



1
2
3
4

Document Identifier: DSP0263

Date: 2015-03-20

Version: 2.0.0c

5
6
7

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

8 **Supersedes: 1.1.0**
9 **Document Type: Specification**
10 **Document Class: Normative**
11 **Document Status: Work in Progress**
12 **Document Language: en-US**

13 Copyright Notice

14 Copyright © 2012-2015 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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29 withdrawn or modified after publication, and shall be indemnified and held harmless by any party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

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

CONTENTS

36	Foreword	7
37	1 Scope	9
38	1.1 Document structure.....	9
39	1.2 Document versioning scheme	9
40	1.3 Typographical conventions	9
41	2 Normative references	10
42	3 Terms and definitions	12
43	4 HTTP-based protocol	15
44	4.1 Introduction	15
45	4.1.1 Protocol evolution and client expectations	15
46	4.1.2 XML namespaces	15
47	4.1.3 URI space	15
48	4.1.4 Media types.....	16
49	4.1.5 Request headers.....	16
50	4.1.6 Request query parameters	16
51	4.1.7 Response headers.....	22
52	4.2 Protocol operations	22
53	4.2.1 Common CRUD operations	23
54	4.2.2 Error handling	30
55	4.3 OVF support.....	30
56	5 Model.....	30
57	5.1 Resource wrappers.....	31
58	5.2 Extensibility	31
59	5.3 Identifiers	32
60	5.4 Attribute constraints	32
61	5.5 Data types and their serialization.....	33
62	5.5.1 boolean	33
63	5.5.2 dateTime	33
64	5.5.3 duration	34
65	5.5.4 integer	34
66	5.5.5 string	34
67	5.5.6 ref.....	34
68	5.5.7 map	35
69	5.5.8 structure	35
70	5.5.9 byte[]	36
71	5.5.10 URI.....	36
72	5.5.11 Array	36
73	5.5.12 Collection	37
74	5.5.13 "Any" type	43
75	5.5.14 valueScope	43
76	5.5.15 Empty attribute values	46
77	5.6 Units.....	46
78	5.7 Resources.....	46
79	5.7.1 Common Resource attributes	46
80	5.8 Operations	48
81	5.9 Alternative model formats	49
82	5.10 Relationships between Resources	49
83	5.10.1 Referencing across Resources.....	49
84	5.10.2 Composition Relationship between Resources	49
85	5.11 Resource metadata.....	49
86	5.11.1 Capabilities	53
87	5.11.2 ResourceMetadataCollection Resource	57

88	5.12	Cloud Entry Point	58
89	5.12.1	Operations	64
90	5.13	System Resources and relationships	65
91	5.13.1	System	65
92	5.13.2	SystemCollection Resource	71
93	5.13.3	SystemService Resource	73
94	5.13.4	SystemTemplate Resource	77
95	5.13.5	SystemTemplateCollection Resource	83
96	5.13.6	Service-specific Descriptor attributes	85
97	5.14	Machine Resources and relationships	86
98	5.14.1	Machine	87
99	5.14.2	MachineCollection Resource	101
100	5.14.3	MachineTemplate	103
101	5.14.4	MachineTemplateCollection Resource	108
102	5.14.5	MachineConfiguration Resource	109
103	5.14.6	MachineConfigurationCollection Resource	111
104	5.14.7	MachineImage Resource	112
105	5.14.8	MachineImageCollection Resource	115
106	5.14.9	Credential Resource	116
107	5.14.10	CredentialCollection Resource	118
108	5.14.11	CredentialTemplate Resource	118
109	5.14.12	CredentialTemplateCollection Resource	120
110	5.15	Volume Resources and relationships	120
111	5.15.1	Volume	121
112	5.15.2	VolumeCollection Resource	125
113	5.15.3	VolumeTemplate Resource	126
114	5.15.4	VolumeTemplateCollection Resource	128
115	5.15.5	VolumeConfiguration Resource	129
116	5.15.6	VolumeConfigurationCollection Resource	130
117	5.15.7	VolumeImage Resource	131
118	5.15.8	VolumeImageCollection Resource	133
119	5.16	Network Resources and relationships	134
120	5.16.1	Network	135
121	5.16.2	NetworkCollection Resource	139
122	5.16.3	NetworkTemplate Resource	140
123	5.16.4	NetworkTemplateCollection Resource	144
124	5.16.5	Segments	145
125	5.16.6	ProtocolSegmentCollection Resource	150
126	5.16.7	ProtocolSegmentTemplate Resource	151
127	5.16.8	ProtocolSegmentTemplateCollection Resource	155
128	5.16.9	Endpoints	156
129	5.16.10	ProtocolEndpointCollection Resource	161
130	5.16.11	ProtocolEndpointTemplate Resource	162
131	5.16.12	ProtocolEndpointTemplateCollection Resource	165
132	5.16.13	Interfaces	166
133	5.16.14	NetworkInterfaceCollection Resource	170
134	5.16.15	NetworkInterfaceTemplate Resource	171
135	5.16.16	NetworkInterfaceTemplateCollection Resource	173
136	5.16.17	Services	174
137	5.16.18	NetworkServiceCollection Resource	179
138	5.16.19	NetworkServiceTemplate Resource	180
139	5.16.20	NetworkServiceTemplateCollection Resource	183
140	5.16.21	Policies	184
141	5.17	Monitoring Resources and relationships	184
142	5.17.1	Job Resource	184
143	5.17.2	JobCollection Resource	189

144	5.17.3 Meter Resource	189
145	5.17.4 MeterCollection Resource	195
146	5.17.5 MeterTemplate Resource	196
147	5.17.6 MeterTemplateCollection Resource	198
148	5.17.7 MeterConfiguration Resource	198
149	5.17.8 MeterConfigurationCollection Resource	201
150	5.17.9 EventLog Resource	202
151	5.17.10 EventLogCollection Resource	205
152	5.17.11 EventLogTemplate Resource	206
153	5.17.12 EventLogTemplateCollection Resource	207
154	5.17.13 Event Resource	208
155	6 Security considerations	216
156	ANNEX A (normative) OVF support in CIMI	218
157	ANNEX B (informative) XML Schema	220
158	ANNEX C (informative) Change log	221
159	Bibliography	222
160		

161 **Figures**

162	Figure 1 - Cloud Entry Point.....	59
163	Figure 2 - System Resources.....	65
164	Figure 3 - Machine Resources	87
165	Figure 4 - Volume Resources	121
166	Figure 5 - Network Resources	135
167	Figure 6 - Monitoring Resources.....	184

168 **Tables**

169	Table 1 – XML namespaces	15
170	Table 2 – Named structure.....	35
171	Table 3 – Converting a relative URI to an absolute URI	36
172	Table 4 – Numerical equivalents for attributes.....	46
173	Table 5 – Common attributes.....	46
174	Table 6 – ResourceMetadata attributes	50
175	Table 7 – Capability URIs	54
176	Table 8 – CloudEntryPoint attributes	59
177	Table 9 – System attributes	66
178	Table 10 – RecoverableMachine accessory attributes	75
179	Table 11 – SystemTemplate attributes	77
180	Table 12 – Machine attributes.....	87
181	Table 13 – Disk attributes	91
182	Table 14 – locatedVolume accessory attributes.....	93
183	Table 15 – MachineTemplate attributes.....	103
184	Table 16 – MachineConfiguration attributes	109
185	Table 17 – MachineImage attributes.....	113
186	Table 18 – Credential attributes	116
187	Table 19 – UserName/Password attributes	116
188	Table 20 – Public key attributes	116

189 Table 21 – CredentialTemplate attributes 119

190 Table 22 – Volume attributes 121

191 Table 23 – VolumeTemplate attributes 126

192 Table 24 – VolumeConfiguration attributes..... 129

193 Table 25 – VolumeImage attributes 131

194 Table 26 – Network attributes 135

195 Table 27 – NetworkTemplate attributes 141

196 Table 28 – ProtocolSegment attributes..... 145

197 Table 29 - IPv6 ProtocolSegment parameters..... 147

198 Table 30 – IPv4 ProtocolSegment parameters 148

199 Table 31 – Ethernet ProtocolSegment parameters 148

200 Table 32 – ProtocolSegmentTemplate attributes 152

201 Table 33 – ProtocolEndpoint attributes..... 156

202 Table 34 - IPv6 ProtocolEndpoint parameters 158

203 Table 35 – IPv4 ProtocolEndpoint parameters 159

204 Table 36 – Ethernet ProtocolEndpoint parameters..... 159

205 Table 37 – ProtocolEndpointTemplate attributes..... 162

206 Table 38 – NetworkInterface attributes 166

207 Table 39 – NetworkInterfaceTemplate attributes 171

208 Table 40 – NetworkService attributes 175

209 Table 41 – NetworkServiceTemplate attributes 180

210 Table 42 – Job attributes..... 185

211 Table 43 – Meter attributes 189

212 Table 44 – Sample attributes 192

213 Table 45 – MeterTemplate attributes 196

214 Table 46 – MeterConfiguration attributes..... 199

215 Table 47 – aspect URIs..... 201

216 Table 48 – EventLog attributes 202

217 Table 49 – EventLogTemplate attributes 206

218 Table 50 – Event attributes 208

219 Table 51 – type URIs 211

220

221

222

Foreword

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

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

228 Acknowledgments

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

230 Editors (present and past):

- 231 • Jacques Durand – Fujitsu
- 232 • Marios Andreou – Red Hat (previous)
- 233 • Doug Davis – IBM (previous)
- 234 • Gilbert Pilz – Oracle (previous)

235 Contributors:

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

- 269 • David Lutterkort – Red Hat
- 270 • Fred Maciel – Hitachi, Ltd.
- 271 • Andreas Maier – IBM
- 272 • Ashok Malhotra – Oracle
- 273 • Arturo Martin de Nicolas - Ericsson
- 274 • Jeff Mischkinsky – Oracle
- 275 • Jesus Molina – Fujitsu
- 276 • Efraim Moscovich – CA Technologies
- 277 • Bryan Murray – Hewlett-Packard Company
- 278 • Steven Neely – Cisco
- 279 • Ryuichi Ogawa – NEC Corporation
- 280 • John Parchem– Microsoft Corporation
- 281 • Shishir Pardikar – Citrix Systems Inc.
- 282 • Miguel Peñalvo – Telefónica
- 283 • Gilbert Pilz – Oracle
- 284 • Alvaro Polo – Telefónica
- 285 • Enrico Ronco – Telecom Italia
- 286 • Federico Rossini – Telecom Italia
- 287 • Matthew Rutkowski – IBM
- 288 • Tom Rutt – Fujitsu
- 289 • Hemal Shah – Broadcom
- 290 • Nihar Shah – Microsoft Corporation
- 291 • Alan Sill – Texas Tech University
- 292 • Zhexuan Song – Huawei
- 293 • Marvin Waschke – CA Technologies
- 294 • Eric Wells – Hitachi, Ltd.
- 295 • Jeff Wheeler – Huawei
- 296 • Maarten Wiggers – Fujitsu
- 297 • Daniel Wilson – Ericsson AB
- 298 • Steve Winkler – SAP AG
- 299 • Jack Yu – Oracle
- 300 • Aaron Zhang – Huawei
- 301 • HengLiang Zhang – Huawei
- 302

303 Cloud Infrastructure Management Interface (CIMI) Model and 304 RESTful HTTP-based Protocol

305 1 Scope

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

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

318 1.1 Document structure

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

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

322 1.2 Document versioning scheme

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

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

327 1.3 Typographical conventions

328 This specification uses the following conventions:

329 In the narrative text of the specification:

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

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

344 Inside tables describing the Resource data model:

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

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

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

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

373 2 Normative references

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

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

- 380 DMTF DSP0243, *Open Virtualization Format Specification 1.1*,
381 http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.1.pdf
- 382 DMTF DSP0262, *Cloud Audit Data Federation (CADF) -Data Format and Interface Definitions*
383 *Specification version 1.0.0*,
384 http://dmtf.org/sites/default/files/standards/documents/DSP0262_1.0.0.pdf
- 385 DMTF DSP1001, *Management Profile Specification Usage Guide 1.1*,
386 http://www.dmtf.org/standards/published_documents/DSP1001_1.1.pdf
- 387 DMTF DSP4004, *DMTF Release Process 2.4*,
388 http://www.dmtf.org/sites/default/files/standards/documents/DSP4004_2.4.pdf
- 389 IANA HTTP Header Registry,
390 <http://www.iana.org/assignments/message-headers/perm-headers.html>
- 391 IEC 80000-13:2008, International Organization for Standardization, Geneva, Switzerland, *Quantities and*
392 *units – Part 13: Information science and technology*, April 2008,
393 http://www.iso.org/iso/catalogue_detail?csnumber=31898
- 394 IEEE 802.3-2012, IEEE Standards Association. *IEEE Standard for Ethernet*, December 2012,
395 <http://standards.ieee.org/findstds/standard/802.3-2012.html>
- 396 IETF RFC791, Postel, J., *Internet Protocol*, September 1981,
397 <http://www.ietf.org/rfc/rfc791.txt>
- 398 IETF RFC2460, Deering, S. and R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, December
399 1998,
400 <http://www.ietf.org/rfc/rfc2460.txt>
- 401 IETF RFC2616, R. Fielding et al, *Hypertext Transfer Protocol -- HTTP/1.1*,
402 <http://www.ietf.org/rfc/rfc2616.txt>
- 403 IETF RFC3986, T. Berners-Lee et al, *Uniform Resource Identifiers (URI): Generic Syntax*, August 1998,
404 <http://www.ietf.org/rfc/rfc3986.txt>
- 405 IETF RFC4291, Deering, S. and R. Hinden, *IP Version 6 Addressing Architecture*, February 2006,
406 <http://www.ietf.org/rfc/rfc4291.txt>
- 407 IETF RFC4627, D. Crockford, *The application/json Media Type for JavaScript Object Notation (JSON)*,
408 July 2006,
409 <http://www.ietf.org/rfc/rfc4627.txt>
- 410 IETF RFC5246, T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*,
411 <http://www.ietf.org/rfc/rfc5246.txt>
- 412 ISO 8601:2004, International Organization for Standardization, Geneva, Switzerland, *Data elements and*
413 *interchange formats -- Information interchange - - Representation of dates and times*, March 2008,
414 http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40874
- 415 ISO/IEC 14977:1996, Roger S. Scowen, *Extended BNF — A generic base standard*. Software
416 Engineering Standards Symposium 1993.
417 http://www.iso.org/iso/catalogue_detail?csnumber=26153

- 418 ISO/IEC Directives, Part 2, *Rules for the structure and drafting of International Standards*,
419 <http://isotc.iso.org/livelink/livelink.exe?func=ll&objId=4230456&objAction=browse&sort=subtype>
- 420 NIST Special Publication 800-145, Peter Mell and Timothy Grance, *The NIST Definition of Cloud*
421 *Computing*, Sept. 2011,
422 <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- 423 NIST Special Publication 500-292, Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee
424 Badger and Dawn Leaf, *NIST Cloud Computing Reference Architecture*, Sept. 2011,
425 [http://collaborate.nist.gov/twiki-cloud-](http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/ReferenceArchitectureTaxonomy/NIST_SP_500-292_-_090611.pdf)
426 [computing/pub/CloudComputing/ReferenceArchitectureTaxonomy/NIST_SP_500-292_-_090611.pdf](http://collaborate.nist.gov/twiki-cloud-computing/pub/CloudComputing/ReferenceArchitectureTaxonomy/NIST_SP_500-292_-_090611.pdf)
- 427 Representational State Transfer, Roy Fielding, Doctoral dissertation, University of California, *Architectural*
428 *Styles and the Design of Network-based Software Architectures (Chapter 5)*, 2000,
429 http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm
- 430 Unicode Standard, Unicode Consortium, *The Unicode Standard*, Version 2.0, Addison-Wesley, 1996.
- 431 XMLSchema - Part 1, World Wide Web Consortium (W3C) Recommendation, H. Thompson, et al.,
432 Editors, *XML Schema Part 1: Structures Second Edition*, 28 October 2004,
433 <http://www.w3.org/TR/xmlschema-1/>
- 434 XMLSchema - Part 2, World Wide Web Consortium (W3C) Recommendation, P. Biron, A. Malhotra,
435 Editors, *XML Schema Part 2: Datatypes (Second Edition)*, 28 October 2004,
436 <http://www.w3.org/TR/xmlschema-2/>

437 **3 Terms and definitions**

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

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

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

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

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

453 **3.1** 454 **authentication**

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

458 **3.2**

459 **authorization**

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

462 **3.3**

463 **cloud**

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

466 **3.4**

467 **Cloud Service Consumer**

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

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

479 **3.5**

480 **Cloud Service Provider**

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

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

495 **3.6**

496 **Collection**

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

499 **3.7**

500 **Configuration**

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

503 **3.8**504 **Endpoint**

505 An element within a Network Segment from which communication can originate or to which
506 communication can be sent. Endpoints have a unique, protocol specific, address within a Segment by
507 which they are distinguished.

508 **3.9**509 **Infrastructure as a Service (IaaS)**

510 A cloud computing service model defined in section 2 of the NIST Definition of Cloud Computing [[SP800-](#)
511 [145](#)].

512 **3.10**513 **Interface**

514 An abstract element of virtual hardware that enables connection to a Network via Endpoints.

515 **3.11**516 **message confidentiality**

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

518 **3.12**519 **message integrity**

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

522 **3.13**523 **Network**

524 A construct that supports communications between elements within a Cloud using one or more protocol
525 specific Segments that support addressable Endpoints.

526 **3.14**527 **Resource**

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

531 **3.15**532 **Segment**

533 A component of a Network that supports communication between Endpoints using a single protocol. Also
534 referred to as a Protocol Segment to emphasize that Segments are always bound to a single
535 communication protocol.

536 **3.16**537 **Template**

538 A component Synonym for "CIMI template". A Resource that represents the set of metadata and
539 instructions used to instantiate some other Resource (e.g., a `MachineTemplate` is used to create
540 `Machines`).

541 **4 HTTP-based protocol**

542 **4.1 Introduction**

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

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

555 **4.1.1 Protocol evolution and client expectations**

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

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

- 563 2. Clients shall not assume anything about the operations supported by a server. They are expected
 564 to discover operations that are supported (and permissible) by navigating to Resources from the
 565 cloud entry point. The serializations of Resources encountered indicate which operations are
 566 supported by the server.

567 **4.1.2 XML namespaces**

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

570 **Table 1 – XML namespaces**

Prefix	XML Namespaces	Specification
cimi	http://schemas.dmtf.org/cimi/2	This specification
xs	http://www.w3.org/2001/XMLSchema	XML Schema Part2

571 **4.1.3 URI space**

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

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

```

579 http://example.com/machines/12345
580 http://example.com/machines?id=12345
581 http://example.com/12345
582 http://example.com/Cloud/resource?id=12345

```

583 4.1.4 Media types

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

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

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

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

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

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

602 4.1.5 Request headers

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

606 4.1.6 Request query parameters

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

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

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

620 4.1.6.1 Filtering Collections

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

625 `?$filter=expression`

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

```
629 Filter      ::= AndExpr ( 'or' Filter )* ;
630 AndExpr     ::= Comp ( 'and' AndExpr )*
631 Comp        ::= Attribute Op Value
632              | Value Op Attribute
633              | PropExpr
634              | '(' Filter ')'
635 Op          ::= '<' | '<=' | '=' | '>=' | '>' | '!='
636 Attribute   ::= ? resource attribute name ?
637 Value       ::= IntValue | DateValue | StringValue | BoolValue
638 IntValue    ::= /[0-9]+/
639 DateValue   ::= ? as defined by XML Schema ?
640 StringValue ::= "... " | '...'
641 BoolValue   ::= 'true' | 'false'
642 PropExpr    ::= 'property[' StringValue ']' Op StringValue
```

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

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

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

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

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

656 Examples:

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

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

659 `/machines`

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

661 `/machines/123`

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

663 `/machines/123/disks`

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

665 `/machines/123/volumes`

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

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

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

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

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

674 4.1.6.2 Subsetting Collections

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

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

682 `?$first=number`

683 `?$last=number`

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

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

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

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

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

703 4.1.6.3 Subsetting Resources

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

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

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

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

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

720 is equivalent to:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

750 4.1.6.4 Expanding references

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

756 The serialization shall be performed as follows:

757 JSON serialization:

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

759 shall be expanded to be:

```
760 "name": {  
761   "href": string,  
762   ... attributes of referenced resource...  
763 }
```

764 XML serialization:

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

766 shall be expanded to be:

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

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

772 The format of a `$expand` query parameter shall be:

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

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

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

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

783 `?$expand=volumes`

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

788 4.1.6.5 Specifying the Resource format

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

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

795 `?$format=encoding`

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

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

802 4.1.6.6 Sorting Collections

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

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

808
 809 The `$orderby` expression may include multiple, comma-separated attribute names. Each attribute
 810 name may be optionally followed immediately by a colon and "asc" to denote ascending order (default),
 811 or "desc" to denote descending order for that attribute. If neither `asc` nor `desc` is specified, the order
 812 shall be "ascending".

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

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

816 The following rules apply to the ascending sort order:

- 817 • boolean – 'false' shall come before 'true'.
- 818 • dateTime – An earlier datetime shall come before a later datetime.
- 819 • duration – A shorter duration shall come before a longer duration.

- 820 • integer – Smaller integers shall come before larger integers. Negative integers shall come
821 before positive integers.
- 822 • string – Ordering is based on a binary comparison of the transformed strings according to the
823 rules of the Normalization Form KD of the Unicode standard as defined in [Unicode Standard](#)
824 [Annex \(UAX\), annex #15](#) .

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

826 **Examples:**

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

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

830

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

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

834

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

837 **4.1.7 Response headers**

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

841 **4.1.7.1 Job header**

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

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

845 **4.1.7.2 ETag support**

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

849 **4.2 Protocol operations**

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

857 If appropriate, some of the Resource representations include an "operations" attribute. Providers shall
858 only include the "operations" attribute if the specified operations are accessible to the current client for
859 that particular Resource. This situation means that based on many factors (e.g., authorization rights of the

860 clients, current state of the Resource, etc.), a different set of "operations" shall be returned on each
861 serialization of the Resource.

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

869 The operations attribute shall be serialized as follows:

870 **JSON serialization:**

```
871 { "operations": [
872   { "rel": string, "href": string, ("available": boolean)? }, +
873 ]
874 }
```

875 **XML serialization:**

```
876 <Resource xmlns="http://schemas.dmtf.org/cimi/2">
877   <operation rel="xs:anyURI" href="xs:anyURI" (available="xs:boolean"? /> *
878 </Resource>
```

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

880 **JSON serialization:**

```
881 { "operations": [
882   { "rel": "edit", "href": "<editURI>" }
883 ]
884 }
```

885 **XML serialization:**

```
886 <Resource xmlns="http://schemas.dmtf.org/cimi/2">
887   <operation rel="edit" href="<editURI>"/>
888 </Resource>
```

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

891 4.2.1 Common CRUD operations

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

894 4.2.1.1 Creating a new Resource

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

899 The HTTP POST request shall include:

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

903 For example, the request can be:

```
904 POST <addURI> HTTP/1.1
905 Host: <hostname>
906 Accept: application/(json|xml)
907 Content-Type: application/(json|xml)
908 Content-Length: <length>
909
910 <serialization of request to create a new resource>
```

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

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

```
919 <MachineCreate xmlns="http://schemas.dmtf.org/cimi/2">
920   <name> xs:string </name> ?
921   <description> xs:string </description> ?
922   <property key="xs:string"> xs:string </property> *
923   <machineTemplate href="xs:anyURI"? >
924     ... template attributes ... ?
925   </machineTemplate>
926 </MachineCreate>
```

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

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

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

```
932 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ResourceCreate",
933   "name": "myResourceName", ?
934   "description": "My resource description", ?
935   "properties": { "proplname" : "proplvalue" , + }, ?
936   "resourceTemplate": {
937     <here the template is passed by value>
938   }
939 }
```

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

```

941 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ResourceCreate ",
942   "name": "myResourceName", ?
943   "description": "My resource description", ?
944   "properties": { "prop1name" : "prop1value" , + }, ?
945   "resourceTemplate": { "href": string ,
946     <here some template attribute/value pairs may be added to override values in the
947     referenced template>
948   }
949 }
```

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

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

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

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

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

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

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

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

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

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

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

978 `"attribute": null`

979 for the attribute in the JSON serialization, or

980 `<attribute/>`

981 in the XML serialization for that attribute.

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

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

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

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

- 989 • HTTP Content-Type header
- 990 • HTTP Content-Length header

991 For example, the response can be:

```
992 HTTP/1.1 201 Created
993 Location: <location>
994 Content-Type: application/(json|xml)
995 Content-Length: <length>
996
997 <serialization of new resource>
```

998 4.2.1.2 Retrieving a representation of a Resource

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

1000 For example, the request can be:

```
1001 GET <ResourceURI> HTTP/1.1
1002 Host: <hostname>
1003 Accept: application/(json|xml)
```

1004 If the response has a 200 status code, the response shall include:

- 1005 • HTTP Content-Type header
- 1006 • HTTP Content-Length header

1007 For example, the response can be:

```
1008 HTTP/1.1 200 OK
1009 Content-Type: application/(json|xml)
1010 Content-Length: <length>
1011
1012 <serialization of resource>
```

1013 4.2.1.3 Updating a Resource

1014 To update a Resource's state, an HTTP PUT request containing the complete, updated representation is
 1015 sent to a designated `editURI` for that Resource type. Consumers shall include all non-empty attributes
 1016 of the Resource in the PUT request - including ones that it might not support or understand that were
 1017 returned in a GET response. This is to ensure that a client does not inadvertently modify (erase) data in a
 1018 Resource by excluding it from the full representation of the Resource.

1019 In many cases, this `editURI` is the same as the URI of Resource itself. Retrieving the Resource
 1020 representation shall include an "edit" operation, which contains the `editURI` that is to be used, if the
 1021 requester is allowed to modify the Resource.

1022 While processing a PUT request, if the server detects that an attempt is being made to update a
 1023 read-only, or immutable, attribute, it shall silently ignore that attribute update request and shall not
 1024 generate an error. This rule applies to Resource partial updates as well.

1025 Because of potential conflicts that might occur due to multiple concurrent updates, Consumers should use
 1026 the partial update mechanism, defined in 4.2.1.3.1, to reduce the chances of mistakenly updating
 1027 attributes with out-of-date data.

1028 The HTTP PUT request shall include:

- 1029 • CIMI serialization of the updated Resource in the HTTP Body
- 1030 • HTTP Content-Type header
- 1031 • HTTP Content-Length header

1032 For example, the request can be:

```
1033 PUT <editURI> HTTP/1.1
1034 Host: <hostname>
1035 Accept: application/(json+xml)
1036 Content-Type: application/(json+xml)
1037 Content-Length: <length>
1038
1039 <serialization of request to update a resource>
```

1040 If the response includes a serialization of the updated Resource and has a status code of 200, this
 1041 response shall include:

- 1042 • HTTP Content-Type header
- 1043 • HTTP Content-Length header

1044 For example, the response can be:

```
1045 HTTP/1.1 200 OK
1046 Content-Type: application/(json+xml)
1047 Content-Length: <length>
1048
1049 <serialization of updated resource>
```

1050 4.2.1.3.1 Partial updates to a Resource

1051 For clarity, this clause explains how to use the `$select` query parameter (see clause 4.1.6.3) to subset
 1052 a Resource for the purposes of only operating on a selected set of top-level attributes.

1053 To update only certain top-level attributes of a Resource, a Consumer may include only the altered
 1054 attributes in the representation of the Resource within the HTTP request body. If this request is made, the
 1055 URI to the Resource shall include the attributes to be modified as a comma-separated list of query
 1056 parameters; in other words, the URI shall be of the form:

```
1057 http://example.com/resource?$select=attribute1,attribute2,...
```

1058 Only the attributes listed in the URI's query parameters shall be modified; attributes not listed in the URI
 1059 shall not be directly modified by the request. Note that this circumstance does not preclude the
 1060 modification of one attribute causing side-effects that result in the modification of an attribute not listed in
 1061 the query parameters.

1062 Any attribute listed in the URI but not included within the HTTP request body shall be reset to a Resource
 1063 specific value (e.g., removed).

1064 From an HTTP perspective, the updated subsetted Resource is a distinct one. The semantics of a normal
 1065 HTTP PUT are adhered to; it is a complete replacement update of the specified Resource. From the
 1066 Consumer's perspective, the partial update is interpreted and executed by the Cloud Service Provider,
 1067 and some part of the Resource is changed.

1068 Adhering to the generic PUT semantics defined previously, any attribute of the original (full) Resource
 1069 included within the HTTP request body shall result in an error being generated if that attribute is not listed
 1070 in the `$select` query parameter - see clause 5.4. Note that this is due to these attributes being
 1071 unknown to this subsetted Resource.

1072 The following sample request updates just the name and description attributes of a Machine:

```
1073 PUT /machines/myMachine?$select=name,description HTTP/1.1
1074 Host: <hostname>
1075 Accept: application/xml
1076 Content-Type: application/xml
1077 Content-Length: <length>
1078 <Machine>
1079   <name>My New Machine</name>
1080 </Machine>
```

1081 The name attribute is set to "My New Machine" and the description attribute is erased.

1082 4.2.1.4 Deleting a Resource

1083 To delete a Resource, an HTTP DELETE request is sent to a designated `deleteURI` for that Resource
 1084 type. In many cases, this `deleteURI` is the same as the URI of Resource itself. Retrieving the
 1085 Resource representation shall include a "delete" operation, which contains the `deleteURI` that is to be
 1086 used, if the requester is allowed to delete the Resource.

1087 For example, the request can be:

```
1088 DELETE <deleteURI> HTTP/1.1
1089 Host: <hostname>
```

1090 If the Resource has a `State` attribute, its value shall be "DELETING", while the Provider is processing
 1091 this operation.

1092 For example, the response can be:

```
1093 HTTP/1.1 200 OK
```

1094 4.2.1.5 Other operations

1095 While some modifications to the Resources in the model can be done by the way of a simple update
 1096 (PUT) operation to the Resource's `editURI`, sometimes a more complex set of actions needs to be
 1097 taken. In these cases, the operations shall be modeled as HTTP POSTs to the operation specific URI of
 1098 the Resource.

1099 For each of the Resources that define additional operations, a description of the HTTP request and
 1100 response bodies is provided. However, the general HTTP interaction are as described below.

1101 The request shall be of the following form:

```
1102 POST <operationURI> HTTP/1.1
1103 Host: <hostname>
1104 Accept: application/(json+xml)
1105 Content-Type: application/(json+xml)
1106 Content-Length: <length>
1107
1108 <serialization of request to perform some action>
```

1109 The form of the response varies depending on the operation and is defined by the operation itself.

1110 Note that the definition of the Create operation (see clause 4.2.1.1) follows this same pattern. It is just
 1111 called out for ease of reference.

1112 4.2.1.6 Synchronous operations

1113 If a Provider supports the `Job` Resource, each incoming PUT, DELETE, POST request shall result in a
 1114 Job Resource being created and an absolute URI reference to that Job Resource shall be returned back
 1115 to the client by the way of the CIMI-Job-URI HTTP Header in the HTTP response message:

```
1116 CIMI-Job-URI: <uri-to-Job>
```

1117 In this case, the requested operation shall be complete and the Job URI shall point to a completed Job. If
 1118 the Job is not complete, the server shall return a 202 and follow the instructions for Asynchronous
 1119 operations.

1120 4.2.1.7 Asynchronous operations

1121 In some cases, an operation requested by the client may take an undetermined amount of time to be
 1122 completed. For example, creating a new `Machine` or starting an existing `Machine` may take a relatively
 1123 long time to be completed. In these cases, it is not practical to complete these operations within a
 1124 reasonable HTTP request timeout interval, so the Provider shall return an HTTP "202 Accepted" response
 1125 code.

1126 As with synchronous operations, if a Provider supports the `Job` Resource, it shall create a `Job` Resource
 1127 for the incoming request and return a reference to that `Job` Resource back to the client by the way of the
 1128 CIMI-Job-URI HTTP Header in the HTTP response message. Additionally, in the case of a "202
 1129 Accepted" response code, the Provider may also return any of the following in the HTTP response body:

- 1130 • A representation of the `Job` Resource, if one was created.
- 1131 • A partial representation of the response message as if the operation were a synchronous
 1132 operation. For example, when creating a new `Machine`, the response message may include a
 1133 partial representation of the new `Machine` in the response message. The list of attributes of the
 1134 Resource that is returned is implementation specific and based upon how much information is
 1135 available at the time the response message is generated, but it shall be consistent with the
 1136 definition of the full Resource representation. In the case of a create operation, the Provider may
 1137 also include an HTTP Location header referencing the "to be created" Resource, if it is known.
- 1138 • An empty response body.

1139 Note that the decision as to whether any particular operation is synchronous or asynchronous is at the
1140 server's discretion.

1141 4.2.2 Error handling

1142 In cases where an error occurs during the processing of a request, the Provider shall include a
1143 representation of a `Job` Resource describing the status of the failed operation. This representation of a
1144 `Job` shall be included even in cases where the Provider does not expose `Job` Resources. This is to
1145 ensure that Consumers are provided with sufficient information, in a consistent manner, as to the reason
1146 for the failure regardless of whether the Provider exposes `Jobs`. A transient `Job` Resource may be
1147 created by the Provider just for error reporting. In case a `Job` Resource is not intended to be used for
1148 more than error reporting, the returned "id" attribute shall be an empty path (i.e., "") and the
1149 `nestedJobs` array shall be expanded (see 4.1.6.4) to inline the representation of the pseudo
1150 subordinate `Jobs`.

1151 4.3 OVF support

1152 The *Open Virtualization Format (OVF) Specification* ([DSP0243](#)) describes an open, secure, portable,
1153 efficient, and extensible format for the packaging and distribution of software to be run in virtual
1154 machines. OVF support in CIMI allows an OVF package to be used to create CIMI management
1155 resources by importing the package. Additionally, CIMI management resources can be exported into an
1156 OVF package. The actual support for the OVF package is typically provided by a hypervisor that is
1157 managed by the CIMI provider. The import of an OVF package exposes CIMI specific constructs and
1158 parameters as a result of the import without altering the original OVF package. Thus the CIMI resources
1159 that are created as a result of the import form a "View" of what the hypervisor did; however, other (non-
1160 CIMI mapped) information from the OVF package may have been used by the hypervisor in its import.
1161 This other information is implementation dependent and is not further touched upon by this standard.

1162 An OVF package can support single virtual machines (VMs) corresponding to a single CIMI `Machine` or
1163 `MachineTemplate` (see clause 5.14.1) or may also support a complex hierarchy of VMs and their
1164 related Resources corresponding to a CIMI `System` or `SystemTemplate` (see clause 5.13.1) and
1165 related CIMI management resources.

1166 OVF support is covered in more detail in 0.

1167 5 Model

1168 This model assumes that a business relationship has already been established between the Consumer
1169 and the Provider. This relationship may include financial terms, creating separately administered clouds
1170 that the consuming organization is paying for, and the establishment of authentication credentials to
1171 access the administrative entry point for each cloud. The scope of this model is one separately
1172 administered cloud.

1173 The CIMI model is described here by using a tabular representation. Each table is modeling a significant
1174 cloud resource for which independent access and manipulation is expected. Relationships between
1175 resources use a referential mechanism based on unique identifiers that is expected to be already
1176 supported by the implementation environment and protocol (e.g., URIs for HTTP).

1177 The model is self-describing and allows for querying its own metadata, e.g., to discover which extensions
1178 have been implemented. The model is also extensible in different ways (see clause 5.1).

1179 Along with this model, a serialization of its entities is defined (both in XML and JSON).

1180 An alternative UML diagram representation is provided for each major group of resources.

1181 5.1 Resource wrappers

1182 The serialization of Resource instances in the model follow these conventions. Consider the serialization
1183 of a Resource named "MyResource":

1184 JSON serialization:

1185 The Resource is serialized as an object wrapping all its attributes, but without a wrapper name. The
1186 Resource includes a `resourceURI` with a URI for the type of Resource being serialized. For example:

```
1187 { "resourceURI": "http://example.com/MyResource",
1188   "attribute": "value"
1189 }
```

1190 XML serialization:

1191 The Resource is serialized as an element with name equal to the Resource name; for example:

```
1192 <MyResource xmlns="http://example.com">
1193   <attribute> value </attribute>
1194 </MyResource>
```

1195 5.2 Extensibility

1196 There are two types of extensibility mechanisms defined by the CIMI model; one is intended for use by
1197 Consumers whilst the other is to be used by Providers.

1198 The first allows for a CIMI Consumer to add additional data to a Resource. Each Resource in the CIMI
1199 model has an attribute called "properties". Consumers, when creating or updating a Resource, may
1200 store any name/value pair in the `properties` attribute. CIMI Providers shall store and return these
1201 values to the Consumer. There is no obligation for the Provider to understand or take any action based on
1202 these values; they are there for the Consumer's convenience. Providers shall not add elements to this
1203 `properties` attribute.

1204 The second type of extensibility mechanism allows for Provider defined extensions and this specification
1205 includes the `ResourceMetadata` Resource for this purpose. `ResourceMetadata` may be used to

- 1206 • express constraints on the existing CIMI defined Resource attributes (e.g., express a maximum
1207 for the 'cpu' attribute of the `MachineConfiguration` Resource)
- 1208 • introduce new attributes for CIMI defined Resources together with any constraints governing
1209 these (e.g., a new 'location' attribute for the `Volume` Resource that takes values from a defined
1210 set of strings)
- 1211 • introduce new operations for any of the CIMI defined Resources (e.g., define a new 'compress'
1212 operation for the `Volume` Resource)
- 1213 • express any Provider specific capabilities or features (e.g., the length of time that a `Job`
1214 Resource is retained after `Job` completion and before this is deleted)

1215 It is recommended that Providers use the `ResourceMetadata` Resource to advertise these attributes,
1216 operations, and capabilities along with any constraints that might need to be understood by Consumers.
1217 The `ResourceMetadata` Resource is defined in clause 5.8.

1218 If a Provider receives a message containing an unknown or unsupported attribute, it shall reject the
1219 request. If a Consumer receives a message containing an unknown or unsupported attribute, it shall
1220 silently ignore the attribute. However, Consumers are required to include those attributes in messages

1221 sent back to the Provider. Note in these cases the Consumer is not required to understand or process the
 1222 unsupported attribute, but merely echo it back to the Provider.

1223 5.3 Identifiers

1224 All identifiers (e.g., Resource names, attributes, operations, parameter names) defined by this
 1225 specification, or defined by the way of an extension, shall adhere to the following rules:

- 1226 • Identifier names shall be treated as case sensitive.
- 1227 • Identifier names shall only use the following set of characters:
 - 1228 – Uppercase ASCII (U+0041 through U+005A)
 - 1229 – Lowercase ASCII (U+0061 through U+007A)
 - 1230 – Digits (U+0030 through U+0039)
 - 1231 – Underscore (U+005F)
- 1232 • Identifier names shall not start with a Digit (U+0030 through U+0039).

1233 Note that these rules do not apply to the "name" common attribute defined in clause 5.7.1.

1234 5.4 Attribute constraints

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

1238 **support optional:**

1239 This constraint indicates that support for this attribute is optional. If supported, Providers should advertise
 1240 its support through `ResourceMetadata`. See clause 5.2 for information concerning the processing of
 1241 unsupported and unknown attributes. See clause 5.5.15 regarding empty attribute values.

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

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

1247 **support mandatory:**

1248 This constraint indicates that support for this attribute is required by compliant implementations. If present
 1249 on a nested attribute, this attribute is required to be supported only if the parent attribute is supported.
 1250 See clause 5.5.15 regarding empty attribute values.

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

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

1255 **immutable:**

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

1258 **mutable:**

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

1262 **read-only:**

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

1267 **read-write:**

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

1272 **write-only:**

1273 This Consumer constraint indicates that the attribute may be updated by Consumers but are not
 1274 retrievable by Consumers, typically for security reasons. Write-only attributes shall appear in the
 1275 serialization of Resources sent to Providers but shall never appear in the serialization of Resources sent
 1276 from Providers.

1277 **5.5 Data types and their serialization**

1278 Unless specifically asked to not include certain attributes in the Resource representation, the absence of
 1279 an optional attribute in the representation means that the attribute has no value (i.e., is undefined),
 1280 meaning there is no notion of an optional attribute having an implied value. Note that a client cannot
 1281 distinguish (from just looking at the returned representation) whether a particular attribute is not supported
 1282 from one that does not exist. Likewise, an absent attribute from a Resource representation as the input to
 1283 an update operation means that the Consumer is requesting that the Provider remove that attribute.

1284 The following clauses describe the data types and values that are used within the model definition tables.

1285 **5.5.1 boolean**

1286 A value as defined by `xs:boolean` per [XML Schema – Part 2](#), with the exception that the only allowable
 1287 values are either "true" or "false." The value is case sensitive.

1288 If serialized in JSON, these values shall be of JSON type: *boolean*

1289 If serialized in XML, these values shall be of XML Schema type: *xs:boolean*

1290 **5.5.2 dateTime**

1291 A value as defined by `xs:dateTime` per [XML Schema – Part 2](#), which is consistent with DMTF DSP4004
 1292 and ISO 8601. The timestamp should preserve time zone information, i.e., include a local time component
 1293 and an offset from UTC.

1294 Any constraints on the specific ranges allowed for any particular attribute are specified by that attribute's
 1295 definition or at runtime by the Provider by the way of the metadata discovery mechanisms defined by this
 1296 specification.

1297 For example, Monday, May 25, 2012, at 1:30:15 PM EST is represented as:

1298 `2012-05-25T13:30:15-05:00`

1299 If serialized in JSON, these values shall be of JSON type: *string*

1300 If serialized in XML, these values shall be of XML Schema type: *xs:dateTime*

1301 **5.5.3 duration**

1302 A value as defined by *xs:duration* per [XML Schema – Part 2](#). Any constraints on the specific ranges
1303 allowed for any particular attribute shall be specified by that attribute's definition or at runtime by the
1304 Provider by the way of the metadata discovery mechanisms defined by this specification.

1305 If serialized in JSON, these values shall be of JSON type: *string*

1306 If serialized in XML, these values shall be of XML Schema type: *xs:duration*

1307 **5.5.4 integer**

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

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

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

1313 **5.5.5 string**

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

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

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

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

1320 **5.5.6 ref**

1321 A reference to another Resource.

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

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

1327 **JSON serialization:**

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

1331

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

1332 **XML serialization:**

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

```
1336 <myvolume href="xs:anyURI"/>
```

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

1342 For JSON:

```
1343 "myvolume": { "href": string, ... }
```

1344 and in XML:

```
1345 <myvolume href="xs:anyURI"> xs:any* </myvolume>
```

1346 However, for brevity the extensibility points are excluded from the serialization of the Resources.

1347 **5.5.7 map**

1348 A list of key/value pairs. The same "key" shall not be used more than once within an attribute. The "key" is
 1349 case sensitive.

1350 If serializing an attribute of type map, the serialization shall omit this attribute in case of an empty map.

1351 **5.5.8 structure**

1352 Attributes of this type are complex attributes made up of a set of nested attributes. For each attribute of
 1353 this type, there is an additional table defining those nested attributes.

1354 A nested structure can be considered a complex type definition. Structures may be named or unnamed.
 1355 Table 2 is an example of named structure:

1356 **Table 2 – Named structure**

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

1357 **JSON serialization:**

1358 In JSON, the name of the structure (i.e., of the type it represents) never appears. In other words, whether
 1359 the structure is named or not does not matter. An attribute named `"systemIncidents"` of type
 1360 `"summary"` (as above) is serialized as follows:

```
1361 "systemIncidents": {
1362   "low": number,
1363   "medium": number,
1364   "high": number,
1365   "critical": number
```

1366 }

1367 **XML serialization:**

1368 In XML, the name of the structure (i.e., of the type it represents) never appears. In other words, whether
 1369 the structure is named or not does not matter. The same previous "systemIncidents" example is
 1370 serialized so that the structure sub-attributes become XML attributes of a <systemIncidents> XML
 1371 element wrapper:

```
1372 <systemIncidents low="xs:integer" medium="xs:integer" high="xs:integer"  
1373 critical="xs:integer"/>
```

1374 **NOTE** A large number of sub-attributes of atomic type in a structure may be represented alternatively as XML child
 1375 elements for better readability. Both options are available; however, the same structure shall be serialized the same
 1376 way across Resources.

1377 **5.5.9 byte[]**

1378 An arbitrary set of bytes meant to represent a block of binary data. Any constraints on this type for any
 1379 particular attribute shall be specified by that attribute's definition or at runtime by the Provider by the way
 1380 of the metadata discovery mechanisms defined by this specification.

1381 If serialized in JSON, these values shall be of JSON type: *string*

1382 If serialized in XML, these values shall be of XML Schema type: *xs:hexBinary*

1383 **5.5.10 URI**

1384 The format and syntax of the attributes of type "URI" is defined by [RFC3986](#).

1385 Unless otherwise noted, this specification does not mandate whether Providers use relative or absolute
 1386 URI in the HTTP response bodies.

1387 If URIs are specified as relative URIs, they shall be relative to the *baseURI*.

1388 The algorithm used for converting a relative URI to an absolute URI shall be as described in section 5.2 of
 1389 [RFC3986](#). Table 3 illustrates how relative URIs are resolved against base URIs:

1390 **Table 3 – Converting a relative URI to an absolute URI**

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

1391 If relative URIs are used, the *baseURI* shall end with a trailing slash and relative URIs shall not begin
 1392 with a leading slash. This format is consistent with most URI resolve utilities and produces the same
 1393 results as a simple string concatenation algorithm.

1394 If serialized in JSON, these values shall be of JSON type: *string*

1395 If serialized in XML, these values shall be of XML Schema type: *xs:anyURI*

1396 **5.5.11 Array**

1397 An array represents an ordered list of items of the same type. An array shall always appear as an
 1398 attribute of a Resource, and is only accessible as such (it is not a separately addressable Resource). If a
 1399 Resource is deleted, the items in its arrays shall also be deleted. However, in case these items were just

1400 references to other Resources, these referred Resources are not affected. (See the semantics of
1401 references in 5.7.)

1402 Attributes that are arrays are defined by using the notation `itemType []`, where `itemType` is the type
1403 name for each item of the array. If the type is a structure, not a simple data type, it is recommended as a
1404 convention in the model that the name of an array be the plural of a name that characterizes each item.
1405 For example, an array of volume items or of references to these may be named "volumes."

