
**Cloud Infrastructure Management
Interface (CIMI) Model and RESTful
HTTP-based Protocol — An Interface for
Managing Cloud Infrastructure**

*Model d'interface de management de l'infrastructure du nuage
informatique (CIMI) et protocole RESTful basé HTTP — Une interface
pour le management de l'infrastructure du nuage informatique*

IECNORM.COM : Click to view the full PDF of ISO/IEC 19831:2015

IECNORM.COM : Click to view the full PDF of ISO/IEC 19831:2015



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2015

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
Case postale 56 • CH-1211 Geneva 20
Tel. + 41 22 749 01 11
Fax + 41 22 749 09 47
E-mail copyright@iso.org
Web www.iso.org

Published in Switzerland

CONTENTS

34	Foreword	7
35	1 Scope	9
36	1.1 Document structure	9
37	1.2 Document versioning scheme	9
38	1.3 Typographical conventions	9
39	2 Normative references	10
40	3 Terms and definitions	11
41	4 HTTP-based protocol	14
42	4.1 Introduction	14
43	4.1.1 Protocol evolution and client expectations	14
44	4.1.2 XML namespaces	14
45	4.1.3 URI space	14
46	4.1.4 Media types	15
47	4.1.5 Request headers	15
48	4.1.6 Request query parameters	15
49	4.2 Protocol operations	21
50	4.2.1 Common CRUD operations	22
51	4.3 OVF support	29
52	5 Model	30
53	5.1 Resource wrappers	30
54	5.2 Extensibility	31
55	5.3 Identifiers	31
56	5.4 Attribute constraints	32
57	5.5 Data types and their serialization	33
58	5.5.1 boolean	33
59	5.5.2 dateTime	33
60	5.5.3 duration	33
61	5.5.4 integer	34
62	5.5.5 string	34
63	5.5.6 ref	34
64	5.5.7 map	35
65	5.5.8 structure	35
66	5.5.9 byte[]	36
67	5.5.10 URI	36
68	5.5.11 Arrays	36
69	5.5.12 Collections	37
70	5.5.13 "Any" type	41
71	5.5.14 Empty attribute values	41
72	5.6 Units	41
73	5.7 Relationship semantics	42
74	5.8 Operations	42
75	5.9 Alternative model formats	43
76	5.10 Resources	43
77	5.10.1 Common attributes	43
78	5.11 Resource metadata	45
79	5.11.1 Serialization of attribute value constraints	49
80	5.11.2 Capabilities	51
81	5.11.3 ResourceMetadataCollection Resource	54
82	5.12 Cloud Entry Point	55
83	5.12.1 Operations	61
84	5.13 System Resources and relationships	61
85	5.13.1 System	62

86	5.13.2	SystemCollection Resource	80
87	5.13.3	SystemTemplate Resource	81
88	5.13.4	SystemTemplateCollection Resource	87
89	5.14	Machine Resources and relationships	88
90	5.14.1	Machine	89
91	5.14.2	MachineCollection	107
92	5.14.3	MachineTemplate	109
93	5.14.4	MachineTemplateCollection Resource	116
94	5.14.5	MachineConfiguration Resource	117
95	5.14.6	MachineConfigurationCollection Resource	119
96	5.14.7	MachineImage Resource	120
97	5.14.8	MachineImageCollection Resource	124
98	5.14.9	Credential Resource	125
99	5.14.10	CredentialCollection Resource	126
100	5.14.11	CredentialTemplate Resource	127
101	5.14.12	CredentialTemplateCollection Resource	128
102	5.15	Volume Resources and relationships	130
103	5.15.1	Volume	131
104	5.15.2	VolumeCollection Resource	135
105	5.15.3	VolumeTemplate Resource	136
106	5.15.4	VolumeTemplateCollection Resource	138
107	5.15.5	VolumeConfiguration Resource	139
108	5.15.6	VolumeConfigurationCollection Resource	141
109	5.15.7	VolumeImage Resource	142
110	5.15.8	VolumeImageCollection Resource	144
111	5.16	Network Resources and relationships	145
112	5.16.1	Network	145
113	5.16.2	NetworkCollection Resource	153
114	5.16.3	NetworkTemplate Resource	154
115	5.16.4	NetworkTemplateCollection Resource	156
116	5.16.5	NetworkConfiguration Resource	157
117	5.16.6	NetworkConfigurationCollection Resource	158
118	5.16.7	NetworkPort	160
119	5.16.8	NetworkPortCollection Resource	164
120	5.16.9	NetworkPortTemplate Resource	165
121	5.16.10	NetworkPortTemplateCollection Resource	168
122	5.16.11	NetworkPortConfiguration Resource	169
123	5.16.12	NetworkPortConfigurationCollection Resource	170
124	5.16.13	Address Resource	171
125	5.16.14	AddressCollection Resource	173
126	5.16.15	AddressTemplate Resource	174
127	5.16.16	AddressTemplateCollection Resource	176
128	5.16.17	ForwardingGroup Resource	177
129	5.16.18	ForwardingGroupCollection Resource	180
130	5.16.19	ForwardingGroupTemplate Resource	181
131	5.16.20	ForwardingGroupTemplateCollection Resource	182
132	5.17	Monitoring Resources and relationships	183
133	5.17.1	Job Resource	184
134	5.17.2	JobCollection Resource	188
135	5.17.3	Meter Resource	189
136	5.17.4	MeterCollection Resource	195
137	5.17.5	MeterTemplate Resource	196
138	5.17.6	MeterTemplateCollection Resource	197
139	5.17.7	MeterConfiguration Resource	198
140	5.17.8	MeterConfigurationCollection Resource	201
141	5.17.9	EventLog Resource	202

142	5.17.10 EventLogCollection Resource	205
143	5.17.11 EventLogTemplate Resource	206
144	5.17.12 EventLogTemplateCollection Resource	207
145	5.17.13 Event Resource	208
146	6 Security considerations	216
147	ANNEX A (normative) OVF support in CIMI	217
148	ANNEX B (informative) XML Schema	219
149	ANNEX C (informative) Change log	220
150		

151 Figures

152	Figure 1 - Cloud Entry Point	56
153	Figure 2 - System Resources	62
154	Figure 3 - Machine Resources	89
155	Figure 4 - Volume Resources	130
156	Figure 5 - Network Resources	145
157	Figure 6 - Monitoring Resources	184
158		

159 Tables

160	Table 1 – XML namespaces	14
161	Table 2 – Named structure	35
162	Table 3 – Converting a relative URI to an absolute URI	36
163	Table 4 – Numerical equivalents for attributes	42
164	Table 5 – Common attributes	43
165	Table 7 – Capability URIs	51
166	Table 8 – CloudEntryPoint attributes	56
167	Table 9 – System attributes	63
168	Table 10 – SystemSystem attributes	67
169	Table 11 – SystemMachine attributes	68
170	Table 12 – SystemCredential attributes	70
171	Table 13 – SystemVolume attributes	71
172	Table 14 – SystemNetwork attributes	72
173	Table 15 – SystemNetworkPort attributes	74
174	Table 16 – SystemAddress attributes	75
175	Table 17 – SystemForwardingGroup attributes	76
176	Table 18 – SystemTemplate attributes	82
177	Table 19 – Machine attributes	89
178	Table 20 – Disk attributes	93
179	Table 21 – MachineVolume attributes	94
180	Table 22 – MachineNetworkInterface attributes	96
181	Table 23 – MachineNetworkInterfaceAddress attributes	98
182	Table 24 – MachineSnapshot attributes	100
183	Table 25 – MachineTemplate attributes	109
184	Table 26 – MachineConfiguration attributes	117
185	Table 27 – MachineImage attributes	120

ISO/IEC 19831:2015 (E)

186	Table 28 – Credential attributes	125
187	Table 29 – UserName/Password attributes	125
188	Table 30 – Public key attributes	125
189	Table 31 – CredentialTemplate attributes	127
190	Table 32 – Volume attributes	131
191	Table 33 – VolumeVolumeImage attributes	133
192	Table 34 – VolumeTemplate attributes	136
193	Table 35 – VolumeConfiguration attributes	140
194	Table 36 – VolumeImage attributes	142
195	Table 37 – Network attributes	145
196	Table 38 – NetworkTemplate attributes	154
197	Table 39 – NetworkConfiguration attributes	157
198	Table 40 – NetworkPort attributes	160
199	Table 41 – NetworkPortTemplate attributes	165
200	Table 42 – NetworkPortConfiguration attributes	169
201	Table 43 – Address attributes	171
202	Table 44 – AddressTemplate attributes	174
203	Table 45 – ForwardingGroup attributes	178
204	Table 46 – ForwardingGroupNetwork attributes	179
205	Table 47 – ForwardingGroupTemplate attributes	181
206	Table 48 – Job attributes	185
207	Table 49 – Meter attributes	189
208	Table 50 – Sample attributes	192
209	Table 51 – MeterTemplate attributes	196
210	Table 52 – MeterConfiguration attributes	198
211	Table 53 – aspect URIs	200
212	Table 54 – EventLog attributes	202
213	Table 55 – EventLogTemplate attributes	206
214	Table 56 – Event attributes	208
215	Table 57 – type URIs	211
216		
217		

Foreword

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

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

Acknowledgments

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

Editors (past and present):

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

Contributors:

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

ISO/IEC 19831:2015 (E)

- 265 • David Lutterkort – Red Hat
- 266 • Fred Maciel – Hitachi, Ltd.
- 267 • Andreas Maier – IBM
- 268 • Ashok Malhotra – Oracle
- 269 • Jeff Mischkinsky – Oracle
- 270 • Jesus Molina – Fujitsu
- 271 • Efraim Moscovich – CA Technologies
- 272 • Bryan Murray – Hewlett-Packard Company
- 273 • Steven Neely – Cisco
- 274 • Ryuichi Ogawa – NEC Corporation
- 275 • John Parchem – Microsoft Corporation
- 276 • Shishir Pardikar – Citrix Systems Inc.
- 277 • Miguel Peñalvo – Telefónica
- 278 • Gilbert Pilz – Oracle
- 279 • Alvaro Polo – Telefónica
- 280 • Enrico Ronco – Telecom Italia
- 281 • Federico Rossini – Telecom Italia
- 282 • Matthew Rutkowski – IBM
- 283 • Tom Rutt – Fujitsu
- 284 • Hemal Shah – Broadcom
- 285 • Nihar Shah – Microsoft Corporation
- 286 • Alan Sill – Texas Tech University
- 287 • Zhexuan Song – Huawei
- 288 • Marvin Waschke – CA Technologies
- 289 • Eric Wells – Hitachi, Ltd.
- 290 • Jeff Wheeler – Huawei
- 291 • Maarten Wiggers – Fujitsu
- 292 • Daniel Wilson – Ericsson AB
- 293 • Steve Winkler – SAP AG
- 294 • Jack Yu – Oracle
- 295 • Aaron Zhang – Huawei
- 296 • HengLiang Zhang – Huawei
- 297

IECNORM.COM : Click to view the full PDF of ISO/IEC 19831:2015

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

1 Scope

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

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

1.1 Document structure

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

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

1.2 Document versioning scheme

This document adheres to the versioning scheme defined in clause 6.3 of [DSP4004](#).

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

1.3 Typographical conventions

This specification uses the following conventions:

In the narrative text of the specification:

- The regular or narrative font is Arial.
- Proper CIMI nouns such as Resource names, attribute names, operation names, reserved variable names are in Courier font. (e.g. Machine, volumes, \$expand). The plural form applies to such names to indicate several instances of such Resources (e.g. Machines, Systems).
- Examples text are in small Courier font and over a darker background.
- Quotes are used for any text that needs be distinguished as name or value of a particular concept (e.g. the "value constraints" attribute, the "Resource Name" column, a "false" value). In such cases, the string in quotes is always qualified by the concept it is an instance of.
- Names for CIMI concepts that may be common English words but have a very specific meaning in CIMI, are in narrative font but capitalized, e.g. Provider, Consumer, Resource, Collection.

ISO/IEC 19831:2015 (E)

When used in their common English sense they remain lower-case. However, CIMI modeling concepts that are used in a commonly understood manner remain in lower-case, such as: attribute, operation.

Inside tables describing the Resource data model:

- The narrative font is used for all terms, as the table structure qualifies them sufficiently.
- Where textual descriptions are introduced, the rules for narrative text apply.
- If a name is used as types (i.e., names of embedded structures as well as atomic types such as "integer", "string"), are in *italic*.
- Names that are just placeholders for actual names that may vary with each model instance, are between < > (e.g., <componentTemplate>).

Where the serialization of Resources is described, a pseudo-schema notation is used with the following conventions:

- 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)
- Vertical bars, "|", denote choice. For example, "a|b" means a choice between "a" and "b".
- Parentheses, "(" and ")", are used to indicate the scope of the operators "?", "*", "+" and "|".
- Ellipses (i.e., "...") indicate points of extensibility. Note 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.

Operation names Create, Update, Delete, Read are abstract operations that convey the semantics of concrete corresponding operations, such as HTTP methods or CIMI operation URIs.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated or versioned references, only the edition cited (including any corrigenda or DMTF update versions) applies. For references without a date or version, the latest published edition of the referenced document (including any corrigenda or DMTF update versions) applies.

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

DMTF DSP0243, *Open Virtualization Format Specification 1.1*,
http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.pdf

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

DMTF DSP4004, *DMTF Release Process 2.4*,
http://www.dmtf.org/sites/default/files/standards/documents/DSP4004_2.4.pdf

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

- 374 IEC 80000-13:2008, International Organization for Standardization, Geneva, Switzerland, *Quantities and*
 375 *units – Part 13: Information science and technology*, April 2008,
 376 http://www.iso.org/iso/catalogue_detail?csnumber=31898
- 377 IETF RFC2616, R. Fielding et al, *Hypertext Transfer Protocol -- HTTP/1.1*,
 378 <http://www.ietf.org/rfc/rfc2616.txt>
- 379 IETF RFC3986, T. Berners-Lee et al, *Uniform Resource Identifiers (URI): Generic Syntax*, August 1998,
 380 <http://www.ietf.org/rfc/rfc3986.txt>
- 381 IETF RFC4627, D. Crockford, *The application/json Media Type for JavaScript Object Notation (JSON)*,
 382 July 2006, <http://www.ietf.org/rfc/rfc4627.txt>
- 383 IETF RFC5246, T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*,
 384 <http://www.ietf.org/rfc/rfc5246.txt>
- 385 ISO 8601:2004, International Organization for Standardization, Geneva, Switzerland, *Data elements and*
 386 *interchange formats -- Information interchange -- Representation of dates and times*, March 2008,
 387 http://www.iso.org/iso/catalogue_detail.htm?csnumber=40874
- 388 ISO/IEC 14977:1996, Roger S. Scowen, *Extended BNF — A generic base standard*. Software
 389 Engineering Standards Symposium 1993.
 390 http://www.iso.org/iso/catalogue_detail?csnumber=26153
- 391 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,
 392 <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>
- 393 NIST Special Publication 800-145, Peter Mell and Timothy Grance, *The NIST Definition of Cloud*
 394 *Computing*, Sept. 2011, <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- 395 NIST Special Publication 500-292, Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee
 396 Badger and Dawn Leaf, *NIST Cloud Computing Reference Architecture*, Sept. 2011,
 397 [http://collaborate.nist.gov/twiki-cloud-](http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/ReferenceArchitectureTaxonomy/NIST_SP_500-292_-_090611.pdf)
 398 [computing/pub/CloudComputing/ReferenceArchitectureTaxonomy/NIST_SP_500-292_-_090611.pdf](http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/ReferenceArchitectureTaxonomy/NIST_SP_500-292_-_090611.pdf)
- 399 Representational State Transfer, Roy Fielding, Doctoral dissertation, University of California, *Architectural*
 400 *Styles and the Design of Network-based Software Architectures (Chapter 5)*, 2000,
 401 http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- 402 XMLSchema - Part 1, World Wide Web Consortium (W3C) Recommendation, H. Thompson, et al.,
 403 Editors, *XML Schema Part 1: Structures Second Edition*, 28 October 2004,
 404 <http://www.w3.org/TR/xmlschema-1/>
- 405 XMLSchema - Part 2, World Wide Web Consortium (W3C) Recommendation, P. Biron, A. Malhotra,
 406 Editors, *XML Schema Part 2: Datatypes (Second Edition)*, 28 October 2004,
 407 <http://www.w3.org/TR/xmlschema-2/>

408 3 Terms and definitions

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

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

ISO/IEC 19831:2015 (E)

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

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

The terms defined in [DSP4004](#), [DSP0223](#), and [DSP1001](#) apply to this document. The following additional terms are used in this document.

3.1 authentication

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

3.2 authorization

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

3.3 cloud

Synonymous with "cloud computing" as defined in section 2 of the NIST Definition of Cloud Computing [\[SP800-145\]](#).

3.4 Cloud Service Consumer

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

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

3.5 Cloud Service Provider

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

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

3.6

Collection

A particular kind of Resource that contains a collection of other Resources and has a representation and serialization defined in this specification. Synonym for “CIMI collection”.

3.7

Configuration

A set of metadata, the values of which serve as the parameters of a discrete conformation of a specific type of virtual resource.

3.8

Infrastructure as a Service (IaaS)

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

3.9

message confidentiality

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

3.10

message integrity

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

3.11

Resource

A representation of an entity managed by the [Cloud Service] Provider that is generally available to the [Cloud Service] Consumer to access or operate on by the way of the interface described in this specification. Synonym for “CIMI resource”.

3.12

Template

Synonym for “CIMI template”. A Resource that represents the set of metadata and instructions used to instantiate some other Resource (e.g., a MachineTemplate is used to create Machines). Templates may aggregate other metadata Resources such as other Templates, Configurations, and Images. For example, a MachineTemplate refers to a MachineConfiguration and a MachineImage.

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

1. By reference - allow Consumers to reference a Template (that exists as a Resource in the Provider) as part of the instantiation operation.
2. By value - allow Consumers to dynamically provide the Template information as part of the instantiation operation.
3. Reference with overrides - allow Consumers to reference a Template (that exists as a Resource in the Provider) and provide additional values that override the attributes of that Template as part of the instantiation operation.

4 HTTP-based protocol

4.1 Introduction

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

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

4.1.1 Protocol evolution and client expectations

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

1. Clients shall not assume that the serializations shown in responses in this specification are complete. In particular, clients shall accept responses that contain data mixed in with the serializations shown here, and shall ignore such data. However, per clause 4.2.1.3, clients shall include unknown data in PUT requests to update Resources.
2. Clients shall not assume anything about the operations supported by a server. They are expected to discover operations that are supported (and permissible) by navigating to Resources from the cloud entry point. The serializations of Resources encountered indicate which operations are supported by the server.

4.1.2 XML namespaces

Table 1 lists the XML namespaces that are used in this specification. The choice of any namespace prefix is arbitrary and not semantically significant.

Table 1 – XML namespaces

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

4.1.3 URI space

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

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

```

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

```

4.1.4 Media types

In this specification, Resource and response representations are encoded either in JSON, as specified in [RFC4627](#) or in XML. If serialized in JSON, the media-type for CIMI resources shall be "application/json". If serialized in XML, the media-type shall be "application/xml".

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

Note that this requirement applies even if the \$select attribute is used to subset the Resource being acted upon.

In the XML serialization of Collection representations sent by Providers there shall be a resourceURI attribute, as shown in the example XML serialization of Collections in clause 5.5.12.

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

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

4.1.5 Request headers

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

4.1.6 Request query parameters

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

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

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

4.1.6.1 Filtering Collections

If retrieving the representation of a Collection, Consumers may include the `$filter` query parameter to reduce the number of entries of the Collection that are returned based on the data within the entries of the Collection. Providers shall interpret and process the `$filter` query parameter as described in this section. The `$filter` parameter shall be of the form:

```
?$filter=expression
```

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

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

Where PropExpr is used to find Resources that contain a property with a certain key/value combination. The key is the StringValue within the square brackets ([]) and the value is the StringValue after the Op. The Resource shall be considered to satisfy the search criteria if any of the properties in the Resources match the specified PropExpr.

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

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

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

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

Examples:

In the following examples, the following sample base URIs are used.

The URI to the MachineCollection of the Cloud Entry Point is as follows:

```
/machines
```

The URI to a Machine is as follows:

```
/machines/123
```

The URI to the DiskCollection of a Machine is as follows:

```
/machines/123/disks
```

The URI to the MachineVolumeCollection of a Machine is as follows:

```
/machines/123/volumes
```

To filter the MachineCollection so that just Machines with a "name" attribute of "mine" are returned, use the following filter:

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

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

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

If the \$filter attribute is used, the Collection's "count" attribute shall contain the number of Resources matching the filter expression.

4.1.6.2 Subsetting Collections

If retrieving the representation of a Collection, Consumers may include query parameters to subset the number of entities of the Collection that are returned. Providers shall interpret and process these query parameters as described in this section. While the previous clause discussed how to perform a filter over the data within the Collection, this clause uses ordinal position within the Collection to achieve the desired reduction.

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

```
?$first=number
```

```
?$last=number
```

Where "\$first" indicates the (1-based) ordinal position of the first entity of the Collection to return and "\$last" indicates the (1-based) ordinal position of the last entity of the Collection to return. Consumers are not required to use both at the same time. If \$first is specified but \$last is not, the implied value for \$last shall be the ordinal position of the last entity in the Collection. Conversely, if \$last is specified but \$first is not, the implied value for \$first shall be 1.

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

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

4.1.6.3 Subsetting Resources

If retrieving the representation of a Resource, Consumers may include the \$select query parameter to specify a subset of the Resource to be acted upon. Providers shall interpret and process this query parameter as described in this section. This subsetting shall have the semantic equivalence of

referencing a different Resource whose attributes are a subset of the original Resource as specified by the attribute names listed in the `$select` query parameter. The format of a `$select` query parameter is:

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

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

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

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

is equivalent to:

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

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

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

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

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

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

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

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

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

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

returns a collection of the `Disks` associated with a `Machine` but each entity of the collection just has the `name` and `capacity` attributes and nothing else, not even the `operations` or `id` attributes.

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

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

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

4.1.6.4 Expanding references

If retrieving the representation of a Resource, Consumers may include the \$expand query parameter to specify which of the top-level "reference" attributes of the Resource shall be "expanded". Providers shall interpret and process this query parameter as described in this section. To expand a reference means that the attributes of the Resource being referenced shall be included in the serialization of that attribute. This feature allows for a more optimized retrieval of Resources.

The serialization shall be performed as follows:

JSON serialization:

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

shall be expanded to be:

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

XML serialization:

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

shall be expanded to be:

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

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

The format of a \$expand query parameter shall be:

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

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

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

If the Resource being retrieved is a Collection, the attribute names listed in the \$expand shall apply to the attributes of the entities within the Collection. For example, specifying:

```
?$expand=volumes
```

if retrieving the MachineCollection has the same net effect as applying the "expand" semantics to the specified attribute ("volumes" in this example) of each Machine within the Collection. To be clear, \$expand acts on the attributes of the Resources in the Collection, not on the wrapping Collection Resource itself.

4.1.6.5 Specifying the Resource format

If retrieving the representation of a Resource, the HTTP Accept header is used to specify the encoding style of the response. While it is recommended that Consumers use the Accept header, there might be situations where Consumers are unable to control the values specified in that header. In these cases Consumers may use the `$format` query parameter to override the Accept header values. Providers shall interpret and process the `$format` query parameter as described in this section.

The `$format` parameter shall be of the form:

```
?$format=encoding
```

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

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

4.1.6.6 Sorting Collections

If retrieving the representation of a Collection, Consumers may include the `$orderby` query parameter to sort the entries of the Collection that are returned based on different attributes or in a different order (descending). Providers shall interpret and process the `$orderby` query parameter as described in this section. The `$orderby` parameter shall be of the form:

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

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

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

The sort shall be performed based on the attribute type.

The following rules apply to the ascending sort order:

- boolean – 'false' shall come before 'true'.
- dateTime – Earlier datetime shall come before a later datetime.
- duration – A shorter duration shall come before a longer duration.
- integer – Smaller integer shall come before larger integers. Negative integers shall come before positive integers.
- string – Ordering is based on Unicode/UTF-8 sort order.

For the desc sort order, the reverse of the above shall be performed.

Examples:

To sort the result set of the `MachinesCollection` Resource on the "created" attribute in descending order, the following expression would be used:

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

To sort the result set of the `MachinesCollection` Resource on the "cpu" attribute in descending order, followed by the "memory" attribute in ascending order, the following expression would be used:

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

4.1.6.7 Response headers

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

4.1.6.8 Job header

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

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

In cases where an error occurs during the processing of a request, the Provider shall include a representation of a Job Resource describing the status of the failed operation. This representation of a Job shall be included even in cases where the Provider does not normally support Job Resources 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. If Jobs are not supported in general, any of the references in the Job representation (e.g., "id" or the "href" for `nestedJobs`) shall be empty paths (i.e., "") and the `nestedJobs` array shall be expanded (see 4.1.6.4) to inline the representation of the pseudo subordinate Jobs.

4.1.6.9 ETag support

An ETag header may be provided by a Provider with each Resource as specified in [RFC2616](#). If a Provider does provide an ETag header, it shall also support If-Match header processing on behalf of the Consumer.

4.2 Protocol operations

This clause defines the set of common HTTP operations that a Provider may 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 follow a similar high-level pattern, but each operation is allowed to have slight variations to accommodate its specific needs. The specifics of these special operations are detailed within the clause that defines the Resource.

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

JSON serialization:

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

XML serialization:

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

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

JSON serialization:

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

XML serialization:

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

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

4.2.1 Common CRUD operations

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

4.2.1.1 Creating a new Resource

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

The HTTP POST request shall include:

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

For example, the request can be:

```
POST <addURI> HTTP/1.1
Host: <hostname>
```

```

Accept: application/(json+xml)
Content-Type: application/(json+xml)
Content-Length: <length>

<serialization of request to create a new resource>

```

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

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

```

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

```

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

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

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

```

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

```

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

```

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

```

}

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

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

In both patterns (1) and (2) an element corresponding to the Resource Template (here identified generically as "resourceTemplate" e.g., MachineTemplate) is specifying the Template to be used, either by value (1) or by reference (2).

Direct setting of attributes in the new Resource:

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

The semantics shall be same as of a partial update of the Resource for these attributes (described in a next subsection), immediately following the Resource creation from the Template alone.

Defining or referring to the Resource Template:

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

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

- Any top-level attribute of complex or simple type in the referred Template shall be overridden by providing its name/value pair in the create request inside the resourceTemplate element and immediately under it. For a top-level attribute of complex type (e.g., arrays, Collections, structures), the provided complex value shall also set all underlying attributes – e.g., array elements.
- The semantics shall be same as of modifying (overriding) parts of the referred Template just before it is used for instantiation, but these overrides shall not persist in the referred Template and shall only concern this particular instantiation.

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

```
"attribute": null
```

for the attribute in the JSON serialization, or

```
<attribute/>
```

in the XML serialization for that attribute.

Examples:

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

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate ",
  "name": "myMachine123",
  "description": "A machine to be connected to a pre-existing network",
  "machineTemplate": {
    <here a template passed "by value" i.e. the attribute/value pairs for the
    MachineTemplate template. An example is of the networkInterfaces below: >
```

```

949     "networkInterfaces": [
950       { "addresses": [ { "address": { "href": "http://example.com/addresses/add1"
951     }}, { "address": { "href": "http://example.com/addresses/add2" } } ],
952         "network": { "href": "http://example.com/networks/net1" },
953         "state": "ACTIVE" }
954     ]
955   }
956 }
957 }

```

In the previous example:

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

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

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

```

965 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate ",
966   "name": "myMachine456",
967   "description": "A machine connected to a pre-existing volume",
968   "machineTemplate": { "href": "http://example.com/machineTemplates/72000",
969     "credential": { "href": "http://example.com/myCredential" }
970   },
971   "networkInterfaces": [
972     { "addresses": [ { "address": { "href": "http://example.com/addresses/add4"
973   }}, { "address": { "href": "http://example.com/addresses/add5" } } ],
974       "network": { "href": "http://example.com/networks/net1" },
975       "state": "ACTIVE" }
976   ]
977 }
978 }
979 }

```

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

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

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

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

- HTTP Location header with a reference to the new Resource

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

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

For example, the response can be:

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

<serialization of new resource>
```

4.2.1.2 Retrieving a representation of a Resource

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

For example, the request can be:

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

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

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

For example, the response can be:

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

<serialization of resource>
```

4.2.1.3 Updating a Resource

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

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

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

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

The HTTP PUT request shall include:

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

For example, the request can be:

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

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

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

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

For example, the response can be:

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

<serialization of updated resource>
```

4.2.1.3.1 Partial updates to a Resource

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

To update only certain top-level attributes of a Resource, a Consumer may include only the altered attributes in the representation of the Resource within the HTTP request body. If this request is made, 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 shall be of the form:

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

Only the attributes listed in the URI's query parameters shall be modified; attributes not listed in the URI shall not be directly modified by the request. Note that this circumstance 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 shall be reset to a Resource specific value (e.g., removed).

From an HTTP perspective, the updated subsetting 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 changed.

