



1  
2      Document Identifier: DSP2027  
3      Date: 2016-08-04  
4      Version: 2.0.0

5      **Cloud Infrastructure Management Interface**  
6      **(CIMI) Primer**

- 7      **Supersedes:** 1.1.0  
8      **Document Class:** Informative  
9      **Document Status:** Published  
10     **Document Language:** en-US

11 Copyright notice

12 Copyright © 2014-2016 Distributed Management Task Force, Inc. (DMTF). All rights reserved.

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

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27 withdrawn or modified after publication, and shall be indemnified and held harmless by any party  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

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

33 This document's normative language is English. Translation into other languages is permitted.

34

## CONTENTS

35

36

|  |    |
|--|----|
| Foreword .....   | 7  |
| Introduction.....  | 9  |
| 1 Common scenarios of consumer-provider interactions.....        | 11 |
| 1.1 Creating a new Machine .....                                 | 11 |
| 1.1.1 Retrieve the CEP .....                                     | 11 |
| 1.1.2 Retrieve the list of Machine Images.....                   | 11 |
| 1.1.3 Choose a Machine Image .....                               | 12 |
| 1.1.4 Retrieve the list of Machine Configurations .....          | 12 |
| 1.1.5 Choose a Machine Configuration .....                       | 13 |
| 1.1.6 Create a new Credential Resource.....                      | 14 |
| 1.1.7 Create a new Machine .....                                 | 15 |
| 1.1.8 Query new Machine .....                                    | 16 |
| 1.1.9 Start a Machine.....                                       | 16 |
| 1.1.10 Query a Machine.....                                      | 17 |
| 1.1.11 Stop a Machine .....                                      | 17 |
| 1.1.12 Update a Machine's attributes .....                       | 17 |
| 1.2 Adding a new Volume to a Machine .....                       | 18 |
| 1.2.1 Obtain the Machine URL .....                               | 18 |
| 1.2.2 Retrieve the u.....  | 18 |
| 1.2.3 Get the list of VolumeConfigurations .....                 | 18 |
| 1.2.4 Choose a Volume Configuration.....                         | 19 |
| 1.2.5 Create a new Volume .....                                  | 19 |
| 1.2.6 Retrieve the Volume information .....                      | 20 |
| 1.2.7 Retrieve the Machine's Volume collection .....             | 20 |
| 1.2.8 Connect the new Volume to a Machine .....                  | 22 |
| 1.2.9 Query the Machine's volume collection .....                | 22 |
| 1.3 Defining and using Machine Templates.....                    | 23 |
| 1.3.1 Retrieve the CEP .....                                     | 23 |
| 1.3.2 Create a new Machine Template.....                         | 23 |
| 1.3.3 Create a new Machine by using a Machine Template.....      | 24 |
| 1.4 Creating a new Machine connected to an existing Volume.....  | 24 |
| 1.4.1 Retrieve the CEP .....                                     | 24 |
| 1.4.2 Get the list of Volumes.....                               | 25 |
| 1.4.3 Choose a Volume .....                                      | 25 |
| 1.4.4 Create a new Machine .....                                 | 25 |
| 1.4.5 Query new Machine .....                                    | 26 |
| 1.5 Defining and using System Templates .....                    | 27 |
| 1.5.1 Retrieve the CEP .....                                     | 27 |
| 1.5.2 Create a new System Template .....                         | 27 |
| 1.5.3 Create a new System by using a System Template.....        | 29 |
| 1.5.4 Query the new System .....                                 | 30 |
| 1.6 Editing System Templates .....                               | 30 |
| 1.6.1 Edit an existing System Template .....                     | 30 |
| 1.6.2 Create a new System using a System Template.....           | 33 |
| 1.6.3 Query the new System .....                                 | 33 |
| 1.7 Provider responses and return values for special cases ..... | 34 |
| 1.7.1 Unrecognized attributes in request filters .....           | 34 |
| 1.7.2 Unreasonable requests.....                                 | 34 |
| 2 Resources modeling aspects .....                               | 34 |
| 2.1 Conventions .....  | 34 |
| 2.2 1-to-N relationships using arrays and collections .....      | 35 |

|     |        |   |     |
|-----|--------|---|-----|
| 87  | 2.3    | Template instantiation .....                              | 37  |
| 88  | 2.4    | About the nature of dependencies between resources .....  | 39  |
| 89  | 2.5    | Metadata about a resource .....                           | 39  |
| 90  | 3      | Groups of related Resources and their serialization ..... | 40  |
| 91  | 3.1    | The Cloud Entry Point (CEP) .....                         | 40  |
| 92  | 3.2    | Machine Resources group .....                             | 41  |
| 93  | 3.2.1  | Machine Resources and their relationships .....           | 41  |
| 94  | 3.2.2  | Machine serialization .....                               | 43  |
| 95  | 3.2.3  | DiskCollection serialization .....                        | 45  |
| 96  | 3.2.4  | MachineCollection serialization .....                     | 46  |
| 97  | 3.2.5  | MachineTemplate serialization .....                       | 48  |
| 98  | 3.2.6  | MachineTemplateCollection serialization .....             | 51  |
| 99  | 3.2.7  | MachineConfiguration serialization .....                  | 52  |
| 100 | 3.2.8  | MachineConfigurationCollection serialization .....        | 53  |
| 101 | 3.2.9  | MachineImage serialization .....                          | 54  |
| 102 | 3.2.10 | MachineImageCollection serialization .....                | 55  |
| 103 | 3.3    | Credential Resources group .....                          | 56  |
| 104 | 3.3.1  | Credential serialization .....                            | 56  |
| 105 | 3.3.2  | CredentialCollection serialization .....                  | 57  |
| 106 | 3.3.3  | CredentialTemplate serialization .....                    | 58  |
| 107 | 3.3.4  | CredentialTemplateCollection serialization .....          | 59  |
| 108 | 3.4    | Volume Resources group .....                              | 61  |
| 109 | 3.4.1  | Volume Resources group and their relationships .....      | 61  |
| 110 | 3.4.2  | Volume serialization .....                                | 61  |
| 111 | 3.4.3  | VolumeCollection serialization .....                      | 63  |
| 112 | 3.4.4  | locatedVolumeCollection serialization .....               | 64  |
| 113 | 3.4.5  | VolumeTemplate serialization .....                        | 65  |
| 114 | 3.4.6  | VolumeTemplateCollection serialization .....              | 67  |
| 115 | 3.4.7  | VolumeConfiguration serialization .....                   | 68  |
| 116 | 3.4.8  | VolumeConfigurationCollection serialization .....         | 69  |
| 117 | 3.4.9  | VolumelImage serialization .....                          | 70  |
| 118 | 3.4.10 | VolumelImageCollection serialization .....                | 71  |
| 119 | 3.5    | System Resources group .....                              | 72  |
| 120 | 3.5.1  | System Resources and their relationships .....            | 72  |
| 121 | 3.5.2  | System serialization .....                                | 74  |
| 122 | 3.5.3  | SystemCollection serialization .....                      | 76  |
| 123 | 3.5.4  | SystemService serialization .....                         | 77  |
| 124 | 3.5.5  | SystemTemplate serialization .....                        | 79  |
| 125 | 3.5.6  | SystemTemplateCollection serialization .....              | 81  |
| 126 | 3.6    | Network Resources group .....                             | 83  |
| 127 | 3.6.1  | Network Resources and their relationships .....           | 83  |
| 128 | 3.6.2  | Network serialization .....                               | 85  |
| 129 | 3.6.3  | NetworkCollection serialization .....                     | 87  |
| 130 | 3.6.4  | NetworkTemplate serialization .....                       | 88  |
| 131 | 3.6.5  | NetworkTemplateCollection serialization .....             | 90  |
| 132 | 3.6.6  | NetworkService serialization .....                        | 91  |
| 133 | 3.6.7  | NetworkServiceCollection serialization .....              | 92  |
| 134 | 3.6.8  | NetworkServiceTemplate serialization .....                | 94  |
| 135 | 3.6.9  | NetworkServiceTemplateCollection serialization .....      | 96  |
| 136 | 3.6.10 | ProtocolSegment serialization .....                       | 97  |
| 137 | 3.6.11 | ProtocolSegmentCollection serialization .....             | 98  |
| 138 | 3.6.12 | ProtocolSegmentTemplate serialization .....               | 99  |
| 139 | 3.6.13 | ProtocolSegmentTemplateCollection serialization .....     | 101 |
| 140 | 3.6.14 | ProtocolEndpoint serialization .....                      | 102 |
| 141 | 3.6.15 | ProtocolEndpointCollection serialization .....            | 104 |
| 142 | 3.6.16 | ProtocolEndpointTemplate serialization .....              | 105 |

|     |                       |  |     |
|-----|-----------------------|--|-----|
| 143 | 3.6.17                | ProtocolEndpointTemplateCollection serialization ..... | 106 |
| 144 | 3.6.18                | NetworkInterface serialization.....                    | 107 |
| 145 | 3.6.19                | NetworkInterfaceCollection serialization.....          | 109 |
| 146 | 3.6.20                | NetworkInterfaceTemplate serialization.....            | 110 |
| 147 | 3.6.21                | NetworkInterfaceTemplateCollection serialization.....  | 111 |
| 148 | 3.7                   | Monitoring Resources group.....                        | 112 |
| 149 | 3.7.1                 | Monitoring Resources and their relationships .....     | 112 |
| 150 | 3.7.2                 | Job serialization .....                                | 113 |
| 151 | 3.7.3                 | JobCollection serialization .....                      | 115 |
| 152 | 3.7.4                 | Meter serialization.....                               | 116 |
| 153 | 3.7.5                 | MeterCollection serialization.....                     | 117 |
| 154 | 3.7.6                 | SampleCollection serialization.....                    | 118 |
| 155 | 3.7.7                 | MeterTemplate serialization.....                       | 120 |
| 156 | 3.7.8                 | MeterTemplateCollection serialization.....             | 121 |
| 157 | 3.7.9                 | MeterConfiguration serialization .....                 | 122 |
| 158 | 3.7.10                | MeterConfigurationCollection serialization .....       | 123 |
| 159 | 3.7.11                | Event serialization.....                               | 124 |
| 160 | 3.7.12                | EventLog serialization.....                            | 128 |
| 161 | 3.7.13                | EventCollection serialization .....                    | 130 |
| 162 | 3.7.14                | EventLogCollection serialization.....                  | 131 |
| 163 | 3.7.15                | EventLogTemplate serialization.....                    | 132 |
| 164 | 3.7.16                | EventLogTemplateCollection serialization.....          | 133 |
| 165 | 3.8                   | Other Examples of serialized collections .....         | 134 |
| 166 | 3.8.1                 | Machine Collection.....                                | 134 |
| 167 | 3.8.2                 | Volume Collection in a Machine .....                   | 135 |
| 168 | ANNEX A (informative) | Change log .....                                       | 138 |

**169 Figures**

|     |   |     |
|-----|---|-----|
| 170 | Figure 1 - Graphical conventions .....                                      | 35  |
| 171 | Figure 2 – Array-based 1-n Resource relationship from A to Bs .....         | 36  |
| 172 | Figure 3 – Collection-based 1-n Resource relationship from A to Bs .....    | 37  |
| 173 | Figure 4 – Template instantiation mechanism .....                           | 38  |
| 174 | Figure 5 – Cloud Entry Point resources .....                                | 41  |
| 175 | Figure 6 – Machine Resources group .....                                    | 42  |
| 176 | Figure 7 – Volume Resources group .....                                     | 61  |
| 177 | Figure 8 – System Resources group .....                                     | 73  |
| 178 | Figure 9 – SystemTemplate Resources group .....                             | 74  |
| 179 | Figure 10 – Network Resources group .....                                   | 83  |
| 180 | Figure 11 – Network Resources group: Resource structure relationships ..... | 84  |
| 181 | Figure 12 – Network Resources group: Network connectivity perspective ..... | 85  |
| 182 | Figure 13 – Monitoring Resources group.....                                 | 113 |
| 183 |   |     |

184

## Foreword

185 The *Cloud Infrastructure Management Interface (CIMI) Primer* (DSP2027) was prepared by the Cloud  
186 Management Working Group of the DMTF. This document contains scenarios that describe common uses  
187 of the CIMI protocol.

188 This specification has been developed as a result of joint work with many individuals and teams,  
189 including:

190 Dies Köper Fujitsu (Editor)  
191 Larry Lamers VMware (Editor)

192 **Contributors:**

|                              |                                |
|------------------------------|--------------------------------|
| 193 Hemal Shah               | Broadcom                       |
| 194 John Crandall            | Brocade Communications Systems |
| 195 Paul Lipton              | CA Technologies                |
| 196 Efraim Moscovich         | CA Technologies                |
| 197 Mark Carlson             | DMTF Fellow                    |
| 198 Marvin Waschke           | DMTF Fellow                    |
| 199 Steven Neely             | Cisco                          |
| 200 Shishir Pardikar         | Citrix Systems Inc.            |
| 201 George Ericson           | EMC                            |
| 202 Norbert Floeren          | Ericsson AB                    |
| 203 Arturo Martin de Nicolas | Ericsson AB                    |
| 204 Ruby Krishnaswamy        | France Telecom Group           |
| 205 Jacques Durand           | Fujitsu                        |
| 206 Kazunori Iwasa           | Fujitsu                        |
| 207 Jesus Molina             | Fujitsu                        |
| 208 Tom Rutt                 | Fujitsu                        |
| 209 Maarten Wiggers          | Fujitsu                        |
| 210 Bryan Murray             | Hewlett                        |
| 211 Derek Coleman            | Hewlett Packard Company        |
| 212 Robert Freund            | Hitachi, Ltd.                  |
| 213 Fred Maciel              | Hitachi, Ltd.                  |
| 214 Eric Wells               | Hitachi, Ltd.                  |
| 215 Zhexuan Song             | Huawei                         |
| 216 Jeff Wheeler             | Huawei                         |
| 217 Aaron Zhang              | Huawei                         |
| 218 HengLiang Zhang          | Huawei                         |
| 219 Doug Davis               | IBM                            |
| 220 Mark Johnson             | IBM                            |
| 221 Andreas Maier            | IBM                            |
| 222 Matthew Rutkowski        | IBM                            |
| 223 Keith Bankston           | Microsoft Corporation          |
| 224 Nathan Burkhardt         | Microsoft Corporation          |
| 225 Josh Cohen               | Microsoft Corporation          |
| 226 Yigal Edery              | Microsoft Corporation          |
| 227 Colleen Evans            | Microsoft Corporation          |
| 228 Krishnan Gopalan         | Microsoft Corporation          |
| 229 John Parchem             | Microsoft Corporation          |
| 230 Nihar Shah               | Microsoft Corporation          |
| 231 Hiroshi Dempo            | NEC Corporation                |
| 232 James Livingston         | NEC Corporation                |
| 233 Ryuichi Ogawa            | NEC Corporation                |
| 234 Steve Carter             | Novell                         |
| 235 Ashok Malhotra           | Oracle                         |

|     |                        |                       |
|-----|------------------------|-----------------------|
| 236 | Gilbert Pilz           | Oracle                |
| 237 | Jack Yu                | Oracle                |
| 238 | Marios Andreou         | Red Hat               |
| 239 | David Lutterkort       | Red Hat               |
| 240 | Steve Winkler          | SAP AG                |
| 241 | Enrico Ronco           | Telecom Italia        |
| 242 | Federico Rossini       | Telecom Italia        |
| 243 | Fernando de la Iglesia | Telefónica            |
| 244 | Fermín Galán           | Telefónica            |
| 245 | Miguel Peñalvo         | Telefónica            |
| 246 | Alvaro Polo            | Telefónica            |
| 247 | Alan Sill              | Texas Tech University |
| 248 | Vince Lubsey           | Virtustream Inc.      |
| 249 | Winston Bumpus         | VMware Inc.           |
| 250 | Jim Davis              | WBEM Solutions        |
| 251 | Ghazanfar Ali          | ZTE Corporation       |
| 252 | Junsheng Chu           | ZTE Corporation       |
| 253 | Bhumip Khasnabish      | ZTE Corporation       |

254

## Introduction

255 For the sake of simplicity, in each of the following scenarios, the Cloud provider only supports the  
256 minimum features needed to demonstrate the features highlighted by each scenario. Therefore, the  
257 results of the query to the Cloud Entry Point (CEP) to retrieve the list of supported features are  
258 customized for each scenario. Additionally, the URI of the Cloud Entry Point is assumed to be  
259 <http://example.com/CEP> and all resources are assumed to be available in the example.com domain and  
260 accessible with the same protocol (HTTP). In the HTTP request-response examples in this document, the  
261 creation of the connection and HTTP headers not mandated by the CIMI specification are omitted for  
262 brevity.

263



## 265      Cloud Infrastructure Management Interface (CIMI) Primer

### 266    1 Common scenarios of consumer-provider interactions

#### 267    1.1 Creating a new Machine

268 This scenario creates a new Machine. The new Machine's configuration is based on existing  
 269 configurations and images offered by the provider. However, a new Credential resource (userid and  
 270 password) is created.

##### 271    1.1.1 Retrieve the CEP

272 The CEP provides the links to the set of resources that are available in this Cloud. You retrieve the CEP  
 273 to discover the URL to each collection:

```
274 GET /CEP HTTP/1.1
275
276 HTTP/1.1 200 OK
277 Content-Type: application/json
278
279 { "resourceURI": "http://schemas.dmtf.org/cimi/2/CloudEntryPoint",
280   "id": "http://example.com/CEP",
281   "baseURI": "http://example.com/",
282   "resourceMetadata": { "href": "http://example.com/resourceMetadata" },
283   "machines": { "href": "http://example.com/machines" },
284   "machineConfigs": { "href": "http://example.com/machineConfigs" },
285   "machineImages": { "href": "http://example.com/machineImages" },
286   "credentials": { "href": "http://example.com/credentials" }
287 }
```

##### 288    1.1.2 Retrieve the list of Machine Images

289 Before you can create a new Machine, first decide what kind of operating system or software you want to  
 290 have preinstalled. The Machine Images collection is the set of Machine Images that this Cloud offers -  
 291 note that some Machine Images may be predefined by the Cloud while some may be user created. The  
 292 URI path comes from the data returned in the query to CEP for the `machineImages` key.

293 To retrieve the list of Machine Images, use the following syntax:

```
294 GET /machineImages HTTP/1.1
295
296 HTTP/1.1 200 OK
297 Content-Type: application/json
298
299 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImageCollection",
300   "id": "http://example.com/machineImages",
301   "updated": "2015-01-25T12:00:00Z",
302   "parent": "http://example.com/CEP",
303   "count": 3,
304   "machineImages": [
305     { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImage",
306       "id": "http://example.com/images/WinXP-SP2",
307       "name": "WinXP SP2",
308       "description": "Windows XP with Service Pack 2",
309       "created": "2015-01-15T12:00:00Z",
310       "updated": "2015-01-15T12:00:00Z",
311       "parent": "http://example.com/machineImages",
312       "imageLocation": "http://example.com/data/8934322"
313     },
314   ],
315 }
```

```

314
315     { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImage",
316       "id": "http://example.com/images/Win7",
317       "name": "Windows 7",
318       "description": "Windows 7",
319       "created": "2015-01-10T12:00:00Z",
320       "updated": "2015-01-10T12:00:00Z",
321       "parent": "http://example.com/machineImages",
322       "imageLocation": "http://example.com/data/8934344"
323     },
324
325     { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImage",
326       "id": "http://example.com/images/Linux-SUSE",
327       "name": "Linux SUSE",
328       "description": "Linux SUSE",
329       "created": "2015-01-09T12:00:00Z",
330       "updated": "2015-01-09T12:00:00Z",
331       "parent": "http://example.com/machineImages",
332       "imageLocation": "http://example.com/data/8934311"
333     }
334   ]
335 }
```

### 336 1.1.3 Choose a Machine Image

337 Next examine each Machine Image to find one that meets your needs. The first one is acceptable, so it  
338 will be used later.

339 It is worth noting that if you knew you wanted to use the first item in the list and only wanted to see that  
340 one resource in the previous query, the following syntax could have been used instead:

```

341 GET /machineImages?$first=1&$last=1 HTTP/1.1
342
343 HTTP/1.1 200 OK
344 Content-Type: application/json
345
346 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImageCollection",
347   "id": "http://example.com/machineImages",
348   "updated": "2015-01-25T12:00:00Z",
349   "parent": "http://example.com/CEP",
350   "count": 3,
351   "machineImages": [
352     { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImage",
353       "id": "http://example.com/images/WinXP-SP2",
354       "name": "WinXP SP2",
355       "description": "Windows XP with Service Pack 2",
356       "created": "2015-01-01T12:00:00Z",
357       "updated": "2015-01-01T12:00:00Z",
358       "parent": "http://example.com/machineImages",
359       "imageLocation": "http://example.com/data/8934322"
360     }
361   ]
362 }
```

363 Note that you do not need to specify \$first=1 in this case because "1" is its default value. The first  
364 machineImage is returned.

### 365 1.1.4 Retrieve the list of Machine Configurations

366 Next you decide onto what kind of virtual hardware you want to install your Machine Image. As with  
367 determining the kind of Machine Image you want, first ask for the list of available Machine Configurations:

```
368 GET /machineConfigs HTTP/1.1
```

```

369
370 HTTP/1.1 200 OK
371 Content-Type: application/json
372
373 { "resourceURI":
374     "http://schemas.dmtf.org/cimi/2/MachineConfigurationCollection",
375     "id": "http://example.com/machineConfigs",
376     "updated": "2015-01-25T12:00:00Z",
377     "parent": "http://example.com/CEP",
378     "count": 3,
379     "machineConfigurations": [
380         { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineConfiguration",
381             "id": "http://example.com/machineConfigs/tiny",
382             "name": "tiny",
383             "description": "a teenie tiny one",
384             "created": "2015-01-01T12:00:00Z",
385             "updated": "2015-01-01T12:00:00Z",
386             "parent": "http://example.com/machineConfigs",
387             "cpu": 1,
388             "memory": 4000000,
389             "disks" : [
390                 { "capacity": 50000000 }
391             ]
392         },
393
394         { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineConfiguration",
395             "id": "http://example.com/machineConfigs/small",
396             "name": "small",
397             "description": "a small sized one",
398             "created": "2015-01-01T12:00:00Z",
399             "updated": "2015-01-01T12:00:00Z",
400             "parent": "http://example.com/machineConfigs",
401             "cpu": 1,
402             "memory": 8000000,
403             "disks" : [
404                 { "capacity": 500000000 }
405             ]
406         },
407
408         { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineConfiguration",
409             "id": "http://example.com/machineConfigs/medium",
410             "name": "medium",
411             "description": "a medium one",
412             "created": "2015-01-25T12:00:00Z",
413             "updated": "2015-01-25T12:00:00Z",
414             "parent": "http://example.com/machineConfigs",
415             "cpu": 1,
416             "memory": 16000000,
417             "disks" : [
418                 { "capacity": 1000000000 },
419                 { "capacity": 1000000000 }
420             ]
421         }
422     ]
423 }

```

#### 424 1.1.5 Choose a Machine Configuration

425 Next examine the returned list and select a Machine Configuration that suits your needs. The first one is  
426 acceptable, so it will be used later. It is identified by the id  
427 "<http://example.com/machineConfigs/tiny>".

**428 1.1.6 Create a new Credential Resource**

429 You want to use your own `userName` and `password` attributes for this new Machine, so you need to  
430 create a new Credential resource. This process is done by using the POST operation, but first you need  
431 to retrieve the Credential collection so that you know where to POST a new Credential resource. To  
432 retrieve the Credential resource:

```
433 GET /credentials HTTP/1.1
```

434 The response is:

```
435 HTTP/1.1 200 OK
436 Content-Type: application/json
437
438 { "resourceURI": "http://schemas.dmtf.org/cimi/2/CredentialCollection",
439   "id": "http://example.com/credentials",
440   "operations": [ { "rel": "add", "href": "http://example.com/credentials" } ]
441 }
```

442

443 Notice at this point that there are no Credential resources in the environment. Before you can create a  
444 new Credential resource, you must first discover this Cloud provider's extension attributes for the  
445 Credential resource. By default the CIMI specification does not define how the initial user of a new  
446 Machine is specified; rather it is left open for each Cloud provider to determine how this information  
447 should be provided. Clients can discover this information by querying the Credential resource metadata  
448 resource. To examine this resource, first look through the `ResourceMetadata` collection for this  
449 provider's description of the Credential's resource. Start by retrieving the `ResourceMetadata` collection  
450 from the URI referenced in the CEP:

```
451 GET /resourceMetadata HTTP/1.1
```

452

453

```
454 HTTP/1.1 200 OK
455 Content-Type: application/json
456
457 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ResourceMetadataCollection",
458   "id": "http://example.com/resourceMetadata",
459   "updated": "2015-01-25T12:00:00Z",
460   "parent": "http://example.com/CEP",
461   "count": 1,
462   "resourceMetadatas": [
463     { "resourceURI": "http://schemas.dmtf.org/cimi/2/ResourceMetadata",
464       "id": "http://example.com/resources/Credential",
465       "typeURI": "http://schemas.dmtf.org/cimi/2/Credential",
466       "name": "Credential",
467       "created": "2015-01-01T12:00:00Z",
468       "updated": "2015-01-01T12:00:00Z",
469       "parent": "http://example.com/resourceMetadata",
470       "attributes": [
471         { "name": "userName", "namespace": "http://example.com",
472           "type": "string", "required": "true" },
473         { "name": "password", "namespace": "http://example.com",
474           "type": "string", "required": "true" }
475       ]
476     }
477   ]
478 }
```

478 Now iterate over the list of `resourceMetadata` entries in the collection for the one whose "typeURI" is  
479 "`http://schemas.dmtf.org/cimi/2/Credential`". After you find it, you can now examine the extensions this  
480 provider has added to the Credential resource. The above indicates that the Credential resource has  
481 been extended and must include two attributes called "userName" and "password". Both are of type  
482 "string".

483 Now create a new Credential resource by using the POST operation:

```
484 POST /credentials HTTP/1.1
485 Content-Type: application/json
486
487 { "resourceURI": "http://schemas.dmtf.org/cimi/2/CredentialCreate",
488   "name": "Default",
489   "description": "My Default User",
490   "credentialTemplate": {
491     "userName": "JoeSmith",
492     "password": "letmein"
493   }
494 }
```

```
495
496 HTTP/1.1 201 Created
497 Location: http://example.com/creds/12345
```

498 NOTE While the "userID" and "password" attributes were discovered via the Credential ResourceMetadata, the  
 499 "name" and "description" attributes are part of the common set of attributes available on all resources. In a future  
 500 scenario it is shown how the client knew that "userID" and "password" were the proper attribute names for this image  
 501 type and Cloud provider.

### 502 1.1.7 Create a new Machine

503 Retrieve the Machines collection so that you know to where to POST a new Machine:

```
504 GET /machines HTTP/1.1
505
506 HTTP/1.1 200 OK
507 Content-Type: application/json
508
509 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineCollection",
510   "id": "http://example.com/machines",
511   "updated": "2015-01-25T12:00:00Z",
512   "parent": "http://example.com/CEP",
513   "count": 0,
514   "operations": [ { "rel": "add", "href": "http://example.com/machines" } ]
515 }
```

516 If you only want to know the available operations, issue the following command.

```
517 GET /machines?$select=operations HTTP/1.1
518
519 HTTP/1.1 200 OK
520 Content-Type: application/json
521
522 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineCollection",
523   "operations": [ { "rel": "add", "href": "http://example.com/machines" } ]
524 }
```

525 Now create a new one:

```
526 POST /machines HTTP/1.1
527 Content-Type: application/json
528
529 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineCreate",
530   "name": "myMachine1",
531   "description": "My very first machine",
532   "machineTemplate": {
533     "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
534     "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
535     "credential": { "href": "http://example.com/creds/12345" }
536   }
537 }
```

539 HTTP/1.1 201 Created  
 540 Location: http://example.com/machines/843752

541 The response returns a unique machine reference "http://example.com/machines/843752" that is used in  
 542 the following subclause.

### 543 1.1.8 Query new Machine

544 Retrieve the Machine to get the full representation of the new Machine:

545 GET /machines/843752 HTTP/1.1

546  
 547 HTTP/1.1 200 OK  
 548 Content-Type: application/json

```
549
550 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Machine",
551   "id": "http://example.com/machines/843752",
552   "name": "myMachine1",
553   "description": "My very first machine",
554   "created": "2015-08-15T12:15:00Z",
555   "updated": "2015-08-15T12:15:00Z",
556   "parent": "http://example.com/machines"
557   "state": "STOPPED",
558   "cpu": 1,
559   "memory": 4000000,
560   "disks" : { "href": "http://example.com/machines/843752/disks" },
561   "interfaces": { "href": "http://example.com/machines/843752/NIs" },
562   "operations": [
563     { "rel": "edit", "href": "http://example.com/machines/843752" },
564     { "rel": "delete", "href": "http://example.com/machines/843752" },
565     { "rel": "http://schemas.dmtf.org/cimi/2/action/start",
566       "href": "http://example.com/machines/843752" }
567   ]
568 }
```

569 Notes:

- 570 • The "state" attribute on the Machine is "STOPPED" because that is the initial state of a new  
 571 machine.
- 572 • The parent attribute refers here to the machines CEP collection, as this Machine was created  
 573 by directly using the CEP machines collection as a factory. If a Machine was created as part of  
 574 the creation of a System, the parent attribute would refer to the machines collection inside the  
 575 System.
- 576 • In this simple example there is no resourceMetadata attribute (an attribute optional to  
 577 support).

