



Document Identifier: DSP0263

Date: 2014-09-22

Version: 2.0.0a

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

Information for Work-in-Progress version:

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

Provide any comments through the DMTF Feedback Portal:

<http://www.dmtf.org/standards/feedback>

Document Type: Specification

Document Status: Work in Progress - Not a DMTF Standard

Document Language: en-US

12 Copyright Notice

13 Copyright © 2014 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

14 DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems
15 management and interoperability. Members and non-members may reproduce DMTF specifications and
16 documents, provided that correct attribution is given. As DMTF specifications may be revised from time to
17 time, the particular version and release date should always be noted.

18 Implementation of certain elements of this standard or proposed standard may be subject to third party
19 patent rights, including provisional patent rights (herein "patent rights"). DMTF makes no representations
20 to users of the standard as to the existence of such rights, and is not responsible to recognize, disclose,
21 or identify any or all such third party patent right, owners or claimants, nor for any incomplete or
22 inaccurate identification or disclosure of such rights, owners or claimants. DMTF shall have no liability to
23 any party, in any manner or circumstance, under any legal theory whatsoever, for failure to recognize,
24 disclose, or identify any such third party patent rights, or for such party's reliance on the standard or
25 incorporation thereof in its product, protocols or testing procedures. DMTF shall have no liability to any
26 party implementing such standard, whether such implementation is foreseeable or not, nor to any patent
27 owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is
28 withdrawn or modified after publication, and shall be indemnified and held harmless by any party
29 implementing the standard from any and all claims of infringement by a patent owner for such
30 implementations.

31 For information about patents held by third-parties which have notified the DMTF that, in their opinion,
32 such patent may relate to or impact implementations of DMTF standards, visit
33 <http://www.dmtf.org/about/policies/disclosures.php>.

CONTENTS

35	Foreword	7
36	1 Scope	9
37	1.1 Document structure	9
38	1.2 Document versioning scheme	9
39	1.3 Typographical conventions	9
40	2 Normative references	10
41	3 Terms and definitions	12
42	4 HTTP-based protocol	14
43	4.1 Introduction	14
44	4.1.1 Protocol evolution and client expectations	14
45	4.1.2 XML namespaces	15
46	4.1.3 URI space	15
47	4.1.4 Media types.....	15
48	4.1.5 Request headers.....	16
49	4.1.6 Request query parameters	16
50	4.1.7 Response headers.....	21
51	4.2 Protocol operations	22
52	4.2.1 Common CRUD operations	23
53	4.2.2 Error handling	30
54	4.3 OVF support.....	31
55	5 Model.....	31
56	5.1 Resource wrappers.....	31
57	5.2 Extensibility	32
58	5.3 Identifiers	32
59	5.4 Attribute constraints	33
60	5.5 Data types and their serialization.....	34
61	5.5.1 boolean	34
62	5.5.2 dateTime	34
63	5.5.3 duration	34
64	5.5.4 integer	35
65	5.5.5 string	35
66	5.5.6 ref.....	35
67	5.5.7 map	36
68	5.5.8 structure	36
69	5.5.9 byte[]	37
70	5.5.10 URI.....	37
71	5.5.11 Array	37
72	5.5.12 Collection	38
73	5.5.13 "Any" type	47
74	5.5.14 valueScope	47
75	5.5.15 Empty attribute values	50
76	5.6 Units.....	50
77	5.7 Resources.....	50
78	5.7.1 Primary and secondary Resources.....	50
79	5.7.2 Common attributes.....	50
80	5.8 Operations	52
81	5.9 Alternative model formats	53
82	5.10 Relationship semantics between Resources	53
83	5.10.1 Referencing across Resources.....	53
84	5.10.2 Component Resources	53
85	5.10.3 Associated Resources	54
86	5.11 Resource metadata.....	54

87	5.11.1	Capabilities	58
88	5.11.2	ResourceMetadataCollection Resource	62
89	5.12	Cloud Entry Point	63
90	5.12.1	Operations	69
91	5.13	System Resources and relationships	69
92	5.13.1	System	69
93	5.13.2	SystemCollection Resource	76
94	5.13.3	SystemTemplate Resource	78
95	5.13.4	SystemTemplateCollection Resource	84
96	5.14	Machine Resources and relationships	85
97	5.14.1	Machine	86
98	5.14.2	MachineCollection Resource	102
99	5.14.3	MachineTemplate	103
100	5.14.4	MachineTemplateCollection Resource	110
101	5.14.5	MachineConfiguration Resource	111
102	5.14.6	MachineConfigurationCollection Resource	113
103	5.14.7	MachineImage Resource	114
104	5.14.8	MachineImageCollection Resource	118
105	5.14.9	Credential Resource	119
106	5.14.10	CredentialCollection Resource	121
107	5.14.11	CredentialTemplate Resource	121
108	5.14.12	CredentialTemplateCollection Resource	123
109	5.15	Volume Resources and relationships	124
110	5.15.1	Volume	125
111	5.15.2	VolumeCollection Resource	128
112	5.15.3	VolumeTemplate Resource	129
113	5.15.4	VolumeTemplateCollection Resource	131
114	5.15.5	VolumeConfiguration Resource	132
115	5.15.6	VolumeConfigurationCollection Resource	134
116	5.15.7	VolumeImage Resource	135
117	5.15.8	VolumeImageCollection Resource	137
118	5.16	Network Resources and relationships	138
119	5.16.1	Network	138
120	5.16.2	NetworkCollection Resource	143
121	5.16.3	NetworkTemplate Resource	144
122	5.16.4	NetworkTemplateCollection Resource	147
123	5.16.5	NetworkConfiguration Resource	148
124	5.16.6	NetworkConfigurationCollection Resource	149
125	5.16.7	NetworkPort	151
126	5.16.8	NetworkPortCollection Resource	155
127	5.16.9	NetworkPortTemplate Resource	156
128	5.16.10	NetworkPortTemplateCollection Resource	158
129	5.16.11	NetworkPortConfiguration Resource	159
130	5.16.12	NetworkPortConfigurationCollection Resource	161
131	5.16.13	Address Resource	161
132	5.16.14	AddressCollection Resource	164
133	5.16.15	AddressTemplate Resource	165
134	5.16.16	AddressTemplateCollection Resource	167
135	5.16.17	ForwardingGroup Resource	168
136	5.16.18	ForwardingGroupCollection Resource	169
137	5.16.19	ForwardingGroupTemplate Resource	170
138	5.16.20	ForwardingGroupTemplateCollection Resource	172
139	5.17	Monitoring Resources and relationships	172
140	5.17.1	Job Resource	173
141	5.17.2	JobCollection Resource	177
142	5.17.3	Meter Resource	178

143	5.17.4 MeterCollection Resource	184
144	5.17.5 MeterTemplate Resource	185
145	5.17.6 MeterTemplateCollection Resource	186
146	5.17.7 MeterConfiguration Resource	187
147	5.17.8 MeterConfigurationCollection Resource	190
148	5.17.9 EventLog Resource	191
149	5.17.10 EventLogCollection Resource	194
150	5.17.11 EventLogTemplate Resource	195
151	5.17.12 EventLogTemplateCollection Resource	196
152	5.17.13 Event Resource	197
153	6 Security considerations	205
154	ANNEX A (normative) OVF support in CIMI	207
155	ANNEX B (informative) XML Schema.....	209
156	ANNEX C (informative) Change log.....	210
157	Bibliography	211
158		

159 **Figures**

160	Figure 1 - Cloud Entry Point.....	63
161	Figure 2 - System Resources.....	69
162	Figure 3 - Machine Resources	86
163	Figure 4 - Volume Resources	124
164	Figure 5 - Network Resources	138
165	Figure 6 - Monitoring Resources.....	173
166		

167 **Tables**

168	Table 1 – XML namespaces	15
169	Table 2 – Named structure.....	36
170	Table 3 – Converting a relative URI to an absolute URI	37
171	Table 4 – Numerical equivalents for attributes.....	50
172	Table 5 – Common attributes.....	51
173	Table 6 – ResourceMetadata attributes	54
174	Table 7 – Capability URIs	59
175	Table 8 – CloudEntryPoint attributes	64
176	Table 9 – System attributes	70
177	Table 10 – SystemTemplate attributes	79
178	Table 11 – Machine attributes.....	86
179	Table 12 – Disk attributes	90
180	Table 13 – <code>locatedVolume</code> accessory attributes	92
181	Table 14 – NetworkInterface attributes	93
182	Table 15 – MachineTemplate attributes.....	103
183	Table 16 – MachineConfiguration attributes	111
184	Table 17 – MachineImage attributes.....	115
185	Table 18 – Credential attributes	119
186	Table 19 – UserName/Password attributes	119

187 Table 20 – Public key attributes 119

188 Table 21 – CredentialTemplate attributes 122

189 Table 22 – Volume attributes 125

190 Table 23 – VolumeTemplate attributes 129

191 Table 24 – VolumeConfiguration attributes 133

192 Table 25 – VolumeImage attributes 135

193 Table 26 – Network attributes 138

194 Table 27 – NetworkTemplate attributes 144

195 Table 28 – NetworkConfiguration attributes 148

196 Table 29 – NetworkPort attributes 151

197 Table 30 – NetworkPortTemplate attributes 156

198 Table 31 – NetworkPortConfiguration attributes 159

199 Table 32 – Address attributes 162

200 Table 33 – AddressTemplate attributes 165

201 Table 34 – ForwardingGroup attributes 168

202 Table 35 – ForwardingGroupTemplate attributes 170

203 Table 36 – Job attributes 174

204 Table 37 – Meter attributes 178

205 Table 38 – Sample attributes 181

206 Table 39 – MeterTemplate attributes 185

207 Table 40 – MeterConfiguration attributes 187

208 Table 41 – aspect URIs 189

209 Table 42 – EventLog attributes 191

210 Table 43 – EventLogTemplate attributes 195

211 Table 44 – Event attributes 197

212 Table 45 – type URIs 200

213

214

215

Foreword

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

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

221 Acknowledgments

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

223 Editors (present and past):

- 224 • Jacques Durand – Fujitsu
- 225 • Marios Andreou – Red Hat (previous)
- 226 • Doug Davis – IBM (previous)
- 227 • Gilbert Pilz – Oracle (previous)

228 Contributors:

- 229 • Ghazanfar Ali – ZTE Corporation
- 230 • Marios Andreou – Red Hat
- 231 • Keith Bankston – Microsoft Corporation
- 232 • Winston Bumpus – VMware Inc.
- 233 • Nathan Burkhart – Microsoft Corporation
- 234 • Mark Carlson – Oracle
- 235 • Steve Carter – Novell
- 236 • Junsheng Chu – ZTE Corporation
- 237 • Josh Cohen – Microsoft Corporation
- 238 • Derek Coleman – Hewlett-Packard Company
- 239 • John Crandall – Brocade Communications Systems
- 240 • Doug Davis – IBM
- 241 • Jim Davis – WBEM Solutions
- 242 • Fernando de la Iglesia – Telefónica
- 243 • Hiroshi Dempo – NEC Corporation
- 244 • Jacques Durand – Fujitsu
- 245 • Yigal Ederly – Microsoft Corporation
- 246 • George Ericson – EMC
- 247 • Colleen Evans – Microsoft Corporation
- 248 • Norbert Floeren – Ericsson AB
- 249 • Robert Freund – Hitachi, Ltd.
- 250 • Fermín Galán – Telefónica
- 251 • Krishnan Gopalan – Microsoft Corporation
- 252 • Kazunori Iwasa – Fujitsu
- 253 • Mark Johnson – IBM
- 254 • Bhumip Khasnabish – ZTE Corporation
- 255 • Dies Köper – Fujitsu
- 256 • Vincent Kowalski – BMC Software
- 257 • Ruby Krishnaswamy – France Telecom Group
- 258 • Lawrence Lamers – VMware Inc.
- 259 • Paul Lipton – CA Technologies
- 260 • James Livingston – NEC Corporation
- 261 • Vince Lubsey – Virtustream Inc.

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

296 Cloud Infrastructure Management Interface (CIMI) Model and 297 RESTful HTTP-based Protocol

298 1 Scope

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

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

311 1.1 Document structure

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

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

315 1.2 Document versioning scheme

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

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

320 1.3 Typographical conventions

321 This specification uses the following conventions:

322 In the narrative text of the specification:

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

334 When used in their common English sense they remain lowercase. However, CIMI modeling
 335 concepts that are used in a commonly understood manner remain in lowercase, such as:
 336 attribute, operation.

337 Inside tables describing the Resource data model:

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

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

- 346 • Values in *italics* indicate data types instead of literal values.
- 347 • Characters are appended to items to indicate cardinality:
 - 348 – "?" (0 or 1)
 - 349 – "*" (0 or more)
 - 350 – "+" (1 or more)
- 351 • Vertical bars, "|", denote choice. For example, "a|b" means a choice between "a" and "b".
- 352 • The characters {, }, [, and] are block delimiters within the pseudo-schema. (Blocks may extend
 353 over multiple lines.)
- 354 • Parentheses, "(" and ")" are used in the pseudo-schema only to indicate the scope of the
 355 operators "?", "*", "+ and |".
- 356 • Ellipses (i.e., "...") indicate points of extensibility. Note that the lack of an ellipses does not mean
 357 no extensibility point exists, rather it is just not explicitly called out - usually for the sake of
 358 brevity.
- 359 • The scope of "?", "*", "+" and "|" follows these rules:
 - 360 • If immediately following a block delimiter or an array closing symbol e.g., "], ?" the scope is
 361 the entire block.
 - 362 • If not following any closing block delimiter, the scope is everything that precedes it on the
 363 same single line.

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

366 2 Normative references

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

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

- 373 DMTF DSP0243, *Open Virtualization Format Specification 1.1*,
374 http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.pdf
- 375 DMTF DSP0259, *Cloud Infrastructure Management Interface - CIM Model (CIMI-CIM) 0.0.1*,
376 <http://members.dmtf.org/apps/org/workgroup/cmwg/download.php/yyyy>
377
- 378 DMTF DSP0262, *Cloud Audit Data Federation (CADF) -Data Format and Interface Definitions*
379 *Specification version 1.0.0*,
380 http://dmtf.org/sites/default/files/standards/documents/DSP0262_1.0.0.pdf
- 381 DMTF DSP1001, *Management Profile Specification Usage Guide 1.1*,
382 http://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf
- 383 DMTF DSP4004, *DMTF Release Process 2.4*,
384 http://www.dmtf.org/sites/default/files/standards/documents/DSP4004_2.4.pdf
- 385 IANA HTTP Header Registry, <http://www.iana.org/assignments/message-headers/perm-headers.html>
- 386 IEC 80000-13:2008, International Organization for Standardization, Geneva, Switzerland, *Quantities and*
387 *units – Part 13: Information science and technology*, April 2008,
388 http://www.iso.org/iso/catalogue_detail?csnumber=31898
- 389 IETF RFC2616, R. Fielding et al, *Hypertext Transfer Protocol -- HTTP/1.1*,
390 <http://www.ietf.org/rfc/rfc2616.txt>
- 391 IETF RFC3986, T.Berners-Lee et al, *Uniform Resource Identifiers (URI): Generic Syntax*, August 1998,
392 <http://www.ietf.org/rfc/rfc3986.txt>
- 393 IETF RFC4627, D. Crockford, *The application/json Media Type for JavaScript Object Notation (JSON)*,
394 July 2006, <http://www.ietf.org/rfc/rfc4627.txt>
- 395 IETF RFC5246, T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*,
396 <http://www.ietf.org/rfc/rfc5246.txt>
- 397 ISO 8601:20044, International Organization for Standardization, Geneva, Switzerland, *Data elements and*
398 *interchange formats -- Information interchange - - Representation of dates and times*, March 2008,
399 http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40874
- 400 ISO/IEC 14977:1996, Roger S. Scowen, *Extended BNF — A generic base standard*. Software
401 Engineering Standards Symposium 1993.
402 http://www.iso.org/iso/catalogue_detail?csnumber=26153
- 403 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,
404 <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>
- 405 NIST Special Publication 800-145, Peter Mell and Timothy Grance, *The NIST Definition of Cloud*
406 *Computing*, Sept. 2011, <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- 407 NIST Special Publication 500-292, Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee
408 Badger and Dawn Leaf, *NIST Cloud Computing Reference Architecture*, Sept. 2011,
409 [http://collaborate.nist.gov/wiki-cloud-](http://collaborate.nist.gov/wiki-cloud-computing/pub/CloudComputing/ReferenceArchitectureTaxonomy/NIST_SP_500-292_-_090611.pdf)
410 [computing/pub/CloudComputing/ReferenceArchitectureTaxonomy/NIST_SP_500-292_-_090611.pdf](http://collaborate.nist.gov/wiki-cloud-computing/pub/CloudComputing/ReferenceArchitectureTaxonomy/NIST_SP_500-292_-_090611.pdf)
- 411 Representational State Transfer, Roy Fielding, Doctoral dissertation, University of California, *Architectural*
412 *Styles and the Design of Network-based Software Architectures (Chapter 5)*, 2000,
413 http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
414
- 415 Unicode Standard, Unicode Consortium, *The Unicode Standard*, Version 2.0, Addison-Wesley, 1996.

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

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

422 **3 Terms and definitions**

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

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

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

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

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

438 **3.1**

439 **authentication**

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

443 **3.2**

444 **authorization**

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

447 **3.3**

448 **cloud**

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

451 **3.4**

452 **Cloud Service Consumer**

453 A category of actors that includes the Consumer Business Manager (who approves business and
454 financial expenditures for consumed services; accounts for used service instances; establishes business
455 relationships; sets up accounts, budget, and terms; etc.); the Consumer Service Administrator (who
456 requests service instances and changes to service instances; purchases services within the business
457 relationship; creates Service Users (including policies); allocates resources, such as computer and
458 storage; generates reports, such as usage; etc.); and Service Users (who use service instances provided
459 by a Cloud Service Provider). The term "Consumer" is used if the indicated action or activity could involve

460 one or more of the above actors. In cases where the distinction between the actors in this category is
 461 relevant, the more detailed term is used.

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

464 **3.5**

465 **Cloud Service Provider**

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

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

480 **3.6**

481 **Collection**

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

484 **3.7**

485 **Configuration**

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

488 **3.8**

489 **Infrastructure as a Service (IaaS)**

490 A cloud computing service model defined in section 2 of the NIST Definition of Cloud Computing [[SP800-
 491 145](#)].

492 **3.9**

493 **message confidentiality**

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

495 **3.10**

496 **message integrity**

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

499 **3.11**

500 **Resource**

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

504 **3.12**505 **Template**

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

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

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

519 **4 HTTP-based protocol**520 **4.1 Introduction**

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

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

533 **4.1.1 Protocol evolution and client expectations**

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

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

545 **4.1.2 XML namespaces**

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

548 **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

549 **4.1.3 URI space**

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

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

```
557 http://example.com/machines/12345
558 http://example.com/machines?id=12345
559 http://example.com/12345
560 http://example.com/Cloud/resource?id=12345
```

561 **4.1.4 Media types**

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

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

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

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

572 This attribute is optional for Consumers to include. If included, this attribute's value shall match the
 573 "typeURI" attribute of the corresponding ResourceMetadata Resource (see clause 5.8), if
 574 ResourceMetadata is supported. This value shall also be equivalent to the wrapping element of the
 575 XML serialization; in other words, the namespace of the wrapper element concatenated a "/" and then its
 576 localName.

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

580 4.1.5 Request headers

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

584 4.1.6 Request query parameters

585 Providers may choose to include query parameters as part of the URIs returned to Consumers.
 586 Consumers shall include those query parameters when sending messages to those URIs. CIMI defined
 587 query parameters are prefixed with a dollar sign ("\$"). If Providers choose to define query parameters,
 588 they shall not be prefixed with a dollar sign to avoid conflicts with current and future CIMI defined query
 589 parameters.

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

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

598 4.1.6.1 Filtering Collections

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

```
603 ?$filter=expression
```

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

```
607 Filter      ::= AndExpr ( 'or' Filter ) * ;
608 AndExpr     ::= Comp ( 'and' AndExpr ) *
609 Comp        ::= Attribute Op Value
610              | Value Op Attribute
611              | PropExpr
612              | '(' Filter ')'
613 Op          ::= '<' | '<=' | '=' | '>=' | '>' | '!='
614 Attribute   ::= ? resource attribute name ?
615 Value       ::= IntValue | DateValue | StringValue | BoolValue
616 IntValue    ::= /[0-9]+/
617 DateValue   ::= ? as defined by XML Schema ?
618 StringValue ::= "... " | '...'
619 BoolValue   ::= 'true' | 'false'
620 PropExpr    ::= 'property[' StringValue ']' Op StringValue
```


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

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

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

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

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

634 **Examples:**

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

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

```
637 /machines
```

638 The URI to a `Machine` is as follows:

```
639 /machines/123
```

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

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

642 The URI to the `VolumeCollection` of a `Machine` is as follows:

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

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

```
646 GET /machines?${filter}=name='mine'
```

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

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

650 If the `filter` parameter is used, the Collection's "count" attribute shall contain the number of
 651 Resources matching the filter expression.

652 **4.1.6.2 Subsetting Collections**

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

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

660 `?$first=number`

661 `?$last=number`

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

667 If Consumers include these query parameters, the ordinal positions of entries in the collection before
 668 subsetting shall be stable when no changes are made to the collection or its entries. If filtering or sorting
 669 are used in the same query, the subsetting applies to the collection resulting from those operations.

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

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

678 The inclusion of \$first or \$last does not affect the value of the Collection's returned "count" attribute: it
 679 shall contain the number of Resources in the Collection before subsetting. In case filtering is also used,
 680 "count" shall be the size of the Collection resulting from the filtering.
 681

682 4.1.6.3 Subsetting Resources

683 If retrieving the representation of a Resource, Consumers may include the \$select query parameter to
 684 specify a subset of the Resource to be acted upon. Providers shall interpret and process this query
 685 parameter as described in this section. This subsetting shall have the semantic equivalence of
 686 referencing a different Resource whose attributes are a subset of the original Resource as specified by
 687 the attribute names listed in the \$select query parameter. The format of a \$select query parameter
 688 is:

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

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

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

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

699 is equivalent to:

700 `?$select=name,state`

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

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

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

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

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

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

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

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

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

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

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

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

726 This notation, if supported (see the "QueryPathNotation" capability in 5.11.1), allows for disambiguating
 727 subsettings if the same attribute name can be found for the Collection and for each item in the collection
 728 (which is always the case for `id` and `operations`).

729 4.1.6.4 Expanding references

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

735 The serialization shall be performed as follows:

736 JSON serialization:

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

738 shall be expanded to be:

```
739 "name": {  

    740   "href": string,  

    741   ... attributes of referenced resource...  

    742 }
```

743 **XML serialization:**744

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

745 shall be expanded to be:

746

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

749 Note that in the XML case the nested elements shall not contain the wrapper element of the referenced
750 Resource (e.g., `<Machine>` in the case of a reference to a `Machine` Resource).751 The format of a `$expand` query parameter shall be:752

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

753 The value of the `$expand` query parameter is a comma-separated list of attribute names. Any attribute
754 name erroneously appearing in the list that is not part of the Resource, or is not a reference, shall be
755 ignored by the Provider. An attribute name of "*", or no attribute name list at all, is equivalent to specifying
756 all of the attributes. Any attribute name explicitly appearing more than once in a URI shall have its second
757 (and subsequent) appearances ignored.758 The `$expand` query parameter may appear more than once in a URI, which is semantically equivalent to
759 all of the attribute names appearing as values of a single `$expand` query parameter.760 If the Resource being retrieved is a Collection, the attribute names listed in the `$expand` shall apply to
761 the attributes of the entities within the Collection. For example, specifying:762

```
?$expand=volumes
```

763 if retrieving the `MachineCollection` has the same net effect as applying the "expand" semantics to
764 the specified attribute ("volumes" in this example) of each `Machine` within the Collection. To be clear,
765 `$expand` acts on the attributes of the Resources in the Collection, not on the wrapping Collection
766 Resource itself.767 **4.1.6.5 Specifying the Resource format**768 If retrieving the representation of a Resource, the HTTP Accept header is used to specify the encoding
769 style of the response. While it is recommended that Consumers use the Accept header, there might be
770 situations where Consumers are unable to control the values specified in that header. In these cases
771 Consumers may use the `$format` query parameter to override the Accept header values. Providers
772 shall interpret and process the `$format` query parameter as described in this clause.773 The `$format` parameter shall be of the form:774

```
?$format=encoding
```

775 Where "encoding" is the requested representation of the response. This specification defines two
776 possible values: "json" and "xml". Providers may support others. The value of the `$format` query
777 parameter shall be case insensitive.778 If both an Accept header and `$format` query parameter are present in a request message, the
779 `$format` value shall take precedence. If the `$format` query parameter appears more than once, the
780 second, and subsequent, appearances shall be ignored.

781 4.1.6.6 Sorting Collections

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

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

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

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

794 The sort shall be performed based on the attribute type.

795 The following rules apply to the ascending sort order:

- 796 • boolean – ‘false’ shall come before ‘true’.
- 797 • dateTime – An earlier datetime shall come before a later datetime.
- 798 • duration – A shorter duration shall come before a longer duration.
- 799 • integer – Smaller integers shall come before larger integers. Negative integers shall come
800 before positive integers.
- 801 • string – Ordering is based on a binary comparison of the transformed strings according to the
802 rules of the Normalization Form KD of the Unicode standard as defined in [Unicode Standard](#)
803 [Annex \(UAX\), annex #15](#) .

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

805 Examples:

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

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

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

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

814 If collection subsetting is used in the same query, the subsetting applies to the sorted collection. When no
815 `$orderby` is specified, the order of entries in the returned Collection is not defined.

816 4.1.7 Response headers

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

820 **4.1.7.1 Job header**

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

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

824 **4.1.7.2 ETag support**

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

828 **4.2 Protocol operations**

829 This clause defines the set of common HTTP operations that a Provider may expose. At its core, there
830 are four basic CRUD (Create, Read, Update, and Delete) operations. The manner in which these are
831 used is consistent across all Resources within the model; therefore, their use is defined once and is to be
832 applied consistently. Some Resources support specialized operations that do not fit well into a CRUD
833 style of operation and those follow a similar high-level pattern, but each operation is allowed to have slight
834 variations to accommodate its specific needs. The specifics of these special operations are detailed within
835 the clause that defines the Resource.

836 If appropriate, some of the Resource representations include an "operations" attribute. Providers shall
837 only include the "operations" attribute if the specified operations are accessible to the current client for
838 that particular Resource. This situation means that based on many factors (e.g., authorization rights of the
839 clients, current state of the Resource, etc.), a different set of "operations" shall be returned on each
840 serialization of the Resource.

841 Each operation shall include a "rel" and an "href" field. The "rel" field shall uniquely identify the operation
842 name (e.g., "add", "edit"), while the "href" field is the URI to which the operation's request message shall
843 be sent. Note that the "href" field's URI may be different from the URI of the Resource itself. Each
844 operation may have an "available" field to indicate that the operation can be performed by the Consumer.
845 The "available" field is of type boolean with a default value of "true". If "available" is set to "false" it
846 indicates that the operation is not currently available. This would normally indicate a temporary condition.
847 For example, some Machine operations may not be available depending on the state of the Machine.

848 The operations attribute shall be serialized as follows:

849 **JSON serialization:**

```
850 { "operations": [
851   { "rel": string, "href": string, ("available": boolean)? }, +
852 ]
853 }
```

854 **XML serialization:**

```
855 <Resource xmlns="http://schemas.dmtf.org/cimi/1">
856   <operation rel="xs:anyURI" href="xs:anyURI" (available="xs:boolean"? /> *
857 </Resource>
```

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

859 **JSON serialization:**

```
860 { "operations": [
861     { "rel": "edit", "href": "<editURI>" }
862 ]
863 }
```

864 **XML serialization:**

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

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

870 **4.2.1 Common CRUD operations**

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

873 **4.2.1.1 Creating a new Resource**

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

878 The HTTP POST request shall include:

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

882 For example, the request can be:

```
883 POST <addURI> HTTP/1.1
884 Host: <hostname>
885 Accept: application/(json|xml)
886 Content-Type: application/(json|xml)
887 Content-Length: <length>
888
889 <serialization of request to create a new resource>
```

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

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

```
898 <MachineCreate xmlns="http://schemas.dmtf.org/cimi/1">
899   <name> xs:string </name> ?
900   <description> xs:string </description> ?
901   <property key="xs:string"> xs:string </property> *
902   <machineTemplate href="xs:anyURI"? >
903     ... template attributes ... ?
904   </machineTemplate>
905 </MachineCreate>
```

906 Note that in the XML case the creation of a new `Machine` requires a wrapper element named
 907 `MachineCreate` per the rules specified in clause 5.5.12.2.

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

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

```
911 { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceCreate",
912   "name": "myResourceName", ?
913   "description": "My resource description", ?
914   "properties": { "prop1name" : "prop1value" , + }, ?
915   "resourceTemplate": {
916     <here the template is passed by value>
917   }
918 }
919 }
```

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

```
921 { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceCreate ",
922   "name": "myResourceName", ?
923   "description": "My resource description", ?
924   "properties": { "prop1name" : "prop1value" , + }, ?
925   "resourceTemplate": { "href": string ,
926     <here some template attribute/value pairs may be added to override values in the
927     referenced template>
928   }
929 }
930 }
```

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

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

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

937 **Direct setting of attributes in the new Resource:**

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

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

943 **Defining or referring to the Resource Template:**

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

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

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

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

959 `"attribute": null`

960 for the attribute in the JSON serialization, or

961 `<attribute/>`

962 in the XML serialization for that attribute.

963 **Examples:**

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

```

965 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate ",
966   "name": "myMachine123",
967   "description": "A machine to be connected to a pre-existing network",
968   "machineTemplate": {
969     <here a template passed "by value" i.e., the attribute/value pairs for the
970     MachineTemplate template. An example is of the networkInterfaces below: >
971     "networkInterfaces": [
972       { "addresses": [ { "address": { "href": "http://example.com/addresses/add1"
973       } }, { "address": { "href": "http://example.com/addresses/add2" } } ],
974       "network": { "href": "http://example.com/networks/net1" },
975       "state": "ACTIVE" }
976     ]
977   }
978 }
979 }
980 
```

981 In the previous example:

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

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

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

```

988 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate ",
989   "name": "myMachine456",
990   "description": "A machine connected to a pre-existing volume",
991   "machineTemplate": { "href": "http://example.com/machineTemplates/72000",
992     "credential": { "href": "http://example.com/myCredential" }
993   "networkInterfaces": [
994     { "addresses": [ { "address": { "href": "http://example.com/addresses/add4"
995     }}, { "address": { "href": "http://example.com/addresses/add5" } } ] },
996     { "network": { "href": "http://example.com/networks/net1" },
997       "state": "ACTIVE" }
998   ]
999 }
1000 }
1001 }
1002 }

```

1003 In the above example, a new machine named “myMachine456” is created, also connected to the same
 1004 existing Network as in example (1), but with a different set of Addresses. Two kinds of attributes are
 1005 provided with values at creation time in this example:

- 1006 • Instance-level attribute settings: these shall directly update similar attributes in the created
 1007 Resource, here `name` and `description`.
- 1008 • Template-level overrides: The referred MachineTemplate is used for creating the
 1009 Machine, but the `credential` attribute in this Template is (temporarily) overridden by the
 1010 credential provided in the creation request as is the `networkInterfaces` array. In case
 1011 such attributes were not present in the referred Template; they are added (temporarily) just for
 1012 this Machine creation.

1013 Some of the create requests allow for configuration type of Resources to be passed by-reference or by-
 1014 value as well - e.g., `Credential` on a `Machine` create operation. The processing rules defined above
 1015 applies in those cases as well.

1016 If the response has a 201 status code, the response shall include:

- 1017 • HTTP Location header with a reference to the new Resource

1018 If the response to a create request includes a serialization of the new Resource, the response shall
 1019 additionally include:

- 1020 • HTTP Content-Type header
- 1021 • HTTP Content-Length header

1022 For example, the response can be:

```

1023 HTTP/1.1 201 Created
1024 Location: <location>
1025 Content-Type: application/(json|xml)
1026 Content-Length: <length>

```

1027
 1028 `<serialization of new resource>`

1029 **4.2.1.2 Retrieving a representation of a Resource**

1030 To retrieve a representation of Resource, an HTTP GET request is sent to the Resource's URI.

1031 For example, the request can be:

```
1032 GET <ResourceURI> HTTP/1.1
1033 Host: <hostname>
1034 Accept: application/(json|xml)
```

1035 If the response has a 200 status code, the response shall include:

- 1036 • HTTP Content-Type header
- 1037 • HTTP Content-Length header

1038 For example, the response can be:

```
1039 HTTP/1.1 200 OK
1040 Content-Type: application/(json|xml)
1041 Content-Length: <length>
1042
1043 <serialization of resource>
```

1044 **4.2.1.3 Updating a Resource**

1045 To update a Resource's state, an HTTP PUT request containing the complete, updated representation is
 1046 sent to a designated `editURI` for that Resource type. Consumers shall include all non-empty attributes
 1047 of the Resource in the PUT request - including ones that it might not support or understand that were
 1048 returned in a GET response. This is to ensure that a client does not inadvertently modify (erase) data in a
 1049 Resource by excluding it from the full representation of the Resource.

1050 In many cases, this `editURI` is the same as the URI of Resource itself. Retrieving the Resource
 1051 representation shall include an "edit" operation, which contains the `editURI` that is to be used, if the
 1052 requester is allowed to modify the Resource.

1053 While processing a PUT request, if the server detects that an attempt is being made to update a
 1054 read-only, or immutable, attribute, it shall silently ignore that attribute update request and shall not
 1055 generate an error. This rule applies to Resource partial updates as well.

1056 Because of potential conflicts that might occur due to multiple concurrent updates, Consumers should use
 1057 the partial update mechanism, defined in 4.2.1.3.1, to reduce the chances of mistakenly updating
 1058 attributes with out-of-date data.

1059 The HTTP PUT request shall include:

- 1060 • CIMI serialization of the updated Resource in the HTTP Body
- 1061 • HTTP Content-Type header
- 1062 • HTTP Content-Length header

1063 For example, the request can be:

```
1064 PUT <editURI> HTTP/1.1
1065 Host: <hostname>
1066 Accept: application/(json+xml)
1067 Content-Type: application/(json+xml)
1068 Content-Length: <length>
1069
1070 <serialization of request to update a resource>
```

1071 If the response includes a serialization of the updated Resource and has a status code of 200, this
1072 response shall include:

- 1073 • HTTP Content-Type header
- 1074 • HTTP Content-Length header

1075 For example, the response can be:

```
1076 HTTP/1.1 200 OK
1077 Content-Type: application/(json+xml)
1078 Content-Length: <length>
1079
1080 <serialization of updated resource>
```

1081 4.2.1.3.1 Partial updates to a Resource

1082 For clarity, this clause explains how to use the `$select` query parameter (see clause 4.1.6.3) to subset
1083 a Resource for the purposes of only operating on a selected set of top-level attributes.

1084 To update only certain top-level attributes of a Resource, a Consumer may include only the altered
1085 attributes in the representation of the Resource within the HTTP request body. If this request is made, the
1086 URI to the Resource shall include the attributes to be modified as a comma-separated list of query
1087 parameters; in other words, the URI shall be of the form:

```
1088 http://example.com/resource?$select=attribute1,attribute2,...
```

1089 Only the attributes listed in the URI's query parameters shall be modified; attributes not listed in the URI
1090 shall not be directly modified by the request. Note that this circumstance does not preclude the
1091 modification of one attribute causing side-effects that result in the modification of an attribute not listed in
1092 the query parameters.

1093 Any attribute listed in the URI but not included within the HTTP request body shall be reset to a Resource
1094 specific value (e.g., removed).

1095 From an HTTP perspective, the updated subsetted Resource is a distinct one. The semantics of a normal
1096 HTTP PUT are adhered to; it is a complete replacement update of the specified Resource. From the
1097 Consumer's perspective, the partial update is interpreted and executed by the Cloud Service Provider,
1098 and some part of the Resource is changed.

1099 Adhering to the generic PUT semantics defined previously, any attribute of the original (full) Resource
1100 included within the HTTP request body shall result in an error being generated if that attribute is not listed
1101 in the `$select` query parameter - see clause 5.4. Note that this is due to these attributes being
1102 unknown to this subsetted Resource.

1103 The following sample request updates just the name and description attributes of a Machine:

```
1104 PUT /machines/myMachine?$select=name,description HTTP/1.1
1105 Host: <hostname>
1106 Accept: application/xml
1107 Content-Type: application/xml
1108 Content-Length: <length>
1109
1110 <Machine>
1111   <name>My New Machine</name>
1112 </Machine>
```

1113 The name attribute is set to "My New Machine" and the description attribute is erased.

1114 4.2.1.4 Deleting a Resource

1115 To delete a Resource, an HTTP DELETE request is sent to a designated deleteURI for that Resource
 1116 type. In many cases, this deleteURI is the same as the URI of Resource itself. Retrieving the
 1117 Resource representation shall include a "delete" operation, which contains the deleteURI that is to be
 1118 used, if the requester is allowed to delete the Resource.

1119 For example, the request can be:

```
1120 DELETE <deleteURI> HTTP/1.1
1121 Host: <hostname>
```

1122 If the Resource has a State attribute, its value shall be "DELETING", while the Provider is processing
 1123 this operation.

1124 For example, the response can be:

```
1125 HTTP/1.1 200 OK
```

1126 4.2.1.5 Other operations

1127 While some modifications to the Resources in the model can be done by the way of a simple update
 1128 (PUT) operation to the Resource's editURI, sometimes a more complex set of actions needs to be
 1129 taken. In these cases, the operations shall be modeled as HTTP POSTs to the operation specific URI of
 1130 the Resource.

1131 For each of the Resources that define additional operations, a description of the HTTP request and
 1132 response bodies is provided. However, the general HTTP interaction are as described below.

1133 The request shall be of the following form:

```
1134 POST <operationURI> HTTP/1.1
1135 Host: <hostname>
1136 Accept: application/(json|xml)
1137 Content-Type: application/(json|xml)
1138 Content-Length: <length>
1139
1140 <serialization of request to perform some action>
```

1141 The form of the response varies depending on the operation and is defined by the operation itself.

1142 Note that the definition of the Create operation (see clause 4.2.1.1) follows this same pattern. It is just
1143 called out for ease of reference.

1144 4.2.1.6 Synchronous operations

1145 If a Provider supports the `Job` Resource, each incoming PUT, DELETE, POST request shall result in a
1146 Job Resource being created and an absolute URI reference to that Job Resource shall be returned back
1147 to the client by the way of the CIMI-Job-URI HTTP Header in the HTTP response message:

1148 `CIMI-Job-URI: <uri-to-Job>`

1149 In this case, the requested operation shall be complete and the Job URI shall point to a completed Job. If
1150 the Job is not complete, the server shall return a 202 and follow the instructions for Asynchronous
1151 operations.

1152 4.2.1.7 Asynchronous operations

1153 In some cases, an operation requested by the client may take an undetermined amount of time to be
1154 completed. For example, creating a new `Machine` or starting an existing `Machine` may take a relatively
1155 long time to be completed. In these cases, it is not practical to complete these operations within a
1156 reasonable HTTP request timeout interval, so the Provider shall return an HTTP "202 Accepted" response
1157 code.

1158 As with synchronous operations, if a Provider supports the `Job` Resource, it shall create a `Job` Resource
1159 for the incoming request and return a reference to that `Job` Resource back to the client by the way of the
1160 CIMI-Job-URI HTTP Header in the HTTP response message. Additionally, in the case of a "202
1161 Accepted" response code, the Provider may also return any of the following in the HTTP response body:

- 1162 • A representation of the `Job` Resource, if one was created.
- 1163 • A partial representation of the response message as if the operation were a synchronous
1164 operation. For example, when creating a new `Machine`, the response message may include a
1165 partial representation of the new `Machine` in the response message. The list of attributes of the
1166 Resource that is returned is implementation specific and based upon how much information is
1167 available at the time the response message is generated, but it shall be consistent with the
1168 definition of the full Resource representation. In the case of a create operation, the Provider may
1169 also include an HTTP Location header referencing the "to be created" Resource, if it is known.
- 1170 • An empty response body.

1171 Note that the decision as to whether any particular operation is synchronous or asynchronous is at the
1172 server's discretion.

1173 4.2.2 Error handling

1174 **In cases where an error occurs during the processing of a request, the Provider shall include a**
1175 **representation of a `Job` Resource describing the status of the failed operation. This**
1176 **representation of a `Job` shall be included even in cases where the Provider does not expose `Job`**
1177 **Resources. This is to ensure that Consumers are provided with sufficient information, in a**
1178 **consistent manner, as to the reason for the failure regardless of whether the Provider exposes**
1179 **`Jobs`. A transient `Job` Resource may be created by the Provider just for error reporting. In case**
1180 **a `Job` Resource is not intended to be used for more than error reporting, the returned "id"**
1181 **attribute shall be an empty path (i.e., "") and the `nestedJobs` array shall be expanded (see**
1182 **4.1.6.4) to inline the representation of the pseudo subordinate `Jobs`.**

1183

1184 4.3 OVF support

1185 The *Open Virtualization Format (OVF) Specification* ([DSP0243](#)) describes an open, secure, portable,
 1186 efficient, and extensible format for the packaging and distribution of software to be run in virtual
 1187 machines. OVF support in CIM allows an OVF package to be used to create CIM management
 1188 resources by importing the package. Additionally, CIM management resources can be exported into an
 1189 OVF package. The actual support for the OVF package is typically provided by a hypervisor that is
 1190 managed by the CIM provider. The import of an OVF package exposes CIM specific constructs and
 1191 parameters as a result of the import without altering the original OVF package. Thus the CIM resources
 1192 that are created as a result of the import form a “View” of what the hypervisor did; however, other (non-
 1193 CIM mapped) information from the OVF package may have been used by the hypervisor in its import.
 1194 This other information is implementation dependent and is not further touched upon by this standard.

1195 An OVF package can support single virtual machines (VMs) corresponding to a single CIM *Machine* or
 1196 *MachineTemplate* (see clause 5.14.1) or may also support a complex hierarchy of VMs and their
 1197 related Resources corresponding to a CIM *System* or *SystemTemplate* (see clause 5.13.1) and
 1198 related CIM management resources.

1199 OVF support is covered in more detail in ANNEX A.

1200 5 Model

1201 This model assumes that a business relationship has already been established between the Consumer
 1202 and the Provider. This relationship may include financial terms, creating separately administered clouds
 1203 that the consuming organization is paying for, and the establishment of authentication credentials to
 1204 access the administrative entry point for each cloud. The scope of this model is one separately
 1205 administered cloud.

1206 The CIM model is described here by using a tabular representation. It is inspired from Entity-Relationship
 1207 modeling, where each entity is modeling a significant cloud resource for which independent access and
 1208 manipulation is expected. Relationships between resources use a referential mechanism based on
 1209 unique identifiers that is expected to be already supported by the implementation environment and
 1210 protocol (e.g., URIs for HTTP).

1211 The model is self-describing and allows for querying its own metadata, e.g., to discover which extensions
 1212 have been implemented. The model is also extensible in different ways (see clause 5.1).

1213 Along with this model, a serialization of its entities is defined (both in XML and JSON).

1214 An alternative UML diagram representation is provided for each major group of resources.

1215 5.1 Resource wrappers

1216 The serialization of Resource instances in the model follow these conventions. Consider the serialization
 1217 of a Resource named “MyResource”:

1218 JSON serialization:

1219 The Resource is serialized as an object wrapping all its attributes, but without a wrapper name. The
 1220 Resource includes a `resourceURI` with a URI for the type of Resource being serialized. For example:

```
1221 { "resourceURI": "http://example.com/MyResource",
1222   "attribute": "value"
1223 }
```

1224 XML serialization:

1225 The Resource is serialized as an element with name equal to the Resource name; for example:

```
1226 <MyResource xmlns="http://example.com">
1227   <attribute> value </attribute>
1228 </MyResource>
```

1229 5.2 Extensibility

1230 There are two types of extensibility mechanisms defined by the CIMI model; one is intended for use by
1231 Consumers whilst the other is to be used by Providers.

1232 The first allows for a CIMI Consumer to add additional data to a Resource. Each Resource in the CIMI
1233 model has an attribute called "properties". Consumers, when creating or updating a Resource, may
1234 store any name/value pair in the properties attribute. CIMI Providers shall store and return these
1235 values to the Consumer. There is no obligation for the Provider to understand or take any action based on
1236 these values; they are there for the Consumer's convenience. Providers shall not add elements to this
1237 properties attribute.

1238 The second type of extensibility mechanism allows for Provider defined extensions and this specification
1239 includes the ResourceMetadata Resource for this purpose. ResourceMetadata may be used to

- 1240 • express constraints on the existing CIMI defined Resource attributes (e.g., express a maximum
1241 for the 'cpu' attribute of the MachineConfiguration Resource)
- 1242 • introduce new attributes for CIMI defined Resources together with any constraints governing
1243 these (e.g., a new 'location' attribute for the Volume Resource that takes values from a defined
1244 set of strings)
- 1245 • introduce new operations for any of the CIMI defined Resources (e.g., define a new 'compress'
1246 operation for the Volume Resource)
- 1247 • express any Provider specific capabilities or features (e.g., the length of time that a Job
1248 Resource is retained after Job completion and before this is deleted)

1249 It is recommended that Providers use the ResourceMetadata Resource to advertise these attributes,
1250 operations, and capabilities along with any constraints that might need to be understood by Consumers.
1251 The ResourceMetadata Resource is defined in clause 5.8.

