	03/28/2012	Doug	Make collections use generic wrappers to resolve issue 1644.
0.0.75	03/28/2012	Doug	Use 'any' instead of 'abstract' type to resolve issue 1595. Clarify some aspects around Jobs to resolve issue 1576. Make CPU an integer to resolve issue 1636. Add a force flag to machine.stop/restart to resolve issue 1577.
0.0.76	04/04/2012	Doug	Removed "imageData" from MachineImage and VolumeImage to resolve issue 1203. Added Addresses and AddressTemplates to CEP to resolve issue 1647. Revert some collections to arrays to resolve issue 1648.
0.0.77	04/11/2012	Doug	Update version of doc to 'e'. Fix typos to resolve issue 1653. Modify the general REST usage section to resolve issue 1630.
0.0.78	04/19/2012	Doug	Remove X- from CIMI http headers to resolve issue 1649. Reduce Volume support to just "mapped" to resolve issue 1531.
0.0.79	04/26/2012	Cathi	A Brightleaf review of v77. Technically this is a regression since it doesn't include the edits from v78 - those will be added back in for v80.
0.0.80	04/26/2012	Doug	Add back in the edits from v78.
0.0.81	04/26/2012	Doug	Add more text to the description of CEP to resolve issue 1688. Add a capability example to resolve issue 1686. Use http://schemas.dmtf.org/cimi/1 as our namespace to resolve issue 1641. Big change to the networking stuff to resolve issues 1639,1638, 1637, 1633, 1626, 1625,1624, 1623. Reordering the sections (per the resolution of the networking issues) will be in the next version.
0.0.82	04/26/2012	Doug	Reorder some sections as part of the previous 8 networking issues.
0.0.83	05/02/2012	Doug	RESTful review to resolve issue 1710.
0.0.84	05/02/2012	Doug	Use "Resource" instead of "Entity" to resolve issue 1711.
0.0.85	05/02/2012	Doug	Clear up where the new resource appears in a Job to resolve issue 1714. Add some clarifying text around collections to resolve issue 1715. Move macAddress fro m Address resource to Machine.networkInterface to resolve issue 1672.
0.0.86	05/09/2012	Doug	s/rootURI/baseURI/ to resolve issue 1735. s/MeterConfiguration/MeterConfig/ for attribute names to resolve issue 1732. For attributes like memory and capacity covert it from a structure to a single integer to resolve issue 1734.
0.0.87	05/22/2012	Doug	Per cmwg msg 201205/msg00053.html reordered some stuff in System and CEP to align with the TOC.