### 578 1.1.9 Start a Machine

579 The presence of the "start" operation in the "operations" array of the Machine representation indicates not  
 580 only the URI to which to POST the "start" operation, but that you are able to do it at this time.

581 POST /machines/843752 HTTP/1.1  
 582 Content-Type: application/json  
 583  
 584 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
 585 "action": "http://schemas.dmtf.org/cimi/2/action/start"
 586 }

587  
 588 HTTP/1.1 204 No Content

### 589    1.1.10 Query a Machine

590    Query the Machine again to verify that it is started:

```
591    GET /machines/843752 HTTP/1.1
592
593    HTTP/1.1 200 OK
594    Content-Type: application/json
595
596    { "resourceURI": "http://schemas.dmtf.org/cimi/2/Machine",
597        "id": "http://example.com/machines/843752",
598        "name": "myMachine1",
599        "description": "My very first machine",
600        "created": "2015-08-15T12:15:00Z",
601        "updated": "2015-08-15T12:15:00Z",
602        "parent": "http://example.com/machines"
603        "state": "STARTED",
604        "cpu": 1,
605        "memory": 4000000,
606        "disks": { "href": "http://example.com/machines/843752/disks" },
607        "interfaces": { "href": "http://example.com/machines/843752/NIs" },
608        "operations": [
609            { "rel": "edit", "href": "http://example.com/machines/843752" },
610            { "rel": "delete", "href": "http://example.com/machines/843752" },
611            { "rel": "http://schemas.dmtf.org/cimi/2/action/stop",
612              "href": "http://example.com/machines/843752" }
613        ]
614 }
```

615    Notice the "state" attribute on the Machine is "STARTED" and that the "operations" array no longer  
 616    indicates that the "start" operation is available; but rather the "stop" operation is available now instead.

### 617    1.1.11 Stop a Machine

618    Using the "stop" operation's URL, you can now ask for the Machine to be stopped:

```
619    POST /machines/843752 HTTP/1.1
620    Content-Type: application/json
621
622    { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
623        "action": "http://schemas.dmtf.org/cimi/2/action/stop"
624 }
625
626    HTTP/1.1 204 No Content
```

### 627    1.1.12 Update a Machine's attributes

628    Using a PUT operation on the "edit" operation's URL, you can update some of the attributes of the  
 629    Machine, for example the "name" and "description":

```
630    PUT /machines/843752?$select=name,description HTTP/1.1
631    Content-Type: application/json
632
633    { "resourceURI": "http://schemas.dmtf.org/cimi/2/Machine",
634        "name": "Cool Demo #1"
635 }
636
637    HTTP/1.1 200 OK
638
639    { "name": "Cool Demo #1" }
```

640 Notice that URL of the "edit" operation has been modified to indicate which attributes are being updated;  
641 only those attributes are touched. Because the URL includes the "description" attribute but the HTTP  
642 request body does not, that attribute is erased.

## 643 1.2 Adding a new Volume to a Machine

644 This scenario creates a new Volume and connects it to an existing Machine.

### 645 1.2.1 Obtain the Machine URL

646 Machine:

```
647 http://example.com/machines/843752
```

### 648 1.2.2 Retrieve the CEP

649 The CEP provides the links to the set of resources that are available in this Cloud. Retrieve the CEP to  
650 discover the URL to each collection:

```
651 GET /CEP HTTP/1.1
652
653 HTTP/1.1 200 OK
654 Content-Type: application/json
655
656 { "resourceURI": "http://schemas.dmtf.org/cimi/2/CloudEntryPoint",
657   "id": "http://example.com/CEP",
658   "baseURI": "http://example.com/",
659   "machines": { "href": "http://example.com/machines" },
660   "machineConfigs": { "href": "http://example.com/machineConfigs" },
661   "machineImages": { "href": "http://example.com/machineImages" },
662   "credentials": { "href": "http://example.com/credentials" },
663   "volumes": { "href": "http://example.com/volumes" },
664   "volumeConfigs": { "href": "http://example.com/volumeConfigs" }
665 }
```

### 666 1.2.3 Get the list of VolumeConfigurations

667 When you create a new Volume, you need to decide what kind of Volume to create, e.g., its size, format,  
668 etc. The volumeConfigurations collection is the set of predefined Volume Configurations that this  
669 Cloud offers:

```
670 GET /volumeConfigs HTTP/1.1
671
672 HTTP/1.1 200 OK
673 Content-Type: application/json
674
675 { "resourceURI":
676   "http://schemas.dmtf.org/cimi/2/VolumeConfigurationCollection",
677   "id": "http://example.com/volumeConfigs",
678   "updated": "2015-08-25T12:00:00Z",
679   "parent": "http://example.com/CEP",
680   "count": 3,
681   "volumeConfigurations": [
682     { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeConfiguration",
683       "id": "http://example.com/volumeConfigs/small",
684       "name": "Small",
685       "description": "A pretty small one",
686       "created": "2015-08-15T12:15:00Z",
687       "updated": "2015-08-15T12:15:00Z",
688       "parent": "http://example.com/volumeConfigs",
689       "type": "http://schemas.dmtf.org/cimi/2/mapped",
690       "format": "NTFS",
```

```

691     "capacity": 60000000
692   },
693
694   { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeConfiguration",
695     "id": "http://example.com/volumeConfigs/medium",
696     "name": "Medium",
697     "description": "A medium sized one",
698     "created": "2015-08-15T12:15:00Z",
699     "updated": "2015-08-15T12:15:00Z",
700     "parent": "http://example.com/volumeConfigs",
701     "type": "http://schemas.dmtf.org/cimi/2/mapped",
702     "format": "NTFS",
703     "capacity": 500000000
704   },
705
706   { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeConfiguration",
707     "id": "http://example.com/volumeConfigs/large",
708     "name": "Large",
709     "description": "A large one",
710     "created": "2015-08-15T12:15:00Z",
711     "updated": "2015-08-15T12:15:00Z",
712     "parent": "http://example.com/volumeConfigs",
713     "type": "http://schemas.dmtf.org/cimi/2/mapped",
714     "format": "NTFS",
715     "capacity": 1000000000
716   }
717 ]
718 }
```

## 1.2.4 Choose a Volume Configuration

Next examine each Volume Configuration to find the one that meets your needs. The first one is acceptable, so it will be used later.

## 1.2.5 Create a new Volume

Retrieve the Volumes collection so that you know to where to POST a new Volume:

```

724 GET /volumes HTTP/1.1
725
726 HTTP/1.1 200 OK
727 Content-Type: application/json
728
729 { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeCollection",
730   "id": "http://example.com/volumes",
731   "updated": "2015-08-25T12:00:00Z",
732   "parent": "http://example.com/CEP",
733   "operations": [ { "rel": "add", "href": "http://example.com/volumes" } ]
734 }
```

Now create a new Volume:

```

736 POST /volumes HTTP/1.1
737 Content-Type: application/json
738
739 { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeCreate",
740   "name": "myVolume1",
741   "description": "My first new volume",
742   "volumeTemplate": {
743     "volumeConfig": { "href": "http://example.com/volumeConfigs/small" }
744   }
745 }
```

```
747 HTTP/1.1 201 Created
748 Location: http://example.com/volumes/35782
```

## 749 1.2.6 Retrieve the Volume information

750 To verify that the Volume you created and connected to the Machine is what you wanted, follow the  
 751 reference that was returned from the previous step:

```
752 GET /volumes/35782 HTTP/1.1
753
754 HTTP/1.1 200 OK
755 Content-Type: application/json
756
757 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Volume",
758   "id": "http://example.com/volumes/35782",
759   "name": "myVolume1",
760   "created": "2015-09-15T12:15:00Z",
761   "updated": "2015-09-15T12:15:00Z",
762   "parent": "http://example.com/volumes",
763   "description": "My first new volume",
764   "type": "http://schemas.dmtf.org/cimi/2/mapped",
765   "capacity": 60000000,
766   "operations": [
767     { "rel": "edit", "href": "http://example.com/volumes/35782" },
768     { "rel": "delete", "href": "http://example.com/volumes/35782" }
769   ]
770 }
```

## 771 1.2.7 Retrieve the Machine's Volume collection

772 Before you can connect this new Volume to your Machine, you first need to retrieve the Machine's  
 773 Volume collection so that you know to where to send your request. First retrieve the Machine to get the  
 774 reference to the collection (even if this collection is empty):

```
775 GET /machines/843752 HTTP/1.1
776
777 HTTP/1.1 200 OK
778 Content-Type: application/json
779
780 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Machine",
781   "id": "http://example.com/machines/843752",
782   "name": "myMachine1",
783   "description": "My very first machine",
784   "created": "2015-08-15T12:15:00Z",
785   "updated": "2015-08-15T12:15:00Z",
786   "parent": "http://example.com/machines"
787   "state": "STARTED",
788   "cpu": 1,
789   "memory": 4000000,
790   "disks" : { "href": "http://example.com/machines/843752/disks" },
791   "volumes": { "href": "http://example.com/machines/843752/volumes" },
792   "interfaces": { "href": "http://example.com/machines/843752/NIs" },
793   "operations": [
794     { "rel": "edit", "href": "http://example.com/machines/843752" },
795     { "rel": "delete", "href": "http://example.com/machines/843752" },
796     { "rel": "http://schemas.dmtf.org/cimi/2/action/stop",
797       "href": "http://example.com/machines/843752" }
798   ]
799 }
```

800 Note that in the previous scenario, the "volumes" attribute was not present due to the limited scope of that  
 801 scenario; however, now the "volumes" attribute appears because the scenario (and features of our  
 802 sample provider) are expanded to include support for Volumes.

803 Now retrieve the Volume collection:

```
804 GET /machines/843752/volumes HTTP/1.1
```

```
805
806 HTTP/1.1 200 OK
807 Content-Type: application/json
808
809 { "resourceURI": "http://schemas.dmtf.org/cimi/2/locatedVolumeCollection",
810   "id": "http://example.com/machines/843752/volumes",
811   "updated": "2015-08-15T12:00:00Z",
812   "parent": "http://example.com/machines/843752",
813   "operations": [
814     { "rel": "add", "href": "http://example.com/machines/843752/volumes" },
815     { "rel": "insert", "href": "http://example.com/machines/843752/volumes" },
816     { "rel": "remove", "href": "http://example.com/machines/843752/volumes" }
817   ]
818 }
```

819 Note that there are no Volumes currently connected to this Machine.

820 Alternatively, as an optimization, this collection could have been retrieved at the same time as the original  
821 Machine by using the \$expand query parameter:

```
822 GET /machines/843752?$expand=volumes HTTP/1.1
```

```
823
824 HTTP/1.1 200 OK
825 Content-Type: application/json
826
827 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Machine",
828   "id": "http://example.com/machines/843752",
829   "name": "myMachine1",
830   "description": "My very first machine",
831   "created": "2015-08-15T12:15:00Z",
832   "updated": "2015-08-15T12:15:00Z",
833   "parent": "http://example.com/machines",
834   "state": "STARTED",
835   "cpu": 1,
836   "memory": 4000000,
837   "disks": { "href": "http://example.com/machines/843752/disks" },
838   "volumes": {
839     "href": "http://example.com/machines/843752/volumes",
840     "resourceURI": "http://schemas.dmtf.org/cimi/2/locatedVolumeCollection",
841     "id": "http://example.com/machines/843752/volumes",
842     "updated": "2015-08-15T12:00:00Z",
843     "parent": "http://example.com/machines/843752",
844     "count": 0,
845     "operations": [
846       { "rel": "add", "href": "http://example.com/machines/843752/volumes" },
847       { "rel": "insert", "href": "http://example.com/machines/843752/volumes" },
848       { "rel": "remove", "href": "http://example.com/machines/843752/volumes" }
849     ],
850   },
851   "interfaces": { "href": "http://example.com/machines/843752/NIs" },
852   "operations": [
853     { "rel": "edit", "href": "http://example.com/machines/843752" },
854     { "rel": "delete", "href": "http://example.com/machines/843752" },
855     { "rel": "http://schemas.dmtf.org/cimi/2/action/stop",
856       "href": "http://example.com/machines/843752" }
857   ]
858 }
```

## 859    1.2.8 Connect the new Volume to a Machine

860 You connect the Volume to the Machine by using the "insert" operation on the Volume collection and pass  
 861 in the reference to the new Volume resource:

```
862 POST http://example.com/machines/843752/volumes HTTP/1.1
863 Content-Type: application/json
864
865 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Action",
866   "action": "http://schemas.dmtf.org/cimi/2/action/connectvolume",
867   "initialLocation": "V",
868   "credentials": { "href": "http://example.com/credentials/456" },
869   "volume": { "href": "http://example.com/volumes/35782" }
870 }
```

871

872

```
873 HTTP/1.1 200 OK
874 Content-Type: application/json
875
876 {
877 ...here, either:
878 a serialization of the updated collection at:
879 http://example.com/machines/843752/volumes
880 or
881 (if the Job Resource is supported) a pointer to a job:
882 CIMI-Job-URI: <uri-to-Job>
883 That represents the execution of this connection.
884 ...
885
886 }
```

## 887    1.2.9 Query the Machine's volume collection

888 Retrieve the Machine's volume collection to get the complete list of Volumes and use the list to verify that  
 889 the update was successful:

```
890 GET /machines/843752/volumes HTTP/1.1
891
892 HTTP/1.1 200 OK
893 Content-Type: application/json
894
895 { "resourceURI": "http://schemas.dmtf.org/cimi/2/locatedVolumeCollection",
896   "id": "http://example.com/machines/843752/volumes",
897   "updated": "2015-08-35T12:15:00Z",
898   "parent": http://example.com/machines/843752,
899   "count": 1,
900   "locatedVolumes": [
901     { "resourceURI": "http://schemas.dmtf.org/cimi/2/locatedVolume",
902       "id": "http://example.com/volumes/35782",
903       "created": "2015-09-15T12:15:00Z",
904       "updated": "2015-09-15T12:15:00Z",
905       "parent": "http://example.com/volumes",
906       "initialLocation": "V",
907       "volume": { "href": "http://example.com/volumes/35782" },
908       "operations": [
909         { "rel": "edit", "href": "http://example.com/volumes/35782" },
910         { "rel": "delete", "href": "http://example.com/volumes/35782" }
911       ],
912     },
913     "operations": [
914       { "rel": "add", "href": "http://example.com/machines/843752/volumes" }
915     ]
916   }
```

### 917 1.3 Defining and using Machine Templates

918 This scenario creates a new Machine Template that is used to create a new Machine. Machine  
919 Templates are convenience resources that allow for well-defined descriptions (configuration, image, etc.)  
920 of a Machine to be persisted such that it can be reused later. This feature is particularly useful when the  
921 user of the new Machine may not be technically savvy enough to know all of the details necessary to  
922 create the Machine. Commonly, Machine Templates are created for demos, or complex configurations,  
923 where a particular Machine Image must be used on a particular Machine Configuration. Machine  
924 Templates allow this information to be persisted and easily reused.

925 For convenience, reuse the configuration information already obtained in the previous scenarios.

#### 926 1.3.1 Retrieve the CEP

927 The CEP provides the links to the set of resources that are available in this Cloud. Retrieve the CEP to  
928 discover the URL to each collection:

```
929 GET /CEP HTTP/1.1
```

```
930
```

```
931 HTTP/1.1 200 OK
932 Content-Type: application/json
933
934 { "resourceURI": "http://schemas.dmtf.org/cimi/2/CloudEntryPoint",
935   "id": "http://example.com/CEP",
936   "baseURI": "http://example.com/",
937   "machines": { "href": "http://example.com/machines" },
938   "machineTemplates": { "href": "http://example.com/machineTemplates" },
939   "machineConfigs": { "href": "http://example.com/machineConfigs" },
940   "machineImages": { "href": "http://example.com/machineImages" },
941   "credentials": { "href": "http://example.com/credentials" }
942 }
```

#### 943 1.3.2 Create a new Machine Template

944 From the previous scenarios, you already have the MachineConfiguration, MachineImage, and Credential  
945 resources that are reused for this MachineTemplate:

946 MachineConfiguration:

```
947 http://example.com/machineConfigs/tiny
```

948 MachineImage:

```
949 http://example.com/images/WinXP-SP2
```

950 Credential:

```
951 http://example.com/creds/12345
```

952 Before you can create the new MachineTemplate, you first need to determine the URL to which the POST  
953 is sent. This location is obtained from the MachineTemplate collection URL that was returned as part of  
954 the CEP:

```
955 GET /machineTemplates HTTP/1.1
```

```

956
957 HTTP/1.1 200 OK
958 Content-Type: application/json
959
960 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineTemplateCollection",
961   "id": "http://example.com/machineTemplates",
962   "updated": "2015-08-25T12:00:00Z",
963   "parent": "http://example.com/CEP",
964   "count": 0,
965   "operations": [
966     { "rel": "add", "href": "http://example.com/machineTemplates" }
967   ]
968 }
```

969 Note that there are no MachineTemplates in the environment right now.

970 Now create the new MachineTemplate resource:

```

971 POST /machineTemplates HTTP/1.1
972 Content-Type: application/json
973
974 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineTemplate",
975   "name": "Demo1",
976   "description": "My first demo",
977   "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
978   "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
979   "credential": { "href": "http://example.com/creds/12345" }
980 }
```

```

981
982 HTTP/1.1 201 Created
983 Location: http://example.com/machineTemplates/82754
```

### 984 1.3.3 Create a new Machine by using a Machine Template

985 Now create a new Machine by using this Machine Template:

```

986 POST /machines HTTP/1.1
987 Content-Type: application/json
988
989 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineCreate",
990   "name": "myMachine2",
991   "description": "My second machine",
992   "machineTemplate": { "href": "http://example.com/machineTemplates/82754" }
993 }
```

```

994
995 HTTP/1.1 201 Created
996 Location: http://example.com/machines/843799
```

### 997 1.4 Creating a new Machine connected to an existing Volume

998 This scenario creates a new Machine that boots from an existing Volume. This simple example assumes  
999 that the user knows that there is an existing Volume with the bootable property equal to true.

#### 1000 1.4.1 Retrieve the CEP

1001 The CEP provides the links to the set of resources that are available in this Cloud. Retrieve the CEP to  
1002 discover the URL to each collection:

```

1003 GET /CEP HTTP/1.1
1004
1005 HTTP/1.1 200 OK
1006 Content-Type: application/json
1007
```

```

1008 { "resourceType": "http://schemas.dmtf.org/cimi/2/CloudEntryPoint",
1009   "id": "http://example.com/CEP",
1010   "baseURI": "http://example.com/",
1011   "resourceMetadata": { "href": "http://example.com/resourceMetadata" },
1012   "machines": { "href": "http://example.com/machines" },
1013   "machineConfigs": { "href": "http://example.com/machineConfigs" },
1014   "machineImages": { "href": "http://example.com/machineImages" },
1015   "credentials": { "href": "http://example.com/credentials" },
1016   "volumes": { "href": "http://example.com/volumes" },
1017   "volumeConfigs": { "href": "http://example.com/volumeConfigs" }
1018 }
```

### 1019 1.4.2 Get the list of Volumes

1020 When you create a new Machine from a Volume, you need to decide which Volume to use. The Volume  
 1021 collection is the set of existing Volumes that this Cloud offers:

```

1022 GET /volumes HTTP/1.1
1023
1024 HTTP/1.1 200 OK
1025 Content-Type: application/json
1026
1027 { "resourceType": "http://schemas.dmtf.org/cimi/2/VolumeCollection",
1028   "id": "http://example.com/volumes",
1029   "updated": "2015-08-25T12:00:00Z",
1030   "parent": "http://example.com/CEP",
1031   "count": 2,
1032   "volumes": [
1033     { "resourceType": "http://schemas.dmtf.org/cimi/2/Volume",
1034       "id": "http://example.com/volumes/vol1",
1035       "name": "Win7-Bootable",
1036       "created": "2015-08-15T12:15:00Z",
1037       "updated": "2015-08-15T12:15:00Z",
1038       "parent": "http://example.com/volumes",
1039       "description": "A bootable volume running Windows 7",
1040       "state": "AVAILABLE",
1041       "capacity": 60000000,
1042       "bootable": true
1043     },
1044
1045     { "resourceType": "http://schemas.dmtf.org/cimi/2/Volume",
1046       "id": "http://example.com/volumes/vol2",
1047       "name": "Generic Volume",
1048       "created": "2015-08-15T12:15:00Z",
1049       "updated": "2015-08-15T12:15:00Z",
1050       "parent": "http://example.com/volumes",
1051       "description": "A generic volume for Windows",
1052       "state": "AVAILABLE",
1053       "capacity": 60000000,
1054       "bootable": true
1055     }
1056   ]
1057 }
```

### 1058 1.4.3 Choose a Volume

1059 Next examine each Volume to find the one that meets your needs. The first one is acceptable, so it will be  
 1060 used later.

### 1061 1.4.4 Create a new Machine

1062 Retrieve the Machines collection so you know to where to POST a new Machine:

```

1063 GET /machines HTTP/1.1
1064
1065 HTTP/1.1 200 OK
1066 Content-Type: application/json
1067
1068 { "resourceType": "http://schemas.dmtf.org/cimi/2/MachineCollection",
1069   "id": "http://example.com/machines",
1070   "updated": "2015-08-25T12:00:00Z",
1071   "parent": "http://example.com/CEP",
1072   "count": 0,
1073   "operations": [
1074     { "rel": "add", "href": "http://example.com/machines" }
1075   ]
1076 }
1077

```

1078 Now create a new one, connecting it to the bootable Volume:

```

1079 POST /machines HTTP/1.1
1080 Content-Type: application/json
1081
1082 { "resourceType": "http://schemas.dmtf.org/cimi/2/MachineCreate",
1083   "name": "myMachine2",
1084   "description": "My second machine",
1085   "machineTemplate": {
1086     "volumes": [
1087       { "initialLocation": "V",
1088         "href": "http://example.com/volumes/vol1" }
1089     ]
1090   }
1091 }

```

1092 Note that the MachineTemplate in this case does not specify a MachineImage or MachineConfiguration to  
1093 use. In this example, for simplicity, you can assume that the provider has default values for those.

```

1094
1095 HTTP/1.1 201 Created
1096 Location: http://example.com/machines/852108

```

## 1097 1.4.5 Query new Machine

1098 Retrieve the Machine to get the full representation of the new Machine:

```

1099 GET /machines/852108 HTTP/1.1
1100
1101 HTTP/1.1 200 OK
1102 Content-Type: application/json
1103
1104 { "resourceType": "http://schemas.dmtf.org/cimi/2/Machine",
1105   "id": "http://example.com/machines/852108",
1106   "name": "myMachine2",
1107   "description": "My second machine",
1108   "created": "2015-03-26T10:04:00Z",
1109   "updated": "2015-03-26T10:04:00Z",
1110   "parent": "http://example.com/machines",
1111   "state": "STOPPED",

```

```

1112 "cpu": "1",
1113 "memory": 4000000,
1114 "disks" : { "href": "http://example.com/machines/852108/disks",
1115 "volumes": { "href": "http://example.com/machines/852108/volumes" },
1116 "networkInterfaces": { "href": "http://example.com/machines/852108/NIs",
1117 "operations": [
1118     { "rel": "edit", "href": "http://example.com/machines/852108" },
1119     { "rel": "delete", "href": "http://example.com/machines/852108" },
1120     { "rel": "http://schemas.dmtf.org/cimi/2/action/start",
1121       "href": "http://example.com/machines/852108" }
1122   ]
1123 }
1124
1125 Notice the "state" attribute on the Machine is "STOPPED", which is the initial state of a new machine.

```

## 1.5 Defining and using System Templates

This scenario creates a new System Template that is used to create a new System. System Templates are convenience resources that allow for well-defined descriptions (configuration, image, etc.) of a System to be persisted such that it can be reused later.

### 1.5.1 Retrieve the CEP

The CEP provides the links to the set of resources that are available in this Cloud. Retrieve the CEP to discover the URL to each collection:

```

1133 GET /CEP HTTP/1.1
1134
1135 HTTP/1.1 200 OK
1136 Content-Type: application/json
1137
1138 { "resourceURI": "http://schemas.dmtf.org/cimi/2/CloudEntryPoint",
1139   "id": "http://example.com/CEP",
1140   "baseURI": "http://example.com/",
1141   "systemTemplates": { "href": "http://example.com/systemTemplates" },
1142   "machineTemplates": { "href": "http://example.com/machineTemplates" },
1143   "credentialTemplates": { "href": "http://example.com/credentialTemplates" },
1144   "volumeTemplates": { "href": "http://example.com/volumeTemplates" }
1145 }

```

### 1.5.2 Create a new System Template

A SystemTemplate is defined so that when instantiated the result is a Machine is created, a Volume is connected to the Machine, and a Credential resource exists. To achieve this configuration, the following component Resources are included: a SystemTemplate definition, a MachineTemplate by value, a VolumeTemplate by reference, and a CredentialTemplate by reference. The VolumeTemplate and CredentialTemplate resources are already available:

1152 VolumeTemplate:  
1153 <http://example.com/volumeTemplates/95839>

1154 CredentialTemplate:  
1155 <http://example.com/credentialTemplates/72000>

1156 Note Alternatively, the VolumeTemplate and CredentialTemplate may be included by value in the  
1157 MachineTemplate definition below. However, it is beneficial to immediately see in the SystemTemplate the resources  
1158 that are involved and in general, automatic creation of the credential is more secure.

1159 Before creating a SystemTemplate, the URL to which the POST is sent needs to be determined. This  
1160 location is obtained from the SystemTemplate collection URL that was returned as part of the CEP.

```
1161 GET /systemTemplates HTTP/1.1
```

```
1162
```

```
1163 HTTP/1.1 200 OK
```

```
1164 Content-Type: application/json
```

```
1165 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemTemplateCollection",  
1166   "id": "http://example.com/systemTemplates",  
1167   "updated": "2015-08-25T12:00:00Z",  
1168   "parent": "http://example.com/CEP",  
1169   "count": 0,  
1170   "operations": [  
1171     { "rel": "add", "href": "http://example.com/systemTemplates" }  
1172   ]  
1173 }
```

1174 Now create the new System Template resource:

```
1175 POST /systemTemplates HTTP/1.1
```

```
1176 Content-Type: application/json
```

```
1177  
1178 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemTemplate",  
1179   "name": "System Demo1",  
1180   "description": "My first system template demo",  
1181   "componentDescriptors": [  
1182     { "name": "MyMachine",  
1183       "type": "http://schemas.dmtf.org/cimi/2/Machine",  
1184       "machineTemplate":  
1185         { "name" : "Machine in system demo",  
1186           "description" : "Machine in system",  
1187           "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },  
1188           "machineImage": { "href": "http://example.com/images/WinXP-SP2" },  
1189           "credential": { "href": "#MyCredential" },  
1190           "volumes": [  
1191             { "initialLocation": "/vol",  
1192               "href": "#MyVolume"  
1193             }  
1194           ]  
1195         },  
1196       },  
1197       { "name": "MyCredential",  
1198         "type": "http://schemas.dmtf.org/cimi/2/Credential",  
1199         "credentialTemplate":  
1200           { "href": "http://example.com/credentialTemplates/72000" }  
1201       },  
1202       { "name": "MyVolume",  
1203         "type": "http://schemas.dmtf.org/cimi/2/Volume",  
1204         "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839" }  
1205     }  
1206   }  
1207 }
```

```

1205     }
1206   ]
1207 }
1208
1209 HTTP/1.1 201 Created
1210 Location: http://example.com/systemTemplates/48920

```

### 1.5.3 Create a new System by using a System Template

1212 Now create a new System by using this System Template:

```

1213 POST /systems HTTP/1.1
1214 Content-Type: application/json
1215
1216 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemCreate",
1217   "name": "MySystem1",
1218   "description": "My first system",
1219   "systemTemplate": { "href": "http://example.com/systemTemplates/48920" }
1220 }
1221
1222 HTTP/1.1 201 Created
1223 Location: http://example.com/systems/78342

```

1224 Note that, alternatively, the provider could have decided to return a reference to a Job resource instead of  
1225 waiting until the System is completely created. Instead of the above 201 response, this type of request  
1226 could have resulted in the following response:

```

1227 HTTP/1.1 202 Accepted
1228 CIMI-Job-URI: http://example.com/Jobs/90001
1229
1230 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Job",
1231   "id": "http://example.com/Jobs/90001",
1232   "name": "SystemCreationJob",
1233   "created": "2015-03-15T12:15:00Z",
1234   "updated": "2015-03-15T12:15:00Z",
1235   "parent": "http://example.com/CEP",
1236   "targetResource": { "href": "http://example.com/systems" },
1237   "affectedResources": {
1238     "href": "http://example.com/systems/110001",
1239   },
1240   "action": "add",
1241   "status": "RUNNING",
1242   "progress": 30,
1243   "timeOfStatusChange": "2015-03-15T12:15:00Z",
1244   "isCancellable": "true",
1245   "nestedJobs": [
1246     { "href": "http://example.com/Jobs/90002" },
1247     { "href": "http://example.com/Jobs/90003" }
1248   ]
1249 }

```

1250 According to this response, the provider chose to create two nested Jobs and the "affectedResources"  
1251 attribute includes a reference to the newly created System. Periodic retrieval of the Job's representation

1252 allows the Consumer to determine when the Job is completed, i.e., it is completed when the "progress"  
1253 attribute has a value of 100.

