Document Number: DSP0263

Date: 2011-12-14

Version: 1.0.0c

Work Group Version: 0.0.55c

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: 2012-05-15

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

Abstract

This document is a deliverable from 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.

Acknowledgments

TBD

CONTENTS

FIGURES

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 cloud 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 DSP4004 section 6.3.

1.3 Typographical Conventions

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

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

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

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

204 **IETF RFC 4288**, N. Freed and J. Klensin, *Media Type Specifications and Registration Procedures*,
205 December 2005, http://www.ietf.org/rfc/rfc4288.txt

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

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

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

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

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

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

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

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

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

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

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

235 **DMTF DSP-0243**, Distributed Management Task Force, Inc., *Open Virtualization Format Specification*
236 *1.1.0*, http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.0.pdf

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

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

241 **DMTF DSP-XXXX**, Distributed Management Task Force, Inc., *Cloud Infrastructure Management Interface*
242 *- RelaxNG Model (CIMI-RNG) 0.0.1,*
243 *http://members.dmtf.org/apps/org/workgroup/cmwg/download.php/zzz*

244 **DMTF DSP-ISXXXX**, Distributed Management Task Force, Inc., *Scoping Framework for Cloud*
245 *Management Models and Protocol Requirements 0.1.5*,
246 http://members.dmtf.org/apps/org/workgroup/cmwg/download.php/56339/Cloud%20Management%20Fra
247 mework_v015.doc

248 *Note: the CIMI-RNG document is not yet available*

249 3 Terms and Definitions

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

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

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

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

263 3.1 Authentication

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

267 3.2 Authorization

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

270 3.3 Cloud Service Consumer

271 A category of actors that includes the Consumer Business Manager (who approves business and
272 financial expenditures for consumed services, accounts for used service instances, establishes business
273 relationships; sets up accounts, budget, and terms; etc.), the Consumer Service Administrator (who
274 requests service instances and changes to service instances, purchase services within the business
275 relationship; create Service Users (including policies), allocate resources, such as compute and storage,
276 generate reports (usage), etc.), and Service Users (who uses service instances provided by a Cloud
277 Service Provider). The term “**Consumer**” is used when the indicated action or activity could involve one or
278 more of the above actors. In cases where the distinction between the actors in this category is relevant,
279 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 were the disctinction 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 a "uri" property 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 Cloud 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

Cloud Providers SHALL support secure HTTP connections using TLS. Cloud 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, all CIMI clients and servers shall support the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite (hexadecimal value {0x0013}), which is also the mandatory cipher suite for TLS 1.0 (see [RFC2246] Section 9, Mandatory Cipher Suites), as well as the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite (hexadecimal value {0x002F}) shall be implemented, which is the mandatory cipher suite for both TLS 1.1 and TLS 1.2. Finally, the TLS_RSA_WITH_AES_128_CBC_SHA256 cipher suite (hexadecimal value {0x003C}) should be included with all recommended TLS 1.2 implementations to meet the transition to a security strength of 112 bits (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 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**

4.1.3 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 4.1.6. Providers shall not use the CIMI-defined query parameter reserved namespace (i.e. names starting with "CIMI").

4.1.4 Media Types

In this specification, resource representations and request bodies are encoded in either JSON, as specified in [RFC4627] or in XML.

Each type of resource has its own media-type, which matches the pattern application/.Xxxxx+json, where "Xxxxx" represents the portion of the identifier unique to a particular representation format for each resource (entity in the model). The identifier MUST either be a DMTF standard identifier as defined in this

specification and as registered in accordance to [RFC4288], or it must be a vendor specific identifier that is globally unique (vendor extension).

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.

4.1.5 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). When more than one value is present the server shall choose one of 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 DSP4004 [DSP4004], 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.

4.1.6 Request Query Parameters

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

4.1.6.1 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 qualified 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.

  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[href.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[href.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.

4.1.6.2 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=machines
```

Would result in something similar to this:

```
<MachineCollection xmlns="http://www.dmtf.org/cimi">
  <machine href="http://example.com/machines/1234">
    <self> http://example.com/machines/1234 </self>
    <name> mine </name>
    <state> STARTED </state>
    ...
  </machine>
</MachineCollection>
```

And in JSON it would be:

```
{ "machines": [
    { "href": "http://example.com/machines/1234",
      "self": "http://example.com/machines/1234",
      "name": "mine",
      "state": "STARTED",
      ...
  ]
}
```

4.1.7 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 4.1.5 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. "self" or the "href" for nestedJobs) shall be empty paths (i.e. "") and the "nestedJobs" array shall be expanded (see 4.1.6.2) to inline the representation of the pseudo subordinate Jobs

4.1.8 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. This is used in the create/update/delete of MachineConfiguration and MachineImages to indicate that the provider only supports a fixed set of immutable entities.
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

4.1.9 Serialization of References

References, as indicated by the type 'ref' in the model, are defined to be URIs in the REST/HTTP protocol mapping. In the JSON serialization they will appear as type "string" and in the XML serialization they will appear as type "xs:anyURI".

References in both JSON and XML have an extensibility point that allows for additional information (such as the target resource to be included "by value") if supported. For example, a reference to a Volume in this specification will appear like this in JSON:

```
"volume": { "href": string }
```

and this in XML:

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

For convience the JSON and XML, as shown above, excludes the implicit extensibility points that would allow for the attributes of the target Volume to be included if desired. So, technically the above should be written as:

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

and this in XML:

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

however, for brevity they are excluded.

4.1.10 Serialization of Arrays

Within this specification, arrays in JSON are serialized with a wrapper property. When serializing arrays, conformant implementations SHALL NOT include empty arrays (i.e. arrays that contain no child properties) in the JSON serialization. For example, an array of references to a list of Volumes attached to a Machine is serialized as:

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

Notice that the child of the "volumes" 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 "volumes" element if, and only if, there are volumes.

4.2 Protocol Resource Operations

This section defines the set of common REST/HTTP operations that a Cloud 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.

4.2.1 Operational Principles

4.2.1.1 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/CIMI-...
X-CIMI-Specification-Version: 1.0
```

returns a message containing the following:

JSON serialization:

```
{ "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/CIMI-...
X-CIMI-Specification-Version: 1.0
```

returns a message containing the following:

JSON serialization:

```
{ "attrY": : "bye" }
```

XML serialization:

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

Notes:

1) It is possible that the retrived reference obtained with the GET <ResourceURI> operation does not directly refer to a related resorce, 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 hierachy with just the knowledge of the root URI of the Cloud Provider.

4.2.1.2 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:

```
{ "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/CIMI-...
Content-Type: application/CIMI-...
X-CIMI-Specification-Version: 1.0

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

4.2.2 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 5 then defines resource specific information such as the serialization of each resource's properties and which specific actions are supported.

4.2.2.1 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 contain and "addLink" property which contains the "addURI" that is to be used.

The request will be of the following form:

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

The response will be of the following form:

```
HTTP/1.1 201 Created
Location: ...
Content-Type: application/CIMI-...
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.

4.2.2.2 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/CIMI-...
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/CIMI-...
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.

4.2.2.3 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 "editLink"property, 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 4.2.2.3.1, 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/CIMI-...
Content-Type: application/CIMI-...
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/CIMI-...
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 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.

4.2.2.3.1 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/CIMI-Machine
Content-Type: application/CIMI-Machine
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.

4.2.2.4 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 "deleteLink"property, 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 new 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 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.

4.2.2.5 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 "Link" property/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/CIMI-...
Content-Type: application/CIMI-...
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 4.2.2.1) follows this same pattern - it is just called out for ease of reference.

4.2.2.6 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 a 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.

4.2.2.7 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.

5 Model

This model assumes that a business relationship has already been established between the Cloud Consumer and the Cloud 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.

5.1 Extensibility

There are three types of extensibility mechanisms defined by the CIMI model. 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. Providers may extend the CIMI model by defining new attributes for any of the CIMI resources. It is recommended that providers use the EntityMetadata resource to advertise these new attributes along with any constraints that might need to be understood by consumers.

Finally, providers may also choose to extend the CIMI model by defining new resource types. As with new attributes, it is recommended that providers use EntityMetadata to advertise the presence of these new resources along with any constraints.

5.2 Identifiers

All idenfitiers (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 sensisitive
- 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)

5.3 Attribute "Properties"

Each attribute of the entities in the CIMI model is augmented by a set of "Properties" that further qualify the attribute being defined. The following describes the possible "Properties".

Optionality:

The entity definition tables contain an indicator as to whether the specified attribute (and its corresponding feature) is required to be supported by Cloud providers. Possible values are:

- Optional - indicates that the specified attribute/feature may be supported

- Mandatory - indicates that the specified attribute/feature shall be supported
- Conditional - indicates the specified attribute/feature is mandatory if the condition is satisfied. The condition will be described in the description cell. If the condition is not met, the attribute is optional.

Mutability:

Attributes are either “Immutable” (their values are fixed for the lifetime of the entity), or “Mutable” (their values may change over the lifetime of the entity). Unless otherwise noted, all attributes are mutable.

Writability:

Mutable attributes are either “Writeable” (their value may be changed by the Consumer) or “Read-Only” (their value may only be changed by the Provider). Unless otherwise noted, all mutable attributes are writeable.

5.4 Data Types and Values

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 can not 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.

URIs:

Note that the format and syntax of the attributes of type “URI” is defined by RFC 3986 [RFC3986] with the following, additional constraints: Relative URIs MUST start with a "/", otherwise the URI is assumed to be absolute and no URI processing (to determine the full path) will be performed. Relative URIs are interpreted as being relative to the root URI of the CloudEntryPoint.

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 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.

5.6 Alternative Model Formats

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

This model is available in a CIM/MOF format [CIMI-CIM] as well as a RelaxNG format [CIMI-RNG].

Note: the CIMI-RNG document is not yet available.

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.

5.7 Entities

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

5.7.1 Common Attributes

The entities described by this document share the following, common attributes.

Attribute	Type	Description
self	ref	The unique self-reference to this entity; assigned upon entity creation. This attribute value shall be **unique** in the provider’s cloud. **Properties:** Mandatory / Immutable
name	string	The human readable name of this entity; assigned by the creator as a part of the entity creation input. **Properties:** Optional / Mutable
description	string	The human readable description of this entity; assigned by the creator as a part of the entity creation input. **Properties:** Optional / Mutable

created	DateTimeUTC	The timestamp when this entity was created. The format should be unambiguous, and the value is **immutable**. **Properties:** Optional / Immutable
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. **Properties:** Optional / Mutable

5.8 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 that the Provider may have defined. The mechanism by which this metadata is made available will be protocol specific.

Each entity's metadata will contain the following pieces of information:

Name	EntityMetadata	
Type URI	http://www.dmtf.org/cimi/EntityMetadata	
Attribute	**Type**	**Description**
self	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. **Properties:** Mandatory / Mutable
typeURI	URI	A unique URI associated with, and denoting, this entity type. **Properties:** Mandatory / Mutable
name	string	The name of the entity type. **Properties:** Mandatory / Mutable
attributes	attribute[]	A set of Provider defined metadata that can be used by clients to discover any metadata associated with each attribute, as well we the set of extension attributes. Each attribute will contain the following nested data:

Name	attribute	
Data	**Type**	**Description**
name	string	The name of the attribute. **Properties:** Mandatory / Mutable

<table>
<tr><td></td><td></td><td>
<table>
<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.

Properties: Mandatory / Mutable</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.

Properties: Mandatory / Mutable</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".

Properties: Mandatory / Mutable</td></tr>
<tr><td>constraints</td><td>abstract</td><td>Type specific data that describes the constraints of this attribute. When absent the there are no constraints.

Properties: Mandatory / Mutable</td></tr>
</table>
Properties: Optional / Mutable
</td></tr>
<tr><td>operations</td><td>operation[]</td><td>
A set of Provider defined operations that can be used by clients to act on the entity.

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.

Properties: Mandatory / Mutable</td></tr>
<tr><td>uri</td><td>URI</td><td>A URI that uniquely identifies the operation at a global level.

Properties: Mandatory / Mutable</td></tr>
<tr><td>description</td><td>string</td><td>The human readable description of the semantic of the operation.

Properties: Mandtory / Mutable</td></tr>
<tr><td>method</td><td>string</td><td>The method to use to perform the operation.</td></tr>
</table>
</td></tr>
</table>

				Properties: Mandatory / Mutable
		inputMessage	string	The body mimeType of the request message, it may depend on the model format choosen by the provider. **Properties:** Mandatory / Mutable
		outputMessage	string	The body mimeType of the response message, it may depend on the model format choosen by the provider. **Properties:** Mandatory / Mutable
		Properties: Optional / Mutable		

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

JSON media type: application/CIMI-EntityMetadata+json

JSON serialization:

```
{ "self": string,
  "typeURI": URI,
  "name": string,
  "attributes" : [
    { "name": string,
      "namespace": string, ?
      "type": string, ?
      "required": boolean, ?
      ...constraints...? } *
  ], ?
  "operations" : [
    { "name": string
      "description": string, ?
      "method": string,
      "inputMessage": string, ?
      "outputMessage": string ? }, *
  ] ?
  ...
}
```

XML media type: application/CIMI-EntityMetadata+xml

XML serialization:

```
<EntityMetadata xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name>
  <typeURI> xs:anyURI </typeURI>
  <attribute name="xs:string" namespace="xs:anyURI"? type="xs:string"
             required="xs:boolean"? >
    ...constraints...?
  </attribute> *
  <operation name="xs:string" description="xs:string"? method="xs:string"
             inputMessage="xs:string"? outputMessage="xs:string"? /> *
  <xs:any>*
</EntityMetadata>
```

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

5.8.1 Attribute Types

The following describes the constraint metadata corresponding to the attribute's "type" value.

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

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'.