1252 If a Provider receives a message containing an unknown or unsupported attribute, it shall reject the
1253 request. If a Consumer receives a message containing an unknown or unsupported attribute, it shall
1254 silently ignore the attribute. However, Consumers are required to include those attributes in messages
1255 sent back to the Provider. Note in these cases the Consumer is not required to understand or process the
1256 unsupported attribute, but merely echo it back to the Provider.

1257 5.3 Identifiers

1258 All identifiers (e.g., Resource names, attributes, operations, parameter names) defined by this
1259 specification, or defined by the way of an extension, shall adhere to the following rules:

- 1260 • Identifier names shall be treated as case sensitive.
- 1261 • Identifier names shall only use the following set of characters:
 - 1262 – Uppercase ASCII (U+0041 through U+005A)
 - 1263 – Lowercase ASCII (U+0061 through U+007A)

- 1264 – Digits (U+0030 through U+0039)
- 1265 – Underscore (U+005F)

- 1266 • Identifier names shall not start with a Digit (U+0030 through U+0039).

1267 Note that these rules do not apply to the "name" common attribute defined in clause 5.7.2.

1268 5.4 Attribute constraints

1269 Each attribute of the Resources in the CIMI model is augmented by a set of constraints that further qualify
 1270 the attribute that is being defined. For each attribute, there is a Provider and a Consumer set of
 1271 constraints because each might differ. The following constraints are possible:

1272 **support optional:**

1273 This constraint indicates that support for this attribute is optional. If supported, Providers should advertise
 1274 its support through `ResourceMetadata`. See clause 5.2 for information concerning the processing of
 1275 unsupported and unknown attributes. See clause 5.5.15 regarding empty attribute values.

1276 Non-empty, Consumer-supported, writeable (i.e., read-write and write-only) attributes shall always be
 1277 included as part of the Resource representation sent from Consumers to Providers, including create
 1278 requests.

1279 Non-empty, Provider-supported attributes shall always be included as part of the Resource representation
 1280 sent from Providers to Consumers.

1281 **support mandatory:**

1282 This constraint indicates that support for this attribute is required by compliant implementations. If present
 1283 on a nested attribute, this attribute is required to be supported only if the parent attribute is supported.
 1284 See clause 5.5.15 regarding empty attribute values.

1285 Non-empty, mandatory, writeable (i.e., read-write and write-only) attributes shall always be included as
 1286 part of the Resource representation sent from Consumers to Providers - including create requests.

1287 Non-empty, Provider, mandatory attributes shall always be included as part of the Resource
 1288 representation sent from Providers to Consumers.

1289 **immutable:**

1290 This Provider constraint indicates that the attribute, once set, shall never change for the lifetime of the
 1291 Resource.

1292 **mutable:**

1293 This Provider constraint indicates that the attribute may be modified. Providers shall always have the
 1294 ability to modify these attributes. Whether Consumers have the ability to modify these attributes shall be
 1295 indicated by the read-only, read-write, and write-only constraints.

1296 **read-only:**

1297 This Consumer constraint indicates that the attribute may be retrieved but not updated by Consumers.
 1298 Read-only attributes are not required to appear in the serialization of Resources in create or update
 1299 request messages. If present, they shall be silently ignored by the Provider. Read-only attributes shall
 1300 appear in the serialization of Resources sent from Providers.

1301 **read-write:**

1302 This Consumer constraint indicates that the attribute may be retrieved and/or updated by Consumers.
 1303 Read-write attributes shall appear in the serialization of Resources sent to and from Providers. Providers

1304 may further constrain whether Consumers can update these attributes and should indicate this by the way
1305 of `ResourceMetadata`.

1306 **write-only:**

1307 This Consumer constraint indicates that the attribute may be updated by Consumers but are not
1308 retrievable by Consumers, typically for security reasons. Write-only attributes shall appear in the
1309 serialization of Resources sent to Providers but shall never appear in the serialization of Resources sent
1310 from Providers.

1311 **5.5 Data types and their serialization**

1312 Unless specifically asked to not include certain attributes in the Resource representation, the absence of
1313 an optional attribute in the representation means that the attribute has no value (i.e., is undefined),
1314 meaning there is no notion of an optional attribute having an implied value. Note that a client cannot
1315 distinguish (from just looking at the returned representation) whether a particular attribute is not supported
1316 from one that does not exist. Likewise, an absent attribute from a Resource representation as the input to
1317 an update operation means that the Consumer is requesting that the Provider remove that attribute.

1318 The following clauses describe the data types and values that are used within the model definition tables.

1319 **5.5.1 boolean**

1320 A value as defined by `xs:boolean` per [XML Schema – Part 2](#), with the exception that the only allowable
1321 values are either "true" or "false." The value is case sensitive.

1322 If serialized in JSON, these values shall be of JSON type: *boolean*

1323 If serialized in XML, these values shall be of XML Schema type: *xs:boolean*

1324 **5.5.2 dateTime**

1325 A value as defined by `xs:dateTime` per [XML Schema – Part 2](#), which is consistent with DMTF DSP4004
1326 and ISO 8601. The timestamp should preserve time zone information, i.e., include a local time component
1327 and an offset from UTC.

1328 Any constraints on the specific ranges allowed for any particular attribute are specified by that attribute's
1329 definition or at runtime by the Provider by the way of the metadata discovery mechanisms defined by this
1330 specification.

1331 For example, Monday, May 25, 2012, at 1:30:15 PM EST is represented as:

1332 `2012-05-25T13:30:15-05:00`

1333 If serialized in JSON, these values shall be of JSON type: *string*

1334 If serialized in XML, these values shall be of XML Schema type: *xs:dateTime*

1335 **5.5.3 duration**

1336 A value as defined by `xs:duration` per [XML Schema – Part 2](#). Any constraints on the specific ranges
1337 allowed for any particular attribute shall be specified by that attribute's definition or at runtime by the
1338 Provider by the way of the metadata discovery mechanisms defined by this specification.

1339 If serialized in JSON, these values shall be of JSON type: *string*

1340 If serialized in XML, these values shall be of XML Schema type: *xs:duration*

1341 **5.5.4 integer**

1342 A value as defined by xs:integer per [XML Schema – Part 2](#). Any constraints on the specific ranges
 1343 allowed for any particular attribute shall be specified by that attribute's definition or at runtime by the
 1344 Provider by the way of the metadata discovery mechanisms defined by this specification.

1345 If serialized in JSON, these values shall be of JSON type: *number*

1346 If serialized in XML, these values shall be of XML Schema type: *xs:integer*

1347 **5.5.5 string**

1348 A value as defined by xs:string per [XML Schema – Part 2](#). Any constraints on this type for any particular
 1349 attribute shall be specified by that attribute's definition or at runtime by the Provider by the way of the
 1350 metadata discovery mechanisms defined by this specification.

1351 If serialized in JSON, these values shall be of JSON type: *string*

1352 If serialized in XML, these values shall be of XML Schema type: *xs:string*

1353 If serializing an attribute of type string, the serialization shall omit this attribute in case of an empty string.

1354 **5.5.6 ref**

1355 A reference to another Resource.

1356 References allow for Consumers to navigate to Resources. By starting at the Cloud Entry Point and
 1357 following the references that appear in the retrieved Resources, Consumers are able to recursively
 1358 discover and navigate to all other Resources.

1359 As a general rule, if an attribute is of type "ref", its value shall be held by an attribute named "href"
 1360 (both in JSON and XML).

1361 **JSON serialization:**

1362 In the JSON serialization the href property appears as of type "string." If an attribute is of type
 1363 "ref", the name of this attribute shall appear as a key, with the href property as a nested value. For
 1364 example, a Resource attribute "myvolume" of type "ref" is serialized as:

1365

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

1366 **XML serialization:**

1367 In the XML serialization the href attribute appears as type "xs:anyURI." If an attribute is of type
 1368 "ref," the name of this attribute shall appear as name of an XML element with the href property as an
 1369 (XML) attribute. For example, a Resource attribute "myvolume" of type "ref" is serialized as:

1370

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

1371 References in both JSON and XML have an extensibility point that allows for additional information (such
 1372 as the target Resource to be included "by value") if supported. For convenience, the JSON and XML
 1373 representations, as shown above, exclude the implicit extensibility points that would allow for the
 1374 attributes of the target Resource to be included if desired. So, more accurately the above representations
 1375 might be written as follows:

1376 For JSON:

1377

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

1378 and in XML:

```
1379 <myvolume href="xs:anyURI"> xs:any* </myvolume>
```

1380 However, for brevity the extensibility points are excluded from the serialization of the Resources.

1381 5.5.7 map

1382 A list of key/value pairs. The same "key" shall not be used more than once within an attribute. The "key" is
1383 case sensitive.

1384 If serializing an attribute of type map, the serialization shall omit this attribute in case of an empty map.

1385 5.5.8 structure

1386 Attributes of this type are complex attributes made up of a set of nested attributes. For each attribute of
1387 this type, there is an additional table defining those nested attributes.

1388 A nested structure can be considered a complex type definition. Structures may be named or unnamed.
1389 Table 2 is an example of named structure:

1390

Table 2 – Named structure

Name	<i>summary</i>	
Attribute	Type	Description
low	<i>number</i>	Number of "low" occurrences
medium	<i>number</i>	Number of "medium" occurrences
high	<i>number</i>	Number of "high" occurrences
critical	<i>number</i>	Number of "critical" occurrences

1391 JSON serialization:

1392 In JSON, the name of the structure (i.e., of the type it represents) never appears. In other words, whether
1393 the structure is named or not does not matter. An attribute named "systemIncidents" of type
1394 "summary" (as above) is serialized as follows:

```
1395 "systemIncidents": {
1396   "low": number,
1397   "medium": number,
1398   "high": number,
1399   "critical": number
1400 }
```

1401 XML serialization:

1402 In XML, the name of the structure (i.e., of the type it represents) never appears. In other words, whether
1403 the structure is named or not does not matter. The same previous "systemIncidents" example is
1404 serialized so that the structure sub-attributes become XML attributes of a <systemIncidents> XML
1405 element wrapper:

```
1406 <systemIncidents low="xs:integer" medium="xs:integer" high="xs:integer"
1407   critical="xs:integer"/>
```

1408 NOTE A large number of sub-attributes of atomic type in a structure may be represented alternatively as XML child
1409 elements for better readability. Both options are available; however, the same structure shall be serialized the same
1410 way across Resources.

1411 **5.5.9 byte[]**

1412 An arbitrary set of bytes meant to represent a block of binary data. Any constraints on this type for any
 1413 particular attribute shall be specified by that attribute's definition or at runtime by the Provider by the way
 1414 of the metadata discovery mechanisms defined by this specification.

1415 If serialized in JSON, these values shall be of JSON type: *string*

1416 If serialized in XML, these values shall be of XML Schema type: *xs:hexBinary*

1417 **5.5.10 URI**

1418 The format and syntax of the attributes of type "URI" is defined by [RFC3986](#).

1419 Unless otherwise noted, this specification does not mandate whether Providers use relative or absolute
 1420 URI in the HTTP response bodies.

1421 If URIs are specified as relative URIs, they shall be relative to the `baseURI`.

1422 The algorithm used for converting a relative URI to an absolute URI shall be as described in section 5.2 of
 1423 [RFC3986](#). Table 3 illustrates how relative URIs are resolved against base URIs:

1424 **Table 3 – Converting a relative URI to an absolute URI**

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

1425 If relative URIs are used, the `baseURI` shall end with a trailing slash and relative URIs shall not begin
 1426 with a leading slash. This format is consistent with most URI resolve utilities and produces the same
 1427 results as a simple string concatenation algorithm.

1428 If serialized in JSON, these values shall be of JSON type: *string*

1429 If serialized in XML, these values shall be of XML Schema type: *xs:anyURI*

1430 **5.5.11 Array**

1431 An array represents an ordered list of items of the same type. An array shall always appear as an
 1432 attribute of a Resource, and is only accessible as such (it is not a separately addressable Resource). If a
 1433 Resource is deleted, the items in its arrays shall also be deleted. However, in case these items were just
 1434 references to other Resources, these referred Resources are not affected. (See the semantics of
 1435 references in 5.7.)

1436 Attributes that are arrays are defined by using the notation `itemType[]`, where `itemType` is the type
 1437 name for each item of the array. If the type is a structure, not a simple data type, it is recommended as a
 1438 convention in the model that the name of an array be the plural of a name that characterizes each item.
 1439 For example, an array of volume items or of references to these may be named "volumes."

1440 If an attribute is of type of references (`ref[]`) – and more generally array of an atomic type - the
 1441 definition in the model shall include an "Array item name" that may be used in its serialization.

1442 JSON serialization:

1443 Within this specification, arrays in JSON are serialized with a wrapper property. The wrapper name shall
 1444 be same as the attribute name for the array. For example, a "things" attribute of type "thing[]" is
 1445 serialized as:

```
1446 "things" : [  
1447   { ... }, +  
1448 ] ?
```

1449 If the items in the array are structures, the structure name shall not be present in the JSON serialization.

1450 In the case of an array of references, i.e., where the "ref" type applies to each element of the array,
 1451 each element shall simply be serialized as an href property within a JSON array. For example, an array
 1452 "things" of type "ref[]" is serialized as:

```
1453 "things": [  
1454   { "href": string }, +  
1455 ] ?
```

1456 **NOTE** If serializing arrays, conformant implementations shall not include empty arrays (i.e., arrays that contain no
 1457 child properties) in the JSON serialization. Notice that the child of the "things" property is defined with a "+",
 1458 meaning at least one child is required. This requirement ensures that the JSON serialization is minimized and only
 1459 includes the wrapping "things" element if, and only if, there is at least one "thing" in the array.

1460 XML serialization:

1461 The XML serialization of arrays requires each item of the array to be represented as an element. These
 1462 elements shall be consecutive and contiguous in the serialization and the name of each element (tag
 1463 name) shall be the name of the element type (the name that appears before "[]" in the array type). For
 1464 example, a "things" attribute shall be serialized as a list of items named "thing", where "thing" is
 1465 the name of a structure:

```
1466 <thing>  
1467   ...  
1468 </thing> *
```

1469 There is no wrapper element for an array in XML.

1470 In the case of an array of references, i.e., where the "ref" type applies to each element of the array, the
 1471 array is serialized as a list of XML elements without wrapper. Each element is named per the "Array
 1472 item name" value specified in the attribute's definition. For example, an array "things" of type
 1473 "ref[]" where the "Array item name" is "thing" is serialized as:

```
1474 <thing href="xs:anyURI"/> +
```

1475 5.5.12 Collection

1476 Like arrays, Collections are groupings of Resources of the same type. In contrast with arrays, Collections
 1477 are themselves Resources that have their own URI and can be independently accessed. Collections also
 1478 allow for an optimized and convenient interaction pattern by providing a specialized set of operations that
 1479 avoid replacing a large number of items when updating the set, as with arrays.

1480 This specification uses Collections if the set of grouped items is modified often and potentially by multiple
 1481 Consumers. Conversely, arrays are used if it is expected that the list of items is not modified often or can

1482 be easily modified by substitution of the entire list, and thus the overhead of managing these items as
 1483 separate Resources might be unjustified and burdensome.

1484 Collections are also used to represent 1-n relationships between Resources: a Resource that is
 1485 associated with a set of same-typed Resources, shall use a Collection attribute to represent such an
 1486 association, the items of which refer to each one of the Resources of the associated set.

1487 Each element in a Collection is called a Collection item or entry. A Collection item is actually a reference
 1488 to a Resource, not the Resource itself. For convenience, each referred Resource is called here a
 1489 Resource item of the Collection, and these Resource items are still considered being grouped by the
 1490 Collection (instead of just their references). These Resources items are assumed to be of a complex type
 1491 and are separately addressable and manageable. A primary Resource (see definition in Resources
 1492 section) can be an item in more than one Collection. If such a Resource is deleted, all the Collections that
 1493 share this Resource item shall remove their reference to that Resource.

1494 While different Collections contain entries of different Resource types, all Collections follow the pattern
 1495 described below:

- 1496 • A Collection shall contain an `id` attribute that acts as a "self pointer." Retrieving the data at this
 1497 reference shall return the Collection. In the XML representation, each Collection shall be wrapped
 1498 by a `<Collection>` element.
- 1499 • A Collection shall contain a `count` attribute that indicates the number of Resources in the
 1500 Collection at the time the Collection was queried.
- 1501 • Adding new Resources to the Collection shall be done through either the "add" operation defined
 1502 within the Collection (when the Resource is also created) or the "insert" operation (when the
 1503 Resource already exists).
- 1504 • Deleting Resources from the Collection shall be done either through a "delete" operation on the
 1505 Resource itself (if the Resource has to be discarded) or the "remove" Collection operation (if the
 1506 Resource must still exist outside the Collection).
- 1507 • Collections shall be deleted if their owning Resource is deleted.
- 1508 • Unless the Resource items in the collection are secondary Resources (see later), deleting a
 1509 Collection does not cause the deletion of the collected Resources.

1510 Collections that are attributes of other Resources are represented with attribute type
 1511 "collection[itemType]." The Resource type of the Collection items are specified inside the
 1512 brackets; for example an attribute that is a Collection of Machines is expressed as
 1513 "collection[Machine]." Attributes of such types are serialized as a reference to a Collection
 1514 Resource instead of holding the Collection itself as value. For brevity, while these attributes are
 1515 "references" the word "ref" or "reference" does not appear in the model definition tables - simply the type
 1516 "collection[itemType]" appears. Also, the description of the attribute will be directly of the
 1517 Collection it refers to instead of being described as a reference to such a Collection.

1518 **Serialization:**

1519 The serialization of Collections shall adhere to the following pattern:

1520 **JSON serialization:**

```
1521 { "resourceURI": string,  
1522   "id": string,  
1523   "count": number,  
1524   "resourceSpecificGroupingName": [
```

```

1525     { "resourceURI": string,
1526       "id": string,
1527       "name": string, ?
1528       "description": string, ?
1529       "created": string, ?
1530       "updated": string, ?
1531       "properties": { string: string, + }, ?
1532       ... resource specific data ...      "operations": [
1533         { "rel": "edit", "href": string }, ?
1534         { "rel": "delete", "href": string } ?
1535       ] ?
1536       ...
1537     } +
1538 ], ?
1539 "operations": [
1540   { "rel": "add", "href": string } ?
1541   { "rel": "insert", "href": string } ?
1542   { "rel": "remove", "href": string } ?
1543 ]
1544 ...
1545 }

```

1546 XML serialization:

```

1547 <Collection resourceURI="xs:anyURI" xmlns="http://schemas.dmtf.org/cimi/1">
1548   <id> xs:anyURI </id>
1549   <count> xs:integer </count>
1550   <ResourceSpecificElementName>
1551     <id> xs:anyURI </id>
1552     <name> xs:string </name> ?
1553     <description> xs:string </description> ?
1554     <created> xs:dateTime </created> ?
1555     <updated> xs:dateTime </updated> ?
1556     <property key="xs:string"> xs:string </property> *
1557     ... resource specific data ...
1558     <operation rel="edit" href="xs:anyURI"/> ?
1559     <operation rel="delete" href="xs:anyURI"/> ?
1560     <xs:any>*
1561   </ResourceSpecificElementName> *
1562   <operation rel="add" href="xs:anyURI"/> ?
1563   <operation rel="insert" href="xs:anyURI"/> ?
1564   <operation rel="remove" href="xs:anyURI"/> ?

```



```
1565 <xs:any>*
1566 </Collection>
```

1567 Where the `resourceURI` attributes shall contain the Collection or Resource specific URIs for that type
 1568 of Collection, and `resourceSpecificGroupingName` and `ResourceSpecificElementName`
 1569 shall be replaced with the name of the Collection-specific Resource name, e.g., `machines` in JSON or
 1570 `Machine` in XML.

1571 The above serialization shows that each entry in a Collection may contain “resource specific data” beside
 1572 the reference to the Resource item and the common attributes. This placeholder represents two kinds of
 1573 data:

- 1574 a) Optionally some *accessory attributes* that represent accessory information for the use of this
 1575 reference in the context of the Resource owning that Collection (the accessory attributes) – e.g.,
 1576 the “initial location” of a referenced `Volume`, in a Collection of Volumes associated with a
 1577 `Machine`. Accessory attributes – if any - are part of the definition of each specific Collection..
- 1578 b) All or a subset of the attributes of the corresponding Resource items. How much of the
 1579 Resource item is expanded in the serialization of the Collection is controlled by expansion
 1580 mechanisms described later.

1581 If accessory attributes exist for items in a Collection, the “*resourceSpecificGroupingName*” or
 1582 “*ResourceSpecificElementName*” is not just identifying the Resource type of Collection items, but is a
 1583 unique name specific to this combination of accessory attributes and Resource type – e.g., for Volumes
 1584 with initial location, it may be “`locatedVolume`”. Also the `resourceURI` of the Collection is unique to this
 1585 combination. Because of this accessory attribute, the Collection of Volumes is said to be “enhanced”, as
 1586 opposed to “basic” for a Collection without accessory attribute.

1587 The serialization of Collections follows these additional rules:

- 1588 • A Provider may limit the number of Resources returned in the Collection. The Consumer can
 1589 determine this has occurred by comparing the number of returned Resources with the value of
 1590 the “Count” attribute and any Collection subsetting query parameters it specified. In this case,
 1591 the Consumer is advised to specify filter query parameters (see 4.1.6.1) to reduce the number
 1592 of entries returned, or retrieve them in batches by issuing multiple requests with Collection
 1593 subsetting query parameters (see 4.1.6.2)
- 1594 • As with all Resources in the CIMI model, each Resource in the Collection shall have an `id`
 1595 attribute that acts as a “self pointer.” Retrieving the data at this reference shall return just that
 1596 one Resource and not any parent Resource, such as the Collection or array attribute.
- 1597 • The serialization of a Collection may be controlled (see 4.1.6.4 `$expand` query parameter) to
 1598 show more or less of each Resource item. By default, each entry in the Collection will show just
 1599 a reference (URL) to the Resource item, along with the “common” attributes of the Resource
 1600 item. Alternatively, the Resource item may be expanded partially or fully when querying the
 1601 Collection.
- 1602 • As with all arrays, if there are no Resources in the Collection, the serialization of the list shall be
 1603 omitted.

1604 **Examples:**

1605 5.5.12.1.1 Machine Collection

1606 The Resource type for each item of this Collection is “`Machine`”. There is no accessory attributes in this
 1607 Collection, which is then called a “basic” Machine Collection. In the example below, each Machine item in
 1608 the Collection is not expanded except for its common attributes. An expanded serialization showing all or
 1609 parts of each Machine is also an option.

1610 **JSON serialization:**

```

1611 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCollection",
1612   "id": string,
1613   "count": number,
1614   "Machines": [
1615     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
1616       "id": string,
1617       "name": string, ?
1618       "description": string, ?
1619       "created": string, ?
1620       "updated": string, ?
1621       "properties": { string: string, + }, ?
1622       "machine": { "href": string },
1623       "operations": [
1624         { "rel": "edit", "href": string }, ?
1625         { "rel": "delete", "href": string } ?
1626       ] ?
1627       ...
1628     }, +
1629   ], ?
1630   "operations": [
1631     { "rel": "add", "href": string } ?
1632     { "rel": "insert", "href": string } ?
1633     { "rel": "remove", "href": string } ?
1634   ]
1635   ...
1636 }

```

1637 **XML serialization:**

```

1638 <Collection
1639   resourceURI="http://schemas.dmtf.org/cimi/1/MachineCollection"
1640   xmlns="http://schemas.dmtf.org/cimi/1">
1641   <id> xs:anyURI </id>
1642   <count> xs:integer </count>
1643   <Machine>
1644     <id> xs:anyURI </id>
1645     <name> xs:string </name> ?
1646     <description> xs:string </description> ?
1647     <created> xs:dateTime </created> ?
1648     <updated> xs:dateTime </updated> ?
1649     <property key="xs:string"> xs:string </property> *

```

```

1650 <machine href="xs:anyURI"/>
1651 <operation rel="edit" href="xs:anyURI"/> ?
1652 <operation rel="delete" href="xs:anyURI"/> ?
1653 <xs:any>*
1654 </Machine> *
1655 <operation rel="add" href="xs:anyURI"/> ?
1656 <operation rel="insert" href="xs:anyURI"/> ?
1657 <operation rel="remove" href="xs:anyURI"/> ?
1658 <xs:any>*
1659 </Collection>
    
```

1660 5.5.12.1.2 Volume Collection in a Machine

1661 The Resource type for each item of this Collection is “Volume”. Because this Collection is representing
 1662 an association between a Machine and a set of Volumes, the initial location of these Volumes is added as
 1663 an accessory attribute. In the example below, each Volume item in the Collection is not expanded except
 1664 for its common attributes. An expanded serialization showing all or parts of each Volume is also an
 1665 option.

1666 Note that the last part of the Collection resourceURI is not just VolumeCollection but
 1667 locatedVolumeCollection, in order to uniquely identify the combination of {initial location +
 1668 Volume}.

1669 JSON serialization:

```

1670 { "resourceURI": "http://schemas.dmtf.org/cimi/1/locatedVolumeCollection",
1671   "id": string,
1672   "count": number,
1673   "locatedVolumes": [
1674     { "resourceURI": "http://schemas.dmtf.org/cimi/1/locatedVolume",
1675       "id": string,
1676       "name": string, ?
1677       "description": string, ?
1678       "created": string, ?
1679       "updated": string, ?
1680       "properties": { string: string, + }, ?
1681       "initialLocation": string, ?
1682       "volume": { "href": string },
1683       "operations": [
1684         { "rel": "edit", "href": string }, ?
1685         { "rel": "delete", "href": string } ?
1686       ] ?
1687       ...
1688     }, +
1689   ], ?
    
```

```

1690     "operations": [
1691         { "rel": "add", "href": string } ?
1692         { "rel": "insert", "href": string } ?
1693         { "rel": "remove", "href": string } ?
1694     ]
1695     ...
1696 }

```

1697 XML serialization:

```

1698 <Collection
1699     resourceURI="http://schemas.dmtf.org/cimi/1/locatedVolumeCollection"
1700     xmlns="http://schemas.dmtf.org/cimi/1">
1701     <id> xs:anyURI </id>
1702     <count> xs:integer </count>
1703     <locatedVolume>
1704         <id> xs:anyURI </id>
1705         <name> xs:string </name> ?
1706         <description> xs:string </description> ?
1707         <created> xs:dateTime </created> ?
1708         <updated> xs:dateTime </updated> ?
1709         <property key="xs:string"> xs:string </property> *
1710         <initialLocation> xs:string </initialLocation> ?
1711         <volume href="xs:anyURI"/>
1712         <operation rel="edit" href="xs:anyURI"/> ?
1713         <operation rel="delete" href="xs:anyURI"/> ?
1714         <xs:any>*
1715     </locatedVolume> *
1716     <operation rel="add" href="xs:anyURI"/> ?
1717     <operation rel="insert" href="xs:anyURI"/> ?
1718     <operation rel="remove" href="xs:anyURI"/> ?
1719     <xs:any>*
1720 </Collection>

```

1721 5.5.12.2 Adding items to Collections

1722 Invoking the "add" operation of a Collection shall create and add a new Resource to the Collection. The
 1723 contents of the request body shall be either a representation of the new Resource being added to the
 1724 Collection, or a representation of the Template associated with the new Resource being created and
 1725 resource specific data attributes.

1726 If the Collection is a Collection of primary Resources, then the Resource shall be added also to the CEP
 1727 Collection that collects these Resources.

1728 If creating a new Resource the "add" operation shall contain:

- 1729 • The "common attributes" as defined by clause 5.7.2
- 1730 • The Resource specific data needed to create it. This data shall either be a reference to the
1731 Resource-specific Template Resource or the Resource-specific Template Resource itself
1732 inlined.
- 1733 • Accessory attributes—if any—that represent accessory information for the use of the reference in
1734 the context of the Resource owning that Collection (the associative attributes)
- 1735 • In the XML case, a wrapper element (named after the pattern <ResourceNameCreate>)

1736 For example, to create a new Machine (which requires the use of a Template) and add it to the
1737 MachineCollection, the "add" operation of the MachineCollection shall be serialized as
1738 follows:

1739 **JSON serialization:**

```
1740 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate", ?
1741   "name": string, ?
1742   "description": string, ?
1743   "properties": { string: string, + }, ?
1744   "machineTemplate": { "href": string ?}
1745   ...
1746 }
```

1747 **XML serialization:**

```
1748 <MachineCreate xmlns="http://schemas.dmtf.org/cimi/1">
1749   <name> xs:string </name> ?
1750   <description> xs:string </description> ?
1751   <property key="xs:string"> xs:string </property> *
1752   <machineTemplate href="xs:anyURI"? />
1753   <xs:any>*
1754 </MachineCreate>
```

1755 The MachineCollection has a new Machine:

1756 **JSON serialization:**

```
1757 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
1758   "id": string,
1759   "name": string,
1760   ...
1761 }
```

1762 **XML serialization:**

```
1763 <Machine xmlns="http://schemas.dmtf.org/cimi/1">
1764   <id> xs:anyURI </id>
1765   <name> xs:string </name>
1766   ...
```

1767 `</Machine>`

1768 The processing of the "add" operation shall adhere to the semantics defined in clause 4.2.1.1.

1769 Regardless of whether a Template is used, the "add" operation shall create the new Resource and add it
1770 to the Collection and a reference (URI) to the new entry shall be returned in the response message in the
1771 HTTP Location header.

1772 5.5.12.3 Inserting items in Collections

1773 Invoking the "insert" operation of a Collection shall add to the Collection a new reference to an existing
1774 Resource. The contents of the request body shall specify the URL of the existing Resource being added.

1775 In order to add an existing `Volume` to the `volumes` Collection of a `Machine`, the request body of the
1776 "insert" operation shall be serialized as follows:

1777 JSON serialization:

```
1778 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Volume",
1779   "initialLocation": string,
1780   "volume": { "href": string }
1781 }
```

1782 XML serialization:

```
1783 <Volume xmlns="http://schemas.dmtf.org/cimi/1">
1784   <initialLocation> xs:string </initialLocation>
1785   <volume href="xs:string"/>
1786 </Volume>
```

1787 Note that "initialLocation" is an accessory attributes to each reference of `Volume`. The definition of the
1788 `volumes` Collection of the `Machine` Resource describes the accessory attribute(s) for this Collection.

1789 5.5.12.4 Removing items to Collections

1790 Invoking the "remove" operation of a Collection shall delete the corresponding reference entry in the
1791 Collection, along with accessory attributes if any. The contents of the request body shall be the URL of
1792 the Resource being removed.

1793 In order to remove a `Volume` from the `volumes` Collection of a `Machine`, the request body of the
1794 "remove" operation shall be serialized as follows:

1795 JSON serialization:

```
1796 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Volume",
1797   "volume": { "href": string }
1798 }
```

1799 XML serialization:

```
1800 <Volume xmlns="http://schemas.dmtf.org/cimi/1">
1801   <volume href="xs:string"/>
1802 </Volume>
```

1803 Removing the referenced Resource (here a `Volume`) deletes the related entry from the Collection. This
1804 deletes the reference but not the Resource itself.

1805 Deleting the referenced Resource via a DELETE operation on the Resource itself (here a `Volume`) also
 1806 deletes the related entry from the Collections that reference this Resource – i.e., it has the effect of a
 1807 “remove” on the Collection, in addition to deleting the referenced Resource.

1808 5.5.13 "Any" type

1809 Some attributes are polymorphic and can hold various data types, the list of which is indicated in their
 1810 description. In such cases, the type of the attribute shall be indicated as "any" in the model
 1811 representation.

1812 5.5.14 valueScope

1813 The valueScope type is a specialized map. Its goal is to define possible values for a list of attributes of a
 1814 Resource. The possible values for an attribute are called the “value scope” of the attribute, and a
 1815 combination of attribute value scopes (in form of a map) in a Resource or in the ResourceMetadata is
 1816 called the value scope of the Resource.

1817 Each item in a valueScope is a key/value pair where:

- 1818 • The key is the name of an attribute of a Resource – or “**scoped attribute**” – for which a set of
 1819 possible values is defined.
- 1820 • The value is a structure that defines the “**scope**”, i.e., a range, an enumeration or a single
 1821 assigned value for the scoped attribute.

1822 The scope structure:

1823 A “scope” structure – or the value part of a key-value item in a valueScope – can take one of four forms:

- 1824 1) An assigned single value, along with its (optional) `units`, e.g., for a scoped attribute named
 1825 “cpu”:

```
1826 "cpu": { "value": 2000, "units": "megahertz" }
```

- 1827 2) A range of values, along with its optional `units`, and an optional `increment` e.g., for a
 1828 scoped attribute named "memory". The range may be open-ended: either the `minimum` or the
 1829 `maximum` may be missing. The `increment` specifies the allowed values starting from the
 1830 `minimum` and upward - i.e., the allowed values are of the form: `minimum+N*(increment)`,
 1831 where $N \geq 0$, or starting from the `maximum` and downward in case there is no `minimum`, i.e.,
 1832 allowed values are of the form: `maximum-N*(increment)`.

```
1833 "memory": { "minimum": 4000, "maximum": 10000, "units": "kibibytes", "default":  
1834 4000, "increment": 2000 }
```

- 1835 3) An enumeration (or values), along with its (optional) `units`, e.g., for a scoped attribute
 1836 named “cpuArch”:

```
1837 "cpuArch": { "values": [ "68000", "Alpha", "ARM", " PA_RISC"], "default": "PA_RISC"  
1838 }
```

- 1839 4) Simply a required `units`, e.g., for a scoped attribute named “capacity”:

```
1840 "capacity": { "units": "megabytes" }
```

1842 If a valueScope is associated with a Resource type, it shall be in form of an attribute named “`vscope`”, of
 1843 type array of valueScope (i.e., `valueScope[]`).

1844 An example of valueScope for the MachineConfiguration Resource:
1845

```
1846     "vscope" : [ {
1847         "cpu": { "value": 1 },
1848         "memory": { "minimum": 4, "maximum": 32, "units": "GbB", "default": 4, "increment":
1849         2 },
1850         "cpuArch": { "values": [ "68000", "Alpha", "ARM", " PA_RISC", "i5"], "default":
1851         "i5" }
1852     } ]
```

1853 Semantics

1854 A value scope may be defined either for the attributes of a Resource type described in
1855 ResourceMetadata, or for attribute(s) of a particular Resource, or for both. The semantics is as follows:

- 1856 • If a value scope is associated with a Resource (i.e., this Resource has a “vscope” attribute), a
1857 scoped attribute of this Resource shall only take values and units within its scope, when
1858 updated or when set (if it were not set at creation time).
- 1859 • If a value scope is associated with a Resource type as described in ResourceMetadata (i.e., the
1860 ResourceMetadata instance for this Resource type has a “vscope” attribute), any Resource of
1861 this type shall have its attributes take values within the defined scope.
- 1862 • If both a Resource and its related ResourceMetadata have some value scope associated with
1863 them, then the value scope of the Resource should be defined so that any attribute value within
1864 this value scope is also within the value scope of its related ResourceMetadata (i.e., the value
1865 scope of the Resource attribute is included in the value scope of the ResourceMetadata for this
1866 attribute if any. The actual value scope of an attribute that is scoped both in its Resource and in
1867 its ResourceMetadata , is the intersection of the two value scopes.

1868 The semantics of a value scope for Consumer and Provider is as follows:

- 1869 • If an attribute of a Resource is scoped, a Consumer shall set a value (creation or update
1870 request) compatible with the value scope of this attribute, including constraints specified by an
1871 increment if it is present.
- 1872 • For any other case where the Consumer sets an incompatible value, the Provider shall return a
1873 4xx error code.

1874 Usage in a template

1875 When defined in a template Resource, or a Resource used in a template (e.g., MachineConfiguration),
1876 the value scope is intended to restrict also the similar attributes in Resources generated from this
1877 template. In such a case, the attributes of the generated Resource that were scoped in the template of
1878 this Resource, are also scoped similarly in the generated Resource. In order to make this scope more
1879 explicit, a Provider should replicate in the generated Resource the value scope – or the relevant part of it
1880 – defined in the template.

1881 In order to better enforce the value scope of Resources, a Provider may predefine a set of templates that
1882 a Consumer may use. This Provider may prevent the Consumer from creating additional templates while
1883 letting the Consumer modify (within scope) the attributes of the predefined templates.

1884 For example, a Provider may create a set of predefined MachineConfiguration Resources with a read-
1885 only *vscope* attribute. The Provider may further prevent Consumers from creating new
1886 MachineConfiguration instances – or only by offering a “copy” operation on existing ones. In this way, the
1887 Provider effectively constrains the Consumer to only use the predefined MachineConfiguration Resources
1888 yet allows the Consumer to modify the configuration attributes within the value scope of each predefined
1889 MachineConfiguration.

1890 Semantics of valueScope array in a Resource

1891 The value scope of a Resource shall be represented by an array of valueScope instances, even if in
 1892 many cases this array will contain a single valueScope instance. This allows for expressing dependencies
 1893 between values of different attributes of a same Resource. In such cases, the scoped attributes of the
 1894 Resource must satisfy either valueScope instance in this array.

1895 In the following example, vscope is an array of two valueScope items:

```
1896 "vscope": [ {
1897   "cpuSpeed": { "minimum": 2, "maximum": 4, "units": "GHz", "default": 2.5},
1898   "memory": { "minimum": 2000000, "maximum": 10000000, "units": "KbB", "increment":
1899   2000000 },
1900   "cpuArch": { "value": "i5" }
1901 }, {
1902   "memory": { "minimum": 4000000, "maximum": 32000000, "units": "KbB" },
1903   "cpuArch": { "values": [ "68000", "Alpha", " PA_RISC" ] }
1904 } ]
```

1905 This valueScope means that the Provider supports MachineConfigurations with either cpuArch of value
 1906 i5, or of a value that is one of { "68000", "Alpha", " PA_RISC" }. In the first case (i5), the
 1907 memory must be within the 2GbB-10GbB range and cpuSpeed must be between 2-4 GHz, while in the
 1908 second case the memory must be within the 4GbB-32GbB range.

1909 The following pseudo-schemas describe the serialization of the valueScope map in both JSON and XML:

1910 JSON serialization:

```
1911 ( "value": any,
1912  "units": string ? ) |
1913 ( "values": [ any,+ ],
1914  "units": string ,?
1915  "default": string ? ) |
1916 ( "minimum": number, ?
1917  "maximum": number, ?
1918  "units": string ,?
1919  "default": number, ?
1920  "increment": number ? )
```

1922 XML serialization:

```
1923 ( <value> xs:any </value>
1924  <units> xs:string </units> ? ) |
1925 ( <value> xs:any </value> +
1926  <units> xs:string </units> ?
1927  <default> xs:any </default> ? ) |
1928 ( <minimum> xs:integer </minimum> ?
1929  <maximum> xs:integer </maximum> ?
1930  <units> xs:string </units> ?
1931  <default> xs:integer </default> ?
1932  <increment> xs:integer </increment> ? )
```

1933 A Provider who supports value scopes shall set the ValueScopes capability (ResourceMetadata) to “true”.

1934 5.5.15 Empty attribute values

1935 Attributes of the following types are omitted in cases where they have an empty value: string, map, array,
1936 and Collection. Apart from being “Provider optional” or “Consumer optional”, an empty value is the third
1937 reason that the serialization schema contains an ‘?’ or an ‘*’ for an attribute.

1938 Other attribute types do not have empty values and shall not be omitted from the serialization for this
1939 reason.

1940 5.6 Units

1941 Some of the Resources defined by this specification have attributes that describe an amount of
1942 something that belongs to, or is associated with, that Resource. For example, the `Machine` Resource
1943 has a `memory` attribute that describes "the size of the memory allocated to this machine." The allowable
1944 units of these attributes are listed in Table 4. Their meaning is defined in [IEC 80000-13:2008](#). Their
1945 numerical equivalents are provided here for convenience:

1946 **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

1947 5.7 Resources

1948 5.7.1 Primary and secondary Resources

1949 The Resources described by this document are classified either as primary or as secondary Resources.

- 1950 • A primary Resource is one that is listed in the Cloud Entry Point, i.e., there is a Collection in the
1951 `CloudEntryPoint` Resource that uses this Resource type for its items. The
1952 `CloudEntryPoint` itself is a primary CIMI Resource.
- 1953 • All other Resources (including Collection Resources) shall be secondary CIMI resources. A
1954 secondary Resource is always dependent on another Resource (typically a primary Resource)
1955 that “owns” it. Deletion of an owning Resource (primary or secondary) causes the deletion of
1956 the owned secondary Resource.

1957 For example, `Machine` is a primary CIMI resource as the `CloudEntryPoint` has a Collection with
1958 `Machine` as its element type. However, for example, `Disk` is a secondary CIMI resource because the
1959 `CloudEntryPoint` does not have a Collection with `Disk` as its element type. A `Disk` does not exist
1960 separately from a `Machine`, and is automatically deleted when the `Machine` is deleted.

1961 5.7.2 Common attributes

1962 Resources share the following common attributes; see Table 5. There are different requirements for
1963 primary and secondary CIMI resources.

Table 5 – Common attributes

Attribute	Type	Description												
id	<i>URI</i>	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	<i>string</i>	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	<i>string</i>	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	<i>dateTime</i>	The timestamp when this Resource was created. The format should be unambiguous, and the value is immutable . 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. The initial value is the time the resource is created. 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" data-bbox="532 1373 1214 1656"> <thead> <tr> <th>Name</th> <th colspan="2"><i>property</i></th> </tr> <tr> <th>Data</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	<i>property</i>		Data	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	<i>property</i>													
Data	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												

Attribute	Type	Description
vscope	<i>valueScope[]</i>	<p>A value scope for this Resource. When the Resource is a template or used in a template, the value scope constrains the similar attributes in generated Resources and is replicated (or its relevant subset) in the generated Resources. This attribute is only defined for primary Resources.</p> <p>Constraints for primary Resources: Provider: support optional; mutable Consumer: support optional; read-only</p>

1965 The following pseudo-schemas describe the serialization of these attributes in both JSON and XML:

1966 **JSON serialization:**

```

1967 "id": string,
1968 "name": string, ?
1969 "description": string, ?
1970 "created": string, ?
1971 "updated": string, ?
1972 "properties": { string: string, + }, ?
1973 "vscope" : [ valueScope, * ], ?
    
```

1974 **XML serialization:**

```

1975 <id> xs:anyURI </id>
1976 <name> xs:string </name> ?
1977 <description> xs:string </description> ?
1978 <created> xs:dateTime </created> ?
1979 <updated> xs:dateTime </updated> ?
1980 <property key="xs:string"> xs:string </property> *
1981 <vscope> valueScope </vscope> *
    
```

1982 **5.8 Operations**

1983 All Resource operations defined by this specification are optional for Providers to support. Consumers, by
 1984 the way of examination of a Resource's ResourceMetadata, can determine which operations are
 1985 supported. However, even for those operations that are supported Consumers still need to examine each
 1986 Resource's representation to determine which operations are supported at that moment. Whether an
 1987 operation is supported is based on a number of factors, including the state of the Resource and access
 1988 control rights of the Consumer. Also see clause 4.2. Operations and states are coupled; i.e., if
 1989 implementing a state-changing Resource operation defined in this specification, the corresponding
 1990 state(s) shall also be implemented. See the Resource-specific "Operations" clauses for additional detail.

1991 The "State" attribute of Resources that have this attribute shall only change value if

- 1992 • an operation is performed on this Resource and this operation requires a state change, or
- 1993 • an error occurred, in this case the "State" attribute shall obtain the value "ERROR".

1994 For example, for a 'start' operation on a Machine both the STARTING and the STARTED states are
 1995 required to be supported by the Machine, while the Machine can only leave the STARTED state after
 1996 another state changing operation is requested, unless an error occurs.

1997 Providers can define additional operations and states. Such extensions shall fall into one of these
 1998 categories:

- 1999 a) A new operation that starts from a CIMI-defined state, or leads to a CIMI-defined state, or both.
 2000 In the latter case, if a CIMI-defined operation already exists for this transition between two
 2001 CIMI-defined states, it shall also be supported by the Provider in addition to the new operation.
- 2002 b) A new Resource state. In that case, a new operation that leads to that state shall also be
 2003 created. In other words, a Provider-defined operation has to be performed before a
 2004 Provider-defined state can be reached.
- 2005 c) A new operation that transitions between two Provider-defined states.

2006 5.9 Alternative model formats

2007 It is expected that this specification is implemented by using a variety of technologies. As a convenience,
 2008 the definition of the model elements are provided in alternative formats that are easily consumable by
 2009 technology-specific tooling.

2010 This model is also available in a CIM/MOF format [\[DSP0259\]](#).

2011 In the event of inconsistencies between the various formats, the normative text within this specification
 2012 takes precedence over the XML Schemas and alternative formats, which in turn take precedence over
 2013 examples.

2014 5.10 Relationship semantics between Resources

2015 5.10.1 Referencing across Resources

2016 Resources may refer each other. This referencing expresses a directional relationship in which there is a
 2017 *referring* Resource and a *referred* Resource. Depending on the cardinality of such relationships, there are
 2018 two representations:

- 2019 • For 1-to-1 referencing, the URL of the referred Resource appears as an attribute in the referring
 2020 Resource.
- 2021 • For 1-to-n referencing, the referred Resources (all of the same type) are grouped in a
 2022 Collection, the URL of which appears as an attribute in the referring Resource. In that case, the
 2023 *referring* Resource does not refer directly to the referred Resources, but instead to a Collection
 2024 Resource that contains references to the *referred* Resources.