#### 1254 **1.5.4 Query the new System**

1255 Retrieve the System to get the full representation of the new System:

```
1256 GET /systems/87342
```

1257

```
1258 HTTP/1.1 200 OK
```

```
1259 Content-Type: application/json
```

1260

```
1261 { "resourceURI": "http://schemas.dmtf.org/cimi/2/System",
1262   "id": "http://example.com/systems/78342",
1263   "name": "MySystem1",
1264   "description": "My first system",
1265   "created": "2015-08-15T12:15:00Z",
1266   "updated": "2015-08-15T12:15:00Z",
1267   "parent": "http://example.com/systems",
1268   "state": "STOPPED",
1269   "machines": { "href": "http://example.com/systems/87432/machines" },
1270   "credentials": { "href": "http://example.com/systems/87342/creds" },
1271   "volumes": { "href": "http://example.com/systems/87342/vols" },
1272   "operations" : [
1273     { "rel": "edit", "href": "http://example.com/systems/78342" }
1274   ]
1275 }
```

### 1276 **1.6 Editing System Templates**

1277 In this scenario a second Machine is added to an existing System Template.

#### 1278 **1.6.1 Edit an existing System Template**

1279 Edit the System Template created in a previous scenario and add another machine that shares its  
1280 credential and volume resources:

1281 SystemTemplate:

```
1282 http://example.com/systemTemplates/48920
```

1283 Retrieve the existing SystemTemplate definition:

```
1284 GET /systemTemplates/48920 HTTP/1.1
```

1285

```
1286 HTTP/1.1 200 OK
```

```
1287 Content-Type: application/json
```

1288

```
1289 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemTemplate",
1290   "name": "System Demo1",
1291   "description": "My first system template demo",
1292   "created": "2015-08-15T12:15:00Z",
1293   "updated": "2015-08-15T12:15:00Z",
1294   "parent": "http://example.com/systemTemplates",
1295   "componentDescriptors": [
1296     { "name": "MyMachine",
```

```

1296     "type": "http://schemas.dmtf.org/cimi/2/Machine",
1297     "machineTemplate":
1298       { "name" : "Machine in system demo",
1299         "description" : "Machine in system",
1300         "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
1301         "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
1302         "credential": { "href": "#MyCredential" },
1303         "volumes": [
1304           { "initialLocation": "/vol",
1305             "href": "#MyVolume"
1306           }
1307         ]
1308       }
1309     },
1310     { "name": "MyCredential",
1311     "type": "http://schemas.dmtf.org/cimi/2/Credential",
1312     "credentialTemplate":
1313       { "href": "http://example.com/credentialTemplates/72000" }
1314     },
1315     { "name": "MyVolume",
1316       "type": "http://schemas.dmtf.org/cimi/2/Volume",
1317       "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839" }
1318     }
1319   ],
1320   "operations" : [
1321     { "rel": "edit", "href": "http://example.com/systemTemplates/48920" }
1322   ]
1323 }
```

1324 Now update the SystemTemplate resource:

```

1325 PUT /systemTemplates/48920 HTTP/1.1
1326 Content-Type: application/json
1327
1328 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemTemplate",
1329   "name": "System Demo1",
1330   "description": "My first system template demo",
1331   "parent": "http://example.com/systemTemplates",
1332   "componentDescriptors": [
1333     { "name": "MyMachine",
1334     "type": "http://schemas.dmtf.org/cimi/2/Machine",
1335       "machineTemplate":
1336         { "name" : "Machine in system demo",
1337           "description" : "Machine in system",
1338           "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
1339           "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
1340           "credential": { "href": "#MyCredential" },
1341           "volumes": [
1342             { "initialLocation": "/vol",
1343               "href": "#MyVolume",
1344             }
1345           ]
1346         }
1347       }
1348     ]
1349   }
1350 }
```

```
1345        ]
1346        },
1347        "quantity": 2
1348    },
1349    { "name": "MyCredential",
1350    "type": "http://schemas.dmtf.org/cimi/2/Credential",
1351    "credentialTemplate":
1352        { "href": "http://example.com/credentialTemplates/72000" }
1353    },
1354    { "name": "MyVolume",
1355    "type": "http://schemas.dmtf.org/cimi/2/Volume",
1356    "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839" }
1357    }
1358]
1359}
1360
1361HTTP/1.1 200 OK
1362Content-Type: application/json
1363
1364{ "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemTemplate",
1365  "name": "System Demo1",
1366  "description": "My first system template demo",
1367  "created": "2015-09-15T12:15:00Z",
1368  "updated": "2015-09-15T12:15:00Z",
1369  "parent": "http://example.com/systemTemplates",
1370  "componentDescriptors": [
1371    { "name": "MyMachine",
1372    "type": "http://schemas.dmtf.org/cimi/2/Machine",
1373      "machineTemplate":
1374        { "name" : "Machine in system demo",
1375          "description" : "Machine in system",
1376          "machineConfig": { "href": "http://example.com/machineConfigs/tiny" },
1377          "machineImage": { "href": "http://example.com/images/WinXP-SP2" },
1378          "credential": { "href": "#MyCredential" },
1379          "volumes": [
1380            { "initialLocation": "/vol",
1381              "href": "#MyVolume"
1382            }
1383          ]
1384        },
1385        "quantity": 2
1386    },
1387    { "name": "MyCredential",
1388    "type": "http://schemas.dmtf.org/cimi/2/Credential",
1389    "credentialTemplate":
1390        { "href": "http://example.com/credentialTemplates/72000" }
1391    },
1392    { "name": "MyVolume",
1393    "type": "http://schemas.dmtf.org/cimi/2/Volume",
```

```

1394     "volumeTemplate": { "href": "http://example.com/volumeTemplates/95839" }
1395   }
1396 ],
1397 "operations": [
1398   { "rel": "edit", "href": "http://example.com/systemTemplates/48920" }
1399 ]
1400 }
```

## 1401 1.6.2 Create a new System using a System Template

1402 Now create a new System using this System Template by sending a POST to the same URL that you  
 1403 used in the previous scenario to create a new System:

```

1404 POST /systems HTTP/1.1
1405 Content-Type: application/json
1406
1407 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemCreate",
1408   "name": "MySystem2",
1409   "description": "My second system",
1410   "systemTemplate": { "href": "http://example.com/systemTemplates/48920" }
1411 }
1412
1413 HTTP/1.1 201 Created
1414 Location: http://example.com/systems/78343
```

## 1415 1.6.3 Query the new System

1416 Retrieve the System to get the full representation of the new System:

```

1417 GET /systems/87343
1418
1419 HTTP/1.1 200 OK
1420 Content-Type: application/json
1421
1422 { "resourceURI": "http://schemas.dmtf.org/cimi/2/System",
1423   "id": "http://example.com/systems/78342",
1424   "name": "MySystem2",
1425   "description": "My second system",
1426   "created": "2015-10-15T12:15:00Z",
1427   "updated": "2015-10-15T12:15:00Z",
1428   "parent": "http://example.com/systems",
1429   "state": "STOPPED",
1430   "machines": { "href": "http://example.com/systems/87432/machines" },
1431   "credentials": { "href": "http://example.com/systems/87342/creds" },
1432   "volumes": { "href": "http://example.com/systems/87342/vols" },
1433   "operations" : [
1434     { "rel": "edit", "href": "http://example.com/systems/78342" }
1435   ]
1436 }
```

1437

## 1438 1.7 Provider responses and return values for special cases

### 1439 1.7.1 Unrecognized attributes in request filters

1440 A syntax error in the filter expression of a request results in an error being generated. The provider  
1441 returns a 400 'Bad Request' response to a query with an unrecognized attribute name used in the  
1442 \$filter query parameter.

1443 For example:

1444 GET /machines?\$filter=aaa='bbb' HTTP/1.1

1445 The Machine resource has no "aaa" attribute: The recommended action is to return a 400 'Bad Request'  
1446 because the consumer did not follow \$filter's syntax, i.e., "aaa" is not a "resource attribute name".

### 1447 1.7.2 Unreasonable requests

1448 CIMI consumers are expected to make reasonable requests. CIMI providers are not required to advertise  
1449 maximum attributes for all resources. A CIMI provider may set limits on the length of attribute values it  
1450 finds reasonable. It may reject a request it deems unreasonable. This is common practice in web-based  
1451 protocols today.

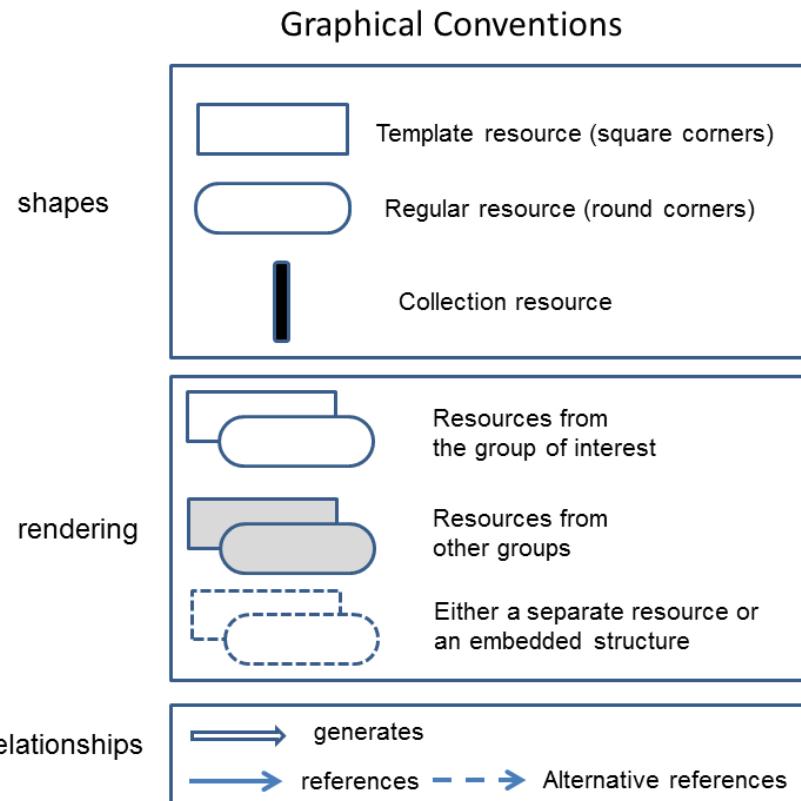
1452 These limits may not all be advertised in the ResourceMetadata, although it is recommended that CIMI  
1453 providers do so. A CIMI provider that receives a request that exceeds any of these limits, returns a  
1454 response with an appropriate standard HTTP status code, e.g., HTTP return code 413 'Request Entity  
1455 Too Large'.

## 1456 2 Resources modeling aspects

### 1457 2.1 Conventions

#### 1458 Figures:

1459 The figures of this section do not show the attributes of resources, but only the relationships across  
1460 resources. In all the figures, some referred resources such as `resourceMetadata`, `meters` and  
1461 `eventLog` are not represented, as they play the same role for all resources. The following graphical  
1462 notations apply in the remaining figures of this document:



1463

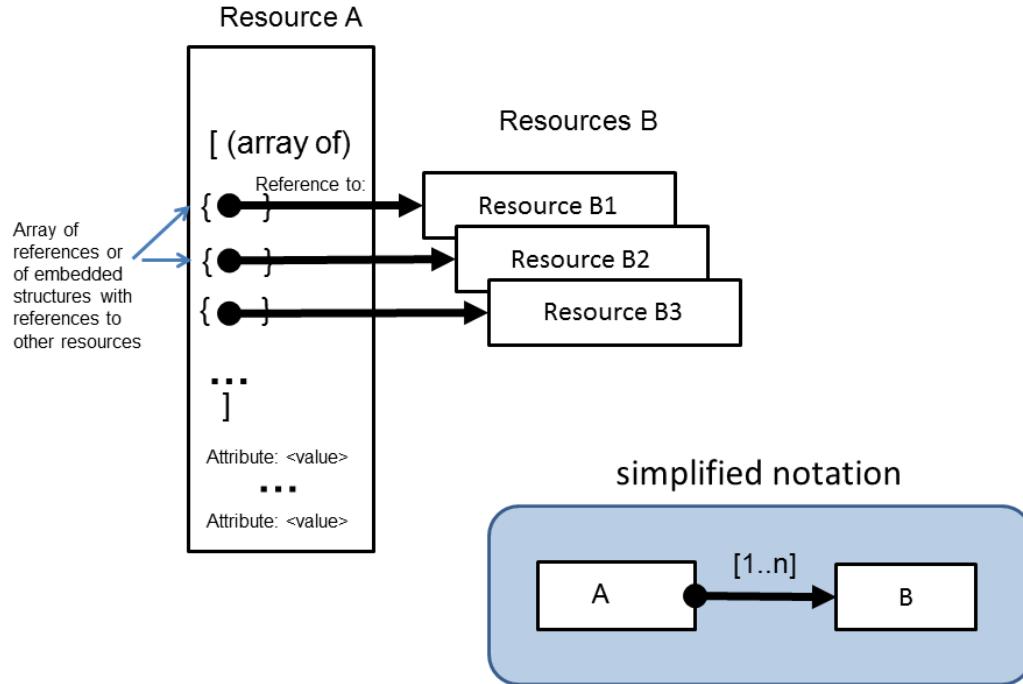
1464

**Figure 1 - Graphical conventions****2.2 1-to-N relationships using arrays and collections**

1466 There are two ways in CIMI for representing 1-N relationships between Resources: either by using arrays,  
1467 or by using collection resources.

- 1468 1. Array-based relationships. Such relationships are expected to be fairly static, not often updated,  
1469 and not involving large quantities of Resources. This is expected to be the case in templates.  
1470 Arrays in templates are not supposed to require special operations (like insert, remove, for  
1471 Collections) for their updates: such arrays in templates are supposed to have a fairly stable state  
1472 and may be changed by entire substitutions (HTTP PUT) if required. The ordinality of these  
1473 arrays (e.g., 0..n vs. 1..n, as well as possible size limits) are to be advertised in  
1474 ResourceMetadata. Such arrays are also supposed to have a small enough cardinality that they  
1475 can be sent back all at once when retrieving their container Resource.

### Array-based 1-n Resource relationship from A to Bs



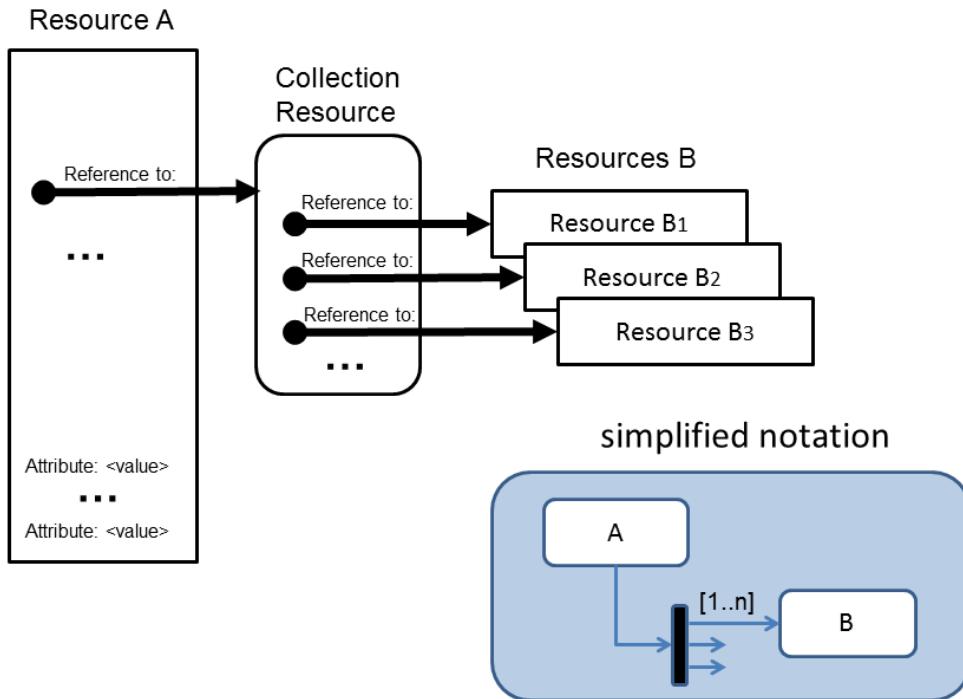
1476

1477

**Figure 2 – Array-based 1-n Resource relationship from A to Bs**

- 1478     2. Collection-based relationships. Such relationships are expected to be conveniently modified, (i.e.,  
 1479       not by substituting the entire relationship frame), by fine-grain operations on the Collection  
 1480       Resource such as insert, remove. Such collections may have a large cardinality but then they are  
 1481       not sent back all at once when retrieving their container Resource: only the reference to the  
 1482       Collection shows in the response to a GET resource.

### Collection-based 1-n Resource relationship from A to Bs



1483

1484

**Figure 3 – Collection-based 1-n Resource relationship from A to Bs**

1485 **2.3 Template instantiation**

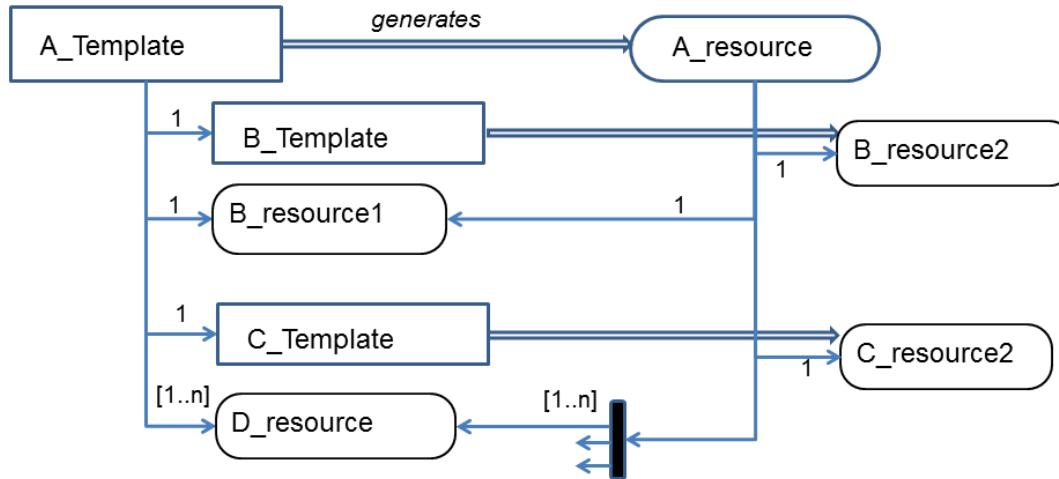
1486 Templates are Resources that are used to generate other Resources. For example, a MachineTemplate  
 1487 is used to generate Machines. It is said here that the Resources generated from a Template are  
 1488 instances of that Template.

1489 A Template may refer to other templates when the Resource it instantiates is composed. For example, a  
 1490 SystemTemplate typically refers to many other Templates that are needed to instantiate different parts of  
 1491 the System Resource.

1492 Also, a Template may refer to existing Resources other than Templates. For example, a  
 1493 MachineTemplate may refer to an existing Volume that the new Machine is expecting to connect to. Every  
 1494 time the MachineTemplate is used, it will create a new Machine that connects to that same Volume.

1495 Figure 4 shows what happens when instantiating a Resource from a Template that uses other Templates  
 1496 as well as other existing Resources.

## Template instantiation mechanism



1497

1498

**Figure 4 – Template instantiation mechanism**

1499 When instantiating the Template for “A” (A\_Template), a new “A” Resource is created (A\_resource in the  
1500 Figure 4),

1501 If the Template refers to other Templates – such as B\_Template and C\_Template in the figure – then new  
1502 Resources will be instantiated from these sub-templates (here, B\_resource2 and C\_resource2) every time  
1503 the top-level Template (here A\_Template) is instantiated. However, the existing Resources that were  
1504 referred to by the Template – here B\_resource1 and D\_resource(s) – are just reused and referenced by  
1505 the new Resource (A\_resource). This means every time the Template (here A\_template) is used to  
1506 generate a new Resource (here A\_resource), this new Resource will refer to the same existing Resources  
1507 that were used by this Template (here B\_resource1 and D\_resource(s)): in some way, any existing  
1508 Resource referred to by a Template is “shared” by all instances of that Template.

1509 Also note that Templates usually only use arrays for 1..N relationships, as such relationships are  
1510 supposed to be relatively static and do not justify the additional overhead of an intermediate Resource  
1511 such as a Collection. However, the generated Resources tend to use Collections as these allow for easier  
1512 updates and are more scalable. The D\_resource Resource above illustrates this case: the A\_Template is  
1513 referring to several of such Resources using an array (shown using the array simplified notion), but the  
1514 instantiated Resource (A\_resource) is referring to the same existing Resources (D\_resource instances)  
1515 using a Collection.

1516

1517

## 1518 2.4 About the nature of dependencies between resources

1519 Resources often relate to each other. One of the most pervasive relationships is the “parent” relationship.  
1520 The parent attribute present in every resource refers to its parent resource, i.e., another resource that  
1521 has control over its lifecycle. A resource is said to be a “child component” of its parent. All resources must  
1522 have a parent. If the resource is created by POSTing to a Collection of resources of this type, the  
1523 Collection becomes its parent.

1524 The lifecycle of a resource is controlled by its parent. When the parent resource is deleted, all its children  
1525 resources (and transitively, their children too) are also deleted. For example, if the parent attribute of  
1526 MachineImage is set to the MachineTemplate resource, then the MachineImage is a child component of  
1527 the MachineTemplate and will be deleted when its parent (the MachineTemplate) is deleted.

1528 How the parent attribute is set:

1529 Generally there are three ways a resource is created:

- 1530 (1) By POSTing to a Collection that exists directly under the CEP. In this case the Collection  
1531 resource will be the parent of the new resources. The parent of such a Collection is the CEP  
1532 itself. The POST contains the description of the new resource, typically a template for it – either  
1533 by value or by reference.
- 1534 (2) By POSTing to a Collection that is under another existing resource, of which the new resource  
1535 will be a component, e.g., by POSTing to a machines Collection of an existing System in order  
1536 to create and add a new Machine to the System. This can be seen as a variant of #1. In this case  
1537 the parent of the new component resource is the Collection under the composed resource, itself  
1538 having the composed resource as parent.
- 1539 (3) As the side effect of creating a composed resource. The template of the composed resource  
1540 would typically refer to a template for the component resource – e.g., a System template that  
1541 points to a machine Template. The composed resource generally becomes then the parent (or  
1542 grandparent if a Collection is involved) of the composed resource. Note that the template for the  
1543 composed resources may also refer to existing external resources (e.g., a Machine template may  
1544 refer to existing Volumes to be accessible by that Machine).

1545 The parent attribute value of a resource may be changed later by a Consumer only if the access mode  
1546 of this attribute allows for it. This would be advertised by a Provider in the ResourceMetadata associated  
1547 with that resource.

1548 The parent attribute must be set wisely for shared resources. If a resource must be shared by several  
1549 other resources as component– e.g., a canonical Linux image mandated by an employer would be shared  
1550 by several MachineTemplates - then it would not be wise to pick one as its parent unless there is a clear  
1551 understanding that this parent should not be deleted as long as the shared resource is needed  
1552 elsewhere. In such a case, it is better to have the parent of the shared resource (e.g., the canonical Linux  
1553 MachineImage) set to the machineImages CEP collection.

## 1554 2.5 Metadata about a resource

1555 Metadata about attributes:

1556 Metadata about attributes of a resource is indicated in a ResourceMetadata resource associated with this  
1557 resource. Metadata is indicative of such attribute characteristics as the optionality, the access right (read-  
1558 write, read-only) and possible values for the attribute.

1559 A ResourceMetadata resource may apply to a resource in three ways:

- 1560        (i) The ResourceMetadata resource is directly associated with this particular resource (see the  
1561            resourceMetadata common attribute.)
- 1562        (ii) The ResourceMetadata resource associated with the Template used to generate this particular  
1563            resource (using the genResourceMetadata attribute) – e.g., the ResourceMetadata  
1564            associated this way with a MachineTemplate will apply to all Machines generated from this  
1565            template, unless overridden later by a ResourceMetadata associated to this specific Machine  
1566            (case (i) above).
- 1567        (iii) The ResourceMetadata resource associated by default with this resource type in this CEP. Such  
1568            ResourceMetadata applies to all instances of this resource type under the CEP, unless  
1569            overridden by more specific ResourceMetadata instances (see cases (i) or (ii) above).

1570        Metadata about resources:

1571        At resource level, ResourceMetadata will indicate what attributes are supported by the resource. There  
1572            might be more – or less – attributes than specified in this document.

1573        ResourceMetadata will also indicate what associations are or are not supported by a resource. For  
1574            example, some Machines may not be associated with any Volume at creation time if their  
1575            MachineTemplate does not support the volumes and volumeTemplates attributes. In other cases,  
1576            some Machines may be associated with a limited number of Volumes. Such limitations will be indicated  
1577            by the ResourceMetadata associated with the template (via the genResourceMetadata attribute),  
1578            which applies to all resources generated from this template.

1579        Optionality of attributes:

1580        Except for some of the “common attributes” (see section 5.7.1 “Common Resource attributes in the  
1581            specification), the optionality of attributes is governed by their associated metadata, which may vary from  
1582            Provider to Provider, or from Consumer to Consumer (i.e., may depend on the CEP that a Consumer has  
1583            access to).

1584        The common attributes that a Provider must support (as stated in the CIMI specification) for any  
1585            Resource it implements are listed in this JSON serialization excerpt:

```
1586        "id": string,  
1587        "name": string,  
1588        "description": string,  
1589        "parent": string,  
1590        "properties": { string: string, + },
```

1591        The common attributes that a Consumer must always include for any Resource it serializes is the id  
1592            attribute (as this is the only attribute always required to be supported by the Consumer as stated in the  
1593            CIMI specification).

### 1594        **3 Groups of related Resources and their serialization**

#### 1595        **3.1 The Cloud Entry Point (CEP)**

1596        The CEP Resource is represented below in Figure 5, with other associated Resources:

1597        The Cloud Entry Point has references to a list of Collections as illustrated in Figure 5. It refers to both  
1598            Template Resources and other Resources. The CEP does not have a Template itself: it is the only  
1599            Resource that is always supposed to pre-exist to any Consumer access.

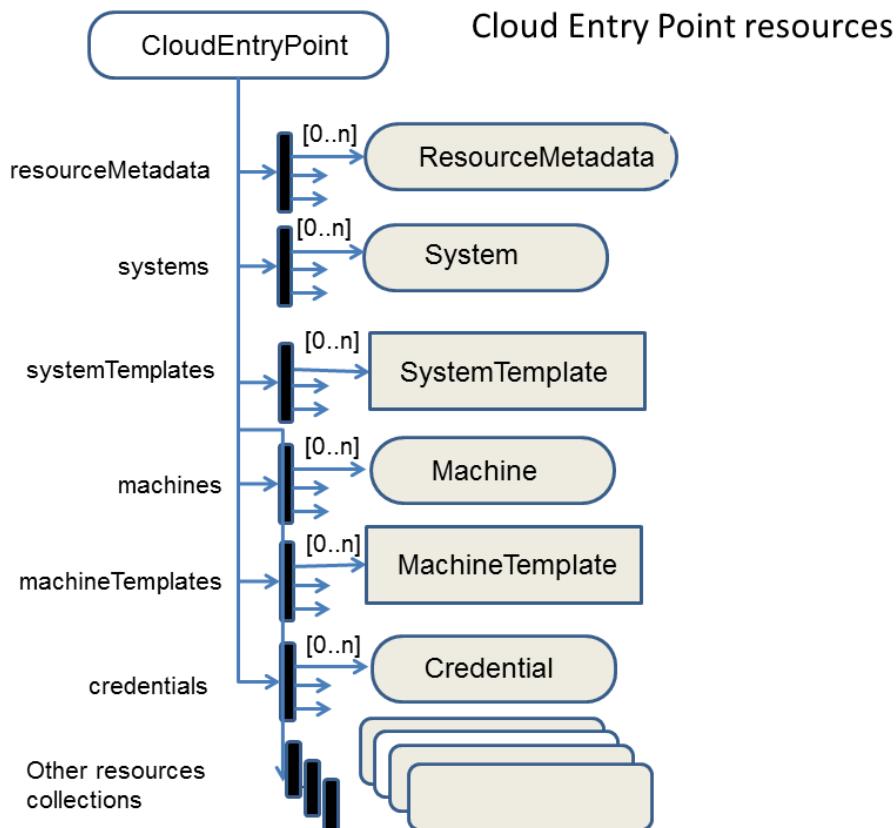


Figure 5 – Cloud Entry Point resources

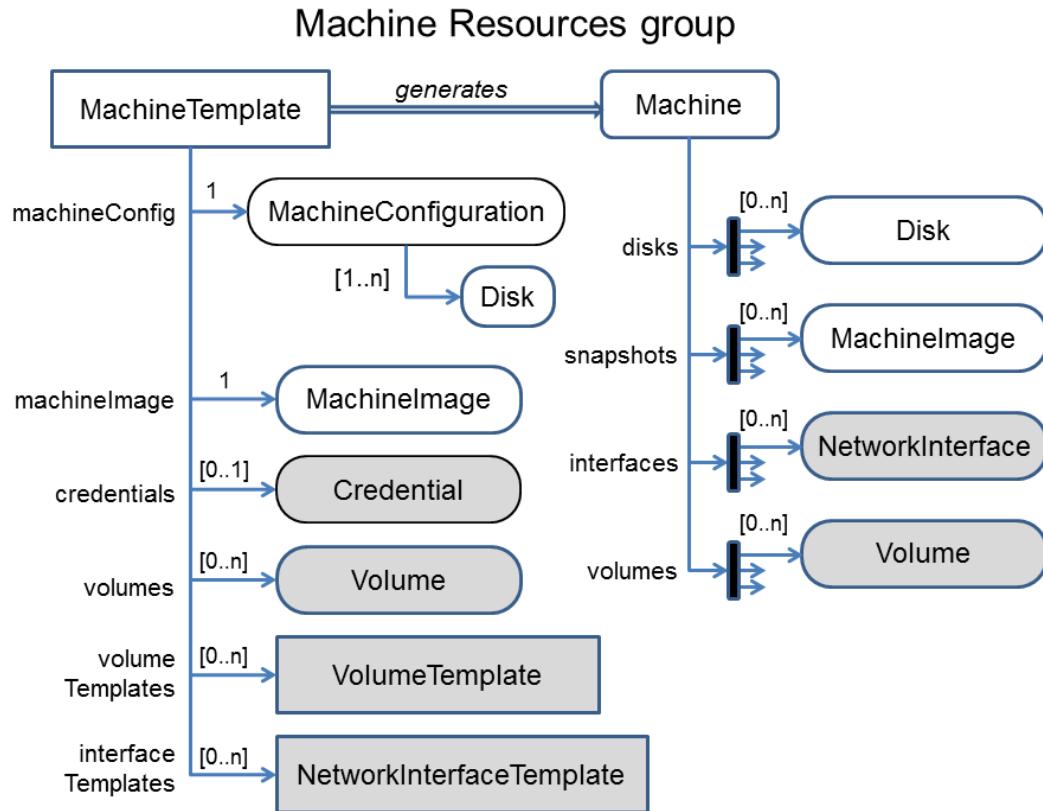
1600

1601

## 1602 3.2 Machine Resources group

### 1603 3.2.1 Machine Resources and their relationships

1604 The group of Machine-related Resources is represented below, with other associated Resources.



1605

1606

**Figure 6 – Machine Resources group**1607 **3.2.1.1 MachineTemplate**1608 A MachineTemplate refers to several other resources, either other templates (e.g., VolumeTemplate) or  
1609 regular resources (MachineImage, Volume...).1610 A Template may refer to other Templates for other types of Resources that need be either associated or  
1611 made components of the new Resource. This is the case here for `volumeTemplates` attribute used to  
1612 create new Volumes to be associated with the new Machine. However, such associated or component  
1613 Resources may also be pre-existing Resources, that the Template knows about and refers to. This is the  
1614 case of the `volumes` attribute here.