5.8.2 Examples

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

```
<EntityMetadata xmlns="http://www.dmtf.org/cimi">
  <self> http://example.org/types/VC </self>
  <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">
  <self> http://example.org/types/VC </self>
  <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:

```
{ "self": "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 wich 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 :

```
{ "uri": "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"
    }
  ]
}
```

5.9 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**
systemTemplates	ref	A reference to the System Template Collection of this CloudEntry Point. **Properties:** Optional / Mutable
systems	ref	A reference to the System Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
machineTemplates	ref	A reference to the Machine Template Collection of this Cloud Entry Point. **Properties:** Optional / Mutable

machineConfigs	ref	A reference to the Machine Configuration Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
machineImages	ref	A reference to the Machine Image Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
machineAdminTemplates	ref	A reference to the Machine Admin Template Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
machineAdmins	ref	A reference to the Machine Admin Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
machines	ref	A reference to the Machine Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
volumeTemplates	ref	A reference to the Volume Template Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
volumeConfigs	ref	A reference to the Volume Configuration Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
volumeImages	ref	A reference to the Volume Image Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
volumes	ref	A reference to the Volume Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
networkTemplates	ref	A reference to the Network Template Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
networkConfigs	ref	A reference to the Network Configuration Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
networks	ref	A reference to the Network Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
vspTemplates	ref	A reference to the VSP Template Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
vspConfigs	ref	A reference to the VSP Configuration Collection of this Cloud Entry

		Point. **Properties:** Optional / Mutable
vsps	ref	A reference to the VSP Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
routingGroupTemplates	ref	A reference to the Routing Group Template Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
routingGroups	ref	A reference to the Routing Group Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
meterTemplates	ref	A reference to the Meter Template Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
meters	ref	A reference to the Meter Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
eventLogs	ref	A reference to the Event Log Collection of this Cloud Entry Point. **Properties:** Optional / Mutable
events	ref	A reference to the Event Collection of this Cloud Entry Point **Properties:** Optional / Mutable
jobTime	long	This value is Provider specific and is the minimum amount of time a Job will be retained by the system after the completion of the Job. **Properties:** Optional / Mutable
entityMetadata	ref[]	List of references to EntityMetadata 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. **Properties:** Optional / Mutable

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

JSON media type: application/CIMI-CloudEntryPoint+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "systemTemplates": { "href": string }, ?
  "systems": { "href": string }, ?
  "machineTemplates": { "href": string }, ?
  "machineConfigs": { "href": string }, ?
```

```
  "machineImages": { "href": string }, ?
  "machineAdminTemplates": { "href" string }, ?
  "machineAdmins": { "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 }, ?
  "vspTemplates": { "href": string }, ?
  "vspConfigs": { "href": string }, ?
  "vsps": { "href": string }, ?
  "routingGroups" { "href": string }, ?
  "routingGroupTemplates" { "href": string }, ?
  "meterTemplates": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLogs": { "href": string }, ?
  "events": { "href": string }, ?
  "job_time": number, ?
  "entityMetadata": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
  ] ?
  ...
}
```

XML media type: application/CIMI-CloudEntryPoint+xml

XML serialization:

```
<CloudEntryPoint xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created>
  <property key="xs:string"> xs:string </property> *
  <systemTemplates href="xs:anyURI"/> ?
  <systems href="xs:anyURI"/> ?
  <machineTemplates href="xs:anyURI"/> ?
  <machineConfigs href="xs:anyURI"/> ?
  <machineImages href="xs:anyURI"/> ?
  <machineAdminTemplates href="xs:anyURI"/> ?
  <machineAdmins 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"/> ?
  <vspTemplates href="xs:anyURI"/> ?
  <vspConfigs href="xs:anyURI"/> ?
  <vsps href="xs:anyURI"/> ?
  <routingGroupTemplates href="xs:anyURI"/> ?
  <routingGroups href="xs:anyURI"/> ?
  <meterTemplates href="xs:anyURI"/> ?
  <meters href="xs:anyURI"/> ?
  <eventLogs href="xs:anyURI"/> ?
```

```
  <events href="xs:anyURI"/> ?
  <job_time> xs:integer </job_time>
  <entityMetadata href="xs:anyURI"/> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</CloudEntryPoint>
```

5.9.1 Operations

This entity supports the Read and Update operations.

5.10 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.

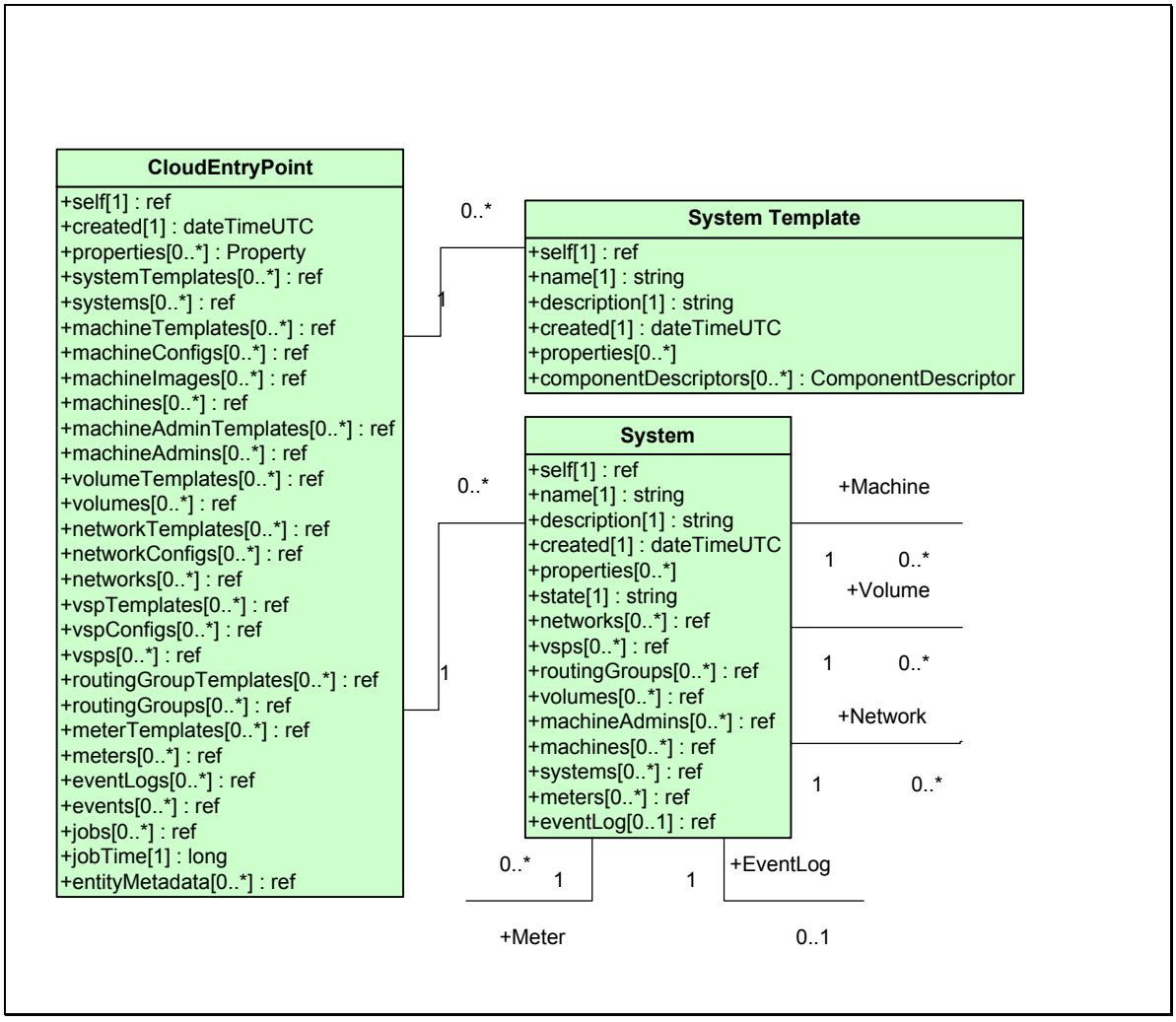

Figure 1 - System Entities

5.10.1 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. **Properties:** Optional / Mutable
description	string	The value of the “description” attribute that will be associated with a System component created from this component descriptor.**Properties:** Optional / Mutable
properties	map	The key/value pairs that will be associated with a System component created from this component descriptor. **Properties:** Optional / Mutable
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

		Properties: Optional / Mutable
componentTemplate	structure	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. **Properties:** Mandatory / Mutable
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.) **Properties:** Optional / Mutable

Properties: Optional / Mutable

JSON media type: application/CIMI-SystemTemplate+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "componentDescriptors": [
    { "name": string, ?
      "description": string, ?
      "properties": { "name": string, + }, ?
      "type": string,
      "componentTemplate": {
        "href": string, ...
      }
    }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
```

```
}
```

XML media type: application/CIMI-SystemTemplate+xml

XML serialization:

```
<SystemTemplate xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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"? > ... </componentTemplate>
  </componentDescriptor> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</SystemTemplate>
```

5.10.1.1 Operations

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

5.10.2 System Template Collection

A System Template Collection entity represents the collection of System Template entities within a Provider. This resource can be used to locate and create System Templates.

Name	SystemTemplateCollection	
Type URI	http://www.dmtf.org/cimi/SystemTemplateCollection	
Attribute	**Type**	**Description**
systemTemplates	ref[]	An array of references to the set of System Templates in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-SystemTemplateCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "systemTemplates": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
```

```
  ...
}
```

XML media type: application/CIMI-SystemTemplateCollection+xml

XML serialization:

```
<SystemTemplateCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <systemTemplate href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</SystemTemplateCollection>
```

5.10.2.1 Operations

This entity supports the Read and Update operations. Creation of new System Template entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.10.3 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.

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**. **AVAILABLE**: The System is available and ready for use. 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. This value is **read-only** and will change based on the state of the System. **Properties:** Mandatory / Mutable

networks	ref[]	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. **Properties:** Optional / Mutable
vsps	ref[]	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. **Properties:** Optional / Mutable
routingGroups	ref[]	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. **Properties:** Optional / Mutable
volumes	ref[]	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. **Properties:** Optional / Mutable
machineAdmins	ref[]	The list of MachineAdmins contained in this System. Adding an item (of type MachineAdmin) to this list is logically equivalent to associating the MachineAdmin to this System with a “containment relationship”. Removing an item from this list is logically equivalent to de-associating the MachineAdmin from this System. **Properties:** Optional / Mutable
machines	ref[]	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. **Properties:** Optional / Mutable
systems	ref[]	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. **Properties:** Optional / Mutable
meters	ref[]	A list of references to Meters monitored for this System. **Properties:** Optional / Mutable
eventLog	ref	A reference to the EventLog of this System.

		Properties: Optional / Mutable

JSON media type: application/CIMI-System+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "state": string,
  "networks": [
    { "href": string }, +
  ], ?
  "vsps": [
    { "href": string }, +
  ], ?
  "volumes": [
    { "href": string }, +
  ], ?
  "machineAdmins": [
    { "href": string }, +
  ], ?
  "machines": [
    { "href": string }, +
  ], ?
  "systems": [
    { "href": string }, +
  ], ?
  "meters": [
    { "href": string }, +
  ], ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-System+xml

XML serialization:

```
<System xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <network href="xs:anyURI"/> *
  <vsp href="xs:anyURI"/> *
  <volume href="xs:anyURI"/> *
  <machineAdmin href="xs:anyURI"/> *
  <machine href="xs:anyURI"/> *
  <system href="xs:anyURI"/> *
  <meter href="xs:anyURI"/> *
  <eventLog href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
```

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

5.10.3.1 Operations

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

5.10.4 System Collection

A System Collection entity represents the collection of System entities within a Provider. This entity can be used to locate and create Systems.

Name	SystemCollection	
Type URI	http://www.dmtf.org/cimi/SystemCollection	
Attribute	**Type**	**Description**
systems	ref[]	An array of references to the set of Systems in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-SystemCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "systems": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-SystemCollection+xml

XML serialization:

```
<SystemCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <system href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</SystemCollection>
```

5.10.4.1 Operations

This entity supports the Read and Update operations.

The following custom operations are also defined:

Creating a New System

/link@rel: add

This operation will create a new System.

Input parameters: Either a reference to a System Template or a System Template itself.

Output parameters: A reference to a new System and optionally the representation of the System.

- **HTTP/REST Protocol**

To create a new System a POST is sent to the "add" URI of the SystemCollection where the HTTP request body SHALL be as described below. Note this structure allows for certain properties to be passed in "by value" or by "reference". The definition of each property can be found in section 5.10.1.

JSON media type: application/CIMI-SystemCreate+json

JSON serialization:

```
{ "name": string, ?
  "description": string, ?
  "properties": { "key": string, + }, ?
  "systemTemplate" : { "href": string , ?
    "properties": { "key": string, + }, ?
    "componentDescriptors": [
      { "name": string, ?
        "description": string, ?
        "properties": { "name": string, + }, ?
        "type": string,
        "componentTemplate": {
          "href": string, ...
        }
      }, +
    ], ?
  }
  ...
}
```

XML media type: application/CIMI-SystemCreate+xml

XML serialization

```
<SystemCreate>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <property key="xs:string"> xs:string </property> *
  <systemTemplate href="xs:anyURI"?>
    <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"? > ... </componentTemplate>
    </componentDescriptor> *
  </systemTemplate>
```

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

The serialization of some reference properties are specified such that a request MAY either include a reference ("href") to an existing entity or to include the entity "inline" as a set of additional properties. Requests SHALL NOT include both a reference and the inlined set of properties.

Upon successful processing of the request, the HTTP response body MAY either be empty or contain a serialization of the System entity.

5.11 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.

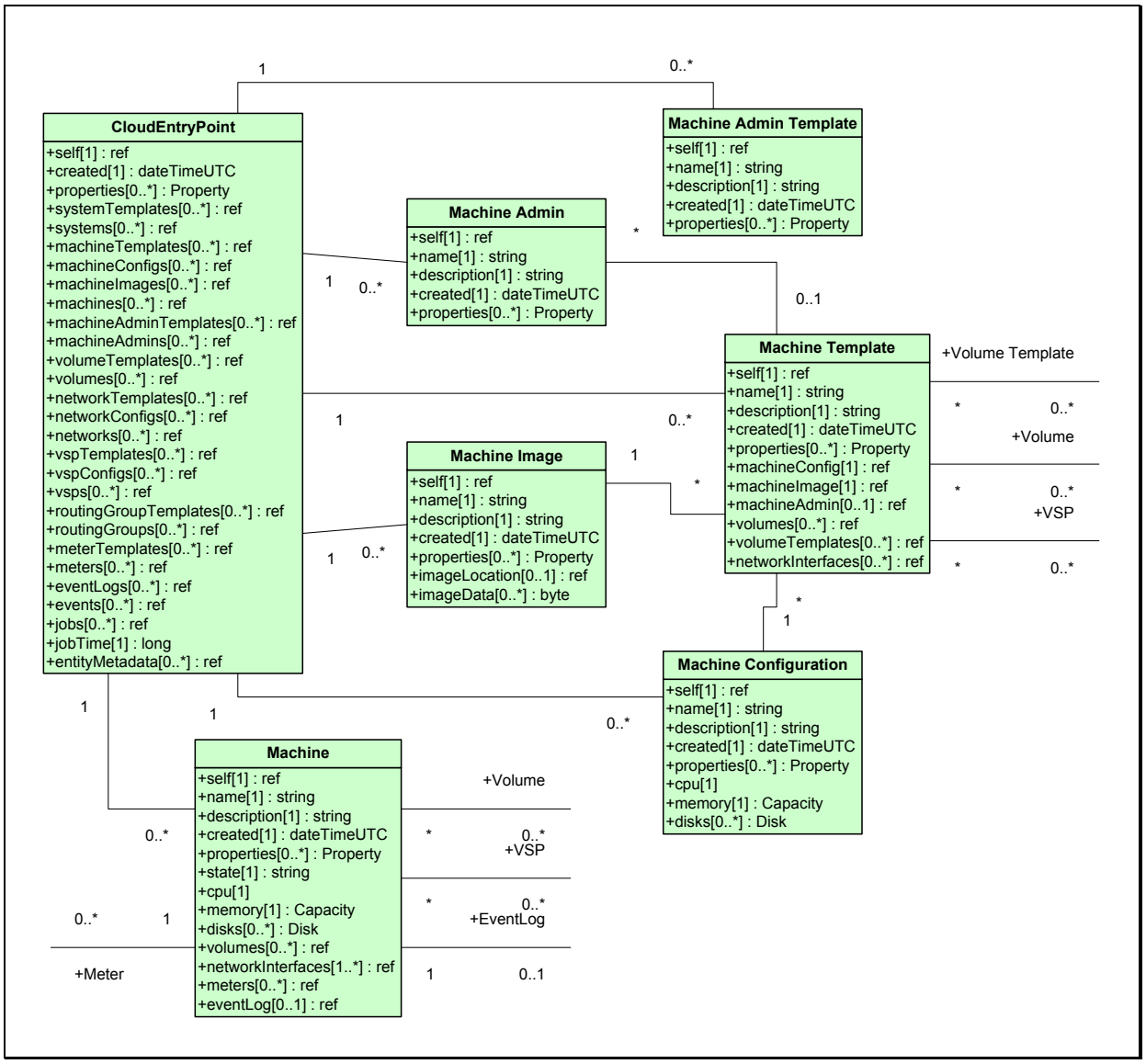

Figure 2 - Machine Entities

5.11.1 Machine Template

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**
machineConfig	ref	A reference to the Machine Configuration that will be used to create a Machine from this Machine Template. **Properties:** Mandatory / Mutable
machineImage	ref	A reference to the Machine Image that will be used to create a Machine from this Machine Template. **Properties:** Mandatory / Mutable
machineAdmin	ref	A reference to the Machine Admin that will be used to create the initial login credential for the new Machine. **Properties:** Optional / Mutable
volumes	volume[]	A list of references to existing Volumes that will be attached to the Machine during its creation. Each volume attribute has the following sub-attributes which describe aspects of the way in which the Machine will be attached to the Volume: **Name**: volume **Attribute** / **Type** / **Description**: attachmentPoint / String / File system path where the Volume will be attached. **Properties:** Mandatory / Mutable volume / ref / Reference to the Volume that will be attached. **Properties:** Mandatory / Mutable **Properties:** Optional / Mutable
volumeTemplates	volumeTemplate[]	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 `volumeTemplates` attribute of the relevant System Template. If the same Volume Template reference is listed in both the `volumeTemplates` attribute of a System Template and in the `volumeTemplates` attribute of a Machine Template contained by

<table>
<tr><td></td><td></td><td>that System Template, this means that multiple, distinct Volume instances will be created as part of the overall System creation.

Each volumeTemplate attribute has the following sub-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.

Properties: Mandatory / Mutable</td></tr>
<tr><td>volumeTemplate</td><td>ref</td><td>Reference to the Volume Template that will be used to create a new Volume.

Properties: Mandatory / Mutable</td></tr>
</table>
Properties: Optional / Mutable</td></tr>
<tr><td>networkInterfaces</td><td>networkInterface[]</td><td>A list of sub-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>vsp</td><td>ref</td><td>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.

Properties: Optional / Mutable</td></tr>
<tr><td>hostname</td><td>string</td><td>DNS resolvable name associated with this network interface.

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.

Properties: Optional / Mutable</td></tr>
</table></td></tr>
</table>

		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. **Properties:** Optional / Mutable
		state	string	The state of an interface configurable to be “Active” or “Standby”. **Properties:** Mandatory / Mutable
		protocol	string	Selected network protocol such as - IPv4 or IPv6. **Properties:** Mandatory / Mutable
		allocation	string	The option for “Dynamic Host Allocation Protocol” or static. **Properties:** Mandatory / Mutable
		address	string	The IP address assigned to a virtual interface. 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. **Properties:** Optional / Mutable
		defaultGateway	string	An IP address to a firewall or router that serves other networks. **Properties:** Optional / Mutable
		dns	string	The IP address of the Domain Name Service from host name to IP resolution. **Properties:** Optional / Mutable

		maxTransmissionUnit	integer	To set the largest supported packet size. **Properties:** Optional / Mutable
		Properties: Optional / Mutable		

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

JSON media type: application/CIMI-MachineTemplate+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "machineConfig": { "href": string },
  "machineImage": { "href": string },
  "machineAdmin": { "href": string }, ?
  "volumes": [
    { "href": string, "attachmentPoint": string }, +
  ], ?
  "volumeTemplates": [
    { "href": string, "attachmentPoint": string }, +
  ], ?
  "networkInterfaces": [
    { "vsp": {"href": string}, "hostname": string, "macAddress": string,
      "state": string, "protocol": string, "allocation": string,
      "address": string, "defaultGateway": string, "dns": string,
      "maxTransmissionUnit": integer }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-MachineTemplate+xml

XML serialization:

