Document Number: DSP0263

Date: 2012-04-05

Version: 1.0.0d

Work Group Version: 0.0.76

Cloud Infrastructure Management Interface (CIMI) Model and REST Interface over HTTP

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: 2012-10-15

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

Document Type: Specification

Document Status: Work In Progress - not a DMTF Standard

Document Language: en-US

CONTENTS

170

FIGURES

Forward

The *Cloud Infrastructure Management Interface (CIMI) Model and REST Interface over HTTP* 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. This model was developed to address the use cases outlined in the "Scoping Framework for Cloud Management Models and Protocol Requirements" document.

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

Acknowledgments

The authors wish to acknowledge the following people.

Editors:

- Davis, Doug - IBM
- Pilz, Gilbert - Oracle

Contributors:

- Ali, Ghazanfar - ZTE Corporation
- Andreou, Marios - Red Hat
- 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
- 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.
- Gopalan, Krishnan - Microsoft Corporation
- Iwasa, Kazunori - Fujitsu
- Johnson, Mark - IBM
- Kowalski, Vincent - BMC Software
- Krishnaswamy, Ruby - France Telecom Group
- Lamers, Lawrence - VMware Inc.
- Lipton, Paul - CA Technologies
- Livingston, James - NEC Corporation
- Lubsey, Vince - Virtustream Inc.
- Lutterkort, David - Red Hat
- Maciel, Fred - Hitachi, Ltd.
- Maier, Andreas - IBM
- Malhotra, Ashok - Oracle

- Mischkinsky, Jeff - Oracle
- Molina, Jesus - Fujitsu
- Moscovich, Efraim - CA Technologies
- Murray, Bryan - Hewlett-Packard Company
- Neely, Steven - Cisco
- Ogawa, Ryuichi - NEC Corporation
- Parchem, John - Microsoft Corporation
- Pardikar, Shishir - Citrix Systems Inc.
- Pe164alvo, Miguel - Telef162nica
- Pilz, Gilbert - Oracle
- Polo, Alvaro - Telef162nica
- Ronco, Enrico - Telecom Italia
- Rossini, Federico - Telecom Italia
- Rutkowski, Matthew - IBM
- Rutt, Tom - Fujitsu
- Shah, Hemal - Broadcom
- Shah, Nihar - Microsoft Corporation
- Sill, Alan - Texas Tech University
- Song, Zhexuan - Huawei
- Song, Zhexuan - Fujitsu
- Waschke, Marvin - CA Technologies
- Wells, Eric - Hitachi, Ltd.
- Wheeler, Jeff - Huawei
- Wiggers, Maarten - Fujitsu
- Winkler, Steve - SAP AG
- Yu, Jack - Oracle
- Zhang, Aaron - Huawei
- Zhang, HengLiang - Huawei

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 entities 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, although the model is designed to be protocol independent and other protocols are possible. 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, these uses are outside the design goals of CIMI.

1.1 Document Structure

This document defines a model and an HTTP/REST-based protocol.

The core REST pattern is defined first and, after each entity is defined, any REST-specific information for that entity will be specified.

1.2 Document Versioning Scheme

This document will adhere to the versioning scheme defined in [**DSP-4004**] section 6.3.

1.3 Typographical Conventions

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

- Entity 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>)

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)
    - "+" (1 or more)
- Ellipses (i.e., "...") indicate points of extensibility. Note that the lack of an ellipses does not mean no extensibility point exists, rather it is just not explicitly called out - usually for the sake of brevity.

2 References

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies:

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

IETF RFC 2045, N. Freed et al, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, November 1996, http://www.ietf.org/rfc/rfc2045.txt

IETF RFC 2616, R. Fielding et al, *Hypertext Transfer Protocol -- HTTP/1.1*, June 1999, http://www.ietf.org/rfc/rfc2616.txt

IETF RFC 2617, J. Franks et al, *HTTP Authentication: Basic and Digest Access Authentication*, June 1999, http://www.ietf.org/rfc/rfc2617.txt

IETF RFC 2246, T. Dierks and C. Allen, *The TLS Protocol Version 1.0*, Janury 1999, http://www.ietf.org/rfc/rfc2246.txt

IETF RFC 3986, T.Berners-Lee et al, *Uniform Resource Identifiers (URI): Generic Syntax*, August 1998, http://www.ietf.org/rfc/rfc3986.txt

IETF RFC 4346, T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.1*, April 2006, http://www.ietf.org/rfc/rfc4346.txt

IETF RFC 4627, D. Crockford, *The application/json Media Type for JavaScript Object Notation (JSON)*, July 2006, http://www.ietf.org/rfc/rfc4627.txt

IETF RFC 5246, T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.1*, http://www.ietf.org/rfc/rfc5246.txt

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

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

ITU-T X.509, Telecommunication Standardization Sector of ITU, *Information technology - Open Systems Interconnection - The Directory: Public- key and attribute certificate frameworks*, November 2008, http://www.itu.int/rec/T-REC-X.509-200811-I

NIST Special Publication 800-57, Elaine Barker et al, *Recommendation for Key Management – Part 1: General (Revised)*, March 2007, http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57-Part1-revised2_Mar08-2007.pdf

NIST Special Publication 800-131A, Elaine Barker and Allen Roginsky, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, January 2011, http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.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/

DMTF DSP-0243, Distributed Management Task Force, Inc., *Open Virtualization Format Specification 1.1.0*, http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf

DMTF DSP-0259, Distributed Management Task Force, Inc., *Cloud Infrastructure Management Interface - CIM Model (CIMI-CIM) 0.0.1*, http://members.dmtf.org/apps/org/workgroup/cmwg/download.php/yyyy

DMTF DSP-4004, Distributed Management Task Force, Inc., DMTF Release Process 2.4.0, http://www.dmtf.org/sites/default/files/standards/documents/DSP4004_2.4.0.pdf

DMTF DSP-ISXXXX, Distributed Management Task Force, Inc., *Scoping Framework for Cloud Management Models and Protocol Requirements 0.1.5*, http://members.dmtf.org/apps/org/workgroup/cmwg/download.php/56339/Cloud%20Management%20Framework_v015.doc

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.

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 involved 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 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, purchase services within the business relationship; create Service Users (including policies), allocate resources, such as compute and storage, generate reports (usage), etc.), and Service Users (who uses 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.

3.4 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.

3.5 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.

3.6 Message Confidentiality

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

3.7 Message Integrity

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

3.8 Template

A Template is the entity 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 entities 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 an entity 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 an entity in the Provider) and provide additional values that override the attributes of that Template as part of the instantiation operation.

4 REST/HTTP Protocol

4.1 Protocol Definition

All operations are based on the HyperText Transfer Protocol, 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, overloaded with semantics by the context of the particular request that was made. Each entity in the model has a MIME standard ContentType that further contextualizes the operation requests and responses.

The entities in the model are identified by URIs and each entity's representation MUST 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 entity will yield a representation of the entity containing attributes and links to associated entities. To begin operations, a client must know the URI to the main entry point of a Provider - also known as the "Cloud Entry Point" entity. All other entities within the environment shall then be discoverable via the iterative following of links to associated resource within each resource retrieved.

4.1.1 Protocol Security

Providers SHALL support secure HTTP connections using TLS. Providers MAY support non-secure HTTP connections. TLS 1.0, which shall be implemented, is specified in [RFC2246], and the TLS 1.1 and TLS 1.2 should be implemented as specified in [RFC4346] and [RFC5246], respectively.

To ensure a minimum level of security and interoperability between implementations, for the following TLS versions, all CIMI clients and servers shall support the following mandatory cipher suites.For TLS 1.0, the mandatory ciphersuite is TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (see [RFC2246] Section 9, Mandatory Cipher Suites). For TLS 1.1, the mandatory ciphersuite is TLS_RSA_WITH_3DES_EDE_CBC_SHA (see [RFC4346] Section 9, Mandatory Cipher Suites). For TLS 1.2, the mandatory ciphersuite is TLS_RSA_WITH_AES_128_CBC_SHA (see [RFC5246] Section 9, Mandatory Cipher Suites).

The TLS_RSA_WITH_AES_128_CBC_SHA256 cipher suite should be included with all recommended TLS 1.2 implementations to meet the transition to a secure hash function (guidance is provided in NIST Special Publication 800-57 [NIST 800-57] and NIST Special Publication 800-131A [NIST 800-131A]). Implementers are free to include additional cipher suites, but must prefer the mandatory ones in negotiation.

4.1.2 Protocol Authentication

There are many authentication protocols that can be used with HTTP/1.1(e.g. Basic and Digest authentication as defined by [RFC 2617]). This specification does not mandate any particular mechanism, however, for interoperability the Provider shall provide enough information such that Consumer can discover and implement the necessary algorithms to successfully communicate with the Provider.

4.1.3 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://www.dmtf.org/cimi	This specification
xs	http://www.w3.org/2001/XMLSchema	XML Schema **XMLSchema - Part 2**

4.1.4 URI Space

While URIs returned by Providers are to be treated as opaque by Consumers, and Consumers MUST NOT make assumptions about the layout of the URIs or the structures of the URIs of the resources, Consumer may augment URIs with any well-defined query parameters which are supported by the Provider as defined in section 4.1.7. Providers shall not use the CIMI-defined query parameter reserved namespace (i.e. names starting with "CIMI").

4.1.5 Media Types

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

In the JSON serialization of CIMI resources sent by Providers there shall be an additional attribute called "entityURI" that will contain the unique URI associated with the type of CIMI entity being serialized. This attribute is optional for Consumers to include. When included, this attribute's value shall match the "typeURI" attribute of the corresponding EntityMetadata resource - if EntityMetadata 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.

The server implementation shall provide representations of all resources available in both JSON and XML as specified herein. The client implementation may thus use either JSON or XML to communicate with any server implementation.

4.1.6 Request Headers

This specification uses general-header, request-header, and entity-header headers as defined in HTTP 1.1 [RFC2616] in request messages to provide metadata about the message. Applications using messages defined in this specification shall use headers consistent with HTTP 1.1.

In addition to headers defined in HTTP 1.1, request messages may include a header defined by this specification to indicate the set of allowable versions of the CIMI API that server shall use to process the message.

```
X-CIMI-Specification-Version = "X-CIMI-Specification-Version" ":" api-version(s)
```

For example:

```
X-CIMI-Specification-Version=1.0
```

The header allows for a list of *api-version* values to be specified (separated by commas) and to be presented in descending order according to the client's preference. When more than one value is present the server shall choose the preferred one from those versions of the specification to process the message. Clients including more than one value are indicating that any of the specified values are acceptable.

Per [**DSP-4004**], the "api-version" string is made up of 3 parts: m.n.u - major.minor.update. When present in this header, it shall include at least the major and minor (m.n) version numbers. It may also include the "update" portion of the version if necessary. Absence of the "update" portion of the "api-version" string implies that any "update" version of that major.minor version of the specification is acceptable to the client.

If the server is unable to support any of the specified versions then it shall generate a fault and not process the message. Absence of this header indicates that the server may choose any version of this specification to process the message.

4.1.7 Request Query Parameters

To modify the behavior of the Provider when processing request messages, Consumers may augment request URIs with the following query parameters:

4.1.7.1 CIMISelect

The CIMISelect query parameter may be used to specify a subset of a resource to be acted upon. This has the semantic equivalence of referencing a different resource whose attributes are a subset of the original resource. The format of a CIMISelect query parameter is:

```
?CIMISelect=attrName,*
```

The value of the CIMISelect 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 shall be ignored by the Provider. An attribute name of "*" is equivalent to specifying all of the attributes of the resource. Any attribute name explicitly appearing more than once in a URI shall have its second (and subsequent) appearances ignored.

The CIMISelect 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 CIMISelect query parameter. For example:

```
?CIMISelect=name&CIMISelect=state
```

is equivalent to:

```
?CIMISelect=name,state
```

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

Attribute names listed in a CIMISelect value may also include further qualifiers depending on the type of attribute as follows:

Arrays

This specification allows for the Consumer to subset the list of entries in an array by including a bracketed expression after the attribute name:

```
?CIMISelect=attrName[expression]
```

Where "expression" can be one of the following:

- index (a single number)
  If the "expression" is a single number then it is to be treated as an index into the array and only that one entry in the array is referenced. Arrays are one-based - i.e. an index of "1" references the first entry in the array. An index value that is beyond the number of entries in the array results in an empty reference.

- range (low-high)
  If the "expression" is expressed as a range then only the entries in that range are referenced. As with the "index" expression, arrays are one-based and any range value(s) outside of the number of entries in the array results in an empty reference.

- value expression
  The "expression" may be a syntactical expression used to select the set of array entries based on the value of the attributes in those entries.

  The "expression" may use the following mathematical constructs: "=", "<", ">", "<=", ">=", "<>", "and", "or", "(" and ")". Each of these shall be percent encoded in the URL as appropriate.

  The value may contain literals that can be strings, numbers, dates, boolean values (true or false) and null. Strings are delimited by single quotes.

For example, an expression of:
`name='mine'`
indicates that the array entry is included in the referenced set if that entry's "name" attribute has a value of "mine".

Normally the "attribute name" portion of an expression is just a single word, the attribute name itself, however, if the attribute is a nested resource, or a reference, then a dot (.) may be included in the attribute name as a way to represent a traversal into that secondary resource.

If this qualifier appears on an attribute name, and that attribute is not an array, then the Provider shall ignore the qualifier.

Examples (the URI of "/machines" references a Machines Collection resource):

```
GET /machines?CIMISelect=operations
```

This returns just the 'operations' attribute of the Machines Collection resource.

```
GET /machines?CIMISelect=machines[1]
```

This returns just the 'machines' attribute of the Machines Collection resource, but just the first machine in the array.

```
GET /machines?CIMISelect=machines[1-100]
```

This returns just the 'machines' attribute of the Machines Collection resource, but just the first 100 machines in the array.

```
GET /machines?CIMISelect=machines[name='mine']
```

This returns just the 'machines' attribute of the Machines Collection resource, but only machines whose 'name' attribute has a value of 'mine' will be included in the array.

```
GET /machines?CIMISelect=*,machines[name='mine']
```

This returns all of the attributes of the Machines Collection resource, but the machines array is subset to just the machines whose 'name' attribute has a value of 'mine'.

```
GET /machines?CIMISelect=machines[memory.quantity>64 and memory.units='byte']
```

This returns just the 'machines' attribute of the Machines Collection resource, but only machines that have more than 64 bytes of memory will be included in the array.

4.1.7.2 CIMIExpand

The CIMIExpand query parameter may be used to specify which of the references should be "expanded". To "expand" a reference means that the attributes of the resource being referenced are to be included in the serialization of that attribute. This allows for a more optimized retrieval of resources.

The format of a CIMIExpand query parameter is:

```
?CIMIExpand=attrName,*
```

The value of the CIMIExpand 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 "*" is equivalent of 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 CIMIExpand 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 CIMIExpand query parameter. For example:

```
?CIMIExpand=foo1&CIMIExpand=foo2
```

is equivalent to:

```
?CIMIExpand=foo1,foo2
```

Example:

```
GET /machines?CIMIExpand=entries
```

Would result in something similar to this:

```
<Collection entityURI="http://www.dmtf.org/cimi/MachineCollection"
            xmlns="http://www.dmtf.org/cimi">
  <id> http://example.com/machines </id>
  <Entry entityURI="http://www.dmtf.org/cimi/MachineCollectionEntry">
    <id> http://example.com/machinesCollection/1234 </id>
    <machine href="http://example.com/machines/1234">
      <id> http://example.com/machines/1234 </id>
      <name> mine </name>
      <state> STARTED </state>
      ...
    </machine>
  </Entry>
  <Entry>
    ...
  </Entry>
  <operation rel="add" href="http://example.com/machines"/>
</Collection>
```

And in JSON it would be:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineCollection",
  "id": "http://example.com/machines",
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/MachineCollectionEntry",
      "id": "http://example.com/machinesCollection/1234",
      "machine": { "href": "http://example.com/machines/1234",
        "entityURI": "http://www.dmtf.org/cimi/Machine",
        "id": "http://example.com/machines/1234",
        "name": "mine",
        "state": "STARTED",
        ...
      }
    },
    { "id", ...
    }
    ...
  ],
  "operations": [ { "rel": "add", "href", string } ? ]
}
```

4.1.8 Response Headers

This specification uses general-header, response-header, and entity-header headers as defined in HTTP 1.1 [RFC2616] in response messages to provide metadata about the message. Applications using messages defined in this specification shall use headers consistent with HTTP 1.1.

In addition to headers defined in HTTP 1.1, response messages shall include a header defined by this specification to indicate the version of the CIMI API that the server used to process the message.

```
   X-CIMI-Specification-Version = "X-CIMI-Specification-Version" ":" api-version
```

See section 4.1.6 for more details on this header.

Additionally, if the server supports the Job entity then response messages shall include a header defined by this specification to indicate the URI for the job created to process the associated request message.

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

In cases where an error occurrs during the processing of a request, the Provider shall include a representation of a Job entity describing the status of the failed operation. This shall be included even in cases where the Provider does not normally support Job entities. This is done to ensure that Consumers are provided with sufficient information, in a consistent manner, as to the reason for the failure regardless of whether the Provider supports Jobs or not. When Jobs are not supported in general, any of the references in the Job representation (e.g. "id" or the "href" for nestedJobs) shall be empty paths (i.e. "") and the "nestedJobs" array shall be expanded (see 4.1.7.2) to inline the representation of the pseudo subordinate Jobs.

4.1.9 HTTP Status Codes

Server implementations will return standard HTTP response codes as described in the following table, under the conditions listed in the description.

Editors Note: These are changes from the basic HTTP semantics that are overloaded for the whole specification. We will remove any status codes that are standard HTTP without overloading.

Table: HTTP Status Codes

HTTP Status	Description
100 Continue	The client SHOULD continue with its request. This interim response is used to inform the client that the initial part of the request has been received and has not yet been rejected by the platform. The client SHOULD continue by sending the remainder of the request or, if the request has already been completed, ignore this response.
200 OK	The request was successfully completed. If this request created a new resource that is addressable with a URI, and a response body is returned containing a representation of the new resource, a 200 status will be returned with a Location header containing the canonical URI for the newly created resource
201 Created	A request that created a new resource was completed, and no response body containing a representation of the new resource is being returned. A Location header containing the canonical URI for the newly created resource will be returned. Per the HTTP/1.1 specification: *The origin server MUST create the resource before returning the 201 status code. If the action cannot be carried out immediately, the server SHOULD respond with 202 (Accepted) response instead.*
202 Accepted	The request has been accepted for processing, but the processing has not been completed. Per the HTTP/1.1 specification, the returned entity (if any) SHOULD include an indication of the request's current status. A Location header containing the canonical URI for the not-yet completed resource would be returned along with the Status attribute indicating its progress. If a service implementing this specification supports the Job entity then it SHOULD return the representation of a Job entity in the HTTP body of the response and shall include a 'X-CIMI-Job-URI' HTTP header indicating the URI of the Job entity itself.

	Per the HTTP/1.1 specification: *The entity returned with this response SHOULD include an indication of the request's current status and either a pointer to a status monitor or some estimate of when the user can expect the request to be fulfilled*
400 Bad Request	The request could not be processed because it contains missing or invalid information (such as validation error on an input field, a missing required value, and so on)
401 Unauthorized	The authentication credentials (TBD) included with this request are missing or invalid
403 Forbidden	The server recognized your credentials, but you do not possess authorization to perform this request
404 Not Found	The request specified a URI of a resource that does not exist
405 Method Not Allowed	The HTTP verb specified in the request (DELETE, GET, HEAD, POST, PUT) is not supported for this request URI.
406 Not Acceptable	The resource identified by this request is not capable of generating a representation corresponding to one of the media types in the Accept header of the request
409 Conflict	A creation or update request could not be completed, because it would cause a conflict in the current state of the resources supported by the platform. This is used in MachineTemplate create/update to indicate that the MachineConfiguration cannot support the given MachineImage, for example.
410 Gone	The requested resource is no longer available at the server and no forwarding address is known. This condition is expected to be considered permanent. Clients with link editing capabilities SHOULD delete references to the Request-URI after user approval. If the server does not know, or has no facility to determine, whether or not the condition is permanent, the status code 404 (Not Found) SHOULD be used instead. This response is cacheable unless indicated otherwise
412 Precondition Failed	The precondition given in one or more of the request-header fields evaluated to false when it was tested on the server. This response code allows the client to place preconditions on the current resource meta-information (header field data) and thus prevent the requested method from being applied to a resource other than the one intended
500 Internal Server Error	The server encountered an unexpected condition which prevented it from fulfilling the request
501 Not Implemented	The server does not support the functionality required to fulfill the request
503 Service Unavailable	The server is currently unable to handle the request due to temporary overloading or maintenance of the server

4.2 Entity Serialization

This section details how the entities in the model are serialized in JSON and XML. This serialization follows rules that are expected to be complied with when serializing extensions to the model. This

serialization is largely independent from the protocol in use. For those aspects that may vary depending on the protocol, it will be indicated so and the serialization will be defined for REST/HTTP,

4.2.1 Entity wrappers

The serialization of entity instance in the model will follow these conventions:

JSON serialization:

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

```
{
  "entityURI": "http://www.dmtf.cimi/Entity",
  "attribute": "value"
}
```

XML serialization:

The entity is serialized as an element with name equal to the Entity name, for example:

```
<Entity>
  <attribute> value </attribute>
</Entity>
```

4.2.2 Basic Types

The following describes how each basic attribute type in the model translates into JSON and XML types.

Data Model Type	XML type	JSON type
string	*xs:string*	*string*
URI	*xs:anyURI*	*string*
integer	*xs:integer*	*number*
boolean	*xs:boolean*	*boolean*

References to entity instances, or to collections, are indicated by the type 'ref' in the model, and are represented as URIs in the REST/HTTP protocol mapping.

As a general rule, when an entity or structure attribute is of type "ref", its value will 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 entity or structure attribute is of type "ref", the name of this attribute will appear as a key, with the "href" property as its value. For example, an entity attribute "myvolume" of type "ref" is serialized as:

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

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

```
"mythings": [ { "href": string },  + ]
```

XML serialization:

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

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

In 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 collection of XML elements without wrapper. Each element is named after the type of entity being referenced (converted so that the first letter is in lower case character). For example, an array "mythings" of type ref[] where each reference points to a Thing instance, is serialized as

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

Extensibility:

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 entity to be included if desired. So, more accurately the above representations should be written as follows:

For JSON:

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

and in XML:

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

however, for brevity they are excluded.

4.2.3 Serialization of Nested Structures

A nested structure can be considered a complex type definition (see section 5.5). Structures may be named (e.g. "networkInterface") or not (e.g. a small structure as used to represent a "memory" attribute that has two fields: units and quantity).

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 "memory" where the type of "memory" is a structure with two sub-attriibutes ("quantity" and "units") is serialized as follows:

```
  "memory": { "quantity": number, "units": string }
```

XML serialization:

In XML, the name of a structure will show as an XML element. In other words, the name of the type represented by this structure serves as an XML wrapper. In case the structure is un-named, the name of the attribute qualified by this structure, is used as XML wrapper. The same previous "memory" example (the type of which is an un-named structure) will be serialized so that the structure sub-attributes ("quantity" and "units") become XML attributes of a <memory> XML element wrapper:

```
  <memory quantity="xs:integer" units="xs:string"/>
```

NOTE: In case there is a large number of sub-attributes of atomic type in a structure, these may be represented alternatively as XML child elements, for better readability. Both options are available, however the same structure must be serialized the same way across entities.

4.2.4 Serialization of Arrays

Arrays are represented as an attribute of type ending with "[]". This section does not concern arrays of references (ref[]) already treated in 4.2.3 due to the peculiar serialization of reference types.

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

NOTE: 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 is done to ensure that the JSON serialization is minimized and only includes the wrapping "disks" element if, and only if, there are disks.

XML serialization:

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

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

There is no wrapper element for an array.

4.3 Protocol Resource Operations

This section defines the set of common REST/HTTP operations that a Provider might expose. At its core there are four basic CRUD (Create, Read, Update and Delete) operations. The manner in which these are used is consistent across all resources within the model; therefore, their use is defined once and is to be applied consistently. Some resources support specialized operations that do not fit well into a CRUD style 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 section that defines the resource.

When appropriate some of the resource representations will include "operation" properties. These either provide URI references that can be used to perform operations on the resource, or they are URI references to other resources that are related to the current resource. Providers shall only include "operation" properties when the specified operation or related resource is accessible to the current client for that particular resource. This means that based on many factors (e.g. authorization rights of the clients, current state of the resource, etc.) a different set of "operation" properties might be returned on each serialization of the resource.

4.3.1 Operational Principles

4.3.1.1 Resource Navigation

The retrieval of the representation of a Resource using (GET <ResourceURI>) shall return the attributes of the resource; these attributes might include a set of references to related resources. In that case, it is

possible to obtain every related resource by repeatedly applying the GET method on the retrieved references.

Example:

If a resourceX contains an attribute "attrA" of string type and an attribute "attrB", where the latter references resourceY, the operation:

```
GET <ResourceURI_X> HTTP/1.1
Host: ...
Accept: application/...
X-CIMI-Specification-Version: 1.0
```

returns a message containing the following:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Resource_X",
  "attrA": "hello",
  "attrB": { "href": "http://example.com/uriB" }
}
```

XML serialization:

```
<Resource_X xmlns="http://www.dmtf.org/cimi">
  <attrA> hello </attrA>
  <attrB href="http://example.com/uriB" />
</Resource_X>
```

Following the attrB uri ("uriB"), the operation:

```
GET <uriB> HTTP/1.1
Host: ...
Accept: application/...
X-CIMI-Specification-Version: 1.0
```

returns a message containing the following:

JSON serialization:

```
{ "entityURI", "http://www.dmtf.org/cimi/Resource_Y",
  "attrY": "bye"
}
```

XML serialization:

```
<Resource_Y xmlns="http://www.dmtf.org/cimi">
  <attrY> bye </attrY>
</Resource_Y>
```

Notes:

1) It is possible that the retrieved reference obtained with the GET <ResourceURI> operation does not directly refer to a related resource, but to a list of homogeneous related resources. In such case, the result of the GET operation will be a list of URIs, and thus the result of a further GET on one of such URIs will return the related resource.

2) this makes it possible to navigate the CIMI resource hierarchy with just the knowledge of the root URI of the Provider.

4.3.1.2 Operations on a Resource

When it is possible to execute specific operations on the resource, then the response to the GET method on the resource URI shall contain information to perform such operations, in particular:

1. The operation name (typology) such as add, delete, edit, start, stop are described using the rel attribute in the "operation" element.
2. The URI to perform the above mentioned operation.

It is possible to understand that such a field isn't a link to a related resource but instead is an operation by the presence of the attribute "operations" in JSON or the element "operation" in XML.

The operation shall be performed by invoking the REST specific function on the specified URI.

Example:

If for the resource X an operation is provided to edit that resource, the Get <ResourceURI> response message will contain the following:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Resource_X",
  "operations": [
    { "rel": "edit", "href": "editURI" }
  ]
}
```

XML serialization:

```
<Resource_X xmlns="http://www.dmtf.org/cimi">
  <operation rel="edit" href="editURI"/>
</Resource_X>
```

In this example the operation will be performed with the HTTP PUT on editURI, as follows:

```
PUT <editURI> HTTP/1.1
Host: ...
Accept: application/...
Content-Type: application/...
X-CIMI-Specification-Version: 1.0

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

4.3.2 Common CRUD (Create Read Update and Delete) Operations

Each of the resources supported by this protocol will adhere to the interaction patterns defined in the following sections. Section 4.4 then defines resource specific information such as the serialization of each resource's properties and which specific actions are supported.

4.3.2.1 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 which references the "addURI" that is to be used.

The request will be of the following form:

```
POST <addURI> HTTP/1.1
Host: ...
Accept: application/...
Content-Type: application/...
```

```
X-CIMI-Specification-Version: 1.0

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

The following provides additional constraints on the request message:

X-CIMI-Specification-Version

This optional HTTP header specifies the list of versions of this specification that the server shall choose from to process this message.

During the process of creating the resource, depending on the resource type, the Provider may set the state of the new resource to a value of "CREATING".

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:

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

Creating a new Machine can be done by including a reference to a MachineTemplate in the HTTP body of the request message, or the individual attributes of the MachineTemplate itself could be included in the message (as denoted by the "*template-values*" text in the above example). And the same applies for nested attributes. When the information is passed by-value the Provider may choose to create instances of those nested resources but they shall be temporal in nature. This means that, the Provider shall not expose those instances to the Consumer and they shall not be included in any query results back to the Consumer.

Additionally, Consumers shall not include both the reference to a Template and the inlined attributes of the Template within the same request message.

When the request to create a new resource allows for a reference to a Template to be included, Consumer may include some of the Template's attributes "by-value". In this case the Provider shall use the "by-value" attributes as override values of any attributes specified within the referenced Template. 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. This overriding mechanism shall only be used on immediate top-level attributes of the Template, and shall not be used to override any sub-attributes.

Note that the "name" and "description" properties of the Template should not be included when passing the Template attributes by-value. This is because those values are defining the name and description of the Template, not of the new resource being created, and since the Template is never persisted within the Provider including these attributes would serve no purpose.

The response will be of the following form:

```
HTTP/1.1 201 Created
Location: ...
Content-Type: application/...
X-CIMI-Specification-Version: 1.0
```

```
<serialization of new resource>
```

The following provides additional constraints on the response message:

X-CIMI-Specification-Version

This REQUIRED HTTP header specifies the version of this specification that was used to process this message.

The HTTP response will also include a status code, as described in the following table:

HTTP Status	Description
201 Created	The new resource was created
202 Accepted	The resource is in the process of being created. Investigate Job to determine the current status of the operation.
400 Bad Request	Invalid parameter or field names in the request.
401 Unauthenticated	Incorrect or missing authentication credentials.
403 Unauthorized	Client lacks the proper authorization to perform this request.

4.3.2.2 Reading a Resource

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

The request will be of the following form:

```
GET <ResourceURI> HTTP/1.1
Host: ...
Accept: application/...
X-CIMI-Specification-Version: 1.0
```

The following provides additional constraints on the request message:

X-CIMI-Specification-Version

This optional HTTP header specifies the list of versions of this specification that the server shall choose from to process this message.

The response will be of the following form:

```
HTTP/1.1 200 OK
Content-Type: application/...
X-CIMI-Specification-Version: 1.0

<serialization of resource>
```

The following provides additional constraints on the response message:

X-CIMI-Specification-Version

This REQUIRED HTTP header specifies the version of this specification that was used to process this message.

The HTTP response will also include a status code, as described in the following table:

HTTP Status	Description
401 Unauthenticated	Incorrect or missing authentication credentials.

403 Unauthorized	Client lacks the proper authorization to perform this request.

4.3.2.3 Updating a Resource

To update the representation of a resource, an HTTP PUT request is sent to a designated "editURI" for that resource type. In many cases, this "editURI" will be the same as the URI of resource itself - retrieving the resource representation MUST 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 then it SHALL silently ignore that attribute update request and SHALL NOT generate an error. This applies to resource partial updates as well.

Due to potential conflicts that might occur due to multiple concurrent updates, Consumers should use the partial update mechanism, defined in 4.3.2.3.1, to reduce the chances of mistakenly updating attributes with out-of-date data.

The request will be of the following form:

```
PUT <editURI> HTTP/1.1
Host: ...
Accept: application/...
Content-Type: application/...
X-CIMI-Specification-Version: 1.0

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

The following provides additional constraints on the request message:

X-CIMI-Specification-Version

This optional HTTP header specifies the list of versions of this specification that the server shall choose from to process this message.

The response will be of the following form:

```
HTTP/1.1 200 OK
Content-Type: application/...
X-CIMI-Specification-Version: 1.0

<serialization of updated resource>
```

The following provides additional constraints on the response message:

X-CIMI-Specification-Version

This REQUIRED HTTP header specifies the version of this specification that was used to process this message.

The HTTP response message body shall include the updated version of the resource representation.

The HTTP response will also include a status code, as described in the following table:

HTTP Status	Description
202 Accepted	The resource is in the process of being updated. Investigate Job to determine the current status of the operation.
400 Bad Request	Invalid parameter or field names in the request.

401 Unauthenticated	Incorrect or missing authentication credentials.
403 Unauthorized	Client lacks the proper authorization to perform this request.

4.3.2.3.1 Partial Updates to a Resource

To update only certain top-level attributes of a resource a Consumer MAY do so by including only the changes attributes in the representation of the resource within the HTTP request body. When this is done the URI to the resource SHALL include the attributes to be modified as a comma separated list of query parameters - in other words the URI will be of the form:

```
http://example.com/resource?CIMISelect=attribute1,attribute2,...
```

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

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

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

For example, the following request will update just the name and description attributes of a Machine:

```
PUT /machines/myMachine?CIMISelect=name,description HTTP/1.1
Host: ...
Accept: application/xml
Content-Type: application/xml
X-CIMI-Specification-Version: 1.0

<Machine>
  <name>My New Machine</name>
</Machine>
```

In this example, the "name" attribute is set to "My New Machine" and the "description" attribute is erased.

4.3.2.4 Deleting a Resource

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

The request will be of the following form:

```
DELETE <deleteURI> HTTP/1.1
Host: ...
X-CIMI-Specification-Version: 1.0
```

The following provides additional constraints on the request message:

X-CIMI-Specification-Version

This optional HTTP header specifies the list of versions of this specification that the server shall choose from to process this message.

During the process of deleting the resource, depending on the resource type, the Provider may set the state of the resource to a value of "DELETING".

The response will be of the following form:

```
HTTP/1.1 200 OK
X-CIMI-Specification-Version: 1.0
```

The following provides additional constraints on the response message:

X-CIMI-Specification-Version

This REQUIRED HTTP header specifies the version of this specification that was used to process this message.

The HTTP response will also include a status code, as described in the following table:

HTTP Status	Description
202 Accepted	The resource is in the process of being deleted. Investigate Job to determine the current status of the operation.
400 Bad Request	Invalid parameter or field names in the request.
401 Unauthenticated	Incorrect or missing authentication credentials.
403 Unauthorized	Client lacks the proper authorization to perform this request.

4.3.2.5 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 will 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 will be of the following form:

```
POST <operationLinkURI> HTTP/1.1
Host: ...
Accept: application/...
Content-Type: application/...
X-CIMI-Specification-Version: 1.0

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

The following provides additional constraints on the request message:

X-CIMI-Specification-Version

This optional HTTP header specifies the list of versions of this specification that the server shall choose from to process this message.

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 section 4.3.2.1) follows this same pattern - it is just called out for ease of reference.

4.3.2.6 Synchronous Operations

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

```
X-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, then the server SHALL return a 202 and follow the instructions for Asynchronous Operations.

4.3.2.7 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. For these cases, the Provider SHALL return an HTTP "202 Accepted" response code.

As with synchronous operations, if a Provider supports the Job entity then it SHALL create a Job entity for the incoming request and return a reference to that Job entity back to the client via the X-CIMI-Job-URI HTTP Header in the HTTP response message. Additionally, in the case of a "202 Accepted" response code and a Job URI being returned, the Provider MAY also return a representation of the Job entity in the body of the HTTP response message. If the request did not include the Job MIME type in the HTTP Accept header, then the encoding style (json vs xml) of the response SHOULD match the encoding style of the request message.

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

4.4 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 paramaters 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 it’s 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 section 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 section 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. This scope of this model is one separately administered cloud.

The CIMI model is described here 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 entities 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 REST/HTTP).

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

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

5.1 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 EntityMetadata resource for this purpose. EntityMetadata may be used to:

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

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

5.2 Identifiers

All identifiers (e.g. entity 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:
    - Upper case ASCII (U+0041 through U+005A)
    - Lower case 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)

5.3 Attribute Constraints

Each attribute of the entities 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 since each might differ. The following describes the possible "Constraints".

support optional:

This indicates that support for this attribute is optional. If supported, Providers should advertise its support via EntityMetadata. 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, Consumer 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.

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 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.

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 will 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 attribute 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 attribute 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 EntityMetadata.

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.

5.4 Simple Data Types

Unless specifically asked to not include certain attributes in the resource representation, the absence of an attribute in the representation means that the attribute has no value (i.e. is undefined); meaning there is no notion of an 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.

boolean:

A value as defined by xs:boolean per [**XMLSchema - Part 2**], with the exception that the only allowable values are either "true" or "false". The value is case sensitive.

dateTime:

A value as defined by xs:dateTime per [**XMLSchema - Part 2**]. 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.

duration:

A value as defined by xs:duration per [**XMLSchema - Part 2**]. 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.

integer:

A value as defined by xs:integer per [**XMLSchema - Part 2**]. 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.

string:

A value as defined by xs:string per [**XMLSchema - Part 2**]. Any constraints on this type 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.

ref:

A reference to another entity. The exact serialization of references will vary depending on the protocol being used. For example, for the REST/HTTP protocol, a "ref" will be serialized as a URI.

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.

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.

byte[]:

An arbitrary set of bytes meant to represent a block of binary data. Any constraints on this type 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.

URIs:

The format and syntax of the attributes of type "URI" is defined by RFC 3986 [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 parent of the CloudEntryPoint unless otherwise noted - in other words the "rootURI" is the parent of the CloudEntryPoint with a trailing slash.

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 root URIs:

Root 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 then the rootURI shall end with a trailing slash and relative URIs shall not begin with a leading slash. Doing this will be consistent with most URI resolve utilities and will produce the same results as a simple string concatenation algorithm.

Units:

Some of the entities defined by this specification have attributes that describe an amount of something that belongs to, or is associated with that entity. For example, the `Machine` entity has a `memory` attribute which describes "the size of the memory allocated to this machine". This specification adopts the convention of representing such attributes via a duple consisting of a `quantity` (represented as an integer) and `units` (represented as a string). The allowable values for `units` are listed in the following table. Their meaning is defined in IEC 80000-13:2008 [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

5.5 Complex Data Types

5.5.1 Structure Types

Some attributes of entities are of a complex data type that is represented as a nested structure – i.e. a table nested in the Entity table in this model representation. When such structures are named, this name is used as the name of the data type represented by the structure.

It may be the case that nested tables are not named, for simplicity. In such cases, the type name for the corresponding attribute is absent, or holds the notice "<unnamed structure>".

5.5.2 Arrays and Collections

This specification represents the grouping of similar typed items in two ways - as Arrays and Collections. The main difference between the two is that arrays represent the grouping within its owning resource as a list of items (referred to as "inlined"), while collections represent the grouping as a separately accessible resource in the model.

Attributes that are arrays are defined using the notation "attributeType[]", where attributeType is the type name for each item of the array. When the type is a structure, not a simple data type, then 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".

Attributes that are collections are represented as type "collection[...]". The resource type of the collection entities 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. See section 5.6 for more information about "Collection" entities.

This specification uses collections when the set of entities in the list will most likely be modified often and potentially by multiple Consumers. The use of collections in these cases allows for a more optimized, and user-friendly, interaction pattern , particularly to avoid needing to update with a single operation a very large number of items. Conversely, inlined arrays are used when it is expected that the list of entities will not be modified often and thus the overhead of separate resources seemed burdensome.

5.5.3 "Any" Type

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

5.6 Collections

Collections are groupings of resources of the same type. While different collections will contain entries of different types, all collections follow the pattern described below:

- Collections shall contain an "id" attribute which acts as a "self pointer". Retrieving the data at this reference shall return the collection. In the XML represenation, each collection shall be wrapped by a collection specific element.
- Collections shall contain an array of items that make up the collection. As with all arrays, if there are no entries in the collection then the serialization of the array shall be omitted.

- Each entry in the collection shall have an "id" attribute which acts as a "self pointer". Retrieving the data at this references shall return just that one entry and not any parent resource such as the collection or array attribute. In the XML representation, each entry shall be wrapped by an <Entry> element.
- Adding new entries 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 entries are not permitted at that time.
- Deleting entries from the collection shall be done via a "delete" operation on the entry itself.
- Attributes that are references to collections shall be read-only for Consumers. In general, they can only be updated by performing CRUD operations on the collection's URL or on the URL of individual members.
- When a resource that contains a reference to a collection that is owned/managed by that resource is deleted, its referenced collections shall also be deleted. For example, deleting a Machine shall also delete its collection of Disks.

The serialization of collections shall adhere to the following pattern:

JSON serialization:

```
{ "entityURI": string,
  "id": string,
  "entries": [
    { "entityURI": string,
      "id": string,
      ... entry specific data ...
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    } +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection entityURI="xs:anyURI" xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="xs:anyURI">
    <id> xs:anyURI </id>
    ... entry specific data ...
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

Where the entityURI attributes shall contain the collection or entry specific URIs for that type of collection.

5.6.1 Adding Items to Collections

There are two types of collections used in this specification. The first represents a list of resources that represents a set of data or defines a relationship between two resources. In these cases when invoking the "add" operation the body of the request shall be the contents of the new entry to be added. In other words just the "...entry specific data..." shown in the previous section. In the case of XML, this data shall be wrapped with an <Entry> element. For example, to add new Volume to a Machine, the "add" operation's request body will be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineVolumeCollectionEntry",
  "attachmentPoint": string,
  "volume": { "href": string }
}
```

XML serialization:

```
<Entry entityURI="http://www.dmtf.org/cimi/MachineVolumeCollectionEntry"
       xmlns="http://www.dmtf.org/cimi">
  <attachmentPoint> xs:string </attachmentPoint>
  <volume href="xs:string"/>
</Entry>
```

Deleting this type of collection entry will delete the entry, which implicitly also removes it from the collection, but it SHALL NOT delete any resource referenced by the entry.

The other type of collection represents a grouping CIMI resources that are created and added to the collection at the same time. Many CIMI resource uses Templates as part of the parameters to the "add" operation - see the definition of the specific resource for details. In these cases the "add" operation shall contain:

- The "common attributes" as defined by section 5.10.1, and in the JSON case, the "entityURI" attribute.
- The resource specific data needed to create it. In many cases this will either be a reference to the resource-specific Template entity or the resource-specific Template entity itself inlined. But it might also be a serialization of the new resource itself - see the resource definition.
- In the XML case, a wrapper element (named <*ResourceName*Create>).

Note that this will create the new resource and add a new entry, which includes that reference to the new resource, to the collection. This reference to the new resource, not to the new collection entry,shall also be returned in the response message in the HTTP Location header.