1615 The MachineTemplate supports both cases:

- When a MachineTemplate refers to a Resource such as a Volume, it means that this Volume as an already created Resource will be connected to the new Machine. When using the Template several times, each new Machine will be connected to the same Volume.
- When a MachineTemplate refers to a Resource such as a VolumeTemplate, it means this VolumeTemplate is used every time a new Machine is created, to create a new Volume to be connected to the new Machine. When using the Template several times, each new Machine will be connected to the a new Volume created from the VolumeTemplate.

1623 If the MachineTemplate refers to both VolumeTemplates and existing Volumes, the resulting machine will  
1624 have a `volumes` attribute referring to a Collection containing both pre-existing Volumes and new  
1625 Volumes.

1626 The ordinality associated with each referenced resource is shown in the above figure with a typical value  
 1627 or range. The actual ordinality supported by a Provider is determined by a ResourceMetadata resource  
 1628 associated with the MachineTemplate resource type. For example, there is typically one  
 1629 MachineConfiguration resource and one MachineImage resource referred by the MachineTemplate, but it  
 1630 maybe that a provider makes the machineConfig attribute optional. Same for the credential  
 1631 attribute, or the volumes attribute.

1632 MachineTemplate has one-to-many relationships to Volume, as well as to VolumeTemplate and  
 1633 NetworkInterfaceTemplate. Such relationships are typically represented by arrays in templates, as these  
 1634 are simpler than Collection resources although not as easily updated. Machine

1635 The Machine resource has relationships to other resources. Unlike for a template, one-to-many  
 1636 relationships are represented with Collection resources acting as intermediaries between the resource  
 1637 and its sets of referred resources. Again, the ordinality shown here is indicative of the most common  
 1638 cases.

### 1639 3.2.2 Machine serialization

1640 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 1641 both JSON and XML:

1642 **JSON media type:** application/json

1643 **JSON serialization:**

```
1644 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Machine",
1645   "id": string,
1646   "name": string,
1647   "description": string,
1648   "created": string,
1649   "updated": string,
1650   "parent": string,
1651   "properties": { string: string, + },
1652   "vscope" : [ valueScope, * ],
1653   "state": string,
1654   "cpu": number,
1655   "memory": number,
1656   "disks" : { "href": string },
1657   "cpuArch": string,
1658   "cpuSpeed": number,
1659   "volumes": { "href": string },
1660   "interfaces": { "href": string },
1661   "latestSnapshot": { "href": string },
1662   "snapshots": { "href": string },
1663   "meters": { "href": string },
1664   "eventLog": { "href": string },
1665   "operations": [
1666     { "rel": "edit", "href": string, ("available": boolean) },
```

```

1667     { "rel": "delete", "href": string, ("available": boolean) },
1668     { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string,
1669     ("available": boolean) },
1670     { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string,
1671     ("available": boolean) },
1672     { "rel": "http://schemas.dmtf.org/cimi/2/action/restart", "href": string,
1673     ("available": boolean) },
1674     { "rel": "http://schemas.dmtf.org/cimi/2/action/pause", "href": string,
1675     ("available": boolean) },
1676     { "rel": "http://schemas.dmtf.org/cimi/2/action/suspend", "href": string,
1677     ("available": boolean) },
1678     { "rel": "http://schemas.dmtf.org/cimi/2/action/snapshot", "href": string,
1679     ("available": boolean) },
1680     { "rel": "http://schemas.dmtf.org/cimi/2/action/restore", "href": string,
1681     ("available": boolean) }
1682   ]
1683   ...
1684 }
```

1685 **XML media type:** application/xml

1686 **XML serialization:**

```

1687 <Machine xmlns="http://schemas.dmtf.org/cimi/2">
1688   <id> xs:anyURI </id>
1689   <name> xs:string </name>
1690   <description> xs:string </description>
1691   <created> xs:dateTime </created>
1692   <updated> xs:dateTime </updated>
1693   <parent> xs:anyURI </parent>
1694   <properties>
1695     <property key="xs:string"> xs:string </property> *
1696   </properties>
1697   <vscope> valueScope </vscope>
1698   <state> xs:string </state>
1699   <cpu> xs:integer </cpu>
1700   <memory> xs:integer </memory>
1701   <disks href="xs:anyURI"/>
1702   <cpuArch> xs:string </cpuArch>
1703   <cpuSpeed> xs:integer </cpuSpeed>
1704   <volumes href="xs:anyURI"/>
1705   <interfaces href="xs:anyURI"/>
1706   <latestSnapshot href="xs:anyURI"/>
1707   <snapshots href="xs:anyURI"/>
1708   <meters href="xs:anyURI"/>
```

```

1709      <eventLog href="xs:anyURI"/>
1710
1711      <operations>
1712          <operation rel="edit" href="xs:anyURI" (available="xs:boolean") />
1713          <operation rel="delete" href="xs:anyURI" (available="xs:boolean") />
1714          <operation rel="http://schemas.dmtf.org/cimi/2/action/start"
1715              href="xs:anyURI" (available="xs:boolean") />
1716          <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
1717              href="xs:anyURI" (available="xs:boolean") />
1718          <operation rel="http://schemas.dmtf.org/cimi/2/action/restart"
1719              href="xs:anyURI" (available="xs:boolean") />
1720          <operation rel="http://schemas.dmtf.org/cimi/2/action/pause"
1721              href="xs:anyURI" (available="xs:boolean") />
1722          <operation rel="http://schemas.dmtf.org/cimi/2/action/suspend"
1723              href="xs:anyURI" (available="xs:boolean") />
1724          <operation rel="http://schemas.dmtf.org/cimi/2/action/capture"
1725              href="xs:anyURI" (available="xs:boolean") />
1726          <operation rel="http://schemas.dmtf.org/cimi/2/action/snapshot"
1727              href="xs:anyURI" (available="xs:boolean") />
1728          <operation rel="http://schemas.dmtf.org/cimi/2/action/restore"
1729              href="xs:anyURI" (available="xs:boolean") />
1730      </operations>
1731      <xss:any>*
1732  </Machine>
```

### 1732 3.2.3 DiskCollection serialization

1733 In the following serializations, the Disk resource is expanded: each item of the Collection shows the Disk  
 1734 attributes, instead of a reference.

#### 1735 JSON serialization:

```

1736      { "resourceURI": "http://schemas.dmtf.org/cimi/2/DiskCollection",
1737          "id": string,
1738          "updated": string,
1739          "parent": string,
1740          "count": number,
1741          "disks": [
1742              { "resourceURI": "http://schemas.dmtf.org/cimi/2/Disk",
1743                  "id": string,
1744                  "name": string,
1745                  "description": string,
1746                  "created": string,
1747                  "updated": string,
1748                  "properties": { string: string, + },
1749                  "capacity": number,
1750                  "initialLocation": string,
```

```

1751         "operations": [
1752             { "rel": "edit", "href": string },
1753             { "rel": "delete", "href": string }
1754         ]
1755         ...
1756     }, +
1757 ],
1758 "operations": [ { "rel": "add", "href": string } ]
1759 ...
1760 }
```

#### 1761 XML serialization:

```

1762 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/DiskCollection"
1763     xmlns="http://schemas.dmtf.org/cimi/2">
1764     <id> xs:anyURI </id>
1765     <updated> xs:dateTime </updated>
1766     <parent> xs:anyURI </parent>
1767     <count> xs:integer </count>
1768     <disks>
1769         <Disk>
1770             <id> xs:anyURI </id>
1771             <name> xs:string </name>
1772             <description> xs:string </description>
1773             <created> xs:dateTime </created>
1774             <updated> xs:dateTime </updated>
1775             <property key="xs:string"> xs:string </property> *
1776             <capacity> xs:integer </capacity>
1777             <initialLocation> xs:string </initialLocation>
1778             <operation rel="edit" href="xs:anyURI"/>
1779             <operation rel="delete" href="xs:anyURI"/>
1780             <xs:any>*
1781         </Disk> *
1782         <disks>
1783             <operation rel="add" href="xs:anyURI"/>
1784             <xs:any>*
1785     </Collection>
```

#### 1786 3.2.4 MachineCollection serialization

1787 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 1788 both JSON and XML:

1789 **JSON serialization:**

```

1790     { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineCollection",
1791         "id": string,
1792         "updated": string,
1793         "parent": string,
1794         "count": number,
1795         "machines": [
1796             { "resourceURI": "http://schemas.dmtf.org/cimi/2/Machine",
1797                 "id": string,
1798                 "name": string,
1799                 "description": string,
1800                 "created": string,
1801                 "updated": string,
1802                 "parent": string,
1803                 "properties": { string: string, + },
1804                 "machine": { "href": string },
1805                 "operations": [
1806                     { "rel": "edit", "href": string },
1807                     { "rel": "delete", "href": string }
1808                 ]
1809             }, +
1810         ],
1811         "operations": [
1812             { "rel": "add", "href": string },
1813             { "rel": "insert", "href": string },
1814             { "rel": "remove", "href": string }
1815         ]
1816         ...
1817     }

```

1818 **XML serialization:**

```

1819 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/MachineCollection"
1820     xmlns="http://schemas.dmtf.org/cimi/2">
1821     <id> xs:anyURI </id>
1822     <updated> xs:dateTime </updated>
1823     <parent> xs:anyURI </parent>
1824     <count> xs:integer </count>
1825     <machines>
1826         <Machine>
1827             <id> xs:anyURI </id>
1828             <name> xs:string </name>

```

```

1829      <description> xs:string </description>
1830      <created> xs:dateTime </created>
1831      <updated> xs:dateTime </updated>
1832      <parent> xs:anyURI </parent>
1833      <properties>
1834          <property key="xs:string"> xs:string </property> *
1835      </properties>
1836      <machine href="xs:anyURI"/>
1837      <operations>
1838          <operation rel="edit" href="xs:anyURI"/>
1839          <operation rel="delete" href="xs:anyURI"/>
1840      </operations>
1841      <xs:any>*
1842  </Machine> *
1843 /<machines>
1844 <operations>
1845     <operation rel="add" href="xs:anyURI"/>
1846     <operation rel="insert" href="xs:anyURI"/>
1847     <operation rel="remove" href="xs:anyURI"/>
1848 </operations>
1849     <xs:any>*
1850 </Collection>
```

### 1851 3.2.5 MachineTemplate serialization

1852 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 1853 both JSON and XML:

1854 **JSON media type:** application/json

1855 **JSON serialization:**

```

1856 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineTemplate",
1857   "id": string,
1858   "name": string,
1859   "description": string,
1860   "created": string,
1861   "updated": string,
1862   "parent": string,
1863   "properties": { string: string, + },
1864   "vscope" : [ valueScope, * ],
1865   "initialState": string,
1866   "machineConfig": {
1867     "href": string | ... MachineConfiguration attributes ...
```

```
1868     },
1869     "machineImage": {
1870         "href": string | ... MachineImage attributes ...
1871     },
1872     "credential": {
1873         "href": string | ... CredentialTemplate attributes ...
1874     },
1875     "volumes": [
1876         { "initialLocation": string, "href": string }, +
1877     ],
1878     "volumeTemplates": [
1879         { "initialLocation": string,
1880             "href": string,
1881             ... VolumeTemplate attributes ...
1882         }, +
1883     ],
1884     "interfaceTemplates": [
1885         { "href": string,
1886             ... NetworkInterfaceTemplate attributes ...
1887         }, *
1888     ],
1889     "userData": string,
1890     "meterTemplates": [
1891         { "href": string,
1892             ... MeterTemplate attributes ...
1893         }, *
1894     ],
1895     "eventLogTemplate": {
1896         "href": string,
1897         ... EventLogTemplate attributes ...
1898     },
1899     "genResourceMetadata": { "href": string }
1900     "operations": [
1901         { "rel": "edit", "href": string },
1902         { "rel": "delete", "href": string }
1903     ]
1904     ...
1905 }
```

1906 **XML media type:** application/xml

1907 **XML serialization:**

```
1908 <MachineTemplate xmlns="http://schemas.dmtf.org/cimi/2">
1909   <id> xs:anyURI </id>
1910   <name> xs:string </name>
1911   <description> xs:string </description>
1912   <created> xs:dateTime </created>
1913   <updated> xs:dateTime </updated>
1914   <parent> xs:anyURI </parent>
1915   <properties>
1916     <property key="xs:string"> xs:string </property> *
1917   </properties>
1918   <vscope> valueScope </vscope>
1919   <initialState> xs:string </initialState>
1920   <machineConfig href="xs:anyURI">
1921     ... MachineConfiguration attributes ...
1922   </machineConfig>
1923   <machineImage href="xs:anyURI">
1924     ... MachineImage attributes ...
1925   </machineImage>
1926   <credential href="xs:anyURI">
1927     ... CredentialTemplate attributes ...
1928   </credential>
1929   <volumes>
1930     <volume initialLocation="xs:string" href="xs:anyURI" /> *
1931   </volumes>
1932   <volumeTemplates>
1933     <volumeTemplate initialLocation="xs:string" href="xs:anyURI" >
1934       ... VolumeTemplate attributes ...
1935     </volumeTemplate> *
1936   </volumeTemplates>
1937   <interfaceTemplates>
1938     <interfaceTemplate href="xs:anyURI" >
1939       ... NetworkInterfaceTemplate attributes ...
1940     </interfaceTemplate> *
1941   </interfaceTemplates>
1942   <userData> xs:string </userData>
1943   <meterTemplates>
1944     <meterTemplate href="xs:anyURI" >
1945       ... MeterTemplate attributes ...
```

```

1946     </meterTemplate> *
1947     </meterTemplates>
1948     <eventLogTemplate href="xs:anyURI" >
1949         ... EventLogTemplate attributes ...
1950     </eventLogTemplate>
1951     <genResourceMetadata href="xs:anyURI" />
1952     <operations>
1953         <operation rel="edit" href="xs:anyURI"/>
1954         <operation rel="delete" href="xs:anyURI"/>
1955     </operations>
1956     <xs:any>*
1957 </MachineTemplate>

```

### 1958 3.2.6 MachineTemplateCollection serialization

1959 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 1960 both JSON and XML:

1961 This Resource shall be serialized as follows:

#### 1962 JSON serialization:

```

1963 {
1964     "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineTemplateCollection",
1965     "id": string,
1966     "updated": string,
1967     "parent": string,
1968     "count": number,
1969     "machineTemplates": [
1970         {
1971             "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineTemplate",
1972             "id": string,
1973             ... remaining MachineTemplate attributes ...
1974         },
1975         ...
1976     }

```

#### 1977 XML serialization:

```

1978 <Collection
1979     resourceURI="http://schemas.dmtf.org/cimi/2/MachineTemplateCollection"
1980     xmlns="http://schemas.dmtf.org/cimi/2">
1981     <id> xs:anyURI </id>
1982     <updated> xs:dateTime </updated>
1983     <parent> xs:anyURI </parent>
1984     <count> xs:integer </count>

```

```

1985      <machineTemplates>
1986          <MachineTemplate>
1987              <id> xs:anyURI </id>
1988                  ... remaining MachineTemplate attributes ...
1989          </MachineTemplate> *
1990      </machineTemplates>
1991      <operation rel="add" href="xs:anyURI"/>
1992      <xs:any>*
1993  </Collection>

```

### 1994 3.2.7 MachineConfiguration serialization

1995 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 1996 both JSON and XML:

1997 **JSON media type:** application/json

1998 **JSON serialization:**

```

1999      { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineConfiguration",
2000          "id": string,
2001          "name": string,
2002          "description": string,
2003          "created": string,
2004          "updated": string,
2005          "parent": string,
2006          "properties": { string: string, + },
2007          "vscope" : [ valueScope, * ],
2008          "cpu": number,
2009          "memory": number,
2010          "disks" : [
2011              { "capacity": number,
2012                  "format": string,
2013                  "initialLocation": string
2014              }, +
2015          ],
2016          "cpuArch": string,
2017          "cpuSpeed": number,
2018          "operations": [
2019              { "rel": "edit", "href": string },
2020              { "rel": "delete", "href": string }
2021          ]
2022          ...
2023      }

```

2024 **XML media type:** application/xml

2025 **XML serialization:**

```

2026 <MachineConfiguration xmlns="http://schemas.dmtf.org/cimi/2">
2027   <id> xs:anyURI </id>
2028   <name> xs:string </name>
2029   <description> xs:string </description>
2030   <created> xs:dateTime </created>
2031   <updated> xs:dateTime </updated>
2032   <parent> xs:anyURI </parent>
2033   <properties>
2034     <property key="xs:string"> xs:string </property> *
2035   </properties>
2036   <vscope> valueScope </vscope> *
2037   <cpu> xs:integer </cpu>
2038   <memory> xs:integer </memory>
2039   <disks>
2040     <disk>
2041       <capacity> xs:integer </capacity>
2042       <format> xs:string </format>
2043       <initialLocation> xs:string </initialLocation>
2044     </disk> *
2045   </disks>
2046   <cpuArch> xs:string </cpuArch>
2047   <cpuSpeed> xs:integer </cpuSpeed>
2048   <operations>
2049     <operation rel="edit" href="xs:anyURI"/>
2050     <operation rel="delete" href="xs:anyURI"/>
2051   </operations>
2052   <xs:any>*
2053 </MachineConfiguration>
```

### 2054 3.2.8 MachineConfigurationCollection serialization

2055 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2056 both JSON and XML:

2057 **JSON serialization:**

```

2058   { "resourceURI":
2059     "http://schemas.dmtf.org/cimi/2/MachineConfigurationCollection",
2060     "id": string,
2061     "updated": string,
2062     "parent": string,
```

```

2063     "count": number,
2064     "machineConfigurations": [
2065         { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineConfiguration",
2066             "id": string,
2067             ... remaining MachineConfiguration attributes ...
2068         }, +
2069     ],
2070     "operations": [ { "rel": "add", "href": string } ]
2071     ...
2072 }
```

#### 2073 XML serialization:

```

2074 <Collection
2075     resourceURI="http://schemas.dmtf.org/cimi/2/MachineConfigurationCollection"
2076     xmlns="http://schemas.dmtf.org/cimi/2">
2077     <id> xs:anyURI </id>
2078     <updated> xs:dateTime </updated>
2079     <parent> xs:anyURI </parent>
2080     <count> xs:integer </count>
2081     <machineConfigurations>
2082         <MachineConfiguration>
2083             <id> xs:anyURI </id>
2084             ... remaining MachineConfiguration attributes ...
2085         </MachineConfiguration> *
2086     </machineConfigurations>
2087     <operations>
2088         <operation rel="add" href="xs:anyURI"/>
2089     </operations>
2090     <xs:any>*
2091 </Collection>
```

#### 2092 3.2.9 MachineImage serialization

2093 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2094 both JSON and XML:

2095 **JSON media type:** application/json

#### 2096 JSON serialization:

```

2097     { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImage",
2098         "id": string,
2099         "name": string,
2100         "description": string,
2101         "created": string,
```

```

2102     "updated": string,
2103     "parent": string,
2104     "properties": { string: string, + },
2105     "state": string,
2106     "type": string,
2107     "imageLocation": string,
2108     "relatedImage": { "href": string },
2109     "operations": [
2110       { "rel": "edit", "href": string },
2111       { "rel": "delete", "href": string }
2112     ]
2113     ...
2114   }

```

2115 **XML media type:** application/xml

2116 **XML serialization:**

```

2117 <MachineImage xmlns="http://schemas.dmtf.org/cimi/2">
2118   <id> xs:anyURI </id>
2119   <name> xs:string </name>
2120   <description> xs:string </description>
2121   <created> xs:dateTime </created>
2122   <updated> xs:dateTime </updated>
2123   <parent> xs:anyURI </parent>
2124   <properties>
2125     <property key="xs:string"> xs:string </property> *
2126   </properties>
2127   <state> xs:string </state>
2128   <type> xs:string </type>
2129   <imageLocation> xs:anyURI </imageLocation>
2130   <relatedImage href="xs:anyURI"/>
2131   <operations>
2132     <operation rel="edit" href="xs:anyURI"/>
2133     <operation rel="delete" href="xs:anyURI"/>
2134   </operations>
2135   <xs:any>*
2136 </MachineImage>

```

### 2137 3.2.10 MachineImageCollection serialization

2138 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2139 both JSON and XML:

2140 **JSON serialization:**

```

2141     { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImageCollection",
2142         "id": string,
2143         "updated": string,
2144         "parent": string,
2145         "count": number,
2146         "machineImages": [
2147             { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineImage",
2148                 "id": string,
2149                 ... remaining MachineImage attributes ...
2150             }, +
2151         ],
2152         "operations": [ { "rel": "add", "href": string } ]
2153         ...
2154     }

```

2155 **XML serialization:**

```

2156 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/MachineImageCollection"
2157     xmlns="http://schemas.dmtf.org/cimi/2">
2158     <id> xs:anyURI </id>
2159     <updated> xs:dateTime </updated>
2160     <parent> xs:anyURI </parent>
2161     <count> xs:integer </count>
2162     <machineImages>
2163         <MachineImage>
2164             <id> xs:anyURI </id>
2165             ... remaining MachineImage attributes ...
2166             </MachineImage> *
2167         </machineImages>
2168         <operations>
2169             <operation rel="add" href="xs:anyURI"/>
2170         </operations>
2171         <xs:any>*
2172     </Collection>

```

### 2173 3.3 Credential Resources group

#### 2174 3.3.1 Credential serialization

2175 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2176 both JSON and XML:

2177 **JSON media type:** application/json

2178 **JSON serialization:**

```

2179     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Credential",
2180         "id": string,
2181         "name": string,
2182         "description": string,
2183         "created": string,
2184         "updated": string,
2185         "parent": string,
2186         "properties": { string: string, + },
2187         "operations": [
2188             { "rel": "edit", "href": string },
2189             { "rel": "delete", "href": string }
2190         ]
2191         ...
2192     }

```

2193 **XML media type:** application/xml

2194 **XML serialization:**

```

2195 <Credential xmlns="http://schemas.dmtf.org/cimi/2">
2196     <id> xs:anyURI </id>
2197     <name> xs:string </name>
2198     <description> xs:string </description>
2199     <created> xs:dateTime </created>
2200     <updated> xs:dateTime </updated>
2201     <parent> xs:anyURI </parent>
2202     <properties>
2203         <property key="xs:string"> xs:string </property> *
2204     </properties>
2205     <operations>
2206         <operation rel="edit" href="xs:anyURI"/>
2207         <operation rel="delete" href="xs:anyURI"/>
2208     </operations>
2209     <xs:any>*
2210 </Credential>

```

### 2211 3.3.2 CredentialCollection serialization

2212 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2213 both JSON and XML:

2214 **JSON serialization:**

```

2215     { "resourceURI": "http://schemas.dmtf.org/cimi/2/CredentialCollection",
2216         "id": string,
2217         "updated": string,
2218         "parent": string,
2219         "count": number,
2220         "credentials": [
2221             { "resourceURI": "http://schemas.dmtf.org/cimi/2/Credential",
2222                 "id": string,
2223                 ... remaining Credential attributes ...
2224             }, +
2225         ],
2226         "operations": [ { "rel": "add", "href": string } ]
2227         ...
2228     }

```

2229 **XML serialization:**

```

2230 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/CredentialCollection"
2231     xmlns="http://schemas.dmtf.org/cimi/2">
2232     <id> xs:anyURI </id>
2233     <updated> xs:dateTime </updated>
2234     <parent> xs:anyURI </parent>
2235     <count> xs:integer </count>
2236     <credentials>
2237         <Credential>
2238             <id> xs:anyURI </id>
2239             ... remaining Credential attributes ...
2240             </Credential> *
2241         </credentials>
2242         <operations>
2243             <operation rel="add" href="xs:anyURI"/>
2244         </operations>
2245         <xs:any>*
2246     </Collection>

```

### 2247 3.3.3 CredentialTemplate serialization

2248 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2249 both JSON and XML:

2250 **JSON media type:** application/json

2251 **JSON serialization:**

```

2252 { "resourceURI": "http://schemas.dmtf.org/cimi/2/CredentialTemplate",
2253   "id": string,
2254   "name": string, ?
2255   "description": string, ?
2256   "created": string, ?
2257   "updated": string, ?
2258   "parent": string, ?
2259   "properties": { string: string, + }, ?
2260   "operations": [
2261     { "rel": "edit", "href": string }, ?
2262     { "rel": "delete", "href": string } ?
2263   ] ?
2264   ...
2265 }
```

2266 **XML media type:** application/xml

2267 **XML serialization:**

```

2268 <CredentialTemplate xmlns="http://schemas.dmtf.org/cimi/2">
2269   <id> xs:anyURI </id>
2270   <name> xs:string </name> ?
2271   <description> xs:string </description> ?
2272   <created> xs:dateTime </created> ?
2273   <updated> xs:dateTime </updated> ?
2274   <parent> xs:anyURI </parent> ?
2275   <properties>
2276     <property key="xs:string"> xs:string </property> *
2277   </properties>
2278   <operations>
2279     <operation rel="edit" href="xs:anyURI"/> ?
2280     <operation rel="delete" href="xs:anyURI"/> ?
2281   </operations>
2282   <xs:any>*
2283 </CredentialTemplate>
```

### 2284 **3.3.4 CredentialTemplateCollection serialization**

2285 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2286 both JSON and XML:

2287       **JSON serialization:**

```
2288 { "resourceURI":  
2289   "http://schemas.dmtf.org/cimi/2/CredentialTemplateCollection",  
2290   "id": string,  
2291   "updated": string,  
2292   "parent": string,  
2293   "count": number,  
2294   "credentialTemplates": [  
2295     { "resourceURI": "http://schemas.dmtf.org/cimi/2/CredentialTemplate",  
2296       "id": string,  
2297       ... remaining CredentialTemplate attributes ...  
2298     }, +  
2299   ], ?  
2300   "operations": [ { "rel": "add", "href": string } ? ]  
2301   ...  
2302 }
```

2303       **XML serialization:**

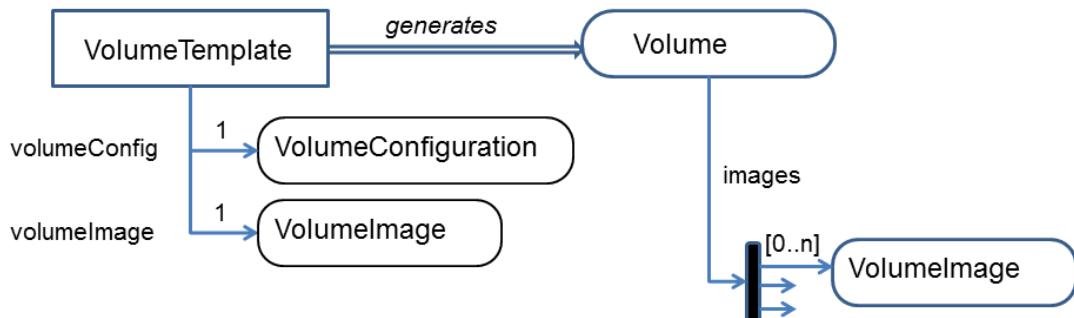
```
2304 <Collection  
2305   resourceURI="http://schemas.dmtf.org/cimi/2/CredentialTemplateCollection"  
2306   xmlns="http://schemas.dmtf.org/cimi/2">  
2307   <id> xs:anyURI </id>  
2308   <updated> xs:dateTime </updated>  
2309   <parent> xs:anyURI </parent>  
2310   <count> xs:integer </count>  
2311   <credentialTemplates>  
2312     <CredentialTemplate>  
2313       <id> xs:anyURI </id>  
2314       ... remaining CredentialTemplate attributes ...  
2315       </CredentialTemplate> *  
2316     </credentialTemplates>  
2317     <operation rel="add" href="xs:anyURI"/> ?  
2318     <xs:any>*  
2319   </Collection>
```

2320 **3.4 Volume Resources group**

2321 **3.4.1 Volume Resources group and their relationships**

2322 The group of Volume-related Resources is represented in Figure 7, with other associated  
2323 Resources.

Volume Resources group



2324

**Figure 7 – Volume Resources group**

2326 **3.4.2 Volume serialization**

2327 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
2328 both JSON and XML:

2329 **JSON media type:** application/json

2330 **JSON serialization:**

```

2331 { "resourceURI": "http://schemas.dmtf.org/cimi/2/volume",
2332   "id": string,
2333   "name": string, ?
2334   "description": string, ?
2335   "created": string, ?
2336   "updated": string, ?
2337   "parent": string, ?
2338   "properties": { string: string, + }, ?
  