1406 **JSON serialization:**

1407 Within this specification, arrays in JSON are serialized with a wrapper property. The wrapper name shall
1408 be same as the attribute name for the array. For example, a "things" attribute of type "thing[]" is
1409 serialized as:

```
1410 "things" : [  
1411   { ... }, +  
1412 ] ?
```

1413 If the items in the array are structures, the structure name shall not be present in the JSON serialization.

1414 In the case of an array of references, i.e., where the "ref" type applies to each element of the array,
1415 each element shall simply be serialized as an href property within a JSON array. For example, an array
1416 "things" of type "ref[]" is serialized as:

```
1417 "things": [  
1418   { "href": string }, +  
1419 ] ?
```

1420 **NOTE** If serializing arrays, conformant implementations shall not include empty arrays (i.e., arrays that contain no
1421 child properties) in the JSON serialization. Notice that the child of the "things" property is defined with a "+",
1422 meaning at least one child is required. This requirement ensures that the JSON serialization is minimized and only
1423 includes the wrapping "things" element if, and only if, there is at least one "thing" in the array.

1424 **XML serialization:**

1425 The XML serialization of arrays requires each item of the array to be represented as an element. These
1426 elements shall be consecutive and contiguous in the serialization and the name of each element (tag
1427 name) shall be the name of the element type (the name that appears before "[]" in the array type). For
1428 example, a "things" attribute shall be serialized as a list of items named "thing":

```
1429 <thing>  
1430   ...  
1431 </thing> *
```

1432 There is no wrapper element for an array in XML.

1433 In the case of an array of references, i.e., where the "ref" type applies to each element of the array, the
1434 array is serialized as a list of XML elements without wrapper. Each element is named per an array "item
1435 name" specified in the attribute's definition. For example, an array "things" of type "ref[]" where the
1436 array "item name" is "thing" is serialized as:

```
1437 <thing href="xs:anyURI"/> +
```

1438 **5.5.12 Collection**

1439 A Collection is a group of Resources of the same type. In contrast with arrays, Collections are themselves
1440 Resources that have their own URI and can be independently accessed. Collections also allow for an

1441 optimized and convenient interaction pattern by providing a specialized set of operations that avoid
1442 replacing a large number of items when updating the set, as with arrays.

1443 This specification uses Collections if the set of grouped items is modified often and potentially by multiple
1444 Consumers. Conversely, arrays are used if it is expected that the list of items is not modified often or can
1445 be easily modified by substitution of the entire list, and thus the overhead of managing these items as
1446 separate Resources might be unjustified and burdensome.

1447 An item in a Collection, i.e. a Collection item, is an embedded structure that contains a reference to a
1448 Resource and optionally additional attributes (see "accessory" attributes, defined later). For convenience,
1449 the Resource referred to by a Collection item is called here a Resource item of the Collection.

1450 A Resource may be referenced by more than one Collection. If such a Resource is deleted, every
1451 Collection that references this Resource shall remove the corresponding item. While different Collections
1452 contain entries of different Resource types, all Collections follow the pattern described below:

- 1453 • A Collection shall contain an `id` attribute that acts as a "self pointer." Retrieving the data at this
1454 reference shall return the Collection. In the XML representation, each Collection shall be wrapped
1455 by a `<Collection>` element.
- 1456 • A Collection shall contain a `count` attribute that indicates the number of Resources in the
1457 Collection at the time the Collection was queried.
- 1458 • Adding new Resources to the Collection shall be done either via the "add" operation defined
1459 within the Collection (when the Resource is also created) or via the "insert" operation (when the
1460 Resource already exists).

1461 Deleting an item from the Collection shall be done either via a "delete" operation on the Resource item
1462 itself (if the Resource has to be discarded) or via the "remove" Collection operation (if the Resource must
1463 still exist outside the Collection). Collections that are attributes of other Resources are represented with
1464 attribute type "collection[itemType]." The Resource type of the Collection items are specified
1465 inside the brackets; for example an attribute that is a Collection of Machines is expressed as
1466 "collection[Machine]." Attributes of such types are serialized as a reference to a Collection
1467 Resource instead of holding the Collection itself as value. For brevity, while these attributes are
1468 "references" the word "ref" or "reference" does not appear in the model definition tables - instead the type
1469 of such an attribute is making abstraction of the reference and more explicitly shows as
1470 "collection[itemType]."

1471 In the serializations below, the Collection items are represented by items in the
1472 *ResourceSpecificGroupingName* JSON array, and by *ResourceSpecificElementName* elements in the
1473 XML representation.

1474 **Serialization:**

1475 The serialization of Collections shall adhere to the following pattern:

1476 **JSON serialization:**

```
1477 { "resourceURI": string,
1478   "id": string,
1479   "updated": string, ?
1480   "parent": string, ?
1481   "count": number,
1482   "resourceSpecificGroupingName": [
1483     { "resourceURI": string,
```

```

1484     "id": string,
1485     "name": string, ?
1486     "description": string, ?
1487     "created": string, ?
1488     "updated": string, ?
1489     "parent": string, ?
1490     "properties": { string: string, + }, ?
1491     ... resource specific data ...
1492     "operations": [
1493         { "rel": "edit", "href": string }, ?
1494         { "rel": "delete", "href": string } ?
1495     ] ?
1496     ...
1497 } +
1498 ], ?
1499 "operations": [
1500     { "rel": "add", "href": string } ?
1501     { "rel": "insert", "href": string } ?
1502     { "rel": "remove", "href": string } ?
1503 ]
1504 ...
1505 }
    
```

1506 **XML serialization:**

```

1507 <Collection resourceURI="xs:anyURI" xmlns="http://schemas.dmtf.org/cimi/2">
1508     <id> xs:anyURI </id>
1509     <updated> xs:dateTime </updated> ?
1510     <parent> xs:anyURI </parent> ?
1511     <count> xs:integer </count>
1512     <ResourceSpecificElementName>
1513         <id> xs:anyURI </id>
1514         <name> xs:string </name> ?
1515         <description> xs:string </description> ?
1516         <created> xs:dateTime </created> ?
1517         <updated> xs:dateTime </updated> ?
1518         <parent> xs:anyURI </parent> ?
1519         <property key="xs:string"> xs:string </property> *
1520         ... resource specific data ...
1521         <operation rel="edit" href="xs:anyURI"/> ?
1522         <operation rel="delete" href="xs:anyURI"/> ?
1523     </ResourceSpecificElementName>
    
```

```

1524     </ResourceSpecificElementName> *
1525     <operation rel="add" href="xs:anyURI"/> ?
1526     <operation rel="insert" href="xs:anyURI"/> ?
1527     <operation rel="remove" href="xs:anyURI"/> ?
1528     <xs:any>*
1529 </Collection>

```

1530 Where the `resourceURI` attributes shall contain the Collection or Resource specific URIs for that type
 1531 of Collection, and `resourceSpecificGroupingName` and `ResourceSpecificElementName`
 1532 shall be replaced with the name of the Collection-specific Resource name, e.g., `machines` in JSON or
 1533 `Machine` in XML.

1534 The above serialization shows that each entry in a Collection may contain “resource specific data” beside
 1535 the reference to the Resource item and the common attributes. This placeholder represents two kinds of
 1536 data:

- 1537 a) Optionally some *accessory attributes* that represent accessory information for the use of this
 1538 reference in the context of the Resource owning that Collection (the accessory attributes) – e.g.,
 1539 the “initial location” of a referenced `Volume`, in a Collection of `Volumes` associated with a
 1540 `Machine`. Accessory attributes – if any - are part of the definition of each specific Collection..
- 1541 b) All or a subset of the attributes of the corresponding Resource items. How much of the
 1542 Resource item is expanded in the serialization of the Collection is controlled by expansion
 1543 mechanisms described later.

1544 If accessory attributes exist for items in a Collection, the “*resourceSpecificGroupingName*” or
 1545 “*ResourceSpecificElementName*” is not just identifying the Resource type of Collection items, but is a
 1546 unique name specific to this combination of accessory attributes and Resource type – e.g., for `Volumes`
 1547 with initial location, it may be “`locatedVolume`”. Also the `resourceURI` of the Collection is unique to this
 1548 combination. Because of this accessory attribute, the Collection of `Volumes` is said to be “enhanced”, as
 1549 opposed to “basic” for a Collection without accessory attribute.

1550 The serialization of Collections follows these additional rules:

- 1551 • A Provider may limit the number of Resources returned in the Collection. The Consumer can
 1552 determine this has occurred by comparing the number of returned Resources with the value of
 1553 the “Count” attribute and any Collection subsetting query parameters it specified. In this case,
 1554 the Consumer is advised to specify filter query parameters (see 4.1.6.1) to reduce the number
 1555 of entries returned, or retrieve them in batches by issuing multiple requests with Collection
 1556 subsetting query parameters (see 4.1.6.2)
- 1557 • As with all Resources in the CIMI model, each Resource in the Collection shall have an `id`
 1558 attribute that acts as a “self pointer.” Retrieving the data at this reference shall return just that
 1559 one Resource and not any parent Resource, such as the Collection or array attribute.
- 1560 • The serialization of a Collection may be controlled (see 4.1.6.4 `$expand` query parameter) to
 1561 show more or less of each Resource item. By default, each entry in the Collection will show just
 1562 a reference (URL) to the Resource item, along with the “common” attributes of the Resource
 1563 item. Alternatively, the Resource item may be expanded partially or fully when querying the
 1564 Collection.
- 1565 • As with all arrays, if there are no Resources in the Collection, the serialization of the list shall be
 1566 omitted.

1567 **5.5.12.1 Adding an item to a Collection**

1568 Invoking the "add" operation of a Collection shall create a new Resource and add it to the Collection. The
 1569 contents of the request body shall be either a representation of the new Resource being added to the
 1570 Collection, or a representation of the Template associated with the new Resource being created and
 1571 resource specific data attributes.

1572 If creating a new Resource the "add" operation shall contain:

- 1573 • The "common attributes" as defined by clause 5.7.1
- 1574 • The Resource specific data needed to create it. This data shall either be a reference to the
 1575 Resource-specific Template Resource or the Resource-specific Template Resource itself
 1576 inlined.
- 1577 • Accessory attributes—if any—that represent accessory information for the use of the reference in
 1578 the context of the Resource owning that Collection (the associative attributes)
- 1579 • In the XML case, a wrapper element (named after the pattern <ResourceNameCreate>)

1580 For example, to create a new Machine (which requires the use of a Template) and add it to the
 1581 MachineCollection, the "add" operation of the MachineCollection shall be serialized as
 1582 follows:

1583 **JSON serialization:**

```
1584 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineCreate", ?
1585   "name": string, ?
1586   "description": string, ?
1587   "properties": { string: string, + }, ?
1588   "machineTemplate": { "href": string ?}
1589   ...
1590 }
```

1591 **XML serialization:**

```
1592 <MachineCreate xmlns="http://schemas.dmtf.org/cimi/2">
1593   <name> xs:string </name> ?
1594   <description> xs:string </description> ?
1595   <property key="xs:string"> xs:string </property> *
1596   <machineTemplate href="xs:anyURI"? />
1597   <xs:any>*
1598 </MachineCreate>
```

1599 The MachineCollection has a new Machine:

1600 **JSON serialization:**

```
1601 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Machine",
1602   "id": string,
1603   "name": string,
1604   ...
1605 }
```

1606 XML serialization:

```

1607 <Machine xmlns="http://schemas.dmtf.org/cimi/2">
1608   <id> xs:anyURI </id>
1609   <name> xs:string </name>
1610   ...
1611 </Machine>

```

1612 The processing of the "add" operation shall adhere to the semantics defined in clause 4.2.1.1.

1613 Regardless of whether a Template is used, the "add" operation shall create the new Resource and add it
 1614 to the Collection and a reference (URI) to the new entry shall be returned in the response message in the
 1615 HTTP Location header.

1616 5.5.12.2 Inserting an item in a Collection

1617 Invoking the "insert" operation of a Collection shall add to the Collection a new reference to an existing
 1618 Resource. The contents of the request body shall specify the URL of the existing Resource being added.

1619 In order to add an existing Volume to the volumes Collection of a Machine, the request body of the
 1620 "insert" operation shall be serialized as follows:

1621 JSON serialization:

```

1622 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Volume",
1623   "initialLocation": string,
1624   "volume": { "href": string }
1625 }

```

1626 XML serialization:

```

1627 <Volume xmlns="http://schemas.dmtf.org/cimi/2">
1628   <initialLocation> xs:string </initialLocation>
1629   <volume href="xs:string"/>
1630 </Volume>

```

1631 Note that "initialLocation" is an accessory attributes to each reference of Volume. The definition of the
 1632 volumes Collection of the Machine Resource describes the accessory attribute(s) for this Collection.

1633 5.5.12.3 Removing an item from a Collection

1634 Invoking the "remove" operation of a Collection shall delete the specified item in the Collection, i.e. the
 1635 Resource reference along with accessory attributes if any, without destroying the referenced Resource
 1636 item itself. The contents of the request body shall be the URL of the Resource item being removed.

1637 In order to remove a Volume from the volumes Collection of a Machine, the request body of the
 1638 "remove" operation shall be serialized as follows:

1639 JSON serialization:

```

1640 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Volume",
1641   "volume": { "href": string }
1642 }

```

1643 **XML serialization:**

```
1644 <Volume xmlns="http://schemas.dmtf.org/cimi/2">
1645     <volume href="xs:string"/>
1646 </Volume>
```

1647 Removing the referenced Resource (here a `Volume`) deletes the related entry from the Collection. This
 1648 deletes the reference but not the Resource itself.

1649 Deleting the referenced Resource via a DELETE operation on the Resource itself (here a `Volume`) also
 1650 deletes the related entry from the Collections that reference this Resource – i.e., it has the effect of a
 1651 “remove” on the Collection, in addition to deleting the referenced Resource.

1652

1653 **5.5.13 "Any" type**

1654 Some attributes are polymorphic and can hold various data types, the list of which is indicated in their
 1655 description. In such cases, the type of the attribute shall be indicated as "any" in the model
 1656 representation.

1657 **5.5.14 valueScope**

1658 The `valueScope` type is a specialized map. Its goal is to define possible values for a list of attributes of a
 1659 Resource. The possible values for an attribute are called the “value scope” of the attribute, and a
 1660 combination of attribute value scopes (in form of a map) in a Resource or in the ResourceMetadata is
 1661 called the value scope of the Resource.

1662 Each item in a `valueScope` is a key/value pair where:

- 1663 • The key is the name of an attribute of a Resource – or “**scoped attribute**” – for which a set of
 1664 possible values is defined.
- 1665 • The value is a structure that defines the “**scope**”, i.e., a range, an enumeration or a single
 1666 assigned value for the scoped attribute.

1667 **The scope structure:**

1668 A “scope” structure – or the value part of a key-value item in a `valueScope` – can take one of four forms:

- 1669 1) An assigned single value, along with its (optional) `units`, e.g., for a scoped attribute named “`cpu`”:

```
1670 "cpu": { "value": 2000, "units": "megahertz" }
```

- 1671 2) A range of values, along with its optional `units`, and an optional `increment` e.g., for a scoped
 1672 attribute named “`memory`”. The range may be open-ended: either the `minimum` or the `maximum`
 1673 may be missing. The `increment` specifies the allowed values starting from the `minimum` and
 1674 upward - i.e., the allowed values are of the form: `minimum+N*(increment)`, where $N \geq 0$, or starting
 1675 from the `maximum` and downward in case there is no `minimum`, i.e., allowed values are of the
 1676 form: `maximum-N*(increment)`.

```
1677 "memory": { "minimum": 4000, "maximum": 10000, "units": "kibibytes", "default":  
1678 4000, "increment": 2000 }
```

- 1679 3) An enumeration (or values), along with its (optional) `units`, e.g., for a scoped attribute named
 1680 “`cpuArch`”:

```
1681 "cpuArch": { "values": [ "68000", "Alpha", "ARM", " PA_RISC"], "default": "PA_RISC"  
1682 }
```

- 1683 4) Simply a required `units`, e.g., for a scoped attribute named “`capacity`”:

```
1684     "capacity": { "units": "megabytes" }
```

1685 If a valueScope is associated with a Resource type, it shall be in form of an attribute named “vscope”, of
1686 type array of valueScope (i.e., valueScope[]).

1687 An example of valueScope for the MachineConfiguration Resource:

```
1688     "vscope" : [ {
1689         "cpu": { "value": 1 },
1690         "memory": { "minimum": 4, "maximum": 32, "units": "GbB", "default": 4, "increment":
1691         2 },
1692         "cpuArch": { "values": [ "68000", "Alpha", "ARM", " PA_RISC", "i5"], "default":
1693         "i5" }
1694     } ]
```

1695 Semantics

1696 A value scope may be defined either for the attributes of a Resource type described in
1697 ResourceMetadata, or for attribute(s) of a particular Resource, or for both. The semantics is as follows:

- 1698 • If a value scope is associated with a Resource (i.e., this Resource has a “vscope” attribute), a
1699 scoped attribute of this Resource shall only take values and units within its scope, when updated or
1700 when set (if it were not set at creation time).
- 1701 • If a value scope is associated with a Resource type as described in ResourceMetadata (i.e., the
1702 ResourceMetadata instance for this Resource type has a “vscope” attribute), any Resource of this
1703 type shall have its attributes take values within the defined scope.
- 1704 • If both a Resource and its related ResourceMetadata have some value scope associated with them,
1705 then the value scope of the Resource should be defined so that any attribute value within this value
1706 scope is also within the value scope of its related ResourceMetadata (i.e., the value scope of the
1707 Resource attribute is included in the value scope of the ResourceMetadata for this attribute if any.
1708 The actual value scope of an attribute that is scoped both in its Resource and in its
1709 ResourceMetadata , is the intersection of the two value scopes.

1710 The semantics of a value scope for Consumer and Provider is as follows:

- 1711 • If an attribute of a Resource is scoped, a Consumer shall set a value (creation or update request)
1712 compatible with the value scope of this attribute, including constraints specified by an increment if it
1713 is present.
- 1714 • For any other case where the Consumer sets an incompatible value, the Provider shall return a 4xx
1715 error code.

1716 Usage in a template

1717 When defined in a template Resource, or a Resource used in a template (e.g., MachineConfiguration),
1718 the value scope is intended to restrict also the similar attributes in Resources generated from this
1719 template. In such a case, the attributes of the generated Resource that were scoped in the template of
1720 this Resource, are also scoped similarly in the generated Resource. In order to make this scope more
1721 explicit, a Provider should replicate in the generated Resource the value scope – or the relevant part of it
1722 – defined in the template.

1723 In order to better enforce the value scope of Resources, a Provider may predefine a set of templates that
1724 a Consumer may use. This Provider may prevent the Consumer from creating additional templates while
1725 letting the Consumer modify (within scope) the attributes of the predefined templates.

1726 For example, a Provider may create a set of predefined MachineConfiguration Resources with a read-
1727 only vscope attribute. The Provider may further prevent Consumers from creating new
1728 MachineConfiguration instances – or only by offering a “copy” operation on existing ones. In this way, the
1729 Provider effectively constrains the Consumer to only use the predefined MachineConfiguration Resources

1730 yet allows the Consumer to modify the configuration attributes within the value scope of each predefined
 1731 MachineConfiguration.

1732 **Semantics of valueScope array in a Resource**

1733 The value scope of a Resource shall be represented by an array of valueScope instances, even if in
 1734 many cases this array will contain a single valueScope instance. This allows for expressing dependencies
 1735 between values of different attributes of a same Resource. In such cases, the scoped attributes of the
 1736 Resource must satisfy either valueScope instance in this array.

1737 In the following example, vscope is an array of two valueScope items:

```
1738     "vscope": [ {
1739         "cpuSpeed": { "minimum": 2, "maximum": 4, "units": "GHz", "default": 2.5},
1740         "memory": {"minimum": 2000000, "maximum": 10000000, "units": "KbB", "increment":
1741         2000000 },
1742         "cpuArch": { "value": "i5" }
1743     }, {
1744         "memory": { "minimum": 4000000, "maximum": 32000000, "units": "KbB" },
1745         "cpuArch": { "values": [ "68000", "Alpha", " PA_RISC" ] }
1746     } ]
```

1747 This valueScope means that the Provider supports MachineConfigurations with either cpuArch of value
 1748 i5, or of a value that is one of { "68000", "Alpha", " PA_RISC" }. In the first case (i5), the
 1749 memory must be within the 2GbB-10GbB range and cpuSpeed must be between 2-4 GHz, while in the
 1750 second case the memory must be within the 4GbB-32GbB range.

1751 The following pseudo-schemas describe the serialization of the valueScope map in both JSON and XML:

1752 **JSON serialization:**

```
1753     ( "value": any,
1754     "units": string ? ) |
1755     ( "values": [ any,+ ],
1756     "units": string ,?
1757     "default": string ? ) |
1758     ( "minimum": number, ?
1759     "maximum": number, ?
1760     "units": string ,?
1761     "default": number, ?
1762     "increment": number ? )
```

1764 **XML serialization:**

```
1765     ( <value> xs:any </value>
1766     <units> xs:string </units> ? ) |
1767     (<value> xs:any </value> +
1768     <units> xs:string </units> ?
1769     <default> xs:any </default> ? ) |
```

```

1770 (<minimum> xs:integer </minimum> ?
1771 <maximum> xs:integer </maximum> ?
1772 <units> xs:string </units> ?
1773 <default> xs:integer </default> ?
1774 <increment> xs:integer </increment> ? )
    
```

1775 A Provider who supports value scopes shall set the ValueScopes capability (ResourceMetadata) to “true”.

1776 **5.5.15 Empty attribute values**

1777 Attributes of the following types are omitted in cases where they have an empty value: string, map, array,
 1778 and Collection. Apart from being “Provider optional” or “Consumer optional”, an empty value is the third
 1779 reason that the serialization schema contains an ‘?’ or an ‘*’ for an attribute.

1780 Other attribute types do not have empty values and shall not be omitted from the serialization for this
 1781 reason.

1782 **5.6 Units**

1783 Some of the Resources defined by this specification have attributes that describe an amount of
 1784 something that belongs to, or is associated with, that Resource. For example, the `Machine` Resource
 1785 has a `memory` attribute that describes "the size of the memory allocated to this machine." The allowable
 1786 units of these attributes are listed in Table 4. Their meaning is defined in [IEC 80000-13:2008](#). Their
 1787 numerical equivalents are provided here for convenience:

1788 **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

1789 **5.7 Resources**

1790 CIMI Resources are representations of actual – either virtual or physical – resources available in a Cloud.
 1791 Resources are identified and separately accessible by their URI. Every Resource has a type which is
 1792 described in this section. A Resource type defines a set of attributes and of operations.

1793 **5.7.1 Common Resource attributes**

1794 Resources, except for the Collection Resource, shall support the following common attributes defined in
 1795 Table 5. A Collection Resource shall support the `id` attribute, the `updated` attribute and the `parent`
 1796 attribute, as defined in Table 5.

1797 **Table 5 – Common attributes**

Attribute	Type	Description
id	URI	The unique URI identifying this Resource; assigned upon Resource creation. This attribute value shall be unique in the Provider’s cloud. Constraints: Provider: support mandatory; immutable

Attribute	Type	Description															
		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: Provider: support mandatory; 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: Provider: support mandatory; 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: 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: Provider: support optional; mutable Consumer: support optional; read-only															
parent	<i>ref</i>	A reference to a Resource of which this Resource is a component (see "composition" relationship, section 5.10.2) – i.e. a reference to its first parent Resource. Constraints: Provider: support mandatory; 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 1272 1214 1556"> <thead> <tr> <th colspan="3">Name</th> </tr> <tr> <th colspan="3"><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: Provider: support mandatory; 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															
vscope	<i>valueScope[]</i>	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. Constraints: Provider: support optional; mutable Consumer: support optional; read-only															

1798 The following pseudo-schemas describe the serialization of these attributes in both JSON and XML:

1799 **JSON serialization:**

```
1800 "id": string,
1801 "name": string, ?
1802 "description": string, ?
1803 "created": string, ?
1804 "updated": string, ?
1805 "properties": { string: string, + }, ?
1806 "vscope" : [ valueScope, * ], ?
```

1807 **XML serialization:**

```
1808 <id> xs:anyURI </id>
1809 <name> xs:string </name> ?
1810 <description> xs:string </description> ?
1811 <created> xs:dateTime </created> ?
1812 <updated> xs:dateTime </updated> ?
1813 <property key="xs:string"> xs:string </property> *
1814 <vscope> valueScope </vscope> *
```

1815 5.8 Operations

1816 All Resource operations defined by this specification are optional for Providers to support. Consumers, by
 1817 the way of examination of a Resource's ResourceMetadata, can determine which operations are
 1818 supported. However, even for those operations that are supported Consumers still need to examine each
 1819 Resource's representation to determine which operations are supported at that moment. Whether an
 1820 operation is supported is based on a number of factors, including the state of the Resource and access
 1821 control rights of the Consumer. Also see clause 4.2. Operations and states are coupled; i.e., if
 1822 implementing a state-changing Resource operation defined in this specification, the corresponding
 1823 state(s) shall also be implemented. See the Resource-specific "Operations" clauses for additional detail.

1824 The "State" attribute of Resources that have this attribute shall only change value if

- 1825 • an operation is performed on this Resource and this operation requires a state change, or
- 1826 • an error occurred, in this case the "State" attribute shall obtain the value "ERROR".

1827 For example, for a 'start' operation on a Machine both the STARTING and the STARTED states are
 1828 required to be supported by the Machine, while the Machine can only leave the STARTED state after
 1829 another state changing operation is requested, unless an error occurs.

1830 Providers can define additional operations and states. Such extensions shall fall into one of these
 1831 categories:

- 1832 a) A new operation that starts from a CIMI-defined state, or leads to a CIMI-defined state, or both.
 1833 In the latter case, if a CIMI-defined operation already exists for this transition between two
 1834 CIMI-defined states, it shall also be supported by the Provider in addition to the new operation.
- 1835 b) A new Resource state. In that case, a new operation that leads to that state shall also be
 1836 created. In other words, a Provider-defined operation has to be performed before a
 1837 Provider-defined state can be reached.
- 1838 c) A new operation that transitions between two Provider-defined states.

1839 5.9 Alternative model formats

1840 It is expected that this specification is implemented by using a variety of technologies. As a convenience,
1841 the definition of the model elements are provided in alternative formats that are easily consumable by
1842 technology-specific tooling.

1843 In the event of inconsistencies between the various formats, the normative text within this specification
1844 takes precedence over the XML Schemas and alternative formats, which in turn take precedence over
1845 examples.

1846 5.10 Relationships between Resources

1847 5.10.1 Referencing across Resources

1848 Resources may refer each other. This referencing expresses a directional relationship in which there is a
1849 *referring* Resource and a *referred* Resource. Depending on the cardinality of such relationships, there are
1850 two representations:

- 1851 • For 1-to-1 referencing, the URL of the referred Resource appears as an attribute in the referring
1852 Resource.
- 1853 • For 1-to-n referencing, the referred Resources (all of the same type) are grouped in a
1854 Collection, the URL of which appears as an attribute in the referring Resource. In that case, the
1855 *referring* Resource does not refer directly to the referred Resources, but instead to a Collection
1856 Resource that contains references to the *referred* Resources.

1857 If a *referred* Resource is deleted but not the *referring* Resource(s), then in case of a 1-to-1 relationship
1858 the reference shall be set to empty in every *referring* Resource, and in case of a 1-to-n relationship the
1859 reference shall be removed from any Collection where it appears as an item.

1860 5.10.2 Composition Relationship between Resources

1861 A Resource is component of another Resource if its `parent` attribute refers to the latter Resource. This
1862 relationship is transitive.

1863 If a Resource is deleted, its component Resource(s) is(are) also automatically deleted.

1864 In case of a Collection Resource that is referred by a Resource R, Expressing a composition relationship
1865 from the Collection Resource items to R is done by setting the `parent` attribute of each Resource item
1866 to the Collection Resource and by setting the `parent` attribute of the Collection Resource to the
1867 Resource R. A Resource is said to be parent of its components.

1868 For example a Machine is parent of its related Disk Resources via the `disks` Collection: the `parent`
1869 attribute of a Disk is set to the `disks` Collection, and the `parent` attribute of the `disks` Collection is
1870 set to the Machine.

1871 5.11 Resource metadata

1872 Implementations of this specification should allow for Consumers to discover the metadata associated
1873 with each supported Resource type, for a given Cloud Entry Point. Doing so allows for the discovery of
1874 Provider-defined constraints on the CIMI defined attributes as well as discovery of any new extension
1875 attributes or operations that the Provider may have defined. A `ResourceMetadata` instance contains
1876 metadata describing a particular Resource type – e.g., `Network`, or `Machine` – including any Provider-
1877 specific capabilities or features Note that while this specification declares the `ResourceMetadata` as
1878 mutable attributes, it is expected that only administrative users associated with the Provider will update
1879 them. Consequently they remain read-only for Consumers.

1880 Each Resource's metadata shall contain the following pieces of information:

1881

Table 6 – ResourceMetadata attributes

Name	ResourceMetadata																			
Type URI	http://schemas.dmtf.org/cimi/2/ResourceMetadata																			
Attribute	Type	Description																		
typeURI	URI	A unique URI associated with, and denoting, the described Resource type. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
name	string	The name of the described Resource type. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
attributes	attribute[]	A set of Provider-defined metadata that can be used by clients to discover any metadata associated with each attribute of the described Resource type, including the set of extension attributes not defined in this specification. Each attribute shall contain the following nested data: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Name</th> <th colspan="2">attribute</th> </tr> <tr> <th>Data</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>string</td> <td>The name of the attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td> </tr> <tr> <td>namespace</td> <td>URI</td> <td>The namespace in which this attribute is defined. It is recommended that a dereference of this URI returns information about the attribute. This shall not be present if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td> </tr> <tr> <td>type</td> <td>string</td> <td>The data type of the attribute. This shall not be present if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI-defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td> </tr> <tr> <td>required</td> <td>boolean</td> <td>Indicates whether this Resource requires this attribute to be present. If absent the implied value is "false." 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	attribute		Data	Type	Description	name	string	The name of the attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	namespace	URI	The namespace in which this attribute is defined. It is recommended that a dereference of this URI returns information about the attribute. This shall not be present if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	type	string	The data type of the attribute. This shall not be present if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI-defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	required	boolean	Indicates whether this Resource requires this attribute to be present. If absent the implied value is "false." Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
Name	attribute																			
Data	Type	Description																		
name	string	The name of the attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
namespace	URI	The namespace in which this attribute is defined. It is recommended that a dereference of this URI returns information about the attribute. This shall not be present if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
type	string	The data type of the attribute. This shall not be present if describing a CIMI-defined attribute, but shall be present if describing a non-CIMI-defined attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
required	boolean	Indicates whether this Resource requires this attribute to be present. If absent the implied value is "false." Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																		
vscope	valueScope[]	The vscope attribute may be present on a ResourceMetadata Resource. In that case, the value scope represented by this attribute does not apply to the attributes of the ResourceMetadata 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 typeURI attribute. Consequently this value scope is about the list of attributes described in the attributes attribute. Constraints: Provider: support optional; mutable Consumer: support optional; read-write																		
capabilities	capability[]	A set of Provider-defined metadata that can be used by Consumer to discover any capability or feature provided by this Provider. Each capability shall contain the following nested data: <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 20%;">Name</th> <td>capability</td> </tr> </thead> </table>	Name	capability																
Name	capability																			

Name	ResourceMetadata																							
Type URI	http://schemas.dmtf.org/cimi/2/ResourceMetadata																							
Attribute	Type	Description																						
		<table border="1"> <thead> <tr> <th>Data</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>string</td> <td>The name of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</td> </tr> <tr> <td>uri</td> <td>URI</td> <td>A URI that uniquely identifies the capability at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td> </tr> <tr> <td>description</td> <td>string</td> <td>The human-readable description of the semantic of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</td> </tr> <tr> <td>value</td> <td>any</td> <td>The value of the capability. The specific type varies depending on the definition of the capability. If not present the capability defaults to a "boolean" type with a value of "true" indicating that the specific capability is supported by the Provider. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td> </tr> </tbody> </table>	Data	Type	Description	name	string	The name of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write	uri	URI	A URI that uniquely identifies the capability at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	description	string	The human-readable description of the semantic of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write	value	any	The value of the capability. The specific type varies depending on the definition of the capability. If not present the capability defaults to a "boolean" type with a value of "true" indicating that the specific capability is supported by the Provider. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	Constraints: Provider: support optional; mutable Consumer: support optional; read-write						
		Data	Type	Description																				
		name	string	The name of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write																				
		uri	URI	A URI that uniquely identifies the capability at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																				
		description	string	The human-readable description of the semantic of the capability. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write																				
value	any	The value of the capability. The specific type varies depending on the definition of the capability. If not present the capability defaults to a "boolean" type with a value of "true" indicating that the specific capability is supported by the Provider. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																						
actions	action[]	A set of Provider-defined operations that can be used by consumers to act on the Resource. This set represents all operations defined for this described Resource type, which may be a superset of those operations a particular Consumer is actually allowed to use. The subset of allowed operations for a particular Consumer shall be those operations returned to this Consumer if querying an instance of the described Resource type. Note that this attribute is called "actions" so as not to conflict with the ResourceMetadata Resource's own operations. Each operation shall contain the following nested data: <table border="1"> <thead> <tr> <th>Name</th> <th colspan="2">action</th> </tr> <tr> <th>Data</th> <th>Type</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>name</td> <td>string</td> <td>The name of the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td> </tr> <tr> <td>uri</td> <td>URI</td> <td>A URI that uniquely identifies the operation at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td> </tr> <tr> <td>description</td> <td>string</td> <td>The human-readable description of the semantic of the operation. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</td> </tr> <tr> <td>method</td> <td>string</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>string</td> <td>The body mimeType of the request message; it may depend on the model format chosen by the Provider. Constraints:</td> </tr> </tbody> </table>		Name	action		Data	Type	Description	name	string	The name of the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	uri	URI	A URI that uniquely identifies the operation at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	description	string	The human-readable description of the semantic of the operation. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write	method	string	The protocol-dependent verb to use to perform the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	inputMessage	string	The body mimeType of the request message; it may depend on the model format chosen by the Provider. Constraints:
Name	action																							
Data	Type	Description																						
name	string	The name of the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																						
uri	URI	A URI that uniquely identifies the operation at a global level. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																						
description	string	The human-readable description of the semantic of the operation. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write																						
method	string	The protocol-dependent verb to use to perform the operation. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write																						
inputMessage	string	The body mimeType of the request message; it may depend on the model format chosen by the Provider. Constraints:																						

Name	ResourceMetadata		
Type URI	http://schemas.dmtf.org/cimi/2/ResourceMetadata		
Attribute	Type	Description	
			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
		Constraints: Provider: support optional; mutable Consumer: support optional; read-write	

1882 When implementing or using `ResourceMetadata`, Providers and Consumers shall adhere to the
 1883 syntax and semantics of its attributes as described in Table 6 as well as in the tables describing
 1884 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
 1885 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 1886 Resource in both JSON and XML:

1887 **JSON media type:** application/json

1888 **JSON serialization:**

```

1889 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ResourceMetadata",
1890   "id": string,
1891   "typeURI": string,
1892   "name": string,
1893   "attributes" : [
1894     { "name": string,
1895       "namespace": string, ?
1896       "type": string, ?
1897       "required": boolean, ? } *
1898     ], ?
1899   "vscope" : [ valueScope, * ], ?
1900   "capabilities": [
1901     { "name": string, ?
1902       "uri": string,
1903       "description": string, ?
1904       "value": any } *
1905     ], ?
1906   "actions" : [
1907     { "name": string,
1908       "uri": string,
1909       "description": string, ?
1910       "method": string,
1911       "inputMessage": string, ?
    
```

```

1912     "outputMessage": string ? }, *
1913 ], ?
1914 "operations": [
1915     { "rel": "edit", "href": string }, ?
1916     { "rel": "delete", "href": string } ?
1917 ] ?
1918 ...
1919 }
    
```

1920 **XML media type:** application/xml

1921 **XML serialization:**

```

1922 <ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/2">
1923     <id> xs:anyURI </id>
1924     <name> xs:string </name>
1925     <typeURI> xs:anyURI </typeURI>
1926     <attribute name="xs:string" namespace="xs:anyURI"? type="xs:string"?
1927         required="xs:boolean"? /> *
1928     </attribute> *
1929     <vscope> valueScope </vscope> *
1930     <capability name="xs:string"? uri="xs:anyURI" description="xs:string"?>
1931         xs:any*
1932     </capability> *
1933     <action name="xs:string" uri="xs:anyURI" description="xs:string"?
1934         method="xs:string" inputMessage="xs:string"?
1935         outputMessage="xs:string"? /> *
1936     <operation rel="edit" href="xs:anyURI"/> ?
1937     <operation rel="delete" href="xs:anyURI"/> ?
1938     <xs:any>*
1939 </ResourceMetadata>
    
```

1940 Additional metadata about the Resource or attributes may be included by the Provider.

1941 5.11.1 Capabilities

1942 Table 7 describes the capability URIs defined by this specification. Providers may define new URIs and it
 1943 is recommended that these URIs be dereferencable such that Consumers can discover the details of the
 1944 new capability. The "Resource Name" column contains the name of the Resource that may contain the
 1945 specified capability within its ResourceMetadata. The "Capability Name" column contains the name
 1946 of the specified capability and shall be unique within the scope of the corresponding Resource. Each
 1947 capability's URI shall be constructed by appending the "Resource Name", a slash (/), and the "Capability
 1948 Name" to "http://schemas.dmtf.org/cimi/2/capability/". For example, the Machine's "InitialState"
 1949 capability shall have a URI of:

1950 <http://schemas.dmtf.org/cimi/2/capability/Machine/InitialState>

1951 Capabilities that apply to the Provider in general, and are not specific to any one Resource, shall be
 1952 associated with the `CloudEntryPoint` Resource (in case a capability applies only to the
 1953 `CloudEntryPoint` Resource itself, its definition indicates this).

1954 Each one of these capabilities may be set to some value, or may be absent. The meaning of an absent
 1955 capability is defined as follows:

- 1956 • For boolean-valued capabilities: same as a “false” value.
- 1957 • For other capabilities that use a single value or a list of values among an enumeration: same as
 1958 no particular preference or restriction being enforced for this value.

1959

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 Resource that may be created via a template.
CloudEntryPoint	MinimalListing	If true, only the Resources that are direct components of the CEP (i.e. with their <code>parent</code> reference set to a CEP Collection URI) shall be referred by the top Collections of the CEP. If false, every top CEP Collection for a particular Resource type will refer to all the Resources of this type within the CEP usage domain.
System	SystemComponentTemplateByValue	If true, the Provider shall support the specification of ComponentTemplates by value in SystemTemplates.
Machine	DefaultInitialState	If this capability is set, unless otherwise provided (e.g., by a MachineTemplate “initialState” attribute), the Provider shall set a new Machine to this state value, assuming the value is compatible with the InitialStates capability, if set.
Machine	InitialStates	If this capability is set, and if using a MachineTemplate that has an “initialState” attribute, a Consumer shall use an initialState value from the set of values of this capability.
Machine	MachineConfigByValue	If true, the Provider shall support specifying MachineConfigurations by value. If true, the MachineTemplateByValue shall also have the value true.
Machine	MachineCredentialByValue	If true, the Provider shall support specifying Credentials by value in Machine create operations. If true, the

Resource Name	Capability Name	Description
		MachineTemplateByValue capability shall also have the value true.
Machine	MachineImageByValue	If true, the Provider shall support specifying MachineImages by value in Machine create operations. If true, the MachineTemplateByValue capability shall also have the value true.
Machine	MachineVolumeTemplatesByValue	If true, the Provider shall support specifying VolumeTemplates by value in Machine create operations. If, then the MachineTemplateByValue capability shall also have the value true.
Machine	MachineTemplateByValue	If true, the Provider shall support specifying MachineTemplates by value in Machine create operations.
Machine	MachineStopForce	If true, the Provider shall support the "force" option on the stop and restart operations on Machines.
Machine	MachineStopForceDefault	If true, the Provider shall forcefully stop Machines if no other indication is provided. Otherwise, the Provider shall gracefully stop Machines.
Machine	RestoreFromImage	If true, the Provider supports restoring Machines from MachineImages that are not SNAPSHOT MachineImages.
Machine	UserData	If set, indicates which userData injection method shall be used by the Provider.
Machine	MachineAvailabilityLevel	If true, the Provider supports the notion of an availability level for the Machine Resource. The availability level and its value constraints are advertised as an extension attribute by the way of the Machine and MachineTemplate ResourceMetadata.
Credential	CredentialTemplateByValue	If true, the Provider shall support specifying CredentialTemplates by value in Credential create operations.
Volume	SharedVolumeSupport	If true, the Provider shall support that a single Volume Resource 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	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

Resource Name	Capability Name	Description
		capability.
NetworkInterface	NetworkInterfaceTemplateByValue	If true, the Provider shall support specifying NetworkInterface Templates by value in NetworkInterface create operations.
NetworkInterface	DefaultInitialState	If this capability is set, unless otherwise provided (e.g., by a NetworkInterfaceTemplate "initialState" attribute), the Provider shall set a new NetworkInterface to this state value, assuming the value is compatible with the InitialStates capability, if set.
NetworkInterface	InitialStates	If this capability is set, and if using a NetworkInterfaceTemplate that has an "initialState" attribute, a Consumer shall use an initialState value from the set of values of this capability.
NetworkService	NetworkServiceTemplateByValue	If true, the Provider shall support specifying NetworkService Templates by value in NetworkService create operations.
NetworkService	DefaultInitialState	If this capability is set, unless otherwise provided (e.g., by a NetworkServiceTemplate "initialState" attribute), the Provider shall set a new NetworkService to this state value, assuming the value is compatible with the InitialStates capability, if set.
NetworkService	InitialStates	If this capability is set, and if using a NetworkServiceTemplate that has an "initialState" attribute, a Consumer shall use an initialState value from the set of values of this capability.
ProtocolEndpoint	ProtocolEndpointTemplateByValue	If true, the Provider shall support specifying ProtocolEndpoint Templates by value in ProtocolEndpoint create operations.
ProtocolEndpoint	DefaultInitialState	If this capability is set, unless otherwise provided (e.g., by a ProtocolEndpointTemplate "initialState" attribute), the Provider shall set a new ProtocolEndpoint to this state value, assuming the value is compatible with the InitialStates capability, if set.
ProtocolEndpoint	InitialStates	If this capability is set, and if using a ProtocolEndpointTemplate that has an "initialState" attribute, a Consumer shall use an initialState value from the set of values of this capability.
ProtocolSegment	ProtocolSegmentTemplateByValue	If true, the Provider shall support specifying ProtocolSegment Templates by value in ProtocolSegment create operations.
ProtocolSegment	DefaultInitialState	If this capability is set, unless otherwise provided (e.g., by a ProtocolSegmentTemplate "initialState" attribute), the Provider shall set a new ProtocolSegment to this state value, assuming the value is compatible with the InitialStates capability, if set.
ProtocolSegment	InitialStates	If this capability is set, and if using a ProtocolSegmentTemplate that has an "initialState" attribute, a Consumer shall use an initialState value from the set of values of this capability.
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.

1960 The following examples show the ResourceMetadata for a Machine that advertises some of its
1961 capabilities:

1962 **JSON serialization:**

```
1963 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ResourceMetadata",
1964   "id": "http://example.com/types/Machine",
1965   "typeURI": "http://schemas.dmtf.org/cimi/2/Machine",
1966   "name": "Machine",
1967   "capabilities": [
1968     { "uri":
1969       "http://schemas.dmtf.org/cimi/2/capability/Machine/MachineConfigByValue",
1970       "value": true },
1971     { "uri":
1972       "http://schemas.dmtf.org/cimi/2/capability/Machine/MachineImageByValue",
1973       "value": true },
1974     { "uri":
1975       "http://schemas.dmtf.org/cimi/2/capability/Machine/DefaultInitialState",
1976       "value": "STARTED" }
1977   ]
1978 }
```

1979 **XML serialization:**

```
1980 <ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/2">
1981   <id> http://example.org/types/Machine </id>
1982   <typeURI> http://schemas.dmtf.org/cimi/2/Machine </typeURI>
1983   <name> Machine </name>
1984   <capability
1985 uri="http://schemas.dmtf.org/cimi/2/capability/Machine/MachineConfigByValue">
1986     true
1987   </capability>
1988   <capability
1989 uri="http://schemas.dmtf.org/cimi/2/capability/Machine/MachineImageByValue">
1990     true
1991   </capability>
1992   <capability
1993 uri="http://schemas.dmtf.org/cimi/2/capability/Machine/DefaultInitialState">
1994     STARTED
1995   </capability>
1996 </ResourceMetadata>
```

1997 **5.11.2 ResourceMetadataCollection Resource**

1998 A ResourceMetadataCollection Resource represents the Collection of ResourceMetadata
1999 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. Note that

2000 modifications of the Resources within this Collection are typically reserved for administrator types of CIMI
 2001 Consumers. This Resource shall be serialized as follows:

2002 **JSON serialization:**

```

2003 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ResourceMetadataCollection",
2004     "id": string,
2005     "count": number,
2006     "resourceMetadatas": [
2007         { "resourceURI": "http://schemas.dmtf.org/cimi/2/ResourceMetadata",
2008           "id": string,
2009           ... remaining ResourceMetadata attributes ...
2010         }, +
2011     ], ?
2012     "operations": [ { "rel": "add", "href": string } ? ]
2013     ...
2014 }
  
```

2015 **XML serialization:**

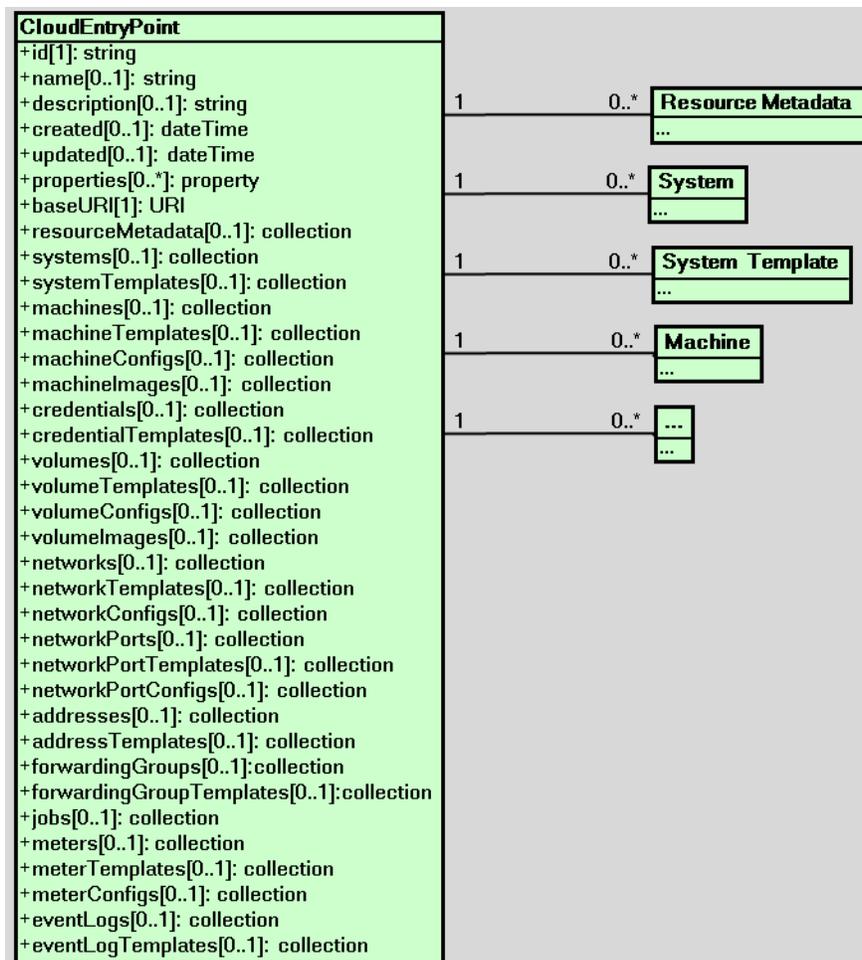
```

2016 <Collection
2017     resourceURI="http://schemas.dmtf.org/cimi/2/ResourceMetadataCollection"
2018     xmlns="http://schemas.dmtf.org/cimi/2">
2019     <id> xs:anyURI </id>
2020     <count> xs:integer </count>
2021     <ResourceMetadata>
2022         <id> xs:anyURI </id>
2023         ... remaining ResourceMetadata attributes ...
2024     </ResourceMetadata> *
2025     <operation rel="add" href="xs:anyURI"/> ?
2026     <xs:any>*
2027 </Collection>
  
```

2028 **5.12 Cloud Entry Point**

2029 The Cloud Entry Point (`CloudEntryPoint` Resource) represents the entry point into the cloud defined
 2030 by the CIMI Model. It provides a Consumer with a single address (URI) from which the Consumer can
 2031 discover and access all Resources usable by this Consumer. A Cloud Provider may provide different
 2032 CEPs to different Consumers. The Cloud Entry Point (CEP) implements a catalog of Resources, such as
 2033 `Systems`, `SystemTemplates`, `Machines`, `MachineTemplates`, etc., that can be queried and
 2034 browsed by the Consumer.

