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

Document Type: Specification
Document Status: DMTF Standard
Document Language: en-US
DMTF is a not-for-profit association of industry members dedicated to promoting enterprise and systems management and interoperability. Members and non-members may reproduce DMTF specifications and documents, provided that correct attribution is given. As DMTF specifications may be revised from time to time, the particular version and release date should always be noted.

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patent owner or claimant, and shall have no liability or responsibility for costs or losses incurred if a standard is withdrawn or modified after publication, and shall be indemnified and held harmless by any party implementing the standard from any and all claims of infringement by a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

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

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>Foreword</td>
<td>7</td>
</tr>
<tr>
<td>35</td>
<td>1 Scope</td>
<td>9</td>
</tr>
<tr>
<td>36</td>
<td>1.1 Document structure</td>
<td>9</td>
</tr>
<tr>
<td>37</td>
<td>1.2 Document versioning scheme</td>
<td>9</td>
</tr>
<tr>
<td>38</td>
<td>1.3 Typographical conventions</td>
<td>9</td>
</tr>
<tr>
<td>39</td>
<td>2 Normative references</td>
<td>10</td>
</tr>
<tr>
<td>40</td>
<td>3 Terms and definitions</td>
<td>11</td>
</tr>
<tr>
<td>41</td>
<td>4 HTTP-based protocol</td>
<td>14</td>
</tr>
<tr>
<td>42</td>
<td>4.1 Introduction</td>
<td>14</td>
</tr>
<tr>
<td>43</td>
<td>4.1.1 Protocol evolution and client expectations</td>
<td>14</td>
</tr>
<tr>
<td>44</td>
<td>4.1.2 XML namespaces</td>
<td>14</td>
</tr>
<tr>
<td>45</td>
<td>4.1.3 URI space</td>
<td>14</td>
</tr>
<tr>
<td>46</td>
<td>4.1.4 Media types</td>
<td>15</td>
</tr>
<tr>
<td>47</td>
<td>4.1.5 Request headers</td>
<td>15</td>
</tr>
<tr>
<td>48</td>
<td>4.1.6 Request query parameters</td>
<td>15</td>
</tr>
<tr>
<td>49</td>
<td>4.2 Protocol operations</td>
<td>21</td>
</tr>
<tr>
<td>50</td>
<td>4.2.1 Common CRUD operations</td>
<td>22</td>
</tr>
<tr>
<td>51</td>
<td>4.3 OVF support</td>
<td>29</td>
</tr>
<tr>
<td>52</td>
<td>5 Model</td>
<td>30</td>
</tr>
<tr>
<td>53</td>
<td>5.1 Resource wrappers</td>
<td>30</td>
</tr>
<tr>
<td>54</td>
<td>5.2 Extensibility</td>
<td>31</td>
</tr>
<tr>
<td>55</td>
<td>5.3 Identifiers</td>
<td>31</td>
</tr>
<tr>
<td>56</td>
<td>5.4 Attribute constraints</td>
<td>32</td>
</tr>
<tr>
<td>57</td>
<td>5.5 Data types and their serialization</td>
<td>33</td>
</tr>
<tr>
<td>58</td>
<td>5.5.1 boolean</td>
<td>33</td>
</tr>
<tr>
<td>59</td>
<td>5.5.2 dateTime</td>
<td>33</td>
</tr>
<tr>
<td>60</td>
<td>5.5.3 duration</td>
<td>33</td>
</tr>
<tr>
<td>61</td>
<td>5.5.4 integer</td>
<td>34</td>
</tr>
<tr>
<td>62</td>
<td>5.5.5 string</td>
<td>34</td>
</tr>
<tr>
<td>63</td>
<td>5.5.6 ref</td>
<td>34</td>
</tr>
<tr>
<td>64</td>
<td>5.5.7 map</td>
<td>35</td>
</tr>
<tr>
<td>65</td>
<td>5.5.8 structure</td>
<td>35</td>
</tr>
<tr>
<td>66</td>
<td>5.5.9 byte[]</td>
<td>36</td>
</tr>
<tr>
<td>67</td>
<td>5.5.10 URI</td>
<td>36</td>
</tr>
<tr>
<td>68</td>
<td>5.5.11 Arrays</td>
<td>36</td>
</tr>
<tr>
<td>69</td>
<td>5.5.12 Collections</td>
<td>37</td>
</tr>
<tr>
<td>70</td>
<td>5.5.13 &quot;Any&quot; type</td>
<td>41</td>
</tr>
<tr>
<td>71</td>
<td>5.5.14 Empty attribute values</td>
<td>41</td>
</tr>
<tr>
<td>72</td>
<td>5.6 Units</td>
<td>41</td>
</tr>
<tr>
<td>73</td>
<td>5.7 Relationship semantics</td>
<td>42</td>
</tr>
<tr>
<td>74</td>
<td>5.8 Operations</td>
<td>42</td>
</tr>
<tr>
<td>75</td>
<td>5.9 Alternative model formats</td>
<td>43</td>
</tr>
<tr>
<td>76</td>
<td>5.10 Resources</td>
<td>43</td>
</tr>
<tr>
<td>77</td>
<td>5.10.1 Common attributes</td>
<td>43</td>
</tr>
<tr>
<td>78</td>
<td>5.11 Resource metadata</td>
<td>45</td>
</tr>
<tr>
<td>79</td>
<td>5.11.1 Serialization of attribute value constraints</td>
<td>49</td>
</tr>
<tr>
<td>80</td>
<td>5.11.2 Capabilities</td>
<td>51</td>
</tr>
<tr>
<td>81</td>
<td>5.11.3 ResourceMetadataCollection Resource</td>
<td>54</td>
</tr>
<tr>
<td>82</td>
<td>5.12 Cloud Entry Point</td>
<td>55</td>
</tr>
<tr>
<td>83</td>
<td>5.12.1 Operations</td>
<td>61</td>
</tr>
<tr>
<td>84</td>
<td>5.13 System Resources and relationships</td>
<td>61</td>
</tr>
<tr>
<td>85</td>
<td>5.13.1 System</td>
<td>62</td>
</tr>
</tbody>
</table>
5.14 Machine Resources and relationships
5.14.1 Machine ................................................. 88
5.14.2 MachineCollection ..................................... 107
5.14.3 MachineTemplate ....................................... 109
5.14.4 MachineTemplateCollection Resource .............. 116
5.14.5 MachineConfiguration Resource ...................... 117
5.14.6 MachineConfigurationCollection Resource .......... 119
5.14.7 MachineImage Resource ................................ 120
5.14.8 MachineImageCollection Resource ................. 124
5.14.9 Credential Resource .................................... 125
5.14.10 CredentialCollection Resource ....................... 126
5.14.11 CredentialTemplate Resource ......................... 127
5.14.12 CredentialTemplateCollection Resource .......... 128

5.15 Volume Resources and relationships
5.15.1 Volume .................................................... 130
5.15.2 VolumeCollection Resource ............................ 135
5.15.3 VolumeTemplate Resource ............................... 136
5.15.4 VolumeTemplateCollection Resource .................. 138
5.15.5 VolumeConfiguration Resource ......................... 139
5.15.6 VolumeConfigurationCollection Resource ............ 141
5.15.7 VolumeImage Resource .................................. 142
5.15.8 VolumeImageCollection Resource ...................... 144

5.16 Network Resources and relationships
5.16.1 Network ................................................. 145
5.16.2 NetworkCollection Resource ............................ 153
5.16.3 NetworkTemplate Resource ................................ 154
5.16.4 NetworkTemplateCollection Resource .................. 156
5.16.5 NetworkConfiguration Resource ......................... 157
5.16.6 NetworkConfigurationCollection Resource .......... 158
5.16.7 NetworkPort ............................................. 160
5.16.8 NetworkPortCollection Resource ....................... 164
5.16.9 NetworkPortTemplate Resource ......................... 165
5.16.10 NetworkPortTemplateCollection Resource .......... 168
5.16.11 NetworkPortConfiguration Resource .................. 169
5.16.12 NetworkPortConfigurationCollection Resource .... 170
5.16.13 Address Resource ....................................... 171
5.16.14 AddressCollection Resource .......................... 173
5.16.15 AddressTemplate Resource ............................. 174
5.16.16 AddressTemplateCollection Resource ................. 176
5.16.17 ForwardingGroup Resource ............................ 177
5.16.18 ForwardingGroupCollection Resource ................. 180
5.16.19 ForwardingGroupTemplate Resource .................. 181
5.16.20 ForwardingGroupTemplateCollection Resource .... 182

5.17 Monitoring Resources and relationships
5.17.1 Job Resource ............................................ 183
5.17.2 JobCollection Resource .................................. 184
5.17.3 Meter Resource .......................................... 188
5.17.4 MeterCollection Resource ............................... 189
5.17.5 MeterTemplate Resource .................................. 195
5.17.6 MeterTemplateCollection Resource ..................... 196
5.17.7 MeterConfiguration Resource ........................... 197
5.17.8 MeterConfigurationCollection Resource ............... 198
5.17.9 EventLog Resource ........................................ 201
5.17.10 MeterEventCollection Resource ....................... 202
6 Security considerations ........................................................................................................... 216
ANNEX A (normative) OVF support in CIMI .............................................................................. 217
ANNEX B (informative) XML Schema ....................................................................................... 219
ANNEX C (informative) Change log ......................................................................................... 220

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

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

Table 28 – Credential attributes ................................................................. 125
Table 29 – UserName/Password attributes .................................................. 125
Table 30 – Public key attributes ................................................................. 125
Table 31 – CredentialTemplate attributes .................................................. 127
Table 32 – Volume attributes ..................................................................... 131
Table 33 – VolumeVolumeImage attributes ............................................... 133
Table 34 – VolumeTemplate attributes ....................................................... 136
Table 35 – VolumeConfiguration attributes ............................................... 140
Table 36 – VolumeImage attributes ............................................................. 142
Table 37 – Network attributes ................................................................... 145
Table 38 – NetworkTemplate attributes ..................................................... 154
Table 39 – NetworkConfiguration attributes .............................................. 157
Table 40 – NetworkPort attributes ............................................................. 160
Table 41 – NetworkPortTemplate attributes .............................................. 165
Table 42 – NetworkPortConfiguration attributes ....................................... 169
Table 43 – Address attributes ................................................................... 171
Table 44 – AddressTemplate attributes ..................................................... 174
Table 45 – ForwardingGroup attributes ....................................................... 178
Table 46 – ForwardingGroupNetwork attributes ......................................... 179
Table 47 – ForwardingGroupTemplate attributes ....................................... 181
Table 48 – Job attributes ........................................................................... 185
Table 49 – Meter attributes ....................................................................... 189
Table 50 – Sample attributes ...................................................................... 192
Table 51 – MeterTemplate attributes .......................................................... 196
Table 52 – MeterConfiguration attributes .................................................. 198
Table 53 – aspect URIs ............................................................................... 200
Table 54 – EventLog attributes ................................................................... 202
Table 55 – EventLogTemplate attributes ..................................................... 206
Table 56 – Event attributes ........................................................................ 208
Table 57 – type URIs .................................................................................. 211

186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215
Foreword

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

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

Acknowledgments

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

Editors (past and present):
- Marios Andreou – Red Hat
- Doug Davis – IBM
- Jacques Durand – Fujitsu
- Gilbert Pilz – Oracle

Contributors:
- Ghazanfar Ali – ZTE Corporation
- Marios Andreou – Red Hat
- Keith Bankston – Microsoft Corporation
- Winston Bumpus – VMware Inc.
- Nathan Burkhart – Microsoft Corporation
- Mark Carlson – Oracle
- Steve Carter – Novell
- Junsheng Chu – ZTE Corporation
- Josh Cohen – Microsoft Corporation
- Derek Coleman – Hewlett-Packard Company
- John Crandall – Brocade Communications Systems
- Doug Davis – IBM
- Jim Davis – WBEM Solutions
- Fernando de la Iglesia – Telefónica
- Hiroshi Dempo – NEC Corporation
- Jacques Durand – Fujitsu
- Yigal Edery – Microsoft Corporation
- George Ericson – EMC
- Colleen Evans – Microsoft Corporation
- Norbert Floeren – Ericsson AB
- Robert Freund – Hitachi, Ltd.
- Fermín Galán – Telefónica
- Krishnan Gopalan – Microsoft Corporation
- Kazunori Iwasa – Fujitsu
- Mark Johnson – IBM
- Bhumip Khasnabish – ZTE Corporation
- Dies Köper – Fujitsu
- Vincent Kowalski – BMC Software
- Ruby Krishnaswamy – France Telecom Group
- Lawrence Lamers – VMware Inc.
- Paul Lipton – CA Technologies
- James Livingston – NEC Corporation
- Vince Lubsey – Virtustream Inc.
• David Lutterkort – Red Hat
• Fred Maciel – Hitachi, Ltd.
• Andreas Maier – IBM
• Ashok Malhotra – Oracle
• Jeff Mischkinsky – Oracle
• Jesus Molina – Fujitsu
• Efraim Moscovich – CA Technologies
• Bryan Murray – Hewlett-Packard Company
• Steven Neely – Cisco
• Ryuichi Ogawa – NEC Corporation
• John Parchem – Microsoft Corporation
• Shishir Pardikar – Citrix Systems Inc.
• Miguel Peñalvo – Telefónica
• Gilbert Pilz – Oracle
• Alvaro Polo – Telefónica
• Enrico Ronco – Telecom Italia
• Federico Rossini – Telecom Italia
• Matthew Rutkowski – IBM
• Tom Rutt – Fujitsu
• Hemal Shah – Broadcom
• Nihar Shah – Microsoft Corporation
• Alan Sill – Texas Tech University
• Zhexuan Song – Huawei
• Marvin Waschke – CA Technologies
• Eric Wells – Hitachi, Ltd.
• Jeff Wheeler – Huawei
• Maarten Wiggers – Fujitsu
• Daniel Wilson – Ericsson AB
• Steve Winkler – SAP AG
• Jack Yu – Oracle
• Aaron Zhang – Huawei
• HengLiang Zhang – Huawei
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

1 Scope

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

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

1.1 Document structure

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

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

1.2 Document versioning scheme

This document adheres to the versioning scheme defined in clause 6.3 of DSP4004.

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

1.3 Typographical conventions

This specification uses the following conventions:

In the narrative text of the specification:

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

attribute, operation.

Inside tables describing the Resource data model:

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

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

- Values in italics indicate data types instead of literal values.
- Characters are appended to items to indicate cardinality:
  - "?" (0 or 1)
  - "*" (0 or more)
  - "+" (1 or more)
- Vertical bars, "|", denote choice. For example, "a|b" means a choice between "a" and "b".
- Parentheses, "(" and ")", are used to indicate the scope of the operators "?", "*", "+" and "|".
- Ellipses (i.e., "...") indicate points of extensibility. Note that the lack of an ellipses does not mean no extensibility point exists, rather it is just not explicitly called out - usually for the sake of brevity.

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

2 Normative references

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

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

DMTF DSP0243, Open Virtualization Format Specification 1.1,

DMTF DSP1001, Management Profile Specification Usage Guide 1.1,

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

3 Terms and definitions

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

The terms "shall" ("required"), "shall not," "should" ("recommended"), "should not" ("not recommended"), "may," "need not" ("not required"), "can" and "cannot" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Annex H. The terms in parenthesis are alternatives for the preceding term, for use in exceptional cases when the preceding term cannot be used for linguistic reasons. Note that ISO/IEC Directives, Part 2, Annex H specifies additional alternatives. Occurrences of such additional alternatives shall be interpreted in their normal English meaning.
The terms "clause," "subclause," "paragraph," and "annex" in this document are to be interpreted as described in ISO/IEC Directives, Part 2, Clause 5.

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

The terms defined in DSP4004, DSP0223, and DSP1001 apply to this document. The following additional terms are used in this document.

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

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

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

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

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

3.5 Cloud Service Provider
A category of actors that includes the Service Operations Manager (who manages the technical infrastructure required for providing cloud services; monitors and measures performance and utilization against SLAs; provides reports from monitoring and measurement; etc.); Service Business Manager (who offers all types of services developed by cloud service developers; accounts for services potentially offered by service Providers themselves and services offered on behalf of cloud service developers; establishes a portfolio of business relationships; and sets up accounts and terms for Consumers, etc.); and Service Transition Manager (who enables a customer to use the cloud service, including "onboarding", integration, and process adoption; defines and creates service offerings based on Templates and Configurations that can be used by Consumers and are populated into the catalog; etc.). The term "Provider" is used if the indicated action or activity could involve one or more of the above actors. In cases where the distinction between the actors in the category is relevant, the more detailed term is used.
For purposes of comparison and alignment, it should be noted that a Cloud Service Provider is equivalent to the “Cloud Provider” actor defined in the NIST Reference Architecture [SP500-292].

### 3.6 Collection

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

### 3.7 Configuration

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

### 3.8 Infrastructure as a Service (IaaS)

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

### 3.9 Message confidentiality

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

### 3.10 Message integrity

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

### 3.11 Resource

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

### 3.12 Template

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

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

1. By reference - allow Consumers to reference a Template (that exists as a Resource in the Provider) as part of the instantiation operation.

2. By value - allow Consumers to dynamically provide the Template information as part of the instantiation operation.

3. Reference with overrides - allow Consumers to reference a Template (that exists as a Resource in the Provider) and provide additional values that override the attributes of that Template as part of the instantiation operation.
4 HTTP-based protocol

4.1 Introduction

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

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

4.1.1 Protocol evolution and client expectations

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

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

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

4.1.2 XML namespaces

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

<table>
<thead>
<tr>
<th>Prefix</th>
<th>XML Namespaces</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>cimi</td>
<td><a href="http://schemas.dmtf.org/cimi/1">http://schemas.dmtf.org/cimi/1</a></td>
<td>This specification</td>
</tr>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>XML Schema Part2</td>
</tr>
</tbody>
</table>

4.1.3 URI space

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

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

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

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

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

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

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

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

4.1.5 Request headers

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

4.1.6 Request query parameters

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

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

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

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

```
?$filter=expression
```

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

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

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

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

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

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

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

Examples:

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

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

```
/machines
```
The URI to a Machine is as follows:

/machines/123

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

/machines/123/disks

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

/machines/123/volumes

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

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

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

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

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

4.1.6.2 Subsetting Collections

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

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

)?$first=number

)?$last=number

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

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

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

4.1.6.3 Subsetting Resources

If retrieving the representation of a Resource, Consumers may include the $select query parameter to specify a subset of the Resource to be acted upon. Providers shall interpret and process this query parameter as described in this section. This subsetting shall have the semantic equivalence of
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

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

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

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

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

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

is equivalent to:

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

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

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

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

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

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

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

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

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

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

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

Optionally, an implementation may also support the alternative attribute name notation:

```plaintext
<collectionName>/<attributeName>
```

for subsetting the items inside a collection. For example, the following subsetting on items of a Disks Collection is equivalent to the one done in the previous example, while in addition listing the operations of the Collection resource itself (not of its items):

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

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

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

The serialization shall be performed as follows:

**JSON serialization:**

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

shall be expanded to be:

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

**XML serialization:**

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

shall be expanded to be:

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

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

The format of a $expand query parameter shall be:

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

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

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

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

```
?$expand=volumes
```

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

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

The $format parameter shall be of the form:

```
?$format=encoding
```

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

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

4.1.6.6 Sorting Collections

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

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

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

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

The sort shall be performed based on the attribute type.

The following rules apply to the ascending sort order:

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

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

Examples:

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

```
GET /machines?$orderby=created:desc
```
To sort the result set of the MachinesCollection Resource on the "cpu" attribute in descending order, followed by the "memory" attribute in ascending order, the following expression would be used:

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

### 4.1.6.7 Response headers

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

### 4.1.6.8 Job header

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

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

In cases where an error occurs during the processing of a request, the Provider shall include a representation of a Job Resource describing the status of the failed operation. This representation of a Job shall be included even in cases where the Provider does not normally support Job Resources to ensure that Consumers are provided with sufficient information, in a consistent manner, as to the reason for the failure regardless of whether the Provider supports Jobs. If Jobs are not supported in general, any of the references in the Job representation (e.g., "id" or the "href" for nestedJobs) shall be empty paths (i.e., "") and the nestedJobs array shall be expanded (see 4.1.6.4) to inline the representation of the pseudo subordinate Jobs.

### 4.1.6.9 ETag support

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

### 4.2 Protocol operations

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

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

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

XML serialization:

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

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

JSON serialization:

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

XML serialization:

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

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

### 4.2.1 Common CRUD operations

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

#### 4.2.1.1 Creating a new Resource

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

The HTTP POST request shall include:

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

For example, the request can be:

```
POST <addURI> HTTP/1.1
Host: <hostname>
```
This example has an Accept header with one of the CIMI supported media types: application/json or application/xml. If the Provider chooses to reply with a serialization, this serialization should be of the specified media type. Omission of the Accept header allows the Provider to reply with a serialization of any media type. If the Resource has a "State" attribute, its value shall be "CREATING" while the Provider is processing this operation.

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

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

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

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

1. Resource creation by passing a Template by value:

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

2. Resource creation by passing a template by reference:

   ```json
   { "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceCreate",
   "name": "myResourceName", ?
   "description": "My resource description", ?
   "properties": { "prop1name": "prop1value" }, + }, ?
   "resourceTemplate": { "href": "string",
       "here some template attribute/value pairs may be added to override values in the referenced template>?
   }
   ```
In case the created Resource is itself a Template, only the first creation pattern - by value - applies.

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

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

Direct setting of attributes in the new Resource:

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

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

Defining or referring to the Resource Template:

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

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

- Any top-level attribute of complex or simple type in the referred Template shall be overridden by providing its name/value pair in the create request inside the resourceTemplate element and immediately under it. For a top-level attribute of complex type (e.g., arrays, Collections, structures), the provided complex value shall also set all underlying attributes – e.g., array elements.

- The semantics shall be same as of modifying (overriding) parts of the referred Template just before it is used for instantiation, but these overrides shall not persist in the referred Template and shall only concern this particular instantiation.

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

"attribute": null

for the attribute in the JSON serialization, or

<attribute/>

in the XML serialization for that attribute.