```
<MachineTemplate xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <machineConfig href="xs:anyURI"/>
  <machineImage href="xs:anyURI"/>
  <machineAdmin href="xs:anyURI"/> ?
  <volume href="xs:anyURI" attachmentPoint="xs:string" /> *
  <volumeTemplate href="xs:anyURI" attachmentPoint="xs:string" /> *
  <networkInterface>
    <vsp href="xs:anyURI"/>
    <hostname> xs:string </hostname>
    <macAddress> xs:string </macAddress>
    <state> xs:string </state>
    <protocol> xs:string </protocol>
    <allocation> xs:string </alloction>
```

```
    <address> xs:string </address>
    <defaultGateway> xs:string </defaultGateway>
    <dns> xs:string </dns>
    <maxTransmissionUnit> xs:integer </maxTransmissionUnit>
  </networkInterface> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineTemplate>
```

5.11.1.1 Operations

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

5.11.2 Machine Template Collection

A Machine Template Collection entity represents the collection of Machine Template entities within a Provider. This entity can be used to locate and create Machine Templates.

Name	MachineTemplateCollection	
Type URI	http://www.dmtf.org/cimi/MachineTemplateCollection	
Attribute	**Type**	**Description**
machineTemplates	ref[]	An array of references to the set of Machine Templates in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-MachineTemplateCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "machineTemplates": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-MachineTemplateCollection+xml

XML serialization:

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

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

5.11.2.1 Operations

This entity supports the Read and Update operations. Creation of new Machine Template entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.11.3 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.

<table>
<tr><td>Name</td><td colspan="2">MachineConfiguration</td></tr>
<tr><td>Type URI</td><td colspan="2">http://www.dmtf.org/cimi/MachineConfiguration</td></tr>
<tr><td>Attribute</td><td>Type</td><td>Description</td></tr>
<tr><td>cpu</td><td>TBD</td><td>Indicate the amount of CPU (based on standard CPU measurement) that a Machine realized from this configuration would have, by default.

Properties: Mandatory / Mutable</td></tr>
<tr><td>memory</td><td>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.

Properties: Mandatory / Mutable</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.

Properties: Mandatory / Mutable</td></tr>
</table>
Properties: Mandatory / Mutable</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 storages to the Machine.

Each disks attribute has the following sub-attributes:
<table>
<tr><td>Name</td><td colspan="2">disk</td></tr>
<tr><td>Attribute</td><td>Type</td><td>Description</td></tr>
<tr><td>capacity</td><td colspan="2">Indicates the initial capacity of the disk described by this attribute. This property has the following, sub-attributes.</td></tr>
</table>
</td></tr>
</table>

			Attribute	**Type**	**Description**
			quantity	integer	A numerical quantity expressed as an integer. **Properties:** Mandatory / Mutable
			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**. **Properties:** Mandatory / Mutable
			Properties: Mandatory / Mutable		
		format	string	The format/type of this disk (e.g. ext4, NTFS). **Properties:** Mandatory / Mutable	
		attachmentPoint	string	File system path where this disk is attached. **Properties:** Mandatory / Mutable	
		Properties: Optional / Mutable			

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/CIMI-MachineConfiguration+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "cpu": string,
  "memory": { "quantity": integer, "units": string },
  "disks" : [
    { "capacity": { "quantity": integer, "units": string },
      "format": string, "attachmentPoint": string }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-MachineConfiguration+xml

XML serialization:

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

```
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <cpu> xs:string </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> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineConfiguration>
```

5.11.3.1 Operations

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

5.11.4 Machine Configuration Collection

A Machine Configuration Collection entity represents the collection of Machine Configuration entities within a Provider. This entity can be used to locate and create Machine Configurations.

Name	MachineConfigurationCollection	
Type URI	http://www.dmtf.org/cimi/MachineConfigurationCollection	
Attribute	**Type**	**Description**
machineConfigurations	ref[]	An array of references to the set of Machine Configurations in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-MachineConfigurationCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "machineConfigurations": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-MachineConfigurationCollection+xml

XML serialization:

```
<MachineConfigurationCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <machineConfiguration href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</MachineConfigurationCollection>
```

5.11.4.1 Operations

This entity supports the Read and Update operations. Creation of new Machine Configuration entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.11.5 Machine Image

This entity represents the information (e.g. an Open Virtualization Format (OVF) package) necessary for hardware virtualized resources to create a Machine Instance; it contains configuration data such as startup instructions, including possible combinations of the following:

- the software image (i.e. a copy of an installed Machine) which is to be instantiated on the disk and other virtual resources
- 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. This value is **read-only** and will change based on the state of the

		MachineImage. **Properties:** Mandatory / Mutable
imageLocation	ref	A reference to the location of the binary data that makes up this image. Either this attribute or imageData SHALL be present; however both values SHALL NOT be present simultaneously. **Properties:** Optional / Immutable
imageData	byte[]	The binary data that makes up this image. Either this attribute or imageLocation SHALL be present; however both values SHALL NOT be present simultaneously. **Properties:** Optional / Immutable

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

JSON media type: **application/CIMI-MachineImage+json**

JSON serialization:

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

XML media type: application/CIMI-MachineImage+xml

XML serialization:

```
<MachineImage xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <imageLocation href="xs:anyURI"/> ?
  <imageData> xs:string </imageData> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineImage>
```

5.11.5.1 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 "machineImage" 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.

5.11.6 Machine Image Collection

A Machine Image Collection entity represents the collection of Machine Image entities within a Provider. This entity can be used to locate and create Machine Images.

Name	MachineImageCollection	
Type URI	http://www.dmtf.org/cimi/MachineImageCollection	
Attribute	**Type**	**Description**
machineImages	ref[]	An array of references to the set of Machine Images in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-MachineImageCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "machineImages": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-MachineImageCollection+xml

XML serialization:

```
<MachineImageCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <machineImage href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</MachineImageCollection>
```

5.11.6.1 Operations

This entity supports the Read and Update operations. Creation of new Machine Image entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.11.7 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** and **capture**. **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** and **capture**. **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** and **capture**. **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** and **capture**. **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**.

<table>
<tr><td></td><td></td><td>PAUSED and SUSPENDED stated are optional and Providers may choose to support them or not.

Providers may define additional values.

This value is read-only and will change based on the state of the Machine.

Properties: Mandatory / Mutable</td></tr>
<tr><td>cpu</td><td>TBD</td><td>The size of the CPU allocated to this Machine to be used. This should adhere to the standard unit of measurement. For example, a Machine with 4 unit worth of CPU would allow the processes in the Machine to use up to 4 units worth of CPU (and be charged thereof).

When this value is increased, it implies that the Machine is allocated more CPU to use, and vice versa when the value is decreased.

Properties: Mandatory / Mutable</td></tr>
<tr><td>memory</td><td>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.

Properties: Mandatory / Mutable</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, pebibyte, exbibyte, zebibyte, and yobibyte.

Properties: Mandatory / Mutable</td></tr>
</table>
Properties: Mandatory / Mutable</td></tr>
<tr><td>disks</td><td>disk[]</td><td>The list of disks (local storages) that are part of the Machine. Adding an element to this list creates a disk.

Each disk attribute has the following sub-attributes which describe aspects of the disk:
<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 has the following, sub-attributes.
<table>
<tr><th>Attrib</th><th>Type</th><th>Description</th></tr>
</table>
</td></tr>
</table>
</td></tr>
</table>

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

Properties: Mandatory / Mutable</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.

Properties: Mandatory / Mutable</td></tr>
</table>
Properties: Mandatory / Mutable
</td></tr>
</table>
Properties: Optional / Mutable
</td></tr>
<tr><td>volumes</td><td>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 attribute has the following sub-attributes which describe aspects of the way in which the Machine is attached to the Volume:
<table>
<tr><th>Name</th><td>volume</td></tr>
<tr><td colspan="2">See "volume" in "5.11.1 Machine Template" for the definition of this sub-entity.</td></tr>
</table>
Properties: Optional / Mutable
</td></tr>
<tr><td>networkInterfaces</td><td>networkInterface[]</td><td>
A list of sub-entities that define the network interfaces on this Machine.
<table>
<tr><th>Name</th><td colspan="2">networkInterface</td></tr>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>vsp</td><td>URI</td><td>A reference to the VSP (Virtual Switch Port) for this network interface.

Properties: Mandatory / Mutable</td></tr>
<tr><td>hostname</td><td>string</td><td>DNS resolvable name associated with this network interface.

Properties: Mandatory / Mutable</td></tr>
<tr><td>macAddress</td><td>string</td><td>Address assigned by the hypervisor when a machine is created or a</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td>
<table>
<tr><td></td><td></td><td>unique address can be manually assigned.

Properties: Optional / Mutable</td></tr>
<tr><td>state</td><td>string</td><td>The state of an interface configurable to be “Active” or “Standby”.

Properties: Mandatory / Mutable</td></tr>
<tr><td>protocol</td><td>string</td><td>Selected network protocol such as - IPv4 or IPv6.

Properties: Mandatory / Mutable</td></tr>
<tr><td>allocation</td><td>string</td><td>The option for “Dynamic Host Allocation Protocol” or static.

Properties: Machine / Mutable</td></tr>
<tr><td>address</td><td>string</td><td>The IP address assigned to a virtual interface.

Properties: Mandatory / Mutable</td></tr>
<tr><td>defaultGateway</td><td>string</td><td>An IP address to a firewall or router that serves other networks.

Properties: Optional / Mutable</td></tr>
<tr><td>dns</td><td>string</td><td>The IP address of the Domain Name Service from host name to IP resolution.

Properties: Optional / Mutable</td></tr>
<tr><td>maxTransmissionUnit</td><td>integer</td><td>To set the largest supported packet size.

Properties: Optional / Mutable</td></tr>
</table>
Properties: Mandatory / Mutable
</td></tr>
<tr><td>meters</td><td>ref[]</td><td>A list of references to Meters monitored for this Machine.

Properties: Optional / Mutable</td></tr>
<tr><td>eventLog</td><td>ref</td><td>A reference to the EventLog of this Machine.

Properties: Optional / Mutable</td></tr>
</table>

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

JSON media type: application/CIMI-Machine+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
```

```
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "state": string,
  "cpu": string,
  "memory": { "quantity": integer, "units": string },
  "disks" : [
    { "capacity": { "quantity": integer, "units": string } }, +
  ], ?
  "volumes" : [
    { "volume": { "href": string },
      "attachmentPoint": string } +
  ], ?
  "networkInterfaces": [
    { "vsp": {"href": string}, "hostname": string, "macAddress": string,
      "state": string, "protocol": string, "allocation": string,
      "address": string, "defaultGateway": string, "dns": string,
      "maxTransmissionUnit": integer }, +
  ], ?
  "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 } ?
  ]
  ...
}
```

XML media type: application/CIMI-Machine+xml

XML serialization:

```
<Machine xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <cpu> xs:string </cpu>
  <memory quantity="xs:integer" units="xs:string"/>
  <disk>
    <capacity quantity="xs:integer" units="xs:string"/>
  </disk> *
  <volume href="xs:anyURI" attachmentPoint="xs:string"/> *
  <networkInterface>
    <vsp href="xs:anyURI"/>
    <hostname> xs:string </hostname>
    <macAddress> xs:string </macAddress>
    <state> xs:string </state>
    <protocol> xs:string </protocol>
    <allocation> xs:string </alloction>
    <address> xs:string </address>
    <defaultGateway> xs:string </defaultGateway>
    <dns> xs:string </dns>
    <maxTransmissionUnit> xs:integer </maxTransmissionUnit>
  </networkInterface> *
```

```
  <meter 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"/> ?
  <xs:any>*
</Machine>
```

5.11.7.1 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 it's 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/start" URI of the Machine where the HTTP request body SHALL be as described below.

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+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: None.

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 is the virtual equivalent of powering off a physical machine. There is no saved CPU or Memory state.

- **HTTP/REST Protocol**

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

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+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.

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: None.

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 it's boot or installation tasks.