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

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

```
PUT /machines/myMachine?$select=name,description HTTP/1.1
Host: <hostname>
Accept: application/xml
Content-Type: application/xml
Content-Length: <length>

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

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

4.2.1.4 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 is the same as the URI of Resource itself. Retrieving the Resource representation shall include a "delete" operation, which contains the deleteURI that is to be used, if the requester is allowed to delete the Resource.

For example, the request can be:

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

If the Resource has a State attribute, its value shall be "DELETING", while the Provider is processing this operation.

For example, the response can be:

```
HTTP/1.1 200 OK
```

4.2.1.5 Other operations

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

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

The request shall be of the following form:

```
POST <operationLinkURI> HTTP/1.1
```

```

Host: <hostname>
Accept: application/(json|xml)
Content-Type: application/(json|xml)
Content-Length: <length>

<serialization of request to perform some action>

```

The form of the response varies depending on the operation and is defined by the operation itself.

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

4.2.1.6 Synchronous operations

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

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

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

4.2.1.7 Asynchronous operations

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

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

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

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

4.3 OVF support

The *Open Virtualization Format (OVF) Specification* ([DSP0243](#)) describes an open, secure, portable, efficient, and extensible format for the packaging and distribution of software to be run in virtual

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

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

OVF support is covered in more detail in ANNEX A.

5 Model

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

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

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

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

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

5.1 Resource wrappers

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

JSON serialization:

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

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

XML serialization:

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

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

</MyResource>

5.2 Extensibility

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

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

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

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

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

If a Provider receives a message containing an unknown or unsupported attribute, it shall reject the request. If a Consumer receives a message containing an unknown or unsupported attribute, it shall silently ignore the attribute. However, Consumers are required to include those attributes in messages sent back to the Provider. Note in these cases the Consumer is not required to understand or process the unsupported attribute, but merely echo it back to the Provider.

5.3 Identifiers

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

- Identifier names shall be treated as case sensitive.
- Identifier names shall only use the following set of characters:
 - o Uppercase ASCII (U+0041 through U+005A)
 - o Lowercase ASCII (U+0061 through U+007A)
 - o Digits (U+0030 through U+0039)
 - o Underscore (U+005F)
- Identifier names shall not start with a Digit (U+0030 through U+0039).

1233 Note that these rules do not apply to the "name" common attribute defined in clause 5.10.1.

1234 5.4 Attribute constraints

1235 Each attribute of the Resources in the CIMI model is augmented by a set of constraints that further qualify
1236 the attribute that is being defined. For each attribute, there is a Provider and a Consumer set of
1237 constraints because each might differ. The following constraints are possible:

1238 **support optional:**

1239 This constraint indicates that support for this attribute is optional. If supported, Providers should advertise
1240 its support through ResourceMetadata. See clause 5.2 for information concerning the processing of
1241 unsupported and unknown attributes. See clause 5.5.14 regarding empty attribute values.

1242 Non-empty Consumer supported writeable (i.e., read-write and write-only) attributes shall always be
1243 included as part of the Resource representation sent from Consumers to Providers, including create
1244 requests.

1245 Non-empty Provider-supported attributes shall always be included as part of the Resource representation
1246 sent from Providers to Consumers.

1247 **support mandatory:**

1248 This constraint indicates that support for this attribute is required by compliant implementations. If present
1249 on a nested attribute, this attribute is required to be supported only if the parent attribute is supported.
1250 See clause 5.5.14 regarding empty attribute values.

1251 Non-empty mandatory writeable (i.e., read-write and write-only) attributes shall always be included as part
1252 of the Resource representation sent from Consumers to Providers - including create requests.

1253 Non-empty Provider mandatory attributes shall always be included as part of the Resource representation
1254 sent from Providers to Consumers.

1255 **immutable:**

1256 This Provider constraint indicates that the attribute, once set, shall never change for the lifetime of the
1257 Resource.

1258 **mutable:**

1259 This Provider constraint indicates that the attribute may be modified. Providers shall always have the
1260 ability to modify these attributes. Whether Consumers have the ability to modify these attributes shall be
1261 indicated by the read-only, read-write, and write-only constraints.

1262 **read-only:**

1263 This Consumer constraint indicates that the attribute may be retrieved but not updated by Consumers.
1264 Read-only attributes are not required to appear in the serialization of Resources in create or update
1265 request messages. If present, they shall be silently ignored by the Provider. Read-only attributes shall
1266 appear in the serialization of Resources sent from Providers.

1267 **read-write:**

1268 This Consumer constraint indicates that the attribute may be retrieved and/or updated by Consumers.
1269 Read-write attributes shall appear in the serialization of Resources sent to and from Providers. Providers
1270 may further constrain whether Consumers can update these attributes and should indicate this by the way
1271 of ResourceMetadata.

write-only:

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

5.5 Data types and their serialization

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

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

5.5.1 boolean

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

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

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

5.5.2 dateTime

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

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

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

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

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

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

5.5.3 duration

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

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

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

1307 **5.5.4 integer**

1308 A value as defined by xs:integer per [XML Schema – Part 2](#). Any constraints on the specific ranges
 1309 allowed for any particular attribute shall be specified by that attribute's definition or at runtime by the
 1310 Provider by the way of the metadata discovery mechanisms defined by this specification.

1311 If serialized in JSON these values shall be of JSON type: *number*

1312 If serialized in XML these values shall be of XML Schema type: *xs:integer*

1313 **5.5.5 string**

1314 A value as defined by xs:string per [XML Schema – Part 2](#). Any constraints on this type for any particular
 1315 attribute shall be specified by that attribute's definition or at runtime by the Provider by the way of the
 1316 metadata discovery mechanisms defined by this specification.

1317 If serialized in JSON these values shall be of JSON type: *string*

1318 If serialized in XML these values shall be of XML Schema type: *xs:string*

1319 If serializing an attribute of type string, the serialization shall omit this attribute in case of an empty string.

1320 **5.5.6 ref**

1321 A reference to another Resource.

1322 References allow for Consumers to navigate to Resources. By starting at the Cloud Entry Point and
 1323 following the references that appear in the retrieved Resources, Consumers are able to recursively
 1324 discover and navigate to all other Resources.

1325 As a general rule, if an attribute is of type "ref", its value shall be held by an attribute named "href"
 1326 (both in JSON and XML).

1327 **JSON serialization:**

1328 In the JSON serialization the href property appears as of type "string." If an attribute is of type "ref",
 1329 the name of this attribute shall appear as a key, with the href property as it a nested value. For example,
 1330 a Resource attribute "myvolume" of type "ref" is serialized as:

1331

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

1332 **XML serialization:**

1333 In the XML serialization the href attribute appears as type "xs:anyURI." If an attribute is of type "ref",
 1334 the name of this attribute shall appear as name of an XML element with the href property as an (XML)
 1335 attribute. For example, a Resource attribute "myvolume" of type "ref" is serialized as:

1336

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

1337 References in both JSON and XML have an extensibility point that allows for additional information (such
 1338 as the target Resource to be included "by value") if supported. For convenience, the JSON and XML
 1339 representations, as shown above, exclude the implicit extensibility points that would allow for the
 1340 attributes of the target Resource to be included if desired. So, more accurately the above representations
 1341 might be written as follows:

1342 For JSON:

1343

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

1344 and in XML:

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

However, for brevity the extensibility points are excluded from the serialization of the Resources.

5.5.7 map

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

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

5.5.8 structure

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

A nested structure can be considered a complex type definition. Structures may be named or unnamed. Table 2 is an example of named structure:

Table 2 – Named structure

Name	summary	
Attribute	Type	Description
low	number	Number of "low" occurrences
medium	number	Number of "medium" occurrences
high	number	Number of "high" occurrences
critical	number	Number of "critical" occurrences

JSON serialization:

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

```
"systemIncidents": {
  "low": number,
  "medium": number,
  "high": number,
  "critical": number
}
```

XML serialization:

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

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

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

5.5.9 byte[]

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

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

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

5.5.10 URI

The format and syntax of the attributes of type "URI" is defined by [RFC3986](#).

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

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

The algorithm used for converting a relative URI to an absolute URI shall be as described in section 5.2 of [RFC3986](#). Table 3 illustrates how relative URIs are resolved against base URIs:

Table 3 – Converting a relative URI to an absolute URI

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

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

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

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

5.5.11 Arrays

An array represents an ordered list of items of the same type. An array shall always appear as an attribute of a Resource, and is only accessible as such (it is not a separately addressable Resource). If a Resource is deleted, the items in its arrays shall also be deleted. However, in case these items were just references to other Resources, these referred Resources are not affected. (See the semantics of references in 5.7.)

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

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

JSON serialization:

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

```

1412 "things" : [
1413     { ... }, +
1414 ] ?

```

If the items in the array are structures, the structure name shall not be present in the JSON serialization.

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

```

1419 "things": [
1420     { "href": string }, +
1421 ] ?

```

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

XML serialization:

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

```

1432 <thing>
1433     ...
1434 </thing> *

```

There is no wrapper element for an array in XML.

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

```

1440 <thing href="xs:anyURI"/> +

```

5.5.12 Collections

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

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

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

ISO/IEC 19831:2015 (E)

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

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

The Resources in a Collection are of two kinds:

- an infrastructure Resource (such as those listed in the Cloud Entry Point, or those embedded in an entity such as the `disks` inside a `Machine`)
- an intermediary Resource that holds a reference to an infrastructure Resource, called the "target Resource".

By convention, intermediary Resources have a name that concatenates the name of the Resource owning the Collection, with the name of the target Resource, e.g., `MachineVolume` is the name of the intermediary Resource that is used to connect a `Machine` to a `Volume`.

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

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

The serialization of Collections shall adhere to the following pattern:

JSON serialization:

```
{ "resourceURI": string,  
  "id": string,
```

```

1497     "count": number,
1498     "resourceSpecificGroupingName": [
1499         { "resourceURI": string,
1500           "id": string,
1501           "name": string, ?
1502           "description": string, ?
1503           "created": string, ?
1504           "updated": string, ?
1505           "properties": { string: string, + }, ?
1506           ... entry specific data ...
1507           "operations": [
1508               { "rel": "edit", "href": string }, ?
1509               { "rel": "delete", "href": string } ?
1510           ] ?
1511           ...
1512         } +
1513     ], ?
1514     "operations": [ { "rel": "add", "href": string } ? ]
1515     ...
1516 }

```

XML serialization:

```

1518 <Collection resourceURI="xs:anyURI" xmlns="http://schemas.dmtf.org/cimi/1">
1519     <id> xs:anyURI </id>
1520     <count> xs:integer </count>
1521     <ResourceSpecificElementName>
1522         <id> xs:anyURI </id>
1523         <name> xs:string </name> ?
1524         <description> xs:string </description> ?
1525         <created> xs:dateTime </created> ?
1526         <updated> xs:dateTime </updated> ?
1527         <property key="xs:string"> xs:string </property> *
1528         ... entry specific data ...
1529         <operation rel="edit" href="xs:anyURI"/> ?
1530         <operation rel="delete" href="xs:anyURI"/> ?
1531         <xs:any>*
1532     </ResourceSpecificElementName> *
1533     <operation rel="add" href="xs:anyURI"/> ?
1534     <xs:any>*
1535 </Collection>

```

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

5.5.12.1 Adding items to Collections

Invoking the "add" operation of a Collection shall add a new Resource to the Collection. The contents of the request body shall be either a representation of the new Resource being added to the Collection, or a representation of the Template associated with the new Resource being created. This specification indicates which Resources require the use of a Template.

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

JSON serialization:

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

XML serialization:

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

Note that while deleting this type of Resource from the Collection deletes and removes the Resource from the Collection, it shall not delete the referenced target Resource itself - in this case the Volume.

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

- The "common attributes" as defined by clause 5.10.1.
- The Resource specific data needed to create it. This data shall either be a reference to the Resource-specific Template Resource or the Resource-specific Template Resource itself inlined.
- In the XML case, a wrapper element (named after the pattern <ResourceNameCreate>).

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

JSON serialization:

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

XML serialization:

```

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

```

The MachineCollection has a new Machine:

JSON serialization:

```

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

```

XML serialization:

```

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

```

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

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

5.5.13 "Any" type

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

5.5.14 Empty attribute values

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

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

5.6 Units

Some of the Resources defined by this specification have attributes that describe an amount of something that belongs to, or is associated with, that Resource. For example, the `Machine` Resource

has a **memory** attribute that describes "the size of the memory allocated to this machine." The allowable units of these attributes are listed in Table 4. Their meaning is defined in [IEC 80000-13:2008](#). Their numerical equivalents are provided here for convenience:

Table 4 – Numerical equivalents for attributes

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
yottabyte	10 ²⁴	yobibyte	2 ⁸⁰

5.7 Relationship semantics

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

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

5.8 Operations

All Resource operations defined by this specification are optional for Providers to support. Consumers, by the way of examination of a Resource's ResourceMetadata, can determine which operations are supported. However, even for those operations that are supported Consumers still need to examine each Resource's representation to determine which operations are supported at that moment. Whether an operation is supported is based on a number of factors, including state of the Resource and access control rights of the Consumer. Also see clause 4.2. Operations and states are coupled; i.e., if implementing a state-changing Resource operation defined in this specification, the corresponding state(s) shall also be implemented. See the Resource-specific "Operations" clauses for additional detail.

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

- an operation is performed on this Resource and this operation requires a state change, or
- an error occurred, in this case the "State" attribute shall obtain the value "ERROR".

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

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

- A new operation that starts from a CIMI-defined state, or leads to a CIMI-defined state, or both. In the latter case, if a CIMI-defined operation already exists for this transition between two CIMI-defined states, it shall also be supported by the Provider in addition to the new operation.
- A new Resource state. In that case, a new operation that leads to that state shall also be created. In other words, a Provider-defined operation has to be performed before a Provider-defined state can be reached."

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

5.9 Alternative model formats

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

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

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

5.10 Resources

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

5.10.1 Common attributes

Except for ResourceMetadata and Collection Resources (see 5.5.12), the Resources described by this document share the following common attributes; see Table 5. There are different requirements for primary and secondary CIMI resources. All Resources that are element types of Collections in the CloudEntryPoint shall be primary CIMI resources. All other Resources shall be secondary CIMI resources. An exception to this rule is that the CloudEntryPoint shall be considered a primary Resource.

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

Table 5 – Common attributes

Attribute	Type	Description
id	URI	The unique URI identifying this Resource; assigned upon Resource creation. This attribute value shall be unique in the Provider's cloud. Constraints for primary and secondary Resources: Provider: support mandatory; immutable Consumer: support mandatory; read-only
name	string	The human-readable name of this Resource; assigned by the creator as a part of the Resource creation input. Constraints for primary Resources: Provider: support mandatory; mutable Consumer: support optional; read-write Constraints for secondary Resources: Provider: support optional; mutable Consumer: support optional; read-write
description	string	The human-readable description of this Resource; assigned by the creator as a part of the Resource creation input. Constraints for primary Resources: Provider: support mandatory; mutable Consumer: support optional; read-write Constraints for secondary Resources: Provider: support optional; mutable Consumer: support optional; read-write
created	dateTime	The timestamp when this Resource was created. The format should be unambiguous, and the value is immutable .

Attribute	Type	Description									
		Constraints for primary and secondary Resources: Provider: support optional; immutable Consumer: support optional; read-only									
updated	<i>dateTime</i>	The time at which the last explicit attribute update was made on the Resource. Note, while operations, such as "stop", do implicitly modify the 'state' attribute, they do not change the 'updated_time'. Constraints for primary and secondary Resources: Provider: support optional; mutable Consumer: support optional; read-only									
properties	<i>map</i>	A map of key/value pairs (each entry called a "property"), some of which may control one or more aspects this Resource. Properties may also serve as an extension point, allowing Consumers to record additional information about the Resource. The same "key" shall not be used more than once within a "properties" attribute. Each property shall contain the following nested data: <table border="1"> <thead> <tr> <th>Name</th><th>Type</th><th>Description</th></tr> </thead> <tbody> <tr> <td>key</td><td><i>string</i></td><td> The name of the property. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write </td></tr> <tr> <td>value</td><td><i>string</i></td><td> The value of the property. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write </td></tr> </tbody> </table> Constraints for primary Resources: Provider: support mandatory; mutable Consumer: support optional; read-write Constraints for secondary Resources: Provider: support optional; mutable Consumer: support optional; read-write	Name	Type	Description	key	<i>string</i>	The name of the property. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	value	<i>string</i>	The value of the property. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
Name	Type	Description									
key	<i>string</i>	The name of the property. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write									
value	<i>string</i>	The value of the property. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write									

1673 The following pseudo-schemas describe the serialization of these attributes in both JSON and XML:

1674 **JSON serialization:**

```

1675 "id": string,
1676 "name": string, ?
1677 "description": string, ?
1678 "created": string, ?
1679 "updated": string, ?
1680 "properties": { string: string, + }, ?

```

1681 **XML serialization:**

```

1682 <id> xs:anyURI </id>
1683 <name> xs:string </name> ?
1684 <description> xs:string </description> ?
1685 <created> xs:dateTime </created> ?
1686 <updated> xs:dateTime </updated> ?
1687 <property key="xs:string"> xs:string </property> *

```

5.11 Resource metadata

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

Note that while this specification declares the ResourceMetadata as mutable attributes, it is expected that only administrative users associated with the Provider will update them. Consequently they remain read-only for Consumers.

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

Table 6 – ResourceMetadata attributes

Name	ResourceMetadata																			
Type URI	http://schemas.dmtf.org/cimi/1/ResourceMetadata																			
Attribute	Type	Description																		
id	URI	The unique URI identifying this Resource; assigned upon Resource creation. This attribute value is immutable , and shall be unique in the Provider's cloud. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only																		
typeURI	URI	A unique URI associated with, and denoting, the described Resource type. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
name	string	The name of the described Resource type. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
attributes	attribute[]	A set of Provider-defined metadata that can be used by clients to discover any metadata associated with each attribute of the described Resource type, including the set of extension attributes not defined in this specification. Each attribute shall contain the following nested data: <table> <tr> <th>Name</th><th colspan="2">attribute</th></tr> <tr> <th>Data</th><th>Type</th><th>Description</th></tr> <tr> <td>name</td><td>string</td><td>The name of the attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td></tr> <tr> <td>namespace</td><td>URI</td><td>The namespace in which this attribute is defined. It is recommended that a dereference of this URI returns information about the attribute. This shall not be present if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td></tr> <tr> <td>type</td><td>string</td><td>The data type of the attribute. This shall not be present if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI-defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td></tr> <tr> <td>required</td><td>boolean</td><td>Indicates whether this Resource requires this attribute</td></tr> </table>	Name	attribute		Data	Type	Description	name	string	The name of the attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	namespace	URI	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 if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	type	string	The data type of the attribute. This shall not be present if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI-defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	required	boolean	Indicates whether this Resource requires this attribute
Name	attribute																			
Data	Type	Description																		
name	string	The name of the attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
namespace	URI	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 if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
type	string	The data type of the attribute. This shall not be present if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI-defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
required	boolean	Indicates whether this Resource requires this attribute																		

Name	ResourceMetadata																				
Type URI	http://schemas.dmtf.org/cimi/1/ResourceMetadata																				
Attribute	Type	Description																			
			to be present. If absent the implied value is "false." Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
		value constraints	any Type-specific data that describes any constraints on values of this attribute. If absent, there are no constraints. Note that the serialization of these "value constraints" shall be determined by the type of the attribute; see clause 5.11.1. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
		Constraints: Provider: support optional; mutable Consumer: support optional; read-write																			
capabilities	capability[]	A set of Provider-defined metadata that can be used by Consumer to discover any capability or feature provided by this Provider. Each capability shall contain the following nested data: <table><tr><td>Name</td><td colspan="2">capability</td></tr><tr><td>Data</td><td>Type</td><td>Description</td></tr><tr><td>name</td><td>string</td><td>The name of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</td></tr><tr><td>uri</td><td>URI</td><td>A URI that uniquely identifies the capability at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td></tr><tr><td>description</td><td>string</td><td>The human-readable description of the semantic of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</td></tr><tr><td>value</td><td>any</td><td>The value of the capability. The specific type varies depending on the definition of the capability. If not present the capability defaults to a "boolean" type with a value of "true" indicating that the specific capability is supported by the Provider. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td></tr></table> Constraints: Provider: support optional; mutable Consumer: support optional; read-write		Name	capability		Data	Type	Description	name	string	The name of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write	uri	URI	A URI that uniquely identifies the capability at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	description	string	The human-readable description of the semantic of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write	value	any	The value of the capability. The specific type varies depending on the definition of the capability. If not present the capability defaults to a "boolean" type with a value of "true" indicating that the specific capability is supported by the Provider. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
Name	capability																				
Data	Type	Description																			
name	string	The name of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write																			
uri	URI	A URI that uniquely identifies the capability at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																			
description	string	The human-readable description of the semantic of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write																			
value	any	The value of the capability. The specific type varies depending on the definition of the capability. If not present the capability defaults to a "boolean" type with a value of "true" indicating that the specific capability is supported by the Provider. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																			
actions	action[]	A set of Provider defined operations that can be used by consumers to act on the Resource. This set represents all operations defined for this described Resource type, which may be a superset of those operations a particular Consumer is actually allowed to use. The subset of allowed operations for a particular Consumer shall be those operations returned to this Consumer if querying an instance of the described Resource type. Note that this attribute is called "actions" so as not to conflict with the ResourceMetadata Resource's own operations. Each operation shall contain the following nested data: <table><tr><td>Name</td><td colspan="2">action</td></tr><tr><td>Data</td><td>Type</td><td>Description</td></tr><tr><td>name</td><td>string</td><td>The name of the operation.</td></tr></table>		Name	action		Data	Type	Description	name	string	The name of the operation.									
Name	action																				
Data	Type	Description																			
name	string	The name of the operation.																			

Name	ResourceMetadata		
Type URI	http://schemas.dmtf.org/cimi/1/ResourceMetadata		
Attribute	Type	Description	
			Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
	uri	URI	A URI that uniquely identifies the operation at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
	description	string	The human-readable description of the semantic of the operation. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write
	method	string	The protocol-dependent verb to use to perform the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
	inputMessage	string	The body mimeType of the request message; it may depend on the model format chosen by the Provider. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
	outputMessage	string	The body mimeType of the response message; it may depend on the model format chosen by the Provider. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
	Constraints: Provider: support optional; mutable Consumer: support optional; read-write		

1700 When implementing or using ResourceMetadata, Providers and Consumers shall adhere to the
1701 syntax and semantics of its attributes as described in the above table as well as in the tables describing
1702 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
1703 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
1704 Resource in both JSON and XML:

1705 **JSON media type:** application/json

1706 **JSON serialization:**

```

1707 { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadata",
1708   "id": string,
1709   "typeURI": string,
1710   "name": string,
1711   "attributes" : [
1712     { "name": string,
1713       "namespace": string, ?
1714       "type": string, ?
1715       "required": boolean, ?
1716       ...value constraints...? } *

```

```

1717 ], ?
1718 "capabilities": [
1719   { "name": string, ?
1720     "uri": string,
1721     "description": string, ?
1722     "value": any } *
1723 ], ?
1724 "actions" : [
1725   { "name": string,
1726     "uri": string,
1727     "description": string, ?
1728     "method": string,
1729     "inputMessage": string, ?
1730     "outputMessage": string ? }, *
1731 ], ?
1732 "operations": [
1733   { "rel": "edit", "href": string }, ?
1734   { "rel": "delete", "href": string } ?
1735 ] ?
1736 ...
1737 }

```

1738 **XML media type:** application/xml

1739 **XML serialization:**

```

1740 <ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/1">
1741   <id> xs:anyURI </id>
1742   <name> xs:string </name>
1743   <typeURI> xs:anyURI </typeURI>
1744   <attribute name="xs:string" namespace="xs:anyURI"? type="xs:string"
1745     required="xs:boolean"? >
1746     .. value constraints...?
1747   </attribute> *
1748   <capability name="xs:string"? uri="xs:anyURI" description="xs:string"?>
1749     xs:any*
1750   </capability> *
1751   <action name="xs:string" uri="xs:anyURI" description="xs:string"?
1752     method="xs:string" inputMessage="xs:string"?
1753     outputMessage="xs:string"? /> *
1754   <operation rel="edit" href="xs:anyURI"/> ?
1755   <operation rel="delete" href="xs:anyURI"/> ?

```

```

1756     <xs:any>*
1757 </ResourceMetadata>

```

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

5.11.1 Serialization of attribute value constraints

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

type="string"

The JSON shall be of the form:

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

The XML shall be of the form:

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

type="integer"

The JSON shall be of the form:

```

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

```

The XML shall be of the form:

```

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

```

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

type="boolean"

The JSON shall be of the form:

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

The XML shall be of the form:

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

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

5.11.1.1 Examples

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

```

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

```

```

1792     </attribute>
1793     <attribute name="Location" namespace="http://example.org/" type="string"/>
1794 </ResourceMetadata>

```

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

```

1797 <ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/1">
1798   <id> http://example.org/types/VC </id>
1799   <typeURI> http://schemas.dmtf.org/cimi/1/VolumeConfiguration </typeURI>
1800   <name> VolumeConfiguration </name>
1801   <attribute name="format" type="string" required="false">
1802     <value> ext4 </value>
1803     <value> ntfs </value>
1804   </attribute>
1805   <attribute name="Location" namespace="http://example.org/" type="string"
1806     required="true">
1807     <value> NYC </value>
1808     <value> LAX </value>
1809   </attribute>
1810 </ResourceMetadata>

```

The following example shows the same VolumeConfiguration serialized in JSON:

```

1811 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
1812   "id": "http://example.org/types/VC",
1813   "typeURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
1814   "name": "VolumeConfiguration",
1815   "attributes": [
1816     { "name": "format",
1817       "type": "string",
1818       "required": false,
1819       "values": [ "ext4", "ntfs" ]
1820     },
1821     { "name": "Location",
1822       "namespace": "http://example.org",
1823       "type": "string",
1824       "required": true,
1825       "values": [ "NYC", "LAX" ]
1826     }
1827   ]
1828 }
1829

```

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

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

5.11.2 Capabilities

Table 7 describes the capability URIs defined by this specification. Providers may define new URIs and it is recommended that these URIs be dereferencable such that Consumers can discover the details of the new capability. The "Resource Name" column contains the name of the Resource that may contain the specified capability within its ResourceMetadata. The "Capability Name" column contains the name of the specified capability and shall be unique within the scope of the corresponding Resource. Each capability's URI shall be constructed by appending the "Resource Name", a slash(/), and the "Capability Name" to "http://schemas.dmtf.org/cimi/1/capability/". For example, the Machine's "InitialState" capability shall have a URI of:

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

Capabilities that apply to the Provider in general, and are not specific to any one Resource, shall be associated with the CloudEntryPoint Resource (in case a capability applies only to the CloudEntryPoint Resource itself, its definition indicates this).

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

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

Table 7 – Capability URIs

Resource Name	Capability Name	Description
CloudEntryPoint	ExpandParameter	If true, the Provider shall support the <code>\$expand</code> query parameter.
CloudEntryPoint	FilterParameter	If true, the Provider shall support the <code>\$filter</code> query parameter.
CloudEntryPoint	FirstParameter	If true, the Provider shall support both the <code>\$first</code> and <code>\$last</code> query parameters.
CloudEntryPoint	SelectParameter	If true, the Provider shall support the <code>\$select</code> query parameter.

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

Resource Name	Capability Name	Description
		Resource can be shared by multiple Machines.
Volume	VolumeConfigByValue	If true, the Provider shall support specifying VolumeConfigurations by value in the Volume create operation. If true, the VolumeTemplateByValue capability shall have the value true.
Volume	VolumeImageByValue	If true, the Provider shall support specifying VolumeImages by value in the Volume create operation. If true, the VolumeTemplateByValue capability shall have the value true.
Volume	VolumeSnapshot	If true, the Provider shall support creating a new VolumeImage by referencing an existing Volume.
Volume	VolumeTemplateByValue	If true, the Provider shall support specifying the VolumeTemplates by value in Volume create operations.
Volume	VolumeAvailabilityLevel	If true, the Provider supports the notion of an availability level for the Volume Resource. The availability level and its value constraints are advertised as an extension attribute by the way of the Volume and VolumeTemplate ResourceMetadata.
Network	NetworkConfigByValue	If true, the Provider shall support specifying NetworkConfigurations by value in Network create operations.
Network	NetworkTemplateByValue	If true, the Provider shall support specifying Network Templates by value in Network create operations.
Network	DefaultInitialState	If this capability is set, unless otherwise provided (e.g., by a NetworkTemplate "initialState" attribute), the Provider shall set a new Network to this state value, assuming the value is compatible with the InitialStates capability, if set.
Network	InitialStates	If this capability is set, and if using a NetworkTemplate that has an "initialState" attribute, a Consumer shall use an initialState value from the set of values of this capability.
NetworkPort	NetworkPortConfigByValue	If true, the Provider shall support specifying NetworkPortConfigurations by value in NetworkPort create operations.
NetworkPort	NetworkPortTemplateByValue	If true, the Provider shall support specifying NetworkPortTemplates by value in NetworkPort create operations.
NetworkPort	DefaultInitialState	If this capability is set, unless otherwise provided (e.g., by a NetworkPortTemplate "initialState" attribute), the Provider shall set a new NetworkPort to this state value, assuming the value is compatible with the InitialStates capability, if set.
NetworkPort	InitialStates	If this capability is set, and if using a NetworkPortTemplate that has an "initialState" attribute, a Consumer shall use an initialState value from the set of values of this capability.
ForwardingGroup	MixedNetwork	If true, a Provider shall support ForwardingGroups that can have both private and public connections at the same time. Otherwise, ForwardingGroups shall have only private or public connections at the same time.
Job	JobRetention	If set, the value of this capability shall indicate the minimum number of minutes a job shall be retained by the Provider before it is deleted.
Meter	MeterConfigByValue	If true, the Provider shall support specifying MeterConfigurations by value in Meter create operations.
Meter	MeterTemplateByValue	If true, the Provider shall support specifying MeterTemplates by value in Meter create operations.
EventLog	Linked	If true, the Provider shall delete EventLogs that are associated with Resources if the Resource is deleted.

The following examples show the ResourceMetadata for a Machine that advertises some of its capabilities:

JSON serialization:

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

XML serialization:

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

5.11.3 ResourceMetadataCollection Resource

A ResourceMetadataCollection Resource represents the Collection of ResourceMetadata Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. Note that

modifications of the Resources within this Collection are typically reserved for administrator types of CIMI Consumers. This Resource shall be serialized as follows:

JSON serialization:

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

XML serialization:

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

5.12 Cloud Entry Point

The Cloud Entry Point (CloudEntryPoint Resource) represents the entry point into the cloud defined by the CIMI Model. The Cloud Entry Point implements a catalog of Resources, such as Systems, SystemTemplates, Machines, MachineTemplates, etc., that can be queried and browsed by the Consumer.

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

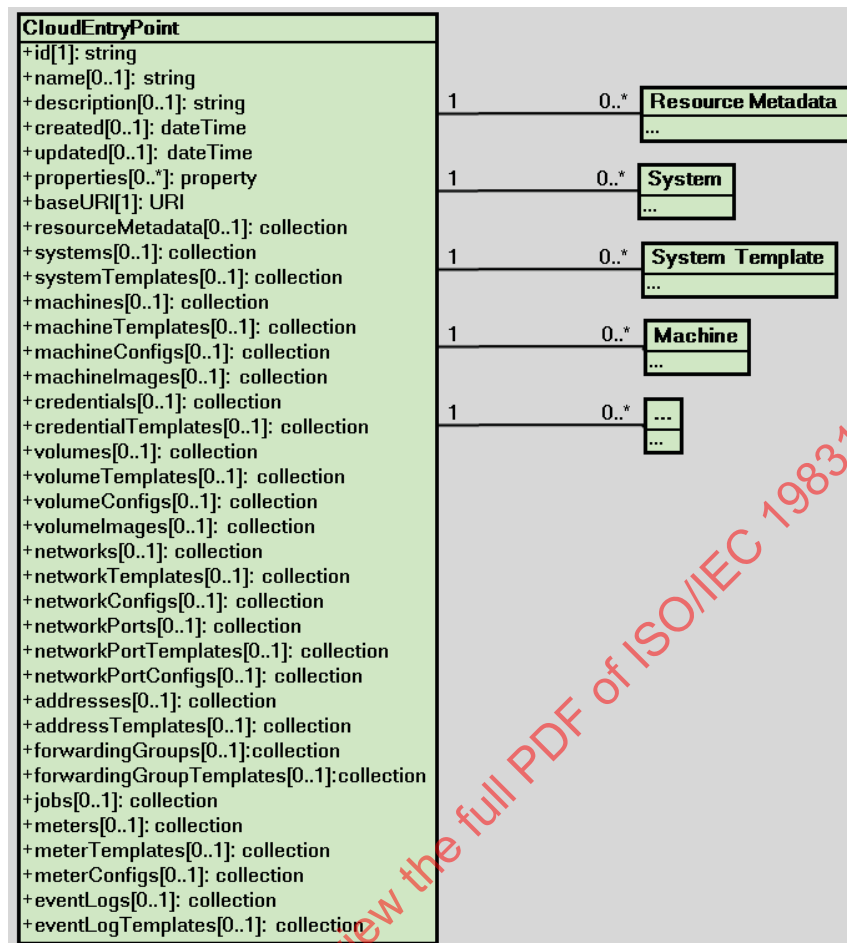


Figure 1 - Cloud Entry Point

If a Consumer issues a read on the CloudEntryPoint Resource, the Provider shall return a CloudEntryPoint Resource that only catalogs Resources on which this Consumer is allowed to perform operations. Table 8 describes the attributes for the CloudEntryPoint Resource.

Table 8 – CloudEntryPoint attributes

Name	CloudEntryPoint	
Type URI	http://www.dmf.org/cimi/CloudEntryPoint	
Attribute	Type	Description
baseURI	URI	An absolute URI that references the "base URI" of the Provider. This URI shall be used to convert relative URIs to Resources within this Provider to absolute URIs. See the "URIs" clause of 5.5. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
resourceMetadata	collection [Resource Metadata]	A reference to ResourceMetadata a Collection of this Cloud Entry Point. The Collection contains a description of the Resources supported by the Provider. If a Resource does not have any metadata, it shall not appear in this list, e.g., it has no constraints beyond what the CIMI specification defines nor does it have any extension attributes. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

Name	CloudEntryPoint	
Type URI	http://www.dmf.org/cimi/CloudEntryPoint	
Attribute	Type	Description
systems	collection [System]	A reference to the SystemCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
systemTemplates	collection [System Template]	A reference to the SystemTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
machines	collection [Machine]	A reference to the MachineCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
machineTemplates	collection [Machine Template]	A reference to the MachineTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
machineConfigs	collection [Machine Configuration]	A reference to the MachineConfigurationCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
machineImages	collection [Machine Image]	A reference to the MachineImageCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
credentials	collection [Credential]	A reference to the CredentialCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
credentialTemplates	collection [Credential Template]	A reference to the CredentialTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumes	collection [Volume]	A reference to the VolumeCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumeTemplates	collection [Volume Template]	A reference to the VolumeTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumeConfigs	collection [Volume Configuration]	A reference to the VolumeConfigurationCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumeImages	collection [Volume Image]	A reference to the VolumeImageCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

Name	CloudEntryPoint	
Type URI	http://www.dmf.org/cimi/CloudEntryPoint	
Attribute	Type	Description
networks	collection [Network]	A reference to the NetworkCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkTemplates	collection [Network Template]	A reference to the NetworkTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkConfigs	collection [Network Configuration]	A reference to the NetworkConfigurationCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkPorts	collection [NetworkPort]	A reference to the NetworkPortCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkPortTemplates	collection [NetworkPort Template]	A reference to the NetworkPortTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkPortConfigs	collection [NetworkPort Configuration]	A reference to the NetworkPortConfigurationCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
addresses	collection [Address]	A reference to the AddressCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
addressTemplates	collection [Address Template]	A reference to the AddressTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
forwardingGroups	collection [Forwarding Group]	A reference to the ForwardingGroupCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
forwardingGroupTemplates	collection [Forwarding Group Template]	A reference to the ForwardingGroupTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
jobs	collection [Job]	A reference to the JobsCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meters	collection [Meter]	A reference to the MeterCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

Name	CloudEntryPoint	
Type URI	http://www.dmf.org/cimi/CloudEntryPoint	
Attribute	Type	Description
meterTemplates	collection [Meter Template]	A reference to the MeterTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meterConfigs	collection [Meter Configuration]	A reference to the MeterConfigurationCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLogs	collection [EventLog]	A reference to the EventLogCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLogTemplates	collection [EventLog Template]	A reference to the EventLogTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

1947 Each of the Collections mentioned in Table 8 are defined within the related Resource definition clauses.
 1948 For example, the MachineCollection Resource is defined in clause 5.14.2 as part of the
 1949 Machine-related Resources.

1950 When implementing or using CloudEntryPoint, Providers and Consumers shall adhere to the syntax
 1951 and semantics of its attributes as described in the above table as well as in the tables describing
 1952 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
 1953 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 1954 Resource in both JSON and XML:

1955 **JSON media type:** application/json

1956 **JSON serialization:**

```

1957 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
1958   "id": string,
1959   "name": string, ?
1960   "description": string, ?
1961   "created": string, ?
1962   "updated": string, ?
1963   "properties": { string: string, + }, ?
1964   "baseURI": string,
1965   "resourceMetadata": { "href": string }, ?
1966   "systems": { "href": string }, ?
1967   "systemTemplates": { "href": string }, ?
1968   "machines": { "href": string }, ?
1969   "machineTemplates": { "href": string }, ?
1970   "machineConfigs": { "href": string }, ?
1971   "machineImages": { "href": string }, ?

```

ISO/IEC 19831:2015 (E)

```

1972 "credentials": { "href" string }, ?
1973 "credentialTemplates": { "href" string }, ?
1974 "volumes": { "href": string }, ?
1975 "volumeTemplates": { "href": string }, ?
1976 "volumeConfigs": { "href": string }, ?
1977 "volumeImages": { "href": string }, ?
1978 "networks": { "href": string }, ?
1979 "networkTemplates": { "href": string }, ?
1980 "networkConfigs": { "href": string }, ?
1981 "networkPorts": { "href": string }, ?
1982 "networkPortTemplates": { "href": string }, ?
1983 "networkPortConfigs": { "href": string }, ?
1984 "addresses": { "href": string }, ?
1985 "addressTemplates": { "href": string }, ?
1986 "forwardingGroups" { "href": string }, ?
1987 "forwardingGroupTemplates" { "href": string }, ?
1988 "jobs": { "href": string }, ?
1989 "meters": { "href": string }, ?
1990 "meterTemplates": { "href": string }, ?
1991 "meterConfigs": { "href": string }, ?
1992 "eventLogs": { "href": string }, ?
1993 "eventLogTemplates": { "href": string }, ?
1994 "operations": [
1995     { "rel": "edit", "href": string } ?
1996 ] ?
1997 ...
1998 }

```

1999 **XML media type:** application/xml

2000 **XML serialization:**

```

2001 <CloudEntryPoint xmlns="http://schemas.dmtf.org/cimi/1">
2002   <id> xs:anyURI </id>
2003   <name> xs:string </name> ?
2004   <description> xs:string </description> ?
2005   <created> xs:dateTime </created> ?
2006   <updated> xs:dateTime </updated> ?
2007   <property key="xs:string"> xs:string </property> *
2008   <baseURI> xs:anyURI </baseURI>
2009   <resourceMetadata href="xs:anyURI"/> ?
2010   <systems href="xs:anyURI"/> ?

```

```

2011 <systemTemplates href="xs:anyURI"/> ?
2012 <machines href="xs:anyURI"/> ?
2013 <machineTemplates href="xs:anyURI"/> ?
2014 <machineConfigs href="xs:anyURI"/> ?
2015 <machineImages href="xs:anyURI"/> ?
2016 <credentials href="xs:anyURI"/> ?
2017 <credentialTemplates href="xs:anyURI"/> ?
2018 <volumes href="xs:anyURI"/> ?
2019 <volumeTemplates href="xs:anyURI"/> ?
2020 <volumeConfigs href="xs:anyURI"/> ?
2021 <volumeImages href="xs:anyURI"/> ?
2022 <networks href="xs:anyURI"/> ?
2023 <networkTemplates href="xs:anyURI"/> ?
2024 <networkConfigs href="xs:anyURI"/> ?
2025 <networkPorts href="xs:anyURI"/> ?
2026 <networkPortTemplates href="xs:anyURI"/> ?
2027 <networkPortConfigs href="xs:anyURI"/> ?
2028 <addresses href="xs:anyURI"/> ?
2029 <addressTemplates href="xs:anyURI"/> ?
2030 <forwardingGroups href="xs:anyURI"/> ?
2031 <forwardingGroupTemplates href="xs:anyURI"/> ?
2032 <jobs href="xs:anyURI"/> ?
2033 <meters href="xs:anyURI"/> ?
2034 <meterTemplates href="xs:anyURI"/> ?
2035 <meterConfigs href="xs:anyURI"/> ?
2036 <eventLogs href="xs:anyURI"/> ?
2037 <eventLogTemplates href="xs:anyURI"/> ?
2038 <operation rel="edit" href="xs:anyURI"/> ?
2039 <xs:any>*
2040 </CloudEntryPoint>

```

5.12.1 Operations

This Resource supports the Read and Update operations.

5.13 System Resources and relationships

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

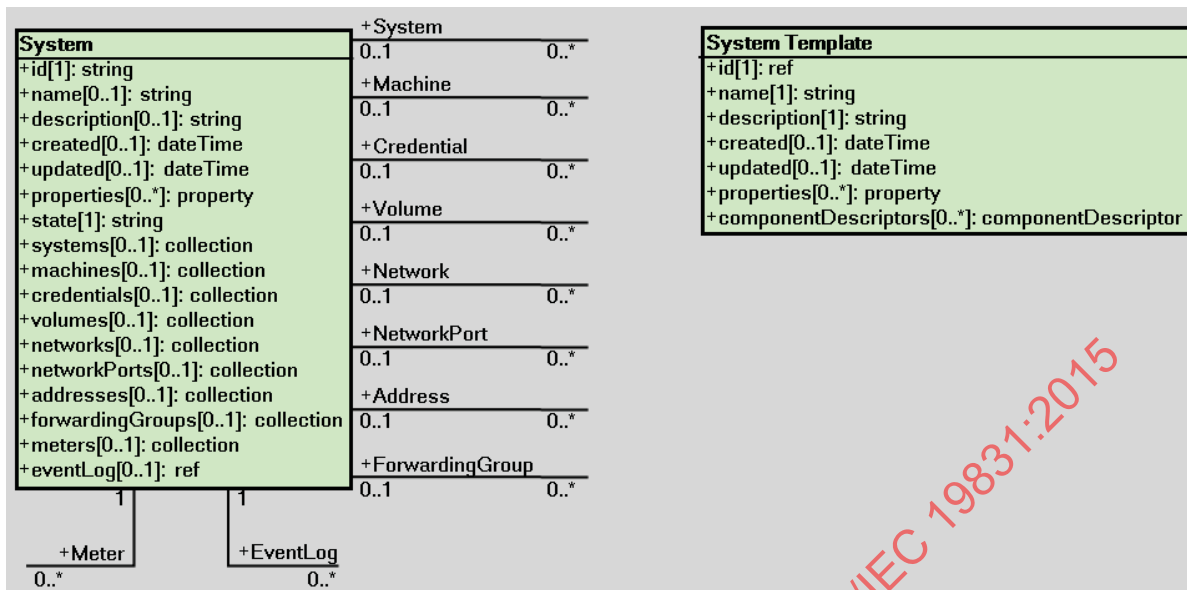


Figure 2 - System Resources

5.13.1 System

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

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

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

- By default, all Resources that are created as the result of a System creation are also owned by the System. (This rule can be overridden by subsequent modifications to the top-level System Collection attributes.)
- Ownership of a Resource by a System is expressed by including the reference to the Resource in the appropriate top-level System Collection attribute, or by ownership to a sub-System of this System (i.e., ownership is transitive across hierarchies of Systems).
- If a Resource other than a System is added to an existing System (i.e., becomes owned by the System by insertion of its reference to the appropriate top-level System Collection attribute), other Resources already referred by this added Resource are by default not owned by the System. (This rule can be overridden by subsequent modifications to the top-level System Collection attributes.)

2073 A Resource shall not be owned by more than one System at any point in time (unless there is an
 2074 ownership relationship between these Systems). Note that a Resource does not need to be owned by a
 2075 System (i.e., part of any of its Collection attributes) to be referenced/used by a Resource in the
 2076 System. By not including it in any of the Collections, the Resource is simply not part of any actions
 2077 performed on the System. Table 9 describes the System attributes.

2078 **Table 9 – System attributes**

Name		System
Type URI		http://schemas.dmtf.org/cimi/1/System
Attribute	Type	Description
state	string	The operational state of the System. Allowable values include: (See 5.14.1.) CREATING: The System is in the process of being created. STARTING/STARTED/STOPPING/STOPPED/PAUSING/PAUSED/SUSPENDING/SUSPENDED: The System shall be in one of these states if all the Machines referenced by the System are in that state. See clause 5.14.1 for the list of available actions based on the state of a Machine. Such transitional states may just indicate that all Machines in a System are undergoing the same operation (e.g., "start"), without the System being actually operated on (e.g., no "start" done at System level). An actual operation on a System may be traced by querying the "job" entity. MIXED: The System shall be in this state if either no Machines are referenced by this System or Machines referenced by this System are in varying states. Such varying states are likely to occur when an operation is in progress on a System, resulting in transitions of its Machine states toward a new common state (e.g., STOPPED, STARTED) but at a different pace, or sequentially one after the other. DELETING: The System is in the process of being deleted. ERROR: The Provider has detected an error in the System. The operations that result in transitions to the above defined states are defined in clause 5.13.1.2. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
systems	collection [System System]	A reference to the list of references to nested Systems owned by this System. Adding an item (of type System) to this list is logically equivalent to associating the referenced System to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the referenced System from this System. Note: The SystemSystem Resource type represents an association between the System and another System. It is defined in clause 5.13.1.1.1. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
machines	collection [System Machine]	A reference to the list of references to Machines owned by this System. Adding an item (of type Machine) to this list is logically equivalent to associating the Machine to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the Machine from this System. Note: The SystemMachine Resource type represents an association between the System and a Machine. It is defined in clause 5.13.1.1.2. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
credentials	collection [System Credential]	A reference to the list of references to Credentials owned by this System. Adding an item (of type Credential) to this list is logically equivalent to associating the Credential to this System with a "containment relationship."

Name	System	
Type URI	http://schemas.dmtf.org/cimi/1/System	
Attribute	Type	Description
		Removing an item from this list is logically equivalent to de-associating the Credential from this System. Note: The SystemCredential Resource type represents an association between the System and a Credential. It is defined in clause 5.13.1.1.3. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumes	collection [System Volume]	A reference to the list of references Volumes owned by this System. Adding an item (of type Volume) to this list is logically equivalent to associating the Volume to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the Volume from this System. Note: The SystemVolume Resource type represents an association between the System and a Volume. It is defined in clause 5.13.1.1.4. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networks	collection [System Network]	A reference to the list of references Networks owned by this System. Adding an item (of type Network) to this list is logically equivalent to associating the Network to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the Network from this System. Note: The SystemNetwork Resource type represents an association between the System and a Network. It is defined in clause 5.13.1.1.5. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkPorts	collection [System NetworkPort]	A reference to the list of references NetworkPorts owned by this System. Adding an item (of type NetworkPort) to this list is logically equivalent to associating the NetworkPort to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the NetworkPort from this System. Note: The SystemNetworkPort Resource type represents an association between the System and a NetworkPort. It is defined in clause 5.13.1.1.6. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
addresses	collection [System Address]	A reference to the list of references Addresses owned by this System. Adding an item (of type Address) to this list is logically equivalent to associating the Address to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the Address from this System. Note: The SystemAddress Resource type represents an association between the System and a Address. It is defined in clause 5.13.1.1.7. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
forwardingGroups	collection [System Forwarding Group]	A reference to the list of references ForwardingGroups owned by this System. Adding an item (of type ForwardingGroup) to this list is logically equivalent to associating the ForwardingGroup to this System with a "containment relationship." Removing an item from this list is logically equivalent to de-associating the ForwardingGroup from this System. Note: The SystemForwardingGroup Resource type represents an association between the System and a ForwardingGroup. It is defined in clause 5.13.1.1.8. Constraints:

Name	System	
Type URI	http://schemas.dmtf.org/cimi/1/System	
Attribute	Type	Description
		Provider: support optional; mutable Consumer: support optional; read-only
meters	<i>collection</i> [Meter]	A reference to the list of Meters monitored for this System. Note that these Meters are for the System and not for any individual component in the System. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLog	<i>ref</i>	A reference to the EventLog of this System. Note that this EventLog is for the System and not for any individual component in the System. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

2079

2080 When implementing or using System, Providers and Consumers shall adhere to the syntax and
 2081 semantics of its attributes as described in the above table as well as in the tables describing embedded
 2082 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 2083 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 2084 both JSON and XML.

2085 **JSON media type:** application/json

2086 **JSON serialization:**

```

2087 { "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
2088   "id": string,
2089   "name": string, ?
2090   "description": string, ?
2091   "created": string, ?
2092   "updated": string, ?
2093   "properties": { string: string, + }, ?
2094   "state": string,
2095   "systems": { "href": string }, ?
2096   "machines": { "href": string }, ?
2097   "credentials": { "href": string }, ?
2098   "volumes": { "href": string }, ?
2099   "networks": { "href": string }, ?
2100   "networkPorts": { "href": string }, ?
2101   "addresses": { "href": string }, ?
2102   "forwardingGroups": { "href": string }, ?
2103   "meters": { "href": string }, ?
2104   "eventLog": { "href": string }, ?
2105   "operations": [
2106     { "rel": "edit", "href": string }, ?
  
```

```

2107     { "rel": "delete", "href": string }, ?
2108     { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
2109     { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string }, ?
2110     { "rel": "http://schemas.dmtf.org/cimi/1/action/restart", "href": string },
2111 ?
2112     { "rel": "http://schemas.dmtf.org/cimi/1/action/pause", "href": string }, ?
2113     { "rel": "http://schemas.dmtf.org/cimi/1/action/suspend", "href": string },
2114 ?
2115     { "rel": "http://schemas.dmtf.org/cimi/1/action/export", "href": string } ?
2116 ] ?
2117 ...
2118 }

```

2119 **XML media type:** application/xml

2120 **XML serialization:**

```

2121 <System xmlns="http://schemas.dmtf.org/cimi/1">
2122   <id> xs:anyURI </id>
2123   <name> xs:string </name> ?
2124   <description> xs:string </description> ?
2125   <created> xs:dateTime </created> ?
2126   <updated> xs:dateTime </updated> ?
2127   <property key="xs:string"> xs:string </property> *
2128   <state> xs:string </state>
2129   <systems href="xs:anyURI"/> ?
2130   <machines href="xs:anyURI"/> ?
2131   <credentials href="xs:anyURI"/> ?
2132   <volumes href="xs:anyURI"/> ?
2133   <networks href="xs:anyURI"/> ?
2134   <networkPorts href="xs:anyURI"/> ?
2135   <addresses href="xs:anyURI"/> ?
2136   <forwardingGroups href="xs:anyURI"/> ?
2137   <meters href="xs:anyURI"/> ?
2138   <eventLog href="xs:anyURI"/> ?
2139   <operation rel="edit" href="xs:anyURI"/> ?
2140   <operation rel="delete" href="xs:anyURI"/> ?
2141   <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
2142     href="xs:anyURI"/> ?
2143   <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
2144     href="xs:anyURI"/> ?
2145   <operation rel="http://schemas.dmtf.org/cimi/1/action/restart"
2146     href="xs:anyURI"/> ?

```

```

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

```

5.13.1.1 Collections

The following clause describes the Collection Resources owned by Systems.

5.13.1.1.1 SystemSystem Collection

The Resource type for each item of this Collection is "SystemSystem", defined in Table 10:

Table 10 – SystemSystem attributes

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

JSON serialization:

```

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

```

```

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

```

XML serialization:

```

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

```

5.13.1.1.2 SystemMachine Collection

The Resource type for each item of this Collection is "SystemMachine", defined in Table 11:

Table 11 – SystemMachine attributes

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

JSON serialization:

```

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

```

```

2213     "id": string,
2214     "name": string, ?
2215     "description": string, ?
2216     "created": string, ?
2217     "updated": string, ?
2218     "properties": { string: string, + }, ?
2219     "machine": { "href": string },
2220     "operations": [
2221         { "rel": "edit", "href": string }, ?
2222         { "rel": "delete", "href": string } ?
2223     ] ?
2224     ...
2225 }, +
2226 ], ?
2227 "operations": [ { "rel": "add", "href": string } ? ]
2228 ...
2229 }

```

XML serialization:

```

2231 <Collection
2232     resourceURI="http://schemas.dmtf.org/cimi/1/SystemMachineCollection"
2233     xmlns="http://schemas.dmtf.org/cimi/1">
2234     <id> xs:anyURI </id>
2235     <count> xs:integer </count>
2236     <SystemMachine>
2237         <id> xs:anyURI </id>
2238         <name> xs:string </name> ?
2239         <description> xs:string </description> ?
2240         <created> xs:dateTime </created> ?
2241         <updated> xs:dateTime </updated> ?
2242         <property key="xs:string"> xs:string </property> *
2243         <machine href="xs:anyURI"/>
2244         <operation rel="edit" href="xs:anyURI"/> ?
2245         <operation rel="delete" href="xs:anyURI"/> ?
2246         <xs:any>*
2247     </SystemMachine> *
2248     <operation rel="add" href="xs:anyURI"/> ?
2249     <xs:any>*
2250 </Collection>

```

2251 **5.13.1.1.3 SystemCredential Collection**

2252 The Resource type for each item of this Collection is "SystemCredential", defined in Table 12:

2253 **Table 12 – SystemCredential attributes**

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

2254 **JSON serialization:**

```

2255 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCredentialCollection",
2256   "id": string,
2257   "count": number,
2258   "systemCredentials": [
2259     { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCredential",
2260       "id": string,
2261       "name": string, ?
2262       "description": string, ?
2263       "created": string, ?
2264       "updated": string, ?
2265       "properties": { string: string, + }, ?
2266       "credential": { "href": string },
2267       "operations": [
2268         { "rel": "edit", "href": string }, ?
2269         { "rel": "delete", "href": string } ?
2270       ] ?
2271       ...
2272     }, +
2273   ], ?
2274   "operations": [ { "rel": "add", "href": string } ? ]
2275   ...
2276 }

```

2277 **XML serialization:**

```

2278 <Collection
2279   resourceURI="http://schemas.dmtf.org/cimi/1/SystemCredentialCollection"
2280   xmlns="http://schemas.dmtf.org/cimi/1">
2281   <id> xs:anyURI </id>
2282   <count> xs:integer </count>
2283   <SystemCredential>

```

```

2284     <id> xs:anyURI </id>
2285     <name> xs:string </name> ?
2286     <description> xs:string </description> ?
2287     <created> xs:dateTime </created> ?
2288     <updated> xs:dateTime </updated> ?
2289     <property key="xs:string"> xs:string </property> *
2290     <credential href="xs:anyURI"/>
2291     <operation rel="edit" href="xs:anyURI"/> ?
2292     <operation rel="delete" href="xs:anyURI"/> ?
2293     <xs:any>*
2294 </SystemCredential> *
2295     <operation rel="add" href="xs:anyURI"/> ?
2296     <xs:any>*
2297 </Collection>

```

5.13.1.1.4 SystemVolume Collection

The Resource type for each item of this Collection is "SystemVolume", defined in Table 13:

Table 13 – SystemVolume attributes

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

JSON serialization:

```

2301 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemVolumeCollection",
2302   "id": string,
2303   "count": number,
2304   "systemVolumes": [
2305     { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemVolume",
2306       "id": string,
2307       "name": string, ?
2308       "description": string, ?
2309       "created": string, ?
2310       "updated": string, ?
2311       "properties": { string: string, + }, ?
2312       "volume": { "href": string },
2313       "operations": [
2314         { "rel": "edit", "href": string }, ?
2315         { "rel": "delete", "href": string } ?

```

```

2317     ] ?
2318     ...
2319     }, +
2320   ], ?
2321   "operations": [ { "rel": "add", "href": string } ? ]
2322   ...
2323 }

```

2324 XML serialization:

```

2325 <Collection
2326     resourceURI="http://schemas.dmtf.org/cimi/1/SystemVolumeCollection"
2327     xmlns="http://schemas.dmtf.org/cimi/1">
2328   <id> xs:anyURI </id>
2329   <count> xs:integer </count>
2330   <SystemVolume>
2331     <id> xs:anyURI </id>
2332     <name> xs:string </name> ?
2333     <description> xs:string </description> ?
2334     <created> xs:dateTime </created> ?
2335     <updated> xs:dateTime </updated> ?
2336     <property key="xs:string"> xs:string </property> *
2337     <volume href="xs:anyURI"/>
2338     <operation rel="edit" href="xs:anyURI"/> ?
2339     <operation rel="delete" href="xs:anyURI"/> ?
2340     <xs:any>*
2341   </SystemVolume> *
2342   <operation rel="add" href="xs:anyURI"/> ?
2343   <xs:any>*
2344 </Collection>

```

2345 5.13.1.1.5 SystemNetwork Collection

2346 The Resource type for each item of this Collection is "SystemNetwork", defined in Table 14:

2347 **Table 14 – SystemNetwork attributes**

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

2348 JSON serialization:

```

2349 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetworkCollection",

```

```

2350     "id": string,
2351     "count": number,
2352     "systemNetworks": [
2353         { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetwork",
2354           "id": string,
2355           "name": string, ?
2356           "description": string, ?
2357           "created": string, ?
2358           "updated": string, ?
2359           "properties": { string: string, + }, ?
2360           "network": { "href": string },
2361           "operations": [
2362             { "rel": "edit", "href": string }, ?
2363             { "rel": "delete", "href": string } ?
2364           ] ?
2365           ...
2366         }, +
2367     ], ?
2368     "operations": [ { "rel": "add", "href": string } ? ]
2369     ...
2370 }

```

XML serialization:

```

2371 <Collection
2372     resourceURI="http://schemas.dmtf.org/cimi/1/SystemNetworkCollection"
2373     xmlns="http://schemas.dmtf.org/cimi/1">
2374     <id> xs:anyURI </id>
2375     <count> xs:integer </count>
2376     <SystemNetwork>
2377         <id> xs:anyURI </id>
2378         <name> xs:string </name> ?
2379         <description> xs:string </description> ?
2380         <created> xs:dateTime </created> ?
2381         <updated> xs:dateTime </updated> ?
2382         <property key="xs:string"> xs:string </property> *
2383         <network href="xs:anyURI"/>
2384         <operation rel="edit" href="xs:anyURI"/> ?
2385         <operation rel="delete" href="xs:anyURI"/> ?
2386         <xs:any>*
2387     </SystemNetwork> *
2388     <operation rel="add" href="xs:anyURI"/> ?
2389

```

2390 <xs:any>*

2391 </Collection>

2392 5.13.1.1.6 SystemNetworkPort Collection

2393 The Resource type for each item of this Collection is "SystemNetworkPort", defined in Table 15:

2394 **Table 15 – SystemNetworkPort attributes**

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

2395 JSON serialization:

```

2396       { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetworkPortCollection",
2397         "id": string,
2398         "count": number,
2399         "systemNetworkPorts": [
2400           { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetworkPort",
2401             "id": string,
2402             "name": string, ?
2403             "description": string, ?
2404             "created": string, ?
2405             "updated": string, ?
2406             "properties": { string: string, + }, ?
2407             "networkPort": { "href": string },
2408             "operations": [
2409               { "rel": "edit", "href": string }, ?
2410               { "rel": "delete", "href": string } ?
2411             ] ?
2412           ...
2413           }, +
2414         ], ?
2415         "operations": [ { "rel": "add", "href": string } ? ]
2416         ...
2417       }
```

2418 XML serialization:

```

2419       <Collection
2420         resourceURI="http://schemas.dmtf.org/cimi/1/SystemNetworkPortCollection"
2421         xmlns="http://schemas.dmtf.org/cimi/1">
2422       <id> xs:anyURI </id>
```

```

2423 <count> xs:integer </count>
2424 <SystemNetworkPort>
2425   <id> xs:anyURI </id>
2426   <name> xs:string </name> ?
2427   <description> xs:string </description> ?
2428   <created> xs:dateTime </created> ?
2429   <updated> xs:dateTime </updated> ?
2430   <property key="xs:string"> xs:string </property> *
2431   <networkPort href="xs:anyURI"/>
2432   <operation rel="edit" href="xs:anyURI"/> ?
2433   <operation rel="delete" href="xs:anyURI"/> ?
2434   <xs:any>*
2435 </SystemNetworkPort> *
2436 <operation rel="add" href="xs:anyURI"/> ?
2437 <xs:any>*
2438 </Collection>

```

5.13.1.1.7 SystemAddress Collection

The Resource type for each item of this Collection is "SystemAddress", defined in Table 16:

Table 16 – SystemAddress attributes

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

JSON serialization:

```

2442 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemAddressCollection",
2443   "id": string,
2444   "count": number,
2445   "systemAddresses": [
2446     { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemAddress",
2447       "id": string,
2448       "name": string, ?
2449       "description": string, ?
2450       "created": string, ?
2451       "updated": string, ?
2452       "properties": { string: string, + }, ?
2453       "address": { "href": string },
2454       "operations": [

```

```

2456     { "rel": "edit", "href": string }, ?
2457     { "rel": "delete", "href": string } ?
2458   ] ?
2459   ...
2460 }, +
2461 ], ?
2462 "operations": [ { "rel": "add", "href": string } ? ]
2463 ...
2464 }

```

XML serialization:

```

2465 <Collection
2466   resourceURI="http://schemas.dmtf.org/cimi/1/SystemAddressCollection"
2467   xmlns="http://schemas.dmtf.org/cimi/1">
2468   <id> xs:anyURI </id>
2469   <count> xs:integer </count>
2470   <SystemAddress>
2471     <id> xs:anyURI </id>
2472     <name> xs:string </name> ?
2473     <description> xs:string </description> ?
2474     <created> xs:dateTime </created> ?
2475     <updated> xs:dateTime </updated> ?
2476     <property key="xs:string"> xs:string </property> *
2477     <address href="xs:anyURI"/>
2478     <operation rel="edit" href="xs:anyURI"/> ?
2479     <operation rel="delete" href="xs:anyURI"/> ?
2480     <xs:any>*
2481   </SystemAddress> *
2482   <operation rel="add" href="xs:anyURI"/> ?
2483   <xs:any>*
2484 </Collection>

```

5.13.1.1.8 SystemForwardingGroup Collection

The Resource type for each item of this Collection is "SystemForwardingGroup", defined in Table 17:

Table 17 – SystemForwardingGroup attributes

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

JSON serialization:

```

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

```

XML serialization:

```

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

```

```

2530     <xs:any>*
2531   </SystemForwardingGroup> *
2532   <operation rel="add" href="xs:anyURI"/> ?
2533   <xs:any>*
2534 </Collection>

```

5.13.1.1.9 SystemMeter Collection

The Resource type for each item of this Collection is "Meter" as defined in clause 5.17.3.

JSON serialization:

```

2538 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemMeterCollection",
2539   "id": string,
2540   "count": number,
2541   "meters": [
2542     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
2543       "id": string,
2544       ... remaining Meter attributes ...
2545     }, +
2546   ], ?
2547   "operations": [ { "rel": "add", "href": string } ? ]
2548   ...
2549 }

```

XML serialization:

```

2551 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/SystemMeterCollection"
2552   xmlns="http://schemas.dmtf.org/cimi/1">
2553   <id> xs:anyURI </id>
2554   <count> xs:integer </count>
2555   <Meter>
2556     <id> xs:anyURI </id>
2557     ... remaining Meter attributes ...
2558   </Meter>*
2559   <operation rel="add" href="xs:anyURI"/> ?
2560   <xs:any>*
2561 </Collection>

```

5.13.1.2 Operations

The System Resource supports the Read, Update, and Delete operations. Create is supported through the SystemCollection Resource.

The following custom operations are also defined:

start/stop/restart/pause/suspend

2567 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/xxx

2568 Where "xxx" is either "start", "stop", "restart", "pause", or "suspend".

2569 This operation shall recursively perform the requested operation on each component of the System
 2570 (Machine or sub-System). Note that not all Machines need to be in the same state for this operation
 2571 to be available and the impact that this operation varies depending on the component's current state; see
 2572 clause 5.14.1.2 for more details about performing operations on Machines. If the operation fails for a
 2573 Machine, that Machine shall not be affected by the operation.

2574 **export**

2575 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/export

2576 This operation shall export a System. If an export package exists at that URI, it is updated with the
 2577 values of the System and any component management Resources. Otherwise, a new export package is
 2578 created at that URI with a Media Type as specified by the "format" parameter. Other formats may be used
 2579 if supported, but are not specified by this standard.

2580 Input parameters:

- 2581 1) "format" - type: string - optional
 2582 Indicates the Media Type of the exported data. If not present, the default value shall be
 2583 "application/ovf."
 2584
- 2585 2) "destination" - type: URI - optional
 2586 Indicates the location to where the exported data is placed. If not present, the HTTP response
 2587 Location header shall contain the URL to the exported data. Based on the specific protocol
 2588 specified within the URI, the Consumer might need to provide additional information (such as
 2589 credentials) in the "properties" field. In the case of HTTP, a PUT shall be used to place the data
 2590 at the specified location.

2591 Output parameters: None.

2592 **HTTP protocol**

2593 To export a System, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/export" URI of the
 2594 System where the HTTP request body shall be as described below.

2595 **JSON media type:** application/json

2596 **JSON serialization:**

```
2597 { "action": "http://schemas.dmtf.org/cimi/1/action/export",
2598   "format": string, ?
2599   "destination": string, ?
2600   "properties": { string: string, + } ?
2601   ...
2602 }
```

2603 **XML media type:** application/xml

2604 **XML serialization**

```
2605 <Action xmlns="http://schemas.dmtf.org/cimi/1">
2606   <action> http://schemas.dmtf.org/cimi/1/action/export </action>
2607   <format> xs:string </format> ?
```

```

2608     <destination> xs:anyURI </destination> ?
2609     <property key="xs:string"> xs:string </property> *
2610     <xs:any>*
2611 </Action>

```

5.13.2 SystemCollection Resource

A SystemCollection Resource represents the Collection of System Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

```

2616 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCollection",
2617   "id": string,
2618   "count", number,
2619   "systems": [
2620     { "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
2621       "id": string,
2622       ... remaining System attributes ...
2623     }, +
2624   ], ?
2625   "operations": [
2626     { "rel": "add", "href": string }, ?
2627     { "rel": "http://schemas.dmtf.org/cimi/1/action/import", "href": string } ?
2628   ]
2629   ...
2630 }

```

XML serialization:

```

2632 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/SystemCollection"
2633   xmlns="http://schemas.dmtf.org/cimi/1">
2634   <id> xs:anyURI </id>
2635   <count> xs:integer </count>
2636   <System>
2637     <id> xs:anyURI </id>
2638     ... remaining System attributes ...
2639   </System> *
2640   <operation rel="add" href="xs:anyURI"/> ?
2641   <operation rel="http://schemas.dmtf.org/cimi/1/action/import"
2642 href="xs:anyURI"/> ?
2643   <xs:any>*
2644 </Collection>

```

5.13.2.1 Operations

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

Resources created during the process of creating a System shall be "owned" by the System (see 5.13.1). For example, a componentDescriptor that references a MachineTemplate, and within that MachineTemplate is a reference to a VolumeTemplate, results in a reference to the new Machine being added to the System.machines attribute and a reference to the new Volume being added to the System.volumes attribute. However, if this MachineTemplate refers to an existing Volume, this Volume shall not be added to the top-level System attributes.

The following custom operations are also defined:

import

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

This operation shall import a System. Not only is a System created, but Machines, Volumes, and Networks and possibly recursive Systems and their components may also be created corresponding to imported descriptor entries. More detail about this process is in ANNEX A.

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

Output parameters: None.

HTTP protocol

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

JSON media type: application/json

JSON serialization:

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

XML media type: application/xml

XML serialization

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

5.13.3 SystemTemplate Resource

The SystemTemplate Resource contains the set of individual descriptors that are necessary to create the components of a System. Each component descriptor can be considered to be the persisted view of the create operation that instantiates the component. In practice, the Provider interprets the set of

2686 component descriptors as a set of creation operations to be executed in an order compatible with the
 2687 dependencies (e.g., attachments or references between components) that are expressed between these
 2688 components.

2689 A SystemTemplate may include component references in the descriptors, used to express links
 2690 between components of the resulting System. A component reference uses the "name" of the target
 2691 (referred) component. For example, `<volume href="#newVolume"/>` would reference a Volume
 2692 named "newVolume." The reference name `#newVolume` is replaced by the actual Resource URL in
 2693 the instantiated System.

2694 A SystemTemplate shall not contain two component descriptors of the same type that would result in
 2695 the same non-null value for the "name" attribute of resulting components. Attempting to create or to
 2696 update a SystemTemplate that fails this rule shall result in an error.

2697 Table 18 describes the SystemTemplate attributes.

2698 **Table 18 – SystemTemplate attributes**

Name	SystemTemplate																						
Type URI	http://schemas.dmtf.org/cimi/1/SystemTemplate																						
Attribute	Type	Description																					
component Descriptors	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 by value), and may also provide additional metadata (name, description, properties). The creation order of components is not specified in SystemTemplate; in particular the order of the component descriptors in this array is not meaningful in terms of creation order.																					
		<table><tr><td>Name</td><td colspan="2">componentDescriptor</td></tr><tr><td>Data</td><td>Type</td><td>Description</td></tr><tr><td>name</td><td>string</td><td>The value of the "name" attribute that is 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 is instantiated. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</td></tr><tr><td>description</td><td>string</td><td>The value of the "description" attribute that is associated with a System component created from this component descriptor. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</td></tr><tr><td>properties</td><td>map</td><td>The key/value pairs that is associated with a System component created from this component descriptor. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</td></tr><tr><td>type</td><td>URI</td><td>The TypeURI of the component to be created from this component descriptor, e.g., for a Machine: http://schemas.dmtf.org/cimi/1/Machine Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td></tr><tr><td>component Template</td><td>any</td><td>Reference either to a component Template or to the Template data itself inlined (i.e., the Template "value"). Note that the exact name of this attribute varies depending on the type of Resource being created, e.g.,</td></tr></table>	Name	componentDescriptor		Data	Type	Description	name	string	The value of the "name" attribute that is 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 is instantiated. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write	description	string	The value of the "description" attribute that is associated with a System component created from this component descriptor. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write	properties	map	The key/value pairs that is associated with a System component created from this component descriptor. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write	type	URI	The TypeURI of the component to be created from this component descriptor, e.g., for a Machine: http://schemas.dmtf.org/cimi/1/Machine Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	component Template	any	Reference either to a component Template or to the Template data itself inlined (i.e., the Template "value"). Note that the exact name of this attribute varies depending on the type of Resource being created, e.g.,
		Name	componentDescriptor																				
		Data	Type	Description																			
		name	string	The value of the "name" attribute that is 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 is instantiated. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write																			
		description	string	The value of the "description" attribute that is associated with a System component created from this component descriptor. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write																			
		properties	map	The key/value pairs that is associated with a System component created from this component descriptor. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write																			
type	URI	The TypeURI of the component to be created from this component descriptor, e.g., for a Machine: http://schemas.dmtf.org/cimi/1/Machine Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																					
component Template	any	Reference either to a component Template or to the Template data itself inlined (i.e., the Template "value"). Note that the exact name of this attribute varies depending on the type of Resource being created, e.g.,																					

Name	SystemTemplate		
Type URI	http://schemas.dmtf.org/cimi/1/SystemTemplate		
Attribute	Type	Description	
		MachineTemplate for a Machine. This attribute shall contain either: • A Template that is provided inline. Such an embedded Template may contain component references, each one of which shall resolve to the URI of a component with same name once created from this SystemTemplate. • A reference to an externally defined Template. Some attribute name/value pairs may be added inside the componentTemplate element to override similar attributes in the referred Template (as described in 4.2.1.1). This example shows how component references can be added to an external Template. <i>Example (JSON):</i> <pre>"machineTemplate": { "href": "http://example.com/machineTemplates/72000", "credential": { "href": "#MyCredential" } }</pre> <i>This "credential" attribute assumes that there is another componentDescriptor item named "MyCredential" of type "Credential" in the SystemTemplate. It shall set or override similar attribute in the referred MachineTemplate if instantiating the Machine component.</i> Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	
	quantity	integer	Number of component instances to be created from this component descriptor. By default, this number is equal to 1. If the value is 2 or more, the actual name assigned to each instance is the "name" value concatenated with a sequential number (e.g., if name="mymachine", and quantity=3, the names are: mymachine1, mymachine2, mymachine3.) Constraints: Provider: support optional; mutable Consumer: support optional; read-write
		Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	
Meter Templates	Meter Templates[]	A list of references to MeterTemplates that shall be used to create and connect a set of new Meters to the new System. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write	
eventLog Template	ref	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new System. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write	
Import Image	ref	If the Template is the result of an import – e.g., of an OVF package - this attribute should be used. If present, it shall reference the import source (e.g., OVF package) used to	

Name	SystemTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/SystemTemplate	
Attribute	Type	Description
		create this Template. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

2699 When implementing or using SystemTemplate, Providers and Consumers shall adhere to the syntax
 2700 and semantics of its attributes as described in the above table as well as in the tables describing
 2701 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
 2702 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 2703 Resource in both JSON and XML

2704 **JSON media type:** application/json

2705 **JSON serialization:**

```

2706 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
2707   "id": string,
2708   "name": string, ?
2709   "description": string, ?
2710   "created": string, ?
2711   "updated": string, ?
2712   "properties": { string: string, + }, ?
2713   "componentDescriptors": [
2714     { "name": string, ?
2715       "description": string, ?
2716       "properties": { string: string, + }, ?
2717       "type": string,
2718       "componentTemplate": {
2719         "href": string, ?
2720         ... ComponentTemplate attributes ... ?
2721       },
2722       "quantity": number ?
2723     }, +
2724   ], ?
2725   "meterTemplates": [
2726     { "href": string, ?
2727       ... MeterTemplate attributes ... ?
2728     }, *
2729   ], ?
2730   "eventLogTemplate": {
2731     "href": string, ?
2732     ... EventLogTemplate attributes ... ?
2733   }, ?

```

```

2734     "importImage": { "href": string }, ?
2735
2736     "operations": [
2737         { "rel": "edit", "href": string }, ?
2738         { "rel": "delete", "href": string }, ?
2739         { "rel": "http://schemas.dmtf.org/cimi/1/action/export", "href": string } ?
2740     ] ?
2741     ...
2742 }

```

2743 **XML media type:** application/xml

2744 **XML serialization:**

```

2745 <SystemTemplate xmlns="http://schemas.dmtf.org/cimi/1">
2746   <id> xs:anyURI </id>
2747   <name> xs:string </name> ?
2748   <description> xs:string </description> ?
2749   <created> xs:dateTime </created> ?
2750   <updated> xs:dateTime </updated> ?
2751   <property key="xs:string"> xs:string </property> *
2752   <componentDescriptor>
2753     <name> xs:string </name> ?
2754     <description> xs:string </description> ?
2755     <property key="xs:string"> xs:string </property> *
2756     <type> xs:anyURI </type>
2757     <componentTemplate href="xs:anyURI"? >
2758       ... ComponentTemplate attributes ... ?
2759     </componentTemplate> *
2760
2761     <quantity> xs:integer </quantity>
2762   </componentDescriptor> *
2763   <meterTemplate href="xs:anyURI"? >
2764     ... MeterTemplate attributes ... ?
2765   </meterTemplate> *
2766   <eventLogTemplate href="xs:anyURI"? >
2767     ... EventLogTemplate attributes ... ?
2768   </eventLogTemplate> ?
2769   <importImage href="xs:anyURI"? >
2770   <operation rel="edit" href="xs:anyURI"/> ?
2771   <operation rel="delete" href="xs:anyURI"/> ?
2772   <operation rel="http://schemas.dmtf.org/cimi/1/action/export"
2773   href="xs:anyURI"/> ?

```

```

2774     <xs:any>*
2775 </SystemTemplate>

```

2776 5.13.3.1 Operations

2777 This Resource supports the Read, Update, and Delete operations. Create is supported through the
2778 SystemTemplateCollection Resource.

2779 The following custom operations are also defined:

2780 **export**

2781 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/export

2782 This operation shall export a SystemTemplate. If an export package exists at that URI, it is updated
2783 with the values of the SystemTemplate and any component management Resources. Otherwise a
2784 new export package is created at that URI with a Media Type as specified by the "format" parameter.
2785 Other formats may be used if supported, but are not specified by this standard.

2786 Input parameters:

- 2787 1) "format" - type: string - optional
2788 Indicates the Media Type of the exported data. If not present, the default value shall be
2789 "application/ovf."
- 2790 2) "destination" - type: URI - optional
2791 Indicates the location to where the exported data is placed. If not present, the HTTP response
2792 Location header shall contain the URL to the exported data. Based on the specific protocol
2793 specified within the URI, the Consumer might need to provide additional information (such as
2794 credentials) in the "properties" field. In the case of HTTP, a PUT shall be used to place the data
2795 at the specified location.

2796 Output parameters: None.

2797 **HTTP protocol**

2798 To export a SystemTemplate, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/export"
2799 URI of the SystemTemplate where the HTTP request body shall be as described below.

2800 **JSON media type:** application/json

2801 **JSON serialization:**

```

2802 { "action": "http://schemas.dmtf.org/cimi/1/action/export",
2803   "format": string, ?
2804   "destination": string, ?
2805   "properties": { string: string, + } ?
2806   ...
2807 }

```

2808 **XML media type:** application/xml

2809 **XML serialization**

```

2810 <Action xmlns="http://schemas.dmtf.org/cimi/1">
2811   <action> http://schemas.dmtf.org/cimi/1/action/export </action>
2812   <format> xs:string </format> ?

```

```

2813     <destination> xs:anyURI </destination> ?
2814     <property key="xs:string"> xs:string </property> *
2815     <xs:any>*
2816 </Action>

```

5.13.4 SystemTemplateCollection Resource

A SystemTemplateCollection Resource represents the Collection of SystemTemplate Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

```

2822 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplateCollection",
2823     "id": string,
2824     "count": number,
2825     "systemTemplates": [
2826         { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
2827           "id": string,
2828           ... remaining SystemTemplate attributes ...
2829         }, +
2830     ], ?
2831     "operations": [
2832         { "rel": "add", "href": string }, ?
2833         { "rel": "http://schemas.dmtf.org/cimi/1/action/import", "href": string } ?
2834     ]
2835     ...
2836 }

```

XML serialization:

```

2838 <Collection
2839     resourceURI="http://schemas.dmtf.org/cimi/1/SystemTemplateCollection"
2840     xmlns="http://schemas.dmtf.org/cimi/1">
2841     <id> xs:anyURI </id>
2842     <count> xs:integer </count>
2843     <SystemTemplate>
2844         <id> xs:anyURI </id>
2845         ... remaining SystemTemplate attributes ...
2846     </SystemTemplate> *
2847     <operation rel="add" href="xs:anyURI"/> ?
2848     <operation rel="http://schemas.dmtf.org/cimi/1/action/import"
2849 href="xs:anyURI"/> ?
2850     <xs:any>*
2851 </Collection>

```

2852 **5.13.4.1 Operations**

2853 The following custom operations are defined:

2854 **import**2855 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/import`

2856 This operation shall import a `SystemTemplate`. Not only is a `SystemTemplate` created, but
 2857 `MachineTemplates`, `VolumeTemplates`, and `NetworkTemplates` and possibly recursive
 2858 `SystemTemplates` and their components may also be created, corresponding to imported descriptor
 2859 entries. More detail about this process is in ANNEX A.

2860 Input parameters:

- 2861 1) "source" - type: URI - mandatory
 2862 Indicates the location from which the imported data is retrieved. Based on the specific protocol
 2863 specified within the URI, the Consumer might need to provide additional information (such as
 2864 credentials) in the "properties" field.

2865 Output parameters: None.

2866 **HTTP protocol**

2867 To import a `SystemTemplate`, a POST is sent to the "`http://schemas.dmtf.org/cimi/1/action/import`"
 2868 URI of the `SystemTemplateCollection` where the HTTP request body shall be as described
 2869 below.

2870 **JSON media type:** `application/json`2871 **JSON serialization:**

```
2872 { "action": "http://schemas.dmtf.org/cimi/1/action/import",
2873   "source": string, ?
2874   "properties": { string: string, + } ?
2875   ...
2876 }
```

2877 **XML media type:** `application/xml`2878 **XML serialization**

```
2879 <Action xmlns="http://schemas.dmtf.org/cimi/1">
2880   <action> http://schemas.dmtf.org/cimi/1/action/import </action>
2881   <source> xs:anyURI </source> ?
2882   <property key="xs:string"> xs:string </property> *
2883   <xs:any>*
2884 </Action>
```

2885 **5.14 Machine Resources and relationships**

2886 Figure 3 illustrates the Resources involved in constructing a `Machine` and their relationships. Although
 2887 this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor
 2888 normative.

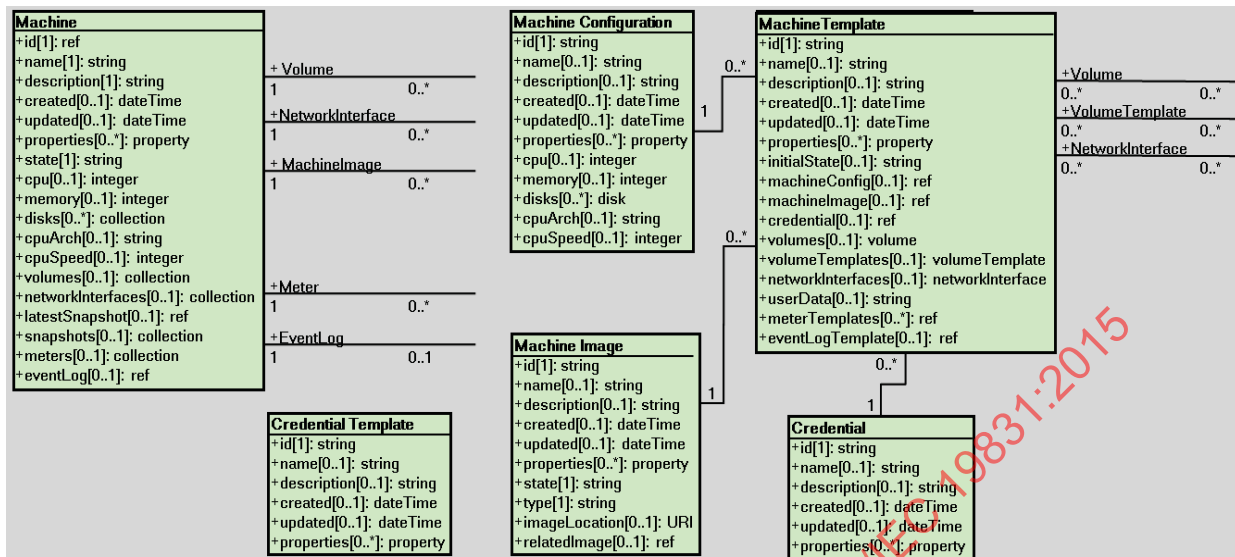


Figure 3 - Machine Resources

5.14.1 Machine

An instantiated compute Resource that encapsulates both CPU and Memory. Table 19 describes the Machine attributes.

Table 19 – Machine attributes

Name	Machine	
Type URI	http://schemas.dmtf.org/cimi/1/Machine	
Attribute	Type	Description
state	string	The operational state of the Machine. Allowable values include: CREATING: The Machine is in the process of being created. STARTING: The Machine is in the process of being started. STARTED: The Machine is available and ready for use. STOPPING: The Machine is in the process of being stopped. STOPPED: This value is the virtual equivalent of powering off a physical Machine. There is no saved CPU or memory state. Clause 5.14.2.1 defines the initial state of a Machine. PAUSING: The Machine in the process of being PAUSED. PAUSED: In this state the Machine and its virtual resources remain instantiated and resources remain allocated, similar to the "STARTED" state, but the Machine and its virtual resources are not enabled to perform tasks. SUSPENDING: The Machine is in the process of being suspended. SUSPENDED: In this state the Machine and its virtual resources are stored on non-volatile storage. The Machine and its resources are not enabled to perform tasks. DELETING: The Machine is in the process of being deleted. ERROR: The Provider has detected an error in the Machine. The operations that result in transitions to the above defined states are defined in clause 5.14.1.2. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
cpu	integer	The amount of CPU that this Machine has.

Name	Machine	
Type URI	http://schemas.dmtf.org/cimi/1/Machine	
Attribute	Type	Description
		Constraints: Provider: support optional; mutable Consumer: support optional; read-write
memory	integer	The size of the memory (RAM) in kibibytes allocated to this Machine. If this value is increased, it implies that the Machine is allocated more RAM, and vice versa if the value is decreased. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
disks	collection [Disk]	A reference to the list of disks (local storage) that are part of the Machine. Adding an element to this list creates a disk. Note: The Disk Resource type is defined in clause 5.14.1.1.1. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
cpuArch	string	The CPU architecture that is supported by Machines created by using this configuration. Allowable values include: 68000, Alpha, ARM, Itanium, MIPS, PA_RISC, POWER, PowerPC, x86, x86_64, z/Architecture, SPARC . Providers may define additional values. Constraints: Provider: support optional; immutable Consumer: support optional; read-only
cpuSpeed	integer	The approximate CPU speed of this Machine - in megahertz. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
volumes	collection [Machine Volume]	A reference to the list of references to Volumes that are connected to this Machine. Adding a Volume to this list means that the Machine has some access to the data on the Volume. Removing a Volume from this list means that the Machine no longer has access to the data on the Volume. Note: The MachineVolume Resource type represents an association between the Machine and a Volume. It is defined in clause 5.14.1.1.2. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkInterfaces	collection [Machine Network Interface]	A reference to the list of MachineNetworkInterfaces on this Machine. Note: The MachineNetworkInterface Resource type represents an association between the Machine and a NetworkInterface. It is defined in clause 5.14.1.1.3. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
latestSnapshot	ref	A reference to the SNAPSHOT representing the latest state captured for this Machine (either most recent Snapshot or the last Snapshot reverted to). Constraints: Provider: support optional; mutable Consumer: support optional; read-only
snapshots	collection [Machine Snapshot]	A reference to the list of references to the SNAPSHOT Machine Images taken of this Machine. Note: The MachineSnapshot Resource type represents an association between the Machine and a Snapshot. It is defined in clause 5.14.1.1.5. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

Name	Machine	
Type URI	http://schemas.dmtf.org/cimi/1/Machine	
Attribute	Type	Description
meters	<i>collection</i> <i>[Meter]</i>	A reference to the list of Meters monitored for this Machine. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLog	<i>ref</i>	A reference to the EventLog of this Machine. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

When implementing or using Machine, Providers and Consumers shall adhere to the syntax and semantics of its attributes as described in the above table, as well as in the tables describing embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "cpu": number,
  "memory": number,
  "disks" : { "href": string }, ?
  "cpuArch": string, ?
  "cpuSpeed": number, ?
  "volumes": { "href": string }, ?
  "networkInterfaces": { "href": string }, ?
  "latestSnapshot": { "href": string }, ?
  "snapshots": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/restart", "href": string },
  ]
}
```

```

2927     { "rel": "http://schemas.dmtf.org/cimi/1/action/pause", "href": string }, ?
2928     { "rel": "http://schemas.dmtf.org/cimi/1/action/suspend", "href": string }
2929 ?
2930     { "rel": "http://schemas.dmtf.org/cimi/1/action/snapshot", "href": string }
2931 ?
2932     { "rel": "http://schemas.dmtf.org/cimi/1/action/restore", "href": string }
2933 ?
2934 ]
2935 ...
2936 }