For example, to create a new Machine and add it to the MachineCollection, the "add" operation of the MachineCollection will be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineCreate", ?
  "name": string, ?
  "description": string, ?
  "properties": { "key": string, + }, ?
  "machineTemplate": { "href": string z}
  ...
}
```

XML serialization:

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

And the MachineCollection will have a new entry:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineCollectionEntry",
  "id": string,
  "machine": { "href": string },
}
```

XML serialization:

```
<Entry entityURI="http://www.dmtf.org/cimi/MachineCollectionEntry"
       xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <machine href="xs:string"/>
</Entry>
```

The processing of the "add" operation shall adhere to the semantics defined in section 4.3.2.1.

When the "add" operation's request message includes a resource whose representation would normally include a reference to a collection (e.g. MachineTemplate.volumes), those attributes shall have the collection "inlined" in the request. This avoids the need for the Consumer to create a new collection resource (and all of its entries) just for the "add" operation.

The entries in these types of collections SHALL NOT have "edit" or "delete" operations. Since there are no Consumer-editable properties within each entry, there is no need to provide an "edit" operation. To delete an entry from the collection the "delete" operation of the referenced resource SHALL be invoked; for example, on the Machine resource in the previous example. Deleting the referenced resource SHALL not only remove the resource from the Providers environment but it SHALL also remove the associated entry from the collection from which the resource was created.

5.7 Relationship Semantics

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

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

5.8 Operations

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

5.9 Alternative Model Formats

Since it is expected that this specification will be implemented 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 Entities

The following sections detail the attributes of the entities defined by the CIMI model.

5.10.1 Common Attributes

Except for EntityMetadata, the entities described by this document share the following common attributes.

Attribute	Type	Description
id	*ref*	The unique self-reference to this entity; assigned upon entity creation. This attribute value shall be **unique** in the Provider's cloud. **Constraints:** **Provider:** support mandatory ; immutable **Consumer:** support mandatory ; read-only
name	*string*	The human readable name of this entity; assigned by the creator as a part of the entity creation input. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support optional ; read-write
description	*string*	The human readable description of this entity; assigned by the creator as a part of the entity creation input. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support optional ; read-write
created	*dateTime*	The timestamp when this entity was created. The format should be unambiguous, and the value is **immutable**. **Constraints:** **Provider:** support optional ; immutable **Consumer:** support optional ; read-only
updated	*dateTime*	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'. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
properties	*map*	A list of key/value pairs, some of which may control one or more aspects this entity. 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. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support optional ; read-write

5.11 Entity Metadata

Implementations of this specification SHOULD allow for Consumers to discover the metadata associated with each supported entity. 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. EntityMetadata can also be used to express 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 EntityMetadata 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 entity's metadata will contain the following pieces of information:

Name	EntityMetadata	
Type URI	http://www.dmtf.org/cimi/EntityMetadata	
Attribute	**Type**	**Description**
id	*ref*	The unique self-reference to this entity; assigned upon entity creation. This attribute value is **immutable**, and shall be **unique** in the Provider's cloud. **Constraints:** **Provider:** support mandatory ; immutable **Consumer:** support mandatory ; read-only
typeURI	*URI*	A unique URI associated with, and denoting, this entity type. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
name	*string*	The name of the entity type. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
attributes	*attribute[]*	A set of Provider defined metadata that can be used by clients to discover any metadata associated with each attribute, as well as the set of extension attributes. Each attribute will contain the following nested data:

Name	*attribute*	
Data	**Type**	**Description**

<table>
<tr>
<td></td>
<td></td>
<td>
<table>
<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 entity 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>constraints</td><td>any</td><td>Type specific data that describes the constraints of this attribute. When absent the there are no constraints.

<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>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 will 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>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<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 default 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 clients to act on the entity. Note that this attribute is called "actions" as to not conflict with the EntityMetadata resource's operations.

Each operation will contain the following nested data:
<table>
<tr><th>Name</th><td colspan="2">operation</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.

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

		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 entity in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/EntityMetadata",
  "id": string,
  "typeURI": URI,
  "name": string,
  "attributes" : [
    { "name": string,
      "namespace": string, ?
      "type": string, ?
      "required": boolean, ?
      ...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:

```
<EntityMetadata xmlns="http://www.dmtf.org/cimi">
  <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"? >
    ...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>*
</EntityMetadata>
```

Additional metadata about the entity or attributes MAY be included by the Provider.

5.11.1 Attribute Types

The following describes the values, syntax and serialization of the "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 which indicates whether the attribute is required to be either 'true' or 'false'.

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. For brevity the "Name & URI" column in the table only shows that last part of the URI - it should be appended to: http://www.dmtf.org/cimi/capability/ and this is also the "name" of each capability. The "Entity" column contains the name of the entity that may contain the specified capability within its EntityMetadata.

Entity	Name & URI	Description
Job	jobRetention	If the Provider supports Job entities as specified in this document, this capability indicates in minutes how long a job will live in the system before its deleted. In this case, the value attribute provides the number of minutes (e.g. 30 min). The value specified is of type "integer".
Volume	sharedVolumeSupport	The Provider supports the sharing of volume entities across Machines. The value specified is of type "boolean".
System	SystemComponentTemplateByValue	Indicates that the Provider supports specifying Component Templates by-value in SystemTemplates.
System	SystemTemplateByValue	Indicates that the Provider supports specifying System Templates by-value in the System create operation.
System	MachineTemplateByValue	Indicates that the Provider supports specifying Machine Templates by-value in Machine create operations.
Machine	MachineConfigByValue	Indicates that the Provider supports specifying Machine Configurations by-value in Machine create operations. If true the MachineTemplateByValue capability shall also be specified with a value of true.
Machine	MachineImageByValue	Indicates that the Provider supports specifying Machine Images by-value in Machine create operations. If true the MachineTemplateByValue capability shall also be specified with a value of true.

Machine	MachineCredentialsByValue	Indicates that the Provider supports specifying Credentials by-value in Machine create operations. If true the MachineTemplateByValue capability shall also be specified with a value of true.
Machine	MachineVolumesByValue	Indicates that the Provider supports specifying Volumes by-value in Machine create operations. If true the MachineTemplateByValue capability shall also be specified with a value of true.
Machine	MachineVolumeTemplatesByValue	Indicates that the Provider supports specifying VolumeTemplates by-value in Machine create operations. If true the MachineTemplateByValue capability shall also be specified with a value of true.
Machine	InitialStates	Indicates the list of allowable initial states that Consumer may choose from when creating a new Machine.
Machine	DefaultInitialState	Indicates what the default initial state of a new Machine will be unless explicitly set by the "initialState" attribute of the MachineTemplate.
Machine	UserData	Indicates which userData injection method will be used. See 5.14.1 for more information.
Machine	MachineStopForce	Indicates that the Provider supports specifying the "force" option on the stop and restart operations.
Machine	MachineStopForceDefault	Indicates the default way in which the Provider will stop/restart a Machine. When set to "true", the Provider will forcefully stop the Machine, as opposed to a value of "false" which indicates that the Provider will attempt to gracefully stop the Machine.
Credentials	CredentialsTemplateByValue	Indicates that the Provider supports specifying Credential Templates by-value in Credential create operations.
Volume	VolumeTemplateByValue	Indicates that the Provider supports specifying Volume Templates by-value in Volume create operations.
Volume	VolumeConfigByValue	Indicates that the Provider supports specifying Volume Configurations by-value in the Volume create operation. If true the VolumeTemplateByValue capability shall also be specified with a value of true.
Volume	VolumeImageByValue	Indicates that the Provider supports specifying Volume Images by-value in the Volume create operation. If true the VolumeTemplateByValue capability shall also be specified with a value of

		true.
Volume	VolumeSnapshot	Indicates that the Provider supports creating a new VolumeImage by referencing an existing Volume.
Network	NetworkTemplateByValue	Indicates that the Provider supports specifying Network Templates by-value in the Network create operation.
Network	NetworkConfigByValue	Indicates that the Provider supports specifying Network Configurations by-value in the Network create operation.
VSP	VSPTemplateByValue	Indicates that the Provider supports specifying VSP Templates by-value in the VSP create operation.
VSP	VSPConfigByValue	Indicates that the Provider supports specifying VSP Configurations by-value in the VSP create operation.
RoutingGroup	MixedNetwork	Indicates whether RoutingGroups can support both private and public connection at the same time.
Meter	MeterTemplateByValue	Indicates that the Provider supports specifying Meter Templates by-value in the Meter create operation.

5.11.3 Examples

The following shows a sample metadata document for a VolumeConfiguration entity in XML that has been extended with a "Location" string attribute:

```
<EntityMetadata xmlns="http://www.dmtf.org/cimi">
  <id> http://example.org/types/VC </id>
  <typeURI> http://www.dmtf.org/cimi/VolumeConfiguration </typeURI>
  <name> VolumeConfiguration </name>
  <attribute name="Location" namespace="http://example.org/" type="string"/>
</EntityMetadata>
```

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

```
<EntityMetadata xmlns="http://www.dmtf.org/cimi">
  <id> http://example.org/types/VC </id>
  <typeURI> http://www.dmtf.org/cimi/VolumeConfiguration </typeURI>
  <name> VolumeConfiguration </name>
  <attribute name="Location" namespace="http://example.org/" type="string"
             required="true">
    <value> NYC </value>
    <value> LAX </value>
  </attribute>
</EntityMetadata>
```

The following shows the same VolumeConfiguration serialized in JSON:

```
{ "entityURI": "http://www.dmtf.org/cimi/VolumeConfiguration",
  "id": "http://example.org/types/VC",
```

```
  "typeURI": "http://www.dmtf.org/cimi/VolumeConfiguration",
  "name": "VolumeConfiguration",
  "attributes": [
    { "name": "Location",
      "namespace": "http://example.org",
      "type": "string",
      "required": true,
      "values": [ "NYC", "LAX" ]
    }
  ]
}
```

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

```
{ "entityURI": "http://www.dmtf.org/cimi/VolumeConfiguration",
  "id": "http://example.org/types/V",
  "typeURI": "http://www.dmtf.org/cimi/Volume",
  "name": "Volume",
  "operations": [
    {
      "name": "compress",
      "uri": "http://example.org/cimi/action/compress"
      "description": "Compress the data stored in the volume",
      "method": "POST"
    }
  ]
}
```

5.11.4 EntityMetadata Collection

A EntityMetadata Collection entity represents the collection of Entity Metadata entities within a Provider and follows the Collection pattern defined in section 5.6.Note that modificiations of the entities within this collection will typically be reserved for administrator type of CIMI Consumers. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/EntityMetadataCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/EntityMetadataCollectionEntry",
      "id", string,
      "entityMetadata": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/EntityMetadataCollection"
            xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/EntityMetadataCollectionEntry">
    <id> xs:anyURI </id>
    <entityMetadata href="xs:anyURI"/>
  </Entry> *
  <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 entities such as Systems, System Templates, Machines, Machine Templates, etc. that can be queried and browsed by the Consumer.

Name	CloudEntryPoint	
Type URI	http://www.dmf.org/cimi/CloudEntryPoint	
Attribute	**Type**	**Description**
rootURI	*URI*	An absolute URI that references the "root URI" of the Provider. This URI shall be used to convert relative URIs to resources within this Provider to absolute URIs. See the "URIs" section of 5.4. **Constraints:** **Provider:** support mandatory ; immutable **Consumer:** support mandatory ; read-only
systemTemplates	*ref*	A reference to the System Template Collection of this CloudEntry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
systems	*ref*	A reference to the System Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
machineTemplates	*ref*	A reference to the Machine Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
machineConfigs	*ref*	A reference to the Machine Configuration Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
machineImages	*ref*	A reference to the Machine Image Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
credentialsTemplates	*ref*	A reference to the Credentials Template Collection of this Cloud Entry Point.

		Constraints: **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
credentials	*ref*	A reference to the Credentials Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
machines	*ref*	A reference to the Machine Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
volumeTemplates	*ref*	A reference to the Volume Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
volumeConfigs	*ref*	A reference to the Volume Configuration Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
volumeImages	*ref*	A reference to the Volume Image Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
volumes	*ref*	A reference to the Volume Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
networkTemplates	*ref*	A reference to the Network Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
networkConfigs	*ref*	A reference to the Network Configuration Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
networks	*ref*	A reference to the Network Collection of this Cloud Entry Point.

		Constraints: **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
addressTemplates	*ref*	A reference to the Address Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
addresses	*ref*	A reference to the Address Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
vspTemplates	*ref*	A reference to the VSP Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
vspConfigs	*ref*	A reference to the VSP Configuration Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
vsps	*ref*	A reference to the VSP Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
routingGroupTemplates	*ref*	A reference to the Routing Group Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
routingGroups	*ref*	A reference to the Routing Group Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
meterTemplates	*ref*	A reference to the Meter Template Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only

meterConfigurations	*ref*	A reference to the Meter Configuration Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
meters	*ref*	A reference to the Meter Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
eventLogs	*ref*	A reference to the Event Log Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
events	*ref*	A reference to the Event Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
jobs	*ref*	A reference to the Jobs Collection of this Cloud Entry Point. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
entityMetadata	*ref*	A reference to EntityMetadata Collection of this Cloud Entry Point. The collection contains the entities supported by the Provider. If an entity does not have any metadata then it will 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

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/CloudEntryPoint",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "rootURI": string,
  "systemTemplates": { "href": string }, ?
  "systems": { "href": string }, ?
  "machineTemplates": { "href": string }, ?
  "machineConfigs": { "href": string }, ?
  "machineImages": { "href": string }, ?
```

```
  "credentialsTemplates": { "href" string }, ?
  "credentials": { "href" string }, ?
  "machines": { "href": string }, ?
  "volumeTemplates": { "href": string }, ?
  "volumeConfigs": { "href": string }, ?
  "volumeImages": { "href": string }, ?
  "volumes": { "href": string }, ?
  "networkTemplates": { "href": string }, ?
  "networkConfigs": { "href": string }, ?
  "networks": { "href": string }, ?
  "addressTemplates": { "href": string }, ?
  "addresses": { "href": string }, ?
  "vspTemplates": { "href": string }, ?
  "vspConfigs": { "href": string }, ?
  "vsps": { "href": string }, ?
  "routingGroups" { "href": string }, ?
  "routingGroupTemplates" { "href": string }, ?
  "meterTemplates": { "href": string }, ?
  "meterConfigurations": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLogs": { "href": string }, ?
  "events": { "href": string }, ?
  "jobs": { "href": string }, ?
  "entityMetadata": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<CloudEntryPoint xmlns="http://www.dmtf.org/cimi">
  <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> *
  <rootURI> xs:anyURI </rootURI>
  <systemTemplates href="xs:anyURI"/> ?
  <systems href="xs:anyURI"/> ?
  <machineTemplates href="xs:anyURI"/> ?
  <machineConfigs href="xs:anyURI"/> ?
  <machineImages href="xs:anyURI"/> ?
  <credentialsTemplates href="xs:anyURI"/> ?
  <credentials href="xs:anyURI"/> ?
  <machines href="xs:anyURI"/> ?
  <volumeTemplates href="xs:anyURI"/> ?
  <volumeConfigs href="xs:anyURI"/> ?
  <volumeImages href="xs:anyURI"/> ?
  <volumes href="xs:anyURI"/> ?
  <networkTemplates href="xs:anyURI"/> ?
  <networkConfigs href="xs:anyURI"/> ?
  <networks href="xs:anyURI"/> ?
  <addressTemplates href="xs:anyURI"/> ?
  <addresses href="xs:anyURI"/> ?
  <vspTemplates href="xs:anyURI"/> ?
  <vspConfigs href="xs:anyURI"/> ?
  <vsps href="xs:anyURI"/> ?
  <routingGroupTemplates href="xs:anyURI"/> ?
```

```
  <routingGroups href="xs:anyURI"/> ?
  <meterTemplates href="xs:anyURI"/> ?
  <meterConfigurations href="xs:anyURI"/> ?
  <meters href="xs:anyURI"/> ?
  <eventLogs href="xs:anyURI"/> ?
  <events href="xs:anyURI"/> ?
  <jobs href="xs:anyURI"/> ?
  <entityMetadata href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</CloudEntryPoint>
```

5.12.1 Operations

This entity supports the Read and Update operations.

5.13 System Entities and Relationships

The following diagram illustrates the entities involved in constructing a System and their relationships. Although this drawing is in the style of an Entity Relationship diagram, the use of UML is neither rigorous nor normative.

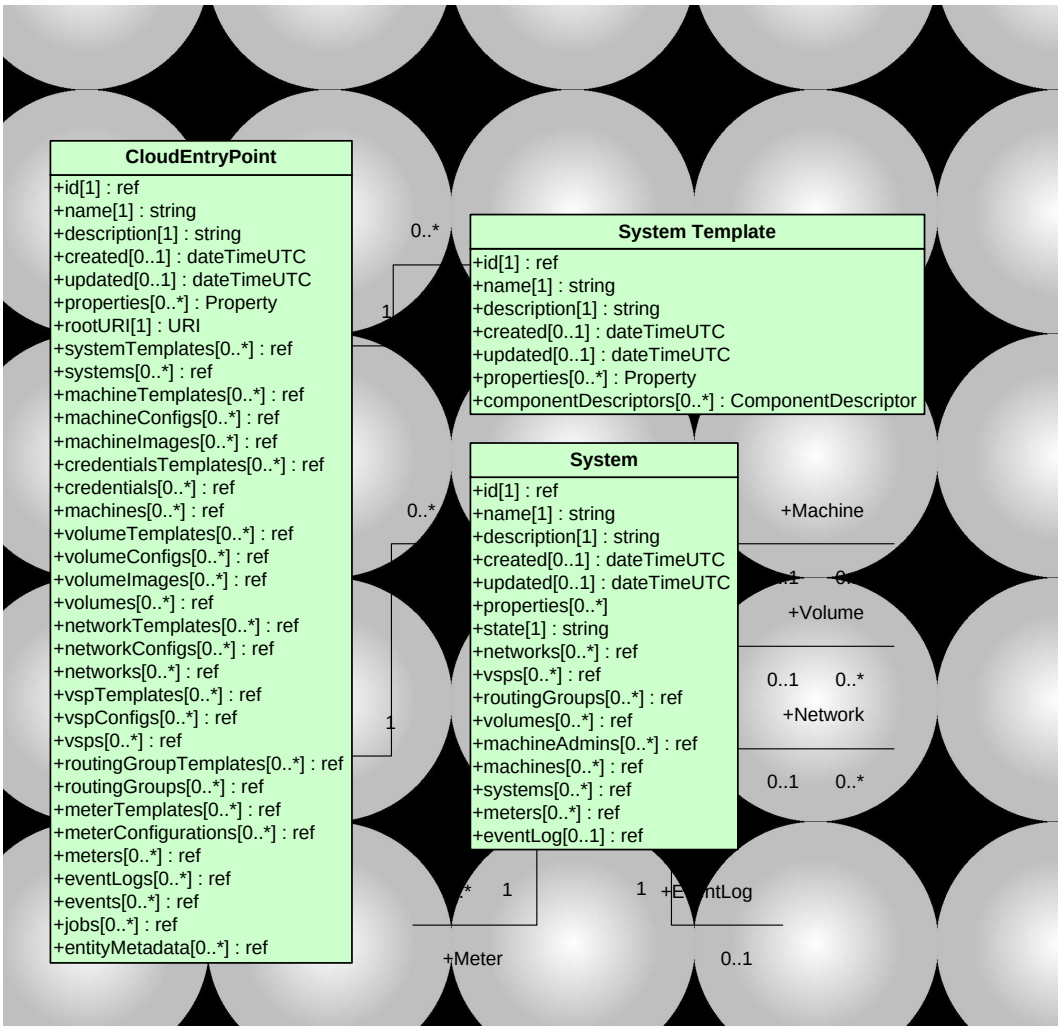


Figure 1 - System Entities

5.13.1 System Template

The System Template contains the set of individual descriptors 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 "newVolume".

Name	SystemTemplate	
Type URI	http://www.dmtf.org/cimi/SystemTemplate	
Attribute	**Type**	**Description**
componentDescriptors	*componentDescriptor[]*	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.

Name	*componentDescriptor*	
Data	**Type**	**Description**
name	*string*	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. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support optional ; read-write
description	*string*	The value of the "description" attribute that will be associated with a System component created from this component descriptor. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support optional ; read-write
properties	*map*	The key/value pairs that will be associated with a System component created from this component descriptor. **Constraints:**

				Provider: support mandatory ; mutable **Consumer:** support optional ; read-write
		type	*URI*	The TypeURI of the component to be created from this component descriptor, e.g. for a machine: http://www.dmtf.org/cimi/Machine **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
		component Template	*any*	Either a reference to a component Template or the Template data itself inlined (i.e. the Template "value"). Note that the exact name of this attribute will vary depending on the type of entity being created. E.g. MachineTemplate for a Machine. Note: Component references (expressing links between components of a resulting System) are to be found – if any - in Templates that are provided inline, as such references contain names that are only relevant to the SystemTemplate where these template values are embedded. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
		quantity	*integer*	Number of component instances to be created from this component descriptor. By default, 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.) **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
		Constraints: **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-only		

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/SystemTemplate",
  "id": string,
  "name": string, ?
```

```
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "componentDescriptors": [
    { "name": string, ?
      "description": string, ?
      "properties": { "name": string, + }, ?
      "type": string,
      "componentTemplate": {
        "href": string, ?
        template-values ?
      }
    }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/export", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<SystemTemplate xmlns="http://www.dmtf.org/cimi">
  <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 name="xs:string"> xs:string </property> *
    <type> xs:anyURI </type>
    <componentTemplate href="xs:anyURI"? >
      template-values ?
    </componentTemplate>
  </componentDescriptor> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/export" href="xs:anyURI"/> ?
  <xs:any>*
</SystemTemplate>
```

5.13.1.1 Operations

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

The following custom operations are also defined:

Exporting a SystemTemplate

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

This operation is defined to 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:

- format - string - optional
  Indicates the Media Type of the exported data. If not present, the default value shall be “application/ovf”.