- **HTTP/REST Protocol**

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

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+xml

XML serialization

```
<Action xmlns="http://www.dmtf.org/cimi">
  <action> http://www.dmtf.org/cimi/action/restart </action>
  <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/pause" URI of the Machine where the HTTP request body SHALL be as described below.

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+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 pause a Machine a POST is sent to the "http://www.dmtf.org/cimi/suspend" URI of the Machine where the HTTP request body SHALL be as described below.

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+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 follows the definition as defined by the creating a new Machine Image operation - see 5.11.5.1 for more details.

5.11.8 Machine Collection

A Machine Collection resource represents the collection of Machine entities within a Provider. This resource can be used to locate and create Machines.

Name	MachineCollection

Type URI	http://www.dmtf.org/cimi/MachineCollection	
Attribute	**Type**	**Description**
machines	ref[]	An array of references to the set of Machines in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-MachineCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "machines": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-MachineCollection+xml

XML serialization:

```
<MachineCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <machine href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</MachineCollection>
```

5.11.8.1 Operations

This entity supports the Read and Update operations.

The following custom operations are also defined:

- **Creating a New Machine**

/link@rel: add

This operation will create a new Machine.

Input parameters: Either a reference to a Machine Template or a Machine Template itself.

Output parameters: A reference to a new Machine and optionally the representation of the Machine.

- **HTTP/REST Protocol**

To create a new Machine a POST is sent to the "add" URI of the MachineCollection where the HTTP request body SHALL be as described below. Note this structure allows for certain properties to be passed in "by value" or by "reference". The definition of each property can be found in section 5.11.1.

JSON media type: application/CIMI-MachineCreate+json

JSON serialization:

```
{ "name": string, ?
  "description": string, ?
  "properties": { "key": string, + }, ?
  "machineTemplate": { "href": string, ?
    "properties": { "key": string, + }, ?
    "machineConfig": { "href": string, ?
      "properties": { "key": string, + }, ?
      "cpu": string, ?
      "memory": { "quantity": integer, "units": string }, ?
      "disks" : [
        { "capacity": { "quantity": integer, "units": string },
          "guestInterface": string }, +
      ] ?
    },
    "machineImage": { "href": string,
      "properties": { "key": string, + }, ?
      "imageLocation": { "href": string }, ?
      "imageData": string, ?
    }, ?
    "machineAdmin": { "href": string, ?
      <provider specific data> ?
    }, ?
    "volumes": [
      { "href": string, "attachmentPoint": string }, +
    ], ?
    "volumeTemplates": [
      { "href": string, ?
        "properties": { "key": string, + }, ?
        "attachmentPoint": string,
        "volumeConfig": { "href": string, ?
          "properties": { "key": string, + }, ?
          "format": string, ?
          "capacity": { "quantity": number, "units": string }, ?
          "supportsSnapshots": boolean, ?
          "guestInterface": string ?
        }, ?
        "volumeImage": { "href": string,
          "properties": { "key": string, + }, ?
          "imageLocation": { "href": string }, ?
          "imageData": string, ?
          "bootable": boolean ?
        } ?
      }, +
    ],
    "networkInterfaces": [
      { "vsp": {"href": string}, "hostname": string, "macAddress": string,
        "state": string, "protocol": string, "allocation": string,
        "address": string, "defaultGateway": string, "dns": string,
        "maxTransmissionUnit": integer }, +
    ] ?
  }
  ...
}
```

XML media type: application/CIMI-MachineCreate+xml

XML serialization

```
<MachineCreate>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <property key="xs:string"> xs:string </property> *

  <machineTemplate href="xs:anyURI"? >
    <property key="xs:string"> xs:string </property> *
    <machineConfig href="xs:anyURI"? >
      <property key="xs:string"> xs:string </property> *
      <cpu> xs:string </cpu> ?
      <memory quantity="xs:integer" units="xs:string"/>
      <disk>
        <capacity quantity="xs:integer" units="xs:string">
        <guestInterface> xs:string </guestInterface>
      </disk> *
    </machineConfig>
    <machineImage href="xs:anyURI">
      <property key="xs:string"> xs:string </property> *
       <imageLocation href="xs:anyURI"/> ?
      <imageData> xs:string </imageData> ?
    </machineImage>
    <machineAdmin href="xs:anyURI"? >
      xs:any*  <!-- provider specific data -->
    </machineAdmin> ?
    <volume href="xs:anyURI" attachmentPoint="xs:string"/> *
    <volumeTemplate href="xs:anyURI"? attachmentPoint="xs:string" >
      <property key="xs:string"> xs:string </property> *
      <volumeConfig href="xs:anyURI">
        <property key="xs:string"> xs:string </property> *
        <format> xs:string </format> ?
        <capacity quantity="xs:integer" units="xs:string"/> ?
        <supportsSnapshots> xs:boolean </supportsSnapshots> ?
        <guestInterface> xs:string </guestInterface> ?
      </volumeConfig>
      <volumeImage href="xs:anyURI">
        <property key="xs:string"> xs:string </property> *
        <imageLocation href="xs:anyURI"/> ?
        <imageData> xs:any* </imageData> ?
        <bootable> xs:boolean </bootable> ?
      </volumeImage> ?
    </volumeTemplate> *
    <networkInterface>
      <vsp href="xs:anyURI"/>
      <hostname> xs:string </hostname>
      <macAddress> xs:string </macAddress>
      <state> xs:string </state>
      <protocol> xs:string </protocol>
      <allocation> xs:string </alloction>
      <address> xs:string </address>
      <defaultGateway> xs:string </defaultGateway>
      <dns> xs:string </dns>
      <maxTransmissionUnit> xs:integer </maxTransmissionUnit>
    </networkInterface> *
  </machineTemplate>

  <xs:any>*

</MachineCreate>
```

The serialization of some reference properties are specified such that a request MAY either include a reference ("href") to an existing entity or to include the entity "inline" as a set of additional properties. Requests SHALL NOT include both a reference and the inlined set of properties.

Upon successful processing of the request, the HTTP response body MAY either be empty or contain a serialization of the Machine entity.

5.11.9 Machine Admin Template

This entity captures the configuration values for realizing a MachineAdmin. A Machine Admin Template may be used to create multiple MachineAdmins.

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

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

JSON media type: application/CIMI-MachineAdminTemplate+json

JSON serialization:

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

XML media type: application/CIMI-MachineAdminTemplate+xml

XML serialization:

```
<MachineAdminTemplate xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  ...
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineAdminTemplate>
```

5.11.9.1 Operations

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

5.11.10 Machine Admin Template Collection

A Machine Admin Template Collection entity represents the collection of Machine AdminTemplate entities within a Provider. This entity can be used to locate and create Machine Admin Templates.

Name	MachineAdminTemplateCollection	
Type URI	http://www.dmtf.org/cimi/MachineAdminTemplateCollection	
Attribute	**Type**	**Description**
machineAdminTemplates	ref[]	An array of references to the set of Machine Admin Templates in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-MachineAdminTemplateCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "machineAdminTemplates": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-VolumeTemplateCollection+xml

XML serialization:

```
<MachineAdminTemplateCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <machineAdminTemplate href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</MachineAdminTemplateCollection>
```

5.11.10.1 Operations

This entity supports the Read and Update operations. Creation of new Machine Template entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.11.11 Machine Admin

A Machine Admin entity contains the information required to create the initial administrative super- user of a newly created Machine. Due to the variation between operating systems and Providers, this

specification does not manadate 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	MachineAdmin	
Type URI	http://www.dmtf.org/cimi/MachineAdmin	
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. **Properties:** Mandatory / Mutable
password	string	Initial superuser's password. **Properties:** Mandatory / Mutable

Public Key:

Attribute	**Type**	**Description**
key	byte[]	The digit of the public key for the initial superuser. **Properties:** Mandatory / Mutable

JSON media type: application/CIMI-MachineAdmin+json

JSON serialization:

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

XML media type: application/CIMI-MachineAdmin+xml

XML serialization:

```
<MachineAdmin xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
```

```
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineAdmin>
```

5.11.11.1 Operations

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

5.11.12 Machine Admin Collection

A Machine Admin Collection entity represents the collection of Machine Admin entities within a Provider. This entity can be used to locate and create MachineAdmins.

Name	MachineAdminCollection	
Type URI	http://www.dmtf.org/cimi/MachineAdminCollection	
Attribute	**Type**	**Description**
machineAdmins	ref[]	An array of references to the set of Machine Admins in the provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-MachineAdminCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "machineAdmins": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-MachineAdminCollection+xml

XML serialization:

```
<MachineAdminCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <machineAdmin href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
```

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

5.11.12.1 Operations

This entity supports the Read and Update operations.

The following custom operations are also defined:

- **Creating a New Machine Admin**

/link@rel: add

This operation will create a new Machine Admin.

Input parameters: Either a reference to a Machine Admin Template or a Machine Admin Template itself.

Output parameters: A reference to a new Machine Admin and optionally the representation of the Machine Admin.

- **HTTP/REST Protocol**

To create a new Machine Admin a POST is sent to the "add" URI of the MachineAdminCollection where the HTTP request body SHALL be as described below. Note this structure allows for certain properties to be passed in "by value" or by "reference". The definition of each property can be found in section 5.11.9.

JSON media type: application/CIMI-MachineAdminCreate+json

JSON serialization:

```
{ "name": string, ?
  "description": string, ?
  "properties": { "key": string, + }, ?
  "machineAdminTemplate": { "href": string, ?
    "properties": { "key": string, + }, ?
    ...
  }
  ...
}
```

XML media type: application/CIMI-MachineAdminCreate+xml

XML serialization

```
<MachineAdminCreate>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <property key="xs:string"> xs:string </property> *

  <machineAdminTemplate href="xs:anyURI"? >
    <property key="xs:string"> xs:string </property> *
    ...
  </machineAdminTemplate>

  <xs:any>*

</MachineAdminCreate>
```

The serialization of some reference properties are specified such that a request MAY either include a reference ("href") to an existing entity or to include the entity "inline" as a set of additional properties. Requests SHALL NOT include both a reference and the inlined set of properties.

Upon successful processing of the request, the HTTP response body MAY either be empty or contain a serialization of the MachineAdmin entity.

5.12 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.

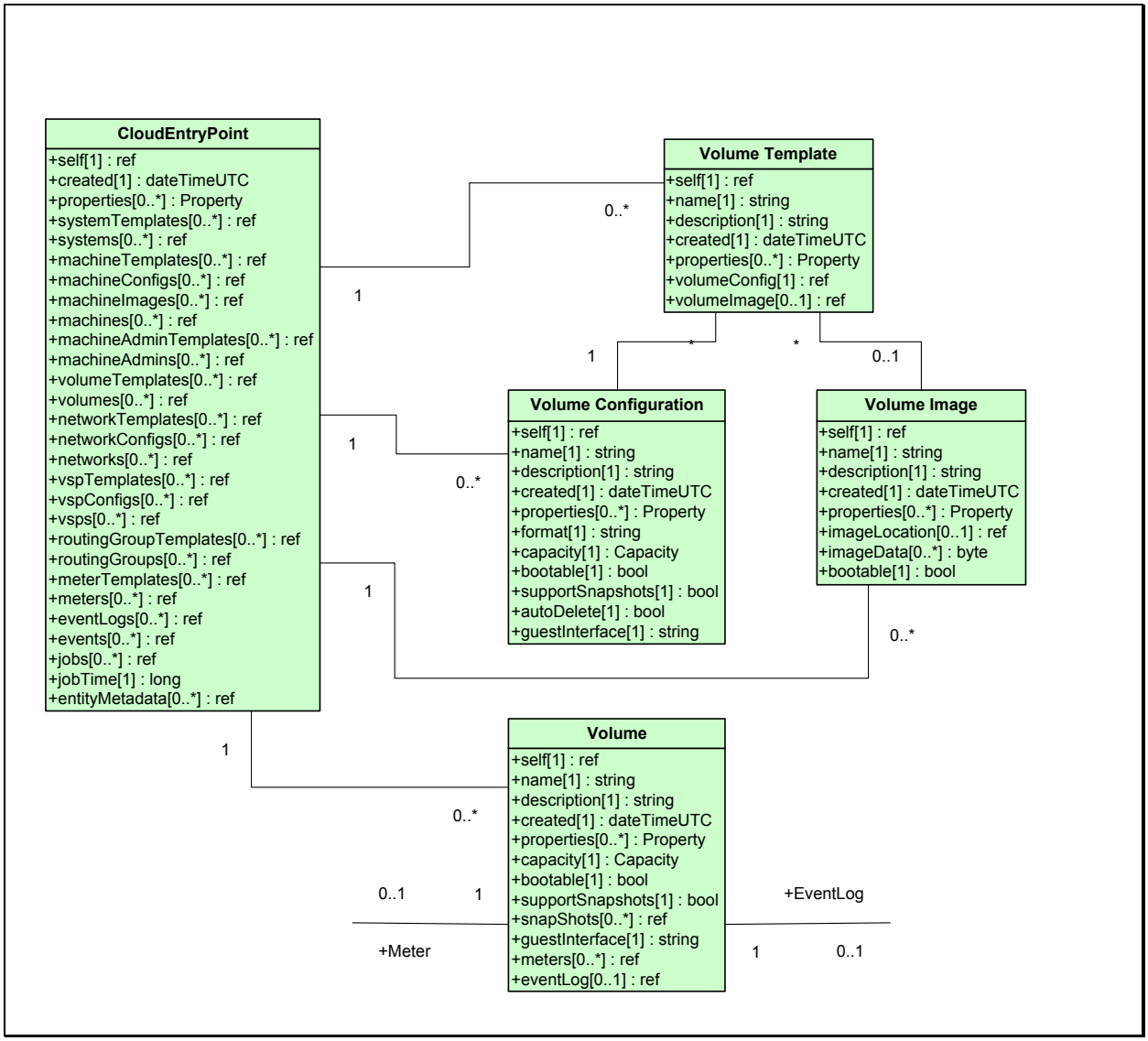

Figure 3 - Volume Entities

5.12.1 Volume Template

This entity captures the configuration values for realizing a Volume. A Volume Template may be used to 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. **Properties:** Mandatory / Mutable
volumeImage	ref	A reference to the Volume Image that will be used to create a Volume from this Volume Template. **Properties:** Optional / Mutable

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

JSON media type: application/CIMI-VolumeTemplate+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "volumeConfig": { "href": string },
  "volumeImage": { "href": string },
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-VolumeTemplate+xml

XML serialization:

```
<VolumeTemplate xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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>
```

5.12.1.1 Operations

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

5.12.2 Volume Template Collection

A Volume Template Collection entity represents the collection of VolumeTemplate entities within a Provider. This entity can be used to locate and create Volume Templates.

Name	VolumeTemplateCollection

Type URI	http://www.dmtf.org/cimi/VolumeTemplateCollection	
Attribute	**Type**	**Description**
volumeTemplates	ref[]	An array of references to the set of Volume Templates in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-VolumeTemplateCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "volumeTemplates": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-VolumeTemplateCollection+xml

XML serialization:

```
<VolumeTemplateCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <volumeTemplate href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</VolumeTemplateCollection>
```

5.12.2.1 Operations

This entity supports the Read and Update operations. Creation of new Volume Template entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.12.3 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". **Properties:** Mandatory / Mutable
capacity	structure	The default size, when limited, of the Volume created from this Volume Configuration. This attribute has the following, sub-attributes. **Attribute** / **Type** / **Description** quantity / integer / A numerical quantity expressed as an integer. **Properties:** Mandatory / Mutable 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**. **Properties:** Mandatory / Mutable **Properties:** Mandatory / Mutable
supportsSnapshots	boolean	This property indicates whether Volumes created from this Volume Configuration will support the ability to take snapshots. **Properties:** Mandatory / Mutable
guestInterface	String	This property indicates the interface that will be offered to a Machine instances by Volumes created from this Volume Configuration. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-VolumeConfiguration+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "format": string,
  "capacity": { "quantity": number, "units": string },
  "supportsSnapshots": boolean,
  "guestInterface": string,
  "operations": [
    { "rel": "edit", "href": string }, ?
```

```
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-VolumeConfiguration+xml

XML serialization:

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

5.12.3.1 Operations

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

5.12.4 Volume Configuration Collection

A Volume Configuration Collection entity represents the collection of Volume Configuration entities within a Provider. This entity can be used to locate and create Volume Configurations.

Name	VolumeConfigruationCollection	
Type URI	http://www.dmtf.org/cimi/VolumeConfigurationCollection	
Attribute	**Type**	**Description**
volumeConfigurations	ref[]	An array of references to the set of Volume Configurations in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-VolumeConfigurationCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "volumeConfigurations": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
```

```
  ] ?
  ...
}
```

XML media type: application/CIMI-VolumeConfigurationCollection+xml

XML serialization:

```
<VolumeConfigurationCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <volumeConfiguration href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</VolumeConfigurationCollection>
```

5.12.4.1 Operations

This entity supports the Read and Update operations. Creation of new Volume Image entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.12.5 Volume Image

This entity represents an image that could be place 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. This value is **read-only** and will change based on the state of the VolumeImage. **Properties:** Mandatory / Mutable
imageLocation	ref	A reference to the location of the binary data that makes up this image. Either this attribute or imageData SHALL be present; however, both values

		SHALL NOT be present simultaneously. **Properties:** Optional / Immutable
imageData	byte[]	The binary data that makes up this image. Either this attribute or imageLocation SHALL be present; however, both values SHALL NOT be present simultaneously. **Properties:** Optional / Immutable
bootable	boolean	This property indicates whether Volumes created from this Volume Configuration will be bootable. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-VolumeImage+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "state": string,
  "imageLocation": { "href": string }, ?
  "imageData": string, ?
  "bootable": boolean,
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-VolumeImage+xml

XML serialization:

```
<VolumeImage xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <imageLocation href="xs:anyURI"/> ?
  <imageData> xs:any* </imageData> ?
  <bootable> xs:boolean </bootable>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</VolumeImage>
```

5.12.5.1 Operations

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

5.12.6 Volume Image Collection

A Volume Image Collection entity represents the collection of VolumeImage entities within a Provider. This entity can be used to locate and create Volume Images.