```

2937 **XML media type:** application/xml

2938 **XML serialization:**

```

2939 <Machine xmlns="http://schemas.dmtf.org/cimi/1">
2940   <id> xs:anyURI </id>
2941   <name> xs:string </name> ?
2942   <description> xs:string </description> ?
2943   <created> xs:dateTime </created> ?
2944   <updated> xs:dateTime </updated> ?
2945   <property key="xs:string"> xs:string </property> *
2946   <state> xs:string </state>
2947   <cpu> xs:integer </cpu>
2948   <memory> xs:integer </memory>
2949   <disks href="xs:anyURI"/> ?
2950   <cpuArch> xs:string </cpuArch> ?
2951   <cpuSpeed> xs:integer </cpuSpeed> ?
2952   <volumes href="xs:anyURI"/> ?
2953   <networkInterfaces href="xs:anyURI"/> ?
2954   <latestSnapshot href="xs:anyURI"/> ?
2955   <snapshots href="xs:anyURI"/> ?
2956   <meters href="xs:anyURI"/> ?
2957   <eventLog href="xs:anyURI"/> ?
2958   <operation rel="edit" href="xs:anyURI"/> ?
2959   <operation rel="delete" href="xs:anyURI"/> ?
2960   <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
2961   href="xs:anyURI"/> ?
2962   <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
2963   href="xs:anyURI"/> ?
2964   <operation rel="http://schemas.dmtf.org/cimi/1/action/restart"
2965   href="xs:anyURI"/> ?
2966   <operation rel="http://schemas.dmtf.org/cimi/1/action/pause"
2967   href="xs:anyURI"/> ?
2968   <operation rel="http://schemas.dmtf.org/cimi/1/action/suspend"

```

```

2969 href="xs:anyURI"/> ?
2970 <operation rel="http://schemas.dmtf.org/cimi/1/action/capture"
2971 href="xs:anyURI"/> ?
2972 <operation rel="http://schemas.dmtf.org/cimi/1/action/snapshot"
2973 href="xs:anyURI"/> ?
2974 <operation rel="http://schemas.dmtf.org/cimi/1/action/restore"
2975 href="xs:anyURI"/> ?
2976 <xs:any>*
2977 </Machine>

```

2978 5.14.1.1 Collections

2979 The following clause describes the Collection Resources owned by Machines.

2980 5.14.1.1.1 Disk Collection

2981 The Resource type for each item of this Collection is "Disk", defined in Table 20.

2982 **Table 20 – Disk attributes**

Name	Disk	
Type URI	http://schemas.dmtf.org/cimi/1/Disk	
Attribute	Type	Description
capacity	<i>integer</i>	The initial capacity, in kilobytes, of the disk. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
initialLocation	<i>string</i>	Operating System-specific location (path) in its namespace where this disk first appears. After deployment, Consumers may consider moving the location of this Disk.. Support of this attribute indicates that the Provider can report this information back to the Consumer. Constraints: Provider: support optional; immutable Consumer: support optional; read-only

2983 JSON serialization:

```

2984 { "resourceURI": "http://schemas.dmtf.org/cimi/1/DiskCollection",
2985   "id": string,
2986   "count": number,
2987   "disks": [
2988     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Disk",
2989       "id": string,
2990       "name": string, ?
2991       "description": string, ?
2992       "created": string, ?
2993       "updated": string, ?
2994       "properties": { string: string, + }, ?
2995       "capacity": number,
2996       "initialLocation": string, ?
2997       "operations": [

```

ISO/IEC 19831:2015 (E)

```

2998     { "rel": "edit", "href": string }, ?
2999     { "rel": "delete", "href": string } ?
3000   ] ?
3001   ...
3002 }, +
3003 ], ?
3004 "operations": [ { "rel": "add", "href": string } ? ]
3005   ...
3006 }

```

XML serialization:

```

3007 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/DiskCollection"
3008     xmlns="http://schemas.dmtf.org/cimi/1">
3009   <id> xs:anyURI </id>
3010   <count> xs:integer </count>
3011   <Disk>
3012     <id> xs:anyURI </id>
3013     <name> xs:string </name> ?
3014     <description> xs:string </description> ?
3015     <created> xs:dateTime </created> ?
3016     <updated> xs:dateTime </updated> ?
3017     <property key="xs:string"> xs:string </property> *
3018     <capacity> xs:integer </capacity>
3019     <initialLocation> xs:string </initialLocation> ?
3020     <operation rel="edit" href="xs:anyURI"/> ?
3021     <operation rel="delete" href="xs:anyURI"/> ?
3022     <xs:any>*
3023   </Disk> *
3024   <operation rel="add" href="xs:anyURI"/> ?
3025   <xs:any>*
3026 </Collection>

```

5.14.1.1.2 MachineVolumeCollection Resource

The Resource type for each item of this Collection is "MachineVolume", defined in Table 21:

Table 21 – MachineVolume attributes

Name	MachineVolume	
Type URI	http://schemas.dmtf.org/cimi/1/MachineVolume	
Attribute	Type	Description
initialLocation	string	Operating System-specific location (path) in its namespace where this Volume first appears. Note, once deployed, Consumers might move the location of this Volume. Support of this attribute indicates that the Provider can report this information back to the Consumer. Constraints:

Name	MachineVolume	
Type URI	http://schemas.dmtf.org/cimi/1/MachineVolume	
Attribute	Type	Description
		Provider: support optional; immutable Consumer: support optional; read-only
volume	ref	A reference to the Volume that is connected. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

JSON serialization:

```

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

```

XML serialization:

```

<collection
  resourceURI="http://schemas.dmtf.org/cimi/1/MachineVolumeCollection"
  xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <MachineVolume>
    <id> xs:anyURI </id>
    <name> xs:string </name> ?

```

```

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

```

5.14.1.1.3 MachineNetworkInterfaceCollection Resource

The Resource type for each item of this Collection is "MachineNetworkInterface", defined in Table 22:

Table 22 – MachineNetworkInterface attributes

Name	MachineNetworkInterface	
Type URI	http://schemas.dmtf.org/cimi/1/MachineNetworkInterface	
Attribute	Type	Description
addresses	<i>collection</i> [Machine Network Interface Address]	A reference to the list of references to the Addresses for this network interface. Note: the MachineNetworkInterfaceAddress Resource type represents an association between the MachineNetworkInterface and an Address. It is defined in clause 5.14.1.1.4. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
network	<i>ref</i>	A reference to a Network for this network interface. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
networkPort	<i>ref</i>	A reference to the NetworkPort for this network interface. If this attribute is provided, the "network" attribute in the referenced NetworkPort shall have the same value as the "network" attribute in this network Interface. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
state	<i>string</i>	The state of the MachineNetworkInterface. Allowable values include: ACTIVE: An active interface is the primary interface, able to forward traffic. PASSIVE: A passive interface is in a standby mode ready to forward traffic if the primary interface fails. DISABLED: A disabled interface is one that is not able to forward traffic. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
macAddress	<i>string</i>	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 if the Template is only used for one particular Machine. Constraints:

Name	MachineNetworkInterface	
Type URI	http://schemas.dmtf.org/cimi/1/MachineNetworkInterface	
Attribute	Type	Description
		Provider: support optional; mutable Consumer: support optional; read-write
mtu	<i>integer</i>	To set the largest supported maximum transmission unit packet size. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

JSON serialization:

```

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

```

XML serialization:

```

<Collection
  resourceURI="http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceCollection"
  xmlns="http://schemas.dmtf.org/cimi/1">

```

```

<id> xs:anyURI </id>
<count> xs:integer </count>
<MachineNetworkInterface>
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <addresses href="xs:anyURI"/>
  <network href="xs:anyURI"/>
  <networkPort href="xs:anyURI"/> ?
  <state> xs:string </state> ?
  <macAddress> xs:string </macAddress> ?
  <mtu> xs:integer </mtu> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineNetworkInterface> *
<operation rel="add" href="xs:anyURI"/> ?
<xs:any>*
</Collection>

```

5.14.1.1.4 MachineNetworkInterfaceAddressCollection Resource

The Resource type for each item of this Collection is "MachineNetworkInterfaceAddress", defined in Table 23:

Table 23 – MachineNetworkInterfaceAddress attributes

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

JSON serialization:

```

{ "resourceURI":
  "http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddressCollection",
  "id": string,
  "count": number,
  "machineNetworkInterfaceAddresses": [
    { "resourceURI":
      "http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddress",

```

```

3147     "id": string,
3148     "name": string, ?
3149     "description": string, ?
3150     "created": string, ?
3151     "updated": string, ?
3152     "properties": { string: string, + }, ?
3153     "address": { "href": string },
3154     "operations": [
3155         { "rel": "edit", "href": string }, ?
3156         { "rel": "delete", "href": string } ?
3157     ] ?
3158     ...
3159 }, +
3160 ], ?
3161 "operations": [ { "rel": "add", "href": string } ? ]
3162 ...
3163 }

```

XML serialization:

```

3165 <Collection
3166 resourceURI="http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddressColle
3167 ction"
3168     xmlns="http://schemas.dmtf.org/cimi/1">
3169     <id> xs:anyURI </id>
3170     <count> xs:integer </count>
3171     <MachineNetworkInterfaceAddress>
3172         <id> xs:anyURI </id>
3173         <name> xs:string </name> ?
3174         <description> xs:string </description> ?
3175         <created> xs:dateTime </created> ?
3176         <updated> xs:dateTime </updated> ?
3177         <property key="xs:string"> xs:string </property> *
3178         <address href="xs:anyURI"/>
3179         <operation rel="edit" href="xs:anyURI"/> ?
3180         <operation rel="delete" href="xs:anyURI"/> ?
3181         <xs:any>*
3182     </MachineNetworkInterfaceAddress> *
3183     <operation rel="add" href="xs:anyURI"/> ?
3184     <xs:any>*
3185 </Collection>

```

5.14.1.1.5 MachineSnapshotCollection Resource

The Resource type for each item of this Collection is "MachineSnapshot", defined in Table 24:

Table 24 – MachineSnapshot attributes

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

JSON serialization:

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

XML serialization:

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

```

3220     <description> xs:string </description> ?
3221     <created> xs:dateTime </created> ?
3222     <updated> xs:dateTime </updated> ?
3223     <property key="xs:string"> xs:string </property> *
3224     <snapshot href="xs:anyURI"/>
3225     <operation rel="edit" href="xs:anyURI"/> ?
3226     <operation rel="delete" href="xs:anyURI"/> ?
3227     <xs:any>*
3228 </MachineSnapshot> *
3229 <xs:any>*
3230 </Collection>

```

NOTE Previous versions of this specification included an "add" operation on this Resource. It is now deprecated in favor of creating a new MachineImage with the imageLocation attribute pointing to the Machine to be taken a snapshot from.

5.14.1.1.6 MachineMeterCollection Resource

The Resource type for each item of this Collection is "Meter" as defined in clause 5.17.3.

JSON serialization:

```

3237 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineMeterCollection",
3238   "id": string,
3239   "count": number,
3240   "meters": [
3241     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
3242       "id": string,
3243       ... remaining Meter attributes ...
3244     }, +
3245   ], ?
3246   "operations": [ { "rel": "add", "href": string } ? ]
3247   ...
3248 }

```

XML serialization:

```

3250 <Collection
3251   resourceURI="http://schemas.dmtf.org/cimi/1/MachineMeterCollection"
3252   xmlns="http://schemas.dmtf.org/cimi/1">
3253   <id> xs:anyURI </id>
3254   <count> xs:integer </count>
3255   <Meter>
3256     <id> xs:anyURI </id>
3257     ... remaining Meter attributes ...
3258   </Meter> *

```

```

3259     <operation rel="add" href="xs:anyURI"/> ?
3260     <xs:any>*
3261   </Collection>

```

3262 5.14.1.2 Operations

3263 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 3264 MachineCollection Resource.

3265 The following custom operations are also defined:

3266 **start**

3267 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/start`

3268 This operation shall start a Machine.

3269 Input parameters: None.

3270 Output parameters: None.

3271 During the processing of this operation, the Machine shall be in the "STARTING" state.

3272 Upon successful completion of this operation, the Machine shall be in the "STARTED" state.

3273 If a Machine is in the "STOPPED" state, starting it shall be the virtual equivalent of powering on a
 3274 physical machine. There is no restored CPU or Memory state, so the guest OS typically performs boot or
 3275 installation tasks.

3276 If the Machine was in the "SUSPENDED" or "PAUSED" state, starting it shall have the effect of
 3277 resuming it.

3278 **HTTP protocol**

3279 To start a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/start" URI of the
 3280 Machine where the HTTP request body shall be as described below.

3281 **JSON media type:** application/json

3282 **JSON serialization:**

```

3283   { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3284     "action": "http://schemas.dmtf.org/cimi/1/action/start",
3285     "properties": { string: string, + } ?
3286     ...
3287   }

```

3288 **XML media type:** application/xml

3289 **XML serialization**

```

3290   <Action xmlns="http://schemas.dmtf.org/cimi/1">
3291     <action> http://schemas.dmtf.org/cimi/1/action/start </action>
3292     <property key="xs:string"> xs:string </property> *
3293     <xs:any>*
3294   </Action>

```

3295 Upon successful processing of the request, the HTTP response body may be empty.

3296 **stop**

3297 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/stop`

3298 This operation shall stop a Machine.

3299 Input parameters:

3300 1) "force" - type: boolean - optional

3301 A flag to indicate whether the Provider shall simulate a power off condition (force=true) or shall
3302 simulate a shutdown operation that allows applications to save their state and the file system to
3303 be made consistent (force=false). Inclusion of this parameter by Consumers is optional and if
3304 not specified, the Provider may choose either mechanism. Providers are encouraged to
3305 advertise this choice by the way of the MachineStopForceDefault capability.

3306 Output parameters: None.

3307 During the processing of this operation, the Machine shall be in the "STOPPING" state.

3308 Upon successful completion of this operation, the Machine shall be in the "STOPPED" state. Stopping a
3309 Machine with force=true shall be the virtual equivalent of powering off a physical machine. There is no
3310 saved CPU or Memory state. Stopping a Machine with force=false shall result in a machine with
3311 consistent file systems.

3312 A Consumer may reissue a stop operation if the state is STOPPING, perhaps with force=true, but
3313 Providers shall not issue a force=true stop operation on their own.

3314 **HTTP protocol**

3315 To stop a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the
3316 Machine where the HTTP request body shall be as described below.

3317 **JSON media type:** application/json

3318 **JSON serialization:**

```
3319 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3320   "action": "http://schemas.dmtf.org/cimi/1/action/stop",
3321   "force": boolean, ?
3322   "properties": { string: string, + } ?
3323   ...
3324 }
```

3325 **XML media type:** application/xml

3326 **XML serialization**

```
3327 <Action xmlns="http://schemas.dmtf.org/cimi/1">
3328   <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
3329   <force> xs:boolean </force> ?
3330   <property key="xs:string"> xs:string </property> *
3331   <xs:any>*
3332 </Action>
```

3333 Upon successful processing of the request, the HTTP response body may be empty.

3334 **restart**

3335 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/restart`

3336 This operation shall restart a Machine. If the Machine is in the "STARTED" state, this operation shall
3337 have the effect of executing the "stop" and then "start" operations. If the Machine is in the "STOPPED"
3338 state, this operation shall have the effect of executing the "start" operation.

3339 Input parameters:

3340 1) "force" - type: boolean - optional
3341 A flag to indicate whether the Provider shall simulate a power off condition (force=true) or shall
3342 simulate a shutdown operation that allows applications to save their state and the file system to
3343 be made consistent (force=false). Inclusion of this parameter by Consumers is optional and if
3344 not specified, the Provider may choose either mechanism. Providers are encouraged to
3345 advertise this choice by the way of the MachineStopForceDefault capability.

3346 Output parameters: None.

3347 During the processing of this operation, the Machine shall be in the "STOPPING" and/or "STARTING"
3348 states, as appropriate depending on its initial state.

3349 Upon successful completion of this operation, the Machine shall be in the "STARTED" state. Restarting
3350 a Machine shall be the virtual equivalent of powering off, and then powering on a physical machine.
3351 There is no restored CPU or Memory state, so the guest OS typically performs boot or installation tasks.

3352 **HTTP protocol**

3353 To restart a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/restart" URI of the
3354 Machine where the HTTP request body shall be as described below.

3355 **JSON media type:** application/json

3356 **JSON serialization:**

```
3357 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3358   "action": "http://schemas.dmtf.org/cimi/1/action/restart",
3359   "force": boolean ?
3360   "properties": { string: string, + } ?
3361   ...
3362 }
```

3363 **XML media type:** application/xml

3364 **XML serialization**

```
3365 <Action xmlns="http://schemas.dmtf.org/cimi/1">
3366   <action> http://schemas.dmtf.org/cimi/1/action/restart </action>
3367   <force> xs:boolean </force> ?
3368   <property key="xs:string"> xs:string </property> *
3369   <xs:any>*
3370 </Action>
```

3371 Upon successful processing of the request, the HTTP response body may be empty.

3372 **pause**3373 **//link@rel:** http://schemas.dmtf.org/cimi/1/action/pause

3374 This operation shall pause a Machine.

3375 Input parameters: None.

3376 Output parameters: None.

3377 During the processing of this operation, the Machine shall be in the "PAUSING" state.

3378 Upon successful completion of this operation, the Machine shall be in the "PAUSED" state. Pausing a
 3379 Machine shall keep the Machine and its resources instantiated, but the Machine shall not be
 3380 available to perform any tasks. The current state of the CPU and Memory shall be retained in volatile
 3381 memory.

3382 **HTTP protocol**

3383 To pause a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action.pause" URI of the
 3384 Machine where the HTTP request body shall be as described below.

3385 **JSON media type:** application/json3386 **JSON serialization:**

```
3387 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3388   "action": "http://schemas.dmtf.org/cimi/1/action/pause",
3389   "properties": { string: string, + } ?
3390   ...
3391 }
```

3392 **XML media type:** application/xml3393 **XML serialization**

```
3394 <Action xmlns="http://schemas.dmtf.org/cimi/1">
3395   <action> http://schemas.dmtf.org/cimi/1/action/pause </action>
3396   <property key="xs:string"> xs:string </property> *
3397   <xs:any>*
3398 </Action>
```

3399 Upon successful processing of the request, the HTTP response body may be empty.

3400 **suspend**3401 **//link@rel:** http://schemas.dmtf.org/cimi/1/action/suspend

3402 This operation shall suspend a Machine.

3403 Input parameters: None.

3404 Output parameters: None.

3405 During the processing of this operation, the Machine shall be in the "SUSPENDING" state.

ISO/IEC 19831:2015 (E)

3406 Upon successful completion of this operation, the Machine shall be in the "SUSPENDED" state.
3407 Suspending a Machine shall keep the Machine and its resources instantiated, but the Machine shall
3408 not be available to perform any tasks. The current state of the CPU and Memory shall be retained in
3409 non-volatile memory.

3410 HTTP protocol

3411 To suspend a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/suspend" URI of
3412 the Machine where the HTTP request body shall be as described below.

3413 **JSON media type:** application/json

3414 JSON serialization:

```
3415 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",  
3416   "action": "http://schemas.dmtf.org/cimi/1/action/suspend",  
3417   "properties": { string: string, + } ?  
3418   ...  
3419 }
```

3420 **XML media type:** application/xml

3421 XML serialization

```
3422 <Action xmlns="http://schemas.dmtf.org/cimi/1">  
3423   <action> http://schemas.dmtf.org/cimi/1/action/suspend </action>  
3424   <property key="xs:string"> xs:string </property> *  
3425   <xs:any>*</xs:any>  
3426 </Action>
```

3427 Upon successful processing of the request, the HTTP response body may be empty.

3428 capture

3429 **//link@rel:** http://schemas.dmtf.org/cimi/1/action/capture

3430 This operation shall create a new MachineImage from an existing Machine. This operation is
3431 defined within the MachineImage Resource; see 5.14.7.1 for more details. Note that while this
3432 operation is performed against a MachineImage, its presence in the Machine serialization is used to
3433 advertise support for the operation.

3434 Snapshotting a Machine

3435 **//link@rel:** http://schemas.dmtf.org/cimi/1/action/snapshot

3436 This operation shall create a new SNAPSHOT MachineImage from an existing Machine. This
3437 operation is defined within the MachineImage Resource; see 5.14.7.1 for more details. Note that while
3438 this operation is performed against a MachineImage, its presence in the Machine serialization is
3439 used to advertise support for the operation.

3440 Restoring a Machine

3441 **//link@rel:** http://schemas.dmtf.org/cimi/1/action/restore

3442 This operation shall restore a Machine from a previously created MachineImage.

3443 Input parameters:

- 3444 1) "image" - type: URI - mandatory
3445 A reference to the Machine Image.

3446 Output parameters: None.

3447 During the processing of this operation, the Machine shall be in the "RESTORING" state.

3448 Upon successful completion of this operation, the Machine shall be in the same state as the state
3449 specified in the MachineImage, if specified. See 5.14.2.1 for more details.

3450 Note that Providers can indicate support for restoring from non-SNAPSHOT MachineImages by the
3451 way of the Machine "RestoreFromImage" capability. If the RestoreFromImage capability is not supported,
3452 and the restore operation is supported, then the restore operation can only restore from a SNAPSHOT
3453 MachineImage.

3454 HTTP protocol

3455 To restore a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/restore" URI of the
3456 Machine where the HTTP request body shall be as described below.

3457 **JSON media type:** application/json

3458 **JSON serialization:**

```
3459 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3460   "action": "http://schemas.dmtf.org/cimi/1/action/restore",
3461   "image": string,
3462   "properties": { string: string, + }?
3463   ...
3464 }
```

3465 **XML media type:** application/xml

3466 **XML serialization**

```
3467 <Action xmlns="http://schemas.dmtf.org/cimi/1">
3468   <action> http://schemas.dmtf.org/cimi/1/action/restore </action>
3469   <image href="xs:anyURI"/>
3470   <property key="xs:string"> xs:string </property> *
3471   <xs:any>*
3472 </Action>
```

3473 Where the "image" URI is a reference to the MachineImage to be used.

3474 Upon successful processing of the request, the HTTP response body may be empty.

3475 5.14.2 MachineCollection

3476 A MachineCollection Resource represents the Collection of Machine Resources within a
3477 Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as
3478 follows:

JSON serialization:

```

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

```

XML serialization:

```

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

```

5.14.2.1 Operations

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

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

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

3520 influenced by the data within the MachineImage used to create the new Machine. For example, if a
 3521 new Machine's initialState is "STARTED" and a SNAPSHOT MachineImage was used to create the
 3522 new Machine, the Machine would not be "booted" but rather resume executing from the saved state in
 3523 the MachineImage.

3524 If a Provider is unable to change the state of the new Machine to the appropriate "initialState" (either as
 3525 specified by the MachineTemplate or as implied by the previous stated rules), the Machine creation
 3526 shall fail.

3527 If a Provider is unable to create the new Machine due to invalid or inconsistent credentials in the
 3528 MachineTemplate, the Machine creation process shall fail. If any credentials are included in the
 3529 MachineTemplate, they shall be part of the new Machine regardless of the type of
 3530 MachineImage used.

3531 5.14.3 MachineTemplate

3532 A MachineTemplate represents the set of metadata and instructions used in the creation of a
 3533 Machine. Table 25 describes the MachineTemplate attributes.

3534 **Table 25 – MachineTemplate attributes**

Name	MachineTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/MachineTemplate	
Attribute	Type	Description
initialState	string	The initial state of the new Machine. Possible values include the non-transient states as specified by the Machine "state" attribute (e.g., STARTED, STOPPED) and are determined by the actions supported by the Provider. Providers should advertise the list of available values through the Machine's "initialStates" capability. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
machineConfig	ref	A reference to the MachineConfiguration that is used to create a Machine from this MachineTemplate. Note that the attributes of the MachineConfiguration may be specified rather than a reference to an existing MachineConfiguration Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
machineImage	ref	A reference to the MachineImage that is used to create a Machine from this MachineTemplate. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
credential	ref	A reference to the Credential that is used to create the initial login credentials for the new Machine. Note that the attributes of the Credential may be specified rather than a reference to an existing Credential Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

Name	MachineTemplate													
Type URI	http://schemas.dmtf.org/cimi/1/MachineTemplate													
Attribute	Type	Description												
volumes	volume[]	A list of structures, each containing a reference to an existing Volume and potentially describing aspects of the way that the given Volume is to be connected to the Machine during its creation from this MachineTemplate. Each volume structure has the following attributes: <table> <tr> <td>Name</td><td colspan="2">volume</td></tr> <tr> <td>Attribute</td><td>Type</td><td>Description</td></tr> <tr> <td>initialLocation</td><td>string</td><td> An Operating System-specific location (path) in its namespace where the Volume appears. Support of this attribute indicates that the Provider allows for Consumers to choose where the Volume appears. Constraints: Provider: support optional; mutable Consumer: support optional; read-write </td></tr> <tr> <td>volume</td><td>ref</td><td> Reference to the Volume that is connected. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write </td></tr> </table> Constraints: Provider: support optional; mutable Consumer: support optional; read-write	Name	volume		Attribute	Type	Description	initialLocation	string	An Operating System-specific location (path) in its namespace where the Volume appears. Support of this attribute indicates that the Provider allows for Consumers to choose where the Volume appears. Constraints: Provider: support optional; mutable Consumer: support optional; read-write	volume	ref	Reference to the Volume that is connected. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
Name	volume													
Attribute	Type	Description												
initialLocation	string	An Operating System-specific location (path) in its namespace where the Volume appears. Support of this attribute indicates that the Provider allows for Consumers to choose where the Volume appears. Constraints: Provider: support optional; mutable Consumer: support optional; read-write												
volume	ref	Reference to the Volume that is connected. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write												

Name	MachineTemplate													
Type URI	http://schemas.dmtf.org/cimi/1/MachineTemplate													
Attribute	Type	Description												
volumeTemplates	<i>volumeTemplate[]</i>	A list of structures, each containing a reference to a VolumeTemplate from which a Volume is created and connected to the Machine resulting from this MachineTemplate. Each structure can potentially also include aspects of the way in which each created Volume is connected to the created Machine. If the Machine is created as part of a System creation, the Volumes created from these Templates are considered as part of that System without the need for these VolumeTemplates to also be listed in the volumeTemplates attribute of the relevant SystemTemplate. If the same VolumeTemplate reference is listed in both the volumeTemplates attribute of a SystemTemplate and in the volumeTemplates attribute of a MachineTemplate contained by that SystemTemplate, this means that multiple, distinct Volume instances are created as part of the overall System creation. Each volumeTemplate has the following attributes: <table> <tr> <td>Name</td><td colspan="2"><i>volumeTemplate</i></td></tr> <tr> <td>Attribute</td><td>Type</td><td>Description</td></tr> <tr> <td>initialLocation</td><td><i>string</i></td><td> An Operating System-specific location (path) in its namespace where the Volume appears. Support of this attribute indicates that the Provider allows for Consumers to choose where the Volume appears. Constraints: Provider: support optional; mutable Consumer: support optional; read-write </td></tr> <tr> <td>volumeTemplate</td><td><i>ref</i></td><td> Reference to the VolumeTemplate that is used to create a new Volume. Note that the attributes of the VolumeTemplate may be specified rather than a reference to an existing VolumeTemplate Resource. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write </td></tr> </table> Constraints: Provider: support optional; mutable Consumer: support optional; read-write	Name	<i>volumeTemplate</i>		Attribute	Type	Description	initialLocation	<i>string</i>	An Operating System-specific location (path) in its namespace where the Volume appears. Support of this attribute indicates that the Provider allows for Consumers to choose where the Volume appears. Constraints: Provider: support optional; mutable Consumer: support optional; read-write	volumeTemplate	<i>ref</i>	Reference to the VolumeTemplate that is used to create a new Volume. Note that the attributes of the VolumeTemplate may be specified rather than a reference to an existing VolumeTemplate Resource. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
Name	<i>volumeTemplate</i>													
Attribute	Type	Description												
initialLocation	<i>string</i>	An Operating System-specific location (path) in its namespace where the Volume appears. Support of this attribute indicates that the Provider allows for Consumers to choose where the Volume appears. Constraints: Provider: support optional; mutable Consumer: support optional; read-write												
volumeTemplate	<i>ref</i>	Reference to the VolumeTemplate that is used to create a new Volume. Note that the attributes of the VolumeTemplate may be specified rather than a reference to an existing VolumeTemplate Resource. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write												