Examples:

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

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

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

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

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

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineCreate ",
 "name": "myMachine456",
 "description": "A machine connected to a pre-existing volume",
 "machineTemplate": { "href": "http://example.com/machineTemplates/72000" ,
   "credential": { "href": "http://example.com/myCredential" } ,
   "networkInterfaces": [ { "addresses": [ { "address": { "href": "http://example.com/addresses/add4" } } ],
   "network": { "href": "http://example.com/networks/net1" } ,
   "state": "ACTIVE" } ]
 }
```

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

- **Instance-level attribute settings**: these shall directly update similar attributes in the created `Resource`, here `name` and `description`.

- **Template-level overrides**: The referred `MachineTemplate` is used for creating the `Machine`, but the `credential` attribute in this Template is (temporarily) overridden by the `credential` provided in the creation request as is the `networkInterfaces` array. In case such attributes were not present in the referred Template, they are added (temporarily) just for this `Machine` creation.

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

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

- **HTTP Location header** with a reference to the new `Resource"
If the response to a create request includes a serialization of the new Resource, the response shall additionally include:

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

For example, the response can be:

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

4.2.1.2 Retrieving a representation of a Resource

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

For example, the request can be:

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

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

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

For example, the response can be:

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

4.2.1.3 Updating a Resource

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

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

While processing a PUT request, if the server detects that an attempt is being made to update a read-only, or immutable, attribute, it shall silently ignore that attribute update request and shall not generate an error. This rule applies to Resource partial updates as well.
Because of potential conflicts that might occur due to multiple concurrent updates, Consumers should use the partial update mechanism, defined in 4.2.1.3.1, to reduce the chances of mistakenly updating attributes with out-of-date data.

The HTTP PUT request shall include:

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

For example, the request can be:

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

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

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

For example, the response can be:

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

4.2.1.3.1 Partial updates to a Resource

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

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

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

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

Any attribute listed in the URI but not included within the HTTP request body shall be reset to a Resource specific value (e.g., removed).
From an HTTP perspective, the updated subsetted Resource is a distinct one. The semantics of a normal HTTP PUT are adhered to; it is a complete replacement update of the specified Resource. From the Consumer’s perspective, the partial update is interpreted and executed by the Cloud Service Provider, and some part of the Resource is changed.

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

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

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

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

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

4.2.1.4 Deleting a Resource

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

For example, the request can be:

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

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

For example, the response can be:

```
HTTP/1.1 200 OK
```

4.2.1.5 Other operations

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

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

The request shall be of the following form:

```
POST <operationLinkURI> HTTP/1.1
```
The form of the response varies depending on the operation and is defined by the operation itself.

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

### 4.2.1.6 Synchronous operations

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

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

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

### 4.2.1.7 Asynchronous operations

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

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

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

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

### 4.3 OVF support

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

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

OVF support is covered in more detail in ANNEX A.

5 Model

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

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

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

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

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

5.1 Resource wrappers

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

JSON serialization:

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

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

XML serialization:

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

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

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

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

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

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

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

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

5.3 Identifiers

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

- Identifier names shall be treated as case sensitive.
- Identifier names shall only use the following set of characters:
  - Uppercase ASCII (U+0041 through U+005A)
  - Lowercase ASCII (U+061 through U+007A)
  - Digits (U+0030 through U+0039)
  - Underscore (U+005F)
- Identifier names shall not start with a Digit (U+0030 through U+0039).
Note that these rules do not apply to the "name" common attribute defined in clause 5.10.1.

5.4 Attribute constraints

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

support optional:

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

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

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

support mandatory:

This constraint indicates that support for this attribute is required by compliant implementations. If present on a nested attribute, this attribute is required to be supported only if the parent attribute is supported.

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

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

immutable:

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

mutable:

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

read-only:

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

read-write:

This Consumer constraint indicates that the attribute may be retrieved and/or updated by Consumers. Read-write attributes shall appear in the serialization of Resources sent to and from Providers.Providers may further constrain whether Consumers can update these attributes and should indicate this by the way of ResourceMetadata.
This Consumer constraint indicates that the attribute may be updated by Consumers but are not retrievable by Consumers, typically for security reasons. Write-only attributes shall appear in the serialization of Resources sent to Providers but shall never appear in the serialization of Resources sent from Providers.

### 5.5 Data types and their serialization

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

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

#### 5.5.1 boolean

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

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

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

#### 5.5.2 dateTime

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

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

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

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

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

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

#### 5.5.3 duration

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

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

If serialized in XML these values shall be of XML Schema type: `xs:duration`
5.5.4 integer
A value as defined by xs:integer per XML Schema – Part 2. Any constraints on the specific ranges allowed for any particular attribute shall be specified by that attribute's definition or at runtime by the Provider by the way of the metadata discovery mechanisms defined by this specification.

If serialized in JSON these values shall be of JSON type: number
If serialized in XML these values shall be of XML Schema type: xs:integer

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

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

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

5.5.6 ref
A reference to another Resource.

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

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

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

"myvolume": { "href": string }

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

<myvolume href="xs:anyURI"/>

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

For JSON:

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

and in XML:
However, for brevity the extensibility points are excluded from the serialization of the Resources.

5.5.7 map

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

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

5.5.8 structure

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>low</td>
<td>number</td>
</tr>
<tr>
<td>medium</td>
<td>number</td>
</tr>
<tr>
<td>high</td>
<td>number</td>
</tr>
<tr>
<td>critical</td>
<td>number</td>
</tr>
</tbody>
</table>

JSON serialization:

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

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

XML serialization:

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

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

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

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

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

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

5.5.10  URI

The format and syntax of the attributes of type "URI" is defined by RFC3986.

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

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

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

<table>
<thead>
<tr>
<th>Base URI</th>
<th>Relative URI</th>
<th>Absolute URI</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://example.com/">http://example.com/</a></td>
<td>p1/file</td>
<td><a href="http://example.com/p1/file">http://example.com/p1/file</a></td>
</tr>
</tbody>
</table>

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

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

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

5.5.11  Arrays

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

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

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

JSON serialization:

Within this specification, arrays in JSON are serialized with a wrapper property. The wrapper name shall
be same as the attribute name for the array. For example, a "things" attribute of type "thing[]" is
serialized as:
If the items in the array are structures, the structure name shall not be present in the JSON serialization.

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

```json
"things": [  
  { "href": string }, +
]?
```

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

**XML serialization:**

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

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

There is no wrapper element for an array in XML.

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

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

### 5.5.12 Collections

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

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

Attributes that are Collections are represented as type "collection[itemType]." The Resource type of the Collection items are specified inside the brackets; for example an attribute that is a Collection of Machines is expressed as "collection[Machine]." These are serialized as a reference to a Collection Resource. For brevity, while these attributes are "references" the word "ref" or "reference" does not appear in the model definition tables - simply the type "collection[itemType]" appears.
To each one of these Resource items, shall correspond an entry in the Collection. These Resources items are assumed to be of a complex type and are separately addressable and manageable. While different Collections contain entries of different Resource types, all Collections follow the pattern described below:

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

The Resources in a Collection are of two kinds:

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

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

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

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

The serialization of Collections shall adhere to the following pattern:

**JSON serialization:**

```json
{
  "resourceURI": string,
  "id": string,
  ...
}
```
"count": number,
"resourceSpecificGroupingName": [
  { "resourceURI": string,
    "id": string,
    "name": string, ?
    "description": string, ?
    "created": string, ?
    "updated": string, ?
    "properties": { string: string, + }, ?,
    ... entry specific data ...
    "operations": [
      { "rel": "edit", "href": string }, ?,
      { "rel": "delete", "href": string } ?
    ] ?
    ...
  } +
  
], ?,
"operations": [ { "rel": "add", "href": string } ? ]

XML serialization:

<Collection resourceURI="xs:anyURI" xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <ResourceSpecificElementName>
    <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> *
    ... entry specific data ...
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </ResourceSpecificElementName>*
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
Where the `resourceURI` attributes shall contain the Collection or Resource specific URIs for that type of Collection, and `resourceSpecificGroupingName` and `resourceSpecificElementName` shall be replaced with the name of the Collection-specific Resource name, e.g., `machines` in JSON or `Machine` in XML.

### 5.5.12.1 Adding items to Collections

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

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

**JSON serialization:**

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

**XML serialization:**

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

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

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

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

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

**JSON serialization:**

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

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

The MachineCollection has a new Machine:

JSON serialization:

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

XML serialization:

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

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

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

5.5.13 "Any" type

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

5.5.14 Empty attribute values

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

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

5.6 Units

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

### Table 4 – Numerical equivalents for attributes

<table>
<thead>
<tr>
<th>String</th>
<th>Numerical Value</th>
<th>String</th>
<th>Numerical Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>kilobyte</td>
<td>10^3</td>
<td>kibibyte</td>
<td>2^10</td>
</tr>
<tr>
<td>megabyte</td>
<td>10^6</td>
<td>mebibyte</td>
<td>2^20</td>
</tr>
<tr>
<td>gigabyte</td>
<td>10^9</td>
<td>gibibyte</td>
<td>2^30</td>
</tr>
<tr>
<td>terabyte</td>
<td>10^12</td>
<td>tebibyte</td>
<td>2^40</td>
</tr>
<tr>
<td>petabyte</td>
<td>10^15</td>
<td>pebibyte</td>
<td>2^50</td>
</tr>
<tr>
<td>exabyte</td>
<td>10^18</td>
<td>exbibyte</td>
<td>2^60</td>
</tr>
<tr>
<td>zettabyte</td>
<td>10^21</td>
<td>zebibyte</td>
<td>2^70</td>
</tr>
<tr>
<td>yottabyte</td>
<td>10^24</td>
<td>yobibyte</td>
<td>2^80</td>
</tr>
</tbody>
</table>

#### 5.7 Relationship semantics

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

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

#### 5.8 Operations

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

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

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

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

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

- A new operation that starts from a CIMI-defined state, or leads to a CIMI-defined state, or both. In the latter case, if a CIMI-defined operation already exists for this transition between two CIMI-defined states, it shall also be supported by the Provider in addition to the new operation.

- A new Resource state. In that case, a new operation that leads to that state shall also be created. In other words, a Provider-defined operation has to be performed before a Provider-defined state can be reached."
c) A new operation that transitions between two Provider-defined states.

5.9 Alternative model formats

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

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

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

5.10 Resources

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

5.10.1 Common attributes

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

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

Table 5 – Common attributes

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>updated</td>
<td><code>dateTime</code></td>
<td>The time at which the last explicit attribute update was made on the Resource. Note, while operations, such as &quot;stop&quot;, do implicitly modify the 'state' attribute, they do not change the 'updated_time'. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>properties</td>
<td><code>map</code></td>
<td>A map of key/value pairs (each entry called a &quot;property&quot;), 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 &quot;key&quot; shall not be used more than once within a &quot;properties&quot; attribute. Each property shall contain the following nested data:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td><code>string</code></td>
<td>The name of the property. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>value</td>
<td><code>string</code></td>
<td>The value of the property. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
</tbody>
</table>

### Constraints for primary and secondary Resources:

**Provider:** support mandatory; mutable  
**Consumer:** support optional; read-write

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

#### JSON serialization:

```json
"id": string,  
"name": string, ?  
"description": string, ?  
"created": string, ?  
"updated": string, ?  
"properties": { string: string, + }, ?
```

#### XML serialization:

```xml
<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> *
```
5.11 Resource metadata

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

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

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

Table 6 – ResourceMetadata attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>ResourceMetadata</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/ResourceMetadata">http://schemas.dmtf.org/cimi/1/ResourceMetadata</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>id</td>
<td>URI</td>
<td>The unique URI identifying this Resource; assigned upon Resource creation. This attribute value is immutable, and shall be unique in the Provider’s cloud. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td>typeURI</td>
<td>URI</td>
<td>A unique URI associated with, and denoting, the described Resource type. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
<td>The name of the described Resource type. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>attributes</td>
<td>attribute[]</td>
<td>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:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Type</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
</tr>
<tr>
<td>namespace</td>
<td>URI</td>
</tr>
<tr>
<td>type</td>
<td>string</td>
</tr>
<tr>
<td>required</td>
<td>boolean</td>
</tr>
<tr>
<td>Name</td>
<td>ResourceMetadata</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/ResourceMetadata">http://schemas.dmtf.org/cimi/1/ResourceMetadata</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>value constraints</td>
<td>any</td>
<td>Type-specific data that describes any constraints on values of this attribute. If absent, there are no constraints. Note that the serialization of these &quot;value constraints&quot; shall be determined by the type of the attribute; see clause 5.11.1.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Name</th>
<th>capability[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>A set of Provider-defined metadata that can be used by Consumer to discover any capability or feature provided by this Provider. Each capability shall contain the following nested data:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>capability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Type</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
</tr>
<tr>
<td>Constraints:</td>
<td></td>
</tr>
<tr>
<td>Provider:</td>
<td>support mandatory; mutable</td>
</tr>
<tr>
<td>Consumer:</td>
<td>support optional; read-write</td>
</tr>
<tr>
<td>uri</td>
<td>URI</td>
</tr>
<tr>
<td>Constraints:</td>
<td></td>
</tr>
<tr>
<td>Provider:</td>
<td>support mandatory; mutable</td>
</tr>
<tr>
<td>Consumer:</td>
<td>support mandatory; read-write</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
</tr>
<tr>
<td>Constraints:</td>
<td></td>
</tr>
<tr>
<td>Provider:</td>
<td>support mandatory; mutable</td>
</tr>
<tr>
<td>Consumer:</td>
<td>support optional; read-write</td>
</tr>
<tr>
<td>value</td>
<td>any</td>
</tr>
<tr>
<td>Constraints:</td>
<td></td>
</tr>
<tr>
<td>Provider:</td>
<td>support mandatory; mutable</td>
</tr>
<tr>
<td>Consumer:</td>
<td>support mandatory; read-write</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Name</th>
<th>action[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 &quot;actions&quot; so as not to conflict with the ResourceMetadata Resource's own operations. Each operation shall contain the following nested data:</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>action</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Type</td>
</tr>
<tr>
<td>name</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>ResourceMetadata</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/ResourceMetadata">http://schemas.dmtf.org/cimi/1/ResourceMetadata</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>uri</td>
<td>URI</td>
<td>A URI that uniquely identifies the operation at a global level.</td>
</tr>
<tr>
<td>description</td>
<td>string</td>
<td>The human-readable description of the semantic of the operation.</td>
</tr>
<tr>
<td>method</td>
<td>string</td>
<td>The protocol-dependent verb to use to perform the operation.</td>
</tr>
<tr>
<td>inputMessage</td>
<td>string</td>
<td>The body mimeType of the request message; it may depend on the model format</td>
</tr>
<tr>
<td>outputMessage</td>
<td>string</td>
<td>The body mimeType of the response message; it may depend on the model format</td>
</tr>
</tbody>
</table>

**Constraints:**

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

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

**JSON media type:** application/json

**JSON serialization:**

```json
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/ResourceMetadata",
  "id": string,
  "typeURI": string,
  "name": string,
  "attributes": [ { "name": string,
                      "namespace": string, ?
                      "type": string, ?
                      "required": boolean, ?
                      ...value constraints...? } *
```
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol  DSP0263

```
], ?,
"capabilities": [
  { "name": string, ?,
    "uri": string,
    "description": string, ?
    "value": any } *
], ?,
"actions" : [
  { "name": string,
    "uri": string,
    "description": string, ?
    "method": string,
    "inputMessage": string, ?
    "outputMessage": string ? }, *
], ?,
"operations": [
  { "rel": "edit", "href": string }, ?
  { "rel": "delete", "href": string } ?
] ?,
...
}

XML media type: application/xml

XML serialization:

```
<ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name>
  <typeURI> xs:anyURI </typeURI>
  <attribute name="xs:string" namespace="xs:anyURI"? type="xs:string"
    required="xs:boolean"? >
    ...value constraints...?
  </attribute> *
  <capability name="xs:string"? uri="xs:anyURI" description="xs:string"?
    xs:any*
  </capability> *
  <action name="xs:string" uri="xs:anyURI" description="xs:string"?
    method="xs:string" inputMessage="xs:string"?
    outputMessage="xs:string"? /> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
```

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

### 5.11.1 Serialization of attribute value constraints

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

**type="string"

The JSON shall be of the form:
```
"values": [ string, + ] ?
```

The XML shall be of the form:
```
<value> xs:string </value> *
```

**type="integer"

The JSON shall be of the form:
```
"values": [ number, + ], ?
"ranges": [ { "low": number, "high": number }, + ] ?
```

The XML shall be of the form:
```
<value> xs:integer </value> *
<range low="xs:integer" high="xs:integer"/> *
```

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

**type="boolean"

The JSON shall be of the form:
```
"value": boolean ?
```

The XML shall be of the form:
```
<value> xs:boolean </value> ?
```

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

### 5.11.1.1 Examples

The following example shows a sample metadata document for a VolumeConfiguration Resource in XML that lists the allowable values for the "format" attribute and has been extended with a "Location" string attribute:
```
<ResourceMetadata xmlns="http://schemas.dmtf.org/cimi/1">
  <id> http://example.org/types/VC </id>
  <typeURI> http://schemas.dmtf.org/cimi/1/VolumeConfiguration </typeURI>
  <name> VolumeConfiguration </name>
  <attribute name="format" type="string" required="false">
    <value> ext4 </value>
    <value> ntfs </value>
  </attribute>
</ResourceMetadata>
```
The following example shows the same VolumeConfiguration, but the "Location" attribute is restricted to a set of values and is required:

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

The following example shows the same VolumeConfiguration serialized in JSON:

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
  "id": "http://example.org/types/VC",
  "typeURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
  "name": "VolumeConfiguration",
  "attributes": [
    {
      "name": "format",
      "type": "string",
      "required": false,
      "values": [ "ext4", "ntfs" ]
    },
    {
      "name": "Location",
      "namespace": "http://example.org/",
      "type": "string",
      "required": true,
      "values": [ "NYC", "LAX" ]
    }
  ]
}
```
The following example shows a Volume serialized in JSON that provides an action of data compression.

In this specific example, the method returned (POST) is for the CIMI HTTP protocol; should another protocol be implemented (e.g., SOAP), the "method" is different:

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

### 5.11.2 Capabilities

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

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

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

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

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

### Table 7 – Capability URIs

<table>
<thead>
<tr>
<th>Resource Name</th>
<th>Capability Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CloudEntryPoint</td>
<td>ExpandParameter</td>
<td>If true, the Provider shall support the $expand query parameter.</td>
</tr>
<tr>
<td>CloudEntryPoint</td>
<td>FilterParameter</td>
<td>If true, the Provider shall support the $filter query parameter.</td>
</tr>
<tr>
<td>CloudEntryPoint</td>
<td>FirstParameter</td>
<td>If true, the Provider shall support both the $first and $last query parameters.</td>
</tr>
<tr>
<td>CloudEntryPoint</td>
<td>SelectParameter</td>
<td>If true, the Provider shall support the $select query parameter.</td>
</tr>
<tr>
<td>Resource Name</td>
<td>Capability Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CloudEntryPoint</td>
<td>FormatParameter</td>
<td>If true, the Provider shall support the $format query parameter.</td>
</tr>
<tr>
<td>CloudEntryPoint</td>
<td>OrderByParameter</td>
<td>If true, the Provider shall support the $orderby query parameter.</td>
</tr>
<tr>
<td>CloudEntryPoint</td>
<td>QueryPathNotation</td>
<td>If true, the Provider shall support the use of path-like notation with query parameter $select (see 4.1.6.3) to disambiguate between attributes of a Collection Resource and attributes of each items in the Collection if subsetting.</td>
</tr>
<tr>
<td>CloudEntryPoint</td>
<td>MaxPropertyItems</td>
<td>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.</td>
</tr>
<tr>
<td>System</td>
<td>SystemComponentTemplateByValue</td>
<td>If true, the Provider shall support the specification of ComponentTemplates by value in SystemTemplates.</td>
</tr>
<tr>
<td>Machine</td>
<td>DefaultInitialState</td>
<td>If this capability is set, unless otherwise provided (e.g., by a MachineTemplate &quot;initialState&quot; attribute), the Provider shall set a new Machine to this state value, assuming the value is compatible with the InitialStates capability, if set.</td>
</tr>
<tr>
<td>Machine</td>
<td>InitialStates</td>
<td>If this capability is set, and if using a MachineTemplate that has an &quot;initialState&quot; attribute, a Consumer shall use an initialState value from the set of values of this capability.</td>
</tr>
<tr>
<td>Machine</td>
<td>MachineConfigByValue</td>
<td>If true, the Provider shall support specifying MachineConfigurations by value. If true, the MachineTemplateByValue capability shall also have the value true.</td>
</tr>
<tr>
<td>Machine</td>
<td>MachineCredentialByValue</td>
<td>If true, the Provider shall support specifying Credentials by value in Machine create operations. If true, the MachineTemplateByValue capability shall also have the value true.</td>
</tr>
<tr>
<td>Machine</td>
<td>MachineImageByValue</td>
<td>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.</td>
</tr>
<tr>
<td>Machine</td>
<td>MachineVolumeTemplatesByValue</td>
<td>If true, the Provider shall support specifying VolumeTemplates by value in Machine create operations. If true, the MachineTemplateByValue capability shall also have the value true.</td>
</tr>
<tr>
<td>Machine</td>
<td>MachineTemplateByValue</td>
<td>If true, the Provider shall support specifying MachineTemplates by value in Machine create operations.</td>
</tr>
<tr>
<td>Machine</td>
<td>MachineStopForce</td>
<td>If true, the Provider shall support the “force” option on the stop and restart operations on Machines.</td>
</tr>
<tr>
<td>Machine</td>
<td>MachineStopForceDefault</td>
<td>If true, the Provider shall forcefully stop Machines if no other indication is provided. Otherwise, the Provider shall gracefully stop Machines.</td>
</tr>
<tr>
<td>Machine</td>
<td>RestoreFromImage</td>
<td>If true, the Provider supports restoring Machines from MachineImages that are not SNAPSHOT MachineImages.</td>
</tr>
<tr>
<td>Machine</td>
<td>UserData</td>
<td>If set, indicates which userData injection method shall be used by the Provider.</td>
</tr>
<tr>
<td>Machine</td>
<td>MachineAvailabilityLevel</td>
<td>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.</td>
</tr>
<tr>
<td>Credential</td>
<td>CredentialTemplateByValue</td>
<td>If true, the Provider shall support specifying CredentialTemplates by value in Credential create operations.</td>
</tr>
<tr>
<td>Volume</td>
<td>SharedVolumeSupport</td>
<td>If true, the Provider shall support that a single Volume</td>
</tr>
<tr>
<td>Resource Name</td>
<td>Capability Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Volume</td>
<td>VolumeConfigByValue</td>
<td>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.</td>
</tr>
<tr>
<td>Volume</td>
<td>VolumelImageByValue</td>
<td>If true, the Provider shall support specifying VolumelImages by value in the Volume create operation. If true, the VolumeTemplateByValue capability shall have the value true.</td>
</tr>
<tr>
<td>Volume</td>
<td>VolumeSnapshot</td>
<td>If true, the Provider shall support creating a new VolumelImage by referencing an existing Volume.</td>
</tr>
<tr>
<td>Volume</td>
<td>VolumeTemplateByValue</td>
<td>If true, the Provider shall support specifying the VolumeTemplates by value in Volume create operations.</td>
</tr>
<tr>
<td>Volume</td>
<td>VolumeAvailabilityLevel</td>
<td>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.</td>
</tr>
<tr>
<td>Network</td>
<td>NetworkConfigByValue</td>
<td>If true, the Provider shall support specifying NetworkConfigurations by value in Network create operations.</td>
</tr>
<tr>
<td>Network</td>
<td>NetworkTemplateByValue</td>
<td>If true, the Provider shall support specifying NetworkTemplates by value in Network create operations.</td>
</tr>
<tr>
<td>Network</td>
<td>DefaultInitialState</td>
<td>If this capability is set, unless otherwise provided (e.g., by a NetworkTemplate &quot;initialState&quot; attribute), the Provider shall set a new Network to this state value, assuming the value is compatible with the InitialStates capability, if set.</td>
</tr>
<tr>
<td>Network</td>
<td>InitialStates</td>
<td>If this capability is set, and if using a NetworkTemplate that has an &quot;initialState&quot; attribute, a Consumer shall use an initialState value from the set of values of this capability.</td>
</tr>
<tr>
<td>NetworkPort</td>
<td>NetworkPortConfigByValue</td>
<td>If true, the Provider shall support specifying NetworkPortConfigurations by value in NetworkPort create operations.</td>
</tr>
<tr>
<td>NetworkPort</td>
<td>NetworkPortTemplateByValue</td>
<td>If true, the Provider shall support specifying NetworkPortTemplates by value in NetworkPort create operations.</td>
</tr>
<tr>
<td>NetworkPort</td>
<td>DefaultInitialState</td>
<td>If this capability is set, unless otherwise provided (e.g., by a NetworkPortTemplate &quot;initialState&quot; attribute), the Provider shall set a new NetworkPort to this state value, assuming the value is compatible with the InitialStates capability, if set.</td>
</tr>
<tr>
<td>NetworkPort</td>
<td>InitialStates</td>
<td>If this capability is set, and if using a NetworkPortTemplate that has an &quot;initialState&quot; attribute, a Consumer shall use an initialState value from the set of values of this capability.</td>
</tr>
<tr>
<td>ForwardingGroup</td>
<td>MixedNetwork</td>
<td>If true, a Provider shall support ForwardingGroups that can have both private and public connections at the same time. Otherwise, ForwardingGroups shall have only private or public connections at the same time.</td>
</tr>
<tr>
<td>Job</td>
<td>JobRetention</td>
<td>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.</td>
</tr>
<tr>
<td>Meter</td>
<td>MeterConfigByValue</td>
<td>If true, the Provider shall support specifying MeterConfigurations by value in Meter create operations.</td>
</tr>
<tr>
<td>Meter</td>
<td>MeterTemplateByValue</td>
<td>If true, the Provider shall support specifying MeterTemplates by value in Meter create operations.</td>
</tr>
<tr>
<td>EventLog</td>
<td>Linked</td>
<td>If true, the Provider shall delete EventLogs that are associated with Resources if the Resource is deleted.</td>
</tr>
</tbody>
</table>
The following examples show the ResourceMetadata for a Machine that advertises some of its capabilities:

**JSON serialization:**

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

**XML serialization:**

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

### 5.11.3 ResourceMetadataCollection Resource

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

**JSON serialization:**

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

**XML serialization:**

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

### 5.12 Cloud Entry Point

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

Figure 1 illustrates the CloudEntryPoint and its relationship to other Resources. Although this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor normative.
If a Consumer issues a read on the **CloudEntryPoint** Resource, the Provider shall return a **CloudEntryPoint** Resource that only catalogs Resources on which this Consumer is allowed to perform operations. Table 8 describes the attributes for the **CloudEntryPoint** Resource.

### Table 8 – **CloudEntryPoint** attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type URI</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| **baseURI**         | http://www.dmf.org/cimi/CloudEntryPoint                                  | **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 CloudEntryPoint. The Collection contains a description of the Resources supported by the Provider. If a Resource does not have any metadata, it shall not appear in this list, e.g., it has no constraints beyond what the CIMI specification defines nor does it have any extension attributes. **Constraints:**  
  Provider: support optional; mutable  
  Consumer: support optional; read-only |
<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>CloudEntryPoint</td>
<td>Name</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>Type URI</td>
<td>cloud grouping</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>Attribute</td>
<td>URI</td>
<td>Description</td>
<td>Description</td>
</tr>
<tr>
<td>systems</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>systemTemplates</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>machines</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>machineTemplates</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>machineConfigs</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>machineImages</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>credentials</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>credentialTemplates</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>volumes</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>volumeTemplates</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>volumeConfigs</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>volumImages</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>systems</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>systemTemplates</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>machines</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>machineTemplates</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>machineConfigs</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>machineImages</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>credentials</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>credentialTemplates</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>volumes</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>volumeTemplates</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>volumeConfigs</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>volumImages</td>
<td>collection</td>
<td>URI</td>
<td>Description</td>
</tr>
</tbody>
</table>

**Attribute Details**

- **systems**: collection of System
  - **Description**: A reference to the SystemCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only

- **systemTemplates**: collection of SystemTemplate
  - **Description**: A reference to the SystemTemplateCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only

- **machines**: collection of Machine
  - **Description**: A reference to the MachineCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only

- **machineTemplates**: collection of MachineTemplate
  - **Description**: A reference to the MachineTemplateCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only

- **machineConfigs**: collection of MachineConfiguration
  - **Description**: A reference to the MachineConfigurationCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only

- **machineImages**: collection of MachineImage
  - **Description**: A reference to the MachineImageCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only

- **credentials**: collection of Credential
  - **Description**: A reference to the CredentialCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only

- **credentialTemplates**: collection of CredentialTemplate
  - **Description**: A reference to the CredentialTemplateCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only

- **volumes**: collection of Volume
  - **Description**: A reference to the VolumeCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only

- **volumeTemplates**: collection of VolumeTemplate
  - **Description**: A reference to the VolumeTemplateCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only

- **volumeConfigs**: collection of VolumeConfiguration
  - **Description**: A reference to the VolumeConfigurationCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only

- **volumImages**: collection of VolumeImage
  - **Description**: A reference to the VolumeImageCollection of this Cloud Entry Point.
  - **Constraints**:
    - Provider: support optional; mutable
    - Consumer: support optional; read-only
<table>
<thead>
<tr>
<th>Name</th>
<th>Type URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>CloudEntryPoint</td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://www.dmf.org/cimi/CloudEntryPoint">http://www.dmf.org/cimi/CloudEntryPoint</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>networks</td>
<td>collection</td>
<td>A reference to the NetworkCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>networkTemplates</td>
<td>collection</td>
<td>A reference to the NetworkTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>networkConfigs</td>
<td>collection</td>
<td>A reference to the NetworkConfigurationCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>networkPorts</td>
<td>collection</td>
<td>A reference to the NetworkPortCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>networkPortTemplates</td>
<td>collection</td>
<td>A reference to the NetworkPortTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>networkPortConfigs</td>
<td>collection</td>
<td>A reference to the NetworkPortConfigurationCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>addresses</td>
<td>collection</td>
<td>A reference to the AddressCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>addressTemplates</td>
<td>collection</td>
<td>A reference to the AddressTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>forwardingGroups</td>
<td>collection</td>
<td>A reference to the ForwardingGroupCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>forwardingGroupTemplates</td>
<td>collection</td>
<td>A reference to the ForwardingGroupTemplateCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>jobs</td>
<td>collection</td>
<td>A reference to the JobsCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>meters</td>
<td>collection</td>
<td>A reference to the MeterCollection of this Cloud Entry Point. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
</tbody>
</table>
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

<table>
<thead>
<tr>
<th>Name</th>
<th>CloudEntryPoint</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://www.dmf.org/cimi/CloudEntryPoint">http://www.dmf.org/cimi/CloudEntryPoint</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meterTemplates</td>
<td>collection</td>
<td>A reference to the MeterTemplateCollection of this Cloud Entry Point.</td>
</tr>
<tr>
<td></td>
<td>[Meter Template]</td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support optional; read-only</td>
</tr>
<tr>
<td>eventLogs</td>
<td>collection</td>
<td>A reference to the EventLogCollection of this Cloud Entry Point.</td>
</tr>
<tr>
<td></td>
<td>[EventLog]</td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support optional; read-only</td>
</tr>
<tr>
<td>eventLogTemplates</td>
<td>collection</td>
<td>A reference to the EventLogTemplateCollection of this Cloud Entry Point.</td>
</tr>
<tr>
<td></td>
<td>[EventLog Template]</td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support optional; read-only</td>
</tr>
</tbody>
</table>

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

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

**JSON media type**: application/json

**JSON serialization**:

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/CloudEntryPoint",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?,
  "baseURI": string,
  "resourceMetadata": { "href": string }, ?,
  "systems": { "href": string }, ?,
  "systemTemplates": { "href": string }, ?,
  "machines": { "href": string }, ?
  "machineTemplates": { "href": string }, ?
  "machineConfigs": { "href": string }, ?
  "machineImages": { "href": string }, ?
```
"credentials": { "href": string }, ?
"credentialTemplates": { "href": string }, ?
"volumes": { "href": string }, ?
"volumeTemplates": { "href": string }, ?
"volumeConfigs": { "href": string }, ?
"volumeImages": { "href": string }, ?
"networks": { "href": string }, ?
"networkTemplates": { "href": string }, ?
"networkConfigs": { "href": string }, ?
"networkPorts": { "href": string }, ?
"networkPortTemplates": { "href": string }, ?
"networkPortConfigs": { "href": string }, ?
"addresses": { "href": string }, ?
"addressTemplates": { "href": string }, ?
"forwardingGroups": { "href": string }, ?
"forwardingGroupTemplates": { "href": string }, ?
"jobs": { "href": string }, ?
"meters": { "href": string }, ?
"meterTemplates": { "href": string }, ?
"meterConfigs": { "href": string }, ?
"eventLogs": { "href": string }, ?
"eventLogTemplates": { "href": string }, ?
"operations": [
    { "rel": "edit", "href": string } ?
] ?
...
5.12.1 Operations