```

```
2339     "state": string,  
2340     "type": string,  
2341     "capacity": number,  
2342     "bootable": boolean,  
2343     "images": { "href": string }, ?  
2344     "meters": { "href": string }, ?  
2345     "eventLog": { "href": string }, ?  
2346     "operations": [  
2347         { "rel": "edit", "href": string }, ?  
2348         { "rel": "delete", "href": string } ?  
2349     ] ?  
2350     ...  
2351 }
```

2352 **XML media type:** application/xml

2353 **XML serialization:**

```
2354 <Volume xmlns="http://schemas.dmtf.org/cimi/2">  
2355     <id> xs:anyURI </id>  
2356     <name> xs:string </name> ?  
2357     <description> xs:string </description> ?  
2358     <created> xs:dateTime </created> ?  
2359     <updated> xs:dateTime </updated> ?  
2360     <parent> xs:anyURI </parent> ?  
2361     <properties>  
2362         <property key="xs:string"> xs:string </property> *  
2363     </properties>  
2364     <state> xs:string </state>  
2365     <type> xs:anyURI </type>  
2366     <capacity> xs:integer </capacity>  
2367     <bootable> xs:boolean </bootable>  
2368     <images href="xs:anyURI"/> ?  
2369     <meters href="xs:anyURI"/> ?  
2370     <eventLog href="xs:anyURI"/> ?  
2371     <operations>  
2372         <operation rel="edit" href="xs:anyURI"/> ?  
2373         <operation rel="delete" href="xs:anyURI"/> ?  
2374     </operations>  
2375     <xs:any>*  
2376 </Volume>
```

2377 **3.4.3 VolumeCollection serialization**

2378 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2379 both JSON and XML:

2380 **JSON serialization:**

```
2381 { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeCollection",
2382   "id": string,
2383   "updated": string,
2384   "parent": string,
2385   "count": number,
2386   "volumes": [
2387     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Volume",
2388       "id": string,
2389       ... remaining Volume attributes ...
2390     }, +
2391   ], ?
2392   "operations": [ { "rel": "add", "href": string } ? ]
2393   ...
2394 }
```

2395 **XML serialization:**

```
2396 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/VolumeCollection"
2397   xmlns="http://schemas.dmtf.org/cimi/2">
2398   <id> xs:anyURI </id>
2399   <updated> xs:dateTime </updated>
2400   <parent> xs:anyURI </parent>
2401   <count> xs:integer </count>
2402   <volumes>
2403     <Volume>
2404       <id> xs:anyURI </id>
2405       ... remaining Volume attributes ...
2406       </Volume> *
2407     </volumes>
2408     <operations>
2409       <operation rel="add" href="xs:anyURI"/> ?
2410     </operations>
2411     <xs:any>*
2412   </Collection>
```

2413

**2414 3.4.4 locatedVolumeCollection serialization**

2415 This collection is using an additional accessory attribute (the initial location of the Volume), compared with  
2416 the basic Volume collection.

2417 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
2418 both JSON and XML:

**2419 JSON serialization:**

```
2420     { "resourceURI": "http://schemas.dmtf.org/cimi/2/locatedVolumeCollection",
2421         "id": string,
2422         "updated": string,
2423         "parent": string,
2424         "count": number,
2425         "locatedVolumes": [
2426             { "resourceURI": "http://schemas.dmtf.org/cimi/2/locatedVolume",
2427                 "id": string,
2428                 "name": string, ?
2429                 "description": string, ?
2430                 "created": string, ?
2431                 "updated": string, ?
2432                 "parent": string, ?
2433                 "properties": { string: string, + }, ?
2434                 "initialLocation": string, ?
2435                 "volume": { "href": string },
2436                 "operations": [
2437                     { "rel": "edit", "href": string }, ?
2438                     { "rel": "delete", "href": string } ?
2439                 ] ?
2440                 ...
2441             }, +
2442         ], ?
2443         "operations": [
2444             { "rel": "add", "href": string } ?
2445             { "rel": "insert", "href": string } ?
2446             { "rel": "remove", "href": string } ?
2447         ]
2448         ...
2449     }
```

**2450 XML serialization:**

```
2451     <Collection
2452         resourceURI="http://schemas.dmtf.org/cimi/2/locatedVolumeCollection"
```

```

2453      xmlns="http://schemas.dmtf.org/cimi/2">
2454      <id> xs:anyURI </id>
2455      <updated> xs:dateTime </updated>
2456      <parent> xs:anyURI </parent>
2457      <count> xs:integer </count>
2458      <locatedVolumes>
2459          <locatedVolume>
2460              <id> xs:anyURI </id>
2461              <name> xs:string </name> ?
2462              <description> xs:string </description> ?
2463              <created> xs:dateTime </created> ?
2464              <updated> xs:dateTime </updated> ?
2465              <parent> xs:anyURI </parent> ?
2466              <properties>
2467                  <property key="xs:string"> xs:string </property> *
2468              </properties>
2469              <initialLocation> xs:string </initialLocation> ?
2470              <volume href="xs:anyURI"/>
2471              <operations>
2472                  <operation rel="edit" href="xs:anyURI"/> ?
2473                  <operation rel="delete" href="xs:anyURI"/> ?
2474              </operations>
2475              <xs:any>*
2476          </locatedVolume> *
2477      </locatedVolumes>
2478      <operations>
2479          <operation rel="add" href="xs:anyURI"/> ?
2480          <operation rel="insert" href="xs:anyURI"/> ?
2481          <operation rel="remove" href="xs:anyURI"/> ?
2482      </operations>
2483      <xs:any>*
2484  </Collection>

```

### 2485 3.4.5 VolumeTemplate serialization

2486 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2487 both JSON and XML:

2488 **JSON media type:** application/json

2489 **JSON serialization:**

```

2490     { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeTemplate",
2491       "id": string,

```

```

2492     "name": string, ?
2493     "description": string, ?
2494     "created": string, ?
2495     "updated": string, ?
2496     "parent": string, ?
2497     "properties": { string: string, + }, ?
2498     "volumeConfig": {
2499         "href": string | ... VolumeConfiguration attributes ...
2500     },
2501     "volumeImage": { "href": string }, ?
2502     "meterTemplates": [
2503         { "href": string, ?
2504             ... MeterTemplate attributes ... ?
2505         }, *
2506     ], ?
2507     "eventLogTemplate": {
2508         "href": string, ?
2509         ... EventLogTemplate attributes ... ?
2510     }, ?
2511     "genResourceMetadata": { "href": string } ?

2512

2513     "operations": [
2514         { "rel": "edit", "href": string }, ?
2515         { "rel": "delete", "href": string } ?
2516     ] ?
2517     ...
2518 }

```

2519 **XML media type:** application/xml

2520 **XML serialization:**

```

2521 <VolumeTemplate xmlns="http://schemas.dmtf.org/cimi/2">
2522     <id> xs:anyURI </id>
2523     <name> xs:string </name> ?
2524     <description> xs:string </description> ?
2525     <created> xs:dateTime </created> ?
2526     <updated> xs:dateTime </updated> ?
2527     <parent> xs:anyURI </parent> ?
2528     <properties>
2529         <property key="xs:string"> xs:string </property> *
2530     </properties> ?

```

```

2531     <volumeConfig href="xs:anyURI"?>
2532         ... VolumeConfiguration attributes ... ?
2533     </volumeConfig>
2534     <volumeImage href="xs:anyURI"/> ?
2535     <meterTemplates>
2536         <MeterTemplate href="xs:anyURI"? >
2537             ... MeterTemplate attributes ... ?
2538         </MeterTemplate> *
2539     </meterTemplates>
2540     <eventLogTemplate href="xs:anyURI"? >
2541         ... EventLogTemplate attributes ... ?
2542     </eventLogTemplate> ?
2543     <genResourceMetadata href="xs:anyURI" /> ?
2544     <operations>
2545         <operation rel="edit" href="xs:anyURI"/> ?
2546         <operation rel="delete" href="xs:anyURI"/> ?
2547     </operations>
2548     <xs:any>*
2549 </VolumeTemplate>

```

### 2550 3.4.6 VolumeTemplateCollection serialization

2551 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2552 both JSON and XML:

2553 This Resource shall be serialized as follows:

#### 2554 JSON serialization:

```

2555     { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeTemplateCollection",
2556         "id": string,
2557         "updated": string,
2558         "parent": string,
2559         "count": number,
2560         "volumeTemplates": [
2561             { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeTemplate",
2562                 "id": string,
2563                 ... remaining volumeTemplate attributes ...
2564             }, +
2565         ], ?
2566         "operations": [ { "rel": "add", "href": string } ? ]
2567         ...
2568     }

```

2569 **XML serialization:**

```

2570     <Collection
2571         resourceURI="http://schemas.dmtf.org/cimi/2/VolumeTemplateCollection"
2572         xmlns="http://schemas.dmtf.org/cimi/2">
2573         <id> xs:anyURI </id>
2574         <updated> xs:dateTime </updated>
2575         <parent> xs:anyURI </parent>
2576         <count> xs:integer </count>
2577         <volumeTemplates>
2578             <VolumeTemplate>
2579                 <id> xs:anyURI </id>
2580                 ... remaining VolumeTemplates attributes ...
2581             </VolumeTemplate> *
2582         </volumeTemplates>
2583         <operations>
2584             <operation rel="add" href="xs:anyURI"/> ?
2585         </operations>
2586         <xss:any>*
2587     </Collection>
```

2588 **3.4.7 VolumeConfiguration serialization**

2589 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2590 both JSON and XML:

2591 **JSON media type:** application/json

2592 **JSON serialization:**

```

2593     { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeConfiguration",
2594         "id": string,
2595         "name": string, ?
2596         "description": string, ?
2597         "created": string, ?
2598         "updated": string, ?
2599         "parent": string, ?
2600         "properties": { string: string, + }, ?
2601         "type": string,
2602         "format": string,
2603         "capacity": number,
2604         "operations": [
2605             { "rel": "edit", "href": string }, ?
2606             { "rel": "delete", "href": string } ?
2607         ] ?
```

2608     ...  
2609 }

2610 **XML media type:** application/xml

2611 **XML serialization:**

```
<VolumeConfiguration xmlns="http://schemas.dmtf.org/cimi/2">
  <id> xs:anyURI </id>
  <name> xs:string </name> ?
  <description> xs:string </description> ?
  <created> xs:dateTime </created> ?
  <updated> xs:dateTime </updated> ?
  <parent> xs:anyURI </parent> ?
  <properties>
    <property key="xs:string"> xs:string </property> *
  </properties> ?
  <type> xs:anyURI </type>
  <format> xs:string </format>
  <capacity> xs:integer </capacity>
  <operations>
    <operation rel="edit" href="xs:anyURI"/> ?
    <operation rel="delete" href="xs:anyURI"/> ?
  </operations>
  <xs:any>*
</VolumeConfiguration>
```

### 2631 3.4.8 VolumeConfigurationCollection serialization

2632 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
2633 both JSON and XML:

2634 **JSON serialization:**

```
{ "resourceURI":  
  "http://schemas.dmtf.org/cimi/2/VolumeConfigurationCollection",  
  "id": string,  
  "updated": string,  
  "parent": string,  
  "count": number,  
  "volumeConfigurations": [  
    { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeConfiguration",  
      "id": string,  
      ... remaining VolumeConfiguration attributes ...  
    }, +  
  ], ?
```

```

2647     "operations": [ { "rel": "add", "href": string } ? ]
2648     ...
2649 }
```

#### 2650 XML serialization:

```

2651 <Collection
2652   resourceURI="http://schemas.dmtf.org/cimi/2/VolumeConfigurationCollection"
2653   xmlns="http://schemas.dmtf.org/cimi/2">
2654   <id> xs:anyURI </id>
2655   <updated> xs:dateTime </updated>
2656   <parent> xs:anyURI </parent>
2657   <count> xs:integer </count>
2658   <volumeConfigurations>
2659     <VolumeConfiguration>
2660       <id> xs:anyURI </id>
2661       ... remaining VolumeConfiguration attributes ...
2662     </VolumeConfiguration> *
2663   </volumeConfigurations>
2664   <operations>
2665     <operation rel="add" href="xs:anyURI"/> ?
2666   </operations>
2667   <xs:any>*
2668 </Collection>
```

#### 2669 3.4.9 VolumeImage serialization

2670 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
2671 both JSON and XML:

2672 **JSON media type:** application/json

#### 2673 JSON serialization:

```

2674 { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeImage",
2675   "id": string,
2676   "name": string, ?
2677   "description": string, ?
2678   "created": string, ?
2679   "updated": string, ?
2680   "parent": string, ?
2681   "properties": { string: string, + }, ?
2682   "state": string,
2683   "imageLocation": string,
2684   "bootable": boolean,
2685   "operations": [
```

```

2686     { "rel": "edit", "href": string }, ?
2687     { "rel": "delete", "href": string } ?
2688   ] ?
2689   ...
2690 }
```

2691 **XML media type:** application/xml

2692 **XML serialization:**

```

2693 <VolumeImage xmlns="http://schemas.dmtf.org/cimi/2">
2694   <id> xs:anyURI </id>
2695   <name> xs:string </name> ?
2696   <description> xs:string </description> ?
2697   <created> xs:dateTime </created> ?
2698   <updated> xs:dateTime </updated> ?
2699   <parent> xs:anyURI </parent> ?
2700   <properties>
2701     <property key="xs:string"> xs:string </property> *
2702   </properties> ?
2703   <state> xs:string </state>
2704   <imageLocation>xs:anyURI</imageLocation>
2705   <bootable> xs:boolean </bootable>
2706   <operations>
2707     <operation rel="edit" href="xs:anyURI"/> ?
2708     <operation rel="delete" href="xs:anyURI"/> ?
2709   </operations>
2710   <xs:any>*
2711 </VolumeImage>
```

### 2712 3.4.10 VolumeImageCollection serialization

2713 The following pseudo-schemas (see notation in CIMI 1.3 describe the serialization of the Resource in  
2714 both JSON and XML:

2715 **JSON serialization:**

```

2716   { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeImageCollection",
2717     "id": string,
2718     "updated": string,
2719     "parent": string,
2720     "count": number,
2721     "volumeImages": [
2722       { "resourceURI": "http://schemas.dmtf.org/cimi/2/VolumeImage",
2723         "id": string,
2724         ... remaining VolumeImage attributes ...
```

```
2725     }, +
2726     ], ?
2727     "operations": [ { "rel": "add", "href": "string" } ? ]
2728     ...
2729 }
```

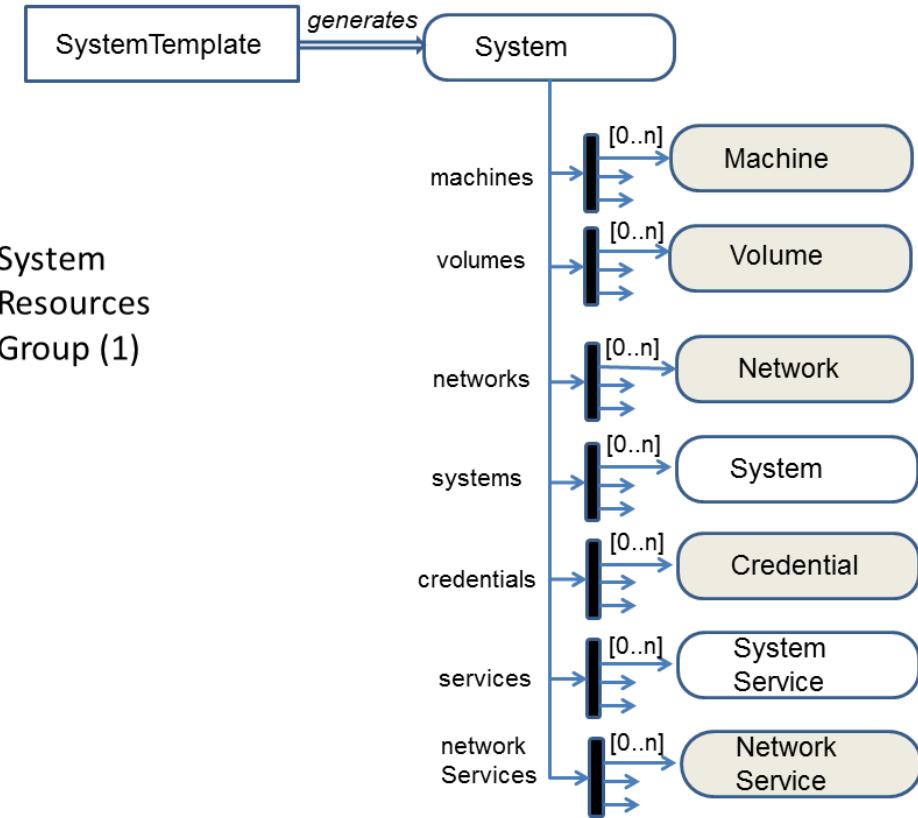
#### 2730 **XML serialization:**

```
2731 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/VolumeImageCollection"
2732   xmlns="http://schemas.dmtf.org/cimi/2">
2733   <id> xs:anyURI </id>
2734   <updated> xs:dateTime </updated>
2735   <parent> xs:anyURI </parent>
2736   <count> xs:integer </count>
2737   <volumeImages>
2738     <VolumeImage>
2739       <id> xs:anyURI </id>
2740       ... remaining VolumeImage attributes ...
2741       </VolumeImage> *
2742     </volumeImages>
2743     <operations>
2744       <operation rel="add" href="xs:anyURI"/> ?
2745     </operations>
2746     <xs:any>*
2747   </Collection>
```

### 2748 **3.5 System Resources group**

#### 2749 **3.5.1 System Resources and their relationships**

2750 The group of System-related Resources is represented in Figure 8, with other associated Resources  
2751 related to System:



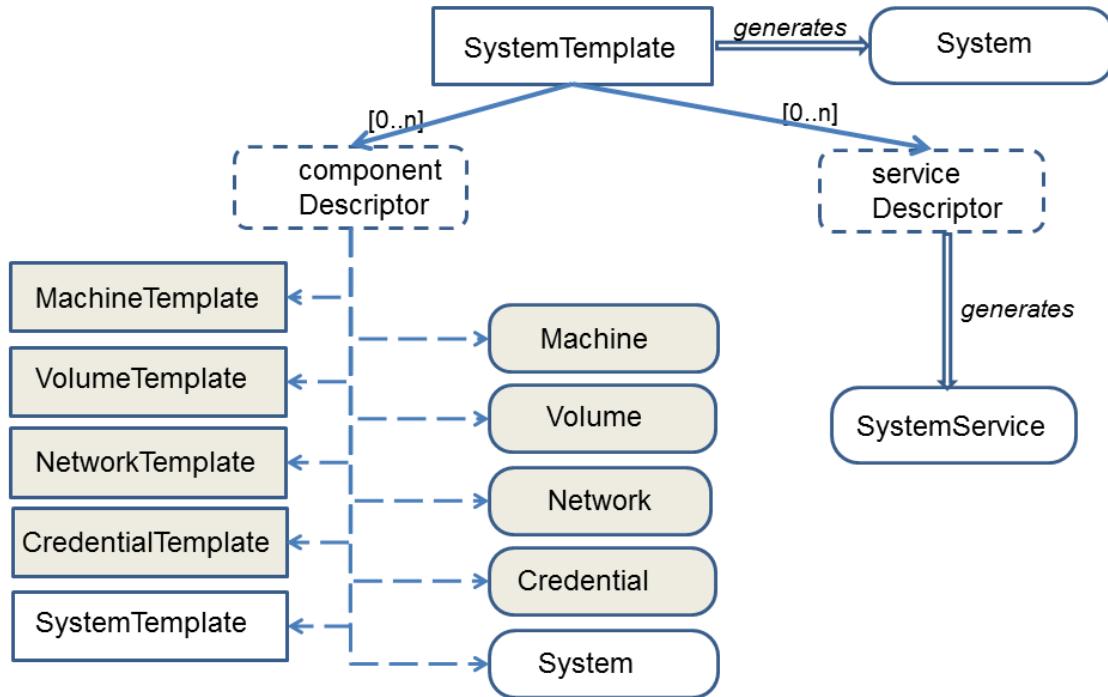
2752

2753

**Figure 8 – System Resources group**

2754 Resources related to SystemTemplate are shown in Figure 9:

## System Resources group (2)



2755

:

2756

**Figure 9 – SystemTemplate Resources group**

2757 **3.5.2 System serialization**

2758 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
2759 both JSON and XML:

2760 **JSON media type:** application/json

2761 **JSON serialization:**

```

2762     { "resourceURI": "http://schemas.dmtf.org/cimi/2/System",
2763       "id": string,
2764       "name": string, ?
2765       "description": string, ?
2766       "created": string, ?
2767       "updated": string, ?
2768       "parent": string, ?
2769       "properties": { string: string, + }, ?
2770       "state": string,
2771       "systems": { "href": string }, ?
2772       "machines": { "href": string }, ?
2773       "credentials": { "href": string }, ?
  
```

```

2774     "volumes": { "href": string }, ?
2775     "networks": { "href": string }, ?
2776     "networkServices": { "href": string }, ?
2777     "meters": { "href": string }, ?
2778     "eventLog": { "href": string }, ?
2779     "operations": [
2780         { "rel": "edit", "href": string, ("available": boolean)? }, ?
2781         { "rel": "delete", "href": string, ("available": boolean)? }, ?
2782         { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string,
2783         ("available": boolean)? }, ?
2784         { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string,
2785         ("available": boolean)? }, ?
2786         { "rel": "http://schemas.dmtf.org/cimi/2/action/restart", "href": string,
2787         ("available": boolean)? }, ?
2788         { "rel": "http://schemas.dmtf.org/cimi/2/action/pause", "href": string,
2789         ("available": boolean)? }, ?
2790         { "rel": "http://schemas.dmtf.org/cimi/2/action/suspend", "href": string,
2791         ("available": boolean)? }, ?
2792         { "rel": "http://schemas.dmtf.org/cimi/2/action/export", "href": string,
2793         ("available": boolean)? } ?
2794     ] ?
2795     ...
2796 }

```

2797 **XML media type:** application/xml

2798 **XML serialization:**

```

2799 <System xmlns="http://schemas.dmtf.org/cimi/2">
2800     <id> xs:anyURI </id>
2801     <name> xs:string </name> ?
2802     <description> xs:string </description> ?
2803     <created> xs:dateTime </created> ?
2804     <updated> xs:dateTime </updated> ?
2805     <parent> xs:anyURI </parent> ?
2806     <properties>
2807         <property key="xs:string"> xs:string </property> *
2808     </properties>
2809     <state> xs:string </state>
2810     <systems href="xs:anyURI"/> ?
2811     <machines href="xs:anyURI"/> ?
2812     <credentials href="xs:anyURI"/> ?
2813     <volumes href="xs:anyURI"/> ?
2814     <networks href="xs:anyURI"/> ?
2815     <networkServices href="xs:anyURI"/> ?

```

```

2816     <meters href="xs:anyURI"/> ?
2817     <eventLog href="xs:anyURI"/> ?
2818     <operations>
2819         <operation rel="edit" href="xs:anyURI" (available="xs:boolean")? /> ?
2820         <operation rel="delete" href="xs:anyURI" (available="xs:boolean")? /> ?
2821         <operation rel="http://schemas.dmtf.org/cimi/2/action/start"
2822             href="xs:anyURI" (available="xs:boolean")? /> ?
2823         <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
2824             href="xs:anyURI" (available="xs:boolean")? /> ?
2825         <operation rel="http://schemas.dmtf.org/cimi/2/action/restart"
2826             href="xs:anyURI" (available="xs:boolean")? /> ?
2827         <operation rel="http://schemas.dmtf.org/cimi/2/action/pause"
2828             href="xs:anyURI" (available="xs:boolean")? /> ?
2829         <operation rel="http://schemas.dmtf.org/cimi/2/action/suspend"
2830             href="xs:anyURI" (available="xs:boolean")? /> ?
2831         <operation rel="http://schemas.dmtf.org/cimi/2/action/export"
2832             href="xs:anyURI" (available="xs:boolean")? /> ?
2833     </operations>
2834     <xs:any>*
2835   </System>
2836

```

### 2837 3.5.3 SystemCollection serialization

2838 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2839 both JSON and XML:

#### 2840 JSON serialization:

```

2841 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemCollection",
2842   "id": string,
2843   "updated": string,
2844   "parent": string,
2845   "count", number,
2846   "systems": [
2847     { "resourceURI": "http://schemas.dmtf.org/cimi/2/System",
2848       "id": string,
2849       ... remaining System attributes ...
2850     }, +
2851   ], ?
2852   "operations": [
2853     { "rel": "add", "href": string }, ?
2854     { "rel": "remove", "href": string } ?