2025 If a *referred* Resource is deleted but not the *referring* Resource(s), then in case of a 1-to-1 relationship
 2026 the reference shall be set to empty in every *referring* Resource, and in case of a 1-to-n relationship the
 2027 reference shall be remove from any Collection where it appears as an item.

2028 5.10.2 Component Resources

2029 The reference relationship from one Resource to another (either 1-1 or 1-n) may have the semantics of a
 2030 "composition" (or whole-part relationship in UML), also called "ownership". A Resource that is a
 2031 component of another Resource is "owned" by this Resource, and is subject to the same access
 2032 conditions from a Consumer. Deleting a Resource causes the deletion of all its components.

2033 Composition interferes with the quality of a Resource – secondary or primary – in the following way:

- 2034 • Secondary Resources: a secondary Resource is always a component of at least one primary
 2035 Resource. Secondary Resources can only own secondary Resources.
- 2036 • Primary Resources: a primary Resource may be a component of one or more primary
 2037 Resources, but never of a secondary Resource.

2038 A reference from primary Resource to secondary Resource shall have composition semantics by default.
 2039 The composition semantics of a reference between primary Resources shall be explicitly indicated in the
 2040 definition of the *referring* Resource.

2041 Note that the composition relationship is transitive: a component of a component of a Resource is also a
 2042 component of this Resource. It is also possible for a Resource to be owned by two or more Resources
 2043 that are not component of each other, meaning that either owning Resource, when deleted, will also
 2044 delete the component Resource. The references to this deleted component must then be removed from
 2045 the remaining owners.

2046 **5.10.3 Associated Resources**

2047 A reference between two primary Resources may have the semantics of a simple "association". In
 2048 contrast with a component relationship, the referred Resource is not affected if deleting the referring
 2049 Resource (i.e., the Delete operation is a "shallow delete" by default).

2050 Note that in the case of a 1-n association, deleting the *referring* Resource shall delete the Collection
 2051 Resource that is mediating the relationship but not the *referred* Resource items themselves.

2052 A reference from primary Resource to primary Resource shall have association semantics by default. If it
 2053 has composition semantics this shall be explicitly indicated in the definition of the *referring* Resource.

2054 Unless specified otherwise, the same Resource can be referred to by more than one *referring* Resource.

2055 **5.11 Resource metadata**

2056 Implementations of this specification should allow for Consumers to discover the metadata associated
 2057 with each supported Resource type, for a given Cloud Entry Point. Doing so allows for the discovery of
 2058 Provider-defined constraints on the CIMI defined attributes as well as discovery of any new extension
 2059 attributes or operations that the Provider may have defined. A *ResourceMetadata* instance contains
 2060 metadata describing a particular Resource type – e.g., *Network*, or *Machine* – including any Provider-
 2061 specific capabilities or features. The mechanism by which this metadata is made available is protocol
 2062 specific.

2063 Note that while this specification declares the *ResourceMetadata* as mutable attributes, it is
 2064 expected that only administrative users associated with the Provider will update them. Consequently they
 2065 remain read-only for Consumers.

2066 Each Resource's metadata shall contain the following pieces of information:

2067 **Table 6 – ResourceMetadata attributes**

Name	ResourceMetadata	
Type URI	http://schemas.dmtf.org/cimi/1/ResourceMetadata	
Attribute	Type	Description
id	<i>URI</i>	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	<i>URI</i>	A unique URI associated with, and denoting, the described Resource type. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
name	<i>string</i>	The name of the described Resource type. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

Name	ResourceMetadata																			
Type URI	http://schemas.dmtf.org/cimi/1/ResourceMetadata																			
Attribute	Type	Description																		
attributes	<i>attribute[]</i>	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 border="1"> <tr> <td>Name</td> <td colspan="2"><i>attribute</i></td> </tr> <tr> <td>Data</td> <td>Type</td> <td>Description</td> </tr> <tr> <td>name</td> <td><i>string</i></td> <td>The name of the attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td> </tr> <tr> <td>namespace</td> <td><i>URI</i></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><i>string</i></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><i>boolean</i></td> <td>Indicates whether this Resource requires this attribute to be present. If absent the implied value is "false." Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td> </tr> </table>	Name	<i>attribute</i>		Data	Type	Description	name	<i>string</i>	The name of the attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	namespace	<i>URI</i>	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	<i>string</i>	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	<i>boolean</i>	Indicates whether this Resource requires this attribute to be present. If absent the implied value is "false." Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
		Name	<i>attribute</i>																	
		Data	Type	Description																
		name	<i>string</i>	The name of the attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																
		namespace	<i>URI</i>	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	<i>string</i>	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	<i>boolean</i>	Indicates whether this Resource requires this attribute to be present. If absent the implied value is "false." Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
Constraints: Provider: support optional; mutable Consumer: support optional; read-write																				
vscope	<i>valueScope[]</i>	The <i>vscope</i> attribute may be present on a <i>ResourceMetadata</i> Resource. In that case, the value scope represented by this attribute does not apply to the attributes of the <i>ResourceMetadata</i> Resource itself, but instead to the attributes of the described Resource, i.e., it is a value scope that applies to all Resources of the type identified by the <i>typeURI</i> attribute. Consequently this value scope is about the list of attributes described in the <i>attributes</i> attribute. Constraints: Provider: support optional; mutable Consumer: support optional; read-write																		
capabilities	<i>capability[]</i>	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 border="1"> <tr> <td>Name</td> <td colspan="2"><i>capability</i></td> </tr> <tr> <td>Data</td> <td>Type</td> <td>Description</td> </tr> <tr> <td>name</td> <td><i>string</i></td> <td>The name of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</td> </tr> <tr> <td>uri</td> <td><i>URI</i></td> <td>A URI that uniquely identifies the capability at a global level. Constraints: Provider: support mandatory; mutable</td> </tr> </table>	Name	<i>capability</i>		Data	Type	Description	name	<i>string</i>	The name of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write	uri	<i>URI</i>	A URI that uniquely identifies the capability at a global level. Constraints: Provider: support mandatory; mutable						
		Name	<i>capability</i>																	
		Data	Type	Description																
name	<i>string</i>	The name of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write																		
uri	<i>URI</i>	A URI that uniquely identifies the capability at a global level. Constraints: Provider: support mandatory; mutable																		
Constraints: Provider: support mandatory; mutable																				

Name	ResourceMetadata																										
Type URI	http://schemas.dmtf.org/cimi/1/ResourceMetadata																										
Attribute	Type	Description																									
			<p>Consumer: support mandatory; read-write</p>																								
		description	<p><i>string</i></p> <p>The human-readable description of the semantic of the capability.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p>																								
		value	<p><i>any</i></p> <p>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.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>																								
		<p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>																									
actions	<i>action[]</i>	<p>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.</p> <p>Each operation shall contain the following nested data:</p> <table border="1"> <thead> <tr> <th>Name</th> <th colspan="2"><i>action</i></th> </tr> <tr> <th>Data</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td><i>string</i></td> <td> The name of the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write </td> </tr> <tr> <td>uri</td> <td><i>URI</i></td> <td> A URI that uniquely identifies the operation at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write </td> </tr> <tr> <td>description</td> <td><i>string</i></td> <td> The human-readable description of the semantic of the operation. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write </td> </tr> <tr> <td>method</td> <td><i>string</i></td> <td> The protocol-dependent verb to use to perform the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write </td> </tr> <tr> <td>inputMessage</td> <td><i>string</i></td> <td> 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 </td> </tr> <tr> <td>outputMessage</td> <td><i>string</i></td> <td> 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 </td> </tr> </tbody> </table> <p>Constraints:</p>		Name	<i>action</i>		Data	Type	Description	name	<i>string</i>	The name of the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	uri	<i>URI</i>	A URI that uniquely identifies the operation at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	description	<i>string</i>	The human-readable description of the semantic of the operation. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write	method	<i>string</i>	The protocol-dependent verb to use to perform the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	inputMessage	<i>string</i>	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	<i>string</i>	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
Name	<i>action</i>																										
Data	Type	Description																									
name	<i>string</i>	The name of the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																									
uri	<i>URI</i>	A URI that uniquely identifies the operation at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																									
description	<i>string</i>	The human-readable description of the semantic of the operation. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write																									
method	<i>string</i>	The protocol-dependent verb to use to perform the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																									
inputMessage	<i>string</i>	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	<i>string</i>	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																									

Name	ResourceMetadata	
Type URI	http://schemas.dmtf.org/cimi/1/ResourceMetadata	
Attribute	Type	Description
		Provider: support optional; mutable Consumer: support optional; read-write

2068 When implementing or using `ResourceMetadata`, Providers and Consumers shall adhere to the
 2069 syntax and semantics of its attributes as described in Table 6 as well as in the tables describing
 2070 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
 2071 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 2072 Resource in both JSON and XML:

2073 **JSON media type:** application/json

2074 **JSON serialization:**

```

2075 { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadata",
2076   "id": string,
2077   "typeURI": string,
2078   "name": string,
2079   "attributes" : [
2080     { "name": string,
2081       "namespace": string, ?
2082       "type": string, ?
2083       "required": boolean, ? } *
2084     ], ?
2085   "vscope" : [ valueScope, * ], ?
2086   "capabilities": [
2087     { "name": string, ?
2088       "uri": string,
2089       "description": string, ?
2090       "value": any } *
2091   ], ?
2092   "actions" : [
2093     { "name": string,
2094       "uri": string,
2095       "description": string, ?
2096       "method": string,
2097       "inputMessage": string, ?
2098       "outputMessage": string ? }, *
2099   ], ?
2100   "operations": [
2101     { "rel": "edit", "href": string }, ?
2102     { "rel": "delete", "href": string } ?
2103   ] ?
    
```

2104 ...
 2105 }

2106 **XML media type:** application/xml

2107 **XML serialization:**

```

2108     <ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/1">
2109         <id> xs:anyURI </id>
2110         <name> xs:string </name>
2111         <typeURI> xs:anyURI </typeURI>
2112         <attribute name="xs:string" namespace="xs:anyURI"? type="xs:string"?
2113             required="xs:boolean"? /> *
2114         </attribute> *
2115         <vscope> valueScope </vscope> *
2116         <capability name="xs:string"? uri="xs:anyURI" description="xs:string"?
2117             xs:any*
2118         </capability> *
2119         <action name="xs:string" uri="xs:anyURI" description="xs:string"?
2120             method="xs:string" inputMessage="xs:string"?
2121             outputMessage="xs:string"? /> *
2122         <operation rel="edit" href="xs:anyURI"/> ?
2123         <operation rel="delete" href="xs:anyURI"/> ?
2124         <xs:any>*
2125     </ResourceMetadata>
  
```

2126 Additional metadata about the Resource or attributes may be included by the Provider.

2127 5.11.1 Capabilities

2128 Table 7 describes the capability URIs defined by this specification. Providers may define new URIs and it
 2129 is recommended that these URIs be dereferencable such that Consumers can discover the details of the
 2130 new capability. The "Resource Name" column contains the name of the Resource that may contain the
 2131 specified capability within its `ResourceMetadata`. The "Capability Name" column contains the name
 2132 of the specified capability and shall be unique within the scope of the corresponding Resource. Each
 2133 capability's URI shall be constructed by appending the "Resource Name", a slash (/), and the "Capability
 2134 Name" to "http://schemas.dmtf.org/cimi/1/capability/". For example, the Machine's "InitialState"
 2135 capability shall have a URI of:

2136 `http://schemas.dmtf.org/cimi/1/capability/Machine/InitialState`

2137 Capabilities that apply to the Provider in general, and are not specific to any one Resource, shall be
 2138 associated with the `CloudEntryPoint` Resource (in case a capability applies only to the
 2139 `CloudEntryPoint` Resource itself, its definition indicates this).

2140 Each one of these capabilities may be set to some value, or may be absent. The meaning of an absent
 2141 capability is defined as follows:

- 2142 • For boolean-valued capabilities: same as a "false" value.

- 2143
- 2144
- 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.

2145

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.
CloudEntryPoint	FormatParameter	If true, the Provider shall support the <code>\$format</code> query parameter.
CloudEntryPoint	OrderByParameter	If true, the Provider shall support the <code>\$orderby</code> query parameter.
CloudEntryPoint	QueryPathNotation	If true, the Provider shall support the use of path-like notation with query parameter <code>\$select</code> (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.
CloudEntryPoint	ValueScopes	If true, the Provider shall support the use of attributes of type <code>valueScope</code> , for any primary Resource.
System	SystemComponentTemplateByValue	If true, the Provider shall support the specification of <code>ComponentTemplates</code> by value in <code>SystemTemplates</code> .
Machine	DefaultInitialState	If this capability is set, unless otherwise provided (e.g., by a <code>MachineTemplate</code> "initialState" attribute), the Provider shall set a new <code>Machine</code> to this state value, assuming the value is compatible with the <code>InitialStates</code> capability, if set.
Machine	InitialStates	If this capability is set, and if using a <code>MachineTemplate</code> that has an "initialState" attribute, a Consumer shall use an <code>initialState</code> value from the set of values of this capability.
Machine	MachineConfigByValue	If true, the Provider shall support specifying <code>MachineConfigurations</code> by value. If true, the <code>MachineTemplateByValue</code> shall also have the value true.
Machine	MachineCredentialByValue	If true, the Provider shall support specifying <code>Credentials</code> by value in <code>Machine</code> create operations. If true, the <code>MachineTemplateByValue</code> capability shall also have the value true.
Machine	MachineImageByValue	If true, the Provider shall support specifying <code>MachineImages</code> by value in <code>Machine</code> create operations. If true, the <code>MachineTemplateByValue</code> capability shall also have the value true.
Machine	MachineVolumeTemplatesByValue	If true, the Provider shall support specifying <code>VolumeTemplates</code> by value in <code>Machine</code> create operations. If, then the <code>MachineTemplateByValue</code> capability shall also have the value true.
Machine	MachineTemplateByValue	If true, the Provider shall support specifying <code>MachineTemplates</code> by value in <code>Machine</code> create operations.
Machine	MachineStopForce	If true, the Provider shall support the "force" option on the stop and restart operations on <code>Machines</code> .

Resource Name	Capability Name	Description
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 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.

Resource Name	Capability Name	Description
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.

2146 The following examples show the ResourceMetadata for a Machine that advertises some of its
 2147 capabilities:

2148 **JSON serialization:**

```

2149 { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadata",
2150   "id": "http://example.com/types/Machine",
2151   "typeURI": "http://schemas.dmtf.org/cimi/1/Machine",
2152   "name": "Machine",
2153   "capabilities": [
2154     { "uri":
2155       "http://schemas.dmtf.org/cimi/1/capability/Machine/MachineConfigByValue",
2156       "value": true },
2157     { "uri":
2158       "http://schemas.dmtf.org/cimi/1/capability/Machine/MachineImageByValue",
2159       "value": true },
2160     { "uri":
2161       "http://schemas.dmtf.org/cimi/1/capability/Machine/DefaultInitialState",
2162       "value": "STARTED" }
2163   ]
2164 }
```

2165 **XML serialization:**

```

2166 <ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/1">
2167   <id> http://example.org/types/Machine </id>
2168   <typeURI> http://schemas.dmtf.org/cimi/1/Machine </typeURI>
2169   <name> Machine </name>
2170   <capability
2171     uri="http://schemas.dmtf.org/cimi/1/capability/Machine/MachineConfigByValue">
2172     true
2173   </capability>
```

```

2174     <capability
2175     uri="http://schemas.dmtf.org/cimi/1/capability/Machine/MachineImageByValue">
2176         true
2177     </capability>
2178     <capability
2179     uri="http://schemas.dmtf.org/cimi/1/capability/Machine/DefaultInitialState">
2180         STARTED
2181     </capability>
2182 </ResourceMetadata>

```

2183 5.11.2 ResourceMetadataCollection Resource

2184 A ResourceMetadataCollection Resource represents the Collection of ResourceMetadata
 2185 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. Note that
 2186 modifications of the Resources within this Collection are typically reserved for administrator types of CIMI
 2187 Consumers. This Resource shall be serialized as follows:

2188 JSON serialization:

```

2189 { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadataCollection",
2190   "id": string,
2191   "count": number,
2192   "resourceMetadatas": [
2193     { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadata",
2194       "id": string,
2195       ... remaining ResourceMetadata attributes ...
2196     }, +
2197   ], ?
2198   "operations": [ { "rel": "add", "href": string } ? ]
2199   ...
2200 }

```

2201 XML serialization:

```

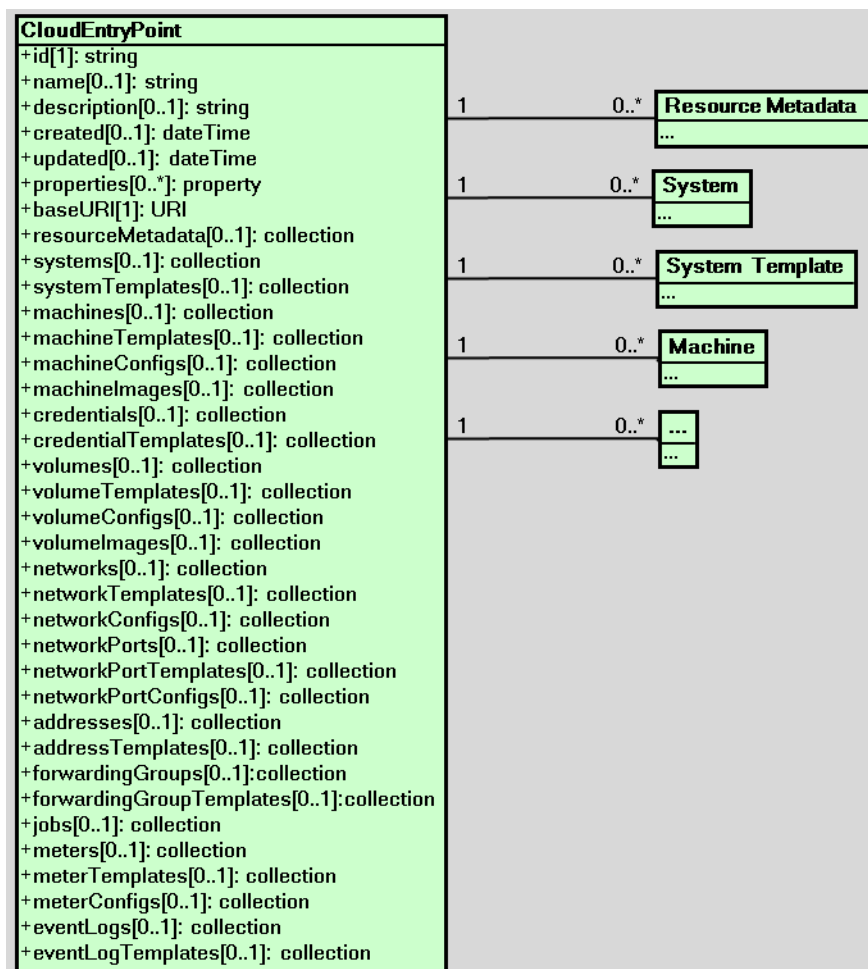
2202 <Collection
2203     resourceURI="http://schemas.dmtf.org/cimi/1/ResourceMetadataCollection"
2204     xmlns="http://schemas.dmtf.org/cimi/1">
2205     <id> xs:anyURI </id>
2206     <count> xs:integer </count>
2207     <ResourceMetadata>
2208         <id> xs:anyURI </id>
2209         ... remaining ResourceMetadata attributes ...
2210     </ResourceMetadata> *
2211     <operation rel="add" href="xs:anyURI"/> ?
2212     <xs:any>*
2213 </Collection>

```

2214 **5.12 Cloud Entry Point**

2215 The Cloud Entry Point (`CloudEntryPoint` Resource) represents the entry point into the cloud defined
 2216 by the CIMI Model. The Cloud Entry Point implements a catalog of Resources, such as `Systems`,
 2217 `SystemTemplates`, `Machines`, `MachineTemplates`, etc., that can be queried and browsed by
 2218 the Consumer.

2219 Figure 1 illustrates the `CloudEntryPoint` and its relationship to other Resources. Although this
 2220 drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor
 2221 normative.



2222

2223 **Figure 1 - Cloud Entry Point**

2224 If a Consumer issues a read on the `CloudEntryPoint` Resource, the Provider shall return a
 2225 `CloudEntryPoint` Resource that only catalogs Resources on which this Consumer is allowed to
 2226 perform operations. Table 8 describes the attributes for the `CloudEntryPoint` Resource.

2227 The relationships from the `CloudEntryPoint` Resource to all Resources in its Collections has a
 2228 composition semantics. Unless indicated otherwise, deleting the `CloudEntryPoint` Resource is also
 2229 deleting all referred Resources.

Table 8 – CloudEntryPoint attributes

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

Name	CloudEntryPoint	
Type URI	http://www.dmf.org/cimi/CloudEntryPoint	
Attribute	Type	Description
		Consumer: support optional; read-only
volumeTemplates	<i>collection</i> [Volume Template]	A reference to the VolumeTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumeConfigs	<i>collection</i> [Volume Configuration]	A reference to the VolumeConfigurationCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumelImages	<i>collection</i> [Volume Image]	A reference to the VolumeImageCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networks	<i>collection</i> [Network]	A reference to the NetworkCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkTemplates	<i>collection</i> [Network Template]	A reference to the NetworkTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkConfigs	<i>collection</i> [Network Configuration]	A reference to the NetworkConfigurationCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkPorts	<i>collection</i> [NetworkPort]	A reference to the NetworkPortCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkPortTemplates	<i>collection</i> [NetworkPort Template]	A reference to the NetworkPortTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkPortConfigs	<i>collection</i> [NetworkPort Configuration]	A reference to the NetworkPortConfigurationCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
addresses	<i>collection</i> [Address]	A reference to the AddressCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
addressTemplates	<i>collection</i> [Address Template]	A reference to the AddressTemplateCollection 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
forwardingGroups	<i>collection</i> [Forwarding Group]	A reference to the ForwardingGroupCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
forwardingGroupTemplates	<i>collection</i> [Forwarding Group Template]	A reference to the ForwardingGroupTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
jobs	<i>collection</i> [Job]	A reference to the JobsCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meters	<i>collection</i> [Meter]	A reference to the MeterCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meterTemplates	<i>collection</i> [Meter Template]	A reference to the MeterTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meterConfigs	<i>collection</i> [Meter Configuration]	A reference to the MeterConfigurationCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLogs	<i>collection</i> [EventLog]	A reference to the EventLogCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLogTemplates	<i>collection</i> [EventLog Template]	A reference to the EventLogTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

2231 Each of the Collections mentioned in Table 8 are defined within the related Resource definition clauses.
 2232 For example, the MachineCollection Resource is defined in clause 5.14.2 as part of the
 2233 Machine-related Resources.

2234 When implementing or using CloudEntryPoint, Providers and Consumers shall adhere to the syntax
 2235 and semantics of its attributes as described in Table 8 as well as in the tables describing embedded
 2236 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 2237 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 2238 both JSON and XML:

2239 **JSON media type:** application/json

2240 **JSON serialization:**

```
2241 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
2242   "id": string,
2243   "name": string, ?
```

```

2244 "description": string, ?
2245 "created": string, ?
2246 "updated": string, ?
2247 "properties": { string: string, + }, ?
2248 "baseURI": string,
2249 "resourceMetadata": { "href": string }, ?
2250 "systems": { "href": string }, ?
2251 "systemTemplates": { "href": string }, ?
2252 "machines": { "href": string }, ?
2253 "machineTemplates": { "href": string }, ?
2254 "machineConfigs": { "href": string }, ?
2255 "machineImages": { "href": string }, ?
2256 "credentials": { "href" string }, ?
2257 "credentialTemplates": { "href" string }, ?
2258 "volumes": { "href": string }, ?
2259 "volumeTemplates": { "href": string }, ?
2260 "volumeConfigs": { "href": string }, ?
2261 "volumeImages": { "href": string }, ?
2262 "networks": { "href": string }, ?
2263 "networkTemplates": { "href": string }, ?
2264 "networkConfigs": { "href": string }, ?
2265 "networkPorts": { "href": string }, ?
2266 "networkPortTemplates": { "href": string }, ?
2267 "networkPortConfigs": { "href": string }, ?
2268 "addresses": { "href": string }, ?
2269 "addressTemplates": { "href": string }, ?
2270 "forwardingGroups" { "href": string }, ?
2271 "forwardingGroupTemplates" { "href": string }, ?
2272 "jobs": { "href": string }, ?
2273 "meters": { "href": string }, ?
2274 "meterTemplates": { "href": string }, ?
2275 "meterConfigs": { "href": string }, ?
2276 "eventLogs": { "href": string }, ?
2277 "eventLogTemplates": { "href": string }, ?
2278 "operations": [
2279     { "rel": "edit", "href": string } ?
2280 ] ?
2281 ...
2282 }

```

2283 **XML media type:** application/xml

2284 **XML serialization:**

```

2285 <CloudEntryPoint xmlns="http://schemas.dmtf.org/cimi/1">
2286   <id> xs:anyURI </id>
2287   <name> xs:string </name> ?
2288   <description> xs:string </description> ?
2289   <created> xs:dateTime </created> ?
2290   <updated> xs:dateTime </updated> ?
2291   <property key="xs:string"> xs:string </property> *
2292   <baseURI> xs:anyURI </baseURI>
2293   <resourceMetadata href="xs:anyURI"/> ?
2294   <systems href="xs:anyURI"/> ?
2295   <systemTemplates href="xs:anyURI"/> ?
2296   <machines href="xs:anyURI"/> ?
2297   <machineTemplates href="xs:anyURI"/> ?
2298   <machineConfigs href="xs:anyURI"/> ?
2299   <machineImages href="xs:anyURI"/> ?
2300   <credentials href="xs:anyURI"/> ?
2301   <credentialTemplates href="xs:anyURI"/> ?
2302   <volumes href="xs:anyURI"/> ?
2303   <volumeTemplates href="xs:anyURI"/> ?
2304   <volumeConfigs href="xs:anyURI"/> ?
2305   <volumeImages href="xs:anyURI"/> ?
2306   <networks href="xs:anyURI"/> ?
2307   <networkTemplates href="xs:anyURI"/> ?
2308   <networkConfigs href="xs:anyURI"/> ?
2309   <networkPorts href="xs:anyURI"/> ?
2310   <networkPortTemplates href="xs:anyURI"/> ?
2311   <networkPortConfigs href="xs:anyURI"/> ?
2312   <addresses href="xs:anyURI"/> ?
2313   <addressTemplates href="xs:anyURI"/> ?
2314   <forwardingGroups href="xs:anyURI"/> ?
2315   <forwardingGroupTemplates href="xs:anyURI"/> ?
2316   <jobs href="xs:anyURI"/> ?
2317   <meters href="xs:anyURI"/> ?
2318   <meterTemplates href="xs:anyURI"/> ?
2319   <meterConfigs href="xs:anyURI"/> ?
2320   <eventLogs href="xs:anyURI"/> ?
2321   <eventLogTemplates href="xs:anyURI"/> ?
2322   <operation rel="edit" href="xs:anyURI"/> ?

```

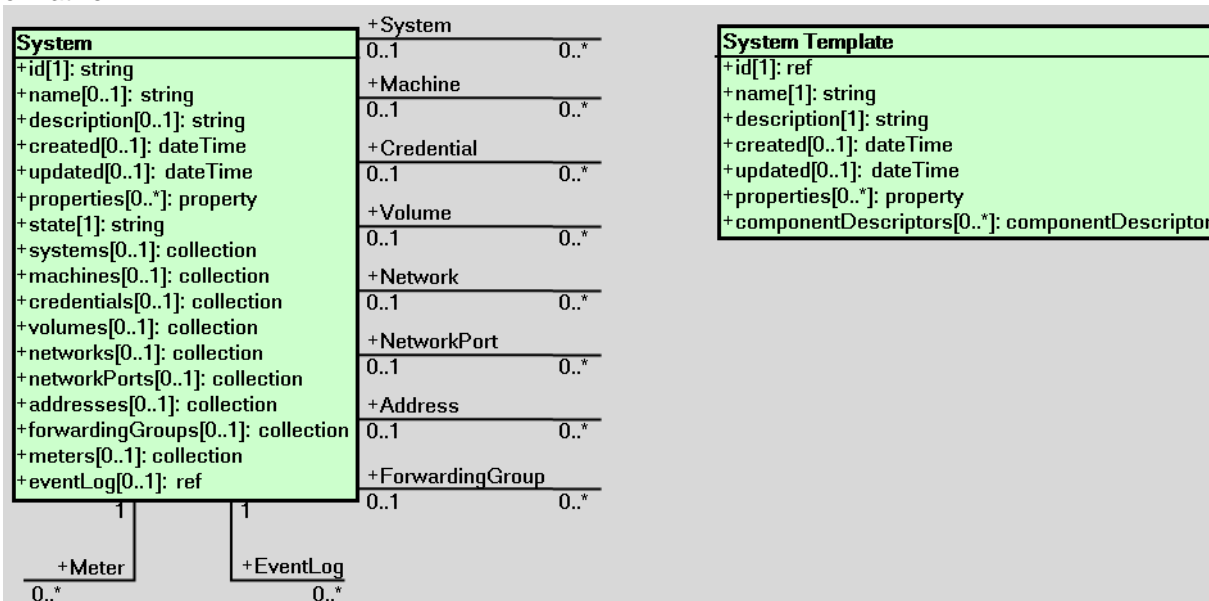
```
2323 <xs:any>*
2324 </CloudEntryPoint>
```

2325 **5.12.1 Operations**

2326 This Resource supports the Read and Update operations.

2327 **5.13 System Resources and relationships**

2328 Figure 2 illustrates the Resources involved in constructing a `System` and their relationships. Although
 2329 this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor
 2330 normative.



2331 **Figure 2 - System Resources**

2332 **5.13.1 System**

2333 A `System` is a realized Resource that consists of one or more `Networks`, `Volumes`, `Machines`,
 2334 (and others) that could be connected and associated with each other. A `System` can be created from the
 2335 interpretation of a `SystemTemplate`. A `System` can be operated and managed as a single Resource
 2336 and usually forms a stack of service. For example, a shopping cart system consists of machines for web
 2337 servers and databases, network addresses for public access, and volumes for database files. A `System`
 2338 may directly provide a user-facing component, or may provide an infrastructure component.

2339 A `System` has several "top-level" attributes that are Collections of references to Resources that are
 2340 components of (owned by) the `System`. A Resource that is owned by a `System` has its life cycle
 2341 directly tied to the life cycle of the `System`. In particular, if a `System` is deleted, all of its owned
 2342 Resources shall also be deleted. Generally, operations on a `System` translate into operations on its
 2343 owned Resources.

2344 However, a Resource owned by a `System` may in turn refer to some other Resources that are not
 2345 owned by this `System`, e.g., a `Machine` in a `System` can refer to a `Volume` that is not listed in the
 2346 `volumes` Collection of the `System`. Instead, this `Volume` is simply associated with this `Machine` with no
 2347 component semantics. Consequently it is not owned by this `System`. More precisely, the following rules
 2348 apply:

- 2349 • By default, all Resources that are created as the result of a System creation are also owned by
- 2350 the *System*. (This rule can be overridden by removal of a Resource from the top-level
- 2351 *System* Collection attributes.)

- 2352 • Ownership of a Resource by a *System* is expressed by including the reference to the
- 2353 Resource in the appropriate top-level System Collection attribute, or by the transitive property of
- 2354 the ownership relationship across layers of components.

- 2355 A Resource shall not be owned by more than one *System* at any point in time (unless there is an
- 2356 ownership relationship between these *Systems*). Note that a Resource does not need to owned by a
- 2357 *System* (i.e., part of any of its Collection attributes) to be referenced/used by a Resource in the
- 2358 *System*. Table 9 describes the System attributes.

2359 **Table 9 – System attributes**

bg	System	
Type URI	http://schemas.dmtf.org/cimi/1/System	
Attribute	Type	Description
state	<i>string</i>	<p>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 <i>System</i> shall be in one of these states if all the <i>Machines</i> referenced by the <i>System</i> are in that state. See clause 5.14.1 for the list of available actions based on the state of a <i>Machine</i>. Such transitional states may just indicate that all <i>Machines</i> in a <i>System</i> are undergoing the same operation (e.g., “start”), without the <i>System</i> being actually operated on (e.g., no “start” done at <i>System</i> level). An actual operation on a <i>System</i> may be traced by querying the “job” entity. MIXED: The <i>System</i> shall be in this state if either no <i>Machines</i> are referenced by this <i>System</i> or <i>Machines</i> referenced by this <i>System</i> are in varying states. Such varying states are likely to occur when an operation is in progress on a <i>System</i>, resulting in transitions of its <i>Machine</i> states toward a new common state (e.g., STOPPED, STARTED) but at a different pace, or sequentially one after the other. DELETING: The <i>System</i> 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</p>
systems	<i>collection [System]</i>	<p>A list of references to nested <i>Systems</i> owned by this <i>System</i>. Adding an item (of type <i>System</i>) to this list is logically equivalent to associating the referenced <i>System</i> to this <i>System</i> with a component semantics. Removing an item from this list is logically equivalent to de-associating the referenced <i>System</i> from this <i>System</i>, i.e., it is no longer a component of this <i>System</i>. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</p>
machines	<i>collection [Machine]</i>	<p>A list of references to <i>Machines</i> owned by this <i>System</i>. Adding an item (of type <i>Machine</i>) to this list is logically equivalent to associating the <i>Machine</i> to this <i>System</i> with a component semantics. Removing an item from this list is logically equivalent to de-associating the <i>Machine</i> from this <i>System</i>. it is no longer a component of this <i>System</i> (unless there is another chain of component relationships to this <i>Machine</i> – e.g., via a sub-System).</p>

bg	System	
Type URI	http://schemas.dmtf.org/cimi/1/System	
Attribute	Type	Description
		Constraints: Provider: support optional; mutable Consumer: support optional; read-only
credentials	<i>collection</i> <i>[Credential]</i>	A list of references to <i>Credentials</i> owned by this <i>System</i> . Adding an item (of type <i>Credential</i>) to this list is logically equivalent to associating the <i>Credential</i> to this <i>System</i> with a component semantics. Removing an item from this list is logically equivalent to de-associating the <i>Credential</i> from this <i>System</i> , i.e., it is no longer a component of this <i>System</i> (unless there is another chain of component relationships to this <i>Credential</i> – e.g., via a sub-System). Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumes	<i>collection</i> <i>[Volume]</i>	A list of references <i>Volumes</i> owned by this <i>System</i> . Adding an item (of type <i>Volume</i>) to this list is logically equivalent to associating the <i>Volume</i> to this <i>System</i> with a component semantics. Removing an item from this list is logically equivalent to de-associating the <i>Volume</i> from this <i>System</i> . it is no longer a component of this <i>System</i> (unless there is another chain of component relationships to this <i>Volume</i> – e.g., via a sub-System). Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networks	<i>collection</i> <i>[Network]</i>	A list of references <i>Networks</i> owned by this <i>System</i> . Adding an item (of type <i>Network</i>) to this list is logically equivalent to associating the <i>Network</i> to this <i>System</i> with a component semantics. Removing an item from this list is logically equivalent to de-associating the <i>Network</i> from this <i>System</i> , i.e., it is no longer a component of this <i>System</i> (unless there is another chain of component relationships to this <i>Network</i> – e.g., via a sub-System).. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkPorts	<i>collection</i> <i>[NetworkPort]</i>	A list of references <i>NetworkPorts</i> owned by this <i>System</i> . Adding an item (of type <i>NetworkPort</i>) to this list is logically equivalent to associating the <i>NetworkPort</i> to this <i>System</i> with a component semantics. Removing an item from this list is logically equivalent to de-associating the <i>NetworkPort</i> from this <i>System</i> , i.e., it is no longer a component of this <i>System</i> (unless there is another chain of component relationships to this <i>NetworkPort</i> – e.g., via a sub-System). Constraints: Provider: support optional; mutable Consumer: support optional; read-only
addresses	<i>collection</i> <i>[Address]</i>	A list of references <i>Addresses</i> owned by this <i>System</i> . Adding an item (of type <i>Address</i>) to this list is logically equivalent to associating the <i>Address</i> to this <i>System</i> with a component semantics. Removing an item from this list is logically equivalent to de-associating the <i>Address</i> from this <i>System</i> , i.e., it is no longer a component of this <i>System</i> (unless there is another chain of component relationships to this <i>Address</i> – e.g., via a sub-System). Constraints: Provider: support optional; mutable Consumer: support optional; read-only
forwardingGroups	<i>collection</i> <i>[Forwarding]</i>	A list of references <i>ForwardingGroups</i> owned by this <i>System</i> . Adding an item (of type <i>ForwardingGroup</i>) to this list is logically equivalent to

bg	System	
Type URI	http://schemas.dmtf.org/cimi/1/System	
Attribute	Type	Description
	<i>Group</i>	associating the ForwardingGroup to this System with a component semantics. Removing an item from this list is logically equivalent to de-associating the ForwardingGroup from this System, i.e., it is no longer a component of this System (unless there is another chain of component relationships to this ForwardingGroup – e.g., via a sub-System). Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meters	<i>collection [Meter]</i>	A list of references to Meters monitored for this System, with component semantics. 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

2360 When implementing or using System, Providers and Consumers shall adhere to the syntax and
 2361 semantics of its attributes as described in Table 9 as well as in the tables describing embedded
 2362 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 2363 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 2364 both JSON and XML.

2365 **JSON media type:** application/json

2366 **JSON serialization:**

```

2367 { "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
2368   "id": string,
2369   "name": string, ?
2370   "description": string, ?
2371   "created": string, ?
2372   "updated": string, ?
2373   "properties": { string: string, + }, ?
2374   "state": string,
2375   "systems": { "href": string }, ?
2376   "machines": { "href": string }, ?
2377   "credentials": { "href": string }, ?
2378   "volumes": { "href": string }, ?
2379   "networks": { "href": string }, ?
2380   "networkPorts": { "href": string }, ?
2381   "addresses": { "href": string }, ?
2382   "forwardingGroups": { "href": string }, ?

```



```

2383     "meters": { "href": string }, ?
2384     "eventLog": { "href": string }, ?
2385     "operations": [
2386         { "rel": "edit", "href": string, ("available": boolean)? }, ?
2387         { "rel": "delete", "href": string, ("available": boolean)? }, ?
2388         { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string,
2389         ("available": boolean)? }, ?
2390         { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string,
2391         ("available": boolean)? }, ?
2392         { "rel": "http://schemas.dmtf.org/cimi/1/action/restart", "href": string,
2393         ("available": boolean)? }, ?
2394         { "rel": "http://schemas.dmtf.org/cimi/1/action/pause", "href": string,
2395         ("available": boolean)? }, ?
2396         { "rel": "http://schemas.dmtf.org/cimi/1/action/suspend", "href": string,
2397         ("available": boolean)? }, ?
2398         { "rel": "http://schemas.dmtf.org/cimi/1/action/export", "href": string,
2399         ("available": boolean)? } ?
2400     ] ?
2401     ...
2402 }
    
```

2403 **XML media type:** application/xml

2404 **XML serialization:**

```

2405 <System xmlns="http://schemas.dmtf.org/cimi/1">
2406   <id> xs:anyURI </id>
2407   <name> xs:string </name> ?
2408   <description> xs:string </description> ?
2409   <created> xs:dateTime </created> ?
2410   <updated> xs:dateTime </updated> ?
2411   <property key="xs:string"> xs:string </property> *
2412   <state> xs:string </state>
2413   <systems href="xs:anyURI"/> ?
2414   <machines href="xs:anyURI"/> ?
2415   <credentials href="xs:anyURI"/> ?
2416   <volumes href="xs:anyURI"/> ?
2417   <networks href="xs:anyURI"/> ?
2418   <networkPorts href="xs:anyURI"/> ?
2419   <addresses href="xs:anyURI"/> ?
2420   <forwardingGroups href="xs:anyURI"/> ?
2421   <meters href="xs:anyURI"/> ?
2422   <eventLog href="xs:anyURI"/> ?
2423   <operation rel="edit" href="xs:anyURI" (available="xs:boolean")? /> ?
2424   <operation rel="delete" href="xs:anyURI" (available="xs:boolean")? /> ?
    
```

```

2425 <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
2426         href="xs:anyURI" (available="xs:boolean"? /> ?
2427 <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
2428         href="xs:anyURI" (available="xs:boolean"? /> ?
2429 <operation rel="http://schemas.dmtf.org/cimi/1/action/restart"
2430         href="xs:anyURI" (available="xs:boolean"? /> ?
2431 <operation rel="http://schemas.dmtf.org/cimi/1/action/pause"
2432         href="xs:anyURI" (available="xs:boolean"? /> ?
2433 <operation rel="http://schemas.dmtf.org/cimi/1/action/suspend"
2434         href="xs:anyURI" (available="xs:boolean"? /> ?
2435 <operation rel="http://schemas.dmtf.org/cimi/1/action/export"
2436         href="xs:anyURI" (available="xs:boolean"? /> ?
2437 <xs:any>*
2438 </System>

```

2439 5.13.1.1 Attributes of type Collection

2440 The following clause describes the Collection Resources owned by Systems.

2441 5.13.1.1.1 systems Collection

2442 The Resource type for each item of this Collection is "System". There is no accessory attribute for the
 2443 items in this Collection, therefore, it is a basic System Collection, the serialization of which follows the
 2444 rules in 5.5.12. See the SystemCollection Resource clause.

2445 5.13.1.1.2 machines Collection

2446 The Resource type for each item of this Collection is "Machine". There is no accessory attribute for the
 2447 items in this Collection, therefore, it is a basic Machine Collection (serialized as described in 5.5.12). See
 2448 the MachineCollection Resource clause.

2449 5.13.1.1.3 credentials Collection

2450 The Resource type for each item of this Collection is "Credential". There is no accessory attribute for
 2451 the items in this Collection, therefore, it is a basic Credential Collection (serialized as described in
 2452 5.5.12). See the CredentialCollection Resource clause.

2453 5.13.1.1.4 volumes Collection

2454 The Resource type for each item of this Collection is "Volume". There is no accessory attribute for the
 2455 items in this Collection, therefore, it is a basic Volume Collection (serialized as described in 5.5.12). See
 2456 the VolumeCollection Resource clause.

2457 5.13.1.1.5 networks Collection

2458 The Resource type for each item of this Collection is "Network". There is no accessory attribute for the
 2459 items in this Collection, therefore, it is a basic Network Collection (serialized as described in 5.5.12).
 2460 See the NetworkCollection Resource clause.

2461 **5.13.1.1.6 networkPorts Collection**

2462 The Resource type for each item of this Collection is "NetworkPort". There is no accessory attribute
 2463 for the items in this Collection, therefore, it is a basic NetworkPort Collection (serialized as described
 2464 in 5.5.12). See the NetworkPortCollection Resource clause.

2465 **5.13.1.1.7 addresses Collection**

2466 The Resource type for each item of this Collection is "Address". There is no accessory attribute for the
 2467 items in this Collection, therefore, it is a basic Address Collection (serialized as described in 5.5.12). See
 2468 the AddressCollection Resource clause.

2469 **5.13.1.1.8 forwardingGroups Collection**

2470 The Resource type for each item of this Collection is "ForwardingGroup". There is no accessory attribute
 2471 for the items in this Collection, therefore, it is a basic ForwardingGroup Collection (serialized as described
 2472 in 5.5.12). See the ForwardingGroupCollection Resource clause.

2473 **5.13.1.1.9 meters Collection**

2474 The Resource type for each item of this Collection is "Meter" as defined in clause 5.17.3. There is no
 2475 accessory attribute for the items in this Collection, therefore it is a basic Meter Collection (serialized as
 2476 described in 5.5.12). See the MeterCollection Resource clause.

2477 **5.13.1.2 Operations**

2478 The System Resource supports the Read, Update, and Delete operations. Create is supported through
 2479 the SystemCollection Resource.

2480 The following custom operations are also defined:

2481 **start/stop/restart/pause/suspend**

2482 **/link@rel:** <http://schemas.dmtf.org/cimi/1/action/xxx>

2483 Where "xxx" is either "start", "stop", "restart", "pause", or "suspend".

2484 This operation shall recursively perform the requested operation on each component of the System
 2485 (Machine or sub-System). Note that not all Machines need to be in the same state for this operation
 2486 to be available and the impact of this operation varies depending on the component's current state; see
 2487 clause 5.14.1.2 for more details about performing operations on Machines. If the operation fails for a
 2488 Machine, that Machine shall not be affected by the operation.

2489 **export**

2490 **/link@rel:** <http://schemas.dmtf.org/cimi/1/action/export>

2491 This operation shall export a System. If an export package exists at that URI, it is updated with the
 2492 values of the System and any component management Resources. Otherwise, a new export package is
 2493 created at that URI with a Media Type as specified by the "format" parameter. Other formats may be used
 2494 if supported, but are not specified by this standard.

2495 Input parameters:

- 2496 1) "format" - type: string - optional
- 2497 Indicates the Media Type of the exported data. If not present, the default value shall be
- 2498 "application/ovf."
- 2499

2500 2) "destination" - type: URI - optional
 2501 Indicates the location to where the exported data is placed. If not present, the HTTP response
 2502 Location header shall contain the URL to the exported data. Based on the specific protocol
 2503 specified within the URI, the Consumer might need to provide additional information (such as
 2504 credentials) in the "properties" field. In the case of HTTP, a PUT shall be used to place the data
 2505 at the specified location.