Version	Date	Who	Description
0.0.88	05/23/2012	Doug	s/Credentials/Credential/g to resolve issue 1748. Allow for non-snapshot images in machine.restore to resolve issue 1737. Allow for config type of data to be passed by-value on create to resolve issue 1733.
0.0.89	05/30/2012	Doug	Tweak our serialization text to resolve issue 1687.
0.0.90	06/01/2012	Doug	Update UML diagrams to resolve issue 1652.
0.0.91	06/06/2012	Doug	Allow for partial responses in the 202 cases to resolve issue 1750. Inline "Events" into the EventLog.events collection to resolve issue 1761. Make "meters" and "eventLogs" owned resources to resolve issue 1757. Inline some collections to resolve issue 1749.
0.0.92	06/13/2012	Doug	Add resource name to capability URIs to resolve issue 1766. Remove CIMI from our query parameters to resolve issue 1767.
0.0.93	06/19/2012	Doug	Add definition of Cloud to resolve issue 1629. Allow CIMIFilter to operate over properties to resolve issue 1768. Make samples a collection to resolve issue 1774.
0.0.94	06/19/2012	Doug	Update security section(s) to resolve issue 1731.
0.0.95	06/20/2012	Doug	s/operation/action/ to resolve issue 1782.
0.0.96	06/21/2012	Doug	Clarify what to do with bad $first/last parameters to resolve issue 1613. Clarify what not $expand w/o attributeNames means to resolve issue 1614. Clarify some stuff around "name" to resolve issue 1747.
0.0.97	07/11/2012	Doug	Added "disabled" state to networkinterfaces to resolve issue. Add 'cpuSpeed' to Machine and MachineConfiguration to resolve issue 1645.
0.0.98	08/01/2012	Doug	Backed out resolution to issue 1645 since its not clear yet how to add it to the spec. s/timeStamp/timestamp/ to resolve issue 1803.
0.0.99	08/29/2012	Doug	Updated UML diagrams to resolve issue 1853. Remove version header to resolve issue 1727.
0.0.100	09/05/2012	Doug	Made Address.dns an array to resolve issue 1622.
0.0.101	09/12/2012	Doug	Clarify MeterConfiguration.associatedTo to resolve issue 1867. Add cpuSpeed to resolve issue 1645.
0.0.102	09/19/2012	Doug	Deprecate some add() operations to resolve issue 1788.
0.0.103	10/02/2012	Doug	Add clarity around the use of special create XML wrappers to resolve issue 1896. Removed the add operation from the Job collection - typo - to resolve issue 1897. Add some clarifying text around opaque URIs to resolve issue 1868. Clarify that intermediary/relationship resources are deleted when their target resources are deleted to resolve issue 1890. Add a statement about ETag suport to resolve issue 1728. Add some clarifying text around System.state to resolve issue 1869.
0.0.104	10/05/2012	Doug	Clarify use of $expand and make our capabilities into more normative statements to resolve issue 1885.
0.0.105	10/17/2012	Doug	Add some text around partial updates to resolve issue 1904.
0.0.106	10/24/2012	Doug	Fix some Network.NetworkPort typos - mainly in the serializations to resolve issue 1915.
0.0.107	11/07/2012	Doug	Clarify text around initialState to resolve issue 1905. Add missing 'restore' action to Machine states to resolve issue 1920. Move some text around how to deal with unknown attributes to resolve issue 1923.
0.0.108	11/24/2012	Doug	Mention presence of resourceURI in XML serialization of Collections to resolve issue 1934.
0.0.109	12/03/2012	Doug	Clarify our use of the Accept header to resolve issue 1937. Add support for the $format query parameter to resolve issue 1924. Clarify how to convert relative to absolute URIs to resolve issue 1948. Remove old text about Job MIME types to resolve issue 1949. Be consistent about our definition of dateTime to resolve issue 1952.

Version	Date	Who	Description
			Clarify that resourceURI does appear even when $select is used to resolve issue 1953.
0.0.110	01/09/2013	Doug	Remove unused references to resolve issue 1966.
0.0.111	01/31/2013	Doug	Add some clarity around "value constraints" in ResourceMetadata to resolve issue 1971. Remove possible misleading text around Machine and Job state values being optional to resolve issue 1979.
0.0.112	02/13/2013	Marios	Change serialization of “latestSnapshot” attribute of Machine resource to resolve issue 1985. Fix minor typo in Network resource's “classOfService” attribute description.
0.0.113	02/28/2013	Marios	Added $orderby query parameter (under section “Request query parameters”) and corresponding entry in ResourceMetadata Capabilities table (a CloudEntryPoint capability) to resolve issue 1976. Fixed various minor typos to resolve issue 1993.
0.0.114	03/14/2013	Marios	Added MachineAvailabilityLevel and VolumeAvailabilityLevel capabilities to ResourceMetadata capabilities table in 5.11.2 to resolve issue 1986. Changes to descriptions of MachineTemplate.Volumes, MachineTemplate.volumeTemplates, MachineTemplate.networkInterfaces and MachineConfiguration.disks to resolve issue 1992.
0.0.115	04/11/2013	Marios	Various changes/clarifications for states and state changing operations of all resources and clarified rules around provider defined operations and states, to resolve mantis issue 1974. Added mechanism for discovering or specifying the initial state of the Network and NetworkPort resources, to resolve mantis issue 2012.
0.0.116	04/24/2013	Marios	Mark the name attribute in section 5.10.1 "Common Attributes" as Provider support optional, to resolve mantis issue 1958.
0.0.117	05/02/2013	Marios	Add 'importImage' attribute to SystemTemplate resource to resolve mantis issue 1957.
0.0.118	05/16/2013	Marios	Editorial changes to the componentTemplate sub-attribute of the componentDescriptor attribute of the SystemTemplate resource, to resolve mantis issue 2013. Editorial cleanup/re-write of section 4.2.1.1 “Creating a new resource” to resolve mantis issue 2032.
0.0.119	06/11/2013	Jacques	Editorial changes to resolve mantis issue 2050.
0.0.120	06/17/2013	Jacques	Editorial changes to resolve mantis issue 2049.
0.0.121	07/22/2013	Jacques	Editorial changes to resolve mantis issue 2050, 2078, 2093, 2094..
1.1.0a	08/13/2013		Release as Work in Progress

6291