Name	VolumeImageCollection	
Type URI	http://www.dmtf.org/cimi/VolumeImageCollection	
Attribute	**Type**	**Description**
volumeImages	ref[]	An array of references to the set of Volume Images in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-VolumeImageCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "volumeImages": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-VolumeImageCollection+xml

XML serialization:

```
<VolumeImageCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <volumeImage href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</VolumeImageCollection>
```

5.12.6.1 Operations

This entity supports the Read and Update operations. Creation of new Volume Image entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.12.7 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><th>Name</th><td colspan="2">Volume</td></tr>
<tr><th>Type URI</th><td colspan="2">http://www.dmtf.org/cimi/Volume</td></tr>
<tr><th>Attribute</th><th>Type</th><th>Description</th></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.

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.

This value is read-only and will change based on the state of the Volume.

Properties: Mandatory / Mutable</td></tr>
<tr><td>capacity</td><td>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 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.

Properties: Mandatory / Mutable</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.

Properties: Mandatory / Mutable</td></tr>
</table>
Properties: Mandatory / Mutable</td></tr>
<tr><td>bootable</td><td>boolean</td><td>This property indicates whether this Volume is bootable.

Properties: Mandatory / Mutable</td></tr>
<tr><td>supportsSnapshots</td><td>boolean</td><td>This property indicates whether the Volume supports the ability to take snapshots.</td></tr>
</table>

		Properties: Mandatory / Mutable
snapShots	ref[]	A list of references to Volume Images that represent snapshots taken from the Volume. **Properties:** Optional / Mutable
guestInterface	String	This property indicates the interface offered to a Machine instance to gain access to the storage contents. **Properties:** Mandatory / Mutable
meters	ref[]	A list of references to Meters monitored for this Volume. **Properties:** Optional / Mutable
eventLog	ref	A reference to the EventLog of this Volume. **Properties:** Optional / Mutable

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

JSON media type: application/CIMI-Volume+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "state": string,
  "capacity": { "quantity": number, "units": string },
  "bootable": boolean,
  "supportsSnapshots": boolean,
  "snapshots": [
    { "href": string }, +
  ], ?
  "guestInterface": string,
  "meters": [
    { "href": string }, +
  ], ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-Volume+xml

XML serialization:

```
<Volume xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <capacity quantity="xs:integer" units="xs:string"/>
```

```
  <bootable> xs:boolean </bootable>
  <supportsSnapshots> xs:boolean </supportsSnapshots>
  <shapshot href="xs:anyURI"/> *
  <guestInterface> xs:string </guestInterface>
  <meter href="xs:anyURI"/> *
  <eventLog href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</Volume>
```

5.12.7.1 Operations

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

5.12.8 Volume Collection

A Volume Collection entity represents the collection of Volumes within a Provider. This entity can be used to locate and create Volumes.

Name	VolumeCollection	
Type URI	http://www.dmtf.org/cimi/VolumeCollection	
Attribute	**Type**	**Description**
volumes	ref[]	An array of references to the set of Volumes in the provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-VolumeCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "volumes": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-VolumeCollection+xml

XML serialization:

```
<VolumeCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
```

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

5.12.8.1 Operations

This entity supports the Read and Update operations.

The following custom operations are also defined:

Creating a New Volume

/link@rel: add

This operation will create a new Volume.

Input parameters: Either a reference to a Volume Template or a Volume Template itself.

Output parameters: A reference to a new Volume and optionally the representation of the Volume.

- **HTTP/REST Protocol**

To create a new Volume a POST is sent to the "add" URI of the VolumeCollection where the HTTP request body SHALL be as described below. Note this structure allows for certain properties to be passed in "by value" or by "reference". The definition of each property can be found in section 5.12.1.

JSON media type: application/CIMI-VolumeCreate+json

JSON serialization:

```
{ "name": string, ?
  "description": string, ?
  "properties": { "key": string, + }, ?
  "volumeTemplate": { "href": string, ?
    "properties": { "key": string, + }, ?
    "volumeConfig": { "href": string, ?
      "properties": { "key": string, + }, ?
      "format": string, ?
      "capacity": { "quantity": number, "units": string }, ?
      "supportsSnapshots": boolean, ?
      "guestInterface": string, ?
    },
    "volumeImage": { "href": string,
      "properties": { "key": string, + }, ?
      "imageLocation": { "href": string }, ?
      "imageData": string, ?
      "bootable": boolean ?
    } ?
  }
  ...
}
```

XML media type: application/CIMI-VolumeCreate+xml

XML serialization

```
<VolumeCreate>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <property key="xs:string"> xs:string </property> *
  <volumeTemplate href="xs:anyURI"? >
```

```
    <property key="xs:string"> xs:string </property> *
    <volumeConfig href="xs:anyURI"? >
      <property key="xs:string"> xs:string </property> *
      <format> xs:string </format> ?
      <capacity quantity="xs:integer" units="xs:string"/> ?
      <supportsSnapshots> xs:boolean </supportsSnapshots> ?
      <guestInterface> xs:string </guestInterface> ?
    </volumeConfig>
    <volumeImage href="xs:anyURI">
      <property key="xs:string"> xs:string </property> *
      <imageLocation href="xs:anyURI"/> ?
      <imageData> xs:any* </imageData> ?
      <bootable> xs:boolean </bootable> ?
    </volumeImage> ?
  </volumeTemplate>
  <xs:any>*
</VolumeCreate>
```

The serialization of some reference properties are specified such that a request MAY either include a reference ("href") to an existing entity or to include the entity "inline" as a set of additional properties. Requests SHALL NOT include both a reference and the inlined set of properties.

Upon successful processing of the request, the HTTP response body MAY either be empty or contain a serialization of the Volume entity.

5.13 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.

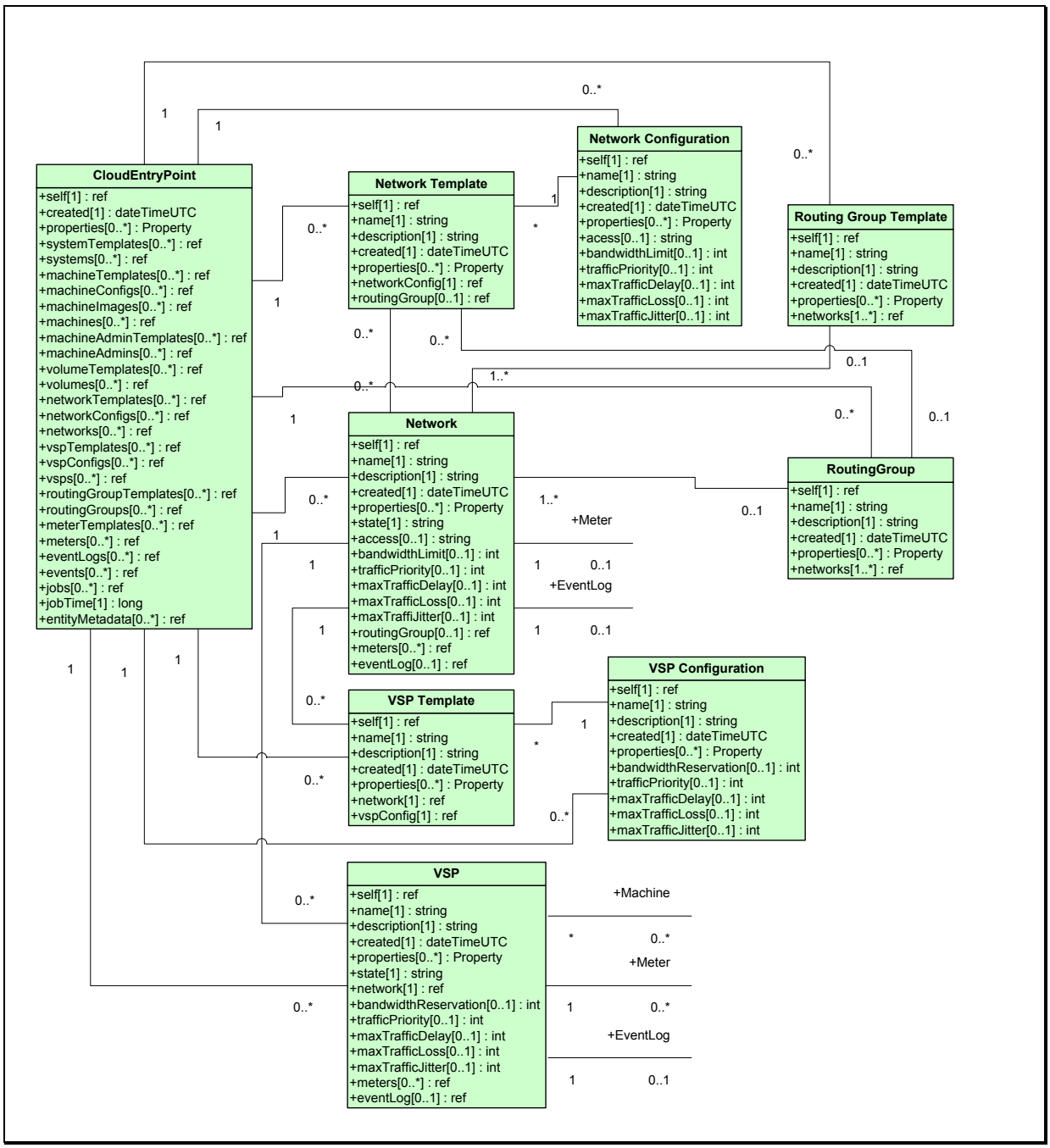

Figure 4 - Network Entities

5.13.1 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. **Properties:** Mandatory / Mutable
routingGroup	ref	A reference to a RoutingGroup that the net Network will be part of. 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. **Properties:** Optional / Mutable

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

JSON media type: application/CIMI-NetworkTemplate+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "networkConfig": { "href": string },
  "routingGroup": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-NetworkTemplate+xml

XML serialization:

```
<NetworkTemplate xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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>
```

5.13.1.1 Operations

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

5.13.2 Network Template Collection

A Network Template Collection entity represents the collection of NetworkTemplateas within a Provider. This resource can be used to locate and create NetworkTemplates.

Name	NetworkTemplateCollection	
Type URI	http://www.dmtf.org/cimi/NetworkTemplateCollection	
Attribute	**Type**	**Description**
networkTemplates	ref[]	An array of references to the set of Network Templates in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-NetworkTemplateCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "networkTemplates": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-NetworkTemplateCollection+xml

XML serialization:

```
<NetworkTemplateCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <networkTemplate href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</NetworkTemplateCollection>
```

5.13.2.1 Operations

This entity supports the Read and Update operations. Creation of new Network Template entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.13.3 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 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. **Properties:** Optional / Mutable
bandwidthLimit	integer	Maximum allowable bandwidth. **Properties:** Optional / Mutable
trafficPriority	integer	Indicates priority of traffic on this network. **Properties:** Optional / Mutable
maxTrafficDelay	integer	The requested maximum delay for end to end transmission specified in nanoseconds with uint64 (i.e. latency). **Properties:** Optional / Mutable
maxTrafficLoss	integer	The requested maximum percentage traffic loss for end to end transmission with uint8, min 0 max 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. The default value is 100. **Properties:** Optional / Mutable
maxTrafficJitter	integer	The requested maximum jitter for end to end transmission with uint32 when traffic is packetized. The variation between packets arriving specified in nanoseconds with uint64. **Properties:** Optional / Mutable

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

JSON media type: application/CIMI-NetworkConfiguration+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": 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/CIMI-NetworkConfiguration+xml

XML serialization:

```
<NetworkConfiguration xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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>
```

5.13.3.1 Operations

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

5.13.4 Network Configuration Collection

A Network Configuration Collection entity represents the collection of Network Configurations within a Provider. This entity can be used to locate and create Network Configurations.

Name	NetworkConfigurationCollection	
Type URI	http://www.dmtf.org/cimi/NetworkConfigurationCollection	
Attribute	**Type**	**Description**
networkConfigurations	ref[]	An array of references to the set of Network Configurations in the provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-NetworkConfigurationCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "networkConfigurations": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-NetworkConfigurationCollection+xml

XML serialization:

```
<NetworkConfigurationCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <networkConfiguration href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</NetworkConfigurationCollection>
```

5.13.4.1 Operations

This entity supports the Read and Update operations. Creation of new Network Collection entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.13.5 Network

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

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: **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 actons 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. This value is **read-only** and will change based on the state of the Network. **Properties:** Mandatory / Mutable
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. **Properties:** Optional / Mutable
bandwidthLimit	integer	Maximum allowable bandwidth. **Properties:** Optional / Mutable
trafficPriority	integer	Indicates priority of traffic on this network. **Properties:** Optional / Mutable
maxTrafficDelay	integer	The requested maximum delay for end to end transmission specified in nanoseconds with uint64 (i.e. latency). **Properties:** Optional / Mutable
maxTrafficLoss	integer	The requested maximum percentage traffic loss for end to end transmission with uint8, min 0 max 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. The default value is 100. **Properties:** Optional / Mutable
maxTrafficJitter	integer	The requested maximum jitter for end to end transmission with uint32 when traffic is packetized. The variation between packets arriving specified in nanoseconds with uint64. **Properties:** Optional / Mutable
routingGroup	ref	A reference to a RoutingGroup that this Netowrk is part of. Note that Networks route to themselves, therefore this attribute will only appear in cases where the Network routes to one or more additional Networks. **Properties:** Optional / Mutable
meters	ref[]	A list of references to Meters monitored for this Network. **Properties:** Optional / Mutable
eventLog	ref	A reference to the EventLog of this Network. **Properties:** Optional / Mutable

2751 The following describes the serialization of the entity in both JSON and XML:

JSON media type: application/CIMI-Network+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "state": string,
  "access": string, ?
  "bandwidthLimit": number, ?
  "trafficPriority": number, ?
  "maxTrafficDelay": number, ?
  "maxTfafficLoss": number, ?
  "maxTrafficJitter": number, ?
  "routingGroup": { "href": string }, ?
  "meters": [
    { "href": string }, +
  ], ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-Network+xml

XML serialization:

```
<Network xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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"/> ?
  <meter href="xs:anyURI"/> *
  <eventLog" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</Network>
```

5.13.5.1 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/start" URI of the Network where the HTTP request body SHALL be as described below.