2506 Output parameters: None.

2507 HTTP protocol

2508 To export a `System`, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/export" URI of the
 2509 `System` where the HTTP request body shall be as described below.

2510 **JSON media type:** application/json

2511 JSON serialization:

```
2512 { "action": "http://schemas.dmtf.org/cimi/1/action/export",
2513   "format": string, ?
2514   "destination": string, ?
2515   "properties": { string: string, + } ?
2516   ...
2517 }
```

2518 **XML media type:** application/xml

2519 XML serialization

```
2520 <Action xmlns="http://schemas.dmtf.org/cimi/1">
2521   <action> http://schemas.dmtf.org/cimi/1/action/export </action>
2522   <format> xs:string </format> ?
2523   <destination> xs:anyURI </destination> ?
2524   <property key="xs:string"> xs:string </property> *
2525   <xs:any>*
2526 </Action>
```

2527 5.13.2 SystemCollection Resource

2528 A `SystemCollection` Resource represents a Collection of `System` Resources and follows the
 2529 Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

2530 JSON serialization:

```
2531 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCollection",
2532   "id": string,
2533   "count", number,
2534   "systems": [
2535     { "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
2536       "id": string,
2537       ... remaining System attributes ...
2538     }, +
```

```

2539 ], ?
2540 "operations": [
2541   { "rel": "add", "href": string }, ?
2542   { "rel": "remove", "href": string } ?
2543 { "rel": "insert", "href": string } ?   { "rel":
2544 "http://schemas.dmtf.org/cimi/1/action/import", "href": string } ?
2545 ]
2546 ...
2547 }
    
```

2548 XML serialization:

```

2549 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/SystemCollection"
2550   xmlns="http://schemas.dmtf.org/cimi/1">
2551   <id> xs:anyURI </id>
2552   <count> xs:integer </count>
2553   <System>
2554     <id> xs:anyURI </id>
2555     ... remaining System attributes ...
2556   </System> *
2557   <operation rel="add" href="xs:anyURI"/> ?
2558   <operation rel="remove" href="xs:anyURI"/> ?
2559   <operation rel="insert" href="xs:anyURI"/> ?
2560   <operation rel="http://schemas.dmtf.org/cimi/1/action/import"
2561 href="xs:anyURI"/> ?
2562   <xs:any>*
2563 </Collection>
    
```

2564 5.13.2.1 Operations

2565 NOTE The "add" operation requires that a SystemTemplate be used (see 4.2.1.1).

2566 Resources created during the process of creating a System shall be "owned" by the System (see
 2567 5.13.1). For example, a componentDescriptor that references a MachineTemplate, and within
 2568 that MachineTemplate is a reference to a VolumeTemplate, results in a reference to the new
 2569 Machine being added to the System.machines attribute and a reference to the new Volume being
 2570 added to the System.volumes attribute. However, if this MachineTemplate refers to an existing
 2571 Volume, this Volume shall not be added to the top-level System attributes.

2572 The following custom operations are also defined:

2573 import

2574 **/link@rel:**http://schemas.dmtf.org/cimi/1/action/import

2575 This operation shall import a System. Not only is a System created, but Machines, Volumes, and
 2576 Networks and possibly recursive Systems and their components may also be created corresponding
 2577 to imported descriptor entries. More detail about this process is in ANNEX A.

2578 1) Input parameters:"source" - type: URI - mandatory
 2579 Indicates the location from which the imported data is retrieved. Based on the specific protocol
 2580 specified within the URI, the Consumer might need to provide additional information (such as
 2581 credentials) in the "properties" field.

2582 Output parameters: None.

2583 HTTP protocol

2584 To import a `System`, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/import" URI of the
 2585 `SystemCollection` where the HTTP request body shall be as described below.

2586 **JSON media type:** application/json

2587 JSON serialization:

```
2588 { "action": "http://schemas.dmtf.org/cimi/1/action/import",
2589   "source": string, ?
2590   "properties": { string: string, + } ?
2591   ...
2592 }
```

2593 **XML media type:** application/xml

2594 XML serialization

```
2595 <Action xmlns="http://schemas.dmtf.org/cimi/1">
2596   <action> http://schemas.dmtf.org/cimi/1/action/import </action>
2597   <source> xs:anyURI </source> ?
2598   <property key="xs:string"> xs:string </property> *
2599   <xs:any>*
2600 </Action>
```

2601 5.13.3 SystemTemplate Resource

2602 The `SystemTemplate` Resource contains the set of individual descriptors that are necessary to create
 2603 the components of a `System`. Each component descriptor can be considered to be the persisted view of
 2604 the create operation that instantiates the component. In practice, the Provider interprets the set of
 2605 component descriptors as a set of creation operations to be executed in an order compatible with the
 2606 dependencies (e.g., attachments or references between components) that are expressed between these
 2607 components.

2608 A `SystemTemplate` may include component references in the descriptors, used to express links
 2609 between components of the resulting `System`. A component reference uses the "name" of the target
 2610 (referred) component. For example, `<volume href="#newVolume"/>` would reference a `Volume`
 2611 named "newVolume." The reference name `–#newVolume –` is replaced by the actual Resource URL in
 2612 the instantiated `System`.

2613 A `SystemTemplate` shall not contain two component descriptors of the same type that would result in
 2614 the same non-null value for the "name" attribute of resulting components. Attempting to create or to
 2615 update a `SystemTemplate` that fails this rule shall result in an error.

2616 Table 10 describes the `SystemTemplate` attributes.

2617

Table 10 – SystemTemplate attributes

Name	SystemTemplate																						
Type URI	http://schemas.dmtf.org/cimi/1/SystemTemplate																						
Attribute	Type	Description																					
component Descriptors	<i>componentDescriptor</i> []	<p>The list of component descriptors describing the components of a <i>System</i> instance realized from this <i>SystemTemplate</i>. For each component descriptor, the corresponding component is created when a <i>System</i> instance is created. Each component descriptor refers to a <i>Template</i> (either by reference or by value), and may also provide additional metadata (name, description, properties). The creation order of components is not specified in <i>SystemTemplate</i>; in particular the order of the component descriptors in this array is not meaningful in terms of creation order.</p> <table border="1"> <thead> <tr> <th>Name</th> <td colspan="2"><i>componentDescriptor</i></td> </tr> <tr> <th>Data</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td><i>string</i></td> <td> <p>The value of the "name" attribute that is associated with a <i>System</i> component created from this component descriptor. Note: This name is not to be confused with the name that may be present in the component <i>Template</i> – e.g., a <i>MachineTemplate</i> – from which this component is instantiated.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p> </td> </tr> <tr> <td>description</td> <td><i>string</i></td> <td> <p>The value of the "description" attribute that is associated with a <i>System</i> component created from this component descriptor.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p> </td> </tr> <tr> <td>properties</td> <td><i>map</i></td> <td> <p>The key/value pairs that is associated with a <i>System</i> component created from this component descriptor.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p> </td> </tr> <tr> <td>type</td> <td><i>URI</i></td> <td> <p>The <i>TypeURI</i> of the component to be created from this component descriptor, e.g., for a <i>Machine</i>: http://schemas.dmtf.org/cimi/1/<i>Machine</i></p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p> </td> </tr> <tr> <td><component Template></td> <td><any></td> <td> <p>A reference either to a component <i>Template</i> or to the <i>Template</i> data itself inlined (i.e., the <i>Template</i> "value"). Note that the exact name of this attribute varies depending on the type of <i>Resource</i> being created, e.g., <i>MachineTemplate</i> for a <i>Machine</i>.</p> <p>This attribute shall contain either:</p> <ul style="list-style-type: none"> • A <i>Template</i> that is provided inline. Such an embedded <i>Template</i> may contain component references, each one of which shall resolve to the <i>URI</i> of a component with same name once created from this <i>SystemTemplate</i>. • A reference to an externally defined <i>Template</i>. Some attribute name/value pairs may be added inside the component <i>Template</i> element to override similar attributes in the referred <i>Template</i> (as described in 4.2.1.1). This example shows how component references can be added to an external <i>Template</i>. <p><i>Example (JSON):</i></p> <pre>"machineTemplate": {</pre> </td> </tr> </tbody> </table>	Name	<i>componentDescriptor</i>		Data	Type	Description	name	<i>string</i>	<p>The value of the "name" attribute that is associated with a <i>System</i> component created from this component descriptor. Note: This name is not to be confused with the name that may be present in the component <i>Template</i> – e.g., a <i>MachineTemplate</i> – from which this component is instantiated.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p>	description	<i>string</i>	<p>The value of the "description" attribute that is associated with a <i>System</i> component created from this component descriptor.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p>	properties	<i>map</i>	<p>The key/value pairs that is associated with a <i>System</i> component created from this component descriptor.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p>	type	<i>URI</i>	<p>The <i>TypeURI</i> of the component to be created from this component descriptor, e.g., for a <i>Machine</i>: http://schemas.dmtf.org/cimi/1/<i>Machine</i></p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>	<component Template>	<any>	<p>A reference either to a component <i>Template</i> or to the <i>Template</i> data itself inlined (i.e., the <i>Template</i> "value"). Note that the exact name of this attribute varies depending on the type of <i>Resource</i> being created, e.g., <i>MachineTemplate</i> for a <i>Machine</i>.</p> <p>This attribute shall contain either:</p> <ul style="list-style-type: none"> • A <i>Template</i> that is provided inline. Such an embedded <i>Template</i> may contain component references, each one of which shall resolve to the <i>URI</i> of a component with same name once created from this <i>SystemTemplate</i>. • A reference to an externally defined <i>Template</i>. Some attribute name/value pairs may be added inside the component <i>Template</i> element to override similar attributes in the referred <i>Template</i> (as described in 4.2.1.1). This example shows how component references can be added to an external <i>Template</i>. <p><i>Example (JSON):</i></p> <pre>"machineTemplate": {</pre>
Name	<i>componentDescriptor</i>																						
Data	Type	Description																					
name	<i>string</i>	<p>The value of the "name" attribute that is associated with a <i>System</i> component created from this component descriptor. Note: This name is not to be confused with the name that may be present in the component <i>Template</i> – e.g., a <i>MachineTemplate</i> – from which this component is instantiated.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p>																					
description	<i>string</i>	<p>The value of the "description" attribute that is associated with a <i>System</i> component created from this component descriptor.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p>																					
properties	<i>map</i>	<p>The key/value pairs that is associated with a <i>System</i> component created from this component descriptor.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p>																					
type	<i>URI</i>	<p>The <i>TypeURI</i> of the component to be created from this component descriptor, e.g., for a <i>Machine</i>: http://schemas.dmtf.org/cimi/1/<i>Machine</i></p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>																					
<component Template>	<any>	<p>A reference either to a component <i>Template</i> or to the <i>Template</i> data itself inlined (i.e., the <i>Template</i> "value"). Note that the exact name of this attribute varies depending on the type of <i>Resource</i> being created, e.g., <i>MachineTemplate</i> for a <i>Machine</i>.</p> <p>This attribute shall contain either:</p> <ul style="list-style-type: none"> • A <i>Template</i> that is provided inline. Such an embedded <i>Template</i> may contain component references, each one of which shall resolve to the <i>URI</i> of a component with same name once created from this <i>SystemTemplate</i>. • A reference to an externally defined <i>Template</i>. Some attribute name/value pairs may be added inside the component <i>Template</i> element to override similar attributes in the referred <i>Template</i> (as described in 4.2.1.1). This example shows how component references can be added to an external <i>Template</i>. <p><i>Example (JSON):</i></p> <pre>"machineTemplate": {</pre>																					

Name	SystemTemplate		
Type URI	http://schemas.dmtf.org/cimi/1/SystemTemplate		
Attribute	Type	Description	
			<pre> "href": "http://example.com/machineTemplates/72000", "credential": { "href": "#MyCredential" } } </pre> <p><i>Note: The "credential" attribute in this example 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></p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>
	quantity	integer	<p>The 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.)</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>
			<p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>
meter Templates	Meter Templates[]	<p>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.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>	
eventLog Template	ref	<p>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.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>	
import Image	ref	<p>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 create this Template.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-only</p>	

2618 When implementing or using `SystemTemplate`, Providers and Consumers shall adhere to the syntax
 2619 and semantics of its attributes as described in Table 10 as well as in the tables describing embedded
 2620 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 2621 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 2622 both JSON and XML

2623 **JSON media type:** application/json

2624 **JSON serialization:**

```

2625 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
2626   "id": string,
2627   "name": string, ?
2628   "description": string, ?
2629   "created": string, ?
2630   "updated": string, ?
2631   "properties": { string: string, + }, ?
2632   "componentDescriptors": [
2633     { "name": string, ?
2634       "description": string, ?
2635       "properties": { string: string, + }, ?
2636       "type": string,
2637       "componentTemplate": {
2638         "href": string, ?
2639         ... ComponentTemplate attributes ... ?
2640       },
2641       "quantity": number ?
2642     }, +
2643   ], ?
2644   "meterTemplates": [
2645     { "href": string, ?
2646       ... MeterTemplate attributes ... ?
2647     }, *
2648   ], ?
2649   "eventLogTemplate": {
2650     "href": string, ?
2651     ... EventLogTemplate attributes ... ?
2652   }, ?
2653   "importImage": { "href": string }, ?
2654
2655   "operations": [
2656     { "rel": "edit", "href": string }, ?
2657     { "rel": "delete", "href": string }, ?

```

```

2658     { "rel": "http://schemas.dmtf.org/cimi/1/action/export", "href": string } ?
2659   ] ?
2660   ...
2661 }

```

2662 **XML media type:** application/xml

2663 **XML serialization:**

```

2664 <SystemTemplate xmlns="http://schemas.dmtf.org/cimi/1">
2665   <id> xs:anyURI </id>
2666   <name> xs:string </name> ?
2667   <description> xs:string </description> ?
2668   <created> xs:dateTime </created> ?
2669   <updated> xs:dateTime </updated> ?
2670   <property key="xs:string"> xs:string </property> *
2671   <componentDescriptor>
2672     <name> xs:string </name> ?
2673     <description> xs:string </description> ?
2674     <property key="xs:string"> xs:string </property> *
2675     <type> xs:anyURI </type>
2676     <componentTemplate href="xs:anyURI"? >
2677       ... ComponentTemplate attributes ... ?
2678     </componentTemplate> *
2679
2680     <quantity> xs:integer </quantity>
2681   </componentDescriptor> *
2682   <meterTemplate href="xs:anyURI"? >
2683     ... MeterTemplate attributes ... ?
2684   </meterTemplate> *
2685   <eventLogTemplate href="xs:anyURI"? >
2686     ... EventLogTemplate attributes ... ?
2687   </eventLogTemplate> ?
2688   <importImage href="xs:anyURI"? >
2689   <operation rel="edit" href="xs:anyURI"/> ?
2690   <operation rel="delete" href="xs:anyURI"/> ?
2691   <operation rel="http://schemas.dmtf.org/cimi/1/action/export"
2692 href="xs:anyURI"/> ?
2693   <xs:any>*
2694 </SystemTemplate>

```

2695 **5.13.3.1 Operations**

2696 This Resource supports the Read, Update, and Delete operations. Create is supported through the
2697 `SystemTemplateCollection` Resource.

2698 The following custom operations are also defined:

2699 **export**

2700 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/export`

2701 This operation shall export a `SystemTemplate`. If an export package exists at that URI, it is updated
2702 with the values of the `SystemTemplate` and any component management Resources. Otherwise a
2703 new export package is created at that URI with a Media Type as specified by the "format" parameter.
2704 Other formats may be used if supported, but are not specified by this standard.

2705 Input parameters:

- 2706 1) "format" - type: string - optional
2707 Indicates the Media Type of the exported data. If not present, the default value shall be
2708 "application/ovf."
- 2709 2) "destination" - type: URI - optional
2710 Indicates the location to where the exported data is placed. If not present, the HTTP response
2711 Location header shall contain the URL to the exported data. Based on the specific protocol
2712 specified within the URI, the Consumer might need to provide additional information (such as
2713 credentials) in the "properties" field. In the case of HTTP, a PUT shall be used to place the data
2714 at the specified location.

2715 Output parameters: None.

2716 **HTTP protocol**

2717 To export a `SystemTemplate`, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/export"
2718 URI of the `SystemTemplate` where the HTTP request body shall be as described below.

2719 **JSON media type:** application/json

2720 **JSON serialization:**

```
2721 { "action": "http://schemas.dmtf.org/cimi/1/action/export",
2722   "format": string, ?
2723   "destination": string, ?
2724   "properties": { string: string, + } ?
2725   ...
2726 }
```

2727 **XML media type:** application/xml

2728 **XML serialization**

```
2729 <Action xmlns="http://schemas.dmtf.org/cimi/1">
2730   <action> http://schemas.dmtf.org/cimi/1/action/export </action>
2731   <format> xs:string </format> ?
2732   <destination> xs:anyURI </destination> ?
2733   <property key="xs:string"> xs:string </property> *
```

```
2734     <xs:any>*
2735 </Action>
```

2736 5.13.4 SystemTemplateCollection Resource

2737 A SystemTemplateCollection Resource represents the Collection of SystemTemplate
2738 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
2739 shall be serialized as follows:

2740 JSON serialization:

```
2741 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplateCollection",
2742   "id": string,
2743   "count": number,
2744   "systemTemplates": [
2745     { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
2746       "id": string,
2747       ... remaining SystemTemplate attributes ...
2748     }, +
2749   ], ?
2750   "operations": [
2751     { "rel": "add", "href": string }, ?
2752     { "rel": "http://schemas.dmtf.org/cimi/1/action/import", "href": string } ?
2753   ]
2754   ...
2755 }
```

2756 XML serialization:

```
2757 <Collection
2758   resourceURI="http://schemas.dmtf.org/cimi/1/SystemTemplateCollection"
2759   xmlns="http://schemas.dmtf.org/cimi/1">
2760   <id> xs:anyURI </id>
2761   <count> xs:integer </count>
2762   <SystemTemplate>
2763     <id> xs:anyURI </id>
2764     ... remaining SystemTemplate attributes ...
2765   </SystemTemplate> *
2766   <operation rel="add" href="xs:anyURI"/> ?
2767   <operation rel="http://schemas.dmtf.org/cimi/1/action/import"
2768 href="xs:anyURI"/> ?
2769   <xs:any>*
2770 </Collection>
```

2771 5.13.4.1 Operations

2772 The following custom operations are defined:

2773 **import**

2774 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/import`

2775 This operation shall import a `SystemTemplate`. Not only is a `SystemTemplate` created, but
 2776 `MachineTemplates`, `VolumeTemplates`, and `NetworkTemplates` and possibly recursive
 2777 `SystemTemplates` and their components may also be created, corresponding to imported descriptor
 2778 entries. More detail about this process is in ANNEX A.

2779 Input parameters:

2780 1) "source" - type: URI - mandatory
 2781 Indicates the location from which the imported data is retrieved. Based on the specific protocol
 2782 specified within the URI, the Consumer might need to provide additional information (such as
 2783 credentials) in the "properties" field.

2784 Output parameters: None.

2785 **HTTP protocol**

2786 To import a `SystemTemplate`, a POST is sent to the "`http://schemas.dmtf.org/cimi/1/action/import`"
 2787 URI of the `SystemTemplateCollection` where the HTTP request body shall be as described
 2788 below.

2789 **JSON media type:** `application/json`

2790 **JSON serialization:**

```
2791 { "action": "http://schemas.dmtf.org/cimi/1/action/import",
2792   "source": string, ?
2793   "properties": { string: string, + } ?
2794   ...
2795 }
```

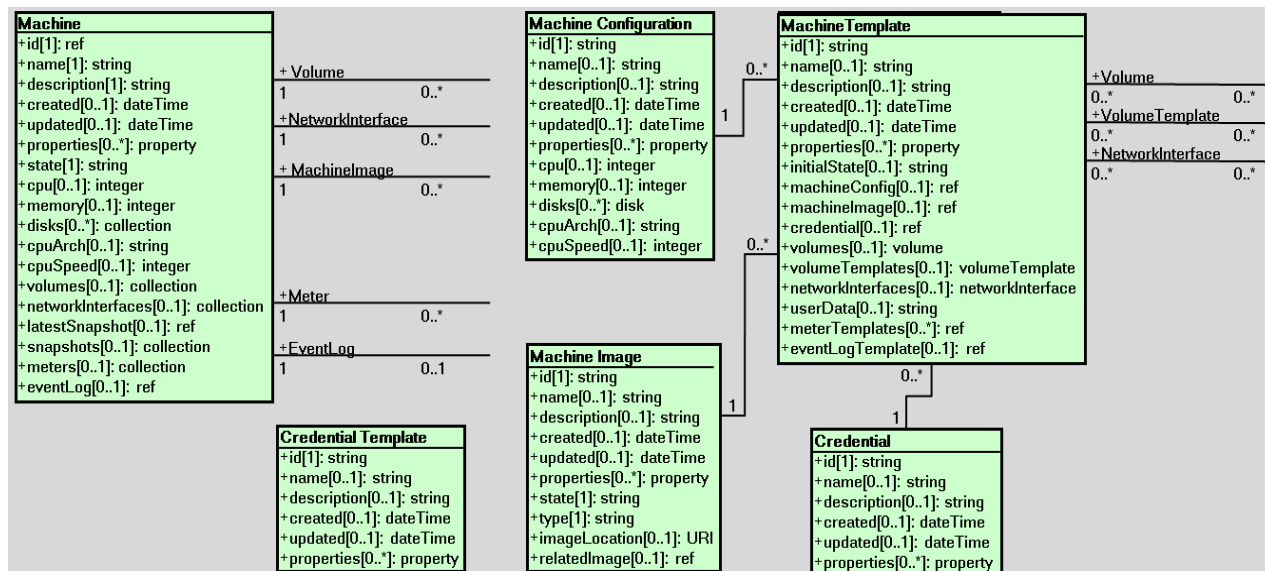
2796 **XML media type:** `application/xml`

2797 **XML serialization**

```
2798 <Action xmlns="http://schemas.dmtf.org/cimi/1">
2799   <action> http://schemas.dmtf.org/cimi/1/action/import </action>
2800   <source> xs:anyURI </source> ?
2801   <property key="xs:string"> xs:string </property> *
2802   <xs:any>*
2803 </Action>
```

2804 5.14 Machine Resources and relationships

2805 Figure 3 illustrates the Resources involved in constructing a `Machine` and their relationships. Although
 2806 this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor
 2807 normative.



2808 **Figure 3 - Machine Resources**

2809 **5.14.1 Machine**

2810 An instantiated compute Resource that encapsulates both CPU and Memory. Table 11 describes the
 2811 Machine attributes.

2812 **Table 11 – 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. CAPTURING: If the Machine is undergoing the "capture" operation its state may be set to "CAPTURING". If some operations that were accepted by the Machine before the capture are no longer available during the capture, the Machine shall be in state "CAPTURING". RESTORING: The Machine is in the process of being restored from a MachineImage. DELETING: The Machine is in the process of being deleted. ERROR: The Provider has detected an error in the Machine.

Name	Machine	
Type URI	http://schemas.dmtf.org/cimi/1/Machine	
Attribute	Type	Description
		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	<i>integer</i>	The amount of CPU that this Machine has. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
memory	<i>integer</i>	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	<i>collection [Disk]</i>	A reference to the list of disks (local storage) that are part of the Machine. Adding an element to this list creates a disk. The Disk Resource is a secondary Resource with component semantics w/r to the Machine. 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	<i>string</i>	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	<i>integer</i>	The approximate CPU speed of this Machine - in megahertz. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
volumes	<i>collection [located Volume]</i>	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: . This Collection has the semantics of an association between the Machine and Volumes (deleting the Machine does not cause the deletion of the referred Volumes). It is defined in clause Error! Reference source not found. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkInterfaces	<i>collection [Network Interface]</i>	A reference to the list of NetworkInterfaces on this Machine. The NetworkInterface Resource is a secondary Resource with component semantics with regard to the Machine. Each NetworkInterface instance represents an association between the Machine and a Network. It is defined in clause 5.14.1.1.3. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
latestSnapshot	<i>ref</i>	A reference to the SNAPSHOT representing the latest state captured for this Machine (either most recent Snapshot or the last Snapshot reverted to).

Name	Machine	
Type URI	http://schemas.dmtf.org/cimi/1/Machine	
Attribute	Type	Description
		Constraints: Provider: support optional; mutable Consumer: support optional; read-only
snapshots	<i>collection</i> <i>[MachineImage]</i>	A reference to the list of references to the <i>MachineImages</i> of type <i>SNAPSHOT</i> taken of this <i>Machine</i> . This <i>Collection</i> has the semantics of an association between the <i>Machine</i> and <i>SNAPSHOT MachineImages</i> . (The deletion of the <i>Machine</i> does not cause the deletion of the referred <i>Snapshots</i> .) Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meters	<i>collection</i> <i>[Meter]</i>	A reference to the list of <i>Meters</i> monitored for this <i>Machine</i> . Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLog	<i>ref</i>	A reference to the <i>EventLog</i> of this <i>Machine</i> . Constraints: Provider: support optional; mutable Consumer: support optional; read-only

2813 When implementing or using *Machine*, Providers and Consumers shall adhere to the syntax and
 2814 semantics of its attributes as described in Table 11, as well as in the tables describing embedded
 2815 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 2816 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 2817 both JSON and XML:

2818 **JSON media type:** application/json

2819 **JSON serialization:**

```

2820 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
2821   "id": string,
2822   "name": string, ?
2823   "description": string, ?
2824   "created": string, ?
2825   "updated": string, ?
2826   "properties": { string: string, + }, ?
2827   "vscope" : [ valueScope, * ], ?
2828   "state": string,
2829   "cpu": number,
2830   "memory": number,
2831   "disks" : { "href": string }, ?
2832   "cpuArch": string, ?
2833   "cpuSpeed": number, ?
2834   "volumes": { "href": string }, ?
2835   "networkInterfaces": { "href": string }, ?
2836   "latestSnapshot": { "href": string }, ?
2837   "snapshots": { "href": string }, ?
    
```



```

2838     "meters": { "href": string }, ?
2839     "eventLog": { "href": string }, ?
2840     "operations": [
2841         { "rel": "edit", "href": string, ("available": boolean)? }, ?
2842         { "rel": "delete", "href": string, ("available": boolean)? }, ?
2843         { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string,
2844           ("available": boolean)? }, ?
2845         { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string,
2846           ("available": boolean)? }, ?
2847         { "rel": "http://schemas.dmtf.org/cimi/1/action/restart", "href": string,
2848           ("available": boolean)? }, ?
2849         { "rel": "http://schemas.dmtf.org/cimi/1/action/pause", "href": string,
2850           ("available": boolean)? }, ?
2851         { "rel": "http://schemas.dmtf.org/cimi/1/action/suspend", "href": string,
2852           ("available": boolean)? }, ?
2853         { "rel": "http://schemas.dmtf.org/cimi/1/action/snapshot", "href": string,
2854           ("available": boolean)? }, ?
2855         { "rel": "http://schemas.dmtf.org/cimi/1/action/restore", "href": string,
2856           ("available": boolean)? } ?
2857     ]
2858     ...
2859 }
    
```

2860 **XML media type:** application/xml

2861 **XML serialization:**

```

2862 <Machine xmlns="http://schemas.dmtf.org/cimi/1">
2863     <id> xs:anyURI </id>
2864     <name> xs:string </name> ?
2865     <description> xs:string </description> ?
2866     <created> xs:dateTime </created> ?
2867     <updated> xs:dateTime </updated> ?
2868     <property key="xs:string"> xs:string </property> *
2869     <vscope> valueScope </vscope> *
2870     <state> xs:string </state>
2871     <cpu> xs:integer </cpu>
2872     <memory> xs:integer </memory>
2873     <disks href="xs:anyURI"/> ?
2874     <cpuArch> xs:string </cpuArch> ?
2875     <cpuSpeed> xs:integer </cpuSpeed> ?
2876     <volumes href="xs:anyURI"/> ?
2877     <networkInterfaces href="xs:anyURI"/> ?
2878     <latestSnapshot href="xs:anyURI"/> ?
2879     <snapshots href="xs:anyURI"/> ?
    
```

```

2880     <meters href="xs:anyURI"/> ?
2881     <eventLog href="xs:anyURI"/> ?
2882     <operation rel="edit" href="xs:anyURI" (available="xs:boolean")? /> ?
2883     <operation rel="delete" href="xs:anyURI" (available="xs:boolean")? /> ?
2884     <operation rel="http://schemas.dmtf.org/cimi/1/action/start" href="xs:anyURI"
2885 (available="xs:boolean")? /> ?
2886     <operation rel="http://schemas.dmtf.org/cimi/1/action/stop" href="xs:anyURI"
2887 (available="xs:boolean")? /> ?
2888     <operation rel="http://schemas.dmtf.org/cimi/1/action/restart"
2889 href="xs:anyURI" (available="xs:boolean")? /> ?
2890     <operation rel="http://schemas.dmtf.org/cimi/1/action/pause" href="xs:anyURI"
2891 (available="xs:boolean")? /> ?
2892     <operation rel="http://schemas.dmtf.org/cimi/1/action/suspend"
2893 href="xs:anyURI" (available="xs:boolean")? /> ?
2894     <operation rel="http://schemas.dmtf.org/cimi/1/action/capture"
2895 href="xs:anyURI" (available="xs:boolean")? /> ?
2896     <operation rel="http://schemas.dmtf.org/cimi/1/action/snapshot"
2897 href="xs:anyURI" (available="xs:boolean")? /> ?
2898     <operation rel="http://schemas.dmtf.org/cimi/1/action/restore"
2899 href="xs:anyURI" (available="xs:boolean")? /> ?
2900     <xs:any>*
2901 </Machine>
    
```

2902 **5.14.1.1 Collections**

2903 The following clause describes the Collection Resources owned by Machines.

2904 **5.14.1.1.1 Disk Collection**

2905 The Resource type for each item of this Collection is "Disk", defined in Table 12:

2906 **Table 12 – Disk attributes**

Name	Disk	
Type URI	http://schemas.dmtf.org/cimi/1/Disk	
Attribute	Type	Description
capacity	integer	The initial capacity, in kilobytes, of the disk. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
initialLocation	string	Operating System-specific location (path) in its namespace where this disk 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

2907 In the following serializations, the Disk resource is expanded: each item of the Collection shows the Disk
 2908 attributes, not a reference.

2909 **JSON serialization:**

```

2910 { "resourceURI": "http://schemas.dmtf.org/cimi/1/DiskCollection",
    
```

```

2911     "id": string,
2912     "count": number,
2913     "disks": [
2914         { "resourceURI": "http://schemas.dmtf.org/cimi/1/Disk",
2915           "id": string,
2916           "name": string, ?
2917           "description": string, ?
2918           "created": string, ?
2919           "updated": string, ?
2920           "properties": { string: string, + }, ?
2921           "capacity": number,
2922           "initialLocation": string, ?
2923           "operations": [
2924             { "rel": "edit", "href": string }, ?
2925             { "rel": "delete", "href": string } ?
2926           ] ?
2927           ...
2928         }, +
2929     ], ?
2930     "operations": [ { "rel": "add", "href": string } ? ]
2931     ...
2932 }
    
```

2933 **XML serialization:**

```

2934 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/DiskCollection"
2935     xmlns="http://schemas.dmtf.org/cimi/1">
2936     <id> xs:anyURI </id>
2937     <count> xs:integer </count>
2938     <Disk>
2939         <id> xs:anyURI </id>
2940         <name> xs:string </name> ?
2941         <description> xs:string </description> ?
2942         <created> xs:dateTime </created> ?
2943         <updated> xs:dateTime </updated> ?
2944         <property key="xs:string"> xs:string </property> *
2945         <capacity> xs:integer </capacity>
2946         <initialLocation> xs:string </initialLocation> ?
2947         <operation rel="edit" href="xs:anyURI"/> ?
2948         <operation rel="delete" href="xs:anyURI"/> ?
2949         <xs:any>*
2950     </Disk> *
    
```

```

2951     <operation rel="add" href="xs:anyURI"/> ?
2952     <xs:any>*
2953 </Collection>
    
```

2954 **5.14.1.1.2 volumes Collection**

2955 The referred Resource type for each item of this Collection is “Volume”. However because there is an
 2956 accessory attribute (initialLocation), this is not a basic but an enhanced Volume Collection. The name
 2957 “locatedVolume” is used to define the type of each Collection item. The accessory attribute is defined in
 2958 Table 13:

2959 **Table 13 – locatedVolume accessory attributes**

Name	locatedVolume	
Type URI	http://schemas.dmtf.org/cimi/1/locatedVolume	
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: Provider: support optional; immutable Consumer: support optional; read-only

2960 **JSON serialization:**

```

2961 { "resourceURI": "http://schemas.dmtf.org/cimi/1/locatedVolumeCollection",
2962   "id": string,
2963   "count": number,
2964   "locatedVolumes": [
2965     { "resourceURI": "http://schemas.dmtf.org/cimi/1/locatedVolume",
2966       "id": string,
2967       "name": string, ?
2968       "description": string, ?
2969       "created": string, ?
2970       "updated": string, ?
2971       "properties": { string: string, + }, ?
2972       "initialLocation": string, ?
2973       "volume": { "href": string },
2974       "operations": [
2975         { "rel": "edit", "href": string }, ?
2976         { "rel": "delete", "href": string } ?
2977       ] ?
2978       ...
2979     }, +
2980   ], ?
2981   "operations": [
2982     { "rel": "add", "href": string } ?
    
```

```

2983 { "rel": "insert", "href": string } ?
2984 { "rel": "remove", "href": string } ?
2985 ]
2986 ...
2987 }
    
```

XML serialization:

```

2989 <Collection
2990     resourceURI="http://schemas.dmtf.org/cimi/1/locatedVolumeCollection"
2991     xmlns="http://schemas.dmtf.org/cimi/1">
2992   <id> xs:anyURI </id>
2993   <count> xs:integer </count>
2994   <locatedVolume>
2995     <id> xs:anyURI </id>
2996     <name> xs:string </name> ?
2997     <description> xs:string </description> ?
2998     <created> xs:dateTime </created> ?
2999     <updated> xs:dateTime </updated> ?
3000     <property key="xs:string"> xs:string </property> *
3001     <initialLocation> xs:string </initialLocation> ?
3002     <volume href="xs:anyURI"/>
3003     <operation rel="edit" href="xs:anyURI"/> ?
3004     <operation rel="delete" href="xs:anyURI"/> ?
3005     <xs:any>*
3006   </locatedVolume> *
3007   <operation rel="add" href="xs:anyURI"/> ?
3008   <operation rel="insert" href="xs:anyURI"/> ?
3009   <operation rel="remove" href="xs:anyURI"/> ?
3010   <xs:any>*
3011 </Collection>
    
```

5.14.1.1.3 networkInterfaces Collection

The Resource type for each item of this Collection is “NetworkInterface”, defined in Table 14:

Table 14 – NetworkInterface attributes

Name	NetworkInterface	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkInterface	
Attribute	Type	Description
addresses	<i>collection</i> <i>[Address]</i>	A reference to the list of references to the Addresses for this network interface. Note: This Collection represents an association between the NetworkInterface and a list of Addresses. The Address collection type is described in the Address section.. Constraints: Provider: support mandatory; mutable

Name	NetworkInterface	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkInterface	
Attribute	Type	Description
		Consumer: support mandatory; read-only
network	<i>ref</i>	A reference to a <i>Network</i> for this network interface. This reference has association semantics. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
networkPort	<i>ref</i>	A reference to the <i>NetworkPort</i> for this network interface. This reference has association semantics. If this attribute is provided, the "network" attribute in the referenced <i>NetworkPort</i> 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 <i>MachineNetworkInterface</i> . 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: 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

3015 **JSON serialization:**

```

3016 { "resourceURI":
3017     "http://schemas.dmtf.org/cimi/1/NetworkInterfaceCollection",
3018     "id": string,
3019     "count": number,
3020     "networkInterfaces": [
3021         { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkInterface",
3022           "id": string,
3023           "name": string, ?
3024           "description": string, ?
3025           "created": string, ?
3026           "updated": string, ?
3027           "properties": { string: string, + }, ?
3028           "addresses": { "href": string },
3029           "network": { "href": string },
3030           "networkPort": { "href": string }, ?

```

```

3031     "state": string, ?
3032     "macAddress": string, ?
3033     "mtu": number, ?
3034     "operations": [
3035         { "rel": "edit", "href": string }, ?
3036         { "rel": "delete", "href": string } ?
3037     ] ?
3038     ...
3039 }, +
3040 ], ?
3041 "operations": [ { "rel": "add", "href": string } ? ]
3042 ...
3043 }
    
```

3044 XML serialization:

```

3045 <Collection
3046 resourceURI="http://schemas.dmtf.org/cimi/1/NetworkInterfaceCollection"
3047     xmlns="http://schemas.dmtf.org/cimi/1">
3048     <id> xs:anyURI </id>
3049     <count> xs:integer </count>
3050     <NetworkInterface>
3051         <id> xs:anyURI </id>
3052         <name> xs:string </name> ?
3053         <description> xs:string </description> ?
3054         <created> xs:dateTime </created> ?
3055         <updated> xs:dateTime </updated> ?
3056         <property key="xs:string"> xs:string </property> *
3057         <addresses href="xs:anyURI"/>
3058         <network href="xs:anyURI"/>
3059         <networkPort href="xs:anyURI"/> ?
3060         <state> xs:string </state> ?
3061         <macAddress> xs:string </macAddress> ?
3062         <mtu> xs:integer </mtu> ?
3063         <operation rel="edit" href="xs:anyURI"/> ?
3064         <operation rel="delete" href="xs:anyURI"/> ?
3065         <xs:any> *
3066     </NetworkInterface> *
3067     <operation rel="add" href="xs:anyURI"/> ?
3068     <xs:any> *
3069 </Collection>
    
```

3070 **5.14.1.1.4 addresses Collection**

3071 The Resource type for each item of this Collection is "Address". It is a basic Address Collection. Its
3072 serialization is described in the AddressCollection Resource clause.

3073 **5.14.1.1.5 snapshots Collection**

3074 The Resource type for each item of this Collection is "MachineImage". It is a basic MachineImage
3075 Collection. Its serialization is described in the MachineImageCollection Resource clause.

3076 **5.14.1.1.6 meters Collection**

3077 The Resource type for each item of this Collection is "Meter" as defined in clause 5.17.3. There is no
3078 accessory attribute for the items in this Collection, therefore it is a basic Meter Collection (serialized as
3079 described in 5.5.12). See the MeterCollection Resource clause.

3080 **5.14.1.2 Operations**

3081 This Resource supports the Read, Update, and Delete operations. Create is supported through the
3082 MachineCollection Resource.

3083 The following custom operations are also defined:

3084 **start**

3085 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/start`

3086 This operation shall start a Machine.

3087 Input parameters: None.

3088 Output parameters: None.

3089 During the processing of this operation, the Machine shall be in the "STARTING" state.

3090 Upon successful completion of this operation, the Machine shall be in the "STARTED" state.

3091 If a Machine is in the "STOPPED" state, starting it shall be the virtual equivalent of powering on a
3092 physical machine. There is no restored CPU or Memory state, so the guest OS typically performs boot or
3093 installation tasks.

3094 If the Machine was in the "SUSPENDED" or "PAUSED" state, starting it shall have the effect of
3095 resuming it.

3096 **HTTP protocol**

3097 To start a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/start" URI of the
3098 Machine where the HTTP request body shall be as described below.

3099 **JSON media type:** application/json

3100 **JSON serialization:**

```
3101 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3102   "action": "http://schemas.dmtf.org/cimi/1/action/start",
3103   "properties": { string: string, + } ?
3104   ...
3105 }
```


3106 **XML media type:** application/xml

3107 **XML serialization**

```
3108 <Action xmlns="http://schemas.dmtf.org/cimi/1">
3109   <action> http://schemas.dmtf.org/cimi/1/action/start </action>
3110   <property key="xs:string"> xs:string </property> *
3111   <xs:any>*
3112 </Action>
```

3113 Upon successful processing of the request, the HTTP response body may be empty.

3114 **stop**

3115 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/stop

3116 This operation shall stop a *Machine*.

3117 Input parameters:

3118 1) "force" - type: boolean - optional
 3119 A flag to indicate whether the Provider shall simulate a power off condition (force=true) or shall
 3120 simulate a shutdown operation that allows applications to save their state and the file system to
 3121 be made consistent (force=false). Inclusion of this parameter by Consumers is optional and if
 3122 not specified, the Provider may choose either mechanism. Providers are encouraged to
 3123 advertise this choice by the way of the *MachineStopForceDefault* capability.

3124 Output parameters: None.

3125 During the processing of this operation, the *Machine* shall be in the "STOPPING" state.

3126 Upon successful completion of this operation, the *Machine* shall be in the "STOPPED" state. Stopping a
 3127 *Machine* with force=true shall be the virtual equivalent of powering off a physical machine. There is no
 3128 saved CPU or Memory state. Stopping a *Machine* with force=false shall result in a machine with
 3129 consistent file systems.

3130 A Consumer may reissue a stop operation if the state is STOPPING, perhaps with force=true, but
 3131 Providers shall not issue a force=true stop operation on their own.

3132 **HTTP protocol**

3133 To stop a *Machine*, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the
 3134 *Machine* where the HTTP request body shall be as described below.

3135 **JSON media type:** application/json

3136 **JSON serialization:**

```
3137 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3138   "action": "http://schemas.dmtf.org/cimi/1/action/stop",
3139   "force": boolean, ?
3140   "properties": { string: string, + } ?
3141   ...
3142 }
```

3143 **XML media type:** application/xml

3144 **XML serialization**

```
3145 <Action xmlns="http://schemas.dmtf.org/cimi/1">
3146   <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
3147   <force> xs:boolean </force> ?
3148   <property key="xs:string"> xs:string </property> *
3149   <xs:any>*
3150 </Action>
```

3151 Upon successful processing of the request, the HTTP response body may be empty.

3152 **restart**

3153 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/restart

3154 This operation shall restart a *Machine*. If the *Machine* is in the "STARTED" state, this operation shall
3155 have the effect of executing the "stop" and then "start" operations. If the *Machine* is in the "STOPPED"
3156 state, this operation shall have the effect of executing the "start" operation.

3157 Input parameters:

- 3158 1) "force" - type: boolean - optional
3159 A flag to indicate whether the Provider shall simulate a power off condition (force=true) or shall
3160 simulate a shutdown operation that allows applications to save their state and the file system to
3161 be made consistent (force=false). Inclusion of this parameter by Consumers is optional and if
3162 not specified, the Provider may choose either mechanism. Providers are encouraged to
3163 advertise this choice by the way of the *MachineStopForceDefault* capability.

3164 Output parameters: None.

3165 During the processing of this operation, the *Machine* shall be in the "STOPPING" and/or "STARTING"
3166 states, as appropriate depending on its initial state.

3167 Upon successful completion of this operation, the *Machine* shall be in the "STARTED" state. Restarting
3168 a *Machine* shall be the virtual equivalent of powering off, and then powering on a physical machine.
3169 There is no restored CPU or Memory state, so the guest OS typically performs boot or installation tasks.

3170 **HTTP protocol**

3171 To restart a *Machine*, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/restart" URI of the
3172 *Machine* where the HTTP request body shall be as described below.

3173 **JSON media type:** application/json

3174 **JSON serialization:**

```
3175 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3176   "action": "http://schemas.dmtf.org/cimi/1/action/restart",
3177   "force": boolean, ?
3178   "properties": { string: string, + } ?
3179   ...
3180 }
```

3181 **XML media type:** application/xml

3182 **XML serialization**

```
3183 <Action xmlns="http://schemas.dmtf.org/cimi/1">
3184   <action> http://schemas.dmtf.org/cimi/1/action/restart </action>
3185   <force> xs:boolean </force> ?
3186   <property key="xs:string"> xs:string </property> *
3187   <xs:any>*
3188 </Action>
```

3189 Upon successful processing of the request, the HTTP response body may be empty.

3190 **pause**

3191 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/pause

3192 This operation shall pause a Machine.

3193 Input parameters: None.

3194 Output parameters: None.

3195 During the processing of this operation, the Machine shall be in the "PAUSING" state.

3196 Upon successful completion of this operation, the Machine shall be in the "PAUSED" state. Pausing a
 3197 Machine shall keep the Machine and its resources instantiated, but the Machine shall not be
 3198 available to perform any tasks. The current state of the CPU and Memory shall be retained in volatile
 3199 memory.

3200 **HTTP protocol**

3201 To pause a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action.pause" URI of the
 3202 Machine where the HTTP request body shall be as described below.

3203 **JSON media type:** application/json

3204 **JSON serialization:**

```
3205 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3206   "action": "http://schemas.dmtf.org/cimi/1/action/pause",
3207   "properties": { string: string, + } ?
3208   ...
3209 }
```

3210 **XML media type:** application/xml

3211 **XML serialization**

```
3212 <Action xmlns="http://schemas.dmtf.org/cimi/1">
3213   <action> http://schemas.dmtf.org/cimi/1/action/pause </action>
3214   <property key="xs:string"> xs:string </property> *
3215   <xs:any>*
3216 </Action>
```

3217 Upon successful processing of the request, the HTTP response body may be empty.

3218 **suspend**

3219 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/suspend`

3220 This operation shall suspend a `Machine`.

3221 Input parameters: None.

3222 Output parameters: None.

3223 During the processing of this operation, the `Machine` shall be in the "SUSPENDING" state.

3224 Upon successful completion of this operation, the `Machine` shall be in the "SUSPENDED" state.

3225 Suspending a `Machine` shall keep the `Machine` and its resources instantiated, but the `Machine` shall
3226 not be available to perform any tasks. The current state of the CPU and Memory shall be retained in
3227 non-volatile memory.