Name	MachineTemplate																						
Type URI	http://schemas.dmtf.org/cimi/1/MachineTemplate																						
Attribute	Type	Description																					
networkInterfaces	<i>networkInterface[]</i>	A list of structures, each containing references to the Resources and attributes defining a network interface to be created on a Machine instantiated from this MachineTemplate. The Resources referenced by each networkInterface structure are a Network, a NetworkPort, and a list of Addresses: <table> <tr> <td>Name</td><td colspan="2"><i>networkInterface</i></td></tr> <tr> <td>Attribute</td><td>Type</td><td>Description</td></tr> <tr> <td>addresses</td><td><i>ref[]</i></td><td> A list of references to the Addresses for this network interface. Array item name: address Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only </td></tr> <tr> <td>network</td><td><i>ref</i></td><td> A reference to the Network for this network interface. It is expected that NetworkPorts and Networks are defined separately and prior to the Machines that connect to them. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write </td></tr> <tr> <td>networkPort</td><td><i>ref</i></td><td> A reference to the NetworkPort for this network interface. Note this is a reference to a NetworkPort and not a NetworkPortTemplate. It is expected that NetworkPorts and Networks are defined separately and prior to the Machines that connect to them. If this attribute is provided, the "network" attribute in the referenced NetworkPort shall have the same value as the "network" attribute in this network Interface. Constraints: Provider: support optional; mutable Consumer: support optional; read-write </td></tr> <tr> <td>state</td><td><i>string</i></td><td> The state of the network interface. Allowable values include: ACTIVE: An active interface is the primary interface, able to forward traffic. PASSIVE: A passive interface is in a standby mode ready to forward traffic if the primary interface fails. DISABLED: A disabled interface is one that is not able to forward traffic. Constraints: Provider: support optional; mutable Consumer: support optional; read-write </td></tr> <tr> <td>mtu</td><td><i>integer</i></td><td> To set the largest supported packet size. Constraints: Provider: support optional; mutable Consumer: support optional; read-write </td></tr> </table> Constraints: Provider: support optional; mutable Consumer: support optional; read-write	Name	<i>networkInterface</i>		Attribute	Type	Description	addresses	<i>ref[]</i>	A list of references to the Addresses for this network interface. Array item name: address Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only	network	<i>ref</i>	A reference to the Network for this network interface. It is expected that NetworkPorts and Networks are defined separately and prior to the Machines that connect to them. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	networkPort	<i>ref</i>	A reference to the NetworkPort for this network interface. Note this is a reference to a NetworkPort and not a NetworkPortTemplate. It is expected that NetworkPorts and Networks are defined separately and prior to the Machines that connect to them. If this attribute is provided, the "network" attribute in the referenced NetworkPort shall have the same value as the "network" attribute in this network Interface. Constraints: Provider: support optional; mutable Consumer: support optional; read-write	state	<i>string</i>	The state of the network interface. Allowable values include: ACTIVE: An active interface is the primary interface, able to forward traffic. PASSIVE: A passive interface is in a standby mode ready to forward traffic if the primary interface fails. DISABLED: A disabled interface is one that is not able to forward traffic. Constraints: Provider: support optional; mutable Consumer: support optional; read-write	mtu	<i>integer</i>	To set the largest supported packet size. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
Name	<i>networkInterface</i>																						
Attribute	Type	Description																					
addresses	<i>ref[]</i>	A list of references to the Addresses for this network interface. Array item name: address Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only																					
network	<i>ref</i>	A reference to the Network for this network interface. It is expected that NetworkPorts and Networks are defined separately and prior to the Machines that connect to them. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																					
networkPort	<i>ref</i>	A reference to the NetworkPort for this network interface. Note this is a reference to a NetworkPort and not a NetworkPortTemplate. It is expected that NetworkPorts and Networks are defined separately and prior to the Machines that connect to them. If this attribute is provided, the "network" attribute in the referenced NetworkPort shall have the same value as the "network" attribute in this network Interface. Constraints: Provider: support optional; mutable Consumer: support optional; read-write																					
state	<i>string</i>	The state of the network interface. Allowable values include: ACTIVE: An active interface is the primary interface, able to forward traffic. PASSIVE: A passive interface is in a standby mode ready to forward traffic if the primary interface fails. DISABLED: A disabled interface is one that is not able to forward traffic. Constraints: Provider: support optional; mutable Consumer: support optional; read-write																					
mtu	<i>integer</i>	To set the largest supported packet size. Constraints: Provider: support optional; mutable Consumer: support optional; read-write																					
userData	<i>string</i>	A Base64 encoded string whose decoded version is to be injected into Machines created by using this Template. See the discussion of injection of user-defined data below.																					

Name	MachineTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/MachineTemplate	
Attribute	Type	Description
		Constraints: Provider: support optional; mutable Consumer: support optional; read-write
meterTemplates	<i>meterTemplates[]</i>	A list of references to MeterTemplates that shall be used to create and connect a set of new Meters to the new Machine. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
eventLogTemplate	<i>ref</i>	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new Machine. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

When implementing or using MachineTemplate, Providers and Consumers shall adhere to the syntax and semantics of its attributes as described in the above table, as well as in the tables describing embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML:

JSON media type: application/json

JSON serialization:

```

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "initialState": string, ?
  "machineConfig": {
    "href": string | ... MachineConfiguration attributes ...
  }, ?
  "machineImage": {
    "href": string | ... MachineImage attributes ...
  }, ?
  "credential": {
    "href": string | ... CredentialTemplate attributes ...
  }, ?
  "volumes": [
    { "initialLocation": string?, "href": string }, +

```

```

3561 ], ?
3562 "volumeTemplates": [
3563   { "initialLocation": string?,
3564     "href": string, ?
3565     ... VolumeTemplate attributes ... ?
3566   }, +
3567 ], ?
3568 "networkInterfaces": [
3569   { "addresses": [
3570     {"href": string}, +
3571   ],
3572     "network": {"href": string},
3573     "networkPort": {"href": string}, ?
3574     "state": string,
3575     "mtu": number ?
3576   }, +
3577 ], ?
3578 "userData": string, ?
3579 "meterTemplates": [
3580   { "href": string, ?
3581     ... MeterTemplate attributes ... ?
3582   }, *
3583 ], ?
3584 "eventLogTemplate": {
3585   "href": string, ?
3586   ... EventLogTemplate attributes ... ?
3587 }, ?
3588 "operations": [
3589   { "rel": "edit", "href": string }, ?
3590   { "rel": "delete", "href": string } ?
3591 ] ?
3592 :
3593 }

```

3594 **XML media type:** application/xml

3595 **XML serialization:**

```

3596 <MachineTemplate xmlns="http://schemas.dmtf.org/cimi/1">
3597   <id> xs:anyURI </id>
3598   <name> xs:string </name> ?
3599   <description> xs:string </description> ?

```

```

3600 <created> xs:dateTime </created> ?
3601 <updated> xs:dateTime </updated> ?
3602 <property key="xs:string"> xs:string </property> *
3603 <initialState> xs:string </initialState> ?
3604 <machineConfig href="xs:anyURI"?>
3605     ... MachineConfiguration attributes ... ?
3606 </machineConfig> ?
3607 <machineImage href="xs:anyURI"?>
3608     ... MachineImage attributes ... ?
3609 </machineImage> ?
3610 <credential href="xs:anyURI"?>
3611     ... CredentialTemplate attributes ... ?
3612 </credential> ?
3613 <volume initialLocation="xs:string"? href="xs:anyURI" /> *
3614 <volumeTemplate initialLocation="xs:string"? href="xs:anyURI"? >
3615     ... VolumeTemplate attributes ... ?
3616 </volumeTemplate> *
3617 <networkInterface>
3618     <address href="xs:anyURI"/> *
3619     <network href="xs:anyURI"/>
3620     <networkPort href="xs:anyURI"/> ?
3621     <state> xs:string </state>
3622     <mtu> xs:integer </mtu> ?
3623 </networkInterface> *
3624 <userData> xs:string </userData> ?
3625 <meterTemplate href="xs:anyURI"? >
3626     ... MeterTemplate attributes ... ?
3627 </meterTemplate> *
3628 <eventLogTemplate href="xs:anyURI"? >
3629     ... EventLogTemplate attributes ... ?
3630 </eventLogTemplate> ?
3631 <operation rel="edit" href="xs:anyURI"/> ?
3632 <operation rel="delete" href="xs:anyURI"/> ?
3633 <xs:any>*
3634 </MachineTemplate>

```

3635 Injection of user-defined data

3636 To simplify the customization of individual Machines, it is possible to pass arbitrary data into the new
 3637 Machine by using the userData parameter. The value of this parameter shall be the Base64-encoded
 3638 payload. The Provider shall arrange for this data to be available from inside the Machine by using one
 3639 of the following three methods:

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

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

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

5.14.3.1 Operations

This Resource supports the Read, Update, and Delete operations. Create is supported through the MachineTemplateCollection Resource.

5.14.4 MachineTemplateCollection Resource

A MachineTemplateCollection Resource represents the Collection of MachineTemplate Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

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

XML serialization:

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

```

3680 <MachineTemplate>
3681   <id> xs:anyURI </id>
3682   ... remaining MachineTemplate attributes ...
3683 </MachineTemplate> *
3684 <operation rel="add" href="xs:anyURI"/> ?
3685 <xs:any>*
3686 </Collection>

```

5.14.4.1 Operations

This Resource supports the Read and Update operations. Creation of new MachineTemplate Resources is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.14.5 MachineConfiguration Resource

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

Table 26 describes the MachineConfiguration attributes.

Table 26 – MachineConfiguration attributes

Name	MachineConfiguration																	
Type URI	http://schemas.dmtf.org/cimi/1/MachineConfiguration																	
Attribute	Type	Description																
cpu	integer	The amount of CPU that a Machine realized from this configuration has. Constraints: Provider: support optional; mutable Consumer: support optional; read-write																
memory	integer	The amount of RAM, in kibibytes, that a Machine realized from this configuration has. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																
disks	disk[]	A list of structures, each containing the attributes defining the disks to be created for the Machine instantiated with this MachineConfiguration Resource. The disks are local storage to the Machine. Each disks attribute has the following sub-attributes: <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>integer</td><td>The initial capacity, in kilobytes, of the disk described by this attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td></tr><tr><td>format</td><td>string</td><td>The format/type of this disk (e.g., ext4, NTFS). Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td></tr><tr><td>initialLocation</td><td>string</td><td>An Operating System-specific location (path) in its namespace where this Disk first appears. After creation of a Machine, Consumers may change the location of this Disk. Constraints: Provider: support optional; mutable</td></tr></table>		Name	disk		Attribute	Type	Description	capacity	integer	The initial capacity, in kilobytes, of the disk described by this attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	format	string	The format/type of this disk (e.g., ext4, NTFS). Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	initialLocation	string	An Operating System-specific location (path) in its namespace where this Disk first appears. After creation of a Machine, Consumers may change the location of this Disk. Constraints: Provider: support optional; mutable
Name	disk																	
Attribute	Type	Description																
capacity	integer	The initial capacity, in kilobytes, of the disk described by this attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																
format	string	The format/type of this disk (e.g., ext4, NTFS). Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																
initialLocation	string	An Operating System-specific location (path) in its namespace where this Disk first appears. After creation of a Machine, Consumers may change the location of this Disk. Constraints: Provider: support optional; mutable																

Name	MachineConfiguration		
Type URI	http://schemas.dmtf.org/cimi/1/MachineConfiguration		
Attribute	Type	Description	
		Consumer: support optional; read-write Constraints: Provider: support optional; mutable Consumer: support optional; read-write	
cpuArch	string	Indicates the CPU architecture that is supported by Machines created by using this configuration. Allowable values include: 68000, Alpha, ARM, Itanium, MIPS, PA_RISC, POWER, PowerPC, x86, x86_64, z/Architecture, SPARC . Providers may define additional values. Constraints: Provider: support optional; mutable Consumer: support optional; read-write	
cpuSpeed	integer	The approximate CPU speed of this Machine in megahertz. Constraints: Provider: support optional; mutable Consumer: support optional; read-write	

3697 NOTE The disk attributes "format" does not appear on Machine Resources because after the Machine is
 3698 created, the user of the Machine is able modify this attribute of a disk, possibly without the Provider's knowledge.
 3699 Therefore these attributes might not be an aspect of the Machine that the Provider can reliably manage.

3700 **JSON media type:** application/json

3701 **JSON serialization:**

```

3702 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
3703   "id": string,
3704   "name": string, ?
3705   "description": string, ?
3706   "created": string, ?
3707   "updated": string, ?
3708   "properties": { string: string, + }, ?
3709   "cpu": number,
3710   "memory": number,
3711   "disks" : [
3712     { "capacity": number,
3713       "format": string,
3714       "initialLocation": string?
3715     }, +
3716   ], ?
3717   "cpuArch": string, ?
3718   "cpuSpeed": number, ?
3719   "operations": [
3720     { "rel": "edit", "href": string }, ?
3721     { "rel": "delete", "href": string } ?
3722   ] ?
3723   ...
3724 }
```

3725 **XML media type:** application/xml

3726 **XML serialization:**

```

3727 <MachineConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
3728   <id> xs:anyURI </id>
3729   <name> xs:string </name> ?
3730   <description> xs:string </description> ?
3731   <created> xs:dateTime </created> ?
3732   <updated> xs:dateTime </updated> ?
3733   <property key="xs:string"> xs:string </property> *
3734   <cpu> xs:integer </cpu>
3735   <memory> xs:integer </memory>
3736   <disk>
3737     <capacity> xs:integer </capacity>
3738     <format> xs:string </format>
3739     <initialLocation> xs:string </initialLocation> ?
3740   </disk> *
3741   <cpuArch> xs:string </cpuArch> ?
3742   <cpuSpeed> xs:integer </cpuSpeed> ?
3743   <operation rel="edit" href="xs:anyURI"/> ?
3744   <operation rel="delete" href="xs:anyURI"/> ?
3745   <xs:any>*
3746 </MachineConfiguration>

```

3747 5.14.5.1 Operations

3748 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 3749 MachineConfigurationCollection Resource.

3750 5.14.6 MachineConfigurationCollection Resource

3751 A MachineConfigurationCollection Resource represents the Collection of
 3752 MachineConfiguration Resources within a Provider and follows the Collection pattern defined in
 3753 clause 5.5.12. This Resource shall be serialized as follows:

3754 **JSON serialization:**

```

3755 { "resourceURI":
3756   "http://schemas.dmtf.org/cimi/1/MachineConfigurationCollection",
3757   "id": string,
3758   "count": number,
3759   "machineConfigurations": [
3760     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
3761       "id": string,
3762       ... remaining MachineConfiguration attributes ...
3763     }, +

```

```

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

```

XML serialization:

```

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

```

5.14.6.1 Operations

This Resource supports the Read and Update operations. Creation of new MachineConfiguration Resources is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.14.7 MachineImage Resource

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

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

Table 27 describes the MachineImage attributes.

Table 27 – MachineImage attributes

Name		
MachineImage		
Type URI		
http://schemas.dmtf.org/cimi/1/MachineImage		
Attribute	Type	Description
state	string	The operational state of the MachineImage. Allowable values include: CREATING : The MachineImage is in the process of being created. AVAILABLE : The MachineImage is available and ready for use. Unless otherwise

Name	MachineImage	
Type URI	http://schemas.dmtf.org/cimi/1/MachineImage	
Attribute	Type	Description
		specified, the MachineImage shall initially be in this state after successful creation. DELETING: The MachineImage is in the process of being deleted. ERROR: The Provider has detected an error in the MachineImage. The operations that result in transitions to the above defined states are defined in clause 5.14.7.1 Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only

IECNORM.COM : Click to view the full PDF of ISO/IEC 19831:2015

Name	MachineImage	
Type URI	http://schemas.dmtf.org/cimi/1/MachineImage	
Attribute	Type	Description
type	string	The type of MachineImage that is represented by this Resource. This specification defines the following values: IMAGE: This type represents the persisted data of a stopped Machine. Unlike "snapshots", it does not contain any runtime information. If this value is used, the "relatedImage" attribute shall not be present. SNAPSHOT: This type represents the persisted data of a Machine. If the Machine was not in a stopped state if this Image was created, it also contains runtime information. If this value is used, the "relatedImage" attribute shall reference the most recently created (or reverted to) snapshot Image for that Machine, which allows for easy discovery of the "previous" snapshot. The "relatedImage" attribute shall not be set by Consumers. PARTIAL_SNAPSHOT: This type follows the same semantics as the "SNAPSHOT" MachineImage except that it contains just the changes (deltas) made to the Machine based on the referenced "relatedImage" MachineImage rather than a complete representation of the Machine. If a MachineImage is deleted, the following semantics shall apply: - Any "SNAPSHOT" MachineImages that have a "relatedImage" value that references the deleted MachineImage shall have that value changed to the "relatedImage" attribute of the delete MachineImage. - Any "PARTIAL_SNAPSHOT" MachineImages that have a "relatedImage" value that references the deleted MachineImage shall also be deleted. This detail applies recursively to any subsequent "PARTIAL_SNAPSHOT" MachineImages as well. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
imageLocation	URI	A reference to the location of the binary data that makes up this image. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
relatedImage	ref	A reference to another MachineImage Resource that is related to this one. The specific meaning of this value varies depending on the type of MachineImage. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

3798 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

3799 **JSON media type:** application/json

3800 **JSON serialization:**

```
3801 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
3802   "id": string,
3803   "name": string, ?
3804   "description": string, ?
3805   "created": string, ?
3806   "updated": string, ?
3807   "properties": { string: string, + }, ?
3808   "state": string,
3809   "type": string,
3810   "imageLocation": string,
3811   "relatedImage": { "href": string }, ?
3812   "operations": [
3813     { "rel": "edit", "href": string }, ?
3814     { "rel": "delete", "href": string } ?
3815   ] ?
3816   ...
3817 }
```

3818 **XML media type:** application/xml

3819 **XML serialization:**

```
3820 <MachineImage xmlns="http://schemas.dmtf.org/cimi/1">
3821   <id> xs:anyURI </id>
3822   <name> xs:string </name> ?
3823   <description> xs:string </description> ?
3824   <created> xs:dateTime </created> ?
3825   <updated> xs:dateTime </updated> ?
3826   <property key="xs:string"> xs:string </property> *
3827   <state> xs:string </state>
3828   <type> xs:string </type>
3829   <imageLocation> xs:anyURI </imageLocation>
3830   <relatedImage href="xs:anyURI"/> ?
3831   <operation rel="edit" href="xs:anyURI"/> ?
3832   <operation rel="delete" href="xs:anyURI"/> ?
3833   <xs:any>*
3834 </MachineImage>
```

3835 **5.14.7.1 Operations**

3836 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 3837 MachineImageCollection Resource.

3838 If creating a new MachineImage, the representation of the new MachineImage may include a
 3839 reference in the "imageLocation" attribute. Providers shall inspect this reference (most likely by the way of
 3840 an HTTP HEAD) to determine if any special processing is required. This specification defines the
 3841 following additional steps that Providers shall take depending on the type of Resource being referenced:

3842 <http://schemas.dmtf.org/cimi/1/Machine>

3843 If the "imageLocation" is a reference to a Machine, the Provider shall create a new SNAPSHOT
 3844 MachineImage based on the Machine being referenced. Upon completion of the create operation,
 3845 the MachineImage's "imageLocation" attribute shall not reference the Machine (as the Machine
 3846 might change over time), but instead it shall reference (or contain the data of) the static representation of
 3847 the Machine. Additionally, the referenced Machine's MachineSnapshotCollection shall be
 3848 updated to include a reference to this newly created SNAPSHOT MachineImage Resource.

3849 **5.14.8 MachineImageCollection Resource**

3850 A MachineImageCollection Resource represents the Collection of MachineImage Resources
 3851 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
 3852 serialized as follows:

3853 **JSON serialization:**

```
3854 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImageCollection",
3855   "id": string,
3856   "count": number,
3857   "machineImages": [
3858     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
3859       "id": string,
3860       ... remaining MachineImage attributes ...
3861     }, +
3862   ], ?
3863   "operations": [ { "rel": "add", "href": string } ? ]
3864   ...
3865 }
```

3866 **XML serialization:**

```
3867 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/MachineImageCollection"
3868   xmlns="http://schemas.dmtf.org/cimi/1">
3869   <id> xs:anyURI </id>
3870   <count> xs:integer </count>
3871   <MachineImage>
3872     <id> xs:anyURI </id>
3873     ... remaining MachineImage attributes ...
3874   </MachineImage> *
```

```

3875     <operation rel="add" href="xs:anyURI"/> ?
3876     <xs:any>*
3877 </Collection>

```

3878 5.14.8.1 Operations

3879 This Resource supports the Read and Update operations. Creation of new MachineImage Resources
 3880 is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1, where the
 3881 request body and the way it is processed are described in clause 5.14.7.1.

3882 5.14.9 Credential Resource

3883 A Credential Resource contains the information required to create the initial administrative superuser
 3884 of a newly created Machine or to represent the credentials needed to perform some operation. Due to
 3885 the variation between operating systems and Providers, this specification does not mandate one
 3886 particular set of attributes that all implementations need to support. However, Providers are expected to
 3887 extend this Resource with additional attributes to meet their requirements.

3888 For example, a Provider might extend this Resource with username and password attributes, which would
 3889 then be the login information for new Machines. These extension attributes would appear as siblings to
 3890 the common attributes like "name" and "description."

3891 Table 28 describes the Credential attributes.

3892 **Table 28 – Credential attributes**

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

3893 Some common extension attributes that Providers might use include:

3894 **Table 29 – UserName/Password attributes**

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

3895 **Table 30 – Public key attributes**

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

3896

3897 When implementing or using Credential, Providers and Consumers shall adhere to the syntax and
 3898 semantics of its attributes as described in the above table, as well as in the table describing related

ISO/IEC 19831:2015 (E)

3899 Collections. Both Consumer and Provider shall serialize this Resource as described below. The following
3900 pseudo-schemas (see notation in 1.3)

3901 **JSON media type:** application/json

3902 **JSON serialization:**

```
3903 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Credential",  
3904   "id": string,  
3905   "name": string, ?  
3906   "description": string, ?  
3907   "created": string, ?  
3908   "updated": string, ?  
3909   "properties": { string: string, + }, ?  
3910   "operations": [  
3911     { "rel": "edit", "href": string }, ?  
3912     { "rel": "delete", "href": string } ?  
3913   ] ?  
3914   ...  
3915 }
```

3916 **XML media type:** application/xml

3917 **XML serialization:**

```
3918 <Credential xmlns="http://schemas.dmtf.org/cimi/1">  
3919   <id> xs:anyURI </id>  
3920   <name> xs:string </name> ?  
3921   <description> xs:string </description> ?  
3922   <created> xs:dateTime </created> ?  
3923   <updated> xs:dateTime </updated> ?  
3924   <property key="xs:string"> xs:string </property> *  
3925   <operation rel="edit" href="xs:anyURI"/> ?  
3926   <operation rel="delete" href="xs:anyURI"/> ?  
3927   <xs:any*  
3928 </Credential>
```

3929 5.14.9.1 Operations

3930 This Resource supports the Read, Update, and Delete operations. Create is supported through the
3931 CredentialCollection Resource.

3932 5.14.10 CredentialCollection Resource

3933 A CredentialCollection Resource represents the Collection of Credential Resources within
3934 a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized
3935 as follows:

JSON serialization:

```

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

```

XML serialization:

```

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

```

5.14.10.1 Operations

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

5.14.11 CredentialTemplate Resource

This Resource captures the configuration values for realizing a Credential Resource. A CredentialTemplate may be used to create multiple Credentials. Table 31 describes the CredentialTemplate attributes.

Table 31 – CredentialTemplate attributes

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

When implementing or using CredentialTemplate, Providers and Consumers shall adhere to the syntax and semantics of its attributes as described in the above table as well as in the table describing related Collections. Both Consumer and Provider shall serialize this Resource as described below. The

ISO/IEC 19831:2015 (E)

3971 following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON
3972 and XML:

3973

3974 **JSON media type:** application/json

3975 **JSON serialization:**

```
3976 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialTemplate",  
3977   "id": string,  
3978   "name": string, ?  
3979   "description": string, ?  
3980   "created": string, ?  
3981   "updated": string, ?  
3982   "properties": { string: string, + }, ?  
3983   "operations": [  
3984     { "rel": "edit", "href": string }, ?  
3985     { "rel": "delete", "href": string } ?  
3986   ] ?  
3987   ...  
3988 }
```

3989 **XML media type:** application/xml

3990 **XML serialization:**

```
3991 <CredentialTemplate xmlns="http://schemas.dmtf.org/cimi/1">  
3992   <id> xs:anyURI </id>  
3993   <name> xs:string </name> ?  
3994   <description> xs:string </description> ?  
3995   <created> xs:dateTime </created> ?  
3996   <updated> xs:dateTime </updated> ?  
3997   <property key="xs:string"> xs:string </property> *  
3998   <operation rel="edit" href="xs:anyURI"/> ?  
3999   <operation rel="delete" href="xs:anyURI"/> ?  
4000   <xs:any> *  
4001 </CredentialTemplate>
```

4002 5.14.11.1 Operations

4003 This Resource supports the Read, Update, and Delete operations. Create is supported through the
4004 CredentialTemplateCollection Resource.

4005 5.14.12 CredentialTemplateCollection Resource

4006 A CredentialTemplateCollection Resource represents the Collection of
4007 CredentialTemplate Resources within a Provider and follows the Collection pattern defined in
4008 clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

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

IECNORM.COM : Click to view the full PDF of ISO/IEC 19831:2015

XML serialization:

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

5.14.12.1 Operations

This Resource supports the Read and Update operations. Creation of new CredentialTemplate Resources is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.15 Volume Resources and relationships

Figure 4 illustrates the Resources involved in constructing a Volume and their relationships. Although this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor normative.

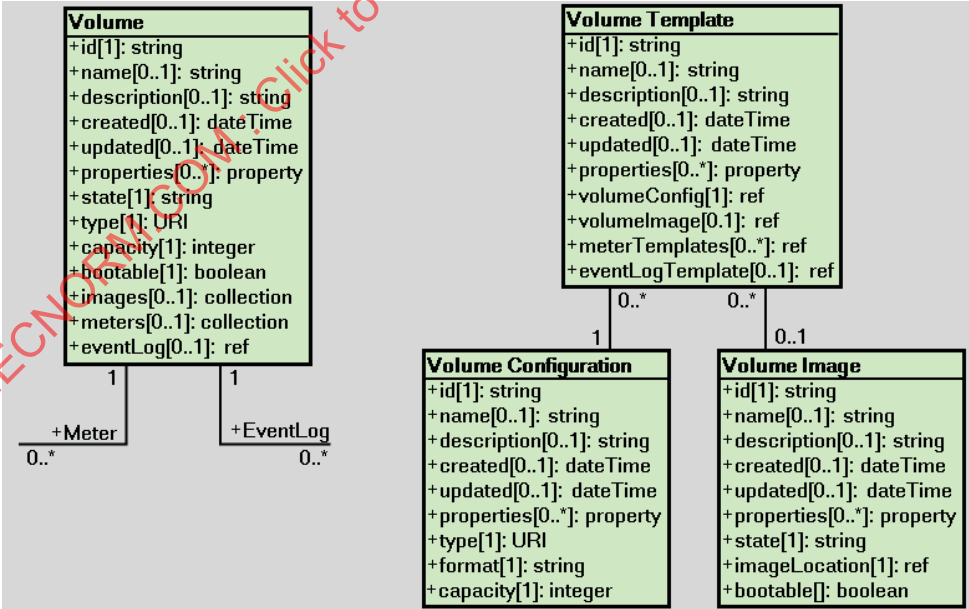


Figure 4 - Volume Resources

4046 **5.15.1 Volume**

4047 A Volume represents storage at either the block or the file-system level. Volumes can be connected to
 4048 Machines. Once connected, Volumes can be accessed by processes on that Machine. Table 32
 4049 describes the Volume attributes.