- destination - 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/REST Protocol**

To export a SystemTemplate a POST is sent to the "http://www.dmtf.org/cimi/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://www.dmtf.org/cimi/action/export",
  "format": string, ?
  "destination": string, ?
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://www.dmtf.org/cimi">
  <action> http://www.dmtf.org/cimi/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 Template Collection

A System Template Collection entity represents the collection of System Template entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/SystemTemplateCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/SystemTemplateCollectionEntry",
      "id", string,
      "systemTemplate": { "href": string },
    }, +
```

```
  ], ?
  "operations": [
    { "rel": "add", "href", string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/import", "href", string } ?
  ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/SystemTemplateCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/SystemTemplateCollectionEntry">
    <id> xs:anyURI </id>
    <systemTemplate href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/import" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.2.1 Operations

The following custom operations are defined:

Importing a SystemTemplate

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

This operation will import/deserialize 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:

- source - 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/REST Protocol**

To import a SystemTemplate a POST is sent to the "http://www.dmtf.org/cimi/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://www.dmtf.org/cimi/action/import",
  "source": string, ?
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

5.13.3 System

A System is a realized entity that consists of for example: 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 entity 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 which 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 can be overridden by subsequent modifications to the top-level System collection attributes.)
- Ownership of a resource to 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 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).

Name	System	
Type URI	http://www.dmtf.org/cimi/System	
Attribute	**Type**	**Description**
state	*string*	Indicates the operational state of the System. Allowable values include: **CREATING**: The System is in the process of being created. Allowable action when in this state is: **delete**. **STARTING/STARTED/STOPPING/STOPPED/PAUSING/PAUSED/SUSPE**

		NDING/SUSPENDED: All of the Machines referenced by this System are one of these states. See section 5.14.7 for the list of available actions based on the state of a Machine. **MIXED**: This state indicates that either there are no Machines referenced by this System or that the Machines referenced by this System are in varying states. Allowable action when in this state is: **delete**. **DELETING**: The System is in the process of being deleted. Allowable action when in this state is: **delete**. **ERROR**: The Provider has detected an error in the System. Allowable action when in this state is: **delete**. Providers may define additional values. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-only
networks	*collection[network]*	The list of Networks contained in 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. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
vsps	*collection[vsp]*	The list of VSPs contained in this System. Adding an item (of type VSP) to this list is logically equivalent to associating the VSP to this System with a "containment relationship". Removing an item from this list is logically equivalent to de-associating the VSP from this System. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
routingGroups	*collection[routinggroup]*	The list of Routing Groups contained in this System. Adding an item (of type RoutingGroup) to this list is logically equivalent to associating the Routing Group to this System with a "containment relationship". Removing an item from this list is logically equivalent to de-associating the Routing Group from this System. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
volumes	*collection[volume]*	The list of Volumes contained in 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. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
credentials	*collecti*	The list of Credentials contained in this System. Adding an item (of type

	on[credential]	Credentials) to this list is logically equivalent to associating the Credentials to this System with a "containment relationship". Removing an item from this list is logically equivalent to de-associating the Credentials from this System. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
machines	*collection[machine]*	The list of Machines contained in 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. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
systems	*collection[system]*	The list of nested Systems contained in 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. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
meters	*collection[meter]*	A list of references to Meters monitored for this System. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
eventLog	*ref*	A reference to the EventLog of this System. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/System",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "state": string,
  "networks": { "href": string }, ?
  "vsps": { "href": string }, ?
  "volumes": { "href": string }, ?
  "credentials": { "href": string }, ?
  "machines": { "href": string }, ?
  "systems": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
```

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

XML media type: application/xml

XML serialization:

```
<System xmlns="http://www.dmtf.org/cimi">
  <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>
  <networks href="xs:anyURI"/> ?
  <vsps href="xs:anyURI"/> ?
  <volumes href="xs:anyURI"/> ?
  <credentials href="xs:anyURI"/> ?
  <machines href="xs:anyURI"/> ?
  <systems 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://www.dmtf.org/cimi/action/start" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/stop" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/restart" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/pause" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/suspend" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/export" href="xs:anyURI"/> ?
  <xs:any>*
</System>
```

5.13.3.1 Collections

The following describes the JSON and XML serialization of the System collection entities.

5.13.3.1.1 Networks

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/SystemNetworksCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/SystemNetworksCollectionEntry",
      "id": string,
      "network": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
```

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

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/SystemNetworksCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http"//www.dmtf.org/cimi/SystemNetworksCollectionEntry">
    <id> xs:anyURI </id>
    <network href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.13.3.1.2 VSPs

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/SystemVSPsCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/SystemVSPsCollectionEntry",
      "id": string,
      "vsp": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/SystemVSPsCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/SystemVSPsCollectionEntry">
    <id> xs:anyURI </id>
    <vsp href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.13.3.1.3 Volumes

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/SystemVolumesCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/SystemVolumesCollectionEntry",
      "id": string,
      "volume": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
```

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

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/SystemVolumesCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/SystemVolumesCollectionEntry">
    <id> xs:anyURI </id>
    <volume href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.13.3.1.4 Credentials

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/SystemCredentialsCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/SystemCredentialsCollectionEntry",
      "id": string,
      "credentials": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection entityURI="SystemCredentialsCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/SystemCredentialsCollectionEntry">
    <id> xs:anyURI </id>
    <credentials href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.13.3.1.5 Machines

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/SystemMachinesCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/SystemMachinesCollectionEntry",
      "id": string,
      "machine": { "href": string },
```

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

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/SystemMachinesCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURi="http://www.dmtf.org/cimi/SystemMachinesCollectionEntry">
    <id> xs:anyURI </id>
    <machine href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.13.3.1.6 Meters

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/SystemMetersCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/SystemMetersCollectionEntry",
      "id": string,
      "meter": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/SystemMetersCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/SystemMetersCollectionEntry">
    <id> xs:anyURI </id>
    <meter href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.13.3.2 Operations

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

The following custom operations are also defined:

Starting/Stopping/Restarting/Pausing/Suspending the Machines in a System

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

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

This operation will 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 what impact this operation will have will vary depending on the component's current state - see section 5.14.7.2 for more details about performing operations on Machines. If a Machine is in a state that makes this operation invalid, then that Machine will not be affected by the operation.

To start, stop, restart, pause or suspend the Machines in a System a POST is sent to the appropriate URI of the System; where the HTTP request body SHALL be as described in the "Operations" section of the Machine entity - section 5.14.7.2.

Exporting a System

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

This operation is defined to 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:

- format - string - optional
  Indicates the Media Type of the exported data. If not present, the default value shall be “application/ovf”.

- destination - 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/REST Protocol**

To export a System a POST is sent to the "http://www.dmtf.org/cimi/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://www.dmtf.org/cimi/action/export",
  "format": string, ?
  "destination": string, ?
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://www.dmtf.org/cimi">
  <action> http://www.dmtf.org/cimi/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 Collection

A System Collection entity represents the collection of System entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/SystemCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/SystemCollectionEntry",
      "id", string,
      "system": { "href": string },
    }, +
  ], ?
  "operations": [
    { "rel": "add", "href", string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/import", "href", string } ?
  ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/SystemCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/SystemCollectionEntry">
    <id> xs:anyURI </id>
    <system href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/import" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.4.1 Operations

Note, the "add" operation requires a SystemTemplate to be used.

Resources created during the process of creating a System shall be "owned" by the System (see 5.13.3). 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 exist Volume, then this Vlume will not be added to the top-level System attributes.

The following custom operations are also defined:

Importing a System

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

This operation will import/deserialize 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 .

Input parameters:

- source - 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/REST Protocol**

To import a SystemTemplate a POST is sent to the "http://www.dmtf.org/cimi/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://www.dmtf.org/cimi/action/import",
  "source": string, ?
  "properties": { "key": string, + } ?
  ...
}
```

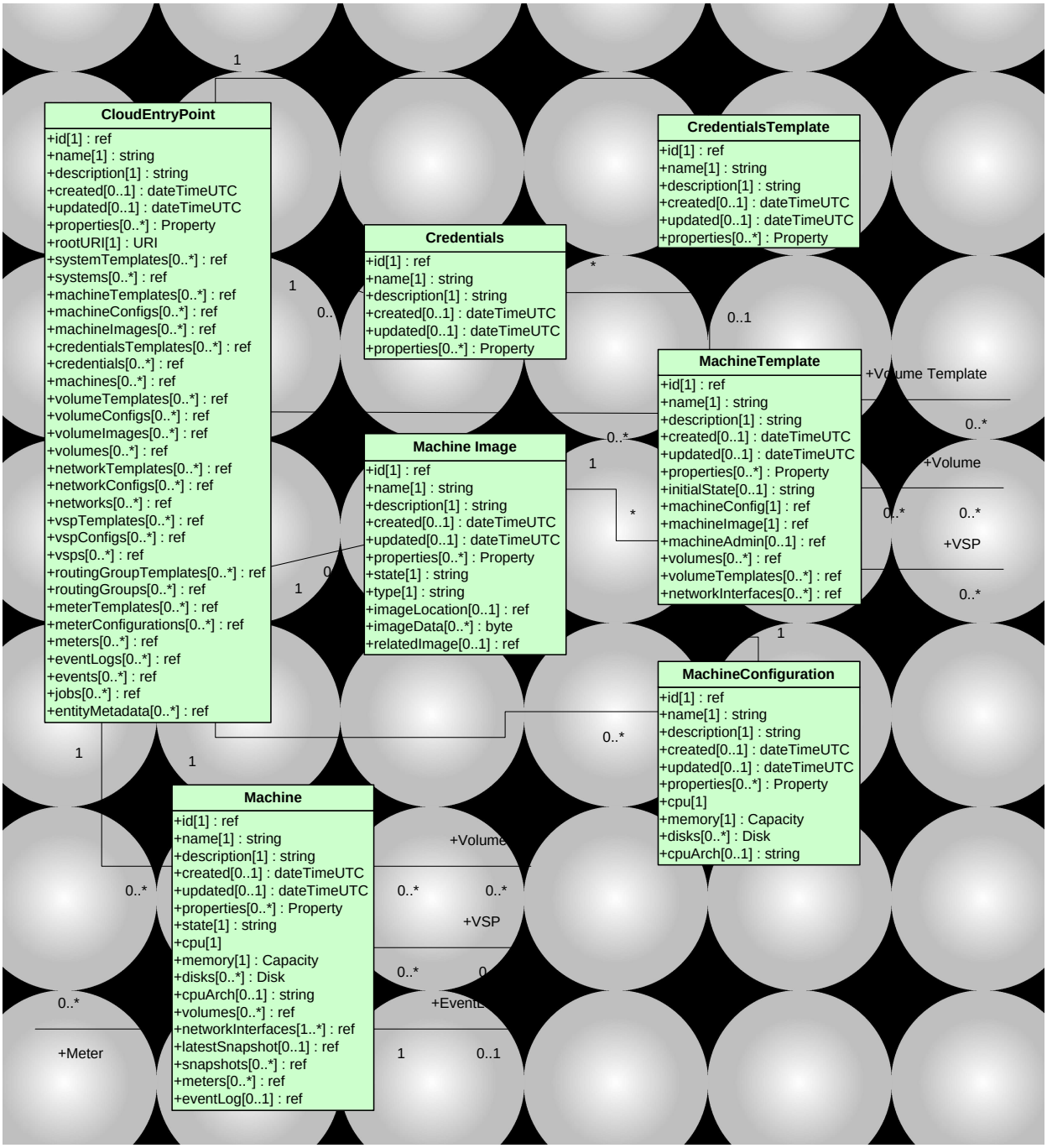
XML media type: application/xml

XML serialization

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

5.14 Machine Entities and Relationships

The following diagram illustrates the entities involved in constructing a Machine and their relationships. Although this drawing is in the style of an Entity Relationship diagram, the use of UML is neither rigorous nor normative.



2211 **Figure 2 - Machine Entities**

2212 ### 5.14.1 Machine Template

2213 A Machine Template represents the set of metadata and instructions used in the creation of a Machine.

Name	MachineTemplate	
Type URI	http://www.dmtf.org/cimi/MachineTemplate	
Attribute	**Type**	**Description**

<table>
<tr><td>initialState</td><td>string</td><td>The initial state of the new Machine.

<u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-write</td></tr>
<tr><td>machineConfig</td><td>ref</td><td>A reference to the Machine Configuration that will be used to create a Machine from this Machine Template.

<u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-write</td></tr>
<tr><td>machineImage</td><td>ref</td><td>A reference to the Machine Image that will be used to create a Machine from this Machine Template.

<u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-write</td></tr>
<tr><td>credentials</td><td>ref</td><td>A reference to the Credentials that will be used to create the initial login credentials for the new Machine.

<u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-write</td></tr>
<tr><td>volumes</td><td>volume[]</td><td>A list of references to existing Volumes that will be attached to the Machine during its creation.

Each volume has the following attributes which describe aspects of the way in which the Machine will be attached to the Volume:
<table>
<tr><th>Name</th><td colspan="2">volume</td></tr>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>attachmentPoint</td><td>string</td><td>File system path where the Volume will be attached.

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

<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-only</td></tr>
</table>

<table>
<tr>
<td>volumeTemplates</td>
<td>volumeTemplate[]</td>
<td>
A list of references to Volume Templates that will be used to create a set of new Volumes that will to be attached to the Machine during its creation.

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 <code>volumeTemplates</code> attribute of the relevant System Template. If the same Volume Template reference is listed in both the <code>volumeTemplates</code> 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 has the following attributes which describe aspects of the way in which the Machine will be attached to the Volume instance that will be created from the template:
<table>
<tr><th>Name</th><td colspan="2">volumeTemplate</td></tr>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>attachmentPoint</td><td>string</td><td>File system path where the Volume will be attached.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; 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.

<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-only
</td>
</tr>
<tr>
<td>networkInterfaces</td>
<td>networkInterface[]</td>
<td>
A list of entities that define the network interfaces that will be created on Machines instantiated from this template.
<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>address[]</td><td>A list of references to the Addresses for this network interface.</td></tr>
</table>
</td>
</tr>
</table>

				Constraints: **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-only
		network	*ref*	A reference to the Network for this network interface. It is expected that VSPs and Networks will be defined separately and prior to the Machines that connect to them. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
		vsp	*ref*	A reference to the VSP (Virtual Switch Port) for this network interface. Note this is a reference to a VSP and not a VSPTemplate. It is expected that VSPs and Networks will be defined separately and prior to the Machines that connect to them. If this attribute is provided then the "network" attribute in the referenced VSP 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 an interface configurable to be "Active" or "Passive". A passive interface is in a standby mode ready to forward traffic if the primary interface fails.

<table>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td><u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-write</td></tr>
<tr><td>maxTransmissionUnit</td><td>integer</td><td>To set the largest supported packet size.

<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-only</td></tr>
<tr><td>userData</td><td>string</td><td>A Base64 encoded string whose decoded version is to be injected into Machines created using this template. See the discussion of injection of user-defined data below.

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

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "initialState": string, ?
  "machineConfig": { "href": string }, ?
  "machineImage": { "href": string }, ?
  "credentials": { "href": string }, ?
  "volumes": [
    { "href": string, "attachmentPoint": string }, +
  ], ?
  "volumeTemplates": [
    { "href": string, "attachmentPoint": string }, +
  ], ?
  "networkInterfaces": [
    { "addresses": [
        {"href": string}, +
      ],
      "network": {"href": string},
      "vsp": {"href": string}, ?
      "state": string,
      "maxTransmissionUnit": number ?
    }, +
  ], ?
```

```
  "userData": string, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<MachineTemplate xmlns="http://www.dmtf.org/cimi">
  <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"/> ?
  <machineImage href="xs:anyURI"/> ?
  <credentials href="xs:anyURI"/> ?
  <volume href="xs:anyURI" attachmentPoint="xs:string" /> *
  <volumeTemplate href="xs:anyURI" attachmentPoint="xs:string" >
    template-values
  </volumeTemplate> *
  <networkInterface>
    <address href="xs:anyURI"/> *
    <network href="xs:anyURI"/>
    <vsp href="xs:anyURI"/> ?
    <state> xs:string </state>
    <maxTransmissionUnit> xs:integer </maxTransmissionUnit> ?
  </networkInterface> *
  <userData> xs:string </userData> ?
  <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 using the userData parameter. The value of this parameter must be the Base64-encoded payload. The Provider shall arrange for this data to be available from inside the Machine using one of the following three methods:

1. *Metadata server*: the data can be retrieved from within the instance 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 <*attachmentPoint*>/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 EntityMetadata 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.

5.14.1.1 Operations

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

5.14.2 Machine Template Collection

A Machine Template Collection entity represents the collection of Machine Template entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineTemplateCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/MachineTemplateCollectionEntry",
      "id", string,
      "machineTemplate": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/MachineTemplateCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/MachineTemplateCollectionEntry">
    <id> xs:anyURI </id>
    <machineTemplate href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.2.1 Operations

This entity supports the Read and Update operations. Creation of new Machine Template entities is supported via a POST to the "add" operation's URI as described in section 4.3.2.1.

5.14.3 Machine Configuration

The Machine Configuration entity 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://www.dmtf.org/cimi/MachineConfiguration	
Attribute	**Type**	**Description**
cpu	*integer*	Indicates the amount of CPU that a Machine realized from this configuration will have.

<table>
<tr><td></td><td></td><td><u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-write</td></tr>
<tr><td>memory</td><td><unnamed structure></td><td>Indicates the amount of RAM that a Machine realized from this configuration will have.

This attribute has the following sub-attributes which serve to describe it:
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>quantity</td><td>integer</td><td>A numerical quantity expressed as an integer.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>units</td><td>string</td><td>An enumerated value that expresses the unit of measurement used. Allowable values are byte, kibibyte, mebibyte, gibibyte, tebibyte, pebibyte, exbibyte, zebibyte, and yobibyte.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
</table>
<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>disks</td><td>disk[]</td><td>Contains the list of metadata of the disks that will be created upon the instantiation of a Machine from this configuration. The disks are local storage to the Machine.

Each disks attribute has the following sub-attributes:
<table>
<tr><th>Name</th><td colspan="2">disk</td></tr>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>capacity</td><td colspan="2">Indicates the initial capacity of the disk described by this attribute. This property is an (unnamed) structure that has the following sub-attributes.
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>quantity</td><td>integer</td><td>A numerical quantity expressed as an integer.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>units</td><td>string</td><td>An enumerated value that expresses the unit of measurement used. Allowable</td></tr>
</table></td></tr>
</table></td></tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>values are byte, kilobyte, megabyte, gigabyte, terabyte, petabyte, exabyte, zettabyte, and yottabyte.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
</table>
<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>attachmentPoint</td><td>string</td><td>File system path where this disk is attached.

<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-only</td></tr>
<tr><td>cpuArch</td><td>string</td><td>This property indicates the CPU architecture that will be supported by Machines created 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>
</table>

Note: The disk attributes "format" and "attachmentPoint" will not appear on Machine entities because once the Machine is created the user of the Machine will be able to modify those attributes of a disk, possibly without the Provider's knowledge. Therefore it might not be an aspect of the Machine that the Provider can reliably manage.

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineConfiguration",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "cpu": number,
```

```
  "memory": { "quantity": number, "units": string },
  "disks" : [
    { "capacity": { "quantity": number, "units": string },
      "format": string, "attachmentPoint": string }, +
  ], ?
  "cpuArch": string, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<MachineConfiguration xmlns="http://www.dmtf.org/cimi">
  <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 quantity="xs:integer" units="xs:string"/>
  <disk>
    <capacity quantity="xs:integer" units="xs:string">
    <format> xs:string </format>
    <attachmentPoint> xs:string </attachmentPoint>
  </disk> *
  <cpuArch> xs:string </cpuArch> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineConfiguration>
```

5.14.3.1 Operations

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

5.14.4 Machine Configuration Collection

A Machine Configuration Collection entity represents the collection of Machine Configuration entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineConfigurationCollection",
  "id": string,
  "entries": [
    { "entityURI":
        "http://www.dmtf.org/cimi/MachineConfigurationCollectionEntry",
      "id", string,
      "machineConfiguration": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/MachineConfigurationCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry
      entityURI="http://www.dmtf.org/cimi/MachineConfigurationCollectionEntry">
    <id> xs:anyURI </id>
    <machineConfiguration href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.4.1 Operations

This entity supports the Read and Update operations. Creation of new Machine Configuration entities is supported via a POST to the "add" operation's URI as described in section 4.3.2.1.