This Resource supports the Read and Update operations.

5.13 System Resources and relationships

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

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

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

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

- By default, all Resources that are created as the result of a System creation are also owned by the System. (This rule can be overridden by subsequent modifications to the top-level System Collection attributes.)

- Ownership of a Resource by a System is expressed by including the reference to the Resource in the appropriate top-level System Collection attribute, or by ownership to a sub-System of this System (i.e., ownership is transitive across hierarchies of Systems).

- If a Resource other than a System is added to an existing System (i.e., becomes owned by the System by insertion of its reference to the appropriate top-level System Collection attribute), other Resources already referred by this added Resource are by default not owned by the System. (This rule can be overridden by subsequent modifications to the top-level System Collection attributes.)
A Resource shall not be owned by more than one System at any point in time (unless there is an ownership relationship between these Systems). Note that a Resource does not need to owned by a System. By not including it in any of the Collections, the Resource is simply not part of any actions performed on the System. Table 9 describes the System attributes.

Table 9 – System attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>System</td>
<td></td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/System">http://schemas.dmtf.org/cimi/1/System</a></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>state</td>
<td>string</td>
<td>The operational state of the System. Allowable values include: (See 5.14.1.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CREATING: The System is in the process of being created.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STARTING/STARTED/STOPPING/STOPPED/PAUSING/PAUSED/SUSPENDING/SUSPENDED: The</td>
</tr>
<tr>
<td></td>
<td></td>
<td>System shall be in one of these states if all the Machines referenced by</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the System are in that state. See clause 5.14.1 for the list of available</td>
</tr>
<tr>
<td></td>
<td></td>
<td>actions based on the state of a Machine. Such transitional states may just</td>
</tr>
<tr>
<td></td>
<td></td>
<td>indicate that all Machines in a System are undergoing the same operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(e.g., &quot;start&quot;), without the System being actually operated on (e.g., no</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;start&quot; done at System level). An actual operation on a System may be traced</td>
</tr>
<tr>
<td></td>
<td></td>
<td>by querying the &quot;job&quot; entity.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MIXED: The System shall be in this state if either no Machines are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>referenced by this System or Machines referenced by this System are in</td>
</tr>
<tr>
<td></td>
<td></td>
<td>varying states. Such varying states are likely to occur when an operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>is in progress on a System, resulting in transitions of its Machine states</td>
</tr>
<tr>
<td></td>
<td></td>
<td>toward a new common state (e.g., STOPPED, STARTED) but at a different pace,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or sequentially one after the other.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DELETING: The System is in the process of being deleted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ERROR: The Provider has detected an error in the System.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The operations that result in transitions to the above defined states are</td>
</tr>
<tr>
<td></td>
<td></td>
<td>defined in clause 5.13.1.2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provider: support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td>systems</td>
<td>collection [System System]</td>
<td>A reference to the list of references to nested Systems owned by this</td>
</tr>
<tr>
<td></td>
<td></td>
<td>System. Adding an item (of type System) to this list is logically equivalent</td>
</tr>
<tr>
<td></td>
<td></td>
<td>to associating the referenced System to this System with a &quot;containment</td>
</tr>
<tr>
<td></td>
<td></td>
<td>relationship.&quot; Removing an item from this list is logically equivalent to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>de-associating the referenced System from this System. Note: The System</td>
</tr>
<tr>
<td></td>
<td></td>
<td>System Resource type represents an association between the System and another</td>
</tr>
<tr>
<td></td>
<td></td>
<td>System. It is defined in clause 5.13.1.1.1.</td>
</tr>
<tr>
<td>machines</td>
<td>collection [System Machine]</td>
<td>A reference to the list of references to Machines owned by this System.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adding an item (of type Machine) to this list is logically equivalent to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>associating the Machine to this System with a &quot;containment relationship.&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removing an item from this list is logically equivalent to de-associating</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the Machine from this System.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: The SystemMachine Resource type represents an association between the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>System and a Machine. It is defined in clause 5.13.1.1.2.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support optional; read-only</td>
</tr>
<tr>
<td>credentials</td>
<td>collection [System Credential]</td>
<td>A reference to the list of references to Credentials owned by this System.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adding an item (of type Credential) to this list is logically equivalent to</td>
</tr>
<tr>
<td></td>
<td></td>
<td>associating the Credential to this System with a &quot;containment relationship.&quot;</td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Name</td>
<td>System</td>
<td></td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/System">http://schemas.dmtf.org/cimi/1/System</a></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>volumes</td>
<td>collection</td>
<td>A reference to the list of references Volumes owned by this System. Adding an item (of type Volume) to this list is logically equivalent to associating the Volume to this System with a &quot;containment relationship.&quot; Removing an item from this list is logically equivalent to de-associating the Volume from this System. Note: The SystemVolume Resource type represents an association between the System and a Volume. It is defined in clause 5.13.1.1.4. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>networks</td>
<td>collection</td>
<td>A reference to the list of references Networks owned by this System. Adding an item (of type Network) to this list is logically equivalent to associating the Network to this System with a &quot;containment relationship.&quot; Removing an item from this list is logically equivalent to de-associating the Network from this System. Note: The SystemNetwork Resource type represents an association between the System and a Network. It is defined in clause 5.13.1.1.5. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>networkPorts</td>
<td>collection</td>
<td>A reference to the list of references NetworkPorts owned by this System. Adding an item (of type NetworkPort) to this list is logically equivalent to associating the NetworkPort to this System with a &quot;containment relationship.&quot; Removing an item from this list is logically equivalent to de-associating the NetworkPort from this System. Note: The SystemNetworkPort Resource type represents an association between the System and a NetworkPort. It is defined in clause 5.13.1.1.6. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>addresses</td>
<td>collection</td>
<td>A reference to the list of references Addresses owned by this System. Adding an item (of type Address) to this list is logically equivalent to associating the Address to this System with a &quot;containment relationship.&quot; Removing an item from this list is logically equivalent to de-associating the Address from this System. Note: The SystemAddress Resource type represents an association between the System and a Address. It is defined in clause 5.13.1.1.7. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>forwardingGroups</td>
<td>collection</td>
<td>A reference to the list of references ForwardingGroups owned by this System. Adding an item (of type ForwardingGroup) to this list is logically equivalent to associating the ForwardingGroup to this System with a &quot;containment relationship.&quot; Removing an item from this list is logically equivalent to de-associating the ForwardingGroup from this System. Note: The SystemForwardingGroup Resource type represents an association between the System and a ForwardingGroup. It is defined in clause 5.13.1.1.8. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
</tbody>
</table>
When implementing or using System, Providers and Consumers shall adhere to the syntax and semantics of its attributes as described in the above table as well as in the tables describing embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

**JSON media type:** application/json

**JSON serialization:**

```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
    "id": string,
    "name": string, ?
    "description": string, ?
    "created": string, ?
    "updated": string, ?
    "properties": { string: string, + }, ?
    "state": string,
    "systems": { "href": string }, ?
    "machines": { "href": string }, ?
    "credentials": { "href": string }, ?
    "volumes": { "href": string }, ?
    "networks": { "href": string }, ?
    "networkPorts": { "href": string }, ?
    "addresses": { "href": string }, ?
    "forwardingGroups": { "href": string }, ?
    "meters": { "href": string }, ?
    "eventLog": { "href": string }, ?
    "operations": [
        { "rel": "edit", "href": string }, ?
```
XML media type: application/xml

XML serialization:

```xml
<System xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name>
  <description> xs:string </description>
  <created> xs:dateTime </created>
  <updated> xs:dateTime </updated>
  <property key="xs:string"> xs:string </property>
  <state> xs:string </state>
  <systems href="xs:anyURI"/>
  <machines href="xs:anyURI"/>
  <credentials href="xs:anyURI"/>
  <volumes href="xs:anyURI"/>
  <networks href="xs:anyURI"/>
  <networkPorts href="xs:anyURI"/>
  <addresses href="xs:anyURI"/>
  <forwardingGroups href="xs:anyURI"/>
  <meters href="xs:anyURI"/>
  <eventLog href="xs:anyURI"/>
  <operation rel="edit" href="xs:anyURI"/>
  <operation rel="delete" href="xs:anyURI"/>
  <operation rel="http://schemas.dmtf.org/cimi/1/action/start" href="xs:anyURI"/>
  <operation rel="http://schemas.dmtf.org/cimi/1/action/stop" href="xs:anyURI"/>
  <operation rel="http://schemas.dmtf.org/cimi/1/action/restart" href="xs:anyURI"/>
</System>
```
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

5.13.1 Collections

The following clause describes the Collection Resources owned by Systems.

5.13.1.1 SystemSystem Collection

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

Table 10 – SystemSystem attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>SystemSystem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type/URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/SystemSystem">http://schemas.dmtf.org/cimi/1/SystemSystem</a></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>system</td>
<td>ref</td>
</tr>
<tr>
<td>Constraints: Provider: support mandatory; mutable</td>
<td></td>
</tr>
<tr>
<td>Consumer: support mandatory; read-only</td>
<td></td>
</tr>
</tbody>
</table>

JSON serialization:

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

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>SystemMachine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/SystemMachine">http://schemas.dmtf.org/cimi/1/SystemMachine</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>machine</td>
<td>ref</td>
<td>Reference to a Machine Resource.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provider: support mandatory; mutable</td>
</tr>
</tbody>
</table>

JSON serialization:
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemMachineCollection",
  "id": string,
  "count": number,
  "systemMachines": [ 
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemMachine", }]}
"id": string,
"name": string, ?
"description": string, ?
"created": string, ?
"updated": string, ?
"properties": { string: string, + }, ?
"machine": { "href": string },
"operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
] ?
...
], +
"operations": [ { "rel": "add", "href": string } ? ]
...
}

**XML serialization:**

```xml
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemMachineCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
    <id xs:anyURI /></id>
    <count> xs:integer </count>
    <SystemMachine>
        <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> *
        <machine href="xs:anyURI"/>
        <operation rel="edit" href="xs:anyURI"/> ?
        <operation rel="delete" href="xs:anyURI"/> ?
        <xs:any>*
    </SystemMachine> *
```

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

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

Table 12 – SystemCredential attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>SystemCredential</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/SystemCredential">http://schemas.dmtf.org/cimi/1/SystemCredential</a></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>credential</td>
<td>ref</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

JSON serialization:

```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCredentialCollection",
    "id": string,
    "count": number,
    "systemCredentials": [
        {
            "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCredential",
            "id": string,
            "name": string, ?
            "description": string, ?
            "created": string, ?
            "updated": string, ?
            "properties": { string: string, + }, ?
            "credential": { "href": string },
            "operations": [?
                { "rel": "edit", "href": string }, ?
                { "rel": "delete", "href": string } ?
            ] ?
            ...  
        }, +
    ], ?
    "operations": [ { "rel": "add", "href": string } ? ]
    ...
}
```

XML serialization:

```xml
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemCredentialCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id>xs:anyURI</id>
  <count>xs:integer</count>
  <SystemCredential>
```
DSP0263  Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

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

5.13.1.1.4 SystemVolume Collection

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

Table 13 – SystemVolume attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>SystemVolume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/SystemVolume">http://schemas.dmtf.org/cimi/1/SystemVolume</a></td>
</tr>
<tr>
<td>Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>volume</td>
<td>ref</td>
</tr>
</tbody>
</table>

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

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemVolumeCollection",
 "id": string,
 "count": number,
 "systemVolumes": [
   { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemVolume",
     "id": string,
     "name": string, ?
     "description": string, ?
     "created": string, ?
     "updated": string, ?
     "properties": [ string: string, + ], ?
     "volume": [ "href": string ],
     "operations": [ 
       { "rel": "edit", "href": string }, ?
       { "rel": "delete", "href": string } ?
   ]
 ]
}
```
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

XML serialization:

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

5.13.1.1.5 SystemNetwork Collection

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

Table 14 – SystemNetwork attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>ref</td>
<td><a href="http://schemas.dmtf.org/cimi/1/SystemNetwork">http://schemas.dmtf.org/cimi/1/SystemNetwork</a></td>
<td>Reference to a Network Resource. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</td>
</tr>
</tbody>
</table>

JSON serialization:

```json
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetworkCollection", } ?
```
"id": string,
"count": number,
"systemNetworks": [
  { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetwork",
    "id": string,
    "name": string, ?
    "description": string, ?
    "created": string, ?
    "updated": string, ?
    "properties": { string: string, + }, ?,
    "network": { "href": string },
    "operations": [
      { "rel": "edit", "href": string }, ?
      { "rel": "delete", "href": string } ?
    ] ?,
    ...
  }, +
], ?,
"operations": [ [ "rel": "add", "href": string ] ? ]
...]

XML serialization:

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>SystemNetworkPort</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td>http/schemas.dmtf.org/cimi/1/SystemNetworkPort</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>networkPort</td>
<td>ref</td>
<td>Reference to a NetworkPort Resource.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provider: support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support mandatory; read-only</td>
</tr>
</tbody>
</table>

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetworkPortCollection",
  "id": string,
  "count": number,
  "systemNetworkPorts": [
    {
      "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemNetworkPort",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "networkPort": { "href": string },
      "operations": [          
      { "rel": "edit", "href": string }, ?
      { "rel": "delete", "href": string } ?
      ] ?
    ...
    }, +
  },
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```

**XML serialization:**

```xml
<Collection
  resourceURI="http://schemas.dmtf.org/cimi/1/SystemNetworkPortCollection"
  xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
</Collection>
```
5.13.1.7 SystemAddress Collection

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

<table>
<thead>
<tr>
<th>Name</th>
<th>SystemAddress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/SystemAddress">http://schemas.dmtf.org/cimi/1/SystemAddress</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>ref</td>
<td>Reference to a Address Resource.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Provider:</strong> support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Consumer:</strong> support mandatory; read-only</td>
</tr>
</tbody>
</table>

JSON serialization:

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemAddressCollection",
  "id": string,
  "count": number,
  "systemAddresses": [
    {
      "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemAddress",
      "id": string,
      "name": string, ?
    }
  ]
}
```
XML serialization:

```xml
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemAddressCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <SystemAddress>
    <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> *
    <address href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </SystemAddress> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</Collection>
```

5.13.1.1.8 SystemForwardingGroup Collection

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

Table 17 – SystemForwardingGroup attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type URI</th>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>forwardingGroup</td>
<td><a href="http://schemas.dmtf.org/cimi/1/SystemForwardingGroup">http://schemas.dmtf.org/cimi/1/SystemForwardingGroup</a></td>
<td>ref</td>
<td>ref</td>
<td>Reference to a ForwardingGroup Resource. Constraints:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Provider: support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Consumer: support mandatory; read-only</td>
</tr>
</tbody>
</table>

76 DMTF Standard Version 1.1.0
JSON serialization:

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

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

XML serialization:

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

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

**JSON serialization:**

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

**XML serialization:**

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

5.13.1.2 Operations

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

The following custom operations are also defined:

`start/stop/restart/pause/suspend`
/link@rel: http://schemas.dmtf.org/cimi/1/action/xxx

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

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

export

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

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

Input parameters:

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

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

Output parameters: None.

HTTP protocol

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

JSON media type: application/json

JSON serialization:

```json
{
  "action": "http://schemas.dmtf.org/cimi/1/action/export",
  "format": string, ?
  "destination": string, ?
  "properties": { string: string, + } ?,
...
}
```

XML media type: application/xml

XML serialization

```xml
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/export </action>
  <format> xs:string </format>
</Action>
```
5.13.2 SystemCollection Resource

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

**JSON serialization:**

```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemCollection",
    "id": string,
    "count": number,
    "systems": [
        {
            "resourceURI": "http://schemas.dmtf.org/cimi/1/System",
            "id": string,
            ... remaining System attributes ...
        }, ...
    ],
    "operations": [
        { "rel": "add", "href": string }, ...
        { "rel": "http://schemas.dmtf.org/cimi/1/action/import", "href": string } ...
    ]
}
```

**XML serialization:**

```xml
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/SystemCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
    <id> xs:anyURI </id>
    <count> xs:integer </count>
    <System>
        <id> xs:anyURI </id>
        ... remaining System attributes ...
    </System>
    <operation rel="add" href="xs:anyURI"/> ...
    <operation rel="http://schemas.dmtf.org/cimi/1/action/import" href="xs:anyURI"/> ...
</Collection>
```

5.13.2.1 Operations

NOTE The "add" operation requires that a SystemTemplate be used (see 4.2.1.1).
Resources created during the process of creating a System shall be "owned" by the System (see 5.13.1). For example, a componentDescriptor that references a MachineTemplate, and within that MachineTemplate is a reference to a VolumeTemplate, results in a reference to the new Machine being added to the System.machines attribute and a reference to the new Volume being added to the System.volumes attribute. However, if this MachineTemplate refers to an existing Volume, this Volume shall not be added to the top-level System attributes.