2035 Figure 1 illustrates the `CloudEntryPoint` and its relationship to other Resources. Although this
 2036 drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor
 2037 normative.



2038

2039

Figure 1 - Cloud Entry Point

2040 If a Consumer issues a read on the CloudEntryPoint Resource, the Provider shall return a
 2041 CloudEntryPoint Resource that only catalogs Resources on which this Consumer is allowed to
 2042 perform operations. Table 8 describes the attributes for the CloudEntryPoint Resource.

2043 If the delete operation is advertised on the CEP, deleting the CloudEntryPoint Resource is also
 2044 deleting all referred Resources.

2045

Table 8 – CloudEntryPoint attributes

Name	CloudEntryPoint	
Type URI	http://www.dmf.org/cimi/2/CloudEntryPoint	
Attribute	Type	Description
baseURI	URI	An absolute URI that references the "base URI" of the Provider. This URI shall be used to convert relative URIs to Resources within this Provider to absolute URIs. See the "URIs" clause of 5.5. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
resourceMetadata	collection [Resource Metadata]	A reference to ResourceMetadata 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

Name	CloudEntryPoint	
Type URI	http://www.dmf.org/cimi/2/CloudEntryPoint	
Attribute	Type	Description
		specification defines nor does it have any extension attributes. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
systems	<i>collection</i> <i>[System]</i>	A reference to the <i>SystemCollection</i> of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
systemTemplates	<i>collection</i> <i>[System</i> <i>Template]</i>	A reference to the <i>SystemTemplateCollection</i> of this CloudEntry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
machines	<i>collection</i> <i>[Machine]</i>	A reference to the <i>MachineCollection</i> of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
machineTemplates	<i>collection</i> <i>[Machine</i> <i>Template]</i>	A reference to the <i>MachineTemplateCollection</i> of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
machineConfigs	<i>collection</i> <i>[Machine</i> <i>Configuration]</i>	A reference to the <i>MachineConfigurationCollection</i> of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
machineImages	<i>collection</i> <i>[Machine</i> <i>Image]</i>	A reference to the <i>MachineImageCollection</i> of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
credentials	<i>collection</i> <i>[Credential]</i>	A reference to the <i>CredentialCollection</i> of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
credentialTemplates	<i>collection</i> <i>[Credential</i> <i>Template]</i>	A reference to the <i>CredentialTemplateCollection</i> of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumes	<i>collection</i> <i>[Volume]</i>	A reference to the <i>VolumeCollection</i> of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumeTemplates	<i>collection</i> <i>[Volume</i> <i>Template]</i>	A reference to the <i>VolumeTemplateCollection</i> of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumeConfigs	<i>collection</i> <i>[Volume</i> <i>Configuration]</i>	A reference to the <i>VolumeConfigurationCollection</i> of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
volumeImages	<i>collection</i>	A reference to the <i>VolumeImageCollection</i> of this Cloud Entry

Name	CloudEntryPoint	
Type URI	http://www.dmf.org/cimi/2/CloudEntryPoint	
Attribute	Type	Description
	<i>[Volume Image]</i>	Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networks	<i>collection [Network]</i>	A reference to the NetworkCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkTemplates	<i>collection [Network Template]</i>	A reference to the NetworkTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
segments	<i>collection [Protocol Segment]</i>	A reference to the ProtocolSegmentCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
segmentTemplates	<i>collection [Protocol Segment Template]</i>	A reference to the ProtocolSegmentTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
endpoints	<i>collection [Protocol Endpoint]</i>	A reference to the ProtocolEndpointCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
endpointTemplates	<i>collection [Protocol Endpoint Templates]</i>	A reference to the ProtocolEndpointTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
interfaces	<i>collection [Network Interface]</i>	A reference to the NetworkInterfaceCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
interfaceTemplates	<i>collection [Network Interface Templates]</i>	A reference to the NetworkInterfaceTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkServices	<i>collection [Network Service]</i>	A reference to the NetworkServiceCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkServiceTemplates	<i>collection [Network Service Template]</i>	A reference to the NetworkServiceTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

Name	CloudEntryPoint	
Type URI	http://www.dmf.org/cimi/2/CloudEntryPoint	
Attribute	Type	Description
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

2046 Each of the Collections mentioned in Table 8 are defined within the related Resource definition clauses.
 2047 For example, the MachineCollection Resource is defined in clause 5.14.2 as part of the
 2048 Machine-related Resources. When implementing or using CloudEntryPoint, Providers and
 2049 Consumers shall adhere to the syntax and semantics of its attributes as described in Table 8 as well as in
 2050 the tables describing embedded Resources or related Collections. Both Consumer and Provider shall
 2051 serialize this Resource as described below. The following pseudo-schemas (see notation in 1.3) describe
 2052 the serialization of the Resource in both JSON and XML:

2053 **JSON media type:** application/json

2054 **JSON serialization:**

```

2055 { "resourceURI": "http://schemas.dmtf.org/cimi/2/CloudEntryPoint",
2056   "id": string,
2057   "name": string, ?
2058   "description": string, ?
2059   "created": string, ?
2060   "updated": string, ?
2061   "properties": { string: string, + }, ?
2062   "baseURI": string,
2063   "resourceMetadata": { "href": string }, ?
2064   "systems": { "href": string }, ?
2065   "systemTemplates": { "href": string }, ?

```

```

2066 "machines": { "href": string }, ?
2067 "machineTemplates": { "href": string }, ?
2068 "machineConfigs": { "href": string }, ?
2069 "machineImages": { "href": string }, ?
2070 "credentials": { "href" string }, ?
2071 "credentialTemplates": { "href" string }, ?
2072 "volumes": { "href": string }, ?
2073 "volumeTemplates": { "href": string }, ?
2074 "volumeConfigs": { "href": string }, ?
2075 "volumeImages": { "href": string }, ?
2076 "networks": { "href": string }, ?
2077 "networkTemplates": { "href": string }, ?
2078 "segments": { "href": string }, ?
2079 "segmentTemplates": { "href": string }, ?
2080 "endpoints": { "href": string }, ?
2081 "endpointTemplates": { "href": string }, ?
2082 "interfaces": { "href": string }, ?
2083 "interfaceTemplates": { "href": string }, ?
2084 "networkServices": { "href": string }, ?
2085 "networkServiceTemplates": { "href": string }, ?
2086 "jobs": { "href": string }, ?
2087 "meters": { "href": string }, ?
2088 "meterTemplates": { "href": string }, ?
2089 "meterConfigs": { "href": string }, ?
2090 "eventLogs": { "href": string }, ?
2091 "eventLogTemplates": { "href": string }, ?
2092 "operations": [
2093     { "rel": "edit", "href": string } ?
2094 ] ?
2095 ...
2096 }

```

2097 **XML media type:** application/xml

2098 **XML serialization:**

```

2099 <CloudEntryPoint xmlns="http://schemas.dmtf.org/cimi/2">
2100   <id> xs:anyURI </id>
2101   <name> xs:string </name> ?
2102   <description> xs:string </description> ?
2103   <created> xs:dateTime </created> ?
2104   <updated> xs:dateTime </updated> ?

```

```

2105     <property key="xs:string"> xs:string </property> *
2106     <baseURI> xs:anyURI </baseURI>
2107     <resourceMetadata href="xs:anyURI"/> ?
2108     <systems href="xs:anyURI"/> ?
2109     <systemTemplates href="xs:anyURI"/> ?
2110     <machines href="xs:anyURI"/> ?
2111     <machineTemplates href="xs:anyURI"/> ?
2112     <machineConfigs href="xs:anyURI"/> ?
2113     <machineImages href="xs:anyURI"/> ?
2114     <credentials href="xs:anyURI"/> ?
2115     <credentialTemplates href="xs:anyURI"/> ?
2116     <volumes href="xs:anyURI"/> ?
2117     <volumeTemplates href="xs:anyURI"/> ?
2118     <volumeConfigs href="xs:anyURI"/> ?
2119     <volumeImages href="xs:anyURI"/> ?
2120     <networks href="xs:anyURI"/> ?
2121     <networkTemplates href="xs:anyURI"/> ?
2122     <segments href="xs:anyURI"/> ?
2123     <segmentTemplates href="xs:anyURI"/> ?
2124     <endpoints href="xs:anyURI"/> ?
2125     <endpointTemplates href="xs:anyURI"/> ?
2126     <interfaces href="xs:anyURI"/> ?
2127     <interfaceTemplates href="xs:anyURI"/> ?
2128     <networkServices href="xs:anyURI"/> ?
2129     <networkServiceTemplates href="xs:anyURI"/> ?
2130     <jobs href="xs:anyURI"/> ?
2131     <meters href="xs:anyURI"/> ?
2132     <meterTemplates href="xs:anyURI"/> ?
2133     <meterConfigs href="xs:anyURI"/> ?
2134     <eventLogs href="xs:anyURI"/> ?
2135     <eventLogTemplates href="xs:anyURI"/> ?
2136     <operation rel="edit" href="xs:anyURI"/> ?
2137     <xs:any>*
2138 </CloudEntryPoint>

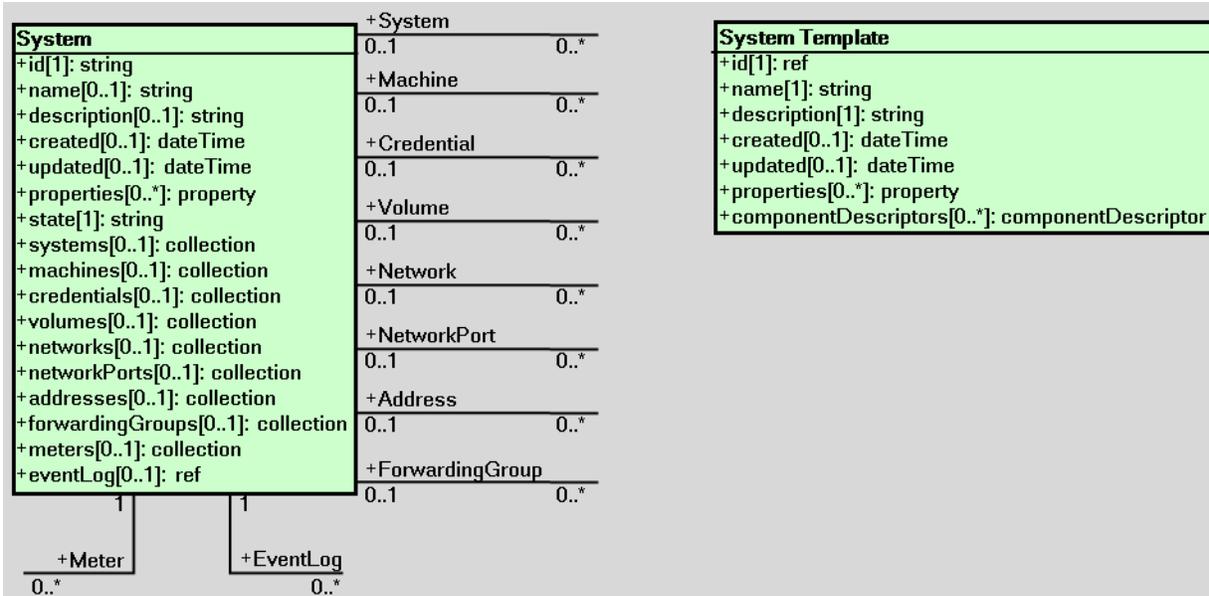
```

2139 5.12.1 Operations

2140 This Resource supports the Read and Update operations.

2141 **5.13 System Resources and relationships**

2142 Figure 2 illustrates the Resources involved in constructing a `System` and their relationships. Although
 2143 this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor
 2144 normative.



2145 **Figure 2 - System Resources**

2146 **5.13.1 System**

2147 A `System` is a realized Resource that consists of one or more `Networks`, `Volumes`, `Machines`,
 2148 (and others) that could be connected and associated with each other. A `System` can be created from the
 2149 interpretation of a `SystemTemplate`. A `System` can be operated and managed as a single Resource
 2150 and usually forms a stack of service. For example, a shopping cart system consists of machines for web
 2151 servers and databases, network addresses for public access, and volumes for database files. A `System`
 2152 has several "top-level" attributes that are Collections of references to Resources of various types. Each
 2153 one of these Collections shall contain references to Resource items of the related type that are either
 2154 components of the `System`, or used by the `System`. In the following, the term "component" [of a `System`]
 2155 means a Resource that has the `System` as parent Resource (either directly or transitively as defined in
 2156 5.10.2). Every Resource item in a top-level Collection attribute of a `System` has `parent` set to this
 2157 Collection.

2158 By default, all Resources that are created as the result of a `System` creation are also (components of the
 2159 `System`. A Resource that is component of a `System` has its life cycle directly tied to the life cycle of the
 2160 `System` as a consequence of the composition semantics. In particular, if a `System` is deleted, all of its
 2161 component Resources are deleted. Generally, operations on a `System` translate into operations on its
 2162 component Resources.

2163 However, a Resource component of a `System` may in turn use some other Resources that are not
 2164 component of this `System`, e.g., a `Machine` in a `System` can use a `Volume` that is neither
 2165 component of the `Machine`, nor a component of the `System`.

2166 A Resource referred by a `System` may be used by the `System` without being its component. Such a
 2167 Resource has its `parent` attribute set to a Resource other than the `System` (e.g. the CEP) or other than
 2168 any of its components, Such a used Resource may be directly referred to in the top-collection of the
 2169 `System`.

2170 For example, a Network may be created independently from any System, directly by adding to the
 2171 `networks` CEP collection. A Consumer may then want a System to use that Network while keeping the
 2172 Network external to the System i.e. not as a component that would be deleted when the System is
 2173 deleted. Such a Network may still be inserted in the `networks` System collection, while having its
 2174 `parent` attribute referring to the CEP as originally set. Alternatively, the Network could be made a
 2175 component of the System by setting its `parent` attribute to the System Resource.

2176

2177 Note that a Resource may not be component of more than one System at any point in time (unless there
 2178 is an component relationship between these Systems.)

2179 Table 9 describes the System attributes.

2180 **Table 9 – System attributes**

Name	System	
Type URI	http://schemas.dmtf.org/cimi/2/System	
Attribute	Type	Description
state	string	The operational state of the System. Allowed values are: (See 5.14.1.) CREATING: The System is in the process of being created. STARTING/STARTED/STOPPING/STOPPED/PAUSING/PAUSED/SUSPENDING/SUSPENDED: The System shall be in one of these states if all the Machines referenced by the System are in that state. See clause 5.14.1 for the list of available actions based on the state of a Machine. Such transitional states may just indicate that all Machines in a System are undergoing the same operation (e.g., "start"), without the System being actually operated on (e.g., no "start" done at System level). An actual operation on a System may be traced by querying the "job" entity. MIXED: The System shall be in this state if either no Machines are referenced by this System or Machines referenced by this System are in varying states. Such varying states are likely to occur when an operation is in progress on a System, resulting in transitions of its Machine states toward a new common state (e.g., STOPPED, STARTED) but at a different pace, or sequentially one after the other. DELETING: The System is in the process of being deleted. ERROR: The Provider has detected an error in the System. The operations that result in transitions to the above defined states are defined in clause 5.13.1.2. <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-only
systems	collection [System]	A list of references to nested Systems that are either components of or used by this System. <u>Constraints:</u> Provider: support optional; mutable Consumer: support optional; read-only
machines	collection [Machine]	A list of references to Machines that are either components of or used by this System. <u>Constraints:</u> Provider: support optional; mutable Consumer: support optional; read-only
credentials	collection [Credential]	A list of references to Credentials that are either components of or used by this System. <u>Constraints:</u> Provider: support optional; mutable

Name	System	
Type URI	http://schemas.dmtf.org/cimi/2/System	
Attribute	Type	Description
		Consumer: support optional; read-only
volumes	<i>collection</i> [Volume]	A list of references <i>Volumes</i> that are either components of or used by this <i>System</i> . Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networks	<i>collection</i> [Network]	A list of references to <i>Network</i> that are either components of or used by this <i>System</i> . Constraints: Provider: support optional; mutable Consumer: support optional; read-only
networkServices	<i>collection</i> [Network Service]	A reference to the <i>NetworkServiceCollection</i> that are either components of or used by this <i>System</i> . Constraints: Provider: support optional; mutable Consumer: support optional; read-only
services	<i>Collection</i> [SystemService]	A list of references to <i>SystemService</i> Resources that represent services supported by this <i>System</i> . Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meters	<i>collection</i> [Meter]	A list of references to <i>Meters</i> monitored for this <i>System</i> , with component semantics. Note that these <i>Meters</i> are for the <i>System</i> and not for any individual component in the <i>System</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>System</i> . Note that this <i>EventLog</i> is for the <i>System</i> and not for any individual component in the <i>System</i> . Constraints: Provider: support optional; mutable Consumer: support optional; read-only

2181 When implementing or using *System*, Providers and Consumers shall adhere to the syntax and
 2182 semantics of its attributes as described in Table 9 as well as in the tables describing embedded
 2183 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 2184 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 2185 both JSON and XML.

2186 **JSON media type:** application/json

2187 **JSON serialization:**

```
2188 { "resourceURI": "http://schemas.dmtf.org/cimi/2/System",
2189   "id": string,
2190   "name": string, ?
2191   "description": string, ?
2192   "created": string, ?
2193   "updated": string, ?
2194   "parent": string, ?
```

```

2195     "properties": { string: string, + }, ?
2196     "state": string,
2197     "systems": { "href": string }, ?
2198     "machines": { "href": string }, ?
2199     "credentials": { "href": string }, ?
2200     "volumes": { "href": string }, ?
2201     "networks": { "href": string }, ?
2202     "networkServices": { "href": string }, ?
2203     "meters": { "href": string }, ?
2204     "eventLog": { "href": string }, ?
2205     "operations": [
2206         { "rel": "edit", "href": string, ("available": boolean)? }, ?
2207         { "rel": "delete", "href": string, ("available": boolean)? }, ?
2208         { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string,
2209         ("available": boolean)? }, ?
2210         { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string,
2211         ("available": boolean)? }, ?
2212         { "rel": "http://schemas.dmtf.org/cimi/2/action/restart", "href": string,
2213         ("available": boolean)? }, ?
2214         { "rel": "http://schemas.dmtf.org/cimi/2/action/pause", "href": string,
2215         ("available": boolean)? }, ?
2216         { "rel": "http://schemas.dmtf.org/cimi/2/action/suspend", "href": string,
2217         ("available": boolean)? }, ?
2218         { "rel": "http://schemas.dmtf.org/cimi/2/action/export", "href": string,
2219         ("available": boolean)? } ?
2220     ] ?
2221     ...
2222 }

```

2223 **XML media type:** application/xml

2224 **XML serialization:**

```

2225 <System xmlns="http://schemas.dmtf.org/cimi/2">
2226   <id> xs:anyURI </id>
2227   <name> xs:string </name> ?
2228   <description> xs:string </description> ?
2229   <created> xs:dateTime </created> ?
2230   <updated> xs:dateTime </updated> ?
2231   <parent> xs:anyURI </parent> ?
2232   <property key="xs:string"> xs:string </property> *
2233   <state> xs:string </state>
2234   <systems href="xs:anyURI"/> ?
2235   <machines href="xs:anyURI"/> ?
2236   <credentials href="xs:anyURI"/> ?

```

```

2237 <volumes href="xs:anyURI"/> ?
2238 <networks href="xs:anyURI"/> ?
2239 <networkServices href="xs:anyURI"/> ?
2240 <meters href="xs:anyURI"/> ?
2241 <eventLog href="xs:anyURI"/> ?
2242 <operation rel="edit" href="xs:anyURI" (available="xs:boolean"? /> ?
2243 <operation rel="delete" href="xs:anyURI" (available="xs:boolean"? /> ?
2244 <operation rel="http://schemas.dmtf.org/cimi/2/action/start"
2245     href="xs:anyURI" (available="xs:boolean"? /> ?
2246 <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
2247     href="xs:anyURI" (available="xs:boolean"? /> ?
2248 <operation rel="http://schemas.dmtf.org/cimi/2/action/restart"
2249     href="xs:anyURI" (available="xs:boolean"? /> ?
2250 <operation rel="http://schemas.dmtf.org/cimi/2/action/pause"
2251     href="xs:anyURI" (available="xs:boolean"? /> ?
2252 <operation rel="http://schemas.dmtf.org/cimi/2/action/suspend"
2253     href="xs:anyURI" (available="xs:boolean"? /> ?
2254 <operation rel="http://schemas.dmtf.org/cimi/2/action/export"
2255     href="xs:anyURI" (available="xs:boolean"? /> ?
2256 <xs:any>*
2257 </System>

```

2258 **5.13.1.1 Attributes of type Collection**

2259 The following clause describes the Collection Resources components of Systems.

2260 **5.13.1.1.1 systems Collection**

2261 The Resource type for each item of this Collection is "System". There is no accessory attribute for the
 2262 items in this Collection, therefore, it is a basic System Collection, the serialization of which follows the
 2263 rules in 5.5.12. See the SystemCollection Resource clause.

2264 **5.13.1.1.2 machines Collection**

2265 The Resource type for each item of this Collection is "Machine". There is no accessory attribute for the
 2266 items in this Collection, therefore, it is a basic Machine Collection (serialized as described in 5.5.12). See
 2267 the MachineCollection Resource clause.

2268 **5.13.1.1.3 credentials Collection**

2269 The Resource type for each item of this Collection is "Credential". There is no accessory attribute for
 2270 the items in this Collection, therefore, it is a basic Credential Collection (serialized as described in
 2271 5.5.12). See the CredentialCollection Resource clause.

2272 **5.13.1.1.4 volumes Collection**

2273 The Resource type for each item of this Collection is "Volume". There is no accessory attribute for the
 2274 items in this Collection, therefore, it is a basic Volume Collection (serialized as described in 5.5.12). See
 2275 the VolumeCollection Resource clause.

2276 **5.13.1.1.5 networks Collection**

2277 The Resource type for each item of this Collection is "Network". There is no accessory attribute for the
 2278 items in this Collection, therefore, it is a basic NetworkCollection Resource as described in
 2279 clause.5.16.2

2280 **5.13.1.1.6 networkServices Collection**

2281 The Resource type for each item of this Collection is "NetworkService". There is no accessory
 2282 attribute for the items in this Collection, therefore, it is a basic NetworkServiceCollection as
 2283 described in clause 5.16.18.

2284 **5.13.1.1.7 meters Collection**

2285 The Resource type for each item of this Collection is "Meter" as defined in clause 5.17.3. There is no
 2286 accessory attribute for the items in this Collection, therefore it is a basic Meter Collection (serialized as
 2287 described in 5.5.12). See the MeterCollection Resource clause.

2288 **5.13.1.2 Operations**

2289 The System Resource supports the Read, Update, and Delete operations. Create is supported through
 2290 the SystemCollection Resource.

2291 The following custom operations are also defined:

2292 **start/stop/restart/pause/suspend**

2293 **/link@rel:** <http://schemas.dmtf.org/cimi/2/action/xxx>

2294 Where "xxx" is either "start", "stop", "restart", "pause", or "suspend".

2295 This operation shall recursively perform the requested operation on each component of the System
 2296 (Machine or sub-System). Note that not all Machines need to be in the same state for this operation
 2297 to be available and the impact of this operation varies depending on the component's current state; see
 2298 clause 5.14.1.2 for more details about performing operations on Machines. If the operation fails for a
 2299 Machine, that Machine shall not be affected by the operation.

2300 **export**

2301 **/link@rel:** <http://schemas.dmtf.org/cimi/2/action/export>

2302 This operation shall export a System along with all Resources component of or used by this System. If
 2303 an export package exists at that URI, it is updated with the values of the System and any component
 2304 management Resources. Otherwise, a new export package is created at that URI with a Media Type as
 2305 specified by the "format" parameter. Other formats may be used if supported, but are not specified by this
 2306 standard.

2307 Input parameters:

- 2308 1) "format" - type: string - optional
- 2309 2) Indicates the Media Type of the exported data. If not present, the default value shall be
 2310 "application/ovf."
- 2311 3)
- 2312 4) "destination" - type: URI - optional
- 2313 5) Indicates the location to where the exported data is placed. If not present, the HTTP response
 2314 Location header shall contain the URL to the exported data. Based on the specific protocol
 2315 specified within the URI, the Consumer might need to provide additional information (such as

2316 credentials) in the "properties" field. In the case of HTTP, a PUT shall be used to place the data
 2317 at the specified location.

2318 Output parameters: None.

2319 **HTTP protocol**

2320 To export a `System`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/export" URI of the
 2321 `System` where the HTTP request body shall be as described below.

2322 **JSON media type:** application/json

2323 **JSON serialization:**

```
2324 { "action": "http://schemas.dmtf.org/cimi/2/action/export",
2325   "format": string, ?
2326   "destination": string, ?
2327   "properties": { string: string, + } ?
2328   ...
2329 }
```

2330 **XML media type:** application/xml

2331 **XML serialization**

```
2332 <Action xmlns="http://schemas.dmtf.org/cimi/2">
2333   <action> http://schemas.dmtf.org/cimi/2/action/export </action>
2334   <format> xs:string </format> ?
2335   <destination> xs:anyURI </destination> ?
2336   <property key="xs:string"> xs:string </property> *
2337   <xs:any>*
2338 </Action>
```

2339 **5.13.2 SystemCollection Resource**

2340 A `SystemCollection` Resource represents a Collection of `System` Resources and follows the
 2341 Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

2342 **JSON serialization:**

```
2343 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemCollection",
2344   "id": string,
2345   "count", number,
2346   "systems": [
2347     { "resourceURI": "http://schemas.dmtf.org/cimi/2/System",
2348       "id": string,
2349       ... remaining System attributes ...
2350     }, +
2351   ], ?
2352   "operations": [
2353     { "rel": "add", "href": string }, ?
```

```

2354     { "rel": "remove", "href": string } ?
2355 { "rel": "insert", "href": string } ?     { "rel":
2356 "http://schemas.dmtf.org/cimi/2/action/import", "href": string } ?
2357 ]
2358 ...
2359 }

```

2360 XML serialization:

```

2361 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/SystemCollection"
2362     xmlns="http://schemas.dmtf.org/cimi/2">
2363     <id> xs:anyURI </id>
2364     <count> xs:integer </count>
2365     <System>
2366         <id> xs:anyURI </id>
2367         ... remaining System attributes ...
2368     </System> *
2369     <operation rel="add" href="xs:anyURI"/> ?
2370     <operation rel="remove" href="xs:anyURI"/> ?
2371     <operation rel="insert" href="xs:anyURI"/> ?
2372     <operation rel="http://schemas.dmtf.org/cimi/2/action/import"
2373 href="xs:anyURI"/> ?
2374     <xs:any>*
2375 </Collection>

```

2376 5.13.2.1 Operations

2377 NOTE The "add" operation requires that a SystemTemplate be used (see 4.2.1.1).

2378 Resources created during the process of creating a System shall be components of the System (see
2379 5.13.1). For example, a componentDescriptor that references a MachineTemplate, and within
2380 that MachineTemplate is a reference to a VolumeTemplate, results in a reference to the new
2381 Machine being added to the System.machines attribute and a reference to the new Volume being
2382 added to the System.volumes attribute. However, if this MachineTemplate refers to an existing
2383 Volume, this Volume shall not be added to the top-level System attributes.

2384 The following custom operations are also defined:

2385 import

2386 **/link@rel:**http://schemas.dmtf.org/cimi/2/action/import

2387 This operation shall import a System. Not only is a System created, but Machines, Volumes, and
2388 Networks and possibly recursive Systems and their components may also be created corresponding
2389 to imported descriptor entries. More detail about this process is in 0.

- 2390 1) Input parameters:"source" - type: URI - mandatory
- 2391 2) Indicates the location from which the imported data is retrieved. Based on the specific protocol
- 2392 specified within the URI, the Consumer might need to provide additional information (such as
- 2393 credentials) in the "properties" field.

2394 Output parameters: None.

2395 **HTTP protocol**

2396 To import a `System`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/import" URI of the
 2397 `SystemCollection` where the HTTP request body shall be as described below.

2398 **JSON media type:** application/json

2399 **JSON serialization:**

```
2400 { "action": "http://schemas.dmtf.org/cimi/2/action/import",
2401   "source": string, ?
2402   "properties": { string: string, + } ?
2403   ...
2404 }
```

2405 **XML media type:** application/xml

2406 **XML serialization**

```
2407 <Action xmlns="http://schemas.dmtf.org/cimi/2">
2408   <action> http://schemas.dmtf.org/cimi/2/action/import </action>
2409   <source> xs:anyURI </source> ?
2410   <property key="xs:string"> xs:string </property> *
2411   <xs:any>*
2412 </Action>
```

2413 **5.13.3 SystemService Resource**

2414 A `SystemService` Resource represents some management service for all or part of the Resources in a
 2415 `System`. A `SystemService` Resource can define diverse types of management services and typically
 2416 holds:

- 2417 (a) Topology information about the service: a list of the Resources concerned by this management
 2418 service, e.g. lists of Machines and Volumes subject to disaster recovery policy, along with the
 2419 Network that connects them, e.g. a Network that supports a load balancing service with an
 2420 external access endpoint.
- 2421 (b) Policy information: configuration data for the service itself.

2422 System components may be listed under more than one `SystemService` Resources. For example, a
 2423 `Machine` may be under a recovery service, while also participating into a load balancing service.

2424 After deployment, management services can be updated although in a way that may be restricted by the
 2425 Provider. Modifying a service configuration can be done in one of the two following ways:

- 2426 (1) Direct attribute update (HTTP PUT): Some attributes of the service may be directly updated by
 2427 the Consumer, if read-write. Such updates typically concern configuration data (see (b) above).
 2428 The Provider will typically advertise the allowed range of values with a `valueScope` field.
- 2429 (2) Operations: Some components of the service may only be modified by operations advertised by
 2430 the Provider for this type of service. Such operations are used for changes that affect the
 2431 managed Resources inside the `System`, e.g. adding a `Machine` in a pool under load balancing, or
 2432 forcing a synchronization with a recovery image. This specification defines a non-exclusive list of
 2433 most common update operations for a basic set of services.

2434 The following list of services are defined in this specification, not exclusive of others:

- 2435 • LoadBalancing service
- 2436 • LocalRecovery service
- 2437 • DisasterRecovery service
- 2438 • Backup service
- 2439 • Autoscaling service

2440 Each one of the above management services requires specific attributes that define a particular service
 2441 type. All SystemService Resources share a `serviceType` attribute that identifies the service type. The
 2442 following sections describe the SystemService Resources for some of the service types.

2443 **5.13.3.1 LocalRecovery service Resource**

2444 This service allows for all or part of the Resources in a System to recover from failure – e.g. a failed
 2445 Machine, or a failed Volume, or both – by maintaining local, up-to-date images of these Resources, inside
 2446 the same System. This service Resource represents the actual service as supported by the Provider,
 2447 showing which System Resources are concerned, and the attributes or configuration attributes of the
 2448 service.

2449 **Table XX – SystemService attributes for LocalRecovery service**

Name	SystemService	
Type URI	http://schemas.dmtf.org/cimi/2/SystemService	
Attribute	Type	Description
<code>serviceType</code>	<i>URI</i>	http://schemas.dmtf.org/cimi/2/SystemService/localrecovery/active or http://schemas.dmtf.org/cimi/2/SystemService/localrecovery/passive
<code>machines</code>	<i>Collection[Recoverable Machine]</i>	<p>A reference to the list of references to <code>Machines</code> in the <code>System</code> that are managed under this <code>SystemService</code>, meaning these benefit from recovery service. Adding a <code>Machine</code> reference to this collection means that the <code>Machine</code> becomes managed under this <code>SystemService</code>.</p> <ul style="list-style-type: none"> • If the <code>serviceType</code> is ending with “/localrecovery/active”: Then each one of the listed <code>Machines</code> has a backup <code>Machine</code>. In case of failure the backup <code>Machine</code> (referred to by the <code>recoverableMachine</code> collection item) shall take over i.e. added to this collection, with a new backup created for it. • If the <code>serviceType</code> is ending with “/localrecovery/passive”: Then each one of the listed <code>Machines</code> has an up-to-date <code>MachineImage</code>. In case of failure the backup <code>Machine</code> is created from the <code>MachineImage</code> and shall replace the failed <code>Machine</code>, i.e. be added to this collection, <p>This Resource items in this Collection are not components of the <code>SystemService</code> Resource: deleting the <code>SystemService</code> does not cause the deletion of the referred <code>Machines</code>.</p> <p>The details of the <code>SystemService</code> behavior (e.g. failover detection, etc.) depends on the Provider’s implementation.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>
<code>networkServices</code>	<i>collection [Network Service]</i>	<p>A reference to the <code>NetworkServiceCollection</code> within the <code>System</code> that support this <code>SystemService</code>.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-only</p>
<code>heartbeat</code>	<i>Integer</i>	Heartbeat frequency, in term of millisecs between an heartbeat and the next.

Name	SystemService	
Type URI	http://schemas.dmtf.org/cimi/2/SystemService	
Attribute	Type	Description
		<p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</p>
replicationType	String	<p>The kind of disk replication data (it does not refer to the Volume Resource) allowable values are: synchronous, asynchronous, none, onlyAtClusterCreation</p> <p>Provider: support mandatory; mutable Consumer: support mandatory; read-only</p>
RPO	Integer	<p>Recovery Point Objective (duration in minutes) in case of asynchronous replica of the disks.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</p>

2450

2451 **5.13.3.1.1 RecoverableMachine Collection**

2452 The referred Resource type for each item of this Collection is “Machine”. However because there are
 2453 accessory attributes, this is not a basic but an enhanced Machine Collection. The accessory attribute is
 2454 defined in Table 14:

2455 **Table 10 – RecoverableMachine accessory attributes**

Name	RecoverableMachine	
Type URI		
Attribute	Type	Description
backupmachine	Ref	<p>An additional reference to the backup Machine in the same System, that supports the Machine referenced by this collection item.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p>

2456 <SERIALIZATIONS TO BE ADDED>

2457 **5.13.3.1.2 Operations**

2458 The localrecovery SystemService Resource supports the Read, Update, and Delete operations. Create is
 2459 supported through the SystemService Collection Resource.

2460 The recoverable Machines collection (`SystemService.machines`) supports the Insert and Remove
 2461 operations, for adding or removing Machines in the recovery service.

2462 The following custom operations are also defined on this SystemService Resource:

2463

2464 **forceSync**

2465 **/link@rel:** <http://schemas.dmtf.org/cimi/2/action/forceSync>

2466 This operation shall synchronize the state of a node onto its backup node, regardless of the scheduled
 2467 synchronization time as dictated by the recovery policies.

2468 Input parameters: primary node.

2469 Output parameters: None.

2470 **HTTP protocol**

2471 <TO BE COMPLETED>

2472

2473 **swapBackup**

2474 **/link@rel:** <http://schemas.dmtf.org/cimi/2/action/swapBackup>

2475 This operation shall swap a Machine and its backup Machine – i.e. replace the Machine with its backup
 2476 and vice-versa.

2477 Some Providers can choose to not make available this operation, not allowing the Consumer to choose
 2478 which backup node turn in primary one.

2479 Input parameters:"node" - type: ref - mandatory
 2480 A reference to the Machine to be replaced by its backup

2481 <TO BE COMPLETED>

2482 Output parameters: None.

2483 **5.13.3.2 DisasterRecovery service Resource**

2484 This service allows for a System to recover from a data center failure – by maintaining a remote, up-to-
 2485 date images of the System.

2486 ...

2487 **5.13.3.3 LocalBalancing service Resource**

2488 This service allows for a System to balance user requests over a pool of equivalent Machines.

2489 **Table XX – SystemService attributes for LocalBalancing service**

Name	SystemService	
Type URI	http://schemas.dmtf.org/cimi/2/SystemService	
Attribute	Type	Description
<i>serviceType</i>	<i>URI</i>	http://schemas.dmtf.org/cimi/2/SystemService/loadbalancing
<i>machines</i>	<i>Collection[Machine]</i>	A reference to the list of references to Machines in the System that are managed under this SystemService. Adding a Machine reference to this collection means that the Machine becomes managed under this SystemService.
<i>networkServices</i>	<i>collection [Network Service]</i>	A reference to the NetworkServiceCollection within the System that support this SystemService. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

2490

2491 **5.13.3.4 Backup service**

2492 This service allows for backing-up Resources (Machines, Volumes) in a System according to some policy.

2493 **5.13.3.5 Autoscaling service**

2494 This service allows for automatically scaling a pool of Machines in a System to accommodate user
2495 requests. It otherwise provides same capabilities as the LocalBalancing service.

2496

2497 **5.13.4 SystemTemplate Resource**

2498 The `SystemTemplate` Resource contains the set of individual descriptors that are necessary to create
2499 the components of a `System`. Each component descriptor can be considered to be the persisted view of
2500 the create operation that instantiates the component. In practice, the Provider interprets the set of
2501 component descriptors as a set of creation operations to be executed in an order compatible with the
2502 dependencies (e.g., attachments or references between components) that are expressed between these
2503 components.

2504 A `SystemTemplate` may include symbolic component references in the descriptors, used to express
2505 links between components of the resulting `System`. A component reference uses the "name" of the target
2506 (referred) component. For example, `<volume href="#newVolume"/>` would reference a `Volume`
2507 named "newVolume." The reference name `–#newVolume–` is replaced by the actual Resource URL in
2508 the instantiated `System`.

2509

2510 Table 11 describes the `SystemTemplate` attributes.

2511

Table 11 – SystemTemplate attributes

Name	SystemTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/SystemTemplate	
Attribute	Type	Description
component Descriptors	<i>componentDescriptor[]</i>	The list of component descriptors describing the components of a <code>System</code> instance realized from this <code>SystemTemplate</code> . For each component descriptor, the corresponding component is created when a <code>System</code> instance is created. Each component descriptor refers to a <code>Template</code> (either by reference or by value), and may also provide additional metadata (name, description, properties). The creation order of components is not specified in <code>SystemTemplate</code> ; in particular the order of the component descriptors in this array is not meaningful in terms of creation order.
	Name	<i>componentDescriptor</i>
	Data	Type Description
	name	<i>string</i> The value of the "name" attribute that is associated with a <code>System</code> component created from this component descriptor. Note: This name is not to be confused with the name that may be present in the component <code>Template</code> – e.g., a <code>MachineTemplate</code> – from which this component is instantiated. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write

Name	SystemTemplate		
Type URI	http://schemas.dmtf.org/cimi/2/SystemTemplate		
Attribute	Type	Description	
		description	<i>string</i> The value of the "description" attribute that is associated with a <code>System</code> component created from this component descriptor. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write
		properties	<i>map</i> The key/value pairs that is associated with a <code>System</code> component created from this component descriptor. Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write
		type	<i>URI</i> The TypeURI of the component to be created from this component descriptor, e.g., for a <code>Machine</code> : http://schemas.dmtf.org/cimi/2/Machine Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
		<component Template>	<i><any></i> A reference either to a component Template or to the Template data itself inlined (i.e., the Template "value"). Note that the exact name of this attribute varies depending on the type of Resource being created, e.g., <code>MachineTemplate</code> for a <code>Machine</code> . This attribute shall contain either: <ul style="list-style-type: none"> • A Template that is provided inline. Such an embedded Template may contain component references, each one of which shall resolve to the URI of a component with same name once created from this <code>SystemTemplate</code>. • A reference to an externally defined Template. Some attribute name/value pairs may be added inside the <code>componentTemplate</code> element to override similar attributes in the referred Template (as described in 4.2.1.1). This example shows how component references can be added to an external Template. Example (JSON): <pre> "machineTemplate": { "href": "http://example.com/machineTemplates/72000", "credential": { "href": "#MyCredential" } </pre> Note: The "credential" attribute in this example assumes that there is another <code>componentDescriptor</code> item named "MyCredential" of type "Credential" in the <code>SystemTemplate</code> . It shall set or override similar attribute in the referred <code>MachineTemplate</code> if instantiating the <code>Machine</code> component. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

Name	SystemTemplate																							
Type URI	http://schemas.dmtf.org/cimi/2/SystemTemplate																							
Attribute	Type	Description																						
		quantity	<p><i>integer</i></p> <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>																						
serviceDescriptors	serviceDescriptor[]	<p>The list of service descriptors for the services to be supported by a <i>System</i> instance realized from this <i>SystemTemplate</i>. For each service descriptor, the corresponding <i>SystemService</i> is created when a <i>System</i> instance is created. The names of the <i>System</i> components subject to the service are listed using the symbolic component reference notation previously described ("<i>#<name></i>"). Because each type of service has different configuration attributes, these are listed separately for service type, in "service type accessory attributes" tables outside this <i>SystemTemplate</i> table.</p> <table border="1"> <thead> <tr> <th colspan="3"><i>serviceDescriptor</i></th> </tr> <tr> <th>Name</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>SystemService</i> instance created from this service descriptor.</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>SystemService</i> instance created from this service 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>SystemService</i> instance created from this service descriptor.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p> </td> </tr> <tr> <td>serviceType</td> <td><i>URI</i></td> <td> <p>The serviceType of the service to be created from this service descriptor, e.g., for a <i>SystemService</i> of type "localRecovery": http://schemas.dmtf.org/cimi/2/SystemService/localrecovery</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p> </td> </tr> <tr> <td><service-specific-attribute></td> <td></td> <td> <p>This is where additional service-specific attributes are listed (see section 5.13.6).</p> </td> </tr> </tbody> </table> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>		<i>serviceDescriptor</i>			Name	Type	Description	name	<i>string</i>	<p>The value of the "name" attribute that is associated with a <i>SystemService</i> instance created from this service descriptor.</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>SystemService</i> instance created from this service 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>SystemService</i> instance created from this service descriptor.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p>	serviceType	<i>URI</i>	<p>The serviceType of the service to be created from this service descriptor, e.g., for a <i>SystemService</i> of type "localRecovery": http://schemas.dmtf.org/cimi/2/SystemService/localrecovery</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>	<service-specific-attribute>		<p>This is where additional service-specific attributes are listed (see section 5.13.6).</p>
<i>serviceDescriptor</i>																								
Name	Type	Description																						
name	<i>string</i>	<p>The value of the "name" attribute that is associated with a <i>SystemService</i> instance created from this service descriptor.</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>SystemService</i> instance created from this service 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>SystemService</i> instance created from this service descriptor.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support optional; read-write</p>																						
serviceType	<i>URI</i>	<p>The serviceType of the service to be created from this service descriptor, e.g., for a <i>SystemService</i> of type "localRecovery": http://schemas.dmtf.org/cimi/2/SystemService/localrecovery</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>																						
<service-specific-attribute>		<p>This is where additional service-specific attributes are listed (see section 5.13.6).</p>																						
meterTemplates	MeterTemplates[]	<p>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>System</i>. Note that the attributes of the <i>MeterTemplate</i> may be specified rather than a</p>																						

Name	SystemTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/SystemTemplate	
Attribute	Type	Description
		reference to an existing MeterTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
eventLog Template	ref	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new System. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
import Image	URI	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. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

2512 When implementing or using `SystemTemplate`, Providers and Consumers shall adhere to the syntax
 2513 and semantics of its attributes as described in Table 11 as well as in the tables describing embedded
 2514 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 2515 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 2516 both JSON and XML

2517 **JSON media type:** application/json

2518 **JSON serialization:**

```

2519 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemTemplate",
2520   "id": string,
2521   "name": string, ?
2522   "description": string, ?
2523   "created": string, ?
2524   "updated": string, ?
2525   "parent": string, ?
2526   "properties": { string: string, + }, ?
2527   "componentDescriptors": [
2528     { "name": string, ?
2529       "description": string, ?
2530       "properties": { string: string, + }, ?
2531       "type": string,
2532       "componentTemplate": {
2533         "href": string, ?
2534         ... ComponentTemplate attributes ... ?
2535       },
2536       "quantity": number ?
2537     }, +
    
```

```

2538 ], ?
2539 "serviceDescriptors": [
2540   { "name": string, ?
2541     "description": string, ?
2542     "properties": { string: string, + }, ?
2543     "serviceType": string,
2544     ...
2545   }, +
2546 ], ?
2547 "meterTemplates": [
2548   { "href": string, ?
2549     ... MeterTemplate attributes ... ?
2550   }, *
2551 ], ?
2552 "eventLogTemplate": {
2553   "href": string, ?
2554   ... EventLogTemplate attributes ... ?
2555 }, ?
2556 "importImage": string , ?
2557
2558 "operations": [
2559   { "rel": "edit", "href": string }, ?
2560   { "rel": "delete", "href": string }, ?
2561   { "rel": "http://schemas.dmtf.org/cimi/2/action/export", "href": string } ?
2562 ] ?
2563 ...
2564 }
    
```

2565 **XML media type:** application/xml

2566 **XML serialization:**

```

2567 <SystemTemplate xmlns="http://schemas.dmtf.org/cimi/2">
2568   <id> xs:anyURI </id>
2569   <name> xs:string </name> ?
2570   <description> xs:string </description> ?
2571   <created> xs:dateTime </created> ?
2572   <updated> xs:dateTime </updated> ?
2573   <parent> xs:anyURI </parent> ?
2574   <property key="xs:string"> xs:string </property> *
2575   <componentDescriptor>
2576     <name> xs:string </name> ?
    
```

```

2577     <description> xs:string </description> ?
2578     <property key="xs:string"> xs:string </property> *
2579     <type> xs:anyURI </type>
2580     <componentTemplate href="xs:anyURI"? >
2581         ... ComponentTemplate attributes ... ?
2582     </componentTemplate> *
2583
2584     <quantity> xs:integer </quantity>
2585 </componentDescriptor> *
2586 <serviceDescriptor>
2587     <name> xs:string </name> ?
2588     <description> xs:string </description> ?
2589     <property key="xs:string"> xs:string </property> *
2590     <serviceType> xs:anyURI </serviceType>
2591     ...
2592 </serviceDescriptor> *
2593
2594 <meterTemplate href="xs:anyURI"? >
2595     ... MeterTemplate attributes ... ?
2596 </meterTemplate> *
2597 <eventLogTemplate href="xs:anyURI"? >
2598     ... EventLogTemplate attributes ... ?
2599 </eventLogTemplate> ?
2600 <importImage > xs:anyURI </importImage> ?
2601 <operation rel="edit" href="xs:anyURI"/> ?
2602 <operation rel="delete" href="xs:anyURI"/> ?
2603 <operation rel="http://schemas.dmtf.org/cimi/2/action/export"
2604 href="xs:anyURI"/> ?
2605     <xs:any>*
2606 </SystemTemplate>

```

2607 5.13.4.1 Operations

2608 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 2609 SystemTemplateCollection Resource.