5.14.5 Machine Image

This entity represents the informationnecessary for hardware virtualized resources to create a Machine Instance; it contains configuration data such as startup instructions, including possible combinations of the following - 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 which 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 original disk image

Name	MachineImage	
Type URI	http://www.dmtf.org/cimi/MachineImage	
Attribute	**Type**	**Description**
state	*string*	Indicates the operational state of the MachineImage. Allowable values include: **CREATING**: The MachineImage is in the process of being created. Allowable action when in this state is: **delete**. **AVAILABLE**: The MachineImage is available and ready for use. Allowable action when in this state is: **delete**. **DELETING**: The MachineImage is in the process of being deleted. Allowable action when in this state is: **delete**. **ERROR**: The Provider has detected an error in the MachineImage. Allowable action when in this state is: **delete**. Providers may define additional values. **Constraints:** **Provider:** support mandatory ; mutable

		Consumer: support mandatory ; read-only
type	*string*	Indicates 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 then 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 - this 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 applies recursively to any subsequent "PARTIAL_SNAPSHOT" Machine Images as well. **Constraints:** **Provider:** support mandatory ; immutable **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
relatedImage	*ref*	A reference to another Machine Image resource that is related this to 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

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineImage",
```

```
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "state": string,
  "type": string,
  "imageLocation": { "href": string },
  "relatedImage": string, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<MachineImage xmlns="http://www.dmtf.org/cimi">
  <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 href="xs:anyURI"/>
  <relatedImage> xs:anyURI </relatedImage> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineImage>
```

5.14.5.1 Operations

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

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 entity being referenced:

http://www.dmtf.org/cimi/Machine

If the "imageLocation" is a reference to a Machine then the Provider shall create a new 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.

5.14.6 Machine Image Collection

A Machine Image Collection entity represents the collection of Machine Image entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineImageCollection",
```

```
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/MachineImageCollectionEntry",
      "id", string,
      "machineImage": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/MachineImageCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/MachineImageCollectionEntry">
    <id> xs:anyURI </id>
    <machineImage href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.6.1 Operations

This entity supports the Read and Update operations. Creation of new Machine Image entities is supported via a POST to the "addL" operation's URI as described in section 4.3.2.1, where the request body and the way it is processed is described in section 5.14.5.1.

5.14.7 Machine

An instantiated compute resource that encapsulates both CPU and Memory.

Name	Machine	
Type URI	http://www.dmtf.org/cimi/Machine	
Attribute	**Type**	**Description**
state	*string*	Indicates the operational state of the Machine. Allowable values include: **CREATING**: The Machine is in the process of being created. Allowable action when in this state is: **delete**. **STARTING**: The Machine is in the process of being started. Allowable actions when in this state are: **start restart**, **stop** and **delete**. **STARTED**: The Machine is available and ready for use. Allowable actions when in this state are: **stop**, **restart**, **pause**, **suspend**, **capture and delete**. **STOPPING**: The Machine is in the process of being stopped. Allowable actions when in this state are: **start**, **restart**, **stop** and **delete**. **STOPPED**: This is the virtual equivalent of powering off a physical Machine. There is no saved CPU or memory state. Allowable actions when in this state are: **start**, **restart**, **capture** and **delete**.

<table>
<tr><td></td><td></td><td>PAUSING: The Machine in the process of being PAUSED. Allowable actions when in this state are: start, restart and delete.

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. Allowable actions when in this state are: start, restart, capture and delete.

SUSPENDING: The Machine is in the process of being suspended. Allowable actions when in this state are:start, restart and delete.

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. Allowable actions when in this state are: start, restart, capture and delete.

DELETING: The Machine is in the process of being deleted. Allowable action when in this state is: delete.

ERROR: The Provider has detected an error in the Machine. Allowable actions when in this state are: start, restart, stop and delete.

PAUSED and SUSPENDED states are optional and Providers may choose to support them or not.

Providers may define additional values.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-only</td></tr>
<tr><td>cpu</td><td>integer</td><td>Indicates the amount of CPU that this Machine has.

<u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-write</td></tr>
<tr><td>memory</td><td><unnamed structure></td><td>The size of the memory (RAM) 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.

This attribute has the following sub-attributes which serve to describe it:
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>quantity</td><td>integer</td><td>A numerical quantity expressed as an integer.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>units</td><td>string</td><td>An enumerated value that expresses the unit of measurement used. Allowable values arebyte, kibibyte, mebibyte, gibibyte, tebibyte,</td></tr>
</table>
</td></tr>
</table>

		pebibyte, **exbibyte**, **zebibyte**, and **yobibyte**. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write

Constraints:
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write

disks	*collection[disk]*	The list of disks (local storage) that are part of the Machine. Adding an element to this list creates a disk. Each disk in the collection has the following attributes which describe aspects of the disk:

Name	disk	
Attribute	**Type**	**Description**
capacity	Indicates the initial capacity of the disk described by this attribute. This property is an (unnamed) structure that has the following sub-attributes.	

Attribute	**Type**	**Description**
quantity	*integer*	A numerical quantity expressed as an integer. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
units	*string*	An enumerated value that expresses the unit of measurement used. Allowable values are **byte**, **kilobyte**, **megabyte**, **gigabyte**, **terabyte**, **petabyte**, **exabyte**, **zettabyte**, and **yottabyte**. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write

Constraints:
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write

Constraints:
Provider: support optional ; mutable
Consumer: support optional ; read-only

<table>
<tr><td>cpuArch</td><td>string</td><td>This property indicates the CPU architecture that will be supported by Machines created 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 ; immutable
Consumer: support optional ; read-only</td></tr>
<tr><td>volumes</td><td>collection[volume]</td><td>The list of networked volumes that are attached 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.

Each volume in the collection has the following attributes which describe aspects of the way in which the Machine is attached to the Volume:
<table><tr><td>Name</td><td>volume</td></tr><tr><td colspan="2">See "volume" in "5.14.1 Machine Template" for the definition of this sub-entity.</td></tr></table>
<u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-only</td></tr>
<tr><td>networkInterfaces</td><td>collection[networkInterface]</td><td>A list of entities that define the network interfaces on this Machine.
<table><tr><td>Name</td><td colspan="2">networkInterface</td></tr><tr><td>Attribute</td><td>Type</td><td>Description</td></tr><tr><td>addresses</td><td>collection[address]</td><td>A list of references to the Addresses for this network interface.

<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 a Network for this network interface.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr><tr><td>vsp</td><td>ref</td><td>A reference to the VSP (Virtual Switch Port) for this network interface.

If this attribute is provided then the</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>"network" attribute in the referenced VSP 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 an interface configurable to be "Active" or "Passive".

A passive interface is in a standby mode ready to forward traffic if the primary interface fails.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>maxTransmissionUnit</td><td>integer</td><td>To set the largest supported packet size.

<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-only</td></tr>
<tr><td>latestSnapshot</td><td>ref</td><td>A reference to the SNAPSHOT representing the latest state captured for this Machine (either most recent Snapshot or the last Snapshot reverted to).

<u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-only</td></tr>
<tr><td>snapshots</td><td>collection[snapshot]</td><td>A list of references to the SNAPSHOT Machine Images taken of this Machine.

<u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-only</td></tr>
<tr><td>meters</td><td>collection[meter]</td><td>A list of references to Meters monitored for this Machine.

<u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-only</td></tr>
<tr><td>eventLog</td><td>ref</td><td>A reference to the EventLog of this Machine.</td></tr>
</table>

		Constraints: **Provider:** support optional ; mutable **Consumer:** support optional ; read-only

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Machine",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "state": string,
  "cpu": number,
  "memory": { "quantity": number, "units": string },
  "disks" : { "href": string }, ?
  "cpuArch", string, ?
  "volumes" : { "href": string }, ?[
  "networkInterfaces": { "href": string }, ?
  "latestSnapshot": string, ?
  "snapshots": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/start", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/stop", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/restart", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/pause", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/suspend", "href": string } ?
    { "rel": "http://www.dmtf.org/cimi/action/snapshot", "href": string } ?
    { "rel": "http://www.dmtf.org/cimi/action/restore", "href": string } ?
  ]
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Machine xmlns="http://www.dmtf.org/cimi">
  <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 quantity="xs:integer" units="xs:string"/>
  <disks href="xs:anyURI"/> ?
  <cpuArch> xs:string </cpuArch> ?
  <volumes href="xs:anyURI"/> ?
  <networkInterfaces href="xs:anyURI"/> ?
  <latestSnapshot> xs:anyURI </latestSnapshot> ?
  <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://www.dmtf.org/cimi/action/start" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/stop" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/restart" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/pause" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/suspend" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/capture" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/snapshot" href="xs:anyURI"/>
?
  <operation rel="http://www.dmtf.org/cimi/action/restore" href="xs:anyURI"/> ?
  <xs:any>*
</Machine>
```

5.14.7.1 Collections

The following describes the JSON and XML serialization of the Machine collection entities.

5.14.7.1.1 Disks

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineDisksCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/MachineDisksCollectionEntry",
      "id": string,
      "capacity": { "quantity": number, "units": string },
      "format": string,
      "attachmentPoint": string,
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/MachineDisksCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/MachineDisksCollectionEntry">
    <id> xs:anyURI </id>
    <capacity quantity="xs:integer" units="xs:string">
    <format> xs:string </format>
    <attachmentPoint> xs:string </attachmentPoint>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.14.7.1.2 Volumes

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineVolumesCollection",
  "id": string,
```

```
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/MachineVolumesCollectionEntry",
      "id": string,
      "attachmentPoint": string,
      "volume": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/MachineVolumesCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/MachineVolumesCollectionEntry">
    <id> xs:anyURI </id>
    <attachmentPoint> xs:string </attachmentPoint>
    <volume href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.14.7.1.3 NetworkInterfaces

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineNetworkInterfacesCollection",
  "id": string,
  "entries": [
    { "entityURI":
        "http://www.dmtf.org/cimi/MachineNetworkInterfacesCollectionEntry",
      "id": string,
      "addresses": { "href": string },
      "network": { "href": string },
      "vsp": { "href": string }, ?
      "state": string, ?
      "maxTransmissionUnit": number, ?
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection
    entityURI="http://www.dmtf.org/cimi/MachineNetworkInterfacesCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry
  entityURI="http://www.dmtf.org/cimi/MachineNetworkInterfacesCollectionEntry">
    <id> xs:anyURI </id>
    <addresses href="xs:anyURI"/>
```

```
    <network href="xs:anyURI"/>
    <vsp href="xs:anyURI"/> ?
    <state> xs:string </state> ?
    <maxTransmissionUnit> xs:integer </maxTransmissionUnit> ?
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.14.7.1.4 NetworkInterfaces.Addresses

JSON serialization:

```
{ "entityURI":
    "http://www.dmtf.org/cimi/MachineNetworkInterfacesAddressesCollection",
  "id": string,
  "entries": [
    { "entityURI":
   "http://www.dmtf.org/cimi/MachineNetworkInterfacesAddressesCollectionEntry",
      "id": string,
      "address": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection
entityURI="http://www.dmtf.org/cimi/MachineNetworkInterfacesAddressesCollection
"
   xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry
entityURI="http://www.dmtf.org/cimi/MachineNetworkInterfacesAddressesCollection
Entry">
    <id> xs:anyURI </id>
    <address href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.14.7.1.5 Snapshots

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineSnapshotsCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/MachineSnapshotsCollectionEntry",
      "id": string,
      "snapshot": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
```

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

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/MachineSnahostsCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/MachineSnapshotsCollectionEntry">
    <id> xs:anyURI </id>
    <snapshot href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.14.7.1.6 Meters

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineMetersCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/MachineMetersCollectionEntry",
      "id": string,
      "meter": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/MachineMetersCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/MachineMetersCollectionEntry">
    <id> xs:anyURI </id>
    <meter href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.14.7.2 Operations

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

The following custom operations are also defined:

Starting a Machine

/link@rel: http://www.dmtf.org/cimi/action/start

This operation will 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 is the virtual equivalent of powering on a physical machine. There is no restored CPU or Memory state, so the guest OS will typically perform its boot or installation tasks.

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

- **HTTP/REST Protocol**

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/start",
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

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

Stopping a Machine

/link@rel: http://www.dmtf.org/cimi/action/stop

This operation will stop, or shutdown, a Machine.

Input parameters:

- "force" - type: boolean
  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 filesystem to be made consistent (force=false).Inclusion of this 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 will be in the "STOPPED" state. Stopping a machine with force=true is the virtual equivalent of powering off a physical machine. There is no saved CPU or Memory state. Stopping a machine with force=false, results in a machine with consistent filesystems.

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/REST Protocol**

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/stop",
  "force": boolean, ?
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://www.dmtf.org/cimi">
  <action> http://www.dmtf.org/cimi/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 will be empty.

Restarting a Machine

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

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

Input parameters:

- "force" - type: boolean
  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 filesystem to be made consistent (force=false).Inclusion of this 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 will be in the "STARTED" state. Restarting a machine is the virtual equivalent of powering off, then on a physical machine. There is no restored CPU or Memory state, so the guest OS will typically perform its boot or installation tasks.

- **HTTP/REST Protocol**

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/restart",
  "force": boolean, ?
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://www.dmtf.org/cimi">
  <action> http://www.dmtf.org/cimi/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 will be empty.

Pausing a Machine

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

This operation will 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 will be in the "PAUSED" state. Pausing a machine will keep the Machine and its resources instantiated but the Machine will not be available to perform any tasks. The current state of the CPU and Memory will be retained in volatile memory.

- **HTTP/REST Protocol**

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/pause",
  "properties": { "name": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

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

Suspending a Machine

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

This operation will 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 will be in the "SUSPENDED" state. Suspending a machine will keep the Machine and its resources instantiated but the Machine will not be available to perform any tasks. The current state of the CPU and Memory will be retained in non-volatile memory.

- **HTTP/REST Protocol**

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/suspend",
  "properties": { "name": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

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

Capturing a Machine

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

This operation will create a new Machine Image from an existing Machine. This operation is defined within the Machine Image entity - see 5.14.5.1 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://www.dmtf.org/cimi/action/snapshot

This operation will create a new SNAPSHOT Machine Image from an existing Machine. This operation is defined within the Machine Image entity - see 5.14.5.1 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://www.dmtf.org/cimi/action/restore

This operation will restore a Machine from a previously created SNAPSHOT Machine Image.

Input parameters: A reference to the SNAPSHOT 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 will be in the same state that the originally snapshotted Machine.

- **HTTP/REST Protocol**

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/restore",
  "snapshot": string,
  "properties": { "name": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

```
<Action xmlns="http://www.dmtf.org/cimi">
  <action> http://www.dmtf.org/cimi/action/restore </action>
  <snapshot href="xs:anyURI"/>
  <property name="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

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

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

5.14.8 Machine Collection

A Machine Collection resource represents the collection of Machine entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MachineCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/MachineCollectionEntry",
      "id", string,
      "machine": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/MachineCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/MachineCollectionEntry">
    <id> xs:anyURI </id>
    <machine href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.8.1 Operations

Note, the "add" operation requires a MachineTemplate to be used.

Within the NetworkInterface portion of the MachineTemplate there may be a reference to an Address entity. If one is not provided then 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 created and shall be deleted by the Provider when the Machine is deleted. Additionaly, if these Provider created Address entities are disassociated from the Machine then the Provider shall delete them. If the Consumer does provide an Address entity then 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 there is no DefaultInitialState capability defined then the default value is "STOPPED".

5.14.9 Credentials Template

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

Name	CredentialsTemplate	
Type URI	http://www.dmtf.org/cimi/CredentialsTemplate	
Attribute	**Type**	**Description**

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/CredentialTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  ...
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<CredentialsTemplate xmlns="http://www.dmtf.org/cimi">
  <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>*
</CredentialsTemplate>
```

5.14.9.1 Operations

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

5.14.10 Credentials Template Collection

A Credentials Template Collection entity represents the collection of CredentialsTemplate entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/CredentialsTemplateCollection",
  "id": string,
  "entries": [
    { "entityURI":
        "http://www.dmtf.org/cimi/CredentialsTemplateCollectionEntry",
      "id", string,
      "credentialsTemplate": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/CredentialsTemplateCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry
      entityURI="http://www.dmtf.org/cimi/CredentialsTemplateCollectionEntry">
    <id> xs:anyURI </id>
    <credentialsTemplate href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.10.1 Operations

This entity supports the Read and Update operations. Creation of new Credentials Template entities is supported via a POST to the "add" operation's URI as described in section 4.3.2.1.

5.14.11 Credentials

A Credentials entity contains the information required to create the initial administrative super- user 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 entity with additional attributes to meet their requirements.

For example, a Provider might extend this entity 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	Credentials	
Type URI	http://www.dmtf.org/cimi/Credentials	
Attribute	**Type**	**Description**

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:

```
{ "entityURI": "http://www.dmtf.org/cimi/Credentials",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "operations": [
    { "rel": "edit", "href": string } ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Credentials xmlns="http://www.dmtf.org/cimi">
  <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>*
</Credentials>
```

5.14.11.1 Operations

This entity supports the Read, Update and Delete operations. Create is supported via the Credentials Collection entity.

5.14.12 Credentials Collection

A Credentials Collection entity represents the collection of Credentials entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/CredentialsCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/CredentialsCollectionEntry",
      "id", string,
      "credentials": { "href": string },
    }, +
```

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

XML serialization:

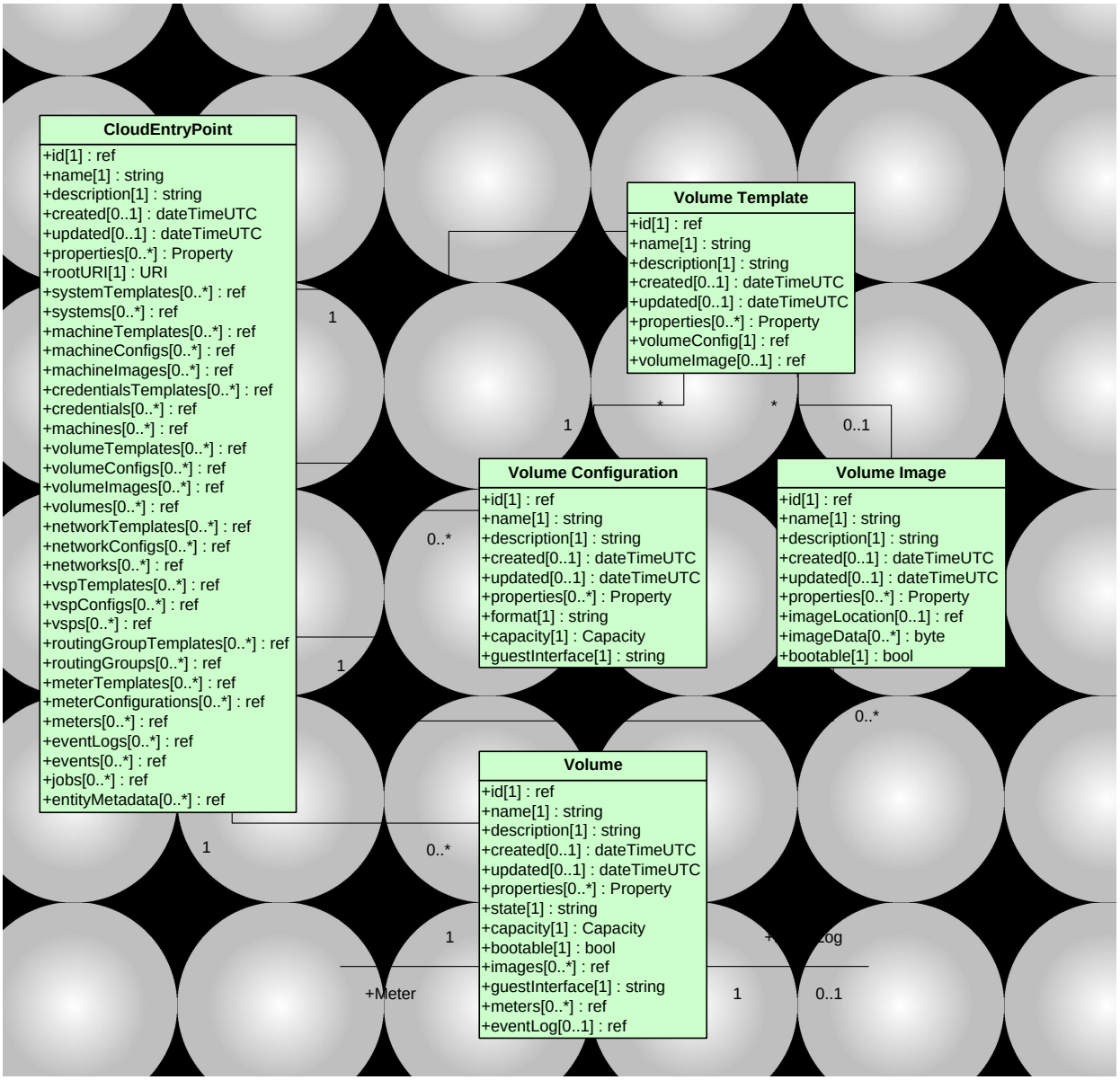
```
<Collection entityURI="http://www.dmtf.org/cimi/CredentialsCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/CredentialsCollectionEntry">
    <id> xs:anyURI </id>
    <credentials href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.14.12.1 Operations

Note, the "add" operation requires a CredentialsTemplate to be used.

5.15 Volume Entities and Relationships

The following diagram illustrates the entities involved in constructing a Volume and their relationships. Although this drawing is in the style of an Entity Relationship diagram, the use of UML is neither rigorous nor normative.