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+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 start a Network a POST is sent to the "http://www.dmtf.org/cimi/suspend" URI of the Network where the HTTP request body SHALL be as described below.

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+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 start a Network a POST is sent to the "http://www.dmtf.org/cimi/stop" URI of the Network where the HTTP request body SHALL be as described below.

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+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.

5.13.6 Network Collection

A Network Collection entity represents the collection of Networks within a Provider. This entity can be used to locate and create Networks.

Name	NetworkCollection	
Type URI	http://www.dmtf.org/cimi/NetworkCollection	
Attribute	**Type**	**Description**
networks	ref[]	An array of references to the set of Networks in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-NetworkCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "networks": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-NetworkCollection+xml

XML serialization:

```
<NetworkCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <network href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</NetworkCollection>
```

5.13.6.1 Operations

This entity supports the Read and Update operations.

The following custom operations are also defined:

Creating a New Network

/link@rel: add

This operation will create a new Network.

Input parameters: Either a reference to a Network Template or a Network Template itself.

Output parameters: A reference to a new Network and optionally the representation of the Network.

- **HTTP/REST Protocol**

To create a new Network a POST is sent to the "add" URI of the NetworkCollection where the HTTP request body SHALL be as described below. Note this structure allows for certain properties to be passed in "by value" or by "reference". The definition of each property can be found in section 5.13.1.

JSON media type: application/CIMI-NetworkCreate+json

JSON serialization:

```
{ "name": string, ?
  "description": string, ?
  "properties": { "key": string, + }, ?
  "networkTemplate": { "href": string, ?
    "properties": { "key": string, + } ?
    "networkConfig": { "href": string, ?
      "properties": { "key": string, + }, ?
      "access": string, ?
      "bandwidthLimit": number, ?
      "trafficPriority": number, ?
      "maxTrafficDelay": number, ?
      "maxTrafficLoss": number, ?
      "maxTrafficJitter": number, ?
    },
    "routingGroup": { "href": string } ?
    ] ?
  }
  ...
}
```

XML media type: application/CIMI-NetworkCreate+xml

XML serialization

```
<NetworkCreate>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <property key="xs:string"> xs:string </property> *
  <networkTemplate href="xs:anyURI"? >
    <property key="xs:string"> xs:string </property> *
    <networkConfig href="xs:anyURI"? >
      <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> ?
    </networkConfig>
    <routingGroup href="xs:anyURI"/> ?
  </networkTemplate>
  <xs:any>*
</NetworkCreate>
```

The serialization of some reference properties are specified such that a request MAY either include a reference ("href") to an existing entity or to include the entity "inline" as a set of additional properties. Requests SHALL NOT include both a reference and the inlined set of properties.

Upon successful processing of the request, the HTTP response body MAY either be empty or contain a serialization of the Network entity.

5.13.7 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. **Properties:** Mandatory / Mutable
vspConfig	ref	A reference to the VSP Configuration that will be used to create a VSP from this VSP Template. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-VSPTemplate+json

JSON serialization:

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

XML media type: application/CIMI-VSPTemplate+xml

XML serialization:

```
<VSPTemplate xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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>
```

5.13.7.1 Operations

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

5.13.8 VSP (Virtual Switch Port) Template Collection

A VSP Template Collection entity represents the collection of VSP Templates within a Provider. This entity can be used to locate and create VSP Templates.

Name	VSPTemplateCollection	
Type URI	http://www.dmtf.org/cimi/VSPTemplateCollection	
Attribute	**Type**	**Description**
vspTemplates	ref[]	An array of references to the set of VSP Templates in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-VSPTemplateCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "vspTemplates": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-VSPTemplateCollection+xml

XML serialization:

```
<VSPTemplateCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <vspTemplate href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</VSPTemplateCollection>
```

5.13.8.1 Operations

This entity supports the Read and Update operations. Creation of new VSP Template entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.13.9 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. **Properties:** Optional / Mutable
trafficPriority	integer	Indicates priority of traffic on this network. **Properties:** Optional / Mutable
maxTrafficDelay	integer	The requested maximum delay for end to end transmission specified in nanoseconds with uint64 (i.e. latency). **Properties:** Optional / Mutable
maxTrafficLoss	integer	The requested maximum percentage traffic loss for end to end transmission with uint8, min 0 max 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. The default value is 100. **Properties:** Optional / Mutable
maxTrafficJitter	integer	The requested maximum jitter for end to end transmission with uint32 when traffic is packetized. The variation between packets arriving specified in nanoseconds with uint64. **Properties:** Optional / Mutable

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

JSON media type: application/CIMI-VSPConfiguration+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": 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/CIMI-VSPConfiguration+xml

XML serialization:

```
<VSPConfiguration xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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>
```

5.13.9.1 Operations

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

5.13.10 VSP (Virtual Switch Port) Configuration Collection

A VSP Configuration Collection entity represents the collection of VSP Configurations within a Provider. This entity can be used to locate and create VSP Configurations.

Name	VSPConfigurationCollection	
Type URI	http://www.dmtf.org/cimi/VSPConfigurationCollection	
Attribute	**Type**	**Description**
vspConfigurations	ref[]	An array of references to the set of VSP Configurations in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-VSPConfigurationCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "vspConfigurations": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-VSPConfigurationCollection+xml

XML serialization:

```
<VSPConfigurationCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <vspConfiguration href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</VSPConfigurationCollection>
```

5.13.10.1 Operations

This entity supports the Read and Update operations. Creation of new VSP Configuration entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.13.11 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	An indicator of whether or not a specified port is on or off [Default = Enabled]. This value is **read-only** and will change based on the state of the VSP. **Properties:** Mandatory / Mutable 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. This value is **read-only** and will change based on the state of the

		Network. **Properties:** Mandatory / Mutable
network	ref	A reference to the network associated with this VSP. **Properties:** Mandatory / Mutable
bandwidthReservation	integer	Minimum Bandwidth requirements. **Properties:** Optional / Mutable
trafficPriority	integer	Indicates priority of traffic on this VSP. **Properties:** Optional / Mutable
maxTrafficDelay	integer	The requested maximum delay for end to end transmission specified in nanoseconds with uint64 (i.e. latency). **Properties:** Optional / Mutable
maxTrafficLoss	integer	The requested maximum percentage traffic loss for end to end transmission with uint8, min 0 max 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. The default value is 100. **Properties:** Optional / Mutable
maxTrafficJitter	integer	The requested maximum jitter for end to end transmission with uint32 when traffic is packetized. The variation between packets arriving specified in nanoseconds with uint64. **Properties:** Mandatory / Mutable
meters	ref[]	A list of references to Meters monitored for this VSP. **Properties:** Optional / Mutable
eventLog	ref	A reference to the EventLog of this VSP. **Properties:** Optional / Mutable

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

JSON media type: application/CIMI-VSP+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "network": { "href": string },
  "state": string, ?
  "bandwidthLimit": number, ?
  "trafficPriority": number, ?
  "maxTrafficDelay": number, ?
  "maxTfafficLoss": number, ?
  "maxTrafficJitter": number, ?
```

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

XML media type: application/CIMI-VSP+xml

XML serialization:

```
<VSP xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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> ?
  <meter href="xs:anyURI"/> *
  <eventLog" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</VSP>
```

5.13.11.1 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/start" URI of the VSP where the HTTP request body SHALL be as described below.

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+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 start a VSP a POST is sent to the "http://www.dmtf.org/cimi/stop" URI of the VSP where the HTTP request body SHALL be as described below.

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+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.

5.13.12 VSP (Virtual Switch Port) Collection

A VSP Collection entity represents the collection of VSPs within a Provider. This entity can be used to locate and create VSPs.

Name	VSPCollection	
Type URI	http://www.dmtf.org/cimi/VSPCollection	
Attribute	**Type**	**Description**
vsps	ref[]	An array of references to the set of VSPs in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-VSPCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "vsps": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-VSPCollection+xml

XML serialization:

```
<VSPCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <vsp href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</VSPCollection>
```

5.13.12.1 Operations

This entity supports the Read and Update operations.

The following custom operations are also defined:

Creating a New VSP

/link@rel: add

This operation will create a new VSP.

Input parameters: Either a reference to a VSP Template or a VSP Template itself.

Output parameters: A reference to a new VSP and, optionally, the representation of the VSP.

- **HTTP/REST Protocol**

To create a new VSP a POST is sent to the "add" URI of the VSPCollection where the HTTP request body SHALL be as described below. Note this structure allows for certain properties to be passed in "by value" or by "reference". The definition of each property can be found in section 5.13.7.

JSON media type: application/CIMI-VSPCreate+json

JSON serialization:

```
{ "name": string, ?
  "description": string, ?
  "properties": { "key": string, + }, ?
  "vspTemplate": { "href": string, ?
    "properties": { "key": string, + } ?
    "network": { "href": string }, ?
    "vspConfig": { "href": string, ?
      "properties": { "key": string, + }, ?
      "bandwidthReservation": number, ?
      "trafficPriority": number, ?
      "maxTrafficDelay": number, ?
      "maxTrafficLoss": number, ?
      "maxTrafficJitter": number ?
    } ?
  }
  ...
}
```

XML media type: application/CIMI-VSPCreate+xml

XML serialization

```
<NVSPCreate>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <property key="xs:string"> xs:string </property> *
  <vspTemplate href="xs:anyURI"? >
    <property key="xs:string"> xs:string </property> *
    <network href="xs:anyURI"/> ?
    <vspConfig href="xs:anyURI">
      <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> ?
    </vspConfig> ?
  </vspTemplate>
  <xs:any>*
</VSPCreate>
```

The serialization of some reference properties are specified such that a request MAY either include a reference ("href") to an existing entity or to include the entity "inline" as a set of additional properties. Requests SHALL NOT include both a reference and the inlined set of properties.

Upon successful processing of the request, the HTTP response body MAY either be empty or contain a serialization of the VSP entity.

5.13.13 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	ref[]	An array of references to the networks in this Routing Group. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-RoutingGroupTemplate+json

JSON serialization:

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

XML media type: application/CIMI-RoutingGroupTemplate+xml

XML serialization:

```
<RoutingGroupTemplate xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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>
```

5.13.13.1 Operations

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

5.13.14 Routing Group Template Collection

A Routing Group Template Collection entity represents the collection of Routing Group Template entities within a Provider. This entity can be used to locate and create Routing Group Templates.

Name	RoutingGroupTemplateCollection	
Type URI	http://www.dmtf.org/cimi/RoutingGroupTemplateCollection	
Attribute	**Type**	**Description**
routingGroupTemplates	ref[]	An array of references to the set of Routing Group Templates in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-RoutingGroupTemplateCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "routingGroupTemplates": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-RoutingGroupTemplateCollection+xml

XML serialization:

```
<RoutingGroupTemplateCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <routingGroupTemplate href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</RoutingGroupTemplateCollection>
```

5.13.14.1 Operations

This entity supports the Read and Update operations. Creation of new Routing Group Template entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.13.15 Routing Group

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

Name	RoutingGroup
Type URI	http://www.dmtf.org/cimi/RoutingGroup

Attribute	Type	Description
networks	ref[]	An array of references to the networks in this Routing Group. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-RoutingGroup+json

JSON serialization:

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

XML media type: application/CIMI-RoutingGroup+xml

XML serialization:

```
<VSP xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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>*
</VSP>
```

5.13.15.1 Operations

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

5.13.16 Routing Group Collection

A Routing Group Collection entity represents the collection of Routing Groups within a Provider. This entity can be used to locate and create Routing Groups.

Name	RoutingGroupCollection	
Type URI	http://www.dmtf.org/cimi/RoutingGroupCollection	
Attribute	**Type**	**Description**
routingGroups	ref[]	An array of references to the set of RoutingGroups in the Provider.

		Properties: Mandatory / Mutable

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

JSON media type: application/CIMI-RoutingGroupCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "routingGroups": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-RoutingGroupCollection+xml

XML serialization:

```
<RoutingGroupCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <routingGroup href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</RoutingGroupCollection>
```

5.13.16.1 Operations

This entity supports the Read and Update operations.

The following custom operations are also defined:

Creating a New RoutingGroup

/link@rel: add

This operation will create a new RoutingGroup.

Input parameters: Either a reference to a Routing Group Template or a Routing Group Template itself.

Output parameters: A reference to a new RoutingGroup and, optionally, the representation of the RoutingGroup.

- **HTTP/REST Protocol**

To create a new RoutingGroup a POST is sent to the "add" URI of the RoutingGroupCollection where the HTTP request body SHALL be as described below. Note this structure allows for certain properties to be passed in "by value" or by "reference". The definition of each property can be found in section 5.13.13.

JSON media type: application/CIMI-RoutingGroupCreate+json

JSON serialization:

```
{ "name": string, ?
  "description": string, ?
  "properties": { "key": string, + }, ?
  "routingGroupTemplate": { "href": string, ?
    "properties": { "key": string, + }, ?
    "networks": [
      { "href": string }, +
     ] ?
  }
  ...
}
```

XML media type: application/CIMI-RoutingGroupCreate+xml

XML serialization