3228 **HTTP protocol**

3229 To suspend a `Machine`, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/suspend" URI of
3230 the `Machine` where the HTTP request body shall be as described below.

3231 **JSON media type:** `application/json`

3232 **JSON serialization:**

```
3233 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3234   "action": "http://schemas.dmtf.org/cimi/1/action/suspend",
3235   "properties": { string: string, + } ?
3236   ...
3237 }
```

3238 **XML media type:** `application/xml`

3239 **XML serialization**

```
3240 <Action xmlns="http://schemas.dmtf.org/cimi/1">
3241   <action> http://schemas.dmtf.org/cimi/1/action/suspend </action>
3242   <property key="xs:string"> xs:string </property> *
3243   <xs:any>*
3244 </Action>
```

3245 Upon successful processing of the request, the HTTP response body may be empty.

3246 **capture**

3247 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/capture`

3248 This operation shall create a new `MachineImage` from an existing `Machine`. This operation is defined
3249 within the `MachineImage` Resource; see 5.14.7.1 for more details. Note that while this operation is
3250 performed against a `MachineImage`, its presence in the `Machine` serialization is used to advertise
3251 support for the operation.

3252 **Snapshotting a Machine**

3253 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/snapshot`

3254 This operation shall create a new `SNAPSHOT MachineImage` from an existing `Machine`. This
 3255 operation is defined within the `MachineImage Resource`; see 5.14.7.1 for more details. Note that while
 3256 this operation is performed against a `MachineImage`, its presence in the `Machine` serialization is
 3257 used to advertise support for the operation.

3258 **Restoring a Machine**

3259 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/restore`

3260 This operation shall restore a `Machine` from a previously created `MachineImage`.

3261 Input parameters:

- 3262 1) "image" - type: URI - mandatory
- 3263 A reference to the `Machine Image`.

3264 Output parameters: None.

3265 During the processing of this operation, the `Machine` shall be in the "RESTORING" state.

3266 Upon successful completion of this operation, the `Machine` shall be in the same state as the state
 3267 specified in the `MachineImage`, if specified. See 5.14.2.1 for more details.

3268 Note that Providers can indicate support for restoring from non-`SNAPSHOT MachineImages` by the
 3269 way of the `Machine "RestoreFromImage"` capability. If the `RestoreFromImage` capability is not supported,
 3270 and the restore operation is supported, the restore operation can only restore from a `SNAPSHOT`
 3271 `MachineImage`.

3272 **HTTP protocol**

3273 To restore a `Machine`, a POST is sent to the "`http://schemas.dmtf.org/cimi/1/action/restore`" URI of the
 3274 `Machine` where the HTTP request body shall be as described below.

3275 **JSON media type:** `application/json`

3276 **JSON serialization:**

```
3277 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3278   "action": "http://schemas.dmtf.org/cimi/1/action/restore",
3279   "image": { "href": string },
3280   "properties": { string: string, + } ?
3281   ...
3282 }
```

3283 **XML media type:** `application/xml`

3284 **XML serialization**

```
3285 <Action xmlns="http://schemas.dmtf.org/cimi/1">
3286   <action> http://schemas.dmtf.org/cimi/1/action/restore </action>
3287   <image href="xs:anyURI"/>
3288   <property key="xs:string"> xs:string </property> *
```

```

3289     <xs:any>*
3290 </Action>

```

3291 Where the "image" URI is a reference to the MachineImage to be used.

3292 Upon successful processing of the request, the HTTP response body may be empty.

3293 5.14.2 MachineCollection Resource

3294 A MachineCollection Resource represents the Collection of Machine Resources within a
 3295 Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as
 3296 follows:

3297 JSON serialization:

```

3298 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCollection",
3299   "id": string,
3300   "count": number,
3301   "machines": [
3302     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
3303       "id": string,
3304       ... remaining Machine attributes ...
3305     }, +
3306   ], ?
3307   "operations": [ { "rel": "add", "href": string } ? ]
3308   "operations": [ { "rel": "insert", "href": string } ? ]
3309   "operations": [ { "rel": "remove", "href": string } ? ]
3310
3311   ...
3312 }

```

3313 XML serialization:

```

3314 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/MachineCollection"
3315   xmlns="http://schemas.dmtf.org/cimi/1">
3316   <id> xs:anyURI </id>
3317   <count> xs:integer </count>
3318   <Machine>
3319     <id> xs:anyURI </id>
3320     ... remaining Machine attributes ...
3321   </Machine> *
3322   <operation rel="add" href="xs:anyURI"/> ?
3323   <operation rel="insert" href="xs:anyURI"/> ?
3324   <operation rel="remove" href="xs:anyURI"/> ?
3325   <xs:any>*
3326 </Collection>

```

3327 **5.14.2.1 Operations**

3328 NOTE The "add" operation requires that a MachineTemplate be used (see 4.2.1.1).

3329 Within the NetworkInterface portion of the MachineTemplate, there may be a reference to an
 3330 Address Resource. If one is not provided, the Provider shall create one on the Consumer's behalf. In
 3331 these cases, and unless some action is taken to change this behavior, the Address is bound to the new
 3332 Machine that is created and shall be deleted by the Provider if the Machine is deleted. Additionally, if
 3333 these Provider-created Address Resources are disassociated from the Machine, the Provider shall
 3334 delete them. If the Consumer does provide an Address Resource, the Address shall not be deleted if
 3335 the Machine is deleted and it is then up to the Consumer to delete the Address through some other
 3336 mechanism.

3337 Upon successful processing of the "add" operation, unless otherwise specified by the way of the
 3338 MachineTemplate "initialState" attribute, the state of the new Machine shall be the value of the
 3339 DefaultInitialState capability, if defined. If no DefaultInitialState capability is defined, the default value shall
 3340 be "STOPPED." The semantics of "initialState" shall be equivalent to the Provider issuing the appropriate
 3341 actions against the new Machine to move it into that state. Note that this controls the actions of the
 3342 hypervisor and the state of the resources within the Machine (e.g., the operating system) are also
 3343 influenced by the data within the MachineImage used to create the new Machine. For example, if a
 3344 new Machine 's initialState is "STARTED" and a SNAPSHOT MachineImage was used to create the
 3345 new Machine, the Machine would not be "booted" but rather resume executing from the saved state in
 3346 the MachineImage.

3347 If a Provider is unable to change the state of the new Machine to the appropriate "initialState" (either as
 3348 specified by the MachineTemplate or as implied by the previous stated rules), the Machine creation
 3349 shall fail.

3350 If a Provider is unable to create the new Machine due to invalid or inconsistent credentials in the
 3351 MachineTemplate, the Machine creation process shall fail. If any credentials are included in the
 3352 MachineTemplate, they shall be part of the new Machine regardless of the type of
 3353 MachineImage used.

3354 **5.14.3 MachineTemplate**

3355 A MachineTemplate represents the set of metadata and instructions used in the creation of a
 3356 Machine. Table 15 describes the MachineTemplate attributes.

3357 **Table 15 – 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

Name	MachineTemplate													
Type URI	http://schemas.dmtf.org/cimi/1/MachineTemplate													
Attribute	Type	Description												
		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												
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 border="1" data-bbox="633 871 1429 1260"> <thead> <tr> <th colspan="2">Name</th> <th>volume</th> </tr> <tr> <th>Attribute</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <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> </tbody> </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>	<p>A list of structures, each containing a reference to a <code>VolumeTemplate</code> from which a <code>Volume</code> is created and connected to the <code>Machine</code> resulting from this <code>MachineTemplate</code>. Each structure can potentially also include aspects of the way in which each created <code>Volume</code> is connected to the created <code>Machine</code>.</p> <p>If the <code>Machine</code> is created as part of a <code>System</code> creation, the <code>Volumes</code> created from these <code>Templates</code> are considered as part of that <code>System</code> without the need for these <code>VolumeTemplates</code> to also be listed in the <code>volumeTemplates</code> attribute of the relevant <code>SystemTemplate</code>. If the same <code>VolumeTemplate</code> reference is listed in both the <code>volumeTemplates</code> attribute of a <code>SystemTemplate</code> and in the <code>volumeTemplates</code> attribute of a <code>MachineTemplate</code> contained by that <code>SystemTemplate</code>, this means that multiple, distinct <code>Volume</code> instances are created as part of the overall <code>System</code> creation. Each <code>volumeTemplate</code> structure has the following attributes:</p> <table border="1"> <thead> <tr> <th>Name</th> <th colspan="2"><i>volumeTemplate</i></th> </tr> <tr> <th>Attribute</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>initialLocation</td> <td><i>string</i></td> <td> <p>An Operating System-specific location (path) in its namespace where the <code>Volume</code> appears.</p> <p>Support of this attribute indicates that the Provider allows for Consumers to choose where the <code>Volume</code> appears.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p> </td> </tr> <tr> <td>volumeTemplate</td> <td><i>ref</i></td> <td> <p>Reference to the <code>VolumeTemplate</code> that is used to create a new <code>Volume</code>.</p> <p>Note that the attributes of the <code>VolumeTemplate</code> may be specified rather than a reference to an existing <code>VolumeTemplate</code> Resource.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p> </td> </tr> </tbody> </table> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>	Name	<i>volumeTemplate</i>		Attribute	Type	Description	initialLocation	<i>string</i>	<p>An Operating System-specific location (path) in its namespace where the <code>Volume</code> appears.</p> <p>Support of this attribute indicates that the Provider allows for Consumers to choose where the <code>Volume</code> appears.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>	volumeTemplate	<i>ref</i>	<p>Reference to the <code>VolumeTemplate</code> that is used to create a new <code>Volume</code>.</p> <p>Note that the attributes of the <code>VolumeTemplate</code> may be specified rather than a reference to an existing <code>VolumeTemplate</code> Resource.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>
Name	<i>volumeTemplate</i>													
Attribute	Type	Description												
initialLocation	<i>string</i>	<p>An Operating System-specific location (path) in its namespace where the <code>Volume</code> appears.</p> <p>Support of this attribute indicates that the Provider allows for Consumers to choose where the <code>Volume</code> appears.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>												
volumeTemplate	<i>ref</i>	<p>Reference to the <code>VolumeTemplate</code> that is used to create a new <code>Volume</code>.</p> <p>Note that the attributes of the <code>VolumeTemplate</code> may be specified rather than a reference to an existing <code>VolumeTemplate</code> Resource.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>												

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 border="1"> <thead> <tr> <th>Name</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <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> </tbody> </table>	Name	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	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																		
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 <i>MeterTemplates</i> that shall be used to create and connect a set of new <i>Meters</i> to the new <i>Machine</i> . Note that the attributes of the <i>MeterTemplate</i> may be specified rather than a reference to an existing <i>MeterTemplate</i> Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
eventLogTemplate	<i>ref</i>	A reference to an <i>EventLogTemplate</i> that shall be used to create and connect a new <i>EventLog</i> to the new <i>Machine</i> . Note that the attributes of the <i>EventLogTemplate</i> may be specified rather than a reference to an existing <i>EventLogTemplate</i> Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

3358 When implementing or using *MachineTemplate*, Providers and Consumers shall adhere to the syntax
 3359 and semantics of its attributes as described in Table 15, as well as in the tables describing embedded
 3360 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 3361 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 3362 both JSON and XML:

3363 **JSON media type:** application/json

3364 **JSON serialization:**

```

3365 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplate",
3366   "id": string,
3367   "name": string, ?
3368   "description": string, ?
3369   "created": string, ?
3370   "updated": string, ?
3371   "properties": { string: string, + }, ?
3372   "vscope" : [ valueScope, * ], ?
3373   "initialState": string, ?
3374   "machineConfig": {
3375     "href": string | ... MachineConfiguration attributes ...
3376   }, ?
3377   "machineImage": {
3378     "href": string | ... MachineImage attributes ...
3379   }, ?
3380   "credential": {
3381     "href": string | ... CredentialTemplate attributes ...
3382   }, ?
3383   "volumes": [
    
```

```

3384     { "initialLocation": string?, "href": string }, +
3385 ], ?
3386 "volumeTemplates": [
3387     { "initialLocation": string?,
3388       "href": string, ?
3389       ... VolumeTemplate attributes ... ?
3390     }, +
3391 ], ?
3392 "networkInterfaces": [
3393     { "addresses": [
3394       {"href": string}, +
3395     ],
3396     "network": {"href": string},
3397     "networkPort": {"href": string}, ?
3398     "state": string,
3399     "mtu": number ?
3400   }, +
3401 ], ?
3402 "userData": string, ?
3403 "meterTemplates": [
3404     { "href": string, ?
3405     ... MeterTemplate attributes ... ?
3406   }, *
3407 ], ?
3408 "eventLogTemplate": {
3409     "href": string, ?
3410     ... EventLogTemplate attributes ... ?
3411 }, ?
3412 "operations": [
3413     { "rel": "edit", "href": string }, ?
3414     { "rel": "delete", "href": string } ?
3415 ] ?
3416 ...
3417 }

```

3418 **XML media type:** application/xml

3419 **XML serialization:**

```

3420 <MachineTemplate xmlns="http://schemas.dmtf.org/cimi/1">
3421   <id> xs:anyURI </id>
3422   <name> xs:string </name> ?

```

```

3423 <description> xs:string </description> ?
3424 <created> xs:dateTime </created> ?
3425 <updated> xs:dateTime </updated> ?
3426 <property key="xs:string"> xs:string </property> *
3427 <vscope> valueScope </vscope> *
3428 <initialState> xs:string </initialState> ?
3429 <machineConfig href="xs:anyURI"?>
3430     ... MachineConfiguration attributes ... ?
3431 </machineConfig> ?
3432 <machineImage href="xs:anyURI"?>
3433     ... MachineImage attributes ... ?
3434 </machineImage> ?
3435 <credential href="xs:anyURI"?>
3436     ... CredentialTemplate attributes ... ?
3437 </credential> ?
3438 <volume initialLocation="xs:string"? href="xs:anyURI" /> *
3439 <volumeTemplate initialLocation="xs:string"? href="xs:anyURI"? >
3440     ... VolumeTemplate attributes ... ?
3441 </volumeTemplate> *
3442 <networkInterface>
3443     <address href="xs:anyURI"/> *
3444     <network href="xs:anyURI"/>
3445     <networkPort href="xs:anyURI"/> ?
3446     <state> xs:string </state>
3447     <mtu> xs:integer </mtu> ?
3448 </networkInterface> *
3449 <userData> xs:string </userData> ?
3450 <meterTemplate href="xs:anyURI"? >
3451     ... MeterTemplate attributes ... ?
3452 </meterTemplate> *
3453 <eventLogTemplate href="xs:anyURI"? >
3454     ... EventLogTemplate attributes ... ?
3455 </eventLogTemplate> ?
3456 <operation rel="edit" href="xs:anyURI"/> ?
3457 <operation rel="delete" href="xs:anyURI"/> ?
3458 <xs:any>*
3459 </MachineTemplate>
    
```

3460 Injection of user-defined data

3461 To simplify the customization of individual Machines, it is possible to pass arbitrary data into the new
 3462 Machine by using the userData parameter. The value of this parameter shall be the Base64-encoded

3463 payload. The Provider shall arrange for this data to be available from inside the `Machine` by using one
3464 of the following methods:

- 3465 1. *Metadata server*: The data can be retrieved from within the instance by using an HTTP GET
3466 request to `http://169.254.169.254/cimi/latest/user-data`.
- 3467 2. *Disk*: The `Machine` has access to a Disk with an ISO 9660 file system on it. The data can be
3468 found in a file at `<location>/cimi/user-data`.
- 3469 3. *Image modification*: The Provider modifies the root file system of the machine image just before
3470 launching the `Machine`. In UNIX-like operating systems, the data can be found in the file
3471 `/var/lib/cimi/user-data`.

3472 It is strongly recommended that Providers implement a `metadata server`, or, failing that, injection by
3473 the way of `Disk`, as `image modification` is brittle and may not work for every operating system in
3474 use. The Provider shall indicate which of these three methods is supported with the `Machine` 'UserData'
3475 capability in the `ResourceMetadata` for `Machines`. The value for this feature shall be one of
3476 `metadata`, `disk`, or `imgmod`, corresponding to the three methods listed above.

3477 The Provider shall preserve this data across restarts of the `Machine`. The data is the Base64-decoded
3478 version of the data that was passed into the `MachineCreate` request.

3479 5.14.3.1 Operations

3480 This Resource supports the Read, Update, and Delete operations. Create is supported through the
3481 `MachineTemplateCollection` Resource.

3482 5.14.4 MachineTemplateCollection Resource

3483 A `MachineTemplateCollection` Resource represents the Collection of `MachineTemplate`
3484 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
3485 shall be serialized as follows:

3486 JSON serialization:

```
3487 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplateCollection",
3488   "id": string,
3489   "count": number,
3490   "machineTemplates": [
3491     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineTemplate",
3492       "id": string,
3493       ... remaining MachineTemplate attributes ...
3494     }, +
3495   ], ?
3496   "operations": [ { "rel": "add", "href": string } ? ]
3497   ...
3498 }
```

3499 XML serialization:

```
3500 <Collection
3501   resourceURI="http://schemas.dmtf.org/cimi/1/MachineTemplateCollection"
3502   xmlns="http://schemas.dmtf.org/cimi/1">
```

```

3503 <id> xs:anyURI </id>
3504 <count> xs:integer </count>
3505 <MachineTemplate>
3506   <id> xs:anyURI </id>
3507   ... remaining MachineTemplate attributes ...
3508 </MachineTemplate> *
3509 <operation rel="add" href="xs:anyURI"/> ?
3510 <xs:any>*
3511 </Collection>

```

3512 **5.14.4.1 Operations**

3513 This Resource supports the Read and Update operations. Creation of new MachineTemplate
 3514 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 3515 4.2.1.1.

3516 **5.14.5 MachineConfiguration Resource**

3517 The MachineConfiguration Resource represents the set of configuration values that define the
 3518 (virtual) hardware resources of a to-be-realized Machine Instance. MachineConfigurations are
 3519 created by Providers and may, at the Providers discretion, be created by Consumers.

3520 Table 16 describes the MachineConfiguration attributes.

3521 **Table 16 – 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. 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. Constraints: Provider: support mandatory; mutable Consumer: support optional; 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 border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Name</th> <td>disk</td> </tr> <tr> <th>Attribute</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <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</td> </tr> </tbody> </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
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														

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

3522 NOTE The disk attributes "format" does not appear on *Machine* Resources because after the *Machine* is
 3523 created, the user of the *Machine* is able modify this attribute of a disk, possibly without the Provider's knowledge.
 3524 Therefore these attributes might not be an aspect of the *Machine* that the Provider can reliably manage.

3525 **JSON media type:** application/json

3526 **JSON serialization:**

```

3527 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
3528   "id": string,
3529   "name": string, ?
3530   "description": string, ?
3531   "created": string, ?
3532   "updated": string, ?
3533   "properties": { string: string, + }, ?
3534   "vscope" : [ valueScope, * ], ?
3535   "cpu": number,
3536   "memory": number,
3537   "disks" : [
3538     { "capacity": number,
3539       "format": string,
3540       "initialLocation": string?
3541     }, +
3542   ], ?
3543   "cpuArch": string, ?
3544   "cpuSpeed": number, ?
3545   "operations": [
3546     { "rel": "edit", "href": string }, ?
3547     { "rel": "delete", "href": string } ?
    
```



```

3548     ] ?
3549     ...
3550 }
    
```

3551 **XML media type:** application/xml

3552 **XML serialization:**

```

3553 <MachineConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
3554   <id> xs:anyURI </id>
3555   <name> xs:string </name> ?
3556   <description> xs:string </description> ?
3557   <created> xs:dateTime </created> ?
3558   <updated> xs:dateTime </updated> ?
3559   <property key="xs:string"> xs:string </property> *
3560   <vscope> valueScope </vscope> *
3561   <cpu> xs:integer </cpu>
3562   <memory> xs:integer </memory>
3563   <disk>
3564     <capacity> xs:integer </capacity>
3565     <format> xs:string </format>
3566     <initialLocation> xs:string </initialLocation> ?
3567   </disk> *
3568   <cpuArch> xs:string </cpuArch> ?
3569   <cpuSpeed> xs:integer </cpuSpeed> ?
3570   <operation rel="edit" href="xs:anyURI"/> ?
3571   <operation rel="delete" href="xs:anyURI"/> ?
3572   <xs:any>*
3573 </MachineConfiguration>
    
```

3574 5.14.5.1 Operations

3575 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 3576 MachineConfigurationCollection Resource.

3577 5.14.6 MachineConfigurationCollection Resource

3578 A MachineConfigurationCollection Resource represents the Collection of
 3579 MachineConfiguration Resources within a Provider and follows the Collection pattern defined in
 3580 clause 5.5.12. This Resource shall be serialized as follows:

3581 **JSON serialization:**

```

3582 { "resourceURI":
3583   "http://schemas.dmtf.org/cimi/1/MachineConfigurationCollection",
3584   "id": string,
3585   "count": number,
    
```

```

3586     "machineConfigurations": [
3587         { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
3588           "id": string,
3589           ... remaining MachineConfiguration attributes ...
3590         }, +
3591     ], ?
3592     "operations": [ { "rel": "add", "href": string } ? ]
3593     ...
3594 }

```

3595 XML serialization:

```

3596 <Collection
3597     resourceURI="http://schemas.dmtf.org/cimi/1/MachineConfigurationCollection"
3598     xmlns="http://schemas.dmtf.org/cimi/1">
3599     <id> xs:anyURI </id>
3600     <count> xs:integer </count>
3601     <MachineConfiguration>
3602         <id> xs:anyURI </id>
3603         ... remaining MachineConfiguration attributes ...
3604     </MachineConfiguration> *
3605     <operation rel="add" href="xs:anyURI"/> ?
3606     <xs:any>*
3607 </Collection>

```

3608 5.14.6.1 Operations

3609 This Resource supports the Read and Update operations. Creation of new `MachineConfiguration`
 3610 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 3611 4.2.1.1.

3612 5.14.7 MachineImage Resource

3613 This Resource represents the information necessary for hardware virtualized Resources to create a
 3614 `Machine` Instance; it contains configuration data such as startup instructions, including possible
 3615 combinations of the following items, depending on the "type" of `MachineImage` created:

- 3616 • The software image (i.e., a copy of an installed `Machine`), that is to be instantiated on the disk
 3617 and other virtual resources. The image can be a snapshot that consists of disk images plus
 3618 memory and other resource state information.
- 3619 • Installation software, which, when executed on the hardware (virtual) resources, builds the
 3620 machine instance.
- 3621 • Both a disk image and a set of software and parameters to install new components not included
 3622 in the original disk image.

3623 Table 17 describes the `MachineImage` attributes.

3624

Table 17 – MachinelImage attributes

Name	MachinelImage	
Type URI	http://schemas.dmtf.org/cimi/1/MachinelImage	
Attribute	Type	Description
state	string	<p>The operational state of the MachineImage.</p> <p>Allowable values include:</p> <p>CREATING: The MachineImage is in the process of being created.</p> <p>AVAILABLE: The MachineImage is available and ready for use. Unless otherwise specified, the MachinelImage shall initially be in this state after successful creation.</p> <p>DELETING: The MachineImage is in the process of being deleted.</p> <p>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</p> <p>Constraints:</p> <p>Provider: support mandatory; mutable</p> <p>Consumer: support mandatory; read-only</p>

Name	MachineImage	
Type URI	http://schemas.dmtf.org/cimi/1/MachineImage	
Attribute	Type	Description
type	string	<p>The type of MachineImage that is represented by this Resource. This specification defines the following values:</p> <p>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.</p> <p>SNAPSHOT: This type represents the persisted data of a Machine. If the Machine was not in a stopped state when his 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.</p> <p>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.</p> <p>If a MachineImage is deleted, the following semantics shall apply:</p> <ul style="list-style-type: none"> 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. <p>Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only</p>
imageLocation	URI	<p>A reference to the location of the binary data that makes up this image.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>
relatedImage	ref	<p>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.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-only</p>

3625 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

3626 **JSON media type:** application/json

3627 **JSON serialization:**

```

3628 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
3629     "id": string,
3630     "name": string, ?
3631     "description": string, ?
3632     "created": string, ?
3633     "updated": string, ?
3634     "properties": { string: string, + }, ?
3635     "state": string,
3636     "type": string,
3637     "imageLocation": string,
3638     "relatedImage": { "href": string }, ?
3639     "operations": [
3640         { "rel": "edit", "href": string }, ?
3641         { "rel": "delete", "href": string } ?
3642     ] ?
3643     ...
3644 }
```

3645 **XML media type:** application/xml

3646 **XML serialization:**

```

3647 <MachineImage xmlns="http://schemas.dmtf.org/cimi/1">
3648     <id> xs:anyURI </id>
3649     <name> xs:string </name> ?
3650     <description> xs:string </description> ?
3651     <created> xs:dateTime </created> ?
3652     <updated> xs:dateTime </updated> ?
3653     <property key="xs:string"> xs:string </property> *
3654     <state> xs:string </state>
3655     <type> xs:string </type>
3656     <imageLocation> xs:anyURI </imageLocation>
3657     <relatedImage href="xs:anyURI"/> ?
3658     <operation rel="edit" href="xs:anyURI"/> ?
3659     <operation rel="delete" href="xs:anyURI"/> ?
3660     <xs:any>*
3661 </MachineImage>
```

3662 5.14.7.1 Operations

3663 This Resource supports the Read, Update, and Delete operations. Create is supported through the
3664 `MachineImageCollection` Resource.

3665 If creating a new `MachineImage`, the representation of the new `MachineImage` may include a
3666 reference in the "imageLocation" attribute. Providers shall inspect this reference (most likely by the way of
3667 an HTTP HEAD) to determine if any special processing is required. This specification defines the
3668 following additional steps that Providers shall take depending on the type of Resource being referenced:

3669 `http://schemas.dmtf.org/cimi/1/Machine`

3670 If the "imageLocation" is a reference to a `Machine`, the Provider shall create a new `MachineImage`
3671 based on the `Machine` being referenced. The machine is captured or snapshotted, depending on
3672 whether the request was sent to the "<http://schemas.dmtf.org/cimi/1/action/capture>" or the
3673 "<http://schemas.dmtf.org/cimi/1/action/snapshot>" URI of the `Machine`. However the resulting resource,
3674 although linked to the `Machine` from which it was originated, shall be a `MachineImage` for all purposes
3675 and can be used for creating new machines.

3676 If creating a SNAPSHOT and upon completion of the create operation, the `MachineImage`'s
3677 "imageLocation" attribute shall not reference the `Machine` (as the `Machine` might change over time),
3678 but instead it shall reference (or contain the data of) the static representation of the `Machine`.
3679 Additionally, the referenced `Machine`'s `MachineSnapshotCollection` shall be updated to
3680 include a reference to this newly created SNAPSHOT `MachineImage` Resource. If the `Machine` is
3681 unable to accept operations at any point while it is being captured to create the `MachineImage`, the
3682 `Machine` shall be in state "CAPTURING".

3683 5.14.8 MachineImageCollection Resource

3684 A `MachineImageCollection` Resource represents the Collection of `MachineImage` Resources
3685 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
3686 serialized as follows:

3687 JSON serialization:

```
3688 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImageCollection",
3689   "id": string,
3690   "count": number,
3691   "machineImages": [
3692     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
3693       "id": string,
3694       ... remaining MachineImage attributes ...
3695     }, +
3696   ], ?
3697   "operations": [ { "rel": "add", "href": string } ? ]
3698   ...
3699 }
```

3700 XML serialization:

```
3701 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/MachineImageCollection"
3702   xmlns="http://schemas.dmtf.org/cimi/1">
```

```

3703 <id> xs:anyURI </id>
3704 <count> xs:integer </count>
3705 <MachineImage>
3706   <id> xs:anyURI </id>
3707   ... remaining MachineImage attributes ...
3708 </MachineImage> *
3709 <operation rel="add" href="xs:anyURI"/> ?
3710 <xs:any>*
3711 </Collection>
    
```

3712 **5.14.8.1 Operations**

3713 This Resource supports the Read and Update operations. Creation of new MachineImage Resources
 3714 is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1, where the
 3715 request body and the way it is processed are described in clause 5.14.7.1.

3716 **5.14.9 Credential Resource**

3717 A Credential Resource contains the information required to create the initial administrative superuser
 3718 of a newly created Machine or to represent the credentials needed to perform some operation. Due to
 3719 the variation between operating systems and Providers, this specification does not mandate one
 3720 particular set of attributes that all implementations need to support. However, Providers are expected to
 3721 extend this Resource with additional attributes to meet their requirements.

3722 For example, a Provider might extend this Resource with username and password attributes, which would
 3723 then be the login information for new Machines. These extension attributes would appear as siblings to
 3724 the common attributes like "name" and "description."

3725 Table 18 describes the Credential attributes.

3726 **Table 18 – 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.

3727 Some common extension attributes that Providers might use include:

3728 **Table 19 – 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

3729 **Table 20 – Public key attributes**

Attribute	Type	Description
key	byte[]	The digit of the public key for the initial superuser.

Attribute	Type	Description
		Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

3730 When implementing or using `Credential`, Providers and Consumers shall adhere to the syntax and
3731 semantics of its attributes as described in the above table, as well as in the table describing related
3732 Collections. Both Consumer and Provider shall serialize this Resource as described below. The following
3733 pseudo-schemas (see notation in 1.3)

3734 **JSON media type:** application/json

3735 **JSON serialization:**

```
3736 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Credential",
3737   "id": string,
3738   "name": string, ?
3739   "description": string, ?
3740   "created": string, ?
3741   "updated": string, ?
3742   "properties": { string: string, + }, ?
3743   "operations": [
3744     { "rel": "edit", "href": string }, ?
3745     { "rel": "delete", "href": string } ?
3746   ] ?
3747   ...
3748 }
```

3749 **XML media type:** application/xml

3750 **XML serialization:**

```
3751 <Credential xmlns="http://schemas.dmtf.org/cimi/1">
3752   <id> xs:anyURI </id>
3753   <name> xs:string </name> ?
3754   <description> xs:string </description> ?
3755   <created> xs:dateTime </created> ?
3756   <updated> xs:dateTime </updated> ?
3757   <property key="xs:string"> xs:string </property> *
3758   <operation rel="edit" href="xs:anyURI"/> ?
3759   <operation rel="delete" href="xs:anyURI"/> ?
3760   <xs:any>*
3761 </Credential>
```

3762 5.14.9.1 Operations

3763 This Resource supports the Read, Update, and Delete operations. Create is supported through the
3764 `CredentialCollection` Resource.

3765 **5.14.10 CredentialCollection Resource**

3766 A CredentialCollection Resource represents the Collection of Credential Resources within
 3767 a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized
 3768 as follows:

3769 **JSON serialization:**

```

3770 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialCollection",
3771   "id": string,
3772   "count": number,
3773   "credentials": [
3774     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Credential",
3775       "id": string,
3776       ... remaining Credential attributes ...
3777     }, +
3778   ], ?
3779   "operations": [ { "rel": "add", "href": string } ? ]
3780   ...
3781 }
```

3782 **XML serialization:**

```

3783 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/CredentialCollection"
3784   xmlns="http://schemas.dmtf.org/cimi/1">
3785   <id> xs:anyURI </id>
3786   <count> xs:integer </count>
3787   <Credential>
3788     <id> xs:anyURI </id>
3789     ... remaining Credential attributes ...
3790   </Credential> *
3791   <operation rel="add" href="xs:anyURI"/> ?
3792   <xs:any>*
3793 </Collection>
```

3794 **5.14.10.1 Operations**

3795 NOTE The "add" operation requires that a CredentialTemplate be used (see 4.2.1.1).

3796 **5.14.11 CredentialTemplate Resource**

3797 This Resource captures the configuration values for realizing a Credential Resource. A
 3798 CredentialTemplate may be used to create multiple Credentials. Table 21 describes the
 3799 CredentialTemplate attributes.

3800

Table 21 – 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.

3801 When implementing or using `CredentialTemplate`, Providers and Consumers shall adhere to the
 3802 syntax and semantics of its attributes as described in Table 21 as well as in the table describing related
 3803 Collections. Both Consumer and Provider shall serialize this Resource as described below. The following
 3804 pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML:

3805 **JSON media type:** application/json

3806 **JSON serialization:**

```
3807 { "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialTemplate",
3808   "id": string,
3809   "name": string, ?
3810   "description": string, ?
3811   "created": string, ?
3812   "updated": string, ?
3813   "properties": { string: string, + }, ?
3814   "operations": [
3815     { "rel": "edit", "href": string }, ?
3816     { "rel": "delete", "href": string } ?
3817   ] ?
3818   ...
3819 }
```

3820 **XML media type:** application/xml

3821 **XML serialization:**

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

3833 5.14.11.1 Operations

3834 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 3835 `CredentialTemplateCollection` Resource.

3836 **5.14.12 CredentialTemplateCollection Resource**

3837 A `CredentialTemplateCollection` Resource represents the Collection of
3838 `CredentialTemplate` Resources within a Provider and follows the Collection pattern defined in
3839 clause 5.5.12. This Resource shall be serialized as follows:

3840 **JSON serialization:**

```
3841 { "resourceURI":  
3842     "http://schemas.dmtf.org/cimi/1/CredentialTemplateCollection",  
3843     "id": string,  
3844     "count": number,  
3845     "credentialTemplates": [  
3846         { "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialTemplate",  
3847           "id": string,  
3848           ... remaining CredentialTemplate attributes ...  
3849         }, +  
3850     ], ?  
3851     "operations": [ { "rel": "add", "href": string }? ]  
3852     ...  
3853 }
```

3854 **XML serialization:**

```

3855 <Collection
3856   resourceURI="http://schemas.dmtf.org/cimi/1/CredentialTemplateCollection"
3857   xmlns="http://schemas.dmtf.org/cimi/1">
3858   <id> xs:anyURI </id>
3859   <count> xs:integer </count>
3860   <CredentialTemplate>
3861     <id> xs:anyURI </id>
3862     ... remaining CredentialTemplate attributes ...
3863   </CredentialTemplate> *
3864   <operation rel="add" href="xs:anyURI"/> ?
3865   <xs:any>*
3866 </Collection>
    
```

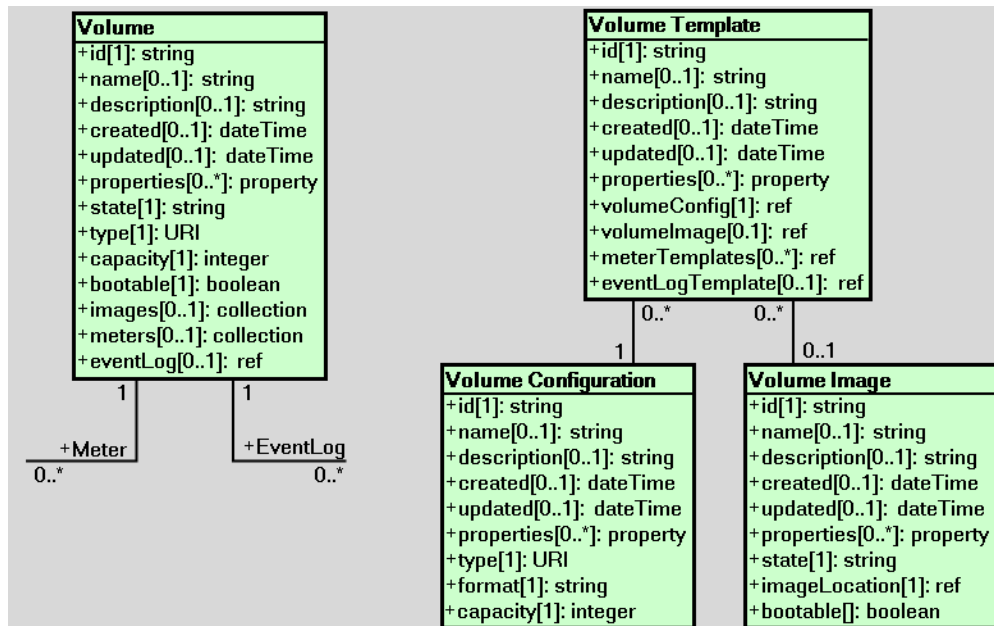
3867 **5.14.12.1 Operations**

3868 This Resource supports the Read and Update operations. Creation of new `CredentialTemplate`
 3869 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 3870 4.2.1.1.

3871 **5.15 Volume Resources and relationships**

3872 Figure 4 illustrates the Resources involved in constructing a `Volume` and their relationships. Although
 3873 this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor
 3874 normative.

3875



3876 **Figure 4 - Volume Resources**

3877 **5.15.1 Volume**

3878 A `Volume` represents storage at either the block or the file-system level. `Volumes` can be connected to
 3879 `Machines`. Once connected, `Volumes` can be accessed by processes on that `Machine`. Table 22
 3880 describes the `Volume` attributes.

 3881 **Table 22 – Volume attributes**

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

Name	Volume	
Type URI	http://schemas.dmtf.org/cimi/1/Volume	
Attribute	Type	Description
eventLog	<i>ref</i>	A reference to the EventLog of this Volume. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

3882 When implementing or using `Volume`, Providers and Consumers shall adhere to the syntax and
3883 semantics of its attributes as described in the above table as well as in the tables describing embedded
3884 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
3885 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
3886 both JSON and XML:

3887 **JSON media type:** application/json

3888 **JSON serialization:**

```
3889 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Volume",
3900   "id": string,
3901   "name": string, ?
3902   "description": string, ?
3903   "created": string, ?
3904   "updated": string, ?
3905   "properties": { string: string, + }, ?
3906   "state": string,
3907   "type": string,
3908   "capacity": number,
3909   "bootable": boolean,
3910   "images": { "href": string }, ?
3911   "meters": { "href": string }, ?
3912   "eventLog": { "href": string }, ?
3913   "operations": [
3914     { "rel": "edit", "href": string }, ?
3915     { "rel": "delete", "href": string } ?
3916   ] ?
3917   ...
3918 }
```

3909 **XML media type:** application/xml

3910 **XML serialization:**

```
3911 <Volume xmlns="http://schemas.dmtf.org/cimi/1">
3912   <id> xs:anyURI </id>
3913   <name> xs:string </name> ?
3914   <description> xs:string </description> ?
3915   <created> xs:dateTime </created> ?
3916   <updated> xs:dateTime </updated> ?
```

```

3917     <property key="xs:string"> xs:string </property> *
3918     <state> xs:string </state>
3919     <type> xs:anyURI </type>
3920     <capacity> xs:integer </capacity>
3921     <bootable> xs:boolean </bootable>
3922     <images href="xs:anyURI"/> ?
3923     <meters href="xs:anyURI"/> ?
3924     <eventLog href="xs:anyURI"/> ?
3925     <operation rel="edit" href="xs:anyURI"/> ?
3926     <operation rel="delete" href="xs:anyURI"/> ?
3927     <xs:any>*
3928 </Volume>

```

3929 5.15.1.1 Collections

3930 The following clauses describe the Collection Resources owned by Volumes.

3931 5.15.1.1.1 images Collection

3932 The Resource type for each item of this Collection is "VolumeImage". There is no accessory attribute
 3933 for the items in this Collection, therefore it is a basic VolumeImage Collection (serialized as described
 3934 in 5.5.12).

3935 See the VolumeImageCollection Resource clause.

3936 NOTE Previous versions of this specification included an "add" operation on this Resource. It is now deprecated in
 3937 favor of creating a new VolumeImage with the imageLocation attribute pointing to the Volume to be captured.

3938 5.15.1.1.2 meters Collection

3939 The Resource type for each item of this Collection is "Meter" as defined in clause 5.17.3. There is no
 3940 accessory attribute for the items in this Collection, therefore it is a basic Meter Collection (serialized as
 3941 described in 5.5.12).

3942 See the MeterCollection Resource clause.

3943 5.15.1.2 Operations

3944 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 3945 VolumeCollection Resource.

3946 In addition also the following custom operations are supported.

3947 snapshot

3948 **/link@rel:** <http://schemas.dmtf.org/cimi/1/action/snapshot>

3949 This operation shall create a new VolumeImage from an existing Volume. This operation is defined
 3950 within the VolumeImage Resource; see 5.15.7.1 for more details. Note that while this operation is
 3951 performed against a VolumeImage, its presence in the Volume serialization is used to advertise
 3952 support for the operation.

3953 If the Volume is unable to accept operations at any point while it is creating the VolumeImage, the
3954 Volume shall be in the state "CAPTURING".

3955 **restore**

3956 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/restore

3957 This operation shall restore a Volume from a previously created VolumeImage.

3958 Input parameters:

3959 1) "image" - type: URI - mandatory
3960 A reference to the Volume Image.

3961 Output parameters: None.

3962 During the processing of this operation, the Volume shall be in the "RESTORING" state.

3963 Upon successful completion of this operation, the Volume shall again be in the state "AVAILABLE".

3964 **HTTP protocol**

3965 To restore a Volume, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/restore" URI of the
3966 Volume where the HTTP request body shall be as described below.

3967 **JSON media type:** application/json

3968 **JSON serialization:**

```
3969 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
3970   "action": "http://schemas.dmtf.org/cimi/1/action/restore",
3971   "image": string,
3972   "properties": { string: string, + } ?
3973   ...
3974 }
```

3975 **XML media type:** application/xml

3976 **XML serialization**

```
3977 <Action xmlns="http://schemas.dmtf.org/cimi/1">
3978   <action> http://schemas.dmtf.org/cimi/1/action/restore </action>
3979   <image href="xs:anyURI"/>
3980   <property key="xs:string"> xs:string </property> *
3981   <xs:any>*
3982 </Action>
```

3983 Where the "image" URI is a reference to the VolumeImage to be used.

3984 Upon successful processing of the request, the HTTP response body may be empty.

3985 5.15.2 VolumeCollection Resource

3986 A VolumeCollection Resource represents the Collection of Volumes within a Provider and follows
3987 the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

3988 **JSON serialization:**

```

3989 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeCollection",
3990     "id": string,
3991     "count": number,
3992     "volumes": [
3993         { "resourceURI": "http://schemas.dmtf.org/cimi/1/Volume",
3994           "id": string,
3995           ... remaining Volume attributes ...
3996         }, +
3997     ], ?
3998     "operations": [ { "rel": "add", "href": string } ? ]
3999     ...
4000 }
```

4001 **XML serialization:**

```

4002 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/VolumeCollection"
4003     xmlns="http://schemas.dmtf.org/cimi/1">
4004     <id> xs:anyURI </id>
4005     <count> xs:integer </count>
4006     <Volume>
4007         <id> xs:anyURI </id>
4008         ... remaining Volume attributes ...
4009     </Volume> *
4010     <operation rel="add" href="xs:anyURI"/> ?
4011     <xs:any>*
4012 </Collection>
```

4013 **5.15.2.1 Operations**

4014 NOTE The "add" operation requires that a VolumeTemplate be used (see 4.2.1.1).

4015 **5.15.3 VolumeTemplate Resource**

4016 This Resource captures the configuration values for realizing a Volume. A VolumeTemplate may be
 4017 used to create multiple Volumes. Table 23 describes the VolumeTemplate attributes.

4018 **Table 23 – 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

Name	VolumeTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/VolumeTemplate	
Attribute	Type	Description
volumedImage	<i>ref</i>	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	<i>Meter Templates[]</i>	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
eventLog Template	<i>ref</i>	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

4019 When implementing or using VolumeTemplate, Providers and Consumers shall adhere to the syntax
4020 and semantics of its attributes as described in the above table as well as in the tables describing
4021 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
4022 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
4023 Resource in both JSON and XML.

4024 **JSON media type:** application/json

4025 **JSON serialization:**

```
4026 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeTemplate",
4027   "id": string,
4028   "name": string, ?
4029   "description": string, ?
4030   "created": string, ?
4031   "updated": string, ?
4032   "properties": { string: string, + }, ?
4033   "volumeConfig": {
4034     "href": string | ... VolumeConfiguration attributes ...
4035   },
4036   "volumeImage": { "href": string }, ?
4037   "meterTemplates": [
4038     { "href": string, ?
4039       ... MeterTemplate attributes ... ?
4040     }, *
4041   ], ?
4042   "eventLogTemplate": {
4043     "href": string, ?
```

```

4044     ... EventLogTemplate attributes ... ?
4045     }, ?
4046     "operations": [
4047         { "rel": "edit", "href": string }, ?
4048         { "rel": "delete", "href": string } ?
4049     ] ?
4050     ...
4051 }
    
```

4052 **XML media type:** application/xml

4053 **XML serialization:**

```

4054 <VolumeTemplate xmlns="http://schemas.dmtf.org/cimi/1">
4055     <id> xs:anyURI </id>
4056     <name> xs:string </name> ?
4057     <description> xs:string </description> ?
4058     <created> xs:dateTime </created> ?
4059     <updated> xs:dateTime </updated> ?
4060     <property key="xs:string"> xs:string </property> *
4061     <volumeConfig href="xs:anyURI"?>
4062         ... VolumeConfiguration attributes ... ?
4063     </volumeConfig>
4064     <volumeImage href="xs:anyURI"/> ?
4065     <meterTemplate href="xs:anyURI"? >
4066         ... MeterTemplate attributes ... ?
4067     </meterTemplate> *
4068     <eventLogTemplate href="xs:anyURI"? >
4069         ... EventLogTemplate attributes ... ?
4070     </eventLogTemplate> ?
4071     <operation rel="edit" href="xs:anyURI"/> ?
4072     <operation rel="delete" href="xs:anyURI"/> ?
4073     <xs:any>*
4074 </VolumeTemplate>
    
```

4075 5.15.3.1 Operations

4076 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 4077 VolumeTemplateCollection Resource.