3155 **Figure 3 - Volume Entities**

3156 ### 5.15.1 Volume Template

3157 This entity captures the configuration values for realizing a Volume. A Volume Template may be used to
3158 create multiple Volumes.

Name	VolumeTemplate	
Type URI	http://www.dmtf.org/cimi/VolumeTemplate	
Attribute	**Type**	**Description**
volumeConfig	*ref*	A reference to the Volume Configuration that will be used to create a Volume from this Volume Template. **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

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VolumeTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "volumeConfig": { "href": string },
  "volumeImage": { "href": string },
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<VolumeTemplate xmlns="http://www.dmtf.org/cimi">
  <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"/>
  <volumeImage href="xs:anyURI"/>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</VolumeTemplate>
```

5.15.1.1 Operations

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

5.15.2 Volume Template Collection

A Volume Template Collection entity represents the collection of VolumeTemplate entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VolumeTemplateCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/VolumeTemplateCollectionEntry",
      "id", string,
      "volumeTemplate": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/VolumeTemplateCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/VolumeTemplateCollectionEntry">
    <id> xs:anyURI </id>
    <volumeTemplate href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.15.2.1 Operations

This entity supports the Read and Update operations. Creation of new Volume Template entities is supported via a POST to the "add" operation'sURI as described in section 4.3.2.1.

5.15.3 Volume Configuration

The Volume Configuration entity 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://www.dmtf.org/cimi/VolumeConfiguration	
Attribute	**Type**	**Description**
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	*<unnamed structure>*	The default size, when limited, of the Volume created from this Volume Configuration. This attribute is an (unnamed) structure that has the following sub-attributes.

<table>
<tr><td></td><td></td><td>
<table>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>quantity</td><td>integer</td><td>A numerical quantity expressed as an integer.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>units</td><td>string</td><td>An enumerated value that expresses the unit of measurement used. Allowable values are byte, kilobyte, megabyte, gigabyte, terabyte, petabyte, exabyte, zettabyte, and yottabyte.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
</table>
<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write
</td></tr>
<tr><td>guestInterface</td><td>string</td><td>This property indicates the interface that will be offered to Machine instances by Volumes created from this Volume Configuration.

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

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VolumeConfiguration",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "format": string,
  "capacity": { "quantity": number, "units": string },
  "guestInterface": string,
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<VolumeConfiguration xmlns="http://www.dmtf.org/cimi">
  <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> *
  <format> xs:string </format>
  <capacity quantity="xs:integer" units="xs:string"/>
  <guestInterface> xs:string </guestInterface>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</VolumeConfiguration>
```

5.15.3.1 Operations

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

5.15.4 Volume Configuration Collection

A Volume Configuration Collection entity represents the collection of Volume Configuration entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VolumeConfigurationCollection",
  "id": string,
  "entries": [
    { "entityURI":
        "http://www.dmtf.org/cimi/VolumeConfigurationCollectionEntry",
      "id", string,
      "volumeConfiguration": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/VolumeConfigurationCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry
      entityURI="http://www.dmtf.org/cimi/VolumeConfigurationCollectionEntry">
    <id> xs:anyURI </id>
    <volumeConfiguration href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.15.4.1 Operations

This entity supports the Read and Update operations. Creation of new Volume Image entities is supported via a POST to the "add" operations' URI as described in section 4.3.2.1.

5.15.5 Volume Image

This entity represents an image that could be placed on a pre-loaded volume.

Name	VolumeImage	
Type URI	http://www.dmtf.org/cimi/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. Allowable action when in this state is: **delete**. **AVAILABLE**: The VolumeImage is available and ready for use. Allowable action when in this state is: **delete**. **DELETING**: The VolumeImage is in the process of being deleted. Allowable action when in this state is: **delete**. **ERROR**: The Provider has detected an error in the VolumeImage. Allowable action when in this state is: **delete**. Providers may define additional values. **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 mandtory ; 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 entity in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VolumeImage",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": 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://www.dmtf.org/cimi">
  <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>
```

5.15.5.1 Operations

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

5.15.6 Volume Image Collection

A Volume Image Collection entity represents the collection of VolumeImage entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VolumeImageCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/VolumeImageCollectionEntry",
      "id", string,
      "volumeImage": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/VolumeImageCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/VolumeImageCollectionEntry">
    <id> xs:anyURI </id>
    <volumeImage href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.15.6.1 Operations

This entity supports the Read and Update operations. Creation of new Volume Image entities is supported via a POST to the "add" operation's URI as described in section 4.3.2.1.

During the creation of a new Volume Image entity, if the "imageLocation" attribute refers to an existing Volume then this shall be interpreted as a request to create a snapshot of the Volume. Once completed, the "imageLocation" attribute of the new Volume Image entity shall not refer to the original Volume, instead it shall refer to a static copy of the Volume. Additionally, the "image" attribute of the referenced Volume entity shall be updated to include a reference to this new Volume Image entity. During this process the Provider may put the Volume into a "CAPTURING" state if necessary.

5.15.7 Volume

A Volume represents storage at either the block or file-system level. Volumes can be attached to Machines. Once attached, Volumes can be accessed by processes on that Machine.

<table>
<tr><td>Name</td><td colspan="2">Volume</td></tr>
<tr><td>Type URI</td><td colspan="2">http://www.dmtf.org/cimi/Volume</td></tr>
<tr><td>Attribute</td><td>Type</td><td>Description</td></tr>
<tr><td>state</td><td>string</td><td>Indicates the operational state of the Volume.

Allowable values include:

CREATING: The Volume is in the process of being created. Allowable action when in this state is: delete.

AVAILABLE: The Volume is available and ready for use. Allowable action when in this state is: delete.

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. Allowable action when in this state is: delete.

ERROR: The Provider has detected an error in the Volume. Allowable action when in this state is: delete.

Providers may define additional values.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-only</td></tr>
<tr><td>capacity</td><td><unnamed structure></td><td>The maximum size, when limited, of the Volume.

When this value is increased, the Volume can contain more data. Decreasing this value may require evaluations.

This attribute is an (unnamed) structure that has the following sub-attributes.
<table>
<tr><td>Attribute</td><td>Type</td><td>Description</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td>quantity</td><td>integer</td><td>A numerical quantity expressed as an integer.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>units</td><td>string</td><td>An enumerated value that expresses the unit of measurement used. Allowable values are byte, kilobyte, megabyte, gigabyte, terabyte, petabyte, exabyte, zettabyte, and yottabyte.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
</table>
<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>bootable</td><td>boolean</td><td>This property indicates whether this Volume is bootable.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>images</td><td>collection[image]</td><td>A list of references to Volume Images that represent snapshots taken from the Volume.

<u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-only</td></tr>
<tr><td>guestInterface</td><td>string</td><td>This property indicates the interface offered to a Machine instance to gain access to the storage contents.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>meters</td><td>collection[meter]</td><td>A list of references to Meters monitored for this Volume.

<u>Constraints:</u>
Provider: support optional ; mutable
Consumer: support optional ; read-only</td></tr>
<tr><td>eventLog</td><td>ref</td><td>A reference to the EventLog of this Volume.

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

3376 The following describes the serialization of the entity in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Volume",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "state": string,
  "capacity": { "quantity": number, "units": string },
  "bootable": boolean,
  "images": { "href": string }, ?
  "guestInterface": 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://www.dmtf.org/cimi">
  <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>
  <capacity quantity="xs:integer" units="xs:string"/>
  <bootable> xs:boolean </bootable>
  <images href="xs:anyURI"/> ?
  <guestInterface> xs:string </guestInterface>
  <meters href="xs:anyURI"/> ?
  <eventLog href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</Volume>
```

5.15.7.1 Collections

The following describes the JSON and XML serialization of the Volume collection entities.

5.15.7.1.1 Images

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VolumeImagesCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/VolumeImagesCollectionEntry",
      "id": string,
      "image": { "href": string },
      "operations": [
```

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

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/VolumeImagesCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/VolumeImagesCollectionEntry">
    <id> xs:anyURI </id>
    <image href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.15.7.1.2 Meters

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VolumeMetersCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/VolumeMetersCollectionEntry",
      "id": string,
      "meter": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/VolumeMetersCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/VolumeMetersCollectionEntry">
    <id> xs:anyURI </id>
    <meter href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.15.7.2 Operations

This entity supports the Read, Update and Delete operations. Create is supported via the Volume Collection entity.

5.15.8 Volume Collection

A Volume Collection entity represents the collection of Volumes within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VolumeCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/VolumeCollectionEntry",
      "id", string,
      "volume": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/VolumeCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/VolumeCollectionEntry">
    <id> xs:anyURI </id>
    <volume href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.15.8.1 Operations

Note, the "add" operation requires a VolumeTemplate to be used.

5.16 Network Entities and Relationships

The following diagram illustrates the entities involved in constructing Networks and their Virtual Switch Ports (VSPs) and their relationships. Although this drawing is in the style of an Entity Relationship diagram, the use of UML is neither rigorous nor normative.

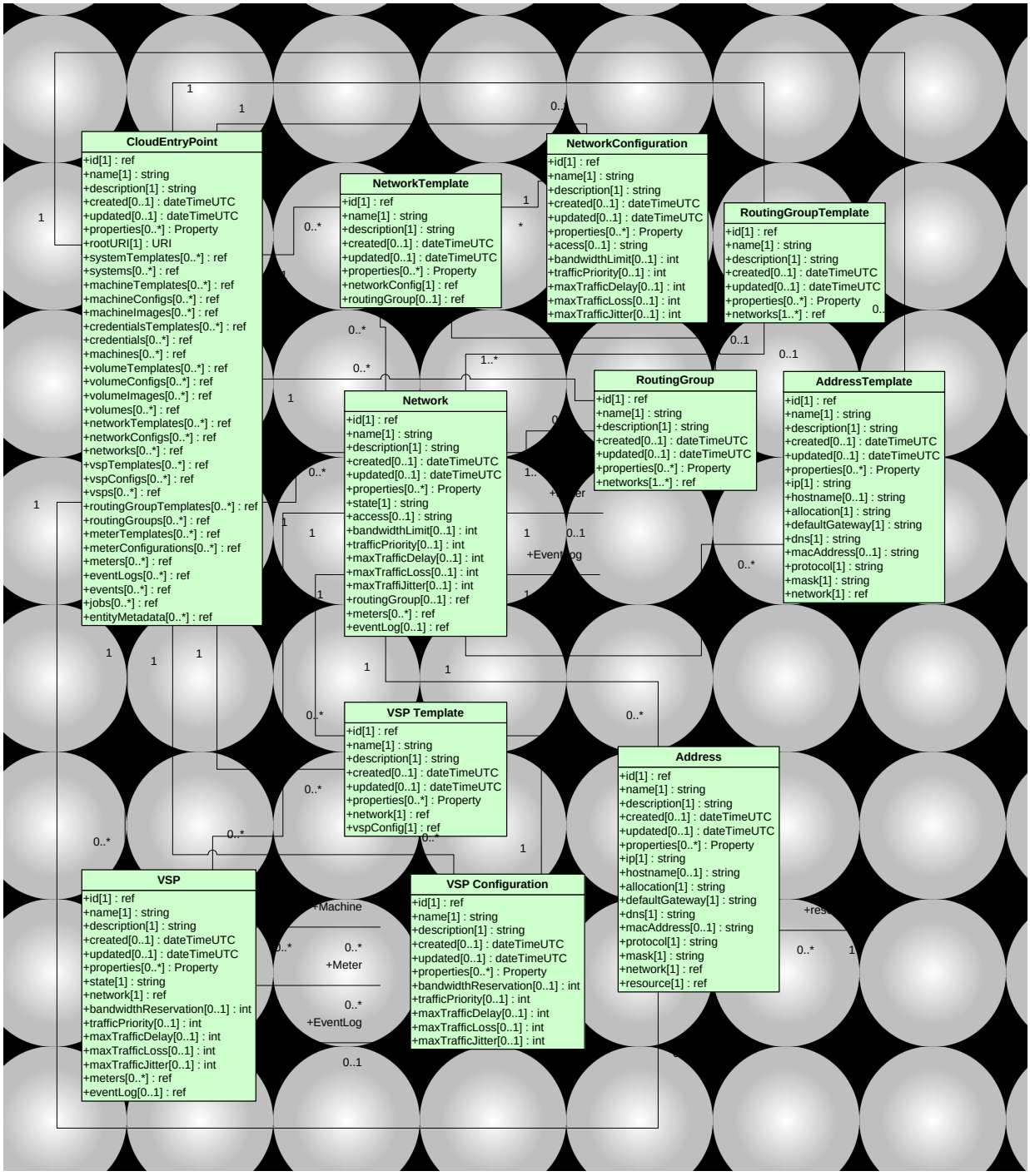


Figure 4 - Network Entities

5.16.1 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://www.dmtf.org/cimi/NetworkTemplate

Attribute	Type	Description
networkConfig	*ref*	A reference to the Network Configuration that will be used to create a Network from this Network Template. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
routingGroup	*ref*	A reference to a RoutingGroup in which this Network will be a part. Note that Networks route to themselves, therefore this attribute will only appear in cases where the Network that will be created from this template routes to one or more additional Networks. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/NetworkTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "networkConfig": { "href": string },
  "routingGroup": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<NetworkTemplate xmlns="http://www.dmtf.org/cimi">
  <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> *
  <networkConfig href="xs:anyURI"/>
  <routingGroup href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</NetworkTemplate>
```

5.16.1.1 Operations

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

5.16.2 Network Template Collection

A Network Template Collection entity represents the collection of NetworkTemplates within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/NetworkTemplateCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/NetworkTemplateCollectionEntry",
      "id", string,
      "networkTemplate": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/NetworkTemplateCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/NetworkTemplateCollectionEntry">
    <id> xs:anyURI </id>
    <networkTemplate href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.2.1 Operations

This entity supports the Read and Update operations. Creation of new Network Template entities is supported via a POST to the "add" operation's URI as described in section 4.3.2.1.

5.16.3 Network Configuration

The set of configuration values representing the information needed to create a Network with certain characteristics.

Name	NetworkConfiguration	
Type URI	http://www.dmtf.org/cimi/NetworkConfiguration	
Attribute	**Type**	**Description**
access	*string*	An indicator of whether or not the Network will be a Public or Private network. Valid values are either "PUBLIC" or "PRIVATE". **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support optional ; read-write
bandwidthLimit	*integer*	Maximum allowable bandwidth.

		Constraints: **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
trafficPriority	*integer*	Indicates priority of traffic on this network. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficDelay	*integer*	The requested maximum delay for end to end transmission specified in nanoseconds (i.e. latency). **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficLoss	*integer*	The requested maximum percentage traffic loss for end to end transmission as expressed as an integer within the range of 0 and 100. The percentage of traffic lost in the transmission traffic. A value of zero indicates that a lossless transmission is requested. A value of 100 indicates a best effort transmission. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficJitter	*integer*	The requested maximum jitter for end to end transmission as specified as an integer value when traffic is packetized. The variation between packets arriving specified in nanoseconds. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/NetworkConfiguration",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "access": string, ?
  "bandwidthLimit": number, ?
  "trafficPriority": number, ?
  "maxTrafficDelay": number, ?
  "maxTrafficLoss": number, ?
  "maxTrafficJitter": number, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<NetworkConfiguration xmlns="http://www.dmtf.org/cimi">
  <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> *
  <access> xs:string </access> ?
  <bandwidthLimit> xs:string <bandwidthLimit> ?
  <trafficPriority> xs:integer </trafficPriority> ?
  <maxTrafficDelay> xs:integer </maxTrafficDelay> ?
  <maxTrafficLoss> xs:integer </maxTrafficLoss> ?
  <maxTrafficJitter> xs:integer </maxTrafficJitter> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</NetworkConfiguration>
```

5.16.3.1 Operations

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

5.16.4 Network Configuration Collection

A Network Configuration Collection entity represents the collection of Network Configurations within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/NetworkConfigurationCollection",
  "id": string,
  "entries": [
    { "entityURI":
        "http://www.dmtf.org/cimi/NetworkConfigurationCollectionEntry",
      "id", string,
      "networkConfiguration": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/NetworkConfigurationCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry
      entityURI="http://www.dmtf.org/cimi/NetworkConfigurationCollectionEntry">
    <id> xs:anyURI </id>
    <networkConfiguration href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.4.1 Operations

This entity supports the Read and Update operations. Creation of new Network Configuration entities is supported via a POST to the "add" operation's URI as described in section 4.3.2.1.

5.16.5 Network

A Network is a realized entity that represents an abstraction of a layer 2 broadcast domain.

It is recommended that Networks not be members of a RoutingGroup containing Networks with varying 'access' attributes. This might (for example) lead to a Network with a value of 'private' for its 'access' attribute being publicly routable because it is a member of a routingGroup containing Networks that have a value of 'public' for their access attribute.

Name	Network	
Type URI	http://www.dmtf.org/cimi/Network	
Attribute	**Type**	**Description**
state	*string*	Indicates the operational state of the System. Allowable values include: **CREATING**: The Network is in the process of being created. Allowable action when in this state is: **delete**. **STARTING**: The Network is in the process of being started. Allowable actions when in this state are: **stop** and **delete**. **STARTED**: The Network is available and ready for use. Allowable actions when in this state are: **suspend**, **stop** and **delete**. **SUSPENDING**: The Network is in the process of being suspended. Allowable actions when in this state are: **suspend**, **stop** and **delete**. **SUSPENDED**: The Network is suspended. Allowable actions when in this state are: **start**, **stop** and **delete**. **STOPPING**: The Network is in the process of being stopped. Allowable actions when in this state are: **stop** and **delete**. **STOPPED**:The Network is stopped and not available for use. Allowable actions when in this state are: **start** and **delete**. **DELETING**: The Network is in the process of being deleted. Allowable action when in this state is: **delete**. **ERROR**: The Provider has detected an error in the Network. Allowable action when in this state is: **delete**. Providers may define additional values. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-only
access	*string*	An indicator of whether or not the Machine entity has access to a Public or Private network. An indication of Public represents an open and Internet routable network. An indication of Private identifies a local non-routed

		network. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support optional ; read-write
bandwidthLimit	*integer*	Maximum allowable bandwidth. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
trafficPriority	*integer*	Indicates priority of traffic on this network. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficDelay	*integer*	The requested maximum delay for end to end transmission specified in nanoseconds (i.e. latency). **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficLoss	*integer*	The requested maximum percentage traffic loss for end to end transmission as expressed as an integer within the range of 0 and 100. The percentage of traffic lost in the transmission traffic. A value of zero indicates that a lossless transmission is requested. A value of 100 indicates a best effort transmission. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficJitter	*integer*	The requested maximum jitter for end to end transmission as specified as an integer value when traffic is packetized. The variation between packets arriving specified in nanoseconds. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
routingGroup	*ref*	A reference to a RoutingGroup of which this Network is a part. Note that Networks route to themselves, therefore this attribute will only appear in cases where the Network routes to one or more additional Networks. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
meters	*collection[meter]*	A list of references to 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 entity in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Network",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "state": string,
  "access": string, ?
  "bandwidthLimit": number, ?
  "trafficPriority": number, ?
  "maxTrafficDelay": number, ?
  "maxTrafficLoss": number, ?
  "maxTrafficJitter": number, ?
  "routingGroup": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/start", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/suspend", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/stop", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Network xmlns="http://www.dmtf.org/cimi">
  <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>
  <access> xs:string </access> ?
  <bandwidthLimit> xs:integer </bandwidthLimit> ?
  <trafficPriority> xs:integer </trafficPriority> ?
  <maxTrafficDelay> xs:integer </maxTrafficDelay> ?
  <maxTrafficLoss> xs:integer </maxTrafficLoss> ?
  <maxTrafficJitter> xs:integer </maxTrafficJiffer> ?
  <routingGroup 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://www.dmtf.org/cimi/action/start" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/suspend" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/stop" href="xs:anyURI"/> ?
  <xs:any>*
</Network>
```

5.16.5.1 Collections

The following describes the JSON and XML serialization of the Network collection entities.

5.16.5.1.1 Meters

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/NetworkMetersCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/NetworkMetersCollectionEntry",
      "id": string,
      "meter": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/NetworkMetersCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/NetworkMetersCollectionEntry">
    <id> xs:anyURI </id>
    <meter href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.16.5.2 Operations

This entity supports the Read, Update and Delete operations. Create is supported via the Network Collection entity.

The following custom operations are also defined:

Starting a Network

/link@rel: http://www.dmtf.org/cimi/action/start

This operation will 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.

NOTE: need a better description - Steve?

- **HTTP/REST Protocol**

To start a Network a POST is sent to the "http://www.dmtf.org/cimi/action/start" URI of the Network; where the HTTP request body SHALL be as described below.

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/start",
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

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

Suspending a Network

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

This operation will suspend a Network.

Input parameters: None.

Output parameters: None.

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

Upon successful completion of this operation the Network shall be in the "SUSPENDED" state.

NOTE: need a better description - Steve?

- **HTTP/REST Protocol**

To suspend a Network a POST is sent to the "http://www.dmtf.org/cimi/action/suspend" URI of the Network; where the HTTP request body SHALL be as described below.

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/suspend",
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

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

Stopping a Network

/link@rel: http://www.dmtf.org/cimi/action/stop

This operation will stop a Network.

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.