```

```

2855     { "rel": "insert", "href": string } ?      { "rel":
2856     "http://schemas.dmtf.org/cimi/2/action/import", "href": string } ?
2857     ]
2858     ...
2859 }
```

#### 2860 XML serialization:

```

2861 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/SystemCollection"
2862     xmlns="http://schemas.dmtf.org/cimi/2">
2863     <id> xs:anyURI </id>
2864     <updated> xs:dateTime </updated>
2865     <parent> xs:anyURI </parent>
2866     <count> xs:integer </count>
2867     <systems>
2868         <System>
2869             <id> xs:anyURI </id>
2870             ... remaining System attributes ...
2871         </System> *
2872     </systems>
2873     <operations>
2874         <operation rel="add" href="xs:anyURI"/> ?
2875         <operation rel="remove" href="xs:anyURI"/> ?
2876         <operation rel="insert" href="xs:anyURI"/> ?
2877         <operation rel="http://schemas.dmtf.org/cimi/2/action/import"
2878 href="xs:anyURI"/> ?
2879     </operations>
2880     <xss:any>*
2881 </Collection>
```

#### 2882 3.5.4 SystemService serialization

2883 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
2884 both JSON and XML:

2885 **JSON media type:** application/json

#### 2886 JSON serialization:

```

2887     { "resourceURI": "http://schemas.dmtf.org/cimi/2SystemService",
2888         "id": string,
2889         "name": string, ?
2890         "description": string, ?
2891         "created": string, ?
2892         "updated": string, ?
2893         "parent": string, ?
```

```
2894     "properties": { string: string, + }, ?
2895     "serviceType": string,
2896     "machines": { "href": string }, ?
2897     "volumes": { "href": string }, ?
2898     "systems": { "href": string }, ?
2899
2900     .."parameters": { string: string, + }, ?
2901
2902     "meters": { "href": string }, ?
2903     "eventLog": { "href": string }, ?
2904     "operations": [
2905         { "rel": "edit", "href": string }, ?
2906         { "rel": "delete", "href": string }, ?
2907     ] ?
2908     ...
2909 }
```

2910 **XML media type:** application/xml

2911 **XML serialization:**

```
2912 <SystemService xmlns="http://schemas.dmtf.org/cimi/2">
2913     <id> xs:anyURI </id>
2914     <name> xs:string </name> ?
2915     <description> xs:string </description> ?
2916     <created> xs:dateTime </created> ?
2917     <updated> xs:dateTime </updated> ?
2918     <parent> xs:anyURI </parent> ?
2919     <properties>
2920         <property key="xs:string"> xs:string </property> *
2921     </properties> ?
2922     <state> xs:string </state>
2923     <serviceType> xs:string </type>
2924     <machines href="xs:anyURI"/> *
2925     <volumes href="xs:anyURI"/> *
2926     <systems href="xs:anyURI"/> *
2927
2928     <parameters key="xs:string"> xs:string </parameters> *
2929
2930     <meters href="xs:anyURI"/> ?
2931     <eventLog" href="xs:anyURI"/> ?
2932     <operations>
```

```

2933     <operation rel="edit" href="xs:anyURI"/> ?
2934     <operation rel="delete" href="xs:anyURI"/> ?
2935   </operations>
2936   <xs:any>*
2937 </SystemService>
```

### 2938 3.5.5 SystemTemplate serialization

2939 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 2940 both JSON and XML:

2941 **JSON media type:** application/json

2942 **JSON serialization:**

```

2943 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemTemplate",
2944   "id": string,
2945   "name": string, ?
2946   "description": string, ?
2947   "created": string, ?
2948   "updated": string, ?
2949   "parent": string, ?
2950   "properties": { string: string, + }, ?
2951   "componentDescriptors": [
2952     { "name": string, ?
2953       "description": string, ?
2954       "properties": { string: string, + }, ?
2955       "type": string,
2956       "componentResource": {
2957         "href": string, ?
2958         ... ComponentResource attributes ... ?
2959       },
2960       "quantity": number ?
2961     }, +
2962   ], ?
2963   "serviceDescriptors": [
2964     { "name": string, ?
2965       "description": string, ?
2966       "properties": { string: string, + }, ?
2967       "serviceType": string,
2968       ...
2969     }, +
2970   ], ?
2971   "meterTemplates": [
```

```
2972     { "href": string, ?
2973         ... MeterTemplate attributes ... ?
2974     }, *
2975 ],
2976     "eventLogTemplate": {
2977         "href": string, ?
2978         ... EventLogTemplate attributes ... ?
2979     }, ?
2980     "importImage": string , ?
2981     "genResourceMetadata": { "href": string } ?
2982
2983     "operations": [
2984         { "rel": "edit", "href": string }, ?
2985         { "rel": "delete", "href": string }, ?
2986         { "rel": "http://schemas.dmtf.org/cimi/2/action/export", "href": string } ?
2987     ] ?
2988     ...
2989 }
```

2990 **XML media type:** application/xml

2991 **XML serialization:**

```
2992 <SystemTemplate xmlns="http://schemas.dmtf.org/cimi/2">
2993     <id> xs:anyURI </id>
2994     <name> xs:string </name> ?
2995     <description> xs:string </description> ?
2996     <created> xs:dateTime </created> ?
2997     <updated> xs:dateTime </updated> ?
2998     <parent> xs:anyURI </parent> ?
2999     <properties>
3000         <property key="xs:string"> xs:string </property> *
3001     </properties>
3002     <componentDescriptors>
3003         <componentDescriptor>
3004             <name> xs:string </name> ?
3005             <description> xs:string </description> ?
3006             <properties>
3007                 <property key="xs:string"> xs:string </property> *
3008             </properties> ?
3009             <type> xs:anyURI </type>
3010             <componentResource href="xs:anyURI"? >
```

```

3011      ... ComponentResource attributes ... ?
3012      </componentResource>
3013
3014      <quantity> xs:integer </quantity>
3015      </componentDescriptor> *
3016      </componentDescriptors>
3017      <serviceDescriptors>
3018      <serviceDescriptor>
3019      <name> xs:string </name> ?
3020      <description> xs:string </description> ?
3021      <properties>
3022      <property key="xs:string"> xs:string </property> *
3023      </properties> ?
3024      <serviceType> xs:anyURI </serviceType>
3025      ...
3026      </serviceDescriptor> *
3027      </serviceDescriptors>
3028
3029      <meterTemplates>
3030      <meterTemplate href="xs:anyURI"? >
3031      ... MeterTemplate attributes ... ?
3032      </meterTemplate> *
3033      </meterTemplates>
3034      <eventLogTemplate href="xs:anyURI"? >
3035      ... EventLogTemplate attributes ... ?
3036      </eventLogTemplate> ?
3037      <importImage > xs:anyURI </importImage> ?
3038      <genResourceMetadata href="xs:anyURI"? /> ?
3039      <operations>
3040      <operation rel="edit" href="xs:anyURI"/> ?
3041      <operation rel="delete" href="xs:anyURI"/> ?
3042      <operation rel="http://schemas.dmtf.org/cimi/2/action/export"
3043 href="xs:anyURI"/> ?
3044      </operations>
3045      <xs:any>*
3046      </SystemTemplate>

```

### 3.5.6 SystemTemplateCollection serialization

3048 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 3049 both JSON and XML:

3050 This Resource shall be serialized as follows:

3051 **JSON serialization:**

```
3052 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemTemplateCollection",
3053   "id": string,
3054   "updated": string,
3055   "parent": string,
3056   "count": number,
3057   "systemTemplates": [
3058     { "resourceURI": "http://schemas.dmtf.org/cimi/2/SystemTemplate",
3059       "id": string,
3060       ... remaining SystemTemplate attributes ...
3061     }, +
3062   ], ?
3063   "operations": [
3064     { "rel": "add", "href": string }, ?
3065     { "rel": "http://schemas.dmtf.org/cimi/2/action/import", "href": string } ?
3066   ]
3067   ...
3068 }
```

3069 **XML serialization:**

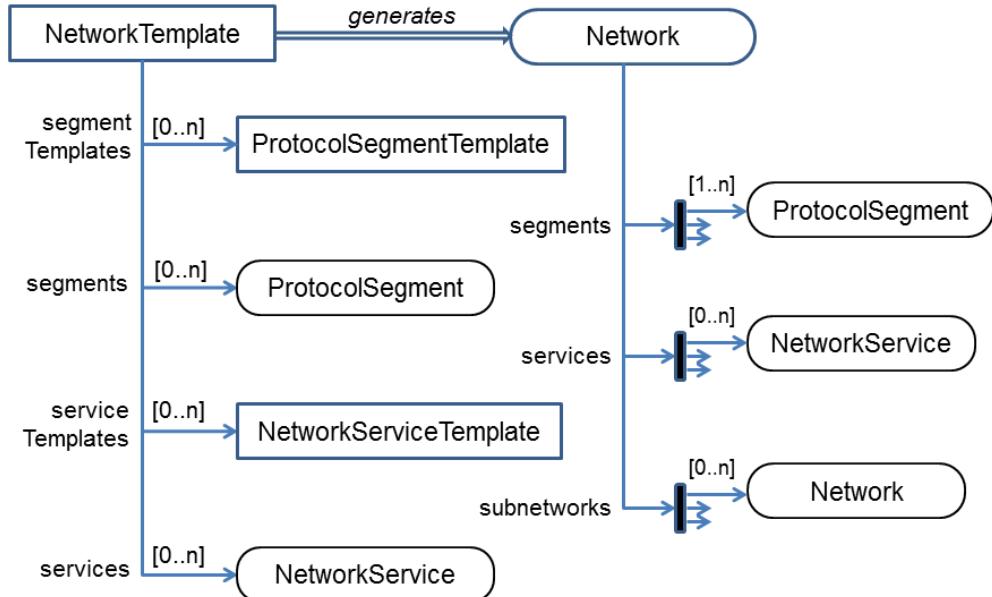
```
3070 <Collection
3071   resourceURI="http://schemas.dmtf.org/cimi/2/SystemTemplateCollection"
3072   xmlns="http://schemas.dmtf.org/cimi/2">
3073   <id> xs:anyURI </id>
3074   <updated> xs:dateTime </updated>
3075   <parent> xs:anyURI </parent>
3076   <count> xs:integer </count>
3077   <systemTemplates>
3078     <SystemTemplate>
3079       <id> xs:anyURI </id>
3080       ... remaining SystemTemplate attributes ...
3081     </SystemTemplate> *
3082   </systemTemplates>
3083   <operations>
3084     <operation rel="add" href="xs:anyURI"/> ?
3085     <operation rel="http://schemas.dmtf.org/cimi/2/action/import"
3086 href="xs:anyURI"/> ?
3087   </operations>
3088   <xss:any>*
3089 </Collection>
```

3090 **3.6 Network Resources group**

3091 **3.6.1 Network Resources and their relationships**

3092 The group of Network-related Resources and their relationships is represented in Figure 10:

**Network Resources group (1)**



3093

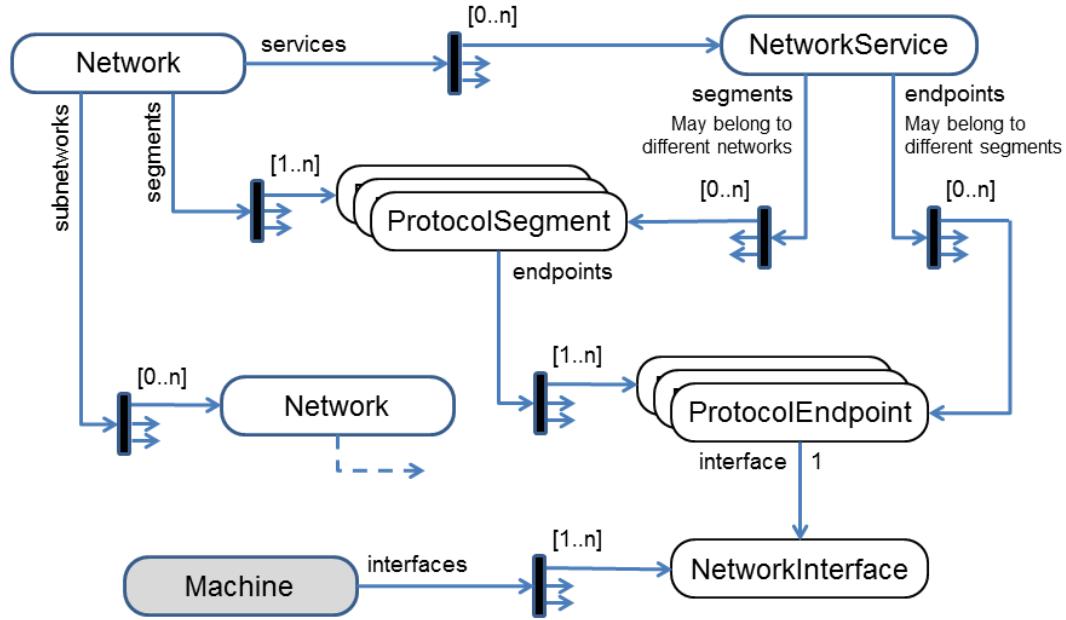
3094

**Figure 10 – Network Resources group**

3095 How network segments and endpoint resources relate to other resources, including Machines is shown in  
 3096 Figure 11. In particular, NetworkService resources may affect only parts of a Network (i.e., some  
 3097 ProtocolSegments). A NetworkService resource may also affect more than one Network, depending on  
 3098 the nature of the service.

## Network Resources group (2)

(Resource Structure Relationships)



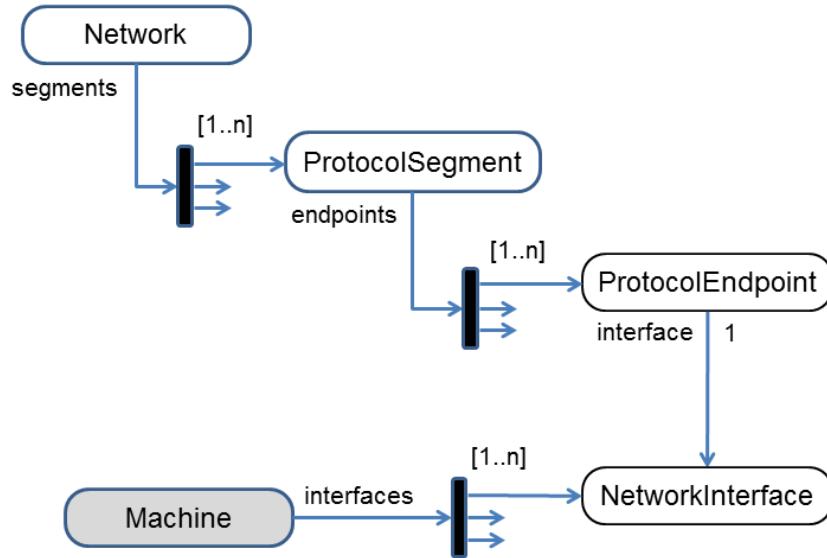
3099 :

3100

**Figure 11 – Network Resources group: Resource structure relationships**

3101 Figure 12 illustrates more particularly how Machine resources connect to Network resources.

### Network Resources group (3) (Network Connectivity Perspective)



3102

**Figure 12 – Network Resources group: Network connectivity perspective**

#### 3.6.2 Network serialization

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

**JSON media type:** application/json

**JSON serialization:**

```

3109  { "resourceURI": "http://schemas.dmtf.org/cimi/2/Network",
3110    "id": string,
3111    "name": string, ?
3112    "description": string, ?
3113    "created": string, ?
3114    "updated": string, ?
3115    "parent": string, ?
3116    "properties": { string: string, + }, ?
3117    "state": string,
3118    "segments": { "href": string },
3119    "services": { "href": string },
3120    "subnetworks": { "href": string }, ?
  
```

```
3121     "meters": { "href": string }, ?
3122     "eventLog": { "href": string }, ?
3123     "operations": [
3124         { "rel": "edit", "href": string }, ?
3125         { "rel": "delete", "href": string }, ?
3126         { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string }, ?
3127         { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string } ?
3128     ] ?
3129     ...
3130 }
```

3131 **XML media type:** application/xml

3132 **XML serialization:**

```
3133 <Network xmlns="http://schemas.dmtf.org/cimi/2">
3134     <id> xs:anyURI </id>
3135     <name> xs:string </name> ?
3136     <description> xs:string </description> ?
3137     <created> xs:dateTime </created> ?
3138     <updated> xs:dateTime </updated> ?
3139     <parent> xs:anyURI </parent> ?
3140     <properties>
3141         <property key="xs:string"> xs:string </property> *
3142     </properties> ?
3143     <state> xs:string </state>
3144     <segments href="xs:anyURI"/>
3145     <services href="xs:anyURI"/>
3146     <subnetworks href="xs:anyURI"/> ?
3147     <meters href="xs:anyURI"/> ?
3148     <eventLog href="xs:anyURI"/> ?
3149     <operation rel="edit" href="xs:anyURI"/> ?
3150     <operation rel="delete" href="xs:anyURI"/> ?
3151     <operations>
3152         <operation rel="http://schemas.dmtf.org/cimi/2/action/start"
3153             href="xs:anyURI"/> ?
3154         <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
3155             href="xs:anyURI"/> ?
3156     </operations>
3157     <xs:any>*
3158 </Network>
```

3159

3160 **3.6.3 NetworkCollection serialization**

3161 The following pseudo-schemas (see notation in CIMI 1.3) describe the serialization of the Resource in  
 3162 both JSON and XML:

3163 This Resource shall be serialized as follows:

3164 **JSON serialization:**

```
3165 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkCollection",
3166   "id": string,
3167   "updated": string,
3168   "parent": string,
3169   "count": number,
3170   "networks": [
3171     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Network",
3172       "id": string,
3173       ... remaining Network attributes ...
3174     }, +
3175   ], ?
3176   "operations": [ { "rel": "add", "href": string } ? ]
3177   ...
3178 }
```

3179 **XML serialization:**

```
3180 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/NetworkCollection"
3181   xmlns="http://schemas.dmtf.org/cimi/2">
3182   <id> xs:anyURI </id>
3183   <updated> xs:dateTime </updated>
3184   <parent> xs:anyURI </parent>
3185   <count> xs:integer </count>
3186   <networks>
3187     <Network>
3188       <id> xs:anyURI </id>
3189       ... remaining Network attributes ...
3190     </Network> *
3191   </networks>
3192   <operations>
3193     <operation rel="add" href="xs:anyURI"/> ?
3194   </operations>
3195   <xs:any>*
3196 </Collection>
```

3197 **3.6.4 NetworkTemplate serialization**

3198 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
3199 JSON and XML:

3200 **JSON media type:** application/json

3201 **JSON serialization:**

```
3202 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkTemplate",
3203   "id": string,
3204   "name": string, ?
3205   "description": string, ?
3206   "created": string, ?
3207   "updated": string, ?
3208   "parent": string, ?
3209   "properties": { string: string, + }, ?
3210   "initialState": string, ?
3211   "segments": { "href": string }, ?
3212   "segmentTemplates": [
3213     { "href": string, ?
3214       ... ProtocolSegmentTemplate attributes ... ?
3215     }, *
3216   ], ?
3217   "services": { "href": string }, ?
3218   "serviceTemplates": [
3219     { "href": string, ?
3220       ... NetworkServiceTemplate attributes ... ?
3221     }, *
3222   ], ?
3223   "subnetworks": { "href": string }, ?
3224   "subnetworkTemplates": [
3225     { "href": string, ?
3226       ... NetworkTemplate attributes ... ?
3227     }, *
3228   ], ?
3229   "meterTemplates": [
3230     { "href": string, ?
3231       ... MeterTemplate attributes ... ?
3232     }, *
3233   ], ?
3234   "eventLogTemplate": {
3235     "href": string, ?
```

```

3236     ... EventLogTemplate attributes ... ?
3237     }, ?
3238     "operations": [
3239         { "rel": "edit", "href": string }, ?
3240         { "rel": "delete", "href": string } ?
3241     ] ?
3242     ...
3243 }
```

3244 **XML media type:** application/xml

3245 **XML serialization:**

```

3246 <NetworkTemplate xmlns="http://schemas.dmtf.org/cimi/2">
3247     <id> xs:anyURI </id>
3248     <name> xs:string </name> ?
3249     <description> xs:string </description> ?
3250     <created> xs:dateTime </created> ?
3251     <updated> xs:dateTime </updated> ?
3252     <parent> xs:anyURI </parent> ?
3253     <properties>
3254         <property key="xs:string"> xs:string </property> *
3255     </properties> ?
3256     <initialState> xs:string </initialState> ?
3257     <segments href="xs:anyURI"/> ?
3258     <segmentTemplates>
3259         <SegmentTemplate href="xs:anyURI"? >
3260             ... ProtocolSegmentTemplate attributes ... ?
3261         </SegmentTemplate> *
3262     </segmentTemplates>
3263     <services href="xs:anyURI"/> ?
3264     <serviceTemplates>
3265         <ServiceTemplate href="xs:anyURI"? >
3266             ... NetworkServiceTemplates attributes ... ?
3267         </ServiceTemplate> *
3268     </serviceTemplates>
3269     <subnetworks href="xs:anyURI"/> ?
3270     <subnetworkTemplates>
3271         <NetworkTemplate href="xs:anyURI"? >
3272             ... NetworkTemplate attributes ... ?
3273         </NetworkTemplate> *
3274     </subnetworkTemplates>
```

```

3275      <meterTemplates>
3276          <MeterTemplate href="xs:anyURI"? >
3277              ... MeterTemplate attributes ... ?
3278          </MeterTemplate> *
3279      </meterTemplates>
3280      <eventLogTemplate href="xs:anyURI"? >
3281          ... EventLogTemplate attributes ... ?
3282      </eventLogTemplate> ?
3283      <operations>
3284          <operation rel="edit" href="xs:anyURI"/> ?
3285          <operation rel="delete" href="xs:anyURI"/> ?
3286      </operations>
3287      <xs:any>*
3288  </NetworkTemplate>

```

### 3.6.5 NetworkTemplateCollection serialization

3290 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 3291 JSON and XML:

3292 This Resource shall be serialized as follows:

#### JSON serialization:

```

3294 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkTemplateCollection",
3295     "id": string,
3296     "updated": string,
3297     "parent": string,
3298     "count": number,
3299     "networkTemplates": [
3300         { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkTemplate",
3301             "id": string,
3302             ... remaining NetworkTemplate attributes ...
3303         }, +
3304     ], ?
3305     "operations": [ { "rel": "add", "href": string } ? ]
3306     ...
3307 }

```

#### XML serialization:

```

3309 <Collection
3310     resourceURI="http://schemas.dmtf.org/cimi/2/NetworkTemplateCollection"
3311     xmlns="http://schemas.dmtf.org/cimi/2">
3312     <id> xs:anyURI </id>
3313     <updated> xs:dateTime </updated>

```

```

3314      <parent> xs:anyURI </parent>
3315      <count> xs:integer </count>
3316      <networkTemplates>
3317          <NetworkTemplate>
3318              <id> xs:anyURI </id>
3319              ... remaining NetworkTemplate attributes ...
3320          </NetworkTemplate> *
3321      </networkTemplates>
3322      <operations>
3323          <operation rel="add" href="xs:anyURI"/> ?
3324      </operations>
3325      <xs:any>*
3326  </Collection>

```

### 3327 3.6.6 NetworkService serialization

3328 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 3329 JSON and XML.

3330 **JSON media type:** application/json

3331 **JSON serialization:**

```

3332 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkService",
3333     "id": string,
3334     "name": string, ?
3335     "description": string, ?
3336     "created": string, ?
3337     "updated": string, ?
3338     "parent": string, ?
3339     "properties": { string: string, + }, ?
3340     "state": string,
3341     "type": string,
3342     "endpoints": { "href": string }, ?
3343     "segments": { "href": string }, ?
3344
3345     .."policies": { string: string, + }, ?
3346
3347     "meters": { "href": string }, ?
3348     "eventLog": { "href": string }, ?
3349     "operations": [
3350         { "rel": "edit", "href": string }, ?
3351         { "rel": "delete", "href": string }, ?
3352         { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string }, ?

```

```
3353 { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string } ?  
3354 ] ?  
3355 ...  
3356 }
```

3357 **XML media type:** application/xml

3358 **XML serialization:**

```
3359 <NetworkService xmlns="http://schemas.dmtf.org/cimi/2">  
3360   <id> xs:anyURI </id>  
3361   <name> xs:string </name> ?  
3362   <description> xs:string </description> ?  
3363   <created> xs:dateTime </created> ?  
3364   <updated> xs:dateTime </updated> ?  
3365   <parent> xs:anyURI </parent> ?  
3366   <properties>  
3367     <property key="xs:string"> xs:string </property> *  
3368   </properties> ?  
3369   <state> xs:string </state>  
3370   <type> xs:string </type>  
3371   <endpoints href="xs:anyURI"/> *  
3372   <segments href="xs:anyURI"/> *  
3373  
3374   <policies key="xs:string"> xs:string </policies> *  
3375  
3376   <meters href="xs:anyURI"/> ?  
3377   <eventLog" href="xs:anyURI"/> ?  
3378   <operations>  
3379     <operation rel="edit" href="xs:anyURI"/> ?  
3380     <operation rel="delete" href="xs:anyURI"/> ?  
3381     <operation rel="http://schemas.dmtf.org/cimi/2/action/start"  
3382 href="xs:anyURI"/> ?  
3383     <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"  
3384 href="xs:anyURI"/> ?  
3385   </operations>  
3386   <xs:any*>  
3387 </NetworkService>
```

### 3388 3.6.7 NetworkServiceCollection serialization

3389 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
3390 JSON and XML.

3391 This Resource shall be serialized as follows:

3392 **JSON serialization:**

```

3393     { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkServiceCollection",
3394         "id": string,
3395         "updated": string,
3396         "parent": string,
3397         "count": number,
3398         "services": [
3399             { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkService",
3400                 "id": string,
3401                 ... remaining NetworkService attributes ...
3402             }, +
3403         ], ?
3404         "operations": [ { "rel": "add", "href": string } ? ]
3405         ...
3406     }

```

3407 **XML serialization:**

```

3408 <Collection
3409     resourceURI="http://schemas.dmtf.org/cimi/2/NetworkServiceCollection"
3410         xmlns="http://schemas.dmtf.org/cimi/2">
3411             <id> xs:anyURI </id>
3412             <updated> xs:dateTime </updated>
3413             <parent> xs:anyURI </parent>
3414             <count> xs:integer </count>
3415             <services>
3416                 <NetworkService>
3417                     <id> xs:anyURI </id>
3418                     ... remaining NetworkService attributes ...
3419                 </NetworkService> *
3420             </services>
3421             <operations>
3422                 <operation rel="add" href="xs:anyURI"/> ?
3423             </operations>
3424             <xs:any>*
3425         </Collection>

```

**3426 3.6.8 NetworkServiceTemplate serialization**

3427 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
3428 JSON and XML.

3429 **JSON media type:** application/json

3430 **JSON serialization:**

```
3431 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkServiceTemplate",
3432   "id": string,
3433   "name": string, ?
3434   "description": string, ?
3435   "created": string, ?
3436   "updated": string, ?
3437   "parent": string, ?
3438   "properties": { string: string, + }, ?
3439   "network": { "href": string }, ?
3440   "initialState": string, ?
3441   "type": string,
3442   "endpoints": { "href": string }, *
3443   "segments": { "href": string }, *
3444
3445   .."policies": { string: string, + }, ?
3446
3447   "meterTemplates": [
3448     { "href": string, ?
3449       ... MeterTemplate attributes ... ?
3450     }, *
3451   ], ?
3452   "eventLogTemplate": {
3453     "href": string, ?
3454     ... EventLogTemplate attributes ... ?
3455   }, ?
3456   "operations": [
3457     { "rel": "edit", "href": string }, ?
3458     { "rel": "delete", "href": string } ?
3459   ] ?
3460   ...
3461 }
```

3462 **XML media type:** application/xml

3463 **XML serialization:**

```
<NetworkServiceTemplate xmlns="http://schemas.dmtf.org/cimi/2">
```

```
3465      <id> xs:anyURI </id>
3466      <name> xs:string </name> ?
3467      <description> xs:string </description> ?
3468      <created> xs:dateTime </created> ?
3469      <updated> xs:dateTime </updated> ?
3470      <parent> xs:anyURI </parent> ?
3471      <properties>
3472          <property key="xs:string"> xs:string </property> *
3473      </properties> ?
3474      <network href="xs:anyURI"/> ?
3475      <initialState> xs:string </initialState> ?
3476      <type> xs:string </type>
3477      <endpoints>
3478          <ProtocolEndpoint href="xs:anyURI"/> *
3479      </endpoints>
3480      <segments>
3481          <ProtocolSegment href="xs:anyURI"/> *
3482      </segments>
3483      <policies>
3484          <policy key="xs:string"> xs:string </policies> *
3485      </policies>
3486
3487      <meterTemplates>
3488          <MeterTemplate href="xs:anyURI"? >
3489              ... MeterTemplate attributes ...
3490          </MeterTemplate> *
3491      </meterTemplates>
3492      <eventLogTemplate href="xs:anyURI"? >
3493          ... EventLogTemplate attributes ...
3494      </eventLogTemplate> ?
3495      <operations>
3496          <operation rel="edit" href="xs:anyURI"/> ?
3497          <operation rel="delete" href="xs:anyURI"/> ?
3498      </operations>
3499      <xss:any>*
3500      </NetworkServiceTemplate>
```

**3501    3.6.9 NetworkServiceTemplateCollection serialization**

3502    The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
3503    JSON and XML.

3504    This Resource shall be serialized as follows:

**3505    JSON serialization:**

```
3506 { "resourceURI":  
3507     "http://schemas.dmtf.org/cimi/2/NetworkServiceTemplateCollection",  
3508     "id": string,  
3509     "updated": string,  
3510     "parent": string,  
3511     "count": number,  
3512     "networkServiceTemplates": [  
3513         { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkServiceTemplate",  
3514             "id": string,  
3515             ... remaining NetworkServiceTemplate attributes ...  
3516         }, +  
3517     ], ?  
3518     "operations": [ { "rel": "add", "href": string } ? ]  
3519     ...  
3520 }
```

**3521    XML serialization:**

```
3522 <Collection  
3523     resourceURI="http://schemas.dmtf.org/cimi/2/NetworkServiceTemplateCollection"  
3524     xmlns="http://schemas.dmtf.org/cimi/2">  
3525     <id> xs:anyURI </id>  
3526     <updated> xs:dateTime </updated>  
3527     <parent> xs:anyURI </parent>  
3528     <count> xs:integer </count>  
3529     <networkServiceTemplates>  
3530         <NetworkServiceTemplate>  
3531             <id> xs:anyURI </id>  
3532             ... remaining NetworkServiceTemplate attributes ...  
3533             </NetworkServiceTemplate> *  
3534         </networkServiceTemplates>  
3535         <operations>  
3536             <operation rel="add" href="xs:anyURI"/> ?  
3537         </operations>  
3538         <xs:any>*  
3539     </Collection>
```

3540 **3.6.10 ProtocolSegment serialization**

3541 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 3542 JSON and XML.

3543 **JSON media type:** application/json

3544 **JSON serialization:**

```

3545   { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolSegment",
3546     "id": string,
3547     "name": string, ?
3548     "description": string, ?
3549     "created": string, ?
3550     "updated": string, ?
3551     "parent": string, ?
3552     "properties": { string: string, + }, ?
3553     "state": string,
3554     "protocol": string,
3555     "noDefaultRouting": boolean,
3556     "endpoints": { "href": string },
3557     .."parameters": { string: string, + }, ?
3558     "meters": { "href": string }, ?
3559     "eventLog": { "href": string }, ?
3560     "operations": [
3561       { "rel": "edit", "href": string }, ?
3562       { "rel": "delete", "href": string }, ?
3563       { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string }, ?
3564       { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string } ?
3565     ] ?
3566     ...
3567   }

```

3568 **XML media type:** application/xml

3569 **XML serialization:**

```

3570 <ProtocolSegment xmlns="http://schemas.dmtf.org/cimi/2">
3571   <id> xs:anyURI </id>
3572   <name> xs:string </name> ?
3573   <description> xs:string </description> ?
3574   <created> xs:dateTime </created> ?
3575   <updated> xs:dateTime </updated> ?
3576   <parent> xs:anyURI </parent> ?
3577   <properties>
3578     <property key="xs:string"> xs:string </property> *

```

```

3579      </properties> ?
3580      <state> xs:string </state>
3581      <protocol> xs:string </protocol>
3582      <noDefaultRouting> xs:boolean </noDefaultRouting> ?
3583      <endpoints href="xs:anyURI"/>
3584      <parameters>
3585          <parameter key="xs:string"> xs:string </parameters> *
3586      </parameters>
3587      <meters href="xs:anyURI"/> ?
3588      <eventLog href="xs:anyURI"/> ?
3589      <operations>
3590          <operation rel="edit" href="xs:anyURI"/> ?
3591          <operation rel="delete" href="xs:anyURI"/> ?
3592          <operation rel="http://schemas.dmtf.org/cimi/2/action/start"
3593 href="xs:anyURI"/> ?
3594          <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
3595 href="xs:anyURI"/> ?
3596      </operations>
3597      <xs:any>*
3598  </ProtocolSegment>

```

### 3.6.11 ProtocolSegmentCollection serialization

3600 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 3601 JSON and XML.