4050 **Table 32 – Volume attributes**

Name	Volume	
Type URI	http://schemas.dmtf.org/cimi/1/Volume	
Attribute	Type	Description
state	string	The operational state of the Volume. Allowable values include: CREATING: The Volume is in the process of being created. AVAILABLE: The Volume is available and ready for use. Unless otherwise specified, the Volume shall be in this state initially after successful creation. CAPTURING: The Volume is in the process of being captured (snapshotted) into a new VolumeImage. Allowable action if in this state is: delete. DELETING: The Volume is in the process of being deleted. ERROR: The Provider has detected an error in the Volume. The operations that result in transitions to the above defined states are defined in clause 5.15.1.2 Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
type	URI	A URI that indicates the type of Volume to be created. This specification defines the following URI: http://schemas.dmtf.org/cimi/1/mapped: Indicates a Volume that shall be used for shared storage that might be available to multiple Machines, but which does not require an explicit mount operation from within the guest operating system. Additional values may be defined. If certain types of Volumes require additional data, it is expected that this Resource is extended. For example, a "sharedFileSystem" type might require additional networking information and credentials to be specified. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
capacity	integer	The maximum size, if limited, of the Volume in kilobytes. If this value is increased, the Volume can contain more data. Decreasing this value may require evaluations. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
bootable	boolean	This property indicates whether this Volume is bootable. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
images	collection [Volume Volume Image]	A reference to the list of references to VolumeImages that represent snapshots taken from the Volume. Note: The VolumeVolumeImage Resource type represents an association between the Volume and a VolumeImage. It is defined in clause 5.15.1.1.1. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meters	collection [Meter]	A reference to the list of Meters monitored for this Volume. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLog	ref	A reference to the EventLog of this Volume. Constraints: Provider: support optional; mutable

Name	Volume	
Type URI	http://schemas.dmtf.org/cimi/1/Volume	
Attribute	Type	Description
		Consumer: support optional; read-only

4051 When implementing or using Volume, Providers and Consumers shall adhere to the syntax and
 4052 semantics of its attributes as described in the above table as well as in the tables describing embedded
 4053 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 4054 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 4055 both JSON and XML:

4056

4057 **JSON media type:** application/json

4058 **JSON serialization:**

```

4059 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Volume",
4060   "id": string,
4061   "name": string, ?
4062   "description": string, ?
4063   "created": string, ?
4064   "updated": string, ?
4065   "properties": { string: string, + }, ?
4066   "state": string,
4067   "type": string,
4068   "capacity": number,
4069   "bootable": boolean,
4070   "images": { "href": string }, ?
4071   "meters": { "href": string }, ?
4072   "eventLog": { "href": string }, ?
4073   "operations": [
4074     { "rel": "edit", "href": string }, ?
4075     { "rel": "delete", "href": string } ?
4076   ] ?
4077   ...
4078 }
```

4079 **XML media type:** application/xml

4080 **XML serialization:**

```

4081 <Volume xmlns="http://schemas.dmtf.org/cimi/1">
4082   <id> xs:anyURI </id>
4083   <name> xs:string </name> ?
4084   <description> xs:string </description> ?
4085   <created> xs:dateTime </created> ?
4086   <updated> xs:dateTime </updated> ?
```

```

4087 <property key="xs:string"> xs:string </property> *
4088 <state> xs:string </state>
4089 <type> xs:anyURI </type>
4090 <capacity> xs:integer </capacity>
4091 <bootable> xs:boolean </bootable>
4092 <images href="xs:anyURI"/> ?
4093 <meters href="xs:anyURI"/> ?
4094 <eventLog href="xs:anyURI"/> ?
4095 <operation rel="edit" href="xs:anyURI"/> ?
4096 <operation rel="delete" href="xs:anyURI"/> ?
4097 <xs:any>*
4098 </Volume>

```

5.15.1.1 Collections

The following clauses describe the Collection Resources owned by Volumes.

5.15.1.1.1 VolumeVolumeImageCollection Resource

The Resource type for each item of this Collection is "VolumeVolumeImage", defined in Table 33:

Table 33 – VolumeVolumeImage attributes

Name	VolumeVolumeImage	
Type URI	http://schemas.dmtf.org/cimi/1/VolumeVolumeImage	
Attribute	Type	Description
volumeImage	ref	Reference to a VolumeImage Resource, which represents a snapshot of this Volume. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only

JSON serialization:

```

4105 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeVolumeImageCollection",
4106   "id": string,
4107   "count": number,
4108   "volumeVolumeImages": [
4109     { "resourceURI":
4110       "http://schemas.dmtf.org/cimi/1/VolumeVolumeImage",
4111       "id": string,
4112       "name": string, ?
4113       "description": string, ?
4114       "created": string, ?
4115       "updated": string, ?
4116       "properties": { string: string, + }, ?
4117       "volumeImage": { "href": string },
4118       "operations": [
4119         { "rel": "edit", "href": string }, ?

```

```

4120         { "rel": "delete", "href": string } ?
4121     ] ?
4122     ...
4123 }, +
4124 ] ?
4125     ...
4126 }

```

XML serialization:

```

4127 <Collection
4128   resourceURI="http://schemas.dmtf.org/cimi/1/VolumeVolumeImageCollection"
4129   xmlns="http://schemas.dmtf.org/cimi/1">
4130   <id> xs:anyURI </id>
4131   <count> xs:integer </count>
4132   <VolumeVolumeImage>
4133     <id> xs:anyURI </id>
4134     <name> xs:string </name> ?
4135     <description> xs:string </description> ?
4136     <created> xs:dateTime </created> ?
4137     <updated> xs:dateTime </updated> ?
4138     <property key="xs:string"> xs:string </property> *
4139     <volumeImage href="xs:anyURI"/>
4140     <operation rel="edit" href="xs:anyURI"/> ?
4141     <operation rel="delete" href="xs:anyURI"/> ?
4142     <xs:any>*
4143   </VolumeVolumeImage> *
4144   <xs:any>*
4145 </Collection>

```

NOTE Previous versions of this specification included an "add" operation on this Resource. It is now deprecated in favor of creating a new VolumeImage with the imageLocation attribute pointing to the Volume to be captured.

5.15.1.1.2 VolumeMeterCollection Resource

The Resource type for each item of this Collection is "Meter" as defined in clause 5.17.3.

JSON serialization:

```

4152 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeMeterCollection",
4153   "id": string,
4154   "count": number,
4155   "meters": [
4156     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
4157       "id": string,
4158       ... remaining Meter attributes ...

```

```

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

```

XML serialization:

```

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

```

5.15.1.2 Operations

This Resource supports the Read, Update, and Delete operations. Create is supported through the VolumeCollection Resource.

5.15.2 VolumeCollection Resource

A VolumeCollection Resource represents the Collection of Volumes within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

```

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

```

XML serialization:

```

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

```

```

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

```

5.15.2.1 Operations

NOTE The "add" operation requires that a VolumeTemplate be used (see 4.2.1.1).

5.15.3 VolumeTemplate Resource

This Resource captures the configuration values for realizing a Volume. A VolumeTemplate may be used to create multiple Volumes. Table 34 describes the VolumeTemplate attributes.

Table 34 – VolumeTemplate attributes

Name	VolumeTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/VolumeTemplate	
Attribute	Type	Description
volumeConfig	ref	A reference to the VolumeConfiguration that is used to create a Volume from this VolumeTemplate. Note that the attributes of the VolumeConfiguration may be specified rather than a reference to an existing VolumeConfiguration Resource. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
volumeImage	ref	A reference to the VolumeImage that is used to create a Volume from this VolumeTemplate. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
meterTemplates	MeterTemplates[]	A list of references to MeterTemplates that shall be used to create and connect a set of new Meters to the new Volume. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
eventLogTemplate	ref	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new Volume. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

When implementing or using VolumeTemplate, Providers and Consumers shall adhere to the syntax and semantics of its attributes as described in the above table as well as in the tables describing

embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "volumeConfig": {
    "href": string | ... VolumeConfiguration attributes ...
  },
  "volumeImage": { "href": string }, ?
  "meterTemplates": [
    { "href": string, ?
      ... MeterTemplate attributes ... ?
    }, *
  ], ?
  "eventLogTemplate": {
    "href": string, ?
    ... EventLogTemplate attributes ... ?
  }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<VolumeTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
```

```

4254 <updated> xs:dateTime </updated> ?
4255 <property key="xs:string"> xs:string </property> *
4256 <volumeConfig href="xs:anyURI">
4257   ... VolumeConfiguration attributes ... ?
4258 </volumeConfig>
4259 <volumeImage href="xs:anyURI"/> ?
4260 <meterTemplate href="xs:anyURI"? >
4261   ... MeterTemplate attributes ... ?
4262 </meterTemplate> *
4263 <eventLogTemplate href="xs:anyURI"? >
4264   ... EventLogTemplate attributes ... ?
4265 </eventLogTemplate> ?
4266 <operation rel="edit" href="xs:anyURI"/> ?
4267 <operation rel="delete" href="xs:anyURI"/> ?
4268 <xs:any>*
4269 </VolumeTemplate>

```

4270 5.15.3.1 Operations

4271 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 4272 VolumeTemplateCollection Resource.

4273 5.15.4 VolumeTemplateCollection Resource

4274 A VolumeTemplateCollection Resource represents the Collection of VolumeTemplate
 4275 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
 4276 shall be serialized as follows:

JSON serialization:

```

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeTemplateCollection",
  "id": string,
  "count": number,
  "volumeTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeTemplate",
      "id": string,
      ... remaining volumeTemplate attributes ...
    }, +
  ], ?
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}

```

XML serialization:

```

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

```

5.15.4.1 Operations

This Resource supports the Read and Update operations. Creation of new VolumeTemplate Resources is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.15.5 VolumeConfiguration Resource

The VolumeConfiguration Resource represents the set of configuration values needed to create a Volume with certain characteristics. VolumeConfigurations are created by Providers and may, at the Providers discretion, be created by Consumers.

Table 35 describes the VolumeConfiguration attributes.

4312

Table 35 – VolumeConfiguration attributes

Name	VolumeConfiguration	
Type URI	http://schemas.dmtf.org/cimi/1/VolumeConfiguration	
Attribute	Type	Description
type	URI	A URI that indicates the type of Volume to be created. This specification defines the following URI: http://schemas.dmtf.org/cimi/1/mapped: Indicates a Volume that shall be used for shared storage that might be available to multiple Machines, but which does not require an explicit mount operation from within the guest operating system. Additional values may be defined. If certain types of Volumes require additional data, it is expected that this Resource is extended. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
format	string	The format of the file system that is placed on Volumes created from this configuration. This attribute is only meaningful for VolumeConfigurations that describe block devices. This attribute is optional; the absence of this attribute indicates that Volumes created from this configuration are not formatted with a file system. Example values: "ext4," "ntfs." Constraints: Provider: support optional; mutable Consumer: support optional; read-write
capacity	integer	The default size in kilobytes, if limited, of the Volume created from this VolumeConfiguration. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

4313 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

4314 **JSON media type:** application/json

4315 **JSON serialization:**

```

4316 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
4317   "id": string,
4318   "name": string, ?
4319   "description": string, ?
4320   "created": string, ?
4321   "updated": string, ?
4322   "properties": { string: string, + }, ?
4323   "type": string,
4324   "format": string,
4325   "capacity": number,
4326   "operations": [
4327     { "rel": "edit", "href": string }, ?
4328     { "rel": "delete", "href": string } ?
4329   ] ?
4330   ...
4331 }
```

4332 **XML media type:** application/xml

4333 **XML serialization:**

```

4334 <VolumeConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
4335   <id> xs:anyURI </id>
4336   <name> xs:string </name> ?
4337   <description> xs:string </description> ?
4338   <created> xs:dateTime </created> ?
4339   <updated> xs:dateTime </updated> ?
4340   <property key="xs:string"> xs:string </property> *
4341   <type> xs:anyURI </type>
4342   <format> xs:string </format>
4343   <capacity> xs:integer </capacity>
4344   <operation rel="edit" href="xs:anyURI"/> ?
4345   <operation rel="delete" href="xs:anyURI"/> ?
4346   <xs:any>*
4347 </VolumeConfiguration>

```

4348 5.15.5.1 Operations

4349 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 4350 VolumeConfigurationCollection Resource.

4351 5.15.6 VolumeConfigurationCollection Resource

4352 A VolumeConfigurationCollection Resource represents the Collection of
 4353 VolumeConfiguration Resources within a Provider and follows the Collection pattern defined in
 4354 clause 5.5.12. This Resource shall be serialized as follows:

4355 **JSON serialization:**

```

4356 { "resourceURI":
4357   "http://schemas.dmtf.org/cimi/1/VolumeConfigurationCollection",
4358   "id": string,
4359   "count": number,
4360   "volumeConfigurations": [
4361     { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
4362       "id": string,
4363       ... remaining VolumeConfiguration attributes ...
4364     }, +
4365   ], ?
4366   "operations": [ { "rel": "add", "href": string } ? ]
4367   ...
4368 }

```

4369 XML serialization:

```

4370 <Collection
4371     resourceURI="http://schemas.dmtf.org/cimi/1/VolumeConfigurationCollection"
4372     xmlns="http://schemas.dmtf.org/cimi/1">
4373     <id> xs:anyURI </id>
4374     <count> xs:integer </count>
4375     <VolumeConfiguration>
4376         <id> xs:anyURI </id>
4377         ... remaining VolumeConfiguration attributes ...
4378     </VolumeConfiguration> *
4379     <operation rel="add" href="xs:anyURI"/> ?
4380     <xs:any>*
4381 </Collection>

```

4382 5.15.6.1 Operations

4383 This Resource supports the Read and Update operations. Creation of new VoLumeImage Resources is
 4384 supported by the way of a POST to the "add" operations' URI as described in clause 4.2.1.1.

4385 5.15.7 VolumelImage Resource

4386 This Resource represents an image that could be placed on a pre-loaded volume. Table 36 describes the
 4387 VoLumeImage attributes.

4388 Table 36 – VolumelImage attributes

Name		
Type URI		
Attribute		
Type		
Description		
state	string	The operational state of the VoLumeImage. Allowable values include: CREATING: The VoLumeImage is in the process of being created. AVAILABLE: The VoLumeImage is available and ready for use. Unless otherwise specified, the VoLumeImage shall initially be in this state after successful creation. DELETING: The VoLumeImage is in the process of being deleted. ERROR: The Provider has detected an error in the VoLumeImage. The operations that result in transitions to the above defined states are defined in clause 5.15.7.1 <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-only
imageLocation	ref	A reference to the location of the binary data that makes up this image. <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-write
bootable	boolean	This property indicates whether Volumes created from this VoLumeImage are bootable. <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-write

4389 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

4390 **JSON media type:** application/json

4391 **JSON serialization:**

```

4392 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeImage",
4393   "id": string,
4394   "name": string, ?
4395   "description": string, ?
4396   "created": string, ?
4397   "updated": string, ?
4398   "properties": { string: string, + }, ?
4399   "state": string,
4400   "imageLocation": { "href": string },
4401   "bootable": boolean,
4402   "operations": [
4403     { "rel": "edit", "href": string }, ?
4404     { "rel": "delete", "href": string } ?
4405   ] ?
4406   ...
4407 }
```

4408 **XML media type:** application/xml

4409 **XML serialization:**

```

4410 <VolumeImage xmlns="http://schemas.dmtf.org/cimi/1">
4411   <id> xs:anyURI </id>
4412   <name> xs:string </name> ?
4413   <description> xs:string </description> ?
4414   <created> xs:dateTime </created> ?
4415   <updated> xs:dateTime </updated> ?
4416   <property key="xs:string"> xs:string </property> *
4417   <state> xs:string </state>
4418   <imageLocation href="xs:anyURI"/>
4419   <bootable> xs:boolean </bootable>
4420   <operation rel="edit" href="xs:anyURI"/> ?
4421   <operation rel="delete" href="xs:anyURI"/> ?
4422   <xs:any>*
4423 </VolumeImage>
```

4424 5.15.7.1 Operations

4425 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 4426 VolumeImageCollection Resource.

5.15.8 VolumeImageCollection Resource

A VolumeImageCollection Resource represents the Collection of VolumeImage Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

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

XML serialization:

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

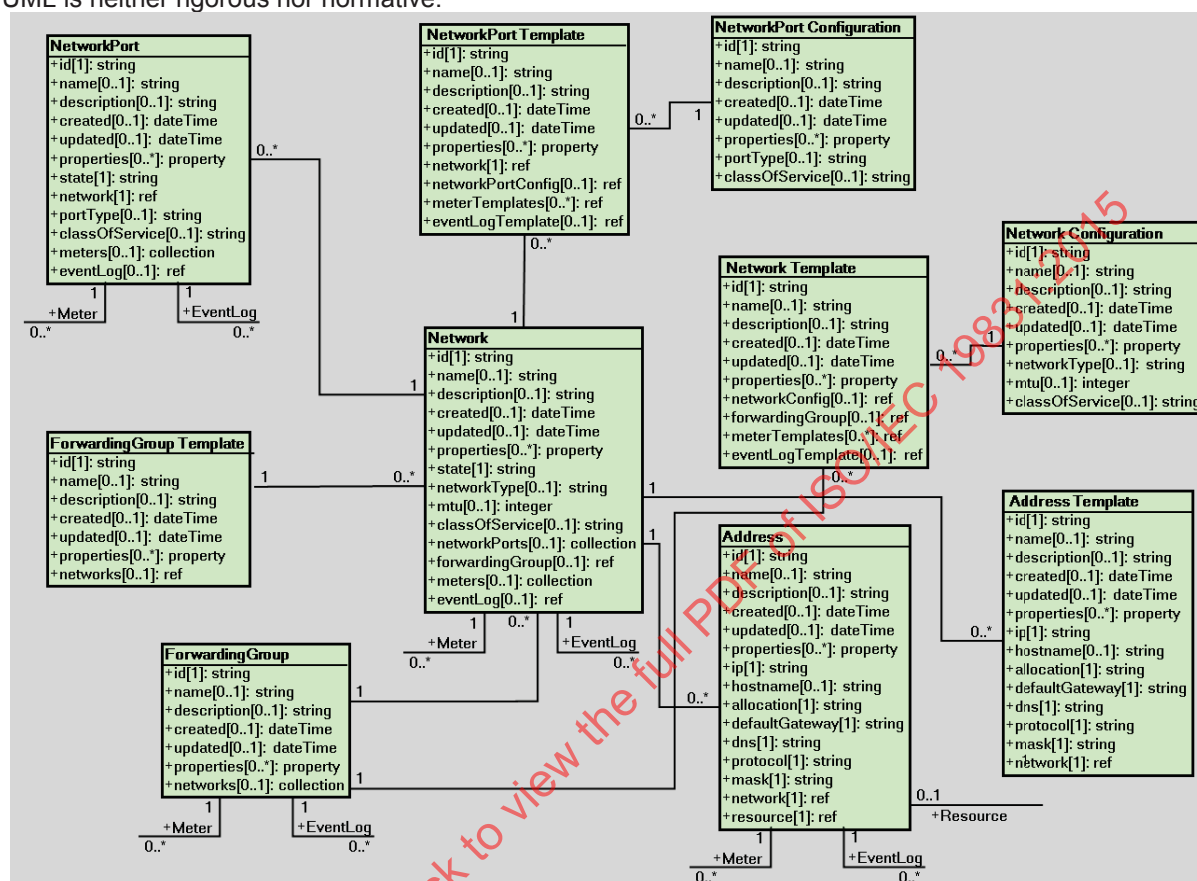
5.15.8.1 Operations

This Resource supports the Read and Update operations. Creation of new VolumeImage Resources is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.

During the creation of a new VolumeImage Resource, if the "imageLocation" attribute refers to an existing Volume, this operation shall be interpreted as a request to create a snapshot of the Volume. Once completed, the "imageLocation" attribute of the new VolumeImage Resource shall not refer to the original Volume; instead it shall refer to a static copy of the Volume. Additionally, the referenced Volume's VolumeVolumeImageCollection shall be updated to include a reference to this newly created snapshot VolumeImage Resource. During this process, the Provider may put the Volume into a "CAPTURING" state if necessary.

4466 5.16 Network Resources and relationships

4467 Figure 5 illustrates the Resources involved in constructing Networks and their NetworkPorts and
 4468 their relationships. Although this drawing is in the style of a Resource Relationship diagram, the use of
 4469 UML is neither rigorous nor normative.



4470 Figure 5 - Network Resources

4471 5.16.1 Network

4472 A Network is a Collection of interconnected logical services with the purpose of forwarding data traffic
 4473 between end points.

4474 Networks in a ForwardingGroup should all have the same "networkType" attributes, which
 4475 prevents a Network with a "private" access attribute from being publicly forwarded because it is a
 4476 member of a ForwardingGroup that also contains Networks with a "public" access attribute.

4477 Table 37 describes the Network attributes.

4478 Table 37 – Network attributes

Name	Network	
Type URI	http://schemas.dmtf.org/cimi/1/Network	
Attribute	Type	Description
state	string	The operational state of the Network. Allowable values include:

Name	Network	
Type URI	http://schemas.dmtf.org/cimi/1/Network	
Attribute	Type	Description
		CREATING: The Network is in the process of being created. STARTING: The Network is in the process of being started. STARTED: The Network is available and ready for use. STOPPING: The Network is in the process of being stopped. STOPPED: The Network is stopped and not available for use. DELETING: The Network is in the process of being deleted. ERROR: The Provider has detected an error in the Network. <u>The operations that result in transitions to the above defined states are defined in clause 5.16.1.2. clause 5.16.2.1 defines the initial state of a Network.</u> Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
networkType	string	An indicator of whether the Machine Resource has access to a Public or Private Network. Allowable values include: PUBLIC: represents an open and Internet routable network. PRIVATE: identifies a local non-routed network. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write
mtu	integer	(Maximum Transmission Unit) The largest Packet size supported on this Network. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
classOfService	string	The Provider's supported category associated with a Collection of attributes characterizing a level of a quality experience. Example values: GOLD: High bandwidth, low latency, low jitter SILVER: An improved service experience over bronze for voice or video traffic BRONZE: Best effort The list of possible values, and their implied quality of service, is out of scope of this specification. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
networkPorts	collection [Network Network Port]	A reference to the list of NetworkPorts that are associated with this Network. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
forwardingGroup	ref	A reference to a ForwardingGroup of which this Network is a part. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meters	collection [Meter]	A reference to the list of Meters monitored for this Network. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLog	ref	A reference to the EventLog of this Network. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

4479 When implementing or using Network, Providers and Consumers shall adhere to the syntax and
4480 semantics of its attributes as described in the above table as well as in the tables describing embedded
4481 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described

below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Network",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "networkType": string, ?
  "mtu": number, ?
  "classOfService": string, ?
  "networkPorts": { "href": string }, ?
  "forwardingGroup": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<Network xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <networkType> xs:string </networkType> ?
```

```

4521 <mtu> xs:integer </mtu> ?
4522 <classOfService> xs:string </classOfService> ?
4523 <networkPorts href="xs:anyURI"/> ?
4524 <forwardingGroup href="xs:anyURI"/> ?
4525 <meters href="xs:anyURI"/> ?
4526 <eventLog href="xs:anyURI"/> ?
4527 <operation rel="edit" href="xs:anyURI"/> ?
4528 <operation rel="delete" href="xs:anyURI"/> ?
4529 <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
4530 href="xs:anyURI"/> ?
4531 <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
4532 href="xs:anyURI"/> ?
4533 <xs:any>*
4534 </Network>

```

5.16.1.1 Collections

The following clauses describe the Collection Resources owned by Networks.

5.16.1.1.1 NetworkNetworkPortCollection Resource

If NetworkPorts are created through a Network's NetworkPortCollection's "add" operation, they shall be added to the global (Cloud Entry Point) NetworkPortCollection as well.

As specified in clause 5.5.12, if a Network is deleted, all of its Collections, and Resources in those Collections, shall also be deleted. This means that all of the NetworkPorts related to that Network shall also be deleted.

The Resource type for each item of this Collection is "NetworkPort" as defined in clause 5.16.7.

JSON serialization:

```

4545 { "resourceURI":
4546     "http://schemas.dmtf.org/cimi/1/NetworkNetworkPortCollection",
4547     "id": string,
4548     "count": number,
4549     "networkNetworkPorts": [
4550         { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkNetworkPort",
4551           "id": string,
4552           "name": string, ?
4553           "description": string, ?
4554           "created": string, ?
4555           "updated": string, ?
4556           "properties": { string: string, + }, ?
4557           "networkPort": { "href": string },
4558           "operations": [
4559             { "rel": "edit", "href": string }, ?

```

```

4560         { "rel": "delete", "href": string } ?
4561     ] ?
4562     ...
4563 }, +
4564 ] ?
4565     ...
4566 }

```

XML serialization:

```

4567 <Collection
4568     resourceURI="http://schemas.dmtf.org/cimi/1/NetworkNetworkPortCollection"
4569     xmlns="http://schemas.dmtf.org/cimi/1">
4570     <id> xs:anyURI </id>
4571     <count> xs:integer </count>
4572     <NetworkNetworkPort>
4573         <id> xs:anyURI </id>
4574         <name> xs:string </name> ?
4575         <description> xs:string </description> ?
4576         <created> xs:dateTime </created> ?
4577         <updated> xs:dateTime </updated> ?
4578         <property key="xs:string"> xs:string </property> *
4579         <networkPort href="xs:anyURI"/>
4580         <operation rel="edit" href="xs:anyURI"/> ?
4581         <operation rel="delete" href="xs:anyURI"/> ?
4582         <xs:any>*
4583     </NetworkNetworkPort> *
4584     <xs:any>*
4585 </Collection>

```

5.16.1.1.2 NetworkMeterCollection Resource

The Resource type for each item of this Collection is "Meter" as defined in clause 5.17.3.

JSON serialization:

```

4590 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkMeterCollection",
4591   "id": string,
4592   "count": number,
4593   "meters": [
4594       { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
4595         "id": string,
4596         ... remaining Meter attributes ...
4597       }, +
4598   ], ?

```

```

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

```

XML serialization:

```

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

```

5.16.1.2 Operations

This Resource supports the Read, Update, and Delete operations. Create is supported through the NetworkCollection Resource.

The following custom operations are also defined:

start

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

This operation shall start a Network.

Input parameters: None.

Output parameters: None.

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

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

HTTP protocol

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

JSON media type: application/json

JSON serialization:

```

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

```

4635
4636

}

IECNORM.COM : Click to view the full PDF of ISO/IEC 19831:2015

4637 **XML media type:** application/xml

4638 **XML serialization**

```
4639 <Action xmlns="http://schemas.dmtf.org/cimi/1">
4640   <action> http://schemas.dmtf.org/cimi/1/action/start </action>
4641   <property key="xs:string"> xs:string </property> *
4642   <xs:any>*
4643 </Action>
```

4644 Upon successful processing of the request, the HTTP response body may be empty.

4645 **stop**

4646 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/stop

4647 This operation shall stop a Network. If stopped, a Network shall not allow data to flow through it.

4648 Input parameters: None.

4649 Output parameters: None.

4650 During the processing of this operation, the Network shall be in the "STOPPING" state.

4651 Upon successful completion of this operation, the Network shall be in the "STOPPED" state.

4652 **HTTP protocol**

4653 To stop a Network, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the
4654 Network where the HTTP request body shall be as described below.

4655 **JSON media type:** application/json

4656 **JSON serialization:**

```
4657 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
4658   "action": "http://schemas.dmtf.org/cimi/1/action/stop",
4659   "properties": { string: string, + } ?
4660   ...
4661 }
```

4662 **XML media type:** application/xml

4663 **XML serialization**

```
4664 <Action xmlns="http://schemas.dmtf.org/cimi/1">
4665   <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
4666   <property key="xs:string"> xs:string </property> *
4667   <xs:any>*
4668 </Action>
```

4669 Upon successful processing of the request, the HTTP response body may be empty.

5.16.2 NetworkCollection Resource

A NetworkCollection Resource represents the Collection of Networks within a Provider and follows the Collection pattern that is defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

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

XML serialization:

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

5.16.2.1 Operations

NOTE The "add" operation requires that a NetworkTemplate be used (see 4.2.1.1).

Upon successful processing of the "add" operation, unless otherwise specified by the way of the NetworkTemplate "initialState" attribute, the state of the new Network shall be the value of the DefaultInitialState capability of the Network Resource's ResourceMetadata, if defined. If no DefaultInitialState capability is defined, the default value shall be "STOPPED." The semantics of "initialState" shall be equivalent to the Provider issuing the appropriate actions against the new Network to move it into that state.

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

4709 **5.16.3 NetworkTemplate Resource**

4710 The NetworkTemplate is a set of configuration values for realizing a Network. An instance of
 4711 NetworkTemplate may be used to create multiple Networks. Table 38 describes the
 4712 NetworkTemplate attributes.