NOTE: need a better description - Steve?

- **HTTP/REST Protocol**

To stop a Network a POST is sent to the "http://www.dmtf.org/cimi/action/stop" URI of the Network; where the HTTP request body SHALL be as described below.

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/stop",
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

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

5.16.6 Network Collection

A Network Collection entity represents the collection of Networks within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/NetworkCollection",
```

```
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/NetworkCollectionEntry",
      "id", string,
      "network": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/NetworkCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/NetworkCollectionEntry">
    <id> xs:anyURI </id>
    <network href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.6.1 Operations

Note, the "add" operation requires a SystemTemplate to be used.

5.16.7 VSP (Virtual Switch Port) Template

The VSP Template is a set of Configuration values for realizing a VSP. A VSP Template may be used to create multiple VSPs.

Name	VSPTemplate	
Type URI	http://www.dmtf.org/cimi/VSPTemplate	
Attribute	**Type**	**Description**
network	*ref*	A reference to the network to be associated with this VSP. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
vspConfig	*ref*	A reference to the VSP Configuration that will be used to create a VSP from this VSP Template. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VSPTemplate",
  "id": string,
  "name": string, ?
```

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

XML media type: application/xml

XML serialization:

```
<VSPTemplate xmlns="http://www.dmtf.org/cimi">
  <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"/>
  <vspConfig href="xs:anyURI"/>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</VSPTemplate>
```

5.16.7.1 Operations

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

5.16.8 VSP (Virtual Switch Port) Template Collection

A VSP Template Collection entity represents the collection of VSP Templates within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VSPTemplateCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/VSPTemplateCollectionEntry",
      "id", string,
      "vspTemplate": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/VSPTemplateCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/VSPTemplateCollectionEntry">
    <id> xs:anyURI </id>
    <vspTemplate href="xs:anyURI"/>
```

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

5.16.8.1 Operations

This entity supports the Read and Update operations. Creation of new VSP Template entities is supported via a POST to the "add" operation's URI as described in section 4.3.2.1.

5.16.9 VSP (Virtual Switch Port) Configuration

The set of configuration values representing the information needed to create a VSP with certain characteristics.

Name	VSPConfiguration	
Type URI	http://www.dmtf.org/cimi/VSPConfiguration	
Attribute	**Type**	**Description**
bandwidthReservation	*integer*	Minimum Bandwidth requirements. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
trafficPriority	*integer*	Indicates priority of traffic on this network. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficDelay	*integer*	The requested maximum delay for end to end transmission specified in nanoseconds (i.e. latency). **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficLoss	*integer*	The requested maximum percentage traffic loss for end to end transmission as expressed as an integer within the range of 0 and 100. The percentage of traffic lost in the transmission traffic. A value of zero indicates that a lossless transmission is requested. A value of 100 indicates a best effort transmission. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficJitter	*integer*	The requested maximum jitter for end to end transmission as specified as an integer value when traffic is packetized. The variation between packets arriving specified in nanoseconds. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VSPConfiguration",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "bandwidthReservation": number, ?
  "trafficPriority": number, ?
  "maxTrafficDelay": number, ?
  "maxTrafficLoss": number, ?
  "maxTrafficJitter": number,?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<VSPConfiguration xmlns="http://www.dmtf.org/cimi">
  <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> *
  <bandwidthReservation> xs:integer </bandwidthReservation> ?
  <trafficPriority> xs:integer </trafficPriority> ?
  <maxTrafficDelay> xs:integer </maxTrafficDelay> ?
  <maxTrafficLoss> xs:integer </maxTrafficLoss> ?
  <maxTrafficJitter> xs:integer </maxTrafficJitter> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</VSPConfiguration>
```

5.16.9.1 Operations

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

5.16.10 VSP (Virtual Switch Port) Configuration Collection

A VSP Configuration Collection entity represents the collection of VSP Configurations within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VSPConfigurationCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/VSPConfigurationCollectionEntry",
      "id", string,
      "vspConfiguration": { "href": string },
    }, +
```

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

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/VSPConfigurationCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/VSPConfigurationCollectionEntry">
    <id> xs:anyURI </id>
    <vspConfiguration href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.10.1 Operations

This entity supports the Read and Update operations. Creation of new VSP Configuration entities is supported via a POST to the "add" operation's URI as described in section 4.3.2.1.

5.16.11 VSP (Virtual Switch Port)

A VSP represents the connection parameters of a network port.

Name	VSP	
Type URI	http://www.dmtf.org/cimi/VSP	
Attribute	**Type**	**Description**
state	*string*	Indicates the operational state of the VSP. Allowable values include: **CREATING**: The VSP is in the process of being created. Allowable action when in this state is: **delete**. **STARTED**: The VSP is available (enabled) and ready for use. Allowable actions when in this state are: **stop** and **delete**. **STOPPED**:The VSP is stopped(disabled) and not available for use. Allowable actions when in this state are: **start** and **delete**. **DELETING**: The VSP is in the process of being deleted. Allowable action when in this state is: **delete**. **ERROR**: The Provider has detected an error in the VSP. Allowable action when in this state is: **delete**. Providers may define additional values. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-only
network	*ref*	A reference to the network associated with this VSP. **Constraints:**

		Provider: support mandatory ; mutable **Consumer:** support mandatory ; read-write
bandwidthReservation	*integer*	Minimum Bandwidth requirements. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
trafficPriority	*integer*	Indicates priority of traffic on this VSP. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficDelay	*integer*	The requested maximum delay for end to end transmission specified in nanoseconds (i.e. latency). **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficLoss	*integer*	The requested maximum percentage traffic loss for end to end transmission as expressed as an integer within the range of 0 and 100. The percentage of traffic lost in the transmission traffic. A value of zero indicates that a lossless transmission is requested. A value of 100 indicates a best effort transmission. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
maxTrafficJitter	*integer*	The requested maximum jitter for end to end transmission as expressed as an integer value when traffic is packetized. The variation between packets arriving specified in nanoseconds. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
meters	*collection[meter]*	A list of references to Meters monitored for this VSP. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only
eventLog	*ref*	A reference to the EventLog of this VSP. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-only

4004 The following describes the serialization of the entity in both JSON and XML:

4005 **JSON media type:** application/json

4006 **JSON serialization:**

4007
```
{ "entityURI": "http://www.dmtf.org/cimi/VSP",
```

```
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "network": { "href": string },
  "state": string, ?
  "bandwidthLimit": number, ?
  "trafficPriority": number, ?
  "maxTrafficDelay": number, ?
  "maxTrafficLoss": number, ?
  "maxTrafficJitter": number, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/start", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/stop", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<VSP xmlns="http://www.dmtf.org/cimi">
  <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">
  <state> xs:string </state> ?
  <bandwidthLimit> xs:integer </bandwidthLimit> ?
  <trafficPriority> xs:integer </trafficPriority> ?
  <maxTrafficDelay> xs:integer </maxTrafficDelay> ?
  <maxTrafficLoss> xs:integer </maxTrafficLoss> ?
  <maxTrafficJitter> xs:integer </maxTrafficJiffer> ?
  <meters href="xs:anyURI"/> ?
  <eventLog" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/start" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/stop" href="xs:anyURI"/> ?
  <xs:any>*
</VSP>
```

5.16.11.1 Collections

The following describes the JSON and XML serialization of the VSP collection entities.

5.16.11.1.1 Meters

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VSPMetersCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/VSPMetersCollectionEntry",
```

```
      "id": string,
      "meter": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/VSPMetersCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/VSPMetersCollectionEntry">
    <id> xs:anyURI </id>
    <meter href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.16.11.2 Operations

This entity supports the Read, Update and Delete operations. Create is supported via the VSP Collection entity.

The following custom operations are also defined:

Starting a VSP

/link@rel: http://www.dmtf.org/cimi/action/start

This operation will start a VSP.

Input parameters: None.

Output parameters: None.

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

NOTE: need a better description - Steve?

- **HTTP/REST Protocol**

To start a VSP a POST is sent to the "http://www.dmtf.org/cimi/action/start" URI of the VSP; where the HTTP request body SHALL be as described below.

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/start",
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

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

Stopping a VSP

/link@rel: http://www.dmtf.org/cimi/action/stop

This operation will stop a VSP.

Input parameters: None.

Output parameters: None.

Upon successful completion of this operation the VSP shall be in the "STOPPED" state.

NOTE: need a better description - Steve?

- **HTTP/REST Protocol**

To stop a VSP a POST is sent to the "http://www.dmtf.org/cimi/action/stop" URI of the VSP; where the HTTP request body SHALL be as described below.

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/stop",
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

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

5.16.12 VSP (Virtual Switch Port) Collection

A VSP Collection entity represents the collection of VSPs within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/VSPCollection",
  "id": string,
  "entries": [
```

```
    { "entityURI": "http://www.dmtf.org/cimi/VSPCollectionEntry",
      "id", string,
      "vsp": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/VSPCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/VSPCollectionEntry">
    <id> xs:anyURI </id>
    <vsp href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.12.1 Operations

Note, the "add" operation requires a SystemTemplate to be used.

5.16.13 Address Template

This entity captures the configuration values for realizing an Address. An Address Template may be used to create multiple Addresses.

Name	AddressTemplate	
Type URI	http://www.dmtf.org/cimi/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*	DNS resolvable name associated with this network interface. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
allocation	*string*	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 to a router that serves other networks. **Constraints:** **Provider:** support mandatory ; mutable

		Consumer: support mandatory ; read-write
dns	*string*	The IP address of the Domain Name Service from host name to IP resolution. **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
protocol	*string*	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 mandatory ; mutable **Consumer:** support mandatory ; read-write
network	*ref*	A reference to the Network to which this Address will be associated. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/AddressTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "ip": string,
  "hostname": string, ?
  "allocation": string,
  "defaultGateway": string,
  "dns": string,
  "macAddress": 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://www.dmtf.org/cimi">
  <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>
  <macAddress> xs:string </macAddress> ?
  <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>
```

5.16.13.1 Operations

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

5.16.14 Address Template Collection

An Address Template Collection entity represents the collection of Address Template entities within a Provider Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/AddressTemplateCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/AddressTemplateCollectionEntry",
      "id", string,
      "addressTemplate": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/AddressTemplateCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/AddressTemplateCollectionEntry">
    <id> xs:anyURI </id>
```

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

5.16.14.1 Operations

This entity supports the Read and Update operations. Creation of new Address Template entities is supported via a POST to the "addLink" URI as described in section 4.3.2.1.

5.16.15 Address

An Address represents an IP address, and its associated metdata, for a particular Network. When a Consumer creates an Address entity 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 manually created by Consumers shall not be implicitly deleted when the resource using it is deleted (e.g. Machines) as they are expected to have a lifetime that goes beyond those resources. Addresses 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 using that Address is deleted or when the Address becomes disassociated with the resource.

Addresses created by Providers may be converted to ones that are under the Consumer's control (i.e. will not be deleted until explicated requested by Consumers) by chaging the "allocation" attribute from "dynamic" to "static", if supported by Providers.

Name	Address	
Type URI	http://www.dmtf.org/cimi/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*	DNS resolvable name associated with this network interface. **Constraints:** **Provider:** support optional ; mutable **Consumer:** support optional ; read-write
allocation	*string*	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 to a router that serves other networks. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
dns	*string*	The IP address of the Domain Name Service from host name to IP resolution.

		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
protocol	*string*	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 mandatory ; mutable **Consumer:** support mandatory ; read-write
network	*ref*	A reference to the Network to 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 entity in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Address",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "ip": string,
  "hostname": string, ?
  "allocation": string,
  "defaultGateway": string,
  "dns": string,
  "macAddress": 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://www.dmtf.org/cimi">
  <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>
  <macAddress> xs:string </macAddress> ?
  <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>
```

5.16.15.1 Operations

This entity supports the Read, Update and Delete operations. Create is supported via the Address Collection entity.

5.16.16 Address Collection

An Address Collection entity represents the collection of Addresses within a Provider that are owned/managed by the Consumer Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/AddressCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/AddressCollectionEntry",
      "id", string,
      "address": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/AddressCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/AddressCollectionEntry">
    <id> xs:anyURI </id>
    <address href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.16.1 Operations

Note, the "add" operation requires a SystemTemplate to be used.

5.16.17 Routing Group Template

This entity captures the configuration values for realizing a RoutingGroup. A Routing Group Template may be used to create multiple RoutingGroup.

Name	RoutingGroupTemplate	
Type URI	http://www.dmtf.org/cimi/RoutingGroupTemplate	
Attribute	**Type**	**Description**
networks	*network[]*	An array of references to the networks in this Routing Group. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-only

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/RoutingGroupTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "networks": [
    { "href": string }, +
   ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<RoutingGroupTemplate xmlns="http://www.dmtf.org/cimi">
```

```
  <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>*
</RoutingGroupTemplate>
```

5.16.17.1 Operations

This entity supports the Read, Update and Delete operations. Create is supported via the Routing Group Template Collection entity.

5.16.18 Routing Group Template Collection

A Routing Group Template Collection entity represents the collection of Routing Group Template entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/RoutingGroupTemplateCollection",
  "id": string,
  "entries": [
    { "entityURI":
        "http://www.dmtf.org/cimi/RoutingGroupTemplateCollectionEntry",
      "id", string,
      "routingGroupTemplate": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/RoutingGroupTemplateCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry
      entityURI="http://www.dmtf.org/cimi/RoutingGroupTemplateCollectionEntry">
    <id> xs:anyURI </id>
    <routingGroupTemplate href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</RoutingGroupTemplateCollection>
```

5.16.18.1 Operations

This entity supports the Read and Update operations. Creation of new Routing Group Template entities is supported via a POST to the "add" operation's URI as described in section 4.3.2.1.

5.16.19 Routing Group

A Routing Group represents a collection of Networks that route to each other.

It is recommended that RoutingGroups not contain any Networks with varying 'access' attributes. This might, for example, lead to a Network with a value of 'private' for its 'access' attribute being publicly routable because it is a member of a routingGroup containing Networks that have a value of 'public' for their access attribute.

Providers shall not allow two Networks to be routable to each other unless they are explicitly connected by being part of a common RoutingGroup.

Name	RoutingGroup	
Type URI	http://www.dmtf.org/cimi/RoutingGroup	
Attribute	**Type**	**Description**
networks	*collection[network]*	An array of references to the networks in this Routing Group. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-only

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/RoutingGroup",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "networks": [
    { "href": string }, +
   ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<RoutingGroup xmlns="http://www.dmtf.org/cimi">
  <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>*
</RoutingGroup>
```

5.16.19.1 Collections

The following describes the JSON and XML serialization of the RoutingGroup collection entities.

5.16.19.1.1 Networks

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/RoutingGroupNetworksCollection",
  "id": string,
  "entries": [
    { "entityURI":
        "http://www.dmtf.org/cimi/RoutingGroupNetworksCollectionEntry",
      "id": string,
      "network": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
      ] ?
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/RoutingGroupNetworksCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry
      entityURI="http://www.dmtf.org/cimi/RoutingGroupNetworksCollectionEntry">
    <id> xs:anyURI </id>
    <network href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
</Collection>
```

5.16.19.2 Operations

This entity supports the Read, Update and Delete operations. Create is supported via the RoutingGroup Collection entity.

5.16.20 Routing Group Collection

A Routing Group Collection entity represents the collection of Routing Groups within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/RoutingGroupCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/RoutingGroupCollectionEntry",
      "id", string,
      "routingGroup": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

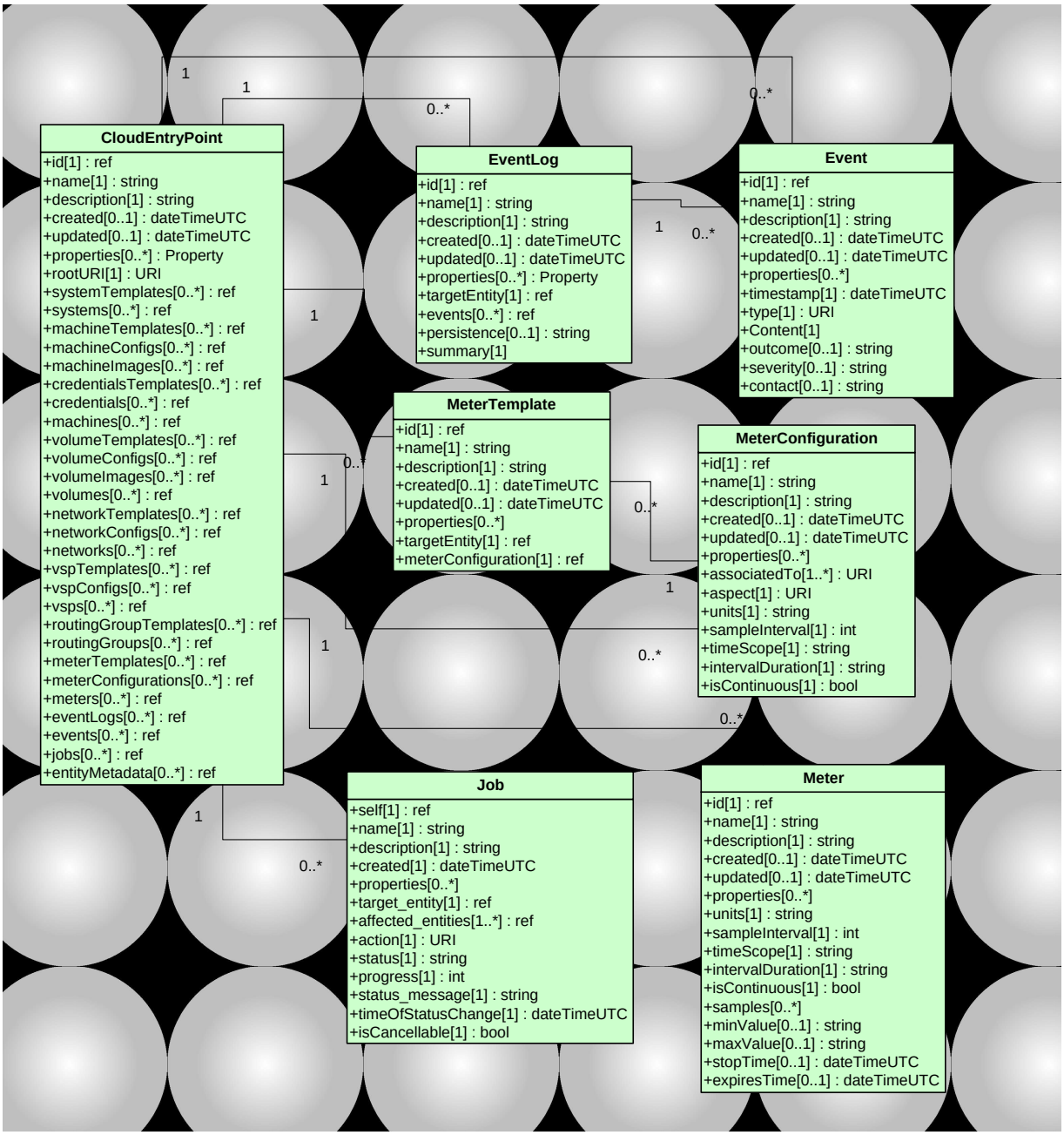
```
<Collection entityURI="http://www.dmtf.org/cimi/RoutingGroupCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/RoutingGroupCollectionEntry">
    <id> xs:anyURI </id>
    <routingGroup href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.16.20.1 Operations

Note, the "add" operation requires a SystemTemplate to be used.

5.17 Monitoring Entities and Relationships

The following diagram illustrates the entities involved in tracking the progress of operations as well as metering and monitoring the status of other entities. Although this drawing is in the style of an Entity Relationship diagram, the use of UML is neither rigorous nor normative.



4522 **Figure 5 - Monitoring Entities**

4523 ### 5.17.1 Job

4524 This entity represents a process (i.e. a sequence of one or more operations directed to accomplish a
4525 specific goal) performed by the Provider.

4526 If a Provider supports exposing Job entities to Consumers then each request from a Consumer that would
4527 result in a change to the environment MUST result in a Job entity being created and an absolute URI
4528 reference to that Job entity MUST be made available to the requesting Consumer. Providers MAY create
4529 additional Job entities for Provider initiated operations if the Provider chooses to expose these Jobs to
4530 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 affectedEntities 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 entities impacted by the Job in the "affectedEntities" attribute, thus allowing Consumers an opportunity to examine the state of each resource themselves. In cases where a resource has been deleted then references to those resources shall not appear in the "affectedEntities" attribute.

The Job entity allows for a 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 then the top-most Job entity 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 entity to reference all affected resource in its "affectedEntities" attribute - Consumer will need to traverse the entire set of nested Jobs to determine the complete list of entities impacted by the Jobs.

Name	Job	
Type URI	http://www.dmtf.org/cimi/Job	
Attribute	**Type**	**Description**
state	*string*	Indicates the state of the process associated with this operation. Allowable values include: **QUEUED**: Indicates that the operation has not yet begun processing. Allowable actions when in this state are: **stop**. **RUNNING**: Indicates that the operation is still being executed. Allowable action when in this state is: **stop**. **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. Allowable action when in this state is: **stop**. **STOPPED**: Indicates that the operation was stopped before completion. **STOPPING** and **STOPPED** states are optional and Providers may choose to support them or not. Providers may define additional values. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-only
targetEntity	*ref*	A reference to the top-level entity upon which the operation is being performed. Typically, this would be the entity on which the operation was invoked.