2610 The following custom operations are also defined:

2611 **export**

2612 **/link@rel:** <http://schemas.dmtf.org/cimi/2/action/export>

2613 This operation shall export a SystemTemplate along with all its component Resources as well as the
 2614 used Resources that are listed in its top-level Collections . If an export package exists at that URI, it is
 2615 updated with the values of the SystemTemplate and any component management Resources.

2616 Otherwise a new export package is created at that URI with a Media Type as specified by the "format"
 2617 parameter. Other formats may be used if supported, but are not specified by this standard.

2618 Input parameters:

- 2619 1) "format" - type: string - optional
- 2620 2) Indicates the Media Type of the exported data. If not present, the default value shall be
 2621 "application/ovf."
- 2622 3) "destination" - type: URI - optional
- 2623 4) Indicates the location to where the exported data is placed. If not present, the HTTP response
 2624 Location header shall contain the URL to the exported data. Based on the specific protocol
 2625 specified within the URI, the Consumer might need to provide additional information (such as
 2626 credentials) in the "properties" field. In the case of HTTP, a PUT shall be used to place the data
 2627 at the specified location.

2628 Output parameters: None.

2629 HTTP protocol

2630 To export a `SystemTemplate`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/export"
 2631 URI of the `SystemTemplate` where the HTTP request body shall be as described below.

2632 **JSON media type:** application/json

2633 **JSON serialization:**

```
2634 { "action": "http://schemas.dmtf.org/cimi/2/action/export",
2635   "format": string, ?
2636   "destination": string, ?
2637   "properties": { string: string, + } ?
2638   ...
2639 }
```

2640 **XML media type:** application/xml

2641 **XML serialization**

```
2642 <Action xmlns="http://schemas.dmtf.org/cimi/2">
2643   <action> http://schemas.dmtf.org/cimi/2/action/export </action>
2644   <format> xs:string </format> ?
2645   <destination> xs:anyURI </destination> ?
2646   <property key="xs:string"> xs:string </property> *
2647   <xs:any>*
2648 </Action>
```

2649 5.13.5 SystemTemplateCollection Resource

2650 A `SystemTemplateCollection` Resource represents the Collection of `SystemTemplate`
 2651 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
 2652 shall be serialized as follows:

2653 **JSON serialization:**

```
2654 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemTemplateCollection",
```

```

2655     "id": string,
2656     "count": number,
2657     "systemTemplates": [
2658         { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemTemplate",
2659           "id": string,
2660           ... remaining SystemTemplate attributes ...
2661         }, +
2662     ], ?
2663     "operations": [
2664         { "rel": "add", "href": string }, ?
2665         { "rel": "http://schemas.dmtf.org/cimi/2/action/import", "href": string } ?
2666     ]
2667     ...
2668 }

```

2669 XML serialization:

```

2670 <Collection
2671     resourceURI="http://schemas.dmtf.org/cimi/2/SystemTemplateCollection"
2672     xmlns="http://schemas.dmtf.org/cimi/2">
2673     <id> xs:anyURI </id>
2674     <count> xs:integer </count>
2675     <SystemTemplate>
2676         <id> xs:anyURI </id>
2677         ... remaining SystemTemplate attributes ...
2678     </SystemTemplate> *
2679     <operation rel="add" href="xs:anyURI"/> ?
2680     <operation rel="http://schemas.dmtf.org/cimi/2/action/import"
2681 href="xs:anyURI"/> ?
2682     <xs:any>*
2683 </Collection>

```

2684 5.13.5.1 Operations

2685 The following custom operations are defined:

2686 import

2687 **/link@rel:** http://schemas.dmtf.org/cimi/2/action/import

2688 This operation shall import a SystemTemplate. Not only is a SystemTemplate created, but
 2689 MachineTemplates, VolumeTemplates, and NetworkTemplates and possibly recursive
 2690 SystemTemplates and their components may also be created, corresponding to imported descriptor
 2691 entries. More detail about this process is in 0.

2692 Input parameters:

2693 1) "source" - type: URI - mandatory

2694 2) Indicates the location from which the imported data is retrieved. Based on the specific protocol
 2695 specified within the URI, the Consumer might need to provide additional information (such as
 2696 credentials) in the "properties" field.

2697 Output parameters: None.

2698 **HTTP protocol**

2699 To import a `SystemTemplate`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/import"
 2700 URI of the `SystemTemplateCollection` where the HTTP request body shall be as described
 2701 below.

2702 **JSON media type:** application/json

2703 **JSON serialization:**

```
2704 { "action": "http://schemas.dmtf.org/cimi/2/action/import",
2705   "source": string, ?
2706   "properties": { string: string, + } ?
2707   ...
2708 }
```

2709 **XML media type:** application/xml

2710 **XML serialization**

```
2711 <Action xmlns="http://schemas.dmtf.org/cimi/2">
2712   <action> http://schemas.dmtf.org/cimi/2/action/import </action>
2713   <source> xs:anyURI </source> ?
2714   <property key="xs:string"> xs:string </property> *
2715   <xs:any>*
2716 </Action>
```

2717

2718 **5.13.6 Service-specific Descriptor attributes**

2719 This section defines the additional attributes specific to each service type that need be added to a
 2720 serviceDescriptor for this service type in the `SystemTemplate`.

2721 **5.13.6.1 Attributes for the LocalRecovery service type**

2722 Service type: http://schemas.dmtf.org/cimi/2/SystemService/localrecovery

2723 **Table XX – Additional attributes for LocalRecovery service**

Service type	localrecovery	
Attribute	Type	Description
machines	<i>String[]</i>	Symbolic references to the <code>Machine</code> components in the <code>System</code> that are subject to the service. Uses the symbolic component reference notation previously described ("#<name>"). Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
network	<i>string</i>	Symbolic reference to the <code>Network Resource</code> in the <code>System</code> that enables this service. The <code>Network</code> shall provide the necessary connections between

Service type	localrecovery	
Attribute	Type	Description
		Machines to support this Service..
heartbeat	Integer	Heartbeat frequency, in term of millisecs between an heartbeat and the next. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
replicationType	String	The kind of disk replication data (it does not refer to the Volume Resource) allowable <u>values are:</u> synchronous, asynchronous, none, onlyAtClusterCreation Provider: support mandatory; mutable Consumer: support mandatory; read- write
RPO	Integer	Recovery Point Objective (duration in minutes) in case of asynchronous replica of the disks. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

2724

2725 **5.13.6.2 Attributes for the LocalBalancing service type**

2726 Service type: <http://schemas.dmtf.org/cimi/2/SystemService/localbalancing>

2727 **Table XX – Additional attributes for LocalBalancing service**

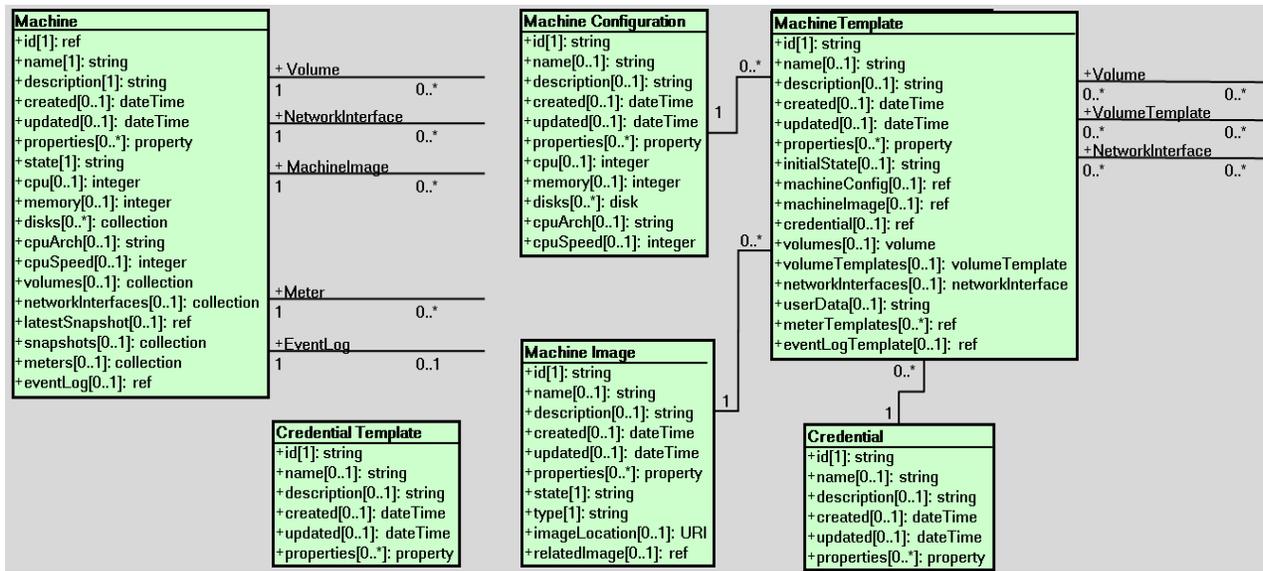
Service type	localbalancing	
Attribute	Type	Description
machines	<i>String[]</i>	Symbolic references to the <i>Machine</i> components in the <i>System</i> that are subject to the service. Uses the symbolic component reference notation previously described (“#<name>”). When a <i>Machine</i> name is listed for which there are several instances specified in the componentDescriptor (<i>quantity</i> attribute), all of them are subject to the service. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
network	<i>string</i>	Symbolic references to the <i>Network</i> Resource in the <i>System</i> that enables this service. The <i>Network</i> shall include a <i>LoadBalancing NetworkService</i> .

2728

2729

2730 **5.14 Machine Resources and relationships**

2731 Figure 3 illustrates the Resources involved in constructing a *Machine* and their relationships. Although
 2732 this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor
 2733 normative.



2734 **Figure 3 - Machine Resources**

2735 **5.14.1 Machine**

2736 An instantiated compute Resource that encapsulates both CPU and Memory. Table 12 describes the
 2737 Machine attributes.

2738 **Table 12 – Machine attributes**

Name	Machine	
Type URI	http://schemas.dmtf.org/cimi/2/Machine	
Attribute	Type	Description
state	string	The operational state of the Machine. Allowed values are: 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. This is equivalent to a "stand-by" state. 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/2/Machine	
Attribute	Type	Description
		<p>FAILED: the <i>Machine</i> is not operational due to some error condition and in accordance to the Provider's policies it is considered <i>failed</i>. This state calls for a recovery procedure, if any.</p> <p>The operations that result in transitions to the above defined states are defined in clause 5.14.1.2.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</p>
cpu	<i>integer</i>	<p>The amount of CPU that this <i>Machine</i> has.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>
memory	<i>integer</i>	<p>The size of the memory (RAM) in kibibytes allocated to this <i>Machine</i>. If this value is increased, it implies that the <i>Machine</i> is allocated more RAM, and vice versa if the value is decreased.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>
disks	<i>collection</i> <i>[Disk]</i>	<p>A reference to the list of disks (local storage) that are part of the <i>Machine</i>. Adding an element to this list creates a disk. The <i>Disk</i> Resources are components of the <i>Machine</i>.</p> <p>Note: The <i>Disk</i> Resource type is defined in clause 5.14.1.1.1.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-only</p>
cpuArch	<i>string</i>	<p>The CPU architecture that is supported by <i>Machines</i> created by using this configuration.</p> <p>Allowed values are: 68000, Alpha, ARM, Itanium, MIPS, PA_RISC, POWER, PowerPC, x86, x86_64, z/Architecture, SPARC. Providers may define additional values.</p> <p>Constraints: Provider: support optional; immutable Consumer: support optional; read-only</p>
cpuSpeed	<i>integer</i>	<p>The approximate CPU speed of this <i>Machine</i> - in megahertz.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>
volumes	<i>collection</i> <i>[located Volume]</i>	<p>A reference to the list of references to <i>Volumes</i> that are connected to this <i>Machine</i>.</p> <p>Adding a <i>Volume</i> to this list means that the <i>Machine</i> has some access to the data on the <i>Volume</i>. Removing a <i>Volume</i> from this list means that the <i>Machine</i> no longer has access to the data on the <i>Volume</i>.</p> <p>Note: . This <i>Collection</i> has the semantics of usage of the <i>Volumes</i> by the <i>Machine</i> (deleting the <i>Machine</i> does not cause the deletion of the referred <i>Volumes</i>). It is defined in clause Error! Reference source not found.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-only</p>

Name	Machine	
Type URI	http://schemas.dmtf.org/cimi/2/Machine	
Attribute	Type	Description
interfaces	<i>collection</i> <i>[Network Interface]</i>	A reference to a list of references to <code>NetworkInterfaces</code> on this <code>Machine</code> . Each <code>NetworkInterface</code> Resource is a component of the <code>Machine</code> Resource. Each <code>NetworkInterface</code> instance represents an association between the <code>Machine</code> and a <code>Network</code> . <code>NetworkInterfaces</code> are defined in clause 5.16.13. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
latestSnapshot	<i>ref</i>	A reference to the <code>SNAPSHOT</code> representing the latest state captured for this <code>Machine</code> (either most recent <code>Snapshot</code> or the last <code>Snapshot</code> reverted to). 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 <code>MachineImages</code> of type <code>SNAPSHOT</code> taken of this <code>Machine</code> . This <code>Collection</code> has the semantics of usage of <code>SNAPSHOT MachineImages</code> by the <code>Machine</code> (The deletion of the <code>Machine</code> does not cause the deletion of the referred <code>Snapshots</code> .) Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meters	<i>collection</i> <i>[Meter]</i>	A reference to the list of <code>Meters</code> monitored for this <code>Machine</code> . Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLog	<i>ref</i>	A reference to the <code>EventLog</code> of this <code>Machine</code> . Constraints: Provider: support optional; mutable Consumer: support optional; read-only

2739 When implementing or using `Machine`, Providers and Consumers shall adhere to the syntax and
 2740 semantics of its attributes as described in Table 12, as well as in the tables describing embedded
 2741 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 2742 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 2743 both JSON and XML:

2744 **JSON media type:** application/json

2745 **JSON serialization:**

```

2746 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Machine",
2747   "id": string,
2748   "name": string, ?
2749   "description": string, ?
2750   "created": string, ?
2751   "updated": string, ?
2752   "parent": string, ?
2753   "properties": { string: string, + }, ?
2754   "vscope" : [ valueScope, * ], ?
2755   "state": string,
2756   "cpu": number,
2757   "memory": number,
```

```

2758     "disks" : { "href": string }, ?
2759     "cpuArch": string, ?
2760     "cpuSpeed": number, ?
2761     "volumes": { "href": string }, ?
2762     "interfaces": { "href": string }, ?
2763     "latestSnapshot": { "href": string }, ?
2764     "snapshots": { "href": string }, ?
2765     "meters": { "href": string }, ?
2766     "eventLog": { "href": string }, ?
2767     "operations": [
2768         { "rel": "edit", "href": string, ("available": boolean)? }, ?
2769         { "rel": "delete", "href": string, ("available": boolean)? }, ?
2770         { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string,
2771         ("available": boolean)? }, ?
2772         { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string,
2773         ("available": boolean)? }, ?
2774         { "rel": "http://schemas.dmtf.org/cimi/2/action/restart", "href": string,
2775         ("available": boolean)? }, ?
2776         { "rel": "http://schemas.dmtf.org/cimi/2/action/pause", "href": string,
2777         ("available": boolean)? }, ?
2778         { "rel": "http://schemas.dmtf.org/cimi/2/action/suspend", "href": string,
2779         ("available": boolean)? }, ?
2780         { "rel": "http://schemas.dmtf.org/cimi/2/action/snapshot", "href": string,
2781         ("available": boolean)? }, ?
2782         { "rel": "http://schemas.dmtf.org/cimi/2/action/restore", "href": string,
2783         ("available": boolean)? } ?
2784     ]
2785     ...
2786 }

```

2787 **XML media type:** application/xml

2788 **XML serialization:**

```

2789 <Machine xmlns="http://schemas.dmtf.org/cimi/2">
2790   <id> xs:anyURI </id>
2791   <name> xs:string </name> ?
2792   <description> xs:string </description> ?
2793   <created> xs:dateTime </created> ?
2794   <updated> xs:dateTime </updated> ?
2795   <parent> xs:anyURI </parent> ?
2796   <property key="xs:string"> xs:string </property> *
2797   <vscope> valueScope </vscope> *
2798   <state> xs:string </state>
2799   <cpu> xs:integer </cpu>

```

```

2800 <memory> xs:integer </memory>
2801 <disks href="xs:anyURI"/> ?
2802 <cpuArch> xs:string </cpuArch> ?
2803 <cpuSpeed> xs:integer </cpuSpeed> ?
2804 <volumes href="xs:anyURI"/> ?
2805 <interfaces href="xs:anyURI"/> ?
2806 <latestSnapshot href="xs:anyURI"/> ?
2807 <snapshots href="xs:anyURI"/> ?
2808 <meters href="xs:anyURI"/> ?
2809 <eventLog href="xs:anyURI"/> ?
2810 <operation rel="edit" href="xs:anyURI" (available="xs:boolean")? /> ?
2811 <operation rel="delete" href="xs:anyURI" (available="xs:boolean")? /> ?
2812 <operation rel="http://schemas.dmtf.org/cimi/2/action/start" href="xs:anyURI"
2813 (available="xs:boolean")? /> ?
2814 <operation rel="http://schemas.dmtf.org/cimi/2/action/stop" href="xs:anyURI"
2815 (available="xs:boolean")? /> ?
2816 <operation rel="http://schemas.dmtf.org/cimi/2/action/restart"
2817 href="xs:anyURI" (available="xs:boolean")? /> ?
2818 <operation rel="http://schemas.dmtf.org/cimi/2/action/pause" href="xs:anyURI"
2819 (available="xs:boolean")? /> ?
2820 <operation rel="http://schemas.dmtf.org/cimi/2/action/suspend"
2821 href="xs:anyURI" (available="xs:boolean")? /> ?
2822 <operation rel="http://schemas.dmtf.org/cimi/2/action/capture"
2823 href="xs:anyURI" (available="xs:boolean")? /> ?
2824 <operation rel="http://schemas.dmtf.org/cimi/2/action/snapshot"
2825 href="xs:anyURI" (available="xs:boolean")? /> ?
2826 <operation rel="http://schemas.dmtf.org/cimi/2/action/restore"
2827 href="xs:anyURI" (available="xs:boolean")? /> ?
2828 <xs:any>*
2829 </Machine>
    
```

2830 **5.14.1.1 Collections**

2831 The following clause describes the Collection Resources components of Machines.

2832 **5.14.1.1.1 Disk Collection**

2833 The Resource type for each item of this Collection is "Disk", defined in Table 13:

2834 **Table 13 – Disk attributes**

Name	Disk	
Type URI	http://schemas.dmtf.org/cimi/2/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
--	--	--

2835 In the following serializations, the Disk resource is expanded: each item of the Collection shows the Disk
2836 attributes, not a reference.

2837 **JSON serialization:**

```
2838 { "resourceURI": "http://schemas.dmtf.org/cimi/2/DiskCollection",
2839   "id": string,
2840   "count": number,
2841   "disks": [
2842     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Disk",
2843       "id": string,
2844       "name": string, ?
2845       "description": string, ?
2846       "created": string, ?
2847       "updated": string, ?
2848       "properties": { string: string, + }, ?
2849       "capacity": number,
2850       "initialLocation": string, ?
2851       "operations": [
2852         { "rel": "edit", "href": string }, ?
2853         { "rel": "delete", "href": string } ?
2854       ] ?
2855       ...
2856     }, +
2857   ], ?
2858   "operations": [ { "rel": "add", "href": string } ? ]
2859   ...
2860 }
```

2861 **XML serialization:**

```
2862 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/DiskCollection"
2863   xmlns="http://schemas.dmtf.org/cimi/2">
2864   <id> xs:anyURI </id>
2865   <count> xs:integer </count>
2866   <Disk>
2867     <id> xs:anyURI </id>
2868     <name> xs:string </name> ?
2869     <description> xs:string </description> ?
2870     <created> xs:dateTime </created> ?
```

```

2871 <updated> xs:dateTime </updated> ?
2872 <property key="xs:string"> xs:string </property> *
2873 <capacity> xs:integer </capacity>
2874 <initialLocation> xs:string </initialLocation> ?
2875 <operation rel="edit" href="xs:anyURI"/> ?
2876 <operation rel="delete" href="xs:anyURI"/> ?
2877 <xs:any>*
2878 </Disk> *
2879 <operation rel="add" href="xs:anyURI"/> ?
2880 <xs:any>*
2881 </Collection>
    
```

2882 **5.14.1.1.2 volumes Collection**

2883 The referred Resource type for each item of this Collection is “Volume”. However because there is an
 2884 accessory attribute (initialLocation), this is not a basic but an enhanced Volume Collection. The name
 2885 “locatedVolume” is used to define the type of each Collection item. The accessory attribute is defined in
 2886 Table 14:

2887 **Table 14 – locatedVolume accessory attributes**

Name	locatedVolume	
Type URI	http://schemas.dmtf.org/cimi/2/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

2888 **JSON serialization:**

```

2889 { "resourceURI": "http://schemas.dmtf.org/cimi/2/locatedVolumeCollection",
2890   "id": string,
2891   "updated": string,
2892   "parent": string,
2893   "count": number,
2894   "locatedVolumes": [
2895     { "resourceURI": "http://schemas.dmtf.org/cimi/2/locatedVolume",
2896       "id": string,
2897       "name": string, ?
2898       "description": string, ?
2899       "created": string, ?
2900       "updated": string, ?
2901       "parent": string, ?
2902       "properties": { string: string, + }, ?
    
```

```

2903     "initialLocation": string, ?
2904     "volume": { "href": string },
2905     "operations": [
2906         { "rel": "edit", "href": string }, ?
2907         { "rel": "delete", "href": string } ?
2908     ] ?
2909     ...
2910 }, +
2911 ], ?
2912 "operations": [
2913     { "rel": "add", "href": string } ?
2914     { "rel": "insert", "href": string } ?
2915     { "rel": "remove", "href": string } ?
2916 ]
2917 ...
2918 }

```

2919 XML serialization:

```

2920 <Collection
2921     resourceURI="http://schemas.dmtf.org/cimi/2/locatedVolumeCollection"
2922     xmlns="http://schemas.dmtf.org/cimi/2">
2923     <id> xs:anyURI </id>
2924     <updated> xs:dateTime </updated>
2925     <parent> xs:anyURI </parent>
2926     <count> xs:integer </count>
2927     <locatedVolume>
2928         <id> xs:anyURI </id>
2929         <name> xs:string </name> ?
2930         <description> xs:string </description> ?
2931         <created> xs:dateTime </created> ?
2932         <updated> xs:dateTime </updated> ?
2933         <parent> xs:anyURI </parent> ?
2934         <property key="xs:string"> xs:string </property> *
2935         <initialLocation> xs:string </initialLocation> ?
2936         <volume href="xs:anyURI" />
2937         <operation rel="edit" href="xs:anyURI" /> ?
2938         <operation rel="delete" href="xs:anyURI" /> ?
2939         <xs:any>*
2940     </locatedVolume> *
2941     <operation rel="add" href="xs:anyURI" /> ?
2942     <operation rel="insert" href="xs:anyURI" /> ?

```

```

2943     <operation rel="remove" href="xs:anyURI"/> ?
2944     <xs:any>*
2945 </Collection>
    
```

2946 **5.14.1.1.3 interfaces Collection**

2947 The Resource type for each item of this Collection is “NetworkInterface”, defined in clause 5.16.13.
 2948 The Collection is a basic NetworkInterfaceCollection as described in clause 5.16.14.

2949 **5.14.1.1.4 snapshots Collection**

2950 The Resource type for each item of this Collection is “MachineImage”. It is a basic MachineImage
 2951 Collection. Its serialization is described in the MachineImageCollection Resource clause.

2952 **5.14.1.1.5 meters Collection**

2953 The Resource type for each item of this Collection is “Meter” as defined in clause 5.17.3. There is no
 2954 accessory attribute for the items in this Collection, therefore it is a basic Meter Collection (serialized as
 2955 described in 5.5.12). See the MeterCollection Resource clause.

2956 **5.14.1.2 Operations**

2957 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 2958 MachineCollection Resource.

2959 The following custom operations are also defined:

2960 **start**

2961 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/start`

2962 This operation shall start a Machine.

2963 Input parameters: None.

2964 Output parameters: None.

2965 During the processing of this operation, the Machine shall be in the “STARTING” state.

2966 Upon successful completion of this operation, the Machine shall be in the "STARTED" state.

2967 If a Machine is in the "STOPPED" state, starting it shall be the virtual equivalent of powering on a
 2968 physical machine. There is no restored CPU or Memory state, so the guest OS typically performs boot or
 2969 installation tasks.

2970 If the Machine was in the "SUSPENDED" or "PAUSED" state, starting it shall have the effect of
 2971 resuming it.

2972 **HTTP protocol**

2973 To start a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/start" URI of the
 2974 Machine where the HTTP request body shall be as described below.

2975 **JSON media type:** application/json

2976 **JSON serialization:**

```

2977     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
    
```

```

2978     "action": "http://schemas.dmtf.org/cimi/2/action/start",
2979     "properties": { string: string, + } ?
2980     ...
2981 }

```

2982 **XML media type:** application/xml

2983 **XML serialization**

```

2984 <Action xmlns="http://schemas.dmtf.org/cimi/2">
2985   <action> http://schemas.dmtf.org/cimi/2/action/start </action>
2986   <property key="xs:string"> xs:string </property> *
2987   <xs:any>*
2988 </Action>

```

2989 Upon successful processing of the request, the HTTP response body may be empty.

2990 **stop**

2991 **/link@rel:** http://schemas.dmtf.org/cimi/2/action/stop

2992 This operation shall stop a `Machine`.

2993 Input parameters:

- 2994 1) "force" - type: boolean - optional
- 2995 2) A flag to indicate whether the Provider shall simulate a power off condition (force=true) or shall
- 2996 simulate a shutdown operation that allows applications to save their state and the file system to
- 2997 be made consistent (force=false). Inclusion of this parameter by Consumers is optional and if
- 2998 not specified, the Provider may choose either mechanism. Providers are encouraged to
- 2999 advertise this choice by the way of the `MachineStopForceDefault` capability.

3000 Output parameters: None.

3001 During the processing of this operation, the `Machine` shall be in the "STOPPING" state.

3002 Upon successful completion of this operation, the `Machine` shall be in the "STOPPED" state. Stopping a

3003 `Machine` with force=true shall be the virtual equivalent of powering off a physical machine. There is no

3004 saved CPU or Memory state. Stopping a `Machine` with force=false shall result in a machine with

3005 consistent file systems.

3006 A Consumer may reissue a stop operation if the state is STOPPING, perhaps with force=true, but

3007 Providers shall not issue a force=true stop operation on their own.

3008 **HTTP protocol**

3009 To stop a `Machine`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/stop" URI of the

3010 `Machine` where the HTTP request body shall be as described below.

3011 **JSON media type:** application/json

3012 **JSON serialization:**

```

3013 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
3014   "action": "http://schemas.dmtf.org/cimi/2/action/stop",
3015   "force": boolean, ?

```

```

3016     "properties": { string: string, + } ?
3017     ...
3018 }

```

3019 **XML media type:** application/xml

3020 **XML serialization**

```

3021 <Action xmlns="http://schemas.dmtf.org/cimi/2">
3022   <action> http://schemas.dmtf.org/cimi/2/action/stop </action>
3023   <force> xs:boolean </force> ?
3024   <property key="xs:string"> xs:string </property> *
3025   <xs:any>*
3026 </Action>

```

3027 Upon successful processing of the request, the HTTP response body may be empty.

3028 **restart**

3029 **/link@rel:** http://schemas.dmtf.org/cimi/2/action/restart

3030 This operation shall restart a *Machine*. If the *Machine* is in the "STARTED" state, this operation shall
 3031 have the effect of executing the "stop" and then "start" operations. If the *Machine* is in the "STOPPED"
 3032 state, this operation shall have the effect of executing the "start" operation.

3033 Input parameters:

- 3034 1) "force" - type: boolean - optional
- 3035 2) A flag to indicate whether the Provider shall simulate a power off condition (force=true) or shall
 3036 simulate a shutdown operation that allows applications to save their state and the file system to
 3037 be made consistent (force=false). Inclusion of this parameter by Consumers is optional and if
 3038 not specified, the Provider may choose either mechanism. Providers are encouraged to
 3039 advertise this choice by the way of the MachineStopForceDefault capability.

3040 Output parameters: None.

3041 During the processing of this operation, the *Machine* shall be in the "STOPPING" and/or "STARTING"
 3042 states, as appropriate depending on its initial state.

3043 Upon successful completion of this operation, the *Machine* shall be in the "STARTED" state. Restarting
 3044 a *Machine* shall be the virtual equivalent of powering off, and then powering on a physical machine.
 3045 There is no restored CPU or Memory state, so the guest OS typically performs boot or installation tasks.

3046 **HTTP protocol**

3047 To restart a *Machine*, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/restart" URI of the
 3048 *Machine* where the HTTP request body shall be as described below.

3049 **JSON media type:** application/json

3050 **JSON serialization:**

```

3051 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
3052   "action": "http://schemas.dmtf.org/cimi/2/action/restart",
3053   "force": boolean, ?
3054   "properties": { string: string, + } ?

```

3055 ...
 3056 }

3057 **XML media type:** application/xml

3058 **XML serialization**

```
3059     <Action xmlns="http://schemas.dmtf.org/cimi/2">
3060         <action> http://schemas.dmtf.org/cimi/2/action/restart </action>
3061         <force> xs:boolean </force> ?
3062         <property key="xs:string"> xs:string </property> *
3063         <xs:any>*
3064     </Action>
```

3065 Upon successful processing of the request, the HTTP response body may be empty.

3066 **pause**

3067 **/link@rel:** http://schemas.dmtf.org/cimi/2/action/pause

3068 This operation shall pause a Machine.

3069 Input parameters: None.

3070 Output parameters: None.

3071 During the processing of this operation, the Machine shall be in the "PAUSING" state.

3072 Upon successful completion of this operation, the Machine shall be in the "PAUSED" state. Pausing a
 3073 Machine shall keep the Machine and its resources instantiated, but the Machine shall not be
 3074 available to perform any tasks. The current state of the CPU and Memory shall be retained in volatile
 3075 memory.

3076 **HTTP protocol**

3077 To pause a Machine, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action.pause" URI of the
 3078 Machine where the HTTP request body shall be as described below.

3079 **JSON media type:** application/json

3080 **JSON serialization:**

```
3081     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
3082       "action": "http://schemas.dmtf.org/cimi/2/action/pause",
3083       "properties": { string: string, + } ?
3084       ...
3085     }
```

3086 **XML media type:** application/xml

3087 **XML serialization**

```
3088     <Action xmlns="http://schemas.dmtf.org/cimi/2">
3089         <action> http://schemas.dmtf.org/cimi/2/action/pause </action>
3090         <property key="xs:string"> xs:string </property> *
```

```
3091     <xs:any>*
3092 </Action>
```

3093 Upon successful processing of the request, the HTTP response body may be empty.

3094 **suspend**

3095 **/link@rel:** <http://schemas.dmtf.org/cimi/2/action/suspend>

3096 This operation shall suspend a *Machine*.

3097 Input parameters: None.

3098 Output parameters: None.

3099 During the processing of this operation, the *Machine* shall be in the "SUSPENDING" state.

3100 Upon successful completion of this operation, the *Machine* shall be in the "SUSPENDED" state.
 3101 Suspending a *Machine* shall keep the *Machine* and its resources instantiated, but the *Machine* shall
 3102 not be available to perform any tasks. The current state of the CPU and Memory shall be retained in
 3103 non-volatile memory.

3104 **HTTP protocol**

3105 To suspend a *Machine*, a POST is sent to the "<http://schemas.dmtf.org/cimi/2/action/suspend>" URI of
 3106 the *Machine* where the HTTP request body shall be as described below.

3107 **JSON media type:** application/json

3108 **JSON serialization:**

```
3109     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
3110       "action": "http://schemas.dmtf.org/cimi/2/action/suspend",
3111       "properties": { string: string, + } ?
3112       ...
3113     }
```

3114 **XML media type:** application/xml

3115 **XML serialization**

```
3116     <Action xmlns="http://schemas.dmtf.org/cimi/2">
3117       <action> http://schemas.dmtf.org/cimi/2/action/suspend </action>
3118       <property key="xs:string"> xs:string </property> *
3119       <xs:any>*
3120     </Action>
```

3121 Upon successful processing of the request, the HTTP response body may be empty.

3122 **capture**

3123 **/link@rel:** <http://schemas.dmtf.org/cimi/2/action/capture>

3124 This operation shall create a new *MachineImage* from an existing *Machine*. This operation is defined
 3125 within the *MachineImage* Resource; see 5.14.7.1 for more details. Note that while this operation is

3126 performed against a `MachineImage`, its presence in the `Machine` serialization is used to advertise
3127 support for the operation.

3128 **Snapshotting a Machine**

3129 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/snapshot`

3130 This operation shall create a new `SNAPSHOT MachineImage` from an existing `Machine`. This
3131 operation is defined within the `MachineImage Resource`; see 5.14.7.1 for more details. Note that while
3132 this operation is performed against a `MachineImage`, its presence in the `Machine` serialization is
3133 used to advertise support for the operation.

3134 **Restoring a Machine**

3135 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/restore`

3136 This operation shall restore a `Machine` from a previously created `MachineImage`.

3137 Input parameters:

- 3138 1) "image" - type: URI - mandatory
- 3139 2) A reference to the Machine Image.

3140 Output parameters: None.

3141 During the processing of this operation, the `Machine` shall be in the "RESTORING" state.

3142 Upon successful completion of this operation, the `Machine` shall be in the same state as the state
3143 specified in the `MachineImage`, if specified. See 5.14.2.1 for more details.

3144 Note that Providers can indicate support for restoring from non-`SNAPSHOT MachineImages` by the
3145 way of the `Machine "RestoreFromImage"` capability. If the `RestoreFromImage` capability is not supported,
3146 and the restore operation is supported, the restore operation can only restore from a `SNAPSHOT`
3147 `MachineImage`.

3148 **HTTP protocol**

3149 To restore a `Machine`, a POST is sent to the "`http://schemas.dmtf.org/cimi/2/action/restore`" URI of the
3150 `Machine` where the HTTP request body shall be as described below.

3151 **JSON media type:** `application/json`

3152 **JSON serialization:**

```
3153 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
3154   "action": "http://schemas.dmtf.org/cimi/2/action/restore",
3155   "image": { "href": string },
3156   "properties": { string: string, + } ?
3157   ...
3158 }
```

3159 **XML media type:** `application/xml`

3160 **XML serialization**

```
3161 <Action xmlns="http://schemas.dmtf.org/cimi/2">
3162   <action> http://schemas.dmtf.org/cimi/2/action/restore </action>
```

```

3163 <image href="xs:anyURI"/>
3164 <property key="xs:string"> xs:string </property> *
3165 <xs:any>*
3166 </Action>

```

3167 Where the "image" URI is a reference to the MachineImage to be used.

3168 Upon successful processing of the request, the HTTP response body may be empty.

3169 5.14.2 MachineCollection Resource

3170 A MachineCollection Resource represents the Collection of Machine Resources within a
 3171 Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as
 3172 follows:

3173 JSON serialization:

```

3174 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineCollection",
3175   "id": string,
3176   "updated": string,
3177   "parent": string,
3178   "count": number,
3179   "machines": [
3180     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Machine",
3181       "id": string,
3182       "name": string, ?
3183       "description": string, ?
3184       "created": string, ?
3185       "updated": string, ?
3186       "parent": string, ?
3187       "properties": { string: string, + }, ?
3188       "machine": { "href": string },
3189       "operations": [
3190         { "rel": "edit", "href": string }, ?
3191         { "rel": "delete", "href": string } ?
3192       ] ?
3193     }, +
3194   ], ?
3195   "operations": [
3196     { "rel": "add", "href": string }, ?
3197     { "rel": "insert", "href": string }, ?
3198     { "rel": "remove", "href": string } ?
3199   ]
3200   ...
3201 }

```

3202 **XML serialization:**

```

3203 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/MachineCollection"
3204     xmlns="http://schemas.dmtf.org/cimi/2">
3205     <id> xs:anyURI </id>
3206     <updated> xs:dateTime </updated>
3207     <parent> xs:anyURI </parent>
3208     <count> xs:integer </count>
3209     <Machine>
3210         <id> xs:anyURI </id>
3211         <name> xs:string </name> ?
3212         <description> xs:string </description> ?
3213         <created> xs:dateTime </created> ?
3214         <updated> xs:dateTime </updated> ?
3215         <parent> xs:anyURI </parent> ?
3216         <property key="xs:string"> xs:string </property> *
3217         <machine href="xs:anyURI"/>
3218         <operation rel="edit" href="xs:anyURI"/> ?
3219         <operation rel="delete" href="xs:anyURI"/> ?
3220         <xs:any>*
3221     </Machine> *
3222     <operation rel="add" href="xs:anyURI"/> ?
3223     <operation rel="insert" href="xs:anyURI"/> ?
3224     <operation rel="remove" href="xs:anyURI"/> ?
3225     <xs:any>*
3226 </Collection>

```

3227 **5.14.2.1 Operations**

3228 **NOTE** The "add" operation requires that a MachineTemplate be used (see 4.2.1.1).

3229 Upon successful processing of the "add" operation, unless otherwise specified by the way of the
3230 MachineTemplate "initialState" attribute, the state of the new Machine shall be the value of the
3231 DefaultInitialState capability, if defined. If no DefaultInitialState capability is defined, the default value shall
3232 be "STOPPED." The semantics of "initialState" shall be equivalent to the Provider issuing the appropriate
3233 actions against the new Machine to move it into that state. Note that this controls the actions of the
3234 hypervisor and the state of the resources within the Machine (e.g., the operating system) are also
3235 influenced by the data within the MachineImage used to create the new Machine. For example, if a
3236 new Machine's initialState is "STARTED" and a SNAPSHOT MachineImage was used to create the
3237 new Machine, the Machine would not be "booted" but rather resume executing from the saved state in
3238 the MachineImage.

3239 If a Provider is unable to change the state of the new Machine to the appropriate "initialState" (either as
3240 specified by the MachineTemplate or as implied by the previous stated rules), the Machine creation
3241 shall fail.

3242 If a Provider is unable to create the new `Machine` due to invalid or inconsistent credentials in the
 3243 `MachineTemplate`, the `Machine` creation process shall fail. If any credentials are included in the
 3244 `MachineTemplate`, they shall be part of the new `Machine` regardless of the type of
 3245 `MachineImage` used.

3246 **5.14.3 MachineTemplate**

3247 A `MachineTemplate` represents the set of metadata and instructions used in the creation of a
 3248 `Machine`. Table 15 describes the `MachineTemplate` attributes.

3249 **Table 15 – MachineTemplate attributes**

Name		MachineTemplate
Type URI		http://schemas.dmtf.org/cimi/2/MachineTemplate
Attribute	Type	Description
initialState	string	The initial state of the new <code>Machine</code> . Possible values include the non-transient states as specified by the <code>Machine</code> "state" attribute (e.g., <code>STARTED</code> , <code>STOPPED</code>) and are determined by the actions supported by the Provider. Providers should advertise the list of available values through the <code>Machine</code> 's "initialStates" capability. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
machineConfig	ref	A reference to the <code>MachineConfiguration</code> that is used to create a <code>Machine</code> from this <code>MachineTemplate</code> . Note that the attributes of the <code>MachineConfiguration</code> may be specified rather than a reference to an existing <code>MachineConfiguration</code> Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
machineImage	ref	A reference to the <code>MachineImage</code> that is used to create a <code>Machine</code> from this <code>MachineTemplate</code> . Constraints: Provider: support optional; mutable Consumer: support optional; read-write
credential	ref	A reference to the <code>Credential</code> that is used to create the initial login credentials for the new <code>Machine</code> . Note that the attributes of the <code>Credential</code> may be specified rather than a reference to an existing <code>Credential</code> Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

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

3250 When implementing or using *MachineTemplate*, Providers and Consumers shall adhere to the syntax
 3251 and semantics of its attributes as described in Table 15, as well as in the tables describing embedded
 3252 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 3253 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 3254 both JSON and XML:

3255 **JSON media type:** application/json

3256 **JSON serialization:**

```
3257 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineTemplate",
3258   "id": string,
3259   "name": string, ?
3260   "description": string, ?
3261   "created": string, ?
3262   "updated": string, ?
3263   "properties": { string: string, + }, ?
3264   "vscope" : [ valueScope, * ], ?
3265   "initialState": string, ?
```

```

3266 "machineConfig": {
3267     "href": string | ... MachineConfiguration attributes ...
3268 }, ?
3269 "machineImage": {
3270     "href": string | ... MachineImage attributes ...
3271 }, ?
3272 "credential": {
3273     "href": string | ... CredentialTemplate attributes ...
3274 }, ?
3275 "volumes": [
3276     { "initialLocation": string?, "href": string }, +
3277 ], ?
3278 "volumeTemplates": [
3279     { "initialLocation": string?,
3280       "href": string, ?
3281       ... VolumeTemplate attributes ... ?
3282     }, +
3283 ], ?
3284 "interfaceTemplates": [
3285     { "href": string, ?
3286       ... NetworkInterfaceTemplate attributes ... ?
3287     }, *
3288 ], ?
3289 "userData": string, ?
3290 "meterTemplates": [
3291     { "href": string, ?
3292       ... MeterTemplate attributes ... ?
3293     }, *
3294 ], ?
3295 "eventLogTemplate": {
3296     "href": string, ?
3297     ... EventLogTemplate attributes ... ?
3298 }, ?
3299 "operations": [
3300     { "rel": "edit", "href": string }, ?
3301     { "rel": "delete", "href": string } ?
3302 ] ?
3303 ...
3304 }

```

3305 **XML media type:** application/xml

3306 **XML serialization:**

```

3307 <MachineTemplate xmlns="http://schemas.dmtf.org/cimi/2">
3308   <id> xs:anyURI </id>
3309   <name> xs:string </name> ?
3310   <description> xs:string </description> ?
3311   <created> xs:dateTime </created> ?
3312   <updated> xs:dateTime </updated> ?
3313   <property key="xs:string"> xs:string </property> *
3314   <vscope> valueScope </vscope> *
3315   <initialState> xs:string </initialState> ?
3316   <machineConfig href="xs:anyURI"?>
3317     ... MachineConfiguration attributes ... ?
3318   </machineConfig> ?
3319   <machineImage href="xs:anyURI"?>
3320     ... MachineImage attributes ... ?
3321   </machineImage> ?
3322   <credential href="xs:anyURI"?>
3323     ... CredentialTemplate attributes ... ?
3324   </credential> ?
3325   <volume initialLocation="xs:string"? href="xs:anyURI" /> *
3326   <volumeTemplate initialLocation="xs:string"? href="xs:anyURI"? >
3327     ... VolumeTemplate attributes ... ?
3328   </volumeTemplate> *
3329   <interfaceTemplate href="xs:anyURI"? >
3330     ... NetworkInterfaceTemplate attributes ... ?
3331   </interfaceTemplate> *
3332   <userData> xs:string </userData> ?
3333   <meterTemplate href="xs:anyURI"? >
3334     ... MeterTemplate attributes ... ?
3335   </meterTemplate> *
3336   <eventLogTemplate href="xs:anyURI"? >
3337     ... EventLogTemplate attributes ... ?
3338   </eventLogTemplate> ?
3339   <operation rel="edit" href="xs:anyURI"/> ?
3340   <operation rel="delete" href="xs:anyURI"/> ?
3341   <xs:any>*
3342 </MachineTemplate>

```

3343 Injection of user-defined data

3344 To simplify the customization of individual `Machines`, it is possible to pass arbitrary data into the new
 3345 `Machine` by using the `userData` parameter. The value of this parameter shall be the Base64-encoded
 3346 payload. The Provider shall arrange for this data to be available from inside the `Machine` by using one
 3347 of the following methods:

- 3348 1. *Metadata server*: The data can be retrieved from within the instance by using an HTTP GET
 3349 request to `http://169.254.169.254/cimi/latest/user-data`.
- 3350 2. *Disk*: The `Machine` has access to a `Disk` with an ISO 9660 file system on it. The data can be
 3351 found in a file at `<location>/cimi/user-data`.
- 3352 3. *Image modification*: The Provider modifies the root file system of the machine image just before
 3353 launching the `Machine`. In UNIX-like operating systems, the data can be found in the file
 3354 `/var/lib/cimi/user-data`.