The following custom operations are also defined:

**import**

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

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

1) Input parameters: "source" - type: URI - mandatory

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

Output parameters: None.

**HTTP protocol**

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

**JSON media type**: application/json

**JSON serialization**:

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

**XML media type**: application/xml

**XML serialization**

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

**5.13.3 SystemTemplate Resource**

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

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

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

Table 18 describes the SystemTemplate attributes.

### Table 18 – SystemTemplate attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attribute</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td><strong>Data</strong></td>
<td><strong>Type</strong></td>
</tr>
<tr>
<td><strong>Type</strong></td>
<td><strong>URI</strong></td>
</tr>
<tr>
<td><strong>type</strong></td>
<td><strong>URI</strong></td>
</tr>
<tr>
<td><strong>component Template</strong></td>
<td><strong>any</strong></td>
</tr>
<tr>
<td>Name</td>
<td>SystemTemplate</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------</td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/SystemTemplate">http://schemas.dmtf.org/cimi/1/SystemTemplate</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| MachineTemplate for a Machine. This attribute shall contain either:  
- A Template that is provided inline. Such an embedded Template may contain component references, each one of which shall resolve to the URI of a component with same name once created from this SystemTemplate.  
- A reference to an externally defined Template. Some attribute name/value pairs may be added inside the componentTemplate element to override similar attributes in the referred Template (as described in 4.2.1.1). This example shows how component references can be added to an external Template.  
Example (JSON):  
```
"machineTemplate": {
  "href": "http://example.com/machineTemplates/72000",
  "credential": { "href": "#MyCredential" }  
}
```
This “credential” attribute assumes that there is another componentDescriptor item named “MyCredential” of type “Credential” in the SystemTemplate. It shall set or override similar attribute in the referred MachineTemplate if instantiating the Machine component.  

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

| quantity    | integer | Number of component instances to be created from this component descriptor. By default, this number is equal to 1. If the value is 2 or more, the actual name assigned to each instance is the "name" value concatenated with a sequential number (e.g., if name="mymachine", and quantity=3, the names are: mymachine1, mymachine2, mymachine3.)  
Constraints:  
Provider: support optional; mutable  
Consumer: support optional; read-write |

| Meter Templates | MeterTemplates[] | A list of references to MeterTemplates that shall be used to create and connect a set of new Meters to the new System.  
Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate Resource.  
Constraints:  
Provider: support optional; mutable  
Consumer: support optional; read-write |

| eventLog Template | ref | A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new System.  
Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate Resource.  
Constraints:  
Provider: support optional; mutable  
Consumer: support optional; read-write |

<p>| Import Image | ref | If the Template is the result of an import - e.g., of an OVF package - this attribute should be used. If present, it shall reference the import source (e.g., OVF package) used to |</p>
<table>
<thead>
<tr>
<th>Name</th>
<th>SystemTemplate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/SystemTemplate">http://schemas.dmtf.org/cimi/1/SystemTemplate</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>create this Template.</td>
</tr>
<tr>
<td>Constraints:</td>
<td></td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td>Consumer:</td>
<td>support optional; read-only</td>
<td></td>
</tr>
</tbody>
</table>

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

**JSON media type**: application/json

**JSON serialization**:

```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
    "id": string,
    "name": string, ?
    "description": string, ?
    "created": string, ?
    "updated": string, ?
    "properties": { string: string, + }, ?
    "componentDescriptors": [
        { "name": string, ?
            "description": string, ?
            "properties": { string: string, + }, ?
            "type": string,
            "componentTemplate": {
                "href": string, ?
                ... ComponentTemplate attributes ... ?
            }
        },
        "quantity": number ?
    ], +
    "meterTemplates": [?
        { "href": string, ?
            ... MeterTemplate attributes ... ?
        }, *
    ], ?
    "eventLogTemplate": {
        "href": string, ?
        ... EventLogTemplate attributes ... ?
    }, ?
```


"importImage": { "href": string }, ?

"operations": [
  { "rel": "edit", "href": string }, ?
  { "rel": "delete", "href": string }, ?
  { "rel": "http://schemas.dmtf.org/cimi/1/action/export", "href": string } ?
] ?
...

XML media type: application/xml

XML serialization:

```xml
<SystemTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <componentDescriptor>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <property key="xs:string"> xs:string </property> *
    <type> xs:anyURI </type>
    <componentTemplate href="xs:anyURI"/>
    ... ComponentTemplate attributes ... ?
  </componentTemplate> *

  <quantity> xs:integer </quantity>
  </componentDescriptor> *
  <meterTemplate href="xs:anyURI"/>
  ... MeterTemplate attributes ... ?
  </meterTemplate> *
  <eventLogTemplate href="xs:anyURI"/>
  ... EventLogTemplate attributes ... ?
  </eventLogTemplate> ?
  <importImage href="xs:anyURI"/>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/export" href="xs:anyURI"/> ?
</SystemTemplate>
```
5.13.3.1 Operations

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

The following custom operations are also defined:

export

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

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

Input parameters:

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

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

Output parameters: None.

HTTP protocol

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

JSON media type: application/json

JSON serialization:

```json
{
   "action": "http://schemas.dmtf.org/cimi/1/action/export",
   "format": string, ?,
   "destination": string, ?
   "properties": { string: string, + } ?,
   ...
}
```

XML media type: application/xml

XML serialization

```xml
<Action xmlns="http://schemas.dmtf.org/cimi/1">
   <action> http://schemas.dmtf.org/cimi/1/action/export </action>
   <format> xs:string </format> ?
</Action>
```
5.13.4 SystemTemplateCollection Resource

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

**JSON serialization:**

```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplateCollection",
    "id": string,
    "count": number,
    "systemTemplates": [
        { "resourceURI": "http://schemas.dmtf.org/cimi/1/SystemTemplate",
          "id": string,
          ... remaining SystemTemplate attributes ...
        }, +
    ],
    "operations": [
        { "rel": "add", "href": string }, ?
        { "rel": "http://schemas.dmtf.org/cimi/1/action/import", "href": string } ?
    ]
}
```

**XML serialization:**

```xml
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/SystemTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
    <id>xs:anyURI</id>
    <count>xs:integer</count>
    <SystemTemplate>
        <id>xs:anyURI</id>
        ... remaining SystemTemplate attributes ...
    </SystemTemplate> *
    <operation rel="add" href="xs:anyURI"/> ?
    <operation rel="http://schemas.dmtf.org/cimi/1/action/import"
             href="xs:anyURI"/> ?
    <xs:any>*
</Collection>
```
5.13.4.1 Operations

The following custom operations are defined:

import

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

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

Input parameters:

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

Output parameters: None.

HTTP protocol

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

JSON media type: application/json

JSON serialization:

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

XML media type: application/xml

XML serialization

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

5.14 Machine Resources and relationships

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

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

### Table 19 – Machine attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type URI</th>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td><a href="http://schemas.dmtf.org/cimi/1/Machine">http://schemas.dmtf.org/cimi/1/Machine</a></td>
<td>state</td>
<td>string</td>
<td>The operational state of the Machine. Allowable values include:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CREATING: The Machine is in the process of being created.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>STARTING: The Machine is in the process of being started.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>CREATED: The Machine is available and ready for use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>STOPPING: The Machine is in the process of being stopped.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>STOPPED: This value is the virtual equivalent of powering off a physical Machine.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>There is no saved CPU or memory state. Clause 5.14.2.1 defines the initial state of a Machine.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>PAUSING: The Machine in the process of being PAUSED.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SUSPENDING: The Machine is in the process of being suspended.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>DELETING: The Machine is in the process of being deleted.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>ERROR: The Provider has detected an error in the Machine.</td>
</tr>
</tbody>
</table>

The operations that result in transitions to the above defined states are defined in clause 5.14.1.2.

**Constraints:**

- **Provider:** support mandatory; mutable
- **Consumer:** support mandatory; read-only

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu</td>
<td>integer</td>
<td>The amount of CPU that this Machine has.</td>
</tr>
</tbody>
</table>
### Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

#### DSP0263

<table>
<thead>
<tr>
<th>Name</th>
<th>Type URI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><a href="http://schemas.dmtf.org/cimi/1/Machine">http://schemas.dmtf.org/cimi/1/Machine</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory</td>
<td>integer</td>
<td>The size of the memory (RAM) in kibibytes allocated to this Machine. If this value is increased, it implies that the Machine is allocated more RAM, and vice versa if the value is decreased.</td>
</tr>
<tr>
<td>disks</td>
<td>collection</td>
<td>A reference to the list of disks (local storage) that are part of the Machine. Adding an element to this list creates a disk. Note: The Disk Resource type is defined in clause 5.14.1.1.1.</td>
</tr>
<tr>
<td>cpuArch</td>
<td>string</td>
<td>The CPU architecture that is supported by Machines created by using this configuration. Allowable values include: 68000, Alpha, ARM, Itanium, MIPS, PA_RISC, POWER, PowerPC, x86, x86_64, z/Architecture, SPARC. Providers may define additional values.</td>
</tr>
<tr>
<td>cpuSpeed</td>
<td>integer</td>
<td>The approximate CPU speed of this Machine - in megahertz.</td>
</tr>
<tr>
<td>volumes</td>
<td>collection</td>
<td>A reference to the list of references to Volumes that are connected to this Machine. Adding a Volume to this list means that the Machine has some access to the data on the Volume. Removing a Volume from this list means that the Machine no longer has access to the data on the Volume. Note: The MachineVolume Resource type represents an association between the Machine and a Volume. It is defined in clause 5.14.1.1.2.</td>
</tr>
<tr>
<td>networkInterfaces</td>
<td>collection</td>
<td>A reference to the list of MachineNetworkInterfaces on this Machine. Note: The MachineNetworkInterface Resource type represents an association between the Machine and a NetworkInterface. It is defined in clause 5.14.1.1.3.</td>
</tr>
<tr>
<td>latestSnapshot</td>
<td>ref</td>
<td>A reference to the SNAPSHOT representing the latest state captured for this Machine (either most recent Snapshot or the last Snapshot reverted to).</td>
</tr>
<tr>
<td>snapshots</td>
<td>collection</td>
<td>A reference to the list of references to the SNAPSHOT Machine Images taken of this Machine. Note: The MachineSnapshot Resource type represents an association between the Machine and a Snapshot. It is defined in clause 5.14.1.1.5.</td>
</tr>
</tbody>
</table>
When implementing or using `Machine`, Providers and Consumers shall adhere to the syntax and semantics of its attributes as described in the above table, as well as in the tables describing embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML:

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Machine",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "cpu": number,
  "memory": number,
  "disks": { "href": string }, ?,
  "cpuArch": string, ?
  "cpuSpeed": number, ?
  "volumes": { "href": string }, ?,
  "networkInterfaces": { "href": string }, ?,
  "latestSnapshot": { "href": string }, ?,
  "snapshots": { "href": string }, ?,
  "meters": { "href": string }, ?,
  "eventLog": { "href": string }, ?,
  "operations": [
    { "rel": "edit", "href": string }, ?,
    { "rel": "delete", "href": string }, ?,
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?,
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string }, ?,
    { "rel": "http://schemas.dmtf.org/cimi/1/action/restart", "href": string }, ?,
  ]
}
```
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

XML media type: application/xml

XML serialization:

```xml
<Machine xmlns="http://schemas.dmtf.org/cimi/1">
  <id>xs:anyURI</id>
  <name>xs:string</name>?
  <description>xs:string</description>?
  <created>xs:dateTime</created>?
  <updated>xs:dateTime</updated>?
  <property key="xs:string">xs:string</property> *
  <state>xs:string</state>
  <cpu>xs:integer</cpu>
  <memory>xs:integer</memory>
  <disks href="xs:anyURI"/>?
  <cpuArch>xs:string</cpuArch>?
  <cpuSpeed>xs:integer</cpuSpeed>?
  <volumes href="xs:anyURI"/>?
  <networkInterfaces href="xs:anyURI"/>?
  <latestSnapshot href="xs:anyURI"/>?
  <snapshots href="xs:anyURI"/>?
  <meters href="xs:anyURI"/>?
  <eventLog href="xs:anyURI"/>?
  <operation rel="edit" href="xs:anyURI"/>?
  <operation rel="delete" href="xs:anyURI"/>?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/start" href="xs:anyURI"/>?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/stop" href="xs:anyURI"/>?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/restart" href="xs:anyURI"/>?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/pause" href="xs:anyURI"/>?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/suspend" href="xs:anyURI"/>?
</Machine>
```
5.14.1.1 Collections

The following clause describes the Collection Resources owned by Machines.

5.14.1.1.1 Disk Collection

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

Table 20 – Disk attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>capacity</td>
<td>integer</td>
<td>The initial capacity, in kilobytes, of the disk.</td>
</tr>
<tr>
<td>initialLocation</td>
<td>string</td>
<td>Operating System-specific location (path) in its namespace where this disk first appears.</td>
</tr>
</tbody>
</table>

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

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

JSON serialization:

```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/DiskCollection",
    "id": string,
    "count": number,
    "disks": [
        {
            "resourceURI": "http://schemas.dmtf.org/cimi/1/Disk",
            "id": string,
            "name": string, ?
            "description": string, ?
            "created": string, ?
            "updated": string, ?
            "properties": { string: string, + }, ?
            "capacity": number,
            "initialLocation": string, ?
            "operations": [ ...
```
XML serialization:

```xml
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/DiskCollection"
  xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Disk>
    <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> *
    <capacity> xs:integer </capacity>
    <initialLocation> xs:string </initialLocation> ?
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
  </Disk> *
</Collection>
```

5.14.1.1.2 MachineVolumeCollection Resource

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

<table>
<thead>
<tr>
<th>Name</th>
<th>MachineVolume</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MachineVolume">http://schemas.dmtf.org/cimi/1/MachineVolume</a></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>initialLocation</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>MachineVolume</td>
</tr>
<tr>
<td>----------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MachineVolume">http://schemas.dmtf.org/cimi/1/MachineVolume</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>volume</td>
<td>ref</td>
<td>A reference to the Volume that is connected.</td>
</tr>
</tbody>
</table>

**Provider:** support optional; immutable  
**Consumer:** support optional; read-only

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

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

**XML serialization:**
```
<Collection
resourceURI="http://schemas.dmtf.org/cimi/1/MachineVolumeCollection"
xmlns="http://schemas.dmtf.org/cimi/1">
<id> xs:anyURI </id>
<count> xs:integer </count>
</MachineVolume>
```
5.14.1.3 MachineNetworkInterfaceCollection Resource

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

<table>
<thead>
<tr>
<th>Name</th>
<th>MachineNetworkInterface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MachineNetworkInterface">http://schemas.dmtf.org/cimi/1/MachineNetworkInterface</a></td>
</tr>
</tbody>
</table>

Table 22 – MachineNetworkInterface attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addresses</td>
<td>collection</td>
<td>A reference to the list of references to the Addresses for this network interface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: the MachineNetworkInterfaceAddress Resource type represents an association between the MachineNetworkInterface and an Address. It is defined in clause 5.14.1.4.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td>network</td>
<td>ref</td>
<td>A reference to a Network for this network interface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>networkPort</td>
<td>ref</td>
<td>A reference to the NetworkPort for this network interface. If this attribute is provided, the “network” attribute in the referenced NetworkPort shall have the same value as the “network” attribute in this network Interface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
<tr>
<td>state</td>
<td>string</td>
<td>The state of the MachineNetworkInterface. Allowable values include: ACTIVE: An active interface is the primary interface, able to forward traffic. PASSIVE: A passive interface is in a standby mode ready to forward traffic if the primary interface fails. DISABLED: A disabled interface is one that is not able to forward traffic.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>macAddress</td>
<td>string</td>
<td>Address assigned by the hypervisor when a machine is created or a unique address can be manually assigned. While this attribute can be specified, in most cases it is expected to be supplied by the Provider. Specifying this value is typically only done if the Template is only used for one particular Machine.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints:</td>
</tr>
</tbody>
</table>
### JSON serialization:

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

### XML serialization:

```xml
<Collection
  resourceURI="http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceCollection"
  xmlns="http://schemas.dmtf.org/cimi/1"/>
```
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

DSP0263

5.14.1.1.4 MachineNetworkInterfaceAddressCollection Resource

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

Table 23 – MachineNetworkInterfaceAddress attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>MachineNetworkInterfaceAddress</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddress">http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddress</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>address</td>
<td>ref</td>
<td>Reference to an Address Resource. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</td>
</tr>
</tbody>
</table>

JSON serialization:

```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddressCollection",
    "id": string,
    "count": number,
    "machineNetworkInterfaceAddresses": [ 
        
        "http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddress",
    ]
}
```
XML serialization:

```xml
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/MachineNetworkInterfaceAddressCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
    <id>xs:anyURI</id>
    <count>xs:integer</count>
    <MachineNetworkInterfaceAddress>
        <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>*
        <address href="xs:anyURI"/>
        <operation rel="edit" href="xs:anyURI"/>?
        <operation rel="delete" href="xs:anyURI"/>?
        <xs:any/>
    </MachineNetworkInterfaceAddress>*
    <operation rel="add" href="xs:anyURI"/>?
    <xs:any/>
</Collection>
```
5.14.1.1.5 MachineSnapshotCollection Resource

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

<table>
<thead>
<tr>
<th>Name</th>
<th>MachineSnapshot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MachineSnapshot">http://schemas.dmtf.org/cimi/1/MachineSnapshot</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>snapshot</td>
<td>ref</td>
<td>Reference to a SNAPSHOT MachineImage Resource.</td>
</tr>
<tr>
<td>Constraints:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Provider:</td>
<td>support mandatory; mutable</td>
<td></td>
</tr>
<tr>
<td>Consumer:</td>
<td>support mandatory; read-only</td>
<td></td>
</tr>
</tbody>
</table>

Table 24 – MachineSnapshot attributes

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

XML serialization:
```
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/MachineSnapshotCollection"
    xmlns="http://schemas.dmtf.org/cimi/1"
><id> xs:anyURI </id>
    <count> xs:integer </count>
    <MachineSnapshot>
        <id> xs:anyURI </id>
        <name> xs:string </name> ?
```
NOTE Previous versions of this specification included an "add" operation on this Resource. It is now deprecated in favor of creating a new MachineImage with the imageLocation attribute pointing to the Machine to be taken a snapshot from.

5.14.1.6 MachineMeterCollection Resource

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

JSON serialization:

```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineMeterCollection",
    "id": string,
    "count": number,
    "meters": [
        {
            "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
            "id": string,
            ...
        }, +
    ],
    "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

```xml
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/MachineMeterCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
    <id xmlns="http://schemas.dmtf.org/cimi/1">
    <count>
    <Meter>
        <id>
        ...
    </Meter> *
```
5.14.1.2 Operations

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

The following custom operations are also defined:

start

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

This operation shall start a Machine.

Input parameters: None.
Output parameters: None.

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

Upon successful completion of this operation, the Machine shall be in the “STARTED” state.

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

If the Machine was in the “SUSPENDED” or “PAUSED” state, starting it shall have the effect of resuming it.

HTTP protocol

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

JSON media type: application/json

JSON serialization:

```json
{
   "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
   "action": "http://schemas.dmtf.org/cimi/1/action/start",
   "properties": { string: string, + } ?
   ...
}
```

XML media type: application/xml

XML serialization

```xml
<Action xmlns="http://schemas.dmtf.org/cimi/1">
   <action> http://schemas.dmtf.org/cimi/1/action/start </action>
   <property key="xs:string"> xs:string </property> *
   <xs:any>*
</Action>
```
Upon successful processing of the request, the HTTP response body may be empty.

**stop**

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

This operation shall stop a Machine.

Input parameters:

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

Output parameters: None.

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

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

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

**HTTP protocol**

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

**JSON media type**: application/json

**JSON serialization**:

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/stop",
  "force": boolean, ?
  "properties": { string: string, + } ?
  ...
}
```

**XML media type**: application/xml

**XML serialization**

```xml
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
  <force> xs:boolean </force> ?
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```
Upon successful processing of the request, the HTTP response body may be empty.

restart

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

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

Input parameters:

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

Output parameters: None.

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

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

HTTP protocol

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

JSON media type: application/json

JSON serialization:

```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
    "action": "http://schemas.dmtf.org/cimi/1/action/restart",
    "force": boolean, ?
    "properties": { string: string, + } ?
    ...
}
```

XML media type: application/xml

XML serialization

```xml
<Action xmlns="http://schemas.dmtf.org/cimi/1">
    <action> http://schemas.dmtf.org/cimi/1/action/restart </action>
    <force> xs:boolean </force> ?
    <property key="xs:string"> xs:string </property> *
    <xs:any>*
</Action>
```

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

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

This operation shall pause a Machine.

Input parameters: None.

Output parameters: None.

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

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

HTTP protocol

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

JSON media type: application/json

JSON serialization:

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/pause",
  "properties": { string: string, + }
...}
```

XML media type: application/xml

XML serialization

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

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

suspend

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

This operation shall suspend a Machine.

Input parameters: None.

Output parameters: None.

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

**HTTP protocol**

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/suspend",
  "properties": { string: string, + } ?
... }
```

**XML media type:** application/xml

**XML serialization**

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

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

**capture**

http://schemas.dmtf.org/cimi/1/action/capture

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

**Snapshotting a Machine**

http://schemas.dmtf.org/cimi/1/action/snapshot

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

**Restoring a Machine**

http://schemas.dmtf.org/cimi/1/action/restore

This operation shall restore a Machine from a previously created MachineImage.
Input parameters:

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

Output parameters: None.

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

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

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

HTTP protocol

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

JSON media type: application/json

JSON serialization:

```json
{
   "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
   "action": "http://schemas.dmtf.org/cimi/1/action/restore",
   "image": string,
   "properties": { string: string, + } ?,
   ...
}
```

XML media type: application/xml

XML serialization

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

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

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

5.14.2 MachineCollection

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

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

XML serialization:

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

### 5.14.2.1 Operations

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

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

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

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

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

5.14.3 MachineTemplate

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Type URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>MachineTemplate</td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MachineTemplate">http://schemas.dmtf.org/cimi/1/MachineTemplate</a></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>volumes</td>
<td>volume[]</td>
<td>A list of structures, each containing a reference to an existing Volume and potentially describing aspects of the way that the given Volume is to be connected to the Machine during its creation from this MachineTemplate. Each volume structure has the following attributes:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>initialLocation</td>
<td>string</td>
<td>An Operating System-specific location (path) in its namespace where the Volume appears. Support of this attribute indicates that the Provider allows for Consumers to choose where the Volume appears. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
<tr>
<td>volume</td>
<td>ref</td>
<td>Reference to the Volume that is connected. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
</tbody>
</table>

Constraints:
Provider: support optional; mutable
Consumer: support optional; read-write
<table>
<thead>
<tr>
<th>Name</th>
<th>MachineTemplate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MachineTemplate">http://schemas.dmtf.org/cimi/1/MachineTemplate</a></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>volumeTemplates</td>
<td>volumeTemplate[]</td>
</tr>
<tr>
<td>Name</td>
<td>volumeTemplate</td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>initialLocation</td>
<td>string</td>
</tr>
<tr>
<td>volumeTemplate</td>
<td>ref</td>
</tr>
</tbody>
</table>
## MachineTemplate

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>meterTemplates</td>
<td>meterTemplates[]</td>
<td>A list of references to MeterTemplates that shall be used to create and connect a set of new Meters to the new Machine. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate Resource.</td>
</tr>
<tr>
<td>eventLogTemplate</td>
<td>ref</td>
<td>A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new Machine. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate Resource.</td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

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

"networkInterfaces": [
  { "addresses": [
    {"href": string}, +
  ],
    "network": {"href": string},
    "networkPort": {"href": string}, ?
    "state": string,
    "mtu": number ?
  }, +
], ?,

"userData": string, ?
"meterTemplates": [
  { "href": string, ?
    "href": string, ?
    ... MeterTemplate attributes ... ?
  }, *
], ?,

"eventLogTemplate": {
    "href": string, ?
    ... EventLogTemplate attributes ... ?
}, ?,

"operations": [
  { "rel": "edit", "href": string }, ?
  { "rel": "delete", "href": string } ?
] ?
...
Injection of user-defined data

To simplify the customization of individual Machines, it is possible to pass arbitrary data into the new Machine by using the userData parameter. The value of this parameter shall be the Base64-encoded payload. The Provider shall arrange for this data to be available from inside the Machine by using one of the following three methods:
1. **Metadata server**: The data can be retrieved from within the instance by using an HTTP GET request to http://169.254.169.254/cimi/latest/user-data.