4078 5.15.4 VolumeTemplateCollection Resource

4079 A VolumeTemplateCollection Resource represents the Collection of VolumeTemplate
 4080 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
 4081 shall be serialized as follows:

4082 **JSON serialization:**

```

4083 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeTemplateCollection",
4084     "id": string,
4085     "count": number,
4086     "volumeTemplates": [
4087         { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeTemplate",
4088           "id": string,
4089           ... remaining volumeTemplate attributes ...
4090         }, +
4091     ], ?
4092     "operations": [ { "rel": "add", "href": string } ? ]
4093     ...
4094 }

```

4095 **XML serialization:**

```

4096 <Collection
4097     resourceURI="http://schemas.dmtf.org/cimi/1/VolumeTemplateCollection"
4098     xmlns="http://schemas.dmtf.org/cimi/1">
4099     <id> xs:anyURI </id>
4100     <count> xs:integer </count>
4101     <VolumeTemplate>
4102         <id> xs:anyURI </id>
4103         ... remaining VolumeTemplates attributes ...
4104     </VolumeTemplate> *
4105     <operation rel="add" href="xs:anyURI"/> ?
4106     <xs:any>*
4107 </Collection>

```

4108 **5.15.4.1 Operations**

4109 This Resource supports the Read and Update operations. Creation of new `VolumeTemplate`
 4110 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 4111 4.2.1.1.

4112 **5.15.5 VolumeConfiguration Resource**

4113 The `VolumeConfiguration` Resource represents the set of configuration values needed to create a
 4114 `Volume` with certain characteristics. `VolumeConfigurations` are created by Providers and may, at
 4115 the Providers discretion, be created by Consumers.

4116 Table 24 describes the `VolumeConfiguration` attributes.

4117

Table 24 – 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

4118 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

4119 **JSON media type:** application/json

4120 **JSON serialization:**

```

4121 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
4122   "id": string,
4123   "name": string, ?
4124   "description": string, ?
4125   "created": string, ?
4126   "updated": string, ?
4127   "properties": { string: string, + }, ?
4128   "type": string,
4129   "format": string,
4130   "capacity": number,
4131   "operations": [
4132     { "rel": "edit", "href": string }, ?
4133     { "rel": "delete", "href": string } ?
4134   ] ?
4135   ...
4136 }
```

4137 **XML media type:** application/xml

4138 **XML serialization:**

```

4139 <VolumeConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
4140   <id> xs:anyURI </id>
4141   <name> xs:string </name> ?
4142   <description> xs:string </description> ?
4143   <created> xs:dateTime </created> ?
4144   <updated> xs:dateTime </updated> ?
4145   <property key="xs:string"> xs:string </property> *
4146   <type> xs:anyURI </type>
4147   <format> xs:string </format>
4148   <capacity> xs:integer </capacity>
4149   <operation rel="edit" href="xs:anyURI"/> ?
4150   <operation rel="delete" href="xs:anyURI"/> ?
4151   <xs:any>*
4152 </VolumeConfiguration>

```

4153 5.15.5.1 Operations

4154 This Resource supports the Read, Update, and Delete operations. Create is supported through the
4155 VolumeConfigurationCollection Resource.

4156 5.15.6 VolumeConfigurationCollection Resource

4157 A VolumeConfigurationCollection Resource represents the Collection of
4158 VolumeConfiguration Resources within a Provider and follows the Collection pattern defined in
4159 clause 5.5.12. This Resource shall be serialized as follows:

4160 **JSON serialization:**

```

4161 { "resourceURI":
4162   "http://schemas.dmtf.org/cimi/1/VolumeConfigurationCollection",
4163   "id": string,
4164   "count": number,
4165   "volumeConfigurations": [
4166     { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
4167       "id": string,
4168       ... remaining VolumeConfiguration attributes ...
4169     }, +
4170   ], ?
4171   "operations": [ { "rel": "add", "href": string } ? ]
4172   ...
4173 }

```

4174 **XML serialization:**

```

4175 <Collection
4176     resourceURI="http://schemas.dmtf.org/cimi/1/VolumeConfigurationCollection"
4177     xmlns="http://schemas.dmtf.org/cimi/1">
4178     <id> xs:anyURI </id>
4179     <count> xs:integer </count>
4180     <VolumeConfiguration>
4181         <id> xs:anyURI </id>
4182         ... remaining VolumeConfiguration attributes ...
4183     </VolumeConfiguration> *
4184     <operation rel="add" href="xs:anyURI"/> ?
4185     <xs:any>*
4186 </Collection>
    
```

4187 **5.15.6.1 Operations**

4188 This Resource supports the Read and Update operations. Creation of new VolumeImage Resources is
 4189 supported by the way of a POST to the "add" operations' URI as described in clause 4.2.1.1.

4190 **5.15.7 Volumelimage Resource**

4191 This Resource represents an image that could be placed on a preloaded volume. Table 25 describes the
 4192 VolumeImage attributes.

4193 **Table 25 – Volumelimage attributes**

Name	Volumelimage	
Type URI	http://schemas.dmtf.org/cimi/1/Volumelimage	
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 Constraints: Provider: support mandatory; mutable 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
bootable	boolean	This property indicates whether Volumes created from this VolumeImage are bootable. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

4194 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

4195 **JSON media type:** application/json

4196 **JSON serialization:**

```

4197 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeImage",
4198   "id": string,
4199   "name": string, ?
4200   "description": string, ?
4201   "created": string, ?
4202   "updated": string, ?
4203   "properties": { string: string, + }, ?
4204   "state": string,
4205   "imageLocation": string,
4206   "bootable": boolean,
4207   "operations": [
4208     { "rel": "edit", "href": string }, ?
4209     { "rel": "delete", "href": string } ?
4210   ] ?
4211   ...
4212 }
```

4213 **XML media type:** application/xml

4214 **XML serialization:**

```

4215 <VolumeImage xmlns="http://schemas.dmtf.org/cimi/1">
4216   <id> xs:anyURI </id>
4217   <name> xs:string </name> ?
4218   <description> xs:string </description> ?
4219   <created> xs:dateTime </created> ?
4220   <updated> xs:dateTime </updated> ?
4221   <property key="xs:string"> xs:string </property> *
4222   <state> xs:string </state>
4223   <imageLocation>xs:anyURI</imageLocation>
4224   <bootable> xs:boolean </bootable>
4225   <operation rel="edit" href="xs:anyURI"/> ?
4226   <operation rel="delete" href="xs:anyURI"/> ?
4227   <xs:any>*
4228 </VolumeImage>
```

4229 5.15.7.1 Operations

4230 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 4231 VolumeImageCollection Resource.

4232 **5.15.8 VolumeImageCollection Resource**

4233 A `VolumeImageCollection` Resource represents the Collection of `VolumeImage` Resources
 4234 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
 4235 serialized as follows:

4236 **JSON serialization:**

```
4237 { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeImageCollection",
4238   "id": string,
4239   "count": number,
4240   "volumeImages": [
4241     { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeImage",
4242       "id": string,
4243       ... remaining VolumeImage attributes ...
4244     }, +
4245   ], ?
4246   "operations": [ { "rel": "add", "href": string } ? ]
4247   ...
4248 }
```

4249 **XML serialization:**

```
4250 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/VolumeImageCollection"
4251   xmlns="http://schemas.dmtf.org/cimi/1">
4252   <id> xs:anyURI </id>
4253   <count> xs:integer </count>
4254   <VolumeImage>
4255     <id> xs:anyURI </id>
4256     ... remaining VolumeImage attributes ...
4257   </VolumeImage> *
4258   <operation rel="add" href="xs:anyURI"/> ?
4259   <xs:any>*
4260 </Collection>
```

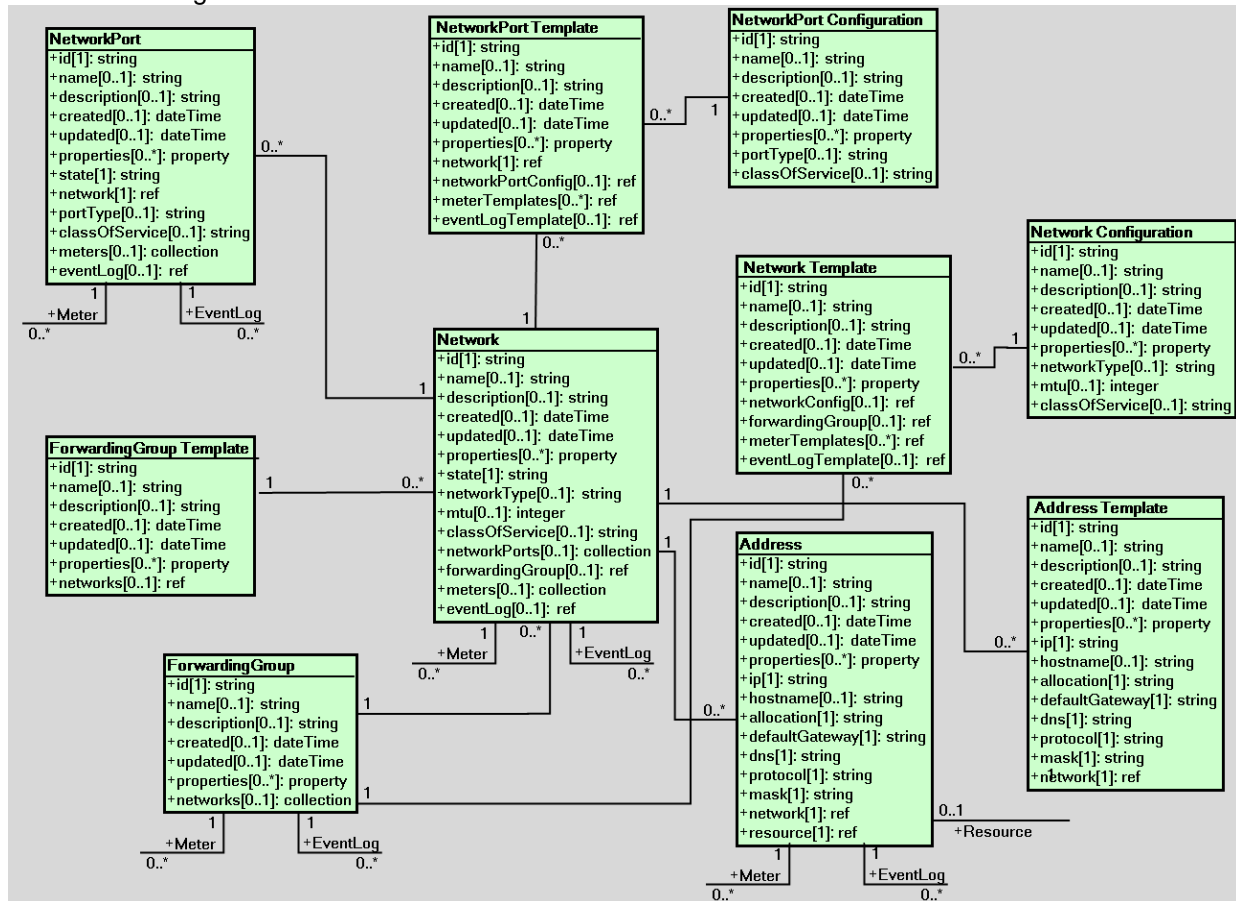
4261 **5.15.8.1 Operations**

4262 This Resource supports the Read and Update operations. Creation of new `VolumeImage` Resources is
 4263 supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.

4264 During the creation of a new `VolumeImage` Resource, if the "imageLocation" attribute refers to an
 4265 existing `Volume`, this operation shall be interpreted as a request to create a snapshot of the `Volume`.
 4266 Once completed, the "imageLocation" attribute of the new `VolumeImage` Resource shall not refer to the
 4267 original `Volume`; instead it shall refer to a static copy of the `Volume`. Additionally, the referenced
 4268 `Volume`'s `VolumeImageCollection` shall be updated to include a reference to this newly created
 4269 snapshot `VolumeImage` Resource. During this process, the Provider may put the `Volume` into a
 4270 "CAPTURING" state if necessary.

4271 **5.16 Network Resources and relationships**

4272 Figure 5 illustrates the Resources involved in constructing *Networks* and their *NetworkPorts* and
 4273 their relationships. Although this drawing is in the style of a Resource Relationship diagram, the use of
 4274 UML is neither rigorous nor normative.



4275 **Figure 5 - Network Resources**

4276 **5.16.1 Network**

4277 A *Network* is a Collection of interconnected logical services with the purpose of forwarding data traffic
 4278 between end points.

4279 *Networks* in a *ForwardingGroup* should all have the same "networkType" attributes, which
 4280 prevents a *Network* with a "private" access attribute from being publicly forwarded because it is a
 4281 member of a *ForwardingGroup* that also contains *Networks* with a "public" access attribute.

4282 Table 26 describes the *Network* attributes.

4283 **Table 26 – Network attributes**

Name	Network	
Type URI	http://schemas.dmtf.org/cimi/1/Network	
Attribute	Type	Description
state	string	The operational state of the <i>Network</i> . Allowable values include: CREATING : The <i>Network</i> is in the process of being created.

Name	Network	
Type URI	http://schemas.dmtf.org/cimi/1/Network	
Attribute	Type	Description
		<p>STARTING: The <i>Network</i> is in the process of being started.</p> <p>STARTED: The <i>Network</i> is available and ready for use.</p> <p>STOPPING: The <i>Network</i> is in the process of being stopped.</p> <p>STOPPED: The <i>Network</i> is stopped and not available for use.</p> <p>DELETING: The <i>Network</i> is in the process of being deleted.</p> <p>ERROR: The Provider has detected an error in the <i>Network</i>. <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></p> <p>Constraints:</p> <p>Provider: support mandatory; mutable</p> <p>Consumer: support mandatory; read-only</p>
networkType	string	<p>An indicator of whether the <i>Machine</i> Resource has access to a Public or Private <i>Network</i>.</p> <p>Allowable values include:</p> <p>PUBLIC: represents an open and Internet routable network.</p> <p>PRIVATE: identifies a local non-routed network.</p> <p>Constraints:</p> <p>Provider: support mandatory; mutable</p> <p>Consumer: support optional; read-write</p>
mtu	integer	<p>(Maximum Transmission Unit) The largest Packet size supported on this <i>Network</i>.</p> <p>Constraints:</p> <p>Provider: support optional; mutable</p> <p>Consumer: support optional; read-write</p>
classOfService	string	<p>The Provider's supported category associated with a Collection of attributes characterizing a level of a quality experience.</p> <p>Example values:</p> <p>GOLD: High bandwidth, low latency, low jitter</p> <p>SILVER: An improved service experience over bronze for voice or video traffic</p> <p>BRONZE: Best effort</p> <p>The list of possible values, and their implied quality of service, is out of scope of this specification.</p> <p>Constraints:</p> <p>Provider: support optional; mutable</p> <p>Consumer: support optional; read-write</p>
networkPorts	collection [<i>Network Port</i>]	<p>A reference to the list of references to <i>NetworkPorts</i> that are associated with this <i>Network</i>. This reference has component semantics for the referred <i>NetworkPorts</i>.</p> <p>Constraints:</p> <p>Provider: support optional; mutable</p> <p>Consumer: support optional; read-only</p>
forwardingGroup	ref	<p>A reference to a <i>ForwardingGroup</i> of which this <i>Network</i> is a part.</p> <p>Constraints:</p> <p>Provider: support optional; mutable</p> <p>Consumer: support optional; read-only</p>
meters	collection [<i>Meter</i>]	<p>A reference to the list of <i>Meters</i> monitored for this <i>Network</i>.</p> <p>Constraints:</p> <p>Provider: support optional; mutable</p> <p>Consumer: support optional; read-only</p>
eventLog	ref	<p>A reference to the <i>EventLog</i> of this <i>Network</i>.</p> <p>Constraints:</p> <p>Provider: support optional; mutable</p> <p>Consumer: support optional; read-only</p>

4284 When implementing or using *Network*, Providers and Consumers shall adhere to the syntax and
 4285 semantics of its attributes as described in Table 26 as well as in the tables describing embedded
 4286 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described

4287 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 4288 both JSON and XML.

4289 **JSON media type:** application/json

4290 **JSON serialization:**

```

4291 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Network",
4292   "id": string,
4293   "name": string, ?
4294   "description": string, ?
4295   "created": string, ?
4296   "updated": string, ?
4297   "properties": { string: string, + }, ?
4298   "state": string,
4299   "networkType": string, ?
4300   "mtu": number, ?
4301   "classOfService": string, ?
4302   "networkPorts": { "href": string }, ?
4303   "forwardingGroup": { "href": string }, ?
4304   "meters": { "href": string }, ?
4305   "eventLog": { "href": string }, ?
4306   "operations": [
4307     { "rel": "edit", "href": string }, ?
4308     { "rel": "delete", "href": string }, ?
4309     { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
4310     { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
4311   ] ?
4312   ...
4313 }
```

4314 **XML media type:** application/xml

4315 **XML serialization:**

```

4316 <Network xmlns="http://schemas.dmtf.org/cimi/1">
4317   <id> xs:anyURI </id>
4318   <name> xs:string </name> ?
4319   <description> xs:string </description> ?
4320   <created> xs:dateTime </created> ?
4321   <updated> xs:dateTime </updated> ?
4322   <property key="xs:string"> xs:string </property> *
4323   <state> xs:string </state>
4324   <networkType> xs:string </networkType> ?
4325   <mtu> xs:integer </mtu> ?
```

```

4326 <classOfService> xs:string </classOfService> ?
4327 <networkPorts href="xs:anyURI"/> ?
4328 <forwardingGroup href="xs:anyURI"/> ?
4329 <meters href="xs:anyURI"/> ?
4330 <eventLog href="xs:anyURI"/> ?
4331 <operation rel="edit" href="xs:anyURI"/> ?
4332 <operation rel="delete" href="xs:anyURI"/> ?
4333 <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
4334 href="xs:anyURI"/> ?
4335 <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
4336 href="xs:anyURI"/> ?
4337 <xs:any>*
4338 </Network>

```

4339 5.16.1.1 Collections

4340 The following clauses describe the Collection Resources owned by Networks.

4341 5.16.1.1.1 networkPorts Collection

4342 The Resource type for each item of this Collection is "NetworkPort". There is no accessory attribute
4343 for the items in this Collection, therefore it is a basic NetworkPort Collection (serialized as described
4344 in 5.5.12).

4345 See the NetworkPortCollection Resource clause.

4346 As specified in clause 5.5.12, if a Network is deleted, all of its Collections, and Resources in those
4347 Collections, shall also be deleted. This means that all of the NetworkPorts related to that Network
4348 shall also be deleted.

4349 5.16.1.1.2 meters Collection

4350 The Resource type for each item of this Collection is "Meter" as defined in clause 5.17.3. There is no
4351 accessory attribute for the items in this Collection, therefore it is a basic Meter Collection (serialized as
4352 described in 5.5.12).

4353 See the MeterCollection Resource clause.

4354 5.16.1.2 Operations

4355 This Resource supports the Read, Update, and Delete operations. Create is supported through the
4356 NetworkCollection Resource.

4357 The following custom operations are also defined:

4358 start

4359 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/start

4360 This operation shall start a Network.

4361 Input parameters: None.

4362 Output parameters: None.

4363 During the processing of this operation, the `Network` shall be in the "STARTING" state.

4364 Upon successful completion of this operation, the `Network` shall be in the "STARTED" state.

4365 **HTTP protocol**

4366 To start a `Network`, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/start" URI of the
4367 `Network` where the HTTP request body shall be as described below.

4368 **JSON media type:** application/json

4369 **JSON serialization:**

```
4370 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
4371   "action": "http://schemas.dmtf.org/cimi/1/action/start",
4372   "properties": { string: string, + } ?
4373   ...
4374 }
```

4375 **XML media type:** application/xml

4376 **XML serialization**

```
4377 <Action xmlns="http://schemas.dmtf.org/cimi/1">
4378   <action> http://schemas.dmtf.org/cimi/1/action/start </action>
4379   <property key="xs:string"> xs:string </property> *
4380   <xs:any>*
4381 </Action>
```

4382 Upon successful processing of the request, the HTTP response body may be empty.

4383 **stop**

4384 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/stop

4385 This operation shall stop a `Network`. If stopped, a `Network` shall not allow data to flow through it.

4386 Input parameters: None.

4387 Output parameters: None.

4388 During the processing of this operation, the `Network` shall be in the "STOPPING" state.

4389 Upon successful completion of this operation, the `Network` shall be in the "STOPPED" state.

4390 **HTTP protocol**

4391 To stop a `Network`, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the
4392 `Network` where the HTTP request body shall be as described below.

4393 **JSON media type:** application/json

4394 **JSON serialization:**

```
4395 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
4396   "action": "http://schemas.dmtf.org/cimi/1/action/stop",
4397   "properties": { string: string, + } ?
4398   ...
4399 }
```

4400 **XML media type:** application/xml

4401 **XML serialization**

```
4402 <Action xmlns="http://schemas.dmtf.org/cimi/1">
4403   <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
4404   <property key="xs:string"> xs:string </property> *
4405   <xs:any>*
4406 </Action>
```

4407 Upon successful processing of the request, the HTTP response body may be empty.

4408 **5.16.2 NetworkCollection Resource**

4409 A NetworkCollection Resource represents the Collection of Networks within a Provider and
 4410 follows the Collection pattern that is defined in clause 5.5.12. This Resource shall be serialized as follows:

4411 **JSON serialization:**

```
4412 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkCollection",
4413   "id": string,
4414   "count": number,
4415   "networks": [
4416     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Network",
4417       "id": string,
4418       ... remaining Network attributes ...
4419     }, +
4420   ], ?
4421   "operations": [ { "rel": "add", "href": string } ? ]
4422   ...
4423 }
```

4424 **XML serialization:**

```
4425 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/NetworkCollection"
4426   xmlns="http://schemas.dmtf.org/cimi/1">
4427   <id> xs:anyURI </id>
4428   <count> xs:integer </count>
4429   <Network>
4430     <id> xs:anyURI </id>
4431     ... remaining Network attributes ...
```

```

4432     </Network> *
4433     <operation rel="add" href="xs:anyURI"/> ?
4434     <xs:any>*
4435 </Collection>
    
```

4436 5.16.2.1 Operations

4437 NOTE The "add" operation requires that a `NetworkTemplate` be used (see 4.2.1.1).

4438 Upon successful processing of the "add" operation, unless otherwise specified by the way of the
 4439 `NetworkTemplate` "initialState" attribute, the state of the new `Network` shall be the value of the
 4440 `DefaultInitialState` capability of the `Network` Resource's `ResourceMetadata`, if defined. If no
 4441 `DefaultInitialState` capability is defined, the default value shall be "STOPPED." The semantics of
 4442 "initialState" shall be equivalent to the Provider issuing the appropriate actions against the new `Network`
 4443 to move it into that state.

4444 If a Provider is unable to change the state of the new `Network` to the appropriate "initialState" (either as
 4445 specified by the `NetworkTemplate` or as implied by the previous stated rules), the `Network` creation
 4446 shall fail.

4447 5.16.3 NetworkTemplate Resource

4448 The `NetworkTemplate` is a set of configuration values for realizing a `Network`. An instance of
 4449 `NetworkTemplate` may be used to create multiple `Networks`. Table 27 describes the
 4450 `NetworkTemplate` attributes.

4451 **Table 27 – NetworkTemplate attributes**

Name	NetworkTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkTemplate	
Attribute	Type	Description
initialState	<i>string</i>	The initial state of the new <code>Network</code> . Possible values include the non-transient states as specified by the <code>Network</code> "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 <code>Network</code> <code>ResourceMetadata</code> "initialStates" capability. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
networkConfig	<i>ref</i>	A reference to the <code>NetworkConfiguration</code> that is used to create a <code>Network</code> from this <code>NetworkTemplate</code> . Note that the attributes of the <code>NetworkConfiguration</code> may be specified rather than a reference to an existing <code>NetworkConfiguration</code> Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
networkPorts	<i>network Port[]</i>	A list of reference to <code>NetworkPorts</code> to be added to <code>NetworkPort</code> collection of the <code>Network</code> during its creation from this <code>NetworkTemplate</code> . Constraints: Provider: support optional; mutable Consumer: support optional; read-write

Name	NetworkTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkTemplate	
Attribute	Type	Description
networkPortTemplates	<i>network Port Template[]</i>	A list of references to <code>NetworkPortTemplates</code> , from every template referenced, a <code>NetworkPort</code> is created and added to the <code>NetworkPort</code> collection of the <code>Network</code> resulting from this <code>NetworkTemplate</code> . Constraints: Provider: support optional; mutable Consumer: support optional; read-write
forwardingGroup	<i>ref</i>	A reference to a <code>ForwardingGroup</code> of which this <code>Network</code> is a part. Note that <code>Networks</code> forward to themselves; therefore, this attribute only appears in cases where the <code>Network</code> that is created from this <code>Template</code> forwards to one or more additional <code>Networks</code> . Constraints: Provider: support optional; mutable Consumer: support optional; read-write
meterTemplates	<i>meter Templates[]</i>	A list of references to <code>MeterTemplates</code> that shall be used to create and connect a set of new <code>Meters</code> to the new <code>Network</code> . Note that the attributes of the <code>MeterTemplate</code> may be specified rather than a reference to an existing <code>MeterTemplate</code> Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
eventLogTemplate	<i>ref</i>	A reference to an <code>EventLogTemplate</code> that shall be used to create and connect a new <code>EventLog</code> to the new <code>Network</code> . Note that the attributes of the <code>EventLogTemplate</code> may be specified rather than a reference to an existing <code>EventLogTemplate</code> Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

4452 When implementing or using `NetworkTemplate`, Providers and Consumers shall adhere to the syntax
 4453 and semantics of its attributes as described in Table 27 as well as in the tables describing embedded
 4454 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 4455 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 4456 both JSON and XML

4457 **JSON media type:** application/json

4458 **JSON serialization:**

```

4459 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkTemplate",
4460   "id": string,
4461   "name": string, ?
4462   "description": string, ?
4463   "created": string, ?
4464   "updated": string, ?
4465   "properties": { string: string, + }, ?
4466   "initialState": string, ?
4467   "networkConfig": {
4468     "href": string |... NetworkingConfiguration attributes ...
4469   }, ?
4470   "forwardingGroup": { "href": string }, ?
    
```

```

4471 "meterTemplates": [
4472   { "href": string, ?
4473     ... MeterTemplate attributes ... ?
4474   }, *
4475 ], ?
4476 "eventLogTemplate": {
4477   "href": string, ?
4478   ... EventLogTemplate attributes ... ?
4479 }, ?
4480 "operations": [
4481   { "rel": "edit", "href": string }, ?
4482   { "rel": "delete", "href": string } ?
4483 ] ?
4484 ...
4485 }

```

4486 **XML media type:** application/xml

4487 **XML serialization:**

```

4488 <NetworkTemplate xmlns="http://schemas.dmtf.org/cimi/1">
4489   <id> xs:anyURI </id>
4490   <name> xs:string </name> ?
4491   <description> xs:string </description> ?
4492   <created> xs:dateTime </created> ?
4493   <updated> xs:dateTime </updated> ?
4494   <property key="xs:string"> xs:string </property> *
4495   <initialState> xs:string </initialState> ?
4496   <networkConfig href="xs:anyURI"?>
4497     ... NetworkConfiguration attributes ... ?
4498   </networkConfig> ?
4499   <forwardingGroup href="xs:anyURI"/> ?
4500   <meterTemplate href="xs:anyURI"? >
4501     ... MeterTemplate attributes ... ?
4502   </meterTemplate> *
4503   <eventLogTemplate href="xs:anyURI"? >
4504     ... EventLogTemplate attributes ... ?
4505   </eventLogTemplate> ?
4506   <operation rel="edit" href="xs:anyURI"/> ?
4507   <operation rel="delete" href="xs:anyURI"/> ?
4508   <xs:any>*
4509 </NetworkTemplate>

```

4510 **5.16.3.1 Operations**

4511 This Resource supports the Read, Update and Delete operations. Create is supported through the
 4512 `NetworkTemplateCollection` Resource.

4513 **5.16.4 NetworkTemplateCollection Resource**

4514 A `NetworkTemplateCollection` Resource represents the Collection of `NetworkTemplates`
 4515 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
 4516 serialized as follows:

4517 **JSON serialization:**

```
4518 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkTemplateCollection",
4519   "id": string,
4520   "count": number,
4521   "networkTemplates": [
4522     { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkTemplate",
4523       "id": string,
4524       ... remaining NetworkTemplate attributes ...
4525     }, +
4526   ], ?
4527   "operations": [ { "rel": "add", "href": string } ? ]
4528   ...
4529 }
```

4530 **XML serialization:**

```
4531 <Collection
4532   resourceURI="http://schemas.dmtf.org/cimi/1/NetworkTemplateCollection"
4533   xmlns="http://schemas.dmtf.org/cimi/1">
4534   <id> xs:anyURI </id>
4535   <count> xs:integer </count>
4536   <NetworkTemplate>
4537     <id> xs:anyURI </id>
4538     ... remaining NetworkTemplate attributes ...
4539   </NetworkTemplate> *
4540   <operation rel="add" href="xs:anyURI"/> ?
4541   <xs:any>*
4542 </Collection>
```

4543 **5.16.4.1 Operations**

4544 This Resource supports the Read and Update operations. Creation of new `NetworkTemplate`
 4545 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 4546 4.2.1.1.

4547 **5.16.5 NetworkConfiguration Resource**

4548 The following set of configuration values (shown in Table 28) represent the information needed to create
 4549 a `Network` with certain characteristics.

4550 **Table 28 – NetworkConfiguration attributes**

Name	NetworkConfiguration	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkConfiguration	
Attribute	Type	Description
networkType	<i>string</i>	An indicator of whether the <code>Network</code> is a Public or Private Network. Allowable values include: PUBLIC : represents an open and Internet routable network. PRIVATE : identifies a local non-Internet network. Constraints: Provider : support optional; mutable Consumer : support optional; read-write
mtu	<i>integer</i>	(Maximum Transmission Unit) The largest supported packet size. Constraints: Provider : support optional; mutable Consumer : support optional; read-write
classOfService	<i>string</i>	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

4551 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

4552 **JSON media type:** application/json

4553 **JSON serialization:**

```

4554 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkConfiguration",
4555   "id": string,
4556   "name": string, ?
4557   "description": string, ?
4558   "created": string, ?
4559   "updated": string, ?
4560   "properties": { string: string, + }, ?
4561   "networkType": string, ?
4562   "mtu": number, ?
4563   "classOfService": string, ?
4564   "operations": [
4565     { "rel": "edit", "href": string }, ?
4566     { "rel": "delete", "href": string } ?
4567   ] ?
4568   ...

```

4569 }
}4570 **XML media type:** application/xml4571 **XML serialization:**

```

4572 <NetworkConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
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   <networkType> xs:string </networkType> ?
4580   <mtu> xs:integer <mtu> ?
4581   <classOfService> xs:string </classOfService> ?
4582   <operation rel="edit" href="xs:anyURI"/> ?
4583   <operation rel="delete" href="xs:anyURI"/> ?
4584   <xs:any>*
4585 </NetworkConfiguration>

```

4586 **5.16.5.1 Operations**

4587 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 4588 NetworkConfigurationCollection Resource.

4589 **5.16.6 NetworkConfigurationCollection Resource**

4590 A NetworkConfigurationCollection Resource represents the Collection of
 4591 NetworkConfigurations within a Provider and follows the Collection pattern defined in clause
 4592 5.5.12. This Resource shall be serialized as follows:

4593 **JSON serialization:**

```

4594 { "resourceURI":
4595   "http://schemas.dmtf.org/cimi/1/NetworkConfigurationCollection",
4596   "id": string,
4597   "count": number,
4598   "networkConfigurations": [
4599     { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkConfiguration",
4600       "id": string,
4601       ... remaining NetworkConfiguration attributes ...
4602     }, +
4603   ], ?
4604   "operations": [ { "rel": "add", "href": string } ? ]
4605   ...
4606 }

```

4607 **XML serialization:**

```
4608 <Collection
4609     resourceURI="http://schemas.dmtf.org/cimi/1/NetworkConfigurationCollection"
4610     xmlns="http://schemas.dmtf.org/cimi/1">
4611     <id> xs:anyURI </id>
4612     <count> xs:integer </count>
4613     <NetworkConfiguration>
4614         <id> xs:anyURI </id>
4615         ... remaining NetworkConfiguration attributes ...
4616     </NetworkConfiguration> *
4617     <operation rel="add" href="xs:anyURI"/> ?
4618     <xs:any>*
4619 </Collection>
```

4620 **5.16.6.1 Operations**

4621 This Resource supports the Read and Update operations. Creation of new NetworkConfiguration
4622 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
4623 4.2.1.1.

4624 **5.16.7 NetworkPort**

 4625 A `NetworkPort` is a realized connection point between a `Network` and a `Resource`, such as a
 4626 `Machine`. Table 29 describes the `NetworkPort` attributes.

 4627 **Table 29 – NetworkPort attributes**

Name	NetworkPort	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkPort	
Attribute	Type	Description
state	<i>string</i>	The operational state of the <code>NetworkPort</code> . Allowable values include: CREATING: The <code>NetworkPort</code> is in the process of being created. STARTED: The <code>NetworkPort</code> is available (enabled) and ready for use. STOPPED: The <code>NetworkPort</code> is stopped (disabled) and not available for use. DELETING: The <code>NetworkPort</code> is in the process of being deleted. ERROR: The Provider has detected an error in the <code>NetworkPort</code> . 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 <code>NetworkPort</code> . <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-only
network	<i>ref</i>	A reference to the <code>Network</code> associated with this <code>NetworkPort</code> . <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-write
portType	<i>string</i>	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	<i>string</i>	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	<i>collection [Meter]</i>	A reference to the list of <code>Meters</code> monitored for this <code>NetworkPort</code> . <u>Constraints:</u> Provider: support optional; mutable Consumer: support optional; read-only
eventLog	<i>ref</i>	A reference to the <code>EventLog</code> of this <code>NetworkPort</code> . <u>Constraints:</u> Provider: support optional; mutable Consumer: support optional; read-only

 4628 When implementing or using `NetworkPort`, Providers and Consumers shall adhere to the syntax and
 4629 semantics of its attributes as described in Table 29 as well as in the tables describing embedded
 4630 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described

4631 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 4632 both JSON and XML.

4633 **JSON media type:** application/json

4634 **JSON serialization:**

```

4635 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPort",
4636   "id": string,
4637   "name": string, ?
4638   "description": string, ?
4639   "created": string, ?
4640   "updated": string, ?
4641   "properties": { string: string, + }, ?
4642   "state": string,
4643   "network": { "href": string },
4644   "portType": string, ?
4645   "classOfService": string, ?
4646   "meters": { "href": string }, ?
4647   "eventLog": { "href": string }, ?
4648   "operations": [
4649     { "rel": "edit", "href": string }, ?
4650     { "rel": "delete", "href": string }, ?
4651     { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
4652     { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
4653   ] ?
4654   ...
4655 }
```

4656 **XML media type:** application/xml

4657 **XML serialization:**

```

4658 <NetworkPort xmlns="http://schemas.dmtf.org/cimi/1">
4659   <id> xs:anyURI </id>
4660   <name> xs:string </name> ?
4661   <description> xs:string </description> ?
4662   <created> xs:dateTime </created> ?
4663   <updated> xs:dateTime </updated> ?
4664   <property key="xs:string"> xs:string </property> *
4665   <state> xs:string </state>
4666   <network href="xs:anyURI"/>
4667   <portType> xs:string </portType> ?
4668   <classOfService> xs:string </classOfService> ?
4669   <meters href="xs:anyURI"/> ?
```



```

4670     <eventLog" href="xs:anyURI"/> ?
4671     <operation rel="edit" href="xs:anyURI"/> ?
4672     <operation rel="delete" href="xs:anyURI"/> ?
4673     <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
4674     href="xs:anyURI"/> ?
4675     <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
4676     href="xs:anyURI"/> ?
4677     <xs:any>*
4678 </NetworkPort>

```

4679 5.16.7.1 Collections

4680 The following clauses describe the Collection Resources owned by `NetworkPorts`.

4681 5.16.7.1.1 meters Collection

4682 The Resource type for each item of this Collection is “Meter” as defined in clause 5.17.3. There is no
 4683 accessory attribute for the items in this Collection, therefore it is a basic `Meter` Collection (serialized as
 4684 described in 5.5.12).

4685 5.16.7.2 Operations

4686 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 4687 `NetworkPortCollection` Resource.

4688 Deleting a `NetworkPort` shall remove that `NetworkPort` from the global (Cloud Entry Point)
 4689 `NetworkPortCollection` as well as from its corresponding `Network`'s
 4690 `NetworkPortsCollection`.

4691 The following custom operations are also defined:

4692 start

4693 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/start`

4694 This operation shall start a `NetworkPort`.

4695 Input parameters: None.

4696 Output parameters: None.

4697 Upon successful completion of this operation, the `NetworkPort` shall be in the "STARTED" state.

4698 HTTP protocol

4699 To start a `NetworkPort`, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/start" URI of the
 4700 `NetworkPort` where the HTTP request body shall be as described below.

4701 **JSON media type:** `application/json`

4702 JSON serialization:

```

4703     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
4704     "action": "http://schemas.dmtf.org/cimi/1/action/start",
4705     "properties": { string: string, + } ?

```

4706 ...
 4707 }

4708 **XML media type:** application/xml

4709 **XML serialization**

```
4710        <Action xmlns="http://schemas.dmtf.org/cimi/1">
4711            <action> http://schemas.dmtf.org/cimi/1/action/start </action>
4712            <property key="xs:string"> xs:string </property> *
4713            <xs:any>*
4714        </Action>
```

4715 Upon successful processing of the request, the HTTP response body may be empty.

4716 **stop**

4717 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/stop

4718 This operation shall stop a `NetworkPort`. If stopped, the `NetworkPort` shall not be available for use
 4719 and no network traffic shall flow through it.

4720 Input parameters: None.

4721 Output parameters: None.

4722 Upon successful completion of this operation, the `NetworkPort` shall be in the "STOPPED" state.

4723 **HTTP protocol**

4724 To stop a `NetworkPort`, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the
 4725 `NetworkPort` where the HTTP request body shall be as described below.

4726 **JSON media type:** application/json

4727 **JSON serialization:**

```
4728        { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
4729            "action": "http://schemas.dmtf.org/cimi/1/action/stop",
4730            "properties": { string: string, + } ?
4731            ...
4732        }
```

4733 **XML media type:** application/xml

4734 **XML serialization**

```
4735        <Action xmlns="http://schemas.dmtf.org/cimi/1">
4736            <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
4737            <property key="xs:string"> xs:string </property> *
4738            <xs:any>*
4739        </Action>
```

4740 Upon successful processing of the request, the HTTP response body may be empty.

4741 5.16.8 NetworkPortCollection Resource

4742 A `NetworkPortCollection` Resource represents the Collection of `NetworkPorts` within a
 4743 Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as
 4744 follows:

4745 JSON serialization:

```
4746 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortCollection",
4747   "id": string,
4748   "count": number,
4749   "networkPorts": [
4750     { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPort",
4751       "id": string,
4752       ... remaining NetworkPort attributes ...
4753     }, +
4754   ], ?
4755   "operations": [ { "rel": "add", "href": string } ? ]
4756   ...
4757 }
```

4758 XML serialization:

```
4759 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/NetworkPortCollection"
4760   xmlns="http://schemas.dmtf.org/cimi/1">
4761   <id> xs:anyURI </id>
4762   <count> xs:integer </count>
4763   <NetworkPort>
4764     <id> xs:anyURI </id>
4765     ... remaining NetworkPort attributes ...
4766   </NetworkPort> *
4767   <operation rel="add" href="xs:anyURI"/> ?
4768   <xs:any>*
4769 </Collection>
```

4770 5.16.8.1 Operations

4771 NOTE The "add" operation requires that a `NetworkPortTemplate` be used (see 4.2.1.1).

4772 If `NetworkPorts` are created through the global (Cloud Entry Point) `NetworkPortCollection`'s
 4773 "add" operation, they are automatically associated with the corresponding `Network`, by addition of the
 4774 `NetworkPort`'s reference in the `networkPorts` Collection of the `Network`.

4775 Upon successful processing of the "add" operation, unless otherwise specified by the
 4776 `NetworkPortTemplate` "initialState" attribute, the state of the new `NetworkPort` shall be the
 4777 value of the `DefaultInitialState` capability of the `NetworkPort` Resource's `ResourceMetadata`, if
 4778 defined. If no `DefaultInitialState` capability is defined, the default value shall be "STOPPED." The
 4779 semantics of "initialState" shall be equivalent to the Provider issuing the appropriate actions against the
 4780 new `NetworkPort` to move it into that state.

4781 If a Provider is unable to change the state of the new `NetworkPort` to the appropriate "initialState"
 4782 (either as specified by the `NetworkPortTemplate` or as implied by the previous stated rules), the
 4783 `NetworkPort` creation shall fail.

4784 **5.16.9 NetworkPortTemplate Resource**

4785 The `NetworkPortTemplate` is a set of Configuration values for realizing a `NetworkPort`. A
 4786 `NetworkPortTemplate` may be used to create multiple `NetworkPorts`. Table 30 describes the
 4787 `NetworkPortTemplate` attributes.

4788 **Table 30 – NetworkPortTemplate attributes**

Name	NetworkPortTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/NetworkPortTemplate	
Attribute	Type	Description
initialState	<i>string</i>	The initial state of the new <code>NetworkPort</code> . Possible values include the non-transient states as specified by the <code>NetworkPort</code> "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 <code>NetworkPort</code> <code>ResourceMetadata</code> "initialStates" capability. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
network	<i>ref</i>	A reference to the network to be associated with this <code>NetworkPort</code> . If this Template is used to create a new <code>NetworkPort</code> through the global (Cloud Entry Point) <code>NetworkPort</code> Collection, this attribute shall be present. If this Template is referred in a <code>NetworkTemplate</code> and used to create a new <code>NetworkPort</code> during the creation of a <code>Network</code> , this attribute shall either be absent or have the same value as the "id" of the <code>Network</code> to which this <code>NetworkPort</code> is being added. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
networkPortConfig	<i>ref</i>	A reference to the <code>NetworkPortConfiguration</code> that is used to create a <code>NetworkPort</code> from this <code>NetworkPortTemplate</code> . Note that the attributes of the <code>NetworkPortConfiguration</code> may be specified rather than a reference to an existing <code>NetworkPortConfiguration</code> Resource. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
meterTemplates	<i>meterTemplates[]</i>	A list of references to <code>MeterTemplates</code> that shall be used to create and connect a set of new <code>Meters</code> to the new <code>NetworkPort</code> . Note that the attributes of the <code>MeterTemplate</code> may be specified rather than a reference to an existing <code>MeterTemplate</code> Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
eventLogTemplate	<i>ref</i>	A reference to an <code>EventLogTemplate</code> that shall be used to create and connect a new <code>EventLog</code> to the new <code>NetworkPort</code> . Note that the attributes of the <code>EventLogTemplate</code> may be specified rather than a reference to an existing <code>EventLogTemplate</code> Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

4789 When implementing or using `NetworkPortTemplate`, Providers and Consumers shall adhere to the
 4790 syntax and semantics of its attributes as described in Table 30 as well as in the tables describing
 4791 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
 4792 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 4793 Resource in both JSON and XML.

4794 **JSON media type:** `application/json`

4795 **JSON serialization:**

```
4796 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortTemplate",
4797   "id": string,
4798   "name": string, ?
4799   "description": string, ?
4800   "created": string, ?
4801   "updated": string, ?
4802   "properties": { string: string, + }, ?
4803   "initialState": string, ?
4804   "network": { "href": string }, ?
4805   "networkPortConfig": {
4806     "href": string | ... NetworkPortConfiguration attributes ...
4807   },
4808   "meterTemplates": [
4809     { "href": string, ?
4810       ... MeterTemplate attributes ... ?
4811     }, *
4812   ], ?
4813   "eventLogTemplate": {
4814     "href": string, ?
4815     ... EventLogTemplate attributes ... ?
4816   }, ?
4817   "operations": [
4818     { "rel": "edit", "href": string }, ?
4819     { "rel": "delete", "href": string } ?
4820   ] ?
4821   ...
4822 }
```

4823 **XML media type:** `application/xml`

4824 **XML serialization:**

```
4825 <NetworkPortTemplate xmlns="http://schemas.dmtf.org/cimi/1">
4826   <id> xs:anyURI </id>
4827   <name> xs:string </name> ?
4828   <description> xs:string </description> ?
```

```

4829 <created> xs:dateTime </created> ?
4830 <updated> xs:dateTime </updated> ?
4831 <property key="xs:string"> xs:string </property> *
4832 <initialState> xs:string </initialState> ?
4833 <network href="xs:anyURI"/> ?
4834 <networkPortConfig href="xs:anyURI"?>
4835     ... NetworkPortConfiguration attributes ... ?
4836 </networkPortConfig>
4837 <meterTemplate href="xs:anyURI"? >
4838     ... MeterTemplate attributes ... ?
4839 </meterTemplate> *
4840 <eventLogTemplate href="xs:anyURI"? >
4841     ... EventLogTemplate attributes ... ?
4842 </eventLogTemplate> ?
4843 <operation rel="edit" href="xs:anyURI"/> ?
4844 <operation rel="delete" href="xs:anyURI"/> ?
4845 <xs:any>*
4846 </NetworkPortTemplate>

```

4847 5.16.9.1 Operations

4848 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 4849 NetworkPortTemplateCollection Resource.