```
<RoutingGroupCreate>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <property key="xs:string"> xs:string </property> *
  <routingGroupTemplate href="xs:anyURII"? >
    <property key="xs:string"> xs:string </property> *
    <network href="xs:anyURI"/> *
  <routingGroupTemplate/>
  <xs:any>*
</RoutingGroupCreate>
```

The serialization of some reference properties are specified such that a request MAY either include a reference ("href") to an existing entity or to include the entity "inline" as a set of additional properties. Requests SHALL NOT include both a reference and the inlined set of properties.

Upon successful processing of the request, the HTTP response body MAY either be empty or contain a serialization of the RoutingGroup entity.

5.14 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.

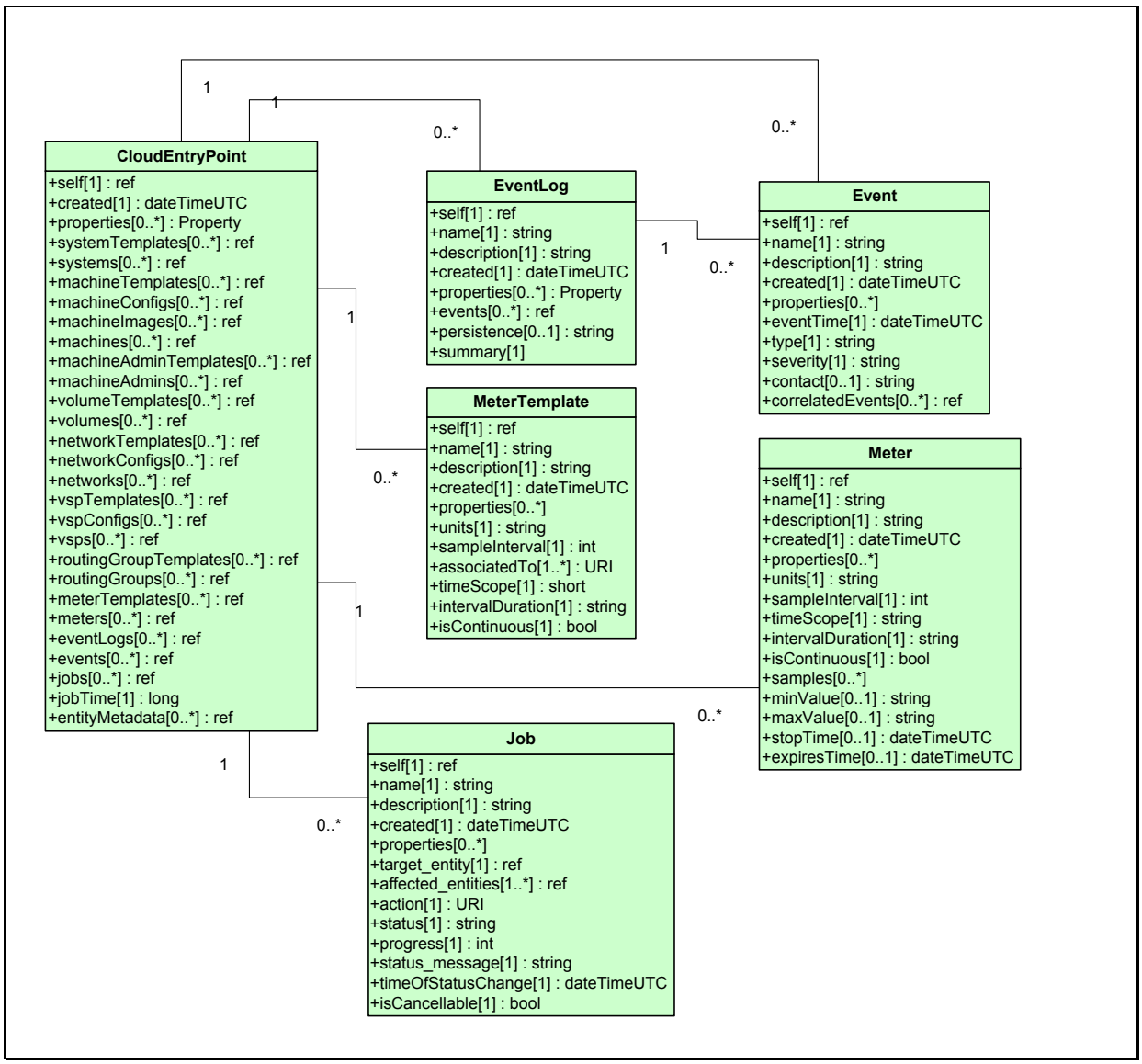

Figure 5 - Monitoring Entities

5.14.1 Job

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

If a Provider supports exposing Job entities to Consumers then each request from a Consumer that would result in a change to the environment MUST result in a Job entity being created and a reference to that Job entity MUST be made available to the requestingConsumer. Providers MAY create additional Job entities for Provider initiated operations.

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

targetEntity	ref	A reference to the top-level entity upon which the operation is being performed. Typically, this would be the instance which was specified when the operation was initiated. This attribute value is **read-only**. **Properties:** Mandatory / Mutable
action	URI	A URI that indicates the type of action being performed. **Properties:** Mandatory / Immutable
status	string	The current status of the process associated with this operation. Sample values for this include "running", "failed", "success", and "cancelled". This attribute value is **read-only**. **Properties:** Mandatory / Mutable
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. This attribute is **read-only**. **Properties**: Mandatory /Immutable
progress	integer	An integer value in the range 0 … 100 that indicates the progress of this Job. This attribute value is **read-only**. **Properties:** Mandatory / Mutable
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. This attribute value is **read-only**. **Properties:** Mandatory / Mutable
timeOfStatusChange	DateTimeUTC	A timestamp indicating the last time that the status of the operation changed. This attribute value is **read-only**. **Properties:** Mandatory / Mutable
isCancellable	boolean	Specifies whether the task being performed by this Job object supports the Cancel operation. **Properties:** Mandatory / Immutable
parentJob	ref	A reference to the Job that this entity is a subordinate of. This attribute is **read-only**. **Properties:** Mandatory / Immutable
nestedJobs	ref[]	An array of references to a set of subordinate Job entities. This attribute is **read-only**. **Properties:** Mandatory / Immutable

3505 The following describes the serialization of the entity in both JSON and XML:

JSON media type: application/CIMI-Job+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "targetEntity": { "href": string },
  "action": string,
  "status": string,
  "returnCode": integer,
  "progress": integer,
  "statusMessage": string,
  "timeOfStatusChange": date,
  "isCancellable": boolean,
  "parentJob": { "href": string }, ?
  "nestedJobs": [
    { "href": string? }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-Job+xml

XML serialization:

```
<Job xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <targetEntity 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"/> ?
  <xs:any>*
</Job>
```

5.14.1.1 Operations

This entity supports the Read, Update and Delete operations.

5.14.2 Job Collection

A Job Collection entity represents the collection of Jobs within a Provider. This resource can be used to locate Jobs.

Name	JobCollection	
Type URI	http://www.dmtf.org/cimi/JobCollection	
Attribute	**Type**	**Description**
jobs	ref[]	An array of references to the set of Jobs in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-JobCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "jobs": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-JobCollection+xml

XML serialization:

```
<JobCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <job href="xs:anyURI"/> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</JobCollection>
```

5.14.2.1 Operations

This entity supports the Read and Update operations.

5.14.3 Meter Template

A Meter Template represents the definition of a Meter.

Name	MeterTemplate	
Type URI	http://www.dmtf.org/cimi/MeterTemplate	
Attribute	**Type**	**Description**
aspect	URI	A unique idenfitier representing the aspect of the resource being metered. See the table below for the set of CIMI defined URIs.

units	string	Human readable name of the used units, e.g. kilobits per second, CPU usage percentage, etc. **Properties:** Mandatory / Mutable
sampleInterval	integer	It indicates the time between consecutive samples in seconds. **Properties:** Mandatory / Mutable
associatedTo	URI[]	An array of URIs that indicate the entities to which a Meter created from this template can be applied. The value space of these URIs is identical to that of EntityMetadata.typeURI - a URI that uniquely identifies an entity type. **Properties:** Mandatory / Mutable
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 MeterTemplate which purpose is to provide the daily average CPU usage. **Properties:** Mandatory / Mutable
intervalDuration	string	It indicates the interval duration when the timeScope is set to “Interval”. Possible values: hourly, daily, weekly, monthly or yearly. **Properties:** Mandatory / Mutable
isContinuous	boolean	It indicates whether or not the Meter value is continuous or scalar. Performance Meters are an example of a linear metric. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-MeterTemplate+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "aspect": string,
  "units": string,
  "sampleInterval": integer,
  "associatedTo": [
    { "href": string }, +
  ], ?
  "timeScope": string,
  "intervalDuration": string,
  "isContinuous": boolean,
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-MeterTemplate+xml

XML serialization:

```
<MeterTemplate xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <aspect> xs:anyURI </aspect>
  <units> xs:string </units>
  <sampleInterval> xs:integer </sampleInterval>
  <associatedTo href="xs:anyURI"/> *
  <timeScope> xs:string </timeScope>
  <intervalDuration> xs:string </intervalDuration>
  <isContinuous> xs:boolean </isContinuous>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MeterTemplate>
```

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 bandwith 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 bandwith 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.

5.14.3.1 Operations

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

5.14.4 Meter Template Collection

A Meter Template Collection entity represents the collection of Meter Templates within a Provider. This entity can be used to locate Meter Templates.

Name	MeterTemplateCollection	
Type URI	http://www.dmtf.org/cimi/MeterTemplateCollection	
Attribute	**Type**	**Description**
meterTemplates	ref[]	An array of references to the set of Meter Templates in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-MeterTemplateCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "meterTemplates": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string } ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-MeterTemplateCollection+xml

XML serialization:

```
<MeterTemplateCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <meterTemplate href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</MeterTemplateCollection>
```

5.14.4.1 Operations

This entity supports the Read and Update operations. Creation of new Meter Template entities is supported via a POST to the "addLink" URI as described in section 4.2.2.1.

5.14.5 Meter

This entity represents an available Meter of some property associated to a given entity.

<table>
<tr><th>Name</th><td colspan="2">Meter</td></tr>
<tr><th>Type URI</th><td colspan="2">http://www.dmtf.org/cimi/Meter</td></tr>
<tr><th>Attribute</th><th>Type</th><th>Description</th></tr>
<tr><td>targetEntity</td><td>ref</td><td>A reference to the entity to which the Meter is related. This attribute is read-only.

Properties: Mandatory / Mutable</td></tr>
<tr><td>aspect</td><td>URI</td><td>A unique idenfitier representing the aspect of the resource being metered.</td></tr>
<tr><td>units</td><td>string</td><td>Name of the used units, e.g. kilobits per second, CPU usage percentage, etc. This attribute is read-only.

Properties: Mandatory / Mutable</td></tr>
<tr><td>sampleInterval</td><td>integer</td><td>It indicates the time between consecutive samples in seconds. This attribute is read-only.

Properties: Mandatory / Mutable</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 Metrer applies to a time interval. For instance, it would be possible to define a MeterTemplate which purpose is to provide the daily average CPU usage. This attribute is read-only.

Properties: Mandatory / Mutable</td></tr>
<tr><td>intervalDuration</td><td>string</td><td>It indicates the interval duration when the timeScope is set to “Interval”. Possible values: hourly, daily, weekly, monthly or yearly. This attribute is read-only.

Properties: Mandatory / Mutable</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. This attribute is read-only.

Properties: Mandatory / Mutable</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>DateTimeUTC</td><td>It indicates when the measure was taken (timeScope=”Point”).

When the timeScope is “Interval”, it indicates the end of the time interval.</td></tr>
</table></td></tr>
</table>

		<table><tr><td></td><td></td><td>**Properties:** Mandatory / Mutable</td></tr><tr><td>value</td><td>string</td><td>It indicates the sampled value of the measure. **Properties:** Mandatory / Mutable</td></tr></table> **Properties:** Mandatory / Mutable
minValue	string	It indicates the expected minimal measure value. This attribute is **read-only**. **Properties:** Mandatory / Mutable
maxValue	string	It indicates the expected maximum measure value. This attribute is **read-only**. **Properties:** Mandatory / Mutable
stopTime	dateTimeUTC	It indicates a time from which the meter stops tracking samples. This attribute is **writable**. **Properties:** Mandatory / Mutable
expiresTime	dateTimeUTC	It indicates the time from which the Meter is not monitored anymore. It implies the deletion of the Meter after this time. This attribute is **writable**. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-Meter+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "targetEntity": { "href": string },
  "aspect": string,
  "units": string,
  "sampleInterval": integer,
  "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 } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-Meter+xml

XML serialization:

```
<Meter xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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:string </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"/> ?
  <xs:any>*
</Meter>
```

5.14.5.1 Operations

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

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/start" URI of the Meter where the HTTP request body SHALL be as described below.

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+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/start" URI of the Meter where the HTTP request body SHALL be as described below.

JSON media type: application/CIMI-Action+json

JSON serialization:

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

XML media type: application/CIMI-Action+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.

5.14.6 Meter Collection

A Meter Collection entity represents the collection of Meters within a Provider. This enity can be used to locate and create Meters.

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

meters	ref[]	An array of references to the set of Meters in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-MeterCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "meters": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "add", "href": string }, ?
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-MeterCollection+xml

XML serialization:

```
<MeterCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <meter href="xs:anyURI"/> *
  <operation rel="add" href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</MeterCollection>
```

5.14.6.1 Operations

This entity supports the Read and Update operations.

The following custom operations are also defined:

Creating a New Meter

/link@rel: add

This operation will create a new Meter.

Input parameters: Either a reference to a Meter Template or a Meter Template itself.

Output parameters: A reference to a new Meter and optionally the representation of the Meter.

- **HTTP/REST Protocol**

To create a new Meter a POST is sent to the "add" URI of the MeterCollection where the HTTP request body SHALL be as described below. Note this structure allows for certain properties to be passed in "by value" or by "reference". The definition of each property can be found in section 5.14.3.

JSON media type: application/CIMI-MeterCreate+json

JSON serialization:

```
{ "name": string, ?
  "description": string, ?
  "properties": { "key": string, + }, ?
  "meterTemplate": { "href": string, ?
    "properties": { "key": string, + }, ?
    "units": string, ?
    "sampleInterval": integer, ?
    "associatedTo": [
      { "href": string }, +
    ], ?
    "timeScope": string, ?
    "intervalDuration": string, ?
    "isContinuous": boolean, ?
  }
  ...
}
```

XML media type: application/CIMI-MeterCreate+xml

XML serialization

```
<MeterCreate>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <property key="xs:string"> xs:string </property> *
  <meterTemplate href="xs:anyURI"? >
    <property key="xs:string"> xs:string </property> *
    <units> xs:string </units> ?
    <sampleInterval> xs:integer </sampleInterval> ?
    <associatedTo href="xs:anyURI"/> *
    <timeScope> xs:string </timeScope> ?
    <intervalDuration> xs:string </intervalDuration> ?
    <isContinuous> xs:boolean </isContinuous> ?
  </meterTemplate>
  <xs:any>*
</MeterCreate>
```

The serialization of some reference properties are specified such that a request MAY either include a reference ("href") to an existing entity or to include the entity "inline" as a set of additional properties. Requests SHALL NOT include both a reference and the inlined set of properties.

Upon successful processing of the request, the HTTP response body MAY either be empty or contain a serialization of the Meter entity.

5.14.7 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.
events	ref[]	A list of references to occurred Events.

<table>
<tr><td></td><td></td><td>Properties: Mandatory / Mutable</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.

Properties: Mandatory / Mutable</td></tr>
<tr><td>summary</td><td>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 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.

Properties: Mandatory / Mutable</td></tr>
<tr><td>medium</td><td>integer</td><td>Number of occurred Events with a medium severity.

Properties: Mandatory / Mutable</td></tr>
<tr><td>high</td><td>integer</td><td>Number of occurred Events with a high severity.

Properties: Mandatory / Mutable</td></tr>
<tr><td>critical</td><td>integer</td><td>Number of occurred Events with a critical severity.

Properties: Mandatory / Mutable</td></tr>
</table>
Properties: Mandatory / Mutable</td></tr>
</table>

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

JSON media type: application/CIMI-EventLog+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": 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/CIMI-EventLog+xml

XML serialization:

```
<EventLog xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <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>
```

5.14.7.1 Operations

This entity supports the Read, Update and Delete operations.

5.14.8 Event Log Collection

A Event Log Collection entity represents the collection of Event Logs within a Provider. This resource can be used to locate EventLogs.

Name	EventLogCollection	
Type URI	http://www.dmtf.org/cimi/EventLogCollection	
Attribute	**Type**	**Description**
eventLogs	ref[]	An array of references to the set of Event Logs in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-EventLogCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "eventLogs": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string } ?
  ] ?
```

```
  ...
}
```

XML media type: application/CIMI-EventLogCollection+xml

XML serialization:

```
<EventLogCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <eventLog href="xs:anyURI"/> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</EventLogCollection>
```

5.14.8.1 Operations

This entity supports the Read and Update operations.

5.14.9 Event

An entity that represents the notification 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**
eventTime	dateTimeUTC	The time and date of creation of the Event. **Properties:** Mandatory / Immutable
type	string	A value that indicates the kind of Event (informational, error, alarm, etc.). This attribute is **read-only**. **Properties:** Mandatory / Mutable
severity	string	A value indicating the Event severity.Possible values are: critical, high, medium or low. This attribute is **read-only**. **Properties:** Mandatory / Mutable
contact	string	An optional identifier that references a contact point to solve the problem (helpdesk, technical staff, etc.). This attribute is **read-only**.