		Constraints: **Provider:** support mandatory ; immutable **Consumer:** support mandatory ; read-only
affectedEntities	*ref[]*	A list of references to resources that have been impacted by this Job. Note that this list will always contain the "targetEntity" reference. **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 implementation specific. 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 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 that this entity is a subordinate of. **Constraints:** **Provider:** support mandatory ; immutable **Consumer:** support mandatory ; read-only

nestedJobs	*ref[]*	An array of references to a set of subordinate Job entities. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-only

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Job",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "state": string,
  "targetEntity": { "href": string },
  "affectedEntities": [ { "href": string }, + ],
  "action": string,
  "returnCode": number,
  "progress": number,
  "statusMessage": string,
  "timeOfStatusChange": date,
  "isCancellable": boolean,
  "parentJob": { "href": string }, ?
  "nestedJobs": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/stop", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Job xmlns="http://www.dmtf.org/cimi">
  <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>
  <targetEntity href="xs:anyURI"/>
  <affectedEntity href="xs:anyURI"/> +
  <action> xs:anyURI </action>
  <status> xs:string </status>
  <returnCode> xs:integer </returnCode>
  <progess> xs:integer <progress>
  <statusMessage> xs:string </statusMessage>
  <timeOfStatusChange> xs:dateTime </timeOfStatusChange>
  <isCancellable> xs:boolean </isCancellable>
  <parentJob href="xs:anyURI"/> ?
```

```
  <nestedJob href="xs:anyURI"/> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/stop" href="xs:anyURI"/> ?
  <xs:any>*
</Job>
```

5.17.1.1 Operations

This entity supports the Read, Update and Delete operations.

Note that 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 doesn't support the "stop" action shall fail.

The following custom operations are also defined:

Stopping a Job

/link@rel: http://www.dmtf.org/cimi/action/stop

This operation will 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/REST Protocol**

To stop a Job a POST is sent to the "http://www.dmtf.org/cimi/action/stop" URI of the Job; where the HTTP request body SHALL be as described below.

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/stop",
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

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

5.17.2 Job Collection

A Job Collection entity represents the collection of Jobs within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/JobCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/JobCollectionEntry",
      "id", string,
      "job": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/JobCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/JobCollectionEntry">
    <id> xs:anyURI </id>
    <job href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.17.3 Meter Template

A Meter Template represents the set of information needed to create a new Meter.

Name	MeterTemplate	
Type URI	http://www.dmtf.org/cimi/MeterTemplate	
Attribute	**Type**	**Description**
targetEntity	*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. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
meterConfiguration	*ref*	A reference to the Meter Configuration that will be used to create a Meter from this Meter Template. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write

5.17.4 Meter Template Collection

A Meter Template Collection entity represents the collection of MeterTemplate entities within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MeterTemplateCollection",
  "id": string,
```

```
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/MeterTemplateCollectionEntry",
      "id", string,
      "meterTemplate": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/MeterTemplateCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/MeterTemplateCollectionEntry">
    <id> xs:anyURI </id>
    <meterTemplate href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.17.4.1 Operations

This entity supports the Read and Update operations. Creation of new Meter Template entities is supported via a POST to the "add" operation'sURI as described in section 4.3.2.1.

5.17.5 Meter Configuration

A Meter Configuration represents the definition of a Meter.

Name	MeterConfiguration	
Type URI	http://www.dmtf.org/cimi/MeterConfiguration	
Attribute	**Type**	**Description**
associatedTo	*URI[]*	An array of URIs that indicate the entities to which a Meter created from this configuration can be applied. The value space of these URIs is identical to that of EntityMetadata.typeURI - a URI that uniquely identifies an entity 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*	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*	It indicates the time between consecutive samples in seconds. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
timeScope	*string*	It indicates 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 which purpose is to provide the daily average CPU usage. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
intervalDuration	*duration*	It indicates 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*	It indicates whether or not 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 entity in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MeterConfiguration",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": 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://www.dmtf.org/cimi">
  <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 that last part of the URI - it should be appended to: "http://www.dmtf.org/cimi/aspect/".

Aspect	Description
cpu	The percentage CPU usage of the resource. Typically associated with CEP, System and Machine resources. For resources which 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 which 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 which 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, VSP 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, VSP and Volume resources. For Machine resources, this aspect provides the aggregated input bandwidth usage of all its network interfaces.

5.17.5.1 Operations

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

5.17.6 Meter Configuration Collection

A Meter Configuration Collection entity represents the collection of Meter Configurations within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MeterConfigurationCollection",
  "id": string,
  "entries": [
    { "entityURI":
        "http://www.dmtf.org/cimi/MeterConfigurationCollectionEntry",
      "id", string,
      "meterConfiguration": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/MeterConfigurationCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry
      entityURI="http://www.dmtf.org/cimi/MeterConfigurationCollectionEntry">
    <id> xs:anyURI </id>
    <meterConfiguration href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.17.6.1 Operations

This entity supports the Read and Update operations. Creation of new Meter Configuration entities is supported via a POST to the "add" operation's URI as described in section 4.3.2.1.

5.17.7 Meter

This entity represents an available Meter of some property associated to a given entity.

Name	Meter	
Type URI	http://www.dmtf.org/cimi/Meter	
Attribute	**Type**	**Description**
targetEntity	*ref*	A reference to the entity 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

<table>
<tr><td>units</td><td>string</td><td>Name of the used units, e.g. kilobits per second, CPU usage percentage, etc.

<u>Constraints:</u>
Provider: support mandatory ; immutable
Consumer: support mandatory ; read-only</td></tr>
<tr><td>sampleInterval</td><td>integer</td><td>It indicates the time between consecutive samples in seconds.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>timeScope</td><td>string</td><td>It indicates 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 which purpose is to provide the daily average CPU usage.

<u>Constraints:</u>
Provider: support mandatory ; immutable
Consumer: support mandatory ; read-only</td></tr>
<tr><td>intervalDuration</td><td>duration</td><td>It indicates the interval duration when the timeScope is set to "Interval". Possible values: hourly, daily, weekly, monthly or yearly.

<u>Constraints:</u>
Provider: support mandatory ; immutable
Consumer: support mandatory ; read-only</td></tr>
<tr><td>isContinuous</td><td>boolean</td><td>It indicates whether or not the Meter value is continuous or scalar. Performance Meters are an example of a linear metric.

<u>Constraints:</u>
Provider: support mandatory ; immutable
Consumer: support mandatory ; read-only</td></tr>
<tr><td>samples</td><td>sample[]</td><td>A list of taken samples

Each sample attribute has the following sub-attributes:
<table>
<tr><th>Name</th><td colspan="2">sample</td></tr>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>timeStamp</td><td>dateTime</td><td>It indicates when the measure was taken (timeScope="Point").

When the timeScope is "Interval", it indicates the end of the time interval.

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

		<table><tr><td>value</td><td>*string*</td><td>It indicates the sampled value of the measure. **Constraints:** **Provider:** support mandatory ; immutable **Consumer:** support mandatory ; read-only</td></tr></table> **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-only
minValue	*string*	It indicates the expected minimal measure value. **Constraints:** **Provider:** support mandatory ; immutable **Consumer:** support mandatory ; read-only
maxValue	*string*	It indicates the expected maximum measure value. **Constraints:** **Provider:** support mandatory ; immutable **Consumer:** support mandatory ; read-only
stopTime	*dateTim*	It indicates a time from which the meter stops tracking samples. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write
expiresTime	*dateTime*	It indicates the time from which the Meter is not monitored anymore. It implies the deletion of the Meter after this time. **Constraints:** **Provider:** support mandatory ; mutable **Consumer:** support mandatory ; read-write

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

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Meter",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "targetEntity": { "href": string },
  "aspect": string,
  "units": string,
  "sampleInterval": number,
  "timeScope": string,
  "intervalDuration": string,
  "isContinuous": boolean,
  "samples": [
    { "timestamp": string, "value": string }, +
```

```
  ], ?
  "minValue": string, ?
  "maxValue": string, ?
  "stopTime": string, ?
  "expiresTime": string, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/start", "href": string }, ?
    { "rel": "http://www.dmtf.org/cimi/action/stop", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Meter xmlns="http://www.dmtf.org/cimi">
  <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> *
  <targetEntity 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>
  <sample timestamp="xs:dateTime" value="xs:string"/> *
  <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://www.dmtf.org/cimi/action/start" href="xs:anyURI"/> ?
  <operation rel="http://www.dmtf.org/cimi/action/stop" href="xs:anyURI"/> ?
  <xs:any>*
</Meter>
```

5.17.7.1 Operations

This entity supports the Read, Update and Delete operations. Create is supported via the Meter Collection entity.

Note: the deletion of a Meter shall remove the Meter from the targetEntity's "meter" attribute.

The following custom operations are also defined:

Starting a Meter

/link@rel: http://www.dmtf.org/cimi/action/start

This operation will start a Meter.

Input parameters: None.

Output parameters: None.

Upon successful completion of this operation the Meter will begin to record samples related to its associated resource.

- **HTTP/REST Protocol**

To start a Meter a POST is sent to the "http://www.dmtf.org/cimi/action/start" URI of the Meter; where the HTTP request body SHALL be as described below.

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/start",
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

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

Stopping a Meter

/link@rel: http://www.dmtf.org/cimi/action/stop

This operation will stop a Meter.

Input parameters: None.

Output parameters: None.

Upon successful completion of this operation the Meter will no longer be recording samples related to its associated resource.

- **HTTP/REST Protocol**

To stop a Meter a POST is sent to the "http://www.dmtf.org/cimi/action/stop" URI of the Meter; where the HTTP request body SHALL be as described below.

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Action",
  "action": "http://www.dmtf.org/cimi/action/stop",
  "properties": { "key": string, + } ?
  ...
}
```

XML media type: application/xml

XML serialization

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

```
  <action> http://www.dmtf.org/cimi/action/stop </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

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

5.17.8 Meter Collection

A Meter Collection entity represents the collection of Meters within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/MeterCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/MeterCollectionEntry",
      "id", string,
      "meter": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/MeterCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/MeterCollectionEntry">
    <id> xs:anyURI </id>
    <meter href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.17.8.1 Operations

Note, the "add" operation requires a SystemTemplate to be used.

The resource upon which the new Meter is related (i.e. the "targetEntity"), shall have its "meters" collection updated to include a reference to this new Meter.

5.17.9 Event Log

An entity that represents a registry of Events.

Name	EventLog	
Type URI	http://www.dmtf.org/cimi/EventLog	
Attribute	**Type**	**Description**
targetEntity	*ref*	A reference to the entity to which the Events are related. **Constraints:** **Provider:** support mandatory ; immutable **Consumer:** support mandatory ; read-only

<table>
<tr><td>events</td><td>ref[]</td><td>A list of references to occurred Events.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-only</td></tr>
<tr><td>persistence</td><td>string</td><td>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.

<u>Constraints:</u>
Provider: support mandatory ; mutable
Consumer: support mandatory ; read-write</td></tr>
<tr><td>summary</td><td><unnamed structure></td><td>A summary of all the events present in the EventLog when the read operation is performed, grouped per 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 entity in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/EventLog",
```

```
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "targetEntity": { "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://www.dmtf.org/cimi">
  <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> *
  <targetEntity href="xs:anyURI"/>
  <event 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 Operations

This entity supports the Read, Update and Delete operations.

5.17.10 Event Log Collection

A Event Log Collection entity represents the collection of Event Logs within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/EventLogCollection",
  "id": string,
  "entries": [
```

```
    { "entityURI": "http://www.dmtf.org/cimi/EventLogCollectionEntry",
      "id", string,
      "eventLog": { "href": string },
    }, +
  ], ?
  "operations": [ { "rel": "add", "href", string } ? ]
  ...
}
```

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/EventLogCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/EventLogCollectionEntry">
    <id> xs:anyURI </id>
    <eventLog href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.17.11 Event

An entity 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 kind of information that the Provider is able to track within its infrastructure and that can constitute useful information for the Consumer. Possible examples, but not limited to, are errors and inconveniences that occur in the (virtual) resources assigned to Consumers, some Provider initiated actions such as maintenance tasks, etc.

Name	Event	
Type URI	http://www.dmtf.org/cimi/Event	
Attribute	**Type**	**Description**
timestamp	*dateTime*	Time of occurrence of the actual event. A datetime field formatted according to the DMTF specification [**DSP-4004**] , which follows [**ISO 8601**]; the timestamp SHOULD preserve time zone information, i.e. include a local time component and an offset from UTC. For example, Monday, May 25, 2012, at 1:30:15 PM EST is represented as: `2012-05-25T13:30:15-05:00` Note: this should not be confused with the time of creation of the Event entity 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. which schema it is associated with. **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 will hold the target resource ID and type, measured attribute(s), status value(s). • In the case of an audit event conforming to the CADF model, will hold the detailed event structure that complies with CADF event schema. • In the case of a CIM Indication, will 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 in order to accommodate possible outcomes of a specific event type. In such a case, the extended set of values will 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 attribute is not relevant to a particular type of event, it should be omitted. **Constraints:** **Provider:** support optional ;im mutable **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 entity in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/Event",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { "key": string, + }, ?
  "timestamp": string,
  "type": string,
  "content": any, ?
  "outcome": string, ?
  "severity": string,
  "contact": string, ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Event xmlns="http://www.dmtf.org/cimi">
  <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 defined or acknowledged by this specification. Additional types may be added by a Provider, for example in order 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 root URI: http://www.dmtf.org/cimi/event/, For brevity the "Event Type" column in the table only shows that last part of the URI (e.g. /cimi/event/state) when the URI is defined by this specification - it should then be appended by default to: http://www.dmtf.org .

<table>
<tr><th>Event Type</th><th>Description</th></tr>
<tr><td>/cimi/event/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 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 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 the state of which is reported (note: this reference may become invalid as the event will often 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 Entity type for this resource).

<u>Constraints:</u></td></tr>
</table>
</td></tr>
</table>

			Provider: support optional; immutable **Consumer:** support optional; read-only.
	state	*string*	The state reported for the resource. Must be the same as the "state" attribute value (if any) of the run-time resource at the time the event is generated. **Constraints:** **Provider:** support mandatory; immutable **Consumer:** support optional; read-only
	previous	*string*	In case the event reports a state change, reports the previous state value. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only.

/cimi/event/alarm

Events of this type report errors or alarms occurring during management operations of Cloud resource. This 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:

Data	Type	Description
resName	*string*	The name of the resource associated with this alarm, if applicable. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only.
resource	*ref*	The reference to the resource associated with this alarm, if applicable. (note: this reference may become invalid as the event will often outlive the resource.) **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 Entity 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*	Detailed information associated with the alarm. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only

/cimi/event/model

Events of this type report changes in the CIMI resource model. This includes creation, modification and destruction of entity instances, updates to metadata (entity 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 concerned by the modification. **Constraints:** **Provider:** support optional; immutable **Consumer:** support optional; read-only
resource	*ref*	The reference to the main model resource concerned by the modification. (note: this reference may become invalid as the event will often 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 Entity 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*	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

/cimi/event/access

Events of this type keep track of all requests to access some resource of a CIMI provider.

The **content** element associated with this event type has the following structure:

<table>
<tr><td></td><td>
<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 REST/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 REST/HTTP protocol, the resource URI or the URI associated with the operation.)

<u>Constraints:</u>
Provider: support mandatory; immutable
Consumer: support optional; read-only</td></tr>
<tr><td>detail</td><td>string</td><td>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>In case this information can be associated with the request, provides details identifying the request initiator.

<u>Constraints:</u>
Provider: support optional; immutable
Consumer: support optional; read-only</td></tr>
</table>
</td></tr>
<tr><td>/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://www.dmtf.org/cimi/event/state",
  "content": {
    "resName": string,
    "resource" : { "href" : string },
    "resType" : string,
    "state" : string,
    "previous" : string ?
  }
  ...
}
```

XML Serialization:

```
<Event xmlns="http://www.dmtf.org/cimi">
  ...
  <type> http://www.dmtf.org/cimi/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://www.dmtf.org/cimi/event/alarm",
  "content": {
    "resName": string ?
    "resource" : { "href" : string }, ?
    "resType" : string ?
    "code" : string,
    "detail" : string ?
  }
  ...
}
```

XML Serialization:

```
<Event xmlns="http://www.dmtf.org/cimi">
  ...
  <type> http://www.dmtf.org/cimi/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://www.dmtf.org/cimi/event/model",
  "content": {
    "resName": string, ?
    "resource" : { "href" : string }, ?
    "resType" : string, ?
    "change" : string,
    "detail" : string ?
  }
  ...
```

```
}
```

XML Serialization:

```
<Event xmlns="http://www.dmtf.org/cimi">
  ...
  <type>http://www.dmtf.org/cimi/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://www.dmtf.org/cimi/event/access",
  "content": {
    "operation": string,
    "resource" : { "href" : string },
    "detail" : string, ?
    "initiator" : string ?
    }
  ...
}
```

XML Serialization:

```
<Event xmlns="http://www.dmtf.org/cimi">
  ...
  <type> http://www.dmtf.org/cimi/event/access </type>
  <content>
    <operation> xs:string </operation>
    <resource href="xs:anyURI"/>
    <detail> xs:string </detail> ?
    <initiator> xs:string </initiator> ?
  </content> ?
  ...
</Event>
```

5.17.11.1 Operations

This entity supports the Read, Update and Delete operations.

5.17.12 Event Collection

An Event Collection entity represents the collection of Events within a Provider and follows the Collection pattern defined in section 5.6. This entity shall be serialized as follows:

JSON serialization:

```
{ "entityURI": "http://www.dmtf.org/cimi/EventCollection",
  "id": string,
  "entries": [
    { "entityURI": "http://www.dmtf.org/cimi/EventCollectionEntry",
      "id", string,
```

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

XML serialization:

```
<Collection entityURI="http://www.dmtf.org/cimi/EventCollection"
    xmlns="http://www.dmtf.org/cimi">
  <id> xs:anyURI </id>
  <Entry entityURI="http://www.dmtf.org/cimi/EventCollectionEntry">
    <id> xs:anyURI </id>
    <event href="xs:anyURI"/>
  </Entry> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

6 Security

This specification considers two separate but related security domains. The first domain, API-level security, concerns the protection of the entities modeled by this specification. For example, insuring that unauthorized users are not allowed to alter a Machine instance. The second domain, resource-level security, deals with the protection of the underlying resources represented by these entities. For example, insuring that unauthorized users cannot login to the Linux instance corresponding to that Machine.

6.1 API Level Security

6.1.1 Authentication

Except in cases where the access control policy allows for anonymous requests, the Provider SHALL authenticate all request messages and determine the identity of the Consumer. The techniques used to authenticate messages are outside the scope of this specification.

Protocol bindings of the CIMI Model specification are encouraged to include requirements for the most common authentication mechanisms applicable to that protocol (e.g. the use of BasicAuth for protocols using HTTP).

6.1.2 Message Integrity

Messages exchanged between the Consumer and the Provider SHOULD have message integrity protections applied. The mechanisms used to provide message integrity are outside the scope of this specification.

Protocol bindings of the CIMI Model specification are encouraged to include requirements for the most common integrity mechanisms applicable to that protocol (e.g. the use of TLS for protocols using HTTP).

6.1.3 Message Confidentiality

Messages exchanged between the Consumer and the Provider MAY have message confidentiality protections applied. The mechanisms used to provide message confidentiality are outside the scope of this specification.

6.1.4 Authorization

The Provider SHOULD process messages only if authorized by access control policy, which may reference the Consumer's identity, the message type and content, and other contextual information when making this decision. The language in which this access control policy is expressed as well as the process by which these authorization decisions are made are outside the scope of this specification.

6.1.5 Multi-Tenancy

In cases where a Provider uses multi-tenancy to support a set of Consumers, the operations in this specification are modeled under the assumption that each Consumer's view of the system (i.e. which entities are visible, discoverable, and accessible) is scoped to those entities provisioned for or created by that Consumer. To the Consumer it appears that the Provider is implementing a sole-use instance of the CIMI API (albeit one whose non-functional characteristics may be influenced the actions of invisible co-Consumers).

6.2 Resource Level Credentials

This specification intentionally avoids constraining the type, nature, or operation of the resources represented by the entities that it defines. It is therefore outside the scope of this specification to define the mechanism(s) used to access the resource represented by the Machine entity. There is, however, an integration point between this specification and such mechanisms, namely the management of the credentials (user names, passwords, keys, etc.) used to provision such access. This information is encapsulated by the Credentials entity (described in Section 5.14.11).

Annex A – OVF Support in CIMI

This annex details how elements of the OVF descriptor map to CIMI management resources and their attributes. This allows the import of an OVF package to create multiple CIMI management 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 DSP 0243 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 management resources. A provider may support the import of OVF pachage for only Systems, only System Templates or both. Support for the actual import and export of OVF packages will typically be handeld 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 which can be later used to create Systems, Machines and other component resources. The import of an OVF pacakage can also be used to directly create Systems, Machines and other component resources, bypassing the step of creating templates.

Section 5.12.2 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* 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** management 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** management resource may be created from the *DiskSection* if there is an *ovf:fileRef* for the virtual disk content 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.

Section 5.12.4 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* 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** management 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** management resource may be created from the *DiskSection* if there is an *ovf:fileRef* for the virtual disk content 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.

5289