4850 5.16.10 NetworkPortTemplateCollection Resource

4851 A NetworkPortTemplateCollection Resource represents the Collection of
 4852 NetworkPortTemplates within a Provider and follows the Collection pattern defined in clause
 4853 5.5.12. This Resource shall be serialized as follows:

4854 JSON serialization:

```

4855 { "resourceURI":
4856     "http://schemas.dmtf.org/cimi/1/NetworkPortTemplateCollection",
4857     "id": string,
4858     "count": number,
4859     "networkPortTemplates": [
4860         { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortTemplate",
4861           "id": string,
4862           ... remaining NetworkPortTemplate attributes ...
4863         }, +
4864     ], ?
4865     "operations": [ { "rel": "add", "href": string } ? ]
4866     ...
4867 }

```

4868 **XML serialization:**

```

4869 <Collection
4870     resourceURI="http://schemas.dmtf.org/cimi/1/NetworkPortTemplateCollection"
4871     xmlns="http://schemas.dmtf.org/cimi/1">
4872     <id> xs:anyURI </id>
4873     <count> xs:integer </count>
4874     <NetworkPortTemplate>
4875         <id> xs:anyURI </id>
4876         ... remaining NetworkPortTemplate attributes ...
4877     </NetworkPortTemplate> *
4878     <operation rel="add" href="xs:anyURI"/> ?
4879     <xs:any*>
4880 </Collection>
    
```

4881 **5.16.10.1 Operations**

4882 This Resource supports the Read and Update operations. Creation of new NetworkPortTemplate
 4883 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 4884 4.2.1.1.

4885 **5.16.11 NetworkPortConfiguration Resource**

4886 The set of configuration values representing the information needed to create a NetworkPort with
 4887 certain characteristics. Table 31 describes the NetworkPortConfiguration attributes.

4888 **Table 31 – NetworkPortConfiguration attributes**

Name	NetworkPortConfiguration	
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. <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

4889 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

4890 **JSON media type:** application/json

4891 **JSON serialization:**

```
4892 { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration",
4893   "id": string,
4894   "name": string, ?
4895   "description": string, ?
4896   "created": string, ?
4897   "updated": string, ?
4898   "properties": { string: string, + }, ?
4899   "portType": string, ?
4900   "classOfService": string, ?
4901   "operations": [
4902     { "rel": "edit", "href": string }, ?
4903     { "rel": "delete", "href": string } ?
4904   ] ?
4905   ...
4906 }
```

4907 **XML media type:** application/xml

4908 **XML serialization:**

```
4909 <NetworkPortConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
4910   <id> xs:anyURI </id>
4911   <name> xs:string </name> ?
4912   <description> xs:string </description> ?
4913   <created> xs:dateTime </created> ?
4914   <updated> xs:dateTime </updated> ?
4915   <property key="xs:string"> xs:string </property> *
4916   <portType> xs:string </portType> ?
4917   <classOfService> xs:string </classOfService> ?
4918   <operation rel="edit" href="xs:anyURI"/> ?
4919   <operation rel="delete" href="xs:anyURI"/> ?
4920   <xs:any>*
4921 </NetworkPortConfiguration>
```

4922 5.16.11.1 Operations

4923 This Resource supports the Read, Update, and Delete operations. Create is supported through the
4924 NetworkPortConfigurationCollection Resource.

4925 5.16.12 NetworkPortConfigurationCollection Resource

4926 A `NetworkPortConfigurationCollection` Resource represents the Collection of
 4927 `NetworkPortConfigurations` within a Provider and follows the Collection pattern defined in
 4928 clause 5.5.12. This Resource shall be serialized as follows:

4929 JSON serialization:

```

4930 { "resourceURI":
4931     "http://schemas.dmtf.org/cimi/1/NetworkPortConfigurationCollection",
4932     "id": string,
4933     "count": number,
4934     "networkPortConfigurations": [
4935         { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration",
4936           "id": string,
4937           ... remaining NetworkPortConfiguration attributes ...
4938         }, +
4939     ], ?
4940     "operations": [ { "rel": "add", "href": string }? ]
4941     ...
4942 }
```

4943 XML serialization:

```

4944 <Collection
4945 resourceURI="http://schemas.dmtf.org/cimi/1/NetworkPortConfigurationCollection"
4946     xmlns="http://schemas.dmtf.org/cimi/1">
4947     <id> xs:anyURI </id>
4948     <count> xs:integer </count>
4949     <NetworkPortConfiguration>
4950         <id> xs:anyURI </id>
4951         ... remaining NetworkPortConfiguration attributes ...
4952     </NetworkPortConfiguration> *
4953     <operation rel="add" href="xs:anyURI" /> ?
4954     <xs:any>*
4955 </Collection>
```

4956 5.16.12.1 Operations

4957 This Resource supports the Read and Update operations. Creation of new
 4958 `NetworkPortConfiguration` Resources is supported by the way of a POST to the "add"
 4959 operation's URI as described in clause 4.2.1.1.

4960 5.16.13 Address Resource

4961 An `Address` represents an IP address, and its associated metadata, for a particular Network. If a
 4962 Consumer creates an `Address` Resource, it is the semantic equivalent of asking for a static IP address
 4963 that can then be associated with Resources at a later point in time. `Addresses` that are manually

4964 created by Consumers shall not be deleted automatically if the Resource (e.g., a *Machine*) that is using
 4965 that *Address* is deleted because these manually created *Addresses* are expected to have a lifetime
 4966 that is different from the Resources that use them. *Addresses* that are created by Providers on the
 4967 Consumer's behalf shall be deleted at the Provider's discretion. In particular, the Provider shall delete
 4968 *Addresses* that it created on behalf of the Consumer if the Resource that is using that *Address* is
 4969 deleted or if the *Address* becomes disassociated from the Resource.

4970 *Addresses* that are created by Providers may be converted to ones that are under the Consumer's
 4971 control (i.e., are not deleted until explicitly requested by the Consumer) by changing the "allocation"
 4972 attribute from "dynamic" to "static," if this feature supported by Providers.

4973 Table 32 describes the *Address* attributes.

4974

Table 32 – 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 Consumer: support optional; read-write
allocation	string	The value is either " dynamic " or " static ". Expresses whether this <i>Address</i> is controlled by the Provider or Consumer. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
defaultGateway	string	An IP address of a router that serves other networks. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
dns	string[]	The IP addresses of the Domain Name Services for host name to IP resolution. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
protocol	string	The selected network protocol, such as IPv4 or IPv6. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
mask	string	The network mask associated with this <i>Address</i> . Constraints: Provider: support optional; mutable Consumer: support optional; read-write
network	ref	A reference to the <i>Network</i> with which this <i>Address</i> is associated. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
resource	ref	A reference to the Resource that is using this <i>Address</i> . Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only

4975 When implementing or using *Address*, Providers and Consumers shall adhere to the syntax and
 4976 semantics of its attributes as described in Table 32 as well as in the table describing related Collections.

4977 Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-
 4978 schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

4979 **JSON media type:** application/json

4980 **JSON serialization:**

```

4981 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Address",
4982   "id": string,
4983   "name": string, ?
4984   "description": string, ?
4985   "created": string, ?
4986   "updated": string, ?
4987   "properties": { string: string, + }, ?
4988   "ip": string,
4989   "hostname": string, ?
4990   "allocation": string,
4991   "defaultGateway": string, ?
4992   "dns": [ string, + ], ?
4993   "protocol": string,
4994   "mask": string, ?
4995   "network": { "href": string },
4996   "resource": { "href": string }, ?
4997   "operations": [
4998     { "rel": "edit", "href": string }, ?
4999     { "rel": "delete", "href": string } ?
5000   ] ?
5001   ...
5002 }
```

5003 **XML media type:** application/xml

5004 **XML serialization:**

```

5005 <Address xmlns="http://schemas.dmtf.org/cimi/1">
5006   <id> xs:anyURI </id>
5007   <name> xs:string </name> ?
5008   <description> xs:string </description> ?
5009   <created> xs:dateTime </created> ?
5010   <updated> xs:dateTime </updated> ?
5011   <property key="xs:string"> xs:string </property> *
5012   <ip> xs:string </ip>
5013   <hostname> xs:string </hostname> ?
5014   <allocation> xs:string </allocation>
5015   <defaultGateway> xs:string </defaultGateway> ?
```

```

5016     <dns> xs:string </dns> *
5017     <protocol> xs:string </protocol>
5018     <mask> xs:string </mask> ?
5019     <network href="xs:anyURI"/>
5020     <resource href="xs:anyURI"/> ?
5021     <operation rel="edit" href="xs:anyURI"/> ?
5022     <operation rel="delete" href="xs:anyURI"/> ?
5023     <xs:any>*
5024 </Address>

```

5025 5.16.13.1 Operations

5026 This Resource supports the Read, Update, and Delete operations. Create is supported through the
5027 AddressCollection Resource.

5028 5.16.14 AddressCollection Resource

5029 An AddressCollection Resource represents the Collection of Addresses within a Provider that
5030 are owned/managed by the Consumer or Provider and follows the Collection pattern defined in clause
5031 5.5.12. This Resource shall be serialized as follows:

5032 JSON serialization:

```

5033 { "resourceURI": "http://schemas.dmtf.org/cimi/1/AddressCollection",
5034   "id": string,
5035   "count": number,
5036   "addresses": [
5037     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Address",
5038       "id": string,
5039       ... remaining Address attributes ...
5040     }, +
5041   ], ?
5042   "operations": [ { "rel": "add", "href": string } ? ]
5043   ...
5044 }

```

5045 XML serialization:

```

5046 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/AddressCollection"
5047   xmlns="http://schemas.dmtf.org/cimi/1">
5048   <id> xs:anyURI </id>
5049   <count> xs:integer </count>
5050   <Address>
5051     <id> xs:anyURI </id>
5052     ... remaining Address attributes ...
5053   </Address> *
5054   <operation rel="add" href="xs:anyURI"/> ?

```

5055 <xs:any>
 5056 </Collection>

5057 **5.16.14.1 Operations**

5058 NOTE The "add" operation requires that an AddressTemplate be used (see 4.2.1.1).

5059 **5.16.15 AddressTemplate Resource**

5060 This Resource captures the configuration values for realizing an Address. An AddressTemplate may
 5061 be used to create multiple Addresses. Table 33 describes the AddressTemplate attributes.

5062 **Table 33 – 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 Consumer: support optional; read-write
allocation	string	A value of either "dynamic" or "static". Expresses whether this address is controlled by the Provider or Consumer. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
defaultGateway	string	An IP address of a router that serves other networks. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
dns	string[]	The IP addresses of the Domain Name Services for host name to IP resolution. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
protocol	string	The selected network protocol, such as IPv4 or IPv6. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
mask	string	The network mask associated with this Address. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
network	ref	A reference to the Network with which this Address is associated. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

5063 When implementing or using AddressTemplate, Providers and Consumers shall adhere to the syntax
 5064 and semantics of its attributes as described in Table 33 as well as in the table describing the related
 5065 AddressTemplateCollection. Both Consumer and Provider shall serialize this Resource as
 5066 described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 5067 Resource in both JSON and XML.

5068 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

5069 **JSON media type:** application/json

5070 **JSON serialization:**

```

5071 { "resourceURI": "http://schemas.dmtf.org/cimi/1/AddressTemplate",
5072     "id": string,
5073     "name": string, ?
5074     "description": string, ?
5075     "created": string, ?
5076     "updated": string, ?
5077     "properties": { string: string, + }, ?
5078     "ip": string,
5079     "hostname": string, ?
5080     "allocation": string,
5081     "defaultGateway": string, ?
5082     "dns": [ string, + ], ?
5083     "protocol": string,
5084     "mask": string, ?
5085     "network": { "href": string },
5086     "operations": [
5087         { "rel": "edit", "href": string }, ?
5088         { "rel": "delete", "href": string } ?
5089     ] ?
5090     ...
5091 }
```

5092 **XML media type:** application/xml

5093 **XML serialization:**

```

5094 <AddressTemplate xmlns="http://schemas.dmtf.org/cimi/1">
5095   <id> xs:anyURI </id>
5096   <name> xs:string </name> ?
5097   <description> xs:string </description> ?
5098   <created> xs:dateTime </created> ?
5099   <updated> xs:dateTime </updated> ?
5100   <property key="xs:string"> xs:string </property> *
5101   <ip> xs:string </ip>
5102   <hostname> xs:string </hostname> ?
5103   <allocation> xs:string </allocation>
5104   <defaultGateway> xs:string </defaultGateway>
5105   <dns> xs:string </dns> +
5106   <protocol> xs:string </protocol>
5107   <mask> xs:string </mask>
```

```

5108     <network href="xs:anyURI"/>
5109     <operation rel="edit" href="xs:anyURI"/> ?
5110     <operation rel="delete" href="xs:anyURI"/> ?
5111     <xs:any>*
5112 </AddressTemplate>
    
```

5113 5.16.15.1 Operations

5114 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 5115 AddressTemplateCollection Resource.

5116 5.16.16 AddressTemplateCollection Resource

5117 An AddressTemplateCollection Resource represents the Collection of AddressTemplate
 5118 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
 5119 shall be serialized as follows:

5120 JSON serialization:

```

5121     { "resourceURI": "http://schemas.dmtf.org/cimi/1/AddressTemplateCollection",
5122       "id": string,
5123       "count": number,
5124       "addressTemplates": [
5125         { "resourceURI": "http://schemas.dmtf.org/cimi/1/AddressTemplate",
5126           "id": string,
5127           ... remaining AddressTemplate attributes ...
5128         }, +
5129       ], ?
5130       "operations": [ { "rel": "add", "href": string } ? ]
5131       ...
5132     }
    
```

5133 XML serialization:

```

5134     <Collection
5135       resourceURI="http://schemas.dmtf.org/cimi/1/AddressTemplateCollection"
5136       xmlns="http://schemas.dmtf.org/cimi/1">
5137       <id> xs:anyURI </id>
5138       <count> xs:integer </count>
5139       <AddressTemplate>
5140         <id> xs:anyURI </id>
5141         ... remaining AddressTemplate attributes ...
5142       </AddressTemplate> *
5143       <operation rel="add" href="xs:anyURI"/> ?
5144       <xs:any>*
5145     </Collection>
    
```

5146 **5.16.16.1 Operations**

5147 This Resource supports the Read and Update operations. Creation of new `AddressTemplate`
 5148 Resources is supported by the way of a POST to the "add" URI as described in clause 4.2.1.1.

5149 **5.16.17 ForwardingGroup Resource**

5150 A `ForwardingGroup` represents a collection of `Networks` that route to each other.

5151 `Networks` in a `ForwardingGroup` should all have the same "networkType" attributes, which
 5152 prevents a `Network` with a "private" networkType attribute from being publicly forwarded because it is a
 5153 member of a `ForwardingGroup` that also contains `Networks` with a "public" networkType attribute.

5154 Providers shall not allow two `Networks` to be forwardable to each other unless they are explicitly
 5155 connected by being part of a common `ForwardingGroup`.

5156 Table 34 describes the `ForwardingGroup` attributes.

5157 **Table 34 – ForwardingGroup attributes**

Name	ForwardingGroup	
Type URI	http://schemas.dmtf.org/cimi/1/ForwardingGroup	
Attribute	Type	Description
networks	<i>collection</i> <i>[Network]</i>	A reference to the list of references to the <code>Networks</code> in this <code>ForwardingGroup</code> . Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only

5158 When implementing or using `ForwardingGroup`, Providers and Consumers shall adhere to the syntax
 5159 and semantics of its attributes as described in Table 34 as well as in the tables describing embedded
 5160 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 5161 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 5162 both JSON and XML.

5163 **JSON media type:** application/json

5164 **JSON serialization:**

```

5165 { "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroup",
5166   "id": string,
5167   "name": string, ?
5168   "description": string, ?
5169   "created": string, ?
5170   "updated": string, ?
5171   "properties": { string: string, + }, ?
5172   "networks": [
5173     { "href": string }, +
5174   ], ?
5175   "operations": [
5176     { "rel": "edit", "href": string }, ?
5177     { "rel": "delete", "href": string } ?
5178   ] ?
    
```


5179 ...
 5180 }

5181 **XML media type:** application/xml

5182 **XML serialization:**

```
5183     <ForwardingGroup xmlns="http://schemas.dmtf.org/cimi/1">
5184         <id> xs:anyURI </id>
5185         <name> xs:string </name> ?
5186         <description> xs:string </description> ?
5187         <created> xs:dateTime </created> ?
5188         <updated> xs:dateTime </updated> ?
5189         <property key="xs:string"> xs:string </property> *
5190         <network href="xs:anyURI"> *
5191         <operation rel="edit" href="xs:anyURI"/> ?
5192         <operation rel="delete" href="xs:anyURI"/> ?
5193         <xs:any>*
5194     </ForwardingGroup>
```

5195 **5.16.17.1 Collections**

5196 The following clauses describe the Collection Resources owned by ForwardingGroups.

5197 **5.16.17.1.1 networks Collection**

5198 The Resource type for each item of this Collection is “Network”. There is no accessory attribute for the
 5199 items in this Collection; therefore, it is a basic Network Collection (serialized as described in 5.5.12).

5200 See the NetworkCollection Resource clause.

5201 **5.16.17.2 Operations**

5202 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 5203 ForwardingGroupCollection Resource.

5204 **5.16.18 ForwardingGroupCollection Resource**

5205 A ForwardingGroupCollection Resource represents the Collection of ForwardingGroups
 5206 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
 5207 serialized as follows:

5208 **JSON serialization:**

```
5209     { "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroupCollection",
5210       "id": string,
5211       "count": number,
5212       "forwardingGroups": [
5213         { "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroup",
5214           "id": string,
5215           ... remaining ForwardingGroup attributes ...
```

```

5216     }, +
5217   ], ?
5218   "operations": [ { "rel": "add", "href": string } ? ]
5219   ...
5220 }
    
```

5221 **XML serialization:**

```

5222 <Collection
5223     resourceURI="http://schemas.dmtf.org/cimi/1/ForwardingGroupCollection"
5224     xmlns="http://schemas.dmtf.org/cimi/1">
5225   <id> xs:anyURI </id>
5226   <count> xs:integer </count>
5227   <ForwardingGroup>
5228     <id> xs:anyURI </id>
5229     ... remaining ForwardingGroup attributes ...
5230   </ForwardingGroup> *
5231   <operation rel="add" href="xs:anyURI"/> ?
5232   <xs:any>*
5233 </Collection>
    
```

5234 **5.16.18.1 Operations**

5235 NOTE The "add" operation requires that a ForwardingGroupTemplate be used (see 4.2.1.1).

5236 **5.16.19 ForwardingGroupTemplate Resource**

5237 This Resource captures the configuration values for realizing a ForwardingGroup. A
 5238 ForwardingGroupTemplate may be used to create multiple ForwardingGroups. Table 35
 5239 describes the ForwardingGroupTemplate attributes.

5240 **Table 35 – ForwardingGroupTemplate attributes**

Name	ForwardingGroupTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/ForwardingGroupTemplate	
Attribute	Type	Description
networks	ref[]	An array of references to the Networks in this ForwardingGroup. Array item name: network Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

5241 When implementing or using ForwardingGroupTemplate, Providers and Consumers shall adhere
 5242 to the syntax and semantics of its attributes as described in Table 35 as well as in the tables describing
 5243 referred Resources. Both Consumer and Provider shall serialize this Resource as described below. The
 5244 following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON
 5245 and XML.

5246 **JSON media type:** application/json

5247 **JSON serialization:**

```
5248 { "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroupTemplate",
5249   "id": string,
5250   "name": string, ?
5251   "description": string, ?
5252   "created": string, ?
5253   "updated": string, ?
5254   "properties": { string: string, + }, ?
5255   "networks": [
5256     { "href": string }, +
5257   ], ?
5258   "operations": [
5259     { "rel": "edit", "href": string }, ?
5260     { "rel": "delete", "href": string } ?
5261   ] ?
5262   ...
5263 }
```

5264 **XML media type:** application/xml

5265 **XML serialization:**

```
5266 <ForwardingGroupTemplate xmlns="http://schemas.dmtf.org/cimi/1">
5267   <id> xs:anyURI </id>
5268   <name> xs:string </name> ?
5269   <description> xs:string </description> ?
5270   <created> xs:dateTime </created> ?
5271   <updated> xs:dateTime </updated> ?
5272   <property key="xs:string"> xs:string </property> *
5273   <network href="xs:anyURI"> *
5274   <operation rel="edit" href="xs:anyURI"/> ?
5275   <operation rel="delete" href="xs:anyURI"/> ?
5276   <xs:any>*
5277 </ForwardingGroupTemplate>
```

5278 5.16.19.1 Operations

5279 This Resource supports the Read, Update, and Delete operations. Create is supported through the
5280 ForwardingGroupTemplateCollection Resource.

5281 5.16.20 ForwardingGroupTemplateCollection Resource

5282 A ForwardingGroupTemplateCollection Resource represents the Collection of
5283 ForwardingGroupTemplate Resources within a Provider and follows the Collection pattern defined
5284 in clause 5.5.12. This Resource shall be serialized as follows:

5285 JSON serialization:

```
5286 { "resourceURI":
5287     "http://schemas.dmtf.org/cimi/1/ForwardingGroupTemplateCollection",
5288     "id": string,
5289     "count": number,
5290     "forwardingGroupTemplates": [
5291         { "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroupTemplate",
5292           "id": string,
5293           ... remaining ForwardingGroupTemplate attributes ...
5294         }, +
5295     ], ?
5296     "operations": [ { "rel": "add", "href": string }? ]
5297     ...
5298 }
```

5299 XML serialization:

```
5300 <Collection
5301     resourceURI="http://schemas.dmtf.org/cimi/1/ForwardingGroupTemplateCollection"
5302     xmlns="http://schemas.dmtf.org/cimi/1">
5303     <id> xs:anyURI </id>
5304     <count> xs:integer </count>
5305     <ForwardingGroupTemplate>
5306         <id> xs:anyURI </id>
5307         ... remaining ForwardingGroupTemplate attributes ...
5308     </ForwardingGroupTemplate> *
5309     <operation rel="add" href="xs:anyURI"/> ?
5310     <xs:any>*
5311 </Collection>
```

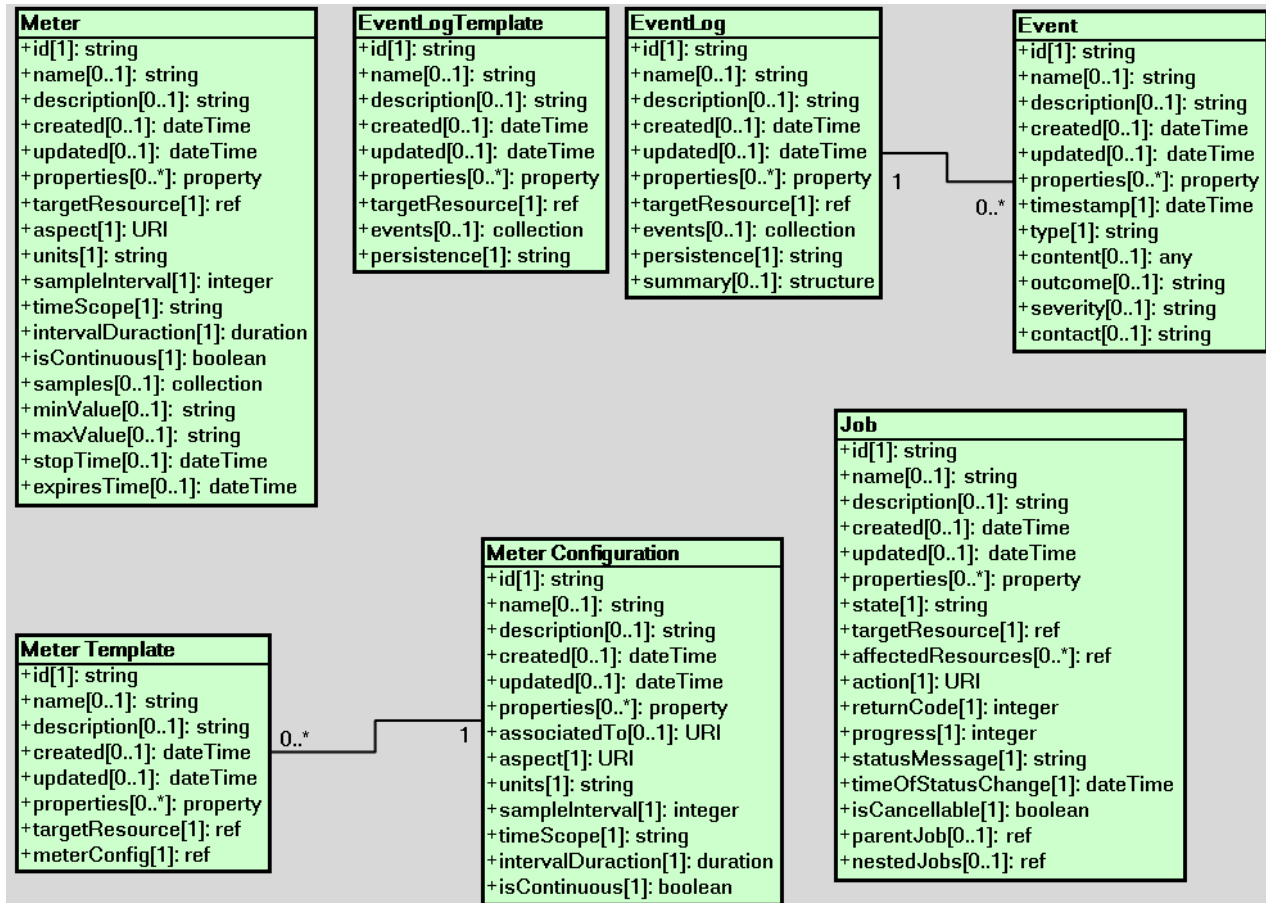
5312 5.16.20.1 Operations

5313 This Resource supports the Read and Update operations. Creation of new
5314 ForwardingGroupTemplate Resources is supported by the way of a POST to the "add" operation's
5315 URI as described in clause 4.2.1.1.

5316 5.17 Monitoring Resources and relationships

5317 Figure 6 illustrates the Resources involved in tracking the progress of operations, as well as, metering
5318 and monitoring the status of other Resources. Although this drawing is in the style of a Resource
5319 Relationship diagram, the use of UML is neither rigorous nor normative.

5320



5321

Figure 6 - Monitoring Resources

5322 **5.17.1 Job Resource**

5323 This Resource represents a process (i.e., a sequence of one or more operations directed to accomplish a
 5324 specific goal) that is performed by the Provider.

5325 If a Provider supports exposing Job Resources to Consumers, each request from a Consumer that the
 5326 Provider responds to with a 202 status code, shall result in a Job Resource being created and an
 5327 absolute URI reference to that Job Resource shall be made available to the requesting Consumer.
 5328 Providers may create additional Job Resources for Provider-initiated operations if the Provider chooses
 5329 to expose these Jobs to Consumers.

5330 If a Job is not completed successfully (e.g., it is in the FAILED or STOPPED state), this specification
 5331 does not place any requirements on the Provider to ensure that the affected Resources are left in certain
 5332 states. Based on the environmental conditions at that time, the Provider might choose to "undo" any
 5333 impact of the operation; simply halt processing; attempt some kind of "cleanup" action; or choose to do
 5334 something else. However, Providers shall list all Resources impacted by the Job in the
 5335 "affectedResources" attribute, thus allowing Consumers an opportunity to examine the state of each
 5336 Resource themselves. In cases where a Resource has been deleted, references to that Resource shall
 5337 not appear in the "affectedResources" attribute.

5338 The Job Resource allows for nesting of Jobs. The determination of when a single operation is
 5339 converted into multiple nested Jobs is out of scope of this specification. However, if there are nested

5340 Jobs, the top-most Job Resource shall report the overall status of all Jobs and shall only be in a
 5341 "SUCCESS" state if all nested Jobs are also in "SUCCESS" state. If nested Jobs are created, there is
 5342 no requirement for the top-most Job Resource to reference all affected Resources in its
 5343 "affectedResources" attribute. The Consumer needs to traverse the entire set of nested Jobs to
 5344 determine the complete list of Resources impacted by the Jobs.

5345 Table 36 describes the Job attributes.

5346 **Table 36 – Job attributes**

Name	Job	
Type URI	http://schemas.dmtf.org/cimi/1/Job	
Attribute	Type	Description
state	<i>string</i>	The state of the process associated with this operation. Allowable values include: QUEUED : Indicates that the operation has not yet begun processing. RUNNING : Indicates that the operation is still being executed. FAILED : Indicates that the operation failed to be completed successfully. SUCCESS : Indicates that the operation was successfully completed. STOPPING : Indicates that the operation is in the process of being stopped. STOPPED : Indicates that the operation was stopped before completion. The operations that result in transitions to the above defined states are defined in clause 5.17.1.1 <u>Constraints:</u> Provider : support mandatory; mutable Consumer : support mandatory; read-only
targetResource	<i>ref</i>	A reference to the top-level Resource upon which the operation is being performed. Typically, this Resource would be the Resource on which the operation was invoked. Note that if an "add" Job is executed against a "Collection" Resource (e.g., MachineCollection), the targetResource attribute shall reference the Collection Resource as that is the Resource on which the operation was performed. Additionally, the newly created Resource shall appear in the "affectedResources" attribute. <u>Constraints:</u> Provider : support mandatory; immutable Consumer : support mandatory; read-only
affectedResources	<i>ref[]</i>	A list of references to Resources that have been impacted by this Job. Note that this list shall always contain the "targetResource" reference. Array item name: affectedResource <u>Constraints:</u> Provider : support mandatory; mutable Consumer : support mandatory; read-only
action	<i>URI</i>	A URI that indicates the type of action being performed. <u>Constraints:</u> Provider : support mandatory; immutable Consumer : support mandatory; read-only
returnCode	<i>integer</i>	The operation return code. The specific value is specific to the implementation. Values in the range of 0 to 9999 are reserved for use by this specification. <u>Constraints:</u> Provider : support mandatory; mutable Consumer : support mandatory; read-only
progress	<i>integer</i>	An integer value in the range 0 ... 100 that indicates the progress of this Job. This value shall be 100 if the Job is no longer executing, regardless of the outcome. <u>Constraints:</u> Provider : support mandatory; mutable Consumer : support mandatory; read-only

Name	Job	
Type URI	http://schemas.dmtf.org/cimi/1/Job	
Attribute	Type	Description
statusMessage	<i>string</i>	A human-readable string that provides information about the operation. It is used to further qualify or provide additional information about the current status of the operation. For example, this attribute may indicate the reason why the operation failed, or whether the operation was cancelled by the Consumer or the Provider. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
timeOfStatusChange	<i>dateTime</i>	A timestamp indicating the last time that the status of the operation changed. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
parentJob	<i>ref</i>	A reference to the <code>Job</code> of which this Resource is a subordinate. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
nestedJobs	<i>ref[]</i>	An array of references to a set of subordinate <code>Job</code> Resources. Array item name: <code>nestedJob</code> Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only

5347 When implementing or using `Job`, Providers and Consumers shall adhere to the syntax and semantics of
 5348 its attributes as described in Table 36 as well as in the tables describing referred Resources or related
 5349 Collections. Both Consumer and Provider shall serialize this Resource as described below. The following
 5350 pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

5351 **JSON media type:** application/json

5352 **JSON serialization:**

```
5353 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Job",
5354   "id": string,
5355   "name": string, ?
5356   "description": string, ?
5357   "created": string, ?
5358   "updated": string, ?
5359   "properties": { string: string, + }, ?
5360   "state": string,
5361   "targetResource": { "href": string },
5362   "affectedResources": [ { "href": string }, + ],
5363   "action": string,
5364   "returnCode": number,
5365   "progress": number,
5366   "statusMessage": string,
5367   "timeOfStatusChange": date,
5368   "parentJob": { "href": string }, ?
5369   "nestedJobs": [
5370     { "href": string }, +
```

```

5371     ], ?
5372     "operations": [
5373         { "rel": "edit", "href": string }, ?
5374         { "rel": "delete", "href": string }, ?
5375         { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
5376     ] ?
5377     ...
5378 }

```

5379 **XML media type:** application/xml

5380 **XML serialization:**

```

5381 <Job xmlns="http://schemas.dmtf.org/cimi/1">
5382     <id> xs:anyURI </id>
5383     <name> xs:string </name> ?
5384     <description> xs:string </description> ?
5385     <created> xs:dateTime </created> ?
5386     <updated> xs:dateTime </updated> ?
5387     <property key="xs:string"> xs:string </property> *
5388     <state> xs:string </state>
5389     <targetResource href="xs:anyURI"/>
5390     <affectedResource href="xs:anyURI"/> +
5391     <action> xs:anyURI </action>
5392     <returnCode> xs:integer </returnCode>
5393     <progress> xs:integer <progress>
5394     <statusMessage> xs:string </statusMessage>
5395     <timeOfStatusChange> xs:dateTime </timeOfStatusChange>
5396     <parentJob href="xs:anyURI"/> ?
5397     <nestedJob href="xs:anyURI"/> *
5398     <operation rel="edit" href="xs:anyURI"/> ?
5399     <operation rel="delete" href="xs:anyURI"/> ?
5400     <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
5401 href="xs:anyURI"/> ?
5402     <xs:any>*
5403 </Job>

```

5404 5.17.1.1 Operations Resource

5405 This Resource supports the Read, Update, and Delete operations. Deleting a Job that is in the
5406 "RUNNING" state shall be the equivalent of first stopping the Job and then deleting it. A request to delete
5407 a running Job that does not support the "stop" action shall fail.

5408 The following custom operations are also defined:

5409 **stop**

5410 **/link@rel:** `http://schemas.dmtf.org/cimi/1/action/stop`

5411 This operation shall stop a Job.

5412 Input parameters: None.

5413 Output parameters: None.

5414 During the processing of this operation, the Job shall be in the "STOPPING" state.

5415 Upon successful completion of this operation, the Job shall be in the "STOPPED" state.

5416 **HTTP protocol**

5417 To stop a Job, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the Job where
5418 the HTTP request body shall be as described below.

5419 **JSON media type:** application/json

5420 **JSON serialization:**

```
5421 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
5422   "action": "http://schemas.dmtf.org/cimi/1/action/stop",
5423   "properties": { string: string, + } ?
5424   ...
5425 }
```

5426 **XML media type:** application/xml

5427 **XML serialization**

```
5428 <Action xmlns="http://schemas.dmtf.org/cimi/1">
5429   <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
5430   <property key="xs:string"> xs:string </property> *
5431   <xs:any>*
5432 </Action>
```

5433 Upon successful processing of the request, the HTTP response body may be empty.

5434 **5.17.2 JobCollection Resource**

5435 A JobCollection Resource represents the Collection of Jobs within a Provider and follows the
5436 Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

5437 **JSON serialization:**

```
5438 { "resourceURI": "http://schemas.dmtf.org/cimi/1/JobCollection",
5439   "id": string,
5440   "count": integer,
5441   "jobs": [
5442     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Job",
5443       "id": string,
5444       ... remaining Job attributes ...
5445     }, +
```

```
5446 ] ?
5447 ...
5448 }
```

5449 **XML serialization:**

```
5450 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/JobCollection"
5451     xmlns="http://schemas.dmtf.org/cimi/1">
5452     <id> xs:anyURI </id>
5453     <count> xs:integer </count>
5454     <Job>
5455         <id> xs:anyURI </id>
5456         ... remaining Job attributes ...
5457     </Job> *
5458     <xs:any>*
5459 </Collection>
```

5460 **5.17.3 Meter Resource**

5461 This Resource represents an available `Meter` of some property associated to a given Resource.

5462 If a `Meter`'s "targetResource" is deleted all `Meters` associated with that Resource shall also be
 5463 deleted. In other words, deleting a Resource-specific `MetersCollection` (e.g., a `Machine`'s
 5464 `MetersCollection`) shall also result in the deletion of the `Meters` referenced from that Collection.

5465 Table 37 describes the `Meter` attributes.

5466 **Table 37 – Meter attributes**

Name	Meter	
Type URI	http://schemas.dmtf.org/cimi/1/Meter	
Attribute	Type	Description
targetResource	<i>ref</i>	A reference to the Resource to which the <code>Meter</code> is related. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
aspect	<i>URI</i>	A unique identifier representing the aspect of the Resource being metered. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
units	<i>string</i>	The name of the used units, e.g., kilobits per second, CPU usage percentage, etc. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
sampleInterval	<i>integer</i>	The time between consecutive samples in seconds. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

Name	Meter	
Type URI	http://schemas.dmtf.org/cimi/1/Meter	
Attribute	Type	Description
timeScope	<i>string</i>	The time scope to which this meter's value applies. Two possible values: "Point" indicates that the Meter applies to a point in time. "Interval" indicates that the Meter applies to a time interval. For instance, it would be possible to define a Meter whose purpose is to provide the daily average CPU usage. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
intervalDuration	<i>duration</i>	The interval duration when the timeScope is set to "Interval". Possible values: hourly, daily, weekly, monthly, or yearly. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
isContinuous	<i>boolean</i>	This value indicates whether the Meter value is continuous or scalar. Performance Meters are an example of a linear metric. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
samples	<i>collection</i> [Sample]	A reference to the list of taken samples Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
minValue	<i>string</i>	The expected minimal measure value. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
maxValue	<i>string</i>	The expected maximum measure value. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
stopTime	<i>dateTime</i>	The time from which the meter stops tracking samples. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
expiresTime	<i>dateTime</i>	The time from which the Meter is not monitored anymore. It implies the deletion of the Meter after this time. Note that a Meter might be deleted before this time if the Resource being metered is deleted. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

5467 When implementing or using Meter, Providers and Consumers shall adhere to the syntax and semantics
 5468 of its attributes as described in Table 37 as well as in the tables describing related Collections. Both
 5469 Consumer and Provider shall serialize this Resource as described below. The following pseudo-schemas
 5470 (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

5471 **JSON media type:** application/json

5472 **JSON serialization:**

```
5473 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
5474   "id": string,
5475   "name": string, ?
5476   "description": string, ?
5477   "created": string, ?
```

```

5478     "updated": string, ?
5479     "properties": { string: string, + }, ?
5480     "targetResource": { "href": string },
5481     "aspect": string,
5482     "units": string,
5483     "sampleInterval": number,
5484     "timeScope": string,
5485     "intervalDuration": string,
5486     "isContinuous": boolean,
5487     "samples": { "href": string }, ?
5488     "minValue": string, ?
5489     "maxValue": string, ?
5490     "stopTime": string, ?
5491     "expiresTime": string, ?
5492     "operations": [
5493         { "rel": "edit", "href": string }, ?
5494         { "rel": "delete", "href": string }, ?
5495         { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
5496         { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
5497     ] ?
5498     ...
5499 }

```

5500 **XML media type:** application/xml

5501 **XML serialization:**

```

5502     <Meter xmlns="http://schemas.dmtf.org/cimi/1">
5503         <id> xs:anyURI </id>
5504         <name> xs:string </name> ?
5505         <description> xs:string </description> ?
5506         <created> xs:dateTime </created> ?
5507         <updated> xs:dateTime </updated> ?
5508         <property key="xs:string"> xs:string </property> *
5509         <targetResource href="xs:anyURI"/>
5510         <aspect> xs:anyURI </aspect>
5511         <units> xs:string </units>
5512         <sampleInterval> xs:integer </sampleInterval>
5513         <timeScope> xs:string <timeScope>
5514         <intervalDuration xs:duration </intervalDuration>
5515         <isContinuous> xs:boolean </isContinuous>
5516         <samples href="xs:anyURI"/> ?

```

```

5517 <minValue> xs:string </minValue> ?
5518 <maxValue> xs:string </maxValue> ?
5519 <stopTime> xs:dateTime </stopTime> ?
5520 <expiresTime> xs:dateTime </expiresTime> ?
5521 <operation rel="edit" href="xs:anyURI"/> ?
5522 <operation rel="delete" href="xs:anyURI"/> ?
5523 <operation rel="http://schemas.dmtf.org/cimi/1/action/start"
5524 href="xs:anyURI"/> ?
5525 <operation rel="http://schemas.dmtf.org/cimi/1/action/stop"
5526 href="xs:anyURI"/> ?
5527 <xs:any>*
5528 </Meter>

```

5529 **5.17.3.1 Collections**

5530 The following clauses describe the Collection resources owned by Meters.

5531 **5.17.3.1.1 SampleCollection Resource**

5532 The Resource type for each item of this Collection is "Sample", defined in Table 38:

5533 **Table 38 – Sample attributes**

Name		Sample
Type URI		http://schemas.dmtf.org/cimi/1/Sample
Attribute	Type	Description
timestamp	dateTime	Indicates when the measure was taken (timeScope="Point"). If the timeScope is "Interval", it indicates the end of the time interval. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
value	string	Indicates the sampled value of the measure. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only

5534 When implementing or using Sample, Providers and Consumers shall adhere to the syntax and
5535 semantics of its attributes as described in Table 38 as well as in the tables describing related Collections.
5536 Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-
5537 schemas (see notation in 1.3) describe the serialization of the Sample Collection in both JSON and
5538 XML.

5539 **JSON serialization:**

```

5540 { "resourceURI": "http://schemas.dmtf.org/cimi/1/SampleCollection",
5541   "id": string,
5542   "count": number,
5543   "samples": [
5544     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Sample",
5545       "id": string,
5546       "name": string, ?
5547       "description": string, ?

```

```

5548     "created": string, ?
5549     "updated": string, ?
5550     "properties": { string: string, + }, ?
5551     "timestamp": string,
5552     "value": string
5553     ...
5554 }, +
5555 ], ?
5556 ...
5557 }

```

5558 XML serialization:

```

5559 <Collection
5560     resourceURI="http://schemas.dmtf.org/cimi/1/SampleCollection"
5561     xmlns="http://schemas.dmtf.org/cimi/1">
5562     <id> xs:anyURI </id>
5563     <count> xs:integer </count>
5564     <Sample>
5565         <id> xs:anyURI </id>
5566         <name> xs:string </name> ?
5567         <description> xs:string </description> ?
5568         <created> xs:dateTime </created> ?
5569         <updated> xs:dateTime </updated> ?
5570         <property key="xs:string"> xs:string </property> *
5571         <sample timestamp="xs:dateTime" value="xs:string"/>
5572         <xs:any>*
5573     </Sample> *
5574     <xs:any>*
5575 </Collection>

```

5576 5.17.3.2 Operations

5577 This Resource supports the Read, Update, and Delete operations. Create is supported via the
5578 MeterCollection Resource. The deletion of a Meter shall remove the Meter from the
5579 targetResource's "meter" attribute.

5580 The following custom operations are also defined:

5581 start

5582 **/link@rel:** <http://schemas.dmtf.org/cimi/1/action/start>

5583 This operation shall start a Meter.

5584 Input parameters: None.

5585 Output parameters: None.

5586 Upon successful completion of this operation, the `Meter` shall start recording samples related to its
 5587 associated Resource.

5588 **HTTP protocol**

5589 To start a `Meter`, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/start" URI of the `Meter`
 5590 where the HTTP request body shall be as described below.

5591 **JSON media type:** application/json

5592 **JSON serialization:**

```
5593 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
5594   "action": "http://schemas.dmtf.org/cimi/1/action/start",
5595   "properties": { string: string, + } ?
5596   ...
5597 }
```

5598 **XML media type:** application/xml

5599 **XML serialization**

```
5600 <Action xmlns="http://schemas.dmtf.org/cimi/1">
5601   <action> http://schemas.dmtf.org/cimi/1/action/start </action>
5602   <property key="xs:string"> xs:string </property> *
5603   <xs:any>*
5604 </Action>
```

5605 Upon successful processing of the request, the HTTP response body may be empty.

5606 **stop**

5607 **/link@rel:** http://schemas.dmtf.org/cimi/1/action/stop

5608 This operation shall stop a `Meter`.

5609 Input parameters: None.

5610 Output parameters: None.

5611 Upon successful completion of this operation, the `Meter` shall no longer be recording samples related to
 5612 its associated Resource.

5613 **HTTP protocol**

5614 To stop a `Meter`, a POST is sent to the "http://schemas.dmtf.org/cimi/1/action/stop" URI of the `Meter`
 5615 where the HTTP request body shall be as described below.