		Properties: Mandatory / Mutable
correlatedEvents	ref[]	A list of Event references whose notifications are correlated with (related to) this one. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-Event+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "eventTime": string,
  "type": string,
  "severity": string,
  "contact": string, ?
  "correlatedEvents": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-Event+xml

XML serialization:

```
<Event xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <eventTime> xs:dateTime </eventTime>
  <type> xs:string </type>
  <severity> xs:string </severity>
  <contact> xs:string </contact> ?
  <correlatedEvent href="xs:anyURI"/> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</Event>
```

5.14.9.1 Operations

This entity supports the Read, Update and Delete operations.

5.14.10 Event Collection

An Event Collection entity represents the collection of Events within a Provider. This entity can be used to locate Events.

Name	EventCollection	
Type URI	http://www.dmtf.org/cimi/EventCollection	
Attribute	**Type**	**Description**
events	ref[]	An array of references to the set of Events in the Provider. **Properties:** Mandatory / Mutable

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

JSON media type: application/CIMI-EventCollection+json

JSON serialization:

```
{ "self": string,
  "name": string, ?
  "description": string, ?
  "created": string,  ?
  "properties": { "key": string, + }, ?
  "events": [
    { "href": string }, +
  ], ?
  "operations": [
    { "rel": "edit", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/CIMI-EventCollection+xml

XML serialization:

```
<EventCollection xmlns="http://www.dmtf.org/cimi">
  <self> xs:anyURI </self>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created>
  <property key="xs:string"> xs:string </property> *
  <event href="xs:anyURI"/> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <xs:any>*
</EventCollection>
```

5.14.10.1 Operations

This entity supports the Read and Update operations.

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 theConsumer. 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 who's 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 Machine Admin entity (described in Section 0).

Annex A (informative) – Change Log

Version	Date	Who	What
0.0.1	10/15/10	Gil, Jack	Initial Draft
0.0.2	10/19/10	Jack	Adding the attribute descriptions and high level operational descriptions on the entities
0.0.3	10/29/10	Gil	Add section on the “Initial Scenario” and the mapping of its required use cases to our model.
0.0.4	11/15/10	Gil	Removed ‘definition’ attribute from System Template, Machine Template, Volume Template, and Network Template (per 912). Added “networkInterfaces” attribute to Machine with sub-properties that define IP address – added “protocol”, “subnet_mask”, “default_gateway” and “dns_servers to Netowork entity (per 910). Removed inline issues and created issues 928, 929, and 930.
0.0.5	11/17/10	Gil	Change “Cloud Site” to “Site per 882. Added Job entity and removed ‘progress’ attributes per 911. Added structure to Machine/disks and Machine/volumes per 915.
0.0.6	12/01/10	Gil	Removed “jobs” attributes from System and System Template to complete 911. Added “capacity” and “format”, removed “type” from Machine/volumes to complete 915.
0.0.7	12/10/10	Gil	Added Image entity to resolve 935. Added new initial scenario to resolve 994.
0.0.8	12/15/10	Gil	Removed “os” attribute from Machine Template to resolve 1032. Added quantity/units sub-properties to describe memory and disk sizes and capacities for Machines and Machine Templates to resolve 1009. Removed “based_on” attribute from System, Machine, Volume, and Network to resolve 1001. Removed all template property descriptions to the effect that “changes to [this attribute] should correspondingly evaluate the [objects] that have been instantiated based on this [object template]” to resolve 1005.
0.0.9	01/05/11	Gil	Add Update operation to all entities as resolution to 1003. Change “Image” entity to “Machine Image” and updated description to resolve 1026. Fixed some capitalization and terminology inconsistencies.
0.0.10	01/18/11	Gil	Add Provider entity as resolution to 1043. Change “params” attribute to “properties” in all entities with that attribute – resolves 1002.
0.0.11	01/19/11	Gil	Move “format” attribute from the Machine-Volume connection to Volume itself and remove “capacity” attribute from the Machine-Volume connection to resolve 956. Add “properties” attribute to all entities that lacked it as completion of 1002. Add “Read” operation to all entities that lacked it; homogenize the description of the “Read” operation across all entities - 1049
0.0.12	01/26/11	Gil	Further edits to tweak the resolution for 1043.

0.0.13	02/02/11	Gil	Added “Create new Machine Image from existing Machine” scenario as per the decision of the WG on 02/02/2011.
0.0.14	02/09/11	Gil	Added “job_time” property to Provider entity as resolution of 1038. Renamed Machine Template to Machine Configuration and created a new Machine Template entity that reflects the resolution of 1045.
0.0.15	02/21/11	Gil	Added definition of “Template” as resolution of 1063. Changed definitions of Machine Configuration and Machine image as resolution of 1069. Updated UML diagram to remove all relationships that weren’t explicitly defined as attributes of the relevant entities.
0.0.16	03/07/11	Gil	Add additional text to description of Template in section 3.2 to resolve 1044. Add column to all entity tables to describe attribute data types to resolve 1073. Changed the 'templates' attribute of Site to 'systemtemplates' to resolve 1075. Changed the description of Machine Configuration to resolve 1079.
0.0.17	03/22/11	Gil	Change attributes on Job entity to resolve 1080. Added “guestInterface” attribute to Machine Configuration, Machine, Volume Template, and Volume to resolve 1083. Changed description of the operations on Machine Configuration to resolve 1084. Moved “hostname” attribute of Machine to a sub-property of the “networkinterface” to resolve 1087. Added “volumes” and “volumetemplates” attributes to Provider entity to resolve 1089. Removed “state” attributes from System Template and Machine Template to resolve 1093.
0.0.18	03/23/11	Gil	Add constraint on relative URIs to heading of Section 4.2, “Attributes”, to resolve 1100. Add “volume” sub-attribute to “volumes” attribute of Machine to resolve 1110. Applied consistent camelCased naming to resolve 1111. Added definitions for “immutable/mutable” and “writeable/read-only” to heading of Section 4.2, “Attributes”; revised all mentions of immutable and mutable to agree with these definitions to resolve 1126.
0.0.19	03/30/11	Gil	Add table that defines attributes common to all entities and remove those same attributes from the entity tables to resolve 1094. Add Volume Configuration and Volume Image entities to resolve 1096.
0.0.20	04/06/11	Gil	Add sub-section that describes unit attributes in both base-10 and base-2 (e.g. kilobyte, kibibyte), change memory units to base-2 names (e.g. kibibyte, mebibyte), change unit designators to singular, add reference to IEC 80000-13:2008 – all to resolve 1101.
0.0.21	04/13/11	Doug	Renamed title of section 4.2 per 1153. Expanded the text for the create operation of a System per 999. Added the delete operation to the list of ops for Network per 1011. Converted all pointers to URIs instead of the name of the type its points to per 1129.
0.0.22	04/20/11	Doug	Added section 4.2.1 per 1113 and 1115. Modified the definition of Site.Create per 1132. Modified Site description per 1133. Moved 'bootable' from volume Config to Volume Image per 1137. Removed 'local' from Volume and Volume Config per 1138. Modified definition of Volume per 1139. Removed 'autoDelete' from Volume per 1140.
0.0.23	04/27/11	Gil	Changed description of unit values in Section 4.2.2.1 to resolve

			1157. Change places where “mutable” is meant as “writeable” to use “writeable”; add Update operation to Provider entity to resolve 1158.
0.0.24	05/11/11	Gil	Added Machine Admin entity to resolve 1164. Added text describing the requirements on when a Job entity is created to resolve 1166.
0.0.25	05/25/11	Gil	Added new Section 4.2 describing the facilities for retrieving metadata about the entities, added “Entity Metadata” to UML diagram, added “Type URI” to each entity type - to resolve 1135. Changed description of VolumeConfiguration:format and removed Volume:format to resolve 1136.
0.0.26	06/01/11	Gil	Added volumes and volumeTemplates attributes to the Machine Templates entity to resolved 1155. Changed description of Provider entity to resolve 1174. Chaged the description of the SystemTemplate.Update operation to resolve 1175. Added Section 6 “Security” (plus relevant definitions) to resolve 1178.
0.0.27	06/07/11	Gil	Merged Site and Provider entities into new Cloud Entry Point entity to resolve 1192 and 1196. Added operations and options to Machine and Machine template to resolve 1204. Replaced “Create and deploy a Machine using a Provider created Machine Template” scenario and added “Create a Machine by passing a Machine Template by value”, and “Create a Machine using a User created Machine Template” to resolve 1205. Added “Create a Machine Template by specifying individual components” and “Create a Machine Template from a template file” scenarios to resolve 1206. Added “Create new Machine Image from an image file” and “Create new Machine Image from Machine instance” scenarios to resolve 1207.
0.0.28	06/09/2011	Gil	Added ‘MeterTemplate’ and ‘Meter’ entities and references to resolve 908. Added Event and EventLog entities and references to resolve 909. Added ‘Cancel’ operation and ‘isCancellable’ attribute to the Job entity to resolve 1012. Added terminology definition for “Configuration” to resolve 1191.
0.0.29	06/20/2011	Gil	Removed “Role in Use Cases” rows from entity tables to resolve 1223. Fixed miscellaneous typos and miscapitalizations to resolve 1226.
0.0.30	07/28/2011	Gil, Doug	Merged CM model and HTTP protocol documents. Added definitions of “Consumer” and “Provider” to Section 3, “Terms and Definitions”; made various changes to use these terms consistently throughout the document to resolve 1180. Fix serializations to resolve 1219. Added support for partial updates to resolve 1154. Fixed the CloudEntryPoint to point to the Collection entities to resolve 1238. Tweaked the uri field of EntityMetadata to resolve 1254. Added resolution of 1171 from HTTP protocol doc. Changed Network Template and Network entities, added VSP Template, VSP Configuration, and VSP to resolve 1010, 1085, 1086, and 1088.
0.0.31	08/01/2011	Doug	Removed text that duplicates the HTTP spec to resolve 1193.
0.0.32	08/22/2011	Doug	Added .../xs:any* to show explicit extensibility points to resolve 1271. Added section 4.1.10 (Serialization of Array) and s/*/+/g on array children to resolve 1270.

0.0.33	08/24/2011	Doug	Used CIMI and Cloud Infrastructure Management Interface where appropriate. Added section 5.1 which points to the CIMI-CIM and CIMI-RNG docs. Add WIP front-matter.
0.0.34	08/28/2011	Doug	s/The follow describes/The following describes/. Fixed the heading style on the "Entities" section - it lost its style at some point. Added the serialization headers for the EntityMetadata JSON/XML - to match the other entities in the model. s/Entity/EntityMetadata/ in the XML to match the entity type and be more descriptive.
0.0.35	08/31/2011	Gil, Doug	Re-factor ER diagram into separate sub-diagrams and re-organize sections around these diagrams. Changed title of doc and added resolution of issue 1310.
0.0.36	09/07/2011	Brightleaf	Various edits as part of the WIP release.
0.0.37	09/09/2011	Gil	Changed description of Job:isCancellable to resolve 1240. Add Volumes and Networks to System entity to resolve 1245. Changed title of Section 6.1.6 to “Control Machine State” to resolve 1246. Added ‘status’ attribute to Network and changed VSP:state to VSP:status to resolve 1255. Changed descriptions of ‘imageLocation’ and ‘imageData’ for both MachineImage and VolumeImage to resolve 1264. Changed wording in 4th paragraph of 4.2.1.3.1 to resolve issue 1266.
0.0.38	09/09/2011	Doug	Removed empty row in Machine table, filled in empty "Optionality" cells for CEP.
0.0.39	09/12/2011	Doug	Added section "4.2.1 Operational Principles" to resolve HTTP issue 1172.
0.0.40	09/21/2011	Doug	Lots of minor editorial changes to resolve 1269.
0.0.41	09/21/2011	Doug	Added text about URIs to resolve issue 1267. Modified the "properties" attribute to resolve issue 1352. Fixed SystemTemplate, it was missing some name, description and networkInterface definition attributes in the pseudo-schema. Added "Model Semantics and Conventions" section to resolve issue 1274. Made CEP.EntityMetadata a URI[] instead of a map to resolve issue 1243. Make 'stop' action URIs consistent to resolve issue 1364.
0.0.42	10/04/2011	Doug	Added typographical convention and preamble text to terms & def'n section to resolve issue 1272. Removed "format" and "attachmentPoint" from Machine.disk to resolve issue 1241. Removed disk.guestInterface from Machine and MachineConfig to resolve issue 1242. Changed most uses of URI to "ref" in the model tables to resolve issue 1351. Changed 'uri' to "self" on entities to resolve 1220.
0.0.43	10/04/2011	Doug	Added support for CIMISelect query parameter to resolve issue 1384. Clarified the optionality of the HTTP version header to resolve issue 1363.
0.0.44	10/05/2011	Doug	Added start/restart to Machine's operations resolve issue 1369. Cleaned up some text around Jobs to resolve issue 1194. Tweak the optionality of some attributes to resolve issue 1412. Add support for operations in EntityMetadata per issue 1168. Added the definition of

			optional, mandatory and condition to resolve issue 1339. Also moved some of the high-level topics about the model (units, identifier, ...) to a common spot at the start of section 5.
0.0.45	10/06/2011	Doug	Moved EntityMetadata into the Entities section to resolve issue 1415. Add some clarifying text about routingGroups to resolve issue 1413. Replace status with state on select resources to resolve issue 1095. Define what 'ref' maps to for REST to resolve issue 1409. Add pious advice about some network properties to resolve issue 1259. Clean up some of the pointers in EventLog, Meter and Event to resolve issue 1383. Add start/stop operations to Meter to resolve issue 1237. Clarify the behavior when updating read-only properties to resolve issue 1118. Adding RoutingGroup as a new entity to resolve issue 1260.
0.0.45a	10/12/2011	Gil	Removed requirement to support TLS NULL cipher to resolve 1244. Updated diagrams to match changes in the text.
0.0.46	10/18/2011	Doug	Added recommendation to use partial updates to avoid overwriting changes to resolve 1360. Removed attr_regex feature to resolve 1418. Clarify just how opaque our URIs really are to resolve 1417.
0.0.47	10/26/2011	Doug	Added an 'extensibility" section to resolve issue 1356. Explain what a missing attributes in the serializations mean, and fix some Optional vs Mandatory flags to resolve issue 1114.
0.0.48	11/03/2011	Doug	Added text around our versioning scheme to resolve issue 1119.
0.0.49	11/09/2011	Doug	Add new scope text to resolve issue 1435. Fix the JSON serialization of "properties" to resolve issue 1436.
0.0.50	11/14/2011	Doug	Added support for enum/query support for collections to resolve issue 1405.
0.0.51	11/28/2011	Doug	Updated description of HTTP error code 501 to resolve issue 1442.
0.0.52	11/30/2011	Doug	Remove Bibliography to resolve issue 1443. Add networkInterfaces to MachineTemplate in machine.create() to resolve issue 1460. Added pause and resume operations to Machine to resolve issue 1434.
0.0.53	12/06/2011	Doug	s/using/used/ to resolve issue 1466. Removed the word "use" to resolve issue 1465. Removed section 6 to resolve issue 1464. Changed the use of the word "avoid" to resolve issue 1469. Add 'aspect' to Meter(Template) to resolve issue 1444. Tweaked the note at the end of MachineConfig to resolve issue 1454.
0.0.54	12/07/2011	Doug	Removed 'protocol' from Machine and Volume entities to resolve issue 1247. Complete the definition of SystemTemplate, add MachineAdminTemplate and RoutingGroupTemplate to resolve issue 1368. Updated the state values on many entities to resolve issue 1446. Use Job as the error response message and allow hierarchical jobs to resolve issue 1452. s/Network/VSP/ in some VSP attributes to resolve issue 1471.
0.0.55	12/08/2011	Doug	Add support for capturing a Machine to a MachineImage to resolve issue 1448.

4077