3602 This Resource shall be serialized as follows:

#### 3603 JSON serialization:

```

3604 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolSegmentCollection",
3605   "id": string,
3606   "updated": string,
3607   "parent": string,
3608   "count": number,
3609   "segments": [
3610     { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolSegment",
3611       "id": string,
3612       ... remaining ProtocolSegment attributes ...
3613     }, +
3614   ], ?
3615   "operations": [ { "rel": "add", "href": string } ? ]
3616   ...
3617 }

```

3618 **XML serialization:**

```

3619 <Collection
3620   resourceURI="http://schemas.dmtf.org/cimi/2/ProtocolSegmentCollection"
3621     xmlns="http://schemas.dmtf.org/cimi/2">
3622       <id> xs:anyURI </id>
3623       <updated> xs:dateTime </updated>
3624       <parent> xs:anyURI </parent>
3625       <count> xs:integer </count>
3626       <segments>
3627         <ProtocolSegment>
3628           <id> xs:anyURI </id>
3629           ... remaining ProtocolSegment attributes ...
3630           </ProtocolSegment> *
3631       </segments>
3632       <operations>
3633         <operation rel="add" href="xs:anyURI"/> ?
3634       </operations>
3635       <xss:any>*
3636     </Collection>
```

3637 **3.6.12 ProtocolSegmentTemplate serialization**

3638 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 3639 JSON and XML.

3640 **JSON media type:** application/json

3641 **JSON serialization:**

```

3642   { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolSegmentTemplate",
3643     "id": string,
3644     "name": string, ?
3645     "description": string, ?
3646     "created": string, ?
3647     "updated": string, ?
3648     "parent": string, ?
3649     "properties": { string: string, + }, ?
3650     "network": { "href": string }, ?
3651     "initialState": string, ?
3652     "protocol": string,
3653     "noDefaultRouting": boolean,
3654     "endpoints": { "href": string }, *
3655     "endpointTemplates": [
3656       { "href": string, ?
```

```

3657         ... ProtocolEndpointTemplate attributes ... ?
3658     }, *
3659     ], ?
3660     .."parameters": { string: string, + }, ?
3661     "meterTemplates": [
3662         { "href": string, ?
3663             ... MeterTemplate attributes ... ?
3664         }, *
3665     ], ?
3666     "eventLogTemplate": {
3667         "href": string, ?
3668         ... EventLogTemplate attributes ... ?
3669     }, ?
3670     "operations": [
3671         { "rel": "edit", "href": string }, ?
3672         { "rel": "delete", "href": string } ?
3673     ] ?
3674     ...
3675 }

```

3676 **XML media type:** application/xml

3677 **XML serialization:**

```

3678 <ProtocolSegmentTemplate xmlns="http://schemas.dmtf.org/cimi/2">
3679     <id> xs:anyURI </id>
3680     <name> xs:string </name> ?
3681     <description> xs:string </description> ?
3682     <created> xs:dateTime </created> ?
3683     <updated> xs:dateTime </updated> ?
3684     <parent> xs:anyURI </parent> ?
3685     <properties>
3686         <property key="xs:string"> xs:string </property> *
3687     </properties> ?
3688     <network href="xs:anyURI"/> ?
3689     <initialState> xs:string </initialState> ?
3690     <protocol> xs:string </protocol>
3691     <noDefaultRouting> xs:boolean </noDefaultRouting >
3692     <endpoints> *
3693         <ProtocolEndpoint href="xs:anyURI"/> *
3694     </endpoints>
3695     <endpointTemplates>

```

```

3696      <ProtocolEndpointTemplate href="xs:anyURI"? >
3697          ... ProtocolEndpointTemplate attributes ... ?
3698      </ProtocolEndpointTemplate> *
3699  </endpointTemplates>
3700  <parameters>
3701      <parameter key="xs:string"> xs:string </parameters> *
3702  </parameters>
3703  <meterTemplates>
3704      <MeterTemplate href="xs:anyURI"? >
3705          ... MeterTemplate attributes ... ?
3706      </MeterTemplate> *
3707  </meterTemplates>
3708  <eventLogTemplate href="xs:anyURI"? >
3709          ... EventLogTemplate attributes ... ?
3710  </eventLogTemplate> ?
3711  <operations>
3712      <operation rel="edit" href="xs:anyURI"/> ?
3713      <operation rel="delete" href="xs:anyURI"/> ?
3714  </operations>
3715  <xss:any>*
3716  </ProtocolSegmentTemplate>

```

### 3.6.13 ProtocolSegmentTemplateCollection serialization

3718 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 3719 JSON and XML.

3720 This Resource shall be serialized as follows:

#### JSON serialization:

```

3722  { "resourceURI":
3723      "http://schemas.dmtf.org/cimi/2/ProtocolSegmentTemplateCollection",
3724      "id": string,
3725      "updated": string,
3726      "parent": string,
3727      "count": number,
3728      "protocolSegmentTemplates": [
3729          { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolSegmentTemplate",
3730              "id": string,
3731              ... remaining ProtocolSegmentTemplate attributes ...
3732          }, +
3733      ], ?
3734      "operations": [ { "rel": "add", "href": string } ? ]

```

```
3735     ...
3736 }
3737 XML serialization:
3738 <Collection
3739   resourceURI="http://schemas.dmtf.org/cimi/2/ProtocolSegmentTemplateCollection"
3740     xmlns="http://schemas.dmtf.org/cimi/2"
3741     <id> xs:anyURI </id>
3742     <updated> xs:dateTime </updated>
3743     <parent> xs:anyURI </parent>
3744     <count> xs:integer </count>
3745     <protocolSegmentTemplates>
3746       <ProtocolSegmentTemplate>
3747         <id> xs:anyURI </id>
3748         ... remaining ProtocolSegmentTemplate attributes ...
3749         </ProtocolSegmentTemplate> *
3750       </protocolSegmentTemplates>
3751     <operations>
3752       <operation rel="add" href="xs:anyURI"/> ?
3753     </operations>
3754     <xss:any>
3755   </Collection>
```

### 3.6.14 ProtocolEndpoint serialization

3758 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
3759 JSON and XML.

3760 **JSON media type:** application/json

3761 **JSON serialization:**

```
3762 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolEndpoint",
3763   "id": string,
3764   "name": string, ?
3765   "description": string, ?
3766   "created": string, ?
3767   "updated": string, ?
3768   "parent": string, ?
3769   "properties": { string: string, + }, ?
3770   "state": string,
3771   "protocol": string, ?
3772   "address": string,
3773   "origin": string,
```

```

3774     "interface": { "href": string },
3775     .."parameters": { string: string, + }, ?
3776     "meters": { "href": string }, ?
3777     "eventLog": { "href": string }, ?
3778     "operations": [
3779         { "rel": "edit", "href": string }, ?
3780         { "rel": "delete", "href": string }, ?
3781         { "rel": "http://schemas.dmtf.org/cimi/2/action/enable", "href": string },
3782     ?
3783         { "rel": "http://schemas.dmtf.org/cimi/2/action/disable", "href": string }
3784     ?
3785     ] ?
3786     ...
3787 }
```

3788 **XML media type:** application/xml

3789 **XML serialization:**

```

3790 <ProtocolEndpoint xmlns="http://schemas.dmtf.org/cimi/2">
3791     <id> xs:anyURI </id>
3792     <name> xs:string </name> ?
3793     <description> xs:string </description> ?
3794     <created> xs:dateTime </created> ?
3795     <updated> xs:dateTime </updated> ?
3796     <parent> xs:anyURI </parent> ?
3797     <properties>
3798         <property key="xs:string"> xs:string </property> *
3799     </properties> ?
3800     <state> xs:string </state>
3801     <protocol> xs:string </protocol> ?
3802     <address> xs:string </address>
3803     <origin> xs:string </origin>
3804     <interface href="xs:anyURI"/>
3805     <parameters>
3806         <parameters key="xs:string"> xs:string </parameters> *
3807     </parameters>
3808     <meters href="xs:anyURI"/> ?
3809     <eventLog href="xs:anyURI"/> ?
3810     <operations>
3811         <operation rel="edit" href="xs:anyURI"/> ?
3812         <operation rel="delete" href="xs:anyURI"/> ?
3813         <operation rel="http://schemas.dmtf.org/cimi/2/action/enable"
3814 href="xs:anyURI"/> ?
```

```
3815      <operation rel="http://schemas.dmtf.org/cimi/2/action/disable"
3816      href="xs:anyURI"/> ?
3817
3818      </operations>
3819      <xss:any>*
3820      </ProtocolEndpoint>
```

### 3.6.15 ProtocolEndpointCollection serialization

3821 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
3822 JSON and XML.

3823 This Resource shall be serialized as follows:

#### 3824 **JSON serialization:**

```
3825 { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolEndpointCollection",
3826   "id": string,
3827   "updated": string,
3828   "parent": string,
3829   "count": number,
3830   "endpoints": [
3831     { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolEndpoint",
3832       "id": string,
3833       ... remaining ProtocolEndpoint attributes ...
3834     }, +
3835   ], ?
3836   "operations": [ { "rel": "add", "href": string } ? ]
3837   ...
3838 }
```

#### 3839 **XML serialization:**

```
3840 <Collection
3841 resourceURI="http://schemas.dmtf.org/cimi/2/ProtocolEndpointCollection"
3842   xmlns="http://schemas.dmtf.org/cimi/2">
3843   <id> xs:anyURI </id>
3844   <updated> xs:dateTime </updated>
3845   <parent> xs:anyURI </parent>
3846   <count> xs:integer </count>
3847   <endpoints>
3848     <ProtocolEndpoint>
3849       <id> xs:anyURI </id>
3850       ... remaining ProtocolEndpoint attributes ...
3851       </ProtocolEndpoint> *
3852     </endpoints>
3853   <operations>
```

```

3854     <operation rel="add" href="xs:anyURI"/> ?
3855     </operations>
3856     <xss:any>*
3857   </Collection>

```

### 3.6.16 ProtocolEndpointTemplate serialization

3859 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 3860 JSON and XML.

3861 Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-  
 3862 schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

3863 **JSON media type:** application/json

3864 **JSON serialization:**

```

3865   { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolEndpointTemplate",
3866     "id": string,
3867     "name": string, ?
3868     "description": string, ?
3869     "created": string, ?
3870     "updated": string, ?
3871     "parent": string, ?
3872     "properties": { string: string, + }, ?
3873     "initialState": string, ?
3874     "address": string, ?
3875     "origin": string,
3876     "interface": { "href": string }, ?
3877     .."parameters": { string: string, + }, ?
3878     "meterTemplates": [
3879       { "href": string, ?
3880         ... MeterTemplate attributes ... ?
3881       }, *
3882     ], ?
3883     "eventLogTemplate": {
3884       "href": string, ?
3885       ... EventLogTemplate attributes ... ?
3886     }, ?
3887     "operations": [
3888       { "rel": "edit", "href": string }, ?
3889       { "rel": "delete", "href": string } ?
3890     ] ?
3891     ...
3892   }

```

3893 **XML media type:** application/xml

3894 **XML serialization:**

```

3895     <ProtocolEndpointTemplate xmlns="http://schemas.dmtf.org/cimi/2">
3896         <id> xs:anyURI </id>
3897         <name> xs:string </name> ?
3898         <description> xs:string </description> ?
3899         <created> xs:dateTime </created> ?
3900         <updated> xs:dateTime </updated> ?
3901         <parent> xs:anyURI </parent> ?
3902         <properties>
3903             <property key="xs:string"> xs:string </property> *
3904         </properties> ?
3905         <initialState> xs:string </initialState> ?
3906         <address> xs:string </address> ?
3907         <origin> xs:string </origin>
3908         <interface href="xs:anyURI"/> ?
3909         <parameters key="xs:string"> xs:string </parameters> *
3910         <meterTemplates>
3911             <MeterTemplate href="xs:anyURI"? >
3912                 ... MeterTemplate attributes ...
3913             </MeterTemplate> *
3914         </meterTemplates>
3915         <eventLogTemplate href="xs:anyURI"? >
3916             ... EventLogTemplate attributes ...
3917         </eventLogTemplate> ?
3918         <operations>
3919             <operation rel="edit" href="xs:anyURI"/> ?
3920             <operation rel="delete" href="xs:anyURI"/> ?
3921         </operations>
3922         <xs:any>*
3923     </ProtocolEndpointTemplate>
```

3924 **3.6.17 ProtocolEndpointTemplateCollection serialization**

3925 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 3926 JSON and XML.

3927 This Resource shall be serialized as follows:

3928 **JSON serialization:**

```

3929     { "resourceURI":
3930         "http://schemas.dmtf.org/cimi/2/ProtocolEndpointTemplateCollection",
3931         "id": string,
```

```

3932     "updated": string,
3933     "parent": string,
3934     "count": number,
3935     "protocolEndpointTemplates": [
3936         { "resourceURI": "http://schemas.dmtf.org/cimi/2/ProtocolEndpointTemplate",
3937             "id": string,
3938             ... remaining ProtocolEndpointTemplate attributes ...
3939         }, +
3940     ], ?
3941     "operations": [ { "rel": "add", "href": string } ? ]
3942     ...
3943 }

```

#### 3944 XML serialization:

```

3945 <Collection
3946
3947     resourceURI="http://schemas.dmtf.org/cimi/2/ProtocolEndpointTemplateCollection"
3948         xmlns="http://schemas.dmtf.org/cimi/2"
3949         <id> xs:anyURI </id>
3950         <updated> xs:dateTime </updated>
3951         <parent> xs:anyURI </parent>
3952         <count> xs:integer </count>
3953         <protocolEndpointTemplates>
3954             <ProtocolEndpointTemplate>
3955                 <id> xs:anyURI </id>
3956                 ... remaining ProtocolEndpointTemplate attributes ...
3957             </ProtocolEndpointTemplate> *
3958         </protocolEndpointTemplates>
3959         <operations>
3960             <operation rel="add" href="xs:anyURI"/> ?
3961         </operations>
3962         <xs:any>*
3963     </Collection>

```

#### 3964 3.6.18 NetworkInterface serialization

3965 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 3966 JSON and XML.

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

#### 3968 JSON serialization:

```

3969     { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkInterface",
3970         "id": string,

```

```
3971     "name": string, ?
3972     "description": string, ?
3973     "created": string, ?
3974     "updated": string, ?
3975     "parent": string, ?
3976     "properties": { string: string, + }, ?
3977     "state": string,
3978     "endpoints": { "href": string }, ?
3979     "speed": number, ?
3980     "mtu": number ?,
3981     "meters": { "href": string }, ?
3982     "eventLog": { "href": string }, ?
3983     "operations": [
3984         { "rel": "edit", "href": string }, ?
3985         { "rel": "delete", "href": string }, ?
3986         { "rel": "http://schemas.dmtf.org/cimi/2/action/enable", "href": string },
3987     ?
3988         { "rel": "http://schemas.dmtf.org/cimi/2/action/disable", "href": string }
3989     ?
3990     ] ?
3991     ...
3992 }
```

3993 **XML media type:** application/xml

3994 **XML serialization:**

```
3995 <NetworkInterface xmlns="http://schemas.dmtf.org/cimi/2">
3996     <id> xs:anyURI </id>
3997     <name> xs:string </name> ?
3998     <description> xs:string </description> ?
3999     <created> xs:dateTime </created> ?
4000     <updated> xs:dateTime </updated> ?
4001     <parent> xs:anyURI </parent> ?
4002     <properties>
4003         <property key="xs:string"> xs:string </property> *
4004     </properties> ?
4005     <state> xs:string </state>
4006     <endpoint href="xs:anyURI"/> ?
4007     <speed> xs:integer </speed> ?
4008     <mtu> xs:integer </mtu> ?
4009     <meters href="xs:anyURI"/> ?
4010     <eventLog" href="xs:anyURI"/> ?
```

```

4011      <operations>
4012          <operation rel="edit" href="xs:anyURI"/> ?
4013          <operation rel="delete" href="xs:anyURI"/> ?
4014          <operation rel="http://schemas.dmtf.org/cimi/2/action/enable"
4015          href="xs:anyURI"/> ?
4016          <operation rel="http://schemas.dmtf.org/cimi/2/action/disable"
4017          href="xs:anyURI"/> ?
4018      </operations>
4019      <xs:any>*
4020  </NetworkInterface>

```

### 4021 **3.6.19 NetworkInterfaceCollection serialization**

4022 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
4023 JSON and XML.

4024 This Resource shall be serialized as follows:

#### 4025 **JSON serialization:**

```

4026 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkInterfaceCollection",
4027     "id": string,
4028     "updated": string,
4029     "parent": string,
4030     "count": number,
4031     "interfaces": [
4032         { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkInterface",
4033             "id": string,
4034             ... remaining NetworkInterface attributes ...
4035         }, +
4036     ], ?
4037     "operations": [ { "rel": "add", "href": string } ? ]
4038     ...
4039 }

```

#### 4040 **XML serialization:**

```

4041 <Collection
4042 resourceURI="http://schemas.dmtf.org/cimi/2/NetworkInterfaceCollection"
4043     xmlns="http://schemas.dmtf.org/cimi/2">
4044     <id> xs:anyURI </id>
4045     <updated> xs:dateTime </updated>
4046     <parent> xs:anyURI </parent>
4047     <count> xs:integer </count>
4048     <interfaces>
4049         <NetworkInterface>
4050             <id> xs:anyURI </id>

```

```

4051     ... remaining NetworkInterface attributes ...
4052     </NetworkInterface> *
4053   </interfaces>
4054   <operations>
4055     <operation rel="add" href="xs:anyURI"/> ?
4056   </operations>
4057   <xs:any>*
4058 </Collection>
```

### 4059 3.6.20 NetworkInterfaceTemplate serialization

4060 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 4061 JSON and XML.

4062 Both Consumer and Provider shall serialize this Resource as described below. The following pseudo-  
 4063 schemas (see notation in 1.3) describe the serialization of the Resource in both JSON and XML.

4064 **JSON media type:** application/json

4065 **JSON serialization:**

```

4066 { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkInterfaceTemplate",
4067   "id": string,
4068   "name": string, ?
4069   "description": string, ?
4070   "created": string, ?
4071   "updated": string, ?
4072   "parent": string, ?
4073   "properties": { string: string, + }, ?
4074   "initialState": string, ?
4075   "endpoints": { "href": string }, ?
4076   "speed": number, ?
4077   "mtu": number ?,
4078   "meterTemplates": [
4079     { "href": string, ?
4080       ... MeterTemplate attributes ... ?
4081     }, *
4082   ], ?
4083   "eventLogTemplate": {
4084     "href": string, ?
4085     ... EventLogTemplate attributes ... ?
4086   }, ?
4087   "operations": [
4088     { "rel": "edit", "href": string }, ?
4089     { "rel": "delete", "href": string } ?
```

```
4090     ] ?
4091     ...
4092 }
```

4093 **XML media type:** application/xml

4094 **XML serialization:**

```
4095 <NetworkInterfaceTemplate xmlns="http://schemas.dmtf.org/cimi/2">
4096   <id> xs:anyURI </id>
4097   <name> xs:string </name> ?
4098   <description> xs:string </description> ?
4099   <created> xs:dateTime </created> ?
4100   <updated> xs:dateTime </updated> ?
4101   <parent> xs:anyURI </parent> ?
4102   <properties>
4103     <property key="xs:string"> xs:string </property> *
4104   </properties> ?
4105   <initialState> xs:string </initialState> ?
4106   <endpoint href="xs:anyURI"/> ?
4107   <speed> xs:integer </speed> ?
4108   <mtu> xs:integer </mtu> ?
4109   <meterTemplates>
4110     <MeterTemplate href="xs:anyURI"? >
4111       ... MeterTemplate attributes ... ?
4112     </MeterTemplate> *
4113   </meterTemplates>
4114   <eventLogTemplate href="xs:anyURI"? >
4115     ... EventLogTemplate attributes ... ?
4116   </eventLogTemplate> ?
4117   <operations>
4118     <operation rel="edit" href="xs:anyURI"/> ?
4119     <operation rel="delete" href="xs:anyURI"/> ?
4120   </operations>
4121   <xs:any>*
4122 </NetworkInterfaceTemplate>
```

4123 **3.6.21 NetworkInterfaceTemplateCollection serialization**

4124 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 4125 JSON and XML.

4126 This Resource shall be serialized as follows:

4127 **JSON serialization:**

```

4128     { "resourceURI":
4129         "http://schemas.dmtf.org/cimi/2/NetworkInterfaceTemplateCollection",
4130         "id": string,
4131         "updated": string,
4132         "parent": string,
4133         "count": number,
4134         "networkInterfaceTemplates": [
4135             { "resourceURI": "http://schemas.dmtf.org/cimi/2/NetworkInterfaceTemplate",
4136                 "id": string,
4137                 ... remaining NetworkInterfaceTemplate attributes ...
4138             }, +
4139         ], ?
4140         "operations": [ { "rel": "add", "href": string } ? ]
4141         ...
4142     }

```

4143 **XML serialization:**

```

4144     <Collection
4145
4146     resourceURI="http://schemas.dmtf.org/cimi/2/NetworkInterfaceTemplateCollection"
4147         xmlns="http://schemas.dmtf.org/cimi/2">
4148         <id> xs:anyURI </id>
4149         <updated> xs:dateTime </updated>
4150         <parent> xs:anyURI </parent>
4151         <count> xs:integer </count>
4152         <networkInterfaceTemplates>
4153             <NetworkInterfaceTemplate>
4154                 <id> xs:anyURI </id>
4155                 ... remaining NetworkInterfaceTemplate attributes ...
4156                 </NetworkInterfaceTemplate> *
4157             </networkInterfaceTemplates>
4158             <operations>
4159                 <operation rel="add" href="xs:anyURI"/> ?
4160             </operations>
4161             <xs:any>*
4162         </Collection>

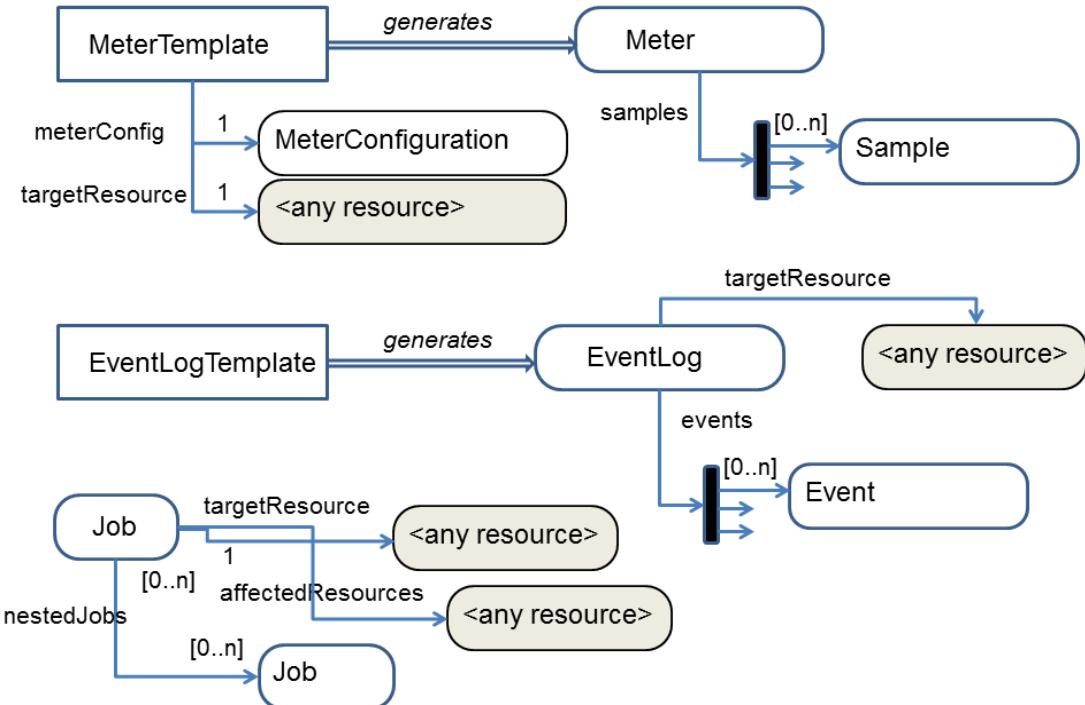
```

## 4163 **3.7 Monitoring Resources group**

### 4164 **3.7.1 Monitoring Resources and their relationships**

4165 The group of Monitoring-related Resources is represented in Figure 13, with other associated Resources:

## Monitoring Resources group



4166

4167

**Figure 13 – Monitoring Resources group**

### 4168 3.7.2 Job serialization

4169 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both JSON  
4170 and XML.

4171    **JSON media type:** application/json

4172 JSON serialization:

```
4173 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Job",
4174     "id": string,
4175     "name": string, ?
4176     "description": string, ?
4177     "created": string, ?
4178     "updated": string, ?
4179     "parent": string, ?
4180     "properties": { string: string, + }, ?
4181     "state": string,
4182     "targetResource": { "href": string },
4183     "affectedResources": [ { "href": string }, + ],
4184     "action": string,
4185     "returnCode": number,
```

```
4186     "progress": number,  
4187     "statusMessage": string,  
4188     "timeOfStatusChange": date,  
4189     "parentJob": { "href": string }, ?  
4190     "nestedJobs": [  
4191         { "href": string }, +  
4192     ], ?  
4193     "operations": [  
4194         { "rel": "edit", "href": string }, ?  
4195         { "rel": "delete", "href": string }, ?  
4196         { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string } ?  
4197     ] ?  
4198     ...  
4199 }
```

4200 **XML media type:** application/xml

4201 **XML serialization:**

```
4202 <Job xmlns="http://schemas.dmtf.org/cimi/2">  
4203     <id> xs:anyURI </id>  
4204     <name> xs:string </name> ?  
4205     <description> xs:string </description> ?  
4206     <created> xs:dateTime </created> ?  
4207     <updated> xs:dateTime </updated> ?  
4208     <parent> xs:anyURI </parent> ?  
4209     <properties>  
4210         <property key="xs:string"> xs:string </property> *  
4211     </properties> ?  
4212     <state> xs:string </state>  
4213     <targetResource href="xs:anyURI"/>  
4214     <affectedResources>  
4215         <affectedResource href="xs:anyURI"/> +  
4216     </affectedResources>  
4217     <action> xs:anyURI </action>  
4218     <returnCode> xs:integer </returnCode>  
4219     <progress> xs:integer <progress>  
4220     <statusMessage> xs:string </statusMessage>  
4221     <timeOfStatusChange> xs:dateTime </timeOfStatusChange>  
4222     <parentJob href="xs:anyURI"/> ?  
4223     <nestedJobs>  
4224         <nestedJob href="xs:anyURI"/> *
```

```

4225      </nestedJobs>
4226
4227      <operations>
4228          <operation rel="edit" href="xs:anyURI"/> ?
4229          <operation rel="delete" href="xs:anyURI"/> ?
4230          <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
4231              href="xs:anyURI"/> ?
4232
4233      </operations>
4234      <xs:any>*
4235  </Job>

```

### 4234 3.7.3 JobCollection serialization

4235 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 4236 JSON and XML.

4237 This Resource shall be serialized as follows:

#### 4238 JSON serialization:

```

4239  { "resourceURI": "http://schemas.dmtf.org/cimi/2/JobCollection",
4240      "id": string,
4241      "updated": string,
4242      "parent": string,
4243      "count": integer,
4244      "jobs": [
4245          { "resourceURI": "http://schemas.dmtf.org/cimi/2/Job",
4246              "id": string,
4247              ... remaining Job attributes ...
4248          }, +
4249      ] ?
4250      ...
4251  }

```

#### 4252 XML serialization:

```

4253  <Collection resourceURI="http://schemas.dmtf.org/cimi/2/JobCollection"
4254      xmlns="http://schemas.dmtf.org/cimi/2">
4255      <id> xs:anyURI </id>
4256      <updated> xs:dateTime </updated>
4257      <parent> xs:anyURI </parent>
4258      <count> xs:integer </count>
4259      <jobs>
4260          <Job>
4261              <id> xs:anyURI </id>
4262              ... remaining Job attributes ...
4263          </Job> *

```

```
4264      </jobs>
4265      <xs:any>*
```

### 4267 3.7.4 Meter serialization

4268 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
4269 JSON and XML.