3355 It is strongly recommended that Providers implement a `metadata server`, or, failing that, injection by
 3356 the way of `Disk`, as `image modification` is brittle and may not work for every operating system in
 3357 use. The Provider shall indicate which of these three methods is supported with the `Machine` 'UserData'
 3358 capability in the `ResourceMetadata` for `Machines`. The value for this feature shall be one of
 3359 `metadata`, `disk`, or `imgmod`, corresponding to the three methods listed above.

3360 The Provider shall preserve this data across restarts of the `Machine`. The data is the Base64-decoded
 3361 version of the data that was passed into the `MachineCreate` request.

3362 5.14.3.1 Operations

3363 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 3364 `MachineTemplateCollection` Resource.

3365 5.14.4 MachineTemplateCollection Resource

3366 A `MachineTemplateCollection` Resource represents the Collection of `MachineTemplate`
 3367 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
 3368 shall be serialized as follows:

3369 JSON serialization:

```

3370 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineTemplateCollection",
3371   "id": string,
3372   "count": number,
3373   "machineTemplates": [
3374     { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineTemplate",
3375       "id": string,
3376       ... remaining MachineTemplate attributes ...
3377     }, +
3378   ], ?
3379   "operations": [ { "rel": "add", "href": string } ? ]
3380   ...
3381 }
```

3382 **XML serialization:**

```

3383 <Collection
3384     resourceURI="http://schemas.dmtf.org/cimi/2/MachineTemplateCollection"
3385     xmlns="http://schemas.dmtf.org/cimi/2">
3386     <id> xs:anyURI </id>
3387     <count> xs:integer </count>
3388     <MachineTemplate>
3389         <id> xs:anyURI </id>
3390         ... remaining MachineTemplate attributes ...
3391     </MachineTemplate> *
3392     <operation rel="add" href="xs:anyURI"/> ?
3393     <xs:any>*
3394 </Collection>
    
```

3395 **5.14.4.1 Operations**

3396 This Resource supports the Read and Update operations. Creation of new MachineTemplate
 3397 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 3398 4.2.1.1.

3399 **5.14.5 MachineConfiguration Resource**

3400 The MachineConfiguration Resource represents the set of configuration values that define the
 3401 (virtual) hardware resources of a to-be-realized Machine Instance. MachineConfigurations are
 3402 created by Providers and may, at the Providers discretion, be created by Consumers.

3403 Table 16 describes the MachineConfiguration attributes.

3404 **Table 16 – MachineConfiguration attributes**

Name	MachineConfiguration													
Type URI	http://schemas.dmtf.org/cimi/2/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;"> <tr> <td>Name</td> <td colspan="2">disk</td> </tr> <tr> <td>Attribute</td> <td>Type</td> <td>Description</td> </tr> <tr> <td>capacity</td> <td>integer</td> <td>The initial capacity, in kilobytes, of the disk described by this attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td> </tr> <tr> <td>format</td> <td>string</td> <td>The format/type of this disk (e.g., ext4, NTFS).</td> </tr> </table>	Name	disk		Attribute	Type	Description	capacity	integer	The initial capacity, in kilobytes, of the disk described by this attribute. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write	format	string	The format/type of this disk (e.g., ext4, NTFS).
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).												

Name	MachineConfiguration		
Type URI	http://schemas.dmtf.org/cimi/2/MachineConfiguration		
Attribute	Type	Description	
			<p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>
		initialLocation	<p><i>string</i></p> <p>An Operating System-specific location (path) in its namespace where this <i>Disk</i> first appears. After creation of a <i>Machine</i>, Consumers may change the location of this <i>Disk</i>.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>
			<p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>
cpuArch	string	<p>The CPU architecture that is supported by <i>Machines</i> created by using this configuration. Allowed values are: 68000, Alpha, ARM, Itanium, MIPS, PA_RISC, POWER, PowerPC, x86, x86_64, z/Architecture, SPARC. Providers may define additional values.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>	
cpuSpeed	integer	<p>The approximate CPU speed of this <i>Machine</i> in megahertz.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>	

3405 NOTE The disk attributes "format" does not appear on Machine Resources because after the *Machine* is
 3406 created, the user of the *Machine* is able modify this attribute of a disk, possibly without the Provider's knowledge.
 3407 Therefore these attributes might not be an aspect of the *Machine* that the Provider can reliably manage.

3408 **JSON media type:** application/json

3409 **JSON serialization:**

```

3410 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineConfiguration",
3411   "id": string,
3412   "name": string, ?
3413   "description": string, ?
3414   "created": string, ?
3415   "updated": string, ?
3416   "properties": { string: string, + }, ?
3417   "vscope" : [ valueScope, * ], ?
3418   "cpu": number,
3419   "memory": number,
3420   "disks" : [
3421     { "capacity": number,
3422       "format": string,
3423       "initialLocation": string?
3424     }, +
3425   ], ?
3426   "cpuArch": string, ?
    
```

```

3427 "cpuSpeed": number, ?
3428 "operations": [
3429   { "rel": "edit", "href": string }, ?
3430   { "rel": "delete", "href": string } ?
3431 ] ?
3432 ...
3433 }

```

3434 **XML media type:** application/xml

3435 **XML serialization:**

```

3436 <MachineConfiguration xmlns="http://schemas.dmtf.org/cimi/2">
3437   <id> xs:anyURI </id>
3438   <name> xs:string </name> ?
3439   <description> xs:string </description> ?
3440   <created> xs:dateTime </created> ?
3441   <updated> xs:dateTime </updated> ?
3442   <property key="xs:string"> xs:string </property> *
3443   <vscope> valueScope </vscope> *
3444   <cpu> xs:integer </cpu>
3445   <memory> xs:integer </memory>
3446   <disk>
3447     <capacity> xs:integer </capacity>
3448     <format> xs:string </format>
3449     <initialLocation> xs:string </initialLocation> ?
3450   </disk> *
3451   <cpuArch> xs:string </cpuArch> ?
3452   <cpuSpeed> xs:integer </cpuSpeed> ?
3453   <operation rel="edit" href="xs:anyURI"/> ?
3454   <operation rel="delete" href="xs:anyURI"/> ?
3455   <xs:any>*
3456 </MachineConfiguration>

```

3457 5.14.5.1 Operations

3458 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 3459 MachineConfigurationCollection Resource.

3460 5.14.6 MachineConfigurationCollection Resource

3461 A MachineConfigurationCollection Resource represents the Collection of
 3462 MachineConfiguration Resources within a Provider and follows the Collection pattern defined in
 3463 clause 5.5.12. This Resource shall be serialized as follows:

3464 **JSON serialization:**

```

3465 { "resourceURI":
3466     "http://schemas.dmtf.org/cimi/2/MachineConfigurationCollection",
3467     "id": string,
3468     "count": number,
3469     "machineConfigurations": [
3470         { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineConfiguration",
3471           "id": string,
3472           ... remaining MachineConfiguration attributes ...
3473         }, +
3474     ], ?
3475     "operations": [ { "rel": "add", "href": string } ? ]
3476     ...
3477 }

```

3478 **XML serialization:**

```

3479 <Collection
3480     resourceURI="http://schemas.dmtf.org/cimi/2/MachineConfigurationCollection"
3481     xmlns="http://schemas.dmtf.org/cimi/2">
3482     <id> xs:anyURI </id>
3483     <count> xs:integer </count>
3484     <MachineConfiguration>
3485         <id> xs:anyURI </id>
3486         ... remaining MachineConfiguration attributes ...
3487     </MachineConfiguration> *
3488     <operation rel="add" href="xs:anyURI"/> ?
3489     <xs:any>*
3490 </Collection>

```

3491 **5.14.6.1 Operations**

3492 This Resource supports the Read and Update operations. Creation of new `MachineConfiguration`
3493 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
3494 4.2.1.1.

3495 **5.14.7 MachineImage Resource**

3496 This Resource represents the information necessary for hardware virtualized Resources to create a
3497 `Machine` Instance; it contains configuration data such as startup instructions, including possible
3498 combinations of the following items, depending on the "type" of `MachineImage` created:

- 3499 • The software image (i.e., a copy of an installed `Machine`), that is to be instantiated on the disk
3500 and other virtual resources. The image can be a snapshot that consists of disk images plus
3501 memory and other resource state information.
- 3502 • Installation software, which, when executed on the hardware (virtual) resources, builds the
3503 machine instance.

- 3504 • Both a disk image and a set of software and parameters to install new components not included
 3505 in the original disk image.

3506 Table 17 describes the MachineImage attributes.

3507 **Table 17 – MachineImage attributes**

Name	MachineImage	
Type URI	http://schemas.dmtf.org/cimi/2/MachineImage	
Attribute	Type	Description
state	string	<p>The operational state of the MachineImage. Allowed values are: CREATING: The MachineImage is in the process of being created. AVAILABLE: The MachineImage is available and ready for use. Unless otherwise specified, the MachineImage shall initially be in this state after successful creation. DELETING: The MachineImage is in the process of being deleted. ERROR: The Provider has detected an error in the MachineImage. The operations that result in transitions to the above defined states are defined in clause 5.14.7.1 Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</p>
type	string	<p>The type of MachineImage that is represented by this Resource. This specification defines the following values: IMAGE: This type represents the persisted data of a stopped Machine. Unlike "snapshots", it does not contain any runtime information. If this value is used, the "relatedImage" attribute shall not be present. SNAPSHOT: This type represents the persisted data of a Machine. If the Machine was not in a stopped state 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. PARTIAL_SNAPSHOT: This type follows the same semantics as the "SNAPSHOT" MachineImage except that it contains just the changes (deltas) made to the Machine based on the referenced "relatedImage" MachineImage rather than a complete representation of the Machine. If a MachineImage is deleted, the following semantics shall apply:</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. 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. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</p>

3508 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

3509 **JSON media type:** application/json

3510 **JSON serialization:**

```

3511 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImage",
3512     "id": string,
3513     "name": string, ?
3514     "description": string, ?
3515     "created": string, ?
3516     "updated": string, ?
3517     "properties": { string: string, + }, ?
3518     "state": string,
3519     "type": string,
3520     "imageLocation": string,
3521     "relatedImage": { "href": string }, ?
3522     "operations": [
3523         { "rel": "edit", "href": string }, ?
3524         { "rel": "delete", "href": string } ?
3525     ] ?
3526     ...
3527 }
```

3528 **XML media type:** application/xml

3529 **XML serialization:**

```

3530 <MachineImage xmlns="http://schemas.dmtf.org/cimi/2">
3531     <id> xs:anyURI </id>
3532     <name> xs:string </name> ?
3533     <description> xs:string </description> ?
3534     <created> xs:dateTime </created> ?
3535     <updated> xs:dateTime </updated> ?
3536     <property key="xs:string"> xs:string </property> *
3537     <state> xs:string </state>
3538     <type> xs:string </type>
3539     <imageLocation> xs:anyURI </imageLocation>
3540     <relatedImage href="xs:anyURI"/> ?
3541     <operation rel="edit" href="xs:anyURI"/> ?
3542     <operation rel="delete" href="xs:anyURI"/> ?
3543     <xs:any>*
3544 </MachineImage>
```

3545 **5.14.7.1 Operations**

3546 This Resource supports the Read, Update, and Delete operations. Create is supported through the
3547 `MachineImageCollection` Resource.

3548 If creating a new `MachineImage`, the representation of the new `MachineImage` may include a
3549 reference in the "imageLocation" attribute. Providers shall inspect this reference (most likely by the way of
3550 an HTTP HEAD) to determine if any special processing is required. This specification defines the
3551 following additional steps that Providers shall take depending on the type of Resource being referenced:

3552 `http://schemas.dmtf.org/cimi/2/Machine`

3553 If the "imageLocation" is a reference to a `Machine`, the Provider shall create a new `MachineImage`
3554 based on the `Machine` being referenced. The machine is captured or snapshotted, depending on
3555 whether the request was sent to the "<http://schemas.dmtf.org/cimi/2/action/capture>" or the
3556 "<http://schemas.dmtf.org/cimi/2/action/snapshot>" URI of the `Machine`. However the resulting resource,
3557 although linked to the `Machine` from which it was originated, shall be a `MachineImage` for all purposes
3558 and can be used for creating new machines.

3559 If creating a SNAPSHOT and upon completion of the create operation, the `MachineImage`'s
3560 "imageLocation" attribute shall not reference the `Machine` (as the `Machine` might change over time),
3561 but instead it shall reference (or contain the data of) the static representation of the `Machine`.
3562 Additionally, the referenced `Machine`'s `MachineSnapshotCollection` shall be updated to
3563 include a reference to this newly created SNAPSHOT `MachineImage` Resource. If the `Machine` is
3564 unable to accept operations at any point while it is being captured to create the `MachineImage`, the
3565 `Machine` shall be in state "CAPTURING".

3566 **5.14.8 MachineImageCollection Resource**

3567 A `MachineImageCollection` Resource represents the Collection of `MachineImage` Resources
3568 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
3569 serialized as follows:

3570 **JSON serialization:**

```
3571 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImageCollection",
3572   "id": string,
3573   "count": number,
3574   "machineImages": [
3575     { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImage",
3576       "id": string,
3577       ... remaining MachineImage attributes ...
3578     }, +
3579   ], ?
3580   "operations": [ { "rel": "add", "href": string } ? ]
3581   ...
3582 }
```

3583 **XML serialization:**

```
3584 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/MachineImageCollection"
3585   xmlns="http://schemas.dmtf.org/cimi/2">
```

```

3586 <id> xs:anyURI </id>
3587 <count> xs:integer </count>
3588 <MachineImage>
3589   <id> xs:anyURI </id>
3590   ... remaining MachineImage attributes ...
3591 </MachineImage> *
3592 <operation rel="add" href="xs:anyURI"/> ?
3593 <xs:any>*
3594 </Collection>
    
```

3595 **5.14.8.1 Operations**

3596 This Resource supports the Read and Update operations. Creation of new MachineImage Resources
 3597 is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1, where the
 3598 request body and the way it is processed are described in clause 5.14.7.1.

3599 **5.14.9 Credential Resource**

3600 A Credential Resource contains the information required to create the initial administrative superuser
 3601 of a newly created Machine or to represent the credentials needed to perform some operation. Due to
 3602 the variation between operating systems and Providers, this specification does not mandate one
 3603 particular set of attributes that all implementations need to support. However, Providers are expected to
 3604 extend this Resource with additional attributes to meet their requirements.

3605 For example, a Provider might extend this Resource with username and password attributes, which would
 3606 then be the login information for new Machines. These extension attributes would appear as siblings to
 3607 the common attributes like "name" and "description."

3608 Table 18 describes the Credential attributes.

3609 **Table 18 – Credential attributes**

Name	Credential	
Type URI	http://schemas.dmtf.org/cimi/2/Credential	
Attribute	Type	Description
TBD		The exact set of attributes is determined by the Provider.

3610 Some common extension attributes that Providers might use include:

3611 **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

3612 **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

3613 When implementing or using `Credential`, Providers and Consumers shall adhere to the syntax and
 3614 semantics of its attributes as described in the above table, as well as in the table describing related
 3615 Collections. Both Consumer and Provider shall serialize this Resource as described below. The following
 3616 pseudo-schemas (see notation in 1.3)

3617 **JSON media type:** application/json

3618 **JSON serialization:**

```
3619 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Credential",
3620   "id": string,
3621   "name": string, ?
3622   "description": string, ?
3623   "created": string, ?
3624   "updated": string, ?
3625   "properties": { string: string, + }, ?
3626   "operations": [
3627     { "rel": "edit", "href": string }, ?
3628     { "rel": "delete", "href": string } ?
3629   ] ?
3630   ...
3631 }
```

3632 **XML media type:** application/xml

3633 **XML serialization:**

```
3634 <Credential xmlns="http://schemas.dmtf.org/cimi/2">
3635   <id> xs:anyURI </id>
3636   <name> xs:string </name> ?
3637   <description> xs:string </description> ?
3638   <created> xs:dateTime </created> ?
3639   <updated> xs:dateTime </updated> ?
3640   <property key="xs:string"> xs:string </property> *
3641   <operation rel="edit" href="xs:anyURI"/> ?
3642   <operation rel="delete" href="xs:anyURI"/> ?
3643   <xs:any>*
3644 </Credential>
```

3645 5.14.9.1 Operations

3646 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 3647 `CredentialCollection` Resource.

3648 5.14.10 CredentialCollection Resource

3649 A `CredentialCollection` Resource represents the Collection of `Credential` Resources within
 3650 a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized
 3651 as follows:

3652 JSON serialization:

```
3653 { "resourceURI": "http://schemas.dmtf.org/cimi/2/CredentialCollection",
3654   "id": string,
3655   "count": number,
3656   "credentials": [
3657     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Credential",
3658       "id": string,
3659       ... remaining Credential attributes ...
3660     }, +
3661   ], ?
3662   "operations": [ { "rel": "add", "href": string } ? ]
3663   ...
3664 }
```

3665 XML serialization:

```
3666 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/CredentialCollection"
3667   xmlns="http://schemas.dmtf.org/cimi/2">
3668   <id> xs:anyURI </id>
3669   <count> xs:integer </count>
3670   <Credential>
3671     <id> xs:anyURI </id>
3672     ... remaining Credential attributes ...
3673   </Credential> *
3674   <operation rel="add" href="xs:anyURI"/> ?
3675   <xs:any>*
3676 </Collection>
```

3677 5.14.10.1 Operations

3678 NOTE The "add" operation requires that a `CredentialTemplate` be used (see 4.2.1.1).

3679 5.14.11 CredentialTemplate Resource

3680 This Resource captures the configuration values for realizing a `Credential` Resource. A
 3681 `CredentialTemplate` may be used to create multiple `Credentials`. Table 21 describes the
 3682 `CredentialTemplate` attributes.

3683

Table 21 – CredentialTemplate attributes

Name	CredentialTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/CredentialTemplate	
Attribute	Type	Description
<i>TBD</i>		The exact set of attributes is determined by the provider.

3684 When implementing or using `CredentialTemplate`, Providers and Consumers shall adhere to the
 3685 syntax and semantics of its attributes as described in Table 21 as well as in the table describing related
 3686 Collections. Both Consumer and Provider shall serialize this Resource as described below. The following
 3687 pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML:

3688 **JSON media type:** application/json

3689 **JSON serialization:**

```

3690 { "resourceURI": "http://schemas.dmtf.org/cimi/2/CredentialTemplate",
3691   "id": string,
3692   "name": string, ?
3693   "description": string, ?
3694   "created": string, ?
3695   "updated": string, ?
3696   "properties": { string: string, + }, ?
3697   "operations": [
3698     { "rel": "edit", "href": string }, ?
3699     { "rel": "delete", "href": string } ?
3700   ] ?
3701   ...
3702 }
```

3703 **XML media type:** application/xml

3704 **XML serialization:**

```

3705 <CredentialTemplate xmlns="http://schemas.dmtf.org/cimi/2">
3706   <id> xs:anyURI </id>
3707   <name> xs:string </name> ?
3708   <description> xs:string </description> ?
3709   <created> xs:dateTime </created> ?
3710   <updated> xs:dateTime </updated> ?
3711   <property key="xs:string"> xs:string </property> *
3712   <operation rel="edit" href="xs:anyURI"/> ?
3713   <operation rel="delete" href="xs:anyURI"/> ?
3714   <xs:any*>
3715 </CredentialTemplate>
```

3716 **5.14.11.1 Operations**

3717 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 3718 `CredentialTemplateCollection` Resource.

3719 5.14.12 CredentialTemplateCollection Resource

3720 A CredentialTemplateCollection Resource represents the Collection of
 3721 CredentialTemplate Resources within a Provider and follows the Collection pattern defined in
 3722 clause 5.5.12. This Resource shall be serialized as follows:

3723 JSON serialization:

```
3724 { "resourceURI":
3725     "http://schemas.dmtf.org/cimi/2/CredentialTemplateCollection",
3726     "id": string,
3727     "count": number,
3728     "credentialTemplates": [
3729         { "resourceURI": "http://schemas.dmtf.org/cimi/2/CredentialTemplate",
3730           "id": string,
3731           ... remaining CredentialTemplate attributes ...
3732         }, +
3733     ], ?
3734     "operations": [ { "rel": "add", "href": string }? ]
3735     ...
3736 }
```

3737 XML serialization:

```
3738 <Collection
3739     resourceURI="http://schemas.dmtf.org/cimi/2/CredentialTemplateCollection"
3740     xmlns="http://schemas.dmtf.org/cimi/2">
3741     <id> xs:anyURI </id>
3742     <count> xs:integer </count>
3743     <CredentialTemplate>
3744         <id> xs:anyURI </id>
3745         ... remaining CredentialTemplate attributes ...
3746     </CredentialTemplate> *
3747     <operation rel="add" href="xs:anyURI"/> ?
3748     <xs:any>*
3749 </Collection>
```

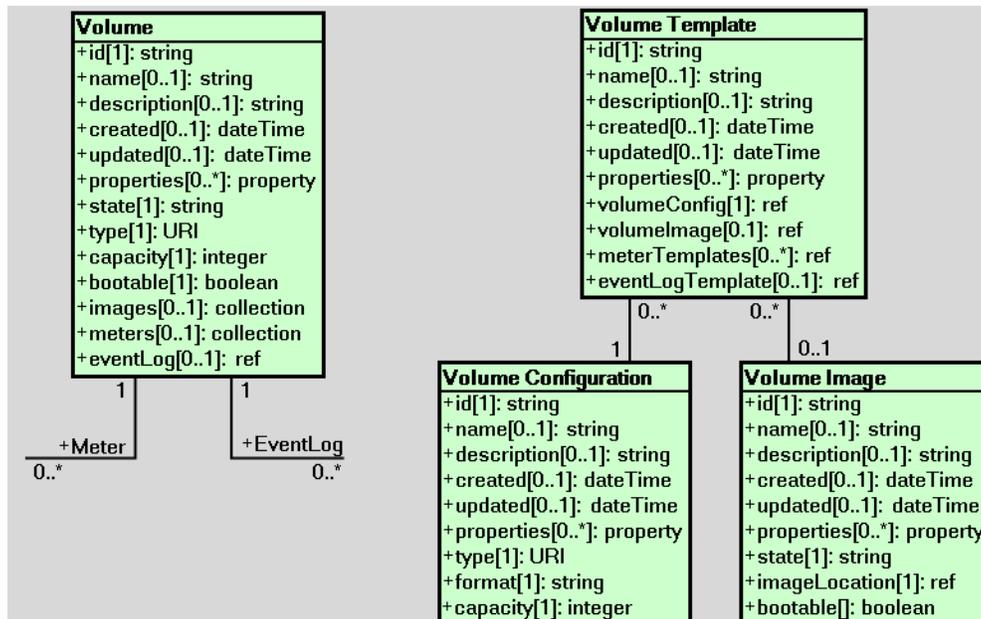
3750 5.14.12.1 Operations

3751 This Resource supports the Read and Update operations. Creation of new CredentialTemplate
 3752 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 3753 4.2.1.1.

3754 5.15 Volume Resources and relationships

3755 Figure 4 illustrates the Resources involved in constructing a Volume and their relationships. Although
 3756 this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor
 3757 normative.

3758



3759

Figure 4 - Volume Resources

3760 **5.15.1 Volume**

3761 A Volume represents storage at either the block or the file-system level. Volumes can be connected to
 3762 Machines. Once connected, Volumes can be accessed by processes on that Machine. Table 22
 3763 describes the Volume attributes.

3764

Table 22 – Volume attributes

Name	Volume	
Type URI	http://schemas.dmtf.org/cimi/2/Volume	
Attribute	Type	Description
state	string	The operational state of the Volume. Allowed values are: CREATING: The Volume is in the process of being created. AVAILABLE: The Volume is available and ready for use. Unless otherwise specified, the Volume shall be in this state initially after successful creation. CAPTURING: The Volume is in the process of being captured (snapshotted) into a new VolumeImage. RESTORING: The Volume is in the process of being restored. DELETING: The Volume is in the process of being deleted. ERROR: The Provider has detected an error in the Volume. The operations that result in transitions to the above defined states are defined in clause 5.15.1.2 Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
type	URI	A URI that indicates the type of Volume to be created. This specification defines the following URI: http://schemas.dmtf.org/cimi/2/mapped: Indicates a Volume that shall be used for shared storage that might be available to multiple Machines, but which does not require an explicit mount operation from within the guest operating system. Additional values may be defined. If certain types of Volumes require additional data, it is expected that this Resource is extended. For example, a "sharedFileSystem" type might require additional networking information and credentials to be specified.

Name	Volume	
Type URI	http://schemas.dmtf.org/cimi/2/Volume	
Attribute	Type	Description
		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</i> <i>[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 usage of <code>VolumeImages</code> by the <code>Volume</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</i> <i>[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
eventLog	<i>ref</i>	A reference to the <code>EventLog</code> of this <code>Volume</code> . Constraints: Provider: support optional; mutable Consumer: support optional; read-only

3765 When implementing or using `Volume`, Providers and Consumers shall adhere to the syntax and
 3766 semantics of its attributes as described in the above table as well as in the tables describing embedded
 3767 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 3768 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 3769 both JSON and XML:

3770 **JSON media type:** application/json

3771 **JSON serialization:**

```

3772 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Volume",
3773   "id": string,
3774   "name": string, ?
3775   "description": string, ?
3776   "created": string, ?
3777   "updated": string, ?
3778   "properties": { string: string, + }, ?
3779   "state": string,
3780   "type": string,
3781   "capacity": number,
3782   "bootable": boolean,
3783   "images": { "href": string }, ?
    
```

```

3784     "meters": { "href": string }, ?
3785     "eventLog": { "href": string }, ?
3786     "operations": [
3787         { "rel": "edit", "href": string }, ?
3788         { "rel": "delete", "href": string } ?
3789     ] ?
3790     ...
3791 }
    
```

3792 **XML media type:** application/xml

3793 **XML serialization:**

```

3794 <Volume xmlns="http://schemas.dmtf.org/cimi/2">
3795     <id> xs:anyURI </id>
3796     <name> xs:string </name> ?
3797     <description> xs:string </description> ?
3798     <created> xs:dateTime </created> ?
3799     <updated> xs:dateTime </updated> ?
3800     <property key="xs:string"> xs:string </property> *
3801     <state> xs:string </state>
3802     <type> xs:anyURI </type>
3803     <capacity> xs:integer </capacity>
3804     <bootable> xs:boolean </bootable>
3805     <images href="xs:anyURI"/> ?
3806     <meters href="xs:anyURI"/> ?
3807     <eventLog href="xs:anyURI"/> ?
3808     <operation rel="edit" href="xs:anyURI"/> ?
3809     <operation rel="delete" href="xs:anyURI"/> ?
3810     <xs:any>*
3811 </Volume>
    
```

3812 5.15.1.1 Collections

3813 The following clauses describe the Collection Resources owned by Volumes.

3814 5.15.1.1.1 images Collection

3815 The Resource type for each item of this Collection is "VolumeImage". There is no accessory attribute
 3816 for the items in this Collection, therefore it is a basic VolumeImage Collection (serialized as described
 3817 in 5.5.12).

3818 See the VolumeImageCollection Resource clause.

3819 **NOTE** Previous versions of this specification included an "add" operation on this Resource. It is now deprecated in
 3820 favor of creating a new VolumeImage with the imageLocation attribute pointing to the Volume to be captured.

3821 **5.15.1.1.2 meters Collection**

3822 The Resource type for each item of this Collection is “Meter” as defined in clause 5.17.3. There is no
 3823 accessory attribute for the items in this Collection, therefore it is a basic Meter Collection (serialized as
 3824 described in 5.5.12).

3825 See the MeterCollection Resource clause.

3826 **5.15.1.2 Operations**

3827 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 3828 VolumeCollection Resource.

3829 In addition also the following custom operations are supported.

3830 **snapshot**

3831 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/snapshot`

3832 This operation shall create a new VolumeImage from an existing Volume. This operation is defined
 3833 within the VolumeImage Resource; see 5.15.7.1 for more details. Note that while this operation is
 3834 performed against a VolumeImage, its presence in the Volume serialization is used to advertise
 3835 support for the operation.

3836 If the Volume is unable to accept operations at any point while it is creating the VolumeImage, the
 3837 Volume shall be in the state “CAPTURING”.

3838 **restore**

3839 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/restore`

3840 This operation shall restore a Volume from a previously created VolumeImage.

3841 Input parameters:

- 3842 1) "image" - type: ref - mandatory
- 3843 2) A reference to the Volume Image.

3844 Output parameters: None.

3845 During the processing of this operation, the Volume shall be in the "RESTORING" state.

3846 Upon successful completion of this operation, the Volume shall again be in the state “AVAILABLE”.

3847 **HTTP protocol**

3848 To restore a Volume, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/restore" URI of the
 3849 Volume where the HTTP request body shall be as described below.

3850 **JSON media type:** application/json

3851 **JSON serialization:**

```
3852 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
3853   "action": "http://schemas.dmtf.org/cimi/2/action/restore",
3854   "image": { "href" : string },
3855   "properties": { string: string, + } ?
3856   ...
```

3857 }
 3858

XML media type: application/xml

3859 **XML serialization**

```
3860 <Action xmlns="http://schemas.dmtf.org/cimi/2">
3861   <action> http://schemas.dmtf.org/cimi/2/action/restore </action>
3862   <image href="xs:anyURI"/>
3863   <property key="xs:string"> xs:string </property> *
3864   <xs:any>*
3865 </Action>
```

3866 Where the "image" ref content is a reference to the VolumeImage to be used.

3867 Upon successful processing of the request, the HTTP response body may be empty.

3868 5.15.2 VolumeCollection Resource

3869 A VolumeCollection Resource represents the Collection of Volumes within a Provider and follows
 3870 the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

3871 **JSON serialization:**

```
3872 { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeCollection",
3873   "id": string,
3874   "count": number,
3875   "volumes": [
3876     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Volume",
3877       "id": string,
3878       ... remaining Volume attributes ...
3879     }, +
3880   ], ?
3881   "operations": [ { "rel": "add", "href": string } ? ]
3882   ...
3883 }
```

3884 **XML serialization:**

```
3885 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/VolumeCollection"
3886   xmlns="http://schemas.dmtf.org/cimi/2">
3887   <id> xs:anyURI </id>
3888   <count> xs:integer </count>
3889   <Volume>
3890     <id> xs:anyURI </id>
3891     ... remaining Volume attributes ...
3892   </Volume> *
3893   <operation rel="add" href="xs:anyURI"/> ?
3894   <xs:any>*
```

3895 </Collection>

3896 **5.15.2.1 Operations**

3897 NOTE The "add" operation requires that a VolumeTemplate be used (see 4.2.1.1).

3898 **5.15.3 VolumeTemplate Resource**

3899 This Resource captures the configuration values for realizing a Volume. A VolumeTemplate may be
 3900 used to create multiple Volumes. Table 23 describes the VolumeTemplate attributes.

3901 **Table 23 – VolumeTemplate attributes**

Name	VolumeTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/VolumeTemplate	
Attribute	Type	Description
volumeConfig	ref	A reference to the VolumeConfiguration that is used to create a Volume from this VolumeTemplate. Note that the attributes of the VolumeConfiguration may be specified rather than a reference to an existing VolumeConfiguration Resource. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
volumeImage	ref	A reference to the VolumeImage that is used to create a Volume from this VolumeTemplate. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
meterTemplates	Meter Templates[]	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	ref	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new Volume. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

3902 When implementing or using VolumeTemplate, Providers and Consumers shall adhere to the syntax
 3903 and semantics of its attributes as described in the above table as well as in the tables describing
 3904 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
 3905 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 3906 Resource in both JSON and XML.

3907 **JSON media type:** application/json

3908 **JSON serialization:**

```
3909 { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeTemplate",
3910   "id": string,
3911   "name": string, ?
3912   "description": string, ?
```

```

3913 "created": string, ?
3914 "updated": string, ?
3915 "properties": { string: string, + }, ?
3916 "volumeConfig": {
3917     "href": string | ... VolumeConfiguration attributes ...
3918 },
3919 "volumeImage": { "href": string }, ?
3920 "meterTemplates": [
3921     { "href": string, ?
3922     ... MeterTemplate attributes ... ?
3923     }, *
3924 ], ?
3925 "eventLogTemplate": {
3926     "href": string, ?
3927     ... EventLogTemplate attributes ... ?
3928 }, ?
3929 "operations": [
3930     { "rel": "edit", "href": string }, ?
3931     { "rel": "delete", "href": string } ?
3932 ] ?
3933 ...
3934 }
    
```

3935 **XML media type:** application/xml

3936 **XML serialization:**

```

3937 <VolumeTemplate xmlns="http://schemas.dmtf.org/cimi/2">
3938     <id> xs:anyURI </id>
3939     <name> xs:string </name> ?
3940     <description> xs:string </description> ?
3941     <created> xs:dateTime </created> ?
3942     <updated> xs:dateTime </updated> ?
3943     <property key="xs:string"> xs:string </property> *
3944     <volumeConfig href="xs:anyURI"?>
3945         ... VolumeConfiguration attributes ... ?
3946     </volumeConfig>
3947     <volumeImage href="xs:anyURI"/> ?
3948     <meterTemplate href="xs:anyURI"? >
3949         ... MeterTemplate attributes ... ?
3950     </meterTemplate> *
3951     <eventLogTemplate href="xs:anyURI"? >
    
```

```

3952     ... EventLogTemplate attributes ... ?
3953 </eventLogTemplate> ?
3954 <operation rel="edit" href="xs:anyURI"/> ?
3955 <operation rel="delete" href="xs:anyURI"/> ?
3956 <xs:any>*
3957 </VolumeTemplate>

```

3958 5.15.3.1 Operations

3959 This Resource supports the Read, Update, and Delete operations. Create is supported through the
3960 VolumeTemplateCollection Resource.

3961 5.15.4 VolumeTemplateCollection Resource

3962 A VolumeTemplateCollection Resource represents the Collection of VolumeTemplate
3963 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
3964 shall be serialized as follows:

3965 JSON serialization:

```

3966 { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeTemplateCollection",
3967   "id": string,
3968   "count": number,
3969   "volumeTemplates": [
3970     { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeTemplate",
3971       "id": string,
3972       ... remaining volumeTemplate attributes ...
3973     }, +
3974   ], ?
3975   "operations": [ { "rel": "add", "href": string } ? ]
3976   ...
3977 }

```

3978 XML serialization:

```

3979 <Collection
3980   resourceURI="http://schemas.dmtf.org/cimi/2/VolumeTemplateCollection"
3981   xmlns="http://schemas.dmtf.org/cimi/2">
3982   <id> xs:anyURI </id>
3983   <count> xs:integer </count>
3984   <VolumeTemplate>
3985     <id> xs:anyURI </id>
3986     ... remaining VolumeTemplates attributes ...
3987   </VolumeTemplate> *
3988   <operation rel="add" href="xs:anyURI"/> ?
3989   <xs:any>*
3990 </Collection>

```

3991 **5.15.4.1 Operations**

3992 This Resource supports the Read and Update operations. Creation of new `VolumeTemplate`
 3993 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 3994 4.2.1.1.

3995 **5.15.5 VolumeConfiguration Resource**

3996 The `VolumeConfiguration` Resource represents the set of configuration values needed to create a
 3997 `Volume` with certain characteristics. `VolumeConfigurations` are created by Providers and may, at
 3998 the Providers discretion, be created by Consumers.

3999 Table 24 describes the `VolumeConfiguration` attributes.

4000 **Table 24 – VolumeConfiguration attributes**

Name	VolumeConfiguration	
Type URI	http://schemas.dmtf.org/cimi/2/VolumeConfiguration	
Attribute	Type	Description
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/2/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. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
format	<i>string</i>	The format of the file system that is placed on <code>Volumes</code> created from this configuration. This attribute is only meaningful for <code>VolumeConfigurations</code> that describe block devices. This attribute is optional; the absence of this attribute indicates that <code>Volumes</code> 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	<i>integer</i>	The default size in kilobytes, if limited, of the <code>Volume</code> created from this <code>VolumeConfiguration</code> . Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

4001 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

4002 **JSON media type:** application/json

4003 **JSON serialization:**

```

4004 { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeConfiguration",
4005   "id": string,
4006   "name": string, ?
4007   "description": string, ?
4008   "created": string, ?
4009   "updated": string, ?
4010   "properties": { string: string, + }, ?
4011   "type": string,
```

```

4012     "format": string,
4013     "capacity": number,
4014     "operations": [
4015         { "rel": "edit", "href": string }, ?
4016         { "rel": "delete", "href": string } ?
4017     ] ?
4018     ...
4019 }

```

4020 **XML media type:** application/xml

4021 **XML serialization:**

```

4022 <VolumeConfiguration xmlns="http://schemas.dmtf.org/cimi/2">
4023     <id> xs:anyURI </id>
4024     <name> xs:string </name> ?
4025     <description> xs:string </description> ?
4026     <created> xs:dateTime </created> ?
4027     <updated> xs:dateTime </updated> ?
4028     <property key="xs:string"> xs:string </property> *
4029     <type> xs:anyURI </type>
4030     <format> xs:string </format>
4031     <capacity> xs:integer </capacity>
4032     <operation rel="edit" href="xs:anyURI"/> ?
4033     <operation rel="delete" href="xs:anyURI"/> ?
4034     <xs:any>*
4035 </VolumeConfiguration>

```

4036 5.15.5.1 Operations

4037 This Resource supports the Read, Update, and Delete operations. Create is supported through the
4038 VolumeConfigurationCollection Resource.

4039 5.15.6 VolumeConfigurationCollection Resource

4040 A VolumeConfigurationCollection Resource represents the Collection of
4041 VolumeConfiguration Resources within a Provider and follows the Collection pattern defined in
4042 clause 5.5.12. This Resource shall be serialized as follows:

4043 **JSON serialization:**

```

4044 { "resourceURI":
4045     "http://schemas.dmtf.org/cimi/2/VolumeConfigurationCollection",
4046     "id": string,
4047     "count": number,
4048     "volumeConfigurations": [
4049         { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeConfiguration",

```

```

4050     "id": string,
4051     ... remaining VolumeConfiguration attributes ...
4052   }, +
4053 ], ?
4054 "operations": [ { "rel": "add", "href": string } ? ]
4055 ...
4056 }
    
```

XML serialization:

```

4057 <Collection
4058   resourceURI="http://schemas.dmtf.org/cimi/2/VolumeConfigurationCollection"
4059   xmlns="http://schemas.dmtf.org/cimi/2">
4060   <id> xs:anyURI </id>
4061   <count> xs:integer </count>
4062   <VolumeConfiguration>
4063     <id> xs:anyURI </id>
4064     ... remaining VolumeConfiguration attributes ...
4065   </VolumeConfiguration> *
4066   <operation rel="add" href="xs:anyURI"/> ?
4067   <xs:any>*
4068 </Collection>
    
```

5.15.6.1 Operations

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

5.15.7 VolumeImage Resource

This Resource represents an image that could be placed on a preloaded volume. Table 25 describes the VolumeImage attributes.

Table 25 – VolumeImage attributes

Name	VolumeImage	
Type URI	http://schemas.dmtf.org/cimi/2/VolumeImage	
Attribute	Type	Description
state	string	The operational state of the VolumeImage. Allowed values are: 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:

Name	VolumeImage	
Type URI	http://schemas.dmtf.org/cimi/2/VolumeImage	
Attribute	Type	Description
		Provider: support mandatory; mutable Consumer: support mandatory; read-write
bootable	<i>boolean</i>	This property indicates whether Volumes created from this VolumeImage are bootable. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

4077 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

4078 **JSON media type:** application/json

4079 **JSON serialization:**

```

4080 { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeImage",
4081   "id": string,
4082   "name": string, ?
4083   "description": string, ?
4084   "created": string, ?
4085   "updated": string, ?
4086   "properties": { string: string, + }, ?
4087   "state": string,
4088   "imageLocation": string,
4089   "bootable": boolean,
4090   "operations": [
4091     { "rel": "edit", "href": string }, ?
4092     { "rel": "delete", "href": string } ?
4093   ] ?
4094   ...
4095 }
```

4096 **XML media type:** application/xml

4097 **XML serialization:**

```

4098 <VolumeImage xmlns="http://schemas.dmtf.org/cimi/2">
4099   <id> xs:anyURI </id>
4100   <name> xs:string </name> ?
4101   <description> xs:string </description> ?
4102   <created> xs:dateTime </created> ?
4103   <updated> xs:dateTime </updated> ?
4104   <property key="xs:string"> xs:string </property> *
4105   <state> xs:string </state>
4106   <imageLocation>xs:anyURI</imageLocation>
4107   <bootable> xs:boolean </bootable>
```

```

4108     <operation rel="edit" href="xs:anyURI"/> ?
4109     <operation rel="delete" href="xs:anyURI"/> ?
4110     <xs:any>*
4111 </VolumeImage>
    
```

4112 5.15.7.1 Operations

4113 This Resource supports the Read, Update, and Delete operations. Create is supported through the
 4114 VolumeImageCollection Resource.

4115 5.15.8 VolumeImageCollection Resource

4116 A VolumeImageCollection Resource represents the Collection of VolumeImage Resources
 4117 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
 4118 serialized as follows:

4119 JSON serialization:

```

4120 { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeImageCollection",
4121   "id": string,
4122   "count": number,
4123   "volumeImages": [
4124     { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeImage",
4125       "id": string,
4126       ... remaining VolumeImage attributes ...
4127     }, +
4128   ], ?
4129   "operations": [ { "rel": "add", "href": string } ? ]
4130   ...
4131 }
    
```

4132 XML serialization:

```

4133 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/VolumeImageCollection"
4134   xmlns="http://schemas.dmtf.org/cimi/2">
4135   <id> xs:anyURI </id>
4136   <count> xs:integer </count>
4137   <VolumeImage>
4138     <id> xs:anyURI </id>
4139     ... remaining VolumeImage attributes ...
4140   </VolumeImage> *
4141   <operation rel="add" href="xs:anyURI"/> ?
4142   <xs:any>*
4143 </Collection>
    
```

4144 5.15.8.1 Operations

4145 This Resource supports the Read and Update operations. Creation of new `VolumeImage` Resources is
4146 supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.

4147 During the creation of a new `VolumeImage` Resource, if the "imageLocation" attribute refers to an
4148 existing `Volume`, this operation shall be interpreted as a request to create a snapshot of the `Volume`.
4149 Once completed, the "imageLocation" attribute of the new `VolumeImage` Resource shall not refer to the
4150 original `Volume`; instead it shall refer to a static copy of the `Volume`. Additionally, the referenced
4151 `Volume`'s `VolumeImageCollection` shall be updated to include a reference to this newly created
4152 snapshot `VolumeImage` Resource. During this process, the Provider may put the `Volume` into a
4153 "CAPTURING" state if necessary.

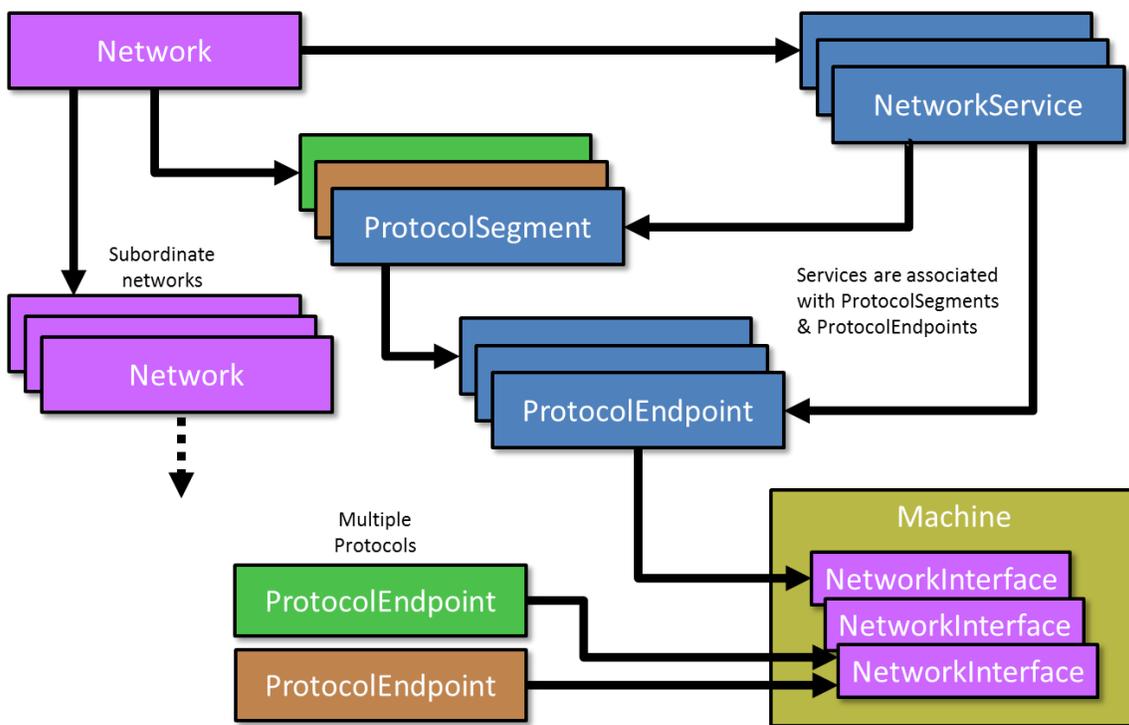
4154 5.16 Network Resources and relationships

4155 A Network is a logical construct that allows communication between defined Endpoints within a Segment.
4156 Each Segment uses a single, fixed, protocol to communicate and access is provided by associating an
4157 Endpoint with an Interface.

4158 Only Endpoints within a Segment can communicate implicitly. All other communication must be explicitly
4159 enabled using Network Services.

- 4160 • Each Network has one or more Segments
- 4161 • Each Segment supports communication using a single protocol
- 4162 • Each Segment may have one or more addressable Endpoints
- 4163 • Each Endpoint is associated with a single Segment
- 4164 • Each Endpoint may be associated with a single Interface
- 4165 • An Interface can be associated with more than one Endpoint
- 4166 • A Network may contain subordinate Networks to form hierarchical structures (similar to Systems)
- 4167 • One or more Services may be associated with a Network to provide additional functionality

4168 Figure 5 illustrates the Resources involved in constructing Networks. Although this drawing is in the style
4169 of a Resource Relationship diagram, the use of UML is neither rigorous nor normative.