2. **Disk**: The Machine has access to a Disk with an ISO 9660 file system on it. The data can be found in a file at <location>/cimi/user-data.

3. **Image modification**: The Provider modifies the root file system of the machine image just before launching the Machine. In UNIX-like operating systems, the data can be found in the file /var/lib/cimi/user-data.

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

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

### 5.14.3.1 Operations

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

### 5.14.4 MachineTemplateCollection Resource

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

#### JSON serialization:

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

#### XML serialization:

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

<MachineTemplate>
  <id xs:anyURI /></id>
  ... remaining MachineTemplate attributes ...
</MachineTemplate> *
<operation rel="add" href="xs:anyURI"/> ?
<xs:any>*
</Collection>

5.14.4.1 Operations

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

5.14.5 MachineConfiguration Resource

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

Table 26 describes the MachineConfiguration attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>MachineConfiguration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MachineConfiguration">http://schemas.dmtf.org/cimi/1/MachineConfiguration</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu</td>
<td>integer</td>
<td>The amount of CPU that a Machine realized from this configuration has. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
<tr>
<td>memory</td>
<td>integer</td>
<td>The amount of RAM, in kibibytes, that a Machine realized from this configuration has. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>disks</td>
<td>disk[]</td>
<td>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:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>disk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>capacity</td>
<td>integer</td>
</tr>
<tr>
<td>format</td>
<td>string</td>
</tr>
<tr>
<td>initialLocation</td>
<td>string</td>
</tr>
</tbody>
</table>
### Name
MachineConfiguration

### Type URI
http://schemas.dmtf.org/cimi/1/MachineConfiguration

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpuArch</td>
<td>string</td>
<td>Indicates the CPU architecture that is supported by Machines created by using this configuration. Allowable values include: 68000, Alpha, ARM, Itanium, MIPS, PA_RISC, POWER, PowerPC, x86, x86_64, z/Architecture, SPARC. Providers may define additional values.</td>
</tr>
<tr>
<td>cpuSpeed</td>
<td>number</td>
<td>The approximate CPU speed of this Machine in megahertz.</td>
</tr>
</tbody>
</table>

### Constraints:
- **Consumer**: support optional; read-write
- **Provider**: support optional; mutable
  - **Consumer**: support optional; read-write

---

**NOTE**

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
  "id": string,
  "name": string,
  "description": string,
  "created": string,
  "updated": string,
  "properties": { string: string, + },
  "cpu": number,
  "memory": number,
  "disks": [
    { "capacity": number,
      "format": string,
      "initialLocation": string?
    }, +
  ],
  "cpuArch": string,
  "cpuSpeed": number,
  "operations": [
    { "rel": "edit", "href": string },
    { "rel": "delete", "href": string }
  ]
}
```
DSP0263  Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

**XML media type:** application/xml

**XML serialization:**

```xml
<MachineConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <cpu> xs:integer </cpu>
  <memory> xs:integer </memory>
  <disk>
    <capacity> xs:integer </capacity>
    <format> xs:string </format>
    <initialLocation> xs:string </initialLocation> ?
  </disk> *
  <cpuArch> xs:string </cpuArch> ?
  <cpuSpeed> xs:integer </cpuSpeed> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineConfiguration>
```

### 5.14.5.1 Operations

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

### 5.14.6 MachineConfigurationCollection Resource

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

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfigurationCollection",
  "id": string,
  "count": number,
  "machineConfigurations": [ 
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineConfiguration",
      "id": string,
      ... remaining MachineConfiguration attributes ...
    }, +
} ```
5.14.6.1 Operations

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

5.14.7 MachineImage Resource

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

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

Table 27 describes the MachineImage attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>MachineImage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MachineImage">http://schemas.dmtf.org/cimi/1/MachineImage</a></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>state</td>
<td>string</td>
</tr>
<tr>
<td>Name</td>
<td>MachineImage</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MachineImage">http://schemas.dmtf.org/cimi/1/MachineImage</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>specified, the MachineImage shall initially be in this state after successful creation. <strong>DELETING:</strong> The MachineImage is in the process of being deleted. <strong>ERROR:</strong> 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  <strong>Constraints:</strong>  <strong>Provider:</strong> support mandatory; mutable  <strong>Consumer:</strong> support mandatory; read-only</td>
</tr>
<tr>
<td>Name</td>
<td>Type URI</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Name</td>
<td>MachineImage</td>
<td></td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MachineImage">http://schemas.dmtf.org/cimi/1/MachineImage</a></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type Description</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>string</td>
<td>The type of MachineImage that is represented by this Resource. This specification defines the following values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>IMAGE:</strong> This type represents the persisted data of a stopped Machine. Unlike &quot;snapshots&quot;, it does not contain any runtime information. If this value is used, the &quot;relatedImage&quot; attribute shall not be present.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>SNAPSHOT:</strong> This type represents the persisted data of a Machine. If the Machine was not in a stopped state if this Image was created, it also contains runtime information. If this value is used, the &quot;relatedImage&quot; attribute shall reference the most recently created (or reverted to) snapshot Image for that Machine, which allows for easy discovery of the &quot;previous&quot; snapshot. The &quot;relatedImage&quot; attribute shall not be set by Consumers.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>PARTIAL_SNAPSHOT:</strong> This type follows the same semantics as the &quot;SNAPSHOT&quot; MachineImage except that it contains just the changes (deltas) made to the Machine based on the referenced &quot;relatedImage&quot; MachineImage rather than a complete representation of the Machine.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>imageLocation</td>
<td>URI</td>
<td>A reference to the location of the binary data that makes up this image.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provider: support mandatory; immutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td>relatedImage</td>
<td>ref</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support optional; read-only</td>
</tr>
</tbody>
</table>
The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/MachineImage",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "type": string,
  "imageLocation": string,
  "relatedImage": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

**XML media type:** application/xml

**XML serialization:**

```xml
<MachineImage xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <type> xs:string </type>
  <imageLocation> xs:anyURI </imageLocation>
  <relatedImage href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MachineImage>
```
5.14.7.1 Operations

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

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

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

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

5.14.8 MachineImageCollection Resource

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

JSON serialization:

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

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/MachineImageCollection"
  xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <MachineImage>
    <id> xs:anyURI </id>
    ... remaining MachineImage attributes ...
  </MachineImage> *
```
5.14.8.1 Operations

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

5.14.9 Credential Resource

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

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

Table 28 describes the Credential attributes.

Table 28 – Credential attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Credential Type</td>
<td>URI</td>
</tr>
<tr>
<td><a href="http://schemas.dmtf.org/cimi/1/Credential">http://schemas.dmtf.org/cimi/1/Credential</a></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TBD</td>
<td></td>
<td>The exact set of attributes is determined by the Provider.</td>
</tr>
</tbody>
</table>

Some common extension attributes that Providers might use include:

Table 29 – UserName/Password attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>userName</td>
<td>string</td>
<td>Initial superuser's user name. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>password</td>
<td>string</td>
<td>Initial superuser's password. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; write-only</td>
</tr>
</tbody>
</table>

Table 30 – Public key attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>key</td>
<td>byte[]</td>
<td>The digit of the public key for the initial superuser. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Credential",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

**XML media type:** application/xml

**XML serialization:**

```xml
<Credential xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</Credential>
```

### 5.14.9.1 Operations

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

### 5.14.10 CredentialCollection Resource

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

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

XML serialization:

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

5.14.10.1 Operations

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

5.14.11 CredentialTemplate Resource

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/CredentialTemplate">http://schemas.dmtf.org/cimi/1/CredentialTemplate</a></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>TBD</td>
<td></td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/CredentialTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```

**XML media type:** application/xml

**XML serialization:**

```xml
<CredentialTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</CredentialTemplate>
```

**5.14.11.1 Operations**

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

**5.14.12 CredentialTemplateCollection Resource**

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

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

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

5.14.12.1 Operations

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

5.15 Volume Resources and relationships

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

![Volume Resources Diagram](image)

Figure 4 - Volume Resources
### 5.15.1 Volume

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

#### Table 32 – Volume attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Volume</th>
<th>Type URI</th>
<th>Description</th>
</tr>
</thead>
</table>
| state  | string | http://schemas.dmtf.org/cimi/1/Volume | The operational state of the Volume. Allowable values include:  
CREATING: The Volume is in the process of being created.  
AVAILABLE: The Volume is available and ready for use. Unless otherwise specified, the Volume shall be in this state initially after successful creation.  
CAPTURING: The Volume is in the process of being captured (snapshotted) into a new VolumeImage. Allowable action if in this state is: delete.  
DELETING: The Volume is in the process of being deleted.  
ERROR: The Provider has detected an error in the Volume. The operations that result in transitions to the above defined states are defined in clause 5.15.1.2  
Constraints:  
Provider: support mandatory; mutable  
Consumer: support mandatory; read-only |
| type   | URI    |                      | A URI that indicates the type of Volume to be created. This specification defines the following URI:  
http://schemas.dmtf.org/cimi/1/mapped: Indicates a Volume that shall be used for shared storage that might be available to multiple Machines, but which does not require an explicit mount operation from within the guest operating system.  
Additional values may be defined. If certain types of Volumes require additional data, it is expected that this Resource is extended. For example, a "sharedFileSystem" type might require additional networking information and credentials to be specified.  
Constraints:  
Provider: support mandatory; immutable  
Consumer: support mandatory; read-only |
| capacity | integer |                      | The maximum size, if limited, of the Volume in kilobytes.  
If this value is increased, the Volume can contain more data. Decreasing this value may require evaluations.  
Constraints:  
Provider: support mandatory; mutable  
Consumer: support mandatory; read-only |
| bootable | boolean |                      | This property indicates whether this Volume is bootable.  
Constraints:  
Provider: support mandatory; mutable  
Consumer: support mandatory; read-write |
| images | collection [Volume Image] |                      | A reference to the list of references to VolumeImages that represent snapshots taken from the Volume.  
Note: The VolumeVolumeImage Resource type represents an association between the Volume and a VolumeImage. It is defined in clause 5.15.1.1.1.  
Constraints:  
Provider: support optional; mutable  
Consumer: support optional; read-only |
| meters | collection [Meter] |                      | A reference to the list of Meters monitored for this Volume.  
Constraints:  
Provider: support optional; mutable  
Consumer: support optional; read-only |
| eventLog | ref |                      | A reference to the EventLog of this Volume.  
Constraints:  
Provider: support optional; mutable |
When implementing or using Volume, Providers and Consumers shall adhere to the syntax and semantics of its attributes as described in the above table as well as in the tables describing embedded Resources or related Collections. Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML:

**JSON media type:** application/json

**JSON serialization:**

```
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Volume",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "type": string,
  "capacity": number,
  "bootable": boolean,
  "images": { "href": string }, ?,
  "meters": { "href": string }, ?,
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?,
    { "rel": "delete", "href": string } ?
  ] ?
}
```

**XML media type:** application/xml

**XML serialization:**

```
<Volume xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
</Volume>
```
5.15.1.1 Collections

The following clauses describe the Collection Resources owned by Volumes.

5.15.1.1.1 VolumeVolumeImageCollection Resource

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

<table>
<thead>
<tr>
<th>Name</th>
<th>VolumeVolumeImage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/VolumeVolumeImage">http://schemas.dmtf.org/cimi/1/VolumeVolumeImage</a></td>
</tr>
</tbody>
</table>

Table 33 – VolumeVolumeImage attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>volumeImage</td>
<td>ref</td>
<td>Reference to a VolumeImage Resource, which represents a snapshot of this Volume.</td>
</tr>
</tbody>
</table>

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

JSON serialization:

```json
{
"resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeVolumeImageCollection",
"id": string,
"count": number,
"volumeVolumeImages": [
  {
    "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeVolumeImage",
    "id": string,
    "name": string, ?
    "description": string, ?
    "created": string, ?
    "updated": string, ?
    "properties": { string: string, + }, ?
    "volumeImage": { "href": string },
    "operations": [ 
      { "rel": "edit", "href": string }, ?
```
XML serialization:

```xml
<Collection
  resourceURI="http://schemas.dmtf.org/cimi/1/VolumeVolumeImageCollection"
  xmlns="http://schemas.dmtf.org/cimi/1">
  <id>xs:anyURI</id>
  <count>xs:integer</count>
  <VolumeVolumeImage>
    <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>*
    <volumeImage href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/>?
    <operation rel="delete" href="xs:anyURI"/>?
    <xs:any>*
  </VolumeVolumeImage>*
  <xs:any>*
</Collection>
```

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

5.15.1.1.2 VolumeMeterCollection Resource

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

JSON serialization:

```json
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeMeterCollection",
  "id": string,
  "count": number,
  "meters": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
      "id": string,
      ...
  ... remaining Meter attributes ...
```
XML serialization:

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

5.15.1.2 Operations

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

5.15.2 VolumeCollection Resource

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

JSON serialization:

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

XML serialization:

```
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/VolumeCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
    <id> xs:anyURI </id>
    <count> xs:integer </count>
    <Meter>
        <id> xs:anyURI </id>
        ... remaining Meter attributes ...
    </Meter>*
    <operation rel="add" href="xs:anyURI"/>
</Collection>
```
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

```xml
<Collection>
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Volume>
    <id> xs:anyURI </id>
    ... remaining Volume attributes ...
    </Volume> *
  <operation rel="add" href="xs:anyURI" /> ?
  <xs:any>*
</Collection>
```

5.15.2.1 Operations

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

5.15.3 VolumeTemplate Resource

This Resource captures the configuration values for realizing a Volume. A VolumeTemplate may be used to create multiple Volumes. Table 34 describes the VolumeTemplate attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>volumeConfig</td>
<td>ref</td>
<td><a href="http://schemas.dmtf.org/cimi/1/VolumeTemplate">http://schemas.dmtf.org/cimi/1/VolumeTemplate</a></td>
<td>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.</td>
</tr>
<tr>
<td>volumeImage</td>
<td>ref</td>
<td></td>
<td>A reference to the VolumeImage that is used to create a Volume from this VolumeTemplate.</td>
</tr>
<tr>
<td>meterTemplates</td>
<td>Meter Templates[]</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>eventLogTemplate</td>
<td>ref</td>
<td></td>
<td>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.</td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": [ string: string, + ], ?,
  "volumeConfig": {
    "href": string | ... VolumeConfiguration attributes ...
  }
  "volumeImage": [ "href": string ], ?,
  "meterTemplates": [
    { "href": string, ?
      ... MeterTemplate attributes ... ?
  }, ?
  ], ?
  "eventLogTemplate": [ ?
    "href": string, ?
    ... EventLogTemplate attributes ... ?
  ), ?,
  "operations": [ ?
    { "rel": "edit", "href": string }, ?,
    { "rel": "delete", "href": string } ?
  ] ?
  ...

```

**XML media type:** application/xml

**XML serialization:**

```xml
<VolumeTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
```

Version 1.1.0  DMTF Standard  137
5.15.3.1 Operations

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

5.15.4 VolumeTemplateCollection Resource

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

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

5.15.5 VolumeConfiguration Resource

The VolumeConfiguration Resource represents the set of configuration values needed to create a Volume with certain characteristics. VolumeConfigurations are created by Providers and may, at the Providers discretion, be created by Consumers.

Table 35 describes the VolumeConfiguration attributes.
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

**Table 35 – VolumeConfiguration attributes**

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>URI</td>
<td>A URI that indicates the type of Volume to be created. This specification defines the following URI: <a href="http://schemas.dmtf.org/cimi/1/mapped">http://schemas.dmtf.org/cimi/1/mapped</a>: 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. <strong>Constraints:</strong> Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>format</td>
<td>string</td>
<td>The format of the file system that is placed on Volumes created from this configuration. This attribute is optional; the absence of this attribute indicates that Volumes created from this configuration are not formatted with a file system. Example values: “ext4,” “ntfs.” <strong>Constraints:</strong> Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
<tr>
<td>capacity</td>
<td>integer</td>
<td>The default size in kilobytes, if limited, of the Volume created from this VolumeConfiguration. <strong>Constraints:</strong> Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "type": string,
  "format": string,
  "capacity": number,
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
  ...
}
```
XML media type: application/xml

XML serialization:

```xml
<VolumeConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <type> xs:anyURI </type>
  <format> xs:string </format>
  <capacity> xs:integer </capacity>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</VolumeConfiguration>
```

5.15.5.1 Operations

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

5.15.6 VolumeConfigurationCollection Resource

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

JSON serialization:

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfigurationCollection",
  "id": string,
  "count": number,
  "volumeConfigurations": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeConfiguration",
      "id": string,
      ... remaining VolumeConfiguration attributes ...
    }, +
  ],
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```
XML serialization:

```xml
<Collection
  resourceURI="http://schemas.dmtf.org/cimi/1/VolumeConfigurationCollection"
  xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <VolumeConfiguration>
    <id> xs:anyURI </id>
    ... remaining VolumeConfiguration attributes ...
  </VolumeConfiguration> *
  <operation rel="add" href="xs:anyURI"/> ?
  <xs:any>*
</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 pre-loaded volume. Table 36 describes the VolumeImage attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>VolumeImage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/VolumeImage">http://schemas.dmtf.org/cimi/1/VolumeImage</a></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>state</td>
<td>string</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>imageLocation</td>
<td>ref</td>
</tr>
<tr>
<td>bootable</td>
<td>boolean</td>
</tr>
</tbody>
</table>

The following pseudo-schemas describe the serialization of the Resource in both JSON and XML:
JSON media type: application/json

JSON serialization:
```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumeImage",
 "id": string,
 "name": string, ?
 "description": string, ?
 "created": string, ?
 "updated": string, ?
 "properties": { string: string, + }, ?
 "state": string,
 "imageLocation": { "href": string },
 "bootable": boolean,
 "operations": [
  { "rel": "edit", "href": string }, ?,
  { "rel": "delete", "href": string } ?
 ] ?
 ... } ?
```

XML media type: application/xml

XML serialization:
```
<VolumeImage xmlns="http://schemas.dmtf.org/cimi/1">
 <id> xs:anyURI </id>
 <name> xs:string </name> ?
 <description> xs:string </description> ?
 <created> xs:dateTime </created> ?
 <updated> xs:dateTime </updated> ?
 <property key="xs:string"> xs:string </property> *
 <state> xs:string </state>
 <imageLocation href="xs:anyURI"/>
 <bootable> xs:boolean </bootable>
 <operation rel="edit" href="xs:anyURI"/> ?
 <operation rel="delete" href="xs:anyURI"/> ?
 <xs:any>*
</VolumeImage>
```

5.15.7.1 Operations

This Resource supports the Read, Update, and Delete operations. Create is supported through the VolumeImageCollection Resource.
5.15.8 VolumelImageCollection Resource

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

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumelImageCollection",
  "id": string,
  "count": number,
  "volumeImages": [
    {
      "resourceURI": "http://schemas.dmtf.org/cimi/1/VolumelImage",
      "id": string,
      ... remaining VolumelImage attributes ...
    }, +
  ],
  "operations": [
    { "rel": "add", "href": string }
  ]
}
```

**XML serialization:**

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

5.15.8.1 Operations

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

During the creation of a new VolumelImage Resource, if the "imageLocation" attribute refers to an existing Volume, this operation shall be interpreted as a request to create a snapshot of the Volume. Once completed, the "imageLocation" attribute of the new VolumelImage Resource shall not refer to the original Volume; instead it shall refer to a static copy of the Volume. Additionally, the referenced Volume's VolumelVolumeImageCollection shall be updated to include a reference to this newly created snapshot VolumelImage Resource. During this process, the Provider may put the Volume into a "CAPTURING" state if necessary.
5.16 Network Resources and relationships

Figure 5 illustrates the Resources involved in constructing Networks and their NetworkPorts and their relationships. Although this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor normative.

Figure 5 - Network Resources

5.16.1 Network

A Network is a Collection of interconnected logical services with the purpose of forwarding data traffic between end points.

Networks in a ForwardingGroup should all have the same "networkType" attributes, which prevents a Network with a "private" access attribute from being publicly forwarded because it is a member of a ForwardingGroup that also contains Networks with a "public" access attribute.

Table 37 describes the Network attributes.

Table 37 – Network attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/Network">http://schemas.dmtf.org/cimi/1/Network</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>string</td>
<td>The operational state of the Network.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Allowable values include:</td>
</tr>
</tbody>
</table>
**Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol**

<table>
<thead>
<tr>
<th>Name</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/Network">http://schemas.dmtf.org/cimi/1/Network</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>networkType</td>
<td>string</td>
<td>An indicator of whether the Machine Resource has access to a Public or Private Network.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Allowable values include: PUBLIC: represents an open and Internet routable network. PRvivate: identifies a local non-routed network.</td>
</tr>
<tr>
<td>mtu</td>
<td>integer</td>
<td>(Maximum Transmission Unit) The largest Packet size supported on this Network.</td>
</tr>
<tr>
<td>classOfService</td>
<td>string</td>
<td>The Provider’s supported category associated with a Collection of attributes characterizing a level of a quality experience. Example values: GOLD: High bandwidth, low latency, low jitter SILVER: An improved service experience over bronze for voice or video traffic BRONZE: Best effort</td>
</tr>
<tr>
<td>networkPorts</td>
<td>collection</td>
<td>A reference to the list of NetworkPorts that are associated with this Network.</td>
</tr>
<tr>
<td>forwardingGroup</td>
<td>ref</td>
<td>A reference to a ForwardingGroup of which this Network is a part.</td>
</tr>
<tr>
<td>meters</td>
<td>collection</td>
<td>A reference to the list of Meters monitored for this Network.</td>
</tr>
<tr>
<td>eventLog</td>
<td>ref</td>
<td>A reference to the EventLog of this Network.</td>
</tr>
</tbody>
</table>

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.

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

<table>
<thead>
<tr>
<th>networkType</th>
<th>Provider: support mandatory; mutable</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td>mtu</td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td></td>
<td>Consumer: support optional; read-write</td>
</tr>
<tr>
<td>classOfService</td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td></td>
<td>Consumer: support optional; read-write</td>
</tr>
<tr>
<td>networkPorts</td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td></td>
<td>Consumer: support optional; read-write</td>
</tr>
<tr>
<td>forwardingGroup</td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td></td>
<td>Consumer: support optional; read-only</td>
</tr>
<tr>
<td>meters</td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td></td>
<td>Consumer: support optional; read-only</td>
</tr>
<tr>
<td>eventLog</td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td></td>
<td>Consumer: support optional; read-only</td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Network",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "networkType": string, ?
  "mtu": number, ?
  "classOfService": string, ?
  "networkPorts": { "href": string }, ?
  "forwardingGroup": { "href": string }, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
  ] ?
...}
```

**XML media type:** application/xml

**XML serialization:**

```xml
<Network xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <networkType> xs:string </networkType> ?
</Network>
```
5.16.1.1 Collections

The following clauses describe the Collection Resources owned by Networks.

5.16.1.1.1 NetworkNetworkPortCollection Resource

If NetworkPorts are created through a Network's NetworkPortCollection's "add" operation, they shall be added to the global (Cloud Entry Point) NetworkPortCollection as well.

As specified in clause 5.5.12, if a Network is deleted, all of its Collections, and Resources in those Collections, shall also be deleted. This means that all of the NetworkPorts related to that Network shall also be deleted.

The Resource type for each item of this Collection is "NetworkPort" as defined in clause 5.16.7.

JSON serialization:

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkNetworkPortCollection",
  "id": string,
  "count": number,
  "networkNetworkPorts": [
    {
      "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkNetworkPort",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "networkPort": { "href": string },
      "operations": [
        { "rel": "edit", "href": string }, ?
      ]
    }
  ]
}
```
XML serialization:

```
<Collection
  resourceURI="http://schemas.dmtf.org/cimi/1/NetworkNetworkPortCollection"
  xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <NetworkNetworkPort>
    <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>
    <networkPort href="xs:anyURI"/>
    <operation rel="edit" href="xs:anyURI"/>
    <operation rel="delete" href="xs:anyURI"/>
    <xs:any/>
  </NetworkNetworkPort> *
</Collection>
```

5.16.1.1.2 NetworkMeterCollection Resource

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

JSON serialization:

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

...  
}

XML serialization:

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

5.16.1.2 Operations

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

The following custom operations are also defined:

**start**

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

This operation shall start a Network.