4713 **Table 38 – NetworkTemplate attributes**

Name	NetworkTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkTemplate	
Attribute	Type	Description
initialState	string	The initial state of the new Network. Possible values include the non-transient states as specified by the Network "state" attribute (i.e., STARTED, STOPPED) and shall be determined by the actions supported by the Provider. Providers should advertise the list of available values by the way of the Network ResourceMetadata "initialStates" capability. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
networkConfig	ref	A reference to the NetworkConfiguration that is used to create a Network from this NetworkTemplate. Note that the attributes of the NetworkConfiguration may be specified rather than a reference to an existing NetworkConfiguration Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
forwardingGroup	ref	A reference to a ForwardingGroup of which this Network is a part. Note that Networks forward to themselves; therefore, this attribute only appears in cases where the Network that is created from this Template forwards to one or more additional Networks. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
meterTemplates	meterTemplates[]	A list of references to MeterTemplates that shall be used to create and connect a set of new Meters to the new Network. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
eventLogTemplate	ref	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new Network. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

4714 When implementing or using NetworkTemplate, Providers and Consumers shall adhere to the syntax
 4715 and semantics of its attributes as described in the above table as well as in the tables describing
 4716 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
 4717 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 4718 Resource in both JSON and XML

4719 **JSON media type:** application/json

4720 **JSON serialization:**

4721 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkTemplate",

```

4722   "id": string,
4723   "name": string, ?
4724   "description": string, ?
4725   "created": string, ?
4726   "updated": string, ?
4727   "properties": { string: string, + }, ?
4728   "initialState": string, ?
4729   "networkConfig": {
4730     "href": string |... NetworkingConfiguration attributes ...
4731   }, ?
4732   "forwardingGroup": { "href": string }, ?
4733   "meterTemplates": [
4734     { "href": string, ?
4735       ... MeterTemplate attributes ... ?
4736     }, *
4737   ], ?
4738   "eventLogTemplate": {
4739     "href": string, ?
4740     ... EventLogTemplate attributes ... ?
4741   }, ?
4742   "operations": [
4743     { "rel": "edit", "href": string }, ?
4744     { "rel": "delete", "href": string } ?
4745   ] ?
4746   ...
4747 }

```

4748 **XML media type:** application/xml

4749 **XML serialization:**

```

4750 <NetworkTemplate xmlns="http://schemas.dmtf.org/cimi/1">
4751   <id> xs:anyURI </id>
4752   <name> xs:string </name> ?
4753   <description> xs:string </description> ?
4754   <created> xs:dateTime </created> ?
4755   <updated> xs:dateTime </updated> ?
4756   <property key="xs:string"> xs:string </property> *
4757   <initialState> xs:string </initialState> ?
4758   <networkConfig href="xs:anyURI"?>
4759     ... NetworkingConfiguration attributes ... ?
4760   </networkConfig> ?

```

```

4761     <forwardingGroup href="xs:anyURI"/> ?
4762     <meterTemplate href="xs:anyURI"? >
4763         ... MeterTemplate attributes ... ?
4764     </meterTemplate> *
4765     <eventLogTemplate href="xs:anyURI"? >
4766         ... EventLogTemplate attributes ... ?
4767     </eventLogTemplate> ?
4768     <operation rel="edit" href="xs:anyURI"/> ?
4769     <operation rel="delete" href="xs:anyURI"/> ?
4770     <xs:any>*
4771 </NetworkTemplate>

```

4772 5.16.3.1 Operations

4773 This Resource supports the Read, Update and Delete operations. Create is supported through the
4774 NetworkTemplateCollection Resource.

4775 5.16.4 NetworkTemplateCollection Resource

4776 A NetworkTemplateCollection Resource represents the Collection of NetworkTemplates
4777 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
4778 serialized as follows:

4779 JSON serialization:

```

4780 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkTemplateCollection",
4781   "id": string,
4782   "count": number,
4783   "networkTemplates": [
4784     { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkTemplate",
4785       "id": string,
4786       ... remaining NetworkTemplate attributes ...
4787     }, +
4788   ], ?
4789   "operations": [ { "rel": "add", "href": string } ? ]
4790   ...
4791 }

```

4792 XML serialization:

```

4793 <Collection
4794     resourceURI="http://schemas.dmtf.org/cimi/1/NetworkTemplateCollection"
4795     xmlns="http://schemas.dmtf.org/cimi/1">
4796     <id> xs:anyURI </id>
4797     <count> xs:integer </count>
4798     <NetworkTemplate>
4799         <id> xs:anyURI </id>

```

```

4800     ... remaining NetworkTemplate attributes ...
4801     </NetworkTemplate> *
4802     <operation rel="add" href="xs:anyURI"/> ?
4803     <xs:any>*
4804 </Collection>

```

4805 5.16.4.1 Operations

4806 This Resource supports the Read and Update operations. Creation of new NetworkTemplate
 4807 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 4808 4.2.1.1.

4809 5.16.5 NetworkConfiguration Resource

4810 The following set of configuration values (shown in Table 39) represent the information needed to create
 4811 a Network with certain characteristics.

4812 **Table 39 – NetworkConfiguration attributes**

Name		
Type URI		
http://schemas.dmtf.org/cimi/1/NetworkConfiguration		
Attribute	Type	Description
networkType	string	An indicator of whether the Network is a Public or Private Network. Allowable values include: PUBLIC : represents an open and Internet routable network. PRIVATE : identifies a local non-Internet network. <u>Constraints:</u> Provider : support optional; mutable Consumer : support optional; read-write
mtu	integer	(Maximum Transmission Unit) The largest supported packet size. <u>Constraints:</u> Provider : support optional; mutable Consumer : support optional; read-write
classOfService	string	The Provider's supported category associated with a Collection of attributes characterizing a level of a quality experience. Example values: GOLD : High bandwidth, low latency, low jitter SILVER : An improved service experience over bronze for voice or video traffic BRONZE : Best effort The list of possible values, and their implied quality of service, is out of scope of this specification. <u>Constraints:</u> Provider : support optional; mutable Consumer : support optional; read-write

4813 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

4814 **JSON media type:** application/json

4815 **JSON serialization:**

```

4816 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkConfiguration",
4817   "id": string,
4818   "name": string, ?
4819   "description": string, ?
4820   "created": string, ?

```

```

"updated": string, ?
"properties": { string: string, + }, ?
"networkType": string, ?
"mtu": number, ?
"classOfService": string, ?
"operations": [
  { "rel": "edit", "href": string }, ?
  { "rel": "delete", "href": string } ?
] ?
...
}

```

XML media type: application/xml

XML serialization:

```

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

```

5.16.5.1 Operations

This Resource supports the Read, Update, and Delete operations. Create is supported through the NetworkConfigurationCollection Resource.

5.16.6 NetworkConfigurationCollection Resource

A NetworkConfigurationCollection Resource represents the Collection of NetworkConfigurations within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

```

{ "resourceURI":
  "http://schemas.dmtf.org/cimi/1/NetworkConfigurationCollection",
  "id": string,

```

```

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

```

XML serialization:

```

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

```

5.16.6.1 Operations

This Resource supports the Read and Update operations. Creation of new NetworkConfiguration Resources is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.

4886 **5.16.7 NetworkPort**

4887 A NetworkPort is a realized connection point between a Network and a Resource, such as a
 4888 Machine. Table 40 describes the NetworkPort attributes.

4889 **Table 40 – NetworkPort attributes**

Name		NetworkPort
Type URI		http://schemas.dmtf.org/cimi/1/NetworkPort
Attribute	Type	Description
state	string	The operational state of the NetworkPort. Allowable values include: CREATING : The NetworkPort is in the process of being created. STARTED : The NetworkPort is available (enabled) and ready for use. STOPPED : The NetworkPort is stopped(disabled) and not available for use. DELETING : The NetworkPort is in the process of being deleted. ERROR : The Provider has detected an error in the NetworkPort. The operations that result in transitions to the above defined states are defined in clause 5.16.7.2. Clause 5.16.8.1 defines the initial state of a NetworkPort. <u>Constraints:</u> Provider : support mandatory; mutable Consumer : support mandatory; read-only
network	ref	A reference to the Network associated with this NetworkPort. <u>Constraints:</u> Provider : support mandatory; mutable Consumer : support mandatory; read-write
portType	string	A port is used as either an Access port (a member of the network) or a Trunk port that becomes a transport for multiple networks. Allowable values include: ACCESS : a member of a network. TRUNK : transport more than one network. <u>Constraints:</u> Provider : support mandatory; mutable Consumer : support mandatory; read-write
classOfService	string	The Provider-supported category associated with a collection of attributes characterizing a level of a quality experience. Example values: GOLD : High bandwidth, low latency, low jitter SILVER : An improved service experience over bronze for voice or video traffic BRONZE : Best effort The list of possible values, and their implied quality of service, is out of scope of this specification. <u>Constraints:</u> Provider : support mandatory; mutable Consumer : support mandatory; read-write
meters	collection [Meter]	A reference to the list of Meters monitored for this NetworkPort. <u>Constraints:</u> Provider : support optional; mutable Consumer : support optional; read-only
eventLog	ref	A reference to the EventLog of this NetworkPort. <u>Constraints:</u> Provider : support optional; mutable Consumer : support optional; read-only

4890 When implementing or using NetworkPort, Providers and Consumers shall adhere to the syntax and
 4891 semantics of its attributes as described in the above table as well as in the tables describing embedded
 4892 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described

below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

JSON media type: application/json

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPort",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "network": { "href": string },
  "portType": string, ?
  "classOfService": string, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
  ] ?
  ...
}
```

XML media type: application/xml

XML serialization:

```
<NetworkPort xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <network href="xs:anyURI"/>
  <portType> xs:string </portType> ?
  <classOfService> xs:string </classOfService> ?
  <meters href="xs:anyURI"/> ?
</NetworkPort>
```

```

4932     <eventLog" href="xs:anyURI"/> ?
4933     <operation rel="edit" href="xs:anyURI"/> ?
4934     <operation rel="delete" href="xs:anyURI"/> ?
4935     <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
4936 href="xs:anyURI"/> ?
4937     <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
4938 href="xs:anyURI"/> ?
4939     <xs:any>*
4940 </NetworkPort>

```

5.16.7.1 Collections

The following clauses describe the Collection Resources owned by NetworkPorts.

5.16.7.1.1 NetworkPortMeterCollection Resource

The Resource type for each item of this Collection is "Meter" as defined in clause 5.17.3.

JSON serialization:

```

4946 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortMeterCollection",
4947   "id": string,
4948   "count": number,
4949   "meters": [
4950     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
4951       "id": string,
4952       ... remaining Meter attributes ...
4953     }, +
4954   ], ?
4955   "operations": [ { "rel": "add", "href": string } ? ]
4956   ...
4957 }

```

XML serialization:

```

4959 <Collection
4960   resourceURI="http://schemas.dmtf.org/cimi/1/NetworkPortMeterCollection"
4961   xmlns="http://schemas.dmtf.org/cimi/1">
4962   <id> xs:anyURI </id>
4963   <count> xs:integer </count>
4964   <Meter>
4965     <id> xs:anyURI </id>
4966     ... remaining Meter attributes ...
4967   </Meter> *
4968   <operation rel="add" href="xs:anyURI"/> ?
4969   <xs:any>*
4970 </Collection>

```

4971 **5.16.7.2 Operations**

4972 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 4973 NetworkPortCollection Resource.

4974 Deleting a NetworkPort shall remove that NetworkPort from the global (Cloud Entry Point)
 4975 NetworkPortCollection as well as from its corresponding Network's
 4976 NetworkPortsCollection.

4977 The following custom operations are also defined:

4978 **start**

4979 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/start

4980 This operation shall start a NetworkPort.

4981 Input parameters: None.

4982 Output parameters: None.

4983 Upon successful completion of this operation, the NetworkPort shall be in the "STARTED" state.

4984 **HTTP protocol**

4985 To start a NetworkPort, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/start" URI of the
 4986 NetworkPort where the HTTP request body shall be as described below.

4987 **JSON media type:** application/json

4988 **JSON serialization:**

```
4989 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
4990   "action": "http://schemas.dmtf.org/cimi/1/action/start",
4991   "properties": { string: string, + } ?
4992   ...
4993 }
```

4994 **XML media type:** application/xml

4995 **XML serialization**

```
4996 <Action xmlns="http://schemas.dmtf.org/cimi/1">
4997   <action> http://schemas.dmtf.org/cimi/1/action/start </action>
4998   <property key="xs:string"> xs:string </property> *
4999   <xs:any>*
5000 </Action>
```

5001 Upon successful processing of the request, the HTTP response body may be empty.

5002 **stop**

5003 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/stop

5004 This operation shall stop a NetworkPort. If stopped, the NetworkPort shall not be available for use
 5005 and no network traffic shall flow through it.

ISO/IEC 19831:2015 (E)

5006 Input parameters: None.

5007 Output parameters: None.

5008 Upon successful completion of this operation, the NetworkPort shall be in the "STOPPED" state.

5009 HTTP protocol

5010 To stop a NetworkPort, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the
5011 NetworkPort where the HTTP request body shall be as described below.

5012 **JSON media type:** application/json

5013 JSON serialization:

```
5014 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",  
5015   "action": "http://schemas.dmtf.org/cimi/1/action/stop",  
5016   "properties": { string: string, + } ?  
5017   ...  
5018 }
```

5019 **XML media type:** application/xml

5020 XML serialization

```
5021 <Action xmlns="http://schemas.dmtf.org/cimi/1">  
5022   <action> http://schemas.dmtf.org/cimi/1/action/stop </action>  
5023   <property key="xs:string"> xs:string </property> *  
5024   <xs:any>*</xs:any>  
5025 </Action>
```

5026 Upon successful processing of the request, the HTTP response body may be empty.

5027 5.16.8 NetworkPortCollection Resource

5028 A NetworkPortCollection Resource represents the Collection of NetworkPorts within a
5029 Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as
5030 follows:

5031 JSON serialization:

```
5032 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortCollection",  
5033   "id": string,  
5034   "count": number,  
5035   "networkPorts": [  
5036     { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPort",  
5037       "id": string,  
5038       ... remaining NetworkPort attributes ...  
5039     }, +  
5040   ], ?  
5041   "operations": [ { "rel": "add", "href": string } ? ]  
5042   ...
```

}

XML serialization:

```

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

```

5.16.8.1 Operations

NOTE The "add" operation requires that a NetworkPortTemplate be used (see 4.2.1.1).

If NetworkPorts are created through the global (Cloud Entry Point) NetworkPortCollection's "add" operation, they are added automatically to the corresponding Network's NetworkPortCollection Resource as well.

Upon successful processing of the "add" operation, unless otherwise specified by the NetworkPortTemplate "initialState" attribute, the state of the new NetworkPort shall be the value of the DefaultInitialState capability of the NetworkPort Resource's ResourceMetadata, if defined. If no DefaultInitialState capability is defined, the default value shall be "STOPPED." The semantics of "initialState" shall be equivalent to the Provider issuing the appropriate actions against the new NetworkPort to move it into that state.

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

5.16.9 NetworkPortTemplate Resource

The NetworkPortTemplate is a set of Configuration values for realizing a NetworkPort. A NetworkPortTemplate may be used to create multiple NetworkPorts. Table 41 describes the NetworkPortTemplate attributes.

Table 41 – NetworkPortTemplate attributes

Name		
NetworkPortTemplate		
Type URI		
http://schemas.dmtf.org/cimi/1/NetworkPortTemplate		
Attribute	Type	Description
initialState	string	The initial state of the new NetworkPort. Possible values include the non-transient states as specified by the NetworkPort "state" attribute (i.e., STARTED, STOPPED) and shall be determined by the actions supported by the Provider. Providers should advertise the list of available values via the NetworkPort ResourceMetadata "initialStates" capability. Constraints: Provider: support optional; mutable

Name	NetworkPortTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkPortTemplate	
Attribute	Type	Description
		Consumer: support optional; read-write
network	<i>ref</i>	A reference to the network to be associated with this NetworkPort. If this Template is used to create a new NetworkPort through the global (Cloud Entry Point) NetworkPort Collection, this attribute shall be present. If this Template is used to create a new NetworkPort through a Network's NetworkPortsCollection, this attribute shall either be absent or have the same value as the "id" of the Network to which this NetworkPort is being added. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
networkPortConfig	<i>ref</i>	A reference to the NetworkPortConfiguration that is used to create a NetworkPort from this NetworkPortTemplate. Note that the attributes of the NetworkPortConfiguration may be specified rather than a reference to an existing NetworkPortConfiguration Resource. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
meterTemplates	<i>meterTemplates[]</i>	A list of references to MeterTemplates that shall be used to create and connect a set of new Meters to the new NetworkPort. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
eventLogTemplate	<i>ref</i>	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new NetworkPort. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

5075 When implementing or using NetworkPortTemplate, Providers and Consumers shall adhere to the
5076 syntax and semantics of its attributes as described in the above table as well as in the tables describing
5077 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
5078 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
5079 Resource in both JSON and XML.

5080 **JSON media type:** application/json

5081 **JSON serialization:**

```

5082 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortTemplate",
5083   "id": string,
5084   "name": string, ?
5085   "description": string, ?
5086   "created": string, ?
5087   "updated": string, ?
5088   "properties": { string: string, + }, ?
5089   "initialState": string, ?

```

```

"network": { "href": string }, ?
"networkPortConfig": {
  "href": string | ... NetworkPortConfiguration attributes ...
},
"meterTemplates": [
  { "href": string, ?
    ... MeterTemplate attributes ... ?
  }, *
], ?
"eventLogTemplate": {
  "href": string, ?
  ... EventLogTemplate attributes ... ?
}, ?
"operations": [
  { "rel": "edit", "href": string }, ?
  { "rel": "delete", "href": string } ?
] ?
...
}

```

XML media type: application/xml

XML serialization:

```

<NetworkPortTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <initialState> xs:string </initialState> ?
  <network href="xs:anyURI"/> ?
  <networkPortConfig href="xs:anyURI"?>
    ... NetworkPortConfiguration attributes ... ?
  </networkPortConfig>
  <meterTemplate href="xs:anyURI"? >
    ... MeterTemplate attributes ... ?
  </meterTemplate> *
  <eventLogTemplate href="xs:anyURI"? >
    ... EventLogTemplate attributes ... ?
  </eventLogTemplate> ?

```

```

5129     <operation rel="edit" href="xs:anyURI"/> ?
5130     <operation rel="delete" href="xs:anyURI"/> ?
5131     <xs:any>*
5132 </NetworkPortTemplate>

```

5.16.9.1 Operations

This Resource supports the Read, Update, and Delete operations. Create is supported through the NetworkPortTemplateCollection Resource.

5.16.10 NetworkPortTemplateCollection Resource

A NetworkPortTemplateCollection Resource represents the Collection of NetworkPortTemplates within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

```

5141 { "resourceURI":
5142     "http://schemas.dmtf.org/cimi/1/NetworkPortTemplateCollection",
5143     "id": string,
5144     "count": number,
5145     "networkPortTemplates": [
5146         { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortTemplate",
5147           "id": string,
5148           ... remaining NetworkPortTemplate attributes ...
5149         }, +
5150     ], ?
5151     "operations": [ { "rel": "add", "href": string } ? ]
5152     ...
5153 }

```

XML serialization:

```

5155 <Collection
5156     resourceURI="http://schemas.dmtf.org/cimi/1/NetworkPortTemplateCollection"
5157     xmlns="http://schemas.dmtf.org/cimi/1">
5158     <id> xs:anyURI </id>
5159     <count> xs:integer </count>
5160     <NetworkPortTemplate>
5161         <id> xs:anyURI </id>
5162         ... remaining NetworkPortTemplate attributes ...
5163     </NetworkPortTemplate> *
5164     <operation rel="add" href="xs:anyURI"/> ?
5165     <xs:any>*
5166 </Collection>

```

5167 **5.16.10.1 Operations**

5168 This Resource supports the Read and Update operations. Creation of new `NetworkPortTemplate`
 5169 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 5170 4.2.1.1.

5171 **5.16.11 NetworkPortConfiguration Resource**

5172 The set of configuration values representing the information needed to create a `NetworkPort` with
 5173 certain characteristics. Table 42 describes the `NetworkPortConfiguration` attributes.

5174 **Table 42 – NetworkPortConfiguration attributes**

Name		
Type URI		
http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration		
Attribute	Type	Description
portType	string	A port is used as an Access port (a member of the network) or a Trunk port that becomes a transport for multiple networks. Allowable values include: ACCESS : a member of a network. TRUNK : transport more than one network. Constraints: Provider : support mandatory; mutable Consumer : support mandatory; read-write
classOfService	string	The Provider-supported category associated with a collection of attributes characterizing a level of a quality experience Example values: GOLD : High bandwidth, low latency, low jitter SILVER : An improved service experience over bronze for voice or video traffic BRONZE : Best effort The list of possible values, and their implied quality of service, is out of scope of this specification. Constraints: Provider : support mandatory; mutable Consumer : support mandatory; read-write

5175 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

5176 **JSON media type:** application/json

5177 **JSON serialization:**

```

5178 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration",
5179   "id": string,
5180   "name": string, ?
5181   "description": string, ?
5182   "created": string, ?
5183   "updated": string, ?
5184   "properties": { string: string, + }, ?
5185   "portType": string, ?
5186   "classOfService": string, ?
5187   "operations": [
5188     { "rel": "edit", "href": string }, ?
5189     { "rel": "delete", "href": string } ?

```

```

5190     ] ?
5191     ...
5192 }

```

5193 **XML media type:** application/xml

5194 **XML serialization:**

```

5195     <NetworkPortConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
5196         <id> xs:anyURI </id>
5197         <name> xs:string </name> ?
5198         <description> xs:string </description> ?
5199         <created> xs:dateTime </created> ?
5200         <updated> xs:dateTime </updated> ?
5201         <property key="xs:string"> xs:string </property> *
5202         <portType> xs:string </portType> ?
5203         <classOfService> xs:string </classOfService> ?
5204         <operation rel="edit" href="xs:anyURI"/> ?
5205         <operation rel="delete" href="xs:anyURI"/> ?
5206         <xs:any> *
5207     </NetworkPortConfiguration>

```

5208 5.16.11.1 Operations

5209 This Resource supports the Read, Update, and Delete operations. Create is supported through the
5210 NetworkPortConfigurationCollection Resource.

5211 5.16.12 NetworkPortConfigurationCollection Resource

5212 A NetworkPortConfigurationCollection Resource represents the Collection of
5213 NetworkPortConfigurations within a Provider and follows the Collection pattern defined in
5214 clause 5.5.12. This Resource shall be serialized as follows:

5215 **JSON serialization:**

```

5216     { "resourceURI":
5217         "http://schemas.dmtf.org/cimi/1/NetworkPortConfigurationCollection",
5218         "id": string,
5219         "count": number,
5220         "networkPortConfigurations": [
5221             { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration",
5222               "id": string,
5223               ... remaining NetworkPortConfiguration attributes ...
5224             }, +
5225         ], ?
5226         "operations": [ { "rel": "add", "href": string } ? ]
5227     }

```

}

XML serialization:

```

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

```

5.16.12.1 Operations

This Resource supports the Read and Update operations. Creation of new NetworkPortConfiguration Resources is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.

5.16.13 Address Resource

An Address represents an IP address, and its associated metadata, for a particular Network. If a Consumer creates an Address Resource, it is the semantic equivalent of asking for a static IP address that can then be associated with Resources at a later point in time. Addresses that are manually created by Consumers shall not be deleted automatically if the Resource (e.g., a Machine) that is using that Address is deleted because these manually created Addresses are expected to have a lifetime that is different from the Resources that use them. Addresses that are created by Providers on the Consumer's behalf shall be deleted at the Provider's discretion. In particular, the Provider shall delete Addresses that it created on behalf of the Consumer if the Resource that is using that Address is deleted or if the Address becomes disassociated from the Resource.

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

Table 43 describes the Address attributes.

Table 43 – Address attributes

Name	Address	
Type URI	http://schemas.dmtf.org/cimi/1/Address	
Attribute	Type	Description
ip	string	The IP address assigned to a virtual interface. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
hostname	string	The DNS resolvable name associated with this network interface. Constraints: Provider: support optional; mutable

Name	Address	
Type URI	http://schemas.dmtf.org/cimi/1/Address	
Attribute	Type	Description
		Consumer: support optional; read-write
allocation	<i>string</i>	The value is either " dynamic " or " static ". Expresses whether this Address is controlled by the Provider or Consumer. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
defaultGateway	<i>string</i>	An IP address of a router that serves other networks. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
dns	<i>string[]</i>	The IP addresses of the Domain Name Services for host name to IP resolution. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
protocol	<i>string</i>	The selected network protocol, such as IPv4 or IPv6. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
mask	<i>string</i>	The network mask associated with this Address. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
network	<i>ref</i>	A reference to the Network with which this Address is associated. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
resource	<i>ref</i>	A reference to the Resource that is using this Address. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only

5261 When implementing or using Address, Providers and Consumers shall adhere to the syntax and
 5262 semantics of its attributes as described in the above table as well as in the table describing related
 5263 Collections. Both Consumer and Provider shall serialize this Resource as described below. The following
 5264 pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

5265 **JSON media type:** application/json

5266 **JSON serialization:**

```

5267 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Address",
5268   "id": string,
5269   "name": string, ?
5270   "description": string, ?
5271   "created": string, ?
5272   "updated": string, ?
5273   "properties": { string: string, + }, ?
5274   "ip": string,
5275   "hostname": string, ?
5276   "allocation": string,
5277   "defaultGateway": string, ?
5278   "dns": [ string, + ], ?

```

```

"protocol": string,
"mask": string, ?
"network": { "href": string },
"resource": { "href": string }, ?
"operations": [
  { "rel": "edit", "href": string }, ?
  { "rel": "delete", "href": string } ?
] ?
...
}

```

XML media type: application/xml

XML serialization:

```

<Address xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <ip> xs:string </ip>
  <hostname> xs:string </hostname> ?
  <allocation> xs:string </allocation>
  <defaultGateway> xs:string </defaultGateway> ?
  <dns> xs:string </dns> *
  <protocol> xs:string </protocol>
  <mask> xs:string </mask> ?
  <network href="xs:anyURI"/>
  <resource href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</Address>

```

5.16.13.1 Operations

This Resource supports the Read, Update, and Delete operations. Create is supported through the AddressCollection Resource.

5.16.14 AddressCollection Resource

An AddressCollection Resource represents the Collection of Addresses within a Provider that are owned/managed by the Consumer or Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

```

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

```

XML serialization:

```

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

```

5.16.14.1 Operations

NOTE The "add" operation requires that an AddressTemplate be used (see 4.2.1.1).

5.16.15 AddressTemplate Resource

This Resource captures the configuration values for realizing an Address. An AddressTemplate may be used to create multiple Addresses. Table 44 describes the AddressTemplate attributes.

Table 44 – AddressTemplate attributes

Name		
AddressTemplate		
Type URI		
http://schemas.dmtf.org/cimi/1/AddressTemplate		
Attribute	Type	Description
ip	string	The IP address assigned to a virtual interface. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
hostname	string	The DNS resolvable name associated with this network interface. Constraints: Provider: support optional; mutable

Name	AddressTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/AddressTemplate	
Attribute	Type	Description
		Consumer: support optional; read-write
allocation	<i>string</i>	A value of either " dynamic " or " static ". Expresses whether this address is controlled by the Provider or Consumer. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
defaultGateway	<i>string</i>	An IP address of a router that serves other networks. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
dns	<i>string[]</i>	The IP addresses of the Domain Name Services for host name to IP resolution. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
protocol	<i>string</i>	The selected network protocol, such as IPv4 or IPv6. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
mask	<i>string</i>	The network mask associated with this Address. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
network	<i>ref</i>	A reference to the Network with which this Address is associated. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

5349 When implementing or using AddressTemplate, Providers and Consumers shall adhere to the syntax
 5350 and semantics of its attributes as described in the above table as well as in the table describing the
 5351 related Collection. Both Consumer and Provider shall serialize this Resource as described below. The
 5352 following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON
 5353 and XML.

5354 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

5355 **JSON media type:** application/json

5356 **JSON serialization:**

```

5357 { "resourceURI": "http://schemas.dmtf.org/cimi/1/AddressTemplate",
5358   "id": string,
5359   "name": string, ?
5360   "description": string, ?
5361   "created": string, ?
5362   "updated": string, ?
5363   "properties": { string: string, + }, ?
5364   "ip": string,
5365   "hostname": string, ?
5366   "allocation": string,
5367   "defaultGateway": string, ?
5368   "dns": [ string, + ], ?

```

```

"protocol": string,
"mask": string, ?
"network": { "href": string },
"operations": [
  { "rel": "edit", "href": string }, ?
  { "rel": "delete", "href": string } ?
] ?
...
}

```

XML media type: application/xml

XML serialization:

```

<AddressTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <ip> xs:string </ip>
  <hostname> xs:string </hostname> ?
  <allocation> xs:string </allocation>
  <defaultGateway> xs:string </defaultGateway>
  <dns> xs:string </dns> +
  <protocol> xs:string </protocol>
  <mask> xs:string </mask>
  <network href="xs:anyURI"/>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</AddressTemplate>

```

5.16.15.1 Operations

This Resource supports the Read, Update, and Delete operations. Create is supported through the AddressTemplateCollection Resource.

5.16.16 AddressTemplateCollection Resource

An AddressTemplateCollection Resource represents the Collection of AddressTemplate Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

```

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

```

XML serialization:

```

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

```

5.16.16.1 Operations

This Resource supports the Read and Update operations. Creation of new AddressTemplate Resources is supported by the way of a POST to the "addLink" URI as described in clause 4.2.1.1.

5.16.17 ForwardingGroup Resource

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

Networks in a ForwardingGroup should all have the same "networkType" attributes, which prevents a Network with a "private" networkType attribute from being publicly forwarded because it is a member of a ForwardingGroup that also contains Networks with a "public" networkType attribute.

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

Table 45 describes the ForwardingGroup attributes.