5616 **JSON media type:** application/json

5617 **JSON serialization:**

```
5618 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
5619   "action": "http://schemas.dmtf.org/cimi/1/action/stop",
5620   "properties": { string: string, + } ?
5621   ...
```

5622 }

5623 **XML media type:** application/xml

5624 **XML serialization**

```
5625     <Action xmlns="http://schemas.dmtf.org/cimi/1">
5626         <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
5627         <property key="xs:string"> xs:string </property> *
5628         <xs:any>*
5629     </Action>
```

5630 Upon successful processing of the request, the HTTP response body may be empty.

5631 **5.17.4 MeterCollection Resource**

5632 A `MeterCollection` Resource represents the Collection of `Meters` within a Provider and follows the
5633 Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

5634 **JSON serialization:**

```
5635     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterCollection",
5636       "id": string,
5637       "count": number,
5638       "meters": [
5639         { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
5640           "id": string,
5641           ... remaining Meter attributes ...
5642         }, +
5643       ], ?
5644       "operations": [ { "rel": "add", "href": string } ? ]
5645       ...
5646     }
```

5647 **XML serialization:**

```
5648     <Collection resourceURI="http://schemas.dmtf.org/cimi/1/MeterCollection"
5649       xmlns="http://schemas.dmtf.org/cimi/1">
5650         <id> xs:anyURI </id>
5651         <count> xs:integer </count>
5652         <Meter>
5653             <id> xs:anyURI </id>
5654             ... remaining Meter attributes ...
5655         </Meter> *
5656         <operation rel="add" href="xs:anyURI"/> ?
5657         <xs:any>*
5658     </Collection>
```


5659 **5.17.4.1 Operations**

5660 NOTE The "add" operation requires that a `MeterTemplate` be used (see 4.2.1.1).

5661 If `Meters` are created through the global (Cloud Entry Point) `MeterCollection`'s "add" operation,
 5662 they shall be added automatically to the corresponding `targetResource`'s "Meters" Collection Resource
 5663 as well.

5664 **5.17.5 MeterTemplate Resource**

5665 A `MeterTemplate` represents the information needed to create a new `Meter`. Table 39 describes the
 5666 `MeterTemplate` attributes.

5667 **Table 39 – MeterTemplate attributes**

Name	MeterTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/MeterTemplate	
Attribute	Type	Description
targetResource	ref	A reference to the Resource that is metered. The type of the Resource shall be one of the "associatedTo" types listed in the <code>MeterConfiguration</code> referenced. If this Template is used to create a new <code>Meter</code> through the global (Cloud Entry Point) <code>MetersCollection</code> , this attribute shall be present. If this Template is used to create a new <code>Meter</code> through a <code>targetResource</code> 's <code>MetersCollection</code> , this attribute shall either be absent or have the same value as the "id" of the <code>targetResource</code> to which this <code>Meter</code> is being added. <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-write
meterConfig	ref	A reference to the <code>MeterConfiguration</code> that is used to create a <code>Meter</code> from this <code>MeterTemplate</code> . Note that the attributes of the <code>MeterConfiguration</code> may be specified rather than a reference to an existing <code>MeterConfiguration</code> Resource. <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-write

5668 When implementing or using `MeterTemplate`, Providers and Consumers shall adhere to the syntax
 5669 and semantics of its attributes as described in Table 39 as well as in the tables describing referred
 5670 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 5671 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 5672 both JSON and XML.

5673 **JSON media type:** application/json

5674 **JSON serialization:**

```
5675 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterTemplate",
5676   "id": string,
5677   "name": string, ?
5678   "description": string, ?
5679   "created": string, ?
5680   "updated": string, ?
5681   "properties": { string: string, + }, ?
5682   "targetResource": { string },
5683   "meterConfig": {
```

```

5684     "href": string | ... MeterConfiguration attributes ...
5685   },
5686   "operations": [
5687     { "rel": "edit", "href": string }, ?
5688     { "rel": "delete", "href": string } ?
5689   ] ?
5690   ...
5691 }

```

5692 **XML media type:** application/xml

5693 **XML serialization:**

```

5694 <MeterTemplate xmlns="http://schemas.dmtf.org/cimi/1">
5695   <id> xs:anyURI </id>
5696   <name> xs:string </name> ?
5697   <description> xs:string </description> ?
5698   <created> xs:dateTime </created> ?
5699   <updated> xs:dateTime </updated> ?
5700   <property key="xs:string"> xs:string </property> *
5701   <targetResource href="xs:anyURI"/>
5702   <meterConfig href="xs:anyURI"?>
5703     ... MeterConfiguration attributes ... ?
5704   </meterConfig>
5705   <operation rel="edit" href="xs:anyURI"/> ?
5706   <operation rel="delete" href="xs:anyURI"/> ?
5707   <xs:any>*
5708 </MeterTemplate>

```

5709 5.17.6 MeterTemplateCollection Resource

5710 A MeterTemplateCollection Resource represents the Collection of MeterTemplate
 5711 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
 5712 shall be serialized as follows:

5713 **JSON serialization:**

```

5714 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterTemplateCollection",
5715   "id": string,
5716   "count": number,
5717   "meterTemplates": [
5718     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterTemplate",
5719       "id": string,
5720       ... remaining MeterTemplate attributes ...
5721     }, +
5722   ], ?

```

```

5723     "operations": [ { "rel": "add", "href": string } ? ]
5724     ...
5725 }
    
```

5726 XML serialization:

```

5727 <Collection
5728     resourceURI="http://schemas.dmtf.org/cimi/1/MeterTemplateCollection"
5729     xmlns="http://schemas.dmtf.org/cimi/1">
5730 <id> xs:anyURI </id>
5731 <count> xs:integer </count>
5732 <MeterTemplate>
5733     <id> xs:anyURI </id>
5734     ... remaining MeterTemplate attributes ...
5735 </MeterTemplate> *
5736 <operation rel="add" href="xs:anyURI" /> ?
5737 <xs:any>*
5738 </Collection>
    
```

5739 5.17.6.1 Operations

5740 This Resource supports the Read and Update operations. Creation of new `MeterTemplate` Resources
 5741 is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.

5742 5.17.7 MeterConfiguration Resource

5743 A `MeterConfiguration` represents the definition of a `Meter`. Table 40 describes the
 5744 `MeterConfiguration` attributes.

5745 **Table 40 – MeterConfiguration attributes**

Name	MeterConfiguration	
Type URI	http://schemas.dmtf.org/cimi/1/MeterConfiguration	
Attribute	Type	Description
associatedTo	<i>URI[]</i>	An array of URIs that indicate the types of Resources to which a <code>Meter</code> created from this configuration can be applied. The value space of these URIs is identical to that of <code>ResourceMetadata.typeURI</code> , which is a URI that uniquely identifies a Resource type. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
aspect	<i>URI</i>	A unique identifier representing the aspect of the Resource being metered. See Table 41 below for the set of CIMI-defined URIs. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
units	<i>string</i>	The human-readable name of the used units, e.g., kilobits per second, CPU usage percentage, etc. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
sampleInterval	<i>integer</i>	The time between consecutive samples in seconds. Constraints: Provider: support mandatory; mutable

Name	MeterConfiguration	
Type URI	http://schemas.dmtf.org/cimi/1/MeterConfiguration	
Attribute	Type	Description
		Consumer: support mandatory; read-write
timeScope	<i>string</i>	The time scope to which the <i>Meter</i> value applies. Two possible values: "Point" indicates that the <i>Meter</i> applies to a point in time. "Interval" indicates that the <i>Meter</i> applies to a time interval. For instance, it would be possible to define a <i>MeterConfiguration</i> whose purpose is to provide the daily average CPU usage. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
intervalDuration	<i>duration</i>	The interval duration when the <i>timeScope</i> is set to "Interval." Possible values: hourly, daily, weekly, monthly, or yearly. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
isContinuous	<i>boolean</i>	This value indicates whether the <i>Meter</i> value is continuous or scalar. Performance Meters are an example of a linear metric. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

5746 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

5747 **JSON media type:** application/json

5748 **JSON serialization:**

```

5749 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterConfiguration",
5750   "id": string,
5751   "name": string, ?
5752   "description": string, ?
5753   "created": string, ?
5754   "updated": string, ?
5755   "properties": { string: string, + }, ?
5756   "associatedTo": [
5757     { "href": string }, +
5758   ], ?
5759   "aspect": string,
5760   "units": string,
5761   "sampleInterval": number,
5762   "timeScope": string,
5763   "intervalDuration": string,
5764   "isContinuous": boolean,
5765   "operations": [
5766     { "rel": "edit", "href": string }, ?
5767     { "rel": "delete", "href": string } ?
5768   ] ?
5769   ...

```

5770 }

5771 **XML media type:** application/xml

5772 **XML serialization:**

```

5773 <MeterConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
5774   <id> xs:anyURI </id>
5775   <name> xs:string </name> ?
5776   <description> xs:string </description> ?
5777   <created> xs:dateTime </created> ?
5778   <updated> xs:dateTime </updated> ?
5779   <property key="xs:string"> xs:string </property> *
5780   <associatedTo href="xs:anyURI"/> *
5781   <aspect> xs:anyURI </aspect>
5782   <units> xs:string </units>
5783   <sampleInterval> xs:integer </sampleInterval>
5784   <timeScope> xs:string </timeScope>
5785   <intervalDuration> xs:duration </intervalDuration>
5786   <isContinuous> xs:boolean </isContinuous>
5787   <operation rel="edit" href="xs:anyURI"/> ?
5788   <operation rel="delete" href="xs:anyURI"/> ?
5789   <xs:any>*
5790 </MeterConfiguration>
    
```

5791 Table 41 describes the "aspect" URIs defined by this specification. Providers may define new aspect
 5792 URIs and it is recommended that these URIs be dereferencable such that Consumers can discover the
 5793 details of the new aspect. For brevity the "URI" column in the table only shows the last part of the URI. It
 5794 should be appended to: "http://schemas.dmtf.org/cimi/1/aspect".

5795 **Table 41 – aspect URIs**

Aspect	Description
cpu	The percentage CPU usage of the Resource. Typically associated with CloudEntryPoint, System, and Machine Resources. For Resources that group other Resources (e.g., CloudEntryPoint or System Resources), this aspect provides the aggregated percentage usage of the CPU.
memory	The amount of memory being used by the Resource. Typically associated with CloudEntryPoint, System, and Machine Resources. For Resources that group other Resources (e.g., CloudEntryPoint or System Resources), this aspect provides the aggregated usage of the memory.
disk	The amount of disk being used by the Resource. Typically associated with CloudEntryPoint, System, Machine, and Volume Resources. For Resources that group other Resources (e.g., CloudEntryPoint or System Resources), this aspect provides the aggregated disk usage.
bandwidth	The amount of network traffic. Typically associated with CloudEntryPoint, System, and Network Resources. For CloudEntryPoint and System Resources, this aspect provides the aggregated bandwidth of all the networks under them.
inputBandwidth	The amount of input bandwidth used by the Resource. Typically associated with Machine, NetworkPort, and Volume Resources. For Machine Resources, this aspect provides the aggregated input bandwidth usage of all its network interfaces .

Aspect	Description
outputBandwidth	The amount of output bandwidth used by the Resource. Typically associated with Machine, NetworkPort, and Volume Resources. For Machine Resources, this aspect provides the aggregated output bandwidth usage of all its network interfaces.

5796 5.17.7.1 Operations

5797 This Resource supports the Read, Update, and Delete operations. Create is supported through the
5798 MeterConfigurationCollection Resource.

5799 5.17.8 MeterConfigurationCollection Resource

5800 A MeterConfigurationCollection Resource represents the Collection of
5801 MeterConfigurations within a Provider and follows the Collection pattern defined in clause 5.5.12.
5802 This Resource shall be serialized as follows:

5803 JSON serialization:

```
5804 { "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterConfigurationCollection",
5805   "id": string,
5806   "count": number,
5807   "meterConfigurations": [
5808     { "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterConfiguration",
5809       "id": string,
5810       ... remaining MeterConfiguration attributes ...
5811     }, +
5812   ], ?
5813   "operations": [ { "rel": "add", "href": string } ? ]
5814   ...
5815 }
```

5816 XML serialization:

```
5817 <Collection
5818   resourceURI="http://schemas.dmtf.org/cimi/1/MeterConfigurationCollection"
5819   xmlns="http://schemas.dmtf.org/cimi/1">
5820   <id> xs:anyURI </id>
5821   <count> xs:integer </count>
5822   <MeterConfiguration>
5823     <id> xs:anyURI </id>
5824     ... remaining MeterConfiguration attributes ...
5825   </MeterConfiguration> *
5826   <operation rel="add" href="xs:anyURI"/> ?
5827   <xs:any>*
5828 </Collection>
```

5829 **5.17.8.1 Operations**

5830 This Resource supports the Read and Update operations. Creation of new `MeterConfiguration`
 5831 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 5832 4.2.1.1.

5833 **5.17.9 EventLog Resource**

5834 A Resource that represents a registry of Events.

5835 If an `EventLog`'s "targetResource" is deleted the `EventLog` associated with that Resource may also
 5836 be deleted. In other words, deleting a Resource (e.g., a `Machine`) may also result in the deletion of the
 5837 `EventLog` referenced from that Resource. This behavior is denoted by the `EventLog` "Linked"
 5838 capability.

5839 If an `EventLog` is deleted, all of its Events shall also be deleted.

5840 Table 42 describes the `EventLog` attributes.

5841 **Table 42 – EventLog attributes**

Name	EventLog	
Type URI	http://schemas.dmtf.org/cimi/1/EventLog	
Attribute	Type	Description
targetResource	<i>ref</i>	A reference to the Resource to which the <code>Events</code> are related. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
events	<i>collection</i> <i>[Event]</i>	A reference to the list of occurred <code>Events</code> . Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
persistence	<i>string</i>	A value that indicates the persistence of the <code>Events</code> within the <code>EventLog</code> . For instance, daily, weekly, monthly, or yearly. Events that exceed the persistence duration may be deleted. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

Name	EventLog																
Type URI	http://schemas.dmtf.org/cimi/1/EventLog																
Attribute	Type	Description															
summary	<unnamed structure>	A summary of all the events present in the EventLog when the read operation is performed, grouped by severity. Each summary attribute is an (unnamed) structure that has the following sub-attributes:															
		<table border="1"> <thead> <tr> <th>Attribute</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>low</td> <td>integer</td> <td>Number of occurred Events with a low severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</td> </tr> <tr> <td>medium</td> <td>integer</td> <td>Number of occurred Events with a medium severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</td> </tr> <tr> <td>high</td> <td>integer</td> <td>Number of occurred Events with a high severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</td> </tr> <tr> <td>critical</td> <td>integer</td> <td>Number of occurred Events with a critical severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</td> </tr> </tbody> </table>	Attribute	Type	Description	low	integer	Number of occurred Events with a low severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only	medium	integer	Number of occurred Events with a medium severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only	high	integer	Number of occurred Events with a high severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only	critical	integer	Number of occurred Events with a critical severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
		Attribute	Type	Description													
		low	integer	Number of occurred Events with a low severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only													
		medium	integer	Number of occurred Events with a medium severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only													
		high	integer	Number of occurred Events with a high severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only													
critical	integer	Number of occurred Events with a critical severity. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only															
Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only																	

5842 When implementing or using EventLog, Providers and Consumers shall adhere to the syntax and
 5843 semantics of its attributes as described in Table 42 as well as in the tables describing embedded
 5844 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 5845 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 5846 both JSON and XML.

5847 **JSON media type:** application/json

5848 **JSON serialization:**

```
5849 { "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLog",
5850   "id": string,
5851   "name": string, ?
5852   "description": string, ?
5853   "created": string, ?
5854   "updated": string, ?
5855   "properties": { string: string, + }, ?
5856   "targetResource": { "href": string },
5857   "events": { "href": string },
5858   "persistence": string,
5859   "summary": {
5860     "low": number,
5861     "medium": number,
5862     "high": number,
5863     "critical": number
```



```

5864     }, ?
5865     "operations": [
5866         { "rel": "edit", "href": string }, ?
5867         { "rel": "delete", "href": string } ?
5868     ] ?
5869     ...
5870 }
    
```

5871 **XML media type:** application/xml

5872 **XML serialization:**

```

5873 <EventLog xmlns="http://schemas.dmtf.org/cimi/1">
5874   <id> xs:anyURI </id>
5875   <name> xs:string </name> ?
5876   <description> xs:string </description> ?
5877   <created> xs:dateTime </created> ?
5878   <updated> xs:dateTime </updated> ?
5879   <property key="xs:string"> xs:string </property> *
5880   <targetResource href="xs:anyURI"/>
5881   <events href="xs:anyURI"/>
5882   <persistence> xs:string </persistence>
5883   <summary>
5884     <low> xs:integer </low>
5885     <medium> xs:integer </medium>
5886     <high> xs:integer </high>
5887     <critical> xs:integer </critical>
5888   </summary>
5889   <operation rel="edit" href="xs:anyURI"/> ?
5890   <operation rel="delete" href="xs:anyURI"/> ?
5891   <xs:any>*
5892 </EventLog>
    
```

5893 5.17.9.1 Collections

5894 The following clauses describe the Collection Resources owned by EventLogs.

5895 5.17.9.1.1 events Collection

5896 The Resource type for each item of this Collection is “Event” as defined in clause 5.17.13.

5897 **JSON serialization:**

```

5898 { "resourceURI": "http://schemas.dmtf.org/cimi/1/EventCollection",
5899   "id": string,
5900   "count": number,
5901   "events": [
    
```

```

5902     { "resourceURI": "http://schemas.dmtf.org/cimi/1/Event",
5903       "id": string,
5904       ... remaining Event attributes ...
5905     }, +
5906   ], ?
5907   "operations": [ { "rel": "add", "href": string } ? ]
5908   ...
5909 }

```

5910 XML serialization:

```

5911 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/EventCollection"
5912   xmlns="http://schemas.dmtf.org/cimi/1">
5913   <id> xs:anyURI </id>
5914   <count> xs:integer </count>
5915   <Event>
5916     <id> xs:anyURI </id>
5917     ... remaining Event attributes ...
5918   </Event> *
5919   <operation rel="add" href="xs:anyURI"/> ?
5920   <xs:any>*
5921 </Collection>

```

5922 5.17.9.2 Operations

5923 This Resource supports the Read, Update, and Delete operations.

5924 5.17.10 EventLogCollection Resource

5925 An EventLogCollection Resource represents the Collection of EventLogs within a Provider and
 5926 follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

5927 JSON serialization:

```

5928 { "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogCollection",
5929   "id": string,
5930   "count": number,
5931   "eventLogs": [
5932     { "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLog",
5933       "id": string,
5934       ... remaining EventLog attributes ...
5935     }, +
5936   ], ?
5937   "operations": [ { "rel": "add", "href": string } ? ]
5938   ...
5939 }

```

5940 **XML serialization:**

```

5941 <Collection resourceURI="http://schemas.dmtf.org/cimi/1/EventLogCollection"
5942     xmlns="http://schemas.dmtf.org/cimi/1">
5943     <id> xs:anyURI </id>
5944     <count> xs:integer </count>
5945     <EventLog>
5946         <id> xs:anyURI </id>
5947         ... remaining EventLog attributes ...
5948     </EventLog> *
5949     <operation rel="add" href="xs:anyURI"/> ?
5950     <xs:any>*
5951 </Collection>
    
```

5952 **5.17.11 EventLogTemplate Resource**

5953 An EventLogTemplate represents the information needed to create a new EventLog. Table 43
 5954 describes the EventLogTemplate attributes.

5955 **Table 43 – EventLogTemplate attributes**

EventLogTemplate		
Name	EventLogTemplate	
Type URI	http://schemas.dmtf.org/cimi/1/EventLogTemplate	
Attribute	Type	Description
targetResource	ref	A reference to the Resource to which the EventLog shall be connected. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
persistence	string	A value that indicates the persistence of the Events in the new EventLog. For instance, daily, weekly, monthly, or yearly. Events that exceed the persistence duration may be deleted. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

5956 When implementing or using EventLogTemplate, Providers and Consumers shall adhere to the
 5957 syntax and semantics of its attributes as described in Table 43 as well as in the tables describing referred
 5958 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 5959 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 5960 both JSON and XML.

5961 **JSON media type:** application/json

5962 **JSON serialization:**

```

5963 { "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogTemplate",
5964   "id": string,
5965   "name": string, ?
5966   "description": string, ?
5967   "created": string, ?
5968   "updated": string, ?
5969   "properties": { string: string, + }, ?
    
```

```

5970     "targetResource": { string },
5971     "persistence": string,
5972     "operations": [
5973         { "rel": "edit", "href": string }, ?
5974         { "rel": "delete", "href": string } ?
5975     ] ?
5976     ...
5977 }

```

5978 **XML media type:** application/xml

5979 **XML serialization:**

```

5980 <EventLogTemplate xmlns="http://schemas.dmtf.org/cimi/1">
5981     <id> xs:anyURI </id>
5982     <name> xs:string </name> ?
5983     <description> xs:string </description> ?
5984     <created> xs:dateTime </created> ?
5985     <updated> xs:dateTime </updated> ?
5986     <property key="xs:string"> xs:string </property> *
5987     <targetResource href="xs:anyURI"/>
5988     <persistence> xs:string </persistence>
5989     <operation rel="edit" href="xs:anyURI"/> ?
5990     <operation rel="delete" href="xs:anyURI"/> ?
5991     <xs:any>*
5992 </EventLogTemplate>

```

5993 **5.17.12 EventLogTemplateCollection Resource**

5994 An EventLogTemplateCollection Resource represents the Collection of EventLogTemplate
5995 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
5996 shall be serialized as follows:

5997 **JSON serialization:**

```

5998 { "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogTemplateCollection",
5999   "id": string,
6000   "count": number,
6001   "eventLogTemplates": [
6002     { "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogTemplate",
6003       "id": string,
6004       ... remaining EventLogTemplate attributes ...
6005     }, +
6006   ], ?
6007   "operations": [ { "rel": "add", "href": string } ? ]
6008   ...

```

6009 }

6010 **XML serialization:**

```

6011 <Collection
6012     resourceURI="http://schemas.dmtf.org/cimi/1/EventLogTemplateCollection"
6013     xmlns="http://schemas.dmtf.org/cimi/1">
6014     <id> xs:anyURI </id>
6015     <count> xs:integer </count>
6016     <EventLogTemplate>
6017         <id> xs:anyURI </id>
6018         ... remaining EventLogTemplate attributes ...
6019     </EventLogTemplate> *
6020     <operation rel="add" href="xs:anyURI"/> ?
6021     <xs:any>*
6022 </Collection>
    
```

6023 **5.17.12.1 Operations**

6024 This Resource supports the Read and Update operations. Creation of new `EventLogTemplate`
 6025 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 6026 4.2.1.1.

6027 **5.17.13 Event Resource**

6028 A Resource that represents the occurrence of an event within the managed infrastructure. Some
 6029 examples of `Event` are:

- 6030 • Machine X has been rebooted by guest OS.
- 6031 • Machine X is not responding to platform services.
- 6032 • A new vCPU has been added to machine X following defined elasticity rules.

6033 The scope of the `Event` concept is any information that the Provider is able to track within its
 6034 infrastructure and that can constitute useful information for the Consumer. Possible examples include, but
 6035 are not limited to, errors and inconveniences that occur in the (virtual) resources assigned to Consumers;
 6036 Provider-initiated actions, such as maintenance tasks; etc.

6037 Table 44 describes the `Event` attributes.

6038 **Table 44 – Event attributes**

Name	Event	
Type URI	http://schemas.dmtf.org/cimi/1/Event	
Attribute	Type	Description
timestamp	<i>dateTime</i>	The time of occurrence of the actual <code>Event</code> . NOTE: This attribute should not be confused with the time of creation of the <code>Event</code> Resource instance, which is captured in the common "created" attribute. Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only

Name	Event	
Type URI	http://schemas.dmtf.org/cimi/1/Event	
Attribute	Type	Description
type	<i>URI</i>	<p>A URI that uniquely identifies the type of the <code>Event</code>. If the "content" attribute is present, this URI determines the actual data structure used for this content, e.g., to which schema it is associated.</p> <p>Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only</p>
content	<i>any</i>	<p>A polymorphic attribute that represents detailed event data, the type of which varies with the <code>Event</code> "type." Typically, a data structure; for example:</p> <p>In the case of a monitoring event, the content shall hold the target Resource ID and type, measured attribute(s), and status value(s).</p> <p>In the case of an audit event conforming to the CADF model, the content shall hold the detailed event structure that complies with CADF event schema.</p> <p>In the case of a CIM Indication, the content shall hold the structure and attributes defined for such events.</p> <p>Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only</p>
outcome	<i>string</i>	<p>A string value that characterizes the general significance of the <code>Event</code>. A core set is defined that may be used regardless of the <code>Event</code> type. For each <code>Event</code> type, the definition of a core outcome value maybe refined in the context of this type, provided it does not conflict with the general meaning of the outcome given below.</p> <p>Core outcomes are:</p> <p>Pending: The <code>Event</code> is about an action or process that is still ongoing.</p> <p>Unknown: The <code>Event</code> is about a request or action that is not known by the Provider.</p> <p>Status: The <code>Event</code> reports on the state or status of a Resource.</p> <p>Success: The <code>Event</code> reports on a successful outcome of some action or process.</p> <p>Warning: The <code>Event</code> reports on a situation that requires attention or remedial action.</p> <p>Failure: The <code>Event</code> reports on a failed outcome of some action or process.</p> <p>This set of core outcome values may be extended to accommodate possible outcomes of a specific <code>Event</code> type. In this case, the extended set of values shall apply to all <code>Events</code> of this type.</p> <p>Constraints: Provider: support optional; immutable Consumer: support optional; read-only</p>
severity	<i>string</i>	<p>A value indicating the <code>Event</code> severity. Possible values are:</p> <p>critical high medium low</p> <p>The meaning of the severity level may vary depending on the <code>Event</code> "type." If such an attribute is not relevant to a particular type of <code>Event</code>, it should be omitted.</p> <p>Constraints: Provider: support optional; immutable Consumer: support optional; read-only</p>
contact	<i>string</i>	<p>A reference to a contact point or processing point to handle the <code>Event</code>. The actual type of this content (e.g., email address, phone number of helpdesk or staff, message queue, URL...) is dependent on, and determined by the <code>Event</code> "type." This attribute is mutable as it may be determined after <code>Event</code> creation by the Provider.</p> <p>Constraints: Provider: support optional; immutable Consumer: support optional; read-only</p>

6039 NOTE There exists a legacy of several `Event` models that have been standardized or designed for various
 6040 domains relevant to IT. The objective in CIMI is not to elect one particular `Event` model, but to select as top-level
 6041 `Event` attributes the most immediately relevant data useful for `Event` processing in a Cloud environment.

6042 Additional `Event` data may still be represented in the variable content attribute that allows for mapping other `Event`
 6043 models into a CIMI `Event`.

6044 When implementing or using `Event`, Providers and Consumers shall adhere to the syntax and semantics
 6045 of its attributes as described in Table 44. Both Consumer and Provider shall serialize this Resource as
 6046 described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 6047 Resource in both JSON and XML.

6048 **JSON media type:** `application/json`

6049 **JSON serialization:**

```
6050 { "resourceURI": "http://schemas.dmtf.org/cimi/1/Event",
6051   "id": string,
6052   "name": string, ?
6053   "description": string, ?
6054   "created": string, ?
6055   "updated": string, ?
6056   "properties": { string: string, + }, ?
6057   "timestamp": string,
6058   "type": string,
6059   "content": any, ?
6060   "outcome": string, ?
6061   "severity": string, ?
6062   "contact": string, ?
6063   ...
6064 }
```

6065 **XML media type:** `application/xml`

6066 **XML serialization:**

```
6067 <Event xmlns="http://schemas.dmtf.org/cimi/1">
6068   <id> xs:anyURI </id>
6069   <name> xs:string </name> ?
6070   <description> xs:string </description> ?
6071   <created> xs:dateTime </created> ?
6072   <updated> xs:dateTime </updated> ?
6073   <property key="xs:string"> xs:string </property> *
6074   <timestamp> xs:dateTime </timestamp>
6075   <type> xs:string </type>
6076   <content> xs:any* </content> ?
6077   <outcome> xs:string </outcome> ?
6078   <severity> xs:string </severity> ?
6079   <contact> xs:string </contact> ?
6080   <xs:any>*
6081 </Event>
```

6082 Table 45 describes the "type" URIs that are defined or acknowledged by this specification. Additional
 6083 types may be added by a Provider, for example to characterize external events mapped into CIMI
 6084 Events. It is recommended that these URIs be dereferencable such that Consumers can discover a
 6085 more detailed description of the type. Event types defined by this specification share the same base
 6086 URI: http://schemas.dmtf.org/cimi/1/event/. For brevity, if the "Event Type" column in the table only shows
 6087 a relative URI (e.g., state) it shall be appended to the end of this base URI.

6088 **Table 45 – type URIs**

Event Type	Description		
state	Events of this type report state information about CIMI run-time resources such as instances of Machines, Systems, Networks, and Volumes. This information includes reports on any change in the "state" of these Resources. The content element associated with this Event type has the following structure:		
	Data	Type	Description
	resName	<i>string</i>	The name of the Resource about the state of which is reported. Constraints: Provider: support optional; immutable Consumer: support optional; read-only
	resource	<i>ref</i>	The reference to the Resource about the state of which is reported. (Note: This reference may become invalid because the event might outlive the Resource.) Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only
	resType	<i>URI</i>	URI denoting this Resource type (same as the type URI associated with the Resource type for this Resource). Constraints: Provider: support optional; immutable Consumer: support optional; read-only.
	state	<i>string</i>	The state reported for the Resource. Shall be the same as the "state" attribute value (if any) of the run-time Resource at the time the event is generated. Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only
	previous	<i>string</i>	The previous state value, if the event reports a state change. Constraints: Provider: support optional; immutable Consumer: support optional; read-only.

Event Type	Description		
alarm	<p>Events of this type report errors or alarms occurring during management operations of Cloud resources. This information includes failures to provision resources, failures to fulfill requests to the CIMI interface, and any critical situation that needs be addressed in a timely manner. The content element associated with this event type has the following structure:</p>		
	Data	Type	Description
	resName	<i>string</i>	The name of the Resource associated with this alarm, if applicable. Constraints: Provider: support optional; immutable Consumer: support optional; read-only.
	resource	<i>ref</i>	The reference to the Resource associated with this alarm, if applicable. (Note: This reference may become invalid because the event might outlive the Resource.) Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only
	restype	<i>URI</i>	URI denoting this Resource type associated with this alarm, if applicable (same as the type URI associated with the Resource type for this Resource). Constraints: Provider: support optional; immutable Consumer: support optional; read-only
	code	<i>string</i>	An alarm code. Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only
	detail	<i>string</i>	The detailed information associated with the alarm. Constraints: Provider: support optional; immutable Consumer: support optional; read-only

Event Type	Description																		
model	<p>Events of this type report changes in the CIMI resource model, which includes creation, modification, and destruction of Resource instances; and updates to metadata (Resource extensions, capabilities and constraints, etc.).</p> <p>The content element associated with this event type has the following structure:</p> <table border="1"> <thead> <tr> <th>Data</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>resName</td> <td>string</td> <td>The name of the main model Resource affected by the modification. Constraints: Provider: support optional; immutable Consumer: support optional; read-only</td> </tr> <tr> <td>resource</td> <td>ref</td> <td>The reference to the main model Resource affected by the modification. (Note: This reference may become invalid because the event might outlive the Resource.) Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only</td> </tr> <tr> <td>resType</td> <td>URI</td> <td>URI denoting this Resource type (same as the type URI associated with the Resource type for this Resource). Constraints: Provider: support optional; immutable Consumer: support optional; read-only</td> </tr> <tr> <td>change</td> <td>string</td> <td>The kind of modification reported (create/update/delete). Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only</td> </tr> <tr> <td>detail</td> <td>string</td> <td>The detailed information associated with the change, typically the data for an update or creation, as used in a request. Constraints: Provider: support optional; immutable Consumer: support optional; read-only</td> </tr> </tbody> </table>	Data	Type	Description	resName	string	The name of the main model Resource affected by the modification. Constraints: Provider: support optional; immutable Consumer: support optional; read-only	resource	ref	The reference to the main model Resource affected by the modification. (Note: This reference may become invalid because the event might outlive the Resource.) Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only	resType	URI	URI denoting this Resource type (same as the type URI associated with the Resource type for this Resource). Constraints: Provider: support optional; immutable Consumer: support optional; read-only	change	string	The kind of modification reported (create/update/delete). Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only	detail	string	The detailed information associated with the change, typically the data for an update or creation, as used in a request. Constraints: Provider: support optional; immutable Consumer: support optional; read-only
Data	Type	Description																	
resName	string	The name of the main model Resource affected by the modification. Constraints: Provider: support optional; immutable Consumer: support optional; read-only																	
resource	ref	The reference to the main model Resource affected by the modification. (Note: This reference may become invalid because the event might outlive the Resource.) Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only																	
resType	URI	URI denoting this Resource type (same as the type URI associated with the Resource type for this Resource). Constraints: Provider: support optional; immutable Consumer: support optional; read-only																	
change	string	The kind of modification reported (create/update/delete). Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only																	
detail	string	The detailed information associated with the change, typically the data for an update or creation, as used in a request. Constraints: Provider: support optional; immutable Consumer: support optional; read-only																	
access	<p>Events of this type keep track of all requests to access some Resource of a CIMI provider.</p> <p>The content element associated with this event type has the following structure:</p> <table border="1"> <thead> <tr> <th>Data</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>operation</td> <td>string</td> <td>The method or name of the operation intended for this access (for the HTTP protocol, the HTTP method for the request). Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only</td> </tr> <tr> <td>resource</td> <td>ref</td> <td>The reference of the primary Resource supporting the operation (for the HTTP protocol, the Resource URI or the URI associated with the operation). (Note: This reference may become invalid because the event might outlive the Resource.) Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only</td> </tr> <tr> <td>detail</td> <td>string</td> <td>The detailed information associated with the change, typically the data for an update or creation, as used in a request Constraints: Provider: support optional; immutable Consumer: support optional; read-only</td> </tr> <tr> <td>initiator</td> <td>string</td> <td>The details identifying the request initiator, in case that information can be associated with the request. Constraints: Provider: support optional; immutable Consumer: support optional; read-only</td> </tr> </tbody> </table>	Data	Type	Description	operation	string	The method or name of the operation intended for this access (for the HTTP protocol, the HTTP method for the request). Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only	resource	ref	The reference of the primary Resource supporting the operation (for the HTTP protocol, the Resource URI or the URI associated with the operation). (Note: This reference may become invalid because the event might outlive the Resource.) Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only	detail	string	The detailed information associated with the change, typically the data for an update or creation, as used in a request Constraints: Provider: support optional; immutable Consumer: support optional; read-only	initiator	string	The details identifying the request initiator, in case that information can be associated with the request. Constraints: Provider: support optional; immutable Consumer: support optional; read-only			
Data	Type	Description																	
operation	string	The method or name of the operation intended for this access (for the HTTP protocol, the HTTP method for the request). Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only																	
resource	ref	The reference of the primary Resource supporting the operation (for the HTTP protocol, the Resource URI or the URI associated with the operation). (Note: This reference may become invalid because the event might outlive the Resource.) Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only																	
detail	string	The detailed information associated with the change, typically the data for an update or creation, as used in a request Constraints: Provider: support optional; immutable Consumer: support optional; read-only																	
initiator	string	The details identifying the request initiator, in case that information can be associated with the request. Constraints: Provider: support optional; immutable Consumer: support optional; read-only																	
http://schemas.dmtf.org/cloud/audit/1.0/	<p>Events of this type represent events that have audit significance, as defined by CADF (...). This type can be subdivided further by extending the URI path (e.g., http://schemas.dmtf.org/cloud/audit/1.0/event/security, for security audit events).</p> <p>The content element associated with this event type has the same structure as the event serialization defined in CADF (DSP0262)</p>																		

6089 The following pseudo-schemas describe the serialization of the "content" property for various types of
 6090 events:

6091 **"state" event:**

6092 **JSON serialization:**

```
6093 { "id": string,
6094   ...
6095   "type": "http://schemas.dmtf.org/cimi/1/event/state",
6096   "content": {
6097     "resName": string,
6098     "resource" : { "href" : string },
6099     "resType" : string,
6100     "state" : string,
6101     "previous" : string ?
6102   }
6103   ...
6104 }
```

6105 **XML serialization:**

```
6106 <Event xmlns="http://schemas.dmtf.org/cimi/1">
6107   ...
6108   <type> http://schemas.dmtf.org/cimi/1/event/state </type>
6109   <content>
6110     <resName> xs:string </resName>
6111     <resource href="xs:anyURI"/>
6112     <resType> xs:anyURI </resType>
6113     <state> xs:string </state>
6114     <previous> xs:string </previous> ?
6115   </content> ?
6116   ...
6117 </Event>
```

6119 **"alarm" event:**

6120 **JSON serialization:**

```
6121 { "id": string,
6122   ...
6123   "type": "http://schemas.dmtf.org/cimi/1/event/alarm",
6124   "content": {
6125     "resName": string ?
6126     "resource" : { "href" : string }, ?
6127     "resType" : string ?
```

```

6128     "code" : string,
6129     "detail" : string ?
6130   }
6131   ...
6132 }

```

6133 **XML serialization:**

```

6134 <Event xmlns="http://schemas.dmtf.org/cimi/1">
6135   ...
6136   <type> http://schemas.dmtf.org/cimi/1/event/alarm </type>
6137   <content>
6138     <resname> xs:string </resname> ?
6139     <resource href="xs:anyURI"/> ?
6140     <restype> xs:anyURI </restype> ?
6141     <code> xs:string </code>
6142     <detail> xs:string </detail> ?
6143   </content> ?
6144   ...
6145 </Event>

```

6146 **"model" event:**

6147 **JSON serialization:**

```

6148 { "id": string,
6149   ...
6150   "type": "http://schemas.dmtf.org/cimi/1/event/model",
6151   "content": {
6152     "resName": string, ?
6153     "resource" : { "href" : string }, ?
6154     "resType" : string, ?
6155     "change" : string,
6156     "detail" : string ?
6157   }
6158   ...
6159 }

```

6160 **XML serialization:**

```

6161 <Event xmlns="http://schemas.dmtf.org/cimi/1">
6162   ...
6163   <type> http://schemas.dmtf.org/cimi/1/event/model </type>
6164   <content>
6165     <resname> xs:string </resname> ?
6166     <resource href="xs:anyURI"/> ?

```

```

6167 <restype> xs:anyURI </restype> ?
6168 <change> xs:string </change>
6169 <detail> xs:string </detail> ?
6170 </content> ?
6171 ...
6172 </Event>

```

6173 **"access" event:**

6174 **JSON serialization:**

```

6175 { "id": string,
6176   ...
6177   "type": "http://schemas.dmtf.org/cimi/1/event/access",
6178   "content": {
6179     "operation": string,
6180     "resource" : { "href" : string },
6181     "detail" : string, ?
6182     "initiator" : string ?
6183   }
6184   ...
6185 }

```

6186 **XML serialization:**

```

6187 <Event xmlns="http://schemas.dmtf.org/cimi/1">
6188   ...
6189   <type> http://schemas.dmtf.org/cimi/1/event/access </type>
6190   <content>
6191     <operation> xs:string </operation>
6192     <resource href="xs:anyURI"/>
6193     <detail> xs:string </detail> ?
6194     <initiator> xs:string </initiator> ?
6195   </content> ?
6196   ...
6197 </Event>

```

6198 5.17.13.1 Operations

6199 This resource supports the Read, Update, and Delete operations.

6200 6 Security considerations

6201 There are many security mechanisms that can be used in conjunction with this specification. This
6202 specification does not mandate any particular mechanism. Providers shall provide enough information
6203 about their security mechanisms so that the Consumer can implement the necessary algorithms to
6204 successfully communicate with the Provider.

6205 An implementation may set limits on the length of attribute values it accepts. An implementation may set
6206 limits on the size of arrays it accepts. An implementation may set limits on the size of the request body or
6207 the length of request URIs it accepts. These limits may not all be advertised in the ResourceMetadata,
6208 although this specification recommends Providers to do so. A Provider that receives a request that
6209 exceeds any of these limits, shall return a response with an appropriate standard HTTP status code.

6210

ANNEX A (normative)

6211
6212
6213
6214
6215

OVF support in CIMI

6216 This annex defines how elements of an OVF descriptor are mapped to CIMI resources and their
6217 attributes. This definition allows the import of an OVF package to create multiple CIMI resources. This is
6218 done by specifying a reference to an OVF package in the import operation of a `SystemCollection` or
6219 `SystemTemplateCollection` (the Media Type at that URI shall be “application/ovf”). Refer to
6220 [DSP0243](#) for more information about OVF.

6221 Support for OVF import and export is optional for a Provider and it is an implementation choice as to how
6222 many of the attributes in the OVF package are exposed through CIMI resources. A Provider may support
6223 the import of OVF package for only `Systems`, only `SystemTemplates` or both. Support for the actual
6224 import and export of an OVF package is handled by a hypervisor under the management of the CIMI
6225 implementation, and thus the CIMI resources that are created reflect what the hypervisor did upon import
6226 and form a “View” into the results.

6227 The import of an OVF package can be reflected in the creation of `Templates` that can be later used to
6228 create `Systems`, `Machines` and other component `Resources`. The import of an OVF package can also
6229 be used to directly create `Systems`, `Machines`, and other component `Resources`, bypassing the step
6230 of creating `Templates`.

6231 Clause 5.13.4 details how to import an OVF file to create a `SystemTemplate` (and component
6232 `Resources`). The `SystemTemplate` thus created contains a reference to a `MachineTemplate` for
6233 every `VirtualSystem` that is defined in the OVF descriptor `VirtualSystemCollection`. Note
6234 that CIMI currently allows `Systems` of `Systems`, so for each `VirtualSystemCollection`
6235 encountered in a nested set of collections, a separate `SystemTemplate` is created within the parent
6236 `SystemTemplate` with `MachineTemplates` for each of the contained `VirtualSystems` in that
6237 `VirtualSystemCollection`.

6238 The values of the attributes for the `MachineTemplate` are taken from the
6239 `VirtualHardwareSection` of the `VirtualSystem` description (required in OVF). If more than
6240 one `VirtualHardwareSection` is used for a given `VirtualSystem` (allowed in OVF), the result
6241 is implementation dependent, but the implementation might choose a `MachineTemplate` from an
6242 existing (perhaps static) set that best matches a `VirtualHardwareSection`. Items in the
6243 `VirtualHardwareSection` are mapped to CIMI `MachineConfiguration` properties and the
6244 corresponding `MachineConfiguration` Resource is created and linked to from the created
6245 `MachineTemplate` for that `VirtualSystem`.

6246 The CIMI `VolumeTemplates` are created according to the `DiskSection` of an OVF descriptor and
6247 can be shared among more than one `VirtualSystem` (CIM) `MachineTemplates` defined in an
6248 OVF package. In addition, a new CIMI `MachineImage` Resource may be created from the
6249 `DiskSection` if an `ovf:fileRef` for the virtual disk content is specified.

6250 The CIMI `NetworkTemplates` are created according to the `NetworkSection` of an OVF descriptor
6251 along with the `Connection` elements in the `VirtualHardwareSection` elements that refer to
6252 these named networks.

6253 Clause 5.13.2.1 details how to import an OVF file to create a `System` (and component `Resources`). The
6254 `System` thus created contains a reference to a `Machine` for every `VirtualSystem` that is defined in

- 6255 an OVF descriptor `VirtualSystemCollection`. Note that CIMI currently allows `Systems of`
6256 `Systems`, so for each `VirtualSystemCollection` encountered in a nested set of collections, a
6257 separate `System` is created within the parent `System` with `Machines` for each of the contained
6258 `VirtualSystems` in that `VirtualSystemCollection`.
- 6259 The values of the attributes for the `Machine` are taken from the `VirtualHardwareSection` of the
6260 `VirtualSystem` description (required in OVF). If more than one `VirtualHardwareSection` is
6261 used for a given `VirtualSystem` (allowed in OVF), the result is implementation dependent. Items in
6262 the `VirtualHardwareSection` are mapped to CIMI `MachineConfiguration` properties and
6263 the corresponding `MachineConfiguration` Resource is created and linked to from the created
6264 `Machine` for that `VirtualSystem`.
- 6265 The CIMI `Volumes` are created according to the `DiskSection` of an OVF descriptor and can be
6266 shared among more than one `VirtualSystem` (CIMI `Machines`) defined in an OVF package. In
6267 addition, a new CIMI `MachineImage` Resource may be created from the `DiskSection` if an
6268 `ovf:fileRef` attribute for the virtual disk content is specified.
- 6269 The CIMI `Networks` are created according to the `NetworkSection` of an OVF descriptor along with
6270 the `Connection` elements in the `VirtualHardwareSection` that refer to these named networks.
- 6271

6272 **ANNEX B**
6273 **(informative)**

6274
6275
6276 **XML Schema**

6277 The XML Schema for the XML serialization of the CIMI model can be found at:

6278 http://schemas.dmtf.org/cimi/1/dsp8009_1.0.xsd

6279 The schema provided does not intend to reflect every single modeling constraint and requirement
6280 specified in the model. This schema is designed to apply more broadly to any model-related serialized
6281 material found in Consumer requests as well as in Provider responses, and is intended to provide a
6282 preliminary, non-exhaustive syntactic check on these. In particular, future updates of this specification
6283 may intermix new XML elements into the Resources using the current CIMI namespace to Resources.
6284 The schema that is provided is just a starting point for those who would find it useful and it might need to
6285 be modified based on specific application's needs.

**ANNEX C
(informative)**

Change log

6286
6287
6288
6289

6290

Version	Date	Description
1.0.0	2012-08-28	
1.0.1	2012-09-12	Errata
1.1.0	2013-10-22	DMTF Standard
2.0.0a	2014-09-22	Released as work in progress
		-

6291

Bibliography

6292 DMTF Standard: *Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based*
6293 *Protocol* specification V1.0 (DSP0263)

6294 http://dmf.org/sites/default/files/standards/documents/DSP0263_1.0.0.pdf

6295 DMTF Standard: *Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based*
6296 *Protocol* specification V1.1 (DSP0263)

6297 https://members.dmf.org/apps/org/workgroup/cmwg/download.php/73648/DSP0263_1.1.0b_RC2.pdf