Input parameters: None.

Output parameters: None.

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

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

HTTP protocol

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

**JSON media type:** application/json

**JSON serialization:**

```json
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/start",
  "properties": { string: string, + } ?
  ...
```
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

4635

4636
Upon successful processing of the request, the HTTP response body may be empty.

**stop**

http://schemas.dmtf.org/cimi/1/action/stop

This operation shall stop a Network. If stopped, a Network shall not allow data to flow through it.

Input parameters: None.

Output parameters: None.

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

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

**HTTP protocol**

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
   "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
   "action": "http://schemas.dmtf.org/cimi/1/action/stop",
   "properties": { string: string, + } ?
   ...
}
```

**XML media type:** application/xml

**XML serialization**

```xml
<Action xmlns="http://schemas.dmtf.org/cimi/1">
   <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
   <property key="xs:string"> xs:string </property> *
   <xs:any>*
</Action>
```

Upon successful processing of the request, the HTTP response body may be empty.
5.16.2 NetworkCollection Resource

A NetworkCollection Resource represents the Collection of Networks within a Provider and follows the Collection pattern that is defined in clause 5.5.12. This Resource shall be serialized as follows:

**JSON serialization:**
```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkCollection",
    "id": "string",
    "count": number,
    "networks": [
        { "resourceURI": "http://schemas.dmtf.org/cimi/1/Network",
            "id": "string",
            ... remaining Network attributes ...
        }, 
        
    ],

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

    ...

}
```

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

5.16.2.1 Operations

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

Upon successful processing of the "add" operation, unless otherwise specified by the way of the NetworkTemplate "initialState" attribute, the state of the new Network shall be the value of the DefaultInitialState capability of the Network Resource's ResourceMetadata, if defined. If no DefaultInitialState capability is defined, the default value shall be "STOPPED." The semantics of "initialState" shall be equivalent to the Provider issuing the appropriate actions against the new Network to move it into that state.

If a Provider is unable to change the state of the new Network to the appropriate "initialState" (either as specified by the NetworkTemplate or as implied by the previous stated rules), the Network creation shall fail.
5.16.3 NetworkTemplate Resource

The NetworkTemplate is a set of configuration values for realizing a Network. An instance of NetworkTemplate may be used to create multiple Networks. Table 38 describes the NetworkTemplate attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>URI</td>
<td>Description</td>
</tr>
<tr>
<td>initialState</td>
<td>string</td>
<td>The initial state of the new Network. Possible values include the non-transient states as specified by the Network &quot;state&quot; attribute (i.e., STARTED, STOPPED) and shall be determined by the actions supported by the Provider. Providers should advertise the list of available values by the way of the Network Resource Metadata &quot;initialStates&quot; capability. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
<tr>
<td>networkConfig</td>
<td>ref</td>
<td>A reference to the NetworkConfiguration that is used to create a Network from this NetworkTemplate. Note that the attributes of the NetworkConfiguration may be specified rather than a reference to an existing NetworkConfiguration Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
<tr>
<td>forwardingGroup</td>
<td>ref</td>
<td>A reference to a ForwardingGroup of which this Network is a part. Note that Networks forward to themselves; therefore, this attribute only appears in cases where the Network that is created from this Template forwards to one or more additional Networks. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
<tr>
<td>meterTemplates</td>
<td>meterTemplates[]</td>
<td>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</td>
</tr>
<tr>
<td>eventLogTemplate</td>
<td>ref</td>
<td>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</td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkTemplate",
  "initialState": "STARTED",
  "networkConfig": "http://schemas.dmtf.org/cimi/1/NetworkConfiguration",
  "forwardingGroup": "http://schemas.dmtf.org/cimi/1/ForwardingGroup",
  "meterTemplates": [
    "http://schemas.dmtf.org/cimi/1/MeterTemplate"
  ],
  "eventLogTemplate": "http://schemas.dmtf.org/cimi/1/EventLogTemplate"
}
```
DSP0263  Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

```
"id": string,
"name": string, ?
"description": string, ?
"created": string, ?
"updated": string, ?
"properties": { string: string, + }, ?
"initialState": string, ?
"networkConfig": { 
  "href": string |... NetworkingConfiguration attributes ...
}, ?,
"forwardingGroup": { "href": string }, ?,
"meterTemplates": [ 
  { "href": string, ?
  ... MeterTemplate attributes ... ?
  }, *
  ], ?,
"eventLogTemplate": [ 
  "href": string, ?
  ... EventLogTemplate attributes ... ?
  ], ?,
"operations": [ 
  { "rel": "edit", "href": string }, ?, 
  { "rel": "delete", "href": string } ?
  ]
... 
}
```

**XML media type:** application/xml

**XML serialization:**

```
<NetworkTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <initialState> xs:string </initialState> ?
  <networkConfig href="xs:anyURI"?>
    ... NetworkingConfiguration attributes ... ?
  </networkConfig> ?
</NetworkTemplate>
```
5.16.3.1 Operations

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

5.16.4 NetworkTemplateCollection Resource

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

**JSON serialization:**

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

**XML serialization:**

```xml
<Collection
    resourceURI="http://schemas.dmtf.org/cimi/1/NetworkTemplateCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
    <id> xs:anyURI </id>
    <count> xs:integer </count>
    <NetworkTemplate>
        <id> xs:anyURI </id>
```
5.16.4.1 Operations

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

5.16.5 NetworkConfiguration Resource

The following set of configuration values (shown in Table 39) represent the information needed to create a Network with certain characteristics.

<table>
<thead>
<tr>
<th>Name</th>
<th>NetworkConfiguration</th>
<th>TYPE URI</th>
<th><a href="http://schemas.dmtf.org/cimi/1/NetworkConfiguration">http://schemas.dmtf.org/cimi/1/NetworkConfiguration</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>networkType</td>
<td>An indicator of whether the Network is a Public or Private Network. Allowable values include: PUBLIC: represents an open and Internet routable network. PRIVATE: identifies a local non-Internet network. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
<td></td>
<td></td>
</tr>
<tr>
<td>mtu</td>
<td>(Maximum Transmission Unit) The largest supported packet size. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
<td></td>
<td></td>
</tr>
<tr>
<td>classOfService</td>
<td>The Provider's supported category associated with a Collection of attributes characterizing a level of a quality experience. Example values: GOLD: High bandwidth, low latency, low jitter SILVER: An improved service experience over bronze for voice or video traffic BRONZE: Best effort The list of possible values, and their implied quality of service, is out of scope of this specification. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

**JSON media type**: application/json

**JSON serialization**:

```
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkConfiguration",
  "id": string,
  "name": string,
  "description": string,
  "created": string
}
```
"updated": string, ?
"properties": { string: string, + }, ?
"networkType": string, ?
"mtu": number, ?
"classOfService": string, ?
"operations": [
  { "rel": "edit", "href": string }, ?
  { "rel": "delete", "href": string } ?
] ?
...
}

XML media type: application/xml

XML serialization:

```xml
<NetworkConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <networkType> xs:string </networkType> ?
  <mtu> xs:integer <mtu> ?
  <classOfService> xs:string </classOfService> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</NetworkConfiguration>
```

5.16.5.1 Operations

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

5.16.6 NetworkConfigurationCollection Resource

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

JSON serialization:

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

XML serialization:

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

5.16.6.1 Operations

This Resource supports the Read and Update operations. Creation of new NetworkConfiguration Resources is supported by the way of a POST to the "add" operation's URI as described in clause 4.2.1.1.
A NetworkPort is a realized connection point between a Network and a Resource, such as a Machine. Table 40 describes the NetworkPort attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type URI</th>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetworkPort</td>
<td><a href="http://schemas.dmtf.org/cimi/1/NetworkPort">http://schemas.dmtf.org/cimi/1/NetworkPort</a></td>
<td>state</td>
<td>string</td>
<td>The operational state of the NetworkPort. Allowable values include: CREATING: The NetworkPort is in the process of being created. STARTED: The NetworkPort is available (enabled) and ready for use. STOPPED: The NetworkPort is stopped (disabled) and not available for use. DELETING: The NetworkPort is in the process of being deleted. ERROR: The Provider has detected an error in the NetworkPort. The operations that result in transitions to the above defined states are defined in clause 5.16.7.2. Clause 5.16.8.1 defines the initial state of a NetworkPort. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td>network</td>
<td>ref</td>
<td>network</td>
<td>ref</td>
<td>A reference to the Network associated with this NetworkPort. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>portType</td>
<td>string</td>
<td>portType</td>
<td>string</td>
<td>A port is used as either an Access port (a member of the network) or a Trunk port that becomes a transport for multiple networks. Allowable values include: ACCESS: a member of a network. TRUNK: transport more than one network. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>classOfService</td>
<td>string</td>
<td>classOfService</td>
<td>string</td>
<td>The Provider-supported category associated with a collection of attributes characterizing a level of a quality experience. Example values: GOLD: High bandwidth, low latency, low jitter SILVER: An improved service experience over bronze for voice or video traffic BRONZE: Best effort The list of possible values, and their implied quality of service, is out of scope of this specification. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>meters</td>
<td>collection [Meter]</td>
<td>meters</td>
<td>collection</td>
<td>A reference to the list of Meters monitored for this NetworkPort. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>eventLog</td>
<td>ref</td>
<td>eventLog</td>
<td>ref</td>
<td>A reference to the EventLog of this NetworkPort. Constraints: Provider: support optional; mutable Consumer: support optional; read-only</td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPort",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "state": string,
  "network": { "href": string },
  "portType": string, ?
  "classOfService": string, ?
  "meters": { "href": string }, ?
  "eventLog": { "href": string }, ?
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string }, ?
    { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string } ?,
    { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
  ] ?
...
}
```

**XML media type:** application/xml

**XML serialization:**

```xml
<NetworkPort xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:string </created> ?
  <updated> xs:string </updated> ?
  <property key="xs:string"> xs:string </property> *
  <state> xs:string </state>
  <network href="xs:anyURI"/>
  <portType> xs:string </portType> ?
  <classOfService> xs:string </classOfService> ?
  <meters href="xs:anyURI"/> ?
```

5.16.7.1 Collections

The following clauses describe the Collection Resources owned by NetworkPorts.

5.16.7.1.1 NetworkPortMeterCollection Resource

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

JSON serialization:

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortMeterCollection",
  "id": string,
  "count": number,
  "meters": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
      "id": string,
      ... remaining Meter attributes ...
    },
    ...
  ],
  "operations": [ { "rel": "add", "href": string } ? ]
}
```

XML serialization:

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

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

Deleting a NetworkPort shall remove that NetworkPort from the global (Cloud Entry Point) NetworkPortCollection as well as from its corresponding Network's NetworkPortsCollection.

The following custom operations are also defined:

start

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

This operation shall start a NetworkPort.

Input parameters: None.

Output parameters: None.

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

HTTP protocol

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

JSON media type: application/json

JSON serialization:

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/start",
  "properties": {
    "string": string, +
  }
}
```

XML media type: application/xml

XML serialization

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

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

stop

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

This operation shall stop a NetworkPort. If stopped, the NetworkPort shall not be available for use and no network traffic shall flow through it.
Input parameters: None.

Output parameters: None.

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

**HTTP protocol**

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

**JSON media type: application/json**

**JSON serialization:**

```
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/stop",
  "properties": {
    string: string, +
  }
}
```

**XML media type: application/xml**

**XML serialization**

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

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

**5.16.8 NetworkPortCollection Resource**

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

**JSON serialization:**

```
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortCollection",
  "id": string,
  "count": number,
  "networkPorts": [ 
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPort",
      "id": string,
      ... remaining NetworkPort attributes ...
    }, +
  ],
  "operations": [ { "rel": "add", "href": string } ]
}
```
XML serialization:

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

5.16.8.1 Operations

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

If NetworkPorts are created through the global (Cloud Entry Point) NetworkPortCollection's "add" operation, they are added automatically to the corresponding Network's NetworkPortCollection Resource as well.

Upon successful processing of the "add" operation, unless otherwise specified by the NetworkPortTemplate "initialState" attribute, the state of the new NetworkPort shall be the value of the DefaultInitialState capability of the NetworkPort Resource's ResourceMetadata, if defined. If no DefaultInitialState capability is defined, the default value shall be "STOPPED." The semantics of "initialState" shall be equivalent to the Provider issuing the appropriate actions against the new NetworkPort to move it into that state.

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

5.16.9 NetworkPortTemplate Resource

The NetworkPortTemplate is a set of Configuration values for realizing a NetworkPort. A NetworkPortTemplate may be used to create multiple NetworkPorts. Table 41 describes the NetworkPortTemplate attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>NetworkPortTemplate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/NetworkPortTemplate">http://schemas.dmtf.org/cimi/1/NetworkPortTemplate</a></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>initialState</td>
<td>string</td>
</tr>
</tbody>
</table>

The initial state of the new NetworkPort. Possible values include the non-transient states as specified by the NetworkPort "state" attribute (i.e., STARTED, STOPPED) and shall be determined by the actions supported by the Provider. Providers should advertise the list of available values via the NetworkPort ResourceMetadata "initialStates" capability.

Constraints:
Provider: support optional; mutable
<table>
<thead>
<tr>
<th>Name</th>
<th>NetworkPortTemplate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/NetworkPortTemplate">http://schemas.dmtf.org/cimi/1/NetworkPortTemplate</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>ref</td>
<td>A reference to the network to be associated with this NetworkPort. If this Template is used to create a new NetworkPort through the global (Cloud Entry Point) NetworkPort Collection, this attribute shall be present. If this Template is used to create a new NetworkPort through a Network's NetworkPortsCollection, this attribute shall either be absent or have the same value as the &quot;id&quot; of the Network to which this NetworkPort is being added. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>networkPortConfig</td>
<td>ref</td>
<td>A reference to the NetworkPortConfiguration that is used to create a NetworkPort from this NetworkPortTemplate. Note that the attributes of the NetworkPortConfiguration may be specified rather than a reference to an existing NetworkPortConfiguration Resource. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>meterTemplates</td>
<td>meterTemplates[]</td>
<td>A list of references to MeterTemplates that shall be used to create and connect a set of new Meters to the new NetworkPort. Note that the attributes of the MeterTemplate may be specified rather than a reference to an existing MeterTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
<tr>
<td>eventLogTemplate</td>
<td>ref</td>
<td>A reference to an EventLogTemplate that shall be used to create and connect a new EventLog to the new NetworkPort. Note that the attributes of the EventLogTemplate may be specified rather than a reference to an existing EventLogTemplate Resource. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "initialState": string, ?
}
```
"network": { "href": string }, ?
"networkPortConfig": {
   "href": string | ... NetworkPortConfiguration attributes ...
},
"meterTemplates": [
   { "href": string, ?
   ... MeterTemplate attributes ... ?
    }, *
], ?,
"eventLogTemplate": {
   "href": string, ?
   ... EventLogTemplate attributes ... ?
}, ?,
"operations": [
   { "rel": "edit", "href": string }, ?,
   { "rel": "delete", "href": string } ?
] ?
...

XML media type: application/xml

XML serialization:

```xml
<NNetworkPortTemplate xmlns="http://schemas.dmtf.org/cimi/1">
   <id> xs:anyURI </id>
   <name> xs:string </name> ?
   <description> xs:string </description> ?
   <created> xs:dateTime </created> ?
   <updated> xs:dateTime </updated> ?
   <property key="xs:string"> xs:string </property> *
   <initialState> xs:string </initialState> ?
   <network href="xs:anyURI"/> ?
   <networkPortConfig href="xs:anyURI"/>
      ... NetworkPortConfiguration attributes ... ?
</networkPortConfig>
   <meterTemplate href="xs:anyURI"/>
      ... MeterTemplate attributes ... ?
</meterTemplate>*
   <eventLogTemplate href="xs:anyURI"/>
      ... EventLogTemplate attributes ... ?
</eventLogTemplate> ?
```
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

5.16.9.1 Operations

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

5.16.10 NetworkPortTemplateCollection Resource

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

**JSON serialization:**

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

**XML serialization:**

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

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

5.16.11 NetworkPortConfiguration Resource

The set of configuration values representing the information needed to create a NetworkPort with certain characteristics. Table 42 describes the NetworkPortConfiguration attributes.

Table 42 – NetworkPortConfiguration attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type URI</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td><a href="http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration">http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration</a></td>
</tr>
<tr>
<td>Name</td>
<td>NetworkPortConfiguration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type URI</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration">http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>portType</td>
<td>string</td>
<td></td>
<td>A port is used as an Access port (a member of the network) or a Trunk port that becomes a transport for multiple networks. Allowable values include: ACCESS: a member of a network. TRUNK: transport more than one network. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>classOfService</td>
<td>string</td>
<td></td>
<td>The Provider-supported category associated with a collection of attributes characterizing a level of a quality experience Example values: GOLD: High bandwidth, low latency, low jitter SILVER: An improved service experience over bronze for voice or video traffic BRONZE: Best effort The list of possible values, and their implied quality of service, is out of scope of this specification. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
</tbody>
</table>

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

JSON media type: application/json

JSON serialization:

```json
{
   "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration",
   "id": string,
   "name": string,
   "description": string,
   "created": string,
   "updated": string,
   "properties": { string: string, + },
   "portType": string,
   "classOfService": string,
   "operations": [
      { "rel": "edit", "href": string },
      { "rel": "delete", "href": string }
   ]
}
```
5.16.11.1 Operations
This Resource supports the Read, Update, and Delete operations. Create is supported through the
NetworkPortConfigurationCollection Resource.

5.16.12 NetworkPortConfigurationCollection Resource
A NetworkPortConfigurationCollection Resource represents the Collection of
NetworkPortConfigurations within a Provider and follows the Collection pattern defined in
clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

```json
{
  "resourceURI":
    "http://schemas.dmtf.org/cimi/1/NetworkPortConfigurationCollection",
  "id": string,
  "count": number,
  "networkPortConfigurations": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/NetworkPortConfiguration",
      "id": string,
      ... remaining NetworkPortConfiguration attributes ...
    }, +
  ],
  "operations": [ { "rel": "add", "href": string } ]
...
```
XML serialization:

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

5.16.12.1 Operations

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

5.16.13 Address Resource

An Address represents an IP address, and its associated metadata, for a particular Network. If a Consumer creates an Address Resource, it is the semantic equivalent of asking for a static IP address that can then be associated with Resources at a later point in time. Addresses that are manually created by Consumers shall not be deleted automatically if the Resource (e.g., a Machine) that is using that Address is deleted because these manually created Addresses are expected to have a lifetime that is different from the Resources that use them. Addresses that are created by Providers on the Consumer's behalf shall be deleted at the Provider's discretion. In particular, the Provider shall delete Addresses that it created on behalf of the Consumer if the Resource that is using that Address is deleted or if the Address becomes disassociated from the Resource.

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

Table 43 describes the Address attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/Address">http://schemas.dmtf.org/cimi/1/Address</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ip</td>
<td>string</td>
<td>The IP address assigned to a virtual interface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provider: support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>hostname</td>
<td>string</td>
<td>The DNS resolvable name associated with this network interface.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td>Name</td>
<td>Address</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------</td>
<td></td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/Address">http://schemas.dmtf.org/cimi/1/Address</a></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocation</td>
<td>string</td>
<td>The value is either &quot;dynamic&quot; or &quot;static&quot;. Expresses whether this Address is controlled by the Provider or Consumer. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td>defaultGateway</td>
<td>string</td>
<td>An IP address of a router that serves other networks. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
<tr>
<td>dns</td>
<td>string[]</td>
<td>The IP addresses of the Domain Name Services for host name to IP resolution. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
<tr>
<td>protocol</td>
<td>string</td>
<td>The selected network protocol, such as IPv4 or IPv6. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>mask</td>
<td>string</td>
<td>The network mask associated with this Address. Constraints: Provider: support optional; mutable Consumer: support optional; read-write</td>
</tr>
<tr>
<td>network</td>
<td>ref</td>
<td>A reference to the Network with which this Address is associated. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>resource</td>
<td>ref</td>
<td>A reference to the Resource that is using this Address. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Address",
  "id": "string",
  "name": "string", ?
  "description": "string", ?
  "created": "string", ?
  "updated": "string", ?
  "properties": { "string": string, + }, ?
  "ip": "string",
  "hostname": "string", ?
  "allocation": "string",
  "defaultGateway": "string", ?
  "dns": [ "string", + ], ?
```

---

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

 DSP0263
DSP0263  Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

5279

```
"protocol": string,
"mask": string, ?
"network": { "href": string },
"resource": { "href": string }, ?
"operations": [ 
  { "rel": "edit", "href": string }, ?
  { "rel": "delete", "href": string } ?
] ?
... }
```

5289  XML media type: application/xml

5290  XML serialization:

```
<Address xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <ip> xs:string </ip>
  <hostname> xs:string </hostname> ?
  <allocation> xs:string </allocation>
  <defaultGateway> xs:string </defaultGateway> ?
  <dns> xs:string </dns> *
  <protocol> xs:string </protocol>
  <mask> xs:string </mask> ?
  <network href="xs:anyURI"/>
  <resource href="xs:anyURI"/> ?
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</Address>
```

5311  5.16.13.1 Operations

5312  This Resource supports the Read, Update, and Delete operations. Create is supported through the
5313  AddressCollection Resource.

5314  5.16.14  AddressCollection Resource

5315  An AddressCollection Resource represents the Collection of Addresses within a Provider that
5316  are owned/managed by the Consumer or Provider and follows the Collection pattern defined in clause
5317  5.5.12. This Resource shall be serialized as follows:
5318 JSON serialization:

```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/AddressCollection",
    "id": string,
    "count": number,
    "addresses": [
        {
            "resourceURI": "http://schemas.dmtf.org/cimi/1/Address",
            "id": string,
            ... remaining Address attributes ...
        }, +
    ],
    "operations": [ { "rel": "add", "href": string } ]
}
```

XML serialization:

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

5.16.14.1 Operations

NOTE The "add" operation requires that an AddressTemplate be used (see 4.2.1.1).

5.16.15 AddressTemplate Resource

This Resource captures the configuration values for realizing an Address. An AddressTemplate may be used to create multiple Addresses. Table 44 describes the AddressTemplate attributes.

### Table 44 – AddressTemplate attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>AddressTemplate Type URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td><a href="http://schemas.dmtf.org/cimi/1/AddressTemplate">http://schemas.dmtf.org/cimi/1/AddressTemplate</a></td>
<td>Type Description</td>
</tr>
<tr>
<td>ip</td>
<td>string</td>
<td>The IP address assigned to a virtual interface.</td>
</tr>
<tr>
<td></td>
<td>Constraints:</td>
<td>Provider: support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td>Constraints:</td>
<td>Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>hostname</td>
<td>string</td>
<td>The DNS resolvable name associated with this network interface.</td>
</tr>
<tr>
<td></td>
<td>Constraints:</td>
<td>Provider: support optional; mutable</td>
</tr>
<tr>
<td>Name</td>
<td>AddressTemplate</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/AddressTemplate">http://schemas.dmtf.org/cimi/1/AddressTemplate</a></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocation</td>
<td>string</td>
<td>A value of either &quot;dynamic&quot; or &quot;static&quot;. Expresses whether this address is controlled by the Provider or Consumer.</td>
</tr>
<tr>
<td>defaultGateway</td>
<td>string</td>
<td>An IP address of a router that serves other networks.</td>
</tr>
<tr>
<td>dns</td>
<td>string[]</td>
<td>The IP addresses of the Domain Name Services for host name to IP resolution.</td>
</tr>
<tr>
<td>protocol</td>
<td>string</td>
<td>The selected network protocol, such as IPv4 or IPv6.</td>
</tr>
<tr>
<td>mask</td>
<td>string</td>
<td>The network mask associated with this Address.</td>
</tr>
<tr>
<td>network</td>
<td>ref</td>
<td>A reference to the Network with which this Address is associated.</td>
</tr>
</tbody>
</table>

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

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/AddressTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "ip": string,
  "hostname": string, ?
  "allocation": string,
  "defaultGateway": string, ?
  "dns": [ string, + ], ?
}```
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

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

XML media type: application/xml

XML serialization:
```
<AddressTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <ip> xs:string </ip>
  <hostname> xs:string </hostname> ?
  <allocation> xs:string </allocation>
  <defaultGateway> xs:string </defaultGateway>
  <dns> xs:string </dns> +
  <protocol> xs:string </protocol>
  <mask> xs:string </mask>
  <network href="xs:anyURI"/>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</AddressTemplate>
```

5.16.15.1 Operations

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

5.16.16 AddressTemplateCollection Resource

An AddressTemplateCollection Resource represents the Collection of AddressTemplate Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:
5.16.16.1 Operations

This Resource supports the Read and Update operations. Creation of new AddressTemplate Resources is supported by the way of a POST to the "addLink" URI as described in clause 4.2.1.1.

5.16.17 ForwardingGroup Resource

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

Networks in a ForwardingGroup should all have the same "networkType" attributes, which prevents a Network with a "private" networkType attribute from being publicly forwarded because it is a member of a ForwardingGroup that also contains Networks with a "public" networkType attribute.

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

Table 45 describes the ForwardingGroup attributes.
Table 45 – ForwardingGroup attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type URI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>networks</td>
<td>collection [Forwarding Group, Network]</td>
<td>A reference to the list of references to the Networks in this ForwardingGroup.</td>
</tr>
</tbody>
</table>

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

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

**JSON media type:** application/json

**JSON serialization:**

```
{   "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroup",
    "id": string,
    "name": string, ?
    "description": string, ?
    "created": string, ?
    "updated": string, ?
    "properties": { string: string, + }, ?
    "networks": [
        { "href": string }, +
    ], ?
    "operations": [
        { "rel": "edit", "href": string }, ?
        { "rel": "delete", "href": string } ?
    ] ?
... }
```

**XML media type:** application/xml

**XML serialization:**

```
<ForwardingGroup xmlns="http://schemas.dmtf.org/cimi/1">
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <network href="xs:anyURI"> *
</ForwardingGroup>
```
5.16.17.1 Collections

The following clauses describe the Collection Resources owned by ForwardingGroups.

5.16.17.1.1 ForwardingGroupNetworkCollection Resource

The Resource type for each item of this Collection is "ForwardingGroupNetwork" defined in Table 46:

<table>
<thead>
<tr>
<th>Name</th>
<th>ForwardingGroupNetwork</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/ForwardingGroupNetwork">http://schemas.dmtf.org/cimi/1/ForwardingGroupNetwork</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>network</td>
<td>ref</td>
<td>A reference to a Network in the ForwardingGroup.</td>
</tr>
</tbody>
</table>

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

**Table 46 – ForwardingGroupNetwork attributes**

**JSON serialization:**

```json
{
   "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroupNetworkCollection",
   "id": string,
   "count": number,
   "forwardingGroupNetworks": [
   {
      "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroupNetwork",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "network": { "href": string },
      "operations": [
      { "rel": "edit", "href": string }, ?
      { "rel": "delete", "href": string } ?
      ] ?
      ... |
      
   }, +
   
   ... |
   
   "operations": [ { "rel": "add", "href": string } ? ]
   ... |
```
5.16.17.2 Operations

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

5.16.18 ForwardingGroupCollection Resource

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

JSON serialization:

```json
{
    "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroupCollection",
    "id": string,
    "count": number,
    "forwardingGroups": [
        { "resourceURI": "http://schemas.dmtf.org/cimi/1/ForwardingGroup",
        "id": string,
        ... remaining ForwardingGroup attributes ...
    ],
}
```
textarea here
XML media type: application/xml

XML serialization:

```xml
<ForwardingGroupTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id>xs:anyURI</id>
  <name>xs:string</name>
  <description>xs:string</description>
  <created>xs:dateTime</created>
  <updated>xs:dateTime</updated>
  <property key="xs:string">xs:string</property>
  <network href="xs:anyURI"/>
  <operation rel="edit" href="xs:anyURI"/>
  <operation rel="delete" href="xs:anyURI"/>
</ForwardingGroupTemplate>
```

5.16.19.1 Operations

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

5.16.20 ForwardingGroupTemplateCollection Resource

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

JSON serialization:

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

XML serialization:

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

5.16.20.1 Operations

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

5.17 Monitoring Resources and relationships

Figure 6 illustrates the Resources involved in tracking the progress of operations, as well as, metering and monitoring the status of other Resources. Although this drawing is in the style of a Resource Relationship diagram, the use of UML is neither rigorous nor normative.
5.17.1 Job Resource

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

If a Provider supports exposing Job Resources to Consumers, each request from a Consumer that would result in a change to the environment shall result in a Job Resource being created and an absolute URI reference to that Job Resource shall be made available to the requesting Consumer. Providers may create additional Job Resources for Provider-initiated operations if the Provider chooses to expose these Jobs to Consumers.

If a Job is not completed successfully (e.g., it is in the FAILED or STOPPED state), this specification does not place any requirements on the Provider to ensure that the affected Resources are left in certain states. Based on the environmental conditions at that time, the Provider might choose to "undo" any impact of the operation; simply halt processing; attempt some kind of "cleanup" action; or choose to do something else. However, Providers shall list all Resources impacted by the Job in the "affectedResources" attribute, thus allowing Consumers an opportunity to examine the state of each Resource themselves. In cases where a Resource has been deleted, references to that Resource shall not appear in the "affectedResources" attribute.

The Job Resource allows for nesting of Jobs. The determination of when a single operation is converted into multiple nested Jobs is out of scope of this specification. However, if there are nested Jobs, the top-most Job Resource shall report the overall status of all Jobs and shall only be in a...
"SUCCESS" state if all nested Jobs are also in "SUCCESS" state. If nested Jobs are created, there is no requirement for the top-most Job Resource to reference all affected Resources in its "affectedResources" attribute. The Consumer needs to traverse the entire set of nested Jobs to determine the complete list of Resources impacted by the Jobs.

Table 48 describes the Job attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>state</strong></td>
<td>string</td>
<td>The state of the process associated with this operation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Provider:</strong> support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Consumer:</strong> support mandatory; read-only</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Allowable values include:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>QUEUED: Indicates that the operation has not yet begun processing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>RUNNING: Indicates that the operation is still being executed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FAILED: Indicates that the operation failed to be completed successfully.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SUCCESS: Indicates that the operation was successfully completed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STOPPING: Indicates that the operation is in the process of being stopped.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>STOPPED: Indicates that the operation was stopped before completion.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The operations that result in transitions to the above defined states are defined in clause 5.17.1.1</td>
</tr>
<tr>
<td><strong>targetResource</strong></td>
<td>ref</td>
<td>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 &quot;add&quot; Job is executed against a &quot;Collection&quot; 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 &quot;affectedResources&quot; attribute.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Provider:</strong> support mandatory; immutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Consumer:</strong> support mandatory; read-only</td>
</tr>
<tr>
<td><strong>affectedResources</strong></td>
<td>ref[]</td>
<td>A list of references to Resources that have been impacted by this Job. Note that this list shall always contain the &quot;targetResource&quot; reference. Array item name: affectedResource</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Provider:</strong> support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Consumer:</strong> support mandatory; read-only</td>
</tr>
<tr>
<td><strong>action</strong></td>
<td>URI</td>
<td>A URI that indicates the type of action being performed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Provider:</strong> support mandatory; immutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Consumer:</strong> support mandatory; read-only</td>
</tr>
<tr>
<td><strong>returnCode</strong></td>
<td>integer</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Provider:</strong> support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Consumer:</strong> support mandatory; read-only</td>
</tr>
<tr>
<td><strong>progress</strong></td>
<td>integer</td>
<td>An integer value in the range 0 ... 100 that indicates the progress of this Job. This value shall be 100 if the Job is no longer executing, regardless of the outcome.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Provider:</strong> support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Consumer:</strong> support mandatory; read-only</td>
</tr>
<tr>
<td><strong>statusMessage</strong></td>
<td>string</td>
<td>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</td>
</tr>
<tr>
<td>Name</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Job</td>
<td>URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/Job">http://schemas.dmtf.org/cimi/1/Job</a></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>failed, or whether the operation was cancelled by the Consumer or the Provider.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints: Provider: support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td></td>
<td>dateTime</td>
<td>A timestamp indicating the last time that the status of the operation changed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints: Provider: support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td></td>
<td>ref</td>
<td>A reference to the Job of which this Resource is a subordinate.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints: Provider: support mandatory; immutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td></td>
<td>ref[]</td>
<td>An array of references to a set of subordinate Job Resources.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Array item name: nestedJob</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Constraints: Provider: support mandatory; mutable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Consumer: support mandatory; read-only</td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
"resourceURI": "http://schemas.dmtf.org/cimi/1/Job",
"id": string,
"name": string, ?
"description": string, ?
"created": string, ?
"updated": string, ?
"properties": { string: string, + }, ?
"state": string,
"targetResource": { "href": string },
"affectedResources": [ { "href": string }, + ],
"action": string,
"returnCode": number,
"progress": number,
"statusMessage": string,
"timeOfStatusChange": date,
"parentJob": [ "href": string ], ?
"nestedJobs": [ 
  { "href": string }, +
], ?
"operations": [ 
```
XML media type: application/xml

XML serialization:

```xml
<Job xmlns="http://schemas.dmtf.org/cimi/1">
  <id>xs:anyURI</id>
  <name>xs:string</name> ?
  <description>xs:string</description> ?
  <created>xs:dateTime</created> ?
  <updated>xs:dateTime</updated> ?
  <property key="xs:string">xs:string</property> *
  <state>xs:string</state>
  <targetResource href="xs:anyURI"/>
  <affectedResource href="xs:anyURI"/> +
  <action>xs:anyURI</action>
  <returnCode>xs:integer</returnCode>
  <progress>xs:integer</progress>
  <statusMessage>xs:string</statusMessage>
  <timeOfStatusChange>xs:dateTime</timeOfStatusChange>
  <parentJob href="xs:anyURI"/> ?
  <nestedJob href="xs:anyURI"/> *
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <operation rel="http://schemas.dmtf.org/cimi/1/action/stop" href="xs:anyURI"/> ?
  <xs:any>*
</Job>
```

5.17.1.1 Operations Resource

This Resource supports the Read, Update, and Delete operations. Deleting a Job that is in the "RUNNING" state shall be the equivalent of first stopping the Job and then deleting it. A request to delete a running Job that does not support the "stop" action shall fail.

The following custom operations are also defined:

stop

/link@rel: http://schemas.dmtf.org/cimi/1/action/stop
This operation shall stop a Job.

Input parameters: None.

Output parameters: None.

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

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

**HTTP protocol**

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/stop",
  "properties": [{
    "string": string,
  }]
}
```

**XML media type:** application/xml

**XML serialization**

```xml
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action>
    http://schemas.dmtf.org/cimi/1/action/stop
  </action>
  <property key="xs:string"> xs:string </property>*
  <xs:any>*
</Action>
```

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

### 5.17.2 JobCollection Resource

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

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/JobCollection",
  "id": string,
  "count": integer,
  "jobs": [
    {
      "resourceURI": "http://schemas.dmtf.org/cimi/1/Job",
      "id": string,
      ... remaining Job attributes ...
    }, +
  ]
}
```
XML serialization:

```xml
<Collection resourceURI="http://schemas.dmtf.org/cimi/1/JobCollection"
    xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Job>
    <id> xs:anyURI </id>
    ... remaining Job attributes ... 
  </Job> *
  <xs:any>*
</Collection>
```

5.17.3 Meter Resource

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

If a Meter's "targetResource" is deleted all Meters associated with that Resource shall also be deleted. In other words, deleting a Resource-specific MetersCollection (e.g., a Machine's MetersCollection) shall also result in the deletion of the Meters referenced from that Collection.

Table 49 describes the Meter attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Meter</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetResource</td>
<td>ref</td>
</tr>
<tr>
<td>Aspect</td>
<td>URI</td>
</tr>
<tr>
<td>Units</td>
<td>string</td>
</tr>
<tr>
<td>SampleInterval</td>
<td>integer</td>
</tr>
<tr>
<td>TimeScope</td>
<td>string</td>
</tr>
</tbody>
</table>

Table 49 – Meter attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetResource</td>
<td>ref</td>
<td>A reference to the Resource to which the Meter is related. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td>Aspect</td>
<td>URI</td>
<td>A unique identifier representing the aspect of the Resource being metered. Constraints: Provider: support mandatory; immutable Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td>Units</td>
<td>string</td>
<td>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</td>
</tr>
<tr>
<td>SampleInterval</td>
<td>integer</td>
<td>The time between consecutive samples in seconds. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td>TimeScope</td>
<td>string</td>
<td>The time scope to which this meter’s value applies. Two possible values: &quot;Point&quot; indicates that the Meter applies to a point in time. &quot;Interval&quot; 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</td>
</tr>
<tr>
<td>Name</td>
<td>Type URI</td>
<td>Type</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td>Name</td>
<td>Meter</td>
<td></td>
</tr>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/Meter">http://schemas.dmtf.org/cimi/1/Meter</a></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>intervalDuration</td>
<td>duration</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Provider:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consumer:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints:</td>
<td></td>
</tr>
<tr>
<td>isContinuous</td>
<td>boolean</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Provider:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consumer:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints:</td>
<td></td>
</tr>
<tr>
<td>samples</td>
<td>collection</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Sample]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Provider:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consumer:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints:</td>
<td></td>
</tr>
<tr>
<td>minValue</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Provider:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consumer:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints:</td>
<td></td>
</tr>
<tr>
<td>maxValue</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Provider:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consumer:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints:</td>
<td></td>
</tr>
<tr>
<td>stopTime</td>
<td>dateTime</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Provider:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consumer:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints:</td>
<td></td>
</tr>
<tr>
<td>expiresTime</td>
<td>dateTime</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Provider:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consumer:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Constraints:</td>
<td></td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Meter",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "targetResource": { "href": string },
  "aspect": string,
  "units": string,
}  ```
"sampleInterval": number,
"timeScope": string,
"intervalDuration": string,
"isContinuous": boolean,
"samples": { "href": string }, ?
"minValue": string, ?
"maxValue": string, ?
"stopTime": string, ?
"expiresTime": string, ?
"operations": [  
  { "rel": "edit", "href": string }, ?
  { "rel": "delete", "href": string }, ?
  { "rel": "http://schemas.dmtf.org/cimi/1/action/start", "href": string }, ?
  { "rel": "http://schemas.dmtf.org/cimi/1/action/stop", "href": string } ?
] ?
...
}

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

```xml
<Meter xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <targetResource href="xs:anyURI"/>  
  <aspect> xs:anyURI </aspect>
  <units> xs:string </units>
  <sampleInterval> xs:integer </sampleInterval>
  <timeScope> xs:string </timeScope>
  <intervalDuration xs:duration </intervalDuration>
  <isContinuous> xs:boolean </isContinuous>
  <samples href="xs:anyURI"/> ?
  <minValue> xs:string </minValue> ?
  <maxValue> xs:string </maxValue> ?
  <stopTime> xs:dateTime </stopTime> ?
  <expiresTime> xs:dateTime </expiresTime> ?
  <operation rel="edit" href="xs:anyURI"/> ?
```
5.17.3.1 Collections

The following clauses describe the Collection resources owned by Meters.

5.17.3.1.1 SampleCollection Resource

The Resource type for each item of this Collection is "Sample", defined in Table 50:

Table 50 – Sample attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>Sample Type URI</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timestamp</td>
<td><a href="http://schemas.dmtf.org/cimi/1/Sample">http://schemas.dmtf.org/cimi/1/Sample</a></td>
<td>dateTime</td>
<td>Indicates when the measure was taken (timeScope=&quot;Point&quot;). If the timeScope is &quot;Interval&quot;, it indicates the end of the time interval. <strong>Constraints:</strong> Provider: support mandatory; immutable Consumer: support mandatory; read-only</td>
</tr>
<tr>
<td>value</td>
<td></td>
<td>string</td>
<td>Indicates the sampled value of the measure. <strong>Constraints:</strong> Provider: support mandatory; immutable Consumer: support mandatory; read-only</td>
</tr>
</tbody>
</table>

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

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/SampleCollection",
  "id": string,
  "count": number,
  "samples": [
    {
      "resourceURI": "http://schemas.dmtf.org/cimi/1/Sample",
      "id": string,
      "name": string, ?
      "description": string, ?
      "created": string, ?
      "updated": string, ?
      "properties": { string: string, + }, ?
      "timestamp": string,
      "value": string
    }
  ]
}```
... }

XML serialization:

```xml
<Collection
  resourceURI="http://schemas.dmtf.org/cimi/1/SampleCollection"
  xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <count> xs:integer </count>
  <Sample>
    <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>*
    <sample timestamp="xs:dateTime" value="xs:string"/>
    <xs:any>*
  </Sample>*
</Collection>
```

### 5.17.3.2 Operations

This Resource supports the Read, Update, and Delete operations. Create is supported via the MeterCollection Resource. The deletion of a Meter shall remove the Meter from the targetResource’s “meter” attribute.

The following custom operations are also defined:

**start**

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

This operation shall start a Meter.

Input parameters: None.

Output parameters: None.

Upon successful completion of this operation, the Meter shall start recording samples related to its associated Resource.

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

**JSON media type: application/json**

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/start",
  "properties": { string: string, + } ?
  ...
}
```

**XML media type: application/xml**

**XML serialization**

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

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

**stop**

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

This operation shall stop a Meter.

Input parameters: None.

Output parameters: None.

Upon successful completion of this operation, the Meter shall no longer be recording samples related to its associated Resource.

**HTTP protocol**

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

**JSON media type: application/json**

**JSON serialization:**

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/Action",
  "action": "http://schemas.dmtf.org/cimi/1/action/stop",
  "properties": { string: string, + } ?
  ...
}
```
XML media type: application/xml

XML serialization:

```xml
<Action xmlns="http://schemas.dmtf.org/cimi/1">
  <action> http://schemas.dmtf.org/cimi/1/action/stop </action>
  <property key="xs:string"> xs:string </property> *
  <xs:any>*
</Action>
```

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

### 5.17.4 MeterCollection Resource

A `MeterCollection` Resource represents the Collection of Meters within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

#### JSON serialization:

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

#### XML serialization:

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

#### 5.17.4.1 Operations

NOTE The "add" operation requires that a `MeterTemplate` be used (see 4.2.1.1).
If Meters are created through the global (Cloud Entry Point) MeterCollection's "add" operation, they shall be added automatically to the corresponding targetResource's "Meters" Collection Resource as well.

5.17.5 MeterTemplate Resource

A MeterTemplate represents the information needed to create a new Meter. Table 51 describes the MeterTemplate attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>TargetResource</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MeterTemplate">http://schemas.dmtf.org/cimi/1/MeterTemplate</a></td>
</tr>
<tr>
<td>MeterConfig</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MeterConfiguration">http://schemas.dmtf.org/cimi/1/MeterConfiguration</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>targetResource</td>
<td>ref</td>
<td>A reference to the Resource that is metered. The type of the Resource shall be one of the &quot;associatedTo&quot; types listed in the MeterConfiguration referenced. If this Template is used to create a new Meter through the global (Cloud Entry Point) MeterCollection, this attribute shall be present. If this Template is used to create a new Meter through a targetResource's MeterCollection, this attribute shall either be absent or have the same value as the &quot;id&quot; of the targetResource to which this Meter is being added. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>meterConfig</td>
<td>ref</td>
<td>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</td>
</tr>
</tbody>
</table>

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

**JSON media type:** application/json

**JSON serialization:**

```json
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "targetResource": { string },
  "meterConfig": {
    "href": string | ... MeterConfiguration attributes ...
  },
  "operations": [
```
XML media type: application/xml

XML serialization:

```
<MeterTemplate xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <targetResource href="xs:anyURI"/>
  <meterConfig href="xs:anyURI">
    ... MeterConfiguration attributes ... ?
  </meterConfig>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any> *
</MeterTemplate>
```

5.17.6 MeterTemplateCollection Resource

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

JSON serialization:

```
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterTemplateCollection",
  "id": string,
  "count": number,
  "meterTemplates": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterTemplate",
      "id": string,
      ... remaining MeterTemplate attributes ...
    }, +
  ],
  "operations": [ { "rel": "add", "href": string } ? ]
  ...
}
```
XML serialization:

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

5.17.6.1 Operations

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

5.17.7 MeterConfiguration Resource

A MeterConfiguration represents the definition of a Meter. Table 52 describes the MeterConfiguration attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>MeterConfiguration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MeterConfiguration">http://schemas.dmtf.org/cimi/1/MeterConfiguration</a></td>
</tr>
<tr>
<td>AssociatedTo</td>
<td>URI[]</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssociatedTo</td>
<td>URI[]</td>
<td>A unique identifier representing the aspect of the Resource being metered. See table 53 below for the set of CIM-defined URIs.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Units</td>
<td>string</td>
<td>The human-readable name of the used units, e.g., kilobits per second, CPU usage percentage, etc.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SampleInterval</td>
<td>integer</td>
<td>The time between consecutive samples in seconds.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeScope</td>
<td>string</td>
<td>The time scope to which the Meter value applies. Two possible values: &quot;Point&quot; indicates that the Meter applies to a point in time. &quot;Interval&quot; indicates that the Meter applies to a time interval. For instance, it would be</td>
</tr>
</tbody>
</table>

```markdown
Table 52 – MeterConfiguration attributes

<table>
<thead>
<tr>
<th>Name</th>
<th>MeterConfiguration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/MeterConfiguration">http://schemas.dmtf.org/cimi/1/MeterConfiguration</a></td>
</tr>
<tr>
<td>AssociatedTo</td>
<td>URI[]</td>
</tr>
<tr>
<td>Description</td>
<td>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.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AssociatedTo</td>
<td>URI[]</td>
<td>A unique identifier representing the aspect of the Resource being metered. See table 53 below for the set of CIM-defined URIs.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Units</td>
<td>string</td>
<td>The human-readable name of the used units, e.g., kilobits per second, CPU usage percentage, etc.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SampleInterval</td>
<td>integer</td>
<td>The time between consecutive samples in seconds.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeScope</td>
<td>string</td>
<td>The time scope to which the Meter value applies. Two possible values: &quot;Point&quot; indicates that the Meter applies to a point in time. &quot;Interval&quot; indicates that the Meter applies to a time interval. For instance, it would be</td>
</tr>
</tbody>
</table>
**Name**: MeterConfiguration

**Type URI**: http://schemas.dmtf.org/cimi/1/MeterConfiguration

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>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</td>
<td></td>
<td></td>
</tr>
<tr>
<td>intervalDuration</td>
<td>duration</td>
<td>The interval duration when the timeScope is set to &quot;Interval.&quot; Possible values: hourly, daily, weekly, monthly, or yearly. Constraints: Provider: support mandatory; mutable Consumer: support mandatory; read-write</td>
</tr>
<tr>
<td>isContinuous</td>
<td>boolean</td>
<td>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</td>
</tr>
</tbody>
</table>

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

**JSON media type**: application/json

**JSON serialization**:

```json
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/MeterConfiguration",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
  "associatedTo": [
    { "href": string }, +
  ], ?
  "aspect": string,
  "units": string,
  "sampleInterval": number,
  "timeScope": string,
  "intervalDuration": string,
  "isContinuous": boolean,
  "operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
  ] ?
... }
```
XML media type: application/xml

XML serialization:

```xml
<MeterConfiguration xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <associatedTo href="xs:anyURI"/> *
  <aspect> xs:anyURI </aspect>
  <units> xs:string </units>
  <sampleInterval> xs:integer </sampleInterval>
  <timeScope> xs:string </timeScope>
  <intervalDuration> xs:duration </intervalDuration>
  <isContinuous> xs:boolean </isContinuous>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</MeterConfiguration>
```

Table 53 describes the "aspect" URIs defined by this specification. Providers may define new aspect URIs and it is recommended that these URIs be dereferencable such that Consumers can discover the details of the new aspect. For brevity the "URI" column in the table only shows the last part of the URI. It should be appended to: "http://schemas.dmtf.org/cimi/1/aspect/".

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu</td>
<td>The percentage CPU usage of the Resource. Typically associated with CloudEntryPoint, System, and Machine Resources. For Resources that group other Resources (e.g., CloudEntryPoint or System Resources), this aspect provides the aggregated percentage usage of the CPU.</td>
</tr>
<tr>
<td>memory</td>
<td>The amount of memory being used by the Resource. Typically associated with CloudEntryPoint, System, and Machine Resources. For Resources that group other Resources (e.g., CloudEntryPoint or System Resources), this aspect provides the aggregated usage of the memory.</td>
</tr>
<tr>
<td>disk</td>
<td>The amount of disk being used by the Resource. Typically associated with CloudEntryPoint, System, Machine, and Volume Resources. For Resources that group other Resources (e.g., CloudEntryPoint or System Resources), this aspect provides the aggregated disk usage.</td>
</tr>
<tr>
<td>bandwidth</td>
<td>The amount of network traffic. Typically associated with CloudEntryPoint, System, and Network Resources. For CloudEntryPoint and System Resources, this aspect provides the aggregated bandwidth of all the networks under them.</td>
</tr>
<tr>
<td>inputBandwidth</td>
<td>The amount of input bandwidth used by the Resource. Typically associated with Machine, NetworkPort, and Volume Resources. For Machine Resources, this aspect provides the aggregated input bandwidth usage of all its network interfaces.</td>
</tr>
</tbody>
</table>
Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>outputBandwidth</td>
<td>The amount of output bandwidth used by the Resource. Typically associated with Machine, NetworkPort, and Volume Resources. For Machine Resources, this aspect provides the aggregated output bandwidth usage of all its network interfaces.</td>
</tr>
</tbody>
</table>

5.17.7.1 Operations

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

5.17.8 MeterConfigurationCollection Resource

A MeterConfigurationCollection Resource represents the Collection of MeterConfigurations within a Provider and follows the Collection pattern defined in clause 5.5.12.

This Resource shall be serialized as follows:

**JSON serialization:**

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

**XML serialization:**

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

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

5.17.9 EventLog Resource

A Resource that represents a registry of Events.

If an EventLog's "targetResource" is deleted the EventLog associated with that Resource may also be deleted. In other words, deleting a Resource (e.g., a Machine) may also result in the deletion of the EventLog referenced from that Resource. This behavior is denoted by the EventLog.Linked capability.

If an EventLog is deleted, all of its Events shall also be deleted.

Table 54 describes the EventLog attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>EventLog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/EventLog">http://schemas.dmtf.org/cimi/1/EventLog</a></td>
</tr>
<tr>
<td>Attribute</td>
<td>Type</td>
</tr>
<tr>
<td>targetResource</td>
<td>ref</td>
</tr>
<tr>
<td>events</td>
<td>collection [Event]</td>
</tr>
<tr>
<td>persistence</td>
<td>string</td>
</tr>
</tbody>
</table>
### EventLog

**Name**: EventLog  
**Type URI**: http://schemas.dmtf.org/cimi/1/EventLog

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>summary</td>
<td><code>&lt;unnamed structure&gt;</code></td>
<td>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:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| 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

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

**JSON media type**: application/json

**JSON serialization**:

```json
{  "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLog",  "id": string,  "name": string,  "description": string,  "created": string,  "updated": string,  "properties": { "string": string, + },  "targetResource": { "href": string },  "events": { "href": string },  "persistence": string,  "summary": {    "low": number,    "medium": number,    "high": number,    "critical": number  
```

---

Version 1.1.0

DMTF Standard

203
XML media type: application/xml

XML serialization:

```xml
<EventLog xmlns="http://schemas.dmtf.org/cimi/1">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <property key="xs:string"> xs:string </property> *
  <targetResource href="xs:anyURI"/>
  <events href="xs:anyURI"/>
  <persistence> xs:string </persistence>
  <summary>
    <low> xs:integer </low>
    <medium> xs:integer </medium>
    <high> xs:integer </high>
    <critical> xs:integer </critical>
  </summary>
  <operation rel="edit" href="xs:anyURI"/> ?
  <operation rel="delete" href="xs:anyURI"/> ?
  <xs:any>*
</EventLog>
```

5.17.9.1 Collections

The following clauses describe the Collection Resources owned by EventLogs.

5.17.9.1.1 EventCollection Resource

The Resource type for each item of this Collection is "Event" as defined in clause 5.17.13.

JSON serialization:

```json
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/EventCollection",
  "id": string,
  "count": number,
  "events": [}
```
[ ... remaining Event attributes ... ]
"operations": [ { "rel": "add", "href": string } ]
... 

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

5.17.9.2 Operations
This Resource supports the Read, Update, and Delete operations.

5.17.10 EventLogCollection Resource
An EventLogCollection Resource represents the Collection of EventLogs within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:
[ ... remaining EventLog attributes ... ]
"operations": [ { "rel": "add", "href": string } ]
... 

{ "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogCollection",
  "id": string,
  "count": number,
  "eventLogs": [
    { "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLog",
      "id": string,
      ... remaining EventLog attributes ...
    }, +
  ],
  "operations": [ { "rel": "add", "href": string } ]
}
### 5.17.11 EventLogTemplate Resource

An EventLogTemplate represents the information needed to create a new EventLog. Table 55 describes the EventLogTemplate attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>EventLogTemplate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type URI</td>
<td><a href="http://schemas.dmtf.org/cimi/1/EventLogTemplate">http://schemas.dmtf.org/cimi/1/EventLogTemplate</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>targetResource</td>
<td>ref</td>
<td>A reference to the Resource to which the EventLog shall be connected.</td>
</tr>
<tr>
<td>persistence</td>
<td>string</td>
<td>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</td>
</tr>
</tbody>
</table>

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

#### JSON media type: application/json

#### JSON serialization:

```json
{
  "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogTemplate",
  "id": string,
  "name": string, ?
  "description": string, ?
  "created": string, ?
  "updated": string, ?
  "properties": { string: string, + }, ?
}```
"targetResource": [ string ],
"persistence": string,
"operations": [
    { "rel": "edit", "href": string }, ?
    { "rel": "delete", "href": string } ?
],
...}

XML media type: application/xml

XML serialization:

```xml
<EventLogTemplate xmlns="http://schemas.dmtf.org/cimi/1">
    <id> xs:anyURI </id>
    <name> xs:string </name> ?
    <description> xs:string </description> ?
    <created> xs:dateTime </created> ?
    <updated> xs:dateTime </updated> ?
    <property key="xs:string"> xs:string </property> *
    <targetResource href="xs:anyURI"/>
    <persistence> xs:string </persistence>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
    <xs:any>*
</MeterTemplate>
```

5.17.12 EventLogTemplateCollection Resource

An EventLogTemplateCollection Resource represents the Collection of EventLogTemplate Resources within a Provider and follows the Collection pattern defined in clause 5.5.12. This Resource shall be serialized as follows:

JSON serialization:

```json
{ "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogTemplateCollection",
    "id": string,
    "count": number,
    "eventLogTemplates": [
        { "resourceURI": "http://schemas.dmtf.org/cimi/1/EventLogTemplate",
            "id": string,
            ... remaining EventLogTemplate attributes ...
        }, +
    ],
    "operations": [ { "rel": "add", "href": string } ? ]
...
XML serialization:

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

5.17.12.1 Operations

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

5.17.13 Event Resource

A Resource that represents the occurrence of an event within the managed infrastructure. Some examples of Event are:

- Machine X has been rebooted by guest OS.
- Machine X is not responding to platform services.
- A new vCPU has been added to machine X following defined elasticity rules.

The scope of the Event concept is any information that the Provider is able to track within its infrastructure and that can constitute useful information for the Consumer. Possible examples include, but are not limited to, errors and inconveniences that occur in the (virtual) resources assigned to Consumers; Provider-initiated actions, such as maintenance tasks; etc.

Table 56 describes the Event attributes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Event</th>
<th>Type URI</th>
<th><a href="http://schemas.dmtf.org/cimi/1/Event">http://schemas.dmtf.org/cimi/1/Event</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Attribute</td>
<td>Type</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>timestamp</td>
<td>date/time</td>
<td>The time of occurrence of the actual Event. NOTE: This attribute should not be confused with the time of creation of the Event Resource instance, which is captured in the common &quot;created&quot; attribute. Constraints: Provider: support mandatory; immutable Consumer: support optional; read-only</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>URI</td>
<td>A URI that uniquely identifies the type of the Event. If the &quot;content&quot; attribute is present, this URI determines the actual data structure used for this content, e.g., to which schema it is...</td>
<td></td>
</tr>
</tbody>
</table>
### Event Attributes Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>Event</td>
<td></td>
</tr>
<tr>
<td><strong>Type URI</strong></td>
<td><a href="http://schemas.dmtf.org/cimi/1/Event">http://schemas.dmtf.org/cimi/1/Event</a></td>
<td></td>
</tr>
<tr>
<td><strong>Attribute</strong></td>
<td><strong>Type</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>content</td>
<td>any</td>
<td>A polymorphic attribute that represents detailed event data, the type of which varies with the Event &quot;type.&quot; 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</td>
</tr>
<tr>
<td>outcome</td>
<td>string</td>
<td>A string value that characterizes the general significance of the Event. A core set is defined that may be used regardless of the Event type. For each Event type, the definition of a core outcome value maybe refined in the context of this type, provided it does not conflict with the general meaning of the outcome given below. Core outcomes are: Pending: The Event is about an action or process that is still ongoing. Unknown: The Event is about a request or action that is not known by the Provider. Status: The Event reports on the state or status of a Resource. Success: The Event reports on a successful outcome of some action or process. Warning: The Event reports on a situation that requires attention or remedial action. Failure: The Event reports on a failed outcome of some action or process. This set of core outcome values may be extended to accommodate possible outcomes of a specific Event type. In this case, the extended set of values shall apply to all Events of this type. Constraints: Provider: support optional; immutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>severity</td>
<td>string</td>
<td>A value indicating the Event severity. Possible values are: critical high medium low The meaning of the severity level may vary depending on the Event &quot;type.&quot; If such an attribute is not relevant to a particular type of Event, it should be omitted. Constraints: Provider: support optional; immutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>contact</td>
<td>string</td>
<td>A reference to a contact point or processing point to handle the Event. 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 Event &quot;type.&quot; This attribute is mutable as it may be determined after Event creation by the Provider. Constraints: Provider: support optional; immutable Consumer: support optional; read-only</td>
</tr>
</tbody>
</table>

NOTE There exists a legacy of several Event models that have been standardized or designed for various domains relevant to IT. The objective in CIMI is not to elect one particular Event model, but to select as top-level Event attributes the most immediately relevant data useful for Event processing in a Cloud environment. Additional Event data may still be represented in the variable content attribute that allows for mapping other Event models into a CIMI Event.
When implementing or using Event, Providers and Consumers shall adhere to the syntax and semantics of its attributes as described in the above table. Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

**JSON media type:** application/json

**JSON serialization:**

```json
{
   "resourceURI": "http://schemas.dmtf.org/cimi/1/Event",
   "id": string,
   "name": string, ?
   "description": string, ?
   "created": string, ?
   "updated": string, ?
   "properties": [ { string: string, + }, ?
   "timestamp": string,
   "type": string,
   "content": any, ?
   "outcome": string, ?
   "severity": string, ?
   "contact": string, ?
   ...
}
```

**XML media type:** application/xml

**XML serialization:**

```xml
<Event xmlns="http://schemas.dmtf.org/cimi/1">
   <id> xs:anyURI </id>
   <name> xs:string </name> ?
   <description> xs:string </description> ?
   <created> xs:dateTime </created> ?
   <updated> xs:dateTime </updated> ?
   <property key="xs:string"> xs:string </property> *
   <timestamp> xs:dateTime </timestamp>
   <type> xs:string </type>
   <content> xs:any* </content> ?
   <outcome> xs:string </outcome> ?
   <severity> xs:string </severity> ?
   <contact> xs:string </contact> ?
   <xs:any>*
</Event>
```

Table 57 describes the "type" URIs that are defined or acknowledged by this specification. Additional types may be added by a Provider, for example to characterize external events mapped into CIMI.
Events. It is recommended that these URIs be dereferencable such that Consumers can discover a more detailed description of the type. Event types defined by this specification share the same base URI: http://schemas.dmtf.org/cimi/1/event/. For brevity, if the “Event Type” column in the table only shows a relative URI (e.g., state) it shall be appended to the end of this base URI.

Table 57 – type URIs

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>state</td>
<td>Events of this type report state information about CIMI run-time resources such as instances of Machines, Systems, Networks, and Volumes. This information includes reports on any change in the “state” of these Resources. The content element associated with this Event type has the following structure:</td>
</tr>
</tbody>
</table>

<table>
<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 Resource about the state of which is reported. Constraints: Provider: support optional; immutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>resource</td>
<td>ref</td>
<td>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</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>state</td>
<td>string</td>
<td>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</td>
</tr>
<tr>
<td>previous</td>
<td>string</td>
<td>The previous state value, if the event reports a state change. Constraints: Provider: support optional; immutable Consumer: support optional; read-only</td>
</tr>
</tbody>
</table>
### Event Type

**alarm**

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:

<table>
<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 Resource associated with this alarm, if applicable.</td>
</tr>
<tr>
<td>resource</td>
<td>ref</td>
<td>The reference to the Resource associated with this alarm, if applicable.</td>
</tr>
<tr>
<td>restype</td>
<td>URI</td>
<td>URI denoting this Resource type associated with this alarm, if applicable</td>
</tr>
<tr>
<td>code</td>
<td>string</td>
<td>An alarm code.</td>
</tr>
<tr>
<td>detail</td>
<td>string</td>
<td>The detailed information associated with the alarm.</td>
</tr>
</tbody>
</table>

**Constraints:**

Provider: support optional; immutable
Consumer: support optional; read-only
### Event Type: model

*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.).

The *content* element associated with this event type has the following structure:

<table>
<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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong> 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong> 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).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong> 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).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong> 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong> Provider: support optional; immutable Consumer: support optional; read-only</td>
</tr>
</tbody>
</table>

### Event Type: access

*Events* of this type keep track of all requests to access some Resource of a CIMI provider.

The *content* element associated with this event type has the following structure:

<table>
<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).</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong> Provider: support mandatory; immutable Consumer: support optional; read-only</td>
</tr>
<tr>
<td>resource</td>
<td>ref</td>
<td>The reference of the primary Resource supporting the operation (for the HTTP protocol, the Resource URI or the URI associated with the operation). (Note: This reference may become invalid because the event might outlive the Resource.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong> 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</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong> Provider: support mandatory; 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Constraints:</strong> Provider: support optional; immutable Consumer: support optional; read-only</td>
</tr>
</tbody>
</table>

http://schemas.dmtf.org/cloud/audit/1.0/ Events of this type represent events that have audit significance, as defined by CADF (...). This type can be subdivided further by extending the URI path (e.g., http://schemas.dmtf.org/cloud/audit/1.0/event/security, for security audit events). The *content* element associated with this event type has the same structure as the event serialization defined in CADF[...]:

---

Version 1.1.0

DMTF Standard

213
The following pseudo-schemas describe the serialization of the "content" property for various types of events:

"state" event:

**JSON serialization:**

```json
{
  "id": string,
  ...
  "type": "http://schemas.dmtf.org/cimi/1/event/state",
  "content": {
    "resName": string,
    "resource": { "href" : string },
    "resType": string,
    "state": string,
    "previous": string ?
  }
}
```

**XML serialization:**

```xml
<Event xmlns="http://schemas.dmtf.org/cimi/1">
  ...
  <type> http://schemas.dmtf.org/cimi/1/event/state </type>
  <content>
    <resName> xs:string </resName>
    <resource href="xs:anyURI"/>
    <resType> xs:anyURI </resType>
    <state> xs:string </state>
    <previous> xs:string </previous> ?
  </content> ?
  ...
</Event>
```

"alarm" event:

**JSON serialization:**

```json
{
  "id": string,
  ...
  "type": "http://schemas.dmtf.org/cimi/1/event/alarm",
  "content": {
    "resName": string ?
    "resource": { "href" : string }, ?
    "resType": string ?
```
DSP0263  Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol

```
"code" : string,
"detail" : string ?
}
...

XML serialization:
```
<Event xmlns="http://schemas.dmtf.org/cimi/1">
  ...
  <type> http://schemas.dmtf.org/cimi/1/event/alarm </type>
  <content>
    <resname> xs:string </resname> ?
    <resource href="xs:anyURI"/> ?
    <restype> xs:anyURI </restype> ?
    <code> xs:string </code>
    <detail> xs:string </detail> ?
  </content> ?
  ...
</Event>
```

"model" event:

JSON serialization:
```
{ "id": string,
  ...
  "type": "http://schemas.dmtf.org/cimi/1/event/model",
  "content": {
    "resName": string, ?
    "resource": { "href" : string }, ?
    "restype": string, ?
    "change": string,
    "detail": string ?
  }
}
```

XML serialization:
```
<Event xmlns="http://schemas.dmtf.org/cimi/1">
  ...
  <type> http://schemas.dmtf.org/cimi/1/event/model </type>
  <content>
    <resname> xs:string </resname> ?
    <resource href="xs:anyURI"/> ?
  </content> ?
</Event>
```
"access" event:

**JSON serialization:**

```json
{
  "id": string,
  ...
  "type": "http://schemas.dmtf.org/cimi/1/event/access",
  "content": {
    "operation": string,
    "resource" : { "href" : string },
    "detail" : string, ?
    "initiator" : string ?
  }
  ...
}
```

**XML Serialization:**

```xml
<Event xmlns="http://schemas.dmtf.org/cimi/1">
  ...
  <type> http://schemas.dmtf.org/cimi/1/event/access </type>
  <content>
    <operation> xs:string </operation>
    <resource href="xs:anyURI"/>
    <detail> xs:string </detail> ?
    <initiator> xs:string </initiator> ?
  </content>
  ...
</Event>
```

### 5.17.13.1 Operations

This resource supports the Read, Update, and Delete operations.

## 6  Security considerations

There are many security mechanisms that can be used in conjunction with this specification. This specification does not mandate any particular mechanism. Providers shall provide enough information about their security mechanisms so that the Consumer can implement the necessary algorithms to successfully communicate with the Provider.
ANNEX A
(normative)

OVF support in CIMI

This annex defines how elements of an OVF descriptor are mapped to CIMI resources and their attributes. This definition allows the import of an OVF package to create multiple CIMI resources. This is done by specifying a reference to an OVF package in the import operation of a SystemCollection or SystemTemplateCollection (the Media Type at that URI shall be "application/ovf"). Refer to DSP0243 for more information about OVF.

Support for OVF import and export is optional for a Provider and it is an implementation choice as to how many of the attributes in the OVF package are exposed through CIMI resources. A Provider may support the import of OVF package for only Systems, only SystemTemplates or both. Support for the actual import and export of an OVF package is handled by a hypervisor under the management of the CIMI implementation, and thus the CIMI resources that are created reflect what the hypervisor did upon import and form a “View” into the results.

The import of an OVF package can be reflected in the creation of Templates that can be later used to create Systems, Machines and other component Resources. The import of an OVF package can also be used to directly create Systems, Machines, and other component Resources, bypassing the step of creating Templates.

Clause 5.13.4 details how to import an OVF file to create a SystemTemplate (and component Resources). The SystemTemplate thus created contains a reference to a MachineTemplate for every VirtualSystem that is defined in the OVF descriptor VirtualSystemCollection. Note that CIMI currently allows Systems of Systems, so for each VirtualSystemCollection encountered in a nested set of collections, a separate SystemTemplate is created within the parent SystemTemplate with MachineTemplates for each of the contained VirtualSystems in that VirtualSystemCollection.

The values of the attributes for the MachineTemplate are taken from the VirtualHardwareSection of the VirtualSystem description (required in OVF). If more than one VirtualHardwareSection is used for a given VirtualSystem (allowed in OVF), the result is implementation dependent, but the implementation might choose a MachineTemplate from an existing (perhaps static) set that best matches a VirtualHardwareSection. Items in the VirtualHardwareSection are mapped to CIMI MachineConfiguration properties and the corresponding MachineConfiguration Resource is created and linked to from the created MachineTemplate for that VirtualSystem.

The CIMI VolumeTemplates are created according to the DiskSection of an OVF descriptor and can be shared among more than one VirtualSystem (CIMI MachineTemplates) defined in an OVF package. In addition, a new CIMI MachineImage Resource may be created from the DiskSection if an ovf:fileRef for the virtual disk content is specified.

The CIMI NetworkTemplates are created according to the NetworkSection of an OVF descriptor along with the Connection elements in the VirtualHardwareSection elements that refer to these named networks.

Clause 5.13.2.1 details how to import an OVF file to create a System (and component Resources). The System thus created contains a reference to a Machine for every VirtualSystem that is defined in
an OVF descriptor VirtualSystemCollection. Note that CIMI currently allows Systems of
Systems, so for each VirtualSystemCollection encountered in a nested set of collections, a
separate System is created within the parent System with Machines for each of the contained
VirtualSystems in that VirtualSystemCollection.

The values of the attributes for the Machine are taken from the VirtualHardwareSection of the
VirtualSystem description (required in OVF). If more than one VirtualHardwareSection is used
for a given VirtualSystem (allowed in OVF), the result is implementation dependent. Items in the
VirtualHardwareSection are mapped to CIMI MachineConfiguration properties and the
corresponding MachineConfiguration Resource is created and linked to from the created
Machine for that VirtualSystem.

The CIMI Volumes are created according to the DiskSection of an OVF descriptor and can be shared
among more than one VirtualSystem (CIMI Machines) defined in an OVF package. In addition, a
new CIMI MachineImage Resource may be created from the DiskSection if an ovf:fileRef
attribute for the virtual disk content is specified.

The CIMI Networks are created according to the NetworkSection of an OVF descriptor along with
the Connection elements in the VirtualHardwareSection that refer to these named networks.
The XML Schema for the XML serialization of the CIMI model can be found at:

http://schemas.dmtf.org/cimi/1/DSP8009_1.0.xsd

The schema provided does not intend to reflect every single modeling constraint and requirement specified in the model. This schema is designed to apply more broadly to any model-related serialized material found in Consumer requests as well as in Provider responses, and is intended to provide a preliminary, non-exhaustive syntactic check on these. In particular, future updates of this specification may intermix new XML elements into the Resources using the current CIMI namespace to Resources. The schema that is provided is just a starting point for those who would find it useful and it might need to be modified based on specific application's needs.
ANNEX C
(informative)

Change log

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.0</td>
<td>2012-08-28</td>
<td></td>
</tr>
<tr>
<td>1.0.1</td>
<td>2012-09-12</td>
<td>Errata</td>
</tr>
<tr>
<td>1.1.0</td>
<td>2013-10-22</td>
<td>Released as DMTF Standard</td>
</tr>
</tbody>
</table>