4270 **JSON media type:** application/json

4271 **JSON serialization:**

```
4272 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Meter",
4273   "id": string,
4274   "name": string, ?
4275   "description": string, ?
4276   "created": string, ?
4277   "updated": string, ?
4278   "parent": string, ?
4279   "properties": { string: string, + }, ?
4280   "targetResource": { "href": string },
4281   "aspect": string,
4282   "units": string,
4283   "sampleInterval": number,
4284   "timeScope": string,
4285   "intervalDuration": string,
4286   "isContinuous": boolean,
4287   "samples": { "href": string }, ?
4288   "minValue": string, ?
4289   "maxValue": string, ?
4290   "stopTime": string, ?
4291   "expiresTime": string, ?
4292   "operations": [
4293     { "rel": "edit", "href": string }, ?
4294     { "rel": "delete", "href": string }, ?
4295     { "rel": "http://schemas.dmtf.org/cimi/2/action/start", "href": string }, ?
4296     { "rel": "http://schemas.dmtf.org/cimi/2/action/stop", "href": string } ?
4297   ] ?
4298   ...
4299 }
```

4300 **XML media type:** application/xml

4301 **XML serialization:**

```

4302 <Meter xmlns="http://schemas.dmtf.org/cimi/2">
4303   <id> xs:anyURI </id>
4304   <name> xs:string </name> ?
4305   <description> xs:string </description> ?
4306   <created> xs:dateTime </created> ?
4307   <updated> xs:dateTime </updated> ?
4308   <parent> xs:anyURI </parent> ?
4309   <properties>
4310     <property key="xs:string"> xs:string </property> *
4311   </properties> ?
4312   <targetResource href="xs:anyURI"/>
4313   <aspect> xs:anyURI </aspect>
4314   <units> xs:string </units>
4315   <sampleInterval> xs:integer </sampleInterval>
4316   <timeScope> xs:string </timeScope>
4317   <intervalDuration xs:duration </intervalDuration>
4318   <isContinuous> xs:boolean </isContinuous>
4319   <samples href="xs:anyURI"/> ?
4320   <minValue> xs:string </minValue> ?
4321   <maxValue> xs:string </maxValue> ?
4322   <stopTime> xs:dateTime </stopTime> ?
4323   <expiresTime> xs:dateTime </expiresTime> ?
4324   <operations>
4325     <operation rel="edit" href="xs:anyURI"/> ?
4326     <operation rel="delete" href="xs:anyURI"/> ?
4327     <operation rel="http://schemas.dmtf.org/cimi/2/action/start"
4328 href="xs:anyURI"/> ?
4329     <operation rel="http://schemas.dmtf.org/cimi/2/action/stop"
4330 href="xs:anyURI"/> ?
4331   </operations>
4332   <xs:any>*
4333 </Meter>
```

### 4334 3.7.5 MeterCollection serialization

4335 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 4336 JSON and XML.

4337 This Resource shall be serialized as follows:

4338 **JSON serialization:**

```

4339 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterCollection",
4340   "id": string,
4341   "updated": string,
4342   "parent": string,
4343   "count": number,
4344   "meters": [
4345     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Meter",
4346       "id": string,
4347       ... remaining Meter attributes ...
4348     }, +
4349   ], ?
4350   "operations": [ { "rel": "add", "href": string } ? ]
4351   ...
4352 }
```

4353 **XML serialization:**

```

4354 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/MeterCollection"
4355   xmlns="http://schemas.dmtf.org/cimi/2">
4356   <id> xs:anyURI </id>
4357   <updated> xs:dateTime </updated>
4358   <parent> xs:anyURI </parent>
4359   <count> xs:integer </count>
4360   <meters>
4361     <Meter>
4362       <id> xs:anyURI </id>
4363       ... remaining Meter attributes ...
4364     </Meter> *
4365   </meters>
4366   <operations>
4367     <operation rel="add" href="xs:anyURI"/> ?
4368   </operations>
4369   <xs:any>*
4370 </Collection>
```

### 4371 3.7.6 SampleCollection serialization

4372 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 4373 JSON and XML.

4374 **JSON serialization:**

```

4375 { "resourceURI": "http://schemas.dmtf.org/cimi/2/SampleCollection",
4376   "id": string,
```

```

4377     "updated": string,
4378     "parent": string,
4379     "count": number,
4380     "samples": [
4381       { "resourceURI": "http://schemas.dmtf.org/cimi/2/Sample",
4382         "id": string,
4383         "name": string, ?
4384         "description": string, ?
4385         "created": string, ?
4386         "updated": string, ?
4387         "properties": { string: string, + }, ?
4388         "timestamp": string,
4389         "value": string
4390         ...
4391       }, +
4392     ], ?
4393     ...
4394   }

```

#### 4395 XML serialization:

```

4396 <Collection
4397   resourceURI="http://schemas.dmtf.org/cimi/2/SampleCollection"
4398   xmlns="http://schemas.dmtf.org/cimi/2">
4399   <id> xs:anyURI </id>
4400   <count> xs:integer </count>
4401   <samples>
4402     <Sample>
4403       <id> xs:anyURI </id>
4404       <name> xs:string </name> ?
4405       <description> xs:string </description> ?
4406       <created> xs:dateTime </created> ?
4407       <updated> xs:dateTime </updated> ?
4408       <properties>
4409         <property key="xs:string"> xs:string </property> *
4410       </properties> ?
4411       <sample timestamp="xs:dateTime" value="xs:string"/>
4412       <xs:any>*
4413     </Sample> *
4414   </samples>
4415   <xs:any>*
4416 </Collection>

```

**4417 3.7.7 MeterTemplate serialization**

4418 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
4419 JSON and XML.

4420 **JSON media type:** application/json

4421 **JSON serialization:**

```
4422 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterTemplate",
4423   "id": string,
4424   "name": string, ?
4425   "description": string, ?
4426   "created": string, ?
4427   "updated": string, ?
4428   "parent": string, ?
4429   "properties": { string: string, + }, ?
4430   "targetResource": { string },
4431   "meterConfig": {
4432     "href": string | ... MeterConfiguration attributes ...
4433   },
4434   "operations": [
4435     { "rel": "edit", "href": string }, ?
4436     { "rel": "delete", "href": string } ?
4437   ] ?
4438   ...
4439 }
```

4440 **XML media type:** application/xml

4441 **XML serialization:**

```
4442 <MeterTemplate xmlns="http://schemas.dmtf.org/cimi/2">
4443   <id> xs:anyURI </id>
4444   <name> xs:string </name> ?
4445   <description> xs:string </description> ?
4446   <created> xs:dateTime </created> ?
4447   <updated> xs:dateTime </updated> ?
4448   <parent> xs:anyURI </parent> ?
4449   <properties>
4450     <property key="xs:string"> xs:string </property> *
4451   </properties> ?
4452   <targetResource href="xs:anyURI"/>
4453   <meterConfig href="xs:anyURI"?>
4454     ... MeterConfiguration attributes ... ?
4455   </meterConfig>
```

```

4456     <operations>
4457         <operation rel="edit" href="xs:anyURI"/> ?
4458         <operation rel="delete" href="xs:anyURI"/> ?
4459     </operations>
4460     <xss:any>*
4461 </MeterTemplate>
```

### 4462 3.7.8 MeterTemplateCollection serialization

4463 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
4464 JSON and XML.

4465 . This Resource shall be serialized as follows:

#### 4466 JSON serialization:

```

4467 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterTemplateCollection",
4468     "id": string,
4469     "updated": string,
4470     "parent": string,
4471     "count": number,
4472     "meterTemplates": [
4473         { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterTemplate",
4474             "id": string,
4475             ... remaining MeterTemplate attributes ...
4476         }, +
4477     ], ?
4478     "operations": [ { "rel": "add", "href": string } ? ]
4479     ...
4480 }
```

#### 4481 XML serialization:

```

4482 <Collection
4483     resourceURI="http://schemas.dmtf.org/cimi/2/MeterTemplateCollection"
4484     xmlns="http://schemas.dmtf.org/cimi/2">
4485     <id> xs:anyURI </id>
4486     <updated> xs:dateTime </updated>
4487     <parent> xs:anyURI </parent>
4488     <count> xs:integer </count>
4489     <meterTemplates>
4490         <MeterTemplate>
4491             <id> xs:anyURI </id>
4492             ... remaining MeterTemplate attributes ...
4493             </MeterTemplate> *
4494         </meterTemplates>
```

```

4495      <operations>
4496          <operation rel="add" href="xs:anyURI"/> ?
4497      </operations>
4498      <xs:any>*
4499  </Collection>
```

### 4500 3.7.9 MeterConfiguration serialization

4501 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 4502 JSON and XML.

4503 **JSON media type:** application/json

4504 **JSON serialization:**

```

4505 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterConfiguration",
4506     "id": string,
4507     "name": string, ?
4508     "description": string, ?
4509     "created": string, ?
4510     "updated": string, ?
4511     "parent": string, ?
4512     "properties": { string: string, + }, ?
4513     "associatedResources": [
4514         { "href": string }, +
4515     ], ?
4516     "aspect": string,
4517     "units": string,
4518     "sampleInterval": number,
4519     "timeScope": string,
4520     "intervalDuration": string,
4521     "isContinuous": boolean,
4522     "operations": [
4523         { "rel": "edit", "href": string }, ?
4524         { "rel": "delete", "href": string } ?
4525     ] ?
4526     ...
4527 }
```

4528 **XML media type:** application/xml

4529 **XML serialization:**

```

4530 <MeterConfiguration xmlns="http://schemas.dmtf.org/cimi/2">
4531     <id> xs:anyURI </id>
4532     <name> xs:string </name> ?
```

```

4533      <description> xs:string </description> ?
4534      <created> xs:dateTime </created> ?
4535      <updated> xs:dateTime </updated> ?
4536      <parent> xs:anyURI </parent> ?
4537      <properties>
4538          <property key="xs:string"> xs:string </property> *
4539      </properties> ?
4540      <associatedResources>
4541          < associatedResourcehref="xs:anyURI"/> *
4542      </associatedResources>
4543      <aspect> xs:anyURI </aspect>
4544      <units> xs:string </units>
4545      <sampleInterval> xs:integer </sampleInterval>
4546      <timeScope> xs:string </timeScope>
4547      <intervalDuration> xs:duration </intervalDuration>
4548      <isContinuous> xs:boolean </isContinuous>
4549      <operations>
4550          <operation rel="edit" href="xs:anyURI"/> ?
4551          <operation rel="delete" href="xs:anyURI"/> ?
4552      </operations>
4553      <xss:any>*
4554  </MeterConfiguration>

```

### 4555 3.7.10 MeterConfigurationCollection serialization

4556 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 4557 JSON and XML.

4558 This Resource shall be serialized as follows:

#### 4559 JSON serialization:

```

4560  { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterConfigurationCollection",
4561    "id": string,
4562    "updated": string,
4563    "parent": string,
4564    "count": number,
4565    "meterConfigurations": [
4566        { "resourceURI": "http://schemas.dmtf.org/cimi/2/MeterConfiguration",
4567          "id": string,
4568          ... remaining MeterConfiguration attributes ...
4569        }, +
4570    ], ?
4571    "operations": [ { "rel": "add", "href": string } ? ]

```

```
4572 ...
4573 }
```

**4574 XML serialization:**

```
4575 <Collection
4576     resourceURI="http://schemas.dmtf.org/cimi/2/MeterConfigurationCollection"
4577     xmlns="http://schemas.dmtf.org/cimi/2">
4578     <id> xs:anyURI </id>
4579     <updated> xs:dateTime </updated>
4580     <parent> xs:anyURI </parent>
4581     <count> xs:integer </count>
4582     <meterConfigurations>
4583         <MeterConfiguration>
4584             <id> xs:anyURI </id>
4585             ... remaining MeterConfiguration attributes ...
4586             </MeterConfiguration> *
4587         </meterConfigurations>
4588         <operations>
4589             <operation rel="add" href="xs:anyURI"/> ?
4590         </operations>
4591         <xs:any>*
4592     </Collection>
```

**4593 3.7.11 Event serialization**

4594 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
4595 JSON and XML.

**4596 JSON media type:** application/json**4597 JSON serialization:**

```
4598 { "resourceURI": "http://schemas.dmtf.org/cimi/2/Event",
4599   "id": string,
4600   "name": string, ?
4601   "description": string, ?
4602   "created": string, ?
4603   "updated": string, ?
4604   "parent": string, ?
4605   "properties": { string: string, + }, ?
4606   "timestamp": string,
4607   "type": string,
4608   "content": any, ?
4609   "outcome": string, ?
4610   "severity": string, ?
```

```

4611     "contact": string, ?
4612     ...
4613 }
```

4614 **XML media type:** application/xml

4615 **XML serialization:**

```

4616 <Event xmlns="http://schemas.dmtf.org/cimi/2">
4617   <id> xs:anyURI </id>
4618   <name> xs:string </name> ?
4619   <description> xs:string </description> ?
4620   <created> xs:dateTime </created> ?
4621   <updated> xs:dateTime </updated> ?
4622   <parent> xs:anyURI </parent> ?
4623   <properties>
4624     <property key="xs:string"> xs:string </property> *
4625   </properties> ?
4626   <timestamP> xs:dateTime </timestamP>
4627   <type> xs:string </type>
4628   <content> xs:any* </content> ?
4629   <outcome> xs:string </outcome> ?
4630   <severity> xs:string </severity> ?
4631   <contact> xs:string </contact> ?
4632   <xs:any>*
4633 </Event>
```

4634 The following pseudo-schemas describe the serialization of the "content" property for various types of  
4635 events:

4636 **"state" event:**

4637 **JSON serialization:**

```

4638   { "id": string,
4639     ...
4640     "type": "http://schemas.dmtf.org/cimi/2/event/state",
4641     "content": {
4642       "resName": string,
4643       "resource" : { "href" : string },
4644       "resType" : string,
4645       "state" : string,
4646       "previous" : string ?
4647     }
4648     ...
4649   }
```

4650 **XML serialization:**

```
4651 <Event xmlns="http://schemas.dmtf.org/cimi/2">
4652 ...
4653 <type> http://schemas.dmtf.org/cimi/2/event/state </type>
4654 <content>
4655   <resName> xs:string </resName>
4656   <resource href="xs:anyURI"/>
4657   <resType> xs:anyURI </resType>
4658   <state> xs:string </state>
4659   <previous> xs:string </previous> ?
4660 </content> ?
4661 ...
4662 </Event>
```

4664 **"alarm" event:**

4665 **JSON serialization:**

```
4666 { "id": string,
4667 ...
4668   "type": "http://schemas.dmtf.org/cimi/2/event/alarm",
4669   "content": {
4670     "resName": string ?
4671     "resource" : { "href" : string }, ?
4672     "resType" : string ?
4673     "code" : string,
4674     "detail" : string ?
4675   }
4676 ...
4677 }
```

4678 **XML serialization:**

```
4679 <Event xmlns="http://schemas.dmtf.org/cimi/2">
4680 ...
4681 <type> http://schemas.dmtf.org/cimi/2/event/alarm </type>
4682 <content>
4683   <resname> xs:string </resname> ?
4684   <resource href="xs:anyURI"/> ?
4685   <restype> xs:anyURI </restype> ?
4686   <code> xs:string </code>
4687   <detail> xs:string </detail> ?
4688 </content> ?
```

```
4689     ...
4690     </Event>
4691 "model" event:
4692 JSON serialization:
4693     { "id": string,
4694         ...
4695         "type": "http://schemas.dmtf.org/cimi/2/event/model",
4696         "content": {
4697             "resName": string, ?
4698             "resource" : { "href" : string }, ?
4699             "resType" : string, ?
4700             "change" : string,
4701             "detail" : string ?
4702         }
4703         ...
4704     }
```

**4705 XML serialization:**

```
4706 <Event xmlns="http://schemas.dmtf.org/cimi/2">
4707     ...
4708     <type> http://schemas.dmtf.org/cimi/2/event/model </type>
4709     <content>
4710         <resname> xs:string </resname> ?
4711         <resource href="xs:anyURI"/> ?
4712         <restype> xs:anyURI </restype> ?
4713         <change> xs:string </change>
4714         <detail> xs:string </detail> ?
4715     </content> ?
4716     ...
4717 </Event>
```

4718 "access" event:

4719 **JSON serialization:**

```

4720 { "id": string,
4721   ...
4722   "type": "http://schemas.dmtf.org/cimi/2/event/access",
4723   "content": {
4724     "operation": string,
4725     "resource" : { "href" : string },
4726     "detail" : string, ?
4727     "initiator" : string ?
4728   }
4729   ...
4730 }
```

4731 **XML serialization:**

```

4732 <Event xmlns="http://schemas.dmtf.org/cimi/2">
4733   ...
4734   <type> http://schemas.dmtf.org/cimi/2/event/access </type>
4735   <content>
4736     <operation> xs:string </operation>
4737     <resource href="xs:anyURI"/>
4738     <detail> xs:string </detail> ?
4739     <initiator> xs:string </initiator> ?
4740   </content> ?
4741   ...
4742 </Event>
```

### 4743 3.7.12 EventLog serialization

4744 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
4745 JSON and XML.

4746 **JSON media type:** application/json

4747 **JSON serialization:**

```

4748 { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLog",
4749   "id": string,
4750   "name": string, ?
4751   "description": string, ?
4752   "created": string, ?
4753   "updated": string, ?
4754   "parent": string, ?
4755   "properties": { string: string, + }, ?
```

```

4756     "targetResource": { "href": string },
4757     "events": { "href": string },
4758     "persistence": string,
4759     "summary": {
4760         "low": number,
4761         "medium": number,
4762         "high": number,
4763         "critical": number
4764     }, ?
4765     "operations": [
4766         { "rel": "edit", "href": string }, ?
4767         { "rel": "delete", "href": string } ?
4768     ] ?
4769     ...
4770 }

```

4771 **XML media type:** application/xml

4772 **XML serialization:**

```

4773 <EventLog xmlns="http://schemas.dmtf.org/cimi/2">
4774     <id> xs:anyURI </id>
4775     <name> xs:string </name> ?
4776     <description> xs:string </description> ?
4777     <created> xs:dateTime </created> ?
4778     <updated> xs:dateTime </updated> ?
4779     <parent> xs:anyURI </parent> ?
4780     <properties>
4781         <property key="xs:string"> xs:string </property> *
4782     </properties> ?
4783     <targetResource href="xs:anyURI"/>
4784     <events href="xs:anyURI"/>
4785     <persistence> xs:string </persistence>
4786     <summary>
4787         <low> xs:integer </low>
4788         <medium> xs:integer </medium>
4789         <high> xs:integer </high>
4790         <critical> xs:integer </critical>
4791     </summary>
4792     <operations>
4793         <operation rel="edit" href="xs:anyURI"/> ?
4794         <operation rel="delete" href="xs:anyURI"/> ?

```

```

4795      </operations>
4796      <xs:any>*
```

### 4798 3.7.13 EventCollection serialization

4799 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 4800 JSON and XML.

#### 4801 JSON serialization:

```

4802 { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventCollection",
4803   "id": string,
4804   "updated": string,
4805   "parent": string,
4806   "count": number,
4807   "events": [
4808     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Event",
4809       "id": string,
4810       ... remaining Event attributes ...
4811     }, +
4812   ], ?
4813   "operations": [ { "rel": "add", "href": string } ? ]
4814   ...
4815 }
```

#### 4816 XML serialization:

```

4817 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/EventCollection"
4818   xmlns="http://schemas.dmtf.org/cimi/2">
4819   <id> xs:anyURI </id>
4820   <updated> xs:dateTime </updated>
4821   <parent> xs:anyURI </parent>
4822   <count> xs:integer </count>
4823   <events>
4824     <Event>
4825       <id> xs:anyURI </id>
4826       ... remaining Event attributes ...
4827       </Event> *
4828   </events>
4829   <operations>
4830     <operation rel="add" href="xs:anyURI"/> ?
4831   </operations>
4832   <xs:any>
4833 </Collection>
```

**4834 3.7.14 EventLogCollection serialization**

4835 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
4836 JSON and XML.

4837 This Resource shall be serialized as follows:

**4838 JSON serialization:**

```
4839 { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLogCollection",
4840   "id": string,
4841   "updated": string,
4842   "parent": string,
4843   "count": number,
4844   "eventLogs": [
4845     { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLog",
4846       "id": string,
4847       ... remaining EventLog attributes ...
4848     }, +
4849   ], ?
4850   "operations": [ { "rel": "add", "href": string } ? ]
4851   ...
4852 }
```

**4853 XML serialization:**

```
4854 <Collection resourceURI="http://schemas.dmtf.org/cimi/2/EventLogCollection"
4855   xmlns="http://schemas.dmtf.org/cimi/2">
4856   <id> xs:anyURI </id>
4857   <updated> xs:dateTime </updated>
4858   <parent> xs:anyURI </parent>
4859   <count> xs:integer </count>
4860   <eventLogs>
4861     <EventLog>
4862       <id> xs:anyURI </id>
4863       ... remaining EventLog attributes ...
4864       </EventLog> *
4865     </eventLogs>
4866     <operations>
4867       <operation rel="add" href="xs:anyURI"/> ?
4868     </operations>
4869     <xs:any>*
4870   </Collection>
```

**4871 3.7.15 EventLogTemplate serialization**

4872 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
4873 JSON and XML.

4874 **JSON media type:** application/json

4875 **JSON serialization:**

```
4876 { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLogTemplate",
4877   "id": string,
4878   "name": string, ?
4879   "description": string, ?
4880   "created": string, ?
4881   "updated": string, ?
4882   "parent": string, ?
4883   "properties": { string: string, + }, ?
4884   "targetResource": { string },
4885   "persistence": string,
4886   "operations": [
4887     { "rel": "edit", "href": string }, ?
4888     { "rel": "delete", "href": string } ?
4889   ] ?
4890   ...
4891 }
```

4892 **XML media type:** application/xml

4893 **XML serialization:**

```
4894 <EventLogTemplate xmlns="http://schemas.dmtf.org/cimi/2">
4895   <id> xs:anyURI </id>
4896   <name> xs:string </name> ?
4897   <description> xs:string </description> ?
4898   <created> xs:dateTime </created> ?
4899   <updated> xs:dateTime </updated> ?
4900   <parent> xs:anyURI </parent> ?
4901   <properties>
4902     <property key="xs:string"> xs:string </property> *
4903   </properties> ?
4904   <targetResource href="xs:anyURI"/>
4905   <persistence> xs:string </persistence>
4906   <operations>
4907     <operation rel="edit" href="xs:anyURI"/> ?
4908     <operation rel="delete" href="xs:anyURI"/> ?
4909   </operations>
```

```

4910      <xs:any>*
4911  </MeterTemplate>

```

### 3.7.16 EventLogTemplateCollection serialization

4913 The following pseudo-schemas (see notation in 1.3) describe the serialization of the Resource in both  
 4914 JSON and XML.

4915 This Resource shall be serialized as follows:

#### 4916 JSON serialization:

```

4917  { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLogTemplateCollection",
4918    "id": string,
4919    "updated": string,
4920    "parent": string,
4921    "count": number,
4922    "eventLogTemplates": [
4923      { "resourceURI": "http://schemas.dmtf.org/cimi/2/EventLogTemplate",
4924        "id": string,
4925        ... remaining EventLogTemplate attributes ...
4926      }, +
4927    ], ?
4928    "operations": [ { "rel": "add", "href": string } ? ]
4929    ...
4930  }

```

#### 4931 XML serialization:

```

4932  <Collection
4933    resourceURI="http://schemas.dmtf.org/cimi/2/EventLogTemplateCollection"
4934    xmlns="http://schemas.dmtf.org/cimi/2">
4935    <id> xs:anyURI </id>
4936    <updated> xs:dateTime </updated>
4937    <parent> xs:anyURI </parent>
4938    <count> xs:integer </count>
4939    <eventLogTemplates>
4940      <EventLogTemplate>
4941        <id> xs:anyURI </id>
4942        ... remaining EventLogTemplate attributes ...
4943        </EventLogTemplate> *
4944      </eventLogTemplates>
4945      <operations>
4946        <operation rel="add" href="xs:anyURI"/> ?
4947      </operations>

```

```
4948     <xs:any>*
4949   </Collection>
```

## 4950 3.8 Other Examples of serialized collections

4951 The following are examples of how collections of Resources are serialized. The first one illustrates  
4952 a basic collection of Machine resources. The second one illustrates an “enhanced” collection of  
4953 Volume resources, i.e., with additional (accessory) attributes for each Volume item, besides those  
4954 defined in the Volume resource type.

### 4955 3.8.1 Machine Collection

4956 The Resource type for each item of this Collection is “Machine”. There is no accessory attributes in this  
4957 Collection, which is then called a “basic” Machine Collection. In the example below, each Machine item in  
4958 the Collection is not expanded except for its common attributes. An expanded serialization showing all or  
4959 parts of each Machine is also an option – this may vary depending on what selection has been used  
4960 when querying Resources of this type (subsetting with \$select attribute).

#### 4961 JSON serialization:

```
4962 { "resourceURI": "http://schemas.dmtf.org/cimi/2/MachineCollection",
4963   "id": string,
4964   "updated": string,
4965   "parent": string,
4966   "count": number,
4967   "Machines": [
4968     { "resourceURI": "http://schemas.dmtf.org/cimi/2/Machine",
4969       "id": string,
4970       "name": string,
4971       "description": string,
4972       "created": string,
4973       "updated": string,
4974       "parent": string,
4975       "properties": { string: string, + },
4976       "machine": { "href": string },
4977       "operations": [
4978         { "rel": "edit", "href": string },
4979         { "rel": "delete", "href": string }
4980       ]
4981     ...
4982   }, +
4983 ],
4984   "operations": [
4985     { "rel": "add", "href": string }
4986     { "rel": "insert", "href": string }
4987     { "rel": "remove", "href": string }
```

```
4988     ]
4989     ...
4990 }
```

#### 4991 XML serialization:

```
4992 <Collection
4993   resourceURI="http://schemas.dmtf.org/cimi/2/MachineCollection"
4994   xmlns="http://schemas.dmtf.org/cimi/2">
4995   <id> xs:anyURI </id>
4996   <updated> xs:dateTime </updated>
4997   <parent> xs:anyURI </parent>
4998   <count> xs:integer </count>
4999   <Machines>
5000     <Machine>
5001       <id> xs:anyURI </id>
5002       <name> xs:string </name>
5003       <description> xs:string </description>
5004       <created> xs:dateTime </created>
5005       <updated> xs:dateTime </updated>
5006       <parent> xs:anyURI </parent>
5007       <property key="xs:string"> xs:string </property> *
5008       <machine href="xs:anyURI"/>
5009       <operation rel="edit" href="xs:anyURI"/>
5010       <operation rel="delete" href="xs:anyURI"/>
5011       <xs:any>*
5012     </Machine> *
5013   </Machines>
5014   <operation rel="add" href="xs:anyURI"/>
5015   <operation rel="insert" href="xs:anyURI"/>
5016   <operation rel="remove" href="xs:anyURI"/>
5017   <xs:any>*
5018 </Collection>
```

#### 5019 3.8.2 Volume Collection in a Machine

5020 The Resource type for each item of this Collection is “Volume”. The initial location of these Volumes (not  
 5021 part of Volume attributes) is added as an accessory attribute to each Collection item. In the example  
 5022 below, each Volume item in the Collection is not expanded except for its common attributes. An  
 5023 expanded serialization showing all or parts of each Volume is also an option.

5024 Note that the last part of the Collection resourceURI is not just VolumeCollection but  
 5025 locatedVolumeCollection, in order to denote the addition of an accessory attribute  
 5026 (initialLocation) to each Resource item of type Volume.

5027 **JSON serialization:**

```
5028 { "resourceURI": "http://schemas.dmtf.org/cimi/2/locatedVolumeCollection",
5029   "id": string,
5030   "updated": string,
5031   "parent": string,
5032   "count": number,
5033   "locatedVolumes": [
5034     { "resourceURI": "http://schemas.dmtf.org/cimi/2/locatedVolume",
5035       "id": string,
5036       "name": string,
5037       "description": string,
5038       "created": string,
5039       "updated": string,
5040       "parent": string,
5041       "properties": { string: string, + },
5042       "initialLocation": string,
5043       "volume": { "href": string },
5044       "operations": [
5045         { "rel": "edit", "href": string },
5046         { "rel": "delete", "href": string }
5047       ]
5048       ...
5049     }, +
5050   ],
5051   "operations": [
5052     { "rel": "add", "href": string }
5053     { "rel": "insert", "href": string }
5054     { "rel": "remove", "href": string }
5055   ]
5056   ...
5057 }
```

5058 **XML serialization:**

```
5059 <Collection
5060   resourceURI="http://schemas.dmtf.org/cimi/2/locatedVolumeCollection"
5061   xmlns="http://schemas.dmtf.org/cimi/2">
5062   <id> xs:anyURI </id>
5063   <updated> xs:dateTime </updated>
5064   <parent> xs:anyURI </parent>
5065   <count> xs:integer </count>
```

```
5066      <locatedVolumes>
5067          <locatedVolume>
5068              <id> xs:anyURI </id>
5069              <name> xs:string </name>
5070              <description> xs:string </description>
5071              <created> xs:dateTime </created>
5072              <updated> xs:dateTime </updated>
5073              <parent> xs:anyURI </parent>
5074              <property key="xs:string"> xs:string </property> *
5075              <initialLocation> xs:string </initialLocation>
5076              <volume href="xs:anyURI"/>
5077          <operations>
5078              <operation rel="edit" href="xs:anyURI"/>
5079              <operation rel="delete" href="xs:anyURI"/>
5080          </operations>
5081          <xs:any>*
5082      </locatedVolume> *
5083  </locatedVolumes>
5084  <operations>
5085      <operation rel="add" href="xs:anyURI"/>
5086      <operation rel="insert" href="xs:anyURI"/>
5087      <operation rel="remove" href="xs:anyURI"/>
5088  </operations>
5089  <xs:any>*
5090 </Collection>
```

**ANNEX A  
(informative)****Change log**

| Version | Date       | Description   |
|---------|------------|---|
| 1.0.0   | 2015-08-28 |   |
| 1.0.1   | 2015-09-12 | Errata  |
| 1.1.0   | 2014-06-09 | This version contains the following updates: <ul style="list-style-type: none"><li>• include network scenario</li><li>• updated to old version of the network writeup</li><li>• updated to include clause 8 for Mantis 2347 and Mantis 2095</li><li>• Dies editorial comments, fixed future tense.</li><li>• Arturo updates folded in.</li><li>• delete subclause on Add Network Port; fixed example in 7.4</li><li>• Output of first editing session on June 2, 2014</li><li>• Dies added a network example</li><li>• Output of last editing session on June 9, 2014</li></ul> |
| 2.0.0   | 2016-08-04 | Upgraded to match CIMI 2.0 changes, added model information (section 2), imported pseudo-schema for Resource serialization from the CIMI specification (section 3).   |