4170 **Figure 5 - Network Resources**

4171 **5.16.1 Network**

4172 Table 26 describes the Network Resource attributes.

4173 **Table 26 – Network attributes**

Name	Network	
Type URI	http://schemas.dmtf.org/cimi/2/Network	
Attribute	Type	Description
state	<i>string</i>	The operational state of the Network. Allowed values are: CREATING: The Network is in the process of being created. STARTING: The Network is in the process of being started. STARTED: The Network is available and ready for use. STOPPING: The Network is in the process of being stopped. STOPPED: The Network is stopped and not available for use. DELETING: The Network is in the process of being deleted. ERROR: The Provider has detected an error in the Network. <u>The operations that result in transitions to the above defined states are defined in clause 5.16.1.2. Clause 5.16.2.1 defines the initial state of a Network.</u> Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
segments	<i>collection [Protocol Segment]</i>	A reference to a Collection of Segments contained within this Network. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
services	<i>collection [Network Service]</i>	A reference to a Collection of Services that may be applied to this Network. Constraints: Provider: support mandatory; mutable

Name	Network	
Type URI	http://schemas.dmtf.org/cimi/2/Network	
Attribute	Type	Description
		Consumer: support mandatory; read-only
subnetworks	<i>collection</i> <i>[Network]</i>	A reference to a Collection of subordinate Networks contained within this Network. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
meters	<i>collection</i> <i>[Meter]</i>	A reference to the list of Meters monitored for this Network. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLog	<i>ref</i>	A reference to the EventLog of this Network. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

4174 The Provider shall supply at least one `Network` Resource in the CEP `Networks` Collection to
 4175 represent communication channels that are external to the Consumers cloud. Typically this would be a
 4176 connection to the Internet. As an alternative the Provider may supply a `NetworkTemplate` Resource
 4177 by which such external Networks can be created when required.

4178 When implementing or using `Network` Resources, Providers and Consumers shall adhere to the syntax
 4179 and semantics of its attributes as described in Table 26 as well as in the tables describing embedded
 4180 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 4181 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 4182 both JSON and XML.

4183 **JSON media type:** application/json

4184 **JSON serialization:**

```

4185 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Network",
4186   "id": string,
4187   "name": string, ?
4188   "description": string, ?
4189   "created": string, ?
4190   "updated": string, ?
4191   "parent": string, ?
4192   "properties": { string: string, + }, ?
4193   "state": string,
4194   "segments": { "href": string },
4195   "seVICES": { "href": string },
4196   "subnetworks": { "href": string }, ?
4197   "meters": { "href": string }, ?
4198   "eventLog": { "href": string }, ?
4199   "operations": [
4200     { "rel": "edit", "href": string }, ?
4201     { "rel": "delete", "href": string }, ?
4202     { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string }, ?
    
```

```

4203     { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string } ?
4204   ] ?
4205   ...
4206 }

```

4207 **XML media type:** application/xml

4208 **XML serialization:**

```

4209 <Network xmlns="http://schemas.dmtf.org/cimi/2">
4210   <id> xs:anyURI </id>
4211   <name> xs:string </name> ?
4212   <description> xs:string </description> ?
4213   <created> xs:dateTime </created> ?
4214   <updated> xs:dateTime </updated> ?
4215   <parent> xs:anyURI </parent> ?
4216   <property key="xs:string"> xs:string </property> *
4217   <state> xs:string </state>
4218   <segments href="xs:anyURI"/>
4219   <services href="xs:anyURI"/>
4220   <subnetworks href="xs:anyURI"/> ?
4221   <meters href="xs:anyURI"/> ?
4222   <eventLog href="xs:anyURI"/> ?
4223   <operation rel="edit" href="xs:anyURI"/> ?
4224   <operation rel="delete" href="xs:anyURI"/> ?
4225   <operation rel="http://schemas.dmtf.org/cimi/2/action/start"
4226     href="xs:anyURI"/> ?
4227   <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
4228     href="xs:anyURI"/> ?
4229   <xs:any>*
4230 </Network>

```

4231 5.16.1.1 Collections

4232 The following clauses describe the Collection Resources that are components of Networks.

4233 5.16.1.1.1 segments Collection

4234 The Resource type for each item of this Collection is "ProtocolSegment". There is no accessory
4235 attribute for the items in this Collection, therefore it is a basic ProtocolSegmentCollection, as
4236 described in 5.16.6.

4237 5.16.1.1.2 services Collection

4238 The Resource type for each item of this Collection is "NetworkService". There is no accessory
4239 attribute for the items in this Collection, therefore it is a basic NetworkServiceCollection, as
4240 described in 5.16.18

4241 **5.16.1.1.3 subnetworks Collection**

4242 The Resource type for each item of this Collection is "Network". There is no accessory attribute for the
4243 items in this Collection, therefore it is a basic `NetworkCollection`, as described in 5.16.2.

4244 **5.16.1.1.4 meters Collection**

4245 The Resource type for each item of this Collection is "Meter" as defined in clause 5.17.3. There is no
4246 accessory attribute for the items in this Collection, therefore it is a basic `MeterCollection` as
4247 described in 5.5.12.

4248 See the `MeterCollection` Resource clause.

4249 **5.16.1.2 Operations**

4250 `Network` Resources support the Read, Update, and Delete operations. Create is supported through the
4251 `NetworkCollection` Resource, as described in 5.16.2.

4252 The following custom operations are also defined:

4253 **start**

4254 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/start`

4255 This operation shall recursively start and enable all the components within a `Network`.

4256 Input parameters: None.

4257 Output parameters: None.

4258 During the processing of this operation, the `Network` shall be in the "STARTING" state.

4259 Upon successful completion of this operation, the `Network` shall be in the "STARTED" state.

4260 **HTTP protocol**

4261 To start a `Network`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/start" URI of the
4262 `Network` where the HTTP request body shall be as described below.

4263 **JSON media type:** `application/json`

4264 **JSON serialization:**

```
4265 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
4266   "action": "http://schemas.dmtf.org/cimi/2/action/start",
4267   "properties": { string: string, + } ?
4268   ...
4269 }
```

4270 **XML media type:** `application/xml`

4271 **XML serialization**

```
4272 <Action xmlns="http://schemas.dmtf.org/cimi/2">
4273   <action> http://schemas.dmtf.org/cimi/2/action/start </action>
4274   <property key="xs:string"> xs:string </property> *
4275   <xs:any>*
```

4276 `</Action>`

4277 Upon successful processing of the request, the HTTP response body may be empty.

4278 **stop**

4279 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/stop`

4280 This operation shall recursively stop and disable all components of a `Network`.

4281 Input parameters: None.

4282 Output parameters: None.

4283 During the processing of this operation, the `Network` shall be in the "STOPPING" state.

4284 Upon successful completion of this operation, the `Network` shall be in the "STOPPED" state.

4285 **HTTP protocol**

4286 To stop a `Network`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/stop" URI of the
4287 `Network` where the HTTP request body shall be as described below.

4288 **JSON media type:** `application/json`

4289 **JSON serialization:**

```
4290 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
4291   "action": "http://schemas.dmtf.org/cimi/2/action/stop",
4292   "properties": { string: string, + } ?
4293   ...
4294 }
```

4295 **XML media type:** `application/xml`

4296 **XML serialization**

```
4297 <Action xmlns="http://schemas.dmtf.org/cimi/2">
4298   <action> http://schemas.dmtf.org/cimi/2/action/stop </action>
4299   <property key="xs:string"> xs:string </property> *
4300   <xs:any>*
4301 </Action>
```

4302 Upon successful processing of the request, the HTTP response body may be empty.

4303 5.16.2 NetworkCollection Resource

4304 A `NetworkCollection` Resource represents the Collection of `Networks` and follows the Collection
4305 pattern that is defined in clause 5.5.12. This Resource shall be serialized as follows:

4306 **JSON serialization:**

```
4307 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkCollection",
4308   "id": string,
4309   "count": number,
4310   "networks": [
```

```

4311     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Network",
4312       "id": string,
4313       ... remaining Network attributes ...
4314     }, +
4315   ], ?
4316   "operations": [ { "rel": "add", "href": string } ? ]
4317   ...
4318 }

```

4319 XML serialization:

```

4320 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/NetworkCollection"
4321   xmlns="http://schemas.dmtf.org/cimi/2">
4322   <id> xs:anyURI </id>
4323   <count> xs:integer </count>
4324   <Network>
4325     <id> xs:anyURI </id>
4326     ... remaining Network attributes ...
4327   </Network> *
4328   <operation rel="add" href="xs:anyURI"/> ?
4329   <xs:any>*
4330 </Collection>

```

4331 5.16.2.1 Operations

4332 NOTE The "add" operation requires that a `NetworkTemplate` be used (see 5.16.3).

4333 Upon successful processing of the "add" operation, unless otherwise specified by the way of the
 4334 `NetworkTemplate` "initialState" attribute, the state of the new `Network` shall be the value of the
 4335 `DefaultInitialState` capability of the `Network` Resource's `ResourceMetadata`, if defined. If no
 4336 `DefaultInitialState` capability is defined, the default value shall be "STOPPED." The semantics of
 4337 "initialState" shall be equivalent to the Provider issuing the appropriate actions against the new `Network`
 4338 to move it into that state.

4339 If a Provider is unable to change the state of the new `Network` to the appropriate "initialState" (either as
 4340 specified by the `NetworkTemplate` or as implied by the previous stated rules), the `Network` creation
 4341 shall fail.

4342 5.16.3 NetworkTemplate Resource

4343 The `NetworkTemplate` is a set of configuration values for realizing a `Network`. An instance of
 4344 `NetworkTemplate` may be used to create multiple `Networks`. Table 27 describes the
 4345 `NetworkTemplate` attributes.

4346

Table 27 – NetworkTemplate attributes

Name	NetworkTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/NetworkTemplate	
Attribute	Type	Description
initialState	<i>string</i>	Sets the initial state of a Network created using this Template. The allowed values are restricted to the non-transient states specified for the state attribute of the Network Resource, described in Table 26. Providers should advertise the list of available values via the Network ResourceMetadata initialState Capability. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
segments	<i>Protocol Segment[]</i>	A list of references to existing ProtocolSegment Resources to be inserted into the “segments” collection of the Network Resource created using this Template. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
segmentTemplates	<i>Protocol Segment Templates[]</i>	A list of references to ProtocolSegmentTemplates, from each of which a ProtocolSegment Resource is created and its reference inserted into the “segments” collection of the Network Resource created using this NetworkTemplate. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
services	<i>Network Service[]</i>	A list of references to NetworkService Resources to be added to the “services” collection of the Network Resource created using this Template. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
serviceTemplates	<i>Network Service Templates[]</i>	A list of references to NetworkServiceTemplates, from each of which a NetworkService Resource is created and its reference inserted into the “services” collection of the Network Resource created using this Template. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
subnetworks	<i>Network[]</i>	A list of references to Network Resources to be added to the subnetworks collection of the Network created from this NetworkTemplate Constraints: Provider: support optional; mutable Consumer: support optional; read-write
subnetworkTemplates	<i>Network Templates[]</i>	A list of references to NetworkTemplates, from each of which a Network Resource is created and added to the subnetworks collection of the Network created using this NetworkTemplate. 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 Network. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

Name	NetworkTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/NetworkTemplate	
Attribute	Type	Description
eventLogTemplate	<i>ref</i>	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new Network. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

4347 When implementing or using NetworkTemplate, Providers and Consumers shall adhere to the syntax
4348 and semantics of its attributes as described in Table 27 as well as in the tables describing embedded
4349 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
4350 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
4351 both JSON and XML

4352 **JSON media type:** application/json

4353 **JSON serialization:**

```
4354 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkTemplate",
4355   "id": string,
4356   "name": string, ?
4357   "description": string, ?
4358   "created": string, ?
4359   "updated": string, ?
4360   "properties": { string: string, + }, ?
4361   "initialState": string, ?
4362   "segments": { "href": string }, ?
4363   "segmentTemplates": [
4364     { "href": string, ?
4365       ... ProtocolSegmentTemplate attributes ... ?
4366     }, *
4367   ], ?
4368   "services": { "href": string }, ?
4369   "serviceTemplates": [
4370     { "href": string, ?
4371       ... NetworkServiceTemplate attributes ... ?
4372     }, *
4373   ], ?
4374   "subnetworks": { "href": string }, ?
4375   "subnetworkTemplates": [
4376     { "href": string, ?
4377       ... NetworkTemplate attributes ... ?
4378     }, *
4379   ], ?
```

```

4380 "meterTemplates": [
4381     { "href": string, ?
4382       ... MeterTemplate attributes ... ?
4383     }, *
4384 ], ?
4385 "eventLogTemplate": {
4386     "href": string, ?
4387     ... EventLogTemplate attributes ... ?
4388 }, ?
4389 "operations": [
4390     { "rel": "edit", "href": string }, ?
4391     { "rel": "delete", "href": string } ?
4392 ] ?
4393 ...
4394 }
    
```

4395 **XML media type:** application/xml

4396 **XML serialization:**

```

4397 <NetworkTemplate xmlns="http://schemas.dmtf.org/cimi/2">
4398     <id> xs:anyURI </id>
4399     <name> xs:string </name> ?
4400     <description> xs:string </description> ?
4401     <created> xs:dateTime </created> ?
4402     <updated> xs:dateTime </updated> ?
4403     <property key="xs:string"> xs:string </property> *
4404     <initialState> xs:string </initialState> ?
4405     <segments href="xs:anyURI"/> ?
4406     <segmentTemplates href="xs:anyURI"? >
4407         ... ProtocolSegmentTemplate attributes ... ?
4408     </segmentTemplates> *
4409     <services href="xs:anyURI"/> ?
4410     <serviceTemplates href="xs:anyURI"? >
4411         ... NetworkServiceTemplates attributes ... ?
4412     </serviceTemplates> *
4413     <subnetworks href="xs:anyURI"/> ?
4414     <subnetworkTemplates href="xs:anyURI"? >
4415         ... NetworkTemplate attributes ... ?
4416     </subnetworkTemplate> *
4417     <meterTemplate href="xs:anyURI"? >
4418         ... MeterTemplate attributes ... ?
    
```

```

4419     </meterTemplate> *
4420     <eventLogTemplate href="xs:anyURI"? >
4421         ... EventLogTemplate attributes ... ?
4422     </eventLogTemplate> ?
4423     <operation rel="edit" href="xs:anyURI"/> ?
4424     <operation rel="delete" href="xs:anyURI"/> ?
4425     <xs:any>*
4426 </NetworkTemplate>

```

4427 5.16.3.1 Operations

4428 The NetworkTemplate Resource supports the Read, Update and Delete operations. Create is
4429 supported through the NetworkTemplateCollection Resource.

4430 5.16.4 NetworkTemplateCollection Resource

4431 A NetworkTemplateCollection Resource represents the Collection of NetworkTemplates
4432 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
4433 serialized as follows:

4434 JSON serialization:

```

4435 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkTemplateCollection",
4436   "id": string,
4437   "count": number,
4438   "networkTemplates": [
4439     { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkTemplate",
4440       "id": string,
4441       ... remaining NetworkTemplate attributes ...
4442     }, +
4443   ], ?
4444   "operations": [ { "rel": "add", "href": string } ? ]
4445   ...
4446 }

```

4447 XML serialization:

```

4448 <Collection
4449     resourceURI="http://schemas.dmtf.org/cimi/2/NetworkTemplateCollection"
4450     xmlns="http://schemas.dmtf.org/cimi/2">
4451     <id> xs:anyURI </id>
4452     <count> xs:integer </count>
4453     <NetworkTemplate>
4454         <id> xs:anyURI </id>
4455         ... remaining NetworkTemplate attributes ...
4456     </NetworkTemplate> *
4457     <operation rel="add" href="xs:anyURI"/> ?

```

```
4458 <xs:any>*
4459 </Collection>
```

4460 **5.16.4.1 Operations**

4461 The NetworkTemplateCollection Resource supports the Read and Update operations. Creation
 4462 of new NetworkTemplate Resources is supported by the way of a POST to the "add" operation's URI
 4463 as described in clause 4.2.1.1.

4464 **5.16.5 Segments**

4465 A Segment is an individual channel within a Network that utilizes a single communication protocol.
 4466 Segments are ProtocolSegment Resources, the attributes of which are described in Table 28
 4467 below. **Error! Reference source not found..**

4468 **Table 28 – ProtocolSegment attributes**

Name	ProtocolSegment	
Type URI	http://schemas.dmtf.org/cimi/2/ProtocolSegment	
Attribute	Type	Description
state	<i>string</i>	The operational state of the Segment. Allowed values are: CREATING: The Segment is in the process of being created. STARTED: The Segment is available (enabled) and ready for use. STOPPED: The Segment is stopped (disabled) and not available for use. DELETING: The Segment is in the process of being deleted. ERROR: The Provider has detected an error in the Segment. The operations that result in transitions to the above defined states are defined in clause 5.16.5.3. Clause 5.16.6.1 defines the initial state of a Segment. <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-only
protocol	<i>string</i>	The official name of the protocol supported by this segment. Allowed values are: Ethernet: As defined by IEEE 802.3 . IPv4: Internet Protocol version 4, as defined in RFC 791 . IPv6: Internet Protocol Version 6 as defined in RFC 2460 . <u>Constraints:</u> Provider: support mandatory; immutable Consumer: support mandatory; read-only
noDefault Routing	<i>boolean</i>	If set to TRUE the default communication between Endpoints within the Segment is disabled. Communication between Endpoints in this case must be performed by a Service. The default value is FALSE which enables communication between endpoints. <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-write
endpoints	<i>collection [Protocol Endpoint]</i>	A reference to a list of references to Endpoints associated with this Segment. <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-only
parameters	<i>map</i>	A polymorphic attribute the contents of which depend on the specific network protocol. As examples this would include "netmask" for IPv4 and "bandwidth" for "Ethernet". See the adjacent tables for details of the data to be included <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-only

Name	ProtocolSegment	
Type URI	http://schemas.dmtf.org/cimi/2/ProtocolSegment	
Attribute	Type	Description
meters	<i>collection</i> <i>[Meter]</i>	A reference to the list of <i>Meters</i> monitored for this Segment. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLog	<i>ref</i>	A reference to the <i>EventLog</i> of this Segment. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

4469 When implementing or using `ProtocolSegment` Resources, Providers and Consumers shall adhere
4470 to the syntax and semantics of its attributes as described in Table 28 as well as in the tables describing
4471 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
4472 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
4473 Resource in both JSON and XML.

4474 **JSON media type:** application/json

4475 **JSON serialization:**

```

4476 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolSegment",
4477   "id": string,
4478   "name": string, ?
4479   "description": string, ?
4480   "created": string, ?
4481   "updated": string, ?
4482   "properties": { string: string, + }, ?
4483   "state": string,
4484   "protocol": string,
4485   "noDefaultRouting": boolean,
4486   "endpoints": { "href": string },
4487   .."parameters": { string: string, + }, ?
4488   "meters": { "href": string }, ?
4489   "eventLog": { "href": string }, ?
4490   "operations": [
4491     { "rel": "edit", "href": string }, ?
4492     { "rel": "delete", "href": string }, ?
4493     { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string }, ?
4494     { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string } ?
4495   ] ?
4496   ...
4497 }
```

4498 **XML media type:** application/xml

4499 **XML serialization:**

```

4500 <ProtocolSegment xmlns="http://schemas.dmtf.org/cimi/2">
4501   <id> xs:anyURI </id>
4502   <name> xs:string </name> ?
4503   <description> xs:string </description> ?
4504   <created> xs:dateTime </created> ?
4505   <updated> xs:dateTime </updated> ?
4506   <property key="xs:string"> xs:string </property> *
4507   <state> xs:string </state>
4508   <protocol> xs:string </protocol>
4509   <noDefaultRouting> xs:boolean </noDefaultRouting >
4510   <endpoints href="xs:anyURI"/>
4511   <parameters key="xs:string"> xs:string </parameters> *
4512   <meters href="xs:anyURI"/> ?
4513   <eventLog" href="xs:anyURI"/> ?
4514   <operation rel="edit" href="xs:anyURI"/> ?
4515   <operation rel="delete" href="xs:anyURI"/> ?
4516   <operation rel="http://schemas.dmtf.org/cimi/2/action/start"
4517 href="xs:anyURI"/> ?
4518   <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
4519 href="xs:anyURI"/> ?
4520   <xs:any>*
4521 </ProtocolSegment>
    
```

4522 **5.16.5.1 Protocol specific parameters**

4523 Each Segment may require additional data that is specific to a communication protocol. This additional
 4524 data is specified using the `parameters` attribute of the `ProtocolSegment`. This specification defines
 4525 the following key – value pairs that must be supplied for the indicated protocols:

4526 **Table 29 - IPv6 ProtocolSegment parameters**

Name	IPv6ProtocolParameters	
Key	Value Type	Description
prefixLength	<i>integer</i>	The length of the prefix for IPv6 addresses that is used to specify a subnet. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
subnetAddress	<i>string</i>	The IPv6 subnet address for this subnet. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only

4527

Table 30 – IPv4 ProtocolSegment parameters

Name	IPv4ProtocolParameters	
Key	Value Type	Description
netmask	<i>string</i>	The IPv4 subnetwork mask that defines the subnet. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
subnetAddress	<i>string</i>	The IPv4 subnet address for this subnet. Constraints: Provider: support optional; immutable Consumer: support optional; read-only

4528

Table 31 – Ethernet ProtocolSegment parameters

Name	EthernetProtocolParameters	
Key	Value Type	Description
speed	<i>integer</i>	The current bandwidth of the Segment in Bits per second. If no accurate determination of speed is possible this attribute should contain the nominal bandwidth. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
mtu	<i>integer</i>	The active or negotiated maximum transmission unit (MTU) that can be supported by this Segment. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

4529 Note that Providers may support additional key – value pairs for the `parameter` attribute to extend the
 4530 existing protocols. Consumers are not required to process any additional key – value pairs but must
 4531 retrun them to the Provider in the serialization of `ProtocolSegments`.

4532 5.16.5.2 Collections

4533 The following clauses describe the Collection Resources that are components of `ProtocolSegments`.

4534 5.16.5.2.1 endpoints Collection

4535 The Resource type for each item of this Collection is a “`ProtocolEndpoint`” as defined in clause
 4536 5.16.9. There is no accessory attribute for the items in this Collection, therefore it is a basic
 4537 `ProtocolEndpointCollection` Resource, serialized as described in 5.16.10.

4538 5.16.5.2.2 meters Collection

4539 The Resource type for each item of this Collection is “`Meter`” as defined in clause 5.17.3. There is no
 4540 accessory attribute for the items in this Collection, therefore it is a basic `Meter` Collection (serialized as
 4541 described in 5.5.12).

4542 5.16.5.3 Operations

4543 The `ProtocolSegment` Resource supports the Read, Update, and Delete operations. Create is
 4544 supported through the `ProtocolSegmentCollection` Resource.

4545 Deleting a `ProtocolSegment` shall remove that Segment from the global (Cloud Entry Point)
 4546 `ProtocolSegmentCollection` and also all references to the Segment in Collections of other
 4547 Resources (e.g. from corresponding `Network segments Collection`).

4548 The following custom operations are also defined:

4549 **start**

4550 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/start`

4551 This operation shall start a `ProtocolSegment`.

4552 Input parameters: None.

4553 Output parameters: None.

4554 Upon successful completion of this operation, the `ProtocolSegment` shall be in the "STARTED"
 4555 state.

4556 **HTTP protocol**

4557 To start a `ProtocolSegment`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/start" URI
 4558 of the `ProtocolSegment` where the HTTP request body shall be as described below.

4559 **JSON media type:** `application/json`

4560 **JSON serialization:**

```
4561 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
4562   "action": "http://schemas.dmtf.org/cimi/2/action/start",
4563   "properties": { string: string, + } ?
4564   ...
4565 }
```

4566 **XML media type:** `application/xml`

4567 **XML serialization**

```
4568 <Action xmlns="http://schemas.dmtf.org/cimi/2">
4569   <action> http://schemas.dmtf.org/cimi/2/action/start </action>
4570   <property key="xs:string"> xs:string </property> *
4571   <xs:any>*
4572 </Action>
```

4573 Upon successful processing of the request, the HTTP response body may be empty.

4574 **stop**

4575 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/stop`

4576 This operation shall stop a `ProtocolSegment`.

4577 Input parameters: None.

4578 Output parameters: None.

4579 Upon successful completion of this operation, the `ProtocolSegment` shall be in the "STOPPED"
4580 state.

4581 HTTP protocol

4582 To stop a `ProtocolSegment`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/stop" URI
4583 of the `ProtocolSegment` where the HTTP request body shall be as described below.

4584 **JSON media type:** application/json

4585 JSON serialization:

```
4586 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
4587   "action": "http://schemas.dmtf.org/cimi/2/action/stop",
4588   "properties": { string: string, + } ?
4589   ...
4590 }
```

4591 **XML media type:** application/xml

4592 XML serialization

```
4593 <Action xmlns="http://schemas.dmtf.org/cimi/2">
4594   <action> http://schemas.dmtf.org/cimi/2/action/stop </action>
4595   <property key="xs:string"> xs:string </property> *
4596   <xs:any>*
4597 </Action>
```

4598 Upon successful processing of the request, the HTTP response body may be empty.

4599 5.16.6 ProtocolSegmentCollection Resource

4600 A `ProtocolSegmentCollection` Resource represents the Collection of `ProtocolSegments`
4601 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
4602 serialized as follows:

4603 JSON serialization:

```
4604 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolSegmentCollection",
4605   "id": string,
4606   "count": number,
4607   "segments": [
4608     { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolSegment",
4609       "id": string,
4610       ... remaining ProtocolSegment attributes ...
4611     }, +
4612   ], ?
4613   "operations": [ { "rel": "add", "href": string } ? ]
4614   ...
4615 }
```

4616 **XML serialization:**

```

4617 <Collection
4618 resourceURI="http://schemas.dmtf.org/cimi/2/ProtocolSegmentCollection"
4619     xmlns="http://schemas.dmtf.org/cimi/2">
4620     <id> xs:anyURI </id>
4621     <count> xs:integer </count>
4622     <ProtocolSegment>
4623         <id> xs:anyURI </id>
4624         ... remaining ProtocolSegment attributes ...
4625     </ProtocolSegment> *
4626     <operation rel="add" href="xs:anyURI"/> ?
4627     <xs:any>*
4628 </Collection>

```

4629 **5.16.6.1 Operations**

4630 NOTE The "add" operation requires that a ProtocolSegmentTemplate be used (see clause 5.16.7).

4631 ~~If ProtocolSegments are created through the global (Cloud Entry Point)~~
 4632 ~~ProtocolSegmentCollection's "add" operation, they are automatically associated with the~~
 4633 ~~corresponding Network, by addition of the ProtocolSegment's reference in the networkPorts Collection of~~
 4634 ~~the Network.~~

4635 Upon successful processing of the "add" operation, unless otherwise specified by the
 4636 ProtocolSegmentTemplate "initialState" attribute, the state of the new ProtocolSegment shall
 4637 be the value of the DefaultInitialState capability of the ProtocolSegment Resource's
 4638 ResourceMetadata, if defined. If no DefaultInitialState capability is defined, the default value shall be
 4639 "STOPPED." The semantics of "initialState" shall be equivalent to the Provider issuing the appropriate
 4640 actions against the new ProtocolSegment to move it into that state.

4641 If a Provider is unable to change the state of the new ProtocolSegment to the appropriate
 4642 "initialState" (either as specified by the ProtocolSegmentTemplate or as implied by the previous
 4643 stated rules), the ProtocolSegment creation shall fail.

4644 **5.16.7 ProtocolSegmentTemplate Resource**

4645 The ProtocolSegmentTemplate is a set of configuration values for realizing a
 4646 ProtocolSegment. A ProtocolSegmentTemplate may be used to create multiple
 4647 ProtocolSegments. Table 32 describes the ProtocolSegmentTemplate attributes.

4648

Table 32 – ProtocolSegmentTemplate attributes

Name	ProtocolSegmentTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/ProtocolSegmentTemplate	
Attribute	Type	Description
network	<i>ref</i>	<p>A reference to the Network to which the Segment created using this Template belongs.</p> <p>If this Template is used to create a new Segment through the global (Cloud Entry Point) <code>ProtocolSegmentCollection</code>, this attribute shall be present.</p> <p>If this Template is referenced from a <code>NetworkTemplate</code> and used to create a new Segment during the creation of a Network, this attribute shall either be absent or have the same value as the "id" attribute of the Network to which this Segment is being added.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>
initialState	<i>string</i>	<p>Sets the initial state of the Segment created using this Template. The allowed values are restricted to the non-transient states specified for the <code>state</code> attribute of the <code>ProtocolSegment</code> Resource, described in 5.16.5. Providers should advertise the list of available values via the <code>ProtocolSegment</code> ResourceMetadata <code>initialStates</code> Capability.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>
protocol	<i>string</i>	<p>Sets the protocol supported by the Segment created using this Template. The allowed values are those specified for the <code>protocol</code> attribute of the <code>ProtocolSegment</code> Resource, described in clause 5.16.5.</p> <p>Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only</p>
noDefault Routing	<i>boolean</i>	<p>Enables or disables default routing for the Segment created using this Template. Values are as described for the <code>noDefaultRouting</code> attribute of the <code>ProtocolSegment</code> Resource, described in clause 5.16.5.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>
endpoints	<i>Protocol Endpoint[]</i>	<p>A list of references to <code>ProtocolEndpoints</code> to be inserted into the <code>endpoints</code> Collection of the Segment created using this Template.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</p>
endpoint Templates	<i>Protocol Endpoint Template[]</i>	<p>A list of references to <code>ProtocolEndpointTemplates</code> that specify a set of Endpoints to be created and inserted into the <code>endpoints</code> Collection for the Segment created using this Template. Note that the Template attributes may be explicitly listed rather than providing a reference to an existing <code>ProtocolEndpointTemplate</code> Resource.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</p>
parameters	<i>map</i>	<p>A polymorphic attribute the contents of which depend on the specific protocol supported. The allowed key – value pairs are as specified in section 5.16.5.1.</p> <p>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</p>

Name	ProtocolSegmentTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/ProtocolSegmentTemplate	
Attribute	Type	Description
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>ProtocolSegment</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>ProtocolSegment</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

4649 When implementing or using *ProtocolSegmentTemplate* Resources, Providers and Consumers
 4650 shall adhere to the syntax and semantics of its attributes as described in Table 32 as well as in the tables
 4651 describing embedded Resources or related Collections. Both Consumer and Provider shall serialize this
 4652 Resource as described below. The following pseudo-schemas (see notation in 1.3) describe the
 4653 serialization of the Resource in both JSON and XML.

4654 **JSON media type:** application/json

4655 **JSON serialization:**

```

4656 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolSegmentTemplate",
4657   "id": string,
4658   "name": string, ?
4659   "description": string, ?
4660   "created": string, ?
4661   "updated": string, ?
4662   "properties": { string: string, + }, ?
4663   "network": { "href": string }, ?
4664   "initialState": string, ?
4665   "protocol": string,
4666   "noDefaultRouting": boolean,
4667   "endpoints": { "href": string }, *
4668   "endpointTemplates": [
4669     { "href": string, ?
4670       ... ProtocolEndpointTemplate attributes ... ?
4671     }, *
4672   ], ?
4673   .."parameters": { string: string, + }, ?
4674   "meterTemplates": [
4675     { "href": string, ?
4676       ... MeterTemplate attributes ... ?
    
```

```

4677     }, *
4678   ], ?
4679   "eventLogTemplate": {
4680     "href": string, ?
4681     ... EventLogTemplate attributes ... ?
4682   }, ?
4683   "operations": [
4684     { "rel": "edit", "href": string }, ?
4685     { "rel": "delete", "href": string } ?
4686   ] ?
4687   ...
4688 }

```

4689 **XML media type:** application/xml

4690 **XML serialization:**

```

4691 <ProtocolSegmentTemplate xmlns="http://schemas.dmtf.org/cimi/2">
4692   <id> xs:anyURI </id>
4693   <name> xs:string </name> ?
4694   <description> xs:string </description> ?
4695   <created> xs:dateTime </created> ?
4696   <updated> xs:dateTime </updated> ?
4697   <property key="xs:string"> xs:string </property> *
4698   <network href="xs:anyURI"/> ?
4699   <initialState> xs:string </initialState> ?
4700   <protocol> xs:string </protocol>
4701   <noDefaultRouting> xs:boolean </noDefaultRouting >
4702   <endpoints href="xs:anyURI"/> *
4703   <endpointTemplate href="xs:anyURI"? >
4704     ... ProtocolEndpointTemplate attributes ... ?
4705   </endpointTemplate> *
4706   <parameters key="xs:string"> xs:string </parameters> *
4707   <meterTemplate href="xs:anyURI"? >
4708     ... MeterTemplate attributes ... ?
4709   </meterTemplate> *
4710   <eventLogTemplate href="xs:anyURI"? >
4711     ... EventLogTemplate attributes ... ?
4712   </eventLogTemplate> ?
4713   <operation rel="edit" href="xs:anyURI"/> ?
4714   <operation rel="delete" href="xs:anyURI"/> ?
4715   <xs:any>*

```

4716 `</ProtocolSegmentTemplate>`

4717 **5.16.7.1 Collections**

4718 The `ProtocolSegmentTemplate.Resource` has no attributes of type `Collection`.

4719 **5.16.7.2 Operations**

4720 The `ProtocolSegmentTemplate` Resource supports the Read, Update, and Delete operations.
 4721 Create is supported through the `ProtocolSegmentTemplateCollection` Resource.

4722 **5.16.8 ProtocolSegmentTemplateCollection Resource**

4723 A `ProtocolSegmentTemplateCollection` Resource represents the Collection of
 4724 `ProtocolSegmentTemplates` within a Provider and follows the Collection pattern defined in clause
 4725 5.5.12. This Resource shall be serialized as follows:

4726 **JSON serialization:**

```
4727 { "resourceURI":
4728     "http://schemas.dmtf.org/cimi/2/ProtocolSegmentTemplateCollection",
4729     "id": string,
4730     "count": number,
4731     "protocolSegmentTemplates": [
4732         { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolSegmentTemplate",
4733           "id": string,
4734           ... remaining ProtocolSegmentTemplate attributes ...
4735         }, +
4736     ], ?
4737     "operations": [ { "rel": "add", "href": string } ? ]
4738     ...
4739 }
```

4740 **XML serialization:**

```
4741 <Collection
4742
4743 resourceURI="http://schemas.dmtf.org/cimi/2/ProtocolSegmentTemplateCollection"
4744     xmlns="http://schemas.dmtf.org/cimi/2">
4745     <id> xs:anyURI </id>
4746     <count> xs:integer </count>
4747     <ProtocolSegmentTemplate>
4748         <id> xs:anyURI </id>
4749         ... remaining ProtocolSegmentTemplate attributes ...
4750     </ProtocolSegmentTemplate> *
4751     <operation rel="add" href="xs:anyURI"/> ?
4752     <xs:any>*
4753 </Collection>
```

4754 **5.16.8.1 Operations**

4755 The `ProtocolSegmentTemplateCollection` Resource supports the Read and Update
 4756 operations. Creation of new `ProtocolSegmentTemplate` Resources is supported by the way of a
 4757 POST to the "add" operation's URI as described in clause 4.2.1.1.

4758 **5.16.9 Endpoints**

4759 An Endpoint is an addressable element within a protocol that is a source, destination , or source and
 4760 destination for communication. Endpoints are `ProtocolEndpoint` Resources, the attributes of which
 4761 are described in Table 33.

4762 **Table 33 – ProtocolEndpoint attributes**

Name	ProtocolSegment	
Type URI	http://schemas.dmtf.org/cimi/2/ProtocolEndpoint	
Attribute	Type	Description
state	<i>string</i>	The operational state of the Endpoint. Allowable values are: CREATING: The Endpoint is in the process of being created. ENABLED: The Endpoint is available and ready for use. DISABLED: The Endpoint is not available for use. DELETING: The Endpoint is in the process of being deleted. ERROR: The Provider has detected an error in the Endpoint. The operations that result in transitions to the above defined states are defined in clause 5.16.9.3. Clause 5.16.10.1 defines the initial state of an Endpoint. <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-only
protocol	<i>string</i>	The official name of the protocol supported by this segment. This attribute is intended as a convenience only and if specified its value must be identical to the value of the <code>protocol</code> attribute of the Segment with which the Endpoint is associated. Possible values are those specified in the <code>ProtocolSegment</code> Resource described in section 5.16.5. <u>Constraints:</u> Provider: support optional; immutable Consumer: support optional; read-only
address	<i>string</i>	The address assigned to this Endpoint in the format required by the supported protocol. <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-write
origin	<i>string</i>	A string representing how protocol specific data is assigned to this Endpoint. Allowable values are: [STATIC DYNAMIC] In general the Consumer is responsible for assignment of static data, usually from within the guest software. The Provider may assign data dynamically when the end point is created, or it may be assigned via a Service associated with the Segment to which the Endpoint belongs. (E.g. DHCP). <u>Constraints:</u> Provider: support mandatory; immutable Consumer: support mandatory; read-only
interface	<i>Network Interface</i>	A reference to the Interface that is used to connect to the Network using this Endpoint. <u>Constraints:</u> Provider: support mandatory; mutable Consumer: support mandatory; read-only

parameters	<i>map</i>	A polymorphic attribute the contents of which depend on the specific network protocol. As examples this would include "netmask" for IPv4 and "bandwidth" for "Ethernet". See the adjacent tables for details of the data to be included Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
meters	<i>collection</i> <i>[Meter]</i>	A reference to the list of <i>Meters</i> monitored for this Endpoint. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLog	<i>ref</i>	A reference to the <i>EventLog</i> of this Endpoint. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

4763 When implementing or using `ProtocolEndpoint`, Providers and Consumers shall adhere to the
 4764 syntax and semantics of its attributes as described in Table 33 as well as in the tables describing
 4765 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
 4766 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 4767 Resource in both JSON and XML.

4768 **JSON media type:** application/json

4769 **JSON serialization:**

```

4770 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolEndpoint",
4771   "id": string,
4772   "name": string, ?
4773   "description": string, ?
4774   "created": string, ?
4775   "updated": string, ?
4776   "properties": { string: string, + }, ?
4777   "state": string,
4778   "protocol": string, ?
4779   "address": string,
4780   "origin": string,
4781   "interface": { "href": string },
4782   .."parameters": { string: string, + }, ?
4783   "meters": { "href": string }, ?
4784   "eventLog": { "href": string }, ?
4785   "operations": [
4786     { "rel": "edit", "href": string }, ?
4787     { "rel": "delete", "href": string }, ?
4788     { "rel": "http://schemas.dmtf.org/cimi/2/action/enable", "href": string },
4789     ?
4790     { "rel": "http://schemas.dmtf.org/cimi/2/action/disable", "href": string }
4791     ?
4792   ] ?
4793   ...
    
```

4794

```
} 
```

4795 **XML media type:** application/xml

4796 **XML serialization:**

4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819

```
<ProtocolEndpoint xmlns="http://schemas.dmtf.org/cimi/2">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <protocol> xs:string </protocol> ?
  <address> xs:string </address>
  <origin> xs:string </origin>
  <interface href="xs:anyURI"/>
  <parameters key="xs:string"> xs:string </parameters> *
  <meters href="xs:anyURI"/> ?
  <eventLog href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/2/action/enable"
href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/2/action/disable"
href="xs:anyURI"/> ?
  <xs:any>*
</ProtocolEndpoint>
```

4820 **5.16.9.1 Protocol specific parameters**

4821 Each Endpoint may require additional data that is specific to the communication protocol supported. This
4822 additional data is specified using the `parameters` attribute of a `ProtocolEndpoint`. This
4823 specification defines the following key – value pairs that provide supplemental information for Endpoints
4824 of specific protocol types:

4825 **Table 34 - IPv6 ProtocolEndpoint parameters**

Name	IPv6ProtocolEndpointParameters	
Key	Value Type	Description
addressType	string	The IPv6 address type as specified by RFC4291 , Section 2.4. Allowed values: [Unspecified Loopback Multicast Link Local Unicast Global Unicast Embedded IPv4 Address Site Local Unicast] If specified this value must match the type of address specified by the <code>address</code> attribute of the IPv6 Endpoint with which it is associated. Constraints: Provider: support optional; immutable Consumer: support optional; read-only

prefixLength	<i>integer</i>	The length of the prefix for IPv6 addresses that is used to specify a subnet. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
--------------	----------------	--

 4826 **Table 35 – IPv4 ProtocolEndpoint parameters**

Name	IPv4ProtocolEndpointParameters	
Key	Value Type	Description
hostname	<i>string</i>	The DNS resolvable name associated with this address. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

 4827 **Table 36 – Ethernet ProtocolEndpoint parameters**

Name	EthernetProtocolEndpointParameters	
Key	Value Type	Description
aliases	<i>string[]</i>	Other unicast addresses that may be used to communicate with the Endpoint Constraints: Provider: support optional; mutable Consumer: support optional; read-write
groupAddresses	<i>string[]</i>	Multicast addresses to which the Endpoint listens. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

4828 Note that Providers may support additional key – value pairs for the `parameter` attribute to extend the
4829 existing protocols. Consumers are not required to process any additional key – value pairs but must
4830 retrun them to the Provider in the serialization of `ProtocolEndpoints`.

4831 5.16.9.2 Collections

4832 The following clauses describe the Collection Resources that are components of
4833 `ProtocolEndpoints`.

4834 5.16.9.2.1 meters Collection

4835 The Resource type for each item of this Collection is “Meter” as defined in clause 5.17.3. There is no
4836 accessory attribute for the items in this Collection, therefore it is a basic `Meter` Collection (serialized as
4837 described in 5.5.12).

4838 5.16.9.3 Operations

4839 The `ProtocolEndpoints` Resource supports the Read, Update, and Delete operations. Create is
4840 supported through the `ProtocolEndpointCollection` Resource.

4841 Deleting a `ProtocolEndpoint` shall remove that Endpoint from the global (Cloud Entry Point)
4842 `ProtocolEndpointCollection`. Additionally, references to the Endpoint in
4843 `ProtocolEndpointCollections` of all other Resources (e.g. `ProtocolSegments`,
4844 `NetworkServices`) must be removed.

4845 The following custom operations are also defined:

4846 **enable**

4847 **/link@rel:** <http://schemas.dmtf.org/cimi/2/action/enable>

4848 This operation shall enable a `ProtocolEndpoint`.

4849 Input parameters: None.

4850 Output parameters: None.

4851 Upon successful completion of this operation, the `ProtocolEndpoint` shall be in the "ENABLED"
4852 state.

4853 **HTTP protocol**

4854 To enable a `ProtocolEndpoint`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/enable"
4855 URI of the `ProtocolEndpoint` where the HTTP request body shall be as described below.

4856 **JSON media type:** application/json

4857 **JSON serialization:**

```
4858 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
4859   "action": "http://schemas.dmtf.org/cimi/2/action/enable",
4860   "properties": { string: string, + } ?
4861   ...
4862 }
```

4863 **XML media type:** application/xml

4864 **XML serialization**

```
4865 <Action xmlns="http://schemas.dmtf.org/cimi/2">
4866   <action> http://schemas.dmtf.org/cimi/2/action/enable </action>
4867   <property key="xs:string"> xs:string </property> *
4868   <xs:any>*
4869 </Action>
```

4870 Upon successful processing of the request, the HTTP response body may be empty.

4871 **disable**

4872 **/link@rel:** http://schemas.dmtf.org/cimi/2/action/disable

4873 This operation shall disable a `ProtocolEndpoint`.

4874 Input parameters: None.

4875 Output parameters: None.

4876 Upon successful completion of this operation, the `ProtocolEndpoint` shall be in the "DISABLED"
4877 state.

4878 **HTTP protocol**

4879 To stop a `ProtocolEndpoint`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/disable"
4880 URI of the `ProtocolEndpoint` where the HTTP request body shall be as described below.

4881 **JSON media type:** application/json

4882 **JSON serialization:**

```
4883 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
4884     "action": "http://schemas.dmtf.org/cimi/2/action/disable",
4885     "properties": { string: string, + } ?
4886     ...
4887 }
```

4888 **XML media type:** application/xml

4889 **XML serialization**

```
4890 <Action xmlns="http://schemas.dmtf.org/cimi/2">
4891     <action> http://schemas.dmtf.org/cimi/2/action/disable </action>
4892     <property key="xs:string"> xs:string </property> *
4893     <xs:any>*
4894 </Action>
```

4895 Upon successful processing of the request, the HTTP response body may be empty.

4896 **5.16.10 ProtocolEndpointCollection Resource**

4897 A ProtocolEndpointCollection Resource represents the Collection of ProtocolEndpoints
 4898 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
 4899 serialized as follows:

4900 **JSON serialization:**

```
4901 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolEndpointCollection",
4902     "id": string,
4903     "count": number,
4904     "endpoints": [
4905         { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolEndpoint",
4906             "id": string,
4907             ... remaining ProtocolEndpoint attributes ...
4908         }, +
4909     ], ?
4910     "operations": [ { "rel": "add", "href": string } ? ]
4911     ...
4912 }
```

4913 **XML serialization:**

```
4914 <Collection
4915 resourceURI="http://schemas.dmtf.org/cimi/2/ProtocolEndpointCollection"
4916     xmlns="http://schemas.dmtf.org/cimi/2">
4917     <id> xs:anyURI </id>
4918     <count> xs:integer </count>
4919     <ProtocolEndpoint>
```

```

4920     <id> xs:anyURI </id>
4921     ... remaining ProtocolEndpoint attributes ...
4922 </ProtocolEndpoint> *
4923 <operation rel="add" href="xs:anyURI"/> ?
4924 <xs:any>*
4925 </Collection>
    
```

4926 **5.16.10.1 Operations**

4927 NOTE The "add" operation requires that a ProtocolEndpointTemplate be used (see clause 5.16.11).

4928 Upon successful processing of the "add" operation, unless otherwise specified by the
 4929 ProtocolEndpointTemplate "initialState" attribute, the state of the new ProtocolEndpoint
 4930 shall be the value of the DefaultInitialState capability of the ProtocolEndpoint Resource's
 4931 ResourceMetadata, if defined. If no DefaultInitialState capability is defined, the default value shall be
 4932 "DISABLED." The semantics of "initialState" shall be equivalent to the Provider issuing the appropriate
 4933 actions against the new ProtocolEndpoint to move it into that state.

4934 If a Provider is unable to change the state of the new ProtocolEndpoint to the appropriate
 4935 "initialState" (either as specified by the ProtocolEndpointTemplate or as implied by the previous
 4936 stated rules), the ProtocolEndpoint creation shall fail.

4937 **5.16.11 ProtocolEndpointTemplate Resource**

4938 The ProtocolEndpointTemplate is a set of configuration values for realizing a
 4939 ProtocolEndpoint. A ProtocolEndpointTemplate may be used to create multiple
 4940 ProtocolEndpoints. Table 37 Table 32 describes the ProtocolEndpointTemplate attributes.

4941 **Table 37 – ProtocolEndpointTemplate attributes**

Name	ProtocolEndpointTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/ProtocolEndpointTemplate	
Attribute	Type	Description
initialState	string	Sets the initial state of the Endpoint created using this Template. The allowed values are restricted to the non-transient states specified for the state attribute of the ProtocolEndpoint Resource, described in clause 5.16.9. Providers should advertise the list of available values via the ProtocolEndpoint ResourceMetadata initialStates Capability. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
address	string	If the origin attribute value is "STATIC" this attribute contains the address to be assigned to this Endpoint in the format required by the supported protocol. If the origin attribute value is "DYNAMIC" this attribute must not be supplied by the Template. Constraints: Provider: support optional; mutable Consumer: support optional; read-write

Name	ProtocolEndpointTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/ProtocolEndpointTemplate	
Attribute	Type	Description
origin	<i>string</i>	A string representing how protocol specific data is assigned to this Endpoint. Allowable values are: [STATIC DYNAMIC] If the value of this attribute is "STATIC" then all protocol specific data for the Endpoint must be supplied by this Template. If the value of this attribute is "DYNAMIC" then the protocol specific data for this Endpoint is allocated by other mechanisms and must not be supplied by this Template. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
interface	<i>Network Interface</i>	A reference to a <i>NetworkInterface</i> Resource with which this new Endpoint is associated. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
parameters	<i>map</i>	A polymorphic attribute the contents of which depend on the specific protocol supported. The allowed key – value pairs are as specified in clause 5.16.9. Whether this data is required to be supplied by this Template is determined by the value of the "origin" attribute described above. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; 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>ProtocolEndpoint</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>ProtocolEndpoint</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

4942 When implementing or using *ProtocolEndpointTemplate* Resources, Providers and Consumers
 4943 shall adhere to the syntax and semantics of its attributes as described in Table 37 as well as in the tables
 4944 describing embedded Resources or related Collections. Both Consumer and Provider shall serialize this
 4945 Resource as described below. The following pseudo-schemas (see notation in 1.3) describe the
 4946 serialization of the Resource in both JSON and XML.

4947 **JSON media type:** application/json

4948 **JSON serialization:**

```

4949 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolEndpointTemplate",
4950   "id": string,
4951   "name": string, ?
4952   "description": string, ?
4953   "created": string, ?
4954   "updated": string, ?
4955   "properties": { string: string, + }, ?
    
```

```

4956     "initialState": string, ?
4957     "address": string, ?
4958     "origin": string,
4959     "interface": { "href": string }, ?
4960 .."parameters": { string: string, + }, ?
4961     "meterTemplates": [
4962         { "href": string, ?
4963           ... MeterTemplate attributes ... ?
4964         }, *
4965     ], ?
4966     "eventLogTemplate": {
4967         "href": string, ?
4968         ... EventLogTemplate attributes ... ?
4969     }, ?
4970     "operations": [
4971         { "rel": "edit", "href": string }, ?
4972         { "rel": "delete", "href": string } ?
4973     ] ?
4974     ...
4975 }

```

4976 **XML media type:** application/xml

4977 **XML serialization:**

```

4978 <ProtocolEndpointTemplate xmlns="http://schemas.dmtf.org/cimi/2">
4979     <id> xs:anyURI </id>
4980     <name> xs:string </name> ?
4981     <description> xs:string </description> ?
4982     <created> xs:dateTime </created> ?
4983     <updated> xs:dateTime </updated> ?
4984     <property key="xs:string"> xs:string </property> *
4985     <initialState> xs:string </initialState> ?
4986     <address> xs:string </address> ?
4987     <origin> xs:string </origin>
4988     <interface href="xs:anyURI"/> ?
4989     <parameters key="xs:string"> xs:string </parameters> *
4990     <meterTemplate href="xs:anyURI"? >
4991         ... MeterTemplate attributes ... ?
4992     </meterTemplate> *
4993     <eventLogTemplate href="xs:anyURI"? >
4994         ... EventLogTemplate attributes ... ?

```

```

4995     </eventLogTemplate> ?
4996     <operation rel="edit" href="xs:anyURI"/> ?
4997     <operation rel="delete" href="xs:anyURI"/> ?
4998     <xs:any>*
4999 </ProtocolEndpointTemplate>
    
```

5000 **5.16.11.1 Collections**

5001 The ProtocolEndpointTemplate Resource has no attributes of type Collection.

5002 **5.16.11.2 Operations**

5003 The ProtocolEndpointTemplate Resource supports the Read, Update, and Delete operations.
 5004 Create is supported through the ProtocolEndpointTemplateCollection Resource.

5005 **5.16.12 ProtocolEndpointTemplateCollection Resource**

5006 A ProtocolEndpointTemplateCollection Resource represents the Collection of
 5007 ProtocolEndpointTemplates within a Provider and follows the Collection pattern defined in
 5008 clause 5.5.12. This Resource shall be serialized as follows:

5009 **JSON serialization:**

```

5010 { "resourceURI":
5011     "http://schemas.dmtf.org/cimi/2/ProtocolEndpointTemplateCollection",
5012     "id": string,
5013     "count": number,
5014     "protocolSegmentTemplates": [
5015         { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolEndpointTemplate",
5016           "id": string,
5017           ... remaining ProtocolEndpointTemplate attributes ...
5018         }, +
5019     ], ?
5020     "operations": [ { "rel": "add", "href": string } ? ]
5021     ...
5022 }
    
```

5023 **XML serialization:**

```

5024 <Collection
5025     resourceURI="http://schemas.dmtf.org/cimi/2/ProtocolEndpointTemplateCollection"
5026     xmlns="http://schemas.dmtf.org/cimi/2">
5027     <id> xs:anyURI </id>
5028     <count> xs:integer </count>
5030     <ProtocolEndpointTemplate>
5031         <id> xs:anyURI </id>
5032         ... remaining ProtocolEndpointTemplate attributes ...
    
```

```

5033     </ProtocolEndpointTemplate> *
5034     <operation rel="add" href="xs:anyURI"/> ?
5035     <xs:any>*
5036 </Collection>
    
```

5037 **5.16.12.1 Operations**

5038 The ProtocolEndpointTemplateCollection Resource supports the Read and Update
 5039 operations. Creation of new ProtocolEndpointTemplate Resources is supported by the way of a
 5040 POST to the "add" operation's URI as described in clause 4.2.1.1.

5041 **5.16.13 Interfaces**

5042 An Interface provides a connection to a Network by associating Endpoints with Machines. The model is
 5043 basically that of a virtual Network Interface Card (vNIC) that can support multiple communication
 5044 protocols at multiple levels. Interfaces are NetworkInterface Resources, the attributes of which are
 5045 described in Table 38 below.

5046 **Table 38 – NetworkInterface attributes**

Name	NetworkInterface	
Type URI	http://schemas.dmtf.org/cimi/2/NetworkInterface	
Attribute	Type	Description
state	<i>string</i>	The operational state of the Interface. Allowable values are: CREATING: The Interface is in the process of being created. ENABLED: The Interface is available and ready for use. DISABLED: The Interface is not available for use. DELETING: The Interface is in the process of being deleted. ERROR: The Provider has detected an error in the Interface. The operations that result in transitions to the above defined states are defined in clause 5.16.13.2. Clause 5.16.14.1 defines the initial state of a Interface. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
endpoints	<i>collection</i> [Protocol Endpoint]	A reference to a list of references to ProtocolEndpoints this Interface supports. Note: This Collection represents an association between the Interface and a list of Endpoints in one or more Segments. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
speed	<i>integer</i>	The current bandwidth of the Interface in Bits per Second. For Interfaces that vary in bandwidth or for those where no accurate estimation can be made, this attribute should contain the nominal bandwidth.. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
mtu	<i>integer</i>	The size in bytes of the active or negotiated maximum transmission unit (MTU) that can be supported by this Interface. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
meters	<i>collection</i> [Meter]	A reference to the list of Meters monitored for this Interface. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

Name	NetworkInterface	
Type URI	http://schemas.dmtf.org/cimi/2/NetworkInterface	
Attribute	Type	Description
eventLog	ref	A reference to the EventLog of this Interface. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

5047 When implementing or using `NetworkInterface`, Providers and Consumers shall adhere to the
 5048 syntax and semantics of its attributes as described in Table 38 as well as in the tables describing
 5049 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
 5050 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 5051 Resource in both JSON and XML.

5052 **JSON media type:** application/json

5053 **JSON serialization:**

```
5054 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkInterface",
5055   "id": string,
5056   "name": string, ?
5057   "description": string, ?
5058   "created": string, ?
5059   "updated": string, ?
5060   "properties": { string: string, + }, ?
5061   "state": string,
5062   "endpoints": { "href": string }, ?
5063   "speed": number, ?
5064   "mtu": number ?,
5065   "meters": { "href": string }, ?
5066   "eventLog": { "href": string }, ?
5067   "operations": [
5068     { "rel": "edit", "href": string }, ?
5069     { "rel": "delete", "href": string }, ?
5070     { "rel": "http://schemas.dmtf.org/cimi/2/action/enable", "href": string },
5071     ?
5072     { "rel": "http://schemas.dmtf.org/cimi/2/action/disable", "href": string }
5073     ?
5074   ] ?
5075   ...
5076 }
```

5077 **XML media type:** application/xml

5078 **XML serialization:**

```
5079 <NetworkInterface xmlns="http://schemas.dmtf.org/cimi/2">
5080   <id> xs:anyURI </id>
5081   <name> xs:string </name> ?
```

```

5082     <description> xs:string </description> ?
5083     <created> xs:dateTime </created> ?
5084     <updated> xs:dateTime </updated> ?
5085     <property key="xs:string"> xs:string </property> *
5086     <state> xs:string </state>
5087     <endpoint href="xs:anyURI"/> ?
5088     <speed> xs:integer </speed> ?
5089     <mtu> xs:integer </mtu> ?
5090     <meters href="xs:anyURI"/> ?
5091     <eventLog href="xs:anyURI"/> ?
5092     <operation rel="edit" href="xs:anyURI"/> ?
5093     <operation rel="delete" href="xs:anyURI"/> ?
5094     <operation rel="http://schemas.dmtf.org/cimi/2/action/enable"
5095     href="xs:anyURI"/> ?
5096     <operation rel="http://schemas.dmtf.org/cimi/2/action/disable"
5097     href="xs:anyURI"/> ?
5098     <xs:any>*
5099 </NetworkInterface>

```

5100 5.16.13.1 Collections

5101 The following clauses describe the Collection Resources that are components of
5102 NetworkInterfaces.

5103 5.16.13.1.1 meters Collection

5104 The Resource type for each item of this Collection is “Meter” as defined in clause 5.17.3. There is no
5105 accessory attribute for the items in this Collection, therefore it is a basic Meter Collection (serialized as
5106 described in 5.5.12).

5107 5.16.13.2 Operations

5108 The NetworkInterfaces Resource supports the Read, Update, and Delete operations. Create is
5109 supported through the NetworkInterfaceCollection Resource.

5110 Deleting a NetworkInterface shall remove that Endpoint from the global (Cloud Entry Point)
5111 NetworkInterfaceCollection. Additionally, references to the Endpoint in
5112 NetworkInterfaceCollections of all other Resources (e.g. ProtocolEndpoints,
5113 NetworkServices) must be removed.

5114 The following custom operations are also defined:

5115 enable

5116 **/link@rel:** http://schemas.dmtf.org/cimi/2/action/enable

5117 This operation shall enable a NetworkInterface.

5118 Input parameters: None.

5119 Output parameters: None.

5120 Upon successful completion of this operation, the `NetworkInterface` shall be in the "ENABLED"
 5121 state.

5122 **HTTP protocol**

5123 To enable a `NetworkInterface`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/enable"
 5124 URI of the `NetworkInterface` where the HTTP request body shall be as described below.

5125 **JSON media type:** application/json

5126 **JSON serialization:**

```
5127 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
5128   "action": "http://schemas.dmtf.org/cimi/2/action/enable",
5129   "properties": { string: string, + } ?
5130   ...
5131 }
```

5132 **XML media type:** application/xml

5133 **XML serialization**

```
5134 <Action xmlns="http://schemas.dmtf.org/cimi/2">
5135   <action> http://schemas.dmtf.org/cimi/2/action/enable </action>
5136   <property key="xs:string"> xs:string </property> *
5137   <xs:any>*
5138 </Action>
```

5139 Upon successful processing of the request, the HTTP response body may be empty.

5140 **disable**

5141 **/link@rel:** http://schemas.dmtf.org/cimi/2/action/disable

5142 This operation shall disable a `NetworkInterface`.

5143 Input parameters: None.

5144 Output parameters: None.

5145 Upon successful completion of this operation, the `NetworkInterface` shall be in the "DISABLED"
 5146 state.

5147 **HTTP protocol**

5148 To stop a `NetworkInterface`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/disable"
 5149 URI of the `NetworkInterface` where the HTTP request body shall be as described below.

5150 **JSON media type:** application/json

5151 **JSON serialization:**

```
5152 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
5153   "action": "http://schemas.dmtf.org/cimi/2/action/disable",
5154   "properties": { string: string, + } ?
5155   ...
```

5156 }

5157 **XML media type:** application/xml

5158 **XML serialization**

```
5159     <Action xmlns="http://schemas.dmtf.org/cimi/2">
5160         <action> http://schemas.dmtf.org/cimi/2/action/disable </action>
5161         <property key="xs:string"> xs:string </property> *
5162         <xs:any>*
5163     </Action>
```

5164 Upon successful processing of the request, the HTTP response body may be empty.

5165 **5.16.14 NetworkInterfaceCollection Resource**

5166 A `NetworkInterfaceCollection` Resource represents the Collection of `NetworkInterfaces`
5167 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
5168 serialized as follows:

5169 **JSON serialization:**

```
5170 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkInterfaceCollection",
5171   "id": string,
5172   "count": number,
5173   "interfaces": [
5174     { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkInterface",
5175       "id": string,
5176       ... remaining NetworkInterface attributes ...
5177     }, +
5178   ], ?
5179   "operations": [ { "rel": "add", "href": string } ? ]
5180   ...
5181 }
```

5182 **XML serialization:**

```
5183 <Collection
5184 resourceURI="http://schemas.dmtf.org/cimi/2/NetworkInterfaceCollection"
5185   xmlns="http://schemas.dmtf.org/cimi/2">
5186   <id> xs:anyURI </id>
5187   <count> xs:integer </count>
5188   <NetworkInterface>
5189     <id> xs:anyURI </id>
5190     ... remaining NetworkInterface attributes ...
5191   </NetworkInterface> *
5192   <operation rel="add" href="xs:anyURI"/> ?
5193   <xs:any>*
5194 </Collection>
```

5195 **5.16.14.1 Operations**

5196 NOTE The "add" operation requires that a `NetworkInterfaceTemplate` be used (see clause 5.16.15).

5197 Upon successful processing of the "add" operation, unless otherwise specified by the
 5198 `NetworkInterfaceTemplate` "initialState" attribute, the state of the new `NetworkInterface`
 5199 shall be the value of the `DefaultInitialState` capability of the `NetworkInterface` Resource's
 5200 `ResourceMetadata`, if defined. If no `DefaultInitialState` capability is defined, the default value shall be
 5201 "DISABLED." The semantics of "initialState" shall be equivalent to the Provider issuing the appropriate
 5202 actions against the new `NetworkInterface` to move it into that state.

5203 If a Provider is unable to change the state of the new `NetworkInterface` to the appropriate
 5204 "initialState" (either as specified by the `NetworkInterfaceTemplate` or as implied by the previous
 5205 stated rules), the `NetworkInterface` creation shall fail.

5206 **5.16.15 NetworkInterfaceTemplate Resource**

5207 The `NetworkInterfaceTemplate` is a set of configuration values for realizing a
 5208 `NetworkInterface`. A `NetworkInterfaceTemplate` may be used to create multiple
 5209 `NetworkInterfaces`. Table 39 describes the `NetworkInterfaceTemplate` attributes.

5210 **Table 39 – NetworkInterfaceTemplate attributes**

Name	NetworkInterfaceTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/NetworkInterfaceTemplate	
Attribute	Type	Description
initialState	<i>string</i>	Sets the initial state of the Endpoint created using this Template. The allowed values are restricted to the non-transient states specified for the state attribute of the <code>NetworkInterface</code> Resource, described in 5.16.13 . Providers should advertise the list of available values via the <code>NetworkInterface</code> <code>ResourceMetadata</code> <code>initialStates</code> Capability. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
endpoints	<i>collection [Protocol Endpoint]</i>	A reference to a list of references to <code>ProtocolEndpoints</code> this Interface supports. Note: This Collection represents an association between the Interface and a list of Endpoints in one or more Segments. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
speed	<i>integer</i>	The initial bandwidth of the Interface in Bits per Second. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
mtu	<i>integer</i>	The size in bytes of the initial maximum transmission unit (MTU) that can be supported by this Interface. Constraints: Provider: support optional; mutable Consumer: support optional; 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>NetworkInterface</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

Name	NetworkInterfaceTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/NetworkInterfaceTemplate	
Attribute	Type	Description
eventLogTemplate	<i>ref</i>	<p>A reference to an <code>EventLogTemplate</code> that shall be used to create and connect a new <code>EventLog</code> to the new <code>NetworkInterface</code>.</p> <p>Note that the attributes of the <code>EventLogTemplate</code> may be specified rather than a reference to an existing <code>EventLogTemplate</code> Resource.</p> <p>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</p>

5211 When implementing or using `NetworkInterfaceTemplate` Resources, Providers and Consumers
5212 shall adhere to the syntax and semantics of its attributes as described in Table 39 as well as in the tables
5213 describing embedded Resources or related Collections. Both Consumer and Provider shall serialize this
5214 Resource as described below. The following pseudo-schemas (see notation in 1.3) describe the
5215 serialization of the Resource in both JSON and XML.

5216 **JSON media type:** application/json

5217 **JSON serialization:**

```

5218 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkInterfaceTemplate",
5219   "id": string,
5220   "name": string, ?
5221   "description": string, ?
5222   "created": string, ?
5223   "updated": string, ?
5224   "properties": { string: string, + }, ?
5225   "initialState": string, ?
5226   "endpoints": { "href": string }, ?
5227   "speed": number, ?
5228   "mtu": number ?,
5229   "meterTemplates": [
5230     { "href": string, ?
5231       ... MeterTemplate attributes ... ?
5232     }, *
5233   ], ?
5234   "eventLogTemplate": {
5235     "href": string, ?
5236     ... EventLogTemplate attributes ... ?
5237   }, ?
5238   "operations": [
5239     { "rel": "edit", "href": string }, ?
5240     { "rel": "delete", "href": string } ?
5241   ] ?
5242   ...
5243 }
```

5244 **XML media type:** application/xml

5245 **XML serialization:**

```

5246 <NetworkInterfaceTemplate xmlns="http://schemas.dmtf.org/cimi/2">
5247   <id> xs:anyURI </id>
5248   <name> xs:string </name> ?
5249   <description> xs:string </description> ?
5250   <created> xs:dateTime </created> ?
5251   <updated> xs:dateTime </updated> ?
5252   <property key="xs:string"> xs:string </property> *
5253   <initialState> xs:string </initialState> ?
5254   <endpoint href="xs:anyURI"/> ?
5255   <speed> xs:integer </speed> ?
5256   <mtu> xs:integer </mtu> ?
5257   <meterTemplate href="xs:anyURI"? >
5258     ... MeterTemplate attributes ... ?
5259   </meterTemplate> *
5260   <eventLogTemplate href="xs:anyURI"? >
5261     ... EventLogTemplate attributes ... ?
5262   </eventLogTemplate> ?
5263   <operation rel="edit" href="xs:anyURI"/> ?
5264   <operation rel="delete" href="xs:anyURI"/> ?
5265   <xs:any>*
5266 </NetworkInterfaceTemplate>

```

5267 5.16.15.1 Collections

5268 The following clauses describe Collection Resources that are components of
 5269 NetworkInterfaceTemplates.

5270 5.16.15.1.1 endpoints Collection

5271 The Resource type for each item of this Collection is "ProtocolEndpoint" as defined in clause
 5272 5.16.9. There is no accessory attribute for the items in this Collection, therefore it is a basic
 5273 ProtocolEndpointCollection (serialized as described in 5.16.10).

5274 5.16.15.2 Operations

5275 The NetworkInterfaceTemplate Resource supports the Read, Update, and Delete operations.
 5276 Create is supported through the NetworkInterfaceTemplateCollection Resource.

5277 5.16.16 NetworkInterfaceTemplateCollection Resource

5278 A NetworkInterfaceTemplateCollection Resource represents the Collection of
 5279 NetworkInterfaceTemplates within a Provider and follows the Collection pattern defined in
 5280 clause 5.5.12. This Resource shall be serialized as follows:

5281 **JSON serialization:**

```

5282 { "resourceURI":
5283     "http://schemas.dmtf.org/cimi/2/NetworkInterfaceTemplateCollection",
5284     "id": string,
5285     "count": number,
5286     "protocolSegmentTemplates": [
5287         { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkInterfaceTemplate",
5288           "id": string,
5289           ... remaining NetworkInterfaceTemplate attributes ...
5290         }, +
5291     ], ?
5292     "operations": [ { "rel": "add", "href": string } ? ]
5293     ...
5294 }

```

5295 **XML serialization:**

```

5296 <Collection
5297
5298 resourceURI="http://schemas.dmtf.org/cimi/2/NetworkInterfaceTemplateCollection"
5299     xmlns="http://schemas.dmtf.org/cimi/2">
5300     <id> xs:anyURI </id>
5301     <count> xs:integer </count>
5302     <NetworkInterfaceTemplate>
5303         <id> xs:anyURI </id>
5304         ... remaining NetworkInterfaceTemplate attributes ...
5305     </NetworkInterfaceTemplate> *
5306     <operation rel="add" href="xs:anyURI"/> ?
5307     <xs:any>*
5308 </Collection>

```

5309 **5.16.16.1 Operations**

5310 The `NetworkInterfaceTemplateCollection` Resource supports the Read and Update
 5311 operations. Creation of new `NetworkInterfaceTemplate` Resources is supported by the way of a
 5312 POST to the "add" operation's URI as described in clause 4.2.1.1.

5313 **5.16.17 Services**

5314 Services provide all additional functionality within Networks beyond basic routing within a single
 5315 Segment. Services can be applied to individual Segments or Endpoints, collections of Segments or
 5316 Endpoints, or combinations of these elements. The actual function provide by a Service is determined by
 5317 policies (see clause 5.16.21). Services are `NetworkService` Resources, the attributes of which are
 5318 described in Table 40 below **Error! Reference source not found..**

5319

Table 40 – NetworkService attributes

Name	NetworkService	
Type URI	http://schemas.dmtf.org/cimi/2/NetworkService	
Attribute	Type	Description
state	<i>string</i>	The operational state of the Service. Allowed values are: CREATING: The Service is in the process of being created. STARTED: The Service is available (enabled) and ready for use. STOPPED: The Service is stopped (disabled) and not available for use. DELETING: The Service is in the process of being deleted. ERROR: The Provider has detected an error in the Service. The operations that result in transitions to the above defined states are defined in clause 5.17. Clause 5.16.18.1 defines the initial state of a Service. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
type	<i>string</i>	Indicates the type of service provided by this <i>NetworkService</i> . Allowed values: [Load Balancer QoS Firewall VPN DHCP DNS NAT Gateway Layer4 Port Forwarding IP Routing Virtual Network Device Other] Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
endpoints	<i>collection</i> [Protocol Endpoint]	A reference to a list of references to individual Endpoints to which the Service is provided. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
segments	<i>collection</i> [Protocol Segment]	A reference to a list of references to complete Segments to which the service is provided. The Service is provided to all Endpoints within each Segment. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
policies	<i>map</i>	*** TBD *** Format & requirements yet to be determined from NSMWG work Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
meters	<i>collection</i> [Meter]	A reference to the list of <i>Meters</i> monitored for this Service. Constraints: Provider: support optional; mutable Consumer: support optional; read-only
eventLog	<i>ref</i>	A reference to the <i>EventLog</i> of this Service. Constraints: Provider: support optional; mutable Consumer: support optional; read-only

5320 When implementing or using *NetworkService* Resources, Providers and Consumers shall adhere to
 5321 the syntax and semantics of its attributes as described in Table 40 as well as in the tables describing
 5322 embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource
 5323 as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
 5324 Resource in both JSON and XML.

5325 **JSON media type:** application/json

5326 **JSON serialization:**

```
5327 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkService",
5328   "id": string,
5329   "name": string, ?
```

```

5330     "description": string, ?
5331     "created": string, ?
5332     "updated": string, ?
5333     "properties": { string: string, + }, ?
5334     "state": string,
5335     "type": string,
5336     "endpoints": { "href": string }, ?
5337     "segments": { "href": string }, ?
5338
5339     .."policies": { string: string, + }, ?
5340
5341     "meters": { "href": string }, ?
5342     "eventLog": { "href": string }, ?
5343     "operations": [
5344         { "rel": "edit", "href": string }, ?
5345         { "rel": "delete", "href": string }, ?
5346         { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string }, ?
5347         { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string } ?
5348     ] ?
5349     ...
5350 }

```

5351 **XML media type:** application/xml

5352 **XML serialization:**

```

5353 <NetworkService xmlns="http://schemas.dmtf.org/cimi/2">
5354     <id> xs:anyURI </id>
5355     <name> xs:string </name> ?
5356     <description> xs:string </description> ?
5357     <created> xs:dateTime </created> ?
5358     <updated> xs:dateTime </updated> ?
5359     <property key="xs:string"> xs:string </property> *
5360     <state> xs:string </state>
5361     <type> xs:string </type>
5362     <endpoints href="xs:anyURI"/> *
5363     <segments href="xs:anyURI"/> *
5364
5365     <policies key="xs:string"> xs:string </policies> *
5366
5367     <meters href="xs:anyURI"/> ?
5368     <eventLog href="xs:anyURI"/> ?

```

```

5369     <operation rel="edit" href="xs:anyURI"/> ?
5370     <operation rel="delete" href="xs:anyURI"/> ?
5371     <operation rel="http://schemas.dmtf.org/cimi/2/action/start"
5372 href="xs:anyURI"/> ?
5373     <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
5374 href="xs:anyURI"/> ?
5375     <xs:any>*
5376 </NetworkService>

```

5377 **5.16.17.1 Collections**

5378 The following clauses describe the Collection Resources that are components of `NetworkServices`.

5379 **5.16.17.1.1 endpoints Collection**

5380 The Resource type for each item of this Collection is a “`ProtocolEndpoint`” as defined in clause
5381 5.16.9. There is no accessory attribute for the items in this Collection, therefore it is a basic
5382 `ProtocolEndpointCollection` Resource, serialized as described in 5.16.10.

5383 **5.16.17.1.2 segments Collection**

5384 The Resource type for each item of this Collection is a “`ProtocolSegment`” as defined in clause
5385 5.16.55.16.9. There is no accessory attribute for the items in this Collection, therefore it is a basic
5386 `ProtocolSegmentCollection` Resource, serialized as described in 5.16.6.

5387 **5.16.17.1.3 meters Collection**

5388 The Resource type for each item of this Collection is “`Meter`” as defined in clause 5.17.3. There is no
5389 accessory attribute for the items in this Collection, therefore it is a basic `Meter` Collection (serialized as
5390 described in 5.5.12).

5391 **5.16.17.2 Operations**

5392 The `NetworkService` Resource supports the Read, Update, and Delete operations. Create is
5393 supported through the `NetworkServiceCollection` Resource.

5394 Deleting a `NetworkService` shall remove that Service from the global (Cloud Entry Point)
5395 `NetworkServiceCollection` and also all references to the Service in Collections of other
5396 Resources (e.g. from corresponding `Network services` Collections).

5397 The following custom operations are also defined:

5398 **start**

5399 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/start`

5400 This operation shall start a `NetworkService`.

5401 Input parameters: None.

5402 Output parameters: None.

5403 Upon successful completion of this operation, the `NetworkService` shall be in the "STARTED" state.

5404 **HTTP protocol**

5405 To start a `NetworkService`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/start" URI of
 5406 the `NetworkService` where the HTTP request body shall be as described below.

5407 **JSON media type:** application/json

5408 **JSON serialization:**

```
5409 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
5410   "action": "http://schemas.dmtf.org/cimi/2/action/start",
5411   "properties": { string: string, + } ?
5412   ...
5413 }
```

5414 **XML media type:** application/xml

5415 **XML serialization**

```
5416 <Action xmlns="http://schemas.dmtf.org/cimi/2">
5417   <action> http://schemas.dmtf.org/cimi/2/action/start </action>
5418   <property key="xs:string"> xs:string </property> *
5419   <xs:any>*
5420 </Action>
```

5421 Upon successful processing of the request, the HTTP response body may be empty.

5422 **stop**

5423 **/link@rel:** http://schemas.dmtf.org/cimi/2/action/stop

5424 This operation shall stop a `NetworkService`.

5425 Input parameters: None.

5426 Output parameters: None.

5427 Upon successful completion of this operation, the `NetworkService` shall be in the "STOPPED" state.

5428 **HTTP protocol**

5429 To stop a `NetworkService`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/stop" URI of
 5430 the `NetworkService` where the HTTP request body shall be as described below.

5431 **JSON media type:** application/json

5432 **JSON serialization:**

```
5433 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
5434   "action": "http://schemas.dmtf.org/cimi/2/action/stop",
5435   "properties": { string: string, + } ?
5436   ...
5437 }
```

5438 **XML media type:** application/xml

5439 **XML serialization**

```
5440 <Action xmlns="http://schemas.dmtf.org/cimi/2">
```

```

5441     <action> http://schemas.dmtf.org/cimi/2/action/stop </action>
5442     <property key="xs:string"> xs:string </property> *
5443     <xs:any>*
5444 </Action>
    
```

5445 Upon successful processing of the request, the HTTP response body may be empty.

5446 **5.16.18 NetworkServiceCollection Resource**

5447 A `NetworkServiceCollection` Resource represents the Collection of `NetworkServices`
 5448 within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be
 5449 serialized as follows:

5450 **JSON serialization:**

```

5451 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkServiceCollection",
5452   "id": string,
5453   "count": number,
5454   "services": [
5455     { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkService",
5456       "id": string,
5457       ... remaining NetworkService attributes ...
5458     }, +
5459   ], ?
5460   "operations": [ { "rel": "add", "href": string } ? ]
5461   ...
5462 }
    
```

5463 **XML serialization:**

```

5464 <Collection
5465 resourceURI="http://schemas.dmtf.org/cimi/2/NetworkServiceCollection"
5466   xmlns="http://schemas.dmtf.org/cimi/2">
5467   <id> xs:anyURI </id>
5468   <count> xs:integer </count>
5469   <NetworkService>
5470     <id> xs:anyURI </id>
5471     ... remaining NetworkService attributes ...
5472   </NetworkService> *
5473   <operation rel="add" href="xs:anyURI"/> ?
5474   <xs:any>*
5475 </Collection>
    
```

5476 **5.16.18.1 Operations**

5477 **NOTE** The "add" operation requires that a `NetworkServiceTemplate` be used (see clause 5.16.19).

5478 Upon successful processing of the "add" operation, unless otherwise specified by the
 5479 `NetworkServiceTemplate` "initialState" attribute, the state of the new `NetworkService` shall be

5480 the value of the `DefaultInitialState` capability of the `NetworkService` Resource's
 5481 `ResourceMetadata`, if defined. If no `DefaultInitialState` capability is defined, the default value shall be
 5482 "STOPPED." The semantics of "initialState" shall be equivalent to the Provider issuing the appropriate
 5483 actions against the new `NetworkService` to move it into that state.

5484 If a Provider is unable to change the state of the new `NetworkService` to the appropriate "initialState"
 5485 (either as specified by the `NetworkServiceTemplate` or as implied by the previous stated rules),
 5486 the `NetworkService` creation shall fail.

5487 **5.16.19 NetworkServiceTemplate Resource**

5488 The `NetworkServiceTemplate` is a set of configuration values for realizing a `NetworkService`.
 5489 A `NetworkServiceTemplate` may be used to create multiple `NetworkServices`. Table 41
 5490 describes the `NetworkServiceTemplate` attributes.

5491 **Table 41 – NetworkServiceTemplate attributes**

Name	NetworkServiceTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/NetworkServiceTemplate	
Attribute	Type	Description
network	<i>ref</i>	A reference to the Network to which the Service created using this Template belongs. If this Template is used to create a new Service through the global (Cloud Entry Point) <code>NetworkServiceCollection</code> , this attribute shall be present. If this Template is referenced from a <code>NetworkTemplate</code> and used to create a new Service during the creation of a Network, this attribute shall either be absent or have the same value as the "id" attribute of the Network to which this Service is being added. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
initialState	<i>string</i>	Sets the initial state of the Service created using this Template. The allowed values are restricted to the non-transient states specified for the <code>state</code> attribute of the <code>NetworkService</code> Resource, described in clause 5.16.17. Providers should advertise the list of available values via the <code>NetworkServiceResourceMetadata initialStateStates</code> Capability. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
type	<i>string</i>	Sets the protocol supported by the Service created using this Template. The allowed values are those specified for the <code>protocol</code> attribute of the <code>NetworkService</code> Resource, described in 5.16.17 Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
endpoints	<i>ProtocolEndpoint[]</i>	A list of references to <code>ProtocolEndpoints</code> to be inserted into the <code>endpoints</code> Collection of the Service created using this Template. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
segments	<i>ProtocolSegment[]</i>	A list of references to <code>ProtocolSegments</code> to be inserted into the <code>segments</code> Collection of the Service created using this Template. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

Name	NetworkServiceTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/NetworkServiceTemplate	
Attribute	Type	Description
policies	<i>map</i>	*** TBD *** Format & requirements yet to be determined form NSMWG work Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
meterTemplates	<i>meterTemplates []</i>	A list of references to MeterTemplates that shall be used to create and connect a set of new Meters to the new NetworkService. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write
eventLogTemplate	<i>ref</i>	A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new NetworkService. 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

5492 When implementing or using NetworkServiceTemplate Resources, Providers and Consumers
 5493 shall adhere to the syntax and semantics of its attributes as described in Table 41Table 32 as well as in
 5494 the tables describing embedded Resources or related Collections. Both Consumer and Provider shall
 5495 serialize this Resource as described below. The following pseudo-schemas (see notation in 1.3) describe
 5496 the serialization of the Resource in both JSON and XML.

5497 **JSON media type:** application/json

5498 **JSON serialization:**

```

5499 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkServiceTemplate",
5500   "id": string,
5501   "name": string, ?
5502   "description": string, ?
5503   "created": string, ?
5504   "updated": string, ?
5505   "properties": { string: string, + }, ?
5506   "network": { "href": string }, ?
5507   "initialState": string, ?
5508   "type": string,
5509   "endpoints": { "href": string }, *
5510   "segments": { "href": string }, *
5511
5512   .."policies": { string: string, + }, ?
5513
5514   "meterTemplates": [
5515     { "href": string, ?
5516       ... MeterTemplate attributes ... ?
    
```

```

5517     }, *
5518   ], ?
5519   "eventLogTemplate": {
5520     "href": string, ?
5521     ... EventLogTemplate attributes ... ?
5522   }, ?
5523   "operations": [
5524     { "rel": "edit", "href": string }, ?
5525     { "rel": "delete", "href": string } ?
5526   ] ?
5527   ...
5528 }

```

5529 **XML media type:** application/xml

5530 **XML serialization:**

```

5531 <NetworkServiceTemplate xmlns="http://schemas.dmtf.org/cimi/2">
5532   <id> xs:anyURI </id>
5533   <name> xs:string </name> ?
5534   <description> xs:string </description> ?
5535   <created> xs:dateTime </created> ?
5536   <updated> xs:dateTime </updated> ?
5537   <property key="xs:string"> xs:string </property> *
5538   <network href="xs:anyURI"/> ?
5539   <initialState> xs:string </initialState> ?
5540   <type> xs:string </type>
5541   <endpoints href="xs:anyURI"/> *
5542   <segments href="xs:anyURI"/> *
5543
5544   <policies key="xs:string"> xs:string </policies> *
5545
5546   <meterTemplate href="xs:anyURI"? >
5547     ... MeterTemplate attributes ... ?
5548   </meterTemplate> *
5549   <eventLogTemplate href="xs:anyURI"? >
5550     ... EventLogTemplate attributes ... ?
5551   </eventLogTemplate> ?
5552   <operation rel="edit" href="xs:anyURI"/> ?
5553   <operation rel="delete" href="xs:anyURI"/> ?
5554   <xs:any>*
5555 </NetworkServiceTemplate>

```

5556 **5.16.19.1 Collections**

5557 The `NetworkServiceTemplate.Resource` has no attributes of type `Collection`.

5558 **5.16.19.2 Operations**

5559 The `NetworkServiceTemplate` Resource supports the Read, Update, and Delete operations.

5560 Create is supported through the `NetworkServiceTemplateCollection` Resource.

5561 **5.16.20 NetworkServiceTemplateCollection Resource**

5562 A `NetworkServiceTemplateCollection` Resource represents the Collection of
 5563 `NetworkServiceTemplates` within a Provider and follows the Collection pattern defined in clause
 5564 5.5.12. This Resource shall be serialized as follows:

5565 **JSON serialization:**

```
5566 { "resourceURI":
5567     "http://schemas.dmtf.org/cimi/2/NetworkServiceTemplateCollection",
5568     "id": string,
5569     "count": number,
5570     "protocolSegmentTemplates": [
5571         { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkServiceTemplate",
5572           "id": string,
5573           ... remaining NetworkServiceTemplate attributes ...
5574         }, +
5575     ], ?
5576     "operations": [ { "rel": "add", "href": string } ? ]
5577     ...
5578 }
```

5579 **XML serialization:**

```
5580 <Collection
5581
5582 resourceURI="http://schemas.dmtf.org/cimi/2/NetworkServiceTemplateCollection"
5583     xmlns="http://schemas.dmtf.org/cimi/2">
5584     <id> xs:anyURI </id>
5585     <count> xs:integer </count>
5586     <NetworkServiceTemplate>
5587         <id> xs:anyURI </id>
5588         ... remaining NetworkServiceTemplate attributes ...
5589     </NetworkServiceTemplate> *
5590     <operation rel="add" href="xs:anyURI"/> ?
5591     <xs:any>*
5592 </Collection>
```

5593 **5.16.20.1 Operations**

5594 The `NetworkServiceTemplateCollection` Resource supports the Read and Update
 5595 operations. Creation of new `NetworkServiceTemplate` Resources is supported by the way of a
 5596 POST to the "add" operation's URI as described in clause 4.2.1.1.

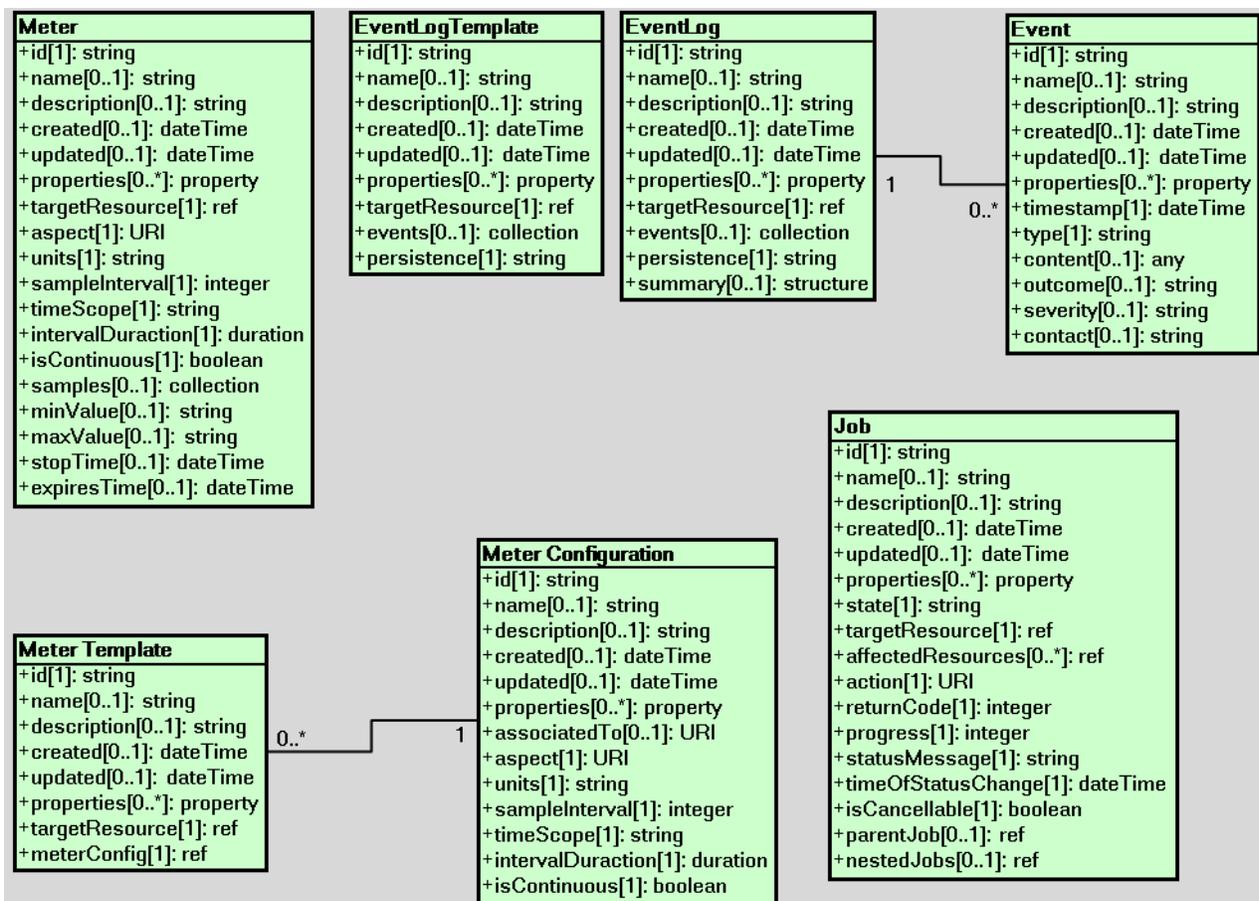
5597 **5.16.21 Policies**

5598 *** TBD ***

5599 *Format & requirements yet to be determined form NSMWG workError! Reference source not found..*

5600 **5.17 Monitoring Resources and relationships**

5601 Figure 6 illustrates the Resources involved in tracking the progress of operations, as well as, metering
 5602 and monitoring the status of other Resources. Although this drawing is in the style of a Resource



5603 Relationship diagram, the use of UML is neither rigorous nor normative.

5604 **Figure 6 - Monitoring Resources**

5605 **5.17.1 Job Resource**

5606 This Resource represents a process (i.e., a sequence of one or more operations directed to accomplish a
 5607 specific goal) that is performed by the Provider.

5608 If a Provider supports exposing Job Resources to Consumers, each request from a Consumer that the
 5609 Provider responds to with a 202 status code, shall result in a Job Resource being created and an
 5610 absolute URI reference to that Job Resource shall be made available to the requesting Consumer.
 5611 Providers may create additional Job Resources for Provider-initiated operations if the Provider chooses
 5612 to expose these Jobs to Consumers.

5613 If a Job is not completed successfully (e.g., it is in the FAILED or STOPPED state), this specification
 5614 does not place any requirements on the Provider to ensure that the affected Resources are left in certain
 5615 states. Based on the environmental conditions at that time, the Provider might choose to "undo" any
 5616 impact of the operation; simply halt processing; attempt some kind of "cleanup" action; or choose to do
 5617 something else. However, Providers shall list all Resources impacted by the Job in the
 5618 "affectedResources" attribute, thus allowing Consumers an opportunity to examine the state of each
 5619 Resource themselves. In cases where a Resource has been deleted, references to that Resource shall
 5620 not appear in the "affectedResources" attribute.

5621 The Job Resource allows for nesting of Jobs. The determination of when a single operation is
 5622 converted into multiple nested Jobs is out of scope of this specification. However, if there are nested
 5623 Jobs, the top-most Job Resource shall report the overall status of all Jobs and shall only be in a
 5624 "SUCCESS" state if all nested Jobs are also in "SUCCESS" state. If nested Jobs are created, there is
 5625 no requirement for the top-most Job Resource to reference all affected Resources in its
 5626 "affectedResources" attribute. The Consumer needs to traverse the entire set of nested Jobs to
 5627 determine the complete list of Resources impacted by the Jobs.

5628 Table 42 describes the Job attributes.

5629 **Table 42 – Job attributes**

Name	Job	
Type URI	http://schemas.dmtf.org/cimi/2/Job	
Attribute	Type	Description
state	string	The state of the process associated with this operation. Allowed values are: 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	ref	A reference to the top-level Resource upon which the operation is being performed. Typically, this Resource would be the Resource on which the operation was invoked. Note that 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	ref[]	A list of references to Resources that have been impacted by this Job. Note that this list shall always contain the "targetResource" reference. Array item name: affectedResource <u>Constraints:</u>

Name	Job	
Type URI	http://schemas.dmtf.org/cimi/2/Job	
Attribute	Type	Description
		Provider: support mandatory; mutable Consumer: support mandatory; read-only
action	<i>URI</i>	A URI that indicates the type of action being performed. Constraints: 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. Constraints: 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 <code>Job</code> . This value shall be 100 if the <code>Job</code> is no longer executing, regardless of the outcome. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
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

5630 When implementing or using `Job`, Providers and Consumers shall adhere to the syntax and semantics of
 5631 its attributes as described in Table 42 as well as in the tables describing referred Resources or related
 5632 Collections. Both Consumer and Provider shall serialize this Resource as described below. The following
 5633 pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

5634 **JSON media type:** application/json

5635 **JSON serialization:**

```
5636 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Job",
5637   "id": string,
5638   "name": string, ?
5639   "description": string, ?
5640   "created": string, ?
5641   "updated": string, ?
5642   "properties": { string: string, + }, ?
```

```

5643     "state": string,
5644     "targetResource": { "href": string },
5645     "affectedResources": [ { "href": string }, + ],
5646     "action": string,
5647     "returnCode": number,
5648     "progress": number,
5649     "statusMessage": string,
5650     "timeOfStatusChange": date,
5651     "parentJob": { "href": string }, ?
5652     "nestedJobs": [
5653         { "href": string }, +
5654     ], ?
5655     "operations": [
5656         { "rel": "edit", "href": string }, ?
5657         { "rel": "delete", "href": string }, ?
5658         { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string } ?
5659     ] ?
5660     ...
5661 }
    
```

5662 **XML media type:** application/xml

5663 **XML serialization:**

```

5664 <Job xmlns="http://schemas.dmtf.org/cimi/2">
5665     <id> xs:anyURI </id>
5666     <name> xs:string </name> ?
5667     <description> xs:string </description> ?
5668     <created> xs:dateTime </created> ?
5669     <updated> xs:dateTime </updated> ?
5670     <property key="xs:string"> xs:string </property> *
5671     <state> xs:string </state>
5672     <targetResource href="xs:anyURI"/>
5673     <affectedResource href="xs:anyURI"/> +
5674     <action> xs:anyURI </action>
5675     <returnCode> xs:integer </returnCode>
5676     <progress> xs:integer </progress>
5677     <statusMessage> xs:string </statusMessage>
5678     <timeOfStatusChange> xs:dateTime </timeOfStatusChange>
5679     <parentJob href="xs:anyURI"/> ?
5680     <nestedJob href="xs:anyURI"/> *
5681     <operation rel="edit" href="xs:anyURI"/> ?
    
```

```

5682     <operation rel="delete" href="xs:anyURI"/> ?
5683     <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
5684 href="xs:anyURI"/> ?
5685     <xs:any>*
5686 </Job>

```

5687 5.17.1.1 Operations Resource

5688 This Resource supports the Read, Update, and Delete operations. Deleting a `Job` that is in the
5689 "RUNNING" state shall be the equivalent of first stopping the `Job` and then deleting it. A request to delete
5690 a running `Job` that does not support the "stop" action shall fail.

5691 The following custom operations are also defined:

5692 **stop**

5693 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/stop`

5694 This operation shall stop a `Job`.

5695 Input parameters: None.

5696 Output parameters: None.

5697 During the processing of this operation, the `Job` shall be in the "STOPPING" state.

5698 Upon successful completion of this operation, the `Job` shall be in the "STOPPED" state.

5699 **HTTP protocol**

5700 To stop a `Job`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/stop" URI of the `Job` where
5701 the HTTP request body shall be as described below.

5702 **JSON media type:** `application/json`

5703 **JSON serialization:**

```

5704     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
5705       "action": "http://schemas.dmtf.org/cimi/2/action/stop",
5706       "properties": { string: string, + } ?
5707       ...
5708     }

```

5709 **XML media type:** `application/xml`

5710 **XML serialization**

```

5711     <Action xmlns="http://schemas.dmtf.org/cimi/2">
5712       <action> http://schemas.dmtf.org/cimi/2/action/stop </action>
5713       <property key="xs:string"> xs:string </property> *
5714       <xs:any>*
5715     </Action>

```

5716 Upon successful processing of the request, the HTTP response body may be empty.

5717 **5.17.2 JobCollection Resource**

5718 A JobCollection Resource represents the Collection of Jobs within a Provider and follows the
 5719 Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

5720 **JSON serialization:**

```

5721 { "resourceURI": "http://schemas.dmtf.org/cimi/2/JobCollection",
5722     "id": string,
5723     "count": integer,
5724     "jobs": [
5725         { "resourceURI": "http://schemas.dmtf.org/cimi/2/Job",
5726           "id": string,
5727           ... remaining Job attributes ...
5728         }, +
5729     ] ?
5730     ...
5731 }
```

5732 **XML serialization:**

```

5733 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/JobCollection"
5734     xmlns="http://schemas.dmtf.org/cimi/2">
5735     <id> xs:anyURI </id>
5736     <count> xs:integer </count>
5737     <Job>
5738         <id> xs:anyURI </id>
5739         ... remaining Job attributes ...
5740     </Job> *
5741     <xs:any>*
5742 </Collection>
```

5743 **5.17.3 Meter Resource**

5744 This Resource represents an available Meter of some property associated to a given Resource.

5745 If a Meter's "targetResource" is deleted all Meters associated with that Resource shall also be
 5746 deleted. In other words, deleting a Resource-specific MetersCollection (e.g., a Machine's
 5747 MetersCollection) shall also result in the deletion of the Meters referenced from that Collection.

5748 Table 43 describes the Meter attributes.

5749 **Table 43 – Meter attributes**

Name	Meter	
Type URI	http://schemas.dmtf.org/cimi/2/Meter	
Attribute	Type	Description
targetResource	ref	A reference to the Resource to which the Meter is related. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only

Name	Meter	
Type URI	http://schemas.dmtf.org/cimi/2/Meter	
Attribute	Type	Description
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
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> <i>[Sample]</i>	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

5750 When implementing or using Meter, Providers and Consumers shall adhere to the syntax and semantics
5751 of its attributes as described in Table 43 as well as in the tables describing related Collections. Both

5752 Consumer and Provider shall serialize this Resource as described below. The following pseudo-schemas
 5753 (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

5754 **JSON media type:** application/json

5755 **JSON serialization:**

```

5756 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Meter",
5757     "id": string,
5758     "name": string, ?
5759     "description": string, ?
5760     "created": string, ?
5761     "updated": string, ?
5762     "properties": { string: string, + }, ?
5763     "targetResource": { "href": string },
5764     "aspect": string,
5765     "units": string,
5766     "sampleInterval": number,
5767     "timeScope": string,
5768     "intervalDuration": string,
5769     "isContinuous": boolean,
5770     "samples": { "href": string }, ?
5771     "minValue": string, ?
5772     "maxValue": string, ?
5773     "stopTime": string, ?
5774     "expiresTime": string, ?
5775     "operations": [
5776         { "rel": "edit", "href": string }, ?
5777         { "rel": "delete", "href": string }, ?
5778         { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string }, ?
5779         { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string } ?
5780     ] ?
5781     ...
5782 }
```

5783 **XML media type:** application/xml

5784 **XML serialization:**

```

5785 <Meter xmlns="http://schemas.dmtf.org/cimi/2">
5786     <id> xs:anyURI </id>
5787     <name> xs:string </name> ?
5788     <description> xs:string </description> ?
5789     <created> xs:dateTime </created> ?
5790     <updated> xs:dateTime </updated> ?
```

```

5791 <property key="xs:string"> xs:string </property> *
5792 <targetResource href="xs:anyURI"/>
5793 <aspect> xs:anyURI </aspect>
5794 <units> xs:string </units>
5795 <sampleInterval> xs:integer </sampleInterval>
5796 <timeScope> xs:string <timeScope>
5797 <intervalDuration xs:duration </intervalDuration>
5798 <isContinuous> xs:boolean </isContinuous>
5799 <samples href="xs:anyURI"/> ?
5800 <minValue> xs:string </minValue> ?
5801 <maxValue> xs:string </maxValue> ?
5802 <stopTime> xs:dateTime </stopTime> ?
5803 <expiresTime> xs:dateTime </expiresTime> ?
5804 <operation rel="edit" href="xs:anyURI"/> ?
5805 <operation rel="delete" href="xs:anyURI"/> ?
5806 <operation rel="http://schemas.dmtf.org/cimi/2/action/start"
5807 href="xs:anyURI"/> ?
5808 <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
5809 href="xs:anyURI"/> ?
5810 <xs:any>*
5811 </Meter>

```

5812 **5.17.3.1 Collections**

5813 The following clauses describe the Collection resources that are components of *Meters*.

5814 **5.17.3.1.1 SampleCollection Resource**

5815 The Resource type for each item of this Collection is “Sample”, defined in Table 44:

5816 **Table 44 – Sample attributes**

Name	Sample	
Type URI	http://schemas.dmtf.org/cimi/2/Sample	
Attribute	Type	Description
timestamp	<i>dateTime</i>	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	<i>string</i>	Indicates the sampled value of the measure. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only

5817 When implementing or using `Sample`, Providers and Consumers shall adhere to the syntax and
 5818 semantics of its attributes as described in Table 44 as well as in the tables describing related Collections.
 5819 Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-
 5820 schemas (see notation in 1.3) describe the serialization of the `Sample` Collection in both JSON and
 5821 XML.

5822 **JSON serialization:**

```
5823 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SampleCollection",
5824   "id": string,
5825   "count": number,
5826   "samples": [
5827     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Sample",
5828       "id": string,
5829       "name": string, ?
5830       "description": string, ?
5831       "created": string, ?
5832       "updated": string, ?
5833       "properties": { string: string, + }, ?
5834       "timestamp": string,
5835       "value": string
5836       ...
5837     }, +
5838   ], ?
5839   ...
5840 }
```

5841 **XML serialization:**

```
5842 <Collection
5843   resourceURI="http://schemas.dmtf.org/cimi/2/SampleCollection"
5844   xmlns="http://schemas.dmtf.org/cimi/2">
5845   <id> xs:anyURI </id>
5846   <count> xs:integer </count>
5847   <Sample>
5848     <id> xs:anyURI </id>
5849     <name> xs:string </name> ?
5850     <description> xs:string </description> ?
5851     <created> xs:dateTime </created> ?
5852     <updated> xs:dateTime </updated> ?
5853     <property key="xs:string"> xs:string </property> *
5854     <sample timestamp="xs:dateTime" value="xs:string"/>
5855     <xs:any> *
5856   </Sample> *
```

```
5857     <xs:any>*
5858 </Collection>
```

5859 5.17.3.2 Operations

5860 This Resource supports the Read, Update, and Delete operations. Create is supported via the
5861 `MeterCollection` Resource. The deletion of a `Meter` shall remove the `Meter` from the
5862 targetResource's "meter" attribute.

5863 The following custom operations are also defined:

5864 **start**

5865 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/start`

5866 This operation shall start a `Meter`.

5867 Input parameters: None.

5868 Output parameters: None.

5869 Upon successful completion of this operation, the `Meter` shall start recording samples related to its
5870 associated Resource.

5871 **HTTP protocol**

5872 To start a `Meter`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/start" URI of the `Meter`
5873 where the HTTP request body shall be as described below.

5874 **JSON media type:** `application/json`

5875 **JSON serialization:**

```
5876 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
5877   "action": "http://schemas.dmtf.org/cimi/2/action/start",
5878   "properties": { string: string, + } ?
5879   ...
5880 }
```

5881 **XML media type:** `application/xml`

5882 **XML serialization**

```
5883 <Action xmlns="http://schemas.dmtf.org/cimi/2">
5884   <action> http://schemas.dmtf.org/cimi/2/action/start </action>
5885   <property key="xs:string"> xs:string </property> *
5886   <xs:any>*
5887 </Action>
```

5888 Upon successful processing of the request, the HTTP response body may be empty.

5889 **stop**

5890 **/link@rel:** `http://schemas.dmtf.org/cimi/2/action/stop`

5891 This operation shall stop a `Meter`.

5892 Input parameters: None.

5893 Output parameters: None.

5894 Upon successful completion of this operation, the `Meter` shall no longer be recording samples related to
5895 its associated `Resource`.

5896 **HTTP protocol**

5897 To stop a `Meter`, a POST is sent to the "http://schemas.dmtf.org/cimi/2/action/stop" URI of the `Meter`
5898 where the HTTP request body shall be as described below.

5899 **JSON media type:** application/json

5900 **JSON serialization:**

```
5901 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
5902   "action": "http://schemas.dmtf.org/cimi/2/action/stop",
5903   "properties": { string: string, + } ?
5904   ...
5905 }
```

5906 **XML media type:** application/xml

5907 **XML serialization**

```
5908 <Action xmlns="http://schemas.dmtf.org/cimi/2">
5909   <action> http://schemas.dmtf.org/cimi/2/action/stop </action>
5910   <property key="xs:string"> xs:string </property> *
5911   <xs:any>*
5912 </Action>
```

5913 Upon successful processing of the request, the HTTP response body may be empty.

5914 **5.17.4 MeterCollection Resource**

5915 A `MeterCollection` Resource represents the Collection of `Meters` within a `Provider` and follows the
5916 Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

5917 **JSON serialization:**

```
5918 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterCollection",
5919   "id": string,
5920   "count": number,
5921   "meters": [
5922     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Meter",
5923       "id": string,
5924       ... remaining Meter attributes ...
5925     }, +
5926   ], ?
5927   "operations": [ { "rel": "add", "href": string } ? ]
5928   ...
```

5929 }

5930 **XML serialization:**

```

5931 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/MeterCollection"
5932     xmlns="http://schemas.dmtf.org/cimi/2">
5933     <id> xs:anyURI </id>
5934     <count> xs:integer </count>
5935     <Meter>
5936         <id> xs:anyURI </id>
5937         ... remaining Meter attributes ...
5938     </Meter> *
5939     <operation rel="add" href="xs:anyURI"/> ?
5940     <xs:any>*
5941 </Collection>
    
```

5942 **5.17.4.1 Operations**

5943 **NOTE** The "add" operation requires that a MeterTemplate be used (see 4.2.1.1).

5944 If Meters are created through the global (Cloud Entry Point) MeterCollection's "add" operation,
 5945 they shall be added automatically to the corresponding targetResource's "Meters" Collection Resource
 5946 as well.

5947 **5.17.5 MeterTemplate Resource**

5948 A MeterTemplate represents the information needed to create a new Meter. Table 45 describes the
 5949 MeterTemplate attributes.

5950 **Table 45 – MeterTemplate attributes**

MeterTemplate		
Name	MeterTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/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 MeterConfiguration referenced. If this Template is used to create a new Meter through the global (Cloud Entry Point) MetersCollection, this attribute shall be present. If this Template is used to create a new Meter through a targetResource's MetersCollection, this attribute shall either be absent or have the same value as the "id" of the targetResource to which this Meter is being added. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
meterConfig	ref	A reference to the MeterConfiguration that is used to create a Meter from this MeterTemplate. Note that the attributes of the MeterConfiguration may be specified rather than a reference to an existing MeterConfiguration Resource. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

5951 When implementing or using MeterTemplate, Providers and Consumers shall adhere to the syntax
 5952 and semantics of its attributes as described in Table 45 as well as in the tables describing referred

5953 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 5954 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 5955 both JSON and XML.

5956 **JSON media type:** application/json

5957 **JSON serialization:**

```
5958 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterTemplate",
5959   "id": string,
5960   "name": string, ?
5961   "description": string, ?
5962   "created": string, ?
5963   "updated": string, ?
5964   "properties": { string: string, + }, ?
5965   "targetResource": { string },
5966   "meterConfig": {
5967     "href": string | ... MeterConfiguration attributes ...
5968   },
5969   "operations": [
5970     { "rel": "edit", "href": string }, ?
5971     { "rel": "delete", "href": string } ?
5972   ] ?
5973   ...
5974 }
```

5975 **XML media type:** application/xml

5976 **XML serialization:**

```
5977 <MeterTemplate xmlns="http://schemas.dmtf.org/cimi/2">
5978   <id> xs:anyURI </id>
5979   <name> xs:string </name> ?
5980   <description> xs:string </description> ?
5981   <created> xs:dateTime </created> ?
5982   <updated> xs:dateTime </updated> ?
5983   <property key="xs:string"> xs:string </property> *
5984   <targetResource href="xs:anyURI"/>
5985   <meterConfig href="xs:anyURI"?>
5986     ... MeterConfiguration attributes ... ?
5987   </meterConfig>
5988   <operation rel="edit" href="xs:anyURI"/> ?
5989   <operation rel="delete" href="xs:anyURI"/> ?
5990   <xs:any>*
5991 </MeterTemplate>
```

5992 **5.17.6 MeterTemplateCollection Resource**

5993 A `MeterTemplateCollection` Resource represents the Collection of `MeterTemplate`
5994 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
5995 shall be serialized as follows:

5996 **JSON serialization:**

```
5997 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterTemplateCollection",
5998   "id": string,
5999   "count": number,
6000   "meterTemplates": [
6001     { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterTemplate",
6002       "id": string,
6003       ... remaining MeterTemplate attributes ...
6004     }, +
6005   ], ?
6006   "operations": [ { "rel": "add", "href": string } ? ]
6007   ...
6008 }
```

6009 **XML serialization:**

```
6010 <Collection
6011   resourceURI="http://schemas.dmtf.org/cimi/2/MeterTemplateCollection"
6012   xmlns="http://schemas.dmtf.org/cimi/2">
6013   <id> xs:anyURI </id>
6014   <count> xs:integer </count>
6015   <MeterTemplate>
6016     <id> xs:anyURI </id>
6017     ... remaining MeterTemplate attributes ...
6018   </MeterTemplate> *
6019   <operation rel="add" href="xs:anyURI"/> ?
6020   <xs:any>*
6021 </Collection>
```

6022 **5.17.6.1 Operations**

6023 This Resource supports the Read and Update operations. Creation of new `MeterTemplate` Resources
6024 is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.

6025 **5.17.7 MeterConfiguration Resource**

6026 A `MeterConfiguration` represents the definition of a `Meter`. Table 46 describes the
6027 `MeterConfiguration` attributes.

6028

Table 46 – MeterConfiguration attributes

Name	MeterConfiguration	
Type URI	http://schemas.dmtf.org/cimi/2/MeterConfiguration	
Attribute	Type	Description
associatedTo	<i>URI[]</i>	An array of URIs that indicate the types of Resources to which a Meter created from this configuration can be applied. The value space of these URIs is identical to that of ResourceMetadata.typeURI, which is a URI that uniquely identifies a Resource type. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
aspect	<i>URI</i>	A unique identifier representing the aspect of the Resource being metered. See Table 47 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 Consumer: support mandatory; read-write
timeScope	<i>string</i>	The time scope to which the Meter value applies. Two possible values: "Point" indicates that the Meter applies to a point in time. "Interval" indicates that the Meter applies to a time interval. For instance, it would be possible to define a MeterConfiguration whose purpose is to provide the daily average CPU usage. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write
intervalDuration	<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; mutable Consumer: support mandatory; read-write
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; mutable Consumer: support mandatory; read-write

6029 The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

6030 **JSON media type:** application/json

6031 **JSON serialization:**

```

6032 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterConfiguration",
6033   "id": string,
6034   "name": string, ?
6035   "description": string, ?
6036   "created": string, ?
6037   "updated": string, ?
6038   "properties": { string: string, + }, ?
    
```

```

6039     "associatedTo": [
6040         { "href": string }, +
6041     ], ?
6042     "aspect": string,
6043     "units": string,
6044     "sampleInterval": number,
6045     "timeScope": string,
6046     "intervalDuration": string,
6047     "isContinuous": boolean,
6048     "operations": [
6049         { "rel": "edit", "href": string }, ?
6050         { "rel": "delete", "href": string } ?
6051     ] ?
6052     ...
6053 }

```

6054 **XML media type:** application/xml

6055 **XML serialization:**

```

6056 <MeterConfiguration xmlns="http://schemas.dmtf.org/cimi/2">
6057     <id> xs:anyURI </id>
6058     <name> xs:string </name> ?
6059     <description> xs:string </description> ?
6060     <created> xs:dateTime </created> ?
6061     <updated> xs:dateTime </updated> ?
6062     <property key="xs:string"> xs:string </property> *
6063     <associatedTo href="xs:anyURI"/> *
6064     <aspect> xs:anyURI </aspect>
6065     <units> xs:string </units>
6066     <sampleInterval> xs:integer </sampleInterval>
6067     <timeScope> xs:string </timeScope>
6068     <intervalDuration> xs:duration </intervalDuration>
6069     <isContinuous> xs:boolean </isContinuous>
6070     <operation rel="edit" href="xs:anyURI"/> ?
6071     <operation rel="delete" href="xs:anyURI"/> ?
6072     <xs:any>*
6073 </MeterConfiguration>

```

6074 Table 47 describes the "aspect" URIs defined by this specification. Providers may define new aspect
6075 URIs and it is recommended that these URIs be dereferencable such that Consumers can discover the
6076 details of the new aspect. For brevity the "URI" column in the table only shows the last part of the URI. It
6077 should be appended to: "http://schemas.dmtf.org/cimi/2/aspect".

6078

Table 47 – aspect URIs

Aspect	Description
cpu	The percentage CPU usage of the Resource. Typically associated with <code>CloudEntryPoint</code> , <code>System</code> , and <code>Machine</code> Resources. For Resources that group other Resources (e.g., <code>CloudEntryPoint</code> or <code>System</code> Resources), this aspect provides the aggregated percentage usage of the CPU.
memory	The amount of memory being used by the Resource. Typically associated with <code>CloudEntryPoint</code> , <code>System</code> , and <code>Machine</code> Resources. For Resources that group other Resources (e.g., <code>CloudEntryPoint</code> or <code>System</code> Resources), this aspect provides the aggregated usage of the memory.
disk	The amount of disk being used by the Resource. Typically associated with <code>CloudEntryPoint</code> , <code>System</code> , <code>Machine</code> , and <code>Volume</code> Resources. For Resources that group other Resources (e.g., <code>CloudEntryPoint</code> or <code>System</code> Resources), this aspect provides the aggregated disk usage.
bandwidth	The amount of network traffic. Typically associated with <code>CloudEntryPoint</code> , <code>System</code> , and <code>Network</code> Resources. For <code>CloudEntryPoint</code> and <code>System</code> 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 <code>Machine</code> , <code>NetworkPort</code> , and <code>Volume</code> Resources. For <code>Machine</code> Resources, this aspect provides the aggregated input bandwidth usage of all its network interfaces .
outputBandwidth	The amount of output bandwidth used by the Resource. Typically associated with <code>Machine</code> , <code>NetworkPort</code> , and <code>Volume</code> Resources. For <code>Machine</code> Resources, this aspect provides the aggregated output bandwidth usage of all its network interfaces.

6079 **5.17.7.1 Operations**

6080 This Resource supports the Read, Update, and Delete operations. Create is supported through the
6081 `MeterConfigurationCollection` Resource.

6082 **5.17.8 MeterConfigurationCollection Resource**

6083 A `MeterConfigurationCollection` Resource represents the Collection of
6084 `MeterConfigurations` within a Provider and follows the Collection pattern defined in clause 5.5.12.
6085 This Resource shall be serialized as follows:

6086 **JSON serialization:**

```
6087 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterConfigurationCollection",
6088   "id": string,
6089   "count": number,
6090   "meterConfigurations": [
6091     { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterConfiguration",
6092       "id": string,
6093       ... remaining MeterConfiguration attributes ...
6094     }, +
6095   ], ?
6096   "operations": [ { "rel": "add", "href": string } ? ]
6097   ...
6098 }
```

6099 **XML serialization:**

```
6100 <Collection
```

```

6101     resourceURI="http://schemas.dmtf.org/cimi/2/MeterConfigurationCollection"
6102     xmlns="http://schemas.dmtf.org/cimi/2">
6103     <id> xs:anyURI </id>
6104     <count> xs:integer </count>
6105     <MeterConfiguration>
6106         <id> xs:anyURI </id>
6107         ... remaining MeterConfiguration attributes ...
6108     </MeterConfiguration> *
6109     <operation rel="add" href="xs:anyURI"/> ?
6110     <xs:any>*
6111 </Collection>
    
```

6112 **5.17.8.1 Operations**

6113 This Resource supports the Read and Update operations. Creation of new MeterConfiguration
 6114 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 6115 4.2.1.1.

6116 **5.17.9 EventLog Resource**

6117 A Resource that represents a registry of Events.

6118 If an EventLog's "targetResource" is deleted the EventLog associated with that Resource may also
 6119 be deleted. In other words, deleting a Resource (e.g., a Machine) may also result in the deletion of the
 6120 EventLog referenced from that Resource. This behavior is denoted by the EventLog "Linked"
 6121 capability.

6122 If an EventLog is deleted, all of its Events shall also be deleted.

6123 Table 48 describes the EventLog attributes.

6124 **Table 48 – EventLog attributes**

Name		EventLog
Type URI		http://schemas.dmtf.org/cimi/2/EventLog
Attribute	Type	Description
targetResource	ref	A reference to the Resource to which the Events are related. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
events	collection [Event]	A reference to the list of occurred Events. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only
persistence	string	A value that indicates the persistence of the Events within the EventLog. For instance, daily, weekly, monthly, or yearly. Events that exceed the persistence duration may be deleted. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write

Name	EventLog																
Type URI	http://schemas.dmtf.org/cimi/2/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															

6125 When implementing or using EventLog, Providers and Consumers shall adhere to the syntax and
 6126 semantics of its attributes as described in Table 48 as well as in the tables describing embedded
 6127 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 6128 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 6129 both JSON and XML.

6130 **JSON media type:** application/json

6131 **JSON serialization:**

```

6132 { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLog",
6133   "id": string,
6134   "name": string, ?
6135   "description": string, ?
6136   "created": string, ?
6137   "updated": string, ?
6138   "properties": { string: string, + }, ?
6139   "targetResource": { "href": string },
6140   "events": { "href": string },
6141   "persistence": string,
6142   "summary": {
6143     "low": number,
6144     "medium": number,
6145     "high": number,
6146     "critical": number
    
```

```

6147     }, ?
6148     "operations": [
6149         { "rel": "edit", "href": string }, ?
6150         { "rel": "delete", "href": string } ?
6151     ] ?
6152     ...
6153 }

```

6154 **XML media type:** application/xml

6155 **XML serialization:**

```

6156 <EventLog xmlns="http://schemas.dmtf.org/cimi/2">
6157     <id> xs:anyURI </id>
6158     <name> xs:string </name> ?
6159     <description> xs:string </description> ?
6160     <created> xs:dateTime </created> ?
6161     <updated> xs:dateTime </updated> ?
6162     <property key="xs:string"> xs:string </property> *
6163     <targetResource href="xs:anyURI"/>
6164     <events href="xs:anyURI"/>
6165     <persistence> xs:string </persistence>
6166     <summary>
6167         <low> xs:integer </low>
6168         <medium> xs:integer </medium>
6169         <high> xs:integer </high>
6170         <critical> xs:integer </critical>
6171     </summary>
6172     <operation rel="edit" href="xs:anyURI"/> ?
6173     <operation rel="delete" href="xs:anyURI"/> ?
6174     <xs:any>*
6175 </EventLog>

```

6176 5.17.9.1 Collections

6177 The following clauses describe the Collection Resources EventLogs.

6178 5.17.9.1.1 events Collection

6179 The Resource type for each item of this Collection is “Event” as defined in clause 5.17.13.

6180 **JSON serialization:**

```

6181 { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventCollection",
6182   "id": string,
6183   "count": number,
6184   "events": [

```

```

6185     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Event",
6186       "id": string,
6187       ... remaining Event attributes ...
6188     }, +
6189   ], ?
6190   "operations": [ { "rel": "add", "href": string } ? ]
6191   ...
6192 }
    
```

6193 XML serialization:

```

6194 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/EventCollection"
6195   xmlns="http://schemas.dmtf.org/cimi/2">
6196   <id> xs:anyURI </id>
6197   <count> xs:integer </count>
6198   <Event>
6199     <id> xs:anyURI </id>
6200     ... remaining Event attributes ...
6201   </Event> *
6202   <operation rel="add" href="xs:anyURI"/> ?
6203   <xs:any>*
6204 </Collection>
    
```

6205 5.17.9.2 Operations

6206 This Resource supports the Read, Update, and Delete operations.

6207 5.17.10 EventLogCollection Resource

6208 An EventLogCollection Resource represents the Collection of EventLogs within a Provider and
 6209 follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

6210 JSON serialization:

```

6211 { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLogCollection",
6212   "id": string,
6213   "count": number,
6214   "eventLogs": [
6215     { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLog",
6216       "id": string,
6217       ... remaining EventLog attributes ...
6218     }, +
6219   ], ?
6220   "operations": [ { "rel": "add", "href": string } ? ]
6221   ...
6222 }
    
```

6223 **XML serialization:**

```

6224 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/EventLogCollection"
6225     xmlns="http://schemas.dmtf.org/cimi/2">
6226   <id> xs:anyURI </id>
6227   <count> xs:integer </count>
6228   <EventLog>
6229     <id> xs:anyURI </id>
6230     ... remaining EventLog attributes ...
6231   </EventLog> *
6232   <operation rel="add" href="xs:anyURI"/> ?
6233   <xs:any>*
6234 </Collection>
    
```

6235 **5.17.11 EventLogTemplate Resource**

6236 An EventLogTemplate represents the information needed to create a new EventLog. Table 49
 6237 describes the EventLogTemplate attributes.

6238 **Table 49 – EventLogTemplate attributes**

Name	EventLogTemplate	
Type URI	http://schemas.dmtf.org/cimi/2/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

6239 When implementing or using EventLogTemplate, Providers and Consumers shall adhere to the
 6240 syntax and semantics of its attributes as described in Table 49 as well as in the tables describing referred
 6241 Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described
 6242 below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in
 6243 both JSON and XML.

6244 **JSON media type:** application/json

6245 **JSON serialization:**

```

6246 { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLogTemplate",
6247   "id": string,
6248   "name": string, ?
6249   "description": string, ?
6250   "created": string, ?
6251   "updated": string, ?
6252   "properties": { string: string, + }, ?
    
```

```

6253     "targetResource": { string },
6254     "persistence": string,
6255     "operations": [
6256         { "rel": "edit", "href": string }, ?
6257         { "rel": "delete", "href": string } ?
6258     ] ?
6259     ...
6260 }
    
```

6261 **XML media type:** application/xml

6262 **XML serialization:**

```

6263 <EventLogTemplate xmlns="http://schemas.dmtf.org/cimi/2">
6264     <id> xs:anyURI </id>
6265     <name> xs:string </name> ?
6266     <description> xs:string </description> ?
6267     <created> xs:dateTime </created> ?
6268     <updated> xs:dateTime </updated> ?
6269     <property key="xs:string"> xs:string </property> *
6270     <targetResource href="xs:anyURI"/>
6271     <persistence> xs:string </persistence>
6272     <operation rel="edit" href="xs:anyURI"/> ?
6273     <operation rel="delete" href="xs:anyURI"/> ?
6274     <xs:any>*
6275 </EventLogTemplate>
    
```

6276 5.17.12 EventLogTemplateCollection Resource

6277 An EventLogTemplateCollection Resource represents the Collection of EventLogTemplate
 6278 Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource
 6279 shall be serialized as follows:

6280 **JSON serialization:**

```

6281 { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLogTemplateCollection",
6282   "id": string,
6283   "count": number,
6284   "eventLogTemplates": [
6285     { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLogTemplate",
6286       "id": string,
6287       ... remaining EventLogTemplate attributes ...
6288     }, +
6289   ], ?
6290   "operations": [ { "rel": "add", "href": string } ? ]
6291   ...
    
```

6292 }

6293 **XML serialization:**

```

6294 <Collection
6295     resourceURI="http://schemas.dmtf.org/cimi/2/EventLogTemplateCollection"
6296     xmlns="http://schemas.dmtf.org/cimi/2">
6297     <id> xs:anyURI </id>
6298     <count> xs:integer </count>
6299     <EventLogTemplate>
6300         <id> xs:anyURI </id>
6301         ... remaining EventLogTemplate attributes ...
6302     </EventLogTemplate> *
6303     <operation rel="add" href="xs:anyURI"/> ?
6304     <xs:any>*
6305 </Collection>
    
```

6306 **5.17.12.1 Operations**

6307 This Resource supports the Read and Update operations. Creation of new `EventLogTemplate`
 6308 Resources is supported by the way of a POST to the "add" operation's URI as described in clause
 6309 4.2.1.1.

6310 **5.17.13 Event Resource**

6311 A Resource that represents the occurrence of an event within the managed infrastructure. Some
 6312 examples of `Event` are:

- 6313 • Machine X has been rebooted by guest OS.
- 6314 • Machine X is not responding to platform services.
- 6315 • A new vCPU has been added to machine X following defined elasticity rules.

6316 The scope of the `Event` concept is any information that the Provider is able to track within its
 6317 infrastructure and that can constitute useful information for the Consumer. Possible examples include, but
 6318 are not limited to, errors and inconveniences that occur in the (virtual) resources assigned to Consumers;
 6319 Provider-initiated actions, such as maintenance tasks; etc.

6320 Table 50 describes the `Event` attributes.

6321 **Table 50 – Event attributes**

Name	Event	
Type URI	http://schemas.dmtf.org/cimi/2/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/2/Event	
Attribute	Type	Description
type	<i>URI</i>	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. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
content	<i>any</i>	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: In the case of a monitoring event, the content shall hold the target Resource ID and type, measured attribute(s), and status value(s). In the case of an audit event conforming to the CADF model, the content shall hold the detailed event structure that complies with CADF event schema. In the case of a CIM Indication, the content shall hold the structure and attributes defined for such events. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only
outcome	<i>string</i>	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. Core outcomes are: Pending: The <code>Event</code> is about an action or process that is still ongoing. Unknown: The <code>Event</code> is about a request or action that is not known by the Provider. Status: The <code>Event</code> reports on the state or status of a Resource. Success: The <code>Event</code> reports on a successful outcome of some action or process. Warning: The <code>Event</code> reports on a situation that requires attention or remedial action. Failure: The <code>Event</code> reports on a failed outcome of some action or process. This set of core outcome values may be extended to accommodate possible outcomes of a specific <code>Event</code> type. In this case, the extended set of values shall apply to all <code>Events</code> of this type. Constraints: Provider: support optional; immutable Consumer: support optional; read-only
severity	<i>string</i>	A value indicating the <code>Event</code> severity. Possible values are: critical high medium low 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. Constraints: Provider: support optional; immutable Consumer: support optional; read-only
contact	<i>string</i>	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. Constraints: Provider: support optional; immutable Consumer: support optional; read-only

6322 NOTE There exists a legacy of several `Event` models that have been standardized or designed for various
 6323 domains relevant to IT. The objective in CIMI is not to elect one particular `Event` model, but to select as top-level
 6324 `Event` attributes the most immediately relevant data useful for `Event` processing in a Cloud environment.

6325 Additional `Event` data may still be represented in the variable content attribute that allows for mapping other `Event`
6326 models into a CIMI `Event`.

6327 When implementing or using `Event`, Providers and Consumers shall adhere to the syntax and semantics
6328 of its attributes as described in Table 50. Both Consumer and Provider shall serialize this Resource as
6329 described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the
6330 Resource in both JSON and XML.

6331 **JSON media type:** `application/json`

6332 **JSON serialization:**

```
6333 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Event",
6334   "id": string,
6335   "name": string, ?
6336   "description": string, ?
6337   "created": string, ?
6338   "updated": string, ?
6339   "properties": { string: string, + }, ?
6340   "timestamp": string,
6341   "type": string,
6342   "content": any, ?
6343   "outcome": string, ?
6344   "severity": string, ?
6345   "contact": string, ?
6346   ...
6347 }
```

6348 **XML media type:** `application/xml`

6349 **XML serialization:**

```
6350 <Event xmlns="http://schemas.dmtf.org/cimi/2">
6351   <id> xs:anyURI </id>
6352   <name> xs:string </name> ?
6353   <description> xs:string </description> ?
6354   <created> xs:dateTime </created> ?
6355   <updated> xs:dateTime </updated> ?
6356   <property key="xs:string"> xs:string </property> *
6357   <timestamp> xs:dateTime </timestamp>
6358   <type> xs:string </type>
6359   <content> xs:any* </content> ?
6360   <outcome> xs:string </outcome> ?
6361   <severity> xs:string </severity> ?
6362   <contact> xs:string </contact> ?
6363   <xs:any>*
6364 </Event>
```

6365 Table 51 describes the "type" URIs that are defined or acknowledged by this specification. Additional
 6366 types may be added by a Provider, for example to characterize external events mapped into CIMI
 6367 *Events*. It is recommended that these URIs be dereferencable such that Consumers can discover a
 6368 more detailed description of the type. *Event* types defined by this specification share the same base
 6369 URI: <http://schemas.dmtf.org/cimi/2/event/>. For brevity, if the "Event Type" column in the table only shows
 6370 a relative URI (e.g., *state*) it shall be appended to the end of this base URI.

6371

Table 51 – 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 <i>Event</i> 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 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 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 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>																		

6372 The following pseudo-schemas describe the serialization of the "content" property for various types of
6373 events:

6374 **"state" event:**

6375 **JSON serialization:**

```
6376 { "id": string,
6377   ...
6378   "type": "http://schemas.dmtf.org/cimi/2/event/state",
6379   "content": {
6380     "resName": string,
6381     "resource" : { "href" : string },
6382     "resType" : string,
6383     "state" : string,
6384     "previous" : string ?
6385   }
6386   ...
6387 }
```

6388 **XML serialization:**

```
6389 <Event xmlns="http://schemas.dmtf.org/cimi/2">
6390   ...
6391   <type> http://schemas.dmtf.org/cimi/2/event/state </type>
6392   <content>
6393     <resName> xs:string </resName>
6394     <resource href="xs:anyURI"/>
6395     <resType> xs:anyURI </resType>
6396     <state> xs:string </state>
6397     <previous> xs:string </previous> ?
6398   </content> ?
6399   ...
6400 </Event>
```

6401

6402 **"alarm" event:**

6403 **JSON serialization:**

```
6404 { "id": string,
6405   ...
6406   "type": "http://schemas.dmtf.org/cimi/2/event/alarm",
6407   "content": {
6408     "resName": string ?
6409     "resource" : { "href" : string }, ?
6410     "resType" : string ?
```

```

6411     "code" : string,
6412     "detail" : string ?
6413   }
6414   ...
6415 }

```

6416 **XML serialization:**

```

6417 <Event xmlns="http://schemas.dmtf.org/cimi/2">
6418   ...
6419   <type> http://schemas.dmtf.org/cimi/2/event/alarm </type>
6420   <content>
6421     <resname> xs:string </resname> ?
6422     <resource href="xs:anyURI"/> ?
6423     <restype> xs:anyURI </restype> ?
6424     <code> xs:string </code>
6425     <detail> xs:string </detail> ?
6426   </content> ?
6427   ...
6428 </Event>

```

6429 **"model" event:**

6430 **JSON serialization:**

```

6431 { "id": string,
6432   ...
6433   "type": "http://schemas.dmtf.org/cimi/2/event/model",
6434   "content": {
6435     "resName": string, ?
6436     "resource" : { "href" : string }, ?
6437     "resType" : string, ?
6438     "change" : string,
6439     "detail" : string ?
6440   }
6441   ...
6442 }

```

6443 **XML serialization:**

```

6444 <Event xmlns="http://schemas.dmtf.org/cimi/2">
6445   ...
6446   <type> http://schemas.dmtf.org/cimi/2/event/model </type>
6447   <content>
6448     <resname> xs:string </resname> ?
6449     <resource href="xs:anyURI"/> ?

```

```

6450     <restype> xs:anyURI </restype> ?
6451     <change> xs:string </change>
6452     <detail> xs:string </detail> ?
6453 </content> ?
6454     ...
6455 </Event>

```

6456 **"access" event:**

6457 **JSON serialization:**

```

6458     { "id": string,
6459       ...
6460       "type": "http://schemas.dmtf.org/cimi/2/event/access",
6461       "content": {
6462         "operation": string,
6463         "resource" : { "href" : string },
6464         "detail" : string, ?
6465         "initiator" : string ?
6466       }
6467       ...
6468     }

```

6469 **XML serialization:**

```

6470 <Event xmlns="http://schemas.dmtf.org/cimi/2">
6471     ...
6472 <type> http://schemas.dmtf.org/cimi/2/event/access </type>
6473 <content>
6474     <operation> xs:string </operation>
6475     <resource href="xs:anyURI"/>
6476     <detail> xs:string </detail> ?
6477     <initiator> xs:string </initiator> ?
6478 </content> ?
6479     ...
6480 </Event>

```

6481 5.17.13.1 Operations

6482 This resource supports the Read, Update, and Delete operations.

6483 6 Security considerations

6484 There are many security mechanisms that can be used in conjunction with this specification. This
6485 specification does not mandate any particular mechanism. Providers shall provide enough information
6486 about their security mechanisms so that the Consumer can implement the necessary algorithms to
6487 successfully communicate with the Provider.

6488 An implementation may set limits on the length of attribute values it accepts. An implementation may set
6489 limits on the size of arrays it accepts. An implementation may set limits on the size of the request body or
6490 the length of request URIs it accepts. These limits may not all be advertised in the ResourceMetadata,
6491 although this specification recommends Providers to do so. A Provider that receives a request that
6492 exceeds any of these limits, shall return a response with an appropriate standard HTTP status code.

6493

6494

6495
6496
6497

ANNEX A (normative) OVF support in CIM

6498 This annex defines how elements of an OVF descriptor are mapped to CIM resources and their
6499 attributes. This definition allows the import of an OVF package to create multiple CIM resources. This is
6500 done by specifying a reference to an OVF package in the import operation of a `SystemCollection` or
6501 `SystemTemplateCollection` (the Media Type at that URI shall be “application/ovf”). Refer to
6502 [DSP0243](#) for more information about OVF.

6503 Support for OVF import and export is optional for a Provider and it is an implementation choice as to how
6504 many of the attributes in the OVF package are exposed through CIM resources. A Provider may support
6505 the import of OVF package for only `Systems`, only `SystemTemplates` or both. Support for the actual
6506 import and export of an OVF package is handled by a hypervisor under the management of the CIM
6507 implementation, and thus the CIM resources that are created reflect what the hypervisor did upon import
6508 and form a “View” into the results.

6509 The import of an OVF package can be reflected in the creation of `Templates` that can be later used to
6510 create `Systems`, `Machines` and other component `Resources`. The import of an OVF package can also
6511 be used to directly create `Systems`, `Machines`, and other component `Resources`, bypassing the step
6512 of creating `Templates`.

6513 Clause 5.13.5 details how to import an OVF file to create a `SystemTemplate` (and component
6514 `Resources`). The `SystemTemplate` thus created contains a reference to a `MachineTemplate` for
6515 every `VirtualSystem` that is defined in the OVF descriptor `VirtualSystemCollection`. Note
6516 that CIM currently allows `Systems of Systems`, so for each `VirtualSystemCollection`
6517 encountered in a nested set of collections, a separate `SystemTemplate` is created within the parent
6518 `SystemTemplate` with `MachineTemplates` for each of the contained `VirtualSystems` in that
6519 `VirtualSystemCollection`.

6520 The values of the attributes for the `MachineTemplate` are taken from the
6521 `VirtualHardwareSection` of the `VirtualSystem` description (required in OVF). If more than
6522 one `VirtualHardwareSection` is used for a given `VirtualSystem` (allowed in OVF), the result
6523 is implementation dependent, but the implementation might choose a `MachineTemplate` from an
6524 existing (perhaps static) set that best matches a `VirtualHardwareSection`. Items in the
6525 `VirtualHardwareSection` are mapped to CIM `MachineConfiguration` properties and the
6526 corresponding `MachineConfiguration` Resource is created and linked to from the created
6527 `MachineTemplate` for that `VirtualSystem`.

6528 The CIM `VolumeTemplates` are created according to the `DiskSection` of an OVF descriptor and
6529 can be shared among more than one `VirtualSystem` (CIM `MachineTemplates`) defined in an
6530 OVF package. In addition, a new CIM `MachineImage` Resource may be created from the
6531 `DiskSection` if an `ovf:fileRef` for the virtual disk content is specified.

6532 The CIM `NetworkTemplates` are created according to the `NetworkSection` of an OVF descriptor
6533 along with the `Connection` elements in the `VirtualHardwareSection` elements that refer to
6534 these named networks.

6535 Clause 5.13.2.1 details how to import an OVF file to create a `System` (and component `Resources`). The
6536 `System` thus created contains a reference to a `Machine` for every `VirtualSystem` that is defined in
6537 an OVF descriptor `VirtualSystemCollection`. Note that CIM currently allows `Systems of`
6538 `Systems`, so for each `VirtualSystemCollection` encountered in a nested set of collections, a

6539 separate System is created within the parent System with Machines for each of the contained
6540 VirtualSystems in that VirtualSystemCollection.

6541 The values of the attributes for the Machine are taken from the VirtualHardwareSection of the
6542 VirtualSystem description (required in OVF). If more than one VirtualHardwareSection is
6543 used for a given VirtualSystem (allowed in OVF), the result is implementation dependent. Items in
6544 the VirtualHardwareSection are mapped to CIMI MachineConfiguration properties and
6545 the corresponding MachineConfiguration Resource is created and linked to from the created
6546 Machine for that VirtualSystem.

6547 The CIMI Volumes are created according to the DiskSection of an OVF descriptor and can be
6548 shared among more than one VirtualSystem (CIMI Machines) defined in an OVF package. In
6549 addition, a new CIMI MachineImage Resource may be created from the DiskSection if an
6550 ovf:fileRef attribute for the virtual disk content is specified.

6551 The CIMI Networks are created according to the NetworkSection of an OVF descriptor along with
6552 the Connection elements in the VirtualHardwareSection that refer to these named networks.

6553

6554 **ANNEX B**
6555 **(informative)**
6556 **XML Schema**

6557 The XML Schema for the XML serialization of the CIMI model can be found at:

6558 http://schemas.dmtf.org/cimi/2/dsp8009_1.0.xsd

6559 The schema provided does not intend to reflect every single modeling constraint and requirement
6560 specified in the model. This schema is designed to apply more broadly to any model-related serialized
6561 material found in Consumer requests as well as in Provider responses, and is intended to provide a
6562 preliminary, non-exhaustive syntactic check on these. In particular, future updates of this specification
6563 may intermix new XML elements into the Resources using the current CIMI namespace to Resources.
6564 The schema that is provided is just a starting point for those who would find it useful and it might need to
6565 be modified based on specific application's needs.

**ANNEX C
(informative)
Change log**

6566
6567
6568

6569

Version	Date	Description
1.0.0a	2012-08-28	DMTF Draft Standard
1.0.1a	2012-09-12	DMTF Draft Standard
1.1.0a	2013-07-22	DMTF Work in Progress release
1.1.0	2013-10-22	DMTF Draft Standard
2.0.0a	2014-09-24	DMTF Work in Progress release
2.0.0b	2014-11-05	DMTF Work in Progress release
2.0.0c	2015-03-20	DMTF Work in Progress release

6570
6571

Bibliography

6572

6573 DMTF Standard: *Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based*
6574 *Protocol* specification V1.0 (DSP0263)

6575 http://dmtf.org/sites/default/files/standards/documents/DSP0263_1.0.0.pdf

6576 DMTF Standard: *Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based*
6577 *Protocol* specification V1.1 (DSP0263)

6578 https://members.dmtf.org/apps/org/workgroup/cmwg/download.php/73648/DSP0263_1.1.0b_RC2.pdf